CHAPTER 3

PERSON AUTHENTICATION USING GEOMETRICAL FEATURES OF HAND

3.1 INTRODUCTION

In today's world different types of hand biometric verification systems are available. Some methods are more efficient than others while some are easy to implement. An authentication system based on hand geometry provides a medium level security but has an upper hand over the other methods due to various factors. The first factor is cost-effectiveness. Just by using a platform and a medium resolution CCD camera the hand images can be captured and used for verification. Another advantage is the number of calculations and computations involved. In hand geometry based verification system the algorithms involved are simple and less complex. This in turn favours a faster system. It uses low computational cost algorithms. Proposed hand geometry feature extraction algorithms by researchers have no consideration of inter-class and intra-class variations. A large set of feature values can be obtained, however only those features that are distinct for different users are given a high weight value while others are made insignificant.

The rest of the Chapter is organized as follows. The adopted methodology, block diagram and algorithm are included in Section-3.2 whereas Section-3.3 explains the feature extraction details. Experimental observations, results and comparison with existing methods are reported in Section-3.4 while Section-3.5 concludes the Chapter.

3.2 CODING METHODOLOGY

As discussed in Chapter 2, a biometric algorithm is implemented by implementing all its blocks shown in Figure 2.1. The detailed block diagram of hand biometric system used in this chapter is shown in Figure 3.1.
The block diagram consists of image acquisition, image pre-processing, feature extraction, template generation and finally matching and decision modules. All the modules are explained in following subsections.

Figure 3.1: Basic block diagram of Hand based biometric system.

3.2.1 Image Acquisition

Peg based acquisition devices are costly and user needs to be trained whereas contact free image acquisition require complex algorithms for preprocessing. So, peg free contact based image acquisition method has been adopted. Good scanners should be used for image acquisition. Figure of merit of color scanners are discussed by Sharma (1997). Different methods and devices are developed for enrollment and verification of biometric information by Mohanty (2011). The hand images are acquired using normal document scanner (HP ScanJet G3010) with the resolution of 200dpi and saved in JPEG format. Sample images of JUET database are shown in Figure 3.2. In the figure, images of three subjects are shown. Each row contains five images of same subject. The database is formed using the hand images of various individuals. Due to the lack of any
pegs to align the hand there might be variations in the hand alignment. A small
and slight rotation is acceptable but complete or large rotation is prohibited. The
users are also trained to keep the hand on the scanner in such a way that the arm
forms almost a right angle with the palm [www.security.iitk.ac.in]. This reduces
the chances of a part of the wrist being included in the image. Also for the similar
reason the user is asked to keep his hands on the bottom part of the scanner rather
than the top part. Since in this way a large part of the hand including the wrist will
not get into the picture and spoil the calculation of the hand geometry features like
the area or the perimeter which will be adversely affected in that scenario.

Figure 3.2: Sample Images of JUET database

Two databases, database1 and database 2 are generated and used for performance
analysis and stability analysis respectively. Database1 is generated by scanning
the left hand of 50 individuals (30 males and 20 females of age between 18 and 30
years) in fully stretched mode to avoid the alignment errors. Five images per
person are saved resulting in total 250 images in database. Database 2 comprises of 50 persons (35 male and 15 female of age between 20-35 years), each given 10 images of their left hand so the database consists of 500 images in the similar manner.

![Figure 3.3: Image of the scanner used.](image)

The used scanner is shown in Figure 3.3. The cover of the scanner was kept open during hand image acquisition for uniform illumination of panel. Users have been provided flexibility to place their hand on scanner at any position in a way that no finger should touch the other. The hand is placed vertically and then preprocessed in such way that jewellery (ring, bracelet) and small variation in position of fingers will not affect the results.

### 3.2.2 Image Preprocessing

The acquired images are preprocessed before extracting the features from hand. Image preprocessing includes image resizing, image enhancement, image segmentation. Image processing finds its application in various fields like segmentation of moving data [Singh and Shrivastava (2015)], analysis of cells and tissues for biomedical purposes [Hwee et al. (2015)] and different face recognition algorithms [Singh et al. (2003)]. The first step of preprocessing is resizing the acquired image. The sample acquired image and the resized images are shown in Figure 3.4 and Figure 3.5. The images captured are resized for uniformity to 256X256 square pixel dimensions. Bi-cubic interpolation technique
is used for resizing the acquired images using (3.1). This is required due to use of full screen of scanner. Bi-cubic interpolation technique gives the most accurate results by preserving the fine details which is based on sixteen nearest-neighbours of a point.

