

2003

Adaptive Multimodal Biometric Fusion Algorithm Using Particle Swarm

Kalyan Veeramachaneni

Syracuse University, Department of Electrical Engineering and Computer Science, kveerama@syr.edu

Lisa Ann Osadciw

Syracuse University, Department of Electrical Engineering and Computer Science, laosadci@syr.edu

Pramod K. Varshney

Syracuse University, Department of Electrical Engineering and Computer Science, varshney@syr.edu

Follow this and additional works at: <https://surface.syr.edu/eecs>

 Part of the [Medical Biomathematics and Biometrics Commons](#)

Recommended Citation

Veeramachaneni, Kalyan; Osadciw, Lisa Ann; and Varshney, Pramod K., "Adaptive Multimodal Biometric Fusion Algorithm Using Particle Swarm" (2003). *Electrical Engineering and Computer Science*. 146.

<https://surface.syr.edu/eecs/146>

This Article is brought to you for free and open access by the College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Adaptive Multimodal Biometric Fusion Algorithm Using Particle Swarm

Kalyan Veeramachaneni, Lisa Ann Osadciw and Pramod K. Varshney

Department of Electrical Engineering and Computer Science

Syracuse University

Syracuse, NY 13244-1240

(315) 443-1319 (office)/(315) 443-2583 (fax)

kveerama,laosadci,varshney@syr.edu

Abstract: This paper introduces a new algorithm called “Adaptive Multimodal Biometric Fusion Algorithm” (AMBF), which is a combination of Bayesian decision fusion and particle swarm optimization. A Bayesian framework is implemented to fuse decisions received from multiple biometric sensors. The system’s accuracy improves for a subset of decision fusion rules. The optimal rule is a function of the error cost and a priori probability of an intruder. This Bayesian framework formalizes the design of a system that can adaptively increase or reduce the security level. Particle swarm optimization searches the decision and sensor operating points (i.e. thresholds) space to achieve the desired security level. The optimization function aims to minimize the cost in a Bayesian decision fusion. The particle swarm optimization algorithm results in the fusion rule and the operating points of sensors at which the system can work. This algorithm is important to systems designed with varying security needs and user access requirements. The adaptive algorithm is found to achieve desired security level and switch between different rules and sensor operating points for varying needs.

Keywords: Bayesian decision fusion, particle swarm optimization, multimodal biometrics

1. INTRODUCTION

The personal safety of the population in public and private buildings has become a larger concern since September 11, 2001. A variety of pilot projects in the area of access control based on a single biometric have been completed recently [2]. The unsatisfactory results from these projects highlight the need to improve biometric security systems to address both customer and user needs. Issues that need to be addressed in a biometric based system include performance, acceptability, and circumvention. Acceptability, referring to the population’s acceptance of biometrics in daily life, is best addressed by society’s leaders and is beyond the scope of this paper. However, system accuracy and circumvention are dependent on the biometric technology as well as system design. This was the focus of our earlier work reported in [13].

Performance of a biometric system is measured by their identifying power, which is calculated using false rejection rate and false acceptance rates. Single modality biometric identification systems force users to trade-off between these two rates, as both of them cannot be reduced simultaneously. Knowing and optimizing system’s identifying power and making sure it is acceptable for the application are critical for a system’s success.

Recently there has been a lot of interest in multimodal biometrics[3, 4, 11, 12]. In general, each biometric sensor has its own limitations and problems. Not all issues can be addressed by a single sensor. Hence integrating multiple sensors to achieve multiple objectives becomes an obvious choice. Many approaches employing fusion for personal identification have been investigated with success [3, 4, 6]. These approaches have explored using different types of sensors to collect the same biometric feature. An example is the fusion of fingerprints collected using both an optical sensor and ultrasound sensor. Others have studied system performance for multimodal biometric fusion such as face and voice. All have demonstrated performance improvement. In a system for the general population, it is paramount that a multimodal system employing fusion be available so that tailoring of the biometric collection and matching process can be accomplished to address the employee’s unique characteristics as well as access needs.

The multisensor fusion in identification systems support a variety of security levels such as, low, medium, high alert levels and potential break-in [13]. The current fusion approaches [11] often skip a number of potential decision fusion rules. This weakens the solution and results in a suboptimal system.

Bayesian decision fusion as presented in this paper formalizes the design of such systems. There are 2^{2^N} possible fusion rules for N sensors if all possible combinations of the sensor decisions are considered. Only a fraction of these rules, however, are monotonic and potentially optimal. This fraction, however, increases with the number of sensors. For example there are 20 rules for three sensors, while only six for two sensors. Apart from this, sensors can work at different operating points or thresholds, which gives an additional feature that can be changed to achieve the desired global system performance. Selection of the fusion rules and sensor operating points leads to a combinatorial explosion and, hence, searching through these two parameter sets to optimize system performance is a NP complete problem.

