A Probabilistic Approach to Integrating Multiple Cues in Visual Tracking

Wei Du and Justus Piater

University of Liège, Department of Electrical Engineering and Computer Science
Montefiore Institute, B28, B-4000 Liege, Belgium
{wei.du,justus.piater}@ulg.ac.be

Abstract. This paper presents a novel probabilistic approach to integrating multiple cues in visual tracking. We perform tracking in different cues by interacting processes. Each process is represented by a Hidden Markov Model, and these parallel processes are arranged in a chain topology. The resulting Linked Hidden Markov Models naturally allow the use of particle filters and Belief Propagation in a unified framework. In particular, a target is tracked in each cue by a particle filter, and the particle filters in different cues interact via a message passing scheme. The general framework of our approach allows a customized combination of different cues in different situations, which is desirable from the implementation point of view. Our examples selectively integrate four visual cues including color, edges, motion and contours. We demonstrate empirically that the ordering of the cues is nearly inconsequential, and that our approach is superior to other approaches such as Independent Integration and Hierarchical Integration in terms of flexibility and robustness.

1 Introduction

From a Bayesian perspective, visual tracking is viewed as a problem of inferring target states over time based on image features or cues. Various types of cues have been used to characterize different object properties, including color [1], texture [2], points [3], edges [4], motion [5] and contours [6]. As no single cue remains reliable in all situations, the integration of multiple cues has proved successful at increasing the robustness of tracking algorithms. For instance, color and appearance are more sensitive to the lighting conditions than gradient features such as edges and points. Therefore, when the scene is subject to fast illumination changes, edges and points may provide complementary information that helps localize the targets being tracked.

Basically, multi-cue tracking is a fusion problem. Each cue tells a story about the targets of interest, and this information is processed and fused to estimate target states. One important issue is how to model the dependence between different cues, which in turn determines the manner in which the cues are combined. Many methods assume that the cues are conditionally independent [7–10], while more sophisticated methods model existing dependencies explicitly by e.g. graphical models [11].

This paper presents a novel probabilistic approach to integrating multiple cues in visual tracking. In contrast to previous work, we perform tracking in different cues by individual but interacting processes, each of which is represented by a Hidden Markov Model (HMM). Chain models are used to link these parallel HMMs and to represent the dependence between these processes. The resulting Linked Hidden Markov Models (lHMMs)[1] naturally allow the use of two powerful inference algorithms, particle filtering and Belief Propagation (BP). By combining them in a unified framework, a Sequential Auxiliary Particle Belief Propagation algorithm is devised to perform inference in multi-cue tracking. In particular, a target is tracked in each cue by a particle filter, and the particle filters in different cues interact with each other via a message passing scheme.

Our approach has several advantages. First, it allows different target representations to be used in different cues so that each tracking process can be implemented separately and efficiently. Second, the approach is highly modular, facilitating the combination, addition and removal of vastly different cues. One only needs to define the potential function between each pair of neighboring cues according to their target representations. Third, the chain topology of the lHMMs reduces the complexity of the integration framework with respect to more elaborate graphical models. Due to the bidirectional propagation of information along the chain, the order of the cues in the chain is largely unimportant. We confirmed empirically that changing this order hardly affects the tracking results.

The rest of the paper is organized as follows. Section 2 discusses related work and highlights our contributions. Section 3 describes the lHMMs that model the multi-cue tracking problem. Section 4 formulates the problem and introduces the inference algorithm. Experimental results are presented in Section 5.

2 Related Work

Numerous approaches to multi-cue tracking have been reported in the literature. They differ in the way the cues are integrated and in the cues adopted. For example, Birchfield combined gradient and color cues for head tracking [7]. Triesch et al. used democratic integration to compute the consensus between the multiple cues [8]. Taylor et al. fused color, edge and texture cues in a Kalman-filter framework [2].

Particle filters, also known as Condensation [6] in the computer vision community, have achieved great success in solving tracking problems. Conventional particle filters maintain the target distributions over time based on a single observation model such as color [1] or contours [6]. It has been shown that multiple cues can easily be fused within a particle-filter framework. By assuming that the cues are conditionally independent, multi-cue observations were integrated by the product [9, 13, 14] or the sum [15] of the likelihoods in different cues. Based

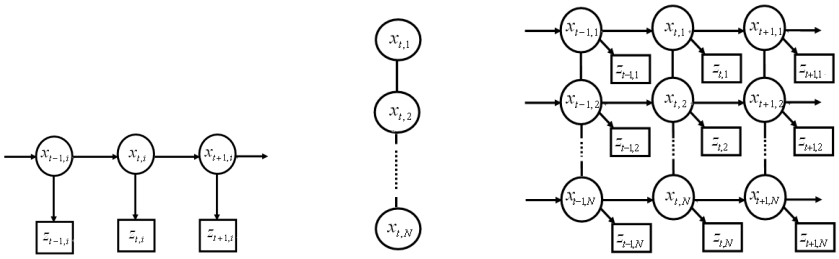
[1] The term *Linked Hidden Markov Models* was introduced by Brand as a way of modeling two interacting processes [12]. Here, we borrow this term and extend it to multiple interacting processes.

