PERGAMON

Pattern Recognition 35 (2002) 1545–1557

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

# 3D head tracking under partial occlusion

Ye Zhang *,[1], Chandra Kambhamettu

*Video/Image Modeling and Synthesis (VIMS) Lab, Department of Computer and Information Sciences, University of Delaware, Newark, Delaware, 19716, USA*

Received 1 December 2000; accepted 3 May 2001

**Abstract**

A new algorithm for 3D head tracking under partial occlusion from 2D monocular image sequences is proposed. The extended superquadric (ESQ) is used to generate a geometric 3D face model in order to reduce the shape ambiguity during tracking. Optical flow is then regularized by this model to estimate the 3D rigid motion. To deal with occlusion, a new motion segmentation algorithm using motion residual error analysis is developed. The occluded areas are successfully detected and discarded as noise. Furthermore, accumulation error is heavily reduced by a new post-regularization process based on edge flow. This makes the algorithm more stable over long image sequences. The algorithm is applied to both synthetic occlusion sequence and real image sequences. Comparisons with the ground truth indicate that our method is effective and is not sensitive to occlusion during head tracking. 

*Keywords:* 3D Head tracking; Face model; Occlusion detection; Motion estimation

## 1. Introduction

The estimation of 3D head rigid motion is crucial in many face related applications such as expression analysis, lip motion analysis, face recognition, etc. With appropriate compensation of head rigid motion, face non-rigid motion analysis and recognition are more accurate and stable. 3D head position also reflects human attention, thus providing important cues for natural user interfaces in interactive environments. Furthermore, head tracking is useful for accurately determining model-based facial image coding parameters (e.g., MPEG-4 FAPs), which are very important in low-bandwidth teleconferencing. Numerous applications call for an unrestricted and robust head tracking system from 2D monocular image sequences.

In this paper, we propose a novel algorithm which can robustly track 3D head position from 2D monocular image sequences. In particular, our algorithm focuses on how to track 3D head position when the head is partially occluded in the input 2D image sequence and how to make the tracking algorithm robust in the presence of both large head motion and occlusion. First, we construct a generic face model by using the extended superquadric (ESQ). Compared with simple geometric models (e.g., ellipsoid), the ESQ face model achieves better approximation of facial shapes. It reduces shape ambiguities while keeping the advantages of simple geometric models. Second, we use this model to regularize optical flow field. Residual error analysis is used to detect the occluded areas. The regularization process is then carried out only on the unoccluded areas. To improve robustness, we further utilize the image information by applying a post-regularization process based on edge flow.

Compared with previous work in 3D head tracking, the greatest advantage of our algorithm is the ability to robustly track head under partial occlusion. Since

* Corresponding author. Tel.: +1-302-831-8235; fax: +1-302-831-8458.
*E-mail address:* zhangye@cis.udel.edu (Y. Zhang).
[1] http://www.cis.udel.edu/ ~vims

0031-3203/02/$22.00 
PII: S0031-3203(01)00140-6

occlusion is very common in practice, our algorithm is less restrictive. Our algorithm was first presented in Ref. [1]. In this paper, we fully describe the formulation and report our recent extensive experimental evaluation on real image sequences.

### 1.1. Previous work

In recent years considerable progress has been made on the problem of head/face tracking from 2D monocular image sequences. Some systems extract the 2D position of the head [2–6], while others retrieve the 3D motion parameters [7–15]. In 3D head tracking, some approaches do not use any face model, such as the work done by Azarbeyajani et al. [10] and Jebara et al. [8]. They determined 3D head position through salient facial feature tracking. The feature trajectories were processed by extended kalman filter (EKF) to recover the 3D structure, camera geometry and facial pose. The recovered 3D structure was further constrained by parameterized models (eigenheads). However, their methods experienced difficulties when the tracked points were not visible over the entire image sequence. Also, these methods are very sensitive to noise. An alternative approach for head tracking uses an explicit face model. Li et al. [7] used an affine model to describe both rigid and non-rigid facial motion. A parameterized face model, Candide model, is used to provide further constraints on motion parameters. Their approach was characterized by a render-feedback loop connecting computer vision and computer graphics. The recovered affine parameters were used in model-based facial image coding. The methodology developed by Black et al. [9] is more stable than the previous work described above. It tracked rigid head motion by using a planar model to intepret optical flow. But the use of a planar model limited the amount of motion that can be tracked by their system. To extract relatively large 3D motions over extended image sequences, Basu et al. [13] used a full 3D rigid model (ellipsoid) to regularize the optical flow. More recently, DeCarlo et al. [11] designed a more sophisticated deformable model and integrated it with optical flow for both motion and shape estimation. Tao [14] proposed a special face model (PBVD) to track the head motion. Cascia et al. [12] modeled the head as a texture mapped cylinder and formulated the head tracking problem as image registration in the texture map. Cascia's system also dealt with varying illumination by using a set of trained illumination templates.

Some of the above systems achieve very good results. Some are also surprisingly efficient. However, few studies have been done to robustly track the head under partial occlusion. Furthermore, since many head tracking systems are based on the minimization of the sum of squared differences (SSD), the accumulation error can be serious for long image sequences with large motion and occlusion.

### 1.2. Our approach

Our algorithm uses a special explicit face model with a closed-form formula—an ESQ face model. We first warp the ESQ face model onto the first frame of the image sequence. Then we compute an optimal set of rigid motion parameters between two successive frames based on the optical flow. The residual error field is used to find the occluded areas. The optimization process is carried on recursively and occluded areas are continuously discarded by the algorithm. To improve robustness, the final results are further adjusted by using edge flow.

The method we describe here relates to the work of Basu et al. [13], where optical flow is employed to constrain a rigid 3D surface model by minimizing motion residual error. But our emphasis is very different in that we focus on robustly tracking head under partial occlusion. Our method includes the following novel features:

(1) The system is built to robustly track the head under partial occlusion. Occluded areas are detected by a new motion segmentation algorithm which is integrated within the head motion estimation algorithm.
(2) A new post-regularization method based on edge flow is designed to reduce the accumulation error.
(3) A novel geometric face model is developed based on the extended superquadric (ESQ). It reduces the shape ambiguity while keeping all the advantages (e.g. closed-form formula representation) of simple geometric models.

In this paper, we first introduce the ESQ face model in Section 2.1. After briefly discussing the rigid motion formulation in Section 2.2, we then describe our integrated motion estimation and occlusion detection algorithm in Section 2.3. To cope with the error accumulation, we introduce a post-regularization strategy using edge flow in Section 2.4. In Section 3, we present our experiments on both synthetic and real sequences to show that our system is not sensitive to occlusion and works robustly over long image sequences. Finally, the conclusions and future plans are discussed in Section 4.

