

# Fingerprint Minutiae Matching Algorithm using Distance Histogram of Neighborhood

Neeraj Sharma<sup>†</sup>, Joon Jae Lee<sup>\*\*</sup>

## ABSTRACT

Fingerprint verification is being adopted widely to provide positive identification with a high degree of confidence in all practical areas. This popular usage requires reliable methods for matching of these patterns. To meet the latest expectations, the paper presents a pair wise distance histogram method for fingerprint matching. Here, we introduced a randomized algorithm which exploits pair wise distances between the pairs of minutiae, as a basic feature for match. The method undergoes two steps for completion i.e. first it performs the matching locally then global matching parameters are calculated in second step. The proposed method is robust to common problems that fingerprint matching faces, such as scaling, rotation, translational changes and missing points etc. The paper includes the test of algorithm on various randomly generated minutiae and real fingerprints as well. The results of the tests resemble qualities and utility of method in related field.

**Keywords:** Pair Wise Distance Distribution, Histogram, Fingerprint Recognition, Minutiae Matching, Person Identification

## 1. INTRODUCTION

Biometrics features have always been of great importance for personal identification. Due to the durability and unchangeable properties, these databases can be used for a long time with respect to particulars. Fingerprint verification has been associated especially with criminal identification and police operations but now it is found widely useful in various applications like, financial securities, access control, driving license and verification of firearm purchasers. In crime operations fingerprint matching used to be done manually in olden time,

which is a tedious, expensive and time-taking process. That's why it doesn't meet requirements of the new applications. To overcome these problems automatic fingerprint identification system was in great demand. Since the long period of last forty years appreciable work has been done in this field. But still in some cases it is felt that these Automatic Fingerprint Identification System (AFIS) should be improved for meeting latest requirements in related field. The main factors in developing these systems are reliable minutiae extraction algorithm; quantitatively defining a fingerprint match, fingerprint classification etc. The lack of efficiency in getting these factors, create bottlenecks in achieving the desired performance[1].

Almost all the biometric authentication systems use the biometric data in its original form or use processed version of biometric as feature data. Among these processed versions minutiae are one of the most popular form, which is generally used in systems now a days. These systems use tem-

---

\* Corresponding Author : Joon Jae Lee, Address : 2139 Daemyungdong, Namgu, Daegu 705-701, Korea, TEL : +82-53-620-2177, FAX : +82-53-620-2391, E-mail:joonlee@kmu.ac.kr

Receipt date : Oct. 6, 2007, Approval date : Dec. 21, 2007

<sup>†</sup> Division of Computer and Information Engineering, Dongseo University, Busan, Korea (E-mail : neeraj@dit.dongseo.ac.kr)

<sup>\*\*</sup> Department of Game and Mobile Contents, Keimyung University, Daegu, Korea (E-mail : joonlee@kmu.ac.kr)

plate based method to perform the matching. In template based matching method a data base is constructed first, in which the template finger print of all legal persons or civilians are save. Then at the moment of verification the system checks whether the incoming data (unknown finger print point pattern) is mapping properly to any save pattern in database or not. For performing this mapping the method consists of choosing an ordered pair of points from the template and then, for every ordered pair of distinct point in the input, computing a transformation mapping the template pair to the input pair. This transformation is then tested to see whether it maps the entire template into the input.

Most of the conventional method's [2,3] use pre-computation of the transformation parameter's for testing local match. However, in proposed method only distance histogram is compared for local matching which result in fast computation and less complexity in calculation. Apparently, the first researchers to consider randomization in connection with point pattern matching by alignment were Irani and Raghaven [4]. Introduction of proposed method is an effort to build on and improve their results. Regarding the use of pair wise distance distribution histogram for point pattern matching, the concept is quiet new and used by C. H. Park and J. J. Lee [5], in their algorithm. The main idea of the paper is to make use of the fact that the points in a fingerprint point pattern are always uniquely related to their neighborhood. However, this method lack in efficient matching as it performs histogram matching globally whereas in our method the histogram matching is performed locally and then transformation parameter is calculated globally.

