ISSN1000-9825 Journal of Software 软件学报 2000,11(9):1141~1150

High Resolution Calibration of Motion Capture Data for Realistic Facial Animation*

Chien-feng Huang I-chen Lin Ming Ouhyoung

(Communication and Multimedia Lab. Dept. of Computer Science and Info. Eng. Taiwan University)

E-mail: {cardy,ichen,ming}@cmlab.csie.ntu.edu.tw

Abstract A novel technique is proposed to make realistic facial animation driven by motion capture data. First, a person's facial expressions are recorded using Oxford Metrics' VICON 8 optical motion capture system, where 23 optical markers are attached on the person's face. The captured 3D motion data are then processed to eliminate 3D global head motion by finding the pivot point for rotation. These processed 3D motion data can be directly applied to a head model. A two and half dimension facial model is used here for implementation because it combines good features from both 2D mesh and 3D model: simple, vivid and natural, when small-scale rotation is applied. An interpolation function is employed to calculate the offsets of other unconstrained mesh vertices. Furthermore, a face is divided into several regions and force constraints are applied to limit the displacement of vertices on the mesh depending on the region to make the animation more natural. The system has an update rate of over 30 frames/second on a Pentium III 500MHz PC with an OpenGL acceleration card.

Key words Motion capture, virtual human, motion compensation, facial animation, synthetic face.

Realistic animation of human face is one of the most difficult goals in computer animation. Although there are good results in body motion, facial animation still appears unnatural and not vivid enough. Indeed, there are a few impressive facial animations, but they take a lot of animators working for a long time. There are several reasons that make facial animation so difficult. First, the geometric form of human face is extremely complex. Moreover, a face exhibits countless tiny creases and wrinkles, as well as subtle variations in color and texture.

A number of approaches have been proposed to model and animate realistic facial expressions in three dimensions[1]. Waters[2] reported a new muscle model approach to facial animation. This approach allowed a variety of facial expressions to be created by controlling the underlying musculature of the face. The development of optical range scanners, such as the Cyberware optical laser scanner[3], provides a new wealth of data for facial animation. A radically different approach is performance-based animation, in which measurements with real ac-

* **Chien-feng Huang** was born in 1976. He is a graduated student of Master degree of Department of Computer Science and Information Engineering, Taiwan University. He received a BS degree in computer science from Taiwan University in 1998, and got the Master degree from Taiwan University in 2000. His research interests are computer graphics, facial motion tracking, and synthetic face modeling. **I-chen Lin** was born in 1976. He is a Ph.D. candidate at Department of Computer Science and Information Engineering, Taiwan University. He received a BS degree in computer science from Taiwan University in 1998. His research interests are virtual reality, synthetic talking head, and facial animation. **Ming Ouhyoung** was born in 1959. He is a professor of the Department of Computer Science and Information Engineering, Taiwan University. He received a Ph.D. degree in computer science from University of North Carolina at Chapel Hill. His current research areas include computer graphics, virtual reality, and graphical user interface.

Manuscript received 2000-03-16, accepted 2000-04-26.

tors are used to derive synthetic characters[4~6]. Recently, Pighin *et al.*[7] presented new techniques for creating photo-realistic textured 3D facial model from photographs, and for creating smooth transitions between different facial expressions to make realistic facial animation. Guenter *et al.*[8] created a system to capture three-dimensional geometry, color, and shading information for human facial expressions, and used these data to reconstruct photo-realistic 3D facial animation.

For image-based animation, one of the most effective approaches to date is to use 2D morphing between photographic images[9]. However, it requires users to specify dozens of correspondences in almost every frame. In addition to 2D morphing, 2D warping techniques provide an alternative way, and one example is our previous work VR-Talk[10,11], a speech-synchronized animation system. To simulate a 3D rotation, "view morphing" technique[12] is developed to provide theoretically correct image interpolation. However, 2D morphing still suffers from arbitrary 3D rotations.

The rest of this paper is organized as the following. In Section 2, the proposed global motion compensation method is first introduced, and the concept and implementation of a two and half dimension model is illustrated in Section 3. Some issues of facial animation are presented in Section 4. Results and experiments are provided in Section 5, and Section 6 offers directions for future research.