## 2. Framework

Our system formulates head tracking as a model-based least-squares problem (similar to Ref. [13]). Extended algorithms are developed to detect occlusion and reduce the accumulation error. The flow diagram of our system is illustrated in Fig. 1. Each component of our system is discussed in the rest of this section.

### 2.1. ESQ face model

There are many ways of modeling a face. In head tracking systems, simple geometric models are pre-

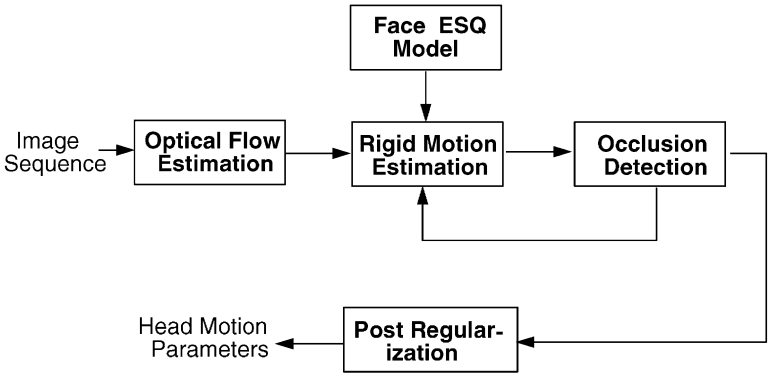


Fig. 1. Flow diagram of the system.

ferred for rigid motion. Ellipsoid [13] and cylinder [12] models have been successfully applied before. However, those models cannot achieve good approximation to face shape. Since the shape ambiguity is one of the reasons for the motion ambiguity, it is desirable for a face model to achieve better approximation while keeping the advantages of simple geometric models. This is our motivation to choose the ESQ to model a face. Another benefit is that by using the ESQ model, we know the position of important features on the face once the tracking is complete. For example, the 3D position of the nose is known via the ESQ face model, which can be used to locate the other 3D MPEG-4 facial definition parameters (FDPs).

Superquadrics [16] are widely used in geometric modeling because of the following advantages: they can model a diverse set of objects, they provide compact representation and they are robust in recovery of 3D models. However, their intrinsical symmetry becomes a problem in modeling many real-world objects. Zhou et al. [17] extended superquadrics with exponent functions, thus improving their ability to model more complex objects including human faces. Essentially, the extended superquadric (ESQ) provides us an economic way to reduce shape ambiguity while keeping all the advantages of simple geometric models.

An extended superquadric can be defined as a set of points $\boldsymbol{X} = [x, y, z]^{\mathrm{T}}$ satisfying:

$$\boldsymbol{X} = \begin{bmatrix} a_1 \mathrm{sign}(c_{\theta_s} c_{\phi_s}) ||c_{\theta_s}||^{f_2(\theta_p)} ||c_{\phi_s}||^{f_1(\phi_p)} \\ a_2 \mathrm{sign}(s_{\phi_s}) ||s_{\phi_s}||^{f_1(\phi_p)} \\ a_3 \mathrm{sign}(s_{\theta_s} c_{\phi_s}) ||s_{\theta_s}||^{f_2(\theta_p)} ||c_{\phi_s}||^{f_1(\phi_p)} \end{bmatrix}, \quad (1)$$

where $\cos(\theta_s), \sin(\phi_s)$, etc. have been abbreviated as $c_{\theta_s}$, $s_{\phi_s}$, etc. The exponents $f_1(\phi_p), f_2(\theta_p)$ are functions of $\phi_p$ and $\theta_p$. $\theta_p, \phi_p$ represent the latitude and longitude angles, respectively, in the spherical coordinate system, and $\theta_s, \phi_s$ represent the superquadric angles. Thus we have

$$\theta_p = \arctan\left(\frac{x}{y}\right),$$

$$\phi_p = \arctan\left(\frac{z}{x}\right). \quad (2)$$

From Eqs. (1) and (2), a parameterized ESQ surface $\boldsymbol{X}(\theta_p, \phi_p)$ can be easily computed from the latitude and longitude angles. This representation greatly increases the controllability of the face model and makes some implementation tasks such as sampling very easy.

In our system, the ESQ face model is constructed by semi-automatically fitting an ESQ to generic face range data. First, we need to define an error-of-fit function to measure the difference between a modeled shape and the face data set [17]. Then the fitting becomes a procedure to find a model that minimizes the error-of-fit function. For extended superquadrics, there is no closed-form error-of-fit function based on the true Euclidean distance. However, we can easily define the error-of-fit function as

$$EOF = \sum_{i=1}^{N \text{ data}} \left[1 - F(x_i, y_i, z_i)^{f_1(\arctan(z_i/\sqrt{x_i^2+y_i^2}))}\right]^2, \quad (3)$$

where $F(x, y, z)$ is an inside–outside function which can be defined by using the implicit representation of Eq. (1). During fitting, the exponent functions are first set as initial Bezier curves with two control points. Then $EOF$ is minimized and $F(x, y, z)$ is computed by using the Levenberg–Marquardt non-linear optimization method. This process continues recursively and each time a new Bezier control point is added to the exponent functions. The optimal ESQ face model (as shown in Fig. 2) is obtained when a specific error threshold is reached. More details on ESQ fitting can be found in Ref. [17].

At initialization, the ESQ model is scaled and warped to the face image in the first frame of the image sequence. Assuming that the first frame is a frontal view, we can

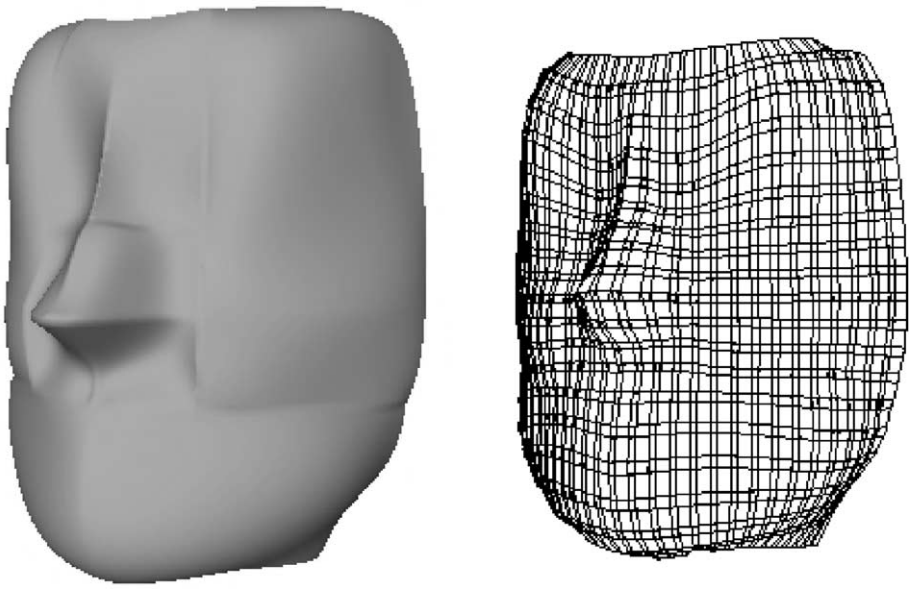


Fig. 2. ESQ face model: shaded and mesh representations. (a) Shaded face ESQ model. (b) Mesh face ESQ model.

do the initialization automatically by using a feature extraction algorithm (e.g. Ref. [13]).

