Fingerprint and Speaker Verification Decisions Fusion Using a Functional Link Network

Kar-Ann Toh, *Senior Member, IEEE,* and Wei-Yun Yau, *Senior Member, IEEE*

***Abstract*—By exploiting the specialist capabilities of each classifier, a combined classifier may yield results which would not be possible with a single classifier. In this paper, we propose to combine the fingerprint and speaker verification decisions using a functional link network. This is to circumvent the nontrivial trial-and-error and iterative training effort as seen in backpropagation neural networks which cannot guarantee global optimal solutions. In many data fusion applications, as individual classifiers to be combined would have attained a certain level of classification accuracy, the proposed functional link network can be used to combine these classifiers by taking their outputs as the inputs to the network. The proposed network is first applied to a pattern recognition problem to illustrate its approximation capability. The network is then used to combine the fingerprint and speaker verification decisions with much improved receiver operating characteristics performance as compared to several decision fusion methods from the literature.**

***Index Terms*—Data fusion, functional link network and multivariate polynomials, pattern recognition, supervised learning.**

I. Introduction

FUSION of several biometrics to improve the verification performance has received considerable attention over recent years owing to an increasing demand for reliable automatic user identity verification systems (e.g., [1]–[4]). Due to possible increase in degree of freedom, fusion of different modalities may allow alleviation of problems intrinsic to individual modalities. The importance of having an effective fusion methodology thus cannot be overemphasized.

The biometric verification problem can be considered as a classification problem wherein a decision is made upon whether or not a claimed identity is genuine with inference to some matching criteria. Fusion of biometrics can thus be treated as a classifier decisions combination [5] problem, where two main types of combination can be identified: *classifier selection* and *classifier fusion*. The difference between these two types lies in whether the classifiers are assumed to be complementary or competitive. Classifier selection assumes that each classifier is a "local expert," while classifier fusion assumes that all classifiers are trained over the entire feature space (e.g., [6]). In this paper, our focus will be on classifier fusion using fingerprint and voice-based biometrics. Our main effort will be to arrive at a fusion technique that optimizes the accuracy of the combined decision.

In general, different classifiers can be combined at one of the following levels [1], [3] according to the information adopted: i) abstract level, ii) rank level, and iii) measurement level. At the abstract level, the output information taken from each classifier is only a possible label for each pattern class, whereas at the rank level, the output information taken from each classifier is a set of ordered possible labels, which is ranked by a decreasing confidence measure. At the measurement level, the output information taken from each classifier is a set of possible labels with an associated confidence measure. We shall work at the measurement level to combine the fingerprint and speaker verification systems. In this way, with the measurement outputs taken from each individual system, the decision is brought forward to the final output of the combined system.

A natural framework to formulate solutions for decisions fusion is a statistical one, wherein the probabilistic nature of both the information to be processed, and the form of results to be expressed are recognized [5]. A direct approach to statistical decisions fusion is to evaluate the prior probability and the class-conditional probability separately and then combine them using the Bayes theorem to generate the posterior probability. The Bayes theorem allows the posterior probability to be expressed in terms of the prior probability and the class-conditional probability. An alternative approach is to estimate the posterior probability functions directly. In general, the outputs of an estimator or a network-like model can be interpreted as the posterior probability with appropriate choice of error formulation [7]. We shall focus on this approach in this paper.

The Feedforward Neural Network (FNN) has been shown to be a universal approximator (e.g., [8]). However, the training process remains much of a trial-and-error effort since no learning algorithm can guarantee convergence to a global optimal solution within finite iterations for general application problems. Backpropagation of error gradients has proven to be useful in FNN learning, but a large number of iterations is usually needed for adapting the weights. The problem becomes more severe especially when a high level of accuracy is required. Global Feedforward Neural Network learning (GFNN) [9], [10] may provide possible speedups in the training process, but it is limited to applications that do not result in numerical ill-conditioning.

The Radial Basis Function Network (RBFN) (e.g., [11]) has been widely used for approximation due to its structural simplicity. Typically, training of the RBFN involves selection of the hidden-layer neuron centers, choice of the scaling parameters, and estimation of the weights that connect the hidden and the output layers. Although the weights can be estimated using the linear least squares algorithm once the centers and the scaling parameters are fixed, selection of these centers and scaling parameters remains a nontrivial task. Other networks like Ridge Polynomial Networks (RPN) (e.g., [12] and [13]), though they

Manuscript received February 28, 2003; revised November 18, 2003. This paper was recommended by Guest Editor D. Zhang.

The authors are with the Institute for Infocomm Research, Singapore 119613 (e-mail: katoh@ieee.org; wyyau@i2r.a-star. edu.sg).

Digital Object Identifier 10.1109/TSMCC.2005.848184

may come with good approximation capability, have similar problems in training since the formulation is usually nonlinear. In the more general Functional Link Network (FLN) and High-Order Perceptrons (HOP) using polynomial and power series expansions, the problem of having a huge number of parameters persists unless a computational intensive evolutional search is performed to reduce the model to an optimal subset of units [14].

Apart from those universal approximators mentioned above, the Optimal Weighting Method (OWM) [15] provides an efficient way to combine different estimators and classifiers. However, it is limited to systems which can be separated by linear separation hyper-planes. As an extension to the OWM, the Multivariate Polynomial model (MP) provides an effective way to describe complex nonlinear input–output relationships since it is tractable for optimization, sensitivity analysis, and prediction of confidence intervals. However, for high-dimensional and high order systems, multivariate polynomial regression becomes impractical due to its prohibitive number of product terms. This is especially true for the case of using a full interaction model. In view of these, our problem here is to derive a network model that does not possess an exponentially increasing number of parameters with respect to the model order and number of inputs and at the same time preserves much of its approximation capability. We address this problem by proposing a Generalized Reduced Multivariate polynomial network (GRM), where the number of parameters to be estimated increases *almost linearly* with the model orders and the number of inputs. This GRM belongs to a class of the more general functional link networks mentioned above. To circumvent possible multicollinearity among the classifiers or estimators and improve generalization, a weight decay regularization is incorporated.

The paper is organized as follows. In the following section, the problem of decisions fusion is stated before some preliminaries on the optimal weighting method are provided. With these backgrounds in place, the multivariate polynomial model is introduced in Section III. An existing multinomial model is discussed before a reduced model is derived in the same section. With a generalized reduced network model in place, in Section IV, we present a validation approach to perform the network structure search. In Section V, a pattern recognition example will be used to illustrate the performance of the proposed network model in terms of the approximation capability. In Section VI, the proposed network is tested using physical data from the fingerprint and voice verification systems. The network is compared with several decision fusion methods from the literature. Finally, in Section VII, some concluding remarks are drawn.

II. Problem Definition and Preliminaries

Assume that each false positive poses the same amount of risk, every false negative presents identical liability, and that the system is under random attack. It remains an issue when combining a set of learned classifiers or estimators with correlation. The higher the degree of correlation, the larger the amount of agreement or linear dependence among the classifiers or estimators will be. This correlation also reflects the amount of redundancy within the set of classifiers. Here, the problem of correlation which can produce unreliable estimates is referred to as a *multicollinearity problem* [16].

With the above preliminaries, we will define our problem of biometric decisions fusion.

A. Problem of Multimodal Biometric Decisions Fusion

Given (l, m, n, p) as positive integers, consider two sets of data: a training set $\mathcal{S}_{\text{train}} = \{\boldsymbol{x}_i \in \mathcal{R}^p, y_i \in \mathcal{R}\}, i = 1, \ldots, m$ and a test set $\mathcal{S}_{\text{test}} = \{\boldsymbol{x}_i \in \mathcal{R}^p, y_i \in \mathcal{R}\}, i = 1, \ldots, n$. Given a set of functions $\mathcal{F} = \{\hat{f}_j(\boldsymbol{x}, y)\}, j = 1 \ldots, l$, where each of its elements $\hat{f}_j(\boldsymbol{x}, y)$ approximates a true function $f(\boldsymbol{x}, y)$ (assuming it exists), the data $\{\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{test}}\}$ is classified as genuine-users or imposters. Given some $\mathcal{F}$ based on $\mathcal{S}_{\text{train}}$, the problem of *biometric decisions fusion* is to find the best possible approximation of $f(\boldsymbol{x}, y)$ using this set of $\mathcal{F}$. The set $\mathcal{S}_{\text{test}}$ which has not been used in training will be used to test the classification performance.

According to this definition, we can treat the problem of biometric decisions fusion as a *pattern recognition* problem in the second level by taking samples of $\mathcal{F}$ as the inputs and $f(\boldsymbol{x}, y)$ as the output as in $\mathcal{S} = \{\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{test}}\}$ and find the best description of y using $\mathcal{S}_{\text{train}}$ in the *classification* sense where a *decision* is made according to the *inference* y. As decision data fusion problems do not usually contain a huge number of fusion dimension, we shall limit our investigation to overdetermined systems with just a few inputs (say, five inputs or less).

