Mode-based Multi-Hypothesis Head Tracking Using Parametric Contours

Face and Gesture 2002 submission

Abstract

This paper describes a probabilistic mode-based multi-hypothesis tracking (MHT) algorithm. The modes are the local maximums refined from initial samples in a parametric state space. Because the modes are highly representative, this technique allows us to use a small number of hypotheses to effectively model non-linear probabilistic distributions. To ensure real-time tracking performance, we propose a novel parametric causal contour model and an efficient dynamic programming scheme to refine the initial contours to nearby modes. Furthermore, to overcome the common drawback of conventional MHT techniques, i.e., producing only the maximum likelihood estimates instead of the desired posterior, we introduce the highly effective importance sampling framework into MHT, and develop a novel procedure to estimate the posterior from the importance function. Experiments on a challenging real-world video sequence demonstrate that the proposed tracking technique is both robust in complex environment (e.g., clutter background and partial occlusion) and efficient in computation.

1. Introduction

Many real-world applications require accurate people tracking. For example, the ability to track moving people in video surveillance and video conferencing systems will greatly increase the chance of their adoption. Unfortunately, robust and efficient people tracking in complex environment is still an open research problem. In this paper, we focus our attention on tracking human head/face, one of the most important branches in people tracking. Let X_t denote the tracker state (e.g., location and orientation) of the head and let Z_t be the image observation, both at time *t*. Our goal is to accurately and efficiently compute the posterior probability of $p(X_t|Z_t)$.

In general, there are three approaches to estimating a probability distribution, i.e., pure parametric, pure non-parametric and semi-parametric. The well-known Kalman filter is a good representative of the pure parametric approaches, where the distribution is assumed to be Gaussian. Unfortunately, because of its uni-mode assumption, Kalman filter has only achieved limited success in real-world tracking applications. To overcome this difficulty, Isard and Blake propose a non-parametric approach, i.e., CONDENSATION, where the distribution is represented and estimated by a set of properly positioned and weighted particles [6]. CONDENSATION not only can easily handle multi-mode distributions, it also works in non-linear dynamic systems. However, as a general drawback to all non-parametric algorithms, CONDENSATION requires large number of particles. The required particles also grow exponentially with the dimensionality of the state space. To overcome this difficulty, several improved techniques have been proposed to make the particles more effective. For example, in [5], Deutscher *et. al.* propose an annealed particle filter for tracking articulated human figure. It is based on probabilistic pruning, and focuses its particles in the neighborhood around the global peaks of the weighting function. This method greatly reduces the number of particles needed. But as noted by its authors, it is not a robust Bayesian framework any more. By discarding inferior peaks in the weighting function, it may lose the true state if large distractions occur.

A more promising direction is to use the semi-parametric approaches, where the to-be-estimated distribution is modeled by a mixture of parametric distributions. These semi-parametric approaches retain the capability of representing multi-mode distributions as CONDENSATION does, but with much fewer samples. Because of the many attractive features that the semi-parametric approaches have, we focus our attention on this paradigm in this paper. Multi-hypothesis tracking (MHT) is one of the most successful semi-parametric approaches used in tracking. It is first developed in radar-tracking systems [11] and recently has been applied in articulated human body tracking by Cham and Rehg [3]. MHT works in a parametric state space. Each hypothesis is a particular configuration of parameters in the state space, and the overall state is represented by a mixture of multiple hypotheses.

One limitation with the classic MHT, used in radar tracking, is that it assumes a set of discrete hypotheses is available at any time step. This assumption is totally valid in radar tracking where the goal is to associate multiple detected targets with multiple airplanes. In visual tracking, however, this assumption cannot be met easily [3]. For example, for human head tracking, it is almost impossible to develop a single *high-level* “feature detector” that can detect a set of discrete hypotheses of the head position/pose at every frame. On the other hand, using *low-level* features such as image edges in this scheme will quickly lead to an intractable number of hypotheses. In [3], Cham and Rehg solve this difficulty by first using an appearance-based gradient local search to generate a set of hypotheses (local maximums), and then constructing the likelihood function as a piecewise Gaussian by combining the multiple hypotheses. While this approach has successfully demonstrated the effectiveness of the MHT paradigm, it has three major difficulties.

