CHAPTER III

DESIGN OF A NEW CLASS OF FINGERPRINT FEATURE EXTRACTION METHODS

The five major challenges felt by many prominent methods spelled out in the previous chapter motivated this research which got set on an adventure to evolve a new class of Fingerprint Feature Extraction methods addressing the above said issues. The following evolved methods deal with all the above said difficulties and have novel features.

1. A novel approach based on the use of 3 x 3 fixed size templates for feature extraction is proposed. Here the template is applied to the binary image at selected potential location to detect the presence of minutiae at these locations, making the process effective and faster. Here, no such assumptions were made as in (FEFST) and (RBGST).

2. In Gabor filter bank based fingerprint feature extraction methods [94-99], the filter frequency determine the bandwidth of the Gabor filter should be selected properly. If it is too large, spurious ridges may be created in the filtered image, whereas if it is too small, nearby ridges may be merged into one. This may be resolved through adaptive frequency selection in fixed length square finger code developed in this research overcoming the limitations of (PIDCT).

3. A position invariant fingerprint feature extraction in DCT Domain method is proposed through finding the core point of each fingerprint image[98], then
crop the fingerprint image into 128×128 pixels using its core point as the reference point. Then the image is transformed to DCT Domain, for extracting the fingerprint features. This gives a better solution for the limitation of (RBGST).

4. A modified algorithm is developed for ridge line following discussed in [39,64,75] for grayscale fingerprint image feature extraction methods; here removal of false minutiae is implemented through projection in the same orientation on the ridge lines, after they terminate or intersect other ridge lines (minutiae detection) up to the thresh count. This method addresses the limitation of (AFS).

5. The minutiae features may be affected by various kinds of noise, such as small ridge segments, ridge breaks lead to false ridge endings which will be misinterpreted as minutiae. An efficient method is proposed to avoid such false minutiae by considering only the ridge bifurcations and for minimizing the time complexity, the templates were grouped based on their similarities overcoming the limitations mentioned in (FMRG).

This thesis revolves around these five major solutions evolved and the details about them are discussed below. The new class of Fingerprint Feature Extraction methods resulted in this research are given below:

1. A novel technique for fingerprint Feature Extraction using Fixed Size Templates - FEFST.
3. An Effective False Minutiae Removal method in Gray scale fingerprint images – FMRG.

4. A Position Invariant fingerprint feature extraction in DCT Domain - PIDCT.

5. An Adaptive Frequency Selection in fixed length square finger code method of fingerprint feature extraction - AFS.

3.1. A Novel Technique for Fingerprint Feature Extraction Using Fixed Size Templates-FEFST

Extracting minutiae from fingerprint image is one of the most important steps in automatic fingerprint identification and classification. Minutiae are local discontinuities in the fingerprint pattern, mainly ridge ending and bifurcation. A novel approach for extraction of minutiae is attempted through a fixed size template matching process. In this proposed method the 3 x 3 fixed size template is applied to the binary image at selected potential location to detect the presence of minutiae at these locations, making the process effective and faster. This approach is simple and efficient than the complicated orientation based template matching methods normally followed in several implementations of feature extraction.

A feature extraction procedure generally consists of three steps: preprocessing, feature extraction and post processing. Preprocessing technique attempts to capitalize on the special ridge and valley nature of finger print images for image filtering. Feature extraction approaches are based on either (a) Thresholding, thinning and minutiae detection [66], [67], [68], [72], [73] and [74] or (b) ridge
following in Grayscale image [39], [64] and [75]. The approach based on Thresholding and thinning is simple in principle [68]. The ridge following approach is generally complex and non-adaptive [39]. The post processing technique attempts to rectify imperfection of the feature extraction process.

This proposed FEFST method also consists of following three steps: Image enhancement using Wiener filter, Binarization and thinning, Template matching, post processing.

3.1.1 Image Enhancement using Wiener Filter

Histogram equalization defines mapping of gray levels \( p \) into gray levels \( q \) such that distribution of gray level \( q \) is uniform, yielding a good contrast stretch which improves the detectability of many image features. Then adaptive Wiener method for noise reduction is applied. The Wiener filter is based on local statistics estimated from a local neighborhood \( M \) at size 3x3 at each pixel.

\[
w(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2} (I(n_1, n_2) - \mu) \quad (3-1)\]

where \( M \) is the set of pixels in a 3x3 neighborhood centered at a particular pixel, \( \nu^2 \) is the noise variance, \( \mu \) and \( \sigma^2 \) are mean and variance of \( M \).

3.1.2. Binarization and Thinning

The operation that converts a grayscale image into a binary image is known as Binarization. The Binarization process is carried out using an adaptive thresholding, where each pixel is assigned a new value (1 or 0) according to the intensity mean in a
local neighborhood \((13 \times 13)\) as follows:

\[
I_{new}(n1, n2) = \begin{cases} 
1 & \text{if } I_{old}(n1, n2) \geq \text{Local Mean} \\
0 & \text{Otherwise} 
\end{cases}
\]  \hspace{1cm} (3-2)

Thinned (one pixel thickness) ridgelines are obtained using morphological thinning operation [61].

---

**Figure 3.1. FEFST Method for Minutiae Extraction**

3.1.3. Template Matching

1. In order to improve the efficiency of previous complicated methods using templates of variable length, width and orientation based template matching, the fixed 3x3 templates [62] are used, which are applied to only a subset of pixels that are
likely to correspond to ridge ends or bifurcation. These points are selected by calculating the crossing number at each pixel. The crossing number $C_n(P)$ at data point $P$ is defined as the half of the cumulative successive differences between pairs of adjacent pixels belonging to the 8-neighbourhood at $P$.

$$C_n(P) = \frac{1}{2} \sum_{i=2}^{8} |Val(P_i) - Val(P_{i-1})|$$

where $P_1, P_2, \ldots, P_8$ are the pixels belonging to the 8 neighborhood of $P$ and $Val(P_i)$ is the value $(0, 1)$ of the Pixel $P_i$. The pixels having $C_n(P) \leq 2$ correspond to ridge ending, while pixels having $C_n(P) \geq 1$ correspond to ridge bifurcation. In the previous methods [73], [39] only the pixels with $C_n(P) = 1$ as ridge endings and those pixels having $C_n(P) \geq 3$ as bifurcations were considered. But in practice this may lead to missing of several true minutiae.

