Long Term Multi-Target Tracking based on Detection and Data Association

Ai Min Li
Shandong Polytechnic University, Jinan, China
Email: lam@spu.edu.cn

Pil Seong Park
University of Suwon, Suwon, Korea
Email: pspark@suwon.ac.kr

Abstract—Multi-target tracking is widely studied, but it is still an attractive but difficult research area because of existence of occlusion and interaction between target images. We propose a novel detection-based multi-target tracking method using data association. The main contribution of our method is providing a strategy to quickly correct the wrong tracker aroused by occlusion. We use somewhat unreliable detection confidence to assist a particle filter tracker. The resulting algorithm is tested using movies having much occlusion. The result shows good performance, fast enough to run real time for a relatively long period of time.

Index Terms—multi-target tracking, Particle filter, Hungarian algorithm, data association

I. INTRODUCTION

Multi-target tracking is very important in many areas, such as surveillance and human-computer interaction applications, but the task is also very difficult because of the existence of frequent occlusion and interaction between target images.

Many researchers have tried to apply particle filters to multi-target tracking, but their performance is poor when the posterior is multimodal as a result of ambiguities or multiple targets. Hence Vermaak et al. [1] proposed a mixture particle filter (MPF) and Okuma et al. [2] combined tracking-by-detection with MPF and proposed a boosted particle filter (BPF).

In recent years detection-based tracking methods became popular. These kinds of algorithms obtain object hypotheses by applying an object detector to images. The detector is learned generally offline from labeled training data. Given detection responses generated by the detector, the tracking method needs to retrieve real objects among those responses and sets an ID for each of them in every frame. To deal with such data association problems, two strategies are commonly adopted — associating the responses locally (frame by frame) or globally.

Our tracking method is also based on detection, simply by applying the basic background subtraction method to get possible target positions. Then according to the number of detected objects in that manner, we can tell if occlusion has occurred or not. If there is no occlusion, we will adopt the Hungarian algorithm to get an optimal matching between targets and detected objects, and then divide the particles into two groups to track targets not only by using the previous tracking results, but also by sampling the most possible detection regions. Since our method considers only the current frame and the previous tracking results, it can run online.

This paper is organized as follows: Sec. II introduces some related works. Sec. III describes the standard particle filter method. In Sec. IV, we introduce our method based on detection and data association in detail. Experimental results are presented in Sec. V, followed by conclusions in Sec. VI.

II. RELATED WORKS

Okuma et al. [2] applied a particle filter to track multiple hockey players. They proposed a mixture proposal distribution that incorporates information from the dynamic models of each player and the detection hypotheses generated by Adaboost. The method utilized offline trained classifiers. If much occlusion exists, then some trackers will lose the target.

The use of independent trackers requires solving a data association problem to assign measurements to targets. Classical approaches include the JPDAF (joint probabilistic data-association filter) proposed by Fortmann et al. [3], and the MHT (multiple hypothesis tracking) proposed by Reid [4]. However the computational complexity grows exponentially with the number of targets and time steps, respectively. We stick to an optimal matching scheme for making the detection-tracker assignments. Such an approach is also used by Wu and Nevatia [5], but there the assignments are made only based on spatial distance, without considering target appearance. This can be problematic for complex scenes with many targets and cluttered background where much false positive detection occurs. They additionally learn color histograms for each part, which however do not always distinguish very well. Recently, Song et al [6]...
presented a background subtraction based tracker that also learns target specific classifiers online, but employs them only when targets split and merge.

Benfold and Reid [7] proposed a multi-target tracking system that performs data association over a sliding window of frames. The approach is multi-threaded and combines asynchronous HOG (histogram of oriented gradients) detection with simultaneous KLT (Kanade-Lucas-Tomasi Feature Tracker) tracking and Markov-Chain Monte-Carlo Data Association (MCMCDA) to provide guaranteed real-time tracking in high definition video. To improve the performance, multi-threaded approach and GPU are implemented.

Breitenstein et al [8] uses the continuous confidence of HOG or ISM (Implicit Shape Model) pedestrian detectors and online trained classifiers as a graded observation model. They use the classifier for data association, and use combination of final detection, continuous detector confidence and classifier output to guide particles. But their algorithm can only process 0.4-2 frames/s.

The Hungarian method is a combinatorial optimization algorithm, and is often used to solve an assignment problem. Some researchers use the method to perform data association in tracking problems in recent years. To maximize consistency of visual and grouping cues for data association in tracking problems, to propagate the particles, a constant velocity motion model is often adopted:

\[
(x,y) = (x,y)_{-t} + (xv, yv)_{-t} \cdot \Delta t + \mu_{(x,y)}
\]

\[
(xv, yv) = (xv, yv)_{-t} + \mu_{(xv,yv)}
\]

Here \(\mu_{(x,y)}\) and \(\mu_{(xv,yv)}\) denote the process noises, which are zero mean white Gaussians.

To calculate the likelihood, we must use some observation model. Commonly used observation models built for particle filter tracking are edge-based, color-based, or contour-based features, etc. Relying on only one feature is less robust and of limited performance in complex scenarios, hence many hybrid feature particles are proposed, for example [10].

