Feature Extraction and Object Recognition using Conditional Morphological Operators

Stephen John Rees

A thesis submitted in partial fulfilment of the requirements of the University of Glamorgan for the degree of Doctor of Philosophy

June 1997

Department of Electronics and Information Technology,
The University of Glamorgan,
Pontypridd,
Mid Glamorgan,
CF37 1DL

Acknowledgements

I would like to thank my Director of Studies, Dr B. F. Jones for his assistance, comments, suggestions, constructive criticism, and general support throughout the course of this work.

I would also like to thank the following people for constructive suggestions and occasional aid, particularly with the more obscure tools, and syntax with document preparation: Mr M. A. McCabe, Dr P. Plassman, Mr T. D. Jones.

My thanks, also, to the Department of Electronics and Information Technology, for the provision of the tools needed to complete this work.

Abstract

Feature Extraction and Object Recognition using Conditional Morphological Operators

Stephen John Rees

This thesis describes the work undertaken on morphological operators for feature extraction from, and segmentation and recognition of, objects within single 2-D images under loosely controlled conditions.

The novel aspects of the work include the development of a conditional morphological operator, the RJ operator, providing a direct measure of the occupancy of one set by another. This was then applied to the direct extraction of structural features from the intensity map in greyscale images, and to the recognition of objects within images using these features. More complex algorithms for feature identification and object recognition, including a mostly hit, mostly miss transform (MHMMT) and a multiscale structural analysis were developed, using occupancy as the metric. The performance and characteristics of these methods were investigated, using a symmetrical probe as the main tool for analysis and manufactured and natural objects as test pieces.

Structural features were used as local descriptors of objects. These were extracted by four methods: edge following, chain coding and curvature estimation; direct probing with the R operator and templates; direct probing with the MHMMT; and a generalised R analysis, a multiscale intersection of R operator templated results. The selectivity of the techniques varied, the MHMMT producing the greatest rejection of data. The generalised R analysis produced the most accurate location of features.

Two methods were adopted to interrelate the extracted features. The first produced a sequence of perimeter features, by estimation of their relative rotations about a calculated feature centroid. The second method interrelated the features as a web skeleton, listing the orientations of each feature relative to the others in the set.

The multivalued function form of the RJ operator was used to identify the specific object from a model library of poses of various objects. Different combinations of the techniques for extraction and modelling were compared. All objects were recognised, and their orientation determined with errors of between 15 and 25 degrees in the worst cases.

Contents **Page**

Appendices

List of Figures

List of Tables

Chapter 1: Introduction and Literature Survey

Chapter 1: Introduction and Literature Survey

1.1 Introduction

The computer recognition of objects through image processing and image analysis is an area of research which has received enormous interest over many years. Yet it is only in the last few years that the necessary hardware and software has become available at reasonable cost, opening a whole host of new applications for image analysis. This diversity of applications has extended the boundaries of algorithm development, and highlighted the deficiencies of current technology. Vision is probably the strongest of the five senses, and therefore the most instinctive and natural to people. It is therefore an unpleasant surprise to users to realise the lack of flexibility of recognition systems, particularly when object recognition seems such a simple task. The great majority of recognition systems are task specific and goal orientated, created for the solution of a specific problem under controlled conditions.

The basic aims of this project were the recognition of objects from single greyscale images, the extraction of object features for recognition, and the identification of texture in texture samples.

The method of recognition adopted for this work uses a local feature descriptor approach, based on structural features. The reasons for selection of this approach are discussed in chapter 3, and relate to its inherent flexibility. The method requires appropriate, extractable feature descriptions, and this fact guides the development of the algorithms for its implementation. The basic problem is that of extracting a good set of appropriate features for the description of the object, allowing comparison with a stored reference equivalent model. Structural features are commonly chosen for local descriptors. They are conceptually easy to understand (for example convex and concave corners, curves, and straight edges on the perimeter of segmented object images) and have a well-understood relationship in terms of perceptual significance (Larry S Davis, "Understanding Shape"[32]).

Structural feature extraction techniques require considerable volumes of processing, often of a sequential nature (chain coding/curvature estimation, polygonal approximation [36, 38, 42, 43, 44]). In the case of the relationship based methods, the benefits of parallel implementation are limited, and the task partitioning is likely to require sophisticated reasoning. Although useful and appropriate to many situations, such methods are difficult where speed is essential to the processing task. For example, the conversion of a guided missile into a "hitile" has been suggested to require a frame processing speed of four hundred frames per second with current technology. Any optical guidance system based on machine vision would need to acquire the image and extract the object, recognise it and estimate rotation, scaling and translation (position) at a viable frame rate. Added to the basic processing problem, the object might offer rapid aspect change as proximity increases, exacerbating the difficulty of using global descriptors.

The extraction of the feature description is, therefore, one of the determinants of the ability to perform the required task. Any processing algorithm takes time to run, and in order to minimise image processing time it follows that an absolute minimum of pre-processing can be used. Some of the basic operations can be pipelined and performed adaptively (histogram equalisation, binarisation) on the input data stream. These operations typically precede cleaning and extraction. The more parallel the implementation of these processes can be made, the more rapid the overall task operation.

More severe problems arise in optical textural analysis. Textured regions may not be separable by simple methods - different textures may exhibit the same average intensity, and overlapping or similar intensity histograms. The algorithms for resolving texture are essentially dependant on area statistics - and are therefore more processing intensive. It is not possible to simplify the area processing in its initial stages - the feature vectors must be generated. Their later use is typically in a matching process, reliant upon multiple feature aggregation. Common methods include: minimum distance classification; mahalanobis distance; and various Bayesian statistical approaches which use weighted aggregation of feature vectors as a decision classifier. The objective is to assign a sample to a particular class of texture. A variety of feature generation schemes are used, exemplified by the statistical classifiers. The first order methods, such as mean, variance, kurtosis, skewness and entropy, are heavily reliant on illumination. The second order methods include co-variance, and estimates based on co-occurrence matrices. They are based on spatial interrelationships, directionally, about a point in the image. In general they are floating point intensive calculations, and, as a result, relatively slow.

In the context of image analysis, the operators of mathematical morphology have much to recommend them. They are essentially integer in nature; they are relatively simple to calculate; their composition from simple processing operators lead to an efficient hardware implementation; and they offer versatile tools for a variety of necessary operations. They produce results based on the relationship between the geometrical shape of the object to be analysed and a structural probe or structuring element. The structuring element is chosen for some desirable relationship between its geometry and the proposed goal in the modification of the image. As we shall discuss later, the morphological operators lend themselves to a cellular implementation [6], offering the potential for high parallelism in use. Their application to the various facets of image analysis has been undertaken, as we shall now describe in the context of the existing literature.

1.2 On Mathematical Morphology

Mathematical morphology in the processing of digital images has found extensive application in filtering, smoothing, enhancement, representation and recognition. As might be anticipated from the wide variety of applications, a large body of research exists into the theory and usage of morphological operators in image processing.

Mathematical morphology is based on the idea of shape as a fundamental principle of a set or function. The basic principles were expounded by H. Minkowski in 1903 as applied to set theory. The primary operators of set erosion and dilation are defined little differently from the Minkowski addition and subtraction operators, from which they are derived.

Binary set dilation is defined by:

$$A \oplus B = \{c \in E^n, a \in A, b \in B \mid c = a+b\}$$

or

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

Binary set erosion is defined by:

$$A \ominus B = \{x \in E^n \mid (x+b) \in A \ \forall b \in B\} \qquad [1]$$

or

$$A \Theta B = \bigcap_{b \in B} (A)_{-b} \qquad [2]$$

where a,b are members of A and B respectively, E^n refers both to Euclidean N-space or its discrete equivalent, the set of N-tuples of integers, $(A)_b$ represents the translation of set A by element b, and $A,B \subset E^n$. The development of morphology as a tool for image processing and pattern recognition can be attributed to G Matheron [2] and J Serra [3]. The definition used here is adopted from the tutorial paper of Haralick et al [4], as do the simple examples used below.

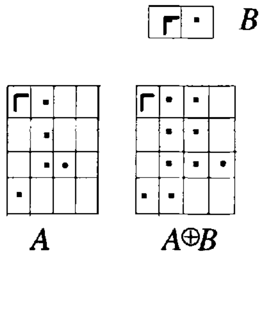
Binary Set Dilation and Set Erosion Examples

Consider the example below, of a binary 2-D set A dilated by a 2-D structuring element set B. The set contents refer simply to the co-ordinates of the elements within them.

Binary Set Dilation Example

A={(0,1),(1,1),(2,1),(2,2),(3,0)}

B={(0,0),(0,1)}



$A \oplus B$={(0,1),(1,1),(2,1),(2,2),(3,0),(0,2),(1,2),(2,2),(2,3),(3,1)}

The physical effect of dilation is to enlarge the initial set, by adding components to it. A conceptual model might be to place the origin of the structuring element at each point in the set A, and add any components not already contained within that set.

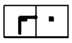
Binary Set Erosion Example

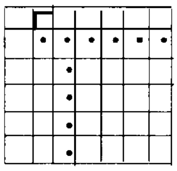
This is a simple example of binary set erosion of a set A by a structuring element set B, as shown below.

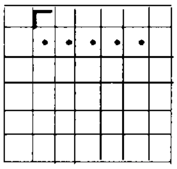
$A=\{(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(3,1),(4,1),(5,1)\}$

$B=\{(0,0),(0,1)\}$

$A\Theta B=\{(1,0),(1,1),(1,2),(1,3),(1,4)\}$


B


A


AΘB

The physical effect is to shrink the original set A. Now define the containment of one set within another as occurring at a location where all its elements have corresponding elements. A conceptual model might consist of placing the origin of the structuring element B at each point in set A, and taking only the points where all the elements of B are fully contained in A as the result. The result would be the erosion of A by B.

<u>The Work of Matheron</u>

Matheron first studied the concepts of opening and closing, these being secondary operations defined as the sequential dilation of a set followed by erosion, and erosion followed by dilation, by the same template set, respectively.

Opening is defined as:

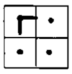
$B \circ K = (B \ominus K) \oplus K$

Closing is defined as:

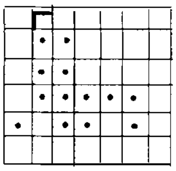
$B \bullet K = (B \oplus K) \ominus K$

The physical interpretation of these operators is shown below, by the opening and closing of a 2-D binary set A by a 2-D structuring element *B*.

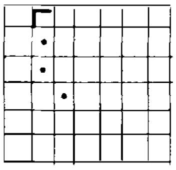
Examples of Binary Opening and Closing of a Binary Set



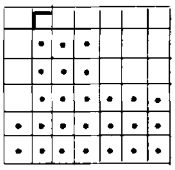
B



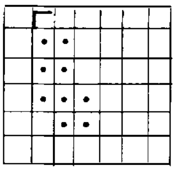
A



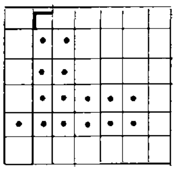
$A \ominus B$



$A \oplus B$



$A \ominus B \oplus B$
$= A \circ B$



$A \oplus B \ominus B$
$= A \bullet B$

Note that opening tends to sever thin isthmuses and remove small structures, whereas the closing fills small holes about the boundary (or within) the set.

The iterative sequential applications of dilations and erosions produce idempotent image transformations: further applications will produce no change in the result. Matheron was primarily concerned with sizes and granularities, and used the opening and closings to obtain sieves for granulometries.

In the Euclidean case, granulometry size distributions were obtained by finding areas resulting from granulometries.

$$u_A(t) = m[(A \circ tE)]$$

where $t > 0$, m = Lebesque measure (area), A = compact set, E =convex compact set, and t= scalar multiplier (scaling factor) for E. tE, therefore, is a scaled version of E.

In the digital case, the digital linear geometric size distribution can be generated using $v(k)$ and $h(k)$ as probes:

$$v(k) = \begin{pmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{pmatrix}, \quad h(k) = (1\,1 \cdots 1)$$

The measures then become:

$$w_1(k) = CARD\big[(S \circ v(k))\big]$$
$$w_2(k) = CARD\big[(S \circ h(k))\big]$$

where:

S = structure under test;
$CARD$ = cardinality of the resulting set, i.e. how many elements it contains.

The result of application of these methods is a characteristic distribution or signature, which is used to analyse the structure of the image under test. This is effective in use with textured data.

Size distributions can be extracted using erosion rather than opening. The method uses the horizontal and vertical covariance functions. Digitally, these may be represented as

$$v(k) = \begin{pmatrix} 1 \\ * \\ \cdot \\ \cdot \\ 1 \end{pmatrix}, \quad h(k) = (1{*}{*}{\cdot\cdot}{*}1)$$

Then

$$w1(k) = CARD\left[\left(S\Theta v(k)\right)\right]$$
$$w2(k) = CARD\left[\left(S\Theta h(k)\right)\right]$$

The symbol * is here used to represent the background value of the probe structuring element, chosen so as to give no real values for locations corresponding to the pixels separating the outliers of the probe structure (often set to -∞). This establishes the covariance function.

Giardina and Dougherty[5] describe a stochastic approach to size distribution, based closely on Matheron's methods, in which the signature of the image is extracted under sequential analysis by paradigm structuring elements. The set of structuring elements $E_1,E_2,\ldots,E_m$ are predetermined, the collection of which play (quoted from Matheron) an "a priori constitutive role" in the definition of structure. The geometric probability for a particular structuring element is defined as:

$$P(E \subset S) = \frac{CARD\left(S\Theta E_k\right)}{CARD(S)}$$

When applied for a set of elements $E_1,E_2, \ldots, E_k, \ldots, E_m$, this generates the probability vector shown below:

$$\overline{P}(E_k \subset S) = \begin{pmatrix} P_1 \\ \cdot \\ \cdot \\ P_m \end{pmatrix}, k \in (1,m)$$

The distribution of the probabilities P_k offers a characteristic signature of the image under test against the probe elements. The determinant of the success of the approach is the correct selection of the structuring elements. Their "a priori constitutive role" should be to offer the maximum separation between the classes of images analysed with the minimum number of structures and repeated applications.

The Work of Serra

The work of Serra [3] underpins much of the modern application of mathematical morphology, including the definition of the Hit and Miss Transform and the application of morphology to binary images using spatial probe sets and functions. Serra was concerned primarily with shape and connectivity measures. In the area of grey scale morphology Serra extended the Hit or Miss transformation and the size distribution work from sets to functions. He describes the use of grey-tone morphology to analyse functions in terms of topological features, and indicates practical results obtained both by himself and others in picture enhancement and cleaning.

The Work of Sternberg

S R Sternberg [6,7] introduces the general extension of mathematical morphology to greyscale images, through the use of localised min and max operations. This approach uses the idea of the top surface of the umbra of the function as the operable surface, yielding the following definitions. (The meanings of top surface and umbra are illustrated in figure (1) overleaf. The concept of the rolling ball transform, the points described by a rolling ball moving over the surface of the function defining the dilated function, as the means of calculating dilation by a circular function is due to Sternberg.

The greyscale dilation (2-D) of a function $f(x,y)$ by a function $g(i,j)$ is defined as:

$$(f \oplus g)(x,y) = TOP[U[f(x,y)] \oplus U[g(i,j)]]$$

$$(f \oplus g)(x,y) = \max_{(i,j) \in g(i,j)} \{ f(x-i, y-j) + g(i,j) \}$$

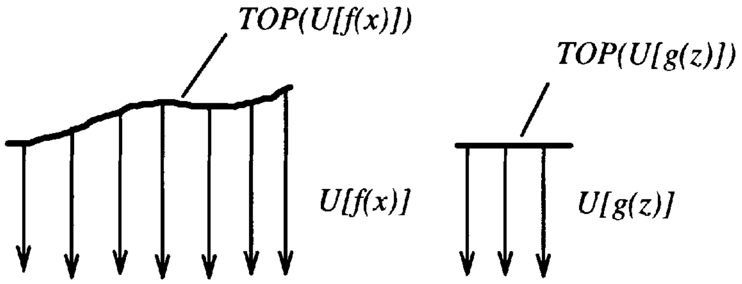
where:

TOP = topsurface of the function;

U = umbra of the function (see figure (1) below);

max = maximum operator, used to define the peak value of the equation over the region $(i,j)\in g(i,j)$.

Figure (1): Umbra and Topsurface of Functions



Taken together, $TOP[U[f(x,y)]] = f(x,y)$.

Greyscale erosion is defined as:

$$(f\Theta g)(x,y) = TOP[U[f(x,y)]\ \Theta\ U[g(i,j)]],$$

and

$$(f\Theta g)(x,y) = \min_{(i,j)\in g(i,j)}\{ f(x+i,y+j) - g(i,j) \}$$

where min = minimum operator, used here to generate the least value of the equation over the given region defined by $(i,j)\in g(i,j)$.

The full set of operations and conditions defined by Serra and Matheron are applicable equally to greyscale processing (see Haralick et al [4]).

H Blum: The Medial Axis Transform

In the search for global descriptors of shape many methods have been used. The medial axis transform [8] basically requires the placement of a maximally sized disk at points within the shape under test. The point belongs to the result if it is one at which a locally maximal disk fits. Implementation of this algorithm through closing is an obvious approach first described by Maragos [9], yielding a robust, rotation invariant 2-D shape analysis technique. A small structuring element is successively applied to the image under test. In the limit, the skeleton left is the medial axis transform of the shape under test. Descriptions of this work include [10] and [11]. It must be noted that pre-processing is required to obtain successful results, which can form a significant proportion of the recognition algorithm work under uncontrolled conditions.

The Work of Bronskill and Venetsanopoulos: The Pecstrum

The pecstrum is a means of quantitatively assessing the geometrical structure of multidimensional signals first defined by Bronskill and Venetsanopoulos [12]. This is defined, for positive valued multidimensional discrete functions f whose umbra is a subset of $E = Z^{M+1}$ as:

$$P(n) = [Mes(f \circ ng) - Mes(f \circ (n+1)g)]/Mes(f)\,,\ n = 0,\ 1,\ 2,\ldots..,\ +\infty,$$

where

Mes(f) denotes the measure of f, which is defined to be area, volume, etc under f for 1D, 2D, etc, functions respectively.

$f \circ ng$ denotes the morphological opening of f by the structuring element ng.
ng represents the function g dilated by itself n times

The paper includes some sample data for 2-D binary and 2-D multilevel shapes, and for 3-D binary object images. The shapes and objects chosen were distinctly different and, in general, characteristically geometrical.

The Work of Maragos: The Pattern Spectrum

Maragos defined a global descriptor, the pattern spectrum, applicable to 2-D shapes. The pattern spectrum offers a means of quantifying aspects of the shape-size content of a signal (image) using successive openings or closings at different scales to remove information from the image. The difference between successive iterations is used as the primary image information, yielding a size spectrum of the pattern content of the function under test relative to the probe function (see [9], pp706 - 710).

The pattern spectrum of an image function is defined as shown below.

Pattern Spectrum for Discrete Binary Images

The pattern spectrum of X is the non negative function

$$PS_X(+n,B)= A[(X \circ nB) - (X \circ (n+1)B],\ n>=0$$

$$PS_X(-n,B)=A[(X \bullet nB) - (X \bullet (n-1)B)],\ n>=1$$

These are calculated using the forward area difference.

Pattern Spectrum for Graytone Images (Multilevel Signals)

The pattern spectrum of $f(x,y),\ (x,y)\in Z^2$ relative to a discrete graytone pattern g is given by the non-negative function

$$PS_f(+n,g)= A[f \circ ng - f \circ (n+1)g],\ 0<=n<=N$$

$$PS_f(-n,g)= A[f \bullet ng - f \bullet (n-1)g],\ 1<=n<=K$$

where:

$A(f) = \sum_{(x,y)} f(x,y)$
$(a-b)(x) = a(x) - b(x)$
$N = (\max n \mid f \circ ng \mathrel{!}= -\infty\ \forall(x,y) \in f(x,y))$
$K = (\min n \mid U(f \bullet ng) = \lim_{n\to\infty} U(f \bullet ng) \subseteq CH[U(f)])$
CH = Convex hull of the function

Maragos expands these ideas to include orientation by using the idea of a unit length line segment orientated at an angle to the axes as the probe structure.

It should be noted that reconstruction from the pattern spectrum is not generally possible because of the risk (in an uncontrolled world model) of two different images with similar pattern spectra.

A Comparison of the Pecstrum and the Pattern Spectrum

The pecstrum (Brontskill and Venetsanopoulos) describes a shape in terms of its decomposition under opening, based on the difference values of some generalised measure Mes(X), made on each subsequent opened version of the image. The sequential opening progressively removes larger features from the image. This difference value is normalised by the same measure applied to the original image.

The pattern spectrum (Maragos) describes shape, again, in the form of a decomposition. He describes the non-normalised use of both successive opening and successive closing to generate a characteristic signature of the image. The particular measure used is the area of the residual image. In the binary case, this is defined as the residual pixel count following each opening (closing). The differences in this measure between successive openings (closings) defines the pattern spectrum. The idea of directionality in the analysis is introduced by Maragos, through the use of orientated line segments as structuring elements.

Several other researchers have reported on the application of global descriptors to information extraction using morphological operators, as is briefly summarised below.

Esselman and Verly [13] describe the use of structuring elements for extraction of appendages and corners from 2-D range images with a modest degree of noise. Additional results for simulated 3-D data were obtained. Corners were extracted by combining a rolling ball transform approach to extracting external (silhouette) corners with a hit or miss transform using an extensive set of bit patterns to indicate in-silhouette "triple" corners (i.e. corners where three planes join in the object). Appendages are found by subtracting the original image from the closing of the same image by a hemispherical structuring element at various scales.

Shih and Mitchell [14] describe the use of greyscale morphological techniques for the skeletonisation and distance transformation of objects (machine tools are used as the examples). Three structuring elements are used, the Euclidean distance, the City-block distance and the Chessboard distance, each of which can be of any odd number size. The image under test is successively eroded until no further change is obtained and a result obtained for each structuring element. Generation of the Medial Axis Transform using distance transformation is also described.

Heijmans and Ronse [15] describe an algebraic approach to mathematical morphology based on lattices. Although no practical work is indicated, this paper is of interest for its theoretical development in a formal mathematical framework.

Pitas and Venetsanopoulos [16] describe an approach to morphological shape decomposition. The requirement is to decompose a binary shape into a union of simple binary shapes, obtaining a rotation, translation and scale invariant result with a unique form for each shape. The problem requires the identification of a set of open sets $\{X_1, \ldots ,X_n\}$ whose union is the original set X:

$$X = \bigcup_{i=1}^{n} X_i$$

The sets X_i are defined as

$$X_i = r_i B,$$
$$or$$
$$X_i = B \oplus B \oplus B \cdots \oplus B \text{ } (r_i \text{ times})$$

The decomposition algorithm described takes the following form:

$$X_i = (X - X'_{i-1}) \circ r_i B$$

$$X'_i = \bigcup_{0<j<=i} X_j$$

$$X'_0 = \varnothing$$

The locus of the centres of the maximal inscribable objects r_iB is given by

$$L_i = (X - X'_{i-1}) \Theta r_i B$$

This algorithm has been successfully applied to complex shapes with a significant compression of information for storage or transmission of the shape in a recognisable form. The theoretical extension to application of similar methods for 3-D objects is described, with no practical results.

This brief summary is meant by no means to be an exhaustive list of skeletonisation applications in recent times.

Schonfeld and Goutsias [17] describe the use of the set-difference distance function and mean difference function as quantitative measures of the degree of geometrical and topological distortion introduced by morphological filtering. A minimax estimation procedure for obtaining an optimal alternating sequential filter for eliminating degradation noise while preserving geometrical and topological features is proposed.

Morphological Decomposition

The use of morphological filters to decompose an image into its primal parts was initially proposed by Matheron, in the context of granulometries. More recently, research has focused on the use of multiresolution image representations, where an iterative application of morphological filters of increasing size is used to extract geometrical structure information about the image under analysis.

Toet [18] describes the use of greyscale morphology to decompose the structural pattern by subtraction of successive layers in the multiresolution representation. The resulting code elements are tightly located and sized. By increasing the size of the structuring element successively larger image details are omitted from the result. The resulting morphological lowpass pyramid is a complete representation of the original image, provided that flat structuring elements are used (non-flat structuring elements, such as lozenges and cylinders, maintain the vertical boundaries in the resulting transformed images). As such, the image can be reconstructed from the description produced, using a corresponding morphological bandpass filter.

More recently, Wang, Haese-Coat, Bruno and Ronsin [19] have reported the use of iterative morphological decomposition for texture analysis. Again, the entire texture image can be reconstructed by the sum of all its component images, through the use of plane (i.e. flat) structuring elements. Successful classification of textures, and of texture boundaries where textures abut, were reported with the use of relatively few texture features. The basic process relies on classifying a window of a given size as belonging to a particular texture region, based on a Bayesian conditional approach.

The decomposition procedure is described using the iterative formulation shown below. The series of component images, $s_i(x,y)$, is the decomposed result.

$$\begin{aligned} &f_0(x,y) = f(x,y) \\ &s_i(x,y) = (f_i \circ B_{n-i})(x,y), \quad i = 0,1,\ldots,n. \\ &f_{i+1}(x,y) = f_i(x,y) - s_i(x,y) \end{aligned}$$

This is summarised as a multisize or multiscale morphological decomposition MMD^i[], where:

$$s_i(x,y) = MMD^i[f(x,y), B_j \mid j = 0,1,\ldots,n], i \in 0,n$$

It is clear that limiting the sizes of structuring elements applied to a necessary minimum set will improve the computational efficiency of the method. They report good classification results based on the use of five vertical and five horizontal line structuring elements, classifying up to 99 percent correctly samples from a chosen set of nine textures, with a window size of sixteen by sixteen pixels.

Morphological Clustering for Pattern Classification

Postaire, Zhang and Lecocq-Botte [23] describe an approach to unsupervised pattern classification based on the use of binary morphology. They suggest the use of a 3x3x..x3 n-dimensional hyper cube structuring element (as appropriate to the dimensionality of the variate data) and opening/closing sequences to segment the cluster data. Practical evaluations against the Isodata and K-Means methods are included, showing an at least comparable performance for the data sets chosen.

Novel Filter Design Approaches

Harvey and Marshall [24] describe the use of artificial neural networks (ANNs) to automate the design process for morphological filters for particular tasks, including soft and rank order filters. Matsopoulos and Marshall [25] describe the use of morphological processing techniques for biological measurements from ultrasound images

The Problems with Mathematical Morphology

Several problems of using the morphological operators are evident in the work described above. These relate primarily to the precision of the structural relationships required between the structuring element and the data in the set under test. These are summarised as follows:

(i) The erosion operator requires that the set or function under erosion must have a complete containment of the structuring element for a point to belong to the result (binary), or to belong to the result unmodified (greyscale);

(ii) the filters are therefore highly sensitive to noise and defects in the image;

(iii) the filtered results tend to exhibit the geometry of the structuring element about the perimeter of significant objects - notably a "blockiness" where area filling is an objective (typically achieved using closing);

(iv) useful perimeter data may be eradicated by the use of morphological cleaning (as also occurs with median and other rank order filters);

(v) the perimeter of the area of interest is moved by filtering;

(vi) pre-filtering may be used to limit the likelihood of defects and noise in the image, resulting in further moving of object boundaries;

(vii) salt and pepper (impulsive) noise can cause difficulties with erosion/dilation couplets.

It is to these problems that the work in this thesis addresses itself, in the context of the object recognition and feature extraction problems.

Several methods have been developed recently to alleviate, or at least to minimise, these problems.

Fuzzy Morphology: The Work of Sinha and Dougherty

The raising of morphology to a fuzzy definition is a relatively recent development. It has obvious implications for the inclusion of a relatively large body of modern fuzzy set theory into the application of morphology for image analysis. Sinha and Dougherty [20] derive a fuzzy morphology based on a fuzzy index function, permitting the definition of the various morphological operators and the corresponding Minkowski algebra in fuzzy terms. The fuzzy index function I is defined as:

$$I(A,B)=\inf_{x\in U}\mu_{A^c\Delta B}(x)$$

This measures the belief in the proposition "A is a subset of B", with A and B fuzzy sets. Using the prefix form **operator(set1, set2)**, the fuzzy erosion, $E(A,B)$, and dilation, $D(A,B)$, operators are defined as:

$$\mu_{E(A,B)}(x)=I(\mathrm{T}(B;x),A)$$

and

$$\mu_{D(A,B)}(x)=\mu(E(A^c,-B)^c)(x)$$

where:

$T(B;x)$ = translation of B by x;

$I(T(B;x),A)$ = index function measure of fuzzified set inclusion;

μ(x) = membership function of the fuzzy set;

Δ = bold union of two sets;

U = Euclidean plane or Cartesian grid (universe of discourse).

Using the notation adopted in this document,

$$\mu_{(A\Theta B)}(x)=I[(B)_x\cap A]$$

where I = index function for fuzzified set inclusion,

and $\mu_{(A\oplus B)}(x)=\mu_{(A^c\Theta-B)}(x)$, derived from the erosion-dilation duality theorem.

Practical application of this work, in the form of fuzzy morphological filters is given in [21] with practical results for object detection and enhancement in a limited set of constrained conditions.

Koskinen, Astola and Neuvo [22] describe a rank order approach to evaluating erosion and dilation, based on the idea of using some other function than the max and min for obtaining a greyscale erosion (dilation) result. This is found to have improved the performance of the morphological filters in the presence of noise to some extent. As is noted in [20], this approach results in an opening and closing that are neither idempotent nor anti-extensive and extensive, respectively. This work is described in more detail in Chapter 2, in comparison with the methods derived in this project.

We ourselves (Rees, Jones [26, 27]) describe a conditional approach to morphology, essentially loosening the erosion process by allowing known levels of uncertainty (defined in pixel count) in the determination of the erosion marker. This work forms the basis of Chapter (2) of this thesis.

Aims of the Project

The aims of the project may be summarised thus:

(1) To investigate the use of modified morphological operators as appropriate markers of position of structural features, as tools for feature extraction, in the context of the problems previously described;

(2) To characterise the operators, in terms of the requirements for initial processing (if any) to permit successful application, and in terms of accuracy with respect to established techniques;

(3) To devise a scheme for the use of the same, or similar, operators for the recognition of objects from stored library feature sets;

(4) To develop an algorithmic architecture capable of the recognition task, suitable for hardware implementation;

(5) To characterise the developed scheme with respect to established techniques.

Chapter 2: Theory 1 - The RJ Operator

Overview

In this chapter, we shall discuss the need for a modified form of set erosion, loosening the conditions on membership of the result, and then describe one approach to development of a possible solution. This forms one of the novel aspects of the research undertaken. We shall compare the resulting filters with those created under another modified form of set erosion, the soft morphological filters, clarifying the relationship between the two and the practical differences that result. The range of morphological chain operations (opening, closing) and direct applications (hit and miss transform, scale analysis) used in the later experimental work will be stated, and illustrated with simple examples.

2.1 Introduction

The morphological erosion operation is a direct measure of the occupancy of one set by another. Binary set erosion is defined by:

$$A \Theta B = \{x \in E^n | (x+b) \in A \forall b \in B\} \qquad [1]$$

or

$$A \Theta B = \bigcap_{b \in B} (A)_{-b} \qquad [2]$$

where a,b are members of A and B respectively, E^n refers both to Euclidean N-space or its discrete equivalent, the set of N-tuples of integers, and $A,B \in E^n$.

The set erosion operation result is the set of points at which the eroding set, often called the structuring element, is wholly contained. Such an operation offers a means of locating one set within another, and therefore may offer a means of recognising an object from extracted data. The problem lies in its precision - all elements of the eroding probe set must exist within the set under test.

S R Sternberg [6,7] introduced the general extension of mathematical morphology to greyscale images, through the use of localised min and max operations. This approach uses the idea of the top surface of the umbra of the function as the operable surface, yielding the following definitions. The concept of the rolling ball transform, the points described by a rolling ball moving over the surface of the function defining the dilated function, as the means of calculating dilation by a circular function is due to Sternberg.

Grey scale dilation of a function $f(x)$ by a function $g(z)$ is defined as:

$$(f \oplus g)(x) = TOP[U[f(x)] \oplus U[g(z)]]$$

$$(f \oplus g)(x) = \max_{(z)\in g(z)} \{ f(x-z) + g(z) \} \qquad [3]$$

Greyscale erosion is defined as:

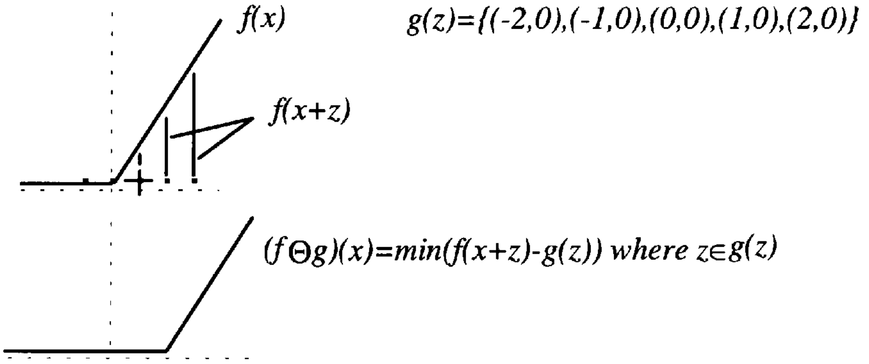
$$(f \Theta g)(x) = TOP[U[f(x)] \Theta U[g(z)]]$$

$$(f \Theta g)(x) = \min_{(z)\in g(z)} \{ f(x+z) - g(z) \} \qquad [4]$$

The full set of operations and conditions defined by Serra and Matheron are applicable equally to grey scale processing (see Haralick et al [4]).

Consider the set erosion example of figure 2 below. The function *g(z)* is defined as {position relative to its own origin, value at that position}.

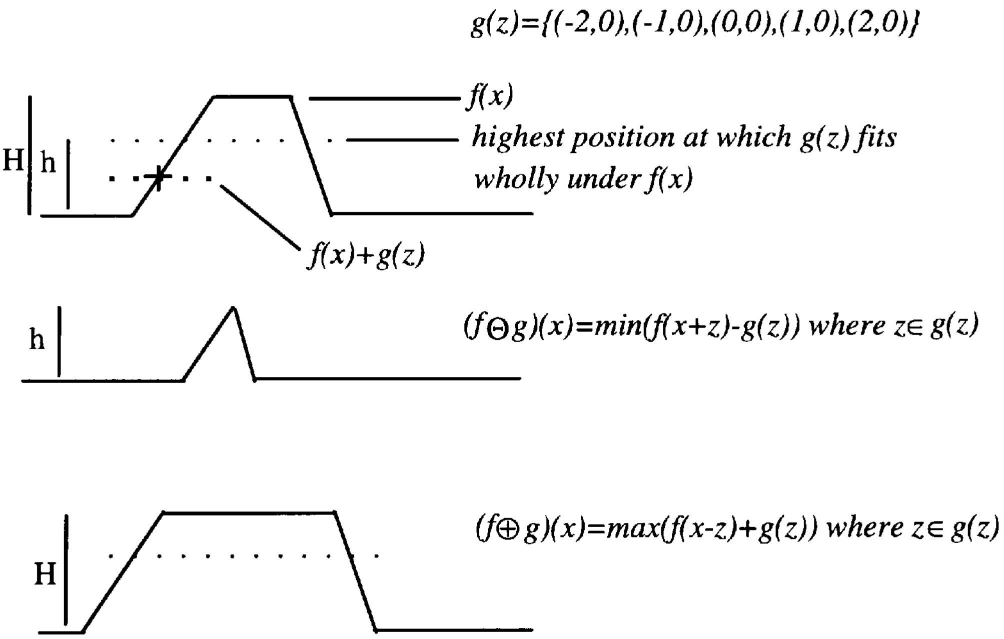
Figure 2: 1-D Example of Grey Scale Erosion Principle


f(x)
g(z)={(-2,0),(-1,0),(0,0),(1,0),(2,0)}
f(x+z)
(fΘg)(x)=min(f(x+z)-g(z)) where z∈g(z)

For this very simple example, the effect is one of a spatial shift in the ramp function *f(x)*. The value at each point in the result is calculated as the minimum of a localised set of results corresponding to the points in the structuring set *g(z)*. In effect, this is a zero order rank filter over the structuring element area.

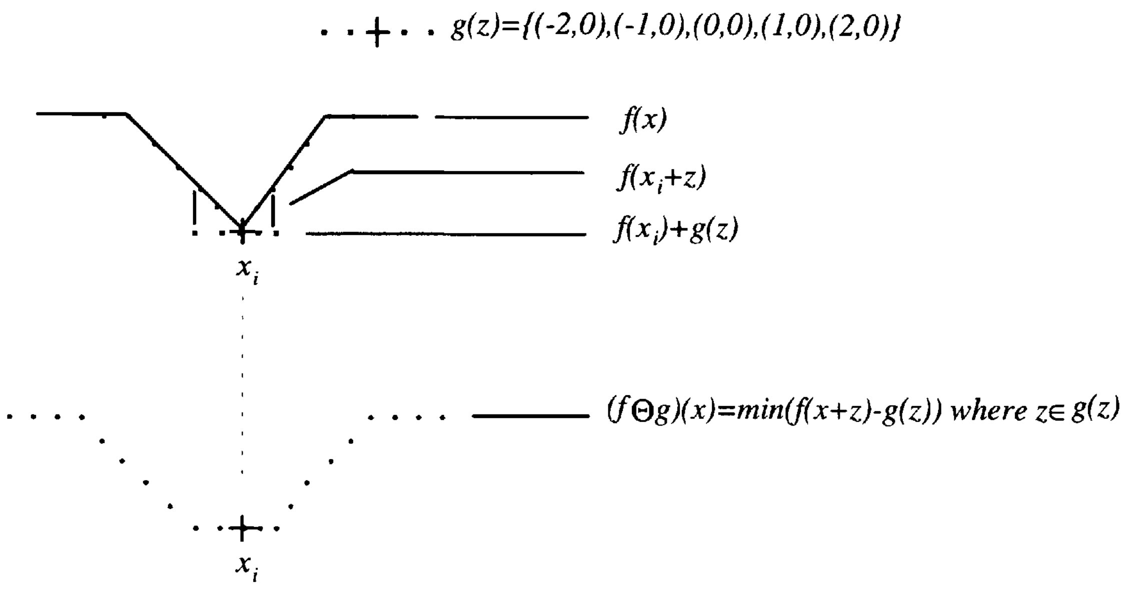
Now consider the more complex situation shown below in figure 3. The erosion is shown to reduce the size of the bright areas (assuming the vertical axis to be intensity). The areas which survive unchanged are those at which the probe set is fully contained in the image surface - in this case, only in the areas preceding the ramp edge of the function. As a corollary, the grey scale dilation increases the size of bright areas. After dilation, the profile of the intensity takes on the shape of the dilating structure. The practical effect is that of sliding the origin of the template over the surface of the image function, and taking the maximum value over the template region (area in 2-D functions).

Figure 3: The effect of Grey Scale Erosion and Dilation on Bright Areas in an Image


g(z)={(-2,0),(-1,0),(0,0),(1,0),(2,0)}
f(x)
highest position at which g(z) fits wholly under f(x)
H
h
f(x)+g(z)
h
(f⊖g)(x)=min(f(x+z)-g(z)) where z∈ g(z)
H
(f⊕g)(x)=max(f(x-z)+g(z)) where z∈ g(z)

Note that height h is the highest level at which *g(z)* is totally contained in *f(x)*. Now consider the function of figure 4 below, illustrating the effect of erosion on dark transitions in an image. Here the dark areas are grown by the application of erosion.

Figure 4: Effect of Erosion on Dark Area Boundaries



$g(z)=\{(-2,0),(-1,0),(0,0),(1,0),(2,0)\}$
$f(x)$
$f(x_i+z)$
$f(x_i)+g(z)$
x_i
$(f\Theta g)(x)=min(f(x+z)-g(z))$ where $z\in g(z)$
x_i

2.1.1 The Justification of a Less Perfect Set Erosion

The requirement for full containment of the probe set renders formal set erosion unwieldy for recognition purposes. It implies a need for a "perfect" set of extracted data if a stored reference set is to be used as the probe, or an extensive reference set if the extracted data is used. Given that feature extraction is not a trivial exercise, due, for example, to noise or overlapping, then the likelihood of spurious inclusion or rejection of a potential feature will result in an incorrect "miss" in the morphological comparison by set erosion.

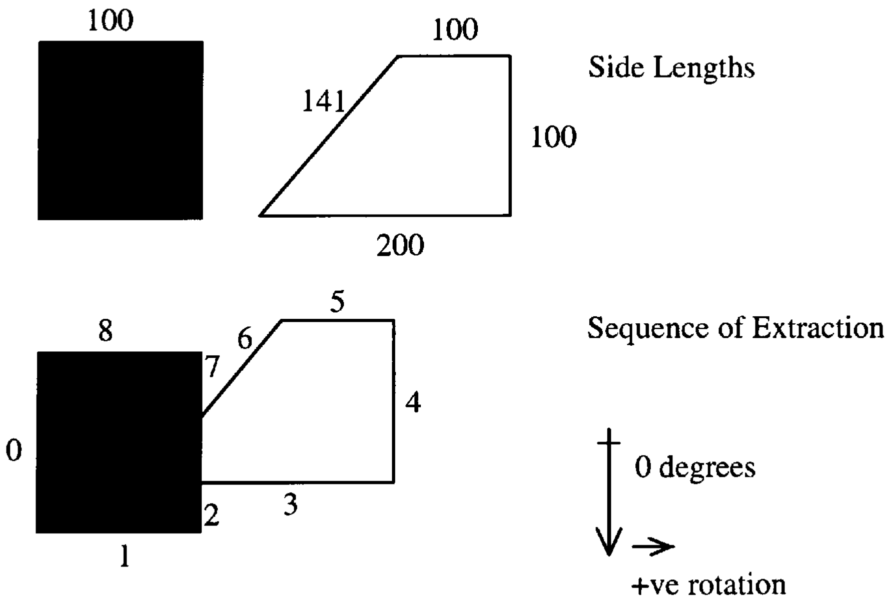
This problem has been observed in picture enhancement applications using morphological filtering. The filters, although achieving the required cleansing of the image by elimination of structures smaller than the probe (structuring element) set, also remove any part of the information in the image smaller than it. This is undesirable, in that elements useful for later processing operations may be eliminated. Consider the example below, using an erosion for a modified slope density function contour signature for a simple shape with reference models.

Example 2.1.1: Contour Following for Square and Trapezium Shapes, Separately and Overlapped

Consider the overlapped trapezium and square of figure (5) below. One simple means of modelling such shapes would be:

Shape Model = {length of side i, orientation of side i relative to some arbitrary axes, position number}

Figure (5): Square, Trapezium and Overlapped Image



For the example, sides aligned with the vertical grid of the image (increasing $y \downarrow$), surface normals directed in the negative horizontal grid direction ($\leftarrow$) are regarded as being at zero degrees. The sequence of extraction is in the anticlockwise direction. Thus the model databases for the two shapes would be:

Trapezium Model: Set $A = \{(200,90,0),(100,180,1),(100,270,2),(141,315,3)\}$

Square Model: Set $B = \{(100,0,0),(100,90,1),(100,180,2),(100,270,3)\}$

The overlapped image data yields:

Overlapped Shape: Set $C = \{(100,0,0),(100,90,1),(25,180,2),(150,90,3),(100,180,4),$
$(100,270,5),(71,315,6),(25,180,7),(100,270,8)\}$

As can be seen from the representational data sets in Table 1 for the two shapes and the combined picture, the straightforward application of set erosion to identifying the shapes will fail, because of the data diminution caused by the overlap.

Table 1: Set Erosion as a Marker of Position

$$C \Theta A = \{\varnothing\}$$
$$C \Theta B = \{\varnothing\}$$

The question, therefore, is one of the reduction of the requirement of complete occupancy without overly degrading the useful performance of the filters. The degree of acceptable degradation of performance is specific to the operations to be performed by the filters.

2.2 Development of the RJ Operator

The obvious approach is to loosen the rejection of the filter in some controlled manner. One means of achieving this objective would be to permit acceptance of a point as part of the eroded result in an unaltered state if some degree of partial containment of the probe function is achieved. We have defined an operator for this purpose.

The RJ operator applies a probe set to a data set, and generates two results. The *R* set indicates the degree of coverage of the probe set at a particular point in the image. The set of sets ***J*** contains a *J* set corresponding to each element of *R*, indicating the required additions to the set for full coverage of the probe set to be obtained. A *J* set is, therefore, generated for each point in the set under test. Whilst this seems unwieldy, and is not used as part of the conditional filter process which follows, it offers benefits in the analysis of the feature sets extracted from the image for recognition purposes.

Consider the diagrams of figure (6) overleaf. The *R* set is a count of how many elements of the probe structuring element *B* are contained at each position in set *A*. In this example, set *B* is fully contained within set *A* at various positions. If we now apply a threshold to the *R* set, requiring a value to be equal to the complete number of elements in set *B* for its position to survive in the result, we can generate an equivalent to set erosion based on the degree of occupancy observed. The operator is represented by the @, as $A@B$.

The diagram of figure (7) takes the concept one stage further. Here set *C* is not fully contained within set *A*, at any of the points belonging to set *A*. The application of formal set erosion to this situation would produce the empty set as its result. We could, however, produce a "best fit" equivalent by considering those points where containment is most complete. In the example two points (figure 7(d)) are nearest to full containment (i.e. the *R* set value is highest, as found by taking the maximum over the *R* set region), the result being designated the $\underline{R}$ set. The elements missing at each of the loci in $\underline{R}$ are contained in the J sets, $J_{0,0}$ and $J_{1,0}$, the subscripts defining the R element to which they are relevant.

Figure 6: Measurement of the Containment of Set B at the Various Points in Set *A*

(a) Set A

┌	●	●	●	
●	●	●	●	
	●	●		
	●	●		

(b) Set B

┌	●	●		

(c) R = r values for A@B

┌3	3	2	1	
3	3	2	1	
	2	1		
	2	1		

(d) Values of A@B corresponding to full containment

$\underline{R}$:

┌3	3			
3	3			

J:∅

(e) D=AΘB = A©B:{r=3}

D:

┌	●			
●	●			

As the definition of erosion used earlier implies, it is equivalent to the positions where set B is fully contained within set A

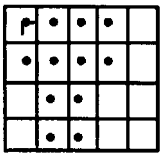
This work is more formally defined in section (2.2.2).

$$R = A@B = \{a \in A | r = \sum_{b \in B}(n),\ n = 1 \text{ where } (b)_a \in A\}, \qquad [5]$$

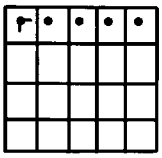
$$J=\{J_a\}\forall a \in \mathrm{A}, \text{ and } J_a = \{j | j = 1 \text{ where } (b)_a \notin A\} \qquad [6]$$

Figure (7): Measurement of the Containment of Set *C* at the Various Points in Set *A*

(a) Set A



(b) Set C



(c) R = r values for A@C
before max operation

4	3	2	1	
4	3	2	1	
	2	1		
	2	1		

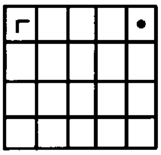
Note that C is not fully contained at any point in set A

(d) R = *max (A@C)*

R:

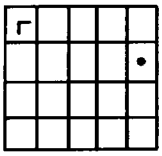
4				
4				

$J_{0,0}$



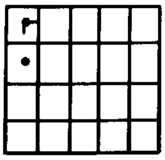
The J sets hold the elements needed to complete the containment of set C within A

$J_{1,0}$



(e) E=A©C:{r=4}

E:



It is a plausible use of this approach to define a locus/loci of maximum likelihood of containment of a feature set within a reference model, with the J sets allowing further analysis of each locus. This would allow the determination of whether the point is a viable candidate under any other constraints that may be applied to assist in identification of a good match. We could also apply any other constraint to R to obtain the equivalent of a set erosion by some portion of the probe structuring element, and thus generate different physical properties in the result.

The grey scale morphological operators affect the shape of the processed function. If the function were regarded as the top of a solid surface, they would affect the profile of its ridges and valleys. This is the basis of Sternberg's [7] lifting of binary to grey scale erosion, and therefore the basis for the R set generation and the conditional operators defined later in this chapter.

2.2.1 Grey Scale Discrete Functions

Consider the top surface form of greyscale erosion of a function f by a structuring function k (see Haralick et al [4], or section (2.1) above):

$$f\Theta k(x) = TOP[U[f]\Theta U[k]]$$

This may be interpreted as the minimum over the structuring function template:

$$f\Theta k(x) = \min_{z \in k(z)} \{f(x+z) - k(z)\}$$

Now, for any point $(x_i) \in f(x) | (x_i + z) \in domain\ off$,

if $(U[f(x_i + z)] \geq U[f(x_i) + k(z)]) \forall z \in k(z)$

then $(f\Theta k)(x_i) = f(x_i)$.

To clarify this, if $(f(x_i + z) - k(z)) \geq (f(x_i) - k(0)), \forall z \in k(z)$ then $f(x_i) - k(0)$ is the result of the erosion over the template area. Any value at the origin of an eroding probe structure other than zero results in a general darkening of the image.

In other words, any point in the original image belongs to the eroded result provided it meets this criterion, subject to the general intensity offset implied by a value at the origin of the structuring element. We can constrain the origin of the probe set to be of zero intensity value with no loss of generality in the intensity profile of the probe sets. In order to relax the criterion, we must allow the point value $f(x_i)$ to belong to the solution if components fail to meet the criterion for formal erosion, i.e. if $(U[f(x_i + z)] \geq U[f(x_i) + k(z)])$ for some proportion of $z \in k(z)$ rather than for its whole. If the test is to be general, rather than directional or for a specific fragment of the probe, then the metric for assessing coverage should simply be the count of how many components of the probe set are contained.

In effect our measure becomes:

$$(f @ k)(x) = R(x), J \qquad [7]$$

where

$$R(x) = \{r(x): N, x \in f(x), (x + z \in \mathit{domain\ of f})|$$
$$r = \sum_{z \in k(z)} m,\ m=1 \text{ where } (f(x+z) \geq (k(z) + f(x));$$
$$m = 0 \text{ otherwise}\}$$

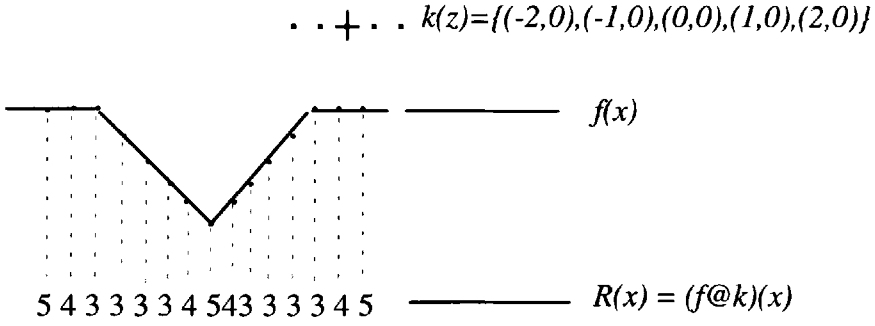
and $J(x) = \{J_r\} \forall r \in R(x)$

$$\begin{aligned} &= \{ j{:}N \mid j = k(z) \text{ where} f(x+z) = 0, \\ &\qquad j = k(z) - f(r+z) \text{ where} f(r+z) < k(z), \\ &\qquad \forall z \in k(z)\} \forall r \in R(x) \end{aligned}$$

A *J(x)* set may be generated for each locus in *f(x)* (and hence *R(x)*), and will contain a number of elements up to the number of elements in *k(z)*. It is primarily of use in recognition model evaluation from set occupancy.

R(x) is, therefore, a measure of the rank of the difference of the origin of the probe template and the current point in the image against the differences between the other points in the template and their corresponding points in the image. Figure 8 below illustrates the situation of a flat template *k(z)* passed over a dark area in the image function *f(x)*. The effect can be conceptually evaluated by placing the origin of *k(z)* at each point of *f(x)* and considering the degree of containment.

Figure 8: R Operator Output over a Boundary Area



The *R* value becomes a goodness of fit measure, closely resembling a fuzzy intersection but constrained in that only fully contained elements count towards the generation of an overall measure. This analogy becomes more obvious if we scale the value obtained by dividing by the number of elements in the *k(z)* set.

i.e. $R_{norm}(x)=R(x)\,/\,N, R_{norm}(x) \in (0,1)$, where N = no of elements in $k(z)$, or $Card\, k(z)$.

This number might be regarded as an elementary measure of fuzzy erosion, in that only wholly contained elements contribute to the value.

The R set result can now be used to generate a conditional erosion. Consider a partial RJ operation generating only the R set. This set is then used as a template to mask the original function. A point in f(x) is part of the result if R(x) meets some predetermined condition. We might indicate the inclusion of a point unaltered by use of a template *T(x)*, as a binary mask showing which points meet the chosen conditions.

Let us denote this result as $(f \copyright k)(x){:}T(x)$, where, for some predefined condition, *cond1*:

$$T(x) = \{t | t = 1 \text{ where } R(x) \in cond1, t = 0 \text{ where } R(x) \notin cond1, \forall x \in R(x)\} \quad [8]$$

Then $g(x) = (f \copyright k)(x){:}T(x)$

and

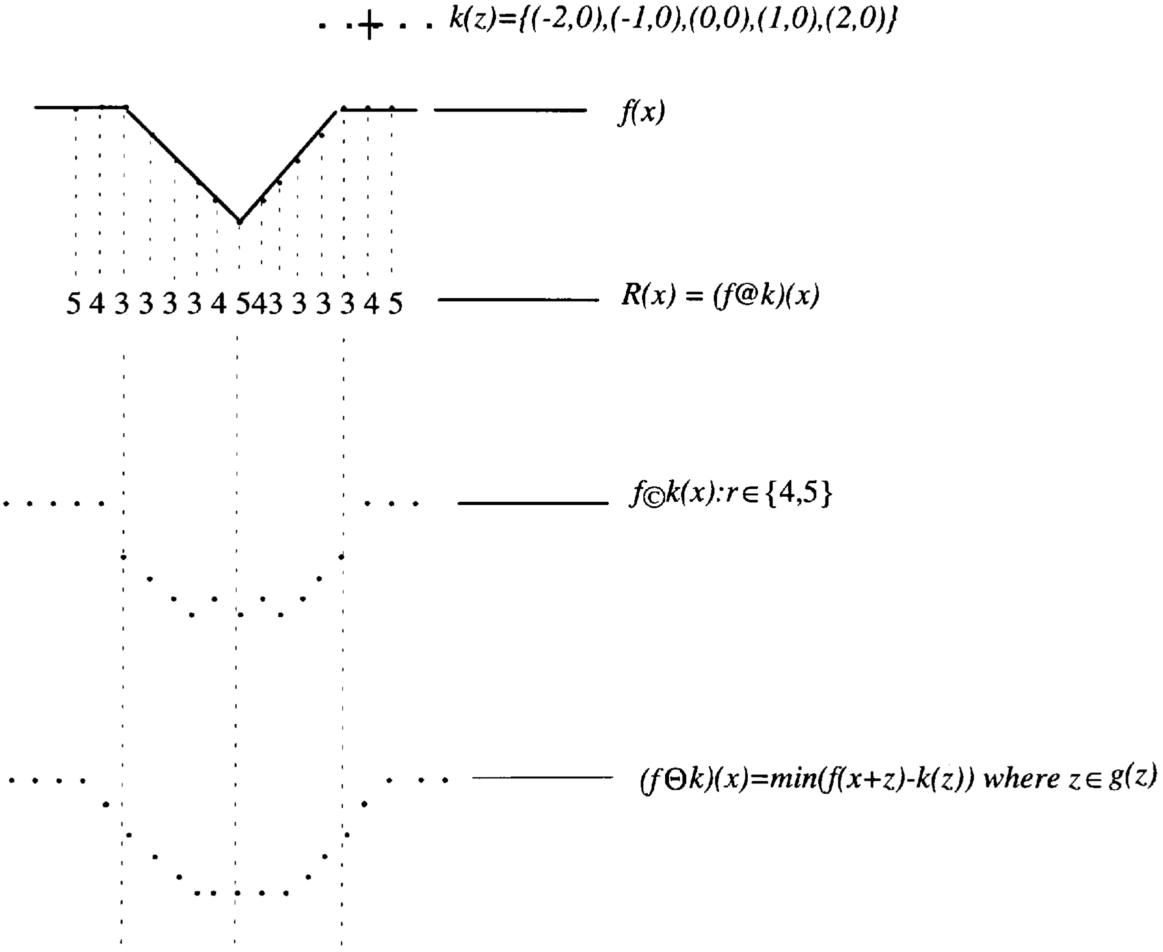
$$\begin{aligned}&(f \copyright k)(x){:}T(x) = \\ &\{g | g = \min f(x+z) - k(z) \forall z \in k(z), \\ &\quad \text{where } t = 0, \\ &\quad g = f(x) \text{ where } t = 1, \ \forall x \in f(x)\}\end{aligned} \qquad [9]$$

or, to state a more direct approach,

$$\begin{aligned}&(f \copyright k)(x){:}cond1 = \\ &\{g | g = \min f(x+z) - k(z) \forall z \in k(z), \\ &\quad \text{where } R(x) \notin cond1, \\ &\quad g = f(x) \text{ where } R(x) \in cond1, \ \forall x \in f(x)\}\end{aligned}$$

The example of figure 9 illustrates the conditional erosion of the same function used in figure 4, with the same probe structuring element. This is applied with the condition $r \in \{4,5\}$, as shown. The formal erosion is included for clarity. Note that several points survive unchanged in the conditionally eroded version which do not occur in the formal erosion result.

Figure 9: Conditional and Formal Erosions of a Grey Scale Function



k(z)={(-2,0),(-1,0),(0,0),(1,0),(2,0)}
f(x)
5 4 3 3 3 3 4 543 3 3 3 4 5
R(x) = (f@k)(x)
f©k(x):r∈{4,5}
(fΘk)(x)=min(f(x+z)-k(z)) where z∈g(z)

We may also use the R set to generate a conditional dilation in a similar form, with properties useful for enhancement and forming within clear limits a dual to the conditional erosion already specified.

It is consistent with the loosening of the conditions of erosion implied within the conditional erosion that the conditional dilation will effectively increase the conditions needing to be met to permit dilation to occur. We here require that a number of elements of the dilating structure greater than a set threshold is part of the umbra of the function under dilation, rather than a point simply being part of the top surface of the umbra under dilation, as the condition that dilation is permitted.

Let us denote this result as ®.

Then:

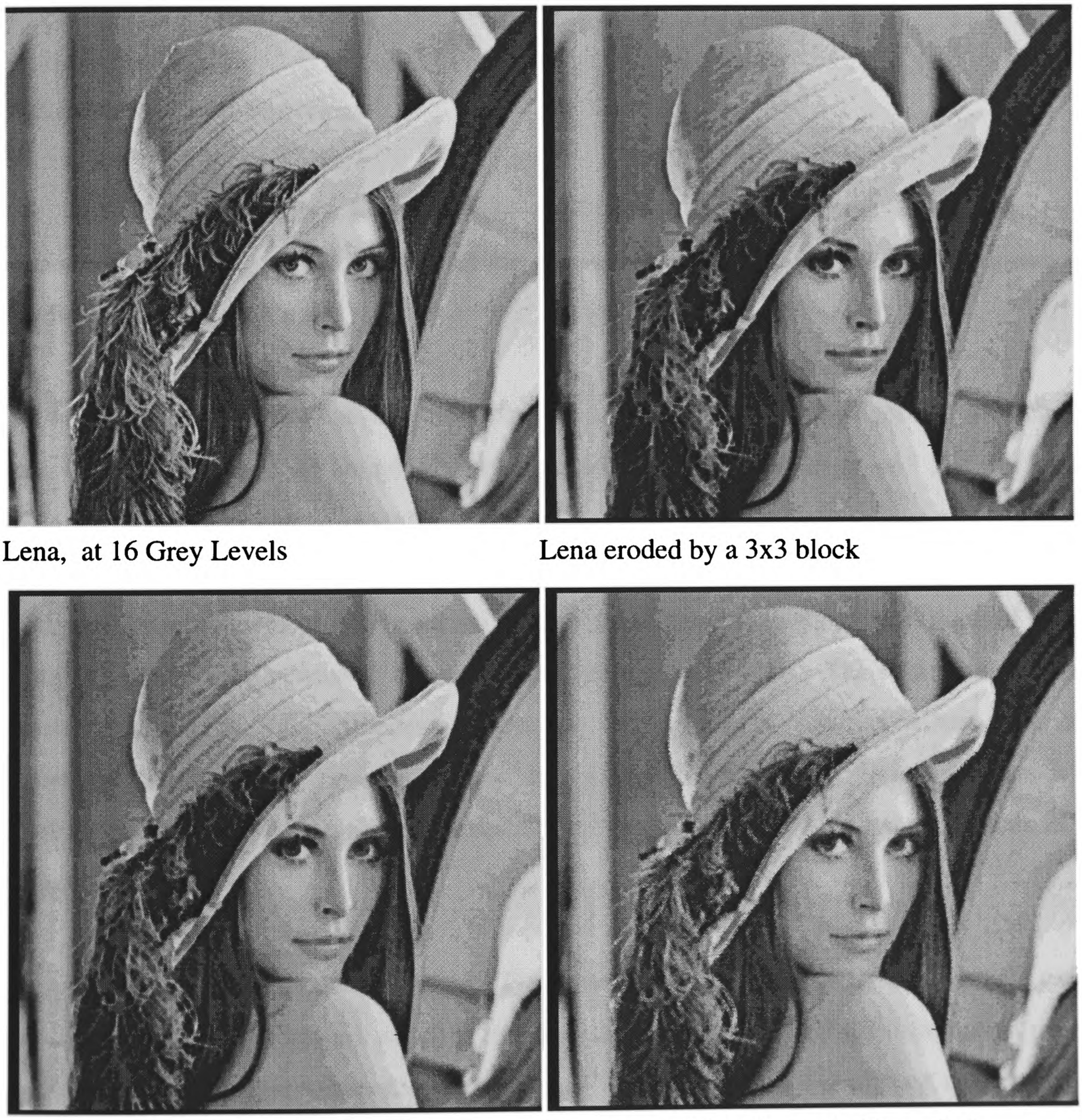
$$\begin{array}{l} (f \circledR k)(x) : T(x) = \\ \{ g | g = \max f(x-z) + k(z) \forall z \in k(z) \\ \quad \text{where } t = 0, \\ \quad g = f(x) \text{ where } t = 1, \forall x \in f(x) \} \end{array} \tag*{[10]}$$

or

$$\begin{array}{l} (f \circledR k)(x) : T(x) = \\ \{ g | g = \max f(x-z) + k(z) \forall z \in k(z) \\ \quad \text{where } R(x) \in cond1, \\ \quad g = f(x) \text{ where } R(x) \notin cond1, \forall x \in f(x) \} \end{array}$$

When the condition is established as a threshold set to zero (i.e. full containment is required), both conditional erosion and dilation forms collapse back to the formal set erosion and dilation definitions. The examples of figure 10 overleaf illustrate the perceptual differences between the formal erosion and dilation and the conditional equivalents generated using the above approach. The probe structure applied is a three by three block of zero height. The conditions applied are that *R(x,y)* exceeds the given threshold, i.e. that not less than the given threshold number of components are not contained at the point in question. Note the increase in residual detail as the condition is progressively relaxed from the formal erosion definition. The growth of the dark areas of the image is also reduced.

