MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28-30, 2000, The University of Tokyo, Japan

13—16 Feature-based real-time human face tracking using Lie algebras

Akira Inoue* Tom Drummond** Roberto Cipolla**

*Computer & Communication Media Research NEC Corporation
**Department of Engineering University of Cambridge

Abstract

We have developed a novel human face tracking system that operates in real time at the video frame rate without a need for any special hardware. Our approach is based on the use of Lie algebras, and uses 3D feature points on the target human face. The resulting tracker performed very well on the task of tracking a human face.

1 Introduction

The visual tracking of human faces is useful for numerous applications, including human computer interaction, surveillance and face identification.

Recovering the human face position and orientation automatically from a video is a challenging task. The difficulty comes from the wide range of appearances that can be generated under various illumination conditions and facial expressions. Black and Yacoob[1] applied parametric 2D models to track the motion of a user's head. Blake and Isard[2] have proposed a probabilistic tracking paradigm, and applied it to a human face with edge features. Cootes and Taylor[3] have proposed a tracking technique with deformable face models which is constrained by a statistical model of shape. While most studies neglect real-time processing, Harville and Rahimi[4] have developed a video-rate face tracking system with stereo cameras and a video-rate range sensor.

Our goal is to develop a video-rate face tracking system that requires no special hardware so that it can be used on popular personal computers. Recently, Drummond and Cipolla[5] have developed a real-time visual tracking system for complex structures like ship models. This fast system used the edge features of the target, and Lie algebras[6] were introduced for estimating motions.

Our proposed face tracking system uses 3D feature points on the target human face instead of edges. We have also adapted Lie algebras to the algorithm of estimating motion matrices of a target human face.

*Address: 4-1-1 Miyazaki, Miyamae-ku, Kawasaki, Kanagawa, 216-8555 Japan
E-mail: a-inoue@cp.jp.nec.com
**Address: Cambridge CB2 1PZ, UK

2 Motion estimation

The approach for tracking is based on estimating the camera projection matrix, P, for the facial 3D model. This projection matrix, P, is represented as the product of a matrix of internal camera parameters, K, and a Euclidean projection matrix, E, representing the position and orientation of the camera relative to the target object:

$$\mathbf{E} = [\mathbf{R} \mid \mathbf{t}] \ \text{with}\ \mathbf{R}\mathbf{R}^{T} = \mathbf{I} \ \text{and}\ |\mathbf{R}| = 1 \quad (1)$$

The projective coordinates[7] of the feature points are given by

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{P}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{KE}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

with actual image coordinates given by

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} u/w \\ v/w \end{pmatrix} \quad (3)$$

We assumed that K is given. Therefore, the key algorithm is to estimate the Euclidean matrix E, which means the rigid motion parameters of the camera. While a 3x4 Euclidean matrix has twelve elements, it has only six d.o.f. Because of this redundancy, it was difficult to obtain a reliable solution.

To do this estimation, we adopted the technique of representing differences in the Euclidean matrix by six parameters that describe camera motion using Lie algebra. By solving the equation, the six motion parameters are given directly.

Rigid motion of the camera between video frames can be represented by the right side multiplication of the form:

$$\mathbf{M} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

This M forms a 4x4 matrix representation of the group SE(3), which is a 6-dimensional Lie group. The following matrices represent the generators of this group we use, which mean translations in the x, y, and z directions, and rotations about the x, y, and z axes.

$$\mathbf{G}_1 = \begin{pmatrix} 0&0&0&1\\0&0&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \mathbf{G}_2 = \begin{pmatrix} 0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0 \end{pmatrix},$$
$$\mathbf{G}_3 = \begin{pmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0 \end{pmatrix}, \mathbf{G}_4 = \begin{pmatrix} 0&0&0&0\\0&0&1&0\\0&-1&0&0\\0&0&0&0 \end{pmatrix}, \tag{5}$$
$$\mathbf{G}_5 = \begin{pmatrix} 0&0&-1&0\\0&0&0&0\\1&0&0&0\\0&0&0&0 \end{pmatrix}, \mathbf{G}_6 = \begin{pmatrix} 0&1&0&0\\-1&0&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}$$

These generators Gi-s form a basis for the vector space of derivatives of SE(3) at identity. We can calculate the matrix M from Gi and α_i, the quantity of each generated motion (m=6).

$$\mathbf{M} = \exp(\sum_{i=1}^{m} \alpha_i \mathbf{G}_i) \tag{6}$$

For small α_i, the equation is approximated as follows.

