4.3 A Novel Homomorphic Filtering Technique using DCT

The coefficients from the Fourier Transform are complex which provides an obstacle to using the Fourier coefficients in classification. This is overcome by using the magnitude, phase or energy of a Fourier transform which leads to loss of information. An alternative is to use the Discrete Cosine Transform (DCT) which has real coefficients [Jain, 1989].

In this section we propose to use the DCT to replace the Fourier transform in the homomorphic filtering process described in Section 4.2. To enable this the filter matrix, $H(u,v)$, must be altered as the DCT results in a different frequency range being represented by the DCT coefficients. Instead of rotating the filter cross-section (shown in Figure 4.6) $360^\circ$ as in Section 4.2, the cross-section is rotated $90^\circ$ to cover one quadrant [Jain, 1989]. The cross-section shown in Figure 4.6 is based on a
parabola, this still provides suppression of the illumination component but also provides suppression of high frequency noise components. Figure 4.5 shows the horomorphobic process which incorporates the DCT.

![Figure 4.5: Homorphic Filtering using DCT](image)

Figure 4.6 shows results of the homorphic filtering procedure and using the filter depicted by Figure 4.6, as described within this section, to the bank, lake and hot cylinder marks originally shown in Figure 2.2, 2.3 and 2.4 respectively.
Figure 4.7: Homomorphic Filtered Images, (a) Lake, (b) Bank and (c) Hot Cylinder Marks
5.0 Neural Network based Classification System

5.1 Introduction to Neural Networks

Artificial neural networks, commonly referred to as ‘neural networks’ provide a unique computing architecture for classification systems [Lippman, 1987]. The computing architecture is based on dense arrangements of interconnections and surprisingly simple processors (neurons). The power of the neural networks lies in the tremendous number of interconnections.

Two key characteristics of a neural network are [Banks, 1990]:

- Knowledge is acquired by the network through a learning process
- Connection strengths known as synaptic weights are used to store the knowledge

The procedure used to perform the learning process is referred to as the training algorithm, the function of which is to modify the synaptic weights of the network in an orderly fashion so as to attain desired design objectives.

5.2 Neural Model

In the model of a neuron shown in Figure 5.1 are three basic elements [Schalkoff, 1989]:

- A set of synapses, each of which is characterised by a synaptic weight or strength of its own. A signal \( x_i \) at the input of the synapse \( i \) connected to the neuron \( j \) is multiplied by the synaptic weight \( w_{ji} \).

It is important to note the manner in which the subscripts of the synaptic weight, \( w_{ji} \), are written. The first subscript refers to the
neuron in question and the second refers to the input end of the synapse with weight \( w_{ji} \) is connected.

- An adder for summing the input signals, weighted by the respective synapses of the neuron.
- An activation function for limiting the amplitude of the output of a neuron. The activation function as also referred to as a squashing function in that it squashes the permissible amplitude range of the output signal to some finite value.

The model of the neuron shown in Figure 5.1 also includes an externally applied threshold \( \theta_j \) that has the effect of increasing or decreasing the net input of the activation function depending on the sign of the threshold. This threshold may be implemented by adding an extra weight, \( w_{j0} \), to the input of the adder, the corresponding input will be fixed at +1 with the weight \( (w_{j0}) \) equal to the negative of the threshold \(-\theta_j\). We may describe a neuron \( j \) by writing the Equations 5.1 and 5.2 [Banks, 1990]:

\[
\begin{align*}
  u_j &= \sum w_{ji} x_i \\
  y_j &= f(u_j - \theta_j)
\end{align*}
\]

where \( x_1, x_2, x_3, \ldots, x_p \) are the input signals;

\( w_{ji}, w_{j2}, w_{j3}, \ldots, w_{jp} \), are the synaptic weights of neuron \( j \);
\( u_j \) is the adder output;
\( \theta_j \) is the threshold
\( f(.) \) is the activation function;
\( y_j \) is the output signal of the neuron.

### 5.3 Types of Activation Functions

The activation function denoted by \( f(.) \), defines the output of a neuron in terms of the activity level at its input. There are three basic types of activation functions:

1. **Threshold Function**: For this type of function (Figure 5.2a) we have [Davalo, 1991] :

   \[
   f(u - \theta) = \begin{cases} 
   1 & \text{for} \quad (u - \theta) \geq 0 \\
   0 & \text{for} \quad (u - \theta) < 0 
   \end{cases}
   \]

   ![Figure 5.2(a) : Threshold Function](image)

2. **Piecewise-Linear Function**: For the piecewise-linear function (Figure 5.2b) we have [Davalo, 1991]:

   \[
   f(u - \theta) = \begin{cases} 
   1 & \text{for} \quad (u - \theta) \geq 1/2 \\
   v & \text{for} \quad -1/2 < (u - \theta) < 1/2 \\
   0 & \text{for} \quad (u - \theta) \leq -1/2 
   \end{cases}
   \]
Sigmoid Function: The sigmoid function is the most common form of activation function used in the construction of artificial neural networks. It is defined as a strictly increasing function that exhibits smoothness and asymptotic properties. An example of the sigmoid is defined by Equation 5.3 [Gonzalez, 1989]:

\[ f(u - \theta) = \frac{1}{1 + e^{a(u - \theta)}} \quad 0.1 < a < 10 \]  

(5.3)

where a is the slope parameter of the sigmoid function. By varying the parameter a, we obtain sigmoid functions of different slopes (Figure 5.4 c). A sigmoid function assumes a continuous range of values from 0 to 1 and is differentiable.
It is sometimes desirable to have the activation function range from -1 to +1. In which case, for a sigmoid we may use the hyperbolic tangent function, defined by Equation 5.4:

\[ f(u - \theta) = \tanh \left[ \frac{a(u - \theta)}{2} \right] = \frac{2}{1 + e^{-a(u - \theta)}} - 1 \quad (5.4) \]

5.4 Network Architectures

5.4.1 Single-Layer Feedforward Networks

A layered neural network is a network of neurons organised in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons but not vice versa. Such a network is illustrated in Figure 5.3 for the case of four input nodes and three output nodes. From Figure 5.3 we see that all the inputs feed into the input of every neuron on the output layer. This type of network is referred to as a single-layer network, with the designation 'single-layer' referring to the output layer of computation nodes [Lippman, 1987]. In other words, we do not count the input layer of source nodes, because no computation is performed there. If a vector \( x \) is used as an input to such a network, the network recognises the vector \( x \) as belonging to class \( m \) if the \( m^{th} \) output of the network is 'high' while all other outputs are 'low' [Davalo, 1991].

![Figure 5.3: Single-Layer Feedforward Neural Network](image)
5.4.2 Multi-layer Feedforward Networks

The second class of feedforward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of the hidden neurons is to intervene between the external input and the network output. By adding one or more hidden layers, the network is enabled to form more complex decision regions in the problem domain [Lippman, 1987]. Figure 5.4 illustrates the layout of multi-layer feedforward neural network for the case of a single hidden layer. This network is referred to a 4-3-2 network as it has 4 source nodes, 3 hidden neurons, and 2 output neurons.

![Multi-layer Feedforward Neural Network](image)

Figure 5.4 : Multi-layer Feedforward Neural Network

The neural network if Figure 5.4 is said to be fully connected in the sense that every node in each layer of the network is connected to every node in the adjacent forward layer [Clarkson, 1995].
5.5 Back Propagation Training

Training is a process by which the parameters (synaptic weights) of a neural network are adapted through a continuing process of simulation by the environment in which the network is applied. This concept of training implies the following sequence of events [Gonzalez, 1989]

- The neural network is stimulated by an environment, in other words a vectors are presented to the network.
- The network undergoes changes as a result of this stimulation.
- The neural network responds in a new way to the environment, because of the changes that have occurred in its internal structure.

Consider the neuron shown in Figure 5.1, where $x_i$ and $v_j$ are connected by a synaptic weight $w_{ji}$. Let $w_{ji}(n)$ denote the value of the synaptic weight $w_{ji}$ at time $n$. At time $n$ an adjustment $\Delta w_{ji}(n)$ is applied to the synaptic weight $w_{ji}(n)$, yielding the updated value $w_{ji}(n+1)$. We may thus write Equation 5.5 to represent the training process [Lippman, 1987]:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (5.5)$$

where $w_{ji}(n)$ and $w_{ji}(n+1)$ may be viewed as the old and new values of the synaptic weight $w_{ji}$, respectively.

The back-propagation training algorithm is a set of well defined rules in which the adjustment $\Delta w_{ji}$ to the synaptic weights $w_{ji}$ is formulated. During the algorithm the input is applied to the network and the distance between the desired output, $d$, and the actual output, $y$, is calculated and serves as an error measure. This error measure is then used to modify the synaptic weights of the network. The back-propagation training algorithm is outlined in the following steps [Lippman, 1987]:
Step 1.  *Initialise Weights and Thresholds*
Set all weights and neuron thresholds to small random values.

Step 2.  *Present Input and Desired Outputs*
Present the input vector \( x_1, x_2, x_3, \ldots, x_p \) and specify the desired outputs \( d_1, d_2, d_3, \ldots, d_p \). When the network is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The inputs are samples from a training set and are presented cyclically until the synaptic weights have stabilised.

