Learning and Synthesizing MPEG-4 Compatible 3-D Face Animation From Video Sequence

Wen Gao, *Member, IEEE*, Yiqiang Chen, Rui Wang, Shiguang Shan, and Dalong Jiang

***Abstract*—In this paper, we present a new system that applies an example-based learning method to learn facial motion patterns from a video sequence of individual facial behavior such as lip motion and facial expressions, and using that to create vivid three-dimensional (3-D) face animation according to the definition of MPEG-4 face animation parameters. The system consists of three key modules, face tracking, pattern learning, and face animation. In face tracking, to reduce the complexity of the tracking process, a novel coarse-to-fine strategy combined with a Kalman filter is proposed for localizing key facial landmarks in each image of the video. The landmarks' sequence is normalized into a visual feature matrix and then fed to the next step of process. In pattern learning, in the pretraining stage, the parameters of the camera that took the video are requested with the training video data so the system can estimate the basic mapping from a normalized two-dimensional (2-D) visual feature matrix to the representation in 3-D MPEG-4 face animation parameter space, in assistance with the computer vision method. In the practice stage, considering that in most cases camera parameters are not provided with video data, the system uses machine learning technology to complement the incomplete 3-D information for the mapping that information is needed in face orientation presentation. The example-based learning in this system integrates several methods including clustering, HMM, and ANN to make a better conversion from a 2-D to 3-D model and better estimation of incomplete 3-D information for good mapping; this will be used to drive face animation thereafter. In face animation, the system can synthesize face animation following any type of face motion in video. Experiments show that our system produces more vivid face motion animation, compared to other early systems.**

***Index Terms*—Face animation, face tracking, machine learning, MPEG-4.**

Manuscript received November, 2001; revised Juen 2003. This work was supported in part by the National Fundamental Research and Development Program (973) of China (2001CCA03300), the Natural Science Foundation of China under Grant 69789301, and the National Hi-Tech Program of China under Contract 2001AA114190.

W. Gao is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, and also with the Graduate School, Chinese Academy of Sciences, and the Department of Computer Science, Harbin Institute of Technology, 150001 Harbin, China (e-mail: wgao@ict.ac.cn).

Y. Chen is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, and also with the Graduate School, Chinese Academy of Sciences, Beijing 100039, China (e-mail: yqchen@ict.ac.cn).

R. Wang is with the Department of Computer Science, Harbin Institute of Technology, 150001 Harbin, China (e-mail: rwang@jdl.ac.cn).

S. Shan is with the ICT-YCNC FRJDL, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, and also with the Graduate School, Chinese Academy of Sciences, Beijing 100039, China (e-mail: sgshan@ict.ac.cn).

D. Jiang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, and also with the Graduate School, Chinese Academy of Sciences, Beijing 100039, China (e-mail: dljiang@ict.ac.cn).

Digital Object Identifier 10.1109/TCSVT.2003.817629

I. Introduction

GIVEN a video sequence of an individual face behavior, such as lip motion, eye motion, facial expression, head motion and so on, it is a great challenge to create vivid his/her graphics face animation in real time. For achieving this task, we need to answer many questions, such as how to design the training set which can cover all pairs of language piece and lip motion, pairs of meaningful facial emotion and facial-organ motion, how many times the individual should repeat the training set for occultation of incidentally inconstant playing, how many individuals should be invited for making a stable training data set, what kind of approaches should be used to obtain the model of facial motion, how to drive a three-dimensional (3-D) facial motion animation using obtained two-dimensional (2-D) tracking facial landmarks, and so on. In this paper, we explain what we are doing in our system design and implementation and try to answer some of the above questions, especially how to synthesize vivid 3-D MPEG-4 facial animation.

A great deal of research has already been undertaken with the above purpose in mind. For tracking facial landmarks, adding markers on the human face was considered in many early works, such as heavy makeup [1] or using a set of colored dots [2], [3]. Once the position of the marker is determined, the facial motion features can be easily derived from image or video sequence. Williams [3] used it to get 2-D facial features from image. Guenter *et al.* [2] used it to track 3-D facial features from video. However, this technique is only suitable for cases in which markers on the face existed or are allowed to be used.

In a normal case, we need to process face tracking in an unmarked face of video. Generally speaking, face tracking is a two-stage process. First, the tracking routine must search any possible location and size of a candidate face in each image of video; the classics techniques for doing this include deformable template, rigid patches [4]–[6], and edge/feature detectors [7]–[9]. Second, after feature points of the possible face located, the tracking routine solves the problem of finding exact face(s). Global constraints on the facial motion itself, such as partial rigidity and prediction schemes [10]–[12] (e.g., Kalman filtering), can also be used to improve tracking accuracy. Some of the most frequently used solutions group the pixels surrounding the tracked point into a single feature signature. Examples are found in the vast literature and extensive studies on the sum of square difference (SSD) methods [13] and Gabor filters [14]. The above approaches are relatively efficient, but the accuracy of modeling results is not satisfactory due to incomplete 3-D information and complex environment caused by variations of facial expression, orientation, and illumination condition. For solving this issue, one can obtain 3-D information with the

1051-8215/03$17.00 © 2003 IEEE

assistance of computer vision-related methods, for example, some approaches based on either shaping from edge or optical flow. Essa and Pentland [15] built a physically based face model and developed a control theoretic technique to fit it to a video sequence. Decarlo and Metaxas [16] allow for simultaneous estimation of 3-D shape and motion based on the integration from optical flow, edges, and other shape constraints. Though those methods can directly obtain the 3-D information, it only performs well under limited circumstances. Another approach is to use a linear class of objects; Vetter and Blanz [17] used a linear class of both images and 3-D geometrics for image matching and face modeling. It can eliminate most of the nonnatural faces and significantly reduces the search space.

For face orientation or posture measurement and reconstruction, many contributions have being made in recent years. Brand [18] used the Parametric Hidden Markov Model to learn motion patterns from a highly varied set of motion capture sequences. The learned model can synthesize novel motion in any interpolation or extrapolation of style. Rosales presented a novel approach for estimating human body posture and motion from video sequence using statistical technology and neural network [19], [20].

In our system, we select the model of high-level face description defined in MPEG-4, and we adopted facial definition parameters (FDPs) and facial animation parameters (FAPs) to describe facial behavior. Even though the simple combination of the above technology could be achieved with good improvement, reliable performance, in the general sense, does not really apply to our case. Our aim is at synthesizing vivid MPEG-4 compatible 3-D face animation using an individual facial motion video. For this purpose, the challenge is to bridge any face action, measured from given 2-D video sequence, to typical 3-D animation, which is constructed in a high-resolution presentation according to MPEG-4 face animation parameter definition. This bridge is known as the mapping from 2-D low-resolution facial landmark to 3-D high-resolution graphics presentation. After face tracking in a 2-D image sequence, the obtained parameter set cannot be directly used to drive a 3-D MPEG-4 animation, because of the gap of representation between the 2-D image and 3-D graphics. For 3-D model-based fitting technology, the learning result is often coarse if the training data is not large enough, and sometime the motion data cannot be accurately obtained in a low-texture facial area. For correspondence solving technology, a lot of points (pixels) can be tracked and obtained with high accuracy, but it is too hard to map these low-level features onto the high-level face model.

