Ensemble-Based Tracking: Aggregating Crowdsourced Structured Time Series Data

Naiyan Wang WINSTY@GMAIL.COM
Dit-Yan Yeung DYYEUNG@CSE.UST.HK
Department of Computer Science and Engineering, Hong Kong Univeristy of Science and Technology
Clear Water Bay, Hong Kong

Abstract

We study the problem of aggregating the contributions of multiple contributors in a crowdsourcing setting. The data involved is in a form not typically considered in most crowdsourcing tasks, in that the data is structured and has a temporal dimension. In particular, we study the visual tracking problem in which the unknown data to be estimated is in the form of a sequence of bounding boxes representing the trajectory of the target object being tracked. We propose a factorial hidden Markov model (FHMM) for ensemble-based tracking by learning jointly the unknown trajectory of the target and the reliability of each tracker in the ensemble. For efficient online inference of the FHMM, we devise a conditional particle filter algorithm by exploiting the structure of the joint posterior distribution of the hidden variables. Using the largest open benchmark for visual tracking, we empirically compare two ensemble methods constructed from five state-of-the-art trackers with the individual trackers. The promising experimental results provide empirical evidence for our ensemble approach to "get the best of all worlds".

1. Introduction

Visual tracking is a fundamental problem in video semantic analysis. Although it is not a new research problem in computer vision, the challenging requirements of many new applications such as terrorist detection, self-driving cars and wearable computers require that some objects of interest possibly with fast and abrupt motion in uncontrolled environments be tracked as they move around in a video. This has led to a resurgence of interest in visual tracking within the machine learning and computer vision communities. In this paper, we consider the most common setting, called single object tracking problem, in which only one object of interest is tracked at a time given its location in the first frame of the video.

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

Many visual tracking methods have been proposed over the last decade. Although significant progress has been made, even state-of-the-art trackers today are still far from perfect and no single method is always the best under all situations. This is due to each tracker's model bias, corresponding to the assumptions made about the object being tracked and the environment, which is valid in some situations but invalid in others. Consequently, different trackers exhibit different advantages. For example, local patch based methods (Adam et al., 2006; Jia et al., 2012) are more robust against occlusion and deformation, while whole template based methods (Ross et al., 2008; Bao et al., 2012; Wang et al., 2013a;b) can track rigid objects better. Also, trackers based on the generative approach (Ross et al., 2008; Bao et al., 2012; Wang et al., 2013a) are generally more accurate when the objects do not vary too much, while those based on the discriminative approach (Grabner et al., 2006; Babenko et al., 2011; Hare et al., 2011; Wang & Yeung, 2013) making explicit use of negative samples from the background perform better when the background is cluttered or contains distractors. Although some recent methods adopt a hybrid approach (Zhong et al., 2012; Kwon & Lee, 2013), they still cannot give satisfactory results at all times. Furthermore, most trackers are sensitive to the parameter setting but it is practically impossible to tune the parameters separately for each video. As a result, developing a tracker that is stable enough for general use in a wide range of application scenarios remains an open and challenging research problem.

The approach we adopt in this paper has been inspired by some recent machine learning methods developed for the crowdsourcing setting, e.g., (Dekel & Shamir, 2009; Raykar et al., 2010; Bachrach et al., 2012). It takes advantage of the wisdom of the crowd by dispatching a problem to multiple imperfect contributors and then aggregating the solutions from them to give a combined solution of higher quality than any individual solution. For our visual tracking problem, each tracker plays the role of an imperfect contributor which solves the tracking problem to obtain its tracking result independently. We refer to our approach as *ensemble-based tracking* (EBT) due to its similarity with ensemble methods which combine multiple models. We also note that there exist trackers which use ensemble-based classification for tracking, e.g., (Avidan, 2007; Bai et al., 2013). Despite the similarities, it is worth to note some significant differences between EBT and previous work in crowdsourcing and ensemble-based classification for tracking. These two aspects will be elaborated separately below.

There exist two crucial differences between EBT and previous work in crowdsourcing. Although different machine learning issues have been studied in the crowdsourcing setting (e.g., active learning (Yan et al., 2011), spamming (Raykar & Yu, 2012), task assignment (Ho et al., 2013), and multitask learning (Mo et al., 2013)), the learning tasks are limited to classification and regression formulated in relatively simple ways. On the other hand, visual tracking is more of a structured prediction problem in which the tracking result is in a form of structured data with attributes including the location, scale and aspect ratio of the object being tracked. It has been demonstrated by (Hare et al., 2011) that exploiting the structured data properties can improve the tracking performance significantly. Moreover, the tracking problem has a temporal dimension which is not present in the classification and regression tasks studied by the previous work.

As for previous work on applying ensemble-based classification for tracking, those methods (Grabner et al., 2006; Avidan, 2007; Babenko et al., 2011; Bai et al., 2013) are mostly discriminative trackers which treat tracking as a binary classification or multiple-instance learning problem. There is only one tracker which is a binary classifier realized by an ensemble method, such as AdaBoost, to combine the classification results of multiple weak classifiers. Unlike these methods, EBT is not a single method but an approach or framework for combining multiple trackers, each of which may be based on a generative, discriminative or hybrid approach. It is this very nature of EBT that we intend to exploit to “get the best of all worlds” without being restricted to just a single type of model bias that underlies only one tracker.

