CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142
ISSN 2072-6287
Versión Digital

REPORTE TÉCNICO
Reconocimiento de Patrones

SERIE AZUL

Fingerprint Minutiae Extraction: A Survey

Katy Castillo Rosado,
Armando Rodríguez Fonte,
and
José Hernández Palancar

RT_055 noviembre 2013

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2142
ISSN 2072-6287
Versión Digital

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

REPORTE TÉCNICO
Reconocimiento de Patrones

SERIE AZUL

Fingerprint Minutiae Extraction: A Survey

Katy Castillo Rosado,
Armando Rodríguez Fonte,
and
José Hernández Palancar

RT_055 **noviembre 2013**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Table of Content

List of Figures

List of Tables

Fingerprint Minutiae Extraction: A Survey

Katy Castillo Rosado, Armando Rodríguez Fonte, and José Hernández Palancar

Dpto. Biometría, Centro de Aplicaciones de Tecnologías de Avanzada(CENATAV),
La Habana, Cuba
{kcastillo,afonte,jpalancar}@cenatav.co.cu

RT_055, Serie Azul, CENATAV
Aceptado: 18 de septiembre de 2013

Resumen. Las huellas dactilares son uno de los rasgos biométricos más comúnmente usados para la identificación de las personas. Esto se debe a que se conoce que son únicos e inmutables. La mayoría de los sistemas existentes se basan en la extracción de rasgos conocidos como minucias, presentes en las huellas. Debido a que las imágenes de huellas dactilares suelen degradarse es necesario en muchos casos, antes de la extracción de rasgos, pasar por un proceso de mejoramiento de la imagen. A continuación se presenta un resumen de algunas técnicas encontradas en las diferentes etapas del proceso de extracción de minucias y su análisis. Se exponen las carácterísticas fundamentales de dos algoritmos de extracción de minucias y se comparan sus resultados. A partir del análisis realizado se definen las líneas de investigación en las que se va a seguir el estudio.
Palabras clave: huella dactilar, calidad, mejoramiento, extracción de minucias, terminación, bifurcación, reconstrucción de huellas dactilares.

Abstract. The fingerprint is one of the most commonly used biometric characteristics for identification of individuals. This is because it is known that they are unique and immutable. Most existing systems are based on minutiae extraction. Because fingerprint images are degraded is necessary in many cases, before feature extraction, an enhancement image step is needed. Below is a summary of some techniques found for the different stages of feature extraction process and their analysis. Two minutiae extraction algorithms are described and their results are compared. From the analysis the investigation areas to future work are defined.
Keywords: fingerprint images, quality, NFIS (NIST Fingerprint Image Software), enhancement, minutiae extraction, ridge endings, ridge bifurcation, fingerprint reconstruction.

1 Introduction

Body measurements have been one way for solving crimes through history. In 1893, Home Ministry Office of United Kingdom accepted that the fingerprints are unique for each individuals. After that discover, many departments of the law saw the potential of the fingerprint in the identification of the offenders that used an alias for each arrest and to avoid the harshest penalties reserved for recidivists in law. Those departments began to stored the criminal's fingerprints, that is the way to have available registers for future arrests. This possibility allows to the authorities collect latents from crime scenes and compare with the previously stored fingerprints. A lot of researchers have worked for improving the efficiency and efficacy of the algorithms to compare fingerprints, but the demands in this field are growing constantly. These demands were the cause to the development of Automatics Fingerprint Identification System (AFIS). Law enforcement agencies were the earliest adopters of the automatic fingerprint recognition technology. More recently, however, increasing concerns about security and identity fraud have created a growing need for fingerprints recognition and other biometric technologies for person recognition in a large number of non-forensic applications.

Fingerprints are completely formed at the seven month of fetus development. The configuration of a fingerprint does not change during the life of an individual except due to accidents such as bruises and cuts on the fingers [26]. This property becomes fingerprints in a very exactly biometric identifier. Physical appearance and fingerprints are part of an individual phenotype. The process of fingerprint formation is similar to growth of capillaries and blood vessels in angiogenesis. There are many variations during formation of fingerprint, it is virtually impossible to have two fingerprints exactly equal.

Fingerprints are considerated a synonymous of individuality, but this affirmation is not probed. The generalized use of the fingerprints provokes some worries about the individuality of fingerprints. If this characteristic were wrong many consequences would take place, because fingerprints are used for recognizing citizens with efficiency and reliability in the protection against fraud of identity. Also AFIS does not use all the information of the fingerprint, but only a representation of this after the application of an algorithm for automatic features extraction.

The rest of this report is organized as follow: the section 2 makes a review of differents techniques at each step inmerse in the minutia extraction process (estimation of fingerprint image quality, fingerprint enhancement, fingerprint reconstruction, fingerprint binarization, minutiae extraction, removing false minutiae), the sections 3 and 4 present an explanation of the algorithms Verifinger and Nbis, respectively. The section 5 shows a comparison between the algorithms discussed above. In the final section the conclusions of the entire work are exposed.

2 State of the Art of Feature Extraction Steps

This chapter conducted a review of the state of art of various processes involved in improving images and latent fingerprints. It covers the following topics: image quality, image enhancement, estimation of the orientation, rebuilding of fingerprints, reconstruction of ridges, detection and removal of false minutiae and fingerprints binarization.

The images of fingerprints can be classified into three categories, namely: rolled, flat and latent. The rolled images are obtained by rolling a finger back and forth in order to capture all the details of the ridges of the finger. The flat impressions are those in which the finger is pressed against a flat surface. Rolled and flat impressions are obtained by scanning inked impressions on paper or by using scanners in live. These types of printing are obtained in a controlled environment with good quality and a large information content. In contrast, latent fingerprints are lifted from surfaces of objects that were inadvertently touched or handled by a person. These latent fingerprints are taken by different techniques ranging from a simple photograph printing, to complex chemical processes. That is why the processes of search and verification of latent fingerprints against a database is very important to catch a criminal. The extraction process is applied to the rolled and flat fingerprints stored into the database and to the latent prints, although the latter is not performed in an automated manner. In the present context, the features are essentially the information that is used for search and verification of a candidate. The original fingerprints are the source for getting these features.

Corresponding to the complexity with which they are obtained, the features are traditionally divided into three hierarchical levels [27]:

1. Level 1 is basically defined by ridge flow classification and singular points (cores and deltas). These features have been widely used in latent matching.

2. Level 2 is determined by features obtained from ridges, the points known as minutiae, the kind of these and relations between ridges and minutiae.
3. Level 3 comprises the features obtained from ridges specific details such as: dots, pores, protrusions, etc. These traits begin to receive more attention now because of its proven contribution to increased efficiency in comparing latent fingerprints.

2.1 Fingerprint Quality Estimation

Obtain the image quality is an important step to avoid process an image that is not a fingerprint or to find out how much reliable is the information obtained from the fingerprint image. Different approach has been used, some of this ones are presented in this section.

Elham Tabassi et al. [24] present an approach to calculate the image quality using a neural network. Fingerprint image quality is defined as a predictor of a matcher performance. They say that the good quality fingerprints must be those with high match scores and well separated from the non-match distribution, and the poor quality fingerprints are those with lower match scores, in particular those where their match scores are in the region of overlap with non-match scores.

Their implementation uses a neural network [28] as the classifier. 5 levels of quality are defined: poor (5), fair (4), good (3), very good (2), and excellent (1), because is the correct for a medium to good quality matcher. NIST [29] shows that 8 to 10 levels would be needed to fully characterize a matcher that is very sensitive to image quality. Matchers that are insensitive to image quality could be well characterized by three levels of image quality.

MINDTCT (package of NFIS which has a fingerprint minutia detector algorithm) is used to generate a quality map (explained in Section 4). This algorithms give a quality value for each extracted minutia, and the number of minutiae of quality 0.5 or better, 0.6 or better, 0.75 or better, 0.8 and better, and 0.9 and better are computed. Their feature vectors are defined using the quality map and minutia quality gived by NFIS, as shown in table 1.

Table 1. NFIQ feature vectors. Taken from [24]

	Name	Description
1	foreground	number of blocks that are quality 1 or better; i.e. $foreground = \sum_{i=1}^{4} U_i$ where U_i is the number of blocks with quality i
2	total # of minutiaes	number of total minutiae found in the fingerprint
3	min05	number of minutiae that have quality 0.5 or better
4	min06	number of minutiae that have quality 0.6 or better
5	min075	number of minutiae that have quality 0.75 or better
6	min08	number of minutiae that have quality 0.8 or better
7	min09	number of minutiae that have quality 0.9 or better
8	quality zone 1	percentage of the foreground blocks of quality map with $quality = 1$
9	quality zone 2	percentage of the foreground blocks of quality map with $quality = 2$
10	quality zone 3	percentage of the foreground blocks of quality map with $quality = 3$
11	quality zone 4	percentage of the foreground blocks of quality map with $quality = 4$

They explain why they state the problem as a classification problem. An artificial neural network is chosen as the nonlinear classification method. The neural network has the capability of acting as an approximation function for an arbitrary nonlinear function and is not dependent on model based distribution functions in either the feature or the classification space. The theory behind the machine learning techniques used by them is explained in [28].

The output of the neural network are the 5 classes of quality number. The input is a $11 - dimension$ feature vector. They chose 22 hidden nodes. The activation functions used for hidden and output nodes are sinusoid. Boltzmann pruning [30] was set, i.e. dynamic removal of connections is performed during trainning.

Its training set consists of plain right index, plain left index, plain right thumb, and plain left thumb fingerprints from datasets DOS-C, DHS2-C, DHS10, TXDPS, and BEN. These datasets contain different quality levels because the fingerprints were collected by different personnel, at different locales, and under different conditions.

The tests to the algorithm were performed on VISIT_POE and VISIT_POE_BVA datasets and a subset of DOS-C, DHS2-C, DHS10, TXDPS, and BEN that was not used for training. They concluded that the method is highly accurate even for matcher and data combinations that were not used in the neural network training because of the test results of the US-VISIT POE data presented.

Tai Pang et al. [1] used to calculate the quality score of the entire image the ridge and valley clarity by using overlapping regions of the distributions and the quality of orientation flow.

A threshold value DT_1 is calculated, and the regions lower than DT_1 are ridges, otherwise are valleys.

The figure 1 shows the gray level distribution of the segmented ridge and valley. The overlapping area is the region of misclassification. The area of the overlapping region can be an indicator of the clarity of ridge and valley.

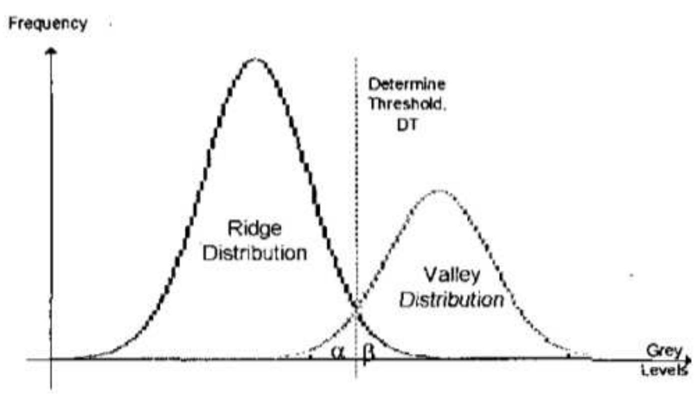


Fig. 1. Distribution of ridge and valley. Taken from [1]

The Local Clarity Score (LCS) is a ridge and valley clarity indicator, which is calculated by the equation 1.

$$LCS = \frac{\alpha + \beta}{2} , \tag{1}$$

$$\alpha = \frac{V_B}{V_T} , \tag{2}$$

$$\beta = \frac{R_B}{R_T} , \tag{3}$$

where V_B is the number of bad pixels in the valley that the intensity is lower than the DT_1, V_T is the total number of pixels in the valley region, R_B is the number of bad pixels in the ridge that the intensity is higher than the DT_1, R_T is the total number of pixels in the ridge region. α and β are the portion of bad pixels. The Local Clarity Score (LCS) is the average value of α and β.

From their experiments they conclude the classifications of the quality of the ridge pattern presented in the table 2.

Table 2. Meaning of clarity score. [1]

Clarity Score	Quality
$LCS < 0.15$	Good quality ridge pattern.
$0.15 \leq LCS < 0.35$	Intermediate quality with noise.
$0.35 \leq LCS < 0.55$	Marginal quality with noise.
$LCS \geq 0.55$	Bad quality ridge pattern.

The Global Clarity Score (GCS) can be used to describe the general ridge clarity of a given fingerprint image and it is computed with the expected values of the Local Clarity Scores (LCS) as shown in the equation 4.

$$GCS = E(LCS(i,j)), where E(\bullet) = \frac{\sum_{i=1}^{H}\sum_{j=1}^{V}(\bullet)}{H * V} , \tag{4}$$

where $LCS(i,j)$ is the local clarity score at location (i,j), i and j are horizontal and vertical index of the image block respectively, H and V are the maximum number of horizontal and vertical block respectively.

Then is calculated the Global Orientation Quality Score (GOQS) which is computed with the average of all the Local Orientation Quality Score (LOQS) using the global orientation flow. GCS and LOQS are used to calculate the Overall Image Quality (OIQ) using the equation 5.

$$OIQ = \varpi_1 * (1 - GCS) + \varpi_2 * (1 - GOQS) , \tag{5}$$

$$\varpi_1 + \varpi_2 = 1 , \tag{6}$$

where ϖ_1 and ϖ_2 are weights for GCS and GOQS respectively and the sum of both should be equal to 1. They express that the OIQ can be used to describe the quality of a fingerprint image.

The proposed algorithm is compared with the human expected score and can conclude that it is reliable to justify the image quality for AFIS. Some results are presented in the figure 2.

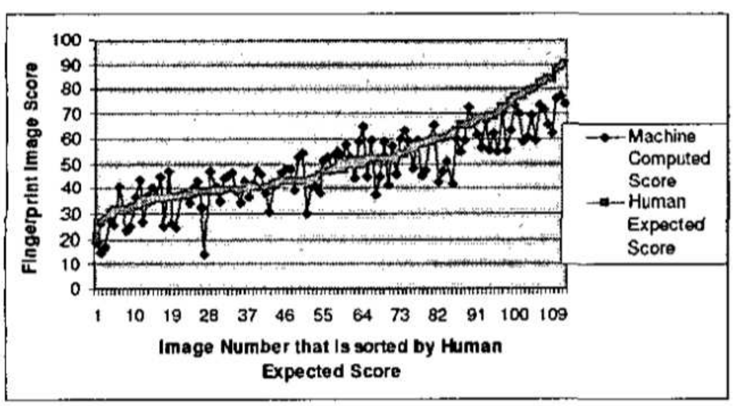


Fig. 2. Score Trend of Overall Quality Score versus Sorted Quality Images by Human Expected Score. Taken from [1]

Yi Chen et al. [31] presented two quality indices for fingerprint images: the energy concentration in the frequency domain as a global feature (Q_f) and the spatial coherence in local regions (Q_s). The image enhancement, the minutiae extraction and the fingerprint matching performance are predicted. They conclude that the Q_f has better predictive capabilities at the image enhancement stage than Q_s, at the feature extraction Q_s is slightly more effective than Q_f, and both indices are effective predicting and improving the matching performance.

Fernando Alonso-Fernandez et al. [32] made a review of differents approaches for fingerprint image-quality estimation. The image quality estimation is divided into 3 major classes: those that use local features of the image; those that use global features of the image; and those that address the problem of quality assessment as a classification problem. A summary of each of one of these classes has been exposed.

The algorithms based on local features usually divide the image into nonoverlapped square blocks and extract features from each block. Then the blocks are classified in groups of different quality. At the final step a local measure of quality is obtained. It can be the percentage of blocks classified with ”high” or ”low” quality, or an elaborated combination. Some methods assume that the blocks near the centroid of the fingerprint can provide more reliable information [31], [33] and those methods assign a relative weight to each block based on this distance. This methods can be based on local direction, gabor filters, pixel intensity, power spectrum, or can be the combination of differents local features.

The methods based on global features analyze the image in a holistic manner and compute a global measure of quality based on the features extracted. The direction field and the power spectrum are global features used to estimate the fingerprint quality in this kind of methods.

Methods using classifiers [24], [34] define the quality measure as a degree of separation between the match and nonmatch distributions of a given fingerprint. The quality measure is seen as a prediction of the matcher behavior. They extract the fingerprint features (minutiae) and calculate the quality of each extracted feature to estimate the quality of the fingerprint image.

Min Wu et al. [35] investigated about the specifications to evaluate the image quality such as the size of the foreground area, the dryness and the wetness of the fingerprint, the orientation coherence of the foreground, the captured image position and the penalty score of the background. With all these measures the final score of the whole fingerprint image is calculated $SCORE = W_1 * Score_1 + W_2 * Score_2 + W_3 * Score_3 + W_4 * Score_4 - W_5 * Score_5$. W_i is a weight for each specification. The image quality is compared with the match score and usually the higher quality image is been associated with the higher match score.

Muhammad Umer Munir et al. [36] used a hierarchical k-means clustering to divide the fingerprint images into four classes: good, wet, dry and normal. Two quality features in the frequency domain and six statistical features in spatial domain are calculated. Using those quality features the fingerprint images are classified. They compare the performance of the minutiae based fingerprint matching system with and without the fingerprint quality classifier incorporated. Their experiments show that the FAR (false accept rate) has been reduced from 1.8 to 0.79, the FRR (false reject rate) was the same in both cases.

For finding patterns in data of high dimension is common use a statistical technique named Principal Component Analysis (PCA). Xunqiang Tao et al. [37] proposed a method based on PCA for fingerprint quality measure. A circular manifold topology in a high-dimensional space is used, which is more regular when the quality is higher. The characterization of manifold topology that is been used represent the local properties of the fingerprint and reflects the local quality of the fingerprint. They extract two features from the residual variances and the manifold topology. The first feature is the two leading eigenvalues of the PCA projection which correspond to the variance along the first two principal components. The first feature (f_1) is defined based on that when the quality is higher then the sum of the first two principal component residuals are higher. The second feature (f_2) represents the differences between the manifold topology and a circle. Then the global image quality measure is calculated incorporating for this the normalized Harris-corner strength (HCS) as weighted value into local block quality. To evaluate the method the 2004DB1 (FVC2004) competition database and their private database (AES2501) are used. They evaluate the influence of their method on matching performance, using VeriFinger 6.1 SDK which is an algorithm based on minutiae. The experimental results confirm that the method is effective finding the fingerprint quality.

Soweon Yoon et al. [38] propose a latent fingerprint image quality. To each latent is assigned one of the following tags: value for individualization (VID), value for exclusion only (VEO), and no value (NV). The VID and VEO were considered for comparison. An AFIS declare that a latent fingerprint is VID if its mated print is found in the top rank 100. The features that can be used to estimate the latent fingerprint quality are separated in quality for value and quality for identification. The quality for value (also knew like qualitative quality) measures the adequacy of the Level-1 features such as fingerprint ridge clarity, pattern class, size of region of interest, fingerprint position (center, side, or tip of a finger), this features determines the utility of the latent for either exclusion or identification. The quality for identification measures the adequacy of Level-2 and Level-3 features (minutiae and pores), and determines how reliable are the matching results by an AFIS in the ”lights-out” mode. An AFIS in the ”lights-out” mode refers to a system that its input only requires the fingerprint images and returns a short list of reference prints as potential mates. They believe that a latent fingerprint quality measure should be dependent of the AFIS. They call this a matcher-dependent latent quality measure. They define a latent quality measure which is the combination of the average ridge clarity in the convex hull enclosing all the minutiae (a qualitative quality feature) and the number of minutiae (a quantitative quality feature), and use it to estimate the wanted quality measure: the probability that the mated print of the latent fingerprint is found in the top rank 100. After evaluated the latent quality measure with the NIST SD27 and WVU latent database they concluded that their latent quality measure reject in an effective way the poor quality latents.

2.1.1 Table of Quality Algorithms

A summary of the studied algorithms is presented in the table 3.

Table 3. Different exposed enhancement algorithms

Algorithm Reference	Technique	Year
Elham Tabassi et al. [24]	Neural Network. They use an $11 - dimension$ features vector.	2004
Tai Pang et al. [1]	Local Clarity Score (LCS) and Local Orientation Quality Score (LOQS).	2004
Yi Chen et al. [31]	Energy concentration in the frequency domain as a global feature (Q_f) and the spatial coherence in local regions (Q_s)	2005
Fernando Alonso-Fernandez et al. [32]	Review	2007
Min Wu et al. [35]	Size of the foreground area, the dryness and the wetness of the fingerprint, the orientation coherence of the foreground, the captured image position and the penalty score of the background.	2011
Muhammad Umer Munir et al. [36]	Hierarchical k-means clustering. They calculate two quality features in the frequency domain and six statistical features in spatial domain.	2012
Xunqiang Tao et al. [37]	The first feature is the two leading eigenvalues of the PCA projection. The second feature represents the differences between the manifold topology and a circle.	2012
Soweon Yoon et al. [38]	The latent quality measure is the combination of the average ridge clarity in the convex hull enclosing all the minutiae and the number of minutiae.	2012

2.1.2 Partial Conclusion and Main Weaknesses Detected

The feasibility of the algorithms for estimating the fingerprint quality, is more or less high depending on the parameters that has been measured in this process. These parameters should be related to the features fingerprint that are used in the various processing applied to the image.

The principal existent problems in the estimated quality algorithms are:

1. Deciding what fingerprint features define the fingerprint image quality.
2. What extent each of the fingerprint features will influence in image quality?

2.2 Fingerprint Enhancement

An important step at the fingerprint processing is the images enhancement. In the literature exist several approaches that perform this step. Among the techniques more used to images enhancement are the Fourier Transform, Gabor filters, Wavelet transform, Anisotropic filters. Some of them are shown below.

2.2.1 Gabor Filter Technique

Lin Hong et al. [2] use a bank of even-symmetric Gabor filters to enhanced the image. The region of interest for a fingerprint image is divided into three categories:

- Well-defined region, in which ridges and valleys are clearly visible for a minutia extraction algorithm to operate reliably.
- Recoverable corrupted region, in which an enhancement algorithm can still correctly recovered the corrupted ridges and valleys.
- Unrecoverable corrupted region, in which is impossible to recover the corrupted ridges and valleys by the severe noise and distortion.