B. Optimal Weighting Method

A practical approach to data fusion is by the optimal weighting method [15] of minimizing the sum of squared errors given by the training samples. Given

$$\hat{f}_{\text{OWM}}(\boldsymbol{\alpha}) = \sum_{j=1}^{l} \alpha_j \hat{f}_j(\boldsymbol{x}, y) \tag{1}$$

and assuming no constraints, then the weights $(\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]^T)$ for OWM can be found from

$$\boldsymbol{\alpha} = (\hat{\mathbf{F}}^T \hat{\mathbf{F}})^{-1} \hat{\mathbf{F}}^T \mathbf{f} \tag{2}$$

where $\hat{\mathbf{F}} \in \mathcal{R}^{m \times l}$ denotes the Jacobian matrix of $\hat{f}_{\text{OWM}}$, i.e.,

$$\hat{\mathbf{F}} = \begin{bmatrix} \hat{f}_1(\boldsymbol{x}_1, y_1) & \ldots & \hat{f}_l(\boldsymbol{x}_1, y_1) \\ \vdots & \ldots & \vdots \\ \hat{f}_1(\boldsymbol{x}_m, y_m) & \ldots & \hat{f}_l(\boldsymbol{x}_m, y_m) \end{bmatrix} \tag{3}$$

and $\mathbf{f} = [f(\boldsymbol{x}_1, y_1), \ldots, f(\boldsymbol{x}_m, y_m)]$ is the training target vector.

Here, it is noted that OWM involves computation of the inverse of a matrix. The problem of multicollinearity may arise if linear dependency among the elements of $\mathcal{F}$ is present.

C. Weight–Decay Regularization

Minimization of the sum of squared errors using (2) may result in a multicollinearity problem when heavy linear dependency of data is present. A simple approach to provide numerical

stability is to perform a weight–decay regularization:

$$s(\boldsymbol{x}, y, \boldsymbol{\alpha}) = \sum_{i=1}^{m} e_i^2(\boldsymbol{\alpha}) + b\|\boldsymbol{\alpha}\|_2^2 = \sum_{i=1}^{m} \left[f(\boldsymbol{x}_i, y_i) - \hat{f}(\boldsymbol{x}_i, y_i, \boldsymbol{\alpha})\right]^2 + b\|\boldsymbol{\alpha}\|_2^2 \quad (4)$$

where $\|\cdot\|_2$ denotes the L_2-norm and b is a regularization constant.

Minimizing the new objective function (4) results in

$$\boldsymbol{\alpha} = (\hat{\mathbf{F}}^T\hat{\mathbf{F}} + b\mathbf{I})^{-1}\hat{\mathbf{F}}^T\mathbf{f} \quad (5)$$

where $\hat{\mathbf{F}} \in \mathcal{R}^{m\times l}, \mathbf{f} \in \mathcal{R}^{m\times 1}$, and $\mathbf{I}$ is a $(l \times l)$ identity matrix. It is noted that this addition of a bias term into the least squares regression model is also termed as *ridge regression* [17].

The addition of the bias term affects the total mean squared error of the estimator. When a large value of b is selected, a large value of the bias component will be included into the total mean squared error for training [17]. Depending on each application, the effect of b on the validation error varies from case to case.

III. Functional Link Network: Multivariate Polynomial Regression

The OWM described above provides an effective way to linearly combine estimators or classifiers. However, important interacting relationships among the data may be ignored, thereby giving rise to inaccurate results. To cater for possible nonlinear effects and interactions, multivariate polynomial regression is considered.

Multivariate polynomial regression provides an effective way to describe complex nonlinear input–output relationships. Also, it is tractable for optimization, sensitivity analysis, and prediction of confidence intervals. A typical polynomial regression model contains the squared and higher order terms of the estimator variable. However, for high-dimensional and high-order problems, multivariate polynomial regression becomes impractical due to its prohibitive number of product terms. This is especially true for the case of using an interaction model. For an rth-order model with input dimension l, the number of independent adjustable parameters would grow like l^r [7]. For medium to large sizes of data dimensions, the MP model would need a huge quantity of training data to ensure that the parameters are well determined (usually overdetermined).

In view of this problem, we resort to possible reduced models whose number of parameters does not increase exponentially and yet preserves the necessary classification and possibly the approximation capability.

In the following, to simplify the expression as well as to avoid possible confusion, the notation of individual classifiers or estimators $\hat{f}_j(\boldsymbol{x}, y), j = 1, \ldots, l$ to be combined will be replaced by $x_j j = 1, \ldots, l$ as polynomial inputs.

A. Multinomials: Special Case of Multivariate Polynomials

A special case of multivariate polynomials is called a multinomial (MN), which can be expressed as

$$(x_1 + x_2 + \cdots + x_l)^r = \sum \frac{r!}{n_1!n_2!\cdots n_l!} x_1^{n_1}x_2^{n_2}\cdots x_l^{n_l} \quad (6)$$

where the summation is taken over all non-negative integers $n_1, n_2, \ldots, n_l$ for which $n_1 + n_2 + \cdots + n_l = r$ with r being the order of approximation.

B. Reduced Multivariate Polynomial Network

To significantly reduce the huge number of terms in multivariate polynomials, we first consider the following multinomial model:

$$\hat{f}_{MN} = \alpha_0 + \sum_{j=1}^{r}(\alpha_{j1}x_1 + \alpha_{j2}x_2 + \cdots + \alpha_{jl}x_l)^j. \quad (7)$$

It is noted that this gives rise to a nonlinear estimation model where the weight parameters $(\alpha_{jk}), j = 1, \ldots, r, k = 1, \ldots, l$ may not be estimated in a straightforward manner. Although an iterative search can be formulated to obtain some solutions, there is no guarantee that these solutions are global. To circumvent this problem, a linearized model is considered.

Given two points $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_1$ on the multinomial function which is differentiable, by the Mean Value Theorem, the multinomial function $f(\boldsymbol{\alpha}) = (\alpha_{j1}x_1 + \alpha_{j2}x_2 + \cdots + \alpha_{jl}x_l)^j, j = 2, \ldots, r$ about the point $\boldsymbol{\alpha}_1$ can be written as

$$f(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}_1) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_1)^T\nabla f(\bar{\boldsymbol{\alpha}}) \quad (8)$$

where $\bar{\boldsymbol{\alpha}} = (1-\beta)\boldsymbol{\alpha}_1 + \beta\boldsymbol{\alpha}$ for $0 \le \beta \le 1$. Let $\boldsymbol{x} = [x_1, \ldots, x_l]^T$. With appropriate choice of terms based on (8), omitting the coefficients within $f(\boldsymbol{\alpha}_1)$ and $\nabla f(\bar{\boldsymbol{\alpha}})$, and including the summation of weighted input terms, the following multivariate model can be written as

$$\hat{f}_{\mathrm{RM}'} = \alpha_0 + \sum_{j=1}^{l}\alpha_j x_j + \sum_{j=1}^{r}\alpha_{l+j}(x_1 + x_2 + \cdots + x_l)^j + \sum_{j=2}^{r}\left(\boldsymbol{\alpha}_j^T \cdot \boldsymbol{x}\right)(x_1 + x_2 + \cdots + x_l)^{j-1}, \quad l, r \ge 2 \quad (9)$$

where the number of terms is given by $K = 1 + r(l+1)$.

To include more individual high-order terms for (9), the following (RM) can be written as

$$\hat{f}_{\mathrm{RM}} = \alpha_0 + \sum_{k=1}^{r}\sum_{j=1}^{l}\alpha_{kj}x_j^k + \sum_{j=1}^{r}\alpha_{rl+j}(x_1 + x_2 + \cdots + x_l)^j + \sum_{j=2}^{r}\left(\boldsymbol{\alpha}_j^T \cdot \boldsymbol{x}\right)(x_1 + x_2 + \cdots + x_l)^{j-1}, \quad l, r \ge 2. \quad (10)$$

The number of terms in this model can be expressed as $K = 1 + r + l(2r-1)$. It is noted that (10) has $(rl - l)$ more terms than (9). The plots for the number of terms over different model orders for each input dimension $(l = 2, 3, \ldots, 8)$ of RM are shown in Fig. 1(a). For comparison, the same figure

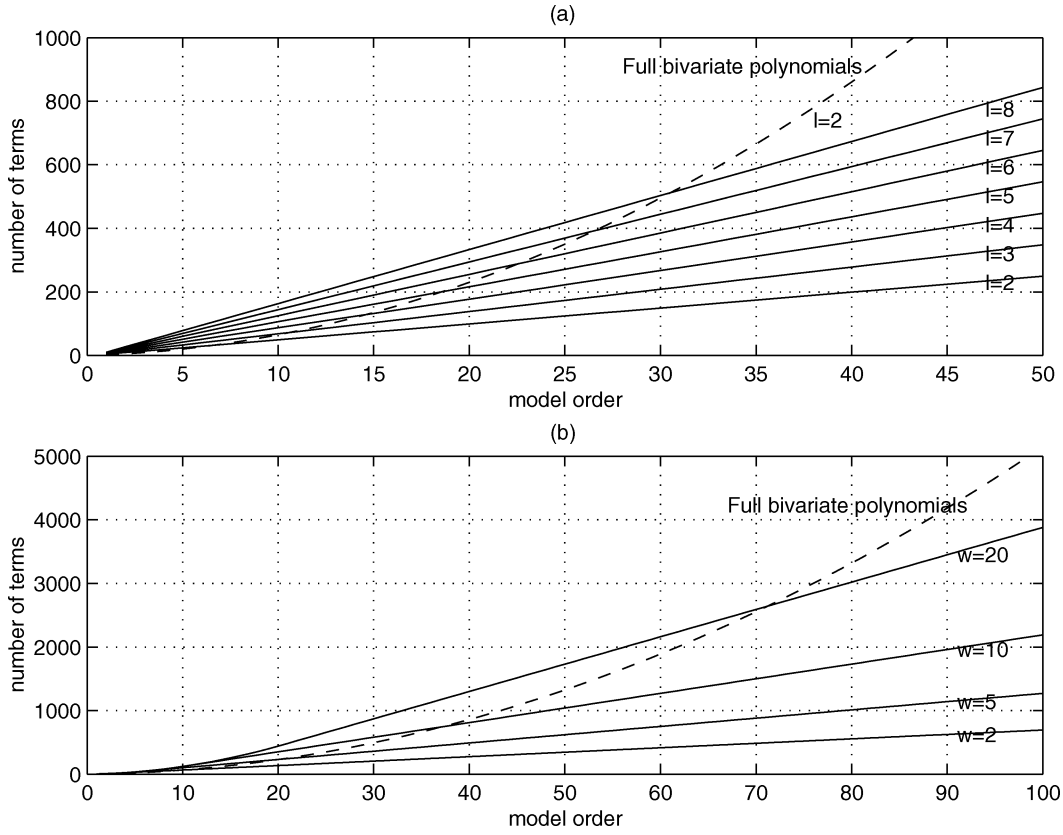

Fig. 1. Number of terms plotted over model order. (a) Number of parameters over model order for different input dimensions. (b) Number of parameters over model order for different w values at $l = 2$.

includes the number of terms plotted over the model orders for a full multivariate polynomial model with input dimension two $(l = 2)$.

