A Real Time Face Tracking And Animation System

Xiaozhou Wei[†], Zhiwei Zhu[‡], Lijun Yin[†][*], Qiang Ji[‡]

[†]Department of Computer Science, SUNY at Binghamton, Binghamton, NY 13902
[‡]Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract

In this paper, a novel system for real time face tracking and animation is presented. The system is composed of two major components: (1) real time infra-red (IR) based active facial feature tracking, and (2) real time facial expression generation based on a 3D face avatar. Twenty-two feature points, head pose orientation and eye close-open status are effectively extracted through a video input. Based on the detected facial features, a 3D face model is animated by a dynamic inference algorithm and a transformation from facial motion parameters to facial animation parameters. The work can be extended to the fields of real time facial expression analysis and synthesis for applications of human-computer interaction, model-based video conferencing and low bit rate avatar communication. The performance of the developed system is evaluated by its real time implementation for facial expression generation.

1 Introduction

Research on face tracking and animation techniques has been intensified due to its wide range of applications in security, entertainment industry, gaming, psychological facial expression analysis and human computer interaction. Recent advances in face video processing and compression have made face-to-face communication be practical in real world applications. However, higher bandwidth is still highly demanded due to the increasing intensive communication. Model based low bit rate transmission with high quality video offers a great potential to mitigate the problem raised by limited communication resources. However, after a decade's effort, robust and realistic real time face tracking and generation still pose a big challenge. The difficulty lies in a number of issues including the real time face feature tracking under a variety of imaging conditions(e.g., lighting variation, pose change, self-occlusion and multiple non-rigid features deformation), and the real time realistic face modeling and animation using a very limited number of feature parameters.

Traditionally, the head motion is modeled as a 3D rigid motion with the local skin deformation [9, 10], the linear motion tracking method cannot represent the rapid head motion and dramatic expression change accurately. The appearance-driven approach requires a significant number of training data to enumerate all the possible appearances of features. The model based approach [8, 11] assumes the knowledge of a specific object is available, meanwhile the requirement of frontal facial views and constant illumination limited its application. All above tracking methods have shown certain limitations for accurate face feature tracking under complex imaging conditions. In this paper, we present an active facial tracking system to tackle the issues of variable illuminations, rigid (head motion) and non-rigid (local skin deformation) feature tracking, and the self-occlusion of features. The system has been successfully applied to fatigue detection, here we extend the system to the prospective application for facial expression production and model based video conferencing for human-to-human communication as well as human-to-computer interaction.

In order to produce a realistic facial expression for avatar communication, a real time facial animation rule must be carefully designed. Performance-driven facial expression generation relies on the ability of facial feature detection. However, it is not trivial to create a high quality face animation given few feature data available. The methods using physical based animation and elastically deformable models show the realistic results [5], however mimicking the anatomical structure and dynamics of human faces involves in a time-consuming procedure, and hence is not feasible for real-time application. Image-based model fitting methods, which make use of the existing models (e.g., general refined model [1] or 3D morphable model [2, 3, 6]), synthesized the impressive facial appearances, however the mapping process or the analysis-by-synthesis loop for error energy minimization requires significantly intensive computations. Recently, modification-based methods attract more attention for facial expression cloning. By using existing animation data in the form of vertex motion vectors, an expression could be copied from one model to another [4].

[*] Contact author: lijun@cs.binghamton.edu. The work is supported in part by the NSF (IIS-0414029) and the AFOSR.

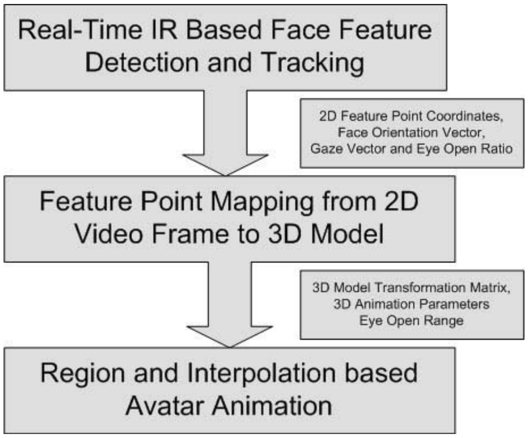
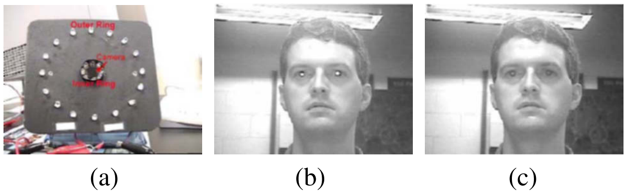

Real-Time IR Based Face Feature Detection and Tracking
2D Feature Point Coordinates, Face Orientation Vector, Gaze Vector and Eye Open Ratio
Feature Point Mapping from 2D Video Frame to 3D Model
3D Model Transformation Matrix, 3D Animation Parameters Eye Open Range
Region and Interpolation based Avatar Animation

Figure 1: The framework of the real-time active face tracking and animation system.