Particle swarm optimization (PSO) is a population based evolutionary algorithm developed by Kennedy and Eberhart [17, 18, 20, 26]. This paper proposes the use of this algorithm for solving the NP- complete problem described above. PSO is an adaptive algorithm that lends itself very well to dynamic changes. This makes it an excellent candidate for this biometric fusion problem. The swarm searches for optima in the solution space and shrinks the search area step by step. If a dynamic change occurs in the system affecting the search area, the PSO will automatically find new optimum without any modification [28]. The system design, however, is problem specific and has many implicit and explicit factors which affect its performance. This paper describes the design of the new PSO for this application, and the results demonstrate that the use of PSO leads to a more robust multimodal biometric system. The new algorithm which is a combination of Bayesian decision fusion and particle swarm optimization is called '*Adaptive Multimodal Biometric Fusion*' algorithm (AMBF).

In the next section, Bayesian decision fusion is described. Section III describes the particle swarm optimization and factors which influence it. Application of PSO to the multimodal biometric problem and setting up of the parameters of PSO are discussed in Section IV. Section V presents the results and analysis. Conclusions and future work are presented in the final section.

2. BAYESIAN DECISION FUSION

A Bayesian framework formalizes the design of a personal identification system that can adaptively increase or reduce the security level as well as adapt to each user's physical characteristics [13]. The key is to use multiple biometric modes, adapt the error costs, and vary the sensor operating points giving the system robustness and adaptability.

As a brief review, the problem of personal identification can be formulated as a hypothesis testing problem where the two hypotheses are

H_0 : the person is an imposter or

H_1 : the person is genuine.

The conditional probability density functions are $p(u_i|H_1)$ and $p(u_i|H_0)$ where u_i is the output of the i^{th} biometric sensor given the genuine person and the imposter, respectively. The decision made by sensor i is

$$u_i = \begin{cases} 0, & \text{person is an imposter} \\ 1, & \text{person is genuine} \end{cases} \quad (1)$$

This decision is made based on the following likelihood ratio test

$$\frac{p(u_i|H_1)}{p(u_i|H_0)} \underset{u_i = 0}{\overset{u_i = 1}{\gtrless}} \lambda_i \quad (2)$$

where λ_i is an appropriate threshold [14, 15, 16].

The four possible decisions are:

1. The genuine person is accepted
2. The genuine person is rejected
3. The imposter is accepted
4. The imposter is rejected.

The optimum Bayesian fusion rule allowing access to a building for N sensors is [15][16]

$$\sum_{i=1}^N \left[u_i \log \left(\frac{1 - F_{RR_i}}{F_{AR_i}} \right) + (1 - u_i) \log \left(\frac{F_{RR_i}}{1 - F_{AR_i}} \right) \right] \underset{u_g = 0}{\overset{u_g = 1}{\propto}} \log \left(\frac{C_{FA}}{2 - C_{FA}} \right) \quad (3)$$

where u_i is the local sensor decision from (1), u_g is the global decision, and N is the number of sensors. The rule in (3)

assumes an equal a priori probability of an imposter and genuine user. There is a total of 2^{2^N} possible fusion rules if all possible combinations of the sensor decisions are considered.

The AMBF algorithm finds the optimum rule based on accuracy given the user selected error costs. The algorithm selects the individual sensor thresholds to minimize errors. An assumption of Gaussian sensor noise is made so the mean and variance of the noise is all that is required by the sensor models. If new threats or sensor degradations affect the system, the algorithm can react by modifying the optimum rule in response to these changes.

Accuracy, which is the focus of this paper, refers to the rates at which the two types of errors occur: false rejection rate (FRR) and false acceptance rate (FAR). We define the error rates as

$$F_{AR_i} = P(u_i = 1 | H_0) \quad \text{and} \quad (4)$$

$$F_{RR_i} = P(u_i = 0 | H_1). \quad (5)$$

The performance of a detector is often represented in terms of receiver operating characteristics (ROC) or a plot of genuine acceptance rate versus FAR. It should be pointed out that the optimum decision rule is defined as the rule that minimizes the probability of error in the AMBF algorithm. The structure of individual biometric sensor decision rules are not modified but the decision thresholds controlling the sensor operating point are changed. The operating point selection uses the biometric sensor's ROC, which is usually available. From this, the fusion of the biometric sensor decisions is analyzed and the optimum decision fusion rules is selected.