In our implementation, the tracking feature is divided into two types. The first type is the class that can be acquired using our coarse-to-fine tracking approach, such as the mouth corner, the eye corner, etc; we named this class as low-resolution visual feature. The second type is the one that cannot be acquired accurately using the 2-D landmark tracking, such as the feature points on the cheek, etc. We obtain the low-resolution type feature points automatically by our tracking system and automatically extract the high-resolution type with some color dots on the face. Then we can acquire knowledge about the relationship between the first type of feature points and the second type. Some unsupervised clustering method and statistical technologies are used to obtain the high-resolution 3-D visual feature configuration sets and the dynamic model, which imposes additional structure on these cluster sets by specifying which cluster trajectories are possible. Also, a neural network was trained to build mapping from low-resolution visual feature representation to high-resolution 3-D FAP. When the training is finished, we track the human face without any marker mounted on the face. Normally we can get the first type of feature points easily at this time, but not for the second type. With the learning result, we can obtain the 3-D FAP from this tracking result directly, and then drive a 3-D synthetic face animation.

Compared with others, the advantages of our system include a real-time accurate tracking system for the low-resolution feature points, MPEG-4 compatible, example-driven 3-D facial-motion synthesis based on machine-learning technology.

The organization of this paper is as follows. We begin with the automatic facial feature extraction, then in Section III we present a learning algorithm for mapping of 2-D low-resolution visual features to high-resolution 3-D FAP, and in Section IV we describe the synthesis of a 3-D virtual face from a given example. Finally, the video-driven MPEG-4 based face animation system is given, followed by the conclusion.

II. Face Definition and Animation in MPEG-4

In the MPEG-4 standard, there are 84 feature points used to describe a face for both FAP and FDP [21]. FDP gives the definition of face shape, size, and texture, and FAP gives the parameter description of facial deformation and expressions. Using these parameters, one can generate any possible facial expressions according to the specified FAP values. The animation parameters are well designed in order to allow a satisfied implementation on any facial model ranging from video-realistic image warps to 3-D cartoon characters. Sections II-A and -B will show how FAP and FDP work.

A. FDP Set

One must have a generic facial model capable of interpreting FAPs. This insures that it can reproduce facial expressions and speech pronunciation for lip motion. MPEG-4 allows the encoder to completely specify the face model that the decoder has to animate. The FDPs can be used to personalize the generic face model by modifying the shape and appearance of the face to make it look like a particular person/character. The FDP fields include FeaturePointsCoord that specifies feature points for the calibration of the proprietary face, TextureCoords that specifies texture coordinates for the feature points, TextureType that contains a hint to the decoder on the type of texture image, FaceDefTables that describes the behavior of FAPs for the face in the FaceSceneGraph, and FaceSceneGraph that can be used as a container of the texture image or the grouping node for the face model rendered in the compositor (therefore it has to contain the face model). Fig. 1 shows the feature points defined in FDP and FAP.

B. FAP Set

There are 68 FAPs categorized into 10 groups related to the parts of the face. FAPs manipulate key feature control points on

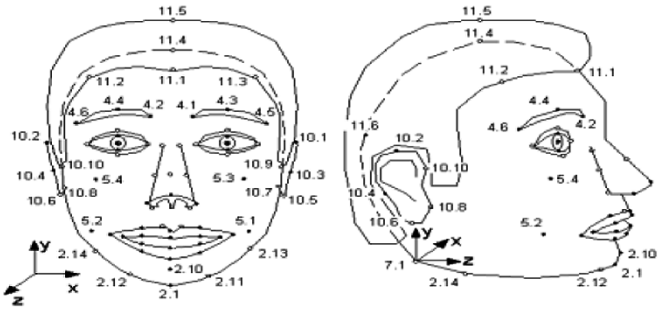


Fig. 1. The feature points definition for FDP and FAP.

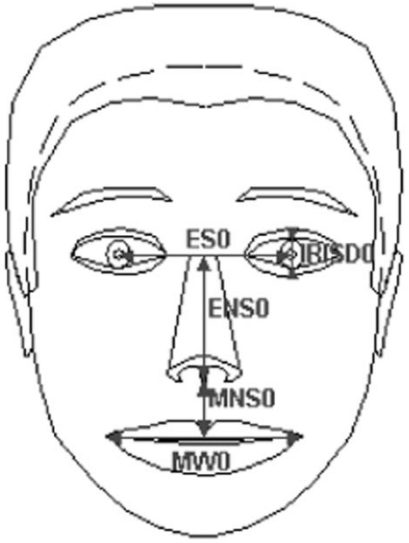


Fig. 2. FAPUs.

a mesh model of the face to represent a complete set of basic facial actions, including head motion, tongue, eye, and mouth control, and visemes (visual counterparts to phonemes). Since the FAPs are required to animate the face in different sizes and proportions, all parameters of FAP involving transnational movement are expressed in terms of FAP units (FAPUs). These correspond to fractions of distances between some essential facial features, e.g., eye distance (see Fig. 2). The fractional units are chosen to maintain sufficient accuracy.

C. Two- and Three-Dimensional Feature Representation

To make the synthesizing face more realistic, the 3-D visual object should be reconstructed from the parameters of a 2-D model that are obtained from face tracking, as we discussed above. For 2-D feature representation, we calculate the relative motion displacement between the location of each tracked face landmark (120 points) and one standard static face landmark and then represent them as 2-D feature vectors of 120-dimension representation. We also normalized it into a consistent representation to avoid the variation of individual faces. By doing this process on the given video, we can obtain a 2-D feature vector of 120-dimension sequence with each vector representing the locating result in each image. For 3-D visual features, the FAPs were adapted to represent the 3-D feature vector. That is to say, for every 2-D feature vector, we can analysis and calculate the corresponding FAP sets. In this paper, we calculate 51 FAPs from all 66 FAPs defined in MPEG-4 for each 2-D feature vector. These 51 FAPs include the FAPs that control the motion of the head, lip, jaw, eyes, eyebrows, and cheeks and not include those FAPs which control distortion of the ear and nose.

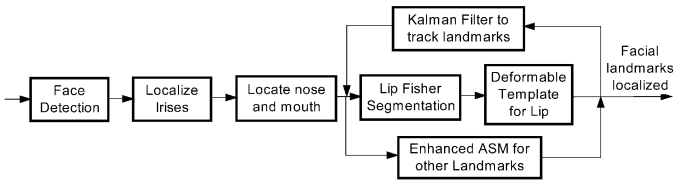


Fig. 3. Framework of automatic facial feature extraction and tracking from a video.

III. Automatic Facial Landmarks Localization and Tracking

Both the pattern learning stage and video-driven face animation stage need the accurate facial landmarks obtained from the face tracking stage. In this section, we describe our coarse-to-fine strategy, as well as the corresponding tracking method based on the Kalman filter.