C. Generalization of the Reduced Model

The model given by (10) can be extended and generalized to include high order terms as follows:

$$\begin{aligned}(\text{GRM}):\ \hat{f}_{\text{GRM}} &= \alpha_0 + \sum_{k=1}^{r}\sum_{j=1}^{l}\alpha_{kj}x_j^k \\ &\quad + \sum_{j=1}^{r}\alpha_{rl+j}(x_1+x_2+\cdots+x_l)^j \\ &\quad + \sum_{j=2}^{r}\left(\boldsymbol{\alpha}_j^T\cdot\boldsymbol{x}\right)(x_1+x_2+\cdots+x_l)^{j-1} \\ &\quad + \sum_{j=3}^{r}\left(\boldsymbol{\alpha}_j^T\cdot\boldsymbol{x}^2\right)(x_1+x_2+\cdots+x_l)^{j-2}+\cdots \\ &\quad + \sum_{j=w+1}^{r}\left(\boldsymbol{\alpha}_j^T\cdot\boldsymbol{x}^w\right)(x_1+x_2+\cdots+x_l)^{j-w} \\ &= \alpha_0 + \sum_{k=1}^{r}\sum_{j=1}^{l}\alpha_{kj}x_j^k \\ &\quad + \sum_{j=1}^{r}\alpha_{rl+j}(x_1+x_2+\cdots+x_l)^j \\ &\quad + \sum_{w=1}^{r-1}\sum_{j=w+1}^{r}\left(\boldsymbol{\alpha}_j^T\cdot\boldsymbol{x}^w\right) \\ &\quad \times (x_1+x_2+\cdots+x_l)^{j-w} \\ &= \alpha_0 + \sum_{j=1}^{r}\alpha_{j+1}(x_1+x_2+\cdots+x_l)^j \\ &\quad + \sum_{w=1}^{r}\sum_{j=w}^{r}\left(\boldsymbol{\alpha}_j^T\cdot\boldsymbol{x}^w\right) \\ &\quad \times (x_1+x_2+\cdots+x_l)^{j-w}, \quad l,r\geq 2 \end{aligned} \tag{11}$$

where $\boldsymbol{x}^w \triangleq [x_1^w, x_2^w, \cdots, x_l^w]$. The total number of terms in (11) can be expressed as $K = 1 + r(l+1) + \sum_{j=1}^{w}(r-j)l$ with $w \leq r$. Notice that (10) can be obtained from (11) by choosing the first two terms, plus two terms from the third term with $w = 1$, and that with $w = r$.

The total number of terms in (11) is plotted against the model order (r) with different w values in Fig. 1(b). Here, for the GRM, we see that the increase in the number of terms is more rapid at low order models, especially when w is large. The increase in the number of terms becomes linear when r is large.

IV. Network Structure Search

With the generalized model given by (11), our next task is to determine the parameters w and r, which are related to network size. As it is known that good training accuracy does not necessarily imply good test accuracy, we adopt the cross-validation approach to find the values of w and r which possess a good generalization property. In this learning framework, we partition the entire data set into three parts: subtraining data $\mathcal{S}_{\rm tr}$, validation data $\mathcal{S}_v$, and test data $\mathcal{S}_{\rm test}$. Here, we note that $\mathcal{S}_{\rm train} = \{\mathcal{S}_{\rm tr}, \mathcal{S}_v\}$. The subtraining and the validation data $\{\mathcal{S}_{\rm tr}, \mathcal{S}_v\}$ are used to obtain the best w and r for final training. The remaining data $\mathcal{S}_{\rm test}$, which are not used in both training and validation, will then be used to test the performance of the classifier using the final trained network.

Notice that the set $\mathcal{S}_{\rm train}$ can be partitioned into $\mathcal{S}_{\rm tr}$ and $\mathcal{S}_v$ in various ways, typically $N(\mathcal{S}_{\rm tr})/N(\mathcal{S}_{\rm train}) \geq 0.5$ with $N(\mathcal{S}_v) = N(\mathcal{S}_{\rm train}) - N(\mathcal{S}_{\rm tr})$, where $N(\bullet)$ denotes the number of elements within the set $\bullet$. In this application, we use $N(\mathcal{S}_{\rm tr})/N(\mathcal{S}_{\rm train}) = 0.9$ and perform a corresponding ten-fold cross-validation using the data sets $\{\mathcal{S}_{\rm tr}, \mathcal{S}_v\}_i, \mathrm{i} = 1, \ldots, 10$.

A search for w and r across various combinations can be performed within a certain integer range to locate the optimal ten-fold validation error for the given training set $\mathcal{S}_{\rm train}$. The values of w and r corresponding to the optimal ten-fold validation error are then selected for the final training using the set $\mathcal{S}_{\rm train}$.

In summary, the network is constructed as

```
»»for w = 1 : w_max do
»»  for r = 1 : r_max do
»»    Perform ten-fold cross-validation
      as follows:
»»    Partition S_train into S_tr and S_v
      with N(S_tr)/N(S_train) = 0.9.
»»    Using the current choice of w and
      r, compute the Mean Squared Errors
      (MSE) for the validation sets:
      {S_v}_i, i = 1,...,10.
»»    Store the total validation MSE for
      current choice of w and r.
»»  end for (r)
»»end for (w)
»»Locate values of w and r corresponding
  to minimum total validation MSE.
»»Retrain the network for the set S_train
  using the above values of w and r
  corresponding to minimum total
  validation MSE.
»»Compute the MSE for test set using the
  final trained network.
```

In the following case study, the entire process summarized above will be repeated 40 times using random partitions of $\{\mathcal{S}_{\rm train}, \mathcal{S}_{\rm test}\}$ with $N(\mathcal{S}_{\rm test})/N(\mathcal{S}_{\rm train}) = 1, 2,$ and 3. With this relatively large number of trials, we hope that a good statistical perception of the network capability can be achieved.

V. Case Study: Gabor Function Approximation

Here, we shall test the proposed model for its function approximation capability. As in [12], a two-dimensional (2-D) Gabor function is used for experimentation. It is well known that the Gabor function plays an important role in many areas of research including pattern recognition and image processing. The Gabor function to be approximated is written as

$$f(x_1, x_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right) \cos(2\pi(x_1 + x_2)) \tag{12}$$

where the scaling factor is chosen similarly to [12] as $\sigma = 0.42$.

A total of 256 points were selected from an evenly spaced 16×16 grid on $[-0.5, 0.5]^2$, and these data were randomly partitioned into two equal sets: one for training and one for testing. It is noted that in the experiment as seen in [12], the random partitioning was performed only once. In our experiment, this random partitioning was performed 40 times, and we hope that by doing this, a better representation of results is achieved.

For each of the 40 trials, the GRM network is allowed to be searched within the intervals $w \in [1, 5] \subset \mathcal{Z}$ and $r \in [1, 15] \subset \mathcal{Z}$. The regularization parameter was set at i) $b = 10^{-4}$ and ii) $b = 10^{-8}$ empirically as these values were found to be able to stabilize the system and did not contribute to serious numerical ill-conditioning. A ten-fold cross-validation was performed on each training set, and the best validated network corresponding to the (w, r) value was used for final training and test errors computation. On top of the experiment on equal partitioning, two additional cases using one third and one fourth of 256 points as the training sets are also included. These correspond to 85 and 64 data points being used for training. The remaining data points are then used for testing.

The resulting networks for the case of using one half of 256 points as the training set are shown in terms of the distribution of w and r in Table I. Here, we see that the most popular network is $(w, r) = (5, 10)$, where ten out of 40 trials had been tested using this network. It is noted that this network contains 101 parameters in the estimation.

