

# Face Recognition by Extending Elastic Bunch Graph Matching with Particle Swarm Optimization

Rajinda Senaratne<sup>1</sup>, Saman Halgamuge<sup>1</sup>, Arthur Hsu<sup>2</sup>

<sup>1</sup>Department of Mechanical Engineering, Melbourne School of Engineering, The University of Melbourne, Australia;

<sup>2</sup>Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research, Australia

Email: rajinda\_s@hotmail.com, saman@unimelb.edu.au, hsu@wehi.edu.au

**Abstract**—Elastic Bunch Graph Matching is one of the well known methods proposed for face recognition. In this work, we propose several extensions to Elastic Bunch Graph Matching and its recent variant Landmark Model Matching. We used data from the FERET database for experimentations and to compare the proposed methods.

We apply Particle Swarm Optimization to improve the face graph matching procedure in Elastic Bunch Graph Matching method and demonstrate its usefulness. Landmark Model Matching depends solely on Gabor wavelets for feature extraction to locate the landmarks (facial feature points). We show that improvements can be made by combining gray-level profiles with Gabor wavelet features for feature extraction. Furthermore, we achieve improved recognition rates by hybridizing Gabor wavelet with eigenface features found by Principal Component Analysis, which would provide information contained in the overall appearance of a face. We use Particle Swarm Optimization to fine tune the hybridization weights.

Results of both fully automatic and partially automatic versions of all methods are presented. The best-performing method improves the recognition rate up to 22.6% and speeds up the processing time by 8 times over the Elastic Bunch Graph Matching for the fully automatic case.

**Index Terms**—Eigenfaces, Elastic Bunch Graph Matching, Face Recognition, Gabor wavelets, Hybridization, Particle Swarm Optimization, and Principal Component Analysis

## I. INTRODUCTION

Face recognition is a challenging problem in pattern recognition research. Many face recognition methods have been proposed in the past few years, and Elastic Bunch Graph Matching (EBGM) [1]–[5] is considered as one of the successful methods. In EBGM, a face is represented by a face graph (FG). The local features of a facial landmark are represented by a jet, where a jet is a set of Gabor wavelet features. A Face Bunch Graph (FBG) is created as a generalized representation of faces of various individuals, thus, it consists of a ‘bunch’ of jets corresponding to a landmark. To obtain the optimal FG to represent a new face, a two-step approach is adopted. The first step is to create a new FG for the new face, which will be fitted to the face in the second step. The fitting of the FG to the face is made by an iterative face graph matching procedure, where its geometrical structure is deformed until the graph similarity between the FG and the FBG is

maximized. When maximizing this graph similarity, the best matching jet is selected from the bunch.

Even though EBGM is considered as a successful technique, it has several deficiencies in obtaining the optimal FG:

- It uses only few sizes of FGs to estimate the size and the location of the face in an image (three sizes in stage 1, and two sizes in stage 2 [2]). The ability to use an FG of any size is more desirable as it may facilitate more accurate estimation of the size and the location of a face, so that the range of different sizes of faces that would exist in a generic face database can be tolerated.
- It initially scans the image only at selected locations (In stage 1, the image is scanned at locations on grid points of a lattice with a spacing of 4 pixels [2]). A non-exhaustive method that is capable of placing the FG at any location, rather than only at the selected locations, would be more effective.
- It is a computationally intensive algorithm [5]–[7], as it has several stages with extensive searches.

A more efficient approach to obtain the optimal FG for a new face would be to optimize the face graph matching procedure using a suitable optimization technique.

To overcome the above problems, inspired by both EBGM and Active Shape Model [8], a new method named Landmark Model Matching (LMM) [9], [10] was proposed, which incorporates an evolutionary optimization method known as Particle Swarm Optimization (PSO) [11]. LMM disregards the concept of using the entire ‘bunch’ of jets to locate a landmark. EBGM compares against the best matching jet in the bunch, whereas LMM compares against only the average of the jets. By using the entire bunch of jets, it would be possible to cover a wide variety of features that would exist in a face. Therefore, in this work, we propose a new method named EBGM<sub>PSO</sub>, which is an extension of EBGM that exploits its unique strength of using a bunch of jets to locate a landmark. In EBGM<sub>PSO</sub>, the face graph matching procedure of EBGM is optimized using PSO.

Above methods use Gabor wavelets for feature extraction to locate the landmarks. In this work, we extended Landmark Model Matching by combining Gabor wavelet

features with gray-level profiles (GLPs) for feature extraction. GLPs provide image intensity information that is unavailable in a jet. We refer to this method as  $LMM_{jets\&glps}$ , whereas we refer to the LMM previously published [10] as  $LMM_{jets}$ . We compare face localization and recognition results of the two proposed methods  $EBGM_{PSO}$  and  $LMM_{jets\&glps}$  with their previously published counterparts  $EBGM$  and  $LMM_{jets}$ , for the datasets provided with FERET database [12].

All four methods ( $EBGM$ ,  $EBGM_{PSO}$ ,  $LMM_{jets}$ , and  $LMM_{jets\&glps}$ ) can be used for face databases where only one image per person is available. They can be used as either fully automatic or partially automatic algorithms. In partially automatic algorithms, the eye coordinates are known, whereas in fully automatic algorithms, they are not. Improved recognition rates were achieved by fully automatic and partially automatic versions of the two proposed methods in comparison to their published counterparts. Furthermore, the computational cost was reduced by more than 85.0% for fully automatic versions, in comparison to the original  $EBGM$  technique.

All four methods use only Gabor wavelet features for recognition. Even though Gabor wavelets are capable of extracting local feature information, they may lack the overall appearance of a face that would be provided by appearance-based features such as eigenfaces in Principal Component Analysis (PCA) [13], [14]. Therefore, all four methods were hybridized with PCA, and the results are compared.

The organization of this paper is as follows. Section II describes the proposed fully automatic  $EBGM_{PSO}$  algorithm. Section III describes how LMM was extended by combining Gabor wavelet features and gray-level profiles. Section IV describes how Gabor wavelet features were hybridized with eigenface features. Section V presents the experimental results, followed by a summary and discussion in section VI.

