5-1-1991

Pixel classification by morphological granulometric features

John T. Newell

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Pixel Classification by Morphological Granulometric Features

by

John T. Newell, III

Rochester Institute of Technology
Center for Imaging Science

May 1, 1991

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Center for Imaging Science in the College of Graphic Arts and Photography of the Rochester Institute of Technology

Signature of Author

Center for Imaging Science

Approved by

Mendi Vaez-Pravani
Coordinator, M. S. Degree Program
I, John T. Newell, III, hereby grant permission to the Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date 6/18/91
Abstract

Pixel classification systems rely on a certain set of features that are sufficient to classify a given pixel into a class defined within a database. Unlike brightness and spectral signature features commonly used in remote sensing applications, texture-based features cannot be defined for a single pixel and must be derived from an area or window surrounding that pixel. In this research, all features are derived from binary morphological granulometries. Once generated, these features comprise a database which can be used to classify images. A Gaussian Maximum Likelihood Classifier is trained with this data base for subsequent classification of both dependent and independent data. Several aspects of these texture-base features require investigation in order to determine their ability to distinguish image textures. Three important aspects are addressed in this study; the effects of maximum noise, the optimal size of the localized window, and the minimum number of optimal features required for accurate classification. A statistical approach has been taken to determine the classification accuracy with varying window size, varying number of features, and varying amounts of four types of maximum noise using granulometric features. Analysis of these investigations indicate four main results. First, classification accuracy in the absence of noise is extremely high. Second, for these textures at the spatial resolution of 75 dpi, classification accuracy decreases dramatically below a window size of 11x11 pixels. Third, the number of features needed for high classification accuracy can be reduced to a fairly small number on the order of 6 features. Finally, these features are generally robust in the presence of maximum noise if the type and amount of noise can be accurately estimated.
Acknowledgements

I would like to thank Jeff B. Pelz for his help and computer programs which were of tremendous help in completing this thesis even though they were written in Pascal.

I would also like to thank Wendy Rosenblum for her algorithms, code and a fine example of a well written thesis.

Finally, I wish to thank Kaleen Moriarty for her immeasurable friendship which has seen me through the heaven and hell of RIT. Thanks for keeping me smiling through the tough times.
This thesis is dedicated to John T. Newell, Jr. and Anne W. Newell for their love, confidence and support throughout my college education and for instilling in me the pride and confidence to complete any task undertaken.
# Table of Contents

Table of Contents vii  
List of Figures ix  
List of Tables xi  

1.0 Introduction 1  

1.1 Morphological Granulometries 1  
1.1.1 Opening 1  
1.1.2 Granulometries 3  
1.1.3 Local Granulometries 7  

1.2 Image Texture 9  

1.3 Image Segmentation Using Granulometric Features 10  
1.3.1 Segmentation 10  
1.3.2 Use of Granulometric Feature for Segmentation 10  
1.3.3 Higher Order Moment Features 11  
1.3.4 Structuring Elements and Derivation of Other Granulometric Features 11  

1.4 Image Classification and Discriminant Analysis 13  
1.4.1 Classification 13  
1.4.2 Feature Probability Distributions 14  
1.4.3 Maximum Likelihood Classification 15  
1.4.4 Gaussian Maximum Likelihood Classification 16  

1.5 Minimal Window Size 19  

1.6 Optimal Feature Selection 20  
1.6.1 Feature Reduction 20  
1.6.2 Mahalanobis-Like Distance Measure 20  
1.6.3 Divergence Measure 22  
1.6.4 Separability Measure 23
1.7 Noise
  1.7.1 Maximum Noise
  1.7.2 Point Noise
  1.7.3 Occlusion Noise
  1.7.4 Scratch Noise
  1.7.5 Spaghetti Noise

2.0 Statement of Work
  2.1 Selection of Texture Images
  2.2 Thresholding of Texture Images
  2.3 Generation of Noise
  2.4 Generation and Selection of Local Granulometric Features
  2.5 Classification of Dependent and Independent Data

3.0 Analysis of Results
  3.1 Dependent Classification
  3.2 Independent Classification
  3.3 Minimal Window Size Determination
  3.4 Optimal Feature Selection
  3.5 Classification with Maximum Noise
    3.5.1 Dependent Classification
    3.5.2 Independent Classification
    3.5.3 Combinations of Noise Models
  3.6 Noise Estimation

4.0 Conclusions
  4.1 Suggestions for Future Work

5.0 References

Appendix A
Appendix B
List of Figures

Figure 1: Image $S$ and structuring element $E$ ................................. 2
Figure 2: $\text{Open} (S,E)$; Opening of image $S$ by structuring element $E$ 2
Figure 3: Simulated binary granulometry resultant images .................. 5
Figure 4: $\Psi(k)$, $\Phi(k)$ and $d\Phi(k)$ from the simulated image granulometry 6
Figure 5: Example $d\Phi_x(k)$ probability distribution .......................... 8
Figure 6: Feature $Z$ value distribution for two classes ......................... 14
Figure 7: Maximum likelihood decision boundary ............................... 16
Figure 8: Feature sets for class separability ..................................... 24
Figure 9: Texture images ............................................................... 31
Figure 9: Binary texture images ...................................................... 34
Figure 10: Examples of binary noise images ..................................... 40
Figure 12: Classification Accuracy vs. Window Size ............................. 50
Figure 13: Classification Accuracy vs. Number of Optimal Features ........ 53
Figure 14: Classification Accuracy vs. $\%$ Point Noise ....................... 58
Figure 15: Classification Accuracy vs. $\%$ Spaghetti Noise .................... 58
Figure 16: Classification Accuracy vs. $\%$ Occlusion Noise ................... 59
Figure 17: Classification Accuracy vs. $\%$ Scratch Noise ....................... 59
Figure 18: Classification Accuracy in the presence of Horizontal, Fixed and Random Scratch Noise ......................................................... 62
Figure 19: Classification Accuracy with combinations of noise models .... 64
Figure 20: Feature distributions for Circular PSSD .............................. 67
Figure 21: Feature distributions for Negative Diagonal PSSD .................. 67
Figure 22: Feature distributions for Positive Diagonal PSM 68
Figure 23: Probability distributions for Circular PSSD 69
Figure 24: Probability distributions for Negative Diagonal PSSD 69
Figure 25: Probability distributions for Positive Diagonal PSM 70
Figure 26: Optimal Feature Classification in Point Noise 73
Figure 27: Optimal Feature Classification in Spaghetti Noise 73
Figure 28: Optimal Feature Classification in Occlusion Noise 74
Figure 29: Optimal Feature Classification in Scratch Noise 74
Figure 30: Classification Accuracy with Noise Estimation 77
**List of Tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Classification of dependent data</td>
<td>46</td>
</tr>
<tr>
<td>Table 2</td>
<td>Classification of independent data</td>
<td>48</td>
</tr>
<tr>
<td>Table 3</td>
<td>Classification of independent data using pooled covariance</td>
<td>49</td>
</tr>
<tr>
<td>Table 4</td>
<td>Optimal Feature Sets using Rosenblum Optimization</td>
<td>55</td>
</tr>
<tr>
<td>Table 5</td>
<td>Classification of dependent texture-plus-noise data</td>
<td>56</td>
</tr>
<tr>
<td>Table 6</td>
<td>Classification of independent data in 5% point noise</td>
<td>66</td>
</tr>
<tr>
<td>Table 7</td>
<td>Classification of independent data in 10% point noise</td>
<td>66</td>
</tr>
<tr>
<td>Table 8</td>
<td>Classification of data with 10% point noise after training with 5% point noise</td>
<td>76</td>
</tr>
<tr>
<td>Table 9</td>
<td>Classification of data with 5% point noise after training with 10% point noise</td>
<td>76</td>
</tr>
</tbody>
</table>
1.0 Introduction

1.1 Morphological Granulometries

Morphological granulometries were conceived by Matheron [1975] as a type of "sieving" operation for binary images in which particles in the image structure are filtered according to their size. Quantification of the rate at which an image is altered in the sieving process produces a numerical size distribution containing image texture information. Binary granulometries are generated by successively opening a binary image by an increasing sequence of convex binary structuring elements. The images which make up a set the structuring element sequence are of a specific shape (i.e. line circle, square, etc.) and the textural information which can be gathered from a granulometry is specific to the shape of the structuring element sequence.

1.1.1 Opening

The opening of a binary image $S$ by a binary structuring element $E$ is defined to be the union of all translations of $E$ which are subsets of $S$. Rigorously, $x \in \text{OPEN}(S,E)$ if and only if there is some translate $(E+z)$ of $E$ such that $x \in (E+z) \subset S$. Consider the example of a binary digital image $S$ and the three pixel horizontal structuring element $E$ represented in Figure 1.
Figure 1: Image S and structuring element E.

