A Crowdsourcing Approach to Tracker Fusion

Wei Liu, Alexander G. Hauptmann

Carnegie Mellon University

Abstract. There are many tracking methods been proposed, using different features and algorithms, but none of them can track object correctly all the time. In this paper, we explore the idea of combining a crowd of trackers, and propose a crowdsourcing tracking method. We model the problem under the Sequential Monte Carlo framework, where we treat different trackers outputs, the bounding boxes, as weak observations, and use the wisdom-of-the-crowds to simultaneously infer both the hidden ground truth bounding box and the corresponding time-varying confidence for each tracker. We have tested our proposed method on two public surveillance video datasets and two of our own video datasets. The results show that the crowdsourcing tracking method can provide more stable and better performance.

1 Introduction

Object tracking [1] is an important yet very challenging task. It aims to stably and accurately estimate the trajectory of the object as it moves in the image plane. Numerous state-of-the-art tracking methods [2],[3],[4],[5],[6],[7],[8],[9] have been proposed using different object representation, different features, and different update mechanisms.

Due to the difficulties of object tracking problem, although each of these methods has its own merit in a particular scenario, none of them, however, can track object correctly all the time. We observe that single tracker is hard to track object correctly in a long video sequence, but different trackers can complement each other. For example, the kernel-based tracking method [4] can track object correctly when it is visually distinctive from the background, but poorly otherwise; while the motion-based tracker [2] can take advantage of motion information to correctly estimate the trajectory of the object, even when the object is non-distinctive from the background. As illustrated in Figure 1, although non of the individual trackers tracks the object accurately, we can achieve more stable and better performance by combining them.

In this paper, we model the meta-level tracker combination problem using the Sequential Monte Carlo (SMC) framework. In the traditional setting, it is assumed that there is only one reliable observation, which can be used to infer the hidden state. The contribution and novelty of this paper is that *we extend the traditional SMC setting by considering multiple weak observations per time step and their time-varying confidence, and infer both of them simultaneously using the wisdom-of-the-crowd strategy.*

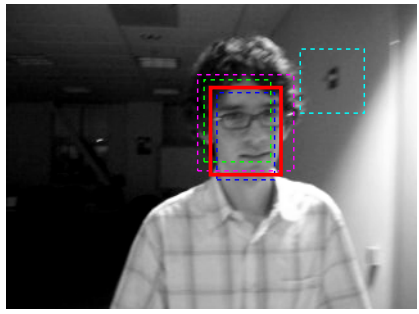


Fig. 1: Four dashed bounding boxes are four individual trackers. Three of them are not accurate enough, either too big, too small, or shifted-away. One of them even lose the target. The red solid bounding box is the "crowdsourcing" result, which is more stable and accurate.

We emphasize that the focus of this paper is not on various challenging issues in tracking problem, such as illumination change, occlusion, etc., but on the meta-level combination of multiple individual trackers, which we call **crowdsourcing tracking**. We take the term crowdsourcing in our method as an analogy to treat each individual tracker as a black box, providing the bounding box, from which we can infer the hidden state.

This work is different from previous related work, such as [10],[17],[11]. In [10], Stenger *et al.* choose the "best" tracker with the lowest error which is mapped from the confidence value returned by each individual tracker, where the mapping function is learned from training videos. However, we argue that high confidence value returned by a single tracker alone is not reliable and thus cannot tell whether a tracker is actually good or not. For example, kernel-based tracker returns high confidence if the tracked target is visually similar to the template even though it already loses the true target. [17] also suffers the same problem. A better strategy is to use the wisdom-of-the-crowds. Zhong *et al.* [11] use such idea. They randomly sample patches from the image and treat the tracking problem as a labeling process by using GLAD [12] model to infer the confidence for each patch and the accuracy for each individual tracker. However, they have not considered the time-smoothness of the bounding box. In our paper, we build our system based on the wisdom-of-the-crowds strategy, and model the tracking problem using the SMC framework, which explicitly considers the time-smoothness of both the hidden bounding box and the confidence, and our method is proved to be able to work better and more robustly than these methods and the like. Beside, our paper provides a clear framework how we should perform the inference and provides detailed explanation of the algorithm.

The rest of the paper is organized as follows. In section 2, we briefly introduce some previous related works. Then we provide the details of our methodvin section 3. Section 4 will present the experiment results. Conclusions and future work can be found in section 5.

2 Related work

The idea of combining multiple trackers results to achieve better performance is not new, several previous approaches have been proposed.

In [13], Siebel *et al.* first use motion detector to detect motion in the video, then use featuring splitting and merging from a region tracker for multiple hypotheses matching, and last use a head detector and Active Shape Tracker to refine the region and combine all the results. [14] combines two trackers, a region tracker and an edge tracker, each of which has complementary failure mode to correct each other. Spengler *et al.* [15] apply the cue-integration based on the principle of self-organization and self-adaptation to increase robustness of tracking results. However, all these methods rely on heuristics and ad hoc rules, which largely limit the usage of these methods in general case.