Step 3  *Calculate Actual Outputs For the Network*
The algorithm at this stage calculates the total square error between the desired outputs, \( d_j \), and the corresponding actual outputs, \( y_j \)

\[
E = \Sigma (d_j - y_j)^2
\]

E is then used to test if the network has converged by testing to see if it is lower than a pre-set level. If the network has converged the algorithm stops but if E is larger than the pre-set value the algorithm continues.

Step 4  *Adapt Weights*
Use a recursive algorithm starting at the output nodes and work back to the first hidden layer. Adjust weights by

\[
w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j x_i
\]

In this equation \( w_{ji}(n) \) is the weight from the hidden node \( i \) or from an input node \( i \) to node \( j \) at time \( n \), \( x_i \) is either the output of node \( i \) or an input node \( i \), \( \eta \) is the learning rate and \( \delta_j \) is an error term for node \( j \). If node \( j \) is an output node, the

\[
\delta_j = y_j (1-y_j)(d_j-y_j)
\]

where \( d_j \) is the desired output of node \( j \) and \( y_j \) is the actual output.

If node \( j \) is an internal hidden node, then
\[ \delta_j = x_j (1-x_j) \sum \delta_k w_{kj} \]

where \( k \) refers to all the nodes in the layer after node \( j \).

**Step 5** Repeat by Going to Step 2

### 5.5.1 Training with Momentum

Momentum is a method for improving the training time of the backpropagation algorithm, while enhancing the stability of the process[1]. The process acts like a low-pass filter allowing the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum, with momentum the network can slide through such a minimum.

The momentum method involves adding a term to the weight adjustment that is proportional to the amount of the previous weight change. Once an adjustment is made it is 'remembered' and serves to modify all subsequent weight adjustments. Weight adjustment is implemented according to the formula [2]:

\[
w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji} + \alpha [ w_{ji}(n) - w_{ji}(n-1) ]
\]

where \( \alpha \), the momentum constant, is commonly set to around 0.9.

### 5.5.2 Adaptive Learning Rate

Training time can be decreased by the use of an adaptive learning rate which attempts to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

The procedure used in this section is based on the error ratio, which is the ratio of the current error versus the previous error. If the error ratio exceeds a predefined value (typically 1.05), the new weights and biases are discarded. In addition the learning rate is decreased, by multiply the learning rate by a number less than 1 (typically 0.7).
Otherwise the new weights, biases are kept. The learning rate will be increased, by multiplying by a number greater than 1 (typically 1.05), if the error ratio is less than 1.

This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus a near optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it is decreased until stable learning resumes.

5.6 Neural Network Defect Classifier

Figure 5.5 shows the classification system used to classify lake, bank and hot cylinder marks developed by the author during the course of the research. This system is based on the homomorphic filtering procedure developed in Section 4.3 with the filter’s output coefficients being feed to a Neural Network classifier. As the filter’s output is of two-dimensional format, the filtered coefficients have to be reordered into a one-dimensional format before being applied to the neural network. As shown in Figure 5.6 this is done by following a zigzag pattern through the coefficients. This leads to a one-dimensional vector whose elements are positioned according to there frequency, i.e. in ascending order.
Construction of Neural Network Input Vector

The neural network uses the filtered DCT coefficients from the homomorphic network described in Section 4.3 as inputs. For a 512-by-512 image this will require over a quarter a million coefficients. Not all these coefficients make a significant contribution to the representation of the image, hence the number of coefficients can be reduced without affecting the quality of the representation significantly [Clarke, 1995]. Figure 5.8 shows the construction of the images shown in 4.7 using only the coefficients indicated in Figure 5.7, comparing the two sets there is visible little difference between them. Hence a representation based coefficients indicated in Figure 5.7 is a good representation of an enhanced and contains sufficient information to describe the defects.

Coefficients indicated in Figure 5.7 are re-ordered into a one-dimensional vector the method indicated in Figure 5.6. Figure 5.9 shows the result of this procedure an image containing a lake mark. The reordered one dimensional vector contains
8511 elements requiring the neural network classifier to have 8511 source nodes (inputs). Note that the DC coefficient is also omitted from the vector inputted to the neural network classifier, this aims to give the classification process a measure of immunity to the colour of the polymer sheet.

Figure 5.7: Coefficients used by Neural Network
Figure 5.8: Homomorphic Filtered Using Select Coefficients
(a) Lake, (b) Bank and (c) Hot Cylinder Marks

Figure 5.9: Example Neural Network Input Vector
5.6.2 Training and Testing Neural Network

The neural network used to classify the defects is a multi-layered feedforward network with a single hidden layer. The hidden layer contains 20 neurons while the output contains 4 output neurons, all neurons have a sigmoid activation function as described in Equation 5.3. The four output neurons are used to indicate that an image presented to the classification process is good or contains one of the following 3 defects, lake, bank or hot cylinder marks. Figure 5.10 shows the structure of the neural network.

A database of images was collected for the purpose of training and testing the neural network shown in Figure 5.10. The database contains five images of each defect and five images with no defects, good sheets. The database is subdivided into training images, four of each type and test images, the remaining four images. Appendix 2 shows these images enhanced. Vectors used for the training and testing of the Neural
network are constructed from the database of images by the method described in Section 5.6.1.

A back propagation training algorithm was used to train the network, the training algorithm included the features of an adaptive learning rate and momentum. The parameters used by the algorithm are as follows:

- Initial Learning Rate: 0.01
- Learning Rate Increase: 1.05
- Learning Rate Decrease: 0.7
- Maximum Error Ratio: 1.04
- Momentum Constant: 0.9

Training was carried out for 300 epochs where training was interrupted at intervals of 25 epoch’s to test for misclassifications. Table 5.1 shows the performance of the network to both the training set and the test set. Figure 5.11 shows the variation of the mean squared error (MSE) and the learning rate during training.

<table>
<thead>
<tr>
<th>No. of epochs</th>
<th>No. of Misclassifications</th>
<th>Training Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No.</td>
<td>MSE</td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>0.1106</td>
<td>0.3326</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>0.0444</td>
<td>0.2107</td>
</tr>
<tr>
<td>75</td>
<td>1</td>
<td>0.0206</td>
<td>0.1434</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0.0072</td>
<td>0.0847</td>
</tr>
<tr>
<td>125</td>
<td>0</td>
<td>0.0057</td>
<td>0.0753</td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td>0.0028</td>
<td>0.0531</td>
</tr>
<tr>
<td>175</td>
<td>0</td>
<td>0.0014</td>
<td>0.0373</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>0.0010</td>
<td>0.0323</td>
</tr>
<tr>
<td>225</td>
<td>0</td>
<td>5.8e-4</td>
<td>0.0242</td>
</tr>
<tr>
<td>250</td>
<td>0</td>
<td>2.4e-4</td>
<td>0.0155</td>
</tr>
<tr>
<td>275</td>
<td>0</td>
<td>1.5e-4</td>
<td>0.0123</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>4.3e-5</td>
<td>0.0065</td>
</tr>
</tbody>
</table>

Table 5.1: Neural Network Results
Figure 5.11: Training Progress
6.0 The Wavelet and Wavelet Packet Transforms

6.1 Introduction

The wavelet transform has been applied to many applications such as data compression, image processing and image analysis as an alternative to Fourier based techniques. The Fourier transform measures the frequency content of a signal by matching the signal with swept sinusoids. This provides a pure frequency description of the signal which is not localized in time.

The Wavelet Transform measures the similarity between the signal and wavelets. Each wavelet element is a scaled version (either compressed or stretched) of a ‘mother’ wavelet. To determine the contribution or weight of each wavelet, a comparison is performed with the signal as the wavelet is varied in scale and translated in time or space. The Wavelet Transform results in the signal been mapped by varying two independent variables, time and scale [Bentley, 1994]. Having two variables to play with has advantages. Translating the wavelet with respect to the signal allows signal characteristics to be isolated in time or space, and scaling the wavelet enables the signal to be viewed over various durations giving alternative content views or resolutions [Young, 1994]. The wavelet transform provides a time-frequency description of the transformed signal which is localized in time.

This chapter examines the wavelet transform by examining common implementations for one-dimensional signals, then extending these procedures to incorporate two-dimensional signals.

6.2 The One-Dimensional Wavelet Transform
The wavelet transform of a signal \( x(t) \) is defined in Equation 6.1 [Bentley, 1994];

\[
\text{WT}(\tau, a) = \frac{1}{\sqrt{a}} \int x(t) \psi \left( \frac{t-\tau}{a} \right) \, dt \tag{6.1}
\]

where \( a \) is the scale and \( \tau \) is time the wavelet is translated and \( \psi(t) \) represents the wavelet. From Equation 6.1 it can be seen that the wavelet transform performs a decomposition of the signal, \( x(t) \), into a weighted set of scaled wavelet functions at different points in time. In general the wavelet, \( \psi(t) \), is a complex-valued function such as the Morlet wavelet. Figure 6.1 shows the real part of the Morlet wavelet at three different levels of scale. Examining the three wavelets, it can be seen that the wavelets are all versions of a common signal, obtained by compressing or stretching a common 'mother wavelet'.

Due to this observed scaling wavelets at high frequencies are of limited duration and wavelets at low frequencies are relatively longer in duration. This varying duration or window structure is reflected in the coverage of the time-frequency plane by the wavelet transform, as shown in Figure 6.2. These variable window length characteristics are suited to the analysis of signals containing short high-frequency components and extended low-frequency components, which is often the case for signals encountered in practice [Zettler, 1990].