### 2.2. Rigid motion formulation

The rigid head motion for each frame $t$ with respect to frame 0 is represented as a vector $\boldsymbol{m}_t$ with six elements:

$$\boldsymbol{m}_t = [r_x, r_y, r_z, t_x, t_y, t_z], \tag{4}$$

where $r_x, r_y, r_z$, respectively, represent the rotations about the $x, y$, and $z$ axes of the local coordinate frame of the ESQ face model. Accordingly, $t_x, t_y, t_z$ represent the translations of the model. The $4 \times 4$ homogeneous transformation matrix $\boldsymbol{M}_t$ is defined as

$$\boldsymbol{M}_t = \boldsymbol{T}\boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z, \tag{5}$$

where $\boldsymbol{R}_x, \boldsymbol{R}_y, \boldsymbol{R}_z$ are rotation matrices corresponding to $r_x, r_y, r_z$. $\boldsymbol{T}$ is the translation matrix. In frame $t$, the face model can be computed as

$$\boldsymbol{X}(\theta_p, \phi_p, t) = \boldsymbol{M}_t\boldsymbol{X}(\theta_p, \phi_p, 0). \tag{6}$$

Though we are using 3D face model, the image sequence is in 2D. So we must project the parameterized ESQ face surface onto the image plane. The projection matrix $P$ can be given as follows:

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix}, \tag{7}$$

where $f$ is the focal length of the camera which has been given. Thus in frame $t$, a model point's projective location $(I_x, I_y)$ on the image plane can be obtained by computing

$$[x', y', w']^{\mathrm{T}} = \boldsymbol{P}\boldsymbol{M}_t\boldsymbol{X}(\theta_p, \phi_p, 0), \tag{8}$$

where $(I_x, I_y) = (x'/w', y'/w')$.

Now we can easily compute the model displacement $\boldsymbol{D}_M = [U_m, V_M]$ on the image place between frames $t$ and $t+1$:

$$[U'_M, V'_M, W'_M]^{\mathrm{T}} = \boldsymbol{P}(\boldsymbol{M}_{t+1} - \boldsymbol{M}_t)\boldsymbol{X}(\theta_p, \phi_p, 0), \tag{9}$$

where $\boldsymbol{D}_M = [U'_M/W'_M, V'_M/W'_M]$.

Since we only have 2D information, only the points which are visible in both frames $t$ and $t+1$ are responsible for the rigid motion estimation. Given the camera position and point normal $\boldsymbol{N}$, we can estimate whether a point $\boldsymbol{X}$ is self-occluded or not by computing the following dot product:

$$v = (\boldsymbol{X} - \boldsymbol{C})N, \tag{10}$$

where $\boldsymbol{C}$ is the camera position vector. Note that we can assume majority of the face is convex. Therefore, if $v \geqslant 0$, the point is self-occluded, otherwise it is not.

### 2.3. Rigid motion estimation under partial occlusion

Head tracking is usually formulated as a model-based SSD problem. Our system follows this approach. Optical flow at image points that correspond to the visible part

of the face model is used to guide the model motion estimation. Obviously if the occluded points are projected onto the image plane, their optical flow cannot reflect the correct 3D head motion. Self-occluded points can be found by Eq. (10). However, in a real video stream, there are many occasions where human heads are occluded by other objects. To be unrestricted and stable, a head tracking system should be able to locate these occlusion areas and discard them as noise. Our algorithm integrates the head motion estimation and occlusion detection through motion residual error analysis.

In our system, it is assumed that the occluded areas are not too large (generally less than 50% of the face area). This is to ensure that we have sufficient information for correct 3D rigid motion estimation. Theoretically speaking, since no model can perfectly describe every detail of the face, there always exist motion ambiguities due to shape ambiguities. However, if the occlusion areas are not too large, we can still get fairly stable results.

Given the 3D motion vector $\boldsymbol{m}_t$ of frame $t$ and the optical flow between frames $t$ and $t+1$, we must measure how good a candidate motion vector $\boldsymbol{m}_{t+1}$ is for frame $t+1$. We define this measurement on a set of visible (neither self-occluded nor occluded by other objects) points $\dot{\boldsymbol{V}}$ in both frames $t$ and $t+1$. $\dot{\boldsymbol{V}}$ is a subset of sample points set $\boldsymbol{Q}$ on the ESQ face model.

If the optical flow field is represented by $\boldsymbol{D}_o = [U_o, V_o]$, the error-of-fit (EOF) function is then defined as follows:

$$d_x = ||\boldsymbol{D}_M - \boldsymbol{D}_o||^2,$$

$$e_x = \begin{cases} d_x & \text{if } d_x < d_t, \\ d_t & \text{if } d_x \geqslant d_t, \end{cases}$$

$$EOF(\dot{\boldsymbol{V}}, \boldsymbol{m}_t, \boldsymbol{m}_{t+1}, \boldsymbol{D}_o) = \frac{1}{N} \sum_{x \in \dot{V}} e_x, \tag{11}$$

where $N$ is the number of points in set $\dot{\boldsymbol{V}}$ and $d_t$ is the error threshold which is used to prevent outliers in the optical flow field from overwhelming the whole algorithm.

The question is how to determine the points in $\dot{\boldsymbol{V}}$ under occlusion. If the non-self-occluded subsets of $\boldsymbol{Q}$ in frames $t$ and $t+1$ are represented by $\boldsymbol{V}_t$ and $\boldsymbol{V}_{t+1}$, respectively, we can initially let $\dot{\boldsymbol{V}} = \boldsymbol{V}_t \cap \boldsymbol{V}_{t+1}$, then minimize Eq. (11) to find an optimal $\boldsymbol{m}^*_{t+1}$. We believe that the motion residual errors for points in the occluded areas are bigger due to the following two reasons: (1) The unoccluded areas are larger, thus contributing more to the minimization. (2) The motion field on occluding objects cannot be well regularized because the occluding objects are normally not of the same shape as the face model. Based on these observations, our integrated motion estimation and occlusion detection algorithm can be described as shown in Algorithm 1.