In [16], Moreno *et al.* use the Bayesian filters to integrate appearance and geometric features to achieve robustness in tracking. Each filter estimates the state of a specific feature, which is conditionally dependent on another feature estimated by another filter. Leichter *et al.* [17] propose a general Bayesian filter framework, and treat the individual trackers as "black boxes", whose outputs are modified before propagating to the next time step. Our method differs from this method because we use the wisdom-of-the-crowds mechanism during the inference and our method is formulated in a more elegant way using SMC framework. In [18] Toyama *et al.* also use a Bayesian network model which contains random variables that serve as context-sensitive indicators of the reliability of different tracking methods. It first learns these parameters offline, then fuses color, motion, and background subtraction into a single estimation. In [19] Du *et al.* selectively integrate four visual cues including color, edges, motion, and contours. The target is then tracked by a particle filter for each cue, and different cues can interact via Belief Propagation to pass messages within different filters. Stenger *et al.* [10] present a method for selecting suitable component observers for particular tracking tasks. It first applies the off-line training to evaluate each individual trackers, then proposes a cascade evaluation and a parallel evaluation to fuse the trackers on-line. [20] decomposes the observation and motion models for the bayesian filter tracker and applies the interactive Markov Chain Monte Carlo (iMCMC) to infer the weight for each decomposition and combine their results online. Avidan [21] treats the tracking problem as a classification problem, and used AdaBoost to combine several weak classifiers learned online to a strong classifier. Many of these methods restrict the types of trackers, and thus can be only applied in specific situations. In contrast, our method provides a general framework which can be applied to different type of trackers by treating each of them as black boxes which provides bounding box.

There are many crowdsourcing methods proposed as well. Most of the methods focus on image labeling problem where the label is binary value ($\{0, 1\}$). A common strategy to solve the labeling problem is to use the majority label as the estimation of the hidden true label by assigning all labelers the same weight. In [22], Raykar treats the problem as the chicken-and-egg problem, where he applies the EM algorithm to iteratively estimate the ground truth label and maximize

the sensitivity and specificity for annotators. [12] proposes a probabilistic model, GLAD, to simultaneously infer the label and difficulty of each image, and the expertise of each labeler. Welinder *et al.* [23] extend GLAD by representing each annotator as a multidimensional entity with variables representing competence, expertise and bias. All these methods can perform much better than majority voting method. However, they have not considered the time varying accuracy for each labeler in the labeling process. Other works, such as [24],[25] model the process of changes of accuracy of labelers during the labeling process. They, however, cannot model the tracking problem.

For the tracking problem, there is also work related to the idea of crowdsourcing. Zhong et. al [11] heuristically select several image patches at each time step where each individual tracker assigns the binary label ($\{0, 1\}$) to all the image patches according to its own result (bounding box); then they apply the GLAD model [12] to optimally estimate the confidence of positive label for each image patch and the accuracy for each tracker. But they treat the problem as a binary labeling problem and does not examine the time continuous nature of the tracking problem well enough.

To our knowledge, we are the first to apply the crowdsourcing idea on a Sequential Monte Carlo method for tracking.

3 Our model

We explore the idea of combining different trackers' results to achieve more stable and better performance, and propose the crowdsourcing tracking method.

Assume that we have N trackers $\mathbf{T} = \{T^j\}_{j=1}^N$ and their corresponding bounding boxes $\mathbf{Z}_t = \{z_t^j\}_{j=1}^N$ at time step t, with confidence $\mathbf{\Phi}_t = \{\phi_t^j\}_{j=1}^N$. The confidence describes how reliable each tracker is. Suppose the confidence is the correct estimation of the true confidence, then the higher the ϕ_t^j is, the more we can trust the j-th tracker T^j. The output of j-th tracker's at time t is the bounding box $z_t^j = [cx_t^j, cy_t^j, w_t^j, h_t^j]$, denoting the center point's x and y coordinates and the width and height of the bounding box; and the state of the ground truth bounding box for the object is $S_t = [cx_t, cy_t, w_t, h_t, vx_t, vy_t]$ where vx_t, vy_t denotes the velocity in x and y direction at time step t.

