Chapter 4
Fingerprint Minutiae Extraction

Automatic Fingerprint Recognition Systems accept or reject the user’s identity by matching against an existing fingerprint database. Since accurate matching of fingerprints depends largely on local ridge features known as minutiae, marking minutiae accurately and rejecting false ones is critically important. A vital step in automatic fingerprint matching is to reliably extract minutiae from the input fingerprint images. Minutiae points (Figure 4-1) are the local ridge discontinuities, which are of two types: ridge endings and ridge bifurcations. It is these minutiae points which are used for determining uniqueness of a fingerprint.

![Figure 4-1: Minutiae Points (a) Ridge ending   (b) Ridge bifurcation](image)

The major goal of this thesis is to secure fingerprint images using watermarking. In the process, firstly the fingerprint images are enhanced as discussed in the previous chapter and then, their minutiae points are extracted so that they can be preserved later in
the watermarking stage. By preserving the minutiae features during watermarking, the accuracy of fingerprint identification system is maintained.

This chapter begins by introducing minutiae extraction in fingerprint images followed by some related work in this area. Section 4.3 presents a brief account of the morphological operators used in image processing. Section 4.4 presents the proposed work for extracting minutiae from fingerprint images using the Hit or Miss Transform of mathematical morphology. Finally, section 4.5 presents the experimental results and analysis.

4.1 Introduction to Minutiae Extraction

An accurate representation of the fingerprint image is critical to automatic fingerprint identification systems, because most deployed commercial large-scale systems are dependent on feature-based matching. Among all the fingerprint features, minutia point features with corresponding orientation maps are unique enough to discriminate amongst fingerprints. The minutiae feature representation reduces the complex fingerprint recognition problem to a point pattern matching problem. Apart from enhancing fingerprint images as mentioned earlier, fingerprints are also preprocessed to enhance the original ridge flow pattern without altering the singularity, join broken ridges, clean artifacts between pseudo-parallel ridges, and not introduce false information. Finally minutiae extraction is done to efficiently and accurately mark the minutiae points.

4.2 Related Work

Once the fingerprint image is enhanced minutiae points can be extracted. There are a lot of minutiae extraction methods available in the literature (Bansal, Sehgal & Bedi, 2011). We can classify these methods broadly into two categories (Figure 4-2):

- Based on gray scale fingerprint images
- Based on binarized fingerprint images
4. Fingerprint Minutiae Extraction

**Based on gray-scale fingerprint images:** Based on the observation that a ridge line is composed of a set of pixels with local maxima along one direction, Maio and Maltoni proposed to extract the minutiae directly from the gray-level image by following the ridge flow lines with the aid of the local orientation field (Maio and Maltoni, 1997; 1998). On similar lines other methods were also proposed (Jinxiang, Zhongyang and Luk, 2000; Jiang, Yau and Ser, 2001). These methods attempt to find a local maximum relative to the cross-section orthogonal to the ridge direction. Ratha et al (Ratha, Chen and Jain, 1995) proposed a minutiae extraction algorithm in which the flow direction of ridges is computed by viewing the fingerprint image as a directional textured image. They used a ridge segmentation algorithm based on a waveform projection to accurately locate the ridges and a thinned ridge image is obtained. Finally the minutiae are extracted from the thinned ridges based on the number of crossings and a post processing step applied to remove spurious minutiae.

Some fuzzy techniques have also been suggested in literature (Sagar, Ngo and Foo, 1995; Sagar and Beng, 1999), to extract minutiae from gray scale images directly. In these techniques, it is proposed that a gray scale image consists of two distinct levels of
gray pixels i.e., the darker pixels, constituting the ridges and the lighter pixels, constituting the valleys and furrows. Using human linguistics, these two levels of gray can be described as DARK and BRIGHT levels and modeled using fuzzy logic along with appropriate fuzzy rules to extract minutiae accurately.

