Efficient Model-based Linear Head Motion Recovery from Movies

Jian Yao Wai-Kuen Cham

Department of Electronic Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
E-mail: {jianyao,wkcham}@ee.cuhk.edu.hk

Abstract

In this paper, we propose an efficient method that estimates the motion parameters of a human head from a video sequence by using a three-layer linear iterative process. In the innermost layer, we estimate the motion of each input face image in a video sequence based on a generic face model and a small set of feature points. A fast iterative least-square method is used to recover these motion parameters. After that, we iteratively estimate three model scaling factors using multiple frames with the recovered poses in the middle layer. Finally, we update 3D coordinates of the feature points on the generic face model in the outermost layer. Since all iterative processes can be solved linearly, the computational cost is low. Tests on synthetic data under noisy conditions and two real video sequences have been performed. Experimental results show that the proposed method is robust and has good performance.

1. Introduction

Head pose estimation is often required in many applications such as 3D face modelling [1–3], face recognition [4, 5], facial expression analysis [6] and lip reading [7]. In this paper, we propose a new algorithm that can accurately estimate the motion and so the pose of a human head from a video sequence by using a generic wireframe model and a three-layer linear iterative process. In computer animation, an important task is to fit a wireframe model to a head in a video sequence automatically. The proposed algorithm can be used to accomplish the task. It can estimate the head poses, the model scaling factors and the 3D coordinates of some critical feature points with good accuracy.

Conventionally, the rigid transformation relating 2D images to known 3D geometry was determined by a nonlinear optimization which is often solved using the Gauss-Newton method and the Levenberg-Marquardt method. Typical examples of these approaches are the works of Lowe [8] and Haralick [9]. An integrating method for fusing different 2D and 3D measurements for pose estimation was also provided in [10]. In [11], Choi et al. estimated 3D facial pose using the iterative EM algorithm by aligning the orthographic projection of a 3D model and the 2D data. Recently, methods were proposed to solve pose estimation using linear iterative processes [12–14]. By these linear pose estimation methods, the computing cost is greatly reduced. All these methods are based on point correspondences.

Face texture and head silhouettes can also be utilized to determine the pose of a head. Cascia et al. [15] proposed a fast and reliable head tracking algorithm by modelling a head as a texture-mapped cylinder. Xiao et al. [16] proposed a robust algorithm for full-motion recovery of a head using a cylindrical model. In [17], structure and motion are determined from silhouettes. In [18, 19], a novel method to obtain the 3D shape of an arbitrary human face using a sequence of silhouette images as input was proposed. However, this approach relies on accurate contour tracking and is applicable only to specific environmental settings.

A set of techniques for modeling and animating realistic faces from photographs and videos was presented in [1]. They can be used to estimate the face position, orientation and facial expression of a face in each frame. Zhang et al. [3] proposed a robust and rapid method for generating animated faces from video sequences based on a model-based modeling approach in which they make use of generic knowledge of faces in the head motion determination, head tracking, model fitting and multiple-view bundle adjustment. For automatic creation of 3D face models from video images, both face detection [20] and face tracking [21] are important steps.

The proposed algorithm is a generalization of the work by Or et al. [14] who introduced a fast and efficient iterative pose estimation algorithm using a two-stage linear optimization process. Recently, a similar algorithm had been proposed [13]. The major difference between the proposed algorithm and those in [13, 14] is that the proposed algorithm does not need the accurate 3D geometry of a head which can be iteratively estimated using multiple frames with the recovered poses. Furthermore, the proposed

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**

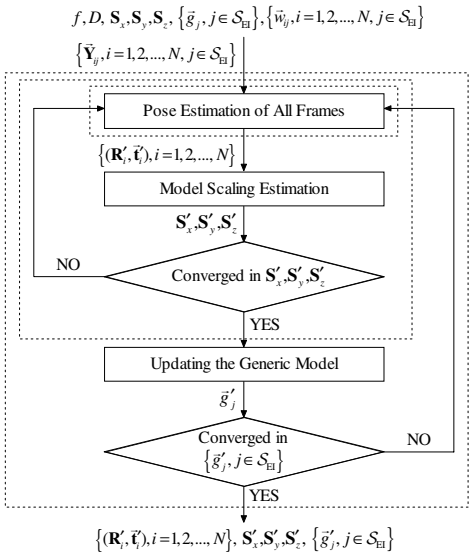


Figure 1: The flowchart of the proposed algorithm.

method can accurately recover the motion parameters even under noisy conditions.

In this work, we formulate the head pose estimation problem as that of minimizing an error function between a face image and a generic face model by using the inverse projection rays as a guide to determine the head pose. The original nonlinear pose estimation problem is divided into three linear estimation stages, namely the global translation approximation stage, the depth approximation stage and the least-square fitting stage. After estimating the poses of multiple frames, we then estimate the three model scaling factors using these recovered poses in the linear iterative process. At the final stage, we update 3D coordinates of the feature points to obtain a more accurate face model using both the recovered model scaling factors and the recovered poses.

The rest of this paper is organized as follows. A system overview is given in Section 2. The pose estimation problem between a 2D face image and a 3D generic face model is formulated and discussed in Section 3. The estimation of three model scaling factors using multiple frames is described in Section 4. The updating algorithm of 3D coordinates of feature points is investigated in Section 5. Experimental results based on synthetic data and real video sequences are given in Section 6. Finally, concluding remarks are provided in Section 7.

