# Fingerprint enhancement by directional Fourier filtering

B.G. Sherlock
D.M. Monro
K. Millard

*Indexing terms: Filters and filtering, Fingerprinting*

**Abstract:** A new method of enhancing fingerprint images is described, based upon nonstationary directional Fourier domain filtering. Fingerprints are first smoothed using a directional filter whose orientation is everywhere matched to the local ridge orientation. Thresholding then yields the enhanced image. Various simplifications lead to efficient implementation on general-purpose digital computers. Results of enhancement are presented for fingerprints of various pattern classifications. A comparison is made with the enhancement used within the automated fingerprint identification system (AFIS) developed by the UK Home Office. Use of the proposed enhancement method leads to significant improvements in the speed and accuracy of the AFIS.

## 1 Introduction

This paper introduces a new fingerprint enhancement algorithm for automated fingerprint indentification systems (AFIS), which are of increasing interest because of growing rates of serious crime worldwide. AFIS systems depend critically upon high quality enhancement, and the proposed algorithm is shown here to improve the speed and accuracy of a particular AFIS, that developed by the UK Home Office.

Several stages of processing take place when an AFIS is used to match an unknown fingerprint. The print is first enhanced to remove noise and any irrelevant information. The enhanced image is then encoded into a form suitable for comparison with the records held in the AFIS database. The encoded record consists of information describing the positions of the fingerprint's key attributes, called minutiae, and their spatial relationships. Matching is then performed by comparing the encoded record against those held in the database, thereby generating a shortlist of most likely candidate matches. Finally, in the verification stage, a fingerprint expert visually compares the unknown print with the shortlist of candidates, determining which (if any) is the correct match. The enhancement stage provides the only information available to the later stages. Consequently the performance of an entire AFIS depends critically upon the quality of enhancement achieved.

### 1.1 Related previous work

The importance of ridge orientation in fingerprint image processing is well established; indeed, many results describing directionality in general images have been applied to fingerprints [1–5]. Applications requiring knowledge of ridge orientation include enhancement/segmentation [6–10], detection of singular points [11], ridge detection during preprocessing [12], postprocessing to reduce numbers of false minutiae [13] and fingerprint pattern classification [14–17].

Fingerprint enhancement algorithms may involve little more than local average thresholding [16, 18, 19]. Methods which require higher noise immunity, such as those used operationally in AFIS systems [20, 21], usually precede the thresholding with directional smoothing. By estimating the local ridge orientations in the fingerprint image, an 'orientation image' may be constructed, giving the ridge orientation at each pixel position. Mehtre *et al.* [8, 9] described an enhancement algorithm based upon statistics derived from the orientation image. O'Gorman and Nickerson [6, 7] enhance fingerprints by applying oriented, matched spatial filter masks, where the mask orientation is determined by the orientation image. More recently, Mehtre [10] has presented an algorithm which uses a linear combination of an averaging and a differentiating directional spatial filter. Because each filter is limited to a 7 by 7 pixel neighbourhood of the point to be filtered, the algorithm is computationally economical, and it has the additional advantage that it can be applied iteratively to produce the desired degree of enhancement.

### 1.2 Overview of the enhancement algorithm

Spatial domain techniques [8–11] involve spatial convolution of the image with filter masks. For computational reasons, such masks must be small in spatial extent (typically of the order of 7 by 7 to 15 by 15 pixels). Our use of Fourier domain filtering and prefiltered images permits us to convolve the fingerprint image with filters of full image size, since the two-dimensional fast Fourier transform (FFT) algorithm can be used to calculate convolutions efficiently. In this way our directional filtering is performed using information from the entire image rather than from a small neighbourhood of the filtered point, and this leads to more effective noise reduction in the filtered image. Computationally efficient two-



Paper 9924K (E4), first received 24th May and in revised form 15th October 1993

B.G. Sherlock is with the Department of Electrical Engineering, Parks College of Saint Louis University, Cahokia, Il 62206, USA

D.M. Monro is with the School of Electronic and Electrical Engineering, University of Bath, Claverton Down, Bath BA2 7AY, United Kingdom

K. Millard is with Fingerprint Technology Associates, 101 Merlin Grove, Beckenham, Kent BR3 3HS, United Kingdom

dimensional FFTs are standard on all modern image-processing systems.

The enhancement consists of a filtering stage followed by a thresholding stage. The filtering stage produces a directionally smoothed version of the image from which most of the unwanted information ('noise') has been removed, but which still contains the desired information (i.e. the ridge structure and minutiae). The thresholding stage produces the binary, enhanced image.

Fingerprints exhibit everywhere a well defined local ridge orientation and ridge spacing. The enhancement algorithm takes advantage of this regularity of spatial structure by filtering the image with a position-dependent directional Fourier domain filter whose passband is everywhere matched to the local ridge orientation and spacing.

Fig. 1 illustrates our enhancement process. Although the directional filter is position-dependent, it is implemented using several position-independent Fourier filters. Choosing a representative set of quantified directions we define a set of directional filters which, when applied to the original image, yield a set of directionally filtered images. These we shall call the 'prefiltered images' (see Fig. 1).

The filtered image is then built up by selecting, for each pixel position, the pixel value from the prefiltered image whose direction of filtering corresponds most closely to the actual ridge orientation at that position (see Fig. 1). In order to perform this selection operation, knowledge of the actual ridge orientation is required, and this is obtained by estimation from the original image.