**Based on binarized fingerprint images:** Most of the techniques in literature are based on extracting minutiae from binarized images. These can be further classified into two classes, those that work on unthinned binarized images and those that work on thinned binarized images. Techniques that work on unthinned binarized images are based on chaincode processing (Zhixin and Govindaraju, 2006), run based methods (Zenzo, Cinque and Levialadi, 1996; Hwan Shin, Hwang and Chien, 2005). Chaincode representation of object contours is extensively used in document analysis. Unlike thinned skeletons, the pixel image can be fully recovered from the chaincode of its contour. In this method, the image is scanned from top to bottom and right to left. The transitions from white (background) to black (foreground) are detected. The contour is then traced counterclockwise and expressed as an array of contour elements (Zhixin et al, 2006). The objective is to attain the ridge orientation for the entire window rather than at every pixel. Run based methods result in fast extraction of fingerprint minutiae that are based on the horizontal and vertical run-length encoding from binary images without a computationally expensive thinning process (Zenzo et al, 1996; Hwan Shin et al, 2005). Fingerprint images are represented by a cascade of runs after run-length encoding. Then runs' adjacency is checked and characteristic runs representing minutiae are detected by introducing some geometric constraints for checking their validity.

However, most fingerprint minutia extraction methods are thinning-based where the skeletonization process converts each ridge to one pixel wide. Minutia points are detected by locating the end points and bifurcation points on the thinned ridge skeleton based on the number of neighboring pixels. The most common technique in this category whose variations exist in the literature is the crossing number method (Xiao and Raafat, 1991; Amengual, Juan, Prez, Prat, Sez and Vilar, 1997; Tico and Kuosmanen, 2000; Akram, Tariq and Khan, 2008). This method is favored over other methods for its
computational efficiency and inherent simplicity. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3×3 window. However, this method and most of its variations extract a very large number of spurious minutiae together with true ones, thereby relying heavily on enormous post processing procedures including one by one minutia validation and invalidation. The presented work uses a morphology based approach to extract minutiae from thinned binarized fingerprint images. The proposed algorithm results in efficient minutiae detection, thereby saving a lot of effort in the post processing stage.

4.3 Morphological Operators in Image Processing

Mathematical morphology refers to a branch of nonlinear image processing and analysis that concentrates on the geometric structure within an image. It is considered to be a powerful tool to extract components from images that are useful in representation and description of region shape such as boundaries, skeletons and convex hull. Morphological techniques are useful in pre or post processing such as thinning. Mathematical morphology finds its place in computer vision as it emphasizes on shape information. It is mathematical in the sense that the analysis is based on set theory, topology, lattice, random functions, etc. We briefly discuss some mathematical morphology transformations, used in this thesis, (Gonzalez et al, 2005) in this section.

4.3.1 Erosion and Dilation

Erosion and dilation are considered the fundamental morphological operations. Erosion shrinks or thins an object in a binary image where as Dilation grows or thickens objects. Most of the complex morphological algorithms are based on these two primitive operations. Erosion and Dilation are defined as follows:

Let \( A: z^2 \rightarrow z \) be an image and \( B: z^2 \rightarrow z \) be a structuring element. The erosion of \( A \) by \( B \) denoted by \( (A \ominus B) \), is expressed as follows by Equation (4-1):
\[(A \ominus B) = \{z \mid (B)_z \cap A^c \neq \emptyset \} \quad (4-1)\]

In other words, erosion of \(A\) by \(B\) is the set of all points \(z\) such that \(B\), translated by \(z\), is fully contained in \(A\).

The dilation of \(A\) by \(B\), is denoted by \((A \oplus B)\), and is expressed as follows by Equation (4-2):
\[(A \oplus B) = \{z \mid (B)_z \cap A \neq \emptyset \} \quad (4-2)\]

In other words dilation of \(A\) by \(B\) is the set of all structuring element origin locations where the reflected and translated \(B\) overlaps at least some portion of \(A\).

### 4.3.2 Opening and Closing

Closing and opening are two examples of elementary combinations of erosion and dilation. Visually, opening smoothes contours, break narrow isthmuses and eliminates small islands. The opening of \(A\) by \(B\), denoted by \((A \circ B)\), is simply erosion of \(A\) by \(B\), followed by dilation of the result by \(B\) as follows:
\[(A \circ B) = (A \ominus B) \oplus B \quad (4-3)\]

On the other hand, closing smoothes the contours, fills narrow gulfs and eliminate small holes. The closing of \(A\) by \(B\), denoted by \((A \bullet B)\) is a dilation followed by erosion, as follows:
\[(A \bullet B) = (A \oplus B) \Theta B \quad (4-4)\]

### 4.3.3 The Hit or Miss Transformation (HMT)

The morphological Hit or Miss Transform is the localization operator in mathematical morphology. It finds occurrences of an object and its nominal surroundings in a set or an image. It is a natural operation to select out pixels that have certain geometric properties, such as corner points, isolated points or border points and performs template matching.