Typically, the FAR and FRR cannot be reduced simultaneously. As the number of sensors increases and the operating points are varied, however, this restriction is vanishes. Error costs affect the rule in (3) and, consequently, the total cost. Since security is usually the prime objective, a low FAR is usually desired. The user assigns a higher cost to the FAR error in the Bayesian framework to express this security need.

Since the algorithm is developed using a Bayesian framework, a total error cost is defined as a weighted sum of the two global errors, GFAR and GFRR. The total cost, minimized by the appropriate rule, is

$$E = C_{FA} GFAR + C_{FR} GFRR \quad (6)$$

where C_{FA} is the cost of falsely accepting an imposter individual, C_{FR} is the cost of falsely rejecting the genuine individual, GFAR is the global F_{AR} , and GFRR is the global F_{RR} . This can be rewritten in terms of a single cost using

$$C_{FR} = 2 - C_{FA} \quad (7)$$

giving

$$E = C_{FA} GFAR + (2 - C_{FA}) GFRR \quad (8)$$

The optimum Bayesian fusion rule that minimizes the total cost (8) is obtained by selecting the rule to combine single biometric sensor decisions into a combined decision. The single sensor observations and the corresponding decisions are assumed to be independent.

3. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization algorithm, originally introduced in terms of social and cognitive behavior by Kennedy and Eberhart in 1995 [17], has come to be widely used as a problem solving method in engineering and computer science. PSO has since proven to be a powerful competitor to evolutionary algorithms such as genetic algorithms [19]. The technique is fairly simple and comprehensible as it derives its simulation form social behavior of individuals. The individuals, called particles henceforth, are flown through the multidimensional search space, with each particle representing a possible solution to the multidimensional problem. The movement of the particles is influenced by two factors: as a result of the first factor, each particle stores in its memory the best position visited by it so far, called *pbest* and experiences a pull towards this position as it traverses through the search space. As a result of the second factor, the particle interacts with all the neighbors and stores in its memory the best position visited by any particle in the search space and experiences a pull towards this position, *gbest*. The first and the second factors are called cognitive and social components respectively. After each iteration the *pbest* and *gbest* are updated if a more dominating solution (in terms of fitness) is found, by the particle and by the population respectively. This process is continued iteratively until either the desired result is achieved or the computational power is exhausted.

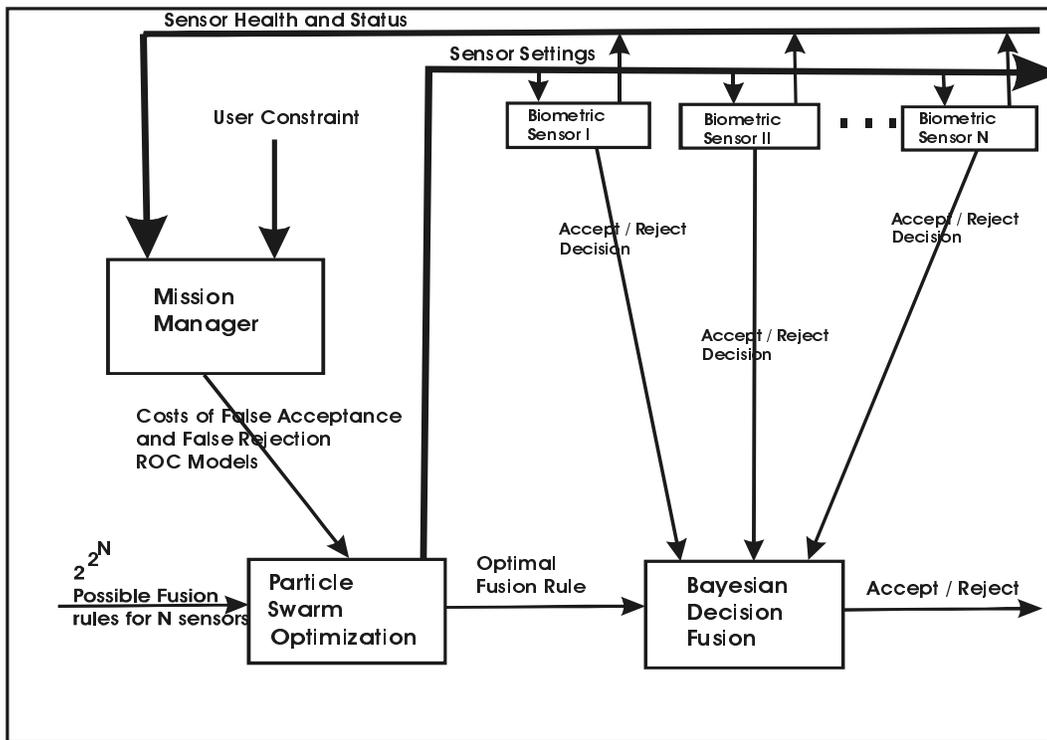


Fig. 1. Illustration of Adaptive Multimodal Biometric Fusion Algorithm.

