Multi-Modal Face Tracking Using Bayesian Network

Fang Liu[1], Xueyin Lin[1], Stan Z Li[2], Yuanchun Shi[1]
[1]*Dept. of Computer Science, Tsinghua University, Beijing, China, 100084*
[2]*Microsoft Research Asia, Beijing, China, 100080*
liufang@tsinghua.org.cn, lxy-dcs@mail.tsinghua.edu.cn, szli@microsoft.com

Abstract

This paper presents a Bayesian network based multi-modal fusion method for robust and real-time face tracking. The Bayesian network integrates a prior of second order system dynamics, and the likelihood cues from color, edge and face appearance. While different modalities have different confidence scales, we encode the environmental factors related to the confidences of modalities into the Bayesian network, and develop a Fisher discriminant analysis method for learning optimal fusion.

The face tracker may track multiple faces under different poses. It is made up of two stages. First hypotheses are efficiently generated using a coarse-to-fine strategy; then multiple modalities are integrated in the Bayesian network to evaluate the posterior of each hypothesis. The hypothesis that maximizes a posterior (MAP) is selected as the estimate of the object state. Experimental results demonstrate the robustness and real-time performance of our face tracking approach.

1. Introduction

Face tracking is important for many vision-based applications such human computer interactions. Different face trackers may be classified into two classes: general tracking methods and learning based methods.

General tracking methods use some low level features such as color and contour to track objects including faces [1,2,3,4,5,6,7,8,9,10]. For example, background models are often built and updated to segment the foreground regions [1,2,3]. Monte Carlo methods [4,5,6,7] adopt sampling techniques to model the posterior probability distribution of the object state and track objects through inference in the dynamical Bayesian network. A robust non-parametric technique, the mean shift algorithm, has also been proposed for visual tracking [8,9,10]. In [8] human faces are tracked by projecting the face color distribution model onto the color frame and moving the search window to the mode (peak) of the probability distributions by climbing density gradients. In [9,10] tracking of non-rigid objects is done through finding the most probable target position by minimizing the metric based on Bhattacharyya coefficient between the target model and the target candidates. Some other methods are presented to track human heads, for example, tracking contour through inference of JPDAF-based HMM [11], an algorithm combining the intensity gradient and the color histogram [12] and motion-based tracking with adaptive appearance models [13].

Learning based methods track the faces using learning approaches [14,15,16,17,18]. Results from face detection should help face tracking. In face detection, the goal is to learn, from training face and non-face examples (sub-windows), a highly nonlinear classifier to differentiate the face from non-face pattern. The learning based approach has so far been the most effective for constructing face/non-face classifiers. Taking advantage of the fact that faces are highly correlated, it is assumed that some low dimensional features that may be derived from a set of prototype face images can describe human faces. The system of Viola and Jones [14] makes a successful application of AdaBoost to face detection. Li et al [15] extend Viola and Jones' work for multi-view faces with an improved boosting algorithm.

A face detection based algorithm can be less sensitive to illumination changes and color distracters than the general tracking methods due to the use of face pattern instead of color and contour only. However, they have their own difficulties. First, face detector may mass some faces and contain false alarms. Second, partially occluded faces and rotated faces are more like to be missed. Third, multiple pose face detection is several times more costly than frontal face detector.

This paper presents a Bayesian face tracker that aims to track the position, scale and pose of multiple faces. The face tracker unifies low level features such as color and contour, and high level features such as face appearance for robust and real-time tracking of multiple faces. As shown in figure 1, the Bayesian Network [19, 20] includes four components: (1) the prior model, a second order system dynamics, (2) color, (3) edge and (4) face appearance likelihood models. The presented method is different from the previous work for example, Monte Carlo methods [4,5,6,7], in the following ways:

First, in the stage of hypotheses generation, our tracker reduces significantly the number of hypotheses needed for robust tracking. In Monte Carlo methods, factored sampling and importance sampling techniques are employed to predict the distributions of the object states.

