Texture Description and Segmentation in Image Processing

Michael Spann B.Sc M.Sc

A thesis submitted to
The University of Aston in Birmingham
for the degree of
Doctor of Philosophy

September 1985

Texture Description and Segmentation in Image Processing

Michael Spann

A thesis submitted to
The University of Aston in Birmingham
for the degree of
Doctor of Philosophy
1985

Summary

Textured regions in images can be defined as those regions containing a signal which has some measure of randomness. This thesis is concerned with the description of homogeneous texture in terms of a signal model and to develop a means of spatially separating regions of differing texture.

A signal model is presented which is based on the assumption that a large class of textures can adequately be represented by their Fourier amplitude spectra only, with the phase spectra modelled by a random process. It is shown that, under mild restrictions, the above model leads to a stationary random process. Results indicate that this assumption is valid for those textures lacking significant local structure.

A texture segmentation scheme is described which separates textured regions based on the assumption that each texture has a different distribution of signal energy within its amplitude spectrum. A set of bandpass quadrature filters are applied to the original signal and the envelope of the output of each filter taken. The filters are designed to have maximum mutual energy concentration in both the spatial and spatial frequency domains thus providing high spatial and class resolutions.

The outputs of these filters are processed using a multi-resolution classifier which applies a clustering algorithm on the data at a low spatial resolution and then performs a boundary estimation operation in which processing is carried out over a range of spatial resolutions.

Results demonstrate a high performance, in terms of the classification error, for a range of synthetic and natural textures.

Key Words

Segmentation, Texture, Region Description
Boundary Estimation

This work is dedicated to my wife Ann.

Acknowledgements

There are a number of people who have helped during the course of this work and in the production of this thesis.

I would like to thank the staff of the microprocessor unit for the use of their word processing facilities. Further acknowledgement is due to Hazell Penzig for typing the mathematical equations, to Andy Crowcombe of the CMU for the production of the photographs and to Andy Tyrrell for making the tea. I would also like to thank the SERC for their funding over the past 3 years. Finally I am indebted to my supervisor Dr Roland Wilson without whose constant encouragement and deep understanding of the subject, this work would not have been possible.

CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Introductory Remarks

Texture is a ubiquitous property of most naturally occurring images. One only has to look at a picture of a landscape to note the leaves of a tree, the surface of a lake or the clouds in the sky all of which, in some sense, constitute texture.

Hawkins [1] characterised a textural field as being a repetition of some elemental unit over the entire image plane, this unit being small compared to the region size. Such a definition appears to cover a large class of textures, for example, a brick wall or a wallpaper pattern. In order to expand the definition, some measure of incoherence or randomness must be introduced. In this case any sub-region of the textural field can be perceived to have the properties characterising the texture irrespective of the position of the sub-region, given that its size is large compared to that of the texture's elemental unit. Such an assumption of position invariance is crucial to the work described in this thesis.

Having an understanding of texture is important in many image processing applications. Image analysis

and understanding systems require knowledge of regions of constant texture to make semantic judgements about the image. This is the case, for example, in systems that analyse satellite imagery in determining the location of particular crops. Certain systems that develop 3-D image descriptions from 2-D data employ texture cues to infer surface orientation [2]. Applications in image coding [3] are also envisaged. Specifically, most conventional coding schemes are unable to deal adequately with texture on a pixel by pixel basis due to the fact that many textures are only correlated over short distances and hence there is a lack of redundancy in such signals. This problem could be solved by coding textured regions in a parametric fashion which reproduces the statistical characteristics of the field but does not attempt to reproduce a pixel by pixel representation.

Also, an important area of study concerns the perception of texture and in particular the performance of the human visual system in discriminating a pair of texture fields.If a necessary and sufficient set of conditions could be established whereby discrimination could be predicted this would be an invaluable aid in the understanding of human visual perception. More will be said about this below.

1.2 Texture Perception and Discrimination

An important insight into the workings of the human visual system is provided by experiments on texture perception and discrimination. How human beings perceive texture and in particular the factors that influence the discriminability of two texture fields is crucial to this question.

Julesz carried out detailed experiments on "effortless" texture discrimination using a pair of texture fields each consisting of a set of sub-patterns having just two grey levels-black and white. He defined "effortless discrimination" as a "perceptual task that can be performed without scrutiny or deliberation on an individual sub-pattern of the texture."

In an early paper [4] Julesz described a set of experiments on random dot arrays placed side by side. He established that textures having the same second order statistics (and hence first order statistics) but differing in third and higher order statistics are indiscriminable. In particular, differences in first order statistics only are perceived as differences in the mean grey level of each texture. Differences in second order statistics only are perceived as differences in granularity of the textures.

Such a conjecture is generally true for all texture pairs comprising random dot arrays due to the absence of any structure at a local level. However, in a subsequent paper [5], Julesz acknowledged that local processes are also important in texture perception. He coined the term texton to describe features of individual sub-patterns that render texture pairs with identical second order statistics discriminable. Indeed, he suggested that it is the first order density of these textons that enable discrimination even in cases of identical second order statistics. He also suggested that the positional information or phase of the textons has no perceptual significance. The phase becomes important when focal attention is paid to local texture elements.

A related approach to that of Julesz is that of Beck [6] who investigated the relationship between the perceptual differences of the local texture elements and the effortless discrimination of the texture pairs. He postulated that texture discrimination is the result of differences in the first order density of stimulus features of the local texture elements, particular features giving stronger discrimination than others. For example, differences in local element orientation and slope stimulate stronger discrimination than differences in curvature of a local line segment. He used perceptual experimentation to assess

different types of local element features to grade them in terms of their contribution to discriminability. As in the work of Julesz, he restricted himself to textures consisting of arrays of sub-patterns, an approach which is clearly related to the structural models described in the next section.

Gagalowicz [7] suggested the use of second order spatial averages as sufficient for visual discrimination between textures and in particular those based on the co-occurrence matrix defined in the next section. He commented that many of the examples proposed by Julesz are not ergodic, that is to say the second order statistics as computed from the generating process are not the same as the second order spatial averages. Hence discrimination is not surprising in this case. However, he concluded that textures with identical second order spatial averages could only be discriminated on the basis of local inhomogeneity, a result in accordance with the texton theory of Julesz.

Without these local inhomogeneities which appear to trigger selective feature detection mechanisms in the visual system, discrimination is based on the second order statistics. Such a conclusion has led workers to consider Fourier methods in the analysis of discrimination experiments since the autocorrelation function, the inverse transform of the power

spectrum, is an expectation over the second order probability density.

Caelli [8] hypothesised that the visual system operates on the Fourier components of the texture using a 'clam shell' filter whose response varies with the frequency and orientation in the spatial frequency domain. A set of filters centred on different parts of the Fourier domain and of different orientation selectivity was assumed. Using a set of natural textures he concluded that there is a good degree of correlation in the spectral similarities and discriminability of the texture fields.

Harvey et al [9] investigated the relationship between the Fourier spectra of the texture and its perceptual appearance. In this work textures were grouped according to their perceptual similarity. They found a relationship between the spatial frequency content of the texture, as represented by the activity evoked in the four independent spatial frequency channels of the visual system, and the perceptual similarity of the textures.

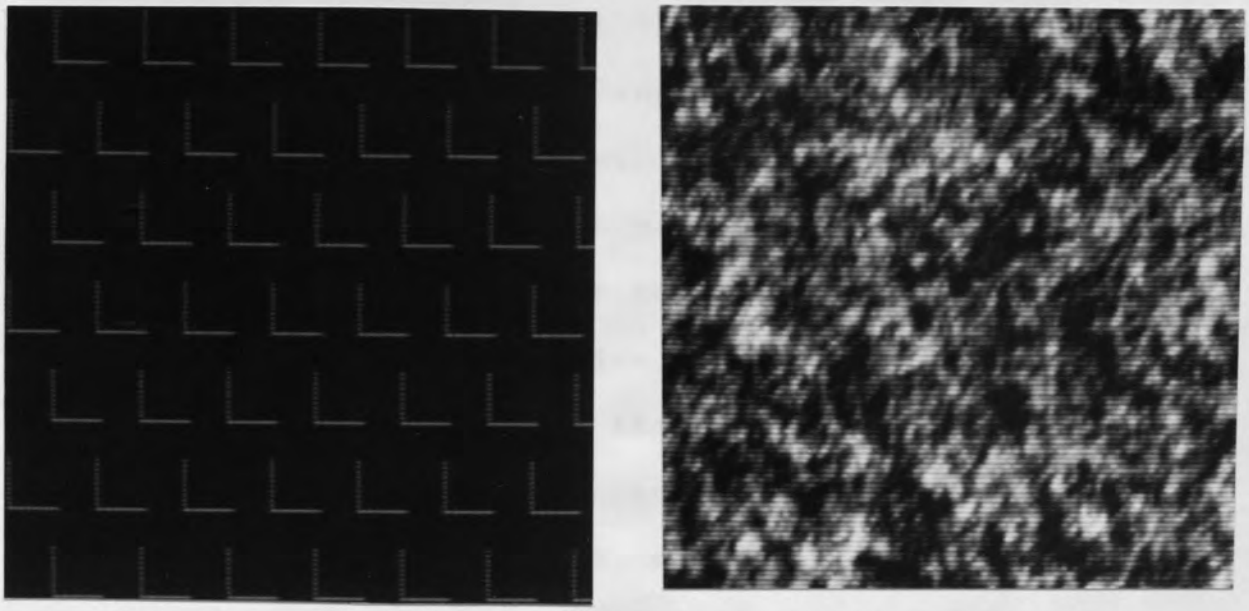
Note that in experiments of this type only the amplitude spectrum is considered. Eklundh [10] considered the use of the phase features in discrimination experiments and reported that they contributed virtually nothing to discrimination.

1.3 Textural Models

Many workers have attempted to model texture, that is, to specify a set of textures in terms of a small set of model parameters.A review of recent research into this problem was given by Haralick in [11]. He identified two distinct approaches-structural and statistical.The former describes the structural primitives of the texture and their placement rules. The latter models the texture in terms of the statistics of the chosen feature set.

Figures 1.1a and 1.1b illustrates a pair of textures generated using a structural and statistical model respectively. Figure 1.1a was generated by defining a local structural primitive as an 'L' shape and a translational rule which specifies the horizontal and vertical periodicities. In the case of figure 1.1b no definite local structural primitive is apparent. In general structural approaches to the modelling of texture lead to generative processes only. However, the parameters of a statistical model can often be estimated from the data, these parameters being used to characterise the original texture.

Various approaches to structural and statistical modelling will now be described.



(a) Structural (b) Statistical

Figure 1.1 Examples of structural and statistical textures.

1.3.1 Structural Models

A class of structural mosaic models can be generated by considering geometrical processes defined on the image plane [12]. These include cell structure models which consider the image plane to be tessellated into a set of bounded convex polygons with a probabilistic colouring model assigned to each cell, where the set of colours can be grey level or any other property. Also included are coverage models, which assume a random arrangement of geometric figures in the plane. This type of model is distinct from cell structure models in that the overlapping of figures is allowed and the colouring strategy at points of overlap must be specified [13].

A related approach is that of Modestino et al. [14] in which the image plane is tessellated using a Poisson line process, the rate of which controls the edge density in the image. A parameter is also defined which specifies the contrast between adjacent regions. This method uses a combination of structural modelling and statistical inferential techniques. In particular a statistical element is introduced into an essentially structural generative model, the statistical parameters being subsequently inferred from the data using parameter estimation techniques.

Zucker [15] defined an ideal texture as a perfectly ordered pre-defined primitive. This tessellates the image plane into a set of periodically arranged regions. He then defined a set of transformations which map from the ideal to observable textures, reflecting physical processes such as distortions to the the imaging process, projective transformations and so on.

1.3.2 Statistical Models

These methods characterise a texture by the statistics of pixel populations and their spatial relationships and in some sense can be considered as structural models with the structural element being condensed down to a single pixel.

A widely used technique is that of the co-occurrence matrix. The grey level co-occurrence matrix set P(d) is defined as a function of a vector d and whose (i,j)th element is the relative frequency of two pixels of spatial separation d having grey levels i and j. Haralick et al performed a number of image identification experiments using features of this matrix with some success [16].

It should be noted that the above technique is an estimate of the joint probability distribution of pixel pairs conditioned on a particular spatial separation. A related technique is the grey level difference method [17] which characterises a texture with an estimate of the first order probability distribution of the difference in a pair of pixel grey levels, again conditioned on spatial separation.

Statistics based on the grey level run length have also proved useful as a model parameter [18]. A grey level run is just a set of linearly adjacent pixels having the same grey level value. A matrix set R(a) is defined whose (i,j)th element specifies the relative frequency of a run of length j for grey level i in the direction a relative to some set of axis. In this case features from R(a) over a finite set of values of a are used to characterise the texture.

Measures based on the autocorrelation of a

textural field have also been adopted. This is equivalent to using a Fourier spectrum analysis approach as the discrete Fourier transform of the biased autocorrelation estimate of a signal is just the periodogram of that signal . Various workers have used spectral features in texture classification experiments. Weszka et al [17] integrated an estimate of the power spectrum over sectors of the Fourier domain between four radial spatial frequency values and four orientations thus obtaining sixteen features. They found that classification of LANDSAT images based on these spectral features in general performed worse than classification based on co-occurrence and grey level differences. However, this degradation in performance can be explained by noting that they used the periodogram of the signal as an estimate of the spectrum . Such an approach introduces spurious side lobes into the true spectrum which will obviously influence the classification. In order to solve this problem, Chen [19] used a maximum entropy method to estimate the spectrum. He reports classification results comparable to those using co-occurrence matrix features.

Pratt et al [20] employed a `time series' analysis approach to model texture. They considered a textural field to be the output of a linear spatial operator acting on an independent identically distributed random noise field. By adjusting the form of the

impulse response of the operator and the statistics of the noise field, they noted that they could predict the outcome of texture discrimination experiments based on the statistics of the texture field.

Related to 'time series' analysis is the technique developed by Deguchi et al [21] in which a non-causal autoregressive model is fitted to a texture field. The parameters of the model are computed in such a way as to minimise the mean squared error between the predicted and measured pixel values and are a function of the autocorrelation function of the texture. An extension of this technique is to use moving average terms in the pixel grey level estimate in addition to autoregressive terms thus increasing the flexibility of the method for constant model order [16].

Kashyap et al [22] extended simple autoregressive and moving average models to models with significant correlation between distant locations - well beyond the span of the AR or MA coefficients. This technique provides a means of parameterising natural textures using linear models.

1.4 Approaches to Image Segmentation

The problem of texture segmentation can be defined as the splitting up of an image into maximally

disjoint regions, each region having uniform texture. It differs from texture discrimination in that a spatial element is present. In other words, not only have the texture fields to be discriminated, they have also to be localised. Its uses includes scene analysis [23] in which texture can give a vital clue to the nature and orientation of physical surfaces. Also it is useful in object-background separation, in which perhaps the background is of uniform texture.

It should be noted that grey level segmentation can be viewed as a special case of texture segmentation and many approaches to this problem have been suggested [24]. Note that the region classification problem remains the same in both cases, but in the case of simple grey level segmentation the problem of feature extraction does not arise. More will be said about the detailed question of classification problems with spatial constraints in chapter 3.

Modestino et al [25] described a texture segmentation algorithm which assumes a stochastic tessellation model for a textural field [14]. Their segmentation algorithm is based on a maximum likelihood discrimination of a grey level co-occurrence matrix computed over a finite window. They reported encouraging results for a range of natural texture segmentations. However, it should be mentioned that their

method requires the algorithm to be presented with a sequence of training samples in order to compute necessary model parameters for the particular class of textures being used.

A simpler approach, adopted by Rosenfeld et al [26], is to compute a measure of texture coarseness at each image point and then to detect changes in this coarseness measure. In general the window size over which this measure is to be computed in order to obtain reliable results limits the spatial resolution in locating texture boundaries.

A more sophisticated gradient approach is due to Wermser et al [27]. They proposed a set of textural features based on the outputs of a set of non-isotropic filters of different bandwidths and orientations. A texture gradient measure is then computed in vertical and horizontal directions from the filter outputs. The attractive feature of this work is that it is based on local properties of the texture field over a range of region sizes and hence can accommodate textures of varying granularities. Also it is an unsupervised approach requiring no a-priori information.The authors reported moderately successful results for a variety of natural texture segmentations.

Coleman et al [28] report on a purely statistical

approach to this problem. A twelve element feature vector is defined at each image point and a clustering technique used to classify these vectors based on a K-means clustering algorithm [29]. Feature rejection is achieved using a Bhattacharya based measure and an eigenvector transformation applied to the remaining features to produce an uncorrelated feature set [29]. The results indicated that the subjective acceptability of a particular segmentation varies with the subset of features being used. Further, by ignoring the spatial position of each vector, spurious classifications occur which could be corrected by invoking some spatial constraints.

A segmentation algorithm based on a pyramid data structure [30] was developed by Burt et al [31]. Such a data structure consists of a set of images comprising different spatial resolution versions of the original image. In this segmentation scheme class membership is passed from lower to higher spatial resolutions whilst smoothing within classes takes place in the opposite direction. Hence processing occurs in both a top down and bottom up fashion. As will be seen a related approach is described in this thesis to perform segmentation but only top down processing is allowed. Note also that in that in Burt's method no spatial constraints are placed on the segmentation.

1.5 Thesis Summary

A statistical approach to texture analysis is adopted in the work described in this thesis. In particular it will be demonstrated in chapter 2 how a model based on the amplitude of the Fourier spectrum, hence ignoring phase information, is appropriate for a limited range of natural and synthetic textures. This is a special case of the Julesz conjecture relating to second order statistics. It will also be shown how a phase independent model will lead to second order stationarity for a particular class of phase distributions. The limitations of the spectral model will be demonstrated with a number of counterexamples.

Chapter 3 introduces a novel image segmentation algorithm based on a combination of global statistical methods and local spatial operations. This algorithm uses a multi-resolution data structure known as a quadtree to perform a downward directed boundary estimation procedure following a non-parametric classification.

In chapter 4 a series of experiments are described which compare the performance of this algorithm with that of a classical statistical technique based on Bayesian hypothesis testing for a set of synthetic Gaussian images. It will be seen that the algorithm compares favourably in performance with the

classical technique even though the latter is optimised for minimum probability of error and requires a priori information.

Chapter 5 describes an extension of this algorithm to the segmentation of vector fields resulting in its applicability to texture segmentation. A set of textural features are used which have maximum mutual energy concentration in the spatial and spatial frequency domains. It is argued that this approach is compatible with the phase independent model described in chapter 2.

Chapter 6 presents the results of this texture segmentation algorithm for a range of synthetic and natural textures.

Finally chapter 7 presents the suggestions for future work and conclusions.

Chapter 2

Phase in Image Texture

2.1 Introduction

In the previous chapter the work of several researchers suggesting texture descriptions and discriminations based on the second order statistics was introduced. It follows that resort must be made to second order spatial averages as estimates of the second order statistics, different estimates corresponding to different techniques. The two most widely used estimates are the co-occurrence matrix and the autocorrelation estimate. This chapter develops a model based on the autocorrelation and shows how it relates to intuitive notions of texture based on shift invariance.

As was mentioned in the context of the work of Gagalowicz [7], texture pairs with identical second order spatial averages can only be discriminated on the basis of local non-homogeneities. Hence it would appear that purely homogeneous texture, that is to say those devoid of any local structure, should be completely characterised by an adequate estimate of the autocorrelation function. Results confirming this conjecture for a set of natural and synthetic textures

will be presented. However, textures with significant local structure will be shown to be inadequately described by this model.

2.2 Mathematical Preliminaries

2.2.1 General Stochastic Processes

A discrete stochastic process x(n) is a sequence of random variables. In general the region of support of the process, defining the range of the index n, is infinite.

The process x(n) defines a family or ensemble of functions. In order to characterise the process statistically for finite N, it is necessary to specify the joint probability density p(..x(-1),x(0),x(1)..) which is a function of all the random variables x(n). Further, one can define expectations over the probability density. With E as the expectation operator, the mean $\eta(n)$, autocorrelation r(m,n) and autocovariance c(m,n) of x(n) are by definition :

$$\eta(n) = E(x(n)) = \int_{-\infty}^{\infty} x(n)p(x(n))\,dx(n) \qquad 2.1$$

$$r(m,n) = E(x(m)x^{*}(n)) = \int_{-\infty}^{\infty} x(m)x^{*}(n)p(x(m),x(n))dx(m)dx(n) \qquad 2.2$$

$$c(m,n) = r(m,n) - \eta(m)\eta^{*}(n) \qquad 2.3$$

In equations 2.1 and 2.2, p(x(n)) and p(x(m),x(n)) are the first and second order marginal probability distributions of the process at point n and at points m and n respectively. Note also that the average power of the process at point n is defined as :

$$E(|x(n)|^2) = r(n,n) \qquad 2.4$$

For most applications a complete specification of the statistics in the form of the joint probability distribution is unavailable. Also in some cases only a single realisation of the process is available, in which case it is common practice to substitute sample means for expectations. This is acceptable if the process is ergodic in the particular statistic in question [32].

One can consider transformations of the random process x(n) into a process y(n) by the application of an arbitrary operator O :

$$y(n) = O\{x(n)\} \qquad 2.5$$

In principle the statistics of y(n) are a function of the operator O and the statistics of x(n). For example, the Fourier transform of the process x(n) is the process X(u) given by :

$$X(u) = \sum_{n=-\infty}^{\infty} x(n) \exp(-jun) \qquad 2.6$$

where unit sampling rate has been assumed. For an infinite region of support the existence of X(u) is somewhat problematical, but for image processing applications there is no difficulty, as only finite supports are generally involved.

As a final point an NxN image can be considered to be a single realisation of a process x(m,n) defined for $0 <= m,n < N$. The extension of the theory outlined below to 2-dimensional cases is trivial.

2.2.2 Stationary Processes

A process x(n) is strict sense stationary if its statistical properties are invariant to a shift in origin. Hence the processes x(n) and x(n+c) have the same statistics for any value of the constant c. In other words any kth order marginal distribution of the process is such that :

$$p(x(n_1) \ldots x(n_k)) = p(x(n_1+c) \ldots x(n_k+c)) \qquad 2.7$$

Thus the first order marginal distribution p(x(n)) is independent of n. Further, the joint second order distribution p(x(m),x(n)) must, from equation 2.7 be such that :

$$p(x(m),x(n)) = p(x(m+c),x(n+c)) \qquad 2.8$$

and therefore must be a function of the difference m-n. This leads to a definition of a wide sense stationary process as one with a constant mean and an autocorrelation function that depends only on the difference between its arguments :

$$E(x(n)) = \eta(n) = \eta \qquad 2.9$$

$$E(x(m)x^{*}(n)) = r(m,n) = r(m-n) \qquad 2.10$$

Note that the definition of wide sense stationarity is less restrictive than that of strict sense stationarity as it does not apply to third and higher order distributions. Furthermore an ergodic process is wide sense stationary if the ergodicity applies to the mean and autocorrelation of the process [32].

An important property of a stationary process concerns the response of a shift invariant system to a stationary input. A shift invariant operator O is defined such that :

$$y(n+c) = O\{x(n+c)\} \qquad 2.11$$

iff

$$y(n) = O\{x(n)\}$$

Generally speaking, if the input to such a system is a

strict sense stationary process then its output is also strict sense stationary [32]. Note further that a linear shift invariant operator can be implemented as a convolution sum :

$$y(n) = h(n)*x(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) \qquad 2.12$$

which is a linear filtering operation.

In the context of the previous discussion on texture discrimination it was noted that homogeneous textures are those lacking any local structure. In this case the statistical properties characterising the texture are invariant of position and in this sense a texture can be regarded as a realisation of a stationary process.

2.2.3 Autocorrelation and Spectra

Rewriting equation 2.10 in terms of the difference d=m-n gives :

$$E(x(m)x^{*}(m-d)) = r(d) \qquad 2.13$$

The Fourier transform of r(d) is the power spectrum density S(u) :

$$S(u) = \sum_{d=-\infty}^{\infty} r(d)\exp(-jud) \qquad 2.14$$

A more intuitive notion of the power spectrum density can be gained by considering a process $[x(n)]_N$ which is zero outside some region of support of size N. Defining $[X(u)]_N$ as the Fourier transform of this process, the power spectrum density is defined as :

$$S(u) = \lim_{N \to \infty} \frac{1}{N} \overline{|X(u)|_N^2} \qquad 2.15$$

the bar indicating an ensemble average over all realisations of $[x(n)]_N$. It can be shown [33] that this definition is consistent with 2.14 if $[x(n)]_N$ converges to x(n) in a mean square sense. From equation 2.15, the power spectrum is independent of the phase of the transform and hence any estimate of S(u) should be independent of the phase of the components $[X(u)]_N$. Also, S(u) is a non-negative function of spatial frequency and consequently its inverse transform, the autocorrelation function is termed non-negative definite.

From equation 2.14 and the properties of the Fourier transform [34] :

$$r(d) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(u) \exp(jud)\, du \qquad 2.16$$

Hence :

$$E(|x(n)|^2) = r(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(u)\, du \qquad 2.17$$

Equation 2.17 states that integrating the power spectrum density across its entire range in the spatial frequency domain gives the mean square energy of the process.

As a final point, a process w(n) is termed white noise if its autocorrelation function is given by :

$$r(m,n) = I(n)\ \delta(m-n) \qquad 2.18$$

where $\delta(m-n)$ is the Kronecker delta which is zero if its argument is non-zero and unity otherwise and I(n) is the mean square energy of the process at position n. If w(n) is stationary then :

$$r(n) = I\ \delta(n) \qquad 2.19$$

and hence :

$$S(u) = I \qquad 2.20$$

Thus stationary white noise is sometimes known as flat noise as its power spectrum density is constant over all spatial frequencies.

2.2.4 Autocorrelation and Spectrum Estimates

In general, only a finite number N of samples, x(0)..x(N-1), of a single realisation of the process x(n) are available. Hence the autocorrelation function can only be estimated. Defining r(m) as the autocorre-

lation of the stationary process x(n) at lag m :

$$r(m) = E(x(n)\, x^{*}(n+m)) \qquad 2.21$$

its estimate $\hat{r}(m)$ can be defined as :

$$\hat{r}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} z(n)\, z(n+m) \qquad 0 \leq m < N \qquad 2.22$$

where z(n) is a single realisation of x(n). From equation 2.22 $\hat{r}(m)$ is biased since :

$$E(\hat{r}(m)) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} r(m)$$

$$= \frac{N - |m| - 1}{N}\, r(m) \qquad 2.23$$

However as N approaches infinity $E(\hat{r}(m))$ approaches r(m) and hence the estimate is asymptotically unbiased. Further, for finite N, as m approaches N the estimate $\hat{r}(m)$ becomes increasingly unreliable as the sum in equation 2.22 is computed over a small number of samples. This effect manifests itself as an increase in variance of $\hat{r}(m)$ as m approaches N. A so-called data window is often used to reduce this effect [34].

In order to introduce spectrum estimation for a process with finite region of support it is necessary to extend equation 2.22 :

$$\hat{r}(m) = \frac{1}{N} \sum_{n=0}^{N-1} z(n)\, z^{*}(n+m) \qquad 0 \leq m < N \qquad 2.24$$

where any index outside the range 0..N-1 is taken modulo N. Hence x(n) and therefore r(n) are assumed to be periodic with period N characterising a cyclo-stationary process [35]. Equation 2.24 is thus a circular convolution :

$$\hat{r}(m) = \frac{1}{N} z(n) * z^{*}(-n) \qquad 2.25$$

Taking the N-point DFT of each side of equation 2.24 and using the convolution theorem :

$$\hat{S}(u) = \sum_{m=0}^{N-1} \hat{r}(m) \exp(-2\pi j u m / N)$$

$$= \frac{1}{N} |\, Z(u) \,|^{2} \qquad 2.26$$

where :

$$Z(u) = \sum_{n=0}^{N-1} z(n) \exp(-2\pi j u n / N) \qquad 2.27$$

The spectrum estimate S(u) is known as the periodogram. Equation 2.26 should be compared with 2.15 for a process with infinite support. It should be mentioned that $\hat{S}(u)$ is an extremely unreliable estimate of the true spectrum of the process. The subject of spectrum estimation is specifically devoted to obtaining better spectrum estimates from a single realisation of the

process [36].

In view of equation 2.26 the spatial average given by 2.24 is phase independent. That is to say that if y(n) is an N-point signal with DFT given by :

$$Y(u) = A(u) \exp(j\phi(u)) \qquad 2.28$$

where A(u) and $\phi(u)$ are real functions, then its inverse transform y(n) has an autocorrelation estimate which is independent of $\phi(u)$.

In the context of texture analysis, one could define a class of textures as those NxN point signals with identical Fourier amplitudes. The extent to which the textures in a single class are perceptually equivalent will be considered in a later section.

It should be pointed out that such a phase invariance idea is reasonable for texture as the phase of the transform contains the positional information of a signal and homogeneous textures are devoid of local structure. As a simple example, the impulse function $\delta(n-r)$ at position r has a Fourier transform $\exp(-2\pi jur/N)$ and hence the phase contains the position of the impulse. These ideas are explored further in [37] where it is observed that intelligibility in an image is preserved if the Fourier amplitude is replaced with some suitable modelling function but the

phase left unchanged. Further it can be shown that, under certain conditions, the phase of a discrete signal uniquely specifies that signal [38].

2.3 A Random Phase Stochastic Process

The previous section suggested that textures within a single class, as defined above, can be considered as realisations of a stochastic process with an arbitrary phase. Specifically, given a real process x(n) defined for $0 <= n < N$, its DFT X(u) is such that :

$$x(n) = \frac{1}{N} \sum_{u=0}^{N-1} X(u) \exp(2\pi j u n / N)$$

$$= \frac{1}{N} [A(0) + \sum_{u=1}^{N-1} A(u) \cos(\frac{2\pi u n}{N} + \phi(u))] \qquad 2.29$$

where :

$$X(u) = X_R(u) + j\, X_I(u)$$

$$= A(u) \exp(j\phi(u)) \qquad 2.30$$

In equation 2.29 A(u) is the amplitude spectrum and $\phi(u)$ is the phase spectrum. Also note that, since x(n) is a real process, $\phi(0)$ is zero and A(0) is just the spatial summation of the process :

$$A(0) = \sum_{n=0}^{N-1} x(n) \qquad 2.31$$

Further, it is a simple matter to prove the following

symmetry relations :

$$A(u) = A(N-u) \quad 1 \leq u < N \qquad 2.32$$

$$\phi(u) = -\phi(N-u) \quad 1 \leq u < N \qquad 2.33$$

which follow as x(n) is defined as a real process.

The amplitude sequence A(u) in equation 2.29 can be assumed to be a given deterministic sequence and the phase $\phi(u)$ a random process. This will then define a process in which each realisation will have identical autocorrelation estimates as defined in equation 2.25 in view of equation 2.26 with :

$$A(u) = |Z(u)| \qquad 2.34$$

In principle the process x(n) is specified by the N-point joint probability density $p(\phi(0) .. \phi(N-1))$. A simplification is obtained by assuming that $\phi(u)$ and $\phi(v)$ are independent random variables for distinct u and v. Hence in this case :

$$p(\phi(0) \ldots \phi(N-1)) = \prod_{u=0}^{N-1} p(\phi(u)) \qquad 2.35$$

Note, however, since x(n) is real :

$$p(\phi(u)) = p(-\phi(N-u)) \quad 0 < u \leq N/2 \qquad 2.36$$

$$p(\phi(0)) = \delta(0) \qquad 2.37$$

The following section considers the expectation values

of the process and define the conditions for which x(n) becomes a stationary random process.

2.3.1 Computation of Averages

The expectation of x(n) follows from equation 2.29 :

$$E(x(n)) = [\frac{1}{N} \sum_{u=1}^{N-1} A(u)\ E(\cos(\frac{2\pi un}{N} + \phi(u)))] + \frac{A(0)}{N} \qquad 2.38$$

By definition of the expectation operator and from equation 2.35 this becomes :

$$E(x(n)) = \frac{A(0)}{N} + \frac{1}{N} \sum_{u=1}^{N-1} A(u) \int_{-\pi}^{\pi} \cos(\frac{2\pi un}{N} + \phi(u))\ p(\phi(u))\ d\phi(u) \qquad 2.39$$

where it is assumed without loss of generality that the phase lies between $-\pi$ and $+\pi$.

This integral is considerably simplified if it is assumed that the marginal probability distribution of the phase at each spatial frequency point u is periodic with period π :

$$p(\phi(u)) = p\ (\phi(u) + \pi) \qquad 2.40$$

Figure 2.1 illustrates a pair of possible distributions with this property, namely a uniform distribution and a delta distribution.

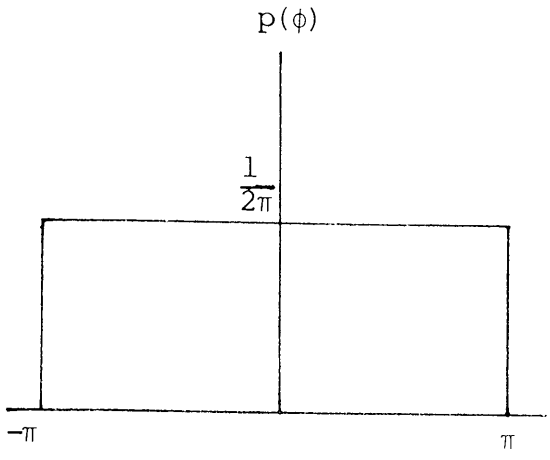


Figure 2.1a $p(\phi) = \frac{1}{2\pi}$

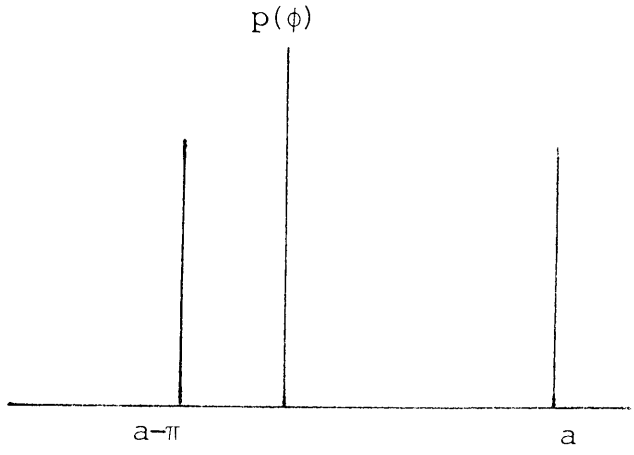


Figure 2.1b $p(\phi) = \frac{1}{2}(\delta(\phi-a) + \delta(\phi-a+\pi))$

Since :

$$\cos A = -\cos (A+\pi) \qquad 2.41$$

and in view of equation 2.40, the integral of equation 2.39 over the first half of its range exactly cancels the integral over the second half of its range and hence reduces to zero. In this case the expectation becomes :

$$E(x(n)) = \frac{1}{N} A(0) \qquad 2.42$$

From equation 2.31 this is just the sample average :

$$E(x(n)) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \qquad 2.43$$

Thus such a process is stationary in its mean with an ensemble expectation value which is the sample average

of any realisation of the process within the class, depending only on A(0).

