Published in IEEE Transactions on Systems, Man and Cybernetics- Part B, Vol. 39, No. 2, pp. 375-388, April 2009

Tracking by Parts: A Bayesian Approach with Component Collaboration

Wen-Yan Chang, Chu-Song Chen, and Yi-Ping Hung

***Abstract*—Instead of using global appearance information for visual tracking, as adopted by many methods, we propose a tracking-by-parts (TBP) approach that uses partial appearance information for the task. The proposed method considers the collaborations between parts, and derives a probability propagation framework by encoding the spatial coherence in a Bayesian formulation. To resolve this formulation, a TBP particle filtering method is introduced. Unlike existing methods that only use the spatial-coherence relationship for particle weight estimation, our method further applies this relationship for state prediction based on the system dynamics. Thus, the part-based information can be utilized efficiently and the tracking performance can be improved. Experimental results show that our approach outperforms the factored-likelihood and particle re-weight methods, which only use spatial coherence for weight estimation.**

***Index Terms*—Component collaboration, contrast histogram, particle filtering, tracking by parts, visual tracking.**

I. Introduction

OBJECT tracking, which is a fundamental problem in computer vision, serves as a basic module for many applications, such as video content understanding, robot navigation, and vision-based user interfaces. Methods of object tracking can be divided into two classes: the holistic approaches and the part-based approaches. The former represent a target object by global information. For example, Comaniciu et al. [1] used a color histogram to model a target, and adopted the Bhattacharyya distance for measurement. However, it is difficult to handle varying illumination conditions with color histograms. To resolve this difficulty, a number of adaptive approaches that employ classification methods to learn the target appearance model have been proposed [2], [3]. In addition, to deal with the nonlinear and/or non-Gaussian properties of real-world problems, particle filtering is a successful method that has been widely used [4], [5]. Based on particle filtering, Ross et al. [3] employed the probabilistic principal component analysis (PCA) to represent target likelihoods by eigenbases. Their method utilizes the Riemannian singular value decomposition (R-SVD) algorithm to incrementally update the bases for lighting changes. As well as probabilistic PCA, other learning/classification methods (e.g., support vector machine (SVM) in [2]) have been applied in tracking tasks to help define the target likelihood function.

Even though a learning process can be used to incrementally model an object's appearance, handling partial occlusions is still a challenging problem. To address this issue, the focus of several studies has shifted to part-based representation. Recent advances in computer vision have shown that part-based approaches are effective for object detection and recognition [6], [7]. In such approaches, an object is represented as a collection of parts or components. By identifying the local parts and considering their inter-relationships, better detection or identification results can be achieved. The part-based strategy is also used for single-object tracking. Using factored likelihood estimation for likelihood measurement, Perez et al. [8] suggested a method that tracks a target using single-chain particle filtering. Under this model, an observation is divided into two independent parts that differ in appearance. Sigal et al. [9] developed a tracking method that locates a target's components by using non-parametric belief propagation (NBP) for iterative measurement based on the spatial constraints of the target. Recently, Hua and Wu [10] addressed the issue of inconsistency in observation measurement for part-based tracking based on factored likelihood measurement. The approach evaluates inconsistency by using the similarity among neighboring components, and ignores inconsistent parts in the likelihood measurement. In [11], Yu and Wu used Gaussian mixtures to model the appearance of a target, and proposed an EM-based algorithm to estimate the transformation by maximizing the likelihood among appearances.

The above methods focused on the observation measurement to improve the tracking performance. Although a target is divided into several components, they assumed that these components are with an identical motion, and only used the components for observation measurement. In addition to dealing with the identical-motion case, multiple motions among components have also been considered in recent studies of multi-object tracking [12]–[14] or articulation tracking [15], [16]. For lip tracking, Patras et al. [12] evaluated the measurement of lips based on the likelihoods of the individual components, and further considered spatial coherence by re-weighting the likelihood measurement based on the auxiliary particle filtering. A similar approach that uses joint likelihood filters was also proposed by Rasmussen and Hager [13]. In [14], Qu et al. exploited the interactive collaboration among objects to resolve the error merge and mis-labeling problems in multi-object tracking. The method employs a magnetic-inertial model to estimate the interactive likelihood. To track articulated hand movements, Sudderth et al. [15] used multiple independent trackers for each hand articulation, and applied NBP to adjust particle locations iteratively in order to obtain better estimations. Instead of NBP, Wu et al. [16] suggested using variational analysis to

W.-Y Chang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, and also with the Institute of Information Science, Academia Sinica. (e-mail: wychang@iis.sinica.edu.tw)

C.-S. Chen (the corresponding author) is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, and also with the Graduate Institute of Networking and Multimedia, National Taiwan University. (e-mail: song@iis.sinica.edu.tw)

Y.-P. Hung is with the Department of Computer Science and Information Engineering, National Taiwan University, and also with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan. (e-mail: hung@csie.ntu.edu.tw)

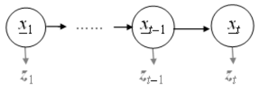


Figure 1. Single-chain BN considered in particle filtering.

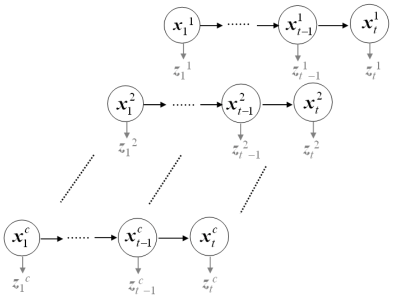


Figure 2. A BN consisting of multiple independent single-chain BNs.

cope with a loopy Markov network in order to maintain spatial coherence during tracking. Although the methods in [15] and [16] can handle spatial coherence well, they tend to be time-consuming because an iterative process is required in each time step.

Part-based approaches have proved effective in enhancing the tracking performance, but many of them only utilized the spatial-coherence information for improving the observation-likelihood or particle-weight. When the location of a component is mis-tracked (as in the case of occlusion), it is difficult for this type of approach to recover the state. Even though the belief propagation-based scheme can be exploited to adjust particle locations, maintaining spatial coherence based on the wrong location (or the outliers of locations) in the same time step usually requires a considerable number of iterative refinements.

In this paper, we propose a Bayesian probability propagation framework to maintain the spatial coherence. Unlike previous approaches in which the spatial-coherence relationship does not affect the dynamic model, our approach allows the components' locations to be *predicted* based on spatial-coherence information during tracking. State recovery is thus more efficient. In our approach, spatial coherence is not only considered in particle weight estimation, but also for temporal-based propagation. This is one of the major differences between our approach and those that only employ spatial coherence for likelihood refinement and particle adjustment in each individual time step [8]–[17]. By further encoding the spatial relationship in a dynamic model, we provide a novel framework that uses part-based spatial information for visual tracking.

The remainder of this paper is organized as follows. Section II introduces TBP particle filtering, and Section III describes its dynamic distribution. The component representation method and likelihood models are presented in Section IV. The experimental results are detailed in Section V. Then, in Section VI, we summarize our conclusions and indicate the direction of our future work.

II. TBP Particle Filtering

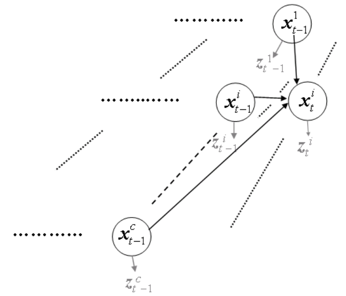


Figure 3. A BN containing more edges than the BN in Fig. 2. For a node $\boldsymbol{x}_t^i$, all the links from $\boldsymbol{x}_{t-1}^k$ to $\boldsymbol{x}_t^i$ are included for $k = 1\ldots c$, instead of including only the link from $\boldsymbol{x}_{t-1}^i$ to $\boldsymbol{x}_t^i$, as in Fig. 2.