on the same independence assumption, Leichter et al. combined Condensation-based or explicit PDF-yielding trackers by fusing only the trackers' output estimates [10]. Although not explicitly stated, the approach required each tracker to integrate information from all the other trackers, implying a fully-connected architecture. On the other hand, Pérez et al. propagated target distributions from cue to cue in a fixed order, hoping that downstream cues resolve the ambiguities introduced by the upstream cues [5]. Similar ideas were introduced under the names of hierarchical particle filters [4] and cascades of cues [16]. One downside of this strategy is that the performance depends on the order in which the cues are incorporated. Generally, heuristics are required to design this order. Wu et al. used factorial HMMs to model the dependency between color and contour cues and proposed a so-called co-inference algorithm [11].

Inspired by some of the cited work [10, 5, 11], this paper presents a general framework for multi-cue integration. Similarly to Leichter et al. [10], we combine particle filter-based trackers, each of which exploits a different cue. However, the dependencies between these trackers are explicitly modeled using lHMMs. The lHMMs link a set of parallel HMMs in a chain topology, which largely reduces both the architectural and algorithmic complexities. In doing so, we synchronously infer the target states in different cues. Unlike Pérez et al. [5] who propagated information from cue to cue in one direction, the undirected links in the chains in our lHMMs enable bidirectional message passing between the cues, relaxing the dependency on the ordering of the cues. The suggested approach is more general than e.g. that of Wu et al. [11] as it allows the integration of an arbitrary number of cues.

The combination of particle filters and BP was originally motivated for inference under non-linear and non-Gaussian models, resulting in Nonparametric Belief Propagation (NBP) [17, 18]. Hua et al. first formulated the inference in temporally evolving graphical models and proposed a Sequential Belief Propagation (SBP) algorithm [19]. Briers et al. addressed the computational issue in the same sequential-inference context [20]. Auxiliary particle filters and the unscented approximation were used to sample particles, avoiding the need for Gibbs sampling. The resulting Sequential Auxiliary Particle Belief Propagation (SAPBP) reduced the computation from quadratic in the number of particles, as in NBP, to linear, which is desirable for online inference.

In this paper, we adopt a simplified SAPBP algorithm to solve the multi-cue tracking problem. The algorithm integrates the temporal evolution of each cue and the inter-cue correlations into a coherent framework. While independent temporal transition kernels are used for each cue, target states in different cues are related through messages passed along a chain. Contrary to the original SAPBP algorithm, we do not use the unscented approximation, as the inter-cue pairwise potentials are linear. Four visual cues are selected to demonstrate the effectiveness of our approach including color, edges, motion and contours. The general framework facilitates a customized combination of different cues in different situations, which is particularly desirable from the implementation point of view. Extensive experiments on tracking various objects in both indoor and



(a) *HMM* **(b) *Chain*** **(c) *lHMMs***

Fig. 1. (a) HMM representing the tracking process in one cue. (b) Chain model representing the dependencies between different cues at a time instant. (c) Linked HMMs representing the interacting processes in different cues.

outdoor environments show that our approach is more flexible and robust than other approaches such as *independent integration* and *hierarchical integration*.

3 Linked Hidden Markov Models

Suppose M visual cues are used and each cue is associated with a different target state. Let $x_{t,i}$ be the target state in the ith cue at time t and $z_{t,i}$ the associated image observation, $i = 1, \ldots, M$. Given these definitions, tracking in the ith cue is formulated as

$$p(x_{t,i}|z_i^t) \propto p(z_{t,i}|x_{t,i}) \int p(x_{t,i}|x_{t-1,i}) p(x_{t-1,i}|z_i^{t-1}) \mathrm{d}x_{t-1,i}, \tag{1}$$

where $z_i^t = \{z_{1,i}, \ldots, z_{t,i}\}$, $p(z_{t,i}|x_{t,i})$ is the image likelihood and $p(x_{t,i}|x_{t-1,i})$ is the temporal transition kernel in the cue. This Bayesian formulation can be represented by a HMM, shown in Fig. 1(a).

In multi-cue tracking, the dependencies between target states at different cues must be taken into consideration. We use graphical models to capture these dependencies. Theoretically, fully-connected graphical models are required since all states in different cues may depend on each other. However, we choose to adopt a chain model (Fig. 1(b)) to reduce the model complexity and to simplify the inference algorithm. Linking the HMMs into chains results in so-called Linked Hidden Markov Models or lHMMs, shown in Fig. 1(c). The lHMMs represent the interacting processes in different cues.

Let $X_t = \{x_{t,1}, \ldots, x_{t,M}\}$ denote the multi-cue target state and $Z_t = \{z_{t,1}, \ldots, z_{t,M}\}$ the multi-cue image observation. Thus, tracking in the multiple cues amounts to the recursive inference of X_t, formulated as

$$p(X_t|Z^t) \propto p(Z_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Z^{t-1}) \mathrm{d}X_{t-1}, \tag{2}$$

where $Z^t = \{Z_1, \ldots, Z_t\}$.