The ones represent activated pixels and the stars are considered undefined or non-activated pixels. All pixels outside an image are also considered non-activated. To open image S by structuring element E, the origin of E is translated to each pixel in S. Wherever E entirely fits over activated pixels in S, all pixels in the resulting image $\text{Open}(S,E)$ are activated. See Figure 2.

$$
\begin{array}{cccccccc}
* & 1 & 1 & 1 & * & * & * & 1 \\
* & * & * & * & * & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & 1 & 1 & 1 & * & 1 & * & 1 \\
1 & 1 & 1 & 1 & * & 1 & * & 1 \\
1 & 1 & * & 1 & 1 & * & 1 & 1 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 1 & 1 & \\
\end{array}
$$

Figure 2: $\text{Open}(S,E)$; Opening of image S by structuring element E

Since E will fit over all pixels in the third row, the entire row is activated in $\text{Open}(S,E)$. Notice that the last pixel in the first row is activated in image S but not in image $\text{Open}(S,E)$ since the E will not fit over that pixel. The same is true for all pixels in the forth and sixth...
Because of the size and the shape of the structuring element in this particular example, any horizontal run length of 3 or more pixels will be activated.

1.1.2 Granulometries

From the definition of an opening, it follows that when OPEN(F,E) = F, OPEN(S,F) is a subimage of OPEN(S,E). As a result, if E₁, E₂, E₃, ... is an increasing sequence of structuring elements such that OPEN(Eₖ₊₁, Eₖ) = Eₖ₊₁, then the filtered images form a decreasing sequence

$$\text{OPEN}(S,E₁) \supset \text{OPEN}(S,E₂) \supset ...$$

Counting the number of pixels remaining in each succeeding opening results in a decreasing function $Ψ(k)$, such that for some $K$, $Ψ(k) = 0$ for $k ≥ K$. Depending on the shape of the structuring elements, various textural information is revealed by studying the function $Ψ(k)$. The image sequence $\{\text{OPEN}(S,Eₖ)\}$ is called a granulometry and the resulting function $Ψ(k)$ is called the size distribution. In practice, $E₁$ consists of a single pixel so that $Ψ(1)$ gives the total number of activated pixels in $S$.

Since $Ψ(k)$ is decreasing, the normalization of $Ψ(k)$ is a probability distribution function given by Equation 1.

$$Φ(k) = 1 - Ψ(k) / Ψ(1) \quad (1)$$
The discrete derivative, \( d\Phi(k) \), is a discrete probability density function. It has become popular to refer to this normalized granulometric-size distribution density as the \textit{pattern spectrum} of the images. This distribution reveals the particle size distribution of the image from which it is calculated and can be described by its moments. The moments of the pattern spectrum can then be used to describe texture information.

Figure 3a shows a simulated binary image made up of four size disks of diameters 4, 7, 15, and 31 pixels. When the image is opened with a series of circular structuring elements \( E_k \) of diameter 1 through 4, the resultant image is unchanged. However, when the image is opened with a circular structuring element of diameter 5, the disks of diameter 4 are filtered out of the image, leaving the image shown in Figure 3b. Opening this image with elements of diameters 6 and 7 produce no further change in the output images. When the image is opened with diameter 8, the disks of diameter 7 are filtered out of the image resulting in Figure 3c. Again, there is no change in the output image until the structuring element diameter reaches 16 pixels and the disks of diameter 15 are filtered out as shown in Figure 3d. Finally, when the structuring element sequence reaches 32, all the disks have been filtered out resulting in a null image. (It should be noted that there may be some digitization error introduced when using real digital images.)
Figure 3: Simulated binary granulometry resultant images

a) original and OPEN(S,E_1) through OPEN(S,E_4)
b) OPEN(S,E_5) through OPEN(S,E_7)
c) OPEN(S,E_8) through OPEN(S,E_{15})
d) OPEN(S,E_{16}) through OPEN(S,E_{31})
Figure 4: $\Psi(k)$, $\Phi(k)$ and $d\Phi(k)$ from the simulated image granulometry
The $\Psi(k)$, $\Phi(k)$ and $d\Phi(k)$ distributions from the simulated image granulometry are shown in Figure 4. All three parameters are functions of the diameter, $k$, of the circular structuring elements. It is important to note that these distributions are based on a pixel count of the filtered image, rather than a particle count.

### 1.1.3 Local Granulometries

A local granulometry is an extension of this concept describing the particle size distribution in a given neighborhood or window about some pixel $x$. $\Psi_x(k)$ is then the pixel count within a window centered on pixel $x$, rather than the pixel count over the entire image. In order to maintain large-scale textural information, the image is opened globally and the pixel count is performed locally. In the same manner described for a global granulometry, the normalized probability distribution $\Phi_x(k)$ is calculated from the local size distribution

$$\Phi_x(k) = 1 - \Psi_x(k) / \Psi_x(1)$$

for each point $x$ in the image. The discrete derivative, $d\Phi_x(k)$, defines the probability density about the pixel $x$. $d\Phi_x(k)$ is then the *local pattern spectrum* at $x$. The result of the binary local granulometry with a given window size is a one-dimensional probability density at each $(i,j)$ pixel location in the image. These probability densities serve as valuable descriptors of the image texture surrounding each pixel location. An example of $d\Phi_x(k)$ probability distribution from a local area of a real image is given in Figure 5.
Figure 5: Example $d\Phi_x(k)$ probability distribution

This distribution is a robust, but impractical descriptor of the local texture. However, the moments can be used to describe the distribution and can be used as a much more practical descriptor of the local image texture. The local granulometric mean, standard deviation, variance and skewness can be used as valuable texture descriptors for image segmentation and classification [Dougherty et al., 1990]. Since these moments are derived from random functions, the moments are random variables.
1.2 Image Texture


Texture is a description of the spatial distribution and spatial dependence among the grey tones [Rosenblum, 1990]. It can be described by perceptual descriptors such as "fine", "smooth", "coarse", "mottled", "lineated" or "irregular". It may also be described in terms of a pattern made up of repeated texture primitives [Nevatia, 1982]. A texture image $J$ can therefore be thought of as a transform from one band of a spectral image $I$ in which $J(i,j)$ is a function of $I(i,j)$ and neighboring pixels [Haralick, 1979]. A texture measure at a point of an image is some function of the observed values within a local neighborhood about the point [Ahuja, 1983]. Granulometries use a structural approach to analyze visual scenes in terms of organization and relationships among its substructures [Haralick, 1986]. Granulometric features describe image textures in terms of the size distributions of the textural substructures.
1.3 Image Segmentation Using Granulometric Features

1.3.1 Segmentation

One of the reasons behind the development of image processing has been the need to identify different objects or regions within a given image. Within the study of image texture has been the development of algorithms for segmentation based on image texture. The intuitive idea behind image segmentation is to divide the image into segments such that each segment is homogeneous in some sense and two neighboring segments differ from one another in the same sense [Kashyap, 1986]. Segmentation is accomplished by separating two or more homogeneous regions which have a significant statistical difference. Since the pixel values of a binary texture regions are inherently non-homogeneous, texture measures need to be assigned to each pixel for subsequent segmentation.

1.3.2 Use of Granulometric Feature for Segmentation

Dougherty and Pelz [1989] developed both a deterministic and a nondeterministic model of image segmentation using texture measures derived from morphological granulometries. Using the deterministic model, an image comprised of two different size discs was segmented by using the mean of the local circular granulometry. A granulometric-mean image was generated by assigning this local circular pattern spectrum mean (PSM) to each point x of an image. Each pixel in the resulting image was therefore a measure of the local texture. The local granulometric mean can therefore be thought of as texture-dependent.
This grey-scale image of mean values was then successfully segmented by thresholding the image.

1.3.3 Higher Order Moment Features

If the local PSM of two texture regions is not sufficiently different to allow segmentation, higher order moments of the local pattern spectrum may be employed. The local pattern spectrum standard deviation (PSSD), variance (PSV), and skewness (PSS) can be used as texture measures to segment an image. Dougherty and Pelz [1989] used the pattern spectrum variance (PSV) to segment an image in which two texture regions had similar PSMs. By viewing a homogeneous texture as a population of pixels, all local granulometric moments can be interpreted as realizations of random variables which are characteristic of the image texture. These random variables possess probability distributions indicative of an image texture.

1.3.4 Structuring Elements and Derivation of Other Granulometric Features

In addition to being texture dependent, all moments of a pattern spectrum are specific to the structuring elements used to generate the spectrum. Many different types of structuring elements have been used to generate a pattern spectrum of an image. Circular, elliptical and linear structuring elements have commonly been employed to generate granulometric texture measures. Linear and non-linear combinations of the moments can also be used as local texture measures. Dougherty, Kraus, and Pelz [1989] introduced three such combinations; AveLin, MaxLin and Linearity. AveLin was defined as the average PSM of
four linear granulometries; horizontal, vertical, positive-diagonal (45°) and negative-diagonal (135°), and MaxLin as the maximum PSM of the same four linear granulometries. AveLin is an example of a linear combination whereas MaxLin is an example of a non-linear combination. Linearity is a scale-invariant feature defined as MaxLin divided by the PSM of the circular granulometry. This ratio will result in a value of 1 for any circular image element regardless of diameter. Elongated image elements will produce higher values of linearity. AveLin, MaxLin and Linearity are all rotationally invariant at 45° increments.
1.4 Image Classification and Discriminant Analysis

1.4.1 Classification

Image classification is the process of assigning a pixel to one of a number of possible classes on the basis of some observations made on features of that pixel and/or its surround. It is a decision making process which uses statistical decision theory to make an intelligent estimate of the class to which a pixel belongs [Schowengerdt, 1983]. In supervised classification, a sample of each class is taken for each observation and the statistical distribution of the elements in each class are analyzed. From that information, the classification algorithm reaches a decision about how to assign pixels not in the sample to the appropriate classes. Schowengerdt [1983] recommends that 10 to 100 pixels be included per training class with more pixels for those classes with higher variability.