Finally, the thresholding stage binarises the directionally filtered image using a local average as the threshold surface.

## 2 Fourier domain filtering of fingerprint images

### 2.1 Local ridge parameters

Fingerprints consist of a pattern of locally parallel ridges, interrupted by minutiae, the fundamental types of which are ridge endings and bifurcations. The ridge structure defines everywhere a direction of ridge 'flow', called the local ridge orientation (LRO). Two types of singular points, cores and deltas, form isolated singularities of the otherwise continuous LRO function. It is the position and relationship of the minutiae which make a fingerprint unique, not the singular points or general classification types [27]. In addition, a local spatial frequency is determined everywhere by the local ridge spacing (LRS). The LRS is constrained within a range of values (typically 2 : 1 within a single print, or 4.5 : 1 over all prints).

In fingerprint image processing, one has the advantage of a well defined local frequency and orientation of information. In forming the Fourier transform of the image, the immediate neighbourhood of each point contributes (to first approximation) a single frequency component. The LRS determines its distance from the Fourier origin, and the LRO determines its angular position. (Strictly, components occur in Hermitian-symmetric pairs; we regard each pair as a single component.)

Our approach is to centre the passband of a filter over this component. We define in this way an anisotropic nonstationary bandpass filter which depends parametrically on LRS and LRO. This filter is applied to each point of the image, with the LRO and LRS as position-dependent parameters which determine the location of the passband. Values of LRS and LRO at each point are required; these are estimated from the image data. This filter passes the ridge information while eliminating most of the noise, since the noise is not localised in the same way as the ridge information. It is of course important that real minutiae are not discarded as noise. We show in our evaluation that this does not occur.

### 2.2 Directional bandpass filters

Using polar coordinates $(\rho, \phi)$, express the filter as a separable function

$$H(\rho, \phi) = H_{radial}(\rho)H_{angle}(\phi)$$

in order to allow independent manipulation of its directional and radial frequency responses. $H_{radial}(\rho)$ depends upon LRS, and $H_{angle}(\phi)$ upon LRO.

Any good classical one-dimensional bandpass filter would be adequate for $H_{radial}(\rho)$; the Butterworth filter was chosen because its implementation is simpler than such alternatives as the Chebyshev or elliptic filter, especially if it is desired to vary the filter order $n$. The expression for this filter is

$$H_{radial}(\rho) = \sqrt{\left[\frac{(\rho\,\rho_{BW})^{2n}}{(\rho\,\rho_{BW})^{2n} + (\rho^2 - \rho_0^2)^{2n}}\right]}$$

where $\rho_{BW}$ and $\rho_0$ are the desired bandwidth and centre frequency. A value of $n = 2$ worked well and was used throughout.

In designing $H_{angle}(\phi)$, one cannot be guided by analogy to one-dimensional filters because there is no meaningful one-dimensional concept of orientation. Knutsson *et al.* [5] used the following function in their

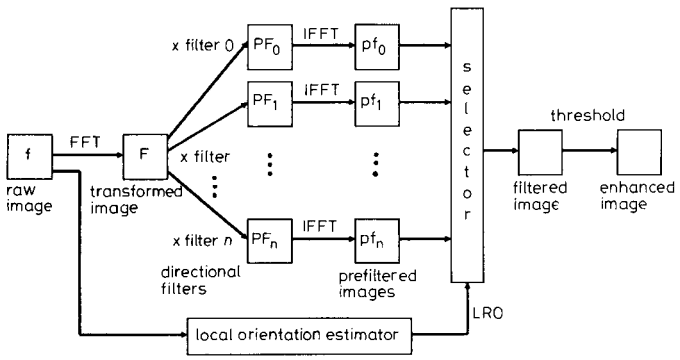



**Fig. 1** *Block diagram of the fingerprint enhancement algorithm*

development of anisotropic 'kernels' for their GOP image processor [22]:

$$H_{angle}(\phi) = \begin{cases} \cos^2 \frac{\pi}{2} \frac{(\phi - \phi_c)}{\phi_{BW}} & \text{if } |\phi| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases}$$

where $\phi_{BW}$ is the 'angular bandwidth' of the filter, i.e. the range of angles for which $|H_{angle}(\phi)| \geq 0.5$, and $\phi_c$ is its 'orientation', i.e. the angle at which $|H_{angle}|$ is maximum.

If $\phi_{BW} = \pi/n$ for some integer $n$, then we can define $n$ directional filters with equally spaced orientations $\phi_c = i\pi/n$, $i = 0, \ldots, n-1$. These filters sum to unity everywhere; therefore they separate an image into $n$ 'directional components' which sum to the original image.

## 3 Implementation of the filter

### 3.1 Discretisation of the filter

The previous section was presented in terms of infinite-extent analogue images and filters. To implement our filter on a digital computer, the images must be spatially sampled, and the continuous Fourier transform replaced by the discrete Fourier transform (DFT). All images were sampled at a resolution of 512 by 512 pixels. Edge effects in the DFT were reduced using a separable split-cosine window with a 10% taper.