The proposed framework for automatic facial feature extraction and tracking is shown in Fig. 3. In the framework, salient facial features, such as irises and the nose and mouth region are first located approximately. Since the lip region is more important for face animation, additional process are further conducted to extract lip contour. The lip region of interest (ROI) is segmented by adaptively choosing a threshold after enhancing the lip region by a Fisher transform to distinguish skin color and lip color. After that, a deformable template is further adopted to refine the location of the lip contour.

Then, an enhanced active shape model (ASM) [22] is further employed to extract other facial landmarks on the eyes, nose, and face contour, based on the coarse location of the salient facial organs, which can significantly prevent the ASM from trapping into a local minimum because of the extracted initial landmark positions.

To speed up localization processing of the landmarks in a video, a Kalman filter has been incorporated into the system for prediction of the landmark appearance. Notice that, for simplifying the complexity of processing in our current system, faces in the video are constrained nearly frontal. However, essentially our strategy can be extended to multiposed face landmarks localization and tracking.

A. Coarse Localization of Salient Facial Organs

Face detection has been a problem well solved in the past few years. In this paper, based on the observations that the two irises are the most salient features, they are localized first, based on the result of our Gravity-Center Template-based face detection system [23]. Then other organs are located by integral projection.

1) Localization of Two Irises: In our previous work, we have proposed an eye detection method based on a region-growing searching [24]. The method is based on the characteristic of the region between the eye and the nose. The method searches the iris center from a start rectangle, which is near the nose. Because

both iris centers are closer to the nose than the eyebrows and sideburns are, the method always reaches the iris centers before it reaches these confusable features. Furthermore, when the head rotates, the position relationship between these features will not change. If the face detection method used can give good estimation on the face rotation angles, our method will work under extremely large rotations both in depth and on the plane as long as both eyes can be seen. If the rotation angles are unknown, the proposed method using a regular search rectangle can still successfully localize the iris centers in faces, which rotate to a comparable large extent. Because the method is not based on the contour of the eye, the quality of the image may be low.

2) Feature Localization Based on Integral Projection: Based on the observations that the nostrils and the lip form two gray valleys below the midpoint of the two irises, horizontal integral projection is adopted to localize the two organs. First, the approximate region range is estimated by facial configuration *a priori*, in which the horizontal integral projection curve is calculated by sliding a window vertically. A local minimum is expected to appear at the point corresponding to the location of the mouth. The same strategy as used in the mouth detection is adopted to extract the nose features. Horizontal integral projection is first used to obtain the location of the nose. Then the position of the two nostrils is calculated by vertical integral projection. The nose tip is located by searching for a high luminance point above the nostrils. The key feature points include the eye corners, mouth corners, and nose, according to which template parameters can be well initialized. The corners are located by integral projection in the edge map.

B. Mouth Feature Extraction

1) Fisher Discriminate for Lip and Skin Color: Fisher's discriminate analysis [25] provides an approach to distinguishing lip color from skin color. Its basic idea is to transform the original data to maximize the separability of the two classes. The RGB space is transformed into YIQ space. Two components Q and phase angle FI are set to a vector x which is used in Fisher transform to distinguish the skin color and lip color. The lip colors are enhanced and the skin and lip colors are separated after the images that are transformed by the Fisher transform.

2) Lip Segmentation: Because the relative area of the lip in relation to that of a face region is almost invariant to a specific person, this characteristic is used to adaptively set the threshold to distinguish the lip color from the skin color. Note that the lip area is enhanced after Fisher transform, i.e., the lip area is the brightest area in a face. The threshold to separate the lip area and skin area can be determined according to the above observation. The threshold is used to binarize the enhanced face image. The ratio of lip area to skin area is within 4%-7%, according to our statistic. Although the ratio is different for different people, its change range is small. The ratio value is set to 5% in our system, andthe corresponding threshold is decided according to statistics. The lip area can be detected for a face under poorer illuminant condition and lip states. The detecting accuracy of ROIs is 100%. This approach to setting threshold has very good adaptively for different environments. This approach not only can increase the accuracy of locating lips but is also fast enough to achieve real-time implementation. The outer lips are easily found using this approach, including the key points of outer lips such as the top point of the upper outer lip and the lowest point of the outer lower lip, as shown in Fig. 4.

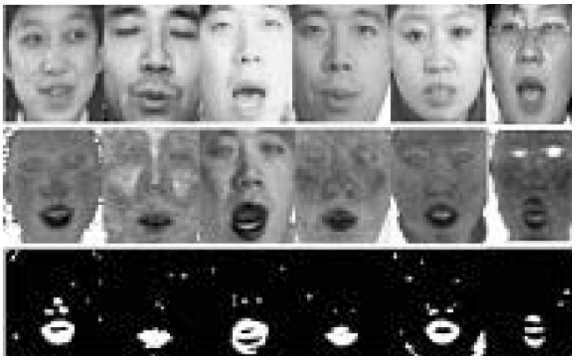


Fig. 4. The ROI detection results of different people under different illuminating conditions.

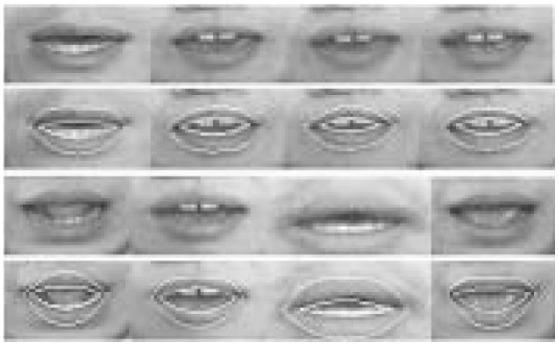


Fig. 5. The results obtained by fit template.

3) Deformable Template Fit: The outer lip contours become clear after the Fisher transform and the initial position of the template is obtained in key points by the adaptive threshold setting approach. Therefore, the outer lip contours are easily matched by using the deformable template approach as shown in Fig. 5.

C. Enhanced ASM for Further Localization of More Facial Landmarks

ASMs and AAMs are both based on statistical models, which are demonstrated to be efficient and effective for image interpretation. In ASMs, local texture on the direction perpendicular to the contour, i.e., the so-called profile, is exploited to model the local texture of each landmark and search for the landmarks locally. The global shape models are then applied to "correct" the local search result according to the statistical shape model. Obviously, in the ASMs, only local texture is used, while the abundant global texture is not utilized to constrain the shape extraction. However, in AAMs, global statistical shape and texture constraints are combined to build an appearance model. In addition, a linear prediction model is constructed to predict and update the parameters of the appearance models for optimization. Therefore, not only the shape but also the texture of the target object can be matched by the analysis-by-synthesis paradigm [26].

We have made some improvement to ASMs for face feature analysis. First, salient features such as the eyes localized pre-

TABLE I
PERFORMANCE COMPARISON FOR FACE ANALYSIS

Method	E (pixel)	Variance of E (pixel)
ASMs	3.05	3.23
Our ASM	2.59	2.71

viously are utilized to initialize the shape model and provide region constraints on the subsequent iterative shape searching. We also exploit the edge information to construct better local texture models for the landmarks on the face contour, which is based on the observation that the landmarks on the face contour usually locate at a strong edge. Our experiments have proved the effectiveness of our method [22].

