Model-based Head Pose Tracking With Stereovision

Ruigang Yang[1] and Zhengyou Zhang

http://research.microsoft.com/~zhang/
E-mail:zhang@microsoft.com

October 2001

Technical Report
MSR-TR-2001-102

We present a robust model-based stereo head tracking algorithm that operates in real time on a commodity PC. The use of an individualized three-dimensional head model, coupled with the epipolar constraint from the stereo image pair, greatly improves the robustness of the tracking. Experimental results have shown that our method is able to track all the six degrees of freedom of the rigid part of head motions, over extended period of time, in the presence of large angular and translational head motions, partial occlusions, and/or dramatic facial expression changes. Applications include human-computer interaction and eye-gaze correction for video conferencing.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1]Current address: Department of Computer Science, University of North Carolina at Chapel Hill, USA

Contents

1 Introduction **2**

2 Related Works **2**

3 System Overview **3**

4 Stereo 3D Head Pose Tracking **3**
4.1 Models 4
4.2 Stereo Tracking 4
4.3 Feature Regeneration 5
4.4 Tracker Initialization and Auto-Recovery 7

5 Experiment Results **7**
5.1 Validation 9

6 Discussions and Conclusions **10**

1 Introduction

This work is motivated by our desire to establish eye contact for video conferencing on desktops. Our plan of attack is to track the head pose so we could intelligently warp a face image to generate a virtual view that preserves eye contact. To be successful, the tracking must (a) be able to track all six degrees of freedom accurately, which eliminates the use of some real-time tracking methods based simplistic schemes such as color histogram or ellipsoidal £tting; and (b) operates in real time, which places considerable constraints on the type of processing that can be performed. Our approach to reconciling these two seemingly incompatible requirements is to incorporate a detailed individualized three-dimensional head model with stereoscopic analysis.

The use of a detailed three-dimensional head model provides the tracker with rich geometric knowledge about the subject, thus (a) we are able to track the head pose with very few feature points, and (b) we can label each tracked feature with a semantic meaning. Such semantic information allows us to deal gracefully with occlusions and facial deformations. On the other hand, stereoscopic analysis provides the important epipolar constraint. By applying this constraint to the stereo image pair, we can easily reject most outliers (false matches from monocular tracking), thus avoiding using robust estimation techniques which tend to be more time-consuming. Furthermore, as to be demonstrated in the experimental section, using an extra camera dramatically improves the tracking accuracy and simpli£es the tracking algorithm. We recognize that there is a tradeoff between the equipment requirement and the tracking accuracy. With today's wide availability of inexpensive video cameras and increasingly better support of streaming video in the OS level, we believe that using a stereovision system (two or even more cameras) is well justi£ed.

2 Related Works

There is a wide variety of work related to 3D head tracking. Virtually all work on face tracking takes advantage of the constrained scenario: instead of using a generic tracking framework which views the observed face as an arbitrarily object, a model-based approach is favored, which incorporates knowledge about facial deformations, motions and appearance [6]. Based on the tracking techniques, we classify previous works into the following categories:

Optical ¤ow: Black and Yacoob [3] have developed a regularized optical-¤ow method in which the head motion is tracked by interpretation of optical ¤ow in terms of a planar two-dimensional patch. Basu et al. [2] generalized this approach by interpreting the optical ¤ow £eld using a 3D model to avoid the singularities of a 2D model. Better results have been obtained for large angular and translational motions. However, their tracking results were still not very accurate; as reported in their paper, angular errors could be as high as 20 degrees. Recently, DeCarlo and Metaxas [6] used optical ¤ow as a hard constraint on a deformable detailed model. Their approach has produced excellent results. But the heavy processing in each frame makes a real-time implementation dif£cult. Other ¤ow based methods include [4, 8].

Features and Templates: Azarbeyajani and Pentland [1] presented a recursive estimation method based on tracking of small facial features like the corners of the eyes or mouth using an extended Kalman-Filter framework. Horprasert [7] presented a fast method to estimate the head pose from tracking only £ve salient facial points: four eye corners and the nose top. Other template-based methods include the work of Darrell et al. [5], Saulnier et al. [11], and Tian et al. [13]. The template-based methods usually have the limitation that the same points must be visible over the entire image sequence, thus limiting the range of head motions they can track.

Skin Color: Yang et al. [15] presented a techniques of tracking human faces using an adaptive stochastic model based on human skin color. This approach is in general very fast. The drawback is that it is usually not very accurate, thus is not suf£cient for our applications.

