Switching Observation Models for Contour Tracking in Clutter

Ying Wu, Gang Hua, Ting Yu
Department of Electrical & Computer Engineering
Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{yingwu,ganghua,tingyu}@ece.nwu.edu

Abstract

We propose a generative model approach to contour tracking against non-stationary clutter and to coping with occlusions by explicit modelling and inferring. The proposed dynamic Bayesian networks consist of multiple hidden processes which model the target, the clutter and the occlusions. The image observation models, which depict the generation of the image features, are conditioned on all the hidden processes. Based on this framework, the tracker can automatically switch among different observation models according to the hidden states of the clutter and occlusions. In addition, the inference of these hidden states provides self-evaluations for the tracker. The tracking and inferencing are implemented based on sequence Monte Carlo techniques. The effectiveness of the proposed approach to robust tracking and inferring non-stationary clutter and occlusion is demonstrated for a variety of image sequences.

1 Introduction

Contour tracking is challenged by cluttered environments and occlusions. The difficulty lies in the fact that the image measurements, which are used to infer the target, are not produced by the target alone, but jointly by the target and the clutter interference generated by the environment, while there are in general no easy ways to differentiate these two sources. This situation is more apparent and difficult when the contour is partially or completely occluded. Thus it is important to accommodate the interference models (e.g., background models and occlusion models) in tracking. In addition, since the image observations are the sources for both target and interference, naturally, it should be feasible to estimate the clutter interference as well as the interaction between the target and the environment, if modelled appropriately.

Learned background and clutter models [5, 15] can greatly alleviate the interference induced by *stationary* clutter. However, in many applications, the clutter presents non-stationary spatial characteristics, e.g., the degrees of clutter interference may vary in different image regions. Thus using different clutter and occlusion models according to different situations is more appropriate than using a fixed model. The question is how to automatically determine a suitable model to use. In addition, it is important for the tracker to have a mechanism of self-evaluation, e.g., evaluating the environment clutter interference, and knowing if the target is tracked, occluded or lost.

To approach to these questions, this paper presents a generative model approach that enables automatic switching among different image observation models for different clutter characteristics (and degrees of occlusion). Target tracking and tracker self-evaluation can be done simultaneously by inferring the hidden states of these generative models.

Specifically, a class of dynamic Bayesian networks are presented in the paper. In addition to the target dynamics, the models incorporate an hidden process that models the clutter and selects image observation models. In our formulation of contour tracking, the image observations are generated from a product of a Gaussian process and a non-stationary Poisson process. The Gaussian process represents the uncertainty of the model for the target contour, while the non-stationary Poisson process models the clutter from the environment.

Within the same framework, an additional hidden process for occlusions can also be incorporated in the generative models. Different degrees of occlusion are formulated as different weights in the linear mixture of the clutter observation and the joint target-clutter observation. By estimating the weighting process, we can easily infer the degrees of occlusion for evaluating the tracker.

Since the above dynamic Bayesian networks are densely connect graphical models, it is difficult to obtain analytical results for the probabilistic inference. Thus, we approximate the inference by sequential Monte Carol strategies.

The proposed generative model approach naturally combines different hidden factors, i.e., the target, the clutter and the occlusion. Thus, the tracking algorithms based on these models are robust with respect to non-stationary environments and occlusions. In addition, the inference of the hidden states of the clutter and occlusion provides more comprehensive information for online tracking evaluation.

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

The paper is organized as follows. Details of the image observation models for contours are presented in Section 3. Section 4 describes the dynamic Bayesian network for switching different clutter models. The occlusion model and the inferencing of occlusions can be found in Section 5. Section 6 reports a set of experiments for theses generative models, and a brief discussion is given in Section 7.

2 Previous Work

We denote the *target state* at time t by $\mathbf{X}_t$. The task of visual tracking is to infer $\mathbf{X}_t$ based on all the observed image evidence $\underline{\mathbf{Z}}_t = \{\mathbf{Z}_1, \cdots, \mathbf{Z}_t\}$, where $\mathbf{Z}_t$ is the image *measurement* (or *observation*) at time t, i.e., to estimate $p(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$. The tracking process can be viewed as the density propagation [2, 5] from $p(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1})$ to $p(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$, and it is governed by the dynamic model $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ and the observation model $p(\mathbf{Z}_t|\mathbf{X}_t)$, since we have

$$p(\mathbf{X}_t|\underline{\mathbf{Z}}_t) \propto p(\mathbf{Z}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1})d\mathbf{X}_{t-1}.$$

Such a probabilistic dynamic system can be depicted graphically by a dynamic Bayesian network in Figure 1.

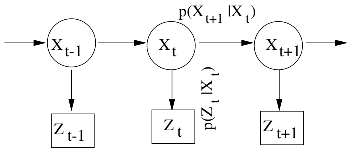

Figure 1: The tracking problem could be represented by a dynamic Bayesian network graphically.