1 Motion Capture Data

3D facial motion is captured at our industrial collaborator, Digimax Production Center, where a VICON 8 motion capture system is used. 8 cameras are set up, and 23 optical markers are attached on a performer's face (Fig. 1). The VICON 8 system captures the performer's facial expressions at 60 frames/second when he's making exaggerated expressions. After the process of feature extraction and 3D reconstruction, the output file with C3D data format contains the 3D positions of 23 features for each frame.

(a) (b)

Fig. 1 (a) The 23 connected to the pivot point captured optical markers. The low darker point in (a) is the rotation pivot, and the upper three points in (a) are used for global motion estimation. (b) Illustrates the placements of the 23 optical markers on one's face for facial motion capture, where the three points forming a rigid white triangle on forehead are identical to the upper three points in (a).

The C3D data file format[13] was developed by Andrew Dainis in 1987 as a convenient and efficient means for storing 3D coordinate data and analog data obtained from motion capture systems. The format is now widely used and becomes a de facto 3D biomechanics data standard. As a result, one could easily compare the results with other systems. In order to support unprocessed raw coordinate sample data, all the parameters and data are stored in binary form. The most common C3D file consists of three sections: a single head record, one or more parameter records, and one or more data records that contain 3D data.

Since the origin of world coordinate is determined when we calibrate the motion capture system, we should change this coordinate such that it is conformed to our environment. The location of rotation pivot is approximately estimated from the positions of features based on domain knowledge. The 3D data are then translated to local coordinate system relative to the estimated pivot. Thus, when the preprocessing is complete, the results become intuitive when replaying these 3D points frame by frame.

2 Global Motion Compensation

The retrieved 3D coordinates of marker points attached on the performer's face fully recorded the facial actions. We are interested in the mouth movement during speaking and facial expressions. Unfortunately, the global motion, such as head rotation and translation, also moves the positions of feature points. It is not a reasonable requirement to ask the performer to fix his head when he acts. Hence, the first problem we have to solve is to compensate for the global motion and then the remaining offsets can be applied to drive the facial animation.

2.1 Algorithm for motion estimation

In this case, the problem lies in "3D-to-3D feature correspondences"[14]. Suppose the features (f_i, f'_i) are 3D coordinates of points on the surface of a rigid body, observed at time t_1 and time t_2. Give N corresponding points (f_i, f'), which obey the relationship of Eq. (1).

$$p'_i = RP_i + t,\ i = 1, \ldots, N. \tag{1}$$

It is well known[15] that three non-collinear point correspondences are necessary and sufficient to determine R and t uniquely. With three point correspondences, we will get nine non-linear equations while there are six unknown motion parameters. Because the 3D points obtained from motion capture system are accurate, linear algorithm is good enough for this application, instead of iterative algorithms or methods based on the least square procedure. The improved method based on translation invariants[16] is adopted to solve the motion estimation problem.

If two points on the rigid body, p_i and p_{i+1}, which undergo the same transformation, move to p'_i and p'_{i+1} respectively, then $p'_i = Rp_i + t$ and $p'_{i+1} = Rp_{i+1} + t$. Subtraction eliminates the translation t, and using the rigidity constraint yields

$$\frac{p'_{i+1} - p'_i}{|p'_{i+1} - p'_i|} = R\,\frac{P_{i+1} - p_i}{|p_{i+1} - p_i|}. \tag{2}$$

Define the above equation as $\hat{m}'_i = R\hat{m}_i, 1 \leqslant i \leqslant 3$, where $\hat{m}$ is a unit vector. If the rigid body undergoes a pure translation, these $\hat{m}$ parameters do not change, which means translation invariants.

After rearranging these three equations, we can solve a 3×3 linear system to get R, and afterward obtain t by substitution into Eq. (1). In order for a unique solution, the 3×3 matrix of unit $\hat{m}$-vectors must be of full rank, that is the three $\hat{m}$-vectors must not be coplanar. As a result, four point correspondences are needed instead of three points, the minimal requirement. To overcome the problem of supplying the linear method with an extra point correspondence, a "pseudo-correspondence" can be artificially constructed due to the property of rigidity. In our case, the problem is resolved by finding a third $\hat{m}$-vector orthogonal to the other two. Via this improvement, the system is of lower dimension, only three point correspondences are required, and it helps to reduce the singularity problem of a matrix.

