EURASIP Journal on Applied Signal Processing 2005:13, 2091–2100

Spatio-Temporal Graphical-Model-Based Multiple Facial Feature Tracking

Congyong Su
College of Computer Science, Zhejiang University, Hangzhou 310027, China
Email: su@cs.zju.edu.cn

Li Huang
College of Computer Science, Zhejiang University, Hangzhou 310027, China
Email: lihuang@cs.zju.edu.cn

Received 1 January 2004; Revised 20 February 2005

It is challenging to track multiple facial features simultaneously when rich expressions are presented on a face. We propose a two-step solution. In the first step, several independent condensation-style particle filters are utilized to track each facial feature in the temporal domain. Particle filters are very effective for visual tracking problems; however multiple independent trackers ignore the spatial constraints and the natural relationships among facial features. In the second step, we use Bayesian inference—belief propagation—to infer each facial feature's contour in the spatial domain, in which we learn the relationships among contours of facial features beforehand with the help of a large facial expression database. The experimental results show that our algorithm can robustly track multiple facial features simultaneously, while there are large interframe motions with expression changes.

Keywords and phrases: facial feature tracking, particle filter, belief propagation, graphical model.

1. INTRODUCTION

Multiple facial feature tracking is very important in the computer vision field: it needs to be carried out before video-based facial expression analysis and expression cloning. Multiple facial feature tracking is also very challenging because there are plentiful nonrigid motions in facial features besides rigid motions in faces. Nonrigid facial feature motions are usually very rapid and often form dense clutter by facial features themselves. Only using traditional Kalman filter is inadequate because it is based on Gaussian density, and works relatively poorly in clutter, which causes the density for facial feature's contour to be multimodal and therefore non-Gaussian. Isard and Blake [1] firstly proposed a face tracker by particle filters—condensation—which is more effective in clutter than comparable Kalman filter.

Although particle filters are often very effective for visual tracking problems, they are specialized to temporal problems whose corresponding graphs are simple Markov chains (see Figure 1). There is often structure within each time instant that is ignored by particle filters. For example, in multiple facial feature tracking, the expressions of each facial feature (such as eyes, brows, lips) are closely related; therefore a more complex graph should be formulated.

The contribution of this paper is extending particle filters to track multiple facial features simultaneously. The straightforward approach of tracking each facial feature by one independent particle filter is questionable, because influences and actions among facial features are not taken into account.

In this paper, we propose a spatio-temporal graphical model for multiple facial feature tracking (see Figure 2). Here the graphical model is not a 2D or a 3D facial mesh model. In the spatial domain, the model is shown in Figure 3, where x^i is a hidden random variable and y^i is a noisy local observation. Nonparametric belief propagation is used to infer facial feature's interrelationships in a part-based face model, allowing positions and states of some features in clutter to be recovered. Facial structure is also taken into account, because facial features have spatial position constraints [2]. In the temporal domain, every facial feature forms a Markov chain (see Figure 1).

After briefly reviewing related work in Section 2, we introduce the details of our algorithm in Sections 3 and 4. Many convincing experimental results are shown in Section 5. Conclusions are given in Section 6.

2. RELATED WORK

After the pioneering work of Isard and Blake [1] who creatively used particle filters for visual tracking, many

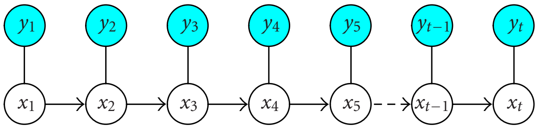


Figure 1: The Markov chain assumption of particle filters. The empty circle x_i represents the hidden state (contour) in time i, and the filled-in one y_i denotes the local observation.

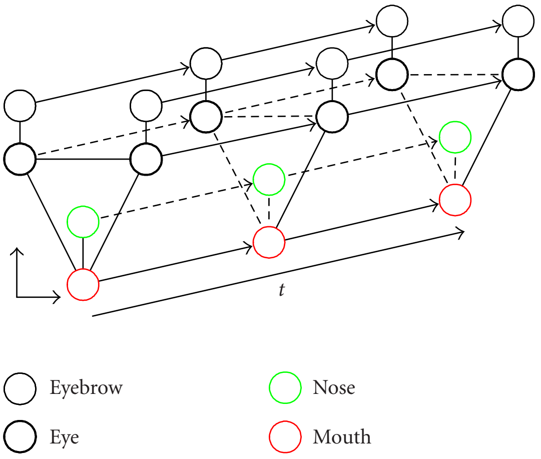


Figure 2: Tracking multiple facial features with a spatio-temporal graphical model. Each facial feature's state (contour) forms a Markov chain in the temporal domain, while facial features are related to each other in each time instant.

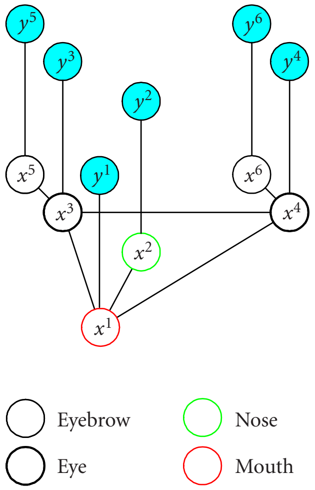


Figure 3: Markov network representation of a face in the spatial domain. x^1, x^2, $x^3(x^4)$, and $x^5(x^6)$ denote the contours of mouth, nose, eyes, and eyebrows, respectively.

researchers have adopted particle filters to track face or facial features [2, 3, 4, 5, 6, 7, 8].