![Figure 6.1: Real Component of Morlet Wavelet at Different Scales](image_url)
Two common implementations of the Wavelet Transform using this methodology are examined in this section, the filter bank and the ‘algorithme à trous’ algorithms.
6.3.1 The Filter Bank Implementation

The wavelet transform can be implemented using an iterated filter bank as shown in Figure 6.3. A highpass filter, $g$, and a lowpass filter, $h$, are used to split the input signal into high and low frequency sub-components [Cody, 1992]. The filters used are quadrature mirror filters which closely approximate complete frequency coverage. After filtering, the two components contain redundancies and it is valid to subsample each component by a factor of 2 without losing any information, every second sample is discarded [Bentley, 1994]. The output from the highpass leg provides the wavelet coefficients, and the lowpass leg provides the input (Residue Component) for the next iteration through the Discrete Wavelet Processing Block. The use of decimation on the lowpass filter’s output causes the sample rate to be halved which doubles the scale for the next stage. Implementing this filter process repeatedly results in a bank of bandpass filters spread logarithmically over frequency [Bentley, 1994] as shown in Figure 6.4.
Before the filter structure illustrated in Figure 6.3 can be considered as a wavelet transform, the filter coefficients must satisfy a number of conditions. One such family of filters is the Daubechies' filters, which contain an even number of coefficients [Nguyen, 1996]. Appendix 3 lists filter coefficients from the Daubechies' family of filters plus shows the scaling function associated with the lowpass filter and the wavelet function associated with the highpass filter.

Figure 6.5 shows the wavelet transform decomposition of a section taken from a speech signal. The signal is iterated through the discrete wavelet processing block 3 times to produce a 3-level wavelet decomposition using Daubechies' 4-tap filters. From this example we can see that the four components of the wavelet decomposition contain the same number of elements as the input which will always be the case for this type of implementation.
Reconstruction of a signal from its wavelet coefficients is achieved using operations similar to the wavelet transform decomposition process. Instead of decimation the signals are interpolated, zeros placed between coefficients, prior to being filtered.

The reconstruction process firstly interpolates the residue component and the highest scale component of the wavelet coefficients. The interpolated residue signal is convoluted with a lowpass filter and the interpolated wavelet coefficients are convoluted with a highpass filter. The filter coefficients used are the same as the filter coefficients used by the decomposition process only in the reverse order [Cody, 1994]. The filtered components are added to form an approximation of the original signal. This filtering process is repeated the same number of times the original signal was iterated through the processing block shown in Figure 6.3, with the approximation signal providing the input to the lowpass leg of each iteration and the wavelet coefficients at the appropriate scale providing the input to the highpass leg.
6.3.2 ‘Algorithme à trous’

This procedure also uses a filter bank except here the output components are each the same size as the input. This occurs because no decimation is implemented after filtering. Instead of the decimation of the lowpass and highpass components, the filters are interpolated by placing a number of zeros in between their coefficients. Figure 6.6 shows the filter bank structure used by the ‘Algorithme à trous’. For this structure to achieve the scale doubling achieved by the decomposition procedure of Section 6.3.1, the filters $h_p$ and $g_p$ are obtained by putting $2^{P-1}$ zeros between each of the coefficients of the filters $h$ and $g$ [Mallat, 1991] used by the decomposition procedure of Section 6.3.1. Figure 6.7 shows a 3 level decomposition using this a method.

![Figure 6.6 : Filter Bank Structure to Implement ‘Algorithme à trous’](image)
6.4 Two-Dimensional Wavelet Transform

The filter bank implementation discussed in Section 6.3.1 can be extended so that it can be applied to a two dimensional signal such as an image. This is achievable because the separability of image coordinates allows the image to be broken down into a series of one-dimensional sequences and allows one-dimensional filtering to be applied [Young, 1994].

The first series one-dimensional sequences are formed by taking the rows of an image. These one-dimensional sequences are individual subjected to a one level wavelet decomposition which essentially splits the sequences into a highpass band and a lowpass band. The resulting outputs from the filtering and subsampling of the horizontal sequences are used to form two sub-images.
A second series of one-dimensional sequences are formed by taking the columns from
the two sub-images. These vertical sequences are also subjected to a one level wavelet
decomposition which results in the formation of four sub-images. These sub-images
are the wavelet domain representation of the original image for a one level two-
dimensional decomposition. This process is shown in Figure 6.8 [Young, 1994]. To
obtain further levels of decomposition the process is repeated with the lowpass
residue, $A_j$, providing the input for further stages.

![Figure 6.8: Single Stage of 2-D Wavelet Decomposition](image)

Figure 6.9 shows the magnitudes of a one level wavelet decomposition for a test
image using Daubechies' 4-tap filters. Noting that if a lowpass filter is inputted with a
slow varying signal then the output will be large relative to the filter response to a
signal containing only fast variations, and opposite for a highpass filter. The
horizontal lines contain within the test image yield a horizontal sequence containing
mainly slow variations while the vertical and diagonal lines yield fast variations
within the one-dimensional horizontal sequences. Hence the band splitting carried out
on the horizontal sequences will yielded a lowpass component mainly influenced by
the horizontal lines, while the highpass component will be mainly contributed to by
the vertical and diagonal lines of the image.
These highpass and lowpass components are subjected to further one-dimensional band splitting but on this occasion the sequences are formed vertically. Hence vertical lines will produce slow varying signals while horizontal and diagonal lines will procedure fast variations.

The highpass component of Figure 6.9 (b) will produce two further sub-images. The sub-image $D_1$ which is the result of vertical and horizontal highpass filtering is mainly influenced by the diagonal lines, this sub-image is commonly referred to as the ‘diagonal detail subband’. The subimage $D_2$ which is the result of horizontal highpass filtering and vertical lowpass filtering is mainly influenced horizontal fast variations of the input image. This sub-image is commonly referred to as the ‘horizontal detail subband’.

The lowpass component of Figure 6.9 (b) is also subjected to the band splitting process which produces two further sub-images. The subimage $D_3$ is the result of lowpass horizontal filtering and highpass vertical filter whose main contributors vertical fast variations. This sub-image is commonly referred to as the ‘vertical detail subband’. The remaining sub-image $A_1$ is the result of lowpass filtering in both directions and produces a blurred version of the original and is commonly referred to as the ‘lowpass residue’. The lowpass residue provides the input to further levels of decomposition. The contents of all the sub-images is also influenced by the choice of wavelet and hence the subbands describe the image in terms of the wavelet used for decomposition.

Figure 6.10 shows the structure for a three level decomposition. The label uses subscripts to indicate the scale which will double for each iteration through the process indicated in Figure 6.8 [Mallat, 1989]. The superscripts indicate the particular subband at the given scale, i.e. 1 indicates the diagonal detail subband, 2 indicates the vertical detail subband, 3 indicates the horizontal subband. The letters A and D are used to distinguish the lowpass residue form the subbands. Figure 6.11 shows a 3-level two-dimensional wavelet decomposition for the a naturally occurring image, note that the wavelet decomposition equalized the subbands for easier viewing.
Figure 6.9: One Level Wavelet Decomposition of Test Image

Figure 6.10: Structure for a Three Level Decomposition
6.5 Wavelet Packets

Wavelet packets provide a decomposition tool where arbitrary time (or space) - frequency resolution which can be chosen according to the signal. The algorithm consist of an extended version of the wavelet transform procedure developed in Section 6.3.1. But unlike the wavelet transform implementations of Section 6.3.1, where only the lowpass residue is presented for further decomposition, the wavelet packet decomposition can decompose any of the subband components. Figure 6.12 shows a typical structure for implementation of a one-dimensional wavelet packet decomposition [Won-ya, 1995]. A primary concern for the implementation of the wavelet packet is which subband components should be decomposed to provide a decomposition that is optimal with respect to a given criterion. The wavelet packet implementation developed in this section attempts to find the near-best representation of the input signal where the parameter to be optimized is the number of coefficients representing a given fraction of the total signal energy [Taswell, 1995].
6.5.1 The Search Criterion

Firstly the vector to be evaluated is sorted so that $y(k)$ is the $k^{th}$ largest absolute value element of the vector. The decreasing absolute value sorted vector, $y(k)$, is raised to a predefined powered $p$, cumulatively summed, and normalized to form the vector $u(k,p)$. The formulae given in Equation 6.2 illustrates this procedure:

$$u(k,p) = \frac{v(k,p)}{v(N,p)} \quad \text{where} \quad v(k,p) = \sum_{i=1}^{k} y(i)^p$$  \hspace{1cm} (6.2)

Note that $0 \leq u(k,p) \leq 1$ because of the normalization. Using the vector $u(k,p)$ we can define the data compression number, $N^p_f$, [Taswell, 1994] in Equation 6.3:

$$N^p_f(k) = \arg\min_{f} |u(k,p) - f|$$  \hspace{1cm} (6.3)
where the power $p$ and the fraction $f$ are parameters chosen from the intervals $0 < p \leq 2$ and $0 < f < 1$. For example, choosing $p = 2$ and $f=0.99$ yields the minimum number of vector coefficients containing 99% of the energy of the entire vector.

