# VR-Face: An Operator Assisted Real-Time Face Tracking System

Tzong-Jer Yang, Chien-Feng Huang, Cheng-Sheng Hung, Ming Ouhyoung

Communication and Multimedia Laboratory,
Dept. of Computer Science and Information Engineering,
National Taiwan University, Taipei, 106, Taiwan
{tjyang, cardy, cshung, ming}@cmlab.csie.ntu.edu.tw
Http://www.cmlab.csie.ntu.edu.tw/~tjyang/research/face.html.

## Abstract

In this paper, a model-based face analysis and synthesis system is presented. The system, named VR-Face, tracks and estimates one's 3D head motion in real time, and represents the estimated motion with a pre-rendered 3D texture-mapped head model. Initially, a user has to identify two eyes and one nostril on the screen for tracking. In this way, the background can be complex, and even dynamic. When the system fails to follow up one's head motion, it prompts the user with a box indicating the original face position to recover itself from tracking errors. The overall performance, including both analysis and synthesis, is above 25 frames/sec on a PC with a 400MHz Pentinum II-MMX CPU. The system has been demonstrated under different lighting conditions with different low-price PC cameras.

## 1. Introduction

An arising approach to very-low bitrate videophone is to transmit only a few significant facial feature parameters to a remote site, and a new face image is therefore synthesized at the remote site according to the parameters. To retrieve facial features, a series of computer vision and image process techniques has to be applied for face analysis. In general, a face analysis process includes four tasks: face segmentation, feature extraction, feature tracking, and 3D motion estimation. The last task, "3D motion estimation", may be the simplest one, compared to the other three ones, since theoretically 3D motion can be determined up to a scale in one translation direction from two views if enough feature correspondences are given [12][13].

One of the most difficult problems in face analysis is how to provide feature correspondences robustly and stably, which involves the first three tasks mentioned above. Extracted features must be on the face or head, and correspondences between two images have to be correct. Moreover, if the same set of facial features is to be used during the whole process, the same features have to be extracted all the time.

The four tasks are tightly coupled, and are still considered as open problems in computer vision community. Many different approaches have been proposed, where cues of human faces are utilized. Useful cues include two eyes and the mouth [1][2][3][4], face color [5][6][10][11], or artificial markers [7][8]. Though more and more complex algorithms have been developed, there is still a long way toward human being's visual system. For a human, it's quite easy to recognize a face, as well as eyes, nose, mouth, etc. Human visual systems can even work well under a very noisy environment, while computer vision systems can not.

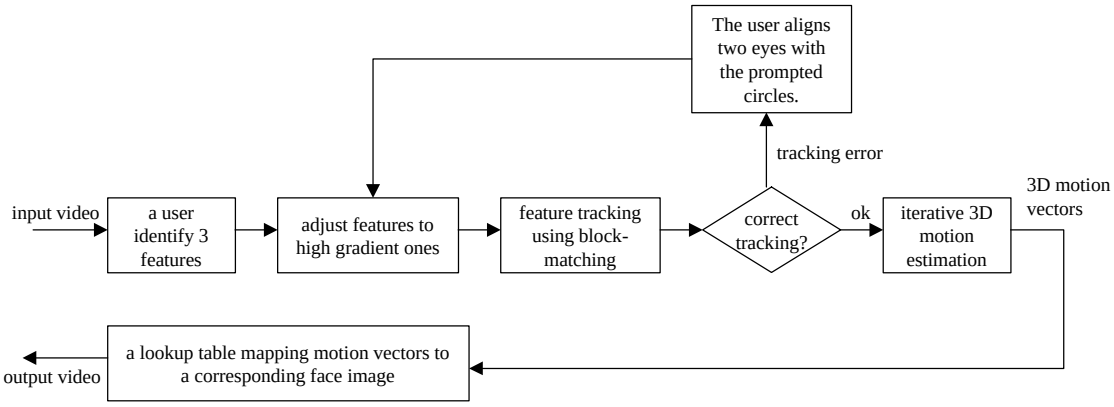


**Figure 1.** Block diagram of the face tracking system.

In this work, we try to build a face tracking system with human's visual system incorporated. The platform used is regular PCs equipped with low-price cameras and capture cards. Signals from these low-price capture devices are quite noisy, and are more difficult to be processed. Initially, two eyes and one nostril are identified by a user manually, and the three feature points are tracked using block-matching. From point correspondences between two consecutive frames, 3D head motion is inferred via an iterative method [8]. If the system fails to track the three features, a box with two circles is displayed to ask the user to align his eyes with the two circles. The system then re-initializes itself automatically and continues the tracking process. Estimated 3D motion parameters are used to display a 3D texture-mapped head model with corresponding orientations. The whole process is depicted in Figure 1.

