Sumit Basu, Irfan Essa, and Alex Pentland. "Motion Regularization for Model-Based Head Tracking." In Proceedings of the IEEE Int'l Conf. on Pattern Recognition (ICPR '96). Vienna, Austria. 1996.

Motion Regularization for Model-Based Head Tracking

Sumit Basu, Irfan Essa, Alex Pentland
Perceptual Computing Section, Media Laboratory,
Massachusetts Institute of Technology, Cambridge, MA., U.S.A.
{sbasu,irfan,sandy}@media.mit.edu

Abstract

This paper describes a method for the robust tracking of rigid head motion from video. This method uses a 3D ellipsoidal model of the head and interprets the optical flow in terms of the possible rigid motions of the model. This method is robust to large angular and translational motions of the head and is not subject to the singularities of a 2D model. The method has been successfully applied to heads with a variety of shapes, hair styles, etc. This method also has the advantage of accurately capturing the 3D motion parameters of the head. This accuracy is shown through comparison with a ground truth synthetic sequence (a rendered 3D animation of a model head). In addition, the ellipsoidal model is robust to small variations in the initial fit, enabling the automation of the model initialization. Lastly, due to its consideration of the entire 3D aspect of the head, the tracking is very stable over a large number of frames. This robustness extends even to sequences with very low frame rates and noisy camera images.

1. Introduction and Motivation

This paper describes a method for robust tracking of head movements in extended video sequences. The main contribution of this paper is the regularization of optical flow using a 3D head model for robust and accurate tracking in 3D using only a single camera. This model-based method does not require the same features on the face to be visible over the entire length of the sequence and is stable over extended sequences, including those with large and rapid head motions. Additionally, this method allows tracking of all the six degrees of freedom of the rigid motion of the head, dealing gracefully with the motion singularities that most template-based methods fail to handle. We will show that the method presented in this paper can be used for tracking of large head motions over extended sequences for both full frame rate (30 frames per second) sequences and low-quality sequences captured at only 5 frames per second.

Our motivation for this work has come from the recent outburst of interest in face recognition, expression interpretation, and model-based coding. To date, most research efforts have assumed that only very small head motions are present [4, 7, 8, 12]. This, of course, limits the applicability of these methods.

Consequently, research in head tracking has become an increasingly important topic. Azarbeyajani and Pentland [2] have presented a recursive estimation method for structure and motion based on tracking of small facial features like the corners of the eyes or mouth. However, its use of feature tracking limited its applicability to sequences in which the same points were visible over most of the image sequence.

Most recently, Black and Yacoob [6] have developed a regularized optical-flow method that uses an eight parameter 2D model of flow and yields surprisingly good results. However, as they point out, the use of a plane-like 2D model limits accurate tracking to medium-size head motions; the method will fail when presented with large head rotations.

2. Our Approach

We were interested in developing a system that could accurately track the head under virtually all conditions including large head motions and low frame rates. Consequently, we became interested in developing a more accurate and robust head tracking method. This meant that we could not depend on the same points on the head being visible over the entire length of the sequence; nor could we use a scheme that would have singularities for certain kinds of motion or certain orientations. It was necessary to have a system that could robustly and accurately track all six degrees of freedom of the rigid motion of the head over a wide range of values.

As a result, we decided to take the approach of interpreting the optical flow field using a three-dimensional model. In doing this there was a tradeoff as to how complex a model of the head to use. Too simple a model, such as a plane, would not track the motion accurately. Too complex a model, such as an actual head, would require a very exact

initial fit. If a detailed model were not fit accurately, the detailed features of the model could cause more harm than good. We thus settled on an ellipsoidal model of the head, which is a reasonable approximate to the entire shape and which can easily be automatically initialized.

The technique we use for tracking this model may be considered as *motion regularization* or *flow regularization*. The unconstrained optical flow is first computed for the entire sequence. The rigid motion of the 3D head model that best accounts for the observed flow is interpreted as the motion of the head. A similar approach is used by Horowitz and Pentland [9] to track non-rigid deformations.

A good amount of previous work exists on the technique of flow regularization; Adiv [1] segmented flow into patches that were consistent with a single 3D motion. Bergen, Anandan, *et al.* [3] described a method for estimating model and motion parameters for several types of motion models using a "direct estimation" technique. Black and Yacoob's method [6] is based on Black and Anandan's [5] robust regression scheme over visual motion, constraining the flow computation by an analytic eight parameter transform.