Since stage-wise estimation is essential for tracking, a tracking task is usually modeled as a state estimation problem; particle filtering is a popular technique for solving this problem. Given a chain of observations, $\boldsymbol{Z}_t = \{z_1,\ldots, z_t\}$, particle filtering estimates state $\underline{x}_t$ of a single-chain Bayesian network (BN), as shown in Fig. 1.

A straightforward way for multi-component tracking is to represent each component as a single-chain BN, and use particle filtering to track each component. This yields a BN consisting of c independent chains, as shown in Fig. 2, where c is the number of components. In this framework, the motions of components are assumed to be independent. This assumption has been adopted by many multi-object tracking methods [18]–[20].

In the framework shown in Fig. 2, however, the state at time t is only influenced by the state of the same component at time t–1. The previous state of a component thus plays a decisive role in estimating the current state of the same component. This means that tracking is difficult to recover when drift occurs, even when properly tracked components give useful hints to the other miss-tracked components.

To overcome this limitation, we consider the new BN shown in Fig. 3. The main difference between Fig. 3 and Fig. 2 is that state $\boldsymbol{x}_t$ of a component is not only influenced by $\boldsymbol{x}_{t-1}$ of the same component, but all components. More specifically, for each node $\boldsymbol{x}_t^i$ that represents the state of the i-th component at time t, the c edges $(\boldsymbol{x}_{t-1}^k, \boldsymbol{x}_t^i)$, $k = 1 \ldots c$, are all included in the graphical representation of the BN in our framework, as shown in Fig. 3, instead of including only the edge $(\boldsymbol{x}_{t-1}^i, \boldsymbol{x}_t^i)$ shown in Fig. 2.

By introducing these extra edges, the BN can no longer be treated as a set of independent chain-based BNs. The advantage is that cross references are allowed and the spatial coherence among components can thus be considered and propagated. Note that when the extra edges are disabled by setting the state-transition probability to

$$p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^1, \boldsymbol{x}_{t-1}^2,\ldots, \boldsymbol{x}_{t-1}^c) = p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^i), \quad (1)$$

the BN in Fig. 3 degenerates to that in Fig. 2. Therefore, our framework is an extension of the existing particle filtering method applied to state estimation of a dynamic system, where the components can be either dependent or independent.

We call the BN in Fig. 3 a *tracking-by-parts Bayesian network* (TBP-BN). The state-estimation problem involved in TBP-BN is not suitable to be solved directly by the particle filtering developed for single-chain BN. We thus address the following questions:

(1) How can probabilities be propagated so that the posterior distribution can be found for TBP-BN?

(2) How can the inference of TBP-BN be performed efficiently?

We consider these two problems in the following.

A. Bayesian Probability Propagation

The posterior probability of TBP-BN can be resolved by

$$\begin{aligned} p(\boldsymbol{x}_t|\,\boldsymbol{Z}_t) &= p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{Z}_t) \\ &\propto p(z_t|\,\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c)\cdot p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{Z}_{t-1}), \end{aligned} \tag{2}$$

where $\boldsymbol{x}_t = (\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c)$. In $p(z_t|\,\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c)$, state $\boldsymbol{x}_t^i$ of the i-th component only links to its local observation z_t^i, where $z_t = (z_t^1,\ldots,z_t^c)$; and the local observation is conditionally independent of the other states when $\boldsymbol{x}_t^i$ is given. The joint conditional likelihood is therefore

$$p(z_t|\,\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c) = \prod_{i=1}^{c} p(z_t^i|\,\boldsymbol{x}_t^i). \tag{3}$$

By substituting (3) into (2), we can rewrite (2) as

$$p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{Z}_t) \propto \prod_{i=1}^{c} p(z_t^i|\boldsymbol{x}_t^i)\cdot \int p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)\cdot p(\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c|\boldsymbol{Z}_{t-1})\,d\boldsymbol{x}_{t-1}. \tag{4}$$

From (4), the evaluation of the posterior probability at time t, $p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{Z}_t)$, is iteratively related to that at time $t-1$, $p(\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c|\,\boldsymbol{Z}_{t-1})$. Equation (4) thus shows how the posterior probability propagates from time $t-1$ to time t in TBP-BN.

To avoid the integral in (4), which is computationally intractable, particle filtering can be introduced to estimate the posterior distribution. A set of weighted particles $\{(\boldsymbol{s}_{t;n}, \pi_{t;n}), n = 1,\ldots, N\}$ is used to represent the posterior $p(\boldsymbol{x}_t|\,\boldsymbol{Z}_t)$, where $\boldsymbol{s}_{t;n}$, $n = 1,\ldots, N$, are drawn from the dynamic model $p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$; and $\pi_{t;n}$, $n = 1,\ldots, N$, are the associated weights evaluated according to the likelihood in (3).

However, since particles are sampled from the joint conditional probability $p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$, directly generating particles based on this probability involves sampling in a D-dimensional space, where $D = c\cdot d$, when the state $\boldsymbol{x}_t^i$ is represented by a d-dimension vector. In the following, we show that the structure of TBP-BN allows this joint conditional probability to be simplified, and a more efficient sampling procedure can be inferred in lower-dimensional spaces.

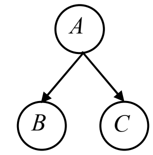


Figure 4. Diverging connection of a BN.

B. Inference of TBP-BN

From the diverging connection property of a BN, as illustrated in Fig. 4, nodes B and C are conditionally independent if A is observed. Relating this property to Fig. 3, let us treat A as a super-node consisting of $(\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$, and B and C as $\boldsymbol{x}_t^i$ and $\boldsymbol{x}_t^j$, respectively, where $i, j \in \{1,\ldots, c\}$ and $i \neq j$. Applying this property to TBP-BN, the states at time t are conditionally independent given all the states at time $t-1$. Hence, the dynamic model $p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$ can be decomposed into

$$p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c) \approx \prod_{j=1}^{c} p(\boldsymbol{x}_t^j \mid \boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c), \tag{5}$$

where $\prod_{j=1}^{c} p(\boldsymbol{x}_t^j \mid \boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$ is the *proposal function* for importance sampling based on the generic particle-filtering scheme [4].

Substituting (5) into (4), we have

$$p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{Z}_t) \propto \prod_{i=1}^{c} p(z_t^i|\,\boldsymbol{x}_t^i) \int \prod_{j=1}^{c} p(\boldsymbol{x}_t^j|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)\cdot p(\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c|\,\boldsymbol{Z}_{t-1})\,d\boldsymbol{x}_{t-1}. \tag{6}$$

Equations (5) and (6) demonstrate that we do not need to perform importance sampling in a D-dimensional space. Instead, due to the independence relationship shown in (5), we can perform the sampling of the dynamic model in a d-dimensional space c times. To draw particles in the D-dimensional space from the joint probability $p(\boldsymbol{x}_t^1,\ldots,\boldsymbol{x}_t^c|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$, we first draw d-dimensional particles from $p(\boldsymbol{x}_t^j|\,\boldsymbol{x}_{t-1}^1,\ldots,\boldsymbol{x}_{t-1}^c)$ for each $j\in\{1,\ldots,c\}$. Composing the c lower-dimensional particles into a single vector thus forms a particle of the dynamic model in the D-dimensional joint-probability space. An illustration of this concept with $c = 2$ is given in Fig. 5.

The advantages of generating particles in this way are two-fold. First, drawing particles from a lower-dimensional space is more efficient. We thus avoid the difficulty of generating particles in a high-dimensional space. Second, since the number of particles grows exponentially with the number of components c, if we draw M particles in each

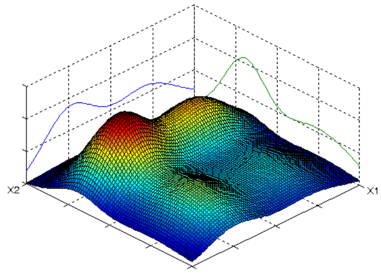


Figure 5. An illustration of TBP particle filtering. Particles in the x_1-x_2 space can be generated by combining the particles generated from the x_1 and x_2 spaces, respectively, when x_1 and x_2 are statistically independent.

lower-dimensional space, we can get M^c particles of the joint conditional probability in the original D-dimensional space. Note that this could be a huge number that would be impossible to sample directly in the original space when c is large. However, by using the independent-sampling strategy, we only need to perform $c{\cdot}M$ sampling operations in lower-dimensional spaces, instead of generating M^c particles in a higher-dimensional space.

