Pattern Recognition, Vol. 26, No. 10, pp. 1441 1460, 1993
Printed in Great Britain

0031-3203/93 $6.00 + .00
Pergamon Press Ltd

FINGERPRINT RECOGNITION IN LOW QUALITY IMAGES

LOUIS COETZEE and ELIZABETH C. BOTHA
Department of Electrical and Electronic Engineering, University of Pretoria, Pretoria 0002, South Africa

(*Received* 28 *February* 1992; *in revised form* 23 *March* 1993; *received for publication* 7 *April* 1993)

Abstract—Algorithms are identified which are best suited for an automatic fingerprint recognition system operating on low quality images. New preprocessing algorithms for noise removal and binarization are described. Three approaches to classification are investigated: a correlation classifier, and two feature-based classification schemes. The best results on a database of 80 fingerprints are obtained with spatial-frequency features. Three classifiers (neural net, linear classifier and nearest neighbour) using these features are successful in identifying an independent test set. Details of the results are shown. In conclusion suggestions are made concerning the most suitable algorithms in each of the processing steps.

Fingerprint recognition Image recognition Access control Low quality image recognition

1. INTRODUCTION

Fingerprint comparison is a fundamental method of identifying a person. The reason for this is well known. The human fingerprint is one of the most reliable ways to identify a person, as no two fingerprints from different persons are the same.

Through the years the police were known as the major user of fingerprints in their quest to identify criminals. Due to the large volume of fingerprints and recent advances in computer technology, there has been increasing interest in automatic classification of fingerprints. An example of an application which is ideally suited for automatic fingerprint recognition is access control.

Various approaches for preprocessing and fingerprint recognition have been investigated, e.g. preprocessing for fingerprint classification was implemented by Ch and Rao;[1] Moayer and Fu[2] developed a syntactic tree system approach to represent and classify fingerprint images and a method for ridge detection in a fingerprint was investigated by Verma *et al.*[3]

One major problem in the automatic recognition of fingerprints is the quality of the original print. If the quality is not of an acceptable standard, automatic fingerprint identification becomes extremely difficult. The reason for this is that normal methods of fingerprint recognition use the small unique features (known as minutiae) in the fingerprint pattern to identify the fingerprint.[4] However, it is extremely difficult to extract these minutiae from the fingerprint image if the quality of the print is not perfect. Problems also exist in extracting these minutiae from the fingers of elderly people as well as manual labourers. The problem with elderly people's prints is that the prominence of the ridges diminishes, with the result that the fingerprint pattern is not very clear. Manual workers have the problem that the skin on the hands is subject to severe punishment, with the result that false minutiae are created by cuts in the skin and in some cases the ridges are worn away.

We developed preprocessing algorithms and classification schemes to implement in an automatic fingerprint system using low quality images. We investigated a system for low quality images as we feel that a reliable recognition system must be applicable throughout the population spectrum. Due to the above-mentioned problems that exist with the fingerprint images of elderly people and manual workers and the fact that good quality prints are difficult to obtain, minutiae are not a suitable basis for our system. Other classification methods have to be employed. These include recognition using a directional image of the fingerprint, the Fourier transform of the fingerprint and a direct comparison technique known as correlation. Preprocessing algorithms ideally suited for low quality images were developed. These algorithms improve the quality of the image so that better classification performance can be obtained. The system has one class per person and is thus suitable for applications where few people have to be recognized.

In the next section (Section 2) we describe two data generation methods that can be used to provide the input to our recognition system. This is followed in Section 3 by preprocessing algorithms ideally suited for images obtained from the optical data generation method as described in Section 2. In Sections 4–6 we describe the various approaches that were investigated for implementation in our recognition system. In Section 7 the results obtained from experiments conducted using these approaches are presented. In conclusion suggestions are made concerning the algorithms which are to be used in each of the processing steps (Section 8).

2. DATA GENERATION

In an automatic fingerprint recognition system, the fingerprint image must be available for processing in

a digital format. In our system fingerprint images are digitized by means of a video camera and a framegrabber. The video camera's output is an analogue signal of the fingerprint image. This analogue signal is then quantized into 256 discrete levels by means of the framegrabber. The framegrabber stores this quantized image as a two-dimensional array of bytes which can be used for subsequent processing.

We investigated two methods of data generation. The first method is the well-known ink-and-paper method that is commonly used by the police. The finger is rolled in ink and then rolled on paper to transform the three-dimensional finger into a two-dimensional print that can be photographed by the video camera and quantized by the framegrabber.

This method is not very reliable because the ink can smear and blur the whole image or part of the image. Plastic distortion plays a role as well, because the finger being rolled on the paper is not pressed equally hard each time the print is taken. Perhaps the most important disadvantage is that this technique is time consuming. This makes it unsuitable for application in any automatic fingerprint classification system, and inconvenient for our experimental purposes.

The second method is a more efficient and reliable optical data generation system. The system is shown in Fig. 1. It consists of a prism and a uniform light beam that transforms the three-dimensional data into two-dimensional data which can be photographed and quantized. This system makes use of the total internal reflection obtained in a 90° prism. The uniform light beam is shone into the prism. Normally all this light is reflected out of the prism by the 45° surface of the prism. However, when a finger is pressed on this surface, the refraction index of that surface is changed at the places where the ridges of the finger touch the prism's surface. At the places where the refraction index is changed, no light is reflected. This means that the ridges of the print appear dark in the image, while the background is of a high intensity. This reflected image of the fingerprint is photographed by the video camera and quantized by the framegrabber.

The optical method of data generation is not perfect either because the contrast and focus of the image obtained are sometimes poor. Most images have a continuous slope in greyscale values superimposed on the finer detail, which means that binarization techniques based on a global threshold are unsuitable. However, the advantages outweigh the disadvantages. The method is clean and very fast and most of the problems can be overcome by good preprocessing techniques such as greyscale-to-binary conversion as described in the next section.

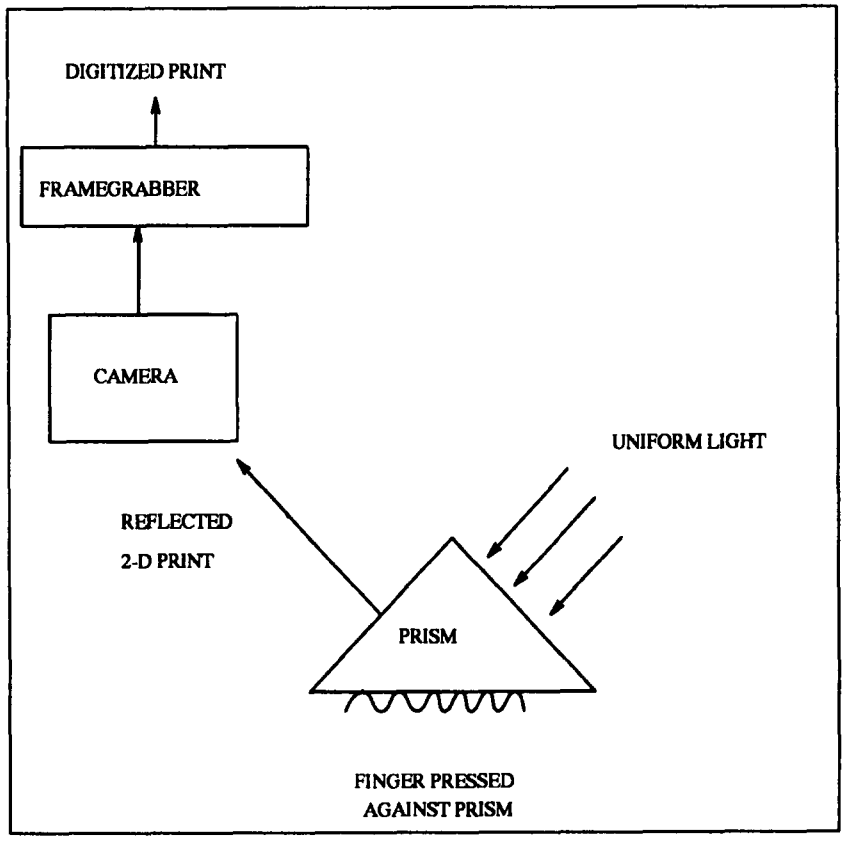

Fig. 1. Optical data generation.

3. PREPROCESSING

Preprocessing plays an integral part in any classification system. Good preprocessing techniques reduce the effect of poor quality data and this usually results in improved classification performance. The effect of such preprocessing is demonstrated in Section 7. In this section we describe an algorithm for the conversion of the greyscale image into a binary image, followed by an algorithm used for smoothing the binary image. We present results obtained from the application of these algorithms.

In our fingerprint recognition system we implemented the optical data generation method described in Section 2. This method is fast and is well suited to any automatic system. However, the data obtained from this system is not perfect. One of the major problems is a continuous slope in greyscale values superimposed on the finer detail. This means that binarization by means of a global threshold technique will be unsuitable for further processing. To combat this problem we developed a recursive binarization algorithm which is described next, which uses local thresholding to binarize the image.

3.1. *Binarization*

The edges of the greyscale fingerprint are extracted using the Marr–Hildreth algorithm.[5] This algorithm extracts the binary outlines (the edges) of the ridges from the original greyscale image. These edges in the edge image are one pixel wide. This "edge image" is then used in conjunction with the greyscale image to extract the binary image from the original greyscale image.

Two adaptive windows are used in each step of a recursion process to perform the binarization. These are the edge window, built in the edge image, and the greyscale window, built in the greyscale image.

The algorithm is based on the following six steps:

(1) Determine a suitable starting point. The current lowest greyscale value (assume it is X) in the greyscale image is used as the reference point around which the greyscale and edge windows are built.

(2) Build the greyscale and edge windows, using X (Step 1) or the location determined in Step 6 as the reference point. A $2d * 2d$ window is formed, centred on X. Figure 2 shows the greyscale window superimposed on part of the greyscale image.

(3) Determine transition regions on the boundary of a window. The calculation of the transition regions is explained later. Figure 3 shows a window with two transition regions.

(4) Evaluate the number of transition regions in both windows:

- if the number of transition regions is equal in both windows, choose the greyscale window;
- if the number is not equal, choose the window with the smallest number;
- if either number is zero, choose the other window and;
- if both zero, the recursion ends, i.e. choose neither.

(5) Binarize the greyscale image using the current windows as will be described subsequently.