Rui and Chen [3] used the unscented particle filter (UPF) [9] to do visual tracking. Zeng and Ma [4] proposed an active particle filtering approach. Vermaak et al. [5] selectively adapted the observation model to obtain better tracking results. Pèrez et al. [6] combined color-based CamShift or MeanShift algorithm with particle filters. Loy et al. [2] utilized multiple cues to track target. All of the above methods only used particle filters to track the whole face or head, not the facial features. De la Torre et al. [7] used particle filters to track eyes or lips while switching between different shape/texture models; however they didn't track both simultaneously. Wang et al. [8] integrated a learned intrinsic object structure into a particle-filter style tracker; however only one facial feature—mouth—was tracked. Therefore the idea of this paper is very new. We use particle filters to track multiple facial features rather than one facial feature.

Isard [10] and Sudderth et al. [11] have independently developed an algorithm for performing belief propagation with the aid of particle sets. Their methods motivated us to use graphical model in multiple facial feature tracking. However they only show their algorithms' effectiveness in 2D graphical models, which are in the spatial domain. As far as multiple facial feature tracking is concerned, the corresponding graphical model is a 3D one, which is spatio-temporal. The 3D graphical model belongs to a specific type, and is directed-cum-undirected. In this paper, we try to seek the relationships between the particle filter in the 1D temporal domain and nonparametric belief propagation in the 2D spatial domain.

Facial feature tracking has also been extensively studied by other methods [12, 13, 14, 15, 16, 17, 18, 19], such as optical flow [20] based [12, 13, 14], ASM/AAM based [15, 16], model-less based [17], infrared camera based [18], and so forth. However, in this paper, the particle-filter-based approach is preferred for performing multiple facial feature tracking.

3. MULTIPLE FACIAL FEATURE TRACKING BY PARTICLE FILTER: THE FIRST STEP

We adopt the condensation algorithm to track each facial feature. After Isard and Blake [1] first proposed an implementation of particle filters, many other researchers also proposed enhanced algorithms for particle filters, for example, Icondensation [21], UPF [9], and the Rao-Blackwellised particle filter [22]; however they still could not solve the "curse of dimensionality" problem, and generally the workable dimensionality was below 10. On the other hand, we should break down the tracking result of particle filters in the spatial domain. Therefore the choice of different particle filters has no key effect on our algorithm. For simplicity, we choose the basic condensation algorithm because it can satisfy our requirements.

Six facial features are tracked in this paper. They are eyebrows, eyes, nose, and mouth. Taking eye for example, we track the eyelid contour. The contour is modeled as B-spline $X_t = x_1, x_2, \ldots, x_t$, and the observation of eye is $Y_t = y_1, y_2, \ldots, y_t$. We need to infer the marginal conditional density $p(x_t|Y_t)$. Isard and Blake [23] have proved that

$$p(x_t|Y_t) = p(x_t|y_t, Y_{t-1}) = c_t p(y_t|x_t) p(x_t|Y_{t-1}), \quad (1)$$

where c_t is a constant, and

$$p(x_t|Y_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|Y_{t-1}) dx_{t-1}. \quad (2)$$

In (1), $p(x_t|Y_{t-1})$ is the effective prior model, and $p(y_t|x_t)$ is the observation model. In (2), $p(x_t|x_{t-1})$ is the dynamic model.

3.1. Why several particle filters?

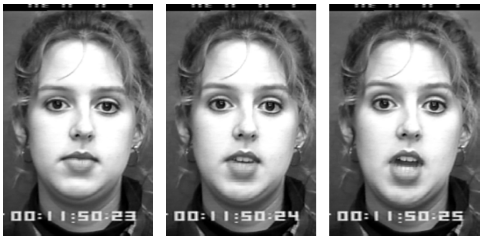
Single particle filter is not suitable to track multiple facial features simultaneously. The reason is as follows: the total dimensionality added by each feature's dimensionality is too high (dozens); the tracking efficiency of the particle filter decreases exponentially along with the linear increasing of dimensionality. Usually, it is extremely difficult to get good results from particle filters in spaces of dimensionality much greater than about 10 [24]. Even if dimensionality can be reduced by principal components analysis (PCA) [25] or other nonlinear methods [8, 26], the total dimensionality of multiple facial features is significantly large. If we reduce the dimensionality too much, valuable state information may be lost.

A human face contains multiple facial features, and it can be decomposed into several parts, such as eyebrows, eyes, nose, and mouth, to form a graphical model in the spatial domain. In this paper, we track each facial feature by its corresponding particle filter, therefore computational complexity is converted from exponential to linear with the size of the graph.

3.2. Particle filter itself is not enough

When there are rapid motions in one facial feature (e.g., mouth) due to the changes of facial expressions (see Figure 4), the corresponding particle filter may fail to track the facial feature's contour. It is difficult to reduce this failure if we only use multiple independent particle filters to track each facial feature. In this paper, we track several facial features simultaneously through using several correlated particle filters. When emotion is presented on the face, different facial features have natural physical interaction. For example, when we smile with blinking the left eye, our left mouth tip will move up; when we surprise with the wide-open mouth, the eyebrows will also move up.

Instead of constructing heuristic rules for these relationships, we learn the relationships among facial features from training data beforehand. During the process of tracking, if we detect that some facial feature tracker's results are poor, we can infer their positions and states from other facial features by Bayesian inference. In this paper, belief propagation is used to carry out Bayesian learning and inference.



(a) (b) (c)

Figure 4: Three consecutive frames at 30 fps show that facial feature motions are rapid.