Turning to the autocorrelation of the process, this is given by :

$$E(x(m)\ x(n)) = \sum_{u=1}^{N-1} A(u)^2 \int_{-\pi}^{\pi} [\cos\ (\frac{2\pi un}{N} + \phi(u))\ \cos\ (\frac{2\pi um}{N} + \phi(u))$$

$$p(\phi(u))\ d\phi(u)]$$

$$+ \sum_{\substack{u,v=1 \\ u \neq v}}^{N-1} A(u)\ A(v) \int_{-\pi}^{\pi} [\cos\ (\frac{2\pi un}{N} + \phi(u))\ \cos\ (\frac{2\pi vm}{N} + \phi(v))$$

$$p(\phi(u))\ p(\phi(v))\ d\phi(u)\ d\phi(v)]$$

$$+ A(0)^2$$

$$+ A(0) \sum_{u=1}^{N-1} A(u) \int_{-\pi}^{\pi} \cos\ (\frac{2\pi un}{N} + \phi(u))\ p(\phi(u)\ d\phi(u)$$

$$+A(0) \sum_{u=1}^{N-1} A(u) \int_{-n}^{\pi} \cos(\frac{2\pi um}{N} + \phi(u))\ p(\phi(u)\ d\phi(u) \qquad 2.44$$

Using the periodicity in the phase assumption of equation 2.40, terms 2, 4 and 5 of the above sum reduce to zero using similar reasoning as above. Further, using the well known trigonometrical identity :

$$\cos A \cos B = \frac{1}{2} (\cos (A-B) + \cos(A+B)) \qquad 2.45$$

equation 2.44 reduces to :

$$E(x(m)\ x(n)) = A(0)^2 + \sum_{u=1}^{N-1} A(u)^2 \int_{-\pi}^{\pi} [(\cos\ (\frac{2\pi u(n+m)}{N}) + 2\ \phi(u)) + \cos\ (\frac{2\pi u(n-m)}{N}\ ))\ p(\phi(u))d\phi(u)] \qquad 2.46$$

which further simplifies to :

$$E(x(m)\ x(n)) = \sum_{u=0}^{N-1} A(u)^2\ \cos\ (\frac{2\pi u(n-m)}{N}) \qquad 2.47$$

It can be seen that the autocorrelation is a function of m-n only and hence the process is wide sense stationary. Also the value of the autocorrelation is just the DFT of the real sequence X(u) and is simply the sample autocorrelation estimate as defined in equation 2.24 :

$$E(x(m)\ x(n)) = \frac{1}{N} \sum_{n=0}^{N-1} z(n)\ z(n+m) \qquad 2.48$$

In summary, the random process as defined by equation 2.29 is wide sense stationary if the phase consists of a sequence of independent random variables with a periodic probability density of period π. The mean and autocorrelation of the process are just the sample mean and biased sample autocorrelation of any realisation of the process.

2.3.2 Generation of a Class of Random Phase Textures

The previous section defined a class of discrete signals on a finite region of support as having identical Fourier amplitude spectra but differing in phase spectra. Such an idea can be applied to 2-dimensional textural fields with the key question being to what extent the textures within a given class are perceptually equivalent. In other words, can the textures be characterised by their amplitude spectra?

In order to generate one texture from another within the same class, it is simply a matter of computing the amplitude and phase spectra of the original texture using the DFT, replacing the phase spectrum with a suitable random sequence and transforming back to the spatial domain. Such a procedure will be termed phase randomisation.

Imposing the stationarity condition on the phase randomised texture implies that perceptual equivalence will not be maintained if the original can not be regarded as a realisation of a stationary random process. For example, local features in the original, such as line or edge segments, are removed by the phase randomisation procedure as such structures require positional information which is contained in the phase spectrum. That this should be the case is not surprising since, as will be demonstrated later,

phase randomisation is equivalent to replacing the original signal with a sum of randomly shifted versions of the original. Hence spatially local information is destroyed.

2.3.3 Interpretation of Phase Randomisation

To obtain a more quantitative statement about the effects of the phase randomisation procedure, consider a discrete signal x(n) defined for $0 <= n < N$. Also consider a set of circular shift operators S(0),S(1)..S(N-1) defined such that :

$$z(n) = S(i)\ x(n)$$

implies :

$$z(n) = x(n+i) \bmod(N) \quad 0 < n < N,\ 0 < i < N \qquad 2.49$$

Clearly :

$$S(0) = I$$

$$S(i)\ S(N-i) = I$$

where I is the identity operator. Consider the process y(n) :

$$y(n) = \sum_{i=0}^{N-1} a(i)\ S(i)\ x(n) \qquad 2.50$$

where a(i) is itself a process defined on $0 <= i < N$. Taking the N-point DFT of each side of equation 2.50 :

$$Y(u) = \sum_{i=0}^{N-1} a(i) \exp(2\pi jui/N)\ A(u) \exp(j\phi(u)) \qquad 2.51$$

where :

$$X(u) = A(u) \exp(j\phi(u))$$

is the DFT of x(n) and 2.51 follows from the fact that the DFT of a circularly shifted sequence simply introduces a multiplicative phase factor. Re-arranging equation 2.51 :

$$Y(u) = A(u) \exp(j\phi(u)) \sum_{i=0}^{N-1} a(i) \exp(2\pi jui/N) \qquad 2.52$$

If a(i) is restricted to an all-pass process :

$$\sum_{i=0}^{N-1} a(i) \exp(2\pi jui/N) = \exp(j\phi_a(u)) \qquad 2.53$$

where $\phi_a(u)$ is a random phase sequence, then in view of equations 2.52 and 2.53, the amplitude spectrum of the process y(n) is unchanged and hence is a phase randomised version of x(n) :

$$Y(u) = A(u) \exp j(\phi(u) + \phi_a(u)) \qquad 2.54$$

Note also that for a real process a(i) :

$$a(i) = \frac{1}{N} \left[1 + \sum_{u=1}^{N-1} \cos\left(\frac{2\pi ui}{N} + \phi_a(u)\right)\right] \qquad 2.55$$

The summation in equation 2.55 involves independent identically distributed random variables if the pro-

cess $\phi_a(u)$ consists of independent uniformly distributed random variables for each value of u. In this case the first order probability distribution of a(i) is independent of i. Thus all shifts are equally likely.

In view of the above discussion the phase randomisation procedure produces a class of signals that involve global shifts of the original. This implies that any local non-stationary information such as a line or an edge will be lost. Also a class of signals are produced that are spatially extended across the whole DFT block and hence textures consisting of spatially localised structures, such as figure 2.3a, will not be perceptually similar to their phase randomised versions.

2.4 Experimental Results

This section describes a set of experiments designed to test the validity of the ideas discussed above. In order to process as wide a range of textures as possible, a set of synthetic textures were generated. As these textures are used elsewhere in this report, the synthesis procedure will be outlined below.

2.4.1 A Set of Low Pass Synthetic Textures

Consider the following stationary random process producing impulse noise at a rate R [32] :

$$w(m,n) = I \quad U(0,1) < R$$

$$= 0 \quad U(0,1) > R \qquad 2.56$$

for $0 <= R <= 1$. This process generates a set of impulses of constant intensity I, the impulses being randomly generated across the image plane. The function U(0,1) is a random number uniformly distributed between 0 and 1. Such a process, as defined by equation 2.56, can be considered to be a digital Poisson point process [13].

In order to generate a textural field, the impulse noise field is convolved with a spatial impulse kernel h(m,n). This takes the form of a (generally) elliptical low pass filter with a specified width eccentricity and direction w,e and d respectively. An LxL operator is thus given by :

$$h(m,n) = \exp\left(-A\left(m_1^2 + \frac{n_1^2}{e^2}\right)\right) \qquad 2.57$$

where m_1, n_1 and A are given by :

$$m_1 = m \cos(d) - n \sin(d)$$

$$n_1 = m \sin(d) + n \cos(d)$$

$$A = -4 \ln(10) / (\omega L)^2 \qquad 2.58$$

for e>=1 and 0<= d <2π. If e is unity then equation 2.57 defines a Gaussian low pass impulse kernel.

Such a synthesis procedure allows control over intuitively important features of a texture, namely the granularity, the extent of non-isotropy and the directionality. Figures 2.2a-d show examples of 4 textures produced by this method for different values of w, e, d and R. In this and subsequent examples in this section all images are 128x128x8 bits.

The incoherence inherent in textures comes about through interaction of neighbouring convolution kernels. For example, if the mean inter-sample distance between impulses (which is a function of impulse rate) is less than the spatial extent of the convolution kernel in any direction then interaction occurs. Alternatively if the mean inter-sample distance is much greater than the extent of the convolution kernel then pseudo-periodic patterns consisting of the

individual elliptical kernels result. Thus by varying the impulse rate of the noise field the degree of textural incoherence can by controlled. This effect is demonstrated in figures 2.3a-d for 4 values of the impulse rate and for fixed convolution kernel parameters.

2.4.2 Phase Randomisation Experiments

The phase randomisation procedure as described in section 2.3.2 was carried out on the 4 synthetic textures of figures 2.2a-d. Also the procedure was carried out on the 3 natural textures grass, water and seafan which are taken from the Brodatz book of textures [39] and are shown in figures 2.4a-c.

The results are shown in figures 2.5a-d and 2.6a-c for the synthetic and natural textures respectively. Note that in this case a uniform probability density for the phase, as shown in figure 2.1a, was used. Using the impulse response type distribution, as shown in figure 2.1b, makes no difference to the result. This is demonstrated by phase randomising the textures of figures 2.2a and 2.2d using this distribution. The results are shown in figures 2.7a and 2.7b respectively.

In the case of the synthetic textures, although on a pixel by pixel basis, a texture and its phase

randomised counterpart are completely different, this procedure produces a perceptually similar texture. Further, phase randomising grass and water produce similar textures. However, in the case of the seafan texture, where significant local structure is involved, no such claim can be made. Hence a model based on the autocorrelation function for this texture is no longer appropriate.

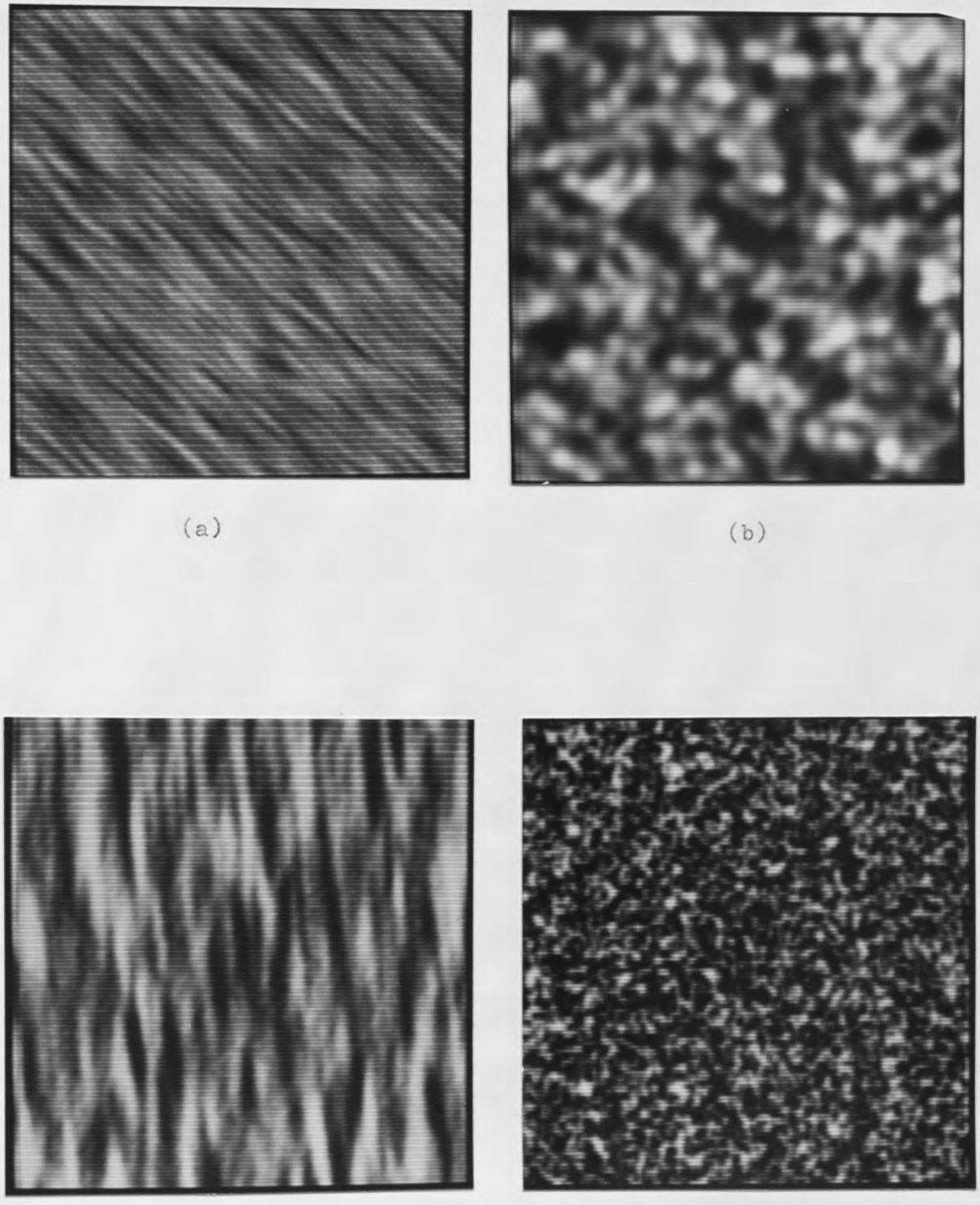
Although it is difficult to make quantitative assessments of perceptual similarities, by comparing figures 2.5b and 2.5d with their originals, the latter fine grained texture is closer to its original following phase randomisation than the coarser texture due to the presence of extended 'blob-like' local structure in the former texture.

2.5 Discussion

The signals produced by adding a random phase function to a pre-defined amplitude function have been shown to be realisations of a stationary random process if the phase has a periodic probability distribution in the sense of equation 2.40. It was also demonstrated that the phase randomisation procedure is equivalent to adding a set of globally shifted versions of the original. This globality is an essential feature of 'perceived stationarity' which can be

defined as those signals lacking any kind of local structure.

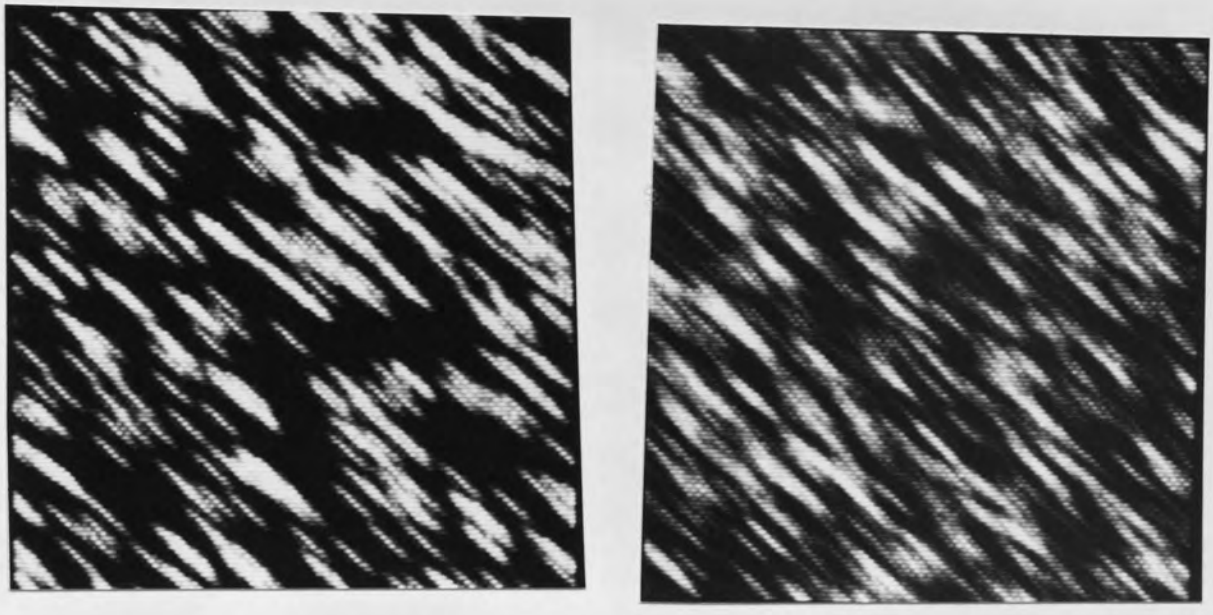
Although stationarity is a necessary condition for perceptual similarity following phase randomisation, it is not a sufficient condition. The random impulse field defined by equation 2.56 is a realisation of a stationary process [32]. However, due to the spatial localisation of the signal energy in any single realisation of the process, a phase randomised version will not resemble its original. In this case the stationarity is a result of spatially localised events being generated by a shift invariant process, that is to say the probability distribution of the random number U(0,1) in equation 2.56 is independent of position. For these kinds of signals statistical models are generally inappropriate since they require a finite region size over which to ascertain class membership as characterised by the model parameters. This is a consequence of the uncertainty principle [40] which places a limitation on simultaneous certainty both spatially and in class space due to the shift invariant nature of class defining operations. Phase randomising the seafan texture of figure 2.4c, for example, is an attempt to impose a global statistical model, characterised by the autocorrelation, on to a signal which is well defined spatially.



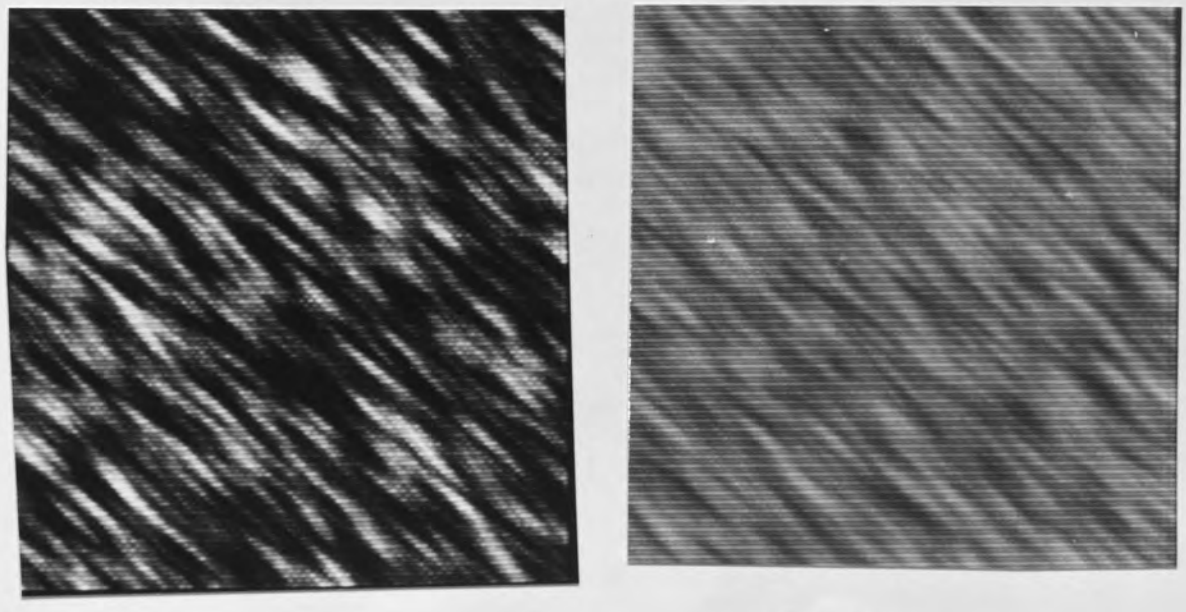
(a) (b)

(c) (d)

Figure 2.2 Synthetic textures generated by low pass filtering impulse noise.

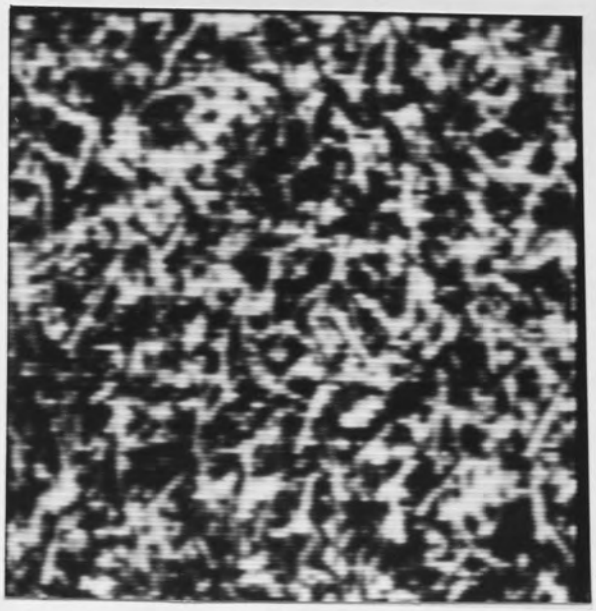


(a) IR=0.02 (b) IR=0.05

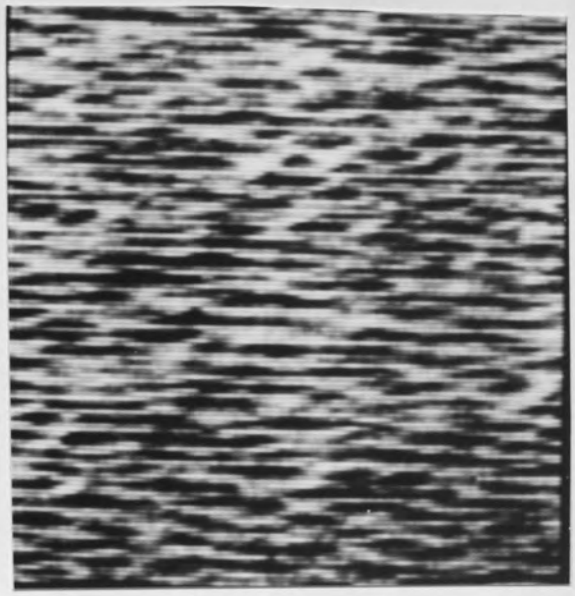


(c) IR=0.1 (d) IR=0.4

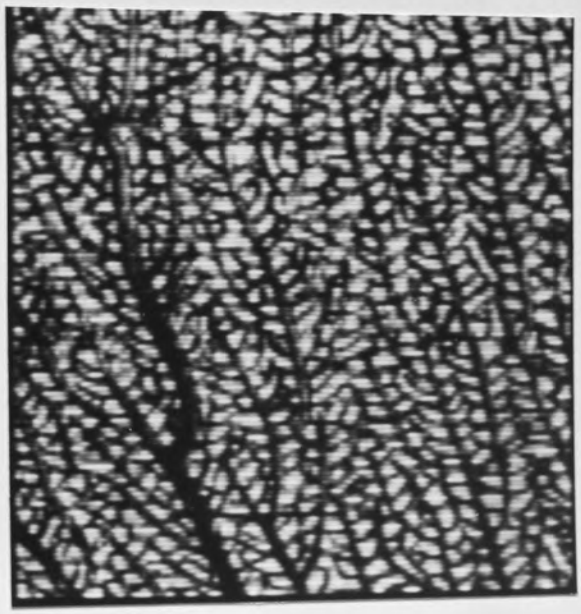
Figure 2.3 Effect of varying the impulse noise rate on the synthetic low pass texture of figure 2.2a.



(a) Grass

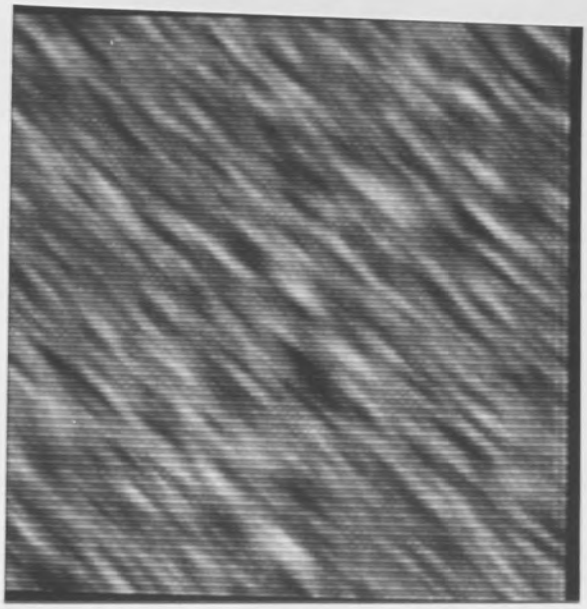


(b) Water

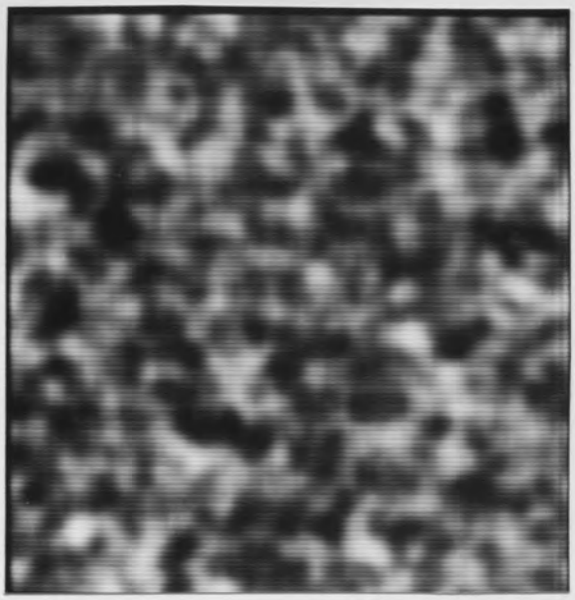


(c) Seafan

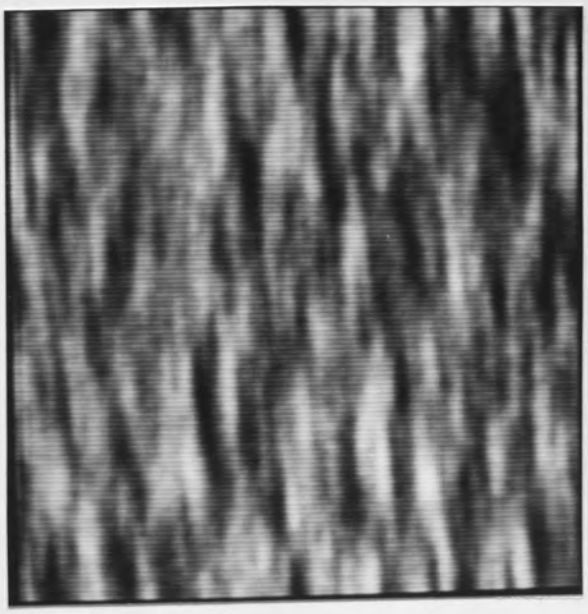
Figure 2.4 Natural textures.



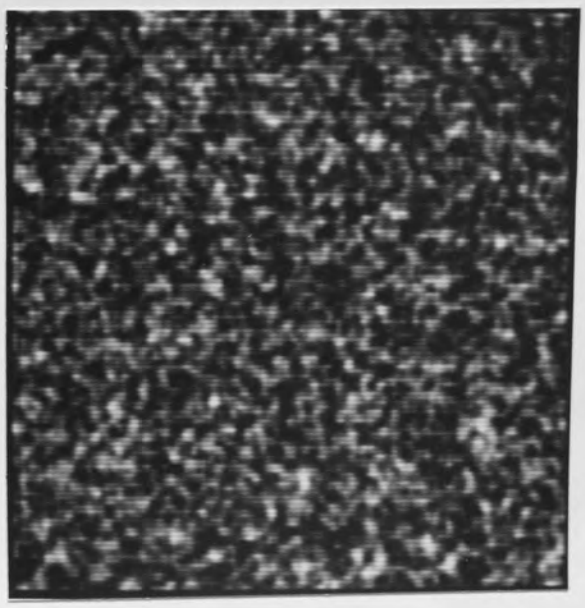
(a)



(b)

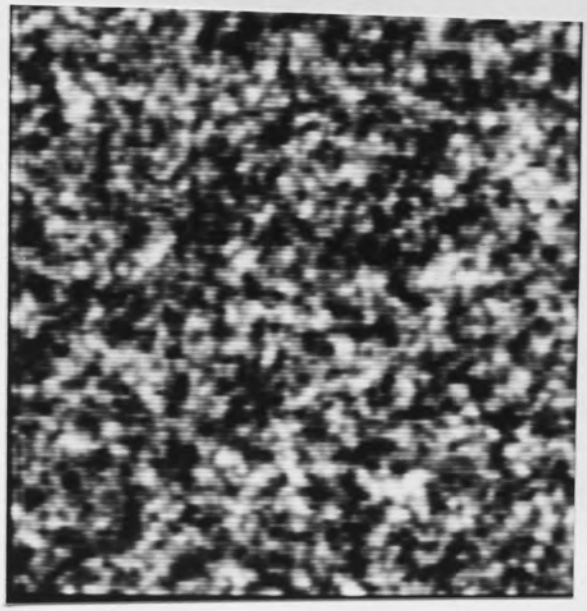


(c)

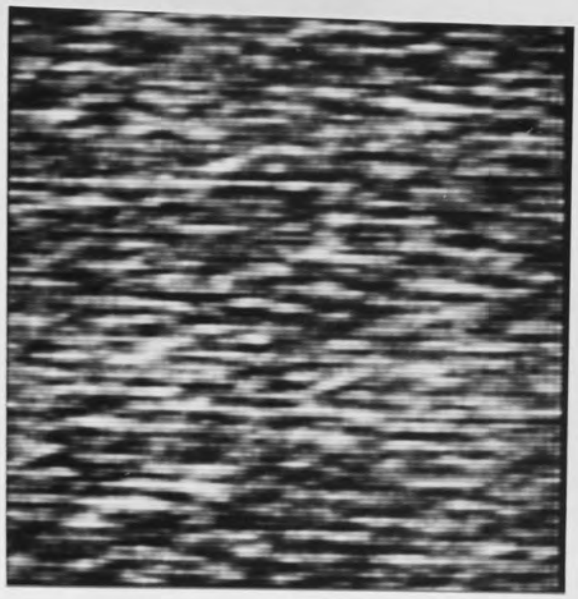


(d)

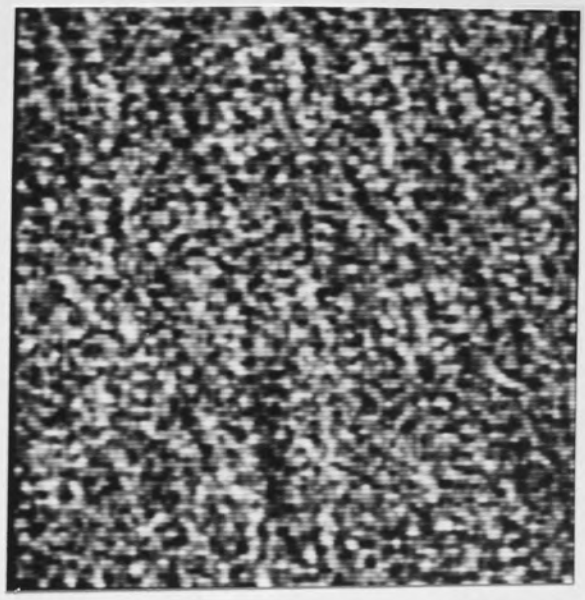
Figure 2.5 Synthetic textures of figure 2.2 with the phase replaced by a uniformly distributed random variable at each spatial frequency.