(6) Determine the location of the windows for the next step in recursion. The endpoints of a transition region of the chosen window (Step 4) are used as the reference points (X and Y) for the construction of the windows. The window is formed by using a buffer of d pixels on each side of X and Y, as shown in Fig. 4. (A value of 2 was used for d in our experiments.) The algorithm then returns to Step 2 to continue the recursion process. Consecutive steps are shown in Fig. 5. Recursion continues for each transition region in the chosen window. In this way all ridges connected to each other are followed to their respective ends.

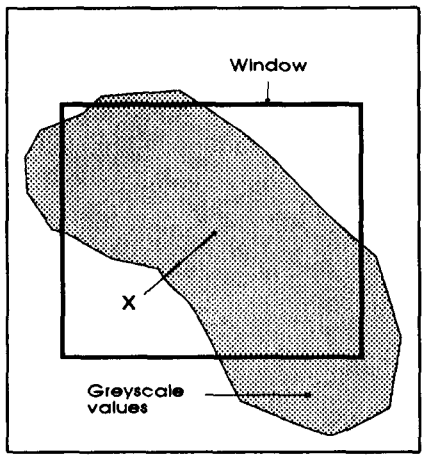

Fig. 2. Greyscale image with window: pixel X is current lowest greyscale value in image. Window centred on X.

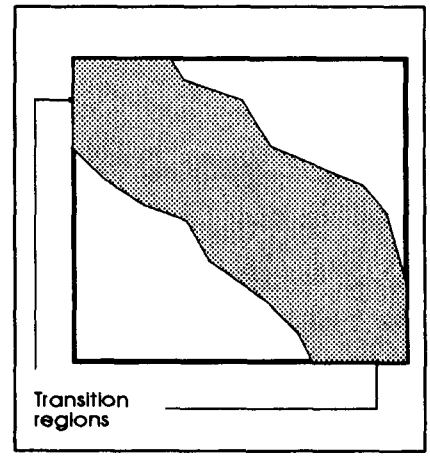

Fig. 3. Window with two transition regions superimposed on fingerprint ridge.

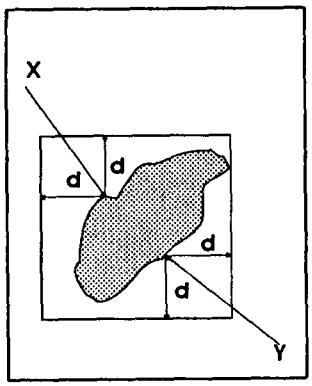

Fig. 4. Window with d pixel buffer.

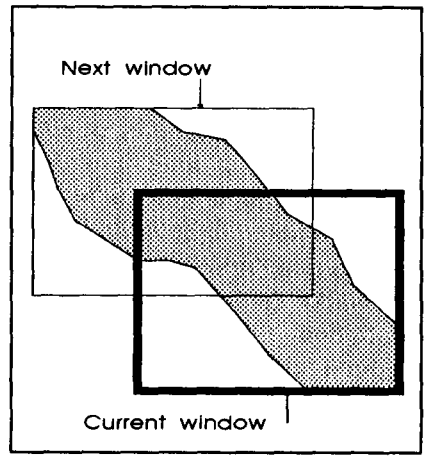

Fig. 5. Consecutive window positions during recursive binarization algorithm.

In order to determine the transition regions for a greyscale window, the boundary of the window is searched for the minimum and maximum greyscale values. If the difference between the two values is larger than a threshold (we used 0.6 of the maximum) the window is situated on a ridge. All greyscale values less than this threshold on the boundary form the transition regions of that window. Figure 3 shows the window with two transition regions on the boundary of the window.

The process of determining transition regions for an edge window is based on a blobcolouring routine[6] which colours between the edges of the ridges in the edge window. Figure 6(a) illustrates the edge window for the corresponding greyscale window (Fig. 6(b)). The algorithm uses the smallest greyscale value in the greyscale window which corresponds to an edge pixel in the edge window as an upper threshold in the blobcolouring routine (point X in Figs 6(a) and (b)). This value is assumed to be the largest greyscale value that could be part of a ridge. The routine colours all pixels with greyscale values lower than the threshold, as long as they lie within the edges of the edge window. This procedure prevents the blobcolouring routine from colouring outside an edge if the edge is not a solid line.

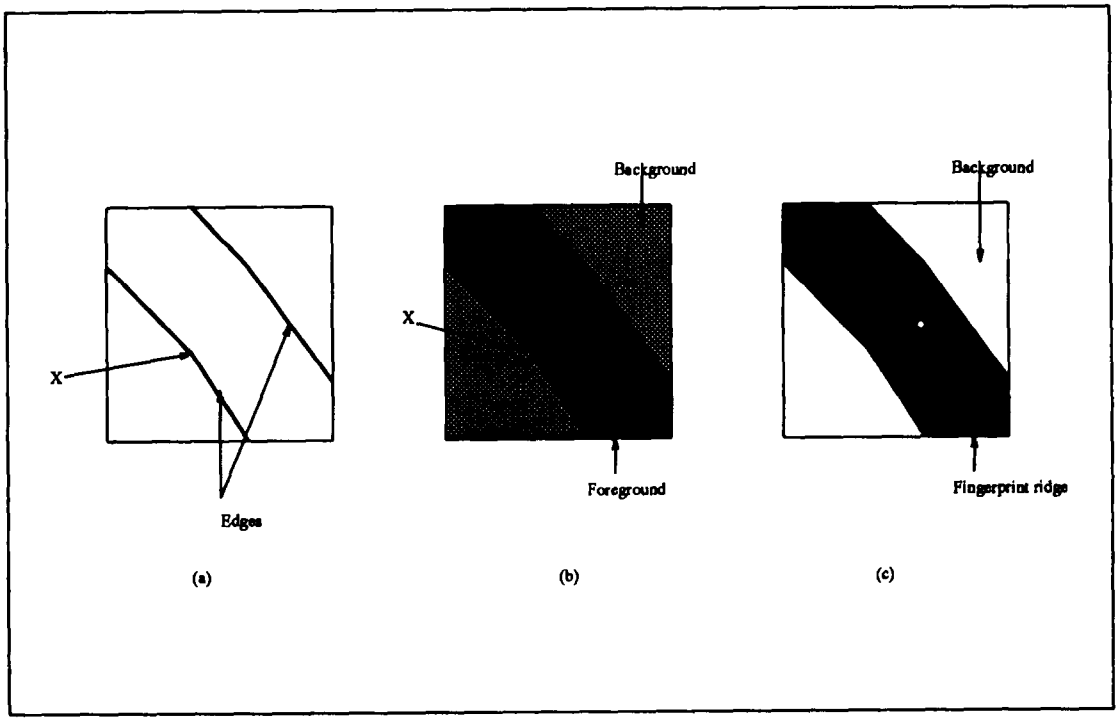

Fig. 6. Determining transition regions for edge window: (a) edge window; (b) corresponding greyscale window; (c) blobcoloured binary window.

The coloured ridge in the binary window is shown in Fig. 6(c). The number of transition regions is then determined by counting the areas in which the blob-colouring colour occurs on the boundary of the edge window.

The binary image for a set of greyscale and edge windows is formed by the logical "OR"ing of the binary image determined from the greyscale window and the binary image determined from the edge window. The binary image of the greyscale window contains all the elements which were lower than the threshold, while the binary elements of the edge window were all those which were blobcoloured.

The algorithm described above works well for images with a continuous slope in greyscale values. Examples of the application of this algorithm are presented in Figs 10–12. One problem with this algorithm is that the binary image is not smooth. It leaves holes and gaps in the ridge part of the resulting images, which can be viewed as noise. One of the eventual goals is to extract the skeleton of the binary image, but the existence of noise deforms the skeleton considerably and can severely handicap the extraction of meaningful features. An example of a deformed skeleton is shown in Fig. 13(a). To remove this noise, we developed an algorithm which is known as "smoothing", which we describe in the following subsection.

3.2. *Improvement of binary image ("smoothing")*

The algorithm for the improvement of the binary image is also based on an adaptive window concept. The first part of the algorithm locates possible isolated holes or gaps. In the next step the isolated holes or gaps in the edge part of the fingerprint are removed, using an adaptive window and a recursive blobcolouring routine.

The binary image is first scanned for isolated holes or gaps. An isolated hole is any island of background colour completely surrounded by foreground colour. Gaps are those regions of background colour, not completely surrounded by foreground and which are not part of merge or branch points. Examples of these are presented in Figs 7 and 8.

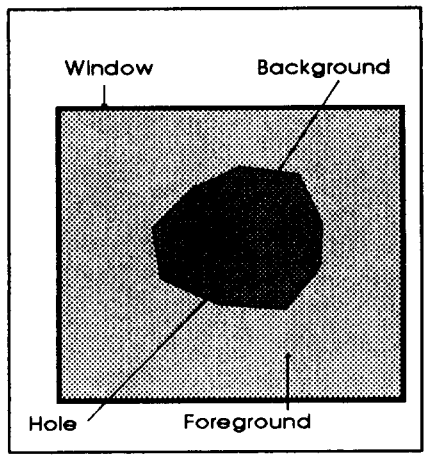

Fig. 7. Isolated hole surrounded by a window.

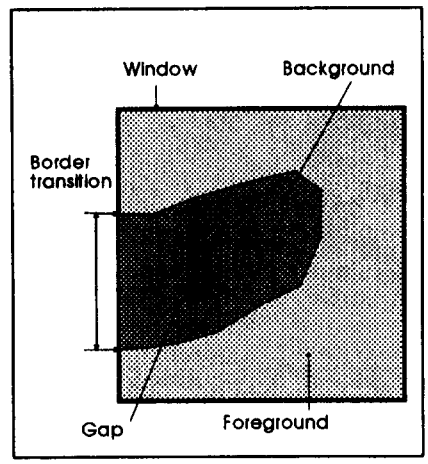

Fig. 8. Gap surrounded by a window.

To find a possible occurrence of a hole or gap, the image is searched for a transition from foreground to background colour (marked as X), followed directly by a transition from background to foreground colour (marked as Y). If the distance between X and Y is smaller than a chosen length, the algorithm continues removing holes, otherwise a new pair of transitions is sought.

To remove holes, a window is built as described earlier using X and Y as reference points. This window contains the possible isolated hole or gap as well as the surrounding data.

A four-connected blobcolouring routine (with starting point between X and Y) then seeks the background colour and colours its path in colour A. If the colour A is not present on the boundary of the window we know it is an isolated hole and it is filled with the foreground colour in the original binary image. If colour A is presented on the boundary the algorithm continues with removing gaps.