The Hit or Miss Transformation of \(A\) by \(B\) is denoted by \((A \odot B)\). Here \(B\) is a structuring element pair \(B = (B_1, B_2)\), rather than a single element as before. The hit and
miss transformation is defined in terms of these two structuring elements using Equation (4-5) as follows:

\[ A \ominus B = (A \ominus B_1) \cap (A^c \ominus B_2) \]  

(4-5)

\( B_1 \) is the set formed from elements of \( B \) associated with an object and \( B_2 \) is the set of elements of \( B \) associated with the corresponding background. Thus, erosion with \( B_1 \) determines the location of foreground pixels and erosion of the complement with \( B_2 \) determines the location of the background pixels. Performing intersection of these two operations outputs an image which consists of all locations that match the pixels in \( B_1 \) (a hit) and that have none of the pixels in \( B_2 \) (a miss). In the proposed work for minutiae extraction, HMT has been used as a tool for spur removal, bridge removal, thinning and minutiae detection.

### 4.4 Extracting true fingerprint minutiae based on the Hit or Miss Transform

The proposed minutiae extraction algorithm extracts minutiae using the morphological Hit or Miss Transform (Bansal, Sehgal & Bedi, 2010). But, the success of any minutiae extraction technique depends on the quality of the input image; hence the fingerprint image is enhanced before processing it for minutiae extraction. Figure 4-3 shows the schematic diagram of the proposed technique. The details of the various steps are explained in the following subsections:

#### 4.4.1 Fingerprint Image Preprocessing

Fingerprint Image Preprocessing performs some operations on the input fingerprint to prepare it for minutiae extraction. These operations are as follows:

- Fingerprint image enhancement
- Binarization
- Morphological preprocessing
- Thinning
Fingerprint image enhancement

The performance of any minutiae detection algorithm relies heavily on the quality of the input fingerprint image. If a fingerprint image is of low quality, a large number of false minutiae are extracted during minutiae extraction. Hence, it is important to enhance the input fingerprint before performing minutiae extraction. The input fingerprint image is enhanced using the methods proposed earlier using type-2 fuzzy logic (Chapter 3, Sections 3.3 and 3.4).

The Hong’s algorithm (Hong et al, 1998), which is based on the convolution of the image with Gabor filters tuned to the local ridge orientation and frequency has been used for ridge enhancement of the input fingerprint image. This algorithm firstly, segments the image to extract it from the background. Next, the image is normalized so that it has a prespecified mean and variance. After calculating the local orientation and ridge frequency around each pixel, the Gabor filter (Alonso-Fernandez, Fierrez-Aguilar
and Ortega-Garcia, 2005) is applied to each pixel location in the image. As a result the filter enhances the ridges oriented in the direction of local orientation.

**Binarization**

Image binarization converts a 256 gray level image to a binary image (i.e. with only two levels – black and white as shown in Figure 4-4. The simplest way to use image binarization is to choose a threshold value, and classify all pixels with values above this threshold as white, and all other pixels as black. The problem is how to select the correct threshold. In many cases, finding one threshold compatible to the entire image is very difficult, and in many cases even impossible. Therefore, adaptive image binarization is needed where an optimal threshold is chosen for each image area (Otsu, 1979). There are other methods also available for image binarization (Trier and Taxt, 1995; Yuheng and Qinghan, 2006).

![Figure 4-4: Original and Binarized images](image-url)

**Morphological Preprocessing**

After close examination of the binarized image, it can be seen that the broken ridges 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 which cleans up any parasitic components that may have been left as a result of thinning. The spur operator is shown in Figure 4-5.

![Figure 4-5: Spur removal](image)

2. Spurious bridge removal

Some linked parallel valleys may be separated to eliminate spurious bridges (Figure 4-6) in the skeleton image.

![Figure 4-6: (a) and (b) Spurious bridges removal](image)

3. Spurious holes removal

Spurious holes are regions (Figure 4-7) with an area (number of pixels) below a threshold \( w_1 \). 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.
4 Isolated islands removal

Islands (Figure 4-8) are short lines with an area below a threshold. Removing these areas eliminates any spurious dots and islands from the binarized image.

