Multi-scale Visual Tracking by Sequential Belief Propagation

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

Abstract

A novel statistical method is proposed in this paper to overcome abrupt motion for robust visual tracking. Existing tracking methods that are based on the small motion assumption are vulnerable to abrupt motion, which may be induced by various factors, such as the unexpected dynamics changes of the target, frame dropping and camera motion, etc. Although with computational benefits, methods based on hierarchical search is inadequate to this problem because the propagation of the searching error may end up with bad estimates in fine scales. Since different scales contain different salient image features, we propose a new formulation in which searching and matching will be done collaboratively in different scales. The theoretical foundation of this new approach is based on dynamic Markov networks, where the bi-directional propagation of the belief of the target's posteriors on different scales reveals the collaboration among them. A nonparametric sequential belief propagation algorithm for the dynamic Markov network is developed by implementing the collaboration of a set of particle filters. Extensive experiments have demonstrated the effectiveness and efficiency of the proposed method to cope with various types of abrupt motions.

1. Introduction

Visual tracking involves many basic computer vision problems. It infers the target states based on image observations at each time instant by searching and matching in the video. Narrowing down the searching range to facilitate efficient tracking, in general, the small motion assumption is made or accurate dynamic models are assumed in advance. In practice, however, we often encounter situations where unexpected abrupt motion between consecutive image frames invalidates the small motion assumption, or prevents the use of accurate dynamic models, and thus fails the trackers instantly.

Some abrupt motions are due to the target itself. For example, the target's dynamics may be changed by an unexpected outer force such as in the case of a bouncing ball; and the target may also intentionally change its dynamics such as the cobra maneuver of a jet fighter. In addition, some other abrupt motions are induced by the video sensors, e.g., frame dropping in video grabbing and shaking cameras. Since these factors are common in practice, dealing with abrupt motion can make visual tracking algorithms more robust.

It is indeed very difficult to cope with abrupt motion due to the large motion uncertainty, a direct but naive solution is to simply enlarge the searching range to make sure it covers motion uncertainty. However, this is not appropriate due to the polynomial increase of the searching volume, which may still demand tremendous computation especially when the dimension of the target state is more than 2. Although the Kalman filters can adaptively change the searching range based on covariance of target state prediction, they are not adequate to solve the abrupt motion problem as well, since such a prediction capacity mainly depends on the employed dynamic models which may be under unexpected changes in real scenarios. To make the dynamic model as accurate as possible, many sophisticated methods have been investigated such as to employ good dynamic models that are learned from training data [1] and to design automatic switching schemes to switch among several predefined dynamic models [2]. However, in practice, it is generally difficult to learn accurate dynamic models for tracking and prediction, although the learned models may be useful for recognition. In addition, the learned models largely depend on the training data. Switching dynamic models is a good strategy, however, specific prior knowledge on the switching models makes this approach less scalable.

Nevertheless, the motion uncertainty may be approached efficiently by using multiscale strategy. The hierarchical search strategy aims at efficient search, since the results in large scales may be refined by that in small scales. This strategy has been widely used in stereo matching and flow computation. Recently, it has been explored in particle filtering techniques for human body tracking [3] and hierarchical face alignment [4]. A potential problem of the above methods is the accumulation of searching error which prop-

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**

agates from large scales to small scales, and the mechanism that rectifies the error in such a hierarchical strategy is very limited. Thus, if the results on the large scales are noisy or even wrong, the final result on the smallest scale will largely deviate from the truth.

Integrating multiple scales may result in more robust tracking, since different image features will be more salient in different scales. For example, we see textures of a tree in large scale and structures of the twigs and leaves in small scale [5]. Conventional way of integrating multiscale visual information is to vectorize them for visual inference, but this is only true when the image observations in different scales are conditional independent which may not be true for multiscale visual features.

In this paper, we propose a new method to conquer the abrupt motion difficulty by providing a new and rigorous formulation for visual tracking through multiple scales. The theoretic foundation of the new approach is based on a dynamic Markov network, where target states in different scales are represented as different but correlated random vectors and image observation in one scale is only associated with the target state in the same scale. The searching processes in each scale will interact with each other to form a collaborative searching scheme which largely alleviates the error accumulation problem as indicated in the hierarchical search methods. An efficient sequential belief propagation algorithm is proposed and the Monte Carlo implementation is used to perform the Bayesian inference in such a complex dynamic model. This algorithm is a new nonparametric and sequential version of the belief propagation algorithm ever proposed in the literature [6–9]. Our method is more robust to motion uncertainties and abrupt motion because of the collaborative searching through multiscales, which have been demonstrated by the experiments in various scenarios.

2. Related Work

There are two approaches for visual tracking: the top-down approach and the bottom-up approach. The top-down approach takes a two step strategy, i.e., target state hypothesis generation and image observation verification as in the sequential Monte Carlo trackers [2, 10]. The bottom-up approach estimates the motion parameters by minimizing deterministic cost functions. To list a few, the mean-shift blob tracker [11] and the efficient region tracker with the parametric model of geometry and illumination [12] are the representatives using such an approach.

