Electronic Letters on Computer Vision and Image Analysis 5(3):157-169, 2005

Bayesian Network Enhanced Prediction for Multiple Facial Feature Tracking

Li Huang and Congyong Su

College of Computer Science, Zhejiang University, Hangzhou 310027, China

Received 20 December 2004; accepted 27 July 2005

Abstract

It is challenging to track multiple facial features simultaneously in video while rich facial expressions are presented in a human face. To accurately predict the positions of multiple facial features' contours is important and difficult. This paper proposes a multi-cue prediction model based tracking algorithm. In the prediction model, CAMSHIFT is used to track the face in video in advance, and facial features' spatial constraint is utilized to roughly obtain the positions of facial features. Second order autoregressive process (ARP) based dynamic model is combined with graphical model (Bayesian network) based dynamic model. Incorporating ARP's quickness into graphical model's accurateness, we obtain the fusion of the prediction. Finally the prediction model and the measurement model are integrated into the framework of Kalman filter. The experimental results show that our algorithm can accurately track multiple facial features with varied facial expressions.

Key Words: Multiple Facial Feature Tracking, Bayesian Network, Graphical Model, CAMSHIFT.

1 Introduction

Multiple facial feature tracking is challenge in computer vision domain. It can be applied in many areas, such as facial feature extraction, facial expression retargeting and human computer interaction, etc. Kass *et al.* [1] proposed Snakes: Active Contour model to track the contour of lips. Snakes are energy-minimizing splines guided by constraints. It can be used to obtain smooth feature contours. But when the total number of control points is large (e.g., dozen), the dimensionality is too high to track contours efficiently. Furthermore, to allow arbitrary variation in positions of control points over time will lead to instability in tracking. Cootes *et al.* [2][3][4] proposed the ASM/AAM algorithms, in which tracking is based on face detection and recognition. However the tracking results depend on the model's initial position and the variations contained in the training set, which makes it difficult to deal with occlusions. Furthermore, during the training, broken line is used to mark the facial features, which is not smooth.

We propose a Bayesian network enhanced prediction model based multiple facial feature tracking algorithm. Our considerations for tracking are as follows:

(1) Choose the B-spline to describe feature's contour. B-spline is smooth. It is better than broken line, since the contour of facial feature is smooth too.

Correspondence to: su@cs.zju.edu.cn

Recommended for acceptance by <Perales F. and Draper B.>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

(2) Similar to dimensionality reduction in AAM algorithm, we utilize Principal Component Analysis (PCA) to reduce dimensionality for each facial feature. Therefore un-plausible contours are eliminated by subspace method.

(3) Propose a multi-cue based prediction model. a) First, use low-level feature based face tracking algorithm - CAMSHIFT [5] to give an estimation for the position of face. Therefore the search space for observation model is narrowed down . b) multiple facial features are tracked simultaneously, spatial constraint among facial features is also taken into account. c) We learn the second-order auto regressive process (ARP) based dynamic model for facial features. d) We use graphical model - Bayesian network to enhance the ARP based dynamic model. The Bayesian network in this paper combines the influence on a facial feature in the current time instant contributed by multiple facial features in the previous time instant. In this way, it is more robust than tracking each facial feature independently. We integrate all the above prediction models as multi-cues into prediction model of the Kalman filter.

(4) Finally, the prediction and observation model make up the Kalman filter framework in the standard way.

1.1 Related Work

There is much prior work on tracking the facial feature. The ASM/AAM algorithm is the most famous method to track multiple facial features, as discussed in the previous section. Kapoor and Picard [6] proposed an infrared camera based tracking algorithm, which didn't need any manual alignment or calibration. However only upper facial features are tracked. Gu *et al.* [7] proposed an active approach to track facial expressions, and they combined the IR sensor with a Kalman filter, however the local graphes they used to capture the spatial relationship do not fit into a probabilistic framework.

Although there are many facial feature tracking algorithms not mentioned here, in this paper, we are specially interested in Bayesian network, a kind of graphical model's application in multiple facial feature tracking.

This paper is organized as follows. Section 2 describes the representation of facial feature's contour. Dimensionality reduction for facial feature is given in Section 3. Section 4 provides the multi-cue based prediction model in detail. Section 5 presents the measurement model. The experimental results are in Section 6.