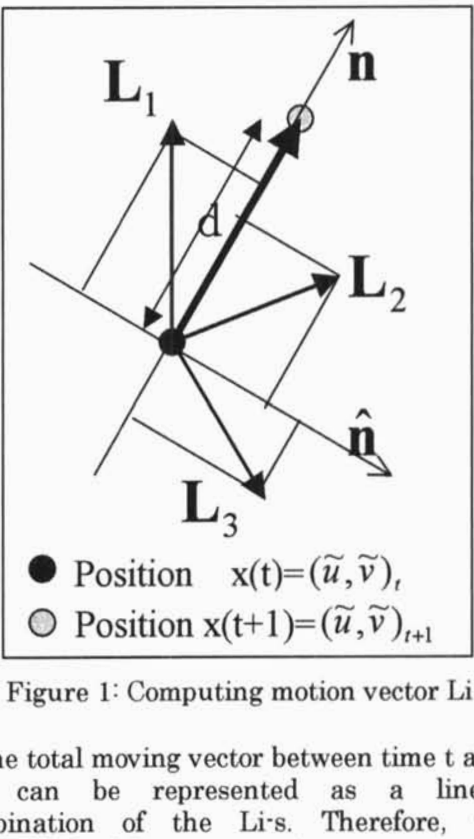
$$\mathbf{M} = \mathbf{I} + \sum_{i=1}^{m} \alpha_i \mathbf{G}_i \tag{7}$$

In order to estimate the quantity α_i, we have considered the partial derivative of the projective image coordinates under the i-th generating motion Li. This can be computed as:

$$\mathbf{L}_i = \begin{pmatrix} \tilde{u}' \\ \tilde{v}' \end{pmatrix} = \begin{pmatrix} \dfrac{u'}{w} - \dfrac{uw'}{w^2} \\ \dfrac{v'}{w} - \dfrac{vw'}{w^2} \end{pmatrix} \tag{8}$$

$$\text{with } \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \mathbf{PG}_i \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{9}$$

Figure 1 shows the relationship between the total motion vector and i-th generating motion vectors Li of a feature point.



● Position $x(t)=(\tilde{u},\tilde{v})_t$

○ Position $x(t+1)=(\tilde{u},\tilde{v})_{t+1}$

Figure 1: Computing motion vector Li

The total moving vector between time t and t+1 can be represented as a linear combination of the Li-s. Therefore, we obtained the following equations.

$$d = \sum_{i=1}^{m} \alpha_i (\mathbf{L}_i \bullet \mathbf{n})$$
$$0 = \sum_{i=1}^{m} \alpha_i (\mathbf{L}_i \bullet \hat{\mathbf{n}}) \tag{10}$$

$$\mathbf{n} = \frac{(\tilde{u},\tilde{v})_{t+1} - (\tilde{u},\tilde{v})_t}{\|(\tilde{u},\tilde{v})_{t+1} - (\tilde{u},\tilde{v})_t\|} \qquad (11)$$
$$(\mathbf{n},\hat{\mathbf{n}}) = 0$$

We estimate α_i, by a least square approach. Now we have the motion matrix M by Eq.(7). Then, the Euclidean matrix E is updated by:

$$\mathbf{E}_{t+1} = \mathit{Euclideanise}(\mathbf{E}_t \cdot (\mathbf{I} + \sum_i a_i \mathbf{G}_i)) \quad (12)$$

The Rotation matrix R of E must be orthogonal, and the singular values of R should be 1. Therefore we applied the following technique "Eulideanise()" using SVD to transform E.

$$\mathbf{R} = \mathbf{UWV}^T, \mathbf{R}' = \mathbf{UV}^T$$
$$\mathit{Euclideanise}(\mathbf{E}) = [\mathbf{R}' \mid \mathbf{t}] \qquad (13)$$

Using the above algorithm, we can fit the observed motion of the feature points. Once the Euclidean matrix is obtained, we can easily compute the pose and orientation of the target human face.

3 Implementation

We assume that the roughly estimated face model (relative coordinates of the 3D feature points) is known. First, we determine the initial position of the face model by a model fitting technique. Then, the tracking is operated by the following sequence.

1. Capture the new video frame and render feature points to the image plane.
2. Search for new positions of the feature points on the image plane.
3. Obtain the Euclidean matrix from the moving vector and the 3D information for the points.
4. Rotate and translate the feature points by using the Euclidean matrix and render the new points on the image.

In our implementation, we use a simple feature detector. That is to say, for detecting such feature points as eyes, nostrils and eyebrows, we used a dark-pixel finder which simply returns the location of the darkest pixel within local search windows. This enabled us to reduce computational cost and maximize tracking performance. Of course other detection methods are also available, such as the use of a corner detector or correlation matching according to the CPU power.

Initial model fitting performs a local feature search on the plane normal to the optical axis (Figure2). Updated 3D positions are given by:

$$\begin{pmatrix} zu \\ zv \\ z \end{pmatrix} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} = \mathbf{KRX} + \mathbf{Kt} \qquad (14)$$

$$\mathbf{X}' = (\mathbf{KR})^{-1}\left(\begin{pmatrix} zu' \\ zv' \\ z \end{pmatrix} - \mathbf{Kt}\right) \qquad (15)$$

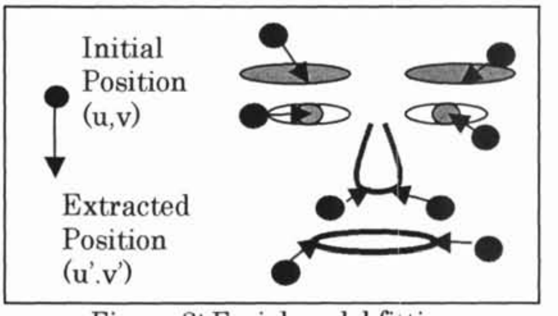


Figure 2: Facial model fitting

4 Experimental results

We did some tracking experiments with a real human face. These experiments were conducted with an SGI O2 workstation (225MHz). Figure 3 shows images captured in the tracking sequences. Facial orientation is represented as a square and line markers in the images.