Figure 10: Examples of Conditional Erosion of Lena in Comparison with Formal Erosion, with a Flat 3x3 Block Template



Lena, at 16 Grey Levels

Lena eroded by a 3x3 block

Lena R conditionally eroded by a 3x3 block, subject to $\{r \in 2,0\}$

Lena R conditionally eroded by a 3x3 block, subject to $\{r \in 4,0\}$

A closer look at the iris of the eyes in the figures, and the levels of detail retained in the feather train from the hat, illustrate the increase in residual information after the use of the filters. The conditional filters allow significantly more detail to remain, and, in particular, produce less growth in the dark areas within the image. This is best exemplified by an examination of the iris and pupil areas of the images as the conditions on erosion are loosened (as defined by increased containment values).

2.2.2 Multivalued Functions

Grey scale functions have the problem of a real value background. Every point in the function will belong to the result, and the meaningful data is the degree of occupancy the probe achieves at each point.

Once feature data are extracted from an image as a set, the background of such data can be disregarded, or, as Dougherty discusses in [5], effectively set to $-\infty$. For this sort of data, the R operator can be calculated using a shift and intersect methodology, as is used for formal binary morphological operations. In the conventional terminology adopted in morphology, each element of a probe set is regarded as generating a shifted version of the test data, the result being generated by intersecting these partial results. The strict intersection of these partials will only yield a result where the formal erosion condition is met. We need therefore to define a *maximal intersection*, being the loci at which the largest number of the partials have a component. The R result is then the count of the number of partials in the maximal intersection.

There is a further consideration. The possibility of partial overlap of an object exists, yielding a description data set which is incomplete. Parts of the required information may be wholly missing, or their values may be changed. To allow for this eventuality, and extract the maximum information from the data obtained, we will need to include not only the completely covered components but also those which are partly contained as contributing to the R evaluation.

The reformation below includes partially covered elements of the test set as contributing to the R value. The R set now yields both the degree of containment of the probe function at each location within the original data, and the corresponding J sets would, for each point, indicate the additions needed to make containment complete. The loci of best fit are, of course, the points of obvious interest. By obtaining the maximum values of R, we are extracting the points of *closest* fit. The J_r set(s) corresponding to the maximum R values indicate the missing data required to complete the containment of the probe at each location.

Assume the multivalued domain has a background value of zero (this is simply done for clarity in the equations below - and reflects a commonly used condition). Then:

$$(F @ K)(x) = \{x \in F(x) | r(x) = \sum_{z \in K(z)} (n), \quad n = 1 \text{ where } F(x+z) \neq 0, \quad n = 0 \text{ elsewhere}\} \qquad [11]$$

$J = \{J_r\} \forall r \in R$, and

$$J_r = \{x \in F(x), j | j = K(z) \text{ where } F(x+z) = 0, \quad j = k(z) - F(x+z) \text{ where } F(x+z) < K(z)\}$$

We can also calculate the R result by a shift and intersect method:

$$R(x) = \{r | r = \text{no of partials in the intersection of } (F(x))_{-z} \forall z \in k(z), \forall x \in f(x) | (x+z) \in D_f\}$$

$$R(x) = \{x_i \in F(x); z_h \in K(z); r(x) = \left[\begin{array}{c} \sum_{z \in K(z)} (n)\text{ , } n = 1 \text{ where } \left[F(x_i)\right]_{-z_h} \cap \left[F(x)\right]_{-z} \neq \varnothing, \\ n = 0 \text{ otherwise} \end{array}\right]\}$$

and, as before,

$$J(x) = \{J_r\} \forall r \in R(x),$$

$$\begin{aligned} J_r = \{x \in F(x), j | j &= K(z) \text{ where } F(x+z) = 0, \\ j &= k(z) - F(x+z) \text{ where } F(x+z) < K(z)\} \end{aligned}$$

To extract the maxima of the function we simply require $\underline{R}(x) = max\, R(x)$, or, to restate the above in a similar manner,

$$\begin{aligned} \underline{R}(x) = \{r | r = &\text{ no of partials in maximal intersection} \\ &\text{of } \ (F(x))_{-z} \forall z \in k(z), \forall x \in f(x) | (x+z) \in D_f\} \end{aligned} \qquad [12]$$

$$\underline{R}(x) = \{x_i \in F(x); z_h \in K(z); r(x) = \max\left[\begin{array}{c} \sum_{z \in K(z)} (n)\text{ , } n = 1 \text{ where } \left[F(x_i)\right]_{-zh} \cap \left[F(x)\right]_{-z} \neq \varnothing, \\ n = 0 \text{ otherwise} \end{array}\right]\}$$

and, calculated as before,

$$J(x) = \{J_r\} \forall r \in \underline{R}(x)$$

$$\begin{aligned} J_r = \{x \in F(x), j | j &= K(z) \text{ where } F(x+z) = 0, \\ j &= k(z) - F(x+z) \text{ where } F(x+z) < K(z)\} \end{aligned}$$

Given that this formulation minimises the unnecessary calculation of *J(x)* sets whilst yielding the relevant information, it is the one used in the practical work later in this thesis.

The binary case is simply a two valued case of this multivalued situation. We can state it as an equivalent form:

$$R = A @ B = \{a \in A | r = \sum_{b \in B} (n),\ n = 1 \text{ where } (b)_{a} \in A\},$$

and

$$J = \{J_a\} \forall a \in \mathrm{A},$$

where

$$J_a = \{j | j = 1 \text{ where } (b)_a \notin A\}$$

Application of this approach to the erosion based shape recognition problem of example 2.1.1 now yields a different solution:

Example 2.2.2: A Simple Monoscale Example for Multivalued Set Analysis

Once again the shape databases for the two shapes would be (see figure 5, page 26) {(side length, orientation, position in sequence of extraction),(...),....,(...))}

Trapezium Model	{(200,90,0),(100,180,1),(100,270,2),(141,315,3)}	= Set A
Square Model	{(100,0,0),(100,90,1),(100,180,2),(100,270,3)}	= Set B

The extracted image data yields:

Overlapped Shape	{(100,0,0),(100,90,1),(25,180,2),(150,90,3),(100,180,4), (100,270,5),(71,315,6),(25,180,7),(100,270,8)}	= Set C

Table (2) below shows the results of the R-J operator on the image data for both shapes. The $\underline{R}$ set elements are arranged as {θ_i, position number, r}. Here θ_i represents the orientation of the component of maximal fit, the position number confirms its position in the sequence of sides of the extracted overlapped shape data, and r is the number of components in the maximal fit. The J_r sets for each $\underline{R}$ element are arranged as {(missing component needed for full containment of element, θ_i, position number)}. The partial shift and intersection is shown explicitly in Appendix A.1 for further clarity.

Although a direct inspection of the overlapped figure shows that the square shape would fit completely within the trapezium, the edge length/orientation model does not offer the position corresponding to this as a most significant locus. This is simply because of the absence of the fourth side information at that position.

Table (2):

$C@A$: $\underline{R} = \{(0,3,4)\}$

: $J_{0,3} = \{(50,90,3),(71,315,6)\}$

$C@B$: $\underline{R} = \{(0,0,3),(90,3,3),(-90,-1,3),(0,2,3)\}$

: $J = \{J_{0,0}, J_{90,3}, J_{-90,-1}, J_{0,2}\}$

$J=\{\{(75,180,2),(100,270,3)\},\{(100,360,6)\},\{(100,-90,-1),(75,180,2)\},$
$\{(100,0,2)\}\}$

The trapezium has only a single position of maximum likelihood, and a correspondingly simple J set. Therefore locate the trapezium data at the indicated position (ie starting at the 3rd element in C) and subtract the data from the scene image. Negative lengths have no meaning in this operation, simply implying that a side has been fully covered. They are therefore dropped from the resulting set {C-A} and the elements of {C-A} renumbered. If, following subtraction, two adjacent sides in the set are at the same angle, it is consistent with a logical physical interpretation to assume that they may be the continuation of the same side and may therefore be consolidated into a single value.

$(C\text{-}A) = \{(100,0,0),(100,90,1),(50,180,2),(100,270,4)\}$

Re-applying the operator yields:

$(C\text{-}A)@B :\underline{R} = \{(0,0,4)\}$
$:J_{0,0} = \{(50,180,2)\}$

Now consider the J sets from *C@A* and *(C-A)@B*. The shape in the accumulated J sets from the accepted orientation estimations indicates the area of overlap. The fact that this forms a closed polygon (sum of components in orthogonal directions is zero) increases confidence in the solution.

Where all elements of a scene are included in the model database this technique might be used recursively to identify each shape or object.

The hypothesis and subtraction technique using the J sets thus offers a means of increasing confidence in the apparent results. Such simple shape, monoscale images are unrepresentative of the general class of recognition problems. The technique is applicable to more complex scenes.

2.3 Soft Erosion Compared with the RJ Conditional Approach

An alternative approach to loosening the constraints of formal erosion is that of the soft morphological filters. The soft erosion of a function *f(x)* by a pair of structuring element sets *A* and *B* is defined by Koskinen, Astola and Neuvo [22] as:

$f\Theta[B,A,k](x) = k^{th}$ smallest value of the multiset $\{k\Diamond f(a): a \in A_x\} \cup \{f(b): b \in (B-A)_x\}$

$A_x = \{x + a: a \in A\}$ ie A translated by x

$k\Diamond x = x, x, x, \ldots, x$ k times

The output of the filter at a point is the minimum of the set $\{f(a): a \in A_x\}$ if that value is smaller than the k^{th} smallest value of the set $\{f(b): b \in (B-A)_x\}$; otherwise it is the k^{th} smallest value of $\{f(b): b \in (B-A)_x\}$.

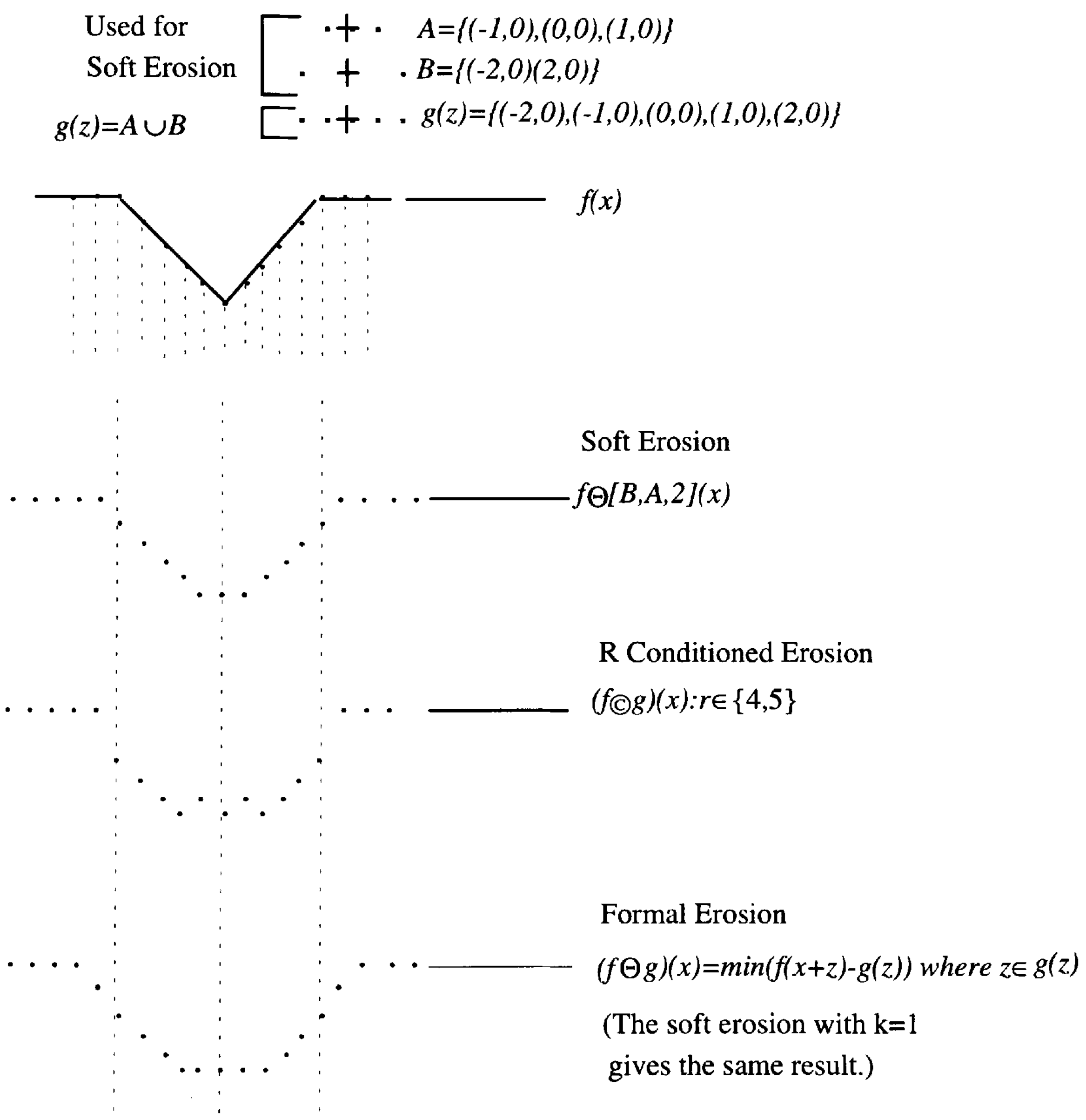
The set A forms the hard core of the erosion, set B forming a soft umbra about it. In the umbra area the min and max operators are replaced with other rank order statistics, defined by the parameter k. This work includes the important idea of using some rank other than the max or min as the determinant of the result with morphological processing. A simple comparison example between the soft erosion of a function, the rank conditioned erosion and the formal erosion of it by an equivalent probe set is shown in figure 11 overleaf.

Now let $A = \varnothing$ (empty set)

then $f\Theta[B,\varnothing,k](x) = k^{th}$ smallest value of $\{f(b): b \in B_x\}$

In other words, the "hard" core A is no longer relevant, and we are using the "soft" umbra B alone to define the solution. This is a straightforward rank filter implementation over the chosen area. The shape of the area is the morphological contribution to the outcome.

Figure 11: Soft Erosion, Conditional Erosion and Formal Erosion of a Function



As an example, if

$$B = \begin{matrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{matrix},$$

then the median filter (3x3) = 5^{th} smallest value of $\{f(b){:}b \in B_x\}, \forall x \in f(x)$, equivalent to $f\Theta[B,\varnothing,5](x)$.

Now let $B = \varnothing$, then $f\Theta[\varnothing, A, k](x) = \min\{f(a){:}a \in A_x\}$, which is the practical statement of set erosion of a function *f(x)* by a flat set *A*.

If *A* is flat, for all elements *a* of *A*, *A(a)*=0, and

$$f\Theta A(x) = \min\{f(x+a) - A(a)\} = \min\{f(x+a){:}a \in A\}$$

In order to make an adequate comparison between the soft morphological erosion and the rank conditioned erosion we propose, it is necessary to lift the soft operator to the grey scale function by function form. To do this, we will follow the basic ideas of Sternberg's original lifting of set by set erosion to function by function using the umbra and top surface operators. The soft erosion of a function *f(x)* by a core function *g(y)* with a soft umbra function *h(z)* becomes:

$f\Theta\big[h,g,k\big](x) = TOP(U[f]\Theta[U[h],U[g],k) =$
k^{th} smallest value of the multiset $\{k\Diamond(TOP(U[f(y)]) - TOP(U[g(y-x)])){:}\, y \in (g(y))_x\} \cup$
$\{TOP(U[f(z)]) - TOP(U[h(z-x)]){:}z \in (h(z))_x, z \notin (g(y))_x\}$

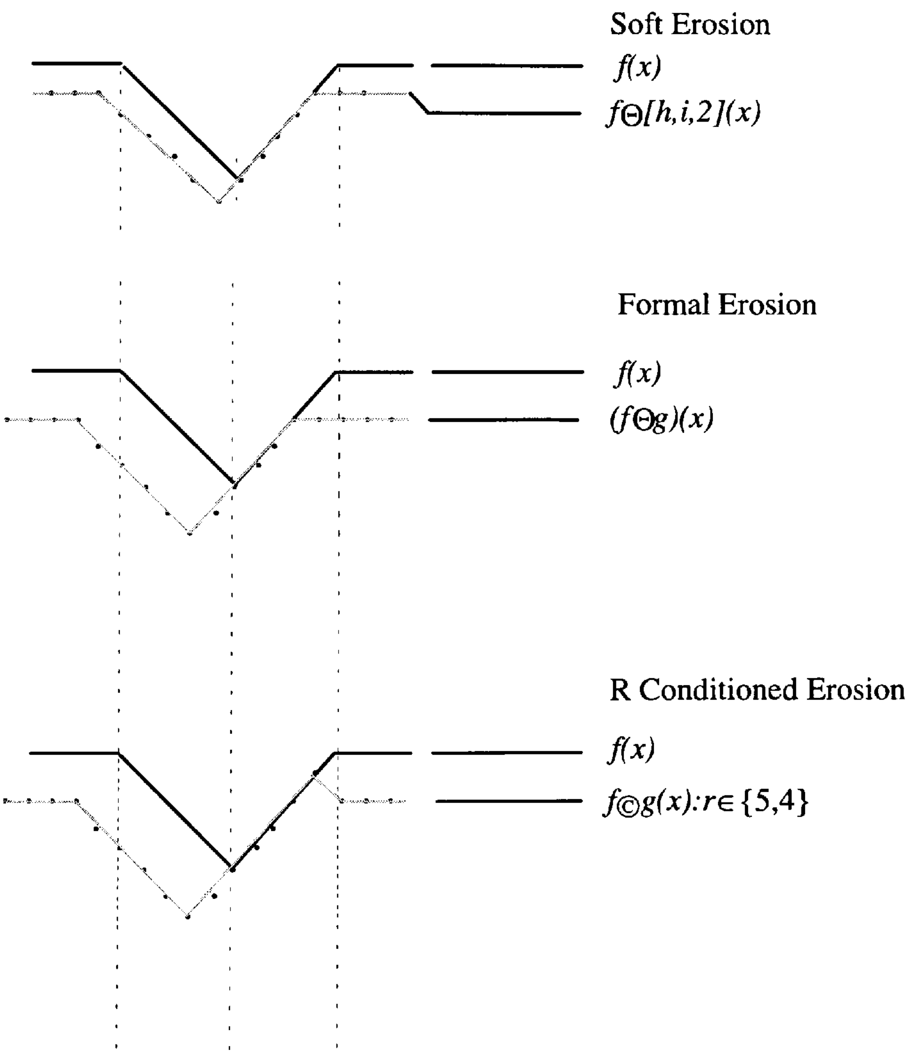
But $TOP(U[f]) = f$, giving rise to the pragmatic greyscale discrete equivalent

$f\Theta\big[h,g,k\big](x) = k^{th}$ smallest value of the multiset
$\{k\Diamond(f(y) - g(y-x)){:}\, y \in (g(y))_x\} \cup \{f(z) - h(z-x){:}z \in (h(z))_x, z \notin (g(y))_x\}$

An example of soft erosion, formal erosion and R conditioned erosion is shown in figure 12 below. The calculation of the point values shown is given in Appendix A.2.

Figure 12: Example of Soft Erosion, Formal Erosion and Conditional Erosion for a Non-Flat Structuring Function

$i(y)=\{(-1,-1),(0,0),(1,1)\}$, $h(z)=\{(-2,-2)(2,2)\}$,
$g(z)=\{(-2,-2),(-1,-1),(0,0),(1,1),(2,2)\}$



The R conditioned erosion does not use a hard core to the function, so let us replace *g(y)* with the null function $\varnothing$. Then

$$f\Theta[h,\varnothing,k](x) = k^{th} \text{ smallest value of } \{f(z) - h(z-x) : z \in (h(z))_x\}$$

The difference now is in the use of the rank as information. The R erosion as it has been defined allows a point to belong to the result unchanged if it meets the rank conditions applied, the min operator being applied over the difference of functions in the region of the probe function if the conditions are not met. The equivalent operation could then be defined by varying the value of *k*, in effect defining it as a function with different values according to position in the image function. This could be stated as:

$$f \copyright h(x){:}cond1 = f\Theta[h,\varnothing,k](x),\ k = r(x) \text{ where } r(x) \in cond1, k = 0 \text{ otherwise}.$$

2.4 RJ Operator Applied to Standard Morphological Operations

The most standard and widely used of morphological operations for image processing are opening and closing. Opening consists of an erosion followed by a dilation with the same set. Closing is defined as a dilation followed by an erosion with the same set. In this work, the RJ operator conditioned erosion and dilation will be substituted for the formal operators in both opening and closing. The practical implementation of such an approach requires some thought.

Consider firstly the conditional closing. If the dilation is conditional, and the erosion unconditional, the result will be closed to the probe function, as would the result of any formal erosion having a non-empty set value. By setting the conditions on the dilation (usually as a straight forward greater than the particular threshold), the closing can be made to produce less of an increase in the size of the probed set. In effect, it is a less complete closing of the surface, but still closed to the particular probe set. If the erosion is also made conditional, the result is neither idempotent nor necessarily closed to the probe set. It is, however, increasing with repeated iteration of the operation.

With opening, if the dilation is made conditional and the erosion unconditional, the result is decreasing with iteration of the operation, but not idempotent. The application produces fewer components in the result than the formal opening. With conditional erosion, the result is neither necessarily decreasing nor idempotent.

In effect we are binarising the R set, choosing the threshold value at which the R locus is set to a 1 or 0 condition, and logically ANDing the result with the function under test. The probe set for the operation would be chosen for specific spatial attributes.

The conditional erosion and dilation have been described previously and denoted by © and ® respectively.

2.4.1 Perceptual Effect of Greyscale Operators

Greyscale opening and closing operators modify the intensity profile of features within the image under test. The opening operation is used to remove small light details while leaving the overall grey levels undisturbed. The closing is used to remove small dark details while leaving overall grey levels relatively unaltered. Large structures are not significantly modified.

The reason for applying the conditional erosion and/or dilation in place of the formal equivalent is to improve the perceptual quality of the result. By a judicious choice of structuring function, specific structural details can be removed with less effect on the perimeter of larger features than can be achieved with formal opening or closing.

2.4.2 Greyscale Opening

The greyscale opening, as in the binary case, can be defined according to the four combinations of operation: formal erosion, formal dilation; conditional erosion, formal dilation; formal erosion, conditional dilation; conditional erosion, conditional dilation.

Using the shorthand form for the definitions:

$$f \circ k = (f \Theta k) \oplus k$$

Substituting the conditional erosion for the formal erosion:

$$f \circ k = (f \copyright k{:}cond1) \oplus k$$

Note that the condition applies to the conditional part of the operation. An alternative operation would substitute the conditional dilation for the formal dilation. In this case, the format of the conditional opening would be:

$$f \circ k = (f \Theta k) \circledR k{:}cond1$$

A third option, combining conditional erosion and conditional dilation, exists. The conditional opening then yields:

$$f \circ k = (f \copyright k{:}cond1) \circledR k{:}cond1$$

2.4.3 Greyscale Closing

The greyscale closing of a function *f* by a function *k* is defined as:

$$f \bullet k = (f \oplus k)\Theta k$$

The conditional variants are then:

(i) Formal dilation / conditional erosion

$$f \bullet k = (f \oplus k) \copyright k \qquad \text{subject to condition cond1}$$

or

$$f \bullet k{:}cond1 = (f \oplus k) \copyright k{:}cond1$$

(ii) Conditional dilation / full erosion

$$f \bullet k = (f \circledR k{:}cond1)\Theta k$$

(iii) Conditional dilation / conditional erosion

The full conditional approach will then give:

$$f \bullet k{:}cond1 = (f \circledR k{:}cond1) \copyright k{:}cond1 \qquad [13]$$

Consider now the closing of an image subject to noise. With 10% pepper noise only, using formal closing and a three pixel square block template is effective in the elimination of the contamination, but at the cost of blockiness in the result. The use of a cross template reduces this effect. In the presence of salt and pepper noise, the formal closing produces far less desirable results: the growth of the bright areas is a severe distortion, which is not compensated by the subsequent erosion where the dilations of the salt noise pixels overlap each other. This is shown in figure 13(g) below.

The conditional openings using symmetrical and asymmetrical conditions and the same block template are shown in figures 13(a) - 13(h) for the same three by three flat block template. The lower residual noise results from the higher requirement for dilation to occur, reducing the likelihood of overlap of bright pixel dilated areas.

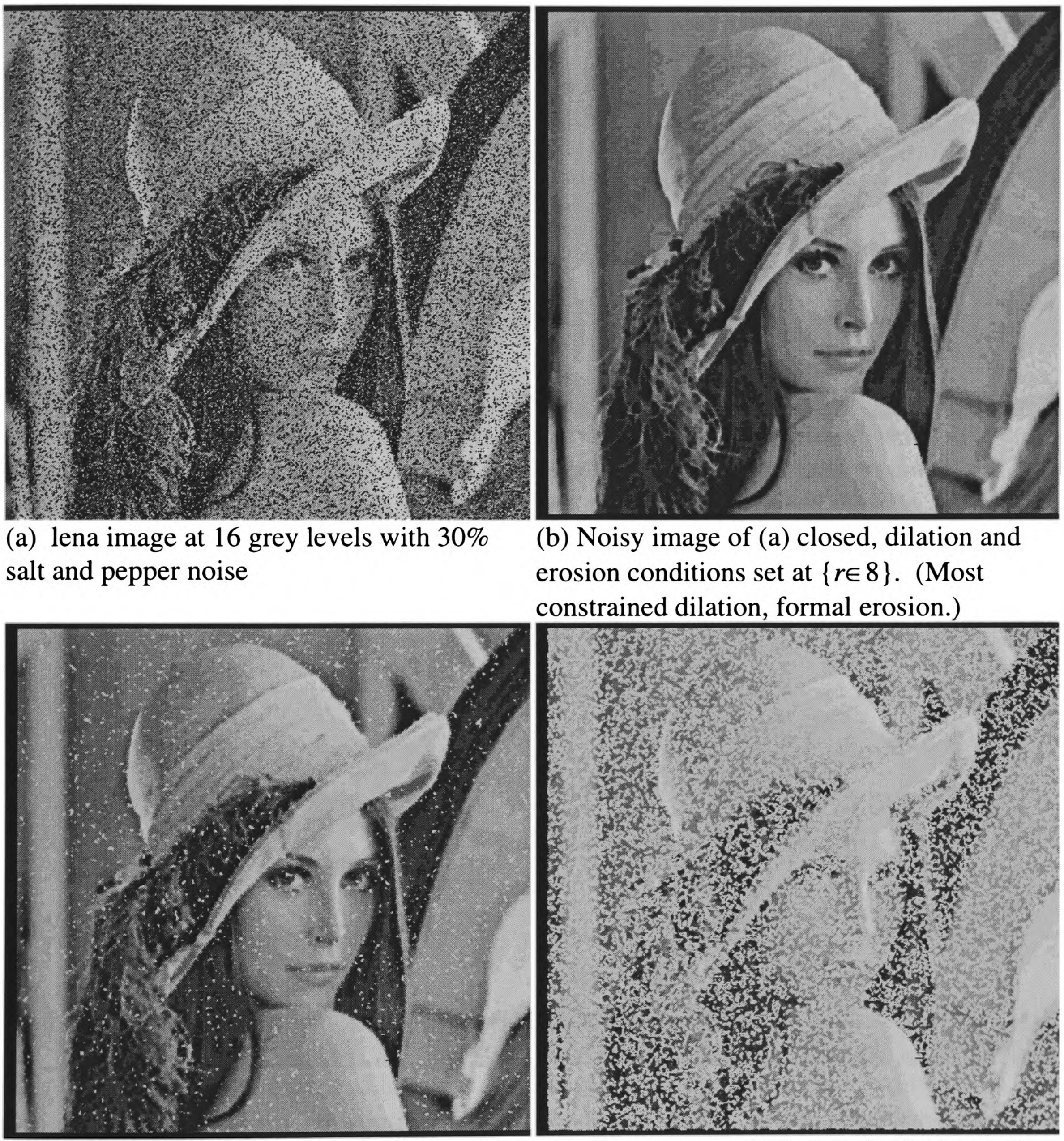
This set of images illustrates the use of symmetrical and non-symmetrical conditions on the conditional dilations and erosions forming the conditional closing. The effects of increasing the conditions for permitting dilation to occur are clear. The lower residual noise in the images where this has occurred results from the reduced likelihood of overlap of the bright pixel dilated areas.

The significant differences brought about by the use of the conditional erosion are best exemplified by figures 13 (b) and figure 13(g). Figure 13(g) is the formal closing of the noise contaminated image. The dilation products of the first stage of the closing overlap to form structures which are then of sufficient size to remain as noise objects (although, inevitably, pruned) following the subsequent erosion.

The case of figure 13(b) represents the limiting case for least information addition through conditional closing. The only points certain to meet the dilation condition of full containment of the block structuring element are the pepper noise impulses. The salt noise impulses, the area of which is not increased because of the dilation condition, are highly likely to miss the formal erosion condition of full containment. They are, therefore, removed from the conditional closing result.

The other figures show a range of intermediate conditions between these limits, illustrating the increased retention of noise as the dilation condition is brought nearer to its formal equivalent containment requirement.

Figure 13: Conditional and Formal Closing of "Lena" Contaminated with 30% Salt and Pepper Noise

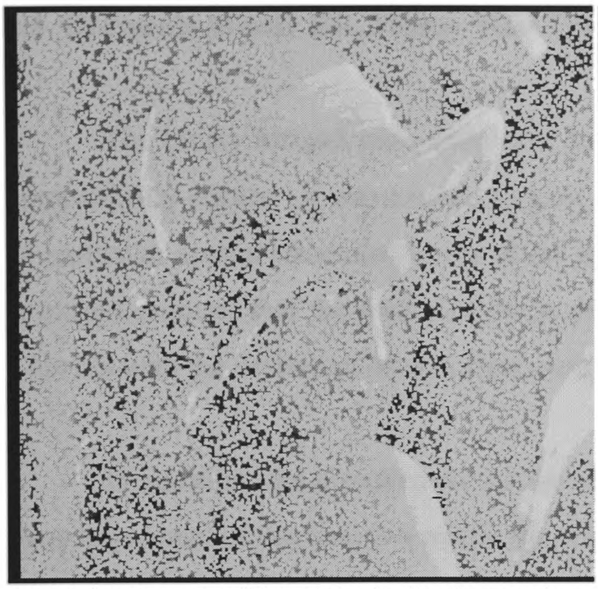


(a) lena image at 16 grey levels with 30% salt and pepper noise

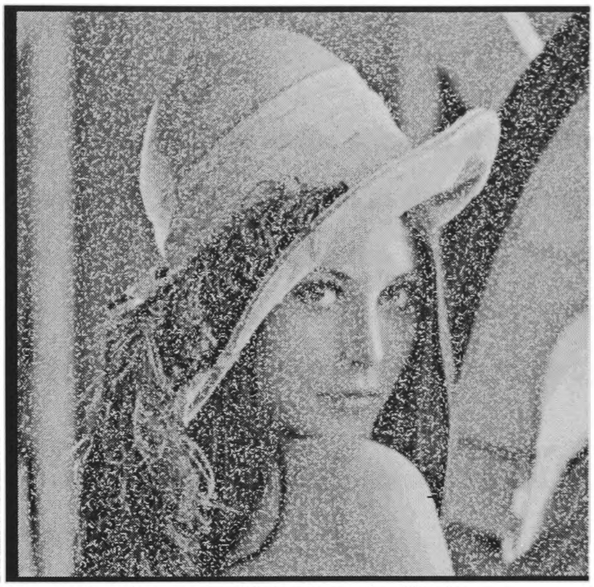
(b) Noisy image of (a) closed, dilation and erosion conditions set at $\{r \in 8\}$. (Most constrained dilation, formal erosion.)

(c) Noisy image of (a) closed, dilation and erosion conditions both set at $\{r \geq 6\}$

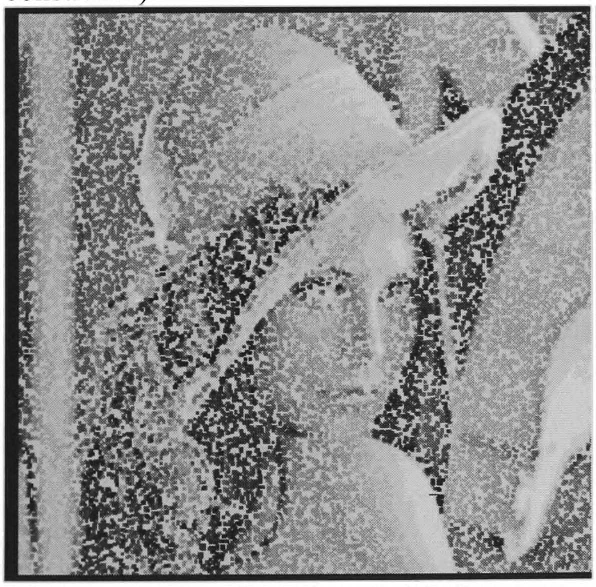
(d) Noisy image of (a) closed, dilation and erosion conditions set $\{r \geq 4\}$



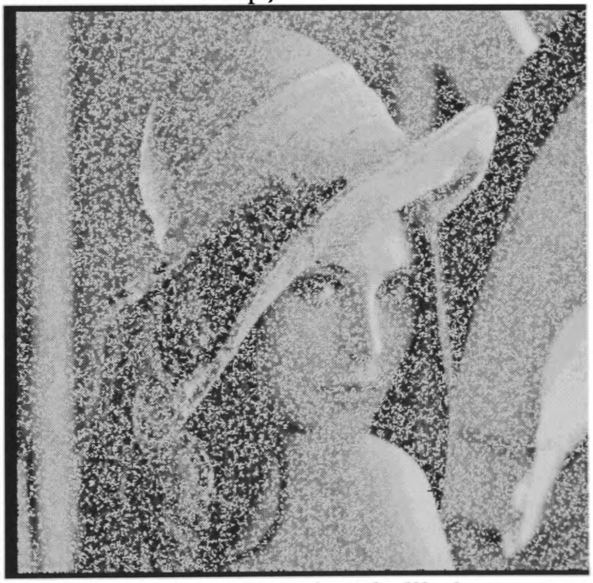
(e) Noisy image of (a) closed, dilation and erosion conditions set at $\{r \geq 0\}$. (Formal dilation, with the loosest possible erosion constraint.)



(f) Noisy image of (a) closed, dilation condition $\{r = 8\}$, and erosion condition {r≥0} (Most constrained dilation, loosest erosion membership}



(g) Noisy image of (a) closed, dilation condition $\{r \geq 0\}$, and erosion condition $\{r=8\}$ (Formal closing)



(h) Noisy image of (a) closed, dilation condition $\{r \geq 6\}$, erosion condition $\{r \geq 2\}$

2.5 Greyscale Mostly Hit, Mostly Miss Transform

The binary hit and miss transform accepts points as part of the solution where shape compatibility occurs. It was created to give a marker of position, useful in the identification of particular spatial structures in the given image or data set. Moving this idea to greyscale functions requires that the profile of the probe functions be used, rather than the basic shape. This fits with the definition of greyscale erosion based on the idea of umbras of functions.

In addition, the complement of the set must be considered. This is considered to be the complement of the function within the domain of the function.

i.e.

$$f^c(x,y) = (I_{\max} - f(x,y))$$

where f^c is the complement of the function *f*, and $I_{\max}$ is the limit of the domain of the function (i.e. its maximum value).

The binary hit and miss transform is defined as:

$$HMT = (A \Theta B) \cap (A^c \Theta D)$$

where *A* is the set under test, A^c is its complement, and *B* and *D* are the structuring elements applied to them respectively.

The question now arises as to the use of a greyscale version of such an algorithm -and therefore the final form assumed by the greyscale variant equations. In the purest sense, the greyscale version collapses to a binary solution, and the overall effect is simply to test for function gradients within the set under test. In effect:

$$\text{Greyscale } HMT = (f(x,y)\Theta g(\varepsilon,\eta)) \cap (f^c(x,y)\Theta h(\mu,v))$$

and the result is a set of points meeting the requirement set by $g(x,y)$ and $h(x,y)$. The problem here is that the formal erosion **does not produce a binary output**. The intersection is then ill-defined: it might be a fuzzy intersection; it might be subject to some condition for degree of intersection to determine the acceptability of a candidate point in (x,y).

Using the conditional erosion does not remove the problem: the result remains only definable in fuzzy terms. We can, however, use the R operator with conditions to produce a directly interpretable result.

The RJ operator as we have defined it previously implicitly provides a solution to this difficulty, in that the conditional acceptance of a point as part of the solution can be determined through the setting of a threshold. The $T(x,y)$ sets provide a direct means of obtaining a non-fuzzy intersection.

The greyscale MHMMT can therefore be established as:

$$\text{MHMMT} = T_1(x,y) \cap T_2(x,y) \qquad [14]$$

where:

$$T_1(x,y) = \{t \mid t = 1 \text{ where } R_1(x,y) \in cond1, \\ t = 0 \text{ elsewhere}\}$$

$$T_2(x,y) = \{t \mid t = 1 \text{ where } R_2(x,y) \in cond2, \\ t = 0 \text{ elsewhere}\}$$

$$R_1(x,y) = f(x,y) @ g(\varepsilon,\eta)$$

$$R_2(x,y) = f^c(x,y) @ h(\mu,\nu)$$

In the trivial case, if the conditions *cond1*, *cond2* are set to be equal to the total number of components in *g(x,y)* and *h(x,y)* respectively, the result is a form of greyscale hit and miss transform.

This mostly hit, mostly miss greyscale transform has obvious applications in direct identification of features within a greyscale image. The result is a binarised found / not found set. Where a symmetrical feature probe set can be established, a quasi non-directional feature analysis can be undertaken, removing one of the major problems of using morphological filters in general case recognition applications (ie freedom in rotation, translation, and, where the feature probe is relatively small, scale).

Chosen structural feature types within an image may be enhanced in the presence of noise and texture using an extension of the MHMMT. By using increasing sizes of a template, features below a predetermined size may be eliminated. Combinations of templates with specific spatial components enable the removal of noise and texture effects within the bounds set by the template. We can therefore define a spatial structure identification scheme based on the results of multiple R analyses as is shown below:

$$\text{Locus} = \bigcap_n T_n(x,y) \qquad [15]$$

where:

$$T_n(x,y) = \{t \mid t = 1 \text{ where } R_n(x,y) \in cond\, n,\\ t = 0 \text{ elsewhere, } \forall x,y \in R_n(x,y)\}$$

$$R_n = f(x,y) @ k_n(\varepsilon,\eta)$$

k_n = specific probe structure, the nth from the set of N to be applied

The set of probe structures is chosen for the appropriate elimination of particular shape and gradient combinations in the original image.

We might further define an analysis with respect to a particular set of increasing sizes of a particular structuring element. This will be used in conjunction with broadly isotropic probe functions to extract features at multiple scales. In this case,

$$k_n = ((\ldots(k_0 \oplus S) \oplus S)\oplus\ldots.\oplus S) \ n \text{ times.}$$

A conventional texture feature analysis can be generated from this methodology

$$\text{Locus} = \bigcap_n T_n(x,y)$$

where:

$$T_n(x,y) = \{t | t = 1 \text{ where } R_n(x,y) \in cond\, \mathrm{C}_n,$$
$$t = 0 \text{ elsewhere, } \forall x,y \in R_n(x,y)\}$$

$$R_n = f(x,y) @ k_n(\varepsilon,\eta)$$

k_n = specific probe structure, the nth from the set of N to be applied

C_n = specific conditions applicable to set n

The set of probe structures is chosen for appropriate properties, to eliminate particular shape and gradient combinations in the original image. We could, therefore, hunt for specific structures in the texture, characteristic of the texture under analysis.

We might further define an analysis with respect to a particular set of increasing sizes of a particular structuring element. Toet [18] and Verhoees [41] have discussed the decomposition of image structures using sequential applications of openings. Progressively larger dilations of a flat probe structuring element are applied. The objective is to produce a signature characteristic of the texture under test. The particular benefit of using a flat structuring element is the resultant ability to reconstruct the original image from the data eliminated in each sequential application (non-flat structuring elements preserve discontinuities in the original). The equivalent process might be of interest, particularly if the probe structuring element were chosen for the identification of a small set of spatial components. The increased size would then correspond to a different small set of spatial components. This is approaching a form of spatial domain component analysis, similar in concept to a frequency domain spectrum analysis.

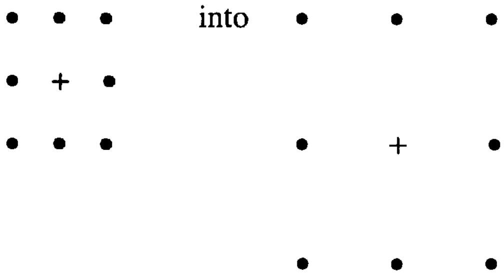
The main difference is in the generation of the structuring element. Other researchers [12, 18, 19] have discussed the use of dilation to generate increasing sizes of templates, and this is appropriate for the use of block type structures. The basic method would involve the use of a point function for k_0, with the structuring element for dilation, S, possessing the spatial attributes required by the analysis. This preserves the required attributes as the increasing sizes of structuring elements are created:

$$k_n = ((\ldots(k_0 \oplus S) \oplus S)\oplus \ldots \oplus S) \quad n \text{ times}$$

where S is the (greyscale) dilating function, and k_0 is the base template, usually a point function.

For the sparse templates used later in this thesis, the base template, g_0, possesses the basic spatial attributes needed for the analysis, and the dilating function capable of the linear increase in size of the basic shape (unfilled) is used for S.

That is, the dilating function capable of transforming



Such a template is an imaginary function, chosen for notational compatibility.

The texture features could now be defined in terms of the histogram (probability density function) of occurrence of each rank of the template in the image, for each size of template applied. The elements, h_i, of the histogram H_n for each size of probe function k_n applied are defined as:

$$H_n = \{i \in CARD\ k_n, h_i = \sum_{x,y \in R_n} m(x,y),\ \ m = 1 \text{ where } r(x,y) = i,\ m = 0 \text{ otherwise}\}$$

[16]

$$R_n = f(x,y) @ k_n(\varepsilon, \eta)$$

$$k_n = ((\ldots(k_0 \oplus S) \oplus S) \oplus \ldots . \oplus S) \ \ n \text{ times for particular size } n.$$

We arrive at a characteristic histogram at size n, $H_n = \{H_i, i \in CARD\ k_n\}$. The overall feature descriptor will be the set of sets $\boldsymbol{H}$, where $\boldsymbol{H} = \{H_n\}$, where $n \in (0, N)$, the set of N probe functions k_n applied to undertake the analysis. The use of this work will be discussed in Chapter (5), section (5.7).

2.6 Summary

The loosening of the conditions on formal erosion has been achieved, offering a potential solution to several problems including the use of erosion as a marker of structure position in a less than perfect environment. This is based on the rank of the point under consideration in the difference domain over the structuring set or function area. As has already been shown, certain advantages accrue in terms of noise rejection in opening and closing, with conditional closing completely removing 30% full range salt and pepper noise, using asymmetric conditions. The ability to grow dark areas in images without the blocky visual effect which results from formal erosion is shown to occur with the use of the conditional erosion.

A simple example of the use of the RJ operator for set occupancy analysis has been demonstrated with incomplete data sets, successfully identifying two overlapped shapes from their orientation spectra.

Three options are available for occupancy measurement of an image function by a probe function, in the form of the direct R operator result, the mostly hit, mostly miss transform, and the intersected R analysis. In addition, an erosion form suited for assessing set occupancy with multivalued and binary data (such as might arise from the feature extraction stage of a recognition system) has been demonstrated to offer benefits with damaged data from overlapping shapes in an image. The use of these operators in feature extraction, recognition from extracted feature data, and texture description will form the basis of the majority of the practical work described in later parts of this work.

Chapter 3:

Theoretical Development of Algorithms 2: Recognition Algorithms

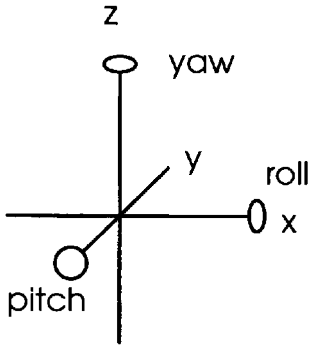
Chapter 3:

Theoretical Development of Algorithms 2: Recognition Algorithms

3.1 Introduction

In discussing recognition, it will be helpful to first define the axes and field of view used. Consider figure (14) below:

Figure (14): Axes and Field of View



In dealing with the recognition of objects with freedom of rotation, translation and scale, the problem of interpretation requires some means of limiting the data used by any algorithm. Speed makes the use of the full richness of the relationships between pixels in even relatively low resolution CCD arrays untenable. We are left with a need to generate some simpler description of the picture under test, and the objects which may (or may not) be in it.

Such a description should be as full as is practicable, and amenable to processing using current computer technology. Ideally, it should be invariant under rotation, translation and scale, and preferably easy to generate from pictures or by direct data extraction from design files. The description should be flexible enough to permit recognition from poor quality data, and from damaged data due to partial overlap and obscuration.

These generic specifications indicate the use of a feature based approach as the probable solution. A feature based approach will allow a sliding scale of description. Provided features are selected correctly, certain faces of the feature model will be visible at any rotation and translation within the field of view. Normalisation of scaling is also possible. The feature based approach can permit definition of the model in the form of a data set, and thereby permits the use of set theory to minimise the computational complexity inherent in dealing with this type of data.

In conjunction with appropriate algorithms, recognition of incomplete data will be possible by using a "best fit" approximation. An objective measure of the degree of confidence in the recognition result will be generated by assessing the degree of occupancy of the model set by the data under test.

A feature based approach used with library models to identify objects belonging to the (necessarily) limited world of the system will be used. This does require additional work to extract appropriate features for use in the recognition process, but the reduction in later processing should compensate for this. It is not unusual to disregard the low level feature extraction work when estimating the computational cost of a recognition algorithm [28].

3.1.1 Features: Local or Global Descriptors?

This is a classic area of consideration for recognition algorithms. The benefits of global features are well established, and are best characterised by two approaches, the method of moments [29], and Fourier descriptors [30]. Both rely on the whole shape being present, and are consequently affected when partial obscuration occurs. A set of shape descriptors for each orientation of the object is calculated relative to the viewer, and a given view is matched against a library or database to evaluate the object. Such descriptors are usually robust and rotation invariant, but many of them may be required to guarantee the uniqueness of the stored descriptions for a particular object.

Global descriptors, in general, are susceptible to noise [31]. Partial obscuration or overlap of objects within an image will degrade the identification ability of these methods. Their use is relatively simple, in that a descriptor set is calculated and evaluated against the stored library of models. A minimum distance approach is used to calculate the best fit object.

Local descriptor methods use the aggregation of multiple small area descriptors to generate a recognition match. These methods are inherently more robust in the presence of overlap and obscuration. They rely on the ability to identify key features (usually areas of high curvature [32]), and are therefore susceptible to the ability of the extraction algorithm to identify and correctly locate features. This can be particularly important for orientation estimation [33]. Various classifiers may be used to evaluate the best fit object, including the minimum distance approach and clustering techniques.

Local descriptors offer greater flexibility in recognition for real world conditions, but impose complications of interpretation. For this reason, and for the possibility of effective direct feature extraction using morphological techniques simplifying the underlying algorithmic architecture of a recognition system, the majority of the work in this project has concentrated on local descriptor features.

3.2 The Problem of Freedom of RST in 3-Dimensions

The degree of constraint on the objects and the domain of the objects considered within the world model of the recognition system determines the difficulty of obtaining recognition. Rotation, translation and scaling, if constrained, will offer the possibility of using one single non-ambiguous view of the object under test for comparison purposes. The features to be used could be searched out in their exact locations. Unfortunately such constraints are by no means acceptable for the broad sweep of recognition applications: they are typically only found in certain industrial inspection applications.

The problems imposed on a recognition system by rotation, translation and scale will be considered separately, and used to suggest the attributes needed to compensate for each. From this discussion a general outline for a recognition approach allowing for compensation for RST will be drawn.

3.2.1 Rotation

The effects of rotation are explicable in geometric terms. A rotation matrix may be defined as [45]:

Rotation by θ about the X axis:

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

about the Y axis:

$$\begin{matrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

about the Z axis:

$$\begin{matrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

The basic effect of rotation is to shield features from the plane of view. Under rotation, the basic rules of geometry apply:

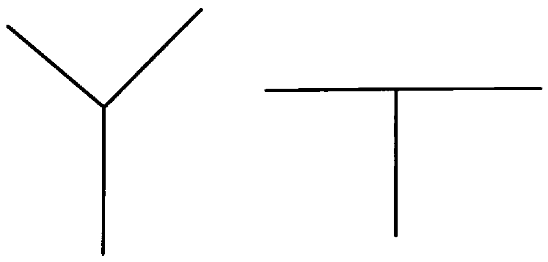
(i) points map to points;
(ii) lines map to lines;
(iii) planes map to planes;
(iv) ratios of measures of angle and length are not preserved in point views.

Any problems with apparent changes of features is a result only of the constrained plane of view. Defined features will map to corresponding positions, all moving through the same change of orientation.

The problems caused by rotation remain twofold.

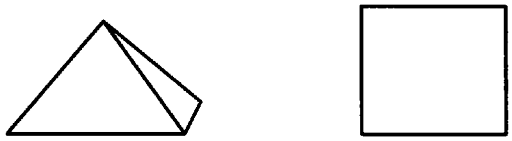
Features such as the intersection points of three planes can, in the worst case while still visible, map to apparent intersections of two planes (see figure (15) below). This will necessitate some form of consistency test for features to verify that the views are feasible.

Figure (15): Rotation of Intersection of Three Planes



The second problem is that of features obscured in the view by the body of the object. The most obvious case is shown below in figure (16). This problem is fundamental, physical, and not capable of solution without constraining the model world to objects unique in every view. Such an approach will not meet the requirements of many simple everyday situations.

Figure (16): Solid Pyramid and Its Rotation Through 90 Degrees about Z-Axis



An "intelligent" approach could be simply to guess (with obviously low confidence) the most likely object based on previous populations. Others might be to target a constrained search where other features exist within the view amenable to classification, or to wait and probe for another view of the object under test.

The result of this discussion is to suggest that any feature based algorithm should contain an object centred model of the objects, either in each discrete feature pose or as a self-contained set capable of generating each feature pose from limited data.

In addition, because length and angle are not preserved in the plane of view, except in so far as acute, obtuse and reflex angles remain in their separate domains, selection of features should not rely on single measures to obtain a recognition. Single simple distinctive features (the meeting of two planes at a distinctive angle, a particular gradient of curve) are not guaranteed to be visible in their useful form.

3.2.2 Translation

(i) Far Field

The effect of translation of the object within the plane of the object may be defined using orthographic projection. Perspective effects are implicitly of such a small order as to be ignored.

Where the point of view is constrained in the acquisition rig (for example by a fixed camera location), the effect of orthographic projection may be likened to a combination of a rotation and a linear translation. This is a far field approach where perspective variation is not a significant factor.

(ii) Near field

Under near field translation, the rules of perspective geometry apply. The techniques engendered for rotation will offer solutions. A perspective shift can be viewed as a rotation combined with a non-linear translation.

Perspective distortion matrix:

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/f \\ 0 & 0 & 0 & 0 \end{matrix}$$

where f is the equivalent focal length (separation of viewpoint and object plane).

3.2.3 Scaling

(i) Linear Scaling in Three Dimensions

Scale invariants include relative positions of features, sequences of features, relative lengths of edges, orientations of edges and types of corners. The usual compensation method is to normalise scaling between model and object. The normalisation factor then gives a direct measure of the scaling in the image under test.

A relatively intractable problem occurs where partial obscuration and overlap are permitted. To engender the greatest accuracy in scale estimation it is obviously necessary to use the longest edge (the greatest separation of features). It will also be necessary to normalise the scaling to permit recognition from a library model. Where partial overlap occurs, any attempt at normalisation based on maximally separated features will immediately invalidate any feature information based on relative positions of edges, corners, etc.

The sequence of features and neighbour separations remain invariant. Scale estimation based on adjacent feature pairs will not offer the highest possible accuracy (within the limits of the picture digitisation) but will enable recognition to occur on a trial and error basis.

(ii) Independent Scaling in Each Axis

This further exacerbates the problems. The only reliable information will be the sequence of features. This will have to be used to obtain a recognition, from which the relative scalings in each axis can be back-calculated.

The general scaling matrix would be:

$$\begin{matrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

where Sx, Sy and Sz are scaling factors in X, Y and Z directions.

3.2.4 Conclusion

The effect of freedom of rotation, translation and scaling is to increase the complexity of the recognition problem. Only the most fundamental of information can be used as model features due to the variance of most forms of data under RST.

The problem is amenable to decomposition.

Freedom of rotation will allow the shielding of important features from a single viewpoint (camera). We may therefore use an intersecting combination of viewpoints giving a complete coverage of the object under test, but this will only be possible under tightly controlled conditions. The alternative is to store multiple views of the object in the library. For many objects the three views of design drawing (six views for asymmetrical coverage) will be sufficient if correctly chosen and orthogonal. Unfortunately this does not cover the general set of objects, and in many cases more views will be needed.

There are significant benefits gained from using multiple stored views of the object. The degree of rotation to align the matching view and the picture under test will be relatively small, allowing the use of the popular inverse affine transform as a means of identifying the rotation of the object from the stored view (see Appendix A.3). This will also prevent the problems of matching reflected views to separate visual isomers.

Scaling necessitates normalisation. For normalisation before analysis, an a priori knowledge of the scaling factors is necessary. This allows for correction of scale changes and greatly eases the recognition problem by allowing matching of feature loci as well as feature types.

Where scaling is not known, it must be inferred after object identification. This will introduce intractable problems, particularly with simple objects (try differentiating between a non-uniformly scaled square block and a similar rectangular object). If the set of features displayed matches only one object in the library as opposed to a class of objects then the scaling can be extracted by normalisation against the library model.

For each view of an object, then, we will require a feature model which includes feature location data as well as feature types (needed also to extract rotation) but which can be used without any location data other than relative position to other features if we intend to compensate for, or to measure, scale variations. The collapsed model would simply indicate nearest neighbours or "up-down-left-right" relative position in the view.

Translation in the far field is the most tractable of object recognition problems, assessed by indicating the location of a key feature and amenable to any system capable of handling rotation problems.

As with scaling, correction for perspective distortion is desirable before matching. If such an approach is not possible and the separation of viewpoint and object plane is unknown then the rotation collapsed model may offer a solution. Combinations of scaling, rotation and perspective offer a particularly severe test for any recognition algorithm claiming the ability to also extract rotation and scaling.

The model generated for each view of the object must include absolute spatial information as to the relative positions of features, types of features, and some simpler logical indication of their positional relationships.

3.3 Structural Features

The benefits of a feature based approach relate primarily to handling limited object information. An attempt can be made to classify the usefulness of various types of data situation at this point.

The previous discussion illustrates that invariance under RST is not characteristic of the majority of what are usually intuitively obvious features. Of more use than the absolute measure of an object feature is its relationship to other (hopefully defined) features. We shall therefore place more reliance on local clusters of features than on the single "unique" feature where high reliability of identification is a necessity.

The ability to map a best fit solution between a model and a test data set is therefore of primary importance. Where imperfect data is used, as is typically the case with real images, the closest match between acquired data and a library model will be taken as the identification of the object. This should be conditioned by a goodness factor for the recognition or a confidence factor in the match.

Structural features can be extracted from design drawings for mechanical parts and manufactured items. They can also be directly extracted from acquired data, offering the possibility of supervised learning for model derivation. Unsupervised learning would require a high quality training set of images, low in noise and complete in degrees of view, for the optimal performance, but is by no means precluded as an option for creating library models.

The choice of object perimeter structures of high curvature (or low curvature) follows the apparent logic of perceptual reasoning (see Davis [32]). The utility of their use in recognition is well documented.

3.3.1 Selection of Features for Models

In defining the model features used for library data, the effects of RST determine to a large extent those chosen. It is not difficult to see why the great majority of work in the object recognition field has concentrated on geometric objects which are relatively rich in well defined and usable intersections of planes, and characterisable corners. These same features are of primary importance where available, but are exceptional rather than normal in the general set of objects. Table (3) below offers some suggestions for structural feature types to be included in models and searched for in acquired data. It should be noted that the chosen features will need to be extracted from the image, which is (in this work) an intensity mapped view of the scene.

Table (3): Structural Features offering Invariance under RST

Geometric objects:

intersections of 3 planes (colloquially known as corners), concave and convex
intersections of 2 planes (edges), concave and convex
plane surfaces

Non-geometrical objects:
points of high curvature (points), concave and convex
points of low curvature (lines)
points of inflexion of curvature (saddles and colls), concave and convex

Cain and Bolles [34] have reported on the use of small local features such as holes to extend the richness of the feature set. The well-known Clowes algorithm [35] interprets the intersections of line edges as features for recognition.

The actual choice of features is made on a pragmatic basis - those that we can extract from the image reliably and repeatably. In practice this means examining the intensity map for particular structures, which are interpreted to indicate the presence of a particular feature. It is natural to look for particularly high or low rates of change in the intensity map as a first basis for evaluation.

Extraction of these features is an entire and major area of work in itself. The approach adopted for this work is detailed in Chapter 4. It should be noted that the success of feature identification controls to a large extent the effectiveness of any recognition system of this type and the confidence level in any results obtained must be conditioned accordingly.

Three approaches were considered for modelling objects: feature aggregation, use of silhouette features only (i.e. those features appearing on the silhouette of the object in the given view); and combining silhouette and in-object features.

For the most part the work concentrated on structural features, but the inclusion of (for example) colour parameters, or textural descriptors, is by no means precluded by the technique developed. Such descriptors would add to the richness of the model and its general utility, but at the cost of higher dimensionality both in the feature listings and in the processing requirements.

The model database was created by storing the set of features (the exact form of the feature description is shown in the following four sections 3.3.1.1-3.3.1.4). A feature set was stored for each pose of the model. In analysis, the feature set for each pose of the model is compared with the acquired feature set extracted from the object in the image under test.

The selection of model poses was made on an intuitive basis, additional models being added where significant differences appeared in the feature set for that particular view. Changes in the feature set occur as the rotation causes particular features to be obscured by the body of the object, or as other additional features appear in the field of view. Where the changes accumulate to the point of requiring the addition of another pose to the library, the pose is added. This decision is made on the basis of the requirement that each pose remain a unique identity within the library, and sufficiently separated from the existing poses to allow discrimination on the basis of the extracted features.

The number of views required varied according to the nature of the object. The cube, offering the simplest example, required three pose models for adequate identification. The hawk trainer (IS6) required a set of twenty four poses for definition within the model world used.

The number of poses stored has implications for the performance of a recognition system in both orientation resolution and identification under occlusion. A close relationship between the orientation of a stored pose and an acquired view allows a high correlation between the stored and extracted feature sets. This can simplify the calculation of orientation. In particular, where features are missed due to extraction problems or occlusion, the matching of the remaining features to the library pose is more likely to be successful, and will offer a higher confidence in the result. The cost resulting from an increased resolution of poses is paid in speed terms: the number of comparisons between object and database is increased.

3.3.1.1 Modelling by Feature Aggregation

When dealing with a feature based approach, the simplest modelling method is to regard each feature as independent and to model the object view as a list of features by type. Recognition is obtained by comparison of the extracted features set from the image under test with the library views. No information about relative positions of features is included: a feature is either present or absent in the view. A match is accepted if enough coverage of a library model occurs in the acquired image.

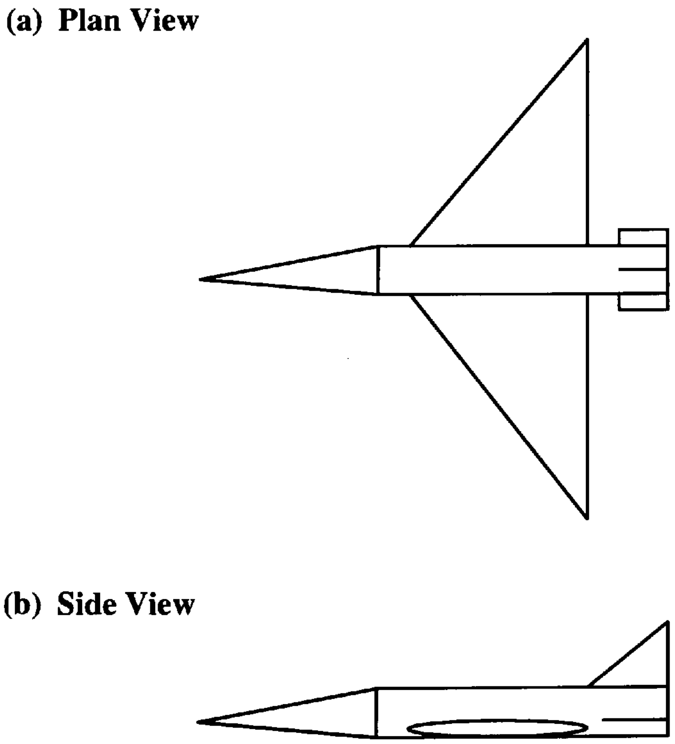
This method is intuitive, and offers low complexity in algorithm development (ordering lists of feature types by sorting, followed by comparison). It will allow simple modelling and a high probability of recognition where single objects of distinctive features appear within the image.

In the presence of a rich set of objects within the system world, unique models are not guaranteed. This is primarily because such a method ignores the sequence or relative positions of features, and therefore omits a significant facet of information about the object.

Feature aggregation is prone to aliasing and will be inefficient in a noisy environment where pseudo-features may appear. In particular, where multiple objects are present in the image, this method will give false results.

An example feature data set for a simple aircraft model is shown in figure (17) below. Gross structural features were used, including points of high and low curvature, long straight lines and line ends. These would be extracted from sharp gradient changes, or ideally discontinuities, in the intensity map. They would, therefore, naturally include the perimeter features, which tend to dominate for this simple aircraft.

Figure (17): Simple Aircraft Model



(a) Plan View

(b) Side View

Feature Aggregation Set Models

	(a) Plan View	**(b) Side View**
Convex high	7	5
Concave high	6	3
Long straight	7	6
Line ends	1	1

For a feature extraction technique using high curvature on the perimeter of an object as the means of location, the model becomes:

model = {no. of concave high curvature areas, no. of convex high curvature areas}

3.3.1.2 Modelling by Silhouette

The silhouette of an object is rich in feature information, and is easily extracted where a reasonable contrast has occurred. It therefore offers a usable modelling method, the features for comparison being efficiently extractable from the acquired image. In the biological world the silhouette is a key recognition parameter - consider the variety of camouflage used by organisms to disrupt their apparent outline to avoid predators.

The features are extracted by segmentation of the object or objects within the image followed by edge following to extract significant information. Silhouette models give a list of edge features in the order they were acquired, thereby preserving feature sequence information around the perimeter of the object view.

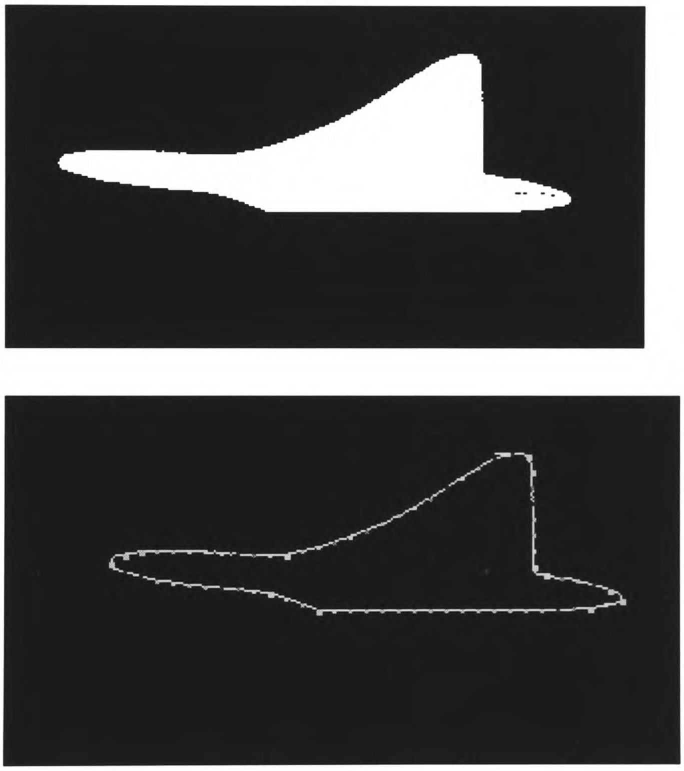
In order to minimise the computational effort required in comparison, the model may simply list the sequence of features. This will yield family type information on the first pass (will not separate simple objects like rectangles and squares). Where higher confidence is required, a second pass including relative co-ordinates of edge features (in x,y) can be used. This will yield the ability to extract rotation and scaling information and will resolve between dissimilar objects with the same sequence of edge features.