This approach makes it easy to create animation on different models by reusing the original model data. Noh *et al.* [4] developed a model deformation method based on RBF (Radial Basis Functions) interpolation. With the assumption of facial surface smoothness, RBF-based animation can produce realistic expressions by using an arbitrary sparse set of 3D control points. However due to the lack of fine modeling on certain expression-rich areas, such as eyes, it is hard to track and create eye opening and gazing given the input of 2D motion vectors.

In this paper, we present a new approach for data transform from 2D feature motion parameters to 3D feature animation parameters, the animation of facial expression on expression-rich regions such as eyes and mouth can be effectively generated given very few control points as input. A dynamic inference algorithm is developed for non-feature vertices animation, and eventually reproduces facial expressions on a virtual avatar with a faithful fidelity.

In general, our real time active face tracking and animation system is outlined in Figure 1, which consists of three modules: (1) IR-based active face tracking, which outputs a set of 2D motion parameters plus a set of auxiliary information (i.e., face orientation, gaze vector and eye-open ratio); (2) feature parameter adaptation from 2D motion data to 3D animation data on the individualized 3D model (e.g., avatar), and (3) region-based dynamic inference of animation parameters for each non-feature vertex using a coarse to fine strategy to finalize the procedure for expression creation. The organization of this paper is as follows: In section 2, we will overview the developed active face tracking system. The individualization of motion parameters and the corresponding animation rule will be described in Section 3 and 4, followed by experimental results and evaluations in Section 5. Finally the concluding remarks will be given in Section 6.



(a) (b) (c)

Figure 2: IR active sensing system: (a) hardware setup: the camera with an active IR illuminator (b) bright pupil in an even field image (c) dark pupils in an odd field image.

2 Active Face Tracking

In order to obtain the motion parameters used for face model animation (e.g., rigid head motion, non-rigid feature deformation, eye open status and gaze), four major components are developed, which are (1) active facial sensing for eye pupil detections; (2) Gabor-wavelet based initial feature identification; (3) prediction and tracking of feature locations by combining Kalman filtering with head and pupil motions, and (4) face pose tracking. The following sections provide an overview of each component, detailed description can be found in [7].

2.1 IR-based eye tracking

In order to locate the face region, the first significant feature on the face must be reliably extracted. Among all the facial features, the eye pupil is believed to be able to provide strong and reliable constraints for initializing the tracking system. Starting from this key feature, the subsequent features are then possible to detect or infer.

IR based eye tracking techniques have been proved to be a stable approach to detect pupils and face location [7, 15]. Our approach to pupil detection relies on an active IR illumination so that it can work robustly under a variety of lighting conditions and head orientations. The active facial sensing system consists of an IR sensitive camera and two concentric rings of IR LEDs as shown in Figure 2(a). The inner ring of LEDs and outer ring of LEDs are synchronized with the even and odd fields of the interlaced image respectively. The person's face is illuminated with an IR illuminator, which produces the bright pupil image in the even field and the dark pupil image in the odd field, as shown in Figure 2(b)(c). Because the two fields are similar, the pupils can be easily extracted from the difference of the two images. Note that besides the accurate detection of pupils, the active sensing system can also produce a normal grayscale image (odd field) which is insensitive to any lighting condition, it thus allows us to use the conventional methods to track other facial features.

The eye tracking result is further improved by a mean shift eye tracker [13]. The shapes of the pupils are extracted by a deformable template method (so-called ellipse fitting technique). The degree of eye opening is characterized by the shape of pupil. It is observed that as eyes close, pupils start getting occluded by the eyelids and their shapes get more elliptical. Therefore, the ratio of pupil ellipse axes is used to characterize the percentage of eye closure, the detailed algorithm is described in [13].