The distribution and averaged results for all 40 trials in each case are presented in Fig. 2 and Table II, respectively. Fig. 2(a)–(c) correspond to results for training sizes one half, one third, and one fourth, respectively. For ease of comparison, the results of the RPN proposed by [12] are also summarized in Table II. It is seen from Table II that both the training and test accuracies of the proposed GRM outperformed that of the RPN remarkably, especially for case ii) with training sizes one half and one third, which are about 27 and six times more accurate than that of the RPN, respectively, in terms of test results. As for training size one fourth in case i), the average MSE is seen to be 0.0368 for test data, which appears to be slightly better than that of the RPN. However, one has to bear in mind that this value corresponds to the average of 40 random trials where some of the training sets (one fourth size) may not be representative. When the largest five MSEs are removed from the statistics, the average MSE for the remaining 35 trials is 0.014, and this value is much lower

TABLE I
GABOR FUNCTION: DISTRIBUTION OF NETWORK SIZES (w, r) FOR THE 40 TRIALS OF THE CASE USING ONE HALF SIZE OF DATA FOR TRAINING

w/r	1	...	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	4	0	0	1	0	0
3	0	0	0	0	0	3	0	3	0	1	0	0	0
4	0	0	0	0	0	3	0	2	0	0	0	0	1
5	0	0	0	0	0	4	0	10	2	3	1	0	0

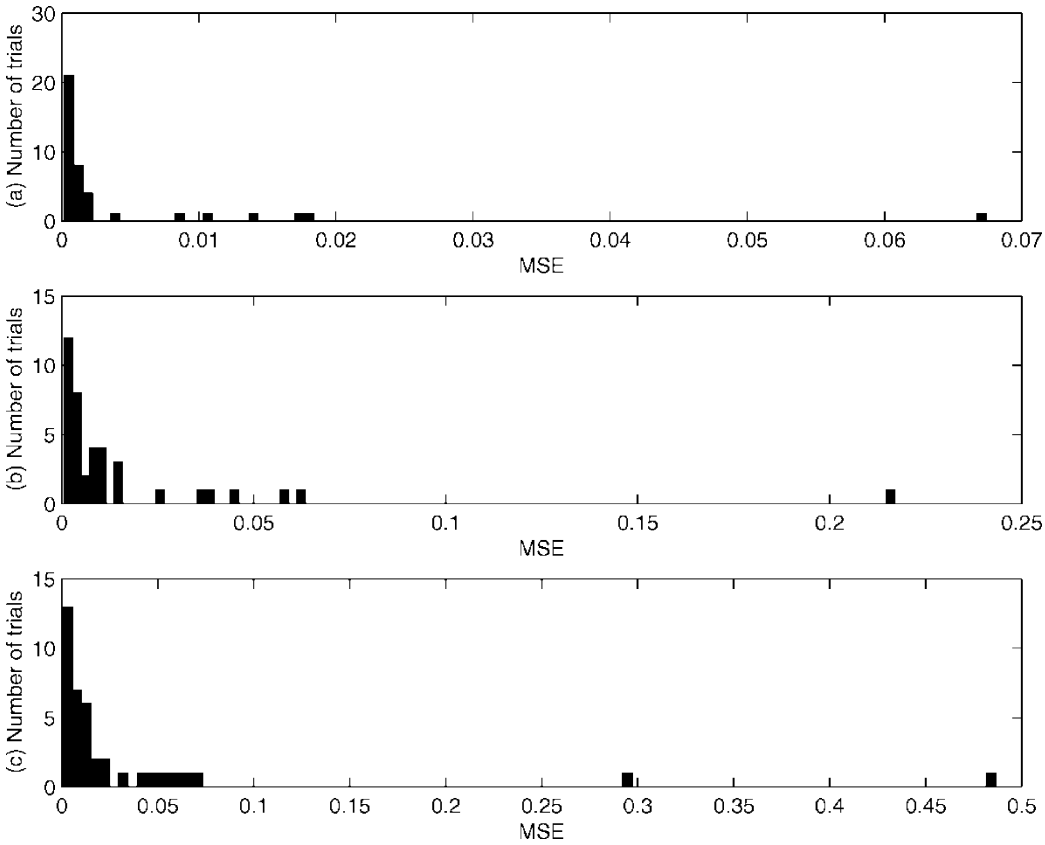

Fig. 2. Distribution of test MSEs for the 40 trials.

than that of the RPN. A count of these trials shows that 31 out of the 40 trials had achieved much better accuracies than that of the RPN. Besides the generalization capability, it is further noted that in all three cases, the superior training of MSEs implies a better approximation capability for the Gabor function for the proposed GRM.

VI. COMBINING FINGERPRINT AND SPEAKER BIOMETRICS

Biometric systems are automated methods for verifying or recognizing the identity of a person based on some physiological or behavioral characteristics that he possesses. In other words, biometric systems use "what you are" rather than "what you have" to recognize a person. As tokens such as keys, cards, passwords, PINs, etc. can be stolen, duplicated, left at home, shared, or forgotten, biometric systems provide a secure and possibly user-friendly means for user authentication. The importance of biometrics thus cannot be overemphasized.

In this section, we perform experiments on the reduced multivariate polynomial network using physical data from two biometrics: fingerprint and voice data. As the neural network and other nonlinear methods rely much on the choice of initial estimates and the iterative training process remains a trial-and-error procedure, we shall compare the performance of the proposed single-step GRM network with several classifiers in the form of Receiver Operating Characteristic (ROC) Curves.

A. Fingerprint Verification

In general, an automatic fingerprint identification or verification (e.g., [18]–[21]) system consists of three main processing

TABLE II
GABOR FUNCTION: AVERAGE ERRORS FROM 40 TRIALS

Nwk	Average results (MSE)		
	Size of the training set: 1/2	1/3	1/4
(a)	0.008	0.006	0.005
(b)	0.008	0.016	0.037
(c)	0.00030 (0.0002)	0.0012 (0.0166)	0.0006 (0.0004)
(d)	0.00420 (0.0112)	0.0045 (0.0360)	0.0368 (0.0874)
(e)	0.00000192 (0.00000215)	0.00001561 (0.0001)	0.0001 (0.0002)
(f)	0.00029350 (0.00051674)	0.0025 (0.0050)	0.0220 (0.0706)

Legends:

(a) RPN Training MSE
(b) RPN Test MSE
(c) Case (i) GRM Training MSE (std) ($b = 10^{-4}$)
(d) Case (i) GRM Test MSE (std) ($b = 10^{-4}$)
(e) Case (ii) GRM Training MSE (std) ($b = 10^{-8}$)
(f) Case (ii) GRM Test MSE (std) ($b = 10^{-8}$)

stages, namely, *image acquisition*, *feature extraction*, and *matching*. In image acquisition, query and template database images are acquired through various input devices. Development over the years has seen through means that mechanically scan the ink-based fingerprints into the computer system to means which directly capture the fingerprints using sophisticated solid-state sensors. With fingerprint images which could be distorted or contaminated with noise, the automated system seeks to *extract* characteristic *features* which are discriminating for different fingers and yet invariant with respect to image orientation for the same fingers. The final stage of fingerprint identification is to search and verify matching image pairs.

Our representation for the fingerprint consists of a global structure and a local structure. The global structure consists of positional and directional information of ridge endings and ridge bifurcations. The local structure consists of relative information of each detected minutia with other neighboring minutiae. Fingerprint verification is then performed by comparing the minutia information between two templates [22]. Fig. 3 shows some samples of the fingerprint images with detected minutiae and area of interest segmentation. Interested readers are referred to [22] and [23] for details of minutiae detection and matching.

B. Speaker Verification

Speaker verification seeks to determine whether an unknown voice matches the known voice of a speaker with known identity. It is a subset of the more general problem of speaker recognition which includes the task of speaker identification (e.g., [24]). Operation of the above systems can either be in fixed-text mode or in free-text mode. In fixed-text mode, a predetermined text is required to be recited for reliable comparison, whereas in free-text mode, speech utterances of unrestricted text can be accepted. The fixed-text system provides a more precise and reliable comparison between two utterances of the same text than that of the free-text system since it works under a better controlled environment. The fixed-text systems are thus primarily

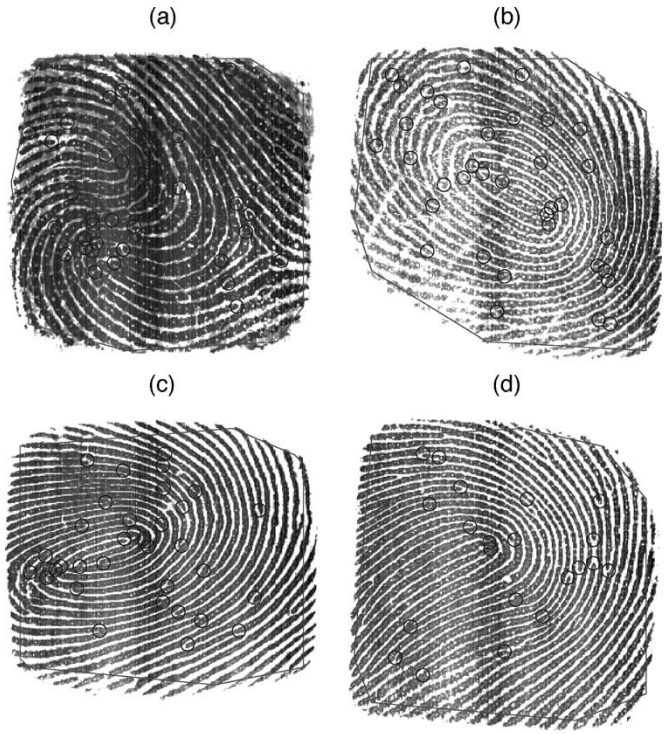

Fig. 3. Fingerprint image samples. (a) Thumb. (b) Index finger. (c) Middle finger. (d) Thumb.

used in access control applications and the free-text systems are more for surveillance and other applications [24].

In this application, the fixed-text mode and the template matching method is adopted for speaker verification. Comparison of two utterances is performed by aligning the two templates at corresponding points in time. To cater for difference in duration of the two utterances, the Dynamic Time Warping (DTW) method is adopted when minimizing a distance metric between two feature sets extracted from the speech data. Fig. 4 shows some samples of voice data uttering the word "zero." The interested reader is referred to [25] for more details about the system (see also [24] and [26] for similar matching designs).