The first row of the R matrix can be determined in component form by

$$\begin{bmatrix} m_{1x} & m_{1y} & m_{1z} \\ m_{2x} & m_{2y} & m_{2z} \\ m_{3x} & m_{3y} & m_{3z} \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} = \begin{bmatrix} m'_{1x} \\ m'_{2x} \\ m'_{3x} \end{bmatrix}, \tag{3}$$

while the other two rows of R can be determined by replacing the right hand side by the y and z components of the $\hat{m}$-vectors. The third vector can be achieved by setting $\hat{m}_3=\hat{m}_1\times\hat{m}_2$ and $\hat{m}'_3=\hat{m}'_1\times\hat{m}'_2$. These artificial vectors are generated to span the three dimension spaces.

2.2 Rotation pivot estimation

In the above algorithm, we assume the rotation pivot is at the origin of world coordinate. Although the position of pivot point does not influence the result of estimated rotation matrix R, it changes the translation vector t slightly. Let O be the rotation pivot, then $p'-O=R\cdot(p-O)+t$. After rearranging this equation, we get:

$$t=p'-R\cdot p+O(R-I). \tag{4}$$

Since the global rotation angle is not large, which implies the rotation matrix is close to the identity matrix, and the pivot position is not far from the world coordinate origin due to the preprocessing based on domain knowledge, the translation vector can be assumed invariant to the position of pivot point. Here we propose an algorithm to estimate the real position of rotation pivot to obtain better 3D data compensated global motion. In Fig. 2, P_1, P_2 and P_3 are points on a rigid body, l_i is the edge length between P_i and pivot point and θ_i is the angle between two vectors OP_i and OP_{i+1}. When the rigid body undergoes a transformation about the pivot, l_i and θ_i are invariant due to the property of rigidity.

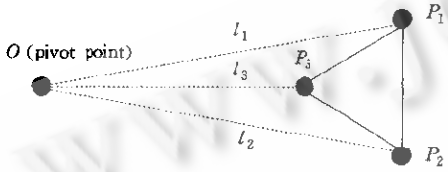


Fig. 2 The relation between rigid body and rotation

In order to estimate the position of pivot point, we need to develop an objective function that measures if the points on rigid body obey the above-mentioned properties. The objective function consists of three metrics as follows:

$$E_k(p^k)=w_1\cdot\Big(\sum_i\sum_j\|\theta_j^0-\theta_j^i\|*l\Big)+w_2\Big(\sum_i\sum_j\|l_j^0-l_j^i\|\Big)+w_3\Big(\frac{\|p^k\|}{C}\Big), \tag{5}$$

where p^k is the candidate pivot at iteration k, θ_j^i is the jth angle on frame i and l_j^i is the jth edge length on frame i. C and l are two parameters which must be adjusted depending on the 3D data themselves and coefficients w_i's are weighting factors, $1\leqslant i\leqslant 4$. It is obvious that the first two terms measure the variation of the angle and edge length, and these terms should be zero if the pivot point is at correct position because of the constraint of rigid body motion. The third metric represents a pulling force to impose restriction on pivot point not far away from the origin. Otherwise, one of the components of p^k will diverge, because the longer the length of vectors is, the smaller the angle between two vectors is, which implies that the first metric will be almost zero. Fortunately, we have preprocessed the 3D raw data according to domain knowledge, so the assumption that the pivot point is close to origin is reasonable. In other words, we only search the neighborhood of the origin to get a more reliable pivot, instead of searching in the whole three-dimension space.

To minimize the objective function $E_k(p^k)$ with unknown parameters $p(x,y,z)$, we follow the concept of the gradient descent algorithms. It is obvious that our objective function is non-differentiable, and hence gradient descent algorithms cannot be directly applied.

Given the real value function $f:\Re^n\to\Re$, the gradient descent algorithm finds the minimal value of $f(x)$ by iteratively updating x at iteration $k+1$,

$$x^{k+1}=x^k-\alpha_k\nabla f(x^k)\ \alpha_k>0. \tag{6}$$

The coefficient α_k is a positive scalar called step size. The step size α_k is a free variable that should be carefully chosen. If the step size is too small, the minimization progress will be too slow. Therefore, we employ a bold-driver method to adaptively adjust the value of α_k. Hence, Eq. (6) is rewritten as below:

$$x^{k+1}=x^k-a_k\nabla f(x^k)$$
$$a_{k+1}=\begin{cases}\rho a_k & \rho\geqslant 1, \quad \text{if } f(x^{k+1})<f(x^k)\\ \sigma a_k & 0<\sigma<1, \quad \text{if } f(x^{k+1})\geqslant f(x^k)\end{cases} \tag{7}$$
$$a_c>0.$$

