Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Model

Boston University, Computer Science Technical Report 99-005, May 1999
This technical report supercedes BU-TR-98-018 and BU-TR-97-020.

Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models

Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos
Computer Science Department
Boston University
Boston, MA 02215

Abstract

An improved technique for 3D head tracking under varying illumination conditions is proposed. The head is modeled as a texture mapped cylinder. Tracking is formulated as an image registration problem in the cylinder's texture map image. The resulting dynamic texture map provides a stabilized view of the face that can be used as input to many existing 2D techniques for face recognition, facial expressions analysis, lip reading, and eye tracking. To solve the registration problem in the presence of lighting variation and head motion, the residual error of registration is modeled as a linear combination of texture warping templates and orthogonal illumination templates. Fast and stable on-line tracking is achieved via regularized, weighted least squares minimization of the registration error. The regularization term tends to limit potential ambiguities that arise in the warping and illumination templates. It enables stable tracking over extended sequences. Tracking does not require a precise initial fit of the model; the system is initialized automatically using a simple 2D face detector. The only assumption is that the target is facing the camera in the first frame of the sequence. The formulation is tailored to take advantage of texture mapping hardware available in many workstations, PC's, and game consoles. The non-optimized implementation runs at about 15 frames per second on a SGI O2 graphic workstation. Extensive experiments evaluating the effectiveness of the formulation are reported. The sensitivity of the technique to illumination, regularization parameters, errors in the initial positioning and internal camera parameters are analyzed. Examples and applications of tracking are reported.

1 Introduction

Three-dimensional head tracking is a crucial task for several applications of computer vision. Problems like face recognition, facial expression analysis, lip reading, *etc.*, are more likely to be solved if a stabilized image is generated through a 3D head tracker. Determining the 3D head position and orientation is also fundamental in the development of vision-driven user interfaces and, more generally, for head gesture recognition. Furthermore, head tracking can lead to the development of very low bit-rate model-based video coders for video telephony, and so on. Most potential applications for head tracking require robustness to significant head motion, change in orientation, or scale. Moreover, they must work near video frame rates. Such requirements make the problem even more challenging.

In this paper, we propose an algorithm for 3D head tracking that extends the range of head motion allowed by a planar tracker[6, 11, 16]. Our system uses a texture mapped 3D rigid surface model for the head. During tracking, each input video image is projected onto the surface texture map of the model. Model parameters are updated via image registration in texture map space. The output of the system is the 3D head parameters and a 2D dynamic texture map image. The dynamic texture image provides a stabilized view of the face that can be used in applications requiring that the position of the head is frontal and almost static. The system has the advantages of a planar face tracker (reasonable simplicity and robustness to initial positioning) but not the disadvantages (difficulty in tracking out of plane rotations).

As will become evident in the experiments, our proposed technique can also improve the performance of a tracker based on the minimization of sum of squared differences (SSD) in presence of illumination changes. To achieve this goal we solve the registration problem by modeling the residual error in a way similar to that proposed in [16]. The method employs an orthogonal illumination basis that is pre-computed off-line over a training set of face images collected under varying illumination conditions.

In contrast to the previous approach of [16], the illumination basis is independent of the person to be tracked. Moreover, we propose the use of a regularizing term in the image registration; this improves the long-term robustness and precision of the SSD tracker considerably. A similar approach to estimating affine image motions and changes of view is proposed by [5]. Their approach employed an interesting analogy with parameterized optical flow estimation; however, their iterative algorithm is unsuitable for real-time operation.

Some of the ideas presented in this paper were initially reported in [22, 23]. In this paper we report the full formulation and extensive experimental evaluation of our technique. In particular the sensitivity of the technique to internal parameters as well as to errors in the initialization of the model are analyzed using ground truth data sensed with a magnetic tracker[1]. All the sequences used for the experiments and the corresponding ground truth data are publicly available [1]. Furthermore, a software implementation of our system is available from this site.

2 Background

The formulation of the head tracking problem in terms of color image registration in the texture map of a 3D cylindrical model was first developed in our previous work [22]. Similarly Schödl, Haro and Essa [30] proposed a technique for 3D head tracking using a full head texture mapped polygonal model. Recently Dellaert, Thrun and Thorpe [12] formulated the 3D tracking of planar patches using texture mapping as the measurement model in an extended Kalman filter framework.

Several other techniques have been proposed for free head motion and face tracking. Some of these techniques focus on 2D tracking (*e.g.,* [4, 9, 14, 16, 27, 35, 36]), while others focus on 3D tracking or stabilization. Some methods for recovering 3D head parameters are based on tracking of salient points, features, or 2D image patches. The outputs of these 2D trackers can be processed by an extended Kalman filter to recover 3D structure, focal length and facial pose [2]. In [21], a statistically-based 3D head model (eigen-head) is used to further constrain the estimated 3D structure. Another point-based technique for 3D tracking is based on the tracking of five salient points on the face to estimate the head orientation with respect to the camera plane[20].

Others use optic flow coupled to a 3D surface model. In [3], rigid body motion parameters of an ellipsoid model are estimated from a flow field using a standard minimization algorithm. In another approach [10], flow is used to constrain the motion of an anatomically-motivated face model and integrated with edge forces to improve tracking results. In [24], a render-feedback loop was used to guide tracking for an image coding application.

[1] `http://www.cs.bu.edu/groups/ivc/HeadTracking/`

Still others employ more complex physically-based models for the face that include both skin and muscle dynamics for facial motion. In [34], deformable contour models were used to track the non-rigid facial motion while estimating muscle actuator controls. In [13], a control theoretic approach was employed, based on normalized correlation between the incoming data and templates.

Finally, global head motion can be tracked using a plane under perspective projection [7]. Recovered global planar motion is used to stabilize incoming images. Facial expression recognition is accomplished by tracking deforming image patches in the stabilized images.

Most of the above mentioned techniques are not able to track the face in presence of large rotations and some require accurate initial fit of the model to the data. While a planar approximation addresses these problems somewhat, flattening the face introduces distortion in the stabilized image and cannot model self occlusion effects. Our technique enables fast and stable on-line tracking of extended sequences, despite noise and large variations in illumination. In particular, the image registration process is made more robust and less sensitive to changes in lighting through the use of an illumination basis and regularization.

3 Basic Idea

Our technique is based directly on the incoming image stream; no optical flow estimation is required. The basic idea consists of using a texture mapped surface model to approximate the head, accounting in this way for self-occlusions and to approximate head shape. We then use image registration in the texture map to fit the model with the incoming data.

To explain how our technique works, we will assume that the head is a cylinder with a 360^o-wide image, or more precisely a video showing facial expression changes, texture mapped onto the cylindrical surface. Only an 180^o-wide slice of this texture is visible in any particular frame; this corresponds with the visible portion of the face in each video image. If we know the initial position of the cylinder then we can use the incoming image to compute the texture map for the currently visible portion, as shown in Fig. 1. The projection of the incoming frame onto the corresponding cylindrical surface depends only on the 3D position and orientation of the cylinder (estimated by our algorithm), and on camera model (assumed known).

As a new frame is acquired it is possible to estimate the cylinder's orientation and position

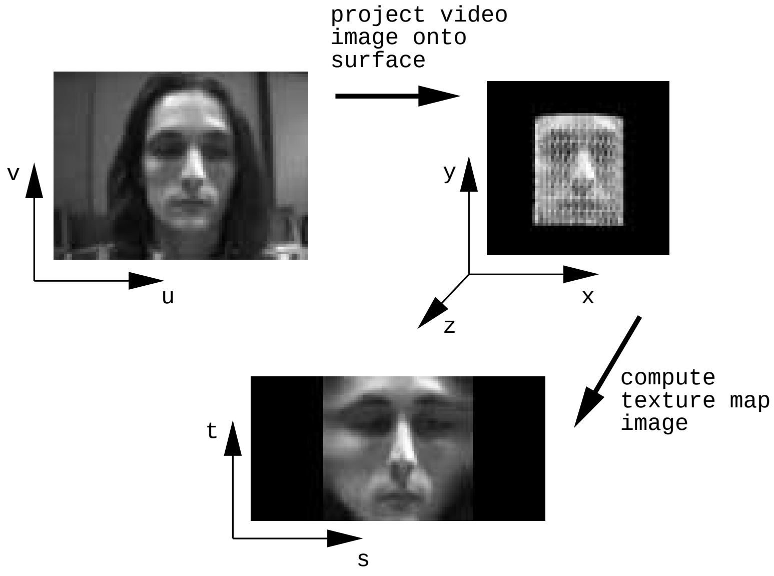

Figure 1: Mapping from image plane to texture map.

such that the texture extracted from the incoming frame best matches the reference texture. In other words, the 3D head parameters are estimated by performing image registration in the model's texture map. Due to the rotations of the head, the visible part of the texture can be shifted with respect to the reference texture. In the registration procedure we should then consider only the intersection of the two textures.

The registration parameters determine the projection of input video onto the surface of the object. Taken as a sequence, the projected video images comprise a *dynamic texture map*. This map provides a stabilized view of the face that is independent of the current orientation, position and scale of the surface model.

In practice, heads are not cylindrical objects, so we should account for this modeling error. Moreover, changes in lighting (shadows and highlights) can have a relevant effect and must be corrected in some way. In the rest of the paper, a detailed description of the formulation and implementation will be given. An extensive experimental evaluation of the system will also be described.

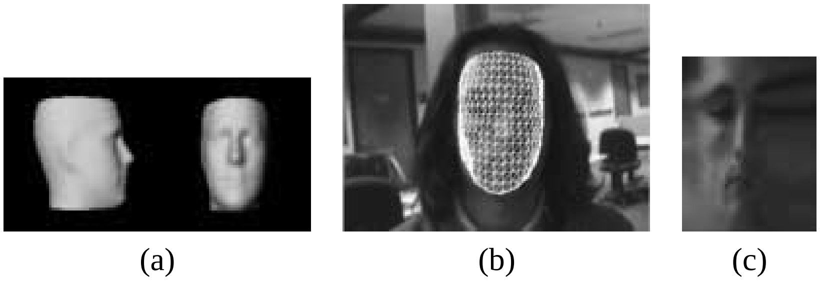

Figure 2: Generalized cylinder model constructed from average Cyberware head data (a). Model registered with video (b) and the corresponding texture map (c). Only the part of the texture corresponding to the visible part of the model is shown

4 Formulation

The general formulation for a 3D texture mapped surface model will now be developed. Fig. 1 shows the various coordinate systems employed in this paper: (x, y, z) is the 3D object-centered coordinate system, (u, v) is the image plane coordinate system, (s, t) is the surface's parametric coordinate system. The latter coordinate system (s, t) will be also referred to as the texture plane, as this is the texture map of the model. The (u, v) image coordinate system is defined over the range $[-1, 1] \times [-1, 1]$, and the texture plane (s, t) is defined over the unit square.