They refer to the first and second categories as recoverable and the last category as unrecoverable. The main objective of a reasonable enhancement algorithm is to get better the clarity of ridge and valley structures of fingerprint images in recoverable regions and to discard the unrecoverable regions. Another very important consideration for a fingerprint enhancement algorithm is that it should not create any spurious ridge and valley structures. The estimation of the local ridge/valley orientations is an important problem for an enhancement algorithm, because several techniques use the local information of the ridge and valley structures to improve the fingerprint image quality. These techniques assume that the local orientation can be reliably estimated, but this assumptions is not true for fingerprint images of poor quality. The purpose of the enhancement algorithm is to improve the ridge/valley clarity in recoverable regions and to label the region where it does not have the capability of find the true ridge/valley structures as unrecoverable regions. Its principal steps are:

1. A bank of even-symmetric Gabor filters is applied to an input fingerprint image and a set of filtered images is produced.
2. A ridge extraction algorithm is applied to each of the filtered images and the corresponding ridge map is obtained.
3. From the extracted ridge maps of filtered images, a voting algorithm is used to generate a coarse-level ridge map and unrecoverable-region mask. The generated coarse-level ridge map is used for orientation field estimation.
4. An orientation estimation algorithm is applied to the generated coarse-level ridge map, and the local orientation at each pixel is obtained.

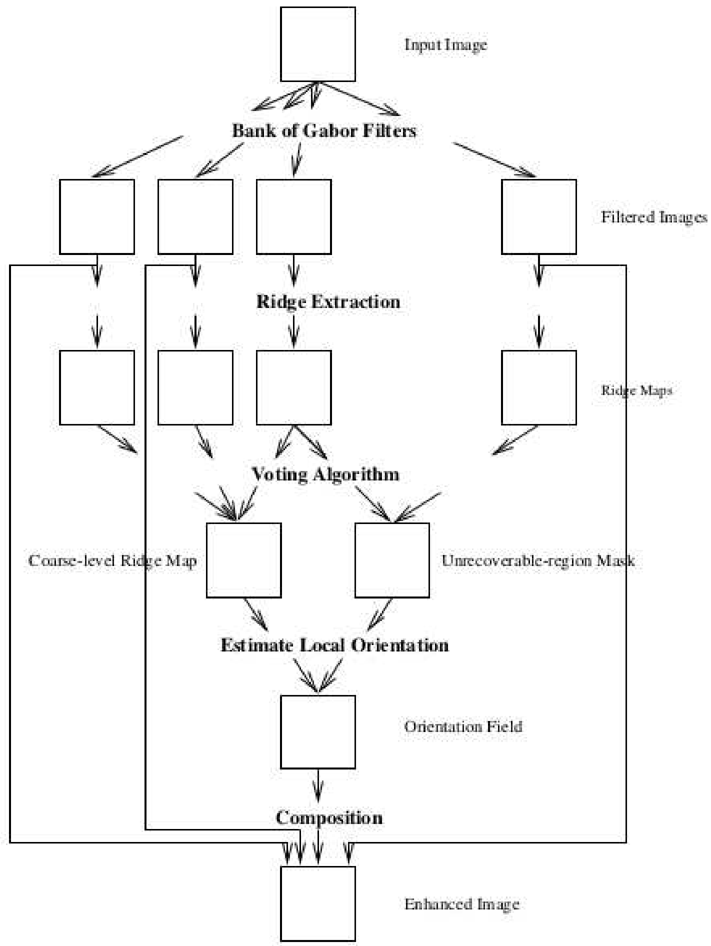


Fig. 3. An overview of the algorithm. Taken from [2]

5. From the computed orientation field and filtered images, an enhanced image is obtained.

An overview of their algorithm is shown in the figure 3. It has two principal steps: 1) orientation field estimation, 2) enhancement. To obtain a better performance of image enhancement stage, the orientation field is estimated from the filtered image to get a more reliable field orientation estimation.

The selection of filter parameters is important in applying Gabor filters. They select $60\ cycles/width$ $(height)$ as the central frequency and $2.5\ octaves$ as the radial bandwidth. They use eight values of central orientation: $0°$, $22.5°$, $45°$, $67.5°$, $90°$, $112.5°$, $135°$, $157.5°$. They select $35°$ as orientation bandwidth. Eight filtered images are obtained applying these 8 Gabor filters to the input fingerprint image.

First a FFT is performed on the fingerprint image. Then to obtain the filtered image, the corresponding Gabor filter with tuned radial and orientation frequency is applied to the frequency image and an inverse FFT is performed. The ridge map is extracted to each filtered image. The next step is to generate coarse-level ridge map and mask of unrecoverable regions. The orientation field is reliably estimated from the coarse-level ridge map by ignoring the unrecoverable regions, because it preserves the local orientation information of the ridge/valley structures of the input fingerprint image. In the postprocessing step the spurious minutiae are detected, because it can be expected that an enhancement algorithm works ideally in all situations. An example of the algorithm performance is shown in the figure 4.

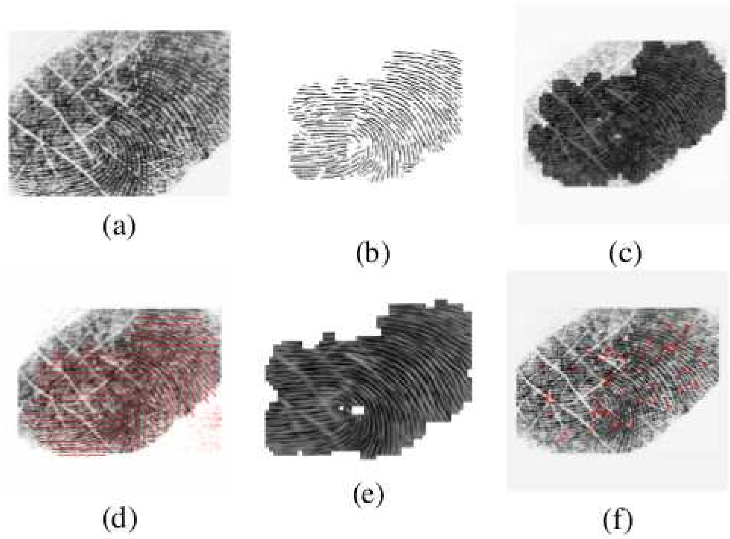


Fig. 4. Perform of the algorithm. Taken from [2]

Two experiments are executed using the MSU fingerprint database. This one have 670 image from 67 individuals. The image size is 640×480. The image quality vary, more than the 90% been captured with satisfactory quality, and the 10% of the images are not of good quality.

The first experiment consist in directly match each image in the database against the others fingerprint image in the database. In the second experiment, the verification is performed after apply their fingerprint enhancement algorithm to each fingerprint image in the database.

Table 4. Recognition and reject rates with different threshold values on the matching score. Taken from [2]

Threshold Value	Recognition Rate	Reject Rate Enhanced	Recognition Rate Enhanced	Reject Rate
20	99.42%	11.23%	99.25%	7.37%
22	99.86%	14.56%	99.95%	9.66%
24	99.89%	16.78%	99.99%	11.07%
26	99.96%	20.20%	100%	14.84%
28	99.98%	23.15%	100%	16.28%
30	99.99%	27.45%	100%	18.21%

In the table 4 is presented the recognition rates and reject rates with different threshold values on the matching score. This experimental results shown that the performance of the verification system has been improved when their algorithm is applied to the input fingerprint images. The major benefit of the enhancement algorithm is that the reject rate is substantially reduced while is keeping essentially the same recognition rate.

En Zhu et al. [39] use a Gabor filter to enhance the image, but the gabor frequency is adjusted to the average frequency of the input image, change its shape from square to circle and dynamically adjusting the filters size based on the average frequency.

The average frequency is applied on the entire image to avoid the blocky effect. A circular Gabor filter is used to evade the directional effect. When the fingerprint images average inter-ridge changes, the size-fixed filter cannot work well. In their experiments the Gabor filter work well when its size is thrice the width of ridge. Therefore they apply the Gabor filter with this size.

Let I denote the gray-level fingerprint image of size $m \times n$, where $I(x, y)(0 \leq x < m, 0 \leq y < n)$ is the intensity of the pixel at the x_{th} row and y_{th} column. They divide I into nonoverlapping blocks of size $w \times w$ (15×15). Each block is denoted as $W(i, j)$ where (i, j) is the location of W. The center coordination of $W(i, j)$ is $(i \times w + w/2, j \times w + w/2)$. Let $M(i, j)$ mark the foreground/background of I. If $M(i, j) = 1$, $W(i, j)$ is a foreground block, else background. ${O^I}_{W(i,j)}$ is defined as the orientation of $W(i, j)$.

Their enhancement process has the following steps:

1. Orientation estimation: compute the orientation of each block.
2. Image segmentation: compute M to mark the background/foreground.
3. Average frequency estimation: estimate each foreground blocks frequency and compute their average value.
4. Enhancement: filter the input image using circular Gabor filter with its frequency tuned to the average frequency computed in the step 3.

Raymond Thai [40] use an even-symmetric Gabor filter in the spatial domain; he decided to use the values of the filter bandwidth to be a function of the ridge frequency parameter.

Li Wang et al. [3] proposed to use a Gaussian filter at first place, then processed the image with their proposed adaptive algorithm (power-law transformation and contrast stretching), and finally the image is filtered with Gabor filter.

The figure 5 shown the principal step of the proposed algorithm in [3].

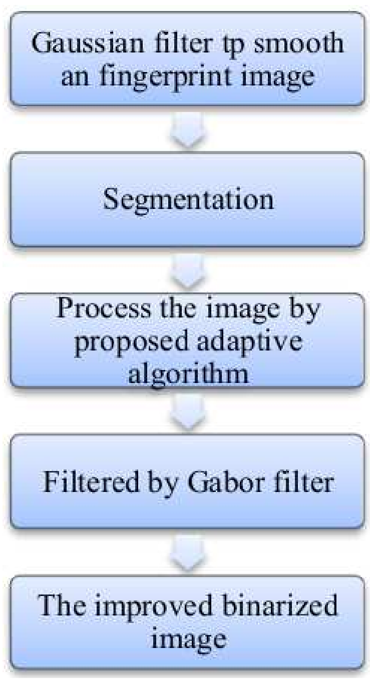


Fig. 5. Algorithm description. Taken from [3]

First, a Gaussian filter is used before their proposed approach to smooth the small breaks existing on ridges because their approach may enlarge those small breaks. A 3×3 Gaussian mask is used. For the second step the gray scale variance and mean intensities of a fingerprint image are calculated, introduced by [41], [42] to generate its mask. In first place the fingerprint image is divided into 5×5 pixel blocks.

For each block the mean intensity and the standard deviation (or variance) (STD) are calculated. For a $m \times n$ block B, the mean intensity M is calculated by the equation 7 and the $STD(V)$ can be calculated by the equation 8.

$$M = \frac{1}{mn} * \sum_{i=1}^{m}\sum_{j=1}^{n} B_{ij} \ , \tag{7}$$

$$V = \sqrt{\frac{1}{mn} * \sum_{i=1}^{m}\sum_{j=1}^{n} (B_{ij} - M)^2} \ . \tag{8}$$

Then the mask can be generated as follows:

$$Mask = \begin{cases} 1 \text{ if } V > T_1 and M > T_2 \\ 0 \text{ otrherwise } \ . \end{cases} \tag{9}$$

T_1 and T_2 are thresholds selected by experimental observation. The selection of T_1 and T_2 may be adjusted according to different databases to get the optimized results. They choose $T_1 = 0.75 * (mean\ STD\ of\ all\ blocks)$ and $T_2 = (mean\ intensity\ of\ the\ whole\ image) + (mean\ STD\ of\ all\ blocks)$.

The third step is adjust the contrast and brightness in digital image processing. The contrast stretching and power-law transformation have been used for this [41]. They build an algorithm based on two approaches to reduce the noise and enhance fingerprint images.

A gray level fingerprint image histogram indicates the distribution of gray levels. When they analyse the histogram of the fingerprint images they divided the gray level into 3 parts. The first part can be relate to as gray levels of ridges, the second part as the gray levels of edges of ridges and noise, and the last part is the gray levels of valleys and background.

Based on the above analysis, the equation constructed to improve fingerprint images is the next one:

$$s = \begin{cases} \frac{rs_1}{r_1} & \text{if } r < r_1 \\ \frac{s_2 - s_1}{r_2{}^{\gamma} - r_1{}^{\gamma}} r^{\gamma} + \frac{s_1 r_2{}^{\gamma} - s_2 r_1{}^{\gamma}}{r_2{}^{\gamma} - r_1{}^{\gamma}} & \text{if } r_1 < r < r_2 \\ L - 1 & \text{if } r_2 < r < L - 1 \ . \end{cases} \tag{10}$$

Their algorithm is similar to contrast stretching but they use non-linear transformation in the middle part. They said that is because contrast stretching can improve the contrast of a fingerprint image, but it does not change the data information except intensity values [41]. When contrast stretching is applying information of noise can remain. If γ is selected according to the noise level the performance of power-law transformation reducing the noise level can be better than linear transformation.

Exist some parameters undefined in the equations 9 and 10. The best choice of these parameters could be different for different blocks. The table 5 shows the estimation of the parameters based on the experiments of some sets of fingerprint database from Fingerprint Verification Competition 2002 (FVC2002) [43].

At the fourth step they chose Gabor filter because it can fix small breaks on ridges. It can remove undesired noises and smooth the ridge and valleys structure. The equation for $2 - D$ even symmetric Gabor filter are as follows [27], [44]:

$$G(x, y; \theta, f) = exp \frac{x_{\theta}{}^2}{\sigma_x{}^2} + \frac{y_{\theta}{}^2}{\sigma_y{}^2} cos(2\pi f x_{\theta}) \ , \tag{11}$$

Table 5. Estimation of parameters of equation 9. Taken from [3]

Parameter	Rules of estimation
r_2 selection	if $M > 200$, $r_2 = M + V$; else if $150 < M \leq 200$, $r_2 = M + \frac{V}{2}$; else if $128 < M \leq 150$, $r_2 = M$; else if $M \leq 128$, $r_2 = M - \frac{V}{2}$;
r_1 and s_1 selection	$r_1 = max(M - V, 25)$; $s_1 = r_1$;
γ selection	if $M \geq 200$, $\gamma = 1.2$; else if $R \geq 0.7$, $\gamma = 1$; else if $R \leq 0.5$, $\gamma = 0.5$; else if $0.5 < R < 0.7$ and $M > 150$, $\gamma = 1.2$; else if $0.5 < R < 0.7$ and $100 \leq M \leq 150$, $\gamma = 0.8$; else if $0.5 < R < 0.7$ and $M < 100$, $\gamma = 0.6$;

$$x_\theta = xsin(\theta) + ycos(\theta) \ , \tag{12}$$

$$y_\theta = -xcos(\theta) + ysin(\theta) \ , \tag{13}$$

where f and θ denote the frequency and orientation of local ridges respectively, whereas σ_x and σ_y represent the standard deviation of Gaussian envelope along x and y axes respectively, and they are both set to be 4.0 based on empirical data [45]. Therefore, before using Gabor filters, two parameters that should be estimated are local ridge orientation and frequency.

To estimate the local ridge orientation and frequency, they choose the approach proposed by Kass and Witkin [46] and Bazen and Gerez [47]. The image is divided into 16×16 blocks then is calculated the average orientation and frequency of each block as input to the Gabor filter. They implement a Gabor filter bank of 60 filters with angle increment of 3. Then each block is convoluted with the Gabor filter with closed angle.

They use the database set Db3 of FVC2002 to carry out the experiments. They use goodness index (GI) [31] to evaluate the performance of their proposed approach. The GI is defined in [31] as follows:

$$GI = \frac{p}{t} + \frac{a+b}{u} \ , \tag{14}$$

where p represents the total number of paired minutiae and t represents the ground truth minutiae in a given fingerprint image, a, b and u are the number of missed minutiae, spurious minutiae and total number of detected minutiae respectively.

The GI is calculated applying their approach to 20 fingerprints in DB3 of FVC2002 which contain good and bad quality images. The comparison between their method with the traditional approach is presented in the table 6. It shows that their approach can detect minutiae more accurate than traditional Gabor filter based approach and improve the performance by 9% on average.

They compare their proposed approach with other algorithms from the literature, and the results are shown in the table 7.

2.2.2 *Fourier Transform in the Frecuency Domain Technique*

Sharat Chikkerur et al. [4] proposed a fingerprint enhancement based on Short Time Fourier Transform (STFT), and make the evaluation of the enhancement algorithm by considering the improvement in matching accuracy for poor quality prints. They compare the proposed algorithm with the matching without

Table 6. Goodness Index comparisson. Taken from [3]

Image	Goodness Index Traditional Approach based on Gabor Filter	Goodness Index Proposed Approach
1	0.60	0.60
2	0.65	0.65
3	0.75	0.67
4	0.29	0.50
5	0.72	0.79
6	0.71	0.71
7	0.79	0.69
8	0.56	0.48
9	0.74	0.72
10	0.60	0.75
11	0.50	0.68
12	0.43	0.83
13	0.42	0.68
14	0.55	0.84
15	0.67	0.72
16	0.49	0.59
17	0.69	0.72
18	0.72	0.78
19	0.40	0.70
20	0.74	0.72
STD	0.14	0.09
Average	0.60	0.69

Table 7. Comparisson between some approach. Taken from [3]

Methods	Min GI	Max GI	Average GI
Hong, Wan, Jain [45]	0.29	0.55	0.39
Zhao and Tang [21]	0.18	0.75	0.50
Simon-Zorita et al [48]	0.33	0.76	0.55
Lee and Bhattacharjee [49]	0.31	0.75	0.55
Proposed [3]	0.48	0.84	0.69

enhancement and with a proposed algorithm by Hong et al. [45], and their method obtained the best performance.

They chose to use the angular measure proposed by Rao [50]. They explain that this is more robust to errors in the orientation estimation and does not require them to compute the singular point locations. They point out that the results presented in [50] also indicate that while the algorithm is able to eliminate most of the false minutiae, it also misses more number of genuine minutiae when compared to other existing algorithms.

They present a fingerprint image enhancement algorithm based on contextual filtering in the Fourier domain. They say that their algorithm can obtain the local ridge orientation and ridge frequency information using short time Fourier Analysis at the same time. Their algorithm can also successfully segment the fingerprint images. They announce some advantages of their proposed approach:

1. The proposed approach obviates the need for multiple algorithms to compute the intrinsic images and replaces it with a single unified approach.
2. This is also a more formal approach for analysing the non-stationary fingerprint image than the local/windowed processing found in literature.

3. The algorithm simultaneously computes the orientation image, frequency image and the region mask as a result of the short time Fourier analysis. In most of the existing algorithms the frequency image and the region mask depend critically on the accuracy of the orientation estimation.
4. The estimate is probabilistic and does not suffer from outliers unlike most maximal response approaches found in literature.
5. The algorithm utilized complete contextual information including instantaneous frequency, orientation and even orientation coherence/reliability.

The figure 6 shows the overview of their approach. The image is divided into overlapping windows in the STFT analysis. They assume that the image is stationary within this small window and can be modeled approximately as a surface wave. The ridge frequency and ridge orientation are estimated probabilistically analyzing the fourier spectrum of this small region. The STFT analysis also provides a map energy that can be used as a region mask to separate the fingerprint and the background regions. The angular coherence is computed using the orientation image [50]. The angular bandwidth is adapted using the coherence image. The resulting contextual information is used to filter each window in the fourier domain. The enhanced image is obtained by tiling the result of each analysis window.

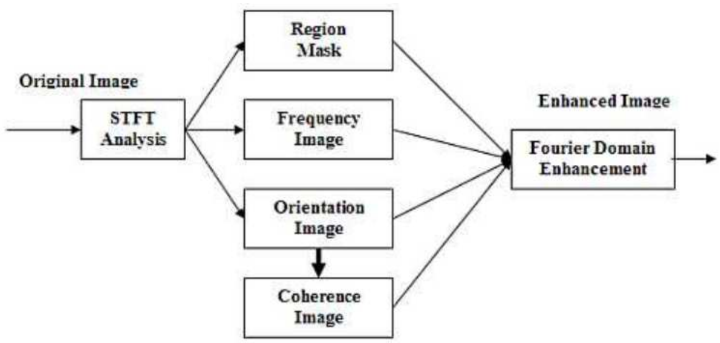


Fig. 6. Overview of the algorithm. Taken from [4]

The algorithm for enhancement is presented in the figure 7. The algorithm presents two stages. The first one consist of STFT analysis and the second one performs the contextual filtering. The ridge orientation image, ridge frequency image and the block energy image are obtained in the first stage. Then is computed the region mask using the block energy image. In this stage are obtained all the information needed to perform full contextual filtering. The filter is separable in angular and frequency domains (equation 15) and is identical to the filters mentioned in [51]. They also adapt the radial bandwidth for each block to cover two octaves around the central frequency ρ_0.

$$H(\rho,\phi) = H_\rho(\rho)H_\phi(\phi)\ . \tag{15}$$

They present some visual effects on fingerprint images at each step of the algorithm. They evaluate the algorithm performance on a set of 800 images (100 users, 8 images each) derived from FVC2002 [43] DB3 database.

They made 2800 and 4950 of genuine and impostor comparison respectively. They used NIST's NFIS2 open source software (http://fingerprint.nist.gov) for the sake of feature extraction and matching. The summary of the results is provided in 8.

Prawit Sutthiwichaiporn et al. [5] presented an iterative enhancement algorithm that is composed by 4 main steps; preprocessing, SNR (signal-to-noise ratio) analysis and quality assessment, matched filtering, and iterative restoration. The second step is performed with the Fast Fourier Transform (FFT).

```
Algorithm:  FFTEnhance
Inputs    :  Image I(x,y)
Outputs   :  Enhanced Image I'(x,y), Ridge Orientation Image O(x,y),
             Ridge Frequency Image F(x,y), Energy Image E(x,y),
             Orientation Coherence Image C(x,y), Region Mask(x,y)
STAGE I: STFT Analysis
1. For each overlapping block B(x,y) in the image
      a. Remove DC content of B, B=B-avg(B)
      b. Multiply by spectral window W
      c. Obtain the FFT of the block, F = FFT(B)
      d. Perform root filtering on F
      e. Perform STFT Analysis. The analysis yields values of
         E(x,y),O(x,y),F(x,y)
   end for
2. Smoothen orientation map O(x,y) by vector averaging to yield O'(x,y)
3. Perform isotropic diffusion on frequency map F(x,y) to yield F'(x,y)
4. Compute coherence image C(x,y) using O'(x,y)
5. Compute region mask R(x,y) by thresholding E(x,y)

STAGE II: Enhancement
6. For each overlapping block B(x,y) in the image
      a. Compute angular filter FA centered around O(x,y) and with
         bandwidth inversely proportional to C(x,y)
      b. Compute radial filter FR centered around frequency F(x,y).
      c. Filter the block in the FFT domain, F= F*FR*FA
      d. Compute the enhanced block B'(x,y) = IFFT(F)
   end for
7. Reconstruct the enhanced image by composing enhanced blocks B'(x,y)
```

Fig. 7. Overview of the algorithm. Taken from [4]

Table 8. Summary of the performance results over FVC2002 DB3. Taken from [4]

Database	Metric Without Enhancement	Hong et al. [45]	Proposed [4]	
DB3	EER	10.35%	7.8%	7.8%
	FMR100	19.50%	13.0%	15.0%

The overview of the algorithm presented in [5] is shown in the figure 8.

