Fingerprint Mosaicking Algorithm to Improve the Performance of Fingerprint Matching System

Sandhya Tarar*, Ela Kumar

School of ICT, Gautam Buddha University, Greater Noida, 201306, U.P., India

*Corresponding Author: tarar.sandhya@gmail.com

Copyright © 2014 Horizon Research Publishing All rights reserved.

Abstract Mosaicking in biometrics is a topic of great interest among researchers these days, since it is an important step towards solving the problem of security of data. This paper presents design and implementation of Mosaicking algorithm. This technique is beneficial for providing authentic access as well as security to the biometric image template. We have designed an algorithm for fingerprint mosaicking in order to achieve the required performance. The proposed algorithm provides the effective solution of security issues regarding biometrics techniques without affecting the fingerprint quality. Results discussion section depicts the efficacy of proposed algorithm.

Keywords Biometric Technology, Fingerprint Identification System (FIS), Fingerprint Verification Competition (FVC), Fingerprint Mosaicking

1. Introduction

A fingerprint based verification system consists of two different stages: i) the enrolment stage where the fingerprint images acquired and stored in the database. In this stage multiple impressions captured for same fingerprint stored as template in database, and ii) the authentication stage in which a query template is matched against the stored fingerprint images for identification of the authorised user. Solid state sensors widely used these days in various applications as commercial, financial and many more. These sensors sense the limited amount of ridges and furrow those there on a fingertip. So, limited amount of information can be extracted from these fingerprints in comparison to the information extracted from elaborated fingerprints. In this scenario the matching performance of the fingerprint verification system is being affected since the small overlap region between the query template and stored template results in fewer corresponding points and increases the false rejection or may accept the false image.

To deal with the above stated problem, fingerprint mosaicking scheme is being designed in which multiple

impressions of same fingerprint taken as input and a composite template of fingerprint is generated that has more information of the finger tip and requires less storage space in comparison to multiple fingerprint impressions. It helps in reducing the matching time as only one template exist corresponding to single finger tip.

It has been observed by the researchers that the contact offered by the solid state fingerprint sensors less and hence do not provide sufficient information for high accuracy during verification of user [1]. Further, if multiple impressions acquired for same finger by these sensors, the overlap region available may be very small and hence affecting the matching performance of the verification system. Fingerprint mosaicking is a method that artificially constructs an approximation of a rolled image from a series of dab images. Dabbed images those fingerprint impression that captured through sensors when the user dabs (presses) the fingertip on it [23]. Fingerprint mosaicking is the reconciliation of the information to generate a composite template [6]. Merging a number of enrolment impressions into a super-fingerprint increases fingerprint a and accounts for low quality impressions [32]. Researcher had given a method in which a finger is rolled over a fingerprint sensor that captures a sequence of images. These images easily combined into a single larger image [26]. Another method is proposed to construct a composite fingerprint image from multiple impressions that taken at different moments [15]. A fingerprint template contains the unique features of the fingerprint image and can be used for fingerprint matching. The exact composition of the fingerprint template is dependent on the algorithm used to extract the unique features.

Fingerprint Mosaicking can be classified as a) Touch-based fingerprint mosaicking: The touched-based mosaicking of fingerprint is done with the images that being collected through the sensors on which human fingers actually come in contact. Image acquisition is done when users actually swipe or roll their fingers on the sensor. The multiple images collected of the same finger then used to compose the template. b) Touchless-based fingerprint mosaicking: In this mosaicking scheme the image of

fingerprint is captured through contactless technology. The human finger do not comes in contact with the sensor but the technology is designed such that the image is captured through the camera (eg. CCD) and stored in the databases. This technology comes in the existence because the image acquired through touch-based system may contain noise (due to non-uniform Pressure, Dirt, Skin Conditions Etc.).