There are many challenges that most of the matching algorithms have to resolve while matching the fingerprints[1-6]. At first we consider the case where noise is added when the feature points are extracted from the input image because of the different devices present. Secondly, random miss-

ing of feature points is also very usual when the fingerprint is taken by device. Sometimes the lossy impression is captured when user place their finger partially. Lastly, with the change in the DPI (Dots per Inch) of scanner or input devices the scaling problem arises.

In this paper we have tried to overcome all these problems in fingerprint matching so as to make the algorithm robust. The histogram of the pair wise distances is calculated locally for each point in it's neighborhood for given input pattern. After matching of histogram the transformation parameter is obtained and then iteration algorithm is applied to find the final transformation parameter globally. Algorithm has been discussed in detail in section 4 with flow diagram.

## 2. PROBLEM FORMULATION

Let us consider two fingerprints of same finger but on different times to match, say one of point pattern (say P) is already save in database, considering as a template pattern and other one is taken as the input point pattern (say Q) as shown in Fig. 1. The pattern Q can have some scaling, rotation, translations changes, missing points and distortions (noise) with respect to template data P depending on the situation. In the form of solution, we have to find a transformation parameter "T" which consists of scaling(s), rotation ( $\theta$ ) and translation ( $t_x$  and  $t_y$ ) in both X and Y direction, in pattern Q with respect to P.

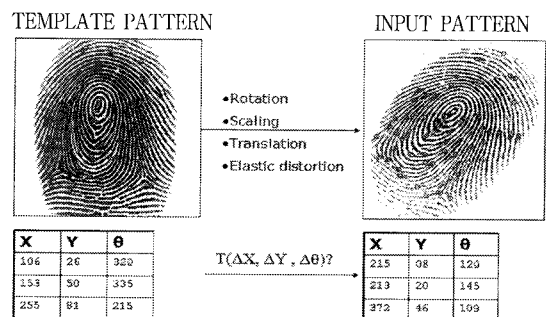


Fig. 1. Template Finger print and Input Finger print.

If we map the points in  $P$  with these parameters( $s$ ,  $\theta$ ,  $t_x$  and  $t_y$ ) according to equations (1), the generated points should almost overlap on the points in input  $Q$  pattern with some considerable error.

$$q = T(p) \Rightarrow \begin{pmatrix} x_q \\ y_q \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} s \cos \theta & -s \sin \theta \\ s \sin \theta & s \cos \theta \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix} \quad (1)$$

Considering the possibility of allowable error,  $T(p)$  may not be exactly equal to the points in pattern  $Q$ . The objective of this paper is to find a registration, so that the correct number of matched pair between  $P$  and  $Q$  is maximum.

### 3. PAIR WISE DISTANCE DISTRIBUTION

In case of point patterns, pair wise distance distribution is a set of mutual distances between the pair of points. These pairs are made by the point and its neighborhood as shown in the Fig. 2[7]. To get this distribution function, first of all we calculate the " $k$ " nearest neighbors (in order to closeness) of all points in given point patterns  $P$  and  $Q$  as shown in Fig 2, where  $k$  is given by [3]

$$k \approx \frac{\ln m}{2(\rho - \lambda^2/4)^2} \quad (2)$$

where  $m$  is number of points in given pattern,  $\rho$  and  $\lambda$  are matching parameters.

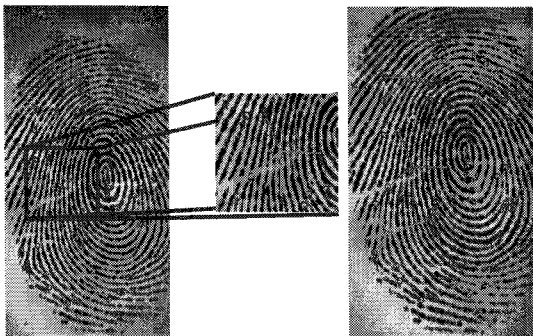


Fig. 2. Selection of nearest neighbors in input point pattern ( $k=5$  is considered in this case).