In their experiment they use a Gaussian low-pass filter with 7 window size to smooth the texture area, at the preprocessing step.

In the SNR analysis and quality assessment step they use the Short Time Fourier Transform (STFT) because it is an efficient tool to explore texture information of fingerprint in both spatial and frequency domains. The fingerprint texture area is partitioned into 32×32 blocks in spatial domain. Then each block is extended to a 64×64 overlaped block, each block is transformed using Fast Fourier transforms (FFT). As the extended area is big enough to avoid blocking effect from discontinuous boundary of FFT blocks, no windowing technique is used. Besides, without windowing technique, spectra diffusion can be achieved in the later stage.

To separate high quality zones and low quality zones, they use the concept of signal-to-noise ratio.To a certain threshold Th_{SNR} ($Th_{SNR} = 1$ in this experiment), the spectrum in each block may be separated into signal and noise. The $SNR_{B(i,j)}$ (Signal-to-Noise) of the block $B(i,j)$ can be defined by:

$$SNR_{B(i,j)} = \frac{\sum |F_q(u,v)|^2 F_q(u,v) > Th_{SNR}; \forall u, vinB(i,j)}{\sum |F_q(u,v)|^2 F_q(u,v) \leq Th_{SNR}; \forall u, vinB(i,j)} \; , \tag{16}$$

where $F_q(u,v)$ is a normalized and quantized Fourier coefficient at the location (u,v) of a block $B(i,j)$. The SNR average, $\overline{SNR}$, can be found by:

$$\overline{SNR} = \frac{1}{N_{B \in ROI}} \sum_{\forall i,j \in ROI} SNR_{B(i,j)} \; , \tag{17}$$

where $N_{B \in ROI}$ is the number of blocks in ROI.

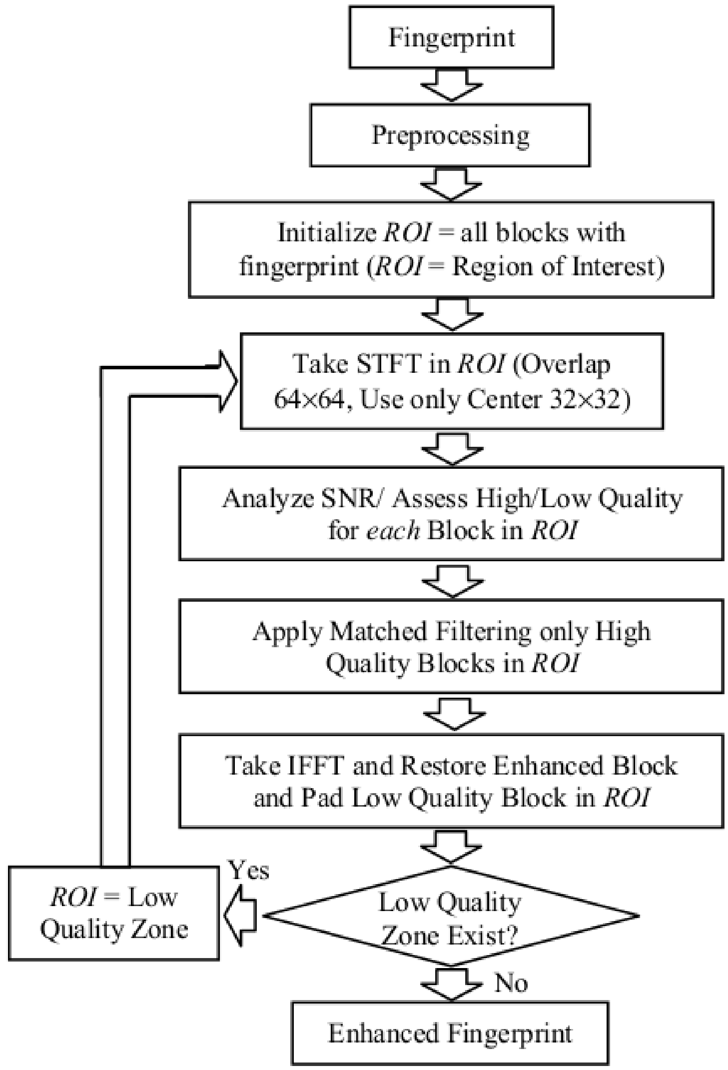


Fig. 8. Overview of the algorithm. Taken from [5]

To classify the block quality (high/low) they use as the threshold the SNR average, $\overline{SNR}$. If $SNR_{B(i,j)}$ is greater than SNR, this block $B(i,j)$ qualifies as a high quality block in this ROI region, and it will be filtered and removed from ROI in this stage. Otherwise, it will be defined as a low quality block, and it will remain in the ROI region for the next stage.

At the matched filtering stage, they obtain the filter shape for each high quality block, by capturing all spectra above the threshold level Th_{SNR}, and by setting all signals below Th_{SNR}, to be zero. This spectrum shape is smoothed with Gaussian window size 5×5 pixels. This smoothed spectrum is used like a matched filter with noise suppression in frequency domain. The resulted filters are arbitrary shapes depending on power spectra of signal above the threshold level Th_{SNR}. Then is multiplied the original Fourier coefficients of each block with its corresponding matched filter to performing the filtering. The figure 9 shows examples of the matched filter's shape for each particular block.

At the iterative restoration step, after all high quality blocks are filtered in frequency domain, inverse Fourier transform is applied to recover enhanced blocks in spatial domain. Then these blocks are padded

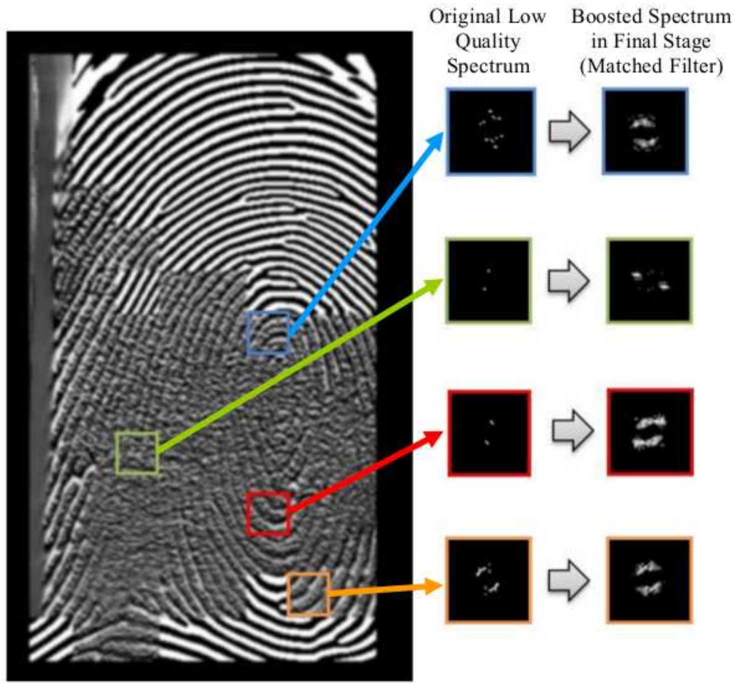


Fig. 9. Examples of power spectra above Th_{SNR} (This equivalent to a matched filter with noise suppression) for original stage and later iteration stage. Taken from [5]

with the low quality blocks resulting in partial enhanced fingerprint image. The remains of low quality blocks are formed the new ROI, and then ROI's boundary is smoothed by 7×7 Gaussian low-pass filter to remove blocking artifacts. Next all blocks in ROI are fed back to repeat the process by taking FFT again. The algorithm is iterated until no low quality zone or ROI exists.

They made the experiments with the FVC2004 Db2 and Db3. They compared three differents enhancement methods: their method, Hong's method (Gabor) [45] and Chikkerur's method (STFT) [4]. They use their own fingerprint matching algorithm by using minutiae and ridges to performance evaluation. The table 9 reports the matching results.

Table 9. Matching Performance Comparison. Taken from [5]

Enhancement Algorithm	EER (%) FVC2004 Db2	EER (%) FVC2004 Db3
Gabor [45] (16×16 block)	7.66	5.43
STFT [4] (16×16 block)	8.39	5.86
Proposed [5] (32×32 block)	7.32	3.57

They explain that their method performs better on FVC2004 Db3 than Db2 because most fingerprints in Db3 contain combination of high quality zone and low quality zone in the same image. They say that their method is not effective if an entire fingerprint image contains the same low quality zone such as some fingerprints in FVC2004 Db2. They explain that in that case, SNR is low and matched filters may introduce noises and propagate them.

Soweon Yoon et al. [52] proposed a latent fingerprint enhancement algorithm, which expects that the region of interest (ROI) and the singular points have been manually marked. They compute the orientations using the short-time Fourier transform (STFT). The orientation field is estimated using R-RANSAC (hypothesize-and-test). Then they enhanced the fingerprint image with Gabor filters [45].

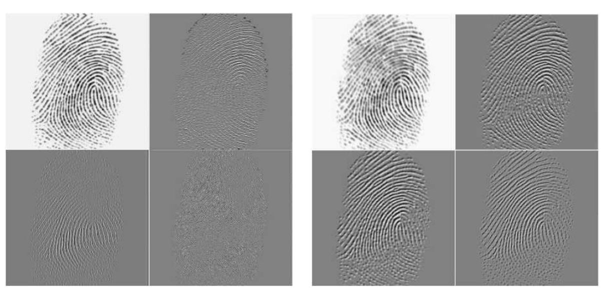
2.2.3 Wavelet Transform in the Frecuency Domain Technique

Jiajia Lei et al. [6] obtained an anhancement algorithm wich combines nontensor product wavelet filter banks and anisotropic filter. Their algorithm has 5 principal steps: normalization, decomposition (with nontensor product wavelet filter), anisotropic filtering, detail subimages adjustment and reconstruction. They concluded that their method avoids block efects and not only enhances well the ridge structures in nonsingular regions, but also accurately preserves the ridge structures in singular regions.

The two principal problems that affect most of the fingerprint enhancement methods (contextual filter-based methods) are: block effect on the enhanced images and they blur or destroy ridge structures around singular points. They develope an anhancement fingerprint method combining nontensor product wavelet filter banks and anisotropic filter to well preserve the ridge structures in singular regions and avoid block effect. They first decompose the fingerprint image with the nontensor product wavelet filter banks, with this the fingerprint image can be decomposed efficiently and can obtain wavelet coefficients. Then, the approximation subimage is modified using anisotropic filtering and the high frequency coefficients of the three other subimages are adjusted using the adaptive approach to reduce the noises and to increase the contrast between ridges and valleys according to the geometry feature of images. At the end, the inverse transform is applied to map the result and a final contrast enhancement is done subsequently.

Tensor wavelets present many advantages, as noted by Baraniuk et al. [53], but they have four drawbacks: (1) oscillations, (2) shift variance, (3) aliasing and (4) lack of directionality. Nontensor product wavelets have received much attention in the literature [54], [55],[56] because reveal the singularities in all directions. Usually, $2-D$ tensor wavelet filter banks are the tensor product of $1-D$ wavelet filter banks. Although it has been shown that $1-D$ wavelet filter banks is compact supported, its tensor product only can find the singularities in the three directions (i.e. horizontal, vertical and diagonal) [57]. Though a fingerprint image contains the singularities in all directions, tensor wavelets can not find them. In fact, the high-frequency subbands of nontensor product WT can expose more features than tensor wavelet, which is normally used. Many mathematicians are studing the discrete nontensor product WT (DNWT) [55] for the need of theory and applications [54], [56].

In the figures 10(b) and 10(a) is shown the improvement of revealing singularities achieved by nontensor product wavelet filter banks in contrast with tensor wavelets respectively. They express that the nontensor product wavelet filter banks can capture singularities in all directions while the traditional tensor wavelet is constraint to the three directions.



(a) (b)

Fig. 10. Wavelet decomposition by tensor wavelet and by nontensor product wavelet filter banks. Taken from [6]

They proposed a fingerprint enhancement algorithm based on the nontensor product wavelet filter banks. Their algorithm have five major steps, which are ilustrated in the figure 11.

The algorithm five major steps are:

1. Normalization

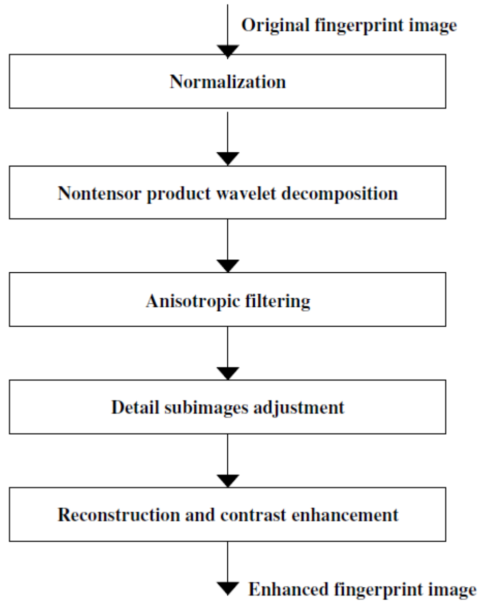


Fig. 11. The flowchart of [6]. Taken from [6]

2. Decomposition
3. Anisotropic filtering
4. Detail subimages adjustment
5. Reconstruction

The reduction of the variations in the gray-level values of the fingerprint image is the purpose of their first step. They normalize using the equation 18, where $I(x, y)$ denote the gray-level value of the original image, M and VAR denote the estimated mean and variance of the original image, respectively. M_0 and VAR_0 represents the desired mean and variance of the normalized fingerprint image I_N.

$$I_N = \begin{cases} M_0 + \sqrt{\frac{VAR_0(I(x,y)-M)^2}{VAR}} \text{ if } I(x,y) < M \\ M_0 - \sqrt{\frac{VAR_0(I(x,y)-M)^2}{VAR}} \text{ otrherwise } . \end{cases} \tag{18}$$

At the second step the normalized fingerprint image I_N is decomposed into a multi-resolution representation using the nontensor product wavelet filter banks. They choose N depending of the image size, and N pairs of parameters (α_k, β_k); $k = 1, .., N$ (in their experiment N is 1). They explain that, in theory, the image can be decomposed into subimages at any level. They select one or two levels for the purpose of the ridge structure to fit the filtering and speed the computation. Then they use nontensor product filter banks to decompose the image. At each level four parts are extracted: A (approximation subimage), H (horizontal detail subimage), V (vertical detail subimage) and D (diag detail subimage).

They estimate the orientation field using an anisotropic diffusion process, and use the estimated result as contextual information for anisotropic filters. The corresponding approximation subimage obtained by nontensor product wavelet lose some information of the ridges, because of that, they down-sample the original image to get a new subimage A_n. Then, the new approximation subimage A_n is enhanced by anisotropic filters.

They calculate the orientation field using gradient-based method with the equations 19, 20 and 21,

$$G_{yy} = \sum_{\Omega} 2G_x(u,v)G_y(u,v) \ , \tag{19}$$

$$G_{yy} = \sum_{\Omega} G^2{}_x(u,v)G^2{}_y(u,v) \ , \tag{20}$$

$$\theta(x,y) = \frac{\pi}{2} + \frac{1}{2} arctan \frac{G_{yy}}{G_{xx}} \ , \tag{21}$$

where Ω is a small neighborhood region of the point, and $arctan$ denotes a fourquadrant inverse tangent function whose output is within the range of $[-\pi, \pi]$. Then the orientation field is smoothed using an anisotropic diffusion process.

They use an anisotropic filter to establish the bandpass filters and enhance the true ridge structure. The general form of the filter kernel is exposed in the equation 22:

$$h(x,y,\theta,f) = z_1 + z_2 exp(-\frac{x^2{}_\theta}{2\sigma^2{}_1} - \frac{y^2{}_\theta}{2\sigma^2{}_2}) \frac{sinfx_\theta}{fx_\theta} \ , \tag{22}$$

$$x_\theta = xcos(\theta) + ysin(\theta) \ , \tag{23}$$

$$y_\theta = -xsin(\theta) + ycos(\theta) \ , \tag{24}$$

where z_1, z_2, σ_1, σ_2 are empirical parameters, and they set $z_1 = -1$, $z_2 = 2$, $\sigma_1 = 2$, $\sigma_2 = \sqrt{2}$ in their algorithm. Their enhanced approximation subimage A_e is the follow:

$$A_e(x,y) = \sum_{\Omega} h(u,v,\theta,f) A_n(x-u, y-v) \ . \tag{25}$$

In the fourth step they perform two procedures for each of the three detailed subimages: (1) Normalization and (2) Adjustment, to adjust them by a specific mapping function in order to remove noise.

At the fifth step, they can reconstruct the final enhanced fingerprint image. Normally, they use to recontruct the same nontensor product filter banks which are used in teh decomposition procedure. Then the final contrast enhancement is done subsequently.

The images that they use in their experiments are taken from the FVC2004 fingerprint image databases. They compare their method with the Gabor filter-based method [45] and a frequency domain method, that is STFT [4] because of their performance. They present some images and the results are that both methods [45],[4] fail to accurately enhance the ridge structures in the singular regions. This method [6] preserves the ridge structures in singular regions and avoid block effect in the enhanced image.

2.2.4 Anisotropic Filter Technique

Shlomo Greenberg et al. [7] proposed two diferents methods to image enhancement. The first one uses local histogram equalization, Wiener filtering, and image binarization. The second one uses an unique anisotropic filter for direct grayscale enhancement. They conclude that the techniques based on direct gray scale enhancement perform better than approaches which require binarization and thinning as intermediate steps.

The process of their first proposed method is shown in the figure 12.

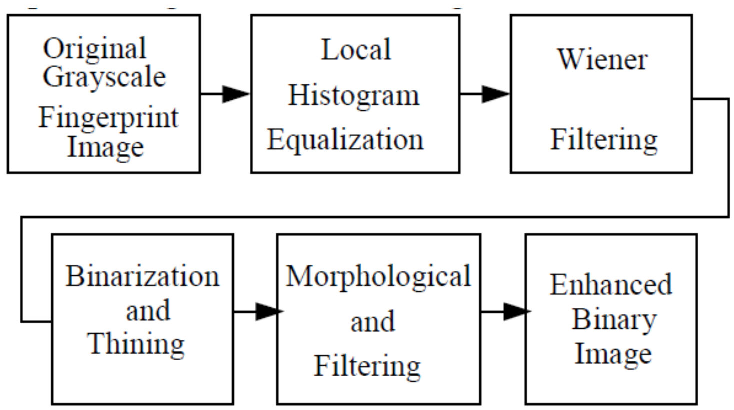


Fig. 12. The algorithm process of [7]. Taken from [7]

They apply the histogram equalization locally by using a local windows of 11×11 pixels. The goal is to expand the contrast locally, and to change the intensity of each pixel according to its local neighborhood.

They propose to use a pixel-wise adaptive Wiener's method. The filter is based on local statistics estimated from a local neighborhood η of size 3×3 of each pixel, and is given by the equation 26:

$$w(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}(I(n_1, n_2) - \mu) \ , \tag{26}$$

where v^2 is noise variance, μ and σ^2 are local mean and variance respectively, I represents the gray level intensity in $n_1, n_2 \in \eta$.

The binary image is obtained assigning to each pixel a new value (1 or 0), according to the intensity mean in a local neighborhood (13×13 pixels), as follow:

$$I_{new}(n_1, n_2) = \begin{cases} 1 \text{ if } I_{old}(n_1, n_2) \geq LocalMean \\ 0 \text{ otherwise } \ . \end{cases} \tag{27}$$

They obtain the thinned image using morphological thinning operations.

They made an algorithm to eliminate two kind of noise which appear in the thinned binary image: false ridgeline connections, and gaps within a true ridgeline. The false ridgeline connections are almost perpendicular to local ridge direction, and empirically found to be of length less than 10 pixels.

They also proposed a direct gray scale enhancement method, which has been stimulated by [45]. In first place they suggest some modification to the original algorithm [45], and then they propose a fast direct gray scale fingerprint enhancement based on an anisotropic filter.

They implement the algorithm proposed in [45] for comparison purposes with some modifications.

They modified the anisotropic filter presented in [58] by shaping the filter kernel and applied it to fingerprint images. The filter kernel applied at each point x_0 in [58] is defined as:

$$k(x_0, x) = \rho(x - x_0) exp{-}[\frac{((x - x_0)n)^2}{{\sigma_1}^2(x_0)} + \frac{((x - x_0)n_\tau)^2}{{\sigma_2}^2(x_0)}] \ , \tag{28}$$

where n and n_τ are mutually normal unit vectors, and n is parallel with the right direction. The shape of the kernel is controlled with ${\sigma_1}^2(x_0)$ and ${\sigma_2}^2(x_0)$, ρ satisfy the condition $\rho(x) = 1$ when $|x| < r$, and r is the maximum support radius. They modify the filter to a band pass filter in order to adapt it to a fingerprint image:

$$h(x_0, x) = -2 + 10k(x_0, x) \ . \tag{29}$$

When a ridgeline edge is encountered the kernel is deformed into an ellipse with a major axis aligned in parallel with the edge. Therefore, smoothing is performed along but not across the ridgeline. Applying this filter only orientation information is required. They replaced the Gabor filter in [45] with the anisotropic filter, and eliminates the need to estimate local frequency information. They concluded that the direct gray scale enhancement techniques perform better than approaches which need binarization and thinning as intermediate steps.

Chaohong Wu et al. [59] proposed an image enhanced algorithm that is based on integration of anisotropic filter and directional median filter(DMF). They said that the Gaussian-distributed noises are reduced effectively by anisotropic filter and the impulse noises are reduced efficiently by DMF. This algorithm fails when image regions are contaminated with heavy noises and orientation field can hardly be estimated.

2.2.5 Other Techniques

Qihong Ye et al. [60] used a two-dimensional empirical mode decomposition (Bidimensional Empirical Mode Decomposition BEMD) to enhanced fingerprint images. The comparison that they made with traditional algorithm (Gabor Filter) showed that their method gave better results.