The model then becomes:

$$\text{model} = \{feature_i, type_i\} \forall features\ i \in (0,I)\ on\ the\ perimeter$$
$$extracted\ in\ a\ clockwise\ or\ anti-clockwise\ direction$$

The major benefit of preserving sequence information is the inherent ability to operate on chains of features. Where overlapping or obscuration occurs, the sequence of edge features is disrupted. Longest chain sequences offer a means of identification of objects from partial data with confidence estimates based on the length of sequence in comparison with the full perimeter expected. Thus recognition is possible even under partial data conditions. Figure (18) below illustrates this point, along with table (4) overleaf.

Figure(18) Partially Obscured Aircraft with Extracted Features in Comparison with Library Model View



Where the features are extracted directly as isolated items, rather than by a sequenced location process (such as tangential curvature estimation, or normal contour distance), sequence can be artificially introduced as, for a set of features *F*:

$$\text{Centroid of Features } (x_c, y_c) = \sum_{0}^{N} \left(\frac{x_n}{N}, \frac{y_n}{N} \right), \text{ where } (x_n, y_n) \in F$$

$$\text{Model } = \{(type_1, \theta_1),..(type_n, \theta_n),..(type_N, \theta_N)\},$$
$$\text{where } (\theta_1 < \theta_n < \theta_N) \text{ about centroid } (x_c, y_c)$$

Table (4): Perimeter Feature Set from Partially Obscured Aircraft

Extracted Feature Set:			**Original Feature Sequence Data (Note no locus information is included):**
X	**Y**	**Type**	**Type**
339	293	1	1
227	341	0	0
155	349	1	1
235	363	0	0
259	369	1	1
395	366	1	0
397	359	1	1
359	339	0	0
348	290	1	1

Key:

1 = high curvature convex
0 = high curvature concave

The major problem with the artificial sequencing of features is the disproportionate effect of missed outlying features (such as the nose, or wing tips of the aircraft of figure 17) - the sequence will change is the centroid moves far enough from its proper locus. Note that the orientation about the centroid could be used could be used to offer rotation information for a particular object pose.

The silhouette approach is widely used, and has been reported extensively (e.g. Wallace and Wintz [36]). The problems of feature extraction for this approach have also received much attention.

On a practical note, where the objects or background in an intensity-mapped image are textured, it may only be possible to delineate the outline of the particular regions. After the regions are separated and assigned particular values, the boundaries formed between them may still be used to form a silhouette from which features could be extracted for recognition purposes..

3.3.1.3 Feature Modelling including In-object Features

This method includes as a subset of its features the outline information used in silhouette based methods, and therefore has the benefits of edge sequence information built into the approach. Edge following cannot, however, be relied upon to produce the feature set needed for recognition purposes, and direct methods have been used (see Chapter (5), Sections 5.3.2 - 5.3.4).

A full formal model includes both feature types and locations. Feature locations may be defined in only x,y and need not include z-plane information where multiple views of the object are to be used from the library (z-plane information can be generated after the recognition process is complete if required). This will be used to extract rotation and scaling information from the image acquired.

A collapsed model is needed in order to minimise processing effort. Several approaches might be adopted. Initial testing using silhouette only is an obvious choice. One method would be to retain perimeter sequence information (the method as for the silhouette collapsed model) and aggregate features within the object. An alternative would be to collapse the full feature model onto a point radius circle, yielding simply the adjacencies of features and not full pixel-relative position information. Note that this partially ignores the z-plane position, only including it as relevant to the particular view, but retains the relative positions of features in a non-linear way.

3.3.1.4 Feature Modelling using Position of Features and Local Web Skeletons

In this approach, the features used would be points of high curvature. The object is modelled as the points of high curvature interconnected by a nearest neighbour web. The model therefore contains both the points of high curvature on the surface of the object and a local nearest neighbour description. Such an approach would enable recognition from partial views of the object and might prove more robust under noisy conditions, in that local clusters could be aggregated to give a higher degree of confidence in the overall recognition. Models could be extracted from design drawings for mechanical items, or directly from image data for comparative recognition.

The model is generated by producing a list of the orientations of all the other features within the acquired set from each feature in turn. Although this may seem exhaustive, given the high likelihood of missed features, such an approach maximised the probability of a correct recognition.

The model then becomes:

Model $= \{(type_n, \theta_{nm})\}, \forall m \in F, \forall n \in F,$
where $F =$ set of extracted features; $m, n \in F$.

This is illustrated in figures 19 and 20 below.

Figure (19): Points of High Convex and Concave Curvature on the Perimeter of the Aircraft of Figure (17)

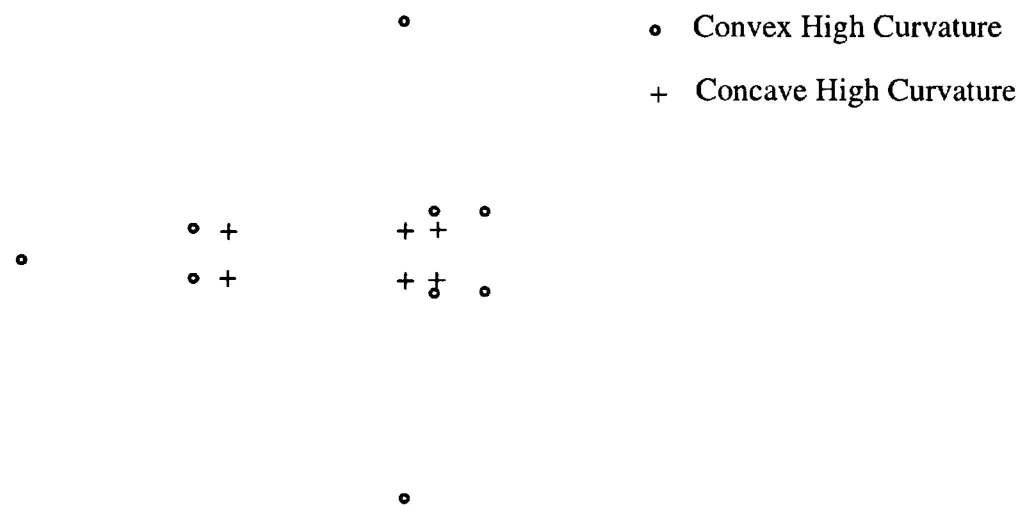
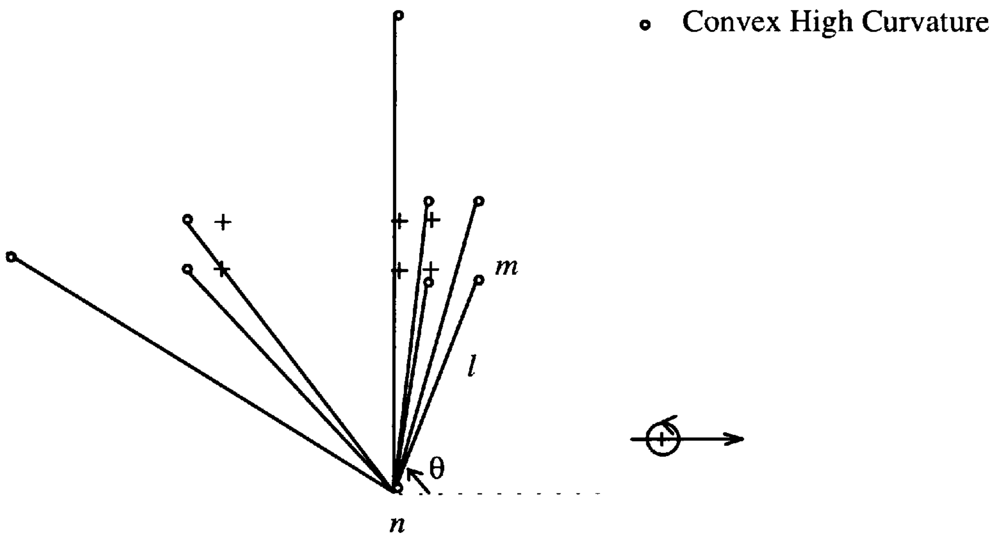


Figure (20): Web of Points of High Convex Curvature on the Aircraft Perimeter



Model = $\{(type_n, \theta_{nm})\}, \forall m \in F, \forall n \in F,$
where $F =$ set of extracted features; $m, n \in F$.

Note that the full extracted model used for rotation and scale estimation includes the separations of the features l_{mn} as well as the type of feature and the orientation of l relative to the grid. The use of this information allows estimation of the foreshortening caused by the rotation of the object relative to the pose of the model. This approach is close in structure and intent to the local-feature-focus of Bolles and Cain [34], but relying on vertices rather than pre-identified small structures such as holes.

Figure 20 illustrates the web of interrelationships between one point of high curvature and the others extracted as a feature description of the object. This is a simple example showing the relationships of one particular feature. An approach based on one focus feature (the example used here is a wing tip) would be highly vulnerable to the absence of that specific feature. It is therefore necessary to replicate the web description for each extracted feature. This approach offers the highest resistance to missed features in the matching process. The full web resulting from such an approach is omitted for reasons of clarity.

Thus the relationships of each feature in the extracted set to the other features belonging to the object forms the basis of the full description used for comparison.

3.4 The Recognition Algorithm

All of the feature based methods of modelling objects described above have a common strategy for recognition. The process consists of extracting features from the acquired data and comparing it with the library models. The crux of such an approach is the ability to determine the best fit of the data onto the library model and then evaluate the information as to recognition (usually according to some sort of minimum match threshold) and later to scale and rotation estimation. The recognition process usually consists of matching pairs of features between the library model and the acquired data.

For comparison of feature sets the RJ operator will be used. The overall algorithmic approach will now become:

The Algorithm - Overview

Image Acquisition

Low level pre-processing (if necessary)

Feature extraction

Matching features with model library

In keeping with the overall aims of this work, a morphological means was developed for achieving a model match. The extracted data set was compared with the pre-stored library models using set erosion. This produced results as shown in sections 6.2 and 6.3.2. The precision of set erosion resulted in problems with damaged and sparse data. The application of the conditional erosion method yielded an improved tolerance to such problems.

The Morphological Algorithm - Overview

Acquire image (standard camera/frame grabber approach)

Feature extraction (using greyscale RJ operator and templates, as section 4.6)

Match features with model library (using binary RJ operator, as section 3.4.3.2)

Extraction of RST from identified object (using standard techniques)

3.4.1 The Classifier Strategy

Many classification strategies are in current use (for an excellent description of a wide variety, see Schalkoff [1]). A simple method of counting paired structural features between the acquired data and the library model was initially adopted. A recognition match was taken to be the highest correlation between library model and acquired set. An elementary degree of confidence was established by simply taking the percentage of model features correctly located. Although simplistic, such an approach has yielded surprisingly good results as has been demonstrated with set erosion as the marker (see Rees, Jones [37]]).

Each of the modelling techniques described previously in sections 3.3.1.1 - 3.3.1.4 were implemented and tested for a variety of objects and conditions. The model library was considered in two forms, as a single complete model for the object, and as separate views of the object with each view being independent of the others. The models were evaluated against the acquired data using the RJ operator. The $\underline{R}$ set yielded the most appropriate rotation of the model view against the features extracted for closest matching. The J_r sets yield (as is discussed later) the missing features, which can be used to enhance the confidence in the recognition result through simple consistency checks.

3.4.1.1 Extraction of Rotation, Translation and Scaling

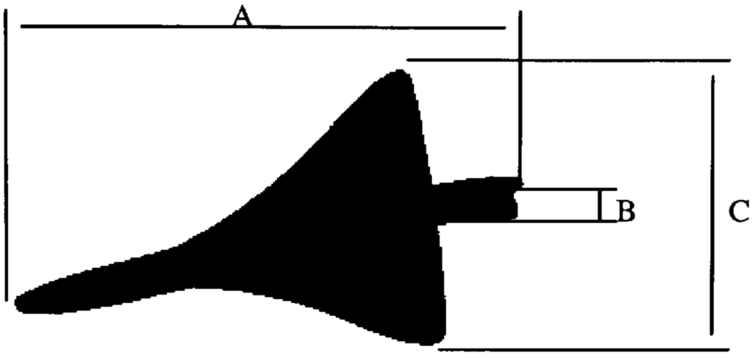
The subset of the structural features of a 3-D object visible to the camera changes with rotation of the object. Whilst the selected features may be broadly invariant, the rotation of the object always obscures part of the set. In the case of a manufactured object the design drawing set will include feature information adequate to the manufacture, and therefore the recognition, of the object.

In the general case, two choices are obvious: to model the entire object and accept the features in the view as a subset of the total object feature set, or to use several partial views of the object. The former option has the benefits of reducing the numbers of models to be tested against the acquired data set, but at the price of more complex modelling. The number of features included in any view is also likely to be a low proportion of the overall model features, increasing the susceptibility to false classification. The latter option requires increased numbers of models (partial models) but will allow calculation of orientation relatively easily through affine transformation. It should offer a more robust approach because of reliance on relatively higher proportions of features present.

The practical difficulties in modelling the whole object (the points of high curvature must have a "shield" area between them and the centroid of the object, and elsewhere, indicating the directions from which they are obscured as the model rotates) rendered this approach untenable. A solid modelling for a manufactured object, a technique supported by current CAD software packages, might offer a suitable means of achieving the goal of a single model, but at very high computational cost.

The estimation of rotation and scaling follow the recognition process. Once the particular object has been (roughly) identified, the features are then rotated using the affine transform to assume the same relative positions as the model set. The positional data is then scaled to bring them into the closest alignment with the model set. The inability to produce a reasonable alignment between model and acquired data will be used as a means of determining a failed test. As part of these transformations the orientation and scaling relative to the model set are generated.

Figure 21: Extraction of RST from Recognised Object



Rotation was extracted after recognition by using the separations of three widely dispersed, identified features within the view and simple geometry, as in figure (21) above.

Scaling was calculated from the two most separated features in the view, allowing for rotation.

Translation was generated simply as a pixel position in the image for a chosen feature.

3.4.2 The Use of Set Erosion for Recognition

The binary set erosion process allows identification of all points within an image under test at which a probe set is fully contained. It may therefore be used for identification purposes.

3.4.2.1 With Orientation Spectra

The set erosion technique described by Rees and Jones [36] required a reasonably complete extracted data set for quality results. The reasons behind this are straightforward. The modelling method used is very similar in practise to the normal contour distance described by Vernon [38] and in Chapter 4, section 4.2. The erosion method locates the probe set at its best position relative to the acquired data. It is a purely spatial arrangement, and intolerant to extraction difficulties moving the apparent orientations of the sides plotted as length l against orientation θ. Missing data is simply described by aligning the orientation of the acquired data with the reference model and subtracting. The major problem encountered was the high sensitivity to small changes in the acquired data. Alignment with the grid produced orientation differences of significant order, requiring the orientation accuracy of the algorithm to be degraded in order to get a result.

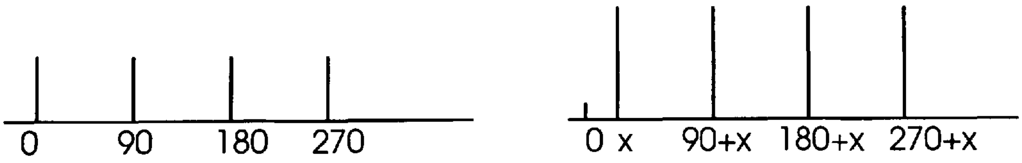
This was a direct use of set erosion. Initially the binary set erosion was used, presence of a shape being determined by erosion of the model library against the extracted perimeter normal orientation set. This simply determined the presence of a particular orientation within the acquired data. It was unable to separate squares and rectangles, and would agree any shape against a circular model.

In order to improve the result, the 1-D binary set erosion was replaced with a 2-D binary erosion and the model enhanced to include both the orientation of the perimeter normals and the length at that orientation (effectively producing a close equivalent to the normal contour distance). The model now became for each side n:

$\{(\text{orientation } \theta_n, \text{ length } l_n)\}$

Consider the example given in figure (22) of the orientation spectra for a square (library model) and its acquired, scaled, rotated analogue.

Figure (22): Orientation Spectra for Library and Acquired Data



A = {(0,a),(90,a),(180,a),(270,a)}

B = {(x,b),(90+x,b),(180+x,b),(270+x,b)}

We obtain:

AΘB = {(-x, b-a)}

This confirms the shape is actually a square, rotated through x degrees and with a scaling difference of (b-a)/b. Scaling and rotation are extracted as the offset of the probe set needed to achieve complete containment, i.e. as the result AΘB.

Again, the susceptibility of this approach to small errors in estimation of side orientation is obvious. One method of alleviating the problem is to degrade the shape resolution and the orientation precision of the estimation by allowing bands (e.g. 90 degrees +/- 2) for the acceptance of the presence of the particular normal. This approach is very similar to that adopted by Grimson [39] for recognition of rigid curved objects from 2-D data. An alternative is to modify the set erosion to permit a degree of uncertainty in the result. A third method would be to change the modelling method for a more appropriate approach.

The method of reducing the precision of orientation was adopted in practise and permitted the successful use of this set erosion method for 2-D images of 3-D objects with some limited amendments to the technique. Due to the line splitting and deviation in extracted orientation caused by the quantisation of the edges of the object the aggregation of weights over local orientation bounds was necessary. The reasoning behind this can be easily understood - in order to get a 1 degree accuracy in orientation of a straight line sequence on a square grid, basic trigonometry indicates that a minimum of 57 pixels in the line is necessary. Dorst and Smeulders [40] includes an interesting discussion of representing quantised straight edges and the corresponding quantisation errors.

The "bucket" regioning of orientation limits the precision of shape identification and orientation estimation.

3.4.2.2 With Feature Sets

The application of the set erosion method with feature sets was prone to the problems already discussed. Where a complete set of features for the model view was obtained, successful recognition was found to occur. The requirement for full feature data within the extracted information was a major limitation, particularly where the quality of image (due to lighting or noise problems) was imperfect. The susceptibility to a single missing feature required either the selection of gross features alone or the inclusion of all apparent features in the extracted feature data. The former leads to poor resolution; the latter to aliasing. Where positional data was included in the feature sets (rather than just sequence) the problems of localisation were found to be as severe as for orientation spectrum analysis.

3.4.3 The Use of the RJ Operator for Recognition

The inclusion of boundaries in set erosion is essentially an unwieldy exercise, having poor properties for multiscale applications. Having introduced the possibility of controlled uncertainty into the erosion process with the RJ operator, its application to recognition is obvious. As has been described, the R set is generated as the number of contained components of the probe set within the set under test, in the local area defined by the probe set. This enables the extraction of the best fit locus (loci) $\underline{\mathrm{R}}$, at which point the probe set need not be fully contained.

In the situation where structural features are used for recognition, this approach offers its maximum benefits, in that recognition may be achieved in the presence of partial data with known confidence levels. The absence of features does not preclude identification, as would be the case with set erosion. The real problem lies in obtaining adequate sophistication in the modelling process to avoid aliasing by textural detail and noise.

In the presence of fixed rotation, scale, and translation, the RJ operator allows the identification of the best fit locus $\underline{R}$, and any features omitted from the feature set as J_r. Uncertainty about exact locations, possibly caused by sampling problems, could be catered for by including multiple points in the reference model for each real feature. This is, of course, the trivial case.

3.4.3.1 Use with Orientation Spectra

The RJ operator does not provide a complete solution to the problems with orientation spectrum approach. The fundamental problem of accuracy in the extraction of the orientation spectrum still exists: where the side orientations cannot be obtained with the requisite precision, components which are actually present will not be accepted unless some gross division of the orientation space is applied. Some alleviation is achieved, in that the absence of components within pre-determined limits is obviously permitted, and therefore an initial attempt at a "best" fit followed by a constrained search for the missing components can be used.

3.4.3.2 Use with Structural Feature Sets

Modelling using structural features is one method of reducing the dependence on ability to extract accurate orientations of lines in the image. Instead the problem becomes (usually) one of identifying points of high curvature in the image. The process is equally valid for 2-D shapes and 3-D objects, is probably more intuitive in its application, and may produce models relatively simply for complex objects.

The RJ operator is used to identify the best correlated probe set and the missing components necessary to complete the fit. The same basic technique is used for aggregated features, silhouette features, and silhouette plus in-object features models.

For each of the N models M_n in the library, with extracted feature set A, the best fit model and loci are defined as:

Best fit = $\max(R_{\underline{n}}/B_n)$

where

$R_n, J_n = A@M_n$,

$R_{\underline{n}} = \max R_n$,

B_n = No of components in probe set M_n, or CARD (M_n),

$j_{\underline{r}}$ = Components of M_n not fully contained in A at best fit.

With simple feature aggregation, the missing components are not particularly beneficial to interpretation of the result.

With the inclusion of feature location, the j sets offer valuable further information about the image scene. Tests for consistency and likelihood can be developed. For a pair of overlapped objects, formation of a closed 2-D shape projection by the missing features, or a consistent block, would add confidence to the resulting recognition of both objects. Where only a few features are found, the missing features forming a contiguous block might indicate obscuration.

Silhouette features are relatively simple to extract (see Chapter 5, section 5.3), and allow the use of a very effective recognition model in a variety of situations. The sequence information implicitly contained in a silhouette description gives high confidence in obtained results with the use of relatively few extracted features. This comes from the added significance if the features found form a contiguous segment of the silhouette perimeter. It is particularly useful where partial obscuration or overlap may occur. The contiguity of features allows chains of pairs of features to be matched, the best fit location usually offering a coarse indication of the difference in orientation between the acquired and reference data. The $\underline{R}$ set and corresponding J sets are used together to generate the result. The quality of the extracted data determines the efficacy of the operation.

With the use of in-object features, the problems are broadly the same. There is an increased likelihood of the absence of features, due to the greater difficulty in their extraction in this situation. The portion of the object set covered completely should give a higher degree of confidence in a positive result than a similar portion of the silhouette. The silhouette data is a subset of this model. If scaling can be pre-arranged, the inclusion of feature loci provides further information about the scene.

The webbed features approach, the RJ operator offers a means of identifying the best fit rotation between the extracted features and the reference model. Such an approach is tolerant of missed features - they are effectively point to point matched, rather than sequenced - and relatively tolerant of sampling errors. The sampling problem is reduced where widely separated features are available for consideration, and can be allowed for at any level by reducing the orientation measure precision.

3.5 Texture Classification

Texture is an important characteristic of many types of image, and is can be a major problem to the analysis of the information they contain. Texture is often an organised relationship based on a particular area or size of the image. It may contain structural grey level primitives, their structure being a characteristic of the texture perception, and may also be determined by the positions of these primitive structures relative to one another (consider wall paper manufacture, the repetitive sequencing of small structures to make an overall impression, as an example). The basic unit of texture is sometimes referred to as a texel. Note that this idea is not always valuable - many structures have random or changing textures, and defining a texel for such a texture is either trivial (each pixel) or inclusive (the whole image).

An objective might be, therefore, to define some characteristics of the texture based on its grey level structure, and the location and frequency of the primitives it might contain. The uses of such a description include the identification of a particular texture (as is typical in surface inspection, weave inspection in textiles, and certain medical classifications of tissues) or the actual segmentation of a test image into regions of differing textures for scene analysis.

Several standard texture measures are used, particularly those based on the grey level co-occurrence matrix, which attempt to classify the texture based on features (mean, variance, and a set of fourteen other parameters) extracted from the co-occurrence matrix itself. The co-occurrence matrix is a measure of the frequency of occurrence of pairs of grey levels in a given direction at a given separation. It is formed as a count of co-occurrence, plotted as a 2-D matrix with the grey levels as the axes. It is often used over short distance relationships (pairs of pixels one pixel, or two pixels apart) and is specified in terms of its direction relative to the sampling grid. Note that a texture aligned with the x or y direction of the sampling grid with have different spatial component placements if sampled at (say) forty five degrees to the grid. This causes problems with rotated, and sometimes with displaced, samples.

Haralick et al [60] have discussed the utility of this approach, their work having been developed and extended by later authors.

The method of application of the *R* operator is quite similar to that of Sun and Wee[61]. They report accuracies of 85% in classifying three textures of geological terrain types in LANDSAT images, based of use of various features such as entropy, energy, etc. They fix a distance *d* and a contrast threshold *c*, and determine the number of pixels each having a grey level *g* and each having *n* neighbours within distance *d* and within contrast *c*. The resulting distribution for an intensity mapped image *I(x,y)* might be represented as:

$$P(g,n) = \{g = \sum_{i,j} l,\ l = 1 \text{ where } I(i,j) = g, \text{ and}$$

$$n = \sum_{i,j} m,\ m = 1 \text{ where [distance } (i,j) \text{ to } (k,l) \leq d].[|I(i,j) - I(k,l)| \leq c]$$

One might consider the *R* operator method described below as a measure of how many pixels at a fixed distance *r*, where *r* is the radius of a circular template, exceed a threshold contract *a* above the current *I(i,j)* value. The principal difference is the retention of spatial data as to pixel locus in the *R* method. This allows the retention of spatial shape relationships. It does, however, have implications in the consideration of illumination changes in the source data.

The extraction of explicit features from the source texture is not the only approach. Vickers and Modestino [62] report a 95% accuracy in classifying twelve textures from the Brodatz set. This is based on a maximal likelihood classifier using the co-occurrence matrix directly.

The weakness of the co-occurrence matrix techniques is their inability to capture shape relations in grey level primitives. They do not work well for large area primitive based textures.

Other standard measures of texture include the use of surface vector displacement techniques, as well as Fourier spectra and discrete cosine transform measures. As has discussed (section 1.1), morphological methods have been applied (Matheron [2], Toet [18], Wang et al [19], Peleg [74]). These in general rely on the changes in the image caused by opening and closing to differentiate between texture samples.

The *R* operator was used to generate a histogram of the *R* values for a sample of texture with a particular structuring element, as is described in chapter 2 section 2.5.

The analysis is formed as a histogram, *H*, where, over a sample window size *MxN*∈*f(x,y)*, with probe structure $k(\xi,\eta)$ the individual histogram values are:

$$H = \{h|h_i = \sum_{x,y \in M,N} p,\ p = 1 \text{ where } r(x,y) = i, i \in CARD(k(\xi,\eta))$$

The set of characteristic histograms for a particular sample window, ***H***, is the feature classifier, which, for a set of N probe structures would be defined as:

$$\boldsymbol{H}_N = \{\{H\}_n\}, n = \text{no of the probe structuring element applied, } n \in N$$

Note that although the sample window applied is square, the resulting shape of area analysed depends, additionally, on the shape of the probe structuring elements and does overlap the perimeter of the window area in its analysis. The texture feature vectors are specified as the histograms themselves. Classification is based on minimum distance measure between the histograms. Where multiple feature vectors are used, the distances are aggregated to obtain an overall result. In the practical work of section 5.7, no attempt to use only the significant parts of the vectors, or to apply a principal components analysis has been undertaken. This work is included to show the utility of the *R* operator approach for the general class of grey level structures which form the intensity map.

3.6 Summary

The problems of determining an appropriate strategy for recognition were considered in this chapter. The need for appropriate modelling strategies, and their reliance on extracted features, were outlined, and a justification of a local feature model based on structural features was made. In essence this results from the desire to use a completely morphological algorithm structure: the feature sets obtained are amenable to pairwise matching with a library model; they can be extracted into a set based description; and their sequence has meaning in the context of the object model.

The use of both set erosion and the RJ operator as the agents for the solution of the recognition task was outlined, with some immediate concerns about their practical use. The effectiveness of the extraction mechanism, its reliability and immunity to noise are seen to have a significant bearing on the given solution.

The extraction of rotation, translation and scaling was considered. In the suggested method, these important items of data would be extracted after recognition had occurred, so greatly simplifying the task. The suggested method used the distances between prominent, well-separated features as the medium for the generation of the information.

The utility of silhouette loci of high and low curvature for recognition was introduced, and its use as a subset of a more complete model including in-object features. The ability to generate a silhouette for separate regions defining textured objects in an image (or a textured background) was discussed as a vehicle for recognition under this scenario.

In consideration of texture analysis and classification, we have defined an analysis based on an aggregate signature relating both to the size and to the gradient of the texture in the intensity map as part of the feature vector generator. Conventional measures such as the grey level co-occurrence matrix cannot include the shape of the texture as part of their result.

The adherence to a fully morphological approach will offer a number of benefits in the speed of operation. As discussed in section 1, the operation of the morphological operators is based on isolated areas, at each point in the data set concerned only with the region corresponding to the structuring element. The processing is usually implemented using purely integer arithmetic. The operators are therefore highly suited to hardware implementation. The small physical size of the hardware implementation, estimated at around a thousand gates, offers the ability to replicate many of them on a single semi-custom device. The small physical size and source data isolation are desirable features for a highly parallel solution.

The method used for identifying the closest match between the acquired feature set and a library reference model is very similar to that used to isolate the features. These are computationally expensive parts of the conventional recognition process. The similarity in the algorithms used offers the ability to use the same hardware for their implementations, offering dual benefits of greatly accelerating both processes.

Any conventional processing stages introduced before the extraction of features has occurred will greatly slow the overall speed achieved. The model matching process uses extracted features, and hence far less data, and additional processing before this stage is therefore unlikely to produce the same effects on performance. The interpretation of the model matching results remains a broadly serial task.

Chapter 4: Feature Extraction using Morphological Methods

Chapter 4: Feature Extraction using Morphological Methods

4.1 Introduction

As has been shown earlier (see section 3.3.1.2) silhouette features have been widely regarded as useful local descriptors for recognition purposes, and a variety of methods have been used to identify points of high curvature on the silhouette of an object in an image. The methods employed follow basically two paths (as Davis [32] suggests): either the consideration of points of high curvature and attempting to isolate such points, or the consideration of lines of low curvature and the isolation of points of intersection of such lines.

In-object features are essentially extracted by similar means, and some method such as the Euler number is used to demonstrate the containment of the feature within the object.

Well defined problems exist when feature extraction is considered. The problem of additive noise caused by the image sensors is well known, and low level processing techniques typically based around linear and non-linear filtering have been developed to reduce its effects. Improved sensors have also contributed to the reduction of noise problems in many applications.

Texture is a characteristic of many objects. For anyone attempting to extract a feature set using intensity mapped images, it a particularly intractable problem requiring careful consideration. Its successful characterisation is likely to lead to its inclusion as a feature in future recognition algorithms. Voorhees and Poggio [41] give a practical example of textural segmentation and its possible use for boundary extraction.

This chapter will summarise a selection of the non-morphological local descriptor techniques, describe a number of morphological approaches, and detail the conditional morphological methodology developed in this project.

4.2 Non-Morphological Local Descriptor Extraction Methods: Extracting Curvature of Digital Curves

Points of high curvature on digital curves have been the focus of much consideration. Rosenberg [42] described an approach to identifying the points of high curvature on the silhouette of a convex blob based on the degree of "curvedness" of the domain at a point relative to the chord connecting the end points of the domain. The domains have sizes which are used as significance measures, in that a significant point within the larger domain of another significant point is disregarded. This is an attempt to mimic perceptual reasoning. Rosenberg referred to such points as "dom points".

Rosenfeld and Johnston [43] defined a model for points of high curvature on a digital curve based on smoothed k-cosines. The brief description given below is taken verbatim from Davis [32].

Let the sequence of points$\{(x_i,y_i)\}_{i=1,\text{n}}$ describe a closed curve so that $(x_1,y_1)=(x_n,y_n)$.

Define

$$a_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$$

$$b_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$$

$$c_{ik} = (a_{ik} \cdot b_{ik})/|a_{ik}||b_{ik}|$$

Here c_{ik} is the cosine of the angle between the vectors a_{ik} and b_{ik}, so that $-1<=c_{ik}<=1$, and $c_{ik} = -1$ for a straight line (180°), and +1 for the sharpest angle (0°).

At each point (x_i,y_i), compute $c_{i1},c_{i2},...,c_{im}$ for some fixed m. Assign size h to point (x_i,y_i), and value c_{ih}, for the largest h such that $c_{i,m}< c_{i,m-1}< \ldots <c_{i,h}<c_{i,h-1}$. Finally retain points $(\text{x}_\text{i},\text{y}_\text{i})$ where $c_{j,hj} \geq c_{k,hk}$ for all k such that $|j-k| \leq h_j/2$.

Chien and Aggarwal [44] applied this method to 3-D object recognition. They describe a means of segmenting vertices into concave and convex types using a vector cross product.

A simple curvature measure based on edge following is described by Ballard and Brown [45] based on local curvature estimation through differentiation, corresponding to difference equations on digital curves.

$$|k(s)|^2 = [d^2x/ds^2]^2 + [d^2y/ds^2]^2 \qquad \text{where } s = \text{distance along contour.}$$

Vernon [38] has described the use of the normal contour distance to record a signature from the silhouette of an object. For any point, a_j, on a contour the direction of the tangent to the contour is calculated. A point b_j on the opposite side of the contour is identified such that the line a_jb_j is perpendicular to the tangent to the contour at a_j. The length of the line a_jb_j is the normal contour distance. The value of the normal contour distance for every point on the contour forms the signature.

4.2.1 Contour Description: Chain Code Description of Edges

The use of chain codes for the description of the perimeters of shapes and of lines is widely used. Chain codes consist of line segments that lie on a fixed grid with a fixed set of possible orientations [46]. The introduced loss of accuracy due to digitising continuous straight lines has been described by Rosenfeld [47], and further defined by Dorst and Smeulders [40]. These methods have been used for shape description and character description for recognition applications. Wilson and Batchelor [48] describe a method of defining the convex hull of a chain-coded blob as a means of generating concavity trees for shape description.

4.3 Grey Scale Corner Detectors

The requirement for pre-processing of an image to segment the object of interest within it is, itself, a major area of work. The perimeter following high curvature extraction methods, and the various binary techniques, require the successful completion of such segmentation before they have any possibility of success. Grey scale corner detectors are not dependent on the same level of prior processing (see Kitchen and Rosenfeld [63] for a more complete discussion).

Kitchen and Rosenfeld employ a measure of "cornerness" based on the product of the intensity gradient magnitude and the instantaneous rate of change of gradient direction. This is evaluated at all points in the image. The calculation is based on a quadratic polynomial grey level surface fit.

The facet model-based detector of Zuniga and Haralick [64] finds corners based on two requirements, the presence of an edge and a significant change in the direction of the edge at, or near, the point under test. This uses information extracted from a local polynomial grey level surface fit at each pixel. Its validity is based on the principle of the image as a piecewise continuous grey level intensity surface, and the extraction of facet parameters from it.

Dreschler and Nagel [65] use a Gaussian curvature based corner location technique, formed from a local quadratic polynomial surface fit in the intensity map. Conditions are placed on the Gaussian curvature about the area to determine the presence of a corner.

Some results based on these techniques are shown later in this chapter (p. 113).

Smith [66] describes the use of the SUSAN principle for the detection of corner in grey scale images. This is based on the idea that an approximately univalue segment exists where a solid circular template is placed around the pixel in question, which is the nucleus or point at which the centre of the template is placed. The number of pixels which have a brightness (intensity) approximately equal to the centre point of the template is estimated, and a geometric threshold used to assign the class of corner. In order to reduce false hits, the univalue area about the nucleus is required to be contiguous. Other than the use of a solid circle rather than an annulus, the requirement for contiguity, and the calculation method, conceptually this is the closest method described in the literature to the procedure adopted for detection in this project (see page 100).

4.4 Detection of Edges

The problem of edge detection has been approached in a variety of ways. The obvious method is to look for discontinuities (rapid changes) in the intensity map, which may well delineate a boundary. The differential approaches look for maxima in the first differential of the intensity map (e.g. Sobel operator, Roberts operator [49]), and share the common problem that differentiation amplifies noise in the image. Various approaches have been applied to reduce this effect (Marr and Hildreth [50], Canny [51], Deriche [52]), with some success.

The edge is effectively a shape in the intensity map. The approach adopted by Haralick [53] concentrates on this idea, as do more recent works on robust methods derived from statistical methods for hypothesis testing (Kundu [54], Petrou and Kittler [55]). As is described in sections 4.6.4 and 4.6.5 of this chapter, we have used a shape based method of direct probing, interpreting the results of the R operator applied at various scales of a template as a means of isolating structures on the perimeter of an object. The method adopted the use of various sizes of the template to minimise the effect of noise, intersection of the results providing a reduction of the spreading of the edge caused by larger templates. The closer the gradient of the template becomes to the gradient of the edge (assuming its absolute level does not exceed the edge height), the more localised the determination of the position of it.

4.5 Morphological Methods

Morphological methods have been applied to feature extraction. Serra [3] describes methods for identification of shapes, edge detection, thinning, and identification of specific types of in-object local features using morphological methods. We shall concentrate on local feature extraction. Pitas [16] and Maragos [9] describe global feature extraction methods based on successive applications of increasing sizes of structuring elements. Signatures based on the morphological operations are derived for the images under test.

4.5.1 Hit and Miss Transform

Let B_1 and B_2 be two structuring elements. The hit and miss transform is defined as:

$$A \otimes (B_1, B_2) = (A \Theta B_1) \cap (A^c \Theta B_2)$$

Directly useful for binary situations, this result is not immediately applicable to the greyscale case. Serra defines the Hit and Miss Transform in the greyscale case in terms of umbras ([3], p450):

$$U(f \otimes g) = U[(U(f) \Theta U(g1)) \cap (U(f) \Theta U(g2))]$$

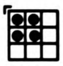
Note that the use of this result is completely dependent on selection of the correct probe functions $g1$ and $g2$.

The hit and miss transform defines the location of structures by the intersection of the erosion of the set under test by one structuring element with the intersection of the erosion of its complement by another. The pair of structuring elements are chosen so that only the loci of structures of interest survive in the result. It is often used to isolate points that have geometric properties, such as corners and border points on shapes, or the significant points such as the locus of a template match.

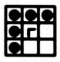
Consider the example below, where we attempt to find the top left hand corner of a square in a binary image.

Example: Hit and Miss Transform Used to Locate Top Left Hand Corner of Shape

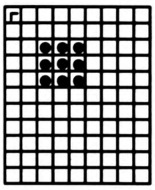
Set B1



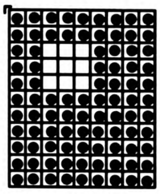
Set B2



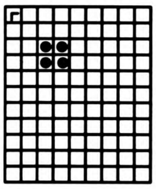
Set A



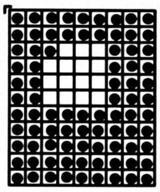
Complement of A



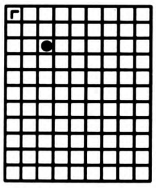
$A\Theta B1$



$A^c\Theta B2$



$A\Theta B1 \cap A^c\Theta B2$



4.5.2 Thinning

This is described quite simply by the following formula [56]:

$$X\mathrm{o}(B_1,B_2) = X\cap[(X\Theta B_1)\cap(X^c\Theta B_2)]^c$$

The thinning of an image produces a thin replica of the input shape, using the hit and miss transform. The hit and miss transform eliminates perimeter details, based on the shape of the structuring element used. One common method of application is based on a series of structuring elements, to produce a symmetrical thinning. The basic structuring element is rotated to obtain the set of structuring elements. Each is applied in turn to thin the image under test until no further changes occur.

Thinning algorithms are used in conjunction with edge detection to produce digital line maps of objects under test. The efficiency and accuracy of the located edge depends on both parts of the location operation.

4.5.3 Rolling Ball Transform

Proposed by Sternberg [57], the rolling ball transform is defined by the difference

$$f - f_g, \text{ where } f_g = (f \ominus \breve{g}) \oplus g$$

The rolling ball transform defines a boundary of a shape as the difference between the original image function f and its opening by a structuring function g. It will tend to retain areas of high rate of change of intensity while lowering the overall grey level of the image.

4.5.4 Skeletonisation Methods

Various skeletonisation methods are used for shape and object description (see Blum [8] and Maragos[9]). It should be noted that these methods offer a means of extracting structural features. As an example, the medial axis transform picks out significant concave features on the perimeter of the view of the object, but does not necessarily locate them to the precision of a single pixel locality.

4.5.5 Practical Examples

There are a variety of examples of the use of morphological methods to extract features. This section illustrates several of the major approaches and relevant applications. Serra [3] describes the use of morphological filters to extract the summits and sinks of a digitised function (M_i and M_i^* respectively) as:

$$M_i(f) = X_i(f) \,/\, [X_{i+1}(f) \oplus \{H\}; X_i(f)] \quad -\infty < i < +\infty$$

and

$$M_i^*(f) = X_i^{\,c}(f) \,/\, [X_{i-1}^{c}(f) \oplus \{H\}; X_i^{c}(f)] \quad -\infty < i < +\infty$$

where:

$X \oplus H;X = (X \oplus H) \cap X$,

H = compact binary filled hexagon in Golay alphabet [74] for sequential analysis.

D P Casasent and R Schaefer [58] describe the use of morphological filters to extract significant features in simulated infra-red (IR) images. The method applies a combination of morphological binary hit and miss transforms to four derived thresholded versions of the same input image. The thresholds chosen are $T_{1\text{-}4}$ = u +/- e and u+/- 0.3e, where u and e are the mean and standard deviation of a predetermined region of the image. The four binary HMT results are UNIONed (ORed) to provide the result. The output is compared with the equivalent result generated using a gray-scale HMT, reaching the conclusion that the non-linear intersection implied in the intersected binary HMT results is more useful for feature extraction.

4.6 Structural Feature Extraction using the R Operator

The R greyscale operator, as indicated in equations (16) and (17), has obvious applications to directly extracting structural features from greyscale images. The methods developed must meet several criteria. In order to ensure correct identification, a wide range of features should be locatable; to minimise the work required of the recognition stage, the features should be appropriate to the identification task; aliasing problems caused by normal noise in the image, usually in the form of small spurious perimeter features, should be minimised; to allow the highest quality estimation of rotation and scaling, features should be located as accurately as possible; aliasing caused by textural differences should be minimised. These constraints have guided the choice of structural features for the recognition process.

Now consider the problems of extracting structural features from an intensity mapped, quantised image. To be capable of extraction an object must have some difference in intensity from the background. The case of textural differences with the same average intensity is noted, but at the edges where the two textures meet there tend to be local changes in directional average intensities.

4.6.1 Low Textured Objects, Directional Edges

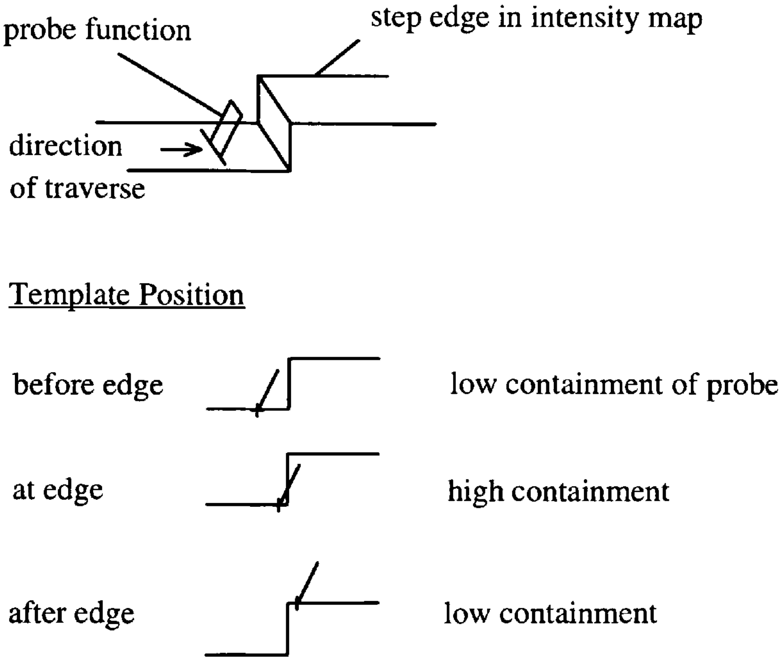
Intensity differences (contrast) between the object and the background will affect the accuracy of location of features. Consider the diagram of figure (23) below. Here a single pixel step change is being used to identify the location of x-directed edges in the image under test. The presence of the edge is assessed by estimation of how many pixels of the probe function are contained within the object function, at each point in the image. As can be clearly seen, the closer the step intensity gradient change is to the actual gradient of the edge of the object, the more localised the result will become. Noise immunity will, however, reduce.

Figure (23): X-directed Edge Extraction using a Simple Structuring Element

(a) Binary Images

0 1
0 1
0 1

(b) Grey Scale



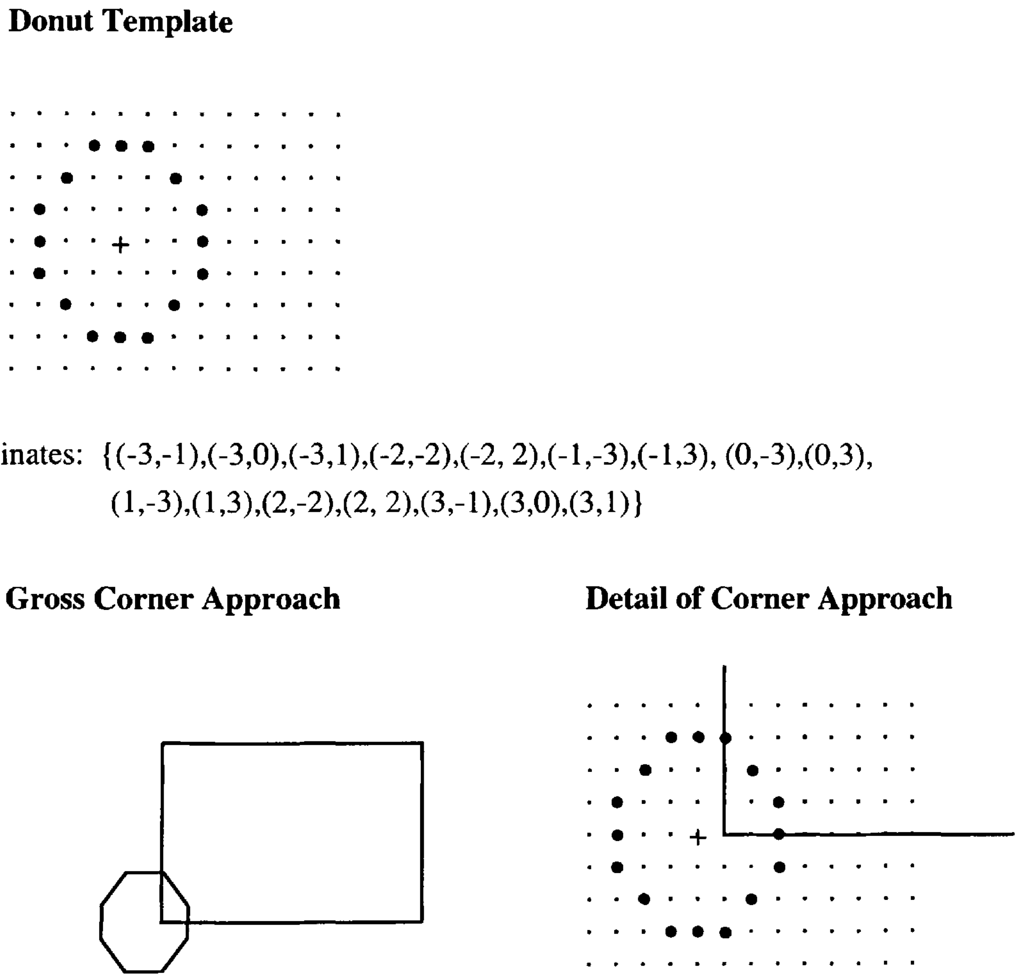
Features can, however, occur at any orientation to the sampling grid.

4.6.2 Principle of Feature Detection

The basic principle is shown below. Assume the structure to be analysed is a two dimensional bright square on a dark background. Both bright and dark areas are smooth (i.e. univalue in intensity). A probe structuring element, or template, is placed at each point within the image under test. Outside the boundaries of an object, if the probe has zero intensity values, it will be fully contained in the surface of the background. It the probe points have an intensity value, none of the probe pixels will be contained in the surface. Both these statements are true for the situation where the pixel at which the probe template is placed cause the template to be fully within the boundaries of the object.

About the perimeter an object, partial containment of a template will occur provided that the intensity gradient of the feature is greater than that of the template. The degree of containment might be used as an indicator of a particular structure on the perimeter of the shape. Consider the diagram of figure (24) below:

Figure 24: Principle of Feature Detection

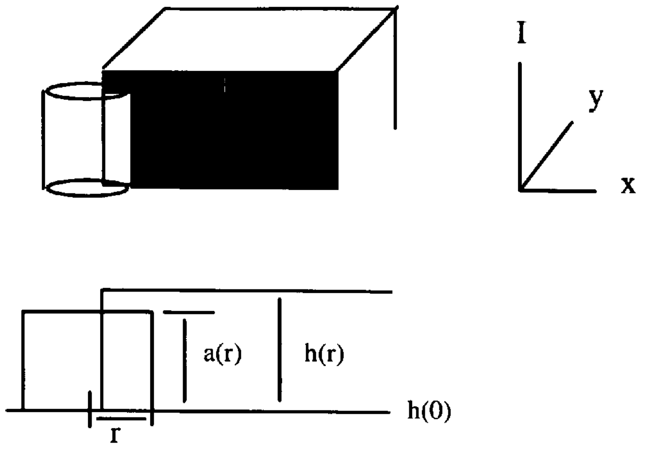


Donut Template

Co-ordinates: {(-3,-1),(-3,0),(-3,1),(-2,-2),(-2, 2),(-1,-3),(-1,3), (0,-3),(0,3),
(1,-3),(1,3),(2,-2),(2, 2),(3,-1),(3,0),(3,1)}

Gross Corner Approach

Detail of Corner Approach



Pixel at radius r is contained if h(r)>a(r)

Let us assume that a condition of three or four elements of the template contained is indicative of a particular class of convex corner. Certain information is clear:

(i) multiple responses occur at a corner;

(ii) equivalent hits may result from approaching a straight or curved edge for certain classes of corner;

(iii) these multiple responses must be resolved, either by clustering or some other means of assessing a good hit;

(iv) equivalent containment could appear due to noise.

Aside

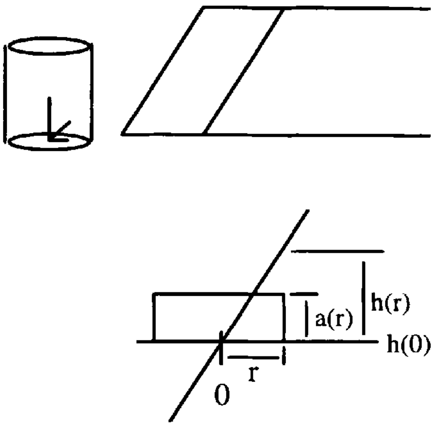
A much larger, dense template could be used, with all pixels utilised. The requirement that the nucleus of the template be contained is included, and the area of a "hit" judged on the number of contained pixels as an area. The contiguity of the "miss" area could be assessed by a conditional erosion of it by a template of appropriate size. Such an approach would lead to an implementation of a variant of the SUSAN approach (Smith et al [66]), which has been demonstrated to have good noise and isolation characteristics. This would, however, imply a considerable additional processing load, and consequent cost in speed.

4.6.2.1 Template Intensity Gradient

The choice of intensity threshold by selection of probe structure profile, and therefore detection gradient, is an important parameter in the localisation of features. With a high contrast between object and background, low noise, and well-defined internal features, the gradient chosen should be of the order of 90% to 95% of the minimum gradient of a required feature. Based on the conventional assumption that the best locus of an edge or feature is the locus of highest rate of change of intensity or gradient (which the differential methods tend to use), this would offer better localisation and probable accuracy of the feature locus. In the presence of noise, blurring, or lower contrast, selection is less obvious.

4.6.3 Noise Analysis

Consider, as a probe structuring function, the donut annulus of intensity height *a* at its radius, and its approach to an edge of a feature, height *H*.



As the annulus reaches the edge of the feature, the noise height, *n*, required to give a false value at a particular point on the radius *r* in $(h@a)(0)$, must carry the annulus height at radius *r* above the feature surface at *r*.

i.e. $n \geq (h(r) - (h(0) + a(r))$

Probability of a change in value $= pr(n \geq (h(r) - (h(0) + a(r))).pr(n \textit{ occurs at } r)$ evaluated over the radius of the annulus. This is, however, only part of the problem. Noise at the origin of the annulus will have an equivalent effect. Either event will cause a change in the *R* value for this position. The events are drawn separately from a random distribution, and can be treated as independent events. Thus

$$\text{Probability of a change in value } = pr(\textit{change occurs at } r) + pr(\textit{change occurs at } 0) - pr(\textit{change occurs at } r).pr(\textit{change occurs at } 0)$$

It can be seen that this is a symmetrical - either positive noise or negative noise will produce the effect.

The number of pixel changes required to achieve a false measurement depends on the geometric thresholds chosen for the areas corresponding to concave, convex and low curvature feature regions. On this basis, the probability of a false result is equivalent to the probability that *m* pixels change their state as far as the R analysis is concerned.

$$\text{Probability of a false result } = [pr(\textit{pixel change at } r)]^m + pr(\textit{pixel change at } 0) \\ - [pr(\textit{pixel change at } r)]^m . pr(\textit{pixel change at } 0)$$

With the annulus template used, and subject to the broad feature type chosen as appropriate for the recognition algorithm, the features produce multiple responses. As a consequence, it is necessary for noise to corrupt the entire response over the feature for it to be removed. The probability of a missed feature is equal to the probability of a false result over the area of grouping corresponding to the feature, say *x* pixels, and

$$\text{Probability of missed feature } = \begin{pmatrix} [pr(\textit{pixel change at } r)]^m + pr(\textit{pixel change at } 0) \\ -[pr(\textit{pixel change at } r)]^m . pr(\textit{pixel change at } 0) \end{pmatrix}^x$$

Assume the geometric boundary limits are set such that only one pixel is required to change to change the state of the response. Then

$$pr(\textit{missed feature}) = \begin{pmatrix} pr(\textit{pixel change at } r) + pr(\textit{pixel change at } 0) \\ - pr(\textit{pixel change at } r) . pr(\textit{pixel change at } 0) \end{pmatrix}^x$$

which, for a random noise distribution becomes simply

$$pr(\textit{missed feature}) = \left(2 \times pr(\textit{pixel change at } r) - pr(\textit{pixel change at } r)^2\right)^x$$

This offers some simple conclusions:

(i) the greater the noise level, the higher the probability of failure;
(ii) the larger the template, the more likely the recovery of features;
(iii) if more than fifty percent of the image area is corrupted with significant noise, the larger the template is, the more likely missed features become;

(iv) the greater the difference between the gradient of the feature in the intensity map and the probe function, the higher the noise immunity;

(v) the noise immunity is governed by the number of pixel changes needed to change state - the use of more pixels and a greater geometric tolerance separation will improve noise immunity;

(vi) the greater the image contrast, the less effect noise will produce

(vii) to some extent, a trade off between localisation and feature accuracy of description against noise performance is possible.

4.6.3.1 Salt and Pepper Noise

The distribution of salt and pepper noise follows the following rules:

Value at *(x,y)* = *N(x,y)* with probability *p*;
Value at *(x,y)* = *f(x,y)* with probability 1-*p*

where *N(x,y)* is the noise distribution.

If the noise is full range (i.e. *N(x,y)* = either maximum or minimum of image range, the presence of noise will guarantee $n \geq (h(r) - (h(0) + a(r))$. As has been discussed above, the probability of a missed feature now becomes:

$$\begin{aligned} pr(missed feature) &= \left(2 \times pr(pixel\ change\ at\ r) - pr(pixel\ change\ at\ r)^2\right)^x \\ &= (2p - p^2)^x \end{aligned}$$

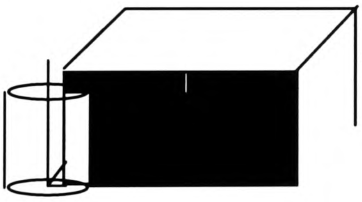
It is usual to use regard *p* as a figure for the noise percentage in the image. The number of pixels corresponding to a feature is crucial to the resulting value. Some practical use of containment estimation in the two forms described in the following sections 4.6.4 and 4.6.5 with the donut template (see appendix A.3) in the presence of salt and pepper noise is shown in Table 6, p112. Due to the rotation of the shapes in this image, and the consequent effect of alignment with the sampling grid (image grid), the positive response to a feature rests in some cases on a single pixel. Given that the feature must be detected by three different templates to survive in the result, as the practical evidence shows, the likelihood of features being missed increases rapidly over the 20% noise mark.

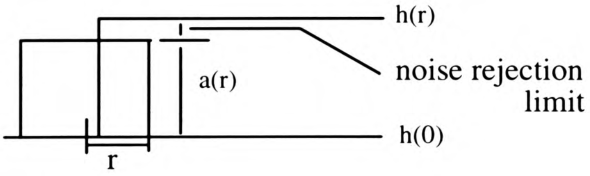
4.6.3.2 Gaussian Noise

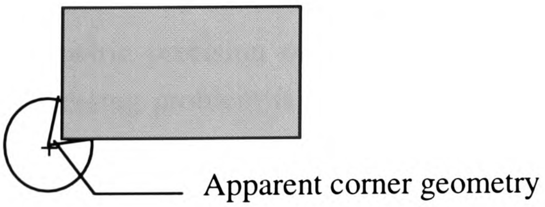
Gaussian noise is characteristically random, but occurs symmetrically about its mean value. Noise addition algorithms add noise at all pixels within the image, the noise level being drawn from a random number system based on a Gaussian characteristic. In practise, then, the concern here is whether the level of noise (additive, or multiplicative) is sufficient to cause a change in the result at a pixel locus in the probe template. The probability of a Gaussian variable lying outside a particular value range is well understood, and forms the basis of the use of the Gaussian distribution in statistical work. The standard deviation of the noise is required, the noise level often being stated in terms of numbers of standard deviations.

4.6.3.3 Template Gradient in Relation to Noise

Structures of interest in the intensity map are often in the form of dark/light and light/dark transitions. Consider the example below, of a bright rectangle on a dark background. The annulus template, with radius r and intensity a(r) is used as a probe structuring element. The section shown is a slice along the perimeter of the rectangle.







Flat Annulus, a(r)=0

A flat template will produce optimal rejection at the perimeter of bright structures - but logic dictates it produces the worst possible results over the background, or regions of uniform intensity in the presence of noise. Its sensitivity will allow interpretation of any noise as a relevant structure (as estimated by pixèl containment count in the probe set). The measure of r immunity to change in pixel value over this type of region is given by:

$$\begin{aligned} diff &= h(r) - [h(0) + a(r)] \\ &= -a(r) \end{aligned}$$

If $a(r) = 0$, then sensitivity to noise is a maximum over the areas of the image not abutting an object boundary or dark/light transition. Noise immunity is, however, at its highest over the high gradient areas.

One of the design goals is the selection of an appropriate gradient to just exceed the image noise threshold. However, if the noise occurrence in the image is low compared to the template pixel count, or of low likelihood over the template, then the zero value (i.e. flat) template is chosen. Its 2-D shape, or footprint, then defines the sole contribution to the geometric probe.

Geometric Sensitivity

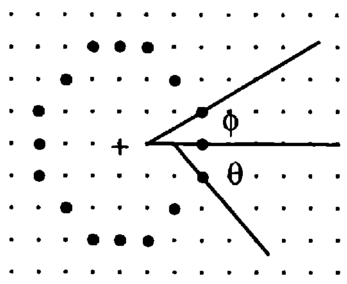
Assuming a broadly circular footprint for the probe structuring element, the geometric threshold of resolution is given below:

$$\begin{aligned} &Geometric\ threshold = \frac{1}{n}\ rads for\ a\ circular template \\ &where\ n = no\ of\ (equally\ spaced)\ probe\ elements \end{aligned}$$

Figure (25) below illustrates the situation where two elements of the probe are required to be contained to indicate a particular shape of corner. As can be seen, the geometric precision of the corner detector is not particularly tightly toleranced. The processing problem is more likely to be one of finding the general classes of perimeter structure (high curvature concave, convex, low curvature, etc.) than of the isolation of a particular single special curvature feature. The solution of this less common problem simply requires a different probe structuring function.

As the template approaches a boundary, several hits are likely to occur. This may result in the inclusion of straight boundary sections in high curvature estimates, and also the generation of hit clusters (depending on the laxity of the condition). Further work will then be required to isolate the required points. The reduction of clusters will increase the degree of uncertainty in the precision of estimation of corner position.

Figure (25): Corner Geometry Resolution with the Donut Template



The geometry of the situation indicates a variation of angle of between θ and ϕ for any detected sharp corner

The requirement for identification of a possible feature locus is two-fold: the shape of the feature in the intensity space must match the geometric requirements for membership of the feature group, and the intensity profile must exceed the profile of the applied probe.

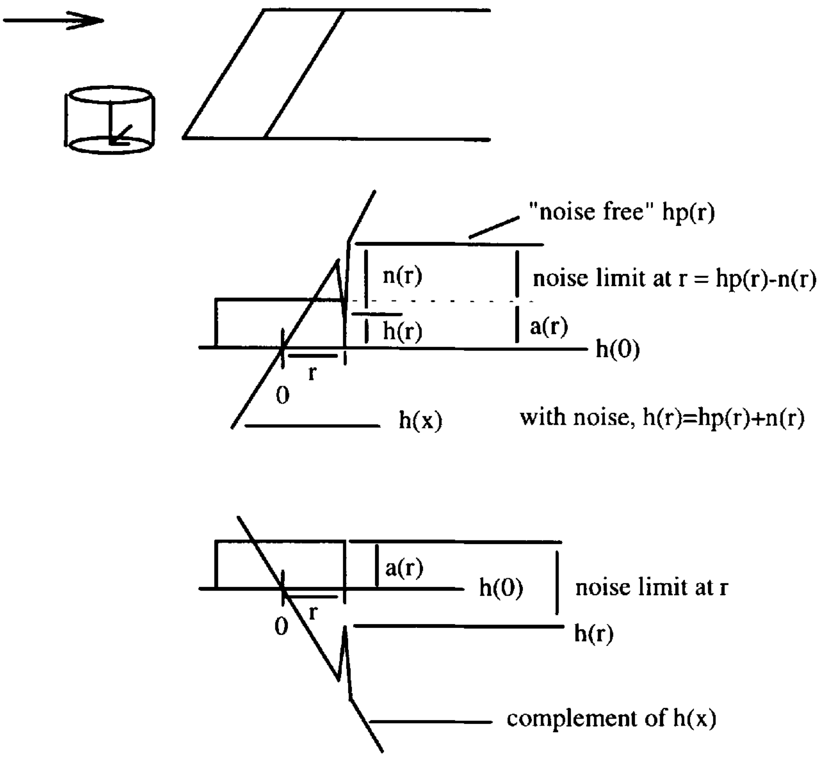
4.6.4 Single Template Analysis - Mostly Hit, Mostly Miss Transform

Consider the approach of an annulus template, height at its radius *a(r)* approaching the perimeter of an intensity map structure or object in an image, h(x). Assume a noise component occurs at the radius point shown below in figure (26), where the origin of the probe template is at some point h(0), on the surface of the function on the rising edge of the object intensity map.