The PSO formulae define each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$ where the subscript i represents the particle number and the second subscript is the dimension. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$ and a velocity along each dimension as $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$ [24]. After each iteration, the velocity term is updated and the particle is pulled in the direction of its own best position, P_i and the global best position, P_g , found so far. This is apparent in the velocity update equation, [17, 18, 24].

$$V_{id}^{(t+1)} = \omega \times V_{id}^{(t)} + rand(1) \times \Psi_1 \times (p_{id} - X_{id}^{(t)}) + rand(1) \times \Psi_2 \times (p_{gd} - X_{id}^{(t)}), \text{ and} \quad (9)$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)}. \quad (10)$$

Constants Ψ_1 and Ψ_2 determine the relative influence of the social and cognition components and often both of these are set to same value to give equal weight to both. The memory of the swarm is controlled by ω .

Figure 1 illustrates the algorithm with PSO incorporated. The costs of the system errors, computed in the Mission Manager, are dependent on many factors especially user constraints and hence are beyond the scope of this paper. Given the costs and the sensor suite, the problem of achieving global optima is difficult due to the existence of multiple peaks in the objective function. The system is dynamic allowing changes in the costs, and sensor performance is fed back to the system.

4. PROBLEM FORMULATION

Each particle in this problem has ' $N+1$ ' dimensions, where N is the number of sensors in the sensor suite. Each of the N dimensions is a threshold at which that particular sensor is set. The ' $N+1$ 'th dimension is the fusion rule, which determines how all the decisions from the sensors are fused. Hence the representation of each particle is

$$X_i = \{\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{in}, f_{in+1}\} \quad (11)$$

The sensor thresholds are continuous. The fusion rule, however, is a binary number having a length of $\log_2 p$ bits, where

$p = 2^N$, with a real value varying from $0 \leq f \leq p - 1$. For binary search spaces, the binary decision model as described in [26] is used. An alternative is to simply evolve a real number representation of the rule. This leads to an additional procedure bounding the resulting real values to lie within the search space. The bounding process results in the particles selecting the rule at the boundary too often. A binary decision model works better for moving through the decision fusion space.

In the algorithm instead of evolving the thresholds explicitly, the false acceptance rates (*FAR*) are evolved for each of the sensors. Thresholds are calculated from this and then the *FRRs* are calculated depending on the mean and standard deviation of the sensor noise determined a priori. The aim of the PSO is to then minimize the cost, (6), in the Bayesian decision fusion problem. The global *GFAR* and the *GFRR* for the fusion rule, F , can be calculated directly from the fusion rule and local *FAR*. Let f be a binary string that represents the fusion rule of length $\log_2 p$. For two sensors the fusion rule consists of 4 bits as represented in the Table I.

TABLE I. FUSION RULE FORMATION FOR 2 SENSORS

u_1	u_2	f
0	0	f_0
0	1	f_1
1	0	f_2
1	1	f_3

In Table I, u_1 is the first sensor decision; u_2 is the second sensor decision. These local decisions are related to the global decision represented by 0s and 1s in place of the f string. This representation of the fusion rule can be used to compute the global error rates using:

$$GFAR = \sum_{i=0}^{p-1} f_i \times \left(\prod_{j=1}^N E_j \right) \quad (12)$$

where $E_j = 1 - FAR_j$ if $u_j = 0$; $f_i \in [0, 1]$, and $E_j = FAR_j$ if $u_j = 1$.

Similarly,

$$GFRR = \sum_{i=0}^{p-1} (1 - f_i) \times \left(\prod_{j=1}^N E_j \right) \quad (13)$$

where $E_j = 1 - FRR_j$ if $u_j = 1$; and $E_j = FRR_j$ if $u_j = 0$.

The cost function, which is optimized by the PSO, is

$$E_a = C_{FA} \times (GFAR_a - GFAR_d) + (2 - C_{FA}) \times (GFRR_a - GFRR_d) \quad (14)$$

where E_a is the total cost used as the PSO objective function. $GFAR_a$ and $GFRR_a$ are the global false acceptance and rejection rates. $GFAR_d$ and $GFRR_d$ are the desired global false acceptance and rejection rates. Thus the algorithm adapts to the required error rates preventing overdesign of the rule.