Pixel classification has long been an integral part of remote sensing and other image processing applications. Spectral and radiometric data from aerial and satellite images have been used as features to classify specific regions of these images according to some predescribed criterion. Linear combinations of this data, such as the red to green band ratio in multispectral aerial images, have also been found to emphasize differences in ground cover types and characteristics of particular interest. Many approaches to classification based on texture have been developed over the years. Some of these approaches include the use of features derived from first order statistics, spectral power density functions, autocorrelation functions, and grey-tone run-length distributions. One of the most successful classification schemes uses grey-tone co-occurrence matrix features which
measure the relative frequencies with which two pixel values, with a certain separation, occur in an image [Haralick and Anderson, 1971].

1.4.2 Feature Probability Distributions

Any number of these or other features can be used to classify an image into some preset number of classes. Each class will have the same number of features in a feature vector. The distribution of values for any one feature for a given class has a certain distribution which can be used to decide the classification of an unknown pixel. Consider the two distributions of some feature $Z$ in Figure 6. Each distribution corresponds to a separate class.

![Feature Z value distribution for two classes](image)

The area under each of these distribution curves is normalized to 1.0 and they are assumed to approximate the feature probability density functions of each class. These functions can then be used to determine the probability that some unknown pixel with feature $Z$ value, $x$, ...
belongs to class i. The probability of finding a feature value of x given that we are sampling from class i is given as \( p(x \mid i) \). The discriminant function is defined as the probability of a pixel belonging to class i given that it has a feature value x or \( p(i \mid x) \).

\[
p(i \mid x) = \frac{p(x \mid i) \ p(i)}{p(x)}
\]  

where \( p(i) \) is the \textit{a priori} probability that class i exists in the image and \( p(x) \) is the probability of finding a pixel from any class.

Assuming that each class has an equal probability of occurring, \( p(i) \) will be equal for each class. The value for \( p(x) \) is simply a normalizing factor and therefore a constant for each class. The discriminant function is then simply a calculation of \( p(x \mid i) \) for each class.

1.4.3 Maximum Likelihood Classification

Maximum likelihood classification compares the discriminant function value for each feature value x calculated for each class and assigns the pixel to the class which produces the highest probability value. For example, consider a pixel with a feature Z value of x, as shown in Figure 6. Since the calculated value of the discriminant function is greater for class 1 than for class 2, the pixel would be assigned or classified into class 1.

The decision boundary for classification, d, lies at the point at which the two distributions cross as shown in Figure 7.
Any pixel with a feature $Z$ value less than $d$ would be assigned to class 1, since there is a higher probability of this value coming from class 1 than coming from class 2. Likewise, any pixel with a feature $Z$ value greater than $d$ would be assigned to class 2. The total error in this classification is represented by the overlap of the two distributions which is shown as the shaded region. This error is minimized by placing the decision boundary at the point at which $p(x \mid 1)$ is equal to $p(x \mid 2)$. This decision boundary is represented by $d$ in Figure 7. Notice that if the decision boundary was moved in either direction, the error would increase.

1.4.4 Gaussian Maximum Likelihood Classification

This is the simplest example of classification since there is only one feature and only two classes. However, the same principles can be extended to more complicated classification models in which there are any number of features and classes. The most commonly used multivariate classifier is the Gaussian maximum likelihood classifier. Estimates of the
mean vectors and covariance matrices of the classes are required to compute the class-
conditional density functions. This classifier requires the distribution of features within
each class to be approximately multivariate normal. However, the classifier is "relatively
tolerant" of deviations from normality [see Swain, 1986].

A discriminant function is developed using a measure of the generalized squared distance
from the mean vector. The classification criterion can be based on either the individual
within-class covariance matrices or a pooled covariance matrix. As with the single variable
case, each unknown pixel is classified into the class from which it has the smallest
generalized squared distance.

The generalized squared distance from \( x \) to class \( t \) is

\[
D_t^2(x) = g_1(x,t) + g_2(t)
\]

where \( x \) is the vector containing the feature values of an unknown pixel and
\( t \) is a subscript to distinguish the classes.

If the within-class covariance matrices are used then

\[
g_1(x,t) = (x - m_t)\, S_t^{-1} (x - m_t) + \ln|S_t|
\]

where \( m_t \) is the vector containing the means of the features of an unknown pixel
and
$S_t$ is the covariance matrix within-class $t$.

If the pooled covariance matrix is used then

$$g_1(x,t) = (x - m_t)' S^{-1} (x - m_t)$$ (6)

where $S$ is the pooled covariance matrix.

If the a priori probabilities are all equal, $g_2(t)$ is zero. However, if they are not all equal

$$g_2(t) = -2\ln(q_t)$$ (7)

where $q_t$ is the a priori probability of class $t$. 
1.5 Minimal Window Size

The size of the window for the local granulometries can have a significant effect on the distributions of the granulometric features. Dougherty, Pelz and Newell [1990] demonstrated that the variance of the granulometric feature distributions decreases with increasing window size. Decreasing the variance decreases the amount of probability overlap between classes so classification accuracy can be improved by increasing the window size. However, increasing the window size can also make it harder to determine the border between adjacent texture regions and lead to misclassification of pixels whose surround includes 2 or more texture classes. Generally speaking, larger windows decrease variability of the granulometric features at the cost of less detailed segmentation.

A feature value from a local granulometry is only an accurate representation of a given image texture if the entire window used to generate that feature lies within the texture region. Otherwise, the feature value can be affected by other image textures and therefore represent a combination of a number of image textures. A pixel lying near the edge of two texture regions should therefore be unclassified. Given two adjacent texture regions in an image, and a window an odd number, x, pixels in length, the number of unclassified pixels in between the two texture regions in the resulting feature image will be (x-1).
1.6 Optimal Feature Selection

1.6.1 Feature Reduction

The key step in any classification problem is the choice of a set of features which reduces the dimension of data to a computationally tractable level while preserving much of the classifying information present in the actual data [Kashyap, 1986]. The number of features used in the classification should still give a minimal probability of mis-classification [Fu, 1976]. Features which do not add to classification accuracy represent a cost since, with a maximum likelihood classifier, the time needed to make a calculation increases quadratically with the addition of features [Richards, 1986].

In recent years there has been much attention paid to determining an optimal set of \( m \) features out of a total set of \( N \) features without significantly degrading the classification ability of the algorithm. These techniques for feature selection attempt to measure the separability of the classes for each combination of \( m \) features out of the total set of features. The subset with the most potential for correct classification is subsequently selected for use in the classifier.

1.6.2 Mahalanobis-Like Distance Measure

The simplest techniques of feature selection use the separation of the feature means in multidimensional space. However, this approach may result in a set of features which are far less than optimal since the in-class variability is discarded. A second approach,
developed by Schott, Salvaggio, and Kraus [1988], uses the Mahalanobis-Like distance as a measure of the separation of classes. The Mahalanobis-Like distance measure is defined as

\[ d_{st} = (\mathbf{m}_s - \mathbf{m}_t)' S^{-1} (\mathbf{m}_s - \mathbf{m}_t) \]  

(8)

where \( \mathbf{m}_s \) and \( \mathbf{m}_t \) are the mean vectors of classes \( s \) and \( t \) and \( S \) is the pooled covariance matrix.

Compared to calculating the exact probability overlap between two classes, this method is very fast since it requires the inversion of only the pooled covariance matrix. Using the pooled covariance has two major drawbacks: 1) the assumption of equal covariance matrices is not usually true and 2) it does not account for the variability of the individual classes. One solution to this problem is to use the individual covariance matrices in place of the pooled covariance matrix [Robert, 1989]. The Mahalanobis-Like distance between classes corrected for the individual covariance matrices takes the form of Equation (9) [Richards, 1986].

\[ d_{st} = [(\mathbf{m}_s - \mathbf{m}_t)' S_t^{-1} (\mathbf{m}_s - \mathbf{m}_t)] + \ln|S_t| \]  

(9)

where \( S_t \) is the covariance matrix within-class \( t \).

However, this introduces other problems since the result depends upon which covariance matrix is used.
1.6.3 Divergence Measure

Richards [1986] describes a way to quantify the separation between two classes by the degree of overlap of the class distributions. The optimal set of features can then be found by finding the set with the least amount of probability overlap. The divergence between two classes, $d_{vst}$, is defined in Equation (10) as a separability measure which takes into account the variability of both classes.

$$d_{vst} = \int \left[ p(X \mid s) - p(X \mid t) \right] \ln \left[ \frac{p(X \mid s)}{p(X \mid t)} \right] dX$$

where $d_{vst}$ is the divergence between class $s$ and class $t$, $p(X \mid s)$ is the probability of finding the feature vector $X$ when sampling from class $s$, and $p(X \mid t)$ is the probability of finding the feature vector $X$ when sampling from class $t$.