The work by Newman et al. [10] is related to our work, and falls in the "Features and Templates" category. It also uses a stereo vision system, although the con£guration is different (we use a vertical setup for higher disambiguation power in feature matching). Their tracking technique is also different. They £rst take three snapshots (frontal, 45° to the left, and 45° to the right), and reconstruct up to 32 features selected on the face. Those 3D points, together with the templates extracted from the corresponding snapshots around each feature, are used for face tracking. In our case, we use a much more detailed face model, and

features are selected at runtime, making our system more robust to lighting change, occlusion and varying facial expression.

3 System Overview

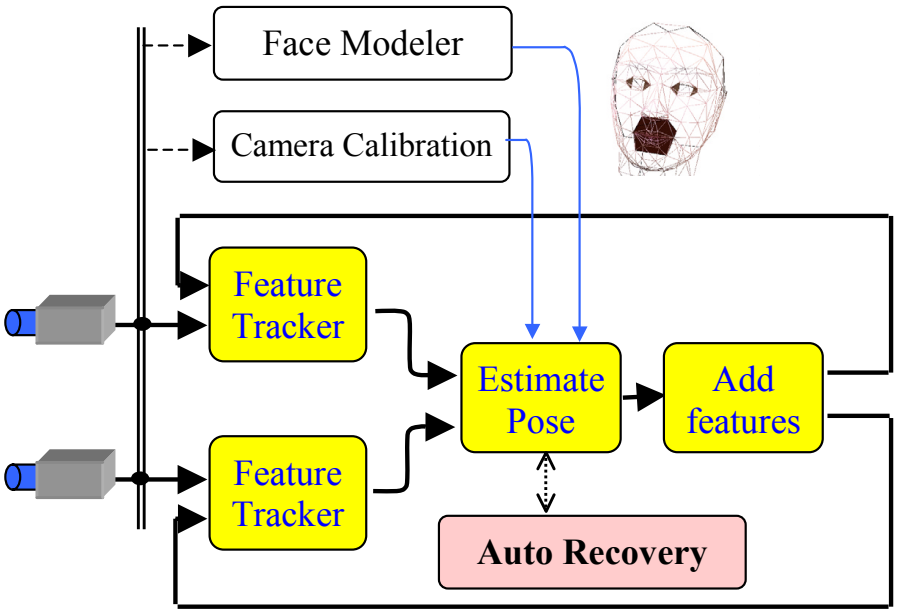

Figure 1: Model-based Stereo 3D Head Tracking System

Figure 1 illustrates the block diagram of our tracking system. Images are captured by a stereo camera pair. In our experimental setup, we use two digital video cameras mounted vertically, one on the top and the other on the bottom of the display screen. They are connected to a PC through 1394 links. We calibrate these cameras using the method in [17]. We choose the vertical setup because it provides higher disambiguation power in feature matching. Matching ambiguity usually involves facial features such as eyes and lip contours aligned horizontally. The user's personalized face model is acquired using a rapid face modeling tool [9]. Both the calibration and model acquisition require little human interaction, and a novice user can complete these tasks within 15 minutes. Furthermore, they only need to be done once per user per £xed setup.

The entire tracking process is automatic, except for the initialization, which requires the user to select seven landmark features in the £rst pair of frames. The subject is required to remain relative still and maintain a neutral expression. Using these marked features, the face model is registered with the images. From then on, our system tracks the optical ¤ow of salient features, rejects outliers based on the epipolar constraint, and updates the head pose on a frame-by-frame basis. A feedback loop supplies fresh salient feature points at each frame to make the tracking more stable under various conditions. Furthermore, an automatic tracking recovery mechanism is also implemented to make the whole system even more robust over extended period of time.

4 Stereo 3D Head Pose Tracking

We now provide more details of our tracking system.

4.1 Models

The face model we use is a triangular mesh consisting approximately 300 triangles. Each vertex in the mesh has semantic information, i.e., eye, chin, etc. We build a personalized face model for each user using the rapid face modeling tool developed by Liu et al. [9]. Figure 2 shows a sample face model[1]. Note that although the face model contains other properties such as textures, we only use the geometric and semantic information in our tracking system.