The mapping between (s, t) and (u, v) can be expressed as follows. First, assume a parametric surface equation:

$$(x, y, z, 1) = \mathbf{x}(s, t), \tag{1}$$

where 3D surface points are in homogeneous coordinates.

If greater generality is desired, then a displacement function can be added to the parametric surface equation:

$$\bar{\mathbf{x}}(s, t) = \mathbf{x}(s, t) + \mathbf{n}(s, t) d(s, t), \tag{2}$$

allowing displacement along the unit surface normal $\mathbf{n}$, as modulated by a scalar displacement function $d(s, t)$. For an even more general model, a vector displacement field can be applied to the surface.

An example of a cylinder with a normal displacement function applied is shown in Fig. 2. The was model computed by averaging the Cyberware scans of several people in known position[2] The inclusion of a displacement function in the surface formula allows for more detailed modeling of

[2] The average Cyberware scan was provided by Tony Jebara, of the MIT Media Lab.

the head. As will be discussed later, a more detailed model does not necessarily yield more stable tracking of the head.

The resulting surface can then be translated, rotated, and scaled via the standard 4×4 homogeneous transform:

$$\mathbf{Q} = \mathbf{D}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z\mathbf{S}, \tag{3}$$

where $\mathbf{D}$ is the translation matrix, $\mathbf{S}$ is the scaling matrix, and $\mathbf{R}_x$, $\mathbf{R}_y$, $\mathbf{R}_z$ are the Euler angle rotation matrices.

Given a location (s,t) in the parametric surface space of the model, a point's location in the image plane is obtained via a projective transform:

$$\left[\begin{array}{ccc} u' & v' & w' \end{array}\right]^T = \mathbf{P}\mathbf{Q}\bar{\mathbf{x}}(s,t), \tag{4}$$

where $(u,v) = (u'/w', v'/w')$, and $\mathbf{P}$ is a camera projection matrix:

$$\mathbf{P} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{array}\right]. \tag{5}$$

The projection matrix depends on the focal length f, which in our system is assumed to be known.

The mapping between (s,t) and (u,v) coordinates can now be expressed in terms of a computer graphics rendering of a parametric surface. The parameters of the mapping include the translation, rotation, and scaling of the model, in addition to the camera focal length. As will be seen in the next section, this formulation can be used to define image warping functions between the (s,t) and (u,v) planes.

4.1 Image Warping

Each incoming image must be warped into the texture map. The warping function corresponds to the inverse texture mapping of the surface $\bar{\mathbf{x}}(s,t)$ in arbitrary 3D position. In what follows we will denote the warping function:

$$\mathbf{T} = \Gamma(\mathbf{I}, \mathbf{a}) \tag{6}$$

where $\mathbf{T}(s,t)$ is the texture corresponding to the frame $\mathbf{I}(u,v)$ warped onto a surface $\bar{\mathbf{x}}(s,t)$ with rigid parameters $\mathbf{a}$. The parameter vector $\mathbf{a}$ contains the position and orientation of the surface. An example of input frame $\mathbf{I}$ with cylinder model and the corresponding texture map $\mathbf{T}$ are shown in Fig. 1.

4.2 Confidence Maps

As video is warped into the texture plane, not all pixels have equal confidence. This is due to nonuniform density of pixels as they are mapped between (u, v) and (s, t) space. As the input image is inverse projected, all visible triangles have the same size in the (s, t) plane. However, in the (u, v) image plane, the projections of the triangles have different sizes due to the different orientations of the triangles, and due to perspective projection. An approximate measure of the confidence can be derived in terms of the ratio of a triangle's area in video image (u, v) over the triangle's area in the texture map (s, t). For parts of the texture corresponding to the non-visible part of the surface $\bar{\mathbf{x}}(s, t)$ contribute no pixels and therefore have zero confidence.

Stated differently, the density of samples in the texture map is directly related to the area of each triangle in the image plane. This implies that the elements of the surface in the (s, t) plane do not all carry the same amount of information. The amount of information carried by a triangle is directly proportional to the number of pixels it contains in the input image $\mathbf{I}(u, v)$.

Suppose we are given a triangle ABC whose vertices in image coordinates are (u_a, v_a), (u_b, v_b), and (u_c, v_c), and in texture coordinates are (s_a, t_a), (s_b, t_b), and (s_c, t_c). Using a well known formula of geometry, the corresponding confidence measure is:

$$\kappa = \frac{\sqrt{|(u_b - u_a)(v_c - v_a) - (v_b - v_a)(u_c - u_a)|}}{\sqrt{|(s_b - s_a)(t_c - t_a) - (t_b - t_a)(s_c - s_a)|}}. \tag{7}$$

Given this formula, it is possible to render a confidence map $\mathbf{T}_w$ in the (s, t) plane. The denominator is constant in the case of cylindrical or planar models, because the (s, t) triangle mesh does not change.

In practice, the confidence map is generated using a standard triangular area fill algorithm. The map is first initialized to zero. Then each visible triangle is rendered into the map with a fill value corresponding to the confidence level. This approach allows the use of standard graphics hardware to accomplish the task.

Note also that, in the case of a cylindrical model, the texture map is 360^o wide but only a 180^o part of the cylinder is visible at any instant. In general, we should associate a zero confidence to the part of the texture corresponding to the back-facing portion of the surface.

The confidence map can be used to gain a more principled formulation of facial analysis algorithms applied in the stabilized texture map image. In essence, the confidence map quantifies the reliability of different portions of the face image. The non-uniformity of samples can also bias the analysis, unless a robust weighted error residual scheme is employed. As will be seen later, the resulting confidence map enables the use of weighted error residuals in the tracking procedure.

4.3 Model Initialization

To start any registration based tracker, the model must be fit to the initial frame to compute the reference texture and the warping templates. This initialization can be accomplished automatically using a 2D face detector [29] and assuming that the subject is approximately facing towards the camera in the first frame. The approximate 3D position of the surface is then computed assuming unit size. Note that assuming unit size is not a limitation, as the goal is to estimate the relative motion of the head. In other words people with a large head will be tracked as “farther from the camera” and people with a smaller head as closer.

It is important to note that using a simple model for the head makes it possible to reliably initialize the system automatically. Simple models, like a cylinder, require the estimation of fewer parameters in automatic placement schemes. As will be confirmed in experiments described in Sec. 8, tracking with the cylinder model is relatively robust to slight perturbations in initialization. A planar model [7] also offers these advantages; however, the experiments indicate that this model is not powerful enough to cope with the self-occlusions generated by large head rotations.

On the other hand, we have also experimented with a complex rigid head model generated averaging the Cyberware scans of several people in known position as shown in Fig. 2. Using such a model we were not able to automatically initialize the model, since there are too many degrees of freedom. Furthermore, tracking performance was markedly less robust to perturbations in the model parameters. Even when fitting the detailed 3D model by hand, we were unable to gain improvement in the tracker precision or stability over a simple cylindrical model. In contrast, the cylindrical model can cope with large out-of-plane rotation, and it is robust to initialization error due to its relative simplicity.

Once the initial position and orientation of the model $\mathbf{a}_0$ is known, we can generate the reference texture and a collection of *warping templates* that will be used for the tracking. The reference texture $\mathbf{T}_0$ is computed by warping the initial frame $\mathbf{I}_0$ onto the surface $\bar{\mathbf{x}}(s,t)$. Each warping template is computed by subtracting from the reference texture $\mathbf{T}_0$ the texture corresponding to the initial frame $\mathbf{I}_0$ warped through a slightly mis-aligned cylinder. Those templates are then used during the track to estimate the change of position and orientation of the cylinder from frame to frame as will be explained later.

For notational convenience, all images are represented as long vectors obtained by lexicographic reordering of the corresponding matrices. Formally, given an initial positioning of the model $\mathbf{a}_0$, and a parameter displacement matrix $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_K]$ we can compute the reference texture $\mathbf{T}_0$ and the warping templates matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_K]$:

$$\mathbf{T}_0 = \Gamma(\mathbf{I}_0, \mathbf{a}_0) \tag{8}$$

$$\mathbf{b}_k = \mathbf{T}_0 - \Gamma(\mathbf{I}_0, \mathbf{a}_0 + \mathbf{n}_k) \tag{9}$$

where $\mathbf{n}_k$ is the parameter displacement vector for the k^{th} difference vector $\mathbf{b}_k$ (warping template).

In practice, four difference vectors per model parameter are sufficient. For the k^{th} parameter, these four difference images correspond with the difference patterns that result by changing that parameter by $\pm\delta_k$ and $\pm 2\delta_k$. In our system $K = 24$ as we have six model parameters and four templates per parameter. The values of the δ_k can be easily determined such that all the difference images have the same energy. Note that the need for using $\pm\delta_k$ and $\pm 2\delta_k$ is due to the fact that the warping function $\Gamma(\mathbf{I}, \mathbf{a})$ is only locally linear in $\mathbf{a}$. Experimental results confirmed this intuition. An analysis of the extension of the region of linearity in a similar problem is given in [8].

Fig. 3 shows a few difference images (warping templates) obtained for a typical initial image using a cylindrical model. Note that the motion templates used in [5, 16] are computed in the image plane. In our case the templates are computed in the texture map plane. A similar approach has been successfully used in [8, 15, 31].

4.4 Illumination

Tracking is based on the minimization of the sum of squared differences between the incoming texture and a reference texture. This minimization is inherently sensitive to changes in illumination. Better results can be achieved by minimizing the difference between the incoming texture

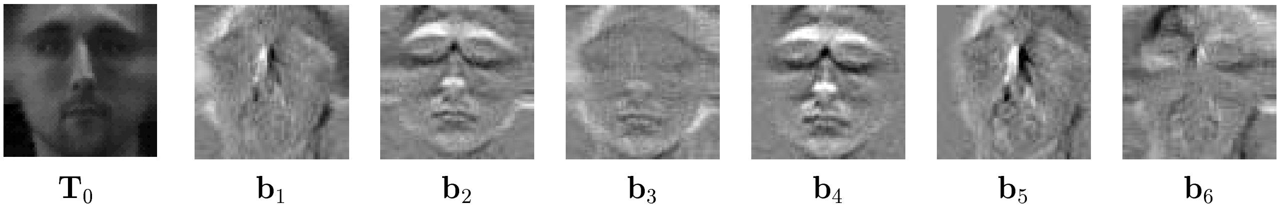

Figure 3: Example of warping templates. $\mathbf{T}_0$ is the reference texture. Warping templates $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$ correspond to translations along the (x, y, z) axes. Warping templates $\mathbf{b}_4$, $\mathbf{b}_5$ and $\mathbf{b}_6$ correspond to the Euler rotations. Note the similarity between the templates for horizontal translation $\mathbf{b}_1$ and vertical rotation $\mathbf{b}_5$. Note also the similarity between vertical translation $\mathbf{b}_2$ and horizontal rotation $\mathbf{b}_4$. Only that part of the template with non-zero confidence is shown.

and an illumination-adjusted version of the reference texture. If we assume a Lambertian surface in the absence of self-shadowing, then it has been shown that all the images of the same surface under different lighting conditions lie in a three-dimensional linear subspace of the space of all possible images of the object[32]. In this application, none of these conditions is met. Moreover, the non-linear image warping from image plane to texture plane distorts the linearity of the three-dimensional subspace. Nevertheless, we can still use a linear model as an approximation along the lines of [16, 17]:

$$\mathbf{T} - \mathbf{T}_0 \approx \mathbf{U}\mathbf{c}. \tag{10}$$

where the columns of the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$ constitute the *illumination templates*, and $\mathbf{c}$ is the vector of the coefficients for the linear combination.

