MIT Media Laboratory, Perceptual Computing Technical Report #180
Appears in IEEE PAMI 15 (6), June 1993

Visually Controlled Graphics

A. Azarbayejani, T. Starner, B. Horowitz, A. Pentland

Abstract— **This correspondence discusses interactive graphics systems driven by visual input. The paper describes the underlying computer vision techniques and presents a theoretical formulation which addresses issues of accuracy, computational efficiency, and compensation for display latency. Experimental results quantitatively compare the accuracy of the visual technique with traditional sensing. An extension to the basic technique to include structure recovery is discussed.**

Keywords— **Egomotion, head tracking, Kalman filter, structure from motion, teleconferencing, virtual holography.**

I. Introduction

Most interactive computer applications require harnessing the user with wires. This detracts both from the user's enjoyment and from the practicality of the system for day-to-day use. In this paper, we describe how a passive visual system can directly provide "real-time" estimates of position and orientation, similar to the measurements provided by the Polhemus sensor, but without the intrusion of wires.

Our system requires a single CCD camera input and uses an extended Kalman filter formulation to recover the six rigid-body motion parameters of an object from a small set of tracked visual feature points. The formulation is efficient, is competitive in accuracy with the Polhemus sensor, and can provide predictive estimates to reduce display lags. The system can track any rigid object, but our discussion focuses on the example of tracking a person's head because of its relevance to interactive graphics applications.

We have demonstrated our system for visual head tracking in two graphics applications: virtual holography and teleconferencing. In virtual holography, the user's head position controls a stereoscopic display so that the user perceives virtual solid objects before him. By moving "around" the displayed objects, the user can see the objects from various viewing positions. In teleconferencing, the user's head position controls the display of the 3-D model of the head to other teleconference participants.

The head-tracking system has three parts: a 2-D image feature finder, described in Section II; an extended Kalman filter that converts the 2-D feature positions into optimal estimates of position and orientation, presented in Section III; and a stereoscopic display that is controlled by the Kalman filter's estimates of head position and orientation, discussed in Section IV.

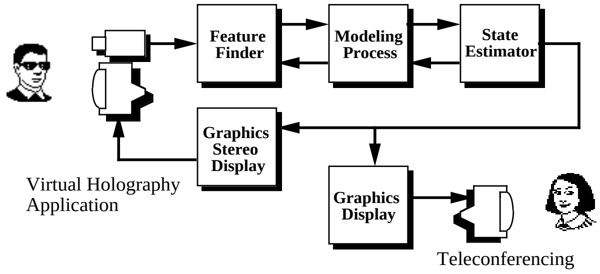

Fig. 1. System Overview. Visual head-tracking can be used to control the user's display (e.g. virtual holography) or to control a remote display for another user (e.g. teleconferencing).

II. Images to features

The 2-D processing includes capture of video images, selection of new features, and tracking of features.

Our system consists of a Sun 4/330 and a Cognex 4400image processing board, which allows video images from a CCD camera to be digitized and buffered in RAM at 30 frames per second.

Selection of distinct features for tracking is based on image characteristics only; we use points where the image intensity surface $I(x, y)$ has a large Hessian, i.e.

$$\left[\left(\frac{d^2I(x,y)}{dx^2}\right)\left(\frac{d^2I(x,y)}{dy^2}\right)-\left(\frac{d^2I(x,y)}{dxdy}\right)^2\right] > \epsilon$$

for some threshold ϵ. Such points are peaks, saddle points, and pits in the image intensity surface, and correspond to features which can be localized in 2-D without "aperture" problems [10]. On heads, these features often (but not always) correspond to the corners of eyes, pupils, nostrils, etc.

For each frame, a set of features satisfying the Hessian criterion are selected for tracking. Since there are 6 motion parameters at each frame and since each tracked feature provides two measurements at each frame (its 2-D coordinates) only 3 points are theoretically required to recover motion. However, many more (at least 10-20) are typically used to overdetermine the solution and reject noise.