(a) Grass

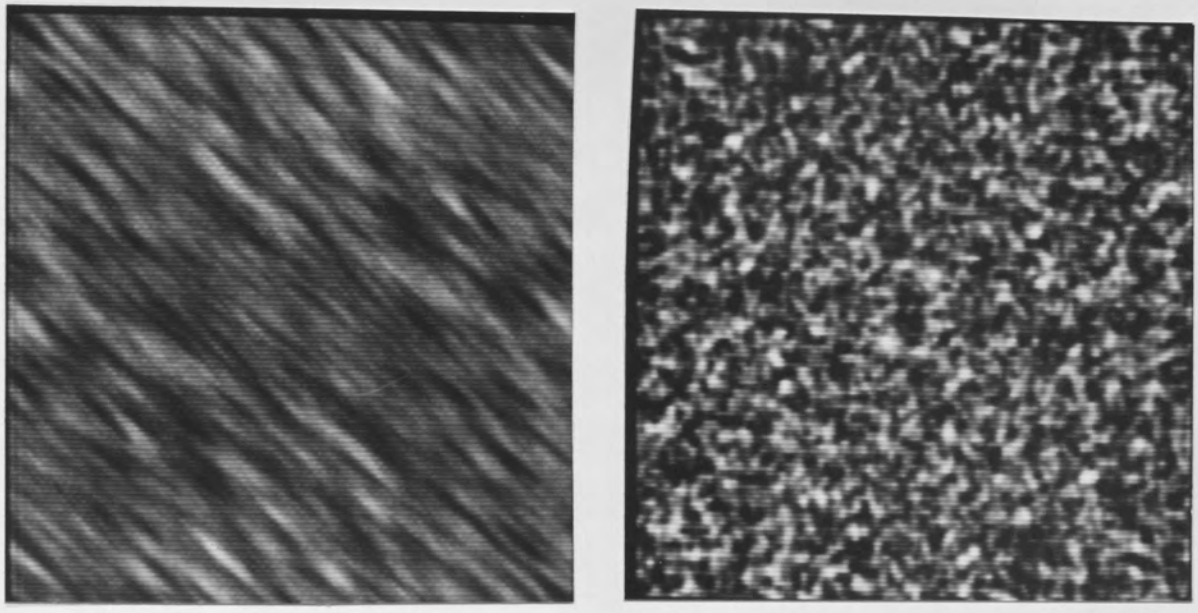


(b) Water



(c) Seafan

Figure 2.6 Natural textures of figure 2.4 with the phase replaced by a uniformly distributed random variable at each spatial frequency.



(a) (b)

Figure 2.7 Synthetic textures of figures 2.2a and 2.2d with the phase at each spatial frequency being replaced by a random variable with the probability distribution shown in figure 2.1b.

Chapter 3

A Novel Image Segmentation Algorithm

3.1 Introduction

The previous chapter described a statistical model of texture based on the Fourier amplitude spectrum. A key question is whether this model can be used to segment out regions of uniform texture. The rest of this thesis is devoted to this problem.

Texture segmentation is an example of a general classification problem in pattern recognition [29] as shown schematically in figure 3.1.

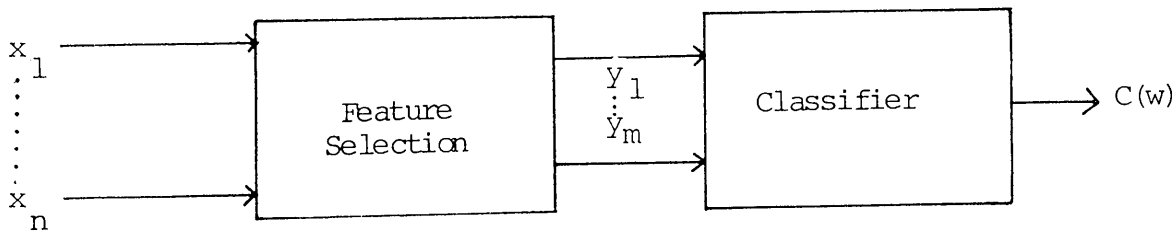


Figure 3.1 General pattern recognition system

The purpose of the system of figure 3.1 is to take an observation vector $X=(x_1,x_2 \ldots x_n)^T$ and classify it into one of a number of classes based on some predetermined criteria. In many applications the raw observation data are not very effective features for classification purposes. Feature selection involves

mapping the original observation vectors into a set of feature vectors which are more suitable for classification. Chapter 5 considers this problem in detail.

In general the classifier has to deal with noisy signals, that is to say signals in which the noise within a class is significant compared to the difference in mean signal level between classes. This chapter describes a general classifier, independent of the feature selection process, which is able to deal with noisy signals of any dimensionality. For the purposes of clarity only 1-dimensional signals will be discussed here- extensions to a multi-dimensional signal space, where appropriate, will be described in chapter 5. In this case the input to the classifier can be simply the image grey level and, as will be seen from the results to be presented in chapter 4, the classifier can perform satisfactory grey level segmentation.

3.2 Statistical Approaches to Image Segmentation

Classical pattern recognition techniques usually classify data on the basis of its statistics. The most common of these is Bayesian classification. In this case, if $p(\underline{x}|w_i)$ is the joint probability distribution of the random data vector $\underline{x}$ conditioned on it belonging to class w_i, then Bayesian classification, assum-

ing equal costs, amounts to [29] :

$$p(\underline{x}|w_1)\ p(w_1) \gtrless p(\underline{x}|w_2)\ p(w_2) \rightarrow \underline{x} \in \begin{cases} w_1 \\ w_2 \end{cases} \qquad 3.1$$

In equation 3.1 $p(w_i)$ is the a priori probability of the vector belonging to class w_i. Note that this equation implicitly assumes that only 2 classes are present and also assumes knowledge of the relevant probability distributions.

In the case of grey level segmentation, the probability distributions of equation 3.1 can be replaced by grey level histograms as estimates of these distributions. Figure 3.2 illustrates that the Bayesian approach is roughly equivalent to identifying classes with modes in the local histograms

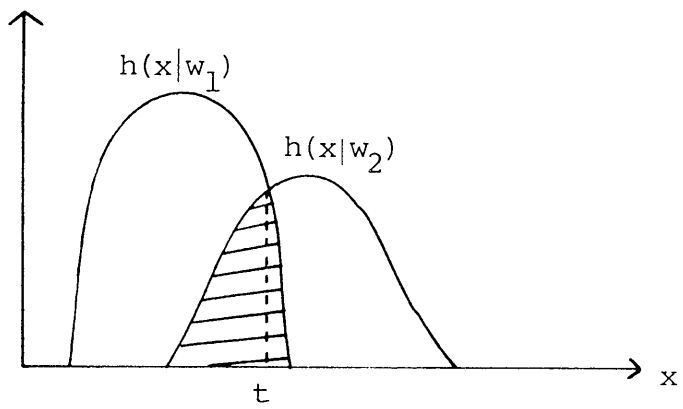


Figure 3.2 Interpretation of Bayesian classification

Hence in figure 3.2, assuming equal a priori probabilities, if the grey level x is greater than some threshold t then it is assigned to class 2 and vice versa. Thresholding techniques based on this principle have

been applied to segmentation problems [41], [42]. Note that the shaded region of figure 3.2 represents the classification error that would result in choosing threshold t. Indeed in many cases of interest, the global histogram of the data is unimodal yet the classes are clearly visible.

Segmentation by partitioning into homogeneous regions is generally more powerful than the simple pixel classification techniques mentioned above. An example is the split and merge algorithm of Pavlidis [43] in which statistical decision theory is used to split regions if they are not sufficiently homogeneous and merge pairs of adjacent regions if their union is still homogeneous. In this case the local nature of the segmentation problem is recognised, but the success of the approach is dependent on the accuracy of the statistical assumptions.

The scheme proposed in this chapter combines region based classification with a local pixel based thresholding procedure in an attempt to overcome several problems inherent in segmentation which will be outlined in the next section.

3.3 Problems Inherent in Segmentation

3.3.1 A Priori Knowledge

A segmentation scheme requiring a priori knowledge implies that it can only deal with the corresponding classes of input signal. Hence it is not the most general scheme possible. Of course, the importance of such a limitation is problem dependent.

Typical of an a priori requirement is that the number of classes to be segmented out be known. This is always the case in Bayesian based methods as a comparison of the costs of choosing between a fixed number of alternatives is computed in order to make the classification. Also it is a frequent limitation in clustering algorithms, for example in the K-means and related algorithms [44].

Another possible a priori requirement is that the statistics of the classes be known. However, only global statistical measures are available to an algorithm before any segmentation. This problem is related to the uncertainty principle [40] in the sense that increased certainty of class membership is at the expense of reduced certainty as to the position of the classes. Such a dilemma also warrants that an algorithm rely on the minimum amount of statistical parameterisation possible.

3.3.2 Spatial Considerations

As was mentioned in the previous section, apart

from their other weaknesses, there is a considerable error inherent in purely statistical techniques based on Bayesian decision theory. However, it is a matter of common observation that the visual system viewing an object in background noise say, only makes classification errors around the object boundary. (See, for example, figures 4.13 and 4.14.) This is because it is making assumptions about spatial properties of the object, such as convexity and compactness. This in turn implies that the classification relies on the spatial position as well as its position in class space. For example, if a single point is surrounded by a set of points that have been statistically classified differently, then the chances are that it has been misclassified. Such ideas must be incorporated into segmentation algorithms in as natural a way as possible.

3.3.3 Spatial Resolution/Class Resolution Trade-offs and Uncertainty

Image segmentation involves making decisions both spatially and also in class space simultaneously. Specifically the classification of regions has to be ascertained as well as the location of the region boundaries. These two requirements are mutually contradictory in that an increase in resolution in class space is at the expense of a loss of resolution

spatially. For example, if one uses simple averaging to reduce the effects of noise, a larger averaging blocksize will increase the certainty of class membership since the variance of the smoothed signal is reduced, but also implies that any region boundaries are smeared out over a larger area.

The above dichotomy is another example of the uncertainty principle in image processing [40], [45]. Uncertainty, moreover, is a restriction that the human visual system is well able to incorporate into its information processing tasks [46]. Indeed, as will be seen in the next chapter, for simple grey level segmentation of noise fields, the algorithm described below detects region boundaries to accuracies of 2 or less pixels even in the presence of considerable intraclass noise. Such a performance was observed to be roughly comparable to that of the human visual system in this case.

In order to solve this problem, processing is performed at multiple levels of resolution, the results at a given level being used to influence the results at a lower level (higher spatial resolution). Such an approach would appear to have a precedent in the human visual system, for example in edge detection [47] and stereo matching [48]. A data structure known as a quadtree is used to represent the image data over

a range of scales in class space and hence over a range of spatial resolutions.

3.4 Properties of Quadtrees

Consider an NxN image d(i,j) defined for $0<=(i,j)<N$ and $N=2^m$. The quadtree [49] of this image is defined as :

$$q(i,j,k) = \frac{1}{4} [\, q(2i,2j,k-1) + q(2i+1,2j,k-1) + q(2i,2j+1,k-1) + q(2i+1,2j+1,k-1) \,] \qquad 3.2$$

where :

$$0 < k \leq m, \quad 0 \leq (i,j) < 2^{m-k}$$

and :

$$q(i,j,0) = d(i,j)$$

Hence a quadtree is based on 2x2 block averaging. The level just above the base consists of nodes representing non-overlapping 2x2 blocks of pixels in the original image so that the size of this level is $2^{m-1} \times 2^{m-1}$. This process can be repeated until the root node is reached, whose value is the average grey level of the entire image. Figure 3.3 represents the averaging procedure where a father node is linked to its four sons.

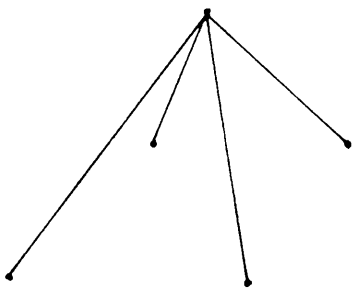


Figure 3.3 A father node and its four sons

Note that any property, not just the image grey level can be associated with a node of the quadtree.

It should be mentioned that other pyramid structures can be used. For example, Hong et al [50] described a segmentation scheme which operates on a pyramid data structure produced by a 4x4 block averaging procedure with a 50% overlap of the averaging blocks. In this case each son can have 4 possible fathers and each father has 16 sons.

3.4.1 Smoothing Gain

Quadtrees are useful in image segmentation because after each averaging process, producing the next quadtree level, the variance of the signal within a single homogeneous region is reduced. Hence if x(i,l) is the random variable corresponding to the ith node at the lth quadtree level and assuming that the signal at the lth level inside a homogeneous region is

wide sense stationary then :

$$x(i,l) = \frac{1}{4} \sum_{j=1}^{4} x(j, l-1) \qquad 3.3$$

Assuming, without loss of generality that the random variables $x(i,l)$ have zero mean, then the variance of the ith node at level l is given by :

$$\sigma^2(l) = E(x(i,l)^2) \qquad 3.4$$

which is independent of i because of the stationarity assumption. Substituting equation 3.3 into equation 3.4, this becomes :

$$\sigma^2(l) = \frac{1}{16} \sum_{j,k=}^{16} r(l-1, j-k) \qquad 3.5$$

where $r(l-1,j-k)$ is the autocorrelation between nodes j and k at level $l-1$. The summation of equation 3.5 can be split up to give :

$$\sigma^2(l) = \frac{1}{4} \sigma^2(l-1) + \frac{1}{16} \sum_{\substack{j,k=1 \\ j \neq k}}^{4} r(l-1, j-k) \qquad 3.6$$

Finally note that :

$$r(l,j-k) = E(x(l,j)x^{*}(l,k))$$

$$= \frac{1}{16} \sum_{j',k'=1}^{4} E(x(l-1,j')x^{*}(l-1,k'))$$

$$= \frac{1}{16} \sum_{j',k'=1}^{4} r(l-1,j'-k') \qquad 3.7$$

In view of equation 3.7, if the nodes inside a homogeneous region of the original image are uncorrelated random variables, then the corresponding nodes at any higher level are also uncorrelated. Thus in this case equation 3.6 becomes :

$$\sigma^2(l) = \frac{1}{4} \sigma^2(l-1) \qquad 3.8$$

In the more general case of the variables being correlated, it can easily be shown that [32] :

$$|r(l-1,j-k)| \leq \sigma^2(l-1) \qquad 3.9$$

and hence :

$$\sigma^2(l) \leq \sigma^2(l-1) \qquad 3.10$$

Clearly the degree of reduction in the variance of the nodes at higher quadtree levels depends on the correlation between nodes at the bottom of the tree.

If a pair of adjacent homogeneous regions have

signal means and variances of (m_1, σ_1^2) and (m_2, σ_2^2) then an inter-region signal to noise ratio $\rho_{12}(l)$ at quadtree level l can be defined as :

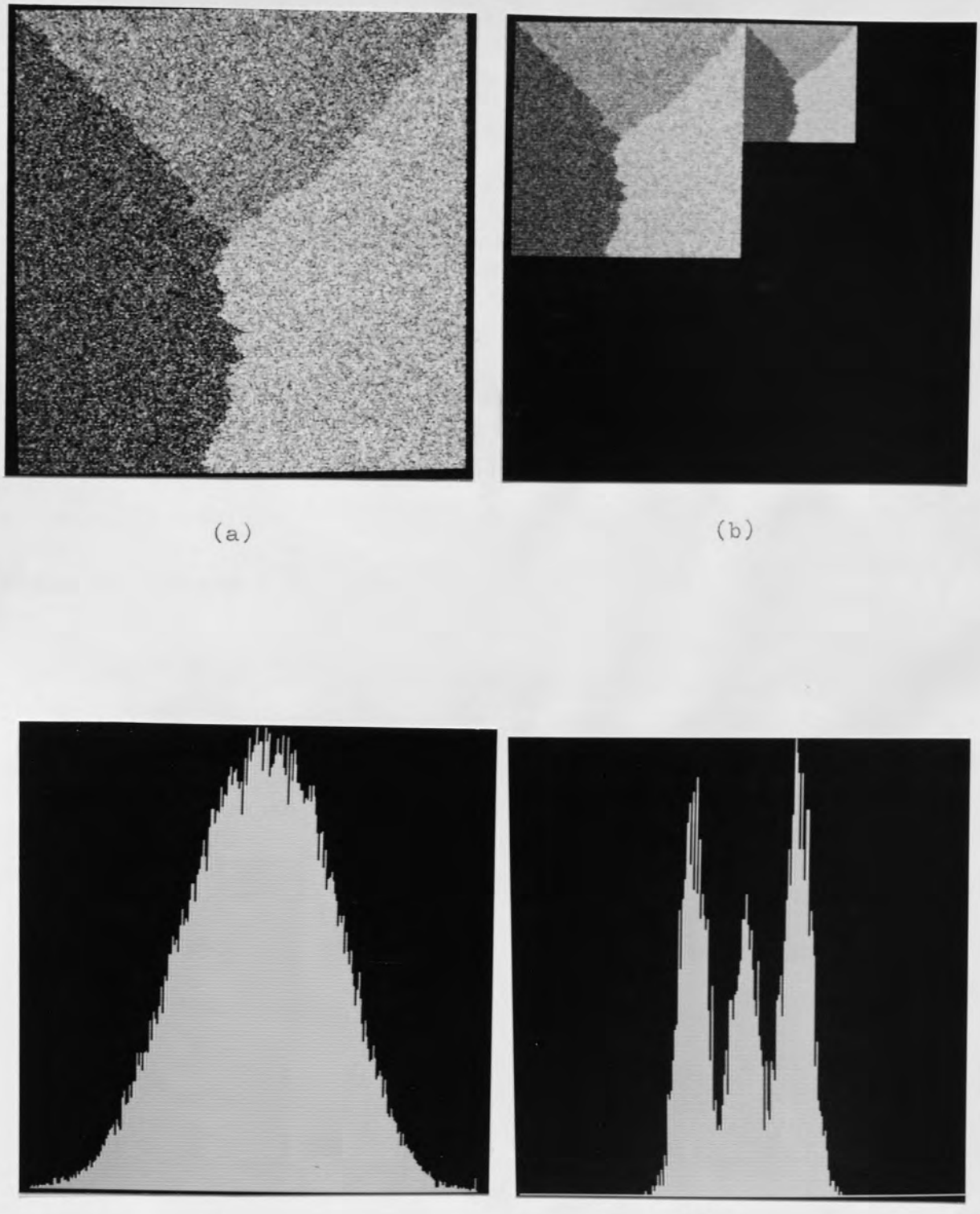
$$\rho_{12}(l) = 2 \frac{|m_1(l) - m_2(l)|}{\sigma_1(l) + \sigma_2(l)} \qquad 3.11$$

Clearly the mean of a homogeneous region remains approximately constant for each quadtree level. Hence in view of equation 3.10 :

$$\rho_{12}(l) > \rho_{12}(l-1) \qquad 3.12$$

Thus quadtree smoothing can be seen as a means of increasing the separation of class pairs in class space.

Figures 3.4a-d show 3 regions of white Gaussian noise, each with a different mean grey level, together with the next 2 quadtree levels. Also shown are the grey level histograms of the first and third quadtree levels indicating an increase in class resolution. In this case the signal to noise ratio between any pair of regions is approximately 1.5 and the global histogram is unimodal as can be seen from figure 3.4c. Subsequent quadtree smoothing operations increase the signal to noise by a factor of 2 in each case as only white noise is involved. Figure 3.4d demonstrates that at the third quadtree level, the 3 classes are



(a) (b)

(c) (d)

Figure 3.4 Effect of quadtree smoothing.

clearly visible.

3.4.2 Aliasing

The quadtree smoothing operation as defined in equation 3.2 can be viewed as a special case of a general low pass filtering / sub-sampling operation of the form :

$$q'(i,j,k) = h_k(i,j) * q(i,j,k)$$

$$q(i/2,j/2,k+1) = q'(i,j,k) \qquad 3.13$$

where as before :

$$q(i,j,0) = d(i,j)$$

d(i,j) being the original NxN image for $N=2^m$. In equation 3.13 $h_k(i,j)$ is a low pass filter. In the case of the quadtree, the low pass filter is just a 2x2 equal weight impulse response :

$$h_k(i,j) = \frac{1}{4} \qquad 0 \leq i,j < 2$$

$$= 0 \qquad \text{otherwise}$$

The DFT of this response for k=0 is then :

$$H_0(u,v) = \frac{1}{4}\left[\, 1 + \cos\frac{2\pi u}{N} + \cos\frac{2\pi v}{N} + \cos\frac{2\pi(u+v)}{N}\right.$$

$$\left. -j\left(\sin\frac{2\pi u}{N} + \sin\frac{2\pi v}{N} + \sin\frac{2\pi(u+v)}{N}\right)\right] \qquad 3.14$$

Taking the modulus of equation 3.15 it can easily be shown that the amplitude spectrum is given by :

$$|H_0(u,v)| = \frac{1}{2}\left[\, 1 + \cos\frac{2\pi u}{N} + \cos\frac{2\pi v}{N} + \cos\frac{2\pi u}{N}\cos\frac{2\pi v}{N}\,\right]^{\frac{1}{2}} \qquad 3.15$$

The section $|H_0(u,0)|$ of this function is sketched in figure 3.5 where a centred coordinate system $-N/2 <= u < N/2$ has been used.

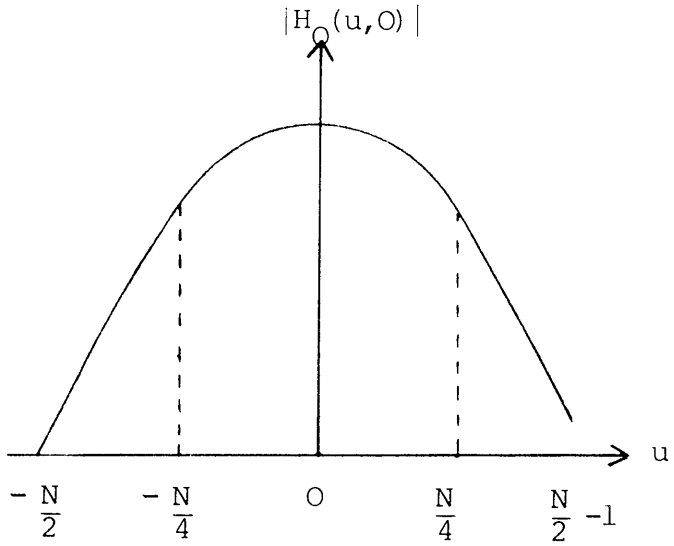


Figure 3.5 The amplitude spectrum $|H_0(u,0)|$

From figure 3.5 there is a non-zero value of the frequency response for $|u| > N/4$. In view of the sub-sampling operation of equation 3.13, those frequencies above N/4 are aliased, that is to say wrapped around and added to the low frequency part of the spectrum [51].

3.5 Quadtree Image Segmentation [52] [53]

3.5.1 Overview

In order to solve the problem of simultaneous high spatial resolution and high resolution in class space a multi-resolution approach is adopted, information gleaned at each resolution being used to control processing at the level below.

Specifically, using a quadtree data structure, a non parametric clustering algorithm is applied to the highest quadtree level (lowest spatial resolution, maximum smoothing gain.) The algorithm chosen requires no a priori class information and hence increases the generality of the segmentation scheme.

Using the class information obtained from the clustering, a downward directed boundary estimation procedure is performed. At each quadtree level a firm segmentation takes place where boundary information obtained from the previous level controls the segmentation at the current level.

Also note that the scheme is adaptive at each level, in that the segmentation relies on the estimated signal to noise ratio between pairs of regions.

The reasoning behind such a multi-resolution approach is that a boundary between a pair of homogeneous regions must appear at each quadtree level given that each region is significantly larger than the largest averaging operator used to create the quadtree. Hence if a boundary can be located at the highest level then this limits the search for the boundary at the level below.

Figure 3.6 outlines the structure of the algorithm and the following sections describe each stage in detail.

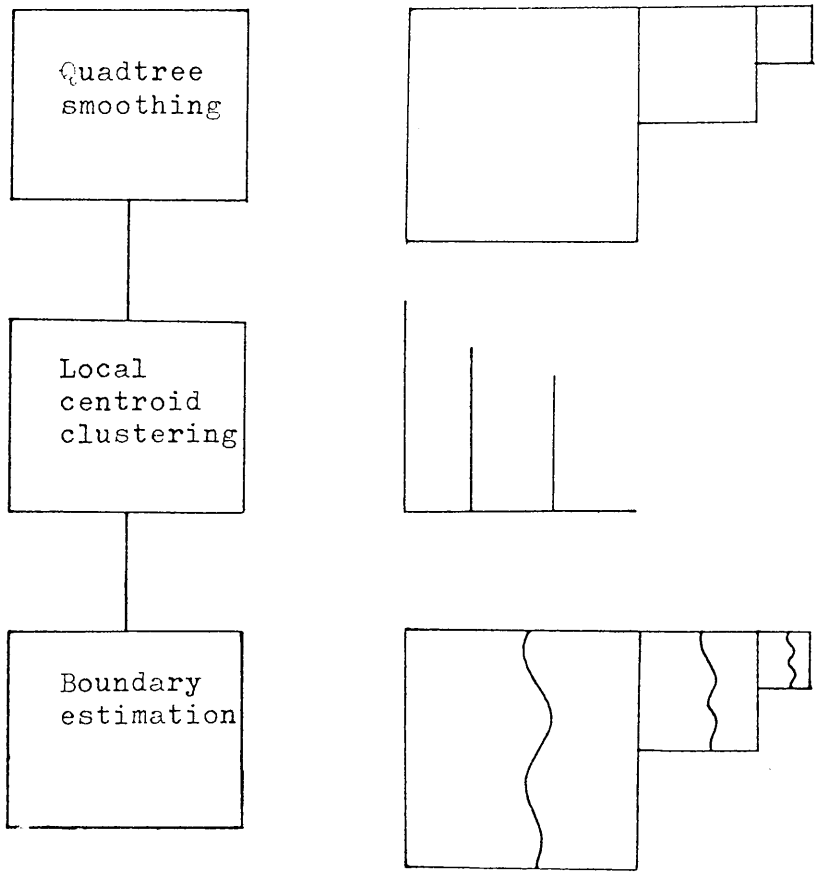


Figure 3.6 Algorithm structure

3.5.2 Quadtree Smoothing

Section 3.4.1 described the smoothing gain property of the quadtree which is crucial to the success of the statistical classifier at the highest quadtree level. Note, however, that the smoothing procedure also introduces a bias due to the merging of data from different regions.

Consider the case in which the data consists of 2 regions with means μ_1 and μ_2 each with white additive noise of variance σ^2 [52]. Let the probability that a node at level k has children with mean μ_1 be p_k . The mean and variance at level k are then given by :

$$m_k = E(q(i,j,k)) = p_k \mu_1 + (1-p_k) \mu_2 \quad k>0 \qquad 3.16$$

$$\sigma_k^2 = 2^{-2k} \sigma^2 \qquad 3.17$$

Defining :

$$\rho_k = \frac{m_k - \mu_2}{\sigma_k} \qquad 3.18$$

and using equations 3.16 and 3.17, this becomes :

$$\rho_k = 2^k p_k \rho_0 \qquad 3.19$$

where in equation 3.18 m_0 is replaced by μ_1 .

From equation 3.19, p_k can be seen to be a

correction factor which depends on the geometry of the situation. For example, consider the case where the region with mean μ_1 is a convex object of radius $r=2^n$ with $k << n$. At level k the radius of the object is approximately $r/2^k$ of which a fraction 2/r are perimeter nodes (for simple convex objects eg. squares, circles etc). Of these nodes, on average 50% have descendants at level 0 of the quadtree which are outside the object. Hence in this case p_k is given by :

$$p_k = 1 - \frac{2^k}{r}$$

$$= 1 - 2^{k-n} \qquad 3.20$$

Clearly the computation of p_k is in general a difficult task. However, if estimates can be made then this leads to an expression for the optimum level of smoothing - that is to say the value of k such that ρ_k is maximum.

3.5.3 Local Centroid Clustering [54]

3.5.3.1 Local Centroids

Consider the probability distribution p(x) where of course :

$$p(x) > 0 \qquad \text{all } x$$

$$\int_{-\infty}^{\infty} p(x)\,dx = 1 \qquad 3.21$$

The local centroid defined at each point x in class space is then given by :

$$\mu(x) = x + \frac{\int_{-X}^{X} x'\,p(x+x')\,dx'}{\int_{-X}^{X} p(x+x')\,dx'} \qquad 3.22$$

Equation 3.22 states that the local centroid at point x is just the centre of mass of the probability distribution calculated over a window of size 2X and centred on x as illustrated in figure 3.7. Note that equation 3.22 can trivially be extended to discrete class space by replacing the integrals by summations. Further, in practical situations, only an estimate of the probability distribution in the form of the global histogram h(x) of the data, is available. Of course equations 3.21 still apply in this case. As a final point equation 3.22 can be seen as a generalised filtering type operation on the probability distribution, the filter impulse response w(x) being the sawtooth function :

$$w(x) = x \qquad |x| < X \qquad 3.23$$

$$= 0 \qquad \text{otherwise}$$

Hence equation 3.22 can be implemented using the DFT

algorithm which represents a considerable computational saving for large values of X. However, for multi-dimensional class space, the local centroid can be even more efficiently computed by considering only the non-zero values of the probability distribution.

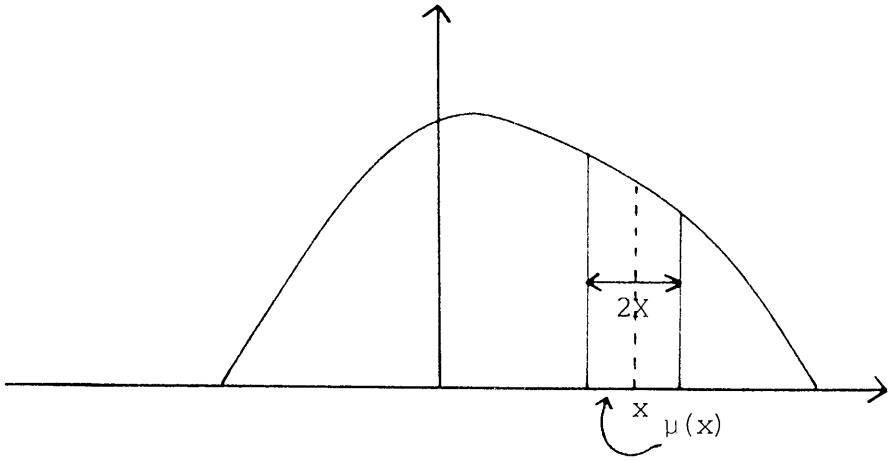


Figure 3.7 Computation of the local centroid

3.5.3.2 Non-Overlapping Class Distributions

In the case of a global probability distribution consisting of the sum of a set of non-overlapping local distributions, it can be shown that the definition of local centroid preserves local class means. To this end consider the distribution :

$$p(x) = \sum_{i=1}^{N} p_i(x) \qquad 3.24$$

Define the local means μ_i such that :

$$\mu_i = \frac{\int_{\mu_i - W}^{\mu_i + W} x\, p_i(x)\, dx}{\int_{\mu_i - W}^{\mu_i + W} p_i(x)\, dx} \qquad 3.25$$

In this case the distributions $p_i(x)$ are defined such that :

$$p_i(x) = 0 \qquad \mu_i - \frac{W}{2} < x < \mu_i + \frac{W}{2}$$

and :

$$|\mu_i - \mu_j| > 2W \quad \text{all } (i,j) \qquad 3.26$$

An example of a distribution $p(x)$ satisfying these constraints is given in figure 3.8 for N=2.

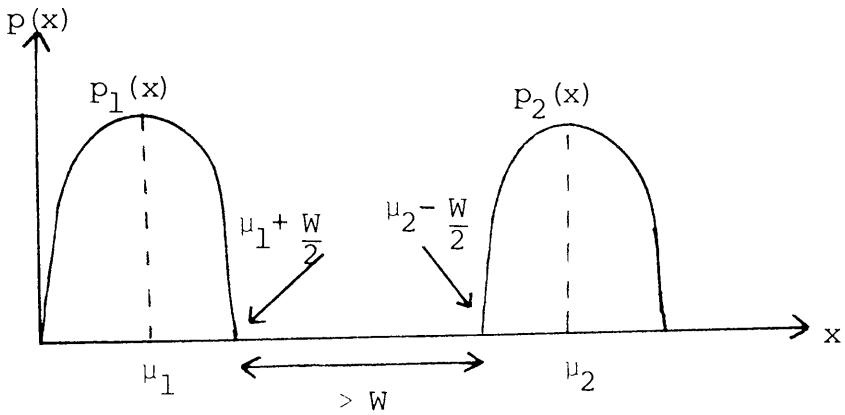


Figure 3.8 Non-overlapping local probability distributions

From equation 3.22, the local centroid at point x_0 for a window of size 2W is given by :

$$\mu(x_0) = x_0 + \frac{\int_{-W}^{W} x' \, p(x_0+x') \, dx'}{\int_{-W}^{W} p(x_0+x') \, dx'} \qquad 3.27$$

where :

$$|x_0 - \mu_i| < W \qquad \text{some } i \qquad 3.28$$

Defining a new variable x'' as :

$$x'' = x' - (\mu_i - x_0) \qquad 3.29$$

equation 3.25 becomes :

$$\mu(x_0) = x_0 + \frac{\int_{-W-\mu_i+x_0}^{W-\mu_i+x_0} (x'' + \mu_i - x_0)\, p(x'' + \mu_i)\, dx''}{\int_{-W-\mu_i+x_0}^{W-\mu_i+x_0} p(x'' + \mu_i)\, dx''}$$

$$= \mu_i + \frac{\int_{-W-\mu_i+x_0}^{W-\mu_i+x_0} x'' p(x'' + \mu_i)\, dx''}{\int_{-W-\mu_i+x_0}^{W-\mu_i+x_0} p(x'' + \mu_i)\, dx''}$$

$$= \mu_i \qquad 3.30$$

where the last step follows because the integral is zero by definition of the local means μ_i . Hence the local centroid at any point under one of the local probability distributions is just the local mean of that distribution. Thus by moving each probability mass of p(x) to its local centroid position a new distribution p'(x) is produced which is just :

$$p'(x) = \sum_{i=1}^{N} n_i\, \delta(x-\mu_i) \qquad 3.31$$

where n_i is the area under each local distribution $p_i(x)$. Thus a classification of p(x) results in which each point x is assigned to a class defined by the class mean of the local distribution which it is under.

In general, of course, the global probability distribution can not be written as a summation of a

set of non-overlapping local distributions. However, using the ideas expressed above an iterative scheme can be developed which results in a classification of the type given by equation 3.31. This scheme is described in the next section.

