M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 362
Submitted (Jan 1996): 13th Int'l. Conference on Pattern Recognition (ICPR '96), Vienna, Austria, August 25-30, 1996.

Motion Regularization for Model-based Head Tracking

Sumit Basu, Irfan Essa, Alex Pentland
Perceptual Computing Section, The Media Laboratory,
Massachusetts Institute of Technology
Cambridge, MA 02139, U.S.A.

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.

Categories: Facial Expressions, Expression Recognition, Face Processing, Facial Analysis, Systems and Applications, Motion Analysis, Pattern Analysis, Vision-based HCI.

1 Introduction & 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 image sequences captured at 5 frames per second.

1.1 Previous Work

Recently there has been a great spurt of interest in face recognition, expression interpretation, and model-based coding. However to date most research efforts have assumed that only very small head motions are present [4, 8, 9, 10, 11, 12, 14, 21, 22]. This, of course, limits the applicability of these methods.

Consequently, research in head tracking has become an increasingly important topic. Perhaps the first paper that concentrated on this problem was by Azarbeyajani and Pentland [2], who presenting a recursive estimation method 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 the entire image sequence.

Template-based methods have also been explored, such as the work of Darrell *et al.* [7] or Saulnier *et al.* [18]. These template-based methods have had the limitations of requiring initial training or initialization, and are also limited in the range of head motions they can track.

Most recently, Black and Yacoob [6] have developed a regularized optical-flow method that has yields surprisingly good results. In their method head motion is tracked by interpretation of optical flow in terms of a planar two-dimensional patch. Using this method they have been able to show stability over extended image sequences. However, as they point out, the use of a planar model limits accurate tracking to medium-size head motions; the method will fail when presented with large head rotations or scaling.

2 Our Approach

We are interested in developing a system that can 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 generalize the approach of Black and Yacoob by interpreting the optical flow field using a three-dimensional model rather than a simple planar model. In doing this there is 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 become disadvantageous. We thus settled on an ellipsoidal model of the head, which is a good approximate to the entire shape and which can also be initialized with reasonable accuracy.

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, and the rigid motion of the 3D head model that best accounts for the observed flow is interpreted as the motion of the head. This is much in the style of Horowitz and Pentland [15]. The model's 3D location and rotation is then modified by these parameters, and used as the starting point for interpreting the next frame, and so on. The details of this method will be described in the section 3 below. Our experiments (shown below) demonstrate that this method can provide very robust tracking over hundreds of image frames for a very wide range of head motions.

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. He then used the resulting clusters to estimate segmentation, structure (from the deviations in the parameters), and motion. Though this technique was attractive in the sense that it did not require prior models and that it could estimate all of these quantities, it only built a 2 1/2 D model of the surfaces involved, and it was not clear it would be robust over many frames. Bergen, Anandan, *et al.* [3] described a method for estimating model and motion parameters for several types of motion models. They used a "direct estimation" technique in their computation: instead of computing the unconstrained flow and then fitting it to a model, they used a series of models to constrain the flow computation. Sawhney *et al.* [19] used a model-based robust estimation technique to extract dominant motions from scenes. Black and Yacoob's Method [6] is based on Black and Anandan's [5] robust regression scheme over visual motion using a planar model, constraining the flow computation by an analytic eight parameter transform.

Our work differs from this in that we use a full 3D rigid model. Several attempts at tracking of rigid and nonrigid motion using complex 3D models has proved to be quite successful [15, 20]. Our method is primarily motivated by these efforts, except that we are incorporating robust estimation techniques to extract large three-dimensional motions. 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. We hope to extend this work by using the deviations of the actual flow from this estimate to determine the head's variations in structure and non-rigid motion from the ellipsoidal model.

3 Methodology

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 $[\ x_n \quad y_n \quad z_n\ ]^T$.

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 particular $\mathbf{a}$ results in the following 4x4 transform $\mathbf{T}$ (note cos(α), sin(β), *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 $\mathbf{T}$.