The contributions of this paper may be summarized by the following three points:

1. Inspired by the *factorial hidden Markov model* (FHMM) (Ghahramani & Jordan, 1997) which generalizes ordinary HMM to a distributed state representation, we propose a state-space model for aggregating the structured time series data contributed by multiple trackers in a crowdsourcing setting.
2. Because of the structured data in the EBT model, no simple analytical form exists for the posterior distribution. We devise an efficient conditional particle filter algorithm for online inference of the hidden variables which are potentially of high dimensionality.
3. We empirically compare some state-of-the-art trackers with a realization of our EBT approach on a benchmark (Wu et al., 2013) which is currently the largest open benchmark for visual tracking. The EBT approach leads to performance improvement that beats any single tracker by a considerable margin.

2. Background

We first review the Bayesian model and a sequential inference algorithm for the visual tracking problem.

2.1. Bayesian Model for Visual Tracking

In what follows, let $\mathbf{I}^f$ denote the observed measurement in video frame f (i.e., the raw pixels or extracted features) and $\mathbf{B}^f$ the hidden state of the target object (i.e., the bounding box). For visual tracking, we usually characterize a bounding box by six affine parameters: horizontal translation, vertical translation, scale, aspect ratio, rotation and skewness.[1] The goal of visual tracking is to estimate the following posterior distribution recursively:

$$p(\mathbf{B}^f \mid \mathbf{I}^{1:f}) \propto p(\mathbf{I}^f \mid \mathbf{B}^f)\cdot \int p(\mathbf{B}^f \mid \mathbf{B}^{f-1})\, p(\mathbf{B}^{f-1} \mid \mathbf{I}^{1:f-1})\, d\mathbf{B}^{f-1}, \tag{1}$$

where the observation likelihood $p(\mathbf{I}^f \mid \mathbf{B}^f)$ defines the probability density function of observing the measurement $\mathbf{I}^f$ given the hidden state $\mathbf{B}^f$, and the term $p(\mathbf{B}^f \mid \mathbf{B}^{f-1})$ models the transition probability between the unknown states in consecutive frames reflecting the fact that the two states are not too far away. The posterior mode is usually used as the tracking result for frame f.

2.2. The Particle Filter

The particle filter is a sequential Monte Carlo algorithm for online Bayesian inference. It approximates the posterior distribution of frame f by a set of weighted particles $\{(w^f_{(i)}, \mathbf{B}^f_{(i)})\}_{i=1}^N$. Its advantage over the commonly used Kalman filter is that it is not limited to the Gaussian distribution, not even to any parametric distribution. We review below a simplified but popular version of particle filter called the bootstrap filter. The bootstrap filter first approx-

[1]Some previous work used only the first two parameters while we use the first four here. As in the benchmark, we set the last two parameters (rotation and skewness) to zero.

imates $p(\mathbf{B}^f \mid \mathbf{I}^{1:f})$ as:

$$p(\mathbf{B}^f \mid \mathbf{I}^{1:f}) \approx c\, p(\mathbf{I}^f \mid \mathbf{B}^f) \sum_{i=1}^{N} w_{(i)}^{f-1}\, p(\mathbf{B}^f \mid \mathbf{B}_{(i)}^{f-1}), \tag{2}$$

where c is a constant that does not depend on $\mathbf{B}^f$. We first draw N samples $\mathbf{B}_{(i)}^f$ in frame f from the following proposal distribution:

$$\mathbf{B}_{(i)}^f \sim \sum_{i=1}^{N} w_{(i)}^{f-1}\, p(\mathbf{B}^f \mid \mathbf{B}_{(i)}^{f-1}). \tag{3}$$

We then set their corresponding weights as

$$w_{(i)}^f = p(\mathbf{I}^f \mid \mathbf{B}_{(i)}^f). \tag{4}$$

The particle $\mathbf{B}_{(i)}^f$ with the largest weight $w_{(i)}^f$ is then chosen as the location of the target for frame f. This procedure is repeated for all frames in the video.

3. Model Formulation

In this section, we define an FHMM for our EBT approach. In addition to estimating the hidden state sequence which corresponds to the ground-truth trajectory of the target object, we also want to estimate the reliability of each tracker in the ensemble for reasons that will become clearer later.

Suppose the video has F frames and there are T trackers in total. In frame f, let $\mathbf{y}^f$ denote the unknown bounding box of the object and $\mathbf{z}_t^f$ and r_t^f denote the output bounding box and reliability of the tth tracker. Note that our method only interacts with the output of each individual tracker but does not involve the input video $\mathbf{I}^f$ directly. For notational convenience in the sequel, we put the vectors $\mathbf{z}_t^f$ for all trackers into a matrix $\mathbf{Z}^f$ and the variables r_t^f for all trackers into a vector $\mathbf{r}^f$. The joint probability of the FHMM is given by

$$\prod_{f=2}^{F} \prod_{t=1}^{T} p(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) p(\mathbf{y}^f \mid \mathbf{y}^{f-1}) p(r_t^f \mid r_t^{f-1}) p(r_t^f \mid a). \tag{5}$$

In the first frame, $\mathbf{y}^1$ is given by the annotation and hence known. We also initialize each r_t^1 to 1. The first term in the above expression is the observation likelihood for each tracker. The next two terms model the continuity of the location of the object and the continuity of the reliability of each tracker along the temporal dimension. The last term gives a frame-independent and tracker-independent prior of the reliability, where a is a hyperparameter which controls the degree of regularization. Fig. 1 depicts the graphical model of the FHMM with the plate notation. We will elaborate each term of the joint probability below.