If the classes are assumed to come from multidimensional normal distributions, the divergence becomes

$$d_{vst} = \frac{1}{2} \text{Tr} \left[ (S_s - S_t) (S_t^{-1} - S_s^{-1}) \right] + \frac{1}{2} \text{Tr} \left[ (S_t^{-1} - S_s^{-1}) (m_s - m_t)(m_s - m_t)' \right]$$

Equation (11)
where \( m_s \) and \( m_t \) are the mean vectors of classes \( s \) and \( t \) and 
\( S_s \) and \( S_t \) are the covariance matrix within-class \( s \) and \( t \) respectively.

Unlike the Mahalanobis-Like distance measure in Equation (9), \( dv_{st} \) is symmetric (i.e. \( dv_{st} = dv_{ts} \)) because both class distributions are taken into account.

This divergence can then be summed over all class pairs to give a measure of the overall divergence. The set of features which results in the greatest overall divergence should give the greatest classification accuracy when a Gaussian maximum likelihood classifier is used.

Mausel, Kramber and Lee. [1990] transformed the divergence between two classes in the form of Equation (12) to emphasize small changes in the divergence resulting from significant changes in the class separability.

\[
tdv_{st} = 2000 \left[ 1 - \exp\left( -dv_{st} / 8 \right) \right] 
\]  

(12)

This value has a limit of 2000 which was designed to limit extremely high divergence values which do not necessarily correspond to complete class separation.

1.6.4 Separability Measure

Rosenblum [1990] developed a similar method to enhance the accuracy of the Mahalanobis-Like distance separation measure given in Equation (9). The overall separability measure was defined as a sum of the normalized distances separating each class from each of the
other classes. Figure 8 shows two feature sets of the same three classes. In feature set A, classes x and y are poorly separated and class z is greatly separated from these two. In feature set B, all three classes are fairly well separated.

![Feature sets A and B](image)

Figure 8: Feature sets for class separability

It is obvious that feature set B does a better job of separating the three classes than feature set A. However, the separability measure for feature set A will be greater than the separability measure for feature set B. Since a large separation of any one class from a group of others can inflate the overall separability measure for a set of features, a distance threshold was developed to normalize and limit the values. Without that inflation, a set of features which separate all of the classes will be chosen instead of a set which separates one class very well.

The probability of finding the mean of class t in a sample from class s is defined as

\[
P(m|s) = \left[ \frac{1}{|S_s|^{1/2} 2\pi^{k/2}} \right] e^{-1/2[(m_t - m_s)' S_s^{-1} (m_t - m_s)]} \tag{13}
\]
where \( k \) is the number of features which are to be optimized.

Using Equation (13) and assuming multivariate normal class distributions, it can be shown that

\[
[(m_t - m_s)' S_s^{-1} (m_t - m_s)] + \ln|S_s| = -2 \ln[P / 2\pi^{k/2}].
\]  

(14)

(See Appendix A). The left hand side of Equation (14) represents the Mahalanobis-Like distance between two class means, \( d_{ts} \), when the individual class covariance for class \( s \) is used. The right hand side of the equation is the minimum distance, \( d_{\text{thresh}} \), which must separate the two class means for the probability of misclassification to be \( P \).

\[
d_{\text{thresh}} = -2 \ln[P / 2\pi^{k/2}].
\]  

(15)

The value of \( P \) should be set to a sufficiently small value to assure near complete separation of the classes.

There are now two distance measures which need to be calculated: the actual distance as calculated by equation (9),

\[
d_{st} = [(m_s - m_t)' S_t^{-1} (m_s - m_t)] + \ln|S_t|
\]  

(16)

and the threshold distance for a preset probability \( P \) given by equation (15). After both \( d_{ts} \) and \( d_{\text{thresh}} \) have been calculated, the ratio of the Mahalanobis-Like distance to the threshold distance can then be used as a relative distance measure between the two classes.
\[ d_{\text{ratio}} = \frac{d_{st}}{d_{\text{thresh}}} \]  

(17)

Since the separability measure is not symmetric, the ratio must be calculated twice for each pair of classes. As a result of calculating two separability measures for each pair of classes, a matrix must be used to represent all relative distance measures. The sum of this matrix can then be used as a measure of the overall separability of the classes. Before this summation however, any \( d_{\text{ratio}} \) values which exceed 1.0 are set to 1.0 to prevent the inflation of the overall separability measure. The overall separability is then computed for all permutations of feature subsets from the whole set. The subset with the highest overall separability should result in the highest classification accuracy.
1.7 Noise

1.7.1 Maximum Noise

Every aspect of image processing and classification is affected by noise. Although there are many types of noise associated with images, because of the nature of granulometric based features, we will concern ourselves only with maximum additive noise. Since this type of noise adds to the activated pixel count of a thresholded image needed for binary granulometries, it is readily apparent that additive maximum noise will skew the granulometric distributions used to generate these features.

There are many types of maximum noise which are inherent to digital images. Of these, we will examine the effects of four basic categories: point noise, occlusion noise, scratch noise and spaghetti noise. Dougherty, Pelz and Newell [1990] briefly examined the effect of maximum point noise and spaghetti noise on granulometric based features. Although it was concluded that the features were generally robust, further examination is required for a deterministic analysis of the effect of additional noise on image texture classification.

1.7.2 Point Noise

Point noise is defined as single random activated pixels. It may be caused by flaws inherent to the detector, by dust and dirt on any digitally scanned image, or by electronic noise at any level of the system. Since uniform response of array detectors is virtually impossible, fluctuations in the output signal caused by nonlinear pixels may vary enough to
"push" some pixel values past a given threshold. The signal may also fluctuate from other electronics in the detector such as photomultipliers and amplifiers.

1.7.3 Occlusion Noise

Occlusion noise is defined as noise which occludes or covers an underlying signal. In an image, this can be thought of as particles which are large enough to alter the apparent shape or boundary of image structures and substructures. This can be caused by larger dirt and dust particles on digitally scanned images or an undesired intersecting object within the original image.

1.7.4 Scratch Noise

Scratch or streak noise is characterized by long, thin straight lines which propagate in a single direction in a particular image. Physical scratches on a photographic negative can appear as maximum scratch noise on a photographic print. These scratches are commonly caused by photographic equipment and processing machinery as the negative is pulled through. Single element flaws in a "push broom" type detector can have a similar effect on digitized images. As the one-dimensional detector array moves across an image, one defective element or an element impaired by dirt can cause a streak in the resultant digital image.
1.7.5 Spaghetti Noise

Spaghetti noise is characterized by a thin "curly" or "windy" line of connected pixels. Depending on the relative size of the detector elements and images being scanned, this type of noise can be caused by hair, dust or other fibers on the image or detector.
2.0 Statement of Work

2.1 Selection of Texture Images

Ten texture images were chosen for the study. All ten were taken from Brodatz' collection of photographic texture images [1966]. These photographs were scanned at 75 dpi into an 8-bit grey-scale digital format. The ten images were selected to represent a large range of textural complexity. This textural complexity can be thought of as the amount of structure in the underlying texture primitives and the variation in that structure. Figure 9 shows all ten images along with Brodatz' original descriptions.

The original digital images were 512 x 512 pixels. However, the 132 x 132 pixel images in Figure 9 were cropped from the original images before processing to limit the computation time. A complete description of the reasons for this exact size are stated in section 2.4. Each texture image represents a separate texture class. Throughout this paper, each texture class will be referred to by the description number given by Brodatz (i.e. d102, d103, etc.).
Figure 9(a-f): Texture images
Figure 9(g-j): Texture images

- g) d67 Plastic pellets (inverted grey scale)
- h) d68 Wood grain
- i) d75 Coffee beans
- j) d84 Raffia looped to a high pile
2.2 Thresholding of Texture Images

Before the local binary granulometries were calculated, these 8-bit grey-scale images were reduced to binary images. The use of a threshold provided the simplest method for the gray level compression. The choice of the threshold value could significantly change the results of the granulometries by changing the activated pixel count in the image. In many cases it is profitable to choose a threshold which results in approximately half the pixels being activated. However, when dealing with image texture, maintaining the underlying textural structure and substructures is the most important aspect. Although the textural structure of each image could be best maintained by choosing a separate threshold value for each image, a single threshold value was chosen which maintained a majority of the underlying structure in all the images and more closely simulated real life conditions for texture classification. Figure 10 shows the thresholded versions of the images in Figure 9.

These binary images could be greatly affected by nonuniformity of the images. Intra-image nonuniformity of the mean gray level of the 8-bit images could cause the size and shape of the texture primitives to vary significantly within a supposedly homogeneous image. This may have inflated the variance and could have skewed the granulometric distributions. Skewing the distributions may have caused a shift in the feature means. The inflated variance of the distributions may have resulted in a decrease of classification accuracy due to the increase of the total classification error. (See Figure 7).
Figure 10(a-f): Binary texture images
Inter-image nonuniformity of the mean gray level could cause a significant difference between the apparent image texture of the 8 bit images and the texture of resultant binary images. As a result, the most significant textural information in a gray scale texture image may not have been represented in the binary texture image. The granulometric features generated from such images may have been less able to distinguish between image classes.
2.3 Generation of Noise

As previously mentioned, four categories of additive maximum noise were investigated. The four noise models simulated were point noise, occlusion noise, scratch noise and spaghetti noise. This simulation was accomplished by directly overlaying noise images on the ten binary texture images. Noise images were created by placing a number of particles of noise, "noise elements", onto a blank image. Each noise element was described by its length, width, "straightness", and initial angle of propagation.

