**Contemporary Engineering Sciences, Vol. 9, 2016, no. 26, 1283 - 1296**
**HIKARI Ltd, www.m-hikari.com**
**http://dx.doi.org/10.12988/ces.2016.66111**

# A Robust Orientation-Independent Fingerprint Matching Technique

**Raju Rajkumar**

Computer Science Department
Assam University, Silchar-788011, India

**K. Hemachandran**

Computer Science Department
Assam University, Silchar-788011, India



**Abstract**

The paper proposes a novel fingerprint matching algorithm using MinHeap and euclidean distance between the core point and minutiae points. In this algorithm, a hit count and an error tolerance, $\epsilon$, are used for determining the level of similarity between the nodes in MinHeap which is constructed from two fingerprint images for comparison. The comparision is done only at the root node of the MinHeap, $H_a$ and $H_b$, where the deletion of the root node and heapify operations are performed after comparision. The proposed algorithm is tested using FVC2002 dataset and yield good results. The algorithm is also rotation invariant and yields better matching rate.

**Keywords**: Minutiae, Heap, Fingerprint matching, core point, Euclidean distance

## 1 Introduction

Biometric identification system is used for accessing and surveillance of a system in which fingerprint, palm veins, face recognition, DNA, palm print, hand geometry, iris recognition, retina or speech/voice are used for identification of the person under surveillance. Among these biometric information, fingerprint is the oldest and more reliable than other biometric system. Fingerprint recognition system consist of preprocessing and postprocessing stages in which fingerprint enhancement and thinning and fingerprint feature extraction and matching plays an important role in preprocessing and postprocessing part respectively. Minutiae matching is also known as point pattern matching[13] in which ridge ending and ridge bifurcation are used. Ridge ending is a point where the ridge terminates and ridge bifurcation is a point where the ridge is diverged into two branches as shown in Fig. 1. In this paper, a novel technique for fingerprint matching is proposed using minimum heap and minutiae point. The minutiae points are extracted from the fingerprint using crossing number(CN)[8] and the core point is determined[22]. The Euclidean distance of the minutiae points from the core point is calculated and a minimum heap is constructed using the distance, with reference to the core point.

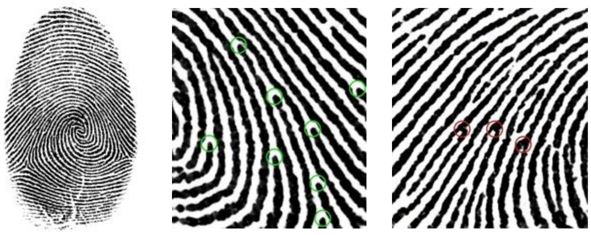


Fig 1: Showing (a) Fingerprint image, (b) Ridge ending (green circle) and (c) Ridge bifurcation (red circle).

Then the heap-based Fingerprint Matching (hFPM) comparison starts with nearest distance from core point, compare second nearest distance from core point and so on. The hFPM algorithm considers the error tolerance rate $\epsilon$ for distance difference for accuracy in the matching. There may be a slight variation of two same fingerprint images, which results mismatch. Further, the hFPM is the fast matching algorithm which is implemented based on MinHeap and the early detection is possible. There is no requirement of comparing entire two different images. Moreover, the hFPM algorithm considers every possible case for matching two fingerprint images, for instance, size difference of the same fingerprint. The hFPM algorithm is based on the distance of minutiae points from core points, that's why the algorithm does not suffer from rotation of an image. If an image is skewed or deteriorated, then the hFPM algorithm may fail and we do not consider such cases. The varieties of fingerprint image quality such as wet images, normal image and dry images from FVC2002 dataset are examined by hFPM algorithm. The novel contribution of the hFPM algorithm to fingerprint matching is enlisted below:

i. Early detection of two different fingerprint.
ii. Image size cannot affect the hFPM.
iii. Accuracy in fingerprint matching is more in comparison to the existing algorithms, found in literature.
iv. A fast fingerprint matching technique with time complexity of O(n) where n is the total number of minutiae points (number of nodes) in heap.
v. For comparison of the fingerprint, most of the algorithms use ridge alignment, whereas the proposed hFPM algorithm, does not consider the alignment as it deals with the distance between the core point and the minutiae and the rotation of a fingerprint image with respect to original image does not affect in the comparison process.
vi. The paper is organized as follows: Section II provides a brief literature review about the fingerprint matching system. Section III gives an outline of the feature extraction. Section IV explains briefly about the proposed heap Fingerprint Matching System (hFPM). Experimental results and discussion are provided in Section V followed by conclusion in Section VI.