Performance evaluation is really an important part for facial feature localization. We have proposed to evaluate the performance by using average error, which is defined as the following distance between the manually labeled shape and the resulting shape of our ASM:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{n}\sum_{j=1}^{n}\mathrm{dist}(P_{ij}, P'_{ij})\right]. \tag{1}$$

where N is the total number of the probe images, n is the number of the landmark points in the shape (for our case, $n = 103$), P_{ij} is the jth landmark point in the manually labeled shape of the ith test image manually labeled, and P'_{ij} is the jth landmark point in the resulting shape of ASM for the ith test image. The function $\mathrm{dist}\,(P_1, P_2)$ is the Euclidean distance between the two points. To evaluate our method, experiments are conducted on the 300-face image database. In order to evaluate the performance more accurately and sufficiently, the leave-one-out strategy is adopted. Table I shows the comparison of our methods with the original ASMs, and some improvement can be observed [22]. Note that the distance between the two eyes of face images in the database are about 60 pixels, from which readers can imagine the error level of our methods.

D. Kalman Filter for Tracking Facial Landmarks

To speed up the facial landmarks searching procedure, the Kalman filter is further exploited to predict the landmarks in the next frame of the video. It is too time-consuming to predict all the positions of the facial landmarks by using a Kalman filter, therefore, we reduce the dimensionality of the state subspace by principal component analysis (PCA) and predict the landmarks in the PCA subspace. It processes a 100-frame video in our experiments. Fig. 6 shows the examples for comparison of the coordinates of tracked landmarks and the manual labeled feature point.

In our system, the performance is at a rate of 10 frames/s and the image size is 720×576. The mean error square for all the frames is 6.250 00, and, after normalization, 0.693 369. Testing results show that our method can maintain an accurate track to about one-half centimeter accuracy (in the image plane) for long sequences (many hundreds of frames) without drift or lost tracking.

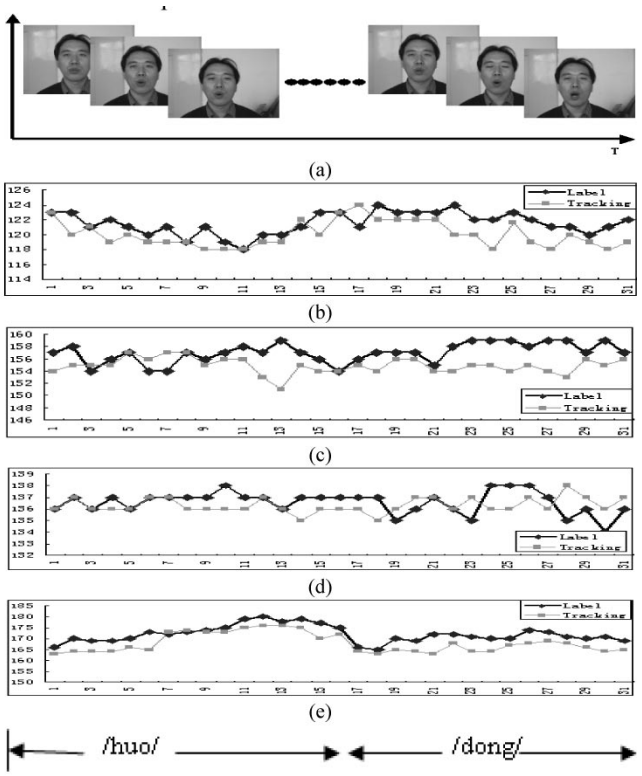


Fig. 6. Kalman filter tracking precision, (a) The video for pronunciation Chinese "Huo Dong." (b) The comparison of the location of left lip corner peak on the X axis (c) The comparison of the location of left lip corner peak on the Y axis (d) The comparison of the location of low lip peak on the X axis (e) The comparison of the location of low lip peak on the Y axis.

E. Three-Dimensional Facial Feature Representation and Recovery

For 3-D high-resolution visual feature recovery from a given video sequence, one classic way we can solve the problem is by computer vision technology, in the following steps: calibrating the camera that took video, getting the camera's intrinsic parameters, tracking the motion of the human face in a video sequence, extracting the shape of the face or of facial component in each image of video, and computing face motion and scene in 3-D space from the corresponding segments in the video sequence. In the last step, the epipolar geometry, by which the geometric constraint on the image pair of one rigid object is considered, is employed to recover the motion and structure of a scene. When given a camera's intrinsic parameters, the geometric constraint can be expressed by the equation $P_1 E P_2 = 0$, where $E = [t]_x R$, $P_1(x_1, y_1, 1)^T$, and $P_2(x_2, y_2, 1)^T$ are corresponding point coordinates between two images, E is the essential matrix, and t and R are the relative translation and rotation between two images. Rewrite the formula to the form of $U^T E = 0$, where $U = [x_1x_2, y_1x_2, x_2, x_1y_2, y_1y_2, y_2, x_1, y_1, 1]^T$, and $E = [E_{11}, E_{12}, E_{13}, E_{21}, E_{22}, E_{23}, E_{31}, E_{32}, E_{33}]^T$. By solving the linear least square problem

$$\min_E ||U^T E||^2$$

we obtain the essential matrix E, from which we can get the translation t and rotation R using the formula

$$\min_t ||E^T t||^2 \tag{3}$$

with the constraint $||t|| = 1$

$$\min_R \sum ||E - [t]_x R||^2 \tag{4}$$

with the constraint $R^T R = I$, $\det(R) = 1$. At last, with the relative motion between two images and pairs of corresponding points, we get the 3-D coordinates of the object using the following formula: $S_1 m_1 = A_1[I\ 0]M$, $S_2 m_2 = A_2[R\ t]M$. In the equations, we know m_1, m_2, A_1, A_2, I, R, and t; we want to get the unknown M and the scale factor S_1 and S_2. Eliminating S_1 and S_2, we obtain

$$\begin{bmatrix} a_1^T - u_1 a_3^T \\ a_2^T - v_1 a_3^T \\ b_1^T - u_2 b_3^T \\ b_2^T - u_2 b_3^T \end{bmatrix} M = \begin{bmatrix} 0 \\ 0 \\ (u_2 c_3 - c_1)^T t \\ (v_2 c_3 - c_2)^T t \end{bmatrix} \tag{5}$$

where a_i^T, b_i^T, and c_i^T are the ith row of the matrix A_1, B_2, and A_2 respectively, and it can be denoted by the simplified form $ZM = z$, then M equals $M = (Z^T Z)^{-1} Z^T z$, which are the 3-D coordinates for which we want to solve. One unknown parameter that needs to be further recovered is the scale factor S; to satisfy this, we manually labeled six known 3-D points.

After the 3-D reconstruction process, we can analysis and calculate the 3-D information to obtain the 3-D FAP. There are about 51 FAP vectors including the FAP that controls the motion of head, lip, cheek, and eye blink, and they can be obtained from the 3-D recovering information. This kind of FAP can directly drive an MPEG-4 compatible face modal.