Observation Likelihood: As discussed above, a notable difference between our model and previous work in crowdsourcing is that our model has to deal with structured time series data for representing the object trajectory in the form of a sequence of bounding boxes. Thus the observation likelihood has to be designed carefully. We formulate it in terms of two factors, $p_c(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f)$ and $p_o(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f)$, which correspond to two common performance metrics for visual tracking, namely, the normalized central-pixel error metric $\mathcal{D}(\cdot,\cdot)$ and the overlap rate metric $\mathcal{O}(\cdot,\cdot)$, respectively. In each frame, $\mathcal{D}(\cdot,\cdot)$ measures the Euclidean distance between the center of the ground-truth bounding box and that of the predicted bounding box, with the horizontal and vertical differences normalized by the width and height, respectively, of the predicted bounding box in the previous frame. The normalization step makes the result more reasonable when the bounding box is far from a square. $\mathcal{O}(\cdot,\cdot)$ is the ratio of the area of intersection of the two bounding boxes to the area of their union. Fig. 2 shows some examples to illustrate the necessity for using both factors to define the observation likelihood. Moreover, when the two bounding boxes do not overlap (i.e., the overlap rate is equal to zero), we need to count on the central-pixel error metric to estimate the chance of recovery.

Mathematically, we have

$$p(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) = p_c(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f)\, p_o(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f), \tag{6}$$

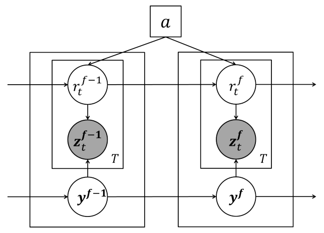


Figure 1. Graphical model with the plate notation representing the FHMM for EBT.

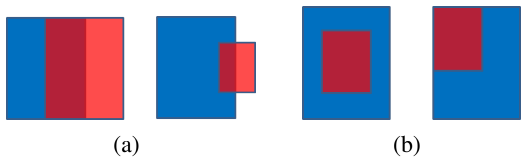


Figure 2. Illustration of the importance of using both performance metrics. The ground-truth bounding box is colored in blue and the predicted one is in red. (a) Although both cases have the same central-pixel error, the predicted bounding boxes have very different scales and hence their overlap rates are also very different. (b) The two cases have the same overlap rate but different central-pixel errors. The left one likely covers more of the salient region than the right one.

where

$$\begin{aligned} p_c(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) &\sim \mathcal{N}(\alpha \mathcal{D}(\mathbf{z}_t^f, \mathbf{y}^f) \mid 0, r_t^f) \\ p_o(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) &\sim \text{TrunExp}(1 - \mathcal{O}(\mathbf{z}_t^f, \mathbf{y}^f) \mid r_t^f), \end{aligned} \tag{7}$$

where α is a parameter which balances the two metrics. We use the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, with mean μ and variance σ^2, to define the factor corresponding to the normalized central-pixel error. Since the overlap rate is in the range $[0, 1]$, we use the truncated exponential distribution in $[0, 1]$ to define the factor corresponding to the overlap rate. Concretely, the probability density function (pdf) is defined as:

$$\text{TrunExp}(x \mid \lambda) = \begin{cases} \frac{\lambda}{Z} \exp(-\lambda x) & x \in [0, 1] \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where $Z = 1 - \exp(-\lambda)$ is a normalization constant. We note that the pdf decreases monotonically in $[0, 1]$ and is 0 elsewhere.

Transition Probability Distributions: For the state transition probability distribution $p(\mathbf{y}^f \mid \mathbf{y}^{f-1})$, we simply use a zero-mean Gaussian distribution independently for each dimension y_d^f:

$$p(y_d^f \mid y_d^{f-1}) \sim \mathcal{N}(y_d^f \mid y_d^{f-1}, \sigma_d^2). \tag{9}$$

For the reliability transition probability distribution $p(r_t^f \mid r_t^{f-1})$, our choice of the distribution form needs to take into consideration that the reliability must be nonnegative. We use a Gamma distribution to model it:

$$p(r_t^f \mid r_t^{f-1}) \sim \mathcal{G}\left(r_t^f \mid k, \frac{r_t^{f-1}}{k}\right), \tag{10}$$

where k is a model parameter. The choice is deliberate since $\mathbb{E}[r_t^f] = r_t^{f-1}$.

Reliability Prior: The purpose of this prior is to prevent the algorithm from assigning a very high weight to one particular tracker and hence overfitting the tracker.[2] To accomplish this, we simply use a time-invariant exponential distribution to penalize high reliability:

$$p(r_t^f \mid a) \sim \text{Exp}(r_t^f \mid a). \tag{11}$$

4. Model Inference

Due to the nature of the visual tracking problem, we are interested in a sequential inference algorithm for our model. We resort to a conditional particle filter algorithm due to its flexibility as well as its ability to deal with the possibly high dimensionality of the hidden variables.

[2]Nevertheless, this may not be ideal if most of the trackers fail and some happen to give very similar results.