2. For each valid crossing number, the fixed size 3x3 template [39] was applied in its corresponding potential location. If the template is matched with the image then mark that potential location as minutiae. Because of applying the template only for the potential locations with valid crossing number, the run time is consistently diminished in comparison with the previous methods.

3.1.4. Post Processing

If the fingerprint image is in poor quality then it will lead to many false minutiae. In this proposed FEFST method these extra minutiae are identified by a
4×4 local area centered in the potential minutiae. If \( \mu_i + K_i\sigma < \mu_g + K_g\sigma \) with the local neighborhood then the particular minutiae is ignored, where \( \mu_i, \mu_g, \sigma \) and \( \sigma_g \) are local mean, global mean, local standard deviation and global standard deviation respectively and \( K_1, K_2 = 1 \), [68].

The size of the template was chosen to be 3×3 in view of the fact that it yields very good results with reduced time and space complexity in identifying the position of the minutiae accurately with practically insignificant false minutiae. If the size of the template is changed to higher order, then time for permuting the template and the time for matching will increase which is unnecessary since 3×3 template itself detects 100% minutiae. By this novel method the efficiency of old method is improved and got reduced time and space complexity, using fixed non-orientation based 3×3 template instead of applying the orientation based variable length and width templates.

3.2. A Novel Method for Extracting Ridge Bifurcations in Fingerprint Images by Grouping of Similar Templates – RBGST

Among various types of minutiae, ridge endings and bifurcations are widely used in Automatic Fingerprint Identification System (AFIS). As there are chances for false ridge endings, here only the bifurcations are considered for the minutiae extraction. The RBGST method proposed an efficient algorithm to extract the bifurcations by using templates for personal identification. This proposed algorithm avoids numerous template comparisons and also it reduces the occurrence of false minutiae.
Minutiae in fingerprint images are not always well defined, and an enhancement algorithm that can improve the clarity of minutiae structures is required. Therefore, enhancement is one of the most important steps in the automatic fingerprint identification system. The enhancement process consists of various steps such as Adaptive Threshold, Histogram Equalization, Wiener Filtering, Binarization and Thinning [72].

In the previous work [72], instead of ridge orientation and ridge frequency estimation, templates were used for minutiae extraction. Even in templates also the variable sized templates were replaced by fixed size 3 x 3 templates. Here they are applied to only a subset of pixels that are likely to correspond to ridge endings or ridge bifurcations. These points are selected by calculating the crossing number at each pixel. If the template is matched with image then mark that potential location as minutiae.

In this RBGST method, only the bifurcations are considered to avoid false ridge endings from the enhanced image. Here 16 sample templates were used to detect the bifurcations. These templates were grouped into six groups based on their similarities and reduce the number of comparisons. Hence this algorithm produce efficient results when compared with previous methods [72] based on template matching.

3.2.1. Fingerprint Image Enhancement

The Input fingerprint images may be poor in quality due to noise. In order to remove the noise, the images are enhanced using the following steps.
Adaptive Threshold

An adaptive threshold separates the foreground from the background with non-uniform illumination. It outputs a binary image with the local threshold mean-C or median-C to the image.

Histogram Equalization

After Adaptive Threshold process, the histogram equalization is used to make the input fingerprint image look clear. Histogram equalization defines the mapping of gray levels \( p \) into gray levels \( q \) such that distribution of gray level \( q \) is uniform. This mapping stretches contrast for gray levels near the histogram maxima, since contrast is expanded for most of the image pixels. This transformation improves the detect ability of the image features [144].

Applying Wiener Filter

A pixel-wise adaptive Wiener method is proposed to use for noise reduction. The filter is based on local statistics estimated from a local neighborhood of size 3x3 of each pixel, and is given by:

\[
W(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2} (I(n_1, n_2) - \mu) \tag{3-4}
\]

where \( \nu^2 \) is the noise variance, \( \mu \) and \( \sigma^2 \) are local mean and variance. I represent the gray level intensity in \( n_1, n_2 \).

Binarization and Thinning

Here the binarization process is carried out by using a global threshold value [72]. Each pixel is assigned a new value (1 or 0) according to the intensity mean in a
local neighborhood (13x13 pixels), as follow:

\[ I_{\text{new}}(n_1, n_2) = \begin{cases} 1 & \text{if } I_{\text{new}}(n_1, n_2) \geq \text{Local Mean} \\ 0 & \text{Otherwise} \end{cases} \]  

(3-5)

Thinned ridgelines are obtained using morphological thinning operations [144]. The thinned fingerprint image is obtained as output for the enhancement process to extract the bifurcations from the fingerprint image.

3.2.2. RBGST Feature Extraction Algorithm

1. In the proposed RBGST algorithm, totally sixteen 3 x 3 fixed size templates are used to represent the bifurcation models as shown in figure 3.2.

![Figure 3.2: Sample Bifurcation](image)

2. Among all these sixteen templates their centre pixel value is 1 and the sum of their neighborhood pixel is 4.

3. Before template matching the centre pixel is checked for 1 and sum of its neighborhood for 4. If it is satisfied, the location may be a bifurcation pixel.