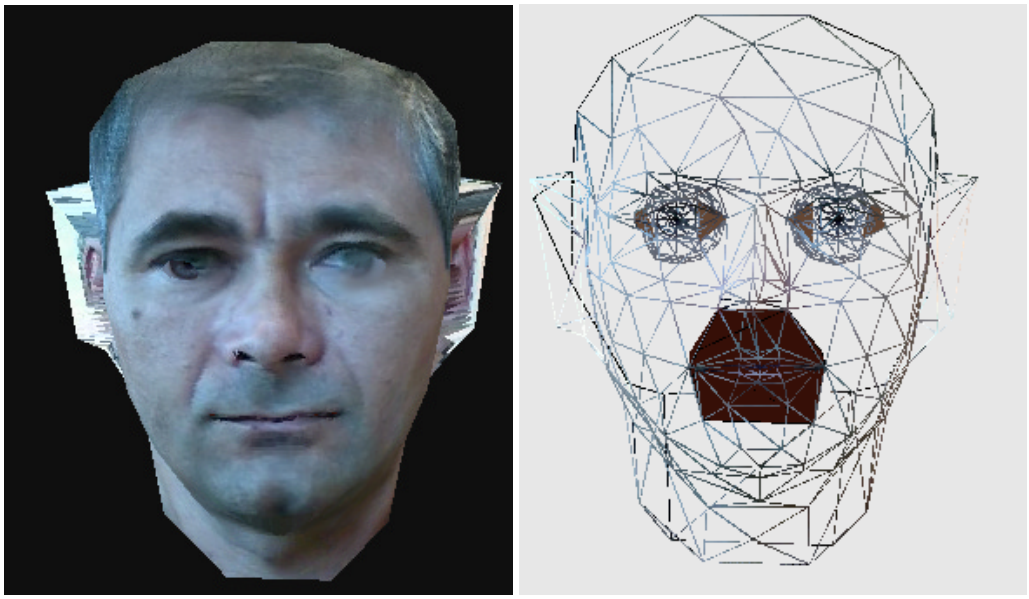

Figure 2: A sample face model. The wireframe on the right reveals the artiFcial eyes and month, they are not being used in our tracking system.

A camera is modeled as a pinhole, and its intrinsic parameters are captured in a 3×3 matrix. The intrinsic matrices for the stereo pair are denoted by $\mathbf{A}_0$ and $\mathbf{A}_1$, respectively. Without loss of generality, we use the Frst camera's (Camera 0) coordinate system as the world coordinate system. The second camera's (Camera 1) coordinate system is related to the Frst one by a rigid transformation $(\mathbf{R}_{10}, \mathbf{t}_{10})$. Thus, a point $\mathbf{m}$ in 3D space is projected to the image planes of the stereo cameras by

$$\mathbf{p} = \phi(\mathbf{A}_0\mathbf{m}) \quad (1)$$

$$\mathbf{q} = \phi(\mathbf{A}_1(\mathbf{R}_{10}\mathbf{m} + \mathbf{t}_{10})) \quad (2)$$

where $\mathbf{p}$ and $\mathbf{q}$ are the image coordinates in the Frst and second camera, and ϕ is a 3d-2D projection function such that $\phi(\begin{bmatrix} u \\ v \\ w \end{bmatrix}) = \begin{bmatrix} u/w \\ v/w \end{bmatrix}$. We use the method in [17] to determine $(\mathbf{A}_0, \mathbf{A}_1, \mathbf{R}_{10}, \mathbf{t}_{10})$.

The face model is described in its local coordinate system. The goal of our tracking system is to determine the rigid motion of the head (*head pose*) in the world coordinate system. The head pose is represented by a 3 rotation matrix $\mathbf{R}$ and a 3D translation vector $\mathbf{t}$. Since a rotation only has three degrees of freedom, the head pose requires 6 parameters.

4.2 Stereo Tracking

Our stereo head tracking problem can be formally stated as follows:

Given a pair of stereo images $I_{0,t}$ and $I_{1,t}$ at time t, two sets of matched 2D points $S_0 = \{\mathbf{p}{=}[u, v]^T\}$ and $S_1 = \{\mathbf{q}{=}[a, b]^T\}$ from that image pair, their corresponding 3D model points $M = \{\mathbf{m}{=}[x, y, z]^T\}$, and a pair of stereo images $I_{0,t+1}$ and $I_{1,t+1}$ at time $t{+}1$,

[1] Their system does not try to model the ears and the back of a head.

determine (i) a subset $M' \subseteq M$ whose corresponding $\mathbf{p}$'s and $\mathbf{q}$'s have matches, denoted by $S_0' = \{\mathbf{p}'\}$ and $S_1' = \{\mathbf{q}'\}$, in $I_{0,t+1}$ and $I_{1,t+1}$, and (ii) the head pose $(\mathbf{R}, \mathbf{t})$ so that the projections of $\mathbf{m} \in M'$ are $\mathbf{p}'$ and $\mathbf{q}'$.