2 Representation of facial feature's contour

In this paper, we track the contours of facial features, and use B-spline $(x(u,t), y(u,t))$ to represent facial feature's contour in time instant t. Suppose there are N spans and N_c control points, we have

$$x(u,t) = B(u)C^x(t), y(u,t) = B(u)C^y(t), 0 \leq u \leq N, \tag{1}$$

where $C^x(t) = [C_1^x(t), ..., C_{N_c}^x(t)]^T$ are the x coordinates for all control points in time instant t, and $C^y(t)$ are the y coordinates. For closed B-spline, the number of control points is equal to the number of spans N, i.e. $N_c = N$. For open B-spline, we have $N_c = N + d$, and the value of d may have appropriate variations when multiple knots are used [8]. The vector $B(u)$ consists of blending coefficients, which is defined by

$$B(u) = [B_1(u), B_2(u), \ldots, B_{N_c}(u)], \tag{2}$$

where $B_i(u)$ is the basis function of B-spline. The control points of B-spline composite into a spline vector $C(t)$:

$$C(t) = [C^x(t) \;\; C^y(t)]^T. \tag{3}$$

The B-splines that represent facial features are shown in Fig. 1, where Control points are shown as yellow dots, and the B-splines are described by magenta curves. The contour of eyebrow is the upper edge of eyebrow; the contour of eye is the boundary between eyelid and eyeball, excluding the eyelid; the contour of nose is the border between nose and the skin of face; the contour of mouth is the edges of upper and lower lips.

Figure 1: The contours of facial features are represented by B-spline.

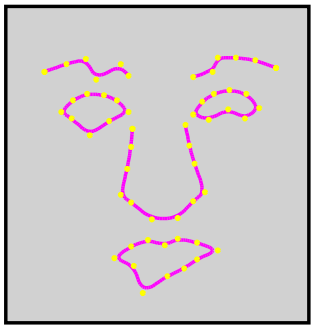


Figure 2: The results of arbitrarily manipulating spline vector.

If we manipulate the control point's positions arbitrarily, it is easy to generate spline that does not look like facial feature's contour (see Fig. 2). Therefore arbitrarily manipulating the control points may lead to tracking failure.

3 Dimensionality reduction for contour of facial feature

In this paper, the movement of facial feature can be decomposed into two parts: rigid motion and no-rigid motion. The rigid motion is caused by the motion of head, while the non-rigid one is the motion of each facial feature, e.g. the motion of eye, eyebrow, and mouth caused by facial expressions. Facial features (eyes, eyebrows, nose, mouth) are generally in the same plane. When rigid motion of head occurs, the contour of facial feature is projected into a two-dimension plane that has six degrees of freedom (DOF), which is translation in X and Y direction, rotation, and zoom in X, Y and diagonal direction. The six DOF actually belongs to an affine transform. For the non-rigid motion of a facial feature, we carry out Principal Component Analysis (PCA) for the contour of facial feature in the training face image sequence. Let the dimensionality of non-rigid motion is reduced to N_{nr}, and the total dimensionality of all facial features is $6 + N_{nr}$. Let s_t denote the parameters of state space after the dimensionality reduction, and the spline vector $C(t)$ can be written as

$$C(t) = W s_t + C_0, \tag{4}$$

where W is a $N_C \times N_S$ shape matrix, N_C denotes the DOF before dimension reduction, and $N_C = 2N_c$. N_S denotes the dimensionality after dimensionality reduction, and $N_S = 6 + N_{nr}$. C_0 is the template of contour, which is usually obtained by manually marking. As described in the following section, this kind of contour model for facial features can be conveniently integrated into the Kalman filter framework.

It is not good to allow arbitrary variation for each control points. Snakes based method is not robust, since it permits more DOF than need.

4 Multi-cue based prediction model

In video based tracking, with the help of dynamic model to predict the behavior of facial feature's motion, tracking will be more robust. In this paper, the parameters of dynamic model are not assigned empirically, but obtained from training. Our prediction model combines the quickness of auto regressive process with accuracy of graphical model.

4.1 Second order auto regressive process based prediction model

The motion of a facial feature's contour can be modelled by a noise driven second-order auto regressive process (ARP), which can be shown as the following second-order linear differential equation:

$$s_t = A_2 s_{t-2} + A_1 s_{t-1} + D_0 + B_0 w_t, \tag{5}$$

where A_1, A_2 and B_0 are matrices, A_1 and A_2 are the deterministic parameters, and B_0 is the stochastic parameter. D_0 denotes a fixed offset, and the distribution of each component of w_t belongs to i.i.d. Gaussian noise. Let $\chi_t = \begin{bmatrix} s_{t-1} \\ s_t \end{bmatrix}$, the Eq. (5) can be written as:

$$\chi_t = A\chi_{t-1} + D + Bw_t, \tag{6}$$

where $A = \begin{bmatrix} 0 & I \\ A_2 & A_1 \end{bmatrix}$, I is an identity matrix, $D = \begin{bmatrix} 0 \\ D_0 \end{bmatrix}$, and $B = \begin{bmatrix} 0 \\ B_0 \end{bmatrix}$. From Eq. (6), we can see that χ_t only depends on χ_{t-1} in the previous time instant. Therefore the dynamic model for one facial feature is actually a Markov chain. But this model doesn't consider the relationship among facial features.

Since second-order ARP can describe constant velocity motion, decay and damped oscillation [9], we use it as the plausible dynamic model.