Since the likelihood is also formed by the products of the individual likelihoods (as shown in (3)), we can separate them in a similar manner to that in the dynamic model. We thus introduce the following particle-filtering framework for TBP-BN, which is a preliminary version of our algorithm:

For each component $i \in \{1,\ldots,c\}$, we form a set of weighted particles

$$\boldsymbol{p}_i = \{(\boldsymbol{s}^i_{t;n}, \pi^i_{t;n}),\ n = 1,\ldots, N_i\},$$

where $\boldsymbol{s}^i_{t;n}$ is a particle and $\pi^i_{t;n}$ is its associated weight for component i. The relationship among the weighted particles of the joint posterior probability and the individual components is formulated as follows. The weighted particles of the joint posterior probability, $\{(\boldsymbol{s}_{t;n}, \pi_{t;n}),\ n = 1,\ldots, \boldsymbol{N}\}$, are represented by the combination of particles derived from each component:

$$\boldsymbol{s}_{t;n} = \{(\boldsymbol{s}^1_{t;i}, \boldsymbol{s}^2_{t;j}, \ldots, \boldsymbol{s}^c_{t;k})\} \quad (7)$$

$$\in \boldsymbol{p}_1 \times \boldsymbol{p}_2 \times \ldots \times \boldsymbol{p}_c,$$

where "×" denotes the Cartesian product, $i \in 1,\ldots, N_1$, $j \in 1,\ldots, N_2$, $k \in 1,\ldots, N_c$,

$$\boldsymbol{N} = \Pi_{i=1,\ldots,c} N_i;$$

and

$$\pi_{t;n} \propto \pi^1_{t;i} \cdot \pi^2_{t;j} \cdot \ldots \cdot \pi^c_{t;k}, \quad (8)$$

where $i \in 1,\ldots, N_1$, $j \in 1,\ldots, N_2$, $k \in 1,\ldots, N_c$. We call the above method of inferring TBP-BN by weighted particles *TBP particle filtering*. Table I lists the process of TBP particle filtering for each component i, where the re-sampling procedure [4] is adopted in our work to avoid the degeneracy problem as suggested in many particle-filtering-based methods. As the illustration in Fig. 5, though only $\Sigma_{i=1,\ldots,c} N_i$ times of sampling operations are performed in this scheme, $\boldsymbol{N} = \Pi_{i=1,\ldots,c} N_i$ particles are equivalently generated to represent the joint posterior probability.

Table I. Concept of TBP particle filtering for each component

Given a set of weighted particles $\{(\boldsymbol{s}^i_{t-1;n}, \pi^i_{t-1;n}),\ n = 1,\ldots, N_i\}$ for each component $i \in \{1,\ldots, c\}$ at time step t−1, the following steps are performed to construct a new set of particles at time step t.
1. Re-sample a particle set $\{(\boldsymbol{s}'^i_{t-1;n}, N_i^{-1}),\ n = 1,\ldots, N_i\}$ from $\{(\boldsymbol{s}^i_{t-1;n}, \pi^i_{t-1;n}),\ n = 1,\ldots, N_i\}$.
2. Generate a set of particles $\{\boldsymbol{s}^i_{t;n},\ n = 1,\ldots, N_i\}$ from the dynamic model $p(\boldsymbol{x}^i_t \mid \boldsymbol{x}^1_{t-1} = \boldsymbol{s}'^1_{t-1;n}, \ldots, \boldsymbol{x}^c_{t-1} = \boldsymbol{s}'^c_{t-1;n})$ for each component i, as being detailed in Section III.
3. Measure the weight $\pi^i_{t;n}$ of particle $\boldsymbol{s}^i_{t;n}$ by combining both likelihood and inter-component information, for $n = 1,\ldots, N_i$, as being detailed in Section IV.

C. Refinement of the Particle Weights by Spatial Relationships

In the above, we showed how to encode spatial-coherence information into the dynamic model of TBP-BN for part-based tracking. In addition to its use in the dynamic model, the spatial relationship can also be applied to refine the particle weights. The preliminary version of our algorithm introduced above only uses the likelihood for particle weight estimation, which can be further refined by inter-component relationships. Typically, standard belief propagation algorithms [21], or other variations [22]–[24], can be applied to refine the particles iteratively. They can be performed by forming a tree-like or a loopy graph among the components in advance, where each node of the graph represents a component and the edges model the compatibility constraints established by the spatial relationship between two nodes. Many previous works have applied this scenario to component-based tracking [9], [15]–[17], [25]. However, the computational load is very heavy for these iteration-based approaches.

Inspired by the work of [12], we derive a more efficient particle re-weighting strategy by assuming that there are some further links representing the inter-dependences among components at the same time step (the dash-dotted arcs in Fig. 6). In [12], a particle re-weighting method was derived by further considering these links based on the auxiliary particle-filtering scheme. In our approach, we derive the re-weighting strategy based on the generic particle-filtering scheme [4].

When the dependency among components of the same time has been incorporated as shown in Fig.6, the particle weight can be re-formulated as

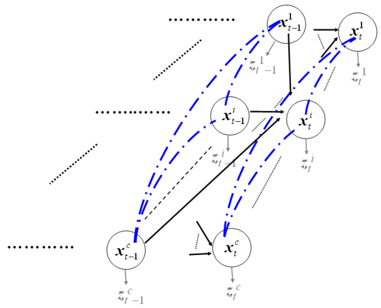


Figure 6. The complete BN used in our framework, where the inter-dependencies in the same time step are further considered (i.e., the dash-dotted arcs).

$$
\begin{aligned}
\pi_t &\propto \frac{p(\boldsymbol{X}_t \mid \boldsymbol{Z}_t)}{q(\boldsymbol{X}_t \mid \boldsymbol{Z}_t)} \\
&\propto \frac{p(\boldsymbol{X}_t, \boldsymbol{Z}_t)}{q(\boldsymbol{X}_t \mid \boldsymbol{Z}_t)} \\
&= \frac{p(z_t|\boldsymbol{X}_t, \boldsymbol{Z}_{t-1})\, p(\boldsymbol{x}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{Z}_{t-1})\, p(\boldsymbol{X}_{t-1}, \boldsymbol{Z}_{t-1})}{q(\boldsymbol{x}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{Z}_t)\, q(\boldsymbol{X}_{t-1} \mid z_t, \boldsymbol{Z}_{t-1})} \\
&\propto \frac{p(z_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, p(\boldsymbol{X}_{t-1}, \boldsymbol{Z}_{t-1})}{q(\boldsymbol{x}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{Z}_t)\, q(\boldsymbol{X}_{t-1} \mid \boldsymbol{Z}_{t-1})} \\
&\propto \pi_{t-1}\, p(z_t \mid \boldsymbol{x}_t) \frac{p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}{q(\boldsymbol{x}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{Z}_t)} \\
&= \pi_{t-1}\, p(z_t \mid \boldsymbol{x}_t) \frac{p(\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^c \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)}{q(\boldsymbol{x}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{Z}_t)}.
\end{aligned}
\tag{9}
$$

where $\boldsymbol{X}_t = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$. Since the proposal function $q(\cdot)$ is $\prod_i p(\boldsymbol{x}_t^i \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)$ defined in (5), we have

$$
\pi_t \propto \pi_{t-1} \prod_i p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t^i) \frac{p(\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^c \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)}{\prod_i p(\boldsymbol{x}_t^i \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)}. \tag{10}
$$

Hence, the particle weight is proportional to the product of the likelihoods and the ratio factor defined below:

$$
r(\boldsymbol{x}_t; \boldsymbol{x}_{t-1}) = \frac{p(\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^c \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)}{\prod_i p(\boldsymbol{x}_t^i \mid \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)}. \tag{11}
$$

The denominator of the ratio factor is the product of marginal probabilities, which reflects the probability when we consider different $\boldsymbol{x}_t^i$ independently. On the other hand, its numerator is the joint probability, which considers the inter-dependencies between different $\boldsymbol{x}_t^i$. Hence, the re-weighting process prefers particles for which the joint probability is higher than the product of the marginal probabilities. The stronger is the inter-dependencies between components, the larger is the numerator, and thus the higher is the particle weight.