1. For visual tracking, the appearance/template-based approaches only work with relatively rigid objects and with objects that rarely change orientation and intensity. For head tracking, however, the head orientation and the environment lighting can change from frame to frame, causing head appearance change dramatically.
2. This approach uses an iterative Gauss-Newton method to generate hypotheses, which is computationally expensive and not suitable for real-time tracking.
3. Most importantly, as pointed out in [4], this approach only produces maximum likelihood estimates, but not the desired posterior $p(X_t \mid Z_t)$. This can significantly degrade the tracking performance.

In this paper, we propose various techniques to overcome the above difficulties, and we present an effective head tracking system using the MHT paradigm. The rest of the paper is organized as follows. In Section 2, to overcome the first difficulty, we propose to use parametric contours, instead of the appearance, to model the object-of-interest. This is particularly effective in head tracking, where the head can be well modeled by a parametric ellipse. While the head orientation and lighting can dramatically change the head appearance, the contour of the head remains almost the same shape. Furthermore, to deal with the second difficulty, we propose a novel causal contour model to avoid iterative refinement, enabled by an efficient dynamic programming scheme. In Section 3, we overcome the third difficulty by casting the MHT technique in the importance sampling framework, and show how to effectively estimate the desired posterior $p(X_t \mid Z_t)$. Specifically, we describe how to compute the importance function, the observation likelihood and the transition probability. In Section 4, we apply our proposed head tracking technique on a challenging real-world video sequence and report promising tracking results. Concluding remarks are given in Section 5.

2. Causal Contour Model for MHT

There are two important terminologies in our proposed mode-based MHT. We use "sample" to denote a state space configuration obtained from some prior distribution or prediction scheme. We use "mode" to denote a refined "sample" that corresponds to a local maximum in the distribution. Note that both "sample" and "mode" represent a particular configuration of parameters in the state space. To refine an initial contour (the sample) to the best local contour (the mode), the active contour technique, e.g., [1][8][10][12], has been proved to be a powerful tool. However, in the context of real-time MHT, it has the following difficulties:

1. The mode can only be obtained by an iterative search in the 2D image plane, which is inefficient for real-time tracking.
2. Because the traditional active contour is non-parametric, it can easily be distracted by background clutter, and more importantly, not in a ready-to-use form for MHT.

To address difficulty 1, in Sections 2.1 and 2.2, we propose a novel causal 1D contour model to facilitate efficient sample refinement. To overcome difficulty 2, in Section 2.3, we propose to use a parametric ellipse as the state space, which can easily take domain knowledge (e.g., shape prior) into account to avoid background distraction, and can readily be used in MHT.

2.1 1D contour representation

In our proposed MHT, given a sample, we want to find its corresponding mode, i.e., the best contour within the vicinity. Because of the well-known aperture effect, only the deformations along the normal lines of a contour can be detected. We can therefore restrict the contour searching to the set of normal lines of the contour (see Figure 1). Let ϕ, $\phi=1, \ldots, M$, be the index of the normal lines and λ, $\lambda = -N, \ldots, N$, be the index of pixels along a normal line. Furthermore, let $\rho_\phi(\lambda)$ denote the image intensity at pixel λ on line ϕ. That is, $\rho_\phi(\lambda) = I(x_{\lambda\phi}, y_{\lambda\phi})$, where $(x_{\lambda\phi}, y_{\lambda\phi})$ is the corresponding image coordinate of pixel λ on line ϕ and $I(x_{\lambda\phi}, y_{\lambda\phi})$ is the image intensity.