2. System Overview

Figure 1 shows the flowchart of the proposed algorithm. In the system, we use a generic 3D wireframe face model as shown in Figure 10(a) which was developed at Instituto Superior Tecnico (IST) [22]. The input is a video sequence containing a human face with the 2D coordinates of needed feature points extracted from the 2D input face images. We assume that the projection system can be regarded as a weak perspective projection system (see Figure 2). The output consists of the pose of a human head for each frame, the three model scaling factors and the updated 3D coordinates of the feature points on the generic face model (see Figure 1).

The proposed system consists of three linear iterative estimation processes as shown in Figure 1. In the innermost layer, we estimate the motion parameters of each input face image based on a face model and a set of feature points. The three model scaling factors are then estimated using multiple frames with the recovered poses in the middle layer. Finally, we update 3D coordinates of the feature points on the generic face model in the outermost layer.

3. Pose Estimation

For convenience, we consider the problem in which the camera is fixed and the head moves. We recover the motion of a human head for each frame by computing six absolute motion parameters – θ_x, θ_y, θ_z, t_x, t_y and t_z.

3.1. Problem Formulation

Consider a camera with the origin of the image projective plane placed at $(0, 0, f)$ from the perspective center $\mathbf{O}$ where f is the focal length of the image projective plane and D is the distance between the origin $\mathbf{O}$ of the camera coordinate system and the origin $\mathbf{o}_v$ of the 3D face models (see Figure 2). The center of the generic face model shown in Figure 10(a) is the origin $\mathbf{o}_v$. In our perspective projection system shown in Figure 2, the coordinates of the point $\vec{P}_j = (X_{P_j}, Y_{P_j}, Z_{P_j})^{\mathrm{T}}$ and the point $\vec{v}_j = (x_{v_j}, y_{v_j}, z_{v_j})^{\mathrm{T}}$ are with respective to the origins $\mathbf{O}$ and $\mathbf{o}_v$ respectively. These two vectors represent the same location and the point $\vec{w}_j = (x_{w_j}, y_{w_j})^{\mathrm{T}}$ is the projected 2D point on the image plane from the point $\vec{P}_j$. Assume that a set of feature points $\{\vec{v}_j, j \in \mathcal{S}_{\text{EI}}\}$[1] on the generic face model with respective to $\mathbf{o}_v$ are given. This point set corresponds to another point set $\{\vec{P}_j\}$ with respective to $\mathbf{O}$. Now we consider the rigid transformation of the 3D generic face model by scaling, rotating and translating the point set $\{\vec{v}_j\}$, yielding $\{\vec{v}'_j = \mathbf{R}\mathbf{S}\vec{v}_j + \vec{\mathbf{t}}\}$ where $\mathbf{S}$ denotes a 3×3 diagonal model scaling matrix, $\mathbf{S} = diag(\mathbf{S}_x, \mathbf{S}_y, \mathbf{S}_z)$, and $\mathbf{R}$ and $\vec{\mathbf{t}}$ stand for the rotation matrix and the translation vector respectively. It corresponds to another point set $\{\vec{P}'_j = (X'_{P_j}, Y'_{P_j}, Z'_{P_j})^{\mathrm{T}}\}$ w.r.t.

1 Throughout this paper, the symbol $\mathcal{S}_{\text{EI}}$ denotes the set containing the number of our defined feature points on a 3D face model (see Figure 3). The subscript 'j' denotes the j-th feature point and $j \in \mathcal{S}_{\text{EI}}$.

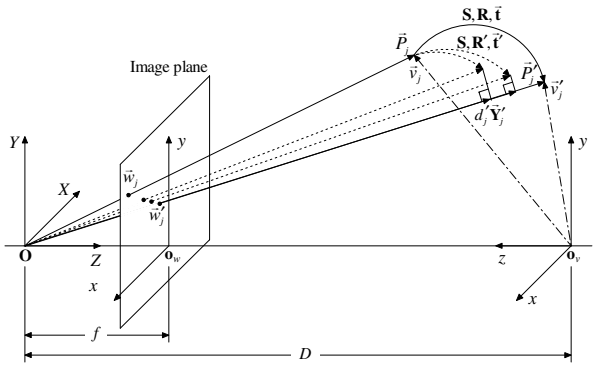


Figure 2: Perspective projection system.

$\mathbf{O}$, which is then projected on the image projective plane, giving $\{w'_j = (x'_{w_j}, y'_{w_j})^{\mathrm{T}}\}$ (see Figure 2).