Tracking of features is performed using normalized correlation. Correlation templates are extracted for each feature from the original image and from subsequent images in which a substantially novel viewpoint occurs. Viewpoint changes are detected explicitly by using the 3-D motion estimates from the Kalman filter and implicitly by monitor-

Manuscript received January 20, 1993; revised February 25, 1993. Recommended for acceptance by T. Huang.

The authors are with the Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

ing degradation of the correlation indices of the features.

III. Features to Motion Parameters

The extended Kalman filter (EKF) converts the 2-D feature position measurements into 3-D estimates of the position and orientation of the head [3;4;7;1]. A Kalman filter formulation is used because it provides the optimal linear estimate for dynamic systems, because it is recursive and therefore computationally efficient, and because it is based on physical dynamics, which allows for predictive estimation [6].

The EKF requires a physical dynamic model of the motion and a measurement model relating image feature locations to motion parameters. Additionally, some representation of the object (user's head) is required. These are discussed below.

A. Dynamic Model

The dynamic model is a discrete-time Newtonian physical model of rigid body motion. The model has the form

$$\mathbf{x}(t+\Delta t) = \mathbf{\Phi}(\Delta t)\mathbf{x}(t) + \boldsymbol{\xi}(t)$$

where t is time, $\mathbf{x}$ is the *state vector*, $\mathbf{\Phi}$ is the *state transition matrix*, and $\boldsymbol{\xi}$ is an error term, modeled as Gaussian white noise. The 18D state vector and noise vector contain six variables for the translation and rotation of the head, six for velocities, and six for accelerations, i.e.

$$\mathbf{x}(t) \triangleq \begin{pmatrix} \mathbf{p}(t) \\ \boldsymbol{\rho}(t) \\ \mathbf{v}(t) \\ \boldsymbol{\vartheta}(t) \\ \mathbf{a}(t) \\ \boldsymbol{\alpha}(t) \end{pmatrix} \quad \text{and} \quad \boldsymbol{\xi}(t) = \begin{pmatrix} \boldsymbol{\xi}_p(t) \\ \boldsymbol{\xi}_\rho(t) \\ \boldsymbol{\xi}_v(t) \\ \boldsymbol{\xi}_\vartheta(t) \\ \boldsymbol{\xi}_a(t) \\ \boldsymbol{\xi}_\alpha(t) \end{pmatrix}$$

where $\mathbf{p}$ is 3-D translation, $\mathbf{v}$ and $\mathbf{a}$ are velocity and acceleration (translational); $\boldsymbol{\rho}$ is three small Euler angles and $\boldsymbol{\vartheta}$ and $\boldsymbol{\alpha}$ are rotational velocity and acceleration. Large global rotations are handled externally using unit quaternions (see [4;1;2]).

From Newtonian physics,

$$\boldsymbol{\Phi}(\Delta t) = \begin{pmatrix} I & 0 & I\Delta t & 0 & I(\Delta t)^2 & 0 \\ 0 & I & 0 & I\Delta t & 0 & I(\Delta t)^2 \\ 0 & 0 & I & 0 & I\Delta t & 0 \\ 0 & 0 & 0 & I & 0 & I\Delta t \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{pmatrix}.$$

B. Measurement Model

The measurement model relates the state vector $\mathbf{x}$ to the 2-D image location, $\mathbf{y}_i$, of each image feature point, $\mathbf{p}_i$. The transformation and projection equations

$$\mathbf{p}_{i:c}(t) = \mathbf{p}(t) + \mathbf{R}(t)\mathbf{p}_{i:h}$$

$$\mathbf{y}_i(t) = \frac{f}{f + p_{i:c3}(t)} \begin{pmatrix} p_{i:c1}(t) \\ p_{i:c2}(t) \end{pmatrix},$$

are combined to produce the nonlinear measurement relation (for a single feature point):

$$\mathbf{y}_i(t) = \mathbf{h}_i(\mathbf{x}(t)),$$

where $\mathbf{p}_{i:h}$ is the (constant) 3-D location of the feature point in the reference frame of the object (head), f is the focal distance (finite for central projection, infinite for orthographic), and the rotation matrix $\mathbf{R}$ comes from the global quaternion [4;1;2]. The set of constants $\{\mathbf{p}_{i:h}\}$, $i = 1 \ldots N$, where N is the number of features, describe the structure of the tracked object (user's head) and can be modeled *a priori* or estimated on line, as discussed below.