The actual image function, h(x), can be regarded as the sum of two images, the perfect (noise free) image function hp(x), and the noise image n(x). The actual image is then h(x)=hp(x)+n(x).

If the complement of the image function is taken, the same probe structure used, and appropriate conditions applied, it can be seen in figure (26) that the effect of the noise component does not produce a similar false hit (miss) in the complemented image. This is subject to selection of appropriate conditions on the "hit" and the "miss" structuring elements. It is reasonable to assume that using the complement of the conditions will produce the required result - but incorrect. If any offset of the apparent position of the feature is present, the hit and miss approach will miss the feature under these conditions, unless noise is present in sufficient quantities to produce a false hit in the region of the complementary acceptance.

Figure (26): R Operation on a Noisy Corner Function and its Complement



Conclusion

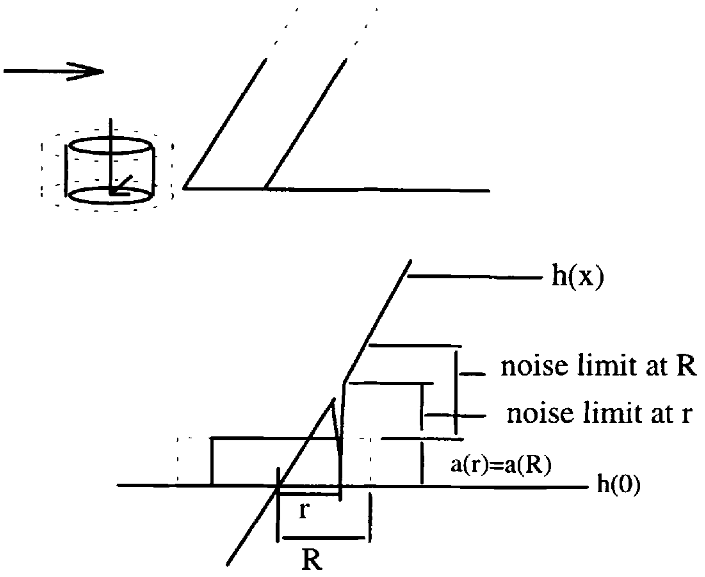
A "hit and miss" strategy will allow the elimination of false hits, but fails to remove false misses. It will tend to over limit the data contained in the result. To compensate for this, a wider geometric tolerance might be adopted. It will achieve the removal of false inclusion as the probes approach the perimeter of the object in the image.

The level of spurious results, mostly missed features, is likely to be in excess of those generated by direct application of the conditioned R operator for equivalent conditions.

4.6.5 Multiple Template Intersections

The larger the applied template, the larger the area of response to it. The noise rejected responses of larger templates can be used to reduce the apparent area of response. In effect, by using a looser geometric tolerance, coupled with several equivalent templates of similar shapes but different sizes the area of clusters produced can be minimised, and the false hits approaching straight edges removed. This is illustrated for a simple noise condition in figure (27) below.

Figure(27): Multiple Probe Templates Approaching a Corner



The effect on accuracy of location of features is generally beneficial. Based on a reasonable selection of probe structure gradient, the maximum noise rejection lies in the area of peak rate of change of gradient; therefore the most likely response lies in the area of best feature location (or, depending on the type of probe and feature assessed, at a determined distance from it).

The removal of false side hit inclusion is illustrated overleaf in figure (28).

As has been stated in equation (15) p. 57, this is usually used with multiple probe templates, often three, which can be increasing sizes of the same basic shape and gradient probe.

Inference Rules

Basic rules of containment apply. If we start with the donut template and produce symmetrical dilations of it, perimeter features are inferred using conditions as shown in Table (5) overleaf. The selection of limits is obvious: if more than 60% of pixels in the template are contained, the point under test is likely to be close to a concave perimeter structure; if less than 40% of pixels in the template are contained, the pixel under test is

Figure (28): False Hit Rejection with Multiple Templates

- At corner, containment = 3 and 3 (small and large probe)
- At side, containment = 3 (small) and 5 (large probe)

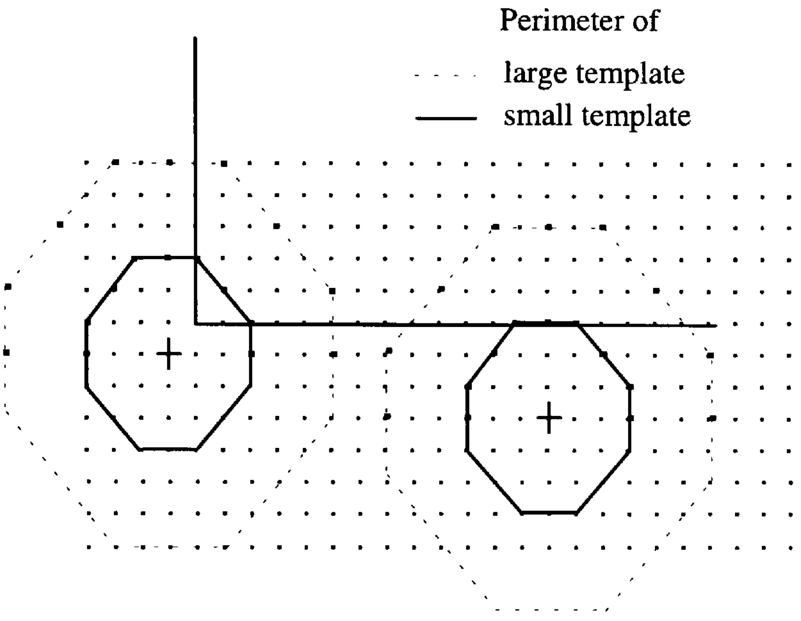


Table (5): Containment Thresholds for Determining Perimeter Curvature with the R Operator

CURVATURE	R VALUE LIMITS	% EQUIVALENTS
High, Convex	(14,11)*	88 - 69
Low, Convex	(10,9)	63 - 56
Low	(9,7)	56 - 44
High Concave	(4,1)	25 - 7
Low Concave	(7,5)	44 - 31
General, Convex	(13,10),or (14,11)	
General, Concave	(6,1)	
General, Low	(9,7)	

*Containment values where more than 14 out of 16 pixels miss the template are not practical due to noise effects - a single pixel noise element will trigger a response in the background region.

probably close to a convex feature. The problem of noise is significant where low containment values are used to infer the presence of a feature.

The data planes are generated, each corresponding to a particular size of the probe template. The presence of a particular type of feature, at a particular scale, is inferred from meeting the given probe containment requirements across adjacent sequences of planes. A single plane is not sufficient to meet the inference limits, as figure (27) earlier showed. The presence of noise and textural detail makes this inevitable, and it is the averaging effect of multiple sizes that offers the possibility of inference. A minimum of three consecutive planes were used at any one level.

4.7 Feature Detection Testing

4.7.1 Corner Detection

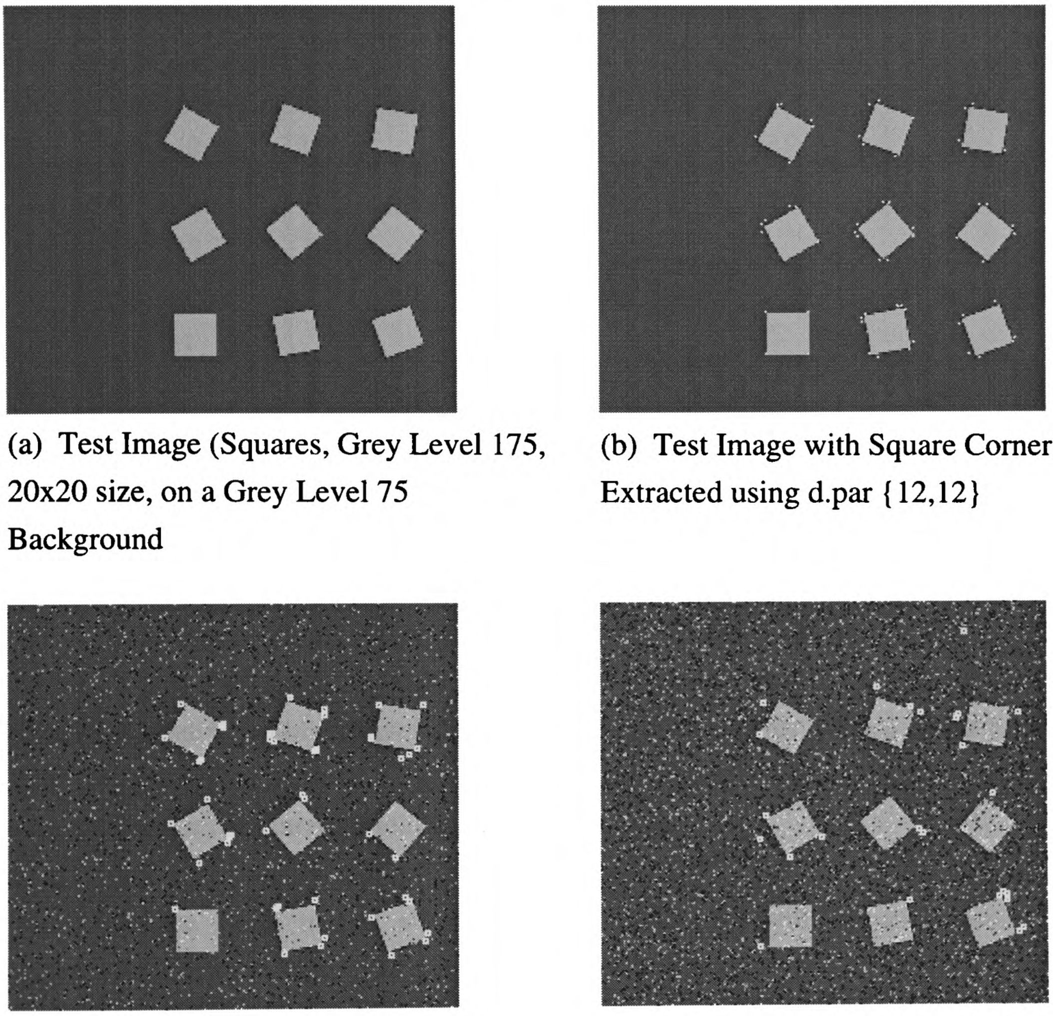
A standard image was used for the tests, as suggested by Haralick et al [64]. Nine squares of brightness one hundred and seventy five grey levels and size twenty by twenty pixels are placed on a background of brightness seventy five grey levels. The squares are shifted in orientation by ten degree increments between zero and eighty degrees.

Square corners, as can be seen from figure (24) earlier, can be found as corresponding to a containment of four pixels for the donut template used. Note that this should lead to a double hit (i.e. two loci for each corner), each of which is offset from the corner by one pixel for the zero orientation case. This does presuppose the alignment of the square corner and the sampling grid (pixel grid).

The donut set of three templates d.par (see Appendix A.3) was applied with the limits set for a strict containment of four elements only. The results are shown in figure [29]. Appendix A.4.1 shows the actual corner loci, along with some explanation as to missed features. The average error in corner position was found to be 1.6 pixels from the actual locus for the clean image.

In addition, the effect of added random noise was investigated. Salt and pepper noise contamination of ten, twenty and thirty percent of pixels was added. The number of missed corners and false hits increased with the noise level, as did the degree of multiple responses for features. Choosing the intensity threshold so as to eliminate the majority of the noise, rather than relying on a "best nominal" improved the results. No pre-processing was applied, and no attempt made to reduce the effect of the noise prior to analysis. The results obtained are illustrated in Table (6) overleaf.

Figure (29): Actual and Extracted Corner Positions with Test Image, Salt and Pepper Noise Added to the Percentage Levels Indicated



(a) Test Image (Squares, Grey Level 175, 20x20 size, on a Grey Level 75 Background

(b) Test Image with Square Corners Extracted using d.par {12,12}

(c) Test Image, 10% Added Salt and Pepper Noise, Square Corners using d.par{12,12}

(d) Test Image, 20% Added Salt and Pepper Noise, Square Corners using d.par{12,12}

The average error in corner detection is of the order of 1.6 pixels value - this is slightly more than one pixel error (the calculation of errors in the original would regard one pixel across, one down as a position error of 1.414 pixels, not 1). This increases with added noise. It must be noted that multiple hits do occur at some corners. This, again, is a predictable feature of the use of the intersected R analysis with clustering. This data includes no attempt to loosen the geometric threshold to permit greater clustering at the corners. The results are summarised in Table(6) below:

Table (6): Corner Detection in Salt and Pepper Noise

Noise %	Templates	% Corners at actual locus		% Corners at +/- 1 pixel from locus		% Corners at +/- 2 pixels from locus		% Corners at +/- >=3 pixels from locus		% Missed =N/36	
		No.	%	No.	%	No.	%	No.	%	No.	%
0	d,par{12:12}	1	1.7	22	37.9	20	34.5	15	25.9	2	5.5
10	d,par{12:12}	0		14	28.0	22	44.0	14	28.0	5	13.9
20*	d,par{12:12}	1	3.8	2	7.7	5	19.2	18	69.2	21	58.3
30	d,par{12:12}	failed†									
10	d50.par {12:12}	1	2.2	16	35.6	16	35.6	12	26.7	6	16.7
20	d50.par {12:12}	0	0.0	9	25.0	10	27.7	17	47.2	10	27.8

*Plus 11 false hits †Too many responses to adequately classify

Gaussian noise of standard deviation ten grey levels was then added to the original image. This corresponds to the test image used by Haralick [67] for the Kitchen-Rosenfeld and Dreschler-Nagel corner detectors, along with Zuniga and Haralick's best facet model-based detector. Smith [68] applied the SUSAN method and the Plessey corner detector (or Harris detector) to the same data.

On the given image, shown here as figure (30) with the identified corner loci added as white dots, the error represents a significant deviation from the corner position when considered in terms of the apparent percentage length change in the side separating the corners. However, further trials have shown that the same degree of absolute error in pixel position occurs on larger sizes of the shape. The sample used was a square of 300 pixels per side with the same object and background intensity levels and added noise. This constancy of error is useful, inferring as it does that the error in relative terms decreases geometrically with the increasing size of the object, or with decreasing size of sampling grid.

The algorithm has an in-built error of one pixel from the locus of the square corner. A table of results for a variety of detectors is shown in Table [7] below.

Table[7]: Corner Detection Results for Various Algorithms

	d=0		d=1		d=2	
	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
Multiple Intersection						
d50.par {9:9,12:12, 12:12}	0.083	0.077	0.72	0.74	0.83	0.92
d50hug.par {12:12}all	0.083	0.052	0.72	0.45	0.97	0.86
Mostly Hit, Mostly Miss						
Rjhma5 with donut20.dat {6:1,6:1}	0.167	0.12	0.61	0.59	0.72	0.76
Other Grey Scale Methods						
Facet* model detector	0.361	0.361	0.97	0.97		
Kitchen-Rosenfeld* detector, no gradient threshold	0.055	0.021	0.36	0.36		
Kitchen-Rosenfeld* detector, gradient threshold=20	0.055	0.05	0.83	0.84		
Dreschler-Nagel* detector, gradient threshold = 20	0.055	0.059	0.33	0.35		

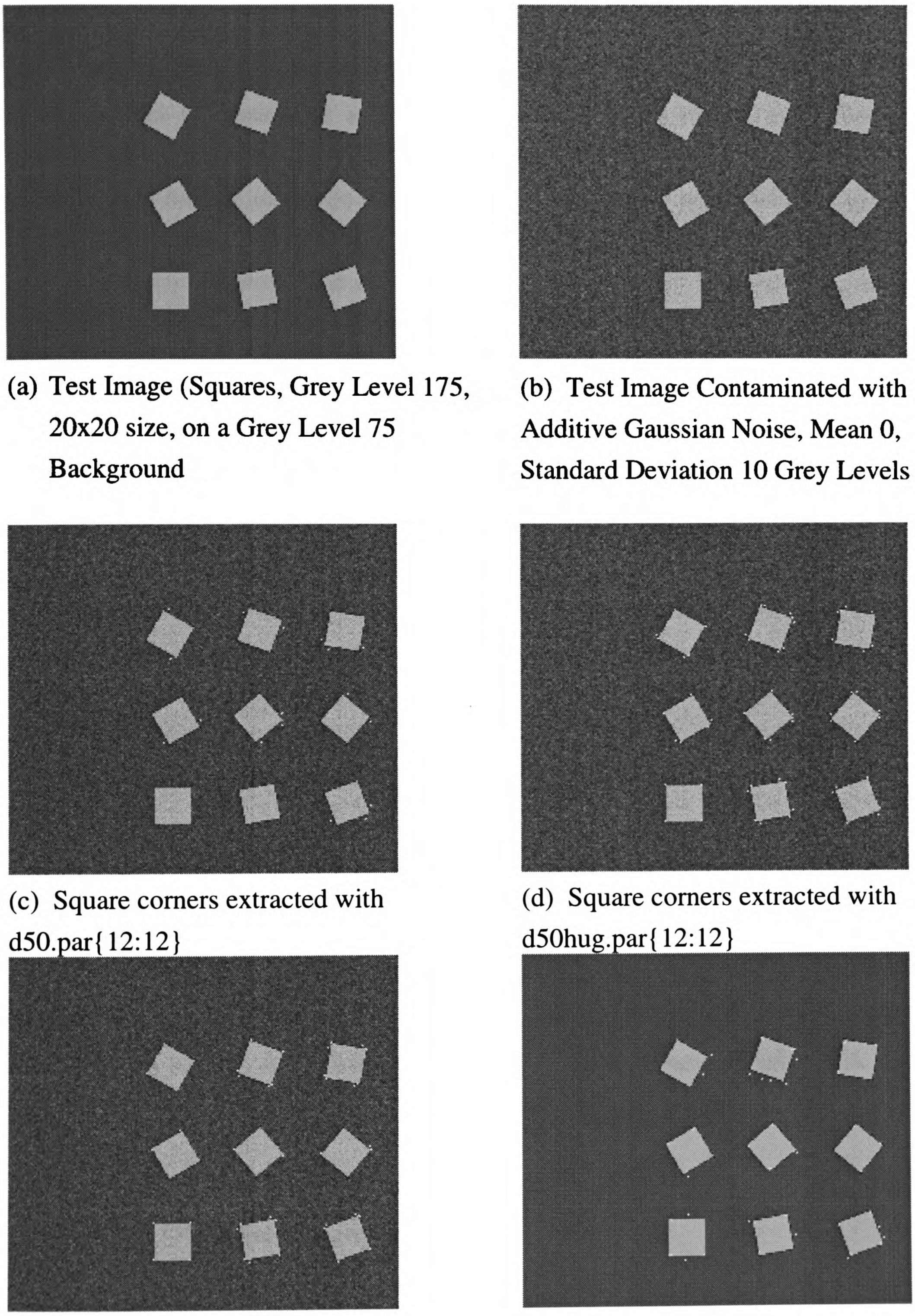
d = distance between actual and assigned corner

P(AC|TC) = pr(assign corner and corner exists within specified distance)=(corners within distance/no of corners); note this does not include multiple hits.

P(TC|AC) = pr(true corner within specified distance if corner assigned)=corners within distance/no of corners assigned); this does include multiple hits on the same corner

*These results are taken from "Computer and Robot Vision, Volume 1", R M Haralick, L G Shapiro, Addison Wesley, 1992, pp 418.

Figure (30): Corner Detection in Noise



(a) Test Image (Squares, Grey Level 175, 20x20 size, on a Grey Level 75 Background

(b) Test Image Contaminated with Additive Gaussian Noise, Mean 0, Standard Deviation 10 Grey Levels

(c) Square corners extracted with d50.par{12:12}

(d) Square corners extracted with d50hug.par{12:12}

(e) Square corners extracted with d50.par{9:9,12:12,12:12}

(f) Square corners extracted by MHMMT with donut20.dat, {6:1,6:1}

Discussion

Whilst this undoubtedly offers a valid comparison of the capability of identifying the exact loci of corners, it is less than generous in its treatment of the conditional morphological methods described. For these algorithms, the correct identification of pixel locus is outside of the boundary of the object, by one pixel, with a high probability of a dual response. Nevertheless, it does illustrate the facility of the methods applied in detection of corner loci. The best response came from the intersected *R* sets approach with three sizes of template (d50hug.par as defined in AppendixA.3), with a relatively high intensity threshold of 50 grey levels. Given that the mostly hit, mostly miss transform results are, in effect, a form of profile fitting, the intensity threshold of 20 was chosen to give closer comparison with the other grey level detectors used, all of which rely on some form of surface fitting within the models they adopt.

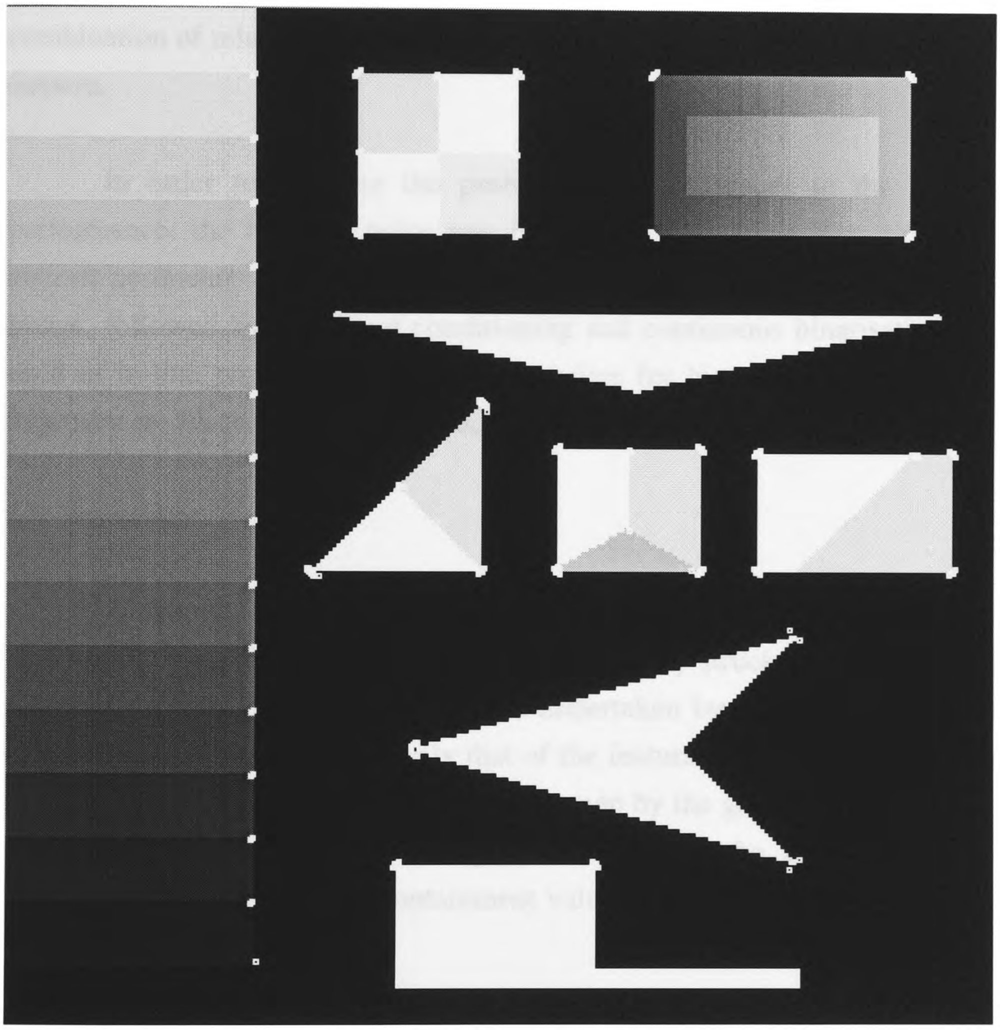
The results are favourably accurate in comparison with the Kitchen-Rosenfeld and the Dreschler-Nagel detectors, but not as accurate as those achieved with the best performance of the Zuniga and Haralick facet model detector. When considered in the context of the expected response, the figures rival, but do not equal, those of the best facet detector (expected response is 1 pixel from the corner, corresponding to d = 2 from the corner for a +/- 1 pixel distance from the expected locus).

The qualitative data given in Smith [68] would indicate a superior performance for both the SUSAN and Plessey detectors. Both these systems use contiguity constraints to limit the response windows. In this case, this would correspond to the assertion that not only the geometric threshold is exceeded as measured by pixel count, but that the block of pixels causing the response form a contiguous area within the image. As was mentioned in section 4.6.2 earlier, the SUSAN detector uses a solid disk template to identify features, offering better spatial resolution of geometry at a cost of more processing.

4.7.2 Other Geometric Features

For a wider variety of corner shapes, the synthetic test image of figure (31) was used. This was generated as a feature test image, generated by doubling the size (256x256 to 512x512) of an image obtained from Mr S Smith, Department of Clinical Neurology, Oxford University via the robotics research group web site www.robots.ox.ac.uk. It contains useful characteristics, including a variety of abutting intensities and profiles. Its generation is content is formally discussed in [66]. The features illustrated in figure [31] below were extracted from it using the conditions indicated. Results obtained for the extraction of right corners in both the clean image and the same image with 10% salt and pepper noise added are shown in Appendix (5).

Figure(31): Extraction of Various Geometric Corner Shapes



Convex Features Extracted using *R* intersected operator as
result = (I(rin, d.par{12:12})∩ I(rin, d.par{13:13})∩ I(rin, d.par{14:14}))

The broadened threshold for extraction of a variety of feature types inevitably leads to multiple responses at the corners. This can be reduced by clustering in the results plane. The picture above shows a true response of the detector used.

A discussion of the extraction of features from real images is contained in the following chapter.

4.8 Blurring Noise

In the context of the actual system noise for the optical missile guidance problem, the high speed of data acquisition required will necessarily limit the motion smearing of the target object image. However, the motion of the missile itself, and the levels of high frequency vibration present due to engine and flight effects, will produce blurring in the acquired data. The blurring produced is unlikely to be unidirectional, being caused by a combination of relatively random effects acting on the support of the system acquisition camera.

In order to evaluate the problems this introduces to the recognition system performance, the blurring noise was introduced as volume distortion of the binarised aircraft perimeter. This is an attempt to simulate the effects of random blurring on the image, followed by the signal conditioning and continuous binarisation which are often used in in-line processing systems appropriate for high speed use. The results of this approach are discussed in Chapter 5, section 5.6

4.9 Summary

A feature detector for identifying structures in the intensity map has been developed, using the R operator to evaluate intensity structure geometry as the parameter for classification. This classification is undertaken by creating a probe structure of the required geometry (not necessarily that of the feature sought) and evaluating its presence at each point of the surface of the intensity map by the generation of a containment value indicating how many of the components of the probe are contained at any point. Inference is drawn from the containment value as to the structure of the intensity map at that point.

Two methods were developed to limit the likelihood of false inference being drawn due to noise and structure proximity, based on the intersected R method and the mostly hit, mostly miss transform. The former uses multiple templates to identify the features; the latter relies on a closer definition of the surface profile achieved by evaluating both a normal and a complemented image about each point.

These methods were evaluated against standard techniques for the isolation of particular feature types using standard images. The results show a relatively good performance in comparison with the Kitchen-Rosenfeld detector and the Dreschler-Nagel detector, but being bettered in terms of feature location by the facet model detector of

Zuniga and Haralick, and apparently in completeness of discovery by the Smith's SUSAN detector. Corners were reliably detected in noise conditions of up to 20% salt and pepper noise within 3 pixels of their actual loci. The *R* intersected algorithm found features more reliably, the MHMMT more accurately for the given data.

Chapter 5: Feature Extraction Results from Real Images

Chapter 5: Feature Extraction Results from Real Images

5.1 Introduction

In acquiring results, the problems of general purpose practical systems become apparent. Whilst no difficulty occurred in generating high quality examples through operator intervention, it proved considerably more difficult to produce generic, automatic algorithms capable of operating over a variety of lighting and object background situations. It is no coincidence that the majority of industrial imaging applications require carefully controlled lighting and known orientation and scale. As has already been stated, the reduction of degrees of freedom, and the elimination of unwanted reflections and shadows, allows the imaging system to work within the tight constraints necessary for reducing processing effort. The real difficulty in handling general data comes from its variability.

5.1.1 Equipment

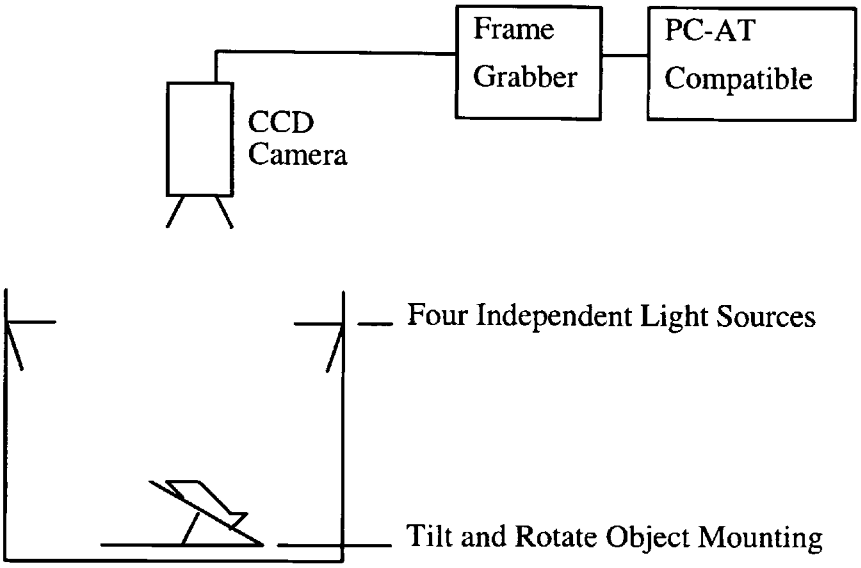
A relatively modest equipment base was used for this project. For image acquisition an Imaging Technologies PC Vision Plus frame grabber was used with a Pulnix CCD camera. For later images a Truevision Targa+ frame grabber was used. Both yielded 512x512, 8 bit deep grey level images. For illumination three angle-poise lamps were used, offering a limited control over intensity and direction of illumination. Such an approach was regarded as adequate given the aim of the project to allow direct feature extraction from loosely controlled environment images.

Control over orientation of objects under test required the use of a fixed camera position relative to the object platform. The object platform allowed limited roll and pitch positioning, and full yaw rotation.

A block diagram of the rig is shown in figure (32) below:

All algorithms were written in the C programming language, and variously compiled using Microsoft C version 5.1, Borland C++ version 3.1, and Microsoft Visual C version 2, initially on a Viglen 286-16 PC AT compatible under MS-DOS and later on a Viglen 486DX66 PC AT compatible under MS-Windows.

Figure (32): Image Acquisition Rig


Frame
Grabber
PC-AT
Compatible
CCD
Camera
Four Independent Light Sources
Tilt and Rotate Object Mounting

5.1.2 Test objects

The selection of test objects was made on the basis of obtaining a wide selection of object feature types. The objects were then used with a variety of background and lighting combinations to yield a set of test images. The resulting images include: high and low contrast examples; noisy images; and textured objects with a variety of textured and non-textured backgrounds. The set of test images are shown in Appendix (B), with specific examples reproduced in the following sections where appropriate. The template co-ordinates and values, and the sets of templates use for the intersected analysis, are also listed in Appendix (B). Natural objects were used to demonstrate the enhancement and delineation capabilities of the algorithms.

The generation of a stable, reproducible, noise source was made through the use of synthetic noise, added to the image under test. This is in addition to any existing noise within the image, generated as part of the acquisition process.

5.1.3 Conventions for Notation

(i) Use of the R Operator (r)

The name of the image under analysis is followed by the applied algorithm and the name of the template in brackets ( ), with the conditions used given as a pixel count in braces { } where appropriate. The name of the result file follows.

For example:

is1b.img (r,donut{13,1}) is1bd131.per

would indicate the application of the r operator with donut template with conditions of between 13 and 1 pixel containment to file is1b.img, the result stored as is1bd131.per.

(ii) Use of the MHMM Transform (rjhma)

The convention here is essentially as described in (i) above, but the two sets of conditions inside the braces, hit set conditions first, are separated by a colon where the same probe is used both for hit and miss:

is1b.img(rjhma, donut{6,1:6,1})

(iii) Use of Intersected R Analysis (rin)

The name of the analysed image is followed by the applied algorithm and the name of the template set in brackets, with the sets in square braces [ ] and their respective conditions in braces { } separated by commas:

booze.img(rin,dset.par[smalldo{11,1},donut{13,1},rest{13,1}])bozlo.rin

5.2 Feature Extraction

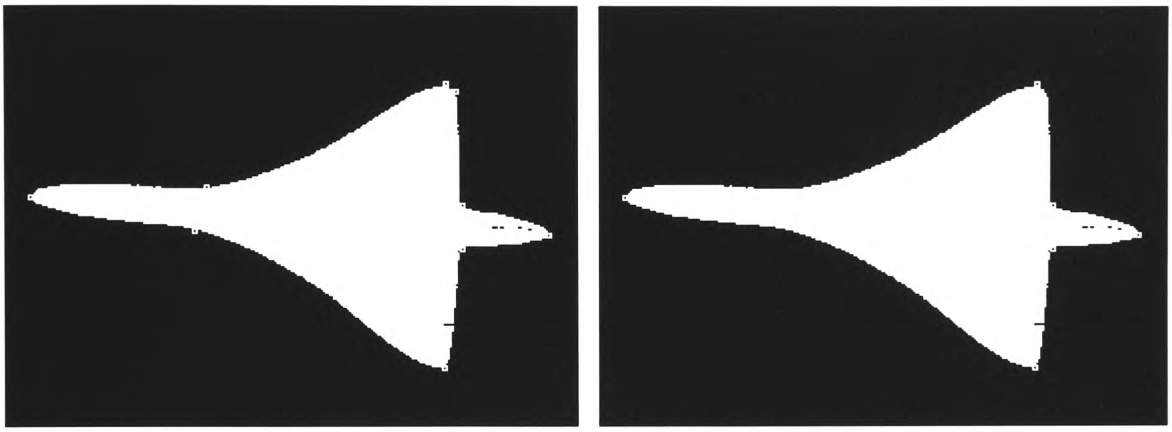
The feature extraction methods described in Chapter 4 Section 4.6 were applied to a variety of test objects, and the results shown and listed below were obtained. The question as to which features are pertinent to the perceptual reasoning about an object in an image, and how the loci of such features are to be obtained, is an important one. As has already been discussed (see section 3.3, or Davis [32]), the structural features used for recognition are likely to mimic those used for human perception. However, the exact loci of the features' perceptual point of action is a matter of the viewer's interaction with the scene.

In order to obtain a valid estimate of the loci of perceptually important features for the test object images used, a group of five people were instructed to assign points of importance in terms of high concave and convex curvature by locating control points on the image perimeter. The process was repeated by each person weekly over a period of ten weeks, to remove training errors. The results were then averaged to obtain a best locus for each feature. The data obtained is used for comparison with that extracted from the images using the various feature identification algorithms applied later in this chapter. This should offer a better measure of the actual locus of the perceptually important perimeter changes than a single person estimation. The variation in locus assignment rarely exceeded three pixel positions for the common set of features; consequently, this limit was applied to the set of features used for the evaluation of practical results.

The set of features identified by the test subjects included some that were relatively small, and disappeared for low levels of rotation from the pose evaluated. Others were of relatively low curvature, but perceptually significant for appearing between two perimeter runs of lower curvature. Given that the method adopted requires a specified level of curvature for the geometry to meet the identification criteria, several of the features identified for each object are unlikely to be found by the algorithm applied. On this basis, a second pass was made through the features to identify a control group of features, usually structurally gross and likely to remain of importance through reasonable levels of rotation of a particular pose of the object. A further criterion, that of relatively high curvature, was added. The features identified are those likely to be automatically selected by a perimeter following or line fitting algorithm as the vertices of the object.

In the quantitative data presented in section 5.4 later in this chapter, data for extraction efficiency is presented for the control group features, in addition to the full set of perceptually important features. This is justified on the basis of the method of feature extraction - it is not entirely appropriate to judge the efficiency of the algorithm against criteria it is not intended to meet. An example of the control group features, and the perceptually significant full set, is shown for the Concord image of IS1 in figure (33) below:

Figure (33): **Concord IS1 Concave and Convex Features Based on High Curvature**



(a) Full Perceptual Set **(b) Control Group Features**

5.3 QUALITATIVE RESULTS

5.3.1 Perimeter Feature Extraction using Chain Codes and Edge Following

Chain coded descriptions of object perimeters contain rich descriptions of structural features in an easily extractable form. The quality and accuracy of the results depended almost entirely on the quality of the pre-processing. In this lie both the strength and weakness of the method: the ability to operate after a variety of segmentation algorithms offers flexibility and easy introduction of enhancements, but at the price of possibly lengthy and sophisticated pre-processing. Because of the sensitivity of the method to edge perturbations, it was found necessary to allow limited edge distortion by lowpass filtering before binarisation, preventing the appearance of edge isthmuses. The process adopted for the practical results obtained is shown in the algorithm of figure (34) below.

An 8-connected chain code was generated by templated edge following. Features were then identified by using a local curvature measure along the extracted perimeter. A multilevel threshold of inferences was used to separate areas of high and low curvature, and to identify areas of high curvature as concave or convex relative to the object body.

The curvature was calculated relative to the perimeter itself by difference equations as:

$$\text{curvature } = d\theta / dS$$

The orientation of the perimeter, θ, was calculated piecewise as:

$$\begin{aligned}\theta &= \Delta y / \Delta x \\ &= (y_n - y_{n-1}) / (x_n - x_{n-1})\end{aligned}$$

where y_n, x_n are points on the perimeter. A separation of around ten pixels between successive points was arbitrarily chosen as giving reasonable results for objects of adequate size within the field of view. Using this curvature measure, points of high and low convexity and concavity on the perimeter were isolated and used for recognition purposes.

Figure (34): Algorithm for the Extraction of Perimeter Features using Chain Codes

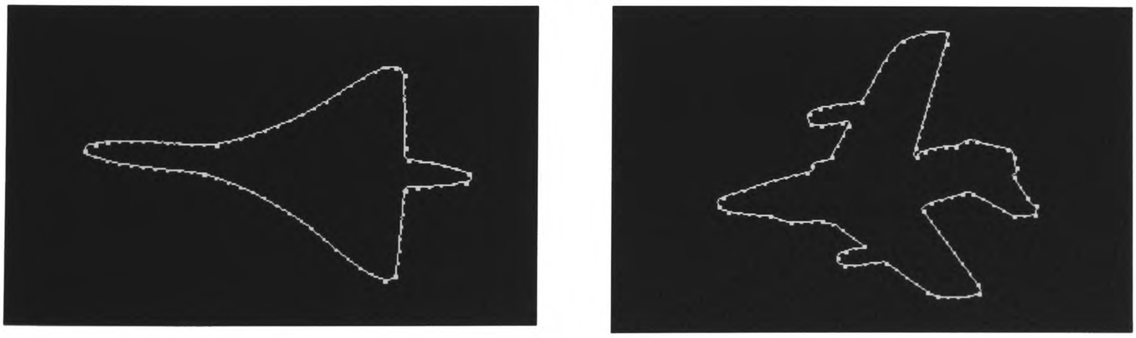
```
Acquire Image

Histogram Equalisation (if necessary)

Lowpass Filtering

Binarisation by Thresholding

Median Filtering

8-Connected Chain Code Generation

Extraction of Features using Curvature Estimation
```

This method again proved useful for objects of high and low contrast (subject to the pre-processing requirements), and with partial obscuration and overlapping. Lowpass filtering improved the quality of the perimeter delineation after binarisation, at the obvious cost of a limited movement of the edge. In spite of this improvement, the perimeter still contained "broken up" data, where the threshold approached the values of the perimeter. This is inevitable with a curved object with lighting constrained from a single source, where the lighting produces variations in intensity across a curved surface. This was minimised by median filtering after the binarisation. Again, some distortion of the perimeter is introduced. This is less significant where gross structural features are to be used for later analysis.

Textured objects proved problematical, in that the very techniques used to segment object and background enhanced the texture in the image. Segmentation supported by textural means would have alleviated these difficulties. The technique, by its very nature, relies on changes in the intensity map to locate perimeter data. It assumes the most significant local variations will be between object and background, and that binarisation will allow isolation of the object as a result. In the presence of deep texture, this cannot be assumed to be the case.

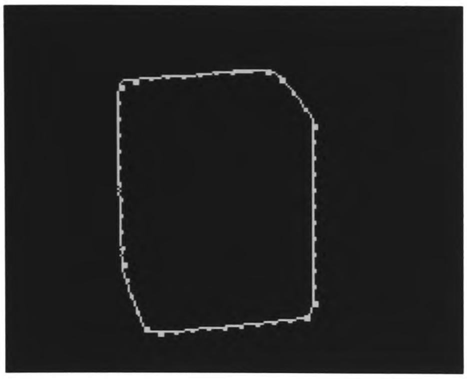
Example results are shown in figures 35(a) and (b) below. The points at which the curvature was calculated are indicated by the smaller blocks on the contours, the points of significant high curvature being indicated by the larger blocks. Extracted feature sets for the adequate contrast images are tabulated in Table (8). The full set of extracted data are included in the comparative features section included in Appendix B.2.1, along with data from other methods.

Figure 35(a): Perimeter Extraction for High Contrast Objects



(i) is1bin.img (gab, diffre)

(ii) is6bin.img (gab, diffre)



(iii) is12bin.img (gab, diffre)

The table illustrates the sequence of features from the perimeter of the objects under test. It is this sequence information that makes this data particularly useful in recognition applications, in that a relatively small sample is required to obtain a positive result.

With objects of adequate contrast, this technique proved effective in the delineation of the required features. Points not belonging to regions of high curvature must, by definition, belong to the areas of relatively low perimeter curvature.

Table (8): Extracted Features (in comparison with features manually extracted by eye placement of cursor)

Is1b.img (eye extracted)			**is1b.img (gab, diffre extracted)**		
x	y	type	x	y	type
353	288	1	352	290	1
234	340	0	227	341	0
147	345	1	155	349	1
228	362	0	235	363	0
352	431	1	355	419	1
361	371	0	367	369	0
404	364	1	393	357	1
361	349	0	359	343	0
358	292	1	339	293	1

Is12b.img (eye extracted)

x	y	type
295	315	1
296	408	1
310	442	1
392	431	1
393	332	1
372	308	1

is12b.img (gab, diffre extracted)

x	y	type
296	326	1
300	414	1
316	440	1
392	425	1
-		
361	310	1

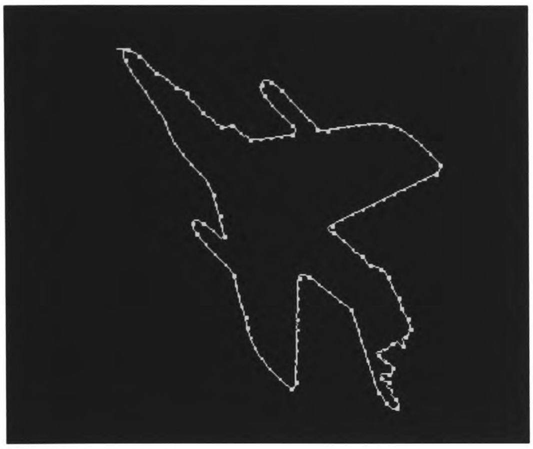
In the example overleaf, figure 35(b), the problem is one of difficulty in delineation. To improve the situation, the following procedure was adopted: a low pass filter was applied, to smooth the perimeter; the image was binarised by thresholding; and a median filter applied to tidy up the ragged edges of the boundary. This necessarily introduces perimeter distortion, but allows the gross description of the boundary to be used to extract structural features. The extracted feature set exhibit problems with misplaced positive responses. Reducing the binarisation threshold caused inclusion of islands of noise and texture in the result.

The increasing of the run length over which the perimeter orientation is calculated offers a means of reducing the problems by averaging. It does, however, produce gross distortion of the position of features, beyond that reasonably worthy of consideration for a feature extractor for a recognition algorithm.

Given the over-inclusion of features in the resultant data set, including false features which are generated as a result of poor extraction at positions widely at variance from the actual nearest perceptually significant point, some further processing to improve the situation is required

The problem now becomes one of the isolation of the appropriate features for the object description. With a fixed scaling and orientation, the method will work adequately. The required features may then be matched point by point, eliminating consideration of the bulk of spurious features.

Figure 35(b): Perimeter Extraction for Low Contrast Objects



bozlb.img(gab, diffre) bgdboz.clp

Such an approach is, of course, unsuitable for recognition where rotation, translation and scaling are not fixed. Here the features must form part of the recognition process, rather than simply offering the data solution. The excessive inclusion of spurious features removes the possibility of using edge sequences for recognition, which would require some portion of the perimeter to be uncorrupted.

5.3.2 Perimeter Feature Extraction using the R Operator and Conditions

The appropriate selection of template conditions permitted the direct extraction of perimeter features, with varying degrees of success. The basic algorithmic approach is described below.

Figure 36: Algorithm for Perimeter Feature Extraction using the R Operator

```
Acquire image

Apply R operator, generate $R(x,y)$

Apply conditions to $R(x,y)$ to extract perimeter features
```

As is described in section 4.6, this method uses conditions applied to the R set to determine the presence of specific types of feature. Where the object and background are easily separable, perimeter features are clearly isolated. The categories of gross structural features are usually identified, along with a limited range of spurious effects.

This method relies on the spatial averaging implicit in the template area to minimise the aliasing caused by random noise effects. Inevitably, this limits the usefulness of the technique in its simple form under noisy environments. The intensity gradient of the template is an important parameter. As the gradient of the template comes closer to that of the object/background boundary, the extracted features become more localised and the spurious effects more limited. The establishment of the optimum gradient would require either more control of the environment or prior processing to select it (via some method such as selection from the Rayleigh distribution of the intensity map of the image). Given that this is not usually possible, the use of a lower gradient and acceptance of a higher number of spurious features offers an alternative.

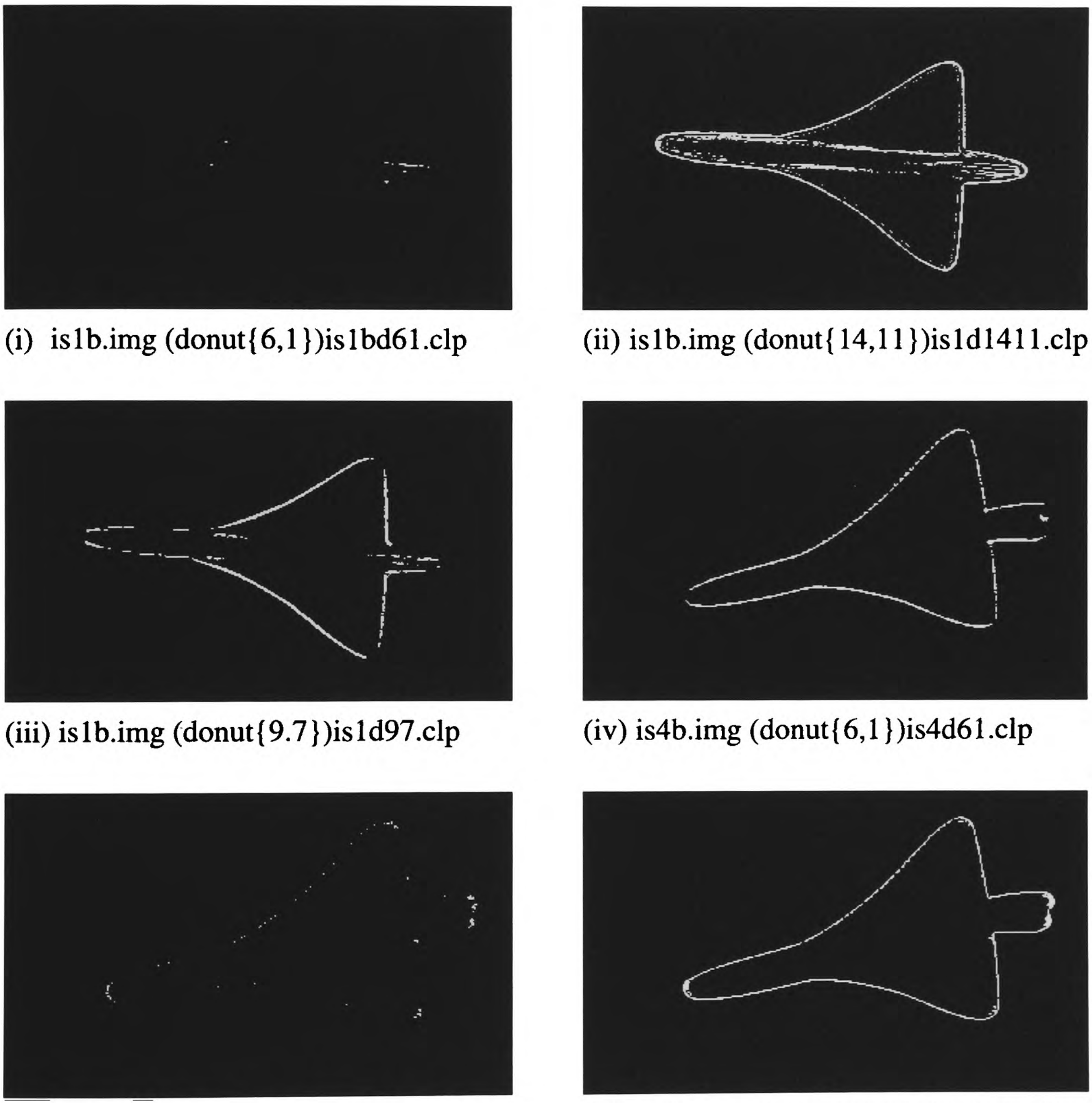
Where the noise effects are truly random, using multiple sizes (scales) of the same template should reduce the level of interference by evaluating on the basis of a different set of points. If the feature required is of substantial size, several different scales should be evaluated and the presence of the feature deduced from the multiple results. It is true to say that a larger scale will necessarily reduce the localisation of the feature. This multiscale approach is described in section 4.6.1, and has been used to produce the images of section 5.3.4.

In practice, it was found that attempts to completely localise a feature were more likely to lead to its disappearance. A better result was achieved by accepting a localised cluster about the feature locus, and improving the targeting by shrinking the resulting cluster rather than applying higher levels of analysis at the earlier stage. Example results are shown below in figure 37.

The R operator produces a rich feature environment, and requires further processing to extract the relevant information. This may take the form of filtering the image, or the extraction of cluster location from the conditioned result. The principle objective, that of direct extraction of structural features in a form appropriate for passing to a recognition algorithm, is not adequately achieved. In practice, the perimeter following algorithm of section 5.3.1 can extract feature data from the perimeter images, and the median filter and its morphological analogue can reduce the size of cluster to an acceptable extent, as is indicated in Appendix B.2.

Figure 37: Perimeter Feature Extraction with the R Operator and Conditions

(a) Adequate Contrast, Low Noise Examples



(i) is1b.img (donut{6,1})is1bd61.clp

(ii) is1b.img (donut{14,11})is1d1411.clp

(iii) is1b.img (donut{9.7})is1d97.clp

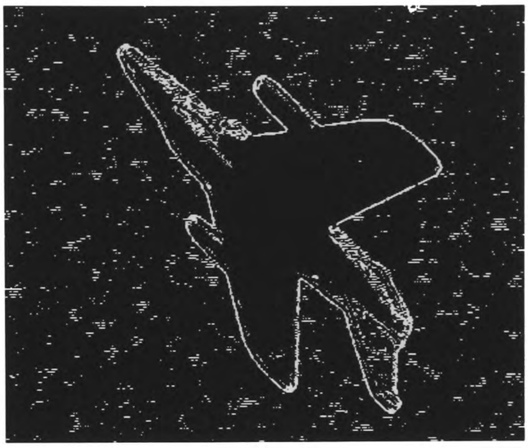
(iv) is4b.img (donut{6,1})is4d61.clp

(v) is4b.img (donut{10,10}) is4d1010.clp

(vi) is4b.img (donut{14,10}) is4d1410.clp

(b) Low Contrast Example

(i)booze.img(donut{6,1})bozd61.clp



(ii)booze.img (donut{14,11})bozd1411.clp

(iii) booze.img (donut{3,1})bozd31.clp

5.3.3 Extraction of Perimeter Features using the R Operator based MHMMT and Templates

The use of the conditional mostly hit, mostly miss transform as defined in equation (14) gives added scope for the elimination of unwanted perimeter data. The examples below show the use of the donut template with suitable conditions to extract points of high curvature, concave and convex, from a variety of high and low contrast images. Whilst highly effective with low noise, low textured images, this algorithm offered little improvement for the more complex situations.

Figure (38): Examples of MHMMT Extraction of Features

(a)Reasonable Contrast, Low Noise Images

(i) is1bdocv.hma(donut{6,1:6,1})

(ii) is1bdocc.hma(donut{14,9:14,9})

(iii) is4bdocv.hma(donut{6,1:6,1})

(iv) is4bdocc.hma(donut{14,10:14,10})

(v) is4bdoc1.hma(donut10{14,9:14,9})

(vi) is6b hma(donut{14:11,14:11})

(vii) is6b hma(donut{6:1,6:1})

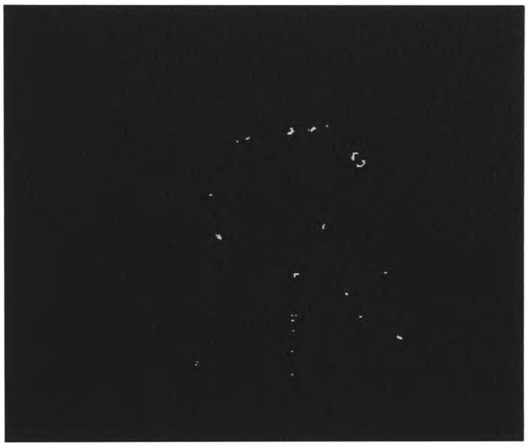
The increased selectivity of the MHMMT is clearly shown in figures 38 (a)(i) - (vi). Figures 38 (a)(iv) and (v) show the effect of increasing the gradient of the template. The features are more localised, but with further increases in template gradient threshold may well be missed altogether. The feature sets extracted here are of an appropriate form for passing directly to a structural recognition algorithm. The example image referred to as IS6B (figs (vi) and (vii)) is used later for quantitative analysis of the effects of noise on the algorithms applied.

is4b.img (eye extracted)			MHMMT		
x	y	type	x	y	type
365	271	1	360	271	1
423	326	1	420	325	1
			421	327	1
			420	329	1
420	342	1	418	343	1
166	388	1	168	386	1
			168	387	1
			168	388	1
382	406	1			
377	326	0	376	327	0
418	334	0	416	332	0
381	348	0	379	348	0
248	357	0	-		
260	380	0	-		

The low contrast image of figures b(i) - (iii) below are examples of the peak performance of such an algorithm. The features are located and localised, with little spurious data included. The template had been optimised neither in shape nor gradient for the solution.

(b) Low Contrast, Low Noise Images

(i)amber.img (rjhma4, donut{6,1:6,1})



(ii)amber1.img(rjhma4, donut{14,9:14,9})

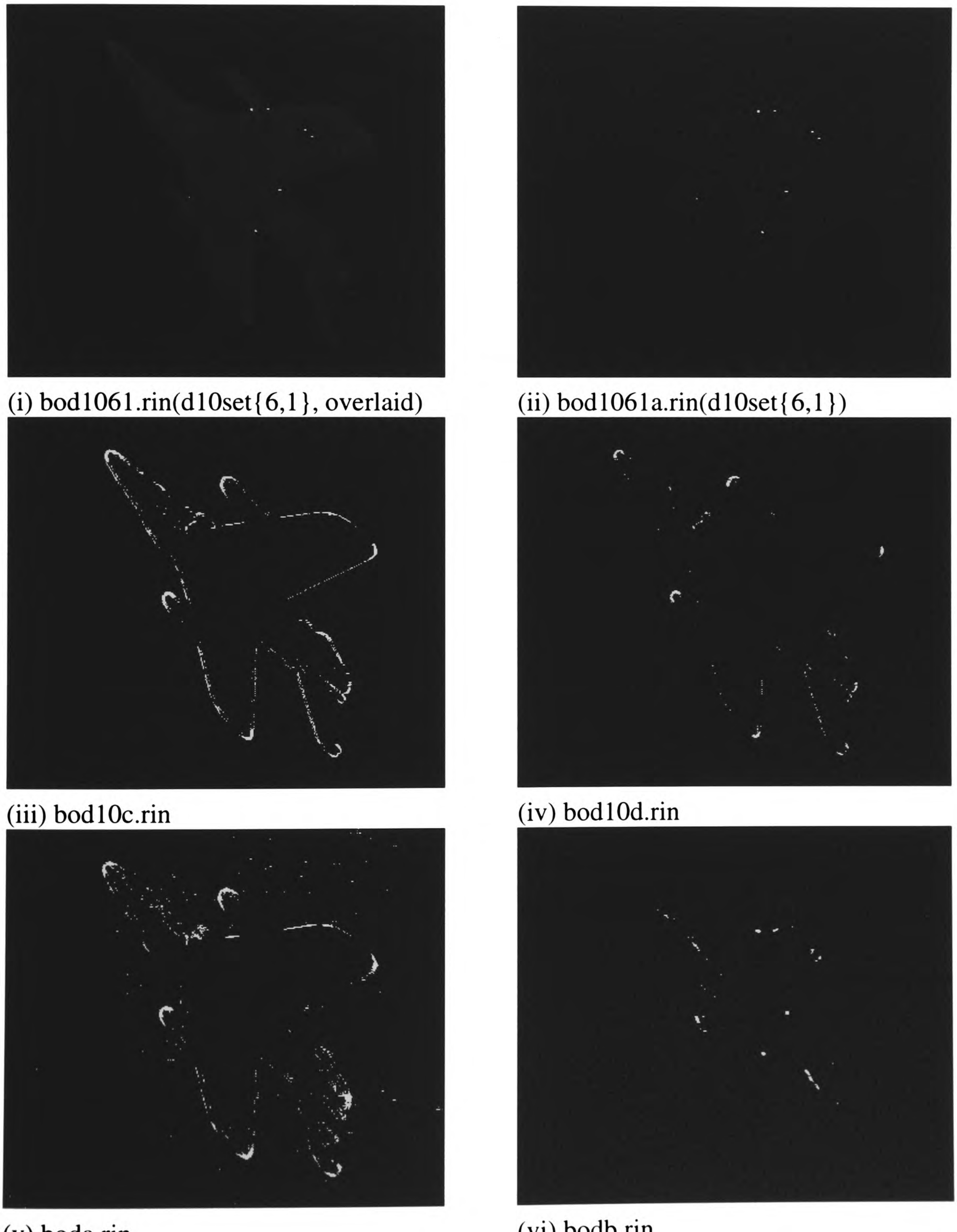
(iii)amber2.img(rjhma4,donut{14,10:14,10})

5.3.4 Perimeter Feature Extraction using the Intersected *R* Operator and Conditions

The basic operation of the method, as was illustrated in figure (28), is to apply sequences of templates, take the extracted R sets and apply conditions to them, then intersect the results. The rules applied are generally in common with those used for the single R analysis. Examples of low contrast images are given in figure (39) below. The tightening of the filter here is obtained by multiple intersections, allowing elimination of the perimeter while retaining the significant perimeter features. The clustering is adequate, but, as is shown by figures (39) (iii) - (vi), not sufficiently independent of template gradient.

Extracted concave features for the low contrast image are listed in Table 9 below.

Figure (39): Perimeter Feature Extraction for Low Contrast Images with R Intersected



(i) bod1061.rin(d10set{6,1}, overlaid)

(ii) bod1061a.rin(d10set{6,1})

(iii) bod10c.rin

(iv) bod10d.rin

(v) boda.rin

(vi) bodb.rin

Table (9): Concave Features Extracted using the *R* intersected Method

booze.img (eye extracted)		**rin extracted**	
x	y	x	y
218	186	-	
289	189	283	191
		284	191
269	192	268	191
		269	191
		268	192
		269	192
242	197		
315	209	317	210
		318	210
323	218	323	216
		324	216
204	238	-	
295	266	293	265
		294	265
		295	265
213	273	212	272
219	299	-	
273	304	272	302
		272	303
		273	303
310	331	-	
312	322	-	
349	358	-	
348	398	-	

As can be seen, there is a tendency to multiple responses to recognised corner types. Certain features are missed because of the geometric thresholds used. Although perceptually significant to the eye, their curvature is not adequate to match the requirement set. Other features are simply missed, because of the sampling shape, or the sweep of the profile in the intensity domain.

5.3.5 Extraction of In-object Features using the R Operator and Templates

The extraction of in-object features used the same basic algorithmic approach as shown in 5.3.2 above. The difference comes in the sophistication of the interpretation of the results. Having already shown a method of extracting the object perimeter, the localisation of features within the object is a matter of position: if the result falls within the perimeter it is within the object. The key problem in this work is characterised by the extraction of shape from shading information from the object. The attempt is made to characterise surface features from the intensity map profile relative to the background area.

There are two broad strategies applicable to this problem. The first is to locate the structural components of the feature from the intensity map. This may be enhanced by oblique lighting, a constraint which has not been previously needed. There is little reason to assume general features will be easily isolated due to significant intensity differences between the feature and the background, which is now the body of the object itself. The second is to define the feature itself as the template, and attempt to identify the feature as a whole within the intensity map. This will require the fixing of scale and rotation prior to the analysis.

Only the first method was applied. The separation of feature and background proved an intractable task, with shading effects often more significant than the features. The degree of lighting control needed to produce successful results was difficult to achieve using the simple test rig. The best results were obtained by using a dual lighting strategy, initially attempting to obtain uniform illumination to optimise the object/background segmentation, and then using directional illumination to highlight the required features.

Specular reflections from points of high curvature proved useful. The significant increase in intensity caused by off-specular to specular transition were easy to identify, and were used as markers of high curvature when localised in appearance. It is recognised that these are the features which could be easily extracted through straightforward binarisation by thresholding. In general, high curvature areas, by producing significant variation in the intensity map, proved easiest to extract. The coincidental extraction of the wing marking on the low contrast image in figures 38(b)(ii) and (iii), and 39(i) and (ii) illustrate this point.

5.3.6 Noise

The presence of noise causes distortion of the shape of genuine perimeter features, and incorrect location of feature clusters both outside and inside the object. The effect of noise on the R perimeter feature detection algorithm is shown in the following quantitative results. Both salt and pepper noise and Gaussian noise were added to the aircraft images, and the accuracy and completeness of the feature extraction compared.

Problems generally arise when the noise structures are of the size of the features sought by the extraction algorithm. The reduction of noise was attempted by application of the probe templates with appropriate gradient and geometric thresholds. The alternative method, that of removal of the noise by pre-processing, was not considered as part of this work. It would, however, offer an additional enhancement to the quality of the source data, but usually at the cost of some disturbance to the positions of features.

5.4 Quantitative Results

Table (10) below represents the results obtained from the previous analysis considered against the human visually extracted data for the same objects. Some caution is needed in the consideration of the data. The act of placing a cursor at the perceived point of high curvature on a screen image is not guaranteed to give the perceptually important point. It is a conscious act, mediated by the process of placement, and therefore offers the point thought to be the significant one, but not necessarily that on which the unconscious mind bases its decisions as to locus and type. Insofar as the placement is correct, this offers a realistic estimate of the position of point of perceptual significance.

It is clear that the measures applied by the algorithm and the measure applied by the text subjects are similar in the case of most features, by less so where the features are large (the nose cone, the wing tips) or subject to a gradual gradient profile. (See Appendix (B2.1) for details).

