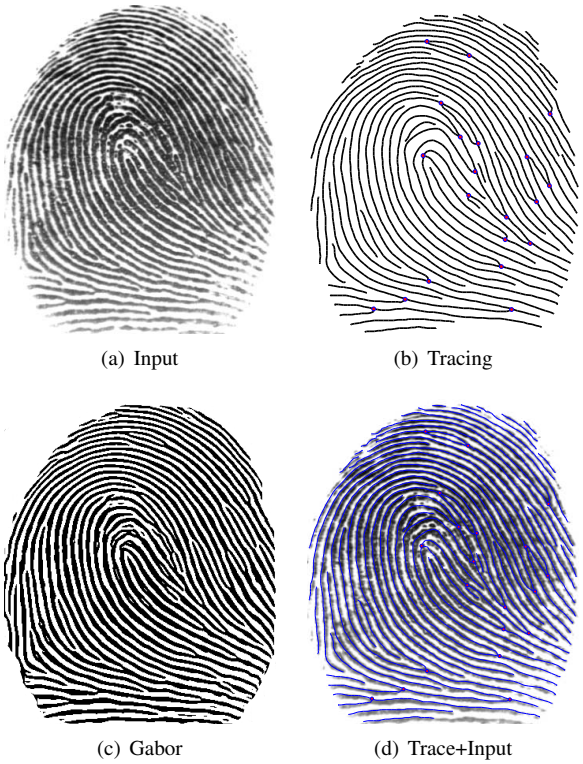
# Fingerprint Feature Extraction from Gray Scale Images by Ridge Tracing

Devansh Arpit and Anoop Namboodiri
IIIT Hyderabad
Gachibowli, Hyderabad 500 032, India
devansharpit@gmail.com, anoop@iiit.ac.in

## Abstract

*This paper deals with extraction of fingerprint features directly from gray scale images by the method of ridge tracing. While doing so, we make substantial use of contextual information gathered during the tracing process. Narrow bandpass based filtering methods for fingerprint image enhancement are extremely robust as noisy regions do not affect the result of cleaner ones. However, these method often generate artifacts whenever the underlying image does not fit the filter model, which may be due to the presence of noise and singularities. The proposed method allows us to use the contextual information to better handle such noisy regions. Moreover, the various parameters used in the algorithm have been made adaptive in order to circumvent human supervision. The experimental results from our algorithm have been compared with those from Gabor based filtering and feature extraction, as well as with the original ridge tracing work from Maio and Maltoni [11]. The results clearly indicate that the proposed approach makes ridge tracing more robust to noise and makes the extracted features more reliable.*



(a) Input (b) Tracing (c) Gabor (d) Trace+Input

Figure 1. Tracing vs. Enhancement: Tracing result in ridges and minutiae extracted from the input image, while enhancement result in a cleaner (often binary) image.

## 1. Introduction

Fingerprints can be characterized by their local ridge and valley flow patterns, which ultimately result in a set of distinctly distributed minutiae (also called Galton's characteristics [3]). Among the 150 classified minutiae, the two most prominent ones are ridge endings and bifurcations, which are sudden changes in the ridge flow pattern. The combination of such minutiae along with the neighborhood ridge flow pattern in a fingerprint is observed to be unique for every person. However, the acquired image from a finger tends to vary considerably due to skin conditions, presentation, noise in sensor, etc. Hence a method that can reliably detect the ridge lines, ridge endings and bifurcations can clearly improve the accuracy and usefulness of fingerprint based identification systems.

Due to the variety in noise present in the fingerprint images acquired in practice, most fingerprint identification systems employ an enhancement stage to filter out as much noise as possible and work on the resulting cleaner image. An alternate approach is to try and extract the ridges and minutiae directly from the original image using a method that is insensitive to the noise present. Figure 1 shows an example of tracing ridges in a gray scale fingerprint image (using the proposed method) as compared to enhancing it to obtain a binary image for feature extraction. As noted from the figure, while the enhancement stage reduces the



978-1-4577-1359-0/11/$26.00 ©2011 IEEE

noise and variability in a fingerprint, it results in removal of some of the useful information as well as introduction of certain artifacts, which can adversely affect further stages. These are essentially the result of incompatibility between the fingerprint model used for enhancement and the input fingerprint. The process of ridge tracing with gray scale images are also prone to incorrect feature extraction if we do not use an appropriate ridge model. The current work tries to develop a meaningful ridge model that can effectively see through a wide variety of noises.

Typical methods for automatic fingerprint feature extraction follows through a sequence of steps such as image enhancement, binarization, thinning, minutiae extraction, and post processing [7]. Common approaches to image enhancement include anisotropic and Gabor filtering [4], and filtering in the frequency domain [9]. The most popular method of spatial domain filtering uses a bank of Gabor filters as proposed by Hong *et al.* [5]. Due to the ability of this method to deal with a wide variety of noises, it has been adopted and extended in a variety of ways. Such attempts include Greenberg *et al.* [4], Bernard *et al.* [1], Yang *et al.* [13] and Zhu *et al.* [14].