Because our objective function $E_k(p^k)$ is non-differentiable, we cannot obtain the gradient ∇E, so we approximate the gradient descent algorithm by evaluating the objective function in six possible directions, and choose the set of parameters p^k that cause the minimal value of $E_k(p^k)$ at iteration k. The six directions to be evaluated are defined by

$$\nabla_1=\begin{bmatrix}+1\\0\\0\end{bmatrix},\nabla_2=\begin{bmatrix}-1\\0\\0\end{bmatrix},\nabla_3=\begin{bmatrix}0\\+1\\0\end{bmatrix},\nabla_4=\begin{bmatrix}0\\-1\\0\end{bmatrix},\nabla_5=\begin{bmatrix}0\\0\\+1\end{bmatrix},\nabla_6=\begin{bmatrix}0\\0\\-1\end{bmatrix}. \tag{8}$$

The iteration stops when either the maximum number N of iterations is reached or the value of $E_k(p^k)$ is smaller than a predefined threshold ε.

2.3 Motion compensation

The above two algorithms depend on the results of each other; in other words, for global motion estimation, it needs the position of rotation pivot, and for rotation pivot estimation, the translation offsets for each frame are required. Hence, we run these two procedures alternately to update the unknown parameters until these unknowns converge. Once the rotation matrix and translation vector for each frame are determined, the inverse of these affine transformations can be applied directly following Eq. (5) on feature points to get the new position without global motion. Thus, three points on the rigid body will be almost stationary, while the movements of other features represent the actions when the performer talks and emotes.

$$p_i=R^{-1}(p'_i+t),\ i=1,\ldots,N. \tag{9}$$

3 Head Modeling

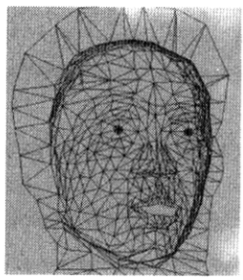
Photo-realistic rendering is our goal. 2D mesh and warping technique were employed to a single face image in VR-Talk[10,11], which was our previous speech driven talking head system. When developing a system based on 3D model, to fast construct one's head model is not very easy; besides, we can't overcome the problem of hair rendering in real-time. Thus, we adopt a two and half dimension head model[19], which consists of front side view head image patches and a half-cut 3D model (Fig. 3). The major advantage of this model is to combine nice features from 2D mesh and 3D model: simple, vivid, and natural when small-scale rotation is applied.

3.1 Head model fitting

First, a frontal and neutral face image is needed. In order to fit the generic 3D head model, a user must specify about 30 feature points, such as eyes, nose and mouse boundary. Some predefined control points on the 3D head model are adjusted to proper position, and the texture data are obtained by orthogonal projection. As shown in Fig. 4, an editor tool is also developed to help the user edit his model in 3 minutes.

Because the 3D head mask contains half the frontal half head model, additional image patches with texture of the neutral face image, including human hair, are used to cover the other part. It looks real if the rotation angle is constrained to less than 30 degrees.

(a)



(b)

Fig. 3 (a) The 2-1/2D face model (b) The wire-frame model with hair in the plate at the back

Fig. 4 Specify feature points on an image for adjusting a generic 3D head model

3.2 Other features of the head model

Eye blinking and head motion are common actions when a person is speaking. In order to make the talking head more realistic, the above functions should be implemented. To keep head moving, a sequence of transformation parameters based on domain knowledge is applied to the head model. In this way, we can simulate a person's action, such as nodding, head shaking, naturally. Generally, a person's eyes blink once every four seconds. The method used is similar to our previous system, the Image Talk[2], dragging the control points at upper eyeshade down.

Teeth usually are ignored in many talking head systems except sample-based method because of the lack of teeth information. We propose a generic teeth model to simulate one's teeth inside the mouse. The teeth model can be separated into two parts: the upper teeth and the lower ones. Both of these teeth models are composed of some rectangle patches connected in the shape of a hoof. In order to change teeth texture easily by users, we use rectangle patches connected in the shape of a hoof instead of constructing a precise teeth model. Besides, the alpha blending approach is applied to hide the non-teeth part image in patches.