Table (10): Feature Detection Results for Various Objects

(A) Concord IS1

Distance	d=3		d=10		d=20	
Algorithm	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
Multiple *R* Template Intersection	0.52	0.56	0.80	0.55	-	-
Mostly Hit, Mostly Miss (1)	0.32	0.22	0.48	0.44		
Edge Following and Curvature Estimation	0	0	0.67	0.67	1.0	0.89
***R* Extracted Perimeter Perimeter Following and Curvature Estimation**	0.11	0.11	0.56	0.56	0.89	0.77
(1) Key Features only		0.8				

d = distance between actual and assigned corner

P(AC|TC) = pr(assign corner and corner exists within specified distance)=(corners within distance/no of corners); note this does not include multiple hits.

P(TC|AC) = pr(true corner within specified distance if corner assigned)=corners within distance/no of corners assigned); this does include multiple hits on the same corner

(B) Rotated Concord IS4

Distance	d=3		d=10		d=20	
Algorithm	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
Multiple *R* Template Intersection (1)	0.85	0.80	0.94	0.8		
Mostly Hit, Mostly Miss	0.82	0.5	0.91	0.6		
Edge Following and Curvature Estimation	0	0	0.71	0.5	1.0	0.6
***R* Extracted Perimeter Perimeter Following and Curvature Estimation**	0	0	0.33	0.4	0.67	0.6
(1) Key Features Only		1.0				

(C) Hawk Trainer IS6

Distance	d=3		d=10		d=20	
Algorithm	P(ACITC)	P(TCIAC)	P(ACITC)	P(TCIAC)	P(ACITC)	P(TCIAC)
Multiple *R* Template Intersection (1)	0.57	0.27	0.90	0.40		
Mostly Hit, Mostly Miss	0.86	0.55	0.92	0.61		
Edge Following and Curvature Estimation	0.08	0.06	0.72	0.58	0.92	0.68
***R* Extracted Perimeter Perimeter Following and Curvature Estimation**	0.17	0.03	0.67	0.13	1.0	0.16
(1) Key Features only		0.56				

(D) Square Block IS12

Distance	d=3		d=10		d=20	
Algorithm	P(ACITC)	P(TCIAC)	P(ACITC)	P(TCIAC)	P(ACITC)	P(TCIAC)
Multiple *R* Template Intersection (1)	1.0	0.5				
Mostly Hit, Mostly Miss	0.25	0.17	1.0	0.17		
Edge Following and Curvature Estimation	0	0	0.6	0.5	1.0	0.83
***R* Extracted Perimeter Perimeter Following and Curvature Estimation**	0	0	0.8	0.67	1.0	0.67
(1) Key Features only		0.56				

(E) Low Contrast, Rotated Hawk BOZ

Distance	d=3		d=10		d=20	
Algorithm	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
Multiple *R* Template Intersection (1)	0.88	0.4	1.0	0.4		
Mostly Hit, Mostly Miss	0.72	0.6	0.92	0.67		
Edge Following and Curvature Estimation	0	0	0.47	0.5	0.77	0.7
***R* Extracted Perimeter Perimeter Following and Curvature Estimation (d10.par)**	Invalid - excessive false hits					
***R* Operator Single Template donut10.dat {6:1}**	0.89	0.47	0.96	0.53		
(1) Key Features only		0.69				

The tables above illustrate several points. In general, where corner are indicated, they do exist within reasonable distances of an actual corner in the image. Our basis of evaluation of a perceptually significant feature has been discussed previously. It is evident that the extraction of features misses many of the features, typically of the order of fifty percent. This raises questions as to how the object should be modelled, which are dealt with explicitly in section (6.5).

The algorithms were not specifically tuned for best results. The donut template was used (gradient threshold five grey levels) for the great majority of the work, with some success. Closer tuning of the gradient can assist in reducing multiple hits with the *R* intersected method. However, it is a truer test of the utility of the algorithms to use a relatively low threshold, and compare the results so obtained.

The results of Table (11) below compare the features extracted from the is6 image using four different gradients of the donut.par set of templates (see Appendix (A.3) for details of the templates contained within the set).

Table (11): Comparison of IS6B.IMG Intersected *R* Extracted Convex Features at Four Different Gradients of the D.par Series Template Set, Limits at {13,12}

Distance	**d=3**		**d=10**	
Probe Template Set	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
d.par{13:12}	0.73	0.38	1.0	0.38
d10hug.par{13:12}	0.79	0.44	1.0	0.44
d20hug.par{13:12}	0.71	0.44	1.0	0.44
d30hug.par{13:12}	0.5	0.13	1.0	0.19

There is an apparent optimal gradient threshold band for feature detection. Too low a gradient and features are masked by a surfeit of noise-generated false hits; too high a gradient and they are not found. This is, of course, dependent on the contrast in the image. More specific forms of surface fitting (such as is implied in Zuniga and Haralick's facet based detector, with its underlying cubic fitting to the intensity surface) can make better use of this information.

The general assignment of features occurs close to the feature loci in the majority of cases, particularly where adequate contrast and low noise coincide.

The results of Table (12) below compare the use of the MHMMT method and the *R* intersected method in the presence of Gaussian noise. Noise was added at a signal to noise ration of five. As can be seen, the MHMMT method produces the more accurate results, but rejects more of the features. The *R* intersected method is less effective in rejecting false features, but includes significantly more of the object features. Given the implicit surface fitting with the MHMMT, this is to be expected. The MHMMT also requires a better estimate of detection gradient to work fully effectively than does the *R* intersected method. The common factor is, however, the relatively low number of perceptually significant features found. The confidence in any recognition of an object based on these figures is likely to be low.

The choice between the two rests on the need for exact location and low numbers of multiple hits balanced against a higher number of features located. The decision rests on the use to which the data is to be put. As is discussed later, the recognition algorithms used in this work generally produced better results with the larger number of features located by the *R* intersected method, particularly where the multiple hits occur very close to the corner which generates them.

Table (12): Gaussian Noise at Signal to Noise Ratio 5 added to the IS6 Test Image

Distance	**d=3**		**d=10**	
Probe Template Set	P(AC|TC)	P(TC|AC)	P(AC|TC)	P(TC|AC)
Concave Features				
rin, d10hug.par{5:3}	0.47	0.33	0.8	0.53
mhmmt, donut10{6:2,14:10}	1.0	0.40		
Convex Features				
rin, d10hug{13:12}	0.70	0.38	1.0	0.38
mhmmt, donut10{6:1}={15:10}	1.0	0.25		

5.5 Summary of Discussion

There is a basic point to note in the previous work. Having loosened the constraints on the probe template, we are effectively applying all the possible variants, made from the *m* pixels out of *n* of the template, to the image at the same time. The result simply shows that one of the sequences of variants is applicable at the given point, but does not identify which one.

This is, then, a coarse tool, likely to pick up various spurious formations as well as the desired objective. In spatial frequency terms, we are broadening the response of the filter (albeit in a controlled way). This will allow more information to pass through, and should prevent the over-tight response of the direct operator equivalents such as the hit and miss transform and set erosion with hard limits.

There would be information handling advantages to determining which of the possible responses had occurred, but this merely returns us to the "hard power" technique of trying each variant in turn. This is inefficient on processing grounds alone. [Assume a template has 16 defined points within it. For a template limit of 9 points contained, the number of applications would be

$$^{n}C_{r} = \frac{n!}{(n-r)!r!} \text{ where n = 16, r = 9,}$$

to cover the range of combinations. This yields a possible 11440 combinations. The advantage of such an approach is that the source of the positive result is identified, enabling rule-based elimination of a variety of sources of incorrect response.]

In practice, of course, we would only apply the nine pixel template in each of the sixteen possible complete orientations, and any acceptable variations. We have over-loosened the filter to some extent, and the remainder of the work consists of considering how the re-tightening might be accomplished practically and to an appropriate degree, to allow adequate selectivity with enough flexibility to accept the likely range of correct input.

The three methods attempted produced varying degrees of success.

R Operator Method

The R operator represents the broadest filter. In this case, the options for tightening the response are twofold, by increasing the gradient of the template, and by tightening the conditions for a positive response. Neither method is without dangers: over-tightening the conditions will reject useful features; increasing the gradient will eliminate weaker intensity structures which may be relevant to recognition.

As can be seen from Table (13) below, the increased gradient of template is effective in excluding additional data from the result. The extent to which this is applicable to the general case of object recognition is arguable. It presupposes a knowledge of whether the image is textured or non-textured; it assumes that the most significant changes in the intensity map will relate to features of interest (there is an obvious flaw in this assumption); it assumes that the relevant features for recognition will be those extracted at the highest gradient. As can be seen from the data, other features may appear as the threshold gradient eliminates more of the structures in the intensity map. There may be some case, where possible, in reducing the intuitive supervision of the algorithm inferred by using "perceptually significant" features, and allowing it to find its own feature models based on live data. This would require the rotation of the object within a sequence of images, the relevant feature sets being generated whenever a new significant feature appears or an established one is obscured. Later rationalisation could be applied to reduce the likely excess of library models created.

Table (13): is6 concave features extracted using r operator with limits 6,1 at different template gradients

donut				donut20		donut30	
x	y	x	y	x	y	x	y
317	157	274	185	313	158	275	186
314	158	273	186	318	158	276	186
315	158	274	186	313	159	285	279
316	158	275	186	314	159	286	279
317	158	272	187	318	159		
318	158	273	187				
314	159	274	187	275	185		
315	159	275	187	274	186		
316	159	276	187	275	186		
317	159	272	188	276	186		
318	159	273	188	275	187		
314	160	274	188				
315	160	275	188	328	209		
316	160			329	209		
317	160	329	207				
318	160	328	208	371	230		
315	161	329	208				
316	161			335	251		
		371	229	335	252		
		372	229				
		373	229	285	279		
		371	230	286	279		
		372	230	287	279		
		373	230	286	280		
				287	280		

MHMMT Method

The MHMMT approach has several advantages. It is limited in the degree of processing required, to an image inversion, two R operations, and two comparisons (but only for candidate points not rejected by the first comparison). This is a pointwise operation, and therefore of order (absolute worst case) [(NxM)+2(NxMxP)], N and M being the width and height of the image in pixels respectively, P being the number of pixels in the probe template. These are, however, all purely integer operations, mostly add and comparison and increments for the probe at each point in the image. As will be discussed later, such an approach lends itself to a coarse grained parallel implementation.

The obvious increase in selectivity is highlighted clearly by the examples shown, both in the isolation of features as a positive benefit and in the incomplete boundaries for perimeter extraction as a problem. By adopting the method of intersection, the MHMMT produces a clear delineation of the information required, and is effective in the rejection of noise. Some spurious data is contained, but this results more from the logical difficulties in designing the right spatial pattern for the probe than from inherent inadequacy of selectivity.

The selection of the complementary template limits for both hit and miss probes is slightly more subtle approach than it first appears. The reliance is on the fact that some elements which meet and exceed the hit criteria will fail on the miss criteria. The noise elements can be assumed to increase or reduce the value of the grey level at a particular pixel. Thus it may come to be part of one or other of the solutions as a result (see equations [7] and [14]), but will consequently undershoot the requirements of the complementary operation.

It is only at the perimeter of structures in the intensity map that the conditional uncertainty specified can permit inclusion in the result. By the use of complementary conditions, the degree of exclusion is controllable. [Again, the basic rule applies - widen the filter response and accept more included noise]. The noise effects, which will increase the size of cluster produced for a given feature type, are limited, reducing the likelihood of spurious inclusion as the template approaches the boundary. The rejection of actual features is common with this method.

Generalised R Analysis Method

Consider a situation in which the gradient of the template is fixed. There are now two ways of increasing the noise rejection of the probes. Firstly, we could increase the size of the current shape of the probe, effectively requiring more feature to produce a positive response at the given location, and more noise to produce a spurious one. Secondly, we could change the shape of the probe, but retaining its size. Depending on the nature of the noise in the image, and the size of features sought, either could prove effective. The former, however, should prove more robust under "blocky" noise, due to its weaker localisation in effect offering similar advantages to averaging over a larger template in conventional means. It will prove equally prone to random noise, if it is to contain the same number of pixels as the alternative shape of template. As a result, this method was adopted for the majority of the generalised R analysis work.

The majority of decomposition and analysis work recently discussed in the literature relies on flat probe templates, allowing reconstruction from the decomposed images. However, the objective here was not decomposition, but instead extraction and characterisation. The vertical edges preserved by non-flat templates are likely to be the prime candidates for features, and as a result the gradient of the template forms one of the determinants of the success of extraction by elimination of spurious data.

5.6 Indirect Extraction - Application of the *R* Intersected Method to Previously Segmented Data

The direct feature extraction processes discussed above include a dual requirement of an adequate gradient threshold to permit proper feature extraction, coupled with the correct geometric threshold for the classes of perimeter structures required. This may risk obscuring the capability of the techniques developed for feature extraction. The *R* intersected algorithm was therefore applied to data from pre-segmented images of aircraft, and the images contaminated with appropriate boundary noise to simulate noisy and inefficient segmentation at a pre-processing stage. The results of this activity are summarised below, using the Hawk trainer of image IS6 as the example.

Noise is added in two ways, as pixellated noise, and as a bulk volume distortion. The objective is to demonstrate the utility of the feature extraction process under standard conditions, without the initial problem of segmenting the image.

Pixellated noise was added to the aircraft perimeter, simulating the deviation of perimeter pixels produced by binarisation of a noisy image. The concave and convex features were extracted, using the donut set of templates with suitable thresholds (it should be noted that only the shape of the template matters in this instance - the gradient being ignored due to the pre-segmentation and binarisation of the object in the image under test). The thresholds adopted, as in the previous work, were three or four pixels contained across all templates for convex perimeter features, and eleven to thirteen pixels contained for concave features. The extracted vertices are listed in Appendix B.4.

The results obtained are shown below in figures 41 to 43:

Figure (41) :IS6 Convex Feature Extraction in Pixellated Noise

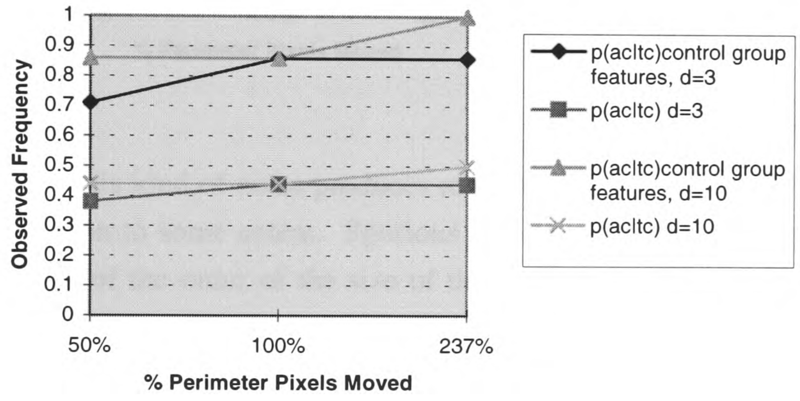


Figure (42) :IS6 Concave Feature Extraction in Pixellated Noise

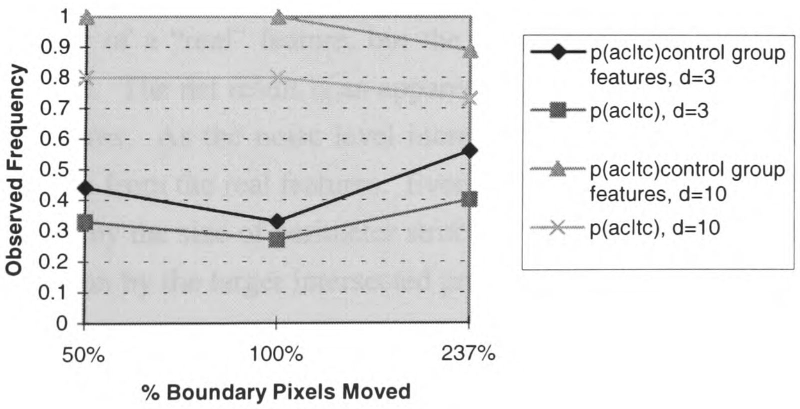
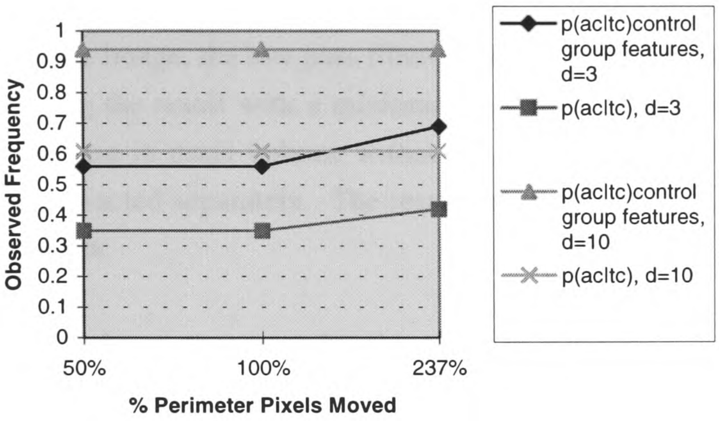


Figure (43): IS6, All Features in Pixellated Noise



This kind of noise produces relatively little effect on the vertex isolation, moving its position to some extent. Spurious hits do start to occur once the perimeter distortion becomes of the order of the size of the features the template are capable of identifying. Concave features are normally more robust under these conditions. For the low deviation limit feature estimation, the higher levels of noise produce significant problems, in that the likelihood of the feature locus having moved by more than the acceptable distance is relatively high.

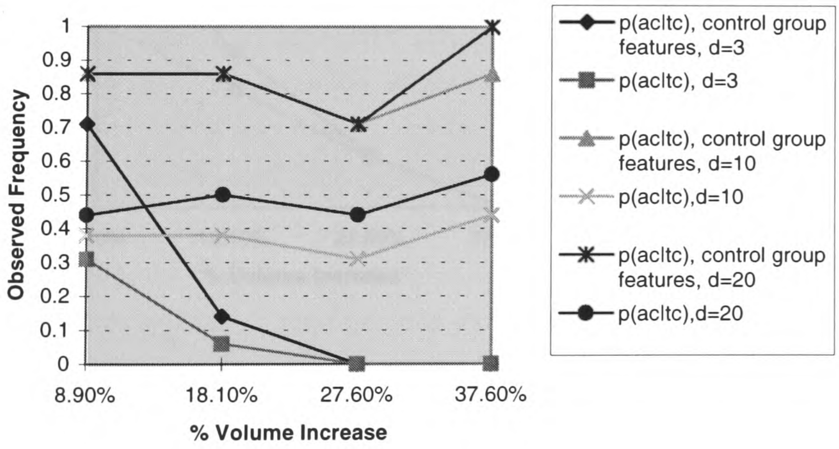
The apparent improvement as noise increases is caused by the occurrence of perimeter structures of high convex curvature. The larger probe templates still require the proximity of a "real" feature, but the noise introduces elements with high curvature of small size. The net result is an apparent increase in detection success, at loci close to the real features. As the noise level increases further, false results will occur at significant deviations from the real features. Eventually these will come to dominate the result. This is caused by the size of perimeter structures becoming of the order of the size appropriate to detection by the larger intersected probe templates.

The methods developed should be less resilient than the line matching algorithms in the presence of pixel orientated boundary noise. This can be explained by considering the averaging inherent in the line matching solution. The line segment is usually subject to a minimum size, normally longer than the nine pixel diameter of the largest template in the intersected *R* template set used here.

Bulk noise was added, in the form of a volume distortion of the perimeter of the image. This was generated by adding random pixel noise to the perimeter of the object within the image, the low pass filtering the image with a three by three template, and then binarising the result with a minimum threshold. The volume distortion is calculated by the increase in pixel volume within the object. The concave and convex features were again extracted separately. The results obtained are shown graphically in figures (44) to (46) below.

Volume distortion (blocky noise) of this type might occur due to over processing to reduce noise, followed by binarisation, producing a blocky structural appearance. Once again, spurious hits occur where the noise is of the size of the features picked by the algorithm. Here the perimeter distortion is much more substantial, and the convex features forming the extrema of the object are found to be more reliable for extraction in its presence. The quantity of false hits generated for concave vertices was generally high (of the order of, or greater than, the number of correct hits).

Figure (44): IS6 Convex Features in Bulk Noise



Figure(45): IS6 Concave Features in Bulk Noise

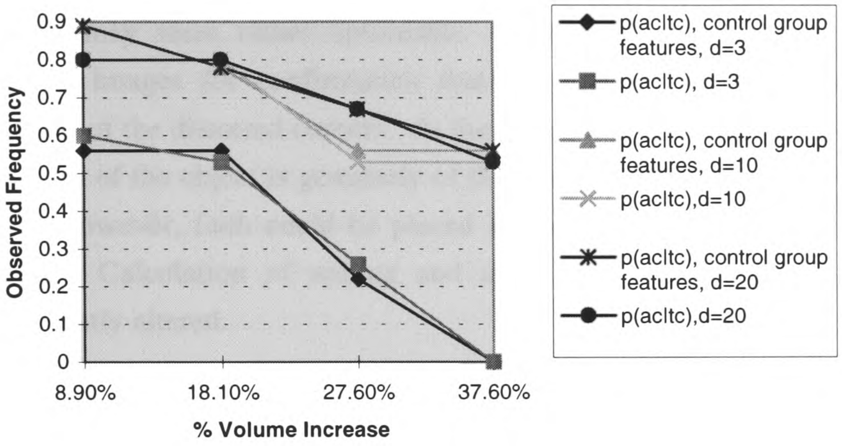
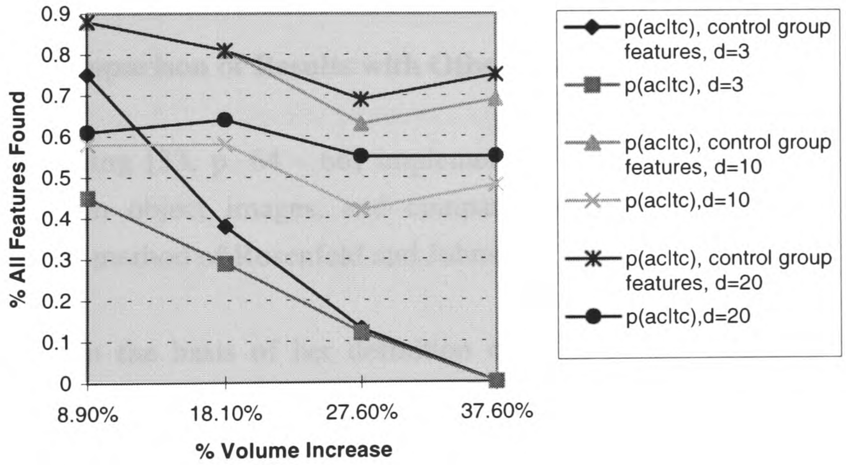


Figure (46): IS6 All Features in Bulk Noise



Volume noise produces distortion in the perimeter of the object. The size of the perimeter structures produced rapidly reaches the order of the size of the largest template in the intersected extraction probe set. False hits, therefore, start to occur. The apparent increased performance at the loosest limit is due to the presence of false hits.

Once again, as with pixellated noise, while the structural distortion remains relatively small, the likelihood is that false hits will occur only in close proximity to real features. The main difference here is the rate of increase of size of perimeter noise structures is significantly more rapid.

The extraction of features at greater than ten pixels deviation from the original position may seem rather optimistic. The results were visually compared with the distorted images for confirmation that the features were apparently at, or near, the positions of the distorted corners. At the higher levels of bulk noise, the movement of the perimeter of the object is genuinely of the order of ten pixels in the image. The extent to which, however, faith could be placed in this sort of result for recognition purposes is limited. Calculation of scaling and orientation relative to the model pose will be significantly altered.

As an example, for an object with two identified, well separated features two hundred pixels apart, the orientation error could be as large as sixteen degrees due to the feature location inaccuracy, ignoring any other sources of error. Worst case scaling error could be twenty five percent. For less well separated features, the quality of estimation deteriorates rapidly.

5.6.1 Comparison of Results with Other Methods

Illing [33, p. 64 - 66] implements a line-fitting algorithm for the discovery of vertices in object images, and compares it with the points of maximum curvature detection method of Rosenfeld and Johnston [43].

On the basis of her definition of noise in the boundary (corresponding to the definition of pixellated noise used here, but generated from a single closed contour before filling to a solid shape), vertices were detected to within five pixels of their actual locus at a level of around ninety percent for noise levels of up to one hundred percent, corresponding to twenty five percent of the boundary pixels being moved by two pixel positions towards or away from the bulk of the shape, and the resulting open boundary reconnected. The method we have adopted is capable of matching the rate of success for the major structural features (indicated above as the control group features, for which the results for d=5 were identical to those for d=10).

The line fitting algorithm shares a significant benefit with the method of Rosenfeld and Johnston - it produces very low levels of false hits. This is due to the basic premise underpinning their calculation, that of an area of high curvature following a sequence of low curvature of a defined minimum length as the basis of identification of a vertex or corner. The *R* intersected algorithm is prone to multiple hits at a corner (see Appendix B.4), and, for higher levels of noise, it will introduce false hits as the perimeter structures become part of the set of recognised structures. The levels of false hits are relatively low for pixellated noise.

The method of Rosenfeld and Johnston averaged around fifty percent vertex detection in levels up to 100% noise in Illing's tests. This is significantly lower than has been achieved, but their method has the benefit of not requiring a prior selection of the shape of any vertices sought, other than a basic threshold of inference requirement for the degree of orientation change of the perimeter.

Both the line fitting method of Illing and the curvature estimation of Rosenfeld and Johnston are essentially sequential in their approach - they require a sequence of boundary pixels from which to infer their results. The R intersected method is an isolated one. The pixel window is considered as an individual entity, down to the level of a single image pixel as identified by the locations corresponding to its surrounds in the probe templates used for analysis. For a massively parallel approach, based on simple logic design rather than extensive processor implementation, this is a distinct advantage, and, in terms of speed for hardware solution, a decisive one. The process requires the replication of the data stream, not cross feeding of results; we thus achieve mutual exclusion and data flow dependency only on the source data.

5.7 Texture Analysis as an Example of Feature Classification

One example of the application of feature detection in a general environment is the classification of texture. The characteristics of the intensity map of textured samples potentially include the full range of gradient and shape variations - indeed, the intensity map spatial profile is the characteristic of the texture in the visual sense. A good feature identification scheme, if it is not solely tailored to one particular feature but rather permits identification of a variety of features, might offer a suitable source of extracted features. If the features extracted are characteristic of the texture, then they could be used for texture classification.

Any feature extraction scheme results in a simplification of the rich set of relationships between pixels in the intensity map. One "goodness" measure could be the degree of information reduction achieved in comparison with the quality and resolution of the classification results. In some ways this mirrors Rutherford's comments on statistical interpretation of experimental results, which might be summarised as "if you need to use statistics to explain the results of your experiment, improve the experiment".

The successful application of a simple classification strategy with a small set of features of limited size will only occur if the set of features extracted are characteristic of the texture, and specific to that texture, in the context of other texture samples. This is not to say that such a measure of capability is appropriate to all feature classifiers: very specific classifiers may perform poorly in this kind of application, while fully meeting their design goals. The evaluation of performance of a feature classifier in such an application will, however, give a measure of its utility across the general case. To this end, the *R* operator was used to generate a histogram of the *R* values for a sample of texture with a particular structuring element, as is described in chapter 2 section 2.5.

The analysis is formed as a histogram, *H*, where, over a sample window size $M\text{x}N \in f(x,y)$, with probe structure $k(\xi,\eta)$ the individual histogram values are:

$$H_i = \sum_{x,y \in M,N} p,\ p = 1 \text{ where } r(x,y) = i, i \in CARD(k(\xi,\eta))$$

The *R* operator is applied to the image with a particular probe structuring element. The result is then used to produce a histogram, representing the frequency of occurrence of each *R* value within the image calculated as the count of the number of pixels at which it is found. The histogram is then used as a characteristic signature of the image for that structuring element. By applying several structuring elements with different geometric properties, the set of corresponding histograms generated is used as the description vector for the classification of the texture under analysis.

The set of characteristic histograms for a particular sample window, ***H***, is the feature classifier, which, for a set of N probe structures would be defined as:

$$\boldsymbol{H}_N = \{\{H\}_n\}, n = \text{no of probe structuring elements applied, } n \in N$$

Note that although the sample window applied is square, the resulting shape of area analysed depends, additionally, on the shape of the probe structuring elements and does overlap the perimeter of the window area in its analysis.

This histogram is then treated as the classification vector for the sample. Initial tests were made using data taken from the Brodatz set of textures, with samples drawn from a rich test set of data for Galvannealed steel samples at various states of anneal for later work. The classification results indicate a need for the application of multiple sizes of templates for good class separation.

5.7.1 Experimental Results

The structuring elements used were a set of increasing sizes of a sparse annular template, the donut template used in the examples of figure (38) and (39). These were applied in turn to the acquired texture images and the resulting feature histograms built, and applied to the classification of other samples of textural data. The textures used were drawn from the natural texture examples in the Brodatz [6] set of samples, and also from sample images of galvannealed steel with different degrees of annealing. The Brodatz set textures included D3, D4, D5, D9, D15, D54, D57, D62, D67, D92, D98, D103. Typical examples of natural textures are shown below in figures (47) and (48):

Figure (47): Texture Sample Beans

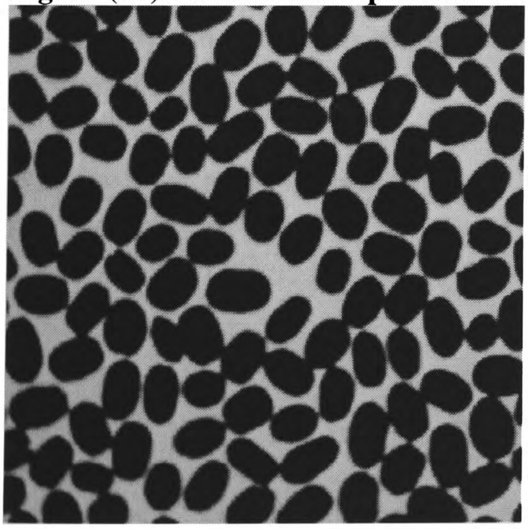
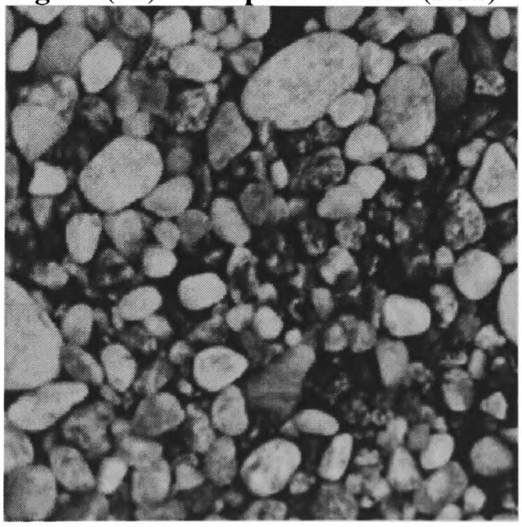


Figure (48): Sample Pebbles2 (D54)



5.7.1.1 Texture Classification Results - Brodatz Texture Samples

Nineteen samples of 256 by 256 pixels, at 8 bit resolution of intensity were used in the tests. Six test samples were generated as subsets of the original images, and these were used to assess the effectiveness of the R histogram as a feature vector for texture classification. Each sample was assessed against the full set of samples using a minimum distance classification. A successful classification is assumed where the maximum distance within the sample group is smaller than the distance to any member of any of the other groups. The inference of this is that of separation of the feature space into classes based on a spherical cluster in the dimensions of the feature vectors.

The effects of sample window size on the histograms obtained was evaluated. The test data samples are of 256x256 pixels, 8 bits intensity resolution. With the largest size of window, some overlap of values within samples is unavoidable. Whilst this reduces the value of the data, it does not invalidate it completely. The samples were chosen to minimise this problem. The results are shown in Tables (14) and (15) below.

The results illustrate a basic point about the use of thresholds in terms of intensity with the donut template. Any gradient inherently reduces the sensitivity of the probe to structures within the image of smaller intensity gradient than the template. Thus, in rejecting "noise", certain structural information in the image is removed.

Table (14): Application of Single Vector (Donut0.dat Template R Values) with Brodatz Texture Samples

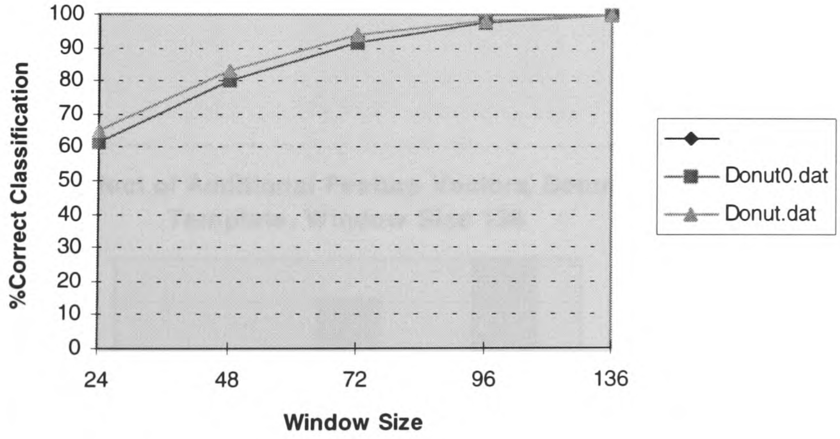
Window Size (MxM)	no of samples	average no of misclassifications	percentage misclassifications	percentage correct classifications
48	684	135	19.7	80.3
72	684	57.68	8.4	91.6
96	684	17.58	2.57	97.4
136	684	0	0	100

Table (15): Application of Single Vector (Donut.dat Template R Values) with Brodatz Texture Samples

Window Size (MxM)	no of samples	average no of misclassifications	percentage misclassifications	percentage correct classifications
48	684	112.6	16.6	83.4
72	684	40.84	6.0	94.0
96	684	13.79	2.0	98.0
136	684	0.89	0.13	99.87

As the graph of figure (49) below shows, full separation for the nineteen texture samples used was possible using a single template, but only at large window sizes. The need for a 136 by 136 window for full success is a major drawback in a practical situation. Again, it should be noted that the values used are generated by a single pass of the *R* operator. It is the later use that is computation intensive. Some sample overlap does occur at this highest window size.

Figure (49): Effect of Window Size on Texture Classification



The use of multiple templates, and therefore multiple vectors, was investigated. This method is close in concept to that adopted in the variety of morphological decomposition algorithms. The scale (size) of the probe template is increased, causing the result to depend on larger and larger area features within the image under test. The addition of the resulting vector separations produced the results shown in Table (16) below. The extra information does permit the separation of all samples at the given window size, which is an improvement over the single template case.

Table (16):Effect of Multiple Size Analysis on Texture Classification, Sample Size 136x136

Test Probe Sets	No of Samples	Average No of Misclassifications	Percentage Misclassifications	Percentage Correct Classifications
Smalldo0, Donut0, Bignut0	684	0	0	100
Smalldo, Donut, Bigdonut,	684	0.32	0.047	99.953
Smalldo, Donut, Bigdonut, Hugedonut, Vastnut	684	0	0	100

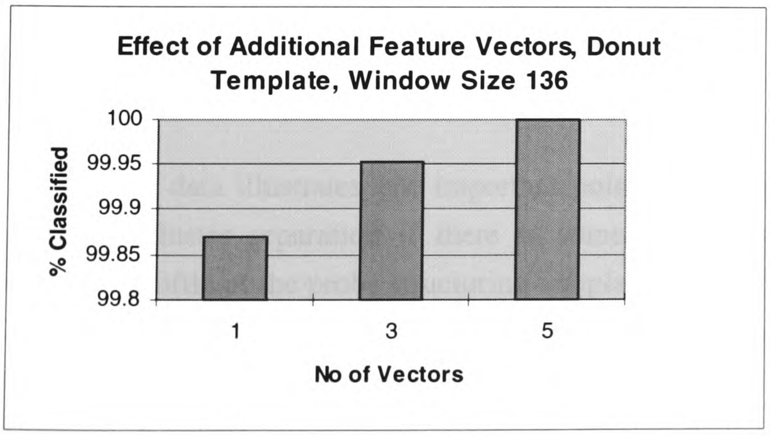
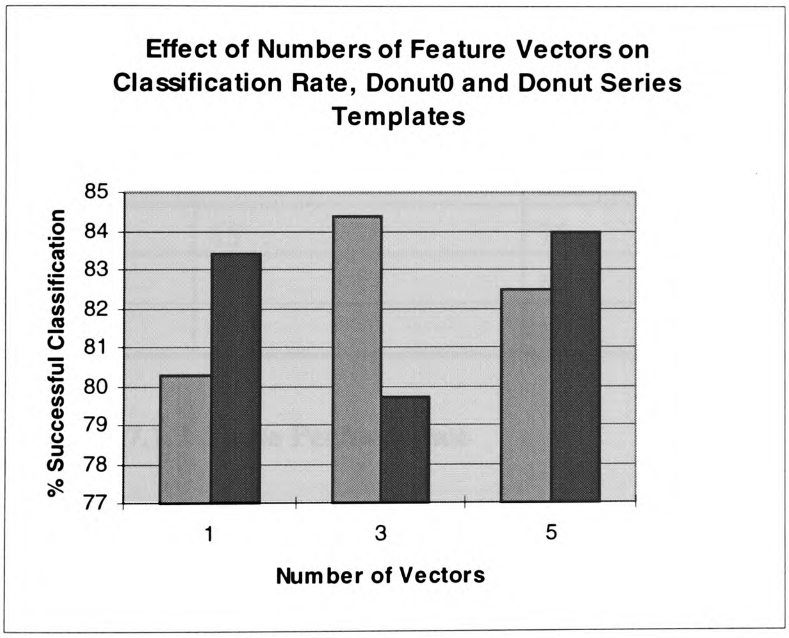


Table (17):Effect of Multiple Size Analysis on Texture Classification, Sample Size 48x48

Test Probe Sets	No of Samples	Average No of Misclassifications	Percentage Misclassifications	Percentage Correct Classifications
donut0 3 sizes	684	106	15.6	84.4
donut0 5 sizes	684	118.5	17.5	82.5
donut 3 sizes	684	137.8	20.3	79.7
donut 5sizes	684	108.3	16.0	84.0
donut20 5 sizes	684	106.1	15.6	84.4
3x3 disk 3 levels	684	86.5	12.8	87.2



The data illustrates one important point. The addition of vector distances only increases cluster separation if there is some significance about the size, shape and intensity profile of the probe structuring template in relation to the texture under analysis. Otherwise the addition of irrelevant, random information may reduce the precision of the analysis. This suggests a need to guide the choice of vectors used for the particular set of samples. A supervised learning procedure, perhaps with a Bayesian classification scheme based on the extracted vectors, might offer a superior classification performance.

5.7.1.2 Texture Classification Results - Galvanneal Samples

The feature extraction method was then applied to a natural, random texture sample, using 380 samples of galvannealed steel in five known, calibrated coating conditions. These images were acquired at a x500 magnification, and were subject in some cases to depth of field problems causing partial blurring of the acquired image as is shown in figure (50). The images were of 512 x 512 pixels at 8 bits grey scale resolution. As can be seen from the results of Table (18), there is a strong correlation between the coating condition and the optical texture as it was measured, but the classification scheme is by no means perfect in its present form.

The samples of figure (50)(a) and (b) are under-annealed, sample (c) is near optimal, and samples (d) and (e) are over-annealed.

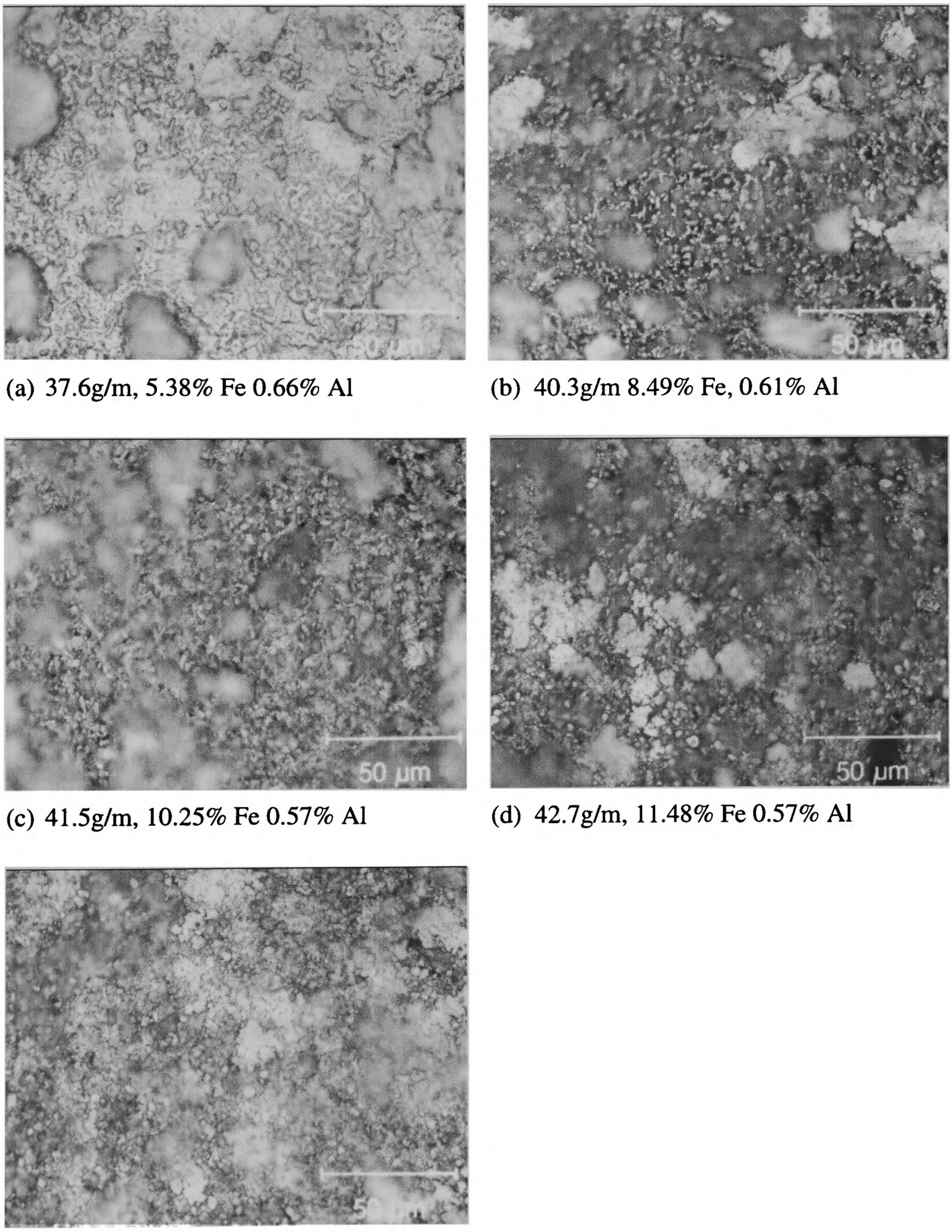
Table (18): Classification of Galvanneal Samples

Window Size (MxN)	No of Vectors Used	% Assigned to Correct Class (of 380 Samples, 5 Classes)
400x400	15	75
200x200	15	65
400x400	3	60

5.7.1.3 Noise Performance

Salt and pepper noise was added to the Brodatz set texture samples, and the algorithms run to classify them. In the context of the rank of the containment of the probe sets, this impulsive noise produces limited distortion of the histogram, but causes the raising of the outliers in the data (corresponding to no containment and full containment at a point). The algorithms were adjusted to ignore these outliers, producing a significant improvement in the performance of the classification scheme (Table (19)).

Figure (50): Galvanneal Samples, 512x512 Images at 8 Bit Resolution



(a) 37.6g/m, 5.38% Fe 0.66% Al

(b) 40.3g/m 8.49% Fe, 0.61% Al

(c) 41.5g/m, 10.25% Fe 0.57% Al

(d) 42.7g/m, 11.48% Fe 0.57% Al

(e) 51.2g/m, 13.64 % Fe 0.5% Al

Table (19): Noise Performance of Classification Based on the Use of 15 Feature Vectors

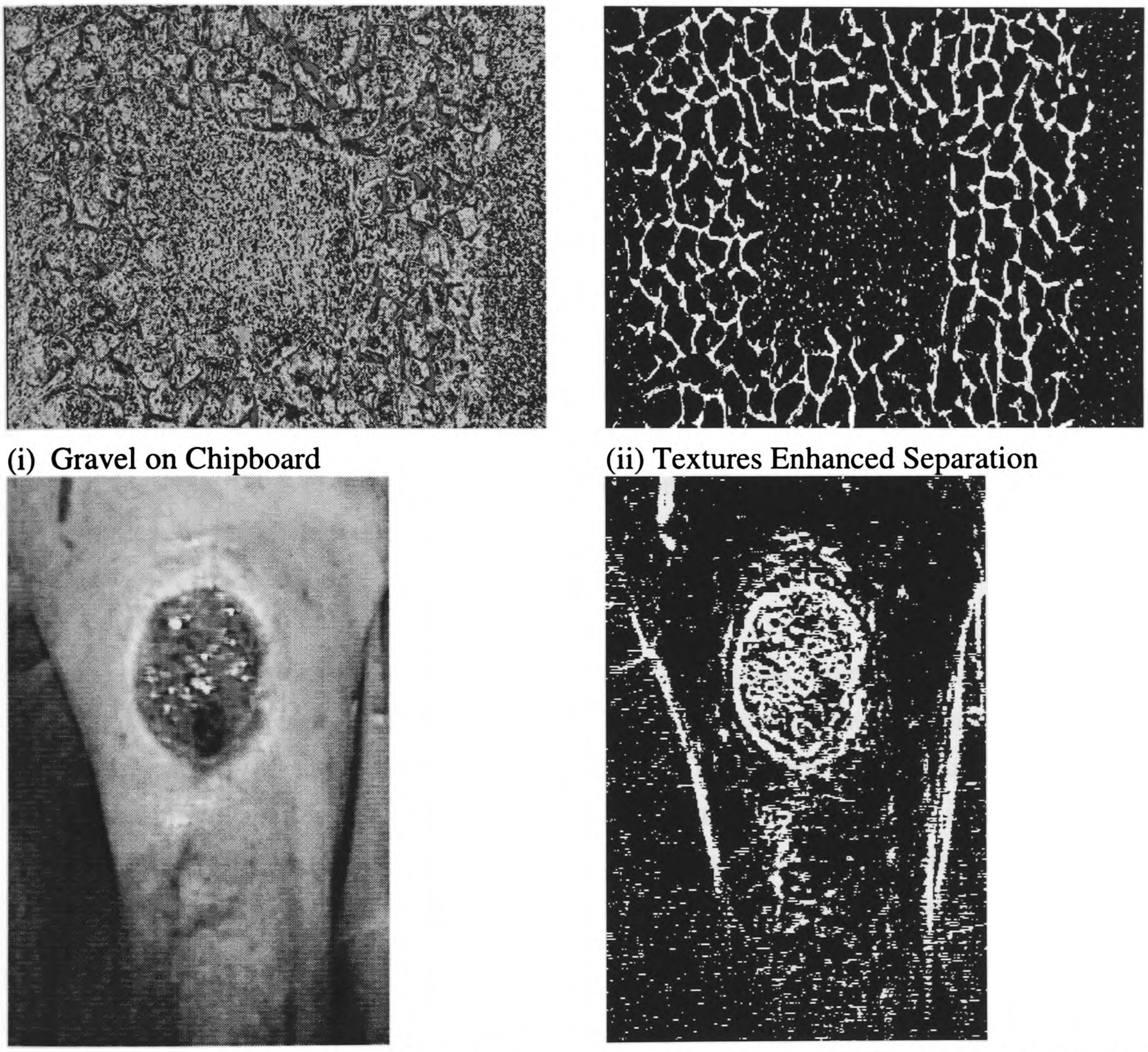
% Noise	Window Size	% Correctly Classified (of 19 Texture Samples)	% Correctly Classified on Removal of Data Outliers
0	150x150	100	100
1	150x150	90	95
2	150x150	79	85
3	150x150	63	70
4	150x150	52	60

5.7.1.4 Boundary Detection

In order to locate boundaries between textures, it is necessary to assign pixels in close proximity to the boundary to particular texture regions. The larger the probe structuring element is about the point, the more likely it is to overlap the "other" texture region. The approach used was to adopt a 96x96 pixel window size, and classify the regions initially on this basis. Where the classification metric exceeds the permitted variation, typically near a texture boundary, then the larger sizes of template were eliminated and the window size halved, down to a limit of 12 pixels window size. All remaining point are assigned on a nearest neighbour basis. With test data of the form shown in figure (51) below, the pixel classification was found to be accurate down to the minimum window size, and the system classified between 30% and 70% of the remaining pixels correctly, dependant upon the chosen boundary shape.

In practise, the use of the *R* operator is far more effective in enhancing textures for segmentation. By choosing which structures to eliminate, and applying an *R* conditioned operation (erosion, dilation, duet, or one of the structure detectors), the structural differences between textures can be increased. Figures 52 (i) - (iv) overleaf illustrate this for two situations, a natural gravel on chipboard example, and a soft tissue would perimeter enhancement

Figure (51): Textured Object and Background Enhancement



(i) Gravel on Chipboard

(ii) Textures Enhanced Separation

(iii) Soft Tissue Wound (u2b.img)

(iv) u2b.img (r,donut20{13,1})u2bd20b.clp

5.7.2 Discussion of Results

The texture analysis method developed has been successfully applied to a variety of natural textures. It is based on extraction of integer data, and the histogram feature data extraction is capable of implementation in hardware, based on a shift and compare algorithm on a pixel by pixel basis, comparison being undertaken over the chosen template shape size. Subject to the replication of the shape size area, the extraction of data is highly amenable to parallel implementation down to the single pixel level. The generation of a histogram, and hence a feature vector for analysis, remains essentially a serial task. However, if the initial R extraction is rapid, then the processing times will be acceptable in comparison with conventional techniques such as the GLCM approach.

For the very large volumes of data likely to be generated by an in-line classification scheme for galvanneal strip, then the use of mean and variance data will allow further localisation of clusters. This was not attempted with the roughly fifty samples per class allocated for off-line consideration.

Comparison with the work of Wang, Haese-Coat et al [19] in terms of noise performance and discrimination indicate a less effective result for equivalent window sizes. Their methods require the calculation of grey level mean and variance for each image calculated by the decomposition discussed in section 1.1, p16. They report results of between 93.52% and 99.07% classification success with one hundred and eight sub-images from nine selected classes drawn from the Brodatz set, based on a window size of sixty four pixels square.

The process adopted requires the successive opening (erosion, followed by dilation) of the image under test by small structuring elements. After each stage, the grey level means of the images are calculated. These form the texture features fed to a Bayesian classifier for analysis. The lowest risk classification is adopted (which is necessarily an optimal solution to the data). The work earlier in this chapter has achieved results of between 91.6% and 94% classification success with a window size of seventy two by seventy two pixels.

The lower classification accuracy is due in part to the non-optimal classification method adopted, and is compensated to some extent by lower computational intensity. The numbers of templates and densities of (pixel numbers within) templates are significantly lower. Given that the R approach does not require the generation of many intermediate images, the computational approach adopted could offer significant advantages for in-line usage. Their work is, however, much more suitable for segmentation of multiple textures within a single image due to the good performance for small window sizes. It is difficult to justify the use of a one hundred and thirty six pixel square window, iteratively applied for each pixel in the image, as a valid approach to rapid identification of texture grouping. Their approach does, as ours, include consideration of shape as well as size of texture, although in a measure closely linked to those of granulometry.

In general analysis and use, there is no practical method of regenerating the original image from the R results.

In practise, with the classification of Galvanneal samples, the classification scheme performed well in comparison to a fourteen feature grey-level co-occurrence matrix technique (see Haralick et al [69] for details). Typical results show that a performance of between 25 and 45 percent correct classification on a per-sample basis is likely with these features. Better results should be obtained by mean and variance analysis of the various features over a larger data sample. Application of Fourier spectra techniques to the same data produced comparable results, but with vastly greater computational burden [70].

As an illustration of the generality of the applicability of the template based feature classification scheme suggested, the above work is a reasonably convincing demonstration of its utility. In reality we are mapping a correlation between a template and the surface of the intensity map, as measured by the rank of point under consideration in its local surface.

5.8 Conclusion

The various methods described have been applied to feature detection for real images with some success. The nature of the objects used for testing result in multiple hits at significant perimeter features, as would be expected due to real profiles and sampling deficiencies.

The extracted data sets were evaluated against human visually extracted information, being those features regarded as perceptually significant by the test volunteers. The resulting features were quite close to the visually important points, but a relatively large number of points were not found. Typically the figure were of the order of distance 3 pixels from the human assigned point, and around 50% of the points found for the general set of perceptually significant features, with around 90% of the strongest, or control group, features identified. The implications of this for the recognition process will be evaluated in the next chapter.

Chapter 6: Results and Discussion for Object Recognition

Chapter 6: Results and Discussion for Object Recognition

6.1 Introduction

The recognition algorithms developed in Chapter 3 were implemented, and applied to feature data extracted, as shown in the last chapter, from images of the various objects. Freedom of rotation, translation and scaling led to a need to interrelate the extracted features, as is later described, to create comparison models with appropriate properties for recognition use. The resulting models were compared against reference model poses for recognition, and the identified features were used to extract orientation information from the object. The test objects used are relatively well separated, including the hawk trainer, the concord and the block model from the earlier work for 3-D recognition.

6.2 Set Erosion

The set erosion process is capable of isolating the loci of containment of a particular pattern within a data set. This is its primary function. The question of recognition then becomes one of identifying a suitable range of features to extract from an acquired image, and then applying a characteristic set of features from the library model of the required object as a probe set for the erosion process. If the feature set is correctly chosen, is stable under the image conditions, and is unique, then the full erosion result indicates the probable presence of the object within the image.

This method was applied early in the project as a strategy for recognition. As can be seen from the table of results below, it can produce high quality information. It became obvious that the method as applied required a highly accurate extraction of features from the original image. The counter measure developed to prevent this problem was to spread the lines in the structural pattern spectrum, thus allowing toleranced deviations within the scope of the library model. It was necessary to sum the side components within the same tolerance limits in the extracted data. The reasoning behind this is obvious: it requires a separation of 57 pixels to yield a 1 degree accuracy in orientation measurement of a straight line. Few features in a 512x512 image are likely to be of this order of length. In the examples used, the camera focus and field of view were optimised to ensure exactly this situation. The results obtained are shown in Table (20) below, and presented in more detail in the 1989 conference paper [37]. A simple uncertainty measure was included with the recognition information, along with rotation and scaling.

The set erosion method was effective, but was felt to be unwieldy. Its problems become more significant when applied to 3-D objects.

The perimeter definitions were later extracted by edge following and chain code generation. The simple 2-D shapes lend themselves very well to such an approach. As a result, the data obtained included both sequences of sides and types of features. The use of sequences of edge features, independently from the actual approximated loci of the features, provided a model capable of recognition under these elementary conditions. Rotation, and scaling, were extracted directly from the set erosion process, where the offset for alignment of the probe model set yields the rotation, and the ratios of the longest sides the scaling.

The techniques used rely on the extraction of adequate feature data. With 3-D objects, the problems of extracting accurate orientations for edges were found to be less tractable. The degree of inference required to generate adequate data increased significantly, and became unwieldy when added to the requirement for single line data.

Table (20): 2-D Object Recognition Results using Set Erosion

Shape	Recognized	Orientation			Scale		
		Actual	Calc	Err %	Actual	Calc	Err %
Square	y	0	0	0	0.95	0.96	1
Rectangle1	y	10	11	10	3.28	3.31	1
Rectangle2	y	20	22	10	1.48	1.50	2
Pentagon	y	70	70	0	1.77	1.75	1
Hexagon	y	4	4	0	1.00	1.03	3
Polygon	y	20	21	5	1.60	1.57	2
Chord	y	10	8	20	1.20	1.17	3
Trapezoid	y	324	322	1	0.95	0.97	2
Uconcave	y	10	11	10	0.97	0.95	2

Table (21): Characterisation of Occluded Shapes

Shape	Overlap (%)	Recognize	Orientation		Scale	
			Actual	Algorithm	Actual	Algorithm
Trapezoid	10	Both				
and Square		Trapezoid	-2	-2	1.80	1.70
		Square	0	1	1.80	1.67
Pentagon	15	Both				
and Square		Pentagon	2	4	1.00	1.05
		Square	50	52	1.80	1.65

The principal features of the object under test must be included fully in the extracted feature data for a successful recognition using formal set erosion. If this occurs, the technique is effective and efficient, using simple integer processing to obtain the result. Such a requirement has implications for the modelling technique used, in that the object model must include the various views of the object including the different combinations of principle features. As has been previously indicated, the 3-D structural features found most reliable under extraction were corners and perimeter details (which are most easily extracted when amenable to extraction as 2-D structures).

6.3 RJ Operator

The inherent capability in the RJ operator for handling less complete data proved a useful enhancement in recognition. The RJ operator is applied to the extracted feature set, the locus of best fit generated, and the corresponding J sets retained for analysis where a complete fit is not obtained. This is not a minimum distance classifier strategy as such, but shares the same basic pair matching philosophy. The locus of highest number of completely fitted components is used with the grey scale method; using the multivalued function version, the locus of the best number of completely and partially fitted components is used. The two RJ methods (greyscale and multivalued function) differ in the use of the presented data, and, as is shown in the following section, the latter proved more useful for the extracted data sets, and particularly with overlapping and partial obscuration in the objects under test.

The feature sets used with this algorithm were extracted using the methods illustrated in Chapter 5. The problem is one of classification of the given data. For the most part, the data illustrated has been extracted by different methods but at the same scaling. This is not generally the case. Part of the recognition algorithm requirement is to extract (and compensate) for scaling effects.

6.3.1 Use of Features

Having successfully extracted features from the test images, the next task is to produce a set of feature attributes amenable to recognition use. The modelling techniques used have been discussed previously (Chapter 3, section 3.3). As this work suggests, the relationships created from the features to produce the data set has a significant effect on its utility for recognition purposes. The extracted features were used to generate data sets of the types indicated in the following sections. Examples of the data sets are shown in Appendices B.5, B.6 and B.7.

The feature sets were pre-processed to a limited extent. A minimum separation limit for features was adopted, and adjacent features of the same type were amalgamated, limiting the likelihood of multiple responses from a single feature, and sets of concave and convex features were added to produce the full description of the object. The different extraction techniques used were then compared, using the accuracy of recognition as the primary basis of evaluation.

6.3.1.1 Lumped Model

This was generated simply as the sum of the occurrences of high convex curvatures and high concave curvatures observed within the data. As such, it is an exceptionally crude model, and obviously highly dependent on a quality initial image. It is prone to missing features, and easily aliased. The recognition process does not need to use the R operator: a simple subtraction of counts provides adequate interpretation. The uncertainty of extraction with the R operator methods with acquired data makes such an approach untenable.

Model = $\{n_{convex}, n_{concave}\}$

6.3.1.2 Modelling by Silhouette Perimeter Sequence of Features

The model was generated from the extracted data by two separate methods: conventionally, by edge following, chain coding and curvature estimation; and using the R operator. The former method allows extraction of silhouette perimeter feature sequence information as a natural result of the technique. In order to obtain sequence information from the R approach, the centroid of the extracted features was calculated, and a sequence generated by orientation estimation in an anticlockwise direction about that point.

$$\text{Centroid of Features } (x_c, y_c) = \sum_0^N \left(\frac{x_n}{N}, \frac{y_n}{N} \right), \text{ where } (x_n, y_n) \in F$$

$$\text{R Model } = \{(type_1, \theta_1), ..(type_n, \theta_n), ..(type_N, \theta_N)\}, \text{where } (\theta_1 < \theta_n < \theta_N)$$

6.3.1.3 Modelling by Web Skeletons

The model was generated by producing a list of the orientations of all the other features within the acquired set from each feature in turn. Although this may seem exhaustive, given the high likelihood of missed features, such an approach maximised the probability of a correct recognition.

$$\begin{aligned} &\text{Model } = \{(type_n, \theta_{nm})\}, \forall m \in F, \forall n \in F, \\ &\text{where } F = \text{ set of extracted features}; m, n \in F. \end{aligned}$$

The data sets were compared with library models created from data extracted by eye, using the RJ operator. The results shown below were obtained.

6.3.2 The Results

The tables in the following sections show the results obtained for each of the modelling methods, using features extracted from the images of Chapter 5. The data is presented in two forms, firstly for a direct extraction of features with the donut series templates (annulus height of five grey levels), and then with further optimisation of that threshold. The added benefit obtained from the use of the *R* operator is the ability to use the knowledge of the probable identity of features as a guide to the estimation of roll and yaw. Where available, both convex and concave features were used.

The quality of the initial feature extraction is of paramount importance to the generation of successful, accurate recognition data. The increased sophistication of the modelling techniques has benefits, and permits recognition with less certain feature information.

6.3.2.1 Direct Extraction using the Donut Series Templates

The results shown below were based on unprocessed extraction for the *R* operator based methods, relying on data with no pre-processing. Fixed geometric and gradient thresholds were used, in this case based on the donut template series (perimeter values of five grey levels) and containment of between 2 and 5 pixels for convex features, and of between 10 and 15 pixels for concave features. This represents an attempt at a direct automation of the recognition process for the images used.

The chain coded edge following method requires binarisation for use, and the series of results for this method are generated after the required processing has occurred.

The results are adequate, but produced problems for the objects of low perimeter curvature (such as the block), and excessive feature generation for the higher contrast object (the hawk trainer) for convex features.

Table 22: Recognition Results using the R Operator with Sequenced Silhouette Perimeter Data, Not Optimised for Probe Template Gradients

Object	**Recognition of Object/Pose by Indicated Method**			
	Edge Following/ Chain Code	**Mostly Hit, Mostly Miss Transform**	**Intersected R Operator (Rin)**	**Direct R Operator with Edge Following**
concord	y/y	y/y	y/y	y/y
rotated concord	y/y	y/n[1]	y/y	y/y
hawk trainer	y/y	y/y	y/y[2]	y/y[3]
block[4]	y	n	n	y
rotated hawk trainer	y/y	y/y	y/y[2]	n/n[5]

[1] Found non-rotated pose.

[2] Working with concave features only.

[3] Chain code generator produced description of adequate portion of boundary.

[4] Symmetrical in each axis - the pose question is not sensible as only one pose exists.

[5] Excessive generation of spurious results over the tail portion due to low contrast.

Table 23: Recognition Results using the R Operator with Web Skeleton Data, Not Optimised for Probe Template Gradients

Object	Recognition of Object/Pose by Indicated Method			
	Edge Following/ Chain Code	Mostly Hit, Mostly Miss Transform	Generalized R Operator (Rin)	Direct R Operator with Edge Following
concord	y/y	y/y	y/y	y/y
rotated concord	y/y	y/n	y/y	y/y
hawk trainer	y/y	y/y	y/y	y/y[1]
block	y	n	n	y
rotated hawk trainer	y/y	y/y	y/y	n/n

[1] Insufficient description of boundary, low confidence in result

6.3.2.2 Orientation Estimation

The calculation of yaw, pitch and roll is greatly simplified by the predetermination of object type and approximate pose. The acquired features represent known points of the object. The choice of likely wing tips or nose and tail, for example, is a trivial task by reference to the acquired and model data, presenting the ideal information for calculation of orientation.

For the results of table 24 below, the scale was fixed and the objects compared with the stored model for the pose. The orientation relative to the model pose was calculated.

As has already been observed in section 5.6, the accuracy of the determination of feature loci in the extraction process then determines the resolution of orientation. This is only partially true. The separations of the features used for orientation estimation are additional sources of error. The deviation is, therefore, indicative of the orientation error inherent in the adopted process. The use of a single view of the object greatly exacerbates these problems, in that many of the poses have the object such that its principal axis is orthogonal to the view. This inevitably increases the errors found.