3.5.3.3 General Class Distributions - An Iterative Scheme

Let h(x) be the histogram value for position x in class space. The algorithm works by continually updating the histogram by moving probability masses to the positions of their local centroids until no change in the histogram is observed. Hence if $h^n(x)$ is the updated histogram on the nth iteration, the algorithm proceeds as follows :

$n = 1$

$h^0(x) = h(x)$

$h^n(x) = \sum_{y \in B(x)} h^{n-1}(y)$

where $y \in B(x)$ iff $\mu^n(y) = x$

where $\mu^n(y) = y + \dfrac{\sum_{y'=-m}^{m} y' \, h^{n-1}(y+y')}{\sum_{y'=-m}^{m} h^{n-1}(y+y')}$

$n=n+1$ $\xleftarrow{\text{no}}$ $h^n(x) = h^{n-1}(x)$ all x $\xrightarrow{\text{yes}}$ update window size

In this case the local centroids are computed within a

window of width 2m+1. As in equation 3.31 a final histogram of the form :

$$h^n(x) = \sum_{j=1}^{Nc} n_j \, \delta(x-x_j) \qquad 3.32$$

with :

$$\min_{i,j} |x_i - x_j| > 2m+1 \qquad 3.33$$

will obviously be unchanged by passage through any further iteration loops. Such a distribution will thus terminate the algorithm. In equation 3.32 N_c is the number of classes and the values x_j are approximately :

$$x_j \sim \frac{\sum_{x \varepsilon c_j} x \, h(x)}{\sum_{x \varepsilon c_j} h(x)} \qquad 3.34$$

where $x \varepsilon c_j$ iff x is classified into the jth class. Hence, as shown rigorously in the previous section for a specific probability distribution, in general the iterative procedure approximately preserves local class means.

While the convergence properties of the algorithm are hard to determine, except in simple cases, it has been observed in practice [54] that it always converges in a small number of iterations (typically 10-20). The number of classes on final convergence depends both on the window size and the 'peakiness' of the original histogram h(x). The dependence on the

window size was removed in the following way : the algorithm is run for a set of window sizes increasing by 5 points on each run and the result is accepted when successive runs produce consistent results. Using this method it has been found that successful classifications can be achieved for a pair of Gaussian distributions of differing means when the inter-region signal to noise ratio (equation 3.11) is 3 or more.

In order to classify the data on successful termination of the clustering algorithm, a lookup table is maintained which keeps a track on the movement of data throughout each iteration. Specifically, if x is a point in the original histogram h(x) then the final classification is given by :

$$x \rightarrow l^n(x)$$

where :

$$l^0(x) = x$$

and :

$$l^j(x) = l^{j-1}(y) \quad \text{for all y such that } x = \mu^j(y) \quad j=1,n$$

$l^p(x)$ is the lookup table at the start of the pth iteration representing the new position of the probability masses originally at position x in class space. Hence final classification amounts to indexing into a lookup table using each quadtree value q(i,j,k) at the highest quadtree level as an index.

Finally, any isolated nodes (those nodes all of whose 8 neighbours belong to a different class) are assigned to the class containing the majority of its 8 neighbours as such isolated classifications are unreliable.

3.5.4 Spatial Compactness as a Consistency Check

Following the clustering procedure a test is applied to the resulting classified data to check that the spatial properties of the classes are consistent with a priori assumptions of compactness. This assumption is implicit in the design of the segmentation scheme in which statistical classification follows quadtree smoothing - any classes which appear following smoothing with large averaging blocksizes must have a significant number of spatially contiguous points.

The test took the form of computing a coefficient d for each classified region where d is defined as :

$$d = \frac{\text{Number of points in the region}}{(\text{Squared deviation in the region})^{\frac{1}{2}}} \qquad 3.35$$

Clearly the larger the value of d, the more spatially compact the region. Only regions which have values of d above some suitably chosen threshold are accepted. All other classes have their points re-classified to

the nearest above threshold class and the test re-applied until a consistent result is obtained.

In order to determine a suitable value for the threshold, consider an ellipse with major and minor axes 2a and 2b respectively. Assuming unit inter-pixel separation and that a and b are sufficiently large compared to unity in which case the effects of the ellipse boundary can be ignored. The number of points inside this region is then πab. An infinitesimal point (x,y) of area dxdy contributes an amount $(x^2 + y^2)dxdy$ to the squared deviation. Hence the squared deviation for the whole ellipse is given by :

$$sd = \int\int_S x^2 \, dxdy + \int\int_S y^2 \, dxdy \qquad 3.36$$

where the integral is taken over the surface of the ellipse. The first integral can be written explicitly as :

$$I_x = \int_{-a}^{a} x^2 \, dx \int_{-b\sqrt{1-\frac{x^2}{a^2}}}^{b\sqrt{1-\frac{x^2}{a^2}}} dy$$

$$= 2b \int_{-a}^{a} x^2 \sqrt{1-\frac{x^2}{a^2}} \, dx \qquad 3.37$$

Using the substitution $x = a \sin\theta$ this integral can be readily evaluated to give :

$$I_x = \frac{a^3 b \pi}{4} \qquad 3.38$$

By symmetry :

$$I_y = \frac{b^3 a \pi}{4} \qquad 3.39$$

Substituting these results and the result for the number of points contained within the ellipse, the value of d in equation 3.35 is then :

$$d = \frac{2 a b \sqrt{\pi}}{(a^3 b + b^3 a)^{\frac{1}{2}}} \qquad 3.40$$

For a circle with a=b=r, $d = \sqrt{2\pi} = 2.51$, independent of r. For the more general case where a=kb equation 3.40 becomes :

$$d = 2 \sqrt{\pi} \sqrt{\frac{k}{1 + k^2}} = 2 \sqrt{\pi} \sqrt{\frac{1/k}{1 + (1/k)^2}} \qquad 3.41$$

which is independent of the absolute values of a and b. This expression has a maximum for k=1 of 2.51. For values of k of 2, 3, 4 and 5, d has values of 2.24, 1.96, 1.72 and 1.55 respectively. A threshold for d of 1.5 was chosen which therefore means that elongated regions are rejected. Such a restriction does not tend to degrade the algorithm's practical performance and the threshold deals successfully with spatially non-compact classified regions.

3.5.5 Boundary Estimation

It was mentioned in section 3.5.2 that quadtree smoothing is a means of trading off resolution in class space with spatial resolution. Hence following the clustering procedure at the highest quadtree level, each boundary node at this level defines an LxL block of pixels at the lowest quadtree level with $L=2^k$, k being the height of the quadtree. The problem now becomes how can full spatial resolution be restored?

A solution can only be found by making an additional assumption. This is that the classification introduced at the highest level of the quadtree is valid at lower levels. Thus a boundary region is defined; nodes not in the boundary region are given the same class as their father; nodes in the boundary region are classified in such a way that the boundary region width is reduced by a factor of 2 on each step down the quadtree. The result is a boundary between pixels at the lowest level of the tree and thus at full spatial resolution. As stated above, the key assumption made in obtaining the boundary is that positional information of each classified region is invariant to the scale over which it is viewed [45]. Hence constraints resulting from the previous level of processing can be introduced which control processing at the current level.

A more precise description of the boundary estimation procedure is as follows. Since it can be assumed that a successful classification has been made at the highest quadtree level, an inductive argument can be used. Hence assume that a classification is to be made at level k where a classification at level k+1 has already taken place. Define q(i,j,k) as the (i,j)th node at level k and c(q(i,j,k)) as the class of this node. Initially each node is given the same class as its father :

$$c(q(i,j,k)) = c(q(i/2,j/2,k+1) \qquad 3.42$$

From this classification the boundary region B(k) is defined as :

$$q(i,j,k) \in B(k) \quad \text{iff } c(q(i,j,k)) \neq c(q(i',j',k)) \qquad 3,43$$
$$(i',j') \in N_8(i,j)$$

where $N_8(i,j)$ is the 8-neighbour set of (i,j). Once B(k) is determined for all (i,j) it is augmented by the set $B_1(k)$ of nodes which have an 8-neighbour in B(k) :

$$q(i,j,k) \in B_1(k) \quad \text{iff } q(i',j',k) \in B(k) \quad (i',j') \in N_8(i,j)$$

$$B_c(k) = B(k) \cup B_1(k) \qquad 3.44$$

Hence a strip of nodes is defined which, for a straight line boundary at level k+1, is 4 nodes wide.

Region $B_c(k)$ is then smoothed with a linear filter whose spatial width depends on the estimated signal to noise ratio between regions r and s at level k, $\rho(r,s,k)$:

$$B_2(k) = B_c(k) * h(i,j,\rho) \qquad 3.45$$

where * denotes convolution and where ρ is given by :

$$\rho = \rho(r,s,k) = 2\,\frac{|\mu(r,k) - \mu(s,k)|}{\sigma(r,k) + \sigma(s,k)} \qquad 3.46$$

In this case $\mu(p,k)$ and $\sigma(p,k)$ are the estimated signal mean and standard deviation in region p at level k. $\mu(p,k)$ is found from the local centroid clustering algorithm and $\sigma(p,k)$ is given by :

$$\sigma^2(p,k) = \frac{1}{N(\bar{B}_c(p,k))} \sum_{(i,j)\varepsilon\bar{B}_c(p,k)} (q(i,j,k) - \mu(p,k))^2 \qquad 3.47$$

where $\bar{B}_c(p,k)$ is the set of non-boundary nodes at level k that have father nodes assigned to region p and N(A) is the number of points inside region A.

The filter $h(i,j,\rho)$ is formed by iterated convolution using a 3x3 filter :

$$h_1(i,j,\rho) = \lambda(\rho) \quad (i,j) = 0$$

$$= \frac{(1 - \lambda(\rho))}{8} \quad -1 \leq (i,j) \leq 1 \quad (i,j) \neq (0,0) \qquad 3.48$$

The function $\lambda(\rho)$ which governs the relative width of the resulting filter was selected after some experimentation to be a piecewise linear function of ρ and is shown in figure 3.9. Values of ρ which result in maximum smoothing and no smoothing were chosen to be 2 and 8 respectively.

After smoothing, a firm classification is made on all nodes in the boundary region using a 'nearest class mean' criterion. After removal of all resulting isolated nodes, the process is repeated for level k-1 until full spatial resolution is restored.

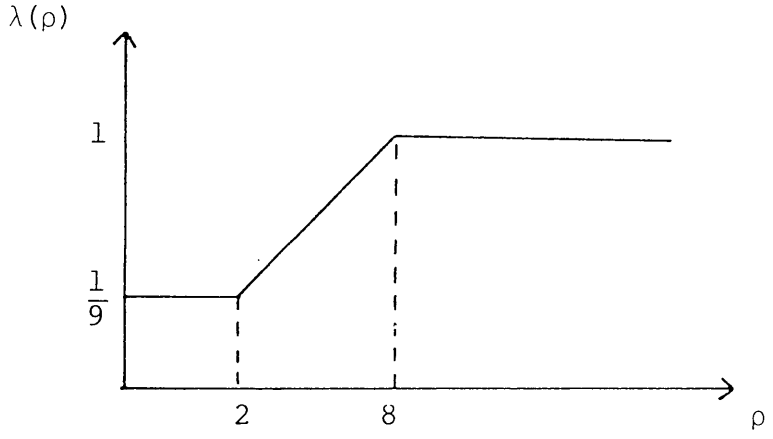


Figure 3.9 Piecewise linear function $\lambda(\rho)$

3.6 Discussion

The algorithm described above is an attempt to solve the segmentation problem in a non-parametric way which both avoids the problem of a priori knowledge and overcomes the weaknesses of purely statistical methods.

Specifically, information in class space is obtained by using a non-parametric clustering algorithm which requires no a priori information as to the number of classes present. It does so, however, at the expense of spatial resolution in accordance with the uncertainty principle [40]. Spatial resolution is restored by making the assumption that the spatial properties of the regions are invariant over the scales of resolution defined by the quadtree [45] [55]. Further, because of this assumption, simple smoothing and thresholding can be used to make firm classifications at each level. The noise adaptive nature of the filters are designed to remove the bias in classification error at zero noise and to prevent spurious features being introduced into the boundary estimate at high noise. As will be seen in the next chapter, such an approach is consistent with perceived results.

Chapter 4

Image Segmentation Results

4.1 Introduction

This chapter describes a set of experiments designed to test the performance of the quadtree segmentation scheme. The experiments were of both a quantitative and qualitative nature. The probability of misclassification for a range of inter-region signal to noise ratios was measured and the performance of the scheme was observed for a variety of region geometries. In all of the experiments described in this chapter, the images are 256x256x8 bits and the segmentations are based on the grey level only.

4.2 Comparison with Bayesian Classification

In order to compare the performance of the new method with a simple statistical segmentation based on a Bayes criterion for minimum error [29], 8 test images were synthesised, each consisting of a pair of white Gaussian noise fields separated by an irregularly shaped boundary. (The boundary was generated using a 1st order Markov process.) In each case, the boundary separating the noise fields was identical and its position noted. The noise fields had identical means

of 135 and 145 and variances of 6.25, 11.11, 25, 100, 277.8, 400, 625 and 1111.1. These values correspond to inter-region signal to noise ratios ρ of 4, 3, 2, 1, .6, .5, .4 and .3 respectively.

Both the Bayes test for minimum error and quad-tree segmentation were run on each of the test images. The probability of misclassification is given by :

$$p(e) = P(1)\, e(1) + P(2)\, e(2) \qquad 4.1$$

where e(1) and e(2) are the probabilities of misclassifying class 1 as class 2 and class 2 as class 1 respectively. P(1) and P(2) are the a priori class membership probabilities which depend on the boundary shape. If there are a total of N pixels in the test image of which N_1 belong to class 1 and N_2 belong to class 2 and n_1 class 1 pixels and n_2 class 2 pixels are misclassified then :

$$e(1) = n_1 / N_1$$

$$e(2) = n_2 / N_2$$

$$P(1) = N_1 / N$$

$$P(2) = N_2 / N \qquad 4.2$$

Hence the probability of error can be found by measuring n_1 , n_2 , N_1 and N_2 and substituting the calcu-

lated values of e(1), e(2), P(1) and P(2) into equation 4.1.

4.2.1 Bayesian Classification of Gaussian Test Images

A simple Bayesian classification was performed on each test image. As mentioned in section 3.2, this amounts to thresholding each point in the image where pixels with grey levels above the threshold are assigned to one class and those with grey levels below the threshold are assigned to the other class.

If $p(x|i)$ is the probability density of the image grey levels conditioned on it belonging to class i for i=1 or 2 then the Bayes criterion for minimum error states that the class chosen is the one with the maximum value of $p(x|i)P(i)$. The expression for $p(x|i)$ is given by :

$$p(x|i) = \frac{1}{(\sigma(i)\sqrt{2\pi})} \exp\left(-(x - m(i))^2 / 2\sigma(i)^2\right) \qquad 4.3$$

where $m(i)$ and $\sigma(i)$ are the mean and standard deviation of class i. Figure 4.1 shows the density $P(i)p(x|i)$ for 2 classes with means $m(1)$ and $m(2)$ where the shaded area indicates the misclassification error which occurs when x_0 is used as the threshold.

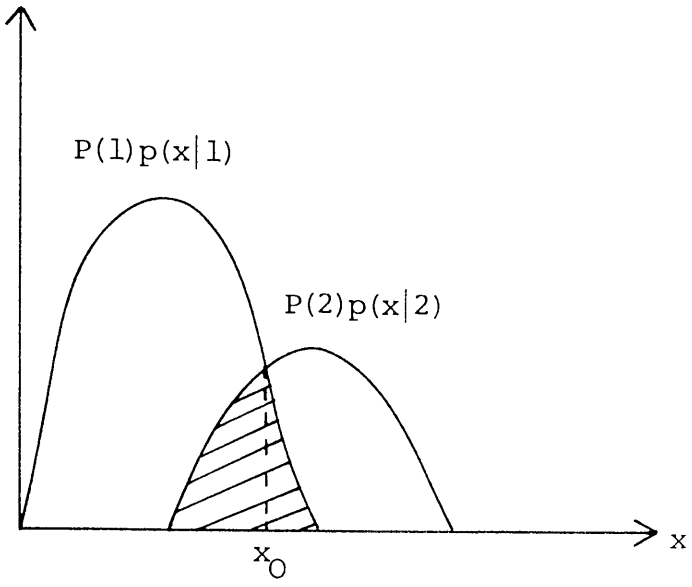


Figure 4.1 P(i)p(x|i) for 2 classes

The probability of error in assigning a particular class to the random variable x is just :

$$p(e) = P(1)\ e(1) + P(2)\ e(2) \qquad 4.4$$

where e(1) and e(2) are the areas of the shaded region to the right and left of the threshold x_O respectively. Indeed equation 4.4 corresponds exactly to equation 4.1. The values of e(1) and e(2) can be calculated using the expression for the conditional probability distribution given in equation 4.3. In order to do this an expression for x_O, the intersection of the two distributions must be found. Clearly x_O is a solution to :

$$\frac{P(1)}{\sigma(1)} \exp\left[-\frac{1}{2\sigma(1)^2}(x_0 - m(1))^2\right] = \frac{P(2)}{\sigma(2)} \exp\left[-\frac{1}{2\sigma(2)^2}(x_0 - m(2))^2\right] \quad 4.5$$

which is a quadratic in x_0 whose two solutions, x_{01} and x_{02} can easily be found. Expressing e(1) and e(2) in equation 4.4 in terms of the distributions p(x|1) and p(x|2) gives :

$$p(e) = P(1) \int_{R(2)} p(x|1)\, dx + P(2) \int_{R(1)} p(x|2)\, dx \quad 4.6$$

where R(i) is defined as the region of integration along the x axis such that P(i)p(x|i) is maximum. Defining :

$$\text{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp(-y^2/2)\, dy \quad 4.7$$

and :

$$\text{erfc}(x) = 1 - \text{erf}(x) \quad 4.8$$

and substituting the expressions for the conditional probability distribution into equation 4.6, it can be shown that this equation reduces to :

$$p(e) = P(1) \sum_{i=1}^{2} \text{erfc}\left[(x_{oi} - m(1)) / \sigma(1)\right] + P(2) \sum_{i=1}^{2} \text{erfc}\left[(x_{oi} - m(2)) / \sigma(2)\right] \quad 4.9$$

Values of erfc(x) are tabulated and so the probability of error can be evaluated. Also an actual classifica-

tion was carried out using the Bayesian criteria which requires a priori knowledge of P(1), P(2), m(1), m(2), $\sigma(1)$ and $\sigma(2)$ in order to compute the thresholds x_{01} and x_{02}. The theoretical and actual values of p(e) are plotted in figures 4.2 and 4.3 for a range of signal to noise ratios where it can be seen that they are in close agreement. Figures 4.4 and 4.5 show Bayesian classifications for values of ρ of 0.4 and 1.0 respectively in which considerable classification error is apparent.

In practice the values of P(i), m(i) and $\sigma(i)$ for the two class distributions would be unknown and hence the conventional Bayesian approach would not be possible. However, a modified procedure was developed in which the noise fields are classified after 4 quadtree smoothing operations as described in section 3.5.3. This amounts to classifying each quadtree node outside a boundary region consisting of a $2^L \times 2^L$ block of pixels at the lowest level for every node at the highest level, L being the height of the quadtree. In this case the values of P(i), m(i) and $\sigma(i)$ can be estimated from the classified nodes and a Bayesian classification carried out on just the boundary region pixels. Theoretical and actual classification error probabilities are plotted in figures 4.2 and 4.3 where the theoretical probability of error is just a factor B/N times the probability of error given in equation

4.4, B being the number of boundary nodes. From figures 4.2 and 4.3 it can be seen that there is close agreement between theoretical and actual probabilities of error. Figures 4.6 and 4.7 show classifications using this method for ρ=0.6 and ρ=3.0 respectively.

4.2.2 Quadtree Segmentation Performance on Gaussian Test Images

The quadtree segmentation scheme was run on the set of test images. In this case clustering was applied after 4 quadtree smoothing operations, that is to say on a 16x16 block of nodes (the effects of averaging blocksize are studied in the next section.)

Figures 4.8 and 4.9 show a pair of segmentations corresponding to values of ρ of 0.5 and 2.0. In these figures the original image and the original image with the superimposed region boundary are shown. The boundary obtained from the segmentation corresponds very closely to the perceived region boundary of the original.

In figure 4.2 the probability of error is plotted as calculated from the measured number of misclassifications. Comparing the plots of figure 4.2, the quadtree segmentation scheme is superior to conventional Bayesian classification even though, in the case of the unmodified Bayesian approach, a priori knowledge

of class means and variances is required.

4.3 Effects of Averaging Blocksize

As mentioned above, a quadtree consisting of 5 levels was used to perform the segmentation. This represents an averaging blocksize of 16x16, that is to say each node at the highest quadtree level is the average of a 16x16 block of nodes at the lowest level.

In order to gain insight into the effect of using different averaging blocksizes, a segmentation of each test image was carried out using blocksizes of 8x8, 4x4 and 2x2 corresponding to quadtree heights of 4, 3 and 2 respectively. As before the probability of error was measured, a graph of which is shown in figure 4.10 for each blocksize. Plotted on the same graph is the probability of error when the original 16x16 blocksize is used.

From figure 4.10 it appears that there is a minimum signal to noise ratio at which the classification can proceed, the value of which is determined by the size of averaging blocksize used. This is a result of the inability of the clustering algorithm to resolve the two classes there being insufficient smoothing gain. The results indicate a close agreement with theoretical expectations in that the signal to

noise ratio at which clustering becomes no longer possible approximately halves on doubling the blocksize and hence doubling the smoothing gain. However, once the clustering successfully classifies the data, there is close agreement between the error probabilities for each of the averaging operator sizes.

The choice of quadtree height should be such that it provides sufficient smoothing gain to resolve all of the classes in the image. Clustering at a quadtree level greater than that necessary to resolve the classes does not introduce any significant error into the boundary estimation procedure. However, the greater the height of the quadtree, the larger the region size must be in order to successfully classify it. This is a result of the bias introduced due to averaging across region boundaries which has a larger proportionate effect the smaller the region size. The 16x16 blocksize chosen for the experiments described in this chapter provides an acceptable compromise, both producing sufficient smoothing gain for most cases and being able to resolve small regions.

4.4 Effects of Segmentation Level.

Instead of terminating the boundary propagation at the lowest quadtree level, segmentation was terminated at each of the 4 levels above the base. The

children at the lowest level of each classified node at the termination level were assigned to the same class as their father and the classification error measured in the usual way. Figure 4.11 plots error probability against the segmentation termination level for values of the inter-region signal to noise ratio of 2 and 4. From the graph it can be seen that, for the lower signal to noise value, there is only a marginal advantage to be gained from terminating the segmentation at the lowest level as opposed to the level above.

4.5 A Simple Model Predicting Algorithm Performance

In general, analysing the performance of the segmentation scheme, that is to say developing a model that predicts the probability of classification error as a function of signal to noise ratio, is a very difficult task. The reason is that the performance depends upon the boundary shape and the downward propagation of errors. However, using certain simplifying assumptions, an approximate analysis can be undertaken [52].

Consider a straight line separating a pair of regions of white Gaussian noise of means m(1) and m(2) each with a variance of σ^2 . Assume that the classification error at each quadtree level is due entirely to

nodes on or adjacent to the boundary since, for such nodes, the support of the smoothing filter centred on those nodes includes on average 50% of nodes belonging to a different class. Also an error in boundary placement at the bottom level of the quadtree certainly causes a classification error although this does not apply to errors at higher levels due to the width of the boundary region. Hence an 'error propagation' probability must be introduced as a free parameter which specifies the probability of an error occurring at a level k>0 propagating down to level zero and thus producing a classification error. The value of the free parameter was chosen to give the best fit to the experimental results.

Let the probability of error at level k be P(k), the error propagation parameter be p and the cumulative error at level k be e(k). The value of e(k) represents the error at level 0 that has accrued from classification errors up to and including the current level k. A node at level k has 2 sons (in the case of a 1-d boundary) and is either misclassified with probability P(k), in which case the error propagates down to level zero with probability p, or classified correctly, in which case it adds nothing to the cumulative error. Hence e(k) can be written as :

$$e(k) = p\,2^{k}\,P(k) + (1 - P(k))\,e(k-1) \qquad k > 0$$
$$e(0) = P(0) \qquad 4.10$$

The probability of error P(k) depends on the inter-region signal to noise ratio at level k and the smoothing gain due to the filter $h(i,j,\rho)$ (cf equation 3.49). From the specification of the filter given in equation 3.52, for a node on the boundary, the smoothing gain is 1 for $\rho > 8$, 0 for $\rho <= 2$ and linearly varying in between where the smoothing gain s is :

$$s = \frac{\rho_k'}{\rho_k} \qquad 4.11$$

ρ_k' being the value of the inter-region signal to noise ratio of the boundary nodes following application of the smoothing filter $h(i,j,\rho)$. From equation 4.9 with $\sigma(1) = \sigma(2) = \sigma$, $x_{O1} = (m(1)+m(2))/2$ and $x_{O2} = 0$ and assuming equally probable classes, P(k) becomes :

$$P(k) = \mathrm{erfc}\,(\rho_k\, s / 2) \qquad 4.12$$

where :

$$s\rho_k = 0 \qquad \rho_k \leq 2$$
$$= \rho_k \qquad \rho_k > 8$$
$$= \frac{(\rho_k - 2)}{6}\,\rho_k \qquad 2 < \rho_k \leq 8 \qquad 4.13$$

The results of this model are plotted in figure 4.2 for each of the original test images with a value of p of 0.6. It can be seen from a comparison with figure 4.3 that the theoretical results show a satisfactory fit to the experimental results bearing in mind the crudeness of the approximations.

If the segmentation is terminated at a level k'and the boundary propagated down to the bottom level by assigning all nodes at lower levels to the class of their father node, the cumulative error becomes :

$$e(k,k') = \sigma_{k'}(x) \qquad k \leq k'$$

$$= p\, 2^k P(k) + (1 - P(k))\, e(k-1,k') \qquad k>k' \qquad 4.14$$

The second equality of equation 4.14 follows from the arguments given above leading to equation 4.10. $\sigma_{k'}(x)$ is the error resulting from simply assigning all nodes at level 0 to the classes of their father nodes at level k' in which case the boundary position is uniformly distributed over a span from $-2^{k'-1}$ to $2^{k'-1}$ nodes. Therefore :

$$\sigma_k^2(x) = (2^k + 1)^{-1} \sum_{j=-2^{k-1}}^{2^{k-1}} j^2$$

$$= \sqrt{\frac{2^{k-1}(2^{k-1} + 1)}{3}} \qquad 4.15$$

Using equations 4.10 and 4.15 and the expression for P(k) in equation 4.12, the theoretical values of the cumulative error is plotted in figure 4.11 for inter-region signal to noise ratios of 2 and 4. Again the theoretical results provide a satisfactory fit to the experimental results.

4.6 General Grey Level Segmentation Results

A series of grey level segmentations were carried out on a variety of region structures both for white Gaussian noise fields, for natural and synthetic textures and for natural images. Unless otherwise stated 5 quadtree levels were used corresponding to a maximum averaging blocksize of 16x16.

Figures 4.12a-d, figure 4.13 and figure 4.14 show a series of object/background segmentations. Figure 4.12a shows 3 irregularly shaped objects, each with a uniform grey level and a background of uniform grey level. Zero mean white Gaussian noise was then added to these images leading to figure 4.12b where the inter-region signal to noise for each of the 3 objects is 3, 1.5 and .75 starting at the top left and moving clockwise. Figure 4.12c is the result of running the segmentation on 4.12b and figure 4.12d is the result of superimposing the boundary of figure 4.12c onto figure 4.12b. This example illustrates the effect of

the noise adaptive smoothing filters that are used in the boundary estimation procedure. The object boundaries obtained bear a close resemblance to the perceived boundaries where in the case of the region closest in grey level to the background, a considerable degree of local smoothing has taken place. Figure 4.13 and 4.14 show pairs of segmentations where the objects are circles placed close together in background noise. In these two cases the boundaries of the segmented regions are shown superimposed on the originals. Again there is close agreement between the actual region boundaries and the boundaries obtained by the segmentation scheme although there is considerable noise, the object/background signal to noise ratio being 1 in both figures 4.13 and 4.14 for each object.

Figures 4.15 and 4.16 show grey level texture segmentation results and demonstrate the non-parametric nature of the scheme. Figure 4.15 consists of a pair of synthetic textures both generated by low-pass filtering a random impulse noise field as described in chapter 2. Figure 4.16 consists of the 3 natural Brodatz textures cork, grass and paper where each one has a different mean grey level. Again the segmentation result is shown by superimposing the region boundaries on the original indicating close agreement between the boundary obtained and the

perceived boundary.

Finally figures 4.17 and 4.18 show the result of running the segmentation on natural images, in the first case a girl's face and in the second a landscape scene. In these two cases the clustering algorithm was run on the image resulting from the application of 3 quadtree smoothing steps corresponding to an averaging blocksize of 8x8 in order to resolve smaller regions. The segmented regions are shown as constant grey level regions, the grey level corresponding approximately to the local mean of the class as determined by the clustering algorithm. Although results of this nature are difficult to interpret, in both cases there is good agreement with regions obtained from the segmentation and actual regions of uniform grey level in the image.

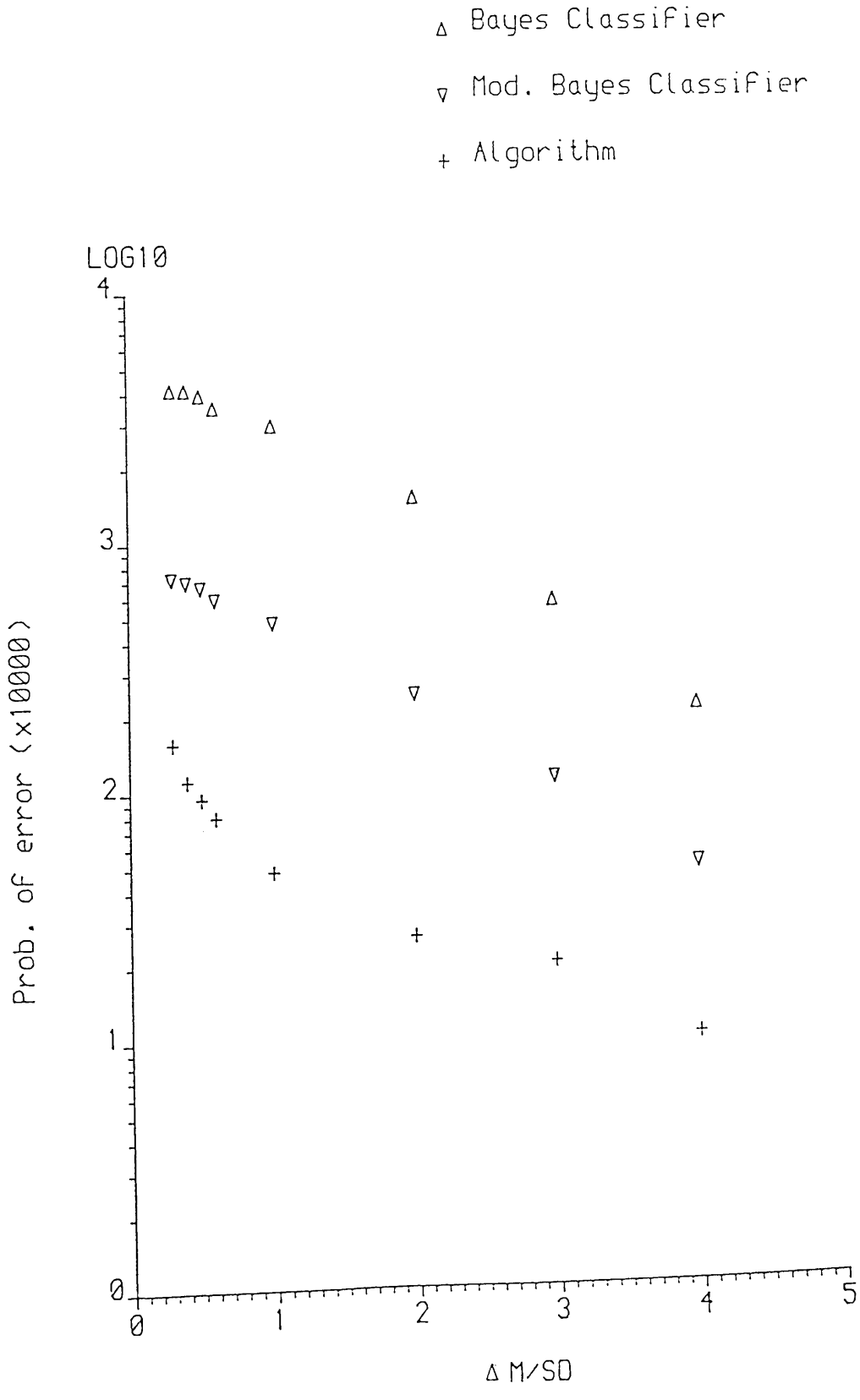


Figure 4.2 Theoretical probability of error v S/N for Gaussian test images.