Filter based approaches work on enhancing the image independently at every pixel, taking into account a spatial neighborhood. This makes the methods robust in the sense that noise at one point does not affect the restoration at other regions. However, these methods may not be able to resolve any ambiguities present in the noisy regions of a fingerprint, resulting in formation of spurious minutiae in the final result. A method that directly deals with ridges, valleys and minutiae can make use of the larger contextual information to perform better in such cases.

The first effort in this direction was made by Maio and Maltoni [11]. They proposed the idea of seeding of ridges and tracing them using the intensity signature in a direction perpendicular to the orientation field. This idea was followed by others who tried to improve on [11]. Such attempts include improving the robustness of ridge tracking based on ridge connectivity and randomicity by Feng *et al.* [2], and the use of probabilistic ridge tracking by maintaining multiple hypotheses during tracking by Ma*et al.* [10] based on the condensation algorithm by Isard and Blake [6]. The primary advantage of direct tracing is that one can employ a ridge model that is versatile, lending the method the ability to work with all natural variations of fingerprints. However, the fundamental difficulty in all the above algorithms has been the lack of ability of the tracing step to work in presence of noise. In practice all the above algorithms resort to a filter-based enhancement step before reliable tracing of ridge can be carried out.

This work focuses on developing a method for tracing that work directly on an unfiltered gray-scale fingerprint image, thus avoiding any artifacts of filtering. To achieve this we propose robust methods for: i) seeding pixels to start tracing, ii) determining ridge direction during tracing, taking into account the common ridge variations including pores, varying ridge width, ridge frequency and intensity, and iii) terminating the tracing when minutiae or noisy regions are encountered. Moreover, we adaptively update all the parameters used as part of the tracing step itself, thus removing the need for any manual input.

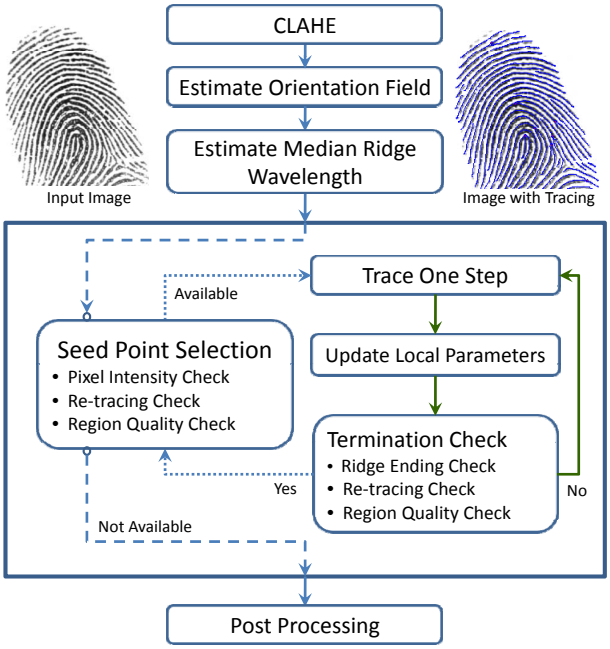



Figure 2. Overview of the proposed tracing algorithm.

## 2. Ridge Tracing of Fingerprints

Figure 2 provides an overview of the tracing process. The high-level steps are similar to what was proposed by Maio and Maltoni [11], and the differences are primarily in how each step is carried out. As a pre-processing step, the input fingerprint images are first passed through a *contrast limited adaptive histogram equalization* (CLAHE) stage to reduce the variations in intensity and contrast. This involves block-wise histogram equalization of parts of the image followed by integrating them using bilinear interpolation to remove any artificial boundaries. The second stage involves computation of an orientation field of the image, resulting in the average ridge orientation for every block ($16 \times 16$) using the LMS orientation estimation algorithm [5]. Figure 3 shows the result of the above steps on a fingerprint image. We also compute the median ridge wavelength, $\lambda_m$, of the image from the inter-ridge distances computed at local windows.

The core ridge tracing algorithm consists of repetitions of two consecutive stages: seeding and tracing of a spe-

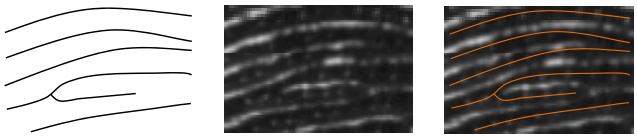
cific ridge. These two stages are shown on either side in the flowchart in Figure 2. Once all the ridges are traced, a post-processing step joins any broken ridges to avoid detection of spurious minutiae. We now look at the specific methods we use in the seeding and tracing steps.