4. Then the template matching is performed to confirm the Bifurcation. In template matching, the templates are grouped based on their similarities as follows.
5. Consider the templates in first row with numbers 1 to 4, second row with numbers 5 to 8, third row with numbers 9 to 12 and last row with numbers 13 to 16.

6. For example, in templates numbered 1, 10 and 14 the pixels 1, 3 and 5 are filled. Hence any one of these 6, 7 and 8 pixels are filled with above 1, 3 and 5 pixels then a bifurcation occurs. Hence the above three templates can be avoided through a single condition to reduce the number of comparisons.

7. In this manner, the templates 6, 8 and 11 are grouped as one with 4, 5 and 7 pixels as common, the templates 2, 3 and 4 as one with 2, 5 and 6 pixels as common, the templates 9, 12 and 16 as one with 5, 7 and 9 pixels as common, the templates 7, 13 as one with 1, 5 and 6 pixels as common and the templates 5, 15 as one with 4, 5 and 9 pixels as common.

8. To reduce the false bifurcations, 5 X 5 templates are used as shown as in figure 3.3 whose sum of pixel is greater than 7, its centre value is 1 and sum of central 3 x 3 pixels is 4. If it is satisfied, then true bifurcation minutiae is confirmed.

![Figure 3.3. 5 X 5 Template](image)
In this RBGST algorithm, the bifurcation locations are easily obtained by calculating the sum of neighborhood of 3 x 3 pixels and checking the centre pixel for value 1. Hence there is no need for matching each set of pixels with all sets of sample templates. Then the bifurcations are confirmed by checking them with their groups and not with each sample template. In addition, the false bifurcations are minimized by matching the marked locations with 5 x 5 templates. Here again there is no direct template matching is done; Rather, the sum of neighborhood are checked for value greater than or equal to 7 and the sum of 3 x 3 centre pixel sum as 4 with the centre most pixel value as 1. In this way, a large number of comparisons are reduced when compared with the AFERF method [72] and the time complexity also reduced.

3.3 An Effective False Minutiae Removal Method in Gray Scale Fingerprint Images-FMRG

FMRG method presents an effective false minutiae removal algorithm in gray scale fingerprint images. Several Fingerprint feature extraction algorithms in gray scale images [39, 64, 75] have been proposed in the literature, capable of extracting minutia by following the ridge line through a starting point and an oriented direction by "sailing" according to the fingerprint directional image. A set of starting points is determined by superimposing a square-meshed grid on the gray scale image. For each starting point, the algorithm keeps following the ridge lines until they terminate or intersect other ridge lines. When a ridge line terminates or intersects another ridge line (originating a minutia) the algorithm stops and gives the characteristics (coordinates and direction) of the minutia found. Usually, the fingerprint image may
be corrupted by various kinds of noise. Despite the existence of noise, a trained fingerprint expert is often able to correctly identify the minutiae by using various visual clues such as local ridge/valley orientation, ridge/valley continuity, etc. Therefore, we have developed an algorithm that can rely on these visual clues to remove false minutia.

From the mathematical point of view, a ridge line is defined as a set of points which are local maxima along one direction [39]. The ridge line extraction algorithm [39] attempts to locate, at each step, a local maximum relative to a section orthogonal to the ridge direction. By connecting the consecutive maxima, a polygonal approximation of the ridge line can be developed. This algorithm is capable of extracting a ridge line given a starting point and an oriented direction. When a ridge line terminates (originating a ridge ending) or intersects another ridge line (originating a ridge bifurcation), the algorithm stops and gives the characteristics (coordinates and direction) of the minutia found.

This proposed feature extraction process with an effective false minutiae removal method in gray scale fingerprint images can be carried out as follows:

1) Compute the tangent direction \( \varphi_z \) in the ridge line nearest to the starting point \((i_s, j_s)\).

2) Section S in \((i_s, j_s)\) along direction \( \varphi_z + \frac{\pi}{2} \) and with length \( 2\sigma + 1 \).

3) Regularize the section and compute & choose the local maxima.

4) The events which stop the ridge line following process.

6) Eliminate the false minutiae.
Gray scale fingerprint image

Determine a set of ridge starting points

Compute the tangent direction

Sectioning and maximum determination

Regularize the section

Compute and choose the local maxima

Ridge line following method

Excessive bending

Intersection

Termination

Exit from interest area

No minutiae found

C, D, E, F

Set of minutiae points

Any un visited starting points available

Yes

No
3.3.1 Computation of Tangent Direction

In the literature, several methods for estimating image directional information have been proposed. The simplest and efficient method is based on gradient computation [39]. In this method, the gradient phase angle denotes the direction of the intensity maximum change. Thus, the direction $\phi$ of a hypothetical edge which crosses the region centered in pixel $(i, j)$ is orthogonal to the gradient phase angle in $(i, j)$. This method, suffers from the non-linearity due to the computation of the gradient phase angle [39].
In Kawagoe and Tojo method [54], for each 2 x 2 pixel neighborhood, a straight comparison against four edge templates to extract a rough directional estimate was created, which was then arithmetically averaged over a larger region to obtain a more accurate estimate. Stock and Swonger [146], Mehtre, et al. [145] approached to evaluate the tangent direction on the basis of pixel alignments relative to a fixed number of reference directions. Here Donahue and Rokhlin [112] method was used for computing the tangent direction. It used a gradient type operator to extract a directional estimate from each 2 x 2 pixel neighborhood, which is then averaged over a local window by least-squares minimization to control noise. This method also allowed for an unoriented direction to be computed. The computation of an oriented direction was subordinate to the choice of an orientation. In each step of the ridge line following process, the orientation was choused in such a way that $\varphi_e$ comes closest to the direction computed at the previous step. To make this method as computationally inexpensive through precomputing the directional image over a discrete grid and then determining the direction $\varphi_e$ through lagrangian interpolation were performed.

