MODEL−BASED HEAD POSE ESTIMATION FOR AIR−TRAFFIC CONTROLLERS

Xavier L.C. Brolly, Constantinos Stratelos and Jeffrey B. Mulligan

NASA Ames Research Center
Mail Stop 262−2, Moffett Field, CA 94035−1000

ABSTRACT

We present a method for estimating the point of fixation of an air traffic controller from a low resolution video sequence. A geometric model of the head is used to estimate head orientation; head pose estimates are combined with a 3D model of the environment to compute the target of gaze. The head model is constructed from a small set of images. Two methods are considered: in the first, we treat the head as a textured object and ignore lighting effects; in the second, we jointly estimate the albedo of each facet of the head model, and the parameters of a simple lighting model. Because ground-truth data are unavailable, the absolute accuracy of the gaze estimates is unknown, but incorporation of the lighting model does appear to reduce the noise level. With either method, the results are sufficiently accurate to answer questions of operational interest, such as "is the controller looking out the window."

1. INTRODUCTION

Gaze-tracking is an important component of behavioral analyses in a number of application areas. We are interested in the problem of air-traffic control displays. Tower-based ground controllers rely both on computer displays, and direct out-the-window views of the runways and taxiways. When a change is made to the user interface of the computer system, we would like to know how it affects controller behavior, and, ultimately, the safety of the system. A simple measure is how much time is spent fixating the display, versus objects out the window. Of course, increased time spent fixating the display could mean a number of things: it might mean that the display is hard to understand and therefore requires more study (a situation we would like to correct), or it might mean that the display has been improved and can deliver more information than the out-the-window view (a situation we would like to achieve). Discrimination between these alternatives will be left to the experts and designers of the interfaces; our task is merely to provide the raw gaze data for their consideration.

Gaze tracking is most often done by imaging the eyes themselves. This approach provides the most accurate estimates of gaze, but imposes requirements that are impractical in applied settings. We have therefore concentrated our efforts on estimating the "head gaze" of the controller, as observed from a remote wide-field camera. Unlike previous approaches to head coding for video telephony [1], the head is a relatively small part of our images, subtending a mere 30 pixels or so. For our initial efforts, we have used a short sequence of video collected in the Future Flight Central control tower simulator at NASA Ames Research Center. In the remainder of the paper, we describe the methods we have applied to video-based estimation of head gaze, and present our results.

2. ESTIMATION OF HEAD POSE

To obtain the location of the head in each image, we applied a simple correlation-based template-matching approach. While this method sufficed to get us started, it is not particularly robust. More sophisticated methods have been proposed [2], which we hope to incorporate in the future.

Our approach to head-pose estimation is an iterative one, using the analysis-by-synthesis method. We construct a textured model of the subject's head which we can manipulate and render in any orientation. We then search for the orientation which maximizes the similarity between the rendered model and the input image. Direct measurement of head shape is not an option, because the video was recorded in the past and the subjects are no longer available. We are therefore primarily interested in systems that construct head models from sequences of images [1] [3].

Photo-realistic modeling of the head requires knowledge of its shape, pigmentation (texture), and the lighting conditions. Recovery of any one of these components is relatively easy if the other two are known exactly [4] but this is rarely the case.

2.1. Head shape

Ultimately, we would like to have a fully automated way of generating head shapes and textures from a small set of images. As of this writing, however, the implementation of our shape optimizer is not complete, and we therefore present results obtained using a generic head model adjusted man-

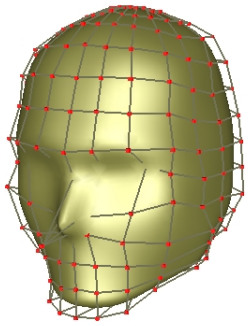

Fig. 1. The 3D NURBS head shape model

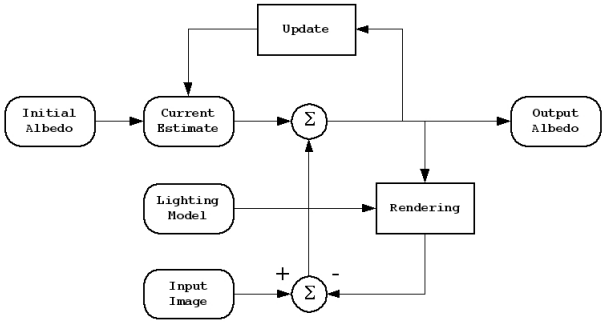

Fig. 2. Single frame albedo estimation procedure

ually to approximate the subject's head (Figure 1). This is reasonable given the low resolution of our source imagery. Head shape was described using Non-Uniform Rational B-Splines - NURBS. Once the head model was constructed, we stored the resulting vertices for further use. We treat the head as a rigid object, ignoring facial expression changes.

2.2. Texture

We have explored two approaches to texturing the surface of the head model. Initially, we chose to ignore lighting effects, lumping surface reflectance and illumination into a single texture color, which we rendered without enabling lighting effects. Because the texture is estimated from many views, non-generic features such as specular highlights are averaged out and do not become part of the stored texture. Therefore a separate procedure was used to estimate the surface reflectance, or albedo. This is discussed in detail in a later section.