2.2 Initial feature detection by Gabor wavelet

Twenty-two facial features around eyes and mouth are selected for tracking. We use the multi-scale and multi-orientation Gabor wavelet to represent each feature. To identify each facial feature at the initial frame, 18 Gabor coefficients are used to represent each feature pixel and its vicinity [16], a set of 18 Gabor coefficients $\Omega(\vec{x})$ is obtained by the convolution operation:

$$\Omega(\vec{x}) = \int I(\vec{x}')\mathbf{\Psi}[\mathbf{k},(\vec{x}-\vec{x}')]d^2\vec{x}' = (c_1, c_2, ..., c_{18})^T \tag{1}$$

where 2D Gabor kernels are defined as: $\mathbf{\Psi}(\mathbf{k},\vec{x}) = (\mathbf{k}^2/\sigma^2)e^{-\mathbf{k}^2\vec{x}^2/2\sigma^2}(e^{i\mathbf{k}\cdot\vec{x}} - e^{-\sigma^2/2})$; $\sigma = \pi$ is set for 128×128 images. The set of Gabor kernels consists of 3 spatial frequencies (with wavenumber k: $\pi/2, \pi/4, \pi/8$), and 6 distinct orientations from 0^o to 180^o in the interval of 30^o.

2.3 Feature tracking by Kalman filter

With the assumption of smooth motion of each facial feature, the well known tracking method, Kalman filter, can be used for facial feature tracking. The motion state of each feature at each frame (time instance) can be formulated by the following state model:

$$\mathbf{S_{t+1}} = \mathbf{\Phi}\mathbf{S_t} + \mathbf{W_t} \tag{2}$$

where $\mathbf{\Phi}$ is the transition matrix and $\mathbf{W_t}$ models system perturbation. The state vector $\mathbf{S_t}$ at time t is characterized by its position and velocity, i.e., $\mathbf{S_t} = (\vec{P}, \vec{V})$. A measurement model used in Kalman filter is further defined as:

$$\mathbf{O_t} = H\mathbf{S_t} + \mathbf{U_t} \tag{3}$$

where H is the measurement matrix and $\mathbf{U_t}$ models measurement uncertainty. Based on above two models and some initial conditions (e.g., detected pupil position), the state vector $\mathbf{S_{t+1}}$ along with its covariance matrix can be updated, and therefore the position prediction of each feature ($\vec{P} = (x, y)^T$) can be obtained. Finally, by combining the head motion with the Kalman filtering, we can obtain the accurate feature location in the current frame. In the system implementation, a three-level coarse to fine phase-sensitivity function is used. For each feature only three displacement values are calculated to determine the optimal feature position, which dramatically speeds up the detection process and makes the real-time implementation possible. In order to handle the case of mis-tracking or self-occlusion, a confidence verification procedure is further carried out as a post processing. The detected features are verified and inferred by both qualitative and quantitative evaluations. The qualitative evaluation checks the correctness of the spatial relationships, such as whether the two corner points of the eye are on the different sides of the pupil, etc. The quantitative one compares the extracted relationship data (e.g., aspect ratios and joint angles) with the corresponding data in the customized facial model. If the wrong feature is detected by the verification, the previous known relationship data are used for the make-up.

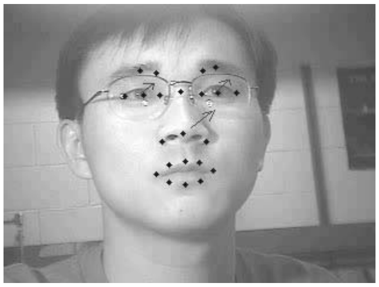


Figure 3: A sample frame showing the tracking result: 22 features (dark diamonds), head orientation vector (dark arrow on the nose tip) and gaze vectors (dark arrow in the locations of two pupils).

2.4 Face pose tracking

Based on the detected eyes and certain anthropometric statistics, a front-parallel face can be detected automatically. The detected face region will serve as the initial 3D planar face model. The 3D face pose is then tracked in 3D space using Kalman filter, starting from the front-parallel face pose. During the process of tracking, face detection and face pose estimation are synchronized such that the detected face and the estimated face pose are kept consistent with each other. In addition, the 3D face model is updated continuously in order to accommodate face change due to variations in illumination, face aspect, and facial expression. The detected face normal vector is perpendicular to the face plane, which is represented by three angles of orientation $(\alpha_{roll}, \beta_{pitch}, \gamma_{yaw})$ (detailed description can be found in [14]). An example of tracking face is shown in Figure 3, where 22 most reliable features distributed on areas of eye, iris, eyebrow, nose and mouth, head orientation vector and gaze vectors are successfully tracked.