C. Combining Fingerprint and Speaker Verification Systems

In this experiment, both the databases for fingerprint verification and speaker verification consist of 16 different identities; each comes with six different fingers or six different words with each fingerprint or word containing ten samples. A total of 960 samples was thus used for each fingerprint and voice verification system. The fingerprint images were collected using Veridicom's i Touch sensor and the voice data were taken from the TIDIGIT database. Arbitrary one-to-one correspondences were taken between the two biometric databases. Both databases are partitioned into two equal sets for training and testing, i.e., set-1 with 480 ($16 \times 6 \times 5$) samples for training and set-2 with 480 ($16 \times 6 \times 5$) samples for testing.

Depending on individual implementation, the matching output ranges for different modalities may differ significantly. For such cases, numerical sensitivity may be affected and hence a

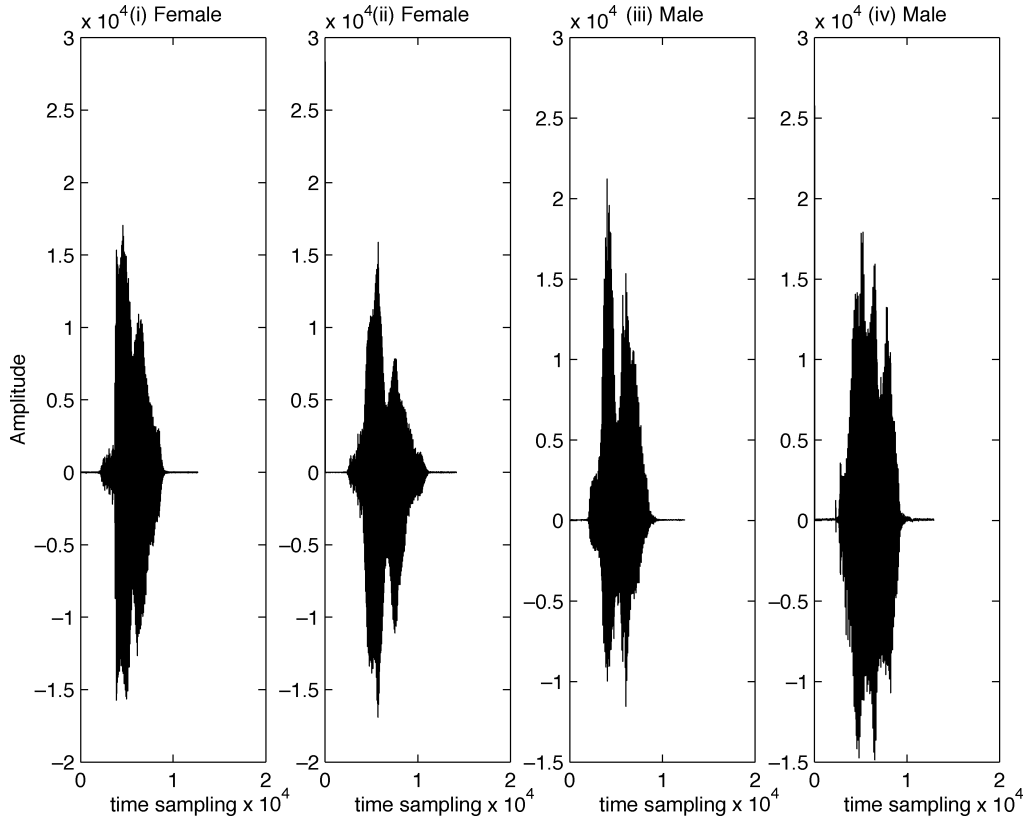

Fig. 4. Voice samples.

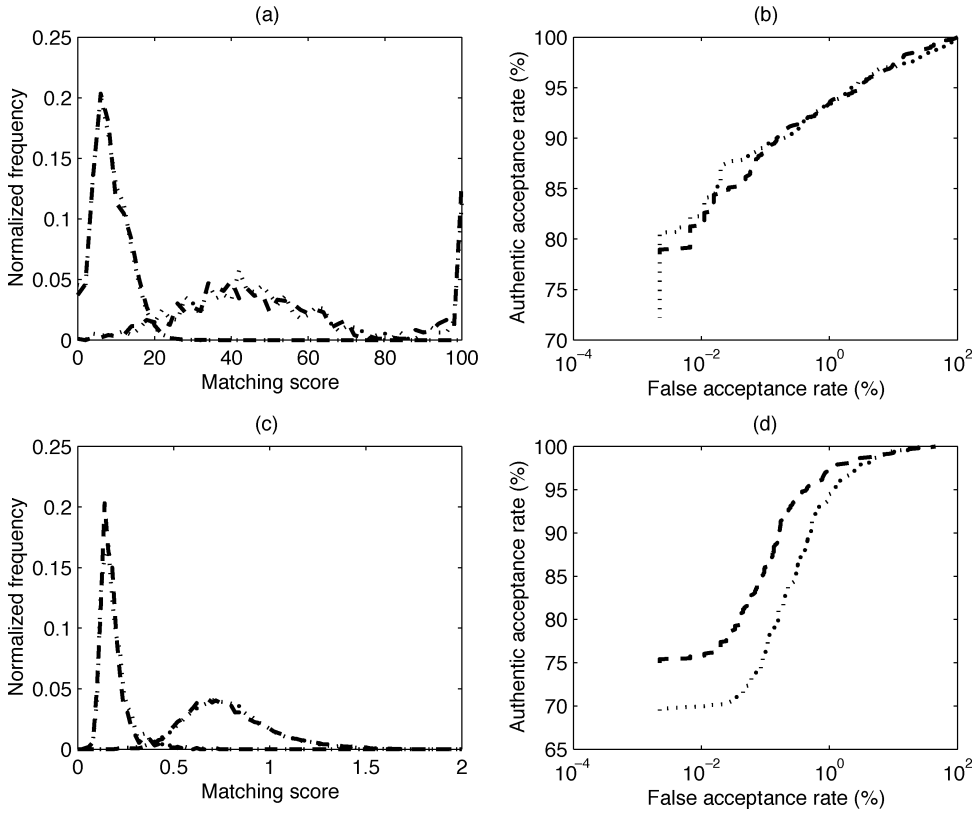

Fig. 5. Matching performance for fingerprint and speaker verification systems training (dashed lines) and test (dotted lines) sets. (a) Match score distribution (fingerprint). (b) Receiver operating curve (fingerprint). (c) Match score distribution (voice). (d) Receiver operating curve (voice).

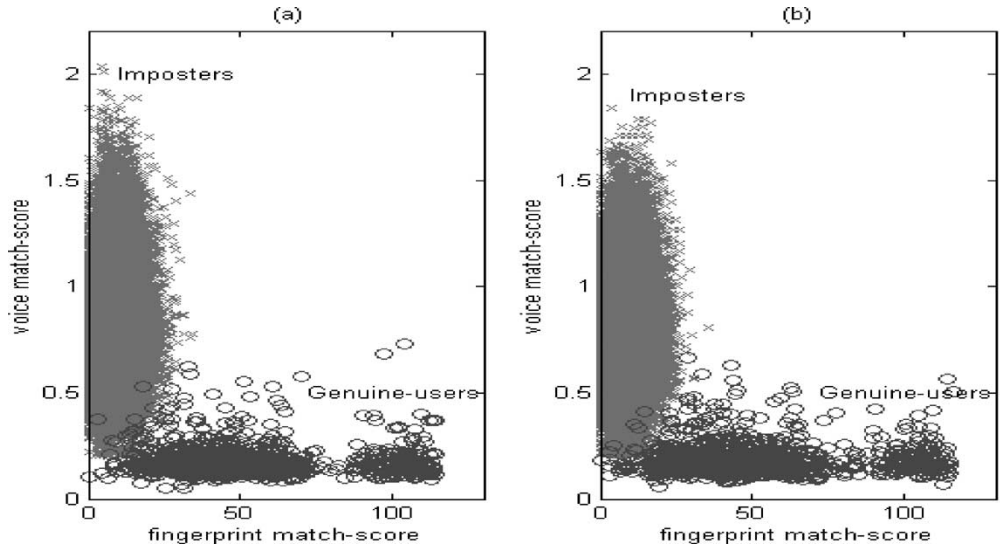

Fig. 6. Distribution of genuine and imposter scores for all users. (a) Distribution of scores (training data). (b) Distribution of scores (test data).