## 2. Feature Tracking with User Assistance

The mechanism applied in feature extraction and feature tracking depends on the algorithm used for motion estimation. In this work, an iterative 3D head motion estimation method is used [8]. The motion estimation method would be addressed in the next section.

In the iterative motion estimation, three feature points that are two eyes and one nostril are required. The work of feature extraction is accomplished by asking a user to point out the three features manually. The points, or actually pixels, selected by a user are snapped to nearby pixels with the highest gradient values automatically, because these points may fall on non-texture regions that are less useful for low level image processing, e,g, gradient calculation. Once a user specifies the three features in order, the system begins to track them with block-matching.

To speed up the block-matching process, only ten pixels within a search window in the next frame are selected as candidates for one feature point. The ten candidates are those pixels with gradient values closest to the feature point. For each candidate, a correlation score is computed, and the one with highest score is selected as the corresponding point. The correlation equation is given as the following [9]:

$$score(m_1, m_2) = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left(\left[I_1(u_1+i, v_1+j) - \overline{I_1(u_1, v_1)}\right]\times\left[I_2(u_2+i, v_2+j) - \overline{I_2(u_2, v_2)}\right]\right)}{(2n+1)(2m+1)\sqrt{s^2(I_1)\times s^2(I_2)}}$$

where $I(u, v)$ is the image intensity at point $(u, v)$,

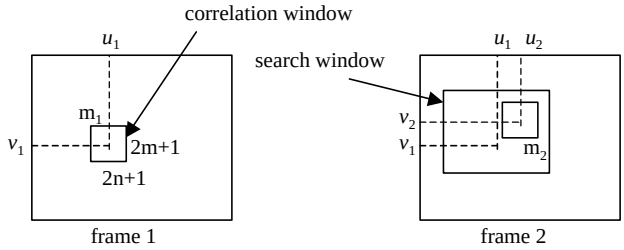


**Figure 2.** Block-matching between two consecutive frames.

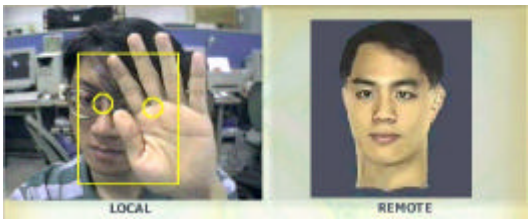


**Figure 3.** A face boundary box and two circles are displayed for error recovery.

$\overline{I_k(u,v)} = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k(u+i, v+j)}{(2n+1)(2m+1)}$ is the averaged intensity at point $(u, v)$ of frame $I_k$, $k$=1, 2,

and $\mathrm{s}(I_k) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k^2(u+i)(v+j)}{(2n+1)(2m+1)} - \overline{I_k(u,v)}^2}$ is the standard deviation of the frame $I_k$ in the neighborhood (2n+1)x(2m+1) of (u, v). The correlation score ranges from –1, not similar at all, to 1, identical. The correlation window is of the size (2n+1)x(2m+1). Figure 2 illustrates the relationship between two frames.

In current implementation, for 320x240 video resolution, the search window is of the size 13x13, and the correlation window is of the size 17x17. A smaller search window is chosen because for eye regions, there are eyebrows and maybe eyeglasses that are all regions with high gradient values. To reduce the possibility of mismatching, we prefer a smaller searching window. In fact, because the distance of eye blinking is smaller than the distance between eyeballs and eyebrows, eye blinking is also allowed with such a small search window. One problem induced from small searching ranges is the limitation for rapid head motion, since corresponding points may fall outside the search window. However, because the system can achieve a processing speed at over 25 frames/sec, motion differences between two consecutive frames would be relatively small.