6.5.2 The Search Method

The wavelet packet algorithm developed here uses a depth-first top-down search to determine the structure of the filter bank for a particular signal. The search terminates along any branch as soon as the cost of the children nodes is greater than the cost of the parent node. The children nodes refer to the decomposition components of a Discrete Wavelet Processing Block, Figure 6.3, while the parent node refers to the input to the Discrete Wavelet Processing Block. The cost of these nodes is evaluated by the data compression number, $NP_f$, developed in Section 6.5.1. Also the algorithm developed here is given a limit to the depth of the search [Taswell, 1995], i.e. the levels of decomposition.

6.5.3 One-Dimensional Wavelet Packet Example

A 512 sample section of a speech signal is decomposed using the wavelet packet algorithm developed in the Section 6.5 using Daubechies 8-tap filters, listed in Appendix 3. The parameters for the data compression number are, $p=2$ and $f=0.99$ and the decomposition was limited to 4 levels. Figure 6.13 shows the speech signal, its wavelet transform decomposition and the wavelet packet decomposition. The wavelet decompositions are represented on a normalized time-frequency axis. Figure 6.14 shows the structure of the filter bank implementation for this wavelet packet decomposition. From these figures it can be see that the wavelet packet decomposition differs from the wavelet transform decomposition for frequencies greater than 0.5. While the wavelet transform depicts a signal with frequencies spread broadly over the frequency range 0.5 to 1 for the complete duration of the signal, the wavelet packet indicates that the signal’s main contribution for this frequency range is relatively narrow about the frequency 0.8.
Figure 6.13: Example Wavelet Packet Decomposition
6.5.4 Two-Dimensional Wavelet Packet Example

The primary difference between the one-dimensional and the two-dimensional wavelet packet algorithms is that for the two-dimensional case each component (parent node) has four children components all of which can be subjected to further decomposition. Another difference is that the mean of the image is calculated and subtracted from the image [Taswell, 1995]. This ensures that edges, which are visual important, are consider as high energy components. The search method and search criterion described in Sections 6.5.1 and 6.5.2 are used by the two-dimensional wavelet packet decomposition.
Figure 6.15 illustrates the wavelet packet decomposition for an example image, the image shown is prior to the subtraction of the image's mean. The parameters for the wavelet packet algorithm are $p=2$, $f=0.99$ and a maximum depth of 4 levels. Decomposition is achieved using Daubechies 4-tap filters as listed in Appendix 3. The wavelet packet subbands which result from the decomposition are indicate in Figure 6.15 (c).

From the Figure 6.15 it can be seen that the wavelet packet algorithm developed here yields a decomposition that differs from the wavelet algorithm developed earlier. The differences mainly occur in the vertical detail subband. The wavelet packet algorithm provides a better representation than the wavelet transform in the sense that fewer coefficients contain the same amount of energy. This is due to the particular search criterion used in this section and different search criterion may yield different representations. Also a better representation is not always provided by the wavelet packet algorithm, for the image shown in Figure 6.11 the wavelet packet decomposition with $p=2$, $f=0.99$ and a maximum depth of 4 levels, results in the same decomposition as the wavelet transform.
Figure 6.15: Two-Dimensional Wavelet Packet Example

(a) Input Image

(b) Wavelet Packet Coefficients

(c) Wavelet Packet Subbands
6.6 Image Compression using Wavelets

The application of signal compression using the wavelet transform is explored within this section with the objective of examining the properties and the information distribution of the wavelet coefficients. As interest is mainly directed at the wavelet coefficients there is the omission of quantisation and entropy encoding usually associated with practical compression techniques.

The compression is achieved by setting coefficients with magnitudes less than a threshold value to zero, hence omitting these coefficients from the image representation [Clarke, 1995]. Comparisons between different representations are made based on differences between reconstructed and original images.

A commonly used standard for still image compression is the JPEG algorithm which is based on a two-dimensional Direct Cosine Transform (DCT) of 8-by-8 sub-images [Gonzalez, 1992]. A system based on JPEG algorithm which divides the original image into 8-by-8 pixel blocks and applies a DCT on each of the individual blocks (Block DCT) was used to compare with wavelet techniques. Prior to any transformation, the mean of the image is calculated and subtracted from the image, this ensures that the visually important edges are considered as high energy components.

Throughout this chapter three error measures given by Equations 6.4, 6.5 and 6.6 are used to measure the difference between the reconstructed and original image [Nguyen, 1996].

\[ \text{Mean Square Error} \quad \text{MSE} = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} |f(x,y) - \tilde{f}(x,y)|^2 \] (6.4)

\[ \text{Peak Signal Noise Ratio} \quad \text{PSNR} = 10 \log_{10} \left( \frac{2^b - 1}{\text{MSE}} \right) \] (6.5)

\[ \text{Maximum Error} \quad \text{MaxError} = \max f(x,y) - \tilde{f}(x,y) \] (6.6)
Where \( f(x,y) \) is the original image of size \( M \)-by-\( N \) using \( b \) bits to represent each pixel value and \( \tilde{f}(x,y) \) is an estimation of the original image.

6.6.1 Image Compression using Wavelet Transform

The wavelet transform filter bank implementation developed in Section 6.4 was used to decompose the image shown in Figure 6.11 to 6 levels using Daubechies 4-tap filters as listed in Appendix 3. Compression is achieved by omitting a number of the wavelet coefficients as indicated in Section 6.6. A variable percentage of the coefficients where maintained by adjusting the magnitude of the threshold below which coefficients are omitted.

The reconstructed image form the selected wavelet coefficients is compared with the reconstructed image based on the 'Block DCT' representation containing the same number of coefficients. Table 6.1 compares the error measurements for the compression of the input image shown in Figure 6.11 using the wavelet transform and the 'Block DCT' coefficients.
The wavelet transform provides better error measurements for all the examples indicated in table 6.1, but this doesn’t indicate the perceptual quality of the image. For PSNR values greater than 32 dB’s the perceptual quality of the images between the two systems is virtually unnoticeable, Figure 6.16 shows the reconstructed images from both representations when using 10% of the coefficients. When the PSNR value drops below 32 dB’s the perceptual quality between the images becomes more noticeable with the reconstructed image from the wavelet coefficients providing the better image [Nguyen, 1996]. Figure 6.17 shows the reconstructed images when using 3% of the coefficients.

<table>
<thead>
<tr>
<th>Percentage of Non-Zero Coefficients</th>
<th>Wavelet Transform</th>
<th>Block DCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>MSE 8.7636</td>
<td>9.1532</td>
</tr>
<tr>
<td></td>
<td>PSNR 38.738</td>
<td>38.5491</td>
</tr>
<tr>
<td></td>
<td>MaxError 13.6920</td>
<td>29.8784</td>
</tr>
<tr>
<td>15%</td>
<td>MSE 14.8165</td>
<td>15.2151</td>
</tr>
<tr>
<td></td>
<td>PSNR 36.4574</td>
<td>36.3421</td>
</tr>
<tr>
<td></td>
<td>MaxError 21.7764</td>
<td>39.864</td>
</tr>
<tr>
<td>10%</td>
<td>MSE 28.3072</td>
<td>28.9545</td>
</tr>
<tr>
<td></td>
<td>PSNR 33.6458</td>
<td>33.5476</td>
</tr>
<tr>
<td></td>
<td>MaxError 35.8571</td>
<td>52.9062</td>
</tr>
<tr>
<td>5%</td>
<td>MSE 66.8425</td>
<td>70.3627</td>
</tr>
<tr>
<td></td>
<td>PSNR 29.9143</td>
<td>29.6914</td>
</tr>
<tr>
<td></td>
<td>MaxError 58.721</td>
<td>92.1875</td>
</tr>
<tr>
<td>3%</td>
<td>MSE 107.8971</td>
<td>123.1874</td>
</tr>
<tr>
<td></td>
<td>PSNR 27.8347</td>
<td>27.2591</td>
</tr>
<tr>
<td></td>
<td>MaxError 93.8178</td>
<td>131.6211</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of Wavelet and Block DCT
Figure 6.16: Using 10% of Coefficients

Figure 6.17: Using 3% of Coefficients
A common distortion caused by the block DCT is visible in Figure 6.17 commonly referred to as blocking artifacts. This is the result of omitting basis functions which do not decay to zero at boundaries, hence creating discontinuities between blocks [Nguyen, 1996]. This is overcome by wavelets which decay to zero, see Appendix 3 to view some wavelet functions.

Images consist mainly of large low-frequency regions, separated generally by short edges or boundaries [Zettler, 1990]. Intuitively, one would like to represent the short edges or boundaries with short functions and the large low-frequency regions with long functions. This is offered by the wavelet transform. The low frequency basis functions are formed by a cascade of interpolated versions of the lowpass filter, $h$, hence it’s length is large. Higher frequency bands are iterated less, the basis functions become shorter [Nguyen, 1996]. Figure 6.18 shows the selected coefficients from the wavelet domain used in the reconstruction of the image shown in Figure 6.16, i.e. when only 10% of the coefficients were used. From Figure 6.18 we can see that the coefficients selected in the subbands representing high frequencies, i.e. at low scales, correspond to boundary locations within the image. As the scale increases the selected coefficients become more dense and correspond to the low-frequency slow varying regions.