To remove a gap, the number and size of transition regions in the window are first determined (connected boundary elements of colour A form a transition region). A gap is bridged using a few heuristic rules:

- If the window only has one transition region and if the size is smaller than a certain threshold, it is a gap. This gap is then bridged by connecting the two endpoints of the transition region and filling it with the foreground colour.
- If the size of the transition region is less than the threshold or if more than one transition region is present a larger window is built around the original window (Fig. 9) and the blobcolouring process is repeated.

The transition regions are counted again:

- If the number of transition regions is two or less and if the distance (D in Fig. 9) between the regions is

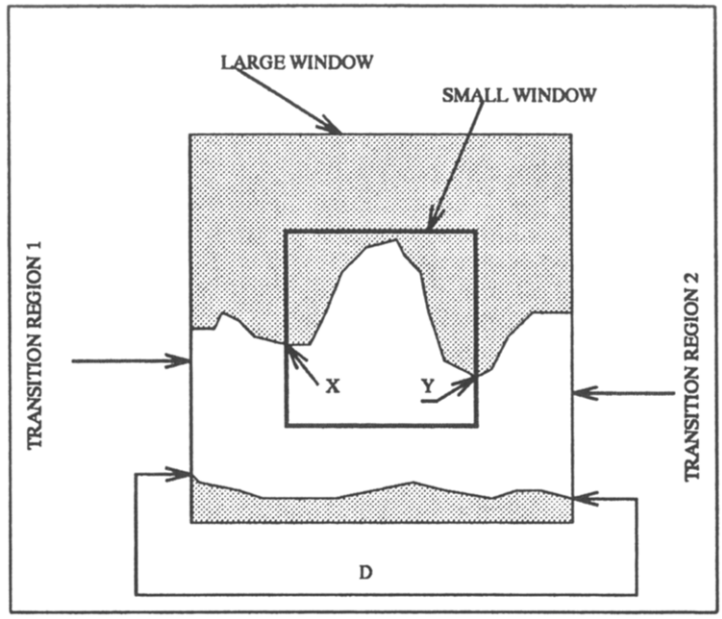

Fig. 9. Large and small window superimposed on gap.

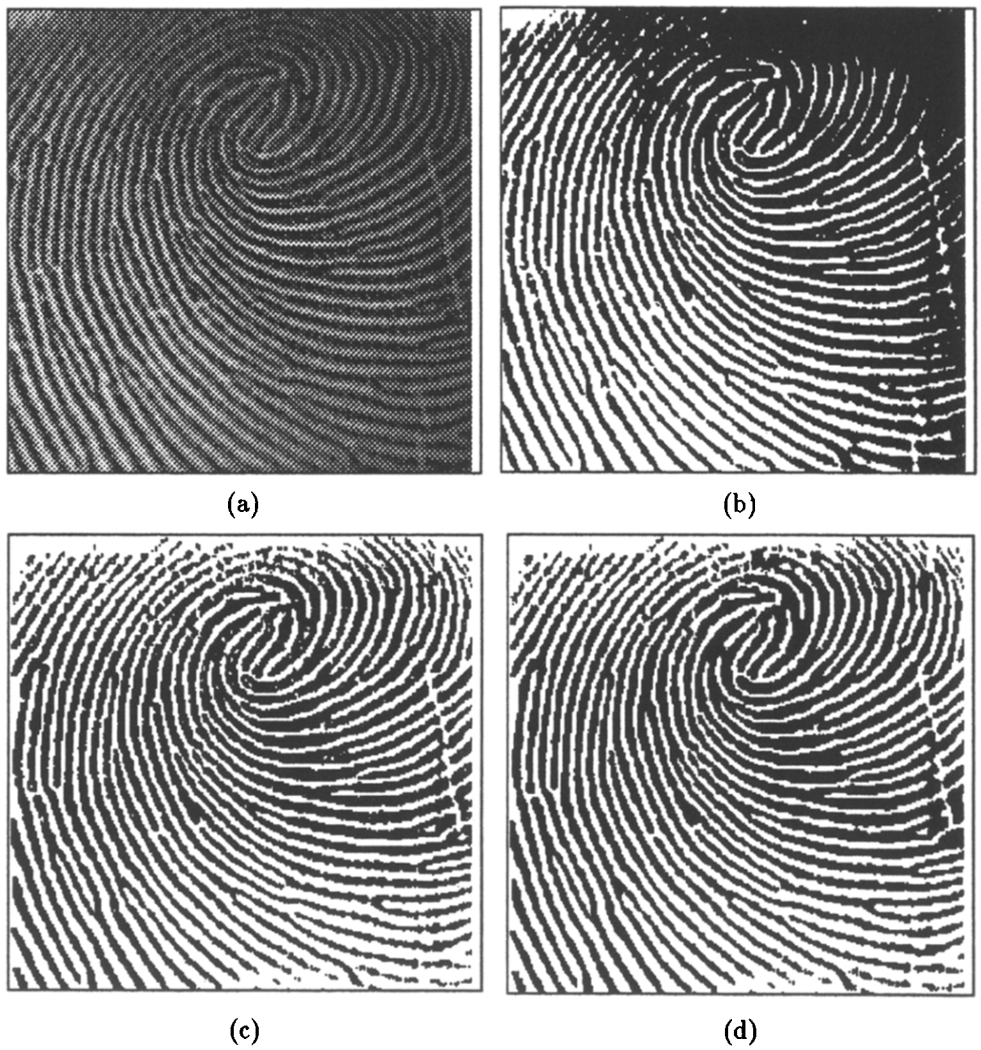

Fig. 10. First example of binarization algorithm: (a) greyscale image; (b) binary image through global threshold; (c) unsmoothed binary from algorithm in Section 3.1; (d) smoothed binary extracted from unsmoothed image in (c).

greater than another threshold, the gap (X and Y of the original window) is bridged as described previously.

- If three or more transition regions exist, the "gap" is considered to be noise and no gaps are filled.

The result of the application of this algorithm is a smoothed binary image which is more suitable for reliable feature extraction.

3.3. *Thinning the binary image*

For some of the subsequent feature extraction procedures, the skeleton of the fingerprint is needed. The skeleton is a thinned binary image in which the ridges are only one pixel wide. We implemented an adaptive line thinning by line following algorithm[7] to extract the skeleton from the binary image.

3.4. *Preprocessing results*

To evaluate the performance of our preprocessing algorithms, we tested them on various images of fingerprints. Typical results are shown in Figs 10–12. All these results were obtained from the greyscale images shown in Figs 10(a), 11(a) and 12(a), which were produced by means of the optical data generation method as described in Section 2.

Figures 10(b), 11(b) and 12(b) show the binary images of the three greyscale images as obtained from a global correlation threshold technique.[8] As can be seen, this method is quite inadequate for our application—various regions in the binary images are washed out completely. The reason for this is the continuous slope in greyscale values superimposed on the greyscale image due to the lighting during data generation. In compar-

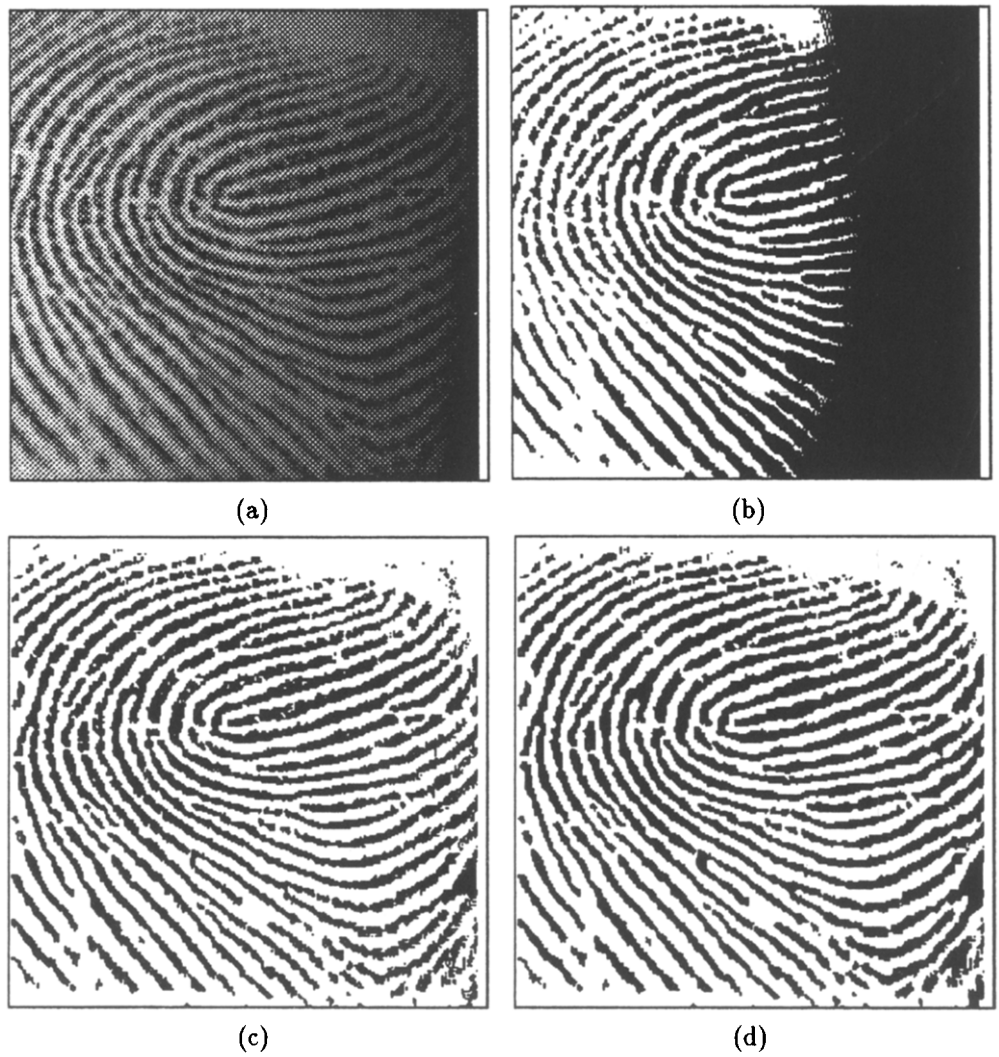

Fig. 11. Second example of binarization algorithm: (a) greyscale image; (b) binary image through global threshold; (c) unsmoothed binary from algorithm in Section 3.1; (d) smoothed binary extracted from unsmoothed image in (c).