We are interested in the posterior distribution of the unknown ground truth $\mathbf{y}^f$ given the full history of the observations $\mathbf{Z}^{1:f}$, i.e., $p(\mathbf{y}^f \mid \mathbf{Z}^{1:f})$. We expand it recursively as follows:

$$\begin{aligned} & p(\mathbf{y}^f \mid \mathbf{Z}^{1:f}) \\ &= \int p(\mathbf{y}^f, \mathbf{r}^f \mid \mathbf{Z}^{1:f})\, d\mathbf{r}^f \\ &\propto \int p(\mathbf{Z}^f \mid \mathbf{y}^f, \mathbf{r}^f)\, p(\mathbf{y}^f, \mathbf{r}^f \mid \mathbf{Z}^{1:f-1})\, d\mathbf{r}^f \\ &= \int p(\mathbf{Z}^f \mid \mathbf{y}^f, \mathbf{r}^f) \int\int p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1}) \cdot \\ & \quad p(\mathbf{y}^f \mid \mathbf{y}^{f-1}) p(\mathbf{r}^f \mid \mathbf{r}^{f-1}) p(\mathbf{r}^f \mid a)\, d\mathbf{r}^{f-1} d\mathbf{y}^{f-1} d\mathbf{r}^f. \end{aligned} \tag{12}$$

Since the analytical form of the posterior is intractable, we approximate the distribution $p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1})$ by a set of weighted particles according to the particle filter approach. However, a direct particle filter approach does not work well since the dimensionality of the hidden variables increases with the number of trackers and hence the number of particles needed has to increase exponentially in order to give satisfactory result. Fortunately, the problem is well structured so that the joint distribution can be decomposed as follows:

$$\begin{aligned} & p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1}) \\ &= p(\mathbf{y}^{f-1} \mid \mathbf{Z}^{1:f-1}) \prod_{t=1}^{T} p(r_t^{f-1} \mid \mathbf{y}^{f-1}, \mathbf{z}_t^{1:f-1}). \end{aligned} \tag{13}$$

This observation is illuminating since it allows us to approximate the distribution by a set of *conditional* weighted particles $\left\{ \left(w_{(n)}^{f-1}, \mathbf{y}_{(n)}^{f-1}, \pi_{t,(m,n)}^{f-1}, r_{t,(m,n)}^{f-1} \right) \right\}$ for $t = 1, \ldots, T$, $m = 1, \ldots, M$, $n = 1, \ldots, N$, where M, N denote the numbers of particles for r_t^{f-1} and $\mathbf{y}^{f-1}$, respectively. That is, the particles for reliability are conditional on the particles for the unknown ground-truth bounding box. Formally, we have:

$$\begin{aligned} & p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1}) \\ &\approx \sum_{n=1}^{N} w_{(n)}^{f-1} \delta(\mathbf{y}^{f-1} - \mathbf{y}_{(n)}^{f-1}) \prod_{t=1}^{T} p(r_t^{f-1} \mid \mathbf{y}_{(n)}^{f-1}, \mathbf{z}_t^{1:f-1}) \\ &\approx \sum_{n=1}^{N} w_{(n)}^{f-1} \prod_{t=1}^{T} \sum_{m=1}^{M} \pi_{t,(m,n)}^{f-1} \delta(\mathbf{y}^{f-1} - \mathbf{y}_{(n)}^{f-1}) \delta(r_t^{f-1} - r_{t,(m,n)}^{f-1}), \end{aligned} \tag{14}$$

where $\delta(\cdot)$ is the Dirac delta function. Substituting Eqn. 14

Algorithm 1 Conditional Particle Filter Algorithm

```
Initialize the particle set $\left\{\left(w^0_{(n)}, \mathbf{y}^0_{(n)}, \pi^0_{t,(m,n)}, r^0_{t,(m,n)}\right)\right\}$
for each frame $f$ do
    for $i = 1, 2, \dots, N$ do
        Select one particle $n$ according to $w^{f-1}_{(n)}$
        Sample new particle $\mathbf{y}^f_{(n)} \sim p(\mathbf{y}^f_{(n)} \mid \mathbf{y}^{f-1}_{(n)})$
        for $t = 1, 2, \dots, T$ do
            for $j = 1, 2, \dots, M$ do
                Select one particle $(t, m)$ according to $\pi^{f-1}_{t,(m,n)}$
                Sample new particle $r^f_{t,(m,n)} \sim p(r^f_{t,(m,n)} \mid r^{f-1}_{t,(m,n)})$
                Evaluate new weight $\pi^f_{t,(m,n)}$ using Eqn. 16
            end for
        end for
        Set the new weight $w^f_n = \prod_{t=1}^T \sum_{i=1}^M \pi^f_{t,(m,n)}$
        Normalization: $w^f_n = w^f_n / \sum_{i=1}^N w^f_{(i)}$
        Normalization: $\pi^f_{t,(m,n)} = \pi^f_{t,(m,n)} / \sum_{i=1}^M \pi^f_{t,(i,n)}$
        Check if any tracker has failed (refer to Sec. 5.1)
    end for
end for
```

into Eqn. 12 yields:

