https://doi.org/10.1109/ITNG.2008.252

Hanmandlu, M. and Kumar, Amioy and Madasu, Vamsi K. and Yarlagadda, Prasad (2008) Fusion of Hand Based Biometrics using Particle Swarm Optimization. In Latifi, Shahram, Eds. *Proceedings Fifth International Conference on Information Technology: New Generations*, pages pp. 783-788, Las Vegas, USA.

Fusion of Hand Based Biometrics using Particle Swarm optimization

M. Hanmandlu
Dept. of Elec. Eng.
I.I.T. Delhi, India
mhmandlu@ee.iitd.ac.in

Amioy Kumar
Dept. of Elec. Eng.
I.I.T. Delhi, India
amioy.iitr@gmail.com

Vamsi K. Madasu
School of Eng. Systems
QUT, Australia
v.madasu@qut.edu.au

Prasad Yarlagadda
School of Eng. Systems
QUT, Australia
y.prasad@qut.edu.au

Abstract

Multi-modal biometrics has numerous advantages over uni-modal biometric systems. Decision level fusion is the most popular fusion strategy in multimodal biometric systems. Recent research has shown promising performance of hand based biometrics, i.e. palmprint and hand geometry over other biometric modalities. However, the improvement in performance is constrained by the lack of optimal sensor points and fusion strategy. In this paper, we have implemented a particle swarm based optimization technique for selecting optimal parameters through decision level fusion of two modalities: palmprint and hand geometry. The experimental evaluation on a database of 100 users confirms the utility of the decision level fusion using particle swarm optimization.

Keywords: modalities, Biometrics, palmprint, hand geometry, PSO, fusion, rules.

1. Introduction

Biometric systems suffer from several problems like noisy sensor data, non-universality, lack of individuality, non-availability of invariant representations, etc, [1]. These problems are responsible for an increase in error rates and decrease in system reliability for high security needs. Multimodal biometric systems overcome some of the problems associated with unimodal biometric systems by combining the decisions from different biometrics using an effective fusion rule, thus achieving higher accuracy and better performance.

The fusion in multimodal systems can be performed at four major levels: sensor, feature, score and decision. The first two levels of fusion are preferable to conduct prior to matching, while the other two levels can take place during the fusion after matching. Fusion after matching is split up into four categories: dynamic classifier fusion, decision level fusion, rank level fusion and score level fusion. Dynamic classifier selection scheme works upon the idea of choosing certain input pattern that is likely to give the most correct decisions [2]. The rank level fusion is achieved by sorting the possible matches given by each biometric matcher in a decreasing order of confidence [5]. The score level fusion is performed by combining the matching scores of different matchers. It involves the matching of scores generated by the features of the biometrics by different sensors and fusion of these scores by sum, product, and weighed sum rules. The features of individual matcher can be classified into one of the two classes: Genuine (Accept) or imposter (Reject). These classifiers are then used to make decisions. The system error rates can be represented in terms of F_{AR} (False acceptance rate) and F_{RR} (False rejection rate). The decision level fusion comes into action when individual matcher presents its decisions based on its input patterns. Each classifier under the binary hypothesis gives its decision based on its input pattern. The classifier decisions are further fused under some rule like, majority voting rule [3] or Chair-Varshney [4] fusion rule.

Fusion strategies are an important aspect of any multimodal biometric system. These strategies help us to choose some optimal rule for the fusion of multimodal biometrics. Some of the approaches that employ an optimal fusion are: Deterministic methods, Probabilistic methods, and Evolutionary methods. The deterministic methods involve an application of some traditional heuristic approaches like, trajectory methods which modify trajectories for optimization, penalty methods which imposes penalties for optimal decisions, etc. The probabilistic methods rely upon probabilistic judgments to yield an optimal decision [10]. In comparison to different adaptive stochastic search algorithms, Evolutionary Computations (EC) techniques [11] generate a set of relevant solutions, called *population* and then find an optimal solution through searching and updating the past history of the particles (i.e. memories) of the population. Some of the examples of such approaches are: Genetic Algorithm (GA), Swarm Intelligence (SI) [8], Ant Colony Optimization (ACO), Bacteria Foraging (BF), etc.