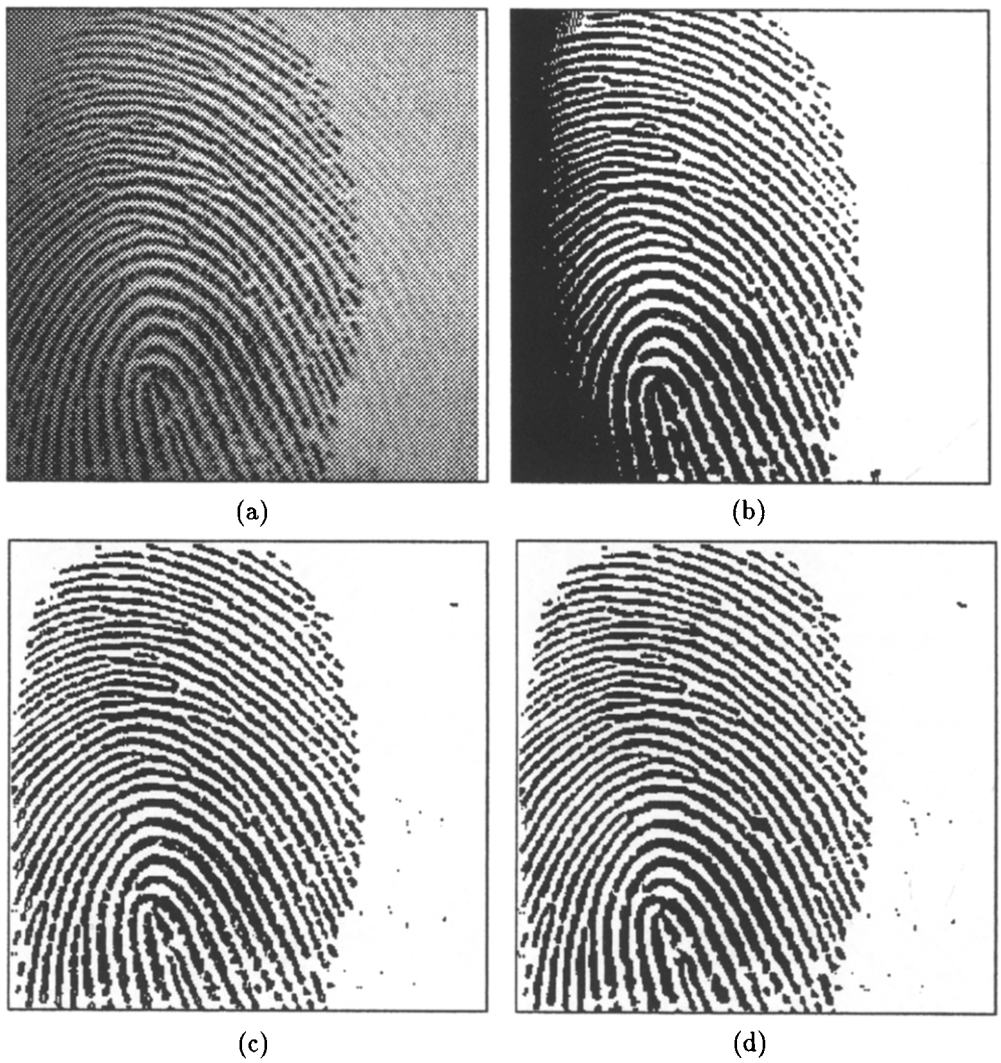

Fig. 12. Third example of binarization algorithm: (a) greyscale image; (b) binary image through global threshold; (c) unsmoothed binary from algorithm in Section 3.1; (d) smoothed binary extracted from unsmoothed image in (c).

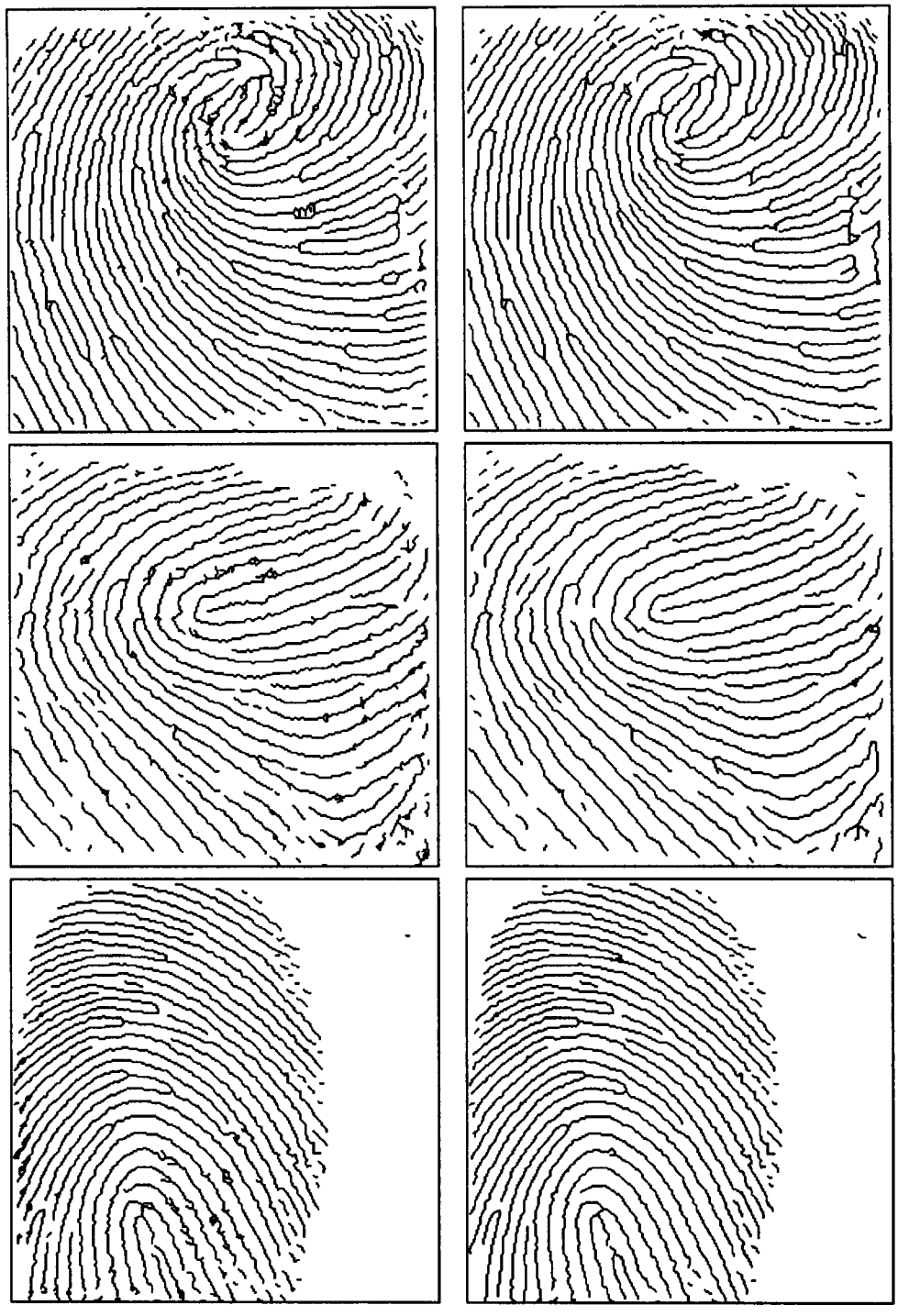

Fig. 13. Examples of results of thinning algorithm: (a) unsmoothed binary image's skeleton; (b) smoothed binary image's skeleton.

ison, the binary images as generated by our recursive binarization algorithm, shown in Figs 10(c), 11(c) and 12(c), have no completely spoilt regions and improved ridge continuity has resulted.

Figures 10(d), 11(d) and 12(d) show the binary images after the application of our smoothing algorithm.

Figure 13(b) shows the skeletons obtained from the smoothed binary images. (The smoothing algorithm described earlier was first applied to the binary images of Figs 10(c)–12(c) before the skeletons were extracted, and is shown in Figs 10(d)–12(d).) Figure 13(a) shows the skeletons obtained from the unsmoothed binary images of Figs 10(c), 11(c) and 12(c). Comparison of these images clearly demonstrates the improvement in the quality of the skeletons of the images which were smoothed. The unsmoothed images in Fig. 13(a) are disrupted by noise and no reliable features can be extracted from such images. On the other hand, reliable features can probably be extracted from the skeletons in Fig. 13(b).

The preprocessing algorithms described in this section are well-suited for implementation in an automatic fingerprint recognition system. However, the extent to which these algorithms are applied depends on the type of approach followed during classification. Some classification schemes require no preprocessing, while others require binarization or skeleton extraction. We investigated three approaches for classification, as described in the following section.

4. CLASSIFICATION BASED ON THE DIRECTIONAL IMAGE

The directional image is calculated from the greyscale image. We describe the manner in which a directional image is calculated, as well as feature extraction from the directional image. The features can be classified employing a nearest-neighbour, linear or neural-net classifier.

In order to calculate the directional image, the image is subdivided into small blocks. (We chose blocks of 16×16 pixels in our 256×256 images.) The directional image consists of the dominant directions of each of the small blocks. The dominant direction of a block is calculated in the following manner. The possible directions are quantized and each of these is called a subdirection. Refer to Fig. 14 where a quantization of four is illustrated. The four directions are labelled 0–3. Numerical values for each of the subdirections are calculated from the pixel intensities, after which the dominant direction is chosen from these. The value of a subdirection is equal to the sum of the absolute differences between the average value and the greyscale values along the subdirection. The subdirection with the smallest value (i.e. smallest variation) is the dominant ridge direction.

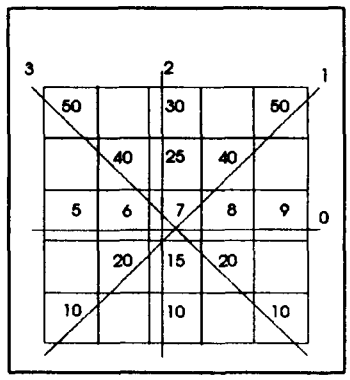

Fig. 14. Block quantized into 4 subdirections.

Refer to Fig. 14 for an example of this calculation. The small block shown is of dimension 5×5. Subdirection 0 consists of greyscale values 5, 6, 7, 8, 9 with an average of 7, subdirection 1 of 10, 20, 7, 40 and 50 (average of 25.4), subdirection 2 of 10, 15, 7, 25 and 30 (average of 17.4) and subdirection 3 of 10, 20, 7, 40 and 50 (average of 25.4). Consider subdirection 2: the average value is $(10 + 15 + 7 + 25 + 30)/5 = 17.4$. The absolute values of the differences are: $|10 - 17.4| = 7.4$, $|15 - 17.4| = 2.4$, $|7 - 17.4| = 10.4$, $|25 - 17.4| = 7.6$ and $|30 - 17.4| = 12.6$. The sum of these values is 40.4. In a similar fashion the sum of differences is computed for subdirection 0 (resulting in a value of 6), 1 (78.4) and 3 (also 78.4). For this example subdirection 0 is the dominant direction as it has the smallest sum of absolute differences value (6 compared to 78.4 and 40.4).