IV. Estimation of a Three-Dimensional FAP From a Two-Dimensional Feature

In above section, the 3-D information can be recovered from 2-D features while knowing the parameters of the camera by some classic computer vision method. However, in many cases, it is difficult to automatically estimate the parameters of the camera taking the video. Therefore, the method above sometimes cannot work well at all. An alternative method is to estimate the 3-D information from a 2-D feature directly without any knowledge of the camera's parameters. Since the face true behavior is complex and manifold and the map from the 2-D visual feature to the 3-D FAP is inevitably ambiguous, it is a great challenge to do this direct mapping. To solve this kind of problem, we want to use some machine leaning technology, particularly context-based learning technology, to estimate the 3-D FAP from 2-D object features directly. Our learning engine integrates the Discrete Hidden Markov Model (DHMM) and Artificial Neural Network (ANN). The DHMM can constructed to piecewise approximate the face behavior manifold with quasi-linear submanifolds and glue together these pieces with transition probabilities [18]. Each state of the DHMM is described by a Gaussian probability of 3-D FAP. Under each state of DHMM, a neural network was built for mapping from the 2-D feature to the state label. There are several reasons for us to use a neural network instead of the original Gaussian Mixture Modal on each state. First, the number of states of the DHMM is quite large in our system, so calculating the transition and state probability for each state in each step is too intensive for real-time practicality. Second, we are not sure that the distribution of the 2-D feature, in terms of each state, are segmented according to the 3-D FAP, whether or not it satisfies the Gaussian probability. Therefore, we choose the neural network for the task of direct mapping. The advantage of using neural network is that, not only can we bypass the step of prior assumption of distribution of event, but we can also use this method to separate the whole complex learning process into two relevant simple parts.

Formally, the estimation approach proposed in this paper can be described as follows.

1) Obtain a video sequence with a given camera parameter. Track face landmarks in each frame of video for 2-D features, recovering 3-D information from the 2-D feature by computer vision technology, to obtain the 2-D and 3-D pair training data set. This 3-D information is used to generate a data set Ψ, called the FAP set, while the 2-D features are analyzed to generate the data set Φ.
2) The data set Ψ is initially clustered by anunsupervised approach to obtain some parameters for the DHMM. Then a DHMM is constructed and model the 3-D FAP in each discrete state. This is done using the Expectation-Maximization (EM) algorithm. In this way, we obtain a set of Ω of m clusters, each with roughly similar configurations. Then Φ will also be segmented to some subsets according to Ω.
3) For each cluster (state) i, we train a multilayer perceptron P_i to map 2-D visual features to a cluster label. In our experiment, the mapping is from Φ_i to Ω_j.
4) In the analysis and synthesis stage, the DHMM and neural networks are combined together to analyze tracked 2-D features, resulting in the most likely facial state sequence. The output of the DHMM is the 3-D FAP. This is rather like the case of speech recognition, except that the units of interest are facial states rather than phonemes.

A. Three-Dimensional FAP Configuration

To obtain a 3-D FAP configuration, the DHMM was considered for classification. Unfortunately, even for a small problem such as individual phoneme recognition, finding an adequate state machine is a matter of guesswork. This fact limits the utility of DHMM for more complex modeling because the structure of the state machine (pattern of available transition) is the most important determinant of a successful model. In our approach, the unsupervised clustering algorithm was proposed first for finding the number of states for DHMM. The Iterative Self-organizing Data (ISODATA) algorithm is proposed for our clustering [27]. The clustering parameters include: C: number of expected classes; MaxIterate: the maximal times for adjusting; MinSamples: the minimal number of objects in one class; I: combination parameter; and J: partition parameter. In order to evaluate the cluster algorithm in quantity and obtain suitable number of states, we conducted an experiment in calculating the squared error measure of divergence between the ground truth (X) and data mapping to cluster (Y)

$$\mathrm{ErrorSquare(X, Y)} = \frac{\sqrt{(X-Y)*(X-Y)^T}}{||X||}. \tag{6}$$

Table II indicates that the error square decreased while the number of clusters increased. The decreasing gradient will also become smaller. We choose the compromise number of 29 and

Fig. 7. Twenty-nine 3-D FAP Patterns. The mean errors of ground truth and clustering pattern mapping is 2.859 213 for all the tracking data.

TABLE II
PERFORMANCE COMPARISON FOR FACE ANALYSIS

	MinS	I and J	Result	Errorsquare
1	32	I=0.5-1,J=1-1.5	18	3.559787
2	20	I=0.5-1,J=1-1.5	21	4.813459
3	10	I=0.5-1,J=1-1.5	23	2.947106
4	5	I=0.5-1,J=1-1.5	29	2.916784
5	3	I=0.5-1,J=1-1.5	33	2.897993

simply pick the most probable configuration from each state. It can be viewed as the mean face. Many of them resemble visemes and common facial morph targets, augmented with dynamical content (see Fig. 7).

B. Statistical Dynamic Three-Dimensional Visual Model

For each state in the DHMM, the initial value will be set to each cluster center. Then the DHMM can be specified by $\theta = \{S, P_i, P_{i->j}\}$, where $S = s_1, s_2, \ldots, s_n$ is the set of discrete states. The stochastic matrix $P_{i->j}$ is the probability of transitioning from state i to state j. and stochastic vector P_i is the probability of a sequence beginning in state i.

The EM algorithm will be taken on the DHMM for training. Let us denote

$$\theta_i = \left(u_i, \sum_i\right) \tag{7}$$

to be the learned distribution parameters for cluster i. For

$$\theta = \{\theta_1, \theta_2 \cdots \theta_n\} \tag{8}$$

we want to seek an optimal embedded model

$$\theta^* = \{\theta_1, \theta_2 \cdots \theta_n\} \tag{9}$$

that maximizes the posterior given by the Bayesian rule $\theta^* = \arg\max_\theta P(\theta|X) \propto P(X|\theta)P(\theta)$ where the likelihood $P(X|\theta)$ measures accuracy in modeling the data and $P(\theta)$ measures consistency with the perplexity. Because $P(\theta)$ will be constant if the number of states is determined, the optimal embedded model will be obtained by finding the maximum-likelihood (ML) estimation of the following function:

$$\arg\max P(X|\theta) = \underset{s}{\operatorname{argmax}} \prod_{t=0}^{T} \theta_{s(t)|s(t+1)} N(x_t, u_{s(t)}, \Sigma_{s(t)}). \tag{10}$$

TABLE III
COMPARISON OF PERPLEXITY.

	Number of state	Perplexity (pp)
1	18	2.479012
2	21	2.152096
3	26	1.799709
4	29	1.623896
5	33	1.478828

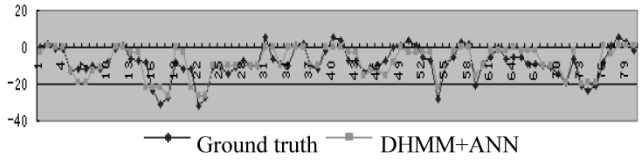


Fig. 8. Comparison of relative displacement of FAP4 according to feature point.