The tracking and identification problems have been well studied for the linear dynamic systems, and the Kalman filtering (KF) technique provides a closed form solution under the assumptions of linear dynamics, linear observations and Gaussian noise. Extended Kalman filters (EFK) can be derived by linearizing the nonlinear dynamics and observations if possible. To approach to the problem of random interference (clutter), the data association approaches have been investigated, e.g., the probabilistic data association filter (PDAF) [1] for tracking a single target in clutter and the joint probabilistic data association filter (JPDAF) [1] for tracking multiple targets. All of these methods imply that the detection (or search) of the target can be easily done, and the the positions of the detected targets are treated as the observations directly. Then, the Kalman innovation, the difference between the predicted observation and the true observation, can be obtained to calculate the Kalman gain to correct the state predictions.

Unfortunately, for contour tracking and many other visual tracking scenarios where the target is more complex and higher level than a point or a line segment, target detection in images is much more difficult than we can assume, since what we can detect directly from images are low-level image features rather than a target candidate itself.

Thus, a special difficulty for visual tracking roots in the matching between the target model (e.g., a parametric shape model) and the noisy image features (e.g., a set of edge points). The search for the target contour from noisy image features is generally quite difficult [3, 8], especially when we need to maintain multiple hypotheses of the target for robust tracking. To alleviate this difficulty, we can embed the search for target candidates into a top-down process consisting of the motion prior prediction step and the observation likelihood correction step as in CONDENSATION [2, 5], where the observation model $p(\mathbf{Z}_t|\mathbf{X}_t)$, which measures the likelihood of the image features, plays a critical role.

It is clear that the image observations are jointly produced by the target and the clutter from the environments. We can assume the presence of the clutter bear a Poisson process with parameter λ encoding the clutter density [1]. It is plausible to use a learned clutter model in calculating the image observation likelihood [12]. Having a clutter model would also allow the discrimination of the sources [11], i.e., to tell if the image observation is generated by the target or the clutter *a posteriori*. Thus the knowledge of the background [14] and the foreground [16] would greatly enhance the robustness for tracking in clutter. To cope with occlusions explicitly, an exclusive principle for modelling the occlusion of multiple known contours has been proposed [12].

However, the clutter interference may present non-stationary characteristics in different image regions, which prevents the tracker from using a single observation model with a preset clutter model. The following sections present a solution of switching multiple observation models for non-stationary clutter and occlusions based on a class of generative models.

3 The Observation Models

The calculation of the observation likelihood $p(\mathbf{Z}|\mathbf{X})$ is critical for contour tracking. Now, the first question we should answer is: *what are the measurements* $\mathbf{Z}$ *for contours?* or *how to model* $p(\mathbf{Z}|\mathbf{X})$ *analytically?*

We follow the idea of using a set of measurement lines to collect image features [2], but end up with a slightly different answer. The length of the measurement line is L. For the i-th measurement line, where $i = 1, \cdots, n$, we denote the predicted contour point position by x_i, so that the contour is discretized by the set of $\{x_i\}$. After applying 1-D edge detection along the measurement line, all the locations of the edge points $\{z_i^1, \cdots, z_i^{m_i}\}$ (m_i is the number of detected feature points) on the measurement line are collected as illustrated in Figure 2(a). Obviously, these features points $\{z_i^1, \cdots, z_i^{m_i}\}$ are jointly produced by the target and the clutter.

However, different from [2], the observation likelihood is not the joint probability of the positions of

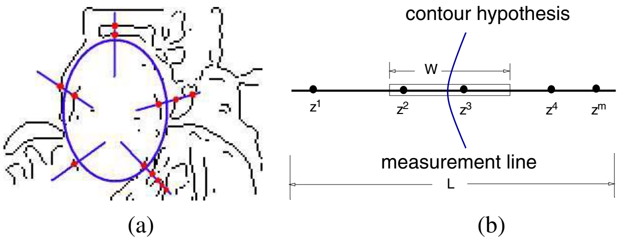

Figure 2: (a) Contour observations on an image. (b) The edge points on a measurement line.

$\{z_i^1, \cdots, z_i^{m_i}\}$, since different measurement lines would have different numbers of such feature points and then the likelihoods for different contour hypotheses would not be comparable. On the other hand, since at most one of the $\{z_i^k\}$ is produced by the target point x_i, we should be concerned about its position. And the other feature points are associated with the clutter which are modelled by a Poisson process, thus we should be only concerned about their numbers instead of their actual positions. In this sense, the likelihoods calculated for different contour hypotheses can be compared. Therefore, the observations for calculating the likelihood consist of (a) the position of one edge point (which is not known yet), and (b) the number of detected feature points.

We assume that the detected feature points associated with the clutter distribute along the measurement line and bear a Poisson process $\mathcal{N}(m : \lambda)$, where m is the number of features. If m features are associated with clutter, then

$$p(m(L)|\mathcal{C}) = \mathcal{N}(m(L) : \lambda) = \frac{\lambda L^m}{m!} e^{-\lambda L}, \qquad (1)$$

where $m(L)$ is the number of features along the measurement line, and $\mathcal{C}$ denotes the clutter.

We also assume the feature points associated with the target contour point x_i is produced by a Gaussian distribution $\mathcal{G}(z : x_i, \sigma)$ within a window W, i.e.,

$$p(z|x_i) = \mathcal{G}(z - x_i : 0, \sigma), \qquad (2)$$

where z is a feature point located inside W. We denote the features inside the window W by $\mathbf{z}_i(W) = \{z_i^1, \cdots, z_i^{m(W)}\}$. But at most one of these features should be associated with the target. We denote it by $z_i^*(W)$.

