Research Article

Robust Head Pose Estimation Using a 3D Morphable Model

Ying Cai,^{1,2,3} Menglong Yang,⁴ and Ziqiang Li³

¹School of Computer Science, Sichuan University, Chengdu 610064, China

²Wisesoft Software Co., Ltd., Chengdu 610045, China

³College of Information Engineering, Sichuan Agricultural University, Ya'an 625014, China

⁴School of Aeronautics and Astronautics, Sichuan University, Chengdu 610064, China

Correspondence should be addressed to Menglong Yang; steinbeck@163.com

Received 30 September 2014; Accepted 17 November 2014

Academic Editor: Hui Zhang

Copyright © 2015 Ying Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Head pose estimation from single 2D images has been considered as an important and challenging research task in computer vision. This paper presents a novel head pose estimation method which utilizes the shape model of the Basel face model and five fiducial points in faces. It adjusts shape deformation according to Laplace distribution to afford the shape variation across different persons. A new matching method based on PSO (particle swarm optimization) algorithm is applied both to reduce the time cost of shape reconstruction and to achieve higher accuracy than traditional optimization methods. In order to objectively evaluate accuracy, we proposed a new way to compute the pose estimation errors. Experiments on the BFM-synthetic database, the BU-3DFE database, the CUBiC FacePix database, the CMU PIE face database, and the CAS-PEAL-R1 database show that the proposed method is robust, accurate, and computationally efficient.

1. Introduction

As one of the most active research topics, head pose estimation has received significant attention in computer vision field. Human visual system is able to interpret the orientation of a human head naturally, but it is challenging to make computer have this ability. There are so many difficulties we have to overcome for it. However, the head pose as an important indication can be widely used in many applications; for example, in video surveillance, it helps the computer to determine human identity and focus of attention in the scene [1]. In human computer interaction, it helps the computer to understand the intentions of human being [2]. In video conferencing, it helps to estimate user's focus of attention in meetings [3]. In monitoring driver's attention, it helps to ensure the state of driver [4]. Most automatic face recognition algorithms especially try to normalize facial images in order to remove variations caused by anything but the identity of the person. Head pose is just one great part of undesired variations in facial images [5].

There have been numerous approaches to tackle the head pose estimation problem. The temporal information from face video sequence could improve the head pose estimation

[6], but the applications of head pose estimation from a single 2D image are more extensive. Furthermore it is still difficult. So we focus on it. Recent works demonstrate the depth information is the key for overcoming many of the problems inherent to 2D image data. Fortunately, all kinds of sensors have become both affordable and accurate recently including depth sensors [7]. Now many researches could be completed under multiple sensors, such as [8]. So the use of 3D data to assist pose estimation from single 2D image is already practicable.

2. Related Work

Murphy-Chutorian and Trivedi [9] reviewed different algorithms used for head pose estimation. Most head pose estimation algorithms can be divided into two steps. The first step is feature extraction. The second step estimates the head pose according to the obtained features. We can roughly categorize head pose estimation approaches into model based approaches and appearance-based approaches.

Model based approaches: these approaches usually use geometric features. In earlier work, Gee and Cipolla [10] used five key features (points, nose tip, the outer eye corners, and

mouth corners) to estimate the facial orientation. Wang and Sung [11] used two outer eye corners, two inner eye corners, and two mouth corners to estimate pose. They considered the three lines between the outer eye corners, the inner eye corners, and the mouth which are parallel. The ratio of their lengths could be used to estimate the head orientation with the assumption that the eye corners and mouth corners are approximately in the same plane. Because different persons have different ratios, the precision is not very ideal. Lanitis et al. [12] extracted face feature by active shape model (ASM). They adopted a greedy search to match the feature points. But such techniques in [12] usually need a lot of feature landmarks more than 60.

Appearance-based approaches: these approaches use features that are modeled and learned from the training data. Distance metric learning [13], subspace analysis [14], and manifold embedding [15] are popular methods used to extract features. But such techniques are usually sensitive to data alignment and background noise.

Robustness and accuracy both are important in head pose estimation; hence many hybrid approaches are proposed. Reference [16] gives a coarse-to-fine framework to estimate pose. The first stage is based on the subspace analysis. In the second stage, the estimated pose is refined by analyzing finer geometrical structure details captured by bunch graphs. Reference [17] presents a 3D head pose estimation approach based on 3D morphable appearance model. But it considers yaw angle just spanning from -16° to $+16^\circ$ and pitch angle spanning from -8° to $+8^\circ$.

