IEEE Instrumentation and Measurement
Technology Conference
Budapest, Hungary, May 21-23, 2001.

3D Head Pose Recovery for Interactive Virtual Reality Avatars

M.D. Cordea [1], E.M. Petriu [1], N.D. Georganas [1], D.C. Petriu [2], and T.E. Whalen [3]
[1] School of Information Technology and Engineering, University of Ottawa, Canada
161 Louis Pasteur, Ottawa, Ont., K1N 6N5, Canada
mcordea@uottawa.ca, petriu@site.uottawa.ca, georgana@site.uottawa.ca
[2] Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, petriu@sce.carleton.ca
[3] CRC - Communications Research Center, Ottawa, Canada, thom.whalen@crc.ca

Abstract *– This paper discusses a 3D tracking method allowing real-time recovery of the 3D position and orientation of a moving head. The described method uses a 3D wireframe model of the head, a 2D feature-based matching algorithm, and an Extended Kalman Filter (EKF) estimator. The resulting motion tracking system works in a realistic environment without makeup on the face, with uncalibrated camera, and unknown lighting conditions and background.*

Keywords *– real-time tracking, 3D head, avatar, virtual reality*

I. INTRODUCTION

Model-based video coding (MBVC) has recently emerged as a very low bit rate video compression method suitable for Collaborative Virtual Environment (CVE) applications [1]. The MBVC increases coding efficiency by using knowledge about the scene content and describing the real world geometry by 3D model objects. The principle of this compression is to generate a parametric model of the image seen at the emission end and to transmit only the characteristic parameters describing how the model changes in time. These differential parameters are then used to animate the model of the image recovered at the reception end.

The first step in a full automatic MBVC system is the *face detection* allowing the identification and location of the face in first image frames. The next step is *motion estimation* encompassing global 3D-motion recovery, local motion estimation, expression and emotion analysis, etc. The problem is technologically difficult, as 3D motion parameters have to be extracted from a sequence of 2D images of the performer's head-and-shoulders.

This paper discusses a *3D tracking method* for the real-time measurement of six head motion parameters, namely 3D position and orientation, and the focal length of the camera. This method uses a 3D wireframe head model, a 2D feature-based matching algorithm, and an *Extended Kalman Filter* (EKF) estimator. Our global motion tracking system is meant to work in a realistic CVE without makeup on speaker's face, with uncalibrated camera, unknown lighting conditions and background.

II. TRACKING HEAD MOTION

The general problem of recovering 3D position parameters from 2D images could be solved using different 2D views of the 3D objects. If these images are taken at the same time the problem is solved by *stereovision* [2], [3], or *trifocal tensor* [4]. Another approach using monocular 2D images of moving objects is known as *Structure-From-Motion* (SFM) [5].

Given 2D-object images the SFM problem aims to recover:
(i) the 3D object coordinates
(ii) the relative 3D camera- object motion
(iii) camera geometry (camera calibration)

The SFM framework (Fig.1) consists of two main modules:
(i) *Tracking module*, delivering the 2D point measurements $p_i(u_i, v_i)$ of the tracked features, where $i=1,\ldots,m$, and m is the number of measurement points.
(ii) *Estimator module* (for the estimation of 3D geometry and motion), delivering a state vector

$$s = (t_x, t_y, t_z, \alpha, \beta, \lambda, f, X_i, Y_i, Z_i) \quad (1)$$

where $(t_x, t_y, t_z, \alpha, \beta, \lambda)$ are the six 3D camera/object relative motion, namely translation and rotation, f is the camera focal length, and $P_i(X_i, Y_i, Z_i)$ is the object geometry, where $i=1,\ldots,m$, and m is the number of tracked features.