Having constructed a DHMM from a training video database, perplexity was adapted for measuring the performance of this model. The perplexity (PP) of model θ can be defined as

$$pp = 2^{H(F,\theta)} \approx 2^{-1/n \log p(F|\theta)} \tag{11}$$

where $F = f_1 f_2 \cdots f_Q$ denotes the FAP sequence of sentences, and

$$p(F|\theta) = \sum_i p(f_{i+1}|f_i \ldots f_1). \tag{12}$$

denotes the probability of FAP sequence F under model θ. The perplexity of different numbers of clusters is shown in Table III.

The purpose of the DHMM is to provide a mechanism that allows us to find the optimal face trajectory for a whole sentence, making use of forward and backward context. The DHMM can estimate the probability of a video sequence. After training, we obtain a learned probability distribution of state trajectory. The Viterbi algorithm can be applied to find the most likely sequence of predicted facial states.

C. Learning Mapping With an ANN

We can view the mapping from a low-resolution visual feature to the cluster center as a pattern recognition task. There are many learning machines that can be chosen for this task, such as HMMs, SVMs, and neural networks. Since neural networks have been shown to be an efficient and robust learning machine that solves input–output nonlinear mapping, we choose and train neural networks against many recorded sentences. A feed-forward ANN with three layers was constructed under each state. A multilayer perceptron with one hidden layer is employed for label (state). The explicit expression for this network is

$$y_k = f_2\left(\sum_{j=0}^{n_2} w_{kj}^{(2)} f_1\left(\sum_{i=0}^{n_1} w_{ji}^{(1)} x_i\right)\right) \tag{13}$$

where $x \in \Phi$ is the feature at a given instance, $w^{(1)}$ and $w^{(2)}$ are each layer's synaptic weights and biases, and f_1 and f_2 are sigmoid and linear functions, respectively. Training is relatively simple. Once we have obtained the data sets, we train the neural networks via Levenberg–Marquardt optimization to update the weights and biases.

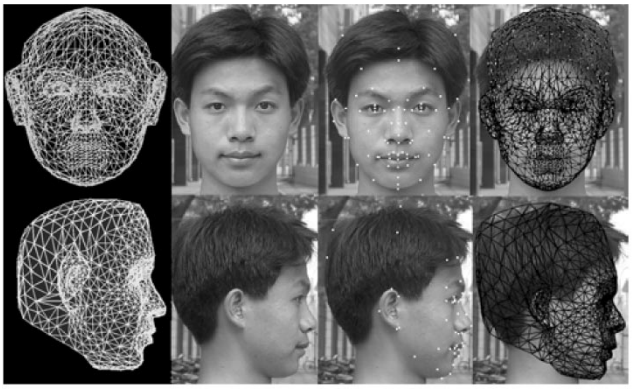


Fig. 9. Fitting a generic 3-D-face model to a personal one. (a) Generic 3-D-face model. (b) Feature points extracted. (c) Personal 3-D-face model.

For the neural network, the number of hidden units was 30 and the learning rate is set to be 0.001, and the network sum-squared error rate is 0.005.

The mean errors of ground truth and the prediction value of our proposed learning algorithm are 3.863 582. Fig. 8 shows the comparison of relative displacement of FAP4 according to feature point.

V. Individual Three-Dimensional Face Synthesis Based on Two Orthogonal Photos

Feature points automatically extracted as described in the previous section sometimes are not sufficiently accurate as requested. In order to rectify these possible errors, an interactive correcting mechanism is adapted [28]. Then, based on these feature points, a generic 3-D-face model is fitted to a personal one. Multidirection texture mapping is also proposed to synthesize a personal and lifelike face.

A. Fitting a Generic Three-Dimensional Model to a Personal One

Global and local transforms are designed to fit the generic 3-D-face model to a personal one. Global transform is used to adjust the global facial contour and the positions of the organs in the face. This is accomplished by scaling the coordinate values of each vertex of the model. The scaling factors are calculated according to the relations between the coordinates of the feature points in the generic face model and in the photos. The local transform aims at adjusting the shape of each organ such as the eyes, eyebrows, mouth, nose, and chin to fit the given persons' characteristic. Details can be found in [29]. Fig. 9 illustrates the adjusting procedure of the model.

B. Multidirection Texture Mapping

Texture mapping provides a valuable technique for further enhancing the factuality of the synthetic face. A multidirection texture mapping technique is presented to map the appropriate texture to the surface of the 3-D model. For each Bézier patch in the face surfaces from which the photo its texture is mapped depends on the whole normal direction of the Bézier patch. When the angle of the directional vector is less than 30 degrees, the frontal face image is used. Otherwise, the profile image is used. Fig. 10 illustrates the synthetic face for a given person.

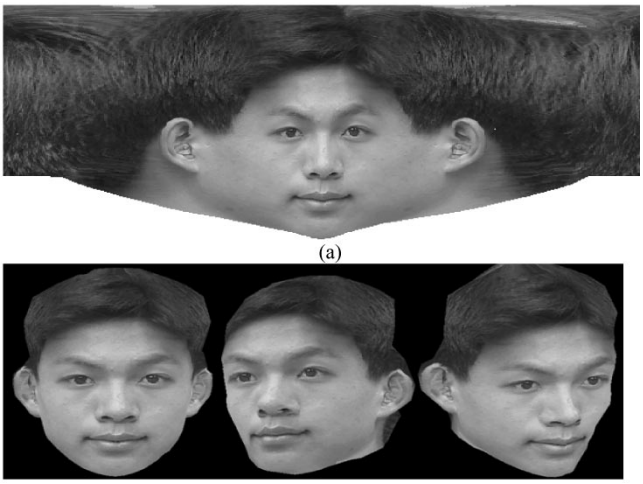


Fig. 10. Given person's synthetic face. (a) Texture from front and side images. (b) Synthetic lifelike face.

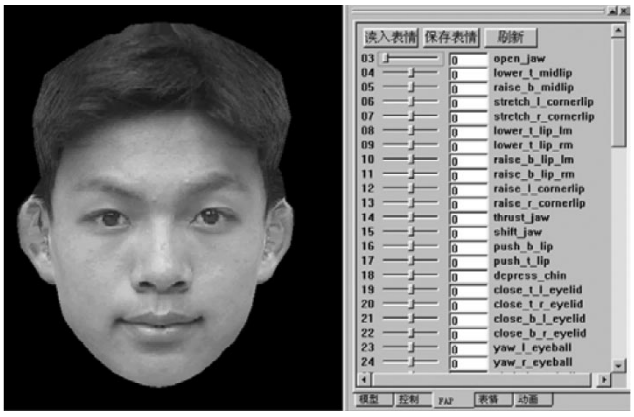


Fig. 11. Tools for 3-D high-resolution FAP modification.

VI. Experiment

We recorded subjects to read the text information of a speech synthesis database called CoSS-1. CoSS-1 includes the pronunciation of all isolate syllables, 2–4-word phrases, and some sentences. The 2-D feature was tracked by our tracking system automatically, and some 2-D features were tracked with some color dots in the low-texture facial area for the next 3-D information generation. We also built serial tools for FAP modification. Then we can eliminate the noise in data translation and modify the expression to the mean face under our DHMM state. Fig. 11 shows the tools and Fig. 12 shows the deformation result after modification by the tools..

