

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
*San Jose State University
[*broollyx costrat jbm*]@eos.arc.nasa.gov

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 lighting models are considered: in the first, we only use ambient lighting; in the second, we add a finite distance point source. In both cases, we jointly estimate the albedo of each facet of the head model and the parameters of the lighting model. Because ground-truth data are unavailable, the absolute accuracy of the gaze estimates is unknown. 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 esti-

mates 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. HEAD POSE ESTIMATION

Photo-realistic modeling of the head requires knowledge of its shape, pigmentation (albedo) and the lighting conditions. Recovery of any one of these components is relatively easy if the other two are known exactly [2] but this is rarely the case.

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 [3], 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 position and 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] [4].

2.1. Head shape

Ultimately, we would like to have a fully automated way of generating head shapes and albedos from a small set of im-

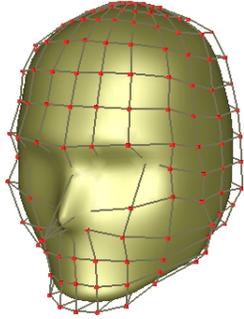


Fig. 1. The 3D NURBS head shape model

ages. 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 manually to approximate the subject’s head (Figure 1). This is reasonable given the low resolution of our source imagery. Head shape was described using NURBS (Non-Uniform Rational B-Splines). 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. Albedo and lighting

Two different lighting models were used to render the synthetic head. For each of the models a corresponding albedo was computed.

The first model (model A) consisted of an ambient light. In that case, surface reflectance and illumination were lumped together into a single albedo. The simplicity of the model doesn’t require lighting effects to be enabled while rendering, and the albedo was directly extracted from the image intensity values.

The second model (model B) contained an ambient illuminant as well as a single point omni-directional source at finite distance. This additional light source allows the model to fit better the lighting of the actual environment, which is not just ambient. However, it requires the estimation of additional parameters: position and brightness of point source. When using this model, the head is rendered with all lighting effects.

2.3. Pose estimation

When the head model shape and the complete albedo 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



Fig. 2. Training set used for albedo estimation

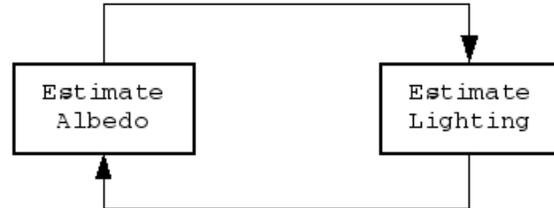


Fig. 3. Iterative estimation of albedo and lighting

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. ALBEDO ESTIMATION

A set of training images was selected in such a way as to provide enough information to get a comprehensive texturing of the entire head (Figure 2). Those images were extracted from the input video.

For model A, there are no lighting parameters and the albedo is estimated by a sampling of the pixel intensities.

For model B, we use an iterative approach to jointly estimate skin albedo and lighting geometry. We first estimate the parameters of this lighting model – 3 position parameters of the point source, brightness of point source, brightness of ambient light – using a frontal view of the face and a constant skin albedo of 0.5. The skin is assumed to be Lambertian with no specular component. We then use STEPIT to find the parameters of the lighting model which maximize the correlation of the rendered model with the image data.

Now, having initial estimates of both the albedo and the lighting, we iteratively refine both estimates in alternation (Figure 3). To update the estimate of the albedo, we revisit each image in the training set. For each image, we use an estimate of the pose manually input by an operator. We attempted to use automatic pose estimation, but perhaps because of errors in the shape and lighting model, this was unstable.

Given an estimate of the pose corresponding to a particular training image, we update our estimate of the albedo by projecting the model vertices onto the image plane and sampling the corresponding pixel intensities in the input and

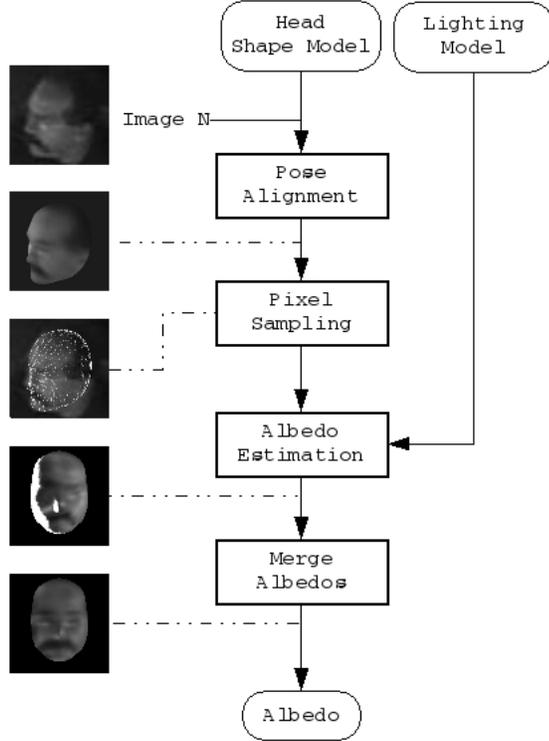


Fig. 4. Albedo estimation procedure

rendered images. Albedo is recovered from the sampled intensities by factoring out the lighting effects as follows:

$$a_{n+1} = \frac{\mathcal{I}_{\text{input}}}{\mathcal{I}_{\text{rendered}}} a_n \quad (1)$$

a_n being the albedo coefficient at iteration n and \mathcal{I} the intensity.

