Chapter 5

Working at Fingerprint Image Level

In this research, we have worked towards solving interoperability issue in fingerprint recognition systems at fingerprint Image Level as well as at the Matcher Level. This chapter focuses on interoperability issue at the fingerprint image level. One of the major causes of the interoperability issue is the non-linear distortions that get added to fingerprint images due to the difference in sensors hardware [5]. If these distortions are corrected at the fingerprint image level, then the existing matching algorithms, i.e. Bozorth3 in this case, can match images acquired from different sensors with better accuracy.

5.1 Pre-Work: Modeling the distortions

A. Ross and R. Nadgir proposed a non-linear calibration scheme [5] to register two fingerprint sensors using Thin Plate Splines (TPS). For the calibration, they manually found minutiae correspondence between few pairs of representative images from two sensors which serve as inputs to the TPS model and derived inter-sensor distortion compensation model. Even though the proposed inter-sensor distortion compensation model shows better inter-sensor accuracy, it has a dependency on sensor hardware. There must be a TPS model designed for each sensor pair. Hence, every new fingerprint sensor must be calibrated with all existing sensors. Also, this method is not applicable in case if the source (fingerprint sensor) of the fingerprint image is not known. Thus we claim that, hardware specific distortions need to be compensated at the time of fingerprint image generation. We worked on distortion compensation technique by modeling fingerprint sensors to generate hardware independent distortion free fingerprint images.
5.1.1 Distortion Compensation

Digital fingerprint image is generated by mapping 3D ridge structure on the finger skin to the 2D surface of the fingerprint sensor. This is analogous to a camera which captures 2D image (photograph) of 3D objects (scene). Imagine a tree and its image captured by a camera. If the same tree is captured with another camera with different characteristics (resolution, lens quality, size etc) then the generated image would look different. The mapping between 3D world co-ordinates to 2D image co-ordinates is dependent on intrinsic (focal length, image sensor format, and principle point) and extrinsic (shear, scaling and translation) properties of the camera which are collectively represented by its projection matrix 'P'. Every camera has its own unique projection matrix. Hence with above mentioned analogy, interoperability issue between fingerprint images from two sensors arises because every fingerprint sensor has its own unique projection matrix. Similar approach is used by Schuckers et al [125] for angle angle compensation of iris images.

As shown in Fig.5.1, M represents the finger to be acquired. When the finger is enrolled by sensor 1, fingerprint image m1 is generated and when enrolled by sensor 2, fingerprint image m2 is generated. Let 'P1' and 'P2' be the projection matrices of sensor 1 and 2 respectively. According to projective geometry concepts, fingerprint image 'm1' is a projection of M through 'P1' and fingerprint image 'm2' is a projection of M through 'P2'. Thus we identify that, interoperability issue arises when matching algorithm is not able to match 'm1' and 'm2' because P1 and P2 are different. The process of finding projection matrix 'P' of a camera is called as camera calibration.

As shown in Fig.5.2, if two sensors are calibrated before i.e. if projection matrices 'P1' and 'P2' are known, then fingerprint images 'm1' and 'm2' can be treated as projection of finger M through 'P1' and 'P2' respectively. With the concepts of projective geometry, we can perform inverse operation to recover M by back projecting m1 through P1 and by back projecting m2 through P2. In order to have a compatibility with existing matching algorithms, M has to be converted to 2D image. If constant projection matrix is used for both the sensors, the effect of the sensor specific distortions can be compensated. Thus, we propose universal fingerprint sensor projection matrix called 'canonical projection matrix' to generate identical fingerprint images 'm' from all the sensors. As shown in Fig.5.2, in order to eliminate interoperability issue each fingerprint sensor should generate fingerprint image 'm' instated of m1 and
m2 etc. We tried to calibrate the fingerprint sensors with similar technique used for camera calibration.