Direct inference using Eq. 2 is intractable due to the high dimensionality of the state space and the non-Gaussian nature of the target distributions. Therefore, we infer each $p(x_{t,i}|Z^t)$, $i = 1, \ldots, M$, collaboratively by a set of particle-filter-based processes, as detailed below.

4 Multi-Cue Tracking by Interacting Processes

4.1 Problem Formulation

We first consider the inference in the chain model in Fig. 1(b). BP performs inference in graphical models by first computing messages and then computing beliefs. The chain topology admits a two-pass message-passing implementation. Specifically, the local messages passed from top to bottom are defined by

$$m_{i,i+1}(x_{t,i+1}) \propto \int p(z_{t,i}|x_{t,i}) m_{i-1,i}(x_{t,i}) \psi_{i,i+1}(x_{t,i}, x_{t,i+1}) \mathrm{d}x_{t,i}, \tag{3}$$

where $\psi_{i,j}$ is the potential function that describes the dependency between node i and j. The definition of this potential function depends on the target representations in the two neighboring cues and will be discussed later. Likewise, the messages passed from bottom to top have a symmetric form,

$$m_{i,i-1}(x_{t,i-1}) \propto \int p(z_{t,i}|x_{t,i}) m_{i+1,i}(x_{t,i}) \psi_{i-1,i}(x_{t,i-1}, x_{t,i}) \mathrm{d}x_{t,i}. \tag{4}$$

Then, the belief of $x_{t,i}$ is obtained by

$$p(x_{t,i}|Z_t) \propto p(z_{t,i}|x_{t,i}) m_{i-1,i}(x_{t,i}) m_{i+1,i}(x_{t,i}). \tag{5}$$

Note that Eqs. 3–5 are slightly different for the nodes at the ends of the chain.

In the sequential context, the message and belief equations for the lHMMs in Fig. 1(c) have similar forms as Eqs. 3, 4 and 5, adding only the terms of the temporal priors,

$$m_{i,i+1}(x_{t,i+1}) \propto \int p(z_{t,i}|x_{t,i}) p(x_{t,i}|Z^{t-1}) m_{i-1,i}(x_{t,i}) \psi_{i,i+1}(x_{t,i}, x_{t,i+1}) \mathrm{d}x_{t,i} \tag{6}$$

$$m_{i,i-1}(x_{t,i-1}) \propto \int p(z_{t,i}|x_{t,i}) p(x_{t,i}|Z^{t-1}) m_{i+1,i}(x_{t,i}) \psi_{i-1,i}(x_{t,i-1}, x_{t,i}) \mathrm{d}x_{t,i} \tag{7}$$

$$p(x_{t,i}|Z^t) \propto p(z_{t,i}|x_{t,i}) p(x_{t,i}|Z^{t-1}) m_{i,i-1}(x_{t,i}) m_{i,i+1}(x_{t,i}), \tag{8}$$

where the temporal prior is

$$p(x_{t,i}|Z^{t-1}) = \int p(x_{t,i}|x_{t-1,i}) p(x_{t-1,i}|Z^{t-1}) \mathrm{d}x_{t-1,i}.$$

Thus, the sequential inference in the lHMMs consists of passing messages using Eqs. 6 and 7 followed by belief updating using Eq. 8. Due to the lack of analytic representations of the above formulations, Monte Carlo approximations are required and can be obtained by using importance sampling techniques. A technical issue is the design of proper importance functions. We address this by adapting Sequential Auxiliary Particle Belief Propagation [20].

4.2 Sequential Auxiliary Particle Belief Propagation

The Monte Carlo approximation to Eqs. 6–8 requires the sampling of particles from products of individual terms, that is, from $p(x_{t,i}|Z^{t-1})m_{i-1,i}(x_{t,i})$ in Eq. 6, from $p(x_{t,i}|Z^{t-1})m_{i+1,i}(x_{t,i})$ in Eq. 7, and from $p(x_{t,i}|Z^{t-1})m_{i-1,i}(x_{t,i})m_{i+1,i}(x_{t,i})$ in Eq. 8. This sampling task requires an efficient way of combining, at each cue, the temporal prior and the incoming messages from the neighboring cue or cues.

Assume there are K terms in the product and each term consists of N particles. In order to compute the product of these terms, each term is represented as a mixture of Gaussians, and the computation amounts to multiplying K Gaussian mixtures of N components, which has a complexity of $O(N^K)$. Gibbs sampling is generally used for approximation and reduces the complexity to $O(\kappa KN^2)$, where κ is the number of iterations [17, 18]. However, the chain topology of our lHMMs makes the Gibbs sampler ineffective since the number of terms K is only 2 for the messages and 3 for the beliefs. In the following, we will refer to the full multiplication of the terms as the full-product NBP in contrast to the traditional Gibbs-sampling NBP.