As the directional image is a numerical description of the fingerprint pattern, different feature vectors can be extracted from the numerical values. In the following subsections we describe the types of features we implemented in our experiments.

4.1. *Feature vectors*

4.1.1. *Histogram feature vectors.* Our first type of feature vector is based on the histogram of the dominant ridge directions. Each bin in the histogram is used as an element of the feature vector.

We demonstrate this with an example. In Figs 15 and 16 a small portion of a fingerprint pattern which has been divided into small blocks is shown. There is a total of 25 blocks in this portion of the print: 5 rows and 5 columns. Each block is quantized into four subdirections for the calculation of the dominant ridge direction. This means the dominant direction can be any direction from 0 through to 3. Figure 15 shows the dominant direction for each small block while Fig. 16 shows the numerical value assigned to that dominant direction in a small block. Because the original small blocks were quantized into four subdirections the histogram consists of four bins, one bin for each quantized subdirection. The feature vector for the directional image in Fig. 15 is the same as the values in the bins (the histogram values), i.e. (6, 4, 9, 6).

Variations on this type of feature vector can be obtained by excluding some of the small blocks from contributing to the histogram. We define two types of feature vectors: a large feature vector where all the small

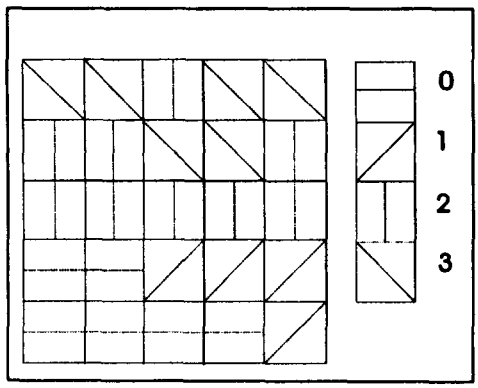

Fig. 15. Directional image.

3	3	2	3	3
2	2	3	3	2
2	2	2	2	2
0	0	1	1	1
0	0	0	0	1

Fig. 16. Numerical values of directional image.

blocks of the directional image contribute to the histogram and a small feature vector where the outer ring of small blocks does not contribute to the histogram. The idea with the small type of feature vector being that noise on the edge of the image is ignored in the calculation of the feature vector. The small feature vector for our example is constructed when rows 1 and 5 as well as columns 1 and 5 are ignored in the calculation of the histogram. The small feature vector in this case is (1, 2, 4, 2).

4.1.2. *Feature vectors for direct comparison.* In this type of feature vector the value of each element corresponds to the dominant ridge direction in each small block of the directional image. Thus the feature vector's dimension is equal to the number of blocks. For the directional image of Fig. 15 the corresponding feature vector is (3,3,2,3,3, 2,2,3,3,2, 2,2,2,2,2, 0,0,1,1,1, 0,0,0,0,1). This type of feature vector is defined as a large type of feature vector.

The small feature vector is again constructed by ignoring the outer ring of numerical values. That means all the elements that correspond to a block situated on the outer ring are assigned the value 0. The small feature vector for Fig. 15 is (0,0,0,0,0, 0,2,3,3,0, 0,2,2,2,0, 0,0,1,1,0, 0,0,0,0,0).

4.1.3. *Feature vector with variance bins.* When a direction for a small block is calculated, a certain amount of uncertainty about this direction exists. This uncertainty can be utilized in classification by calculating the variance from the dominant direction for each block. The variance is computed perpendicular to the dominant direction for each block and in a similar manner to the sum of differences value which is used to determine the dominant direction from the subdirections. The calculation of the variance is illustrated in Fig. 17. For each position in the dominant direction, the sum of differences is calculated along a line perpendicular to the dominant direction. In Fig. 17 the dominant direction is represented by the thick line and each of the seven thin lines represents the area where each calculation of the sum of differences is performed. These seven values are then summed to form the variance for that block.

The feature vector is constructed by using the variance as a means to increase the number of bins in the histogram of directions. For instance: each direction has three bins associated with it, a low, medium and high variance. When the dominant direction is calculated for a block, the variance determines which bin of

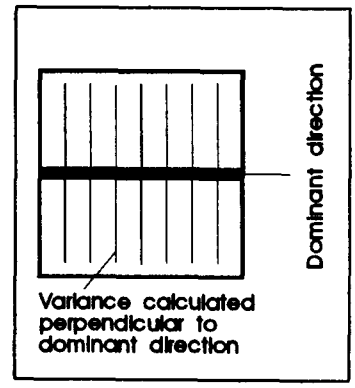

Fig. 17. Calculation of the variance for a block.

Table 1. Variance type of feature vector

Bin number	Variance
0	Low Medium High
1	Low Medium High
2	Low Medium High
3	Low Medium High

that direction (in the histogram of directions) is incremented. Table 1 shows this type of construction with the various variance bins for each quantized level.

Different feature vectors can be constructed by adjusting the number of bins for the variance and by adjusting the number of quantization levels for the dominant direction. Once again large and small feature vectors are defined. Large feature vectors correspond to the case where all the small blocks contribute to the histogram of variance bins, i.e. 256 small blocks in the 256×256 images. Small feature vectors are vectors constructed similarly, but the outer ring of small blocks is ignored.

In the next section we present another approach to the classification of fingerprint images. In this approach features are extracted from the spatial frequency domain of the fingerprint image.

5. CLASSIFICATION USING THE FOURIER TRANSFORM AND WEDGE-RING DETECTOR

In the previous approach features were extracted from the directional image calculated from the grey-scale image. In the following approach we transform the spatial greyscale image to the frequency domain and then extract "wedge-ring" features from the frequency domain image.[9] The frequency domain seems to be an appropriate feature space since although ridges in the spatial domain transform to a fairly constant frequency (in the spatial frequency domain), the distinguishing characteristics of a fingerprint such as the specific ridge directions and the minutiae manifest themselves as small deviations from the dominant spatial frequency of the ridges. These deviations or high-frequency components will be unique for each fingerprint. A "wedge-ring detector" is a way of partitioning the spatial frequency image. A wedge-ring, as shown in Fig. 18 is overlayed on the frequency image and the pixel values in each of the individual areas are added to form an element of the wedge-ring detector feature vector. These feature vectors can be used with the same classifiers as the features that were extracted from the directional image. The "wedge-ring detector" provides some invariance with regard to

- rotational variations (since the rings integrate along an angular variable) and
- scale variations (since the wedges integrate along a radial variable).

The wedge-ring features are translation invariant because the magnitude of the Fourier transform is used.

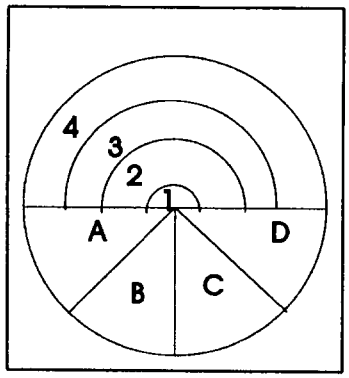

Fig. 18. Wedge-ring detector.

5.1. *Description of features*

The image is transformed from the spatial domain into the frequency domain by the application of a two-dimensional Fourier transform. The image is centred around the origin in the frequency domain by the addition of phase terms in the spatial domain. The mathematical operation for this is

$$e^{(j\pi t)} f(t) \leftrightarrow F(w - \pi).$$

Features are extracted from the frequency domain by using a wedge-ring detector[10] whose centre corresponds to the origin. An example of a wedge-ring detector with 4 rings (labelled 1, 2, 3 and 4) and 4 wedges (labelled A, B, C and D) was shown in Fig. 18. The summation of the elements (integration of the energy) in each ring or each wedge forms one element of the feature vector. Therefore, the feature vector constructed from the wedge-ring detector in Fig. 18 is

$$\left(\sum_{A} F(w_i), \sum_{B} F(w_i), \sum_{C} F(w_i), \sum_{D} F(w_i), \sum_{1} F(w_i), \sum_{2} F(w_i), \sum_{3} F(w_i), \sum_{4} F(w_i) \right)$$

with $\sum_{X} F(w_i)$ the summation of all the elements in the frequency domain for wedge or ring X. By varying the number of rings and wedges the dimension of the feature vector is adjusted.

In this, as well as the previous section, we described various methods of extracting features from the fingerprint image. These features can be used with a number of classifiers to obtain a positive classification. In the next section we describe another approach to the classification of fingerprint images. This approach uses a classifier not based on features, namely the correlation classifier. No features are extracted from the fingerprint image, but the prints are matched directly.

6. CLASSIFICATION USING THE CORRELATION CLASSIFIER

Classification using the correlation classifier does not require any type of special feature vectors. With this classifier the prints are compared directly. An unknown fingerprint image is classified in the following manner. The correlation between the unknown test print and one of the prints in a database is computed. This process is repeated for all the prints in the database. The test print is labelled as belonging to the class of

the training image that resulted in the largest correlation value. The correlation classifier is investigated as a "worst-case" classifier in our comparative study, since it does not provide any meaningful mechanism for scale or rotation invariance. To accommodate scale and rotational variations, the correlation classifier must be explicitly trained with scaled and rotated images of the object.

For good classification by means of a correlation classifier extensive preprocessing is needed to transform the greyscale image into a usable binary image. The preprocessing implemented consisted of the three stages described in Section 3: binarization of the greyscale image using the recursive greyscale edge follower, smoothing of the binary image using an adaptive window concept, and thinning of the smoothed image into its skeleton. The images obtained by the application of these preprocessing algorithms are noiseless binary skeletons of the original images. Examples of the various stages of preprocessing were shown in Section 3.

Correlation in the spatial domain is very time consuming, especially for large images. For this reason the correlation classifer was implemented in the frequency domain. Correlation in the spatial domain is equivalent to the inverse Fourier transform of the product of the Fourier transforms of the images being correlated. For two images f_1 and f_2, this can be mathematically expressed as follows:

$$f_1(x, y) \otimes f_2(x, y) \leftrightarrow F_1(u, v) \times F_2^*(u, v)$$

where $\otimes$ denotes the correlation in the spatial domain, * denotes the complex conjugate and F_1 and F_2 are the Fourier transforms of f_1 and f_2, respectively.

