# 3D model-based pose invariant face recognition from multiple views

Q. Chen, J. Yao and W.K. Cham

**Abstract:** A 3D model-based pose invariant face recognition method that can recognise a human face from its multiple views is proposed. First, pose estimation and 3D face model adaptation are achieved by means of a three-layer linear iterative process. Frontal view face images are synthesised using the estimated 3D models and poses. Then the discriminant 'waveletfaces' are extracted from these synthesised frontal view images. Finally, corresponding nearest feature space classifier is implemented. Experimental results show that the proposed method can recognise faces under variable poses with good accuracy.

## 1 Introduction

Face recognition has been an active research area over the past few years and numerous face recognition algorithms have been proposed. Good reviews can be found in [1–5]. Most automatic face recognition (AFR) algorithms are for face recognition under controlled conditions. For example, satisfactory recognition rates on face images which are uncovered, in frontal view, with neutral expression and controlled lighting have been reported in [6–11]. However, when the input face images are not so ideal and have some variations on such conditions, the performance of these AFR algorithms will deteriorate.

The problem related to variations in poses received much attention and many algorithms have been developed to tackle this problem. An early attempt is the 2D appearance based approach that describes faces under varying pose with a set of 2D features and achieves pose analysis (such as pose estimation and pose classifying) and face recognition by comparing these features. Murase and Nayar [12] present a method for pose invariant face recognition in the entire eigenspace. Pentland *et al.* [13] and Huang *et al.* [14] achieved pose invariant face recognition in the viewspace which is a subspace of the eigenspace. Demir's method [15] is similar to that of Pentland *et al.* [18], but employing a sub-linear discriminant analysis (LDA) space as the viewspace. This problem was tackled in the discriminant 'waveletface' space [16] and the kernel LDA space [17]. De Vel and Aeberhard [18] describe a line-based algorithm for pose invariant face recognition. When the gallery has an image having a pose similar to that of the test images, these appearance-based methods have good recognition results. Otherwise, their performance deteriorates. Hence, these methods require dense sampling of the continuous pose. This not only increases the gallery size but also makes the recognition process more time-consuming. Therefore the 3D model-based approach was proposed, which has stronger generalisation to pose invariant face recognition, though its implementation is more complex.

In the 3D model-based approach, a 3D face model is built to represent the 3D geometry of human faces in 2D images. This approach removes the effect of pose variations on face recognition by estimating and aligning poses with a 3D face model and then extracting features under a uniform pose for classification. Generally, pose estimation is the most critical and challenging operation in 3D model based approach. Methods described in [19–24], which may be referred to as single-view-based methods, achieve pose invariant face recognition using only one view image for each candidate in the gallery. In [19, 20], a fixed generic 3D face model was proposed to be used for all candidates. On the basis of this face model, the pose estimation was achieved by using affine transforms. These methods are simple but the pose of a particular face cannot be estimated accurately by using a fixed 3D model. In [23, 24], simple adaptive 3D face models that can adapt to fit a particular person were proposed. The pose estimation and the model adaptation were achieved synchronously using geometrical measurements. As the 3D models used in [27, 28] are simple and rough, these methods cannot achieve accurate pose estimation, especially they cannot estimate seesaw rotation of faces. Recently, Blanz *et al.* [21, 22] built a 3D morphable face model from a large set of real 3D face data for pose invariant face recognition. On the basis of this model, the pose estimation and the model adaptation were achieved by hybrid geometric information and texture information based optimisation. In fact, the 3D geometry and the pose of a particular face can hardly be exactly recovered from only one of its 2D projections. Hence, in [25], rangefinder is used to construct accurate 3D face model for pose invariant face recognition. However, each registered candidate should be captured by rangefinder, the use of which is the limitation of this system as this is expensive equipment. To achieve good performance on pose estimation and model adaptation just from 2D grey-face images, Zhang *et al.* [26] proposed a multiple-view approach which created a 3D face structure and estimated poses from multiple views of a particular face by adapting a generic 3D face model with a cubic explicit polynomial. Pose estimation was achieved by minimising the distance map residual error using the Levenberg–Marquardt optimisation method. Such nonlinear pose

doi:10.1049/iet-cvi:20060014

Paper first received 18th January 2006 and in revised form 15th February 2007

The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong Shatin, New Territories, Hong Kong

E-mail: qrchen@ee.cuhk.edu.hk

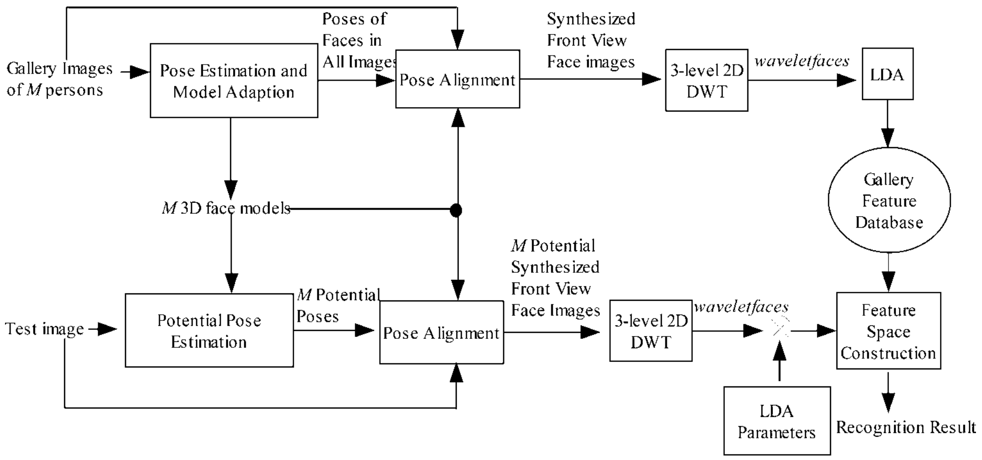
Gallery Images of $M$ persons
Pose Estimation and Model Adaption
Poses of Faces in All Images
Pose Alignment
Synthesized Front View Face images
3-level 2D DWT
waveletfaces
LDA
Gallery Feature Database
$M$ 3D face models
$M$ Potential Synthesized Front View Face Images
Test image
Potential Pose Estimation
$M$ Potential Poses
Pose Alignment
3-level 2D DWT
waveletfaces
Feature Space Construction
LDA Parameters
Recognition Result

**Fig. 1** *Overall method architecture of the face recognition*

estimation algorithm requires large computation cost and is sensitive to the initial values (local minimal problem).

This paper proposes a 3D model based pose invariant face recognition method using the multiple-view approach. The proposed method retains the advantages of the 3D model based approach and overcomes most limitations of previous methods using this approach. The major contribution of the proposed method is that it uses our linear iterative algorithm [27] to achieve accurate, fast and robust pose estimation from multiple views for pose invariant face recognition. In addition, by performing linear discriminant analysis (LDA) on synthesised frontal view images and employing corresponding nearest feature space (CNFS) classifier, our method makes full use of the statistic information to extract disriminant features and achieves robust classification.

## 2 Overview of the proposed method

In this paper, we propose a multiple view 3D model based pose invariant face recognition method that can recognise a face from its multiple views. This method is composed of four steps: (1) pose estimation and 3D model adaptation, (2) pose alignment, (3) feature extraction and (4) classification. The major challenge of this approach is step (1) in which we need to estimate the pose of a face and adapt the 3D model to fit the face. In this paper, we solve this problem by using the inverse projection rays based geometric constraint in a three-layer linear iterative process.

The block diagram of the proposed method is given in Fig. 1. First, the poses of the faces in the test image and gallery images are estimated using a three-layer linear iterative process. This algorithm achieves pose estimation in a linear iterative way like [28–30], but does not require the accurate 3D geometry of the face in the images before estimation. The 3D geometry of a face is obtained by iteratively updating the reference face model to fit a particular person. On the basis of the estimated poses and 3D face models, pose alignment is implemented by synthesising frontal view face images from the test face image and gallery face images. We extract the waveletfaces from these synthesised frontal view images. Finally, linear discriminant analysis is performed on these waveletfaces and the CNFS classifier is used for classification.

## 3 Pose estimation and 3D model adaptation

We adopt our model-based linear pose estimation (*MBLPE*) algorithm described in [27] for pose estimation and model adaptation. In the algorithm, we use a 3D reference face model to represent the 3D geometry of a face. The input is a set of $N_s$ face images of the same candidate with the 2D coordinates of needed facial feature points extracted from these face images. The output consists of the pose of a human face for each input face image and an adapted reference face model for the face in the input images.