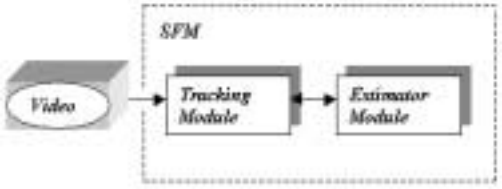

Fig. 1: The Structure-From-Motion (SFM) framework

To detect and locate a human face, the system will process the image, identifying relevant features, and then use these features to recognize and determine the location of the face. Tracking finds and locates the relevant facial features in a sequence of images. Tracking should allow estimating the motion while locating the face. There are three main tracking techniques [6], [7]:

(i) Feature-based methods, which extract image features and track their movement from frame to frame. Image features are low level image descriptors, such as "regions", "edges", and "point features". Reliable tracking of *regions* is often difficult, since minor changes between frames can lead to very different segmentation in consecutive frames. Arbitrarily curving *edges* are difficult to describe and track. Trackers based on *point features* such as nostrils, corners of eyes, mouth endpoints, tips of eyebrows are increasingly used in computer vision applications [8], [9]. However, in a scene where objects move erratically, the noisy image data and spatial and temporal sub-sampling can make motion and acceleration estimation difficult.

(ii) Optical-flow methods, which use spatial and temporal partial derivatives to estimate the image flow at each location in the image. Algorithms for recovering optical flow [6] are based on a set of assumptions about the world that, by necessity, are simplifications and hence may be violated in practice resulting in gross measurement errors. Moreover, the extraction of the optical flow from an image sequence is a highly computational task.

(iii) Correlation-based methods are popular for tracking objects [10], [11]. They use the sum of the absolute differences between template and search area pixel intensities as a difference measure. On the negative side, the correlation tracking methods are sensitive to changes in overall illumination changes between frames of the sequence.

We employ a *feature-based* tracking technique to obtain the 2D observations , which SFM can use to infer the 3D information.

The SFM problem can be formulated as a parameter estimation problem: "Given a number of noisy measurements of 2D-tracker positions, we have to optimally recover the SFM components of equation (1)".

We have adapted the SFM approach of Azarbayejani and Pentland [5] to recursively recover the 3D motion and perspective camera geometry from feature correspondences over a sequence of 2D images. To speed up the calculations we are using a motion model that simplifies the Jacobian. EKF is used to solve the SFM problem resulting in an accurate, stable and real time solution. The EKF takes in consideration the non-linear aspect of mapping. We use a perspective camera model to reflect the mapping between the 3D world and its projection. In the next section we present an EFK based technique, used to recover 3D motion parameters and camera focal length.

III. EXTENDED KALMAN FILTER FOR 3D TRACKING

The continuous imaging process is sampled at discrete time intervals by grabbing images at a constant time interval. These images are then sequentially analyzed using an EKF to determine the motion trajectory of the face within a determined error range.

The EKF converts the 2-D feature position measurements, using a perspective camera model into 3-D estimates of the position and orientation of the head [5], [12], [13]. The EKF recursive approach captures both the cause-effect and the dynamic nature of the tracking, offering also a probabilistic framework for uncertainty representation.

The EKF is applied to nonlinear systems and consists of two stages: time updates (or prediction) and measurement updates (or correction). At each iteration, the filter provides an optimal estimate of the current state using the current input measurement, and produces an estimate of the future state using the underlying state model. The values, which we want to smooth and predict independently, are the tracker state parameters.

The EKF state and measurement equations and can be expressed as:

$$s(k+1) = As(k) + \xi(k) \quad (2)$$

$$m(k) = Hs(k) + \eta(k) \quad (3)$$

where *s* is the state vector, *m* is the measurement vector, *A* is the state transition matrix, *H* is the Jacobian that relates state to measurement, and $\xi(k)$ and $\eta(k)$ are error terms modeled as Gaussian white noise.

The observations are the 2D feature coordinates *(u,v),* which are concatenated into a measurement vector $m(k)$ at each time step. The observation vector is the back-projection of the *s* state vector containing the relative 3D camera-scene motion, and the camera internal geometry, namely the focal length. In our case the state vector is $s(translation, rotation, velocity, focal\_length)$ that contains the relative 3D camera-object translation, rotation and their velocities, and camera focal length.