In this method the intensity value at \((x, y)\) is obtained by following relation –

\[
g(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j
\]  

(3.1)

where \(a_{ij}\) is the intensity at \(x = i \text{ and } y = j\). The input image to the biometric system is a RGB image. The image is converted into gray scale image. The next process is to binarize the gray scale image such that it is easy to obtain the hand geometry features by the edge of the hand image.

Let, the acquired colored hand image is represented as \(H(x, y, c_i)_{i=1,2,3}\) where \(c_1, c_2, c_3\) are three color channels \((R,G,B)\) and \(x \text{ and } y\) are spatial coordinates and can be represented using (3.2) as follows-

\[
R(x,y) = H(x,y, c_1), G(x,y) = H(x,y, c_2) \text{ and } B(x,y) = H(x,y, c_3)
\]

(3.2)

After image acquisition stage the color images are converted into gray scale images by using (3.3) given below –

\[
G(x,y) = a_1 R(x,y) + a_2 G(x,y) + a_3 B(x,y)
\]

(3.3)
where, \( G(x, y) \) is the grayscale image and \( a_1, a_2, a_3 \) are weighing factor with values 0.2989, 0.5870, and 0.1140 respectively.

The document scanner used for image acquisition may introduce some amount of noise in original acquired hand image which is object and panel as background given in (3.4). This noise will affect the performance of object separation while object is separated with the background. The problem is reduced to image restoration while the gradation is only due to noise as given below

\[
f(x, y) = g(x, y) + (x, y)
\]

where, \( f(x, y), g(x, y) \) and \( (x, y) \) are the degraded image, original gray hand image and the additive noise term respectively. A special filtering operation has been applied for restoring the hand image from the noisy version of image and an order-statistical alpha trimmed filter [Gonzalez and Woods (2009)] as described by (3.5) has been used for this purpose.

\[
f'(x, y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} f_r(s, t)
\]

where, \( S_{xy} \) represent the set of coordinates in a rectangular sub-image window of size \( m \times n \), centered at point \( (x, y) \). Here, \( d/2 \) lowest and \( d/2 \) highest intensity values of \( f(s, t) \) in the neighborhood \( S_{xy} \) are deleted, where \( d \) is number of
pixels and can range from 0 to $mn - 1$. Remaining $mn - d$ pixels are represented by $f_r(s,t)$. In this case the value of $d = mn - 1$ [Gonzalez and Woods (2009)].

Filtered gray scale hand image contains hand image and background. To separate the hand from background, entire image is segmented to have two distinct objects each consisting of two different intensity levels. To achieve this, image segmentation is done by thresholding which will assign 0 intensity to all background pixels and 256 to all the pixels present inside hand using relation

$$A(x,y) = \begin{cases} 
1, & f(x,y) > k \\
0, & f(x,y) < k 
\end{cases}$$

\[ (3.6) \]

![Figure 3.6: Binarized Image](image)

Optimum value of threshold $k$ is evaluated using Otsu’s method (1979), which maximizes the inter-class variance and is based on histogram of the image to be segmented. The gray scale hand image is having 256 intensity levels ($Z = 0 - 255$). Let $n_z$ denotes the number of pixels with intensity $Z$, then-

$$256 \times 256 = n_0 + n_1 + n_2 + n_3 \ldots \ldots + n_{255}$$

The selected threshold $k$ divides the gray scale image in two classes $C_1$ and $C_2$ such that $C_1$ consists of all pixels of image with intensity values in the range $[0,k]$ and $C_2$ consists of the pixels with values in the range $[k+1,255]$. The probabilities of classes $C_1$ and $C_2$ occurrence is found by (3.7) –

$$P_1(k) = \sum_{i=0}^{k} p_i \quad \text{and} \quad P_2(k) = \sum_{i=k+1}^{255} p_i = 1 - P_1(k)$$

\[ (3.7) \]
where, the normalized histogram has components

\[ p_i = \frac{n_i}{256 \times 256} \quad \text{so that} \quad \sum_{i=0}^{255} p_i = 1. \]