Each of the four noise models must have a certain mean and range for each of the four descriptive parameters mentioned above. For each noise model, appropriate values for the mean and range of the length and width were defined. Beta distributions were used to determine the length and width of each individual noise element. Since the beta distribution is only defined between 0 and 1.0, a scaling factor and/or a shift factor was applied to alter the range of the distribution so that the predefined mean number of pixels would lie within this range. The beta distribution parameters, \( \eta \) and \( \gamma \), were subsequently set to the appropriate values to determine the shape of the distribution and thereby the variation of the length and width of all the noise element in a noise image.

The initial angle of propagation could either be set or randomly chosen from a uniform distribution for each noise element. A beta distribution was used to determine how straight or curly a noise element would be. The maximum change in the angle of propagation was set by the mean of a scaled beta distribution. The variation of the direction of a line of pixels in the noise element was set by the \( \eta \) and \( \gamma \) values of this beta distribution. One pixel
was activated in the image. An adjacent second pixel was then activated at the initial propagation angle. A change in the angle of propagation was calculated using the beta distribution for the angle. The new angle of propagation then became the initial angle plus the angular change.

Point noise was the simplest of all the models. The length and the width of each noise element was a constant of 1 pixel. The straightness and angle of propagation of the noise elements were therefore irrelevant. The range scaling factor for the length and width distributions was set to 2 pixels. This forced the midpoint of the range to 1 pixel. Both \( \eta \) and \( \gamma \) were set to 10E+15 so that the final length and width distributions were effectively delta functions at 1 pixel.

The occlusion noise model was specified so that the length and width of each individual noise element were equal. Again, the straightness and angle of propagation of the noise elements were irrelevant. The range scaling factor for the length and width distributions was set to 8 pixels so that the midpoint occurred at 4 pixels. Both \( \eta \) and \( \gamma \) were set to 3.0 so that the final length and width distributions were symmetric, centered at 4 pixels with a standard deviation of approximately 2 pixels.

In the scratch noise model, the width was again set to 1 pixel for all noise elements by setting the range to 2 pixels and \( \eta \) and \( \gamma \) to 10E+15. The range of the length distribution was set to 40 pixels to center the distribution about 20 pixels. Parameters \( \eta \) and \( \gamma \) were set to 3.0 and 1.5, respectively, in order to skew the distribution to higher numbers of pixels without excluding the possibility of generating some noise elements of shorter length.
order to produce scratches in the same direction, the initial angle of propagation for all the noise elements in a single noise image was set to a random constant. This was accomplished by setting the range of the angle distribution to $2\pi$ radians, the range shift constant to a random number between 0 and $2\pi$, and the $\eta$ and $\gamma$ of the distribution to $10E+15$ to ensure straight propagation.

The spaghetti noise model also had the width set to 1 pixel for all noise elements. The range of the length distribution was set to 80 pixels to create noise elements approximately twice the length of the scratch noise. As with the scratch noise, $\eta$ and $\gamma$ for the length distribution were set to 3.0 and 1.5, respectively. The initial angle of propagation for each noise element was set to a random angle. The appropriate values for $h$ and $g$ of the angle distribution were found by varying these parameters until the noise elements had the desired curliness.

The initial image position $(i,j)$ of each noise element (i.e. the position of the first pixel of the element) was chosen from a two-dimensional uniform distribution the same size as the original texture images. A $132 \times 132$ non-activated pixel image was created as a template for the addition of the noise elements. After generation, each noise element was added to this image in order to create a noise image. This addition operation allowed for overlapping of the noise elements. A threshold of 1 was then applied to this image in order to create a binary noise image.

Six noise conditions were created for each noise model. These noise conditions varied by the percentage of activated pixels in the binary noise images. The six noise conditions
chosen for each of the four noise models were 0%, 5%, 10%, 15%, 20%, and 25% activated noise pixels. Examples of the noise models under different conditions are shown in Figure 11.
Figure 11(a-f): Examples of binary noise images
For each of the ten binary texture images, an independent random noise image of each noise model and condition was generated. The noise images were then added to each of the binary texture images. The texture-plus-noise images were subsequently thresholded at a value of 1 creating a total of 240 texture-plus-noise binary images.

Figure 11(g-l): Examples of binary noise images
2.4 Generation and Selection of Local Granulometric Features

In order to create the granulometric features needed for classification, local granulometries were run with five types of structuring elements. Four linear element sequence granulometries: horizontal, vertical, positive-diagonal (+45°) and negative-diagonal (-45°) as well as a sequence of circular elements were run on all 240 images. For each pixel, the local PSM, PSSD, and PSS were calculated for all five structuring element granulometries. The PSM of the MaxLin and Linearity measures were also calculated resulting in a total of 17 granulometric features for each image.

There were two main concerns about the selection of the feature data: 1) the need for good estimates of the class distributions and 2) the need to limit the amount of data to some computationally tractable amount. In accordance with Schowengerdt's [1983] recommendation, 100 pixels from each class were used in the study. Since each of the pixels in a class was to be represented by 17 features, a total of 1700 real data values were needed for each of the 240 binary images.

To ensure these pixels would accurately represent an entire texture class with or without noise, all pixels were randomly selected from 100 x 100 pixel "feature images". The feature images consisted of real numbers representing some local granulometric statistic about each pixel in the binary texture image. Since each binary image was assumed to represent a homogeneous texture, the feature images resulting from the local granulometries were assumed to be wide sense stationary. The 100 data values from each feature image were then randomly selected with the aid of a uniform pseudo-random number generator.
No pixel was allowed to be chosen a second time to ensure accurate estimates of the mean and variance of the distributions.

A 33 x 33 pixel window size was used to generate each of the feature images. Edge effects may be caused when this window does not lie entirely within a given image. Since the local granulometric statistics for areas lying near the edge of an image can significantly differ from those for interior image areas, a 132 x 132 pixel binary texture image was required so that the central 100 x 100 pixels could be sampled without edge effects.
2.5 Classification of Dependent and Independent Data

The initial step in all classification algorithms is training the classifier. Supervised training is used to identify an area representative of each class. In most cases, great care must be taken to include only pixels or data which belong to a given class. However, in this case, data from each texture class was easily separated since the granulometries were run separately on each class. This supervised training is conducted by inputting the 17 features for each of the 100 data points of each class into the classifier. The mean vector and covariance matrix for each class is then calculated, and a discriminant function is developed from these means and covariances.

Dependent data is defined as the set of data used to train the classifier. Classification of the dependent data can be used as an initial measure of the goodness of the classifier. A low degree of classification accuracy of the dependent data can imply an inadequate statistical difference among the classes. However, a high degree of accuracy of dependent data merely implies a reasonable statistical difference among the classes in the training data. Further examination is needed to determine the overall goodness of the feature vectors for classification of data not included in the training set.

After the classifier has been trained, independent data can be classified using the maximum likelihood discriminant function developed from the dependent data. This independent data typically contains some or all of the same classes as the dependent data. In this case, any set of feature values from all 10 texture classes could be used as the dependent training
data. Any other set of feature values from 1 to all 10 texture classes can then be used as the independent data.

The classification accuracy was determined by dividing the number of correctly classified pixels by the total number of pixels classified. This could then be used as a measure of the ability of the granulometric features to discriminate between the texture classes. The minimum window size for generating the granulometric features could be found by determining the point at which the classification accuracy became unacceptable. The minimum number of optimal features could be found in a similar manner. The classification accuracy could also be used as a measure of the robustness of the features in the presence of noise.
3.0 Analysis of Results

3.1 Dependent Classification

The initial indication of the power of the granulometric features was found by classifying the dependent data used to train the classifier. All 17 features were employed in the feature vectors for each class. The granulometries were run on the original 10 binary texture class images without any additional noise. The 17 features from 100 random pixels from each of the 10 texture classes were used to train a Gaussian maximum likelihood classifier. These same 1000 pixels were subsequently classified using the discriminant function developed. The results of this dependent classification are in the form of the confusion matrix in Table 1.

Table 1: Classification of dependent data

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Overall classification accuracy = 100%
This confusion matrix shows that all of the dependent data were correctly classified. The rows of the matrix represent the original class of each pixel. The columns represent the class into which each pixel was classified. Since there were 100 pixels from each class, the values in the matrix represent both the number and the percentage of pixels classified into the class designated by the column.

Although the data failed a homogeneity test for equal covariance of the classes, the classifier was trained a second time with the same data using a pooled covariance to test the statistical separation of the means. In order to achieve a high classification accuracy using the pooled covariance, the feature means had to be sufficiently separated to minimize the probability distribution overlap. The results of classifying the dependent data using a pooled covariance were identical to the results using within-class covariance. This demonstrates that the mean vectors of all ten classes were well separated and indicates that the granulometric features sufficiently represented the basic textural differences between the classes.