**Figure 4-8: Islands removal**

**Thinning**

The next step is to thin the processed binary image using the morphological thinning operation. The thinning algorithm removes pixels from ridges until the ridges are one pixel wide (Espinosa, 2002). The thinning of an image $I$ by a structuring element $J = (J_1, J_2)$ is given by Equation (4-6) as follows:

$$\text{Thin}(I, J) = I - (I \odot J) \quad (4-6)$$

where, the subtraction is the logical subtraction defined by:

$$X - Y = X \cap Y^c \quad (4-7)$$
Figure 4-9: (i) to (viii) The structuring element sequence $J_1 = (J_1^1, J_1^2, J_1^3, J_1^4, J_1^5, J_1^6, J_1^7, J_1^8)$. (ix) to (xvi) The structuring element sequence $J_2 = (J_2^1, J_2^2, J_2^3, J_2^4, J_2^5, J_2^6, J_2^7, J_2^8)$.

Figure 4-10: (a), (c) and (e) Thinning results on binarized images (b), (d) and (f) Thinning results on the corresponding morphologically processed binarized images
The operation is repeatedly applied until it causes no further changes to the image (i.e., until convergence). The structuring element sequence $J$ used by us is shown in Figure 4.9. Structuring elements from (i) to (viii) show the sequence for $J_1$ and from (ix) to (xvi) show the sequence for $J_2$ (complement of $J_1$). The image is thinned using the structuring element pairs $(J_1^i, J_2^i)$, $i = 1...8$, in sequence. Doing so produces a connected skeleton of the image. The process is repeated in cyclic fashion until the operation produces any further change in the image. Figure 4-10 shows that the thinning results after morphological preprocessing are better than without it.

### 4.4.2 Minutiae Extraction Using HMT

In this step, we shall extract the two basic types of minutiae points (ridge terminations and ridge bifurcations) from the thinned image obtained from the previous step using the Hit or Miss Transformation. For this purpose, structuring elements were developed for different types of minutiae present in the fingerprint image, to be utilized by the HMT. This detects all the minutiae point but some false minutiae may occur due to some superfluous information in the thinned image. They are removed in the last step of the algorithm. However, it can be seen from the final results in Section 4.5 (Table 4-3), that the number of false minutiae is lesser as compared to other techniques as the image is already preprocessed with morphological operators before thinning.

#### Extracting Ridge Terminations

Ridge endings are those pixels in an image which have only one neighbour in a $3 \times 3$ neighbourhood. The minutiae image $M1$ containing ridge terminations is given by applying Hit or Miss Transform on $I$ by $J$ as follows:

$$ M1 = I \ominus J \quad (4-8) $$

where, $I$ is the thinned image and $J$ is the sequence of structuring element pairs $(J_1, J_2)$. Applying Equation (4-5),

$$ I \ominus J = (I \ominus J_1) \cap (I \ominus J_2) \quad (4-9) $$
The structuring element sequence $J$ that we have designed for determining endpoints in an image is shown in Figure 4-11.

![Figure 4-11: The structuring element sequence](image)

The structuring elements have been designed in such a way so as to select all pixels from image $I$ that have a single neighbour in a $3\times3$ neighbourhood and leave all pixels that have more than one neighbour. Structuring elements from (i) to (viii) show the sequence for $J_1$ and from (ix) to (xvi) show the sequence for $J_2$ (complement of $J_1$). Each of the structuring element pairs $(J_1^i, J_2^i)$, $i = 1...8$, is applied in sequence and the collective output gives the ridge endpoints in the fingerprint image.

**Extracting Ridge Bifurcations**

Ridge bifurcations are those pixels in an image which have only three neighbours in a $3\times3$ neighbourhood and these neighbours are not adjacent to each other. The minutiae image $M_2$ containing ridge terminations is given by:

$$M_2 = I \odot J$$  \hspace{1cm} (4-10)

where, $I$ is the thinned image and $J$ is the sequence of structuring element pairs $(J_1, J_2)$. Applying Equation (4-5),
\[ I \ominus J = (I \ominus J_1) \cap (I^c \ominus J_2) \]  \hspace{1cm} (4-11)

The structuring element sequence \( J \) that we have designed for determining bifurcations in an image is shown in Figure 4-12.