2. Previous Work

When multiple impressions of the same fingerprint available, one can in principle expect to make more reliable decisions about the presence or quality of fingerprint features by combining information from the samples in some fashion [34]. Algorithms that combine multiple impressions based on two main approaches: (i) mosaicking, which combine the data at the image level [26 and 31], and (ii) template synthesis, which combine the data at the feature level [34]. Different researchers have posed different algorithms to achieve the task of mosaicking of fingerprint. Ratha *et al.* tiled the image sequence of a rolling fingerprint grabbed by a large a scanner. They had developed a mosaicking scheme to integrate multiple snapshots of a fingerprint. The multiple snapshots acquired as the user rolls the finger on the surface of the sensor and, therefore, a specific temporal order is imposed on the image frames when constructing the composite image. The authors examines five composition schemes that stack the gray scale images together and construct a composite mosaicked image, by associating a confidence value with every pixel. They evaluate the efficacy of these schemes by observing the a and quality (in terms of the number of valid minutiae points detected) of the composite image [26]. Jain *et al.* found that fingerprint mosaicking is a 3D surface registration problem that can be solved using a modified the Iterative Closest Point algorithm. Authors have given 5 steps to create the template and then using the composite template for the comparison with the query image. Steps involved - i) Pre-processing the Fingerprint Image: This step removes any undesirable “salt-and-pepper” noise that may be present in the valleys of the fingerprint image. ii) Fingerprint Segmentation: The purpose of segmentation is to separate the foreground regions (that have ridge and valley information) from the background regions (that have no fingerprint information) in the image. This distinction is necessary to prevent the ICP algorithm from choosing control points in the background regions (due to the homogeneity in intensity in these regions). iii) Fingerprint as a Range Image: The intensity values directly used as range values. The planar coordinate (x; y) is treated as the range value, z, at that location. Final Transformation matrix is calculated. iv) Constructing a Composite Image: The intensity images IP and IQ integrated into a new image IR by using final transformation matrix calculated in the above step to compute the new spatial coordinate of every pixel in IP. v) Augmenting the Minutiae Sets: If MP and MQ refer to the minutiae sets extracted from

IP and IQ, respectively, then a composite minutiae set, MR, is obtained by augmenting MP and MQ. The minutiae orientation, θ , is not recomputed [1]. But this algorithm does not work for non-linear deformation of fingerprints.

A hybrid algorithm given by Zhang *et al.* mosaick the stream of swipe fingerprint frames. Algorithm works on hybrid mosaicking scheme for the stream of swipe fingerprint frames, which uses the minimum mean absolute error as the registration criterion to find an integer translation shift that registers two adjacent swipe fingerprint frames to the nest integral pixel coordinates while the extension phase correlation method with singular value decomposition is used to get sub-pixel precision [35]. The mosaicking scheme discussed assumes no rotation, scaling and shear effects present in individual frames of the swipe fingerprint images. Due to the nature of the skin, some amount of elastic deformation is expected which is a serious drawback of the given algorithm. Another method implemented by Choi *et al.* tries to mosaick different parts of a fingerprint collected by having the subject roll and slide his/her finger on the surface of small a fingerprint scanner [23]. The algorithm that is being proposed for elastic deformation present in constituent fingerprints before stitching them is given by Shah *et al.* [31]. The researchers have proposed an algorithm called Thin Plate Splines (TPS) which is a 2D spatial generalization of the cubic spline, is an effective tool for estimating the deformation in fingerprints based on landmark points. The algorithm consists of three main steps-

i) Coarse Alignment ii) Fine Alignment iii) Blending. The composite image is then enhanced in order to obtain the mosaicked template [31]. This paper focuses on removal of elastic deformation but unable to identify the best pair of images suitable for mosaicking process. In [34], Jiang et al. created and updated a super-template as the user provided new samples during verification. They assigned a weight to each minutia according to its frequency of occurrence. Depending on the weight values, spurious minutiae can fade away over time or missing minutiae can appear at some point in time. The drawback was that they simply discard minutiae having weights less than a threshold. An improvement in accuracy relative to using individual templates was reported; however, a comparison with other fusion approaches was not provided. This reported problem was overcome by Tamer *et al.*, their approach utilizes these weights for matching [32].