In [16], these templates are obtained by taking the singular value decomposition (SVD) for a set of training images of the target subject taken under different lighting conditions. An additional training vector of ones is added to the training set to account for global brightness changes. The main problem of this approach is that the illumination templates are subject-dependent.

In our system, we generate a user-independent set of illumination templates. This is done by taking the SVD of a large set of textures corresponding to faces of different subjects, taken under varying illumination conditions. The SVD was computed after subtracting the average texture from each sample texture. The training set of faces we used was previously aligned and masked as explained in [26]. In practice, we found that ten illumination templates were sufficient to account for illumination changes.

Note that the illumination basis vectors tend to be low-frequency images. Thus any mis-

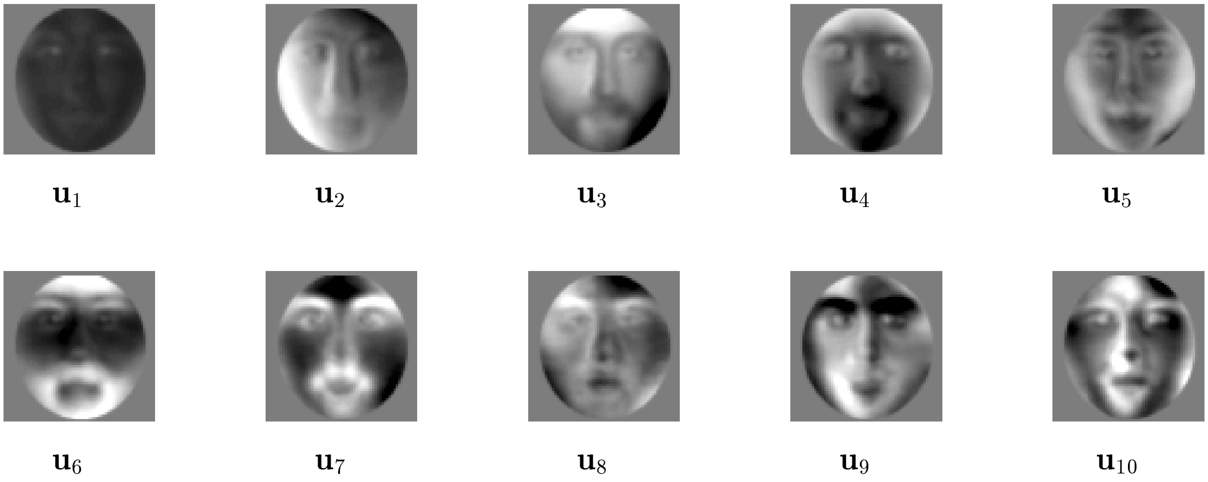

Figure 4: User-independent set of illumination templates. Only the part of the texture with non-zero confidence is shown.

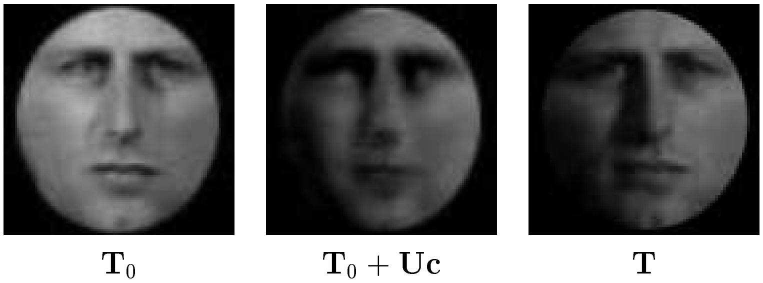

Figure 5: Example of the lighting correction on the reference texture. For a given input texture $\mathbf{T}$, the reference texture $\mathbf{T}_0$ is adjusted to account for change in illumination. The illumination-corrected reference texture is computed in terms of a linear combination of illumination templates $\mathbf{T}_0 + \mathbf{Uc}$.

alignment between the illumination basis and the reference texture is negligible. In addition, an elliptical binary mask $\mathbf{T}_l$ is applied on the illumination basis to prevent the noisy corners of the textures from biasing the registration.

The illumination basis vectors for the cylindrical tracker are shown in Fig. 4. Fig. 5 shows a reference texture and the same image after the masking and the lighting correction (in practice $\mathbf{T}_0$, $\mathbf{T}_0 + \mathbf{Uc}$, and $\mathbf{T}$).

4.5 Combined Parameterization

Following the line of [5, 16], a residual image is computed by taking the difference between the incoming texture and the reference texture. This residual can be modeled as a linear combination of illumination templates and warping templates:

$$\mathbf{T} - \mathbf{T}_0 \approx \mathbf{Bq} + \mathbf{Uc} \tag{11}$$

where $\mathbf{c}$ and $\mathbf{q}$ are the vector of the coefficients of the linear combination. In our experience this is a reasonable approximation for low-energy residual textures. A multi-scale approach using Gaussian pyramids [28] is used so that the system can handle higher energy residual textures [33].

5 Registration and Tracking

During initialization, the model is automatically positioned and scaled to fit the head in the image plane as described in Sec. 4.3. The reference texture $\mathbf{T}_0$ is then obtained by projecting the initial frame of the sequence $\mathbf{I}_0$ onto the visible part of the cylindrical surface. As a pre-computation, a collection of warping templates is computed by taking the difference between the reference texture $\mathbf{T}_0$ and the textures corresponding to warping of the input frame with slightly displaced surface parameters as described in Sec. 4.3.

Once the warping templates have been computed, the tracking can start. Each new input frame $\mathbf{I}$ is warped into the texture map using the current parameter estimate $\mathbf{a}^-$. This yields a texture map $\mathbf{T}$. The residual pattern (difference between the reference texture and the warped image) is modeled as a linear combination of the warping templates $\mathbf{B}$ and illumination templates $\mathbf{U}$ that model lighting effects (Eq. 11).

To find the warping parameters $\mathbf{a}$, we first find $\mathbf{c}$ and $\mathbf{q}$ by solving the following weighted least squares problem:

$$\mathbf{W}(\mathbf{T} - \mathbf{T}_0) \approx \mathbf{W}(\mathbf{Bq} + \mathbf{Uc}) \tag{12}$$

where $\mathbf{W} = diag[\mathbf{T}_w] * diag[\mathbf{T}_l]$ is the weighting matrix, accounting for the confidence weights $\mathbf{T}_w$ and the elliptical binary mask $\mathbf{T}_l$ mentioned earlier.

If we define:

$$\begin{aligned} \mathbf{R} &= \mathbf{T} - \mathbf{T}_0, & (13)\\ \mathbf{x} &= \begin{bmatrix} \mathbf{c} \\ \mathbf{q} \end{bmatrix}, & (14)\\ \mathbf{M} &= [\mathbf{U}|\mathbf{B}]. & (15) \end{aligned}$$

The solution can be written:

$$\begin{aligned}\mathbf{x} &= \arg\min_{\mathbf{x}} \|\mathbf{R} - \mathbf{M}\mathbf{x}\|_W && (16)\\ &= [\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M}]^{-1}\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{R} && (17)\\ &= \mathbf{K}\mathbf{R} && (18)\end{aligned}$$

where $\mathbf{K} = [\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M}]^{-1}\mathbf{M}^T\mathbf{W}^T\mathbf{W}$ and $\|\mathbf{x}\|_W = \mathbf{x}^T\mathbf{W}^T\mathbf{W}\mathbf{x}$ is a weighted L-2 norm. Due to possible coupling between the warping templates and/or the illumination templates, the least squares solution may become ill-conditioned. As will be seen, this conditioning problem can be averted through the use of a regularization term.

If we are interested only in the increment of the warping parameter $\Delta\mathbf{a}$, we may elect to compute only the $\mathbf{q}$ part of $\mathbf{x}$. Finally:

$$\mathbf{a} = \mathbf{a}^{-} + \Delta\mathbf{a} \qquad (19)$$

where $\Delta\mathbf{a} = \mathbf{N}\mathbf{q}$.

Note that this computation requires only a few matrix multiplications and the inversion of a relatively small matrix. No iterative optimization [5] is involved in the process. This is why our method is fast and can run at near NTSC video frame rate on inexpensive PCs and workstations.

5.1 Regularization

Independent of the weighting matrix $\mathbf{W}$, we have found that the matrix $\mathbf{K}$ is sometimes close to singular. This is a sort of *general aperture problem* and is due mainly to the intrinsic ambiguity between small horizontal translation and vertical rotation, and between small vertical translation and horizontal rotation. Moreover, we found that a coupling exists between some of the illumination templates and the warping templates.

Fig. 6 shows the matrix $\mathbf{M}^T\mathbf{M}$ for a typical sequence using the cylindrical model. Each square in the figure corresponds to an entry in the matrix. Bright values correspond with large values in the matrix, dark squares correspond with small values in the matrix. If the system were perfectly decoupled, then all off-diagonal elements would be dark. In general, brighter off-diagonal elements indicate a coupling between parameters.

By looking at the figure, it is possible to see the coupling that can cause ill-conditioning. The top-left part of the matrix is diagonal because it corresponds with the orthogonal illumination

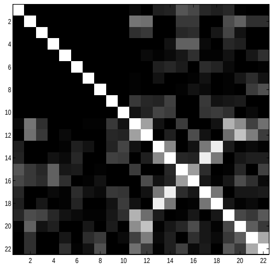

Figure 6: Example of matrix $\mathbf{M}^T\mathbf{M}$.

basis vectors. This is not true for bottom-right block of the matrix. This block of the matrix corresponds with the warping basis images. Note that the coupling between warping parameters and appearance parameters is weaker than the coupling within the warping parameter space. Such couplings can lead to instability or ambiguity in the solutions for tracking. To reduce the last kind of coupling Schödl, Haro and Essa [30] used parameters that are linear combinations of position and orientation; however, under some conditions this may lead to uncorrelated feature sets in the image plane.