The *measurement vector* contains all the measurements in order, $\mathbf{y}(t) = (\mathbf{y}_1(t), \mathbf{y}_2(t), \ldots \mathbf{y}_N(t))^T$ and thus has $2N$ elements. The full measurement model can be written as

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \boldsymbol{\eta}(t)$$

where the structure parameters $\{\mathbf{p}_{i:h}\}$ are expressed as part of $\mathbf{h}()$ and $\boldsymbol{\eta}$ is an error reflecting uncertainty in the measurement relation and the structural model, modeled as Gaussian white noise.

C. 3-D Object Modeling

For the head-tracking example, the structure parameters $\mathbf{p}_{i:h}$ are initialized for each point when it is first selected. The initialization is based on a simple ellipsoidal model of the head, where the center of the ellipsoid is the origin of the *head reference frame.*

Each feature point can be represented in the camera reference frame as

$$\mathbf{p}_{i:c} = \begin{pmatrix} y_{i:1} \\ y_{i:2} \\ 0 \end{pmatrix} + \alpha_i \begin{pmatrix} y_{i:1}/f \\ y_{i:2}/f \\ 1 \end{pmatrix}$$

where the scalar α_i is the unknown depth and f is the focal length ($f \in (0, \infty]$ and $(f = \infty) \Longleftrightarrow$ (orthographic projection)). Each point can be transformed into the head frame using the motion parameters $\mathbf{p}(t)$ and $\mathbf{R}(t)$:

$$\mathbf{p}_{i:h} = \mathbf{R}^{-1}(t)(\mathbf{p}_{i:c} - \mathbf{p}(t)).$$

Thus, each feature point can be represented in the head frame with one unknown parameter, α_i, which describes the location of the feature along a ray prescribed by the motion parameters. This parameter α_i is computed by intersecting the ray with the ellipsoid. In this way, structure parameters $\{\mathbf{p}_{i:h}\}$ are computed for the measurement model whenever new features are selected for tracking.

It is worthy to note that although simple structural models give sufficient accuracy for many applications, including head-tracking, we have shown in related work [2] that the basic motion-tracking technique can be extended to directly recover the structural parameters simultaneously with the motion parameters, eliminating the requirement for *a priori* models and improving accuracy. In this extended formulation, one structure parameter is estimated for each feature point along with the six motion parameters. When N points are tracked, there are $6 + N$ motion plus structure parameters and $2N$ measurements at each frame. Thus, both structure *and* motion are overdetermined at each frame by tracking 6 or more points through a sequence. This result, that simultaneous structure and motion recovery is in fact an overdetermined problem, is described in detail in [2]. It is sufficient here to note the implications, which are that the accuracy of head-tracking can be significantly increased by refining the structural parameters on-line, and that tracking of more complicated unmodeled objects is possible as well.

D. Extended Kalman Filter

An *extended Kalman filter (EKF)* is used to recursively estimate the state vector at each frame using the feature measurement vector $\mathbf{y}(t)$ and a state prediction $\hat{\mathbf{x}}(t|t-1)$ and the state prediction error covariance $\mathbf{P}(t|t-1)$. A detailed explanation of the EKF and the notation can be found in [5;1;2] and others. The relevant equations for state estimation at frame t are:

$$\mathbf{H}(t) = \left(\frac{\partial \mathbf{h}(t)}{\partial \mathbf{x}}\right)_{\mathbf{x}=\hat{\mathbf{x}}(t|t-1)}$$

$$\mathbf{R}(t) = E[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$$

$$\mathbf{K}(t) = \mathbf{P}(t|t-1)\mathbf{H}^T(t)\left(\mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}^T(t) + \mathbf{R}(t)\right)^{-1}$$

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-1) + \mathbf{K}(t)\left(\mathbf{y}(t) - \mathbf{h}(\hat{\mathbf{x}}(t|t-1))\right)$$

$$\mathbf{P}(t|t) = \left(\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t)\right)\mathbf{P}(t|t-1).$$