**Camera Calibration Process**

Camera calibration is the process of finding projection matrix P. We used Camera Calibration Toolbox for Matlab [126] where multiple images of the test object (chess board in this care) are captured as shown in Fig.5.3. The dimensions of the chess board squares are predefined and their lengths are measured in the captured images to calculate the projection matrix P [126]. Fig.5.4 illustrates the camera calibration
Fingerprint Sensor Calibration

We propose a similar approach like camera calibration to find the projection matrix of fingerprint sensor. Fig.5.5 (a) shows the test pattern used for fingerprint sensor calibration. The distance between each points in the pattern on finger skin is measured and their relative distance in the captured fingerprint images is measured as shown in the Fig.5.5 (b). After collecting multiple samples from same finger with different orientations we calibrated the projection matrix similar to camera calibration technique using Camera Calibration Toolbox for Matlab [126].

5.1.2 Difficulties Encountered

The distortion compensation technique explained above has two steps. First step is to calibrate each fingerprint sensor (i.e. find its projection matrix 'P') and use
CHAPTER 5. WORKING AT FINGERPRINT IMAGE LEVEL

Figure 5.4: Camera extrinsic properties

Figure 5.5: Fingerprint sensor calibration process
it to perform inverse operation on 2D fingerprint image \( m \) to the imaginary 3D finger \( M \). The second step is to project the imaginary 3D finger \( M \) to 2D canonical fingerprint image via universal projection matrix. The success of this technique depends on calculation of projection matrix \( P \) which involves the measurement of distance between pre-defined pattern drawn on the finger skin.

The difference between 2D image capture from camera and fingerprint image capture is physical contact of the object with the sensor. The elastic nature of the finger skin also adds non-linear distortions which are incorporated along with the distortions introduced by the sensor hardware. The elastic distortions introduced in the fingerprint image can not be modeled easily as it depends on the individual user’s skin type, pressure applied at the time of finger enrolment or the way finger is enrolled (e.g. tilt and twist) etc. Hence we couldn’t calibrate the fingerprint sensor accurately.

Thus with this pre-work, we came to a conclusion that, along with the sensor hardware distortions, non-linear elastic distortions are also need be considered while designing the distortion compensation technique. With this consideration, we have developed a method to standardise the fingerprint images and represent them in the canonical form as explained in the following section.

5.2 Canonical Fingerprint Representation

From the experience of above mentioned pre-work, we came to the conclusion that in order to solve interoperability issue between fingerprint sensors both hardware specific distortions and elastic distortions of finger skin must be modeled together. As elastic distortions are personalised with individual, it is difficult to fit a single generalised distortion compensation model for a particular fingerprint sensor. Thus we shifted our focus from sensor calibration to fingerprint image standardization and worked to develop a hardware independent fingerprint representation called as canonical fingerprint image.

5.2.1 Empirical Understanding of Bozorth3 Matching Algorithm

Before designing a canonical representation, it is important to analyse why existing fingerprint matching algorithms are unable to incorporate inter-sensor variations. In
Figure 5.6: BOZORTH3 matching algorithm working principle

Bozorth3 matching algorithm relative measurement of features (angles and euclidian distance) for each minutia from all other minutiae is done as shown in the Fig.5.6. Thus for each minutia point ‘k’ the feature vector in the form of \((d_{kj}, \beta_1, \beta_2, k, j, \theta_{kj})\) is created from all remaining j minutiae points. For the determination of matching minutiae pairs in two images, following three tests are conducted:

\[
\Delta_d (d(P_m), d(G_n)) \leq (T_d) \\
\Delta_\beta (\beta_1(P_m), \beta_1(G_n)) \leq (T_\beta) \\
\Delta_\beta (\beta_2(P_m), \beta_2(G_n)) \leq (T_\beta)
\]

‘d’ is Euclidean distance between two minutiae and \(T_d\) is the distance threshold. Distance threshold is a critical parameter because, if it is set too high then chances of false rejection are more and if it is set too low then chances of false acceptance are more. The Euclidean distance based threshold approach fails to incorporate non-linear distortions introduced by different technology based sensors. In case of inter-sensor matching, because of difference in hardware specific distortions the euclidian distance between neighbouring minutiae points never remains same but varies in non-linear fashion. Thus performance of Bozoth3 matching algorithm decrease for inter-sensor matching.