Suppose for a point  $p_i$  in  $P$  pattern neighbors are  $p_{i1}, p_{i2}, p_{i3}, \dots, p_{ik}$  and similarly for points in  $Q$  pattern are  $q_{i1}, q_{i2}, q_{i3}, \dots, q_{ik}$ . Now pair wise distances between the point set  $(p_i, p_{i1}, p_{i2}, p_{i3}, \dots, p_{ik})$  is calculated. These distances will be  $(p_i \rightarrow p_{i1}, p_{i1} \rightarrow p_{i2}, p_{i2} \rightarrow p_{i3}, \dots, p_{ik} \rightarrow p_{i1})$ , here  $i$  varies from 1 to  $k$ . In order to obtain the histogram, we quantize all the distances in some certain levels depending on the maximum and the minimum pair wise distances calculated and then assign the levels to all distances. Distance histogram for points  $p_i$  and  $q_i$  are shown in Fig. 3. In the same way, the distance histogram can be determined for all points in pattern  $P$  and  $Q$ . One of the important features of this method is its scale invariant property. DPI for a fingerprint is depending on the device used on the time of capturing. So different device is used DPI is also different, while these fingerprints are matched, scaling in the feature point is figured out. But the histogram difference for the pattern scaled by any factor is independent of this change (explained in section 4).

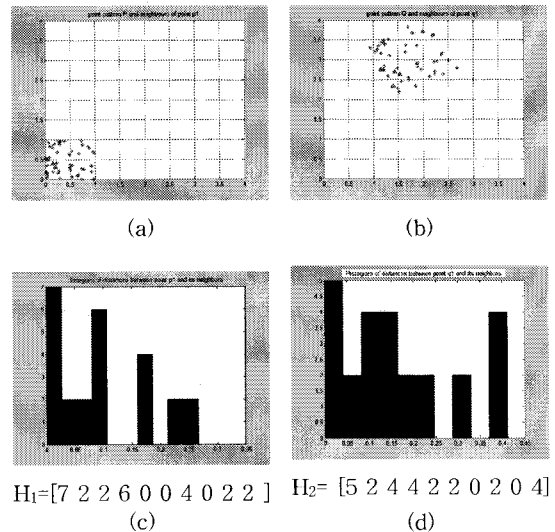


Fig. 3. (a) Point pattern  $P$ , (b) Point pattern  $Q$  scaled by factor 1.4 with respect to pattern  $P$ , (c), (d) Pair wise distance histogram of one point and its neighborhood in point sets  $P$  and  $Q$ .

## 4. ALGORITHM

This matching algorithm performs in two sections mainly: (i) Local matching, (ii) Global matching.

### 4.1 Local matching

This is a randomization algorithm. But, in order to make the calculations less in comparison to other randomization algorithms, we have a basis of selection for local match here. As a requirement of this section we get the histogram for both patterns as per description in section 3. Choose any point from pattern P and compare its histogram with that of points in pattern-Q. To decide the local matching between the points average histogram difference is calculated by subtracting the histogram of one from other. These histogram differences are scaling invariant because in calculation of histogram step size, we have used difference between maximum and minimum distances as given in formula

$$\text{Step size} = (\text{maximum distance} - \text{minimum distance}) / n_s$$

where  $n_s$  is number of steps of quantization. Further histogram differences also come in the form of matrices as  $[c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7 \ c_8 \ c_9 \ \dots \ c_{n_s}]_{1 \times n_s}$ . Where  $c_1, c_2, \dots$  are values of histogram difference for each quantization level, after subtracting two histograms and these will come out in order of 0,  $\pm 2$ ,  $\pm 4 \dots$  etc., as shown in Fig. 3 (c) and (d). So there is no need to calculate normalization factor in case of scaling changes. For the histograms shown in Fig. 3(c) and (d) histogram difference is  $(H_1 - H_2)$

$$H_1 - H_2 = [7 \ 2 \ 2 \ 6 \ 0 \ 0 \ 4 \ 0 \ 2 \ 2] - [5 \ 2 \ 4 \ 4 \ 2 \ 2 \ 0 \ 2 \ 0 \ 4]$$

$$H_1 - H_2 = [2 \ 0 \ -2 \ 2 \ -2 \ -2 \ 4 \ -2 \ 2 \ -2]$$

If average histogram difference of two points is less than a fixed threshold level, the points are lo-