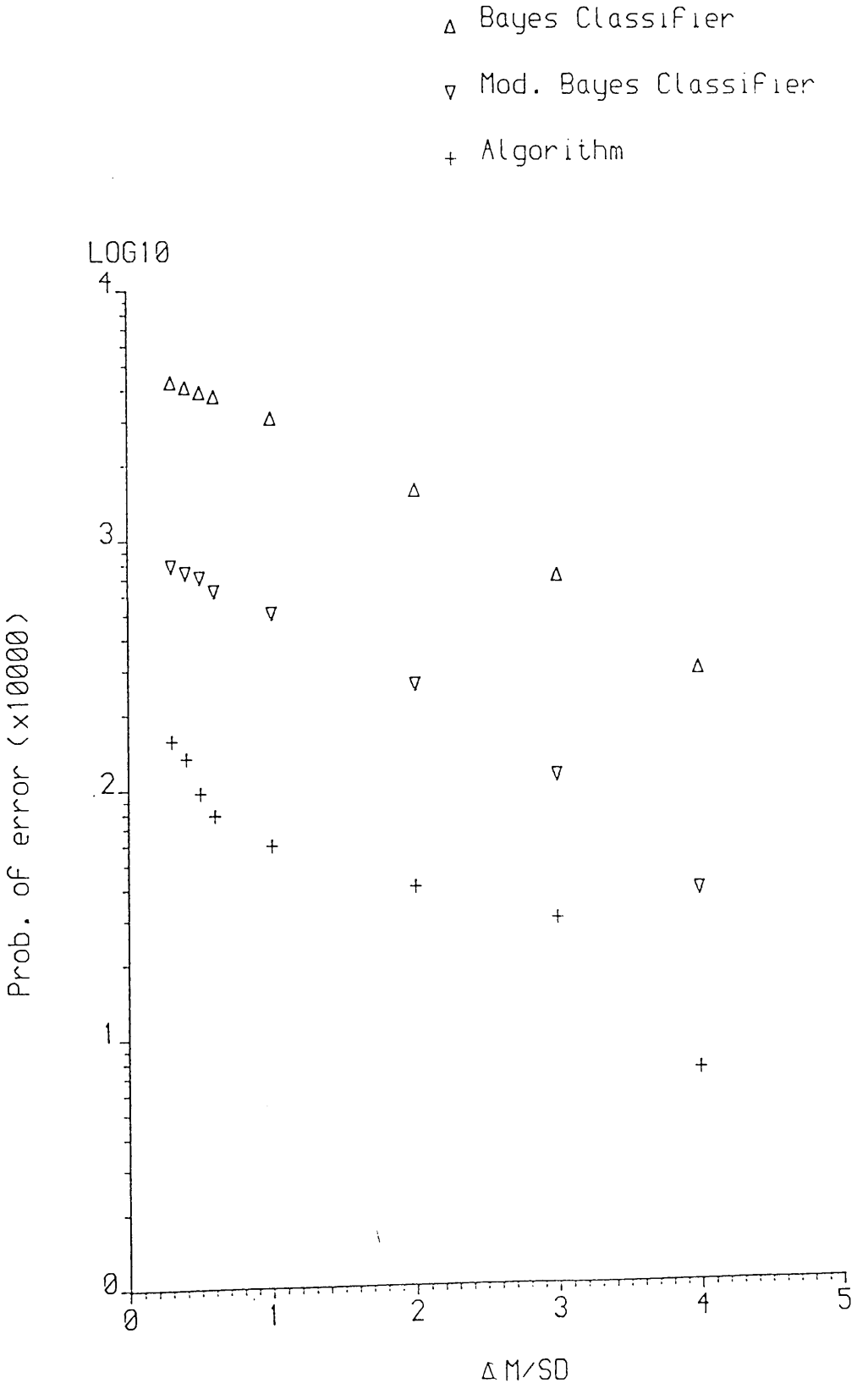


Figure 4.7 Estimated probability of error v S/N for Gaussian test images.

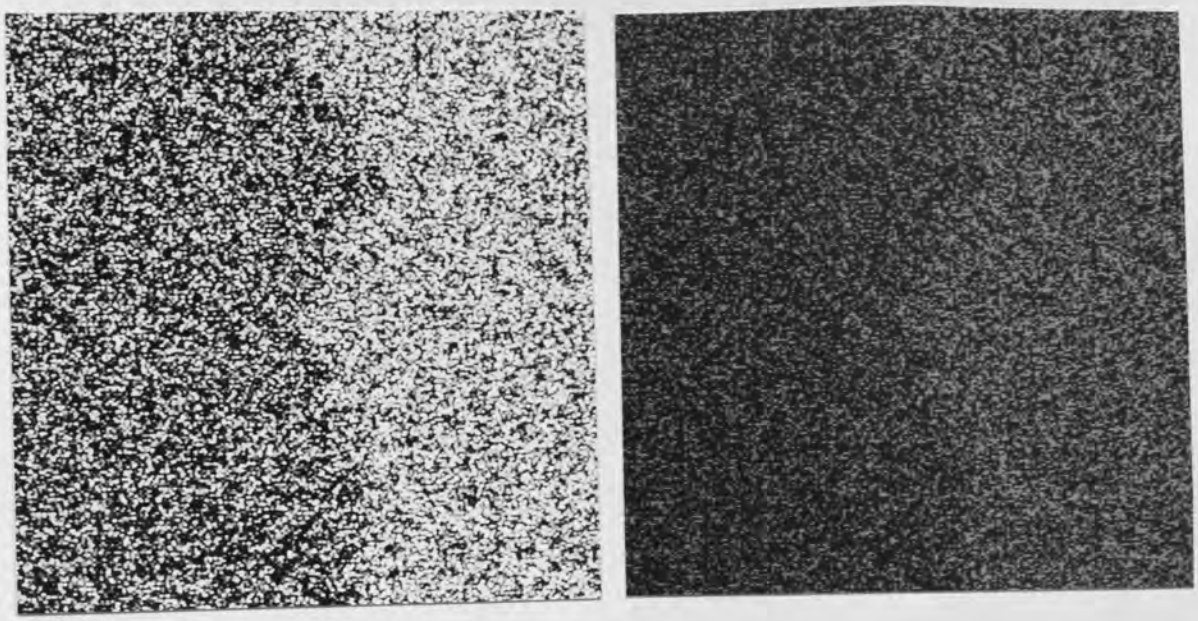


Figure 4.4 Bayesian classification of a pair of white Gaussian noise fields. Inter-region S/N = 0.4.

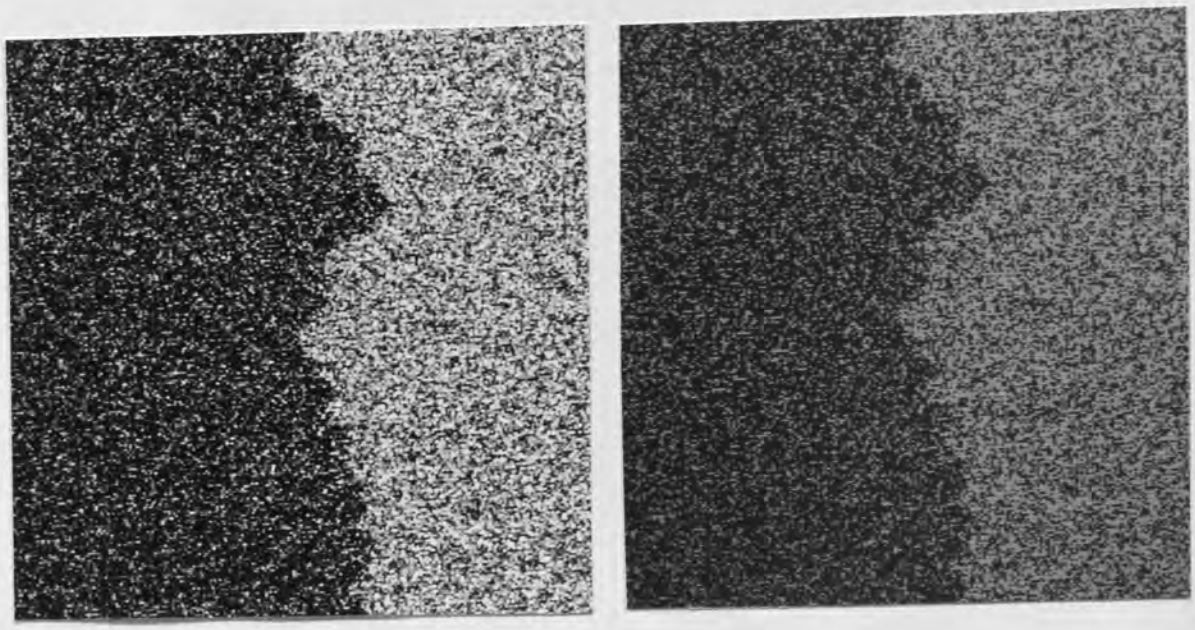


Figure 4.5 Bayesian classification of a pair of white Gaussian noise fields. Inter-region S/N = 1.

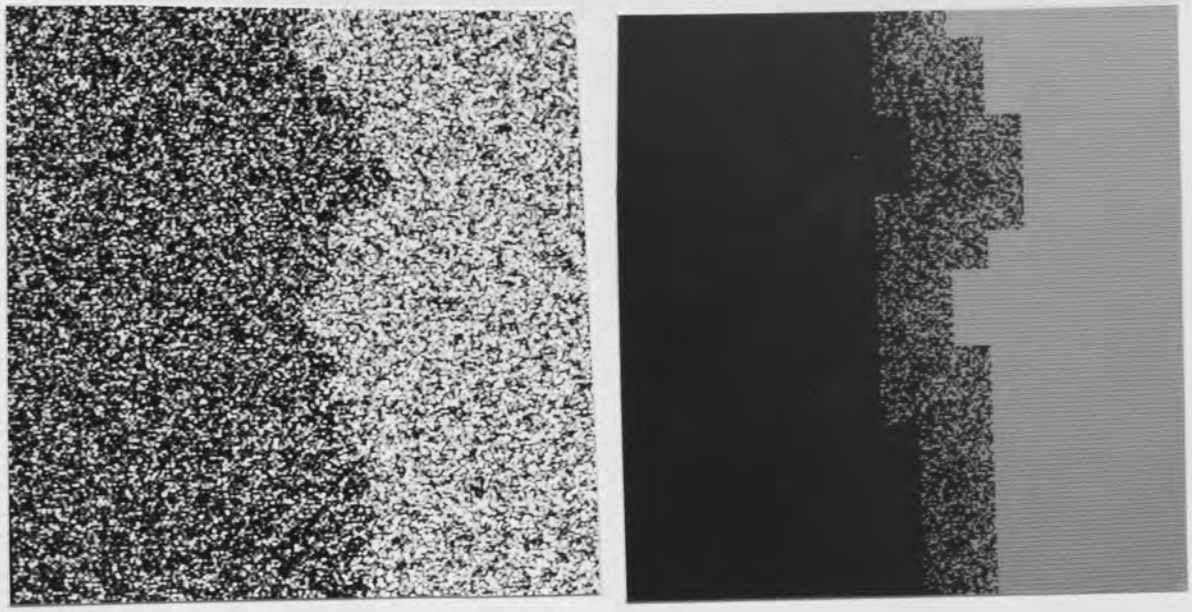


Figure 4.6 Bayesian classification of the boundary region of a pair of white Gaussian noise fields. Inter-region S/N = 0.6.

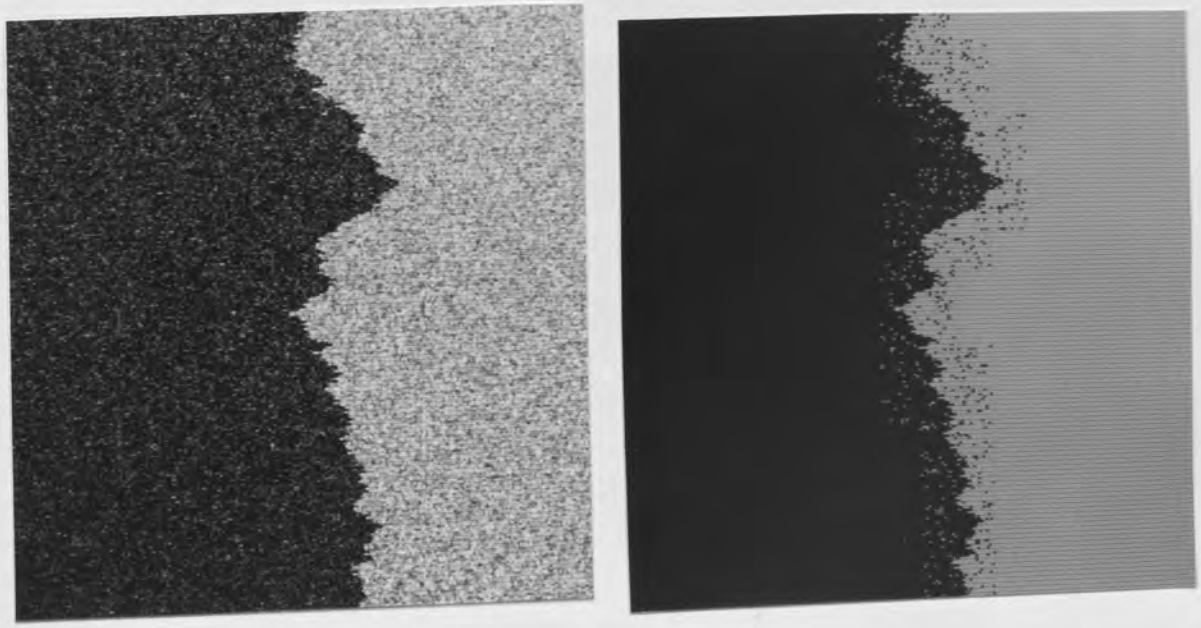


Figure 4.7 Bayesian classification of the boundary region of a pair of white Gaussian noise fields. Inter-region S/N = 3.

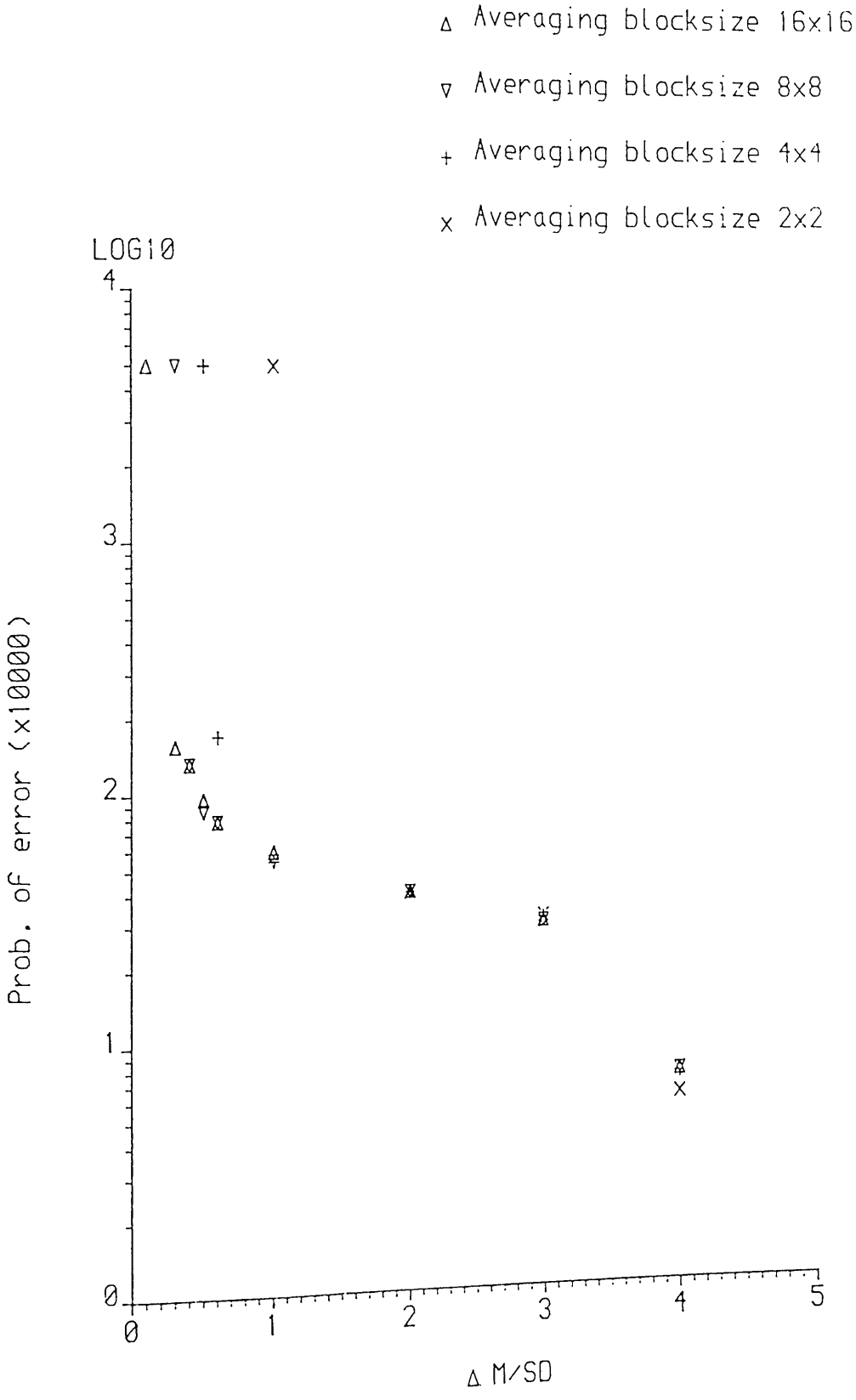


Figure 4.10 Effect of averaging blocksize for Gaussian test images.

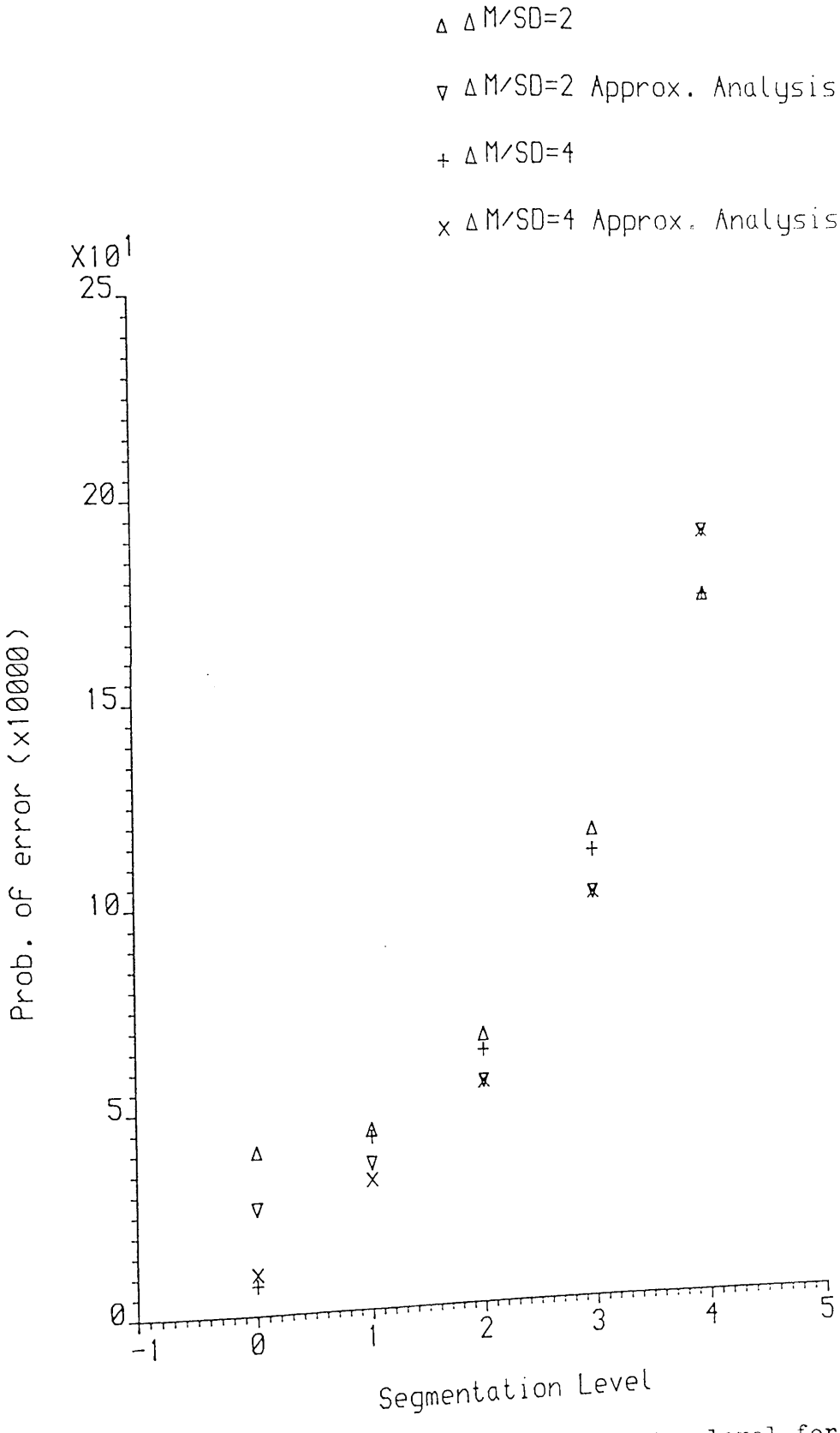
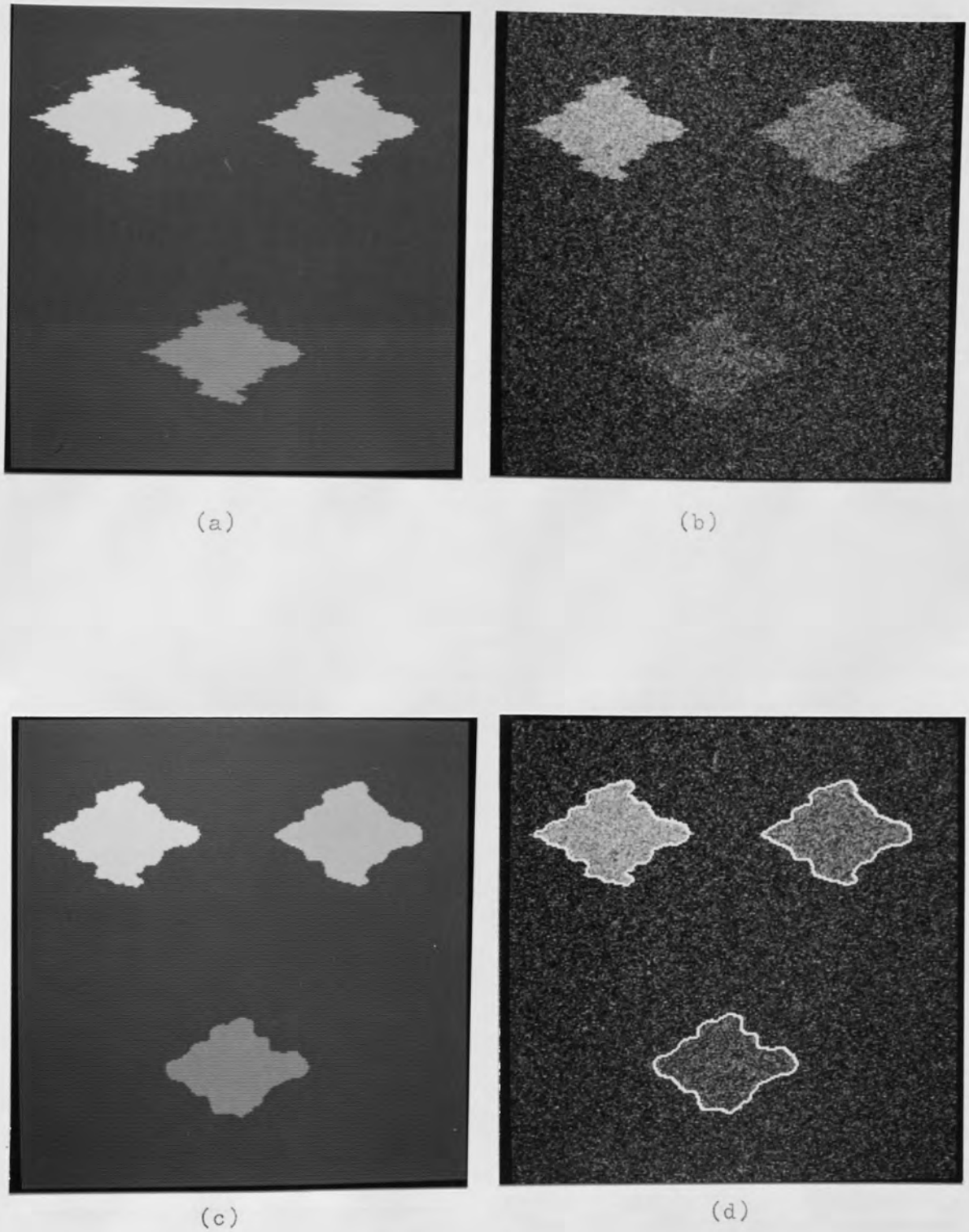


Figure 4.11 Effect of segmentation termination level for Gaussian test images.



(a) (b)

(c) (d)

Figure 4.12 Irregular object/background segmentation.

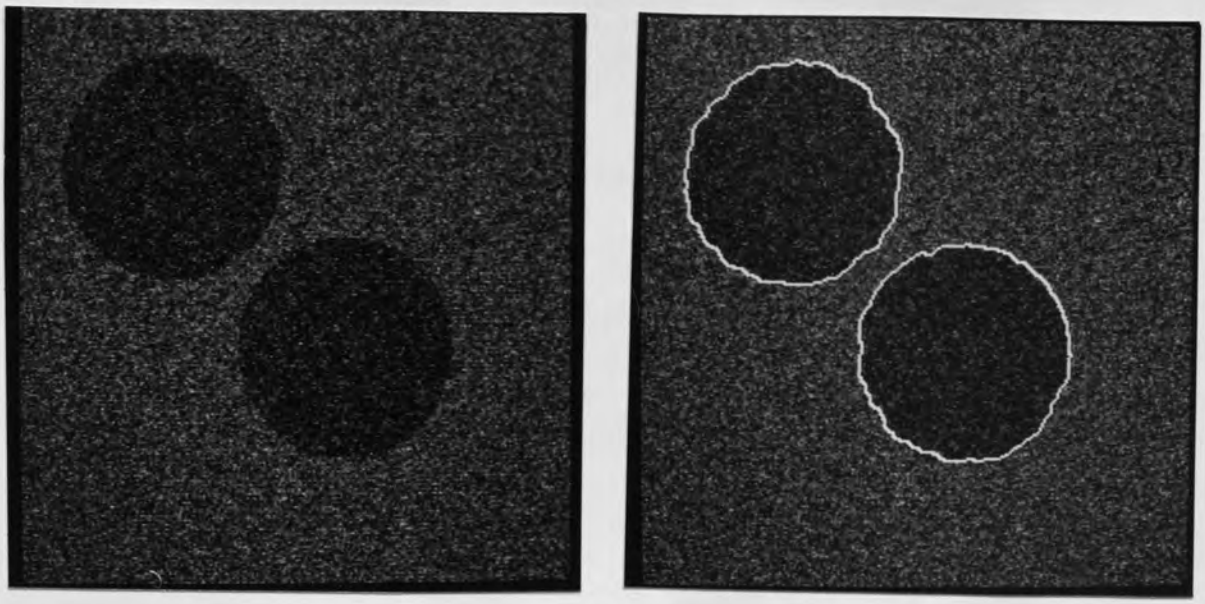


Figure 4.13 Object/background segmentation.

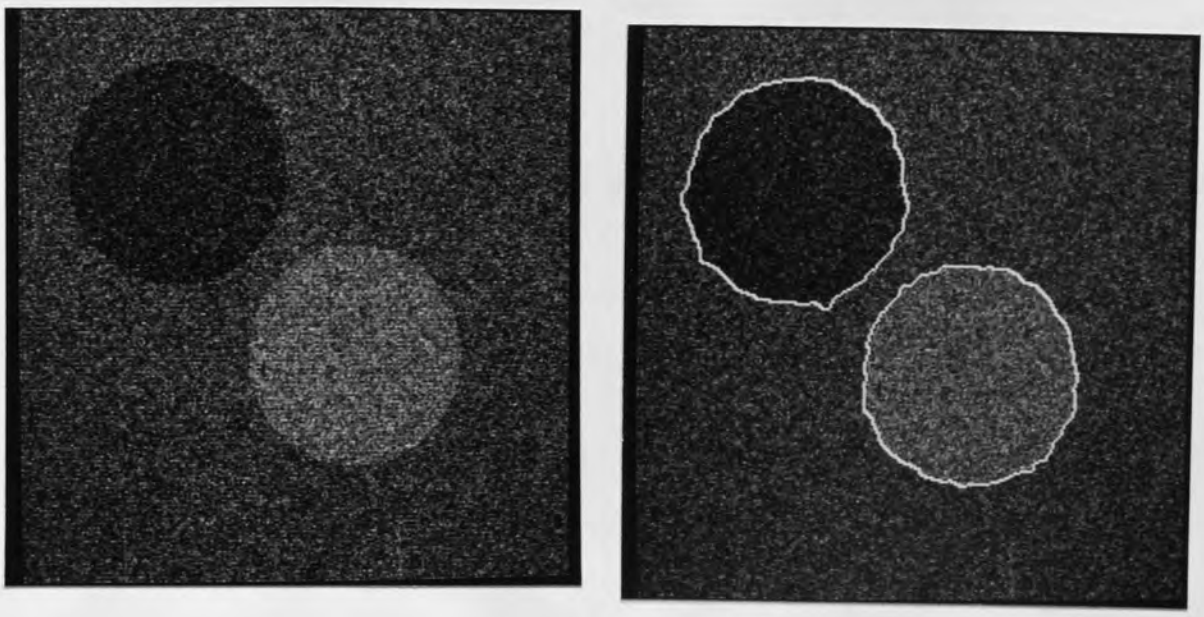


Figure 4.14 3 class object/background segmentation.

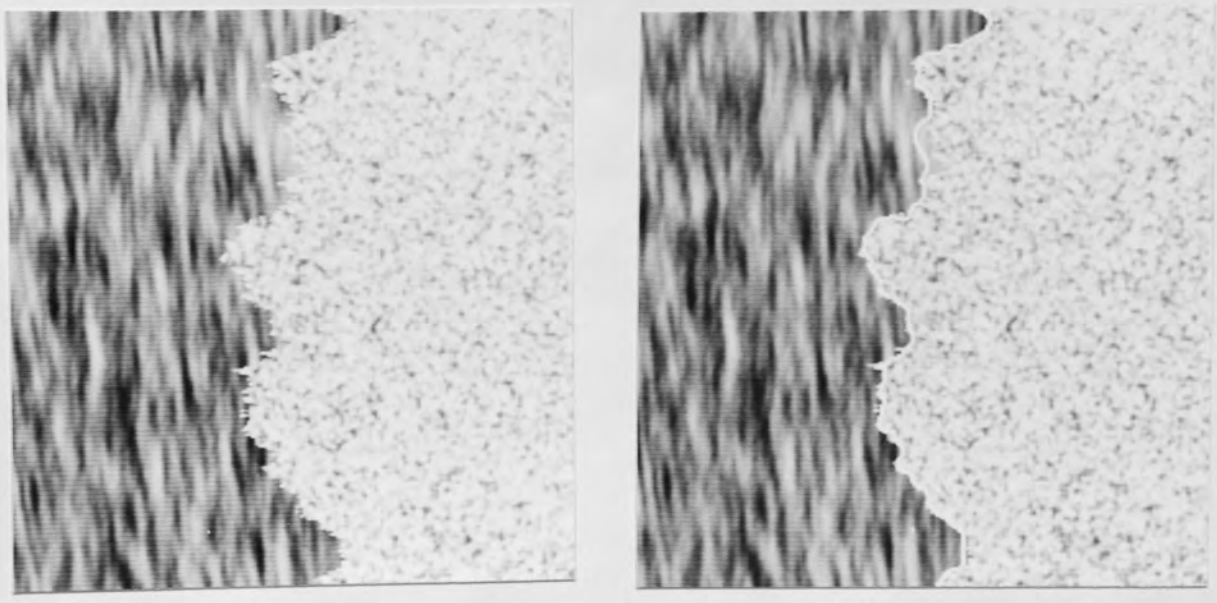


Figure 4.15 Grey level segmentation of synthetic textures.

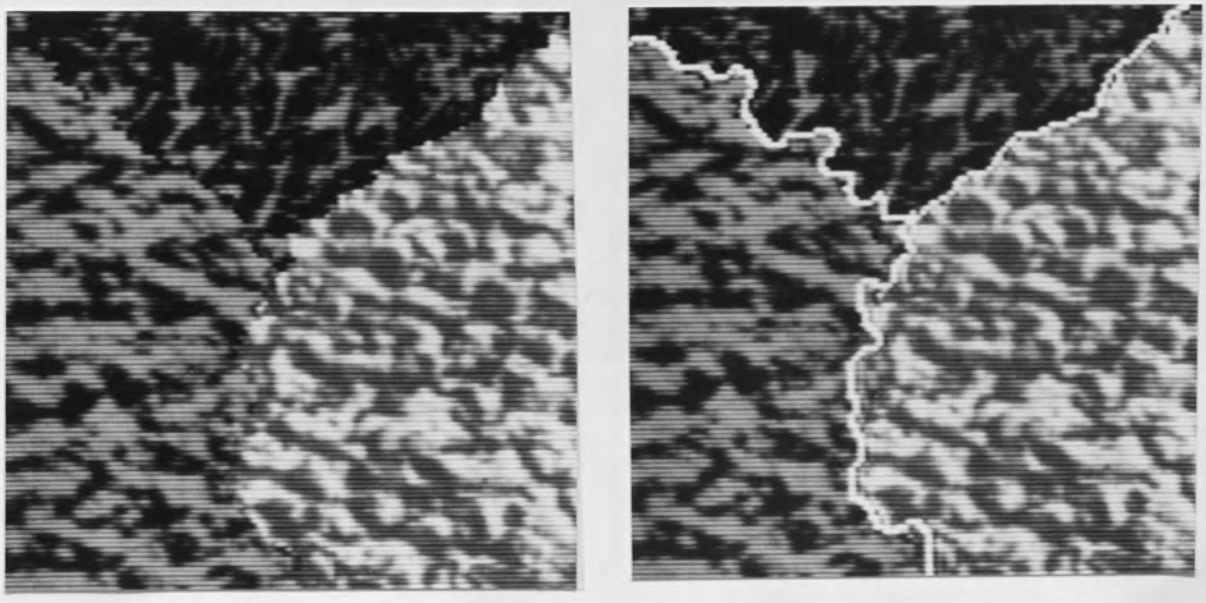


Figure 4.16 Grey level segmentation of the natural textures paper, grass and cork.

Figure 4.17 Grey level segmentation of a face.

Figure 4.18 Grey level segmentation of a landscape.