### 3.3.2. Sectioning and Maximum Determination

In each step this method computes a point $(i, j)$, moving $\mu$ pixels from $(i_c, j_c)$ along with direction $\varphi_c$. Then, it calculate the section set $\Omega$ as the set of points belonging to the section segment lying on the ij-plane and having median point $(i, j)$ direction orthogonal to $\varphi_c$ and length $2\sigma + 1$. A new point $(i_n, j_n)$ belonging to the ridge line is selected among the local maxima of the set $\Omega$. The point
$(i_s, j_s)$ becomes the current point $(i_c, j_c)$ and a new direction $\phi_c$, is calculated. $\mu$ and $\sigma$ are parameters whose value can be determined according to the average thickness of the image ridge lines.

The method used in this work for sectioning and maximum determination, proposed by D. Maio, and D. Maltoni [39], sectioning the surface $S$ corresponding to the image $I$, by intersecting $S$ with a cutting plane parallel to the $z$ direction. The section set $\Omega((i_1, j_1), \phi, \sigma)$ centered in $(i_1, j_1)$ with direction $\phi = \phi_c + \frac{\pi}{2}$, and length $2\sigma + 1$ pixels, is defined as:

$$\Omega = \{(i, j) | (i, j) \in I, (i, j) \in \text{segment}((i_{\text{start}}, j_{\text{start}}), (i_{\text{end}}, j_{\text{end}}))\} \quad (3-6)$$

$$(i_{\text{start}}, j_{\text{start}}) = (\text{round} (i_1 - \sigma \cos \phi), \text{round} (j_1 - \sigma \sin \phi))$$

$$(i_{\text{end}}, j_{\text{end}}) = (\text{round} (i_1 + \sigma \cos \phi), \text{round} (j_1 + \sigma \sin \phi))$$

$$\text{round}(x) = \begin{cases} 
\lfloor x + 0.5 \rfloor & \text{if } x > 0 \\
\lceil x + 0.5 \rceil & \text{otherwise}
\end{cases} \quad (3-7)$$

Were segment $\{(i_{\text{start}}, j_{\text{start}}), (i_{\text{end}}, j_{\text{end}})\}$ is the set of points belonging to the discrete segment whose extremes are $(i_{\text{start}}, j_{\text{start}})$ and $(i_{\text{end}}, j_{\text{end}})$.

By means of sorting the points of $\Omega$ from $(i_{\text{start}}, j_{\text{start}})$ to $(i_{\text{end}}, j_{\text{end}})$,

$$(i_1, j_1) = (i_{\text{start}}, j_{\text{start}}), (i_2, j_2), ... (i_m, j_m) = (i_{\text{end}}, j_{\text{end}}), m \approx 2\sigma + 1$$

are obtained.

Local maximum of the section set $\Omega$ can be computed simply by comparing the gray levels of the points belonging to $\Omega$. 

138
3.3.3. Regularize the Section and Compute and Choose the Local Maxima

By the above sectioning and maximum determination process the ridges are easily located, but detecting the corresponding maxima is not straightforward; sometimes, in the middle of a ridge (where a local maximum should exist) there is a local minimum which produces a typical volcano silhouette. So an approach described by D. Maio, and D. Maltoni [39], aimed at regularizing the section silhouette, which makes the determination of the local maxima more reliable. During the ridge line following process, each time a new section is determined and regularizes its silhouette by means of two steps:

i) The first step is based on a local average of the gray levels of the pixels belonging to a number of parallel adjacent sections. This can be produced by sectioning $S$ with $2h+1$ parallel planes ($h \geq 0$), distant one pixel from each other, and by computing $m$ local averages to determine the new gray levels. The section produced by the plane $h+1$ (the central one) originates the section set $\Omega$, but the level $\overline{\text{gray}}(i, j)$ of each point $(i, j \in \Omega)$ is calculated as the average of the gray levels of corresponding points over the $2h+1$ sections.

ii) The next step is based on a convolution with a constant mask $d$ resembling the gaussian silhouette. Let $(i_1, j_1)...(i_m, j_m)$ be the points belonging to $\Omega$ and $\overline{\text{gray}}(i, j),...\overline{\text{gray}}(i_m, j_m)$ be the gray levels computed at step (i); let $d_k, k=1,...,2p+1, (p \geq 0, d_k \geq 0, \sum d_k = 1$ be the elements of the mask $d$. Then the new gray levels $\overline{\text{gray}}(i_{p+1}, j_{p+1}),...\overline{\text{gray}}(i_{m-p}, j_{m-p})$ are computed as:
\[
\text{gray}(i_k, j_k) = \frac{1}{(2p+1)} \sum_{v=(p)}^{(2p+1)} d_{p,v} \cdot \text{gray}(i_{k+v}, j_{k+v})
\]

(3-8)

\[
k = p + 1, \ldots, m - (p)
\]

Then the local maximum required is easily located by comparing the above gray levels and choosing the weak local maximum closest to the center \((i_l, j_l)\). The gray level \(\text{gray}(i_k, j_k)\) is a weak local maximum if and only if

\[
\text{gray}(i_{k-1}, j_{k-1}) \leq \text{gray}(i_k, j_k) \leq \text{gray}(i_{k+1}, j_{k+1}).
\]

With weak maxima instead of strong maxima the ridge line following is guaranteed to work correctly even when a ridge line presents a flat profile.

3.3.4. The Events which Stop the Ridge Line Following:

The events which stop the ridge line following Process are:

1) Exit from interest area: During the ridge line following process, every time a new point \((i_l, j_l)\) is computed, the algorithm checks whether the new point is external to a rectangular window \(W\) which represents the sub image whose minutiae are to be detected. Hence no minutiae have been found.