Figure 6.18: Selected Coefficients from Wavelet Decomposition
6.6.2 Image Compression using Wavelet Packets

The wavelet packet decomposition method developed in Section 6.5 attempts to find a representation of an image which minimizes the number of coefficients to represent a predefined fraction of the image's energy. Hence if the wavelet packet algorithm provides an alternative representation than the wavelet transform, more energy will be packed into the same number of coefficients and there will be less error associated with the reconstructed image.

The input image shown in Figure 6.11 was subjected to a wavelet packet decomposition as described in Section 6.5. The parameters for the wavelet packet decomposition are, \( p=2, f=0.95 \) and a maximum depth of 4 decomposition levels using Daubechies' 4-tap filters. This decomposition yield the wavelet packet coefficients as displayed in Figure 6.18(a) with Figure 6.18(b) showing the wavelet packet subbands.

This decomposition yields a representation which has 95% of it's energy contained within 2,460 (3.75%) wavelet packet coefficients. Comparing this to the wavelet decomposition of the same image, the largest 2,460 wavelet coefficients contained 94.83% of the original image's energy. Table 6.2 compares the error measurements of reconstructions based on the largest 2,460 coefficients for the wavelet, wavelet transform and block DCT decompositions. For this particular image the wavelet packet provides a better reconstructed image, this was generally the case for images subjected to the wavelet packet decomposition described in Section 6.5.

<table>
<thead>
<tr>
<th></th>
<th>Maximum Error</th>
<th>MSE</th>
<th>PSNR (dB's)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet Packet</td>
<td>68.08</td>
<td>86.07</td>
<td>28.78</td>
</tr>
<tr>
<td>Wavelet Transform</td>
<td>87.14</td>
<td>89.07</td>
<td>28.63</td>
</tr>
<tr>
<td>Block DCT</td>
<td>101.94</td>
<td>96.75</td>
<td>28.27</td>
</tr>
</tbody>
</table>

Table 6.2: Comparison of Reconstructions based on 3.75% of Coefficients
Figure 6.18: Wavelet Packet Decomposition

(a) Wavelet Packet Coefficients

(b) Wavelet Packet Subbands
7.0 Classification based on Wavelet Zero-Crossings

7.1 Introduction

Examining the defective sheets displayed in the Image Database of Appendix 2 we see that vertical ridges play a crucial role in the description of the defects. Hence a description of the defects based on the vertical ridges will provide a representation which can be used to discriminate the defects.

The sharp variations of a signal amplitude are generally among the most meaningful features of a signal, for example the discontinuities of an image intensity provide the contours of different objects. A common method for detecting the discontinuities of an image is to detect the zero-crossings of the image's second-order derivative. This chapter examines the use of a particular wavelet filter pair, obtained by defining the wavelet function as the second derivative of a smoothing function, to provide wavelet coefficients. From these wavelet coefficients a representation of the defects can be extracted and used in the classification of the surface defects.

7.2 Wavelet Zero-Crossings

The zero-crossings of a wavelet decomposition using wavelets normally associated with the Wavelet Transform, such as the Daubechies family of wavelets, do not relate to the input signal in a clear manner. If the analysing wavelet is specified as the second derivative of a smoothing function, the zero-crossings of the wavelet coefficients are more closely related to the locations of the sharp variations of the
input signal [Mallat, 1991]. Table 7.1 gives the lowpass and highpass filter coefficients of such a wavelet, the filters are symmetric with respect to time (about \( n = 0 \)). Figure 7.1 shows the scaling and wavelet functions associated with the filter coefficients of Table 7.1.

Figure 7.2 shows the 3-level wavelet decomposition of a one-dimensional signal containing different types of variations using the filters specified in Table 7.1. From Figure 7.2 we see that a prominent zero-crossing exists on scale 1 at the location of the fastest variation, as the scale increases this zero-crossing becomes less prominent and the crossings at the locations of the slower variations become more prominent. Using the wavelet associated with the filter coefficients on Table 7.1, a wavelet decomposition will provide zero-crossings related to the variations of the input signal. The scale on which the zero-crossing is prominent indicates the speed of the variation.

<table>
<thead>
<tr>
<th>( n )</th>
<th>Lowpass</th>
<th>Highpass</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4347</td>
<td>0.7118</td>
</tr>
<tr>
<td>1</td>
<td>0.2864</td>
<td>-0.2309</td>
</tr>
<tr>
<td>2</td>
<td>0.0450</td>
<td>-0.1120</td>
</tr>
<tr>
<td>3</td>
<td>-0.0393</td>
<td>-0.0226</td>
</tr>
<tr>
<td>4</td>
<td>-0.0132</td>
<td>0.0062</td>
</tr>
<tr>
<td>5</td>
<td>0.0032</td>
<td>0.0039</td>
</tr>
</tbody>
</table>

Table 7.1: Filter Coefficients

Corresponding to Wavelet based on Second Derivative of Smoothing Function

Figure 7.1: Scaling and Wavelet Functions
7.3 Ridge Detection within One-Dimensional Signals

For the analysis of surface defects on extruded polymer sheets it was noticed that the defects, after enhancement by homomorphic filtering (described in Section 4.3), are constructed mainly by the formation of vertical ridges. Hence ridge detection could play a central role in providing descriptors for a classification system.

Figure 7.3 shows an artificially created ridge within a one-dimensional signal plus the wavelet coefficients of a 4-level wavelet decomposition using the wavelet described in Section 7.2. From the resulting wavelet coefficients it can be observed that a negative to positive zero-crossing occurs on all scales of the wavelet coefficients for the positive transition of the ridge. While a positive to negative zero-crossing occurs for the negative transition of the ridge. Hence scanning the wavelet coefficients for these
transitions can be used to indicate the presence of a ridges within a one-dimensional signal.

Figure 7.4 shows the wavelet coefficients resulting when a small amount of Gaussian noise is added to the artificially created ridge. It is observed from these coefficients that the noise component introduces unwanted zero-crossings into the wavelet coefficients. To combat these unwanted zero-crossings some form of noise reduction needs to be performed. A method commonly used in conjunction with the wavelet transform for the denoising of Synthetic Aperture Radar (SAR) images is known as Soft Thresholding.

Soft Thresholding or 'wave shrinkage' involves the application of the non-linear function defined in Equation 7.1 to the wavelet coefficients, note the function is not applied to the residue component of the wavelet transform [Odegard, 1995];

\[
\begin{align*}
    w_a &= \begin{cases} 
    w_p - t & \text{for } w_p > t \\
    0 & \text{for } -t \leq w_p \leq t \\
    w_p + t & \text{for } w_p < -t
    \end{cases} \\
\end{align*}
\]  

where \(w_p\) are the wavelet coefficients prior to soft thresholding, \(w_a\) the wavelet coefficients after soft thresholding and \(t\) a threshold which is calculated for each scale component using Equation 7.2 [Wei, 1995];

\[
t = \sqrt{2\log(N)}\sigma
\]

where \(N\) is the number of wavelet coefficients and \(\sigma\) is the standard deviation of the coefficients.

Figure 7.5 shows the impact of the soft thresholding method on the noisy wavelet coefficients shown in Figure 7.4. If we consider crossings between negative and non-negative values, the transition from a negative to non-negative value followed by the
transition from a non-negative to negative can be used to indicate the presence of a ridge within the input signal.

Figure 7.3: Wavelet Decomposition of Artificially Created Ridge

Figure 7.4: Wavelet Decomposition for Noisy Input
7.4 Ridge Detection within Images

The procedures described in Section 7.3 can be extended for two-dimensional inputs using the wavelet described in Section 7.2. As the images to be consider contain mainly ridges of a vertical nature only the ‘horizontal detail subband’ of each scale needs to be considered. Also because of this characterisation we need only scan the ‘horizontal detail subband’ in a horizontal direction for the presence of a ridge.

The homomorphic filtered images (as described in Section 4.3), are subjected to a 3-level wavelet decomposition using the wavelet function described in Section 7.2. Soft thresholding, similar to that described in Section 7.3, is applied to the ‘horizontal detail subband’ of each scale. The selection of the threshold differs from the selection implemented by Equation 7.2. for the lowest scale, an experimentally fixed threshold is used. This threshold is selected so that zero-crossing associated with non-ridge
features (noise) of the input image are reduced to zero. The thresholds for the two remaining scales are calculated as follows.

The standard deviation ($\sigma_1$) is calculated for the 'horizontal detail subband' of an individual scale. A second standard deviation ($\sigma_2$) is obtained where the coefficients greater than $C\sigma_1$ ($C$ is a constant) are not considered. These coefficients are likely to be outliers associated with large ridges. The threshold used by the soft thresholding nonlinearity of Equation 7.1 is now calculated as $K\sigma_2$ ($K$ is a constant). The constants $C$ and $K$ where determined experimentally as 2 and 1.5 respectively for the each of the scales, but these constants can be scale variable. Figure 7.6, 7.7 and 7.9 shows the resultant 'horizontal detail subbands', after soft thresholding, for an images containing a lake, bank and hot cylinder marks respectively. From these images we can see the zero-crossings associated with the ridges for each type of defect.