## II. EBGM WITH PSO

In  $EBGM$ ,  $N$  points known as *fiducial points* [2], or *landmarks* [3], are selected at locations  $\vec{x}_n, n \in \{1, \dots, N\}$ , based on the a priori knowledge about the face. These  $N$  landmarks generally represent important features of a face which are decided by the user. A face is represented by an FG consisting of  $N$  nodes, corresponding to the  $N$  landmarks, and  $E$  edge vectors. Each node is labeled with a jet that describes the local features of the area surrounding the node. The  $E$  edge vectors  $\vec{x}_e, e \in \{1, \dots, E\}$ , are computed as  $\vec{x}_e = \vec{x}_n - \vec{x}_{n'}$  (displacements between the nodes), where  $e^{th}$  edge vector connects node  $n'$  ( $n' \in \{1, \dots, N\}$ ) with  $n$ . The geometrical structure of an FG, unlabeled by jets, is known as the *grid*. The nodes of the FG are overlaid on top of the landmarks of the face on a given image by deforming the grid. An FBG is created as a generalized representation of faces of various individuals. The grid of the FBG is identical to that of any FG. The FBG is designed to cover a wide range of possible variations in a number of faces. To

form the FBG,  $\alpha$  face images (referred to as FBG source images) that cover major deviations in a database, such as faces of different genders and races, and faces with open eyes, closed eyes, spectacles, etc., are selected. For each FBG source image, the  $N$  landmarks are manually selected, placing each of the  $N$  points on the selected location of the face. When a new image is given, the  $N$  nodes of the new FG that has to be computed are placed on the new image at the locations given by the average locations of the corresponding nodes of the  $\alpha$  grids of the FBG. The new FG is computed by extracting the jets at its nodes and calculating the edge vectors. For each node of the new FG, a jet similarity between its jet and the jet of the corresponding node of the FBG is calculated. The graph similarity between the new FG and the FBG is calculated based on the jet similarities and the edge vector differences. The optimal FG to represent the new face is found by maximizing the graph similarity, while deforming the new FG placed on the image.

PSO is an evolutionary computation technique [11], [15], which can be used to find optima in complex functions. It relies on the exchange of information between individuals, called *particles*, of the population, called *swarm*. Each particle represents a solution, and may consist of several dimensions. Particles move within the search space with velocities, which are dynamically adjusted according to their past trajectories. Each particle adjusts its trajectory toward its own previous best position, called *personal best*, and toward the best previous position attained by any member of the entire swarm, called *global best*. Therefore, the swarm efficiently uses the information available from its previous iterations to converge to the global optima.

The reasons behind choosing PSO and how a PSO algorithm works are discussed in our previous work [10]. One disadvantage of PSO is that it sometimes gets stagnated at local optima. However, the local optimum found by PSO is very close to the global optimum [16]. Thus, PSO would be a suitable method when quick solutions are desired, compromising the accuracy slightly.

The main phases of fully automatic  $EBGM_{PSO}$  are (1) creation of the FBG, (2) face finding, (3) landmark finding, and (4) recognition, as illustrated in Fig. 1. The set of known face images available is called the gallery. A new face that has to be recognized is called a probe (query) image, and the set of probe images used for testing is called the probe set. In the first phase, an FBG was created as described in subsection II-A. By using the FBG, the optimal FG for each image in the gallery is automatically computed as described in subsections II-B and II-C, and stored to represent the face. Two phases were used: face finding (FF) phase and landmark finding (LF) phase. In the face finding phase, the size and the location of a new face are estimated approximately, whereas in the landmark finding phase, the locations of the landmarks are found more accurately. Once the face had been approximately located by the face finding phase, the image is pre-processed, where the face region

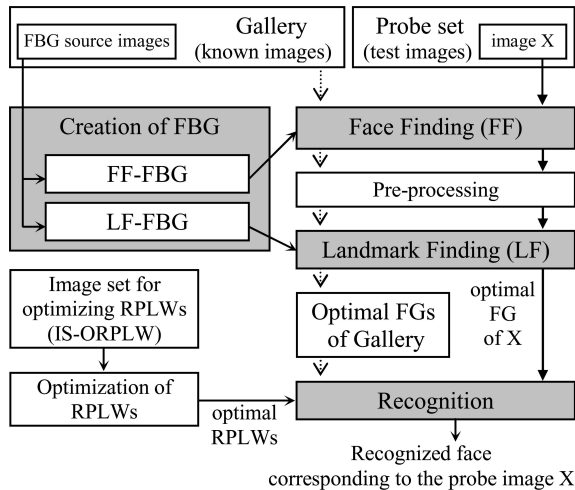


Figure 1. The phases of the fully automatic EBGM<sub>PSO</sub>

is cropped, resized, and centered, in order to be used as the input for the landmark finding phase. Therefore, the uncertainty that the landmark finding phase had to cope with the input image is reduced. When a probe image is given, its FG is computed and compared with the FG of every gallery image. The gallery image with the FG that matches best with the FG of the probe image, is selected as the recognized face image, as described in subsection II-D.

#### A. Creation of the Face Bunch Graph

To form the FBG, 48 images ( $\alpha=48$ ) were selected from the gallery (no images from the test sets were used to create the FBG). Two FBGs were created: face finding FBG (FF-FBG) for the face finding phase, and landmark finding FBG (LF-FBG) for the landmark finding phase. Since the purpose of the face finding phase is to estimate the location of the face and its size approximately, fewer landmarks are adequate for the face finding FBG, and conversely for the landmark finding phase. The number of landmarks was chosen intuitively. We selected 30 and 40 landmarks to form the face finding FBG and the landmark finding FBG, respectively. These landmarks are as same as those selected in [10]. All the faces in the two FBGs were normalized (similar to the way in CSU's EBGM [17]), so that their sizes were equal and the head orientations were straight.

To create each of the two FBGs, an FG was computed for each FBG source image as described below. For every FBG source image, a jet was computed at each landmark similar to the way in [10], and stored. A jet describes the local features of the area surrounding a landmark. The set of  $\alpha$  jets of all  $\alpha$  images corresponding to one node forms the bunch of that node. During matching of the FBG with the deformable FG, the jet computed at a node of the latter was compared with every jet in the bunch of the corresponding node of the FBG.

In our experiments, we intuitively selected 72 and 99 edge vectors for the face finding FBG and the landmark finding FBG, respectively, as shown in Fig. 2. The

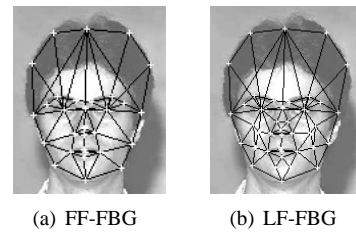


Figure 2. Edge vectors of the FF-FBG and the LF-FBG

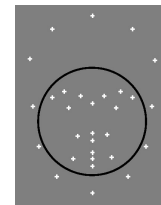


Figure 3. Interior and head-boundary nodes of the FF-FBG

corresponding edge vectors of the  $\alpha$  FGs of the FBG source images were averaged, and stored. Furthermore, the average landmark locations were found by averaging the node locations corresponding to the same landmark over the  $\alpha$  FBG source images.

#### B. Face Finding Phase

The face finding phase was carried out in two stages. First, the head was approximately located in stage 1, and then the face region was approximately located in stage 2.

1) *Stage 1:* The face finding FBG was formed by averaging over the  $\alpha$  jets of the nodes corresponding to the same landmark. Instead of using all the jets of an entire bunch, using the average jet of the bunch reduces the computational cost in initial stages. This approach is similar to the initial stage of the original EBGM, as the average jet of the bunch is used in stage 1 of the original EBGM [2].