Color likelihood is modeled as a mixture of Gaussians, and more hypotheses are generated where there is stronger color likelihood. However, it is computationally expensive with thousands of hypotheses to model the probability distribution of the object state. We adopt a much more efficient method for hypotheses generation. A bottom-up strategy is used to generate hypotheses that represent not the probability distribution of the object state but the probable hypotheses. The color likelihood is explicitly encoded in the Bayesian network as an efficient measurement. Though only several hypotheses are used, the result (that is, the MAP hypothesis) is not biased in most cases because they are generated from observations.

Second, in the stage of hypotheses evaluation, a Bayesian network is used to evaluate the posterior probability of every hypothesis. The Bayesian network may integrate new constraints easily to improve the tracker's performance. Besides, it incorporates multiple cues in a uniform and explicit way, which makes it easy to evaluate the confidence level of each modality and allows a learning algorithm for multiple modalities fusion. We introduce environmental factors to model the confidence of each modality and encode them into the Bayesian network to fuse multiple modalities more effectively.

In the remainder of this paper, section 2 introduces the general overview of our algorithm. Section 3 describes the method to generate hypotheses. Section 4 explains hypotheses evaluation in the Bayesian Network framework. Section 5 describes learning algorithm for modalities fusion. Section 6 shows experimental results and finally section 7 contains conclusions and discussions.

2. General Overview

The tracker is initialized by the output of face detectors. Once a new face is detected, it will be added to the objects list being tracked.

Multiple hypotheses are used to improve the robustness of the tracker. The hypotheses are generated via a coarse-to-fine strategy. The strategy is made up of two steps: first, a coarse sampling process in which hypotheses are generated from the local maxima in the probability distribution image (PDI); second, a boosting face filter is applied to refine the hypotheses. The coarse hypotheses are refined by the result of the face filter.

Then a Bayesian Network (figure 1) is used to infer the hidden object state in each frame. Bayesian network is an efficient tool for data fusion and used to integrate the modalities such as the prior probability, color, edge and face appearance likelihoods. The posterior probability of every hypothesis is evaluated by the Bayesian network and the object state is approximated by the hypothesis that maximizes a posterior.

The Bayesian network is similar with the dynamical one in that it also integrates the prior model. However, this work adopts a multiple hypothesis Bayesian network structure where each hypothesis has a posterior probability after the evidences are integrated.

We further introduce the environmental factors into the Bayesian network to model the confidence of every modality during the tracking process. Then we present a learning algorithm based on Fisher discriminant analysis [21] to learn the way to fuse the modalities.

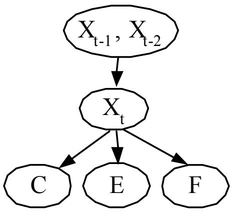


Figure 1. The Structure of the Bayesian network. X_{t-1} and X_{t-2} are the previous object states; X_t represents the current object state. C, E and F are the color, edge and face appearance measurements respectively.

3. Hypotheses Generation

Multiple hypotheses are widely used in robust tracking in the cluttered environment. We adopt a coarse-to-fine strategy to generate the hypotheses.

First, the hypotheses are generated from the local maxima in the probability distribution image (PDI).

The PDI is converted from original color frames via color or blob models of the object, color/depth background model, or motion detection. The hypotheses generation does not require the use of a certain model. In this paper, the face is modeled as a blob. The blob is represented as a Gaussian model. It has a spatial (x, y) and color (r, g, s) (r=R/s, g=G/s, s=R+G+B)) component.

$$\mathrm{P(O)}=\frac{\exp[-\frac{1}{2}(\mathrm{O}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathrm{O}-\mu)]}{(2\pi)^{m/2}\,|\,\Sigma\,|^{1/2}} \qquad (1)$$

where the dimension m is 5, O is a 5-dimension vector (x, y, r, g, s), μ is the mean value, Σ is the covariance matrix of the Gaussian model. Because of different semantics of the spatial and color attributes, their distributions are assumed to be independent, i.e. $\Sigma=\begin{pmatrix}\Sigma_s & 0\\ 0 & \Sigma_c\end{pmatrix}$, where Σ_s and Σ_c means the covariance matrix of the spatial and color distributions of the blob respectively.