Chapter 5

Texture Segmentation

5.1 Introduction

The previous chapters have considered a classifier which is able to deal with generally noisy signals, that is to say signals in which there is considerable statistical fluctuation about the means within each class. As will be seen, this will be useful in the segmentation of texture because of the inherent randomness which characterises texture. This chapter describes appropriate modifications which need to be made to the quadtree segmentation scheme in order for it to be able to deal with multi-dimensional signals.

Before this can be done however, the problem of feature extraction must be considered. Feature extraction can be defined as operations that are performed on the original signal which result in a more effective set of features for discrimination purposes. In the context of feature extraction for texture segmentation this entails going from a signal space which is one dimensional, simply the grey level values at each image point, to a multi-dimensional signal space in which a vector quantity is defined at each image point.

Choosing a sufficient set of features which enable the partioning of an image into regions of uniform texture is related to the ideas discussed in chapter 1 concerning texture perception and discrimination. The early ideas of Julesz [4] were mentioned which explain differences in perceived texture in terms of differences in second order statistics. Using this kind of method, in which a statistical model is fitted to the data, to solve texture segmentation problems is generally inappropriate since such models are not of a sufficiently local nature. Specifically, the problem is not only to identify the different texture classes in the image, but also to find the boundary regions between them as accurately as possible. The restrictions implied by the uncertainty principle [40] have a bearing on this matter because increased certainty of class membership is at the expense of spatial resolution.

Traditionally techniques have concentrated on obtaining the maximum certainty either spatially or in class space. For example, Therrien [56] treated the texture segmentation problem as a statistical estimation problem for a set of connected regions using large filters to reduce the classification error with a consequent loss of resolution at the region boundaries. Alternatively, Mitiche et al [57] described an edge detection approach to the problem in which class

resolution is sacrificed for an accurate representation of the boundary between disjoint regions.

The feature extraction procedure adopted consists of using a set of orthogonal filters (ie. those with spatial frequency responses occupying disjoint regions of the Fourier domain) with optimal simultaneous concentration of signal energy in space and spatial frequency. Implicitly assumed is that the textures can be discriminated on the basis of their spectra. In this context, class resolution is determined by the bandwidth of the filters, a smaller bandwidth and hence a larger impulse response giving higher class resolution and lower spatial resolution. Thus using a filter design procedure which maximises the mutual energy concentration in space and spatial frequency of the filter responses leads to optimum class and spatial resolution in this case.

5.2 Orthogonal Feature Sets

As mentioned above, textural features are used in the classification process which are based on the outputs of orthogonal filters. Hence if $h_i(n)$ is the impulse response of the ith filter with frequency response $H_i(u)$ then :

$$\sum_u H_i(u)\, H_j^*(u) = \sum_n h_i(n)\, h_j^*(n) = \delta_{ij} \qquad 5.1$$

where δ_{ij} is the Kronecker delta. The first equality in equation 5.1 follows because the DFT preserves scalar product and the second because of the definition of orthogonality. Also it is assumed that the filters are normalised to unit energy.

Orthogonality of stationary signals implies that the signals are uncorrelated. To prove this, let $\underline{x}=\{ x_0 \cdot\cdot x_{N-1} \}$ and $\underline{y}=\{ y_0 \cdot\cdot y_{N-1} \}$ be signal vectors corresponding to zero mean N-point stationary signals. The circular cross correlation vector $\underline{r}$ between $\underline{x}$ and $\underline{y}$ is such that :

$$r_i = \underline{x}^{*T}\, S(i)\, \underline{y} \qquad 5.2$$

with S(i) the ith circular shift operator. Defining F and F^* as the forward and inverse DFT operator :

$$F\, S(i) = W(i)\, F \cdot \qquad 5.3$$

where W(i) is a diagonal matrix such that :

$$[W(i)]_{kl} = \exp\left(\frac{2\pi j}{N}\, ik\right) \delta_{kl} \qquad 5.4$$

Substituting equation 5.3 into 5.2

$$r_i = \underline{x}^{*T} F^{*} W(i) F \underline{y} = \underline{X}^{*T} W(i) \underline{Y} \qquad 5.5$$

with :

$$\underline{X} = F \underline{x}$$

$$\underline{Y} = F \underline{y} \qquad 5.6$$

Expanding the right hand side of equation 5.5 :

$$r_i = \sum_k \exp\left(\frac{2\pi j}{N} ik\right) X_k^{*} Y_k \qquad 5.7$$

Since the signals are orthogonal, equation 5.7 is identically zero as the range of non-zero values of X_k and Y_k are disjoint. This result implies that, for an input signal that is a stationary Gaussian process, the outputs of the orthogonal filters are mutually uncorrelated and hence are independent processes. Thus any pointwise function of the output signals are also independent and hence uncorrelated.

5.2.1 Quadrature Filters

The filters used in the feature extraction process are analytic. For signals defined on a 1-d region of support, analyticity implies zero frequency response at negative frequencies [34]. Let z(n) be an analytic signal with transform Z(u) where :

$$Z(u) = 0 \quad u < 0 \qquad 5.8$$

The signal $z(n)$ is complex with real and imaginary parts $x(n)$ and $\hat{x}(n)$ respectively :

$$z(n) = x(n) + j\,\hat{x}(n) \qquad 5.9$$

If $X(u)$ and $\hat{X}(u)$ are the DFT's of $x(n)$ and $\hat{x}(n)$ respectively, then the following relations hold :

$$X(u) = X(-u)$$

$$\hat{X}(u) = -\hat{X}(-u)$$

$$\hat{X}(u) = -j\,X(u) \quad u > 0$$

$$= j\,X(u) \quad u < 0 \qquad 5.10$$

Taking the DFT of each side of equation 5.9 and substituting the expression for $\hat{X}(u)$ in terms of $X(u)$ gives :

$$Z(u) = 2\,X(u) \quad u > 0$$

$$= 0 \quad u < 0 \qquad 5.11$$

Note that for signals defined on 2-d supports, an analytic filter has a zero frequency response in at least one half of the Fourier plane [58].

Let $h(n)$ be an analytic filter with frequency

response H(u) and y(n) the result of filtering an input signal x(n). In general y(n) would be complex and have a bandpass frequency response. The envelope of y(n) is taken which effectively demodulates this bandpass response. Hence :

$$y(n) = h(n) * x(n) \qquad 5.12$$

$$e(n) = |y(n)|$$

$$= (|h_r(n) * x(n)|^2 + |h_i(n) * x(n)|^2)^{\frac{1}{2}} \qquad 5.13$$

where :

$$h(n) = h_r(n) + j\, h_i(n)$$

and e(n) is the envelope of y(n). The lowpass nature of the envelope of the signal can be seen by squaring both sides of equation 5.13 and interpreting the result in the frequency domain. Thus :

$$e^2(n) = |y(n)|^2 = y(n)\, y^*(n) \qquad 5.14$$

Taking the DFT of each side of equation 5.14 and using the convolution theorem :

$$E_1(u) = Y(u) * Y^*(-u) \qquad 5.15$$

$E_1(u)$ being the transform of $e^2(n)$. The right hand side of equation 5.15 is the autocorrelation of Y(u)

which produces the lowpass symmetric signal $E_1(u)$. Figures 5.1a and 5.1b illustrate the signals Y(u) and its autocorrelation respectively.

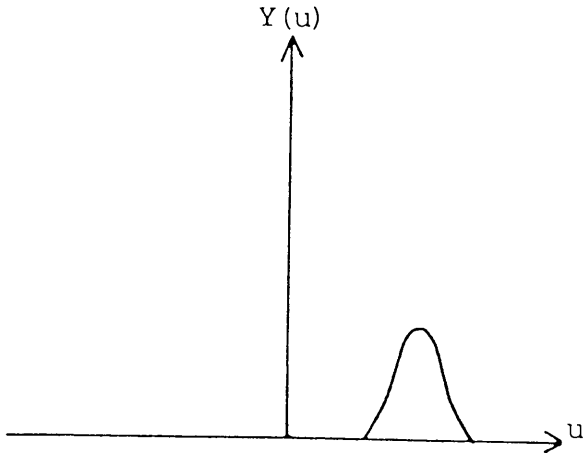


Figure 5.1a Y(u)

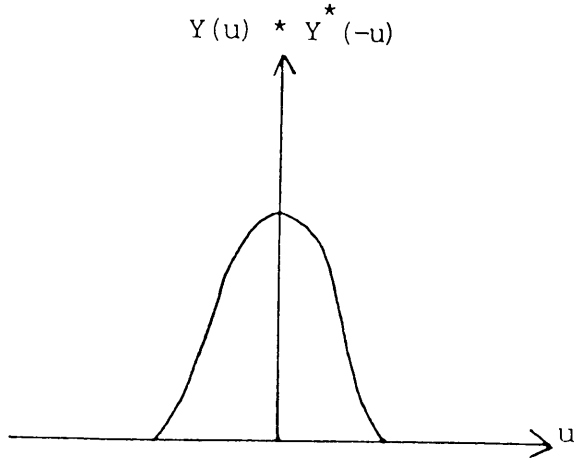


Figure 5.1b $Y(u) * Y^*(-u)$

If x(n) is a lowpass signal superimposed on a carrier signal of frequency ϕ then the envelope of the response of an analytic filter is independent of ϕ as is shown in equation 5.17.

$$x(n) = a(n) \exp jn\phi \qquad 5.16$$

$$E_1(u) = A_1(u-\phi) * A_1^*(\phi-u) = A_1(u) * A_1^*(-u) \qquad 5.17$$

where :

$$A_1(u) = H(u)\ A(u) \qquad 5.18$$

A(u) being the transform of a(n) and H(u) the frequency response of the analytic filter.

The lowpass nature of the envelope of the output

of analytic filters and the fact that the envelope is always positive are exploited in [58] and [59] to perform line and edge detection. Further the envelope detection property of the filters was used to obtain phase independent local orientation estimation in [58].

In the context of feature extraction for texture segmentation, the envelope of the outputs of the filters at each point are dependent on the energy of the input signal in the region of the Fourier domain defined by the bandpass region of the filter. This energy is computed from a local neighbourhood of the input signal whose size depends on the spatial extent of the impulse response of the filters.

5.2.2 Optimum Filter Design

As mentioned above the aim of the filters is to obtain high resolution both spatially and in feature space. This implies a maximisation of the mutual energy concentration both spatially and in spatial frequency. Specifically, assume that the filters have a frequency response limited to some pass band as defined by a truncation operator T_f. Hence for a signal defined on a 1-d support T_f is such that :

$$(T_f)_{ij} = k(i)\ \delta_{ij}$$

where :

$$k(i) = 1 \quad \text{iff } H(i) \neq 0 \qquad 5.19$$

where H(i) is the frequency response of the filter. Extension to signals defined on 2-d supports is straightforward. Let $\underline{\phi}$ be the signal vector corresponding to the frequency response which is bandlimited to the region specified by the truncation operator T_f and has maximum signal energy in the spatial domain within a region specified by the truncation operator T_s. In this case $\underline{\phi}$ is a solution to the eigenvalue problem [60] :

$$T_f F T_s F^* \underline{\phi}_i = \lambda_i \underline{\phi}_i \qquad 5.20$$

where F and F^* are forward and inverse DFT operators respectively and $\underline{\phi}_i$ and λ_i are the ith eigenvector and eigenvalue of equation 5.20. $\underline{\phi}$ corresponds to that eigenvector with maximum eigenvalue.

To show that this is the case consider an arbitrary signal $\underline{\gamma}$ bandlimited to the same region of the Fourier domain as $\underline{\phi}_i$. The eigenvectors $\underline{\phi}_i$ form a complete set of orthogonal basis vectors over the space of such bandlimited signals due to the Hermiticity of the operator $T_f F T_s F^* T_f$. Hence :

$$\underline{\gamma} = \sum_i a_i \underline{\phi}_i \qquad 5.21$$

where, if $\underline{\gamma}$ is normalised to unit energy :

$$\sum_i |a_i|^2 = 1 \qquad 5.22$$

Defining $(\underline{a}, \underline{b})$ as the scalar product between vectors $\underline{a}$ and $\underline{b}$:

$$(\underline{a}, \underline{b}) = \sum_i a_i b_i^* \qquad 5.23$$

and letting e_1 be the energy of the signal within the region defined by T_s :

$$e_1 = (F\, T_s\, F^*\, \underline{\gamma},\ F\, T_s\, F^*\, \underline{\gamma})$$

$$= \underline{\gamma}^T\, T_f\, F\, T_s\, F^*\, \underline{\gamma} \qquad 5.24$$

Substituting equation 5.21 into equation 5.24 :

$$e_1 = \sum_{ij} a_i\, a_j^*\, \underline{\phi}_i^T\, T_f\, F^*\, T_s\, F\, \underline{\phi}_j^*$$

$$= \sum_{ij} a_i\, a_j^*\, \lambda_j\, \underline{\phi}_i^T\, \underline{\phi}_j^*$$

$$= \sum_i |a_i|^2\, \lambda_i \qquad 5.25$$

Assume that the eigenvalues λ_i are ordered in increasing values of i. Then the expression on the right hand side of equation 5.25 is maximum when :

$$a_i = 1 \quad i = 1$$

$$= 0 \quad \text{otherwise} \qquad 5.26$$

and hence $\underline{\gamma}$ is the eigenvector of $T_f F T_s F^*$ corresponding to the maximum value of λ_i. Note also that by substituting this eigenvector into equation 5.24 the expression for the energy e_1 is just :

$$e_1 = \lambda_1 (\underline{\gamma}, \underline{\gamma}) = \lambda_1 \qquad 5.27$$

and therefore λ_1 represents the fraction of the total signal energy constrained within a region specified by the truncation operator T_f .

Thus the filter design procedure amounts to specifying the truncation operators and solving the resulting eigenvalue problem. Note that in equation 5.20 the term FT_sF^* is circulant and hence the matrix product is specified by a single column obtained by performing a DFT on the signal produced by taking the diagonal elements of T_s [60].

5.2.3 A Cartesian Separable Set of Filters for Texture Feature Extraction

Let $\underline{\phi}$ be an N-point signal satisfying equation 5.20 with truncation operators T_f and T_s given by :

$$(T_f)_{ij} = \delta_{ij} \quad i < \frac{N}{4}$$

$$= 0 \quad \text{otherwise} \qquad 5.28$$

$$(T_s)_{ij} = \delta_{ij} \quad i \leq M \text{ or } i > N - M$$

$$= 0 \quad \text{otherwise} \qquad 5.29$$

Hence an analytic frequency response is specified where negative frequencies are those with index greater than N/2.

λ is the fraction of the energy $(\underline{\phi},\underline{\phi})$ of $\underline{\phi}$ constrained in a region of size N/4 in the Fourier domain and of size 2M+1 in the spatial domain, $\underline{\phi}$ being the eigenvector corresponding to the largest eigenvalue of equation 5.20. In practice a transform size N=128 was taken and the value of M corresponding to the smallest value of λ greater than 0.5 chosen.

A set of filters are defined on a 2-d support with frequency responses occupying the 6 equally sized analytic regions of the Fourier plane labelled 1-6 in figure 5.2 where in this case negative frequencies are defined for v>N/2.

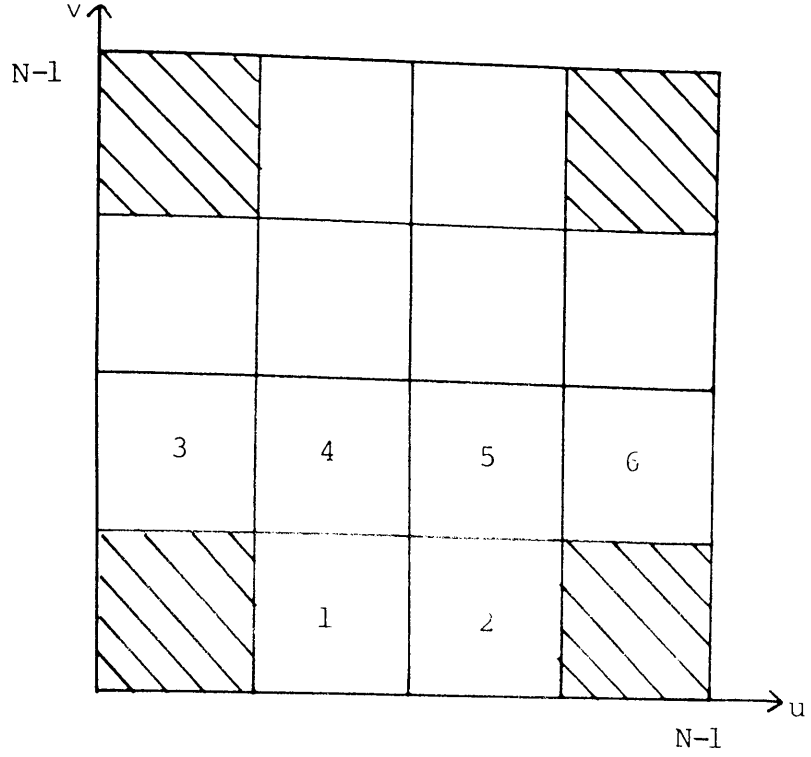


Figure 5.2 A set of cartesian separable frequency responses

The frequency response in any one region is obtained by shifting a pair of 1-d responses $\underline{\phi}$ defined above by appropriate amounts and forming the cartesian product of the shifted versions. Thus let S(i) be the shift operator corresponding to a right circular shift of i multiples of N/4 for i=0 to i=3. Define :

$$\underline{\phi}(i) = S(i)\,\underline{\phi} \qquad 5.30$$

Hence the frequency response of the filter corresponding to region 6 for example is given by :

$$H(6) = \underline{\phi}(3)\,\underline{\phi}(1)^{T} \qquad 5.31$$

where H(6) is an NxN matrix, $[H(6)]_{uv}$ being the response at spatial frequency coordinate (u,v). The

frequency response H(j) for j=1,6 are solutions to the equivalent 2-d eigenvalue problem specified by the operator $T_f F T_s F^*$. In this case operators T_s and T_f define 2-d regions in space and spatial frequency and F is the 2-d NxN point DFT operator. Each of the individual matrices of the above matrix product can be specified as a Kronecker product of equivalent 1-d operators [60]. Hence :

$$M = T_f \; F \; T_s \; F^* \qquad 5.32$$

$$M = M_1 \times M_2 \qquad 5.33$$

where M_1 and M_2 are the matrices of the 1-d operators and X defines a Kronecker product. The eigenvector $\underline{\Psi}$ of M is such that :

$$\underline{\Psi} = \underline{\Psi}_1 \times \underline{\Psi}_2 \qquad 5.34$$

where $\underline{\Psi}_1$ and $\underline{\Psi}_2$ are solutions to the 1-d problems defined by M_1 and M_2 respectively. In this case $\underline{\Psi}$ is an N^2 element vector equivalent to the matrix $\underline{\Phi}\,\underline{\Phi}^T$ with $\underline{\Phi}$ defined above and where :

$$[\underline{\Phi}\,\underline{\Phi}^T]_{uv} = (\underline{\Psi})_{Nu+v} \qquad u,v = 0 \,..\, N-1 \qquad 5.35$$

5.2.4 Combining Features at Different Frequency Scales

If the frequency response of the textures to be segmented are restricted to the shaded regions of figure 5.2, then there will be zero output if the set of filters described above are used. Thus a second set of 6 filters were specified occupying the corresponding 6 analytic regions of the low pass region each response being 1/2 the size in each spatial frequency dimension. This was done by re-specifying the truncation operator T_f in equation 5.28 and solving the new eigenvalue problem. Also the 1 dimensional filter responses have to be shifted in frequency by an appropriate amount before taking the cartesian product. More sets of 6 filters could be produced with logarithmically decreasing size. However, because of computational restrictions, only 2 sets of 6 filters were used in this work.

Note that the above logarithmic arrangement of filters is equivalent to taking the original set, sub-sampling in the spatial frequency domain and applying the new set to an ideal low pass filtered and sub-sampled version of the original signal. In this case filtering with two corresponding filters over different frequency scales is equivalent to running the same convolution kernel over two versions of the image at different spatial resolutions. Hence the dif-

ferent sets of filters respond to features in the textures at different spatial resolutions.

As a final point, included in the filter set were a pair of low pass filters one with a pass band indicated by the shaded region of figure 5.2 and the corresponding filter at the next frequency scale. These filters were designed by specifying a symmetric truncation operator in the frequency domain thus producing a real impulse response and the outputs of the filters used without any further processing. Hence a 14 element vector is produced at each image point. Note that this produces an over complete representation of the signal as the filters are not orthogonal.

A problem with combining different types of filtering operations is normalisation, in other words, how to scale the outputs of each filter. This was resolved by insisting that the variance of the outputs of each filter, for a white Gaussian noise input, are the same. This amounts to a simple linear scaling of the filter outputs.

The envelopes of the outputs of the set of filters thus define a random vector field $\underline{y}(m,n)$ at each image point. Thus for an input signal $x(m,n)$:

$$y_i(m,n) = | x(m,n) * h_i(m,n) | \qquad 5.36$$

where $h_i(m,n)$ is the impulse response of the ith analytic filter and $y_i(m,n)$ is the ith component of $\underline{y}(m,n)$.

5.3 Extension of Quadtree Segmentation

This section describes the extension to the quadtree segmentation scheme which allows it to operate on multi-dimensional vectors. Modifications to the three stages - quadtree smoothing, local centroid clustering and boundary estimation will be described in turn.

5.3.1 Quadtree Smoothing

A multi-dimensional quadtree is a trivial extension of the 1-d case defined in equation 3.2. The 1th component of the quadtree at level k, $\underline{q}(i,j,k)$ is just :

$$[\underline{q}(i,j,k)]_l = \frac{1}{4}\{[\underline{q}(2i,2j,k-1)]_l + [\underline{q}(2i+1,2j,k-1)]_l + [\underline{q}(2i,2j+1,k-1)]_l + [\underline{q}(2i+1,2j+1,k-1)]_l\} \quad 5.37$$

Thus, for example, a 6 dimensional quadtree can be defined, each dimension corresponding to 1 of the filters whose frequency response bandpass regions are shown in figure 5.2.

5.3.2 Multi-dimensional Local Centroid Clustering

The multi-dimensional clustering procedure is identical in principle to the 1 dimensional case described in section 3.5.3. Hence if $h(\underline{x})$ is the N dimensional joint histogram of the data at the highest quadtree level then the local centroid $\underline{\mu}(\underline{x})$ is given by :

$$\underline{\mu}(\underline{x}) = \underline{x} + \frac{\sum_{\underline{x}'\epsilon W} \underline{x}' h(\underline{x} + \underline{x}')}{\sum_{\underline{x}'\epsilon W} h(\underline{x} + \underline{x}')} \qquad 5.38$$

where in this case W is an N dimensional hypersphere and the local centroid is itself an N dimensional vector. The algorithm proceeds as in the 1 dimensional case where the histogram is updated by moving points in feature space to their local centroid position. As a practical note, the histogram $h(\underline{x})$ can initially have at most m^2 non-zero points where m is the spatial dimension at the highest quadtree level. This leads to a fast implementation of the algorithm which only considers points in feature space with non-zero histogram values.

Final classification is again a lookup table operation. In this case the lookup table at point $\underline{x}$ in feature space $\underline{l}(\underline{x})$ is an N dimensional vector which indicates the current position of the feature vector originally at point $\underline{x}$.

Note that because clustering is used to classify the feature vectors, a feature rejection procedure based on the eigenvector transform [29] is generally inappropriate. Such techniques reject those features containing least energy. In figure 5.3, a 2 dimensional joint histogram of the vector $\underline{x} = (x_1 x_2)$ is shown, the ellipses representing contours of equal histogram value. Although subspace x_1 has all of the information necessary to discriminate the 2 classes it contains less energy, as specified by the variance of the histogram along the x_1 axis, than does x_2 and hence will be rejected in favour of x_2.

This problem relates to the lack of a priori knowledge of class conditional statistics as discussed in chapter 3. Thus the use of within class and between class scatter matrices to optimise class separability is precluded [61].

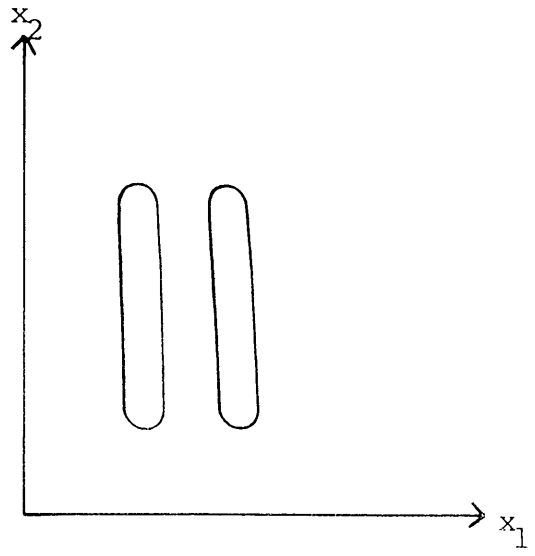


Figure 5.3 2 class 2-d joint histogram

5.3.2.1 Class Resolution and Signal Dimensionality

In general, if the signal dimensionality increases, the average Euclidean separation between data vectors increases. This then increases the minimum window size necessary to resolve classes with a consequent loss in resolution in class space. To quantify these effects in general is not possible. However, to give some indication, consider an N dimensional signal consisting of a zero mean white Gaussian noise process in each dimension of variance σ^2 . Also assume that each N element vector is independent of any other vector. The joint probability distribution of the signal is given by :

$$p(\underline{x}) = N(0,\Sigma) \qquad 5.39$$

where the notation $N(0,\Sigma)$ means a multi-variate Gaussian distribution with zero mean and covariance matrix Σ . In this case Σ is just the diagonal matrix :

$$(\Sigma)_{ij} = \sigma^2 \delta_{ij} \qquad 5.40$$

Defining :

$$R = || \underline{x} - \underline{y} || \qquad 5.41$$

as the Euclidean norm between vectors $\underline{x}$ and $\underline{y}$, the probability distribution of R is given by [32] :

$$p(R) = \frac{2}{(\sigma_1 \sqrt{2})^N} \quad \frac{R^{N-1} \exp(-R^2/2\sigma_1^2)}{\Gamma(N/2)} \quad U(R) \qquad 5.41$$

where :

$$\sigma_1 = \sigma\sqrt{2} \qquad 5.43$$

and :

$$\Gamma(a) = \int_0^\infty y^{a-1} \exp(-y)\, dy \qquad 5.44$$

and where :

$$U(R) = 1 \quad R \geq 0$$

$$= 0 \quad \text{otherwise} \qquad 5.45$$

The key quantity in this discussion is the probability

P defined as :

$$P = Pr\ (R > w) \qquad 5.46$$

where W is the radius of an N dimensional hypersphere centred on a randomly chosen vector. P is then given by :

$$P = \int_{w}^{\infty} p(R)\ dR$$

$$= \frac{2}{(\sigma_1\sqrt{2})^N\ \Gamma\ (N/2)} \int_{w}^{\infty} R^{N-1} \exp\ (-R^2/2\sigma_1^2)\ dR \qquad 5.47$$

Re-arranging the integral in equation 5.47, it can be reduced to the form :

$$P = 1 - \frac{1}{\Gamma\ (N/2)}\ \gamma(N/2,\ w^2/2\sigma_1^2) \qquad 5.48$$

where the quantity $\gamma(a,z)$ is an incomplete gamma function defined as [62] :

$$\gamma(a,z) = \int_{0}^{z} t^{a-1} \exp\ (-t)\ dt$$

$$= a^{-1} z^a\ {}_1F_1\left(\begin{matrix} a \\ a+1 \end{matrix} \middle| -z\right) \qquad 5.49$$

with ${}_1F_1\left(\begin{matrix} a \\ c \end{matrix} \middle| z\right)$ a hypergeometric series defined by :

$$ {}_1F_1\left(\begin{matrix}a\\c\end{matrix}\middle| z\right) = \sum_{k=0}^{\infty} [(a)_k z^k/(c)_k k!] \qquad 5.50 $$

with :

$$ (a)_k = a(a+1) \ldots (a+k-1) \qquad 5.51 $$

Substituting equation 5.49 into equation 5.48 :

$$ P = 1 - \frac{2}{N\,\Gamma(N/2)} \frac{w^N}{2^{N/2}\sigma_1^N} \; {}_1F_1\left(\begin{matrix}N/2\\1+N/2\end{matrix}\middle| -\frac{w^2}{2\sigma_1^2}\right) \qquad 5.52 $$

The hypergeometric series in equation 5.52 is given by :

$$ {}_1F_1\left(\begin{matrix}N/2\\1+N/2\end{matrix}\middle| -\frac{w^2}{2\sigma_1^2}\right) = \sum_{k=0}^{\infty} \frac{(N/2)_k}{(1+N/2)_k k!} \frac{w^{2k}}{2^k\sigma_1^{2k}} (-1)^k $$

$$ = \sum_{k=0}^{\infty} \left(\frac{N}{N+2k}\right) \frac{(-1)^k}{k!} \frac{w^{2k}}{2^k\sigma_1^{2k}} \qquad 5.53 $$

Defining $s=w/\sigma$ and substituting equation 5.53 into equation 5.52 the expression for P in terms of s and the signal dimensionality N becomes :

$$ P = P(s,N) = 1 - \frac{2^{1-N}}{\Gamma(N/2)} s^N \sum_{k=0}^{\infty} \frac{s^{2k}}{(N+2k)} \frac{(-1)^k}{k!2^k} \qquad 5.54 $$

In figure 5.4 the values of P(s,N) are sketched for N=1 to 14 and for s=0.5, 1.0, 1.5 and 2.0.

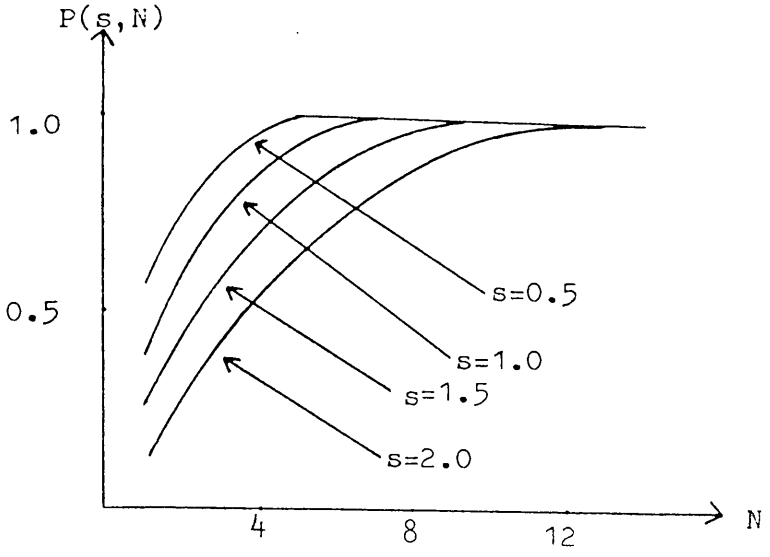


Figure 5.4 Graph of P(s,N)

When P is sufficiently close to unity, there is a high probability that a vector chosen at random will have none of its neighbours within a distance W. Hence it will be classed by itself on convergence of the clustering algorithm. When the average number of such single classifications is significant then the algorithm will not produce meaningful results.

5.3.3 Boundary Estimation

The local centroid clustering algorithm classifies the data vectors at the highest quadtree level by assigning a class identifier to each one, this identifier being an arbitrary scalar value. Thus boundary estimation proceeds in a manner identical to the 1 dimensional case where an uncertainty region $B_c(k)$ is defined at each level k of the quadtree. The set of data vectors that have class identifiers belonging to this set is then smoothed with a linear filter whose

impulse response depends on the estimated signal to noise ratio in each dimension :

$$[\underline{q}'(i,j,k)]_l = [q(i,j,k)]_l * h(i,j,\rho_l) \qquad 5.55$$

where :

$$c(\underline{q}(i,j,k)) \in B_c(k)$$

c(q(i,j,k)) being the classification of q(i,j,k). In equation 5.55 $(q'(i,j,k))_l$ is the lth component of the smoothed data. The signal to noise ratio of the lth dimension between regions r and s at level k is given by :

$$\rho_l(r,s,k) = 2\,\frac{|\,\mu_l(r,k) - \mu_l(s,k)\,|}{\sigma_l(r,k) + \sigma_l(s,k)} \qquad 5.56$$

where as before $\mu_l(p,k)$ and $\sigma_l(p,k)$ are the estimated signal mean and standard deviation in region p at level k of the lth component of the quadtree. The filter $h(i,j,\rho_l)$ is formed in a manner identical to that in the 1 dimensional case using iterated convolution of a 3x3 mask (equation 3.48) where the cut off parameters of the piecewise linear function defining the weights of the masks are the same.

Final classification of the data following smoothing is again according to a 'nearest class mean' criterion where the distance between a pair of vectors in class space is taken to be the Euclidean norm

between the vectors. Also each element of the vectors used to compute the norm are weighted according to the estimated signal to noise ratio in that dimension. Thus final classification of a vector $\underline{q}(i,j,k)$ at level k is given by :

$$\underline{q}(i,j,k) \rightarrow c(\underline{\mu}(r,k))$$

iff

$$||W(\rho(r,s,k))[\underline{q}(i,j,k)-\underline{\mu}(r,k)]|| < ||W(\rho(r,s,k))[\underline{q}(i,j,k)-\underline{\mu}(s,k)]|| \qquad 5.57$$

where $c(\underline{\mu}(r,k))$ is the class associated with a region r with local mean $\underline{\mu}(r,k)$ at level k and the nearest neighbours of the father node of $\underline{q}(i,j,k)$ have been assigned to regions r and s. $W(\rho(r,s,k))$ is an NxN diagonal weighting matrix, N being the dimensionality of the data vectors, where the diagonal elements depend on the local signal to noise ratio at level k :

$$[W(\rho(r,s,k)]_{ii} = \frac{\rho_i(r,s,k)}{Tr(W(\rho(r,s,k))} \qquad 5.58$$

where Tr(A) is the trace of matrix A. The linear weighting of each dimension according to the signal to noise was chosen experimentally and found to produce good results.