Therefore, different from [2], for the i-th measurement line, the observations Z_i consist of the number of detected features $m(L)$, and the location of the feature z_i^* in the window W, i.e., $Z_i = \{z_i^*(W), m(L)\}$ as shown in Figure 2(b).

Since we can not determine which feature inside W should be associated with the target, we integrate all the possibilities. In addition, since accommodating the missing detection of the feature associated with the target can make the tracker more robust, we denote events: $\psi_0 = \{z_i^*(W)$ is miss detected$\}$, and $\psi_1 = \{z_i^*(W)$ is detected$\}$. Conditioned on ψ_0, the likelihood is set as:

$$p(Z_i|x_i, \mathcal{C}, \psi_0) = \mathcal{G}(W/4 : 0, \sigma)\mathcal{N}(m(L)). \qquad (3)$$

Similarly, the likelihood $p(Z_i|x_i, \mathcal{C}, \psi_1)$ is

$$\mathcal{G}(W/2 : 0, \sigma)\mathcal{N}(m(L)); \qquad m(W) = 0 \qquad (4)$$

$$\frac{\sum_{z_i^k \in W} \mathcal{G}(z_i^k : x_i, \sigma)}{m(W)} \mathcal{N}(m(L) - 1). \quad m(W) \neq 0 \qquad (5)$$

where $m(W)$ and $m(L)$ denote the number of detected features inside the window W and the measurement line L, respectively. Therefore,

$$p(Z_i|x_i, \mathcal{C}) = p(Z_i|x_i, \mathcal{C}, \psi_0)p(\psi_0) + p(Z_i|x_i, \mathcal{C}, \psi_1)p(\psi_1).$$

Since we assume that the measurement lines are independent, the target observation models are:

$$p(\mathbf{Z}|\mathbf{X}, \mathcal{C}) = \prod_{i=1}^{n} p(Z_i|x_i, \mathcal{C}). \qquad (6)$$

4 Switching Observation Models

Since the feature detectors are unable to differentiate the associations of the detected image features, i.e., which features should be associated with the target and which should not, we face a situation where the observation likelihood models have to take both into account jointly, and the likelihood probability has to be conditioned on both target $\mathbf{X}$ and the clutter $\mathcal{C}$. In our contour observation models, the clutter $\mathcal{C}$ is characterized by λ, the parameter of the Poisson process. Different λs reflect different clutter.

If the clutter $\mathcal{C}$ is spatially stationary, i.e., λ does not change too much, we can obviously treat λ as a fixed parameter and the observation model $p(\mathbf{Z}|\mathbf{X}, \mathcal{C})$ reduces to $p(\mathbf{Z}|\mathbf{X} : \lambda)$. The value of λ affects the likelihood and thus affects the posterior density $p(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$. Once a λ is preset by learning from training sequences [2], the observation model reflects a class of backgrounds whose clutter can be characterized by the λ.

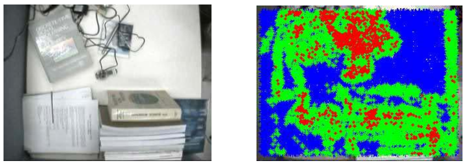

Figure 3: The image contains three cluttered regions.

However, we always encounter the situation where different regions of the environments generate different clutter, e.g., some part of the background is quite clean while other areas are rich in terms of image edges (See Figure 3). It means that the clutter $\mathcal{C}$ is not spatially stationary. Presetting a single λ for the observation model would make the tracker sensitive to different backgrounds. We often observe that a discrete set of clutter models are good enough to capture complex environments. Thus, we need to have a mechanism to switch among different observation models for different clutter. For example, in our experiments, we train three clutter models ($N_\beta = 3$): light ($\beta = 1$), medium ($\beta = 2$) and heavy ($\beta = 3$). Different values of λs are trained for different βs. Our scheme is shown in Figure 4.

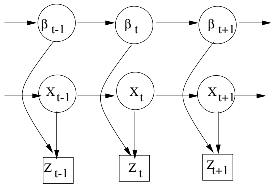

Figure 4: The observations are produced jointly by the target and the clutter. β is a hidden variable which switches different observation models.

Comparing with Figure 1, the dynamic Bayesian network in Figure 4 introduces one more hidden Markov process $\{\beta_t\}$, which switches different observation models according to different classes of clutter. In this approach, $\beta_t \in \{1, \cdots, N_\beta\}$ is a discrete variable to indicate which clutter model is selected. The different clutter models are learned from training sequences. (See Section 6.)