In a weak perspective projection system, the coordinates of $\vec{P}'_j$ and $\vec{w}'_j$ are related by:

$$x'_{w_j} = -f\frac{X'_{P_j}}{Z'_{P_j}} \approx -f\frac{X'_{P_j}}{D},\ \ y'_{w_j} = f\frac{Y'_{P_j}}{Z'_{P_j}} \approx f\frac{Y'_{P_j}}{D}. \quad (1)$$

In this case, any point $\vec{w}'_j$ tracked in an image will give an inverse projection ray with unit vector $\vec{\mathbf{Y}}'_j$ given by $(x'^2_{w_j} + y'^2_{w_j} + f^2)^{-1/2}(x'_{w_j}, y'_{w_j}, f)^{\mathrm{T}}$. We can then calculate the actual coordinates of the transformed point $\vec{P}'_j$ on a 3D face model as follows:

$$\vec{P}'_j = d'_j\vec{\mathbf{Y}}'_j, \quad (2)$$

where d'_j denotes the depth of the transformed point $\vec{P}'_i$ on the 3D face model from the perspective center $\mathbf{O}$.

3.2. Three-Stage Linear Pose Estimation

The problem of pose estimation between an input face image and a generic face model can be described as follows. Given a set of 2D feature points $\{w'_j\}$ on the input face image, three model scaling factors and a set of 3D feature points $\{\vec{v}_j\}$ on the generic face model, we seek $\mathbf{R}$, $\vec{\mathbf{t}}$ and $\{d'_i\}$ by minimizing the following error function:

$$\varepsilon^2(\mathbf{R},\vec{\mathbf{t}},\{d'_j\}) = \sum_{j\in\mathcal{S}_{\mathrm{EI}}}\lambda_j\left\|\mathcal{D}(d'_j\vec{\mathbf{Y}}'_j) - \left(\mathbf{RS}\vec{v}_j + \vec{\mathbf{t}}\right)\right\|^2, \quad (3)$$

where d'_j is the depth value of the point $\vec{P}'_j$ on the input face image and $\vec{\mathbf{Y}}'_j$ is an unit inverse projective ray vector of the feature point $\vec{w}'_j$ from the perspective center $\mathbf{O}$. The coefficient λ_j is determined by the availability of the j-th feature point in the input face image. It is 1 if we can successfully extract the j-th feature point in the input face image, otherwise it is 0. The function $\mathcal{D}(.)$ denotes the coordinate conversion relationship between $\vec{P}'_j$ w.r.t. $\mathbf{O}$ and $\vec{v}'_j$ w.r.t. $\mathbf{o}_v$, which is given as follows:

$$\vec{v}'_j = \mathcal{D}(\vec{P}'_j) = diag(-1,1,-1)(\vec{P}'_j - [0\ \ 0\ \ D]^{\mathrm{T}}), \quad (4)$$
$$\vec{P}'_j = \mathcal{D}(\vec{v}'_j) = diag(-1,1,-1)(\vec{v}'_j + [0\ \ 0\ \ D]^{\mathrm{T}}). \quad (5)$$

The computation requirement of the nonlinear pose estimation problem increases quickly as the number of used feature points increases. In order to reduce the computation time, the pose estimation is divided into three linear iterative estimation problems which can be solved efficiently. The first stage approximates the global translation vector of the generic model. The second stage updates the depth values $\{d'_j\}$ of the feature points $\{\vec{P}'_j\}$ on the estimating face model using recovered motion parameters. The third stage is to determine the rigid motion parameters by using a least-square minimization algorithm. The above three stages are repeated in turn until changes of six motion parameters θ_x, θ_y, θ_z, t_x, t_y and t_z between two consecutive iterations are less than predefined threshold values.

3.2.1. Global Translation Approximation Stage Given a fixed rotation matrix $\mathbf{R}$ and a known model scaling matrix $\mathbf{S}$, the optimal value for the global translation vector $\vec{\mathbf{t}}$ can be computed in closed form as:

$$\vec{\mathbf{t}}' = diag(-1,1,-1)\left(\sum_{j\in\mathcal{S}_{\mathrm{EI}}}\lambda_j(I - \vec{\mathbf{Y}}_j\vec{\mathbf{Y}}_j^{\mathrm{T}})\right)^{-1} \sum_{j\in\mathcal{S}_{\mathrm{EI}}}\lambda_j(\vec{\mathbf{Y}}_j\vec{\mathbf{Y}}_j^{\mathrm{T}} - I)\mathcal{D}(\mathbf{RS}\vec{v}_j), \quad (6)$$

where I is an identity matrix and $\mathbf{R} = I$ in the first iteration.

3.2.2. 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 feature points and the true feature points becomes small. The 3D positions of the feature points $\{\vec{P}'_j\}$ on the estimating face model from the perspective center $\mathbf{O}$ are approximated by the perpendicular intersection of the corresponding feature points $\{\mathcal{D}(\mathbf{R}'\mathbf{S}\vec{v}_j + \vec{\mathbf{t}}')\}$ on the transformed generic face model (see Figure 2). Hence, the depth values can be updated as follows:

$$d'_j = \vec{\mathbf{Y}}'^{\mathrm{T}}_j\mathcal{D}(\mathbf{R}'\mathbf{S}\vec{v}_j + \vec{\mathbf{t}}'). \quad (7)$$

By this means, the estimated depth values of feature points on the input face image will be gradually moved toward the true depth values.