For real-time applications such as tracking, a more efficient method is required. By assuming that the terms in the product are independent, the complexity of this computation can be further reduced to $O(KN)$ by using SAPBP.

SAPBP adopts a two-step sampling procedure that first samples a component label from each term to construct an importance function and then samples particles from the constructed importance functions. An auxiliary variable $\theta_k \in \{1, \ldots, N\}$, $k = 1, \ldots, K$, is introduced to denote the component label in the kth term. As the component labels in different terms are independent, i.e., $p(\theta_1, \ldots, \theta_K) = \prod_{k=1}^{K} p(\theta_k)$, the importance function for sampling θ_k is chosen as

$$q(\theta_k = l_k) \propto \int \phi(x) f_k^{l_k}(x) \mathrm{d}x,$$

where $\phi(x)$ is the likelihood term and $f_k^{l_k}(x)$ is the l_kth component in the kth term. Sampling θ_k from this importance function is analogous to an auxiliary particle filter and is thus computationally efficient.

In our lHHMs, this label-sampling procedure is much simpler than in a general graphical model since the products contain few terms. For instance, to compute the messages in Eq. 6, we need to sample from $p(x_{t,i}|Z^{t-1})m_{i-1,i}(x_{t,i})$. The importance functions for sampling component labels from the above two terms are given by

$$q(\theta_1 = l_1) \propto \int p(z_{t,i}|x_{t,i}) p(x_{t,i}|x_{t-1,i}^{l_1}) \mathrm{d}x_{t,i},$$
$$q(\theta_2 = l_2) \propto \int p(z_{t,i}|x_{t,i}) \psi_{i,i-1}(x_{t,i}, x_{t,i-1}^{l_2}) \mathrm{d}x_{t,i},$$

where x_i^l denotes the lth particle for the state x_i. Similarly, to compute the beliefs in Eq. 8, the three importance functions for sampling component labels

from $p(x_{t,i}|Z^{t-1})m_{i-1,i}(x_{t,i})m_{i+1,i}(x_{t,i})$ are given by

$$q(\theta_1 = l_1) \propto \int p(z_{t,i}|x_{t,i})p(x_{t,i}|x_{t-1,i}^{l_1})\mathrm{d}x_{t,i},$$
$$q(\theta_2 = l_2) \propto \int p(z_{t,i}|x_{t,i})\psi_{i,i-1}(x_{t,i}, x_{t,i-1}^{l_2})\mathrm{d}x_{t,i},$$
$$q(\theta_3 = l_3) \propto \int p(z_{t,i}|x_{t,i})\psi_{i,i+1}(x_{t,i}, x_{t,i+1}^{l_3})\mathrm{d}x_{t,i}.$$

After a label is sampled for each of the K terms, a Gaussian distribution is formed by the product of these individual (Gaussian) components and is used as an importance function $q(x_i|\theta_1, \ldots, \theta_K)$, from which a particle is generated. In this way, N particles are sampled for each product. These sampled particles are used to approximate the messages and beliefs in Eqs. 6–8. The normalized weights for the messages (for Eq. 6 only) and beliefs (for Eq. 8) are given by

$$w_{i,i+1}^{t,n} = \frac{p(z_{t,i}^n|x_{t,i}^n)p(x_{t,i}^n|x_{t-1,i}^{\theta_1})w_{i-1,i}^{t,\theta_2}\psi_{i,i-1}(x_{t,i}^n, x_{t,i-1}^{\theta_2})}{q(\theta_1)q(\theta_2)q(x_{t,i}^n|\theta_1, \theta_2)},$$
$$w_{t,i}^n = \frac{p(z_{t,i}^n|x_{t,i}^n)p(x_{t,i}^n|x_{t-1,i}^{\theta_1})w_{i-1,i}^{t,\theta_2}\psi_{i,i-1}(x_{t,i}^n, x_{t,i-1}^{\theta_2})w_{i+1,i}^{t,\theta_3}\psi_{i,i+1}(x_{t,i}^n, x_{t,i+1}^{\theta_3})}{q(\theta_1)q(\theta_2)q(\theta_3)q(x_{t,i}^n|\theta_1, \theta_2, \theta_3)}.$$

The intuition behind this algorithm is that, to combine the temporal priors and the incoming messages, we ignore the dependencies between them and sample independently from them by taking into account the anticipated "merit", i.e., the likelihoods evaluated after applying the temporal-transition kernel (for particles sampled from the temporal priors) or the inter-cue potentials (for particles sampled from the incoming messages). Ignoring the dependencies between the terms in a product may cause some loss of accuracy. However, in our case, this loss is unlikely to be dramatic since there are only two or three terms in each product. Moreover, the above sampling procedure takes into account the common dependence on the underlying targets as it is guided by the likelihood term. A comparison of our SAPBP algorithm with the full-product NBP will be shown in Section 5.