5. RESULTS AND ANALYSIS

This section demonstrates the effectiveness of the swarm in selecting the proper rule and operating points dynamically to meet the general security requirements of the system. The convergence speed of the swarm allows it to dynamically respond to system changes. The optimality of the resulting rule and operating points are demonstrated in this section. Two experiments are conducted with varying false acceptance costs: 1.8 and 1.9. The same a priori sensor models given in Table 3 are used in both runs. Gaussian distributions have been assumed for both sensors and both an imposter and genuine user case. These distributions approximate the biometric sensor performance determined empirically in [10]. The distribution parameters are used by the swarm to compute threshold and the false rejection rate for a particular false acceptance rate. The small cost change results in a significant change in the rule and operating points. This demonstrates that the optimal solutions change as a result of seemingly insignificant security changes. The swarm parameters used for the run are shown in Table 2.

Table 4 shows the results obtained by running the swarm with $C_{FA} = 1.9$ and $C_{FA} = 1.8$, respectively. The two cases with only a cost difference of 0.1 result in the swarm switching from an AND rule to an OR rule with very different sets of operating points. For the higher cost, the swarm converged to a final solution after 550 iterations. The slightly lower cost resulted in a solution only after 3500 iterations. It is important to prevent premature convergence by properly selecting ω or the swarm memory. Premature convergence results in a suboptimal solution, which is the risk of using any evolutionary algorithm approach. The dominance of sensor 2 over sensor 1 in terms of accuracy is also the cause of this slow convergence.

TABLE 2. Table Showing the Particle Swarm Optimization Parameters

Ψ_1	Ψ_1	ω	No. of particles
1	1	0.8	10

TABLE 3. Table Showing the Means and Standard Deviations of the Two Sensors

Parameter	Sensor 1	Sensor 2
Mean, μ_1 , for imposter	12	12
Standard Deviation, σ_1 , for imposter	8	8
Mean, μ_2 , for genuine	36	40
Standard Deviation, σ_2 , for genuine	12	4

The swarm converges to optimal solutions, which may seem illogical at first as illustrated by the $C_{FA} = 1.8$ case. The false acceptance costs are high in both the 1.8 and 1.9 case. The swarm, however, switches to an OR rule for the $C_{FA} = 1.8$ case. This does not make sense until one analyzes the thresholds and sensor operating points selected. Figure 2 and Figure 3 illustrate the sensor distributions and threshold selected by the swarm after 10000 iterations. The threshold for sensor 1 is very high resulting in an extremely low false acceptance rate. This supports the final selection of the OR rule. Sensor 1 will reject genuine users 64% of the time. Thus, the sensor is sometimes ignored, and the more accurate sensor, sensor 2, is relied upon. This illustrates the need to incorporate some other performance factor into the problem to prevent this condition. The total cost, however, is .0138 and meets the criteria given to the swarm as given in Table 4 and plotted as a function of swarm iteration in Figure 4. It is interesting to note that if the AND rule is applied to this set of sensor operating points then the total cost increases dramatically to 0.1270.

For the second case with $C_{FA} = 1.9$, the AND rule is selected by the swarm as given in Table 4 after 1000 iterations. The operating point for sensor 2 is nearly identical to the previous case and illustrated in Figure 6. The threshold for sensor 1, however, is reduced significantly as illustrated in Figure 5. Sensor 1 now has an extremely high rate of false acceptance but a more tolerable false rejection rate. The accuracy of sensor 2 is used to offset the high false acceptance rate of sensor 1. The total cost is nearly the same as in the previous case or .0102. If the wrong rule such as the OR rule is applied, the total cost increases to 0.9116.

TABLE 4. Table Containing the Resulting Operating Points and Error Rates After 10000 Iterations

Parameter	$C_{FA}=1.9$	$C_{FA}=1.8$
Fusion Rule	AND rule	OR rule
Minimum cost achieved	0.0102	0.0138
F_{AR} /Sensor 1	0.4772	2.5301e-004
F_{RR} /Sensor 1	0.0249	0.6249
F_{AR} /Sensor 2	0.0049	0.0055
F_{RR} /Sensor 2	0.0333	0.0281