Based on the graphical model in Figure 4, tracking becomes the propagation of $p(\mathbf{X}_t, \beta_t|\underline{\mathbf{Z}}_t)$, and we have

$$p(\mathbf{X}_{t+1}, \beta_{t+1}|\underline{\mathbf{Z}}_{t+1}) \propto \\ p(\mathbf{Z}_{t+1}|\mathbf{X}_{t+1}, \quad \beta_{t+1})p(\mathbf{X}_{t+1}, \beta_{t+1}|\underline{\mathbf{Z}}_t), \quad (7)$$

where $p(\mathbf{X}_{t+1}, \beta_{t+1}|\underline{\mathbf{Z}}_t) =$

$$\int\int p(\mathbf{X}_{t+1}|\mathbf{X}_t)p(\beta_{t+1}|\beta_t)p(\mathbf{X}_t, \beta_t|\underline{\mathbf{Z}}_t)d\mathbf{X}_t d\beta_t, \quad (8)$$

where $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ describes the dynamic model of the target and $p(\beta_{t+1}|\beta_t)$ stipulates the transition which is specified by a finite state machine $\mathbf{T}_\beta$, i.e.,

$$\mathbf{T}_\beta = [T_\beta(i,j)] = [p(\beta_j|\beta_i)], \qquad i,j \in \{1, \cdots, N_\beta\}.$$

The structure of this densely connected graphical model in Figure 4 is complex for straightforward analytical inference. Thus, we approximate the probabilistic inference based on sequential Monte Carlo techniques [4, 9, 10]. The posterior density $p(\mathbf{X}_t, \beta_t|\underline{\mathbf{Z}}_t)$ is represented by a set of weighted particles $\{x_t^{(n)}, \beta_t^{(n)}, \pi_t^{(n)}\}$. Then the estimates $\hat{\mathbf{X}}_t$ and $\hat{\beta}_t$ are given by:

$$\hat{\beta}_t = \arg\max_k \sum_{n \in \mathcal{B}_k} \pi_t^{(n)}; \qquad \hat{\mathbf{X}}_t = \frac{\sum_{n \in \mathcal{B}_k} x_t^{(n)} \pi_t^{(n)}}{\sum_{n \in \mathcal{B}_k} \pi_t^{(n)}},$$

where $\mathcal{B}_k = \{n|\beta_t^{(n)} = k\}$.

The graphical model in Figure 4 is related to the co-inference model [18], since in both cases the observations are jointly determined by a set of factors. But our case does not involve as high dimensionality as the case in [18]. Certainly, the co-inference algorithm can apply here.

In our experiments, we found that each type of clutter is generally associated with a range of λ values instead of a fixed value. To make the switching more flexible, we should allow the uncertainties in λ_t associated with β_t, instead of keeping a set of fixed values. Since the change of the clutter could be dramatic, e.g., the target moves from a light clutter to a heavily cluttered region, we modelled it by a hidden Markov model $\{\beta_t, \lambda_t\}$, where β_t is the discrete variable as before, but λ_t is continuous and it is the output of β_t. The graphical model for this HMM-driven observation model switch is illustrated in Figure 5. The transition of β_t trigs the switch of the observation models. The output of the HMM $\{\beta_t, \lambda_t\}$ is the λ_t, the parameter controlling the clutter observations. We model the output probability $p(\lambda_t|\beta_t)$ of the HMM as a Gaussian model.

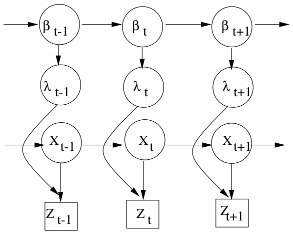

Figure 5: The switch of the observation models is driven by a hidden Markov model $\{\beta_t, \lambda_t\}$.

Since the exact inference of such a Bayesian network in Figure 5 is difficult, we still approximate the inference by the sampling-based sequential Monte Carlo techniques. The algorithm of the HMM-driven observation model switching is elaborated in Figure 6.

Our approach is also different from the idea of the mixed-state tracker [6, 13], which switches among different motion models. To see the difference, a graphical representation of the mixed-state tracker can be illustrated in Figure 7. Our approach is symmetric to the mixed-state approach by switching the observation models.

5 Inferring Occlusion

Is the observation likelihood uniquely determined once conditioned on the target state $\mathbf{X}_t$ and the clutter model?

It is true if there is no occlusion. However, when the target is partially or fully occluded in the environment, the image observation likelihood has a different story, since the occluded part of the target becomes invisible and should not produce image features anymore. Thus, the occlusion introduces more hidden factors into the generative model for

Generate $\{x_{t+1}^{(n)}, \beta_{t+1}^{(n)}, \lambda_{t+1}^{(n)}, \pi_{t+1}^{(n)}\}$ from $\{x_t^{(n)}, \beta_t^{(n)}, \lambda_t^{(n)}, \pi_t^{(n)}\}$.

1. `Re-sampling.` Resample $\{x_t^{(n)}, \beta_t^{(n)}, \lambda_t^{(n)}\}$ to produce $\{x_t'^{(n)}, \beta_t'^{(n)}, \lambda_t'^{(n)}\}$ based on $\{\pi_t^{(n)}\}$.
2. `Prediction.` For each $(x_t'^{(n)}, \beta_t'^{(n)}, \lambda_t'^{(n)})$:
   (a) sample the density of the target dynamics $p(x_{t+1}|x_t)$ to produce $x_{t+1}^{(n)}$;
   (b) sample the transition density $p(\beta_{t+1}|\beta_t)$ to produce $\beta_{t+1}^{(n)}$;
   (c) sample the HMM (β_t, λ_t) observation process $p(\lambda_{t+1}|\beta_{t+1})$ to produce $\lambda_{t+1}^{(n)}$.
3. `Correction.` Re-weight each particle by calculating the likelihood $\pi_{t+1}^{(n)} = p(\mathbf{Z}_{t+1}|x_{t+1}^{(n)}, \lambda_{t+1}^{(n)})$. Then normalize all the new weights so that $\sum_n \pi_{t+1}^{(n)} = 1$.