Each normal line has $2N+1$ pixels, which are indexed from $-N$ to N. The center point of each normal line is placed on the initial contour (the sample) and indexed as 0. Let $c(\phi)$ denote the best local contour (the mode) location on line ϕ. If we can detect all $c(\phi), \phi \in [1, M]$ then we can obtain the best local

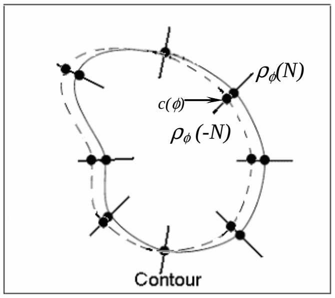


Figure 1: Illustration of the 1D contour model. At frame t, the solid curve is the initial contour (the sample) based on the tracking results at frame t-1. The dashed curve is the best local contour (the mode) that we want to find. A set of measurements are collected along the M normal lines of the initial contour. $c(\phi)$ denotes the best local contour location on line ϕ. The best local contour can be obtained if we can detect all $c(\phi), \phi \in [1, M]$.

contour. Note that instead of representing the contour by a 2D image coordinate, i.e., $(x_{\lambda\phi}, y_{\lambda\phi})$, we can now represent it by a much simpler 1D function $c(\phi)$, $\phi = 1, \ldots, M$.

2.2 Efficient contour refinement

If the initial contour matched the best local contour exactly, the detected contour points on all normal lines would have been exactly at the center, i.e., $c(\phi) = 0, \forall \phi \in [1, M]$. In reality, however, we need to find the best local contour $c(\phi)$ based on the measurements. Like in the traditional active contour model, this is achieved by optimizing an objective function, which favors a smooth contour along pixels having sharp intensity changes. To do the optimization efficiently instead of the slow iterative search, we define the contour smoothness constraint in a causal way (Section 2.2.2). The optimal contour can therefore be found by a single iteration of dynamic programming. The objective function and the optimization procedure are described below.

2.2.1. Edge likelihood term

Contour points are likely to be signified by large color/intensity changes [1][8][10][12]. We therefore choose edge likelihood as a term in the objective function. We represent the edge likelihood in energy form, which is usually called the *external energy*. The edge likelihood of pixel λ on line ϕ, $E_e(\rho_\phi, \lambda)$, can therefore be computed as a function of the image gradient along the direction of the line:

$$E_e(\lambda_\phi) = g\left(-\left|\frac{d}{d\lambda_\phi}\rho_\phi(\lambda_\phi)\right|^2\right) \approx g\left(-\left(\rho_\phi(\lambda_\phi+1) - \rho_\phi(\lambda_\phi)\right)^2\right) \quad (1)$$

where $g(.)$ is an appropriate monotonically increasing function [1][8][10][12].

Considering the initial contour is relatively accurate, we can further refine the objective function by putting a zero-mean Gaussian kernel at the center of the normal line. Therefore, an extra energy term, which favors the edge points in the center part of the normal line is defined as:

$$E_s(\lambda_\phi) = \lambda_\phi^2 / \sigma_s{}^2 \quad (2)$$

where σ_S controls how strong this constraint should be. For example, when the motion of the object is difficult to predict or no accurate motion model can be obtained, the σ_S should be large enough to incorporate uncertainties, and hence lowering the influence of this constraint.

Because the above edge detection scheme only examines each normal line individually, it does not have enough information to ensure good overall contour detection results in cluttered environments. We therefore need to take into account the relationship between contour points on adjacent normal lines. If the normal lines are dense (e.g., 20-60 in our experiments), it is easy to see from Figure 1 that the best local contour points on adjacent normal lines tend to have similar amount of displacement from the initial contour points (indexed as 0 on each normal line). This inter-normal-line correlation can be modeled effectively by the smoothness constraint.