### 5.2.2 Canonical Form

We define a new representation called Canonical Fingerprint Image, which is a mathematical reconstruction of the fingerprint image captured from any fingerprint sensor in a standard format. Even if the fingerprint image is captured from different sensors...
and non-linear distortions are added, one feature which is more likely to remain constant is the number of ridges (ridge-count) between two minutiae points. We use this feature to compensate non-linearity introduced in the fingerprint images and generate standardised fingerprint image called as canonical fingerprint image. Steps involved in canonical fingerprint generation are as below.

### 5.2.3 Feature Extraction

![Figure 5.7: Illustration of location of descriptors in a fingerprint model](image)

In order to increase the accuracy of matching algorithms, many researchers have considered additional information than \( \{X, Y, \theta\} \) for each minutiae, known as descriptors. We propose six dimensional feature vector as \( \{X, Y, Rc, \phi, \theta, Q\} \). Where, 'Rc' is ridge-count number of the minutiae point from core point and the position angle '\( \phi \)' is an angle between line joining between minutiae point and core point with line of symmetry measured in anti-clock wise direction. In current implementation, usage of quality parameter 'Q' is reserved for future development. Fig.5.7 illustrates the proposed descriptors. For extracting descriptors or addition features than \( \{X, Y, \theta\} \) i.e. \( \{Rc, \phi\} \), we need to follow the steps mentioned in flow chart illustrated in Fig.5.8.
CHAPTER 5. WORKING AT FINGERPRINT IMAGE LEVEL

Figure 5.8: Feature extraction process

Generating Feature vector

Process of identification of core point, line of symmetry and base point is explained in the section 3.1.2, 3.1.3 and 3.1.4 respectively. Fig.5.9 shows fingerprint image with few sample minutiae points. As explained above, we extract \{X,Y,\theta\} features using chain code technique given by J.Abraham et al [119] and the quality Q is reserved for future enhancement. We propose two new additional features i.e. \{Rc,\phi\}. The extracted feature vectors for these sample minutiae points are illustrated in Table 5.1. Similar features are extracted for all remaining minutiae points.

As shown in Fig.5.9, let the co-ordinates of the core point C be (Xc,Yc) and co-ordinates of the minutiae point M be (Xm,Ym). We trace (pixel by pixel) the straight line joining the two points and increment the ridge count number with every occurrence of a black pixel. As shown in Fig.5.9, the ridge count for minutia point M1 is 2 and position angle \phi is an angle between a point B, the core point C and the minutiae point M1, measured in an anticlockwise direction i.e. 30 degrees. Similarly we extract the descriptors for all remaining minutiae points. Table 5.1. illustrates the six dimensional feature vectors for all six sample minutiae points highlighted in
Canonical mapping is a process of mapping fingerprint minutiae points on concentric circles centered at core with constant incremental radius ‘d’. As shown in Fig.5.10, Point C is the core point and point B is the base point. A line joining point C and B is the line of symmetry in a canonical form. As illustrated in Table 5.1, the minutia point ‘M1’ has ridge count 2 and position angle 30 degrees. Thus, the minutia point ‘M1’ is mapped on second concentric circle in Fig. 5.10 at an angle 30 degrees anti-clockwise from line BC. Similarly, all minutiae points from Fig.5.9 are mapped to a standard format shown in Fig.5.10.

Let \([X, Y, \theta]\) be the minutiae features and \([X', Y', \theta']\) be their canonical representation. The pseudo code to transform \([X, Y]\) to \([X', Y']\) is presented in Fig.5.11. Let
CHAPTER 5. WORKING AT FINGERPRINT IMAGE LEVEL