(a) Input (b) CLAHE (c) Orientation Field

Figure 3. Outputs of adaptive histogram equalization and the orientation field computed from the image.

## 2.1. The Tracing Algorithm

As the name suggests, the process of ridge tracing involves moving along a fingerprint ridge while at the same time recording the positions traversed so that once all the ridges in a fingerprint have been traversed, we get a digital representation of the ridges, their local orientations and the detected minutiae. Ideally, tracing a thin ridge line would involve simply proceeding in the direction of local orientation field (see Figure 4(a)). But this does not hold true in most cases. In practice, ridges are thick with pores and considerable amount of noise in them (see figure 4(b)). Hence the process of tracing should be thought of as generating a trace for every ridge that traverses through the center of the ideal ridges (see figure 4(c)).



(a) Ideal Ridges (b) Real Ridges (c) Traced Ridges

Figure 4. Difference between tracing ideal ridges and noisy real world ridges with pores.

This seemingly simple change makes the process of tracing extremely difficult as one needs to determine the center of the ideal ridge in presence of artifacts such as cuts, creases, pores and dirt and various noises from the sensor. We now look at this important step of determining the direction of the next step in tracing assuming we are already on a ridge.

The process of ridge tracing starts with choosing a starting point on a ridge. Once this is done, the algorithm goes on tracing that ridge until a terminating condition is met.

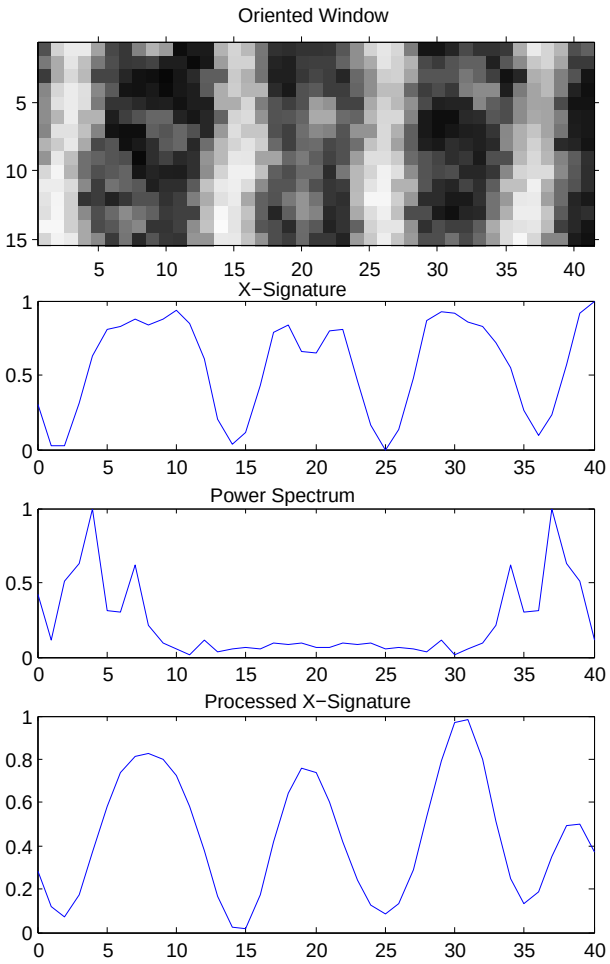



Figure 5. Oriented window of a part of fingerprint followed by its x-signature, power spectrum of the x-signature and conditioned x-signature.

The process of ridge tracing proposed by Maio and Maltoni [11] involves stepping a fixed distance in the direction of local orientation and then adjusting the current position to move to a local maximum. This step-and-correct approach often leads to poor tracing in noisy fingerprints. The fundamental improvement we introduce here is a method that analyzes a local window ahead of current position in the direction of tracing, and determines the quality of the region, the presence and location of the ridge and the presence of any ridge ending. This careful approach allows us to trace more consistently over the correct ridges, even in presence of noise. We now look at the process of analysis of the local window.

Once a ridge is seeded, at each step in tracing, an oriented window is extracted around the current position that is orthogonal to the local orientation field. The width and height of the window are a function of the local ridge wavelength calculated from previous step.

Figure 5 shows such an oriented window from a portion of the fingerprint such as the top-left corner of Figure 4(b). We then calculate the x-signature of the oriented window, which is analyzed using it's power spectrum to determine the next step. The x-signature and the power spectrum computed from the oriented window are shown below the window in Figure 5. The power spectrum is used to compute the direction of movement as shown by the peak in the last row of Figure 5.

We make use of power spectrum of the x-signature of the oriented window around a pixel of concern for analyzing various aspects of the region and around that pixel throughout the process of ridge tracing. We make the following observations about the power spectrum of the x-signature. A signal with zero mean with $n$ ($> 1$) number of cycles in it will show a peak at the $n^{th}$ harmonic (see figure 5).