Now the obvious difficulty in many applications of a single 2D face image is still pose, light, and expression invariance. 3D information of the object is very useful to overcome this difficulty. Flexible and expressive 3D face models are a powerful tool to estimate the 3D shape, texture, and pose transfer between individuals using a single photograph [18]. For this reason, 3D morphable models (3DMM) have been introduced a decade ago [19]. A morphable face model represents the shape and texture of samples by a vector space. In this vector space, any convex combination of shape and texture vectors of some face samples describes a realistic human face [20], and the constraints of shape and texture are derived from the statistics of sample faces [18]. But the expensive and tedious construction process makes it impractical. On an SGI R10000 processor, computation time was 50 minutes [18]. IN FER [21] Keeping the optimization time below 180 s, on a 2.8 GHz processor. And iN FER [17] 33.5 s runtimes are measured in MATLAB TM using an Intel Core 2 Duo processor running at 2.4 GHz. Therefore we simplify traditional 3D morphable models to a 3D morphable shape model that only encodes shape in terms of model parameters. Only reconstructing the shape of model is enough to estimate head pose.

During the procedure of 3D face shape reconstruction, model matching is an important step. It is a high-dimensional optimization problem for which both the speed and matching precision have to be considered. Traditionally, the stochastic gradient descent method is hard to converge at optimal point in a short time. In addition, converging result is not stable since it depends too much on initial values. The PSO

algorithm originally is proposed by Kennedy and Eberhart [22, 23], which is a swarm intelligence-based optimization algorithm, which is independent of initial values and the gradient information of object function. And it is central control, which helps to avoid the optimization problem being failed by several “bad” individuals. So we choose it. Furthermore, we improve the PSO algorithm to make it better suitable 3D face shape reconstruction.

This paper presents a novel hybrid head pose estimation method which consists of three major parts. Equation (1) gets the coordinates of five feature points on 2D input image (left and right eye centers, nose tip, and left and right mouse corners). Automatically capturing facial feature points is an important and open problem. There are so many methods presented to complete this task. In this paper, we employ two approaches to get feature points: FACE++ SDK [24] used for 2D database and we recorded the values of feature points during the process of projection for 3D database. Experiments show that five feature points are enough to accurately estimate pose angles (2). According to the best suitable dimensions of shape parameters we found and improved data distribution of shape parameters to structure cost function (3). We combine an improved particle swarm optimization algorithm to minimize the fitting error between the five feature points on 2D input image and 3D shape model. So far we reconstruct a 3D model with the pose we want to get. It is the rotational Euler angles relative to the x , y and z -axes of the head, a.k.a. the input images, three degrees of freedom (yaw, pitch, and roll).

The paper is organized as follows. First, we introduce the 3D morphable shape model in Section 3. In Section 4 we introduce our proposed method of shape parameters estimation and PSO optimization process. In Section 5, we give the experiment results on five databases. Finally, we conclude the paper and discuss potential future work in Section 6.

3. 3D Morphable Face Shape Model

As we know, we could represent any face shape as a linear combination of a limited basis set of face shape prototypes. Now we use $3m$ -dimensional vectors to represent an exemplar face shape as $sv = (x_1, y_1, z_1, \dots, x_m, y_m, z_m)^T$ and define these vectors as shape vectors. In a 3D face scan, the m is the number of vertices and the x_m , y_m , and z_m are coordinates of the m th vertex. When we have n face shape vectors sv_i $i = 1, \dots, n$, we do dense correspondence between them; then the n vectors have same length and a 3D average face shape model $\bar{s}$ can be constructed. Then we perform a principal component analysis (PCA [25, 26]) on the n shape vectors sv_i $i = 1, \dots, n$. The covariance matrix is calculated. Let its eigenvectors be $s_1, s_2, \dots, s_{n-1}$, and let its eigenvalue be $\sigma_{s,1}^2 \geq \sigma_{s,2}^2 \geq \sigma_{s,n-1}^2$. Using the eigenvectors as an orthogonal basis, any face shape can be represented as follows:

$$S = \bar{s} + \sum_{i=1}^{n-1} \alpha_i s_i. \quad (1)$$

FIGURE 1: Virtual camera sketch showing the centre of projection, the focal length, the optical axis, and the image plane.

Obviously different α_i will construct different person; we define α_i as shape parameters.

The Basel face model (BFM) [27] can be constructed with 200 persons' higher resolution laser scans data with a large variety in face shapes and appearance, which were registered with each other by using an improved dense correspondence algorithm. After alignment all the 200 faces are unified with 53490 vertices.