Figure 2 shows the framework of our model. As noted, there are two Markov chains: the upper chain is the dynamic model of the state of the ground truth bounding box S_t; and the lower one for the confidence of the crowd of trackers $\mathbf{\Phi}_t$. Both of them follow the first-order Markov assumption

$$\begin{aligned} p(S_t|S_0, \cdots, S_{t-1}) &= p(S_t|S_{t-1}) \\ p(\mathbf{\Phi}_t|\mathbf{\Phi}_0, \cdots, \mathbf{\Phi}_{t-1}) &= p(\mathbf{\Phi}_t|\mathbf{\Phi}_{t-1}) \end{aligned} \tag{1}$$

The observation $\mathbf{Z}_t$ is conditional independent of all the previous states and all the previous confidence given the current state S_t and current confidence $\mathbf{\Phi}_t$.

$$p(\mathbf{Z}_t|S_0, \mathbf{\Phi}_0, \cdots, S_t, \mathbf{\Phi}_t) = p(\mathbf{Z}_t|S_t, \mathbf{\Phi}_t) \tag{2}$$

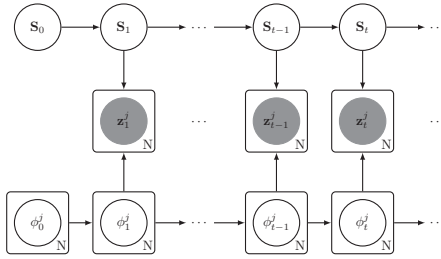


Fig. 2: The framework of our model. S_i is the hidden state of the bounding box we have to infer. ϕ_i^j is the hidden confidence for the j-th tracker at time step i. Given the hidden state, all the observations are conditional independent.

Our model aims to infer both the unknown state of ground truth bounding box S_t and the confidence of each tracker $\mathbf{\Phi_t} = \{\phi_t^j\}_{j=1}^N$ given the observation from N trackers' output bounding box to the current time step $\mathbf{Z}_{1:t} = \{z_{1:t}^j\}_{j=1}^N$.

First, we define the dynamic model of the state of the ground truth bounding box $p(S_t|S_{t-1})$ by using the first order auto-regression (AR) as follows

$$S_t = AS_{t-1} + q \tag{3}$$

where A is the transition matrix with the form of $\begin{pmatrix} I_{4\times4} & I_{2\times2} \\ 0 & I_{2\times2} \end{pmatrix}$ and q is a zero mean gaussian random variable with covariance matrix Q which determines the possible changes of the ground truth bounding box. In our experiment, we set the covariance matrix Q to be constant.

The confidence of j-th tracker $p(\phi_t^j|\phi_{t-1}^j)$ is modeled similar to (3)

$$\phi_t^j = \phi_{t-1}^j + h_j \tag{4}$$

where h_j is a zero mean gaussian random variable with variance σ_j. The smaller σ_j, the more stable. We also set it to be constant manually according to our prior knowledge of the individual trackers.

Next, we define the observation model based on the state of the ground truth bounding box and on the confidence of the crowd of trackers. Because we have N observations (bounding boxes) from the crowd of trackers, for simplicity reason, we assume that the trackers are conditional independent given the state and the confidence. Thus we can define the observation model as following

$$p(\mathbf{Z}_t|S_t, \mathbf{\Phi}_t) = \prod_{j=1}^{N} p(z_t^j|S_t, \phi_t^j) = \prod_{j=1}^{N} \beta(abs(f1_t^j - \phi_t^j); a, b) \tag{5}$$

where $f1_t^j$ is the F1score[1] between bounding box z_t^j with the bounding box y_t corresponding to the state S_t, where y_t is defined as $y_t = HS_t$ where $H = [I_{4\times4}, 0]$, $abs(\cdot)$ returns the absolute value, and $\beta(\cdot)$ is the Beta-distribution with $a = 0.5$ and $b = 2$. We use such Beta distribution because confidence ϕ_t^j should have high probability if it is close to the tracker's F1score $f1_t^j$.

3.1 Sequential Monte Carlo Inference

Notice that there are two unknown variables: the state of the ground truth bounding box S_t and the confidence for the crowd of trackers $\mathbf{\Phi}_t = \{\phi_t^j\}_{j=1}^N$. Here we use the particle filter technique to approximate the posterior density function with a discrete approximation using a set of random samples. We draw N_S random samples (particles) $\{S_t^i\}_{i=1}^{N_S}$ and the associated weights $\{w_{S_t}^i\}_{i=1}^{N_S}$ for the state S_t; and N_T random samples (particles) $\{\phi_t^{j,k}\}_{k=1}^{N_T}$ and the associated weights $\{w_{\phi_t^j}^k\}_{k=1}^{N_T}$ for the confidence $\{\phi_t^j\}_{j=1}^N$ of all the tracker $\{T^j\}_{j=1}^N$.

The estimation of the S_t and $\mathbf{\Phi}_t$ is a chicken-and-egg problem. In other words, to estimate the confidence for each tracker, we have to know the ground truth bounding box, and vice versa. We use the wisdom-of-the-crowds trick to fill this gap. We will describe the details of the estimation in later part of this section.