Hartwig Fronthaler et al. [61] decomposed the original grayscale image into sub-bands corresponding to different spatial scales using a Laplacian-like image pyramid to enhances the image. Then on each level a contextual smoothing is performed, and the filtering directions are obtained from the frequency-adapted structure tensor (linear symmetry features). Like their enhanced method applied the enhancement progressively (blockwise operations are avoided, in the spatial domain), the enhanced image not suffer from blocking artifacts.

Jiangang Cheng and Jie Tian [62] used the scale space theory to enhance the fingerprint. They decompose the fingerprint image into a series of images at different scales. After that, whole characters and details are analyzed and organized. Finally the information is combined to enhance the fingerprint image.

Arora and Garg [25] made a comparison of various enhancement techniques (Histogram Equalization, WFT based, Gabor Filters, Log-Gabor, Fast Gabor Filters based, Gabor Filter in Wavelet Domain, Composite Method) and concluded that with composite method of spatial domain and Wavelet based enhancement the best result was obtained. The results can be observed in the table 10.

Table 10. Performance comparison of some enhancement algorithms [25]

Enhancement Technique	EER	Pick Signal To Noise	Computaion Time
Histogram Equalization	11.2%	31.036db	0.253 secs
WFT based	10.978%	35.67db	0.78 secs
Gabor Filters	9.8%	43.32db	0.928 secs
Log-Gabor	9.767%	36.461db	0.849 secs
Fast Gabor Filters based	9.89%	38.24 db	0.534 secs
Gabor filter in Wavelet Domain	9.976%	39.45db	0.707 secs
Composite Method	9.345%	41.56db	0.894 secs

Some authors use the orientation field to perform the image enhancement like Yuliang He et al. [63]. They used the 8 ridge directions of a pixel in the estimation of the orientation field. Then they made the binarization and thinning of the image.

2.2.6 Table of Enhancement Algorithms

A summary of the studied algorithms is presented in the table 11.

Table 11. Different exposed enhancement algorithms

Algorithm Reference	Enhancement Technique	Year
Lin Hong et al. [2]	even-symmetric Gabor filters	1996
Shlomo Greenberg et al. [7]	unique anisotropic filter for direct grayscale enhancement	2000
Raymond Thai [40]	even-symmetric Gabor filter in the spatial domain	2003
Yuliang He et al. [63]	Orientation field to perform the image enhancement	2003
Chaohong Wu et al. [59]	integration of anisotropic filter and directional median filter(DMF)	2004
Jiangang Cheng and Jie Tian [62]	Scale space theory to enhance the fingerprint	2004
En Zhu et al. [39]	Gabor filter	2006
Sharat Chikkerur et al. [4]	Short Time Fourier Transform (STFT)	2007
Hartwig Fronthaler et al. [61]	Laplacian-like image pyramid	2008
Li Wang et al. [3]	Gaussian filter, power-law transformation and contrast stretching and Gabor filter	2010
Sutthiwichaiporn et al. [5]	iterative enhancement algorithm and Fast Fourier Transform (FFT)	2010
Soweon Yoon et al. [52]	compute the orientations using the short-time Fourier transform (STFT), and enhance the fingerprint image with Gabor filters [45]	2011
Arora and Garg [25]	Composite method of spatial domain and Wavelet based	2011
Chopra and Upadhyay [64]	STFT or wavelet	2012
Jiajia Lei et al. [6]	combines nontensor product wavelet filter banks and anisotropic filter	2012
Qihong Ye et al. [60]	Bidimensional Empirical Mode Decomposition BEMD	2012

2.2.7 Partial Conclusion and Main Weaknesses Detected

After the above analysis we can conclude that the filters have good results in the elimination of small breaks of ridges and slight noise. When the image has a very low quality (holes and unrecoverable-regions) the effectiveness of these algorithms is significantly decrease and tend to generate false information in these areas. Therefore in such cases it is best to use reconstruction algorithms, an example of this is the latent prints.

Some of the main objectives sought by a fingerprint enhancement algorithm are:

- Increase the ridges and valleys clarity.
- No generate spurious minutiae.
- Avoid the block effect.

The principal existent problems founded in the literature are:

1. Several techniques used the local orientations to improve the fingerprint image quality, but the local orientation can not be reliably estimated for fingerprint images of poor quality.

2. In some techniques, the use of windows to scan the image produce blocking effect in the enhanced image.
3. Some algorithms blur or destroy ridge structures around singular points.

2.3 Fingerprint Reconstruction

Since minutiae template has become in a compact representation of a fingerprint, it has been assumed that it is not possible to reconstruct the original fingerprint from a minutiae template. Recently, however, this belief has been challenged by some researchers who were successful in reconstructing a fingerprint image from the given minutiae template. We may divide this problem in two aspects:

1. Firstly when you have a fingerprint with low quality due to broken ridges and you need to repair that fingerprint (this problem is called fingerprint synthesis).
2. Secondly when you have only the information about structure of minutiae and you want to reconstruct the fingerprint or latent (this problem is called fingerprint reconstruction).

Novikov and Glushenko [8] presented one model for fingerprints directional image simulation and two basic models for generating virtual ridges structure and minutiae. The figure 13 shows an example of Novikov's ridge reconstruction.

Fig. 13. Regeneration of real fingerprints by adaptive resonance method. (a) Initial image distorted by white noise and 25% of data loss. (c) 50% data loss. (b) and (d) Restored images. Taken from [8]

Araque et al. [65] proposed a paper for synthesis of fingerprint images. Global features are condensed in a linear model whose parameters are generated according to the statistical distribution of natural fingerprint patterns. Local features have been synthesized applying recursively a simple finite state filter.

Cappelli et al. [9] presented a novel approach to reconstruct fingerprint images from standard templates and investigates to what extent the reconstructed images are similar to the original ones. Reconstruction approach is based on a sequence of steps that receive the minutia template and attempt to estimate various aspects of the original unknown fingerprint (figure 14): the fingerprint area, the orientation image and the ridge pattern.

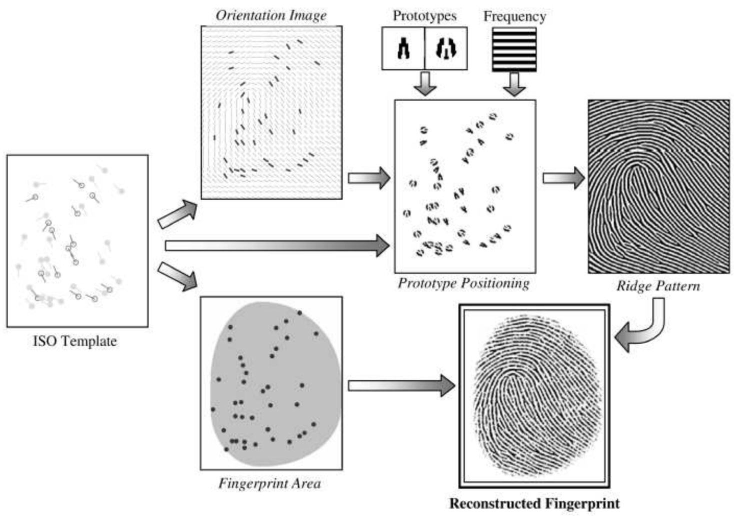


Fig. 14. Algorithm of the reconstruction approach. Taken from [9]

They reconstruct the fingerprint area using a simple mathematical model introduced in [66]. The orientation model adopted in that work was originally proposed in [67] and extended in [66] to enable the generation of synthetic orientation images. Given the minutiae set, the estimated orientation image and the frequency, the ridge pattern reconstruction involves the followings steps:

1. Minutiae prototype positioning
2. Iterative pattern growing

Step 1 is completed using the information of minutia and frequency.

Step 2 iteratively grows the minutia prototypes by applying at each pixel a Gabor filter adjust according to the frequency and the local orientation.

Ross, Shah and Jain [10] was proposed a technique that use minutiae triplet information to estimate the orientation map of the parent fingerprint. The estimated orientation map is observed to be remarkably consistent with the underlying ridge flow. The algorithm for generating the orientation map has four main stages: triplet generation, orientation prediction, triplet pruning and orientation smoothing. We have an example in figure 15.

The proposed algorithm of fingerprint reconstruction is based in the Gabor-like filter and is performance as follows:

1. An empty fingerprint image of size 512 x 512 is divided into non-overlapping blocks.
2. Each block is associated with an orientation value O(z) estimated with the algorithm. Many blocks may not have orientation information since the estimated orientation map can be incomplete.
3. The block is next initialized with a noisy blob and is convolved with the Gabor filter whose parameters are tuned using O(z). This results in a new image, which is again subjected to the convolution procedure.

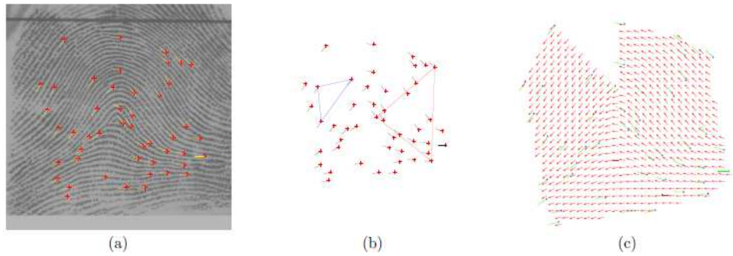


Fig. 15. (a) Minutiae distribution of a fingerprint. (b) Examples of a good quality triplet (blue) and a bad quality triplet (red). (c) Estimated orientation map. Taken from [10]

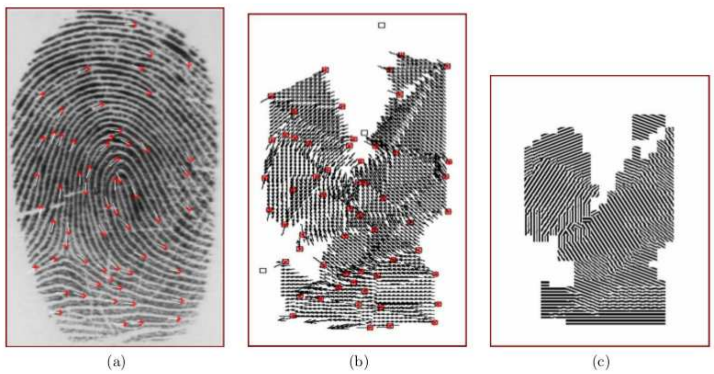


Fig. 16. Reconstructing fingerprints. (a) Minutiae distribution of a fingerprint image. (b) Predicted orientation map (c) Reconstructed fingerprint. Taken from [10]

4. This process is repeated k times resulting in an image which exhibits ridge-like patterns.

Figure 16 shows an example of fingerprint reconstruction from the estimate orientation map. This technique has a common problem: the reconstruction of the image is partial.

These researchers proposed another method for Reconstructing Fingerprints in [68]. They do the reconstruction from minutiae template. They obtain the orientation field information, the class or type information and the friction ridge structure for parent fingerprint. The orientation estimation algorithm determines the direction of local ridges using the evidence of minutiae triplets [10]. The estimated orientation field and minutiae distribution are used to predict the class of the fingerprint. Finally, the ridge structure of the parent fingerprint is generated using streamlines that are based on the estimated orientation field (see figure 17) and Line Integral Convolution (LIC) [69], this is basically a texture synthesis technique that is used to visualize 2D data. The algorithm is as follows:

1. Estimating orientation map [10].
2. Constructing streamlines using orientation map. The first action for this step is seed point selection. These points are the origin of streamlines in the orientation map. After the streamlines are constructed using linear interpolation scheme.
3. Generating ridge structure using LIC. Given a streamline the LIC technique involves calculating the intensity of all pixels constituting the streamline. It locally blurs an uncorrelated input texture image, such as white noise, along the path of the streamlines to impart a dense visualization of the flow field.
4. Enhancing the ridge map. In order to increase the ridge width, they use a lowpass filter to smooth the texture image generated using LIC and then perform histogram equalization of the ridge structure for contrast enhancement.

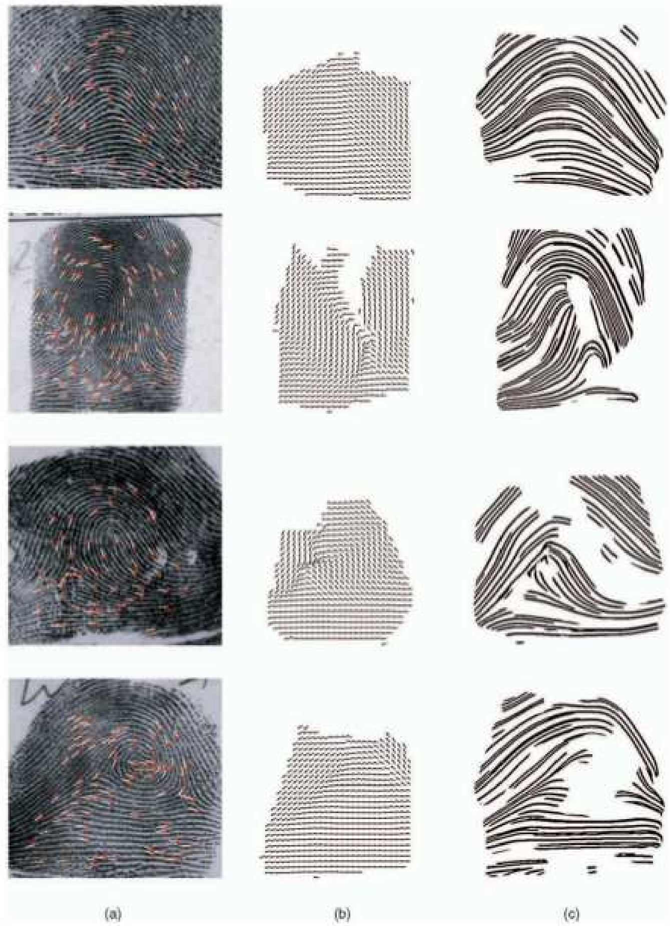


Fig. 17. Reconstructing the ridge structure. (a) Original fingerprint and its minutiae plot. (b) Estimated orientation map. (c) Enhanced ridge structure after application of the Verifinger software. Reproduced from [11]

Feng and Jain [11] proposed an algorithm to reconstruct fingerprint from minutiae template minimizing the problem of reconstruct a partial fingerprint and the another problem: many spurious minutiae not included in the original minutiae template are generated in the reconstructed template. This algorithm receives the minutiae template:

$$\{X_n, Y_n, \partial_n\}\ , \tag{30}$$

where (X_n, Y_n) and ∂_n are location and direction of minutiae respectivatly.

The following three steps are performed to obtain the reconstruct image (see figure 18):

1. Orientation field reconstruction.
2. Continuous phase reconstruction.
3. Combination of the spiral phase and the continuous phase.

The algorithm was evaluated matching original fingerprint against reconstructed fingerprint and match reconstructed fingerprint against different impressions of the original fingerprint.

Later in [12], the same authors proposed another algorithm for fingerprint recognition. In this algorithm a fingerprint image is represented as a phase image which consists of the continuous phase and the spiral phase. This algorithm minimize the problem of spurious minutiae and the reconstruction the partial image like [11].

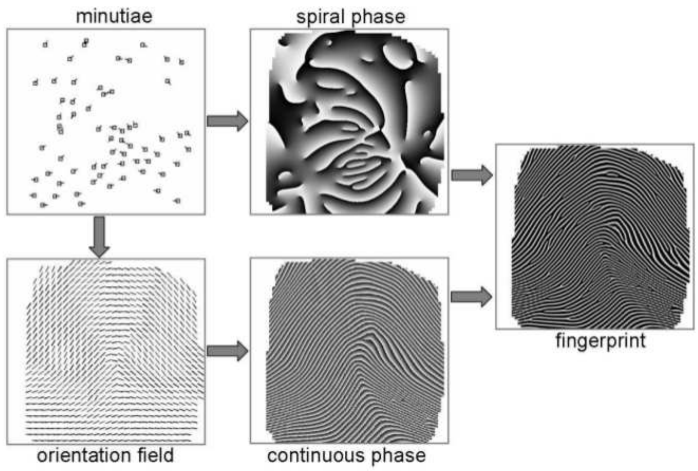


Fig. 18. Flow chart of the proposed fingerprint reconstruction algorithm. Taken from [11]

The steps of the algorithms are followings (see figure 19):

1. Orientation field reconstruction.
2. Estimation of gradient of continuous phase.
3. Continuous phase reconstruction.
4. Combination of the spiral phase and the continuous phase.

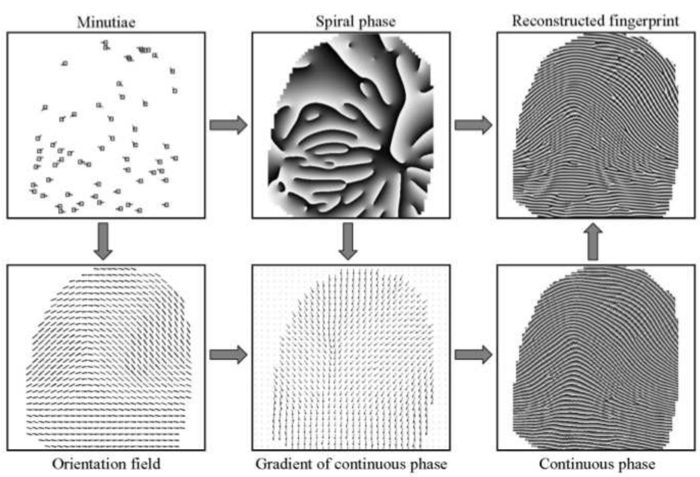


Fig. 19. Flow chart of the proposed fingerprint reconstruction algorithm. Taken from [12]

2.3.1 Table of Fingerprint Reconstruction References

Now we present a table with the most cited methods for fingerprint synthesis and reconstruction in the state of the art. These are presented in chronological order.

Table 12. Existing Fingerprint Reconstruction (R) and Synthesis (S) Methods

Method	Goal	Input	Model	year
Novikov and Glushchenko [8]	S	ridge orientation	iterative filtering	1997
Araque et al. [65]	S	ridge orientation and frequency	second-order orientation model; filtering using binary mask	2002
Hill [70]	R	singular points, minutiae	orientation model of Sherlock and Monro [71]; line drawing	2001
Ross et al. [10]	R	minutiae	minutiae triplets; gabor filter	2005
Ross et al. [68]	R	minutiae	minutiae triplets; stream lines; Line Integral Convolution	2007
Cappelli et al. [72]	R	minutiae	orientation model of [67]; Gabor filtering	2007
Feng and Jain [11]	R	minutiae	FM Model	2009
Feng and Jain [12]	R	minutiae	FM Model	2011

2.3.2 Partial Conclusion and Main Weaknesses Detected

The issue of fingerprint reconstruction has many important applications now a days. Solving the problem: reconstructing ridges structure you can enhance your fingerprint and improve the matching because recently some techniques of fingerprint matching are based in ridge structure. When an expert analyse a latent he can detect minutiae and makes the minutia template. This template may be used for reconstructing latent and improving very much the efficacy of matching algorithms.

There are two main weakness in the fingerprint reconstruction from minutiae template. They are following:

1. Usually it is not possible reconstruct a whole fingerprint image and the reconstruction is only partial: this problem was tried by [11].
2. When the fingerprint is reconstructed many spurious minutiae could be included: this problem was tried by [11] and [12].

2.4 Fingerprint Binarization

Many fingerprint recognition systems use enhancement and separation of fingerprints in a pre-processing step denoted binarization [27]. An interesting binarization category widely used in many algorithms is denoted contextual filtering.

Fingerprints are captured as grey-scale images, however many systems will binarize the image before matching. The effectiveness of a matching algorithm is heavily dependent on the quality of the binary image.

In real applications, often the images are not well balanced on the greyscale; some areas are lighter or darker than the rest of the image. This causes a lot of trouble for binarization. This effect may arise if the user applies uneven pressure on the sensor while his fingerprint is captured. Also, when a fingerprint is wet, ridges tend to be darker and thicker, whereas in dry conditions ridges will appear lighter and not as continuous [66]. There are various binarization approaches. Although rather different from one another, most of these methods transform fingerprint images into binary images through a threshold method. Typically, the threshold for binarization is selected in two general ways. The first method is carried out by finding a global threshold and then comparing each pixel of the image against that threshold. The other approach, called locally adaptive binarization, is done by computing a threshold for each pixel based on information in the neighbourhood of that pixel.

The problem of image binarization has been widely studied in the field of image processing. Otsu [73] proposed an optimum global thresholding method based on an idea of minimizing the between class variance. Moayer and Fu [74] proposed a binarization technique based on an iterative application of laplacian operator and a pair of dynamic thresholds. A similar method proposed by Xiao and Raafat [75] in which a local threshold is used after the convolution step to deal with regions with different contrast. Verma, Majumdar, and Chatterjee [76] proposed a fuzzy approach that uses adaptive threshold to preserve the same number of 1 and 0 valued pixels for each neighbourhood.

Kim and Park [13] proposed a new fingerprint binarization method based on convex threshold for effective minutiae extraction. Main processes include noise reduction, directional vector estimation, and ridge detection. For noise reduction and directional vector estimation, they use the conventional method. For ridge detection they propose a convex threshold technique using fingerprint structures. The proposed algorithm is composed of four processes: estimation of the local ridge orientation, the local ridge orientation quantization, fingerprint regularization filtering, and convex threshold (figure 20). The local ridge orientation is a local direction in a fingerprint image. The fingerprint ridge flows with uniform direction in local area and it can be used for next processes of filtering and convex threshold. To estimate the local ridge orientation, they use the modified least mean square orientation estimation algorithm [46]. The local ridge quantization divides the ridge direction into one of four quantized directions bands, because convex threshold uses only three pixels, four-direction quantization is sufficient. The fingerprint ridge structure is irregular because of noise, pressure of finger touching in image acquisition, and so on. Thus the fingerprint ridge structure regularization is necessary and important. Basic process and the filter used for this goal are similar to those in [17].