$$\begin{aligned}
&p(\mathbf{y}^f \mid \mathbf{Z}^{1:f}) \\
&\propto \int p(\mathbf{Z}^f \mid \mathbf{y}^f, \mathbf{r}^f) \int\int p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1}) \cdot \\
&\quad p(\mathbf{y}^f \mid \mathbf{y}^{f-1}) p(\mathbf{r}^f \mid \mathbf{r}^{f-1}) p(\mathbf{r}^f \mid a)\, d\mathbf{r}^{f-1} d\mathbf{y}^{f-1} d\mathbf{r}^f \\
&\approx \int p(\mathbf{Z}^f \mid \mathbf{y}^f, \mathbf{r}^f) \sum_{n=1}^N w^{f-1}_{(n)} \prod_{t=1}^T \sum_{m=1}^M \pi^{f-1}_{t,(m,n)} p(\mathbf{y}^f \mid \mathbf{y}^{f-1}_{(n)}) \cdot \\
&\quad p(r^f_t \mid r^{f-1}_{t,(m,n)}) p(r^f_t \mid a) d\mathbf{r}^f \\
&= \int \sum_{n=1}^N w^{f-1}_{(n)} p(\mathbf{y}^f \mid \mathbf{y}^{f-1}_{(n)}) \cdot \\
&\quad \prod_{t=1}^T \sum_{m=1}^M \pi^{f-1}_{t,(m,n)} p(r^f_t \mid r^{f-1}_{t,(m,n)}) p(\mathbf{z}^f_t \mid \mathbf{y}^f, r^f_t) p(r^f_t \mid a)\, d\mathbf{r}^f.
\end{aligned} \tag{15}$$

The above formula implies an efficient particle filter algorithm for inferring the posterior. For each particle, the weight is set as:

$$\pi^f_{t,(m,n)} = p(\mathbf{z}^f_t \mid \mathbf{y}^f_{(n)}, r^f_{t,(m,n)}) p(r^f_{t,(m,n)} \mid a). \tag{16}$$

We summarize our proposed conditional particle filter algorithm for EBT in Alg. 1.

5. Implementation Details

In this section, we provide some implementation details of the proposed algorithm, its time complexity, and the parameter setting used in the experiments which will be reported in the next section.

5.1. Failure Detection

If the predicted location of a tracker stays far from the true object location for an extended period of time, it would help to exclude the result of this tracker from the ensemble. Not only can this lead to speedup, but, more importantly, the failed tracker may adversely impair the performance of the ensemble. The question then is how to detect such failure effectively. This is where the reliability variable r^f_t comes into play. Specifically, we monitor the expectation of the marginalized posterior of r^f_t. When it falls below a threshold θ for p successive frames, we will mark it as a failed tracker.

5.2. Self-Correction

Unlike in typical crowdsourcing tasks, the nature of visual tracking makes it quite unlikely for a failed tracker to recover by itself. In view of this characteristic, we introduce a novel feature into our EBT approach. Whenever a failed tracker is detected, it will be sent the current result of the ensemble to initiate a restart, or self-correction, in the tracker. Doing so allows us to fully utilize all the trackers in the ensemble. However, to support this feature, two-way communication is needed between the individual trackers and the ensemble algorithm. This poses some technical challenges when the trackers are implemented in different programming languages. We have developed a web service to provide a generic interface through which different trackers possibly implemented in different languages and running on different operating systems can communicate easily with the ensemble algorithm. This convenient platform allows us to incorporate essentially any tracker into the ensemble. For the sake of referencing, we refer to this variant of our algorithm with self-correction as SC-EBT.

5.3. Time Complexity

The time complexity of the proposed algorithm is $O(MNFT)$, which is linear with respect to the number of particles used. This allows us to make a tradeoff between quality and speed, in that increasing the number of particles in the particle filter typically improves the approximation quality at the expense of computation time.

5.4. Parameter Setting

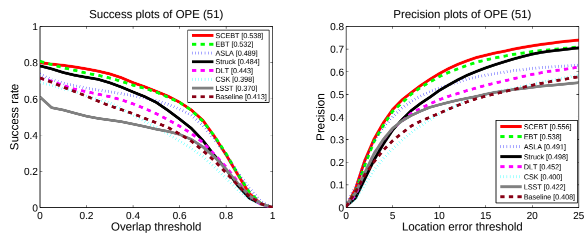
We set $M = 50$, $N = 400$ for the particle filter, $k = 0.1$, $\alpha = 2$, $a = 0.1$ for the model. For failure detection, we set $p = 10$ and $\theta = 0.8^T$ and 0.9^T in EBT and SC-EBT, respectively. For the Gaussian transition probability distributions for horizontal translation, vertical translation, scale and aspect ratio, their standard deviations are 4, 4, 0.01, and 0.001, respectively. Unlike some practices in the visual tracking literature which tune the parameters for each video sequence to get the best result, here *we fix the values of all these parameters for all the 51 video sequences tested*. Using this parameter setting, our EBT and SC-EBT algorithms run at about 1fps (frame per second) if including the running time of the individual trackers, and about 5fps if excluded.

6. Experiments

To facilitate objective comparison, we use a recently released benchmark (Wu et al., 2013) in our experiments. It is currently the largest open benchmark for visual tracking, which comprises 51 video sequences covering 11 challenging aspects of visual tracking. To choose the trackers for inclusion in the ensemble, we mainly take two criteria into consideration. First, we tend to choose the best performers which perform well in different categories so that they can complement each other. Second, since the running speed of the proposed algorithms is determined by the slowest tracker, we only consider trackers which can run at 5fps or above. As a result, five trackers are included in the ensemble: one local patch based method (ASLA) (Jia et al., 2012), one based on structured output kernel SVM (Struck) (Hare et al., 2011), one based on deep learning (DLT) (Wang & Yeung, 2013), one based on correlation filter (CSK) (Joao et al., 2012), and one based on robust template learning (LSST) (Wang et al., 2013a). We also include a simple baseline ensemble method which reports the mean of the bounding boxes reported by the individual trackers. For fair comparison with the results reported in (Wu et al., 2013), we fix all the parameters of the trackers included. The implementation of EBT and SC-EBT can be found on the project page: `http://winsty.net/ebt.html`.