One method of discretising an analogue filter is the bilinear transform with prewarping of the centre and cut-off frequencies [23]. If these frequencies are well below the sampling frequency, the bilinear transform/prewarping design is closely approximated simply by sampling the analog filter. This approach was used because the centre frequency of 50 pixel units and bandwidth of 80 units were well below the sampling frequency of 256 units.

### 3.2 Obtaining a set of prefiltered images

The position dependence of our filter implies a different filtering action for each pixel position. Direct implementation as an explicit function of position involves considerable computational effort. Noting that the direct dependence of the filter is upon LRO and LRS rather than position, we can make an approximation which reduces the computation substantially. Choosing a representative set of values for LRO and LRS determines some number $q$ of different stationary filters. Applying these to the image yields $q$ 'prefiltered images' (see Fig. 1). The filtered fingerprint is then built up by scanning the input fingerprint, selecting at each point the pixel value from whichever prefiltered image corresponds to LRS and LRO values closest to the true values.

Care must be taken when choosing discrete values of LRS and LRO for generating prefiltered images. Obviously it is desirable to keep the number of prefiltered images small, but this must be done without unacceptable degradation of the final filtered image. Two approaches helped to achieve this.

1. Eliminating of the LRS parameter: as already mentioned, the LRS parameter is constrained in range. By increasing the bandwidth of the filter so that it tolerates variations of LRS within its expected range, LRS may be assumed constant. The wider bandwidth results in lower attenuation of noise, but many fewer prefiltered images are required. This approach also eliminates the need to estimate LRS.

2. Coarser discretisation of LRO: The number of discrete values of LRO may be reduced if intermediate values are dealt with by interpolating between pixel values from prefiltered images rather than by selecting the pixel value from one of them. Our implementation used linear interpolation.

With the LRS parameter eliminated, the number of discrete orientations equals the number of prefiltered images. An angular bandwidth of $\pi/8$ was chosen, allowing LRO to be estimated to the nearest $\pi/16$. This implies a minimum of eight orientations. For the sake of caution, 16 equally spaced orientations were used. This corresponds to 16 prefiltered images organised as two sets of eight directional components. Enhancements produced using eight orientations were found to be almost indistinguishable from those presented here.

### 3.3 Regions of high ridge curvature

High ridge curvature occurs where the LRO changes rapidly, i.e. near cores and deltas. Away from these singular points, ridge curvature tends asymptotically to zero. In regions of higher curvature, a wider range of orientations is present. The angular bandwidth of the directional filter must therefore increase near singular points.

Away from singular points, the angular bandwidth is $\pi/8$ as discussed earlier. The angular bandwidth must equal $\pi$ at singular points since all orientations are present. Fig. 2 shows an empirical piecewise linear

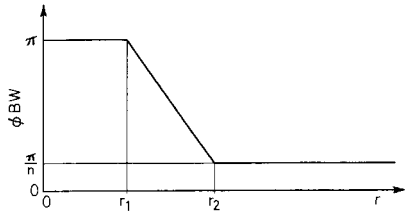



**Fig. 2** *Filter angular bandwidths near singular points. The graph shows the angular bandwidth as a function of the distance r to the nearest core or delta point*

relationship giving angular bandwidth as a function of distance from the nearest singular point. Values of $r_1 = 0$ and $r_2 = 20$ pixels are used throughout.

Where necessary, the effect of a filter of wider bandwidth than $\pi/8$ is simulated by an appropriately weighted combination of pixel values from two or more prefiltered images.

## 4 Estimation of the LRO parameter

The value of LRO at each pixel (i.e. the orientation image) is required as parametric input to the filter. Since determining LRO reliably can be computationally demanding, it may not be feasible to estimate LRO directly for every pixel. Our approach is to determine LRO at a square grid spaced (say) 16 pixels apart, and obtain intermediate values by interpolation. An alternative, equally acceptable approach is to use a faster but perhaps less reliable algorithm to estimate orientation at every pixel position and to smooth the resultant orientation image [8, 9]. Either approach could of course be used in conjunction with our filter.

Other researchers have worked with LRO values determined as one of four [16] or eight [10, 18] possible orientations. Our algorithm, which determines LRO as one of 16 orientations $\theta_i = i\pi/16$, $i = 0$–$15$, i.e. to a precision of $\pm\pi/32$, follows.

### 4.1 Algorithm for estimating LRO at a point

A window of size 32 by 32 pixels is centred at the point where the LRO is to be found. This window is rotated to 16 different orientations, $\theta_i = i\pi/16$, for $i = 0$–15. At each orientation a projection along the $y$-axis of the window is formed:

$$p_i(x) = \frac{1}{32} \sum_{y=0}^{31} W_i(x, y) \quad x = 0\text{–}31$$

where $W_i(x, y)$ is the data inside the window at angle $\theta_i$.

When the window is aligned with its $x$-axis perpendicular to the ridges, one expects maximum variation of the projection, since ridges are crossed as $x$ varies. Alignment of the $x$-axis along the ridges should lead to minimum variation. This is illustrated in Fig. 3.

A second-order Butterworth bandpass filter removes noise from the projections. The total variation $V_i$ of each filtered projection $fp_i(x)$ is evaluated as

$$V_i = \sum_{x=0}^{31} |fp_i(x+1) - fp_i(x)|$$

The LRO estimate is given by $i_{max}\pi/16$, where $V_{i_{max}}$ is the maximum of the 16 variations. This algorithm produces the correct value except in the noisiest regions. The typical failure rate for inked fingerprints is approximately 0.5%. We have developed a simple model describing the behaviour of LRO, and incorrect estimates can be dealt with by reference to this model [25, 26].