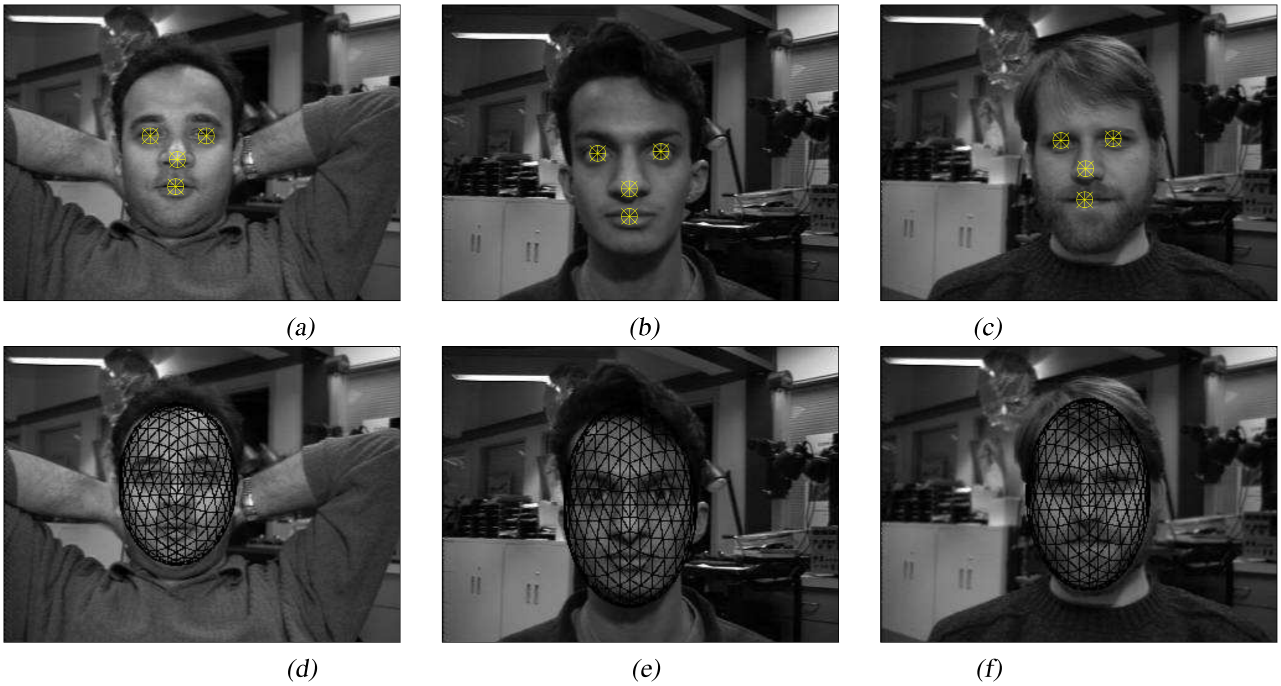


Figure 1: *Initialization of the model on the image for three different subjects. The top row shows the out from modular eigenspace head and feature detection system [13, 16]. The bottom row shows the coarse estimate of the ellipsoidal model on basis of this data.*

Initialization

During the development of the system, the parameters $\mathbf{a_0}$ 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 [13, 16] in order to parameterize and fit this ellipsoid automatically. This system finds the location of the head itself and the locations of the eye, nose, and mouth within the head. We developed expressions for the scales and initial location of the ellipsoid in terms of these coordinates based on a database of hand-fit ellipsoids.

Moghaddam and Pentland's system is optimized for the frontal view (i.e., where the head is upright and facing the camera). It was thus necessary to ensure that the first frame of a given sequence would contain the head posed in such a fashion. This was an easy constraint to satisfy: we found we could simply tell our subjects to "look at the camera" when the sequence began. Example outputs from the face and feature detection and corresponding automatically initialized ellipsoids are shown in Figure 1.

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 trivial 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.

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 $\mathbf{V_j}$.

Comparing Generated Flow with Actual Flow

The next task is to see how well the model flow for the candidate parameters $\mathbf{a_j}$ fit 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:

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

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

$$E(\mathbf{P_O}, \mathbf{a_i}, \mathbf{a_j}, \mathbf{F_A}) = \frac{1}{n_c} \sum_{k=1}^{n_c} e_k \quad (5)$$

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 [17]) with the error function E defined above, and a starting point of $\mathbf{a_i}$ (i.e., the current parameters).

4 Experiments & Results

Tracking

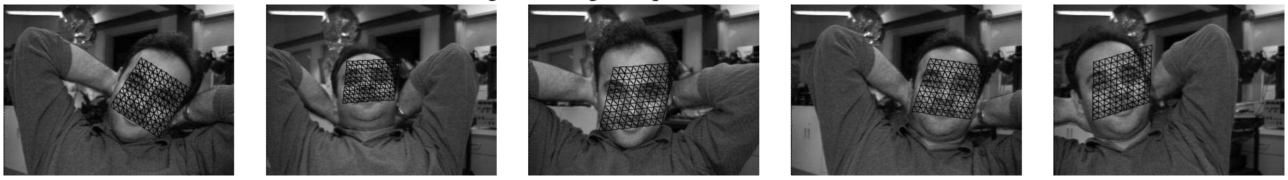
To demonstrate the tracking performance of this system we have presented several example sequences in the figures below. In figure 2, several key frames from a sequence captured at 30 FPS with a Sony HandyCam are shown. The first row of images contains the original images from the sequence, while the 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 three sequences are intended to show the robustness of the system over a variety of users and operating conditions. These are shown in figure 3 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, i.e., 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. The second sequence shows robustness to large angular motions and features such as facial hair.