The MBLPE algorithm consists of three linear iterative updating processes as shown in Fig. 2. In the innermost layer, we estimate the rotation and translation parameters of each input face image based on a reference face model and a set of facial feature points. In the middle layer, the facial feature points on the reference model are globally updated by three model scaling factors which are estimated using multiple face images with the recovered poses. Then, we locally update the facial feature points on the reference face model in the outermost layer. Finally, the whole reference model can be obtained by model deformation. Before the three-layer iterative process, the rotation matrix for each face image is initially set equal to an identity matrix (diag(1, 1, 1)), and the default 3D generic wireframe face

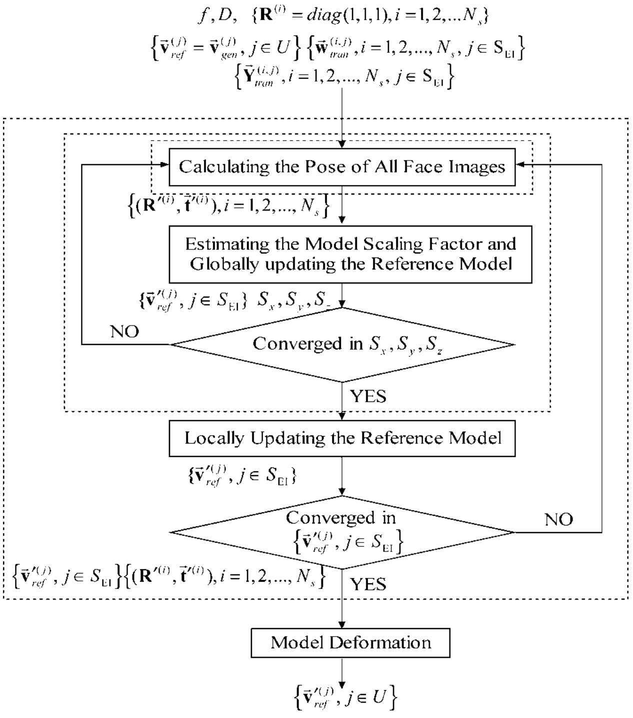
$f, D, \{\mathbf{R}^{(i)} = diag(1,1,1), i = 1,2,...N_s\}$
$\{\vec{\mathbf{v}}^{(j)}_{ref} = \vec{\mathbf{v}}^{(j)}_{gen}, j \in U\}\{\vec{\mathbf{w}}^{(i,j)}_{tran}, i = 1,2,...,N_s, j \in S_{EI}\}$
$\{\vec{\mathbf{Y}}^{(i,j)}_{tran}, i = 1,2,...,N_s, j \in S_{EI}\}$
Calculating the Pose of All Face Images
$\{(\mathbf{R}'^{(i)}, \vec{\mathbf{t}}'^{(i)}), i = 1,2,...,N_s\}$
Estimating the Model Scaling Factor and Globally updating the Reference Model
$\{\vec{\mathbf{v}}'^{(j)}_{ref}, j \in S_{EI}\}$ $S_x, S_y, S_z$
NO
Converged in $S_x, S_y, S_z$
YES
Locally Updating the Reference Model
$\{\vec{\mathbf{v}}'^{(j)}_{ref}, j \in S_{EI}\}$
Converged in $\{\vec{\mathbf{v}}'^{(j)}_{ref}, j \in S_{EI}\}$
NO
$\{\vec{\mathbf{v}}'^{(j)}_{ref}, j \in S_{EI}\}\{(\mathbf{R}'^{(i)}, \vec{\mathbf{t}}'^{(i)}), i = 1,2,...,N_s\}$
YES
Model Deformation
$\{\vec{\mathbf{v}}'^{(j)}_{ref}, j \in U\}$

**Fig. 2** *Flowchart of the MBLPE algorithm*

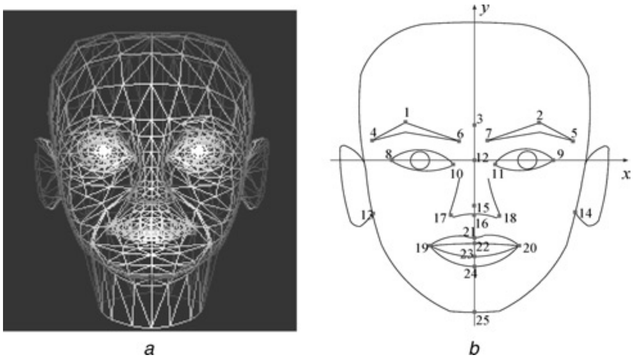

**Fig. 3** *Generic wireframe face model and facial feature points*

*a* Generic wireframe face model
*b* Facial feature points to be extracted in each face

model, which comprises 1039 control points shown in Fig. 3*a*, is used as the initial reference face model. This generic wireframe model was developed by Instituto Superior Tecnico [31].

In this paper, the default generic wireframe model is represented by control points $\{\vec{v}_{\text{gen}}^{(j)} = (x_{v_{\text{gen}}}(j), y_{v_{\text{gen}}}(j), z_{v_{\text{gen}}}(j))^{\text{T}}, j \in U\}$, where $U$ is the set containing integers from 1 to 1039. The 25 points, shown in Fig. 3*b*, are called facial feature points which represent noticeable facial features on a face. We assume that these facial feature points have been extracted, as long as they are not occluded in a face image. If the mouth in an image is closed, facial feature points 1–25 are regarded as forming a rigid body. Otherwise, only facial feature points 1–16 are regarded as forming a rigid body. Only facial feature points on a rigid body $S_{\text{EI}}$ are used in the MBLPE algorithm. Actually, $S_{\text{EI}}$ is a subset of $U$. If the distance between points 21 and 24 is shorter than six times the distance between points 22 and 23, the mouth will be considered as open.

Consider an imaging system as shown in Fig. 4 with the origin of the image plane placed at $(0, 0, f)$ from the perspective center O, where $f$ is the focal length of image plane and $D$ is the distance between the origin O and the origin $\text{o}_v$ of a 3D reference face model. We assume it is a weak perspective projection system, that is, $D >> f$. The reference model is represented by control points $\{\vec{P}_{\text{ref}}^{(j)} = (X_{P_{\text{ref}}}(j), Y_{P_{\text{ref}}}(j), Z_{P_{\text{ref}}}(j))^{\text{T}}, j \in U\}$ or $\{\vec{v}_{\text{ref}}^{(j)} = (x_{v_{\text{ref}}}(j), y_{v_{\text{ref}}}(j), z_{v_{\text{ref}}}(j))^{\text{T}}, j \in U\}$ w.r.t. origin O or origin $\text{o}_v$, respectively. $\{\vec{w}_{\text{ref}}^{(j)} = (x_{w_{\text{ref}}}(j), y_{w_{\text{ref}}}(j))^{\text{T}}, j \in U\}$ represent the projected 2D points on the image plane from $\{\vec{P}_{\text{ref}}^{(j)}, j \in U\}$. Now we consider the rigid transformation of the 3D reference face model by rotating and translating the control points $\{\vec{v}_{\text{ref}}^{(j)}, j \in U\}$, yielding $\{\vec{v}_{\text{tran}}^{(j)} = \boldsymbol{R}\vec{v}_{\text{ref}}^{(j)} + \vec{t}, j \in U\}$, where $\boldsymbol{R}$ and $\vec{t}$ stand for the rotation matrix and the translation vector, respectively. Therefore $\{\vec{v}_{\text{tran}}^{(j)}, j \in U\}$ and corresponding $\{\vec{P}_{\text{tran}}^{(j)} = (X_{P_{\text{tran}}}(j), Y_{P_{\text{tran}}}(j), Z_{P_{\text{tran}}}(j))^{\text{T}}, j \in U\}$ represent the control points on the transformed reference model w.r.t. $\text{o}_v$ and O, respectively. $\{\vec{w}_{\text{tran}}^{(j)} = (x_{w_{\text{tran}}}(j), y_{w_{\text{tran}}}(j))^{\text{T}}, j \in U\}$ are projection of $\{\vec{P}_{\text{tran}}^{(j)}, j \in U\}$ on the image plane (Fig. 4). In a weak perspective projection system, coordinates of $\vec{P}_{\text{tran}}^{(j)}$ and $\vec{w}_{\text{tran}}^{(j)}$ are related by

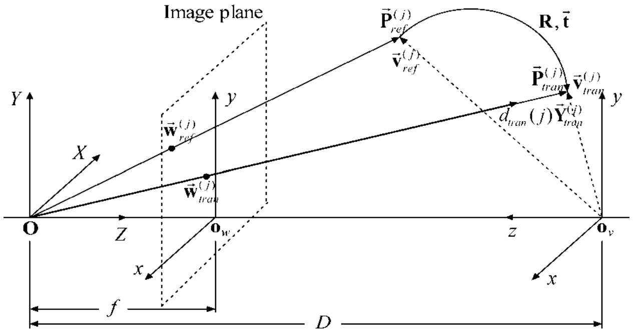

**Fig. 4** *Perspective projection system*

$$x_{w_{\text{tran}}}(j) = -f\frac{X_{P_{\text{tran}}}(j)}{Z_{P_{\text{tran}}}(j)} \simeq -f\frac{X_{P_{\text{tran}}}(j)}{D},$$
$$y_{w_{\text{tran}}}(j) = f\frac{Y_{P_{\text{tran}}}(j)}{Z_{P_{\text{tran}}}(j)} \simeq f\frac{Y_{P_{\text{tran}}}(j)}{D} \tag{1}$$

In this case, $\vec{w}_{\text{tran}}^{(j)}$ tracked in an image will give an inverse projection ray with unit vector $\vec{Y}_{\text{tran}}^{(j)}$ specified by $(x_{w_{\text{tran}}}(j)^2 + y_{w_{\text{tran}}}(j)^2 + f^2)^{-1/2}(x_{w_{\text{tran}}}(j), y_{w_{\text{tran}}}(j), f)^{\text{T}}$. So we can calculate the actual coordinates of the control point $\vec{P}_{\text{tran}}^{(j)}$ on the transformed 3D reference face model as follows

$$\vec{P}_{\text{tran}}^{(j)} = d_{\text{tran}}(j)\vec{Y}_{\text{tran}}^{(j)} \tag{2}$$

where $d_{\text{tran}}(j)$ denotes the depth of $\vec{P}_{\text{tran}}^{(j)}$ from the perspective center O. Here, the depth $d_{\text{tran}}(j)$ is unknown and it would be iteratively updated in the innermost-layer iterative process.