2.2.2. Causal smoothness constraint

The contour smoothness constraint has been used in many contour models [1][8][10][12]. It is achieved by defining an *internal energy* term to penalize the roughness of a contour. In the traditional snake model, the roughness is characterized by the first and second derivatives of the contour. Because the first and second derivatives of the current contour point depend on the contour points both before and after it, this representation of the smoothness constraint is not causal, and the best local contour can only be obtained iteratively [1][8]. For real-time head tracking, it is imperative to have an efficient contour refinement process. Enabled by our 1D contour model, we can easily define the smoothness constraint in a causal way:

$$E_i(\lambda_{\phi-1}, \lambda_\phi) = |\lambda_\phi - \lambda_{\phi-1}|^2 \quad (3)$$

This causal definition allows us to design a very efficient dynamic-programming-based contour refinement process (see Section 2.2.3), and we can obtain the best local contour in a single iteration.

Given all the constraints, the total objective function of a contour $c(\phi)$, $\phi = 1, \ldots, M$, is defined as follows:

$$E(c(\phi)) = \sum_{\phi=0}^{M} \left(\alpha_i E_i(c(\phi-1), c(\phi)) + \alpha_e E_e(c(\phi)) + \alpha_s E_s(c(\phi))\right) \quad (4)$$

where α_i, α_e and α_s are appropriate weights for each of the energy terms. The best local contour is the $c(\phi)$,$\phi = 1, \ldots, M$, that gives the minimum total energy. Because on each normal line there are $2N+1$ locations for $c(\phi)$, a naïve algorithm would require $(2N+1)^M$ tries before finding the best contour. Fortunately, because of the new causal definition of the smoothness constraint, it is possible to find the best local contour efficiently by using a dynamic programming scheme.

2.2.3. Energy minimization: finding the modes

To obtain the best local contour (the mode) using dynamic programming, the optimization process is divided into multiple stages, starting from $\phi = 0$ to $\phi = M$. If the total energy ($E^o(\lambda_\phi)$) of the best contour ending at point λ_ϕ is known, it can be propagated to every point on line $(\phi + 1)$ to compute the total energy for $\lambda_{\phi+1}$ ($E^o(\lambda_{\phi+1})$). This dynamic programming propagation process can be explained as follows:

$$E^o(\lambda_{\phi+1}) = \min_{\lambda_\phi \in [-N,N]} \left\{E^o(\lambda_\phi) + \alpha_i E_i\left(\lambda_\phi, \lambda_{\phi+1}\right)\right\} + \alpha_e E_e(\lambda_{\phi+1}) + \alpha_s E_s\left(\lambda_{\phi+1}\right), \quad \lambda_{\phi+1} \in [-N, N] \quad (5)$$

After the energy is propagated to the last line $\phi = M$, the best contour can be obtained by first finding the minimum energy point on line M, $\min_{\lambda \in [-N,N]} E^o(\lambda_M)$, and then back-tracking through all the lines to obtain the contour points on each line. Note that the computation complexity has reduced from the naïve approach's $(2N+1)^M$ to dynamic programming's $(2N+1)^2 M$.

To summarize, unlike the traditional active contour model, our proposed causal 1D contour model allows us to obtain the best contour without iteratively searching the 2D image plane. Note that the best contour (the mode) is *with respect to* a given initial contour (the sample). If two samples are far from each other, the modes obtained can be quite different, which is exactly what we need in MHT.

2.3 Shape prior: parametric contours

So far we have discussed the contour in a non-parametric form – each individual contour point can move arbitrarily, as long as the overall contour minimizes the objective function (Equations (4) and (5)). This means that a contour can deform to any shape. Because of its high degree of freedom, this non-parametric representation is both susceptible to background clutter and not easily used in MHT. We therefore propose to use the parametric ellipse to represent the contours. First, human head can be very well modeled by a parametric ellipse, regardless of the head orientation [2]. This domain knowledge, i.e., shape prior, can help the contour avoid erroneous evolvement, therefore greatly improving the tracking results (see Figure 4). Second, the parametric ellipse represents an elegant state space whose samples/modes can be readily used in MHT. Specifically, we use a five dimensional parametric ellipse to represent the head contour:

$$X = [x_c, y_c, \alpha, \beta, \Phi] \qquad (6)$$

where (x_c, y_c) is the center of the ellipse, α and β are the lengths of the major and minor axes of the ellipse, and Φ is the orientation of the ellipse. Note that the initial samples are always ellipse. But after the refinement process, the obtained modes may not be ellipses any more. We therefore use the least mean square (LMS) technique to fit the modes to the five-dimensional ellipse state space before tracking.