<table>
<thead>
<tr>
<th>Minutiae Point</th>
<th>X</th>
<th>Y</th>
<th>θ</th>
<th>Q</th>
<th>R_C</th>
<th>φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>140</td>
<td>136</td>
<td>290</td>
<td>-</td>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>M2</td>
<td>135</td>
<td>120</td>
<td>110</td>
<td>-</td>
<td>3</td>
<td>110</td>
</tr>
<tr>
<td>M3</td>
<td>60</td>
<td>70</td>
<td>25</td>
<td>-</td>
<td>9</td>
<td>178</td>
</tr>
<tr>
<td>M4</td>
<td>15</td>
<td>90</td>
<td>65</td>
<td>-</td>
<td>15</td>
<td>230</td>
</tr>
<tr>
<td>M5</td>
<td>43</td>
<td>140</td>
<td>82</td>
<td>-</td>
<td>4</td>
<td>285</td>
</tr>
<tr>
<td>M6</td>
<td>128</td>
<td>243</td>
<td>123</td>
<td>-</td>
<td>5</td>
<td>330</td>
</tr>
</tbody>
</table>

Table 5.1: Illustration of the sample features extracted in the proposed format for the fingerprint image shown in the Fig.5.9

β be the angle made by line of symmetry with the positive direction of X-axis (Refer Fig.5.12). The pseudo code to transform \([\theta]\) to \([\theta']\) is presented in Fig.5.13.
Figure 5.10: Proposed canonical representation of the sample minutiae points shown in Fig. 5.9
**Require:** Let, 'd' be the radius of the first concentric circle

\[ [X,Y,\theta,Q,R_c,\phi]_i \text{, feature vector of } i^{th} \text{ minutiae point} \]

\[ N \text{ be the total number of minutiae points} \]

\[ \beta \text{ be the angle made by the line of symmetry with +ve X-axis} \]

\[ [X',Y',\theta']_i \text{, feature vector of } i^{th} \text{ minutiae point in canonical form} \]

**for** \( i = 1 : N \) **do**

**if** \( \phi \leq 90^0 \) **then**

map the minutiae in first quadrant

\[ \delta X = R_c * d * \sin \phi \]

\[ \delta Y = R_c * d * \cos \phi \]

\[ X' = X + \delta X \]

\[ Y' = Y + \delta X \]

**end if**

**if** \( 90^0 < \phi \leq 180^0 \) **then**

% map the minutiae in second quadrant

\[ \phi = \phi - 90 \]

\[ \delta X = R_c * d * \cos \phi \]

\[ \delta Y = R_c * d * \sin \phi \]

\[ X' = X + \delta X \]

\[ Y' = Y - \delta X \]

**end if**

**if** \( 180^0 < \phi \leq 270^0 \) **then**

% map the minutiae in third quadrant

\[ \phi = 270 - \phi \]

\[ \delta X = R_c * d * \cos \phi \]

\[ \delta Y = R_c * d * \sin \phi \]

\[ X' = X - \delta X \]

\[ Y' = Y - \delta X \]

**end if**

**if** \( 270^0 < \phi \leq 360^0 \) **then**

% map the minutiae in fourth quadrant

\[ \phi = \phi - 270 \]

\[ \delta X = R_c * d * \cos \phi \]

\[ \delta Y = R_c * d * \sin \phi \]

\[ X' = X + \delta X \]

\[ Y' = Y + \delta X \]

**end if**

**end for**

**Figure 5.11:** Pseudo code for canonical mapping of X and Y
Figure 5.12: Orientation angle $\beta$ made by line of symmetry with positive direction of X-axis

Require: Let, 'd' be the radius of the first concentric circle

$[X, Y, \theta, Q, R_c, \phi]$; feature vector of $i^{th}$ minutiae point

$N$ be the total number of minutiae points

$\beta$ be the angle made by the line of symmetry with +ve X-axis

$[X', Y', \theta']$, feature vector of $i^{th}$ minutiae point in canonical form

for $i = 1:N$ do
  if $\beta \leq 90^0$ then
    $\theta' = \theta + \beta$
    if $\theta' > 360$ then
      $\theta' = \theta' - 360$
    end if
  end if
  else
    $\theta' = \theta - \beta$
    if $\theta' < 0$ then
      $\theta' = \theta' + 360$
    end if
  end if
end for

Figure 5.13: Pseudo code for canonical mapping of $\theta$
Fingerprint Reconstruction

In order to have a compatibility with other feature extraction algorithms and matching algorithms we can reconstruct the canonical fingerprint image from canonical minutiae feature set \([X', Y', \theta']\). The spiral phase concept explained by Feng and Jain [16] can be used to reconstruct the canonical fingerprint mathematically (Refer Fig.5.14). There are other methods also invented to reconstruct the fingerprint image from the minutiae point [127]. These methods have their own limitation of generating spurious minutiae. Reconstruction of a fingerprint image from the minutiae point is not in the scope of this work.