Since the re-sampling procedure [4] is used as shown in Table I, the particle weight in (10) can be re-written as

$$
\pi_t \propto \prod_i p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t^i) \cdot r(\boldsymbol{x}_t; \boldsymbol{x}_{t-1}). \tag{12}
$$

It is worth noting that, though both [12] and our approach re-weight the particles based on inter-component relationships, a significant difference is that the approach in [12] does not consider the inter-component relationships in the transition model. Although a general graphic model was established, this method simply used the intra-component propagation, $p(\boldsymbol{x}_t^i \mid \boldsymbol{x}_{t-1}^i)$, in its transition model (as described in the first paragraph of Section 3.3 in [12]). However, our approach considers both intra- and inter-component propagations in the transition model. The collaborative transition model formulated can help predict the component states in the next time step. This makes our method capable of recovering the component state when the tracker drifts. We have also experimentally verified that our method performs better than that of [12] in Section V. Since collaboration among components can influence both the dynamic propagation and particle-weight estimation, our framework can be viewed as a general model for part-based tracking.

III. Dynamic Distribution

In order to consider the collaboration among components in (5), we formulate the dynamic model in this section. Inspired by some previous works [17], [18] and without loss of generality, we define the dynamic model as a mixture distribution.

$$
\begin{aligned}
&p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c) \\
&\quad = \alpha_0 \cdot p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^i) + (1-\alpha_0) \cdot p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^{i-1}, \boldsymbol{x}_{t-1}^{i+1}, \ldots, \boldsymbol{x}_{t-1}^c),
\end{aligned}
\tag{13}
$$

where α_0 is a positive weight, $p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^i)$ is the *probability caused by the state transition of the same component* (PSTSC), and $p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^{i-1}, \boldsymbol{x}_{t-1}^{i+1}, \ldots, \boldsymbol{x}_{t-1}^c)$ is the *probability caused by the state transitions of the other components* (PSTOC).

Based on $p(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^1, \ldots, \boldsymbol{x}_{t-1}^c)$, we generate a particle set $\{\boldsymbol{s}_{t;n}^i, n = 1, \ldots, N_i\}$, in which there are $\alpha_0 \cdot N_i$ particles $\{\hat{\boldsymbol{s}}_{t;k}^i, k = 1, \ldots, \alpha_0 \cdot N_i\}$ generated by PSTSC, and the other $(1-\alpha_0) \cdot N_i$ particles $\{\tilde{\boldsymbol{s}}_{t;k}^i, k = 1, \ldots, (1-\alpha_0) \cdot N_i\}$ are generated by PSTOC. In this formulation, we want to generate not only intra-component particles, but inter-components particles, so that the coherence relationships can be employed to boost the tracking performance.

Hence, in TBP particle filtering, the particle set for the i-th component at time t is the union of two sets, $\{\hat{\boldsymbol{s}}_{t;k}^i\}$ and $\{\tilde{\boldsymbol{s}}_{t;k}^i\}$. In the illustration in Fig. 7, we show the concept of particle distributions of these two sets. The middle region represents the particles $\{\hat{\boldsymbol{s}}_{t;k}^i\}$ and the other regions represent the particles $\{\tilde{\boldsymbol{s}}_{t;k}^i\} \equiv \bigcup_{j=1,\ldots,i-1,i+1,\ldots,c} \{\tilde{\boldsymbol{s}}_{t;k'}^{i;j}\}$. Compared to single-chain particle filtering for which the particle set is

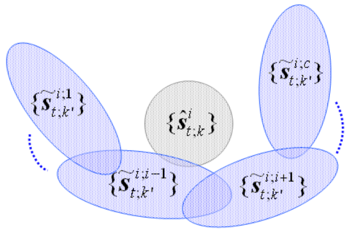


Figure 7. The distributions for particle generation in TBP particle filtering.

only composed of $\{\hat{s}^i_{t;k}\}$, TBP particle filtering allows the use of one more set, $\{\tilde{s}^i_{t;k}\}$, derived from spatial-coherence relationships. In our experiment, the PSTSC, $p(\boldsymbol{x}^i_t | \boldsymbol{x}^i_{t-1})$, is modeled as a Gaussian distribution,

$$p(\boldsymbol{x}^i_t | \boldsymbol{x}^i_{t-1}) \sim \mathcal{N}(\boldsymbol{x}^i_{t-1}, \Sigma),$$

where Σ is a diagonal covariance matrix.

A. General Spatial Constraints

To model the PSTOC, $p(\boldsymbol{x}^i_t | \boldsymbol{x}^1_{t-1},\ldots, \boldsymbol{x}^{i-1}_{t-1}, \boldsymbol{x}^{i+1}_{t-1}, \ldots, \boldsymbol{x}^c_{t-1})$, different types of constraints can be represented consistently. Consider the pair-wise constraint that is often used to model the distance between two components. It is suitable for applications such as body gesture tracking or articulated hand tracking [15], [16]. In this case, possible locations of each component at time t are predicted according to the Euclidean distances from each of the other components at time t–1, and particles are sampled around those locations. In general, the pair-wise constraint can be modeled by the following mixture distribution, where each term in the mixture involves two state variables, namely, $\boldsymbol{x}^i_t$ and $\boldsymbol{x}^j_{t-1}$ ($j = 1,\ldots, c$):

$$p(\boldsymbol{x}^i_t | \boldsymbol{x}^1_{t-1},\ldots,\boldsymbol{x}^{i-1}_{t-1},\boldsymbol{x}^{i+1}_{t-1},\ldots,\boldsymbol{x}^c_{t-1}) = \sum_{j\in\{1,\ldots,c\}, j\neq i} \omega_j \cdot p(\boldsymbol{x}^i_t | \boldsymbol{x}^j_{t-1}), \quad (14)$$

and ω_j is a positive weight. One way to model the distance constraint with the probability $p(\boldsymbol{x}^i_t | \boldsymbol{x}^j_{t-1})$ is to use a ring-like distribution, as shown by the blue region in Fig. 8(a).

For triplet or higher-order constraints, more than two components are considered simultaneously. In this case, strong geometric relationships, such as the angle constraint or similarity-transform constraint, can be considered. For example, a triple-node relationship can model the spatial constraint, where the angle formed by the i-, j-, and k-th components, $\angle\boldsymbol{x}^i\boldsymbol{x}^j\boldsymbol{x}^k$, is approximately fixed, as illustrated in Fig. 8(b). It can also model the similarity-transform constraint when we set $\boldsymbol{x}^j$ as the origin of a coordinate system, $\overrightarrow{\boldsymbol{x}^j\boldsymbol{x}^k}$ as its X-axis (with $\overline{\boldsymbol{x}^j\boldsymbol{x}^k}$ being the unit length), and determine its Y-axis as being perpendicular to the X-axis by the right-hand rule. Then, the coordinate of $\boldsymbol{x}^i$ is fixed to the coordinate system thus defined. An illustration is shown in Fig. 8(c).