The FG corresponding to a particle was deformed by varying PSO dimensions of that particle. The nodes of the deformable FG that has to be fitted to the new image were grouped into two rigid grids: the interior-nodes-grid, formed by the interior nodes shown inside the circle in Fig. 3, and the head-boundary-nodes-grid, formed by the nodes on the head boundary. In a rigid grid, the nodes are at fixed locations relative to each other, however, the grid as a single structure, is allowed to change its location. The size and the location of each grid were varied using a PSO algorithm to find their optimal values. The size of a grid was varied by multiplying the grid structure with a size factor. In order to vary the location of a grid, the center point between the two eye nodes was used as a reference point.

In the PSO algorithm, a particle corresponds to a deformable FG. This FG is deformed by varying the values of the dimensions of the particle. In this stage, 8 particles were used, and they were initialized in a

uniformly distributed manner. Each particle consists of 6 dimensions:

- $x$  and  $y$  coordinates of the location of the reference point corresponding to the interior-nodes-grid:  $x_1$  and  $y_1$ ,
- its size factor (to scale the grid):  $sf_1$ ,
- $x$  and  $y$  coordinates of the location of the reference point corresponding to the head-boundary-nodes-grid (as a shift from  $x_1$  and  $y_1$ ):  $x_2$  and  $y_2$ , and
- its size factor (as a multiplication factor for  $sf_1$ ):  $sf_2$ ; i.e., the size factor used to scale the head-boundary-nodes-grid =  $sf_2 \times sf_1$ .

The final node locations of the deformable FG calculated in each iteration using the above dimensions are rounded to integer values, since the pixel locations of an image are always integers. The jet similarity  $S_a(J, J')$  between two jets  $J$  and  $J'$  was calculated as  $S_a(J, J') = \frac{\sum_{j=0}^{39} a_j a'_j}{\sum_{j=0}^{39} a_j^2 \sum_{j=0}^{39} a'^2_j}$ , where  $a_j$  and  $a'_j$  are the magnitudes of the  $j^{th}$  component of the jets  $J$  and  $J'$ , respectively. The jet  $J' = J(\vec{x})$  of a node of the FG is computed at variable locations  $\vec{x}$  in the new image. The jet  $J$  is from the corresponding node of the face finding FBG. As the face finding FBG is used as an average FBG,  $J$  is the average of the  $\alpha$  jets of that node. The true position of a landmark (actual position of the landmark on the given image) can be located by finding the pixel location in the image that gives the maximum jet similarity. The graph similarity between the FG of the new image and the face finding FBG was computed as  $GS(FG, FF-FBG) = \frac{1}{N} \sum_{n=1}^N w_n S_a(J_n^{FG}, J_n^{FF-FBG})$ , where  $N$  is the number of nodes ( $N=30$ ),  $J_n^{FG}$  is the jet at node  $n$  of the FG,  $J_n^{FF-FBG}$  is the jet of node  $n$  of the face finding FBG, and  $w_n$  is the weight assigned to node  $n$ . The objective function maximized by PSO was the graph similarity. The weights  $w_n$ s were determined intuitively and experimentally by testing with a few different combinations of different weights on a few images.

A particle tries to achieve a higher graph similarity value by adjusting its variables of the 6 PSO dimensions after every iteration. The velocities calculated based on the history of the graph similarity values of the particles, guide the swarm to the maxima. PSO iterations were terminated if the best solution did not improve in 10 consecutive iterations, or the number of iterations reached 20. For example, the nodes of an FG that corresponds to a particle during initialization and after iterations 1, 2, 3, 4, 5, 8, 12, 16 and 20, are shown in Fig. 4. The size factor and the location of the reference point of the interior-nodes-grid corresponding to the best solution, were the input for the stage 2.

The original EBGm uses the gradient descent method to locate a landmark. Since many landmarks must be reliably located, using only the gradient descent method would be inefficient as it may often get stagnated at local optima when the number of dimensions of the search space increases. Therefore, the original EBGm uses a heuristic algorithm in conjunction with the gradient descent method to optimize the face graph matching

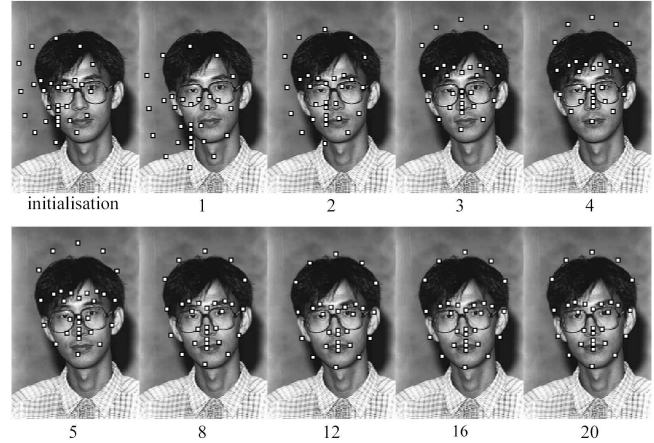


Figure 4. Nodes of a face graph that corresponds to a particle during initialization and after iteration 1, 2, 3, 4, 5, 8, 12, 16 and 20

procedure. As discussed earlier in section I, this approach is inefficient. Hence, we used PSO and improved the search mechanism in EBGm<sub>PSO</sub>.

2) *Stage 2*: In this stage, each node was moved locally to find its best fitting location. The search space was much smaller, since the head had already been approximately located in the previous stage. Similar to the way in [2], the edge vectors, in addition to jets, were used to compute the graph similarity.

### C. Landmark Finding Phase

Landmark finding phase is similar to face finding phase. In contrast, the locations of the landmarks are found more accurately. Instead of face finding FBG, landmark finding FBG was used. The landmark finding FBG was used without averaging (i.e., the entire bunch of a node is considered for matching). After running the landmark finding phase, the jets of the landmarks were computed on the histogram-equalized images, and stored to represent the face.

### D. Recognition Phase

When a probe image is given, its FG is computed and a similarity score between this FG and the FG of each gallery image is computed. The gallery image with the highest similarity score is selected as the recognized face.

Let the probe image be  $P$ , and the  $x^{th}$  image in the gallery be  $G^x$ . The similarity score between  $P$  and  $G^x$  is calculated as,  $MS(P, G^x) = \sum_{n=1}^N w_n j s_n^x$ , where  $j s_n^x$  is the jet similarity between  $n^{th}$  landmark of  $G^x$  and that of  $P$ ,  $w_n$  is the weight assigned to  $j s_n^x$ , and  $N$  is the number of landmarks. We refer to the weights  $w_n$ s as Recognition-Phase-Landmark-Weights (RPLWs). Assigning different weights to different landmarks varies the capability of a method to distinguish/discriminate different faces. Therefore, with the optimal RPLWs, highest recognition rates would be achieved.