4. COMBINING PARTICLE FILTER WITH BELIEF PROPAGATION: THE SECOND STEP

4.1. Loopy belief propagation

In every time instant, facial features are contained in an undirected graphical model G_f (see Figure 3). Let V denote the set of nodes (facial features). Nodes are connected by edges E to describe the relationship between facial features. The neighborhood of a node i is $\mathrm{NB}(i) = \{j|(i,j) \in E\}$. Let x^i denote the hidden variable (contour of facial feature), and let y^i denote the observed variable (facial feature image). Let $\{x^i\} = \{x^i|1 \le i \le N\}$ and $\{y^i\} = \{y^i|1 \le i \le N\}$, where N is the number of nodes in the graphical model G_f. The joint probability density function factorizes as

$$p(\{x^i\},\{y^i\}) = \frac{1}{C} \prod_{(i,j)\in E} \psi_{ij}(x^i,x^j) \prod_{i\in V} \phi_i(x^i,y^i), \quad (3)$$

where C is a normalization constant, and $\psi_{ij}(x^i,x^j)$ and $\phi_i(x^i,y^i)$ are compatibility functions. $\psi_{ij}(x^i,x^j)$ is a correlation function between x^i and its neighbor variable x^j, and $\phi_i(x^i,y^i)$ is an observation function that denotes the evidence of x^i [27].

From Figure 3, we can see that it is a Markov network with loops. Pearl [28] warned that belief propagation might not converge in this kind of graphical models. However, some experimental [29] and theoretical results [30, 31, 32, 33] motivate us to apply the belief propagation rules in the Markov network with loops, and Murphy et al. called it loopy belief propagation [31].

In belief propagation, we need to calculate the conditional marginal distribution $p(x^i|\{y^i\})$ for the nodes that have less confidence in the tracking results by particle filters. The intuitive meaning is that we can infer the facial feature i's position (contour) and state (e.g., expression) from all facial features' observations $\{y^i\}$.

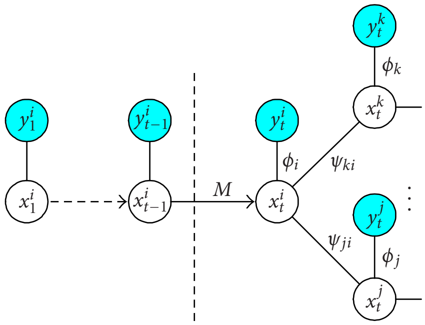


Figure 5: Message passing in a directed-cum-undirected graphical model.

4.2. *Belief propagation in spatio-temporal graphical model*

In this paper, the graphical model is the combination of directed graph (Markov chain) and undirected graph (Markov network). In order to do Bayesian inference, the key point is belief propagation or message passing.

The messages of directed graph are passing through the time axis. In Figure 5, the message passing from x_{t-1}^i to x_t^i is denoted by $M(x_{t-1}^i \rightarrow x_t^i)$. We have

$$M(x_{t-1}^i \longrightarrow x_t^i) = p(x_t^i | \{Y_{t-1}^i\}), \tag{4}$$

where $\{Y_{t-1}^i\} = \{Y_{t-1}^i | 1 \leq i \leq N\}$, and

$$p(x_t^i | \{Y_{t-1}^i\}) = \int_{x_{t-1}^i} p(x_t^i | x_{t-1}^i) b(x_{t-1}^i) dx_{t-1}^i. \tag{5}$$

$b(x_t^i)$ is the conditional marginal probability distribution in node x_t^i, and it is what we have to calculate. $b(x_{t-1}^i)$ means the tracking result in facial feature i by graphical model in the previous time instant. The belief at node (i, t) is proportional to the product of the local evidence $\phi_i(x_t^i, y_t^i)$ at that node and all the messages coming into it [34]. There are two kinds of messages: one comes from the immediate preceding node x_{t-1}^i temporally, and the other is from the neighbors of node (i, t) spatially. Therefore, we have

$$b(x_t^i) = K\phi_i(x_t^i, y_t^i) M(x_{t-1}^i \longrightarrow x_t^i) \prod_{j \in \mathrm{NB}(i,t)} m_{ji}(x_t^i). \tag{6}$$

In (6), K is a normalization constant and $\mathrm{NB}(i, t)$ denotes the nodes neighboring the node (i, t). As defined in (4) and (5), the message from the previous time is $M(x_{t-1}^i \rightarrow x_t^i)$. Furthermore, the message from the spatial neighbor is determined self-consistently by the following message update rule:

$$\begin{aligned} m_{ji}(x_t^i) = \alpha \int_{x_t^j} & \psi_{ji}(x_t^j, x_t^i) \phi_j(x_t^j, y_t^j) M(x_{t-1}^j \longrightarrow x_t^j) \\ & \times \prod_{k \in \mathrm{NB}(j,t) \setminus (i,t)} m_{kj}(x_t^j) dx_t^j. \end{aligned} \tag{7}$$

In the right-hand side of (7), we take the product of messages going into node (j, t) except for the one coming from node (i, t). Note that the message $M(x_{t-1}^j \rightarrow x_t^j)$ from the previous time instant is also taken into account.

Based on a factorization described by [27], we use the observation function $\phi_i(x_t^i, y_t^i) = p(y_t^i | x_t^i)$, and it can be seen that $\phi_i(x_t^i, y_t^i)$ is equal to the observation model in [23]. We also use the correlation function $\psi_{ji}(x_t^j, x_t^i) = p(x_t^j, x_t^i)/p(x_t^i)$, and fit this probability with mixtures of Gaussians [35].

The message $M(x_{t-1}^i \rightarrow x_t^i)$ passing from x_{t-1}^i to x_t^i can be viewed as the effective prior: a prediction taken from the marginal probability $b(x_{t-1}^i)$ from the previous time step, onto which is superimposed one time step from the dynamical model.

From (6) and (7), we can see that ϕ always comes with M. By the analysis above, the product of them is

$$\begin{aligned} \phi_i(x_t^i, y_t^i) & M(x_{t-1}^i \longrightarrow x_t^i) \\ &= p(y_t^i | x_t^i) p(x_t^i | \{Y_{t-1}^i\}) \\ &= \frac{1}{c_t} p(x_t^i | y_t^i, \{Y_{t-1}^i\}) \quad (\text{using } (1)). \end{aligned} \tag{8}$$

Equation (8) means that the product is effectively the posterior probability of x_t^i conditioned on y_t^i and $\{Y_{t-1}^i\}$, and this shares the same idea with the condensation algorithm. This property is important because it allows us to firstly run the particle filter to track each facial feature in one time step, and the output of particle filter is naturally fitted into a loopy belief propagation process (see (6) and (7)).