3.2.3. Least-Square Fitting Stage After the depth values $\{d'_j\}$ are determined, the pose represented by $\mathbf{R}$ and $\vec{\mathbf{t}}$ of the input face image can be estimated by minimizing the following error function:

$$\varepsilon^2(\mathbf{R},\vec{\mathbf{t}}) = \sum_{j\in\mathcal{S}_{\mathrm{EI}}}\lambda_j\|\vec{v}'_j - (\mathbf{RS}\vec{v}_j + \vec{\mathbf{t}})\|^2, \quad (8)$$

where $\vec{v}'_j$ is the estimated j-th feature points in 3D space w.r.t. $\mathbf{o}_v$ given by:

$$\vec{v}'_j = \mathcal{D}(\vec{P}'_j) = \mathcal{D}(d'_j \vec{\mathbf{Y}}'_j). \quad (9)$$

The singular value decomposition method described in [9] is used to solve this least-square minimization problem.

4. Model Scaling Estimation

After estimating the motion parameters of all frames in the above pose estimation process, we further estimate three model scaling factors using these frames with recovered poses by minimizing the following error function:

$$\varepsilon^2(\mathbf{S}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} \left\| \vec{v}'_{ij} - \left(\mathbf{R}'_i \mathbf{S} \vec{g}_j + \vec{\mathbf{t}}'_i \right) \right\|^2, \quad (10)$$

where N is the total number of frames used. $\mathbf{R}'_i$ and $\vec{\mathbf{t}}'_i$ denote the recovered absolute rotation matrix and absolute translation vector of the i-th frame respectively. $\vec{g}_j$ denotes the j-th feature point on the generic face model. $\vec{v}'_{ij}$ is the last updated 3D coordinate vector of the j-th feature point in the i-th frame and it can be calculated using (9). The coefficient λ_{ij} is determined by the availability of the j-th feature point in the i-th frame. It is 1 if we can successfully extract the j-th feature point in the i-th frame, otherwise it is 0.

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

$$\mathbf{S}'_x = \frac{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} \left(\vec{v}'_{ij} - \vec{\mathbf{t}}'_i \right)^{\mathrm{T}} \vec{\mathbf{r}}'_{i1} x_{g_j}}{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} (x_{g_j})^2}, \quad (11)$$

$$\mathbf{S}'_y = \frac{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} \left(\vec{v}'_{ij} - \vec{\mathbf{t}}'_i \right)^{\mathrm{T}} \vec{\mathbf{r}}'_{i2} y_{g_j}}{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} (y_{g_j})^2}, \quad (12)$$

$$\mathbf{S}'_z = \frac{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} \left(\vec{v}'_{ij} - \vec{\mathbf{t}}'_i \right)^{\mathrm{T}} \vec{\mathbf{r}}'_{i3} z_{g_j}}{\sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} (z_{g_j})^2}, \quad (13)$$

where $\vec{\mathbf{r}}'_{i1}$, $\vec{\mathbf{r}}'_{i2}$ and $\vec{\mathbf{r}}'_{i3}$ denote the first, second and third column vector of the rotation matrix $\mathbf{R}'_i$ respectively, i.e., $\mathbf{R}'_i = [\vec{\mathbf{r}}'_{i1}\ \vec{\mathbf{r}}'_{i2}\ \vec{\mathbf{r}}'_{i3}]$, and x_{g_j}, y_{g_j} and z_{g_j} denote the x-, y- and z-coordinate values of the j-th feature point $\vec{g}_j$ on the generic face model respectively, i.e., $\vec{g}_j = (x_{g_j}, y_{g_j}, z_{g_j})^{\mathrm{T}}$.

5. Updating the Generic Model

We have recovered the model scaling factors which can then be used to scale the generic face model to fit input face images. However, accurate 3D coordinates of feature points are also very important for realistic modeling of a particular person in a video sequence. Based on the recovered head poses in all frames and the model scaling factors, we can further update the locations of feature points $\{\vec{g}_j\}$ on the generic model in order to match a particular person closely by minimizing the following error function:

$$\varepsilon^2(\{\vec{g}_j\}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{S}_{\text{EI}}} \lambda_{ij} \left\| \vec{v}'_{ij} - \left(\mathbf{R}'_i \mathbf{S}' \vec{g}_j + \vec{\mathbf{t}}'_i \right) \right\|^2, \quad (14)$$

where N, $\mathbf{R}'_i$, $\vec{\mathbf{t}}'_i$, $\vec{v}'_{ij}$ and λ_{ij} have the same meaning as in (10) and $\mathbf{S}'$ is the recovered model scaling matrix.

By taking the partial derivative of ε^2 in (14) w.r.t. $\vec{g}_j$ and then setting it to zero, we update $\vec{g}_j$ as follows:

$$\vec{g}'_j = \mathbf{S}'^{-1} \frac{\sum_{i=1}^{N} \lambda_{ij} \mathbf{R}'^{\mathrm{T}}_i \left(\vec{v}'_{ij} - \vec{\mathbf{t}}'_i \right)}{\sum_{i=1}^{N} \lambda_{ij}}. \quad (15)$$

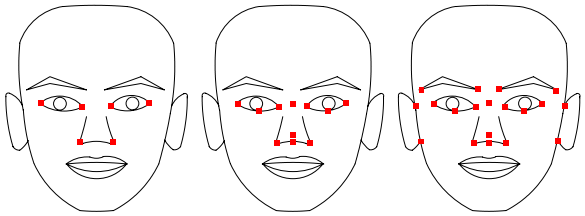
In this work, we assume that the human head is a symmetric body. We keep the symmetry by both averaging the coordinate values of symmetric points and keeping z-coordinate values of the points on the symmetric line still at zero while $\vec{g}_j$ are being updated.