Initialization We assume the system has a initialization for both S_0 and $\mathbf{\Phi}_0 = \{\phi_0^j\}_{j=1}^N$. Then we can sample $\{S_0^i\}_{i=1}^{N_S}$ from the Gaussian distribution with S_0 as the mean and $\Sigma_0 = diag[5; 5; 3; 3; 1; 1]$ as the covariance matrix, and the weights for the samples $\{w_{S_0}^i\}_{i=1}^{N_S}$ are initially set equally to $\frac{1}{N_S}$.

And we do the same for the confidence $\{\phi_0^j\}_{j=1}^N$ of each tracker $\{T^j\}_{j=1}^N$ where we sample $\{\phi_0^{j,k}\}_{k=1}^{N_T}$ with $\{\phi_0^j\}_{j=1}^N$ as the mean and σ_j as the variance (which can be set differently manually), and the weights for the samples of confidence $\{w_{\phi_0^j}^k\}_{k=1}^{N_T}$ is set to be $\frac{1}{N_T}$.

Once we have the initialized samples for those variables, we can draw samples according to the dynamic model defined in (3) and (4). Then the inference is performed in two steps: prediction and update.

Prediction For time step t, when the new observation $\mathbf{Z}_t = \{z_t^j\}_{j=1}^N$ arrives, we first estimate the $\hat{p}(S_t = S_t^i|\mathbf{Z}_t)$ using the estimated expected accuracy of the trackers $\{E[\phi_{t-1}^j]\}_{j=1}^N$ and $p_{t-1}(S_t^i)$ from the previous time step t-1 as follows.

$$\begin{aligned}&\hat{p}(S_t = S_t^i|\mathbf{Z_t})\\ &= \frac{p_{t-1}(S_t^i)\prod_{j=1}^N p(z_t^j|E(\phi_{t-1}^j), S_t^i)}{\sum_{i=1}^{N_S} p_{t-1}(S_t^i)\prod_{j=1}^N p(z_t^j|E(\phi_{t-1}^j), S_t^i)}\end{aligned} \tag{6}$$

[1] $F1 = 2 \cdot \frac{prec \cdot rec}{prec + rec}$, where $prec = \frac{|z_t^j \cap y_t|}{|z_t^j|}$, $rec = \frac{|z_t^j \cap y_t|}{|y_t|}$, $|\cdot|$ is the size of the region, z_t^j is the observed bounding box, and y_t is the bounding box correspond to hidden state S_t.

where $p_{t-1}(S_t^i)$ is set to be $p(S_{t-1}^i|\mathbf{Z_{t-1}})$. We will describe how to estimate $E[\phi_{t-1}^j]$ in the next section.

Update After we compute the prediction of each sample of the state of the ground truth bounding box S_t, and then use the wisdom-of-the-crowds to update the samples of the confidence $\{\phi_t^{j,k}\}_{k=1}^{N_T}$ for each tracker$\{T^j\}_{j=1}^{N}$.

$$p(z_t^j|\phi_t^{j,k}, \mathbf{Z}_t) = \sum_{i=1}^{N_S} p(z_t^j|\phi_t^{j,k}, S_t^i)\hat{p}(S_t^i|\mathbf{Z}_t) \tag{7}$$

According to [27], if the dynamic model for the confidence of the crowd of trackers is chosen as the importance density distribution, then the weight $\{w_{\phi_t}^{j,k}\}_{k=1}^{N_T}$ for the confidence $\{\phi_t^j\}_{j=1}^{N}$ can be updated as follows

$$\begin{aligned} \hat{w}_{\phi_t}^{j,k} &= p(z_t^j|\phi_t^{j,k}, \mathbf{Z}_t)\hat{w}_{\phi_{t-1}}^{j,k}, t > 1 \\ w_{\phi_t}^{j,k} &= \frac{\hat{w}_{\phi_t}^{j,k}}{\sum_{i=1}^{N_T} \hat{w}_{\phi_{t-1}}^{j,k}} \end{aligned} \tag{8}$$

The posterior density can be approximated using a set of discrete random samples (particles) and associated weights to compute the expected confidence.

$$\begin{aligned} p(\phi_t^j|\mathbf{Z}_t) &\approx \sum_{k=1}^{N_T} w_{\phi_t}^{j,k}\delta(\phi_t^j - \phi_t^{j,k}) \\ E_{p(\phi_t^j|\mathbf{Z}_t)}[\phi_t^j] &= \sum_{k=1}^{N_S} p(\phi_t^{j,k}|\mathbf{Z}_t)\phi_t^{j,k} \end{aligned} \tag{9}$$

After applying the-wisdom-of-the-crowds strategy, we can estimate $p(S_t^i|\mathbf{Z_t})$ using (6), by replacing $p_{t-1}(S_t^i)$ with $\hat{p}(S_t^i|\mathbf{Z_t})$, and $E(\phi_{t-1}^j)$ with $E(\phi_t^j)$.