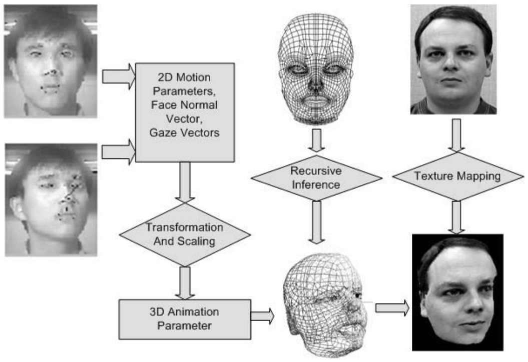

2D Motion Parameters, Face Normal Vector, Gaze Vectors
Recursive Inference
Texture Mapping
Transformation And Scaling
3D Animation Parameter

Figure 4: Feature Mapping and Animation Diagram

3 Obtaining 3D Animation Data

The faithful reproduction of facial expressions on a 3D avatar model requires a sufficient number of 3D animation parameters, for example, MPEG4 defines 68 features as animation parameters. However, in our current performance-driven system, the most dependable features are distributed over 22 expression-rich areas including eyes, eyebrow, nose and mouth. The small number of features can significantly reduce the amount of data to transmit and thus increase the compression ratio. However, it requires a more advanced animation rule to be designed for creating a realistic animation. The motion parameters obtained from our tracking system can not be directly applied to the 3D facial model, the reason for that lies in: (1) the motion vectors to be derived from a source subject could be different with the target subject (e.g., 3D avatar); (2) the motion vectors are represented in 2D plane; (3) our 3D avatar model contains a large number of vertices. Although the higher resolution model is believed to be better for creating a more realistic animation, large amount of non-feature vertices increase the animation difficulty because the existing features are two few to infer enough cues for expression generation. In view of above factors, we propose a new animation scheme to enhance the resulting effect. Figure 4 shows the framework of our animation system. First, the set of 2D motion parameters (MP) of tracked feature points are transfered to 3D animation parameters (AP) on the corresponding feature vertices based on a ”depth pattern” of the source subject and the head orientation vector; Second, the 3D animation parameters are adapted to the individual 3D avatar by a scaling and normalization process, the resulting APs are transformed to a new set from the current view to the front view, on which the subsequent deformation of non-feature vertices will be done. Third, the 3D model is animated by using the transformed AP values, a so-called region based dynamic inference algorithm is proposed to animate the large amount of non-feature vertices, and finally, the animated 3D model in the front view is eventually rotated back to the view of the current frame.

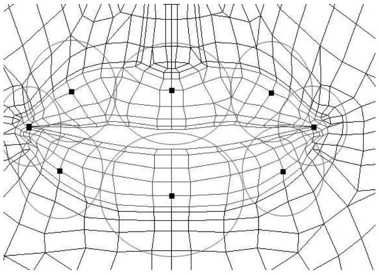


Figure 5: Regions splitting and definition on the face model. Here the mouth is shown as an example: black squares represent feature points, circled area is the region controlled by center control point

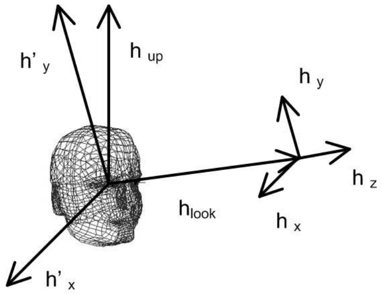


Figure 6: Construction of Rotation Matrix H

3.1 Transfer from source MPs to target APs

The individual 3D face avatar is firstly created by our face modeling algorithm based on two views of an individual face. We developed a novel algorithm LMCT (local maximum curvature tracing) [17] to reliably identify features on the profile such as chin tip, mouth, nose tip, and nose bridge. Based on the fiducial points detected on the both views of the face, a generic wireframe model is modified separately for the front view and side view in 2D space. A 3D individual model is then constructed by combining two 2D models modified by each view. The algorithm is general enough to work similarly when more facial views are given as input. After the process of static face modeling, regions like mouth, nose, eye, eyeballs, and eyebrows are then marked with numbered labels. The areas to be influenced by the feature points are defined as well. Some units that have relative movement between different sub-regions within one region have to be further divided. For instance, eight feature points are tracked surrounding the mouth area, thus the lip has to be divided into eight regions in order to apply motion vectors accordingly as shown in Figure 5. Likewise the eye and the eyebrow are also flagged to several distinguished sub-regions.

In order to obtain APs, we need to process source MPs by a series of transformations. All the feature points with arbitrary views must be transformed to the front view such that the animation parameters can be derived with respect to the front view.