To determine the peak of the current ridge, we filter the above signal by a suppressing all frequencies outside the range $[n-1,\ n+3]$ along with the harmonic $2n$. Note that the filtering of the x-signature is different from applying a band-pass filter to the fingerprint image itself. An ideal x-signature is very close to a single frequency around $n$. In our case, since we have an estimated value ridge frequency at each step, the above mentioned range of frequencies clearly allows for any discrepancy with the actual ridge frequency estimate generated due to measurement error as well as the presence of minutiae. Another advantage of the above filtering is that it conserves the location of the signal peaks as the phase values of the primary signal are not removed in the filtering process. Hence we can find the ridge centers more reliably during the tracing, even in the presence of noise. This happens because of the slight variations in local ridge frequency in a window in fingerprints. The removal of lower harmonics helps us to eliminate smudges between ridges, while that of higher harmonics (including $2n$) allows us to remove pores and other high frequency noises.

Another use of the spectrum analysis is in determining the quality a given local window. We note that the major peaks observed in case of noisy regions are in the lower harmonics and the higher harmonics rarely have amplitude higher than that of lower or allowed harmonics. Hence, any peak in the lower harmonic of the spectrum will necessarily be a result of noise. Thus a ratio of the maximum amplitude amongst the harmonics in the allowed spectrum to the maximum amplitude amongst the lower harmonics gives a good estimate of the noise present in a region. On the other hand, a peak in the allowed window of spectrum gives us a good estimate of the actual ridge frequency, which is then used as the ridge frequency value in the proceeding steps of ridge tracing.

## 2.2. Determining Seeding Points

A starting point for tracing a specific ridge is referred to as a seed. Determination of seeding points is extremely crucial for good tracing. We must ensure that the seeding point is always on a ridge in a clean region. Starting from a noisy region or non-ridge location can result in erratic traces and interconnects. While the former is ensured by a simple heuristic check of the intensity value of a candidate pixel, the latter is done based on a power spectrum based quality assessment of the seeding point, as mentioned before. A third check ensures that the point is not already traced. More specifically, these conditions that are checked for seed point selection are:

1. *Pixel Intensity Check:* Since we have already performed histogram equalization on the fingerprint image, a simple threshold check is performed on every pixel. i.e., seed the pixel only if $I(j,k) \leq \alpha$, where $\alpha$ is a threshold.
2. *Retracing Check:* If a pixel qualifies the intensity check, it is tested to avoid retracing of an already traced ridge. The information whether a pixel has already been traced or not is stored in an image $I'$, which has the same size as that of the original fingerprint image. This image has markings for the already traced pixels and is continuously updated during tracing. The retracing check is done by looking for any connectivity between the candidate ridge (at which the seed point lies) and the label image (which stores the already traced ridges). This can be mathematically formulated as follows:

1. Compute an oriented window $\Omega$ of size $w \times h$ ($\lambda_m \times \lambda_m$) centered at pixel $(j,k)$, $\lambda_m$ is the median ridge wavelength calculated previously.

2. Binarize the window $\Omega$ by thresholding at it's mean intensity and calculate the new window $\Omega'$ such that ridges are 1 and valleys are 0.

$$\Omega'(l,m) = \begin{cases} 1 & \text{if } \Omega(l,m) < \tau; \\ 0 & \text{otherwise.} \end{cases}$$

   where $\tau$ is the mean intensity of the window.

3. Set the value of all the pixels in $\Omega'$ to 0 which are not a part of the center ridge block (the ridge being traced). Thus, now only the value of pixels belonging to the ridge to be traced are 1.

4. Similarly compute an oriented window $\Omega_t$ of size $w \times h$ ($\lambda_m \times \lambda_m$) centered at pixel $(j,k)$ in the label image $I'$. $\lambda_m$ is the median ridge wavelength calculated previously.

5. Multiply the matrix elements of $\Omega'$ and $\Omega_t$, i.e each element in $\Omega'$ with it's corresponding element in $\Omega_t$ .Then compute the sum of all the elements in the resultant

matrix. T=sum($\Omega'$ .*$\Omega_t$) where '.*' denotes the above mentioned multiplication operation. Thus, a ridge is deemed already traversed if: T>0

3. *Region Quality Check:* If a pixel qualifies the above two conditions, a quality check is performed on it as follows:

1. Compute an oriented window $\Omega$ of size $w \times h$ ($2.5\lambda_m \times \lambda_m$) centered at pixel $(j, k)$, $\lambda_m$ is the median ridge wavelength calculated previously.

2. Recompute the intensity value of pixels in $\Omega$ to form $\Omega'$ such that ridges are maxima and valleys are minima:

$$\Omega'(l, m) = g - \Omega(l, m) \forall (l, m) \in \Omega,$$

where g is the gray levels in image I. $\Omega'$ will now be recognized as our new oriented window.