## 2 Literature Review

This section provides a brief literature survey of the fingerprint matching system. The fingerprint matching can be achieved using minutiae, correlation and pattern[2].

David[1] proposed a fingerprint matching based on the graph-matching algorithm using graduated assignment.

Jain, Arun and Salil[3] proposed a hybrid fingerprint matching algorithm which used both minutiae points and texture information in local region.

Gabriel, Oluwole and Olumuyiwa[5] proposed a fingerprint matching based on the correlation coefficiency using distance feature from core to minutiae.

Subhas et. al.[6] proposed a fingerprint matching based on spatial information (distance) of minutiae point only and used indexing technique to speed up the matching process.

Feng, Ouyang and Cai[9] proposed a fingerprint matching based on both ridge and minutiae correspondences. The N initial substructures found in the novel alignment (adjacent ridge and minutiae) are used for fingerprint matching.

Andrej, Alexej and Justas[11] proposed minutiae based fingerprint matching algorithm using local structure. The characteristic in the local structure are: rotation and translation invariance, locality (for tolerance to deformations) and fast and easy comparable.

Ito, Nakajima, Kobayashi, Aoki and Higuchi[12], proposed a fingerprint matching system using phase-only correlation which is invariant to shift, brightness and noise.

Ning, Yilong and Hongwei[13] proposed a fingerprint matching algorithm based on Delaunay Triangulation in which co-ordinates and orientation of the minutiae point are used. The algorithim is even applicable for finger print images of different resolutions.

Asker and Sabih[15] proposed elastic minutiae matching algorithm using non-linear transformation model. The non-linear transformation model consists of two stages: first is local matching based on local similarity measurement and the second is global matching which uses the possible correspondences to estimate a global non-rigid transformation.

Karthik[16] proposed a fingerprint matching algorithm based on two stage i.e. local phase spectra and global phase spectra. The final matching decision is based on the fusion of both the global and local similarity score.

Xifeng, Jianhua, Xianglong and Daming[20] proposed fingerprint minutiae matching using the adjacent feature vector. The Adjacent Feature Vector (AFV) consists of four adjacent relative orientations and six ridge count of a minutiae.

Xuejun and Bir[21] proposed a minutiae fingerprint matching based on genetic algorithm. The fitness function which used in genetic algorithm is based on the properties of minutiae which include angles, traingle handedness, traingle direction, maximum side, minutiae density and ridge count.

Weiguo, Howells, Fairhurst and Deravi[23] proposed a memetic fingerprint matching algorithm, which aims at identifying the optimal and near optimal minutiae matching using genetic algorithm. Minutiae descriptor containing orientation and circular region around the minutiae point is used as local feature.

## 3 Fingerprint Minutiae Extraction

In this section, minutiae extraction method is discussed which used the crossing number defined by Rutovitz[8] and widely used by many authors[4,7,10,11,17]. For a pixel P in the snap shot of an image as shown in Fig. 2, the crossing number is calculated using the equation 1.

$$\begin{bmatrix} P_4 & P_3 & P_2 \\ P_5 & P & P_1 \\ P_6 & P_7 & P_8 \end{bmatrix}$$

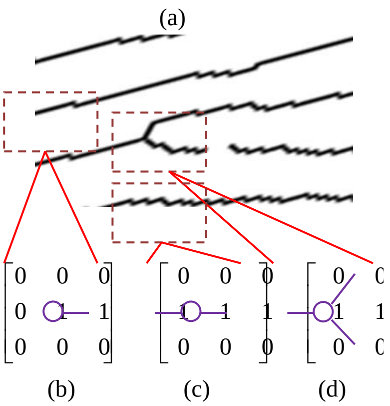
Fig 2. A snap shot of an image in the 3X3 matrix form.