1) *Optimization of Recognition-Phase-Landmark-Weights*: In the previously presented LMM [10], three data sets known as *fa*, *fb*, and *dup1* of the third FERET test [12] were used to test the performance of the LMM algorithm. The same data sets were used to optimize the RPLWs. Intention of using all the images of the above three data sets was to achieve a generalized set of RPLWs which would represent a large set of samples. Even though a large number of images is desirable to optimize the RPLWs, they should be independent of testing images. If the testing images are used to optimize the RPLWs, the resulting optimal RPLWs would be fine-tuned for the testing data sets, thus would produce higher recognition rates than that would be produced by a set of RPLWs optimized using a different data set. Hence, they cannot be considered as a generalized and reasonable set of RPLWs. Therefore, the approach in the previously presented LMM [10] may be inappropriate. In order to obtain a more fair and generalized set of optimal RPLWs, we used 200 independent image pairs of 200 different individuals (400 images, exclusive of *fa*, *fb*, and *dup1* sets) from the FERET database. We refer to this image set as IS-ORPLW (image set for optimizing RPLWs). As the gallery of IS-ORPLW, 200 images were used, and as the probe set of IS-ORPLW, the other 200 images which correspond to the same individuals of the 200 gallery images, were used.

We optimized the RPLWs by using a PSO algorithm, similar to the way in the previous LMM [10]. By its nature, PSO may not produce the same result for different runs because it is a randomized search process. Therefore, the optimal FGs found for the gallery and the probe images of IS-ORPLW were slightly different for different runs. This resulted in slightly different sets of RPLWs for different runs. Hence, they were averaged over three runs, to obtain a reasonable and generalized set of RPLWS.

### III. EXTENDING LANDMARK MODEL MATCHING

In LMM [10], a face is represented by a Landmark Model (LM). A Landmark Distribution Model (LDM) is created using a few gallery images (referred to as LDM source images). The optimal LM that represent a new face can be found by a model matching procedure where the similarity between the LM and the LDM is maximized. LMM uses Gabor wavelets for feature extraction in the model matching procedure. Gabor wavelets are less sensitive to lighting variations, and robust against translation, distortion, and rotation [2]. Even though Gabor wavelets provide local frequency information, they may lack the information provided by the image intensities. Intensity variations, e.g. a uniform light patch and a uniform dark patch, can easily be distinguished by gray-level profiles (GLPs), whereas Gabor wavelets cannot. Therefore, we propose extending LMM by combining the features extracted by both Gabor wavelets and 2D GLPs.

The LDM can be considered analogous to the FBG in EBGM<sub>PSO</sub>. Two LDMs were formed: face finding LDM (FF-LDM) for the face finding phase, and landmark

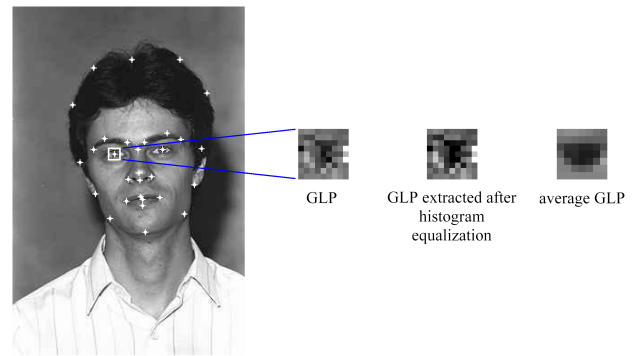


Figure 5. GLP extraction process

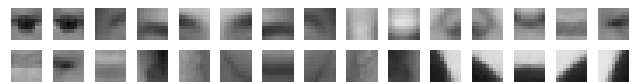


Figure 6. The 30 average GLPs corresponding to the 30 nodes of the FF-LDM (ordered based on the node number shown in Fig. 1 in [10])

finding LDM (LF-LDM) for the landmark finding phase. The face finding LDM consists of 30 nodes, the same as those of face finding FBG in EBGM<sub>PSO</sub>. The landmark finding LDM consists of 40 nodes, the same as those of landmark finding FBG in EBGM<sub>PSO</sub>. Each LDM was formed using the same images that were used to form the FBG in EBGM<sub>PSO</sub>.

For each LDM source image, an LM was computed as follows: Jets were computed at the landmarks of the faces in LDM source images, and averaged over the  $\alpha$  LMs of the LDM. Following the approach used in stage 1 of the original EBGM [2], only the average magnitude of the Gabor wavelet coefficients in a bunch were used. Experiments were conducted using the average of the phase of the Gabor wavelet coefficients, but did not improve the performance.

For every LDM source image, a GLP was extracted at each landmark. Each LDM source image was histogram equalized prior to the extraction of GLPs, in order to reduce the effects of different lighting conditions. The GLP  $G(i)$  is the set of image intensity values of the square window within  $\pm p_n$  pixels surrounding the node represented as a vector, where  $i \in \{1, \dots, s_{win}^2\}$ , and  $s_{win} = 2p_n + 1$ . We chose  $p_n = 4$  for the face finding LDM, and  $p_n = 3$  for the landmark finding LDM. The set of  $\alpha$  GLPs of all  $\alpha$  LDM source images corresponding to one node were averaged to form an average GLP for that node of the LDM. The GLP extraction process of the left eye is illustrated in Fig. 5 for one LDM source image. The right-most image shows the average GLP calculated by averaging all the GLPs of the left eye of all the LDM source images. The 30 average GLPs corresponding to the 30 nodes of the face finding LDM are shown in Fig. 6.

The face finding and landmark finding phases were similar to those in previously presented LMM [10]. The LM was deformed using the the principal components of the LDM as described in previ-

ous LMM [10]. The jet similarity  $S_a(J, J')$  was calculated similarly as in  $EBGM_{PSO}$ . The *total jet similarity* was calculated as,  $S_{tot_a}(LM, FF-LDM) = \sum_{i=1}^n w_i S_a(J_i^{LM}, J_i^{FF-LDM})$ , where  $n$  is the number of nodes ( $n=30$ ),  $J_i^{LM}$  is the jet at node  $i$  of the LM,  $J_i^{FF-LDM}$  is the jet of node  $i$  of the face finding LDM, and  $w_i$  is the weight assigned to node  $i$ . The *GLP similarity* between two GLPs  $G$  and  $G'$  was calculated as,  $S_{glp}(G, G') = 1 - \frac{1}{I_{max} s_{win}^2} \sum_{j=1}^{s_{win}^2} |G(j) - G'(j)|$ , where  $G'$  is the GLP of a node of the LM computed at variable locations  $\vec{x}$  in the new image,  $G$  is from the corresponding node of the face finding LDM, and  $I_{max}$  is the maximum possible intensity ( $I_{max}=255$ ). The *total GLP similarity* between the LM and the face finding LDM was computed as,  $S_{tot_{glp}}(LM, FF-LDM) = \sum_{i=1}^n w_i S_{glp}(G_i^{LM}, G_i^{FF-LDM})$ , where  $n$  is the number of nodes ( $n=30$ ),  $G_i^{LM}$  is the GLP at node  $i$  of the LM,  $G_i^{FF-LDM}$  is the GLP of node  $i$  of the face finding LDM, and  $w_i$  is the weight assigned to node  $i$ . The objective function maximized by PSO was the model similarity between the LM and the face finding LDM, which was computed as,  $MS(LM, FF-LDM) = w_a S_{tot_a} + w_{glp} S_{tot_{glp}}$ , where  $w_a$  and  $w_{glp}$  are the weights assigned to combine  $S_{tot_a}$  and  $S_{tot_{glp}}$ , respectively. We intuitively chose  $w_a = w_{glp} = 0.5$ .