The EKF requires a physical dynamic model of the motion and a measurement model relating image feature locations to

motion parameters. Additionally, a representation of the object (user's head) is required.

3.1. The Motion Model

The dynamic model is a discrete-time Newtonian physical model of a rigid body motion, moving with constant velocity. The state vector:

$$s(t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, f, \dot{t}_x, \dot{t}_y, \dot{t}_z, \dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z)$$

consists of 13 elements grouped as follows: the relative camera-object translation (t_x, t_y, t_z), the small inter-frame rotation $(\omega_x, \omega_y, \omega_z)$, the camera focal length *f*, the translational velocity $(\dot{t}_x, \dot{t}_y, \dot{t}_z)$, and the rotational velocity $(\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z)$.

The state equation (1) could be written as:

$$\begin{pmatrix} t_i \\ \omega_i \\ f \\ \dot{t}_i \\ \dot{\omega}_i \end{pmatrix}_{k+1} = \begin{pmatrix} I & 0 & 0 & I\Delta\tau & 0 \\ 0 & I & 0 & 0 & I\Delta\tau \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} t_i \\ \omega_i \\ f \\ \dot{t}_i \\ \dot{\omega}_i \end{pmatrix}_k + \xi(k) \tag{4}$$

where $i = x, y, z$ is the index of the coordinate axes of the camera reference frame, *I* is the identity matrix and $\Delta\tau$ is the inter-frame time.

3.2. The Measurement Model

The measurement model relates the state vector *s* to the 2D-image location (u_k, v_k) of each image feature point, p_k. The point $p_k(X_k, Y_k, Z_k)$ of the object reference frame becomes the point $p_{ck}(X_{ck}, Y_{ck}, Z_{ck})$ of the camera reference frame, where:

$$\begin{pmatrix} X_{ck} \\ Y_{ck} \\ Z_{ck} \end{pmatrix} = T(t_x, t_y, t_z) + R(\alpha, \beta, \gamma) \begin{pmatrix} X_k \\ Y_k \\ Z_k \end{pmatrix}, \; k=1,\ldots,N, \tag{5}$$

where *T* and *R* represent the object (or camera) translation and rotation matrices, and *N* is the number of points.

The observed perspective projection is given by:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \frac{f}{Z_{ck}} \begin{pmatrix} X_{ck} \\ Y_{ck} \end{pmatrix}, \; k=1,\ldots,N, \tag{6}$$

where *f* is the camera focal length.

At each filter cycle we have to calculate the partial derivatives of *u* and *v* with respect to each of the unknown parameters. Lowe [14] proposed a reparameterization of the projection equations, to simplify the calculation of *H* Jacobian, by expressing the translations in the camera coordinate system rather than model coordinates. In this case the measurement equation will take the following form:

$$\begin{pmatrix} X_{ck} \\ Y_{ck} \\ Z_{ck} \end{pmatrix} = R(\alpha, \beta, \gamma) \begin{pmatrix} X_k \\ Y_k \\ Z_k \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} \frac{f}{Z_{ck} + t_z} X_{ck} + t_x \\ \frac{f}{Z_{ck} + t_z} Y_{ck} + t_y \end{pmatrix} \tag{8}$$

When *N* points are tracked, there are *2N* measurements (coordinates of point projections) at each frame and 7 parameters to be recovered (six motion parameters plus camera focal length). Both motion and focal length are over-determined at each frame when $2N > 7$, which happens when $N \geq 4$, i.e. when tracking 4 or more points. When camera parameters are known beforehand, we need $N \geq 3$ points to recover the 3D motion.

We employ a three-parameter incremental rotation $(\omega_x, \omega_y, \omega_z)$, similar to that used in [5] to estimate inter-frame rotation. The incremental rotation computed at each frame step is combined into a global quaternion vector (q_0, q_1, q_2, q_3) used in the *EKF* linearization process and rotation of the 3D-model [15].