2. Background Work

There has been a lot of interest in multimodal biometric systems. Frischholz et al. [7] proposed a multimodal system called *BioID* based on the fusion of face, voice and lip movement. They chose different fusion strategies in order to vary the security levels. However, their algorithm is restricted to only a few fusion rules, typically the AND and OR rules. Their system has fixed threshold values and hence yields the fixed error rates and reduces one of the error rates successfully but not both. Jain et al. [17] proposed the integration of more than one matcher for the fingerprint verification system and developed a decision level fusion for fingerprints by combining four different matching algorithms.

978-0-7695-3099-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ITNG.2008.252

Despite proven efficiency of multimodal fusion, only few works have been reported till date. Moreover attempts on decision level fusion using optimization techniques based on social behavior of individuals are comparatively new. Kalyan et al. [9] developed an adaptive multimodal biometric management algorithm for multisensory fusion by combining biometric modalities. This algorithm can adaptively select the optimal Bayesian fusion rule as well as the individual sensor operating points. The algorithm not only reduces both error rates but also yields a broad range of fusion rules to combine the biometric. However for the experimental evaluation, they used simulated data and generated the Gaussian distribution using mean and standard deviation of the genuine and imposter scores. The present work is influenced by this approach.

3. The proposed modalities

The biometric modalities considered in the paper are palmprint and hand geometry. One of the key objectives of this work is to evaluate the usage of palmprint and hand geometry for the decision level multimodal fusion. Despite the recent popularity of palmprint based systems [19], there have been no attempts on PSO based decision level fusion. We therefore investigate the possible uses of palmprint and hand geometry for the adaptive multimodal biometric management algorithm (AMBMA) described in [9]. Each biometric involves feature extraction, matching, and decision making. The PSO algorithm fuses the single modality decisions.

3.1. Palmprint

The palmprint features employed in this work are extracted using Discrete Cosine Transform (DCT). The palmprint image database from 100 users, with 10 samples per user, is used to show the performance of the fusion. The discrete cosine transform based 144 features from each of the palmprints, using 24 × 24 pixels block with an overlapping of 6 pixels, are extracted. The feature extraction from each of these 300 × 300 pixels palmprint images is similar to that in [18]. These features are then used to calculate genuine and imposter scores using similarity measure and by taking the first five images for training and the rest five for testing for each user. The error rates are generated using different threshold values.

3.2. Hand Geometry

Hand geometry is the geometry of the hand image with palm and fingers. The features of the hand geometry are represented by the length of fingers, distances between knuckle points, height and thickness of the hand and the fingers etc. This is an important biometric in multimodal fusion as it is extremely user friendly and requires a very low cost acquisition system. The hand geometry database consists of 100 users, with 10 images each having 23 extracted features. The genuine and imposter scores are calculated using distance similarity. The error rates are generated by setting different thresholds values.

4. Decision Making

A classifier can make its decision in binary mode according to the hypothesis testing approach. Let the stored biometric template be represented by T and the input template for authentication be represented by I. The null and the alternate hypothesis are:

$$H_0:\ T \neq I \text{ the person is an imposter.} \qquad (1)$$
$$H_1:\ T = I \text{ the person is genuine.}$$

The two associated decisions denoted by:

$$s_i = 0, \text{ the person is an imposter.}$$
$$s_i = 1, \text{ the person is genuine.} \qquad (2)$$

The most likely decisions are genuine acceptance and imposter rejection. These are difficult to realize in practice. Hence the accuracy in decisions is specified in the terms of error rates: False rejection rate (F_{RR}) and false acceptance rate (F_{AR}). These terms are defined in terms of conditional probabilities as:

$$F_{AR_i} = P(s_i=1/H_0). \qquad (3)$$

$$F_{RR_i} = P(s_i=0/H_1). \qquad (4)$$

The decision concerning a person's genuineness is made through the following likelihood ratio test:

$$\frac{P(s_i/H_1)}{P(s_i/H_0)} \gtrless_{s_i=0}^{s_i=1} \lambda_i. \qquad (5)$$

where, λ_i is an appropriate threshold that should be set depending upon sensor's performance criteria.