Figure 6: The algorithm for the HMM-driven observation model switching.

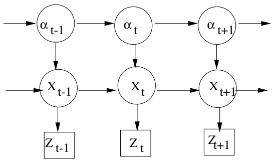

Figure 7: A hidden variable α is used to indicate different motion models. The $\{\alpha_t\}$ process switch the target state process $\{\mathbf{X}_t\}$.

image observation. A question of great interests is: can we infer the occlusion situation from the image sequences? To answer it, we need to model $p(\mathbf{Z}|\mathbf{X}, \mathcal{C}, \mathcal{O})$, where $\mathcal{O}$ indicates the occlusion factor.

To take the occlusion into account, we introduce an event decomposition: $\phi_0 = \{z_i^*(W) \text{ is invisible}\}$, and $\phi_1 = \{z_i^*(W) \text{ is visible}\}$, where $z_i^*(W)$ is the features associated with the target on the i-th measurement line. Thus, we can write the likelihood conditioned on occlusion:

$$p(Z_i|x_i, \mathcal{C}, \phi_1) = p(Z_i|x_i, \mathcal{C}), \quad (9)$$

$$p(Z_i|x_i, \mathcal{C}, \phi_0) = p(Z_i|\mathcal{C}). \quad (10)$$

Thus, the likelihood $p(\mathbf{Z}|\mathbf{X}, \mathcal{C}, \mathcal{O})$ conditioned on the occlusion $\mathcal{O}$ can be modelled as:

$$p(\mathbf{Z}|\mathbf{X}, \mathcal{C}, \mathcal{O}) = \prod_{i=1}^{n} \{p(Z_i|\mathcal{C})\gamma + p(Z_i|x_i, \mathcal{C})(1-\gamma)\}, \quad (11)$$

where

$$p(Z_i|\mathcal{C}) = \frac{1}{W}\mathcal{N}(m(L) : \lambda), \quad (12)$$

and $0 \le \gamma \le 1$ models the degree of occlusion, e.g., $\gamma = 0$ means no occlusion, while $\gamma = 1$ means complete occlusion. The denominator W in Equation 12 is induced because $Z_i = \{z_i^*(W), m(L)\}$, and $z_i^*(W)$ is now uniformly distributed in the window W since z_i^* is invisible, and all the feature points are associated with the clutter.

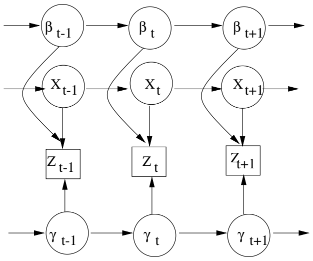

Figure 8: The occlusion process $\{\gamma_t\}$ contributes another hidden Markov chain in the generative model.

As a hidden factor, occlusion introduces another continuous Markov process $\{\gamma_t\}$ in the generative model as illustrated in Figure 8, compared with the model in Figure 4. Then, the observation should be conditioned on the occlusion γ_t additionally, i.e., $p(\mathbf{Z}|\mathbf{X}, \beta, \gamma)$. A different γ switches to a different observation model according to a different degrees of occlusion. The dynamics for the process $\{\gamma_t\}$ is modelled by a Gaussian random walk, i.e.,

$$p(\gamma_t|\gamma_{t-1}) = \begin{cases} 0; & \gamma_t < 0; \\ \mathcal{G}(\gamma_t : \gamma_{t-1}, \sigma_\gamma); & 0 \le \gamma_t \le 1 \\ 1. & \gamma_t > 1 \end{cases} \quad (13)$$

Given the complexity of the graphical model, the inference of occlusion is also approximated by sequential Monte Carlo. The algorithm is straightforward to augment the particle by one more variable γ_t. The transition of γ_t is sampled from $p(\gamma_{t+1}|\gamma_t)$, and the estimate of γ_t is approximated by:

$$\hat{\gamma}_t = E[\gamma_t|\underline{\mathbf{Z}}_t] = \sum_{i=1}^{n} \gamma_t^{(n)}\pi_t^{(n)}. \quad (14)$$

The estimates of γ_t would roughly reveal if the occlusion occurs (ends) and the degrees of occlusion.

When the target is occluded, the tracker would be following the occluding environments, although some hypotheses for the occluded target can be kept based on its motion trajectories. During the occlusion, these hypotheses will become weaker and weaker, since no image evidence can be used to support the existence of the target, and the uncertainty of the occluded target increases and thus enlarges the search area for the target candidates. It eventually becomes a detection (or re-initialization) problem if the duration of occlusion is long. Therefore, occlusion can be viewed as a temporary loss track.