The last sequence shows the system's robustness to poor operating conditions. The sequence was digitized with a very poor quality camera (an IndyCam) and contained large amounts 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.

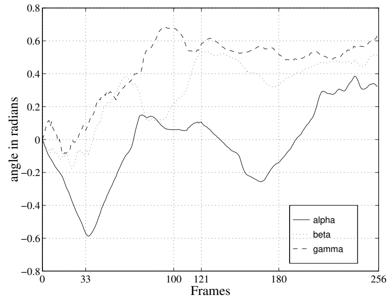
Frame 33 Frame 100 Frame 121 Frame 180 Frame 256

(a) Original Image Sequence [300 frames].

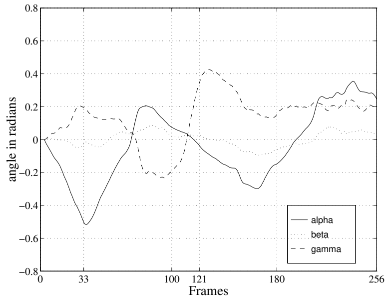


(b) Tracking using 2D planar model.

(c) Tracking using 3D ellipsoidal model.



Planar Model



Ellipsoidal Model

Figure 2: *Results of tracking on sequence acquired at 30 fps (using JPEG compression) and 320x240 resolution using. The plots show the change in orientation through the sequence. Results using a planar model and an 3D model are plotted*

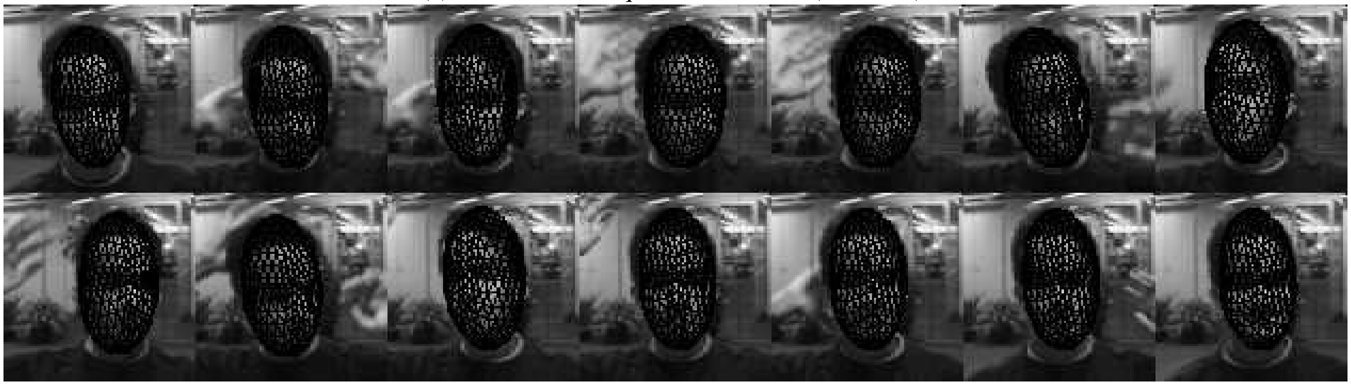
Validation

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 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 4 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" line corresponds to the actual rigid parameters of the animated head, the "planar" line corresponds to the parameters estimated for a planar model, and the "ellipsoid" line corresponds to the parameters estimated for a planar model.

As in the sequence shown in figure 2, it is clear that both 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

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

(a) A 300 frame sequence at 30 FPS (320x240).



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

Figure 3: *Results of tracking on three different seqiences acquired at different frame rates resolution with different image quality*

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.

4.1 Discussion & 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 then 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. Additionally, by using an ellipsoidal model, the system considers the motion of the entire head and not a set of planar patches. Similarly, 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.

5 Acknowledgments

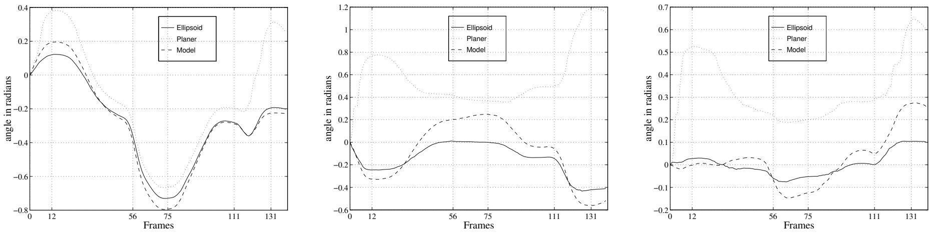
This material is based upon work supported in part by a National Science Foundation Graduate Fellowship. Many thanks also to John Wang for his DYNAMO optical flow routines, to Baback Moghaddam for his face and feature detection code, to Andy Mortlock for his crucial additions to that code, and to Ali Azarbayejani for his many helpful discussions.