#### IV. HYBRIDIZING GABOR WAVELET AND EIGENFACE FEATURES

Even though the Gabor wavelets are capable of extracting rich local feature information, they may lack the overall appearance of a face that would be provided by appearance-based features such as eigenfaces. Therefore, when performing the recognition in the recognition phase, it would be possible to improve the recognition rates of all four methods:  $EBGM$ ,  $EBGM_{PSO}$ ,  $LMM_{jets}$  and  $LMM_{jets\&glps}$  by hybridizing them with PCA [13]. The hybridization process is similar for all four methods. For example, consider the case of hybridizing  $EBGM_{PSO}$  with PCA, as illustrated in Fig. 7. Once the FG has been found by fully automatic  $EBGM_{PSO}$ , the face region in each image was masked to avoid the effect of the background. To select the face region, the landmark locations on the head-boundary estimated by the landmark finding phase were used, as indicated by the dashed-line arrow in Fig. 7. Images were resized to  $80 \times 100$ , and histogram equalization was performed to reduce the effect of lighting variations.

##### A. Computation of Eigenface Features

An eigenface-based similarity score,  $SS_{ef}$ , between a given probe image and each gallery image was computed by performing PCA [13]. All the gallery images were used to form the covariance matrix. The number of eigenfaces used for image representation was selected such that 95% of the total variance was retained.

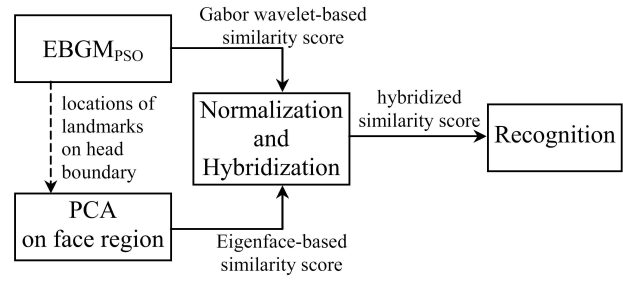


Figure 7. Hybridization of fully automatic  $EBGM_{PSO}$  with PCA

##### B. Hybridization with Gabor Wavelet Features

Hybridizing Gabor wavelet features with eigenface features have not been investigated in previous works, even though the performance of applying PCA to Gabor wavelet features had been investigated [18]. Methods of Hybridizing different classifiers based on probabilistic estimations have been investigated in previous works [19], [20]. These methods are more suitable when multiple samples per class are available. However, the number of samples available per class (person) in face recognition is often very limited in practice, due to the time, effort and cost involved in collecting data [21]. Hence, the practicality is taken into consideration in this work, we focus on using only one image per subject for the gallery. In such a case, score-level fusion has to be performed to hybridize different methods [21]–[23].

1) *Normalization of similarity scores*: Prior to hybridization, similarity scores obtained by different methods must be normalized in order to transform them into a common domain, so that they are comparable. A number of score normalization methods have been investigated in previous works [21], [23], [24]. We tested three well-known normalization techniques: Min-Max, Z-score, and Tanh. They were implemented similar to the way carried out by Snelick et. al. [24], but adapted to suit face recognition context. Brief descriptions and the modified changes are given below:

**Min-Max**: This method maps the raw similarity scores to the  $[0,1]$  range, based on their minimum and maximum possible values. Min-Max normalization is suited for the case where the minimum and the maximum bounds of the similarity scores produced by a method is known [21]. In the face recognition problem, it is not generally possible to find the bounds for any probe image using Min-Max normalization. However, it is possible to estimate the bounds that are specific for a given probe image, and then use them to normalize the raw similarity scores.

**Z-score**: This method transforms the raw similarity scores to a distribution with a mean ( $\mu$ ) of 0 and a standard deviation ( $\sigma$ ) of 1. One advantage of this method is that  $\mu$  and  $\sigma$  can be estimated by using a fixed training set. We used IS-ORPLW as the training data set to estimate  $\mu$  and  $\sigma$ .

**Tanh**: This method maps the raw similarity scores to the  $[0,1]$  range using a hyperbolic tangent function, based on  $\mu$  and  $\sigma$ . Both  $\mu$  and  $\sigma$  can be estimated as in Z-score

normalization.

2) *Hybridization of similarity scores*: For each normalization method, we tested LMM with four well-known fusion rules [24]. Let the normalized similarity score between a given probe image and the  $x^{th}$  gallery image be  $NSS_i^x$  for the  $i^{th}$  method, the hybridized similarity score for the given probe image be  $HSS^x$ , and the number of methods being hybridized be  $N_m$  (in our work,  $N_m=2$ ). Following fusion rules were tested:

- *Simple-Sum*:  $HSS^x = \sum_{i=1}^{N_m} NSS_i^x$ .
- *Min-Score*:  $HSS^x = \min_i (NSS_i^x)$ .
- *Max-Score*:  $HSS^x = \max_i (NSS_i^x)$ .
- *Weighted-Sum*:  $HSS^x = \sum_{i=1}^{N_m} w_i NSS_i^x$ , where  $w_i$  is the hybridization weight assigned to the  $i^{th}$  method, and  $\sum_{i=1}^{N_m} w_i = 1$ .

Fusion rules such as Simple-Sum, Min-Score, and Max-Score have been investigated in [21], [24]. Simple-Sum rule can be considered as a special case of the Weighted-Sum rule, where equal weights are assigned to different methods. It has been used in a number of previous work [22], [25]. Assigning equal weights may not be the optimal way of fusing different similarity scores. Therefore, we propose to optimize the hybridization weights. By optimizing hybridization weights, it would be possible to find the optimal relative contribution from each method toward the single hybridized method that would produce the optimum hybridized similarity score for a given probe. In our work, the two methods are the Gabor wavelet-based method and the PCA. Hybridization weights were optimized using a PSO algorithm, as described in subsection IV-C. The gallery image with the highest  $HSS$  was selected as the recognized face for the given probe image.

### C. Optimization of Hybridization Weights

We used the images in IS-ORPLWs to find the optimal hybridization weights. Consider the case of hybridizing LMM<sub>jets</sub> with PCA.