cally matched and other point-pairs are discarded. Now take these locally matched point-pairs one by one and calculate the affine transformation parameter. For minimizing the possibility of error the transformation parameter is calculated with the help of least square error method, as described in [2]. Equation (3) describes the complete solution to find the transformation parameter "r" for a particular locally matched pair. In the equation subscript A is concerned to template pattern and B is concerned to input pattern. x and y are 2-D co-ordinates of feature points and k is the number of nearest neighbors calculated. For other unknowns, definition has been given below,

$$r = \frac{1}{D} \begin{pmatrix} l_A & 0 & -\mu_{xA} & \mu_{yA} \\ 0 & l_A & -\mu_{yA} & -\mu_{xA} \\ -\mu_{xA} & -\mu_{yA} & l & 0 \\ \mu_{yA} & -\mu_{xA} & 0 & l \end{pmatrix} \begin{pmatrix} \mu_{xB} \\ \mu_{yB} \\ l_{A+B} \\ l_{A-B} \end{pmatrix} \quad (3)$$

where

$$\begin{aligned} \mu_{xA} &= \sum_{i=1}^k x_{a_i}, & \mu_{xB} &= \sum_{i=1}^k x_{b_i} \\ \mu_{yA} &= \sum_{i=1}^k y_{a_i}, & \mu_{yB} &= \sum_{i=1}^k y_{b_i} \\ l_{A+B} &= \sum_{i=1}^k (x_{a_i} x_{b_i} + y_{a_i} y_{b_i}), & l_{A-B} &= \sum_{i=1}^k (x_{a_i} y_{b_i} - y_{a_i} x_{b_i}) \\ l_A &= \sum_{i=1}^k (x_{a_i}^2 + y_{a_i}^2) \\ D &= l_A - \mu_{xA}^2 - \mu_{yA}^2 \end{aligned}$$

The solution comes in the form  $r = (t_x \ t_y \ \cos \theta \ \sin \theta)^T$ . With these parameters we can calculate the transformation parameters for all locally matched point-pairs of whole the pattern. All these transformation parameter for locally matched points are shown in Fig. 4. The figure shows that many points are concentrated on same point, which has been enlarged in the rectangular section. Actually this point represents the transformation parameters from correct and true matching pairs only. An ultimate Transformation Parameter ( $s$ ,  $\theta$ ,  $t_x$  and  $t_y$ ) has to be determined for global match.

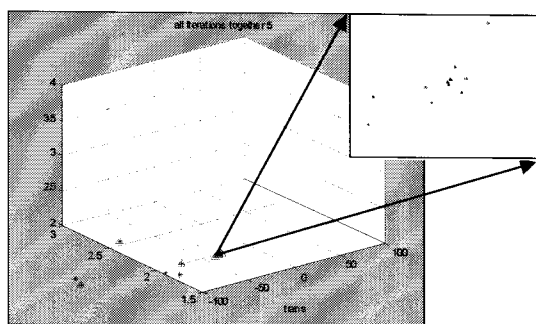
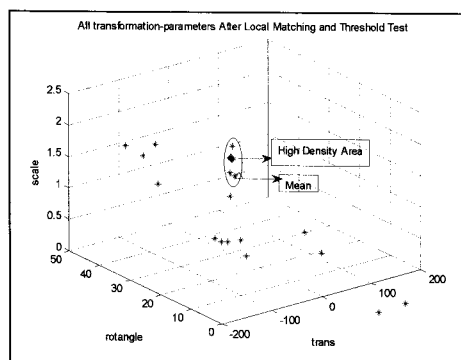


Fig. 4. (a) All transformation parameters, (b) Final transformation parameter after iteration.