$$\frac{1}{2}\sum_{I=1}^{8}|P_I - P_{I+1}| \qquad (1)$$

where, $P_i$ is the neighbourhood of P with binary value of either 0 or 1.
The CN value using Equation 1 can have values ranging from 0 to 4, which shows the properties of point P providing P=1 as shown in Fig. 3. The properties of CN are as follows:

i. when CN=0, P is known as Isolated point.
ii. when CN=1, P is known as Ending point.
iii. when CN=2, P is known as Connecting point.
iv. when CN=3, P is known as Bifurcation point.
v. when CN=4, P is known as Crossing point.

The occurrence of isolated points and crossing points are treated as false points, so these points are filtered using morphological function in post-processing stage.



(a)Skeleton Image (b)Ending point (CN=1) (c)Connective point(CN=2) (d)Bifurcation point (CN=3)

Fig 3: Illustration of crossing number properties.

## 4 Heap-Based Fingerprint Matching

The heap-based Fingerprint Matching (hFPM) algorithm is based on heap tree property which allows to rotate an image. The hFPM algorithm is very simple, yet robust and powerful. Let $I_a$ be a 2D fingerprint image with a core point. Let $M_{ij}^{a}$ be the minutiae point of Image $I_a$, where i is number of minutiae points and j=1,2,3,4. A row comprises of four points $< d_{i,1}^{a}, X_{i,2}^{a}, Y_{i,3}^{a}, b_{i,4}^{a} >$, where $d_i$ is distance of $i^{th}$ minutiae point from one core point $(X_c^a, Y_c^a)$ and it is calculated using Euclidean distance, i.e.,

$$d_i^a = \sqrt{(X_i^a - X_c^a)^2 + (Y_i^a - Y_c^a)^2}$$

$(X_i^a, Y_i^a)$ is the coordination of i[th] minutiae point and $b_i^a$ is Boolean value of minutiae i.e. either ridge ending or ridge bifurcation. The $H_a$ is a MinHeap for $M_{ij}^a$ with respect to the distance $d_i^a$ from core point $(X_c^a, Y_c^a)$. Let us consider two MinHeap $H_a$ and $H_b$ and compare the top and delete it as shown in the Algorithm. The hFPM takes a core point of the images and calculates the distances. The hFPM constructs $H_a$ and $H_b$ based on the distance of every minutiae point from the core point $(X_c^a, Y_c^a)$ and $(X_c^b, Y_c^b)$ respectively.

**hFPM algorithm**

**Input Data:** $I_b$ is a image of fingerprint and $H_a$ is a preprocessed MinHeap of image $I_a$.
**Output Result:** Hit values

```
Get the direction of I_b as I_b^d ;
Get the ROI of I_b^d as I_b^r ;
Get the core of I_b^r as ;
The minutiae points of I_b^r are calculated and stored in M_ij^a ;
 for i=1 to n do
      d_i^b = sqrt((X_i^b - X_c^b)^2 + (Y_i^b - Y_c^b)^2) ;
end
for i=1 to n do
      for j=1 to 4 do
                 M_ij^b = < d_{i,1}^b, X_{i,2}^b, Y_{i,3}^b, b_{i,4}^b > ;
            end
end
hit = 0;
H_a/b=BuildMinHeap M_ij^b ;
while H_a and H_b not empty   do
         Extract Minimum from Root_a=H_a and Root_b=H_b;
         If   Root_a -> d == Root_b -> d or | Root_a -> d – Root_b -> d |   ≤  ϵ   then
                   If   Type of minutiae is not same   then
                          delete Root_a and Root_b;
               heapify H_a/b;
                   else
                          hit = hit+1;
                          delete Root_a and Root_b;
               heapify H_a/b;
                   end
        end
end
```

After applying the above algorithm, a hit value is determined which gives the number of matching minutiae. Using these hit values, a score is calculated to determine the matching parameter as given in Equation 2.

$$Score = \frac{hit}{Number\ of\ nodes\ in\ H_a\ or\ H_b\ whichever\ is\ minimum} \qquad (2)$$

The construction of minimum Heap is the main task of the proposed algorithm. In this construction, minutiae and core points are detected and the distance between core point (blue circle) and the minutiae points (red and green circle for ridge ending and bifurcation respectively) are calculated as shown in the Fig. 4 (distance is marked with red dot lines). Using the distances, d, minimum heap is constructed where T1<T2<….. ....<T13<T14 and T1, T2,..., T14 are the tuples.