The current position of the face is predicted by Kalman filter. The prediction is not included in the dynamics of the BN because the process of hypotheses generation is independent of that of hypotheses evaluation. Then a set of initial hypotheses is uniformly distributed in the neighborhood region of the predicted position. Then mean shift algorithm is used to move the initial hypotheses to the place where local maxima occur in the PDI. If the hypotheses are inter-overlapped, they are merged into a representative.

The mean shift algorithm [8,9,10] has some benefits for the tracking task: it finds the local maxima of a non-parametric probability distribution efficiently, which is important for real-time applications. It also tends to ignore outliers in the data and thus compensates for noise and distracters in the observations.

Second, the coarse hypotheses are refined by using a boosted face filter (BFF). The BFF is based on a boosted face detector [14,15]. The detector is to classify a subimage of standard size (e.g. 20×20 pixels) into either face or non-face. The face filter also computers most information needed by face detection; however, it does not make decisions – it provides a confidence value as the likelihood of every subimage being a face.

The likelihood is computed based on a set of local features. There are a huge number of candidate features. AdaBoost learning is used to select a good collection of features to best represent the face pattern as opposed to nonfaces. Then the likelihood is computed based on the selected features.

For the example x, AdaBoost assumes that a procedure is available for learning sequence of weak classifiers $h_m(x) \in \{0,1\}$ (m=1,2,…,M) from the training examples. A stronger classifier is a linear combination of the M weak classifiers:

$$H_M(x) = \frac{\sum_{m=1}^{M} \alpha_m h_m(x)}{\sum_{m=1}^{M} \alpha_m} \qquad (2)$$

where α_m are the combining coefficients. The classification of x is obtained as y(x)=sgn[H_M(x)-0.5]. The AdaBoost learning procedure is used to derive α_m and $h_m(x)$.

Assume that one of the coarse hypotheses is (x, y, s), (x, y) is the position component, s is the scale component of the hypothesis. Then the image is scanned in the neighborhood of the hypothesis at several scales at 0.83s, 1.0s, 1.2s and three views, frontal, left and right profile, producing a large number of subwindows of varying locations and scales. The squared subwindows are effectively normalized to a fixed size of 20×20. The BFF is applied to each normalized subwindow x and the face confidence is empirically defined based on the output of the BFF for each subwindow. This gives a confidence map in the (i, j, s, p) (position-scale-pose) space. For each coarse hypothesis (x,y,s), we generate the corresponding hypothesis (x*, y*, s*, p*), for which the multi-view BFF produces the most significant confidence in the neighborhood of the hypothesis (x,y,s).

4. Hypotheses Evaluation

Hypotheses are then evaluated in a Bayesian network. In Monte Carlo methods [4,5,6,7], the cues integration is done through different stages such as hypotheses generation and measurement. In this paper, the posterior probability of each hypothesis is evaluated through integrating multiple cues in an efficient and flexible way.

Hypotheses are evaluated by a posterior probability as the following:

$$P(X_t \mid C, E, F, X_{t-1}, X_{t-2}) \qquad (3)$$

where X_{t-1} and X_{t-2} are the previous object state, X_t represent the current object state (i.e. one of the hypotheses). C, E and F are the color, edge and face appearance measurements respectively.

By using the Bayesian formula and the Bayesian Network as shown in figure 1, we may interpret the posterior probability of each hypothesis as follows:

$$\begin{aligned} &P(X_t \mid C, E, F, X_{t-1}, X_{t-2}) \\ &\propto P(X_t, C, E, F, X_{t-1}, X_{t-2}) \\ &\propto P(C \mid X_t) P(E \mid X_t) P(F \mid X_t)\ P(X_t \mid X_{t-1}, X_{t-2}) \quad (4) \end{aligned}$$

In the following, more details are given about the prior and likelihood models.

4.1. Prior Model

The prior model $P(X_t \mid X_{t-1}, X_{t-2})$ is derived from the dynamics of object motion, which is modeled by a second order autoregressive process (ARP).

$$X_t = A_2 X_{t-2} + A_1 X_{t-1} + D + B w_t \qquad (5)$$

where the ARP parameters A_1, A_2 and B are matrices representing the deterministic and stochastic components of the dynamical models respectively. w_t is Gaussian noise drawn from N(0, I). D is a fixed offset. Therefore, the prior model is defined as:

$$\begin{aligned} &\log P(X_t \mid X_{t-1}, X_{t-2}) \\ &= -\frac{1}{2} \| B^{-1}(X_t - A_2 X_{t-2} - A_1 X_{t-1} - D) \|^2 \qquad (6) \end{aligned}$$

where ||…|| is the Euclidean norm. The ARP parameters A_1, A_2, B and D, are learned from training examples via maximum likelihood estimation [21].