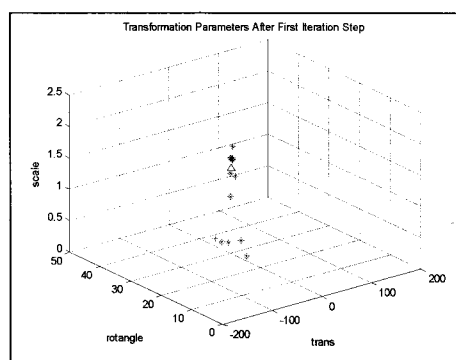
## 4. 2 Global matching

Global matching will decide whether the two point patterns are from the same finger or not. In order to perform this section, plot all local matching transformation parameter obtained in previous

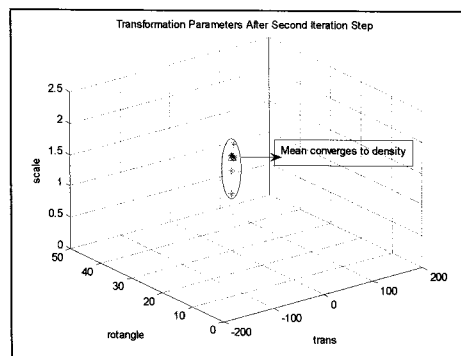
section, and then calculate the mean and standard deviation of all parameters. Iteration Algorithm is applied to the means and deviations repetitively until the distribution with minimum possible standard deviation is found. Mean calculated in the previous step gives the final transformation parameter for maximum matching pairs. These iteration steps are shown in Fig. 5. In every step some points are eliminated which are having standard deviation more than calculated in previous stage. This process is continued until final transformation parameter is found. After several iterations we have the final transformation parameter as shown in Fig. 5(d). With this parameter the overlapping test is done again and the points having allowable error are considered as properly matched pair others are discarded. If the number of properly over-



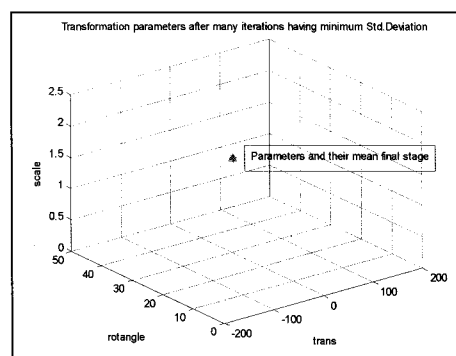
(a)



(b)



(c)



(d)

Fig. 5. (a) All transformation parameters calculated from local matching pairs, (b) Transformation parameters and their mean after first iteration. (c) Transformation parameters and mean after second iteration. (d) Transformation parameters and their mean after final iteration stage.

lapped points is more than 80% of total points in pattern, then fingerprints are declared as globally matched patterns otherwise not. Flow chart in Fig. 6 depicts the algorithm steps more clearly.

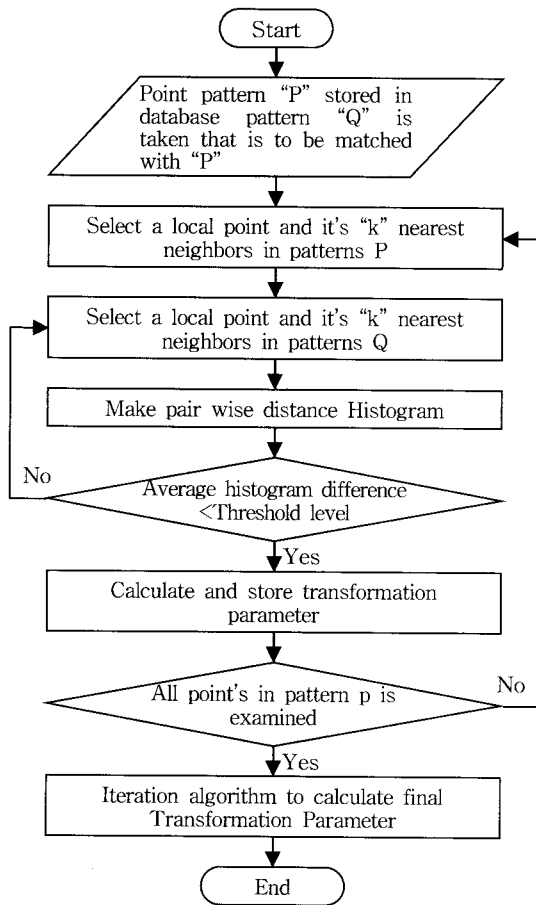


Fig. 6. Flow chart for fingerprint matching algorithm.