Our work differs from this in that we use a full 3D rigid model. The model we chose to use was an ellipsoid; however, the framework we have created allows any set of 3D points to be used as a model for tracking. Certainly, this method does not account for all of the different motions of the head. However, it captures the rigid motions very accurately.

3. Methodology

3.1. The Model

The ellipsoid itself is parameterized by the sizes of its major axes, r_x, r_y, and r_z. These values are determined by automatically fitting an ellipsoid to the head in the first frame of the sequence (details of the initialization are described below). The surface of the resulting ellipsoid is then sampled to produce a set of 3D points, $\mathbf{P}_o$, and corresponding outward-pointing normal vectors, $\mathbf{N}_o$. The kth column of $\mathbf{P}_o$ is $[\ x_k \quad y_k \quad z_k \quad 1\ ]^T$, while the kth column of $\mathbf{N}_o$ is $[\ x_n \quad y_n \quad z_n\ ]^T$.

3.2. Rigid Motion Formulation

The rigid motion of the model is described by a vector of six parameters:

$$\mathbf{a} = [\ \alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z\ ]^T.$$

The first three parameters describe the rotations about the z, y, and x axes (respectively) of the local coordinate frame of the ellipsoid. The last three parameters define the 3D translation of the model. A given vector $\mathbf{a}$ results in the following 4x4 transform $\mathbf{T}$ (note $\cos(\alpha)$, $\sin(\beta)$, *etc.*, . have been abbreviated as c_α, s_β, *etc.*, .):

$$\mathbf{T} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & t_x \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & t_y \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

The current state of the model points, $\mathbf{P}$, can then be computed with $\mathbf{P} = \mathbf{T} \cdot \mathbf{P}_o$. The current normal vectors can be similarly found with $\mathbf{N} = \mathbf{T}_R \cdot \mathbf{N}_o$, where $\mathbf{T}_R$ is the 3x3 (pure rotational) transform contained in the first three rows and columns of $\mathbf{T}$.

3.3. Automatic Initialization

During the development of the system, the parameters $\mathbf{a}_o$ for the initial frame were obtained using a graphical tool in which an ellipsoid could be moved along all six degrees of freedom. In addition, the axes of the ellipsoid could be adjusted to obtain x_r, y_r, and z_r.

However, since our goals required the ability to find and track people automatically, we incorporated the modular eigenspace face and feature detection work of Moghaddam and Pentland [10] in order to parameterize and fit this ellipsoid. This system finds the location of the head itself and the locations of the eye, nose, and mouth within the head. We have developed expressions for the scales and initial location of the ellipsoid in terms of these coordinates based on a database of hand-fit ellipsoids. These expresions are then applied to the output of the feature-finding system to automatically scale and fit the ellipsoid in the first frame. Since Moghaddam and Pentland's system is optimized for the frontal view (i.e., where the head is facing the camera), it was necessary to ensure that each sequence began with a near-frontal view.

3.4. Projecting the Model onto the Viewing Plane

Though our model is a 3D representation, the image sequence is in 2D, and thus we must project this representation onto the viewing plane of the sequence. This can be done with a simple perspective transformation. Consider the x, y origin to be at the center of the viewing plane. Then, for each x, y, z triple in $\mathbf{P}$, the corresponding 2D point will have coordinates:

$$x_v = \frac{x}{1 - z/z_d}, \qquad y_v = \frac{y}{1 - z/z_d} \tag{2}$$

The z_d term specifies how significant the effect of perspective is and thus corresponds roughly to focal length. Note that this value does not have to be estimated for a given

sequence: it simply determines the magnitude of the z parameter. Clearly, the numerical values will vary with the actual focal length of the camera. If actual physical distances (i.e., depth in meters) are required, it is a simple matter to calibrate this value to a given camera's focal length.

We now define $\mathbf{Q}$ as the matrix of 2D points x_v, y_v corresponding to the 3D points of $\mathbf{P}$, with each column of the matrix containing one coordinate pair. At this point, we also take into consideration $\mathbf{N}$, the matrix of normals we have been carrying along. We are looking at the 3D world from our viewing plane with a "view vector" (gaze direction) of $[\ 0 \quad 0 \quad -1\ ]^T$. We will be able to view only those parts of the model for which the dot product of the surface normal and the view vector is negative. Because of our particular view angle, this means that only the points with positive z_n values (the z component of the surface normal) will be visible.