3. Mode-based Multi-Hypothesis Tracking

As pointed out in Section 1, one of the major limitations with the MHT approach proposed in [3] is that it only produces maximum likelihood estimates, but not the desired posterior [4]. In this section, we present how to estimate the posterior from MHT by using the importance sampling.

3.1 Constructing importance function

Let q be a known proposal distribution (also called the *importance function*). It has been proven [9] that as I tends to infinity, the *unknown* posterior distribution p can be approximated by a set of *properly weighted* particles drawn from the *known* importance function q:

$$\hat{p}(X_t \mid Z_t) = \sum_{i=1}^{I} \pi_t^i \, \delta_{X_t^i}(dX_t) \qquad (7)$$

Where I is the number of particles, δ is the *Dirac* delta function, and the weights for the particles are calculated as:

$$\pi_t^i = \frac{p(X_t^i \mid X_{t-1}^i)}{q(X_t^i \mid X_{t-1}^i, Z_t)} \cdot p(Z_t \mid X_t^i) \qquad (8)$$

The process of drawing particles X_t^i from the importance function q and calculating the particle weights π_t^i is called *importance sampling*. There are infinite number of choices of the importance function, as long as its support includes that of the posterior distribution. But of course, when q is close to the true posterior p, the particles are more effective. The idea is then to put more particles in areas where posterior may have higher density to avoid useless particles [7]. The mode-based MHT fits in this importance sampling framework well.

Before we proceed further, it is beneficial to first define some terminologies. We will use X_t to denote a *general* state variable, as used in Equations (7) and (8). Furthermore, let $\overline{X}_t^k$, $k = 1, \ldots, K$, denote the *raw* samples drawn from a prior distribution, and let $\tilde{X}_t^l$, $l = 1, \ldots, L$, denote the modes refined from the raw samples. Note that because of the refinement process in Section 2.2, the best contour obtained may not be an ellipse any more. Here, we use $\tilde{X}_t^l$ to denote the best contour after fitting the ellipse (Equation (6)).

If we model each mode as a local Gaussian, and use the mixture of the modes as the importance function q, we have:

$$q(X_t \mid X_{t-1}, Z_t) \equiv \frac{1}{L} \sum_{l=1}^{L} N(\tilde{X}_t^l, \sigma_q) \qquad (9)$$

where "$\equiv$" denotes "defined as", and σ_q is the variance of the Gaussian for the modes. Once the importance function q is constructed, we can draw particles $\hat{X}_t^i$, $i = 1, \ldots, I$, from it, and estimate the posterior by using Equations (7) and (8). Note that, to preserve all the L modes in the importance function, the number of particles should be greater than or equal to the number of modes, i.e., $I >= L$.

Given the importance function q (Equation (9)), we can evaluate the probability of a particle $\hat{X}_t^i$ as:

$$q(X_t = \hat{X}_t^i \mid \tilde{X}_t^l) = \frac{1}{\sqrt{2\pi}\sigma_q} \frac{1}{L} \sum_{l=1}^{L} \exp\left(-\frac{(\hat{X}_t^i - \tilde{X}_t^l)^2}{2\sigma_q^2} \right) \qquad (10)$$

Referring to Equation (8), in order to calculate the particle weights, in addition to evaluating Equation (10), we also need to calculate the particle likelihood $p(Z_t \mid \hat{X}_t^i)$ and the particle transition probability $p(\hat{X}_t^i \mid \hat{X}_{t-1})$. We discuss those two terms in the following two sub-sections.