To alleviate this problem, we regularize our system. The simplest possible approach consists of using truncated singular value decomposition (SVD)[18] to solve the least square problem of Eq. 16. Given the truncated SVD of the matrix $\mathbf{M}$:

$$\mathbf{M} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{20}$$

and defining $\tilde{\mathbf{x}} = \mathbf{V}^T\mathbf{x}$ we can rewrite Eq. 12 as:

$$\mathbf{W}\mathbf{R} \approx \mathbf{W}\mathbf{U}\mathbf{\Sigma}\tilde{\mathbf{x}}. \tag{21}$$

The regularized solution can be obtained as follows:

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}} = \mathbf{V}(\mathbf{U}^T\mathbf{W}\mathbf{U}\mathbf{\Sigma})^{-1}\mathbf{U}^T\mathbf{W}\mathbf{R}. \tag{22}$$

A more principled approach consists of defining the regularizer by adding a penalty term to the image energy shown in the previous section, and then minimize with respect to $\mathbf{c}$ and $\mathbf{q}$:

$$\begin{aligned} E &= \|(\mathbf{T} - \mathbf{T}_0) - (\mathbf{B}\mathbf{q} + \mathbf{U}\mathbf{c})\|_W + \gamma_1[\mathbf{c}^T\Omega_a\mathbf{c}] \\ &\quad +\gamma_2[\mathbf{a}^- + \mathbf{N_a}\mathbf{q}]^T\Omega_w[\mathbf{a}^- + \mathbf{N_a}\mathbf{q}]. \end{aligned} \tag{23}$$

The diagonal matrix Ω_a is the penalty term associated with the appearance parameter $\mathbf{c}$, and the diagonal matrix Ω_w is the penalty associated with the warping parameters $\mathbf{a}$.

We can define:

$$\begin{aligned} \mathbf{p} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{a}^{-} \end{bmatrix}, & (24) \\ \mathbf{N} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{N_a} \end{bmatrix}, & (25) \\ \Omega &= \begin{bmatrix} \frac{\gamma_1}{\gamma}\Omega_a & \mathbf{0} \\ \mathbf{0} & \frac{\gamma_2}{\gamma}\Omega_w \end{bmatrix}. & (26) \end{aligned}$$

and then rewrite the energy as:

$$E = ||\mathbf{R} - \mathbf{M}\mathbf{x}||_W + \gamma[\mathbf{p} + \mathbf{N}\mathbf{x}]^T\Omega[\mathbf{p} + \mathbf{N}\mathbf{x}]. \qquad (27)$$

By taking the gradient of the energy with respect to $\mathbf{x}$, and equating it to zero we get:

$$\begin{aligned} \mathbf{x} &= [\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M} + \gamma\mathbf{N}^T\Omega\mathbf{N}]^{-1}\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{R} & (28) \\ &+ \gamma[\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M} + \gamma\mathbf{N}^T\Omega\mathbf{N}]^{-1}\mathbf{N}^T\Omega\mathbf{p} & (29) \\ &= \tilde{\mathbf{K}}\mathbf{R} + \mathbf{Q}\mathbf{p} & (30) \end{aligned}$$

where $\tilde{\mathbf{K}} = [\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M}+\gamma\mathbf{N}^T\Omega\mathbf{N}]^{-1}\mathbf{M}^T\mathbf{W}^T\mathbf{W}$ and $\mathbf{Q} = \gamma[\mathbf{M}^T\mathbf{W}^T\mathbf{W}\mathbf{M}+\gamma\mathbf{N}^T\Omega\mathbf{N}]^{-1}\mathbf{N}^T\Omega$.

As before, if we are interested only in the warping parameter estimate, then we can save computation by solving only for the $\mathbf{q}$ part of $\mathbf{x}$. We can then find $\Delta\mathbf{a}$.

The choice of a diagonal regularizer implicitly assumes that the subvectors $\mathbf{c}$ and $\mathbf{q}$ are independent. In practice this is not the case. However, our experiments consistently showed that the performance of the regularized tracker is considerably superior with respect to the unregularized one. Evaluation experiments will be described in Sec. 8.

The matrices Ω_a and Ω_w were chosen for the following reasons. Recall that the appearance basis $\mathbf{U}$ is an eigenbasis for the texture space. If Ω_a is diagonal and with elements equal to the inverse of the corresponding eigenvalues, then the penalty term $\mathbf{c}^T\Omega_a\mathbf{c}$ is proportional to the *distance in feature space*[26]. This term thus prevents an artificially large illumination term from dominating and misleading the tracker.

The diagonal matrix Ω_w is the penalty associated with the warping parameters (cylinder translation and rotation). We assume that the parameters are independently Gaussian distributed around the initial position. We can then choose Ω_w to be diagonal, with diagonal terms equal to the inverse of the expected variance for each parameter. In this way we prevent the parameters from exploding when the track is lost. Our experience has shown that this term generally makes it possible to swiftly recover if the track is lost. We defined the standard deviation for each parameter as a quarter of the range that keeps the model entirely visible (within the window).

Note that this statistical model of the head motion is particularly suited for video taken from a fixed camera (for example a camera on the top of the computer monitor). In a more general case (for example to track heads in movies) a random walk model [2, 21] would probably be more effective. Furthermore, the assumption of independence of the parameters could be removed and the full non-diagonal 6×6 covariance matrix estimated from example sequences.

6 System Implementation

For sake of comparison, we implemented the system using both a cylindrical and a planar surface $\bar{\mathbf{x}}(s,t)$. To allow for larger displacements in the image plane we used a multi-scale framework. The warping parameters are initially estimated at the higher level of a Gaussian pyramid and the parameters are propagated to the lower level. In our implementation we found that a two level pyramid was sufficient. The first level of the texture map pyramid has a resolution of 128×64 pixels.

The warping function $\Gamma(\mathbf{I}, \mathbf{a})$ was implemented to exploit texture mapping acceleration present in modern computer graphics workstations. We represented both the cylindrical and the planar models as sets of texture mapped triangles in 3D space. When the cylinder is superimposed onto the input video frame, each triangle in image plane maps the underlying pixels of the input frame to the corresponding triangle in texture map. Bilinear interpolation was used for the texture mapping.

The confidence map is generated using a standard triangular area fill algorithm. The map is first initialized to zero. Then each visible triangle is rendered into the map with a fill value corresponding to the confidence level. This approach allows the use of standard graphics hardware to accomplish the task.

The illumination basis has been computed from a MIT database[26] of 1,000 aligned frontal view of faces under varying lighting conditions. Since all the faces are aligned, we had to determine by hand the position of the surface only once and then used the same warping parameters to compute the texture corresponding to each face. Finally, the average texture was computed and subtracted from all the textures before computing the SVD. In our experiments we found that the first ten eigenimages are in general sufficient to model the global light variation. If more eigenimages were employed, the system could in principle model more precisely effects like self-shadowing. In practice, we observed that there is a significant coupling between the higher-order eigenimages and the warping templates, which would make the tracker less stable. The eigenimages where computed from the textures at 128×64 resolution. The second level in the pyramid was approximated by scaling the eigenimages.

The system was implemented in C++ and OpenGL on a SGI O2 graphic workstation. The current version of the system runs at about 15 frames per second when reading the input from a video stream. The off-line version used for the experiments can process five frames per second. This is due to I/O overhead and decompression when reading the video input from a movie file. The software implementation, along with the eigenimages, and a number of test sequences is available from the web site: http://www.cs.bu.edu/groups/ivc/HeadTracking/.

7 Experimental Setup

During real time operation, in many cases, the cylindrical tracker can track the video stream indefinitely – even in the presence of significant motion and out of plane rotations. However, to better test the sensitivity of the tracker and to better analyze its limits, we collected a large set of more challenging sequences, such that the tracker breaks in some cases. Ground truth data was simultaneously collected using a magnetic tracker.

The test sequences were collected with a Sony Handycam on a tripod. Ground truth for these sequences was simultaneously collected via a “Flock of Birds” 3D magnetic tracker [1]. The video signal was digitized at 30 frames per second at a resolution of 320×240 non-interleaved using the standard SGI O2 video input hardware and then saved as Quicktime movies (M-JPEG compressed).

To collect ground truth of the position and orientation of the head, the transmitter of the magnetic tracker was attached on the subject's head. The “Flock of Birds” system [1] measures the relative position of the transmitter with respect to the receiver (in inches) and the orientation (in Euler angles) of the transmitter. The magnetic tracker, in an environment devoid of large metal objects and electro-magnetic frequencies, has a positional accuracy of 0.1 inches and angular accuracy of 0.5 degrees. Both accuracies are RMS averaged over the translational range. In a typical laboratory environment, with some metal furniture and computers, we experienced a lower accuracy. However, the captured measurements were still good enough to evaluate a visual tracker. In Figs. 8 and 9 it is possible to see how the noise level is certainly larger than the nominal accuracy of the magnetic tracker.

7.1 Test Data

We collected two classes of sequences. One set of sequences was collected under uniform illumination conditions. The other set was collected under time varying illumination. The time varying illumination has a uniform component and a sinusoidal directional component. All the sequences are 200 frames long (approximatively seven seconds) and contain free head motion of several subjects.

The first set consists of 45 sequences (nine sequences for each of five subjects) taken under uniform illumination where the subjects perform free head motion including translations, and both in-plane and out-of-plane rotations. The second set consists of 27 sequences (nine sequences for each of three subjects) taken under time varying illumination and where the subjects perform free head motion. These sequences were taken such that the first frame is not always at the maximum of the illumination. All of the sequences and the corresponding ground truth are available on-line: http://www.cs.bu.edu/groups/ivc/HeadTracking/. The reader is encouraged to visit the web site and watch them to have a precise idea of the typology of motion and illumination variation.

Note that the measured ground truth and the estimate of the visual tracker are expressed in two different coordinates systems. The estimated position is in a coordinate system that has its origin in the camera plane and is known only up to a scale factor. This is an *absolute orientation problem* [19] as we have two sets of measurements expressed in two coordinate systems with different position, orientation, and units. To avoid this problem, we carefully aligned the magnetic receiver and the camera such that the two coordinate systems were parallel (see Fig. 7). The scale factor in the three axis directions was then estimated using calibration sequences. All visual tracker estimates are then transformed according to these scale factors before comparison with ground truth data.