4 Facial Animation

To apply the 3D motion data to a head model, firstly, we have to modulate the data to fit the facial features of the model. For feature points on the upper part of the face, the motion data are scaled according to the distance between two lower eyelids, the distance between the forehead and nose tip; for points on the lower part of face, the data are scaled in proportion to the mouth width and the distance between nose tip and the chin; and the scale value in z-axis direction is determined by the distance between the cheek and the lips.

How to interpolate the unmarked vertices and the constraints for generating the facial animation are described in the following subsections.

4.1 Scatter data interpolation

Once applying 3D motion data to the two and half dimension model described previously, we can deform the feature points on the face mesh directly. But we need to construct a smooth interpolation function that gives the 3D displacements between the original point positions and the new positions in the following frames for every vertex. Constructing such an interpolation function is a standard problem in scattered data modeling. Given a set of known displacements $u_i=p_i-p_i^{(0)}$ away from the original positions $p_i^{(0)}$ at every constrained vertex i, we will construct a function that finds the displacement u_j for every unconstrained vertex j. If we can find such a smooth function $f(p)$ fitted with the known data $u_i=f(p_i)$, from which we can compute $u_j=f(p_j)$.

There are several useful functions to model scattered data[6], for example, Shepard's method, volume spline, multi-quadric method, etc. In different applications, various considerations should be taken to select a method for modeling scattered 3D data with minimum error. In our case, a method based on radial basis functions is adopted, that is, functions of the form

$$f(p)=\sum_i C_i\varphi(\|p-p_i\|)+Mp+t, \tag{10}$$

where $\varphi(r)$ is radial symmetric basis functions. p_i is the constrained vertex; low-order polynomial terms M, t are added as affine basis. Many kinds of function for $\varphi(x)$ have been proposed[17]. We have chosen to use $\varphi(r)=e^{-r/64}$.

To determine the unknown coefficients c_i and the affine components M and t, we must solve a set of linear equations that includes $u_i=f(p_i)$, the constraints $\sum_i c_i=0$ and $\sum_i c_i p_i^T=0$. In general, if there are n feature point correspondences, we will have $n+4$ unknowns and $n+4$ equations with the following form:

$$\begin{bmatrix} \cdot & \cdot & \cdots & & P_1 & 1 \\ \cdot & e^{-\|p_i-p_j\|/64} & \cdots & & P_2 & 1 \\ \vdots & \vdots & \cdot & \vdots & \vdots & \vdots \\ \cdot & \cdot & \cdots & \cdot & P_n & 1 \\ 1 & 1 & \cdots & 1 & 0 & 0 \\ P_1^T & P_2^T & \cdots & P_n^T & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{11}$$

where $1\leqslant i,j\leqslant 3$, $P_i=(x_i,y_i,z_i)$.

4.2 Face regions and force constraints

Since there are only 23 markers with our current captured data, and the facial actions of human beings are so subtle, some constraints must be applied to generate reasonable and smooth animation. In this system, we divide the head model into five regions: the hindbrain, the upper lip, the lower lip, the face, and the neck.

(i) The hindbrain

By the method described in Section 2, we can compensate the global head motion, and thus the hindbrain becomes stationary when compared with the feature point motion. To avoid the abnormal motions of vertices intruding into this stationary region, some static points around the hindbrain are considered as feature points in the radial basis function computation. With this approach, this kind of abnormal motions can be gradually reduced from the hindbrain.

(ii) Face

We took all the feature points including still points as feature points in the computation of facial vertices motion. This is because that the influence of a feature point decreases exponentially with the increase of the distance in our interpolation, and the effects of feature points far away are almost zero. However, taken all the vertices in the same "field of force" can help us to avoid the problem of discontinuousness at the boundary between different regions.

(iii) Upper and lower lips

Since the upper lip of human beings is controlled by muscles on the upper mouth and cheeks, the motion of vertices on the upper lip is interpolated from the feature points on the upper mouth and cheeks. Similarly, the vertices on the lower lip are interpolated from the feature points on the lower mouth and the chin. In certain drastic motion involving lips, the discontinuity may occur at corners of the mouth. Some curve, such as Bezier curves or B-spline, can be applied to smooth the boundary.