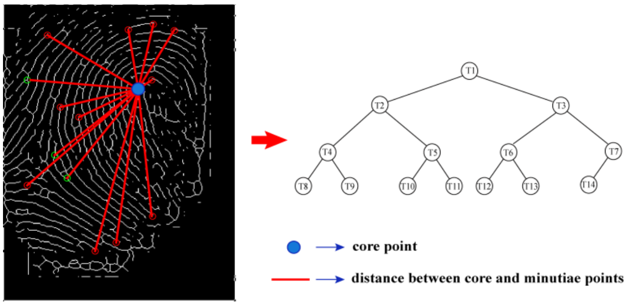



Fig 4: Showing Construction of minimum Heap from minutiae points.

There are some test cases for hFPM which is significant in proving robustness of the algorithm.

i. There is a possibility of having same distance between two different images, but the hFPM algorithm takes care of four column, and these are $< d^a_{i,1}, X^a_{i,2}, Y^a_{i,3}, b^a_{i,4} >$. The hFPM assumes that these four columns have at least a difference with two different images.

ii. Same two images can have different distances of same minutiae due to different pressure applied during acquisition of fingerprint image. For these reason, the hFPM algorithm considers an error tolerance ϵ for matching which is shown in Fig. 5.

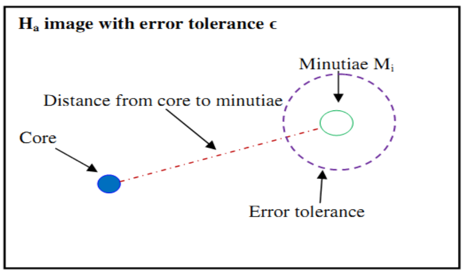



Fig 5: Showing error tolerance ϵ

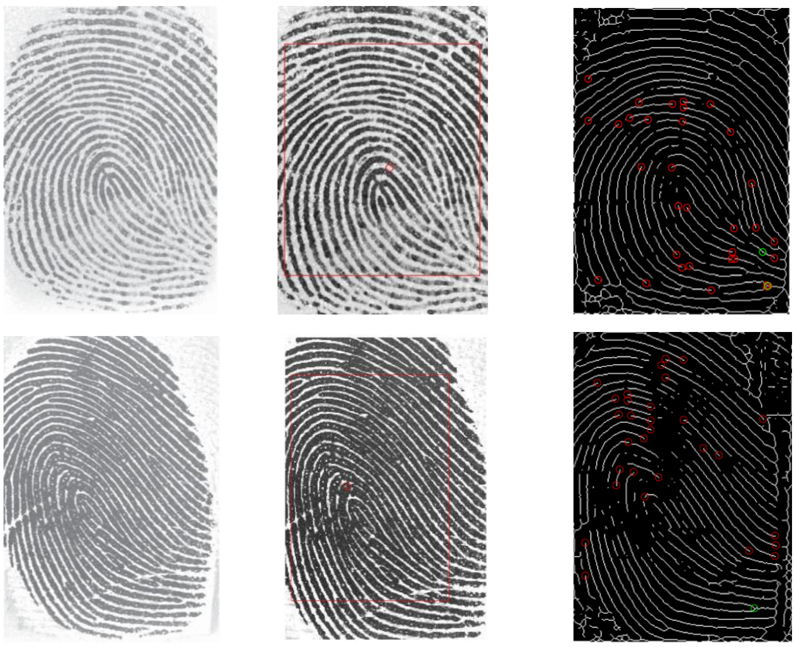
## 5 Results and Discussion

***Dataset:*** For experimentation of the proposed system, DB1, DB2, DB3 and DB4 fingerprint images in database FVC2002 are used. Each database contained 800 fingerprints from 100 different fingers. The description of the FVC2002 database is given in Table 1.