3.2 Calculating the likelihood

Let $Z_{t,\phi}$ denote the edge detection observation on line ϕ at time t. Because of background clutter, there can be multiple edges along each normal line. Let J be the number of detected edges

($Z_{t,\phi}$ = (Z_1, Z_2, ..., Z_J)). Of the J edges, at most one is the true contour. With the assumption that the clutter is a Poisson process along the line with spatial density γ and the true target measurement is normally distributed with standard deviation σ_z, we can obtain the edge likelihood model as follows:

$$p(Z_{t,\phi} \mid \lambda_\phi = \hat{X}^i_{t,\phi}) \propto 1 + \frac{1}{\sqrt{2\pi}\sigma_z q_0 \gamma} \sum_{j=1}^{J} \exp\left(-\frac{(Z_j - \lambda_\phi)^2}{2\sigma_z^2} \right)$$

where q_0 is the prior probability that none of the J edges is the true contour. By assuming independence between different normal lines, we have the following overall likelihood function:

$$p(Z_t \mid \hat{X}^i_t) = \prod_{\phi=1}^{M} p(Z_{t,\phi} \mid \hat{X}^i_{t,\phi}) \qquad (11)$$

3.3 System dynamics and particle transition probability

Similar to [14], we adopt the Langevin process to model the human head movement dynamics:

$$\begin{bmatrix} X_t \\ \dot{X}_t \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ 0 & a \end{bmatrix} \begin{bmatrix} X_{t-1} \\ \dot{X}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} m_t \qquad (12)$$

where a = exp(-$\beta_\theta \tau$), $b = \bar{v}\sqrt{1-a^2}$, β_θ is the rate constant, m_t is a thermal excitation process drawn from Gaussian distribution $N(0, Q)$, τ is the discretization time step and $\bar{v}$ is the steady-state root-mean-square velocity. Assuming that each particle forms a local Gaussian, the particle transition probability can be computed as:

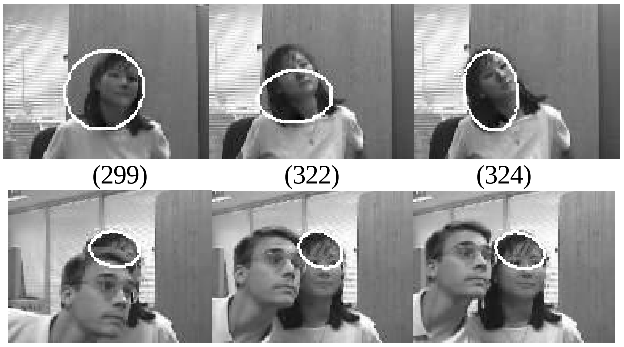
$$p(\hat{X}^i_t \mid \hat{X}_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{I} \sum\nolimits_{r=1}^{I} \exp\left(-\frac{(\hat{X}^i_t - \hat{X}^r_{t-1})^2}{2\sigma^2} \right) \qquad (13)$$

where σ is the variance of the Gaussian kernel.

3.4 The complete algorithm

By formulating MHT in the importance sampling framework, we have derived the desired posterior estimates, rather than the maximum likelihood estimates. We can therefore represent the posterior using the set of particles and propagate them to the next frame. Because the particles are drawn from the mixture of all the modes (i.e., the importance function), this algorithm is more robust than single-hypothesis approaches, and can recover quickly after large distractions. The proposed mode-based MHT technique, cast in the importance sampling framework, is summarized as follows:

1. **Generating importance function:**
   a) Given the particle set obtained at t-1, i.e., { $\hat{X}^i_{t-1}$, π^i_{t-1}, i = $1,\ldots,I$}, draw K raw samples $\bar{X}^k_{t-1}$, $k = 1, \ldots, K$, from the set. Passing the raw samples through the system dynamics (Equation (12)), we obtain the predicted raw samples $\bar{X}^k_t$.
   b) For each raw sample $\bar{X}^k_t$, find the best-fit contour $\tilde{X}^l_t$, i.e., the mode within its neighborhood. A robust and efficient dynamic programming based mode-finding process is explained in detail in Section 2.2. After finding the modes, we generate the importance function using Equation (9).
2. **Importance sampling:**
   a) Draw I particles ($\hat{X}^i_t$, i = $1,\ldots,I$) from the importance function (Equation (9)).
   b) Weight particles using Equations (8),(10),(11) and (13).
3. **Output**:
   Once all the weights are calculated, the probabilistic tracking result can be estimated by this newly obtained particle set { $\hat{X}^i_t$, π^i_t, $i = 1,\ldots,I$ } [6][7].