## 5. ADVANTAGES OF THE ALGORITHM

If we talk about the salient features of this algorithm, which make it robust and easy to implement are:-

1) The algorithm is invariant to missing feature points and effect of local noise. For translation, rotation and scale changes by any factor occur in pattern, method is robust as well.

2) Among all randomization algorithms, in the local matching step, a huge calculation has to be done. But, in this method we have to find the local matching pairs on the basis of similarity in distance histogram in their neighborhood. So less calculation is required.

## 6. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Experiments were made on many randomly generated data sets having different no. of points. These data sets are generated by adding scaling, rotation, translation changes, missing points and other distortions in original point set. The performance of algorithm is best unless number of missing points is less than 30% of total number of points. The local noise also affects the performance, when it is higher than the minimum pair wise distance in the input pattern. Table 1 shows the performance of proposed algorithm on some random point pattern samples.

Table 1. Performance on randomly generated data sets.

No.	Total no of point taken	No of missing points	No. of matched points when no error added	No. of matched points when 2% error added	No. of matched points when 3% error added
1	30	10	20	19	17
2	40	15	25	23	21
3	50	20	30	28	26
4	60	15	45	41	40
5	60	20	40	36	34
6	60	25	35	28	27

To test the algorithm on a real data, we applied this method to match a fingerprint recognition problem. The two fingerprints shown below in Fig. 7 are taken from the FVC 2000 database. The matching is performed by this method after the feature points extracted as shown by dots. The results are as shown in Fig. 8.

From Fig. 8, the performance of the method is clear that only matching points are overlapping to each other. The missing points are also present in the figure, the places having only “ $\Delta$ ” and no “\*” is enclosed to them. In this input pattern of 45 points, 5 points are missing with respect to template pattern of 50 points, and 39 points are showing perfect match. The performance of this algorithm was tested over 78 real fingerprint patterns

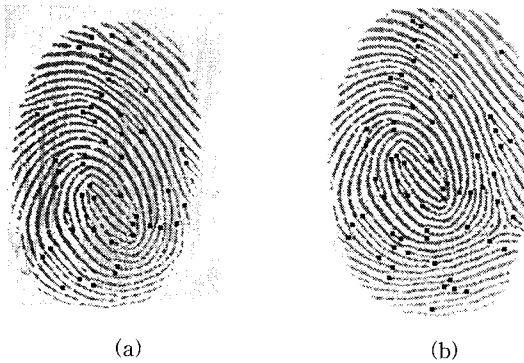


Fig. 7. (a), (b) Real fingerprint data taken from the same finger.

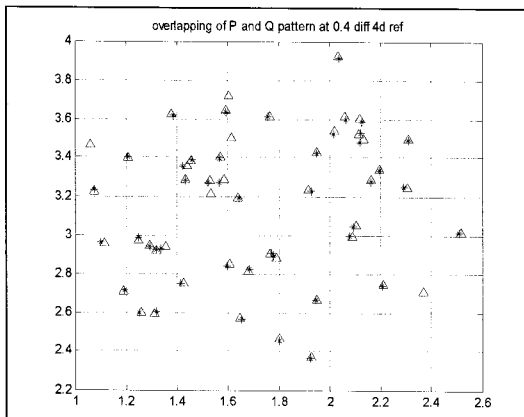


Fig. 8. Result of real fingerprint matching.

from the same database. To evaluate the performance of algorithm, a new method has been proposed here in which the basis of calculation is the number points included by a particular area. One assumption is made that the feature point distribution is uniform in all patterns. Following mathematical equations can describe the method very well.