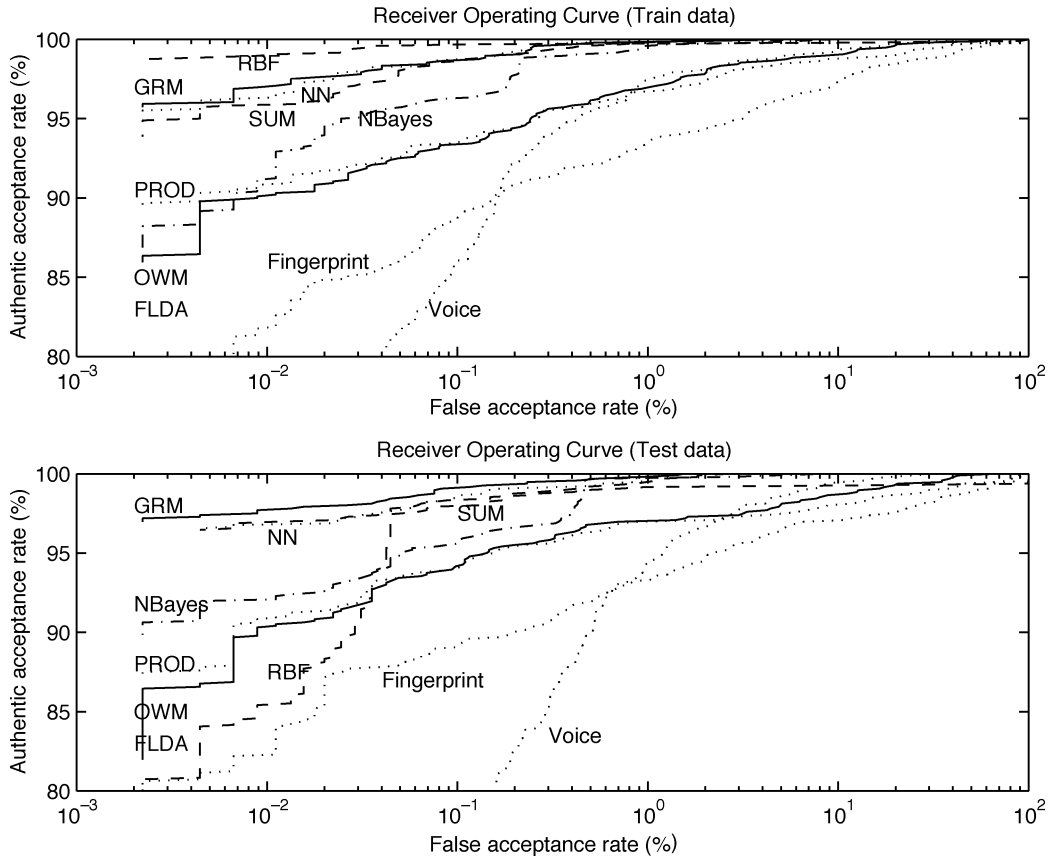

Fig. 7. ROC curve of GRM as compared to other fusion methods (GRM: continuous, NN: dotted, SUM: dashed, RBF: dashed, NBayes: dashed-dotted, PROD: dotted, FLDA: dotted, OWM: continuous).

score normalization should be performed between the outputs of different modalities. Otherwise, for reasonably small differences between the scores like in our case, the weighting parameters can be adapted automatically. Fig. 5(a)–(d) show the original matching performances for the training and test sets, respectively, for individual fingerprint verification and speaker verification, using the above-mentioned database, before multimodal fusion. The distributions of genuine (960 samples) and imposter (45 000 samples) scores for both biometrics in each training and test data set used are shown in Fig. 6. It can be seen from this figure that the separability of the two classes of scores has been improved in the 2-D decision plane as compared to

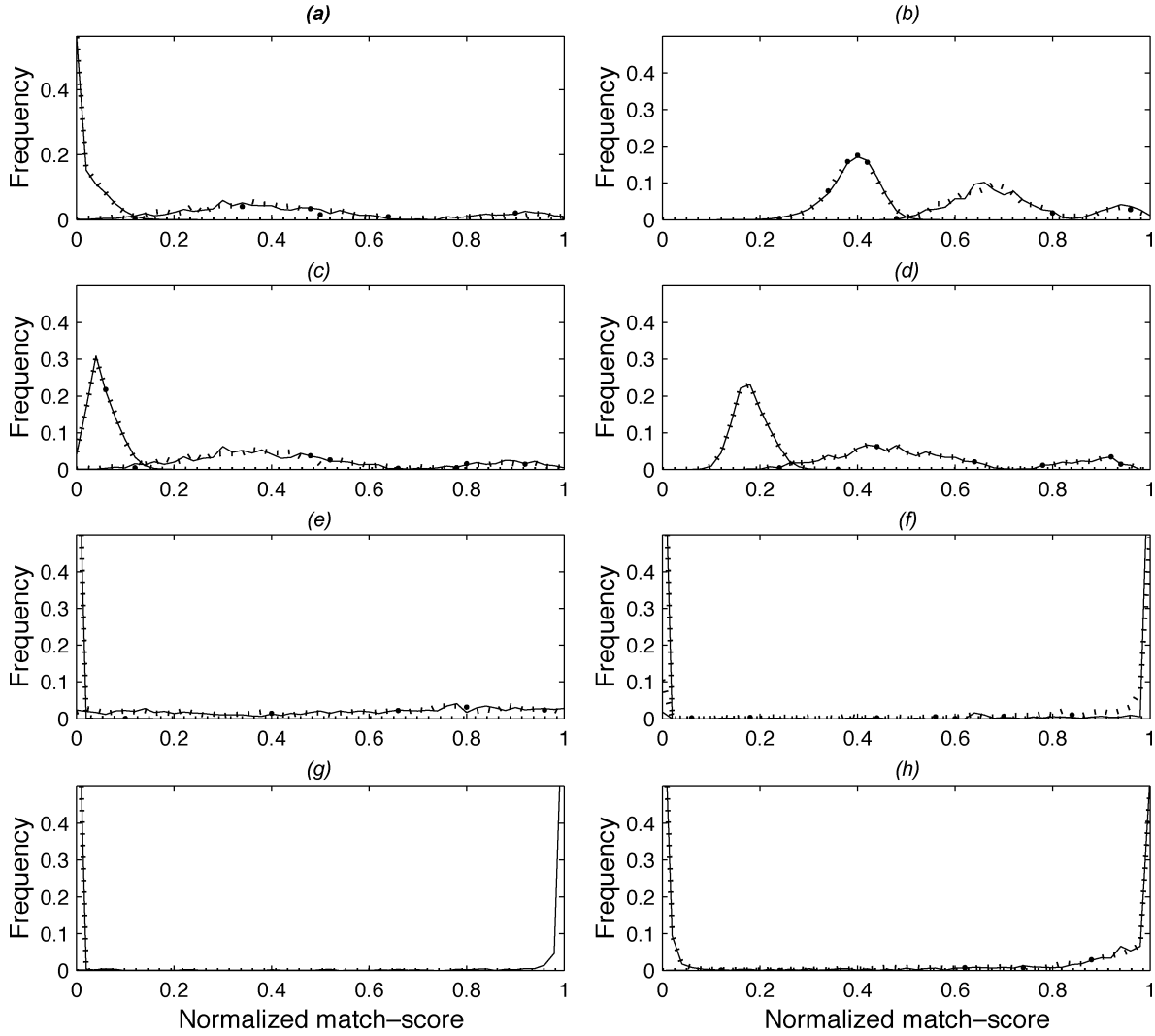

Fig. 8. Match score distributions for the compared combined systems (continuous: training data, dotted: test data). (a) OWM. (b) SUM. (c) PROD. (d) FLDA. (e) NBayes. (f) SVM-RBF. (g) Neural network. (h) GRM.

that of the 1-D decision based on single biometric. Our problem here is to find the best decision hyperplane from the data given by Fig. 6 and observe how much improvement can be achieved for both training and test sets.

As in previous examples, the network structure is selected according to the search procedure listed in Section IV using a ten-fold validation process. With the regularization parameter empirically set at $b = 10^{-4}$, the search covers the model orders $w \in [1, 3] \subset \mathcal{Z}$ and $r \in [1, 6] \subset \mathcal{Z}$. This results in a GRM network with $(w, r) = (1, 6)$ being chosen for the final training and test since it has the minimum MSE for the ten-fold validation. We shall label all results corresponding to this network as GRM in the sequel.

The performance of the GRM is compared with several decision fusion methods as seen in [1]–[3]: PROD, NN, RBF, NBayes, and FLDA.[1] PROD denotes a simple PRODUCT-rule (simplified version of those seen in [1], [3], and [28]). NN denotes the neural network method [2], and RBF denotes the Support Vector Machines (SVM) learning [2] using the RBF kernel [29]. Since application of the SVM using a comparable polynomial kernel of sixth-order did not converge, only the SVM-RBF is included for this comparison. NBayes is the Bayesian method assuming all inputs are independent [2]. FLDA is the Fisher's Linear Discriminant Analysis method optimizing the between-class over within-class scatters [2]. In addition, the well-known SUM-rule [30] and the OWM [15] are included for comparison. The SUM-rule (labeled as SUM for simplicity) uses the following scaling for each biometric output: $\bar{f}_1 = \hat{f}_1/a$, where $\hat{f}_1 \in [0, 115]$ (fingerprint: high score denotes genuine user) and $\bar{f}_2 = d(c - \hat{f}_2)$, where $\hat{f}_2 \in [0, 2.05]$ (speech: low score denotes genuine user). In this application, $a = 1.15, d = 48$, and $c = 2.05$ are selected arbitrarily. OWM is the linear optimal weighting method mentioned in

[1]The C4.5 decision tree algorithm [27] compared in [2] is also experimented. The resulted test False Reject Rate and False Accept Rate are, respectively, 0% and 6.04%. This is well above the False Accept Rate of the GRM, which is about 1.53% at the zero False Reject Rate point. No Receiver Operating Curve will be plotted for C4.5 since it has only one decision operating point.