For simplicity, we assume that all the targets have some color difference from each other to some extent. Hence we only adopt the RGB color histogram for observation measurement. And we apply the Bhattacharyya similarity coefficient to measure the distance \(\xi[H,H(X)]\) on the RGB histograms, where \(H\) is the initial target color histogram and \(H(X)\) is the candidate color histogram.

Once we obtain the distance \(\xi\) on the RGB color histograms, we use the following likelihood distribution given by P’erez et al. [11]:

\[
p(Y_t | X_t) \propto e^{-\xi^2[H,H(X)]}
\]

It is easy to update each particle’s weight according to its likelihood.

\[
W_t^{x_i} = W_{t-1}^{x_i} p(Y_t | X_t)
\]

Then we normalize the weights.

\[
W_t^i = \frac{W_t^{x_i}}{\sum_{i=1}^{N} W_t^{x_i}}
\]

The estimation of Maximum a Posteriori(MAP) is obtained over \(N\) samples at each time \(t\), which explains well the current state with the given observation,

\[
X_t^{MAP} = \arg \max_{X_t} p(X_t | Y_{1:t})
\]

for \(i=1,2, \ldots, N\).

After a few iterations, all but a small number of particles will have negligibly small weights. This phenomenon is called degeneracy. A re-sampling method is adopted to solve such a degeneracy problem, that is to say, to sample \(N\) particles with replacement from the set of particles \(X_t^i\) according to importance weight \(W_t^i\). The particles with small weights will be replaced.
IV. OUR ADAPTIVE TRACKING METHOD

To track multiple targets simultaneously for a long time, we applied a particle filter method based on detection results.

A. Target Detection

First, to detect targets, we adopt the simple background subtraction method in each frame. However the result may have much small false detection because of existence of noise, and sometimes, because one target may be represented by more than one blobs, which should be merged.

To remove such small false outliers and merge those small blobs belonging to the same target, we first set an appropriate distance threshold $\text{Dist}_{th}$ to merge those small blobs which are very close. Then an appropriate area threshold $\text{Area}_{th}$ is set to filter out those small outliers.

Fig.1 shows some detection results. We can see that after the filter and merging operation, the detection result is much better.

![Detection results in different frames.](image)

The figures in the left column show results without filtering and merging operation. Those in the right column show the comparison results with filering and merging operation.

B. Associating Detected Blobs with Targets

Depending on the detection results, the number of detected blobs may not equal to the number of the initialized targets. If the number of detected blobs is greater than that of existing targets, then there may be some false detection or some new targets. We can use some methods to initialize new particles to track the new target. In this paper, we assume that the number of targets is fixed. Our method can be extended to track unfixed number of targets.

In order to decide which detection should guide which tracker, we need to solve a data association problem by assigning at most one detection to at most one target. The optimal single-frame assignment can be obtained by using the Hungarian algorithm, which is known to solve the assignment problem in polynomial time.

To describe the algorithm easily, we denote the problem as a bipartite graph when the number of detected blobs is larger than the number of the target being tracked.

We have a complete bipartite graph $G=(D,T;E)$ with $n$ detection vertices ($D$) at time $t$ and $m$ target vertices ($T$), and each edge $E$ has nonnegative confidence $C(i,j)$. The bold green lines in Fig. 2 represent the perfect matches with maximum confidence.

In Fig. 2, $D_i(i=1...n)$ are the detected blobs at time $t$, some of which may be false detection or new targets. If a detection appears consecutively in some frames at almost the same position, and no existing tracker keeps tracking it, we can assume that a new target appeared, and initialize a new particle filter tracker to track it.

In Fig. 3, the blue boxes surrounding the targets are the detection results. The green, magenta and cyan rectangles represent the tracking results by particle filters. They are labeled with ID’s. Near the right boundary of the image, the three targets are shown. We calculate each detection’s color histogram $H(D_t,i)$, $i=1...n$. We already have the color histograms of the targets being tracked. So we can calculate the likelihood between the detected blobs and the targets. Then we can get a matrix, the entries of which are the confidence values between the detected blobs and the targets being tracked.

![The association between detected blobs and targets.](image)
To match the targets and detected blobs, normally, (3) is used. However, to obtain a better likelihood difference, we consider the spatial layout of color distribution of a body. The detected regions can be divided into $k$ sub-regions each. The initialized target blobs are divided in the same way. Then we can apply the likelihood as the sum of the reference histograms associated with each sub-region by:

$$\sum_{i=1}^{k} \log p(Y_t | X_t, H_i)$$

A similar idea has been adopted in [2]. And in our work, we just divide each detection region and target blob into two sub-regions, as shown in Fig. 4.

For multi-target tracking, our aim is to find the most possible target matching to the separated detection $D_{t,i}$. We can formulate it in this way:

$$Y_{t,m}^\text{MAP} = \arg \max_{Y_{t,m}} p(Y_{t,m} | D_{t,i})$$

$T_{t,m}$ represents the positions of all $m$ targets. Using the Hungarian algorithm, we can get a good match.