### 3.1 Pose estimation

For convenience, we consider the camera fixed and the head moving. In the innermost-layer iteration, we update the motion of a human head for each input image by computing six absolute motion parameters $\theta_x$, $\theta_y$, $\theta_z$, $t_x$, $t_y$ and $t_z$.

The problem of estimating the pose of the face in an input face image w.r.t. a 3D reference face model can be described as follows. Given a set of 2D facial feature points $\{\vec{w}_{\text{tran}}^{(j)}, j \in S_{\text{EI}}\}$ on the input face image and a set of 3D facial feature points $\{\vec{v}_{\text{ref}}^{(j)}, j \in S_{\text{EI}}\}$ on the reference face model, we seek rotation matrix $\boldsymbol{R}$, translation vector $\vec{t}$ and $\{d_{\text{tran}}(j), j \in S_{\text{EI}}\}$ by minimising the following error function

$$\varepsilon^2(\boldsymbol{R}, \vec{t}, \{d_{\text{tran}}(j)\}) = \sum_{j \in S_{\text{EI}}} \lambda_j \times \left\| \Psi(d_{\text{tran}}(j)\vec{Y}_{\text{tran}}^{(j)}) - \left(\boldsymbol{R}\vec{v}_{\text{ref}}^{(j)} + \vec{t}\right) \right\|^2 \tag{3}$$

where $\{\vec{Y}_{\text{tran}}^{(j)}, j \in S_{\text{EI}}\}$ are unit inverse projective ray vectors of $\{\vec{w}_{\text{tran}}^{(j)}, j \in S_{\text{EI}}\}$ from the perspective center O. The coefficient $\lambda_j$ is determined by the availability of the facial feature point, which corresponds to the $j$th control point, in the input face image. It is 1 if we can successfully extract this facial feature point in the input face image, otherwise it is 0. The function $\Psi(\cdot)$ denotes the coordinate conversion relationship between the coordinates of the point $\vec{P}_{\text{tran}}^{(j)}$ w.r.t. O and the coordinates of the corresponding point $\vec{v}_{\text{tran}}^{(j)}$ w.r.t. $\text{o}_v$, which is given as follows

$$\vec{v}_{\text{tran}}^{(j)} = \Psi\left(\vec{P}_{\text{tran}}^{(j)}\right) = \text{diag}(-1, 1, -1)\left(\vec{P}_{\text{tran}}^{(j)} - [0 \quad 0 \quad D]^{\text{T}}\right) \tag{4}$$

$$\vec{P}_{\text{tran}}^{(j)} = \Psi\left(\vec{v}_{\text{tran}}^{(j)}\right) = \text{diag}(-1, 1, -1)\left(\vec{v}_{\text{tran}}^{(j)} - [0 \quad 0 \quad D]^{\text{T}}\right) \tag{5}$$

The computation requirement of the nonlinear pose estimation algorithm increases quickly as the number of used facial feature points increases [32, 33]. Local minima problem is another limitation of the nonlinear pose estimation algorithm. In order to reduce the computation time and to avoid being trapped at a local minimum solution, the pose estimation is divided into three linear iterative problems that can be solved efficiently [29]. The first stage approximates the global translation vector of the reference model. The second stage updates the depth values $\{d_{\text{tran}}(j), j \in S_{\text{EI}}\}$ of the facial feature points $\{\vec{\boldsymbol{P}}^{(j)}_{\text{tran}}, j \in S_{EI}\}$ on the transformed reference face model using recovered motion parameters. The third stage is to determine the rigid motion parameters by using a least-square minimisation algorithm. The above three stages are repeated in turn until convergent values of $\theta_x, \theta_y, \theta_z, t_x, t_y$ and $t_z$ are obtained.

*3.1.1 Global translation approximation stage:* Given a certain rotation matrix $\boldsymbol{R}$, the optimal value for the global translation vector $\vec{\boldsymbol{t}}$ can be computed in closed form as

$$\vec{\boldsymbol{t}}' = \operatorname{diag}(-1, 1, -1) \sum_{j \in S_{\text{EI}}} \lambda_j \left(\boldsymbol{I} - \vec{\boldsymbol{Y}}^{(j)}_{\text{tran}} \vec{\boldsymbol{Y}}^{(j)\text{T}}_{\text{tran}}\right)^{-1} \times \sum_{j \in S_{\text{EI}}} \lambda_j \left(\vec{\boldsymbol{Y}}^{(j)}_{\text{tran}} \vec{\boldsymbol{Y}}^{(j)\text{T}}_{\text{tran}} - \boldsymbol{I}\right) \Psi\left(\boldsymbol{R} \vec{\boldsymbol{v}}^{(j)}_{ref}\right) \tag{6}$$

where $\boldsymbol{I}$ is an identity matrix.

*3.1.2 Depth approximation stage:* After the motion parameters are approximated in the global translation approximation stage or in the previous pose estimation stage, the deviation between the recovered facial feature points and the true facial feature points becomes small. The facial feature points $\{\vec{\boldsymbol{P}}^{(j)}_{\text{tran}}, j \in S_{\text{EI}}\}$ on the transformed reference model are approximated by the perpendicular intersection of the corresponding facial feature points $\{\Psi(\boldsymbol{R}\vec{\boldsymbol{v}}^{(j)}_{\text{ref}} + \vec{\boldsymbol{t}}'), j \in S_{\text{EI}}\}$. Hence, the depth values of $\{\vec{\boldsymbol{P}}^{(j)}_{\text{tran}}, j \in S_{\text{EI}}\}$ can be updated as follows

$$d'_{\text{tran}}(j) = \vec{\boldsymbol{Y}}^{(j)\text{T}}_{\text{tran}} \Psi\left(\boldsymbol{R}\vec{\boldsymbol{v}}^{(j)}_{ref} + \vec{\boldsymbol{t}}'\right) \tag{7}$$

In this way, the estimated depth values will be gradually moved toward the true depth values.

*3.1.3 Least-square fitting stage:* After the depth values $\{d'_{\text{tran}}(j), j \in S_{\text{EI}}\}$ are determined, the pose represented by $\boldsymbol{R}$ and $\vec{\boldsymbol{t}}$ can be further updated by minimising the following error function

$$\varepsilon^2(\boldsymbol{R}, \vec{\boldsymbol{t}}) = \sum_{j \in S_{\text{EI}}} \lambda_j \left\| \vec{\boldsymbol{v}}'^{(j)}_{\text{tran}} - (\boldsymbol{R}\vec{\boldsymbol{v}}^{(j)}_{ref} + \vec{\boldsymbol{t}}) \right\|^2 \tag{8}$$

where $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{tran}}, j \in S_{\text{EI}}\}$ are the updated estimation of facial feature points on the transformed reference model given by

$$\vec{\boldsymbol{v}}'^{(j)}_{\text{tran}} = \Psi\left(\vec{P}'^{(j)}_{\text{tran}}\right) = \Psi\left(d'_{\text{tran}}(j)\vec{\boldsymbol{Y}}^{(j)}_{\text{tran}}\right) \tag{9}$$

The singular value decomposition method described in [32] is used to solve this least-square minimisation problem and we obtain updated pose estimation $\boldsymbol{R}'$ and $\vec{\boldsymbol{t}}'$.

## 3.2 Globally updating the reference model

Only when the reference model represents the 3D geometry of the face in the input images with good accuracy, reliable pose estimation in the innermost-layer iterative process can be obtained. Thus, we need to adapt the reference model to fit the particular face in the input images. The model adaptation is achieved by globally updating the reference model and locally updating the reference model iteratively. Locally updating the reference model will be discussed in Section 3.3. Global updating of the reference model is implemented by scaling the 3D coordinates of the facial feature points on the reference model with scaling matrix $\boldsymbol{S} = \operatorname{diag}(\text{S}_x, \text{S}_y, \text{S}_z)$, where $S_x$, $S_y$ and $S_z$ are three model scaling factors. We compute these model scaling factors using multiple face images with poses recovered in the above pose estimation process by minimising the following error function

$$\varepsilon^2(\boldsymbol{S}) = \sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij} \left\| \vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \left(\boldsymbol{R}'^{(i)}\boldsymbol{S}\vec{\boldsymbol{v}}^{(j)}_{ref} + \vec{\boldsymbol{t}}'^{(i)}\right) \right\|^2 \tag{10}$$

where $N_s$ is the total number of input face images. $\boldsymbol{R}^{'(i)}$ and $\vec{\boldsymbol{t}}'^{(i)}$ denote the recovered absolute rotation matrix and absolute translation vector of the $i$th input face image, respectively. $\{\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}}, j \in S_{\text{EI}}\}$ calculated using (9) denote the last updated estimation of facial feature points on the transformed reference model which corresponds to pose in the $i$th input face image. The coefficient $\lambda_{ij}$ is similar to $\lambda_j$ in (3) but defined just for the $i$th face image.