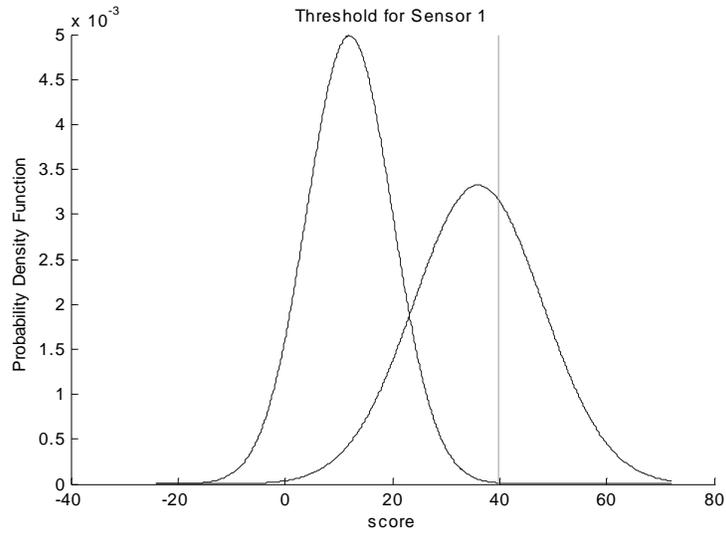


Fig. 2. Threshold and Distributions for Sensor 1 with a $C_{FA}=1.8$ After 10000 Iterations

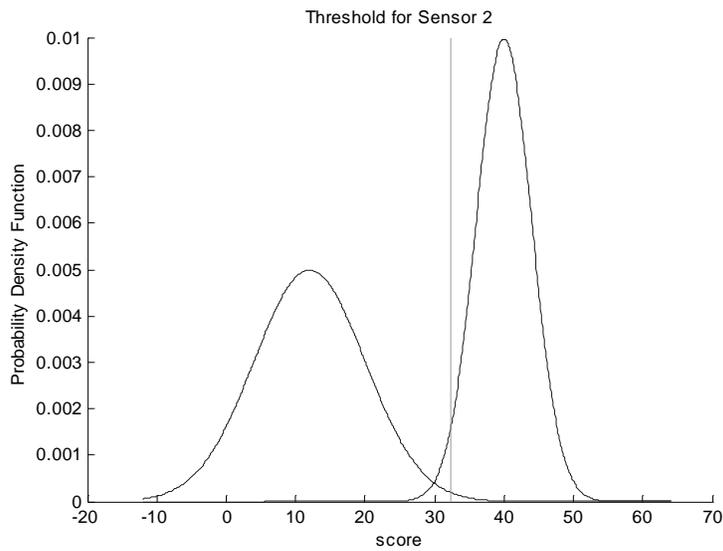


Fig. 3. Threshold and Distributions for Sensor 2 with $C_{FA}=1.8$ After 10000 Iterations

Figure 4 represents the minimum of total cost achieved by the swarm after each iteration. This plot illustrates the step characteristic of the final cost, which results from the 'minimum cost' only being replaced when a better minimum is found. The swarm particles, however, are moving through the search space testing different parameter sets during this search. Convergence occurs when all particles produce the same minimum total cost. It is possible for the particles to present more than one

optimal solution. If multiple solutions yield the same total cost, the solutions are viewed as being on the pareto surface. Any one of the resulting solutions can be selected as a final solution. In such a case, other constraints such as transaction time and ease of use may be introduced to select the best solution.

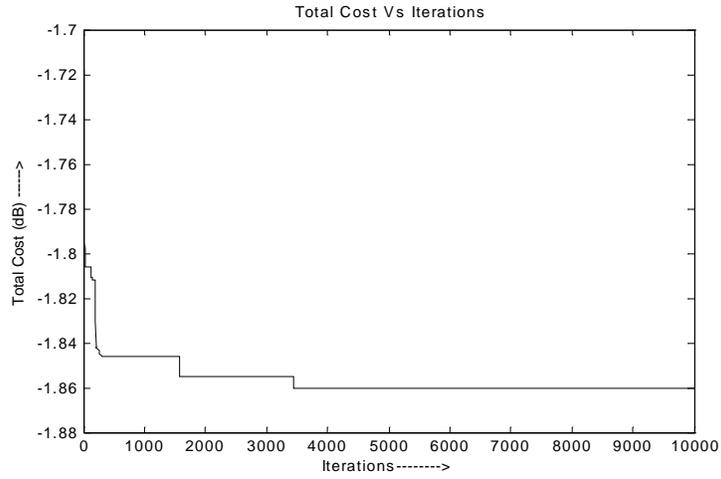


Fig. 4. Minimum cost vs. Number of Iterations for $C_{FA}=1.8$ After 10000 Iterations