We show a schematic diagram of the tracking procedure in Figure 3.

We first conduct independent feature tracking for each camera from time t to $t+1$. We use the KLT tracker [12] which works quite well. However, the matched points may be drifted or even wrong. Therefore, we apply the epipolar constraint to remove any stray points. The epipolar constraint states that if a point $\mathbf{p} = [u, v, 1]^T$ (expressed in homogeneous coordinates) in the first image and a point $\mathbf{q} = [a, b, 1]^T$ in the second image correspond to the same 3D point $\mathbf{m}$ in the physical world, they must satisfy the following equation:

$$\mathbf{q}^T \mathbf{F} \mathbf{p} = 0 \tag{3}$$

where $\mathbf{F}$ is the fundamental matrix [2] that encodes the epipolar geometry between the two images. In fact, $\mathbf{F}\mathbf{p}$ defines the epipolar line in the second image, thus Equation (3) means that the point $\mathbf{q}$ must pass through the epipolar line $\mathbf{F}\mathbf{p}$, and vice versa.

In practice, due to inaccuracy in camera calibration and feature localization, we cannot expect the epipolar constraint to be satisfied exactly. For a triplet $(\mathbf{p}', \mathbf{q}', \mathbf{m})$, if the distance from $\mathbf{q}'$ to the $\mathbf{p}'$s epipolar line is greater than a certain threshold, this triplet is considered to be an outlier and is discarded. We use a distance threshold of three pixels in our experiments.

After we have removed all the stray points that violates the epipolar constraint, we update the head pose $(\mathbf{R}, \mathbf{t})$ so that the re-projection error of $\mathbf{m}$ to $\mathbf{p}'$ and $\mathbf{q}'$ is minimized. The re-projection error e is defined as

$$e = \sum_i \big(\|\mathbf{p}_i' - \boldsymbol{\phi}(\mathbf{A}_0(\mathbf{R}\mathbf{m}_i + \mathbf{t}))\|^2 + \|\mathbf{q}_i' - \boldsymbol{\phi}(\mathbf{A}_1[\mathbf{R}_{10}(\mathbf{R}\mathbf{m}_i + \mathbf{t}) + \mathbf{t}_{10}])\|^2 \big) \tag{4}$$

We solve $(\mathbf{R}, \mathbf{t})$ using the Levenberg-Marquardt algorithm, and the head pose at time t is used as the initial guess.

4.3 Feature Regeneration

After the head pose is determined, we replenish the matched set S_0', S_1' and M' by adding more *good* feature points. We select a *good* feature point based on the following three criteria:

- **Texture**: The feature point in the images must have rich texture information to facilitate the tracking. We first select 2D points in the image using the criteria in [12], then back-project them back onto the face model to get their corresponding model points.
- **Visibility**: The feature point must be visible in both images. We have implemented an intersection routine that returns the first visible triangle given an image point. A feature point is visible if the intersection routine returns the same triangle for its projections in both images.
- **Rigidity**: We must be careful not to add feature points in the non-rigid regions of the face, such as the month region. We define a bounding box around the tip of the nose that covers the forehead, eyes, nose, and cheek region. Any points outside this bounding box will not be added to the feature set.

This Regeneration scheme improves our tracking system in two ways. First, it replenishes the features points lost due to occlusions or non-rigid motion, so the tracker always has a sufficient number of features to start with in the next frame. This improves the accuracy and stability. Secondly, it alleviates the problem of tracker drifting by adding fresh features at every frame.

[2] The fundamental matrix is related to the camera parameters as $\mathbf{F} = {\mathbf{A}_1}^{-T}[\mathbf{t}_{10}]_\times \mathbf{R}_{10} {\mathbf{A}_0}^{-1}$.