By taking the partial derivative of $\varepsilon^2$ in (10) w.r.t. $\text{S}_x$, $\text{S}_y$ and $\text{S}_z$ and then setting them equal to zeros, three scaling factors can be computed as follows

$$\text{S}_x = \frac{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}\left(\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \vec{\boldsymbol{t}}'^{(i)}\right)\vec{\boldsymbol{r}}'^{(i)}_1 x_{v_{\text{ref}}}(j)}{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}(x_{v_{\text{ref}}}(j))^2} \tag{11}$$

$$\text{S}_y = \frac{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}\left(\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \vec{\boldsymbol{t}}'^{(i)}\right)\vec{\boldsymbol{r}}'^{(i)}_2 y_{v_{\text{ref}}}(j)}{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}(y_{v_{\text{ref}}}(j))^2} \tag{12}$$

$$\text{S}_z = \frac{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}\left(\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \vec{\boldsymbol{t}}^{(i)'}\right)\vec{\boldsymbol{r}}'^{(i)}_3 z_{v_{\text{ref}}}(j)}{\sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij}(z_{v_{\text{ref}}}(j))^2} \tag{13}$$

where $\vec{\boldsymbol{r}}'^{(i)}_1$, $\vec{\boldsymbol{r}}'^{(i)}_2$ and $\vec{\boldsymbol{r}}'^{(i)}_3$ denote the first, second and third column vectors of the rotation matrix $\boldsymbol{R}'^{(i)}$, respectively, that is, $\boldsymbol{R}'^{(i)} = [\vec{\boldsymbol{r}}'^{(i)}_1, \vec{\boldsymbol{r}}'^{(i)}_2, \vec{\boldsymbol{r}}'^{(i)}_3]$. We obtain the globally updated facial feature points $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$ by $\{\boldsymbol{S} \cdot \vec{\boldsymbol{v}}^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$.

$\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$ will be used in a new turn of pose estimation process. In the middle-layer iterative process, both the pose estimation process for each face image described in Section 3.1 and the global model updating step are performed in turn until convergent values of $S_x$, $S_y$ and $S_z$ are obtained.

## 3.3 Locally updating the reference face model

We have recovered the model scaling factors for globally updating the reference face model to fit the input face images. However, accurate 3D coordinates of the facial feature points are also very important for realistic modelling of a particular person in input face images. On the basis of the recovered face poses in all input face images, we can locally update the locations of facial feature points $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}})$ on the reference model in order to fit a particular person closely by minimising the following error function

$$\varepsilon^2(\{\vec{\boldsymbol{v}}^{(j)}_{\text{ref}}\}) = \sum_{i=1}^{N_s} \sum_{j \in S_{\text{EI}}} \lambda_{ij} \left\| \vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \left(\boldsymbol{R}'^{(i)}\vec{\boldsymbol{v}}^{(j)}_{\text{ref}} + \vec{\boldsymbol{t}}'^{(i)}\right) \right\|^2 \tag{14}$$

where $N_s$, $\boldsymbol{R}'^{(i)}$, $\vec{\boldsymbol{t}}'^{(i)}$, $\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}}$ and $\lambda_{ij}$ have the same meaning as in (10).

By taking the partial derivative of $\varepsilon^2$ in (15) w.r.t. $\vec{\boldsymbol{v}}^{(j)}_{\text{ref}}$ and then setting it to zero, we locally update $\vec{v}^{(j)}_{\text{ref}}$ as follows

$$\vec{\boldsymbol{v}}^{(j)}_{\text{ref}} = \frac{\sum_{i=1}^{N_s} \lambda_{ij}\left(\boldsymbol{R}'^{(i)}\right)^{\text{T}}\left(\vec{\boldsymbol{v}}'^{(i,j)}_{\text{tran}} - \vec{\boldsymbol{t}}'^{(i)}\right)}{\sum_{i=1}^{N} \lambda_{ij}} \tag{15}$$

In the system, we assume that the human face is a symmetric body. We keep the symmetry by averaging the coordinate values of symmetric points and keeping $x$-coordinate values of the points on the symmetric line still at zero, while $\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}$ are being updated. The locally updated $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$ will be used in a new turn of pose estimation process. In the outmost-layer iterative process, both the pose estimation process for each face image described in Section 3.1, the global model updating step described in Section 3.2 and the local model updating step are performed in turn until convergent values of $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$ are obtained.

On the basis of final set of updated facial feature points $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in S_{\text{EI}}\}$, the other control points on the reference model $\{\vec{\boldsymbol{v}}'^{(j)}_{\text{ref}}, j \in (U - S_{\text{EI}})\}$ can be deformed from the default generic face model by utilising the radial basis function interpolation method [34]. Thus, the final adapted reference model is employed as the estimated face model of the particular person in the input face images, which is represented by control points $\{\vec{\boldsymbol{v}}^{(j)}_{\text{est}}, j \in U\}$. In our method, the MBLPE algorithm will be used for the training images of each candidate in the gallery database. Therefore for each gallery face image, there is an estimated pose, and for each candidate in the gallery, there is an estimated 3D face model. When a test face image is received, the MBLPE algorithm estimates the pose assuming that the test image is an image of the $i$th candidate for $i = 1, \ldots, M$ and $M$ being the number of candidates in the gallery. We estimate the $i$th potential pose of the face in the test image using the innermost-layer iterative process described in Section 3.1. Thus, we obtain $M$ potential poses for the test face image referring to all candidates in the gallery.

Fig. 5 shows some examples to illuminate the performance of the MBLPE algorithm for pose estimation. Figs 5*a* and *c* show four gallery images of a same person with varying poses and the corresponding face model with estimated poses, respectively. The results show that the MBLPE algorithm can successfully estimate the poses and 3D face model from gallery images of different poses. Figs. 5*b* and *d* show a set of test images and their poses estimated based on the face model generated from the set of images in Fig. 5*a*. The result shows that the MBLPE algorithm can obtain a good estimate of poses for these test images. A more detailed evaluation of the MBLPE algorithm will be given in Section 6.

## 4 Pose alignment

In this section, we propose an algorithm to synthesise a frontal view image from a face image of varying pose. First, we adjust the 3D structure and the orientation of the estimated wireframe model to fit the face in the image. Then the adjusted wireframe model is overlaid on the face image. The frontal view face can be synthesised by rotating the 3D wireframe model and performing texture mapping.

Let $\{\vec{\boldsymbol{v}}^{(j)}_{\text{adj}} = (x_{v_{\text{adj}}}(j), y_{v_{\text{adj}}}(j), z_{v_{\text{adj}}}(j))^{\text{T}}, j \in U\}$ represent the control points of the adjusted face model. Hence, the adjustment should obey the two rules: (i) The 2D projection of $\{\vec{\boldsymbol{v}}^{(j)}_{adj}, j \in S_{\text{EI}}\}$ on the image plane should coincide with facial feature points on the face image; (ii) the 3D structure of $\{\vec{\boldsymbol{v}}^{(j)}_{\text{adj}}, j \in U\}$ should be as similar as possible to that of $\{\vec{\boldsymbol{v}}^{(j)}_{v_{\text{nor}}} = (\text{x}_{v_{\text{nor}}}(j), \text{y}_{v_{\text{nor}}}(j), \text{z}_{v_{\text{nor}}}(j))^{\text{T}}, j \in U\}$ which is obtained by normalising and rotating $\{\vec{\boldsymbol{v}}^{(j)}_{\text{est}}, j \in U\}$ as follows

$$\vec{\boldsymbol{v}}^{(j)}_{v_{\text{nor}}} = S_{\text{norm}} \cdot \boldsymbol{R} \cdot \vec{\boldsymbol{v}}^{(j)}_{\text{est}}, \quad \text{where}$$
$$S_{\text{norm}} = \frac{\left\| R_s \cdot \vec{v}^{(l)}_{\text{est}} - \boldsymbol{R}_s \cdot \vec{\boldsymbol{v}}^{(r)}_{\text{est}} \right\|}{\left\| \vec{\boldsymbol{w}}^{(l)}_{\text{tran}} - \vec{\boldsymbol{w}}^{(r)}_{\text{tran}} \right\|} \tag{16}$$