4.1.1 Training dynamic model

In the second-order ARP dynamic model (see Eq. (6)), the parameters A, D and B are unknown. Although it is possible to specify the parameters empirically, it is more convincible to estimate these parameters from training image sequences. In this paper, we choose a bootstrapping strategy to learn the parameters for dynamic model.

First, we preset the parameters empirically to construct an initial dynamic model, and track slow and simple facial feature motion. By this way, we obtain a trained parameter sequence $\chi_1^1, \ldots, \chi_M^1$, where M denotes the number of frames in training image sequence. From Eq. (6), we can see that it is a Expectation-Maximization (EM) problem [10] to solve A, D and B from χ_i^1. Let $\overline{A^1}$, $\overline{D^1}$ and $\overline{B^1}$ denote the parameters trained from the EM in the first time. Use the parameters recently obtained, we can construct a new dynamic model. Then use it to track the previous image sequence more accurately, or track more complex motions of facial features. Generally, We can obtain an effective dynamic model by 2 to 3 times training.

Based on the Markov property of dynamic model, From the tracking results s_{t-2} and s_{t-1} in the previous instants, we can predict the state s_t of facial feature contour in the current time instant by the Eq. (4) and Eq. (6). Therefore the dynamic model obtained by training can be viewed as a prediction model in the Kalman filter framework. From Eq. (4), we know that s_t is corresponding to $C(t)$; i.e., from the state parameter s_t, we can conveniently obtain the facial feature's contour $C(t)$.

4.2 Using graphical model to enhance prediction

The dynamic model in Section 4.1 is for one facial feature, and we can build serval dynamic models, each for one different facial feature. But the multiple independent dynamic models ignore the natural relationship among facial features. Actually, the motions of each facial feature relate to each other. For example, when one frowns, his eyes will become smaller; when one surprises with wide open mouth, the eyebrows will move up. It is difficult to describe this kind of interrelationship deterministically. In this paper, we use probabilistic graphical model - Bayesian network to describe it non-parametrically.

4.2.1 Bayesian network

Bayesian network [11][12][13] is a directed acyclic graph (DAG). The Bayesian network used in the paper is shown in Fig. 3, where the filled circle denotes observation node, and the empty circle denotes hidden node.

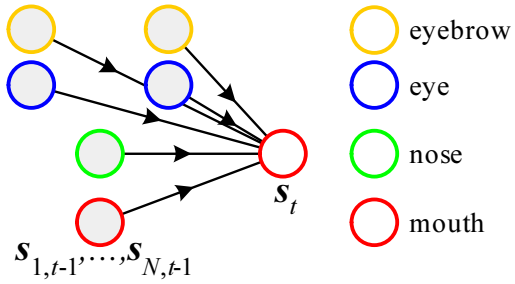


Figure 3: Bayesian network based dynamic model for multiple facial feature prediction.

The directed edge represents the statistical dependency between two nodes, and the direction is from the parent node to the child node. The intuitive meaning of Fig. 3 is that we can predict the current position of mouth's contour on condition that we have already known the positions of each facial feature's contours in the previous time instant.

4.2.2 Bayesian network based dynamic model

We utilize Bayesian inference to calculate the marginal probability $p(s_{j,t}|\{s_{i,t-1}\}_{i=1}^{N})$. For multiple facial feature tracking, the intuitive meaning is to predict the contour state parameter $s_{j,t}$ in current time instant t on condition that each facial feature's contour state parameters $\{s_{i,t-1}\}_{i=1}^{N}$ are already known. The result of prediction is $\hat{s}_{j,t}$ that maximizes the marginal probability.

$$\hat{s}_{j,t} = \arg\max_{s_{j,t}} p(s_{j,t}|\{s_{i,t-1}\}_{i=1}^{N}) \tag{7}$$

Generally, the Bayesian model based dynamic model can not be decomposed except that $s_{1,t-1}, \ldots, s_{i,t-1}, \ldots, s_{N,t-1}$ are mutually independent on condition of $s_{j,t}$. But for the convenience of computing, we use Eq. (8) to approximate the joint marginal probability. Murphy and Weiss proved that it is feasible [12].

$$p(s_{j,t}|\{s_{i,t-1}\}_{i=1}^{N}) = \prod_{i=1}^{N} p(s_{j,t}|s_{i,t-1}) \tag{8}$$

4.2.3 Training Bayesian network based dynamic model

Different from the parametric second-order ARP based dynamic model, Beyesian network based dynamic model $p(s_{j,t}|\{s_{i,t-1}\}_{i=1}^{N})$ is non-parametrical. From Eq. (8), we know that in order to solve the non-parametric dynamic model, the key point is to calculate $p(s_{j,t}|s_{i,t-1})$. From the conditional probability theorem, we have

$$p(s_{j,t}|s_{i,t-1}) = p(s_{j,t}, s_{i,t-1})/p(s_{i,t-1}), \tag{9}$$

where $p(s_{j,t}, s_{i,t-1})$ is joint probability, and $p(s_{i,t-1})$ is the probability of facial feature i in the previous time instant. From the training data, we fit the mixtures of Gaussians to $p(s_{j,t}, s_{i,t-1})$ and $p(s_{i,t-1})$. We can obtain $p(s_{j,t}|s_{i,t-1})$ by evaluating Eq. (9). The sketch maps for $p(s_{j,t}, s_{i,t-1})$ and $p(s_{i,t-1})$ are shown in Fig. 4 and Fig. 5.