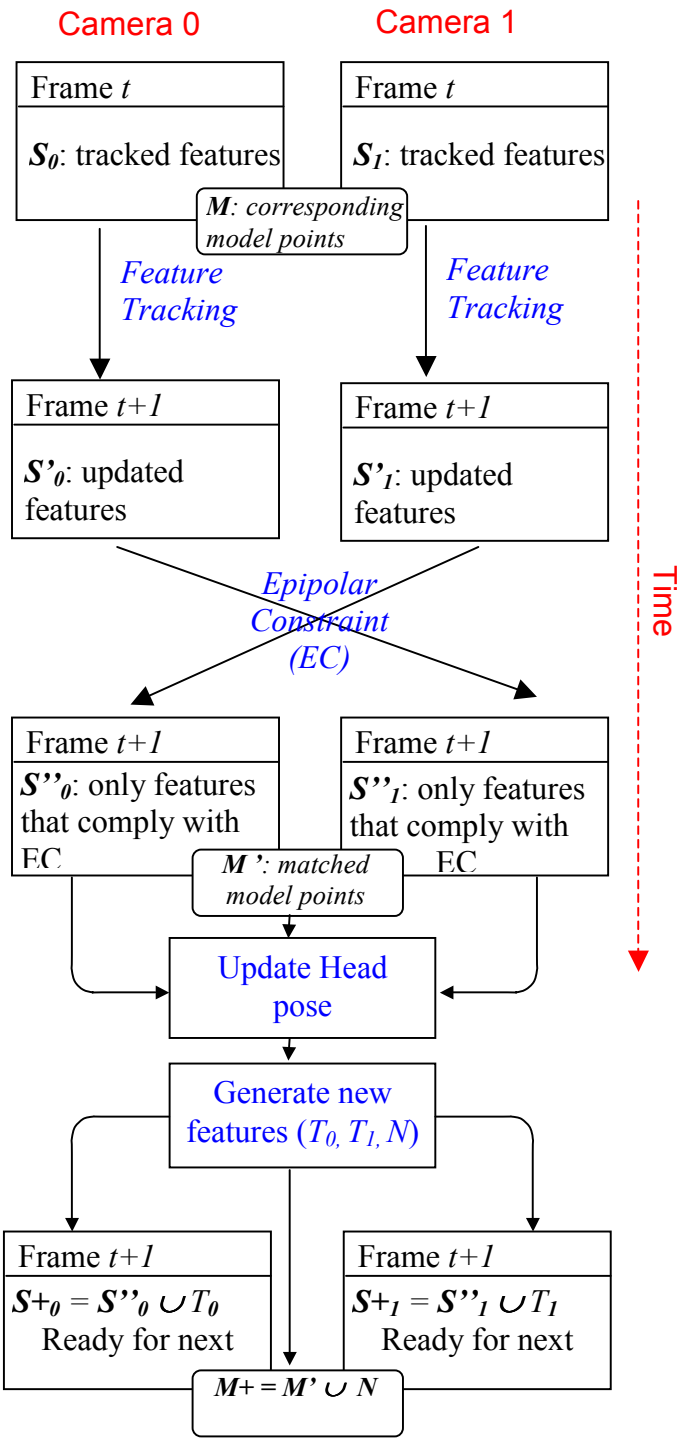

Figure 3: Model-based stereo 3D head tracking

4.4 Tracker Initialization and Auto-Recovery

The tracker needs to know the head pose at time 0 to start tracking. We let the user interactively select seven landmark points in each image, from which the initial head pose can be determined. We show an example of the selected feature points in Figure 4, where the epipolar lines in the second image is also drawn. The manual selection does not have to be very accurate. We automatically re£ne the selection locally to satisfy the epipolar constraint.

The initial selection is also used for tracking recovery when the tracker loses tracking. This may happen when the user moves out of the camera's £eld of view or rotates her head away from the cameras. When she turns back to the cameras, we prefer to continue tracking with minimum or no human intervention. During the tracker recovery process, the initial set of landmark points is used as templates to £nd the best match in the current image. When a match with a high con£dence value is found, the tracker continues the normal tracking.

Figure 4: Manually selected feature points; the epipolar lines are overlayed on the 2nd image.

Furthermore, we also activate the auto-recovery process whenever the current head pose is close to the initial head pose. This further alleviates the tracker drifting problem, the accumulative error is reduced after tracker recovery. This scheme could be extended to include multiple templates at different head poses. This is expected to further improve the robustness of our system.

5 Experiment Results

We have implemented our tracking algorithm using C++ under the MS Windows environment and tested with live real data. Our current implementation runs in real-time (20-30fps) on a PC with a 1.5 GHz Pentium 4 CPU. We will here present results from three test sequences, all of which were collected with a resolution of 320×240 at 30 frame per second. The £rst two sequences were captured with a pair of inexpensive web cameras while the last one was captured with a pair of SONY digital video camera (DFW-L500). The SONY camera produces better images under low lighting conditions. Under bright light, the image quality from web cameras is comparable to the SONY camera. Thus we shot all of them under relatively bright lighting. Consequently, there is no noticeable difference of tracking quality among these sequences.

