Robust 3D Head Tracking Under Partial Occlusion

Ye Zhang and Chandra Kambhamettu
Department of Computer & Information Sciences
University of Delaware
Newark, Delaware 19716
zhangye/chandra@cis.udel.edu

Abstract

This paper describes a novel system for 3D head tracking under partial occlusion from 2D monocular image sequences. In this system, The Extended Superquadric (ESQ) is used to generate a geometric 3D face model in order to reduce the shape ambiguity. Optical flow is then employed with 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 by the system. Also, accumulation error is heavily reduced by a new post-regularization process based on edge flow. This makes the system more stable over long occlusion image sequences. To show the accuracy, the system is applied on a synthetic occlusion sequence and comparisons with the ground truth are reported. To show the robustness, experiments on long occlusion image sequences, including synthetic and real ones, are reported.

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 head rigid motion compensation, 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 determining model-based facial image coding parameters (*e.g.*, MPEG4 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.

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 [9, 11], while others retrieve the 3D motion parameters [6, 8, 5, 3, 1, 4, 7, 6, 2, 10]. In this paper we concentrate on 3D head tracking.

Li *et al.* [8] used an affine model to describe both rigid and non-rigid facial motion. 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. Azarbeyajani *et al.* [1] and Jebara *et al.* [5] 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 (eigen-heads). However, their methods experienced difficulties when the tracked points were not visible over the entire image sequence. Black *et al.* [3] developed a more stable methodology which tracked rigid head motion by using a planar model to interpret 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.* [2] used a full 3D rigid model (ellipsoid) to regularize the optical flow. More recently, DeCarlo *et al.* [4] designed a more sophisticated deformable model and integrated it with optical flow for both motion and shape estimation. Tao [10] proposed a special face model (PBVD) to track the head motion. Cascia *et al.* [6, 7] modeled head as a texture mapped cylinder and formulated the head tacking 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 Contributions

The method we propose here relates to the work of [2], where optical flow is employed to constrain a rigid 3D

surface model by minimizing motion residual error. Our method extends this framework to include 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.* close-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 [2]). Extended algorithms are developed to detect occlusion and reduce the accumulation error. The flow diagram of our system is illustrated in Figure 1. Each component of our system is discussed in the rest of this Section.

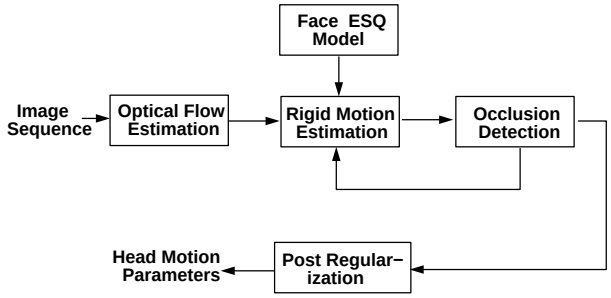


Figure 1. Flow diagram of the system

2.1. ESQ Face Model

There are many ways of modeling a face. In head tracking system, simple geometric models are preferred since we only care about rigid motion. Ellipsoid [2] and cylinder [6, 7] models have been successfully applied in several systems. However, those models can not 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 the simple geometric models. This is our motivation to choose the ESQ to model a face.

Superquadrics can model a diverse set of objects because of their compact representation and robust methods for recovering 3D models. However, their intrinsical symmetry fails in modeling many real-world objects including human faces. Zhou *et al.* [13] extended superquadrics with exponent functions, thus improving their ability to model more complex objects including the face. Essentially, the extended superquadric (ESQ) provides us an economic way to reduce the shape ambiguity while keeping all the advantages of simple geometric models.