Let the  $x^{th}$  image in the gallery of the IS-ORPLW be  $G^x$ , and the  $y^{th}$  image in the probe set of the IS-ORPLW be  $P^y$ . Let the corresponding image of  $P^y$  in the gallery be  $f(P^y)$ , where  $f$  is the function that gives the corresponding image in the gallery for the given probe image  $P^y$ . The hybridized similarity score  $SS_{com}(P^y, G^x)$ , between  $P^y$  and  $G^x$  is calculated as,

$$HSS_{com}(P^y, G^x) = w_{wav} SS_{wav}^{y,x} + w_{ef} SS_{ef}^{y,x}, \quad (1)$$

where  $SS_{wav}^{y,x}$  is the Gabor wavelet-based similarity score between  $P^y$  and  $G^x$ , and  $SS_{ef}^{y,x}$  is the eigenface-based similarity score between  $P^y$  and  $G^x$ . Let  $G^{y*}$  be the gallery image that has the highest hybridized similarity score with  $P^y$ ; i.e.,

$$G^{y*} = G^{x*} |_{x^* = \arg \max_x \{HSS(P^y, G^x)\}}. \quad (2)$$

Then, the recognition rate was calculated as,  $RR = \{\sum_{y=1}^{N_P} c_y\} / N_P$ , where  $N_P$  is the number of images in the probe set, and  $c_y$  is given by,

$$c_y = \begin{cases} 1, & \text{if } G^{y*} = f(P^y), \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

TABLE I.  
COMPARISON OF RECOGNITION RATES (%) FOR DIFFERENT  
NORMALIZATION METHODS AGAINST DIFFERENT FUSION RULES

Algorithm	Probe set: <i>fb</i>			Probe set: <i>dup1</i>		
LMM <sub>jets</sub>	92.1			51.6		
PCA	51.6			17.7		
Normalization:	(a)	(b)	(c)	(a)	(b)	(c)
Fusion rule						
Simple-Sum	73.4	83.3	83.3	30.9	37.7	37.7
Min-Score	68.7	67.1	67.1	28.1	29.5	29.5
Max-Score	69.7	83.0	83.0	26.6	38.1	38.1
Weighted-Sum	<b>92.7</b>	92.6	92.6	<b>52.1</b>	51.7	51.7

(a) Min-Max (b) Z-score (c) Tanh

The objective function maximized by PSO was the recognition rate,  $RR$ . A particle corresponds to  $w_{wav}$ , and consists of only one dimension. We used 20 particles, and 70 PSO iterations. The allowed range of the search space for the dimension was from 0 to 1.

To compare the performance of different normalization methods against different fusion rules, average of recognition rates were calculated for LMM<sub>jets</sub> algorithm over 3 runs. To use the Weighted-Sum rule with optimal hybridization weights, for each normalization method we found the optimal  $w_{wav}$  as described above for three runs on IS-ORPLWs, and then they were averaged to obtain an optimal generalized set of hybridization weights for that normalization method. The recognition rates are compared in Table I for different normalization methods against different fusion rules. The best result for each probe set is shown in boldface. The Weighted-Sum rule with optimal hybridization weights produced the highest recognition rates. Recognition rates achieved by other fusion rules are less than those of LMM<sub>jets</sub>. When the Weighted-Sum rule is used with optimal hybridization weights, the recognition rates achieved by different normalization methods are similar. This shows that the effect of normalization is small when the hybridization weights are optimized. Since Min-Max normalization method produced the best results, we chose it as the normalization method for the comparison of the other algorithms. When Min-Max normalization was used, the optimal weights calculated for LMM were,  $w_{wav}=0.946$  and  $w_{ef}=0.054$ . Similarly, an optimal generalized set of hybridization weights were found when hybridizing EBGMP<sub>SO</sub> with PCA by using Min-Max normalization, and they were,  $w_{wav}=0.952$  and  $w_{ef}=0.048$ .

## V. EXPERIMENTAL RESULTS

### A. Data Sets Used

In our experiments, three standard data sets known as *fa*, *fb*, and *dup1* of the third FERET test [12] were used. The images in all three sets had been selected from the FERET database, and taken on the frontal pose. The *fa* set was used as the gallery. The *fb* and the *dup1* sets were used as probe sets. The images in the *fb* set had

been taken on the same session that the images in the *fa* set had been taken, whereas the images in the *dup1* set had been taken on a different day. Every image in each probe set has a corresponding image of the same person in the gallery. The *fa* and the *fb* sets consist of 1196 and 1195 images, respectively, with one image per person. The *dup1* set consists of 722 images, with one or more images per person. As expected, the variations among the corresponding faces of the *fa* and the *dup1* sets are larger than those of the *fa* and the *fb* sets. Therefore, the faces in the *dup1* set are more difficult to recognize than those in the *fb* set. When creating the FBG and the LDM, 48 images from the gallery were selected. Obviously, no images from the testing sets (*fb* or *dup1*) were used to create the FBG or LDM.

*B. Face Finding Results and the Computational Costs*

We assume that the mid point of the two eyes would be an approximate representation for the location of a face. We compared the error between the true midpoint calculated using the known eye coordinates and the estimated mid point between the eyes. Since the sizes of the faces were significantly different, measuring the face localization error based on the pixel error between the true location and the estimated location, would not be sensible. This is due to the fact that, an error of one pixel on a smaller face is more significant than that on a larger face. Therefore, it would be more appropriate to measure the error relative to the size of the face. Hence, the error  $\epsilon$  was calculated relative to the distance between the left and the right eye as,  $\epsilon = \|\vec{x}_{true} - \vec{x}_{est}\| / \|\vec{x}_{lefteye} - \vec{x}_{righteye}\|$ , where  $\vec{x}_{true}$  is the true location of the mid point between the two eyes,  $\vec{x}_{est}$  is the estimated location of the mid point between the two eyes,  $\vec{x}_{lefteye}$  and  $\vec{x}_{right}$  are the true locations of the left and the right eye, respectively. The mean error was calculated by averaging the errors over these 1196 images. Table II shows the mean error, and the computational costs of the face finding phase only and both the face finding and landmark finding phases of fully automatic EBGM<sub>PSO</sub> and LMM for one image, compared with the fully automatic EBGM algorithm (cost assumed as 100%) implemented according to [2] and [3]. The computational costs of the recognition phase for any of the methods (EBGM, EBGM<sub>PSO</sub>, LMM, PCA, and hybridized versions) are not shown as they are comparatively negligible. Two experiments were performed for LMM: In LMM<sub>jets</sub>, only the Gabor wavelet features were used, whereas in LMM<sub>jets&glps</sub>, both the Gabor wavelet features and the GLPs were used. The recognition rates obtained by using only the GLPs were much lower, thus, not shown. EBGM<sub>PSO</sub> and LMM<sub>jets&glps</sub> reduced the computational cost of EBGM by 85.5% and 87.7%, and the mean error of the face finding phase by 3.4% and 5.8%, respectively. Gabor wavelet features of all the methods in Table II were computed in spatial domain. Advantages of computing them in spatial domain over Fourier domain are discussed in [26].

TABLE II.  
FACE FINDING ERRORS AND COMPUTATIONAL COSTS

Algorithm	Mean error of face finding		Computational cost relative to * (%)	
	pixels	$\epsilon$ (%)	only FF	FF and LF
EBGM	4.6	13.8	44.0	100.0*
EBGM <sub>PSO</sub>	3.5	10.4	4.6	14.5
LMM <sub>jets</sub>	2.8	8.4	2.7	12.1
LMM <sub>jets&amp;glps</sub>	2.7	8.2	3.0	12.3

By its nature, PSO may not produce the same result for different runs. Furthermore, the number of iterations we used was small. Therefore, the results of EBGM<sub>PSO</sub> and LMM were slightly different for different runs. The averaged values were taken over three runs.