Resampling Besides, because of the *degeneracy phenomenon*, we will calculate the effective sample size $\hat{N}_{eff}^t$ which is defined as

$$\hat{N}_{eff}^t = \frac{1}{\sum_{i=1}^{N_S}(w_t^i)^2} \tag{10}$$

where w_t^i is the normalized weight defined in (8). Notice that $\hat{N}_{eff}^t \leq N_S$, and if $\hat{N}_{eff}^t$ is small, for example $\hat{N}_{eff}^t < \alpha N_S, \alpha = 0.5$, indicating most of the particles are of small weight, the approximation of (9) will not be accurate. Because such degeneracy problem is ineluctable, we use resampling to reduce such effect. The idea of resampling is to replace the particles with low weight with ones of large weight in (9), and after that, all the samples' weights are reset to $1/N_S$.

We should also do the resampling step for the samples for the confidence ϕ_t^j for each tracker. See Algorithm 1 for more details.

Algorithm 1 Crowdsourcing Tracking

Require: Initialize the state of the ground truth bounding box S_0 for the first frame and the confidence for N trackers $\mathbf{\Phi}_0 = \{\phi_0^j\}_{j=1}^N$

Ensure: $0 \leq \phi_0^j \leq 1 \quad j = 1, \cdots, N$

Draw samples from the initial distribution $S_0^i \sim p(S_0)$ for $i = 1, \cdots, N_S$ and assign weights $w_{S_0}^i = \frac{1}{N_S}$

Draw samples from the initial distribution $\phi_0^{j,k} \sim p(\phi_0^j)$ for $j = 1, \cdots, N; k = 1, \cdots, N_T$ and assign weights $w_{\phi_0}^{j,k} = \frac{1}{N_T}$

for $t > 0$ **do**

  Run the N individual trackers $\{T^j\}_{j=1}^N$

  Draw samples S_t^i using (3) and estimate the prior for the samples via (6) for $i = 1, \cdots, N_S$

  for $j = 1, \cdots, N$ **do**

    Draw samples $\phi_t^{j,k}$ using (4)

    for $k = 1, \cdots, N_T$ **do**

      Compute $p(z_t^j | \phi_t^{j,k}, \mathbf{Z}_t)$ via (7)

      Update weight $\hat{w}_t^{j,k} = p(z_t^j | \phi_t^{j,k}, \mathbf{Z}_t) \hat{w}_{t-1}^{j,k}$

    end for

    Normalize the weights $w_{\phi_t}^{j,k} = \frac{\hat{w}_{\phi_t}^{j,k}}{\sum_{i=1}^{N_S} \hat{w}_{\phi_{t-1}}^{j,k}}$

    Compute $\hat{N}_{z_t^j}^{eff} = \frac{1}{\sum_{k=1}^{N_T} (w_{\phi_t}^{j,k})^2}$

    if $\hat{N}_{z_t^j}^{eff} < thresold$ **then**

      Resample $\phi_t^{j,k} \sim pmf[w_{\phi_t}^j]$

      Reassign weights for samples $w_{\phi_t}^{j,k} = \frac{1}{N_T}$

    end if

    Compute the estimated expected confidence for the tracker $E_{p(\phi_t^j | \mathbf{Z}_t)}[\phi_t^j]$ via (9)

  end for

  reestimate $p(S_t | \mathbf{Z}_t)$ using (6) with the updated $E_{p(S_t | \mathbf{Z}_t)}(\phi_t^j)$

  Compute $\hat{N}_S^{eff} = \frac{1}{\sum_{i=1}^{N_S} (w_{S_t}^i)^2}$

  if $\hat{N}_S^{eff} < thresold$ **then**

    Resample $S_t^i \sim pmf[w_{S_t}]$

    Reassign weights for samples $w_{S_t}^i = \frac{1}{N_S}$

  end if

  Compute the estimated weighted combination of the samples of the state of ground truth bounding box via (9)

end for

4 Experiment

We have taken many state-of-the art trackers as the crowd of trackers. We list the main reference for each individual tracker in Table 1. For most of the trackers, we use the codes and default parameters provided by their authors. We also implemented some of the trackers by ourself, such as **bg**, **flow**, **mosift**, **ms**, and **pf**. The whole system can be run in MATLAB. The crowdsourcing tracking method takes about 1 second per frame on Intel(R) Core(TM) 2 Duo CPU E7500 @ 2.93GHz machine. If we include the computational time of the crowd of trackers, it cost about 6 seconds per frame on a single core. As noticed, we also include some non-tracking method, such as **pls** and **sfm**. It reflects the essential idea of crowdsourcing that it treats the tracker as black box.