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

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

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

$$\theta_p = arctan(\frac{x}{y})$$
$$\phi_p = arctan(\frac{z}{x}) \quad (2)$$

From Eq. 1 and 2, a parameterized ESQ surface $\mathbf{X}(\theta_p, \phi_p)$ 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 (as shown in Fig. 2) is constructed by semi-automatically fitting an ESQ to a set of real face range data. More details on ESQ fitting can be found in [13]. As initialization, this model is scaled and warped on the face image in the first frame of the image sequence. Assuming that the first frame is a frontal view, we can do the initialization automatically by using a feature extraction algorithm (*e.g.* [2]).

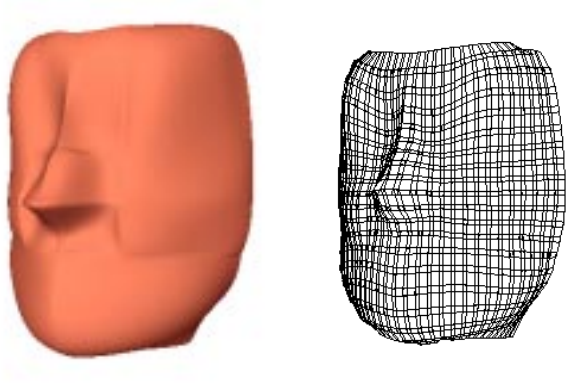
2.2. Rigid Motion Formulation

The rigid head motion in each frame t from frame 0 is represented as a vector $\mathbf{m}_t$ with 6 elements:

$$\mathbf{m}_t = [r_x, r_y, r_z, t_x, t_y, t_z] \quad (3)$$

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 4×4 homogeneous transform matrix $\mathbf{M}_t$ is defined as:

$$\mathbf{M}_t = \mathbf{T}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z \quad (4)$$



(a) Shaded face ESQ model (b) Mesh face ESQ model

Figure 2. ESQ face model

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

$$\mathbf{X}(\theta_p, \phi_p, t) = \mathbf{M}_t \mathbf{X}(\theta_p, \phi_p, 0) \tag{5}$$

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 represented as follows:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \tag{6}$$

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:

$$[x', y', w']^T = \mathbf{P}\mathbf{M}_t \mathbf{X}(\theta_p, \phi_p, 0) \tag{7}$$

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

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

$$[U'_M, V'_M, W'_M]^T = \mathbf{P}(\mathbf{M}_{t+1} - \mathbf{M}_t)\mathbf{X}(\theta_p, \phi_p, 0) \tag{8}$$

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

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 $\mathbf{N}$, we can estimate whether a point $\mathbf{X}$ is self-occluded or not by computing the following dot product:

$$v = (\mathbf{X} - \mathbf{C}) \cdot N \tag{9}$$

where $\mathbf{C}$ is the camera position vector. Note that we can assume majority of a face is convex, therefore if $v \geq 0$, the point is self-occluded, otherwise not.

2.3. Rigid Motion Estimation Under Partial Occlusion

Head tracking is normally formulated as a model-based SSD problem. So is our system. Optical flow at image points which correspond to the visible part of the face model is used to guide the model motion estimation. Obviously if the occluded points are projected to the image plane, their optical flow can not reflect the correct 3D motion. Self-occluded points can be found by Eq. 9. However, in 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 those occlusion areas and discard them as noise. Our algorithm integrates 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%). This is to ensure we have fairly enough information for correct 3D rigid motion estimation. Theoretically speaking, since no model can perfectly describe every detail of the face, there anyway exists 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 $\mathbf{m}_t$ of frame t and the optical flow between frames t and $t+1$, we must measure how good a candidate motion vector $\mathbf{m}_{t+1}$ is for frame $t+1$. We define this measurement on a set of visible (neither self-occluded nor occluded by other objects) points $\dot{\mathbf{V}}$ in both frames t and $t+1$. $\dot{\mathbf{V}}$ is a subset of a sample points set $\mathbf{Q}$ on the ESQ face model.