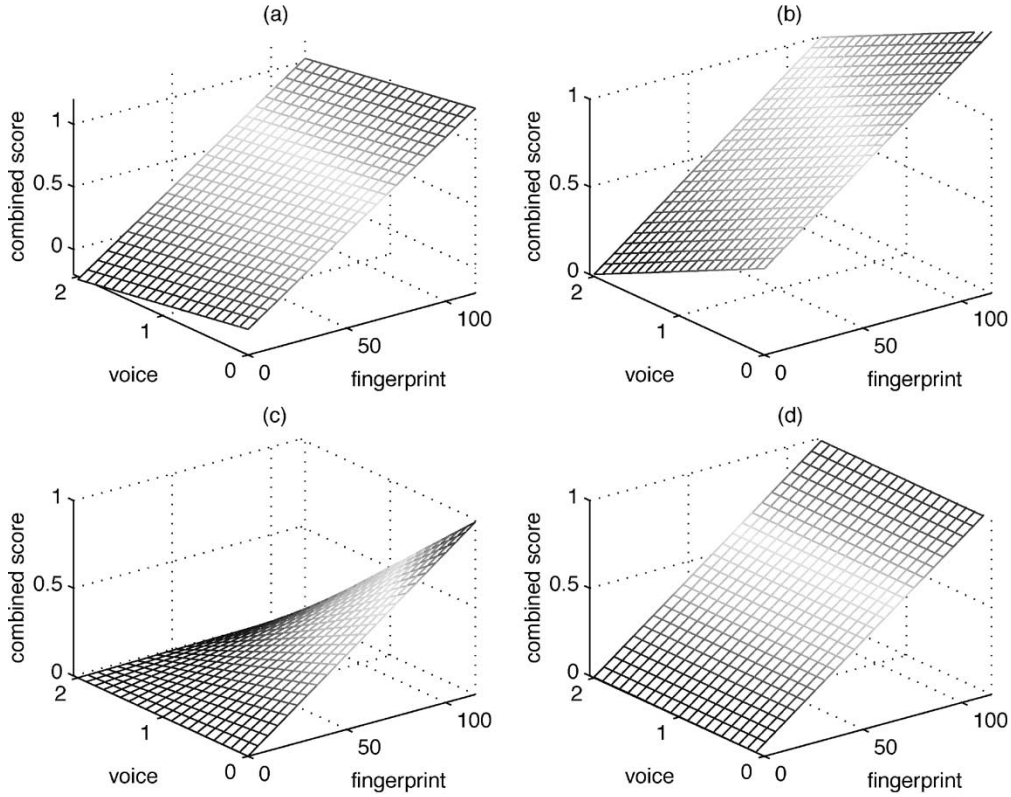

Fig. 9. Decision landscapes for OWM, SUM, PROD, and FLDA. (a) OWM. (b) SUM-rule. (c) PRODUCT-rule. (d) FLDA.

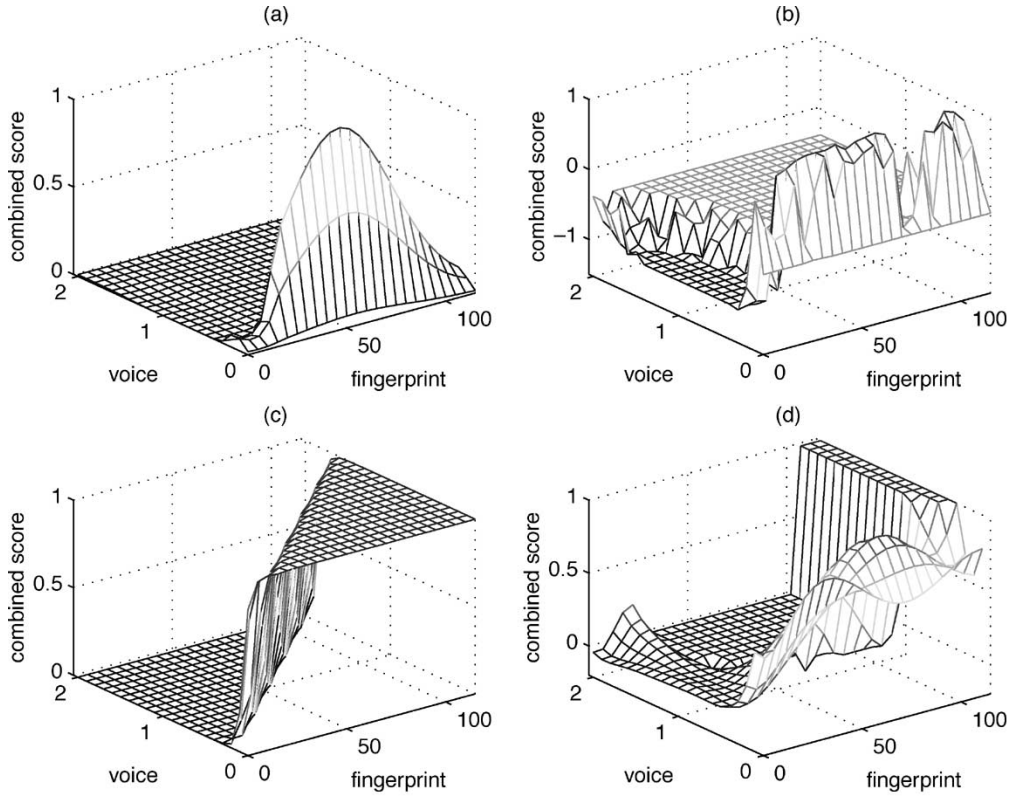

Fig. 10. Decision landscapes for NBayes, SVM-RBF, NN, and GRM. (a) NBayes. (b) SVM-RBF. (c) Neural network. (d) GRM.

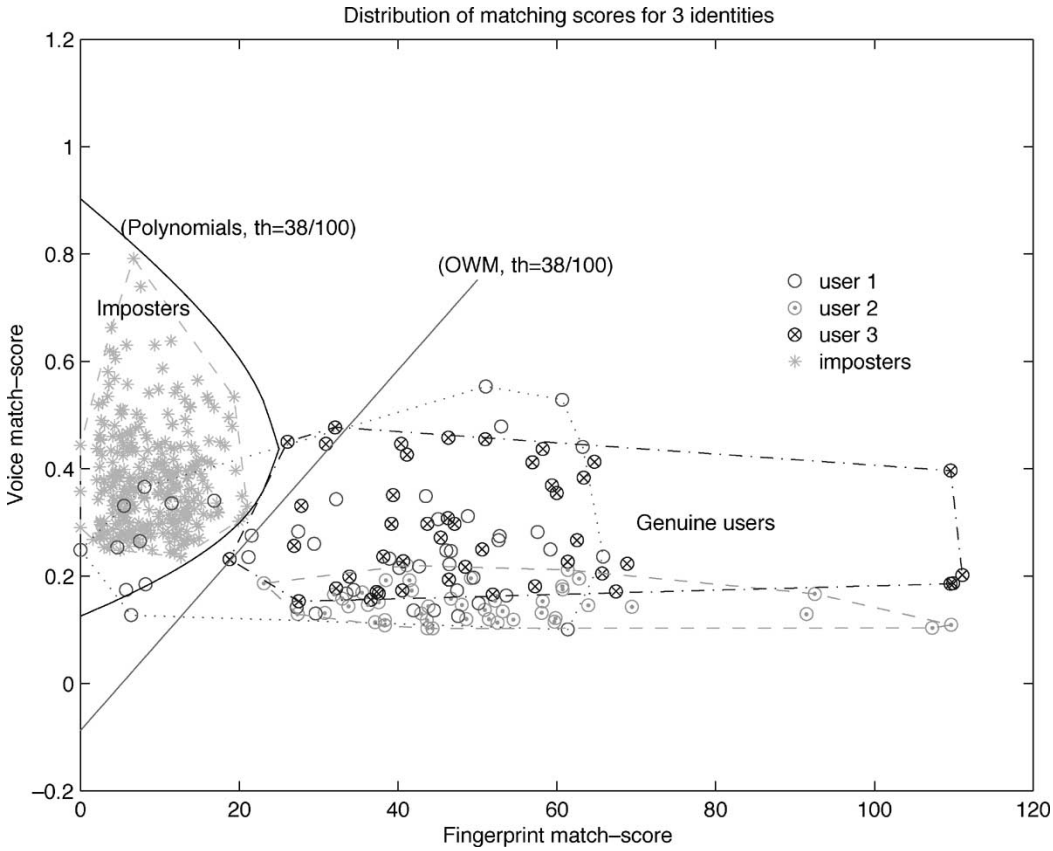

Fig. 11. Distribution of genuine and imposter scores for three users.

Section II-B. To be consistent in comparison, scalings similar to that in SUM are used in all compared methods.

Fig. 7 shows the training and test ROC results for the biometrics data fusion using the above-mentioned methods. The ROC curves for the original fingerprint and speaker verifications are also included in the same plots for comparison purposes. It can be seen from the test results that the proposed GRM improves significantly over the entire operating range of fingerprint and speaker verification systems as compared to the above-mentioned methods from literature. It is worth noting here that the authentic acceptance rates for GRM are found to be above 97% over the entire operating range for the given test set, and this is well above those of the OWM, FLDA, RBF, PROD, and NBayes, which range from 80% to 90% (OWM and FLDA have very close ROC performance). The performance of the GRM is about 1% to 2% better than that of the NN and SUM, even at this high operating range (above 97%) for the test data.

Remark 1: The reason that SUM outperforms the OWM can be explained using the match score distribution plots in Fig. 8 because the OWM [see scores distribution in Fig. 8(a)] optimizes more toward widening the score distributions as compared to SUM [see Fig. 8(b)]. As can be seen from the decision landscapes in Fig. 9(a)–(d), this widening of score distributions for OWM, however, does not improve upon the slope orientation responsible for segregating the much overlapped scores for the imposter and genuine classes. From the distribution plots (Fig. 8), it is also evident that the RBF, NN, and GRM optimize even more toward widening the score distribution as compared to the OWM. The widening has nonetheless been effective at spreading the overlapping imposters and genuine scores apart for the RBF, NN, and GRM methods as seen from their nonlinear decision landscapes in Fig. 10(a)–(d).