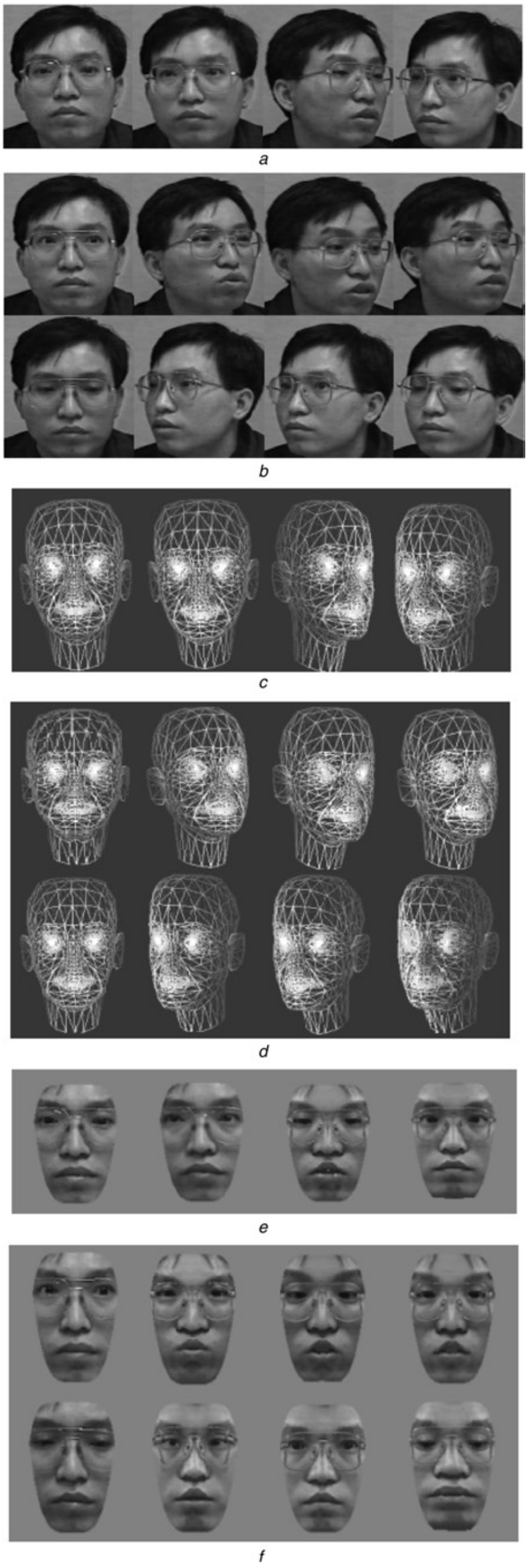

**Fig. 5** *Performance of the proposed algorithms for pose estimation and pose alignment*

*a* Four gallery face images of a same person with different poses
*b* Eight test face images of the same person with different poses
*c* Estimated face model with the estimated poses for gallery images
*d* Face model with the estimated poses for test images
*e* Synthesised frontal view images obtained from gallery images in *a*
*f* Synthesised frontal view images obtained from test images in *b*

Here, $\vec{\boldsymbol{w}}^{(l)}_{\text{tran}}$ denotes the left outside eye corner on the face image if it can be extracted, otherwise $\vec{\boldsymbol{w}}^{(l)}_{\text{tran}}$ denotes the left inside eye corner. $\vec{\boldsymbol{w}}^{(r)}_{\text{tran}}$ has similar definition as $\vec{\boldsymbol{w}}^{(l)}_{\text{tran}}$ but for the right one. $\vec{\boldsymbol{v}}^{(l)}_{\text{est}}$ and $\vec{\boldsymbol{v}}^{(r)}_{\text{est}}$ denote the 3D coordinates of the corresponding eye corners in the estimated 3D face model and $\boldsymbol{R}_{\text{s}}$ is a 2 × 3 matrix, which is the first two rows of matrix $\boldsymbol{R}$.

Therefore, we achieve model adjustment as follows

$$\begin{aligned}
&\forall j \in S_{\text{EI}} \cap \lambda_j = 1{:}x_{v_{\text{adj}}}(j) = x_{w_{\text{tran}}}(j),\\
&\text{y}_{v_{\text{adj}}}(j) = y_{w_{\text{tran}}}(j),\ \text{z}_{v_{\text{adj}}}(j) = \text{z}_{v_{\text{nor}}}(j) \qquad (17)
\end{aligned}$$

$$\begin{aligned}
&\forall j \in U - S_{\text{EI}} \cup \lambda_j = 0{:}\\
&x_{v_{\text{adj}}}(j) = \frac{\sum_{k\in S_{\text{EI}}} \Theta\left(\left|x_{v_{\text{nor}}}(j) - x_{v_{\text{nor}}}(k)\right|\right) \cdot \left(x_{w_{\text{tran}}}(k) + (x_{v_{\text{nor}}}(j) - x_{v_{\text{nor}}}(k))\right)}{\sum_{k\in S_{\text{EI}}} \Theta\left|\left(x_{v_{\text{nor}}}(j) - x_{v_{\text{nor}}}(k)\right|\right)}\\
&y_{v_{\text{adj}}}(j) = \frac{\sum_{k\in S_{\text{EI}}} \Theta\left(\left|y_{v_{\text{nor}}}(j) - y_{v_{\text{nor}}}(k)\right|\right) \cdot \left((y_{w_{\text{tran}}}(k) + \left(y_{v_{\text{nor}}}(j) - y_{v_{\text{nor}}}(k)\right)\right)}{\sum_{k\in S_{\text{EI}}} \Theta\left(\left|y_{v_{\text{nor}}}(j) - y_{v_{\text{nor}}}(k)\right|\right)}\\
&\Theta(r) = \text{e}^{-(r/\sigma_e)^2}\\
&z_{v_{\text{adj}}}(j) = z_{v_{\text{nor}}}(j) \qquad (18)
\end{aligned}$$

where $\{\vec{w}_{\text{tran}}(x_{w_{\text{tran}}}(j), y_{w_{\text{tran}}}(j)), j \in S_{\text{EI}}\}$ represent the facial feature points on the face image and $\sigma_e$ is a constant. Then we synthesise the frontal view by two steps. The first step is to transform the adjusted 3D model $\{\vec{\boldsymbol{v}}^{(j)}_{\text{adj}}, j \in U\}$ into frontal view w.r.t. the image plane and the second step is the triangulation texture mapping [35].

When the face in an image deviates significantly from the frontal view, part of the face will be occluded. With our assumption that human faces are of bilateral symmetry, whenever there is a rotation around $y$-axis, we can reconstruct the occluded part according to the bilateral symmetry of human faces. In our method, if $|\theta_y| > 15°$, the frontal view face image is synthesised using symmetry information to compensate the missing information, otherwise symmetry information will not be used.

The pose alignment operations for gallery images and test images are the same. Figs. 5*e* and *f* give the synthesised frontal view face images for the gallery images shown in Fig. 5*a* and for the test images shown in Fig. 5*b*, respectively. The results indicate that good synthesised frontal images can be obtained by using the proposed pose alignment algorithm.

## 5 Discriminant waveletface analysis and CNFS classifier

For synthesised frontal view images, many existing face recognition algorithms such as those in [7–11, 16, 17] can be applied. In this section, we adopt discrimant waveletface analysis and CNFS classifier for feature extraction and classification similar to Chien's method [16].

First, 2D discrete wavelet transform is used to decompose a synthesised frontal view image into subimages via the high-pass and low-pass filtering. In our method, the three-level lowest frequency sub-image is extracted as the waveletface represented by vector $\boldsymbol{y}$. In general, low frequency components are the most informative subimages gearing with the highest discriminating power. Then we apply LDA to convert the vector $\boldsymbol{y}$ into a new discriminant feature vector $z = (\boldsymbol{W}_{\text{lda}} \cdot \boldsymbol{y})$. The optimal transform matrix $\boldsymbol{W}_{\text{lda}}$ is estimated by maximising the ratio of the determinants of between-class scatter matrix $\boldsymbol{S}_{\text{b}}$ and within-class scatter matrix $\boldsymbol{S}_{\text{w}}$ of the transformed training samples, i.e. $|\boldsymbol{W}^{\text{T}}\boldsymbol{S}_{\text{b}}\boldsymbol{W}|/|\boldsymbol{W}^{\text{T}}\boldsymbol{S}_{\text{W}}\boldsymbol{W}|$. LDA can pull apart the centroid of different classes and reduce the degree of data scattering within the same class. In our method, the discriminant feature vector $\mathbf{z}$ is used for classification.

The NFS classifier is more robust to facial variations than traditional nearest neighbour and nearest feature line classifiers [16]. In our method, we adapt NFS classifier to a CNFS classifier which can achieve classification for a test image with a set of $M$ potential discriminant feature vectors. Let $\{\boldsymbol{z}^{(c,1)}, \boldsymbol{z}^{(c,2)}, \ldots, \boldsymbol{z}^{(c,\,N_s(c))}\}$ denote the independent discriminant feature vectors associated with class $c$ and $N_s(c)$ be the number of gallery images of the $c$th candidate. $\{\boldsymbol{z}^{(c,1)}, \boldsymbol{z}^{(c,2)}, \ldots, \boldsymbol{z}^{(c,\,N_s(c))}\}$ span a feature space $S^{(c)}_{\text{gal}} = \text{sp}\{\boldsymbol{z}^{(c,1)}, \boldsymbol{z}^{(c,2)}, \ldots, \boldsymbol{z}^{(c,\,N_s(c))}\}$. A test image has $M$ potential discriminant feature vectors $z^{(i)}_{\text{test}}$, $i = 1, \ldots, M$ corresponding to its $M$ potential synthesised frontal view images. Let $\boldsymbol{P}^{(i)}_{\text{proj}}$ be the projection of the $i$th potential discriminant feature vector $z^{(i)}_{\text{test}}$ on the feature space $\boldsymbol{S}^{(i)}_{\text{gal}}$. The distance between the potential discriminant feature $z^{(i)}_{\text{test}}$ and the gallery feature space $\boldsymbol{S}^{(i)}_{\text{gal}}$ is calculated by $d(z^{(i)}_{\text{test}}, \boldsymbol{S}^{(i)}_{\text{gal}}) = \|\boldsymbol{z}^{(i)}_{\text{test}} - \boldsymbol{P}^{(i)}_{\text{proj}}\|$. On the basis of Gram-Schmidt process, a set of orthogonal basis $\{\boldsymbol{u}^{(i,1)}, \boldsymbol{u}^{(i,2)}, \ldots, \boldsymbol{u}^{(i,\,N_s(c))}\}$ on the feature space $\boldsymbol{S}^{(i)}_{\text{gal}}$ can be calculated, and the projection $\boldsymbol{P}^{(i)}_{\text{proj}}$ is determined by

$$\begin{aligned}
\boldsymbol{P}^{(i)}_{\text{proj}} = {} & \frac{z^{(i)}_{\text{test}} \cdot \boldsymbol{u}^{(i,1)}}{\boldsymbol{u}^{(i,1)} \cdot \boldsymbol{u}^{(i,1)}} + \frac{z^{(i)}_{\text{test}} \cdot \boldsymbol{u}^{(i,2)}}{\boldsymbol{u}^{(i,2)} \cdot \boldsymbol{u}^{(i,2)}} + \cdots\\
& + \frac{z^{(i)}_{test} \cdot \boldsymbol{u}^{(i,\,N_s(i))}}{\boldsymbol{u}^{(i,\,N_s(i))} \cdot \boldsymbol{u}^{(i,\,N_s(i))}} \qquad (19)
\end{aligned}$$