To model the triplet constraint, we set the PSTOC by

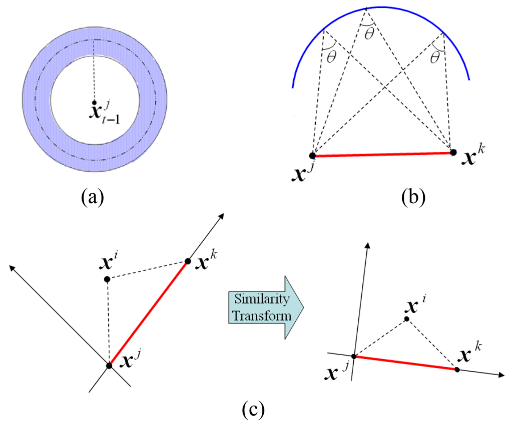


Figure 8. Constraints for modeling the dynamic distribution. (a) Pair-wise constraints. (b) Angle constraints. The blue arc represents some possible locations of $\boldsymbol{x}^i$, for which $\angle\boldsymbol{x}^i\boldsymbol{x}^j\boldsymbol{x}^k$ is fixed. (c) Similarity-transform constraints.

$$p(\boldsymbol{x}^i_t | \boldsymbol{x}^1_{t-1},\ldots,\boldsymbol{x}^{i-1}_{t-1},\boldsymbol{x}^{i+1}_{t-1},\ldots,\boldsymbol{x}^c_{t-1}) = \sum_{j,k\in\{1,\ldots,c\}, i\neq j\neq k} \omega_{jk} \cdot p(\boldsymbol{x}^i_t | \boldsymbol{x}^j_{t-1}, \boldsymbol{x}^k_{t-1}), \quad (15)$$

which is similar to the pair-wise case (14), except each term in the mixture involves three components, $\boldsymbol{x}^i_t$, $\boldsymbol{x}^j_{t-1}$, $\boldsymbol{x}^k_{t-1}$ (j, k = $1\ldots c$), instead of two.

Similarly, the constraints involving more nodes can be induced easily by generalizing (14) and (15). An illustration of the triplet constraint and higher-order constraints are shown in Figs. 9(a) and 9(b), respectively. In practice, what kind of constraints is suitable is problem dependent. In general, it is also allowed to combine pair-wise, triplet, and higher-order constraints in a single mixture to represent the PSTOC. One can see that, the relationship among components is not restricted to a particular form, but is generally unlimited in our approach. Note that standard belief-propagation-based methods of adjusting particle locations at the same time step were designed for pair-wise Markov random fields. To employ high-order constraints, the original graph topology has to be changed by adding function nodes via the factor-graph principle [26], and the complexity will be further increased. Unlike the standard belief-propagation algorithm used for message passing that can employ only pair-wise relationships between components, various spatial constraints can be directly encoded into the PSTOC, which is a general advantage of the TBP-BN formulation.

B. Variations of the Dynamic Model

We have introduced the filtering distributions of the TBP-BN dynamic model. Some possible variations of the dynamic model are addressed below.

1) Partial Connections

The TBP-BN shown in Fig. 3 is fully connected between adjacent layers, i.e., all the components at time t–1 and time t are connected to each other. This forms a complete graph among components if we do not consider the time

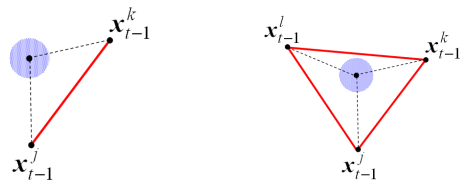


Figure 9. Other constraints. (a) Triplet constraints. (b) Higher-order constraints.

difference.

In many applications, it is not necessary to represent the spatial relationship among components by a complete graph. Thus, we only need to build a partially connected bipartite graph between adjacent layers in Fig. 3. Our approach allows a partially connected graph to be built for TBP-BN. The only restriction is that all the links associated with the same component, i.e., the links between $\boldsymbol{x}_{t-1}^i$ and $\boldsymbol{x}_t^i$, $i = 1 \ldots c$, must be present. In an extreme case, when only these edges are included, the TBP-BN degenerates to c independent chains, as formulated in (1).

Note that the filtering distributions of the dynamic model, no matter whether they are pair-wise, triplet, or higher order, can still be well represented for the partially connected case if the cardinality support is sufficient. The cardinality support of a component at time t is defined as the number of its incoming links from time t–1. More formally, when the cardinality support of a component is m, we can set an m_1-th order constraint among the associated nodes when $m_1 \leq m$ (e.g., when $m = 3$, pair-wise and triplet constraints can be set). With this extension, our approach can be easily generalized to process all partially-connected cases, under the same concept of the TBP particle filtering algorithm presented in Section II.B.

2) *Fixed versus Time-Varying Spatial Coherence*

In above discussion on spatial constraints, we inherently assume that the spatial coherence relationship is fixed during tracking. This is a limitation of our current approach; the relationship is not always fixed, but may vary over time in some applications. For the tracking problems containing significant scale changes or 3D rotations, it will be better to adapt the observation model and/or the spatial relationship over time, so as to enhance the tracking performance. In our framework, there can be no restrictions on the time invariance/variance of the spatial coherence constraints in TBP particle filtering. By adapting a spatial constraint set initially, the spatial relationship can be adjusted or learned incrementally over time, similar to the techniques used in adaptive appearance models based on a single-chain BN [3]. We will investigate the extension in the future.

IV. Likelihood and Particle Weight Estimation

An important issue when dealing with a small region is how to represent its texture effectively and discriminatively. The color histogram approach [1], [12] is an option for textural description, but it is sensitive to illumination changes. To construct a representation to deal with varying illumination and avoid a complex learning process, we use local information to describe a component represented by an image patch. In recent years, many local representation methods have been proposed. Scale-invariant feature transform (SIFT) [27] is a popular local-descriptor technique. However, the need for a relatively long execution time is a limitation of SIFT. In contrast, Huang et al. [28] proposed a *contrast histogram* approach in which the histogram is constructed based on intensity differences. The contrast-histogram description achieves a comparable image matching performance to SIFT, but it is computationally faster. We adopt the contrast histogram [28] for component representation in our work.

A. *Component Representation and Likelihood Measurement*

In this section, we give a brief review of the contrast histogram in [28] and introduce the likelihood measurement used in our method based on the contrast histogram. Given an image patch T, let I_C be the intensity of the center pixel of T. Assume that y is an arbitrary point in T with intensity I_y. The contrast value $C_T(y)$ of pixel y is computed by

$$C_T(y) = I_y - I_C. \tag{16}$$

To achieve rotation invariance, the gradient orientation of the center of T is computed in advance, and the image patch T is rotated according to this orientation. T is then divided into several non-overlapping sub-regions based on the log-polar coordinate system suggested in [29]. For the i-th sub-region T_i, a 2-bin histogram called the contrast histogram is constructed. The values accumulated in the positive bin are defined as

$$B_i^+ = \frac{\sum\{C_T(y) \mid y \in T_i \text{ and } C_T(y) > 0\}}{R_i^+}, \tag{17}$$

where R_i^+ is the number of pixels with positive contrast values in the i-th sub-region.

Similarly, the values in the negative bin are defined as

$$B_i^- = \frac{\sum\{C_T(y) \mid y \in T_i \text{ and } C_T(y) \leq 0\}}{R_i^-}, \tag{18}$$

where R_i^- is the number of pixels with negative contrast values. By composing the contrast histograms of all the sub-regions into a single vector, a contrast-histogram representation of T is defined as

$$H = \{B_1^+, B_1^-, \ldots, B_r^+, B_r^-\}, \tag{19}$$

where r is the number of sub-regions. This vector is then normalized to a unit vector to overcome linear lighting changes. In the log-polar coordinate system applied in our work, there are 8×3 sub-regions for a circular image patch with a diameter of 21 pixels; thus, a 48(8×3×2)-dimensional vector is formed. Then, given a state $\boldsymbol{x}_t^i$ and its local observation z_t^i, we set

$$p(z_t^i | x_t^i) \propto \exp(-||H_0 - H_t||^2 / 2\sigma_1^2), \quad (20)$$

where σ_1 is a variance constant, H_0 and H_t are the contrast-histogram representations at time 0 and time t respectively.

A limitation of the contrast-histogram representation is that it does not perform very robustly on flat areas. Automatic feature selection techniques, e.g., Harris corner, can be used to avoid selecting such areas.