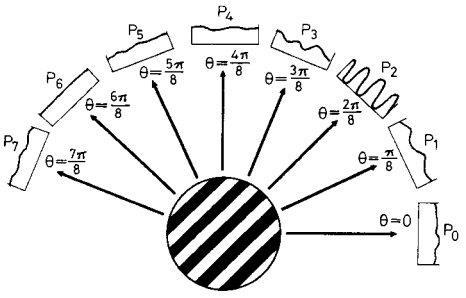



**Fig. 3** *Projections of a window of fingerprint image data. The projection which exhibits the greatest variation corresponds to the orientation of the ridges within the window (here $\theta = 2\pi/8$). For clarity, eight rather than 16 projections are shown*

## 5 Fingerprint enhancement by directional filtering

Previous sections developed a position-dependent anisotropic filter and means to estimate the LRO parameter which it requires. This section brings these results together to present an algorithm which produces high quality enhancements of fingerprint images. The enhancement algorithm consists of directional filtering followed by thresholding. The description which follows may be understood by reference to Fig. 1.

### 5.1 Directional filtering

The input to this first stage consists of a 512 by 512 pixel raw fingerprint image and a 30 by 30 array of LRO values. The filtering algorithm follows.

1. Obtain 16 prefiltered images PF[i] from the raw fingerprint PRINT, as described by the following pseudocode:

```
WINDOW(PRINT)                  { Window the image, to reduce edge effects. }
PRINTFOURIER := FFT(PRINT)     { Take the 2-D DFT of the windowed image. }
for i := 0 to 15 do
   TEMP := PRINTFOURIER*F[i]   { Multiply this by the ith filter function, and }
   PF[i] := IFFT(TEMP)         { inverse transform this to get the ith }
endfor                         { prefiltered image PF[i] }
```

2. Obtain the LRO at each point by interpolation between the 30 by 30 sampled values.

3. Form the output image OUT($x$, $y$) by appropriate combination of pixel values from the prefiltered images, as follows:

```
for x := 0 to 511 do
   for y := 0 to 511 do
```

Determine the distance $r$ from ($x$, $y$) to the nearest singular point. Using the function in Fig. 2, determine the angular bandwidth that is required, and hence the number $j$ of prefiltered images that must be combined.

Form the output pixel OUT($x$, $y$) as a weighted average of the ($x$, $y$) pixels of the $j$ prefiltered images nearest in orientation to the LRO (as determined in step 2 above).

```
   endfor
endfor
```

The result of applying this algorithm is a filtered image which has been smoothed in the direction of the ridges. Additionally, only wavelengths within the permissible range of ridge spacings have been passed. Fig. 4*a* and *b* show a fingerprint image and the filtered image produced by the above algorithm. Some of the 16 prefiltered images are shown in Fig. 5.

### 5.2 Thresholding

The output of the filtering stage is an image which renders the fingerprint ridge structure in the form of smooth greyscale 'ripples' with very little residual noise. The thresholding stage converts this into a binary image where pixels located on ridges are black, and pixels not on ridges are white. Local average thresholding within 32 by 32 pixel neighbourhoods yields the threshold surface

$$g_T(x, y) = \frac{1}{32^2} \sum_{i=0}^{31} \sum_{j=0}^{31} f(x - 16 + i, y - 16 + j)$$

where $f(x, y)$ is the filtered image.

The binary image produced as the output of the thresholding stage is the enhanced fingerprint (Fig. 4*c*). This image is intended to be close to the idealised one 'imagined' by a trained fingerprint officer when examining a raw print.

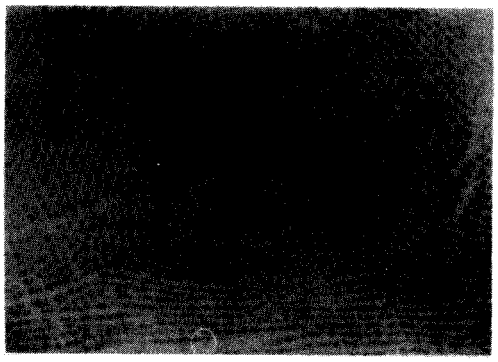
## 6 Results

The enhancement algorithm was applied to fourteen inked fingerprint images of various classification types.

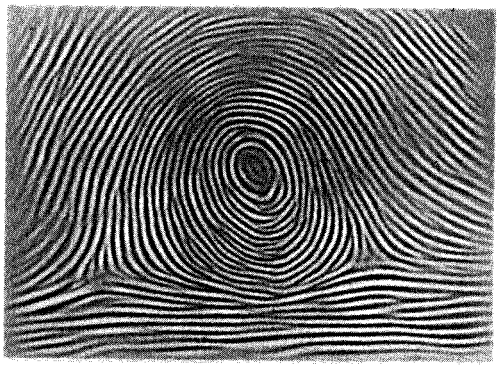
Fig. 6 shows a representative sample of these images, together with their filtered and enhanced versions.