4.2 Color Likelihood Model

The color likelihood of hypotheses $P(C \mid X)$ is derived directly from the PDI presented in section 3. The color likelihood of each hypothesis is defined as:

$$P(C \mid X) = \frac{1}{n_H} \sum_{(x,y)\in H} I_p(x,y) \qquad (7)$$

where H represents the hypothetical region. n_H is the scale of the hypothesis. $I_p(x,y)$ is the pixel value at location (x, y) in its PDI.

4.3 Edge Likelihood Model

Contour is used to compute the edge likelihood. A contour, described as an ellipse, is constructed based on each hypothetical region in the PDI. A contour is defined by 5 parameters: the center point, two axis and rotation angle. The center point is the centroid of the probability distribution in the region. The axes are defined as the two eigen-values of the probability distributions in the region. The rotation angle is defined as the 2D orientation of the probability distribution. All the ellipse parameters may be easy to obtain by using the first and second moments of the PDI in the region. The edge likelihood is approximated by examining a set of points that lie on the contour:

$$\log P(E \mid X) = \frac{1}{N_c} \sum_{i=1}^{N_c} f(|G(n(i))|) \qquad (8)$$

where $n(i)$ is the normal line to the contour centered at the i-th pixel on the contour. $G(l)$ is the gray-level gradient along the line l at its center point. f(x) is a non-decreasing function. N_c is the number of examined points on the contour.

4.4 Face Appearance Likelihood Model

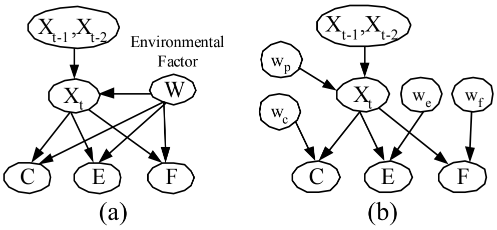
The face appearance likelihood $P(F \mid X_t)$ is defined as the confidence value corresponding to each hypothesis, which is empirically defined based on the BFF output.

Every hypothesis is evaluated by using the above models and the MAP one is chosen as the estimated of the object state.

5. Learning Bayesian Network

Bayesian network allows a probabilistic way for data fusion. However different modalities have different confidences, which make it necessary to learn the way to fuse multiple modalities. We introduce the environmental factors to model the confidences of different modalities, and encode them into the Bayesian network. The environmental factors provides a mechanism to fuse these modalities so that the hypothesis, which estimates the object state best, can be robustly chosen from all the hypotheses in every frame by using MAP criterion.

The Bayesian network that encodes the environmental factors explicitly is shown as figure 2(a).



(a) (b)

Figure 2. The Bayesian network that embeds environmental factors. Left image shows the embedding of the environmental factors W. Right image gives the factorized form of the environmental factors.

We assume that the environmental factors are made up of 4 independent components: w_p, w_c, w_e, and w_f, which reflects the prior probability, color, edge and face appearance likelihoods respectively. Therefore, the BN in figure 2(a) can be factorized into the form in figure 2(b). Now the posterior probability of every hypothesis can be computed as:

$$\begin{aligned}
&P(X_t \mid C,E,F,X_{t-1},X_{t-2},W) \\
&\propto P(X_t,C,E,F,X_{t-1},X_{t-2},W) \\
&\propto P(C \mid X_t,W)\ P(E \mid X_t,W) \\
&P(F \mid X_t,W)\ P(X_t \mid X_{t-1},X_{t-2},W) \\
&\propto P(C \mid X_t,w_c)\ P(E \mid X_t,w_e) \\
&P(F \mid X_t,w_f)\ P(X_t \mid X_{t-1},X_{t-2},w_p) \qquad (9)
\end{aligned}$$