3. Compute the x-signature $\bar{X} = X[0 \cdots w-1]$ of the oriented window $\Omega'$. and normalize it to have $X[i] \in [0, 1]$.

4. Calculate the power spectrum $P[k]$ of the x-signature, where $k = 0$ denotes the zeroth harmonic, $k = 1$ denotes the first harmonic and so on.

5. Calculate the harmonic ratio (HR) defined as:

$$HR = \frac{\max(P[n-1], P[n], \ldots, P[n+3])}{\max(P[0], P[1], \ldots, P[n-2])},$$

where $n$ is the total number of ridges present in the oriented window.

6. A pixel qualifies the quality check if $HR > \beta_1$, where $\beta_1$ is a threshold.

Finally, if a pixel qualifies all the three conditions mentioned above, it is declared a valid seed to start a trace. Otherwise it is skipped and the next pixel is tested for seeding.

## 2.3. Steps for Tracing

Once a seed has been chosen, the process of tracing a ridge begins. Tracing is a continuous process that is made to stop only when a termination condition is met. During this process, two strings are made to grow from a seed at the local orientation angle in directions opposite to each other. This is done to be able to trace the entire ridge even if the seeding point is not exactly at a terminal point of that ridge. Each of these two strings can grow independent of each other. In case of each string, the angle $\theta_t$ in which the string moves from its current node to the next node is the sum of the local orientation angle $\theta_o$ of the ridges, that has been precalculated, added to the angular change $\theta_c$ required to make the string reach the center of the ridge at the next step. The distance (between two consecutive nodes) moved at the angle $\theta_t$ should be such that it is large enough to reduce the computational requirements but small enough to prevent the string from jumping over to another ridge at high curvature areas. Thus it becomes obvious that the inter-node distance value should be a function of the local ridge wavelength $\lambda$. The optimal value of inter-node distance value was observed to be $\lambda/2$. At each step of tracing, the following operations are performed

1. Compute an oriented window $\Omega$ of size $w \times h$ ($2.5\lambda \times \lambda$) centered at pixel $(j, k)$, and orthogonal to the local ridge orientation $\theta_o$. $\lambda$ is the ridge wavelength calculated in the previous step of tracing.

2. Recompute the intensity value of pixels in $\Omega$ to form $\Omega'$ such that ridges are maxima and valleys are minima:

$$\Omega'(l, m) = g - \Omega(l, m) \forall (l, m) \in \Omega,$$

where g is the gray levels in image I. $\Omega'$ will now be recognized as our new oriented window.

3. Compute the x-signature $\bar{X} = X[0 \cdots w-1]$ of the oriented window $\Omega'$. and normalize it to have $X[i] \in [0, 1]$.

4. Calculate the power spectrum $P[k]$ of the x-signature, where $k = 0$ denotes the zeroth harmonic, $k = 1$ denotes the first harmonic and so on.

5. Calculate the harmonic ratio (HR) defined as:

$$HR = \frac{\max(P[n-1], P[n], \ldots, P[n+3])}{\max(P[0], P[1], \ldots, P[n-2])},$$

where $n$ is the total number of ridges present in the oriented window.

6. If $HR < \beta_2$, where $\beta_2$ is a threshold, increase the width of the oriented window to $4.5\lambda$ so that it accommodates 5 ridges (2 ridges on either side of the ridge being traced) in it and perform steps 2 through 5 again. If the value of harmonic ratio is still below the threshold, terminate the string.

7. If the previous condition is satisfied, eliminate harmonic 2n along with all harmonics from the x-signature signal $\bar{X}$ other than $[n-1, n+3]$ with the exception of 2n to get the new signal $X_{new}$, where $n$ is the total number of ridges in the oriented window.

8. Let $X[c]$ correspond to the current position of the string in $X$. Then the change $\Delta c$ required for the string to reach the center of the ridge will be achieved by finding the position of local maxima in $X_{new}$ such that no minima exists in $X_{new}$ between positions $c$ and $(c + \Delta c)$. From the most basic definition of a local maxima,

$$X_{new}[c + \Delta c - 1] < X_{new}[c + \Delta c] > X_{new}[c + \Delta c + 1],$$

such that

$$X_{new}[i] \geq X_{new}[c] \forall c < i < c + \Delta c$$

The latter condition has been applied to ensure that the string does not jump on to another ridge in search of a local maxima.