If the system fails to find a feature point's correspondence, where the highest correlation score is still under a pre-defined threshold, an error recovery process is invoked, and information from the first frame would be used. A bounding box around the original face position, and two circles centered at the original eye positions would be displayed on the screen, as shown in Figure 3. If the distance between two eyes is *eye_len*, the width of the bounding box is 2×*eye_len*, and the height is 2.6×*eye_len*. The user is asked to align his two eyes with the circles to help the system to find corresponding points within known regions. In the error recovery process, correspondences are established between the first frame and current frame.

The tracking algorithm is summarized as the following:

**: Tracking**

**For** video frame $\mathbf{F}_{n+1}$
**For** feature point $\mathbf{P}_{ni}$
Find candidates within the search window whose gradient values close to $\mathbf{P}_{ni}$
**For** candidate $\mathbf{C}_j$
$\mathbf{S}_j$ = ***score***($\mathbf{P}_i$, $\mathbf{C}_j$)
$\mathbf{P}_i$ = **max**($\mathbf{S}_j$)
**If max**($\mathbf{S}_j$) < **threshold**, **goto Error_Recovery**

**: Error_Recovery**

Display a bounding box at the original face position, and two circles at the original eye positions $\mathbf{P}_{n0}$, $\mathbf{P}_{n1}$
$\mathbf{P}_n = (\mathbf{P}_{n0}, \mathbf{P}_{n1}, \mathbf{P}_{n2})$
goto **Tracking**

$\mathbf{F}_{n+1}$ is the $(n+1)$th frame, and $\mathbf{P}_{ni}$ is the $i$th feature point on frame $n$.

## 3. Iterative 3D Motion Estimation

When three feature points are obtained, a steepest-decent based iterative method can be employed to infer the corresponding 3D motion [8]. In the method, two eyes and one nostril form a 3D feature triangle. Initially, the feature triangle is automatically calibrated from one's frontal face. The calibration process, as shown in Figure 4, is simplified to be only a calculation of the triangle's depth value, because sizes of human heads, or the side length $l$ of the 3D feature triangle, are assumed to be the same. As a result, the 3D feature triangle's depth $Z$ can be computed from the equation $Z = l \times F / L$, where $l$ is a pre-defined side length of the 3-D feature triangle, $L$ is the measured side length on a video frame, and $F$ is a pre-defined distance from the projection plane to the camera.

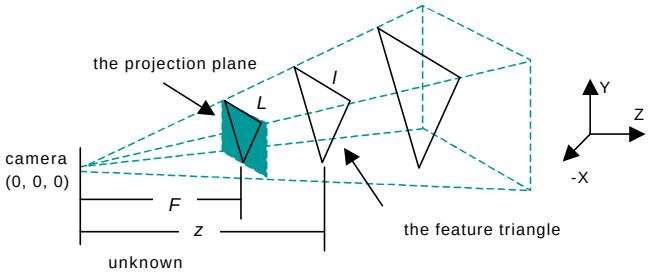


**Figure 4.** 3-D feature triangle calibration.

After the calibration, a steepest-descent iterative method that transforms the 3-D feature triangle with small variations 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 5. The iterative method that is actually performing a local optimization can work because head motion is relatively small in a video sequence.

However, it still has chances to have unacceptable estimation result. A prediction algorithm, the Grey Predictor [14], is adopted to compensate error estimations. In such a situation, three new feature points are predicted,

$$T_1 = \sum_{i=0}^{2} \left| P_i^{real} - P_i^{estimated} \right|, \text{ where } P_i^{real} \text{ and } P_i^{estimated} \text{ are the real and the estimated 2D feature points.}$$
$\equiv$ 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 5.** Four criteria used to measure the distance between the real and estimated feature triangles.

with the transformation calculated using singular value decomposition [15]. If the prediction result is better, the prediction result is selected; otherwise, the estimation result is used.