## 7 Evaluation of results

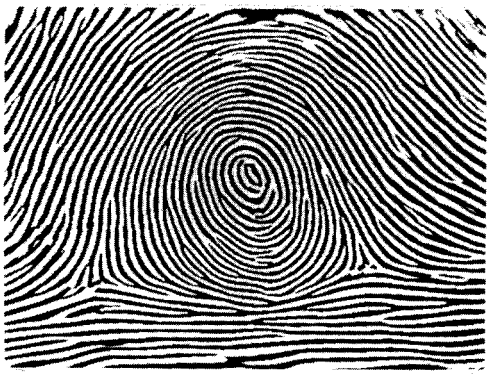
Image enhancement has been defined [24] as the processing of an image so that the enhanced version is more suitable than the original for a specific application. The quality of this fingerprint enhancement technique should therefore be judged in terms of the suitability of the enhanced prints for the specific application of personal identification using an AFIS.



a



b



c

**Fig. 4** *Stages of the fingerprint enhancement process*
*a* Original fingerprint
*b* After filtering
*c* After thresholding

To perform the evaluation, use was made of the AFIS developed by the Home Office in the UK [20]. First, two enhancements of each fingerprint were obtained, one using the Home Office algorithm, and the other using the algorithm proposed in this paper. These enhancements were both submitted to the encoding stage of the Home Office AFIS, and the encoded outputs were then compared quantitatively.

The encoding stage of the AFIS consists of two phases.

1. Minutiae detection: analysis of the enhanced image yields a list of candidate minutiae. Among these are usually a large proportion of 'false minutiae', i.e. points which have been incorrectly identified as minutiae. These are further categorised into 'hard' and 'soft' false minutiae. Soft false minutiae are caused by noisy imperfections in the enhanced image, such as jagged ridge edges, isolated dots and small holes in the ridges. Hard false minutiae occur where the enhanced image possesses apparently true minutiae which are not present in the original fingerprint.

2. Minutiae reduction: further processing of the image and minutiae list removes most of the false minutiae. Unfortunately, this often yields 'missed minutiae', i.e. the removal of true minutiae from the list.

Table 1 shows numbers of minutiae found by phase 1, numbers of minutiae remaining after phase 2, and processing times for phase 2. A human fingerprint expert determined 'ground truth' numbers of minutiae within each print, as well as the numbers of missed and false minutiae remaining after the completion of phase 2. These numbers are presented in Table 2, which also includes sensitivities and specificities, defined as follows:

$$\text{Sensitivity} = 1 - \frac{\text{Missed minutiae (after reduction)}}{\text{Ground truth number of minutiae}}$$

$$\text{Specificity} = 1 - \frac{\text{False minutiae (after reduction)}}{\text{Ground truth number of minutiae}}$$

Tables 1 and 2 yield the following comparisons between the two enhancement algorithms:

1. Total number of candidate minutiae detected (before reduction): this indicates the relative degree of noisiness of the two enhancements, since noisy artefacts cause false minutiae. In every case, considerably more minutiae were detected on the Home Office enhancements. The Home Office enhancements generated 319.5 minutiae per image (3.50 time ground truth). The proposed algorithm generated 133.7 minutiae per image (1.43 time ground truth).

2. Time taken for minutiae reduction: because of the greater numbers of false minutiae detected on the Home Office enhancements, one expects longer processing times for minutiae reduction. Indeed, the processing times were found to average 2.43 times longer for the Home Office enhancements. Using the proposed enhancements saved an average of 58 seconds per image.

3. False minutiae remaining after minutiae reduction: since the minutiae reduction process removes almost all of the 'soft' false minutiae, the number of minutiae remaining approximates the number of 'hard' errors that have occurred during the enhancement process. Consequently this number provides an important evaluation of the enhancement algorithm. Averages of 21.5 and 6.9 false minutiae per image were produced using the Home Office and proposed enhancements, respectively. The respective specificities are 76.0% and 92.3%. The pro-

posed method reduces the false minutiae count by a factor of 3.1, again a very favourable figure.

4. Missing minutiae after minutiae reduction: 11.2 and 19.6 minutiae per image were missed by the Home Office and our enhancements, respectively. The respective sensitivities are 88.0% and 79.6%.

All of the above comparisons, except 4, strongly favour our algorithm. Because of the disproportionately large numbers of candidate minutiae associated with the Home Office algorithm, the Home Office minutiae reduction algorithm has been made particularly stringent. When applied to the proposed enhancements, therefore, a larger than desirable proportion of the true minutiae are removed. We would expect that a less severe tuning of this algorithm would reduce the number of missed minutiae without significantly increasing the number of false minutiae.

The enhancement may be performed using 268 million multiply/add operations. If eight orientations are used rather than sixteen, the number of operations is reduced to 141 million. This latter figure is comparable to the 100 million operations required for the method of Mehtre [10], which also uses eight orientations. The comparison becomes more favourable when we note that in our algorithm, most of the operations are accounted for by the two-dimensional FFTs used to form the prefiltered images. Highly efficient hardware to perform FFTs is standard on modern image-processing systems. Excluding FFT-related operations, our algorithm requires only 33 million operations when using 16 orientations.