| **Dataset** | **Scanner type** | **Image size** | **Resolution** |
|---|---|---|---|
| DB1 | Low-cost Optical Sensor | 388x374 | 500 dpi |
| DB2 | Low-cost Capacitive Sensor | 296x560 | 500 dpi |
| DB3 | Capacitive Sensor | 300x300 | 500 dpi |
| DB4 | SFinGe v2.51 | 288x384 | 500 dpi |

Table 1: Description of the Fvc2002 Fingerprint Database

Core point extraction and minutiae feature from the FCV2002 fingerprint image is shown in Fig. 6. The core point is extracted using [25] and the minutiae feature is extracted using Raju and Hemachandran[18] which is free from erroneous pixels, i.e., spur and h-break.



(a) Fingerprint (b) Core point (c) Minutiae point

Fig 6: Showing original image, core point and minutiae points of the FVC2002 fingerprint dataset.

***Evaluation:*** For the evaluation of the proposed system two parameters, namely False Matching Rate (FMR) and False Non-Matching Rate (FNMR), are used. Equation 3 and 4 provides the FMR and FNMR.

$$FMR = \frac{Number\,of\ imposter\ fingerprint\ accepted}{Total\,number\,of\,impostertest} \qquad (3)$$

$$FNMR = \frac{Number\,of\ genuine\,fingerprint\,rejected}{Total\,number\ of\ genuinetest} \qquad (4)$$

In our experiment, we used the first 100 individuals to evaluate the FMR and FNMR. FMR is computed based on comparing the first impression of each individual with the first of the others and thus the total number of impostor tests equals ($\frac{100*99}{2}$ =) 4950 [19]. FNMR is computed based on the total number of genuine tests (compare each impression with the seven others for all individuals) and equals ($\frac{8*7}{2}*100$ =) 2800.