In our system, we use $\boldsymbol{m}_{t+1} = \boldsymbol{m}_t$ as an initial guess. The Levenberg–Marquardt algorithm is used to solve the equation in Step 4a of Algorithm 1. During minimization, penalties are also added on very large motion candidates. The penalty term can be represented as

$$P(\boldsymbol{m}_t, \boldsymbol{m}_{t+1}) = \alpha \max(||\boldsymbol{m}_{t+1} - \boldsymbol{m}_t|| - \Gamma, 0.0),\ \alpha, \Gamma > 0, \tag{12}$$

where $\alpha$ is a positive constant and $\Gamma$ is the possible maximal motion. The values of $\alpha$ and $\Gamma$ are determined empirically. In our experiments, the above algorithm normally converges in 3–4 iterations. Note that, if the 2D motion of the occluding object is very similar to that of the occluded areas, our algorithm may not find the occluded areas. However, in this case the occluded areas do not harm the motion estimation process. The results of our occlusion detection algorithm are illustrated in Figs. 4, 6, 7 and 8.

**Algorithm 1:** Algorithm for 3D Head Tracking under Partial Occlusion

**begin**

1. Sample a set of points $\boldsymbol{Q}$ on the parameterized surface $\boldsymbol{X}(\theta_p, \phi_p)$ and compute their normals.
2. Compute $\boldsymbol{V}_t$ and $\boldsymbol{V}_{t+1}$ from $\boldsymbol{Q}$ according to Eq. (10).
3. Construct a flag vector $\boldsymbol{F}_t = [f_0, f_1, \ldots, f_N]$ corresponding to the points in set $\boldsymbol{V}_t \cap \boldsymbol{V}_{t+1}$. $f_x = 1$ means point $x$ is not occluded while $f_x = 0$ means occluded. Initially, set $\boldsymbol{F}_0 = [1, 1, \ldots, 1]$.
4. **while** (flag vector $\boldsymbol{F}_t$ is changed and maximum iteration number is not exceeded) **do**
   a. Compute $\dot{\boldsymbol{V}}$ by discarding those points whose corresponding $f_x$ is 0 from $\boldsymbol{V}_t \cap \boldsymbol{V}_{t+1}$. Then solve: $\boldsymbol{m}^*_{t+1} = arg(min(EOF(\dot{\boldsymbol{V}}, \boldsymbol{m}_t, \boldsymbol{m}_{t+1}, \boldsymbol{D}_o)))$.
   b. Compute the motion residual error $e_x$ at each point $x$ in set $\boldsymbol{V}_t \cap \boldsymbol{V}_{t+1}$ using $\boldsymbol{m}^*_{t+1}$.
   c. Re-set flag vector $\boldsymbol{F}$ by:
   $$f_x = \begin{cases} 0 & \text{if } e_x > \alpha d_t, \\ 1 & \text{otherwise;} \end{cases}$$
   where $\alpha$ is initially set as 0.9 in our experiments. To prevent the discarding of unoccluded areas, it is adjusted automatically in an adaptive fashion.

**end**

### 2.4. Post-regularization

Inaccurate optical flow estimation and lack of 3D information generate accumulation error in head tracking systems that are based on SSD minimization. Large

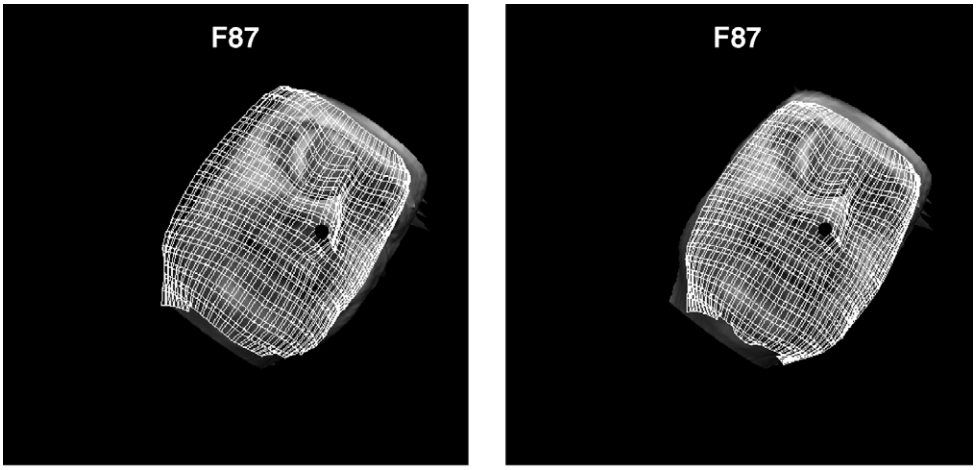


Fig. 3. Tracking results without/with post-regularization. (a) Without post-regularizaiton. (b) With post-regularization.

motion and occlusion make the problem even worse. This means that the tracking algorithm may not be robust over long image sequences. Dealing with accumulation error becomes necessary. One possible solution is to use both image field and motion field information (i.e. edge force and optical flow force) simultaneously to constrain a deformable face model [11]. Our algorithm, however, uses edge information differently.

The idea is to use edge flow as a post-regularization heuristic. After we get $\boldsymbol{m}^*_{t+1}$ from the integrated algorithm introduced in Section 2.3, edge flow is used to refine $\boldsymbol{m}_{t+1}$ in a small neighborhood around $\boldsymbol{m}^*_{t+1}$ until the optimal motion vector is found. We believe that the quality of our edge flow (computed by the following algorithm) is generally more reliable than optical flow because edge flow strictly captures the motion of the salient features (i.e. edge points) while optical flow smoothes out some useful information. Evidence of the effectiveness of the edge information can be found in Ref. [18] where edge matching alone is used to extract the camera motion. We compute edge flow based on this matching technique. The post-regularization algorithm can be described as shown in Algorithm 2.

**Algorithm 2:** Post-regularization algorithm

**begin**

1. Detect edge points in the areas of interest (face areas) in frames $t$ and $t+1$.
2. Detect edge features in frame $t$.
3. Detect the perfect matches for the frame $t$ features in frame $t+1$. Finally we find a set of edge features $\boldsymbol{P}$ in frame $t$. Each point $\boldsymbol{E}_t$ in set $\boldsymbol{P}$ has a perfectly matched point $\boldsymbol{E}_{t+1}$ in frame $t+1$.
4. Compute the edge flow $\boldsymbol{D}_E = [U_E, V_E]$ at point $\boldsymbol{E}_t$ as follows:

   $\boldsymbol{D}_E = \boldsymbol{E}_{t+1} - \boldsymbol{E}_t$.
5. Remove the "outliers" in edge flow based on the motion vector $\boldsymbol{m}^*_{t+1}$. The motion residual error at each edge point $\boldsymbol{P}$ is given by:

   $e'_x = ||\boldsymbol{D}_M - \boldsymbol{D}_E||^2$.