Fingerprint mosaicking refers to the process of combining two or more fingerprint images into one mosaicked template such that when a query fingerprint is matched against a mosaicked template fingerprint stored in database to establish the identity of an individual, the result of matching process yields the outcome in more accurate way. Various schemes have been implemented to combine the fingerprints and make a mosaic image of it for matching procedure. Ross *et al.* [5] had given that mosaicing can be accomplished at two distinct levels: (a) The image level (image mosaicing), (b) the feature level (feature mosaicing). Ratha *et al.* [3] have developed a mosaicking scheme to integrate multiple snapshots of a fingerprint. The multiple snapshots acquired

as the user rolls the finger on the surface of the sensor. The authors examine five composition schemes that stack the gray scale images together and construct a composite mosaicked image, by associating a confidence value with every pixel. They evaluate the efficacy of these schemes by observing the a and quality (in terms of the number of valid minutiae points detected) of the composite image. Jain *et al.* [4] found that fingerprint mosaicking is a 3D surface registration problem that can be solved using a modified Iterative Closest Point algorithm. Steps involved - Fingerprint image pre-processing, segmentation of the image on the basis of foreground and background regions, and then image representation. At the end a composite image is prepd and minutiae sets augmented. But this algorithm does not work for non-linear deformation of fingerprints.

The algorithm that had being proposed for elastic deformation removal is [2]. Shah *et al.* had given an algorithm called Thin Plate Splines (TPS) which is a 2D spatial generalization of the cubic spline, an effective tool for estimating the deformation in fingerprints based on landmark points. This paper focuses on removal of elastic deformation but unable to identify the best pair of images suitable for mosaicking process. Choi *et al.* [7] had given a Recursive Ridge Mapping algorithm which eliminates erroneous correspondences and decides how well the transformation is estimated by calculating the registration error with a normalized distance map. The method consists of three phases: feature extraction, transform estimation, and mosaicking. Zhang *et al.* [8] had given an algorithm that is not minutiae based. The image of fingerprint captured is a stream of swipe fingerprint frames, which contiguous and sh some mutual support. The frame data then 'stitched' or 'registered' together to form a fingerprint image. Researchers had used the Pixel level and sub-pixel level both as hybrid mosaicing for constructing the final image of the fingerprint. Another existing technology for fingerprint mosaicking is touchless based mosaicking which capture the data where fingerprint do not come in contact with the sensor. Choi *et al.* [9] had designed a touchless system which consists of a camera and two planar mirrors. Three different views captured and mosaicked to form a single template.

3. Material and Methods: System Modelling

In order to design an algorithm, two impressions stitched together to generate a composite image and minutiae points then detected from the image generated after mosaicking. The image is matched with the query template and result is stored for analysis. Now a fingerprint image from database of the same finger is fetched and used for matching followed by the minutia for extraction. The result generated is stored. The outcome of both matching process is compd and will be analyzed. We have demonstrated proposed algorithm in following steps:

Step1: Load two fingerprint images of same finger tip in the frame for mosaicking of the images.

```
[imagefile1 , pathname] = uigetfile('*.*bmp;*.BMP;*.tif;*.TI
F;*.jpg','Open An Fingerprint image');
```

```
if imagefile1 ~= 0
```

```
cd(pathname);
```

```
image1= readimage(char(imagefile1));
```