6. Experimental Results

To test the validity of our proposed algorithms, both synthetic data and real video sequences were utilized. For synthetic data, we can compare the recovered results with the true values. In this work, we use root mean square error for performance evaluation. Real video testing reflects both robustness and applicability of our algorithms.

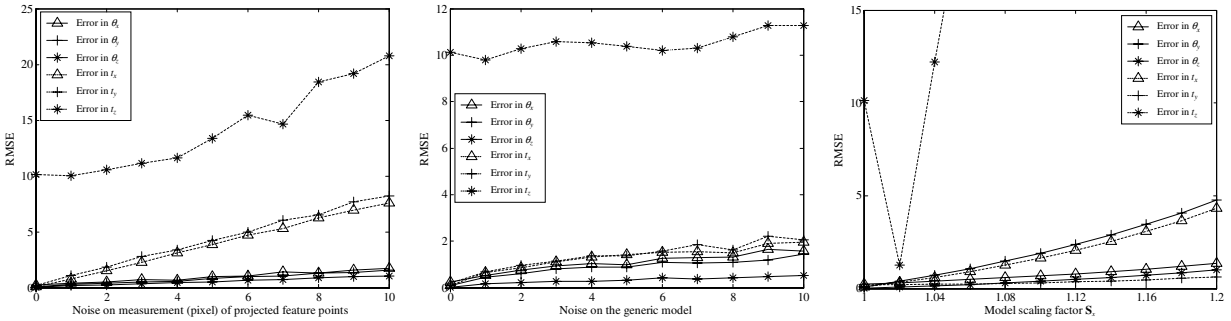
6.1. Synthetic Data

In these experiments, the width and the height of the generic head model are 170 and 270 respectively. We set the focal length of lens $f = 10$ and the distance between the origin $\mathbf{O}$ of the camera coordinate system and the origin $\mathbf{o}_v$ of the 3D face models $D = 2000$. The projection system used here is assumed to be a week perspective projection system. In addition, we assume that 3D face models will be



(a) $\mathcal{S}^1_{\text{EI}}$ (b) $\mathcal{S}^2_{\text{EI}}$ (c) $\mathcal{S}^3_{\text{EI}}$

Figure 3: Location of used feature points.



(a) (b) (c)

Figure 4: The RMSE of each motion parameter: (a) versus the measurement noise (pixel) on the projected feature points; (b) versus the noise on the 3D coordinates of feature points on the generic face model; (c) versus a model scaling factor with other two model scaling factors equal one.

projected on an image plane of size 4×4 and then digitized to a screen resolution of 512×512. As a result, the width and the height of the observed head models on the screen plane are about 217 and 345 respectively. The process of digitization introduces noise in this case. In practice, there are other kinds of noise. When we automatically or manually extract the feature points of the input face images, noise will be added on the x- and y-coordinates of feature points. For a particular person in a given video sequence, the generic face model can be regarded as an approximation that contains noise on its three model scaling factors and noise on its feature points in 3D space. We shall simulate these noise in our experiments using Gaussian random noise. The sizes of noise shown in all the plots denote the variance of Gaussian random noise. In our experiments, the root mean square error (RMSE) is used to measure the accuracy of the recovered results. The RMSE of a recovered vector is defined as the square root of the average Euclidean norm of the difference between the recovered vector and the true vector.

In order to evaluate how the number of feature points affects the performance of the proposed algorithm, three sets of feature points as shown in Figure 3 were examined using synthetic data and they are denoted by $\mathcal{S}_{\text{EI}}^1$, $\mathcal{S}_{\text{EI}}^2$ and $\mathcal{S}_{\text{EI}}^3$. $\mathcal{S}_{\text{EI}}^3$ will be chosen for evaluating the performance of our algorithms except in the experiments on the comparison of the performance using different sets of feature points. All these feature points were marked manually so as to remove any uncertainty due to feature point extraction.

6.1.1. Testing the pose estimation algorithm First we tested the accuracy of each motion parameter estimated by the proposed pose algorithm using a generic face model and the 2D feature points projected from the transformed face model by varying one of the parameters while keeping all the other parameters at zeros. Under a noiseless condition, the experimental results show that we can accurately recover each motion parameter.

The performance of the proposed pose estimation algorithm with different sets of feature points under noisy conditions was investigated by 100 data sets obtained by transforming the generic wireframe face model with six fixed motion parameters ($\theta_x = 8^{\text{o}}, \theta_y = 8^{\text{o}}, \theta_z = 8^{\text{o}}, t_x = 9, t_y = 11, t_z = 7$) and different kinds of random noise added.

The first set of experiments tested the performance of the proposed pose estimation algorithm under noisy conditions. In addition to the digitization noise which was always added, we added noise to one of the three data which are $\vec{w}_{ij}$, $\vec{v}_j$ and a model scaling factor. The results are shown in Figure 4 from which we can observe that larger noise will result in larger error. We also observe that the algorithm cannot accurately recover t_z even with only digitization noise on the coordinates of the extracted feature points from the input face image.