3.2 Independent Classification

Independent data was employed to determine the overall goodness of the classifier. After training with feature values from the original set of 1000 dependent pixels, a second set of 100 pixels was randomly selected using a uniform distribution. Again, all 17 features were included in the feature vectors for each class. This was considered an independent set of data since the probability of a repeat pixel was only 0.01 using the uniform distribution. However, it should be noted that no attempt was made to test for rotational or
magnificational robustness of the features since the most of the granulometric features were inherently size and direction dependent. The results of the classification are given in the confusion matrix in Table 2.

Table 2: Classification of independent data

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Overall classification accuracy = 99.8%

Table 2 shows that only two of the independent pixels were misclassified. The overall classification accuracy of 99.8% indicates that assumption of within-class homogeneity of the 17 features was justified. This also indicates that the basic textural differences between the classes were well represented by these granulometric features.

As with the dependent classification, the classifier was trained a second time using a pooled covariance matrix to assure that the feature means were well separated. The results of this classification are given in Table 3.
Table 3: Classification of independent data using pooled covariance

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Overall classification accuracy = 99.6%

Note that the overall classification accuracy decreased by only 0.2% when compared to classification using within-class covariance. The difference in classification accuracy was due to the difference of the estimated feature distributions for each class. Since the pooled covariance matrix was an estimate of the average covariance of the ten classes, the estimates of wide within-class variances tended to be narrower using pooled covariance. Likewise, the estimates of narrow within-class variances tended to be wider using pooled covariance. In general, this caused an increase of the probability overlap and introduced some misclassification.
3.3 Minimal Window Size Determination

Overall classification accuracy was used to determine the minimum local window size needed for classification. Six local granulometries were run on each of the ten texture images using square windows with sides of length 7, 11, 15, 19, 25, and 33 pixels. Two sets of feature data were collected in order to determine the effect of window size on both dependent and independent data.

Figure 12 shows results of classification with the 6 different size windows. The side length of the window is referred to as the window size. Note that the classification accuracy axis on this graph ranges only from 80% to 100%.

![Figure 12: Classification Accuracy vs. Window Size](image-url)
Over 99% classification accuracy of the dependent data was achieved for all window sizes greater than 11 pixels. Although the classification accuracy of the independent data was less than that of the dependent data, the classification was still 94.6% accurate using a window size of 11 pixels. It also should be noted that the classification accuracy for both the dependent and independent data fell dramatically below the window size of 11 pixels. This indicates that most of the underlying texture primitives which distinguish these image textures were no smaller than 11 pixels. However, it should be kept in mind that the minimal effective window size is a function of the image texture set.
3.4 Optimal Feature Selection

A number of available methods for determining an optimal feature set were applied. Richards' [1986] method for determining an optimal feature set by the degree of overlap of the class distributions is considered the most accurate since it uses the covariance of all classes and requires only the assumption of Gaussian normal distributed features in each class. However, this method is also the most computationally intensive. The total number of calculations needed to determine the divergence is determined by the number of permutations of optimal features to choose out of the total number of features. For example, to choose the optimal 6 features out of a total of 17 for all 10 classes, the number of calculations would be:

\[
\frac{17!}{3! (17-3)!} \cdot \frac{10!}{2! (10-2)!} = 30600
\]  

For each of the 30600 divergence measures, 2 matrix inverses must be computed.

The next viable option for optimal feature selection was the class separation method developed by Rosenblum [1990]. The results should be similar to Richards' method since the covariance matrices of all classes were incorporated into the separation measure. This method had the advantage of being faster because only one matrix inversion is required for each separation measure between two classes.

In order to determine the optimal number of features needed for adequate classification, the ten image textures were classified using sets of optimal features. After each set of \( m \)
optimal features was found from the 17 features, all other features were removed from the feature vectors. The optimal feature data from the ten image textures was then used to train the classifier. Both dependent and independent data were subsequently classified and the overall classification accuracy was determined for each optimal feature set. The results of classification with the optimal feature sets are given in Figure 13.

![Classification Accuracy vs. Number of Optimal Features](image)

**Figure 13:** Classification Accuracy vs. Number of Optimal Features

Note that the classification accuracy axis on this graph ranges from 60% to 100%. Over 99% classification accuracy of both the dependent and independent data was achieved with 6 optimal features. Additional features contributed very little to improving this accuracy. The first 6 optimal feature sets used in these classifications can be found in Table 4. A complete listing of all optimal feature sets can be found in Appendix B.
Notice that the classification accuracy using 5 optimal features was slightly less than the classification accuracy using 4 optimal features. The addition of more features to the feature vectors does not necessarily correspond to higher classification accuracy. The classification accuracy may even decrease if the probability overlap between the classes is increased by the addition of more features. In this case, there was more probability overlap between the ten classes with any set of five features than there was with the optimal set of 4 features.

Although, as previously stated, most of these granulometric features used were size and direction dependent, it is interesting to note that the circular PSM, which is rotationally invariant was the most significant of all 17 features and appeared in each of the first four optimal feature sets. Linearity PSM, which is invariant to both direction and scale, also appeared in the set of 3 optimal features. Although these optimal feature sets are dependent on the image texture classes, given the diverse range of image texture classes in this study, an optimal set of 6 features for any given set of texture classes can be expected to give similar results.
Table 4: Optimal Feature Sets using Rosenblum Optimization

<table>
<thead>
<tr>
<th>Number of Features</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 feature</td>
<td>circular PSM</td>
</tr>
<tr>
<td>2 features</td>
<td>circular PSM</td>
</tr>
<tr>
<td></td>
<td>horizontal PSSD</td>
</tr>
<tr>
<td>3 features</td>
<td>circular PSM</td>
</tr>
<tr>
<td></td>
<td>horizontal PSSD</td>
</tr>
<tr>
<td></td>
<td>Linearity PSM</td>
</tr>
<tr>
<td>4 features</td>
<td>circular PSM</td>
</tr>
<tr>
<td></td>
<td>horizontal PSM</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSSD</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSS</td>
</tr>
<tr>
<td>5 features</td>
<td>horizontal PSM</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSM</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSSD</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSS</td>
</tr>
<tr>
<td></td>
<td>positive-diagonal PSM</td>
</tr>
<tr>
<td>6 features</td>
<td>horizontal PSM</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSM</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSSD</td>
</tr>
<tr>
<td></td>
<td>negative-diagonal PSS</td>
</tr>
<tr>
<td></td>
<td>positive-diagonal PSM</td>
</tr>
<tr>
<td></td>
<td>vertical PSM</td>
</tr>
</tbody>
</table>

Note: The order of features within any optimal set has no significance.
3.5 Classification in the Presence of Maximum Noise

3.5.1 Dependent Classification

After the addition of maximum noise, the feature values from 100 random pixels for each of the ten texture-plus-noise images were used to train the classifier. The results of the subsequent dependent classification are given in Table 5.

Table 5: Classification of dependent texture-plus-noise data

<table>
<thead>
<tr>
<th>Noise Model</th>
<th>% Image Area Covered by Noise*</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Noise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>Occlusion Noise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Spaghetti Noise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Scratch Noise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Calculated from noise image before addition to texture image
Excluding the results of classification with 0% noise, the overall classification accuracy of the texture-plus-noise dependent data was 99.984%. This indicates a good separation of the feature vectors of the texture-plus-noise classes derived from the local granulometries. However, since the classifier was trained with the features generated from the texture-plus-noise images, the granulometric texture features may have describe only the noise, not the underlying image textures.

3.5.2 Independent Classification

In order to test whether the image texture or the noise was being classified, an independent data set derived from texture images with no additional noise was used to train the classifier before classification of the texture-plus-noise pixels. The results of the subsequent independent classification are shown in Figures 14-17. The classification accuracy after training with both within-class covariance and pooled covariance is shown on the graphs.
Figure 14: Classification Accuracy vs. % Point Noise

Figure 15: Classification Accuracy vs. % Spaghetti Noise
Figure 16: Classification Accuracy vs. % Occlusion Noise

Figure 17: Classification Accuracy vs. % Scratch Noise
In general, the highest classification accuracy was achieved with the images degraded with point noise. From this, it can be deduced that these granulometric texture features were most robust in the presence of point noise. The PSM feature values for the images degraded with point noise were shifted to smaller values since single pixels were added to the granulometric distributions. The addition of the point noise increased the values at the low end of the pattern spectrum which tended to inflate all the PSSD feature values of the classes. This also caused the PSS feature values to increase since the skewness of the granulometric distributions tended to be more positive.

The second highest classification accuracy was achieved with the images degraded with spaghetti noise. The results were not as good as those achieved with point noise, however, the classification accuracy was dramatically higher than that achieved with occlusion or scratch noise. Although the particles of spaghetti noise were very long, they had little rigid structure. Since all structuring elements used in granulometries must be convex and therefore highly structured, the addition of spaghetti noise did not have any consistent effects on granulometric feature values. Although the variance of all feature distributions increased with the addition of spaghetti noise, the lack of structure made it difficult to determine the effects on the granulometric feature values.

The well defined circular structure of occlusion noise tended to obscure the underlying image textures. Although occlusion noise closely resembled point noise in many respects, the classification accuracy in this type of noise was far worse than the classification in point noise. From the graph, the classification accuracy is seen to be significantly decreased between 5% to 10% point noise. Since the diameter of this occlusion noise was four times
that of the point noise on average, all PSSD feature values were abruptly enlarged. The addition of small occlusion noise had the same affect on the PSM feature values as the addition of point noise. However, the PSM values may have been only slightly shifted since the addition of larger noise particles tended to increase PSM values. The variance of the PSM feature values was inflated from both these effects which caused many of the values to fall far from the mean assumed by the classifier.