Similarly, the mean intensity values \( m_1 \) and \( m_2 \) of both the classes of hand image and the average intensity or global mean of entire hand image \( m_G \) is determined. Between classes variance is given by (3.8) –

\[
\sigma^2_B = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2
\]

The value of \( k \) is selected which maximizes \( \sigma^2_B \). With such value of \( k \), the gray scale hand image is converted to bi-level image as shown in Figure 3.6.

### 3.2.3 Image smoothening and boundary detection

Binary hand image \( A(x,y) \) as shown in Figure 3.6, obtained after image segmentation, contains some thin breaks, minute holes and narrow gaps. Smoothening is required to fuse thin breaks, to eliminate minute holes and to fill the gaps in hand contour. Morphological operations like dilation and erosion have been used for the same. The image is dilated as well as eroded (the two morphological processes). Mathematical morphology is applied using set theory. This is done by using a proper structuring element. Two different structuring elements which are small sets or sub-images have been defined for formulating dilation and erosion [Gonzalez and Woods (2004)].

Dilation grows or thickens objects in a binary image. A flat diamond of size 2 as shown below is used as structuring element for dilation operation.

\[
B_1 = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]
Dilation of $A$ by $B_1$ is the set of all displacements, $z$, such that reflected $B_1$ and $A$ overlap by at least one element as given below in (3.9) –

$$A_1 = A \oplus B_1 = \left\{ z \mid (\hat{B}_1 \cap A) \neq \phi \right\} \quad (3.9)$$

Erosion shrinks or thins objects in a binary image. A square structuring element of width 3 pixels shown below is used for erosion operation.

$$B_2 = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}$$

$$A_1 \ominus B_2 = \left\{ z \mid (B_2 \subseteq A_1) \right\}$$

$$A_1 \ominus B_2 = \left\{ z \mid (B_2 \cap A_1^c = \phi) \right\} \quad (3.10)$$

Erosion of $A_1$ by $B_2$ is set of all points $z$, such that, $B_2$ translated by $Z$, is contained in $A_1$ and is equivalent to $B_2$ not sharing any common elements with background.

The smooth binary image is obtained after applying the discussed morphological operations.

### 3.2.4 Algorithm for Edge Detection

In order to derive required geometrical feature vector only the edge of the input hand image is needed. Thus it is required that the white portion of the image must be clubbed together as a single entity and then utilized for further geometric calculations while the black portion completely emerge out as a different region. This is why the edge detection algorithm is used. This algorithm converts the pixels lying inside and outside of the palm boundary to black while the boundary emerges out in white color. This gives a visual image of the detected edge of the palm. In MATLAB edge detection can be done by various methods like canny,
sobel or prewitt methods. The Marr-Hildreth edge detector represented by (3.15) is chosen which uses Laplacian operator $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and 2D-Gaussian function, $G$ as given in (3.11) and (3.12).

$$G(x, y) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$  \hspace{1cm} (3.11)

with standard deviation $\sigma$. To find $\nabla^2 G$, following differentiations are performed.

$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2}$$ \hspace{1cm} (3.12)

$$\nabla^2 G(x, y) = \frac{\partial}{\partial x} \left[ -\left( \frac{x}{\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \right) \right] + \frac{\partial}{\partial y} \left[ -\left( \frac{y}{\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \right) \right]$$ \hspace{1cm} (3.13)

$$\nabla^2 G(x, y) = \left[ \frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} + \left[ \frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$ \hspace{1cm} (3.14)

$$\nabla^2 G(x, y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$ \hspace{1cm} (3.15)

This expression is called the Laplacian of a Gaussian (LoG). $\nabla^2 G$ operator is chosen as it is capable of being tuned to act at any desired scale, so that large operators can be used to detect blurry edges and small operators to detect sharply focused fine details. The Gaussian part of operator blurs the image, thus reducing the intensity of structures at scales much smaller than $\sigma$ and the Laplacian is an isotropic operator (invariant to rotation). LoG filter is convolved with the smoothened binary image and then zero crossings are found to determine the edges of image. The obtained hand contour/boundary from the edges is shown in Figure 3.7.
3.3 FEATURE EXTRACTION

Various algorithms and processes are used to obtain geometrical features which include finding the ROI, calculating the valley points, calculating the tip points, diameter, area, perimeter of the palm etc. These have been implemented using MATLAB utilizing the various functions and inbuilt utilities to make code simpler and efficient.