In the correlation classifier as implemented in the frequency domain, an unknown test image is classified in the following manner. The Fourier transform is computed for the unknown test image. This Fourier transform is multiplied with each of the known "training" images' Fourier transforms (which are calculated and stored beforehand). The inverse Fourier transform is computed for this product, resulting in a two-dimensional correlation function. The correlation image is then searched for its largest value. This value is normalized with the number of binary "ones" in the unknown test image. This normalized value is known as the correlation value. The process of multiplying, taking the inverse Fourier transform, searching and normalizing is repeated for all the "training" images. Finally the largest correlation value between the test image and all the training images is found and the test image is labelled as belonging to that class for which the correlation value was the largest. This procedure was first suggested for optical implementation.[11]

We normalize the largest value of the inverse Fourier transform with the number of binary ones in the test image so that the maximum correlation between two images cannot be more than 1.

In the next section we describe experiments conducted using the three different classification approaches described in the previous sections. The aim is to identify the best classification scheme for use in our recognition system.

7. RESULTS

The features described in Sections 4 and 5 were used in conjunction with three classifiers: the neural-net classifier, the linear classifier[12] and the nearest-neighbour classifier.[12]

The neural-net classifier used in our experiments is a three-layer Perceptron-type trained with the conjugate-gradient algorithm.[13]

A three-layer neural-net classifier is shown in Fig. 19. The first layer is the input layer whose outputs are the values of the elements in the feature vector to be classified. Thus the number of input neurons is equal to the dimension of the feature vector plus one. The second layer is the hidden layer while the third layer is known as the output layer. Each neuron in the second layer computes the sum of all its inputs. This sum is then used in conjunction with a transfer function to determine that neuron's output. The number of hidden neurons is problem specific and must be determined by means of repeated experiments with different numbers of hidden neurons. The output layer has one neuron for each class. Each output neuron performs the same calculation as the hidden layer neurons. We used a sigmoidal transfer function for both hidden and output neurons. If a specific neuron in the output layer has a high output value for an input feature vector, the feature vector is labelled as belonging to that class for which the output neuron is high.

Each connection between the input layer and hidden layer, as well as each connection between the hidden layer and the output layer has a certain numerical value associated with it. All these numerical values are contained in the weight matrix **W**. The training of the neural net consists of determining the best weight matrix **W** for that specific problem. The initial values of the weight matrix are determined randomly but must adhere to specific rules. These rules state that the values must not be too large or too small and that they must be distributed randomly.[14]

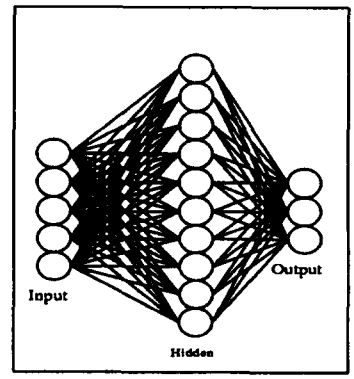

Fig. 19. Three-layer neural-network classifier.

The training procedure for the linear classifier was a gradient descent method using a minimum squared-error criterion function. For the nearest-neighbour classifier we used the euclidian distance E computed with the following equation:

$$E_{i,j} = \sum_{d=1}^{D} |\mathbf{T}_d^i - \mathbf{U}_d^j|$$

with D the dimension of the feature vectors, $\mathbf{T}^i$ the ith test vector and $\mathbf{U}^j$ the jth training vector as distance measure.

We used the same data set in all our experiments, generated optically as described in Section 2. The data set consisted of 60 training images and 20 test images from 20 classes. That is, each class consisted of three training images and one test image. The resolution of each image was 256 × 256 pixels.

The results of our experiments are shown in Figs 20–25 as well as Table 4. We show average values obtained from repeated experiments starting from 10 different initial sets of weights in the training of the linear and neural-net classifiers.

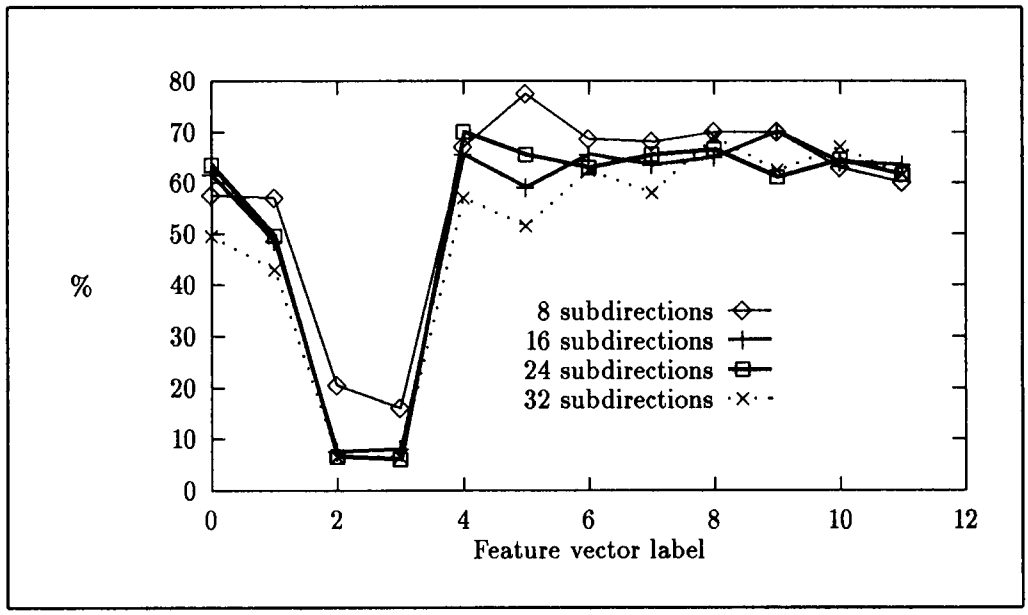

Fig. 20. Neural-net classifier performance—directional image.

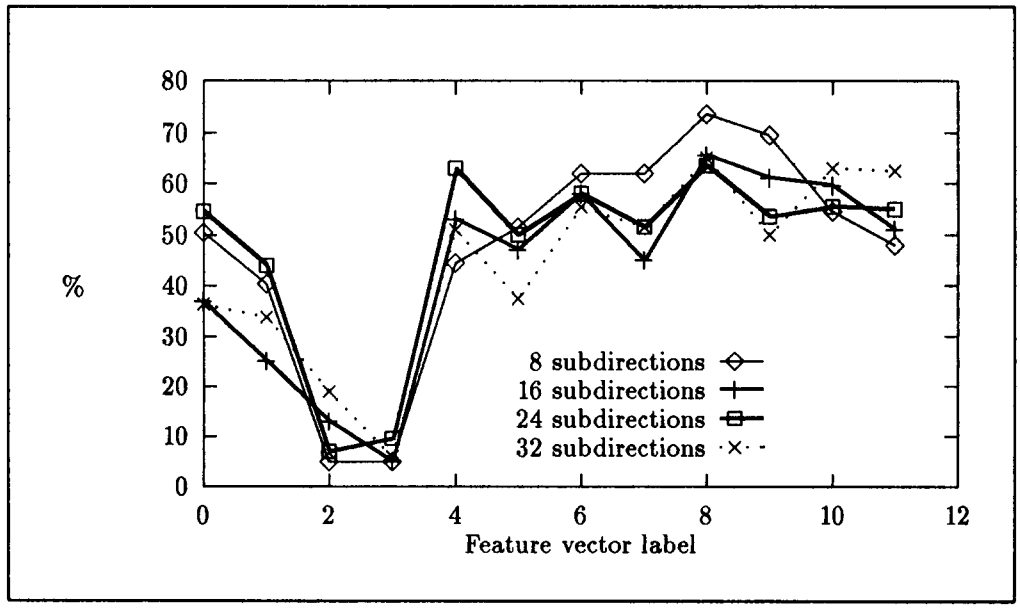

Fig. 21. Linear classifier performance—directional image.

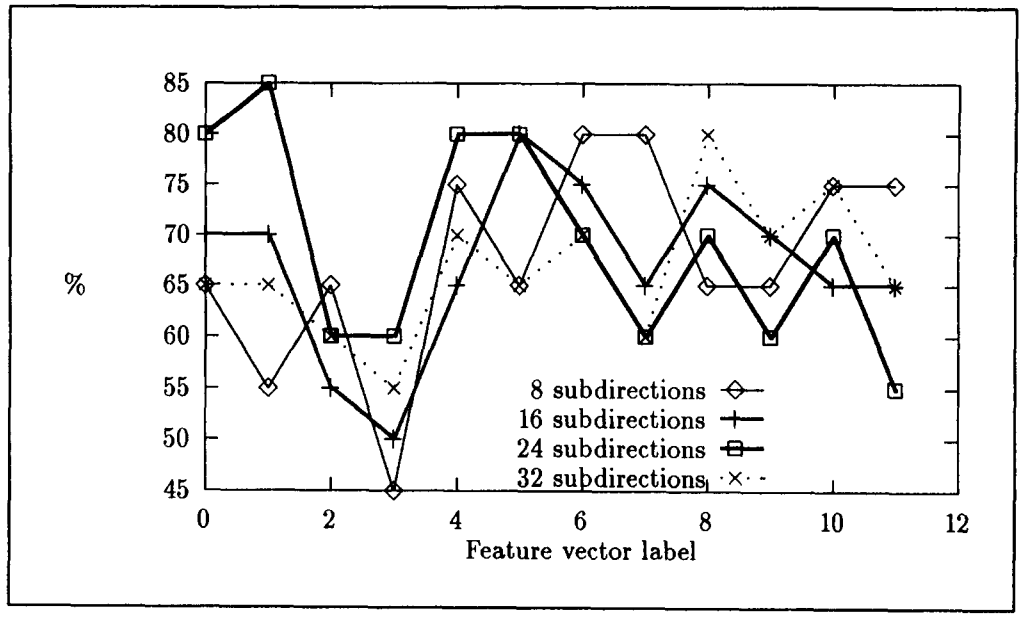

Fig. 22. Nearest-neighbour classifier performance—directional image.