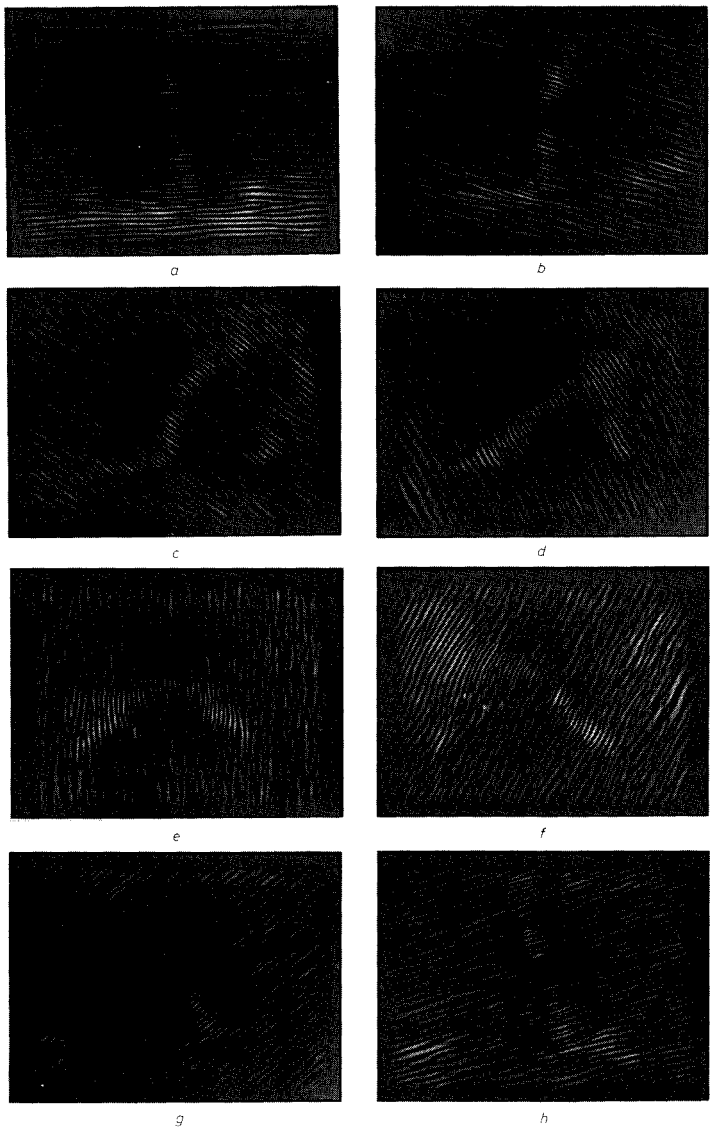



**Fig. 5** *Eight of the 16 directionally filtered versions of Fig. 3a. These are the prefiltered images which were combined to generate Fig. 3b*
*a* 0 degrees *b* 22.5 degrees *c* 45 degrees *d* 67.5 degrees *e* 90 degrees *f* 112.5 degrees *g* 135 degrees *h* 157.5 degrees

Our algorithm incorporates features which constitute improvements over earlier techniques. Incorporation of a mathematical model of ridge topology leads to improved LRO estimates in the difficult high-curvature regions near singular points, and hence to more effective filtering in such regions. Use of prefiltered images and Fourier