As a final point, because of the finite spatial extent of the filters, there is little point in

continuing the boundary estimation procedure down to full spatial resolution. In particular, for each dimension, the boundary is propagated down to that quadtree level appropriate for the filter size used to produce the signal in that dimension. Thus for the filters of size N/4xN/4 in the Fourier domain, boundary propagation is terminated at the first quadtree level above the base, this level effectively being the result of a low pass filtering operation with a filter of approximately the same spatial width. Further for the filters of 1/2 this size in each spatial frequency dimension, the boundary propagation is terminated at level 2 of the quadtree.

In order to produce a segmentation result at full spatial resolution, a simple interpolation scheme was adopted. This involved replacing the signal in the low pass filtering operation of equation 5.55 with the mean of the signal in the appropriate region and doing the final classification in the usual way.

Chapter 6

Texture Segmentation Results

6.1 Introduction

The previous chapter described a texture segmentation scheme based on a set of cartesian separable orthogonal filters and the quadtree classifier described in chapter 3. This chapter is concerned with the results of the segmentation scheme for a range of synthetic and natural textures. The results are of a qualitative nature only as it is not possible, in general, to give theoretical estimates of the probability of classification error due to the lack of a statistical model of the input texture fields.

6.2 A Simple Mean/Variance Segmentation

Figure 6.1a shows an image containing 3 regions, each region consisting of white Gaussian noise, where all the images in this section are 256x256x8 bits. The means and standard deviations of the signal in each region are (100,20), (100,45) and (130,20) the first figure inside the parentheses being the mean signal level. Hence in figure 6.1a there is one grey level boundary, one boundary across which there is a difference in signal variance and across the third boundary

there is a difference in both mean grey level and variance. Thus running the segmentation on either the grey level image or on an image representing the difference in variance would only pull out two classes. These two operations could of course be run sequentially and the result combined but this would lead to ambiguities across the third boundary.

In order to resolve this problem a 2 element vector $\underline{z}=(f,g)$ is defined at each point in the image. The first element f is just the image grey level at that point. The second element g is an estimate of the signal energy in the high-pass region of the Fourier domain as defined by a low pass filter h(x,y). Thus if h(x,y) is a Gaussian impulse kernel of variance σ^2 :

$$h(x,y) = A \exp(-(x^2+y^2) / \sigma^2) \qquad 6.1$$

where A is chosen to normalise h(x,y) to unit dc :

$$\sum_{x,y \in W} h(x,y) = 1 \qquad 6.2$$

W defining the region of support of h, then g is given by :

$$g(x,y) = |f(x,y) - h(x,y) * f(x,y)| \qquad 6.3$$

Figure 6.1b shows the result of this operation for $\sigma = 2.0$.

The actual value of σ determines the mean and variance of $g(x,y)$ in such a way that the inter-region signal to noise is independent of σ. To see this let the variance of the zero mean signal $f(x,y)$ within a single region be σ_f^2. By squaring each side of equation 6.3 and taking expectations it can be shown that :

$$E(g(x,y)^2) = \sigma_f^2 \left(1 + \sum_{x',y' \in W} |h(x',y')|^2 - 2h(0,0)\right)$$

$$= s^2 \sigma_f^2 \qquad 6.4$$

where :

$$s = \left(1 + \sum_{x',y' \in W} |h(x',y')|^2 - 2h(0,0)\right)^{\frac{1}{2}} \qquad 6.5$$

Letting :

$$t(x,y) = f(x,y) - h(x,y) * f(x,y) \qquad 6.6$$

then $t(x,y)$ is zero mean and hence :

$$\sigma_t^2 = \sigma_f^2 s^2 \qquad 6.7$$

σ_t^2 being the variance of $t(x,y)$. From equations 6.6 and 6.3 :

$$g(x,y) = |t(x,y)| \qquad 6.8$$

Define $p_g(x)$ and $p_t(x)$ as the probability densities of $g(x,y)$ and $t(x,y)$ respectively. Also note from

equation 6.6 that $p_t(x)$ is a zero mean Gaussian distribution because $f(x,y)$ and $h(x,y)*f(x,y)$ are both Gaussian, the convolution being the sum of independent Gaussian random variables. Further it can easily be shown that [32] :

$$p_g(x) = 2p_t(x)\ U(x) \qquad 6.9$$

where :

$$U(x) = 1 \quad x \geq 0$$

$$= 0 \quad \text{otherwise} \qquad 6.10$$

Hence :

$$E(g(x,y)) = \int_{-\infty}^{\infty} x\ p_g(x)\ dx$$

$$= \frac{2}{\sigma_t\sqrt{2\pi}} \int_{0}^{\infty} x \exp\ (-x^2/2\sigma_t^2)\ dx$$

$$= \sqrt{\frac{2}{\pi}}\ \sigma_t \qquad 6.11$$

The variance of $g(x,y)$, σ_g^2 is then given by :

$$\sigma_g^2 = E(g(x,y)^2) - E(g(x,y))^2$$

$$= \sigma_t^2\ (1 - \frac{2}{\pi}) \qquad 6.12$$

For a pair of noise fields $f_1(x,y)$ and $f_2(x,y)$ with variances $\sigma_{f_1}^2$ and $\sigma_{f_2}^2$ respectively and with output envelopes $g_1(x,y)$ and $g_2(x,y)$, the inter-region

signal to noise is, as before :

$$\rho(g_1\ g_2) = 2\ \frac{|\ E(g_1) - E(g_2)\ |}{\sigma_{g_1} + \sigma_{g_2}} \qquad 6.13$$

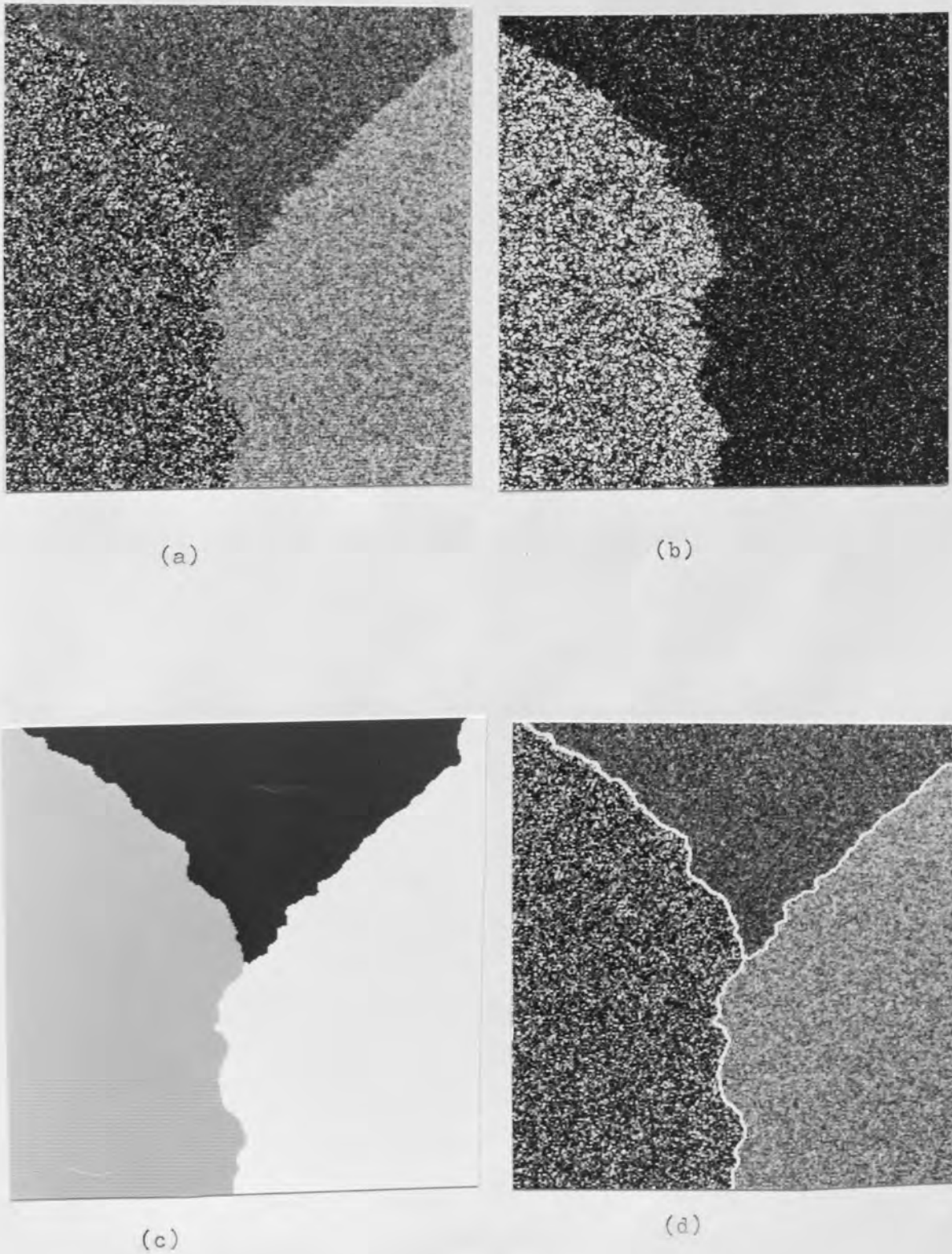
Substituting equations 6.11, 6.12 and 6.13 and using equation 6.7, this expression becomes :

$$\rho(g_1\ g_2) = k\ \frac{|\ \sigma_{f_1} - \sigma_{f_2}\ |}{\sigma_{f_1} + \sigma_{f_2}} \qquad 6.14$$

where k is a constant independent of s and hence independent of the size of the low pass filter h(x,y).

The clustering algorithm was run on the 2 dimensional joint histogram of $\underline{z}$ following 4 quadtree smoothing operations. The boundary was propagated down to the zeroth quadtree level in the manner described in the previous chapter. Figure 6.1c shows the segmentation and figure 6.1d the result of superimposing the region boundaries onto the original image. It can be seen from figure 6.1d that there is good agreement between the perceived boundaries and those obtained by the segmentation scheme.

This example demonstrates the benefit of combining individual features, in this case the grey level and the envelope of the high pass filtering operation, into a single vector and operating on the joint histogram of the vector field. It is only by using this



(a) (b)

(c) (d)

Figure 6.1 Mean/variance segmentation of white Gaussian noise fields.

technique that the 3 classes can be found quite naturally by the clustering algorithm.

6.3 Multi-dimensional Texture Segmentation Results

6.3.1 Segmentation of Synthetic Bandpass Textures

Let $H_i(u,v)$ be the frequency response in the ith analytic region of figure 5.2 for i=1 to 6. Further let $G_i(u,v)$ be the frequency response in the corresponding ith region of the low pass area of figure 5.2. Using a zero centred set of spatial frequency axes, that is to say for u and v defined in the range $(-N/2, N/2-1)$ the following symmetric filters can be specified :

$$H_i'(u,v) = H_i(u,v) + H_i(-u,-v) \qquad 6.15$$

and corresponding filters $G_i'(u,v)$. From equation 6.15 the symmetry relation :

$$H_i'(u,v) = H_i'(-u,-v) \qquad 6.16$$

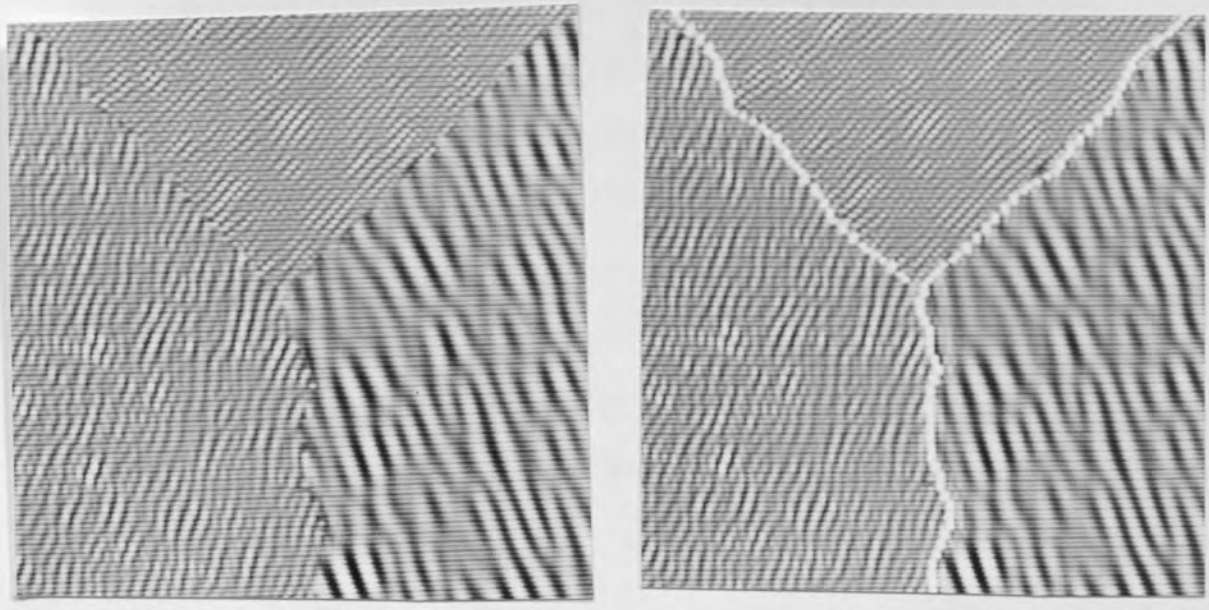
holds and hence the impulse responses of the filters are real.

These sets of filters were applied to an image consisting of white Gaussian noise. In this case the textures obtained are similar in appearance to those

obtained in [63] although here the real part of the complex impulse response of the analytic filters are taken. Such a method does not allow a cartesian separable representation of the frequency response of the filters.

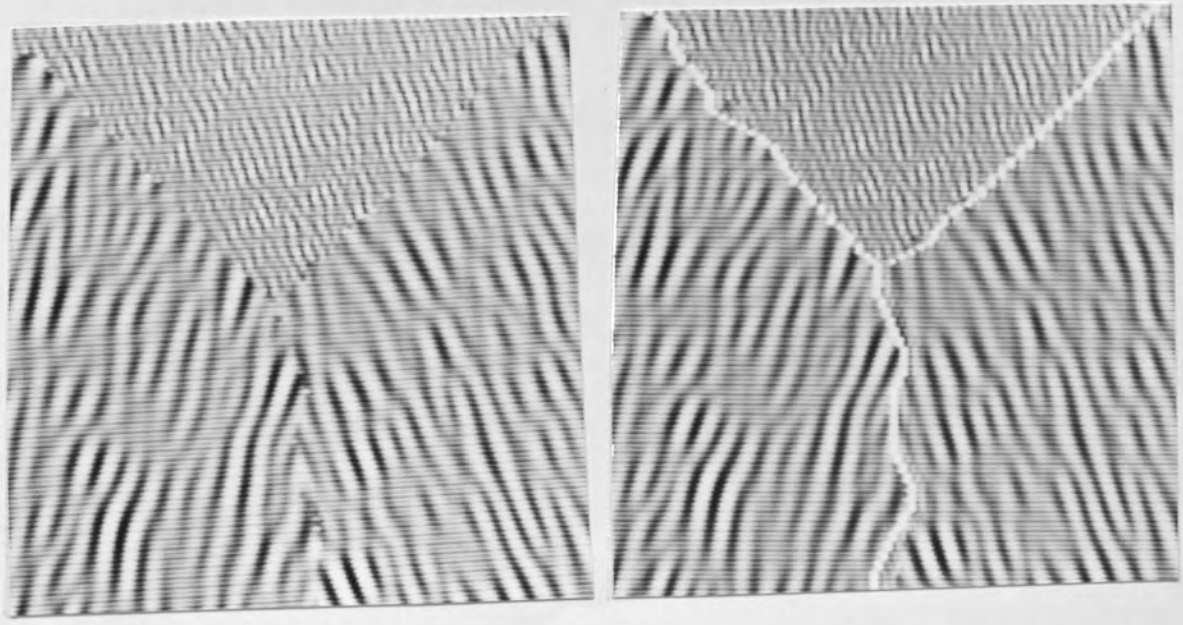
Figures 6.2a and 6.3a show a pair of images each consisting of 3 of these bandpass textures (the images in this section are all 128x128x8 bit.) Specifically 6.2a consists of the textures produced by $H_1'(u,v)$, $H_4'(u,v)$ and $G_1'(u,v)$ and 6.3a the textures produced by $H_2'(u,v)$, $G_1'(u,v)$ and $G_2'(u,v)$. Figures 6.2b and 6.3b show the results of the segmentation where only the region boundaries are shown. It can be seen that there is close agreement between the perceived boundaries and those obtained from the texture segmentation scheme.

As in the first segmentation example of figure 6.1, the 3 texture classes present in each image are not separable in any single 1 dimensional sub-space. Only by combining the filter outputs into one large feature vector can the 3 classes be separated in a single operation. Note in particular that this approach copes with the fact that each example contains textures of different coarsenesses produced by filters with bandpass regions covering different frequency scales.



(a) (b)

Figure 6.2 Segmentation of synthetic bandpass textures.



(a) (b)

Figure 6.3 Segmentation of synthetic bandpass textures.

6.3.2 Texture Synthesis Using the Outputs of the Clustering Algorithm

A question of some interest is whether the images of figures 6.2a and 6.3a can be synthesised using the local mean vectors of each texture class. One precondition that must be met is that the original texture must have a phase randomised counterpart that is perceptually similar to it. The reason is that the synthetic textures are produced by filtering white noise with an appropriate zero phase filter and hence have the same phase as the noise.

The local centroid clustering algorithm assigns to each region a 12 element vector $\underline{k}$ (the 2 lowpass filter outputs have been excluded in these synthesis experiments for simplicity) where :

$$k_i = E\,|\,h_i(x,y) * t(x,y)\,| \qquad i=1,12 \qquad 6.17$$

where $h_i(x,y)$ is the impulse response of the ith analytic filter. Let $h_i'(x,y)$ be the corresponding impulse response of the symmetric filter as defined in equation 6.15 and $f(x,y)$ be a white Gaussian noise signal. The synthesis procedure amounts to the computation of the coefficients a_i for i=1 to 12 as a function of the expectations k_i where the a_i are defined by :

$$t'(x,y) = \sum_j a_j f(x,y) * h'_j(x,y) \qquad 6.18$$

$t'(x,y)$ being the synthesised texture. Substituting this expression for $t'(x,y)$ into equation 6.17 :

$$k_i = E \left| h_i(x,y) * \sum_j a_j f(x,y) * h'_j(x,y) \right| \qquad 6.19$$

Since the frequency responses $H_i(u,v)$ and $H_j(u,v)$ are disjoint for $i \neq j$:

$$h_i(x,y) * h_j(x,y) = A \delta_{ij} \qquad 6.20$$

A being a constant depending on normalisation, equation 6.19 becomes :

$$E \left| a_i h_i(x,y) * h'_i(x,y) * f(x,y) \right| = k_i \qquad 6.21$$

Defining :

$$g_i(x,y) = h_i(x,y) * h'_i(x,y) \qquad 6.22$$

with transform :

$$G_i(u,v) = H_i(u,v)^2 \qquad 6.23$$

the expression for a_i becomes :

$$a_i = \frac{k_i}{E\ |g_i(x,y) * f(x,y)|} \qquad 6.24$$

Let :

$$q_i(x,y) = |g_i(x,y) * f(x,y)|$$

$$= |g_i^{(r)}(x,y) * f(x,y) + j\, g_i^{(i)}(x,y) * f(x,y)| \qquad 6.25$$

where superscripts r and i stand for real and imaginary respectively. Both the real and imaginary parts of equation 6.25 are the sum of independent Gaussian random variables and hence are themselves Gaussian random variables. Furthermore they are zero mean because the filters are bandpass and hence have a zero dc term, are of equal variance because the real and imaginary components of the filters are of equal energy and they are independent of each other. This last point may be seen by calculating the covariance between the outputs of the real and imaginary components of the analytic filter $g_i(x,y)$:

$$E[(g_i^{(r)}(z)*f(z))(g_i^{(i)}(z)*f(z))] = \sum_{z'z''} g_i^{(r)}(z')g_i^{(i)}(z'')E(f(z-z')f(z-z''))$$

$$= \sigma_f^2 \sum_{z'} g_i^{(r)}(z')\, g_i^{(i)}(z')$$

$$= 0 \qquad 6.26$$

where z is the 2 element vector (x,y), the signals are

assumed to be real and $f(x,y)$ is zero mean and of variance σ_f^2 . The last equality in equation 6.26 follows because the scalar product between the symmetric and anti-symmetric parts of the analytic frequency response is zero and scalar product is preserved under the DFT operation.

In view of the above mentioned properties of the real and imaginary components of equation 6.25, the probability density of $q_i(x,y)$ is a Rayleigh distribution [32] of the form :

$$p(x) = \frac{x}{\sigma_1^2} \exp\left(-x^2/2\sigma_1^2\right) U(x) \qquad 6.27$$

σ_1^2 being the variance of the outputs of filters $g_i^{(r)}(z)$ and $g_i^{(i)}(z)$ which is given by :

$$\begin{aligned} \sigma_1^2 &= \frac{\sigma_f^2}{2} \sum_z |g_i(z)|^2 \\ &= \frac{\sigma_f^2}{2} \sum_z |h_i(z) * h_i'(z)|^2 \\ &= \frac{\sigma_f^2}{2} \sum_{u,v} |H_i(u,v)|^4 \end{aligned} \qquad 6.28$$

the last equality being a consequence of Parseval's theorem [34]. The expectation of $p(x)$ can be readily evaluated to give :

$$E(q_i) = \sigma_1 \sqrt{\frac{\pi}{2}}$$

$$= \sigma_f \sqrt{\frac{\pi}{4}} \left(\sum_{u,v} |H_i(u,v)|^4 \right)^{\frac{1}{2}} \qquad 6.29$$

Substituting this expression into equation 6.24 :

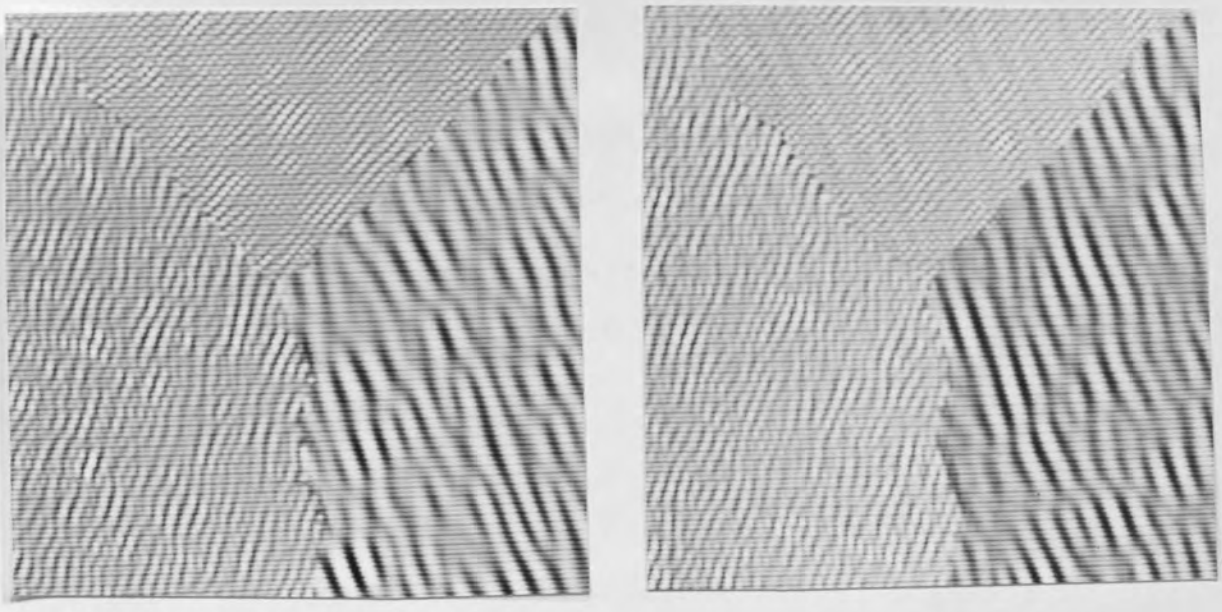
$$a_i = \frac{2k_i}{\sigma_f \sqrt{\pi} \left(\sum_{u,v} |H_i(u,v)|^4 \right)^{\frac{1}{2}}} \qquad 6.30$$

Thus using the mean vector $\underline{k}$ as calculated by the clustering algorithm, the weights a_i of each filter can be evaluated.

Figures 6.4a-b and 6.5a-b show a pair of reconstructions using this method, the original texture being on the left with its reconstruction on the right. Note that the filtering operations in this case had to be performed spatially as the filter to be used at each image point depends on the mean vector k at that point. This necessitated truncating the impulse response of the filter to a 15x15 mask to reduce the computational burden.

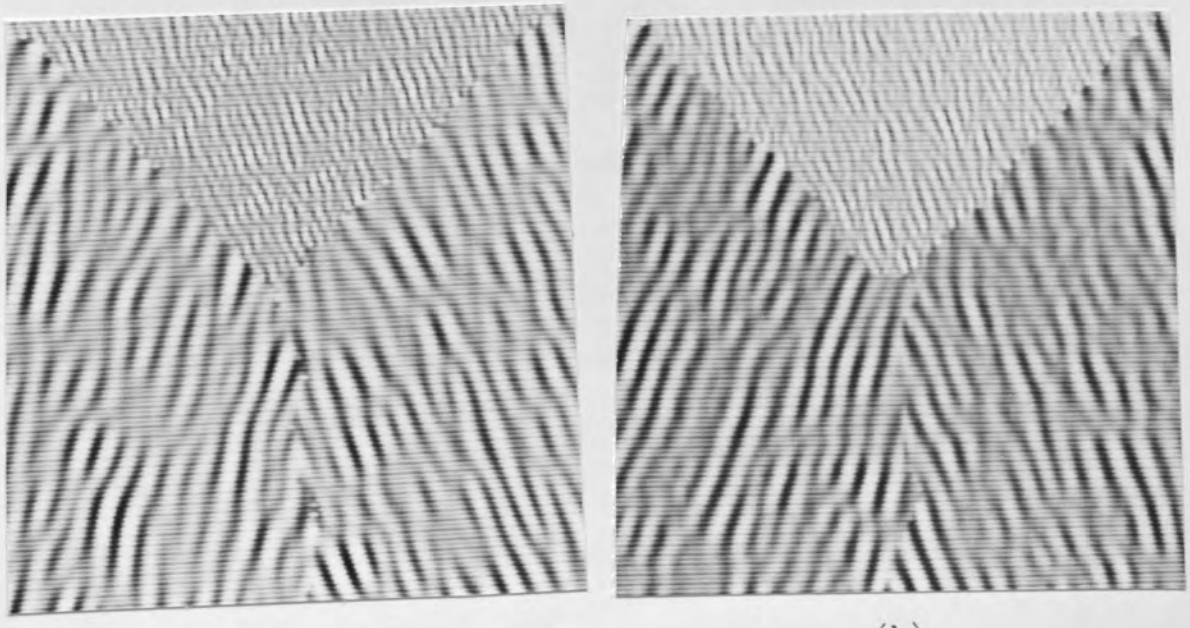
6.3.3 General Texture Segmentation Results

A number of segmentations were carried out on a range of synthetic and natural textures. In each case



(a) (b)

Figure 6.4 Reconstruction of synthetic bandpass textures.



(a) (b)

Figure 6.5 Reconstruction of synthetic bandpass textures.

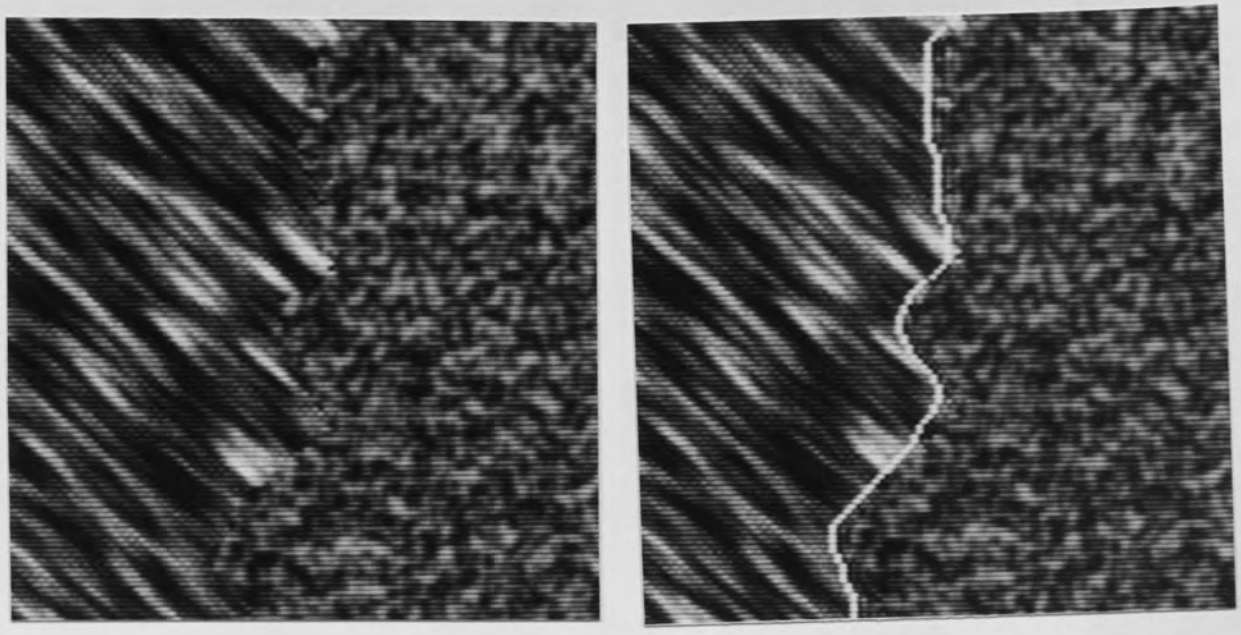
only the region boundaries are shown which are superimposed on the originals. The full 14 dimensional feature set, as described in section 5.2.4, was used in each case.

Figures 6.6a-b and 6.7a-b show a pair of synthetic texture segmentations, the textures being generated by low pass filtering a random impulse noise field as described in chapter 2. In particular, figure 6.7 consists of a set of 3 textures, the textures being identical in every respect except for directionality. Although the filters are not wedge shaped filters tuned to particular orientations, figure 6.7b indicates that textures differing in directionality only are still separable using the arrangement of filters described in the previous chapter.

Figures 6.8a-b, 6.9a-b, 6.10a-b and 6.11a-b show pairs of natural textures and their segmentations. The natural textures are taken from Brodatz and are (cork, seafan), (grass, water), (paper, seafan) and (sand, paperfibre). Again adequate segmentations are observed. Figure 6.12a-b shows an object/background segmentation, the object texture being bubble and the background being cork. Finally figure 6.13a-b shows a 3 class natural texture segmentation, the textures in this case being paperfibre, water and cork.

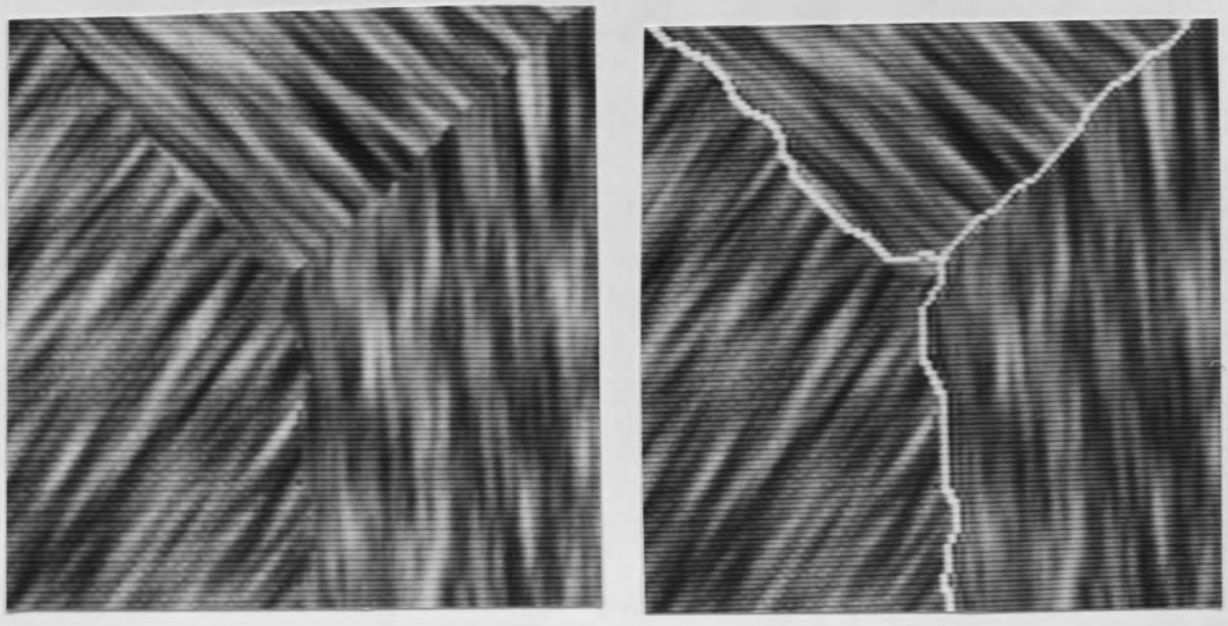
The classification error in each of these cases

was measured in terms of the number of misclassifications per boundary point. Figure 6.14 gives a table of these results where it can be seen that the error ranges from 1.8 to 2.9 errors/boundary point.