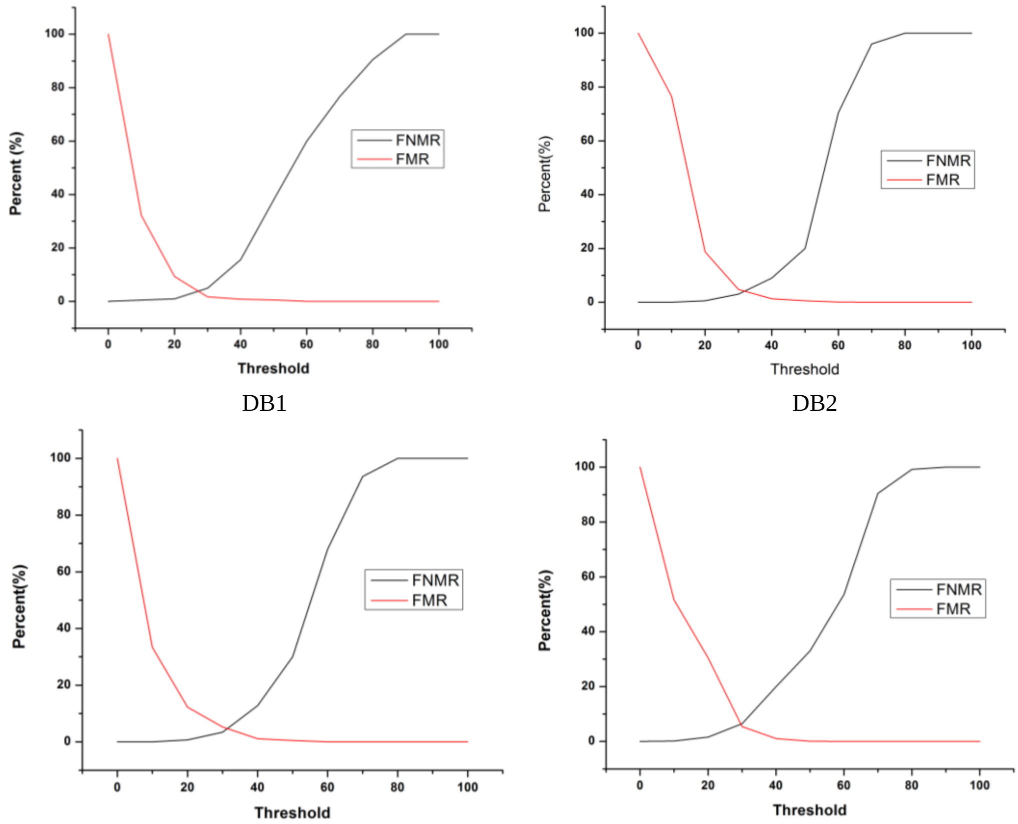
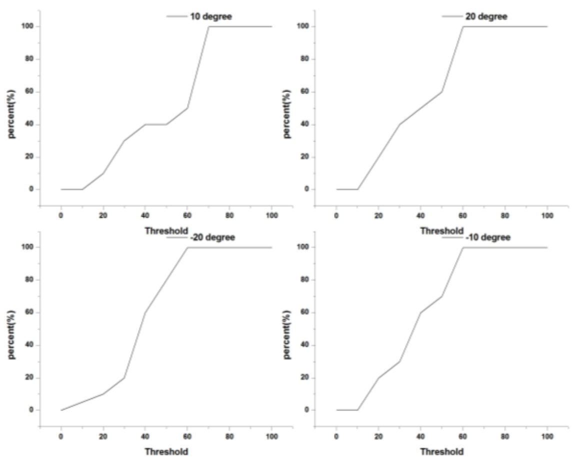
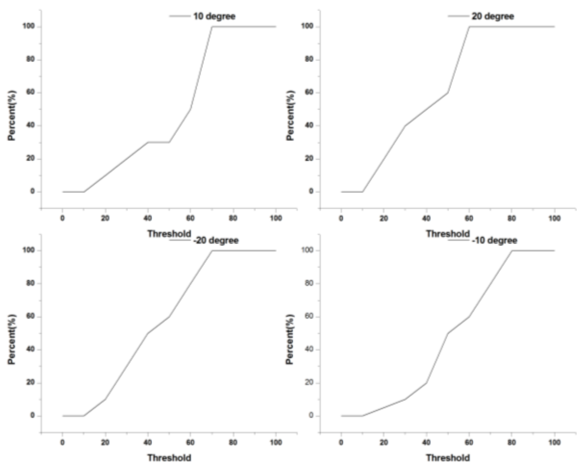



Fig 7: Showing graph of FNMR and FMR against matching threshold using FVC2002 database.

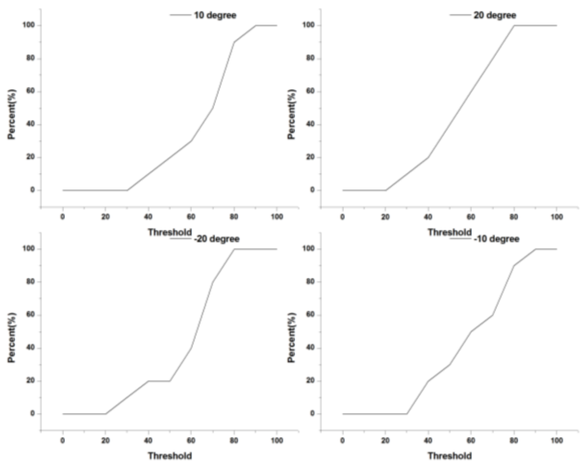
Fig. 7 shows FNMR and FMR using DB1, DB2, DB3 and DB4 FVC2002 dataset. From this figure, it is observed that the crossing pointof FNMR and FMR , i.e EER(Equal Error Rate), is obtained around at the threshold values of 25 to 35.




Dry Image




Dark Image




Normal Imgae

Fig 8: Showing graph of different quality of images at different degree of rotation.