We also can modify the FAP and obtain the expression. Fig. 13 shows the basic six expressions defined in MPEG-4.

VII. Real-Time Video-Driven MPEG-4 Based Three-Dimensional Face Animation

To integrate all the technologies mentioned in this paper, an integrated video-driven face animation system is implemented. In Fig. 14, the overview of the system is given.

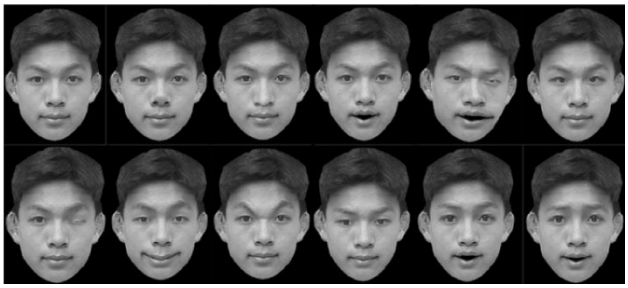


Fig. 12. Deformation for any face organs.

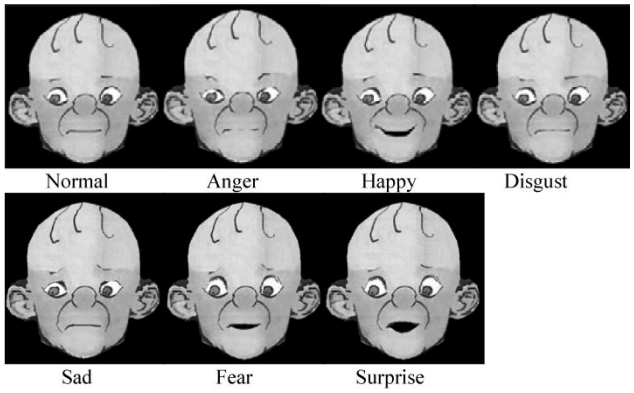


Fig. 13. Basic expressions of "San Mao" in MPEG-4.

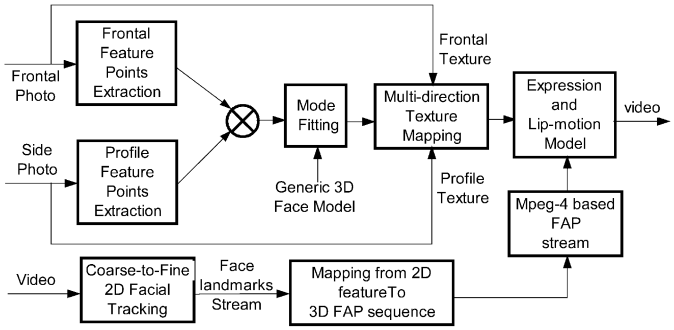


Fig. 14. System overview.

The individual 3-D face could be synthesized based on two orthogonal photos. With given video, the low-resolution facial feature can be detected by a tracking system. We can also infer the high-resolution FAP by mapping the low-resolution 2-D visual feature to a high-resolution FAP with the learned DHMM/ANN model. After determining the head motion, we then can reconstruct most of the MPEG-4 feature points (we do not include the teeth and the tongue) in 3-D space and generating the texture image from the video sequence. Fig. 15 shows the process.

VIII. Conclusion and Future Work

In this paper, the framework of our system of video-driven individual face animation is presented in which a new methodology for tracking and recovering 3-D MPEG-4 facial expression and lip motion from video sequence is proposed. Our method has the advantages over previous systems of real-time robust face tracking in video and of mapping and recovering the FAP from 2-D feature points to 3-D parameters.

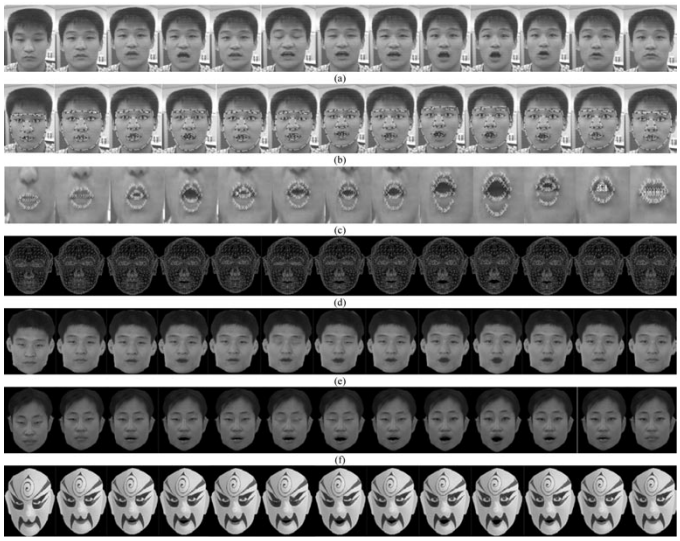


Fig. 15. System overview of the video-driven MPEG-4 compatible 3-D face animation system. (a) Image from video. (b) Tracking result based on our face AAM tracking system. (c) Tracking result based on our lip ASM tracking system. (d) Mapping tracking result to high-resolution FAP. (e) Synthesis face animation. (f) Synthesis other person's face animation using (d) FAP. (g) Synthesis of the Beijing opera face animation using (d).

We also present a technique for individual MPEG-4-based face synthesis using two given photos taken in the orthogonal direction, frontal and side. We applied all proposed techniques to our MPEG-4 compatible video-driven face animation system. The system can automatically detect and track facial feature points of a given individual, calculate the appropriate features, estimate the MPEG-4-based FAP parameters, and then drive the synthesized face animation. Results show that the system performs quite satisfisfactorily in the current testing environment, and we believe that this technology is the next step toward a pure application in making individual face animation based on MPEG-4 FAP.

In the future, we will consider improving our system in the following aspects. For 3-D FAP, now we only use some computer vision technology; since obtaining accurate data was more important than the testing tracking method, we can build the 3-D FAP database via two cameras and with more marks on the face or just using motion capture equipment. Also, we should record the audio-visual database, use training machine mapping from speech feature to visual feature, and integrate this information to our learned 3-D FAP to generate more realistic expressions, lip shape, and face animation.

As we try to build computer systems that interact with their users in much more natural ways, analyzing face images is becoming a key issue. We believe that tracking, learning, and animating the human face as defined in MPEG-4 is a powerful paradigm.

Acknowledgment

The authors would like thank all contributors involved in system development, including Dr. B. Yin for graciously providing the face rendering and Z. Li, L. Zuo, and J. Wang for performing the training data and assisting with processing and analyzing the data.

REFERENCES

[1] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 569–579, 1993.

[2] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making faces," in *Proc. SIGGRAPH*, July 1998, pp. 55–66.

[3] L. Williams, "Performance-driven facial animation," in *Proc. SIGGRAPH*, vol. 24, Aug. 1990, pp. 235–242.