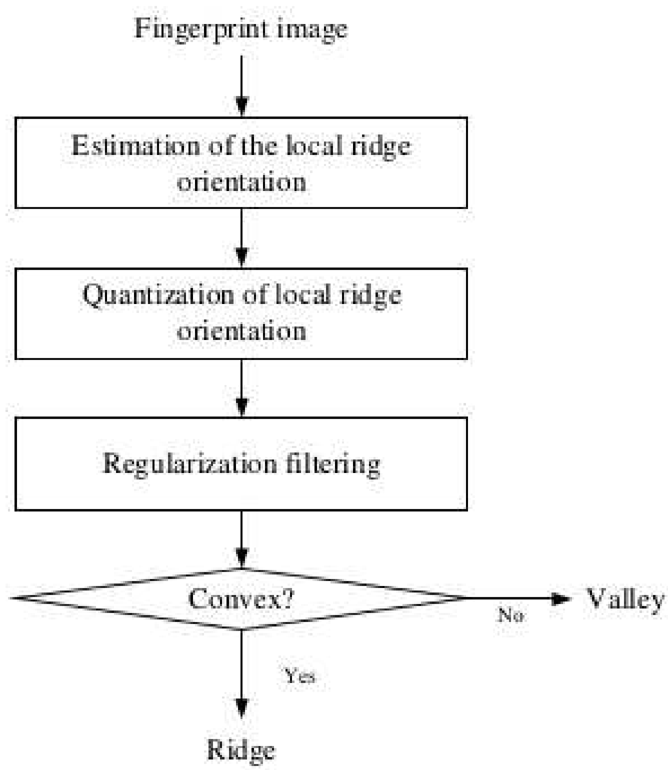


Fig. 20. Flowchart of the fingerprint binarization. Taken from [13]

Liang et al. [77] proposed a combinatorial algorithm for binarization of fingerprint images based on Euclidean distance transform. Most of the previous algorithms discussed here are heuristics in that they do not start with a definition of an optimal threshold. In contrast, they define a condition for an optimal threshold based on equal widths of ridges and valleys. They show how distance transform can be used as a measure for width and then design an algorithm to efficiently compute the threshold for binarization. They exploit the property of almost equal widths of ridges and valleys for binarization.

Zhang and Xiao [14] presented a novel approach for fingerprint binarization. The motivation was to simulate the human exercise that distinguishes ridges from valleys in uniform areas. They showed an algorithm for performing such a region generation on fingerprint images, and a data structure for saving generated regions is developed. Thresholding and binarization is then performed individually on each generated region using traditional binarization methods. This approach is particularly useful for processing fingerprint images captured with non-uniform inking or light. They said that the human approach is to first get an overall picture and then deal with the finer details. In binarization would be to find the dark images areas, detect the uniform regions, calculate a local threshold for each region, and perform the binarization with local threshold region-by-region. The finer details approach involves treating each pixel individually. Figure 21a shows a logical division of the image performed by a human vision. An algorithm has been developed to simulate the human exercise on uniform region segmentation. The processing procedures are as follows:

1. Divide the image into n pixel by n pixel blocks.
2. Calculate the mean value for each block. If the original image was p x q in size, the result of dividing the image is a p/n x q/n matrix containing the mean value of each block.
3. Perform traversal of the mean-value matrix; for each block, look for other blocks in the neighborhood that have grey-scale values within a certain threshold to form a region.
4. Perform binarization on each region.

After regions have been generated, binarization is then done on the individual regions. A threshold can then be calculated for the region using the algorithms in [73]. One setback of using the histogram is that the region generation algorithm could sometimes create regions that are one block long. These small regions could have scarce histograms and may lead to the determination of an incorrect threshold.

Josef Strom et al. [15] presented a new form to fingerprints enhancement using directionals filters and binarization. A simple method for automatic adjust of the size of local area is obtained through the analysis of fingerprint image in the frequency domain. The algorithm will adjust adaptively to the local area of the fingerprint image, independent of the characteristics of the sensor of fingerprint or the physical aspect of the fingerprint. The frequency analysis is performed in the locals areas to design directionals filters. The proposed adaptive fingerprint binarization algorithm shows a good skill to tune itself to each fingerprint image. The method used to map signals from the spatial domain into the frequency domain is the Fast Fourier Transform (FFT).

These researchers in [16] showed a new method for an entirely automated process adjusting itself to each fingerprint. Abilities such as automatically regulate size of the local area and adjustment of the directional filters demonstrates interesting adaptive behavior of the algorithm. They did an improve of the algorithm. Fingerprints usually occupy only a smaller part of the entire fingerprint image. Therefore task of fingerprint mask is to cover all non-relevant information in the image; only the valid fingerprint part is kept (figure 22).

Munshi and Mitra [78] presented a rough-set based approach for binarization of fingerprint image. Maximization of rough entropy and minimization of roughness of the image lead to an optimum threshold for binarization. Following are the steps for the binarization of image as proposed in this article:

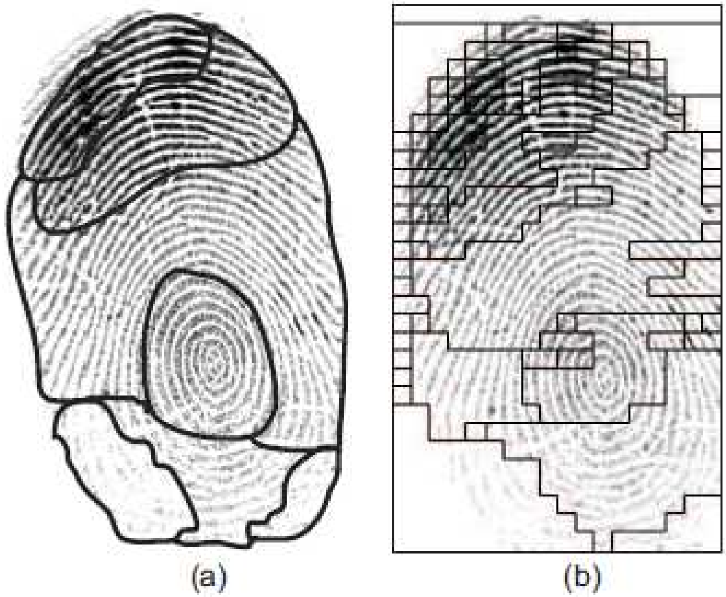


Fig. 21. An example of region generation: (a) region divided based on human vision; (b) image divided by region generation algorithm. Taken from [14]

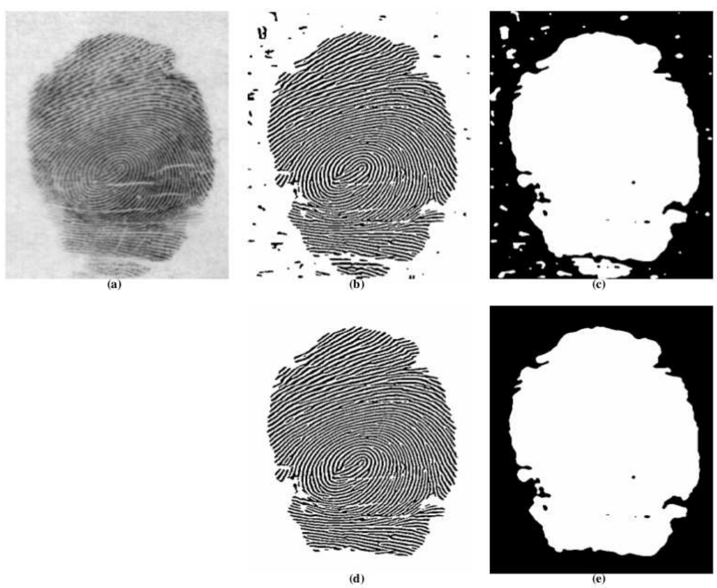


Fig. 22. (a) Original fingerprint image; (b) and (c) are the results from previously proposed algorithm [15]; (d) and (e) are the improved result by the new modified algorithm. Taken from [16]

1. Represent the image in the form of quad-tree decomposition.
2. For a threshold value T, $0 < T \leq 255$, separate the blocks obtained from decomposition into object and background.
3. Find lower and upper approximation of object and background.
4. Compute object and background roughness and hence find out rough entropy.

5. Repeat steps 2 to 4 for all values of T, i.e. from T=1 to 255.
6. The value of T for which Rough entropy is maximum, is selected as a threshold for binarization.
7. Binarize the image using optimum threshold obtained.

S. M Rajbhoj and P. B. Mane [79] proposed an algorithm for improving minutiae detection for fingerprint recognition providing accurate automatic personal identification. In their approach they have used optical sensor which captures image of excellent quality and helps to extract many minutiae in an image and hence increase the accuracy. The recognition algorithm uses histogram equalization technique to improve the global contrast of an image and a fast efficient enhancement technique before using binarization based method to extract minutiae. This overcomes the serious problem of poor quality fingerprint images.

2.4.1 Table of Binarization References

We present the table 13 with the most cited methods for binarization in the state of the art. These are presented in chronological order.

Table 13. Existing Binarization Methods

Method	Technique	Year
Otsu [73]	Global thresholding based on an idea of minimizing the between class variance	1979
Moayer and Fu [74]	Laplacian operator and a pair of dynamic thresholds	1986
Xiao and Raafat [75]	Convolution step to deal with regions with different contrast; Local threshold	1991
Verma, Majumdar, and Chatterjee [76]	Fuzzy approach that uses adaptive threshold	1987
Kim and Park [13]	Convex threshold	2003
Liang et al. [77]	Euclidean distance transform	2004
Zhang and Xiao [14]	Human appreciation; Local threshold for each region	2006
Josef Strom et al. [15]	Frequency analysis; Directionals filters	2006
Josef Strom et al. [16]	Frequency analysis; Directionals filters	2008
Munshi and Mitra [78]	Rough Set	2012
S. M Rajbhoj and P. B. Mane [79]	histogram equalization	2012

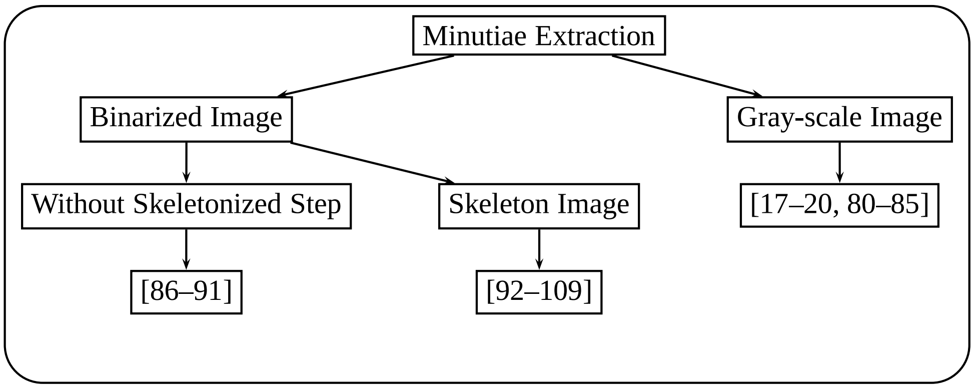
2.4.2 Partial Conclusion and Main Weaknesses Detected

Many AFIS use binarization as part of the process for feature extraction, that is why this step is very important and has been studied by a lot of researchers. The main weakness that we saw in this matter is that there is not reported an exhaustive comparison between binarization algorithms, this is important to know what techniques may be used at each moment. We propose to do a platform with several algorithms of binarization that let to do many tests in feature extraction and matching and arrive to conclusions about the performance of these.

2.5 Fingerprint Minutiae Extraction

Exist some different kinds of minutiae, but the most common are terminations and bifurcations. The step of minutiae extraction is very important. It can be turn on the binarized fingerprint image, the esqueletize fingerprint image or direct on the gray-scale fingerprint image.

Minutiae Extraction Taxonomy



2.5.1 Extraction From Binarized Fingerprint Image

In literature, the most commonly used minutiae extraction methods are based on the extraction on the skeletonized image, in which the width of each peak is reduced to a pixel in the skeletization process. The minutiae are detected looking for the terminations and bifurcations of ridges in the skeleton image, using for this the neighboring pixels. Terminations are those that have only one neighboring pixel, and bifurcations that have more than 2 pixels neighbors. Examples of these methods are [92–109].

Another used technique is to extract the minutiae from the binarized image, ignoring the step of skeletonization [86–91].

These methods are very sensitive to noise. The skeleton of the image is not created in an intuitive manner. The minutiae in the skeletonized image in many cases do not correspond with the real minutiae from the original fingerprint image. The binarization and skeletonization steps generate a lot of false minutiae, so is needed a post-processing step for its elimination.

2.5.2 Extraction Direct on the Gray-Scale Fingerprint Image

The minutiae extraction process can be done direct on the gray-scale fingerprint image [17–20, 80–85]. It is still topic of research. Extract minutiae directly from the gray-scale image have a great relevance because the following reasons:

- The binarization process may provoke the lost of a lot of information.
- Binarization and thinning are time-consuming.
- It has been prove that the binarization and skeletonization techniques are unsatisfactory with low quality images.
- The binarization and thinning may introduce a considerable number of spurious minutiae.

Some of these approach are presented below.

Dario Maio and Davide Maltoni [17] present a technique to minutiae extraction based on ridge line following direct on the gray-scale fingerprint image.

The basic idea is to follow the ridge lines on the grey scale image, by ”sailing” according to the fingerprint directional image. A set of starting points by superimposing a square-meshed grid on the gray scale image is determined. For each starting point, the algorithm follow the ridge lines until they terminate or intersect with other ridge lines (minutiae detection). They maintain a labeling strategy to examine each ridge line just one time and locate the intersections between ridge lines.

Fig. 23. Example of the algorithm performance. Taken from [17]

An example of the performance of the algorithm is shown in the figure 23.

The results shown the superiority of their technique over the previous approaches (convencional thresholding and thinning approaches), in terms of eficiency and robustness.

Dario Maio and Davide Maltoni [81] also present an approach to minutiae filtering based on Neural Network using the algorithm exposed in [17].

Jinxiang Liu et. al [18] expose a method based on tracking the relationship of the ridges and furrows in the fingerprint image. The minutiae change this relationship, so they can be detected by finding these changes. The relationship change are exrtacted following the ridges and forrows along the local direction.

An example of the algorithm performance can be seen in the figure 24.

Fig. 24. Example of the algorithm performance. Taken from [18]

Xudong Jiang et. al [19] present a technique that uses the basic idea of [17]. Oriented filters are applied to some regions that need to be smoothed in the image and adaptively is traced the gray-level ridge of the original fingerprint. They say that one of the obvious advantages of the ridge detection by tracing is that it approximates the ridge with piecewise linear lines. The post-processing is based not only on the location relationship of the minutiae, also in the associate ridge relationship and the certainty level of the minutiae.

In their ridge detection algorithm while they trace the gray-level ridge, the skeleton ridge is created, by searching the maximum and minimum points of the local ridge section set.

An oriented low-pass filter is used to reduce the noise of fingerprint image and to link the small breaks in a ridge, because the band-pass filtering due to estimation error of the ridge frequency may cause minutiae location error and spurious minutiae.

Minutiae are detected while the ridges are traced and the skeleton image is formed. Tracing should take place only within the region of interest of the fingerprint image.

Some spurious minutiae should be eliminated in the post-processing step, because the minutiae are detected without considering its spatial and structural relationship to other minutiae and ridges in the neigborhood.

They only recognize the minutiae structures shown in the figure 25. The two reasons for only recognizing those minutia structures are that their approach avoids producing some other spurious minutia structures, and that the post-processing should rely on the reliable information, because removing spurious minutiae and leave genuine minutiae are two equally important tasks.

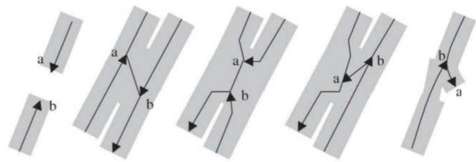


Fig. 25. Spurious minutiae structures. Taken from [19]

An example of the algorithm performance can be seen in the figure 26.

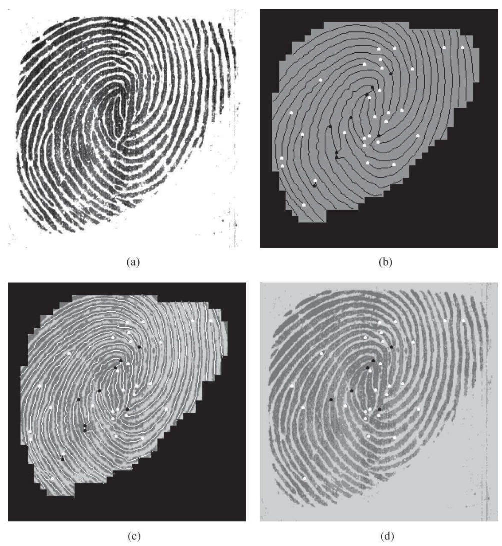


Fig. 26. (a) A sample of fingerprint, the detected skeleton ridges and minutiae before the post processing (b), (c) and after the post processing (d). Taken from [19]

Fronthaler et. al [20] use filters sensitive to parabolic and linear symmetries to find the minutia position and direction. Ridge bifurcation and termination are the wanted minutia types. They use the minutiae just for alignment (registration) of two fingerprint, this allows that the number of minutiae remain low.

Parabolic and linear symmetry are used to model and extract the local structure in a fingerprint. Both symmetries can be derived by separable filtering of the orientation tensor. For a more detailed information about symmetry filters, i.e. symmetry derivatives of Gaussians, they refer to [83].

In the figure 27 is shown minutia points (bifurcation and termination) with their direction. Besides the corresponding filter responses (parabolic symmetry of order $m = 1$) are displayed.

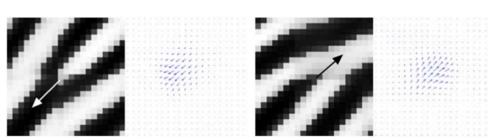


Fig. 27. Left: ridge bifurcation and filter response; Right: ridge ending and filter response. Taken from [20]

An example of the algorithm performance can be seen in the figure 28.

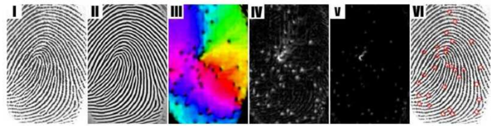


Fig. 28. Minutiae point detection process. Taken from [20]

Xin Gao et. al [85] propose a method based on the Gabor phase field. The approach is worked on the transform domain. The fingerprint image is convolved by a Gabor filter and a complex image is obtained. Then, it is transformed into the amplitude and phase part. The minutiae are extracted directly from the Gabor phase field.

Some fuzzy techniques have been introduced in the literature to extract minutiae from the gray-scale fingerprint image, like Sagar et al [80, 82].

Roli Bansal et al [110] make a review of the minutiae extraction process. Rozita Mohd Yusof and Norrozila Sulaiman [111] present a review of this topic too.

2.5.3 Table of Minutiae Extraction Algorithms

A summary of the studied algorithms is presented in the table 14.

Table 14. Different exposed minutiae extraction algorithms on direct gray scale image

Algorithm Reference	Minutiae Extraction Technique	Year
Dario Maio and Davide Maltoni [17]	based on ridge line following direct on the gray-scale fingerprint image	1997
Jinxiang Liu et. al [18]	based on tracking the relationship of the ridges and furrows on the gray-scale fingerprint image	2000
Xudong Jiang et. al [19]	uses the basic idea of [17], minutiae are detected while the ridges are traced and the skeleton image is formed	2001
Fronthaler et. al [20]	use filters sensitive to parabolic and linear symmetries to find the minutia position and direction	2005
Xin Gao et. al [85]	based on the Gabor phase field	2010

2.5.4 Partial Conclusion and Main Weaknesses Detected

The minutiae extraction algorithms are directly dependent on the quality image to have a good performance. The low quality images need preprocessing steps to increase the contrast and reduce noise that affect them, as they may generate false minutiae. The steps of binarization and esqueletonization cause loss of information and often generate false minutiae.

It should be more emphasis on the definition of local criteria to validate a minutiae (something like a quality measure of the minutiae), which would be very helpful to the matching step. It would also be good to have more sophisticated identification models that allow extend the classification of the minutiae by including trifurcation, islands, bridges, spurs, etc.

There fingerprint images like the latent prints that besides having very poor quality, contain very little amount of minutiae points, which means that in the matching step this information is not enough to identify the fingerprint. This implies the need to create a descriptor with more fingerprint features than only the minutiae points.

The principal issues detected on this topic are:

1. Balancing the false minutiae extraction and the loss of the real minutiae.

2. Choose the representation of the features descriptor for the matching process.
3. Choose the features that will be extracted from the fingerprint image when the minutiae are not enough.

2.6 Detecting and Removing False Minutiae

During the process of minutiae extraction many false minutiae are listed for the features extraction algorithms. To improve the overall performance of an automatic fingerprint identification system, it is very important to preserve true minutiae while removing spurious minutiae in postprocessing. Usually, it happens due to the existence of the low quality areas in the fingerprint images, or due to scars or cuts. This problem has been attacked by some researchers. Most of them try to find the false minutiae selected as candidates at an earlier stage of the algorithm and eliminated.

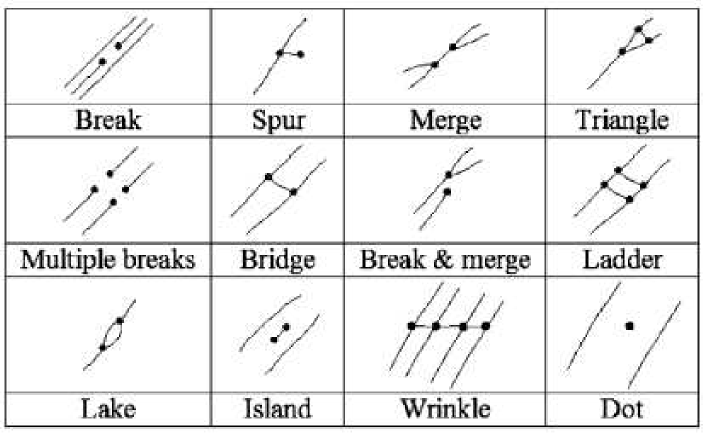


Fig. 29. Examples of false minutiae (black dots). Taken from [21]

Ratha et al. [112] presented a postprocessing stage for eliminating spurious feature points based on the structural and spatial relationships of the minutiae. For instance, two minutiae in a real fingerprint cannot occur within a very short distance of each other. The following heuristics are used to validate minutia points found in minutiae extraction step:

1. Ridge break elimination: Two end points with the same orientation and within a distance threshold are eliminated.
2. Spike elimination: An end point which is connected to a bifurcation point and is also within a distance threshold is eliminated.
3. Boundary effects: The minutiae detected within a specified border of the boundary of the foreground areas are deleted.

Kim et al.[113] propose a method to eliminating false minutiae. They propose a minutiae postprocessing algorithm. The algorithm was based on the orientation and flow of ridges as well as minutiae distance and connectivity. Experimental results showed that this algorithm is indeed very effective; eliminating great deal of false minutiae while retaining most of true minutiae. It was also shown that the proposed algorithm improves the fingerprint matching performance.