If the optical flow field is represented by $\mathbf{D}_O = [U_O, V_O]$, the Error-of-Fit (EOF) function is then defined as follows:

$$\begin{aligned} d_x &= |s\mathbf{D}_M - \mathbf{D}_O|^2 \\ e_x &= \begin{cases} d_x & if \;\; d_x < d_t; \\ d_t & if \;\; d_x \geq d_t; \end{cases} \\ EOF(\dot{\mathbf{V}}, \mathbf{m}_t, \mathbf{m}_{t+1}, \mathbf{D}_O) &= \tfrac{1}{N} \textstyle\sum_{x \in \dot{V}} e_x \end{aligned} \tag{10}$$

where s is a scaling factor between object coordinates and screen coordinates. N is the number of points in set $\dot{\mathbf{V}}$. 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 $\dot{\mathbf{V}}$ under occlusion. If the non-self-occluded subsets of $\mathbf{Q}$ in frames t and $t+1$ are represented by $\mathbf{V}_t$ and $\mathbf{V}_{t+1}$ respectively, we can initially let $\dot{\mathbf{V}} = \mathbf{V}_t \bigcap \mathbf{V}_{t+1}$, then minimize Eq. 10 to find an optimal $\mathbf{m}^*_{t+1}$. We believe 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 can not 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 follows:

1. Sample a set of points $\mathbf{Q}$ on the parameterized surface $\mathbf{X}(\theta_p, \phi_p)$ and compute their normals;

2. Compute $\mathbf{V}_t$ and $\mathbf{V}_{t+1}$ from $\mathbf{Q}$ according to Eq. 9;

3. Construct a flag vector $\mathbf{F}_t = [f_0, f_1, \ldots, f_N]$ corresponding to the points in set $\mathbf{V}_t \bigcap \mathbf{V}_{t+1}$. $f_x = 1$ means point x is not occluded while $f_x = 0$ means occluded. Initially, set $\mathbf{F}_0 = [1, 1, \ldots, 1]$;

4. Compute $\dot{\mathbf{V}}$ by discarding those points whose corresponding f_x is 0 from $\mathbf{V}_t \bigcap \mathbf{V}_{t+1}$. Then solve:

$$\mathbf{m}^*_{t+1} = arg(min(EOF(\dot{\mathbf{V}}, \mathbf{m}_t, \mathbf{m}_{t+1}, \mathbf{D}_O))) \quad (11)$$

5. Compute the motion residual error e_x at each point x in set $\mathbf{V}_t \bigcap \mathbf{V}_{t+1}$ using $\mathbf{m}^*_{t+1}$;

6. Re-set flag vector $\mathbf{F}$ by:

$$f_x = \begin{cases} 0 & if \;\; e_x > \alpha d_t; \\ 1 & otherwise; \end{cases} \quad (12)$$

where α is initially set as 0.9 in our experiments. To prevent the discarding of unoccluded areas, it is adjusted adaptively.

7. If flag vector $\mathbf{F}_t$ is not changed, or maximum iteration number has been exceeded, end the algorithm. Otherwise go to 4.

In our system, we use $\mathbf{m}_{t+1} = \mathbf{m}_t$ as initial guess. Levenberg-Marquardt algorithm is used to solve Eq. 11. During minimization, penalties are also added on very large motion candidates. In our experiments, the above algorithm normally converges in 3 to 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 do much harm to motion estimation. The results of our occlusion detection algorithm are illustrated in Figure 4 and 5.

2.4. Post-Regularization

Inaccurate optical flow estimation and lack of 3D information cause accumulation error in head tracking systems that are based on SSD minimization. Large motion and occlusion make the problem even worse. 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 [4]. Our algorithm uses edge information differently.