4.1. Binary Fusion

The decisions made by the biometric sensors are binary based on their presence or absence and hence they need to be fused by some binary fusion rule. Let N be the number of sensors and their binary decisions be dented by s_i, i= 1,2,3,….,N. The binary decisions are given by:

$$s_i = 0 \text{ if } i^{th} \text{ sensor decides for } H_0.$$
$$= 1 \text{ if } i^{th} \text{ sensor decides for } H_1. \qquad (6)$$

All the decisions made by sensors are treated as binary strings of length: $L=\log_2(p)$. (7)

where, $p= 2^{2^N}$ is the number of possible rules for N sensors. The fusion rule R_i is an integer of length L varying from $0 \le R_i \le p-1$ For N input sensors the output is a fusion rule as shown in Fig. 1. The final decision R_i can be made in p possible ways and is subject to the desired performance. The most frequently used fusion rules are AND rule and OR rule [6]. In the AND rule the output decision is 1 if and only if all the input decisions are one.

$R_2 = 1 \quad \forall \wedge s_i = 1.$ (8)
$= 0$ otherwise.

In the OR rule the output decision is 1 if any one of the input sensor's decision is 1:

$R_8 = 1 \quad \forall \vee s_i = 1.$
$= 0$ otherwise. (9)

The 16 fusion rules for 2 sensors are shown in Table 1. The rule R_2 represents the AND rule while the rule R_8 represents the OR rule. The rule R_1 is selected when all the decisions are zero, i.e. all modalities rejected and the rule R_{16} is selected when all modalities accepted. The rule R_3 gives the acceptance of the first sensor while the rule R_7 accepts the second sensor.

Table1. Fusion rules for two sensors

s_1	s_2	R_1	R_2	R_3	R_4	R_5	R_6	R_7
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1
1	0	0	0	1	1	0	0	1
1	1	0	1	0	1	0	1	0

R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	d
0	1	1	1	1	1	1	1	1	d_0
1	0	0	0	0	1	1	1	1	d_1
1	0	0	1	1	0	0	1	1	d_2
1	0	1	0	1	0	1	0	1	d_3

4.2. Multi-Modal Fusion

If N=3, p=256 requires many rules. To circumvent this problem, two-modal fusion is extended to the case of multi-modal fusion. Let R_i^1 be the fusion rule selected for the two sensors s_1 and s_2. Consider now the availability of a third sensor s_3. With R_i^1 and s_3 we can generate the second- level 16 fusion rules denoted by R_i^2. Out of this one fusion rule is selected by optimization technique. Taking the selected fusion rule and the fourth sensor s_4 we form the next level 16 combinations and then select one from them. Hence, this procedure which may be coined "hierarchical" is continued for any number of sensors. The burden of computation is considerably reduced each time dealing with only 16 rules. With this, we require only 48 rules to be checked for 4 modalities. However this approach is suboptimal, but for optimality we need to try the combination of input error rates.

4.3. Optimal Fusion Rule

One of the tasks of decision level fusion is to select an optimal fusion rule that minimizes the total errors of the system. There are 16 possible fusion rules corresponding to two sensors but most of them have no significant role to play in the improvement of performance. Only monotonic rules need to be selected as they are shown to yield better performance experimentally [9]. The most frequently used rules are AND (R_2) rule and OR (R_8) rule. The worst performing rule is NAND rule (R_9) which is rarely of interest. The individual error rates fused by AND rules are as follows: $F_{AR}=F_{AR_1} * F_{AR_2}$ and

$$F_{RR}=F_{RR_1}+F_{RR_2}-F_{RR_1}*F_{RR_2} \tag{10}$$

This rule can improve F_{AR} but degrades F_{RR} and hence GAR. The OR rules can be opted for the reverse effect. Fusion by OR rule leads to:

$$F_{AR}=F_{AR_1}+F_{AR_2}-F_{AR_1}*F_{AR_2}$$

$$F_{RR}=F_{RR_1}*F_{RR_2} \tag{11}$$

In Table 1, s_1 is the decision of the first sensor while s_2 is the decision of the second sensor. The global decisions d_i arise from the fusion rules in Table 1. These global decisions result in the global error rates given by:

$$GF_{AR}=\sum_{i=0}^{L-1} d_i \times \left(\prod_{j=1}^{N} \phi_{AR_j} \right) \tag{12}$$

where, $\varphi_{AR_j} = 1-F_{AR_j}$ (s_j=0). & $\varphi_{AR_j} = F_{AR_j}$ (s_j=1).

$$GF_{RR} = \sum_{i=o}^{L-1} (1-d_i) \times \left(\prod_{j=1}^{N} \varphi_{RR_j} \right) \tag{13}$$

where, $\varphi_{RR_j} = F_{RR_j}$ (s_j=0) & $\varphi_{RR_j} = 1-F_{RR_j}$ (s_j=1)

These global error rates can be evaluated by any fusion rule like majority voting rule or Chair-Varshney fusion rule to arrive at an optimal decision.