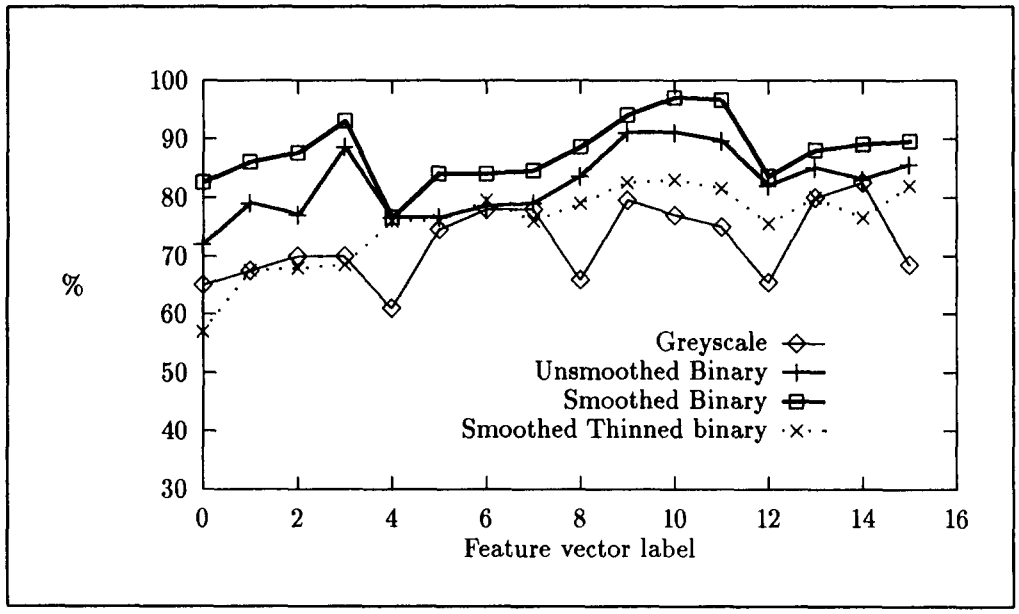

Fig. 23. Neural-net classifier performance—wedges and rings.

7.1. *Results of directional image*

The first group of experiments was conducted on features extracted from the directional image of the fingerprint. The experiments were conducted using three classifiers, i.e. the neural-net, linear and nearest-neighbour classifiers.

We used the features as described in Section 4 for these experiments. Each type of feature vector used in our directional experiments is labelled with a number to facilitate graphical representation. This association between the type and its label is shown in Table 2. The average performance of the different classifiers for different feature vectors based on the directional image can be seen in Figs 20–22. The performance of a specific type of feature vector can be determined by reading its label from the figure and cross referencing it with Table 2.

7.1.1. *Neural-net classifier performance.* Each experiment with a specific type of feature vector was repeated with 10 different randomly chosen initial sets of weight

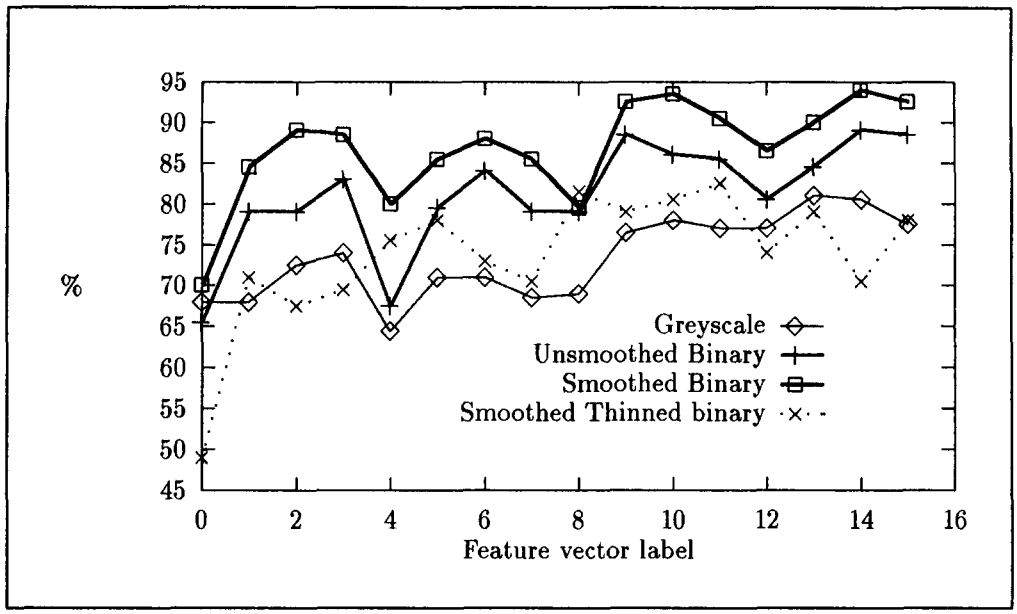

Fig. 24. Linear classifier performance—wedges and rings.

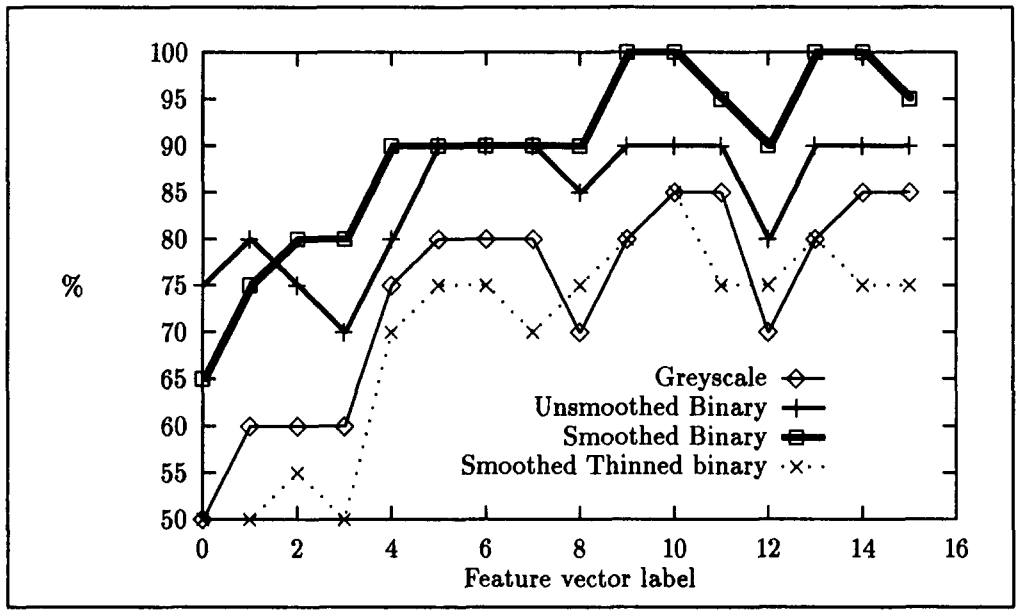

Fig. 25. Nearest-neighbour classifier performance—wedges and rings.

Table 2. Label associated with each type of feature vector—directional image

Label	Type of feature vector	Dimension of feature vector
0	Large histogram	Number of quantized levels
1	Small histogram	Number of quantized levels
2	Large direct	Number of blocks
3	Small direct	Number of blocks
4	4 variance bins, large	Number of quantized levels $\times 4$
5	4 variance bins, small	Number of quantized levels $\times 4$
6	8 variance bins, large	Number of quantized levels $\times 8$
7	8 variance bins, small	Number of quantized levels $\times 8$
8	12 variance bins, large	Number of quantized levels $\times 12$
9	12 variance bins, small	Number of quantized levels $\times 12$
10	16 variance bins, large	Number of quantized levels $\times 16$
11	16 variance bins, small	Number of quantized levels $\times 16$

values to obtain statistical results. The greyscale images were divided into small blocks of dimension 16 × 16 pixels, i.e. 256 small blocks in the 256 × 256 images. No preprocessing was done on any of the images prior to the extraction of the directional image.

Figure 20 shows the average test classifier performance for each of the quantized levels. The best average test performance was achieved with type 5 (four variance bins, small feature vector). The average test classification values all start near 60% before dropping very low for the direct comparison feature vectors. It then climbs to 70% where it stays close to 70% for all subsequent feature vector types. From Fig. 20 it is clear that the direct type of comparison feature vector is not suited for use with a neural-net classifier. It is also clear that the variance type of feature vector performs better than any of the other types of feature vectors. The variation of the number of quantized levels does not greatly affect the average classification performance, although the best performance was obtained from the lowest quantization level. The maximum performance obtained was 90% for a quantization level of 8 for 9 hidden neurons in the neural net.

7.1.2. *Linear classifier performance.* Figure 21 shows the average test classifier performance for the features of Table 2 using a linear classifier. Once again the direct comparison type of feature vector fails. It is also clear that the variance type of feature vector performs better than the histogram type of feature vector. The best average test classification performance was obtained for type 8 (12 variance bins, large) with a quantization level of 8. The average test classification varies more for the different number of quantization levels.

Once again the maximum test classification performance was 90% on the variance type of feature vector with 8 quantization levels.

7.1.3. *Nearest-neighbour classifier performance.* We repeated the experiments using the features as in Table 2, using the nearest-neighbour classifier. The results are shown in Fig. 22.

It can be seen in Fig. 22 that the best classification was 85% for 24 quantized levels using the small histogram feature vector (type 1). The worst classification was again on the direct-comparison type of feature vector. In general, classifier performance was centred around the 70% mark with variations to either side. The best classification performance was on features using the small type of construction but no definite trend can be seen.

7.2. *Results from Fourier transform and wedge-ring detector*

Experiments on this feature set were performed with a varying number of wedges and rings. Each of these feature vectors with a different number of wedges and rings is labelled. The relationships between the different feature vectors and their respective labels are given in Table 3.

Table 3. Labels associated with variations in wedges and rings

Label	Number of wedges	Number of rings	Dimension of feature vector
0	2	6	8
1	2	12	14
2	2	18	20
3	2	24	26
4	4	6	10
5	4	12	16
6	4	18	22
7	4	24	28
8	6	6	12
9	6	12	18
10	6	18	24
11	6	24	30
12	8	6	14
13	8	12	20
14	8	18	26
15	8	24	32

Experiments were performed on features extracted from greyscale images, unsmoothed binary images, smoothed binary images and thinned, smoothed binary images.

7.2.1. *Neural-net classifier performance.* The same experimental setup was used in these experiments as in the experiments using the directional image and the neural-net classifier. Each experiment was repeated with 10 different initial sets of weight values. Once again the number of hidden neurons was varied. We show the average classification performance in Fig. 23 for the greyscale images, unsmoothed binary images, smoothed binary images and the skeleton of the smoothed binary images. We see from Fig. 23 that the best classifier performance is obtained from the smoothed binary image, followed by the unsmoothed binary images, the smoothed binary images' skeletons and then the greyscale images. One would expect the best performance from the skeletons of the smoothed binary images, but apparently meaningful information is lost by the skeleton extraction. The effect of good preprocessing in the form of the smoothing algorithm described in this paper is evident in the better performance obtained from the smoothed binary images compared to those obtained from the unsmoothed binary images. It does appear that the optimum number of wedges and rings for this data set and the neural-net classifier is 6 wedges and either 12 or 18 rings.