Almost all the existing methods using multiscale searching and matching take a hierarchical search strategy. The pyramid representation of the image makes this strategy a really efficient searching scheme and it has been applied successfully in fast stereo matching and face alignment [4]. The idea behind the hierarchical strategy is that the searching and matching in large scale can be very fast and it will guide more efficient search in fine scale. However, the inaccuracy and failure of the search in large scale may put the search in fine scale into risk.

For Bayesian inference based on graph models, the sum-product algorithm [6] can obtain the exact inference result but it only viable for small directed acyclic graph (DAG) models. When there is no loop in the graph model, belief propagation (BP) [6, 7] can obtain the exact inference more efficiently through a local message passing process. When there are loops in the graph model, the loopy BP [13] can obtain good approximate results [7]. As an approximation, Monte Carlo techniques can be used for simulation of Bayesian inference [10]. In addition, statistical variational approach provides a principled way for approximate inference such as the mean field variational method [14] which seeks the best approximate result by minimizing the K-L divergence between the mean field approximation and the real posterior distributions.

The Non-parametric BP [8] and the PAMPAS algorithm [9]combine the BP algorithm with MCMC technique to implement the inference. Also, a mean field Monte Carlo algorithm (MFMC) has been proposed in [15] for tracking articulated body by integrating sequential Monte Carlo technique with the mean field variational method.

Different from the hierarchical search methods, this paper proposes a new formulation for multiscale visual tracking based on a dynamic Markov network. This formulation results in a collaborative way of searching through multiscales instead of using the hierarchical strategy. A sequential belief propagation algorithm and its Monte Carlo implementation, namely sequential belief propagation Monte Carlo, are proposed to efficiently perform the Bayesian inference in the proposed Markov network. This is a new nonparametric and sequential version of the belief propagation algorithm.

3. Representation

The target state at each scale is denoted by $\mathbf{x}_i$ where $i \in \{1, \ldots, L\}$ indicates the scale with 1 indicating the largest scale and L the smallest. Putting multiscale states together results in a redundant representation for the target, denoted by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$. The benefit is that the multiscale target models make possible the integration of multiscale image observations which may conquer the abrupt motion. The image observation associated with the target state $\mathbf{x}_i$ in the same scale is denoted by $\mathbf{z}_i$ and $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_L\}$.

The error accumulation in the hierarchical search method is mainly due to the unidirectional information propagation from large scales to small scales. Our approach allows bidirectional information propagation to alleviate this prob-

lem based on a Markov network, as shown in Figure 1 as an example of three scales.

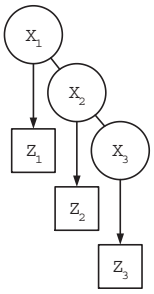


Figure 1: Markov network for the target state in multiple resolutions.

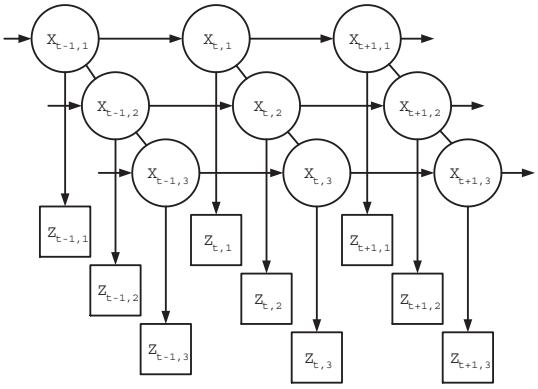


Figure 2: Dynamic Markov network for the motion of the target.

The undirected links describe the mutual influence of multiple scales, and the directed links represent the image observation processes. Each undirected link is associated with a potential function $\psi_{i,j}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_j))$, where $f_i(\mathbf{x}_i)$ and $f_j(\mathbf{x}_j)$ map $\mathbf{x}_i$ and $\mathbf{x}_j$ into the same state space since they may represent the target states in different state space. Each directed link is associated with an image likelihood function $p_i(\mathbf{z}_i|\mathbf{x}_i)$. According to the Bayesian rule, it is easy to show

$$P(\mathbf{X}|\mathbf{Z}) = \frac{1}{\mathbf{Z}_{\mathcal{Q}}} \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_j)) \prod_{i\in\mathcal{V}} p_i(\mathbf{z}_i|\mathbf{x}_i), \tag{1}$$

where $\mathbf{Z}_{\mathcal{Q}}$ is a normalization constant, $\mathcal{E}$ is the set of all the undirected links and $\mathcal{V}$ is the set of all the directed links.

The above Markov network is a generative model at one time instant. When putting it into the temporal context, we construct a dynamic Markov network as shown in Figure 2. The target state at scale i and time t are denoted by $\mathbf{X}_t = \{\mathbf{x}_{t,i}, i = 1, \ldots, L\}$. Also, we denote the image observations at time t under all scales by $\mathbf{Z}_t = \{\mathbf{z}_{t,i}, i = 1, \ldots, L\}$ and denote $\underline{\mathbf{Z}}_t = \{\mathbf{Z}_k, k = 1, \ldots, t\}$, then the tracking problem is to perform the Bayesian inference of the dynamic Markov network to obtain the marginal posterior probability $\mathbf{P}(\mathbf{x}_{t,L}|\underline{\mathbf{Z}}_t)$ where $\mathbf{x}_{t,L}$ is the target state at the smallest scale.