Thus, for a test image, with all distances $\{d(\boldsymbol{z}^{(c)}_{\text{test}}, \boldsymbol{S}^{(c)}_{\text{gal}}), c = 1, 2 \ldots, M\}$, the classification result $\widehat{c}$ is obtained by

$$\begin{aligned}
\text{d}\left(z^{(\widehat{c})}_{\text{test}}, \boldsymbol{S}^{(\widehat{c})}_{gal}\right) &= \min_{1\le c\le M} \text{d}\left(z^{(c)}_{\text{test}}, \boldsymbol{S}^{(c)}_{\text{gal}}\right)\\
&= \min_{1\le c\le M} \left\| z^{(c)}_{test} - \boldsymbol{P}^{(c)}_{\text{proj}} \right\| \qquad (20)
\end{aligned}$$

## 6 Experimental results

Experimental results on real data for pose estimation and frontal view image synthesis are shown in Fig. 5. In this section, synthetic data were utilised to test the MBLPE algorithm in the proposed face recognition method to achieve pose estimation and model adaptation from multiple views. Then the overall performance of the proposed face recognition method was evaluated using two real face image databases.

### *6.1 Testing the MBLPE algorithm using synthetic data*

In these experiments, the width and the height of the generic head model are 170 and 270 pixels, respectively. We set the focal length of lens $f = 10$ pixels and the distance between the origin O of the camera coordinate system and the origin $\text{o}_v$ of the 3D face models $D = 2000$. The projection system used here is assumed to be a weak perspective projection system. In addition, we assume that 3D face models will be projected on an image plane of size 2 × 2 and then digitised to a screen resolution of 256 × 256. As a result, the width and the height of the observed head models on the screen plane are about 109 and 173, respectively.

In our experiments, a 3D synthetic face model for a particular person was generated by scaling the default generic wireframe model (Fig. 3) with the model scaling factors $\boldsymbol{S} = (S_x, S_y, S_z)$ and then adding Gaussian random noise on 3D coordinates of facial feature points on this scaled model. The standard variance of this Gaussian random noise is denoted as $\sigma_{3d}$. It was introduced in a symmetric way and the 3D facial feature points of a synthetic face model are described as $\{\vec{\boldsymbol{v}}_{\text{syn}}^{(j)} = (x_{v_{\text{syn}}}(j), x_{v_{\text{syn}}}(j), x_{v_{\text{syn}}}(j))^{\text{T}}, j \in S_{\text{EI}}\}$. Then the synthetic face model was rotated and translated with rotation parameter sets $(\theta_x(i), \theta_y(i), \theta_z(i))$, $i = 1, \ldots, N_s$ and translation vectors $(t_x(i), t_y(i), t_z(i))^{\text{T}}$, $i = 1, \ldots, N_s$, where $N_s$ denotes the total number of the face images for a particular person. The facial feature points on the transformed synthetic faces model were projected to the 2D image plane. In order to simulate the facial feature extraction error, Gaussian random noise with fixed standard variance $\sigma_{2d} = 3$ was added on the $x$- and $y$-coordinates of the projected facial feature points in the following experiments except in those to test the performance using different 2D noise. We utilised the MBLPE algorithm to recover the rotation parameter sets $(\theta_x'(i), \theta_y'(i), \theta_z'(i))$, $i = 1, \ldots, N_s$, the translation vectors $(t_x'(i), t_y'(i), t_z'(i))^{\text{T}}$, $i = 1, \ldots, N_s$, and the 3D structure of facial feature points on the synthetic face model $\{\vec{\boldsymbol{v}}_{\text{est}}^{(j)}, j \in S_{\text{EI}}\}$. We utilise the absolute pose error defined as $\Delta\theta_{\text{err}} = (|\theta_x'(i) - \theta_x(i)| + |\theta_y'(i) - \theta_y(i)| + |\theta_z'(i) - \theta_z(i)|)/3$ to measure the accuracy of the recovered pose. We did not test the performance of the recovered translation vectors since they have nothing to do with the following steps in our proposed face recognition method.

First, we investigated the performance of the pose estimation using different $N_s$ which corresponds to the number of input face images. We randomly generated 100 3D synthetic faces with $S_x = 1$, $S_y = 0.9$, $S_z = 1.1$ and $\sigma_{3d} = 3$. Each synthetic face was rotated and translated with $N_s$ uniformly distributed random rotation parameter sets $(\theta_x(i), \theta_y(i), \theta_z(i) \in [-50°, +50°])$, $i = 1, \ldots, N_s$, and $N_s$ uniformly distributed random translation vectors $(t_x(i), t_y(i), t_z(i) \in [-100, +100])^{\text{T}}$, $i = 1, \ldots, N_s$. From the 2D projected faces of these face models, the MBLPE algorithm recovered the pose parameter sets for each synthetic face. The average absolute pose error for all the projected face images $\Delta\bar{\theta}_{\text{err}}$ was calculated. We did the same experiments on other two sets of 100 synthetic faces generated with parameters ($S_x = 1$, $S_y = 0.85$, $S_z = 1.15$, $\sigma_{3d} = 5$) and ($S_x = 1$, $S_y = 0.8$, $S_z = 1.2$, $\sigma_{3d} = 7$) to test the pose estimation performance of the synthetic face model with different extent of deviation from the default generic face model. The results are shown in Fig. 6*a* from which we find that the larger the $N_s$ is, the more accurate the recovered poses are. Especially the performance of $N_s \geq 2$ is much better than that of $N_s = 1$. It means that the MBLPE algorithm is more appropriate for pose estimation from multiple views than from a single view. However, the performance does not improve significantly when the number of input images $N_s$ is larger than 4.

Model adaptation is also important in the MBLPE algorithm. The performance of model adaptation was tested in the following experiment. For given scaling factors $S_x$, $S_y$ and $S_z$ and a given $\sigma_{3d}$, we randomly generated 100 3D synthetic faces and each synthetic face was rotated and translated with four uniformly distributed random rotation parameter sets $(\theta_x(i), \theta_y(i), \theta_z(i) \in [-50°, +50°])$, $i = 1, \ldots, 4$, and four uniformly distributed random translation vectors $(t_x(i), t_y(i), t_z(i) \in [-100, +100])$, $i = 1, \ldots, 4$. Then we projected these transformed synthetic face models on the 2D image plane and utilised these projected face images to recover the poses and the 3D structure of facial feature points on the synthetic face models. In the MBLPE algorithm, the absolute size of the synthesised model cannot be estimated because the size of the synthesised model is related to the translation vector in a week perspective projection system. Thus, each set of recovered facial feature points $\{\vec{\boldsymbol{v}}_{\text{est}}^{(j)}, j \in S_{\text{EI}}\}$ in 3D space is just a scaled estimation of the facial feature points $\{\vec{\boldsymbol{v}}_{\text{syn}}^{(j)}, j \in S_{\text{EI}}\}$ on corresponding synthetic face model with an unknown scaling factor. We develop a normalised distance

$$D_{\text{norm}}\langle \boldsymbol{v}_{\text{obj}}, \boldsymbol{v}_{\text{src}} \rangle = \min_{s_{opt}} \frac{\sum_{j \in S_{EI}} \left\| \vec{\boldsymbol{v}}_{\text{obj}}^{(j)} - s_{\text{opt}} \cdot \vec{\boldsymbol{v}}_{\text{src}}^{(j)} \right\|}{\sum_{j \in S_{\text{EI}}} \left\| \vec{\boldsymbol{v}}_{\text{obj}}^{(j)} \right\|} \qquad (21)$$

to measure the similarity between the objective facial feature points set $\boldsymbol{v}_{\text{obj}} = \{\vec{\boldsymbol{v}}_{\text{obj}}^{(j)}, j \in S_{\text{EI}}\}$ and the source facial feature points set $\boldsymbol{v}_{\text{src}} = \{\vec{\boldsymbol{v}}_{\text{src}}^{(j)}, j \in S_{\text{EI}}\}$ by ignoring the scaling difference. For all the 100 synthetic face