2) Termination. In the ridge line following, the algorithm checks the local maxima of the new point \((i_n, j_n)\). If there is no local maxima, then the segments having extremes \((i_c, j_c) (i_n, j_n)\) form angles less than the threshold value with the direction \(\phi_c\), could be found in \(\Omega\). According to this decisive factor the ridge line following process stops independently on the gray level of the current region, and the algorithm can work both on saturated regions and on contrast-deficient
regions with no need for a particular tuning. This event originates a ridge ending minutia.

3) Intersection. A ridge line already examined has been intersected. During the ridge line following process, the algorithm checks whether the new point \((i_n, j_n)\) has already been labeled. In this case the point \((i_n, j_n)\) belongs to two ridge lines (i.e., it is an intersection point). This event can occur either when the current ridge line and the intersected ridge line form a bifurcation minutia.

4) Excessive bending. The ridge line local direction is defined as the average of the directions of the segments \((i_c, j_c) (i_n, j_n)\) relative to the last \(k\) steps \((k = 2, \ldots, 4)\). Then the segment delimited by \((i_c, j_c) (i_n, j_n)\) forms with the ridge line local direction an angle greater than the threshold value. This factor allows for the ridge line following process to be stopped when the ridge line direction changes suddenly. Due to the ridge line continuity, excessive bending always denotes an error in the ridge line following process.

3.3.5. False Minutiae Removal method

The method for extracting the entire ridge lines in the fingerprint image and, consequently, detecting all the minutiae is now defined. The main problems arise from the complexity of examining each ridge line only once and locating the intersections with the ridge lines already extracted. This method uses an auxiliary image \(T\) of the same dimension as \(I\). \(T\) is initialized by setting its pixel values as 0. When ever a new ridge line is extracted from \(I\), the pixels of \(T\) corresponding to the ridge line are labeled by assigning them an identifier. The pixels of \(T\) relevant to a
ridge line are the pixels belonging to the polygonal, e-pixels thick, which links the consecutive maximum points \((i_n, j_n)\) located by the ridge line following algorithm on the ridge line. The algorithm finds minutia (ridge ending and bifurcation) by following the ridge line nearest to the starting point \((i_1, j_1)\) in both directions and stops for the above events.

Typically, the fingerprint image may be corrupted by various kinds of noise, such as creases, smudges, holes, etc. Even with the existence of such noise, a trained fingerprint expert is often able to correctly identify the minutiae by using various visual clues such as local ridge/valley orientation, ridge/valley continuity, etc. Here the proposed feature extraction algorithm can rely on these visual clues to remove false minutia. This is achieved through projection in the same orientation of the ridge lines, after they terminate or intersect other ridge lines (minutiae detection) up to the thresh count. In the case of termination, if the projection finds another termination with in the thresh count, then the both ridge endings may be consider as false minutia. In the case of intersect, if the projection finds another intersection with in the thresh count, with another ridge line then the both ridge bifurcations may be consider as false minutia. This proposed method is efficiently avoiding such false minutiae by Effective False Minutiae Removal algorithm and the average computational time of our approach is considerably lower than the other approaches.

3.4 A Position Invariant Fingerprint Feature Extraction in DCT Domain - PIDCT.

This method proposes a position invariant fingerprint features extraction in DCT Domain. Usually, the fingerprint features contained in a fingerprint image, such
as ridge line patterns and minutiae point, can be extracted from the DCT domain, and used for fingerprint matching. To produce the fingerprint features from the curve-scanned DCT coefficients, first the gray-scale fingerprint image is cropped to the size of 64x64 pixels, where the reference point was at the center of the cropped image [82]. To make the fingerprint feature extraction as position invariant the proposed PIDCT method fixes the core point through the orientation algorithm, and marking the core point as the reference point, then the cropping is done in the reference point. According to the results, the proposed method outperforms the others, particularly in position invariant features extraction and processing.

Generally, the image-based fingerprint feature extraction approach is considered to be one of several effective approaches to automatic fingerprint matching [36]. This method not only achieves a high recognition rate, but also requires lower computational effort than most fingerprint recognition methods proposed to date [78]. In this method, the fingerprint features extraction is based on the curve-scanned DCT coefficients method [82]. Theoretically, by creating features from appropriate DCT coefficients, those informative features can subsequently be used for fingerprint matching purpose. Fingerprint Features Extraction Using Curve-scanned DCT Coefficients [82] method extracting ridge line patterns and minutiae points from the DCT domain, for fingerprint authentication. By considering the oscillate pattern contained in the top-left corner of DCT coefficients, and those coefficients can be divided in curve-scanned fashion, extract DCT features from the divided DCT coefficients, and use them for verification.
The advantage of using the DCT for fingerprint features extraction is its speed. Hence DCT is better suited in extracting informative features than the other methods [127]. In [127] the feature extraction is done by dividing all DCT coefficients containing the oscillate pattern inside in zigzag-scanned fashion, extract DCT features from the divided DCT coefficients, and use them for fingerprint features extraction. Therefore, all DCT coefficients in each non-overlapping image were divided into 12 areas and each feature are extracted from each area, generating 12 features for each non-overlapping image. However, the informative features used for authentication purpose exist within the low frequency area (top-left corner) of the distribution plane only. From the study, magnitude of most features extracted from the middle and high frequency areas (around 75%) of DCT coefficients, from different fingerprint images, are hardly changed [82]. For this reason, the energy compactness area is bound to be used for creating the fingerprint features by a white arch. This boundary is really defined by the oscillated pattern contained within the top-left corner of the distribution plane of the DCT coefficients, which disperses equally in all directions from the DC component. The DCT coefficients divided in this fashion are thus referred to as curve-scanned DCT coefficients [82]. The area within the boundary is approximately 25% of the distribution plane [82].