6.1. Quantitative Results

For quantitative comparison, we use two performance measures analogous to the *area under curve* (AUC) measure for the *receiver operating characteristic* (ROC) curve. Specifically, for a given overlap threshold in $[0, 1]$, a tracking method is considered successful in a frame if its overlap rate exceeds the threshold. The success rate measures the percentage of successful frames over an entire video. By varying the threshold gradually from 0 to 1, it gives a plot of the success rate against the overlap threshold for each tracking method. A similar performance measure is defined for the central-pixel error. Both measures give very similar results. Note that when a tracker fails, its result may become quite unstable. Thus, for making meaningful comparison, we threshold the horizontal axis to $[0, 25]$ instead of $[0, 50]$ in the original benchmark. So the results of central-pixel error cannot be compared directly to the ones in (Wu et al., 2013) directly. Due to space constraints, the results for each video category are left to the supplemental material.



(a) Overlap rate (OR) (b) Central-pixel error (CPE)

Figure 3. ROC curves based on the overlap rate and central-pixel error metrics for all the test sequences.

Fig. 3 shows the results in ROC curves based on these two metrics. Not surprisingly, both EBT and SC-EBT outperform all the individual trackers with AUC equal to 0.538 and 0.532, respectively, under the OR metric. Even when compared with SCM (Zhong et al., 2012), the best tracker which achieves an AUC of 0.499 as reported in (Wu et al., 2013), our ensemble methods are still significantly better. For the CPE metric, the advantages are even more significant. SC-EBT outperforms EBT mainly for low- to medium-ranged overlap rates. Although the ensemble algorithm cannot get the most accurate results for some difficult cases, the self-correction scheme is effective in maintaining the results in a reasonable range.

We notice that our ensemble methods are inferior to some trackers mainly under the situation when the tracked object moves fast. This issue is related to the motion models used by the individual trackers. Under fast motion, most trackers cannot correctly generate a candidate set that contains the target object, with the exception of Struck. Since four of the five trackers fail at the very beginning, it is impossible for the ensemble to get correct result. We note that SC-EBT performs significantly better than EBT, verifying the effectiveness of the self-correction mechanism. With self-correction incorporated, the problem of fast motion can be partially alleviated because it makes a sudden jump from some incorrect tracking result to a correct one possible.

We also report in Tab. 1 and Tab. 2 the average success rates at several thresholds for different methods. Under all settings, SC-EBT ranks first while EBT ranks second. Their advantages are especially significant for medium-ranged thresholds which are often used in practice. We further compare our ensemble methods with each of the individual trackers by counting the number of video sequences in which our methods win, draw, or lose. Fig. 4 shows the

	p@0.3	p@0.5	p@0.7
SC-EBT	**0.7381**	**0.6416**	**0.4819**
EBT	0.7151	0.6376	0.4767
ASLA	0.6396	0.5893	0.4545
Struck	0.6890	0.5806	0.3740
DLT	0.6122	0.5181	0.3591
CSK	0.5781	0.4426	0.2815
LSST	0.4832	0.4324	0.3349
Baseline	0.5615	0.4681	0.3144

Table 1. Success rates at different thresholds based on the overlap rate metric for different tracking methods.

	p@5	p@10	p@15
SC-EBT	**0.4333**	**0.5915**	**0.6728**
EBT	0.4258	0.5782	0.6520
ASLA	0.4027	0.5278	0.5847
Struck	0.3426	0.5172	0.6171
DLT	0.3499	0.4762	0.5400
CSK	0.2610	0.4123	0.4989
LSST	0.3503	0.4541	0.5008
Baseline	0.2953	0.4165	0.4921

Table 2. Success rates at different thresholds based on the central-pixel error metric for different tracking methods.

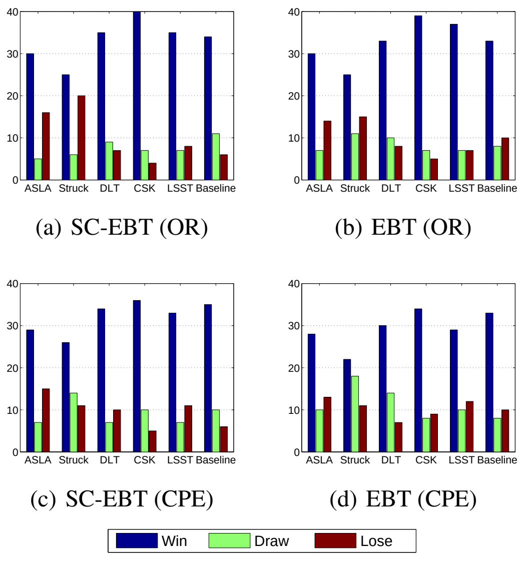


Figure 4. Comparison of SC-EBT and EBT with other trackers based on two evaluation metrics in terms of the number of sequences in which our methods win, draw, or lose.

results. To make the comparison stable and meaningful, the comparison is said to draw if the difference is less than 0.01. We believe the comparison is substantial enough to demonstrate the effectiveness of our ensemble methods.