It is further assumed that the three terms in (9) can be written as:

$$P(C \mid X_t,w_c) \propto P(C \mid X_t)^{w_c} \qquad (10)$$

$$P(E \mid X_t,w_e) \propto P(E \mid X_t)^{w_e} \qquad (11)$$

$$P(F \mid X_t,w_f) \propto P(F \mid X_t)^{w_f} \qquad (12)$$

$$P(X_t \mid X_{t-1},X_{t-2},w_p) \propto P(X_t \mid X_{t-1},X_{t-2})^{w_p} \qquad (13)$$

With the above assumptions (10)-(12), we obtain a new representation of the posterior probability. Since it is difficult to learn the environmental factors directly from the Bayesian network, they are learnt through an approximation approach based on Fisher discriminant

[21]. Note that log-posterior is changed into the form of a weighted sum after taking logarithm:

$$\log P(X_t \mid C, E, F, X_{t-1}, X_{t-2}, W)$$
$$\propto w_c \log P(C \mid X_t) + w_e \log P(E \mid X_t) + w_f \log P(F \mid X_t)$$
$$+ w_p \log P(X_t \mid X_{t-1}, X_{t-2}) \qquad (14)$$

The right four terms in (14) is proportional to the projection of the vector ($\log P(C \mid X_t)$, $\log P(E \mid X_t)$, $\log P(F \mid X_t)$, $\log P(X_t \mid X_{t-1}, X_{t-2})$) (We call this vector log-likelihood vector later) onto the factor vector (w_c, w_e, w_f, w_p). For the convenience of learning the factor vector, we approximate the MAP criterion as a two-category classifier. One category includes the log-likelihood vectors of the MAP hypotheses; the other includes the log-likelihood vectors of the non-MAP hypotheses.

In the training process, the object is properly tracked on training sequences with the environmental factors empirically defined. For every frame, if a hypothesis is a MAP one, its log-likelihood vector is labeled as class A; or else its log-likelihood vector is labeled as class B. The training samples are collected from a large number of frames and then are used to find the best factor vector.

The best factor vector is expected to be the projection direction for which the projected samples are best separated (in the sense described in the following). This is exactly the goal of classical discriminant analysis. A good separation of the projected data means that the difference between the means of two classes to be large relative the some measures of the standard deviations for each class. The best factor vector is given by Fisher's linear discriminant - the projection direction yielding the maximum ratio of between-class scatter to within-class scatter.

However, there is no need to find the threshold for the approximate classifier. This is because that the factor vector is only used for fusing multiple modalities and we still select the hypothesis based on MAP criterion, but not from the output of the approximate two-category classifier during the tracking process.

6. Experiments

Various experiments have been conducted on real world video sequences in order to examine the effectiveness and robustness of the Bayesian face tracker.

The face tracker is initialized by a background thread in which AdaBoost face detector runs on whole images due to its huge computational cost. When a new object is detected, it is added to the objects list to be tracked.

A face tracker is constructed based on only low-level cues including prior model, color and edge likelihood models. The face is modeled as a blob and hence the tracker does not rely on fixed skin color model. This tracker do not use face appearance in the tracking process. The hypotheses are generated from the only coarse step of the coarse-to-fine strategy and evaluated using the low level cues.

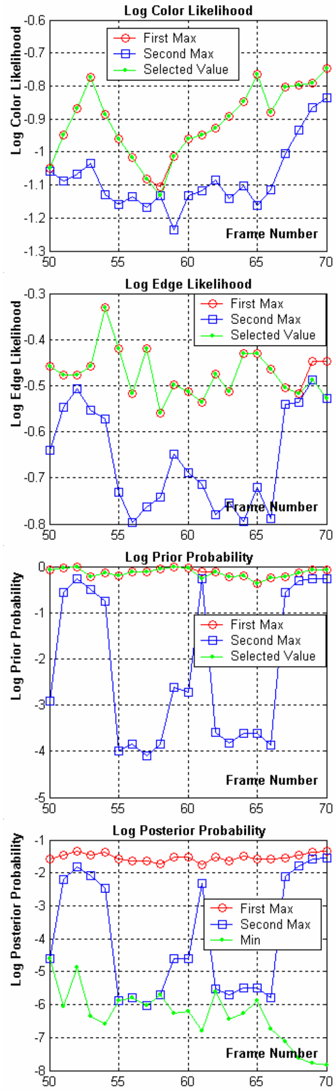


Figure 3. Cues Evaluation on a sequence of face tracking. This figure shows the log-likelihoods of color and edge, log prior and log posterior of the hypotheses in each frame. First max and second max mean the first and second maximum values among all hypotheses in each frame. The selected value is chosen according to MAP.