4. Proposed Method

4.1. Cost Function. Through fitting the 3D shape model to an input 2D image we can estimate the pose of the face in this 2D image, because, in the process of fitting, not only the projective parameters and the shape parameters α_i but also the pose parameters will be ascertained. We concatenate the projective parameters (the focal length f of the camera, the 3D translation t_w) and the pose parameters (the rotation angles ϕ, θ , and γ) into a vector ρ .

The primary goal of the fitting is to minimize the sum of square differences about the five facial feature points in the input 2D image ($q_{x,j}, q_{y,j}$) and the projective positions (p_{x,k_j}, p_{y,k_j}) of the five points corresponding vertices k_j in BFM

$$E_F = \sum_{j=1}^5 \left\| \begin{pmatrix} q_{x,j} \\ q_{y,j} \end{pmatrix} - \begin{pmatrix} p_{x,k_j} \\ p_{y,k_j} \end{pmatrix} \right\|^2, \quad (2)$$

where p_{x,k_j} and p_{y,k_j} can be computed through the following method.

The object-centred coordinates of the k th vertex $X_k = (x_k, y_k, z_k)^T$ could be mapped to a position relative to the camera $W_k = (\omega_{x,k}, \omega_{y,k}, \omega_{z,k})^T$ by the rigid transformation [28]:

$$(\omega_{x,k}, \omega_{y,k}, \omega_{z,k})^T = R_\gamma R_\theta R_\phi X_k + T_w, \quad (3)$$

where the angles ϕ and θ control the BFM's rotation around the vertical and horizontal axis (i.e., yaw and pitch) and

γ defines a rotation around the camera axis (i.e., roll). $T_w(t_x, t_y, t_z)$ is a spatial shift.

According to the perspective projection method we could map vertex k to image plane coordinates $(p_{x,k}, p_{y,k})$

$$p_{x,k} = P_x + f \frac{\omega_{x,k}}{\omega_{z,k}}, \quad p_{y,k} = P_y + f \frac{\omega_{y,k}}{\omega_{z,k}}, \quad (4)$$

where f is the focal length of the camera and (P_x, P_y) defines the principal point which is the intersection point of the optical axis and the image plane.

We suggest that the principal point (P_x, P_y) should be equal to the projective position of the origin of object-centered coordinates, Like Figure 1. Then we can assume $t_x = 0, t_y = 0$ and fix t_z to an empirical value. Then only $\alpha, \gamma, \theta, \phi$, and f need to be estimated; this could reduce fitting time. In order to find the projective position of the origin of object-centered coordinates on the 2D input image, we can adjust the origin of object-centered coordinates of the BFM to nose tip and keep optical axis through the origin of object-centered coordinates and collinear with the z -axis of object-centered coordinates.

If we only minimize the E_F in (2), the overfitting is unavoidable. So we add the maximum a posteriori estimator (MAP) into the cost function. Now given the feature points F on the input 2D image our task is to find model parameters with maximum posterior probability $p(\alpha, \rho | F)$. According to Bayes rule,

$$p(\alpha, \rho | F) \sim p(F | \alpha, \rho) p(\alpha, \rho). \quad (5)$$

If we neglect correlations between the variables, the right-hand side is $p(F | \alpha, \rho) p(\alpha) p(\rho)$. We assume that feature point coordinates may be subject to Gaussian pixel noise with a standard deviation σ_F . So

$$p(F | \alpha, \rho) \sim \exp\left(\frac{-1}{2\sigma_F^2} E_F\right). \quad (6)$$

$p(\rho)$ is a normal distribution with the starting values for $\bar{\rho}$ and ad hoc values for σ_R :

$$p(\rho) \sim \exp\left(\frac{-1}{2\sigma_R^2}(\rho - \bar{\rho})^2\right). \quad (7)$$

In [29], Ding et al. propose that the shape parameters are more likely to be subject to Laplace distribution than Gaussian distribution. So the prior probability is

$$p(\alpha) \sim \gamma \exp\left(-\frac{|\alpha|}{b}\right), \quad (8)$$

where γ is a normalization factor.

Then, posterior probability could be maximized by minimizing

$$E = -2 \log p(\alpha, \rho | F), \quad (9)$$

$$E = \frac{1}{\sigma_F^2} E_F + \sum_i \frac{|\alpha_i|}{b_i} + \sum_j \frac{(\rho_j - \bar{\rho})^2}{\sigma_{R,j}^2}. \quad (10)$$

The value of σ_F could be adjusted after empirical experimentation. The value of b_i could be obtained through trial and error method [29]. The value of σ_R could be decided through 3σ theorem.