Another set of experiments tested the performance of the proposed pose estimation algorithm with different sets of

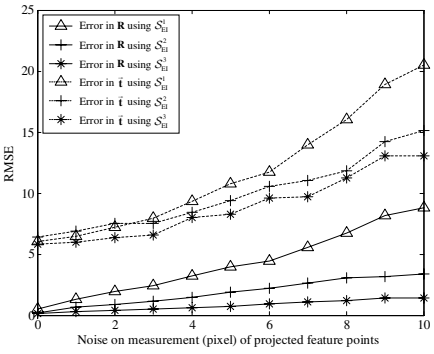
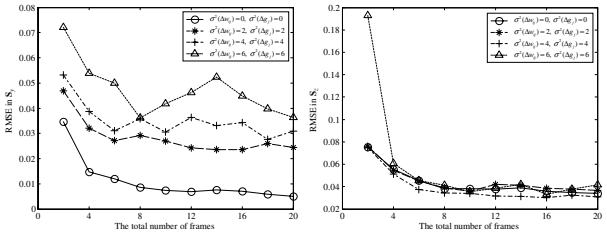


Figure 5: The RMSE of rotation angles and translation values using different set of feature points versus the measurement noise (pixel) on the projected feature points.



(a) (b)

Figure 6: The RMSE of the recovered scaling factors using different number of frames with the input scaling factors $\mathbf{S}_x$=1, $\mathbf{S}_y$=1.2 and $\mathbf{S}_z$=1.2 under different noisy conditions.

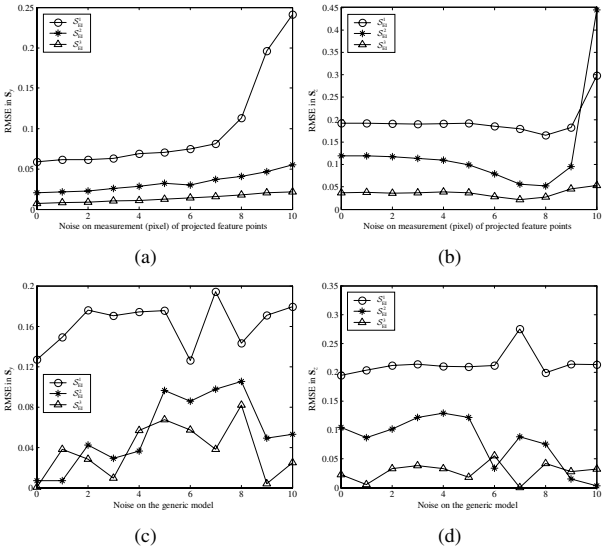
feature points under each kind of noise while keeping other kinds of noise at zero except digitization noise. For saving the paper space, only one group of experimental results is shown in Figure 5. From these experimental results, we observe that more feature points produce more accurate estimation results.

6.1.2. Testing the model scaling estimation algorithm In this experiment, the proposed model scaling estimation algorithm was tested using 100 randomly generated data sets by randomly rotating and translating the known scaled generic wireframe face model with uniform random rotation angles $(\theta_x, \theta_y, \theta_z)$ and uniform random translation vectors (t_x, t_y, t_z) where $\theta_x, \theta_y, \theta_z \in [-50^o, 50^o]$ and $t_x, t_y, t_z \in [-100, 100]$. Each data set can be regarded as a video sequence.

To evaluate the performance of the recovered model scaling factors, we should compare their ratios instead of the true recovered scaling factors. We compare the recovered model scaling factors in the y- and z-axes while keeping the same model scaling factor in the x-axis.

First we tested the performance of the recovered model scaling factors using different number of frames under different noisy conditions. The results are shown in Figure 6 where $\sigma^2(\Delta w_{ij})$ and $\sigma^2(\Delta g_j)$ stand for the variance of Gaussian random noise on measurement (pixel) of projected feature points and on 3D coordinates of feature points on the generic face model respectively. We can observe that more frames will produce more accurate scaling factors. However, the performance does not improve when the total number of frames is larger than 10.

In another set of experiments, we tested the recovered performance under different noisy conditions by utilizing different sets of feature points. The total number of frames used in these experiments is 10. The tested results are shown in Figure 7. From the experimental results, we can observe that the noise on measurement (pixel) of 2D projected feature points smoothly affects the recovered performance. However, the noise on the 3D coordinates of feature points on the generic face model affects the performance in a random way. Generally, more feature points used will produce more accurate estimation.



(a) (b)

(c) (d)

Figure 7: The RMSE of the recovered scaling factors with the input scaling factors $\mathbf{S}_x$=1, $\mathbf{S}_y$=1.2 and $\mathbf{S}_z$=1.2: (a)-(b) under different noise on measurement (pixel) of 2D projected feature points; (c)-(d) under different noise on the 3D coordinates of feature points on the generic face model.

6.1.3. Testing the generic model estimation algorithm In this experiment, the proposed generic model estimation algorithm, which estimates $\vec{j}$ from the projected feature points and the poses, was tested using 100 random data sets generated in the same way as the above tests.