(iv) Neck

The same as the hindbrain, the connecting region of the neck and the head is stationary, and some still points are also located at the connecting region.

5 Results

For the time being, the proposed method has been implemented on Windows NT, and OpenGL is adopted as our graphics library. Face images with 1024×1024 resolution are used for face modeling and texture mapping. The system has an update rate of over 30 frames/second on a Pentium III 500MHz PC with an OpenGL acceleration card. From the animation results, we can.see that even though the captured data were obtained from another person, when applied to one of the coauthors, the visual effects are quite convincing. (Fig. 5)

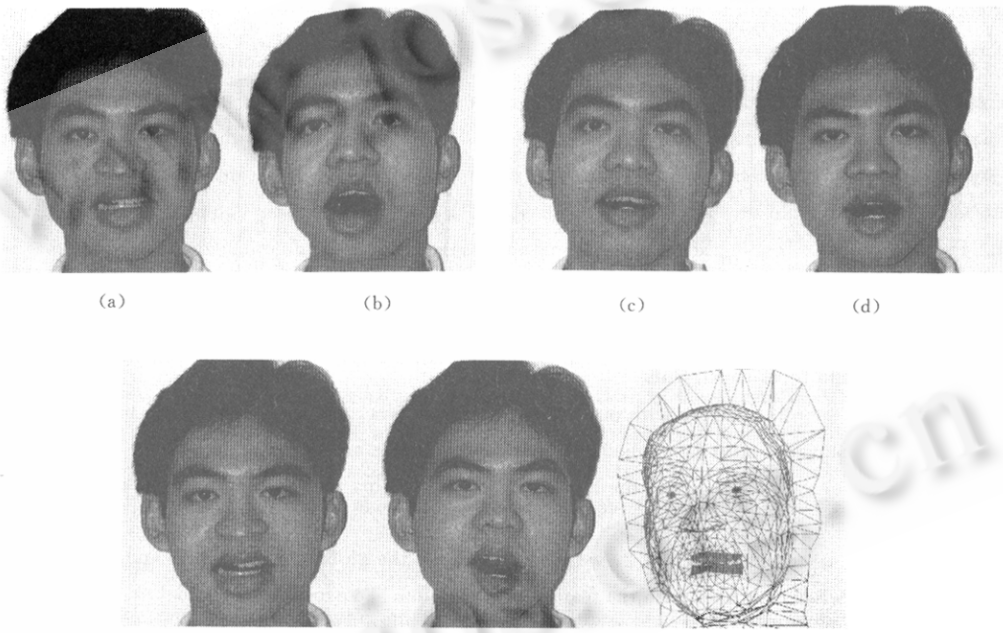


Fig. 5 From (a) to (f) are the animated snapshots with different viewing angles and mouth shapes, and (g) is the associated wire-frame model

6 Future Work

The work described in this paper is only a prototype, and the ultimate goal is to build a simple software system to drive the head model with motion capture data conveniently. Of course, the quality of the generated facial animation is of the chief consideration. There are many ways to enhance and extend the proposed method, as described below:

Using 3D head model. At this moment a two and half dimension model is used to generate photo-realistic head model, however, it is constrained with small angle rotation. If we want to make an arbitrary head rotation, a 3D head model is necessary. It is still a challenging issue in semi-automatic 3D head modeling with minimal user intervention, but it is possible in the near future due to the recent impressive results in this topic[7,18].

More feature points and regions. In this work, only 23 feature points were used. In order to capture the

detailed movement, more feature points are required. Thus, the face subdivisions and force constraints need to be re-considered carefully. Ideally, we should simulate subtle changes on the performer' face, such as eye blinking.

Improved accuracy of global motion estimation. If we want to replay human facial expression, instead of only head motion, global head motion has to be compensated. The accuracy of global head motion estimation result is the key point to determine whether the animation looks real. Other algorithms can be explored to get more reliable result, or maybe some auxiliary markers can help to determine global motion exactly.

Acknowledgments We would like to thank Digimax Production Center for providing the 3D facial motion data and technical supports in motion capture.