![Figure 4-12: (i) to (viii) The structuring element sequence \( J_1 \) = \( (J_1^1, J_1^2, J_1^3, J_1^4, J_1^5, J_1^6, J_1^7, J_1^8) \). (ix) to (xvi) The structuring element sequence \( J_2 \) = \( (J_2^1, J_2^2, J_2^3, J_2^4, J_2^5, J_2^6, J_2^7, J_2^8) \).](image)

Structuring elements from (i) to (viii) show the sequence for \( J_1 \) and from (ix) to (xvi) show the sequence for \( J_2 \) (complement of \( J_1 \)). The structuring elements are designed as per an actual bifurcation point in a fingerprint image, where each pixel has exactly three neighbours which are not next to each other. Each of the structuring element pairs \( (J_1^i, J_2^i) \), \( i = 1...8 \), is applied in sequence and the collective output gives the ridge bifurcations in the fingerprint image. When the Hit or Miss structuring elements are small, a faster way to compute the HMT is to use a lookup table (Gonzalez et al, 2005).

### 4.4.3 Post Processing

The minutiae set, extracted in the previous step, contains many false or spurious minutiae. The following simple post processing steps remove them.

1. If the distance between two bifurcations is less than a threshold \( T \) and they are in the same ridge, remove both of them (eg. merge, bridge, ladder, lake in Figure 4-13).
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 (break, multiple breaks in Figure 4-13) 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, they are removed.

![Break, Spur, Merge, Break and merge, Multiple breaks, Bridge, Ladder, Lake](image)

**Figure 4-13: Common false minutiae points (black dots)**

### 4.5 Experimental Results and Analysis

In our experimental work, we have tested the above algorithm on FVC (2004) fingerprint database. The database contains 40 different fingers and eight impressions of each finger, which makes it a total of 40×8=320 fingerprints. Minutiae are extracted at four stages from the images as follows and the results are tabulated.

1). Original raw images.
2). Enhanced and then thinned (repairing broken ridges etc.).
3). Preprocessed thinned images (Removal of spurs, spurious bridges, filling holes etc.).
4). Post processing the minutiae to get rid of the spurious ones.

The proposed algorithm has been tested using two quantity measures namely Sensitivity (Equation 4-12) and Specificity (Equation 4-13) which indicate the ability of the algorithm to detect the genuine minutiae and remove the false minutiae for fingerprint image (Tico et al, 2005). The performance of the proposed algorithm has been measured based on the missing and the spurious minutiae after each stage (Figure 4-14).

\[
Sensitivity = 1 - \frac{Missed\ Minutiae}{Ground\ Truth\ Minutiae} \tag{4-12}
\]
4. Fingerprint Minutiae Extraction

\[
\text{Specificity} = 1 - \frac{\text{False Minutiae}}{\text{Ground Truth Minutiae}} \tag{4-13}
\]

Table 4-1 shows the total number of minutiae together with the number of ridge endings and bifurcations separately at each stage. It also shows the reduction in ridge endings and bifurcations after each stage. It further shows the ground truth minutiae in each case. The ground truth minutiae have been calculated by manually inspecting the fingerprint image.

Figure 4-14: (a), (b), (c) Minutiae on enhanced, thinned, but not morphologically preprocessed images I1, I2 and I3 respectively. (d), (e), (f) Minutiae on enhanced, thinned and morphologically preprocessed images I1, I2 and I3 respectively. (g), (h) and (i) show finally post processed minutiae on images I1, I2 and I3 respectively. In all the images endings are shown as x and bifurcations are shown as o.
<table>
<thead>
<tr>
<th>Image</th>
<th>Ground Truth Minutiae</th>
<th>Total Minutiae</th>
<th>Ridge Endings</th>
<th>Ridge Bifurcations</th>
<th>Total Red. (%)</th>
<th>End. Red. (%)</th>
<th>Bif. Red. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Stage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Original</td>
<td>347</td>
<td>201</td>
<td>146</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I1</td>
<td>59(36 + 23)</td>
<td>Enhanced</td>
<td>315</td>
<td>180</td>
<td>135</td>
<td>9.2</td>
<td>10.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preprocessed</td>
<td>215</td>
<td>101</td>
<td>114</td>
<td>31.7</td>
<td>43.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post processed</td>
<td>62</td>
<td>38</td>
<td>24</td>
<td>71.1</td>
<td>62.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Original</td>
<td>114</td>
<td>75</td>
<td>39</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I2</td>
<td>23(12 + 11)</td>
<td>Enhanced</td>
<td>88</td>
<td>56</td>
<td>29</td>
<td>22.8</td>
<td>25.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preprocessed</td>
<td>61</td>
<td>42</td>
<td>19</td>
<td>30.7</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post processed</td>
<td>23</td>
<td>11</td>
<td>12</td>
<td>62.2</td>
<td>73.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Original</td>
<td>201</td>
<td>106</td>
<td>95</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I3</td>
<td>23(9 + 14)</td>
<td>Enhanced</td>
<td>73</td>
<td>30</td>
<td>43</td>
<td>63.7</td>
<td>71.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preprocessed</td>
<td>54</td>
<td>20</td>
<td>34</td>
<td>26.0</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post processed</td>
<td>22</td>
<td>8</td>
<td>14</td>
<td>57.4</td>
<td>60.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Original</td>
<td>360</td>
<td>110</td>
<td>250</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I4</td>
<td>32(13 + 19)</td>
<td>Enhanced</td>
<td>121</td>
<td>52</td>
<td>89</td>
<td>66.3</td>
<td>52.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preprocessed</td>
<td>76</td>
<td>24</td>
<td>52</td>
<td>37.2</td>
<td>53.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post processed</td>
<td>33</td>
<td>14</td>
<td>17</td>
<td>56.6</td>
<td>41.7</td>
</tr>
</tbody>
</table>