Wu et al. [36] proposed a mean-field Monte Carlo algorithm for visual tracking of articulated human body, which is similar to ours in using the dynamic Markov network.

4.3. *Particle propagation in spatio-temporal graphical model*

Since in our spatio-temporal graphical model, messages are not needed to pass backward in the temporal domain, therefore the choice of importance function can be omitted.

In conventional particle filter algorithms, the probability distribution of possible interpretations is represented by a randomly sampled set, which can be called "particle set."

In particle set form, our algorithm is also the combination of particle filter and loop belief propagation as indicated by (8).

Each sample is a $(s_t^i(m), \pi_t^i(m))$ pair, in which $s_t^i(m)$ is a value of $x_t^i(m)$ and $\pi_t^i(m)$ is a corresponding sampling probability. $m \in [1, M]$, and M is the total number of samples for one facial feature.

Step 1. Firstly, a particular $s_{t-1}^i(m)$ is drawn randomly from $b_{t-1}^i(m)$ by choosing it with probability $\pi_{t-1}^i(m)$ from the set of M samples at time $t-1$.

Step 2. Draw $s\_\mathrm{pf}_t^i(m)$ randomly from $p(x_t^i | x_{t-1}^i = s_{t-1}^i(m))$, one time step of the dynamic model, where pf denotes the particle filter.

Step 3. A value $s\_\mathrm{pf}_t^i(m)$ chosen in Step 2 is a fair sample from $p(x_t^i | \{Y_{t-1}^i\})$. Set $\pi\_\mathrm{pf}_t^i(m) = \phi_i(x_t^i = s\_\mathrm{pf}_t^i(m), y_t^i)$, therefore we obtain the particle set form of $\phi_i(x_t^i, y_t^i)M(x_{t-1}^i \to x_t^i) \equiv LL(x_t^i)$, which can be viewed as a likelihood function in belief propagation.

Actually, $LL(x_t^i)$ is the tracking result of particle filter for one facial feature, since we have

$$\begin{aligned} LL(x_t^i) &\equiv \phi_i(x_t^i, y_t^i)M(x_{t-1}^i \longrightarrow x_t^i) \\ &= \frac{1}{c_t} p(x_t^i | y_t^i, \{Y_{t-1}^i\}) \quad \text{(using (8))}. \end{aligned} \tag{9}$$

Using the sampling method similar to conventional particle filter (as described in Steps 1, 2, and 3), we can obtain a nonparametric approximation $(s\_\mathrm{pf}_t^i(m), \pi\_\mathrm{pf}_t^i(m))$ to $LL(x_t^i)$. We can further use a bandwidth selection method to construct a kernel density estimate $\widetilde{LL}(x_t^i)$ from $(s\_\mathrm{pf}_t^i(m), \pi\_\mathrm{pf}_t^i(m))$; therefore we can evaluate it in nonparametric belief propagation.

Step 4. Let $p_{ji}^{\mathrm{msg}}(x_t^j)$ denote the foundation of message $m_{ji}(x_t^i)$ as follows:

$$\begin{aligned} p_{ji}^{\mathrm{msg}}(x_t^j) \equiv{} & K_j \phi_j(x_t^j, y_t^j) M(x_{t-1}^j \longrightarrow x_t^j) \\ & \times \prod_{k \in \mathrm{NB}(j,t) \setminus (i,t)} m_{kj}(x_t^j), \end{aligned} \tag{10}$$

where K_j is a constant which makes $p_{ji}^{\mathrm{msg}}(x_t^j)$ a probability density.

Step 5. Draw M samples

$$s\_bp_t^j(m) \sim K_j \prod_{k \in \mathrm{NB}(j,t)} m_{kj}(x_t^j). \tag{11}$$

In order to obtain the integral of (7), we should compute a weight for each sample:

$$\begin{aligned} & \pi\_bp_t^j(m) \\ & \quad \propto \frac{p_{ji}^{\mathrm{msg}}(s\_bp_t^j(m))}{\prod_{k \in \mathrm{NB}(j,t)} \tilde{m}_{kj}(s\_bp_t^j(m))} \\ & \quad \propto \frac{\phi_j(x_t^j, y_t^j) M(x_{t-1}^j \longrightarrow x_t^j)|_{x_t^j = s\_bp_t^j(m)}}{\tilde{m}_{ij}(s\_bp_t^j(m))} \\ & \quad \propto \frac{\widetilde{LL}(s\_bp_t^j(m))}{\tilde{m}_{ij}(s\_bp_t^j(m))}. \end{aligned} \tag{12}$$

where p_{ji}^{msg} is defined in (10), and $\tilde{m}_{ij}$ is obtained from the message update in the last iteration. $\phi_j(x_t^j, y_t^j)M(x_{t-1}^j \to x_t^j)$ is the result of temporal filter for each facial feature, and we use it to calculate sample weights of message $m_{ji}(x_t^i)$ in (7) for nonparametric belief propagation.

In (12), although $\widetilde{LL}(x_t^j)$ is in particle set form, it still can be evaluated.

For iterations of message passing, the procedure is initialized with all messages set to constant values.

Step 6. The approximation of message $m_{ji}(x_t^i)$ is obtained by

$$\begin{aligned} & \tilde{m}_{ji}(x_t^i) \\ & \quad = \frac{1}{\sum_{m=1}^{M} \pi\_bp_t^j(m)} \sum_{m=1}^{M} \left(\pi\_bp_t^j(m) \times \psi_{ji}(s\_bp_t^j(m), x_t^i)\right). \end{aligned} \tag{13}$$

Step 7. Generally, after several iterations of message passing, the belief distribution has converged. We should obtain the marginal estimate for $b(x_t^i)$ in (6) to get the final results.