B. Particle Re-weighting

In the above likelihood measurement, appearance information is used. However, in some tracking problems, the appearance of certain components may be very similar. Dividing a target into several local regions causes a problem in that the tracked states of similar components sometimes get trapped in the same location. To avoid this problem, the particle re-weighting scheme in (10) is used, where the ratio factor $r(\boldsymbol{x}_t; \boldsymbol{x}_{t-1})$ is defined by employing spatial constraints as shown below.

Our entire algorithm is given in Table II. In each iteration of this algorithm, the particle sets of each component at the previous time step, $\{(\boldsymbol{s}^i_{t-1;n}, \pi^i_{t-1;n}), n = 1,\ldots, N_i\}$ are given. The final output of this algorithm are state-space vectors of the components, denoted as $\{\Omega_t^i, i = 1\ldots c\}$. In the re-weighting scheme, distance constraints between pairs of components are learned from the previous tracking results. Let $\mu_{t;ij}$ denotes the distance constraint at time t, which is pre-given in the beginning of the tracking, and is updated as the tracking algorithm keeps running as follows:

$$\mu_{t;ij} = \alpha_1 \cdot \mu_{t-1;ij} + (1-\alpha_1) \cdot d(\Omega^i_{t-1}, \Omega^j_{t-1}), \quad (21)$$

where $\alpha_1 \in [0,1]$, $d(\cdot,\cdot)$ denotes the Euclidean distance, and Ω_{t-1} is the tracking output of our algorithm at time $t-1$. For the re-sampled particle set of the j-th component, $\{ \boldsymbol{s}'^j_{t-1;n}, n = 1,\ldots, N_j\}$, its expected value is computed by

$$\bar{\boldsymbol{s}}'^j_{t-1} = \frac{1}{N_j} \sum_n \boldsymbol{s}'^j_{t-1;n}. \quad (22)$$

Then, the ratio factor is estimated by the recent distance relationships between components in our implementation,

$$r(\boldsymbol{x}_t; \boldsymbol{x}_{t-1}) \cong \exp\left(- \sum_{i \in \{1\ldots c\}} \sum_{j \in \{1\ldots c\}} D(\boldsymbol{x}_t^i, \bar{\boldsymbol{s}}'^j_{t-1})^2 / 2\sigma_2^2 \right), \quad (23)$$

where $D(\boldsymbol{x}_t^i, \bar{\boldsymbol{s}}^j_{t-1}) = | d(\boldsymbol{x}_t^i, \bar{\boldsymbol{s}}'^j_{t-1}) - \mu_{t;ij} |$. Note that (23) can be re-written as

$$r(\boldsymbol{x}_t; \boldsymbol{x}_{t-1}) \cong \prod_i \exp\left(- \sum_{j \in \{1\ldots c\}} D(\boldsymbol{x}_t^i, \bar{\boldsymbol{s}}'^j_{t-1})^2 / 2\sigma_2^2 \right). \quad (24)$$

By substituting (24) to (12), the particle weight for each component at time t is re-weighted by

$$\pi^i_{t;n} \propto p(z_t^i \mid \boldsymbol{x}_t^i = \boldsymbol{s}^i_{t;n}) \exp\left(- \sum_{j \in \{1\ldots c\}} D(\boldsymbol{s}^i_{t;n}, \bar{\boldsymbol{s}}'^j_{t-1})^2 / 2\sigma_2^2 \right), \quad (25)$$

Table II. The complete TBP-PF algorithm

Given a set of weighted particles $\{(\boldsymbol{s}^i_{t-1;n}, \pi^i_{t-1;n}), n = 1,\ldots, N_i\}$ for each component $i \in \{1,\ldots, c\}$at time step $t-1$, the following steps are performed to construct a new set of particles at time step t.

1. Re-sample a particle set $\{(\boldsymbol{s}'^i_{t-1;n}, N_i^{-1}), n = 1,\ldots, N_i\}$ from $\{(\boldsymbol{s}^i_{t-1;n}, \pi^i_{t-1;n}), n = 1,\ldots, N_i\}$ for component i.
2. Generate a particle set $\{\boldsymbol{s}^i_{t;n}, n = 1,\ldots, N_i\} = \{\hat{\boldsymbol{s}}^i_{t;k}\} \cup \{\tilde{\boldsymbol{s}}^i_{t;k}\}$ for the i-th component as follows:
   - 2.1. Randomly select $(\alpha_0 \cdot N_i)$ particles $\{\boldsymbol{s}'^i_{t-1;k}, k = 1,\ldots, \alpha_0 \cdot N_i\}$ from $\{\boldsymbol{s}'^i_{t-1;n}\}$, and generate $\{\hat{\boldsymbol{s}}^i_{t;k}, k = 1,\ldots, \alpha_0 \cdot N_i\}$ based on the PSTSC model $p(\boldsymbol{x}_t^i | \boldsymbol{x}^i_{t-1} = \boldsymbol{s}'^i_{t-1;k})$.
   - 2.2. Sample the particle set $\{\tilde{\boldsymbol{s}}^i_{t;k}, k = 1,\ldots, (1-\alpha_0) \cdot N_i\}$ based on the PSTOC model described in Section III.A.
3. Compute the weight $\pi^i_{t,n}$ of sample $\boldsymbol{s}^i_{t;n}$ based on the likelihood and re-weighting procedure in (25).
4. Estimate the state vector $\{\Omega_t^i\}$ by using the maximum mode mentioned in [30] for the display.

where $\boldsymbol{s}^i_{t;n}$ is the n-th particle of the i-th component at time t, $n = 1\ldots N_i$. Hence, for those particles whose distances to the other components are consistent to the distance constraints learned recently, the particle weights will be enhanced. By using this strategy, the re-weighting is a soft spatial adjustment. We can image that there is a spring between components, resulting in a particle weight re-adjustment step that is effective when ambiguity occurs.

V. Experimental Results

A. Implementations

In this section, we present some experimental results to demonstrate the performance of TBP particle filtering. The motion types considered in the experiments are a combination of rough translations and in-plane rotations with slight scaling. Although the motion types are simple, they form strong spatial-coherence relationships suitable for demonstrating the effectiveness of our method. We employ the triplet constraint in the experiments to exploit the strong spatial coherence among the tracked components. The same experiments that employ only pair-wise constraints have also been done for performance comparison. In our current work, we manually specified the image positions of the components in the first frame, and the spatial constraints are modeled based on these positions. We particularly focus on our approach's ability to deal with varying illumination conditions and partial occlusions. During tracking, TBP-BN is fully connected between layers. As discussed in Section III.B, these conditions can be varied in practice according to the application.

A similarity transform with four degrees of freedom (in-plane rotation, scaling, and 2D translation) is used to form

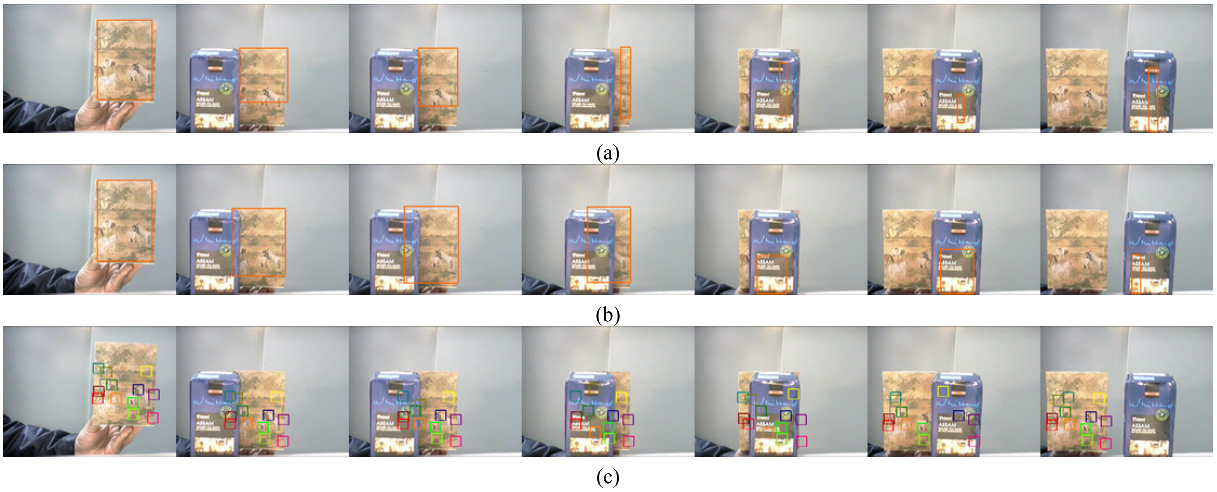


Figure 10. Comparison of our algorithm with global-appearance methods, where the appearance is represented by a color histogram, under occlusions. (a) Results obtained by using the global appearance method, where the tracking drifts easily. (b) Results of the global appearance method with factored likelihood estimation. (c) Results of the proposed method, where the target is stably tracked under different lighting conditions. (This figure is better viewed in color.)