3.5. Generating Flow Fields from the Model

The optic flow at each point x, y in an image is traditionally defined as the vector $[\ u \quad v\ ]^T$, which describes the displacement from the corresponding point in the previous image (i.e., the point in the previous frame was $x-u, y-v$). To find the corresponding measure for our model given a set of initial parameters $\mathbf{a}_i$ for one frame and a candidate set $\mathbf{a}_j$ for the next frame, we first need to find the subset of points in the model which are visible for both frames (for all other model points, the flow is undefined). We define $\mathbf{V}_i$ and $\mathbf{V}_j$ as the appropriate subsets of $\mathbf{Q}_i$ and $\mathbf{Q}_j$.

The "model flow" between these two frames of the model is then $\mathbf{F}_M$ = $\mathbf{V}_j$ - $\mathbf{V}_i$. The kth column of $\mathbf{F}_M$, $[\ u_{M,k} \quad v_{M,k}\ ]^T$, is the model flow vector for the image coordinates x_k, y_k specified by the kth column of $\mathbf{V}_j$.

3.6. Comparing Generated Flow with Actual Flow

The next task is to see how well the model flow for the candidate parameters $\mathbf{a}_j$ fits the actual flow (as computed by a general optic flow algorithm). The metric we will use is a "robust" mean squared error between the actual and the model flow. Since the model flow only has values for some x, y locations while the actual flow is defined everywhere, we sum over only the n_c common locations. Using the notation previously defined, we have the following expression for the error between the model flow $\mathbf{F}_M$ and the actual flow $\mathbf{F}_A$, where v_k is the vector error for one pair of model and actual flow vectors, v_t is the error threshold of the robust norm, and e_k is the contribution to the total error from this pair:

$$v_k = (u_{M,k} - u_A(x_k, y_k))^2 + (v_{M,k} - v_A(x_k, y_k))^2 \quad (3)$$

$$e_k = \begin{cases} v_k & \text{if } v_k < v_t \\ v_t & \text{if } v_k \geq v_t \end{cases} \quad (4)$$

$$E(\mathbf{P}_o, \mathbf{a}_i, \mathbf{a}_j, \mathbf{F}_A) = \frac{1}{n_c} \sum_{k=1}^{n_c} e_k \quad (5)$$

3.7. Finding the Optimal Parameter Set

We now need to find the locally optimal parameter set $\mathbf{a}_j^*$ which results in the flow that best matches the actual flow:

$$\mathbf{a}_j^* = \arg(\min_j E(\mathbf{P}_o, \mathbf{a}_i, \mathbf{a}_j, \mathbf{F}_A)) \quad (6)$$

Exhaustively searching through the six-dimensional space of $\mathbf{a}$ would of course be impossible; we thus settle for a local minimum. This minimum is found by using the "simplex" gradient descent technique (implemented as described by [11]) with the error function E defined above, and a starting point of $\mathbf{a}_i$ (i.e., the current parameters).

4. Experiments and Results

4.1. Tracking

To demonstrate the tracking performance of this system we have presented several example sequences in the figures below. In figure 1, several key frames from a sequence captured at 30 FPS with a Sony HandyCam are shown. The first row of images contains the original images from the sequence, while the next two show tracking with a planar and an ellipsoidal model respectively. Both models were initialized automatically. The plots below the images show the values of the rotations around the axes of the model's coordinate frame (α, β, and γ). Though these parameters are difficult to interpret at a glance, it is clear that all three angles should return to zero when the face passes through its original, frontal orientation (see the plots at time 0, where $\alpha = \beta = \gamma = 0$). We can see that this is the case for the ellipsoidal model around frames 160 and 110, where the face is frontal. For the planar model, though, we do not see these convergences. While its point to point correspondence (i.e., a point on the model to a feature on the face) is quite good, the planar model does not seem to follow the orientations nearly as well as the ellipsoidal model, as can be seen by comparing the states of the models at the key frames shown.

The next two sequences are intended to show the robustness of the system over a variety of users and operating conditions. These are shown in figure 2 below. Several key frames are shown for each sequence with the ellipsoidal model superimposed on the image. The first sequence shows a head in normal conversation and shows the system's robustness to the non-rigid motions of the eyes and mouth. Because it uses all of the visible region of the head and a robust norm, it is not confused by the outliers that do not correspond to rigid motion.