Step2: Mosaic the images loaded in the frame with the help of mosaicking algorithm proposed in which images constructed by: i) Fast Fourier Transformation in two dimensions, ii) finding conjugate of the result generated and iii) calculating the height and width of the mosaicked image on the basis of input images.

```
for i = 1 to mosaicked image height
```

```
if i < m
```

```
for j = 1:Width of Loaded image
```

```
A(i, j) = I3(i, j);
```

```
elseif i > m-1 and i< Height of Loaded Image+1
```

```
for j = 1 to mosaicked image width
```

```
if j < n
```

```
A(i, j) = I3(i, j);
```

```
else
```

```
A(i, j) = I4 (i-m+1, j-n+1);
```

```
else
```

```
for j = n to mosaicked image width
```

```
A(i, j)= I4 (i-m+1, j-n+1);
```

where I3 and I4 the transformed images, m and n the rows and columns respectively of the loaded images and A is the array used to store image.

Step3: Save the mosaicked image in the database.

```
W = inputdlg(textSaveName); // dialog box opens to write
the name for mosaicked image
```

```
W = char(W);
```

```
F = getframe(AxesHandle);
```

```
image(F.cdata);
```

```
imwrite(F.cdata,W);
```

Step4: Extract minutiae points of the mosaicked image by the extraction algorithm.

Step5: Extract minutiae points of the fingerprint image that is the query template.

```
if ~ isempty(end_list)
```

```
plot(end_list(:,2),end_list(:,1),'r');
```

```
if size(end_list,2) == 3
```

```
hold on
```

```
[u,v] = pol2cart(end_list(:,3),10);
```

```
quiver(end_list(:,2),end_list(:,1),u,v,0,'g');
```

Step6: Perform matching algorithm between images: mosaicked image (after minutiae extraction) and fingerprint image (created after extraction of minutiae).

a) Choose the minimum ridge length

b) Comp the similarity certainty of two ridge

c) Transfer all the minutia in two fingerprint based on the reference pair of minutia

d) Perform Matching

```
for i =1:minuN1
```

```
for j =1:minuN2
```

```

if (abs(fullXY1(1,i)-fullXY2(1,j)) < xyrange &
abs(fullXY1(2,i)-fullXY2(2,j))<xyrange)
angle = abs(fullXY1(3, i) - fullXY2(3, j) );
if or (angle < pi/3, abs(angle-pi) < pi/6)
num_match = num_match+1; break;
Step7: Comp and store the result
Get the largest matching score
current_match_percent=num_match;
if current_match_percent > max_percent(1,1);
max_percent(1,1) = current_match_percent;
max_percent(1,2) = k1;
max_percent(1,3) = k2;
end;
Step8: Perform analysis

```

Figure 1 shows the process of fingerprint mosaicking in terms of different blocks those correspondingly depicts the steps those followed during the process.

4. Result Discussion

4.1. Experimental Results

Fingerprint mosaicking ensures design of information presented by two or more impressions of a finger in order to generate composite information. It can be accomplished by blending these impressions into a single mosaic, or by integrating the feature sets (i.e. minutiae information) pertaining to these impressions [10].

The main aim is to develop a technique which identifies stable features from the image because the existing algorithm uses the minutiae as the initial factor for mosaicking and if it is being extracted wrong then remaining procedure conducted results in wrong result .

We have traversed various journal papers and conference papers in order to analyze the material and to get a complete understanding and historical data about extraction techniques for the fingerprint mosaicking technology.

For our research work we have collected information regarding different algorithms available for extraction. Apart from this, the database according to the research a is selected on which the algorithm designed is implemented and hence a comparison can be shown between the existing algorithm and the modified or designed algorithm. To achieve this step we have collected information about the availability of the globally available data-sets on the internet. Identification of individuals is a very basic societal requirement. Fingerprint verification and identification is used as a primary technique

for identifying the individual's identity.

For applying the algorithm designed we need the fingerprint database to be created and database creation is a time consuming task for any biometric system. Therefore we have used an open access database Fingerprint Verification Competition (FVC) 2002 to implement our proposed algorithm. It helps in reduction of data collection time. FVC2002 is an open access database and has been created by Fingerprint Competition Verification team (FVC) [48].