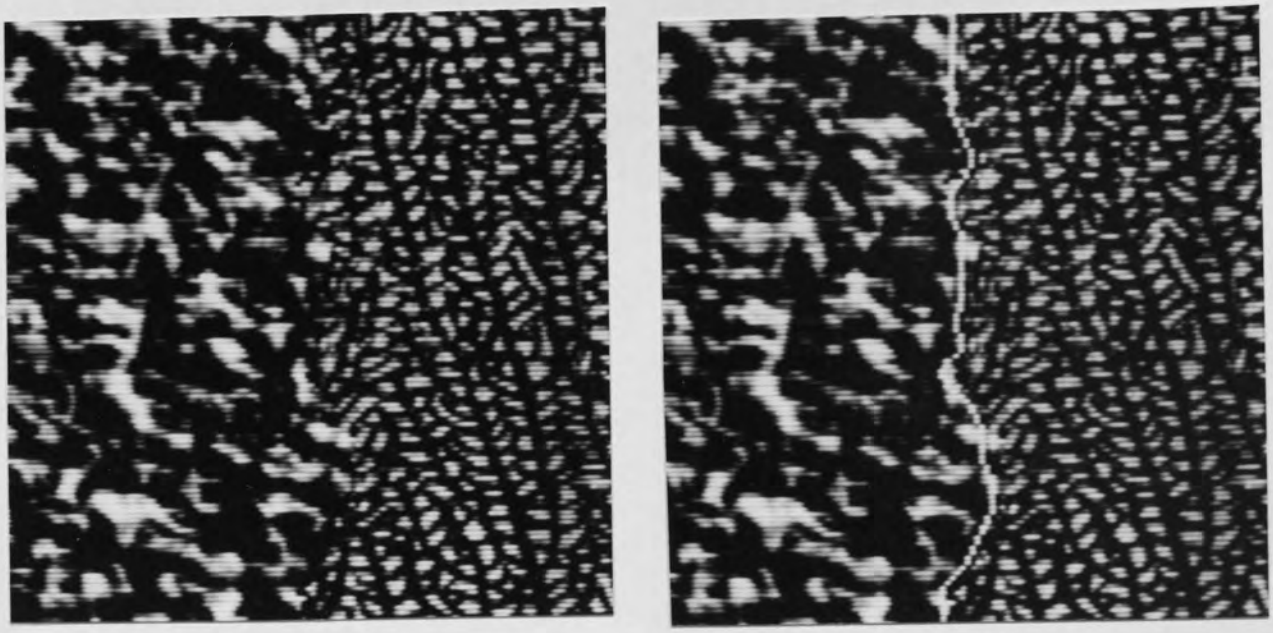
(a) (b)

Figure 6.6 Synthetic texture segmentation.



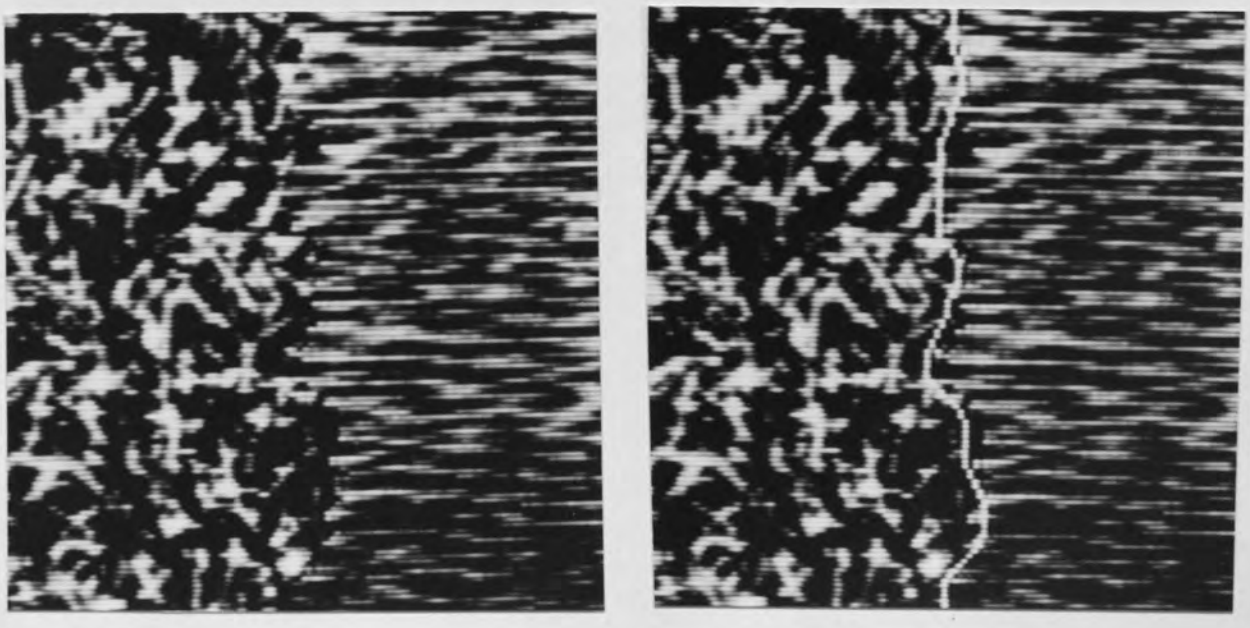
(a) (b)

Figure 6.7 3 class synthetic texture segmentation.



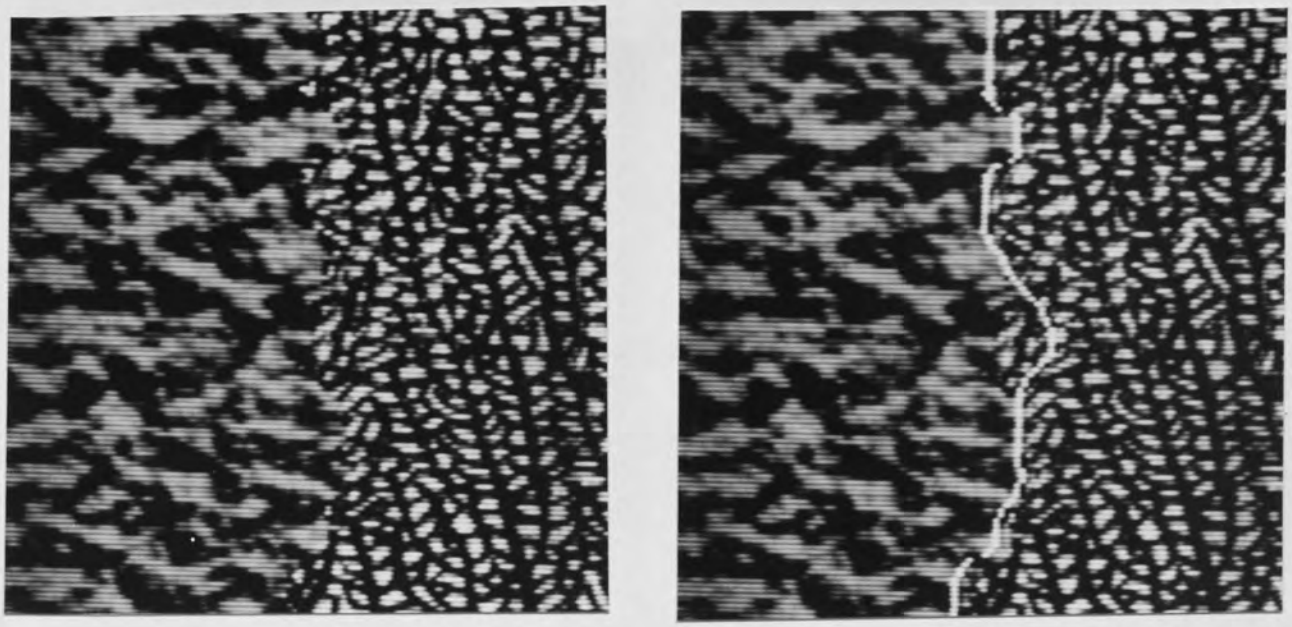
(a) (b)

Figure 6.8 Segmentation of the natural textures cork and seafan.



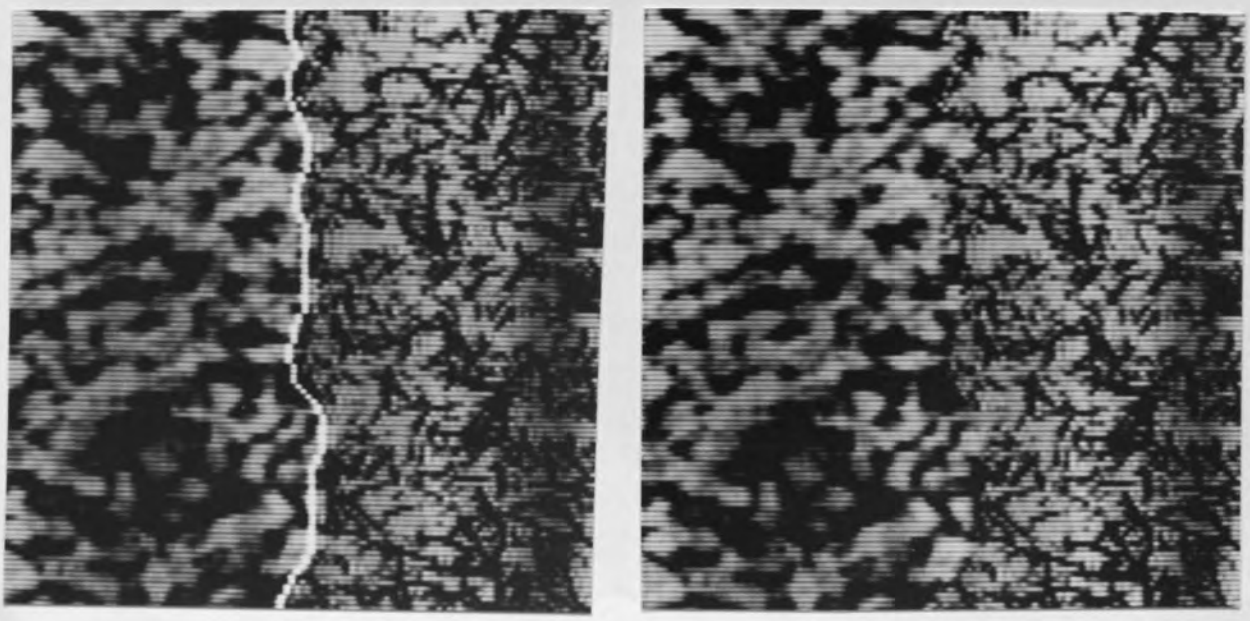
(a) (b)

Figure 6.9 Segmentation of the natural textures grass and water.



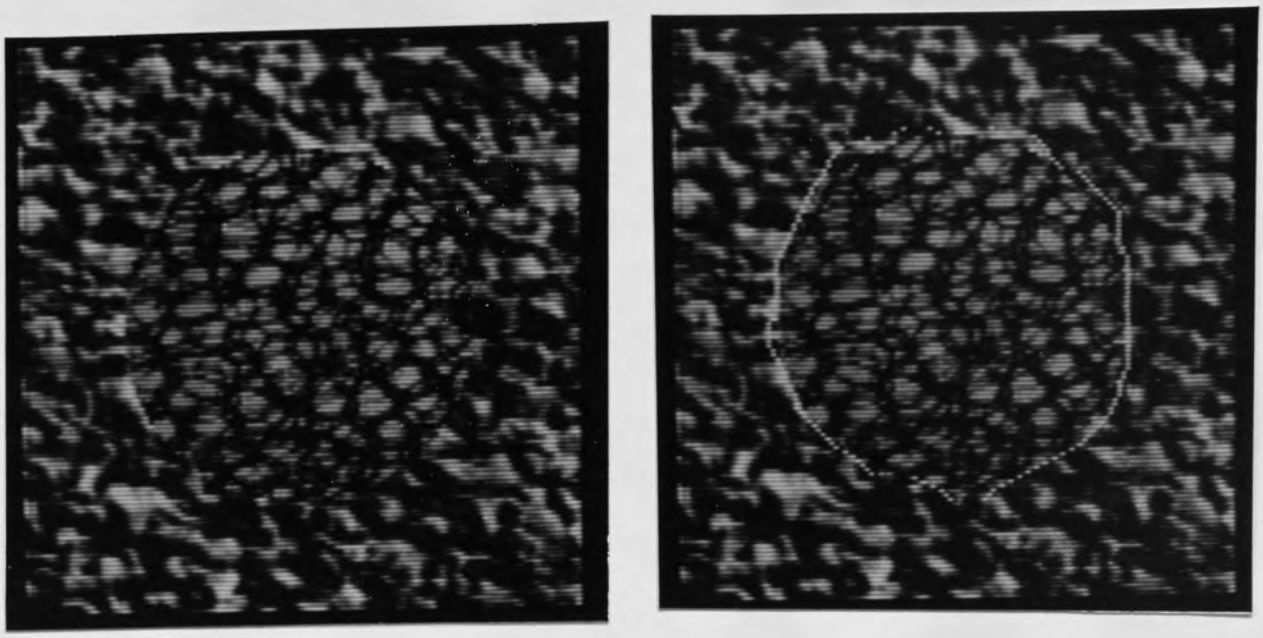
(a) (b)

Figure 6.10 Segmentation of the natural textures paper and seafan.



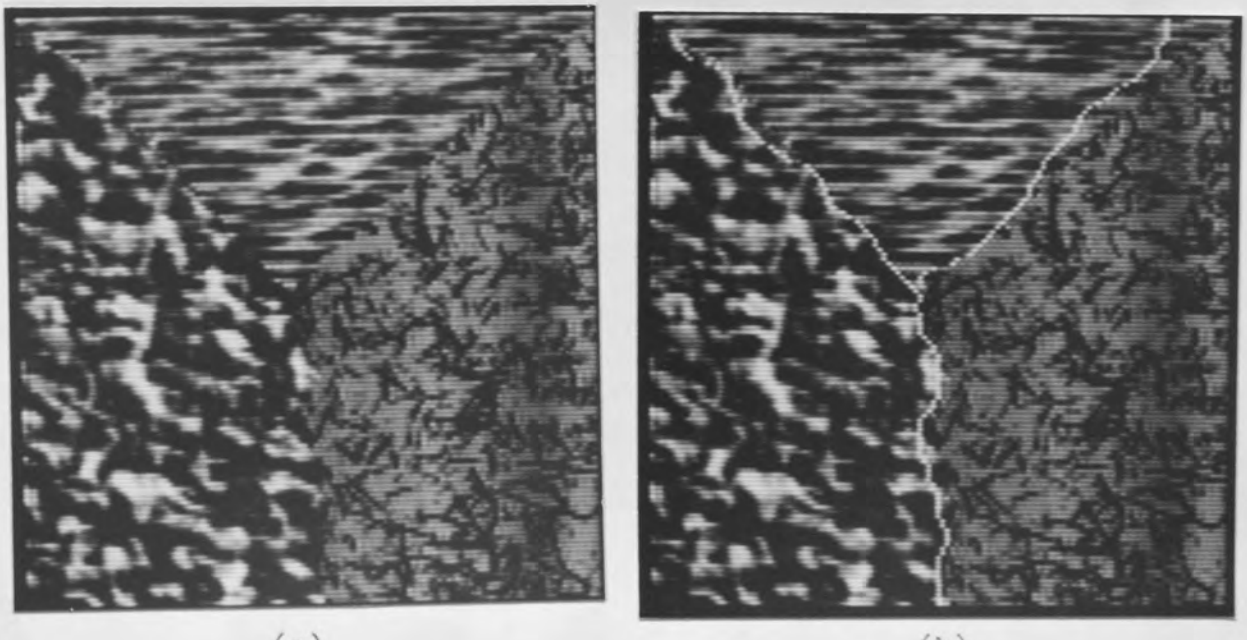
(a) (b)

Figure 6.11 Segmentation of the natural textures sand and paperfibre.



(a) (b)

Figure 6.12 Segmentation of the natural textures cork and bubble.



(a) (b)

Figure 6.13 Segmentation of the natural textures paperfibre, water and cork.

Segmentation	Classification error/boundary point
Synthetic texture pair (figure 6.6)	2.2
3 synthetic directional textures (figure 6.7)	1.8
Cork/Seafan (figure 6.8)	1.8
Grass/Water (figure 6.9)	2.3
Paper/Seafan (figure 6.10)	2.9
Sand/Paperfibre (figure 6.11)	2.5
Cork/Bubble object/background (figure 6.12)	2.5
Paperfibre/Water/Cork (figure 6.13)	2.3

Figure 6.14. Texture Classification Errors

Chapter 7

Conclusions and Future Work

The work described in this thesis is concerned with the description and segmentation of texture.

Chapter 2 introduced a stochastic model of texture which is phase independent. It was noted how this model corresponds to more conventional views of texture based on second order spatial averages and in particular the autocorrelation. However, such views provide no insight into the nature of the signals which cause the perceptual invariance when these averages remain constant. Adapting a phase independent interpretation bridges the gap between the properties of the signals constituting texture and its visual perception. Specifically the phase independent interpretation provides a distinction between the perceptual stationarity of signals and actual stationarity of the process, the observed signal being a realisation of that process. Evidence was presented indicating that those textures that are realisations of a random phase process, given a simple constraint on the marginal probability distributions of the phase, appear 'stationary' in the sense of lack of any local structure. Further, it was pointed out that stationarity of a generating process does not necessarily imply

perceptual stationarity of a realisation of that process if the stationarity is a result of spatially concentrated events generated at random locations. That this is the case is not surprising in view of the interpretation of phase randomisation in terms of global shifts of a given signal.

The above discussion seems to imply that their are 2 distinct types of texture - those that can be modelled with global statistical measures (such as the autocorrelation) and those that have a local structure where statistical modelling approaches are inappropriate. This poses the question : can a model be found that unifies these 2 viewpoints? Indeed, in what sense does the human visual system relate differently to statistical and structural texture? These questions are certainly worthy of further study.

Chapter 3 described an image segmentation scheme as a precursor to a full texture segmentation procedure. The segmentation combines assumptions about the statistical nature of classes as defined in feature space with spatial assumptions concerning the properties of regions, each point of which is assigned to the same class. This has the advantage over techniques which operate exclusively in either the spatial domain or in feature space, for example boundary detection and histogram thresholding respectively,

because it exploits the constraints implicit in the definition of a region as defined in feature space and spatially. It was shown how this leads to high resolution in terms of class conditional overlap whilst maintaining high spatial resolution at region boundaries. The constraints inherent in the problem, as described by the uncertainty principle [40], are overcome by assuming consistency of region properties over a range of spatial resolutions [55].

A necessary part of the process is a statistical clustering algorithm at a low spatial resolution. The clustering assumes that the classes are characterised statistically as consisting of a high concentration of feature vectors around the class mean. However, it assumes no particular model and requires no a priori information as to the number of classes. This gives it a considerable advantage over existing techniques.

The clustering is performed after a fixed number of quadtree smoothing operations, the number chosen to provide a compromise between good smoothing gain and the ability to detect small regions. This aspect of the procedure implies that there is no measure of local adaption in terms of the spatial resolution/class resolution trade-off. Specifically a very small region with a relatively large object-background signal to noise ratio would be lost in the

smoothing process whilst visually it would be apparent. Hence, whilst classes are defined in terms of the statistical distribution of features, it appears that purely global statistical processes designed to pick out classes characterising each region are inappropriate. (It should be noted that the above problem could be resolved by combining results obtained from different quadtree heights. However, this could lead to ambiguous results when a pair of regions requiring different amounts of smoothing have a common boundary.)

In view of the above discussion, a fruitful avenue of future research would be an extension of the segmentation scheme so that it can handle cases where different smoothing gains are required at different image locations. This would require a more sophisticated data structure than the simple quadtree which performs the same operation at each point. In particular the smoothing/sub-sampling operations would be controlled by the results of local processing at each level in the data structure [64].

Chapter 4 described a set of experiments designed to test the performance of the quadtree segmentation. In particular, the superiority of the technique over a standard Bayesian classifier is demonstrated even though the latter classifier relies heavily on a

priori knowledge as to the statistical specification of the classes. This superiority is not surprising since Bayesian classification is a purely statistical operation on the histogram of the data.

It was also noted how, by using noise adaptive smoothing on the boundary estimation part of the procedure, the perceptual appearance of region boundaries correspond to the boundaries obtained by the scheme. Although quadtree segmentation is not intended to be a paradigm of human visual processing in this context, local noise adaptive smoothing is likely to be involved.

In chapter 5 a texture segmentation procedure consisting of 2 parts was described. The first part is a feature extraction process and the second involves classification of the generally noisy feature space. This classification is an extension of the quadtree segmentation to multi-dimensional signals.

In order that the classification is able to find the texture boundaries with high accuracy, the feature extraction has to produce high resolution in feature space in terms of the overlap in the class conditional distributions whilst maintaining the locality essential if accurate boundary estimation is to occur.

It is assumed that the textures can be discrim-

inated on the basis of their Fourier spectra, this assumption being related to the ideas of phase independence discussed in chapter 2. A filter design procedure was adopted which maximises the energy constrained within a given region spatially for filters with truncated spatial frequency responses. By using the envelope of the outputs of analytic filters, at each image point a quantity is obtained which is related to the energy of the input signal in that part of the spectrum spanned by the passband of the filter. In this sense the procedure can be regarded as one of local spectrum estimation, the locality being guaranteed by having minimally sized filter impulse responses.

This method is different from those that compute textural features by integrating the signal energy over pre-defined regions of the Fourier domain. In particular such techniques are based on a windowed global Fourier transform. Spatial locality can be introduced by windowing the signal prior to the integration of the energy of the transformed windowed signal. However, this involves a loss in spatial frequency resolution as the spectrum of the signal is now convolved with that of the spatial window.

The tessellation of the spatial frequency domain into square regions of constant relative bandwidth is

consistent with the results of psycophysical experimentation, for example [65]. Such a logarithmic arrangement of the frequency responses implies that resolution decreases with increasing spatial frequency. This is not surprising in view of the nature of most naturally occurring images whose power spectra generally tail off at high frequency thus removing the need for a high density of detectors at these frequencies.

In order to extend the feature extraction process to more frequency scales thus increasing the dimensionality of the data, resort would have to be made to feature rejection. It was demonstrated that when clustering is to be performed on the joint histogram of the data, feature rejection based on the eigenvector transformation is inappropriate in some circumstances. A topic of future research could be to investigate this problem and in particular to develop a procedure that takes into account the spatial distribution of data within classes as well as the statistical nature of the classes themselves.

Chapter 6 illustrated the results of running the overall texture segmentation scheme on a variety of synthetic and natural textures and it was noted that satisfactory results were obtained in each case. Also, the benefit of combining data into a single feature

vector defined at each image location instead of operating on a set of vectors each of lower dimensionality and combining the results was noted. However, in spite of this the human visual system is unlikely to combine information in this way. An interesting question and one worthy of further study is how the visual system combines independent features within a single region in order to segment that region.

APPENDIX

Pattern Recognition Vol. 18, Nos 3/4, pp. 257–269, 1985.
Printed in Great Britain.

0031-3203/85 $3.00 + .00
Pergamon Press Ltd.

A QUAD-TREE APPROACH TO IMAGE SEGMENTATION WHICH COMBINES STATISTICAL AND SPATIAL INFORMATION

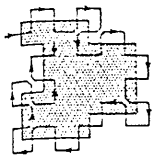
M. SPANN and R. WILSON
University of Aston in Birmingham, Gosta Green, Birmingham B4 7ET, U.K.

(*Received* 21 *August* 1984; *in revised form* 30 *October* 1984; *received for publication* 8 *November* 1984)

Abstract—A new approach to the problem of image segmentation is presented. By combining a non-parametric classifier, based on a clustering algorithm, with a quad-tree representation of the image, the scheme is both simple to implement and performs well, giving satisfactory results at signal-to-noise ratios well below 1. The results of an analysis of the algorithm are borne out by a comprehensive set of tests on Gaussian images and synthetic textures, which demonstrate its principal features.

Segmentation Quad-tree Texture Region description Boundary estimation

4th SCIA



A MULTI-RESOLUTION APPROACH TO IMAGE SEGMENTATION

M. Spann, Research Assistant — University of Aston in Birmingham — England
R. G. Wilson, Lecturer — University of Aston in Birmingham — England

Abstract

This paper describes a multi-resolution approach to image segmentation. A clustering algorithm operating on a low spatial resolution representation of the image is combined with a boundary estimation procedure at successively higher spatial resolutions to yield a segmentation at full resolution. Results are presented which demonstrate good performance on a set of Gaussian images and synthetic textures for inter-region signal to noise ratios well below 1.

REFERENCES

REFERENCES

[1] Hawkins J.K. Textural Properties for Pattern Recognition. Picture Processing and Psycophysics. Ed. B.S. Lipkin, A. Rosenfeld. New York Academic Press 1970.

[2] Witkin A.P. Recovering Surface Shape and Orientation from Texture. Artificial Intelligence, Vol. 17, 1981.

[3] Jain A.K. Image Data Compression: A Reveiw. Proc. IEEE, Vol. 69, 1981.

[4] Julesz B. Visual Pattern Discrimination. IRF Trans. Information Theory, Vol. 8, Feb. 1962.

[5] Julesz B. Textons, the Elements of Texture Perception and their Interactions. Nature, Vol. 290, March 1981.

[6] Beck J. Textural Segmentation. Organization and Representation in Perception. Ed. J. Beck, L. Erlbaum Assoc., 1982.

[7] Gagalowicz A. A New Method of Texture Field Synthesis: Some Applications to the Study of Human Vision. IEEE Trans. PAMI., Vol. 3, No. 5, Sept. 1981.

[8] Caelli T. On Discriminating Visual Textures and

Images. Perception and Psycophysics, Vol. 31, No. 2, 1982.

[9] Harvey L.O., Gervais M.J. Visual Texture Perception and Fourier Analysis. Perception and Psycophysics, Vol. 24, No. 6, 1978.

[10] Eklundh J.O. On the use of Fourier Phase Features for Texture Discrimination. Computer Graphics and Image Processing, Vol. 9, 1979.

[11] Haralick R.M. Statistical and Structural Approaches to Texture. Proc. IEEE, Vol. 67, No. 5, 1979.

[12] Schacter B., Ahuja N. Random Pattern Generation Processes. Computer Graphics and Image Processing. Vol. 10, 1979.

[13] Ahuja N., Rosenfeld A. Mosaic Models for Textures. IEEE Trans. PAMI., Vol. 3, No. 1, Jan. 1981.

[14] Modestino J.W., Fries R.W. Construction and Properties of a Useful Two-Dimensional Random Field. IEEE Trans. Information Theory, Vol. 26, Jan. 1980.

[15] Zucker S.W. Towards a Model of Texture. Computer Graphics and Image Processing, Vol. 5, 1976.

[16] Haralick R.M. Textural Features for Image Classification. IEEE Trans. SMC., Vol. 3, Nov. 1973.

[17] Weszka J.S., Dyer C.R., Rosenfeld A. A Compara-

tive Study of Texture Measures for Terrain Classification. IEEE Trans. SMC., Vol. 6, April 1976.

[18] Galloway M.M. Texture Analysis Using Grey Level Run Lengths. Computer Graphics and Image Processing, Vol. 4, June 1975.

[19] Chen C.H. A Study of Texture Classification Using Spectral Features. Non-Linear Maximum Entropy Spectral Analysis Methods for Signal Recognition. John Wiley and Sons, 1982.

[20] Pratt W.K., Faugeras O., Gagalowicz A. Applications of Stochastic Texture Field Models to Image Processing. Proc. IEEE, May 1981.

[21] Deguchi K., Morishita I. Texture Characterization and Texture Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques. IEEE Trans. Computers, Vol. 27, August 1978.

[22] Kashyap R.L., Lapsa P.M. Synthesis and Estimation of Random Fields Using Long Correlation Models. IEEE Trans. PAMI., Vol. 6, Nov. 1984.

[23] Duda R.O., Hart P.E. Pattern Classification and Scene Analysis, Wiley Interscience, 1973.

[24] Rosenfeld A. Image Analysis: Problems, Progress and Prospects. Pattern Recognition, Vol. 17, No. 1, 1984.

[25] Modestino J.W., Fries R.W., Vickers A.L. Texture Discrimination Based on an Assumed Stochastic Texture Model. IEEE Trans. PAMI., Vol. 3, No. 5, Sept. 1981.

[26] Rosenfeld A., Thurston M. Edge and Curve Detection for Visual Scene Analysis. IEEE Trans. Computers, Vol. 20, No. 5, May 1971.

[27] Wermser D., Liedtke C. Texture Gradient: A New Tool for the Unsupervised Segmentation. Proc. Third Scandinavian Conf. on Image Analysis, Copenhagen, 1983.

[28] Coleman G.B., Andrews H.C. Image Segmentation by Clustering. Proc. IEEE, Vol. 67, 1979.

[29] Fukunaga K. Introduction to Statistical Pattern Recognition. Academic Press, 1979.

[30] Rosenfeld A. Pyramids: Multi-Resolution Image Analysis, Proc. Third Scandinavian Conf. on Image Analysis, 1983.

[31] Burt P.J., Hong T., Rosenfeld A. Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation. IEEE Trans. SMC., Vol. 11, No. 12, Dec. 1981.

[32] Papoulis A. Probability, Random Variables and Stochastic Processes. McGraw-Hill, 1965.

[33] Middleton D. Introduction to Statistical Communication Theory. McGraw-Hill, 1960.

[34] Papoulis A. Signal Theory. McGraw-Hill, 1977.

[35] Gardner W.A., Frank L.E. Characterization of Cyclostationary Random Signal Processes. IEEE Trans. Information Theory, Vol. 21, Jan. 1975.

[36] Thomson D.J. Spectrum Estimation and Harmonic Analysis. Proc. IEEE, Vol. 70, Sept. 1982.

[37] Oppenheim A.V., Lim J.S. The Importance of Phase in Signals. Proc. IEEE, Vol. 69, May 1981.

[38] Hayes M.H., Lim J.S., Oppenheim A.V. Signal Reconstruction from Phase or Magnitude. IEEE Trans. ASSP., Vol. 28, Dec. 1980.

[39] Brodatz P. A Photographic Album for Artists and Designers. New York: Dover 1956.

[40] Wilson R.G., Granlund G.H. The Uncertainty Principle in Image Processing. IEEE Trans. PAMI., Vol. 6, No. 6, Nov. 1984.

[41] Dahlqvist B. Segmentation of Cell Images by Minimum Error Thresholding. Proc. 2nd Scandinavian Conference on Image Analysis, Helsinki, Finland, 1981.

[42] Panda D.P. Segmentation of FLIR Images by Pixel Classification. University of Maryland Computer Cen-

tre, TR-508, Feb. 1977.

[43] Pavlidis T. Structural Pattern Recognition, Springer New York, 1977.

[44] Ball G.H., Hall D.J. ISODATA, a Novel Method of Data Analysis and Pattern Classification. Stanford Res. Inst., Menlo Park, CA., April 1965.

[45] Wilson R.G. From Signals to Symbols. The Inference Structure of Perception. Proc. COMPINT '85, Montreal, 1985.

[46] Wilson R.G. The Uncertainty Principle in Vision. Int'l Rept. LiTH-ISY-I-0581, Linkoping University, Sweden.

[47] Marr D., Hildreth E. Theory of Edge Detection. Proc. R. Soc. of London, B 207, 1980.

[48] Marr D. Vision. W.H. Freeman, San Francisco, 1981.

[49] Tanimoto S.L., Pavlidis T. A Hierarchical Data Structure for Picture Processing, Comput. Graphics and Image Proc., 4, 1975.

[50] Hong T.H., Rosenfeld A. Compact Region Extraction Using Weighted Pixel Linking in a Pyramid. IEEE Trans. PAMI., Vol. 6, No. 2, March 1984.

[51] Pratt W.K. Digital Image Processing. Wiley Inter-

science, 1978.

[52] Spann M., Wilson R.G. A Quadtree Approach to Image Segmentation Which Combines Statistical and Spatial Information. Pattern Recognition, No. 3, 1985.

[53] Spann M., Wilson R.G. A Multiresolution Approach to Image Segmentation. Proc. 4th Scandinavian Conf. on Image Analysis, Trondeim, Norway, 1985.

[54] Wilson R.G. A Class of Local Centroid Algorithms for Classification and Quantization in Spaces of Arbitrary Dimensions. Int'l Rept. LiTH-ISY-I-0610, Linkoping University, Sweden.

[55] Witkin A.P. Scale Space Filtering. Proc. ICASP, San Diego, March 1984.

[56] Therrien C.W. Linear Filtering Models for Texture Classification and Segmentation. Proc. 5th Int. Conf. on Pattern Recognition, Miami, Dec. 1980.

[57] Mitiche A. Davis L. Theoretical Analysis of Edge Detection in Textures. Proc. 5th Int. Conf. on Pattern Recognition, Miami, Dec. 1980.

[58] Knuttson H.E. Filtering and Reconstruction in Image Processing. PhD. Thesis, Linkoping University, Sweden, 1982.

[59] Knuttson H.E., Wilson R.G., Granlund G.H. Aniso-

tropic Non-Stationary Image Estimation and Its Applications: Part 1 - Restoration of Noisy Images. IEEE Trans. Comm., Vol. 31, March 1983.

[60] Wilson R.G. Uncertainty, Eigenvalues Problems and Filter Design. Int'l Rept. LiTH-ISY-I-0580, Linkoping University, Sweden.

[61] Savoji M.H. A Unifying View of the Different Methods Based on the Karhunen-Loeve Expansion Used in Image Processing and Pattern Recognition. Proc. Third Scandinavian Conf. on Image Analysis, Copenhagen, 1983.

[62] Luke J.L. The Special Functions and Their Approximations. Vol. 1. Mathematics in Science and Engineering. New York and London Academic Press, 1969.

[63] Wilson R.G. Texture Synthesis and Analysis Using Minimum Uncertainty Filters. Proc. 1st Image Proc. Symp. Biarrits, May 1984.

[64] Granlund G.H. In Search of a General Picture Processing Operator. Computer Graphics and Image Processing, Vol. 8, 1978.

[65] Caelli T., Moraglia G. On the Detection of Gabor Signals and Discrimination of Gabor Textures. Vision Res., Vol. 25, No. 5, 1985.