5. Need for Optimization

The goal of a fusion system is to minimize the errors, F_{AR} and F_{RR} by using their weighted sum. The design of the system is such that it should itself choose the optimum decision fusion rule (Table 1) using the Bayesian framework. The fusion rules are used to calculate global error rates GF_{AR} and GF_{RR}, which in turn are used to calculate the weighted sum in Eqn. (14), where the weights are the associated costs with these errors. The optimization technique must determine the optimal sensor points adaptively. The objective function E required to optimize is defined as follows:

$$\text{Minimize } E = C_{FA} * GF_{AR} + C_{FR} * GF_{RR} \tag{14}$$
$$C_{FR} = 2 - C_{FA}$$

C_{FA} is the cost of falsely accepting an imposter individual. C_{FR} is the cost of falsely rejecting the genuine individual.

The error rates (F_{AR} and F_{RR}) for both sensors become input to the optimization technique. The objective function E should be minimized at each step by selecting one set of error rates. The optimal values correspond to the minimum E. In this work, particle swarm optimization technique is used to arrive at the optimal fusion rule and sensor points (error rates).

5.1. Particle Swarm Optimization

Particle swarm optimization (PSO) was proposed by Eberhart and Kennedy in [13] for the solution of optimization problems using social and cognitive behavior of swarm. In PSO each particle has some velocity according to which it moves in the multi-dimensional solution space; and memory to keep information of its previous visited space. Hence, its movement is influenced by two factors: the local best solution due to itself and the global best solution due to all particles participating in the solution space. The algorithm is guided by two factors: the movement of particles in the global neighborhood and the movement in the local neighborhood. In the global neighborhood each particle searches for the best position (solution) and towards the best particle in the whole swarm while in the local neighborhood, each particle moves towards the best position (solution) towards the best particle in the restricted neighborhood (swarm). During an iteration of the algorithm, the local best position and the global best position are updated if better solution is found and the process is repeated till the desired results are achieved or specified number of iterations are exhausted.

Let us consider an N-dimensional solution space. The i[th] particle of the swarm can be represented as an N-dimensional vector, $X_i = (x_{i1}, x_{i2}, .., x_{iN})$ such that $X_{id} = x_{id}$, where the first subscript denotes the particle number and the second subscript denotes the dimension. The velocity of this particle is denoted by a N-dimensional vector, $V_i = (v_{i1}, v_{i2}, .., v_{iN})$ such that $V_{id} = v_{id}$. The memory of the previous best position of the particle is represented by an N-dimensional vector $Pos_i = (p_{i1}, p_{i2}, ..., p_{iN})$ such that $Pos_{id} = p_{id}$ and the global best position by $Pos_g = (p_{g1}, p_{g2}, ..., p_{gN})$ such that $Pos_{gd} = p_{gd}$. The particle's motion is affected by its own best position and global best position.

The velocity of a particle at *k* th iteration is updated by:

$$V_{id}^{k+1} = \omega V_{id}^{k} + r_1\alpha\left(Pos_{id} - X_{id}^{k}\right) + r_2\beta\left(Pos_{gd} - X_{id}^{k}\right) \qquad (15)$$

The corresponding position of the particle is updated by:

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \qquad (16)$$

where, i = 1,2,3…..M; M being the number of swarm and d=1,2,3,…..N is the dimension of a swarm; α and β are the positive constants, called cognitive parameter and social parameter respectively. These indicate the relative influence of the local and global positions. r_1 and r_2 are the random numbers distributed uniformly in [0 1]; and *k* =1,2,3…is the iteration step. ω is called inertia weight. In order to apply PSO approach to the fusion problem, we take the first N dimensions to be sensor thresholds λ_i that are continuous and (N+1)[th] dimension for the fusion rule, $X_{i(N+1)} = R_i$. With this each particle will have (N+1) dimensions; so that $X_i = (\lambda_{i1}, \lambda_{i2}, ..., \lambda_{iN}, R_i)$. This is an integer model because R_i takes an integer value. It suffers from slow convergence hence the need for binary PSO algorithm where F_{AR} are evolved instead of thresholds for each of the sensors, i.e., $X_i = (F_{AR_i}, R_i)$. The thresholds are computed using F_{AR}. The binary PSO not only leads to the optimal convergence with high accuracy but is also capable of making binary decisions [12] unlike others.