We have designed an algorithm for fingerprint mosaicking which takes two impressions as input and stitch it together to generate a composite image and minutiae points then detected from the image generated after mosaicking. Figure 1 shows the block diagram of mosaicking process where the image is matched with the query template and result is stored for analysis. Now a fingerprint image from database of the same finger is fetched and used for matching followed by the minutiae extraction process. The result generated is stored. The outcome of both matching process is compd and will be analyzed. The Extraction algorithm takes the input as fingerprint image (mosaicked fingerprint image or non-mosaicked fingerprint image), processes the image and gives the output after detecting the minutiae points in the fingerprint image. The matching algorithm matches the minutiae sets of the two images and give the output in percentage form that how much the images similar to each other or can say the matching percentage is the output of this algorithm.

Fingerprint Mosaicking can be classified as a) Touch-based fingerprint mosaicking: The touched-based mosaicking of fingerprint is done with the images that being collected through the sensors on which human fingers actually come in contact. b) Touchless-based fingerprint mosaicking: In this mosaicking scheme the image of fingerprint is captured through contactless technology. The human finger do not comes in contact with the sensor but the technology is designed such that the image is captured through the camera and stored in the databases. This technology comes in the existence because the image acquired through touch-based system may contain noise (due to non-uniform pressure, dirt, skin conditions etc.) in the fingerprint image.

The algorithm designed will help to improve the performance of fingerprint matching system by reducing the time of matching of images as the number of fingerprint images stored corresponding to one finger tip is reduced and the overlapping region between query template and composite template is more.

Figure 1. Block Diagram of Fingerprint Mosaicking Process

Fingerprint Identification System requires fingerprint images either live or from stored database. In order to evaluate the proposed algorithm we have used an open access database of fingerprint, Fingerprint Verification Competition (FVC) 2002 database. Fingerprint Verification Competition (FVC) is international competition that focuses on fingerprint verification softw assessment. It provides fingerprint database to the users and saves the time of data collection. FVC2002 is an open access database and has been created by Fingerprint Competition Verification (FVC). FVC2002 is a second Fingerprint Verification Competition [48]. The First International Fingerprint Verification Competition (FVC2000) was organized by the authors in the year 2000. The Organizers of FVC : Biometric System Laboratory (University of Bologna), Pattern Recognition and Image Processing Laboratory (Michigan State University), Biometric Test Center (San Jose State University), Biometric Recognition Group - ATVS (Universidad Autonoma de Madrid). Two years after the first edition, a new Fingerprint Verification Competition (FVC2002) was organized by the authors, with the aim of determining the state-of-the art in this challenging pattern recognition application [11]. In FVC, we have used four types of databases as DB1, DB2, DB3 and DB4 based on different sensors. Each set consisting

of 10*8 fingerprint images and as a total 320 fingerprint images have been incorporated during system testing.

The Running time of an algorithm depends on various factors as: i. Processor of Machine ii. Read/Write speed to memory iii. 32-bit v/s 64 bit iv. Input (Rate of growth of time with respect to the input)

In our research work we have focused on Input to calculate the time complexity. As far as run time is concern, it varies at different stages of processing. Working time calculation is mentioned in this section as follows: At fingerprint image loading stage run time complexity is linear i.e. $O(1)$, since execution takes unit time. Mosaicking process increased the complexity because of loop invariant and induction variable, that become $O(n^2)$. Rest all of the stages as minutia extraction, matching with template, false minutia removal have linear complexities i.e. $O(1)$. Hence, the overall complexity of algorithm is $O(n^2)$.

As far as noise is concerned we have also eliminated false minutia during minutia extraction process, as a means of noise free system.

Snapshots after implementing proposed algorithm shown in the following figures: i.e. figure 2 to 12. Figure 2 shows the design of Graphical User Interface which is important to show the images at various stages during the whole process.