4.2.4 Using Bayesian network based dynamic model to predict contour of feature

On condition that the facial feature i's state parameter $s_{i,t-1}$ is equal to $\xi_{i,t-1}$ in the previous time instant, we can predict the state of facial feature j based on Eq. (7) and Eq. (8).

$$\hat{s}_{j,t} = \arg\max_{s_{j,t}} p(s_{j,t}|\{\xi_{i,t-1}\}_{i}^{N})$$

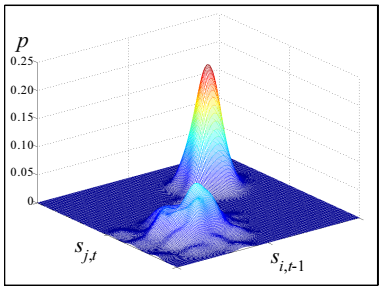


Figure 4: The sketch map of $p(s_{j,t}, s_{i,t-1})$.

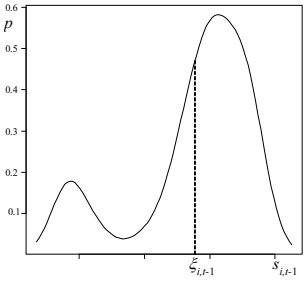


Figure 5: The sketch map of $p(s_{i,t-1})$.

$$
\begin{aligned}
&= \arg\max_{s_{j,t}} \prod_{i=1}^{N} p(s_{j,t}|\xi_{i,t-1}) \\
&= \arg\max_{s_{j,t}} \prod_{i=1}^{N} (p(s_{j,t}, \xi_{i,t-1})/p(\xi_{i,t-1})) \qquad (10)
\end{aligned}
$$

When $s_{i,t-1} = \xi_{i,t-1}$, $p(s_{j,t}, s_{i,t-1})|_{s_{i,t-1}=\xi_{i,t-1}}$ is single variable MOG, its one dimensional sketch map is shown in Fig. 6. From Eq. (10), we know that we need to calculate the maximum of the product of N MOGs. Since it is difficult to obtain the maximum directly, practically approximative methods are used, e.g. starting from an arbitrary point, use gradient descent algorithm to obtain the local maximum; utilizing discretization, draw ns samples, then find the maximum probability of them. We tend to use the latter method, since the global maximum can be obtained.

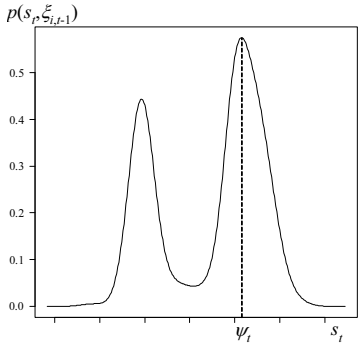


Figure 6: The sketch map of $p(s_{j,t}, s_{i,t-1})|_{s_{i,t-1}=\xi_{i,t-1}}$.

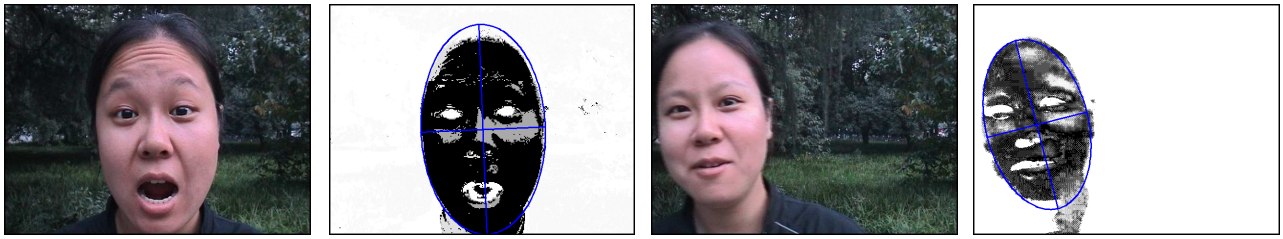
4.3 Low-level feature based face tracking in advance

The second order ARP based dynamic model is used for each facial feature, not for the whole face. To avoid the tracked facial feature's contour drifting out of the face, it is necessary to track the whole face firstly. Therefore we could narrow down the search range for the facial feature tracking.