Table 1: List of crowd of trackers

Tracker	Main Reference
bg	Non-parametric model for background subtraction [3]
bh	Visual Tracking with Histograms and Articulating Blocks [28]
B	Real-time tracking via on-line boosting [7]
SB	Semi-supervised On-line Boosting for Robust Tracking [29]
BSB	Beyond Semi-Supervised Tracking [30]
ems	An EM-like algorithm for color-histogram-based object tracking [31]
esm	Real-time image-based tracking of planes [32]
flow	An iterative image registration technique [2]
frag	Robust fragments-based tracking using the integral histogram [6]
ivt	Incremental learning for robust visual tracking [8]
MIL	Visual tracking with online multiple instance learning [9]
mosift	MoSIFT: Reocgnizing Human Actions in Surveillance Videos [33]
ms	Kernel-based object tracking [4]
pf	A boosted particle filter: Multitarget detection and tracking [5]
pls	Human Detection Using Partial Least Squares [34]
sfm	Sparse Field Methods - Technical Report [35]

4.1 Results on Caremedia dataset

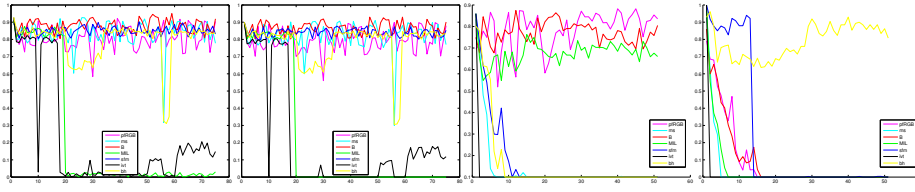
To test the robustness of our algorithm, and compare the individual trackers with our method, we have run our system on the Caremedia dataset, which consists of 13 video sequences recorded indoor in a nursing home, each of which has between 200 $\sim$ 2000 frames. For these 13 video sequences, we labeled the ground truth bounding box for each frame for some patients appearing in the video sequence. All videos are in RGB format and are of 720×480 pixels.

To demonstrate our method is better, we simply choose the following individual trackers: **pf**, **ms**, **B**, **SB**, **BSB**, **MIL**, **bg**, and **flow**. For **pf**, we used two different features: RGB and HoG, which corresponds to two distinctive trackers

pfRGB and **pfHoG**. We ran all these individual trackers on each frame and then used our proposed crowdsourcing tracking algorithm to combine the individual trackers' results. Our method achieved the best performance over all the trackers on average in F1score.

Table 2: F1score measure for Caremedia dataset

video	Worst Single	Best Single	**Ours**
c102	0.15 (**SB**)	**0.73 (pfHoG)**	0.73 (#3)
c102g	0.09 (**bg**)	**0.92 (pfHoG)**	0.91 (#3)
c102m	0.05 (**pfRGB**)	**0.83 (ms)**	0.81 (#2)
c102r	0.06 (**pfHoG**)	0.64 (**ms**)	**0.65 (#1)**
c102w	0.13 (**BSB**)	**0.79 (flow)**	0.76 (#5)
c106	0.01 (**SB**)	**0.55 (B)**	0.54 (#2)
c122	0.09 (**BSB**)	**0.41 (flow)**	0.38 (#3)
c131	0.06 (**bg**)	**0.75 (pfHoG)**	0.74 (#4)
c197	0.05 (**SB**)	0.42 (**ms**)	**0.43 (#1)**
c198	0.05 (**BSB**)	0.70 (**pfHoG**)	**0.72 (#1)**
c206	0.02 (**BSB**)	**0.68 (MIL)**	0.62 (#7)
c211	0.09 (**BSB**)	0.58 (**pfHoG**)	**0.59 (#1)**
c216	0.03 (**BSB**)	**0.63 (bh)**	0.33 (#8)
AVG	0.14 (**BSB**)	0.62 (**ms**)	**0.63 (#1)**



(a) Success estimate (b) Ground truth (c) Failure estimate (d) Ground truth

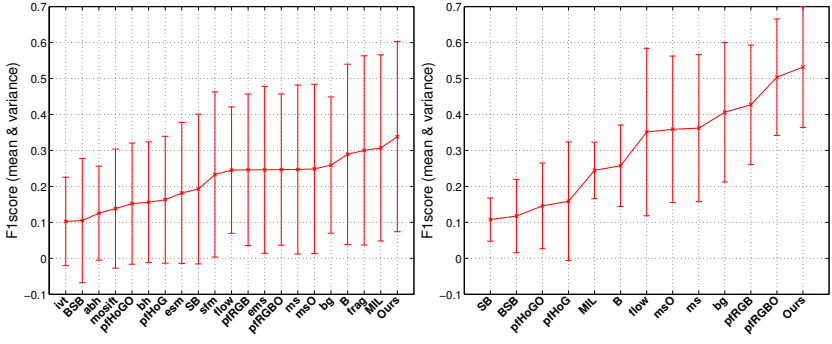
Fig. 3: Estimation of confidence for 5 trackers based on crowdsourcing tracking and on ground truth