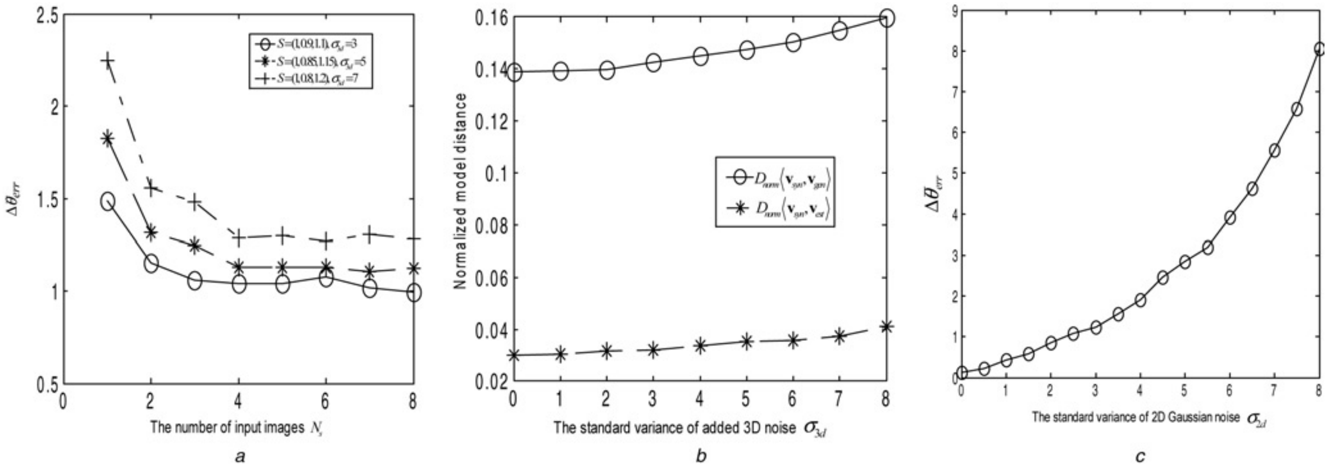

**Fig. 6** *Results of the pose estimation performance*

*a* Average absolute errors of the recovered poses using different number of input images with different 3D noise on 3D coordinates of feature points on the scaled face models with different scaling factors

*b* Average normalised model distances $D_{\text{norm}}\langle \boldsymbol{v}_{\text{syn}}, \boldsymbol{v}_{\text{est}} \rangle$ and $D_{\text{norm}}\langle \boldsymbol{v}_{\text{syn}}, \boldsymbol{v}_{\text{gen}} \rangle$ under different 3D noise on 3D coordinates of feature points on scaled face models

*c* Average absolute errors of the recovered poses of the test images under different 2D noise on measurement (pixel) of 2D projected feature points models, we calculated the average normalised distances $\boldsymbol{D}_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{est}}\rangle$ and $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{gen}}\rangle$, where $\boldsymbol{v}_{\mathrm{syn}} = \{\vec{\boldsymbol{v}}^{(j)}_{\mathrm{syn}}, j \in S_{\mathrm{EI}}\}$, $\boldsymbol{v}_{\mathrm{est}} = \{\vec{\boldsymbol{v}}^{(j)}_{\mathrm{est}}, j \in S_{\mathrm{EI}}\}$ and $\boldsymbol{v}_{\mathrm{gen}} = \{\vec{\boldsymbol{v}}^{(j)}_{\mathrm{gen}}, j \in S_{\mathrm{EI}}\}$ which represent the facial feature points on the default generic model. Two curves of $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{est}}\rangle$ and $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{gen}}\rangle$ with $S_x = 1$, $S_y = 0.8$, $S_z = 1.2$ and different $\sigma_{3d}$ are shown in Fig. 6*b* from which we observe that $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{est}}\rangle$ is evidently smaller than $D_{\mathrm{norm}}\langle v_{\mathrm{syn}}, v_{\mathrm{gen}}\rangle$ at the same $\sigma_{3d}$. It indicates that in average the estimated models are closer to the synthesised faces than the default generic model. Also it can be observed that the larger the $\sigma_{3d}$ is, the larger the distance between $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{est}}\rangle$ and $D_{\mathrm{norm}}\langle \boldsymbol{v}_{\mathrm{syn}}, \boldsymbol{v}_{\mathrm{gen}}\rangle$ is. It means that the larger the dissimilarity of structure of facial feature points between the synthesised and the default generic model is, the more obvious the effect of model adaptation is.

The preceding experiments were carried out to examine the performance of the pose estimation and model adaptation algorithms for gallery images. We also examined the performance of the pose estimation algorithm for a test image under the condition that the 3D model used in the pose estimation process for the test face image belongs to the same candidate. One hundred synthetic face models were randomly generated with $S_x = 1$ and uniformly distributed $S_y \in [0.8, 1.2]$, $S_z \in [0.8, 1.2]$ and $\sigma_{3d} \in [0,8]$. Each synthetic face model is rotated and translated with five uniformly distributed random rotation parameter sets $(\theta_x(i) \in [-30°, 30°], \theta_y(i) \in [-40°, 40°], \theta_z(i) \in [-20°, 20°])$, $i = 1, \ldots, 5$, and five uniform translation vectors $(t_x(i), t_y(i), t_z(i) \in [-100,100])^{\mathrm{T}}$, $i = 1, \ldots, 5$. We use the face images projected from the first four transformed synthetic face model as gallery data, and the face image projected from the fifth transformed synthetic face model is regarded as test data. From the gallery data, each synthetic face model can be estimated using the MBLPE algorithm. Then we recovered the pose of the test images based on these estimated face models. Finally, the average absolute pose error $\Delta\bar{\theta}_{\mathrm{err}}$ for these test images is calculated. The performance of the recovered test images with different $\sigma_{2d}$ is shown in Fig. 6(c) from which we observe that $\Delta\bar{\theta}_{\mathrm{err}}$ exponentially increases with the increase of $\sigma_{2d}$. Although the MBLPE is affected by 2D noise, fortunately, the absolute pose error is smaller than 4° when $\sigma_{2d} \leq 6$. Hence, pose with good accuracy can be obtained when the 2D noise is not too high.

**Fig. 7** *Examples of face images in ORL face image database*

### 6.2 Overall performance of the face recognition method

IIS face image database (accessible at http://smart.iis.sinica.edu.tw/html/download.html) and ORL face image database (accessible at http://www.cam-orl.co.uk/facedatabase.html) were used to evaluate the overall performance of the proposed face recognition method. There are 100 persons and 30 pictures for each person in IIS database (Fig. 5). For the 30 pictures of the same candidate, 10 are in nearly frontal view, 10 are in left-side view and 10 are in right-side view. ORL database include 40 persons and each person has 10 pictures with pose variations as shown in Fig. 7. We manually marked the facial feature points as shown in Fig. 3*b* on each image in these two databases. These marked facial feature points were used in the following experiments. Actually, extracting facial feature points automatically for pose estimation and face recognition should be a possible future work of the proposed approach.

First, we did experiments on IIS database. For each candidate, 4 images were randomly selected to construct the gallery database and the remaining 26 images were used as test images. In the training process, first we performed pose estimation for each image in the gallery database and obtained an estimated 3D model for each candidate by the MBLPE algorithm. On the basis of the estimated pose and the corresponding estimated model, a frontal view face image (size of 112 × 168 as shown in Fig. 5) was synthesised for each gallery image. Through three-level Harr DWT, we obtained the lowest frequency subimage (size of 14 × 21) of each synthesised frontal view image. LDA was then performed on these waveletfaces. We obtained the linear transformation matrix and a discriminant feature vector for each gallery sample. The dimension of the

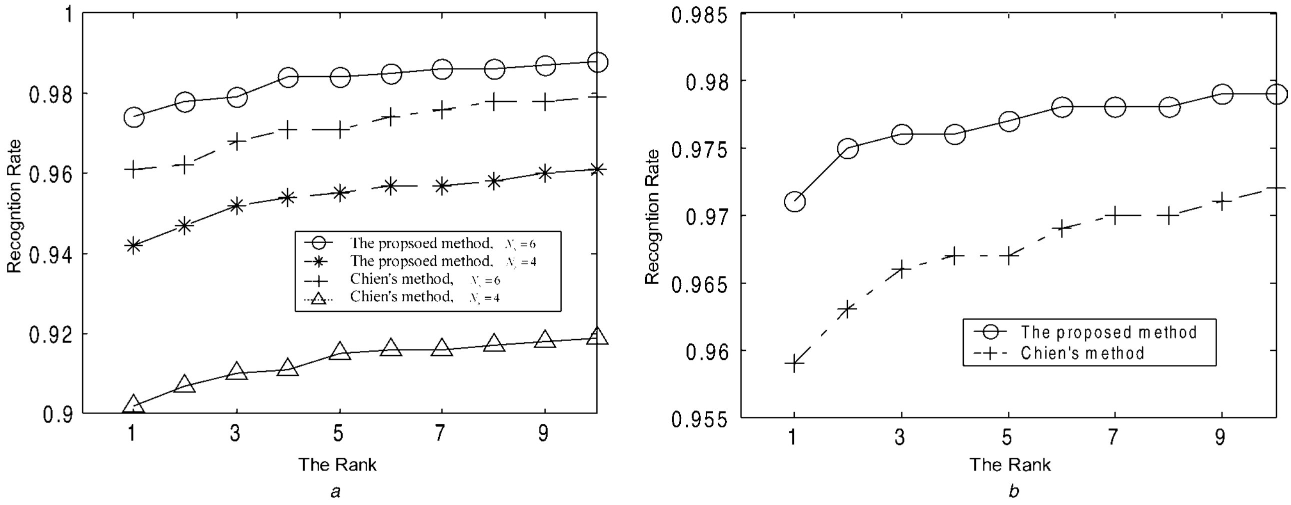

**Fig. 8** *Face recognition results*

*a* Results on IIS database of the proposed method and Chien's method using four and six training images
*b* Results on ORL database of the proposed method and Chien's method using five training images

discriminant feature vector in our experiments was set as 60. These feature vectors of each candidate span a feature space. For a test image, we obtained 100 potential pose estimations, 100 corresponding potential synthesised frontal view images, 100 corresponding potential waveletfaces and 100 corresponding potential discriminant feature vectors, since there are 100 candidates in the gallery database. The distance between each potential feature vector and its corresponding feature space, which is spanned by the corresponding feature vectors in the gallery, is calculated. The candidate corresponding to the shortest distance is the first rank recognition result. The candidate corresponding to the second shortest distance is the second-rank recognition result and so on. We developed a program using Visual C++ 6.0 to do these experiments on a P4-1.8 GHz PC. The offline computation of pose estimation and model adaptation for a person with four gallery images generally spends 0.7 s. In the case of 100 candidates in the gallery, the recognition time (not including feature points extraction) is not more than 1 s for a test face.