Rigid motion can be tracked well by low-level methods, which are fast and robust, but cannot obtain the detail information for motion of facial features. For the non-rigid motion, the dynamic model based prediction model is suitable, since high-level method can tackle complex variation of features in the high-dimensional space.

In this paper, for the rigid motion, we use color histogram based CAMSHIFT algorithm [5] to track the face, and obtain the location of human face (see Fig. 7(a) and 7(b)). By this means, we set a search range for the observation model of facial feature tracking. Since observation is the most time-consuming part of the facial

feature tracking, narrowing down the search range make the tracking more efficient. CAMSHIFT tracking algorithm can also be used to obtain the orientation of face (see Fig. 7(c) and 7(d)), and this make preparation for spatial constraint in the following section.

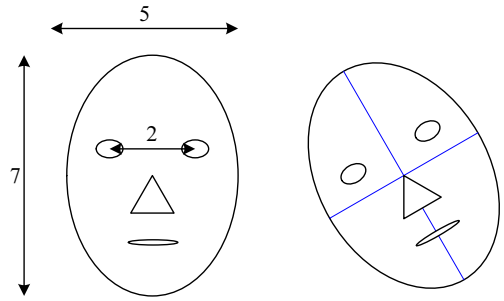


(a) (b) (c) (d)

Figure 7: CAMSHIFT algorithm can be used to obtain the face's position and orientation. (a) one frame contains the frontal face; (b) the probabilistic distribution map of face, the tracked face area is shown in the blue ellipse; (c) one frame contains the face with orientation; (d) the probabilistic distribution map of face, the blue across approximately gives the face's orientation.

In the experiments, the training for dynamic model was carried out for frontal faces. When the tracked face has orientation, we use CAMSHIFT algorithm to approximately get its orientation, and then warp the face image into the canonical position. We do the facial feature tracking on the canonical face, we undo the warping after tracking for graphical display.

4.4 Spatial constraint among facial features



(a) (b)

Figure 8: (a) The spatial constraint among facial features; (b) The spatial constraint also holds for face with orientation.

Human face image belongs to a special class. The facial feature position of different person only varies in a local area [14][15]. If we know the position and orientation of a face, we can use the spatial constraint among facial features to obtain the approximate position of each facial feature. As shown in Fig. 8(a), the human face can be described by an ellipse. The ratio between length and width is $7:5$, and the distance between two eye's centers is about $2/5$ of the width of face. For face with orientation, this spatial constraint still holds (see Fig. 8(b)).

The face position obtained by CAMSHIFT combined with spatial constraint is just an estimation for each facial feature's position. We include rigid motion in second order ARP model to further distill the residue rigid motion for each facial feature.

4.5 Multi-cue fusion for prediction

The spatial constraint among facial features can be combined with face tracking to specify the approximate position of facial features in the current time instant. This kind of low-level prediction can be integrated with dynamic model based prediction to improve the accuracy of prediction. The low-level CAMSHIFT algorithm based face tracking and spatial constraint among facial features are the preprocessing for prediction, and they can be easily fused into the prediction model.

4.5.1 Integrate second-order ARP based dynamic model with graphical model based one

Second-order ARP based dynamic model is very quick to prediction, but it ignores the influence on the facial feature's position in the current time instant caused by the position of other facial features in the previous time instant. The graphical model based prediction can obtain better result than ARP based method theoretically, but its non-parametric property determines that finding the global maximum is time consuming. This paper combines the advantages of these two method. The procedure of the algorithm is as follows:

Step1. Based on Eq. (5), we use reject sampling [16] method to draw ns samples (e.g. 10 or 20) from w_t. By this way, ns ARP based prediction results $s_{j,t}^k$ are generated, where $0 < k < ns$.

Step2. Substitute $s_{j,t}^k$ for $s_{j,t}$ in Eq. (10), we can find the best prediction $\hat{s}_{j,t}$ from the ns predictions.

Step3. Based on Eq. (4), we can solve the contour $\hat{C}(t)$ of current facial feature in time instant t.

In the ASM/ASM based multiple facial tracking algorithms, their dynamic models are only zero-order or first-order linear model, which can only describe uniform motion or uniform acceleration/deceleration motion. Therefore, the prediction based on these dynamic model is not enough. These tracking algorithms usually converge to correct position only when the initial position of facial feature's contour is reasonably appropriate. If the initial position is not very good, the tracker tends to be locked on a local maximum or fails.

5 Measurement model

After we have the prediction result of the facial feature's contour position, the result should be verified and adjusted by a measurement model. Compared with the prediction model, the measurement model is relatively easy to construct.