3.4.1. Proposed Feature Extraction method

The Fingerprint images are typically acquired using a contact-based sensor wherein a user place there finger on the surface of the sensor. The elastic nature of the human skin, coupled with the non-uniform pressure applied by the finger on the sensor, will result the fingerprint images with different positions. Hence the
Fingerprint position is depending upon several parameters including the orientation of the sensor with respect to the finger, the amount of pressure applied, the motion of the finger prior to its placement on the sensor, the moisture content of the skin (dry, oily or wet), the elasticity of the skin, etc. Hence the proposed method fixes the core point through the orientation algorithm, and marking the core point as the reference point, then the cropping is done in the reference point. Now Features Extraction Using Curve-scanned DCT Coefficients [82] method becomes positional invariant.

To extract the position invariant fingerprint features from the curve-scanned DCT coefficients, the following steps are performed sequentially:

A. Find the core point of the fingerprint image.

B. Crop the fingerprint image into the size of 64×64 pixels using its core point as the center.

C. Extract the fingerprint features from the DCT coefficients,

Locating the Core Point

Fingerprints have many obvious landmark structures and a combination of them could be used for establishing a reference point. Here the reference point of a fingerprint may be the core point (the point of maximum curvature of the concave ridges in the fingerprint image). The previous methods to determination of the core point are critically relied on the local features like Poincaré index or some other related properties of the orientation field. While these methods perform well in good quality fingerprint images, they fail to appropriately localize the core point in poor
quality fingerprints with cracks and scars, dry skin, or poor ridge and valley contrast. Hong and Jain have attempted to combine the orientation field information with available ridge details in a fingerprint image [147]. Still, this method does not consistently handle poor quality fingerprints when the orientation field is very noisy and can be misled by poor structural cues in the existence of finger cracks. In order that the core point detection algorithm handles local noise in a poor quality fingerprint image, the detection should necessarily consider a large neighborhood in the fingerprint image. Alternatively, for an accurate localization of the core point, the method should be sensitive to the local variations in a small neighborhood. To meet these contradictory requirements of an accurate and reliable localization, the method of the core point determination based on multiple resolution analysis of the orientation fields [94] is used.

This method locates the core point more precisely than the algorithm proposed by Hong and Jain [147] has the following steps.

1) Divide the input image, into non-overlapping blocks of size 8×8.

2) Compute the gradients \( \partial_x(i, j) \) and \( \partial_y(i, j) \) at each pixel \((i, j)\). Depending on the computational requirement, the gradient operator may vary from the simple Sobel operator to the more complex Marr-Hildreth operator [148].

3) Calculate the local orientation of each block centered at pixel \((i, j)\) by

\[
O(i, j) = \frac{1}{2} \tan^{-1} \left( \frac{V_x(i, j)}{V_y(i, j)} \right)
\]  

(3-9)
Where, 

\[
V_r(i, j) = \sum_{u=-4}^{i+4} \sum_{v=-4}^{j+4} (\partial_x^2(u, v) - \partial_y^2(u, v))
\]

The value of \( O(i, j) \) is the least square estimate of local ridge orientation in the block centered at pixel \((i, j)\). It represents the direction that is orthogonal to the dominant direction of the Fourier spectrum of the 8x8 window.

4) Smooth the orientation field in a local neighborhood. To perform smoothing (low pass filtering), the orientation image needs to be converted into a continuous vector field, which is defined as

\[
\Phi_x(i, j) = \cos(2O(i, j))
\]
\[(3-10)\]

and

\[
\Phi_y(i, j) = \sin(2O(i, j))
\]
\[(3-11)\]

Where, \( \Phi_x \) and \( \Phi_y \), are the x and y components of the vector field, correspondingly.

With the resultant vector field, the low pass filtering can be performed as

\[
\Phi_x(i, j) = \sum_{u=-w/2}^{w/2} \sum_{v=-w/2}^{w/2} W(u, v) \Phi_x(i-wu, j-wv)
\]
\[(3-12)\]

and

\[
\Phi_y(i, j) = \sum_{u=-w/2}^{w/2} \sum_{v=-w/2}^{w/2} W(u, v) \Phi_y(i-wu, j-wv)
\]
\[(3-13)\]

Here \( W(.) \) is a two dimensional low pass filter with unit integral and \( w \times w \) representing the filter size. Note that smoothing operation is performed at the block...
level. Here for experimentation a 5×5 mean filter is used. The smoothed orientation field \( O(i, j) \) is calculated as

\[
O(i, j) = \frac{1}{2} \tan^{-1} \left( \frac{\Phi_y(i, j)}{\Phi_x(i, j)} \right)
\]

(3-14)

5) Initialize the label image \( R \), to use to point out the core point.

6) For each pixel \((i, j)\) in \( O \), calculate the Poincare index and assign the corresponding pixels in \( R \). The Poincare index at pixel \((i, j)\) is enclosed by a digital curve, which consists of a sequence of pixels that are on or within a distance of one pixel apart from the corresponding curve, is computed as:

\[
Poincare(i, j) = \frac{1}{2\pi} \sum_{k=0}^{2} \Delta(k)
\]

(3-15)

\[
\Delta(k) = \begin{cases} 
\delta(k), & \text{if } (|\delta(k)| < \frac{\pi}{2}) \\
\pi + \delta(k), & \text{if } (\delta(k) < -\frac{\pi}{2}) \\
\pi - \delta(k), & \text{otherwise}
\end{cases}
\]

(3-16)

\[
\delta(k) = O(i', j') - O(i, j) \\
i = (i + 1) \mod(8) \\
j = (j + 1) \mod(8)
\]

Where \( i' \) and \( j' \) is the x and y co-ordinates of the closed digital curve with 8 surrounding pixels of 3×3 mask.