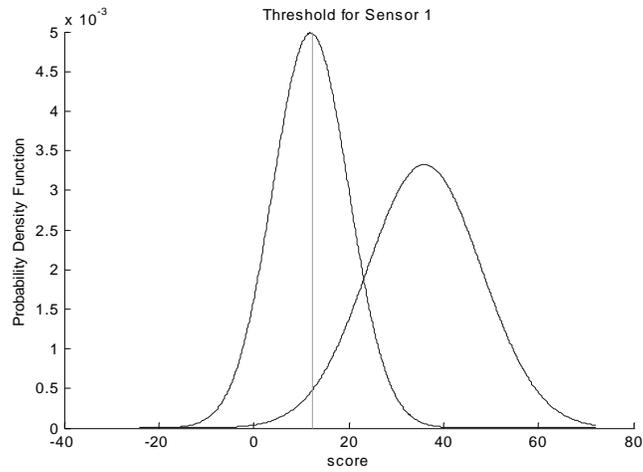


Fig. 5. Threshold and Distributions for Sensor 1 for $C_{FA}=1.9$ After 1000 Iterations

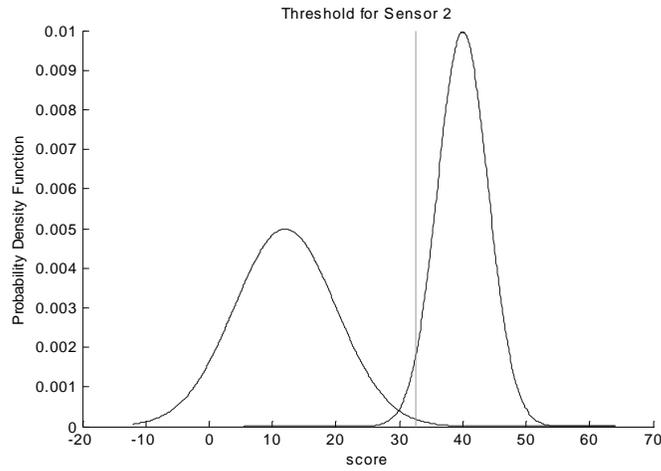


Fig. 6. Threshold and Distributions for Sensor 2 with $C_{FA}=1.9$ After 1000 Iterations

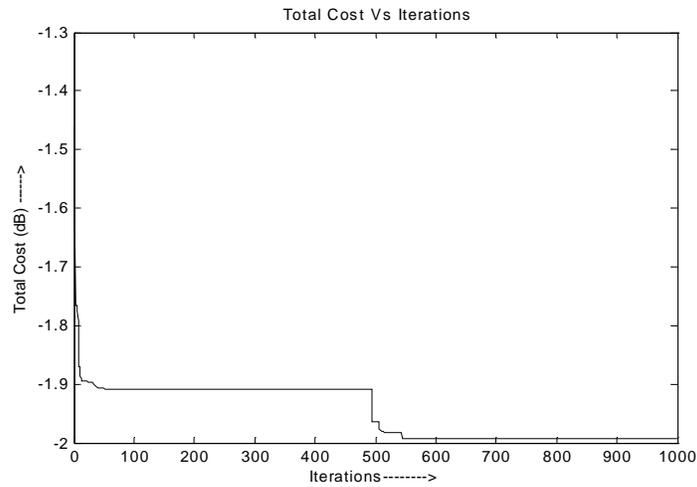


Fig. 7. Minimum cost vs. Number of Iterations for $C_{FA}=1.9$

6. CONCLUSIONS

The paper has demonstrated that by combining particle swarm with Bayesian decision fusion results an adaptive and dynamic fusion design emerges. The particle swarm is able to search through the entire search space defined by the fusion rules and sensor operating points. Some of the solutions that meet the system performance criteria may not be intuitive solutions. This approach provides a more comprehensive way to consider all the fusion rule and sensor operating point sets. In some cases, a better solution that was not previously considered may emerge. Similarly, the swarm can easily handle the scalability issue as the number of sensors increases and efficiently search through the highly large fusion rule search space.

Future work will focus on increasing the robustness of the algorithm design in terms of sensor models. Other sensor models will be investigated. Also new constraints such as transaction time for the user and ease of use will be incorporated to improve the quality of the fusion rules selected by the swarm. Improvements to this evolutionary algorithm as well as other contemporary evolutionary algorithms will be considered for this multimodal or multiple peak search space.