Then we differentiate three situations indicated by a discrete variable α_t: *"target is locked"* where γ_t tends to have small values , *"target is partially occluded"* where γ_t should uniformly distributed in a large range, and *"target is lost"* where γ_t likely to have large values. Interestingly, a HMM $\{\alpha_t, \gamma_t\}$ can be used to modelled this hidden relation by specifying $p(\alpha_{t+1}|\alpha_t)$ and $p(\gamma_t|\alpha_t)$. Different values of α_t output different distributions of γ_t, and the hidden variable α_t indicates different status of the tracker. Then, it is natural to incorporate the occlusion HMM into the generative model to drive the model switch. The model in Figure 8 can be augmented to the model in Figure 9, where the process $\{\gamma_t\}$ is replaced by the HMM $\{\alpha_t, \gamma_t\}$. By inferencing α_t, we can have a rough online evaluation for the tracker.

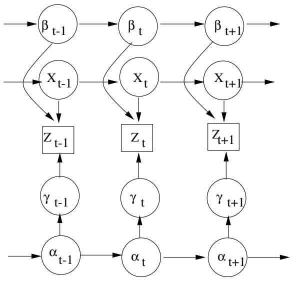

Figure 9: Incorporating the occlusion HMM $\{\alpha_t, \gamma_t\}$ into the generative model to drive the switch of observation models.

6 Experimental Results

Since the contours were roughly round-shaped in all of our experiments, we employed a conics model to simplify the shape representation [1], i.e., $\mathbf{y}'A\mathbf{y}' + 2B\mathbf{y} + C = 0$. A shape template was initialized by conics fitting. The deformation of the shape was governed by an affine transformation,

$$\mathbf{y}' = A\mathbf{y} + \mathbf{t} = \begin{bmatrix} a_{11} \; a_{12} \\ a_{21} \; a_{22} \end{bmatrix} \mathbf{y} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}.$$

Following the shape space [2], given a shape template, a contour can be represented by the affine parameters, i.e., $\mathbf{X} = (a_{11}, a_{12}, a_{21}, a_{22}, t_1, t_2)'$. The dynamic model $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ of the target contour was assumed a constant acceleration model.

To measure the likelihood of the features given a contour hypothesis, a set of m measurement lines were cast along the contour (here $m = 15$). The length L of the measurement line was 20 pixels. The standard deviation σ in Equation 2 was set to 2 pixels in our experiments.

6.1 Inferencing Clutter

Contour tracking and clutter inferencing were handled by the method shown in Figure 5 and 6, i.e., the HMM-driven model switching. In our experiment, the scene consisted of three distinguishable types of clutter, each of which was parameterized by an individual Poisson distribution, then $\beta_t = k, k \in \{1, 2, 3\}$ indicated the switch of the k-th clutter model with parameter λ_k.

The parameters λ of these Poisson processes were learnt from a set of training images. A measurement line with length L was thrown to the images $20,000$ times, and the number of detected edge points on the line was collected each time. Then the k-means clustering was performed to learn the values of λ for three dominate clusters. The output density $p(\lambda|\beta = k)$ was modelled by a Gaussian density $\mathcal{G}(\lambda : \bar{\lambda}_k, \sigma_{\lambda k})$. These parameters were learnt from the clustering. Specifically, $\bar{\lambda}_1 = 0.01, \sigma_{\lambda 1} = 0.002$, $\bar{\lambda}_2 = 0.0824, \sigma_{\lambda 2} = 0.01$, $\bar{\lambda}_3 = 0.2144, \sigma_{\lambda 3} = 0.05$.

We used a finite state machine (FSM) to model the state transition $p(\beta_{t+1}|\beta_t)$ of the switching process $\{\beta_t\}$. The FSM parameters were manually set:

$$\mathbf{T}_\beta = p(\beta_j|\beta_i) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}. \quad (15)$$

We tried different parameters for $\mathbf{T}_\beta$ and found it was not sensitive and could tolerant large inaccuracy. That was the reason that we used manually set parameters.

The results of using the HMM can be seen in the sequence named as "`mclutter.mpg`"[2]. Different βs are shown in different colors (G/B/R is for 1/2/3 respectively). Some sample frames are shown in Figure 10. In this result, when the target was in a relatively clean background, the tracker gave an estimate of $\hat{\beta}_t = 1$ as expected. When the target moved to a region with medium clutter, the tracker automatically switched to an appropriate clutter model and show $\hat{\beta}_t = 2$. Although the target visited the heavily cluttered regions less frequently than the other two, the switch to the model $\beta_t = 3$ was still observed in our experiments. In addition, the recovered switch processes $\{\hat{\beta}_t\}$ and $\{\hat{\lambda}_t\}$ are shown in Figure 11. We can see that the curves do reflect the clutter model transitions for different cluttered regions.

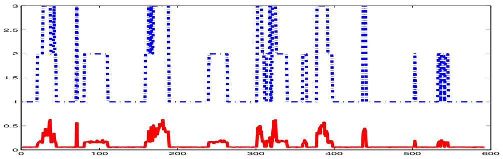

Figure 11: The recovered switching processes $\{\hat{\beta}_t\}$ and $\{\hat{\lambda}_t\}$.

[1] Of course, our methods are also applicable to complex shapes using B-spline representations as in [2].