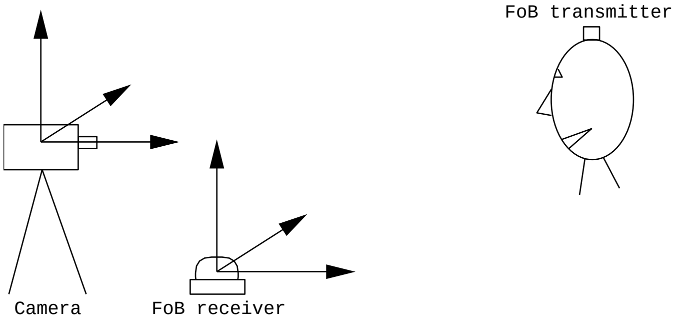

Figure 7: Camera and magnetic tracker coordinates systems. All the sequences were taken under this condition.

For the sake of comparing ground truth *vs.* estimated position and orientation, we can safely assume that at the first frame of the sequence the visual estimate is coincident with the ground truth. The graphs reported in Figs. 8 and 9 are based on this assumption.

7.2 Performance Measures

Once the coordinate frames of magnetic tracker and visual tracker are aligned, it is straightforward to define objective measures of performance of the system. We are mainly concerned about *stability* and *precision* of the tracker.

We formally define these measures as a function of the Mahalanobis distance between the estimated and measured position and orientation. The covariance matrices needed for the computation of the distance have been estimated over the entire set of collected sequences. In particular we define for any frame of the sequence two normalized errors:

$$e_{t,i}^2 = [\mathbf{a}_{t,i} - \tilde{\mathbf{a}}_{t,i}]^T \Sigma_t [\mathbf{a}_{t,i} - \tilde{\mathbf{a}}_{t,i}] \tag{31}$$

$$e_{r,i}^2 = [\mathbf{a}_{r,i} - \tilde{\mathbf{a}}_{r,i}]^T \Sigma_r [\mathbf{a}_{r,i} - \tilde{\mathbf{a}}_{r,i}], \tag{32}$$

where $e_{t,i}$ and $e_{r,i}$ are the error in the estimates of the translation and rotation at time i, The vectors $\mathbf{a}_{t,i}$ and $\mathbf{a}_{r,i}$ represent the visually estimated position and orientation at time i after the alignment to the magnetic tracker coordinate frame. The corresponding magnetically measured values for translation and rotation are represented by $\tilde{\mathbf{a}}_{t,i}$ and $\tilde{\mathbf{a}}_{r,i}$ respectively.

We can now define a measure of tracker stability in terms of the average percentage of the test sequence that the tracker was able to track before losing the target. For the sake of our analysis, we defined the track as lost when $e_{t,i}$ exceeded a fixed threshold. This threshold has been set equal to 2.0 by inspecting different sequences where the track was lost and the measuring the corresponding error as given by Eq. 32.

The precision of the tracker can be formally defined for each sequence as the root mean square error computed over the sequence up to the point where the track was lost (according to the definition of *losing track* from above). It is important to discard the part of the sequences after the track is lost as the corresponding estimates are totally insignificant and make the measure of the error useless. The positional and angular estimation error err_t and err_r for a particular sequence can then be expressed as:

$$err_t{}^2 = \frac{1}{N}\sum_{i=1}^{N} e_{t,i}^2 \tag{33}$$

$$err_r{}^2 = \frac{1}{N}\sum_{i=1}^{N} e_{r,i}^2 \tag{34}$$

where N is the number of frames tracked before losing the track. For some of the experiments, it is also useful to analyze the precision of the single components of the estimate that can be defined in a similar way.

8 System Evaluation

We evaluated our technique using the full set of sequences collected as described above. We compared the effectiveness of a texture mapped cylindrical model as opposed to a planar model. We also evaluated the effect of the lighting correction term. Finally, experiments were conducted to quantify sensitivity to errors in the initial positioning, regularization parameter settings, and internal camera parameters.

Three versions of the head tracker algorithm were implemented and compared. The first tracker employed the full formulation: a cylindrical model with illumination correction and regularization terms (Eq. 27). The second tracker was the same as the first cylindrical tracker, except without the

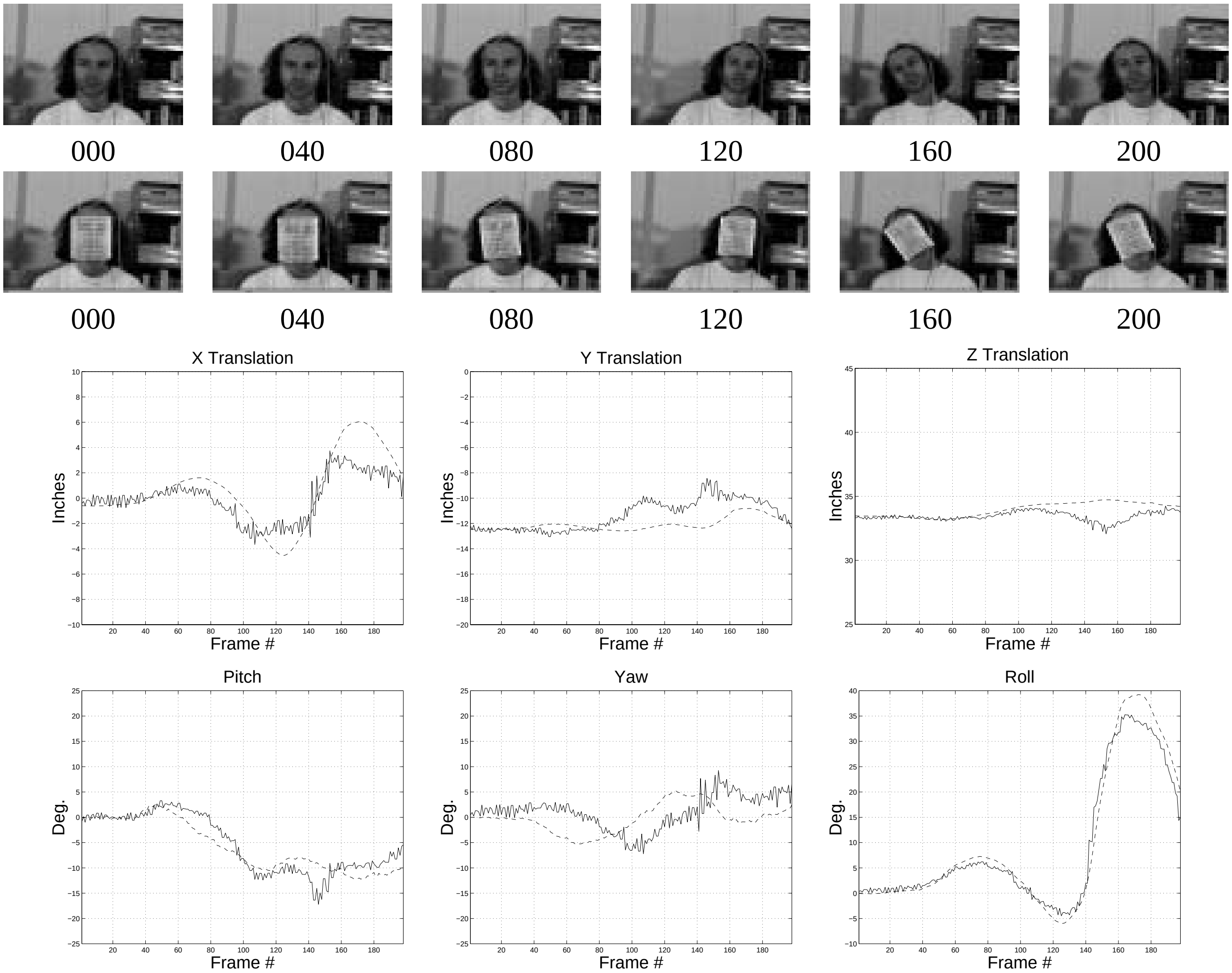

Figure 8: Example tracking sequence collected with uniform illumination. Ground truth was collected via a 3D “Flock of Birds” sensor. Images taken from test sequence and tracking with the cylindrical model are shown in the top rows. The transmitter for the magnetic tracker is visible in the image sequence; it is the small box with the incoming wire that hangs on the subject's head. The graphs depict estimated head parameters *vs.* ground truth. In all of the graphs, the dashed curve depicts the estimate gained via the visual tracker and the solid curve depicts the ground truth. The first row of graphs shows the x, y and z translation respectively, where translation is measured in inches. The second row of graphs shows estimates for rotation around the x, y, and z axes respectively, as measured in degrees.

illumination correction term. The third tracker utilized a 3D planar model to define the warping function $\Gamma(\mathbf{I}, \mathbf{a})$; this model was meant to approximate planar head tracking formulations reported in [5, 16]. Our implementation of the planar tracker included a regularization term, but no illumination correction term.

Before detailed discussion of the experiments, two examples of tracking will be shown. These