Table (24): A Comparison of the Deviations between True and Estimated Orientations of Objects with Various Extraction Methods

Object	Deviation in Yaw/Pitch/Roll with Indicated Method (Degrees)			
	Edge Following/ Chain Code	Mostly Hit, Mostly Miss Transform	Generalized R Operator (Rin)	Direct R Operator with Edge Following
concord	-1.7/22/25	0.4/15/error	0.6/0/0	2.1/14/21
rotated concord	0.7/18/22	0.3/10/error	0.9/5/10	4.5/16/12
hawk trainer	-1.2/19/16	0.9/10/14	not available[1]	not available[2]
block (principal axis)	9.4	not recognised[2]	not recognised[2]	6.2
rotated hawk trainer	-0.4/21/16	0.5/9/10	not available[1]	not recognised[2]

[1] Generated by concave features only, offering limited base for estimation.
[2] Insufficient description of object in extracted features. Pose information only.

The reliance on point locus estimations for regions of high curvature contributes to the overall error in the method. The perimeter following, chain code method of delineating features used a difference method for estimation of orientation. The window size for the window calculations is then a parameter in the accuracy of delineation. A smaller window offers a more accurate position estimation, but at the price of higher susceptibility to noise and lower reliability for larger, structurally more significant, features. In practice, a balance is chosen, which may, as in this case, yield significant errors in position estimation but offer good set of structural features for recognition.

The mostly hit, mostly miss method offers a better resolution of position. The extraction technique is based on a smaller window, this being the size of the template used. It is, however, prone to elimination of valid features, resulting in difficulties with the alignment of structures between the model and the acquired data.

The *R* intersected method is also based on templates. The size of the templates is a function of the structures sought, but, in general, the smallest of these will be more localised than is practicable for the chain code method. It is this smallest template that defines the localisation of the sought features. The problem here was more one of insufficient selectivity, rather than the over-selectivity of the MHMMT. This method produced accurate results, but would require the post-processing of feature data to reduce the feature count, particularly in the convex structure case.

The views used were not beneficial to the estimation of roll and yaw. The top view of the objects used initially for the concord and hawk trainer models allows the worst case estimation of roll and yaw, relying on the apparent change in separations of the features used. Any error in the position estimation of the features has a marked effect on the apparent orientations under these conditions.

6.3.2.3 Recognition with Optimisation of Probe Template Gradients

The results of tables 25 and 26 following illustrate the benefits obtained from optimising the detector for the object features to be extracted and adjusting the gradient threshold. This was undertaken automatically, the gradient adaptively set to that of the nearest of the donut *nn* series template gradients to the average grey level of the image for single object images. For the test images here, this sets the threshold at between one third and one half of the average object grey level in the image.

Table (25): Recognition Results using the R Operator with Sequenced Silhouette Perimeter Data, Optimised for Probe Template Gradients

Object	**Recognition of Object/Pose by Indicated Method**			
	Edge Following/ Chain Code	**Mostly Hit, Mostly Miss Transform**	**Intersected R Operator (Rin)**	**Direct R Operator with Edge Following**[1]
hawk trainer	y/y	y/y	y/y	y/y
block[2]	y	n	y	y
rotated hawk trainer	y/y	y/y	y/y	n/n

[1] Results unchanged - data already uses optimised gradients and pre-processing

[2] Symmetrical in each axis - the pose question is not sensible as only one pose exists.

The concord and rotated concord information was unchanged - in the context of the recognition algorithms used, the features extracted in the initial tests were near optimal for these images.

Table (26): Recognition Results using the R Operator with Web Skeleton Data, Optimised for Probe Template Gradients

Object	Recognition of Object/Pose by Indicated Method			
	Edge Following/ Chain Code	Mostly Hit, Mostly Miss Transform	Generalized R Operator (Rin)	Direct R Operator with Edge Following
hawk trainer	y/y	y/y	y/y	y/y[1]
block	y	n	y	y
rotated hawk trainer	y/y	y/y	y/y	n/n

[1] Insufficient description of boundary, low confidence in result

Table 27: A Comparison of the Deviations between True and Estimated Orientations of Objects with Various Extraction Methods

Object	Deviation in Yaw/Pitch/Roll with Indicated Method (Degrees)			
	Edge Following/ Chain Code	Mostly Hit, Mostly Miss Transform	Generalized R Operator (Rin)	Direct R Operator with Edge Following
hawk trainer	-1.2/19/16	0.9/10/14	0.0/0.32/14	not available[2]
block (principal axis)	9.4		6.8	6.2
rotated hawk trainer	-0.4/21/16	0.5/9/10	0.5/9/10	not recognised[2]

[1] Generated by concave features only, offering limited base for estimation.

[2] Insufficient description of object in extracted features. Pose information only.

The results presented in tables 25 to 27 above show clearly the benefits of optimisation of the probe template intensity gradients for the extraction of features. The basic adaptive technique used allowed reasonable results with single object images with relatively low noise backgrounds. A more sophisticated measure would be required for use under live data conditions with low constraints on lighting and picture contents. The adaptive optimisation of gradient thresholds for the probe templates for complex images (multiple objects, noisy background) was not studied as part of this work.

The accuracy of the orientation measures used relates directly to the accuracy of location of the extracted features. The performance of the algorithm in noise can, therefore, be assessed in terms of the accuracy of feature location, as is shown in figures (43) and (46). The accuracy of location of features is quite resilient in the presence of pixellated noise, but the increasing numbers of false hits offer potential problems for the recognition process itself.

6.4 Evaluation of the RJ Operator for Classification of Extracted Feature Sets

The problems of feature extraction, and their effects on the recognition process, having been discussed, the following sections compare the RJ operator as a recognition technique with standard measures.

6.4.1 A Comparison Between the RJ Operator Method and the Minimum Euclidean Distance Classification

The basic difficulties are the same: given a partially complete, possibly flawed description of the acquired image object features, the best fit between the library models and the object must be obtained, and evaluated for its accuracy and completeness.

With the minimum distance classifier, the library model set is aligned with each of the features in the acquired set, and an evaluation of the separation of the features undertaken. This evaluation is in the form of two numbers, the sum of the distances between the elements of the feature vector, and the square of distances, thus allowing for cancellation between positive and negative variations. Each feature distance should be calculated to its closest dual, and, given that a particular feature can only be matched once, it requires some effort to produce the minimum result. The result is inherently tolerant of missing features - they have no particular significance in this approach, and might be included as a separate count or as part of the overall distance calculations. Note, however, that the inclusion of wholly missing features within the count necessitates a high quality feature extraction methodology, to prevent the rejection of correct, but sparse, responses.

This is not so with the R operator method. Like all erosion-based techniques, this produces a result conditioned solely on the presence of features. The feature vectors are shifted and intersected, and it is effectively the separation between feature vectors that is the determinant of quality of fit. In computational terms, the model set is shifted by each element of the acquired set, and the resulting partials intersected. This yields a computational cost for shifts equal to the product of the numbers of features in acquired and reference sets, the intersections then being a subtraction and comparison for each element in the first partial. (A little thought will show that the most commonly occurring partial will be one of those present in the first shifted set).

The result is a count of the components present in both the acquired and the model data sets. The best fit from the library is initially determined by choosing the result with the highest R value. It is necessary to introduce a variation about the ideal feature locus, to accommodate the predictable variations in the feature position inherent in its extraction. The J set corresponding to the best fit R contains the detailed differences between the feature vectors.

This similarity measure is based on found and partially found components of the feature vector, rather than on minimising the alignment differences between the components in the two sets. It would be expected, therefore, to be less likely to produce a spurious identification, but more susceptible to errors in the feature loci where the image is distorted or noisy. The RJ operator will produce a more reliable result in a feature dense environment.

We might implement a form of minimum distance classifier by nearest neighbour measure between each of the complete components in the J set, indicative of completely missed components in the acquired data, and its nearest dual in that data. This should give a similar result to implementing a distance classifier with the additional constraint that a component of the model must align with a component of the acquired set for each locus at which the distance is calculated.

6.4.2 Evaluation of the RJ Operator for Feature Set Classification in Comparison with the Hough Transform

Several of the problems Grimson [72] discusses with the use of the Hough transform are equally applicable to the technique used here. Recognition in both systems relies on the matching of elements of the object model with instances of those elements in the extracted feature data. The Hough approach generates ranges of transformations consistent with feature pairs. Each transformation is instanced as an increment in the "bucket" for that particular transformation. The transformation adopted as the solution is the highest bucket count. The RJ operator method positions the model feature set at each position in the extracted feature set and evaluates the level of containment. Missing features are indicated by the contents of the J set for the positions of maximum containment. The likelihood of a particular position, pose or orientation (equivalent to the Hough transformation) is indicated by the R value, much as the Hough maximum probability is indicated by the largest bucket value.

The size of the Hough transform bucket accumulator array is acknowledged as presenting problems for storage and speed of search [73].

In a feature rich environment, the process of generating possible object instances from isolated vertices has a considerable likelihood of misclassification of clusters of vertices as objects, as there is from clusters of sides. The likelihood increases as the number of vertices increases, and also as the degree of uncertainty about loci of features is permitted to increase. This presents problems in a multiple object or a texture-rich environment. The limited set of feature types chosen exacerbates the problem. This is an unavoidable flaw in the chosen process. The use of confirmatory data such as separation of features as well as relative orientations can allow additional confidence in the result.

Unlike the Hough transform, the usual applications of which rely on clustering many lines into a match, the process used here relies on points of high curvature. These are the end points of lines in the Hough terminology. As a result there is no possibility of the generation of multiple hits by a line sliding along its own length.

6.4.3 Tightening the Response

The discussion above indicates one of the methods of tightening the response of the RJ operator for recognition purposes, that of including feature separation in the evaluation of a response. Given the requirement for loosely constrained scaling, it is vital that such an evaluation should occur only once an apparent match has occurred.

The other method of reducing the likelihood of false hits is to tighten the requirement for correct feature orientation, by reducing the permitted deviation from the required relative positions. Such an approach requires fundamentally good source data, offering a better input image and hence higher quality vertex location, coupled with a close proximity between the probe and object intensity gradients and optimal choice of geometric thresholds. This would certainly require additional poses for each object model, the number increasing as the orientation specifications are tightened. Ultimately, constraints as to lighting, position and rotation would be imposed, taking the situation outside that the work was aimed at supporting and into the area where simpler techniques might provide a better solution.

6.5 Number of Poses Required for Recognition of an Object

The number of poses stored varies between objects, primarily dependent on their structure and complexity. The rotation of the object brings different structural features into view, while others are obscured. These are compound effects, roll, pitch and yaw all affecting the single view appearance. The aim must be to provide a sufficiently rich set of poses to ensure that a limited set of features will consistently recover the required view.

As a guide, for the objects used in this section of work, the number of views required varied between three for the cube to twenty four for the Hawk trainer. These were evaluated by determining the unique poses in the feature sets.

The set of stored poses for the objects was generated by eye. As such, it contains an inherent set of assumptions about the priority and utility of certain features. As has been discussed in Chapter 5, there is considerable similarity between the perceptually important points of high curvature and those extracted by the algorithm, but a complete contiguity is not found. A more objective procedure might offer a better linkage to the algorithms' strengths. It is likely that a steady rotation coupled with a continuous tracking of the object features will enable the production of the richest set of poses, based on the inclusion of a new model as new features appear or are obscured. This set of pose models would later be minimised for storage, eliminating unnecessary versions. However, adequate equipment for such a test was not available at the time of writing.

It should be noted that further compression would be obtained for limited object domains, where differentiation from other objects is more significant than producing its best description.

6.6 On Occlusion

Recognition can be achieved with partial obscuration of the object, subject to the constraint that a sufficient body of evidence about the object exists to give a unique match. The reconstruction of the object boundary is not a simple task given the method of recognition adopted - it is feasible to identify the missing vertices, but this yields insufficient information about which portion of the boundary is absent. The process then becomes one of aligning the full object image for a pose, stored as a library model along with the vertices summary, with the apparent position of the pose in the image, and identifying the missing portions. General comments of the order of "a wing is missing" (see figure (21)) can be made, but a high utility for reconstruction is not a feature of the approach. This is an area where the line fitting approaches offer genuine superiority of performance.

For fixed scale, recognition was achieved reliably with up to thirty percent of the object removed. It is difficult to draw a general rule for this situation - if the area removed is contiguous, and carefully chosen then recognition could be claimed at levels of even eighty percent removal. Common sense, however, mitigates against such claims. The requirement is simply that enough of the characteristic features of the object remain in the residual view. The degree of obscuration is only relevant in its relationship to this requirement, and depends wholly on the object under analysis.

In order to provide an estimate of the utility of the RJ operator method under obscuration (or missing) of features, a game based test was derived. The algorithm was run many times on data from which a fixed percentage of the features were removed. The features were randomly chosen for each test from the acquired data for the Hawk trainer IS6, including false hits where they occurred in the noisy extraction process. The data set was then assessed against the library model poses for all the objects with a requirement of the highest occupancy being set as indicating the best result, and therefore the chosen object. The results are shown in Table (28) below, for 60% added pixellated noise.

Table (28): Recognition with Partial Object Features under 60% Pixellated Noise

% FEATURES REMOVED	% RECOGNISED
10	100
20	100
30	92
40	84
50	72

6.7 Limitations of the Algorithms as Implemented

The algorithms as currently implemented are essentially "success orientated", in that they are designed to locate the best fit of the given model within the acquired data set. This poses several problems when the more general problems of, for example, the bin of parts identification task, where the possibility of overlap and significant obscuration are likely to occur. The feature rich environment engendered will offer relatively high containment of various models, whilst the partial occlusion of the objects will conceal many of the features of the objects sought. The likelihood of false identification, or of failure to locate the relevant object, is high.

The algorithm is not designed to identify several instances of the object, for repeated application to the residual features after each more confidently found version is accepted and removed from the feature world. The current implementation will simply take the highest confidence locus as the place of the object (singular) within the image.

In order to improve the response under these conditions, the fixing of scale and the formal inclusion of feature loci in the library model, as well as their relative orientations and separations, offers one possible path to success.

The model world used for object recognition was relatively small. This inherently reduces the likelihood of multiple instances of very similar feature sets appearing in the model library, and consequently reduced the risk of the acceptance of two separate views as equally valid for the particular acquired feature set. This problem of uniqueness of view occurs for many recognition systems. The resolution of apparent recognition contention would require full use of the ***J*** sets data, where the added constraints of consistency with the missing features could offer a directed search to locate data to support the hypothesised match.

6.8 Comparison with Previous Work

There are two major methods of identifying object and orientation using this type of structural feature approach. Both methods use the object silhouette as the primary source of information, when working from single views. The points of high curvature may be identified, and used as indicators of the object structure. Alternatively, the areas of low curvature which separate them are used, either as aligned segments of boundary or to locate the areas of high curvature between them.

Two methods based on the identification of feature points were adopted by Chien and Aggarwal [44] and Illing [33]. Chien and Aggarwal reported the use of k-curvatures to identify the points of high curvature. Their location offered an apparent error in orientation estimation of better than 5 degrees. With images of objects under noisy conditions, causing up to 60% of boundary pixels to be perturbed, Illing reported orientation errors of up to 15 degrees, with an average uncertainty of around 7 degrees, through the use of boundary section alignment.

The worst case figures found for orientation error in any plane with the R operator methods was of the order of 16 degrees. This resulted primarily from errors in the location of features during the extraction process. Errors of up to 25 degrees in orientation were found with the chain coded method. The ability to recognise objects is comparable with these other techniques. The relatively few views required for each object offer computational benefits in use.

Pitas and Venetsanopoulis [16] describe the use of decomposition by morphology for the classification of objects. This is based on the use of repeated openings to generate a paradigm for a view of an object, being the largest size of a basic shape (circle, rectangle, square, triangle) it can contain. The sum of the component parts so defined forms a reasonable description of the original shape under decomposition.

Its application to pose based model recognition is fairly straightforward. The method has some of the benefits of global feature based systems: it can deal with large structures in a single feature; it can offer a good level of noise resilience; but, unlike these algorithms, it can handle some degree of partial obscuration provided that a syntactic or graph-like approach to the layout of the primary shapes is adopted. Under obscuration, the deviation of the features used by this method will be greater than that with the vertex based approaches.

The method requires the repeated opening of the image, and as such can be computationally intensive. The number of operations is indicated in the paper to vary with the order of the number of pixels in the object in the image, which seems overly optimistic. The object is pre-processed, segmented and binarised, and the problems these stages entail are not considered in the original work. The same amenability to parallel implementation occurs here as with the RJ operator approach, but with a larger degree of data replication due to the whole image dependency of the shape paradigms, and the sequential nature of the application of the opening process. This may offer problems with hardware implementation.

Its performance under pixellated noise (described in section 5.6) is better than that of the vertex based methods, due to its reliance on the bulk of the view and not its outliers. It cannot, however, equal the full precision of the vertex based solutions in high quality image conditions. The separation of vertices is necessarily greater than that of the centres or control points of the paradigm shapes used in decomposition.

Liu and Srinath [71] use the smoothed curvature of the perimeter contour to isolate the vertices of curved objects, which form the control points for the object, breaking the boundary into straight line segments. They state that a target of forty segments is appropriate. A two pass matching process first matches the pairs of segments, then possible groups of segments. The distance measure is based on the chamfer ¾ distance transformation, amended as a partial distance for each segment. Full classification of the object set with up to twenty percent distortion, for shape scales between 0.8 and 1.2 times the size of the shape under test, are reported. The method suggested, based on side alignment, is far more susceptible to the missed vertex problem than the RJ operator solution. A single missed vertex for an outlying feature will contribute disproportionately to the distance measure, sharing similar difficulties with the artificial centre of gravity sequence of features method we have described earlier in section 3.3.1.2. The results obtained offer little greater accuracy of vertex location than the chain code method we have described.

6.9 Conclusions

The primary objective of the project was to produce an object recognition system based on morphological methods. The modified techniques used have proved successful for the recognition of objects, but required the use of non-morphological techniques for the generation of models appropriate to the use of the data, particularly under free rotation and scaling. This, as a result, fails the objective of maintaining a purely morphological approach throughout the algorithmic architecture. Some other processing is required to produce models used for recognition, although this processing is limited. The recognition process, itself, is once again based on the RJ operator. If rotation and scaling are controlled, or known, a purely morphological architecture is possible. Pre-calculation bears the same problems as sophisticating the use of extracted features - these require conventional processing operations.

Objects within the model set were recognised, and their orientations identified to within fifteen degrees in the worst case. Recognition under noisy and partially obscured conditions was achieved, subject to the requirement that a sufficient portion of the boundary carrying areas of high curvature remains to form a classifiable data set. The results are comparable to those obtained by other methods.

A high degree of isolation of processing is achieved in the computation intensive recognition part of the scheme, provided the source data (the extracted feature set) is replicated for analysis. The ability to use the same architecture for extraction as for identification is a significant benefit of this technique.

The recognition results obtained through the use of the multivalued RJ operator are adequate, and accurately map the feature set to the model pose. This offers benefits in ease of calculation of orientation. The models respond well to missing features, offering a reasonable recognition capability with partial obscuration. The reconstruction of the missing boundary requires a different approach, as the vertex based method adopted offers only limited information about the area lost.

Chapter 7: Conclusions and Suggestions for Future Work

Chapter 7: Conclusions and Suggestions for Future Work

7.1 The Operators

The operators developed as part of this work broadly achieve their initial objectives. The R conditioned erosion is tolerant of defects in the data set under test, where formal set erosion is not. This permits defined levels of containment for the existence of a given element in the result set for binary sets, or a given function value at a result location for grey scale applications.

The R conditioned erosion operator is equivalent to the application of an "umbra only" soft morphological filter with a multiple response condition, and a minimum value calculation over the template area where the condition is not met. The effects of this are perceptually different: the R conditioned erosion tends to enhance the structures selected by the combination of probe structuring element and geometry limits more rapidly than do the soft filters, and retain the ability to generate a form of mostly hit, mostly miss transform and a meaningful multiple probe result intersection. These have proved useful for feature extraction and object recognition.

The grey scale R conditional erosion grows the dark areas of an image in much the same way as formal erosion, but avoids the "blocky" nature of the result and the inherent destruction of all fine detail (smaller than the applied structuring function area). This may offer benefits in the preservation of perimeter fine detail in pre-processing. It also permits the removal of full range salt and pepper noise with up to 30% contamination in the image through opening or closing. The method is perceptually superior to formal opening in that sense, and achieves a similar success to the soft morphological filters (but with a less smoothed result). Where the objective is to modify the structure of the image, to enhance specific structures in the intensity map, the conditional filters produce greater variations between iterations of the algorithm than do the soft filters for equivalent conditions. They can preserve structures under transformation to a superior degree, particularly where those structures form (for example) the boundary of an object in an image.

7.2 Feature Extraction

The methods developed work, and provide a performance equal to that of the standard approaches reported in the literature. Corners are identified to within three pixel positions in up to 20% salt and pepper noise with a reliability of 60 - 70%; with Gaussian noise, at a signal to noise ratio of ten, they were identified with a reliability of 72% for the given target geometry. The direct feature extraction method thus offers a better performance than the Dreschler-Nagel and Kitchen-Rosenfeld corner detectors, and rivals the performance of Haralick and Zuniga's best facet detector.

These results were achieved with no pre-processing. This avoids the problems of pre-processing distorting the object perimeter, but does reduce the proportion of vertices found.

The results with pre-segmented data displayed a high resistance to pixellated noise, achieving a 92% recovery of key features, under greater than 100% noise (perimeter pixels moved by up to two pixel locations) within five pixel locations of their true loci. Figures of 72% successful identification of features within 10 pixels of their true positions were achieved for 27.6% volume distortion. The test images reflect likely result levels with binarisation of a noisy image, and with an overly pre-processed situation. False responses were generated, as were multiple responses to true vertices. The algorithms perform comparably with Illing's line fitting algorithm under noisy conditions. The level of missed and false vertices increased as the noise level increased.

The elimination of pre-processing allows a less sophisticated algorithmic solution, which can work under a variety of conditions without necessarily requiring changes to produce adequate results. The problem of lighting is fundamental: if the object is inadequately lit, the extracted data will not contain enough information to permit feature extraction to occur. Whilst not overcoming this problem, the solutions suggested offer scope for considerable variation in lighting conditions.

The probe sets used were relatively sparse. This is particularly important, given the computational requirements of the algorithms. The shape of the probe sets was selected for pragmatic reasons, and to cater for rotation of the object within the image. It must be noted, however, that the solutions adopted were prone to over-selectivity (MHMMT), and to inclusion of excessive detail when attempts were made to eliminate this problem (intersected R operator) unless some adaptive method of calculating the template gradient is adopted.

The output produced was in the form of discrete locations of points of high convex and concave curvature. These features were relatively simple to identify about the perimeter of the object boundary (silhouette), where the contrast change was typically at its greatest level. In-object features were less amenable, because of the risk of interpreting specular reflections and rapid shading changes as structural features in the intensity map.

Gross structural features are used in the recognition process (nose cones, wing tips), which offers benefits in terms of the reliability of extraction. These types of features were found with a reliability of around thirty percent greater than the more subtle perceptually significant features. This is important for the identification of objects under rapid aspect change, where a rapid isolation process must link with a reliable identification for successful recognition to occur.

The generality of the solution, for a variety of feature types, is demonstrated by the texture classification work. This achieved results comparable to much of the ongoing work in this area with a non-optimised analysis based on the histogram of the R values for the texture image. The relationship of both size and shape to textural analysis may offer a useful tool in this area, with much further development.

7.3 Recognition

The choice of structural features as local descriptor was appropriate to the task, and allowed recognition where the object boundary was incomplete. This imperfection in the extracted data sets resulted from noise, missed features and partial obscuration. The need to interrelate the features resulted from the need to allow interpretation of objects under rotation and scaling changes. Where rotation and scaling were fixed, a fully morphological solution was obtained.

The RJ operator approach is capable of producing a good recognition resolution, but this is utterly dependent on the feature model used. Using the web skeleton approach, all objects were recognised, both as to type and pose, and to orientation where the feature extraction methods were optimised in terms of response. This method is less susceptible to missed vertices than the method of Liu and Srinath, which rely on line segment matching from the joining of vertices and are therefore prone to error with missed data outliers.

In the presence of clutter, caused by poor segmentation or multiple objects, the RJ operator offers a better resolution of object than the minimum distance classifier. Again, this is due to its resilience in the presence of missed elements in the data set. The performance appears to offer similar qualities to the Hough transform methods for equivalent data. Hough methods are often used with line and curve matching algorithms. This sort of data can yield multiple hits where a line crosses a quantisation boundary in terms of orientation, and also where a line appears to slide along its equivalent where scaling is uncertain. These problems do not occur for vertex matching techniques, but are replaced with the equivalent problem of multiple responses for true vertices. Both problems lead to false recognition.

The memory requirements are, however, markedly lower for the RJ operator technique. Its performance should also be significantly faster for reasonable model library sizes and volumes of extracted features.

The calculation of orientation is undertaken after recognition has occurred. This offers benefits to the complexity of the calculation. The basic limit on the accuracy of the estimation are the resolution of the sampling grid (the imaging array), the separation of the most distant identified features within the view, and the accuracy of location of those features. The results obtained were comparable with the work of Illing, Liu, and Rosenfeld and Johnston.

No attempt was made to optimise the code used for identification in this project. Typical recognition times ranges between two and ten seconds for the basic PC equipment used. Given that the majority of the code was sixteen bit based and generated under the debug build option, this could be improved significantly even for the basic serial implementation.

7.4 Architecture

The architecture achieved is predominantly morphological in nature, and purely so as far as the computationally intensive tasks of feature extraction and model matching are concerned. The major benefits of that architecture are therefore available to exploit, in the form of an efficient implementation on a cell-based parallel basis.

This needs some development to clarify the benefits. The data flow is independent, within the identification stages, of the other cells in the architecture - there is no cross-feed of data required. As a result, the cell structure could be replicated down to the level of a single input (output) pixel, gaining the highest possible benefit from parallelism (at a considerable silicon cost). The complexity of each cell is the same; co-ordination problems occur only at the interpretation stage.

Initial study shows the possibility of implementing a single cell with between one and two thousand logic gates. This compactness offers the possibility of implementing many such cells on a single semi-custom device; a single circuit board might contain many hundreds of cells. Allowing for the inevitable data replication problems, high levels of acceleration for the recognition process might be achieved. Provided the architectural problems of data replication and post-interpretation can be handled, this may well offer a progress path towards the four hundred frames per second processing goal.

7.5 Future Work

There are areas of this work which need development to reach fruition. The study does not treat in depth the use of the conditional operators for conventional image processing and enhancement tasks; the feature extraction methods might be enhanced to include non-structural data (colour being an obvious candidate for inclusion); the reliability of the feature extraction method under loosely controlled lighting conditions might be enhanced; the texture analysis work is incomplete, requiring considerable additional work in optimising the use of the extracted texture features and detailed consideration of the meaning of those features in the context of the texture itself; and the repeated application of the RJ operator to scenes involving multiple objects needs further consideration.

The selection of template gradient for feature enhancement and extraction could certainly be improved. The simple technique adopted here is not applicable to multiple object scenes where the level of background is largely variable. Initial studies of distribution based techniques risk losing the inherent benefits of a mostly morphological architecture, but may well be essential to high quality feature extraction. The established alternative, that of in-line equalisation and binarisation, could offer a solution to the problem, and would simplify the choice of probe structure to a discussion of appropriate geometric thresholds (not gradient thresholds).

The classification of galvanneal samples using this technique will require the improvement of the source images (the levels of poor focus caused by the degree of magnification are a serious obstacle to progress) and the optimisation of the data for classification, perhaps by a structural enhancement using the conditional operators. Analysis of the characteristic features of each texture might offer a means of discovery for a suitable implementation path. ANNs offer a possible alternative to the distance classification for extracted texture feature analysis. For line use, with strip material travelling at greater than ten metres per second, the only viable solution for control purposes would rely on hardware implementation of the algorithms. The architectural necessities are discussed in 7.4 above, and need no further dilation here.

There is scope for the use of the J sets for identification of missing portions of objects, and hence for the application of the RJ operator method to multiple object scenes. This work requires a wider consideration of feature types, the inclusion of feature location, and hence scale normalisation, for successful application.

7.6 Conclusion

The application of local structural features (vertices, points of high convex and concave curvature) to object recognition offers a viable method, which is robust under noise and partial obscuration of the object. The problem is then one of how to extract these features, and then to interrelate them in a manner tolerant of incomplete data and spurious features.

The basic problem of reliable feature identification under loosely controlled conditions remains a major difficulty, particularly in the context of rapid feature access and real time processing.

Interrelation of vertices in the form of a web skeleton for the object pose was found to offer a reliable recognition schema, which did not require massive computational effort and yet enabled identification of the pose, simplifying the later calculation of orientation relative to the library model.

The work undertaken has demonstrated the utility of a relatively simple measure for the determination of the occupancy of one set by another (the coverage of one function by another) for the range of tasks required in feature extraction and object recognition. The operators used form a unique feature of the project, and are amenable to efficient hardware implementation.

The problems described have (at least partially) been solved; the architecture developed may offer a fruitful path to the completion of the task.

References

References:

1. R Schalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches", J Wiley & Sons, New York, 1992.

2. G Matheron, "Random Sets and Integral Geometry", Wiley, New York, 1975.

3. J Serra, "Image Analysis and Mathematical Morphology", Academic Press, New York, 1982.

4. R M Haralick, S R Sternberg, X Zhuang, "Image Analysis using Mathematical Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-9, No. 4, July 1987.

5. C Giardina, E Dougherty, "Morphological Methods in Image and Signal Processing", Prentice Hall, 1988, pp106 - 109.

6. S R Sternberg, "Parallel Architecture for Image Processing", Proceedings of Third International IEEE Compsac, Chicago, 1979.

7. S R Sternberg, "Greyscale Morphology", Computer Vision, Graphics and Image Processing, Vol. 35, pp333-355, 1986.

8. H Blum, R N Nagel, "Shape Description using Weighted Symmetric Axis Features", Pattern Recognition, Vol. 10, pp.167 - 180, 1978.

9. P Maragos, "Pattern Spectrum and Multiscale Shape Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989.

10. P Maragos, R W Shafer, "Morphological Skeleton Representation and Coding of Binary Images", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol ASSP-34, pp1228 - 1244, Oct 1986.

11. P Maragos, "Morphology-based Multidimensional Signal Processing", Proccedings of 21st Annual Conference on Information Science and Systems, John Hopkins University, Baltimore, MD, March 1987, pp513-518.

12. J Brontskill, A N Venetsanapoulis, "Multidimensional Shape Description and Recognition using Mathematical Morphology", Journal of Intelligent and Robotic Systems, Vol 1, pp117 -143, 1988.

13. T R Esselman, J G Verly, "Feature Extraction from Range Imagery using Mathematical Morphology", SPIE Vol. 845, Visual Communications and Image Processing 11, 1987.

14. P Y Shih, O R Mitchell, "Skeletonization and Distance Transformation by Greyscale Morphology", in "Automated Inspection and High Speed Vision Architectures", SPIE Vol. 849, November 1988, pp80-86.

15. H Heijmans, C Ronse, "The Algebraic Basis of Mathematical Morphology 1: Dilations and Erosions", Computer Vision, Graphics and Image Processing, Vol 50, 1990, pp245 - 295.

16. I Pitas, A N Ventsanapoulis, "Morphological Shape Decomposition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, Jan 1990.

17. D Schonfeld, J Goutsias, "Optimal Morphological Pattern Restoration from Noisy Binary Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 1, Jan 1991.

18. A Toet, "A Morphological Pyramidal Image Decomposition", Pattern Recognition Letters, Vol 9, May 1989, pp255 - 261.

19. D Wang, V Haese-Coat, A Bruno, J Ronsin, "Texture Classification and Segmentation Based on Iterative Morphological Decomposition", Journal of Visual Communications and Image Representation, Vol 4, No 3, September 1993, pp197 - 214.

20. D Sinha, E R Dougherty, "Fuzzy Mathematical Morphology", Journal of Visual Communications and Image Representation, Vol 3, Part 3, September 1992, pp286 - 302.

21. E R Dougherty, D Sinha, P Sinha, "Fuzzy Morphological Filters", Intelligent Robots and Computer Vision XI, D P Casasent(Ed.), SPIE Vol 1825, November 1992.

22. L Koskinen, J Astola, Y Neuvo, "Soft Morphological Filters", Image Algebra and Morphological Image Processing II, SPIE Vol 1568, July, 1991, pp262 - 270.

23. J-G Postaire, R D Zhang, C Lecocq-Botte, "Cluster Analysis by Binary Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 2, Feb 1993.

24. N R Harvey, S Marshall, "The Design of Different Classes of Morphological Filter using Genetic Algorithms", 5th International Conference on Image Processing and its Applications, IEE Conference Publication No. 410, Edinburgh, July, 1995, pp 227 -231.

25. G Matsopoulos, S Marshall, "Use of ANNs for filter design", Ultrasound Images, May, 1994.

26. S J Rees, B F Jones, "Operator for Object Recognition and Scene Analysis by Estimation of Set Occupancy with Noisy and Incomplete Data Sets", Proceedings of the SPIE Vol. 1825 Intelligent Robots and Computer Vision XI, D Casasent (Ed.), Boston, Nov 1992, pp 289 -297.

27. S J Rees, B F Jones, "Direct Feature Extraction using Conditional Morphological Operators", Proceedings of 10th International Conference on Systems Engineering, Coventry Univeristy, Sept 1994, pp998-1005.

28. H S Baird, Model-Based Image Matching using Location, ACM Distinguished Dissertations Series, MIT Press, Mass, USA, 1985, ISBN 0-262-02220-6.

29. M K Hu, "Visual Pattern Recognition by Moment Invariants", IRE Transactions on Information Theory, Vol IT-8, 1962, pp 179-187.

30. C T Zahn, R Z Roskies, "Fourrier Descriptors for Plane Closed Curves", IEEE Transactions on Computers, C-21, 1972, pp 269-281.

31. C H Teh, R T Chin, "On Image Analysis by the Method of Moments", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-10, No. 4, 1988, pp 496-513.

32. Larry S Davis, "Understanding Shape: Angles and Sides", IEEE Transactions on Computers, Vol C-26, No 3, March, 1977, pp 236-242.

33. D P Illing, "Orientation and Recognition of Both Noisy and Partially Occluded 3-D Objects from Single 2-D Images", PhD Thesis, The Polytechnic of Wales, September, 1990, pp 64-66.

34. R Bolles, R Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method", International Journal of Robotics Research, Vol 1, Part 3, 1982, pp 57-81.

35. M B Clowes, "On Seeing Things", Artificial Intelligence 2, 1, Spring 1971, pp 79 - 116.

36. Wallace, T, Wintz, P, "An Efficient Three-dimensional Aircraft Recognition Algorithm using Normalized Fourrier Descriptors", Computer Graphics and Image Processing, Vol 13, 1980, pp96 - 126.

37. S J Rees, B F Jones, "Recognition of 2-D Shapes using Set Erosion", in "Intelligent Robotics and Computer Vision X: Algorithms and Techiques", D P Casasent (Ed.), Proc SPIE 1607, 1992, pp 206-216.

38. D Vernon, "Two-Dimensional Object Recognition using Partial Contours", Image and Vision Computing, Vol 5, No. 1, Feb 1987.

39. W E L Grimson, "On the Recognition of Curved Objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11, No. 6, June 1989, pp 632-643.

40. L Dorst, A W M Smeulders, "Discrete Representation of Straight Lines", IEEE Transactions on PAMI, Vol PAMI-6, No. 4, July 1984.

41. H Voorhees, Tomaso Poggio, "Detecting Textons and Texture Boundaries in Natural Images", Proceedings of the First International Conference on Computer Vision, IEEE Computer Society, London, June 1987, pp250 - 258.

42. B Rosenberg, "The Analysis of Convex Blobs", Computer Graphics and Image Processing, Vol 1, 1972, pp183 - 192.

43. A Rosenfeld, E Johnston, "Angle Detection on Digital Curves", IEEE Transactions on Computers, Vol. C-22, 1973, pp875 - 878.

44. C Chien, J K Aggarwal, "Model Construction and Shape Recognition from Occluding Contours", IEEE Transactions on PAMI, Vol PAMI-11, No 4, pp372 - 389.

45 D Ballard, C Brown, "Computer Vision", Prentice Hall Inc, New Jersey, 1982.

46 H Freeman, "Computer Processing of Line Drawing Images", Computer Surveys 6, 1, March 1974, pp57 - 98.

47 A Rosenfeld, "Digital Straight Line Segments", IEEE Transactions on Computing, Vol. C-23, 1974, pp1264 - 1269.

48 G Wilson, B Batchelor, "Convex Hull of Chain-Coded Blob", IEE Proceedings, Vol 136, Part E, No. 6, Nov 1989, pp530 - 534.

49 L G Roberts, "Machine Perception of Three-dimensional Solids", in Optical and Electro-optical Information Processing, J P Tippet et al. (eds), Cambridge, MA, MIT Press, 1965.

50. D Marr , E Hildreth, "Theory of Edge Detection", Proceedings of the Royal Society of London, Vol B 207, 1980, pp187 - 217.

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

52. R Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", International Journal of Computer Vision, Vol. 1, No 2, May 1987.

53. R Haralick, "Edge and Region Analysis for Digital Image Data", Computer Graphics and Image Processing, Vol 12, 1980, pp 60 - 73.

54. A Kundu, "Robust Edge Detection", Pattern Recognition, Vol. 23, 1990, pp 423 - 440.

55. M Petrou, J Kittler, "A Robust Method for Edge Detection", Proceedings of the SPIE Vol. 1708 Applications of Artificial Intelligence X: Machine Vision and Robotics, 1992, pp 267 - 281.

56. J Serra, "Digital Morphology in the 3-D Space", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Paris, France, 1982, pp843 - 845.

57. S R Sternberg, "Cellular Computer and Biomedical Image Processing", in "Real-Time Medical Processing", Plenum Press, 1980.

58. D Casasent, R Schaefer, "Optical Gray Scale Morphology for Target Detection", in P Gader, E Dougherty, Eds, Proceedings of the SPIE Vol 1586, July 1992, pp313 - 326.

59. "Noise Reduction using Median Filters", IEE Colloquium on Morphological and Non-linear Image Processing Techniques, IEE Digest No 1993/145, London, 10th June, 1993.

60. R M Haralick, “Statistical and Structural Approaches to Texture”, Proceedings of the IEEE, Vol. 67, No. 5, May 1979, pp 786-804.

63. L Kitchen, A Rosenfeld, “Gray Level Corner Detection”, Technical Report 887, Computer Science Center, University of Maryland, College Park, 1980.

64. O Zuniga, R M Haralick, “Corner Detection using the Facet Model”, Proceedings of the IEEE Computer Vision and Pattern Recognition Conference, Washington, USA, 1983, pp30 - 37.

65. L Dreschler, H Nagel, “Volumetric Model and 3-D Trajectory of a Moving Car Derived from Monocular TV-Frame Sequences of a Street Scene”, Proceedings of the International Joint Conference on Artificial Intelligence, Vancouver, British Columbia, 1981, pp 692 - 697.

66. S M Smith, “Feature Based Image Sequence Understanding”, D.Phil. thesis, Robotics Research Group, Dept of Engineering Science, Osford University, 1992.

67. R M Haralick, L G Shapiro. Computer and Robot Vision, Volume 1, Addison Wesley, 1992.

68. S M Smith, J M Brady, “SUSAN - A New Approach to Low Level Image Processing”, Technical Report TR95SMS1c, Defence Research Agency, Farnborough, Hampshire BU14 6TD.

69. R M Haralick, K Shanmugam, I Dinstein, “Textural Features for Image Classification”, IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-3, Nov 1973, pp 610 - 621.

70. Private communication Rees/ V Worner, British Steel Research Laboratories.

71. Liu, Srinath, "Partial Shape Matching using Contour Matching in Distance Transformation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 11, Nov 1990, pp 1072 - 1079.

72. W E L Grimson, D P Huttenlocker, "On the Sensitivity of the Hough Transform for Object Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 3, Mar 1990, pp 255 - 274.

73. J Illingworth, J Kittler, "The Adaptive Hough Transform", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-9, No 5, Sept 1987.

74. M J E Golay, "Hexagonal Parallel Pattern Transformation", IEEE Transactions on Computers, Vol. C-18, 1969, pp 733 - 740.

Appendices

Appendix A.1: Shifted Elements for Example 2.2.2

Model = {(length, orientation relative to grid, sequence position around shape),....}

Trapezoid Model	{(200,90,0),(100,180,1),(100,270,2),(141,315,3)}	= Set *A*
Square Model	{(100,0,0),(100,90,1),(100,180,2),(100,270,3)}	= Set *B*
Overlapped Shape	{(100,0,0),(100,90,1),(25,180,2),(150,90,3),(100,180,4), (100,270,5),(71,315,6),(25,180,7),(100,270,8)}	= Set *C*

The process applied uses only the orientation and the sequence number. Given that the presence of an element, even if it fails to contain its corresponding component, is adequate for the result, the length co-ordinate is removed from the images of *C* shifted by the elements of *A* and *B*.

$(C)_{-a}$:

$a_{-(90,0)}$: {(-90,0),(0,1),(90,2),(0,3),(90,4),(180,5),(225,6),(90,7),(180,8)}
$a_{-(180,1)}$: {(-180,-1)(-90,0),(0,1),(90,2),(0,3),(90,4),(135,5),(0,6),(90,7) }
$a_{-(270,2)}$: {(-270,-2),(-180,-1)(-90,0),(-180,1),(-90,2),(0,3),(45,4),(-90,5),(0,6)}
$a_{-(315,3)}$: {(-315,-3),(-225,-2),(-135,-1),(-225,0),(-135,1),(-45,2),(0,3),(-135,4),(-45,5)}

By inspection, the only element existing in all the shifted images of *C* is (0,3). This would yield the $\underline{R}$ set as a single result {orientation, position, *R*}={(0,3,4)}

$(C)_{-b}$:

$b_{-(0,0)}$ = {(0,0),(90,1),(180,2),(90,3),(180,4),(270,5),(315,6),(180,7),(270,8)}
$b_{-(90,1)}$ = {(-90,-1),(0,0),(90,1),(0,2),(90,3),(180,4),(225,5),(90,6),(180,7)}
$b_{-(180,2)}$ = {(-180,-2),(-90,-1),(0,0),(-90,1),(0,2),(90,3),(135,4),(0,5),(90,6)}
$b_{-(270,3)}$ = {(-270,-3),(-180,-2),(-90,-1),(-180,0),(-90,1),(0,2),(45,3),(-90,4),(0,5)}

Four of the elements exist in three of the shifted images of the data set. Thus:

$\underline{R}$= {(0,0,3),(90,3,3),(-90,-1,3),(0,2,3)}

$$
\begin{aligned}
\boldsymbol{J} &= \{J_{(0,0)}, J_{(90,3)}, J_{(-90,-1)}, J_{(0,2)}\} \\
&= \{\{(75,180,2),(100,270,3)\},\{(100,360,6)\},\{(100,-90,-1),(75,180,2)\}, \\
&\quad \{(100,0,2)\}\}
\end{aligned}
$$

The dissimilar numbers of elements in the *J* sets arise from partial coverage of an element in the shifted version.

Appendix A.2 - Erosion Calculations

Example 1: Formal Erosion by g(z) = {(-2,-2),(-1,-1),(0,0),(1,1),(2,2)}

Position	Evaluation Basis = $\min\{f(x+z) - g(z)\}, \forall z \in g(z)$	Output Result at Position
	$=\min\{(f(x-2)-(-2)), (f(x-1)-(-1)), (f(x)), (f(x+1)-1), (f(x+2)-2)\}$	
1	{7,6,5,4,3}	3
2	{7,6,5,4,2}	2
3	{7,6,5,3,1}	1
4	{7,6,4,2,0}	0
5	{7,5,3,1,-1}	-1
6	{6,4,2,0,-2}	-2
7	{5,3,1,-1,-1}	-1
8	{4,2,0,0,0}	0
9	{3,1,1,1,1}	1
10	{2,2,2,2,2}	2
11	{3,3,3,3,3}	3
12	{4,4,4,4,3}	3
13	{5,5,5,4,3}	3
14		3
15		3
16		3

Example 2: R Conditioned Erosion by {(-2,-2),(-1,-1),(0,0),(1,1),(2,2)}, Condition R∈{4,5}

Position	Evaluation Basis = $(f \copyright g)(x){:}T(x) =$	R value	Output Result at Position
	$\{k \mid k = \min f(x+z) - g(z) \forall z \in g(z)$ where $t = 0$, $k = f(x)$ where $t = 1$, $\forall x \in f(x)\}$		
1	{7,6,5,4,3}	3	3
2	{7,6,5,4,2}	3	2
3	{7,6,5,3,1}	3	1
4	{7,6,4,2,0}	3	0
5	{7,5,3,1,-1}	3	-1
6	{6,4,2,0,-2}	3	-2
7	{5,3,1,-1,-1}	3	-1
8	{4,2,0,0,0}	3	0
9	*f(9)*	4	1
10	*f(10)*	5	2

11	*f(11)*	5	3
12	*f(12)*	4	4
13	{5,5,5,4,3}	3	3
14			3
15			3
16			3

Example 3: Soft Erosion, Core = *i(y)*={(-1,-1),(0,0),(1,1)}, Soft Umbra = {(-2,-2), (2,2)}, Rank = 2.

Position	Evaluation Basis = 2nd Smallest Value of Multiset $f\Theta[h,g,k](x) = k^{th}$ smallest value of the multiset $\{k\Diamond(f(y)-g(y-x)): y \in (g(y))_x\} \cup \{f(z)-h(z-x): z \in (h(z))_x, z \notin (g(y))_x\}$ =2nd smallest of {*(f(x-1)-(-1)), (f(x-1)-(-1)), (f(x)),(f(x)), (f(x+1)-(1)), (f(x+1)-(1)),(f(x-2)-(-2)),(f(x+2)-2)}*	Output Result at Position
1	{6,6,5,5,4,4, 7,3}	4
2	{6,6,5,5,4,4, 7,2}	4
3	{6,6,5,5,3,3, 7,1}	3
4	{6,6,4,4,2,2, 7,0}	2
5	{5,5,3,3,1,1, 7,-1}	1
6	{4,4,2,2,0,0, 6,-2,}	0
7	{3,3,1,1,-1,-1, 5,-1}	-1
8	{2,2,0,0,0,0, 4,0}	0
9	{1,1,1,1,1,1, 3,1}	1
10	{2,2,2,2,2,2, 2,2}	2
11	{3,3,3,3,3,3, 3,3}	3
12	{4,4,4,4,4,4, 4,3}	4
13	{5,5,5,5,4,4, 5,3}	4
14	{6,6,5,5,4,4, 6,3}	4
15	{6,6,5,5,4,4, 7,3}	4
16		4

Appendix A3: Probe Sets and Sets of Sets

For each of the sizes of set the digits following the name indicate the intensity height of the elements. For example, the template sets donut10.dat and donut20.dat would have elements of intensity value 10 and 20 respectively.

Donut.dat

y	x	height
-3	-1	5
-3	0	5
-3	1	5
-2	-2	5
-2	2	5
-1	-3	5
-1	3	5
0	-3	5
0	3	5
1	-3	5
1	3	5
2	-2	5
2	2	5
3	-1	5
3	0	5
3	1	5

Smalldo.dat

y	x	height
-2	-1	5
-2	0	5
-2	1	5
-1	-2	5
-1	2	5
0	-2	5
0	2	5
1	-2	5
1	2	5
2	-1	5
2	0	5
2	1	5

Bigdonut.dat

y	x	height
-6	-2	5
-6	0	5
-6	2	5
-4	-4	5
-4	4	5
-2	-6	5
-2	6	5
0	-6	5
0	6	5
2	-6	5
2	6	5
4	-4	5
4	4	5
6	-2	5
6	0	5
6	2	5

Hugenut.dat

y	x	height
-9	-3	5
-9	0	5
-9	3	5
-6	-6	5
-6	6	5
-3	-9	5
-3	9	5
0	-9	5
0	9	5
3	-9	5
3	9	5
6	-6	5
6	6	5
9	-3	5
9	0	5
9	3	5

Vastnut.dat

y	x	height
-18	-6	5
-18	0	5
-18	6	5
-12	-12	5
-12	12	5
-6	-18	5
-6	18	5
0	-18	5
0	18	5
6	-18	5
6	18	5
12	-12	5
12	12	5
18	-6	5
18	0	5
18	6	5

Gnut.dat

y	x	height
-36	-12	5
-36	0	5
-36	12	5
-24	-24	5
-24	24	5
-12	-36	5
-12	36	5
0	-36	5
0	36	5
12	-36	5
12	36	5
24	-24	5
24	24	5
36	-12	5
36	0	5
36	12	5

dset.par

smalldo.dat
donut.dat
bigdonut.dat
hugenut.dat
vastnut.dat

d20set .par

smdo20.dat
donut20.dat
bigdo20.dat
hugnut20.dat
vnut20.dat

d50.par

smdo50.dat
donut50.dat
bigdo50.dat

d50hug.par

donut50.dat
bigdo50.dat
hugnut50.dat

Appendix A.4: Corner Feature Extraction

A4.1: Actual and Extracted Corner Positions, Salt and Pepper Noise

Image3 corners	**im3d1212 corners (Clean Image)**	**im3n10d1212 corners (10% Salt and Pepper Noise)**	**im3n20d1212 corners (20% Salt and Pepper Noise)**	**im3n30d1212 corners (30% Salt and Pepper Noise)**
329,338	329 337 331 337	326 338	321 339	Excessive inclusion of incorrect hits - effectively an irrelevant discussion unless specifically hunting known loci for shapes.
346,348	346 345 347 348	347 348 346 349 347 349 347 350		
336,365	335 366	336 366 335 367		
319,355	318 353	318 355	319 355	
379,336	380 335	380 335	378 331	
396,344	397 344	397 341 397 344	395 341 400 344	
390,362	388 363	393 361 392 362 393 362		
372,355	370 353 372 357 375 337	370 353 371 353 370 356 372 357		
426,339	427 337 423 339	424 339	419 345 418 347	
444,342		446 339	449 344	
441,361	443 360	443 361 435 366 439 364		
421,358	420 355 423 361	420 355 420 356	421 361	
322,397	323 395 321 397 321 401	321 398	322 396	
339,389	335 388 338 388	339 386		

349,405	349 406	350 404 351 404 350 405 349 408	350 407
333,415		335 418	334 417
370,399	369 397 368 401 369 403	368 402	
384,386	382 386 385 386	387 386 386 384	
397,402	397 399 397 403		399 403 401 405
383,414	380 415 384 415		
420,402	419 404	419 404	
432,388	430 387 434 387		436 385
447,400	447 398 448 402		
434,415	432 416	432 416 437 436	
324,442	323 441	323 441	
344,442	344 441		
344,462	344 462		
324,462	323 462		319 462
372,442	373 442 371 444 371 446	373 441 374 440	
392,439	393 440 389 438 391 438	392 437	394 439
395,459	397 456 395 460	397 456 395 460	
376,462	375 462 379 464	377 464	
422,445	422 443	420 446	

	420 447		
440,438	438 436	439 438	439 437 443 439 441 435 443 436
446,456	448 454 447 456	448 454 447 458	449 455 451 453
429,463	425 462 430 464	430 465	
			False hits 154 91 418 166 416 181 162 240 43 300 422 303 203 380 142 381 52 383 319 408 50 452 187 417

A4.2: Testing of image3.tga with Gaussian Noise Added,Standard Deviation 10 Grey Levels, with d51.par, d50.par {12;12}

Image3 corners	im3g10 corners extracted by R intersection with d50{12:12}	im3g10 corners extracted by R intersection with d51{12:12}	im3g10 corners extracted by R intersection with d50{9:9,12:12}	im3g10 corners extracted by MHMMT with donut20.dat and limits{6:1,6:1}
329,338	332 337			
346,348		347 348		349 347
336,365	333 366	337 365 334 366		
319,355		318 353 319 355	319 355	
379,336	375 339	379 336 381 336	379 336	378 337 379 338
396,344	398 348	396 341 397 343 398 346	397 343	395 344 396 344 397 344
390,362		392 361	391 362	392 364

		389 363	389 363	
372,355		370 353	371 354	371 358
		372 356	372 357	
426,339		428 337	425 338	426 340
		425 339	423 341	
		423 340		
444,342		443 340	443 340	444 342
			445 342	
441,361	444 360	443 360	440 362	
		438 363	438 363	
421,358	424 361	420 356	421 357	422 357
	420 355		422 359	422 358
322,397		322 399	322 399	324 398
339,389		338 388	338 388	338 390
349,405	350 402	350 403		
		348 407		
333,415	329 415	329 415		334 414
		332 416		335 417
370,399		369 399	369 399	370 400
384,386		385 386		384 388
397,402	397 398	397 400	397 403	396 403
	397 405	397 404		398 404
383,414	385 415	381 414		382 413
		384 415		383 413
420,402		419 400	419 402	
		419 404		
432,388	435 387	431 387		431 389
		434 387		
447,400	448 403	447 398	447 398	446 399
		447 402	448 400	
434,415		433 416		435 415
324,442		323 441	323 441	325 442
344,442		344 441	344 441	
344,462		344 462	344 462	344 462

324,462		323 462	323 462	324 462 325 462
372,442		373 442 371 445	373 442 371 445	370 443 373 444
392,439		389 438 393 440	390 438 393 440	391 440 392 440
395,459		395 460 397 456	395 460	397 451
376,462		375 462 378 464	375 462 378 464	376 461
422,445		422 443 420 446	422 443 421 445	422 445
440,438		440 437	439 437 441 438	440 439
446,456	448 452 445 460	447 455 446 459	447 457	448 453 446 456 446 457
429,463	425 461	428 464	429 464	428 462 429 463 427 465
				False Hits
				49 268 344 358 398 350 378 361 383 362 335 439 445 447 333 465

Appendix 5: Steve.tga, d0.par{12,12} for the Detection of Right Corners

Synthetic Image Features	Clustered features, 10% Salt and Pepper Noise, extracted using d0.par{12,12}	Clustered Features, No Noise d.par{12,12}	Location Details
Intensity Steps			
122 32	miss	122 33	three intensity levels meet at intersection of cadence in intensity, including two different foreground and the background grey levels
122 64	miss	122 65	
122 96	121 102	122 97	
122 128	122 131	122 129	
122 160	122 162	122 161	
122 192	miss	122 193	
122 224	miss	122 225	
122 256	122 257	122 257	
122 288	miss	122 289	
122 320	122 322	122 321	
122 352	123 353	122 353	
122 384	miss	122 385	
122 416	121 419	122 417	
122 448	123 449	122 449	
	122 451		
Intensity gradient rectangle within rectangle			
321,32	320 34	321 32	
	322 32		
321,112	319 110	321 112	
337,52	miss		low contrast corner inside object
337,92	329 88		
433,52	miss		low contrast corner inside object
433,92	414 83		
449,32	450 32	449 32	
449,112	449 112	449 112	
Square formed of four squares of different intensity squares			
173 32	173 33	173 32	
	174 30		
253 32	254 34	253 32	
173 112	173 112	173 112	
253 112	256 111	253 112	
213 32	212 33		

253 72	miss	252 73	½ way down side of "four squares"
213 112	215 109	214 111	
173 72	175 72	174 71	
213 72	miss		Centre of "four squares"
Large Triangle			
161 152	geometry		Triangle with two sharp acute and one obtuse
465 152	geometry		corner - should miss these three
313 192	geometry		
Right Angled Triangle			
193 240	194 239	194 239	
234 198	231 199		
234 282	231 285	235 282	
150 282	153 277		
Square with three intensities forming a "Y" shaped boundary inside it			
273 221	275 218	273 220	
309 221	310 221	310 221	
345 221	miss	345 220	Trailing upper corner in "Y"square
309 261	geometry		Centre of "Y"
273 281	276 285	273 282	
345 281	miss	345 282	Trailing lower corner in "Y" square
		341 282	
Rectangle with internal two intensities forming a diagonal boundary			
373 223	374 221		
	376 221		
	371 226		
451 223	geometry		Diagonal in-object intensity change
471 223	miss	471 222	Trailing upper corner of rectangle
373 282	372 281	373 282	
394 282	geometry		Diagonal in-object intensity change
471 282	470 283	471 282	
	473 282		

**"Flying V"
Object**

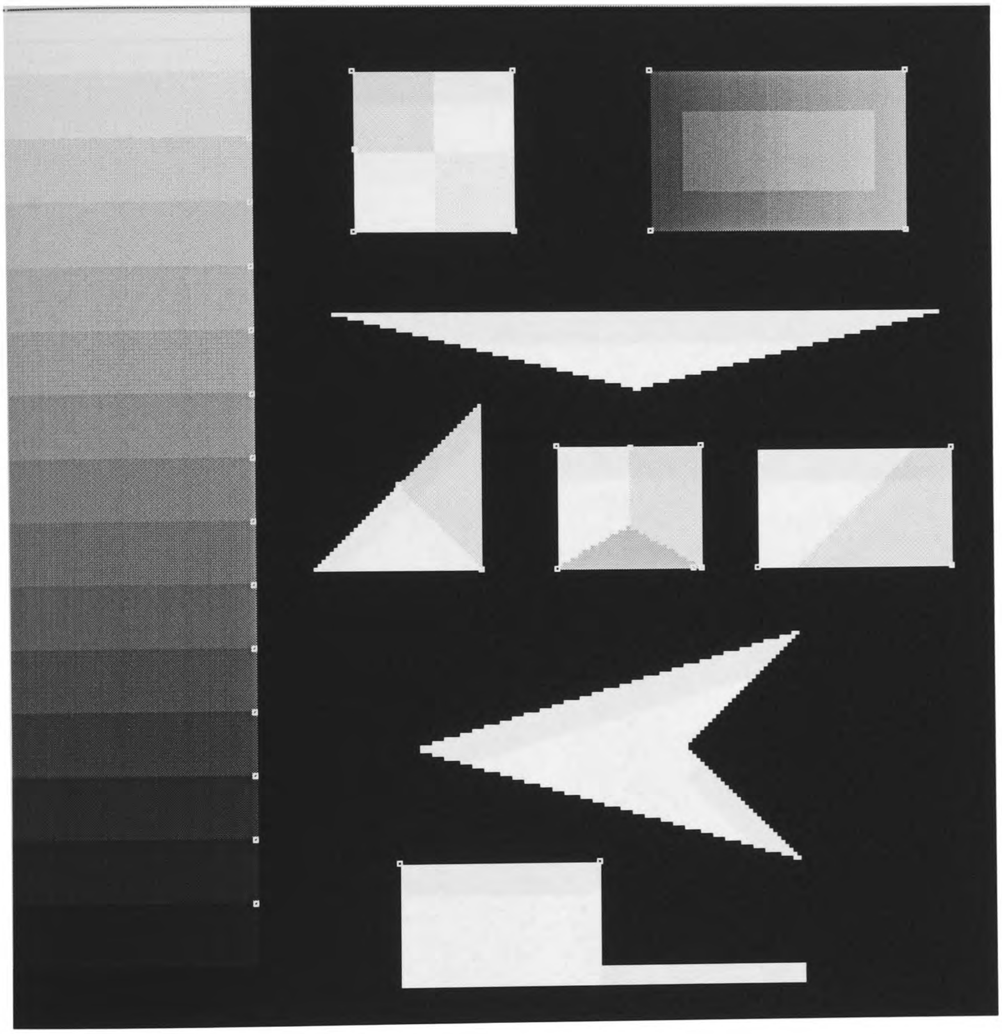
393 315	geometry		tail point of "flying V"
373 372	geometry		concave centre of "flying V"
393 430	geometry		tail point of "flying V"
204 372	205 369		

L shaped area

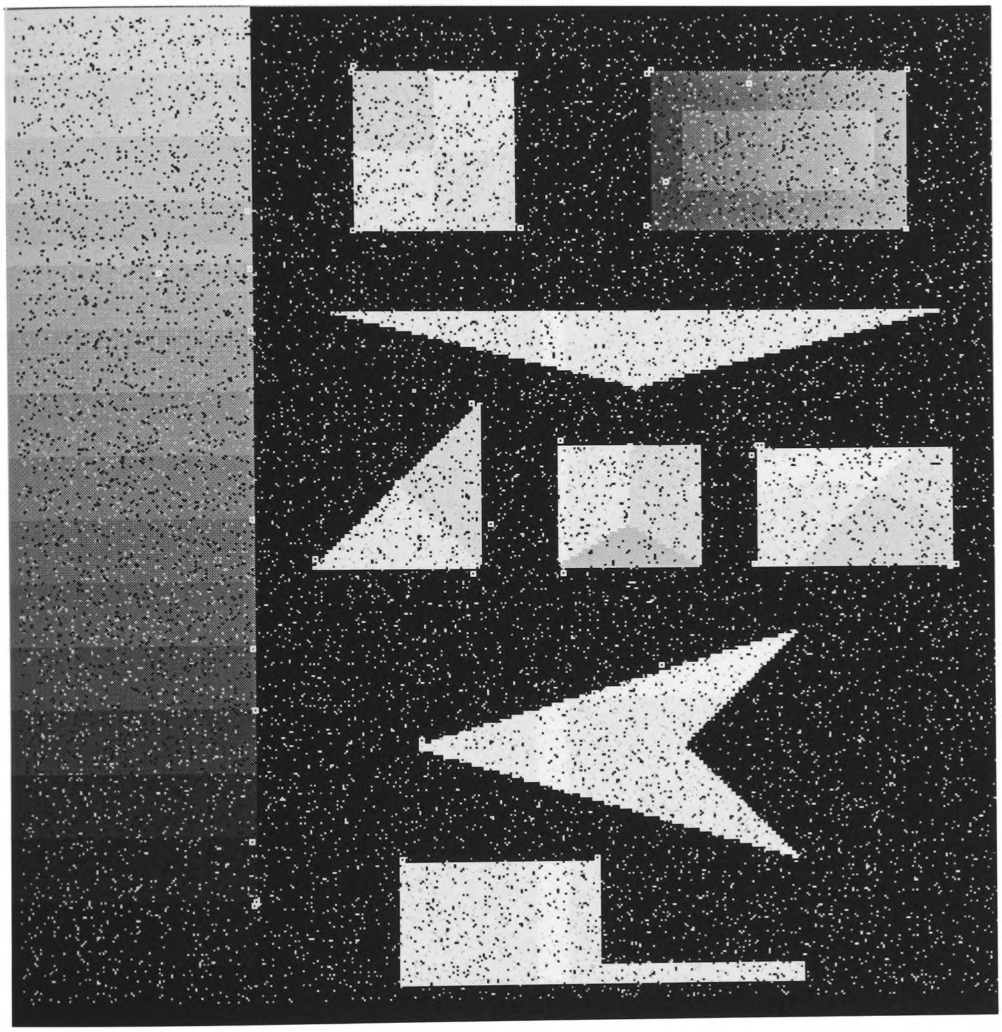
193 430	195 429	193 430	
293 430	292 429	293 430	
293 482	-	-	These point fall outside the limits of feature extraction algorithm test area.
394 482	-	-	
193 492	-	-	
395 492	-	-	

	77 133		False hits
	325 332		
	240 260		
	371 39		

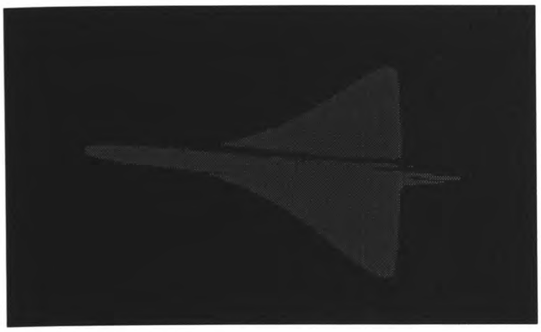
Steve1.tga, No noise, with d.par{12,12} for the Detection of Right Corners