Following algorithms are used for calculating the various hand geometry feature values. The geometrical features such as finger length, tip to centroid distance and valley to centroid distance depend upon the calculation of the tip and the valley points.

3.3.1 Steps for calculating valley points

Location of valley points is required for finding the features. The valley points are calculated using the following steps:

Step I: The edge of the input image is found using the steps discussed in section 3.2.4.

Step II: The pixel list of this edge is obtained by using the ‘regionprops’ function which is available in the image processing toolbox of MATLAB.
Step III: A loop is started taking an array of ten pixel values at a time.

Step IV: In this list there might be a point whose y coordinate is lesser than those of the pixels proceeding and those following it. Thus a point of local minima is obtained in the neighborhood of high y coordinate values. This is taken as the valley point. The four valley points obtained using above steps are shown in Figure 3.8.

### 3.3.2 Steps for calculating tip points

Location of tip points is also required for finding the geometrical features. The tip points are calculated using the following steps:

Step I: The edge of the input image is obtained by using the steps discussed in section 3.2.4.

Step II: The pixel list of this edge is obtained by using the ‘regionprops’ function.

Step III: A loop is started taking an array of ten pixel values at a time.

Step IV: In this list there might be a point whose y coordinate is greater than those of the pixels proceeding and those following it. Thus a point of local maxima is obtained in the neighborhood of low y coordinate values. This is taken as the tip point.

![Figure 3.8: Valley Points](image1.png)  ![Figure 3.9: Tip Points](image2.png)

The five finger tip points obtained using above steps are shown in Figure 3.9.
3.3.3 Steps for calculating the geometrical features

The centroid of hand contour is calculated. The centroid of hand and tip and valley points of fingers as shown in Figure 3.10 are used for calculating geometrical features like palm diameter, finger lengths, finger widths, tip to centroid distance, valley to centroid distance.

The area and perimeter is given by the ‘regionprops’ function defined in the image processing toolbar. Diameter of palm is used as one feature. The straight horizontal line joining two points lying on the contour and passing through centroid is considered as the diameter of palm. Steps for calculating diameter of palm are as follows:

![Figure 3.10: a) Centroid on hand b) Tip and Valley points](image)

Step I: The edge of the input image is obtained using the steps discussed in section 3.2.4.

Step II: The pixel list of this edge is obtained by using the ‘regionprops’ function.

Step III: The centroid of this edge is found by using the ‘regionprops’ function.

Step IV: A loop is started taking one pixel value at a time.

Step V: In this list there might be a point whose y coordinate is same as that of the centroid y coordinate. Two such points are obtained on the edge.
Step VI: Distance formula is applied between these obtained points and this distance is the required diameter of the palm.

The finger length as shown in Figure 3.11(a) is calculated by applying distance formula between the tip and the valley points. The centroid to valley distance is calculated by applying distance formula between the centroid and the valley points. Similarly, the centroid to tip distance as shown in Figure 3.11(b) is calculated by applying distance formula between the centroid and the tip points.

A system of weights is introduced which decides the importance of a particular feature value. Some feature values are less important than others and thus must not be given a high weight. Therefore there is a need to find which feature is to be considered important and which not to. For this the inter class and the intra class variance are found. Intra class variance is taken among the feature vectors of the different images obtained from same person at different instants. For optimal results it should be low. Inter class variance is taken among the feature vector of the images obtained from different persons. For optimal result it should be very high. Therefore such feature values are found for which the interclass variance is more and intra class variance is less. That feature is given a high weight. The feature value for which the interclass variance is less and intra class variance is more is given a low weight. The inter and intra class variance are calculated for the hand geometry features.