References

1 Parke F I, Keith Waters. Computer Facial Animation. In: Peters A K ed. Wellesley, Massachusetts, 1996

2 Waters K. A muscle model for animating three-dimensional facial expression. In: Proceedings of ACM Computer Graphics (SIGGRAPH'87). 1987,21:17~24

3 Cyberware Laboratory Inc. 4020/RGB 3D Scanner with Color Digitizer. Monterey, CA, 1990

4 Philippe Bergeron, Pierre Lachapelle. Controlling facial expressions and body movements in the computer-generated short 'Tony De Peltrie'. In: SIGGRAPH'85 Advanced Computer Animation Seminar Notes. 1985

5 Irfan Essa, Sumit Basu, Trevor Darrel *et al*. Modeling, tracking and interactive animation of faces and heads using input from video. In: Computer Animation Conference. 1996. 68~79

6 Lance Williams. Performance-driven facial animation. In: Proceedings of ACM Computer Graphics (SIGGRAPH'90). 1990,24:235~242

7 Frederic Pighin, Jamie Hecher, Dani Lischinski *et al*. Synthesizing realistic facial expressions from photographs. In: Proceedings of ACM Computer Graphics (SIGGRAPH'98). 1998. 75~84

8 Brian Guenter, Cindy Grimm, Daniel Wood *et al*. Making faces. In: SIGGRAPH'98 Conference Proceedings, ACM SIGGRAPH. 1998. 55~66

9 Thaddeus Beier, Shawn Neely. Feature-based image metamorphosis. In: SIGGRAPH'92 Conference Proceedings, ACM SIGGRAPH. 1992. 35~42

10 Perng Woei-luen, Wu Yung-kang, Ouhyou Ming. Image talk: a real time synthetic talking head using one single image with Chinese text-to-speech capability. In: Proceedings of PacificGraphics'98. Singapore, 1998. 140~148

11 Lin I-chen, Cheng-Sheng Hung, Tzong-Jer Yang, Ming Ouhyoung. A speech driven talking head based on a single face image. In: Proceedings of PacificGraphics'99. Seoul, Korea, 1999. 43~49

12 Seitz S M, Dyer C R. View morphing. In: SIGGRAPH'96 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH. 1996. 21~30

13 The C3D file format. http://www.c3d.org

14 Huang T S, Netravali Arun N. Motion and structure from feature correspondences: a review. In: Proceedings of the IEEE. 1994,82(2):252~268

15 Goldstein H. Classical Mechanics. MA: Addison Wesley, 1980

16 Blostein S D, Huang T S. Algorithms for motion estimation based on 3-D correspondences. In: Martin W, Aggrawal J K eds. Motion Understanding, Norewell, MA: Kluwer, 1988

17 Nielson G M. Scattered data modeling. In: IEEE Computer Graphics and Applications, 1993,13(1):60~70

18 Fua P, Miccio C. Animated heads from ordinary images: a least-squares approach. Computer Vision and Image Understanding, 1999,75(3):247~259

19 Ouhyoung Ming, Lin I-chen, Lee D S D. Web-enabled speech driven facial animation, (invited). In: Proceedings of ICAT'99 (International Conference on Artificia. Reality and Tele-existence). Tokyo, Japan, 1999. 23~28

将高解析度动作撷取资料校正应用在脸部动画系统

黄建峰　林奕成　欧阳明

(台湾大学资讯工程学系通讯与多媒体实验室)

摘要　提出一个新的方法来产生脸部动画,即利用动作撷取系统捕捉真人脸上的细微动作,再将动态资料用来驱动脸部模型产生动画.首先,利用 Oxford Metrics' VICON 8 系统,在真人的脸上贴了 23 个反光标记物,用以进行动作撷取.得到三维动态资料后,必须经过后续处理才能使用,因此,提出了消除头部运动的方法,并估计人头的旋转支点,经过处理后,剩余的动态资料代表脸部表情的变化,因此,可以直接运用到脸部模型.用 2.5D 的脸部模型来实作系统,这样可兼得二维模型与三维模型的优点:简单、在小角度旋转时显得生动、自然.在脸部动画的制作中,利用一个特殊的内差公式来计算非特征点的位移,并将脸部分成数个区域,用以限制模型上三维点的移动,使动画更加自然.此动画系统在 Pentium III 500MHz 的机器上,并配有 OpenGL 的加速卡,更新率可以超过每秒 30 张.

关键词　动态撷取,虚拟人物,动态补偿,脸部动画,合成的脸.

中图法分类号　TP391