The motion of each feature point on a face can be modeled as a local skin deformation plus a global head affine transformation. Given a feature point $\vec{v_o}$ with the front view of a source subject, its corresponding point ($\vec{v}_p$) after the local deformation due to expressions and global motion due to the head pose change is formulated as follows:

$$\vec{v}_p = T \cdot R \cdot \vec{v}_f, \quad \vec{v}_f = \delta\vec{v} + \vec{v}_o \tag{4}$$

where

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$\mathbf{R} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$\vec{v}_p = \{x_p, y_p, z_p, 1\}^T \quad \vec{v}_f = \{x_f, y_f, z_f, 1\}^T \tag{7}$$

$$\vec{v}_o = \{x_o, y_o, z_o, 1\}^T, \quad \delta\vec{v} = \{\delta x, \delta y, \delta z, 1\}^T \tag{8}$$

where $\vec{v}_o$ is a feature point with neutral expression of the front view. $\delta\vec{v}$ is the deformation applied on $\vec{v}_o$. By translation (T) and rotation (R), the face features are projected to the image plane with the head pose of the current frame. Thus by invert transformation the conversion from projection space to model space is derived by:

$$\vec{v}_f = H \cdot T^{-1} \cdot \vec{v}_p \tag{9}$$

where H is the inverse matrix of R:

$$\mathbf{H} = \begin{pmatrix} h_{x_1} & h_{x_2} & h_{x_3} & 0 \\ h_{y_1} & h_{y_2} & h_{y_3} & 0 \\ h_{z_1} & h_{z_2} & h_{z_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

$$\mathbf{T^{-1}} = \begin{pmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

H can be derived by the head orientation vector: First we get the face normal vector from video tracking, namely $\vec{h}_{look}$. $\vec{h}_{look}$ is perpendicular to the face frontal plane. The head-up vector $\vec{h}_{up}$ is an initial constant vector for computing the vector $\vec{h}_x$ as shown in Figure 6.

From the relation of $\vec{h}_{up}$ and $\vec{h}_{look}$, $\vec{h}_x$ can be derived by the cross-product operation. $\vec{h}_y$ is then obtained by $\vec{h}_z$ and $\vec{h}_x$ (see Figure 6):

$$\vec{h}_z = \frac{\vec{h}_{look}}{|\vec{h}_{look}|}, \quad \vec{h}_x = \frac{\vec{h}_z \times \vec{h}_{up}}{|\vec{h}_z \times \vec{h}_{up}|}, \quad \vec{h}_y = \vec{h}_z \times \vec{h}_x \tag{12}$$

From above, we can derive the feature point $\vec{v}_f$ of front view from arbitrary view $\vec{v}_p$ as shown in Equation 13:

$$\vec{v}_f = \begin{pmatrix} h_{x_1}(x_p - T_x) + h_{x_2}(y_p - T_y) + h_{x_3}(z_p - T_z) \\ h_{y_1}(x_p - T_x) + h_{y_2}(y_p - T_y) + h_{y_3}(z_p - T_z) \\ h_{z_1}(x_p - T_x) + h_{z_2}(y_p - T_y) + h_{z_3}(z_p - T_z) \\ 1 \end{pmatrix} \tag{13}$$

As such, the APs are derived as:

$$\delta\vec{v} = \vec{v}_f - \vec{v}_o \tag{14}$$

As we can see, z_p values are unavailable from the projection plane (i.e., image plane). Simply omitting the depth information will result in the error in estimating the APs. Here we use an approximation method to minimize the error. A so-called "depth pattern" is defined for z_o values of all feature points based on our face modeling values from the side view of the specific person's face [17]. z'_p, the approximation of z_p, is obtained by transforming $\vec{v}_o$ to $\vec{v}'_p$, where

$$\vec{v}'_p = T \cdot R \cdot \vec{v}_o \tag{15}$$

Then we can derive:

$$z'_p = w_x x_o + w_y y_o + w_z z_o + T_z \tag{16}$$

Let's denote $w_r = w_x x_o + w_y y_o + w_z z_o$, and replace the z_p of the Equation 13 by z'_p, we obtain the approximation $\vec{v}'_f$:

$$\vec{v}'_f = \begin{pmatrix} h_{x_1}(x_p - T_x) + h_{x_2}(y_p - T_y) + h_{x_3} w_r \\ h_{y_1}(x_p - T_x) + h_{y_2}(y_p - T_y) + h_{y_3} w_r \\ h_{z_1}(x_p - T_x) + h_{z_2}(y_p - T_y) + h_{z_3} w_r \\ 1 \end{pmatrix} \tag{17}$$

The approximation of APs is derived as:

$$\delta\vec{v'} = \vec{v}'_f - \vec{v}_o \tag{18}$$

Note that the approximation of z_p makes the calculation of $\vec{v}'_f$ omit the influence of T_z. In other words, limited by the T_z estimation, our approximation approach is based on the assumption that the head movement in the back-and-forth direction is limited in a very small range. Therefore, we assume $T_z = 0$.