2.3. Pose estimation

When the head model shape and the complete texture information become available, we can render the head at any orientation, position and scale. Therefore, provided that the model is accurate enough, we should be able to match the image produced by it with the target if the correct pose parameters are used. Using the STEPIT package [5], we try to find the 6 optimum parameters - 3 orientation angles, 2 position displacements and a scaling factor - that will produce a synthetic image matching the target. While computationally expensive, STEPIT has certain advantages over approaches that rely upon linearization of the problem [6], in that large changes in orientation can be successfully tracked.

3. ESTIMATION OF ALBEDO AND LIGHTING

We use a set of training images that were selected in such a way as to provide enough information to get a comprehensive texturing of the entire head . Those images where extracted from the input video.

3.1. Albedo

Given the head model and a pose corresponding to a particular training image, extraction of the color information from the image is done by sampling pixel intensities at the projection of the vertices onto the image plane of the input.

From this initial texture mapping, assuming a known lighting and head shape, we want to find the underlying albedo.

Skin properties were assumed uniform, and lighting conditions (ambient, diffuse, and position) constant. But the resulting color of each skin vertex also depends on the orientation of the facet it belongs to. So the effect of lighting on the head color is not additive. Therefore, to erase the lighting from the extracted texture, a simple subtraction cannot be done. Instead, we decided to use an analysis-by-synthesis method (Figure 2).

Once we have extracted the albedo from an initial training image, and mapped it onto the model, we obtain a partially textured model of the head. This model can be useful to estimate the head pose on other training images, that are close enough to the initial one. Having a new training image and its corresponding pose another texture extraction and albedo estimation can be performed as explained earlier.

Once a new albedo is extracted, a merging step is performed. A weighted average of the albedo color at each vertex is computed.

For N_{views} and M_{vert} vertices

$$a_i = \frac{\sum_{j=1}^{N_{views}} w_{ij} a_{ij}}{\sum_{j=1}^{N_{views}} w_{ij}} \quad i = 1, 2, \ldots, M_{vert} \qquad (1)$$

where a_i is the albedo color at vertex i, w_{ij} is the weight at vertex i for the view j and a_{ij} is the sampled albedo color at vertex i at view j.

The weight w_{ij} assigned to vertex i at view j is proportional to the length of the depth component of the normal to the facet that contains the vertex.

$$w_{ij} \quad = \quad max(-n_{ij}^T \, e_z, 0) \tag{2}$$

For each view of the head, the weighting of a particular vertex will be different. Vertices that belong to facets that are more viewable by the camera are weighted higher.

3.2. Lighting

When the skin color for a given lighting has been optimized (as explained in the previous part), we optimize the lighting parameters for a given albedo. We assumed the subject's head was illuminated by a unique point source described by 5 parameters - ambient light, diffuse light and relative position to the head that stayed constant during the whole movie sequence. Those parameters were estimated using an analysis-by-synthesis method, that minimizes the error between the synthetic images and the training ones.

3.3. Initialization

An initial albedo and lighting configuration are required to bootstrap the algorithm. A training image showing the head in frontal view was selected, and the model pose was adjusted manually to match. We chose to initialize pose as the frontal one, as it would render the albedo having the biggest number of features (eyes, nose, mouth) that could be useful for further matching.

The frontal view of a person mainly shows skin. To get an initial estimate of the lighting, we assume a uniform gray albedo, set the head model in the pose found manually, and then looked for the optimum lighting parameters.

Once having those lighting conditions, we can estimate the albedo of the first training image (quasi-frontal view) and use it as the initial albedo of the algorithm depicted in Figure 3.

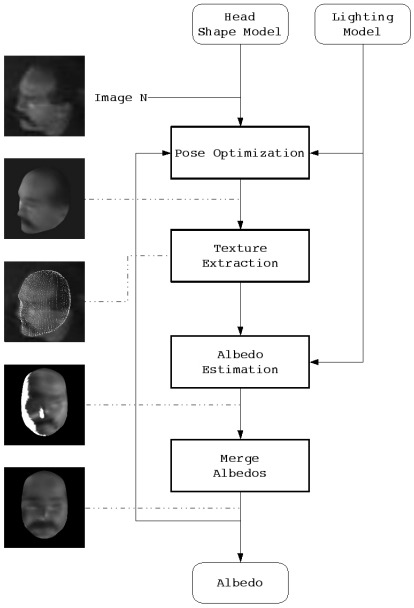

Fig. 3. Albedo estimation procedure

4. GAZE ESTIMATION

The pose or orientation of the head is not sufficient by itself to determine the target of gaze. The head must also be located within the three-dimensional scene. Also, in order to describe the target of gaze in a meaningful form, it is necessary to construct a three dimensional model of the surfaces in the subject's environment, and label the objects within it.

We constructed a three-dimensional model of the interior of the control tower simulator, using data from architectural drawings and direct measurement. We then estimated the intrinsic and extrinsic camera parameters necessary to align a rendering of the model with the image data (Figure 5). Once the correspondence between the image data and the scene model has been established, the surfaces of the scene model can be textured with data extracted from the video images in much the same way that the head model was textured. Novel views of the scene can then be rendered using the model.