Table III. The average errors in the state space for the fist experiment (Fig. 10)

	Global Appearance Method	Factored Likelihood Estimation	TBP Particle Filtering
Errors	78.15	73.63	**21.56**

a triplet constraint

$$p(\boldsymbol{x}_t^i \mid \boldsymbol{x}_{t-1}^j, \boldsymbol{x}_{t-1}^k) \sim \mathcal{N}(L^i, \Sigma),$$

where L^i denotes a possible location of the i-th component estimated by the similarity transform computed by components j and k. The possible location of the i-th component can then be predicted according to the transformation.

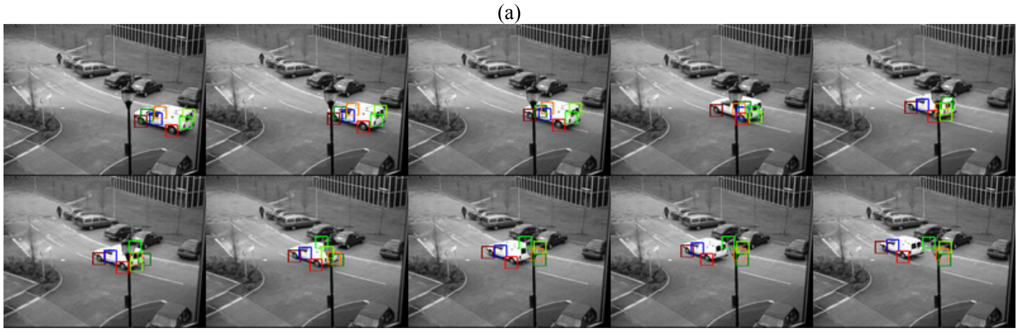
We perform a series of experiments, and compare our approach with methods that use global information and independent-motion assumptions in the situation where significant occlusions and lighting variations occur. In our implementation, two hundred particles are used for each component, and the diameter of the image patch of each component is 21 pixels. The state vector dimension of each component is set as two to identify the 2D location of the component in an image, and α_0 in the proposal function (13) is set at 0.5. The distance parameter μ_{ij} is initialized based on the average distance of the three initial frames, and the spatial difference is computed according to the function $D(\cdot)$ in (23). For the convenience of comparing different methods, we choose the initial parts in our experiments manually. Of course, the proposed method can also be performed by selecting features automatically.

B. Results

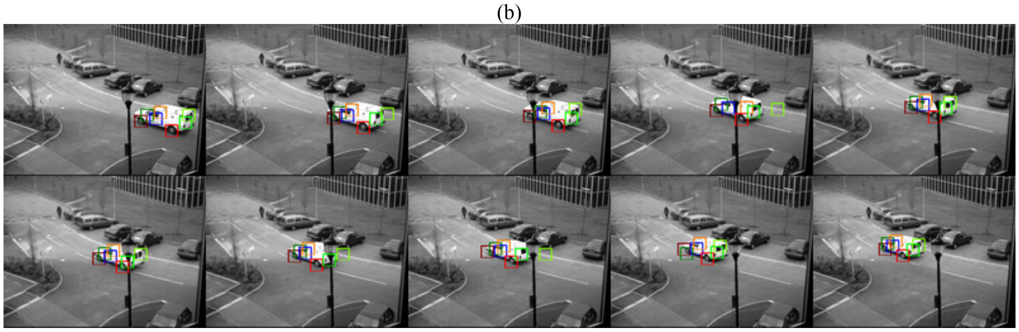
In the first experiment, we apply the proposed method to track an object under substantial occlusions. In this sequence, there are 460 frames and the resolution of a frame is 320×240. The target object is difficult to track since it moves behind another foreground object, resulting in very serious occlusions. The object is represented by thirteen parts in the proposed method. For comparison, the global-appearance approach and the factored-likelihood approach [8], [10] are implemented in a particle filtering framework. The HSV color histograms suggested in [8] are used to represent the global appearance of the object. In the factored likelihood approach, a global region is divided into fixed sub-regions. The partition of the sub-regions remains unchanged, and thus strong spatial constraints among the sub-regions are imposed. The observation is estimated by multiplying the likelihoods of these sub-regions. In our experiment, the appearance is divided into twenty-five sub-regions uniformly. The results of global appearance-based, factored likelihood and our TBP-PF methods are shown in Fig. 10(a), 10(b) and 10(c), respectively. One can see that both the global appearance-based method and the factored-likelihood method drift easily during tracking, particularly when occlusion occurs; however, the proposed TBP particle filtering successfully tracks the components, as shown in Fig. 10(c). In this figure, the small rectangles represent the locations of the components.

To quantify the performance, numerical evaluations are performed. We measure the errors in the state space, and the ground truths are labeled manually in advance. The error is calculated from the average Euclidean distance between the target state and the ground truth in the 4D state space including a 2D location and a 2D size. When our approach is compared with a global approach, a bounding box of all the components is calculated. The location and size of the bounding box are then treated as the predicted state in our method for evaluation. The evaluation results are list in Table III. From these results, one can see that

(a)



(b)



(c)

Figure 11. Comparison of part-based tracking methods with different strategies. (a) Results obtained from the method with independence motion assumption among components, where component trackers drift easily and mislead when the car passes the lamp-post. (b) Results obtained by using the particle re-weighting strategy only, where drifted components still can not be recovered. (c) Results of the proposed method, where the target is successfully tracked, even when occlusions occur. (This figure is better viewed in color.)

TBP particle filtering outperforms the other methods.

In another experiment, we compare our method with some multi-component tracking methods. A gray-scale sequence consisting of 30 frames is employed. Seven components are selected to represent the vehicle. In the first method, we assume that the spatial coherence among components is neglected by setting the component motions to be independent. A component is regarded as occluded when its likelihood is less than a given threshold, and its state will not be displayed. The results of independent-motion tracking are shown in Fig. 11(a). Since spatial coherence is not considered, the components are easily trapped in a local minimum. Components of similar appearance also mislead the tracker when the vehicle passes by the lamp-post. Second, the particle re-weighting method is used by applying the distance constraint in particle weight estimation (as introduced in Section IV.B), so that the particle weights can be refined by the spatial constraints. In this method, the spatial coherence is used only for particle weight refinement, but not for state prediction through system dynamics. Fig. 11(b) shows the results. Though the problem of components with similar appearance can be reduced by this method, it still cannot provide good results. In particular, a component's location is difficult to recover once it drifts. Finally, we adopt the proposed method, which encodes the triplet spatial coherence in the dynamics of the same sequence. Our method can resolve the difficulties caused by occlusions effectively, and convincing results are obtained, as shown in Fig. 11(c). The method of applying pair-wise constraints to system dynamics is also implemented for comparison. We show the numerical results of this experiment in the third column of Table IV, where the error is measured from the average distance between the components' state and the ground truth. We also implemented the method proposed in [12] which is based on auxiliary particle-filtering. This method also re-weights the particles according to spatial relationships but has not employed the spatial relationships in the dynamic model. In our implementation, a similarity

(a)

(b)

(c)

(d)

Figure 12. Comparison of tracking under both varying illumination conditions and partial occlusions. (a) Results obtained using the global appearance method. (b) Results obtained by the method based on the independence motion assumption among components, where the trackers of components drift easily and trap in the same location. (c) Results obtained by using particle re-weighting strategy only, where drifted components are hard to recover. (d) Results of the proposed method, where the target tracking is stable. (This figure is better viewed in color.)

transformation was used to model the function $q(\cdot)$ defined in [12]. Because the collaborative dynamic model is used, our method still outperforms that in [12] as shown in the results.