The idea is to use edge flow as a post-regularization heuristic. After we get $\mathbf{m}^*_{t+1}$ from the integrated algorithm introduced in Section 2.3, edge flow is used to refine $\mathbf{m}_{t+1}$ in a small neighborhood around $\mathbf{m}^*_{t+1}$ until the optimal motion vector is found. We believe 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. An evidence of the effectiveness of the edge information can be found in [12] 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 follows:

1. Detect edge points in the interested area (face area) in frames t and $t+1$ (in frame t, the interested area is the face model's projective area, while in frame $t+1$ the interested area should be large enough to cover the face area in frame t and the possible large motion.);

2. Detect edge features in frame t. We refer to an 8×8 block as a "edge feature" if it contains more than 8 edge points. To avoid aperture ambiguity, blocks with strictly vertical and horizontal edges are discarded;

3. Detect the perfect matches of the frame t features in frame $t+1$. Two edges are said to be perfectly matched if they are identical in the binary edge domain. This can be done very strictly by increasing the size of the observing domain. For those features where we could not find the perfect matches we simply discard them. Finally we find a set of edge features $\mathbf{P}$ in frame t. Each point $\mathbf{E}_t$ in set $\mathbf{P}$ has a perfectly matched point $\mathbf{E}_{t+1}$ in frame $t+1$;

4. Compute the edge flow $\mathbf{D}_E = [U_E, V_E]$ at point $\mathbf{E}_t$ as follows:

$$\mathbf{D}_E = \mathbf{E}_{t+1} - \mathbf{E}_t \quad (13)$$

5. Refine the motion vector $\mathbf{m}^*_{t+1}$ with edge flow. Before refinement, the "outliers" must be first removed. The motion residual error at each edge point in $\mathbf{P}$ can be computed by:

$$e'_x = \left| s\mathbf{D}_M - \mathbf{D}_E \right|^2 \quad (14)$$

If the error is bigger than a threshold, we consider it as outlier and remove it from $\mathbf{P}$. This is because we believe that we have found a reasonably good solution before post-regularization. It is not possible to get very big error on any edge point unless the edge flow is wrong. The EOF function can be defined as:

$$EOF'(\mathbf{P}, \mathbf{m}_t, \mathbf{m}_{t+1}, \mathbf{D}_O) = \frac{1}{N'} \sum_{x \in \mathbf{P}} e'_x \quad (15)$$

Using $\mathbf{m}^*_{t+1}$ as the initial guess, the post-regularization process tries to find an optimal solution $\mathbf{m}^{**}_{t+1}$ in a small neighborhood around $\mathbf{m}^*_{t+1}$ to minimize EOF':

$$\mathbf{m}^{**}_{t+1} = arg(min(EOF'(\mathbf{P}, \mathbf{m}_t, \mathbf{m}_{t+1}, \mathbf{D}_O))) \quad (16)$$

Our experiments have shown that the above algorithm heavily reduces the accumulation error. The tracking results of our system without/with post-regularization are illustrated in Figure 3.

3. Experiments

Our system has been tested on extensive occlusion image sequences. To show the accuracy, a synthetic occlusion image sequence with ground truth is used. To show the robustness, some real occlusion image sequences are tested.

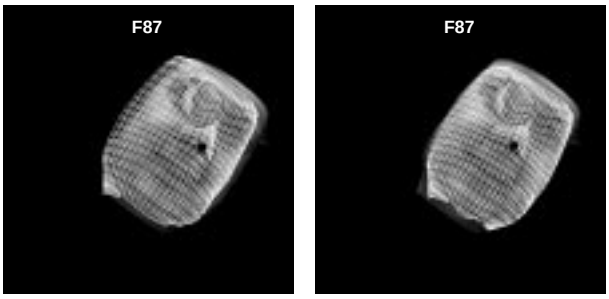


Figure 3. Tracking results without/with post-regularization

For ground truth validation, we generated a synthetic image sequence (Figure 4) by using a set of known motion vectors to animate real face range data. The occluding object, a sphere undergoing a $sine$-like motion, is added to the scene. The image resolution is 460×450. The largest motion between 2 adjacent frames is about 15 pixels. The occlusion detection and tracking results in some key frames are shown in Figure 4. The black dots in the tracked frames indicate the nose position of the ESQ face model. The graphs in Figure 4 show the comparisons between the 6 estimated parameters and the ground truth. Through the graphs we can see that our system tracked the motion accurately during occlusion. We have found that the estimation of the translation along z axis has bigger error, which is due to lack of 3D information. One possible solution is to formulate the z scaling factor of ESQ face model into the motion estimation framework. This makes the system more stable.