6.2. Qualitative Results

Besides quantitative comparison, let us also try to gain a deeper understanding of the EBT approach by looking at some test sequences in which our methods succeed or fail.

The first row of Fig. 5 shows an easy example that both EBT and SC-EBT can estimate the ground-truth object location accurately. LSST and CSK fail when the walking person is occluded by a pedestrian in frame 42. Because they are the minorities among the five trackers, both EBT and SC-EBT can detect their failure and lower their reliability accordingly. Consequently, the aggregated result is accurate in tracking the target.

The second row shows a failure case. In the beginning, all five trackers can track the target correctly and hence they are all assigned high weights. Later, three of them drift away to the same cluttered background. As a result, the ensemble algorithms assign high weights to this incorrect background and hence lead to failure of the ensemble.

In the third row, we demonstrate a case in which SC-EBT can correctly track the object to the end but EBT fails. Before frame 300, the results of EBT and SC-EBT agree with each other although EBT has already eliminated the failed trackers LSST and Struck. When it is near frame 315, DLT and ASLA have high reliability but they both give incorrect votes. The only correct tracker is CSK. Thus the tracking result drifts. For SC-EBT, however, the results of LSST and Struck are corrected so that the ensemble can withstand the incorrect results of DLT and ASLA. As a result, SC-EBT can track the target accurately to the end.

In the last row, we show an example in which the self-correction mechanism actually impairs the performance. We note that this is a rare case among all the sequences tested though. Over a large portion of the video, all the trackers actually agree well to produce stable results. The difference between EBT and SC-EBT appears in about frame 400 when the lady moves fast and hence makes most of the trackers fail. However, EBT can survive because it has eliminated two failed trackers, DLT and CSK, before the fast motion and has assigned a high weight to Struck which performs well. Coincidentally, EBT can track the lady correctly even though four trackers actually drift to the same place in the background.

7. Discussion

Before closing, let us investigate the relationship between this work and two recent methods.

Briefly speaking, (Zhong et al., 2010) applies a crowdsourcing algorithm called GLAD (Whitehill et al., 2009), originally proposed for image annotation, to the visual tracking problem. Their method differs from ours in at least two important aspects. First, it still treats the ensemble

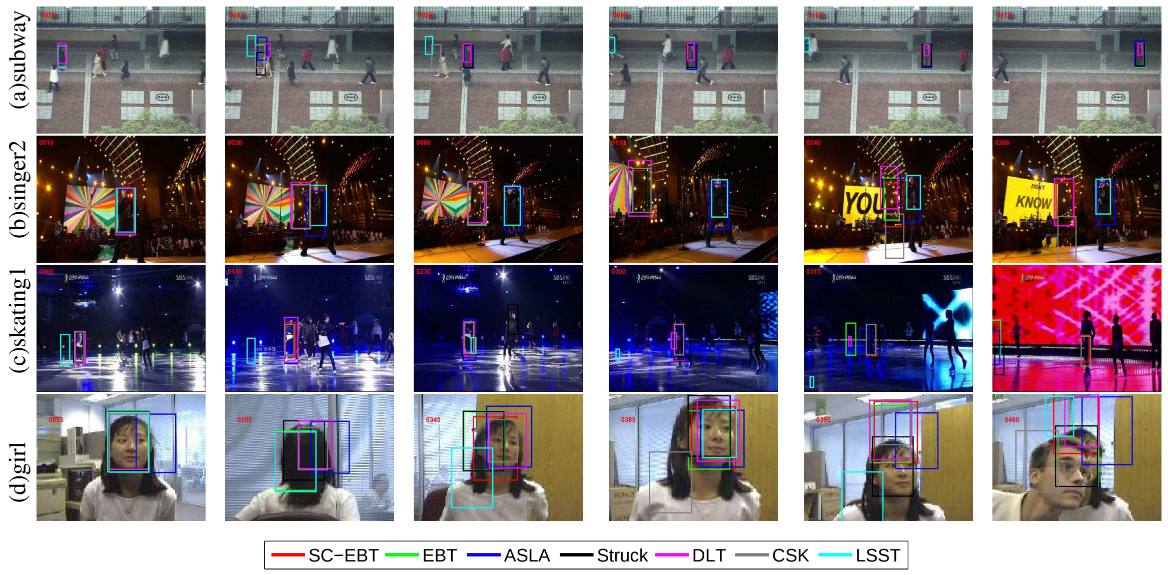


Figure 5. Results of some test sequences in the visual tracking benchmark. Details can be found in Sec. 6.2.

learning problem as a binary classification problem. As discussed earlier, ignoring the structure of the bounding box data can impair the results. Second, it totally ignores the spatial smoothness of the unknown ground truth and uses a heuristic to update the reliability of each tracker. These two issues are exactly what this work intends to address under a principled framework.

The recent work by (Kwon & Lee, 2013) shares the same motivation as ours in that it also integrates several basic and simple trackers. It decomposes the complicated tracking problem into four basic ingredients: appearance model, motion model, state representation, and observation type. For each ingredient, several candidates are available to choose from. They propose an interactive Markov chain Monte Carlo method to infer the relative importance of each candidate and then simply combine the candidates together. Despite its simplicity, it yields significant improvement over other simpler trackers in sophisticated environments. Nevertheless, the method has some obvious drawbacks. First, the candidates for each ingredient are quite simple and similar and hence do not provide sufficient diversity to form a strong ensemble. Second, its ensemble is still limited to the model level but not the output level. To the contrary, our EBT approach allows essentially any tracker to be added to the ensemble with minimal effort.