*C. Face Recognition Results*

1) *Computing Recognition Rates:* Face recognition rate is the percentage of the number of probe images recognized (correctly matched to the corresponding image in the gallery) to the total number of probe images. They are compared for all the methods in Table III. The results of the fully automatic and partially automatic algorithms are shown in the top and the bottom halves of Table III, respectively. We consider original EBGM [12] and LMM<sub>jets</sub> [10] as the reference methods. Previously published results are shown in vertical boldface, and the best method in general is shown in italic boldface. In [10], only the recognition rates of fully automatic LMM<sub>jets</sub> were reported. In this work, we found its partially automatic recognition rates, and they are not shown in vertical boldface as they have not been reported elsewhere.

All PCA methods are performed on the input images, where the face region was selected using the landmark locations found by fully automatic methods in their subscript. Recognition rates of PCA alone are not shown as they are not competitive with other methods, including the reference methods. Hybridized versions of the algorithms are indicated by '+'. The ranking of the best three methods for each probe set are shown within parentheses.

As described in subsection V-B, the results of the methods that use PSO were slightly different for different runs. Therefore, the recognition rates of the algorithms that use PSO were averaged over three runs.

As can be seen from Table III, in this work, we have improved the recognition rates by several methods compared to LMM<sub>jets</sub>. However, the recognition rates of LMM<sub>jets</sub> in previously presented LMM [10] are higher than the recognition rates of LMM<sub>jets</sub> and other improved methods in this work, for some cases. This is due to the fact that in previously presented LMM [10], the RPLWs were optimized using the same images in the test sets *fb* and *dup1*, thus the resulting optimal RPLWs would be fine-tuned to the test sets. However, in this work, optimal RPLWs are obtained using an image set independent of the test sets, thus they would be more generic and reasonable.

TABLE III.  
COMPARISON OF FACE RECOGNITION RATES

Algorithm	Recognition Rate (%)	
	<i>fb</i>	<i>dup1</i>
Fully Automatic Algorithms:		
<b>EBGM</b>	<b>86.3</b>	<b>43.4</b>
EBGM + PCA <sub>EBGM</sub>	86.3	43.4
EBGM <sub>PSO</sub>	92.0	50.0
EBGM <sub>PSO</sub> + PCA <sub>EBGM<sub>PSO</sub></sub>	92.4	50.7
<b>LMM<sub>jets</sub></b>	<b>92.1</b>	<b>51.6</b>
LMM <sub>jets</sub> + PCA <sub>LMM<sub>jets</sub></sub>	92.7 (3)	52.1 (3)
LMM <sub>jets&amp;glps</sub>	92.8 (2)	52.7 (2)
<b>LMM<sub>jets&amp;glps</sub> + PCA<sub>LMM<sub>jets&amp;glps</sub></sub></b>	<b>93.4 (1)</b>	<b>53.2 (1)</b>
Partially Automatic Algorithms:		
<b>USC's EBGM [12]</b>	<b>95.0</b>	<b>59.1</b>
<b>CSU's EBGM [17]</b>	<b>89.8</b>	<b>46.3</b>
EBGM <sub>PSO</sub>	96.1	59.3
EBGM <sub>PSO</sub> + PCA	96.5	60.0
LMM <sub>jets</sub>	96.5	61.5
LMM <sub>jets</sub> + PCA	96.9 (2)	63.3 (1)
LMM <sub>jets&amp;glps</sub>	96.9 (2)	62.0 (3)
<b>LMM<sub>jets&amp;glps</sub> + PCA</b>	<b>97.1 (1)</b>	<b>63.2 (2)</b>

The details on USC's partially automatic EBGM algorithm are insufficient to implement and obtain results for comparison. Therefore, we did not implement it. The recognition rates of partially automatic EBGM shown in the Table III are those of the USC's original EBGM reported in the third FERET test [12], and CSU's EBGM reported by D. Bolme [17]. As the similarity values of USC's EBGM corresponding to the images are not available, we could not hybridize it with PCA. When hybridizing partially automatic EBGM<sub>PSO</sub> and LMM with PCA, the face region was selected from an image using the known eye coordinates. Therefore, the landmark locations found by EBGM<sub>PSO</sub> or LMM were not required to select the face region.

2) *Comparison of Recognition Rates:* The fully automatic algorithms have more practical significance than the partially automatic algorithms, since the manually selected eye coordinates are not available in practice. When comparing the recognition rates of the fully automatic algorithms, it can be seen from Table III that all the proposed extensions in this work, which are the optimization of EBGM, combined usage of Gabor wavelet and GLPs for feature extraction, and the hybridization of Gabor wavelet and eigenface features, have improved the performance.

By optimizing the face graph matching procedure with PSO for locating landmarks, and hybridizing Gabor wavelet features with eigenface features for recognition, recognition rate of fully automatic EBGM was improved by  $(92.0-86.3)/86.3 = 6.3\%$  for *fb* set, and  $(50.0-43.4)/43.4$

$= 15.2\%$  for *dup1* set. By combining Gabor wavelet features and GLPs for locating landmarks, and hybridizing Gabor wavelet features with eigenface features for recognition, recognition rates of fully automatic LMM<sub>jets</sub> were improved by 1.4% and 3.1% for *fb* and *dup1* sets, respectively. The best method in general is LMM<sub>jets&glps</sub> + PCA, and incorporates all three approaches: model matching procedure with PSO, combined usage of Gabor wavelet features and GLPs for locating landmarks, and hybridizing Gabor wavelet features with eigenface features for recognition. It improved the recognition rates of EBGM by 8.2% and 22.6% for *fb* and *dup1* sets, respectively. We could not achieve an improvement in the recognition rates by EBGM + PCA<sub>EBGM</sub>. This would be due to the fact that the landmarks located by EBGM are not highly accurate, resulting in less accurate Gabor wavelet features.

A similar trend in the improvements were achieved by the proposed extensions for partially automatic versions, in most cases. However, the improvements are small in some cases, as it is easier for most of the considered algorithms in Table III to perform well when the eye coordinates are provided. In general, the best three methods are LMM<sub>jets&glps</sub> + PCA, LMM<sub>jets&glps</sub>, and LMM<sub>jets</sub> + PCA, which are introduced in this work.

## VI. SUMMARY AND DISCUSSION

The contributions of this work are,

- 1) optimization of EBGM with PSO,
- 2) combined usage of Gabor wavelets and 2D GLPs for feature extraction,
- 3) hybridization of Gabor wavelet and eigenface features, and
- 4) comparison of experimental results of both fully automatic and partially automatic versions of EBGM<sub>PSO</sub> and LMM with EBGM.

We introduced the concept of 2 previously [9]. In this paper, we have described it in detail, illustrated it more clearly, and improved the results.