Figure 3 depicts the point when fingerprint image is uploaded. Figure 4 depicts mosaicked image generation. After stitching with other image mosaicked image can be generated. After mosaicking one need to compute the image with the authentic one. So, for that purpose, mosaicked image is saved in database for future reference. That is shown in Figure 5. Figure 6 shows the design of another GUI for matching purpose. Figure 8 and Figure 9 shows the minutiae extraction process and real minutiae extraction process correspondingly. For matching phase we have set a threshold value for image matching that is 75 %. That means if 75% portion of a captured image is matched with stored image then we are going to consider captured image as an authentic image.

1) Graphical User Interface

- 2) After loading the fingerprint image
- 3) Mosaicked Image is generated
- 4) Image saved in the database for matching
- 5) For matching another GUI is created
- 6) Fingerprint image is loaded for matching with query template. Before matching pre-processing of the loaded image is done.
- 7) Minutiae extraction
- 8) False minutiae that exist in the above image removed and real minutiae that helpful in matching process extracted.
- 9) Save the minutiae extracted image
- 10) Matching procedure
- 11) Output of matching process

Figure 2. GUI Interface

Figure 3. Loaded Fingerprint image in the axes

Figure 4. Mosaicked Images

Figure 5. Saving the Mosaicked Image

Figure 6. GUI for Matching

Figure 7. Fingerprint image loaded for matching

Figure 8. Minutiae Extracted Image

Figure 9. Real minutiae extracted image

Figure 10. Image saved in database

Figure 11. Matching of fingerprint templates

Figure 12. Matching result

5. Performance Analysis

In FVC, we have used four types of databases as DB1, DB2, DB3 and DB4 based on different sensors. Each set consisting of 10×8 fingerprint images and as a total 320 fingerprint images have been incorporated during system testing. Different impressions of each of these fingers are obtained over different sessions. The impressions obtained from the first session are used to construct the composite template while the impressions obtained from the second session are used as query images during the test phase of the experiment. So, 160 pairs of images are used to construct minutia templates (extracting minutia from the mosaicked image) and the rest is used as query images. During performance analysis, when the composite template is used an improved performance is observed compared to using the individual templates. We have described a fingerprint template construction technique that integrates information available in two different impressions of the same finger, by using proposed algorithm. Experiments indicate that mosaicking the impressions together and then extracting the minutia, results in a better performance of matching system. Figure 13 duly reflects the performance analysis in terms of Genuine Acceptance Rate (GAR) and False Acceptance Rate (FAR).

The Running time of an algorithm depends on various factors as: i. Processor of Machine ii. Read/Write speed to memory iii. 32-bit v/s 64 bit iv. Input (Rate of growth of time with respect to the input)

In our research work we have focused on Input to calculate the time complexity. As far as run time is concern, it varies at different stages of processing. Working time calculation is mentioned in this section as follows: At fingerprint image loading stage run time complexity is linear i.e. $O(1)$, since execution takes unit time. Mosaicking process increased the complexity because of loop invariant and induction variable, that become $O(n^2)$. Rest all of the stages as minutia

extraction, matching with template, false minutia removal have linear complexities i.e. $O(1)$. Hence, the overall complexity of algorithm is $O(n^2)$.

As far as noise is concerned we have also eliminated false minutia during minutia extraction process, as a means of noise free system.

Figure 13. The Receiver Operating Characteristics Curve

6. Conclusion and Future Work

This research describes and discusses the mosaicking process in order to improve the quality of fingerprint images. Proposed algorithm shows that accuracy of matching process is increased with the help of fingerprint mosaicking. This is measured in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR). We have used an open access database Fingerprint Verification Competition (FVC 2002) in our experiments. In FVC, we have four types of databases as DB1, DB2, DB3 and DB4 based on different sensors. Each set consisting of 10×8 fingerprint images and as a total 320 fingerprint images, those have been incorporated during proposed algorithm execution. In order to reduce errors of

various types occurred during system execution, we have reduced False Acceptance Rate. We are working on the algorithm implementation and system installation in the security system for our university.