![Figure 7.6: Soft Thresholding of Wavelet Coefficients for Lake Mark](image-url)
A binary image is constructed from the denoised 'horizontal detail subbands' of the individual scales. Each row of the wavelet component is scanned for a negative to
non-negative transition followed by a non-negative to negative transition. This pattern indicates the presence of a one dimensional ridge. The ridge is marked at the midpoint between the two transitions by the value one, and zero elsewhere. Figures 7.9, 7.10 and 7.11 show the binary images created form the denoised wavelet coefficients displayed in Figures 7.6, 7.7 and 7.8. It can be seen from these figures that the ridges in the homomorphic filtered images are well represented in the binary images created form the denoised wavelet coefficients. Also present in the binary images are ridge marks not associated with ridges on the homomorphic filtered image, but these are generally short. The binary image created from scale 4 associated with the image containing a bank mark is distorted, this is due to the close proximity of the ridges in the original image.

Figure 7.9: Binary Images Created by Lake Mark
Figure 7.10: Binary Images Created by Bank Marks

Figure 7.11: Binary Images Created by Hot Cylinder Marks
7.5 Classification of the Defects

Classification of the defects is implemented by extracting descriptors from the binary images created by the process developed in Section 7.4. As there exists ridge marks not associated with ridges of the defect and spurious breaks, further processing needs to be implemented prior to the extraction of the defects.

The isolated short ridge marks are removed from the binary images, using morphological algorithms, to reduce the number of marks not associated with ridges of the defect. Short breaks between ridge marks are linked, the linking algorithm joins the ends of two ridge marks if two conditions are meet. Firstly the ends of both ridge marks are within a predefined neighbourhood, e.g. a 5-by-5 rectangular neighbourhood, and the ridge marks have similar directions.

The binary image created from scale 4 of the wavelet coefficients of an image containing no defects, will have no ridge marks. Hence if this binary image contains no ridge marks, the classification system can infer that the image is taken from a 'good sheet'. On the other hand if this binary image contains ridge marks the classification system infers the presence of a defect. The remaining binary images extracted from the wavelet coefficients of scales 1 and 2 are used to determine the type of defect.

Two sets of chain codes are extracted from the binary images of scales 1 and 2. These chain codes provide a one-dimensional descriptor of the ridge marks. Classification of the defect is carried out by examining statistical values obtained from the chain codes plus shape features. The characteristics exploited are as follows, bank marks contain a relatively large number of ridges that are similar in lengths and long. The ridges of lake marks are of different orientations and lengths while ridge marks associated with hot cylinder marks are relatively short.
The classification system developed within this section was subjected to the twenty images content within Appendix 2. Six of the images within this database resulted in misclassifications, with the remainder providing correct classification. Two of these were caused by good sheets, the top right good sheet of the training set and the good sheet from the test set. In both cases the misclassification was caused by imperfections on the sheet which were not categorised as defects. This occurrence suggests the ability of this classification system to be expanded, so that other types of defects can be classified.

The remainder of the misclassifications were the bottom right lake mark of the training set. The ridge marks provided by this image were all short and hence classified as hot cylinder marks. The bottom right bank mark of the training set, ridge marks within the binary images contented many large breaks. This resulted in the ridge marks being small and the system also classified these as hot cylinder marks. Two of the hot cylinder marks from the training set were identified as lake marks, this was caused by the linking algorithm joining ridge marks of close proximity. Based on these misclassifications it is suggested that an improvement can possible be made to the classification system by altering the linking algorithm. One possibility is to use a global processing algorithm such as the Hough transform.
8.0 Discussion

This thesis has described the application of image processing techniques to raw images of extruded polymer sheets containing surface defects with the aim of classifying the defects. A variety of techniques are applied including spatial-domain, frequency-domain as well as emerging space/frequency domain algorithms. From the results of experiments described within this thesis, several of these procedures provided encouraging results on initial testing.

The spatial-domain technique of Dynamic Thresholding described in Section 3.2 provides a promising method for the detection of surface defects on embossed polymer sheets. As this procedure is a spatial-domain technique real-time implementation is a viable option. The Dynamic Thresholding algorithm is an automatic localised thresholding procedure that can adapt to variations in colour of the embossed polymer sheet. The procedure segments the defect from the remainder of the sheet and is adaptable to defects other than the examples given in Chapter 3. Also, parameters such as sub-image size and overlap of sub-images could be varied to provide better performance.

Chapter 4 described a novel homomorphic filtering enhancement algorithm which incorporates the Direct Cosine Transform. This novel approach provides significant enhancement of the surface defects occurring on non-embossed polymer sheets. A classification system is developed which uses the filtered Direct Cosine Transform coefficients as the input to a neural network classifier, (Chapter 5). The results for the classification system indicate the ability of the system to distinguish between images of non-embossed extruded polymer sheets containing lake, bank, hot cylinder marks and with no defects. The Direct Cosine Transform, which is central to the enhancement process, is implemented in a similar manner to the Fast Fourier Transform and the parallel structure of the neural network make this classification system a possible candidate for real time implementation.
The third procedure, described in Chapter 7, is based on the emerging Wavelet Transform. This classification system capitalises on the characteristics of a specialised wavelet function defined as the second derivative of a smoothing function. To the knowledge of the author this technique has not been applied to this type of problem previously. The raw image of a sheet with a surface defect is pre-processed using a homomorphic filtering procedure to enhance the defect, as described in Section 4.3. Following pre-processing, the image is analysed by a Wavelet Transform using the specialised wavelet. Noise contained within the image causes additional zero-crossings; to combat the effects of this noise, soft thresholding is applied to the wavelet coefficients with a fixed threshold at the lowest scale and an automatic thresholding procedure for the remaining scales. After soft thresholding, the wavelet coefficients are scanned for specific patterns of zero-crossings to indicate the presence of ridges at each scale and a binary image is generated to represent the centre line of the ridge. From the binary images, the structure of the ridges can be determined and used to provide descriptors for classification of the surface defects.

Central to this procedure is the use of a wavelet which is the second derivative of a smoothing function [Mallat, 1989] and the resultant zero-crossings of the wavelet coefficients. Another technique which produces zero-crossings involves the convolution of the Laplacian Operator with an image, explained in [Gonzalez, 1992]. Comparing these two techniques it is observed that the wavelet transform technique performs more operations depending on the size of the filters used to implement the wavelet transform and the number of decomposition levels. For the specific case described in Chapter 7 it is estimated that the wavelet transform technique performed almost twice as many operations as the technique using the Laplacian Operator. As a result of these extra operations pattern recognition techniques have access to more meaningful information as the zero-crossings are orientation and scale dependent. This characteristic is exploited by the procedure developed in Chapter 7.

Preliminary results for classification of the surface defects are promising, showing potential for the system to be expanded to include more types of surface defects.
wavelet decomposition at the centre of this final system can be implemented with a similar number of operations as the Discrete Cosine Transform making this classification system a viable option for real time implementation.

It should be noted that the database used for training and testing of the classification system contained 20 images, giving a good performance of the classification system. However, further evaluation would need to be carried out using a larger database to obtain a more accurate indication of the performance of the system.
References


Bibliography

Appendix 1

Spatial Filters

Lowpass Filters

(Coefficients are positive and the sum of the coefficients equals one)

\[
\begin{bmatrix}
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
\end{bmatrix} \quad \begin{bmatrix}
1/16 & 1/8 & 1/16 \\
1/8 & 1/4 & 1/8 \\
1/16 & 1/8 & 1/16 \\
\end{bmatrix} \quad \begin{bmatrix}
0.1 & 0.1 & 0.1 \\
0.1 & 0.2 & 0.1 \\
0.1 & 0.1 & 0.1 \\
\end{bmatrix}
\]

3 x 3 Averaging Filter

Lowpass 2

Lowpass 3

\[
\begin{bmatrix}
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
\end{bmatrix}
\]

5 x 5 Averaging Filter

Highpass Filters

(Coefficients can be negative or positive and the sum of the coefficients equals zero)

\[
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1 \\
\end{bmatrix} \quad \begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0 \\
\end{bmatrix} \quad \begin{bmatrix}
1 & -2 & 1 \\
-2 & 4 & -2 \\
1 & -2 & 1 \\
\end{bmatrix}
\]

Highpass 1

Laplacian

Highpass 3

92
**Highpass Filters with D.C. Bias**

(Coefficients can be negative or positive but the sum of the coefficients is a non-zero number)

\[
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & 9 & -1 \\
-1 & -1 & -1
\end{bmatrix} \quad \begin{bmatrix}
1 & -2 & 1 \\
-2 & 5 & -2 \\
1 & -2 & 1
\end{bmatrix}
\]

Filter 1 \hspace{1cm} Filter 2

**Sobel Edge Enhancement Filters**

(Specialised highpass filters)

\[
\begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix} \quad \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix}
\]

Horizontal \hspace{1cm} Vertical
Appendix 2

Image Database

This appendix consists of enhanced images of polymer sheets used in the training and testing of the neural network defect classifier, described in Section 5.6. The images are displayed in an ordered manner based on the images presence in the training set or test set.
Training Set

Lake Marks
Bank Marks
Hot Cylinder Marks
Test Set

Lake Mark

Hot Cylinder Marks

Bank Mark

Good Sheet
Appendix 3

Daubechies’ Wavelets

This appendix gives filter coefficients of the highpass, $g$, and lowpass, $h$, associated with some of Daubechies’ Wavelets. These filters are used when implementing a Discrete Wavelet Transform as described in Chapter 6. For each pair of filters the associated scaling and wavelet functions are displayed.