   If the error is larger than a threshold, we consider it as outlier and remove it from $\boldsymbol{P}$. The EOF function for post-regularization can be defined as:

   $EOF'(\boldsymbol{P}, \boldsymbol{m}_t, \boldsymbol{m}_{t+1}, \boldsymbol{D}_E) = \frac{1}{N'} \sum_{x \in \boldsymbol{P}} e'_x$.
6. Using $\boldsymbol{m}^*_{t+1}$ as the initial guess, solve

   $\boldsymbol{m}^{**}_{t+1} = arg(min(EOF'(\boldsymbol{P}, \boldsymbol{m}_t, \boldsymbol{m}_{t+1}, \boldsymbol{D}_E)))$.

**end**

In Step 1 of Algorithm 2, the area of interest in frame $t$ is the face model's projective area, while in frame $t+1$ the area of interest should be large enough to cover the face area in frame $t$ and the possible large motion. In Step 2, we refer to an $8 \times 8$ block as an "edge feature" if it contains more than eight edge points. To avoid aperture ambiguity, blocks with strictly vertical and horizontal edges are discarded. In Step 3, two edge features are said to be perfectly matched if they are identical in the binary edge domain. It can be easily seen that the bigger the observing window, the more strict the matching criterion is. For those features where we could not find the perfect matches we simply discard them. The reason we remove "outliers" in Step 5 is that we believe

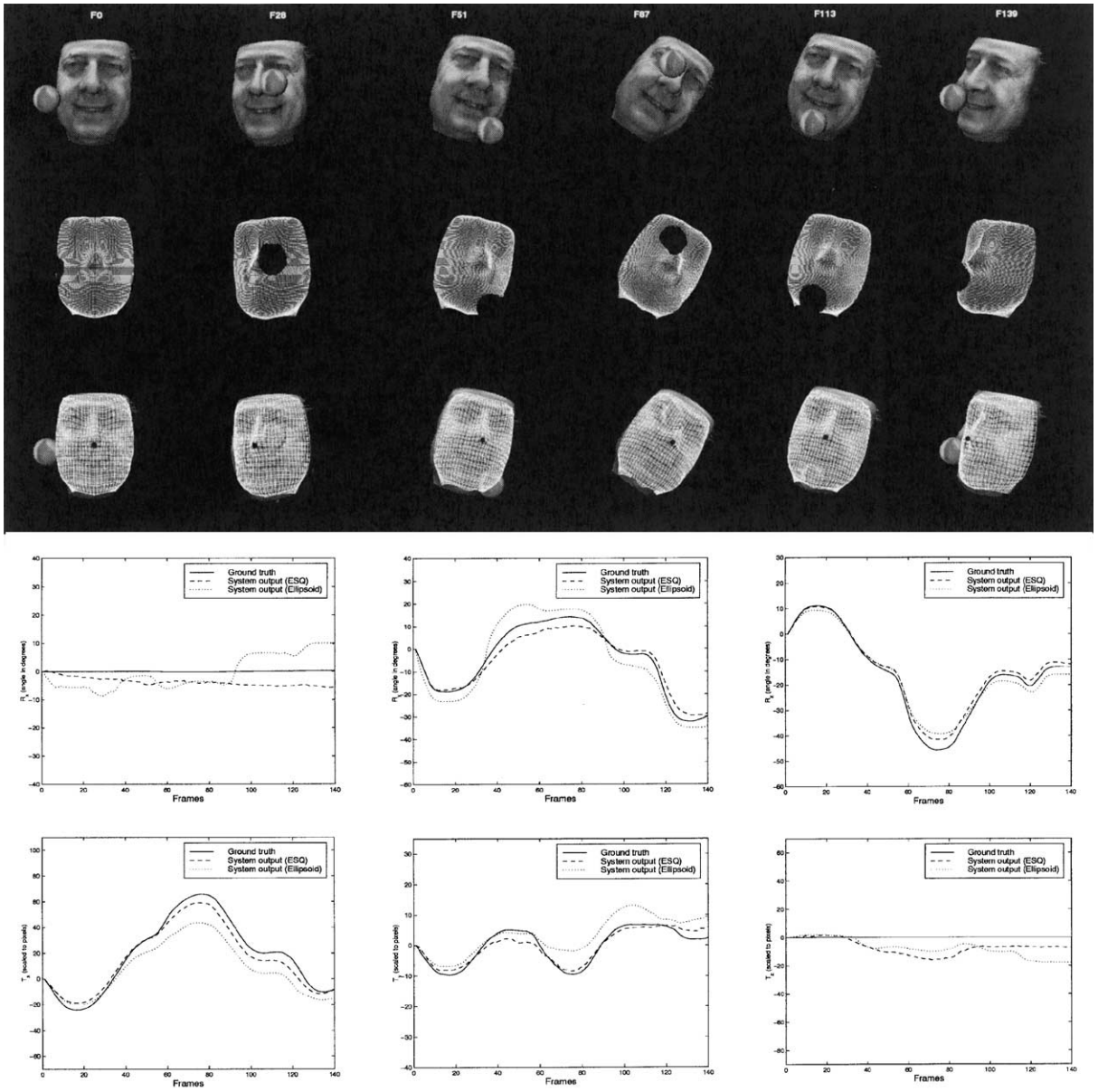


Fig. 4. Experimental results on the synthetic image sequence. The first row includes some key frames of the original synthetic sequence. The second row indicates the occlusion detection results. The third row shows the tracking results (black dots indicate the nose position in each frame). The graphs show the ground truth validation.

that we have found a reasonably good solution before post-regularization (Algorithm 1). It is not possible to get a very large error on any edge point unless the edge flow is wrong. Essentially, the post-regularization process tries to find an optimal solution $\boldsymbol{m}_{t+1}^{**}$ in a small neighborhood around $\boldsymbol{m}_{t+1}^{*}$ to minimize $EOF'$ based on image information.

Our experiments have shown that Algorithm 2 heavily reduces the accumulation error. The tracking results of our system without/with post-regularization are illustrated in Fig. 3. We can see that the tracking result on frame 87 without post-regularization has a larger accumulation error when compared with the result with post-regularization.