After implementing the proposed method FRR is higher while FAR is reduced as shown in figure 13. Matching percentage is increased since FAR is decreased. Since the complexity of mosaicking algorithm is $O(n^2)$, it will not make any adverse impact on the performance of fingerprint identification system.

REFERENCES

- [1] Anil Jain and Arun Ross, May 13 - 17, Fingerprint Mosaicking, Proc of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, Florida, pp. IV-4064 - IV-4067, 2002.
- [2] A. K. Jain and J. Feng, Latent Fingerprint Matching. Technical Report MSUCSE- 09-10, Michigan State University, 2009.
- [3] A. K. Jain, L. Hong, and S. Pankanti, Biometrics: Promising Frontiers for Emerging Identification Market, Communications of the ACM, vol. 43, no.2, pp. 91–98, (2000).
- [4] A. K. Jain, L. Hong, and R. Bolle. On-line Fingerprint Verification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):302–314, April 1997.
- [5] Arun Abraham Ross, Information Fusion in Fingerprint Authentication, Doctor of Philosophy Dissertation, 2003.
- [6] Arun Ross, Samir Shah and Jidnya Shah, Image versus Feature Mosaicking: A case study in Fingerprints, Conference on Biometric Technology for Human Identification III, (Orlando, USA), pp. 620208-1 – 620208-12, April 2006.
- [7] B.Y. Hiew, Andrew B.J. Teoh and Y.H. Pang, Touch-less Fingerprint Recognition System, Automatic Identification Advanced Technologies, IEEE Workshop, pp. 24 – 29, 2007.
- [8] Chulhan Lee, Sanghoon Lee, and Jaihie Kim, A Study of Touchless Fingerprint Recognition System, Springer-Verlag Berlin Heidelberg, 2006.
- [9] C. Wu, J. Zhou, Z. Bian, and G. Rong. Robust Crease Detection in Fingerprint Images. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 505–510, Madison, Wisconsin, June 2003.
- [10] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, New York: Springer-Verlag, 2003.
- [11] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K. Jain, "FVC2002: Second Fingerprint Verification Competition", in proceedings 16th International Conference on Pattern Recognition, Québec City, vol.3, pp.811-814, 2002.
- [12] [12] D. Maio and D. Maltoni, Direct Gray-Scale Minutiae Detection in Fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19-1:27–40, 1997.
- [13] D. R. Ashbaugh, Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology, CRC Press, 1999.
- [14] F. Galton, Personal Identification and Description, Nature, 38:173–177, June 1888.
- [15] [15] Heeseung Choi, Kyoungtaek Choi, and Jaihie Kim, Fingerprint Matching Incorporating Ridge Features With Minutiae, IEEE Transactions On Information Forensics And Security, Vol. 6, No. 2, pp. 338 – 345, June 2011.
- [16] Heeseung Choi, Kyoungtaek Choi, and Jaihie Kim, Mosaicing Touchless and Mirror-Reflected Fingerprint Images, IEEE Transactions on Information Forensics and Security, Vol. 5, No. 1, pp. 52 – 61, March 2010.
- [17] J. D. Stosz and L. A. Alyea. Automated System for Fingerprint Authentication Using Pores and Ridge Structure. In Proc. SPIE Conference on Automatic Systems for the Identification and Inspection of Humans, Vol. 2277, pp. 210–223, San Diego, 1994.
- [18] J. Feng, Z. Ouyang, and A. Cai, Fingerprint matching using ridges, Pattern Recognition, 39:2131–2140, 2006.
- [19] J. Zhou, C. Wu, and D. Zhang, Improving Fingerprint Recognition Based on Crease Detection, In Proc. International Conference on Biometric Authentication (ICBA), pp. 287–293, Hong Kong, China, July 2004.
- [20] J. Zhou, D. He, G. Rong, and Z.Q. Bian. Effective Algorithm for Rolled Fingerprint Construction. Electronics Letters, 37(8):492–494, April 2001.
- [21] K. C. Chan, Y. S. Moon, and P. S. Cheng, Fast Fingerprint Verification Using Sub regions of Fingerprint Images, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 14, No. 1, January 2004.
- [22] K. Kryszczuk, A. Drygajlo, and P. Morier, Extraction of Level 2 and Level 3 Features for Fragmentary Fingerprints, In Proc. COST Action 275 Workshop, pages 83–88, Vigo, Spain, 2004.
- [23] Kyoungtaek Choi, Heeseung Choi and Sangyoun Lee, Fingerprint Image Mosaicking by Recursive Ridge Mapping, IEEE Transactions on System, Man and Cybernetics, vol. 37, No. 5, October 2007.
- [24] Neil Yager and Adnan Amin, Coarse Fingerprint Registration Using Orientation Fields, EURASIP Journal on Applied Signal Processing, pp. 13, 2043–2053, 2005.
- [25] Nelson C. Francisco, Alexandre Zaghetto, Bruno Macchiavello, Eduardo A. B. da Silva, Mamede Lima-Marques, Nuno M. M. Rodrigues and Sérgio M. M. de Faria, Compression of Touchless Multiview Fingerprints, Biometric Measurements and Systems for Security and Medical Applications (BIOMS), IEEE, 2011.
- [26] N. K. Ratha, J. H. Connell, and R. M. Bolle, Image mosing for rolled fingerprint construction, in Proc. Int. Conf. Pattern Recognition, pp. 1651– 1653, 1998.
- [27] N. K. Ratha and V. Govindaraju, Advances in Biometrics: Sensors, Algorithms and Systems, New York: Springer, 2008.
- [28] P. Hymer, Extraction and Application of Secondary Crease Information in Fingerprint Recognition Systems, Master's thesis, Department of Science and Technology, Linköpings University, Sweden, March 2005.