According to the Bayesian rule and the Markovian property, we have

$$P(\mathbf{X}_t|\underline{\mathbf{Z}}_t) \propto P(\mathbf{Z}_t|\mathbf{X}_t) \int_{\mathbf{X}_{t-1}} P(\mathbf{X}_t|\mathbf{X}_{t-1}) P(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1}), \tag{2}$$

and $\mathbf{P}(\mathbf{x}_{t,L}|\underline{\mathbf{Z}}_t)$ can be obtained by marginalizing $P(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$. But there are two reasons that we must seek for other solutions: firstly, the closed form solution to the joint posterior probability $P(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$ is very difficult to obtain especially when the probability distributions are non-Gaussian; secondly, even if we can obtain a closed form solution to $P(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$, marginalization is not an efficient method because of the computation of multiple integral. Therefore, we seek a more efficient way to calculate the marginal posterior $\mathbf{P}(\mathbf{x}_{t,i}|\underline{\mathbf{Z}}_t)$. In this paper, we develop a sequential BP algorithm with Monte Carlo implementation in the following sections.

4. Sequential Belief Propagation

To perform the Bayesian inference of $\mathbf{P}(\mathbf{x}_{t,L}|\underline{\mathbf{Z}}_t)$ in the dynamic Markov network in Figure 2, let's first solve the Bayesian inference problem of $\mathbf{P}(\mathbf{x}_L|\mathbf{Z})$ in the Markov network in Figure 1. Actually, the belief propagation algorithm calculate the exact inference of $\mathbf{P}(\mathbf{x}_i|\mathbf{Z}), i = 1, \ldots, L$ through a local message passing process [6, 7]. The local message passing from node i to node j is

$$\mathbf{m}_{ji}(\mathbf{x}_j) \leftarrow \int_{\mathbf{x}_i} [p_i(\mathbf{z}_i|\mathbf{x}_i)\psi_{i,j}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_j)) \times \prod_{k\in\mathcal{N}(\mathbf{x}_i)\backslash j} \mathbf{m}_{ik}(\mathbf{x}_i)] d\mathbf{x}_i, \tag{3}$$

where $\mathcal{N}(\mathbf{x}_i)$ denotes the neighborhood of $\mathbf{x}_i$ that consists of the set of nodes connected to $\mathbf{x}_i$ through a undirected link and $\mathcal{N}(\mathbf{x}_i)\backslash j$ means the neighbor of $\mathbf{x}_i$ except $\mathbf{x}_j$. Equation 3 is actually a set of fixed point equation. Iterating message passing until convergence, then, the marginal posterior probability of $\mathbf{x}_i$ can be obtained by

$$P(\mathbf{x}_i|\mathbf{Z}) \propto p_i(\mathbf{z}_i|\mathbf{x}_i) \prod_{j\in\mathcal{N}(\mathbf{x}_i)} \mathbf{m}_{ij}(\mathbf{x}_i). \tag{4}$$

To infer $\mathbf{P}(\mathbf{x}_{t,i}|\underline{\mathbf{Z}}_t)$, We extend the BP algorithm to the dynamic Markov model shown in Figure 2. We assume independent dynamic models in each resolution, i.e.,

$$P(\mathbf{X}_t|\mathbf{X}_{t-1}) = \prod_i p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}). \tag{5}$$

Then, given the inference results $P(\mathbf{x}_{t-1,i}|\underline{\mathbf{Z}}_{t-1}), i = 1,\ldots,L$ at previous time $t-1$, we show that the message updating at time t is

$$\mathbf{m}_{ji}(\mathbf{x}_{t,j}) \leftarrow \int_{\mathbf{x}_{t,i}} [p_i(\mathbf{z}_{t,i}|\mathbf{x}_{t,i})\psi_{i,j}(f_i(\mathbf{x}_{t,i}), f_j(\mathbf{x}_{t,j})) \times \int_{\mathbf{x}_{t-1,i}} p(\mathbf{x}_{t,i}|\mathbf{x}_{t-1,i})P(\mathbf{x}_{t-1,i}|\underline{\mathbf{Z}}_{t-1})d\mathbf{x}_{t-1,i} \times \prod_{k\in\mathcal{N}(\mathbf{x}_{t,i})\backslash j} \mathbf{m}_{ik}(\mathbf{x}_{t,i})]d\mathbf{x}_{t,i}, \quad (6)$$

and the marginal posterior probability at time t is given by

$$P(\mathbf{x}_{t,i}|\mathbf{Z}_t) \propto p_i(\mathbf{z}_{t,i}|\mathbf{x}_{t,i}) \prod_{j\in\mathcal{N}(\mathbf{x}_{t,i})} \mathbf{m}_{ij}(\mathbf{x}_{t,i}) \times \int_{\mathbf{x}_{t-1,i}} p(\mathbf{x}_{t,i}|\mathbf{x}_{t-1,i})P(\mathbf{x}_{t-1,i}|\underline{\mathbf{Z}}_{t-1})d\mathbf{x}_{t-1,i}. \quad (7)$$

From Equation 5 to Equation 7, we have developed a sequential belief propagation algorithm (SBP). To the best of our knowledge, this is a novel extension of the BP for visual tracking.