8. Conclusion and Future Work

We have proposed a framework for aggregating crowdsourced time series data in a form of structured data and have applied it to the visual tracking problem by combining the tracking results of multiple trackers. To realize this approach, we propose a novel FHMM as a state-space model to account for the continuity of both the location of the target and the reliability of each tracker. For model inference, we devise an efficient conditional particle filter algorithm for online estimation of the hidden variables which are potentially of high dimensionality. The curse of dimensionality is alleviated by exploiting the structure of the joint posterior distribution of the hidden variables. We are very excited by the promising results of our ensemble methods when they are compared with some state-of-the-art trackers on the largest open benchmark (Wu et al., 2013) to date.

Although our ensemble methods can achieve remarkable improvement over the individual trackers, they can still fail in some challenging situations which make human intervention inevitable. How to determine the minimal set of frames to seek help from human annotators is an interesting research problem. It is related to active learning but is more complicated than many well-studied active learning tasks in the machine learning community. Although some preliminary work (Vondrick & Ramanan, 2011; Vondrick et al., 2013) has been pursued along this line for video annotation, their problem setting is very different from ours. We will pursue research in this direction as our future work.

Acknowledgement

This research has been supported by General Research Fund 621310 from the Research Grants Council of Hong Kong. We thank Tiansheng Yao for the helpful discussions during the preparation of this paper.

References

Adam, A., Rivlin, E., and Shimshoni, I. Robust fragments-based tracking using the integral histogram. In *CVPR*, pp. 798–805, 2006.

Avidan, S. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

Babenko, B., Yang, M.-H., and Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

Bachrach, Y., Graepel, T., Minka, T., and Guiver, J. How to grade a test without knowing the answers—a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *ICML*, pp. 1175–1182, 2012.

Bai, Q., Wu, Z., Sclaroff, S., Betke, M., and Monnier, C. Randomized ensemble tracking. In *ICCV*, pp. 2040–2047, 2013.

Bao, C., Wu, Y., Ling, H., and Ji, H. Real time robust L1 tracker using accelerated proximal gradient approach. In *CVPR*, pp. 1830–1837, 2012.

Dekel, O. and Shamir, O. Good learners for evil teachers. In *ICML*, pp. 233–240, 2009.

Ghahramani, Z. and Jordan, I.M. Factorial hidden Markov models. *Machine Learning*, 29(2-3):245–273, 1997.

Grabner, H., Grabner, M, and Bischof, H. Real-time tracking via on-line boosting. In *BMVC*, pp. 47–56, 2006.

Hare, S., Saffari, A., and Torr, P.H.S. Struck: Structured output tracking with kernels. In *ICCV*, pp. 263–270, 2011.

Ho, C.-J., Jabbari, S., and Vaughan, J.W. Adaptive task assignment for crowdsourced classification. In *ICML*, pp. 534–542, 2013.

Jia, X., Lu, H., and Yang, M.-H. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pp. 1822–1829, 2012.

Joao, F.H., Caseiro, R., Martins, P., and Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pp. 702–715, 2012.

Kwon, J. and Lee, K. Tracking by sampling and integrating multiple trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. Accepted.

Mo, K., Zhong, E., and Yang, Q. Cross-task crowdsourcing. In *KDD*, 2013.

Raykar, V.C. and Yu, S. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012.

Raykar, V.C., Yu, S., Zhao, H.L., Valadez, G.H., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *Journal of Machine Learning Research*, 99:1297–1322, 2010.

Ross, D.A., Lim, J., Lin, R.S., and Yang, M.-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1):125–141, 2008.

Vondrick, C. and Ramanan, D. Video annotation and tracking with active learning. In *NIPS*, pp. 28–36, 2011.

Vondrick, C., Patterson, D., and Ramanan, D. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

Wang, D., Lu, H., and Yang, M.-H. Least soft-threshold squares tracking. In *CVPR*, pp. 2371–2378, 2013a.

Wang, N. and Yeung, D.-Y. Learning a deep compact image representation for visual tracking. In *NIPS*, pp. 809–817, 2013.

Wang, N., Wang, J., and Yeung, D.-Y. Online robust non-negative dictionary learning for visual tracking. In *ICCV*, pp. 657–664, 2013b.

Whitehill, J., Wu, T.-F., Bergsma, J., Movellan, R.J., and Ruvolo, L.P. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pp. 2035–2043, 2009.

Wu, Y., Lim, J., and Yang, M. Online object tracking: A benchmark. In *CVPR*, pp. 2411–2418, 2013.

Yan, Y., Rosales, R., Fung, G., and Dy, J.G. Active learning from crowds. In *ICML*, pp. 1161–1168, 2011.

Zhong, B., Yao, H., Chen, S., Ji, R., Yuan, X., Liu, S., and Gao, W. Visual tracking via weakly supervised learning from multiple imperfect oracles. In *CVPR*, pp. 1323–1330, 2010.

Zhong, W., Lu, H., and Yang, M.-H. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pp. 1838–1845, 2012.