Figures 5 shows some results of the £rst sequence (A). The 3D face mesh is projected according to the estimated head pose and is overlayed on the input stereo images. This sequence contains large head rotations close to 90 degrees. This type of out-of-plane rotation is usually dif£cult for head tracking, but we can see that our algorithm determines accurately the head pose, thanks to the 3D mesh model.

The second sequence (B), shown in Figure 6, contains predominantly non-rigid motion (dramatic facial expressions). We also show the original images to better appreciate the non-rigid motion. Because we classify the face into rigid and non-rigid areas and use features from the rigid areas, our tracker is insensitive to non-rigid motion.

Figure 7 shows the last sequence (C) in which large occlusions and out-of-plane head motions frequently appear. Our system maintains accurate tracking throughout the entire 30- second sequence.

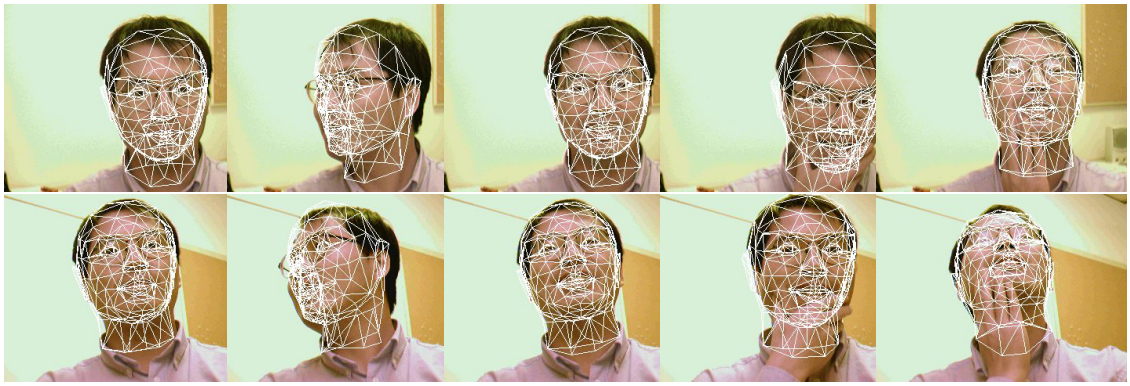

Figure 5: Stereo tracking result for Sequence A (320x240 @ 30 FPS). Images from the £rst camera are shown in the upper row, while those from the second camera are shown in the lower row. From left to right, the frame numbers are 1, 130, 325, 997, and 1256.

Figure 6: Stereo tracking result for Sequence B (320x240 @ 30 FPS); The £rst row shows the input images from the upper camera. The second and third rows show the projected face model overlayed on the images from the upper and lower camera, respectively. From left to right, the frame numbers are 56, 524, 568, 624, and 716.

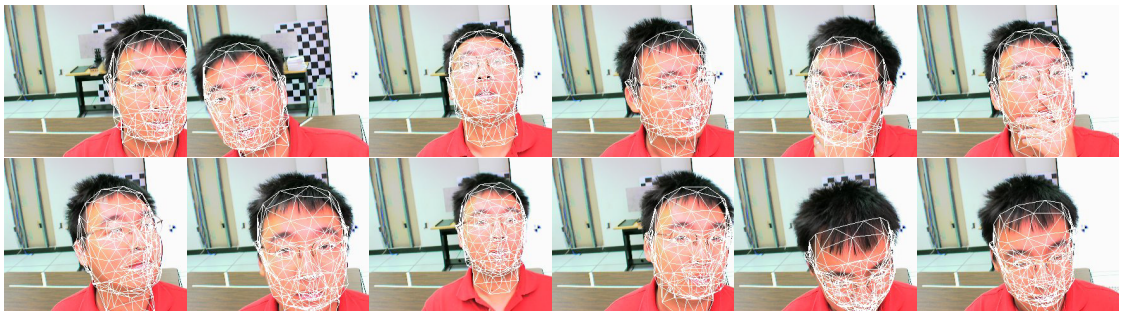

Figure 7: Stereo tracking result for Sequence C (320x240 @ 30 FPS) The frame numbers, from left to right and from top to bottom, are 31, 67, 151, 208, 289, 352, 391, 393, 541, 594, 718, and 737.

5.1 Validation

For the purpose of comparison, we have also implemented a model-based monocular tracking technique. Like most prevalent methods, we formulate it as an optimization problem that seeks to minimize the re-projection errors between the projected 3D features points and the actual tracked features. Using the same notions as in (4), the monocular cost function is de£ned by

$$e_m = \sum_i \|\mathbf{p}'_i - \phi(\mathbf{A}_0(\mathbf{R}\mathbf{m}_i + \mathbf{t}))\|^2 \tag{5}$$

We solve the optimization problem using the Levenberg-Marquardt method.