![Fingerprint Reconstruction Image](image)

**Figure 5.14:** Reconstruction of canonical fingerprint image (with sample two minutiae only) using minutiae to phase approach [16]
5.3 Experimental Evaluation

We follow the same FVC protocol [124] explained in section 4.1. We use Bozorth3 matching algorithm for the performance evaluation. Experimentation procedure is illustrated in Fig.5.15.

![Diagram](image_url)

**Figure 5.15:** Experimentation procedure for performance comparison with canonical fingerprint representation

5.3.1 MCYT-100 database

Fig.5.16 (a) shows ROC curves for same and inter-sensor performance using NIST's *mindtct*+Bozorth3 matching algorithm. Fig.5.16 (b) shows ROC curves for same and inter-sensor performance using canonical representation. In this case, both the core point and the line of symmetry is extracted as per the methods explained in 3.1.2 and 3.1.3. Identification of accurate core point is still a challenging task for biometrics research community. We perform an experiment of manual selection of core point and automatic identification of line of symmetry from the core point as explained in section 3.1.3. The ROC curves for for same and inter-sensor performance using canonical representation with manual selection of core point is illustrated in Fig.5.17 (a). In order to test the accuracy of canonical representation we eliminate the effect of inaccuracy of core point and line of symmetry identification by manually selecting both the core point and base point. Fig.5.17 (b) shows the optimum performance of canonical representation. Table 5.2 illustrates the performance evaluation of proposed canonical templates using the same Bozorth3 matching algorithm on MCYT-100 database.
Figure 5.16: ROC curve using Bozorth3 matching algorithm on MCYT-100 fingerprint database [1]

<table>
<thead>
<tr>
<th>Experiment</th>
<th>EER</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB1 (optical) Vs. DB1 (Optical)</td>
<td>0.40</td>
<td>0.9999</td>
</tr>
<tr>
<td>DB2 (capacitive) Vs. DB2 (capacitive)</td>
<td>0.48</td>
<td>0.99978</td>
</tr>
<tr>
<td>DB1 (optical) Vs. DB2 (capacitive)</td>
<td>0.73</td>
<td>0.9983</td>
</tr>
</tbody>
</table>

Table 5.2: Performance evaluation: Equal Error Rate (EER), of canonical templates + Bozorth3 matching algorithm on MCYT-100 database
Figure 5.17: ROC curve using Bozorth3 matching algorithm on MCYT-100 fingerprint database [1]
5.3.2 Chinese Academy of Science Database

The ROC curves for same and inter-sensor performance on Chinese Academy of Science multi-sensor database using canonical representation with manual selection of core point and base point is illustrated in Fig.5.18 and Fig.5.19.
CHAPTER 5. WORKING AT FINGERPRINT IMAGE LEVEL

Figure 5.18: Comparison of NIST’s *mindtct* templates and Canonical templates on Chinese Academy of Science Multi-Sensor fingerprint database [18] using NIST’s Bozorth3 matching algorithm [17]
(a) Sensor-3 (DB3) vs. all other sensors using NIST’s mindtct template
(b) Sensor-3 (DB3) vs. all other sensors using Canonical Representation
(c) Sensor-4 (DB4) vs. all other sensors using NIST’s mindtct template
(d) Sensor-4 (DB4) vs. all other sensors using Canonical Representation

Figure 5.19: Comparison of NIST’s mindtct templates and canonical representation on Chinese Academy of Science Multi-Sensor fingerprint database [18] using NIST’s Bozorth3 matching algorithm [17]
Table 5.3 illustrates the performance evaluation of proposed canonical templates using the same Bozoarh3 matching algorithm on Chinese Academy of Science Multi-sensor database.