Two sets of experiments were performed to evaluate the performance on different number of frames used and under different noisy conditions. Experimental results as shown in Figure 8 show that more frames used will produce more accurate estimation. However, more feature points used will result in worse performance especially at large noise level (see Figure 9(a)). The 3D coordinates of features points of a person are not necessary the same as those of the generic model. The difference is represented as the original noise in Figure 9(b). The proposed algorithm can update the generic model to form a new one with smaller errors. Also, more feature points produce more accurate model.

6.2. Real Videos

In this section, the poses, the model scaling factors and the generic face model were estimated from two real video sequences using the proposed algorithms. In these two experiments, feature points $\mathcal{S}_{\mathrm{EI}}^3$ were used. The generic face model, the focal length f and the distance D between the origins $\mathbf{O}$ and $\mathbf{o}_v$ are the same as those in the synthetic

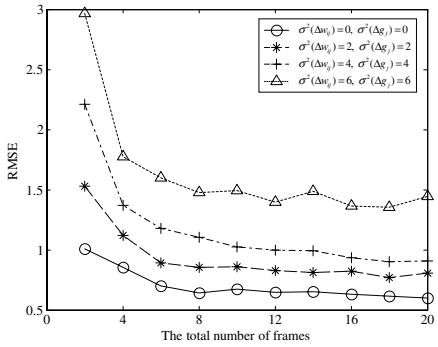
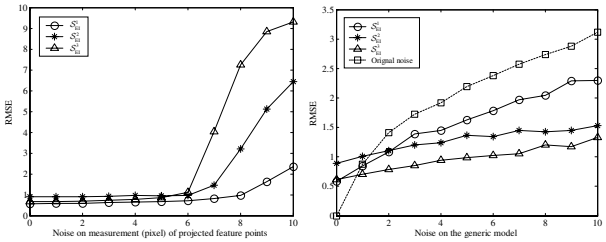


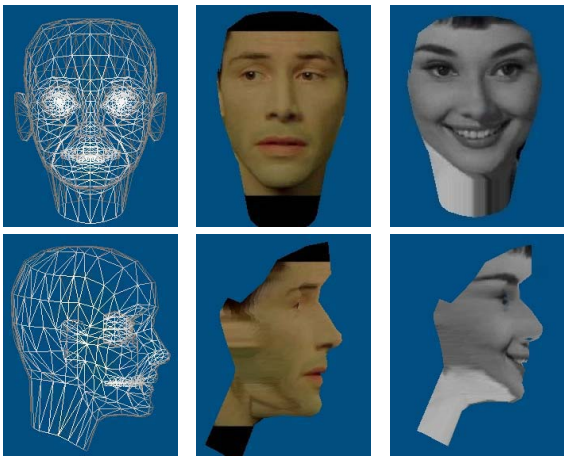
Figure 8: The RMSE of the recovered generic face models using different number of frames with the input scaling factors $\mathbf{S}_x$=1, $\mathbf{S}_y$=0.8 and $\mathbf{S}_z$=1.2 under different noisy conditions.



(a) (b)

Figure 9: The RMSE of the recovered generic face models with the input scaling factors $\mathbf{S}_x$=1, $\mathbf{S}_y$=0.8 and $\mathbf{S}_z$=1.2 under different noise: (a) on measurements (pixel) of 2D projected feature points; (b) on the 3D coordinates of feature points on the generic face model.

data experiments. Each of these two video sequences has 18 consecutive frames. Figure 10 shows the results. The scaled generic face model was first deformed based on the recovered 3D coordinates of the chosen feature points in the generic model estimation process by utilizing the radial basis function interpolation method [23]. The textures were then mapped on the deformed face models. Figures 10(a) shows the generic face model in frontal and side views. Figures 10(b) and (c) show the reconstructed face models. We observe that the scaled and deformed face models can match those in the input video sequences. Both the input face images and the 3D face models in the (1,5,9,13,17)-th frames from these two video sequences are shown in Figures 11 and 12 from left to right respectively. The computation times of these two experiments are about 0.94 seconds and 1.03 seconds. In these two experiments, the estimated values of θ_x, θ_y, θ_z, t_x and t_y agree with conceived values and they are also very smooth in the whole sequence. However, the recovered translation values t_z in the whole sequence wobbles slightly. That may imply that the estimated values of t_z are less accurate.



(a) (b) (c)

Figure 10: (a) Generic face model; (b) The scaled and deformed face model from the movie "The Matrix" – the recovered scaling factors $\mathbf{S}'_x$=1.028, $\mathbf{S}'_y$=1.245, and $\mathbf{S}'_z$=1.187; (c) The scaled and deformed face model from the movie "Roman Holiday" – the recovered scaling factors $\mathbf{S}'_x$=1.141, $\mathbf{S}'_y$=1.020, and $\mathbf{S}'_z$=1.019.

7. Conclusions

In this paper, we proposed an efficient method to estimate the pose and a face model of a human head in each frame by using a three-layer linear iterative process. Experimental results on synthetic data with noise and two real video sequences show that the proposed method is efficient and able to provide estimates for the pose and a face model of good accuracy in most conditions.

Acknowledgements

The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK4167/01E).

References

[1] F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *Int. Journal of Comp. Vision*, 50(2):143–169, 2002.

