Real-Time 3-D Head Motion Estimation in Facial Image Coding

Tzong-Jer Yang, Fu-Che Wu, Ming Ouhyoung
Communication & Multimedia Laboratory,
Dept. of Computer Science and Information Engineering,
National Taiwan University, Taipei, 106, Taiwan
{tjyang, joyce, ming}@cmlab.csie.ntu.edu.tw

Abstract

A simple procedure that uses only three feature points to infer 3-D head motion from consecutive video frames is presented. In this procedure, a feature triangle formed using the three feature points is automatically calibrated, and an iterative method that simulates steepest descent method is applied to estimate one's head motion. A prediction algorithm is adopted in occasional cases where the estimated result is not acceptable. This procedure has been applied to live video with an update rate of 7 frames/sec (250 frames/sec without feature extraction) on a Pentium-II 233MHz PC, independent of cameras and users.

1. Introduction

In researches of model-based videoconference or videophone, one key problem is how to obtain one's head motion in 3-D space. The problem is usually resolved in two steps. First, feature points have to be extracted and correlated between two consecutive video frames. Second, a minimization procedure is applied to find an optimal transformation that fulfills the established point correspondences. In [1], Huang and Netravali presented a comprehensive review of related algorithms.

Methods of feature extraction vary widely from the use of optical flow [2][3], artificial markers [4], face color [5], to template-matching [6][7]. Feature extraction is a time-consuming process, and the complexity of 3-D head motion estimation mostly depends on the method of feature extraction. In general, if more features are used, the motion estimation requires more time in finding an optimal solution.

In Section 2, a simple procedure using only three feature points to recover one's 3-D head motion is presented. Results and conclusions are addressed in Section 3.

2. The Proposed 3-D Head Motion Estimation Procedure

In the procedure, only three feature points, the eyes and the nose, are required to form a 3-D feature triangle. Notice that these feature points can also be used to facilitate facial expression extraction if the "clip-and-paste" mechanism is adopted.

Initially, the feature triangle is automatically calibrated using a video camera capturing one's front face. The calibration process, as shown in Figure 1, is simplified to be only a calculation of the triangle's depth value, because human heads can be considered to have similar size, and we can assume the size of the 3-D feature triangle is fixed. As a result, the feature triangle's depth value, Z, can be obtained quickly from the equation $Z = l \times F / L$, where l is a pre-defined edge length of the 3-D feature triangle, L is the measured edge length from a video frame, and F is a known distance from the projection plane to the camera in an internal camera geometry.

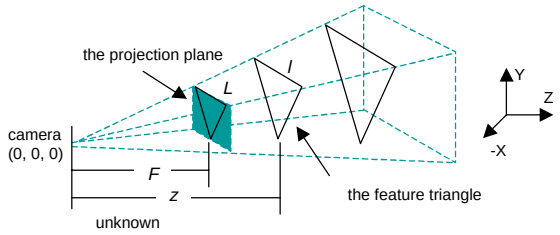

Figure 1. The calibration of the 3-D feature triangle.

After the calibration, a steepest-descent iterative method that transforms the 3-D feature triangle with a small variation is applied. In one iteration, there are totally 12 transformations, which are rotations about ±X-, ±Y-, ±Z-axes, and translations along ±X-, ±Y-, ±Z-axes. The transformation with the smallest error is selected for the next iteration, until the error is within a given threshold. The error is calculated by measuring the distance between the real feature triangle on a video frame and the projected 3-D feature triangle. Four criteria are developed to measure the distance, as listed in Figure 2. The iterative method that is actually performing a local optimization can work because head motion is relatively small in a video sequence, as indicated in [8].

$$T_1 = \sum_{i=0}^{2} \left| P_i^{real} - P_i^{estimated} \right|, \text{ where } P_i^{real} \text{ and } P_i^{estimated} \text{ are the real and the estimated 2D feature points.}$$

$\equiv$ vertex distances between the real and the estimated 2D feature points

$$T_2 = \sum_{i=0}^{2} \left| aligned(P_i^{real}) - aligned(P_i^{estimated}) \right|, \text{ where } aligned(P_i) = P_i - P_c, P_c = \frac{1}{3}\sum_{i=0}^{2} P_i$$

$\equiv$ vertex distances after aligning both centers of gravity to represent triangle similarity

$$T_3 = \left| \frac{\overline{P_0^{rea} P_1^{real}}}{\overline{P_1^{real} P_2^{real}}} - \frac{\overline{P_0^{estimated} P_1^{estimated}}}{\overline{P_1^{estimated} P_2^{estimated}}} \right| + \left| \frac{\overline{P_1^{rea} P_2^{real}}}{\overline{P_2^{real} P_0^{real}}} - \frac{\overline{P_1^{estimated} P_2^{estimated}}}{\overline{P_2^{estimated} P_0^{estimated}}} \right|$$

$\equiv$ ratios of edge lendth to represent triangle shape similarity

$$T_4 = \sum_{i=0}^{2} \left| slope(\overline{P_i^{real} P_j^{real}}) - slope(\overline{P_i^{estimated} P_j^{estimated}}) \right|, \text{ where } j = (i+1) \bmod 3, \text{ and } slope(\overline{P_i P_j}) = \frac{Y_j - Y_i}{X_j - X_i}$$

$\equiv$ edge slopes is used to represent triangle shape similarity

Figure 2. Four criteria used to measure the distance between the real and estimated feature triangles.