9. Calculate angle $\theta_c = tan^{-1}\left(\frac{\Delta c}{0.5\lambda}\right)$, and $\theta_t = \theta_o + \theta_c$

10. The new node of the string is given by

$$\begin{aligned} j_{new} &= j + 0.5\lambda \cos\theta_t \\ k_{new} &= k + 0.5\lambda \sin\theta_t \end{aligned}$$

11. Mark the label image $I'$ with the new node and its path following the previous node.

12. Finally, we calculate the value of ridge wavelength at the new node as follows

$$\lambda_{new} = w/H_{max},$$

Set

$$\lambda = \lambda_{new},$$

where $H_{max}$ is the harmonic index in the allowed window of the power spectrum $P$, which has the maximum amplitude expressed as: $P[H_{max}] = \max(P[n-1], P[n]...P[n+3])$.

13. If the value of wavelength falls in the valid range of 3 to 20 pixels, it is accepted and added to an array which stores the values of ridge wavelength at each step. Otherwise a mode of the wavelength values stored in the mentioned array (calculated in the previous steps) is used.

In step 6 of the tracing algorithm we have increased the width of the oriented window in order to make better use of local context of ridge flow patterns in cases of noisy regions as has been sensed from the low value of harmonic ratio. We otherwise make use of a smaller window accommodating three ridges (two ridges on either side of the ridge being traced) because of the fact that the local orientation estimates are more inconsistent as we increase the window size, especially in cases of regions with high curvature or with minutiae in them.

#### 2.3.1 Termination Conditions

Termination of ridge tracing occurs primarily when it encounters a ridge ending or an already traced ridge (bifurcation). Tracing is also terminated if an extremely noisy region is encountered. Of the above three conditions, the second and third are already discussed in connection with seeding. The detection of ridge ending proceeds as follows:

1. Compute an oriented window $\Omega$ of size $w \times h$ ($\lambda \times \lambda$) centered at pixel $(j, k)$. A width of $\lambda$ ensures that the window contains only the ridge being traced in it.

2. Binarize the window $\Omega$ by thresholding at it's mean intensity and calculate the new window $\Omega'$ such that ridges are 1 and valleys are 0.

$$\Omega'(l,m) = \begin{cases} 1 & \text{if } \Omega(l,m) < \tau; \\ 0 & \text{otherwise.} \end{cases}$$

where $\tau$ is the mean intensity of the window.

3. Calculate the fraction of ridge at the x-coordinate $(c + \Delta c)$ of the window $\Omega'$ by summing the vertical column at $(c + \Delta c)$.

$$RF(RidgeFraction) = \frac{1}{\lambda}\sum_{m=0}^{\lambda} \Omega'(c + \Delta c, m),$$

where $c$ denotes the x-coordinate center of the window $\Omega'$ and $\Delta c$ is the value calculated from the tracing algorithm.

4. If the value of ridge fraction is below a certain threshold, the string is made to stop tracing further. In our case we have chosen this value to be 0.3.

In case a retrace is detected during the termination condition check, the point is marked as a bifurcation along with the point in the traced ridge, where it meets.

### 2.4. Post-Processing

Once all the ridges in a fingerprint have been traced, we start with post processing, in which all string terminals facing each other are joined in order to avoid spurious ridge endings.

## 3. Experimental Results

In this section we compare our results quantitatively and qualitatively with those obtained from Gabor filtering based enhancement and feature extraction. The performance of a fingerprint matching algorithm relies not only on the number of correctly detected minutiae from a fingerprint but also on the number of false minutiae. Higher number of spuriously detected minutiae cause a matching algorithm to perform worse even if the number of true minutiae are high. Keeping this in mind, we focus on comparing these values from the two algorithms. In addition to Gabor filter based enhancement and feature extraction, we have also compared our results with the original ridge tracing work by Maio and Maltoni, using our own implementation of the work. While we note that the implementation used may not be as exact as the proposed original, a major cause of the shortcomings as compared to the other results is due to the fundamental difference we noted in the tracing step.

We have evaluated the performance of our algorithm on two different datasets: one with known ground truth generated using the SFinge software, and the other one with manually marked ground truth with images coming from DB1_A database of FVC 2004. The SFinge database contains 100 fingerprints with varying noise levels, while 62 fingerprints from 31 users (two impressions from each user) were selected from FVC 2004 for manual ground truthing and comparison.

Table 1. Results of minutiae extracted from Gabor enhanced images as well as using the proposed ridge tracing method on a synthetic dataset generated using SFinGe.

| | Ground Truth | RTrace | Gabor | MMTrace |
|---|---|---|---|---|
| Endings | 1852 | 1796 (97.0%) | 1789 (96.6%) | 1378 (74.5%) |
| Bifurcations | 1674 | 1477 (88.2%) | 1593 (95.2%) | 1002 (62.8%) |
| Exchanged | 0 | 208 (5.9%) | 106 (3.0%) | 495 (14.4%) |
| Spurious | 0 | 61 (1.7%) | 100 (2.8%) | 413 (12.0%) |
| Dropped | 0 | 45 (1.3%) | 38 (1.1%) | 570 (16.5%) |

### 3.1. Qualitative Results

We first carry out a qualitative visual comparison based on typical outputs on fingerprint images. Figure 6 shows the results from a selection of typical images from both FVC and SFinGe datasets. The methods are referred to as RTrace for the proposed ridge tracing method, and MMTrace for the tracing approach proposed by Maio and Maltoni [11]. We note that the proposed approach works very well on a variety of images, while the original MMTrace approach fails in presence of noise. We also note that the proposed approach is able to overcome noisy regions where the Gabor filter output shows artifacts.