The best test classification performance obtained on features extracted from the frequency domain was 95% for the greyscale as well as the unsmoothed binary images. The best test classification performance obtained from the smoothed binary images improved to 100%. Classification performance dropped to 95% for the skeletons of the smoothed binary images.

7.2.2. *Linear classifier performance.* The average classifier performance obtained on the features of Table 3 extracted from the Fourier transform of the greyscale, unsmoothed binary, smoothed binary and thinned binary images and by using the linear classifier is shown in Fig. 24. The best average classification performance is 93% for type 14 using features extracted from the smoothed binary images. Once again features extracted from smoothed binary images perform the best, followed by the unsmoothed binary images and thinned binary images. In general the worst performance was obtained from features extracted from the greyscale images.

The same best performance was obtained for classification using the linear classifier as when the neural-net classifier was used. For greyscale images and unsmoothed binary images 95% was obtained. The performance improved to 100% when features were extracted from the frequency domain of the smoothed binary images. Once again the best performance dropped to 95% for the skeletons of the smoothed binary images. We assume that this drop in classifier performance can be attributed to a decreased generalization when using a thinned image.

7.2.3. *Nearest-neighbour classifier performance.* In this experiment the nearest-neighbour classifier was used with the features in Table 3.

Figure 25 shows that the best classifier performance of 100% was obtained with features extracted from the smoothed binary image. The feature types for which this excellent classification performance was obtained are 9, 10, 13 and 14. The worst classification performance was on the smoothed and thinned binary images. The classification performance obtained from the smoothed binary images increases as the number of wedges and rings increases, but an upper limit exists where after performance drops if the dimension of the feature vector increases further.

In all our experiments the best performance on the test set was obtained from features extracted from the frequency domain of the smoothed binary images. All three classifiers (neural-net, linear and nearest-neighbour) were able to classify the test set correctly (100%). The general trend in classifier performance suggests that more than 6 rings perform better for any number of wedges, but that an upper limit exists for the dimension of the feature vector, as classifier performance drops for feature vector types with higher dimensions.

7.3. *Correlation classifier results*

We also performed experiments with the correlation classifier on binary images at different stages in the preprocessing as described in Section 3. The first experiment was performed on unsmoothed binary images. This was followed by an experiment on the smoothed binary images. The last two experiments were conducted on thinned and on smoothed and thinned binary images. Table 4 shows the labels assigned to each class for each experiment. The correlation classifier was able to correctly classify 75% of the classes (classes 8, 9, 12, 13 and 19 incorrectly labelled) of unsmoothed binary images. The classifier performance did not improve for the smoothed binary images (the same classes were incorrectly labelled), but the average correlation value

Table 4. Correlation classifier performance

	Unsmoothed binary	Smoothed binary	Skeleton (unsmoothed)	Skeleton (smoothed)
Correct class	Labelled as			
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	1	1	8	8
9	5	5	9	9
10	10	10	10	10
11	11	11	11	11
12	10	10	12	12
13	5	5	5	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	5	5	5	5
20	20	20	20	20
Performance	75%	75%	90%	95%

did improve from 0.51 to 0.54, which means that a marginally better certainty was obtained. The performance improved to 90% with the skeleton of the unsmoothed binary images (13 and 19 incorrectly labelled). A classification performance of 95% was obtained for the skeletons extracted from the smoothed binary images; the only class incorrectly labelled was class 19.

The results clearly show the improvement obtained in classifier performance by the application of the preprocessing algorithms. However, the computational burden of computing the Fourier transform for each test image and each training image, followed by an inverse Fourier transform calculation of the product of these two Fourier transforms makes the correlation classifier unsuitable for use in a fingerprint recognition system where speed is important.

7.4. *Comparison*

Overall, the results obtained from features extracted from the frequency domain of the smoothed binary fingerprint image were extremely encouraging. All three classifiers (neural-net, linear and nearest-neighbour) were able to classify the test set correctly. This means that the classes are well separated in this feature space. The average performance of these classifiers was also better than any of the other methods investigated.

Classification performance obtained from the correlation classifier was worse compared to the 100% test classification obtained from the features of the frequency space. The best classification performance using the correlation classifier was on the skeleton of smoothed binary images (95%). The performance of this classifier on the smoothed binary images and unsmoothed binary images was similar (75%), but not as good as its performance on the skeletons of images (smoothed binary skeleton 95% and unsmoothed binary skeleton 90%). The classification obtained from the thinned binary images was good, but the computational burden associated with the correlation classifier severely handicaps this approach compared to the feature extraction from the frequency domain approach.

The best performance obtained from features extracted from the directional image was that based on the variance-bins type of feature vector. The best classification performance was 90% with the neural-net classifier using a feature vector constructed from a directional image which was quantized into 8 subdirections. The same classification performance was obtained with these features using the linear classifier. The best classification performance obtained with the nearest-neighbour classifier was 85%, where a histogram-type of feature vector was used, constructed from a directional image quantized into 24 subdirections.

8. CONCLUSION

In this paper subsystems for a possible automatic fingerprint recognition system for low quality images were described. These subsystems include data generation, preprocessing and various classification schemes. The main purpose was to determine what results can be obtained with low-quality images, and to develop preprocessing algorithms to improve recognition.

Low-quality images result from two widely used data generation methods namely, the (cumbersome) paper-and-ink method as well as a fast and clean optical data generation method. We introduced a new binarization algorithm as well as a new smoothing algorithm which greatly enhances classification performance. These algorithms were developed specifically for use on fingerprint data generated optically. Other approaches to fingerprint recognition are called for when high-quality inputs are available or when a large number of people have to be recognized.

Three classification schemes were investigated with the purpose of identifying the scheme which is best suited for implementation in our automatic fingerprint recognition system which operates on low-quality images. The three approaches investigated were classification using a directional image with the neural-net, linear and nearest-neighbour classifiers, classification using features extracted from the frequency domain of the fingerprint image with the same three classifiers and finally classification using a correlation classifier. No preprocessing was done on the data for the experiments using the directional image. Experiments were performed on greyscale, unsmoothed binary, smoothed binary and skeletons extracted from the smoothed binary images in the frequency domain approach. In experiments using the correlation classifier the images were tested (classified) at various stages of preprocessing. The images used were unsmoothed binary, smoothed binary, and the skeletons extracted from the smoothed and unsmoothed binary images.

The best performance obtained from the directional image experiments was 90% using either the neural-net or linear classifiers where the type of feature used was a variance-based feature vector. Only 85% classification performance was obtained by using the nearest-neighbour classifier. The performance obtained from these experiments was not reliable enough to merit the use of this approach in our fingerprint recognition system. However, this approach might be useful in combination with other types of features such as minutiae to improve classifier performance.

The best classification performance obtained from experiments using the correlation classifier was 95%. This is better than the performance obtained from the experiments using the directional image, but the extensive preprocessing needed to extract the skeleton from the smoothed binary images, as well as the amount of time spent in classifying a test image renders this approach unsuitable for implementation in a fast classification system.

The best overall approach to classifying low-quality fingerprint images is that of feature extraction from the frequency domain of the fingerprint and by using the linear classifier. The only preprocessing needed is that of binarization and smoothing. All three classifiers were

able to correctly classify our test set, but the speed advantage offered by the linear classifier makes it more suitable for implementation.

Our final suggestion for a system for automatic fingerprint recognition using low quality images based on the results of our comparative investigations and improved preprocessing algorithms uses:

- an optical data generation method;
- the preprocessing consisting of recursive binarization and smoothing;
- a two-dimensional Fourier transformation;
- feature extraction using the wedge-ring detector and;
- classification performed by means of the linear classifier.

We believe that the quality of the input data plays a crucial role in the classification rates that can be obtained, and therefore a comparison with existing systems would not be meaningful. In order to improve the performance of our system, more robust features would have to be invented.

Acknowledgement—The authors would like to thank Etienne Barnard for many fruitful technical discussions.

REFERENCES

1. T. Ch and Malleswara Rao, Feature extraction for fingerprint classification, *Pattern Recognition* **8**, 181–192 (1976).
2. B. Moayer and K.-S. Fu, A tree system approach for fingerprint pattern recognition, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-8**(3), 376–387 (May 1986).
3. M. R. Verma, A. K. Majumdar and B. Chatterjee, Edge detection in fingerprints, *Pattern Recognition* **20**, 513–523 (1987).
4. A. K. Hrechak and J. A. McHugh, Automated fingerprint recognition using structural matching, *Pattern Recognition* **23**, 893–904 (1990).
5. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. R. Soc. London B* **207**, 187–217 (1980).
6. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey (1982).
7. O. Baruch, Line thinning by line following, *Pattern Recognition Lett.* **8**, 271–276 (1988).
8. A. D. Brink, Grey-level thresholding of images using a correlation criterion, *Pattern Recognition Lett.* **9**, 335–341 (1989).
9. L. Coetzee and E. C. Botha, Fingerprint recognition with a neural-net classifier, *Proc. First South African Workshop on Pattern Recognition*, Vol. 1, pp. 33–40, November (1990).
10. D. E. Glover, An optical Fourier/electronic neurocomputer automated inspection system, *Int. Conf. on Neural Networks*, San Diego, pp. I-569–I-576. IEEE (July 1988).
11. J. W. Goodman, *Introduction to Fourier Optics*. McGraw-Hill, New York (1967).
12. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, New York (1973).
13. E. Barnard and R. A. Cole, A neural-net training program based on conjugate-gradient optimization, Oregon Graduate Center Technical Report No. CSE 89-014, July (1989).
14. L. F. A. Wessels and E. Barnard, Avoiding false local minima by proper initialization of connections, *IEEE Trans. Neural Networks* **3**(6), 899–905 (1992).

About the Author—LOUIS COETZEE was born in South Africa. He received his B.Eng. (electronics) and M.Eng. (computers) degrees from the University of Pretoria, South Africa, in 1989 and 1991, respectively. He is currently pursuing his Ph.D. degree in electronic engineering at the University of Pretoria. His current research interests are in pattern recognition, image processing and parallel processing.

About the Author—ELIZABETH C. BOTHA was born in South Africa. She received her B.Eng. (electronics) and M.Eng. (electronics) from the University of Pretoria, South Africa, in 1983 and 1985, respectively. From 1985 to 1989 she was a research assistant in the Center for Excellence in Optical Data Processing at Carnegie Mellon University where she received her Ph.D. in electrical and computer engineering in 1989. Since 1989 she has been at the University of Pretoria where she is a Professor in the Department of Electrical and Electronic Engineering. Her research activities are in image processing, pattern recognition and neural networks.