The equation (10) is the final cost function.

4.2. Improved PSO Algorithm. PSO algorithm is a swarm intelligence-based optimization algorithm which is independent of initial values and the gradient information of object function. Furthermore, the most important step in head shape estimation based on morphable 3D model is model matching. Some models low-dimensional parameters can be determined by skillfully solving a set of linear matrix inequalities [30]. But face model matching is a high-dimensional optimization problem, where both speed and precision are important. So we dynamically adjust the inertia weight in PSO algorithm according to the probability distribution of shape parameters α . The improved matching method will reduce the running time and better the fitting result. The flowchart of the improved PSO algorithm is shown in Figure 2.

In the PSO model, each particle in the search space represents a potential solution, and it is comprised of three parts: a position, a velocity, and a record of its past performance. The best solution can be smoothly found through cooperation and information sharing among individuals in the swarm.

We encode each particle $X = (f, \gamma, \theta, \phi, \alpha_1, \alpha_2, \dots, \alpha_n)$; there f is focal length, γ, θ , and ϕ are the three pose angles, and $\alpha_1, \alpha_2, \dots, \alpha_n$ is the n -dimensional shape coefficients. So the dimension of particle is $m = 4 + n$.

The particles move according to the following equations:

$$\begin{aligned} V_i[t+1] &= \omega V_i[t] + c_1 \times \text{rand}(\cdot) \times (P_i - X_i) \\ &\quad + c_2 \times \text{rand}(\cdot) \times (P_g - X_i), \\ X_i[t+1] &= X_i[t] + V_i[t+1], \end{aligned} \quad (11)$$

FIGURE 2: The flowchart of the improved PSO algorithm.

where $V_i = [v_{i1}, v_{i2}, \dots, v_{im}]$ is the velocity of the particle i . $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is its position. p_i is the best previous position of the particle i (called $pbest$), and p_g is the best previous position among all the particles in the swarm (called $gbest$), and c_1 and c_2 are acceleration factors. c_1 is used to move the particle to the own best position and c_2 is used to move the particle to the global best position. The changes of c_1 and c_2 determine the relative pull for each particle towards $pbest$ and $gbest$. The $\text{rand}(\cdot)$ is random function in the range of $[0, 1]$, which injects random factors into the optimization process. ω is inertia weight. When the value of ω is big, the whole species can jump out of local maximum points easily, while the small ω is beneficial to fast convergence.

Dynamically adjusting the inertia weight can get a better result than fixed value. In model fitting, the value of ω is determined according to the following equation [31]:

$$\omega_i = \frac{1}{p(\alpha)}, \quad (12)$$

where $p(\alpha)$ is obtained from (8).

Particle swarm optimization iterates according to above equations and gives a solution until the E in (10) reach to the minimum.

FIGURE 3: The synthetic 3D faces from BFM model.

FIGURE 4: The relation between the dimensions of shape components and the estimation absolute error.

5. Experimental Results

5.1. The Dimension of Shape Components. The location of feature points in the face of a person related not only to his/her head pose but also to the unique identity. Hence, this method refines shape deformation to ensure the reasonableness of the five feature points' positions on the same pose but different persons.

The dimension of BFM's shape components is 199. But as mentioned in literature [32], some prior components control the most aspects of a face shape while some other components may decrease the reconstruction precision. So we investigate how the dimension of shape components affects the quality of pose estimation.

We use BFM model to construct 20 synthetic 3D faces as reasonable random shape parameter shown in Figure 3. Then we rotate and project them to 2D images; meanwhile we accurately record the five feature points' coordinates.

In Figure 4(a), we give the pose estimation absolute error on three degrees of freedom when the number of shape components varies from 0 to 199 and the pose angle of every degree of freedom varies from -90° to $+90^\circ$ in steps of 10° (19 pose angles in total). In Figure 4(b), we give the estimation absolute error on two degrees of freedom including pitch and yaw (fixing the roll angle to zero). There are 361 pose angles varying from -90° to $+90^\circ$ in steps of 10° .

In Figure 4(a), we can see that when the number of shape components is 20 in yaw direction could get the min absolute error, 35 in pitch and 25 in roll. So we vary shape components from 10 to 35 to test the best value for two degrees of freedom. In Figure 4(b) we can see that when the number of components is 25, the best result is achieved. We show the best result in terms of mean, standard deviation, median, and upper quartile (Q25) and lower quartile (Q75) of the absolute error in Table 1.