We evaluate all the three models on a sequence. Figure 3 shows the first-max, second-max and MAP values of every modality from frame 50 to 70. It is easy to find that tracking with multi-modal fusion is more robust than that with single modality from this figure. For example, in frame 58, the best estimate of the object state is the MAP hypothesis, whose color likelihood is not maximum. Color likelihood alone is not enough to select the proper hypothesis. The similar situation occurs to edge likelihood model in frame 69 and 70 and to the prior model in frame 61. Figure 4 shows the 4 frames and all the hypotheses of every frame.

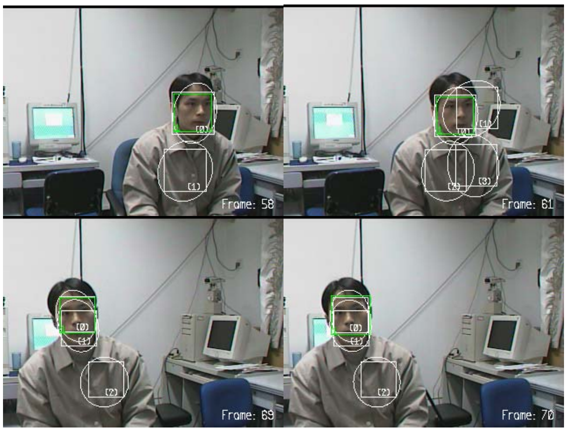


Figure 4. All the hypotheses in Frame 58,61,69,70

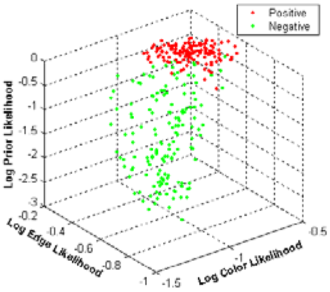


Figure 5. The samples of the log-likelihood vectors. Positive samples are the log-likelihood vectors from MAP hypotheses. Negative samples are the log-likelihood vectors from non-MAP hypotheses.

We further conducted experiments to learn the environmental factors. First the training samples are collected from the result of tracking objects in a sequence. Fig. 5 shows an example of the training samples. The best factor vector is given by the Fisher linear discriminant (0.0402, 0.0236, 0.0105). We applied the unlearned factor vector and the learned one to track objects on another sequence and compare the results in these two cases. Figure 6 shows the comparison between the tracking results of un-learned and learned cases. Figure 6 shows the first maximum, second maximum and minimum log-posteriors from frame 100 to 200. The learned factor vector leads to a better separation between the MAP hypothesis and the other ones in most frames. This is also confirmed by the values of the Fisher criterion function: 0.0028 in the unlearned case and 0.0101 in the learned case. Though the unlearned and learned trackers both track the objects successfully in the test sequence, it is expected that the learned tracker will get better result in very difficulty situation than the unlearned one.

BFF is then integrated into the above face tracker based on low level cues. The initialization stage is the same. After initialization, hypotheses are generated using the coarse-to-fine strategy and evaluated using not only the low level cues but also the high level cue: face appearance likelihood.

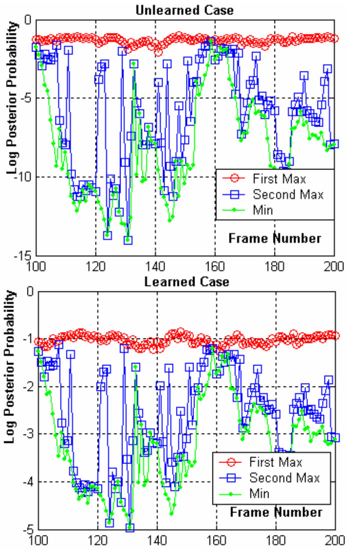


Figure 6 The log-posterior values of frame 100 to 200.