7) Get the connected components in \( R \). If the area of a connected component is greater than seven, a core is detected at the centroid of the connected component.
If the area of a connected component is greater than 20, two cores are detected at the centroid of the connected component.

8) If two cores are detected, the center is fixed as the coordinates of the core point with the least y value (the upper core). If only one core is located, the center is assigned to the co-ordinates of the core point.

9) If no core point is detected, calculate the covariance matrix of the vector field in a local neighborhood (q x q) of each point in the orientation field. A feature image F with the biggest eigenvalue of the covariance matrix for each element in the orientation image is identified. A core is located at the centroid of the largest connected component in the threshold image of F and the center is assigned to the co-ordinates of the core.

Feature Extraction

After locating the core point, the gray-scale fingerprint image is cropped to the size of 64×64 pixels, where the reference point (core point) was at the center of the cropped image. Then the cropped image was quartered to get four non-overlapping images of size 32×32 pixels. After that, the DCT was applied to each nonoverlapping image. The DCT coefficients within the defined boundary were divided into 5 areas in the curve-scanned fashion, where each area was 2-pixel width. Then the standard deviation of the DCT coefficients in each area from all 4 non-overlapping images was calculated to create a feature vector of the length 20 (5 features from each non-overlapping image).
Principally, a fingerprint image can be viewed as a collection of white and black lines, which normally causes oscillatory patterns in the mid-frequency band. The energy distribution over the middle scales in the frequency domain can hence be considered as useful features and used for fingerprint pattern classification. In addition, this method has the ability to resolve the variations in position, scale and rotation angle, thro registering the finger print images with respect to a reference point, e.g. core point, which is consistently detected in the fingerprint. Now the features extraction becomes positional invariant.

3.5 An Adaptive Frequency Selection in Fixed Length Square Finger Code Method of Fingerprint Feature Extraction

Here a novel method is presented for Selection of an Adaptive fingerprint ridge Frequency in fixed length square finger code method for fingerprint feature extraction. It uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. For extracting features from gray scale image, fingerprint image is cropped in the size of 128 ×128 pixels using its core point as the center. Then the frequency of the filter is adaptively fixed based on the average inter-ridge distance in the fingerprints. This reveals that by setting the parameters to appropriate values, the finger code generated is more efficient and suitable than conventional methods for a small-scale fingerprint recognition system.

The smooth flow pattern of ridges and valleys in a fingerprint image can be viewed as an oriented texture field [149]. The image intensity surface in fingerprint
image is comprised of ridges whose direction varies continuously, which constitute an oriented texture. Most textured images have limited range of spatial frequencies, and mutually distinct textures vary significantly in their dominant frequency. Textured regions possessing different spatial frequency, orientation, or phase can be simply discriminated by decomposing the texture in several spatial frequency and orientation channels.

Fingerprints are identified by means of quantitative measures associated with the flow of patterns (oriented texture) as features. The characteristics of the Gabor filter, particularly the frequency and orientation representations, are similar to those of human visual system. Therefore, Gabor filter based features have been successfully and broadly applied to texture segmentation [139], face recognition [141], handwriting recognition [142], and fingerprint feature extraction [69].

To generate the Gabor filter-based finger code from the fingerprint image following steps are performed sequentially:

1. Find the core point of the fingerprint image.
2. Crop the fingerprint image into 128×128 pixels using its core point as the center.
3. Set the Gabor filter frequency \( f \) to the reciprocal of the inter-ridge distance.
4. Divide the cropped image into a set of 8×8 non overlapping blocks and sample the fingerprint image in eight different directions using a bank of Gabor filters (eight directions are required to completely capture the local ridge characteristics in a fingerprint while only four directions are required to capture the global configuration [40]).
5) Compute the average absolute deviation from the mean which is referred as 16×16× 4 finger code.
3.5.1. Locating the Core Point

1) Divide the input image, into non-overlapping blocks of size 8 × 8.

2) Compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel $(i, j)$.

3) Calculate the local orientation of each block centered at pixel $(i, j)$.

4) Smooth the orientation field in a local neighborhood.

5) Initialize the label image $R$, to use to point out the core point.

6) For each pixel $(i, j)$ in $O$, calculate the Poincare index and assign the corresponding pixels in $R$.

7) Get the connected components in $R$. If the area of a connected component is greater than seven, a core is detected at the centroid of the connected component.

3.5.2. Normalization

Prior to decomposing the fingerprint image $I(x, y)$, normalize the region of interest $N_i(x, y)$ in each sector block separately to a constant mean and variance. Normalization is made to remove the effects of sensor noise and finger pressure differences. Let $I(x, y)$ denote the gray value at pixel $(x, y)$, $\mu_i$ and $\sigma_i^2$, the estimated mean and variance of the sector block $S_i$ correspondingly and $N_i(x, y)$, the normalized gray-level value at pixel $(x, y)$. For all the pixels in sector $S_i$, the normalized image is
Here \( M_0 \) and \( V_0 \) are the desired mean and variance values, respectively.

Normalization is a pixel-wise operation that doesn’t vary the precision of the ridge and furrow structures. If normalization is made for the entire image, then it cannot compensate for the intensity variations in the different parts of the finger due to finger pressure differences. Normalization of each sector separately alleviates this problem.

In the proposed algorithm, both \( M_0 \) and \( V_0 \) were set to a value 100.

3.5.3. Extraction of Gabor Features-General form of a 2D Gabor filter

The ridge and furrow structures of a fingerprint image are very much accentuated by applying appropriately tuned Gabor filters. These accentuated ridge and furrow structures comprise an efficient representation of a fingerprint image.