It is noted here that if good normalization and weighting can be found for SUM, it can be implemented with less memory than most of the above methods. However, tuning of these weighting parameters remains a trial-and-error effort even though it may be based on a small training data set. The GRM relieves the tuning effort by having a larger parameter memory requirement.

Remark 2: For the NN method, a $(2 \times 2 \times 1)$ sigmoidal network was selected as it was shown to be sufficient for the approximation and did not result in overfitting. Several training trials were performed using the Levenberg–Marquardt search, which was reported to have a faster convergence rate than that of Backpropagation. Over the several training trials, the best converged NN result was selected for this comparison. As the training in NN has always been iterative and kind of trial-and-error in initialization, the proposed GRM has an advantage over the NN method in terms of its least-squares optimal single-step learning. It is noted here that the NN took a total of 20 iterations (about 10.97 s per iteration using Matlab) for computation of nine neural weights. The RBF by SVM learning took about 2678 s to locate the solution containing 7641×3 learning parameters (one set for the 2-D support vectors and one set for

the Lagrange coefficients) under a similar computing platform. The FLDA took about 3.89 s and the NBayes took about 0.23 s for training. The OWM and GRM took, respectively, about 0.14 s and 2.12 s for the single-step computation of three and 29 model parameters under a similar computing platform using Matlab. The network structure search took approximately 225 s to go through the 18 network structures as much overheads were spent on data partitioning and computation of validation errors. Based on the above results, we see that the GRM is both accurate and computationally cost-effective for biometrics decision fusion.

The good performance of the GRM over the OWM can further be explained using a smaller data set as shown in Fig. 11, where the distributions of genuine and imposter scores are plotted for three different users. Due to the wide variation in characteristic distribution of the genuine scores of each user, the separation hyperplane from those imposters may well be nonlinear. It is clear from the figure that a linear decision with a threshold at 38/100 given by the OWM yields a lower classification accuracy than that of a nonlinear decision given by a polynomial network with a similar threshold.

Based on this observation, it may be suggested that the design of multiple modalities shall focus on obtaining narrow Gaussian-distributed imposter scores with a much-relaxed requirement for the genuine scores. A nonlinear separation hyperplane provided by the GRM can then be used to effectively classify the data.

VII. Conclusion and Future Work

In this paper, we proposed a generalized reduced multivariate polynomial network which belongs to a class of functional link networks for combining fingerprint and speaker verification decisions. Since computation of the model parameters is only a single-step procedure, the model allows a quick network structural search such that a suitable network size can be selected for possibly good generalization. Unlike the case of a general multivariate polynomial model, the number of parameters of this reduced model increases almost linearly with model order and number of inputs. The main advantage of this network model over more complex neural-network-like models is its single-step network parameters computation. The network model is first applied to a pattern recognition problem to illustrate its approximation capability. This is followed by a biometrics fusion problem combining fingerprint and voice data. The combined results using the proposed network show significant superiority of performance over several decision fusion methods from the literature in terms of the receiver operating characteristics. Our future work includes analysis of the approximation capability of the proposed network model and extension to other applications with much higher dimensions.

References

[1] L. Hong and A. Jain, "Integrating faces and fingerprints for person identification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 12, pp. 1295–1307, Dec. 1998.

[2] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of face and speech data for person identity verification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1065–1074, Sep. 1999.

[3] R. Brunelli and D. Falavigna, "Personal identification using multiple cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 10, pp. 955–966, Oct. 1995.

[4] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 3, pp. 418–435, May–Jun. 1992.

[5] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[6] L. I. Kuncheva, J. C. Bezdek, and R. Duin, "Decision templates for multiple classifier design: An experimental comparison," *Pattern Recognit.*, vol. 34, no. 2, pp. 299–314, Feb. 2001.

[7] C. M. Bishop, *Neural Networks for Pattern Recognit.*. New York: Oxford Univ. Press, 1995.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[9] K.-A. Toh, "Deterministic global optimization for FNN training," *IEEE Trans. Syst., Man Cybern., B*, vol. 33, no. 6, pp. 977–983, Dec. 2003.

[10] ——, "Global optimization by monotonic transformation," *Comput. Optim. Appl.*, vol. 23, pp. 77–99, Oct. 2002.

[11] F. Schwenker, H. A. Kestler, and G. Palm, "Radial-basis-function networks: Learning and applications," in *Proc. 4th Int. Conf. Knowledge-Based Intell. Engin. Syst. Allied Technol.*, Brighton, U.K., 2000, pp. 33–43.

[12] Y. Shin and J. Ghosh, "Ridge polynomial networks," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 610–622, May 1995.

[13] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Comput.*, vol. 25, no. 5, pp. 76–79, May 1992.

[14] A. Sierra, J. A. Macías, and F. Corbacho, "Evolution of functional link networks," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 54–65, Feb. 2001.

[15] N. Ueda, "Optimal linear combination of neural networks for improving classification performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 2, pp. 207–215, Feb. 2000.

[16] C. J. Merz, "A principal components approach to combining regression estimates," *Machine Learn.*, vol. 36, no. 2, pp. 9–32, Jul. 1999.

[17] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied Linear Regression Models*, 3rd ed., Chicago, IL: Irwin, 1996.

[18] A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 4, pp. 302–313, Apr. 1997.

[19] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proc. IEEE*, vol. 85, no. 9, pp. 1365–1388, Sep. 1997.

[20] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint databases," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 799–812, Aug. 1996.

[21] U. Halici, L. C. Jain, and A. Erol, "Introduction to fingerprint recognition," in *Intelligent Biometric Techniques in Fingerprint and Face Recognit.*, L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, and S. Tsutsui, Eds. Passfield, Hants., U.K.: Holborn, 1999, pp. 3–34.

[22] X. Jiang and W. Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proc. 15th Int. Conf. Pattern Recog.*, vol. 2, 2000, pp. 1042–1045.

[23] X. Jiang, W. Y. Yau, and W. Ser, "Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge," *Pattern Recognit.*, vol. 34, no. 5, pp. 999–1013, May 2001.

[24] J. M. Naik, "Speaker verification: A tutorial," *IEEE Commun. Mag.*, vol. 28, no. 1, pp. 42–48, Jan. 1990.

[25] C. Li and R. Venkateswarlu, "High accuracy connected digits recognition system with less computation," in *Proc. 6th World Multiconference Systemics, Cybernetics, Informatics*, Orlando, FL, Jul. 2002.

[26] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[27] O. L. Mangasarian, "Mathematical programming in neural networks," *ORSA J. Comput.*, vol. 5, no. 4, pp. 349–360, Sep. 1993.

[28] L. Hong, A. Jain, and S. Pankanti, "Can multibiometrics improve performance?" in *Proc. Auto ID*, Summit, NJ, 1999, pp. 59–64.

[29] J. Ma, Y. Zhao, and S. Ahalt, *OSU SVM Classifier Matlab Yoolbox (Ver 3.00)*, The Ohio State Univ., Columbus, OH, 2002, http://eewww.eng.ohio-state.edu/~maj/osu_svm/ [Online].

[30] A. Ross, A. Jain, and J.-Z. Qian, "Information fusion in biometrics," in *Proc. 3rd Int. Conf. Audio- and Video-Based Person Authentication (AVBPA)*, Halmstad, Sweden, June 2001, pp. 354–359.

Kar-Ann Toh (M'92–SM'03) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 1999.

Prior to his postdoctoral appointments at research centers in NTU from 1998 to 2002, he had nearly two years of work experience in the aerospace industry. Currently, he is affiliated with the Institute for Infocomm Research (Singapore). His research interests include biometrics and decision fusion, pattern classification, optimization, and neural networks. He has made several PCT filings related to biometric applications and has actively published papers in the above areas of interest.

"Dr. Toh has served as a member of the technical program committee for international conferences related to biometrics (AVBPA'05 and ICBA'06) and artificial intelligence (ICONIP'04 and ICNC'05). He has also served as a reviewer for international journals including several IEEE Transactions."

Wei-Yun Yau (S'90–M'98–SM'05) received the B.E.E. degree with Honors from the National University of Singapore in 1992. He received the M.Eng. degree in the field of biomedical image processing in 1995 and the Ph.D. degree in the area of computer vision in 1999, both from the Nanyang Technological University, Singapore.

From 1997 to 2002, he was a Research Engineer and then Program Manager at the Centre for Signal Processing, Singapore, leading the research and development effort in the area of biometric processing. His team won the top three positions in both speed and accuracy in the international Fingerprint Verification Competition 2000 (FVC2000). Currently, he is with the Institute for Infocomm Research as a Department Manager leading the research and development effort in the area of human computer interaction. His research interests include biomedical engineering, biometrics, computer vision, and intelligent systems.

He was a recipient of the Kuok Foundation Undergraduate Scholarship and his undergraduate project won the top prize in the Electronic Engineering/Telecommunications category of the Technology Fair in 1992.

Dr. Yau initiated and served as the Program Director of the Biometrics Enabled Mobile Commerce (BEAM) Consortium from 2001 to 2002. In addition, he actively participates in both national and international biometric standard activities. Currently he is the Chair of the Biometrics Pro-tem Technical Committee, Singapore. He is also the recipient of the TEC Innovator Award in 2002 and the Tan Kah Kee Young Inventors' Award 2003 (Merit).