- [29] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley & Sons, 2001.
- [30] Ruggero Donida Labati, Vincenzo Piuri and Fabio Scotti, A Neural-based Minutiae Pair Identification Method for Touch-less Fingerprint Images, IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM), pp. 96-102, 11-12 April 2011.
- [31] S. Shah, A. Ross, J. Shah, S. Crialmeanu, Fingerprint Mosaicing Using Thin Plate Splines, IEEE Conference, 2005.
- [32] Tamer Uz, George Bebis, Ali Erol, Salil Prabhakar: Minutiae-based template synthesis and matching for fingerprint authentication, Computer Vision and Image Understanding, 133(9) pp 979-992, 2009.
- [33] W. Yau, K. Toh, D. Jiang, T. Chen, and J. Lu, On fingerprint template synthesis, in Proceedings of Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV 2000). Singapore, December 2001.
- [34] Xudong Jiang and Ser W., Fingerprint Template Improvement, Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 24, pp. 1121-1126, Aug. 2002.
- [35] Yong-liang Zhang, Jie Yang, and Hong-tao Wu, A Hybrid Swipe Fingerprint Mosaicing Scheme, Springer-Verlag Berlin Heidelberg, 2005.
- [36] Y. Chen, G. Parziale, E. Diaz-Santana, and A. K. Jain, 3D touchless fingerprints: Compatibility with legacy rolled images, in Proc. Biometric Consortium Conf., Baltimore, MD, 2006.
- [37] Yi Chen, Extended Feature Set and Touchless Imaging for Fingerprint Matching, Doctor of Philosophy Dissertation, 2009.
- [38] Y.S. Moon, H.W. Yeung, K.C. Chan, S.O. Chan, Template Synthesis And Image Mosaicking For Fingerprint Registration: An Experimental Study, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 409-12 vol.5, 2004.