Given the input messages $\tilde{m}_{ji}(x_t^i)$ from the spatial neighbors $\mathrm{NB}(i,t)$,

(1) draw M independent samples $s_t^i(m)$, $m \in [1, M]$, from the product

$$s_t^i(m) \sim K \prod_{j \in \mathrm{NB}(i,t)} \tilde{m}_{ji}(x_t^i); \tag{14}$$

(2) compute the weight for each sample $s_t^i(m)$:

$$\begin{aligned} \pi_t^i(m) &\propto \phi_i(x_t^i, y_t^i) M(x_{t-1}^i \longrightarrow x_t^i)|_{x_t^i = s_t^i(m)} \\ &\propto \widetilde{LL}(s_t^i(m)). \end{aligned} \tag{15}$$

We also use the result $\widetilde{LL}(x_t^i)$ of particle filter in this step besides Step 5. Therefore our algorithm combines temporal particle filters with spatial belief propagations.

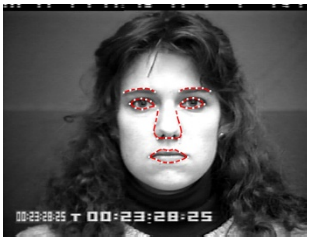


Figure 6: Six facial features are described by quadric B-splines.

For sampling $s\_bp_t^j(m)$ in Step 5 and $s_t^i(m)$ in Step 7, we use a similar method to [10]. Sampling from the product can be decomposed into two steps: randomly select one of the product density's components and then draw a sample from the corresponding Gaussian.

The algorithm in this paper is summarized in the above steps.

4.4. Learning the correlation function

In the training database, we manually mark some face's features; therefore we obtain the ground-truth position of the contour x^i. First we reduce the dimensionality of facial feature i's contour x^i by PCA. Then from the training data, we fit mixtures of Gaussians to $p(x_t^i)$ and the joint probabilities $p(x_t^j, x_t^i)$ for neighboring facial feature i and j. We evaluate $p(x_t^j | x_t^i) = p(x_t^j, x_t^i)/p(x_t^i)$, therefore the correlation function $\psi_{ji}(x_t^j, x_t^i)$ is obtained.

4.5. Optimizing Bayesian inference for Markov network

Considering that Bayesian inference using belief propagation costs substantial time, we only initiate it when the particle filter's tracking result is poor.

For the corresponding particle filter on one facial feature, the tracking result on time t can be described by the moments [1]:

$$E(f(x_t) | Y_t) = \sum_{m=1}^{M} \pi_t^{(m)} f\left(s_t^{(m)}\right). \tag{16}$$

As in [1], a mean position using $f(x_t) = x_t$ can be utilized for graphical display. Moreover, let $f(x_t) = x_t x_t^T$, and we obtain the variance $\sigma_t = E(x_t x_t^T | Y_t)$ of the tracking result. We use the variance σ_t as a metric of the tracking quality.

For each facial feature, we have σ_t^i, $i = 1, \dots, N$, where N is the number of all facial features (in this paper, it is 6). For the facial features that have larger variances, we determine that their tracking results are worse than others. Therefore belief propagation is carried out to infer the more plausible positions of their contours. Based on experimental results, if the $\sigma^i > 0.5 * \text{Area}(x^i)$, we consider that the tracking result on facial feature i is bad, where $\text{Area}(x^i)$ denotes the pixels occupied by facial feature i in the video stream. In implementation, the $\text{Area}(x^i)$ is obtained by computing the bounding box for the facial feature i.

5. EXPERIMENTAL RESULTS

We have developed a prototype system on Windows platform using Visual C++ to implement the algorithm in this paper. There are 6 contour models for facial features: eyebrows, eyes, nose, and mouth. Each contour is a quadric B-spline curve, in which contours of nose and eyebrows are open curves, and others are closed curves. As shown in Figure 6, there are 6, 9, 12, 12 control points for left (right) eyebrow, left (right) eye, nose, and mouth, respectively. The total number of control points is 54; therefore the dimensionality is 108.

We choose Cohn-Kanade [37] facial expression database as the training set, because it contains plenty of frontal faces with rich facial expressions. This database is stored as 30 fps grayscale image sequences. To learn the relationships among facial features, we have selected 496 frame frontal face images, which belong to 98 different persons, and used interactive program to mark each facial feature's contours. PCA is used to reduce the dimensionality for each facial feature's contour. After that, the dimensionality of left (right) eyebrow, left (right) eye, nose, and mouth is 4, 7, 9, and 9, respectively; therefore the total dimensionality after dimension reduction is 40, accounting for 99% of the total variance.

After constructing the PCA bases, we compute the corresponding PCA coefficients for each individual in the training set. For each of facial feature's contour pairs connected by edges in Figure 3, we determine kernel-based nonparametric density estimates for each node itself $p(x_t^i)$ and their joint probabilities $p(x_t^j, x_t^i)$. Figure 7 shows several marginalizations of $p(x_t^j, x_t^i)$, each of which relates a single pair of PCA coefficients (e.g., the first mouth and second left eye contour's coefficients). We can see that simple Gaussian approximations would lose most of this data set's meaningful structure.

Using the similar method in [23], we have also trained the dynamic model for each facial feature. For observation model, a set of independent measurement lines that are perpendicular to the hypothesized contour are used to measure the likelihood of detected edge points.

Using a single condensation tracker with multiple contours to track multiple facial features is infeasible because the dimensionality is much higher than 10. Here we compare our results with those of multiple independent condensation-style trackers. We have tested our algorithm on the image sequences in Cohn-Kanade database and the videos (640 × 480, 30 fps) that we captured by a digital video camera. The test image sequences are not included in the training database.