Figure 3.11: a) Finger lengths  b) Tip to centroid distance  c) Finger Widths
After calculating the tip and valley points, geometrical features are calculated for a sample image as shown in Figure 3.11. Feature vector comprises of seventeen geometrical features which include five finger lengths, four finger widths, area, perimeter, palm width and five tips to centroid distances. Analysis of the feature values for 10 individuals is done by calculating the seventeen feature values for 10 images of each individual. It is found that the maximum interclass variation and minimum intraclass variation is for tip to centroid distance. So out of seventeen features, the five tips to centroid distances are scaled by 1.5 to generate feature vector.

3.4 MATCHING ALGORITHM

The working of this biometric system is conducted by calculating the features obtained from the input image and are then matched against the images in the database. Even under the best conditions, it is not likely that the features calculated match exactly with the features of the previous image templates of the same individual. Thus the difference in the feature values obtained from the image and that of the template is calculated. The matching possibility is more if the difference is small. Also the lesser the difference between query feature set and the correct feature set stored in database, the more significant the feature becomes.

For matching module, a feature array of the extracted features is formed and stored as database. The aim is to find a matching array. If the image matches the prototype image the element is set to one. If the image does not match the prototype image the element is set to zero.

3.4.1 Enrolment phase

Step I: A directory is defined in which all the database images are stored.
Step II: The file type is defined to be taken as the input using the ‘fullfile’ function. In this case it will be all the files having .jpg extension. The images are taken as a list.

Step III: A loop is started taking one image at a time from the list.

Step IV: The feature values of the images are extracted and stored in an excel file using the ‘xlswrite’ command.

3.4.2 Verification phase

Step I: The feature values of input image are calculated.

Step II: The feature values of template images are imported. This is done using the ‘xlsread’ command.

Step III: The Euclidian distance between the feature values is calculated.

Step IV: Based on this distance, a threshold value is set. This threshold value is changed again and again so as to reach a value which gives the minimum EER.

Step V: The element of the matching array is set to zero if distance is greater than the threshold; else it is set to one.

Step VI: The elements of the array are recorded and then added.

Step VII: The sum will denote the number of matching images.

This section discussed the methods for extraction of geometrical feature values and then matching them with the already stored templates in the database.

3.5 RESULTS AND DISCUSSION

The false rejection rate and false acceptance rate are plotted for different threshold values in order to find the usefulness of system. A false rejection rate is
obtained by comparing templates stored in database with the same hand feature vector while a false acceptance rate is obtained by comparing the templates of different hands. When a template in the database is matched against those templates representing a different user and after comparing if the match value falls below the chosen threshold, it is considered to be a false acceptance by the system. This process is repeated for all the users in the database. On the other hand when a template in the database is matched against those templates representing a same user and if the match value is more than the chosen threshold then it is considered to be false rejection. If the value selected for threshold decreases, FRR increases. The reason being that due to low threshold, some genuine users may be rejected because of noise or some other factors. In general applications, threshold is chosen in a way that gives same value of FAR and FRR. Such threshold is EER. In case of a very high security system the threshold may be raised while for a system where false rejects are of more concern the threshold might be lowered.

The threshold is determined by testing various hand images. For matching, the matching vector is calculated using the determined threshold on the basis of which the system accepts the user as an authorized user. In case the match score is below the threshold the user is rejected as an unauthorized user.

The system was tested on a database with 10 subjects taking 10 images per subject. These images contain some images of the same individuals taken at different time intervals. For the new enrolments it requires input of ten images, from which the features are extracted and their mean values are stored in the database. For the verification the input image is taken and extracted features are matched with the database. The FAR-FRR curve for the different value of threshold is shown in Figure 3.12. First of all match score is calculated without any weight. In this experiment the value of false acceptance rate is calculated and then false rejection rate is calculated at different threshold values. The obtained EER% is 6.7.
Secondly the match score is calculated with weighted features. The FAR-FRR curve for the different value of threshold is shown in Figure 3.13. In this experiment the value of false acceptance rate is calculated and then false rejection rate is calculated at different threshold values. The obtained EER is 4.8%. 
Finally the experiment is done on larger database. The graph in Figure 3.14 is plotted using the feature values obtained by the hand geometry features of 50 users without consideration of weights. Corresponding to the difference obtained between the template image and the input image the range of the threshold is taken from 0 to 100.