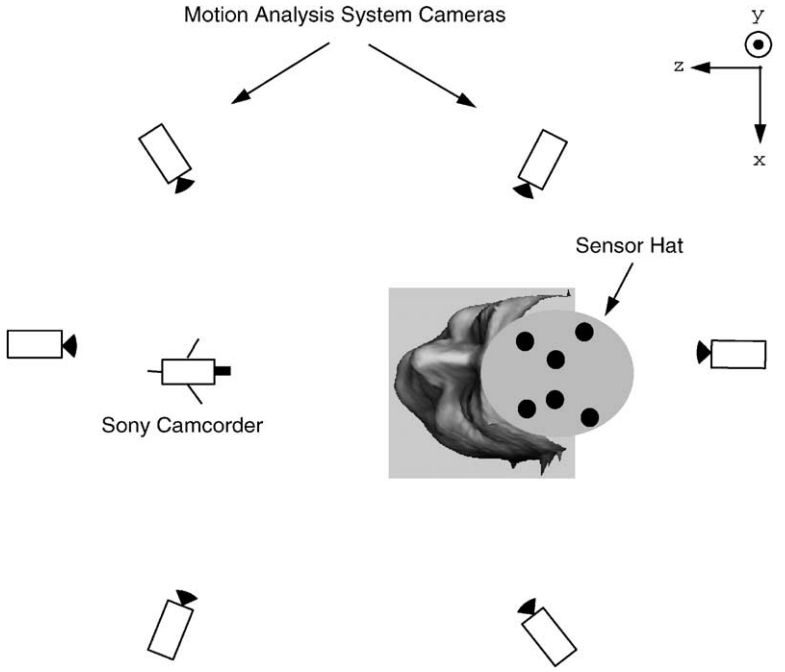


Fig. 5. Experimental setup for collecting real occlusion sequence with ground truth.

## 3. Experiments

To show the accuracy and robustness, our algorithm has been tested on extensive synthetic and real occlusion image sequences. The experimental setup and comparisons with the ground truth are reported in the following sections.

### 3.1. Data acquisition

First, we applied our algorithm on a synthetic occlusion sequence so that we can see how the algorithm behaves under ideal conditions (i.e., with less noise, complete and accurate camera parameters, etc.). The synthetic sequence (Fig. 4) was generated by using a set of known motion vectors to animate real face range data. The occluding object, a sphere undergoing a *sine*-like motion, was added to the scene.

Second, we did experiments on extensive real occlusion sequences to show the applicability of our algorithm in practice. The real image sequences were collected from two different sources: one is a Sony DCR-TRV900 digital camcorder, the other is a Sony EVI-D30 analog camera. When we collected image sequences by using digital camcorder, ground truth data for those sequences were simultaneously collected by using a Motion Analysis EVA HiRes system (Fig. 5). The subjects wore a hat with six retroreflective markers attached on it. The six calibrated cameras of the EVA HiRes system were then used to track the position of the markers in real time (30 fps). The positional accuracy is within 1 mm. The 3D head position is computed accurately from the marker positions. Note that the marker hat has no influence on our algorithm because it does not block any face area.

### 3.2. Evaluation and discussion

We quantitatively evaluated our algorithm on both synthetic and real data. The occlusion detection and tracking results in some key frames of the synthetic sequence are shown in Fig. 4. The black dots in the tracked frames indicate the nose position of the ESQ face model. The graphs in Fig. 4 show the comparisons between the six estimated parameters and the ground truth (note that the motion parameters shown in Ref. [1] correspond to relative motion between two successive frames, while the motion parameters shown in this paper correspond to global motion relative to the first frame). Through the graphs we can see that our system tracked the motion accurately during occlusion. We have also found that the estimation of the translation along the $z$-axis has a slightly larger error, which is due to lack of 3D information.

In the synthetic experiment, we also substituted the ESQ face model with a simple ellipsoid model. We found that the ESQ face model outperformed the ellipsoid model (Fig. 4), demonstrating the effectiveness of using the ESQ face model. The better performance of

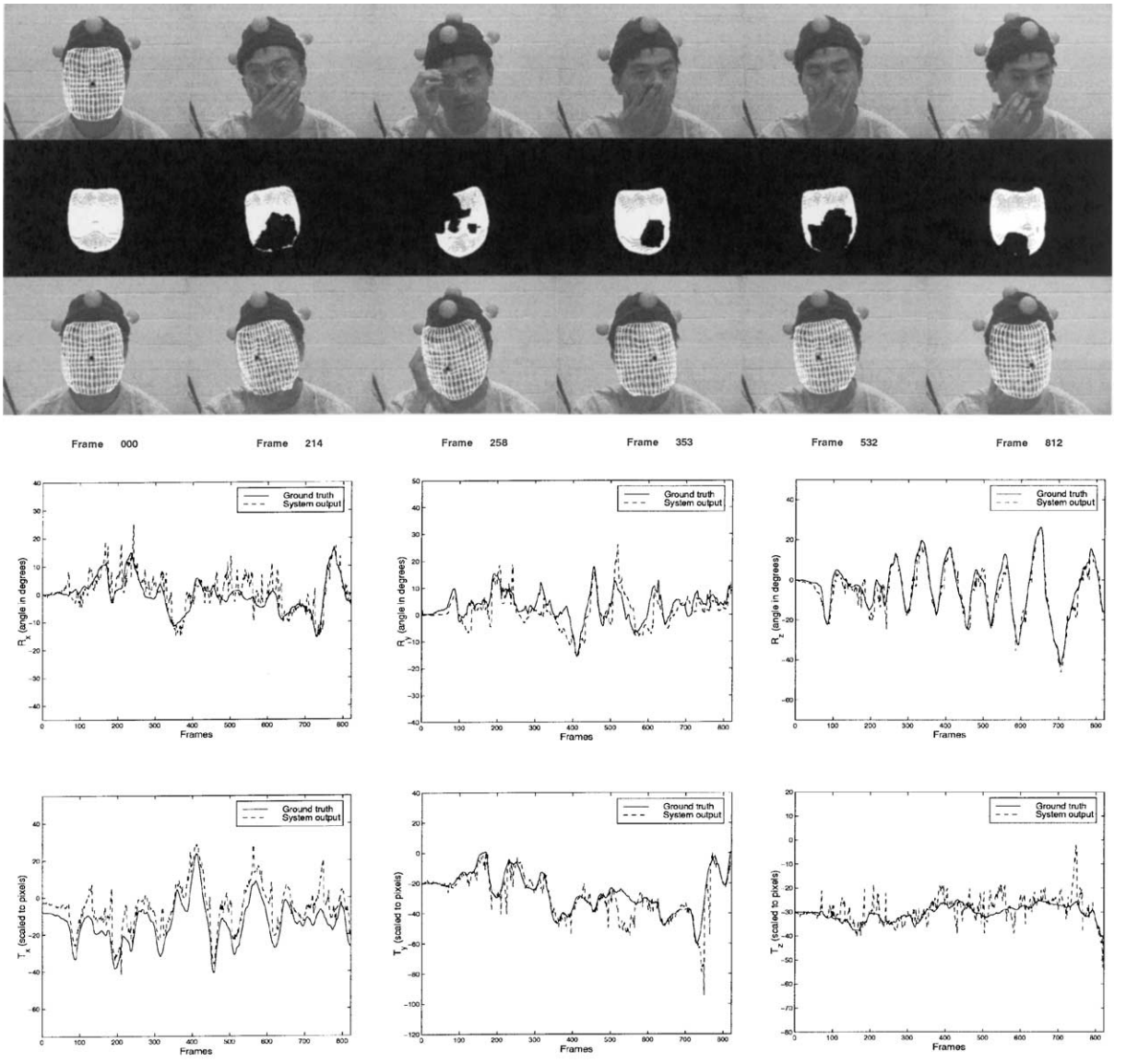


Fig. 6. Experimental results on real image sequence. The image sequence was collected by using a Sony digital camcorder. The graphs show the ground truth validation. Black dots indicate the nose position in each frame.

the ESQ face model is due to its better approximation of human facial shapes.