As stated in Section 4.2, our algorithm can be viewed as conventional condensation tracker plus contour adjustment by belief propagation. The experimental results (see

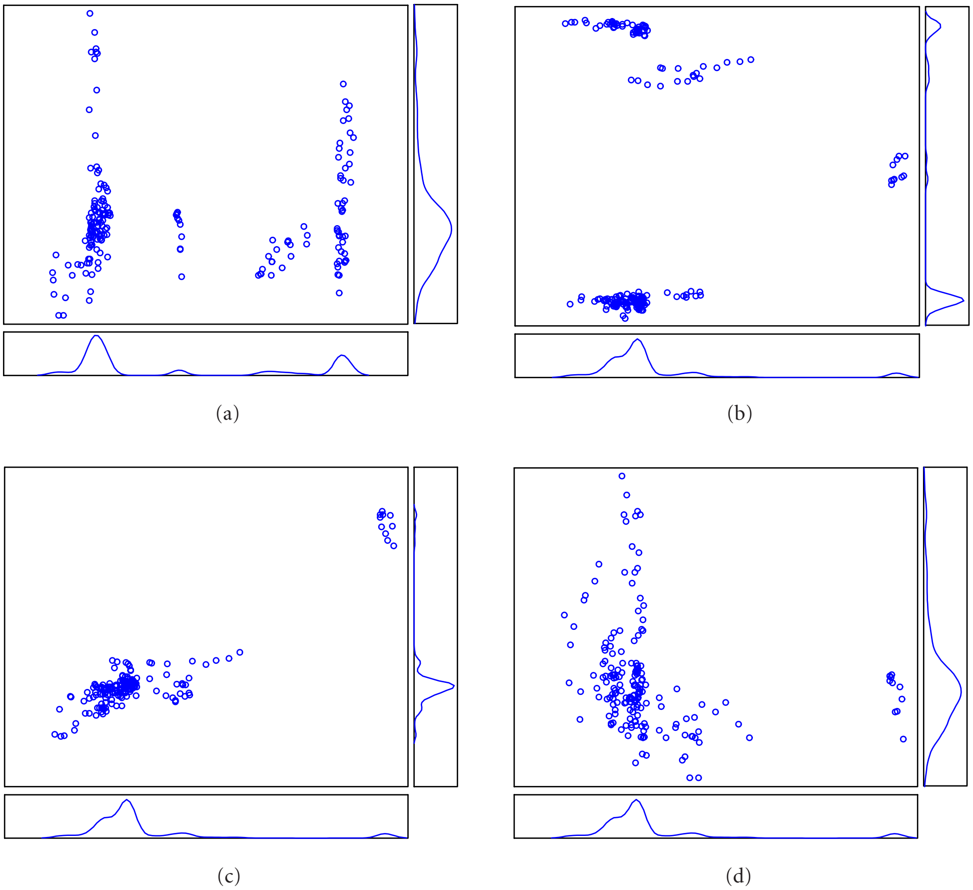


FIGURE 7: Joint density of four different pairs of PCA coefficients. It can be seen that the marginal distributions are multimodal.

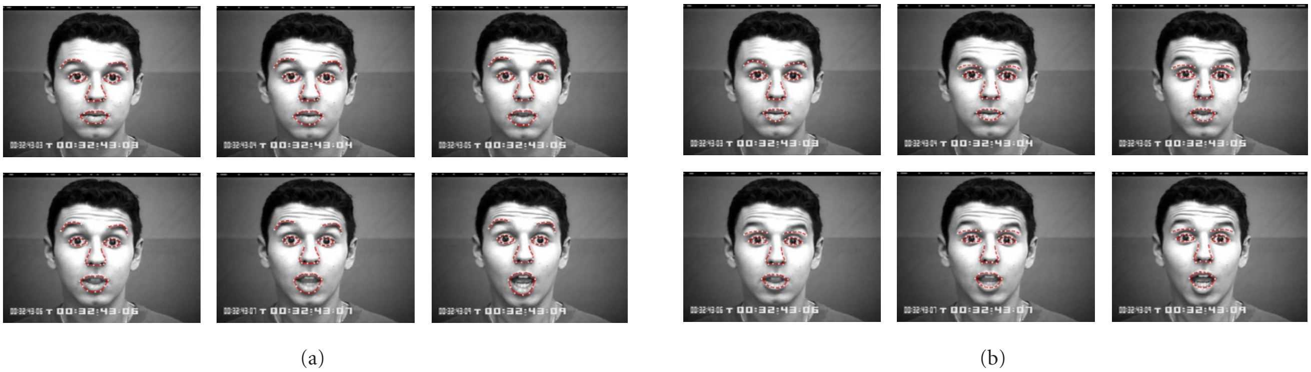
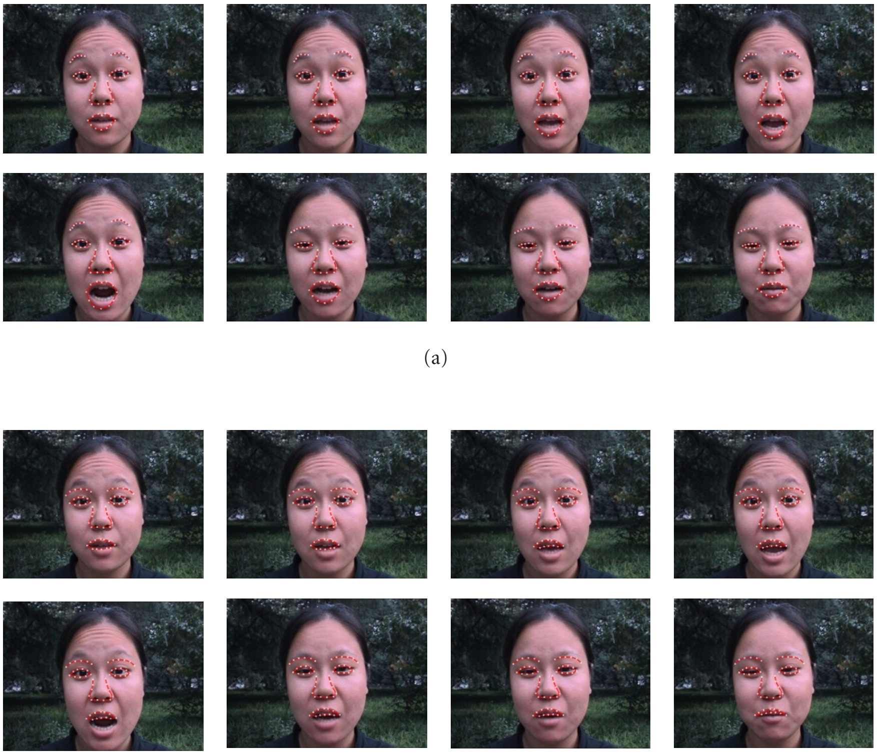