Once a new albedo is extracted from one of the images, it is merged with the overall estimate derived from the previously processed images. A weighted average of the albedo color at each vertex is computed:

$$a_i = \frac{\sum_{j=1}^N w_{ij} a_{ij}}{\sum_{j=1}^N w_{ij}} \quad i = 1, 2, \dots, M \quad (2)$$

where a_i is the albedo color at vertex i , w_{ij} is the weight at vertex i for the view j , a_{ij} is the sampled albedo color at vertex i at view j , N is the number of different views and M the number of vertices.

Facets that are seen obliquely will be weighted less than the ones that are more nearly normal to the line of sight. The weight w_{ij} assigned to vertex i at view j is proportional to the length of the depth component of the normal n_{ij} to the facet that contains the vertex.

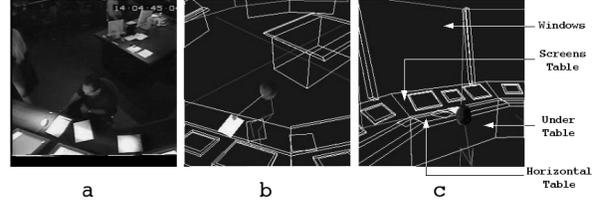


Fig. 5. (a). Original video frame. (b) Simulated camera view. (c) Simulated back view.

$$w_{ij} = \max(-n_{ij}^T e_z, 0) \quad (3)$$

where e_z is the unit vector of the direction of the line of sight.

After each update of the albedo estimate (procedure shown in Figure 4), we reestimate the lighting parameters.

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 3D scene. Also, in order to describe the target of gaze in a meaningful form, it is necessary to construct a 3D model of the surfaces in the subject’s environment, and label the objects within it.

We constructed a 3D 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 3D scene is determined by the 2D 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

Qualitatively, we noticed that the simple ambient lighting model gave better estimates of the head orientation than the

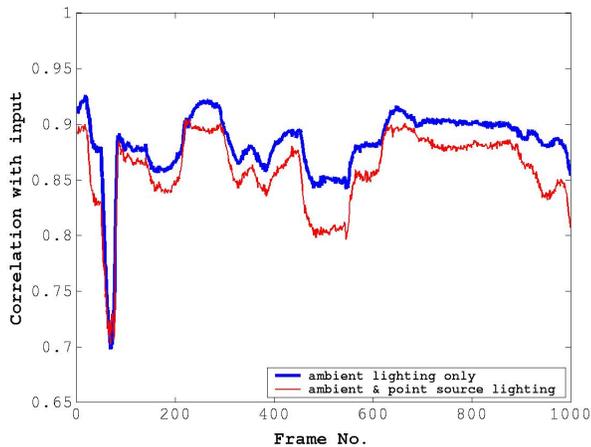


Fig. 6. Normalized correlation between input head and synthetic one

one with the point source. Figure 6 shows the normalized correlation between the input image head and the rendered one for a 1000 frame segment of the video. We see that the correlation obtained with the simple model is always greater than the one obtained when the point source is included. So the simple model creates synthetic head images that better match the input image. This can be explained by the fact that the additional point source didn't represent the actual lighting accurately. The real head in the actual environment was lit by several extended sources (3 LCD screens and 3 large windows in front of the operator). Our estimates of the additional light source were probably not precise enough. Then this simplification of the lighting model induced distortions of the estimated albedo and a lack of robustness of the pose estimation for frames that didn't belong to the training set of head images. To improve our system, we intend to use the model of the interior of the control tower, and place the different lighting sources where they actually are (screens and windows). We would then only need to estimate the brightness of each source.

Figure 7 shows the relative fixation times for various objects in the scene that were computed given the first simplified lighting model (left) and the more elaborate one (right).

6. DISCUSSION

We have demonstrated the recovery of crude gaze information using head pose estimated 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 automated behavioral analyses. Lighting compensation has the potential to improve the quality of the results both in model

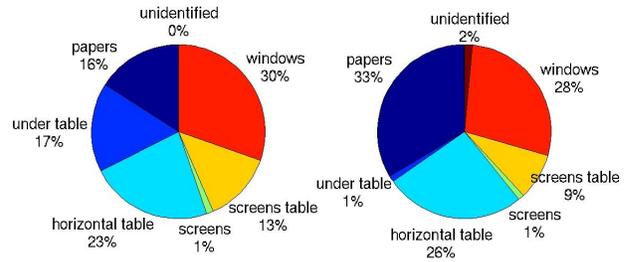


Fig. 7. Distribution of object fixations for model A (left) and model B (right)

construction and pose estimation, although our current results are inconclusive regarding the magnitude of the improvements. Future work will focus on improving the lighting model and automatically optimizing the head shape.

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] 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.
- [3] 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.
- [4] 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.
- [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.