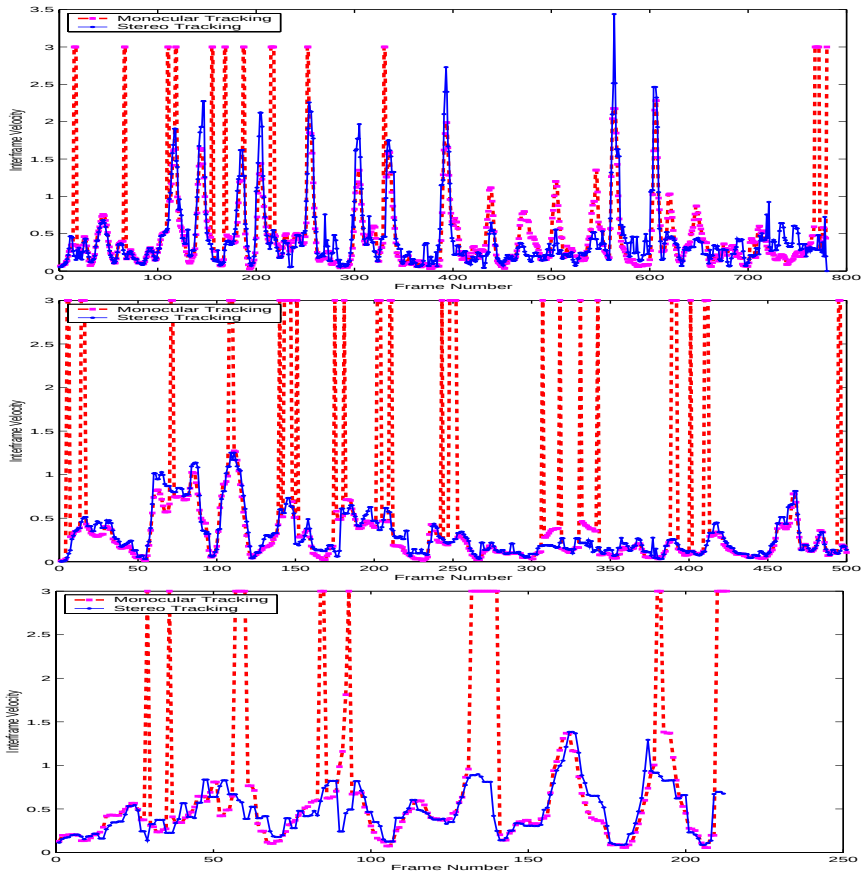

Figure 8: A comparison between monocular and stereo tracking in terms of the estimated velocity of head motion. Results from Sequence A, B, and C are shown from top to bottom.

We run the monocular tracking algorithm over the three sequences. Since we do not know the ground truth of head motions, it is meaningless to compare the absolute values from the two algorithms. Instead, we compare the approximate velocity $\tilde{v} = \|\mathbf{t}_{i+1} - \mathbf{t}_i\|/\delta t$. The head motion is expected to be smooth, and so is the velocity curve. We plot the velocity curves of the three sequences in Figure 8. The x-axis is the frame number and the y-axis is the speed (inches/frame). The velocity curve computed using the monocular algorithm is plotted in red, while that from the stereo in blue. In the red curves, there are several spikes that well exceed the limit of normal head motion (a maximum cap of 3 inches/frame is put in the plots; some of the spikes are actually higher than that). We suspect that they indicate that tracking is lost or the optimization is trapped in a local minimum. On the other hand, the blue curves have signi£cant less or even no spikes. The only spikes in blue curves are in the £rst sequence (A), which indeed contains abrupt head motions. We also visually compare the results for Sequence C between the monocular and stereo tracking method in Figure 9. These images are selected corresponding to the spikes in the red curve

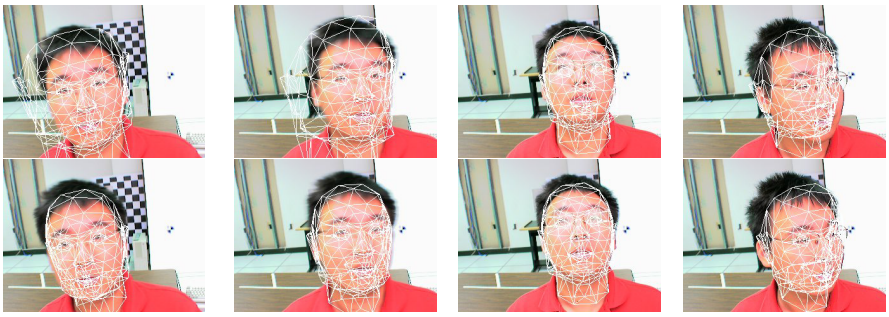

Figure 9: Visual Comparison of the monocular (upper row) vs. stereo (lower row) tracking method. From left to right, the frame numbers are 54, 90, 132, and 207.