In the situation that the depth values of the source face are not available, we can use the depth pattern of a generic face model or the similar person's depth pattern to approximate the z_o value. Our experiment shows that this method is effective and feasible in our real time application. The existence of translation could be judged by the variation of eye distance in projection plane. If the distance is not changed while there exists rotation, there is translation occurred. The displacement of the center of in-between two eyes is used as our translation parameters.

3.2 Animation Data Adaptation

The animation data obtained in the previous step are from the source subject, which may not be suitable for the target model. Further adjustment of these parameters is necessary in order to adapt these data to the avatar model. First, the APs are scaled to fit to the 3D model size. The individual scaling process is conducted in the individual feature regions, such as eye, nose, mouth, etc. The size of these individual region can be used for normalizing the adjusted APs.

After the normalization, the AP data are ready to apply to the target face model. Eventually, the animated model is transformed (by $T \cdot R$) to the view which is corresponding to the face pose of the current frame.

To animate vertices that are not grouped into any region yet influenced by the movement of feature points, we animate them based on a control-points-based dynamic inference algorithm, the detail is discussed in the next section.

4 Region Based Animation By Dynamic Inference

For feature points and interpolated feature vertices on 3D generic model, the animation parameter is obtained by feature point mapping and normalization. The animation parameter of non-feature vertices can be derived by the interpolation of AP of feature vertices and the derived AP of non-feature vertices in the previous steps.

4.1 Feature points animation

In addition to the motion vectors of tracked face features, eye open ratio and gaze vectors are captured as well. These parameters are used to control the opening and closure of eyes and the motion of pupils. A finer modeling of eyes and mouth is needed to ensure detail semantic coding. Ten small regions are defined to model surrounding area of each eye. For each region one vertex is selected to represent the region location, other vertices in the same region are grouped. A parabolic eye model as shown in Figure 7 is used to control the eye lid motion. Since the eye open ratio is provided, the motion vectors of eye lids can be obtained based on the ten feature points defined on the eye lid model.

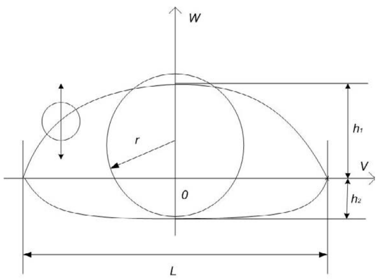


Figure 7: Animation of Eyes and eyeball

The eye template model (as shown in Figure 7) consists of a pair of parabolic curves (W_1, W_2), and a circle with radius r. L is the width of the eye, h_1, h_2 are the opening heights of the upper and lower eyelid respectively. The parabolic curves are defined as:

$$W_i = h_i \cdot (1 - (\frac{V}{L})^2) \qquad i = 1, 2$$

The width of eye can be calculated as the distance between left and right corner point of eye. The upper and lower heights of eyelid can be calculated as the distance between up-most(under-most) vertex and central point.

4.2 Non-feature points animation

The 3D animation vector V of non-feature vertex i (denoted as V^i) can be derived by the following equation:

$$V^i = \sum_{j=1}^{N} (\omega(d_{i,j}) \cdot V_k^j)$$

where $d_{i,j}$ is the distance between vertex i and vertex j, $d_{i,j}, j = 1, 2, \ldots, N, (e.g.\ N = 3)$ have been arranged in increasing order. $\omega(d_{i,j})$ is a weight function which will have a large output of weight value for a small input of $d_{i,j}$ (see Figure 8),

$$\omega(d_{i,j}) = \frac{d_{i,j'}}{\sum_{j=1}^{N} d_{i,j}}, \qquad where\ j' = (N + 1) - j$$

Note that the previously derived non-feature points are also added into the feature points list. This dynamic growing method infers the APs of non-feature points based on the updated feature points list. The newly added feature points will influence the non-feature points of the local region, thereby the region of influence on a non-feature point is dynamically changed. While more feature points are considered, the calculation turns out to be more complex. In view of the fact that control points which are further from current vertex pay less influence on the final movement of this vertex, we could use only those features that contribute

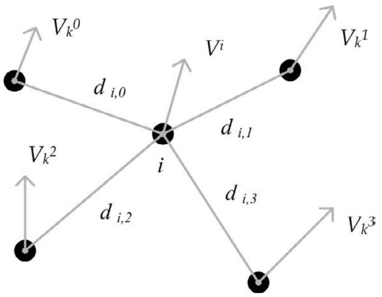


Figure 8: The animation parameter for a non-feature vertex i is obtained by the dynamically inferred surrounding feature vertices.

significantly to the movement of current vertex. In our experiment, less than 10 nearest features points are considered for achieving the satisfactory results.