Although scratch noise seems to be very similar to spaghetti noise, this type of noise had considerably more structure. As with occlusion noise, this structure tended to obscure the underlying image texture of the classes. As expected, the features derived from linear structuring elements were the most affected by the linear structure of scratch noise. As a result of this linear structure and the random selection of the initial angle of propagation, the classification accuracy with scratch noise was quite poor.

Two other types of scratch noise were investigated to determine how well the granulometric features would classify the images in the presence of fixed direction scratch noise. Figure 18 shows the results of independent classification with three types of scratch noise. The original scratch noise shown in Figure 17 as "unpooled cov" is reproduced on this graph as "Random direction". The two other types of noise are horizontal and fixed scratch noise. The initial angles of propagation of the horizontal and fixed scratch noise were 0° and 67.5° respectively.
Figure 18: Classification Accuracy in the presence of Horizontal, Fixed and Random Scratch Noise

The worst classification results came from the horizontal scratch noise using all 17 features in the feature vectors. The classification accuracy dropped below 50% with only 5% added noise. The same ten texture-plus-horizontal-noise images were reclassified after removal of all features derived from horizontal granulometries. The horizontal PSM, PSSD, and PSS were removed from the feature vectors because the addition of horizontal linear noise might have obscured any textural information about the underlying image classes contained in these features. The MaxLin and Linearity were also removed since both could have been largely influenced by the inflated horizontal feature values. Results of classification with these reduced feature vectors support these assumptions. In contrast to the previous classification, a classification accuracy of 93% was achieved in the presence 5% noise. In
fact, the classification accuracy using the reduced feature vectors were consistently higher than the classification using all 17 features.

Out of all four classifications with maximum scratch noise, the highest classification accuracy occurred in the presence of fixed direction scratch noise. The initial angle of propagation was set to 67.5° in order to minimize possible effects on features generated from any of the four linear structuring elements. Although still much worse than spaghetti and point noise at higher noise levels, classification accuracy exceeded 85% in the presence of 10% fixed noise.

3.5.3 Combinations of Noise Models

To test the robustness of the granulometric features in the presence more than one type of noise, combinations of the four noise models were added to the ten texture images. Subsequent classification resulted in accuracies similar to the individual noise models. As would be expected, Figure 19 shows the combination of point and spaghetti noise achieved the highest classification accuracy. Combinations which included noise with more structure produced poorer results. The sporadic jumps in the classification accuracy with the point plus scratch noise were due to the random direction of the scratch noise.
Figure 19: Classification accuracy with combinations of noise models
An examination of some confusion matrices revealed much more information about the
decrease of classification accuracy in the presence of maximum noise. Tables 6 and 7 contain the confusion matrices for independent classification in the presence of 5% and 10% point noise, respectively.

While point noise did reduce classification accuracy, the overall classification accuracy in the presence of 5% point noise was still 98.4%. Note that only pixels from classes d103, d52 and d65 were misclassified. The overall classification accuracy decreased to 86.5% for the independent classification in the presence of 10% point noise. Note, however, that the classification error was not uniformly distributed over the set of all classes. In fact, d102 was totally misclassified into d64.

An examination of the two image textures revealed some possible reasons for this erroneous classification. Class d102 was the simplest of the ten image texture classes. Since the original 8-bit image had very high contrast, the binary image was nearly the same as the grey scale image. There were only two basic textural primitives which made up the entire image structure; small circular particles and large circular particles. Consequently, the granulometric feature distributions tended to be well defined and compact.

On the other hand, d64 had the most subjective textural complexity of the classes. Although there was a systematic pattern to the 8-bit grey scale image, the lower contrast of this image and greater complexity resulted in widely varying structures in the resulting binary image after thresholding. Because of the variance in the underlying texture
primitives in class d64, the granulometric feature distributions of this texture class tended to possess much more dispersion than those of class d102.

Table 6: Classification of independent data in 5% point noise

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>89</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 7: Classification of independent data in 10% point noise

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>73</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>96</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>
Figures 20-22 show the feature value distributions of classes d64, d102, and d102-plus-10% point noise for three of the granulometric features.

Figure 20: Feature distributions for Circular PSSD

Figure 21: Feature distributions for Negative-Diagonal PSSD
Remember that all of the feature distributions were approximated by Gaussian normal distributions in the Gaussian maximum likelihood classifier. The approximate probability distributions for each of the three features represented in Figures 20-22 are shown in Figures 23-25. Each curve represents the approximate Gaussian feature value probability distributions of three image textures; d64, d102 and d102 with 10% point noise.
Figure 23: Probability distributions for Circular PSSD

Figure 24: Probability distribution for Negative-Diagonal PSSD
Note that the d64 distributions in all three figures were much wider than the distributions of d102 or d102 with 10% added point noise. In Figures 23 and 24, the mean of class d102 shifted to higher values when the point noise was added. In both cases, the estimated Gaussian normal probability distributions of d102 and d102-plus-noise barely overlapped. However, almost the entire distribution of the feature values for the d102-plus-noise class lay less than 2 standard deviations from the mean of the d64 distributions. For this reason, the probability of the feature values of d102-plus-noise coming from class d64 was higher than the probability of the feature values coming from class d102.

In Figure 25, the mean of class d102 shifted to smaller values and there was some overlap of the d102 and d102-plus-noise distributions. However, the majority of the d102-plus-
noise distribution lay very near the mean of the broader d64 distribution. Most of the positive-diagonal PSM feature values still had a higher probability of coming from the class d64 than class d102.

All covariance values for the 17 features of the multivariate Gaussian normal distribution for d64 were greater than the covariance values for the d102 multivariate distribution. Since there was a shift in the mean of the feature distributions with very little change in the variance, each pixel of the d102-plus-noise image had a higher probability of coming from the class d64 than class d102. Consequently all of the pixels of d102 with 10% point noise were classified into d64 when the classifier was trained with independent data without noise.

This type of rationale can also be used to explain the higher classification accuracy achieved with the pooled covariance when in the presence of noise (See Figures 14-17). The pooled covariance increased the variance of the narrower feature probability distributions, thereby increasing the probability of correctly classifying those classes with very narrow feature distributions. There was a greater amount of misclassification of the more texturally complex classes. However, since the distributions had large variance, the percentage of misclassification of the complex classes in this case was less than that of the texturally simple classes using within-class covariance. The overall classification accuracy was therefore higher using the pooled covariance when in the presence of noise.

Results of classification of noisy data with only optimal features are given in Figures 26 - 29. Note that in each case only the first few features added to the classification accuracy.
When features with less ability to discriminate between texture classes were added to the feature vectors, the classification accuracy decreased. The noncritical data from these features actually interfered with the ability of the classifier to distinguish between the texture classes.
Figure 26: Optimal Feature Classification in Point Noise

Figure 27: Optimal Feature Classification in Spaghetti Noise
Figure 28: Optimal Feature Classification in Occlusion Noise

Figure 29: Optimal Feature Classification in Scratch Noise
3.6 Noise Estimation

Although the classification accuracy in the presence of noise was acceptable given the amount of noise added to the images, any means by which to improve this accuracy could enhance the usefulness of the classifier. One way to improve the classification accuracy was to estimate the noise. In order to utilize noise estimation in classification, noise was added to the image class set before generation of the features.

To simulate this, the classifier was initially trained with feature data calculated in the presence of noise. An independent set of pixels with a different amount of maximum noise was subsequently classified. Since the noise could be over- or underestimated, classification was performed after training in the presence of both more and less noise than the noise level of the data to be classified. Table 7 shows the results of classifying data corrupted with 10% point noise after training with data corrupted by 5% point noise. Table 8 shows the results of classifying data corrupted with 5% point noise after training with data corrupted by 10% point noise.
Table 8: Classification of data with 10% point noise after training with 5% point noise

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Overall classification accuracy = 99.8%

Table 9: Classification of data with 5% point noise after training with 10% point noise

<table>
<thead>
<tr>
<th></th>
<th>d102</th>
<th>d103</th>
<th>d20</th>
<th>d52</th>
<th>d64</th>
<th>d65</th>
<th>d67</th>
<th>d68</th>
<th>d75</th>
<th>d84</th>
</tr>
</thead>
<tbody>
<tr>
<td>d102</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d103</td>
<td>0</td>
<td>96</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d20</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d52</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d65</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>d84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Overall classification accuracy = 99.1%
Note that the overall classification accuracy for 10% point noise was only 86.5% when training without noise estimation. This simple estimation therefore produced a 13.3% rise in classification accuracy. Although not as pronounced, the classification accuracy of 5% point noise still rose 0.7%. Figure 30 shows the results of estimating the noise for all four noise models.