5.2. Binary PSO

The original PSO is for continuous population but is later extended by Kennedy and Eberhart [13] to the discrete valued population. In the binary PSO thus emerged, the particles are represented by binary values (0 or 1). The velocity and particle updating for binary PSO are the same as in the case of continuous one. However, the final decisions are made in terms of 0 or 1. Sigmoid function in [15] is used to restrict the decision in the range [0,1] :

$$v_{ri}^{k+1} = \frac{1}{1 + e^{-v^{k+1}}} \qquad (17)$$

The particles change positions called states from 0 to 1 or vice versa satisfying the condition:

$$X_i^{k+1} = 1 \text{ if } r < v_{ri}^{k+1}$$
$$= 0 \quad \text{otherwise.} \qquad (18)$$

where, *r* is the random number generated in the range [0, 1]. Now the binary fusion rule comes handy to fuse the decisions given by the individual matchers. The optimal fusion rule is selected with the use of binary PSO that sets the appropriate parameters. We will now discuss the effect of parameters on the algorithm.

5.3. Parameters of PSO

The convergence and performance of PSO are largely dependent upon parameters chosen. ω is termed as inertia weight [15] and is incorporated in the algorithm to control the effect of the previous velocity vector of the swarm on the new one. It facilitates the trade-off between the local and the global exploration abilities of the swarm and may result in less number of iterations of the algorithm while searching for an optimal solution. It is experimentally found that inertia weight ω in the range [0.8, 1.2] yields a better performance [14]. It is initially set to 1.2 and then decreased to zero during the subsequent iterations. This scheme of decreasing inertia weight is found to be better than the fixed one [18]. It controls the rapid motion of the particle while searching for optimum from region to region. The velocity lies in the range [-$\boldsymbol{V}_{\mathbf{max}}$, $\boldsymbol{V}_{\mathbf{max}}$] where

- $V_{\max}$ denotes the lower range and $V_{\max}$ is the upper range of the motion of the particle.

The roles of α and β are not so critical in the convergence of PSO, however, a suitably chosen and fine tuned value can lead to a faster convergence of the algorithm. A default value of $\alpha = \beta = 2$ is suggested for general purpose, but somewhat better results are found with $\alpha = \beta = 0.5$ [19]. However, the values of cognitive parameter, α larger than the social parameter β are preferred from the performance point of view with the constraint $\alpha + \beta \leq 4$ [16]. In the present work, we fix $\alpha = 0.9$ and $\beta = 1$. The parameters r_1 and r_2 used to maintain the diversity of the population in (15).

The implementation of the binary PSO is a bit different from the continuous one. So switching over to binary PSO requires re-setting of the parameters. This is because the higher value of $V_{\max}$ works well for better exploration in the case of continuous PSO whereas the lower value of $V_{\max}$ will do the same in the case of binary PSO [15]. To overcome this situation we take $V_{\max} = 1$ thus specifying the range [-1, 1] for the motion of the particle in the search space.

5.4. Hybrid PSO

For biometric fusion we need optimized decisions from different sensors and a fusion rule to combine them. As the fusion rules are binary we need binary PSO for better convergence. However we use a hybrid type of PSO algorithm to reap benefits from both types. The continuous PSO is used for calculating the updates of the position and velocity of a particle and the binary PSO for the purpose of arriving at a fusion rule. Next the global error rates are calculated using the fusion rule. These error rates are further used to calculate the weighted sum serving as the objective function. The error rates and the fusion rule corresponding to the minimum weighted sum of all the sensors constitute the output of the algorithm.