The common form of a 2D Gabor filter is defined as

\[
g(x, y, f, \theta_s, \alpha, \beta) = \exp \left[ -\frac{1}{2} \left( \frac{x'^2}{\alpha^2} + \frac{y'^2}{\beta^2} \right) \right] \times \exp(2\pi f x_0) \tag{3-18}
\]

where,

\[
x_0 = x \cos \theta_s + y \sin \theta_s, \\
y_0 = x \cos \theta_s - y \sin \theta_s
\]

\( \theta_s = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ \text{ and } 157.5^\circ \) (with respect to the X axis) and

\( f \) is the frequency of the sinusoidal plane wave along with the direction \( \theta_s \), \( \theta_s \) is the orientation of the Gabor filter, \( \partial_x \) and \( \partial_y \) specify the Gaussian envelope along with X and Y axes, which determines the bandwidth of the Gabor filter.
To consider the Gabor filter in terms of the even symmetric and odd symmetric can be expressed in complex form as

\[ g(.) = g_{\text{even}}(.) + j g_{\text{odd}}(.) \]

where,

\[
\begin{align*}
  g_{\text{even}}(x, y, f, \theta, \sigma_x, \sigma_y) &= \exp \left[ -\frac{1}{2} \left( \frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right) \right] \times \cos(2\pi fx\theta) \\
  g_{\text{odd}}(x, y, f, \theta, \sigma_x, \sigma_y) &= \exp \left[ -\frac{1}{2} \left( \frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right) \right] \times \sin(2\pi fx\theta)
\end{align*}
\]

A fingerprint image can be decomposed into four component images using eight different values of \( \theta_k = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ \text{ and } 157.5^\circ \).

3.5.4. Adaptive Ridge Frequency Estimation

If the Gabor filter frequency \( f \) is too large, false ridges may be created in the filtered image, whereas if \( f \) is too small, nearby ridges may be merged into one. In the proposed algorithm, the Gabor filter frequency \( f \) is set to the reciprocal of the inter-ridge distance because most local ridge structures of fingerprints come with well defined local frequency and orientations.

Let \( f_{a,v}(x) = a \sin(vx) \) be a sinusoidal signal of amplitude \( a \in \mathbb{R} \) and frequency \( v \in \mathbb{R}^+ \) then for each \( n \in \mathbb{R}^+ \) such that \( (n,v) \in \mathbb{Z}^+ \) and for each \( g = 0,1,\ldots,\infty \)

\[
\Gamma^g(f_{a,v}) = \frac{1}{n} \int_0^n \left| \frac{d^g f_{a,v}(x)}{dx^g} \right| dx = a v^g
\]  

(3-20)
the frequency \( \nu \) can be calculated as

\[
\nu = \frac{\Gamma^\nu + 1(\sigma_\nu)}{\Gamma^\nu(\sigma_\nu)} \quad g = 0, 1, \ldots, \infty
\]  

(3-21)

In practice, the above method [150] requires the choice of a valid \( n \) (satisfying the condition \( (n, \nu) \in \mathbb{Z}^+ \)).

3.5.5. Feature Extraction

The bandwidth of the Gabor filters is determined by \( \sigma_x \) and \( \sigma_y \). If the values of \( \sigma_x \) and \( \sigma_y \) are too large, the filter is more robust to noise, but is possible to smooth the image to the extent that the ridge and furrow details in the fingerprint are lost. On the other hand, if they are too small, the filter is not effectively removing noise. Here, the values of \( \sigma_x \) and \( \sigma_y \) were empirically determined and both were set to 4.0.

After fixing all the parameters of the Gabor filters, the magnitude Gabor feature, the even and the odd Gabor feature at sampling point \((X, Y)\) are calculated as follows

\[
G_{mag}(X, Y, \theta_k, f, \sigma_x, \sigma_y) = \left| \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} N_i(X + x, Y + y) \times g(x, y, f, \theta_k, \sigma_x, \sigma_y) \right| \tag{3-22}
\]

\[
G_{even}(X, Y, \theta_k, f, \sigma_x, \sigma_y) = \left| \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} N_i(X + x, Y + y) \times g_{even}(x, y, f, \theta_k, \sigma_x, \sigma_y) \right| \tag{3-23}
\]

\[
G_{odd}(X, Y, \theta_k, f, \sigma_x, \sigma_y) = \left| \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} N_i(X + x, Y + y) \times g_{odd}(x, y, f, \theta_k, \sigma_x, \sigma_y) \right| \tag{3-24}
\]

Where \( N_i(., .) \) represents a sector block of normalized fingerprint image \( I(x, y) \) of size \( M \times N \), and having 256 gray-levels. The magnitude Gabor features at the
sample point and its neighboring points within three pixels are similar, while the others are not. This is because the magnitude Gabor feature has the shift-invariant property.

3.5.6. Finger Code Generation:

To compute the average absolute deviation from the mean which is referred as 16×16×4 finger code, following steps are performed sequentially.

Now, ∀ i ∈ {1, 2, ..., 256} and θ_i ∈ {0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°},

The feature values are the average absolute deviation from the mean is defined as

\[
F_{iθ} = \frac{1}{n_i} \left( \sum_{n_i} |F_{iθ}(x, y) - P_{iθ}| \right)
\]

where,

\( n_i = 64, \text{ is the number of pixels in the block of 8×8,} \)

\( P_{iθ} \text{ is the mean of pixel values in that block.} \)

Thus, the average absolute deviation of each 8×8 block of the eight filtered images defines the components of the finger code (16×16×4).

By setting the parameters to appropriate values, the finger code generated is more efficient and suitable than conventional methods for a small-scale fingerprint recognition system. Since, the fingerprint matching is based on the Euclidean distance between the two corresponding finger codes, it is extremely fast.