5 Experiments and Analysis

The real time system is realized with the setup of regular Pentium-IV PC and an IR-associated video camera, which runs fully automatically from active face tracking to 3D model animation. Our 3D face model consists of 2997 vertices. Note that the 3D avatar model is created off-line from two views of a person's face. This face modeling process allows us to create an individualized 3D avatar for each subject efficiently. Figure 9 shows several sample frames captured from our real time video clip [1] are illustrated in Figure 9. It shows that the motion of one source subject is faithfully transferred to the different target avatars (Row 2 for a male avatar and Row 3 for a female avatar). Experiment shows the good visual quality on created expressions with movements of eye, eyebrow and mouth, eye open/closure and head pose change. The active tracking system works well as shown in Figure 10, for example, features and head pose are reliably extracted even with eye-glasses or in the condition of different lighting and self-occlusion.

Our face tracker always starts with eye tracking. As such, the eye positions from our eye tracker can provide strong and reliable information about the rough location of face as well as the face motion between two consecutive frames. The robustness of the face tracker relies on the result of the eye tracker. Since our IR based eye tracker can robustly track the eyes under variable illuminations, the face location and motion information can be effectively inferred from the detected eyes for the face tracker. In general, our system can handle people with glasses [19] well when pupils are not occluded by the glares (highlights). But in certain occasions when the glares (highlights) on the glasses totally occlude the pupils in the image, our system may fail to detect the pupils.

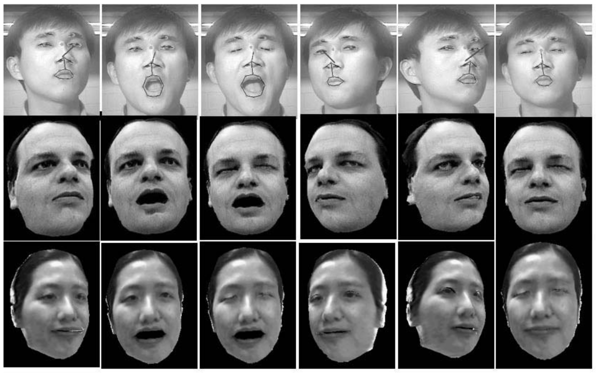


Figure 9: Example of the real time facial expression generation based on our active facial tracking system. 1st row: sample frames (#10, #30, #60, #90, #120, #150) are chosen from the original video sequence with detected features; 2nd row: head motion and expression cloned on a 3D male avatar; 3rd row: head motion and expression generated on a 3D female avatar

Real time tests have been conducted intensively for a large number of people who performed motions and expressions in front of the camera. Note that our algorithm requires all the facial features to be visible with respect to the camera for the effective detection. The facial features tracker can tolerate about +- 30 degrees for both tilt and pan angles. When face rotations are beyond these angles, some non-visible features may not be reliably tracked. In this situation, the distortion of face animation may occur due to the large rotation of the head and its non-trivial movement in the depth direction. However, as soon as the facial features become visible after the head moves back, our algorithm is able to grasp the features immediately, and consequently the animation parameters can be reliably derived. It is worth to mention that the system works in a fully automatic fashion without any human intervention, and it works for any person.

5.1 Performance analysis

(1) Speed: The real time generation of facial expressions relies on the high performance of the active face tracking system. Running on dual processors (Pentium-IV PC 1.9G), the 22 features are tracked at the speed of 15 frames per second. From the output of the tracking system, the 3D high resolution model (2997 vertices) can be animated in the speed of 33 frames per second. The high performance is achieved by the reduced number of control points, pre-defined control regions through a built-in lookup table, fast 3D vertex transformation and texture-mapping.

(2) Data rate: Since the animation system requires

[1] See real time demo at http://www.cs.binghamton.edu/∼lijun/Demo.html or http://www.ecse.rpi.edu/∼cvrl/Demo/demo.html

each frame to provide only twenty-two 2D feature points with extra information for pose/gaze orientation and eye closure percentage, the data rate for animating each frame is 64 bytes per frame. Given the animation speed from 15 frames/second to 30 frames/second, the data transmission rate ranges from 7.5K bits/s to 15K bits/s, which is extremely low as compared to the existing methods such as MPEG techniques that require at least 64K bits/s bitrate. Note that due to our advanced fine modeling technique, the system is well able to carry out the eye and mouth movement without creating separate models, as such the extra cost for both data and speed is reduced.