5.5. Decision Rule

Once the optimal sensor points (error rates) are selected by the optimization techniques, the next step is to make use of decision making using these points as inputs. Here we use Chair-Varshney fusion rule for decision making, which is defined as:

$$\sum_{i=1}^{N}\left[s_i \log\left(\frac{1-F_{RR_i}}{F_{AR_i}}\right) + (1-s_i)\log\left(\frac{F_{RR_i}}{1-F_{AR_i}}\right)\right] \gtrless \log\left(\frac{C_{FA}}{2-C_{FA}}\right) \quad (19)$$

The weighted sum given by (19) is then compared with a threshold on the r.h.s. The output decision is 1 if the weighted sum is greater than the threshold and 0 otherwise. A user is authenticated if the output is 1 otherwise rejected. For 3 modalities Eqn. (19) has to be repeated with R_i^1 and s_3 by taking the values of $F^1{}_{AR_i}, F^1{}_{RR_i}, C^1_{FA}$ .Similar is the case with 4 modalities. The following algorithm adaptively selects the weights such that the cost function is minimized.

Adaptive Multimodal Biometric Management (AMBM) algorithm

1. Calculate the error rates (F_{AR} & F_{RR}) by fixing 1000 thresholds for each modality.
2. Initialize the error rates and costs (C_{FA} and C_{FR}) to feed into the PSO algorithm for optimal values.
3. Run the PSO algorithm till the optimal decisions and the corresponding fusion rules are obtained.
4. Fuse the decisions using Chair-Varshney fusion rule for each of the cost.
5. Repeat the process till the desired performance is achieved.

6. Results of Implementation

The proposed fusion approach is implemented on real data consisting of palmprint and hand geometry images. The database is made up of 100 users, each providing 10 images. For the experimental evaluation the first five images from each user are taken as a training set and the rest five as a testing set. We have generated the genuine and imposter scores using distance similarity. The error rates are generated by setting some thresholds. These error rates along with the random numbers are treated as particles in PSO optimization technique and optimized using the algorithm. We have considered 10 particles for optimization.

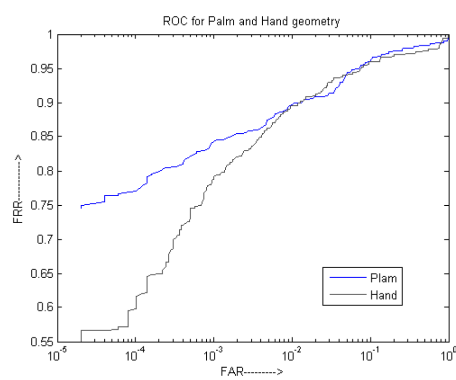

Fig. 1. The Combined ROC of Palm and Hand Geometry

Figure 1 shows the performance of both the modalities on the same curve. The objective function for the PSO algorithm is given in (14). We vary the cost of C_{FA} from 0.1 to 1.9 and run the PSO algorithm 100 times for the same cost with 1000 iterations per run. It is observed that if C_{FA} is less than 1, the OR rules is selected by the algorithm most of the times. On the other hand for C_{FA}

more than 1.5 AND rule is selected most of the times. For the costs between 1 and 1.5 both AND and OR rules are selected equally, indicating that for this cost both rules perform equally well. Figure 2 shows ROC due to both AND and OR rules.

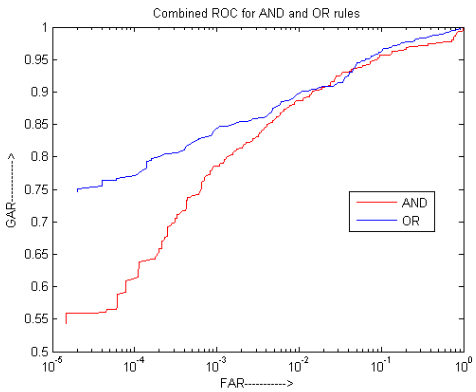

Fig.2 The combined ROC of AND and OR rule

The ROC shown above has very less improvement in terms of error rates. We recalculate the optimal sensor points using PSO and fusion strategy by varying C_{FA}. This is done in each case by combing them using AND and OR rules. Fig.3 shows the combined ROC for all the 12 points (0.1 to 1.9). It can be seen that in OR case the least G_{AR} (i.e. 1-F_{RR}) is 96% while for AND case the least F_{AR} is 10^{-6} %. Note that AND fusion rule improves G_{AR} but deteriorates F_{AR} whereas OR rule improves F_{AR} but deteriorates G_{AR} as can be seen from Fig. 3. For the given cost Chair-Varshney rule is verified thus demonstrating the applicability of decision level fusion approach using PSO. The calculated global F_{AR} and F_{RR} of both the sensors are used as input to the Chair-Varshney decision rule. The final decision is subject to satisfaction of fusion rules.