On condition that the predicted contour of a facial feature is $\hat{C}(t)$, one measurement in time instant t is to find feature (e.g. edge) along the normal vector $n(pt)$ of one point pt on the contour curve:

$$f(pt,t) = (C(t) - \hat{C}(t)) \cdot n(pt) + g(pt,t), \tag{11}$$

where $g(pt,t)$ is an i.i.d. Gaussian noise, and its variance δ^2 is a constant. The visual effect of measurement along the normal vector is to pull the contour curve $\hat{C}(t)$ along the normal direction based on feature found. Let $\tilde{C}(t)$ denote the contour curve after measurement, and it is the final tracking result in time instant t for current facial feature.

Already have the prediction and measurement model separately, they can be integrated into the Kalman filter framework in a standard manner.

6 Experimental results

We have implemented a prototype system MF^{2}T by Visual C++ and Matlab on Windows platform. MF^{2}T tracks 6 contours of facial features, which are left eyebrow (right eyebrow), left eye (right eye), nose and mouth. The contours are all quadric B-spline, where eyebrow and nose are open B-spline. Open curve's end points are triple knots. Other facial features are described by closed B-spline. The number of control points for eyebrow,

eye, nose and mouth is 10, 9, 16, 12 respectively (see Fig. 1). The total number of control points is 66, i.e., the total dimensionality is 132.

We choose Cohn-Kanade facial expression database as the training set [17], since it contains a lot of fontal expressive face images, and is stored as 30fps image sequence. The dimensionality reduction and training for prediction model are carried out on this database.

In order to reduce the dimensionality for contour model, we select 100 frame frontal face images from the training set, and these images belong to 48 different persons. We do PCA for each facial feature. After dimensionality reduction, the dimensionality for eyebrow, eye, nose and mouth is 7, 7, 12 and 9 respectively. The total dimensionality is 49, accounting for 99% variations.

In the training for second-order ARP based dynamic model, we obtain 6 dynamic models from image sequences, each for eyebrows (left and right), eyes (left and right), nose and mouth. In the training, we use interactive editing to manually mark feature points in order to get the ground truth. In the training of Bayesian network based non-parametric dynamic model, we use the same image sequence as the second-order ARP model. For image sequence with M frames, there are $(M-1)C_6^2 = 15(M-1)$ pairs of training data. In other words, there are 15 kinds of data for joint probability $p(s_{j,t}, s_{i,t-1})$, and we fit 8 cluster mixtures of Gaussians to them. For the state $p(s_{i,t-1})$ in the previous time instant, there are $C_6^1 = 6$ kinds of data, we also fit 8 cluster mixtures of Gaussians to them. We can calculate conditional probability $p(s_{j,t}|s_{i,t-1})$ from the fitted probabilities.

The reason to use mixture of Gaussian is that the relationship between contour of a facial feature in the previous instant and that in the current time instant is multimodal, and is not Gaussian. We make experiments to prove this. We choose 3 most important PCA coefficients for left eye in the previous time instant, and 3 for mouth in the current time instant. Therefore we obtain 9 kinds of PCA coefficient pairs, and each kind of PCA coefficients' number is $M-1$. The spatial distribution for 4 different kinds of PCA coefficient pairs is shown in Fig. 9, and we can see that simple Gaussian approximations would obscure this data set's meaningful information.

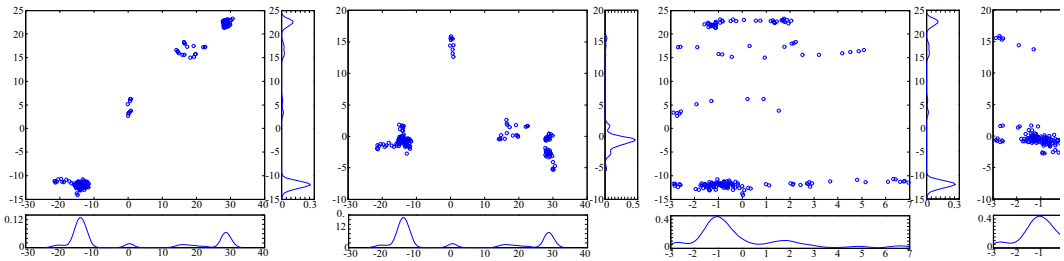
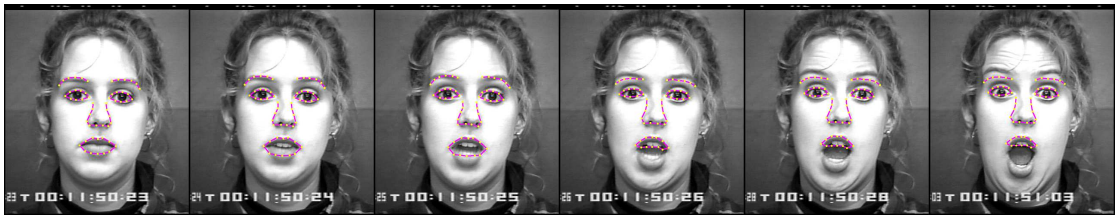