State predictions are obtained using

$$\hat{\mathbf{x}}(t+\Delta t|t) = \boldsymbol{\Phi}(\Delta t)\hat{\mathbf{x}}(t|t)$$

$$\mathbf{Q}(t) = E[\boldsymbol{\xi}(t)\boldsymbol{\xi}^T(t)]$$

$$\mathbf{P}(t+1|t) = \boldsymbol{\Phi}(t)\mathbf{P}(t|t)\boldsymbol{\Phi}^T(t)$$

Note that in addition to the necessary prediction required for each step of the EKF, this mechanism can be used to predict over larger time intervals to compensate for graphics display lag, discussed below.

IV. Graphics Display

The head tracking system uses the ThingWorld modeling system [11] to control a Tektronics stereoscopic display. As with most graphics systems, there can be a significant delay between the time ThingWorld receives object information and the time that it can render the new view. Such lags in updating the user's view can cause anything from a feeling of system sluggishness to actual motion sickness.

This problem can be alleviated by inserting a process between the head tracker and ThingWorld which predicts the position of the head one frame in advance, giving the ThingWorld renderer enough time to maintain synchronization with head position [8]. The optimal linear technique for such prediction is the Kalman filter. Since the head tracker is based on the Kalman filter, we can simply use its predictions of head position and velocity to maintain display synchronization.

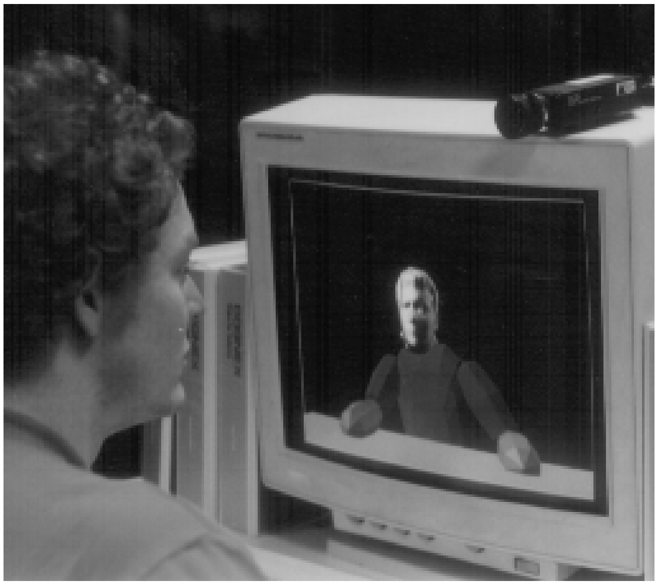

Fig. 2. The head tracking system in virtual holography mode, note camera at top of monitor.

V. Experimental Results

The system described above is implemented on a Sun 4/330 with a Cognex 4400 subsystem, which digitizes incoming video and tracks feature templates. The frame rate has reached 10 frames per second (fps) with our implementation. The state estimates of head position and orientation are propagated ahead, using the prediction mechanism of the EKF, and then transmitted to a second workstation running the ThingWorld modeling environment. The state predictions control display of 3-D models for virtual holography or teleconferencing. Display latency is reduced when predictions are used.

Figure 2 illustrates the system operating in the virtual holography mode. In this mode the head position is tracked and used to control a stereographic display in front of the user, thus simulating the experience of viewing a real object. The camera on top of the computer monitor is the sole input to the system.

Figure 3 compares the accuracy of the Polhemus sensor and our Kalman filter for tracking head position. A person's head was tracked using the Polhemus sensor and the vision system simultaneously. The state estimates were aligned to each other using techniques similar to those used in *absolute orientation* [9]. Scale and bias were removed by performing a linear regression of the Polhemus data to the vision estimate.

As illustrated in the figure, the vision estimates and Polhemus estimates are similar. The RMS difference between vision and Polhemus estimates is 1.67 cm and 2.4 degrees. These statistics are comparable to the observed Polhemus accuracy, indicating that the vision estimate is as least as accurate as the Polhemus. Other tests of the vision estimator's accuracy show that it is substantially better than these figures indicate [2].