The tracking results and validation of our algorithm on two real image sequences collected by using the digital camcorder are shown in Figs. 6 and 7. In Fig. 6, the subject freely did some natural movements which occluded the face. The image sequence is 822 frames long (approximately 27 s). In Fig. 7, the subject was told to do not only some natural occluding movements, but also some non-rigid expressions (surprise, smile, etc.). The image sequence is 800 frames long (approximately 26 s). The graphs in Figs. 6 and 7 also show the comparisons between the six estimated parameters and the ground truth (note that the motion parameters shown here correspond to global motion relative to the first frame). We also show some tracking results on the image sequences collected by using the analog camera in Fig. 8. Our various experimental results show that the proposed algorithm is able to robustly track long occlusion image sequences. The average time to find the 3D head position for one frame

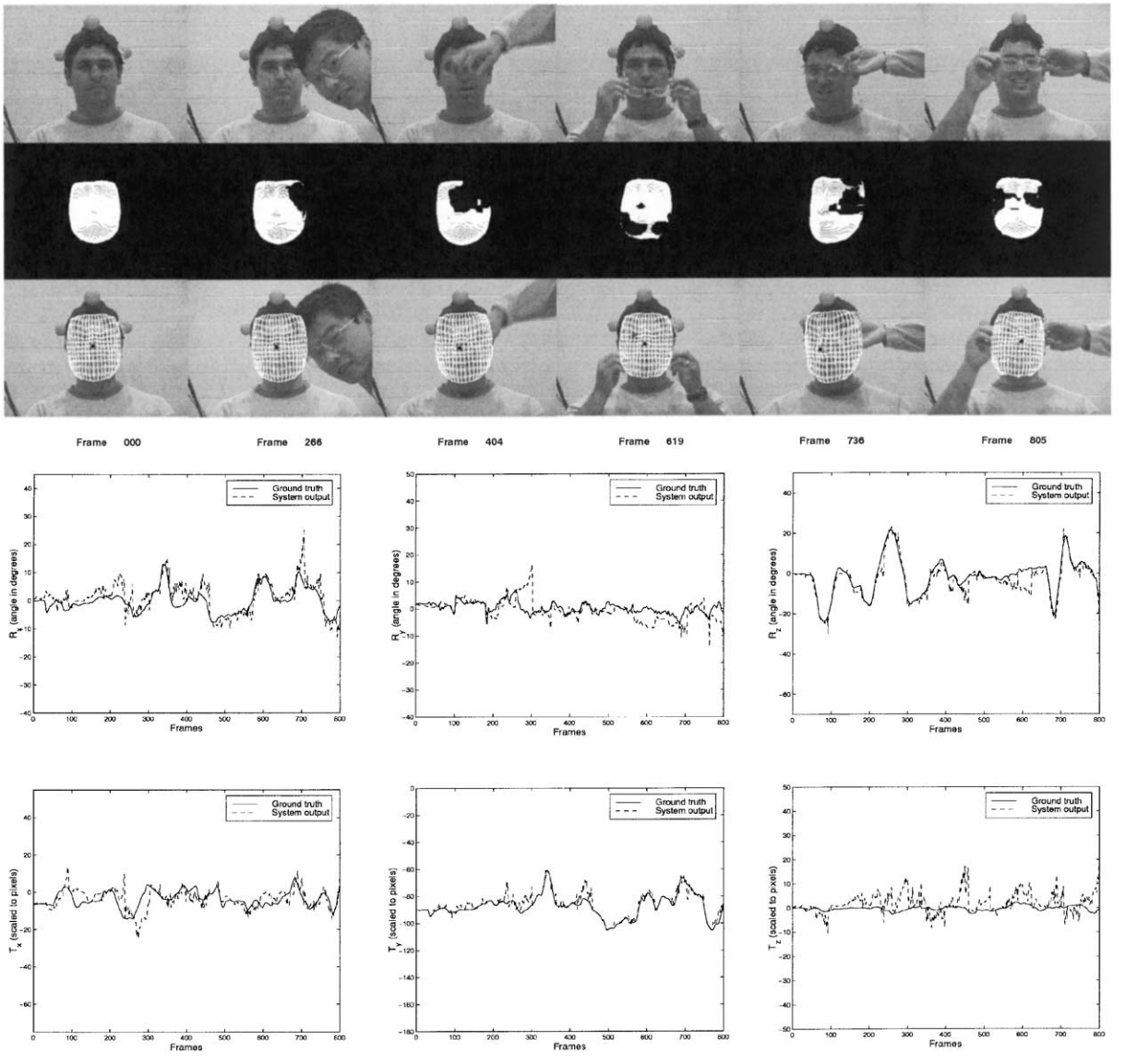


Fig. 7. Experimental results on real image sequence. The image sequence was collected by using Sony digital camcorder. The graphs show the ground truth validation. Black dots indicate the nose position in each frame.

is around 50–60 s on a 195 MHz R10000 SGI Octane. Most of the computational burden is due to optical flow computation and recursive optimization.

Note that due to lack of 3D information, head tracking algorithms always experience difficulties when large rotation or $z$ translation exist, let alone when occlusion exists. One of the most important advantages of our algorithm is that it can robustly track the head even when there exist both occlusion and relatively large rotation or $z$ translation in the image sequences. This is achieved by using our occlusion detection and post-regularization schemes. However, from the ground truth evaluation on both synthetic and real image sequences (Figs. 4, 6 and 7), we can still see that the larger errors happen to depth-sensitive parameters (i.e., $R_y, R_x, T_z$). If we have multiview image sequences, or if we utilize some depth heuristics (e.g., structure from X), we may achieve better results. Another possible solution is to incorporate the $z$ scaling factor of the ESQ face model into the motion estimation framework. This makes the system more stable. Generally, the

Fig. 8. Experimental results on two real image sequences. The image sequences were collected by using a Sony analog camera. Black dots indicate the nose position in each frame. (a) Tracking results of a 266 frame sequence. (b) Tracking results of a 199 frame sequence.

performance of our system degrades with the increase in the occluded area. However, we found that the performance of our system is not very sensitive to the percentage of the occluded area when the percentage is less than a certain threshold. As we can see in Figs. 6 and 7, if the occluded area is not too large, the performance is fairly stable.