Actually, one special example which is able to show the necessity of the proposed method is that the target is very thin or even can not be seen in the image of the largest scale. In this case, for the proposed SBP algorithm, it only means that the 'belief' propagated from the largest scale to the smaller scales is uniformly distributed and thus non-informative. It will not affect the posterior motion $P(\mathbf{x}_{t,i}|\mathbf{Z}_t)$, $i = 1,\ldots,L$ at all. Thus the proposed SBP algorithm may still get good tracking result as long as the smaller scales can still provide confident 'beliefs' to help the searching and matching in the largest scale. While for the coarse-to-fine searching strategy, since the searching and matching will definitely fail in the largest scale and there are no mechanisms for the searching and matching in the smaller scales to recover from the failure , the whole hierarchical searching process will also fail.

Due to the non-Gaussian nature of the image observation density induced for example by cluttered backgrounds, the parametric form solution to this algorithm is intractable. In the next section, we will develop a Monte Carlo algorithm to implement the SBP algorithm.

5. SBP Monte Carlo

In this section, we firstly develop a BP Monte Carlo (BPMC) algorithm , then extend it to a SBP Monte Carlo (SBPMC) algorithm.

Each message in BP is represented by a set of weighted samples, i.e.,

$$\mathbf{m}_{ji}(\mathbf{x}_j) \sim \{\mathbf{s}_j^{(n)}, \omega_j^{(i,n)}\}_{n=1}^N, i \in \mathcal{N}(j). \quad (8)$$

where $\mathbf{s}_j^{(n)}$ and $\omega_j^{(i,n)}$ denote the sample and its weight of the message passing from $\mathbf{x_i}$ to $\mathbf{x_j}$, respectively. The marginal posterior probability in each node is also represented by a set of Weighted samples, i.e.,

$$P(\mathbf{x}_j|\mathbf{Z}) \sim \{\mathbf{s}_j^{(n)}, \pi_j^{(n)}\}_{n=1}^N. \quad (9)$$

where $\mathbf{s}_j^{(n)}$ is the same in equation 8, and $\pi_j^{(i,n)}$ is the belief corresponding to it. Then the message updating process is based on these sets of weighted samples. The algorithm is described in Figure 3.

Generate $\{\mathbf{s}_{j,k+1}^{(n)}, \omega_{j,k+1}^{(i,n)}\}_{n=1}^N$ and $\{\mathbf{s}_{j,k+1}^{(n)}, \pi_{j,k+1}^{(n)}\}_{n=1}^N$ from $\{\mathbf{s}_{j,k}^{(n)}, \omega_{j,k}^{(i,n)}\}_{n=1}^N$, $\{\mathbf{s}_{j,k}^{(n)}, \pi_{j,k}^{(n)}\}_{n=1}^N$, $j = 1,\ldots,L$ and $i \in \mathcal{N}(j)$

1. IMPORTANCE SAMPLING: Sample $\{\mathbf{s}_{j,k+1}^{(n)}\}_{n=1}^N$ from a suitable importance function $I_j(\mathbf{x}_j)$.

2. RE-WEIGHT: For each sample $\mathbf{s}_{j,k+1}^{(n)}$ and each $i \in \mathcal{N}(j)$, set the weight $\omega_{j,k+1}^{(i,n)} = G_j^{(i)}(\mathbf{s}_{j,k+1}^{(n)})/I_j(\mathbf{s}_{j,k+1}^{(n)})$ where

$$G_j^{(i)}(\mathbf{s}_{j,k+1}^{(n)}) = \sum_{m=1}^N [\pi_{i,k}^{(m)} \times p_i(\mathbf{z}_{i,k}^{(m)}|\mathbf{s}_{i,k}^{(m)}) \times \psi_{i,j}(f_i(\mathbf{s}_{i,k}^{(m)}), f_j(\mathbf{s}_{j,k+1}^{(n)})) \prod_{l\in\mathcal{N}(i)\backslash j} \omega_{i,k}^{(l,m)}]$$

3. NORMALIZATION: Normalize $\omega_{j,k+1}^{(i,n)}, i \in \mathcal{N}(j)$ and set and normalize

$$\pi_j^{(n)} = p_j(\mathbf{z}_{j,k+1}^{(n)}|\mathbf{s}_{j,k+1}^{(n)}) \prod_{l\in\mathcal{N}(j)} \omega_{j,k+1}^{(l,n)}$$

We get $\{\mathbf{s}_{j,k+1}^{(n)}, \omega_{j,k+1}^{(i,n)}\}_{n=1}^N$, $\{\mathbf{s}_{j,k+1}^{(n)}, \pi_{j,k+1}^{(n)}\}_{n=1}^N$.

4. ITERATION: $k \leftarrow k+1$, iterate $1 \rightarrow 4$ until convergence.

5. INFERENCE: $p(\mathbf{x}_j|\mathbf{Z}) \sim \{\mathbf{s}_j^{(n)}, \pi_j^{(n)}\}_{n=1}^N$ where $\mathbf{s}_j^{(n)} = \mathbf{s}_{j,k}^{(n)}$ and $\omega_j^{(i,n)} = \omega_{j,k+1}^{(i,n)}$.