for sequence C. The top row shows the monocular tracking results and the second row shows the stereo tracking results. For those in the £rst row, some obviously have lost tracking, while the others have poor accuracy in head pose estimation.

We should point out that the plots only show the results up to when the monocular tracker reported that the optimization routine failed to converge for 10 consecutive frames. On the other hand, the stereo tracker continued until the end of the sequence. The rich information from the stereo cameras enables the stereo tracker to achieve a much higher level of robustness than the monocular version.

6 Discussions and Conclusions

We have presented a robust method for real-time 3D face tracking using stereovision. The combined use of a detailed 3D head model with stereoscopic analysis allows accurate full 3D head pose estimation in the presence of partial occlusions and dramatic facial deformations, as demonstrated with several real sequences. Furthermore, we have compared our method against a monocular tracking method. Experiment results have shown signi£cant improvements in both robustness and accuracy.

There are still places we want to improve, however. One of them is the way to deal with facial deformations. In our current work, we use a simple £xed classi£cation of rigid and non-rigid facial regions. A dynamic classi£cation, according to actual facial expression, would be preferred.

Looking into the future, techniques that automatically locating the face and its various feature points [16, 14] can be integrated to initialize the tracker, and then the entire system will be fully automatic. With rapidly reduced cost of video cameras, we expect to £nd its use in a variety of applications from multimedia user interfaces to video coding.

Acknowledgment

The authors would like to thank all members of the Collaboration and Multimedia Group at Microsoft Research for help and stimulating discussions.

References

[1] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive estimation of structure and motion using the relative orientation constraint. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 70–75, 1993.

[2] Sumit Basu, Irfan Essa, and Alex Pentland. Motion Regularization for Model-based Head Tracking. In *Proceedings of International Conference on Pattern Recognition*, Wien, Austria, 1996.

[3] M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric model of image motion. In *Proceedings of International Conference on Computer Vision*, pages 374–381, Cambridge, MA, 1995.

[4] C. Choi, K. Aizawa, H. Harashima, and T. Takebe. Analysis and synthesis of facial image sequences in model-based image coding. *IEEE Circuits and Systems for Video Technology*, 4(3):257–275, 1994.

[5] T. Darrell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. In *In IEEE Computer Vision and Pattern Recognition*, pages 67–72, 1996.

[6] Douglas DeCarlo and Dimitris Metaxas. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2001.

[7] T. Horprasert. Computing 3-D Head Orientation from a Monocular Image. In *International Conference Automatic Face and Gesture Recognition*, pages 242–247, 1996.

[8] H. Li, P. Roivainen, and R. Forchheimer. 3-D motion estimation in model-based facial image coding. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[9] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen. Rapid Modeling of Animated Faces From Video. In *The Third International Conference on Visual Computing (Visual 2000)*, pages 58–67, Mexico City, September 2000. Also available as Technical Report MSR-TR-99-21.

[10] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, pages 122–128, Grenoble, France, 2000.

[11] A. Saulnier, M. L. Viaud, and D. Geldreich. Real-time Facial Analysis and Synthesis Chain. In M. Bichsel, editor, *International Workshop on Automatic Face and Gesture Recogntion*, pages 86–91, Zurich, Switzerland, 1995.

[12] J. Shi and C. Tomasi. Good features to track. In *the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, Washington, June 1994.

[13] Y.-L. Tian, T. Kanade, and J.F. Cohn. Recognizing Action Units for Facial Expression Analysis. *Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[14] Y. Yacoob and L. S. Davis. Computing Spatio-temporal Representations of Human Faces. In *Proceeding of CVPR*, pages 70–75, 1994.

[15] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Real-time face and facial feature tracking and applications. In *Proceedings of AVSP'98*, pages 79–84, Terrigal, Australia, 1998.

[16] A. L. Yuille, D. S. Cohen, and P. Halliman. Feature Extraction from Faces Using Deformable Templates. *International Journal of Computer Vision*, 8:104–109, 1992.

[17] Zhengyou Zhang. A ¤exible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.