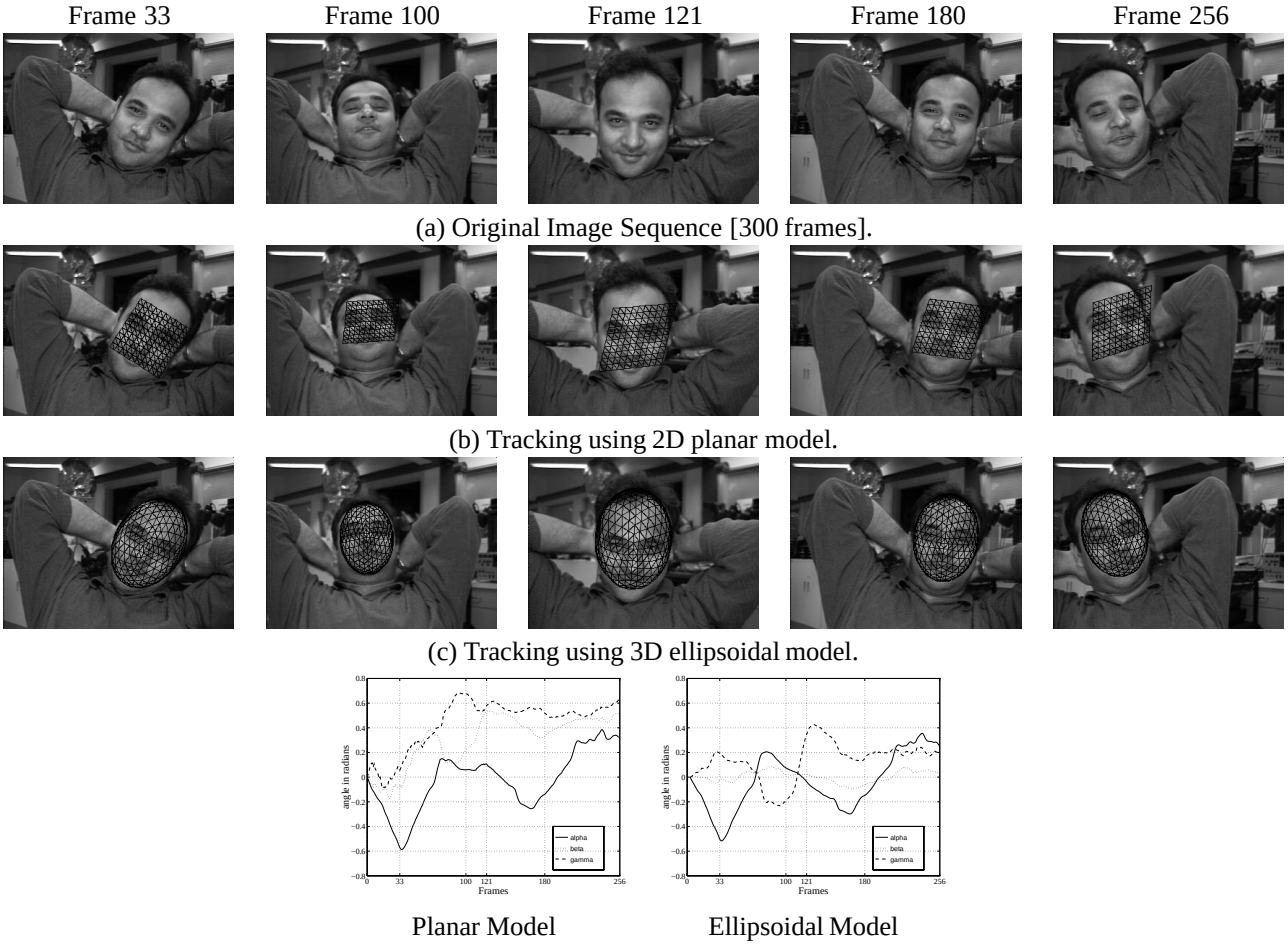


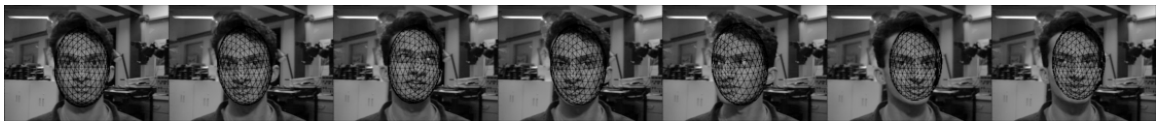
Figure 1. Results of tracking on a sequence acquired at 30 fps (using JPEG compression) and 320x240 resolution. The plots show the tracked orientation through the sequence.