### 3.2. Quantitative Results

The results of minutiae extraction for the database from SFinge as well as FVC are given in Table 1 and Table 2 respectively. GT in the tables refer to the number of ground truth minutiae while the other rows show the number of minutiae for each corresponding row and it's equivalent percentage. For the SFinGe dataset, we are able to achieve a minutiae detection rate of 98.7%, compared to the detection rate of 98.9% for Gabor based enhancement and feature extraction, and 83.5% for the original ridge tracing approach by Maio and Maltoni [11]. In the case of natural fingerprints from noisy images in the FVC 2004 dataset, the corresponding rates are 90.4% and 94.2%, respectively. We note that the total detection rates are slightly lower in the proposed method, which is primarily due to the elimination of smaller ridge features in the post-processing step. For approach proposed by Maio and Maltoni [11], we restricted the results reported to the cleaner SFinGe dataset, and the noise in FVC dataset makes the corresponding results, significantly worse.

Figure 7(a) shows a histogram that compares the frequency of images against the number of spurious minutiae detected by the two algorithms. Figure 7(b) on the other hand shows a histogram which compares the frequency of images against the percentage of correctly detected minutiae. It is clearly visible from the results that the proposed algorithm generates 15% lesser spurious minutiae than Gabor based fingerprint image enhancement at the cost of dropping 4% more minutiae. Moreover, the histogram of spurious minutiae indicate that the proposed algorithm tends to produce more number of images with fewer spurious minutiae. The difference is more in case of noisy fingerprints such as from the FVC database. We also note that a majority of spurious minutiae in case of the proposed algorithm came from 4 fingerprint images with very high noise levels.

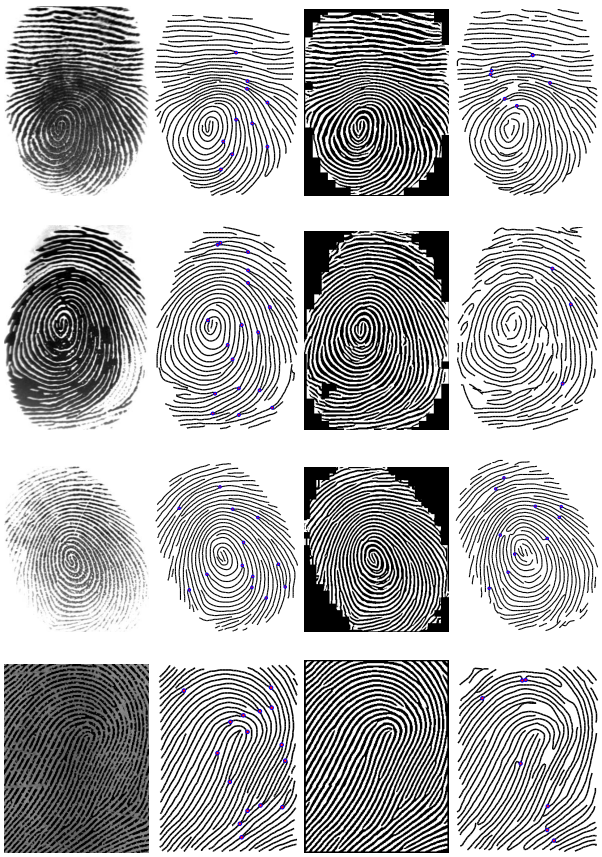
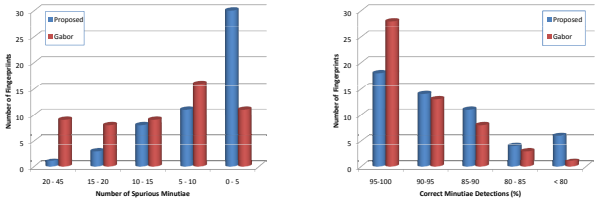



Figure 6. Feature Extraction or Restoration of fingerprints with various degradations using the Proposed approach, Gabor filtering, and MMTrace methods on sample images from FVC as well as SFinGe datasets.