Table 2 shows the F1score for the best and worst single tracker and our proposed method. According to these results, we can see that our method has the highest F1score in 4 out of 13 videos. For the other 8 videos, our method's F1score is very close to the best single tracker for 7 of them. This implies that our method can estimate the confidence of the individual trackers correctly on those video sequences, as shown in Figure 3a. However we also notice that for video c216, the performance of the best single tracker (**bh**) is much higher than our method. The reason for this is that most of the single trackers' performance

is very low, so even one of them (**bh**) has high confidence, our method still follow the majority. This is the problem of "the-madness-of-crowds". Figure 3c illustrates such case. But on average, our method is better than all the single trackers. It is the goal for the combination, to provide more robust and stable performance for all video sequences.

4.2 Results on CAVIARDATA1 and Traffic Stops

To test that our method can generalize well to different scenario, we run the similar process on two other datasets: CAVIARDATA1 and Traffic Stops. CAVIARDATA1[2] is a public surveillance video dataset, which has 78 video sequences, including 412 objects, with ground truth labeled. Each video is about 1000 $\sim$ 3000 frames. Because the dateset is so huge, we select the first frame whose ground truth bounding box's width and height is larger than 10 pixels as the initial bounding box for all the trackers. That may cause the low performance for many trackers. The other is our own dataset, Traffic Stops. It includes 30 videos monitoring the action of the policeman during the traffic stops. Each video has about 1000 $\sim$ 2000 frames. All videos are in RGB format and 704 $\times$ 480 pixels.



(a) CAVAIRDATA1 (b) Traffic Stops

Fig. 4: For different dataset, the performance of each individual trackers vary largely, but our method can always outperform the single trackers.

Figure 4 clearly shows that individual tracker's performance varies differently in different dataset. For example, in the CAVAIRDATA1, as shown in Figure 4a, **MIL** tracker performs the best among the individual trackers, however, we cannot simply take **MIL** and use it on all the Traffic Stops dataset. As shown in Figure 4b, we can see that **MIL** ranks 8-th out of 12 trackers. The similar thing also happens to other individual trackers. This affirms our assumption that non of the existing tracking method can track object correctly all the time. However,

[2] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

our method can getting better performance than all the individual trackers on the two datasets on average. It thus shows that our method can generalize well for different datasets.

4.3 Compare with other combining methods

In order to prove our method is better than other methods which share the similar idea, we compare with two methods, **majority** and **glad** [11]. We also compare with the-state-of-the-art method **vtd** as described in[20]. We use the online public dataset[3] to evaluate these methods.

Table 3 shows the comparison results of different methods. For **majority** tracker, we simply assign the same confidence to all the individual trackers and combine them. We can notice that both the **glad** tracker and our method is better than it, for the fact that both of the two methods use the wisdom-of-the-crowds to get better estimation of the confidence for each individual tracker. For the **glad** tracker, we implement it according to [11]. Since it has to generate samples per frame to be labeled, we experiment with two different numbers, where **glad** uses 1000 samples and **glad2** uses 10000 samples. For **glad**, it needs $1 \sim 2$ seconds to infer the bounding box, and $6 \sim 8$ seconds for **glad2**. Our method only needs less than 1 seconds per frame. We can notice that **glad2** is almost always better than **glad** for all the video sequences, except a little decrease on **coke11** and **dollar**, because more samples can cover more possible space in the image. Our method is always better than both of them. The reason is that **glad** tracker consider the tracking problem as a labeling process, which is not necessary the correct way to do it. Our methods, on the other hand, use the SMC framework, can naturally model the problem in an elegant way. What's more, **vtd** tracker is the-state-of-the-art tracker, we can see that our method can outperform it significantly by combining multiple weaker trackers results.

5 Conclusion and Future work

In this work, we examed the idea of crowdsourcing in the tracking scenario where the trackers are the crowd, their output bounding boxes are the weak observations. We provide a natural way to apply the Sequential Monte Carlo method in the crowdsourcing problem where we have multiple weak observations with different weight at every time step. Our experimental results all prove that our method can provide more stable and better performance by combining a crowd of individual trackers.