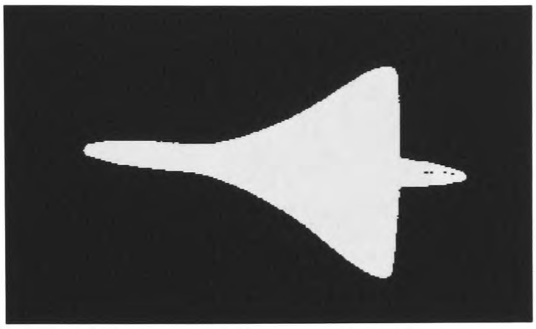
Steve1.tga, 10% Random Noise Added, d0.par{12,12} for the Detection of Right Corners



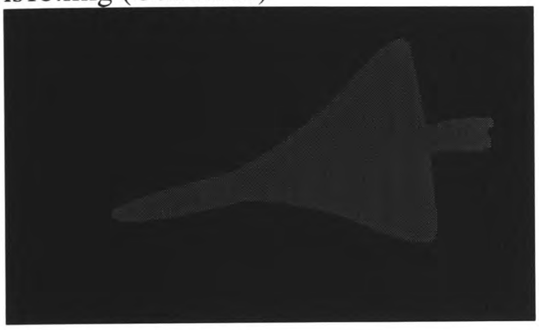
Appendix B.1: Test Objects



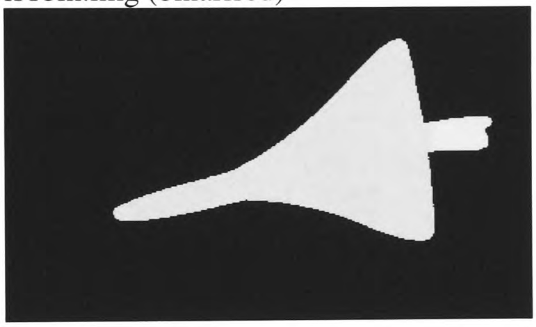
is1b.img (Concorde)



is1bin.img (binarised)

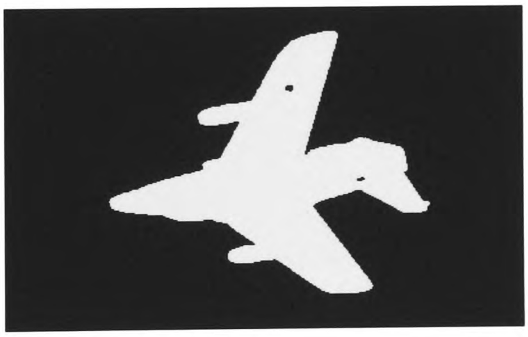


is4b.img (Concorde)

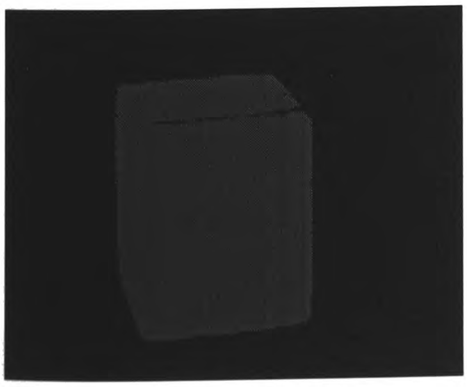


is4bin.img (binarised)

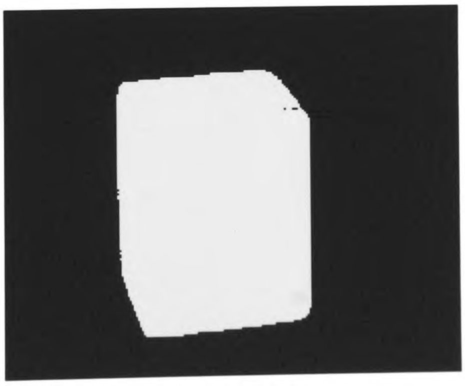
is6b.img (Hawk trainer)



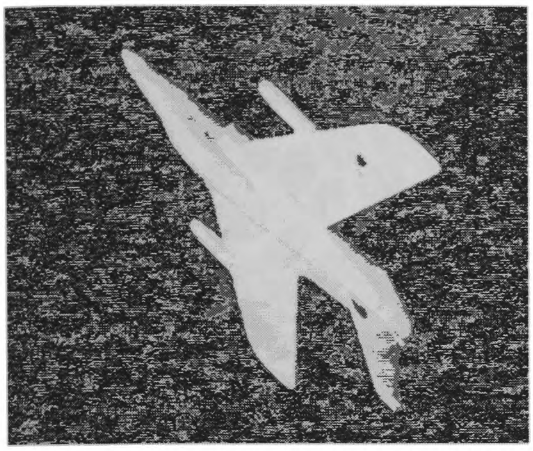
is6bin.img (binarised)



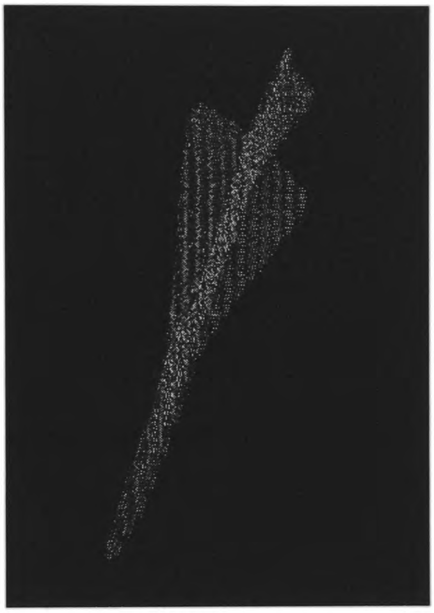
is12b.img (matt white block)



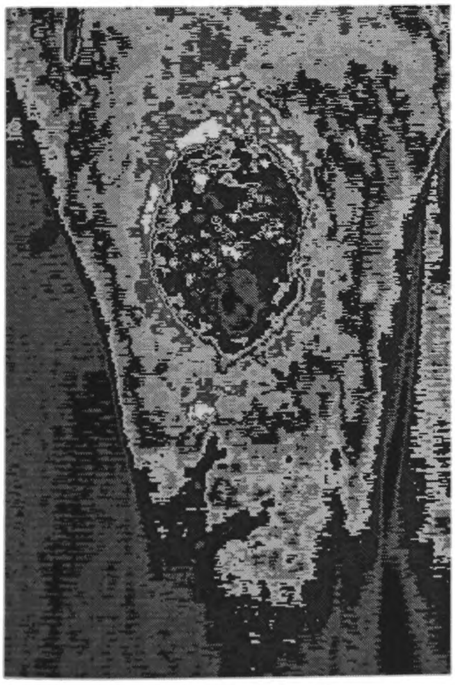
is12bin.img (binarised)



booze.img (low contrast Hawk trainer), histogram equalised for clarity

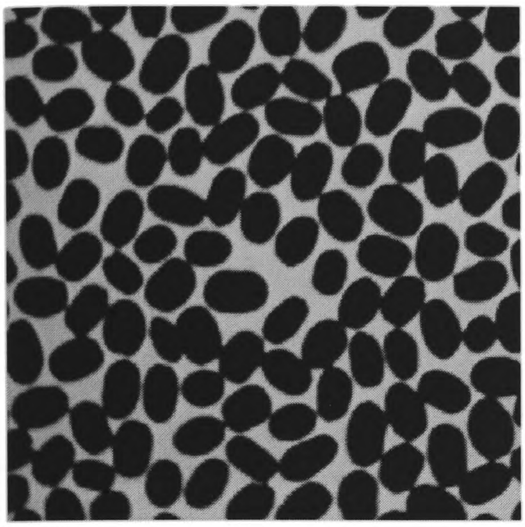


crp.img (noisy Concorde)

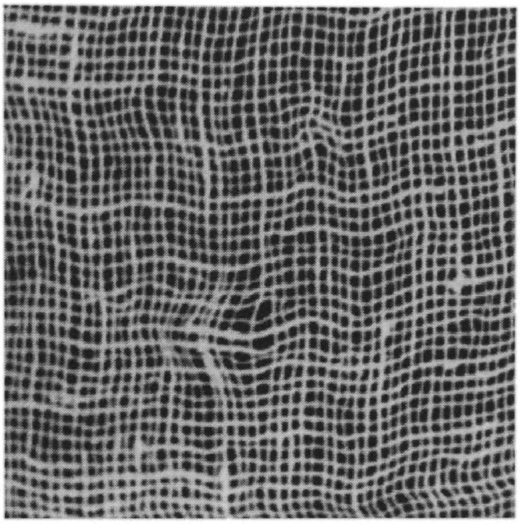


u2b.img (soft tissue ulcer)

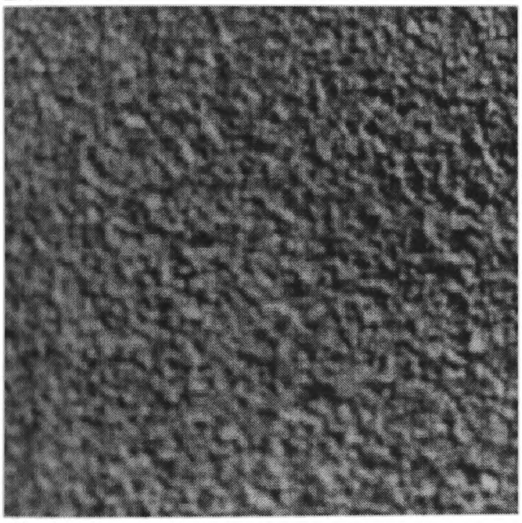
Texture Test Samples



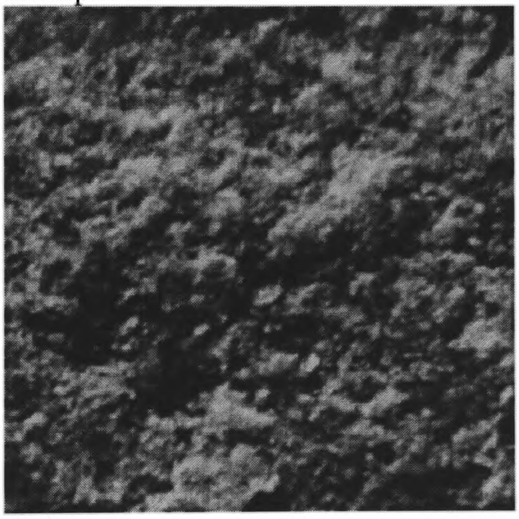
Beans



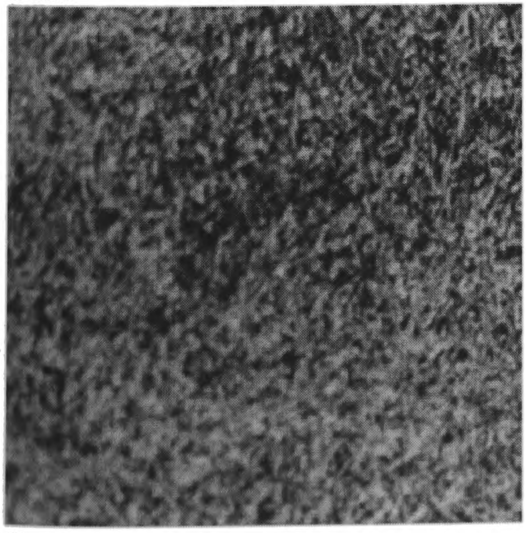
Burlap



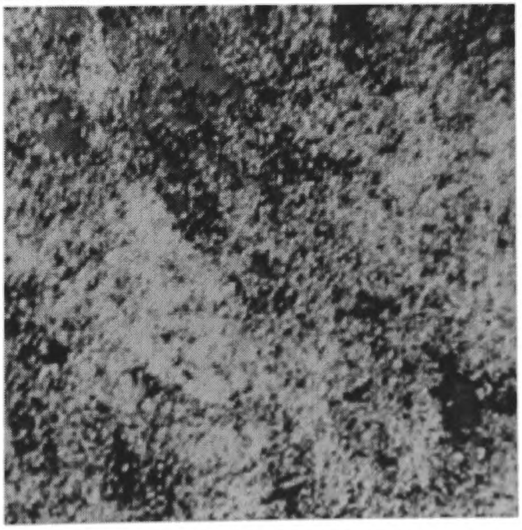
Cork



Fieldsto

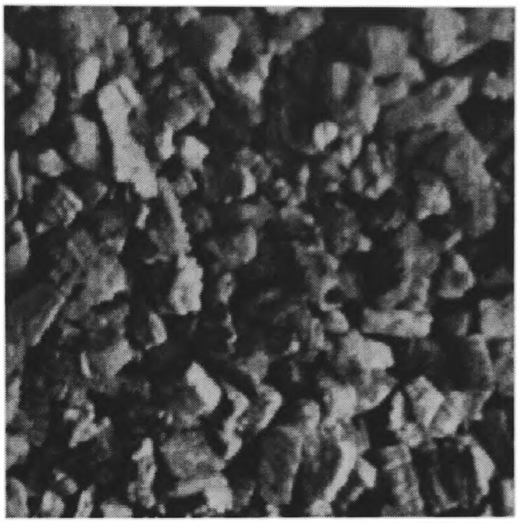


Grass

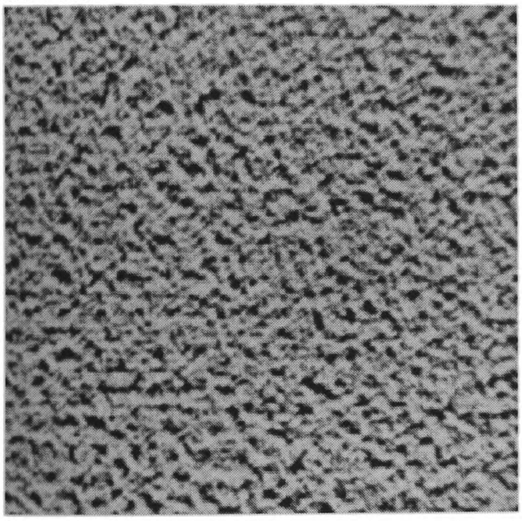


Ice

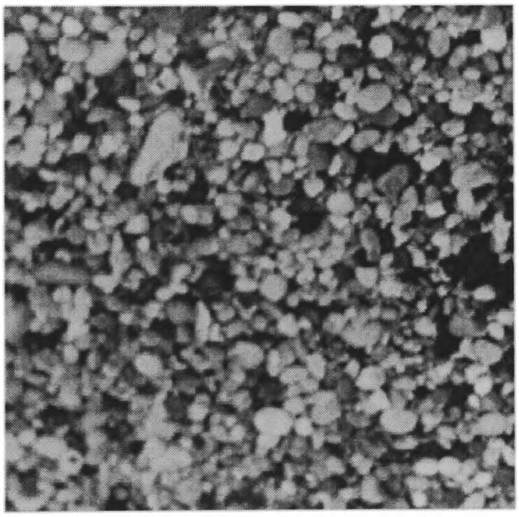
Marble



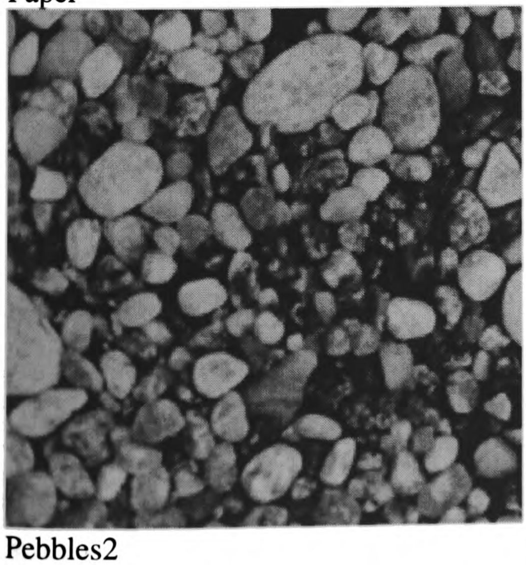
Mica



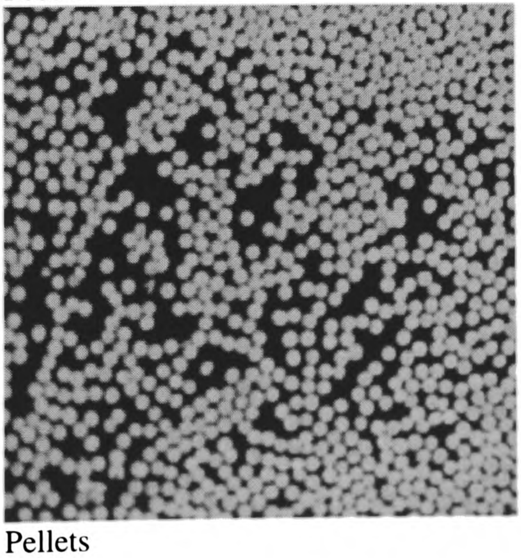
Paper



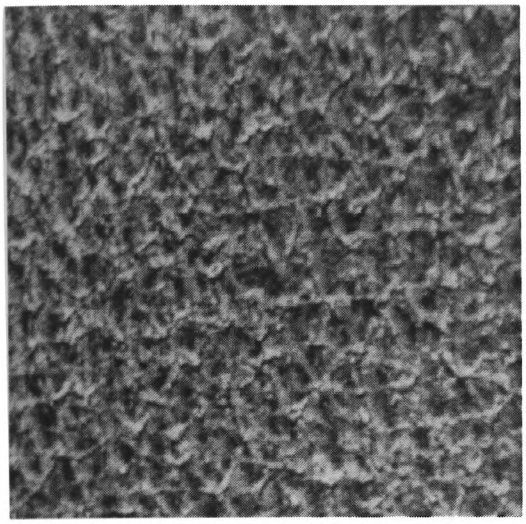
Peb54



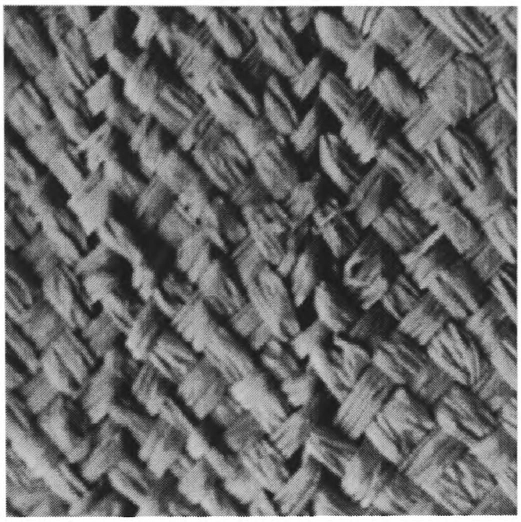
Pebbles2



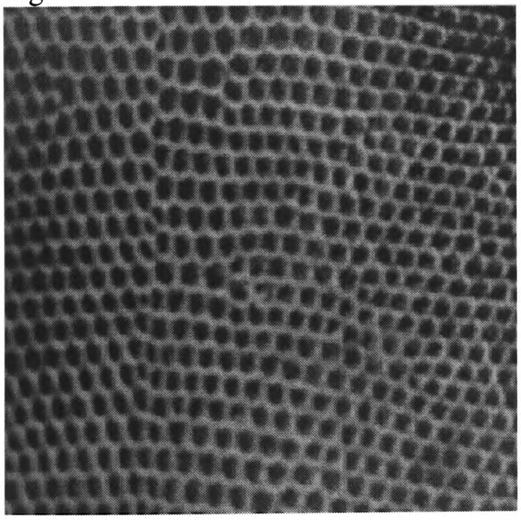
Pellets



Pig

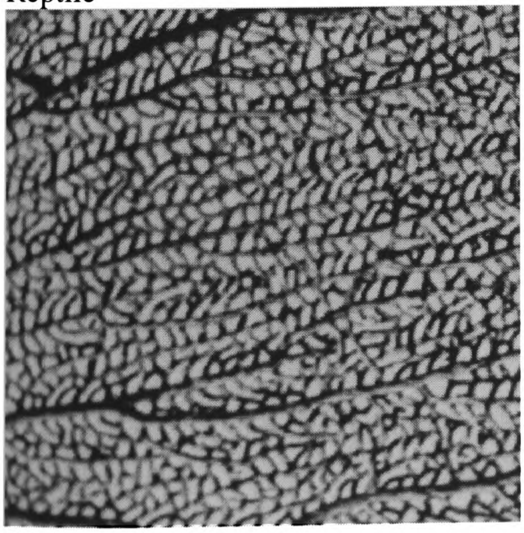


Raffia

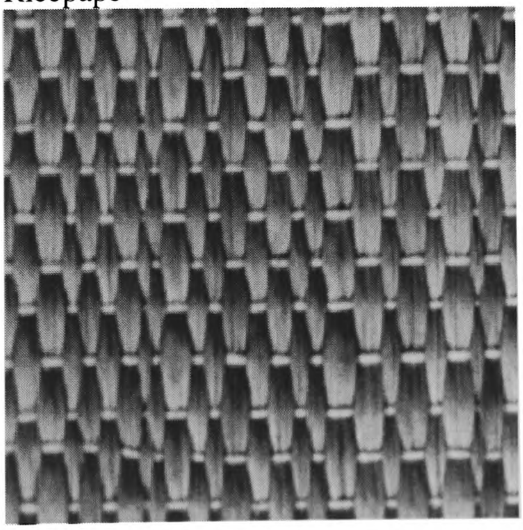


Reptile

Ricepape



Seafan



Straw

Tree

Appendix B.2 : Feature Sets Extracted from Test Images

B.2.1: Unstructured Feature Sets (May Include In-Object Features)

Is1b.img
(eye extracted)

x	y	type
353	288	1
147	345	1
352	431	1
404	364	1
358	292	1
234	340	0
228	362	0
361	371	0
361	349	0

is1b.img
(gab, diffre extracted)

x	y	type
339	293	1
155	349	1
355	419	1
393	357	1
352	290	1
235	363	0
367	369	0
359	343	0
227	341	0

mhmmt

x	y	type
241	340	1
150	343	1
149	344	1
150	344	1
149	345	1
150	346	1
372	361	1
373	361	1
374	361	1
235	362	1
382	362	1
383	362	1
238	363	1
239	363	1
392	363	1
393	363	1
397	363	1
384	364	1
385	364	1
392	364	1
393	364	1
397	364	1
361	369	0
360	370	0
359	371	0

rin

x	y	type
349	288	1
350	288	1
351	288	1
352	288	1
353	288	1
354	288	1
354	289	1
355	289	1
356	289	1
357	290	1
356	291	1
358	291	1
356	293	1
359	293	1
359	294	1
151	339	1
152	339	1
147	346	1
148	346	1
378	353	1
382	354	1
386	354	1
387	354	1
404	362	1
301	364	1
370	368	1
373	368	1
375	368	1
357	380	1
352	427	1
355	428	1
353	429	1
354	429	1
348	430	1
353	430	1
354	430	1
351	431	1
352	431	1
346	432	1
347	432	1
348	432	1
349	432	1
350	432	1
360	371	0

r(gab,diffre
extracted)

x	y	type
341	289	1
261	331	0
221	338	0
165	337	1
154	352	1
226	363	0
357	422	1
368	372	0
404	361	1

is4b.img (eye extracted)	is4b (gab.diffre extracted)	mhmmt	rin	r (gab, diffre)
365 271 1	350 275 1	360 271 1	361 269 1	351 272 1
166 388 1	172 392 1	420 325 1	362 269 1	309 312 0
382 406 1	382 397 1	421 327 1	421 323 1	169 393 1
420 342 1	417 336 1	420 329 1	422 324 1	289 385 0
423 326 1	416 324 1	418 343 1	423 325 1	353 405 1
381 348 0	385 346 0	168 386 1	423 326 1	385 396 1
418 334 0	374 320 0	168 387 1	423 327 1	392 349 0
377 326 0		168 388 1	423 328 1	420 333 1
248 357 0		376 327 0	422 330 1	416 321 1
260 380 0		416 332 0	421 341 1	384 324 1
		379 348 0	421 342 1	376 314 0
			420 344 1	361 269 1
			419 345 1	
			166 385 1	
			166 386 1	
			166 388 1	
			166 389 1	
			167 391 1	
			169 393 1	
			383 405 1	
			376 326 0	
			377 326 0	
			376 327 0	
			377 327 0	
			378 327 0	
			417 332 0	
			417 333 0	
			379 348 0	
			380 348 0	
			379 349 0	
			380 349 0	
			309 388 0	
			320 391 0	

is6b.img (eye extracted)			is6b.img (gab, diffre extracted)			mhmmt extracted			rin extracted			r (gab, diffre)		
x	y	t	x	y	t	x	y	t	x	y	t			
353	116	1	334	117	1	353	118	1	318	157	0	335	114	1
315	123	1	311	129	1	247	186	1	312	159	0	311	125	1
243	182	1	255	187	1	249	217	1	275	185	0	273	168	0
248	216	1	244	223	1	176	246	1	276	185	0	252	190	1
174	248	1	184	253	1	422	248	1	275	186	0	265	201	0
237	264	1	240	261	1	423	249	1	276	186	0	268	202	1
267	290	1	273	294	1	423	250	1	277	186	0			
348	320	1	376	308	1	268	288	1	329	208	0			
381	314	1	421	248	1	379	312	1	328	209	0			
408	259	1	391	203	1	379	313	1	329	209	0			
421	256	1	359	203	1	379	314	1	327	210	0			
425	248	1	343	205	1	325	118	0	328	210	0			
407	221	1	329	211	1	313	157	0	329	210	0			
403	205	1	344	115	1	312	160	0	367	230	0			
376	197	1	278	170	0	287	168	0	368	230	0			
368	197	1	272	195	0	285	169	0	373	230	0			
336	251	0	256	214	0	276	185	0	367	231	0			
369	239	0	224	259	0	277	186	0	334	251	0			
419	246	0	264	263	0	276	187	0	335	251	0			
407	226	0	282	281	0	274	189	0	276	272	0			
361	201	0	313	297	0	360	202	0	285	279	0			
331	203	0	340	246	0	327	209	0	286	279	0			
285	168	0	380	243	0	247	222	0	287	279	0			
276	186	0	406	219	0	404	225	0	288	279	0			
261	212	0	331	196	0	404	226	0	286	280	0			
247	221	0				405	227	0	287	280	0			
217	259	0				369	238	0	288	280	0			
256	262	0				371	239	0	303	293	0			
266	265	0				335	250	0	304	293	0			
286	279	0				334	252	0	305	293	0			
306	294	0				335	253	0						
						218	257	0						
						265	263	0						
						266	263	0						
						286	278	0						
						288	279	0						
						305	292	0						

Is12b.img (eye extracted)			is12b.img (gab, diffre extracted)			mhmmt extracted			rin extracted			r (gab, diffre)		
x	y	type	x	y	type	x	y	type	x	y	type			
295	315	1	296	326	1	299	317	1				293	321	1
296	408	1	300	414	1	298	318	1				301	425	1
310	442	1	316	440	1	299	318	1				315	443	1
392	431	1	392	425	1	297	319	1				395	422	1
393	332	1	361	310	1							390	326	1
372	308	1												

booze.img (eye extracted)

x	y	type
138	128	1
174	201	1
190	260	1
234	358	1
268	390	1
329	390	1
346	408	1
349	404	1
351	359	1
359	339	1
338	299	1
381	221	1
344	187	1
242	154	1
222	184	1
204	238	0
213	273	0
219	299	0
273	304	0
310	331	0
312	322	0
315	209	0
323	218	0
348	398	0
349	358	0
295	266	0
289	189	0
269	192	0
242	197	0
218	186	0

booze (actual via gab,diffre)

x	y	type
140	137	1
193	270	1
268	384	1
344	397	1
333	381	1
337	368	1
349	327	1
342	311	1
333	297	1
312	282	1
372	209	1
245	164	1
142	129	1
208	248	0
212	264	0
206	268	0
223	310	0
231	342	0
278	303	0
313	336	0
350	356	0
356	343	0
304	259	0
279	183	0
258	196	0
227	193	0
206	183	0
191	169	0
171	152	0
157	141	0

mhmmt extracted

x	y	type
360	97	1
140	128	1
141	128	1
140	129	1
141	129	1
142	129	1
245	153	1
246	153	1
246	154	1
242	159	1
221	186	1
222	186	1
223	186	1
221	187	1
222	187	1
284	188	1
210	201	1
380	220	1
379	222	1
193	259	1
194	259	1
192	260	1
193	260	1
194	260	1
191	261	1
190	263	1
191	263	1
329	296	1
49	311	1
352	349	1
348	353	1
351	357	1
266	387	1
267	388	1
265	389	1
266	389	1
267	389	1
265	390	1
266	390	1
267	390	1
349	402	1
348	403	1
349	403	1
343	406	1

x	y	type
296	188	0
285	190	0
286	190	0
269	191	0
282	191	0
284	191	0
285	191	0
269	192	0
270	192	0
268	193	0
269	193	0
316	209	0
317	209	0
318	209	0
316	212	0
323	215	0
324	215	0
324	216	0
324	217	0
320	219	0
321	219	0
293	265	0
293	266	0
293	267	0
211	272	0
213	272	0
212	273	0
213	273	0
212	274	0
213	274	0
214	274	0
340	301	0
271	302	0
272	302	0
273	302	0
271	303	0
311	318	0
272	334	0
322	335	0
352	351	0

rin extracted

268	191	0
269	191	0
283	191	0
284	191	0
268	192	0
269	192	0
317	210	0
318	210	0
323	216	0
324	216	0
293	265	0
294	265	0
295	265	0
212	272	0
272	302	0
272	303	0
273	303	0

is6b concave features extracted using R operator with limits 6,1 (all features type 0)

donut				donut20		donut30	
x	y	x	y	x	y	x	y
317	157	274	185	313	158	275	186
314	158	273	186	318	158	276	186
315	158	274	186	313	159	285	279
316	158	275	186	314	159	286	279
317	158	272	187	318	159		
318	158	273	187	275	185		
314	159	274	187	274	186		
315	159	275	187	275	186		
316	159	276	187	276	186		
317	159	272	188	275	187		
318	159	273	188	328	209		
314	160	274	188	329	209		
315	160	275	188	371	230		
316	160	329	207	335	251		
317	160	328	208	335	252		
318	160	329	208	285	279		
315	161	371	229	286	279		
316	161	372	229	287	279		
		373	229	286	280		
		371	230	287	280		
		372	230				
		373	230				

Booze concave features extracted with d10 and r operator, limits 6,1 (all features type 0)

Concave features extracted with donut10

x	y	x	y	x	y
267	191	323	216	272	302
268	191	324	216	272	303
269	191	293	265	273	303
283	191	294	265	274	303
284	191	295	265	345	346
268	192	294	267		
269	192	211	270		
318	209	212	271		
317	210	212	272		
318	210	213	272		
318	211	213	273		

Appendix B.3: IS6 Concave features, Gaussian Noise at S/N Ratio 5 Added

eye extracted	concave features, rjhma5, donut10{6:2,14:10}	concave features, rin, d10hug{5:3}
336 251	334 252	335 251
369 239	369 238	373 229
419 246		
407 226		
361 201		
331 203		330 207
		328 208
285 168		284 168
276 186	276 185	275 187
	276 187	
261 212		260 212
247 221	247 222	
217 259		
256 262	255 260	
266 265		
286 279	288 279	278 274
		286 278
		288 279
		286 280
306 294 0		291 295
		315 300
		false hits
		326 308
		303 137

IS6 S/N 5, CONVEX EXTRACTED FEATURES

Eye Extracted	convex features, rjhma5, donut10{6:1}	convex features, rin, donut20{13:12}
353 116	353 118	353 116
315 123		
243 182		252 174
248 216		
174 248		177 243
		175 245
		175 248
237 264		
267 290	268 288	267 287

		267 290
348 320		
381 314	379 313	380 310
		380 313
408 259		
421 256		
425 248	423 249	
407 221		
403 205		
376 197		377 197
368 197		

Appendix B.4: IS6B in Pixellated Noise

Convex eye extracted
vol=20698
per=906
	is6nd50 d101312
	vol=20957
	per=980
		is6nd100 d101312
		vol=21234
		per=1050
			is6nd503d1 01312	is6 major controls set (convex)

353 116	349 112 354 115	354 115	346 109 354 114 355 117	353 116
315 123				
243 182	242 179 243 184	241 181	242 180 243 184	243 182
248 216			403 203	
174 248	176 241 174 244 177 252	176 241 176 243 173 245 177 253	176 241 173 243	174 248
237 264				
267 290	267 283 265 287 266 292 268 295	268 280 266 284 269 283 265 289 267 293	267 283 265 287 266 292 268 295	267 290
348 320				
381 314	381 311 381 313 379 314 380 316	384 311 378 313 381 313 383 313 380 316	381 312 380 316	381 314
408 259				
421 256	424 254 421 257	421 257	424 254 421 257	425 248
425 248	423 248	423 249	424 248	
407 221				
403 205				403 205(?)
376 197				
368 197				

185 258

Concave, eye extracted	is6nd50d1053	is6nd100d1053	is6nd503d1053	is6 major controls set (concave)
336 251	333 250	337 245 332 250	341 244 335 249	336 251
369 239	370 233	369 232	369 233 370 236	369 239
419 246			372 237 367 238	
407 226	402 227	402 227	403 228	407 226
361 201				
331 203	326 205 328 208 326 209 331 211	325 205 329 207 326 209 330 211	326 204 329 207 333 209 331 210	331 203
285 168	286 170	290 169 287 170	288 168 284 171	285 168
276 186	269 182 273 182 274 185 277 186	272 182 274 185 277 186	274 186	276 186
261 212	262 215	264 213 260 216	264 211 261 214 259 215	
247 221	246 225	248 225		247 221
217 259			214 254	
256 262	251 257 257 258	255 258	251 258 255 258	
266 265	265 260 267 261	265 260		
286 279	285 275 288 277 284 279 285 282	287 277 285 283	283 274 284 278 283 282	286 279
306 294	304 289	303 288	300 290 306 290	306 294
		356 315	343 149 274 173	

IS6 in Blocky Noise

Convex eye extracted vol=20698 per=906	**is6ndis5 vol=22544 per=1003**	**is6ndi52 vol=24452 per=1057**	**is6ndi53 vol=26404 per=1064**	**is6ndi54 vol=28386 per=1070**	**is6 major controls set (convex)**
353 116	356 114 357 117	360 126	361 127	364 128	353 116
315 123			300 125 298 128	299 124	
243 182	240 183	237 185 248 194	234 185 247 195	233 185	243 182
248 216					
174 248	172 243 173 252 176 255	169 241	171 236 168 239 167 248	168 236 166 250 173 261	174 248
237 264	225 266	225 268	223 270		
267 290	263 286 264 290 267 296	261 289	430 246	257 282 255 284	267 290
348 320		385 311	387 315 385 320		
381 314	383 311 381 318	379 297	370 327	389 315 388 318 366 330	381 314
408 259					
421 256	426 249 426 253	426 243 428 246		426 263 422 266	425 248
425 248				431 244	
407 221		418 230			
403 205				407 194 405 194 409 196	403 205(?)
376 197					

368 197					
	false hits	false hits 397 263	false hits 369 279	false hits 370 277 399 269	
	343 180		288 137 357 144	320 108 182 227	

Concave, eye extracted	**is6ndis5d53.clu**	**is6ndi52d53.clu**	**is6ndi53d53.clu**	**is6ndi54d53.clu**	**is6 major controls set (concave)**
336 251	338 249	340 250		346 254	336 251
369 239	368 238	372 242 367 240	364 243 379 247	362 245	369 239
419 246		415 240 417 243		420 241	
407 226	406 229	407 226	410 225 418 241		407 226
361 201					
331 203	329 203 333 207	332 196 330 200 332 203 335 205	336 199 348 200		331 203
285 168	287 167 269 172 271 172 274 172	286 165 270 170	284 165 271 169 275 169	281 164 272 167	285 168
276 186	271 185 274 189 273 192 273 194	264 188 268 190 268 192 270 194	262 191 264 196		276 186
261 212	259 212	268 200 262 208 257 211			
247 221					247 221
217 259	214 255 218 256	213 257 217 257	216 261 343 253		
256 262	252 259	254 262 249 261	250 262	251 264	
266 265	276 270	273 272 273 278	265 268 270 275	267 275	

286 279	277 278				286 279
306 294	301 291	297 293	300 294	300 296	306 294
	307 292	300 293			
		305 293			
	false hits	false hits	false hits	false hits	
	364 317	350 133	329 116	333 113	
	333 186	346 149	352 133	330 114	
	338 256	351 244	301 137	354 134	
	220 257	233 261	348 148	299 136	
	234 260	356 280	294 150	293 147	
		377 307	256 190	340 196	
			232 264	255 193	
			358 279	260 199	
			367 289	396 200	
			368 291	229 222	
			378 307	191 233	
			325 309	179 237	
			329 312	380 249	
			358 320	413 260	
				405 261	
				232 265	
				359 277	
				368 287	
				379 305	
				327 314	
				359 322	

B.5 Sequenced Perimeter Feature Sets

Sequence information retained for use with recognition algorithms

Extracted Features

Is1b.img (eye extracted)

x	y	type	x	y	type
353	288	1	361	371	0
234	340	0	404	364	1
147	345	1	361	349	0
228	362	0	358	292	1
352	431	1			

is1b.img (gab, diffre extracted)

x	y	type	x	y	type
339	293	1	367	369	0
227	341	0	393	357	1
155	349	1	359	343	0
235	363	0	352	290	1
355	419	1			

r(gab,diffre extracted)

x	y	type			
341	289	1	226	363	0
261	331	0	357	422	1
221	338	0	368	372	0
165	337	1	404	361	1
154	352	1			

is6b.img (eye extracted)

x	y	t	x	y	t
381	314	1	353	116	1
348	320	1	331	203	0
306	294	0	361	201	0
267	290	1	368	197	1
286	279	0	376	197	1
266	265	0	403	205	1
256	262	0	407	221	1
237	264	1	407	226	0
217	259	0	419	246	0
174	248	1	425	248	1
247	221	0	421	256	1
248	216	1	408	259	1
261	212	0	369	239	0
276	186	0	336	251	0
243	182	1			
285	168	0			
315	123	1			

is6b.img (gab, diffre extracted)

x	y	t	x	y	t
334	117	1	421	248	1
311	129	1	406	219	0
278	170	0	391	203	1
255	187	1	359	203	1
272	195	0	343	205	1
256	214	0	329	211	1
244	223	1	331	196	0
184	253	1	344	115	1
224	259	0			
240	261	1			
264	263	0			
282	281	0			
273	294	1			
313	297	0			
376	308	1			
340	246	0			
380	243	0			

r (gab, diffre)

x	y	t
335	114	1
311	125	1
273	168	0
252	190	1
265	201	0
268	202	1

Is12b.img (eye extracted)

x	y	type
392	431	1
310	442	1
296	408	1
295	315	1
372	308	1
393	332	1

is12b.img (gab, diffre extracted)

x	y	type
296	326	1
300	414	1
316	440	1
392	425	1
361	310	1

r (gab, diffre)

x	y	type
293	321	1
301	425	1
315	443	1
395	422	1
390	326	1

booze.img (eye extracted)

x	y	type	x	y	type	x	y	type
349	404	1	204	238	0	381	221	1
346	408	1	174	201	1	295	266	0
329	390	1	138	128	1	338	299	1
310	331	0	218	186	0	359	339	1
273	304	0	222	184	1	351	359	1
268	390	1	242	197	0	349	358	0
234	358	1	269	192	0	348	398	0
219	299	0	242	154	1	323	218	0
190	260	1	289	189	0	315	209	0
213	273	0	344	187	1	312	322	0

booze (actual via gab,diffre)

x	y	type	x	y	type	x	y	type
140	137	1	344	397	1	372	209	1
208	248	0	333	381	1	279	183	0
212	264	0	337	368	1	245	164	1
206	268	0	350	356	0	258	196	0
193	270	1	356	343	0	227	193	0
223	310	0	349	327	1	206	183	0
231	342	0	342	311	1	191	169	0
268	384	1	333	297	1	171	152	0
278	303	0	312	282	1	157	141	0
313	336	0	304	259	0	142	129	1

B6: Example Clustered Set

Islb.img (eye extracted)

x	y	type
353	288	1
147	345	1
352	431	1
404	364	1
358	292	1
234	340	0
228	362	0
361	371	0
361	349	0

mhmmt extracted

x	y	type
241	340	1
149	344	1
373	361	1
235	362	1
383	363	1
238	363	1
392	363	1
397	363	1
360	370	0

rin extracted

x	y	type
354	289	1
151	339	1
147	346	1
358	349	1
377	353	1
386	354	1
301	364	1
370	368	1
374	368	1
350	430	1
360	371	0

Appendix B.7 Sequenced Object Models (About Centroid of Features)

Is1 (eye extracted series)

Locus		Type	Angle
x	y		(rads)
228	362	0	2.9843652
352	431	1	1.0974402
361	371	0	0.4072562
404	364	1	0.1582402
361	349	0	0.0000002
358	292	1	-0.8709032
353	288	1	-0.9567772
234	340	0	-3.0237212
147	345	1	-3.1170582

Is1 (mhmmt extracted), Centroid of Features [307,358]

Locus		Type	Angle
x	y		(rads)
235	362	1	3.0860942
238	363	1	3.0692552
360	370	0	0.2226612
383	363	1	0.0656952
392	363	1	0.0587562
397	363	1	0.0554992
373	361	1	0.0454232
241	340	1	-2.8753412
149	344	1	-3.0532162

Is1 (rin extracted), Centroid of Features = [320,357]

Locus		Type	Angle
x	y		(rads)
301	364	1	2.7886022
350	430	1	1.1808782
360	371	0	0.3366752
370	368	1	0.2165502
374	368	1	0.2009542
386	354	1	-0.0454232
377	353	1	-0.0700612
358	349	1	-0.2074962
354	289	1	-1.1071492
151	339	1	-3.0354842
147	346	1	-3.0780942

B.8 Elementary Sequence Checking, using Type Data Only and Procession About the Perimeter

```
C:\VIG2\ED>merlyn testis1.sr1 is1snake.cd r1.dat j1.dat

r              j
1 0 0          4 0 0
               7 0 0
               9 0 0

3 0 0          4 0 0
               6 0 0
               9 0 0

5 0 0          6 0 0
               8 0 0
               13 0 0
8 0 0          9 0 0
               14 0 0
               16 0 0

10 0 0         13 0 0
               16 0 0
               18 0 0
```

```
C:\VIG2\ED>merlyn testis1.sr1 is4snake.cd r1.dat j1.dat

r       j
0 0 0   0 0 0
        4 0 0
        6 0 0
        8 0 0

3 0 0   3 0 0
        5 0 0
        7 0 0
        9 0 0

5 0 0   5 0 0
        7 0 0
        9 0 0
        13 0 0

7 0 0   7 0 0
        9 0 0
        13 0 0
        15 0 0
```

testis1.sr1 with is6snake.cd

r	j	j cont.
-2 0 0	-2 1 0	20 0 0
	-1 0 0	21 1 0
	1 0 0	22 1 0
	4 0 0	23 0 0
	5 0 0	24 0 0
	7 0 0	25 1 0
	13 0 0	26 0 0
	14 0 0	27 1 0
	18 1 0	28 0 0
	19 0 0	
-5 0 0	-5 1 0	17 0 0
	-4 0 0	18 1 0
	-3 1 0	19 1 0
	-2 0 0	20 0 0
	-1 0 0	21 0 0
	1 0 0	22 1 0
	4 0 0	23 0 0
	8 0 0	24 1 0
	10 0 0	25 0 0
	16 0 0	
-4 0 0	-4 1 0	18 0 0
	-3 0 0	19 1 0
	-2 1 0	20 1 0
	-1 0 0	21 0 0
	0 0 0	22 0 0
	3 0 0	23 1 0
	5 0 0	24 0 0
	9 0 0	25 1 0
	12 0 0	26 0 0
	17 0 0	
-11 0 0	-11 1 0	-1 1 0
	-10 0 0	4 0 0
	-9 1 0	5 0 0
	-8 0 0	10 0 0
	-7 0 0	14 0 0
	-6 1 0	15 0 0
	-5 0 0	17 0 0
	-4 0 0	18 1 0
	-3 1 0	19 0 0
	-2 0 0	

testis1.sr1 with bozsnake.cd

r	j	j cont.
2 0 0	4 0 0	23 1 0
	8 0 0	24 0 0
	10 0 0	25 1 0
	12 0 0	26 0 0
	17 0 0	27 1 0
	18 0 0	28 1 0
	19 1 0	29 0 0
	20 0 0	30 1 0
	21 0 0	31 0 0
	22 0 0	
-4 0 0	-4 0 0	16 0 0
	-3 1 0	18 0 0
	-2 0 0	19 1 0
	-1 1 0	20 0 0
	4 0 0	21 1 0
	5 0 0	22 1 0
	6 0 0	23 0 0
	12 0 0	24 1 0
	14 0 0	25 0 0
	15 0 0	

testis4.sr1 with is1snake.cd

r	j
2 0 0	∅

testis4.sr1 with is4snake.cd

r	j
-1 0 0	-1 0 0
	7 0 0
	8 1 0
6 0 0	6 0 0
	14 0 0
	15 1 0
-3 0 0	-3 0 0
	-2 1 0
	-1 0 0
4 0 0	4 0 0
	5 1 0
	6 0 0

testis4.sr1 with is6snake.cd

r	j	j cont.
-1 0 0	-1 1 0	19 1 0
	0 0 0	20 0 0
	1 1 0	21 0 0
	2 0 0	22 1 0
	6 0 0	23 1 0
	10 1 0	24 0 0
	14 0 0	25 0 0
	15 0 0	26 1 0
	16 1 0	27 0 0
	17 1 0	28 1 0
	18 1 0	29 0 0
-8 0 0	-8 1 0	12 1 0
	-7 0 0	13 0 0
	-6 1 0	14 0 0
	-5 0 0	15 1 0
	-4 0 0	16 1 0
	-3 1 0	17 0 0
	-2 0 0	18 0 0
	-1 0 0	19 1 0
	3 1 0	20 0 0
	7 0 0	21 1 0
	10 1 0	22 0 0
-10 0 0	-10 1 0	6 0 0
	-9 0 0	8 1 0
	-8 1 0	10 1 0
	-7 0 0	11 0 0
	-6 0 0	14 1 0
	-5 1 0	15 0 0
	-4 0 0	16 0 0
	-3 0 0	17 1 0
	-2 1 0	18 0 0
	-1 0 0	19 1 0
	1 1 0	20 0 0
-18 0 0	-18 1 0	-7 1 0
	-17 0 0	-6 1 0
	-16 1 0	-5 0 0
	-15 0 0	-4 1 0
	-14 0 0	-3 0 0
	-13 1 0	-2 0 0
	-12 0 0	-1 1 0
	-11 0 0	1 1 0
	-10 1 0	4 0 0
	-9 0 0	5 1 0
	-8 1 0	7 0 0

testis4.sr1 with bozsnake.cd

r	j	j cont.
-17 0 0	-17 0 0	-7 0 0
	-16 1 0	-6 1 0
	-15 0 0	-5 1 0
	-14 1 0	-4 1 0
	-13 1 0	-3 1 0
	-12 1 0	-2 0 0
	-11 0 0	-1 0 0
	-10 1 0	2 0 0
	-9 0 0	7 0 0
	-8 0 0	8 1 0
-19 0 0	-19 0 0	-9 0 0
	-18 1 0	-8 1 0
	-17 0 0	-7 1 0
	-16 1 0	-6 1 0
	-15 1 0	-5 1 0
	-14 1 0	-4 0 0
	-13 0 0	-3 0 0
	-12 1 0	-2 1 0
	-11 0 0	-1 0 0
	-10 0 0	0 0 0

Appendix C: Work Published During the Project

Appendix C.1: S J Rees, B F Jones, "Recognition of 2-D Shapes using Set Erosion", Proceedings of the SPIE Vol. 1607 Intelligent Robots and Computer Vision X: Algorithms and Techniques, D Casasent (Ed.), Boston, Nov 1991, pp 206 -216.

Recognition of 2-D Shapes using Set Erosion

S J Rees, B F Jones, Department of Electronics and Information Technology, The Polytechnic of Wales, Pontypridd, Mid Glamorgan, UK.

Keywords Image Processing / Pattern Recognition / Set Erosion / Orientation

ABSTRACT

Set erosion is an efficient algorithm which has been used to recognize shapes irrespective of orientation, translation and scaling. The technique has successfully recognized complex shapes, even when two shapes overlap. The uncertainty in the measured estimate of scaling rose to 8% from the 2% figure obtained for separate shapes.

The image picture is segmented between shape and background. The orientation and length of each side or arc on the perimeter of the shape is extracted using a chain code based technique and a set composed of the orientation and angle information formed. This set of data is then morphologically eroded with the orientation/angle spectra of each of the shapes in a pre-defined library of reference shapes, the reference shapes being scaled to the acquired image data. If the set of angle/weight reference data is found to be contained within the acquired set, the reference shape is recognized as being part of the solution. The required shift of the reference spectrum to match the acquired spectrum yields the rotation of the shape relative to the reference data. Scale information is generated as part of the pre-conditioning of the reference data prior to the erosion process. Location data are generated by tagging extracted vertices within the chain code extraction of side data.

1. INTRODUCTION

One means of recognizing and cataloguing shape is to measure the lengths of the sides and their relative orientations. These fundamental parameters may be estimated in a variety of ways. The relative lengths of the sides of a shape, and their relative orientations are of course invariant under 2-D rotation, translation and scaling[1]. The erosion technique developed in this paper requires a set composed of an angle (representing the orientation of a point on the shape relative to some reference axes) and a weighting (representing the number of occurrences of the particular angle on the perimeter of the shape). Using this information the content of an image can be analysed to determine the presence, scaling and orientation of any defined shape by comparison with pre-stored library data.

Congruency and similarity in triangles are well known elementary theorems. The technique used in this project is a development of similarity to the general case of arbitrary shapes.

Extraction of the orientation of the sides of a shape relative to a fixed set of axes gives a measure of the relative orientations between the sides. Coupled with a weight (pixel count) measure of relative side length, sufficient information is available to provide shape recognition through similarity. The erosion of the acquired set of side orientations and weights by the reference set for a particular shape can prove similarity and extract the necessary recognition, scale and orientation data.

2. SET EROSION

The concepts of dilation and erosion are basic ideas of mathematical morphology. Shapes and objects in images may be considered as sets of points differing in some way from the background complementary set. Excellent discussions of morphology and examples of its use in image processing may be found in Haralick et al[2] and Maragos[3].

0-8194-0744-5/92/$4.00

The erosion of a set A by a set B is denoted by AΘB and is defined by

$$A\Theta B = \{ x\varepsilon E^{n} | x+b\varepsilon A \text{ for every } b\varepsilon B \} \quad ...[1] \qquad n$$

which may be written as

$$A\Theta B = \{x\varepsilon E^{n} | (B)x\subseteq A\} \qquad ...[2] \qquad n$$

In a very real sense the structuring element B is a spatial probe. If B translated by x can be contained in A, x must belong to the erosion AΘB.

Re-writing [2] in its computationally efficient form,

$$A\Theta B = \bigcap_{b\varepsilon B} (A)_{-b} \qquad ...[3]$$

If set A is the acquired shape information and set B is a pre-defined library set of shape data, this offers an algorithmically efficient method of determining the presence of set B within set A.

3. DEFINITION OF SHAPE RECOGNITION USING SIMILARITY

Definition: Similarity in 2-D Shapes

Two simply connected shapes are similar if all sides of the shapes are at the same relative orientations and in the same proportionate lengths.

Definition: Similarity in 3-D Objects

Two simply connected objects are similar if all faces of the objects are at the same relative orientations and of the same proportionate areas.

Definition: Similarity in N-D Objects

Two simply connected objects are similar if the surfaces of the objects are at the same relative orientations and have the same relative proportions in (N-1) dimensions.

The generalization to N-dimensions permits the inclusion within sets of extension data such as colour and textural information.

To prove similarity we therefore require:-

(i) Proof that relative orientations of sides are the same.

(ii) Proof that relative weights of sides are the same.

Set erosion may prove both orientation and weight (length) equivalence, yielding a measure of the relative scaling of the shapes and their relative orientations.

Proof of similarity therefore solves the recognition problem. The remaining problem of location may be solved by noting the position of the first vertex found on the acquired shape. Retroactively fitting the reference image after analysis will give positive confirmation of the results.

4. BASIC TECHNIQUE

The orientation spectrum in 2-D of a shape is defined as the histogram of the edge normals at all points on the pattern perimeter plotted against their orientation to a particular axis set.

A number of binary shapes are shown in figure (1). Extraction of perimeter normal orientation and weighting proportional to side length yields the angle spectrum tables of figure (2).

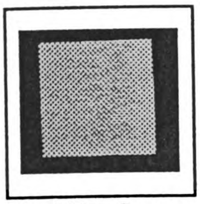


Fig 1(a): Square

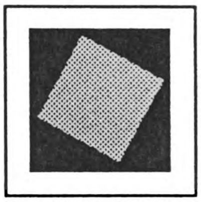


Fig 1(b): Rotated Square

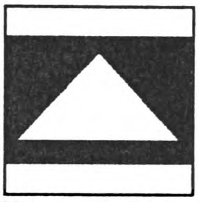


Fig 1(c): Triangle

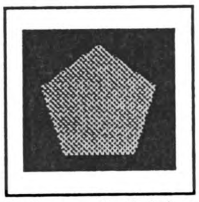


Fig 1(d): Pentagon

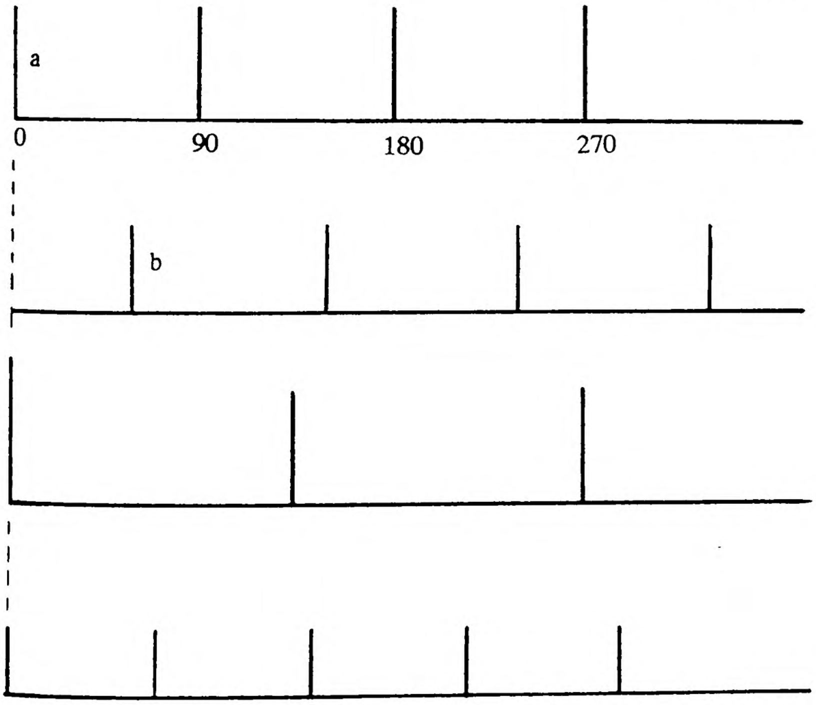


Fig (2): Orientation Spectra for Simple Shapes

(a) Square

(b) Rotated Square

(c) Triangle

(d) Pentagon

Consider the square and its rotated analogue. Each perimeter normal is shifted through an angle ϕ relative to the original square and is scaled to give a side weighting (length) of b. Set erosion offers a means of identifying whether the shape is actually a rotated image of the square as described below.

Hand Worked Example

Applying equation [3] to figures 2(a) (set A) and 2(b) (set B):

$A = \{(0,a),(90,a),(180,a),(270,a)\}$

$B = \{(\phi,b),(90+\phi,b),(180+\phi,b),(270+\phi,b)\}$

We obtain:

$A\Theta B = \{(-\phi,b-a)\}$

This confirms that figure 1(b) is figure 1(a) rotated through ϕ and with a scaling difference of (b-a). In practice, in order to differentiate between shapes which have the same angle set with differing weights (such as squares and rectangles), the probing reference set must be prescaled.

The result therefore becomes:-

$A\Theta B' = \{(-\phi,(b-a*((b/a) - \delta)))\}$ where δ = 2% of b/a

A successful result is indicated by a small positive value for the result of the erosion.

Erosion of set A by set C (angle spectrum for the triangle of figure 2(c)) yields

$A\Theta B = E$ (the empty set)

This confirms A and C are not the same shape. The erosion process may be terminated as soon as the empty set is obtained. A library of reference shapes is created either analytically or by direct extraction from the system. The unknown shape is eroded sequentially until recognition is achieved.

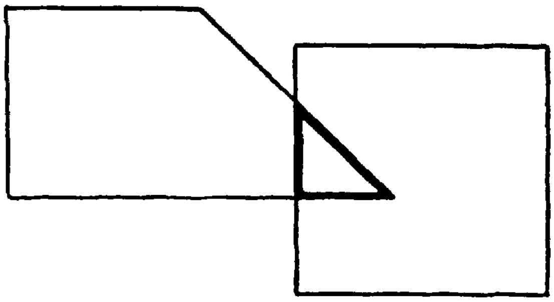
Applying the technique to curved surfaces demands more storage space and the processing takes longer due to the higher volume of intermediate data.

The set of side weights and orientations can be generated in a variety of ways. In the practical implementation of the technique a chain code based algorithm has been developed which yields a high accuracy and stability under rotation.

5. THE PROBLEMS OF OBSCURATION

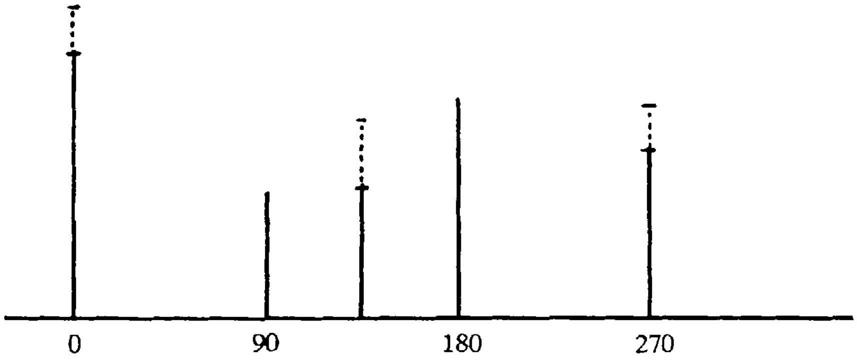
The classic problem with many pattern recognition algorithms stems from partial obscuration of the shapes under test. Many techniques use sophisticated rules to identify sets of parameters which may define characteristics of the object in a way which permits recognition from a limited subset of features[5,6,7].

When any two objects overlap, the area of overlap forms a closed polygon. Figure(3) below illustrates this with a simple example in which the enclosed polygon is a triangle. As a direct result of this fact, the sum of the x and y components of orientation of the sides of the polygon must resolve to zero.



Figure(3): Trapezoid overlapping Square

The closed polygon is obtained by postulating two shapes according to the observed angle content of the shape and subtracting the observed shape from the sum. If the weights in the x and y directions both sum to zero, confidence in the recognition is increased. Confirmation may be achieved by reconstructing the observed shapes and subtracting from the original. This is illustrated in figure (4).



Figure(4): Orientation Spectrum for Trapezoid overlapping Square

Furthermore, the technique is commutative. Provided some proportion of the edges of overlapping objects are available, several objects may be identified, located and given an estimate of orientation relative to the reference axes and shape sets.

The uncertainty of both recognition of the shapes and estimation of rotation, translation and scaling increases with the degree of obscuration. This remains acceptable while two sides of all the overlapped shapes under test are visible. Scaling is not possible without at least two sides visible for each partially obscured shape, but a generic description is possible without this.

In the practical implementation of this algorithm, two adjacent sides of a shape must be visible in their entirety to permit correct scaling and therefore unambiguous recognition of a particular shape rather than a generic family identification. The possible presence of a particular reference shape is tested using the angle only erosion, then the reference data is scaled, aligned and subtracted from the acquired spectrum of angles/weights. This process is repeated until possible shapes have been identified for the various overlapping shapes in the image data. The polygon summimg is then applied to confirm or disprove the presence of the various reference shapes in the overall solution.

6. RECOGNITION AND UNCERTAINTY

The recognition process is dependent on the information content of the source image data. Increasing thesize of the shape under test within the image, increasing its resolution in pixels, reducing noise levels, increasing the shape/background contrast, colour and textural information, and increasing constraints on the allowed shape domain all reduce the uncertainty of the recognition process.

Let us consider the facets of the recognition process and look at the unique factors affecting uncertainty within the set erosion based technique.

The chain code description contains the full information content of the shape perimeter after preprocessing. The angle spectrum calculated from this description loses only the concise description of edge sequence, and, where a means of resolving contention is required, this is avalable in the chain code file edge sequence.

The preprocessing of the image to eliminate edge holes and smooth the edge profile results in some measure of image distortion (this is its objective). Although this reduces the purity of the source data, its effect is most significant with textured image data such as the crenellated edge of a coin. The closing technique does result in elimination of a few corner pixels, and therefore has an adverse effect on scale estimation. This typically involves removal of one or two pixels when the edge length data is generated.

Binarization of the image data should introduce only minimal corruption of the source image if the correct thresholding is chosen. Colour and textural data are of course eliminated.

The recognition algorithm itself introduces uncertainty into the process, depending primarily on scale and orientation of sides estimation accuracy. Quantisation effects caused by digitising what is in essence a continuous domain data item (ie the shape) through the CCD acquisition process is one significant effect.

Orientation

Quantisation yields

$\phi_{error} = \arctan(1/n)$ where n is the length of the side in pixels.

Scale

Scale uncertainty due to quantisation yields:-

$S_{error} = [n/m - (n-1)/m] = 1/m$ where m is the length of the scaling side in pixels.

or $(n/(n-1))\%$

The uncertainty measure should offer a guide to the acceptability of the results obtained in the erosion process. It is therefore useful to create two items of data, an uncertainty budget for the shape under test, and, in the case of a constrained set of possible shapes, some indication of the closeness of the shape match to the matching reference shape and to the next closest shape in the reference set.

7. ORIENTATION SPECTRUM EXTRACTION

The reference tables of angle spectra for shapes can be obtained directly from an acquired image through the use of various techniques such as the Sobel operator[4] or chain coding. Practical images obtained in this way offer a simple and cheap means of creating a reference library. The angle spectrum stored is effectively a measure of the spatial frequency components in the shape. Discrete images obtained via an image processor/camera combination do not exhibit the perfect corners and continuous slopes displayed in geometry. The angle spectrum will therefore contain additional components (termed the fine structure) for even simple symmetrical shapes when template based techniques are used.

7.1 Chain Code Extraction Technique

The chain code description of the perimeter of the shape under test is generated by tracking a 3x3 template around the edge of the shape in an anti-clockwise direction. Four directions are defined, as illustrated in figure(5). Each change from a horizontal or vertical run of pixels is noted as a positive or negative change in the particular direction of track, dependant on whether an increase or decrease in the size of the shape is implied. This information is used to determine the quadrant for the orientation calculations. Possible ambiguities are resolved in the rule set for change notation.