Our experimental results were very satisfactory and showed that the tracker can successfully compute the position and attitude of a target human face in real time at video frame rate.

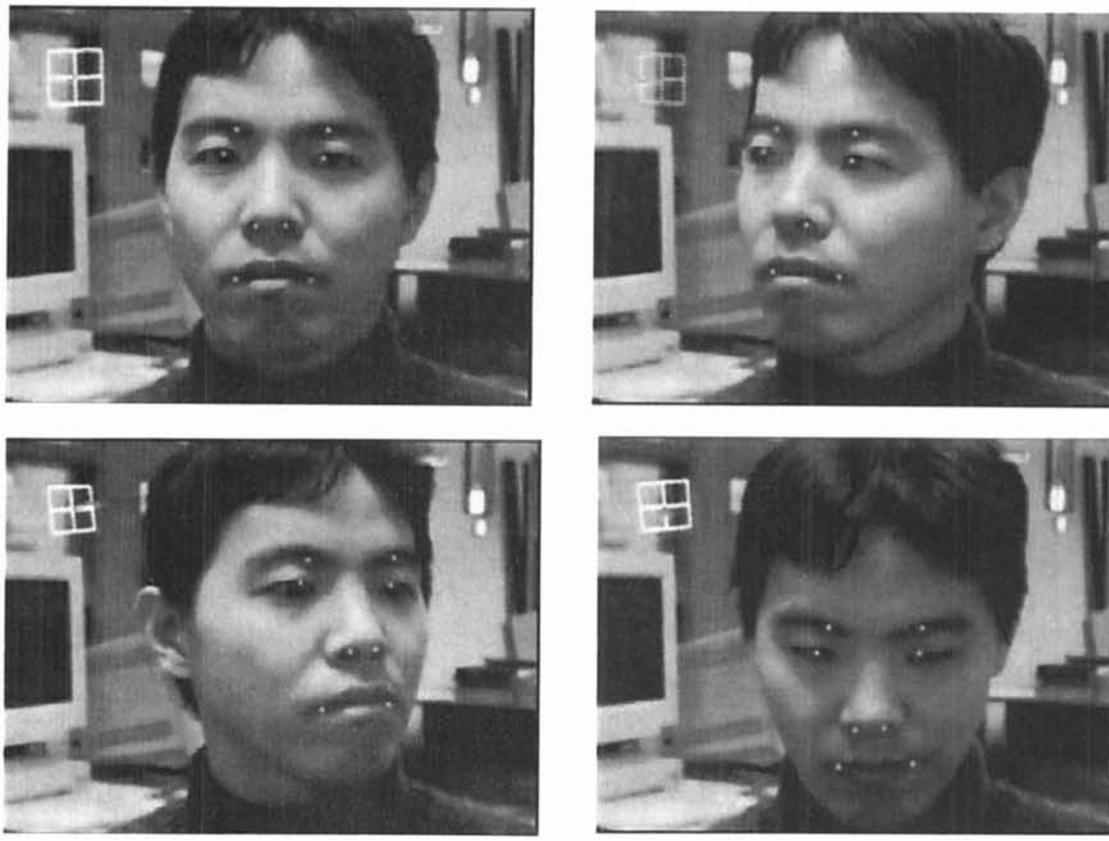


Figure 3: Experimental tracking results

5 Conclusion

We have developed a fast human face tracking system that can track in real-time at video-rate. The advantage of our system is that it does not need any special hardware such as an image processing chip or stereo cameras.

Additionally, we have applied a novel motion estimation technique using Lie algebras to the system. The six 3D motion parameters are directly estimated using this algorithm. This leads to better performance for the tracking system. In our experiments, the system has shown better stability than a conventional tracker using such tracking methods as the weak perspective technique.

References

[1] M. Black and Y. Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions", In Proceedings ICCV, pages 374-381. 1995

[2] A. Blake and M. Isard, "Active Contours", Addison Wesley, 1998

[3] T. F. Cootes, C. J. Taylor, "Active shape models – their training and application", Computer Vision and Image Understanding, 61(1): pages 38-59, 1995

[4] M. Harville, A. Rahimi, et.al. "3D Pose Tracking with Linear Depth and Brightness Constrains", In Proceedings ICCV, 1999

[5] T. Drummond, R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration", Proceedings of British Machine Vision Conference, pp.574-583, 1999.

[6] D. H. Sattinger, O. L. Weaver, "Lie groups and algebras with applications to physics, geometry, and mechanics", Number 61 in Applied Mathematical Sciences. Springer-Verlag, 1986

[7] O. Faugeras, "Three-dimensional computer vision: a geometric viewpoint", MIT Press, 1993