Figure 5 shows the experimental results on some real image sequences. (a), (b) are image sequences acquired from a Sony camera at 30 FPS. The image resolution is 320×240. The largest motion between 2 adjacent frames is about 12 pixels. The figure shows the occlusion detection results and the tracking results. We can clearly see that our system captures the 3D head motion robustly and correctly.

4. Conclusions and Future Plans

In this paper, we have presented a system which can robustly track occluded heads from 2D monocular image sequences. We have designed a face model with close-form formula based on ESQ and then used it in our system. We have also demonstrated a method which can successfully detect the occluded areas on the face. In our system, 3D motion estimation and motion segmentation are integrated. Furthermore, we have developed a post-regularization method which heavily reduces the accumulation error incurred by motion ambiguities and occlusion. The experiments in Section 3 clearly show that our system is robust and accurate. Additionally, it is possible to apply our occlusion detection and post-regularization algorithms to other tracking systems. As we can see from our experiments, 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 analysis. This is extremely useful when we perform facial motion tracking to extract the FAPs from a video sequence in image encoding.

Our future plans include: 1. Improving the efficiency of our system - our implementation is not real time (mainly because of optical flow computation). We are currently working on more efficient motion estimation and numerical minimization algorithms. 2. Reducing the translation error along z axis by formulating the z scaling factor into the motion estimation framework.

Acknowledgments

Research funding was provided by the National Science Foundation Grants IRI-9619240 and CISE CDA-9703088. The authors would like to thank Lin Zhou (U. of Delaware VIMS Lab) for his ESQ face model, Sumit Basu (MIT Media Lab) for his ground truth motion data and useful advice, and Dr. Thomas Huang (U. of Illinois) for cyberware face data.

References

[1] A. Azarbayejani, T. Starner, B. Horowitz, and A.P. Pentland. Visually controlled graphics. In *Vismod*, 1991.

[2] S. Basu, I.A. Essa, and A.P. Pentland. Motion regularization for model-based head tracking. In *ICPR96*, page C8A.3, 1996.

[3] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *UMD*, 1995.

[4] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models: Applications to human face shape and motion estimation. In *CVPR96*, pages 231–238, 1996.

[5] T.S. Jebara and A.P. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *CVPR97*, pages 144–150, 1997.

[6] M. LaCascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *CVPR98*, pages 508–514, 1998.

[7] M. LaCascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. In *CVPR99*, pages I:604–610, 1999.

[8] H. Li, P. Roivainen, and R. Forchheimer. 3-d motion estimation in model-based facial image coding. *PAMI*, 15(6):545–555, June 1993.