Table 4-1: Total minutiae, Ridge endings and Ridge bifurcations with % reduction in false minutiae after each stage.
Table 4-2 lists the total, missed and false number of minutiae before and after the post processing stage together with the sensitivity and specificity values. The high values of sensitivity and specificity in the end, confirm the effectiveness of the proposed technique. Our technique works on thinned binarized images and the most widely used technique applicable to such images in the literature is the crossing number technique. Hence, we have compared both of them and Table 4-3 shows the comparative results of our technique with the much used crossing number technique. It can be clearly seen that the results of the proposed algorithm are better at both before and after post processing stages. This is so because most crossing number techniques rely on complicated and extensive post processing algorithms to improve their result. The results of Table 4-3 are further illustrated by the chart depicted in Figure 4-15. It can be clearly seen that the minutiae points extracted by the proposed technique are closer to the ground truth minutiae as compared to the Crossing Number technique both before post processing and after post processing.

Table 4-2: Total, missed and false number of minutiae before and after the post processing stage together with the Sensitivity and Specificity in the proposed technique

<table>
<thead>
<tr>
<th>Image</th>
<th>Before Post Processing</th>
<th>After Post Processing</th>
<th>Final Sensitivity</th>
<th>Final Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total  Missed False</td>
<td>Total  Missed False</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>I1</td>
<td>215 1 157</td>
<td>62 1 4</td>
<td>98.3</td>
<td>93.2</td>
</tr>
<tr>
<td>I2</td>
<td>61 1 40</td>
<td>23 1 1</td>
<td>95.7</td>
<td>95.7</td>
</tr>
<tr>
<td>I3</td>
<td>64 1 43</td>
<td>22 1 0</td>
<td>95.7</td>
<td>100</td>
</tr>
<tr>
<td>I4</td>
<td>76 2 45</td>
<td>31 2 1</td>
<td>93.6</td>
<td>96.9</td>
</tr>
</tbody>
</table>
Table 4-3: Total minutiae together with endings and bifurcations before and after the post processing stage in the proposed as well as the crossing number technique

<table>
<thead>
<tr>
<th>Stage</th>
<th>Image</th>
<th>Proposed Technique</th>
<th>Crossing Number Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>Endings</td>
</tr>
<tr>
<td>Before Post Processing</td>
<td>I1</td>
<td>215</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>I2</td>
<td>61</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>I3</td>
<td>54</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>I4</td>
<td>76</td>
<td>24</td>
</tr>
<tr>
<td>After Post Processing</td>
<td>I1</td>
<td>62</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>I2</td>
<td>23</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>I3</td>
<td>22</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>I4</td>
<td>33</td>
<td>14</td>
</tr>
</tbody>
</table>

Figure 4-15: Comparative performance of the proposed technique with the Crossing Number Technique with respect to the ground truth minutiae
The results depict the successful usage of the Hit or Miss Transform on thinned fingerprint images for efficient minutiae extraction. It can be clearly seen from all tables that preprocessing a fingerprint image with morphological operators before thinning removes superfluous information. It extracts a clear and reliable ridge map structure from the input fingerprint image thereby giving better results in minutiae extraction. This also reduces a lot of effort in the post processing stage as the number of spurious minutiae is comparatively lesser. The high values of Sensitivity and Specificity in Table 4-2 and the comparative results with the crossing number technique in Table 4-3 illustrate the encouraging performance of our method.