Our algorithm treats each cue equally with no explicit preference. However, as pointed out by Sun et al. [21], the asymmetric message passing in BP guarantees that the information is propagated mainly from high-confidence cues to low-confidence cues due to the smaller entropy of the messages in this direction. An extreme example is when the appearance of the target changes completely. Then, the color cue still "contributes" by propagating mostly uniformly distributed beliefs. Although this cue is not informative, it will not affect the tracking results at other cues.

4.3 Cues and Inter-Cue Potentials

For this paper, we chose four simple and complementary cues including color, edges, motion and contours. The general framework of our approach allows an easy customization for various scenarios by combining different cues.

Color Two different color histogramming methods are implemented. The first one is the traditional color histogramming proposed by Pérez et al. [1]. To model the spatial layout of the color distribution, a multi-part color model is obtained by splitting the tracked region into subregions, each with an individual color model. The second method is spatiogramming proposed by Birchfield et al. [22], which models the spatial layout of each color bin by a Gaussian distribution.

Edge Edge orientation histogramming is used to disambiguate confusing colors in the background. Edges are detected using the horizontal and vertical Sobel operators, and the gradient magnitudes and orientations are computed. The detected edges are then histogrammed into orientation bins weighted by their strengths [4].

Motion In the case of a static camera, the absolute image difference between each pair of successive frames provides motion information about the targets being tracked. A motion histogram is constructed by treating this absolute image difference as a grayscale image and histogramming the pixels in the regions of interest into intensity bins. To compute detection likelihoods, each candidate histogram is then compared to a uniform reference histogram [5].

Contour The gradient information along the contour of a target is collected. A set of points are sampled along the contour and the distances to the closest edge points are computed and converted to a likelihood density.

In the color, edge and motion cues, we extract the histogram associated with each particle and compare it to a pre-learned reference histogram. This procedure was implemented using integral histograms [23]. In the contour cue, the Condensation algorithm [6] was implemented using a distance transformation. As the construction of the integral histograms and the distance maps is cheap with cost proportional to the size of the region of interest, we are able to evaluate a large number of particles in real time. In general, edge and motion cues alone are not discriminative, as edge orientation histograms do not capture the spatial distribution of the edges, and motion histograms favor equally all regions with moving objects. Nevertheless, integrating them does help disambiguate clutter in color and contour cues.

We model the targets of interest in the first three cues by rectangular regions, and in the contour cue by parametrized shape models. Ellipses and circles are adopted to model the simple targets in our experiments. Thus, a target is separately represented by

$$x_t^{\text{color}} = x_t^{\text{edge}} = x_t^{\text{motion}} = [u_t, v_t, s_t^u, s_t^v],\ x_t^{\text{contour}} = [u_t, v_t, s_t^u, s_t^v, \theta_t],$$

where $[u_t, v_t]$ is the translation of the target, $[s_t^u, s_t^v]$ is the scale change, and θ_t is the relative rotation when elliptic models are used in the contour cue. Representing the same target in different cues differently allows a separate and efficient implementation of each tracking process.

Given the target representations in different cues, we can define the potential functions ψ between each pair of neighboring cues. When the two neighboring

cues share the same target representations, i.e., when color, edge and motion cues are neighboring to each other, then the potential between them is given by

$$\psi_{i,j}(x_{t,i}, x_{t,j}) = G(x_{t,i} - x_{t,j}; 0, \Sigma_{i,j}),$$

where $G(\cdot)$ is the Gaussian kernel function and $\Sigma_{i,j}$ is the covariance matrix. When different target representations are used in the two neighboring cues, i.e., when the contour cue is neighboring to the color, edge or motion cues, a similar potential function is defined by ignoring θ_t in x_t^{contour}. This inter-cue potential function models the mutual influence between the target states at different cues, and can be similarly defined for other target representations.

5 Results

We tested the performance of our approach on sequences of various objects taken in both indoor and outdoor environments. Here, the constant-velocity motion model was used for all cues in all experiments. We manually initialized the targets of interest in the first frame of each sequence and learned the reference models. The reference models were updated gradually with exponentially forgetting the past models, or were kept unchanged when dramatic changes to the models indicated occlusions [24].

The first experiment was tracking a dish under poor illumination, shown in Fig. 2. Four cues were integrated in the order of color (color histogram), edge, motion and contour. This sequence is difficult in that it contains both occlusions and fast appearance changes due to the mirror surface of the dish and the shadows. Thus, the color cue is only discriminative when the observed appearance is similar to the reference model. The motion cue helps the tracker concentrate on the areas with moving objects. As the edge and contour cues are based on gradient features, they are more reliable and play an important role during tracking. No single cue alone was able to track the dish for the entire sequence.

The second experiment was tracking a pedestrian in one of the PETS sequences and tracking a vehicle in a traffic sequence, shown in Fig. 3. The edge, color (color histogram) and motion cues were integrated in the given order for both sequences. The PETS sequence contains fast illumination changes and there is a camera rotation in the middle of the traffic sequence in order to keep the vehicle inside the view. Both problems violated the fundamental assumption of the motion cue that absolute image differences are introduced by moving objects only. However, thanks to the bidirectional information propagation between the cues, the poor information in the motion cue did not break the whole tracking system. In this experiment, the color cue alone is sufficient to track the targets most of the time. The motion cue helps overcome the partial occlusions and the edge cue helps localize the targets more precisely.