REFERENCES

- 1 John D. Woodward, Jr., "Biometrics: Facing Up to Terrorism", The Biometrics Consortium Conference 2002, Arlington Virginia, February, 2000.
- 2 Steven King, "Personal Identification Pilot Study," The Biometrics Consortium Conference 2002, Arlington Virginia, February, 2002.
- 3 Lin Hong and Anil Jain, "Integrating Faces and Fingerprints for Personal Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12, Dec., 1998, pp. 1295 - 1307.
- 4 Salil Prabhakar and Anil Jain, "Decision-level Fusion in Fingerprint Verification", Pattern Recognition, vol. 35, 2002, pp. 861-874.
- 5 Sharath Panikanti, Ruud M. Bolle, Anil Jain, "Biometrics : The Future of Identification", IEEE Computer, Vol. 33, No. 2, February 2000.
- 6 A.K. Jain, S. Prabhakar, S. Chen, "Combining multiple matchers for a high security fingerprint verification system", Pattern Recognition Letters 20 (11-13) (1999) 1371-1379.
- 7 A.K. Jain, R.M. Bolle, S. Pankanti (Eds.), "Biometrics : Personal Identification in a Network Society", Kluwer Academic Publishers, MA, 1999.
- 8 Forrester Research, Inc, <http://www.forrester.com>, 2001.
- 9 Gartner Group, <http://www.gartner.com>, 2001.
- 10 Tony Mansfield, Gavin Kelly, David Chandler, and Jan Kane, Biometric Product Testing Final Report, Computing, National Physical Laboratory, Crown Copyright, UK, March, 2001.
- 11 Robert W. Frischholz, Ulrich Deickmann, "BioID: A Multimodal Biometric Identification System" , IEEE Computer, Vol. 33, No. 2, February 2000.
- 12 L.Hong, A.K. Jain, S. Panikanti, "Can multibiometrics improve performance?,"Proceedings AutoID'99, Summit, NJ, October 1999, pp. 59-64.
- 13 L.Osadcw, P.K.Varshney, and K. Veeramachaneni, " Improving Personal Identification Accuracy Using Multisensor Fusion for Building Access Control Applications", Proceedings of the Fifth International Conference on Information Fusion, July 2002, Annapolis, Maryland.
- 14 Steven M. Kay, Fundamentals of Statistical Signal Processing: Detection Theory, Vol. II, Prentice-Hall, Inc., 1998.
- 15 Ramanarayanan Viswanathan and Pramod K. Varshney, "Distributed Detection With Multiple Sensors: Part I - Fundamentals", Proceedings of the IEEE, Vol. 85, No. 1, Jan., 1997, pp. 54 - 63.
- 16 Pramod K. Varshney , Distributed Detection and Data Fusion, Springer, New York, 1997.
- 17 Kennedy J. and Eberhart, R., "Particle Swarm Optimization", IEEE International Conference on Neural Networks, 1995, Perth, Australia.
- 18 Eberhart R. and Kennedy J., "A New Optimizer Using Particles Swarm Theory", Sixth International Symposium on Micro Machine and Human Science, 1995, Nayoga, Japan.
- 19 Eberhart R. and Shi Y., "Comparison between Genetic Algorithms and Particle Swarm Optimization", The 7th Annual Conference on Evolutionary Programming, 1998, San Diego, USA.
- 20 Shi Y. H., Eberhart R. C., "A Modified Particle Swarm Optimizer", IEEE International Conference on Evolutionary Computation, 1998, Anchorage, Alaska.

- 21 Kennedy J., "Small Worlds and MegaMinds: Effects of Neighbourhood Topology on Particle Swarm Performance", Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1931-1938. IEEE Press.
- 22 Lovbjerg M, Rasmussen, T. K. Krink, T., "Hybrid Particle Swarm Optimiser with Breeding and Subpopulations", Proceedings of Third Genetic Evolutionary Computation, (GECCO 2001).
- 23 Shi Y. H., Eberhart R. C., "Parameter Selection in Particle Swarm Optimization", The 7th Annual Conference on Evolutionary Programming, San Diego, USA.
- 24 Carlisle A. and Dozier G. "Adapting Particle Swarm Optimization to Dynamic Environments", Proceedings of International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, pp. 429-434, 2000.
- 25 Carlisle A., Dozier G. "An Off- The- Shelf PSO", PSO Workshop, IUPUI, Indianapolis, IN, April, 2001.
- 26 Kennedy J., Eberhart R. C., and Shi, Y. H., Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
- 27 P. J. Angeline, "Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences", Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 601-610. Springer.
- 28 X. Hu, Eberhart R.C., "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems", Proceedings of Congress on Evolutionary Computation, 2002. pp. 1666-1670. Hawaii, USA.