[2] All the results can be accessed from http://www.ece.nwu.edu/~yingwu

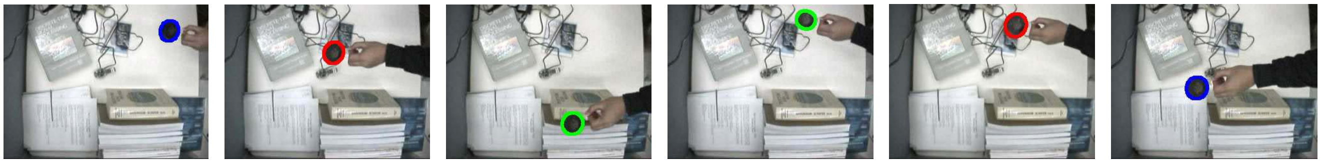

Figure 10: Different clutter situations are inferred in addition to the tracking, and the tracker switches to the corresponding observation models accordingly. (B/G/R represents $\beta = 1/2/3$ respectively.) (See `mclutter.mpg` for details.)

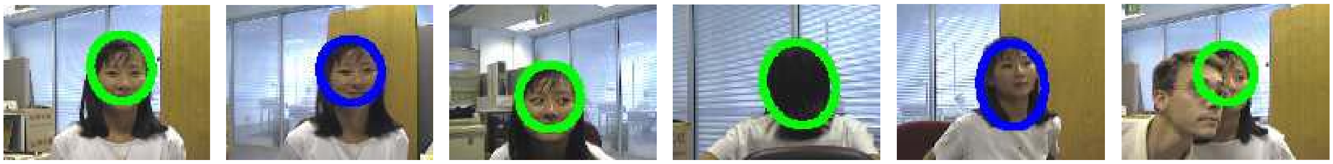

Figure 12: The switching of different cluttered regions are clearly shown in the sequence `girl.mpg`.

We also compared this algorithm with the one without using the HMM. The transition of β_t was modelled the same as Equation 15. By allowing continuous λ_ts, the method using HMM provided better tracking results than the one without, since the scene contained many small regions that can not be well represented by a fixed set of λs as in the method without the HMM.

This HMM-driven switching was also applied to another sequence "`girl.mpg`", which clearly showed that the switching of clutter models in different backgrounds, e.g., the door area and the blind window area. Sample images are shown in Figure 12.

6.2 Inferencing Occlusion

The algorithm in Figure 8 was applied to infer the degrees of occlusion. Since the occlusion process $\{\gamma_t\}$ is a continuous process, the dynamics $p(\gamma_{t+1}|\gamma_t)$ is modelled as in Equation 13, and $\sigma_\gamma = 0.1$.

The results can be seen in the two sequences named as "`occlusion_1.mpg`" and "`occlusion_2.mpg`". Some sample frames are shown in Figure 13. Different colors show the starting and ending of occlusion (B/G/R shows γ from low to high). In addition, the recovered occlusion process $\{\hat{\gamma}_t\}$ is shown in Figure 14. We can see from Figure 14, the curve of γ_t does reflect the evolution of the degrees of occlusion in the sequence where the target moves into, underneath and out from an occluding object, as expected in our theory.

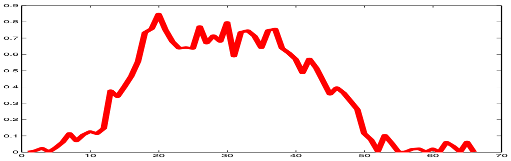

Figure 14: The recovered occlusion process $\{\hat{\gamma}_t\}$ of occlusion_1.

6.3 On-line Tracking Evaluation

The algorithm in Figure 9 was applied to provide a rough on-line tracking evaluation, i.e., to tell whether or not the target is locked, partially occluded or lost. The parameters of the transition of $\{\alpha_t\}$ were manually set:

$$\mathbf{T}_\alpha = p(\alpha_j|\alpha_i) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}.$$

And the output density $p(\gamma|\alpha = k)$ was modelled by a uniform density $\mathcal{U}([a_k, b_k])$. There parameters were set as $a_1 = 0.0, b_1 = 0.2, a_2 = 1, b_2 = 0.9, a_3 = 0.8, b_1 = 1.0$.

The result is in "`eval.mpg`". Some sample frames are shown in Figure 15. Different colors show the status of a tracker. B/G/R represents $\alpha = 1/2/3$. In addition, the recovered processes $\{\hat{\alpha}_t\}$ and $\{\hat{\gamma}_t\}$ are shown in Figure 16.

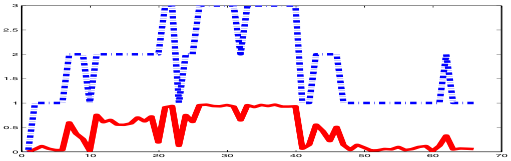

Figure 16: The recovered occlusion process $\{\hat{\alpha}_t\}$ and $\{\hat{\gamma}_t\}$.