Figure 30: Classification Accuracy with Noise Estimation

Negative noise estimation refers to underestimating the amount of noise, and therefore training in too low a noise environment. Positive noise estimation refers to overestimating the amount of noise. Three points should be noted: 1) Accurate noise estimation significantly increased classification accuracy, 2) degradation with point noise and spaghetti noise still resulted in the highest classification, and 3) the results for all four noise models
were better when overestimating the amount of noise rather than underestimating the amount of noise to a similar degree.
4.0 Conclusions

The present study indicates that local granulometric moments provide good feature sets for texture-based pixel classification. Classifying the ten texture images using all 17 granulometric features in a Gaussian Maximum Likelihood classifier, classification is nearly perfect with only 0.2% misclassification. With 6 or more optimal granulometric features, classification accuracy of independent data in the absence of noise exceeds 99%. With 3 or more optimal granulometric features, classification accuracy of independent data still exceeds 90%. This minimization analysis indicates that the number of granulometric features may reduce to a fairly small number for any given set of images textures.

The smallest practical window size is required for accurate detection of the boundaries between local image textures. Using texture images from Brodatz' collection at spatial resolution of 75 dpi, classification accuracy exceeding 95% was achieved using a local window size of 11x11 pixels if all 17 granulometric features are employed. Classification accuracy decreased dramatically below the window size necessary to include at least the smallest of the critical textural primitives for each class. It is preferable to use local windows of greater size if the location of the boundaries is not critical since classification using features derived within larger windows produce more accurate results for pixels located entirely within one texture region.

These features are generally robust to maximum point noise and spaghetti noise. Independent classification of the image textures in the presence of 5% point or spaghetti noise produces classification accuracy exceeding 95%. If the pooled covariance is used,
this level of classification accuracy can be achieved in the presence of up to 10% noise. Although pooling the covariance is not generally considered beneficial, the values of granulometric features in the presence of noise tend to shift with little change in the overall variance. Pooling the covariance causes the variance of the narrower feature distributions to widen which tends to include the shifted feature values which would be excluded using the within-class covariance. Since the distributions of the granulometric features for these classes are fairly well separated, pooling the covariance causes little change to the overall probability overlap of the texture classes.

These features appear to be less robust to maximum occlusion and scratch noise. The critical difference between these two types of noise and the point and spaghetti noise is the amount of underlying image structure within the noise elements. Both the point and spaghetti noise have very little underlying multi-pixel structure. By definition, point noise contains only randomly activated pixels. Although the pixels in spaghetti noise are contiguous, this multi-pixel structure does not consistently affect any granulometric features since the direction of propagation is random. In contrast, the circular structure of occlusion noise and the linear structure of scratch noise tend to significantly influence the granulometric feature distributions of the classes. This causes the classification accuracy in the presence of occlusion and scratch noise to be far worse than the accuracy in the presence of similar amounts of point or spaghetti noise.

Finally, these features are generally robust to the presence of all 4 types of maximum noise if the type and amount of noise can be accurately estimated. Accuracy once again reaches very high levels if the classifier is trained in a noise environment which is a fairly good
estimate of the actual noise to be encountered. The results for all 4 noise models are better when the noise level is overestimated rather than underestimated to a similar degree. It should be noted, however, that even using noise estimation, the classification accuracy is still better in the presence of point noise and spaghetti noise than in the presence of occlusion noise or scratch noise.

4.1 Suggestions for Future Work

Although this exhaustive empirical study of binary granulometric features has demonstrated the potential for application in pixel classification, these features need to be tested in a real situation classification. The classification potential of these features may rise considerably if combined with other textural and/or non-textural features. In remote sensing applications, these features may be combined with spectral data. They may also prove valuable in medical imaging using image information from X-ray, ultra sound and magnetic resonance imaging.

The use of multiple window sizes and multiple thresholds has the potential to provide much more textural information about any given set of classes. If a threshold probability for classification is set for larger windows, the boundaries between classes may be found by using smaller windows. Areas between classes should be left as unclassified using the features generated within larger windows. Granulometric features generated within smaller windows may be used to classify the unclassified pixels between classes.
The use of features derived from grey scale granulometries may provide much more
information and classification power than even multiple thresholds. However, without the
use of a high speed processor to calculate these three dimensional granulometries, the
increased information may be outweighed by the speed and relative accuracy of the multiple
binary granulometries.

These features could also be employed for texture-based object recognition in
circumstances where accurate edge detection is not practically feasible. If an object within a
scene possesses some unique textural characteristics, local granulometric moments may be
used to identify the object even if the boundaries between the object and the background are
ill defined. Once the local window lies entirely within the object, so that the resulting
features are free of boundary effects, pixels can be classified as belonging to that object's
texture.
5.0 References


Dougherty, E. and J. B. Pelz, Textured-Based Segmentation by Morphological Granulometries, *Electronic Imaging East*, October 1989

Dougherty, E., J. Pelz, and J. Newell, Granulometric Texture-Based Classification, *Electronic Imaging West*, February 1990


James, M., *Classification Algorithms*, Wiley-Interscience, 1976


Appendix A

Derivation of the separation distance for probability overlap

The probability of finding the mean of class t in a sample from class s is described as (Schowengert, 1983)

\[ P(m|s) = \left[ \frac{1}{\left| S_s \right|^{1/2} 2\pi^{k/2}} \right] e^{-1/2[(m_t - m_s)' S_s^{-1} (m_t - m_s)]} \]

where

- \( P(m|s) \) is the probability that the mean of class t belongs to class s
- \( k \) is the number of features which are to be optimized
- \( S_s \) is the covariance matrix for class s
- \( m_s \) is the mean vector for class s
- \( m_t \) is the mean vector for class t.

By taking the natural log of both sides, the equation becomes

\[ \ln[P(m|s)] + 1/2 \ln|S_s| + k/2 \ln[2\pi] = -1/2 [(m_t - m_s)' S_s^{-1} (m_t - m_s)]. \]

Multiplying by -2 and adding \( \ln|S_s| \) to both sides gives the equation

\[ [(m_t - m_s)' S_s^{-1} (m_t - m_s)] + \ln|S_s| = -2 \ln[P] - k \ln[2\pi]. \]

This equation can be modified to the final form of

\[ [(m_t - m_s)' S_s^{-1} (m_t - m_s)] + \ln|S_s| = -2 \ln[P / 2\pi^{k/2}]. \]
### Appendix B

**Optimal Feature Sets using Rosenblum Optimization**

(Note: The order of features within any optimal set has no significance.)

1 feature:  
circular PSM

2 features:  
circular PSM  
horizontal PSSD

3 features:  
circular PSM  
horizontal PSSD  
Linearity PSM

4 features:  
circular PSM  
horizontal PSM  
negative-diagonal PSSD  
negative-diagonal PSS

5 features:  
horizontal PSM  
negative-diagonal PSM  
negative-diagonal PSSD  
negative-diagonal PSS  
positive-diagonal PSM

6 features:  
horizontal PSM  
negative-diagonal PSM  
negative-diagonal PSSD  
negative-diagonal PSS  
positive-diagonal PSM  
vertical PSM

7 features:  
horizontal PSM  
negative-diagonal PSM  
negative-diagonal PSSD  
negative-diagonal PSS  
positive-diagonal PSM  
vertical PSM  
Linearity PSM
8 features:
- horizontal PSM
- negative-diagonal PSM
- negative-diagonal PSSD
- negative-diagonal PSS
- positive-diagonal PSM
- vertical PSM
- Linearity PSM
- MaxLin PSM

9 features:
- horizontal PSM
- negative-diagonal PSM
- negative-diagonal PSSD
- negative-diagonal PSS
- positive-diagonal PSM
- positive-diagonal PSSD
- vertical PSM
- Linearity PSM
- MaxLin PSM

10 features:
- horizontal PSM
- negative-diagonal PSM
- negative-diagonal PSSD
- negative-diagonal PSS
- positive-diagonal PSM
- positive-diagonal PSS
- vertical PSM
- vertical PSSD
- Linearity PSM
- MaxLin PSM

11 features:
- horizontal PSM
- negative-diagonal PSM
- negative-diagonal PSSD
- negative-diagonal PSS
- positive-diagonal PSM
- positive-diagonal PSS
- vertical PSM
- vertical PSSD
- vertical PSS
- Linearity PSM
- MaxLin PSM
12 features:
circular PSM
circular PSSD
horizontal PSS
negative-diagonal PSS
positive-diagonal PSM
positive-diagonal PSSD
positive-diagonal PSS
vertical PSM
vertical PSSD
vertical PSS
Linearity PSM
MaxLin PSM

13 features:
circular PSM
circular PSSD
horizontal PSM
horizontal PSSD
negative-diagonal PSM
negative-diagonal PSSD
negative-diagonal PSS
positive-diagonal PSM
positive-diagonal PSSD
positive-diagonal PSS
vertical PSM
vertical PSS
Linearity PSM

14 features:
circular PSM
circular PSSD
horizontal PSM
horizontal PSSD
negative-diagonal PSM
negative-diagonal PSSD
negative-diagonal PSS
positive-diagonal PSM
positive-diagonal PSSD
positive-diagonal PSS
vertical PSM
vertical PSS
Linearity PSM
MaxLin PSM
15 features:
circular PSM
circular PSSD
horizontal PSM
horizontal PSSD
negative-diagonal PSM
negative-diagonal PSSD
negative-diagonal PSS
positive-diagonal PSM
positive-diagonal PSSD
positive-diagonal PSS
vertical PSM
vertical PSSD
vertical PSS
Linearity PSM
MaxLin PSM