Figure 3: Belief Propagation Monte Carlo

Our BPMC algorithm is different from both the NBP algorithm [8] and the PAMPAS algorithm [9]. They all model the messages in BP as Gaussian mixtures and complex MCMC samplers are used to sample the new Gaussian mixture kernels of the updated messages. In this sense their algorithms are semi-parametric. While our algorithm represents all the densities in pure nonparametric form and importance sampling technique is used to generate the new samples in each iteration of message passing. Therefore, our algorithm avoids complex MCMC samplers. In fact, The BPMC algorithm is similar to the MFMC algorithm in the sense of using importance sampling. Since the proposed

Markov network has no loop and BP can get the exact inference result, this may be better than MFMC since mean field variational method can only get an approximate result.

Following almost the same strategy, we can represent the messages and marginal posterior probabilities at each time instant as weighted samples, i.e.,

$$\mathbf{m}_{t,ji}(\mathbf{x}_{t,j}) \sim \{\mathbf{s}_{t,j}^{(n)}, \omega_{t,j}^{(i,n)}\}_{n=1}^{N}, i \in \mathcal{N}(j). \quad (10)$$

and

$$P(\mathbf{x}_{t,j}|\underline{\mathbf{Z}}_t) \sim \{\mathbf{s}_{t,j}^{(n)}, \pi_{t,j}^{(n)}\}_{n=1}^{N}. \quad (11)$$

Then following Equation 5 to Equation 7, the Monte Carlo implementation of the sequential belief propagation algorithm, namely SBPMC, is shown in Figure 4. Compared with NBP [8] and PAMPAS [9], the uniqueness of the SBPMC algorithm is obvious, none of the former two is sequential.

6. Experiments

The proposed algorithm has been applied to tracking targets with abrupt motion in various scenarios. The target of interest is modelled as a rectangular region, and the target state $\mathbf{x_i}$ at each resolution is a four dimensional vector with two for displacements and two for scalings. The motion model $p_i(\mathbf{x}_{t,i}|\mathbf{x}_{t-1,i})$ in each scale is standard second order constant acceleration model with Gaussian noise. Since the purpose of our algorithm is to deal with motion uncertainty, we do not learn the parameters of the motion models but preset them to cover enough uncertainties. The tracker uses different observation likelihood models $p_i(\mathbf{z}_i|\mathbf{x}_i)$ at different scales, where different PCA-based appearance models are trained and adopted.

6.1. Sudden Dynamic Changes

The first scenario is a tennis bounced back from a desk. The dynamics of the tennis is suddenly changed when it hits the desk. This situation is hard for tracking algorithms that rely on a single motion model. Our experiment show that the proposed multiscale tracking algorithm can successfully cope with this problem. In the experiment, 50 samples are used for each scale and the number of iterations for the sequential belief propagation algorithm is set to 5. Sample frame of our algorithm at the bouncing stage are shown in Figure 5, where the results in each scale have been displayed as well. The details of the results can be seen in the video "`BouncingTennis.avi`" as part of the submission.

We also implemented CONDENSATION for comparison. However, it can hardly handle the large motion presented in this sequence even with 1000 particles. Figure 6 shows its results (with 1000 particles) on the same period as in Figure 5. Details of the failure can be seen in the video `Condensation.avi`. It is clear that CONDENSATION loses

Generate $\{\mathbf{s}_{t,j}^{(n)}, \pi_{t,j}^{(n)}\}_{n=1}^{N}$ from $\{\mathbf{s}_{t-1,j}^{(n)}, \pi_{t-1,j}^{(n)}\}_{n=1}^{N}$.

1. INITIALIZATION: Sequential Monte Carlo, $k \leftarrow 1$

1.1. Re-sampling: For each $j = 1, \ldots, L$, re-sampling $\{\mathbf{s}_{t-1,j}^{(n)}\}_{n=1}^{N}$ according to the weights $\pi_{t-1,j}^{(n)}$ to get $\{\mathbf{s}_{t-1,j}^{(n)}, \frac{1}{N}\}_{n=1}^{N}$

1.2. Prediction: For each $j = 1, \ldots, L$, for each sample in $\{\mathbf{s}_{t,j}^{(n)}, \frac{1}{N}\}_{n=1}^{N}$, sampling from $p(\mathbf{x_{t,j}}|\mathbf{x_{t-1,j}})$ to get $\{s_{t,j,k}^{(n)}\}_{n=1}^{N}$

1.3.Belief and Message Initialization: For each $j = 1..L$, assign weight $\omega_{t,j,k}^{(i,n)} = \frac{1}{N}, \pi_{t,j,k}^{(n)} = p_j(\mathbf{z}_{t,j,k}^{(n)}|\mathbf{s}_{t,j,k}^{(n)})$ and normalize them where $i \in \mathcal{N}(j)$.

2. ITERATION: Belief Propagation Monte Carlo, $k \leftarrow k + 1$

2.1. Importance Sampling: Sample $\{\mathbf{s}_{t,j,k+1}^{(n)}\}_{n=1}^{N}$ from $p(\mathbf{x_{t,j}}|\mathbf{x_{t-1,j}})$.