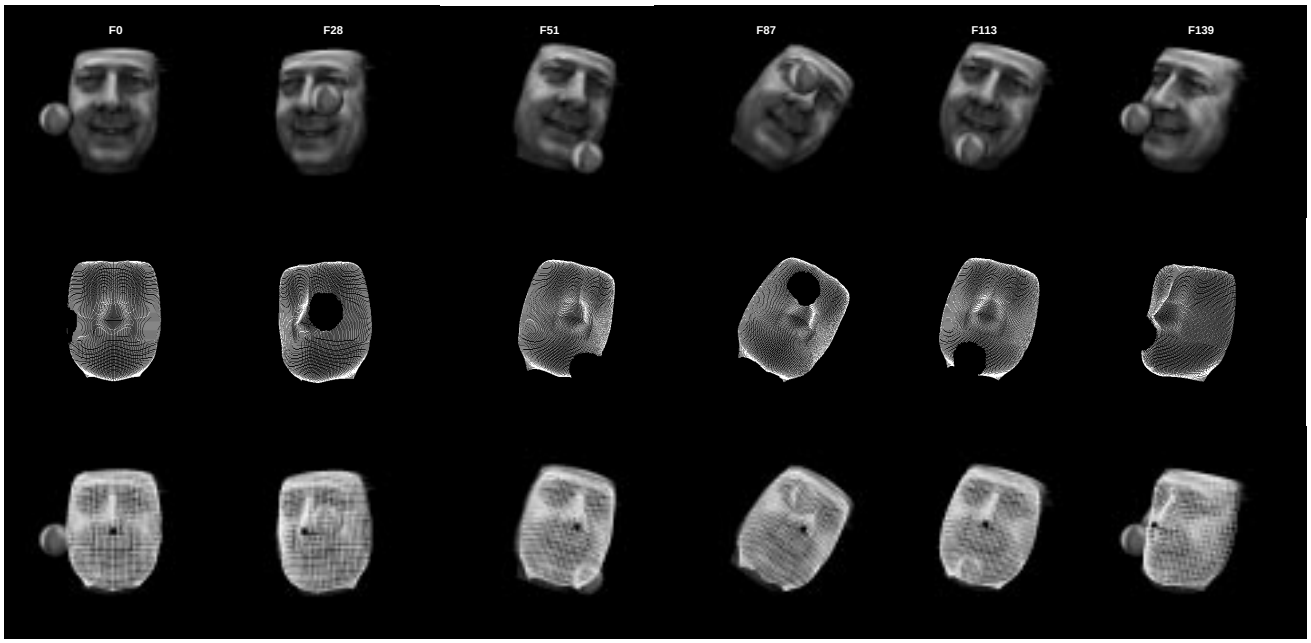
[9] N. Oliver, A.P. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *CVPR97*, pages 123–129, 1997.

[10] H. Tao. Nonrigid motion modeling and analysis in video sequences for realistic facial animation. In *PhD Thesis*, 1999.

[11] Y. Yacoob and L.S. Davis. Computing spatio-temporal representations of human faces. In *CVPR94*, pages 70–75, 1994.

[12] A. Zakhor and F. Lari. Edge-based 3-d camera motion estimation with application to video coding. *IP*, 2(4):481–498, 1994.

[13] L. Zhou and C. Kambhamettu. Extending superquadrics with exponent functions: Modeling and reconstruction. In *CVPR99*, pages II:73–78, 1999.