4-tap Filters

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h(n)$</td>
<td>-0.1294</td>
<td>0.2241</td>
<td>0.8365</td>
<td>0.4830</td>
</tr>
<tr>
<td>$g(n)$</td>
<td>-0.4830</td>
<td>0.8365</td>
<td>-0.2241</td>
<td>-0.1294</td>
</tr>
</tbody>
</table>
**6-tap Filters**

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n)</td>
<td>0.0352</td>
<td>-0.0854</td>
<td>-0.1350</td>
<td>0.4599</td>
<td>0.8069</td>
<td>0.3327</td>
</tr>
<tr>
<td>g(n)</td>
<td>-0.3327</td>
<td>0.8069</td>
<td>-0.4599</td>
<td>-0.1350</td>
<td>0.0854</td>
<td>0.0352</td>
</tr>
</tbody>
</table>
8-tap Filters

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n)</td>
<td>-0.0106</td>
<td>0.0329</td>
<td>0.0308</td>
<td>-0.1870</td>
<td>-0.0280</td>
<td>0.6309</td>
<td>0.7148</td>
<td>0.2304</td>
</tr>
<tr>
<td>g(n)</td>
<td>-0.2304</td>
<td>0.7148</td>
<td>-0.6309</td>
<td>-0.0280</td>
<td>0.1870</td>
<td>0.0308</td>
<td>-0.0329</td>
<td>-0.0106</td>
</tr>
</tbody>
</table>
**10-tap Filters**

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n)</td>
<td>0.0033</td>
<td>-0.0126</td>
<td>-0.0062</td>
<td>0.0776</td>
<td>-0.0322</td>
</tr>
<tr>
<td>g(n)</td>
<td>-0.1601</td>
<td>0.6038</td>
<td>-0.7243</td>
<td>0.1384</td>
<td>0.2423</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n)</td>
<td>-0.2423</td>
<td>0.1384</td>
<td>0.7243</td>
<td>0.6038</td>
<td>0.1601</td>
</tr>
<tr>
<td>g(n)</td>
<td>-0.0322</td>
<td>-0.776</td>
<td>-0.0062</td>
<td>0.0126</td>
<td>0.033</td>
</tr>
</tbody>
</table>

**Scaling Function**

**Wavelet Function**

103
Application of the Wavelet Transform to the Classification of Surface Defects on Extruded Polymer Sheets

D. Kiernan, E. Ambikairajah and P. Rowan

School of Engineering, Regional Technical College, Athlone, Co. Westmeath, Ireland

Abstract

The Discrete Wavelet Transform has been successfully used in many image processing applications, including image compression and denoising. This paper examines the use of a particular wavelet which relates the locations of an image’s sharp variations at different scales to the zero-crossings of the wavelet coefficients. The use of the wavelet for the classification of surface defects on extruded polymer sheets is described.

1 Introduction

There are three types of surface defects commonly associated with extruded polymer sheets, namely ‘bank marks’, ‘lake marks’ and ‘hot cylinder marks’. The classification system developed within this paper deals with these defects as well as determining sheets that contain none of these defects, (‘good sheet’). The defects are mainly characterised by the formation of vertical ridges.

The raw image of a surface defect is visually undetectable, thus requiring a pre-processing step to enhance the defect. The pre-processing step used is a homomorphic filtering procedure [Gonzalez, 1992], which enhances the ridges of the defects. Figure 5 shows an example of an enhanced bank mark while Figure 9 shows an example of an enhanced lake mark. A procedure, which can provide descriptors based on these ridges will provide a good description of the defect that can be used to classify the surface defects.

In this paper, the classification system is developed by firstly considering the response of the one-dimensional wavelet transform using the wavelet described in Section 2. To demonstrate aspects of the classification system an artificial ridge, with and without additive noise, is created and inputted to the one-dimensional wavelet transform. The one-dimensional responses provide us with viable methods for noise reduction and ridge detection which are extended to two-dimensions and, with further development, are applied to enhanced images containing defects to provide an efficient description of these defects. This description is used in the classification of the defects.

2 Wavelet Function

A feature which could be exploited in the analysis of signals is the zero-crossings of the wavelet transform output [Mallat, 1991], but the relationship between a particular transforms output’s zero-crossings and the input signal may not be clear for the
wavelets normally associated with the discrete wavelet transform, such as the Daubechies family of wavelets.

Specifying the analysing wavelet as the second derivative of a smoothing function causes the zero-crossings of the wavelet coefficients to be related more closely to the locations of the image’s sharp variations at different scales, hence providing a meaningful feature by which a signal can be characterised. Table 1 gives the lowpass and highpass filter coefficients used by the Discrete Wavelet Transform for such a wavelet; both filters are symmetrical with respect to time. Figure 1 shows the scaling and wavelet functions.

<table>
<thead>
<tr>
<th>n</th>
<th>Lowpass</th>
<th>Highpass</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4347</td>
<td>0.7118</td>
</tr>
<tr>
<td>1</td>
<td>0.2864</td>
<td>-0.2309</td>
</tr>
<tr>
<td>2</td>
<td>0.0450</td>
<td>-0.1120</td>
</tr>
<tr>
<td>3</td>
<td>-0.0393</td>
<td>-0.0226</td>
</tr>
<tr>
<td>4</td>
<td>-0.0132</td>
<td>0.0062</td>
</tr>
<tr>
<td>5</td>
<td>0.0032</td>
<td>0.0039</td>
</tr>
</tbody>
</table>

Table 1: Filter Coefficients Corresponding to the Wavelet

3 One-Dimensional Wavelet Transform

Using the wavelet described in Section 2 the Discrete Wavelet Transform algorithm [Mallat, 1989] is applied to the artificially created ridge in Figure 2. From the resulting wavelet coefficients shown in Figure 2 it can be observed that a negative to positive zero-crossing occurs on all scales of the wavelet coefficients for the positive transition of the ridge while a positive to negative zero-crossing occurs for the negative transition of the ridge. A noise signal is added to the artificially create ridge and subjected to four-level wavelet transform. Figure 3 shows the wavelet coefficients. It can be seen that the noise introduces additional zero crossings. In order to eliminate the additional unwanted zero crossings a soft thresholding noise reduction method is applied to the wavelet coefficients. This is described in Section 4.

4 Soft Thresholding

Soft thresholding is a method of noise reduction that has been applied successfully for denoising Synthetic Aperture Radar images [Wei, 1996]. An advantage of the soft thresholding method is that sharp
features are preserved. The method involves the application of the non-linear function defined in Equation 1 to the wavelet coefficients. However, this is not applied to the residue component of the wavelet decomposition.

\[
v_A = \begin{cases} 
v_p - t & \text{for } v_p > t \\ 0 & \text{for } -t \leq v_p \leq t \\ v_p + t & \text{for } v_p < -t \end{cases} \quad (1)
\]

where \( v_p \) are the wavelet coefficients prior to soft thresholding, \( v_A \) the wavelet coefficients after soft thresholding and \( t \) a threshold which is calculated for each scale component using Equation 2.

\[
t = \lfloor \sqrt{2 \log(N)} \rfloor \sigma \quad (2)
\]

where \( N \) is the number of wavelet coefficients and \( \sigma \) is the standard deviation of the coefficients.

Figure 4 shows the impact of the soft thresholding method on the noisy wavelet coefficients shown in Figure 3. If we consider crossings between negative values and non-negative values, the crossings associated with the positive and negative transitions of the ridge remain intact except for those at scale 1, while most of the crossings associated with the noise have been eliminated.

5 Ridge Detection Within Images

The procedures described previously can be extended for two-dimensional inputs using the wavelet described in Section 2. As the images contain ridges of a mainly vertical nature only the 'horizontal fine detail' component of each scale needs to be considered.

The enhanced image is subjected to a 3-level wavelet decomposition. Soft thresholding, similar to that described in Section 4, is applied to the horizontal fine detail components of each scale. The selection of the threshold differs from the selection carried out by Equation 2. For the lowest scale, an experimentally fixed threshold is used. The threshold for the two remaining scales are calculated as follows.

The standard deviation \( (\sigma) \) is calculated for the horizontal fine detail component of an individual scale. A second standard deviation \( (\sigma') \) is obtained where the coefficients greater than \( C.\sigma \) (\( C \) is a constant) are not considered. These coefficients are likely to be outliers associated with very large ridges. The threshold used by the soft thresholding nonlinearity of Equation 1 is now calculated as \( K.\sigma \), (\( K \) is a constant). The constants \( C \) and \( K \) are determined experimentally and can be scale variable.

A binary image is constructed from the denoised horizontal fine detail components of the individual scales. Each row of the wavelet component is scanned for a negative to non-negative transition followed by a non-negative to negative transition. This pattern indicates the presence of a one-dimensional ridge. The ridge is marked at the midpoint between the two transitions by the value one, and zero elsewhere.
Figure 5 shows a homomorphic filtered image containing a surface defect with Figures 6, 7 and 8 showing the binary images at each scale created by the procedure described in this section. It can be seen from these Figures that the ridges of the defect are well represented in the binary images. The binary image created from scale 4 is distorted due to the lowpass filtering of the scaling function, the decimation and soft thresholding. Also present in the binary images are ridge marks not associated with ridges on the homomorphic filtered image shown in Figure 5, but these are generally short. Figure 9 shows an enhanced lake mark with Figures 10, 11 and 12 show the corresponding binary images at each scale.