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1. EKF Initialization

The 3D model provides the initial structure parameters

(X_i, Y_i, Z_i) of the Kalman filter. Each 2D-feature point (u_i, v_i) corresponds to a structure point $p_i(X_i, Y_i, Z_i)$. As shown in Fig. 2, these (u_i, v_i) points are obtained by intersecting the 2D image plane with a ray rooted in the camera's center of projection COP and aiming to the 3D structure point on the head model.

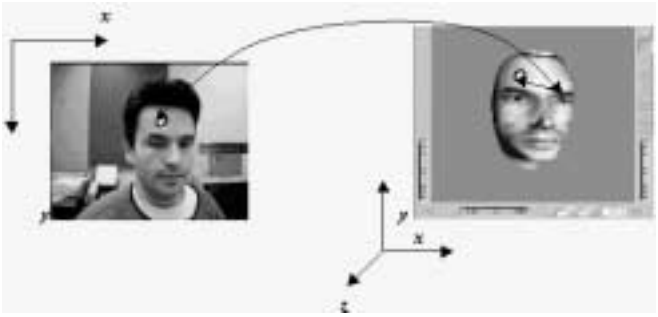

Fig. 2: Identical point selection process on Marius' image and the corresponding 3D model projection.

The typical point identification problem of the 3D pose recovery from 2D images is solved in our case by identifying corresponding points in both the 2D live image of the subject and the 3D model of the subject's head. In order to aid the point identification process, we are using an augmented reality technique by projecting in the 2D live image the 3D mesh used to model the head.

At this development stage it is still up to the user to arrange the scale matching between the live face image and the projected mesh. The steps of the EKF initialization algorithm for multiple "point identification" procedure using this augmented reality technique are as follows:

Step 1.
The user positions his/her face at the center of the screen, and adjust the matching of the live image and the projected mesh , so that the projected mesh covers the entire facial region.

Step 2.
Left mouse click on every rigid feature point of interest on the live image. An automatic program function takes care to properly align the selected live feature point to a vertex of the projected mesh.

Step 3.
Right mouse click anywhere on the active Windows "live" image triggers the tracking process (by "booting the EKF module).

4.2. EKF Update

The EKF update stage is illustrated in Fig. 3. At each iteration, the EKF computes an estimate of the rigid 3D motion that must probably correspond to the motion of the 2D live image. We employ the Kanade-Lucas-Tomasi (KLT) [16] 2D-gradient feature tracking method, which robustly performs the tracking reinforced by the EFK estimation output. An estimate of motion and camera focal length is found at each step. After the 3D-motion and focal length are recovered, a perspective transformation will project feature points back onto the image to determine an estimated position of the 2D feature trackers. At the next frame in the sequence a 2D tracking is performed starting at this 2D estimated position. The current matching coordinates of tracked features are fed back into the Kalman filter as the observation vector, and the loop continues. The feedback from EFK is used to update the 3D-model pose parameters, i.e. provides the 3D head tracking information.