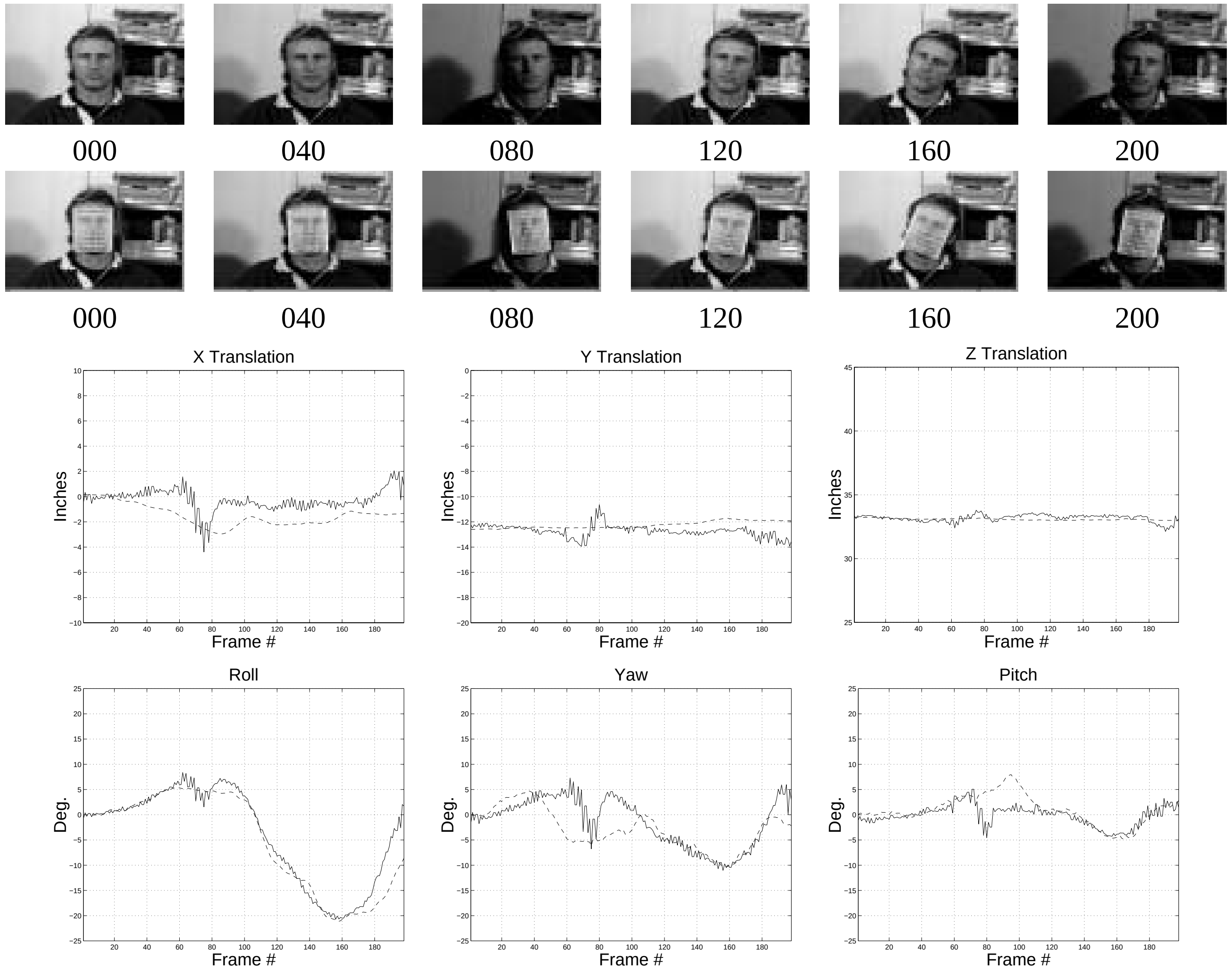

Figure 9: Example test sequence and tracking with time varying illumination. Frames taken from test sequence and tracking with the cylindrical model are shown in the top rows. The graphs depict estimated head parameters *vs.* ground truth. In all of the graphs, the dashed curve depicts the estimate gained via the visual tracker and the solid curve depicts the ground truth. The first row of graphs shows the x, y and z translation respectively, where translation is measured in inches. The second row of graphs shows estimates for rotation around the x, y, and z axes respectively, as measured in degrees.

are intended to give an idea of the type of test sequences gathered and the tracking results obtained.

In Fig. 8 a few frames from one of the test sequences are shown together with the tracking results. Three-dimensional head translation and orientation parameters were recovered using the full tracker formulation that includes illumination correction and regularization terms. The graphs show the estimated rotation and translation parameters during tracking compared to ground truth. The version of the tracker that used a planar model was unable to track the whole sequence without losing track.

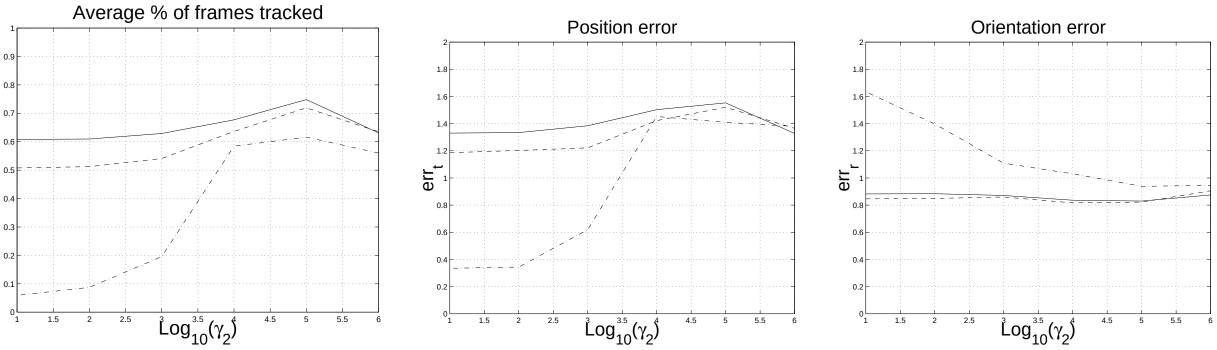

Figure 10: Experiment 1: sensitivity of head trackers to the regularization parameter γ_2. Average performance was determined over all the 45 sequences taken under uniform illumination. In each graph, the solid curve depicts performance for the cylindrical head tracker with illumination correction, the dashed curve depicts performance for the cylindrical tracker without the illumination correction, and the dash-dot curve depicts performance for the planar tracker. The first graph shows the average number of frames tracked over the full test set of 45 sequences, taken under uniform illumination. The second graph shows the average position error err_t. The third graph shows the average orientation error err_r. The unit on the horizontal axis of each graph is $\log_{10}(\gamma_2)$.

Fig. 9 shows a test sequence with varying illumination. Tracking results using illumination correction is shown together with ground truth. The version of the cylindrical tracker without lighting correction diverged around frame 60.

8.1 Experiment 1: General Performance of the Tracker

The first experiment was designed to test sensitivity of the three different trackers to variation in the warping regularization parameter γ_2. Multiple trials were conducted. In each trial, γ_2 was fixed at a value ranging from 10 to 10^6. At each setting of γ_2, the number of frames tracked and the precision of the trackers was determined for all sequences in the first dataset (45 sequences taken under uniform illumination). For all trials in this experiment the focal length $f = 10.0$, the global regularization parameter $\gamma = 1.0$, and $\gamma_1 = 10^5$.

Graphs showing average stability and precision for the different trackers are shown in Fig. 10. On each graph, the solid curve depicts performance for the full head tracker using the cylindrical model with illumination correction. The dashed curve depicts performance for the cylindrical tracker without the illumination correction. Finally, the dash-dot curve corresponds to the performance of the planar tracker. The horizontal axis is $\log_{10}(\gamma_2)$. The curves show average performance taken over all the 45 sequences.

The performance of the two cylindrical trackers (with and without the illumination term) is nearly identical. This is reasonable as the sequences used in this experiment where taken under uniform illumination; therefore, the lighting correction term should have little or no effect on tracking performance. In contrast, the planar tracker performed generally worse than the cylindrical trackers; performance was very sensitive to setting of the regularization parameter. Note also that the precision of the planar tracker's position estimate seems better for low values of γ_2 (smaller error). This is due to the error computation procedure that takes into account only those few frames that were tracked before track is lost. In our experience, when the tracker is very unstable and can track on average less than 50% of each the test sequences, the corresponding precision measure is not very useful.

8.2 Experiment 2: Lighting Correction

The second experiment was designed to evaluate the effect of the illumination correction term in performance of the cylindrical tracker. In this experiment, the second set of test sequences was used (27 sequences taken under time varying illumination conditions). For all the test sequences in the dataset, we computed the number of frames tracked and the precision of the tracker while varying γ_1 over the range of 10^2 to 10^9. For all trials in this experiment, the focal length $f = 10.0$, the global regularization parameter $\gamma = 1.0$, and $\gamma_2 = 10^5$.

The results of this experiment are reported in Fig. 11. In each graph, the solid curve depicts performance of the head tracker with illumination correction term. For comparison, the performance of the cylindrical tracker without the illumination correction term was tested, as shown by the dashed curve in the graph. Each graph shows the average performance over the full test set of 27 sequences taken under time varying illumination. The unit on the horizontal axis of each graph is $\log_{10}(\gamma_1)$. The first graph in Fig. 11 shows the average number of frames tracked before losing track, as determined by Eq. 32. The second graph shows the average position error err_t. The third graph shows the average error in estimating orientation, err_r.

As can be seen in the graphs, the stability of the tracker is greatly improved through inclusion of the illumination correction term. It is also interesting to note that the system is not very sensitive to the regularization parameter γ_1. For a wide range of values of this parameter performance is

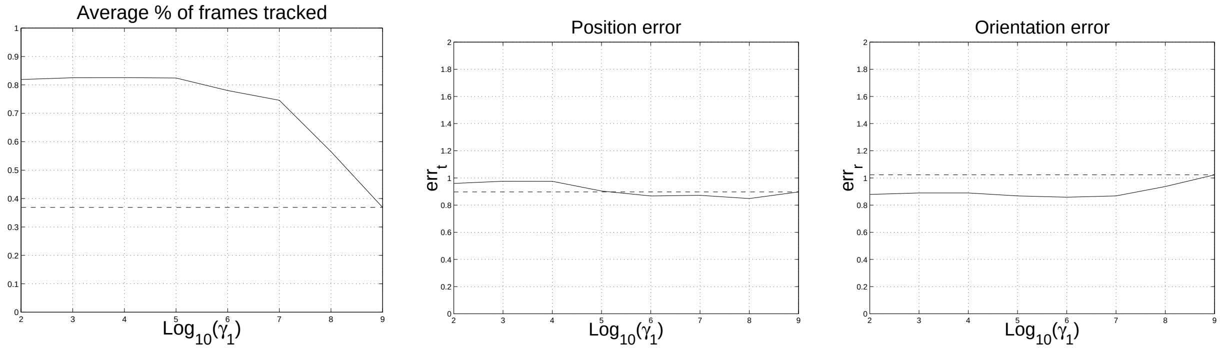

Figure 11: Experiment 2: sensitivity of the cylindrical head tracker to the illumination regularization parameter γ_1. Average performance was measured over a test set of 27 sequences taken under time varying illumination, as described in the text. In each graph, the solid curve depicts performance of the cylindrical tracker with illumination correction term. For comparison, performance of the cylindrical tracker without illumination correction is reported (shown as dashed curve). The first graph shows the average number of frames tracked before loosing track. The second graph shows the average position error err_t, while tracking (before loosing track). The third graph shows the average error in estimating head orientation err_r, while tracking (before loosing track). The unit on the horizontal axis of each graph is $\log_{10}(\gamma_1)$.

approximatively constant, with performance dropping to the level of the non-illumination corrected tracker only when over-regularizing.

In this experiment, the precision of the tracker does not seem improved by the lighting correction. This is reasonable as the precision is averaged only over those frames before losing the track of the target. The tracker without lighting correction is as good as the one using the lighting correction up to the first change in illumination; at that point the non-illumination corrected model usually loses the track immediately while the illumination-corrected model continues tracking correctly.

8.3 Experiment 3: Sensitivity to Initial Positioning of the Model

Experiments were conducted to evaluate the sensitivity of the tracker to the initial placement of the model. Given that our system is completely automatic and that the face detector we use [29] is sometimes slightly imprecise, it is important to evaluate if the performance of the tracker degrades when the model is initially slightly misplaced. The experiments compared sensitivity of the planar tracker *vs.* the cylindrical tracker.