## 4. Generating Face Images with Corresponding Orientations

At this moment, a corresponding face image is generated by searching an image among a 2D array of pre-stored face images with the orientation closest to the estimated 3D motion. Some images from the 2D array are listed in Figure 6. The 3D head model used to render these face images is generated by taking one's face from three different angles: the left face, the frontal face, and the right face [16]. There are in total 121 (11x11) images in the 2D array, and the rotation angle between two consecutive images is $9^{o}$. We also take real photos from people, as shown in Figure 7(d).

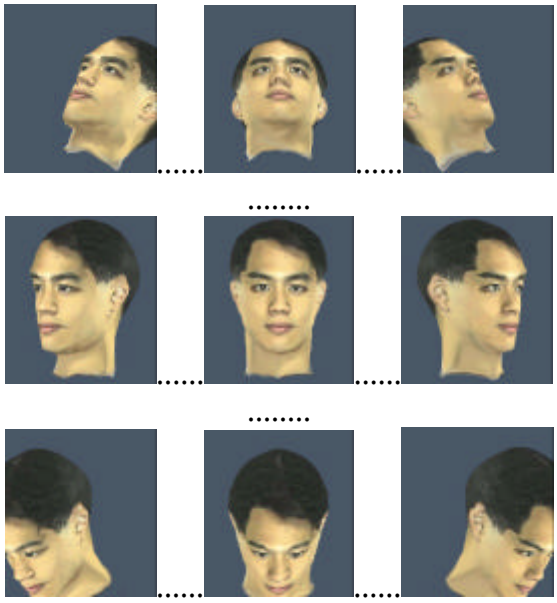

**Figure 6.** The 2D array of face images used to represent one's head motion.

## 5. Results

We have developed the system on a PC with a low-price video camera. The overall performance is above 25 frames/sec on a PC with a Pentium-II 400 MHz CPU. The background can be complex and dynamic, as shown in Figure 7.

The system has been used as a prototype system demonstrating the concept of model-based videophone. Because of the assistance from a user, the system can work with different people, different video cameras, and different lighting conditions. The requirement of user assistance is reasonable and practical since only three features are necessary. Furthermore, the three features, two eyes and one nostril, are easy to be identified manually. The identification process can be done within 3 seconds. From our experiences, the system can perform in a more flexible environment, which proves the policy to be effective and practical.

## 6. Acknowledgement

This work is part of the project "MPEG-4 Hybrid-Media Virtual Environment" developed by Communication and Multimedia Laboratory. The project is supported by the National Science Council under grant number ?????.

## 7. References

1. David Machin, *"Real-Time Facial Motion Analysis for Virtual Teleconferencing,"* Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (ICAFGR'96), Killington, Vermont, USA, Oct. 14-16, 1996, pp. 340-344.
2. Liang Zhang, *"Tracking a face for knowledge-based coding of videophone sequences,"* Signal Processing: Image Communication, 10, 1997, pp. 93-114.

3. R. Pappu, P. A. Beardsley, *"A Qualitative Approach to Classifying Gaze Direction,"* Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (ICAFGR'98), Nara, Japan, Apr. 14-16, 1998, pp. 160-165.
4. Jochen Heinzmann, Alexander Zelinsky, *"3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm,"* Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (ICAFGR'98), Nara, Japan, Apr. 14-16, 1998, pp. 142-147.
5. Jie Yang, Rainer Stiefelhagen, Uwe Meier, Alex Waibel, *"Visual Tracking for Multimodal Human,"* Proceedings of Conference on Human Factors in Computing Systems (CHI98), 18-23 Apr. 1998, pp. 140-147.
6. Qian Chen, Haiyuan Wu, Takeshi Fukumoto, Masahiko Yachida, *"3D Head Pose Estimation without Feature Tracking,"* Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (ICAFGR'98), Nara, Japan, Apr. 14-16, 1998, pp. 88-93.
7. K. Aizawa, H. Harashima, T. Saito, *"Model-Based Analysis Synthesis Image Coding (MBASIC) System for a Person's Face,"* Signal Processing: Image Communication, 1, 1989, pp. 139-152.
8. Tzong-Jer Yang, Fu-Che Wu, Ming Ouhyoung, *"Real-Time 3D Head Motion Estimation in Facial Image Coding,"* Proceedings of Multimedia Modeling (MMM'98), Lausanne, Switzerland, 12-15 Oct., 1998, pp. 50-51.
9. Zhengyou Zhang, Richid Deriche, Olivier Faugeras, Quang-Tuan Luong, *"A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry,"* Artificial Intelligence, 78, 1995, pp. 87-119.
10. Gary R. Bradski, "*Computer Vision Face Tracking for Use in a Perceptual User Interface*," Intel Technical Journal Q2, 1998. Http://developer.intel.com/technology/itj/q21998/articles/art_2.htm.
11. Jie Yang, Alex Waibel, "*A Real-Time Face Tracker*," Proc. of the 3rd IEEE workshop on Applications of Computer Vision, Sarasota, Florida, 1996, pp. 142-147 (also in Technical Report CMU-CS-95-210, 1995).
12. Thomas S. Huang, Arun N. Netravali, "*Motion and Structure from Feature Correspondences: A Review*," Proc. of the IEEE, 82(2), Feb. 1994, pp. 252-268.
13. Juyang Weng, Thomas S. Huang, Narendra Ahuja, "*Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation*," IEEE Trans. On Pattern Analysis and Machine Intelligence, 11(5), May 1989, pp. 451-476.
14. 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, 1994, pp. C503-512.
15. 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, Sep. 1987, pp. 698-700.
16. EPFL's generating face models from three images. Check.