TABLE 1: The absolute error for the BFM-synthetic database with 25 shape components on two degrees of freedom (yaw and pitch): 361 pose angles varying from -90° to $+90^\circ$.

BFM-synthetic dataset	Absolute error [$^\circ$]									
	Yaw $-90^\circ \sim +90^\circ$					Pitch $-90^\circ \sim +90^\circ$				
	Mean	Std	Median	Q25	Q75	Mean	Std	Median	Q25	Q75
Our approach	1.18	1.11	0.93	0.39	1.64	1.14	1.13	0.85	0.35	1.68

FIGURE 5: The relation between the number of feature points and the estimation absolute error.

5.2. The Number of Feature Points. In this section we compare the estimation absolute errors when the number of feature points varies from 15 to 5. The absolute error on BFM-synthetic database is very small when the number of feature points is 5 and the tendency of change is not notable. So we do this experiment on the BU-3DFE 3D face database. We project the 100 3D data and record their coordinates of the 15 feature points. On every degree of freedom the pose angle varies from -90° to $+90^\circ$ in steps of 10° . Single degrees of freedom count up 19 pose angles. The 15 feature points include two upper eye midpoints of outline, two lower eye midpoints of outline, two outer eye corners, two inner eye corners, nostril demarcation point, the jaw tip, tow midpoint of eye socket, two mouth corners, and the nose tip. The number of feature points reduces to 13, 11, 9, 7, 6, and 5 when we remove the preceding points in order. In Figure 5, we give the comparison result. The BU-3DFE database [33] consists of 100 individuals with 7 emotional states. We only use neutral emotional images. Each facial model is composed of more than 7,700 vertices and 14,000 triangles. Images of the individuals can be projected in arbitrary pose.

BU-3DFE 3D face database did not supply the two midpoints of eye socket, the nose tip, and nostril demarcation point, so we utilize some supplied points to compute these absent points. We use the points' mean of 1, 5, 3, and 7 (the release file, 4DFE-Featurepoints83.JPG, in BU-3DFE gives this 83 points' definition) to compute the x and y coordinates of right midpoint of eye socket; then according to the x and y coordinates, the nearest points' z coordinate is found, as the z coordinate of right midpoint of eye socket. In the same way,

we compute the left midpoint of eye socket using 9, 11, 13, and 15, compute the nostril demarcation point using 42, 43, and compute the nose tip using 40, 45.

In Figure 5, we can see that when the number of feature points is five the absolute error of pose angle (<5 pixels on average) is enough for many practical applications. Furthermore, it is much easier to obtain 5 feature points than 15 feature points, and it is more efficient during fitting. In the following experiments, we set the number of feature points to 5.

5.3. Comparison with Other Methods. We do head pose estimation on two publicly available datasets (the BU-3DFE 3D face database and the CUBiC FacePix database) to compare with 3D MAM [17] and 3D morphable model (3DMM) method [19].

The CUBiC FacePix database [34] consists of 30 individuals. Three sets of images are available for each individual. The first set contains images in which only yaw angle is annotated and taken from the individuals right to left in one-degree increments. The second and third sets are targeted to illumination experiments. We only use the first set for our pose estimation experiments.

We prepare experiment data according to [17] and then compare the proposed method with 3D MAM and the well known 3DMM method. But we failed to apply for the USF Human ID 3D face database used in [17], so we choose a similar 3D database (the BU-3DFE database) to substitute for it. In BU-3DFE database we use the projected images of all the 100 individuals in the database by only varying the yaw angles from -16° to $+16^\circ$ in steps of 4° while fixing the roll and pitch angle to zero. The five feature points' coordinates are recorded during projection. Table 2(a) presents the comparison results on the USF Human ID and the BU-3DFE. In the CUBiC FacePix database we take those images whose pose angles vary from -16° to $+16^\circ$ in steps of 4° . Table 2(b) presents the comparison result. The five feature points' coordinates are automatically detected by the face++ SDK [24].

Then we extend the previous experiment to two degrees of freedom by changing the yaw angle to $[-16^\circ, 0^\circ, +16^\circ]$ and the pitch angle to $[-8^\circ, 0^\circ, +8^\circ]$ correspondingly. Table 3 presents the results.