Because we have only a single view of the scene, the depth of the subject's head is somewhat ambiguous. This ambiguity was resolved by assuming that the subject's head remained at a constant distance from the floor. With this assumption, the location in the three-dimensional scene is determined by the two-dimensional position in the image. The gaze vector can then be cast from the head location and intersected with the surfaces in the scene model. Labeling of regions in the scene surfaces allows categorization of the gaze target (display, window, papers, etc.).

5. RESULTS

Figure 4 shows estimates of azimuth for a 100 frame segment of the video. The heavy line shows the estimates obtained with the unlit, texture-mapped model, while the light line shows the estimates obtained when lighting compensation is included. The estimates obtained with lighting compensation exhibit significantly less noise. The fact that the two traces seem to be tracking different means implies that at least one of the traces has a bias; precise determination of the biases inherent in each method must be deferred until a data set with ground truth data is obtained.

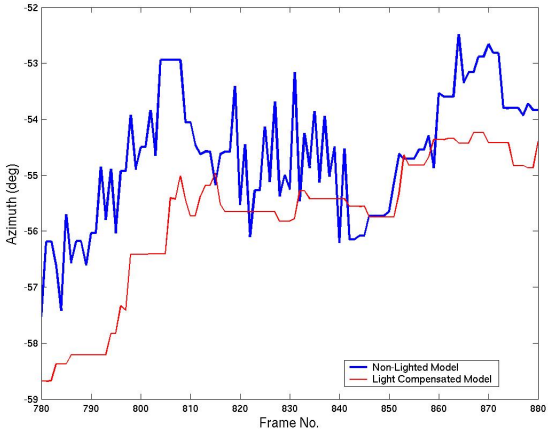

Fig. 4. Estimates of azimuth angle of head over 100 frames

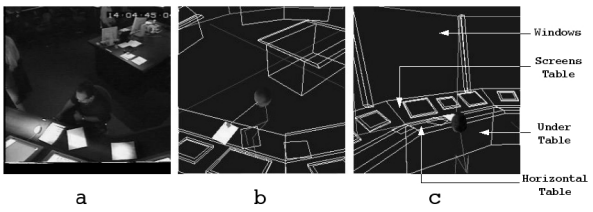

Fig. 5. (a). Original video frame. (b) Simulated camera view. (c) Simulated back view.

Figure 6 shows the relative fixation times for various objects in the scene. Estimates computed without lighting compensation are shown on the left, while those computed with lighting compensation are shown on the right. The main difference is the large percentage of fixations estimated to be below the table when lighting compensation is not applied. Because it is unlikely that the subject was actually looking beneath the table, this is evidence that the lighting compensation has improved the accuracy of the procedure. Furthermore, it is unlikely that this difference results from a simple downward bias in the absence of lighting compensation, because there is also a larger proportion of fixations on the windows, which are the highest objects in the scene.

6. DISCUSSION

We have demonstrated the recovery of crude gaze information using head pose recovered from low-resolution video data. While we have yet to match the performance existing methods have obtained with high-quality images, the results are nonetheless sufficiently accurate to be useful for

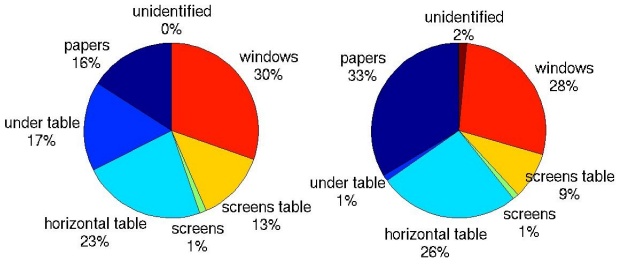

Fig. 6. Distribution of object fixations, estimated both without (left) and with (right) lighting compensation.

automated behavioral analyses. Lighting compensation improves the quality of the results both in model construction and pose estimation.

7. REFERENCES

[1] T. Wiegand P. Eisert and B. Girod, "Model-Aided Coding: A New Approach to Incorporate Facial Animation into Motion-Compensated Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, No. 3, pp. 344–358, 2000.

[2] D.J. Fleet A.D. Jepson and T. El-Maraghi, "Robust, on-line appearance models for vision tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 415–442, 2001.

[3] V. Blanz, S. Romdhani and T. Vetter, "Face Identification Across Different Poses and Illumination with a 3D Morphable Model," *Proc. IEEE Conference on Automatic Face and Gesture Recognition*, pp. 202–207, 2002.

[4] P. Eisert and B. Girod, "Model-Based 3D-Motion Estimation with Illumination Compensation," *Proceedings 6th International Conference on Image Processing and its Applications (IPA 97)*, pp. 194–198, 1997.

[5] J. D. Chandler, "Subroutine STEPIT: Finds local minima of a smooth function of several parameters," *Behavioral Science*, vol. 14, pp. 81–82, 1969.

[6] E. Steinbach P. Eisert and B. Girod, "Automatic Reconstruction of Stationary 3D Objects from Multiple Uncalibrated Camera Views," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 261–277, 2000.