Consider that  $A_t$  is the area of fingerprint stored in template pattern, it is given by

$$A_t = \pi d_t^2 / 4$$

where  $d_t$  is the maximum pairwise distance in the whole template pattern and similarly  $A_s$  is the area of sample fingerprint to be matched with template one, it can be calculated as

$$A_s = \pi d_s^2 / 4$$

where  $d_s$  is the maximum pairwise distance in the whole sample pattern.  $N_t$  and  $N_s$  are the total no of points in these patterns. Therefore the no of points in the template pattern included by area corresponding to sample pattern is  $N_c$ , i.e.

$$N_c = (A_s / A_t) * N_t$$

And the matching score  $S$ , is given by in final equation

$$S = \text{Number of matched points} / N_c$$

## 7. CONCLUSION AND FUTURE WORK

In this paper, an enhanced fingerprint matching method has been proposed. In this method, missing point consideration has been taken besides other important features of this method as it is a scaling, rotation and transformation invariant algorithm. This method is far efficient, faster and accurate in comparison of most of the methods proposed yet.

We have tested this matching algorithm on various randomly generated point patterns and real fingerprint patterns as well; it gives good results even on adding a reasonable amount of distortion (noise). Here for calculation of transformation parameters, we have used least square error method. Although this method is efficient, for higher accu-

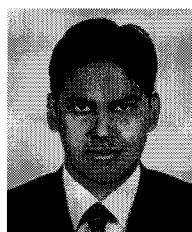
racy we need to have a more accurate method. The reason is that this method has limitations about number of neighbors etc. One more important thing to be developed is a feature extraction algorithm. The error in feature points extraction may result in false match of two point patterns.

## REFERENCES

- [1] D. Maltoni., D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer 2003.
- [2] S. H. Chang, F. H. Cheng, W. H. Hsu, and G. Z. Wu, "Fast algorithm for point pattern matching: Invariant to translation, rotations and scale changes," *Pattern Recognition*, Vol.30, No.2, pp. 311-320, 1997.
- [3] P. B. Van Wamelen, Li Z., and S. S. Iyengar, "A fast expected time algorithm for the 2-D point pattern matching problem," *Pattern Recognition*, Vol.37, pp. 1699-1711, 2004.
- [4] S. Irani and P. Raghavan, "Combinatorial and experimental result on randomized point matching algorithms," *Proceeding of the 12th Annual ACM symposium on computational geometry*, Philadelphia, PA, pp. 68-77, 1996.
- [5] C. H. Park, Mark J. T. Smith, M. Boutin, and J.J. Lee, "Fingerprint matching using the distribution of the pairwise distance between minutiae," *Lecture Notes in Computer Science*, Vol.3546, pp. 693-701, 2005.
- [6] Koji Sakata, Takuji Maeda, Masahito Matsushita, Koichi Sasakawa, and Hishashi Tamaki, "Fingerprint Authentication based on

matching scores with other data," *Lecture Notes in Computer Science*, Vol.3832, pp. 280-286, 2006.

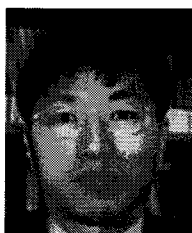
- [7] S. Chikerur and Nalini Ratha, "Impact of singular point detection on fingerprint matching performance, *Fourth IEEE Workshop on Auto-matic Identification Advanced Technologies*, pp. 207-212, Oct. 2005.



**Neeraj Sharma**

2000~2004 U.P. Technical Univ.  
(Lucknow), India (B.Tech.)  
2004~2006 Sterling Telecom  
Limited India, ICT Engineer  
2006~2008. Dongseo Univ.,  
Busan, Korea (MS)

Areas of Interest : Image processing and Fingerprint recognition



**Joon Jae Lee**

1986. 8 Kyungpook Nat'l. Univ.  
(BS)  
1990. 8 Kyungpook Nat'l. Univ.  
(MS)  
1994. 8 Kyungpook Nat'l. Univ.  
(Ph.D)  
1998. 3~1999. 2 Georgia

Institute of Technology, Visiting Professor  
2000. 3~2001. 2 PARMi Corporation, Research  
Manager  
1995. 3~2007. 8 Dongseo Univ., Associate Professor  
2007. 9~ Keimyung Univ., Associate Professor  
Areas of Interest : Image processing, Computer vision,  
and Fingerprint recognition