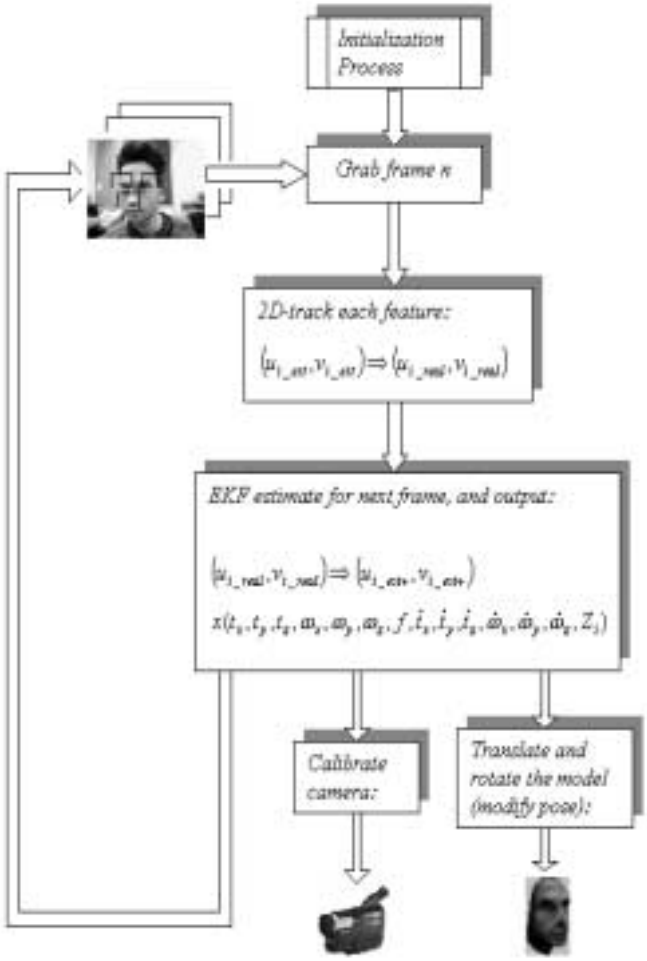

Fig. 3: Continuous 3D pose recovery using EKF

The recovered 3D position and orientation are propagated to the *Head Modeling* block of the CVE system, which renders a new posture of the 3D-model as illustrated in Fig. 4.

Fig. 4: Tracking the head motion.

V. CALIBRATION

In order to validate the accuracy of our 3D-head tracking system, we developed a rapid calibration technique. A previously recorded sequence of 2D images representing 3D head model poses, is played as "live" image, and tracked with our EKF framework.

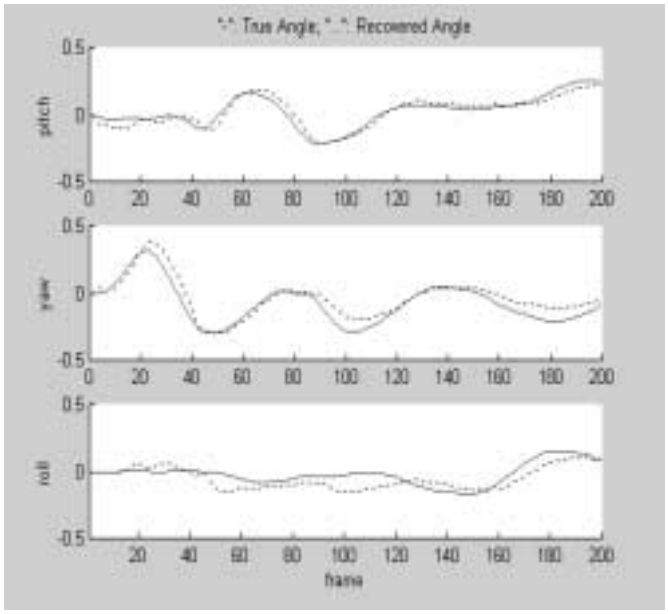

Fig.5: True and recovered rotation angles: EKF-4 points

The estimated motion values $(t_x^e, t_y^e, t_z^e, \theta_x^e, \theta_y^e, \theta_z^e)$ are compared with the measured motion values $(t_x^m, t_y^m, t_z^m, \theta_x^m, \theta_y^m, \theta_z^m)$ of the synthetic image sequence. In the above representation (t_x, t_y, t_z) is the 3D position, and $(\theta_x, \theta_y, \theta_z)$ is the 3D orientation of the head. The resulted errors show the effect of both human-aided 3D/2D point-identification and 3D tracking.

We minimized the errors by fine-tuning the initialization process of the EKF.

Fig. 5 shows the *recovered* vs. *real* 3D-orientation for a calibrated sequence.

We have found experimentally in one case that the RMS difference between true and recovered rotation angles is 3.035 degrees when tracking 4 points. These statistics are comparable to the Polhemus sensor accuracy [5] indicating that the vision estimate is at least as accurate as the Polhemus sensor.