In Table 4, we can see the runtime per facial fit. Our approach is faster than the 3DMM [19] but slightly slower than 3D MAM. The runtimes are measured in *MATLAB* using an Intel Core 2 Duo processor running at 2.4 GHz and the resolution of the image is 180×180 pixels. 3D MAM [17] using the same processor and the resolution of the image is 60×80 pixels.

TABLE 2: The results of the 3D MAM [17], the 3DMM [19], and our proposed approach.

(a) Mean, standard deviation, median, and upper and lower quartile of the absolute error of yaw angle, on the BU-3DFE database

USF Human ID	Absolute error [°]				
	Mean	Std	Median	Q25	Q75
3D MAM	4.30	3.51	3.35	1.71	6.08
BU-3DFE	Mean	Std	Median	Q25	Q75
Our approach	3.28	4.21	2.65	1.25	4.69

(b) Mean, standard deviation, median, and upper and lower quartile of the absolute error of yaw angle, on the FacePix database

FacePix	Absolute error [°]				
	Mean	Std	Median	Q25	Q75
3DMM	4.89	3.15	4.48	2.15	7.06
3D-MAM	6.29	4.15	4.62	2.30	8.57
Our approach	4.63	4.91	3.82	1.43	6.59

TABLE 3: Mean, standard deviation, median, and upper and lower quartile of the absolute error of yaw and pitch angles, on the BU-3DFE database.

USF Human ID	Absolute error [°]									
	Yaw					Pitch				
	Mean	Std	Median	Q25	Q75	Mean	Std	Median	Q25	Q75
3DMM	3.9	3.31	2.81	1.55	5.21	5.14	3.66	4.08	2.11	7.55
3D MAM	5.78	4.22	4.86	2.57	7.66	5.89	4.68	4.76	2.32	8.23
BU-3DFE	Mean	Std	Median	Q25	Q75	Mean	Std	Median	Q25	Q75
Our approach	3.46	4.35	2.87	1.40	4.84	4.43	5.99	3.74	1.81	6.57

TABLE 4: Average runtime per facial fit. The results of the 3D MAM [17], the 3DMM [19], and our proposed approach.

	Average runtimes [s]
Our approach	3.7
3D MAM	3.2
3DMM	33.5

In [35], several experiments were conducted on the CUbiC FacePix database using 68 feature points to train support vector regression and AAM fitting. The mean error obtained there in yaw is 3.96° when pose variation is $\pm 45^\circ$. Our approach could reach 3.84° on the same database using 15 face feature points, but pose variation is $\pm 25^\circ$ in yaw because beyond $\pm 25^\circ$ there are obvious deviations in the locations of the feature points detected by using face++ SDK.

5.4. Experimental Results on Other Databases

5.4.1. On the CMU PIE Face Database. The CMU PIE face database [36] consists of over 40,000 facial images of 68 individuals. We only derived the pictures taken by c27, c07, c09, c29, and c05 cameras with a neutral expression and not wearing glasses. There are three individuals' images with some especial factors such as the color of skin and appearance feature, for which FACE++ SDK could not detect face. Excluding the three individuals, our experimental set totally

includes 65 individuals and contains yaw and pitch angle from -22.5° to $+22.5^\circ$. When estimating the pitch angle we adjust the position of optical center to adapt to the shooting environment. We give the result in Table 5.

Actually, there are so many objective factors to affect the accuracy. Figure 6 shows five individuals' frontal pose images. It can be obviously seen that the frontal face images of different people could appear very different. Some unexpected artificial factors maybe cause this phenomenon. And if we fixed the cameras the height of models will affect the pitch angle obviously. In other words, it is difficult to define "standard frontal face image" of all people.

As a result, the estimation of pose angles will have certain deviations. So we propose a new evaluation method for estimation accuracy. θ_i is the annotation for angle i ; $\hat{\theta}_i$ is the estimation for angle i . Conventionally absolute error is defined as $\hat{\theta}_i - \theta_i$ but our modified absolute error is computed as $(\hat{\theta}_i - \hat{\theta}_0) - \theta_i$. We think this new absolute error will help to avoid effect from inaccurate frontal pose and make the absolute error more objective. So we give the modified absolute error in Table 6 and draw the estimation result and the value of $(\hat{\theta}_i - \hat{\theta}_0)$ in Figures 7 and 8.

From the curves (Figures 7 and 8), we can see that the deviations between the estimated pose angles and the annotated ones are relatively consistent across different people. This demonstrates that the method is reasonably accurate despite the possible deviations in the annotated frontal face images.

TABLE 5: Mean, standard deviation, median, and upper and lower quartile of the absolute error of yaw and pitch angle, on the the CMU PIE face database.