[4] M. Black and Y. Yacoob, "Tracking and recognizing rigid and nonrigid facial motions using local parametric models of image options," in *Proc. Int. Conf. Computer Vision*, 1995, pp. 374–381.

[5] M. Covell, "Eigen-points: Control-point location using principal component analysis," in *Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition*, Oct. 1996, pp. 122–127.

[6] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. Computer Vision and Pattern Recognition*, 1996, pp. 403–410.

[7] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Reading, MA: Addison-Wesley, 1998.

[8] A. Lanitis, C. J. Taylor, and T. F. Cootes, "A unified approach for coding and interpreting face images," in *Proc. 5th Int. Conf. Computer Vision (ICCV 95)*, Cambridge, MA, June 1995, pp. 368–373.

[9] K. Matsino, C. W. Lee, S. Kimura, and S. Tsuji, "Automatic recognition of human facial expressions," *Proc. IEEE*, pp. 352–359, 1995.

[10] T. S. Jebara and A. Pentland, "Parametrized Structure from Motion for 3D Adaptive Feedback Tracking of Faces," MIT, Cambridge, MIT Media Laboratory #410, 1996.

[11] S. Basu, N. Oliver, and A. Pentland, "3D modeling and tracking of human lip motions," in *Proc. ICCV98*, Jan. 1998, pp. 337–343.

[12] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 580–591, June 1993.

[13] G. Hager and K. Toyama, "The Xvision system: A general purpose substrate for portable real-time vision applications," *Comput. Vis. Image Understanding*, vol. 69, no. 1, pp. 23–37, 1997.

[14] E. Elagin, J. Steffens, and H. Neven, "Automatic pose estimatino system for human faces based on bunch graph matching technology," in *Proc. ICAFGR*, 1998, pp. 136–141.

[15] I. Essa and A. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 757–763, July 1997.

[16] D. Decarlo and D. Metaxas, "Deformable model-based shape and motion analysis from images using motion residual error," in *Proc. Int. Conf. Computer Vision*, 1998, pp. 113–119.

[17] T. Vetter and V. Blanz, "Estimating colored 3D face models from single images: An example based approach," in *Proc. Eur. Conf. Computer Vision*, 1998, pp. 499–513.

[18] M. Brand and A. Hertzmann, "Style machines," in *Proc. SIGGRAPH*, 2000, pp. 183–192.

[19] R. Rosales and S. Sclaroff, "Learning and synthesizing human body motion and posture," *Proc. 4th IEEE Int. Conf. Automatic Face and Gesture Recognition (FG2000)*, 2000.

[20] R. Rosales, V. Athitsos, L. signal, and S. Sclaroff, "3D hand pose reconstruction using specialized mapping," *Proc. IEEE Int. Conf. Computer Vision (ICCV2001)*, 2001.

[21] J. Ostermann, "Animation of synthetic faces in MPEG-4," *Computer Animation*, pp. 49–51, June 1998.

[22] W. Wang, S. Shan, W. Gao, and B. Cao, "An improved active shape model for face alignment," in *Proc. 4th Int. Conf. Multi-Modal Interface*, Pittsburgh, PA, Oct. 2002, pp. 523–528.

[23] J. Miao, W. Gao, Y. Chen, and J. Lu, "Gravity-center template based human face feature detection," in *Lecture Notes in Computer Science* Berlin, Germany, 2000, vol. 1948, Proc. Int. Conf. Multimodal Interface, pp. 207–214.

[24] B. Cao, S. Shan, W. Gao, and D. Zhao, "Localizing the iris center by region growing search," *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 2, pp. 129–132, 2002.2.

[25] R. Wang, W. Gao, and J. Ma, "A novel approach to robust and fast locating lip motion," in *Lecture Notes in Computer Science* Berlin, Germany, Aug. 2000, Proc. 3rd Int. Conf. Multimodal Interface, pp. 582–589.

[26] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. Eur. Conf. Computer Vision*, vol. 2, 1998, pp. 484–498.

[27] Y. Chen, W. Gao, T. Zhu, and J. Ma, "Multi-stragy data mining framework for mandarin prosodic pattern," in *Proc. ICSLP2000*, vol. 2, 2000, pp. 59–62.

[28] W. Gao, J. Yan, B. Yin, and Y. Song, "An individual facial image synthesis system for virtual human," in *Proc. 2nd Int. Conf. Multimodal Interface*, 1999, pp. 20–25.

[29] S. Shan, W. Gao, J. Yan, H. Zhang, and X. Chen, "Individual 3D face synthesis based on orthogonal photos and speech-driven facial animation," in *Proc. Int. Conf. Image Process (ICIP2000)*, 2000, pp. 238–241.

Wen Gao (M'99) received the M.S. degree and the Ph.D degree in computer science from Harbin Institute of Technology, Harbin, China, in 1985 and 1988, respectively, and the Ph.D degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Research Fellow at the Institute of Medical Electronics Engineering, the University of Tokyo, in 1992, and a Visiting Professor at the Robotics Institute, Carnegie Mellon University, Puttsburgh, PA, in 1993. From 1994 to 1995 he was a Visiting Professor at MIT Artificial Intelligence Laboratories. Currently, he is the Vice President of the University of Science and Technology of China, the Deputy President of Graduate School of Chinese Academy of Sciences, Professor in Computer Science at Harbin Institute of Technology, and Honor Professor in Computer Science at City University of Hong Kong. He has published 7 books and over 200 scientific papers. His research interests are in the areas of signal processing, image and video communication, computer vision and artificial intelligence.

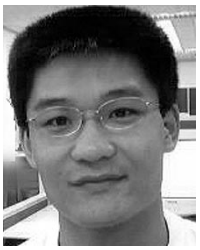
Dr. Gao is the head of Chinese National Delegation to MPEG working group (ISO/SC29/WG11). He is the Editor-in-Chief of the Chinese Journal of Computer, and the general co-chair of the IEEE International Conference on Multimodel Interface in 2002.

Yiqiang Chen (S'02) received the B.S. degree and the M.S. degree in computer science from XiangTan University, China, in 1996 and 1999, respectively, and the Ph.D degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China in 2003.

He is currently an Assistant Researcher with the Digital Laboratory, Institute of Computing Technology. His research interests include machine learning, multimodal interfaces, and bioinformatics.

Rui Wang received the B.S. degree and the M..S. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1990 and 1996, respectively. He is currently working toward the Ph.D. degree at the same university.

His research interests include pattern recognition, computer vision, and image processing.



Shiguang Shan received the B.S. degree and M.S. degree in computer science from Harbin Institute of Computing Technology, Harbin, China, in 1993 and 1999, respectively. He is currently working toward the Ph.D. degree in computer science at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

His research interests include pattern recognition, image and vision computing, and human–computer interface.

Dalong Jiang received the B.S. degree in computer science from Tsinghua University, Tsinghua, China, in 1999. He is currently working toward the Ph.D degree in computer science at the Institute of Computer Technology, Chinese Academy of Sciences, Beijing.

He has been a Research Assistant at the Joint R&D Lab (JDL), Chinese Academy of Sciences, since 1999. His research interests include virtual reality, computer graphics, and animation.