In Fig. 6, we can see that the $T_x$ parameter has a constant shift compared with the ground truth. This is an example of initial warping error: when we warp the ESQ model onto the first frame, we did not find the value of $T_x$ accurately. However, this initial error did not do much harm to our algorithm. Our algorithm is not very sensitive to initial warping errors.

Another note worth mentioning here is that the algorithm is not sensitive to non-rigid motion of the face (Fig. 7). Actually if the non-rigid motion is too large, the algorithm classifies the face areas related to non-rigid motion as occluded areas. This obviously helps to make the algorithm more robust.

## 4. Conclusions and future plans

In this paper, we have presented an algorithm which can robustly track occluded heads from 3D monocular image sequences. We have designed a face model with a closed-form formula based on the ESQ and used it in our system. We have also demonstrated a method that can successfully detect the occluded areas on the face by integrating 3D motion estimation and motion segmentation. Furthermore, we have developed a post-regularization method that heavily reduces the accumulation error incurred by motion ambiguities and occlusion. The experi-

ments in Section 3 clearly show that our system is robust and reliable. It is also possible to apply our occlusion detection and post-regularization algorithms to other tracking systems.

From our experiments, we can see that one potential advantage of the ESQ model is that we can also track the nose position in each frame accurately, thus effectively constraining facial non-rigid motion analysis. This is extremely useful when we perform facial motion tracking to extract the MPEG-4 FAPs from a video sequence in image encoding. Since our algorithm requires optical flow computation and recursive optimization, it is not real-time. However, the ability to robustly track the face under partial occlusion makes our algorithm very useful. Our future directions include improving the efficiency of our algorithm.

## Acknowledgements

Research funding was provided by the National Science Foundation Grants IRI-9619240, CAREER IRI-9984842 and CISE CDA-9703088. The authors would like to thank Lin Zhou (VIMS Lab, U. of Delaware) for his ESQ face model, David Saxe (Biomechanics and Movement Science program, U. of Delaware) for his help in ground truth data acquisition for real image sequences, Pavel Laskov (VIMS Lab, U. of Delaware) for his useful advice, Sumit Basu (Media Lab, MIT) for his head motion data, and Dr. Thomas Huang (U. of Illinois) for Cyberware face data.

## References

[1] Y. Zhang, C. Kambhamettu, Robust 3D head tracking under partial occlusion, Proceedings of Fourth Conference on Automatic Face and Gesture Recognition, 2000, pp. 176–182.

[2] N. Oliver, A.P. Pentland, F. Berard, Lafter: lips and face real time tracker, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 123–129.

[3] Y. Yacoob, L.S. Davis, Computing spatio-temporal representations of human faces, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 70–75.

[4] S.T. Birchfield, Elliptical head tracking using intensity gradients and color histograms, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 232–237.

[5] J.L. Crowley, F. Berard, Multi-model tracking of faces for video communications, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 640–645.

[6] P. Fieguth, D. Terzopoulos, Color based tracking of heads and other mobile objects at video frame rates, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 21–22.

[7] H. Li, P. Roivainen, R. Forchheimer, 3-D motion estimation in model-based facial image coding, IEEE Trans. Pattern Anal. Mach. Intell. 15 (6) (1993) 545–555.

[8] T.S. Jebara, A.P. Pentland, Parametrized structure from motion for 3D adaptive feedback tracking of faces, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 144–150.

[9] M.J. Black, Y. Yacoob, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion, Proceedings of International Conference on Computer Vision, 1995, pp. 374–381.

[10] A. Azarbayejani, T. Starner, B. Horowitz, A.P. Pentland, Visually controlled graphics, IEEE Trans. Pattern Anal. Mach. Intell. 15 (6) (1993) 602–605.

[11] D. DeCarlo, D. Metaxas, The integration of optical flow and deformable models: Applications to human face shape and motion estimation, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 231–238.

[12] M. LaCascia, S. Sclaroff, V. Athitsos, Fast, reliable head tracking under varying illumination: an approach based on registration of textured-mapped 3D models, IEEE Trans. Pattern Anal. Mach. Intell. 22 (4) (2000) 322–336.

[13] S. Basu, I.A. Essa, A.P. Pentland, Motion regularization for model-based head tracking, Proceedings of International Conference on Pattern Recognition, 1996, pp. C8A.3.

[14] H. Tao, Nonrigid Motion Modeling and Analysis in Video Sequences for Realistic Facial Animation. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Department of Electrical and Computer Engineering, 1999.

[15] A. Schődl, A. Haro, I. Essa, Head tracking using a textured polygonal model, Perceptual User Interfaces Workshop, 1998.

[16] D. Terzopoulos, D. Metaxas, Dynamic 3D models with local and global deformations: deformable superquadrics, IEEE Trans. Pattern Anal. Mach. Intell. 13 (7) (1991) 703–714.

[17] L. Zhou, C. Kambhamettu, Extending superquadrics with exponent functions: modeling and reconstruction, Graph. Models 63 (1) (2001) 1–20.

[18] A. Zakhor, F. Lari, Edge-based 3-D camera motion estimation with application to video coding, IEEE Trans. Image Process. 2 (4) (1994) 481–498.

**About the Author**—YE ZHANG received his B.Eng. and M. Eng. degrees in Electronics Engineering from Sichuan University, P.R. China, in 1995 and 1998, respectively. He also received as M.S. degree in Computer and Information Sciences from University of Delaware, USA, in 1999. He is currently a Ph.D. candidate in Video/Image Modeling and Synthesis (VIMS) Lab at University of Delaware. Ye Zhang is a winner of "University Competitive Fellowship Award" from 2000–2001 and "Quantaum Leap Excellence in Artificial Intelligence Award" in 2001. His research interests include Computer Vision, Computer Graphics, Image Processing.

**About the Author**—CHANDRA KAMBHAMETTU received his Bachelor of Engineering degree in Computer Science and Engineering from Osmania University (India) in 1989, his MS and Ph.D. in Computer Science and Engineering from the University of South Florida in 1991 and 1994 respectively. He was awarded the "Outstanding Graduate Student Award" by Tau Beta Pi (Florida Gamma) honor society. From 1994–1996, he was a research scientist at NASA-Goddard, where he received the "Excellence in Research Award". From 1997-present, he is an Assistant Professor in the Department of Computer and Information Sciences at the University of Delaware where he leads the Video/Image Modeling and Synthesis (VIMS) group. Dr. Kambhametu received NSF CAREER award in 2000 and is the associate editor of PATTERN RECOGNITION journal. His interests include Computer vision, Image processing, Computer Graphics and Multimedia systems.