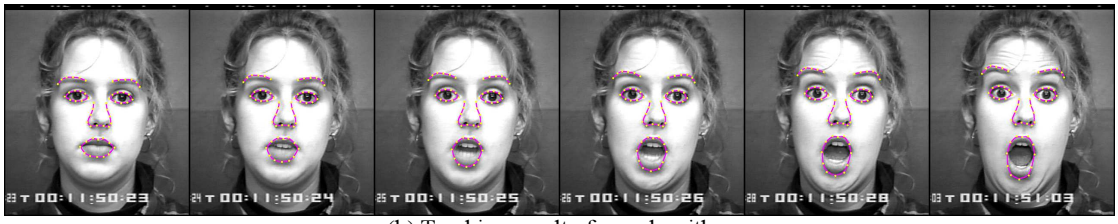
Figure 9: Joint probabilistic density of 4 pairs of PCA coefficients. The corresponding marginal distribution is shown on each figure's right and lower part. We can see that the relationship between PCA coefficients is multimodal and non-Gaussian.

In the experiments, it turns up that facial expressions change very fast, e.g. it only needs 10 frames to change expressions from neutral to happy (for 30fps video); therefore there are relatively large motion in adjacent two frames. We carry out two kinds of experiments: (1) Tracking multiple facial features in fontal expressive face images in Cohn-Kanade database (640×490, 30fps) (see Fig. 10(2) We use digital video camera to capture face image sequence (640×480, 30fps) with expression, orientation and occlusion in the outside. We track these image sequences (see Fig. 11 and Fig. 12). All the tracked image sequences are not included in the training set, and we also compare our algorithm's result with that of AAM's.

The tracking result for a surprise expression sequence is shown in Fig. 10. We can see from the result of edge detection (see Fig. 10(c)) that, when mouth is wide open, the teeth and tongue form dense cluster for the contour of mouth. AAM is locked on local maximum, since it treats the contour of teeth as that of lower lip, and regards the dark circles as contour of eyebrows. Our algorithm correctly predicts that the mouth will



(a) Tracking result of AAM



(b) Tracking result of our algorithm

(c) Edge detection result for image sequence

Figure 10: Tracking results for a surprise expression sequence. (a) Tracking result of AAM. (b) Tracking result of our algorithm. (c) Edge detection result for image sequence.

probably open when the eyebrows are rising and the eyes are opening. The original size of image sequence in Fig. 10 is 640×490 pixels, and the tracking is carried out in that size. However for the purpose of display in this paper, we crop the image down to the size of 432×490. The frame number is shown in the time code at the bottom of the image.

Our algorithm also can robustly track multiple facial features when face has orientation and size variation (see Fig. 11(b)). Since we use CAMSHIFT algorithm to get the position of face in advance, we avoid the AAM tracker's problem that left eyebrow is out of the face (see Fig. 11(a)). Furthermore, in the graphical model based prediction, we consider the spatial constraint of facial features, the problem that contour of upper lip overlaps with that of nose is also avoided. In Fig. 11, the frame numbers are 9, 22, 37, 42, 62, 66, 70 and 157.

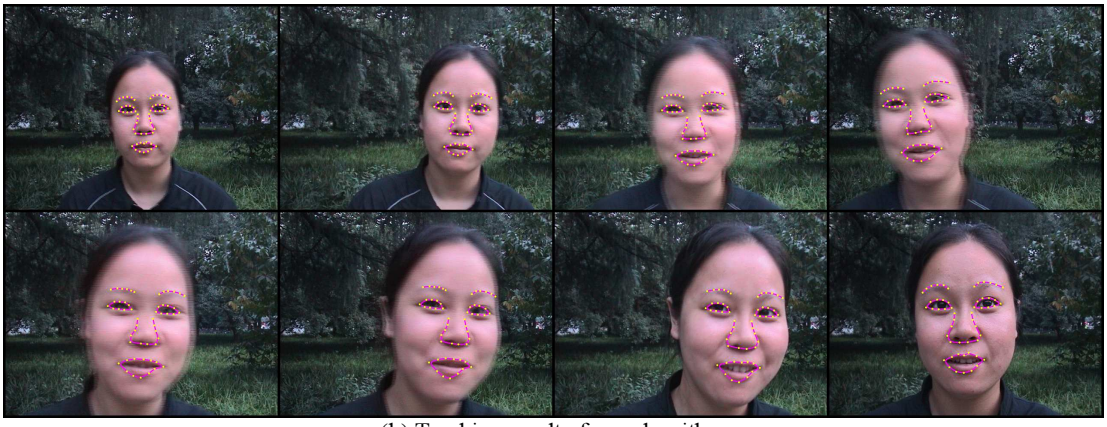
For the classical AAM algorithm, since occlusions occur in various forms (e.g. occlusions on different facial feature), it is difficult to integrate such negative samples into the training set; therefore the training set for AAM only contains faces without occlusions. It is difficult for AAM to deal with occlusions on facial features in image sequence, and AMM tracker usually fails in such situation (see Fig. 12(a)). However our algorithm utilizes the natural relationship among facial features, the contour of the occluded facial feature can be inferred by Bayesian network learning (see Fig. 12(b)). The comparison results are shown in Fig. 12, where the frame numbers are 0, 1, 4, 23, 51, 54, and 55.