Experiments were conducted using the test set of 45 sequences, taken under uniform illumination. Three sets of experimental trials were conducted. Each set tested sensitivity to one parameter that is estimated by the automatic face detector: horizontal position, vertical position, scale. In each trial, the automatic face detector's parameter estimate was altered by a fixed percentage: $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$. Over all the trials, the other parameters were fixed: $f = 10.0$, $\gamma = 1.0$, and $\gamma_1 = \gamma_2 = 10^5$.

In the first set of trials, we perturbed the horizontal head position. by $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$ the estimated face width. The graphs in Fig. 12 show the stability and precision of the two head trackers, as averaged over all 45 test sequences. In each graph, the solid curve depicts performance of the cylindrical tracker and the dashed curve depicts performance of the planar tracker. The first graph shows the average number of frames tracked before loosing track. The second graph shows the average error in estimating err_t, as obtained during parts of the sequences in which tracking was not lost. The third graph shows the error err_r in estimating head orientation, averaged over parts of the sequences in which tracking was not lost. The horizontal axis shows the percentage offset in the detected x position of the face.

Similarly, in the second set of trials, we perturbed the vertical head position by $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$ the estimated face height. The graphs in Fig. 13 show the performance of the two trackers, as averaged over all 45 test sequences. As before, the solid curve in each graph depicts performance of the cylindrical tracker and the dashed curve depicts performance of the planar tracker.

Finally, in the third set of trials, we measured performance of the system when varying the initial size of the detected face. This was meant to evaluate sensitivity of tracking to errors in estimating the initial head scale. Fig. 14 shows graphs of performance of both trackers under such conditions.

As expected, the planar tracker is almost insensitive to perturbations of the initial positioning of the model. The cylindrical tracker, which out performed the planar model in all previous experiments in terms of precision and stability, is also not very sensitive to errors in the initial positioning of the model. This is a very interesting behavior as the main limitation of more detailed 3-D head trackers [10, 13] is the need for a precise initialization of the model. At present, such precise initialization cannot in general be performed in fast or automatic way.

Finally, it should be noted that these experiments were conducted by perturbing only one parameter at the time. In informal experiments, perturbing simultaneously the horizontal position,

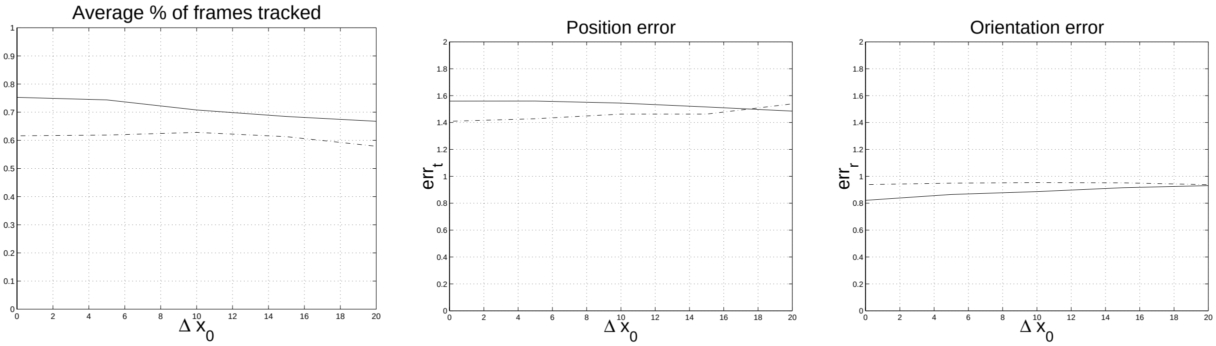

Figure 12: Experiment 3: sensitivity of cylindrical and planar tracker to errors in estimating horizontal position of face. The horizontal position was perturbed by $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$ of the face width. The graphs show performance of the planar *vs.* the cylindrical tracker, as averaged over all 45 test sequences as described in the text. In all the graphs the solid curve corresponds to the performance of the cylindrical tracker, and the dashed curve to the planar tracker. The horizontal axis of each graph is the amount of perturbation added to the x position of the head. The first graph shows the average number of frames tracked before track was lost. The second graph shows the average error in estimating head position err_t. The third graph shows the average error in estimating head orientation.

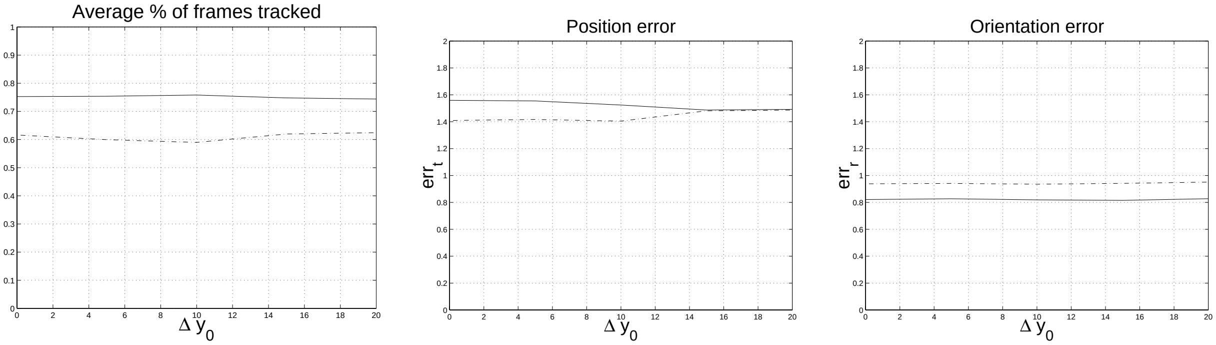

Figure 13: Experiment 3 (continued): sensitivity of cylindrical and planar tracker to errors in estimating vertical position of face, as described in the text. In all the graphs the solid curve corresponds to the performance of the cylindrical tracker, and the dashed curve to the planar tracker. The horizontal axis of each graph shows the offset added to the initial y position of the head. The first graph shows the average number of frames tracked before track was lost. The second graph shows the average error in estimating head position err_t. The third graph shows the average error in estimating head orientation.

the vertical position and the size of the estimated face, yielded similar results.

8.4 Experiment 4: Sensitivity to Focal Length

In our system the focal length is implicitly embedded in the warping function $\Gamma(\mathbf{I}, \mathbf{a})$ of Eq. 6. The focal length is not estimated but it is assumed to be known. This experiment was intended to determine how the performance of the tracker is affected by the choice of the focal length.

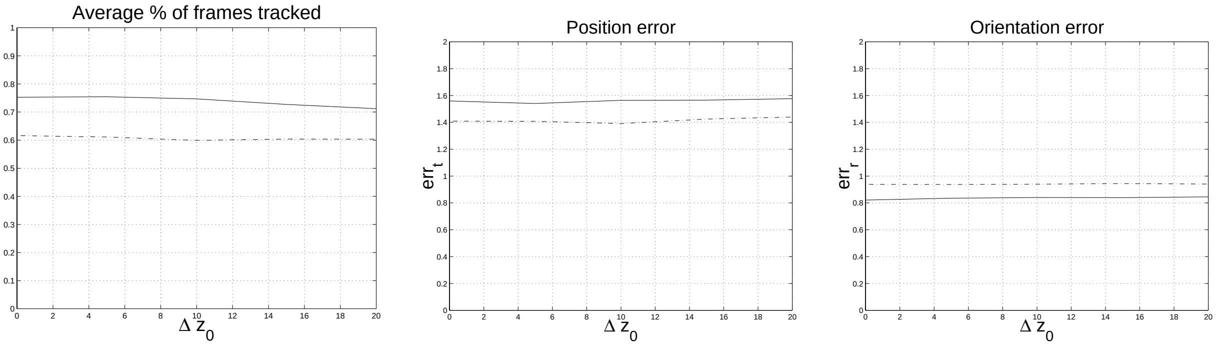

Figure 14: Experiment 3 (continued): Sensitivity of cylindrical and planar tracker to errors in estimating the scale of the face. Estimated head scale was perturbed by $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$. The graphs show performance of the planar *vs.* cylindrical trackers, as averaged over all 45 test sequences as described in the text. In all the graphs the solid curve corresponds to the performance of the cylindrical tracker, and the dashed curve to the planar tracker. The horizontal axis of each graph is the amount of perturbation added to the head scale estimate. The first graph shows the average number of frames tracked before track was lost. The second graph shows the average error in estimating head position $err_{\vec{t}}$ while tracking. The third graph shows the average error in estimating head orientation.

We computed stability and precision for the 45 test sequences taken under uniform illumination conditions using focal length equal to $2, 4, 8, 16, 32$, and 64. The results of this experiment are reported in Fig. 15. In all the graphs the solid curve corresponds to the cylindrical tracker and the dashed line to the planar tracker. For all the trials in this experiment the regularization parameters were fixed: $\gamma = 1.0$, $\gamma_1 = \gamma_2 = 10^5$.

The average number of frames tracked is reported in the top graph in Fig. 15. The precision of the trackers in estimating translation and rotation is reported in the other graphs. For this experiment we reported the precision with respect to the different parameters, as there are significant differences in precision between them. The error graphs for translation along the three axes x, y and z are reported respectively in the second row of Fig. 15. Graphs of error in the estimated rotation are shown in the bottom row of Fig. 15.

Note that the planar tracker is relatively insensitive to the assumed focal length; the only component adversely influenced was the estimate of the depth when the focal length becomes too long. Similarly, the cylindrical tracker was somewhat sensitive for very short focal lengths and also tended to mis-estimate the depth as the focal length became too long.