BFF improves the above face tracker in two ways: one is to reduce the bias of the estimate of the object state; the other is to increase the robustness of the Bayesian face tracker greatly. Figure 7 shows two examples where the improved tracker locates the object regions more exactly than the above tracker. This is because that when the profile face is observed, the estimation from color characteristics is biased, while the BFF may remedy it.

The introduction of BFF also allows us to estimate the poses of faces. In figure 7, the arrows in second row image give the poses of faces.

Tracking multiple objects is useful in real applications. The Bayesian tracker is naturally extended to this case. For each object, an object model (blob model in this paper) is constructed and used to obtain its PDI. To track the object, we just apply the Bayesian tracker in its PDI. Occlusion is a major problem in tracking multiple objects. Occlusion and re-appearance of an object in the scene are viewed as termination of the corresponding tracker and re-initialization of another tracker. Figure 8 shows the result of tracking multiple objects.

This tracker runs comfortably at about 20 fps for 320×240 frames on P1.4GHz PC. This is because of the efficiency of the coarse-to-fine strategy and only several hypotheses being used. At the same time, the Bayesian inference structure makes the tracker very robust to rapid motion, occlusions and clutter.

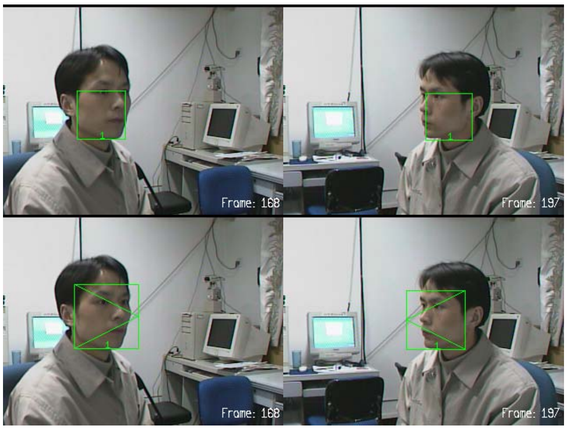


Figure 7. The comparison between the tracker based on low-level cues and the improved tracker.

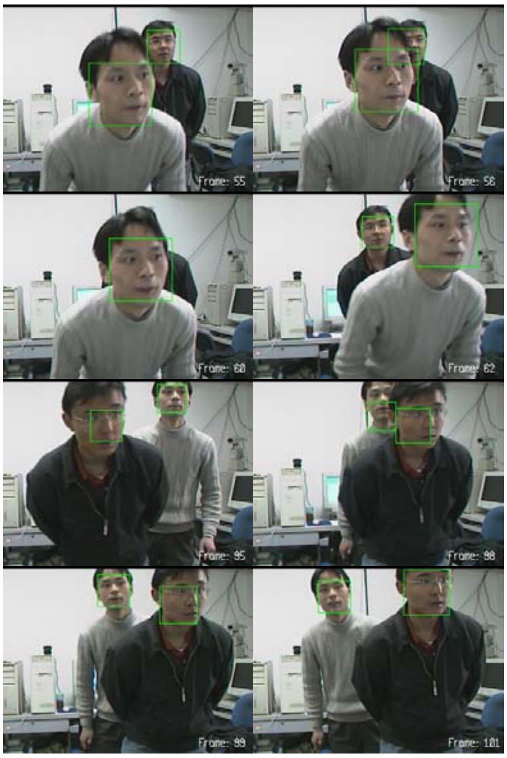


Figure 8. Tracking of multiple faces: occlusions handling. (Frame 55,56,60,62 and 95,98,99,101)

7. Conclusions

In this paper, we have presented a new face tracker based on Bayesian network. It is different from Monte Carlo methods in the scheme of hypotheses generation and the inference structure of the graphical models. The tracker uses several hypotheses and a Bayesian Network to estimate the object state from prior and likelihood models. There are three novel ideas in the face detector and Bayesian network based face tracker, summarized as follows.

First, a coarse-to-fine strategy is used to generate multiple hypotheses efficiently from the PDI and face confidence map. This method improves the efficiency of hypotheses greatly and reduces the number of hypotheses needed for robust tracking.