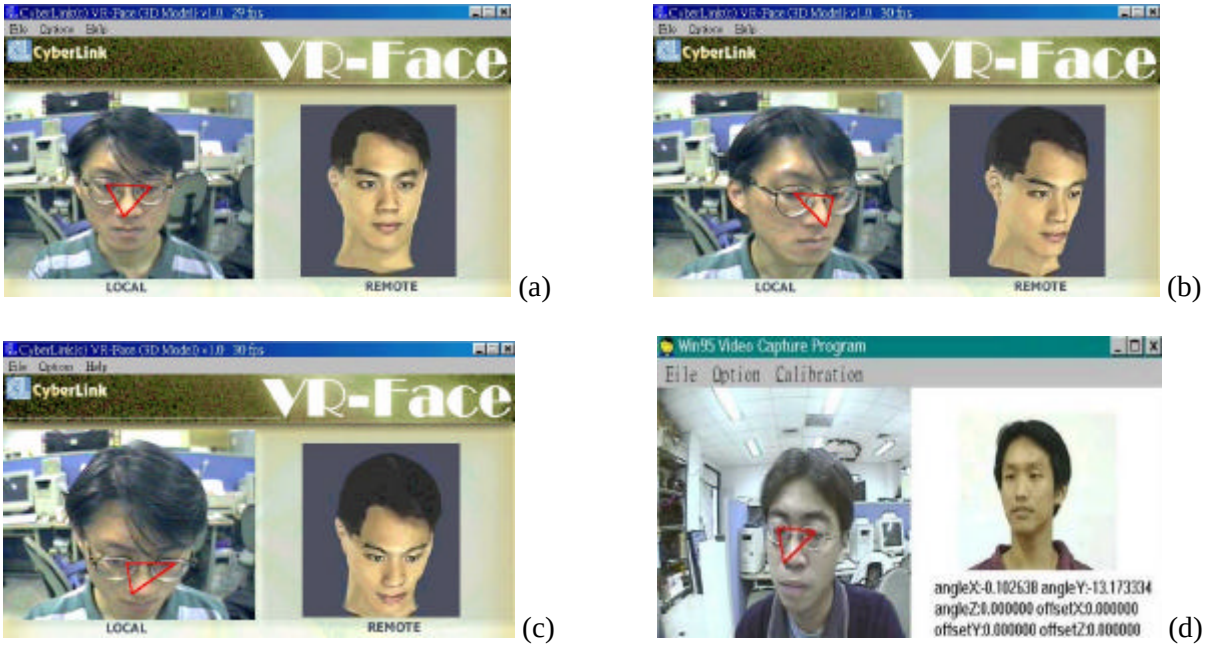


**Figure 7.** (a) The first frame after a user identifying three feature points. The right window shows a face image rendered from a 3D model with corresponding orientation. (b) The user turns his face to the left. (c) Turn down. (d) An alternative way to display corresponding face images using real photos.