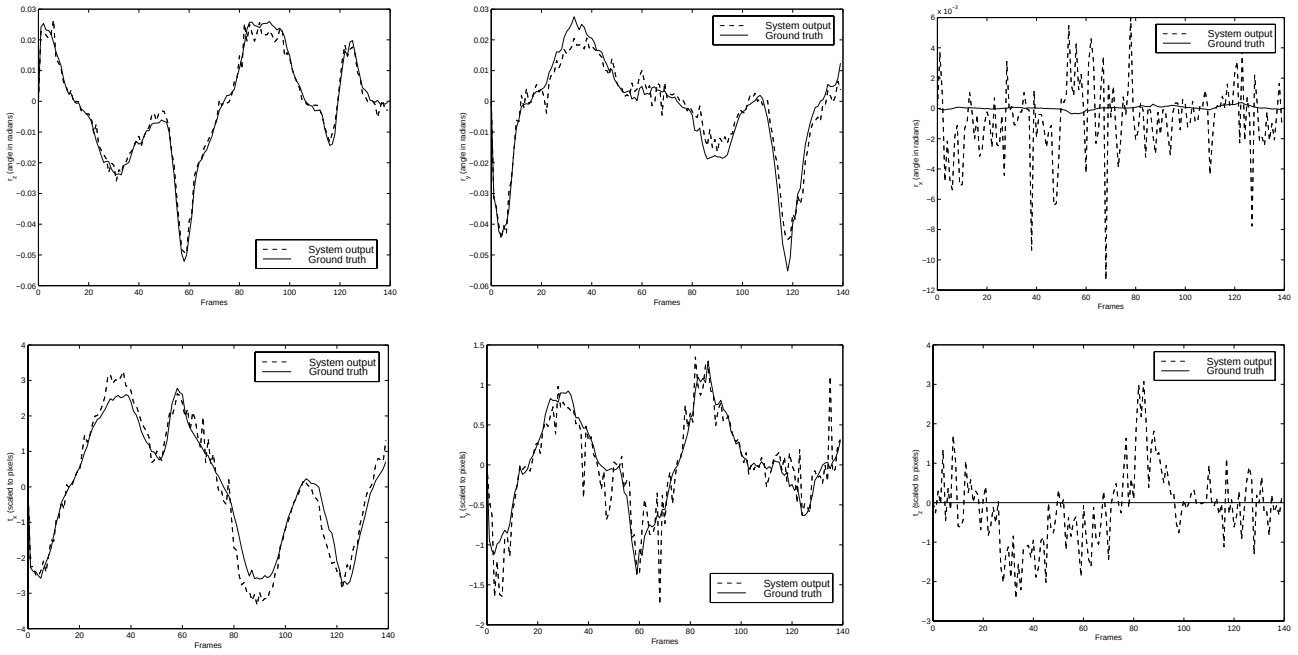
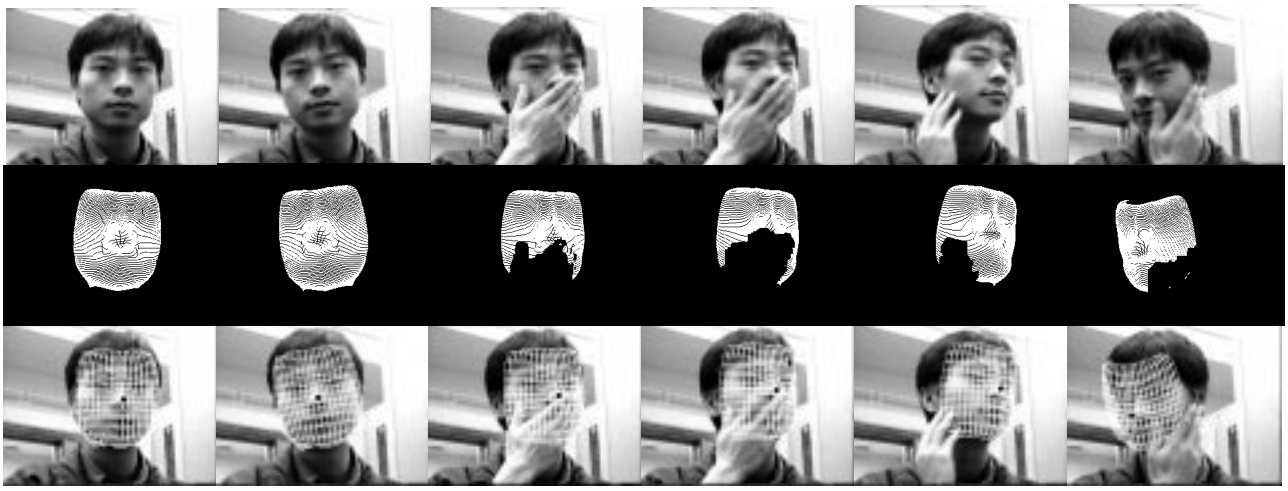


Figure 4. Experimental results on the synthetic image sequence. The first row includes some key frames of the original synthetic sequence. The second row indicates the occlusion detection results. The third row shows the tracking results (black dots indicate the nose position in each frame). The graphs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



(a) Tracking results of a 226 frame sequence

(b) Tracking results of a 199 frame sequence

Figure 5. Experimental results on 2 real image sequences. Black dots indicate the nose position in each frame.