<table>
<thead>
<tr>
<th>Vs.</th>
<th>DB1</th>
<th>DB2</th>
<th>DB3</th>
<th>DB4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB1</td>
<td>03.07</td>
<td>04.48</td>
<td>07.30</td>
<td>04.74</td>
</tr>
<tr>
<td>DB2</td>
<td>04.48</td>
<td>02.71</td>
<td>07.05</td>
<td>05.38</td>
</tr>
<tr>
<td>DB3</td>
<td>07.30</td>
<td>07.05</td>
<td>02.94</td>
<td>09.61</td>
</tr>
<tr>
<td>DB4</td>
<td>04.74</td>
<td>05.38</td>
<td>09.61</td>
<td>02.43</td>
</tr>
</tbody>
</table>

**Table 5.3**: Performance evaluation: Equal Error Rate (EER), of Canonical templates + Bozorth3 matching algorithm on Chinese Academy of Science Multi-Sensor fingerprint database

5.4 Discussion

In the above experimentation, we have used Bozorth3 matching algorithm for both NIST’s Mindtct and canonical representation. In case of MCYT-100 database, for the same sensor comparisons, canonical representation matches the performance of the NIST’s Mindtct template. But, for cross comparisons, canonical representation outperforms NIST’s Mindtct template using Bozorth3 matching algorithm (Refer Table 5.2.). In case of Chinese Academy of Science Multi-Sensor fingerprint database, Table 5.4. highlights following points:

- For cross-sensor comparisons, canonical representation provides better performance than NIST’s Mindtct template
- For same sensor, because of canonical representation, EER increases slightly
- For sensor-4 (DB4), canonical representation outperforms NIST’s Mindtct templates
- For sensor-3 (DB3), canonical representation fails for both same sensor as well as cross-comparisons
Table 5.4: Comparison between NIST’s *Mindtct* template and Canonical representation based on Equal Error Rate (EER) of Bozorth3 matching algorithm on Chinese Academy of Science Multi-Sensor fingerprint database

5.4.1 Identified Limitations of the Canonical Representation

Uncertainty of ridge-count number

Ridge-Count number of minutiae point ideally remains constant irrespective of the non-linear sensor distortions or elastic deformations. However, during practice implementation of fingerprint image enhancement process, sometimes we encounter with situations like broken ridges or two ridges touching each other. In these situations, ridge-count number is uncertain within limits of ± 1. This uncertainty disturbs the local structure during the canonical mapping. Bozorth3 fails to incorporate this uncertainty as canonical form maps that minutiae on different ridges (Refer Table 5.4.).
CHAPTER 5. WORKING AT FINGERPRINT IMAGE LEVEL

Uncertainty of position angle

Due to large non-linear distortions, the position angle of identical minutiae points on two fingers may be different. If the difference between the position angle is large, it may disturb the local structure in the canonical form.

Partial fingerprint problem

One of the reasons behind the interoperability issue with sensor-3 (Db3) is partial fingerprint problem. The size of the fingerprint image from sensor-3 (DB3) is much smaller than the size of other images. Because of the small size, the biometric data captured is less. If the common overlapping region between two fingerprint images to be matched is very small, Bozorth3 fails to match these two images. Hence, canonical mapping of limited biometric data won’t solve the interoperability issue.

5.5 Conclusion

In this work, we have addressed the fingerprint interoperability issue at the image level. We focused on non-linear elastic distortions as one of the major causes of the interoperability issue. We proposed an approach based on minutiae ridge count to convert fingerprint images into a canonical form; irrespective of the underlined sensor hardware.

We performed experiments on the MCYT fingerprint database having one optical and one capacitive sensor. With canonical representation, we observed significant improvement of EER for cross sensor comparisons with NIST’s Bozorth3 matching algorithm.

For Chinese Academy of science database, canonical mapping performs better for cross-comparisons, but its performance is slightly lower for the same sensor than the NIST’s Mindtct template. For sensor-4 (DB4) it outperforms the NIST’s Mindtct template. The reasons for performance drop in case of Sensor-3 (DB3) are discussed above.

Thus, in-order to take the advantage of ridge-count interoperability issue has to be tackled at matching level.