In the third experiment, we tested how dependent our approach was on the ordering of the cues. Two face sequences were selected to perform this test. We integrated all four cues for the first sequence and left out the motion cue for the

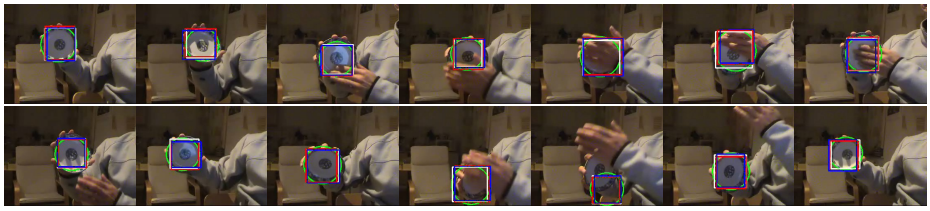


Fig. 2. Results of tracking a dish under poor illumination. The circles and the rectangles are the estimates of the contour (green), color (white), motion (blue) and edge (red) cues. The estimates by the four cues are slightly different from each other due to the Monte Carlo simulation. The same colors are used in the following experiments.

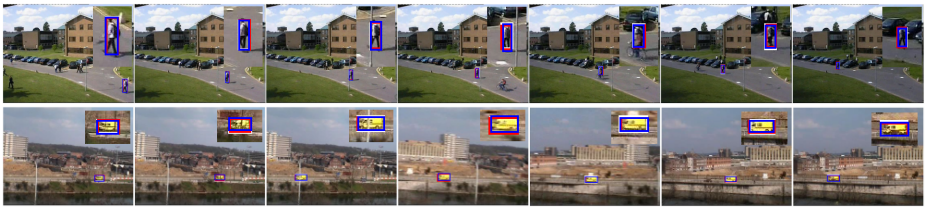


Fig. 3. Results of tracking a pedestrian in one of the PETS sequences (top) and a vehicle in a traffic sequence (bottom).

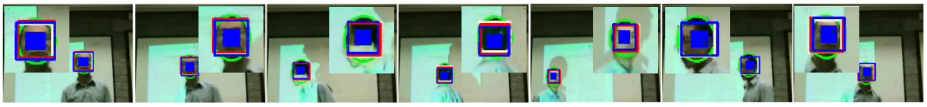


Fig. 4. Results of face tracking. The person's face rotates and undergoes dramatic appearance changes due to the lighting of a projector. Four cues are integrated in the order of color, edge, motion and contour.

second due to the motion of the camera. Spatiograms were used to implement the color cue. Figures 4 and 5 show the tracking results in the two sequences for one particular cue ordering. Similar results were obtained by varying the orders. Figure 6 shows the differences between the estimated target centers in the color cue under different orderings for the first sequence. Figure 7 shows the instantaneous particle approximations to the target distributions in different cues under different orderings for the second sequence. It can be seen that empirically, reordering the cues causes little difference in the tracking results. Nevertheless, setting the contour cue at one end of the chain makes the implementation of the algorithm clearer as x_t^{contour} contains the extra parameter θ_t and this information cannot be provided by other cues.

In the fourth experiment, we compared our approach to two typical particle-filter based multi-cue integration methods. The first selected method assumes that the cues are conditionally independent and fuses them by computing the product of the likelihoods in different cues [9, 13, 14]. We refer to it as *indepen-*

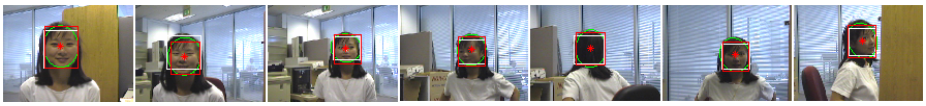


Fig. 5. Results of face tracking. The lady's face rotates, tilts and scales. Due to the motion of the camera, only three cues are integrated in the order of color, edge and contour.

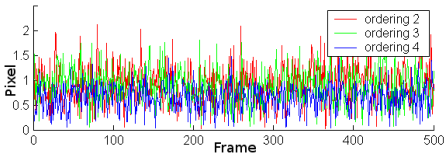


Fig. 6. The differences between the estimated target centers in the color cue under different orderings for the sequence shown in Fig. 4. Four different orderings were tested: (1) color, edge, motion and contour; (2) motion, color, contour and edge; (3) edge, contour, color and motion; (4) color, edge, contour and motion. The pixel distances between the target centers under the last three orderings and under the first ordering are computed respectively and are plotted in different colors.