Another paper about postprocessing minutiae algorithms is presented by Lu et al. [114]. They proposed an effective and efficient postprocessing algorithm for the minutiae extraction. Thus, this new postprocessing algorithm makes efforts to reliably differentiate spurious minutiae from true ones. These efforts include making use of ridge number information, referring to the original gray-level image, designing and arranging various processing techniques properly, and selecting various processing parameters carefully.

Popovic et al. [115] focused their work on detecting and removing spurious minutiae in regions with broken ridges (often called creases). If those regions are large, enhancement techniques can not overcome spurious minutiae detection since ridges remains broken (in cases of small creases it is possible to re-connect ridges during enhancement process). Since most enhancement techniques use information about orientation fields, they tried to analyze if and how this information can be used in minutiae filtering algorithms. They extent their research by including information about singular points position in their minutiae filtering algorithm. In their paper, they presented a method for spurious fingerprint minutiae detection. Although enhancement by directional filtering can reconnect some ridge breaks caused by narrow creases, some broad creases remain. Then multiscale directional information is used to detect and eliminate spurious minutiae in those fingerprint regions of broken ridges.

Feng Zhao and Xiaoou Tang [21] proposed to use the fingerprint valley instead of ridge for the binarization-thinning process to extract fingerprint minutiae. They use several preprocessing steps on the binary image in order to eliminate the spurious lakes and dots, and to reduce the spurious islands, bridges, and spurs in the skeleton image. This allows the true minutiae preserved and false minutiae removed in later postprocessing stages. Finally, using the intrinsic duality property of fingerprint image they developed several postprocessing techniques to efficiently remove spurious minutiae. Duality property says that for each ridge ending, there is generally a corresponding valley bifurcation and vice versa, with the only exception at the singularity points (cores and deltas). Especially, they defined an H-point structure to remove several types of spurious minutiae including bridge, triangle, ladder, and wrinkle all together.

Jabeen and Khan [22] proposed a hybrid false minutiae removal algorithm for accurate matching of low quality fingerprints. This algorithm is designed to be used in efficient security systems. It does not depend on filters like many present algorithms, rather it removes false minutia considering statistical information from a thinned binary image. The algorithm consists of two hybrid processes i.e. boundary elimination and false minutiae removal. Boundary minutiae can be removed by a very simple proposed algorithm named as Plus Rule (figure 30). The rule creates a plus sign on each minutia. It works on each minutia, to find a black pixel across these lines. If a black pixel is not detected, in any line, the minutia is marked as boundary minutia and hence removed. The authors expressed that the algorithm works efficiently for false minutia that occurs frequently in a dry, oily, dark or dirty fingerprint. It has been shown that this algorithm is able to detect and cancel false minutiae associated with bridges, ridge breaks and spikes with boundary minutiae.

Bansal et al. [23] proposed a minutiae extraction algorithm which extracts minutiae using the morphological Hit or Miss Transform. But, the success of any minutiae extraction technique depends on the quality of the input image, the image needs to be enhanced before processing it for minutiae extraction. After close examination of the binarized image, it can be seen that the disconnections and isolated regions (dots, holes, islands etc.) in a binary image may introduce a number of spurious minutiae in thinned images. Therefore some morphological operators are applied to the binarized image as follows:

1. Spur removal: spurs are short length irregularities in the boundary of the original object. They can be removed by a process called pruning.
2. Spurious bridge removal: some linked parallel valleys may be separated to eliminate spurious bridges (figure 31) in the skeleton image.
3. Spurious holes removal: spurious holes are regions (figure 32) with an area (number of pixels) below a threshold. The threshold value has to be selected appropriately so that it is not so small that it does not remove the spurious hole and not so large that it distorts the actual image. These regions are identified and filled so that the subsequent thinning operation does not produce spurious closed loops.

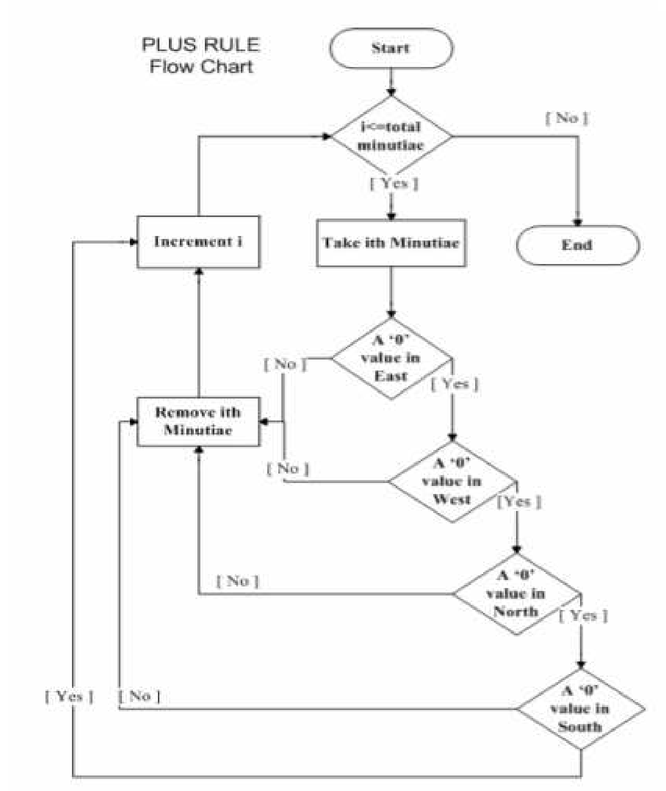


Fig. 30. Plus Rule. Taken from [22]

4. Isolated islands removal: islands (figure 33) are short lines with an area below a threshold. Removing these areas eliminates any spurious dots and islands from the binarized image.

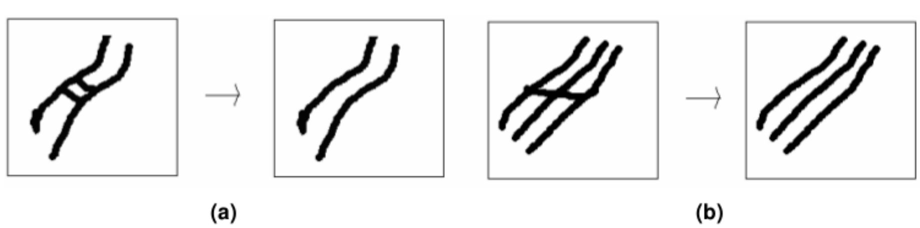


Fig. 31. (a) and (b): Spurious bridges removal. Taken from [23]

In this paper the binarize image is analysed to avoid spurious minutiae. Nevertheless they also propose a method for postprocessing:

1. If the distance between two bifurcations is less than a threshold T and they are in the same ridge, remove both of them.
2. If the distance between two terminations is less than a threshold T and the difference between their angles of orientation is very small, then the two minutiae are regarded as false and are removed.
3. If the distance between a bifurcation and a termination is less than a threshold T such that T is the average inter ridge width, then they are removed

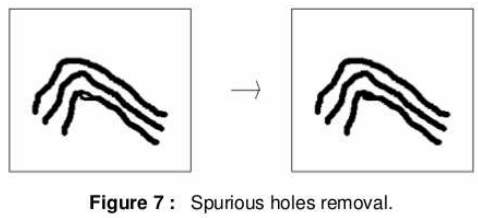


Fig. 32. Spurious holes removal. Taken from [23]

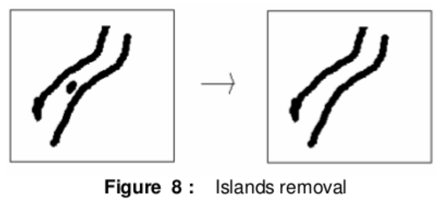


Fig. 33. Islands removal. Taken from [23]

Stephen et al. [116] proposed a paper developing a new idea for removing the false minutiae by implementing some fuzzy rules. This algorithm tests the validity of each minutiae point in thinned image and examines the local neighborhood around the point. The first step in this algorithm is to find the distance between termination and bifurcation. They have used Euclidean method to find distance [117]. After finding distance, they will use some rules to remove these false minutia points. The proposed fuzzy rules are as follows:

1. Rule 1: If the distance between termination and bifurcation is less than D, then remove this minutia.
2. Rule 2: If the distance between two bifurcations is less than D, then remove this minutia.
3. Rule 3: If the distance between two terminations is less than D, then remove this minutia.

2.6.1 Table of Detecting and Removing False Minutiae

Table 15. Removing False Minutiae Methods

Reference	Technique	Year
Ratha et al. [112]	structural and spatial relationships of the minutiae	1995
Kim et al. [113]	orientation and flow of ridges	2001
Lu et al. [114]	ridge number information	2002
Popovic et al. [115]	multiscale directional information	2007
Feng Zhao and Xiaoou Tang [21]	duality property	2007
Jabeen and Khan [22]	hybrid processes: boundary elimination and false minutiae removal	2008
Bansal et al. [23]	morphological preprocessing	2010
Stephen et al. [116]	fuzzy rules	2012

2.6.2 Partial Conclusion and Main Weaknesses Detected

There are several papers about this matter, most of them try to look for minutiae after extraction, during postprocessing step, but only [23] does actions after binarization to avoid or decreased the probability to extract spurious minutiae. This aspect seems to be remarkable and it is possible to get some improvements of the efficacy for feature extraction using this approach. We consider very important to have a platform where the most of these algorithm are implemented. So it is possible to make comparisons and combination of these and so we can select the best combination at each moment.

3 Verifinger feature extraction

Verifinger is a fingerprint identification technology intended for biometric systems developers and integrators [118]. It is available as a software development kit. The algorithm that we referring in this section is the Verifinger 4.2 presented in [119].

The figure 34 exposes the general steps involve in the features extraction of the Verifinger algorithm.

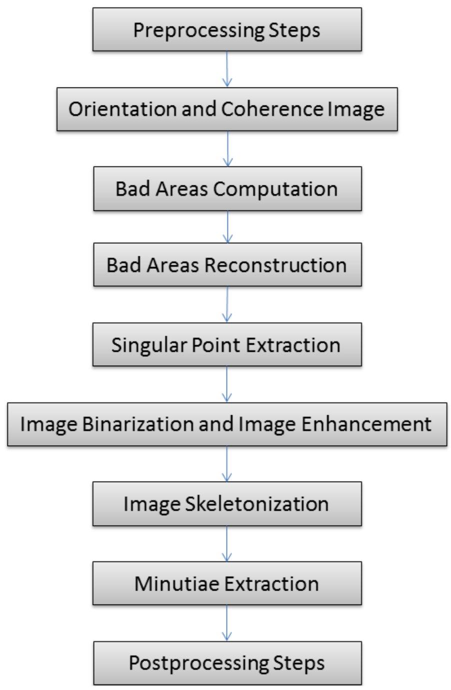


Fig. 34. Algorithm general steps.

The algorithm at first place makes an smoothing of the image (average filter(3×3), average filter(4×4), image normalized(128×128), average filter(3×3))(I_S). Then the orientation and the coherence images are computed and the averages gradients use 3×3 block areas. Using those images are calculated the bad areas of the image using a 3×3 fixed window. Then the bad areas are reconstructed using the orientation.

The singular points are extracted using Poincaré [120, 121], and the high curvature zones. Using the smoothed image (I_S), the orientation and coherence images are calculated using a 12×12 window. Singular points are grouped using the orientation image. The bad areas are computed with a 16×16 window. The image is smoothed using the orientation image.

The image is binarized using *BinarizeImageTwo*, then the image of the ridge image density in the good areas is calculated (I_g). The image is smoothed with the average filter with a 16×16 window. The image is binarized using I_g with *BinarizeImageA*. Then using this image, the orientation and coherence images are computed with a 12×12 window. The image is smoothed using its orientation. Then this image is smoothed using the ridge density image and its orientation image a specific number of times. Then the image is binarized using *BinarizeImageA* and I_g. Then the image is binarized using *BSmoothImage*.

The image is skeletonized, and the lines and eyes formations are eliminated. The minutiae are extracted from the skeletonized image, and the spurious minutiae are marked and eliminated.

3.1 Enhancement

In this part will explain the principal algorithms related to the enhancement step.

The method ***PuttyImage*** foreach pixel (i, j) and with a window of 3, assign to that pixel the average of the entire block which (i, j) is the center. Always uses the original image to smoothed.

NormalizeImage normalizes using smoothed pixel value and and average pixel variance value.

MapBadArea fills with white color the image bad areas.

3.2 Orientation and Coherence

In this subsection it will explain the way to compute the orientation and coherence images.

ComputeOrientImage computes the orientation and coherence images. Foreach pixel is computed the squared gradients using the modify Sobel mask (figure 35).

-1		1
-4		4
-1		1

-1	-4	-1
1	4	1

Fig. 35. The modified Sobel mask.

Then foreach pixel (i, j) is founded the average of its gradients in the block which it is the center and bild 3 matrixs in this process. Then using the matrixs values is building the coherence and orientation images.

ComputeBadArea in first place estimeted a matrix where its values are 0 and 1, which says if the coherence value of the pixel (i, j) is greather than a specified threshold. If more of the half of the pixels of a block meet the condition and there is not singular points, then this zone is marked like bad quality area.

3.3 Singular Points Extraction

The functions which are related to the singular point extraction are explains below.

ExtractSingularPoints extracts singular points from blocked orientation image computing Poincaré index along a small closed curve around point (pi), and along a longer curve (pi_2). If both values are not equals, if the value of $pi * pi_2$ is greather than 0 then $pi = pi_2$, if not then $pi = 0$. If pi is not 0 then that is a singular point (core if 1, double core if 2 and delta if -1).

GroupSingularPoints leaves just one singular point on each group of singular points founded.

3.4 Binarization

There are explained the methods that are related with the binarization step.

BinarizeImage transforms a pixel to 0 if the his value is less than the mean of the pixels that belong to the window wich its the center, and 255 in other case.

BinarizeImageTwo does the same as the previous function but the threshold is the half sum of the averages of two blocks. The window values is given as a parameter and the block centers is the pixel has being analyzed.

ComputeGImage computes what is called G image (G image) which is the image of the ridge image density in the good areas of the image.

At the method ***BinarizeImageA*** the image is binarized with 3 differents average values of the pixels and the window sizes of each of one are specified in the function parameters.

BSmoothImage binarized the image giving value 0 at pixel whose the image smoothed (returned by the method *PuttyImage*) value is less than 128, and it is set to 255 in other case.

3.5 Skeletonize

This section describes the methods that relate to the skeletonization of the image.

SkeletonizeImage thins lines to one pixel width at the binarized image. First, for each pixel black of the image it creates an environment of that pixel as follows $E_{ij} = B_{i-1j} \mid B_{i-1j+1} \ll 1 \mid B_{ij+1} \ll 2 \mid B_{i+1j+1} \ll 3 \mid B_{i+1j} \ll 4 \mid B_{i+1j-1} \ll 5 \mid B_{ij-1} \ll 6 \mid B_{i-1j-1} \ll 7$, where B_{ij} is 1 if I_{ij} is black and 0 if not. I is the binarized image. If $vfSkelet[E_{ij}]$ value is not 0 then the pixel with environment E_{ij} is eliminated (set to white), this is done for each pixel a number of times equal to $maxIterations$ or while not exists pixels were skeletonized by iteration.

EliminateLines eliminates lines breakthroughs in good areas of the image. Find an end of line on the image (not less than $minLength$) and look at the pixel environment, with a window size specified as parameter, other end of a line (not less than $minLength$), if are not black pixel between this pixels and the difference of the pixel directions is not less than $dirDiff$ then draw a line between these points.

EliminateEyes eliminates the eyes formations in the good areas of the image, using the method *EliminateEye*.

3.6 Minutia Extraction

The principal functions involved at the minutiae extraction process are explained below.

ExtractMinutiae extracts the minutiae from the skeletonized image. Its computes the total transitions white-black around each point, if has value of 1 then it is a termination, if its value is 3 then its a bifurcation.

SelectBadMinutiae assigns the backgraound color to the bad minutiae.

3.7 Partial Conclusion and Main Weaknesses Detected

The most current references used by this minutiae extractor dating from 2004, so that we can ensure that in the literature exist better techniques.

The preprocessing algorithms used have poor performance when the image has a certain low quality. When no fingerprint in the image, the algorithm is not able to see it, and performs all processing. The minutiae extraction is performed on skeletonized image causing the generation of false minutiae and sometimes the real points are not in the correct position. The minutia type is wrong classified in many cases. No exist a measure of accuracy of the minutiae, which would represent an aid to get a better performance for the matching process. The only features which extracted from the fingerprints are the singular points and the minutiae.

4 NBis Feature Extraction

Mindtct is a minutiae detection system as part of fingerprint image software distribution developed by the National Institute of Standards and Technology (NIST) for the Federal Bureau of Investigation (FBI) and Department of Homeland Security (DHS) [122]. This algorithm receive a fingerprint file as input and return a minutiae file. It first generates images quality maps, then it binarizes image, detects minutiae, removes false minutiae, gets the neighbors minutiae, assess minutiae quality and finally it returns the minutia file.

4.1 Generate Image Quality Maps

The first step in Mindtct algorithm is to generate the Quality Maps. This map is necessary to compute the minutiae quality and it is the result of the unification of the direction map, low contrast map, low flow map and high curve map.

4.1.1 Direction Map

The goal of the direction map is to represent areas in the image with the ridge structure enough, it let reliable minutiae detection. To analyse fingerprint locally, the image is divided in a grid of blocks. For direction map all pixels in one block will be assign to the same direction of ridge flow. In this process first is determined how much local information is needed to have features extraction reliable. The measures of features inside the window is assign to each pixel in the block. For every block in the image, the surrounding window is roted incrementally and one Discrete Fourier Transform analysis is conducted at each orientation (see figure 36).

Fig. 36. Direction map results.

4.1.2 Low Contrast Map

Often fingerprint has low contrast or smudges zones in background image. This matter difficults the determination of the flow of ridges in those regions. The low contrast map is calculated in low contrast zones of the image (figure 37). This map separates the background image of the fingerprint and remove smudges. Minutiae will not be detected inside blocks of low contrast of the image. One way to identify this kind of blocks from one block of well-defined ridges is to compare their pixel intensity distribution. By definition, there is a small dynamic range in pixel intensity in low contrast areas, so the distribution of pixel intensity will be very narrow. One block with well-defined ridges has a wide range of pixel intensity with pixels ranging from very light in the middle of the valleys to very dark in middle of ridges. This algorithm compute the pixel intensity distribution within surrounding windows of the blocks. The highest and lowest tails are trimmed. With the others values the distribution is computed and if this value is sufficiently low then the block is marked as low contrast block. The threshold for pixel intensity to decide when one block is considerate of low contrast is 5.

Fig. 37. Low contrast map results.

4.1.3 Low Flow Map

It is possible that derived of the initial direction map some blocks do not have a dominant ridge flow. These blocks usually are located in poor quality areas of images. Initially those blocks do not have assigned one orientation in directions map (see figure 38). At the same time some of those blocks may be assigned an orientation by interpolating the ridge flow in neighbours blocks. This map marks the blocks that could not be assigned initially a dominant ridge flow. If it is detected minutiae in these blocks, the quality of them are reduced because have been detected in less reliable zones of the image.

4.1.4 High Curve Map

Another problematic section of the image in the process to extract minutiae is the region of high curvature. This happen usually in regions of cores and delta in the image [122]. The figure 39 show how the blocks with high curvature are marked in the map. Two measures are used. First **vorticity:** measure the cumulative change in direction of ridge flow around all neighbour of the block. Second **curvature:** measures the largest change in direction between the ridge flow of the block and the ridge flow for each of its neighbours.

Fig. 38. Low flow map results.

If it is detected minutiae in these blocks, the quality of them are reduced because have been detected in less reliable zones of the image.

Fig. 39. High curve map results.

4.1.5 Quality Map

This process produces an image map called Quality Map (figure 40). The information obtained in the others steps is integrated in a general map with 5 levels of quality.

4.2 Image Binarization

The algorithm of minutiae detection of this software is designed for work on images where black pixels are ridges and white pixels are valleys. To create binarized image each pixel in grey-scale is analized to

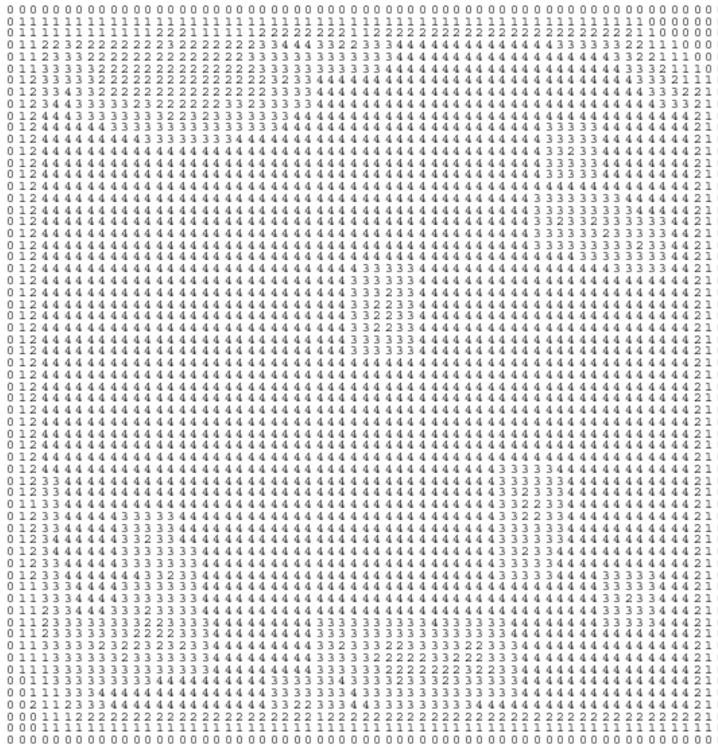


Fig. 40. Quality map results.

determine what color assign: black or white. Each pixel will receive a binary value based in direction of flow related to the ridges in its block (see figure 41).

If ridge flow can not be detected in the block the value for the pixel will be 0. If ridge flow is detected in the block, the pixel intensity around the current pixel is analized using a rotated grid. With current pixel in the center, the grid is rotated so that its rows are parallel to the local ridge flow direction. The intensities of the grey-scale are accumulated along each rotated row in the grid forming a vector of row sums. The value of the central pixel is calculated by multiplying the sums of the central row for the number of row in grid and comparing this value with the grey scale intensity inside the full grid. If multiplying is less than total intensity then central pixel is black, otherwise is white.

The figure 41 show an original fingerprint and the binarized fingerprint from Mindtct algorithm.

4.3 Minutia Detection

This stage methodically scan the binarized image of a fingerprint, identifying patterns of pixels that indicate ending or splitting.