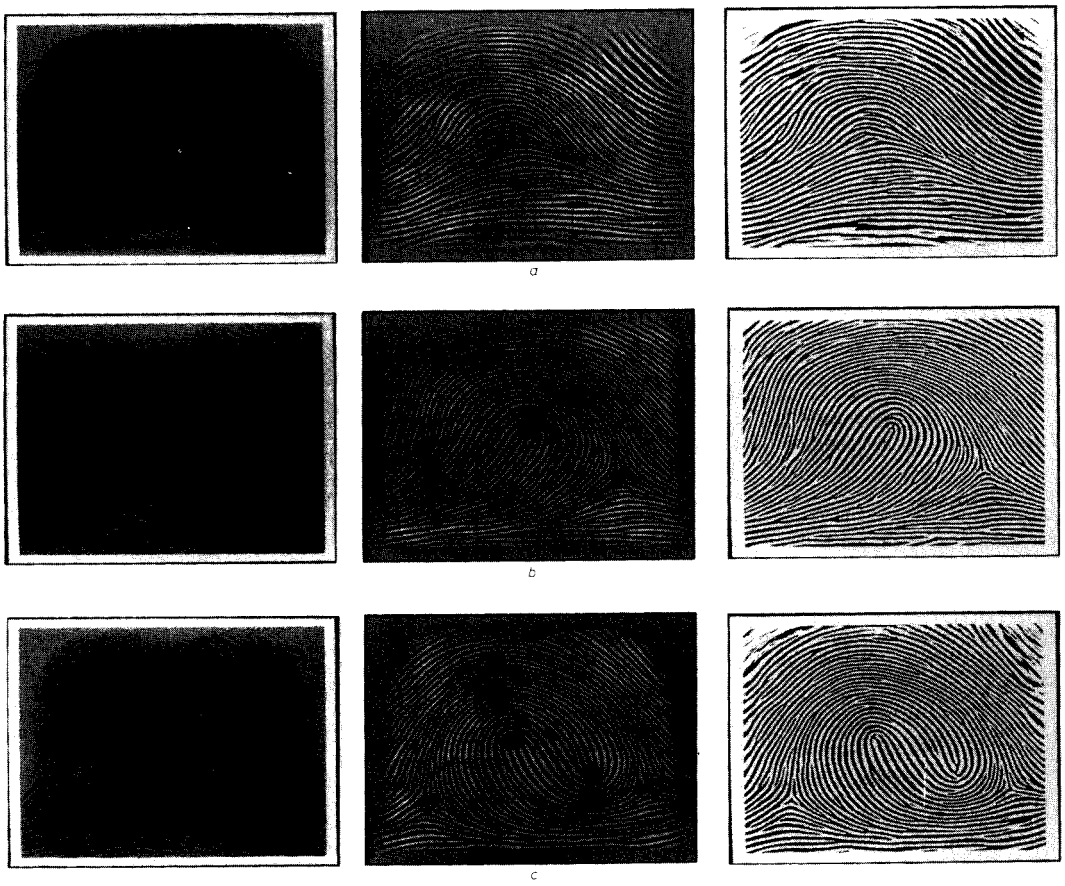


**Fig. 6** *Enhancements of inked fingerprints of various classification types. For each print the raw image, filtered image and thresholded enhancement are shown*

*a* Plain arch *b* Loop *c* Double loop

**Table 1: Results of application of AFIS minutiae detection and reduction algorithms to enhanced fingerprint images**

| Fingerprint | Total minutiae found by minutiae detection algorithm | | Minutiae remaining after minutiae reduction | | Time to perform minutiae reduction (seconds) | |
|---|---|---|---|---|---|---|
| | Proposed* | Existing† | Proposed* | Existing† | Proposed* | Existing† |
| 1 | 114 | 402 | 84 | 100 | 35 | 125 |
| 2 | 113 | 313 | 71 | 78 | 30 | 92 |
| 3 | 190 | 560 | 79 | 122 | 53 | 200 |
| 4 | 143 | 355 | 77 | 92 | 45 | 100 |
| 5 | 165 | 409 | 88 | 122 | 52 | 128 |
| 6 | 132 | 300 | 80 | 99 | 40 | 87 |
| 7 | 132 | 243 | 82 | 118 | 44 | 72 |
| 8 | 125 | 233 | 86 | 122 | 45 | 75 |
| 9 | 169 | 467 | 78 | 101 | 51 | 164 |
| 10 | 98 | 310 | 68 | 89 | 28 | 85 |
| 11 | 95 | 333 | 64 | 82 | 31 | 98 |
| 12 | 116 | 214 | 89 | 92 | 40 | 61 |
| 13 | 139 | 226 | 85 | 105 | 38 | 75 |
| 14 | 158 | 237 | 114 | 135 | 55 | 79 |
| 15 | 116 | 190 | 74 | 92 | 36 | 57 |

* Using proposed enhancement algorithm
† Using Home Office enhancement algorithm

**Table 2: Analysis of enhanced fingerprints by fingerprint expert. The table shows the 'ground truth' numbers of minutiae as well as the numbers of false and missed minutiae**

| Fingerprint | Ground truth (true number of minutiae) | False minutiae | | Missed minutiae | | Sensitivity, % | | Specificity, % | |
|---|---|---|---|---|---|---|---|---|---|
| | | Proposed* | Existing† | Proposed* | Existing† | Proposed* | Existing† | Proposed* | Existing† |
| 1 | 87 | 8 | 24 | 9 | 11 | 89.7 | 87.4 | 90.8 | 72.4 |
| 2 | — | — | — | — | — | — | — | — | — |
| 3 | 74 | 18 | 52 | 13 | 4 | 82.4 | 94.6 | 75.7 | 29.7 |
| 4 | — | — | — | — | — | — | — | — | — |
| 5 | 102 | 6 | 29 | 20 | 9 | 80.4 | 91.2 | 94.1 | 71.6 |
| 6 | 92 | 9 | 18 | 19 | 9 | 79.3 | 90.2 | 90.2 | 80.4 |
| 7 | 113 | 4 | 15 | 27 | 10 | 76.1 | 91.2 | 96.5 | 86.7 |
| 8 | 107 | 4 | 24 | 25 | 9 | 76.6 | 91.6 | 96.3 | 77.6 |
| 9 | 84 | 17 | 41 | 23 | 24 | 72.6 | 71.4 | 79.8 | 51.2 |
| 10 | 83 | 3 | 16 | 18 | 10 | 78.3 | 88.0 | 96.4 | 80.7 |
| 11 | 80 | 7 | 16 | 23 | 14 | 71.3 | 82.5 | 91.3 | 80.0 |
| 12 | 92 | 6 | 10 | 9 | 10 | 90.2 | 89.1 | 93.5 | 89.1 |
| 13 | 108 | 0 | 10 | 23 | 13 | 78.7 | 88.0 | 100.0 | 90.7 |
| 14 | 131 | 7 | 13 | 24 | 8 | 81.7 | 93.9 | 94.7 | 90.1 |
| 15 | 96 | 0 | 11 | 22 | 15 | 77.1 | 84.4 | 100.0 | 88.5 |

* Using proposed enhancement algorithm
† Using Home office enhancement algorithm

domain filtering ensures that our filters' spatial extent is not restricted to a small neighbourhood, leading to more effective reduction of unwanted noise.