In the third experiment, we consider a more difficult case where both lighting variations and partial occlusions occur at the same time. This sequence contains approximately 430 frames of human face that is also divided into five parts. A skin-colored paper is used to occlude the face. Some comparisons are shown in Fig. 12. Fig. 12(a) shows the tracking results by using global-appearance information, where the tracker is unstable and is easily misled by the skin-colored paper. We also show the tracking results based on the independence motion assumption in Fig. 12(b), while those that employ spatial coherence for particle re-weighting are shown in Fig. 12(c). However, the performance of the above methods is still not satisfactory. In contrast, by encoding the spatial coherence in the dynamics, our method can track the target in this sequence very well, as shown in Fig. 12(d). The numerical evaluations are given in the forth column of Table IV.

Finally, we apply our method to the image sequence with deformations including scaling and expression changes. Convincing tracking results can still be obtained under the scale and expression variations, as shown in Fig. 13, and the numerical evaluations are given in the fifth column of Table IV. More tracking results on the *Dudek* sequence [31] are given in Fig. 14.

With regard to efficiency, our method runs at about 7~8 fps on a 2.8 GHz PC with non-optimized C codes when five components are considered. This is much faster than methods that use belief-propagation-based or variation-based iterative particle refinement in similar settings.

VI. CONCLUSION AND FUTURE WORK

We have presented a general part-based approach that employs component collaboration for visual tracking, and formulated the problem as a Bayesian network called TBP-BN in which inter-component relationships are modeled stage-wisely. A probability propagation

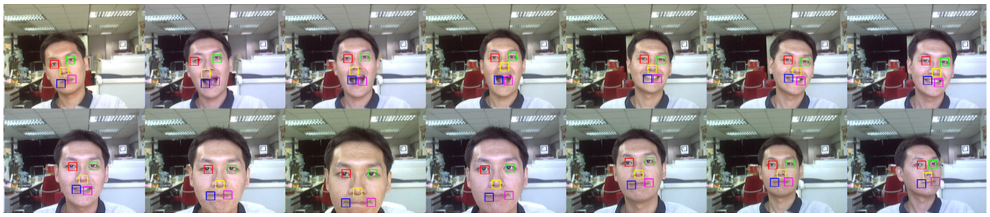


Figure 13. Results of face tracking with expressions and scaling by using TBP particle filtering. (This figure is better viewed in color.)

Figure 14. Tracking results of the *Dudek* sequence. (This figure is better viewed in color.)

Table IV. The average errors in the state space for the second, third and forth experiments

		Exp 2 (Fig. 11)	Exp 3 (Fig. 12)	Exp 4 (Fig. 13)
Global Approach	Global Appearance Method	--	70.95	--
Part-based Approaches	Tracking with independence motion assumption	32.97	21.02	24.52
	Particle Re-weighting Method	16.90	8.26	14.52
	Particle Filtering with Factorized Likelihoods [12]	14.21	10.90	9.80
	TBP Particle Filtering (Pair-wise constraint)	6.14	4.36	6.90
	TBP Particle Filtering (Triplet constraint)	**4.78**	**3.41**	**4.63**

framework was derived to find the posterior distribution of TBP-BN, and TBP particle-filtering was developed to realize the stage-wise probability propagation process. We have also provided a theoretical foundation to show that the dynamic model of TBP-BN can be separated in a component-based manner.

Our method not only uses spatial coherence for particle weight measurement, but also for temporal-based propagation. Because of this characteristic, spatial-structural information can be propagated to predict components' locations, and part-based information can be employed more efficiently.

We adopt a particle re-weighting procedure for part-based tracking. However, our framework also allows the use of belief-propagation-based iterations for refinement (when the computation speed is not an important issue), so as to make more complete use of spatial structural information. Therefore, TBP-BN provides a new and general way to encode spatial coherence into tracking algorithms. The experimental results show that our approach performs well in several situations.

In the future, we will extend our approach to an adaptive one that can learn the spatial coherence relationships and the observation model over time. We will also consider the concept of interactive collaboration [14] to further enhance the particle weight estimation when tracking similar components that are spatially close. By modifying the appearance measures employed and building suitable partially-connected bipartite graphs between adjacent layers, we will study the above problems in the future.

Acknowledgment. This work was supported in part under Grants NSC 96-3113-H-001-011, 95-2221-E-001-028-MY3, and 95-2221-E-002-209-MY3. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 142-151, 2000.

[2] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, pp. 1064-1072, 2004.

[3] D. Ross, J. Lim, and M. H. Yang, "Adaptive probabilistic visual tracking with incremental subspace update," *Proceedings of European Conference on Computer Vision*, Vol. 2, pp. 470-482, 2004.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nolinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 174-188, 2002.

[5] M. Isard and A. Blake, "ICONDENSATION- Unifying low-level and high-level tracking in a stochastic framework," *Proceedings of European Conference on Computer Vision,* Vol. 1, pp. 893-908, 1998.

[6] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, pp. 1475-1490, 2004.

[7] M. Pantic and I. Patras, "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* Vol. 36, No. 2, pp. 433-449, 2006.

[8] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Proceedings of European Conference on Computer Vision*, Vol. 1, pp. 661-675, 2002.

[9] L. Sigal, Y. Zhu, D. Comaniciu, and M. J. Black, "Tracking complex objects using graphical object models," *Proceedings of International Workshop on Complex Motion*, 2004.

[10] G. Hua and Y. Wu, "Measurement Integration under Inconsistency for Robust Tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 650-657, 2006.

[11] T. Yu and Y. Wu, "Differential Tracking based on Spatial-Appearance Model," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 720-727, 2006.

[12] I. Patras and M. Pantic, "Particle filtering with factorized likelihoods for tracking facial features," *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 97-102, 2004.

[13] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 560-576, 2001.

[14] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model," *Proceedings of IEEE International Conference on Computer Vision*, Vol. 1, pp. 535-540, 2005.

[15] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky, "Distributed occlusion reasoning for tracking with nonparametric Belief propagation," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1369-1376, 2004.

[16] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic Markov network," *Proceedings of IEEE International Conference on Computer Vision*, pp. 1094-1101, 2003.

[17] T. Yu and Y. Wu, "Decentralized multiple target tracking using netted collaborative autonomous trackers," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 939-946, 2005.

[18] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," *Proceedings of European Conference on Computer Vision*, Vol. 1, pp. 28-39, 2004.

[19] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multi-modality through mixture tracking," *Proceedings of IEEE International Conference on Computer Vision*, Vol. 2, pp. 1110-1116, 2003.

[20] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," *Proceedings of IEEE International Conference on Computer Vision*, Vol. 1, pp. 212-219, 2005.

[21] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufman, San Mateo, 1988.

[22] T. S. Jaakkola, "Tutorial on variational approximation method," *MIT AI Lab Technical Report*, 2000.

[23] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, Vol. 37, No. 2, pp. 183-233, 2000.

[24] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric belief propagation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 605-612, 2003.

[25] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," *Proceedings of European Conference on Computer Vision*, Vol. 4, pp. 279-290, 2004.

[26] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, pp. 498-519, 2001.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[28] C. R. Huang, C. S. Chen, and P. C. Chung "Contrast context histogram- A discriminating local descriptor for image matching," *Proceedings of International Conference on Pattern Recognition*, Vol. 4, pp. 53-56, 2006.

[29] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 4, pp. 509-522, 2002.

[30] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, Vol. 61, No. 2, pp. 185-205, 2005.

[31] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, pp. 1296-1311, 2003.