In order to ensure the statistical robustness of the recognition performance, we performed ten random rounds of face recognition for each situation and all recognition rates were determined by averaging the ten rounds of face recognition. Furthermore, we repeated the preceding experiment using 6 gallery images (the remaining 24 images for test) per candidate. For comparison, we also conducted the experiment for Chien's face recognition method [16] using the same gallery databases and test sets. Fig. 8*a* shows that our method achieves first-rank recognition rate as high as 94.2% for randomly formed gallery that consists of four images per candidate, whereas Chien's method can just reach 89.3% under the same condition.

We also evaluated the recognition rates of the proposed method using ORL face database. For each person, five images were randomly selected to construct the gallery database and the remaining five images were used as test images. We also performed ten random rounds of face recognition and the recognition rates were determined by averaging the ten rounds of face recognition. For comparison, the same gallery databases and test sets were used to evaluate the performance of Chien's face recognition method. The recognition results shown in Fig. 8*b* verifies the better performance of the proposed method compared to Chien's method.

## 7 Conclusions

In this paper, we propose a 3D model based face recognition method which can recognise a human face under variable pose from its multiple views. Using the MBLPE algorithm, we estimate the poses of faces of each person in the gallery and adapt a 3D face model from all faces of this person. The experimental results show that the MBLPE algorithm can achieve efficient 3D model adaptation from multiple faces of a person and estimate poses of these faces with good accuracy. For each face image in the gallery, its frontal view face image is synthesised using the adapted model and estimated pose. Given a test face image, a pose is estimated and a frontal view face image is synthesised using the 3D model for each person in the gallery. Then LDA is performed on these waveletfaces of these synthesised frontal view face images and classification is achieved using CNFS classifier. Experimental results show that the proposed method has a good performance for pose invariant face recognition.

## 8 References

1 Chellappa, R., Willson, C., and Sirohey, S.: 'Human and machine recognition of faces: a survey', *Proc. IEEE*, 1995, **83**, (5), pp. 705–740

2 Fromherz, T.: 'Face recognition: a summary of 1995–1997'. Int. Computer Science Inst. ICSI TR-98-027, University of California, Berkeley, 1998

3 Zhang, J., Yan, Y., and Lades, M.: 'Face recognition: eigenface, elastic matching, and neural nets', *Proc. IEEE*, 1997, **85**, (9), pp. 1423–1435

4 Pentland, A.: 'Looking at people: sensing for ubiquitous and wearable computing', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (1), pp. 107–119

5 Grudin, A.M.: 'On internal representations in face recognition systems', *Pattern Recognit.*, 2000, **33**, (7), pp. 1161–1177

6 Chen, C.W., and Huang, J.S.: 'Human face profile recognition from a single front view', *Int. J. Pattern Recognit. Artif. Intell.*, 1992, **6**, (4), pp. 571–593

7 Brunelli, R., and Poggio, T.: 'Face recognition: features vs. templates', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1993, **15**, (10), pp. 1042–1053

8 Belhumeur, P.N., Hespanha, J.P., and Kriegman, D.J.: 'Eigenfaces vs. fisherfaces: recognition using class specific linear projection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997, **19**, (7), pp. 711–720

9 Lin, S.H., Kung, S.Y., and Lin, L.J.: 'Face recognition/detection by probabilistic decision-based neural network', *IEEE Trans. Neural Netw.*, 1997, **8**, (1), pp. 114–132

10 Li, S., and Lu, J.: 'Face recognition using nearest feature line', *IEEE Trans. Neural Netw.*, 1999, **10**, (2), pp. 439–443

11 Bartlett, M.S., Movellan, J.R., and Sejnowski, T.J.: 'Face recognition by independent component analysis', *IEEE Trans. Neural Netw.*, 2002, **13**, (6), pp. 1450–1464

12 Murase, H., and Nayar, S.K.: 'Learning and recognition of 3D objects from appearance'. IEEE 2nd Qualitative Vision Workshop, New York, NY, June 1993

13 Pentland, A., Moghaddam, B., and Starner, T.: 'View-based modular eigenspaces for face recognition'. Proc. CVPR 1994, June 1994, pp. 84–91

14 Huang, F.J., Zhou, Z.H., Zhang, H.J., and Chen, T.: 'Pose invariant face recognition'. Proc. 4th IEEE Int. Conf. Automatic Face and Gesture Recognition, March 2000, pp. 245–250

15 Demir, E., Akarun, L., and Alpaydin, E.: 'Two-stage approach for pose invariant face recognition'. Proc. ICASSP2000, June 2000, vol. 6, pp. 5–9

16 Chien, J.-T., and Wu, C.-C.: 'Discriminant waveletfaces and nearest feature classifiers for face recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (12), pp. 1644–1649

17 Lu, J.W., Plataniotis, K.N., and Venetsanopoulos, A.N.: 'Face recognition using kernel direct discriminant analysis algorithms', *IEEE Trans. Neural Netw.*, 2003, **14**, (1), pp. 117–126

18 De Vel, O., and Aeberhard, S.: 'Line-based face recognition under varying pose', *IEEE Trans. Pattern Anal. and Mach. Intell.*, 1999, **21**, (10), pp. 1081–1088

19 Feng, G.C., and Yuen, P.C.: 'Recognition of head-and-shoulder face image using virtual frontal-view image', *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, 2000, **30**, (6), pp. 871–882

20 Chen, Q., Wu, H.Y., Shioyama, S., and Shimada, T.: 'Head pose estimation using both color and feature information'. Proc. 15th ICPR, September 2000, vol. 2, pp. 842–845

21 Blanz, V., Romdhani, S., and Vetter, T.: 'Face identification across different poses and illuminations with a 3D morphable model'. Proc. IEEE 5th Int. Conf. Automatic Face and Gesture Recognition, 20–21 May 2002, pp. 100–105

22 Blanz, V., and Vetter, T.: 'Face recognition based on fitting a 3D morphable model', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (19), pp. 1063–1074

23 Lam, K.M., and Yan, H.: 'An analytic-to-holistic approach for face recognition based on a single frontal view', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998, **20**, (7), pp. 673–686

24 Gao, Y., Leung, M.K.H., Wang, W., and Hui, S.C.: 'Fast face identification under varying pose from a single 2-D model view', *IEE Proc., Vis. Image Signal Process.*, 2001, **148**, (4), pp. 248–253

25 Ishiyama, R., Hamanaka, M., and Sakamoto, S.: 'An appearance model constructed on 3-D surface for robust face recognition against pose and illumination variations', *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 2005, **35**, (3), pp. 326–334

26 Zhang, C.Z., and Cohen, F.S.: '3-D face structure extraction and recognition from images using 3-D morphing and distance mapping', *IEEE Trans. Image Process.*, 2002, **11**, (11), pp. 1249–1259

27 Yao, J., and Cham, W.K.: ‘Efficient model-based linear head motion recovery from movies’. Proc. CVPR2004, July 2004, vol. 2, pp. 414–421

28 Or, S.H., Luk, W.S., Wong, K.H., and King, I.: ‘An efficient iterative pose estimation algorithm’, *Image Vis. Comput.*, 1998, **16**, pp. 353–362

29 Lu, C.P., Hager, G.D., and Mjolsness, E.: ‘Fast and globally convergent pose estimation from video images’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (6), pp. 610–622

30 Ansar, A., and Daniilidis, K.: ‘Linear pose estimation from points or lines’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (5), pp. 578–589

31 Parke, F.I.: ‘Parameterized models for facial animation’, *IEEE Comput. Graph.*, 1982, **2**, (9), pp. 61–68

32 Haralick, R.M., Joo, H., Lee, C.N., Zhuang, X.H., Vaidya, V.G., and Kim, M.B.: ‘Pose estimation from corresponding point data’, *IEEE Trans. Syst. Man Cybern.*, 1989, **19**, (6), pp. 1426–1446

33 Lowe, D.G.: ‘Fitting parameterized three-dimensional models to images’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991, **13**, (5), pp. 441–450

34 Schaback, R.: ‘Creating surfaces from scattered data using radial basis functions’, in Daelhen, M., Lyche, T., and Shumaker, L.L. (Eds.): ‘Mathematical methods in computer-aided geometric design III’ (Vanderbilt University Press, 1995), pp. 477–496

35 Wolberg, G.: ‘Digital image warping’ (IEEE Computer Society Press, Los Alamos, CA, 1992)