The next sequence shows the system's robustness to poor operating conditions. The sequence was digitized with a very poor quality camera (an IndyCam) and contained a large amount of camera noise. In addition, the frame rate varied between 4 and 6 frames per second in the presence of significant (and rapid) head motion. Lastly, there was a great deal of "external motion" in the background from the hands moving around behind the head. Despite these conditions, the system was able to track the head accurately for the full 330 frames of the sequence, as can be seen in the key frames shown.

4.2. Validation

To demonstrate the accuracy of the system's position and orientation estimates, we have compared the results to a calibrated synthetic sequence. This sequence was generated by animating a synthetic head using the Silicon Graphics Inventor graphics libraries. The motion parameters used to drive the model were in the same format as those estimated by the system, and were obtained from running the system on a separate image sequence (not shown). As a result, the exact rigid parameters of the model were known at every frame. The results of this experiment are shown in figure 3 below. Again, several key frames are shown from the original sequence, followed by the tracking by the planar and ellipsoidal models. Below these key frames, a separate plot is shown for each rigid parameter. The "model" (dashed) line corresponds to the actual rigid parameters of the animated head, the "planar" (dotted) line corresponds to the parameters estimated for a planar model, and the "ellipsoid" (solid) line corresponds to the parameters estimated for an ellipsoidal model.

As in the sequence shown in figure 1, it is clear that both



(a) A 150 frame sequence at 30 FPS (320x240).

(b) A 300 frame sequence at about 5 FPS (90x90) Captured using an indycam.

Figure 2. Results of tracking on two sequences of different frame rates, resolution, and image quality

models maintain good point to point correspondence (i.e., point on the model to point on the head) over the whole sequence. However, the estimated orientations are far more accurate for the ellipsoidal model than for the planar model. This is clear from the plots: while the ellipsoidal model rarely varies more than 0.2 radians (10 degrees) from the actual orientation for a given axis of rotation, the planar model is often much further off than this. The ellipsoidal model also produces a slightly better estimate of the translation parameters, as can be seen below. It is the detailed orientation information that this system extracts, though, that is its most significant advantage over other schemes. This is due to the explicit 3D nature of the model.

5. Discussion and Conclusions

We have presented a method for robust tracking of heads in video. We have shown that this method is stable over extended sequences and large head motions and accurately extracts the three-dimensional rigid parameters of the head from a single view. We have shown that this method extracts more accurate information than a simple planar model because the ellipsoidal model represents the overall structure of the head.

We have also shown that flow regularization using a model is sensitive only to the motion being observed and completely ignores other motion in the scene. Unlike feature-based methods, the whole head is tracked, and we are not constrained by some features vanishing from view. We have also shown that robust tracking is possible even under poor digitization conditions. Lastly, the system is robust to variations in the initialization of the ellipsoid and thus can be reliably initialized automatically.

Even though we have framed this technique of model-based motion regularization only in the context of head tracking, we believe the method to be general enough to be applied to other tracking domains. In addition, the method is certainly not restricted to ellipsoidal models - any 3D model can be easily fitted into the framework described above. Even models with significant concavities can be used, since the robust error norm will effectively ignore these points when they are occluded. This framework can thus be applied to a variety of tracking tasks with a variety of models.

Note: Example sequences (with tracking) in QuickTime format can be viewed at http://vismod.www.media.mit.edu-/vismod/demos/faceview/.

Acknowledgements: Many thanks to John Wang, Baback Moghaddam, Andy Mortlock, and Ali Azarbayejani for their help.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *PAMI*, 4:384–401, November 1985.

[2] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *PAMI*, 17(6):562–575, June 1995.

[3] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV92*, pages 239–252, 1992.

[4] D. Beymer and T. Poggio. Face recognition from one example view. In *ICCV95*, pages 500–507, Cambridge, MA, June 1995. IEEE Computer Society.