Finally, we run a minutiae based matcher (NBIS [12]) on the minutiae extracted using the proposed algorithm as well as that by our implementation of Maio and Maltoni [11]. The NBIS minutiae extractor and matcher were used for Ga-

Table 2. Results of minutiae extracted from Gabor enhanced images as well as using the proposed ridge tracing method on a subset of ground truthed images from FVC 2004.

| | GT | Proposed | Gabor |
|---|---|---|---|
| Endings | 1237 | 1093 (88.4%) | 1037 (83.84%) |
| Bifurcations | 925 | 555 (60.0%) | 719 (77.7%) |
| Exchanged | 0 | 300 (13.9%) | 288 (13.3%) |
| Spurious | 0 | 344 (15.0%) | 678 (31.4%) |
| Dropped | 0 | 214 (9.9%) | 118 (5.5%) |




(a) Spurious Minutiae (b) Detected Minutiae

Figure 7. Number of spurious and correct minutiae per fingerprint between Gabor based and the proposed algorithm.

bor filtered images. As we note, the ROC curve 8 based on Gabor enhancement starts higher due to its higher rate of minutiae detection, while the proposed approach performs better at lower FAR due to fewer false minutiae. Note that the implementation of Maio and Maltoni used here includes a Gabor filter stage before tracing.

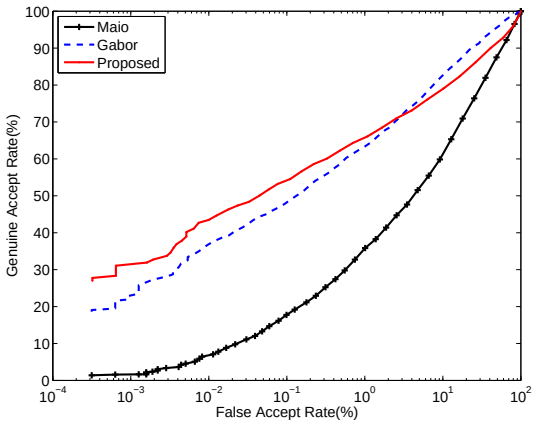



Figure 8. ROC Curve using FVC 2004 DB1_A dataset.

## 4. Conclusions and Future Work

We have presented a robust and efficient method for extraction of minutiae and ridge features directly from gray scale fingerprint images. A robust mechanism for determining the tracing step allows us to deal with considerable noise in the fingerprint images. We are able to achieve significantly lower number of spurious minutiae detection as compared to Gabor based enhancement and feature extraction, although at the cost of a slight increase in dropped minutiae.

In future, we would like to improve the algorithm with an intelligent post-processing step to reduce the number of type-exchanged minutiae as well as dropped features such as short ridges and spurs. Use of mixed spectrum technique can improve the accuracy of ridge frequency estimation [8].

## References


[1] S. Bernard, N. Boujemaa, D. Vitale, and C. Bricot. Fingerprint segmentation using the phase of multiscale gabor wavelets. In *Proceedings of Asian Conference of Computer Vision*, Jan. 2002. 2

[2] J. Feng, F. Su, and A. Cai. Robust ridge following in fingerprints. In *Proc. Sinobiometrics*, pages 424–431, Guangzhou, China, Dec. 2004. 2

[3] F. Galton. *Finger Prints*. Macmillan, London, 1892. 1

[4] S. Greenberg, M. Aladjem, D. Kogan, and I. Dimitrov. Fingerprint image enhancement using filtering techniques. In *Proc. Intenational Conference on Pattern Recognition*, pages 326–329, Barcelona, Spain, Sept. 2000. 2

[5] L. Hong, Y. Wan, and A. K. Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, Aug. 1998. 2

[6] M. Isard and A. Blake. Condensation: Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, Jan. 1998. 2

[7] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–314, Apr. 1997. 2

[8] X. Jiang. Fingerprint image ridge frequency estimation by higher order spectrum. In *Proc. International Conference on Image Processing*, pages 462–465, Vancouver, Canada, Sept. 2000. 8

[9] T. Kamei and M. Mizoguchi. Image filter design for fingerprint enhancement. In *Proc. Intl. Symposium on Computer Vision*, pages 109–114, Florida, USA, Nov. 1995. 2

[10] R. Ma, Y. Qi, C. Zhang, and J. Wang. Fingerprint ridge line extraction based on tracing and directional feedback. In *Proc. International Conference on Computational Intelligence and Security*, volume 1, pages 1033–1038, Xi'an, China, Dec. 2005. 2

[11] D. Maio and D. Maltoni. Direct gray-scale minutiae detection in fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):27–40, Jan. 1997. 1, 2, 3, 7

[12] NIST. Nist biometric image software. http://www.nist.gov/itl/iad/ig/nbis.cfm. 7

[13] J. Yang, L. Liu, T. Jiang, and Y. Fan. A modified gabor filter design method for fingerprint image enhancement. *Pattern Recognition Letters*, 24(1):1805–1817, 2003. 2

[14] E. Zhu, J. Yin, and G. Zhang. Fingerprint enhancement using circular gabor filter. In *International Conference on Image Analysis and Recognition*, Sept. 2004. 2