CMU PIE	Absolute error [°]									
	Mean	Std	Yaw Median	Q25	Q75	Mean	Std	Pitch Median	Q25	Q75
Our approach	3.88	5.72	3.10	1.98	4.94	3.51	5.40	2.30	1.37	4.19

FIGURE 6: Five individuals' frontal pose image.

(a) The estimation of 65 individuals in CMU database on pitch angles $+22.5^\circ$, 0° , and -22.5°

(b) The $(\hat{\theta}_{\pm 22.5^\circ} - \hat{\theta}_0)$ of 65 individuals in CMU database on pitch angle

FIGURE 7: The result on pitch angle.

(a) The estimation of 65 individuals in CMU database on yaw angles $+22.5^\circ$, 0° , and -22.5°

(b) The $(\hat{\theta}_{\pm 22.5^\circ} - \hat{\theta}_0)$ of 65 individuals in CMU database on yaw angle

FIGURE 8: The result on yaw angle.

5.4.2. *On the CAS-PEAL-R1 Database.* The CAS-PEAL-R1 [37], a subset of the CAS-PEAL face database, has been released for the purpose of research. It contains 30,871 images of 1,040 individuals (595 males and 445 females) with varying pose, expression, accessory, and lighting (PEAL). We are interested only in the pose subset. But pitch pose in the CAS-PEAL-R1 was constructed through artificially rotating head. So we did not test on pitch images, but only using the images with -15° to $+15^\circ$ yaw angles. We show the result in Table 7.

6. Conclusion

In this paper, we introduced a novel head pose estimation method. It only relied on five facial feature points and utilizes the shape model of the BFM. Our method does not need training and could keep stable performance on different database. For more objective evaluation, we modified the way to compute the absolute errors. The experiment results show that the proposed method could be used to estimate face pose in most realistic situation and the absolute error is acceptable

TABLE 6: Mean, standard deviation, median, and upper and lower quartile of the modified absolute errors of yaw and pitch angle, on the the CMU PIE face database.

CMU PIE	Modified absolute error [°]									
	Yaw					Pitch				
	Mean	Std	Median	Q25	Q75	Mean	Std	median	Q25	Q75
Our approach	2.74	3.69	2.06	0.87	4.36	2.28	3.19	1.68	0.64	2.94

TABLE 7: Mean, standard deviation, median, and upper and lower quartile of the modified absolute errors of yaw angle, on the CAS-PEAL face database.

CAS-PEAL-R1	Yaw -15~+15				
	Mean	Std	Median	Q25	Q75
Absolute error [°]	3.78	5.41	3.32	1.70	5.27
D0 absolute error [°]	2.81	3.65	2.39	1.32	4.20

for practical applications. In the future we are going to improve our approach from two aspects. First, we want to improve the optimization algorithm to reduce the running time and enhance estimation accuracy. Second, we plan to construct a new 3D morphable face model including more races to deal with more internationalization applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key Scientific Instrument and Equipment Development Project of China (no. 2013YQ49087903), National Natural Science Foundation of China (Grant no. 61402307), and Educational Commission of Sichuan Province of China (no. 15ZA0007).