## 8 Conclusion

The fingerprint enhancement algorithm presented in this paper is of importance because it has been shown to produce significant improvements over the algorithm incorporated in a working AFIS. This demonstrates the usefulness of position-dependent Fourier domain filtering in the processing of fingerprint images within AFIS systems.

## 9 References


1 KASS, M., and WITKIN, A.: 'Analyzing oriented patterns', *Computer Vision, Graphics and Image Processing*, 1987, **37**, pp. 362–385
2 ZUCKER, S.W.: 'Early orientation selection: tangent fields and the dimensionality of their support', *Computer Vision, Graphics and Image Processing*, 1985, **32**, pp. 74–103
3 DAVID, C., and ZUCKER, S.W.: 'Potentials, valleys, and dynamic global coverings', *International Journal of Computer Vision*, 1990, **5**, pp. 219–238
4 PARENT, P., and ZUCKER, S.W.: 'Trace inference, curvature consistency, and curve detection', *IEEE Trans. Acoust., Speech Signal Process.*, 1989, **11**, pp. 823–839
5 KNUTSSON, H.E., WILSON, R., and GRANLUND, G.H.: 'Anisotropic nonstationary image estimation and its applications: Part I — Restoration of noisy images', *IEEE Trans. Commun.*, 1983, **COM-31**, pp. 388–397
6 O'GORMAN, L., and NICKERSON, J.V.: 'An approach to fingerprint filter design', *Pattern Recognition*, 1989, **22**, pp. 29–38
7 O'GORMAN, L., and NICKERSON, J.V.: 'Matched filter design for fingerprint image enhancement'. IEEE International Conference on *Acoustics, Speech, and Signal Processing*, New York, 1988, pp. 916–919
8 MEHTRE, B.M., and CHATTERJEE, B.: 'Segmentation of fingerprint images — a composite method', *Pattern Recognition*, 1989, **22**, pp. 381–385
9 MEHTRE, B.M., MURTHY, N.N., KAPOOR, S., and CHATTERJEE, B.: 'Segmentation of fingerprint images using the directional image', *Pattern Recognition*, 1987, **20**, pp. 429–435
10 MEHTRE, B.M.: 'Fingerprint image analysis for automatic identification', *Machine Vision and Applications*, 1993, **6**, pp. 124–139
11 SRINIVASAN, V.S., and MURTHY, N.N.: 'Detection of singular points in fingerprint images', *Pattern Recognition*, 1992, **25**, pp. 139–153
12 VERMA, M.R., MAJUMDAR, A.K., and CHATTERJEE, B.: 'Edge detection in fingerprints', *Pattern Recognition*, 1987, **20**, pp. 513–523
13 XIAO, Q., and RAAFAT, H.: 'Fingerprint image postprocessing: a combined statistical and structural approach', *Pattern Recognition*, 1991, **24**, pp. 985–992
14 KAWAGOE, M., and TOJO, A.: 'Fingerprint pattern classification', *Pattern Recognition*, 1984, **17**, pp. 295–303
15 GRASSELLI, A.: 'On the automatic classification of fingerprints — some considerations on the linguistic interpretation of pictures', *in* WATANABE, S. (Ed.): 'Methodologies of pattern recognition' (Academic Press, 1969)
16 MOAYER, B., and FU, K.S.: 'A syntactic approach to fingerprint pattern recognition', *Pattern Recognition*, 1975, **7**, pp. 1–23
17 RAO, C.V.K., and BALCK, K.: 'Type classification of fingerprints: a syntactic approach', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1980, **PAMI-2**, pp. 223–231
18 RAO, T.C.M.: 'Feature extraction for fingerprint classification', *Pattern Recognition*, 1976, **8**, pp. 181–192
19 CHIRALO, R.P., and BERDAN, L.L.: 'Adaptive digital enhancement of latent fingerprints'. Proceedings of 1978 Carnahan conference on *Crime Countermeasures*, Lexington, Kentucky, 1978, pp. 131–135
20 MILLARD, K.: 'Developments on automatic fingerprint recognition'. International Carnahan conference on *Security Technology*, Zurich, October 1983, pp. 173–178
21 RIGANATI, J.P.: 'An overview of algorithms employed in automated fingerprint processing'. Proceedings of international conference on *Crime countermeasures — science and engineering*, University of Kentucky, July 1977, pp. 125–131
22 GRANLUND, G.H., and ARVIDSSON, J.: 'The GOP image computer', *in* FAUGERAS, O.D. (Ed.): 'Fundamentals in computer vision: an advanced course' (Cambridge University Press, 1983), pp. 57–67
23 OPPENHEIM, A.V., and SCHAFER, R.W.: 'Digital signal processing' (Prentice-Hall, 1975)
24 GONZALEZ, R.C., and WINTZ, P.: 'Digital image processing'. Applied Mathematics and Computation Textbooks, No. 13, Addison Wesley, 1977
25 SHERLOCK, B.G., and MONRO, D.M.: 'A model for interpreting fingerprint topology', *Pattern Recognition*, 1993, **26**, pp. 1047–1055
26 SHERLOCK, B.G., MONRO, D.M., and MILLARD, K.: 'Algorithm for enhancing fingerprint images', *Electron. Lett.*, 1992, **28**, pp. 1720–1721
27 'The science of fingerprints: classification and uses'. Federal Bureau of Investigations, US Government Printing Office, Washington, DC, 1984