The locations of the nodes of an LM in LMM are varied more effectively than those of an FG in EBGM<sub>PSO</sub>, since the architecture of varying the overall geometric structure of an LM causes less distortions to the generic face structure than an FG. This is due to the fact that in LMM, the variations of the node locations are based on their principal components, whereas in EBGM<sub>PSO</sub>, they are only somewhat guided by the patterns of edge vector variations. The recognition rates in Table III reveal that LMM<sub>jets</sub> is slightly better than EBGM<sub>PSO</sub> in most cases. This may imply that more effective variation of the node locations of a deformable model, even with the average of a bunch of jets, is more productive than less effective variations of node locations, with an entire bunch of jets.

In this work, we implemented the PCA the way presented by M. Turk et. al. [13]. PCA has been improved significantly over the past few years, since then. Hybridizing our algorithms with such improved versions of PCA would improve the recognition rates further.

Recognition rates achieved by our own implementation of the original fully automatic EBGM were lower than their reported recognition rates [12]. There is lack of detailed documentation to implement it the same way, as also mentioned by D. Bolme [17]. A detailed report of our proposed methods is available upon request.

#### ACKNOWLEDGMENT

Portions of the research in this paper use the Color FERET database of facial images collected under the FERET program. This project is partially funded by the Australian Research Council.

#### REFERENCES

- [1] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.
  - [2] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," Ruhr-Universität Bochum, Internal Report 96-08, April 1996.
  - [3] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg, "The Bochum/USC face recognition system and how it fared in the FERET phase III test," in *Face Recognition: From Theory to Applications (H Wechsler, P J Phillips, V Bruce, F Fogeman Saulie, and TS Huang)*. Springer Verlag, 1998, pp. 186–205.
  - [4] J. L. Liu and Z. Q. Liu, "EBGM with fuzzy fusion on face recognition," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Artificial Intelligence, vol. 3809. Springer Verlag, Berlin, 2005, pp. 498–509.
  - [5] H. Y. Zhang and H. D. Ma, "Grid-based parallel elastic graph matching face recognition method," in *Proceedings of Advanced Web and Network Technologies, and Applications*, ser. Lecture Notes in Computer Science. Springer Verlag, Berlin, 2006, vol. 3842, pp. 1041–1048.
  - [6] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans, "Face alignment using statistical models and wavelet features," in *Proceedings of the IEEE Computer Society Conf. on Comp. Vision and Pattern Rec.*, vol. 1, 2003, pp. 321–327.
  - [7] J. C. Fan, H. X. Yao, W. Gao, Y. Z. Liu, and X. Liu, "The bunch-active shape model," in *Proceedings of Effective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science, vol. 3784. Springer Verlag, Berlin, 2005, pp. 16–23.
  - [8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
  - [9] R. Senaratne and S. Halgamuge, "Optimised landmark model matching for face recognition," in *7th International Conference on Automatic Face and Gesture Recognition*. IEEE, 2006, pp. 120–125.
  - [10] R. S. Senaratne and S. K. Halgamuge, "Optimal weighting of landmarks for face recognition," *Journal of Multimedia*, vol. 1, no. 3, pp. 31–41, 2006, <http://academypublisher.com/jmm/vol01/no03/>.
  - [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks*, vol. IV. IEEE, 1995, pp. 1942–1948.
  - [12] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
  - [13] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
  - [14] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.
  - [15] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43.
  - [16] X. Wang and H. Xiao, "PSO-based model predictive control for nonlinear processes," in *Advances in Natural Computation*, ser. Lecture Notes in Computer Science, vol. 3611. Springer, Berlin, 2005, pp. 196–203.
  - [17] D. S. Bolme, "Elastic bunch graph matching," Masters Thesis, Department of Computer Science, Colorado State University, 2003.
  - [18] C. Liu, "Gabor-based kernel PCA with fractional power polynomial models for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 572–581, 2004.
  - [19] J. Kittler, M. Hatef, R. Hatef, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
  - [20] G. Fumera and F. Roli, "A theoretical and experimental analysis of linear combiners for multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 942–956, 2005.
  - [21] A. Ross, K. Nandakumar, and A. Jain, "Score level fusion," in *Handbook of Multibiometrics*. Springer, 2006, pp. 91–141.
  - [22] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, p. 21152125, 2003.
  - [23] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.
  - [24] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain, "Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems," *IEEE TPAMI*, vol. 27, no. 3, pp. 450–455, 2005.
  - [25] J. Heo, S. Kong, B. Abidi, and M. Abidi, "Fusion of visual and thermal signatures with eyeglass removal for robust face recognition," in *Proceedings of the Computer Vision and Pattern Recognition Workshop*, 2004, pp. 122–127.
  - [26] O. Nestares, R. Navarro, J. Portilla, and A. Taberero, "Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions," *Journal of Electronic Imaging*, vol. 7, no. 1, pp. 166–173, 1998.
- Rajinda Senaratne** received his B.Sc. degree in Electrical Engineering in 2002, and M.Eng. degree in Electronic and Telecommunication Engineering in 2004 from the University of Moratuwa, Sri Lanka. He received his Ph.D. degree from the Melbourne School of Engineering, The University of Melbourne, Australia, in 2008. His research interests are computer vision, image processing, pattern recognition, and artificial intelligence.
- Saman Halgamuge** received the B.Sc. degree in Electronic and Telecommunication Engineering from University of Moratuwa, Moratuwa, Sri Lanka, in 1984, and the Dipl.-Ing. and Dr.-Ing. degrees in Computer Engineering from Technical University of Darmstadt, Germany, in 1990 and 1995, respectively.
- In 1985, he was an Engineer at the Ceylon Electricity Board, Colombo, Sri Lanka. From 1990 to 1995, he was a Research Associate at Darmstadt University of Technology. From 1996

to 1997, he was with the Institute for Telecommunications Research and the Centre for Sensor Signal and Information Processing as a Lecturer in Computer Systems Engineering at the University of South Australia, Adelaide, Australia. In 1997, he joined the Department of Mechanical Engineering, The University of Melbourne, Melbourne, Australia, as a Senior Lecturer. From 2002-08, he has been an Associate Professor and Reader and from 2008 he has been a Professor in the Department of Mechanical Engineering. He is a coauthor of over 200 journal/conference papers and has contributed to books in the areas of data mining and analysis, bioinformatics, mechatronics, and computational intelligence. He was the Chair of 8 international conferences and Program Committee Member of over 60 conferences. He is an Associate Editor of 4 journals.

**Arthur Hsu** obtained his Bachelor (1999) and PhD (2006) of Engineering from The University of Melbourne, Australia. He has been actively working on areas of Data Mining and Optimization methods since 2000 with application to real-life problems. He worked as a research fellow in data mining at The University of Melbourne between 2006 and 2008. Currently he is a bioinformatics research scientist at The Walter and Eliza Hall Institute of Medical Research and continued his research in data mining and optimization with a strong focus on computational biology applications.