7 Discussion and Conclusions

Since the detected image features are jointly produced by both the target and the environment, the observation likelihood model plays a key role in tracking. When the environment clutter presents different characteristics in different regions, using a single observation model is not appropriate. Taking into account of different clutter models and degrees of occlusion in the proposed models, more accurate tracking results have been achieved. Additionally, with the proposed

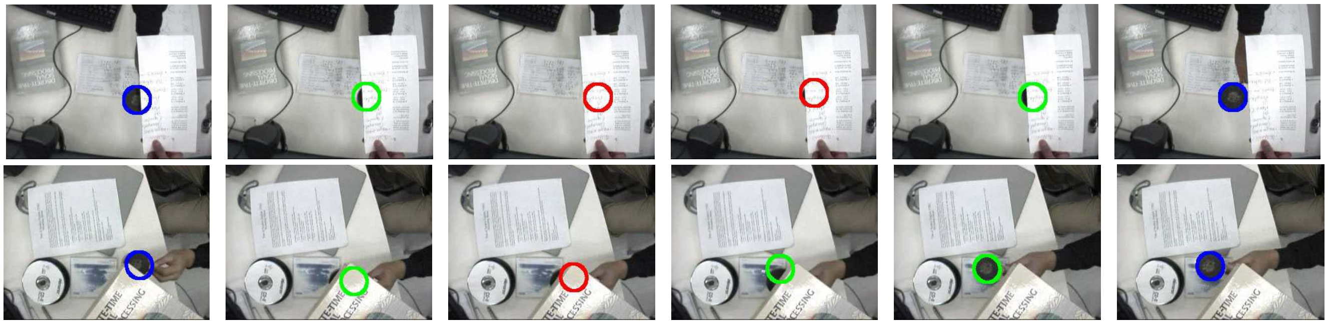

Figure 13: Since the degrees of occlusion are inferred, the tracking of the occluded contour is very robust due to the use of more accurate observation models. B/G/R represents no/partial/full occlusion, respectively. (See `occlusion_1.mpg` and `occlusion_2.mpg`.)

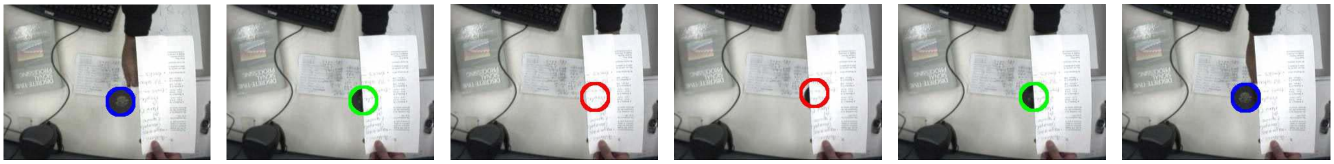

Figure 15: The tracker is able to tell if the target is tracked (in blue), partially occluded (in green), or lost (in red). (See `eval.mpg`)

dynamic Bayesian network models, the tracker begins to have a capacity of self-evaluation by estimating clutter interference and occlusions.

The generative model approaches have demonstrated their effectiveness in many applications such as video modelling [7], examplar-based tracking [17], etc. The learning tasks, i.e., identifying the model parameters, are challenging and computationally expensive. Our future work include the development of efficient learning algorithms for the generative models described in this paper.

Acknowledgments

This work was supported in part by Northwestern startup funds for YW and the Murphy Fellowships for GH and TY. We also thank the anonymous reviewers for their valuable comments.

References

[1] Yaakov Bar-Shalom and Thmoas Fortmann. *Tracking and Data Association*. Academic Press, Orlando, FL, 1988.

[2] Andrew Blake and Michael Isard. *Active Contours*. Springer-Verlag, London, 1998.

[3] Andrew Blake and Alan Yuille, editors. *Active Vision*. MIT Press, Cambridge, MA, 1992.

[4] Arnaud Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[5] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conf. on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[6] Michael Isard and Andrew Blake. A mixed-state condensation tracker with automatic model-switching. In *Proc. of IEEE Int'l Conf. on Computer Vision*, pages 107–112, India, 1998.

[7] Nebojsa Jojic, Nemanja Petrovic, Brendan Frey, and Thomas S. Huang. Transformed hidden Markov models: Estimating mixture models and inferring spatial transformations in video sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000.

[8] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Int'l J. Computer Vision*, 1:767–781, 1988.

[9] Jun Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93:1032–1044, 1998.

[10] Jun Liu, Rong Chen, and Tanya Logvinenko. A theoretical framework for sequential importance sampling and resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2000.

[11] John MacCormick and Andrew Blake. A probabilistic contour discriminant for object localization. In *Proc. of IEEE Int'l Conf. on Computer Vision*, 1998.

[12] John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 572–578, Greece, 1999.

[13] Vladimir Pavlovic, James Rehg, Tat-Jen Cham, and Kevin Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume I, pages 94–101, Corfu, Greece, Sept. 1999.

[14] Jens Rittscher, Jien Kato, Sebastien Joga, and Andrew Blake. A probabilistic background model for tracking. In *Proc. of European Conf. on Computer Vision*, pages 336–350, 2000.

[15] Josephine Sullivan, Andrew Blake, Michael Isard, and John MacCormick. Object localization by Bayesian correlation. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 1068–1075, 1999.

[16] Josephine Sullivan, Andrew Blake, and Jens Rittscher. Statistical foreground modelling for object localisation. In *Proc. of European Conf. on Computer Vision*, pages 307–323, 2000.

[17] Kentaro Toyama and Andrew Blake. Probabilistic tracking in a metric space. In *Proc. IEEE Int'l Conf. on Computer Vision*, Vancouver, Canada, July 2001.

[18] Ying Wu and Thomas S. Huang. Robust visual tracking by co-inference learning. In *Proc. IEEE Int'l Conference on Computer Vision*, volume II, pages 26–33, Vancouver, July 2001.