The algorithm is also tested in different quality of fingerprint image with different angle of rotation. The qualities of fingerprint images are collected from FVC2002 on the basis of dry images, normal images and dark images. A base image is rotated in four angle i.e -20 degree, -10 degree, 10 degree and 20 degree and compared with the image taken from same finger without any rotation. The experimental result is shown in Fig. 8. From this figure, we can observe that the normal and dark images give better results, for comparison, than the dry images with respect to Error rate and threshold values.

The proposed algorithm is prone to error if the fingerprint image acquired is partial where the core point is hard to detect. Some images from FVC2002 dataset is given in Fig. 9 of this nature.

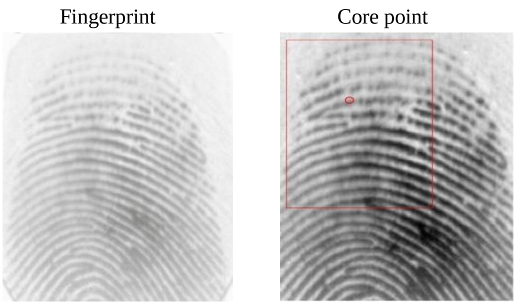



Fig 9: Showing original image and false core point of the FVC2002 fingerprint image.

The proposed system is compared with [14] and [24] and yields better results in terms of ERR as shown in table2.

| **Matching Algorithm** | **ERR** |
|---|---|
| Reference [14] | 3.6 |
| Reference [24] | 1.98 |
| Proposed algorithm | 1.44 |

Table 2. Comparison with other algorithm

## 6 Conclusion

The hFPM algorithm can revolutionize the fingerprint orientation-independent algorithm which is simple but powerful than the orientation based fingerprint matching algorithm as it can match in many possible angle of a fingerprint image. The proposed system uses minimum feature i.e. distance and type of minutiae for matching so it can be used for online verification system. The system is tested using FVC 2002 dataset and yield better result.

**Acknowledgements.** We are thankful to Biometric System Laboratory, University of Bologna for providing the FVC2002 fingerprint database freely for research purpose.

## References

[1] B. David, *Fingerprints: Orientation Free Minutiae Extraction and Using Distances Between Minutiae for Identification and Verification*, dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, 2010.

[2] A. Chandrasekaran and B. Thuraisingham, Fingerprint Matching Algorithm Based on Tree Comparison using Ratios of Relational Distances, *In second International Conference on Availability*, *Reliability and Security*, Vienna, (2007), 273-280. http://dx.doi.org/10.1109/ares.2007.90

[3] A. Jain, A. Ross, S. Prabhakar, Fingerprint matching using minutiae and texture features, *Proc. of Int'l Conference on Image Processing* (ICIP), (2001), 282-285. http://dx.doi.org/10.1109/icip.2001.958106

[4] A. K. Jain, L. Hong and R. Bolle, On-line fingerprint verification, *IEEE Trans. Pattern Ana. and Machine Intelligence*, **19** (1997), no. 4, 302-313. http://dx.doi.org/10.1109/34.587996

[5] B. I. Gabriel, C. A. Oluwole, and J. D. Olumuyiwa, Minutiae Inter-Distance Measure for Fingerprint Matching, *Int'l Conference on Advanced Computational Technologies and Creative Media (ICACTCM'2014), IEEE*, Pattaya (Thailand), (2014), 1-7.

[6] B. Subhas, C. Samiran, S. Debasis, B. Sujoy and S. Goutam, An Efficient Fingerprint Matching Approach Based on Minutiae to Minutiae Distance Using Indexing With Effectively Lower Time Complexity, *International Conference on Information Technology, IEEE*, (2014), 179-183. http://dx.doi.org/10.1109/icit.2014.46

[7] B. M. Mehte, Fingerprint image analysis for automatic identification, *Machine Vision and Applications*, **6** (1993), 124-139. http://dx.doi.org/10.1007/bf01211936

[8] D. Rutovitz, Pattern Recognition, *Journal of the Royal Statistical Society, Series A (General)*, **129** (1966), no. 4, 504-530. http://dx.doi.org/10.2307/2982255

[9] J. Feng, Z. Ouyang and A. Cai, Fingerprint matching using ridges, *Pattern Recognition*, **39** (2006), 2131-2140. http://dx.doi.org/10.1016/j.patcog.2006.05.001

[10] K. Nandakumar, A. K. Jain and A. Ross, Fusion in Multibiometric identification systems: what about the missing data?, Chapter in *Advances in Biometrics*, 2009, 743-752. http://dx.doi.org/10.1007/978-3-642-01793-3_76

[11] K. Andrej, K. Alexej, K. Justas, Fingerprint Minutiae Matching without Global Alignment Using Local Structures, *Informatica*, **19** (2008), no. 1, 31-44.