6 Classification of the Defects

Classification of defects is implemented by examining the structure of the ridge marks from the binary images created by the process described in Section 5. Isolated short ridge marks are eliminated from the binary images, to reduce the number of marks not associated with the ridges of the defect. A binary image created from scale 4 of a sheet containing no defects (a 'good sheet') will show no ridge marks. Hence this binary image can be used to detect if a defect is present, while the two remaining binary images are used to classify the defects.

Chain codes are extracted from the remaining two binary images to provide a one-dimensional descriptor on which classification is based. Prior to this step a local linking algorithm is applied to the binary images to join any spurious breaks that occur on a ridge mark. Classification of the defect is carried out by examining statistical values obtained from the chain codes plus shape features. The characteristics exploited are as follows, bank marks contain a relatively large number of ridges that are similar in lengths and are long. The ridges of lake marks are of different orientations and lengths while hot cylinder marks are short and similar in shape. Preliminary results for classification of the surface defects are promising, showing potential for the system to be expanded to include more types of surface defects.

7 Conclusions

A algorithm is proposed for the classification of surface defects on extruded polymer sheets. This algorithm is based on ridge detection from the wavelet coefficients obtained when using a wavelet which emphasises the relationship between ridges and zero-crossings of the wavelet coefficients. This method can be extended to included further surface defects of similar nature.

References


Wei D., Odegard J.E., Guo H., Lang M., and Burrus C.S., "Simultaneous Noise Reduction and SAR Image Data Compression Using Best Wavelet Packet Basis", Department of electrical and computer Engineering, Rice University, Houston, TX 77251-1892, USA - Internet, 1996
Appendix 5

Equipment Configuration

PHILIPS IMAGING SYSTEM (SBIP2) ................................................................. 111
VIDMEM1, VIDMEM2 .................................................................................. 111
VIDEO .......................................................................................................... 111
DISPLAY and OVERLAY .............................................................................. 111
LUT .............................................................................................................. 112
IRIS and RCSTORE ..................................................................................... 112
MATLAB COMPUTING ENVIRONMENT ..................................................... 114
**Philips Imaging System (SBIP2)**

Figure A5.1 shows a simple block diagram of the Philips Imaging System incorporating the hardware components of the SBIP2. Central to the block diagram of Figure A5.1 is the Crossbar Switch which is used to connect the other functional blocks of the SBIP2.

Also supplied with the SBIP2 imaging system are Software Development Tools which provide 'C' based functions to control the connections of the Crossbar and to exploit the resulting configurations. These 'C' based functions were not used in the development of algorithms during the course of the research as the supplied 'C' functions concentrate on spatial domain techniques. Instead the Software Development Tools are used to capture a still image and store the captured image on disk using the file image format of ITEX PCplus of Imaging Technology. TABLE A5.1 indicate the structure of the such a file.

**VMEM1, VIDMEM2**

These provide two separate video memories which are organised to contain 8-bit pixel information for 1024 columns and 512 rows.

**VIDEO**

The VIDEO block samples a video signal to obtain 8-bit pixel value which can be stored directly in either of the video memories. The VIDEO block also contains a LUT (LookUp Table), which can pre-process the signal.

**DISPLAY and OVERLAY**

These blocks can receive their input from any of the other blocks via the Crossbar. DISPLAY has 8 input bits and OVERLAY has 4 input bits. Both DISPLAY and OVERLAY act as a LookUp Table which are to output a programmable value for any possible input value. The only difference is that a value on the OVERLAY input bits
different from 0 generates an output that overrules any actual DISPLAY output. Thus on top of an image an overlay may be visible.

**LUT**

The LUT (LookUp Table) block represents a general purpose lookup table with an input width of 10 bits and an output width of 8 bits. Inputs are received via the Crossbar Switch and the output is led to the Crossbar Switch.

**IRIS and RCSTORE**

The IRIS block is able to perform real time binary correlations on an image. The input is 8 bit wide and is received via the Crossbar Switch. The output has a width of 10 bits. The RCSTORE block can be used in conjunction with the IRIS block to show matchings found within an image for a specified template.
Figure A5.1: Functional Block Diagram of SBIP2
### Table A5.1: Image File Format

<table>
<thead>
<tr>
<th>BYTE#</th>
<th>CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 1</td>
<td>ID-filed containing &quot;IM&quot;</td>
</tr>
<tr>
<td>2 - 3</td>
<td>Comment field length (MAX. 200 including the NULL character)</td>
</tr>
<tr>
<td>4 - 5</td>
<td>Image-width in pixels or columns</td>
</tr>
<tr>
<td>6 - 7</td>
<td>Image-height in pixels or rows</td>
</tr>
<tr>
<td>8 - 9</td>
<td>X-co-ordinate or column of the upper left corner of the image</td>
</tr>
<tr>
<td>10 -11</td>
<td>Y-co-ordinate or row of the upper left corner of the image</td>
</tr>
<tr>
<td>12 -13</td>
<td>File type, 0 -NORMAL; 1 = COMPRESSED, NOT supported for SBIP2</td>
</tr>
<tr>
<td>14 - 63</td>
<td>Reserved</td>
</tr>
<tr>
<td>64 - n</td>
<td>Comment area with length defined by bytes 2 - 3.</td>
</tr>
<tr>
<td>n +1 - m</td>
<td>Data of the image. One byte per pixel in row order from the top of the image.</td>
</tr>
</tbody>
</table>

---

### MATLAB Computing Environment

MATLAB is a technical computing environment for high performance numeric computation and visualisation supplied by The Math Works Inc.. This environment provides an interactive system whose basic element is a matrix that does not require dimensioning. This allows solution of many numerical problems in a fraction of the time it would take to write a program in a language such as C.

An important feature of MATLAB is its easy extensibility. This allows applications to be developed within the MATLAB computing environment by creating an M-file or a MEX-file. An M-file contains a program developed in the MATLAB computing environment that can execute long sequences of commands. A MEX-file is a MATLAB executable file that allows the MATLAB computing environment to interface with low-level programs such as C and Fortran.
MATLAB also features a family of application-specific solutions that are referred to as ‘toolboxes’. These toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the environment in order to solve particular classes of problems. The toolboxes, supplied by The Maths Works Inc., used during the course of the research include the Signal Processing, Image Processing and Neural Network toolboxes as well as a Wavelet Toolbox supplied by the Universidad de Vigo.

An M-file (ph2mat.m) was developed by the author to convert files containing images using the file image format of ITEX PCplus of Imaging Technology to a matrix variable in the MATLAB computing environment. This M-file and others developed by the author during the course of the research are listed in Appendix 6 and supplied a supplementary disk.
Appendix 6

Matlab M-Files

This appendix provides a list of the m-files provided on the first of the supplementary disks. These m-files are used by the MATLAB computing environment, in addition to functions and m-files supplied with the MATLAB computing environment and its Toolboxes, to implement the algorithms described throughout this thesis. Because of this the m-files are contained in directories corresponding to the chapter in which the m-file is used. The list of m-files below is sub-divided by these directories.

The second disk contains the Wavelet Toolbox provided by Universidad de Vigo, whose m-files implemented the bulk of the algorithms described in chapters 6, 7 and 8.

```
chapter3\

opt_thre.m  : calculates the Optimal Threshold of intensity images or sub-images.
locthres.m  : implements the Dynamic Thresholding method.
il_canc.m   : implements the Illumination Cancellation procedure.
```

```
chapter4\

filt.m      : creates filter matrix for implementation with homomorphic filtering procedures
hf.m        : implements homomorphic filtering enhancement procedure using FFT
hfdct.m     : implements homomorphic filtering enhancement procedure using DCT
```
- *chapter5*
  - **nnvector.m**: creates input vectors for the Neural Network of the classification system from raw images of extruded polymer sheets.

- *chapter6*
  - **wtpk.m**: One-Dimensional Wavelet Packet Decomposition
  - **wtpk2d.m**: Two-Dimensional Wavelet Packet Decomposition
  - **iwtpk.m**: One-Dimensional Wavelet Packet Reconstruction
  - **iwtpk2.m**: Two-Dimensional Wavelet Packet Reconstruction
  - **showwt.m**: Displays one-dimensional wavelet coefficients on a normalised time frequency axis
  - **showwtpk.m**: Displays one-dimensional wavelet packet coefficients on a normalised time frequency axis
  - **com_num.dll**: Mex-file used to calculate the Data Compression Number for the one-dimensional wavelet packet decomposition.
  - **comnum2d.m**: used to calculate the Data Compression Number for the two-dimensional wavelet packet decomposition.

- *chapter7*
  - **blockdct.m**: Implements the ‘Block DCT’ Transform
  - **iblockdc.m**: Inverse ‘Block DCT’ Transform
  - **findth.m**: Calculates the a threshold value that will a given number of absolute values form an input matrix greater than the calculated threshold.

- *chapter8*
  - **mallat.m**: Creates ‘zero-crossing’ wavelet filters.
  - **softhre.m**: Implements Soft Thresholding procedure
  - **denoise.m**: Noise reduction for one-dimensional Wavelet Coefficients
  - **denoise2.m**: Noise reduction for two-dimensional Wavelet Coefficients
cbw.m : Creates binary image from denoised two-dimensional Wavelet Coefficients

link.m : Localised linking algorithm for binary images

chain.m : Extracts chain codes from binary images.

appendix5/

ph2mat.m : Extracts an image contained within a file using the file image format of ITEX Pcplus of Imaging Technology.