Candidate ridge endings are detected in the binarized image by scanning consecutive pairs of pixels in the image looking for sequences that match the patterns in figures 42 and 43.

Using the representation above, a series of minutiae patterns are used to detect candidate minutia points in the binarized fingerprint image.

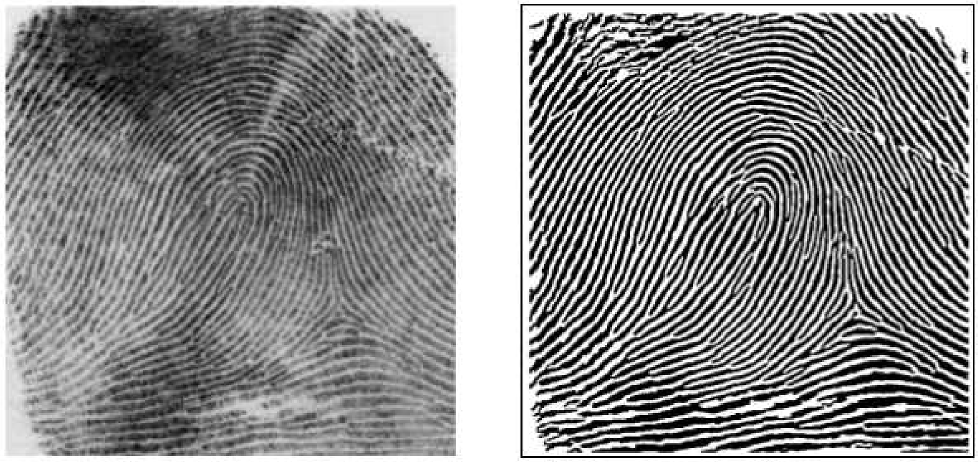


Fig. 41. Binarization results.

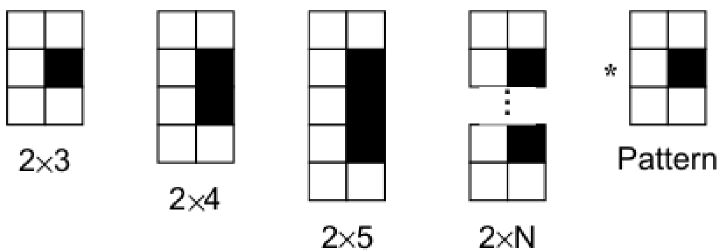


Fig. 42. Pixel pattern used to detect ridge endings.

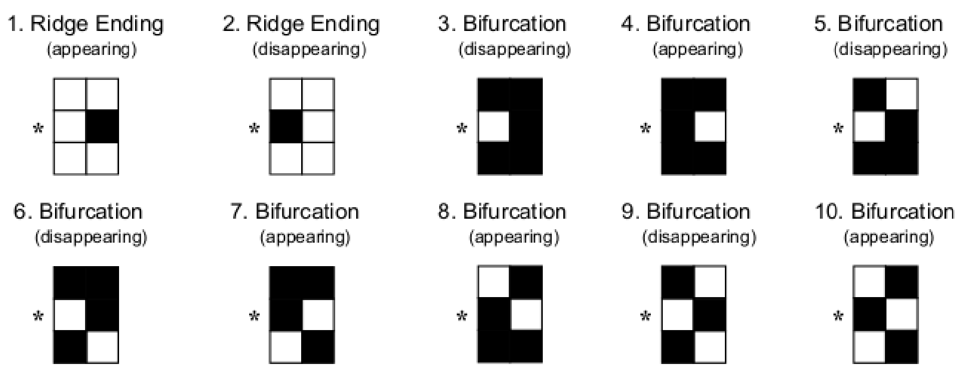


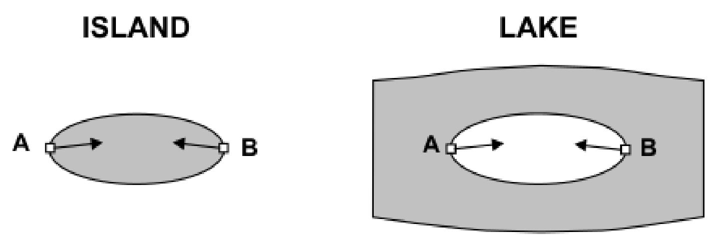
Fig. 43. Pixel patterns used to detect minutiae.

4.4 Removing False Minutiae

Using the patterns in figure 43, the candidates minutiae points are detected with as few as six pixels. This facilitates a particularity greedy detection scheme that minimizes the chance of missing true minutia, however, many false minutiae are included in the candidate list. Because of this, much effort is spent on removing the false minutiae. These steps include removing islands, lakes, holes, minutiae in regions of poor image quality, side minutiae, hooks, overlaps, minutiae that are too wide, and minutiae that are too narrow (pores).

4.4.1 Removing Islands and Lakes

In this step, ridge ending fragments and spurious ink marks (islands) along with interior voids in ridges (lakes) are identified and removed (figure 44). These features are somewhat larger than the size of pores in the friction skin and they are often elliptical in shape; therefore, they typically will have a pair of candidate minutia points detected at opposite ends [114]. The pair of minutia must be within 16 pixels of each other. If so, then the directions of the two minutiae must be nearly opposite ($\geq$ 123.75) each other. Next, both minutiae must lie on the edge of the same loop, and the perimeter of the loop must be $\leq$ 60 pixels. If all these criteria are true, then the pair of candidate minutiae are removed and the binary image is altered because island or lake is filled. This step is the only removal step that modifies the binary image.

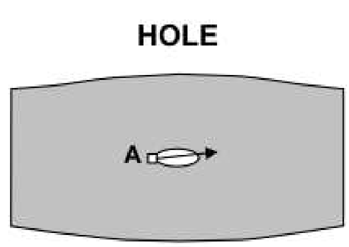



```
1. If (distance(A,B) <= 16 pixels) Then
   2. If (direction_angle(A,B) >= 123.75°) Then
      3. If (on_loop(A) && on_loop(B)) Then
         4. If (loop_length <= 60 pixels) Then
            5. remove(A,B)
            6. fill_loop()
```

Fig. 44. Removal of islands and lakes.

4.4.2 Removing Holes

Here a hole is defined similarly to an island or lake, only smaller, and the loop need only have one minutia point on it (figure 45) [113].




```
1. If (on_loop(A)) Then
   2. If (loop_length <= 15 pixels) Then
      3. remove(A)
```

Fig. 45. Removal of holes.

4.4.3 Removing Pointing to Invalid Block

This step and the next identify and remove candidate minutiae that are located near blocks that contain no detectable ridge flow (figure 46). These blocks are referred to as containing invalid ridge flow direction and represent low-quality areas in the fingerprint image.

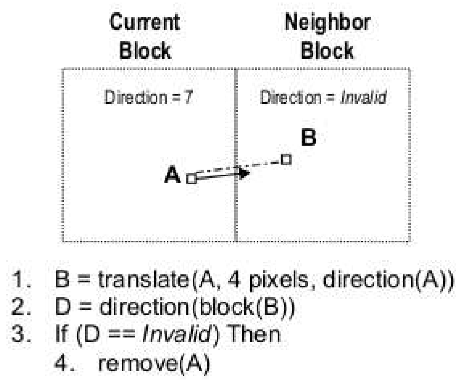


Fig. 46. Removal of minutia pointing to an invalid block.

4.4.4 Removing Near Invalid Blocks

Here, the proximity of a candidate minutia to a number of surrounding blocks with invalid ridge flow direction is evaluated (figure 47). Given a minutia point, the blocks sufficiently close to the minutia, and immediately neighbouring the block in which the minutia resides, are tested in turn. If one of these neighbouring blocks has invalid ridge flow direction, then its surrounding 8 neighbours are tested. The number of surrounding blocks with valid ridge flow direction are counted, and if the number of valid blocks is < 7, then the original minutia point is removed from the candidate list.

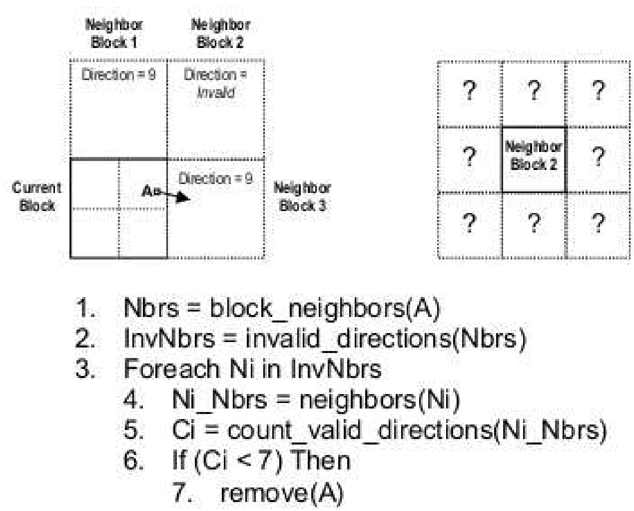
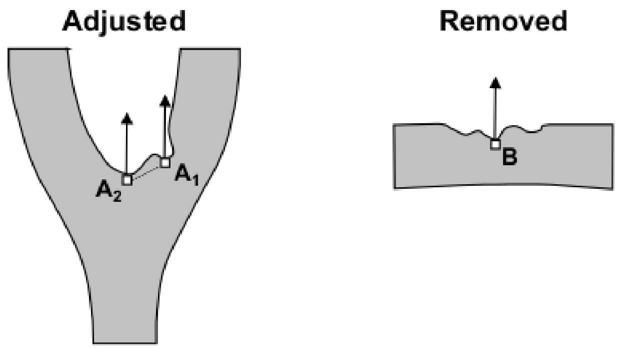


Fig. 47. Removal of minutia near invalid blocks.

4.4.5 Removing or Adjust Side Minutiae

This step accomplishes two purposes. The first is to fine-tune the position of a minutia point so that it is more symmetrically placed on a ridge or valley ending. In the process, it may be determined that there is no clear symmetrical shape to the contour on which the candidate minutia lies. This is often the case with points detected along the side of a ridge or valley instead of the ridge or valley's ending. In this case, the misplaced minutia point is removed. In figure 48, the illustration on the left depicts the adjustment of a minutia point from point A1 to A2. The figure 48 on the right depicts the removal of a side point, B.



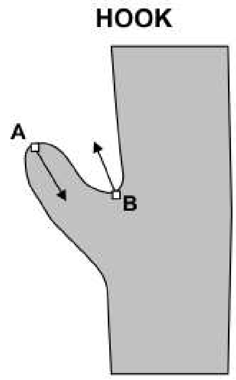

```
1.  Pts = trace_contours(A, 7 pixels)
2.  R_Pts = rotate_points_vertical(Pts, direction(A))
3.  {Min_Ys,Max_Ys} = min_and_max_Ys(R_Pts)
4.  If (#Min_Ys == 1) Then
        Adjust(A, Pts[Min_Y1])
5.  Else If ({Min_Ys,Max_Ys} ==
        {Min_Y1,Max_Y1,Min_Y2}) Then
            Min_Y = point_at_min_Y(R_Pts, Min_Ys)
            Adjust(A, Pts[Min_Y])
6.  Else revove(A)
```

Fig. 48. Removal or adjustment of minutiae on the side of a ridge or valley.

To accomplish this, starting at the candidate minutia point, the edge of either the ridge or valley is traced to the right and to the left seven pixels, producing a list of 15 contour points. The coordinates of these contour points are rotated so that the direction of the candidate minutia is pointing vertical. The rotated coordinates are then analysed to determine the number and sequence of relative maxima and minima in the rotated y-coordinates. If there is only one y-coordinate minima, then the point of the minimum is assumed to lie at the bottom of a bowl-shaped rotated contour, and the candidate minutia is moved to correspond to this position in the original image. If there are more than one y-coordinate minima, then a specific sequence of minima-maxima-minima must exist, in which case the candidate minutia is moved to the point in the original image corresponding to the lowest y-coordinate minima. Again, this is assumed to be the bottom of a relatively bowl-shaped rotated contour. If there is more than one y-coordinate minima and there is not an exact minima-maxima-minima sequence along the rotated contour, then the minutia point is determined to lie along the side of a ridge or valley, and it is removed from the candidate list.

4.4.6 Removing Hooks

This feature is formed by two minutiae with different type, one on small piece of ridge and the other in small valley, that are relatively close to each other (see figure 49). Both points must be within 16 pixels.




```
1.  If (distance(A,B) <= 16 pixels) Then
    2.  If (direction_angle(A,B) >= 123.75°) Then
        3.  If (type(A) != type(B)) Then
            4.  Pts = trace_contours(A, 30 pixels)
            5.  If (in_points(Pts, B)) Then
                6.  remove(A,B)
```

Fig. 49. Removal of hooks.

Their directions must be opposite ($\geq$ 123.75 d). The minutiae must be different type and must be within the same ridge or valley edge of 30 pixels. If all these are true, then the two points are removed from the list of candidates.

4.4.7 Removing Overlaps

An overlap is a discontinuity in a ridge or valley. This occur generally in fingerprint impression process (figure 50). A break in a ridge causes two false ending, while a break in a valley causes two false bifurcations [112]. These cases happen when two minutiae are within 8 pixels of each other and their directions are opposite. Then the direction of the line joining the two minutiae is calculated. If the difference between the direction of first minutia and the joining line is ($\leq$ 90Â°), then the two minutiae are removed from the candidate list. Otherwise, if the minutiae are within 6 pixels of each other, and there are no pixel value transitions along the joining line, then the points are removed from the candidate list.

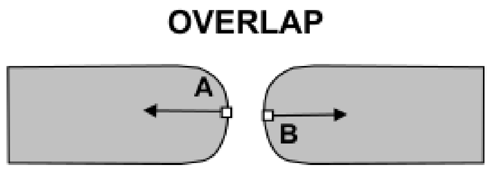
4.4.8 Removing Too Wide Minutiae

An ending usually is comprised by Y-shape valley enveloping a black rod (figure 51). The opposite happen for a bifurcation. Simple tests are applied to evaluate the quality of this Y-shape.

This algorithm evaluates if the structure of enveloping a ridge or valley is relatively Y-shaped and not too wide. The edge of ridge or valley is traced to the left and to the right for 20 pixels, getting two lists of contour points. In each contour are calculated the distances between pixels of index 10 and 20. The radio between these distances is calculated (D20 / D10) and if this value is greater than two the the minutiae is removed.

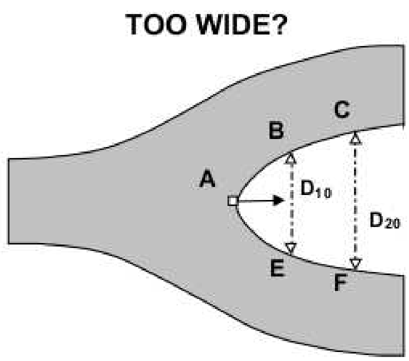
4.4.9 Removing Too Narrow Minutiae

This step finds points in structures too narrow, this happen generally in pores of friction skin (see figure 52). Starting with the candidate minutia point, F, its coordinates are translated 3 pixels opposite the minutia's direction. The top edge and bottom edges of the enveloping structure are then located at (Q,P). From these




```
1. If (distance(A,B) <= 8 pixels) Then
   2. If (direction_angle(A,B) >= 123.75°) Then
      3. If(type(A) == type(B)) Then
         4. J = join_direction(A,B)
         5. If(direction_angle(180°-A,J) <= 90° Then
            6. remove(A,B)
         7. Else If (distance(A,B) <= 6 pixels &&
                  free_path(A,B)) Then
            8. remove(A,B)
```

Fig. 50. Removal of Overlaps.




```
1. Pts1 = trace_contour(A, 20 pixels)
2. Pts2 = trace_contour(A, -20 pixels)
3. B = Pts1[10]; C = Pts1[20]
4. E = Pts2[10]; F = Pts2[20]
5. D10 = distance(B,E)
6. D20 = distance(C,F)
7. If ((D20/D10) > 2.0) Then
   8. remove(A)
```

Fig. 51. Removal of too wide minutiae.

two points, the edge is traced to the left 10 pixels and to the right 8 pixels. The points at the end of the 10 pixel contours are stored (A,B), and the points at the end of the 8-pixel contours are stored (C,D). Next, distances are computed between these pairs of points, and the radius (D1/D2) is computed. If the radius is ≤ 2.25, then the minutia point is removed from the candidate list. In fact, if the process fails to find any of the points in the illustration, then the candidate minutia is removed. It should be noted that, mindtct, only searches for minutiae that are too narrow within high-curvature regions or regions where ridge flow direction is non-determinable.

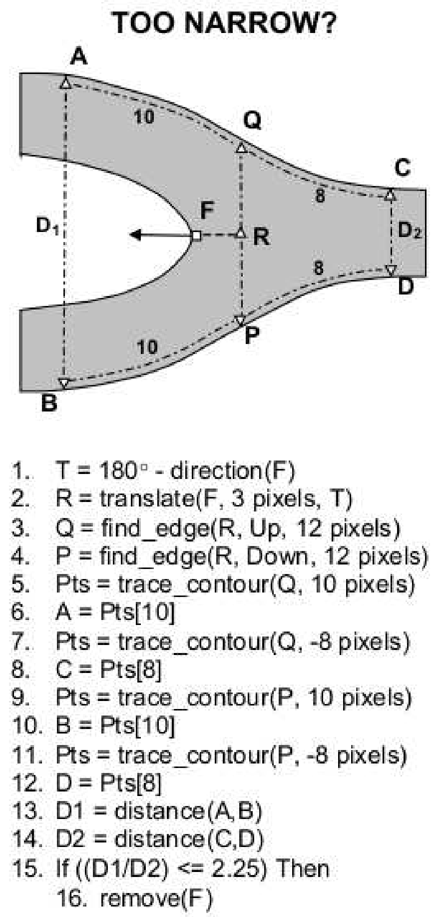


Fig. 52. Removal of too narrow minutiae.

4.5 Obtain the Neighbors Minutiae

Fingerprint minutiae matchers usually use additional information to just the points. This information generally included minutia direction, type and may include relative information of the minutia neighbors. Beyond a minutia position, direction, and type. Different AFIS systems use different minutia neighborhood topologies and attributes. A common attribute is the number of ridges involves (called ridge crossings) between a minutia and each of its neighbors minutiae. For example, the Fbi's IAFIS uses ridges crossing between a minutia and the eight nearest neighbors where each neighbor is the closest within a specified octant [123]. The minutiae vicinity scheme distributed with this system has been directly inherited from HO39 [124]. Up to 5 nearest neighbors are reported. Given a minutia point, the closest neighbors below (in the same pixel column), and to the right (within entire pixel columns) in the image are selected. These nearest neighbors are sorted in order of their direction, starting with vertical and working clockwise. Using this topology, ridge counts are computed and recorded between a minutia point and each of its nearest neighbors.

4.6 Assess Minutia Quality

One of the goals this software package was to compute reliability and quality to be associated with each detected minutia point. Even with the effort to remove false minutiae from the candidates list, usually false minutiae stay in this list. A robust quality measure can help manage this in that false minutiae should be assigned a lower quality than true minutiae. Through dynamic thresholding, a trade off between retaining false minutiae and throwing away true minutiae may be determined. To this end, Mindtct algorithm, computes and reports minutiae qualities. Two factor are considerated to obtain a measure of quality for each detected minutia. The first factor is the Quality Map and the second is based in the value of intensity of the pixels within minutia's neighborhood. The radius of the neighborhood is 11 pixels. This is sufficiently large to contain generous portions of an average ridge and valley. A high quality region within a fingerprint image will have significant contrast that will cover the full gray-scale spectrum. Consequently, the mean pixel intensity of the neighborhood will be very close to 127. For similar reasons, the pixel intensities of an ideal neighborhood will have a standard deviation ≥ 64.

This results in a quality value on the range .01 to .99. A low quality value represents a minutia detected in a lower quality region of the image, whereas a high quality value represents a minutia detected in a higher quality region.

4.7 Partial Conclusions and Main Weaknesses Detected

This algorithm is very useful tool to feature extraction because it gives a quality coefficient for each minutia and maps with important information about fingerprint as direction map, low contrast map, low flow map, high curve map and quality map. But we looked some weaknesses in this algorithm:

1. The minutia detection step is very simple. It is based on find patterns over binarize image. This simplicity introduces many spurious minutiae and makes the algorithm dependant of the post-processing step. Although in this process many false minutiae are removed, the number of minutiae extracted by the algorithm is elevated.

5 Experimental results

Experiments were performed with the fingerprint database NIST27 whose images have marked minutiae, to get a total of 257 fingerprints. For each image run the algorithms VeriFinger and Nbis minutiae extraction. For each instance, the number of minutiae that the algorithm not extracted (False Negatives (FN)), and the number of minutiae extracted that were not right (False Positive (FP)) are computed. Tables 16 and 17, is shown the average value of each parameter, and the number of times each algorithm was better than the other for each case. The comparison was made between NBIS minutiae that have a quality greater or equal to 30 with all the minutiae extracted by VeriFinger, and the minutiae with quality greater than or equal to 18 against all the VeriFinger minutiae, respectively. To ensure that the minutiae are in the right position, allowed error threshold is 10 pixels with the Euclidean distance.

Table 16. Comparison between Verifinger and Nbis (Minutia Quality ≥ 30)

Algorithm	FN Average	FP Average	Victories (FP)	Victories (FN)
Nbis	30	53	236	55
Verifinger	26	79	20	191

Table 17. Comparison between Verifinger and Nbis (Minutia Quality ≥ 18)

Algorithm	FN Average	FP Average	Victories (FP)	Victories (FN)
Nbis	26	75	141	123
Verifinger	26	79	111	126

In the table 18 is presented the comparison between all the minutiae extracted by the two algorithms.

Table 18. Comparison between Verifinger and Nbis

Algorithm	FN Average	FP Average	Victories (FP)	Victories (FN)
Nbis	20	134	3	204
Verifinger	26	79	254	42

Table 19 shows the average of false negatives and false positives minutiae extracted by Nbis and Verifinger algorithms. We taken the Nbis minutiae with value of quality higher than 33 (recommended by NIST in [122]).

Table 19. Comparison between Verifinger and Nbis

Algorithm	FN Average	FP Average	Victories (FP)	Victories (FN)
Nbis	32	48	191	80
Verifinger	28	62	61	163

Table 20 presents an analysis of the Nbis minutiae extraction. It shows the average number of minutiae extracted with post-processing, without post-processing and the ideal average number.