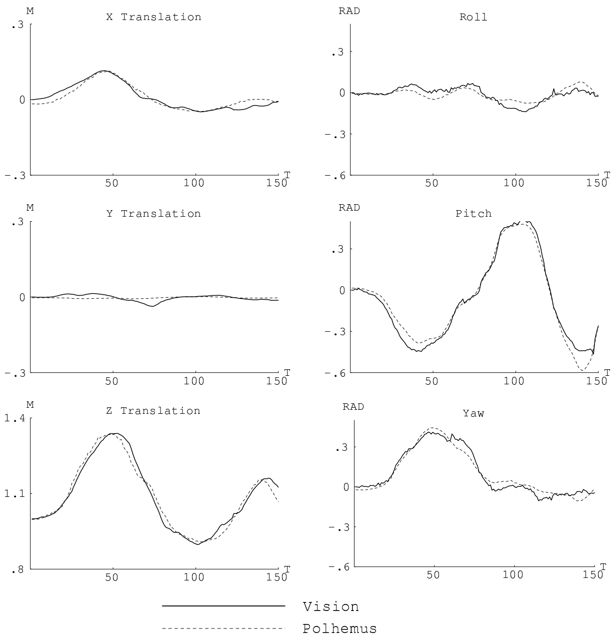

Fig. 3. Polhemus and Vision estimates of head position.

VI. Summary

We have described a passive technique for tracking 3-D position and orientation of rigid objects. Using this technique, we have developed a head-tracking system which has been demonstrated in two interactive graphics applications: virtual holography and teleconferencing. The implementation (Sun 4/330 plus Cognex 4400) demonstrates that a 10 fps rate can be reached in a workstation environment; this can easily be increased to 30 fps or faster with special-purpose hardware.

We have presented experimental results of the basic technique that show the accuracy of the visual system is at least as good as the industry standard Polhemus magnetic sensor system. We also described an extension to the basic technique that recovers structure parameters in addition to motion. This extension is useful when prior models are not available and promises greater accuracy due to better structural information.

Finally, we have described how our formulation of the tracking system facilitates prediction of motion parameters. This prediction can compensate for the display latency that results from the computation of head position estimates and computation of rendered graphics.

Further research continues on 3-D pointwise structure recovery [2], recovery of more highly detailed 3-D models, and feedback of 3-D estimates to enhance 2-D tracking robustness.

VII. Acknowledgements

The authors would like to thank Martin Friedmann, Stan Sclaroff, Irfan Essa, and Trevor Darrell for various forms of assistance in performing the experiments and producing this manuscript.

References

[1] Ali J. Azarbayejani. Model-based vision navigation for a free-flying robot. Master's thesis, Department of Aeronautics and Astronautics and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1991.

[2] Ali J. Azarbayejani, Bradley Horowitz, and Alex Pentland. Recursive estimation of structure and motion using relative orientation constraints. In *1993 IEEE Conference on Computer Vision and Pattern Recognition*, New York, 1993. IEEE Computer Society. (to appear).

[3] Ted J. Broida and Rama Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99, January 1986.

[4] Ted J. Broida and Rama Chellappa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(6):497–513, June 1991.

[5] Robert Grover Brown. *Introduction to Random Signal Analysis and Kalman Filtering*. John Wiley & Sons, New York, 1983.

[6] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[7] Olivier Faugeras and Nicholas Ayache. Maintaining representations of the environment of a mobile robot. *International Journal of Robotics Research*, 1989.

[8] Martin Friedmann, Thad Starner, and Alex Pentland. Device synchronization using an optimal linear filter. In R. A. Earnshaw, M. A. Gigante, and H. Jones, editors, *Virtual Reality Systems*, chapter 9. Academic Press, London, 1993.

[9] Berthold K. P. Horn. Closed-form solution of absolute oreintation using unit quaternions. *Journal of the Optical Society of America*, 4:629, 1987.

[10] B. D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th Intern. Joint Conf. Artif. Intell.*, 1981. (Vancouver.).

[11] Alex Pentland and J. R. Williams. Good vibrations : Modal dynamics for graphics and animation. *ACM SIGGRAPH Conference Proceedings*, 23(4):215–222, 1989.