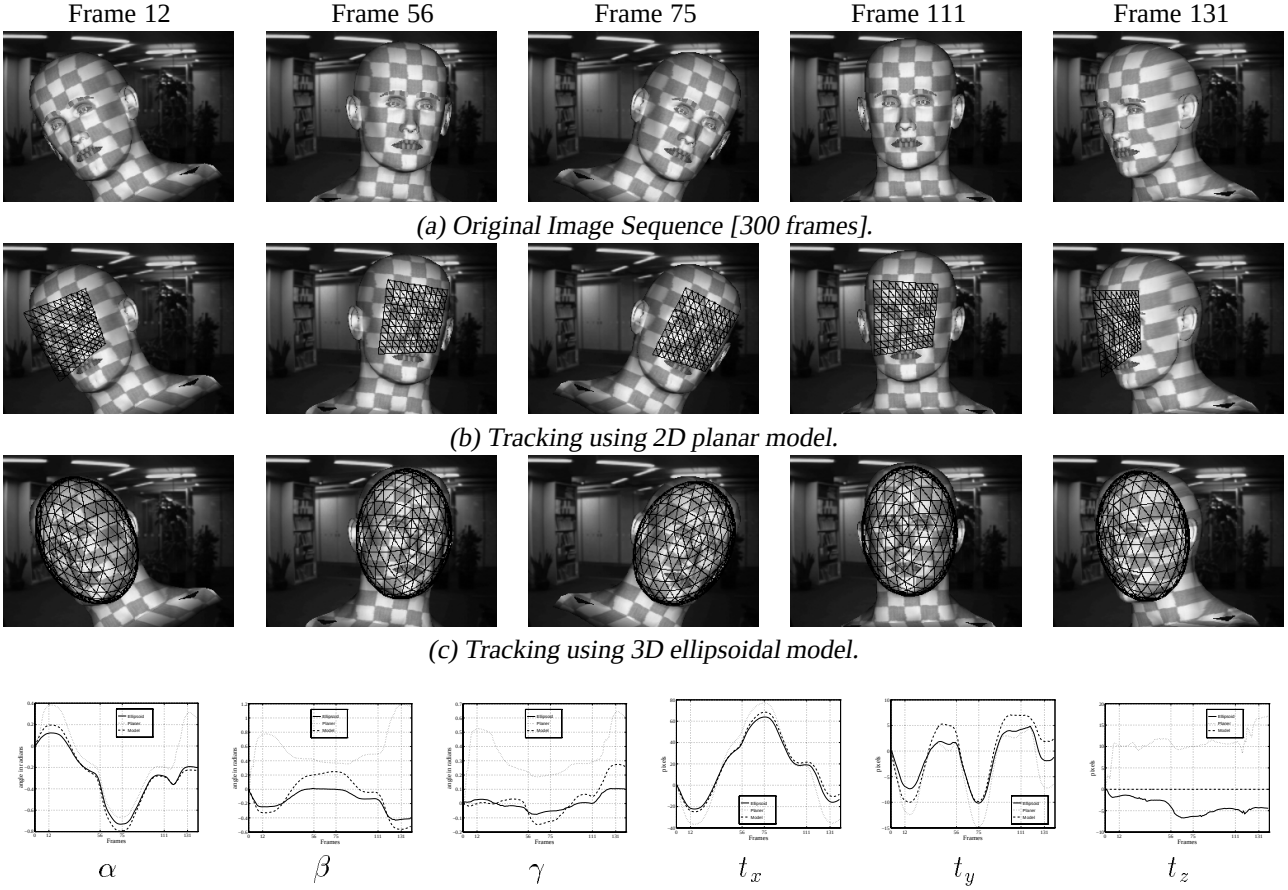


Figure 3. Results of tracking on a synthetic sequence. Row (a) shows the model sequence (dashed line in plots), row (b) shows the tracking using a planar model (dotted line) and (c) shows our 3D model (solid line) for tracking. The plots show the comparision of the tracked parameters with the actual parameters.

[5] M. J. Black and Y. Yacoob. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63:75–104, January 1996.

[6] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parametrized models of image motion. *International Journal of Computer Vision*, 25(1), 1997.

[7] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *ICCV95*, pages 360–367, Cambridge, MA, June 1995.

[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] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[10] A. Pentland, B. Moghaddam, T. Starner, O. Oliyide, and M. Turk. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition*. IEEE, 1994. Also, Media Lab TR 245, http://www-white.media.mit.edu/vismod).

[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipies in C*. Cambridge University Press, 1992.

[12] D. Terzopoulous and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *PAMI*, 15(6):569–579, June 1993.