[12] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki and T. Higuchi, A Fingerprint Matching Technique Based on Phase-Only Correlation, *IEEJ Transactions on Sensors and Micromachines*, **126 (**2006), no. 2, 38-46. http://dx.doi.org/10.1541/ieejsmas.126.38

[13] L. Ning, Y. Yilong and Z. Hongwei, A Fingerprint Matching Algorithm Based on Delaunay Triangulation Net, *Fifth International Conference on Computer and Information Technology (CIT'05),* Shanghai, (2005), 591-595. http://dx.doi.org/10.1109/cit.2005.9

[14] M. D. Garris, C. I. Watson, R. M. McCabe and C. L. Wilson, *Users Guide to NIST Fingerprint Image Software (NFIS)*, Technical Report NISTIR 6813, National Institute of Standards and Technology, 2002.

[15] M. B. Asker and H. G. Sahib, Elastic minutiae matching by means of Thin-Plate Spline Models, *International Conference on Pattern Recognition*, 2002. http://dx.doi.org/10.1109/icpr.2002.1048471

[16] N. Karthik, Fingerprint Matching Based On Minutiae Phase Spectrum, *5$^{th}$ IAPR International Conference on Biometrics*, (2012), 216-221. http://dx.doi.org/10.1109/icb.2012.6199811

[17] N. K. Ratha, S. Chen and A. K Jain, Adaptive flow orientation-based feature extraction in fingerprint images, *Pattern Recognition*, **28** (1995), no. 11, 1657-1672. http://dx.doi.org/10.1016/0031-3203(95)00039-3

[18] R. Rajkumar and K. Hemachandran, A secondary fingerprint enhancement and minutiae extraction, *Singnal and Image Processing*: An International Journal *(SIPIJ)*, **3** (2012), no. 2, 185-196. http://dx.doi.org/10.5121/sipij.2012.3213

[19] S. Tulyakov, F. Farooq, P. Mansukhani and V. Govindaraju, Symmetric hash functions for secure fingerprint biometric systems, *Pattern Recognition Letters*, **28** (2007), no. 16, 2427-2436. http://dx.doi.org/10.1016/j.patrec.2007.08.008

[20] T. Xifeng, H. Jianhua, T. Xianglong, S. Daming, Fingerprint minutiae matching using the adjacent feature vector, *Pattern Recognition Letters*, **26** (2005), 1337-1345. http://dx.doi.org/10.1016/j.patrec.2004.11.012

[21] T. Xuejun and B. Bir, Fingerprint matching by genetic algorithms, *Pattern Recognition*, **39** (2006), 465-477.
http://dx.doi.org/10.1016/j.patcog.2005.09.005

[22] V. S. Srinivasan, Identification of Core and Delta Points in Fingerprint Images, *IAPR Workshop on Machine Vision Applications*, (1990), 263-266.

[23] W. Sheng, G. Howells, M. Fairhurst and F. Deravi, A Memetic Fingerprint Matching Algorithm, *IEEE Transactions on Information Forensics and Security*, **2** (2007), no. 3, 402-412. http://dx.doi.org/10.1109/tifs.2007.902681

[24] Z. Shi and V. Govindaraju, Robust fingerprint matching using spiral partitioning scheme, Chapter in *Advances in Biometrics*, Springer Berlin Heidelberg, 2009, 647-655. http://dx.doi.org/10.1007/978-3-642-01793-3_66

[25] Z. Wu, *Fingerprint Classification Through Self Organization Maps Modified to Treat Uncertainties*, Report for the IS Project, 2002.

**Received: July 1, 2016; Published: September 19, 2016**