FIGURE 8: Tracking results of a surprise sequence. (a) Our algorithm correctly tracks the eyebrows and mouth. (b) The dark circles and teeth distract the MICT tracker; therefore it fails to track them.

Figures 8, 9, and 10) show that our tracker is more robust than multiple independent condensation-style trackers (MICT).

In Figure 10, we also compare our algorithm's results with those of active appearance model (AAM). AAM [15] is based on face alignment, and we use the same training set as MICT to train the active appearance model. From Figure 10, we can see that AAM fails to track mouth in the case of occlusion. The advantage is that our algorithm is more accurate than AAM, while the drawback is that our algorithm is slightly slower than AAM. Our algorithm is more robust than AAM, since even the particle filter fails to track the mouth, the mouth's location, and state can be inferred from the spatial domain by belief propagation. For AAM, it is difficult to



(a)

(b)

Figure 9: Three consecutive facial expressions: neutral, surprise, and happy. From the first row to second row, and left to right, frame numbers are 320, 322, 323, 324, 353, 355, 356, 358. (a) Our results. (b) MICT's results.

incorporate negative samples (e.g., occlusion) into its training set; therefore AAM performs badly when facial feature occlusion happens.

For all the testing image sequences, our algorithm obtains better results than those of MICT and AAM. For the experimental results shown in Figures 8, 9, and 10, the image sequences have 116, 368, and 900 frames, respectively.

Our tracker runs at about 3 Hz, the MICT tracker runs at about 4 Hz, and the AAM tracker runs at about 3.5 Hz on a Pentium 4 1.8 GHz 256 MB RAM computer.

6. CONCLUSIONS

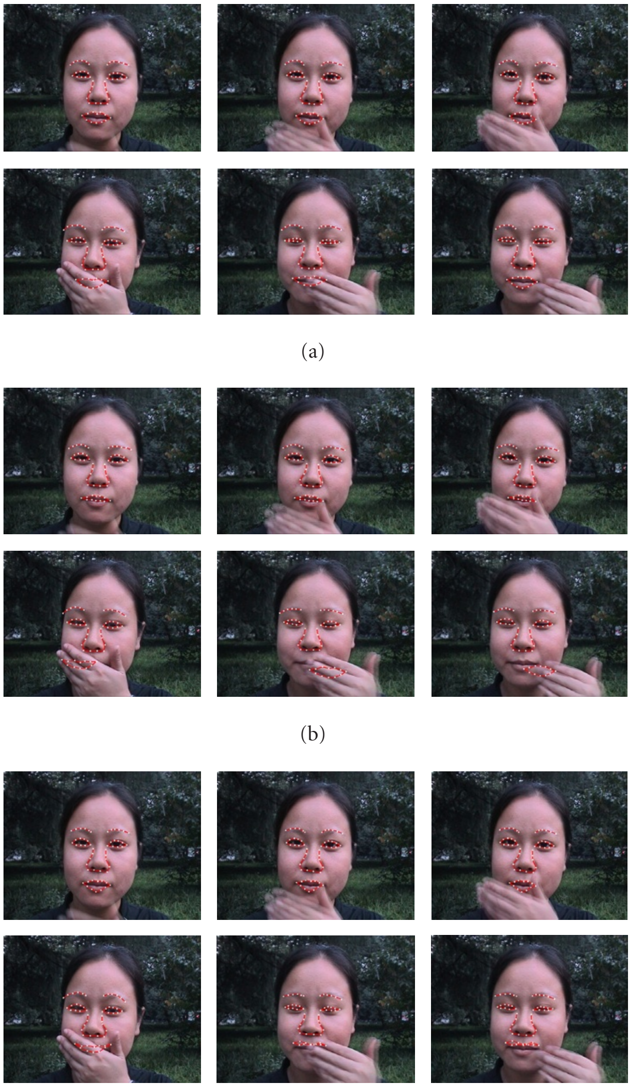
In this paper, we extend the particle filter from the relatively simple Markov chain to the directed-cum-undirected graphical model applied to multiple facial feature tracking problem. Spatial structure information and relationships among nodes in each time instant are effectively considered by Bayesian learning and inference in the loopy belief propagation framework.

The advantages of our algorithm are as follows. Compared with particle filters, we extend conventional particle filters to track multiple facial features simultaneously by exploring the spatial coherence in each time step, and the complexity of tracking is linear rather than exponential in the number of facial features. Compared with AAM, our algorithm is more robust.

The tracking results in this paper can be used as motion capture data. We plan to use these data to derive a 3D face model and generate facial animations. The ultimate purpose of multiple facial feature tracking is for facial animation.

Currently, the tracking results are 2D control points of B-splines in each time instant. In the future, we will use these results as video-based motion capture data. Using performance-driven facial animation techniques, we can obtain 3D facial animation of the tracked human face from 2D mocap data. Finally we will retarget animation from human faces to other virtual avatars.