However, it still has chances to have unacceptable estimation result. A prediction algorithm, the Grey Predictor [9], is adopted to compensate error estimations. In such a situation, three new feature points are predicted, with the transformation calculated using singular value decomposition [10]. If the prediction result is better, the prediction result is selected, otherwise, the estimation result is used.

3. Results and Conclusions

The proposed 3-D head motion estimation procedure has been applied to live video, as shown in Figure 3. Only three feature points are required, and the procedure is simple. On a Pentium-II 233MHz PC, one's head motion can be tracked at a frame rate of 7 frames/sec with artificial markers attached on the lateral canthus of eyes and the nose. The procedure performs over 250 times/sec if the feature extraction process is not counted.

4. References

[1] Thomas S. Huang, Arun N. Netravali, "*Motion and Structure from Feature Correspondences: A Review*," Prceedings of the IEEE, 82(2), pp. 252-268, Feb. 1994.

[2] Jorn Ostermann, "*Object-Based Analysis-Synthesis Coding Based on the Source Model of Moving Rigid 3D Objects*," Signal Processing: Image Communication, 6, pp. 143-161, 1994.

[3] Haibo Li, Pertti Roivainen, Robert Forchheimer, "*3-D Motion Estimation in Model-Based Facial Image Coding*," IEEE Tran. on Pattern Analysis and Machine Intelligence, 15(6), pp. 545-555, Jun. 1993.

[4] K. Aizawa, H. Harashima, T. Saito, "*Model-Based Analysis Synthesis Image Coding (MBASIC) System for a Person's Face*," Signal Processing: Image Communication, 1, pp. 139-152, 1989.

[5] Qian Chen, Haiyuan Wu, Takeshi Fukumoto, Masahiko Yachida, "*3D Head Pose Estimation without Feature Tracking*," Proc. of the Third International Conference on Automatic Face and Gesture Recognition, Japan, pp. 88-93, 1998.

[6] David Machin, "*Real-Time Facial Motion Analysis for Virtual Teleconferencing*," Proc. of the Second International Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, pp. 340-344, Oct. 1996.

[7] Jochen Heinzmann, Alexander Zelinsky, "*3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm*," Proc. of the Third International Conference on Automatic Face and Gesture Recognition, Japan, pp. 142-147, 1998.

[8] A. N. Netravali, and J. Salz, "*Algorithms for Estimation of Three-Dimensional Motion*," AT&T Technical Journal, Vol. 64, No. 2, pp. 335-346, Feb. 1985.

[9] Jiann-Rong Wu, and Ming Ouhyoung, "*Reducing The Latency in Head-Mounted Display by A Novel Prediction Method Using Grey System Theory*," Computer Graphics Forum, Vol. 13, No. 3, pp. C503-512, 1994.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein, "*Least-Squares Fitting of Two 3-D Point Sets*," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 5, pp. 698-700, Sep. 1987.

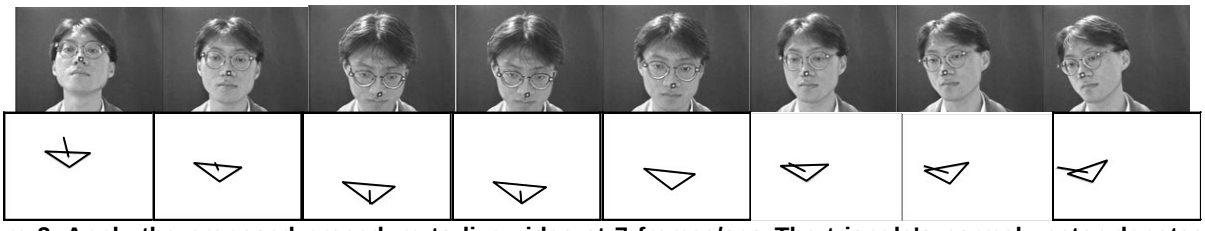

Figure 3. Apply the proposed procedure to live video at 7 frames/sec. The triangle's normal vector denotes the orientation of the nose, and the position of the triangle corresponds to the face's translation.