VI. CONCLUSION

Tracking 3D pose parameters of a moving target (head) from a sequence of 2D-images motion is technologically difficult. The effects of head motion and facial expressions are combined in these images, so it is crucial to successfully separate the rigid from the non-rigid motion of the head ("pose/expression separation"). The head pose has to be accurately computed before attempting to recover the expressions.

The *3D tracking* model-based algorithm discussed in this paper allows automatic recovery of six head-parameters: the 3D position and orientation. Experimental results show that this tracking system works well in a realistic videoconferencing environment, without makeup highlighting the speaker's facial features, unknown lighting conditions, and unknown scene background.

ACKNOWLEDGMENT

This work was funded in part by Communications and Information Technology Ontario (CITO), the STENTOR New Media Fund and the Communications Research Centre (CRC) of Canada.

REFERENCES

[1] K. Aizawa, T.S. Huang, *Model Based Image Coding: Advanced Video Coding Techniques for very Low Bit-Rate Applications*, Proc. IEEE, vol. 3, No. 2, Feb. 1995.

[2] O. Faugeras. *What can be seen in three dimensions from an uncalibrated stereo rig?* In Proceedings of the 2nd European Conference on Computer Vision, pages 563-578, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

[3] R. Hartley, R. Gupta, and T. Chang. *Stereo from uncalibrated cameras.* In Proceedings of the Conference on Computer Vision and Pattern Recognition, pages 761-764, Urbana-Champaign, Illinois, 1992.

[4] R. Hartley. *Lines and points in three views - an integrated approach*. In Proceedings of the ARPA IU Workshop. DARPA, Morgan Kaufmann, 1994.

[5]] A. Azarbayejani and A Pentland. *Recursive estimation of motion, structure, and focal length.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(6), 1995.

[6] B.K.P. Horn, *Robot Vision.* MIT Press, 1986.

[7] L.S. Shapiro, *Affine analysis of image sequences.* PhD Thesis, Sharp Lab. of Europe, Oxford, U.K., 1995.

[8] V.S.S. Hwang. "Tracking feature points in time-varying images using an opportunistic selection approach," *PR*, 22(3), pp. 247-256, 1989.

[9] J. Shi, and C. Tomasi, "Good Features to Track," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, June 1994.

[10] G.D. Hager and P.N. Buelhumeur, "Real-Time Tracking of Image Regions with Changes in Geometry and Illumination," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403-410, 1996.

[11] T. Jebara and A. Pentland, *Parameterized Structure from Motion for 3D Adaptive feedback Tracking of Faces*, Media Laboratory, MIT Cambridge, MA 02139, November 1996.

[6] G.D. Hager and P.N. Buelhumeur, "Real-Time Tracking of Image Regions with Changes in Geometry and Illumination," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403-410, 1996.

[7] T. Jebara and A. Pentland, *Parameterized Structure from Motion for 3D Adaptive feedback Tracking of Faces*, Media Laboratory, MIT Cambridge, MA 02139, November 1996.

[8] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," *Proc.IEEE CVPR*, pages 21–27, 1997.

[9] Stan Birchfield, *Elliptical Head Tracking Using Intensity Gradients and Color Histograms*. Computer Science Department, Stanford University, Stanford, CA 94305

[10] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[11] P. Palavouzis, *Head Tracking for Face Recognition*, Master's thesis, Department of Computer Science, QMW, London, UK, 1994.

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

[13] D.B. Gennery. Visual tracking of known 3-dimensional object. Int. J. of Computer Vision, 7(3), pages 243–270, 1992.

[14] David G. Lowe. *Three-dimensional object recognition from single two-dimensional images.* Artificial Intelligence, 31(3): pages 355-395, March 1987.

[15] Ken Shoemake. *Quaternions.* Department of Computer and Information Science University of Pennsylvania Philadelphia, PA 19104.

[16] Jianbo Shi, and Carlo Tomasi, *Good Features to Track*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR94) Seattle, June 1994.