Table 20. False Minutiae Elimination in NBis and Verifinger Algorithms

Algorithm	Total Images	Ideal Minutiae Count Average	Minutiae Count Average (original extraction)	Minutiae Count Average (without postprocessing)
NBis	257	106	220	3127
Verifinger	257	106	159	377

This table (20) shows that without the post-processing step Nbis extracts a huge amount of false minutiae.

5.1 Partial Conclusions

Nbis algorithm gives a measure of quality for each minutia extracted, the objective of it is to improve the efficacy of the matching algorithms. Due to the strategy of extraction on binarize image many false minutiae are introduced in the first list of candidates to minutia point, that is why the postprocessing step is fundamental to remove almost all of these.

The Verifinger strategy of extraction on thinned image is robust, it is supported by a preprocessing step. For this reason the algorithm introduces less number of false minutiae. Nevertheless, many of these are eliminated by the postprocessing step.

We can conclude that the preprocessing of the image is very important for having high robustness in the extraction, but it is necessary a good postprocessing step, this last is very related with minutia extraction strategy.

6 Conclusions

From the study and analysis of the different steps performed in the minutiae extraction process, we concluded that in each of the reviewed areas there are still research lines to follow, some of which are specified below.

1. Achieving a good reconstruction of the fingerprint image, and reducing the amount of generated false minutiae.
2. Propose descriptors based on fingerprint characteristics to allow, in the matching step, the use not only minutiae, but also the features of the ridges and pores, to improve its effectiveness.

In the study of latent fingerprints is where more work remains to be done. Those images have a very poor quality, and with only minutiae is not enough to get a good performance of the matching process.

References

1. Chen, T.P., Jiang, X., Yau, W.Y.: Fingerprint image quality analysis. In: ICIP. (2004) 1253–1256
2. Hong, L., Jain, A., Pankanti, S., Bolle, R.: Fingerprint enhancement (1996)
3. Wang, L., Bhattacharjee, N., Gupta, G.K., Srinivasan, B.: Adaptive approach to fingerprint image enhancement. In: MoMM. (2010) 42–49
4. Chikkerur, S., Cartwright, A.N., Govindaraju, V.: Fingerprint enhancement using stft analysis. Pattern Recognition **40**(1) (2007) 198–211
5. Sutthiwichaiporn, P., Areekul, V., Jirachaweng, S.: Iterative fingerprint enhancement with matched filtering and quality diffusion in spatial-frequency domain. In: ICPR. (2010) 1257–1260
6. Lei, J., Peng, Q., You, X., Jabbar, H.H., Wang, P.S.P.: Fingerprint enhancement based on wavelet and anisotropic filtering. IJPRAI **26**(1) (2012)
7. Greenberg, S., Aladjem, M., Kogan, D., Dimitrov, I.: Fingerprint image enhancement using filtering techniques. In: ICPR. (2000) 3326–3329
8. Novikov, S., Glushchenko, G.: Fingerprint ridges structure generation models. Proc. SPIE Intl Workshop Digital Image Processing and Computer Graphics (1997) 270–274
9. R. Cappelli, A. Lumini, D.M., Maltoni, D.: Fingerprint image reconstruction from standard templates. IEEE Trans. Pattern Analysis and Machine Intelligence **29**(9) (2007) 1489–1503
10. Ross, A., Shah, J., Jain, A.K.: Towards reconstructing fingerprints from minutiae points. Proceedings of SPIE Conference on Biometric Technology for Human Identification II **5779** (2005) 68–80
11. Feng, J., Jain, A.K.: Fm model based fingerprint reconstruction from minutiae template. In: ICB. (2009) 544–553
12. Feng, J., Jain, A.K.: Fingerprint reconstruction: From minutiae to phase. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2) (2011) 209–223
13. Kim, D.H., Park, R.H.: Fingerprint binarization using convex threshold. In: Computer Graphics and Imaging. (2003) 224–227
14. Zhang, Y., Xiao, Q.: An optimized approach for fingerprint binarization. (2006) 391–395
15. J. S. Bartunek, M. Nilsson, J.N.I.C.: Adaptive fingerprint binarization by frequency domain analysis. Asilomar Conference on Signals, Systems and Computers - ASILOMAR, pp. 598-602 (2006)
16. Bartunek, J.S., Nilsson, M., Nordberg, J., Claesson, I.: Improved adaptive fingerprint binarization. In: CISP ’08 Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 5 - Volume 05. (2008) 756–760

17. Maio, D., Maltoni, D.: Direct gray-scale minutiae detection in fingerprints. IEEE Trans. Pattern Anal. Mach. Intell. **19**(1) (1997) 27–40
18. Liu, J., Huang, Z., Chan, K.L.: Direct minutiae extraction from gray-level fingerprint image by relationship examination. In: ICIP. (2000) 427–430
19. Jiang, X., Yau, W.Y., Ser, W.: Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge. Pattern Recognition **34**(5) (2001) 999–1013
20. Fronthaler, H., Kollreider, K., Bigün, J.: Local feature extraction in fingerprints by complex filtering. In: IWBRS. (2005) 77–84
21. Zhao, F., Tang, X.: Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. Pattern Recognition **40**(4) (2007) 1270–1281
22. Jabeen, S., Khan, S.: A hybrid false minutiae removal algorithm with boundary elimination. System of Systems Engineering. SoSE '08. IEEE (2008)
23. Bansal, R., Sehgal, P., Bedi, P.: Effective morphological extraction of true fingerprint minutiae based on the hit or miss transform. International Journal of Biometric and Bioinformatics. ISSN: 1985-2347 **4**(2) (2010)
24. Tabassi, E., Wilson, C.L., Watson, C.I.: Fingerprint image quality. National Institute of Standars Technology, NISTIR 7151 (August 2004)
25. Arora, K., Garg, D.: A quantitative survey of various fingerprint enhancement techniques. International Journal of Computer Applications **28**(5) (2011)
26. W.J., B.: Embryologic development of epidermal ridges and their configuration. Birth Defects Original Article Series, vol. 27, no. 2 (1991)
27. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of fingerprint recognition. Second Edition, Springer (2009)
28. Wilson, C.L., Blue, J.L., Omidvar, O.M.: Training dynamics and neural network performance. Neural Networks **10**(5) (1997) 907–923
29. C. Watson, A. Wilson, K.M.M.I.R.S.: Studies of one-to-one fingerprint matching with vendor sdk matchers
30. Omidvar, O., Wilson, C.L.: Information content in neural net optimization. **6** (1993) 91–103
31. Chen, Y., Dass, S.C., Jain, A.K.: Fingerprint quality indices for predicting authentication performance. In: AVBPA. (2005) 160–170
32. Alonso-Fernandez, F., Fiérrez-Aguilar, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J., Fronthaler, H., Kollreider, K., Bigün, J.: A comparative study of fingerprint image-quality estimation methods. IEEE Transactions on Information Forensics and Security **2**(4) (2007) 734–743
33. Ratha, N., Bolle, R.: Automatic fingerprint recognition systems, New York: Springer-Verlag (2004)
34. Tabassi, E., Wilson, C.L.: A novel approach to fingerprint image quality. In: ICIP (2). (2005) 37–40
35. Wu, M., Yong, A., Zhao, T., Guo, T.: A systematic algorithm for fingerprint image quality assessment. In: ICIC (2). (2011) 412–420
36. Munir, M.U., Javed, M.Y., Khan, S.A.: A hierarchical k-means clustering based fingerprint quality classification. Neurocomputing **85** (2012) 62–67
37. Tao, X., Yang, X., Zang, Y., Jia, X., Tian, J.: A novel measure of fingerprint image quality using principal component analysis(pca). In: ICB. (2012) 170–175
38. Soweon Yoon, E.L., Jain, A.K.: On latent fingerprint image quality. In: 5th International Workshoop Computacional Forensics, Tsukuba, Japan (2012)
39. Zhu, E., Yin, J., Zhang, G., Hu, C.: A gabor filter based fingerprint enhancement scheme using average frequency. IJPRAI **20**(3) (2006) 417–430
40. Thai, R.: Fingerprint image enhancement and minutia extraction. (2003)

41. Mehtre, B.M., Murthy, N.N., Kapoor, S., Chatterjee, B.: Segmentation of fingerprint images using the directional image. Pattern Recognition **20**(4) (1987) 429–435
42. Mehtre, B.M., Chatterjee, B.: Segmentation of fingerprint images - a composite method. Pattern Recognition **22**(4) (1989) 381–385
43. : Vc2002 - fingerprint verification competition (2002)
44. Jain, A.K., Ratha, N.K., Lakshmanan, S.: Object detection using gabor filters. Pattern Recognition **30**(2) (1997) 295–309
45. Hong, L., Wan, Y., Jain, A.K.: Fingerprint image enhancement: Algorithm and performance evaluation. IEEE Trans. Pattern Anal. Mach. Intell. **20**(8) (1998) 777–789
46. Kass, M., Witkin, A.P.: Analyzing oriented patterns. Computer Vision, Graphics, and Image Processing **37**(3) (1987) 362–385
47. Bazen, A.M., Gerez, S.H.: Systematic methods for the computation of the directional fields and singular points of fingerprints. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7) (2002) 905–919
48. Simon-Zorita, D., Ortega-Garcia, J., Cruz-Llanas, S., Gonzalez-Rodriguez, J.: Minutiae extraction scheme for fingerprint recognition systems. In: ICIP (3). (2001) 254–257
49. Lee, C.E., Bhattacharjee, N.: Fingerprint image processing and minutiae extraction for fuzzy vault. In: MoMM. (2009) 36–43
50. Rao, A.R.: A taxonomy of texture descriptions
51. Sherlock, B., Monro, D., Millard, K.: Fingerprint enhancement by directional fourier filtering. Visual Image Signal Processing **141**(4) (1994) 87–94
52. Yoon, S., Feng, J., Jain, A.K.: Latent fingerprint enhancement via robust orientation field estimation. In: IJCB. (2011) 1–8
53. I. W. Selesnick, R. G. Baraniuk, N.C.K.: Fingerprint enhancement by directional fourier filtering. Signal Processing Magazine IEEE **22**(6) (2005) 123–151
54. Chen, Q., Micchelli, C.A., Peng, S., Xu, Y.: Multivariate filter banks having matrix factorizations. SIAM J. Matrix Analysis Applications **25**(2) (2003) 517–531
55. Kovacevic, J., Vetterli, M.: Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for rn. IEEE Transactions on Information Theory **38**(2) (1992) 533–555
56. Strang, G., Nguyen, T.Q.: Wavelets and filter banks. Wellesley-Cambridge Press (1997)
57. Cohen, A., Daubechies, I.: On the instability of arbitrary biorthogonal wavelet packets. SIAM J. Math. Anal **24**(5) (1993)
58. Yang, G.Z., Burger, P., Firmin, D.N., Underwood, S.R.: Structure adaptive anisotropic image filtering. Image Vision Comput. **14**(2) (1996) 135–145
59. Wu, C., Shi, Z., Govindaraju, V.: Fingerprint image enhancement method using directional median filter, in biometric technology for human identification. In: Proceedings of the SPIE. (2004)
60. Ye, Q., Xiang, M., Cui, Z.: Fingerprint image enhancement algorithm based on two dimension emd and gabor filter. International Workshop on Information and Electronics Engineering (IWIEE) **29** (2012) 1840–1844
61. Fronthaler, H., Kollreider, K., Bigün, J.: Local features for enhancement and minutiae extraction in fingerprints. IEEE Transactions on Image Processing **17**(3) (2008) 354–363
62. Cheng, J., Tian, J.: Fingerprint enhancement with dyadic scale-space. Pattern Recognition Letters **25**(11) (2004) 1273–1284
63. He, Y., Tian, J., Luo, X., Zhang, T.: Image enhancement and minutiae matching in fingerprint verification. Pattern Recognition Letters **24**(9-10) (2003) 1349–1360
64. Chopra, J., Upadhyay, D.P.: Various fingerprint enhancements and matching technique. International Journal of Electronics and Communication Engineering **5**(3) (2012) 279–289

65. J. L. Araque, M. Baena, B.E.C.D.N., Vizcaya, P.R.: Synthesis of fingerprint images. Proc. 16th Intl Conf. Pattern Recognition (2002) 422–425
66. R. Cappelli, D. Maltoni, D.M.A.K.J., Prabhakar, S.: Handbook of fingerprint recognition. Springer (2003)
67. Vizcaya, P., Gerhardt, L.: A nonlinear orientation model for global description of fingerprints. Pattern Recognition **29**(7) (1996) 1221–1231
68. Ross, A., Shah, J., Jain, A.: From template to image: Reconstructing fingerprints from minutiae points. IEEE Trans. Pattern Analysis and Machine Intelligence **29**(4) (2007) 544–560
69. Cabral, B., Leedom, L.: Imaging vector fields using line integral convolution. Proc. 20th Ann. Conf. Computer Graphics and Interactive Techniques (1993) 263–270
70. Hill, C.: Risk of masquerade arising from the storage of biometrics. Master's thesis, Australian Nat Univ (2001)
71. Sherlock, B., Monro, D.: A model for interpreting fingerprint topology. Pattern Recognition **26**(7) (1993) 1047–1055
72. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint image reconstruction from standard templates. IEEE Trans. Pattern Analysis and Machine Intelligence **29**(9) (2007) 1489–1503
73. Otsu, N.: Threshold selection method from grey-level histograms. IEEE Trans. Syst. Man. Cybern. **8** (1979) 62–66
74. Moayer, B., Fu, K.S.: A tree system approach for fingerprint pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. **8**(3) (1986) 376–387
75. Xiao, Q., Raafat, H.: Fingerprint image postprocessing: A combined statistical and structural approach. Pattern Recognition **24**(10) (1991) 985–992
76. Verma, M.R., Majumdar, A.K., Chatterjee, B.: Edge detection in fingerprints. Pattern Recognition **20**(5) (1987) 513–523
77. Liang, X., Bishnu, A., Asano, T.: A near-linear time algorithm for binarization of fingerprint images using distance transform. In: IWCIA. (2004) 197–208
78. Munshi, P., Mitra, S.K.: A rough-set based binarization technique for fingerprint images. (2012)
79. S. M Rajbhoj, P.B.M.: An improved binarization based algorithm using minutiae approach for fingerprint identification. International Journal of Engineering and Advanced Technology (IJEAT) **1** (2012)
80. Sagar, V.K., Ngo, D.B.L., , Foo, K.C.K.: Fuzzy feature selection for fingerprint identification. In: 29th Annual International Carnahan.Security Technology. (1995) 85–90
81. Maio, D., Maltoni, D.: Neural network based minutiae filtering in fingerprints. Fourteenth International Conference Pattern Recognition **2** (1998) 1654–1658
82. Sagar, V.K., Beng, K.J.: Hybrid fuzzy logic and neural network model for fingerprint minutiae extraction. International Joint Conference on Neural Networks (IJCNN) **5** (1999) 3255–3259
83. Bigün, J., Bigun, T., Nilsson, K.: Recognition by symmetry derivatives and the generalized structure tensor. IEEE Trans. Pattern Anal. Mach. Intell. **26**(12) (2004) 1590–1605
84. Fronthaler, H., Kollreider, K., Bigün, J.: Local features for enhancement and minutiae extraction in fingerprints. IEEE Transactions on Image Processing **17**(3) (2008) 354–363
85. Gao, X., Chen, X., Cao, J., Deng, Z., Liu, C., Feng, J.: A novel method of fingerprint minutiae extraction based on gabor phase. In: ICIP. (2010) 3077–3080
86. Zenzo, S.D., Cinque, L., Levialdi, S.: Run-based algorithms for binary image analysis and processing. IEEE Trans. Pattern Anal. Mach. Intell. **18**(1) (1996) 83–89
87. Scotti, F., Gamassi, M., Piuri, V.: Fingerprint local analysis for high-performance minutiae extraction. In: ICIP (3). (2005) 265–268

88. Shi, Z., Govindaraju, V.: A chaincode based scheme for fingerprint feature extraction. Pattern Recognition Letters **27**(5) (2006) 462–468
89. Shin, J.H., Hwang, H.Y., Chien, S.I.: Detecting fingerprint minutiae by run length encoding scheme. Pattern Recognition **39**(6) (2006) 1140–1154
90. Alibeigi, E., Rizi, M.T., Behnamfar, P.: Pipelined minutiae extraction from fingerprint images. In: CCECE. (2009) 239–242
91. Sainath, M., Rao, T.S., Strom, B.J., Mikael, N.: Implementation and evaluation of nist biometric image software for fingerprint recognition. Biosignals and Biorobotics Conference (BRC), ISSNIP (2011) 1–5
92. Xiao, Q., Raafat, H.: Fingerprint image postprocessing: A combined statistical and structural approach. Pattern Recognition **24**(10) (1991) 985–992
93. F., L.W., H., L.S., H., L.W., A., L.: Fingerprint recognition using neural networks. In: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing. (1991) 226–235
94. J.C., A., A., J., J.C., P., F., P., S., S., J.M., V.: Real-time minutiae extraction in fingerprint images. Image Processing and Its Applications **2** (1997) 871–875
95. Alessandro, F., Zsolt, K.V., Alberto, L.: Fingerprint minutiae extraction from skeletonized binary images. Pattern Recognition **32**(5) (1999) 877–889
96. Jiang, X., Yau, W.Y.: Fingerprint minutiae matching based on the local and global structures. In: ICPR. (2000) 6038–6041
97. Tico, M., Kuosmanen, P.: An algorithm for fingerprint image postprocessing. In: In Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers. (2000) 1735–1739
98. Prabhakar, S., Jain, A.K., Pankanti, S.: Learning fingerprint minutiae location and type. Pattern Recognition **36**(8) (2003) 1847–1857
99. Shah, S., Sastry, P.S.: Fingerprint classification using a feedback-based line detector. IEEE Transactions on Systems, Man, and Cybernetics, Part B **34**(1) (2004) 85–94
100. Chikkerur, S., Govindaraju, V., Pankanti, S., Bolle, R.M., Ratha, N.K.: Novel approaches for minutiae verification in fingerprint images. In: WACV/MOTION. (2005) 111–116
101. Zhao, F., Tang, X.: Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. Pattern Recognition **40**(4) (2007) 1270–1281
102. Humbe, V., Gornale, S.S., Manza, R., Kale, K.V.: Mathematical morphology approach for genuine fingerprint feature extraction. IJSCI International Journal of Computer Science Issues **1** (2007) 53–59
103. Usman, A.M., Anam, T., A., K.S., Sarwat, N.: Fingerprint image: pre- and post-processing. Int. J. Biometrics **1**(1) (June 2008) 63–80
104. B.N., L., B., R.K., K.R., V., M., P.L.: Minutiae extraction in fingerprint using gabor filter enhancement. In: Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies. ACT 2009, Washington, DC, USA, IEEE Computer Society (2009) 54–56
105. R., K., P.S., S., A., K.: A novel method for fingerprint feature extraction. In: International Conference on Networking and Information Technology (ICNIT). (june 2010) 1–5
106. R., P.A., A., Z.M.: A novel approach for fingerprint matching using minutiae. In: Proceedings of the 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation. AMS 2010, Washington, DC, USA, IEEE Computer Society (2010) 317–322
107. Pathak, P.: Image compression algorithms for fingerprint system. IJSCI International Journal of Computer Science Issues **7**(9) (2010) 45–50
108. Gnanasivam, P., Muttan, S.: An efficient algorithm for fingerprint preprocessing and feature extraction. Procedia CS **2** (2010) 133–142

109. Bansal, R., Sehgal, P., Bedi, P.: Effective morphological extraction of true fingerprint minutiae based on the hit or miss transform. International Journal of Biometrics and Bioinformatics(IJBB) **4** (2010) 71–85
110. Bansal, R., Sehgal, P., Bedi, P.: Minutiae extraction from fingerprint images - a review. CoRR **abs/1201.1422** (2012)
111. Yusof, R.M., Sulaiman, N.: A review on minutiae extraction of fingerprint. AWER Procedia Information Technology and Computer Science **1** (2012) 453–457
112. Ratha, N.K., Chen, S., Jain, A.K.: Adaptive flow orientation-based feature extraction in fingerprint images. Pattern Recognition **28**(11) (1995) 1657–1672
113. Kim, S., L.D., Kim, J.: Algorithm for detection and elimination of false minutiae in fingerprint images. 2001 Proceedings of the Third International Conference on Audio and Video Based Biometric Person Authentication (2001)
114. Lu, H., J.X., Yau, W.: Effective and efficient fingerprint image postprocessing. 7th International Conference on Control, Automation, Robotics and Vision (ICARCV) **2** (2002)
115. Popovic, B., M.L., Bandur, M.: Spurious fingerprint minutiae detection based on multiscale directional information. Facta Universitatis Series : Electronics and Energetics ISSN 0353-3670 (2007)
116. Stephen, M., Reddy, P., Kartheek, V., Suresh, C.: Removal of false minutiae with fuzzy rules from the extracted minutiae of fingerprint image. Proceedings of the InConINDIA 2012, AISC 132, Springer-Verlag. (2012) 853–860
117. James Stephen, M., Prasad Reddy, P.: Implementation of easy fingerprint image authentication with traditional euclidean and singular value decomposition algorithms. Int. J. Advance. Soft Comput. ISSN: 2074-8523 (2011)
118. Neurotechnology, C.: Neurotechnology biometric and artificial intelligence technology. `http://www.neurotechnology.com/verifinger.html` (1998-2013)
119. Neurotechnology, C.: Verifinger 4.2. algorithm and source code description. (1998-2005)
120. Kawagoe, M., Tojo, A.: Fingerprint pattern classification. Pattern Recognition **17**(3) (1984) 295–303
121. Bazen, A.M., Gerez, S.H.: Systematic methods for the computation of the directional fields and singular points of fingerprints. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7) (2002) 905–919
122. Group, N.: The science of fingerprints. Rev. 12-84, U.S. Department of Justice, Federal Bureau of Investigation. Available from U.S. Government Printing Office, Washington D.C. 20402. (2001)
123. Group, N.: Electronic fingerprint transmission specification. CJIS-RS-0010 (V7). Available from Criminal Justice Information Services Division, Federal Bureau of Investigation, 935 Pennsylvania Avenue, NW, Washington D.C. 20535. (2001)
124. Group, N.: Home office automatic fingerprint recognition system (hoafrs). Science and Technology Group, Home Office, London (1993)

RT_055, noviembre 2013

Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona

Diseño de Portada: Di. Alejandro Pérez Abraham

RNPS No. 2142

ISSN 2072-6287

Indicaciones para los Autores:

Seguir la plantilla que aparece en www.cenatav.co.cu

C E N A T A V

7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;

La Habana. Cuba. C.P. 12200

Impreso en Cuba