Frame 12 Frame 56 Frame 75 Frame 111 Frame 131

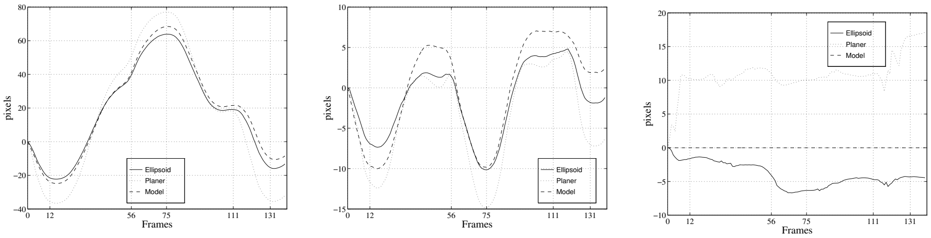
(a) Original Image Sequence [300 frames].

(b) Tracking using 2D planar model.

(c) Tracking using 3D ellipsoidal model.



α β γ



t_x t_y t_z

Figure 4: *Results of tracking on a synthetic sequence. Row (a) shows the model sequence, row (b) shows the tracking using a planar model and (c) shows our 3D model for tracking. The plots show the comparision for the six parameters between model, 2D and 3D analysis.*

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4:384–401, 1985.

[2] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive estimation of structure and motion using the relative orientation constraint. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 1993.

[3] J. R Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of European Conference on Computer Vision 1992*, pages 239–252, 1992.

[4] D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of the International Conference on Computer Vision*, pages 500–507. IEEE Computer Society, 1995.

[5] M. J. Black and P. Anandan. The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. Technical report, Xerox PARC, Dec. 1993. Tech. Report P93-00104.

[6] M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric model of image motion. In *Proceedings of the International Conference on Computer Vision*, pages 374–381. IEEE Computer Society, Cambridge, MA, 1995.

[7] I. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *Proceedings of the Workshop on Motion of Nonrigid and Articulated Objects*, pages 36–42. IEEE Computer Society, 1994.

[8] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *Proceedings of the International Conference on Computer Vision*, pages 360–367. IEEE Computer Society, Cambridge, MA, 1995.

[9] A. Lantis, C. J. Taylor, and T. F. Cootes. A unified approach to coding and interpreting face images. In *Proceedings of the International Conference on Computer Vision*, pages 369–373. IEEE Computer Society, Cambridge, MA, 1995.

[10] H. Li, P. Roivainen, and R. Forchheimer. 3-d motion estimation in model-based facial image coding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[11] K. Mase. Recognition of facial expressions for optical flow. *IEICE Transactions, Special Issue on Computer Vision and its Applications*, E 74(10), 1991.

[12] K. Matsuno, C-W. Lee, S. Kimura, and S. Tsuji. Automatic recognition of human facial expressions. In *Proceedings of the International Conference on Computer Vision*, pages 352–359. IEEE Computer Society, Cambridge, MA, 1995.

[13] B. Moghaddam and A. Pentland. Probabistic visual learning for object detection. In *Proceedings of the International Conference on Computer Vision*. IEEE Computer Society, 1995.

[14] Y. Moses, D. Reynard, and A. Blake. Determining facial expressions in real time. In *Proceedings of the International Conference on Computer Vision*, pages 296–301. IEEE Computer Society, Cambridge, MA, 1995.

[15] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1r991.

[16] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition Conference*, pages 84–91. IEEE Computer Society, 1994.

[17] William H. Press, Brian P. Flannery, Saul A. Teukoisky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[18] A. Saulnier, M. L. Viaud, and D. Geldreich. Real-time facial analysis and synthesis chain. In *International Workshop on Automatic Face and Gesture Recogntion*, pages 86–91, Zurich, Switzerland, 1995. Editor, M. Bichsel.

[19] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2d&3d dominant motion estimation for mosaicing and video representation. In *Proceedings of the International Conference on Computer Vision*, pages 583–590. IEEE Computer Society, 1995.

[20] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.

[21] D. Terzopoulus and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):569–579, June 1993.

[22] Y. Yacoob and L. Davis. Computing spatio-temporal representations of human faces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 70–75. IEEE Computer Society, 1994.