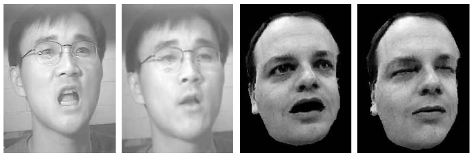


Figure 10: Left two: example of source subject with eye-glasses; Right two: expression generated on a 3D male avatar

.

6 Conclusions

The system is demonstrated to work well under various illumination conditions, even with reflections of eye glasses on subjects. Our experiments show that it is feasible to create face animation and expressions with very limited number of facial features for extremely low bit rate transmission. In order to increase the realism and the robustness, the teeth model and large motion detection in the depth direction will be developed in the future. To produce various subtle expressions, a real time face model instantiation process is expected to carry out as well such that the virtual avatar could be replaced by the sender's real face model. In addition, we will research on the error handling mechanism to ensure the animation to be smoothly transitted in the case of false tracking. For instance, the variance of animation data for each feature is to be monitored. if any unusual change (e.g., beyond certain threshold) happens, the tracked point will be flagged by a low confidence level and the current data will be discarded, eventually data from the previous frame is to be used instead. In the case of long time tracking loss due to some unavoidable disturbance, the system should be able to automatically reset to the initial condition and start over.

References

[1] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, "Synthesizing Realistic Facial Expressions from Photographs", *SIGGRAPH98*, pp 75-84.

[2] K. HIWADA, A. MAKI, A. NAKASHIMA, "A Real-Time Face Tracking System Based on Morphable 3D Model Fitting", ICCV, 2003

[3] Sami Romdhani, Thomas Vetter, "Efficient, Robust and Accurate Fitting of a 3D Morphable Model", ICCV, 2003

[4] Jun-yong Noh, Ulrich Neumann, "Expression Cloning", *SIGGRAPH01*, pp277-288.

[5] D. Terzopoulosa and K. Waters, "Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models." *IEEE Trans. PAMI*, 15(6), 1993

[6] Volker Blanz, Thomas Vetter, "A Morphable Model for the Synthesis of 3D Faces", *SIGGRAPH*, 1999.

[7] Haisong Gu, Qiang Ji, Zhiwei Zhu, "Active Facial Tracking for Fatigue Detection", *IEEE Workshop on Appl. of Computer Vision*, December , 2002

[8] Feng Jiao, Stan Li, Heung-Yeung Shum, Dale Schuurmans, "Face Alignment Using Statistical Models and Wavelet Features", *Proc. of IEEE CVPR*, 2003

[9] C. Tomasi and T. Kanade, "Detection and Tracking of point features". *Carnegie Mellon University Technical Report*, (CMU-CS-91-132), April 1991

[10] Matthew Turk, Changbo Hu, Rogerio Feris, Farshid Lashkari, Andy Beall, "TLA Based Face Tracking". *The 15th International Conference on Vision Interface* , May 2002

[11] Z. Zhang, "Feature-based facial expression recognition: Experiments with a multi-layer perception". *Technical report INRIA* ,(335),1998

[12] L. Yin and A. Basu, "Generating Realistic Facial Expressions with Wrinkles for Model Based Coding", *Computer Vision and Image Understanding*, 84(2): 201-240. Nov. 2001.

[13] Z.Zhu and Q. Ji, Kikuo Fujimura and Kuang-chih Lee, "Combining Kalman Filtering and Mean Shift for Real Time Eye Tracking Under Active IR Illumination", *International Conference on Pattern Recognition*, August 11-15, 2002.

[14] Q. Ji and R. Hu, "3D Face pose estimation and tracking from a monocular camera ", *Image and Vision Computing*, Volume 20, issue 7, pages 499-511, 2002.

[15] Haro, Antonio, M. Flickner and I. Essa, "Detecting and tracking eyes by using their physiological properties, dynamics and appearance", IEEE CVPR, page163-168, 2000.

[16] T. Lee, "Image representation using 2D Gabor wavelets", *IEEE Trans. PAMI*, 18(10):959-971, 1996

[17] H. Ip and L. Yin, "Constructing 3D Individualized Head Model From Two Orthogonal Views", *The Visual Computer - the International Journal in Computer Graphics*, 12(5):254-266, Springer-Verlag, 1996.

[18] Q. Ji, Z. Zhu, and P. Lan, "Real Time Non-intrusive Monitoring and Prediction of Driver Fatigue", to appear in *IEEE Transactions on Vehicular Technology*, 2004

[19] Z. Zhu, K. Fujimura, Q. Ji, "Real-Time Eye Detection and Tracking Under Various Light Conditions and Face Orientations", *2002 ACM SIGCHI Symposium on Eye Tracking Research & Applications*, New Orleans, LA, 2002.