Our algorithm runs at 3 frames per second on a Pentium4 1.8G computer.

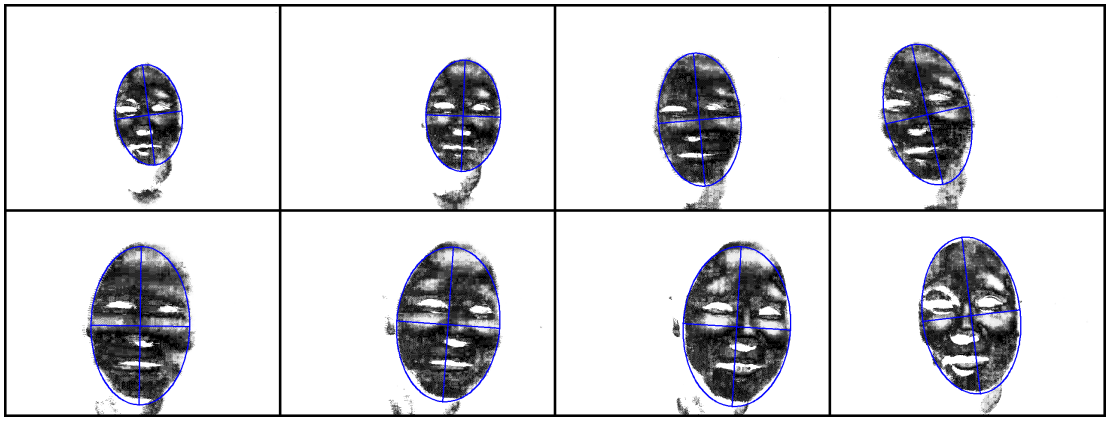
7 Conclusions

In this paper, we propose a Bayesian network enhanced prediction model based multiple facial feature tracking algorithm. We combine second-order ARP based dynamic model with graphical model - Bayesian network based one, and obtain quick and accurate multi-cue based prediction model. The prediction and measurement

(a) Tracking result of AAM



(b) Tracking result of our algorithm



(c) The face tracking result using Meanshift

Figure 11: Comparison of tracking results: from far to near, quickly approaching the camera, and with head orientation and face expression. (a) Tracking result of AAM. (b) Tracking result of our algorithm. (c) The face tracking result using CAMSHIFT.

model are integrated into the Kalman filter framework in a standard way. The experimental results show that our algorithm is effective.

(a) Tracking result of AAM

(b) Tracking result of our algorithm

Figure 12: Comparison of tracking results with occlusions: hiding mouth by hand. (a) Tracking result of AAM. (b) Tracking result of our algorithm.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *IJCV*, 1(4):321–331, 1987.

[2] T. F. Cootes, C. J. Taylor, A. Lanitis, and et al. Building and using flexible models incorporating grey-level information. In *Proc. ICCV*, pages 242–246, 1993.

[3] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Learning to identify and track faces in image sequences. In *Proc. ICCV*, pages 317–322, 1998.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[5] G. R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proc. WACV*, pages 214–219, 1998.

[6] A. Kapoor and R. W. Picard. Real-time fully automatic upper facial feature tracking. In *Proc. FG*, pages 8–13, 2002.

[7] H. Gu, Q. Ji, and Z. Zhu. Active facial tracking for fatigue detection. In *Proc. WACV*, pages 137–142, 2002.

[8] C. de Boor. *A practical guide to splines*. Springer-Verlag, 1978.

[9] A. Blake and M. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proc. SIGGRAPH*, pages 185–192, 1994.

[10] M. Jordan. *Learning in graphical models*. Kluwer Academic Publishers, 1998.

[11] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[12] K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proc. UAI*, pages 378–385, 2001.

[13] J. S. Yedidia and W. T. Freeman. Understanding belief propagation and its generalizations. In *Proc. IJCAI*, 2001.

[14] A. Shashua and T. Riklin-Raviv. The quotient image: class-based re-rendering and recognition with varying illuminations. *IEEE Trans. PAMI*, 23(2):129–139, 2001.

[15] G. Loy, L. Fletcher, N. Apostoloff, and A. Zelinsky. An adaptive fusion architecture for target tracking. In *Proc. FG*, pages 248–253, 2002.

[16] W. Press, S. A. Teukolsky, W. T. Vetterling, and et al. *Numerical recipes in C, 2nd edition*. Cambridge University Press, 1992.

[17] T. Kanade, J. Cohn, and Y. L. Tian. Comprehensive database for facial expression analysis. In *Proc. FG*, pages 46–53, 2000.