(299) (322) (324)

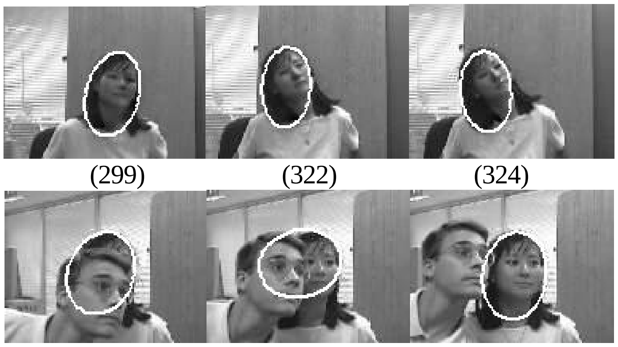
(458) (467) (469)

Figure 2: Tracking results with a single hypothesis. Beginning at frame 299, the tracker is distracted by the sharp edge of the blinder. The distraction becomes larger over time (e.g., frame 322). The tracker resumes tracking again at frame 324 when the person happens to move to the distracting area. In frame 458, the tracker is distracted by occlusion and does not recover afterwards.

4. Application in Human Head Tracking

In the experiment reported in this section, we use 30 normal lines along the ellipse contour, i.e., M = 30. Each line is 21 pixels long, i.e., N = 10, and we use 20 particles during the tracking, i.e., I = 20. The tracking algorithm is implemented in C++ on Windows platform. No attempt is made on code optimization, and the current system runs at 10 frames/sec on a standard PIII 933 PC.

A challenging real-world video sequence in a cluttered


(299) (322) (324)

(458) (467) (469)

Figure 3: Tracking results of MHT. The tracker tracks through out the sequence. In frames 458-467, the tracker is distracted by partial occlusion. But the correct hypothesis emerges in frame 469, and the tracker resumes tracking reliably.

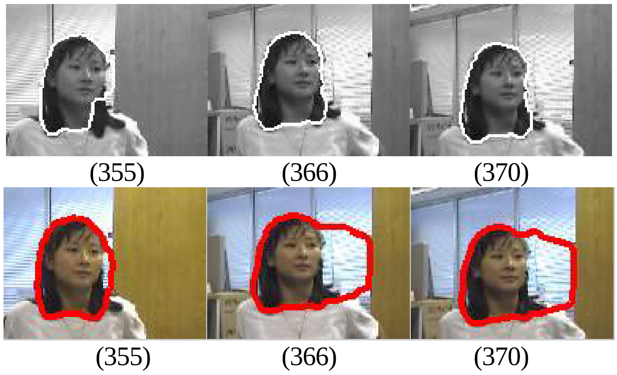
environment with 499 frames is used in the experiment. The sequence simulates various tracking conditions, including appearance changes, quick movement, out-of-plane head rotation, shape deformation, camera zoom in and out, and partial occlusion. Referring to Figures 2-4, note that the blinds and the door (e.g., sharp edges and clutters) impose great challenges to any visual tracking algorithms.

The mode-based MHT enables us to deal with severe distractions. Twenty hypotheses ($I=20$) are enough to successfully track the head throughout the sequence. All the 5 parameters of the ellipse are allowed to change. The tracking results of the single-hypothesis approach and mode-based MHT approach are shown in Figures 2 and 3 for comparison. The single-hypothesis approach is easily distracted, while the MHT is quite robust under various tracking conditions. Without the sample-refining process, it would be almost impossible to track in the 5-dimension parametric state space with only 20 hypotheses.