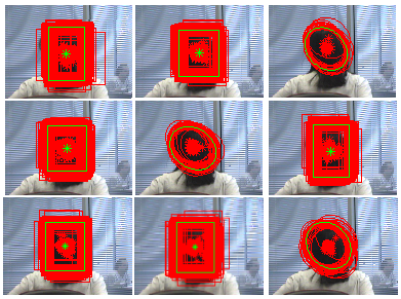


Fig. 7. The instantaneous particle approximations to the target distributions in different cues for the sequence shown in Fig. 5. Results of different orderings are shown in different rows. From left to right, the ordering for the top row is edge, color and contour, for the middle row, color, contour and edge, and for the bottom row, color, edge and contour.

dent integration. This independence assumption is often violated, making the trackers unstable. The second selected method propagates target distributions from cue to cue in a fixed order [5]. We refer to it as *hierarchical integration*. Here, the problem is that background clutter occurs in different cues at different times, rendering the design of a fixed integration order difficult. An inappropriate order often causes the propagation of wrong information from cue to cue. In the sequence shown in Fig. 8, both of these classical methods failed. *Independent integration* was thrown off by conflicts between color and contour cues. For *hierarchical integration*, all shown cue orderings failed because misleading evidence was propagated downstream at some point. Moreover, no single cue was able to track the target by itself, except for the contour cue, which was inaccurate and brittle. In contrast, our approach exploited the dependencies between the cues and enabled bidirectional information propagation between the cues, leading to increased reliability. During system design, all cues were treated equally; no hierarchy needed to be imposed.

In the last experiment, we compared the precision and the efficiency of the SAPBP algorithm with the full-product NBP, shown in Fig. 9 and Fig. 10.

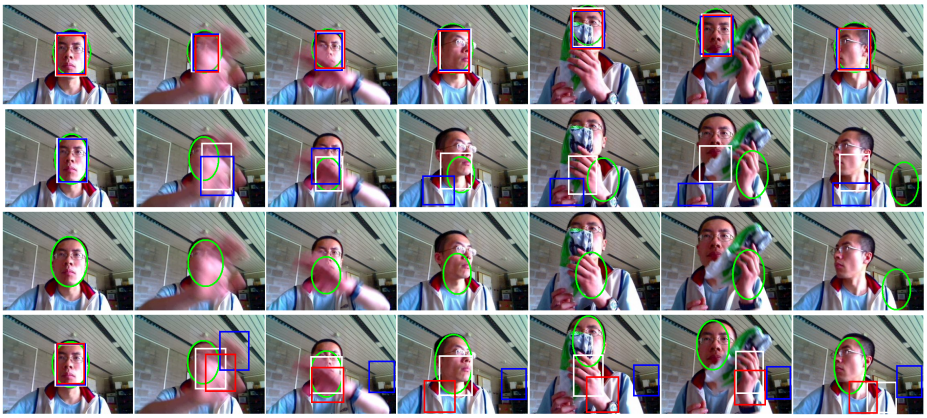


Fig. 8. Comparison of our approach with alternative methods. **First row:** Results by our approach using cues in the order of color (spatiogram), edge, motion and contour. **Second row:** *hierarchical integration* under three different orderings: (1) color→edge→motion→contour (green); (2) contour→edge→motion→color (white); (3) color→contour→edge→motion (blue). Only the estimates of the last cues in the hierarchies under the different orderings are shown. **Third row:** *independent integration*; **Fourth row:** four single-cue trackers. Note that the contour tracker by itself was able to follow the target but with much less accuracy compared to our approach.

It can be seen that both methods produced almost identical results, but our approach runs on average about 15 times faster than the full-product NBP. As the number of particles increases, more efficiency will be gained as the complexity of our approach increases linearly while that of the full-product NBP increases cubically. A comparison of different topologies of cue connections including the chain, the ring and the fully-connected models was also performed, shown in Figure 10. As the latter two models contain loops, a loopy version of our SAPBP algorithm was implemented that iterates 5 times in each step. On the ring and fully-connected models, we obtained results essentially identical to on our chain model but on average 5.2 and 7.4 times slower, respectively.

6 Conclusions

This paper presents a systematic approach to integrating multiple cues in visual tracking. The strength and beauty of the approach lies in its unprejudiced treatment of each individual cue, which permits efficient inference based on linked HMMs and a Sequential Auxiliary Particle Belief Propagation algorithm. The simultaneous cues are arranged in a simple chain topology. Empirically, the ordering of the cues along the chain is inconsequential. In a sense, each cue is a black box and only exchanges messages with its neighbors. It doesn't have to know what other cues are being used and how they are implemented. Therefore, the general architecture allows the easy combination of an arbitrary number of