There are some issue which maybe interested for the future research. For example, in current model, the variance of the change of confidence for the trackers are set manually, in future work, we can consider learning such parameters from some training videos. Also, currently we consider all the trackers as conditional independent given the hidden bounding box, which may not be true because

[3] http://vision.ucsd.edu/ bbabenko/project_miltrack.shtml

Table 3: Comparison of different combining strategy, where F1$\in [0,1]$, the larger the better. dist is measured in pixel, the smaller, the better.

F1/dist	**majority**	**glad**	**glad2**	**vtd**	**Ours**
cliffbar	0.52/26	0.39/36	0.51/28	0.48/31	**0.53/23**
coke11	0.05/168	0.23/52	0.20/54	0.11/45	**0.32/32**
david	0.49/29	0.67/27	0.78/16	0.31/79	**0.81/12**
dollar	**0.47/60**	0.41/64	0.39/66	0.34/69	0.44/62
faceocc	0.92/8	0.83/19	0.90/11	0.90/7	**0.94/6**
faceocc2	0.07/166	0.48/54	0.51/53	0.08/139	**0.62/29**
girl	0.48/38	0.72/30	0.82/17	0.80/16	**0.83/16**
sylv	0.70/12	0.65/18	0.71/14	0.76/9	**0.77/10**
tiger1	**0.43/23**	0.34/29	0.40/25	0.16/49	0.35/25
tiger2	0.04/120	0.12/68	0.15/62	**0.29/38**	0.24/44
avg	0.42/65	0.49/40	0.54/34	0.42/48	**0.58/26**

for the trackers, which use similar feature and strategy, have strong correlation within them. If we can infer such relationship, such as the precision matrix within the trackers, we may get better performance.

Although our method needs to run many individual trackers before performing the combination, we can nevertheless run each of them individually. Because the computer has more and more cores, it is worthwhile and applicable to run our method to get more reliable results.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm Computing Surveys (2006)
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI. (1981)
3. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. ECCV (2000)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. PAMI (2003)
5. Okuma, K., Taleghani, A., Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. ECCV (2004)
6. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR. (2006)
7. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC. (2006)
8. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. IJCV (2008)
9. Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. (2009)
10. Stenger, B., Woodley, T., Cipolla, R.: Learning to track with multiple observers. In: CVPR. (2009)
11. Zhong, B., Yao, H., Chen, S., Ji, R., Yuan, X., Liu, S., Gao, W.: Visual tracking via weakly supervised learning from multiple imperfect oracles. In: CVPR. (2010)

12. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. NIPS (2009)
13. Siebel, N., Maybank, S.: Fusion of multiple tracking algorithms for robust people tracking. ECCV (2002)
14. Shearer, K., D Wong, K., Venkatesh, S.: Combining multiple tracking algorithms for improved general performance. Pattern Recognition (2001)
15. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. Machine Vision and Applications (2003)
16. Moreno-Noguer, F., Sanfeliu, A., Samaras, D.: Dependent multiple cue integration for robust tracking. PAMI (2008)
17. Leichter, I., Lindenbaum, M., Rivlin, E.: A general framework for combining visual trackers–the" black boxes" approach. IJCV (2006)
18. Toyama, K., Horvitz, E.: Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In: ACCV. (2000)
19. Du, W., Piater, J.: A probabilistic approach to integrating multiple cues in visual tracking. ECCV (2008)
20. Kwon, J., Lee, K.M.: Visual Tracking Decomposition. In: CVPR. (2010)
21. Avidan, S.: Ensemble tracking. PAMI (2007)
22. Raykar, V., Yu, S., Zhao, L., Jerebko, A., Florin, C., Valadez, G., Bogoni, L., Moy, L.: Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In: ICML. (2009)
23. Welinder, P., Branson, S., Belongie, S., Perona, P.: The Multidimensional Wisdom of Crowds. NIPS (2010)
24. Welinder, P., Perona, P.: Online crowdsourcing: rating annotators and obtaining cost-effective labels. CVPR Workshop (2010)
25. Donmez, P., Carbonell, J., Schneider, J.: A probabilistic framework to learn from multiple annotators with time-varying accuracy. In: SDM. (2010)
26. Donmez, P., Carbonell, J., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: ACM SIGKDD. (2009)
27. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on signal processing (2002)
28. Shahed Nejhum, S., Ho, J., Yang, M.: Visual tracking with histograms and articulating blocks. In: CVPR. (2008)
29. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. ECCV (2008)
30. Stalder, S., Grabner, H., Van Gool, L.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: ICCV Workshops. (2009)
31. Zivkovic, Z., Krose, B.: An em-like algorithm for color-histogram-based object tracking. In: CVPR. (2004)
32. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: IROS. (2004)
33. Chen, M., Hauptmann, A.: Mosift: Recognizing human actions in surveillance videos. Technical report, CMU CS (2009)
34. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: ICCV. (2009)
35. Lankton, S.: Sparse field methods-technical report. Technical report (2009)