While our current results are promising, details of our implementation could be improved. The spatial and temporal relationships among facial features can also be learnt by recently proposed spatio-temporal manifold learning algorithms. Currently, the testing image sequences are obtained by a fixed camera. For greater applicability, the method should be extended to allow a moving camera. Our tracking algorithm will fail if the face is too small in the video stream, and MICT and AAM will fail too. Maybe some face video super-resolution techniques can solve this problem.



(a)

(b)

(c)

Figure 10: Comparison results of hiding mouth. Frame numbers are 802, 803, 805, 810, 871, 872. (a) Our algorithm can successfully predict the contour of mouth. (b) MICT algorithm failed to track the contour of mouth, which is distracted by the moving hand. (c) AAM algorithm failed too.

ACKNOWLEDGMENT

We would like to thank Dr. Cohn for kindly providing us with the Cohn-Kanade facial expression database.

REFERENCES

[1] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. 4th European Conference on Computer Vision (ECCV '96)*, vol. 1, pp. 343–356, Cambridge, UK, April 1996.

[2] G. Loy, L. Fletcher, N. Apostoloff, and A. Zelinsky, "An adaptive fusion architecture for target tracking," in *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, pp. 248–253, Washington, DC, USA, May 2002.

[3] Y. Rui and Y. Chen, "Better proposal distributions: object tracking using unscented particle filter," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 2, pp. 786–793, Kauai Marriott, Hawaii, USA, December 2001.

[4] Z. Zeng and S. Ma, "Head tracking by active particle filtering," in *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, pp. 82–87, Washington, DC, USA, May 2002.

[5] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: selective adaptation," in *Proc. European Conference on Computer Vision (ECCV '02)*, vol. 1, pp. 645–660, Copenhagen, Denmark, May 2002.

[6] P. Pèrez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. European Conference on Computer Vision (ECCV '02)*, pp. 661–675, Copenhagen, Denmark, June 2002.

[7] F. De la Torre, Y. Yacoob, and L. Davis, "A probabilistic framework for rigid and non-rigid appearance based tracking and recognition," in *Proc. 4th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '00)*, pp. 491–498, Grenoble, France, March 2000.

[8] Q. Wang, G. Xu, and H. Ai, "Learning object intrinsic structure for robust visual tracking," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FGR '03)*, vol. 2, pp. 227–233, Madison, Wis, USA, June 2003.

[9] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 584–590, Denver, Colo, USA, November 2000.

[10] M. Isard, "Pampas: real-valued graphical models for computer vision," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 613–620, Madison, Wis, USA, June 2003.

[11] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 605–612, Madison, Wis, USA, June 2003.

[12] T. D. I. Essa, S. Basu, T. Darrell, and A. Pentland, "Modeling, tracking, and interative animation of faces and heads using input from video," in *Proceedings of Computer Animation Conference*, pp. 68–79, Geneva, Switzerland, June 1996.

[13] Y. Yacoob and L. S. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 6, pp. 636–642, 1996.

[14] M. J. Black and Y. Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion," in *Proc. 5th International Conference on Computer Vision (ICCV '95)*, pp. 374–381, Boston, Mass, USA, June 1995.

[15] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[16] J. Ahlberg, "An active model for facial feature tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 6, pp. 566–571, 2002.

[17] L. Torresani and C. Bregler, "Space-time tracking," in *Proc. European Conference on Computer Vision (ECCV '02)*, pp. 801–812, Copenhagen, Denmark, May 2002.

[18] A. Kapoor and R. W. Picard, "Real-time, fully automatic upper facial feature tracking," in *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, pp. 8–13, Washington, DC, USA, May 2002.

[19] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 2, pp. 97–115, 2001.

[20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application in stereo vision," in *Proc. 7th International Joint Conference of Artificial Intelligence (IJCAI '81)*, pp. 674–679, Vancouver, British Columbia, Canada, April 1981.

[21] M. Isard and A. Blake, "ICONDENSATION: unifying low-level and high-level tracking in a stochastic framework," in *Proc. European Conference on Computer Vision (ECCV '98)*, vol. 1, pp. 893–908, Freiburg, Germany, June 1998.

[22] A. Doucet, N. de Freitas, K. P. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic bayesian networks," in *Proc. 16th Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pp. 176–183, Stanford, Calif, USA, June 2000.

[23] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[24] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, New York, NY, USA, 2002.

[25] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[26] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[27] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[28] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, Calif, USA, 1988.

[29] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Limits on super-resolution and how to break them," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[30] E. C. Liu and J. M. F. Moura, "Fusion in sensor networks: convergence study," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 865–868, Montreal, Quebec, Canada, May 2004.

[31] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in *Proc. Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 467–475, Stockholm, Sweden, July 1999.

[32] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[33] J. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 689–695, Denver, Colo, USA, November 2000.

[34] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalization," Tech. Rep. TR2001-22, MERL, Cambridge, Mass, USA, 2001.

[35] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[36] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic Markov network," in *Proc. 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1094–1101, Nice, France, October 2003.

[37] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FGR '00)*, pp. 46–53, Grenoble, France, March 2000.

Congyong Su was born in Xinyang, China, in 1979. He received the B.S. and M.S. degrees from Zhejiang University in 1999 and 2002, respectively, both in electrical engineering. Currently he is a Ph.D. candidate in the College of Computer Science, Zhejiang University, and he will graduate in March 2005. His research interests include pattern recognition, computer vision, image/video understanding, and computer facial animation.

Li Huang was born in Qidong, China, in 1979. She received the B.S. and M.S. degrees in computer science from Zhejiang University in 2001 and 2004, respectively. Her research interests include computer facial animation, pattern recognition, image/video understanding, and multimedia analysis.