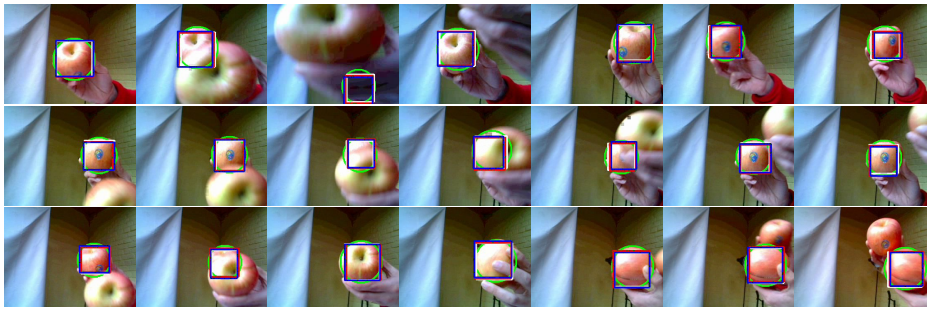


Fig. 9. Results on apple tracking. In the first row, the tracker was disturbed by a severe occlusion but recovered afterwards thanks to the color cue. In the second row, the tracker successfully handled a short occlusion by a distractor apple. When the occlusion lasted long and the scale of the distractor matched that of the target, the tracker accidentally jumped to the distractor apple, shown in the last row.

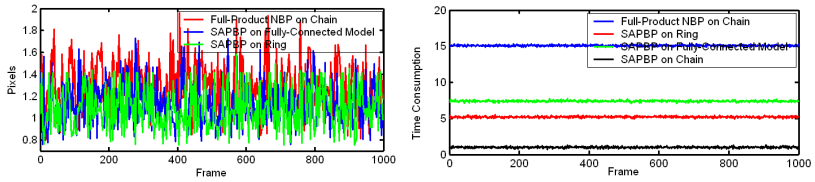


Fig. 10. The comparison of our approach, SAPBP on the chain, with the full-product NBP on the chain, SAPBP on the ring and on the fully-connected model. The left-hand figure plots the distances between the estimated target centers in the color cue by our approach and by the three other methods. As the maximum distance is no larger than 2 pixels, we conclude that they produced identical results. The right-hand figure plots the time consumptions of all the methods in each frame of the sequence.

cues in an arbitrary ordering. Our experiments confirmed a robustness superior to two competing approaches.

In this work, we deliberately chose simple cues and selectively combined them to highlight the flexibility of our approach. Other cues such as keypoints or even audio can easily be incorporated without any changes to the integration framework. The only issue is to define the inter-cue pairwise potentials, which depend solely on the target representations in the cue and its neighbors.

References

1. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conference on Computer Vision. Volume 1., Copenhagen, Denmark (2002) 661–675
2. Taylor, G., Kleeman, L.: Fusion of multimodal visual cues for model-based object tracking. In: acra. (2003)

3. Özuysal, M., Lepetit, V., Fleuret, F., Fua, P.: Feature harvesting for tracking-by-detection. In: ECCV. (2006)
4. Yang, C., Duraiswami, R., Davis, L.: Fast multiple object tracking via a hierarchical particle filter. In: ICCV, Beijing, China (2005)
5. Pérez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proceedings of the IEEE **92**(3) (2004) 495–513
6. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision **29**(2) (1998) 5–28
7. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: CVPR, Santa Barbara, CA (1998) 232–237
8. Triesch, J., Malsburg, C.v.: Self-organized integration of adaptive visual cues for face tracking. In: the Fourth IEEE International Conference on Automatic Face and Gesture Recognition. (2000)
9. Giebell, J., Gavrila1, D., Schnörr, C.: A bayesian framework for multi-cue 3d object tracking. In: ECCV, Prague, Czech Republic (2004)
10. Leichter, I., Lindenbaum, M., Rivlin, E.: A general framework for combining visual trackers: The black boxes approach. IJCV **67**(3) (2006) 343–363
11. Wu, Y., Huang, T.: Robust visual tracking by integrating multiple cues based on co-inference learning. International Journal of Computer Vision **58**(1) (2004) 55–71
12. Brand, M.: Coupled hidden Markov models for modeling interacting processes. Technical report, MIT Media Lab Perceptual Computing (1997)
13. Brasnett, P., Mihaylova, L., Canagarajah, N., Bull, D.: Particle filtering with multiple cues for object tracking in video sequences. SPIE-Image and Video Communications and Proceeding **5685** (2005) 430–441
14. Wang, H., Suter, D.: Efficient visual tracking by probabilistic fusion of multiple cues. In: ICPR, HongKong (2006)
15. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. In: MVA. Volume 14. (2003) 50–58
16. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. International Journal of Computer Vision (2007)
17. Sudderth, E., Ihler, A., Freeman, W., Willsky, A.: Nonparametric belief propagation. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., Madison, WI (2003) 605–612
18. Isard, M.: Pampas: Real-valued graphical models for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., Madison, WI (2003) 613–620
19. Hua, G., Wu, Y.: Multi-scale visual tracking by sequential belief propagation. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., Washington, DC (2004) 826–833
20. Briers, M., Doucet, A., Singh, S.: Sequential auxiliary particle belief propagation. In: the Eighth International Conference on Information Fusion. (2005)
21. Sun, J., Zheng, N., Harry, S.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(7) (2003) 787–800
22. Birchfield, S.T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA (2005)
23. Porkili, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: CVPR, San Diego, CA (2005)
24. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: ECCV. (2006) 606–618