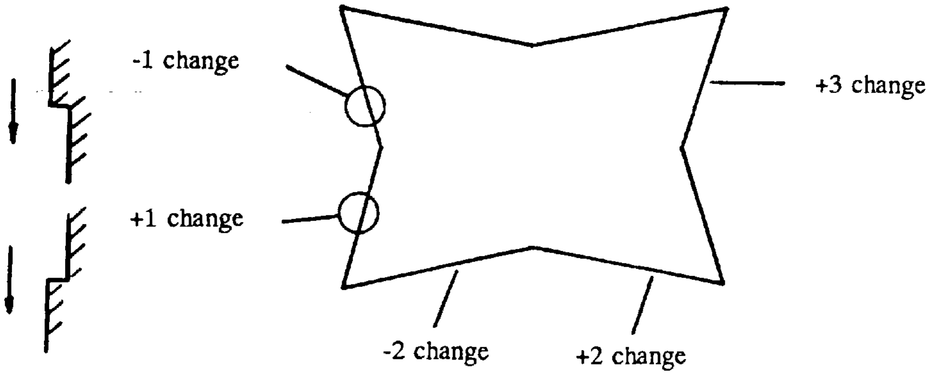


Figure 5: Conventions for Chain Code Extraction Technique

The chain code data contains a full description of the shape perimeter. Sections of the perimeter belonging to the same edge are amalgamated before calculation of orientation, the pixel count and change data being summed.

Orientation calculation is made with respect to the pixel grid using elementary trigonometry:-

Edge orientation = arctan ($\delta y/\delta x$)

where y,x are the accumulated edge pixel totals in the Y and X directions (taken as the axes of the pixel array).

8. THE APPLICATION OF THE RECOGNITION ALGORITHM TO REAL IMAGES

The image was acquired using a Pulnix CCD camera with 512x512 pixel grid and an Imaging Technologies PC Plus frame grabber. The image was segmented and, as colour was not used as a data extension, the shape within the picture was binarized. A morphological closing using a 7x7 block structuring element was applied to reduce edge holes, noise and irregularities.

The chain code description was generated. Note that this description contains sufficient information to regenerate the perimeter of the shape. The chain code description of the perimeter was then compacted to form the side orientation (angle) and length (weight) set required for the erosion algorithm.

The acquired data were eroded with reference shape data sets. The scaling information required to provide a particular solution was extracted as the ratio of the longest sides in the acquired and reference data. Where a match was not found, and the constrained set of permitted shapes contained the possibility of obscuration, the obscuration arbitration process was undertaken. The acquired data was eroded on an angle only basis, and possible results stored. The resulting data was tested for an overlapped match by summing the missing parts of the shape pairs in the x and y component directions. A successful recognition was indicated by a zero (actually less than the permitted threshold) sum in both planes.

8.1 Results

The results shown in Table(1) were produced in accordance with the process described above for the shapes shown in figures (6) (a) - (i). Table (2) lists the practical orientation estimation results obtained for a series of tests in single degree steps with a simple shape. Table (3) lists the practical scale estimation results for a series of tests with 2.5mm variations in size with a 50mm primary shape. Table (4) lists the results obtained for the overlapping pairs of shapes shown in figures (7) - (8).

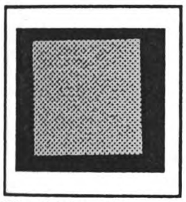


Fig 6(a): Square

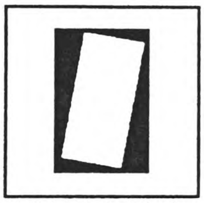


Fig 6(b): Rectangl

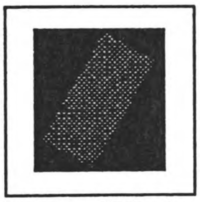


Fig 6(c): Rectang2

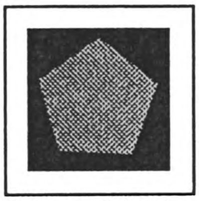


Fig 6(d): Pentagon

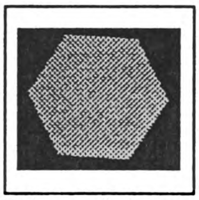


Fig 6(e): Hexagon

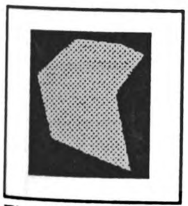


Fig 6(f): Polygon

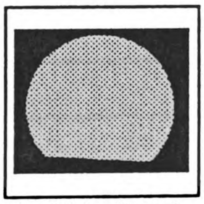


Fig 6(g): Chord

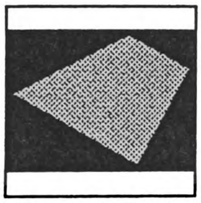


Fig 6(h): Trapezoid

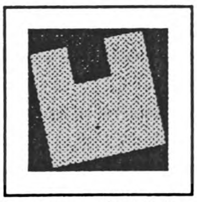


Fig 6(i): Uconcave

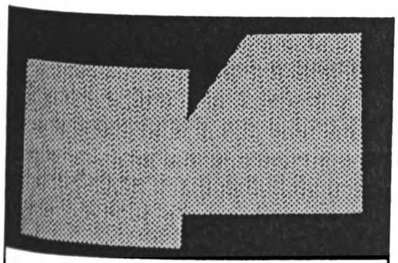


Fig 7: Square obscuring Trapezoid

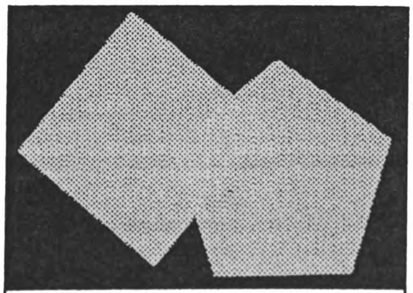


Fig 8: Square and Pentagon

Table (1): Recognition of Primary Shapes

Shape	Recognized	Orientation		Scale	
		Actual	Calc.	Actual	Calc.
Square(6(a))	y	0	0	0.95	0.96
Rectangle1 (b)	y	10	11	3.28	3.31
Rectangle2 (c)	y	20	22	1.48	1.50
Pentagon (d)	y	70	70	1.77	1.75
Hexagon (e)	y	4	4	1.00	1.03
Polygon (f)	y	20	21	1.60	1.57
Chord (g)	y	10	8	1.20	1.17
Trapezoid (h)	y	324	322	0.95	0.97
Uconcave (i)	y	10	11	0.97	0.95

Table (2): Orientation Resolution Tests

Algorithm Orientation	Actual Orientation (+/- 0.5 deg)
1	0
0	1
1	2
3	3
5	4
11	10
32	30
44	45

Table(3): Scale Resolution Tests

Shape = Pentagon

Reference library pentagon has side lengths of 100 pixels, equivalent to 25mm physical length

Algorithm Scale	Actual Scale
1.77	1.80 = 45.0mm
1.87	1.90 = 47.5mm
1.98	2.00 = 50.0mm

Table(4): Characterization of Occluded Shapes

Shape	Overlap (%)	Recognized		Orientation		Scale	
				Actual	Alg.	Actual	Alg.
Figure (7): Trapezoid and	10	Both	Trapezoid	-2	-2	1.80	1.70
Square			Square	0	1	1.80	1.67
Figure (8): Pentagon and	15	Both	Pentagon	2	4	1.00	1.05
Square			Square	50	52	1.80	1.65

9. CRITIQUE OF RESULTS

All the main processing algorithms were written and implemented on a standard PC-AT equivalent (286-12). The library of shapes now numbers several hundred, elements of the library having been generated both by hand and by direct extraction from image data.

Recognition

The recognition algorithm successfully resolved shapes separated by a limited difference in geometry with a high selectivity. Independence of rotation is demonstrated, and size differences produced most variance in the uncertainty of recognition. The reasons for this have been indicated; the decreased side length in pixels reduces both orientation and scale accuracy, and therefore degrades the recognition process.

Orientation

Orientation estimated are dependent to some extent on shape size, but with reasonable shape data orientation estimates to within 2 degrees of absolute accuracy were obtained against measured data. The limiting factors in orientation estimation were basic pixel resolution of the CCD array, and available memory storage space within the host computer system. Higher resolution is available with increased processing time and increased memory usage.

Scale

Scale estimates were accurate to within 2% for simple shapes, but proved a more complex issue with occlusion. In the occluded case the maximum error with successful recognition of both shapes was found to be around 8 to 10%.

Timing

The duration of the recognition process is of the order of 2 to 3 seconds using a standard PC-AT. The current impementation of the preprocessing and angle extraction algorithms requires around 20 seconds to acquire the required data. This will be improved drastically by the use of an image processor for the front end processing, and further still by parallel implementation.

It should be noted that the preprocessing and angle extraction process massively compact the image data. A recognition is achieved from a data file of less than 1Kbytes. The typical image file contains more than 250Kbytes of information.

Initial experimental results have been obtained using grey scale images. Work is ongoing to extend the technique for use with 3-D objects. Over 100 shapes are currently defined within the library.

Shapes may be analytically defined, or may be created by direct acquisition into the system. Care must be taken to ensure directly acquired shapes are definitive.

10. REFERENCES

(1) Partial Shape Classification using Contour Matching in Distance Transformation Hong-Chih Liu and Madyam D Srinath, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12 No11 Nov 1990, pp1072-1079.

(2) Image Analysis using Mathematical Morphology R M Haralick, S R Sternberg and X Zhuang, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 9 No4 July 1987.

(3) Pattern Spectrum and Multiscale Shape Representation P Maragos, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11 No7 July 1989.

(4) Image Processing Gonzalez and Wintz, Addison Wesley,1977.

(5) Skeletonization and Distance Transformation by Greyscale Morphology Frank Y. Shih and O. Robert Mitchell, Proceedings of the SPIE, Vol 849 1988, pp 80-86.

(6) Three-Dimensional Shape Analysis using Local Shape Descriptors Timothy P Wallace, Owen Robert Mitchell and Keinosuke Fukungawa, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 3 No 3 May 1989, pp310-323.

(7) Orientation and Recognition of Both Noisy and Partially Obscured 3-D Objects from Single 2-D Images Diane P Illing, PhD Thesis, Dept of Mathematics and Computing, The Polytechnic of Wales, Pontypridd, Mid Glamorgan Sept 1990.

Appendix C.2: S J Rees, B F Jones, "Direct Feature Extraction using Conditional Morphological Operators", Proceedings of 10th International Conference on Systems Engineering, Coventry Univeristy, Sept 1994, pp998-1005.

DIRECT FEATURE EXTRACTION USING CONDITIONAL MORPHOLOGICAL OPERATORS

S J Rees, B F Jones, T D Jones

Image Processing and Medical Electronics Group, The University of Glamorgan, Pontypridd, Mid Glamorgan.

Morphological operators have found wide acceptance for image enhancement and analysis. Following from initial work on set erosion for image analysis an operator has been defined allowing a conditional erosion and dilation to be generated for binary and grey scale pictures. The more sophisticated operations have been derived using conditional erosion and dilation, including opening, closing and a conditional hit and miss transform. This paper describes the application of the greyscale operators for direct feature extraction and segmentation. Results are shown for a variety of objects including aircraft models and surgical wounds.

1. INTRODUCTION

The development of conditional and fuzzy mathematical morphology techniques has progressed over the last few years [1,2]. Application of morphology to skeletonization and recognition has been undertaken with some success [3,4,5]. This paper describes the use of conditional morphological operators for direct extraction of features in grey-scale images for the purpose of abstracting feature sets for object recognition purposes. This is part of a larger project aimed at using conditional morphology for both the front end feature extraction and the object recognition stages, yielding a consistent algorithmic architecture of a suitable form for implementation as a scalable hardware architecture. Amenity for hardware implementation is important for real time processing: the main attraction of morphological techniques is their logical simplicity, the main problem their frequently lengthy processing times in real applications.

The selection of appropriate templates for extracting particular types of features is an important part of morphological processing, in that the template chosen is a spatial probe used to produce characteristic results. The correct template is a determinant to the discrimination of the algorithm as a whole, and defines the possible features that may be extracted.

2. THEORY: THE R-J OPERATOR

Let us first define the maximal intersection of several sets or subsets as the set of points which occur in the largest numbers of the sets. In other words, consider seven sets which have no common element. If an element exists in any six of the seven sets then it is the maximal intersection. Obviously more than one maximal intersection may occur.

Where a datum exists in all the sets then the maximal intersection and the true intersection are both equal to that point. Points of maximal intersection do not have to belong to the same subset of the sets.

In the binary case, this needs no further interpretation: a point is either present or absent and the degree of occupancy is therefore clear.

2.1 Binary Sets

For two binary sets A and B, the RJ operator (denoted as @) is defined as:

$$A@B = \underline{R},J$$

$$\text{where } \underline{R} = \{r{:}N \mid r = \text{maximal intersections of } (A)_{-b}\ \forall b \in B\} \qquad [3]$$

ie $\underline{R}$ is the set of points that occur in the largest number of sets $(A)_{-b}$. This may be further written in a summation form for calculation purposes, as shown below:

Let

$$R = \{r \mid r = \sum_{b \in B}((a)_{-bi} \cap (A)_{-b})\ \ \forall a \in A, \forall b_i \in B\} \qquad [4]$$

Then

$$\underline{R} = \max(R) \qquad [5]$$

A separate J set occurs for each element of $\underline{R}$, referred to as J_r.

Then

$$J = \{J_r\} \forall r \in \underline{R} \qquad [6]$$

$$= \{j{:}N \mid j = r+b \text{ where } r \notin (A)_{-b},\ \forall b \in B\} \forall r \in \underline{R} \qquad [7]$$

The set $\underline{R}$ indicates the loci of highest occupancy of A by B. The J_r sets indicate the necessary additions to permit that occupancy to become complete.

A conditional erosion process can now be defined in terms of the RJ operator results. The symbol © will be used to denote this operation. A constraint is chosen for the number of components of B required to be present in A, thus providing the condition for conditional erosion. For any arbitrary threshold th where th <= no. of components of B:

$$A©B = R \mid r>=th \qquad [8]$$

Note that this conditional erosion represents a loosening of the constraints of formal erosion. It is logical that a conditional dilation would represent a tightening of the constraints on formal dilation. Such a constraint would be an increased number of components required to be present of B in A to permit dilation to occur. This process will be denoted by ®.

Then:

$A ® B = \cup(B)_r$ where $r \in R \mid r >= th$ [9]

2.2 Image Functions

In image functions each location is effectively an element of R. This imposes a different requirement on the use of the RJ operator. In this instance, the primarily useful data is more likely to be the number of missing or below threshold components rather than the actual missing data. The continuous form will be used to define conditional greyscale erosion and dilations, and to generate the direct extraction of features.

Consider the top surface form of greyscale erosion (see Haralick et al [6]):

$f \Theta k(x) = T[U[f] \Theta U[k]]$

This may be physically interpreted as $(f \Theta k)(x) = \min_{z \in k}\{f(x+z)-k(z)\}$

Now, for any point $(x_i) \in f(x) \mid (x_i+z) \in D_f$,

if $([U[f(x_i+z)] > [U[k(z)+f(x_i)]]) \forall z \in k[z]$

then $(f \Theta k)(x_i) = f(x_i)$

This is intuitively obvious. In order to relax the criterion for a point to belong to the eroded result, we must allow the point value to belong to the solution if components fail to meet the criterion for formal erosion, i.e. if components are missing from the eroding function k placed at the surface of the function f.

In effect our measure becomes:

$(f @ k)(x) = R(x),J$ [10]

where $R(x) = \{r \mid r =$ no of partials in maximal intersection of

$(f(x+z) >= (k(z)+f(x))) \forall z \in k(z), \forall x \in f(x) \mid (x+z) \in D_f\}$ [11]

and $J(x) = \{J_r\} \forall r \in R(x)$ [12]

$= \{j{:}N \mid j = k(z)$ where $f(x+z) = 0,$
$j = k(z)-f(r+z)$ where $f(r+z)<k(z),$
$\forall z \in k(z)\} \forall r \in R$

A J(x) set may be generated for each locus in f(x) (and hence R(x)), and will contain a number of elements up to the number of elements in k(z).

The R(x) set may be calculated in summation form as:

$$R(x) = \{r(x){:}N, x \in f(x), (x+z) \in D_f \mid r = \sum_{z \in k(z)} (f(x+z) >= (k(z)+f(x))\} \qquad [13]$$

The 2D operators can be stated as:

$$(f@k)(x,y) = R, J$$

where

$$R(x,y) = \{r(x,y){:}N, x,y \in f(x,y), (x+\xi, y+\eta) \in D_f \mid r = \sum_{(\xi,\eta) \in k(\xi,\eta)} [f(x+\xi,y+\eta) >= (f(x,y)+k(\xi,\eta))]\} \qquad [14]$$

and

$$\begin{aligned} J &= \{J_r\} \forall r \in R \\ &= \{j \in E^n \mid j = k(\xi,\eta) \text{ where } f(r_x+\xi, r_y+\eta) = 0, \\ &\qquad j = k(\xi,\eta) - f(r_x+\xi, r_y+\eta) \text{ where } f(r_x+\xi, r_y+\eta) < k(\xi,\eta), \\ &\qquad \forall \xi,\eta \in k(\xi,\eta)\} \; \forall r_x, r_y \in R \end{aligned} \qquad [15]$$

The conditional erosion may be denoted by:

$$\begin{aligned} (f \copyright k)(x,y){:}T(x,y) = \{g \mid\ & g = \min f(x+\xi,y+\eta)-k(\xi,\eta) \ \forall \xi,\eta \in k(\xi,\eta) \text{ where } t=0; \\ & g = f(x,y) \text{ where } t=1; \\ & \forall x,y \in f(x,y)\} \end{aligned} \qquad [16]$$

and the conditional dilation by:

$$\begin{aligned} (f \circledR k)(x,y){:}T(x,y) = \{g \mid\ & g = \max f(x-\xi,y-\eta)+k(\xi,\eta) \ \forall \xi,\eta \in k(\xi,\eta) \text{ where } t=1; \\ & g = f(x,y) \text{ where } t=0; \\ & \forall x,y \in f(x,y)\} \end{aligned} \qquad [17]$$

where, for some predefined threshold th:

$$T(x,y) = \{t \mid t=1 \text{ where } R(x,y) >= th;\ t=0 \text{ where } R(x,y) < th,\ \forall x,y \in R(x,y)\} \qquad [18]$$

It is consistent with the loosening of the conditions of erosion implied within the conditional erosion that the conditional dilation will effectively increase the conditions needing to be met to permit dilation to occur. When the threshold is set to zero, both conditional erosion and dilation forms collapse back to the formal set erosion and dilation definitions.

Various greyscale feature identification techniques can be generated. A full hit or miss transform is not particularly helpful in a situation where only the locus and type of feature information is required.

The feature identifier may therefore be stated as

$$\text{Locus} = T_1(x,y) \cap T_2(x,y) \qquad [19]$$

where:

$T_1(x,y) = \{t|t=1 \text{ where } R_1(x,y)>=th_1;\ t=0 \text{ where } R_1(x,y)<th_1,\ \forall x,y \in R_1(x,y)\}$

$T_2(x,y) = \{t|t=1 \text{ where } R_2(x,y)>=th_2;\ t=0 \text{ where } R_2(x,y)<th_2,\ \forall x,y \in R_2(x,y)\}$

$R_1(x,y) = f(x,y)@g(x,y)$

$R_2(x,y) = f^c(x,y)@h(x,y)$

This mostly hit, mostly miss greyscale transform has obvious applications in direct identification of features within a greyscale image. The result is a binarized found / not found feature set.

By suitable choice of templates and condition we may therefore segment useful data within the image or extract particular specific features (such as might be used for a recognition algorithm).

3. CHOICE OF TEMPLATES

Templates for feature extraction can be classified as symmetrical or directional in nature. If suitable, non-directional symmetrical templates are preferred to minimize the number of repeated iterations of the extraction algorithm.

The R operator produces a measure of the correlation of spatial frequencies between the probe set and an area the same size on the surface of the umbra of the function under test. If the probe set is wholly contained a complete match is given. Note that this in no way indicates that the function corresponds to the applied probe, only that it contains the applied probe. By combining (intersecting) according to some predefined threshold two R operations as shown above a conditional hit and miss transform is generated. It is this we need to probe for features.

Where a condition is used to determine the number of components of the probe set needed within the image under test to generate a match several conclusions may be drawn. The entire set of sub-patterns generable from the template set less a given number of pixels are effectively tested, and, by implication, all the spatial frequencies contained within the various patterns. Note that different sized templates will yield differing results where the spatial intensity variation occurs over an area smaller than the defined template. By dilation, the template may therefore be used to differentiate between certain texture variations and the presence of a geometric (or other) feature. This technique will also prevent aliasing by dual structures for odd count probe sets.

The basic shape of the probe set is still of fundamental importance. The donut set (fig(1)) and the tower set (fig(2)) isolate different classes of features, and have yielded good results in experimental tests. Combination of template shape and correlation number offer a means of classifying textures.

Figure (1): Donut Template

		5	5	5		
	5				5	
5						5
5			+			5
5						5
	5				5	
		5	5	5		

Figure (2): Tower Template

5	5	5	5	5	5	5
5	5	5	5	5	5	5
5	5				5	5
5	5		+		5	5
5	5				5	5
5	5	5	5	5	5	5
5	5	5	5	5	5	5

Figure 3(a) was extracted from a volume measurement system for ulcers [7]. Figure 3(b) was generated using the donut template to extract contours within the ulcerated area to enable the operator to improve accurate determination of the perimeter of the wound.

Examples of feature extraction from low textured objects on a low textured background are shown in figures 4(a)-4(c). Figure 4(a) shows the original image, figure 4(b) illustrates the extraction of convex features and figure 4(c) concave features using the donut template with appropriate thresholds.

Figure 3(a): Surgical Wound

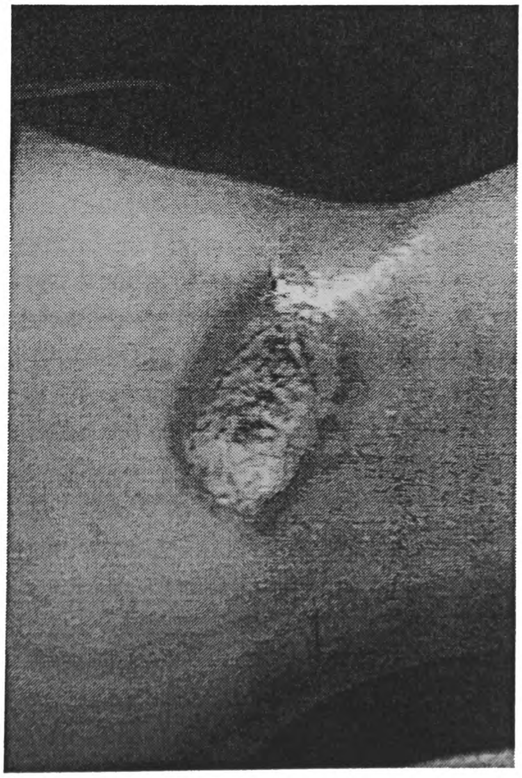


Figure 3(b): Contour Enhanced Wound

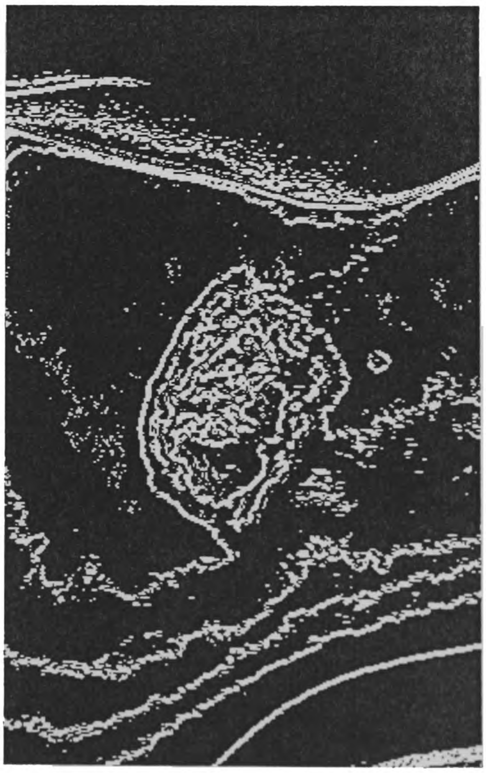


figure 4(a): "Harrier"

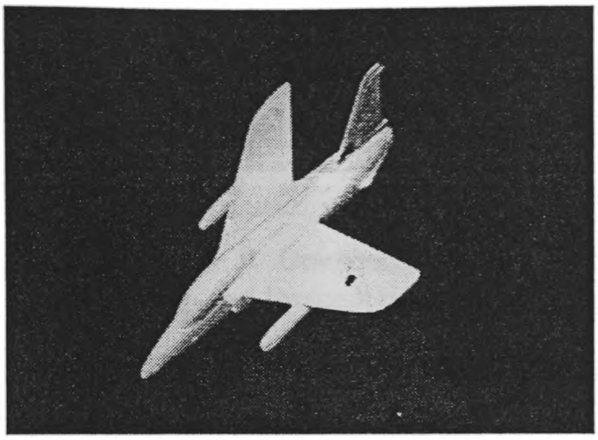


Figure 4(b): Convex high curvature points extracted with "donut"

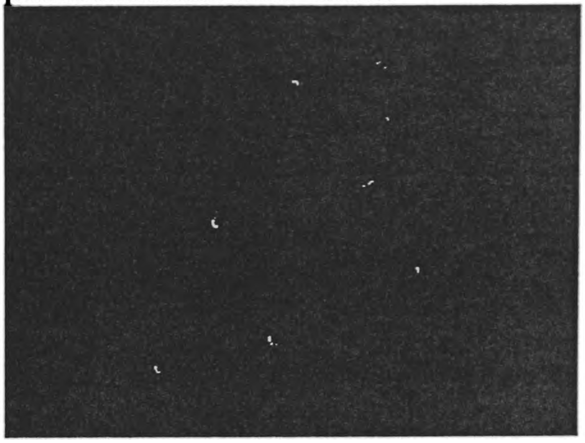
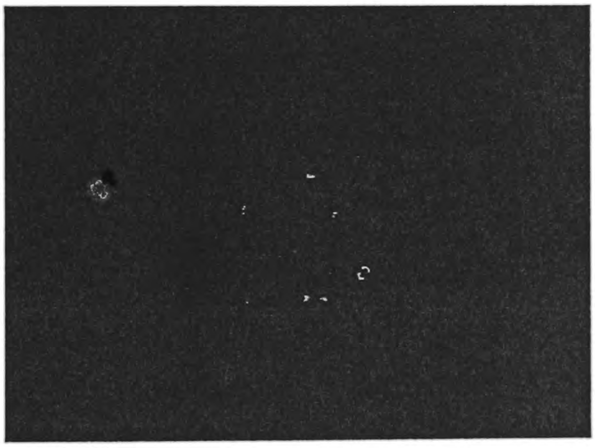


Figure 4(c): Concave high curvature extracted with "donut"



4. DISCUSSION

Successful feature extraction is dependent on a correct combination of probe set and threshold selection. Whilst options are obvious for geometric structures, for larger and curved features they are less so. An adequate clustering of results is achieved in these cases by broadening the acceptable limits of pixels for presence of a feature, but this does increase the probability of aliasing. To minimize aliasing caused by random noise, texture or the presence of a close second object, three templates of the same basic shape but dilated were used and the results logically ANDed.

The surgical wound of figure 3 shows the effectiveness of this technique in enhancing textural differences to aid segmentation.

Where the gradient of the template set in the intensity domain is close to the gradient difference between the object and its background, the method yields finer lines for edge detection and smaller clusters for features (as one might expect). In certain circumstances it may therefore be useful to adaptively set the gradient of the template to mirror that of the image. Do note, however, that this may increase the susceptibility to noise in the image.

The same algorithm can be used identify the presence of one feature set within another, and may therefore be applied for both recognition and extraction purposes. This offers the possibility of implementing a unified algorithmic architecture in hardware for high speed recognition applications, working directly from unprocessed input image data.

5. REFERENCES

1. D Sinha, E R Dougherty, "Fuzzy Mathematical Morphology", Journal of Visual Communication and Image Representation, Vol 3, No 3, September 1992.

2. E R Dougherty, D Sinha, P Sinha, "Fuzzy Morphological Filters" in Intelligent Robots and Computer Vision XI, D P Casasent (Ed.), SPIE Vol. 1825, Boston, 1992.

3. P Maragos, "Pattern Spectrum and Multiscale Shape Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989.

4. T R Esselman, J G Verly, "Feature Extraction from Range Imagery using Mathematical Morphology", Proceedings of the SPIE, Vol 845, Part 27, Oct 1987, pp 233 - 240.

5. I Pitas, A Venetsanopoulos, "Morphological Shape Decomposition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, Jan 1990.

6. R M Haralick, S R Sternberg, X Zhuang, "Image Analysis Using Mathematical Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, July, 1987.

7. P Plassman, Measuring Area and Volume of Leg Ulcers by Structured Light, PhD Thesis, Dept of Electronics and IT, University of Glamorgan, 1992.

Appendix C.3: S J Rees, B F Jones, "Conditional Morphological Operators for Direct Feature Extraction and Enhancement", Procceedings of IEE 5th International Conference on Image Processing and its Applications, Heriot-Watt University, Edinburgh, 4-6th July, 1995, IEE Publication No 410 pp 747 - 751.

CONDITIONAL MORPHOLOGICAL OPERATORS FOR DIRECT FEATURE EXTRACTION AND ENHANCEMENT

S J Rees, B F Jones

The University of Glamorgan, UK.

ABSTRACT

Conditional morphological operators have been used for feature extraction and enhancement in greyscale images. Templates are used to determine the presence of specific spatial structures in the intensity map. Applied conditions permit noise defects preventing full containment of the template to occur, and allow recognition of "broad brush" classes of features as well as specific single structures. The theoretical methods used are explained and practical templates indicated. Practical examples include the delineation of soft tissue wounds, enhancement of textural differences and feature extraction.

INTRODUCTION

Morphological operators have found wide acceptance for image enhancement and analysis. Following from initial work on set erosion for image analysis an operator has been defined allowing a conditional erosion and dilation to be generated for binary and grey scale pictures. This paper will describe the application of the greyscale operators for direct feature extraction and segmentation. Results will be shown for a variety of objects including aircraft models and surgical wounds, and will include descriptions of the templates used for extraction.

The development of conditional and fuzzy mathematical morphology techniques has progressed over the last few years [1,2]. Application of morphology to skeletonization and recognition has been undertaken with some considerable success [3,4,5]. This paper describes the use of conditional morphological operators for direct extraction of features in grey-scale images for the purpose of abstracting feature sets for object recognition purposes. This is part of a larger project aimed at using conditional morphology for both the front end feature extraction and the object recognition stages, yielding a consistent algorithmic approach of a suitable form for implementation as a scalable hardware architecture.

THEORY: THE R-J OPERATOR

Let us first define the maximal intersection of several sets or subsets as the set of points which occur in the largest numbers of the sets. Obviously more than one maximal intersection may occur. Points of maximal intersection do not have to belong to the same subset of the sets.

Consider the top surface form of greyscale erosion (see Haralick et al [6]):

$$f\Theta k(x) = T[U[f]\Theta U[k]]$$

This may be physically interpreted as:

$$f\Theta k(x) = \min_{z\epsilon k}\{f(x+z) - k(z)\}$$

Now, for any point $(x_i) \in f(x)|(x_i+z) \in D_f$, if

$$([U[f(x_i+z)] \geq [U[f(x_i)+k(z)]])\forall z \in k(z)$$

then

$$(f\Theta k)(x_i) = f(x_i)$$

In order to relax the criterion for a point in the original image to belong to the eroded result, we must allow the point value to belong to the solution if components fail to meet the criterion for formal erosion, i.e. if components are missing from the eroding function k placed at the surface of the function f.

Now define an operator @, which generates a set containing a count of how many components of the image function f would meet the formal requirement for erosion by k, and a set of sets J_r where each member set contains the necessary additions at the point to permit the erosion to occur:

$$(f@k)(x) = R(x), J \qquad [1]$$

where

$$\begin{aligned} R(x) = \{r|r = & \text{ no of partials in maximal intersection} \\ & \text{of } (f(x+z) \geq (k(z)+f(x))) \\ & \forall z \in k(z), \forall x \in f(x)|(x+z) \in D_f\} \end{aligned} \qquad [2]$$

and $J(x) = \{J_r\}\forall r \in R(x)$

$$\begin{aligned} = \{j{:}N|\, & j = k(z) \text{ where } f(x+z) = 0, \\ & j = k(z) - f(r+z) \text{ where } f(r+z) < k(z), \\ & \forall z \in k(z)\}\forall r \in R(x) \end{aligned} \qquad [3]$$

A J(x) set may be generated for each locus in f(x) (and hence R(x)), and will contain a number of elements up to the number of elements in k(z). It is primarily of use in recognition model evaluation from set occupancy.

The R(x) set may be calculated in summation form as:

$$\begin{aligned} R(x) = \{r(x)\colon N, x \in f(x), (x+z \in D_f) | \\ r = \sum_{z \in k(z)} (f(x+z) \geq (k(z)+f(x))\} \end{aligned} \qquad [4]$$

We can now define a conditional erosion denoted by ©

$$\begin{aligned} &(f © k)(x,y)\colon T(x,y) = \\ &\{g | g = \min f(x+\varepsilon, y+\eta) - k(\varepsilon,\eta) \forall \varepsilon,\eta \in k(\varepsilon,\eta), \\ &\quad \text{where } t=0, \\ &\quad g = f(x,y) \text{ where } t=1,\ \forall x,y \in f(x,y)\} \end{aligned} \qquad [5]$$

and the conditional dilation denoted by ®:

$$\begin{aligned} &(f ® k)(x,y)\colon T(x,y) = \\ &\{g | g = \max f(x-\varepsilon, y-\eta) + k(\varepsilon,\eta) \forall \varepsilon,\eta \in k(\varepsilon,\eta) \\ &\quad \text{where } t=0, \\ &\quad g = f(x,y) \text{ where } t=1, \forall x,y \in f(x,y)\} \end{aligned} \qquad [6]$$

where, for some predefined set of conditions cond1:

$$\begin{aligned} T(x,y) = \{t | t=1 \text{ where } R(x,y) \in cond1, \\ t=0 \text{ elsewhere}, \ \forall x,y \in R(x,y)\} \end{aligned} \qquad [7]$$

When the threshold is set to zero, both conditional erosion and dilation forms collapse back to the formal set erosion and dilation definitions. This erosion will tend to bring up features of interest, defined by the profile of the probe function or template. The dilation will tend to enhance the background.

A feature identifier may be stated as:

$$\text{Locus} = T_1(x,y) \cap T_2(x,y) \qquad [8]$$

where:

$$\begin{aligned} T_1(x,y) = \{t | t=1 \text{ where } R_1(x,y) \in cond1, \\ t=0 \text{ elsewhere}\} \end{aligned}$$

$$\begin{aligned} T_2(x,y) = \{t | t=1 \text{ where } R_2(x,y) \in cond2, \\ t=0 \text{ elsewhere}\} \end{aligned}$$

$$R_1(x,y) = f(x,y) @ g(\varepsilon,\eta)$$

$$R_2(x,y) = f(x,y) @ h(\varepsilon,\eta)$$

This mostly hit, mostly miss greyscale transform has obvious applications in direct identification of features within a greyscale image. The result is a binarized found / not found feature set. By suitable choice of templates and condition we may therefore segment useful data within the image or extract particular specific features (such as might be used for a recognition algorithm).

Chosen structural feature types within an image may be enhanced in the presence of noise and texture using a modification of this approach. By using increasing sizes of a template, features below a predetermined size may be eliminated. Combinations of templates with specific spatial components enable the removal of noise and texture effects within the bounds set by the template.

$$\text{Locus} = \bigcap_n T_n(x,y) \qquad [9]$$

where:

$$\begin{aligned} T_n(x,y) = \{t | t=1 \text{ where } R_n(x,y) \in cond\, n, \\ t=0 \text{ elsewhere}, \ \forall x,y \in R_n(x,y)\} \end{aligned}$$

$$R_n = f(x,y) @ k_n(\varepsilon,\eta)$$

CHOICE OF TEMPLATES

The R operator produces a measure of the correlation of spatial frequencies between the probe set and an area the same size on the surface of the umbra of the function under test. By combining (intersecting) the results of two R operations a conditional hit and miss transform is generated. It is this we need to probe for features.

Where a condition is used, each of sub-patterns generatable from the template set due to the condition is effectively tested. By implication, all the spatial frequencies contained within the various sub-patterns are tested. Note that different sized templates will yield differing results where the spatial intensity variation occurs over an area smaller than the defined template. By dilation, the template may therefore be used to differentiate between certain texture variations and the presence of a geometric (or other) feature.

Figure (1): Donut Template

```
    5  5  5
 5           5
5             5
5      +      5
5             5
 5           5
    5  5  5
```

RESULTS

The wound image of figure 2(a) was extracted from a volume measurement system for ulcers [7].

Figure 2(a): Soft Tissue Wound

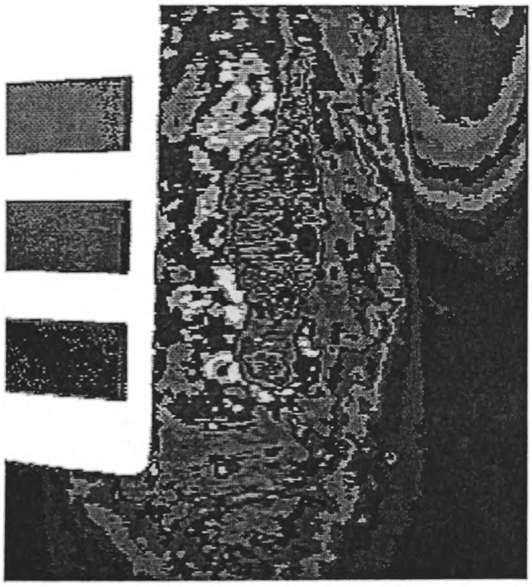
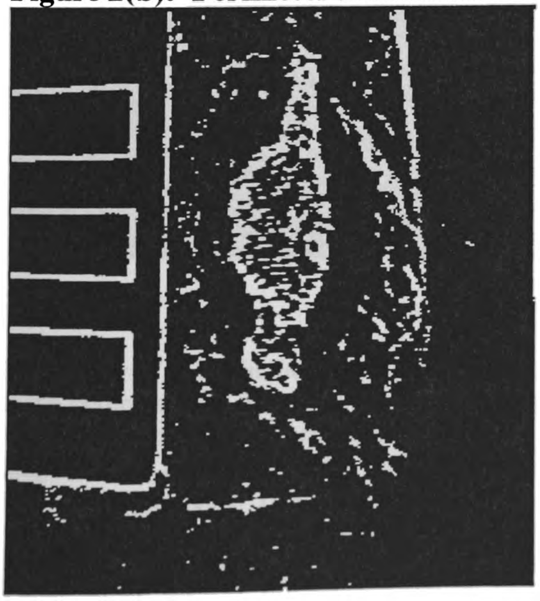


Figure 2(b): Perimeter Delineation



One problem with the measurement system was the need for manual delineation of the perimeter of the wound, usually by an operator drawing around the area of interest with a mouse. This led to problems with precision and repeatability, particularly where several operators used the system. Figure 2(b) was generated using the donut template of figure (1) to enhance the grey level contours around the ulcerated area, thus enabling improved determination of the perimeter of the wound. Whilst not yielding a perfect containment of the area, the operator is only required to join the relevant perimeter segments which have been automatically generated. The system is both repeatable and accurate in its delineation.

The T1 Gnat trainer of figure 3(a) illustrates feature extraction from low textured objects. Figure 3(a) shows the original image, figure 3(b) illustrates the extraction of convex features and figure 3(c) concave features using the donut template with appropriate thresholds and equation [8]. As can be seen from the results, the localization of the features is adequate. This data is suitable for use in recognition algorithms. An adjunct to this work has devised a morphological algorithm for recognition based on erosion, allowing a single architecture for enhancement, extraction and recognition.

Figure 3(a): T1 Gnat Trainer

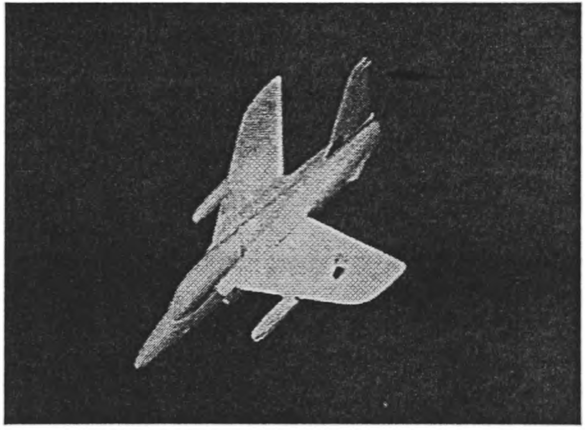


Figure 3(b): T1 Gnat Convex Features

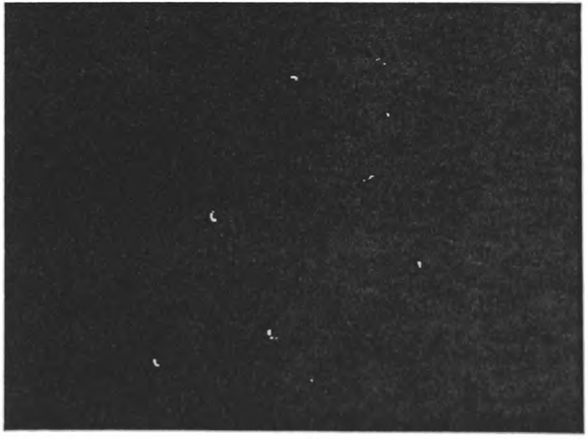


Figure 3(c): T1 Gnat Concave Features

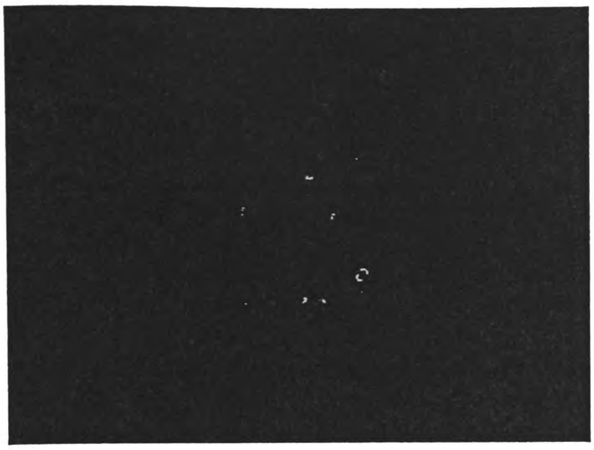
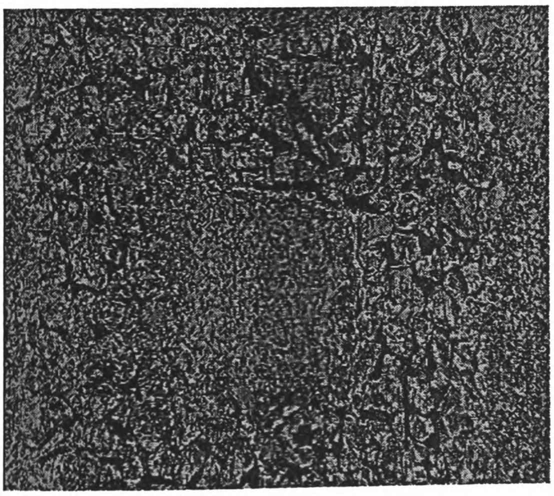


Figure 4(a) is a textural enhancement example, using gravel on chipboard. The gravel has two textural components, the coarse structure of the stone pattern and the fine structure of the texture of the gravel surface. This fine structure is not dissimilar to the texture of the chipboard. Four increasing expansions of the donut template with a gradient of 20 were used to increase the differentiation between the two regions by application of equation [9]. The results shown were found by choosing limits appropriate for concave two dimensional features and straight edges.

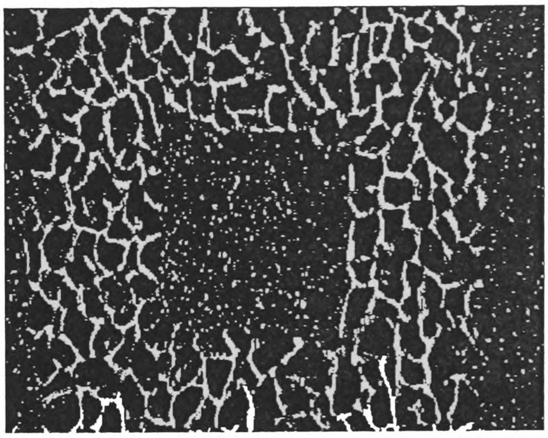
Figure 4(a): Textured Object and Background



DISCUSSION

The number of points in the probe set is directly related to the computational efficiency of the algorithms. A subtraction and compare occurs for each pixel in the image for each point in the probe set. Sparse probe sets are therefore desirable. However, with large area probe sets and low containment values, the risk of random effects becoming significant increases. Such spurious effects might include the boundaries of other real features, causing noise within the result. Some evidence of this can be seen in figure4(b).

Figure 4(b): Enhanced Texture Separation



Structural features may be selectively eliminated from the result in three ways: the selection of appropriate form probe sets; increasing the (x,y) size of the probe set; and increasing the intensity gradient of the probe set.

Where textured surfaces are not a problem, increasing the gradient is effective in eliminating spurious features and increasing the localization of the resulting extracted feature set. Where texture exists and is "deep" then it may form more significant boundaries than the desired object. In this case increasing the gradient may only be of significance if the object disappears before the textured background.

Where the textural gradient approaches that of the background/object perimeter further information must be used to adequately segment perimeter features. Here application of multiple sizes of templates finds significant usage in conditionally rejecting features and objects smaller than the applied spatial template. The shape of the texture (if relevant) can be accounted for by judicious selection of probe conditions and shape. Where the texture of the object and background are of similar size, shape and depth, intensity domain solutions fail.

REFERENCES

1. D Sinha, E R Dougherty, "Fuzzy Mathematical Morphology", Journal of Visual Communication and Image Representation, Vol 3, No 3, September 1992.

2. E R Dougherty, D Sinha, P Sinha, "Fuzzy Morphological Filters" in Intelligent Robots and Computer Vision XI, D P Casasent (Ed.), SPIE Vol. 1825, Boston, 1992.

3. P Maragos, "Pattern Spectrum and Multiscale Shape Representation", IEEE Transactions on Pattern

Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989.

4. T R Esselman, J G Verly, "Feature Extraction from Range Imagery using Mathematical Morphology", Proceedings of the SPIE, Vol 845, Part 27, Oct 1987, pp 233 - 240.

5. I Pitas, A Venetsanopoulos, "Morphological Shape Decomposition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, Jan 1990.

6. R M Haralick, S R Sternberg, X Zhuang, "Image Analysis Using Mathematical Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, July, 1987.

7. P Plassman, Measuring Area and Volume of Leg Ulcers by Structured Light, PhD Thesis, Dept of Electronics and IT, University of Glamorgan, 1992.

Appendix C.4: S J Rees, S Woodham, L S Dooley, "Use of Rank-Conditioned Morphological Operators for Texture Classification", to be published in IEE 6th International Conference on Image Processing and its Applications, Trinity College, Dublin, Ireland, 14-17th July, 1997.

Use of Rank-Conditioned Morphological Operators for Texture Classification

S J Rees, S Woodham, L S Dooley

University of Glamorgan, UK

Introduction

The use of morphological decomposition and classification for texture has received considerable attention [1,2]. The basis of mathematical morphology is the study of the geometrical structure in an image through the use of simple probe forms called structuring elements. Methods have been applied to obtaining structural decomposition using a variety of sequenced applications of morphological filters. These use a sequential elimination of increasing sizes of information from the original image through application of morphological openings. This results in a multidimensional classification of the original texture based on the residual images, in [2] using grey level means for each point in the component images as texture features.

A different formulation has been implemented, using a rank-conditioned morphological implementation [3] to generate the required classification features. The features are based on the rank required for each point in the image to remain part of the rank conditionally eroded result based on the structuring element used. The structuring elements are chosen as increasing sizes of the same basic shape, in the case presented here using a sparse ring or annulus as the shape outline. Intensity variations are considered as part of the feature set through the use of increasing intensity gradients of ring.

The resulting feature vectors are used as discriminants of the texture under analysis.

Rank-Conditioned Morphological Filters

In accordance with the usual morphological filter implementations, the probe structuring element is placed at the top surface each point in the image function *f(x)*. The number of components of the structuring element not contained over the element window are calculated. This, effectively, is the minimum rank condition that would be applied to permit the point to belong to the result unaltered if a rank conditioned erosion were undertaken. This is derived from a use of rank conditioning as a means of reducing the perfection of containment required by formal set erosion. The ranked values are used to generate a characteristic signature of the texture under analysis. Note that this is not a decomposition operation, but an iterative application of different sized structuring elements to generate a multidimensional feature vector.

Consider the top surface form of greyscale erosion (see Sternberg [4], or Haralick et al [5],):

$$f\Theta k(x) = T[U[f]\Theta U[k]]$$

This may be physically interpreted as:

$$f\Theta k(x) = \min_{z\in k}\{f(x+z) - k(z)\}$$

Now, for any point $(x_i) \in f(x) | (x_i + z) \in D_f$,

if $([U[f(x_i+z)] \geq [U[f(x_i)+k(z)]) \forall z \in k(z)$

then $(f\Theta k)(x_i) = f(x_i)$

In other words, any point in the original image belongs to the eroded result provided it meets this criterion. In order to relax the criterion, we must allow the point value $f(x_i)$ to belong to the solution if components fail to meet the criterion for formal erosion, i.e. if $([U[f(x_i+z)] \geq [U[f(x_i)+k(z)])$ for some proportion of $z\in k(z)$ rather than for its whole. If the test is to be general, rather than directional or for a specific fragment of the probe, then the metric for assessing coverage should simply be the count of how many components of the probe set are contained.

In effect our measure becomes:

$$(f @ k)(x) = R(x), J \text{ , where}$$
$$R(x) = \{r(x): N, x \in f(x), (x + z \in D_f)|$$
$$r = \sum_{z\in k(z)} m,\ m{=}1 \text{ where } (f(x+z) \geq (k(z) + f(x));$$
$$m = 0 \text{ otherwise}\}$$

A *J(x)* set may be generated for each locus in *f(x)* (and hence *R(x)*), and will contain a number of elements up to the number of elements in *k(z)*, indicating the structural information missing from *f(x)* to permit complete containment of *k(z)*. It is primarily of use in recognition model evaluation from set occupancy, and is not used in the texture evaluation discussed here.

$$J(x) = \{J_r\} \forall r \in R(x)$$

$$= \{ j{:}\, N \mid j = k(z) \text{ where } f(x+z) = 0,$$
$$j = k(z) - f(r+z) \text{ where } f(r+z) < k(z),$$
$$\forall z \in k(z)\} \forall r \in R(x)$$

The analysis is then formed as a histogram, H, where, over a sample window size MxN∈*f(x,y)*, the individual histogram values are:

$$H_i = \sum_{M,N} r(x,y) = i, i <= CARD(k(z))$$

The set of characteristic histograms, **H**, is the feature classifier.

$$\mathcal{H}_N = \{\{H\}_n\}, n = \text{no of probe structuring elements applied}$$

Note that although the sample window applied is square, the resulting shape of area analysed depends, additionally, on the shape of the probe structuring elements and does overlap the perimeter of the window area in its analysis.

Experimental Results

The structuring elements used were a set of increasing sizes of a sparse annular templates. These were applied in turn to the acquired texture images and the resulting feature histograms built, and applied to the classification of other samples of textural data. The textures used were drawn from the natural texture examples in the Brodatz [6] set of samples, and also from sample images of galvannealed steel with different degrees of annealing. The Brodatz set textures included D3, D4, D5, D9, D15, D54, D57, D62, D67, D92, D98, D103 Typical examples of natural textures are shown below in figures (1) and (2):

Figure 1: Texture Sample Beans

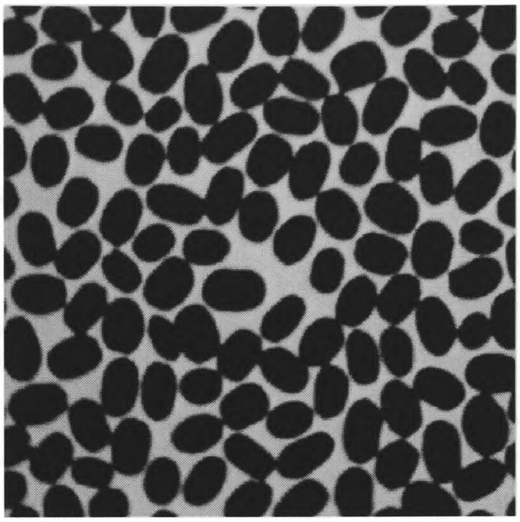
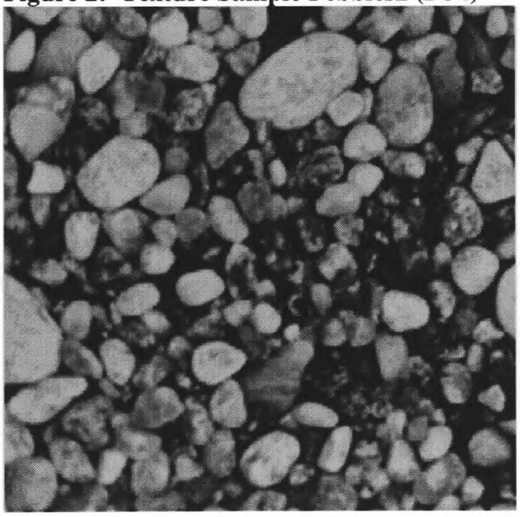


Figure 2: Texture Sample Pebbles2 (D54)



The effects of sample window size on the histograms obtained was evaluated. The resulting classifiers applied to nineteen texture samples drawn from the Brodatz set and the separations of the feature vectors calculated. The images were based on a 256x256, 8 bit deep sampling. The results shown in Table (1) below were obtained using a sparse annular template. Classification was successfully obtained for adequate samples of texture with five sizes and three intensity gradients of template, yielding a fifteen dimensional evaluation vector for each sample. In this example, the fifteen evaluation feature vectors were found to be redundant for the samples taken. It should be noted that, in this case, the samples were taken as sub-images of each texture. Complete isolation of the texture within the system world model is claimed where the in-sample scatter is less than the between sub-samples worst case scatter (the data was then experimentally verified).

Table (1): Effect of Window Size and Number of Vectors on Texture Discrimination

Window Size (M=N)	Number of Feature Vectors	Unique Discrimination of All Texture Samples %	Average Level of Discrimination %	Lowest Level of Discrimination of Textures %
48	15	100	100	100
96	15	100	100	100
48	3	26	80	50
96	3	100	100	100

The feature extraction method was then applied to a natural, random texture sample, using 380 samples of galvannealed steel in five known, calibrated coating conditions. These images were acquired at a x500 magnification, and were subject in some cases to depth of field problems causing partial blurring of the acquired image as is shown in figure 3. As can be seen from the results of table 2, there is a strong correlation between the coating condition and the optical texture as it was measured, but the classification scheme is by no means perfect in its present form.

Table 2: Classification of Galvanneal Samples

Window Size (MxN)	No of Vectors Used	% Assigned to Correct Class (of 380 Samples, 5 Classes)
400x400	15	75
200x200	15	65
400x400	3	60

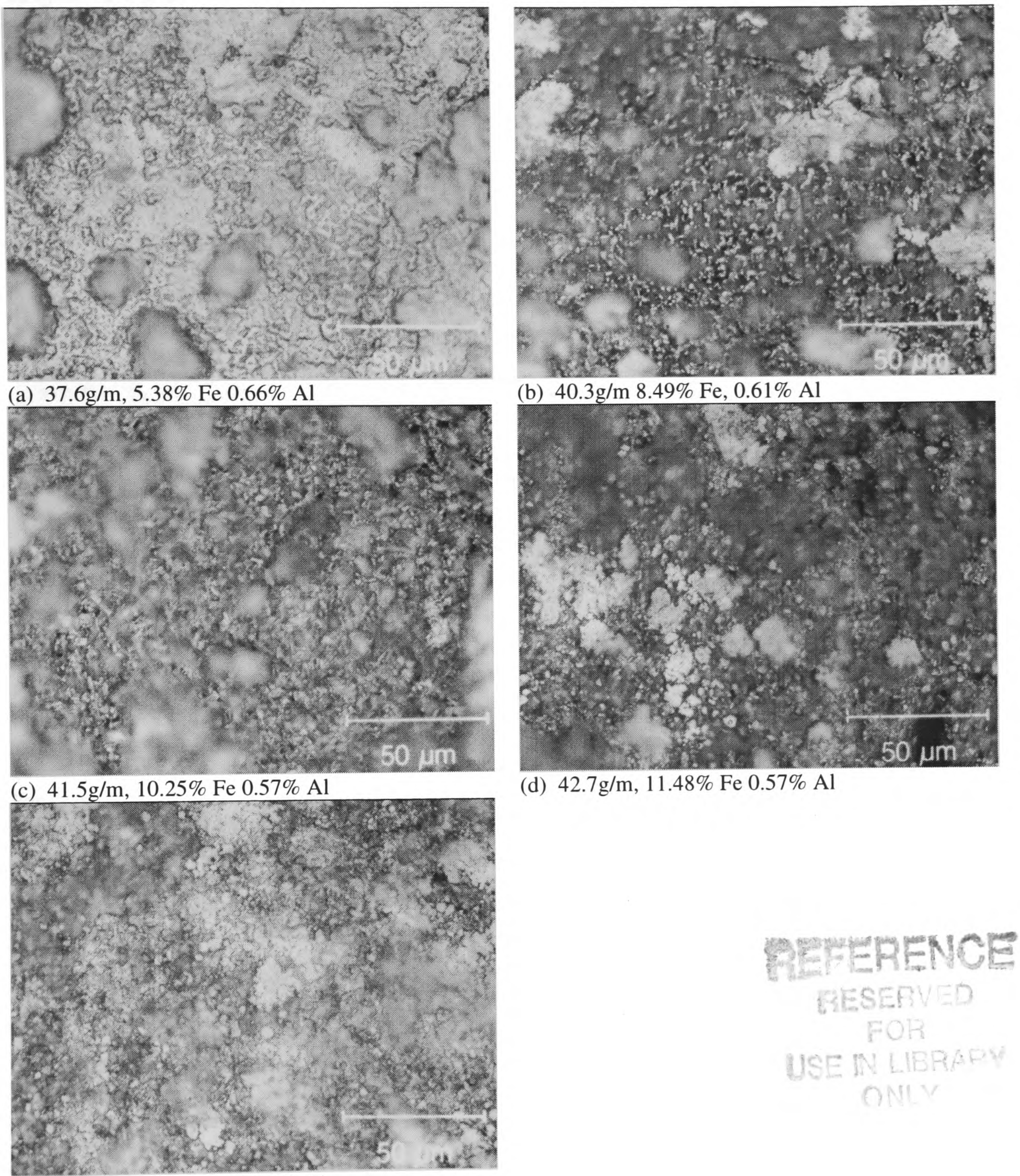
Noise Performance

Salt and pepper noise was added to the Brodatz set texture samples, and the algorithms run to classify them. In the context of the rank of the containment of the probe sets, this impulsive noise produces limited distortion of the histogram, but causes the raising of the outliers in the data (corresponding to no containment and full containment at a point). The algorithms were adjusted to ignore these outliers, producing a significant improvement in the performance of the classification scheme.

Table 3: Noise Performance of Classification Based on the Use of 15 Feature Vectors

% Noise	Window Size	% Correctly Classified (of 19 Texture Samples)	% Correctly Classified on Removal of Data Outliers
0	150x150	100	100
1	150x150	90	95
2	150x150	79	85
3	150x150	63	70
4	150x150	52	60

Figure 3: Galvanneal Samples. 512x512 Images at 8 Bit Resolution



(a) 37.6g/m, 5.38% Fe 0.66% Al

(b) 40.3g/m 8.49% Fe, 0.61% Al

(c) 41.5g/m, 10.25% Fe 0.57% Al

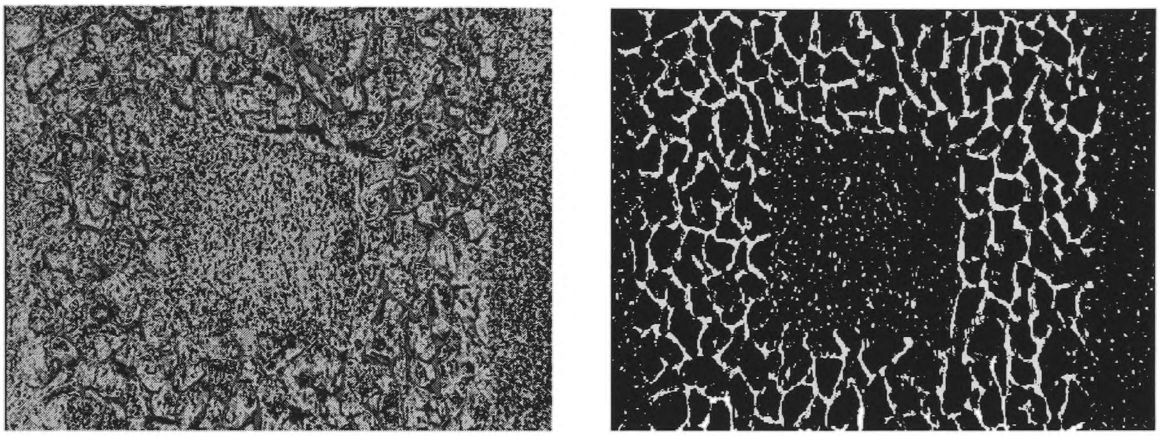
(d) 42.7g/m, 11.48% Fe 0.57% Al

(e) 51.2g/m, 13.64 % Fe 0.5% Al

Boundary Detection

In order to locate boundaries between textures, it is necessary to assign pixels in close proximity to the boundary to particular texture regions. The larger the probe structuring element is about the point, the more likely it is to overlap the "other" texture region. The approach used was to adopt a 96x96 pixel window size, and classify the regions initially on this basis. Where the classification metric exceeds the permitted variation, typically near a texture boundary, then the larger sizes of template were eliminated and the window size halved, down to a limit of 12 pixels window size. All remaining point are assigned on a nearest neighbour basis. With test data of the form shown in figure 4 below, the pixel classification was found to be accurate down to the minimum window size, and the system classified between 30% and 70% of the remaining pixels correctly, dependant upon the chosen boundary shape.

Figure (4): Textured Object and Background Enhancement



(i) Gravel on Chipboard

(ii) Textures Enhanced Separation

Conclusion

The texture analysis method developed has been successfully applied to a variety of natural textures. It is based on extraction of integer data, and the histogram feature data extraction is capable of implementation in hardware, based on a shift and compare algorithm on a pixel by pixel basis, comparison being undertaken over the chosen template shape size. Subject to the replication of the shape size area, this is highly amenable to parallel implementation down to the single pixel level.

Comparison with the work of Wang and Haese-Coat[2] in terms of noise performance and discrimination indicate a less effective result. This is due in part to the non-optimal classification method adopted, and is compensated by lower computational intensity.

References:

1. Toet A., 1989, "A morphological pyramidal image decomposition", Pattern Recognition Letters, No 9, pp255-261.

2. Wang D, Haese-Coat V, Bruno A, and Ronsin J., 1993, "Texture Classification and Segmentation Based on Iterative Morphological Decomposition", Journal of Visual Communication and Image Representation, No 4, pp197-214.

3. Rees S J, Jones B F, "Conditional Morphological Operators for Direct Feature Extraction and Enhancement", 5th International Conference on Image Processing and its Applications, IEE Conference Publication No. 410, Edinburgh, July 1995, pp747-751.

4. Sternberg S R, "Greyscale Morphology", Computer Vision, Graphics and Image Processing, Vol. 35, pp333-355, 1986.

5. Haralick R, Sternberg S R, Zhuang X, "Image Analysis using Mathematical Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, July 1987.

6. Brodatz P, Textures, Dover, New York.

7. Wei Li, Haese-Coat V, "Composite Morphological Filters in Multiresolution Morphological Decomposition", 5th International Conference on Image Processing and its Applications, IEE Conference Publication No. 410, Edinburgh, July 1995, pp752-756.