[2] Y. Shan, Z. Liu, and Z. Zhang. Model-based bundle adjustment with application to face modeling. In *ICCV'03*, volume 2, pages 10–17, 2001.

[3] Z. Zhang, Z. Liu, D. Adler, M. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *Int. Journal of Comp. Vision*, 58(2):93–119, 2004.

[4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(9):1063–1074, 2003.

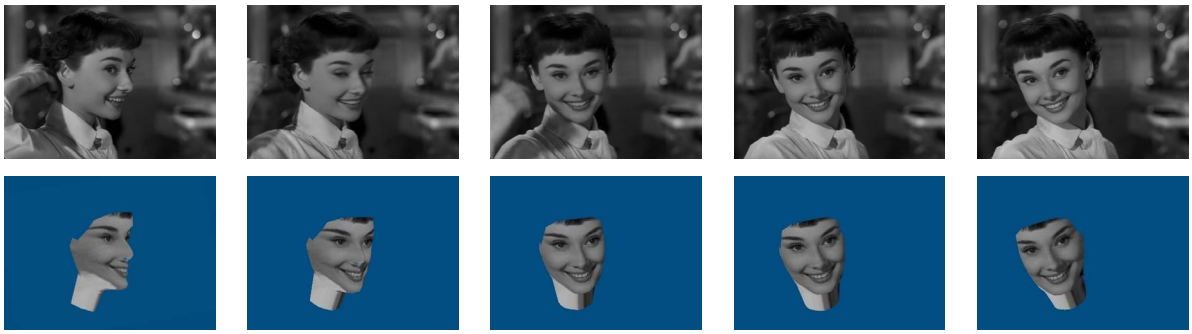


Figure 11: Original 640×464 images extracted from the movie "Roman Holiday" versus deformed face models with map textures.

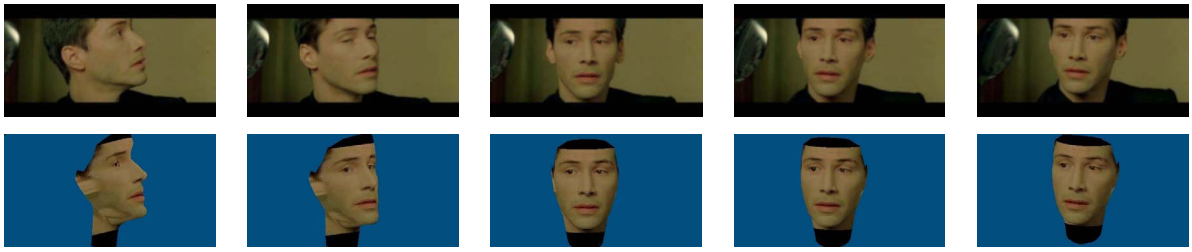


Figure 12: Original 512×272 images extracted from the movie "The Matrix" versus deformed face models with map textures.

[5] R.-L. Hsu, M. Abdel-Mottaleb, and A. Jain. Face modeling for recognition. In *ICIP'01*, volume 2, pages 693–696, 2001.

[6] B. Braathen, M. Bartlett, G. Littlewort, E. Smith, and J. Movellan. An approach to automatic recognition of spontaneous facial actions. In *Proc. of the Fifth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 345–350, 2002.

[7] G. Loy, R. Goecke, S. Rougeaux, and A. Zelinsky. 3d head tracker for an automatic lipreading system. In *Proc. of Australia Conf. on Robitics & Automation (ACRA)*, Melbourne, Australia, 2000.

[8] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(5):441–450, 1991.

[9] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Trans. Syst., Man, Cybern.*, 19(6):1426–1446, 1989.

[10] Y. Hel-Or and M. Werman. Pose estimation by fusing noisy data of different dimensions. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(2):195–201, 1995.

[11] K. Choi, M. Carcassoni, and E. Hancock. Recovering facial pose with the em algorithm. *Pattern Recognition*, 35:2073–2093, 1995.

[12] A. Ansar and K. Daniilidis. Linear pose estimation from points or liness. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):578–589, 2003.

[13] C.-P. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(6):610–622, 2000.

[14] S. Or, W. Luk, K. Wong, and I. King. An efficient iterative pose estimation algorithm. *Image and Vision Comp.*, 16:353–362, 1998.

[15] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(4):322–336, 2000.

[16] J. Xiao, T. Kanade, and J. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. In *Proc. of the Fifth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 156–162, 2002.

[17] K.-Y. Wong and R. Cipolla. Structure and motion from silhouettes. In *ICCV'01*, volume 2, pages 217–222, 2001.

[18] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Silhouette-based 3d face shape recovery. *Graphics Interface*, 2(9):61–68, 2003.

[19] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju. Model-based 3d face capture with shape-from-silhouettes. In *IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, pages 20–27, 2003.

[20] R. Verma, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(10):1215–1228, 2003.

[21] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int. Journal of Comp. Vision*, 38(2):99–127, 2000.

[22] F. Parke. Parameterized models for facial animation. *IEEE Comp. Graphics*, 2(9):61–68, 1982.

[23] R. Schaback. Creating surfaces from scattered data using radial basis functions, in: M. Daelhen, T. Lyche, L.L. Shumaker (Eds.), *Mathematical methods in CAGD III*, pages 1–21, 1995.