The graph in Figure 3.15 is plotted using the feature values obtained by the hand geometry features of 50 users with consideration of weights. The result obtained is EER value of 0.04993 at threshold 336.5 for weighted features.
The weighted feature technique gives better results. The weights depend upon the importance of the feature. Weights assigned are the success rates of a particular feature taken alone. Both the FAR and FRR values get reduced while performing the tests with the weighted feature technique. The obtained results are compared with other geometrical approaches present in literature and are presented in Table 3.1.
Table 3.1: Performance comparison of Peg-Free Contact-Based Hand-Geometry Biometric Algorithm

<table>
<thead>
<tr>
<th>Reference</th>
<th>Features</th>
<th>Classification</th>
<th>Population</th>
<th>Database size</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wong and Shi (2002)</td>
<td>Hierarchical Recognition process using G1 &amp; G2</td>
<td>G1: Gaussian Mixture Model</td>
<td>22</td>
<td>323</td>
<td>FAR: 2.2%</td>
</tr>
<tr>
<td></td>
<td>G1: Finger lengths and widths</td>
<td>G2: Euclidean distance</td>
<td></td>
<td></td>
<td>FRR: 1.3%</td>
</tr>
<tr>
<td></td>
<td>G2: Fingertip region</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bulatov et al (2004)</td>
<td>30 geometric features including length and</td>
<td>Chbyshev Metric</td>
<td>70</td>
<td>714</td>
<td>FAR: 1%</td>
</tr>
<tr>
<td></td>
<td>height of fingers and palm</td>
<td></td>
<td></td>
<td></td>
<td>FRR: 3%</td>
</tr>
<tr>
<td>Nongluk et al (2005)</td>
<td>21 geometric features including finger lengths</td>
<td>Absolute, Weighted-absolute Euclidean,</td>
<td>96</td>
<td>-</td>
<td>EER (verification) ≤ 2.99%</td>
</tr>
<tr>
<td></td>
<td>and widths and palm width</td>
<td>Weighted-Euclidean and Two new distance</td>
<td></td>
<td></td>
<td>EER (identification) ≤ 5.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matrices D1 and S1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ozbay and Watsui (2008)</td>
<td>21 geometric features including finger length</td>
<td>Used eight different functions</td>
<td>54</td>
<td>983</td>
<td>EER ≤ 2.5%</td>
</tr>
<tr>
<td></td>
<td>and width and palm width</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asghari and Sadeghi (2009)</td>
<td>15 geometric features including finger widths</td>
<td>Euclidean, Absolute</td>
<td>50</td>
<td>500</td>
<td>3.198 ≤ EER</td>
</tr>
<tr>
<td></td>
<td>and finger circumferences</td>
<td></td>
<td></td>
<td></td>
<td>≤ 6.973</td>
</tr>
<tr>
<td>Proposed</td>
<td>16 geometrical features including finger</td>
<td>Euclidean distance</td>
<td>50</td>
<td>250</td>
<td>FFR (without</td>
</tr>
<tr>
<td></td>
<td>lengths, widths, centroid to tip distance,</td>
<td></td>
<td></td>
<td></td>
<td>weights) ≤</td>
</tr>
<tr>
<td></td>
<td>area, perimeter</td>
<td></td>
<td></td>
<td></td>
<td>10.28,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FFR (with weights)</td>
</tr>
</tbody>
</table>

3.6 CONCLUSION
A simple and effective peg-free hand-geometry verification system has been developed in this Chapter. The system is experimented with the databases of different sizes. Good verification efficiencies are obtained. Peg free contact based JUET database is created for verification of results. So the image acquisition is free from problems of peg based system.

Intraclass and interclass variations are analyzed to decide the weights which are used to scale some geometrical features. The weighted features give better results. Geometrical features may introduce errors due to their dependence on landmark points. Performance can further be improved by using efficient peak and valley detection and more accurate method for finding geometrical features. The problem is handled in Chapter 4.