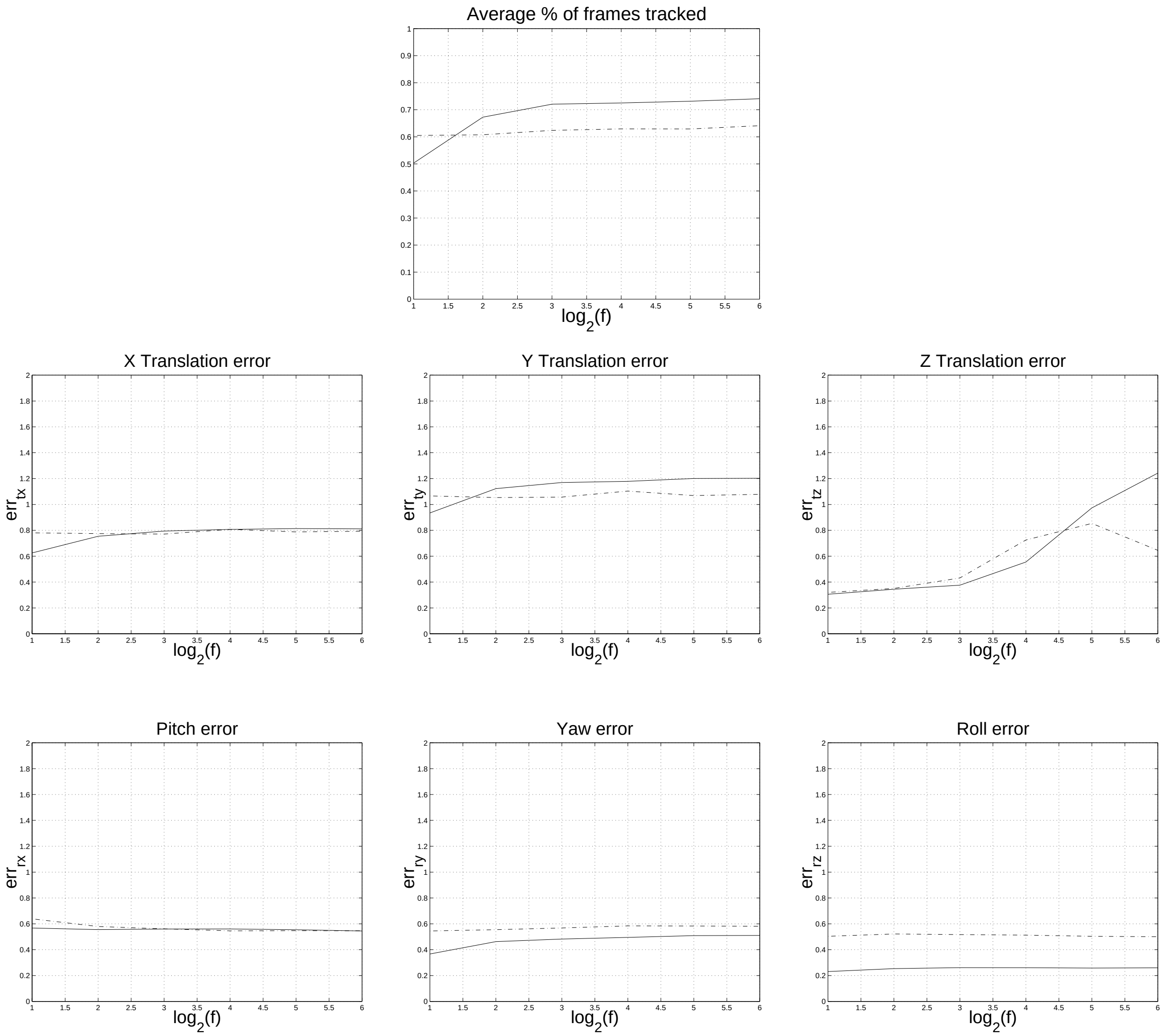

Figure 15: Experiment 4: sensitivity of cylindrical and planar tracker to the focal length. In all the graphs, the solid curve corresponds to the cylindrical tracker and the dashed curve to the planar tracker. The first (top) graph shows the average number of frames tracked over the set of 45 sequences, while varying focal length. The next three graphs show the average error in estimating the x, y and z position, respectively. The bottom three graphs show the average error in estimating the rotation about the x, y and z axes, respectively.

9 Discussion

The experiments indicate that the cylindrical model generally allows tracking of longer sequences than when using a planar model. Furthermore, it allows us to estimate more precisely the 3D rotations of the head. The error in the estimates of the position is on average slightly smaller when using the planar tracker. This is not surprising as the planar tracker can accurately estimate the position of the head but tends to lose the target as soon as there is some significant out of plane rotation. The difference in the behavior of the two trackers is even more evident in the interactive use of the system. Moreover, the cylindrical tracker is much less sensitive to the regularization parameter.

The use of an illumination correction term was shown to greatly improve the performance of the system in the case of sequences taken under time-varying illumination. Furthermore, the experiments indicated that the choice of the regularization parameter is not critical and the performance of the system remains approximately constant in a wide range of variability.

As exhibited in the experiments, the system is relatively insensitive to error in the initial estimate of the position and scale of the face. The precision and stability of the tracker remain approximately constant for a range of initialization errors up to 20% the size of the face detected. It is also interesting to note that the focal length used in the warping function did not seem to be a critical parameter of the system in the experiments. In practice, we have found that this parameter can be chosen very approximately without particular difficulties.

The experiments confirmed our hope that our tracker could overcome the biggest problem of a planar tracker (instability in presence of out of plane rotations) without losing its biggest advantages (small sensitivity to initialization errors and low computational load).

Beyond the quantitative testing reported in the previous section, we analyzed qualitatively the behavior of our technique through interactive use of the real-time version of the system. This analysis coherently confirmed the strengths and weaknesses that emerged from the quantitative testing.

In most cases, the cylindrical tracker is stable and precise enough to be useful in practical applications. For example in an informal experiment we tried to control the mouse pointer with small out of plane rotations of the head. After a few minutes of training the subjects were able to control the pointer all over the computer screen with a precision of about 20-30 pixels. The head tracker has also been successfully tested in head gesture recognition, and expression tracking [22].

We also analyzed which are the most common cases when the tracker fails and loses the target. We noticed that all of the cases where the target was lost were due to one of the following reasons: 1) motion was too fast, 2) simultaneous large rotation around the vertical axis and large horizontal translation, 3) simultaneous large rotations around the vertical and the horizontal axis, 4) very large rotation around the vertical axis.

The first failure mode can easily be addressed, through the use of higher resolution input video, and then more levels in the Gaussian pyramids. Alternatively, this problem could also be addressed by using higher frame rate video input. The second source of instability is due to the general aperture problem. This ambiguity is very well highlighted in Fig. 6 as an off diagonal element in the matrix $\mathbf{M}^T\mathbf{M}$. The use of a regularization term greatly reduced this problem.

The other failure modes are due mainly to the fact the head is only approximated by a cylinder. This sometimes causes error in tracking large out-of-plane rotations of the head. As stated earlier, using a more detailed, displacement-mapped model did not seem to improve tracking substantially; the resulting tracker tended to have greater sensitivity to initialization in our informal experiments. A more promising approach for coping with large out-of-plane rotations would be to use more than one camera in observing the moving head.

10 Summary

In this paper, we proposed a fast, stable and accurate technique for 3D head tracking in presence of varying lighting conditions. We presented experimental results that show how our technique greatly improves the standard SSD tracking without the need of a subject-dependent illumination basis or the use of iterative techniques. Our method is accurate and stable enough that the estimated pose and orientation of the head is suitable for applications like head gesture recognition and visual user interfaces.

Extensive experiments using ground truth data showed that the system is very robust with respect to errors in the initialization. The experiments also showed that the only parameters that we had to choose arbitrarily (the regularization parameters and the focal length) do not affect dramatically the performance of the system. Using the same parameter settings, the system can easily track sequences with different kinds of motion and/or illumination.

The texture map provides a stabilized view of the face that can be used for facial expression recognition, and other applications requiring that the position of the head is frontal view and almost static. Furthermore, the formulation can be used for model-based very low bit-rate video coding of teleconferencing video. Moreover, the proposed technique utilizes texture mapping capabilities that are common on entry level PC and workstations, running at NTSC video frame rates.

Nevertheless, our technique can still be improved on several fronts. For example we believe that the use of two cameras could greatly improve the performance of the tracker in presence of large out of plane rotations. In the future we also plan to develop a version of our method that employs robust cost functions [31]. We suspect that this could further improve the precision and stability of the tracker in presence of occlusions.

References

[1] *The Flock of Birds*. Ascension Technology Corporation. P.O. Box 527, Burlington, VT 05402.

[2] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 15(6), 1993.

[3] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Proc. International Conference on Pattern Recognition, Wien, Austria*, 1996.

[4] S. Birchfield. An elliptical head tracker. In *Proc. of 31st Asilomar Conf. on Signals, Systems and Computers*, November 1997.

[5] M.J. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[6] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. In *Proc. 5th International Conference on Computer Vision, Boston, MA*, 1995.

[7] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 1997.

[8] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance model. In *Proc. 5th European Conference on Computer Vision, Freiburg, Germany*, 1998.

[9] J.L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico*, 1997.

[10] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, CA*, 1996.

[11] F. Dellaert, C. Thorpe, and S. Thrun. Super-resolved texture tracking of planar surface patches. In *Proc. of IEEE/RSJ International Conference on Intelligent Robotic Systems, Victoria*, 1998.

[12] F. Dellaert, S. Thrun, and C. Thorpe. Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In *Proc. of IEEE Workshop on Applications of Computer Vision, Princeton, NJ*, 1998.

[13] I.A. Essa and A.P. Pentland. Coding analysis, interpretation, and recognition of facial expressions. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 19(7):757–763, 1997.

[14] P. Fieguth and D. Terzopoulos. Color based tracking of heads and other mobile objects at video frame rates. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico*, 1997.

[15] M. Gleicher. Projective registration with difference decomposition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico*, 1997.

[16] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[17] P. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, WA*, 1994.

[18] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.

[19] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4), April 1987.

[20] T. Horprasert, Y. Yacoob, and L.S. Davis. Computing 3-d head orientation from a monocular image sequence. In *Proc. International Conference on on Face and Gesture Recognition, Killington, VT*, 1996.

[21] T.S. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptative feedback tracking of faces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico*, 1997.

[22] M. La Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA*, 1998.

[23] M. La Cascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, CO*, 1999.

[24] H. Li, P. Rovainen, and R. Forcheimer. 3-d motion estimation in model-based facial image coding. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 15(6):545–555, 1993.

[25] P. Lipson, A. Yuille, D. O. Keefe, J. Cavanauch, J. Taffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In *Proc. European Conference on Computer Vision*, LNCS 427, pages 413–417. Springer-Verlag, 1990.

[26] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 19(7), July 1997.

[27] N. Olivier, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico*, 1997.

[28] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer-Verlag, New York, 1984.

[29] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 20(1):23–28, 1998.

[30] A. Schödl, A. Haro, and I. Essa. Head tracking using a textured polygonal model. In *Proc. of 1998 Workshop on Perceptual User Interfaces*, 1998.

[31] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, 1998.

[32] A. Shashua. *Geometry and photometry in 3D visual recognition*. PhD thesis, MIT, 1992.

[33] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 8(2):129–139, 1986.

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

[35] Y. Yacoob and L.S. Davis. Computing spatio-temporal representations of human faces. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 18(6):636–642, 1996.

[36] A.L. Yuille, D.S. Cohen, and P.W. Hallinan. Feature extraction from faces using deformable templates. In *Proc. International Conference on Pattern Recognition, Jerusalem*, 1994.