Furthermore, to understand the importance of the *parametric* contour, we compare our parametric contour model against the traditional non-parametric contour models. Because of the high degree of freedom in the non-parametric contour, i.e., $M = 30$ vs. the 5D ellipse, the local smoothness constraints are not sufficient to assure the global shape and the contour is easily distracted by the background clutter. As shown in Figure 4, when the person moves across the door from right to left, the sharp edges on the blinds and the door severely distract the non-parametric contour. For fine-level comparison purpose, we display the raw contour results instead of the fitted ellipse for both methods.

5. Conclusion

In this paper, a mode-based MHT technique is proposed for head tracking. This technique allows us to use a small number of hypotheses to represent highly non-linear probabilistic distributions. To ensure real-time performance, a novel causal contour model is proposed, and techniques based on an efficient dynamic programming scheme are developed to refine the raw samples to nearby modes (local maximums). In addition, a parametric contour model, i.e., the five-dimensional parametric ellipse, is used for increased stability and to model the shape prior.


(355) (366) (370)

(355) (366) (370)

Figure 4: Comparison of our parametric contour model (top row) and traditional non-parametric contour models (bottom row). With only five degree of freedom, our contour model is much more robust. For the non-parametric contours, the tracker is severely distracted by the sharp edges on the background.

To overcome the common drawback of MHT, i.e., producing only the maximum likelihood estimates instead of the desired posterior, we have further introduced the importance sampling framework into MHT, and developed an effective procedure for estimating the posterior from the importance function. We have tested our proposed technique on a challenging real-world video sequence and reported promising tracking results.

Acknowledgements

This work was supported in part by National Science Foundation Grant CDA 96-24396, EIA 99-75019 and in part by a Gift Grant from Microsoft Research. Also thank Stan Birchfield for allowing us to use the test sequence.

References

[1] A. Amini, T. Weymouth, R. Jain, Using dynamic programming for solving variational problems in vision, *IEEE Trans. PAMI*, vol. 12, no. 9, pp.855-67, 1990.

[2] S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", *Proc. CVPR*, 1998. pp. 232-7

[3] T. Cham, J. M. Rehg, Multiple hypothesis approach to figure tracking, *Proc. IEEE CVPR*, vol 2, p 239-245, 1999

[4] K. Choo and D. Fleet, People tracking using hybrid Monte Carlo filtering, *Proc. IEEE ICCV*, Vancouver, Canada, 2001

[5] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, *Proc. CVPR*, 2000.

[6] M. Isard, A. Blake, CONDENSATION -- conditional density propagation for visual tracking, *Int. J. Computer Vision*, 29, 1, 5--28, (1998)

[7] M. Isard, A. Blake, ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework, *Proc. 5th European Conf. Computer Vision*, 1998, pp.893-908.

[8] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, *Int. J. Comput. Vision*, vol. 1, no. 4, pp.321-331, 1988

[9] R. Merwe, A. Doucet, N. Freitas, and E. Wan, The unscented particle filter, *Technical Report CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, August 2000.

[10] N. Peterfreund, Robust tracking of position and velocity with Kalman snakes, *IEEE Trans.PAMI*, vol.21, no.6, 1999, pp.564-9

[11] D.B. Reid, An algorithm for tracking multiple targets, *IEEE Trans. On Automatic Control*, vol. 24, no. 6, pp. 843-854, 1979.

[12] D. Terzopoulos, R. Szeliski, Tracking with Kalman Snakes, *Active Vision*, A. Blake and A. Yuille eds., MIT Press, 1992

[13] K. Toyama, G. D. Hager, Keeping your eye on the ball: Tracking occluding contours of unfamiliar objects without distraction, *IEEE Inter. Conf. on Intelligent Robots and Systems*, pp354-359, 1995

[14] J. Vermaak, and A. Blake, Nonlinear filtering for speaker tracking in noisy and reverberant environments, *Proc. of IEEE ICASSP*, 2000.