References

- [1] J. Tu, T. Huang, and H. Tao, "Accurate head pose tracking in low resolution video," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR '06)*, pp. 573–578, April 2006.
- [2] M. Reale, P. Liu, and L. Yin, "Using eye gaze, head pose, and facial expression for personalized non-player character interaction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '11)*, pp. 13–18, Colorado Springs, Colo, USA, June 2011.
- [3] S. O. Ba and J. M. Odobez, "Recognizing visual focus of attention from head pose in natural meetings," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 16–33, 2009.
- [4] K. Huang and M. M. Trivedi, "Driver head pose and view estimation with single omnidirectional video stream," in *Proceedings of the 1st International Workshop on In-Vehicle Cognitive Computer Vision Systems and the 3rd International Conference on Computer Vision Systems*, Graz, Austria, 2003.
- [5] X. Chai, S. Shan, X. Chen, and W. Gao, "Locally linear regression for pose-invariant face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1716–1725, 2007.
- [6] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying eye center localization by head pose cues," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 612–618, Miami, Fla, USA, June 2009.
- [7] T. Weise, B. Leibe, and L. van Gool, "Fast 3D scanning with automatic motion compensation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, June 2007.
- [8] H. Zhang, Y. Shi, and A. S. Mehr, "Robust weighted H_{∞} filtering for networked systems with intermittent measurements of multiple sensors," *International Journal of Adaptive Control and Signal Processing*, vol. 25, no. 4, pp. 313–330, 2011.
- [9] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, 2009.
- [10] A. Gee and R. Cipolla, "Determining the gaze of faces in images," *Image and Vision Computing*, vol. 12, no. 10, pp. 639–647, 1994.
- [11] J.-G. Wang and E. Sung, "EM enhancement of 3D head pose estimated by point at infinity," *Image and Vision Computing*, vol. 25, no. 12, pp. 1864–1874, 2007.
- [12] A. Lanitis, C. J. Taylor, T. F. Cootes, and T. Ahmed, "Automatic interpretation of human faces and hand gestures using flexible models," in *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, 1995.
- [13] X. Wang, X. Huang, J. Gao, and R. Yang, "Illumination and person-insensitive head pose estimation using distance metric learning," in *Computer Vision—ECCV 2008*, pp. 624–637, Springer, 2008.
- [14] Y. Fu and T. S. Huang, "Graph embedded analysis for head pose estimation," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR '06)*, pp. 3–8, Southampton, UK, April 2006.
- [15] B. Raychev, I. Yoda, and K. Sakaue, "Head pose estimation by nonlinear manifold learning," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 4, pp. 462–466, IEEE, Cambridge, UK, August 2004.
- [16] J. Wu and M. M. Trivedi, "A two-stage head pose estimation framework and evaluation," *Pattern Recognition*, vol. 41, no. 3, pp. 1138–1158, 2008.
- [17] M. Storer, M. Urschler, and H. Bischof, "3D-MAM: 3D morphable appearance model for efficient fine head pose estimation from still images," in *Proceedings of the 12th IEEE International Conference on Computer Vision Workshops (ICCV Workshops '09)*, pp. 192–199, Kyoto, Japan, October 2009.
- [18] S. Romdhani, *Face image analysis using a multiple features fitting strategy [Ph.D. thesis]*, University of Basel, 2005.

- [19] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 187–194, ACM Press/Addison-Wesley, 1999.
- [20] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 733–742, 1997.
- [21] J. R. Tena Rodriguez, *3D face modelling for 2D+ 3D face recognition [Ph.D. thesis]*, Centre for Vision, Speech, and Signal Processing, University of Surrey, 2007.
- [22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.
- [23] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, vol. 1, pp. 39–43, New York, NY, USA, 1995.
- [24] "MEGVII spotlighting vision technology," Face data application research page, <http://www.faceplusplus.com.cn/>.
- [25] I. Jolliffe, *Principal Component Analysis*, John Wiley & Sons, 2005.
- [26] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America*, vol. 4, no. 3, pp. 519–524, 1987.
- [27] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '09)*, pp. 296–301, Genova, Italy, September 2009.
- [28] E. Angel, *Interactive Computer Graphics: A Top-Down Approach Using OpenGL*, Addison-Wesley, 2003.
- [29] L. Ding, X. Ding, and C. Fang, " L_1 -constrained 3-d face sparse reconstruction," *Journal of Tsinghua University (Science and Technology)*, no. 5, pp. 581–585, 2012.
- [30] H. Zhang and J. Wang, "State estimation of discrete-time Takagi–Sugeno fuzzy systems in a network environment," *IEEE Transactions on Cybernetics*, 2014.
- [31] Y. Ge, *3D face modeling based on morphable model [Ph.D. thesis]*, Beijing University of Technology, 2012.
- [32] X. Gong and G. Wang, "A dynamic component deforming model for face shape reconstruction," in *Advances in Visual Computing*, pp. 488–497, Springer, 2007.
- [33] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR '06)*, pp. 211–216, IEEE, April 2006.
- [34] J. A. Black Jr., M. Gargesha, K. Kahol, P. Kuchi, and S. Panchanathan, "A framework for performance evaluation of face recognition algorithms," in *Internet Multimedia Management Systems III*, vol. 4862 of *Proceedings of the SPIE*, pp. 163–174, Boston, Mass, USA, August 2002.
- [35] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. V. Rohith, "Fully automatic pose-invariant face recognition via 3D pose normalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 937–944, IEEE, Barcelona, Spain, November 2011.
- [36] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–51, Washington, DC, USA, May 2002.
- [37] W. Gao, B. Cao, S. Shan et al., "The CAS-PEAL large-scale chinese face database and baseline evaluations," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 38, no. 1, pp. 149–161, 2008.