2.2. Re-weight: For each sample $\mathbf{s}_{t,j,k+1}^{(n)}$ and each $i \in \mathcal{N}(j)$, set the weight

$$\omega_{t,j,k+1}^{(i,n)} = G_{x_{t,j}}^{(i)}(\mathbf{s}_{t,j,k+1}^{(n)})/(\frac{1}{N}\sum_{r=1}^{N} p(\mathbf{s}_{t,j,k+1}^{(n)}|\mathbf{s}_{t-1,j}^{(r)}))$$

where

$$G_{x_{t,j}}^{(i)}(\mathbf{s}_{t,j,k+1}^{(n)}) = \sum_{m=1}^{N}\{\pi_{t,i,k}^{(m)} p_i(\mathbf{z}_{t,i,k}^{(m)}|\mathbf{s}_{t,i,k}^{(m)}) \prod_{l\in\mathcal{N}(i)\setminus j} \omega_{t,i,k}^{(l,m)}$$
$$\times[\frac{1}{N}\sum_{r=1}^{N} p(\mathbf{s}_{t,i,k}^{(m)}|\mathbf{s}_{t-1,i}^{(r)})] \cdot \psi_{i,j}(f_i(\mathbf{s}_{t,i,k}^{(m)}), f_j(\mathbf{s}_{t,j,k+1}^{(n)}))\}$$

2.3. Normalization: Normalize $\omega_{t,j,k+1}^{(i,n)}, i \in \mathcal{N}(j)$ and set

$$\pi_{t,j,k+1}^{(n)} = p_j(\mathbf{z}_{t,j,k+1}^{(n)}|\mathbf{s}_{t,j,k+1}^{(n)}) \prod_{l\in\mathcal{N}(j)} \omega_{t,j}^{(l,n)}$$
$$\times \sum_{r} p(\mathbf{s}_{t,j,k+1}^{(n)}|\mathbf{s}_{t-1,j}^{(r)})$$

and normalize it. Then, $\{\mathbf{s}_{t,j,k+1}^{(n)}, \omega_{t,j,k+1}^{(i,n)}\}_{n=1}^{N}$ and $\{\mathbf{s}_{t,j,k+1}^{(n)}, \pi_{t,j,k+1}^{(n)}\}_{n=1}^{N}$ are obtained .

2.4. Iteration: $k \leftarrow k + 1$, iterate $2.1 \rightarrow 2.4$ until convergence.

3. INFERENCE RESULT:
$p(\mathbf{x}_{t,j}|\mathbf{Z}_t) \sim \{\mathbf{s}_{t,j}^{(n)}, \pi_{t,j}^{(n)}\}_{n=1}^{N}$ where $\mathbf{s}_{t,j}^{(n)} = \mathbf{s}_{t,j,k}^{(n)}$ and $\pi_{t,j}^{(n)} = \pi_{t,j,k+1}^{(n)}$.

Figure 4: Sequential Belief Propagation Monte Carlo

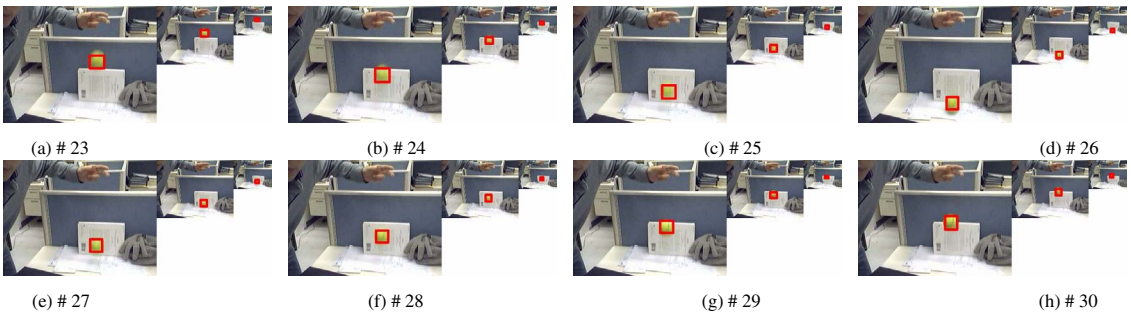


Figure 5: Tracking Bouncing Tennis by SBPMC. Frame numbers are indicated in the bottom of the result image.

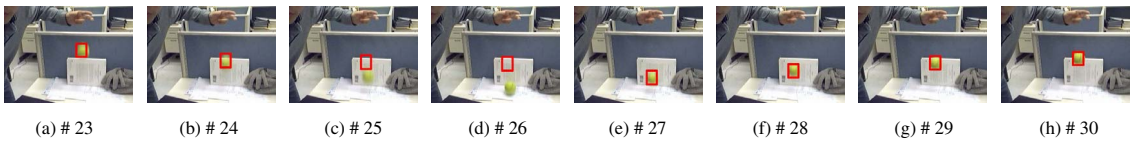


Figure 6: Tracking Bouncing Tennis by CONDENSATION. Frame numbers are indicated in the bottom of the result image

track when the motion uncertainty of the ball is large. Although CONDENSATION works with 2000 samples, it is about two times slower than our algorithm.

A similar experiment of tracking a bouncing ball has been reported by using a mix-state CONDENSATION that switches between two specific motion models was [2]. This approach needs to know the set of specific motion models in advance, which makes this approach less scalable. In contrast to this approach, our method is more general and scalable.