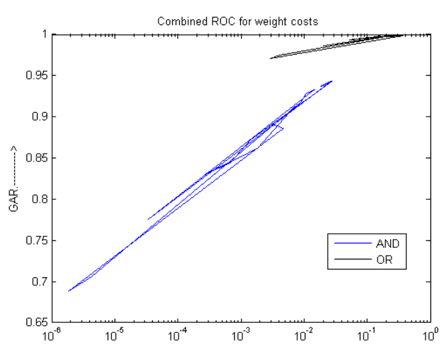

Fig.3 ROC for AND and OR rule with different C_{FA}.

7. Conclusions

A particle swarm optimization based decision level fusion of palmprint and hand geometry biometrics is presented. The sensor points and fusion rules serve as the given input to the PSO algorithm. The algorithm automatically selects the optimal sensor points and one of the 16 described fusion rules to fuse the decisions made by individual matchers. Further the global decisions are computed using Chair –Varshney rule. Extending the fusion to more than two modalities is the future work.

References

[1] K. Nandkumar, “Integration of Multiple Cues in Biometric Systems”, PHD Thesis, Michigan State University, 2005.

[2] K. Woods, K. Bowyer, and W.P. Kegelmeyer, “Combination of Multiple Classifiers using Local Accuracy Estimates”, *IEEE Trans. PAMI*, 19(4):405–410, 1997.

[3] L. Lam and C. Y. Suen, “Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance”, *IEEE Trans. SMC – Part A*, 27(5):553–568, 1997.

[4] P. K. Varshney, Distributed Detection and Data Fusion, New York: Springer, 1997.

[5] T. K. Ho, J. J. Hull, and S. N. Srihari, “Decision Combination in Multiple Classifier Systems”, *IEEE Trans. PAMI*, 16(1):66–75, January 1994.

[6] J. Daugman, “Combining Multiple Biometrics”, <http://www.cl.cam.ac.uk/users/jgd1000/combine/ >

[7] R. W. Frischholz and U. Deickmann, “BioID: A multimodal biometric identification system,” *IEEE Computer*, vol. 33, no. 2, Feb. 2000.

[8] J. Kennedy, R. C. Eberhart, and Y. H. Shi, **Swarm Intelligence**, *CA*: Morgan Kaufmann, Jun. 2001.

[9] K. Veeramachaneni, L. Osadciw, and P.K. Varshney, “An Adaptive Multimodal Biometric Management Algorithm”, *IEEE Trans. SMC —Part C*, vol. 35, no. 3, August 2005.

[10] R. Horst and H. Tuy, **Global Optimization – Deterministic Approaches**, Springer, New York, 1996.

[11] T. Bäck, D. Fogel and Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing and Oxford University Press, New York, 1997.

[12] J. Kennedy and R.C. Eberhart, “PSO optimization,” *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1941–1948.

[13] R.C. Eberhart and J. Kennedy,” A New Optimizer Using Particle Swarm Theory”, *Proc. 6th Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[14] Y. Shi and R.C. Eberhart, “A modified Particle Swarm Optimizer”, *Proc. IEEE Conference on Evolutionary Computation*, 1998.

[15] M.A. Khanesar, M. Teshnehlab, and M.A. Shoorehdeli “A Novel Binary Particle Swarm Optimization”, *Proc. 15th Mediterranean Conference on Control and Automation*, 2007.

[16] A. Carlisle and G. Dozier, “An Off-The-Shelf PSO*”*, *Proc. Particle Swarm Optimization Workshop*, pp. 1–6, 2001.

[17] A. K. Jain, S. Prabhakar, and S. Chen, “Combining multiple matchers for a high security fingerprint verification system,” *Pattern Recognition Letters.*, vol. 20, no. 11–13, pp. 1371–1379, Nov. 1999.

[18] A. Kumar, D. C. M. Wong, Helen C. Shen, and A. K. Jain, “Personal authentication using hand images”, *Pattern Recognition Letters*, vol. 27, pp. 1478-1486, Oct. 2006.

[19] A. Kumar and D. Zhang, “Palmprint authentication using multiple representation,” *Pattern Recognition*, vol. 38, pp. 1695-1704, Oct. 2005.