Second, a Bayesian Network is used to evaluate the hypotheses, integrating prior probability, color, edge and face appearance likelihoods in a uniform and efficient way.

Third, we further extend the Bayesian network of the tracker to learn the environmental factors for efficient data fusion strategy. We interpret the learning problem as the problem of classical discriminant analysis after making some approximations. The introduction and learning of environmental factors can adapt the tracker to different situations.

Experiments have been conducted to demonstrate the robustness and real-time performance of the tracker. Experiments have also been done to evaluate multiple modalities and learn the environmental factors based on Fisher discriminant analysis, which is useful to future tracker implementation.

8. Acknowledgement

This work is supported by NSFC (National Science Foundation of China) No. 60103004 and No. 69975009 and 863 (National High-tech Developing) Plan: 2001AA114171.

9. References

[1] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” IEEE Trans. Pattern Analysis and Machine Intelligence vol. 19, no. 7, July 1997.

[2] I. Haritaoglu, D. Harwood, and L.S. Davis. “W4: Real-time surveillance of people and their activities”, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8):809–830, August 2000.

[3] T. Zhao, R. Nevatia and F. Lv, “Segmentation and Tracking of Multiple Humans in Complex Situations”, CVPR01.

[4] M. Isard and A. Blake. “Condensation – conditional density propagation for visual tracking”. Int. J. Computer Vision, 29(1):5–28, 1998.

[5] M. Isard and A. Blake. “Icondensation: Unifying low-level and high-level tracking in a stochastic framework”. In Proc. European Conf. on Computer Vision, pages 767–781, 1998.

[6] Ying Wu and Thomas S. Huang, "A Co-inference Approach to Robust Visual Tracking", in Proc. IEEE Int'l Conf. on Computer Vision (ICCV'01), Vancouver, July, 2001.

[7] M. Isard and J. MacCormick, “BraMBLe: A Bayesian Multiple-Blob Tracker”, ICCV01.

[8] G. R. Bradski. “Computer Vision Face Tracking as a Component of a Perceptural User Interface”. IEEE Work. On Applic. Comp. Vis., Princeton, pp. 214-219, 1998.

[9] D. Comaniciu, V. Ramesh, P. Meer. “Real-time tracking of non-rigid objects using mean shift”. In Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recog., pp. 142–149, 2000.

[10] D. Comaniciu, V. Ramesh and P. Meer. “Kernel-based object tracking”, IEEE Trans. Pattern Anal. Machine Intell., 25, 564-577, 2003.

[11] S. Birchfield. “Elliptical head tracking using intensity gradients and color histograms”. In Proceedings of CVPR, pages 232--237, 1998.

[12] Y. Chen, Y. Rui and T. S. Huang, JPDAF Based HMM for Real-Time Contour Tracking, CVPR01.

[13] A. Jepson, D. Fleet and T. E1-Maraghi, “Robust Online Appearance Models for Visual Tracking”, CVPR01, pp. 415-422, Vol. 1, 2001.

[14] Paul Viola, Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, in CVPR’01, Hawaii, Dec. 2001.

[15] S. Z. Li, L. Zhu, Z. Q. Zhang, A. Blake, H. Zhang, and H. Shum. ”Statistical learning of multi-view face detection”. In ECCV’02, Copenhagen, Denmark, May 28-June 2 2002.

[16] K. Mikolajczyk, R. Choudhury, and C. Schmid. “Face detection in a video sequence - a temporal approach”. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[17] Z. Liu and Y. Wang. “Face detection and tracking in video using dynamic programming”. In ICIP, 2000.

[18] G. J. Edwards, C. J. Taylor, and T. Cootes. “Learning to identify and track faces in image sequences”. In 8th British Machine Vison Conference, pages 130-139, Colchester, UK, 1997.

[19] K. Murphy. “An introduction to graphical models”. Technical report, Intel Research Technical Report., 2001.

[20] T. A. Stephenson. “An introduction to Bayesian network theory and usage”. IDIAP Research Report 00-03. 2000

[21] R.O. Duda and P.E. Hart, Pattern Classification, (2nd Edition) New York: Wiley, 2000.