6.2. Dropping Frames

Frame dropping in video sequence also causes abrupt motions of the target. The second experiment is to track a human face in a video with more than 1200 original frames. We deliberately simulate the frame dropping scenario by keeping one frame in every 10 frames. With the same setting as in Section 6.1, our algorithm can successfully handle such a jumpy sequence. Sample frames are displayed in Figure 7 and details can be seen in the video "`DroppingFrame.avi`".

To demonstrate the effectiveness of the sequential belief propagation process, we collect the intermediate iteration results of SBPMC in every time frame. The initialization are rather far from the true locations. We have observed many cases where the initial estimates at the largest scale are not satisfactory enough for hierarchical search. Our experiments show that these cases do not pose any difficulty to our algorithm. Figure 8 shows the belief propagation iteration in frame 20 which correspond to frame 200 in the original sequence. Under a bad initialization, the estimate at the largest scale is not good. But our algorithm converges in 5 iterations and the bad estimate at the largest scale is corrected due to the belief propagated from other scales. The reason is the collaboration of multiple scales in our algorithm with the use of different observation models for different scales.

6.3. Shaking Cameras and Changing Scales

Shaking cameras also induce abrupt motion of the target in the video sequence. We have tracked a human head in a video sequence of 477 frames with very large camera motion. Sample frames are presented in Figure 9. The proposed SBPMC tracker achieves good and robust results even under very large camera shaking and the lighting changes. Details can be seen in the video "`ShakingCamera.avi`". One may argue that image stabilization may handle camera motion, but it may not be able to handle the abrupt motion induced by other sources such as the target movement itself, and it is a hard problem itself to recover camera ego-motion in general settings.

The last scenario in our experiments is to track a human with large scale changes. The people walks back and forth to the camera with frequent changes of moving direction and speed. This example contains large scale changes and our algorithm achieves very good results. Sample frames are shown in Figure 10 and details can be see in the video "`WalkingPeople.avi`".

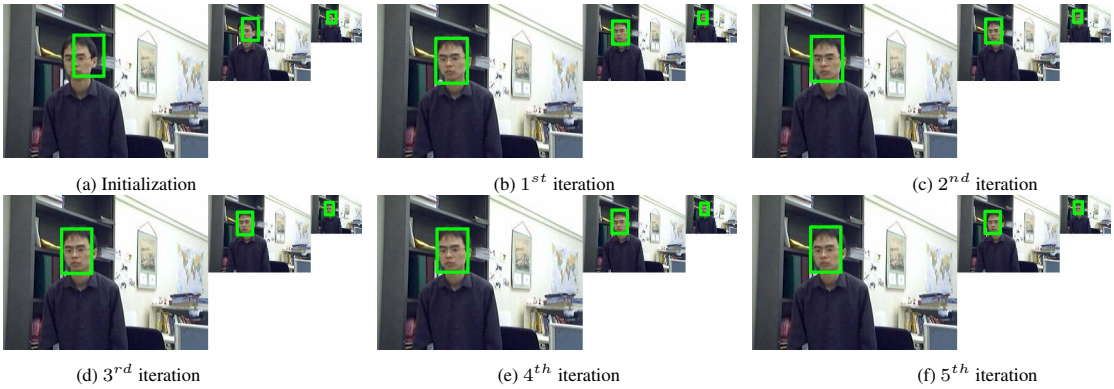
7. Discussion and Conclusion

Existing tracking algorithms with small motion assumption are vulnerable to abrupt motion. In this paper, we propose a novel statistical method to overcome the abrupt motion for robust visual tracking. It is based on a dynamic Markov network representation that models the multiscale tracking

(a)# 34 (b)# 35 (c)# 36

(d) # 37 (e) # 38 (f) # 39

Figure 7: Tracking Head by SBPMC. Frame numbers are indicated in the bottom of the result images



(a) Initialization (b) 1^{st} iteration (c) 2^{nd} iteration

(d) 3^{rd} iteration (e) 4^{th} iteration (f) 5^{th} iteration

Figure 8: SBPMC Iteration in frame 200. The iteration number is indicated at the bottom of each result image.

process. In this framework, a collaborative tracking strategy among different scales is proposed. The SBPMC algorithm for the dynamic Markov network is developed by implementing the collaboration of a set of particle filters. This is a new nonparametric and sequential belief propagation algorithm. Extensive experiments have shown that he real benefit of our new approach is the ability to handle a large spectrum of abrupt motions including sudden dynamics changes, large camera motion, large scaling and dropping frames, etc.

The dynamic Markov network is actually a generative model approach, and has demonstrated its effectiveness in other applications such as tracking articulated human body [15] where a mean field algorithm has been developed. Our future work includes a theoretical comparison of the proposed sequential belief propagation algorithm with this mean field algorithm, and the learning algorithms that estimate the parameters of the dynamic Markov networks.

Acknowledgment

This work was supported in part by National Science Foundation grants IIS-0347877, IIS-0308222, Northwestern faculty startup funds for Ying Wu and Walter P. Murphy Fellowship for Gang Hua.

References

[1] B. North and A. Blake, "Learning dynamical models by expectation maximisation," in *Proc. 6th International Conference on Computer Vision*, 1998, pp. 384–389.

[2] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *Proc. 6th International Conference on Computer Vision*, 1998, pp. 107–112.

[3] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, South Carolina, 2000.

[4] C. Liu, H.-Y. Shum, and C. Zhang, "Hierarchical shape modeling for automatic face localization," in *Proc. European Conference on Computer Vision*, May 2002, pp. 687–703.

[5] C.-E. Guo, S.-C. Zhu, and Y. N. Wu, "Towards a mathematical theory of primal sketch and sketchability," in *Proc. International Conference on Computer Vision*, Nice,Côte d'Azur,France, Octobor 2003, pp. 1228–1235.

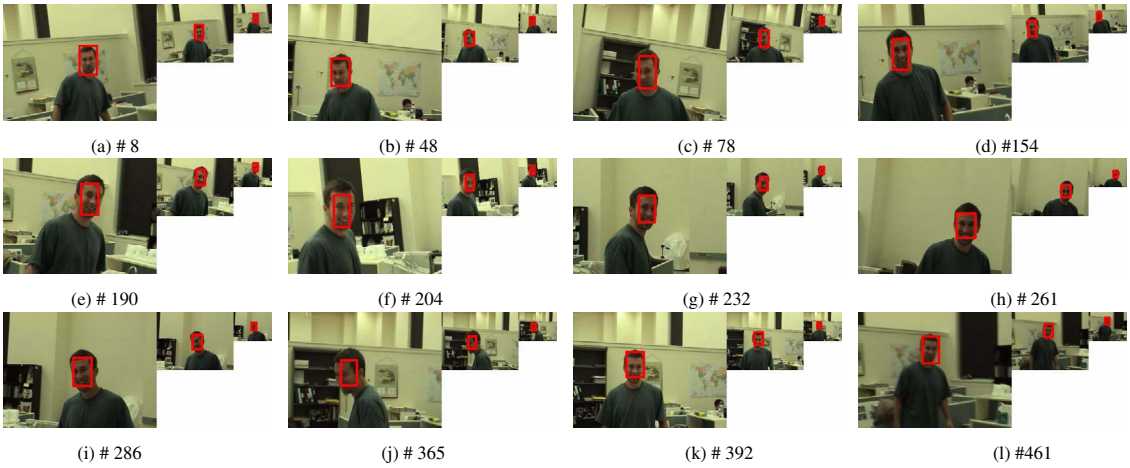


Figure 9: Tracking head under large camera motion by SBPMC. Frame numbers are indicated in the bottom of the result images

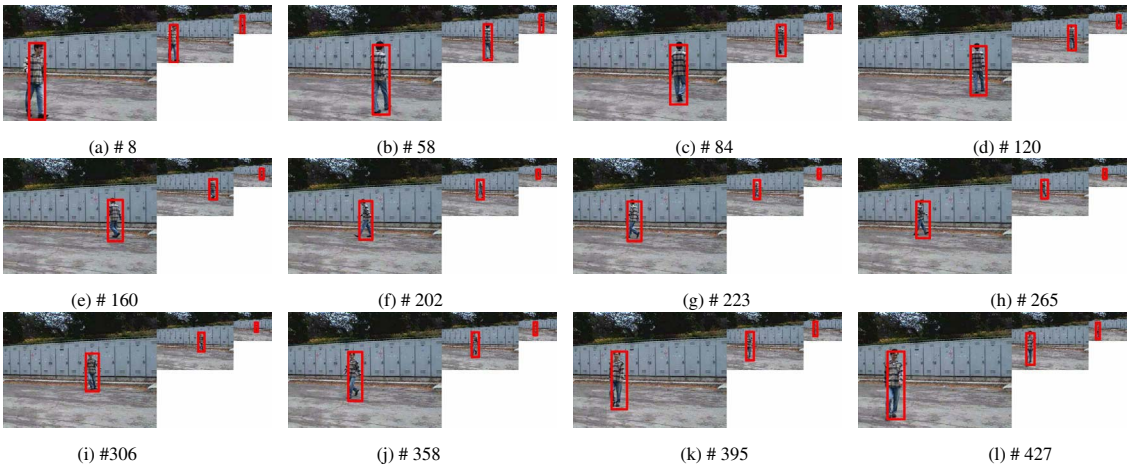


Figure 10: Tracking human with large scale change by SBPMC. Frame numbers are indicated in the bottom of the result images

[6] M. I. Jordan and Y. Weiss, "Graphical models: Probabilistic inference," in *The Handbook of Brain Theory and Neural Network*, 2nd ed. Cambridge, MA: MIT Press, 2002, pp. 243–266.

[7] W. T. Freeman and E. C. Pasztor, "Learning low-level vision," in *Proc. International Conference on Computer Vision*, Kerkyra, Greece, September 1999.

[8] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003, pp. 605–612.

[9] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, June 2003, pp. 613–620.

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

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, South Carolina, 2000, pp. 142–149.

[12] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[13] K. Murphy, Y. Weiss, and M. Jordan, "Loopy-belief propagation for approximate inference: An empirical study," in *Proc. Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, July 1999.

[14] T. S. Jaakkola, "Tutorial on variational approximation method," *Advanced mean field methods: theory and practice. MIT Press*, 2000.

[15] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic markov network," in *Proc. IEEE International Conference on Computer Vision*, Nice,Côte d'Azur,France, October 2003, pp. 1094–1101.