### C. Division of particles based on detection

Let $N$ be the total number of particles we sample for each target. Then we divide the particles into two groups: $M$ particles and $N-M$ particles.

$$N = \alpha \times N + (1-\alpha)N = M + (N-M), M = \alpha \times N$$

$M = \alpha \times N$ is the number of particles to be put at the possible detected position, where $\alpha$ is some coefficient we can adjust according to our experiments. These $M$ particles will be re-sampled around the matched detection position $Y_t$. The rest $N-M$ particles will be re-sampled at the position $X_{t,j}$ in a traditional way according to the weighted confidence we get from the previous frame.

The detection may not be so reliable, and we may not be able to tell which detected blob corresponds to which target. But using the Hungarian method we can get an optimal and more reliable match between the detection regions and the targets.

Without occlusion, our tracking process is nearly the same as a normal particle filter method, because both groups of particles will be re-sampled nearly at the actual target position. However, by adopting the Hungarian method, we can utilize the detection confidence to assist the particle filter tracker to be more robust.

If some target reappears after occlusion and is detected, the algorithm tries to recover the tracker by placing $M$ particles of the most likely target at the detected position, as shown in Algorithm I.

**Algorithm I: Particle Filter Based on Detection**

1. At a certain time, let $T_{t,m}$ represents the $m$ targets initialized by some method.

2. In each frame, for all targets:
   - Detect possible target blobs $D_{t,i}$ by background subtraction.
   - Match $T_{t,m}$ with target blobs $D_{t,i}$ using the Hungarian method.
   - Divide particles into 2 groups to track each target, the $1^\text{st}$ group to be placed around the position obtained by a normal particle filter method, and the other group to be placed at the position of the matched blobs.
   - Get the particle with highest likelihood as the tracking result (as is done in an ordinary particle filter method)

End for

### V. Experiment and Result

We tested our algorithm using a movie that includes complicated and frequent occlusion. The total number of particles used to track each target is $N=100$. According to our experience, the number of particles to be put at the detected position is chosen to be 30% ($\alpha = 0.3$, i.e., $M = 30$). The algorithm is implemented in VS C++ 2010 with OpenCV2.2 library support, and can be run on a normal laptop (Intel i5-480M, 4GB memory) in real time.

In the movie (having 910 frames with resolution 320*240) we use, three targets are moving around in a small area, showing frequent occlusion. To illustrate the behavior of our algorithm, the detection results are drawn in blue. The three tracking boxes and particles (the points at the center position of the tracking box) are drawn in...
green, magenta and cyan respectively, in Figs. 5 and 6. In both figures, (a) and (b) show the tracking results by our method, and (c) and (d) by normal particle filter method.

In Fig. 5, when target 1 (tracked by green particles) is re-detected after occlusion at frame 504, a small group of particles are placed at the position, as matched by the Hungarian method. Only after two frames, the tracker for target 1 is recovered, as shown in (b)(frame 506).

Similarly, in Fig. 6(a,b), target 2 (tracked by magenta particles) reappeared and was detected after occlusion at frame 603. Then the small group of particles of the magenta tracker was placed at the detected position to try re-capture the target. And after a few frames, the tracker was recovered at frame 607.

At frame 504, when target 1 reappears after occlusion. (b) Target 1 is successfully recaptured, and all green particles are placed at the actual position, and the tracker is recovered. However, without our strategy, the 1st target is recaptured at frame 504 in (c), but failed to recapture the 1st target at frame 506 in (d).

To evaluate the performance of our method quantitatively, we define evaluation metrics as in Table I.

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNF</td>
<td>Total Number of Frames</td>
</tr>
<tr>
<td>CTF</td>
<td>Correctly Tracked Frames: The tracking box overlaps with at least 2/3 of the bounding box of the target.</td>
</tr>
<tr>
<td>TLT</td>
<td>Total Number of Lost Targets: The target is not occluded but failed to track correctly.</td>
</tr>
<tr>
<td>IDS</td>
<td>ID Switches: The total number of times that a tracked target changes its ID with some other target.</td>
</tr>
<tr>
<td>RS</td>
<td>Recovery ratio after short term occlusion</td>
</tr>
<tr>
<td>RL</td>
<td>Recovery ratio after long term occlusion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary particle filter method</td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>910</td>
</tr>
<tr>
<td>Ordinary particle filter method</td>
<td>910</td>
</tr>
</tbody>
</table>

From the comparison result in Table II, we can conclude that our strategy can improve the tracking performance greatly. Our method can restore the tracker even after relatively a long term occlusion.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel particle filter re-sampling method in multi-target tracking based on object detection and data association. We use the detection confidence to assist and correct the particle filter, which makes the tracker more robust. Our method can recapture the target even after a long term occlusion. Experiments with our algorithm show good performance. In the future, we will try to combine HOG detection or ISM detection into our method to get a better tracking result.
ACKNOWLEDGMENT

This work was supported by the GRRC program of Gyeonggi province. [GRRC SUWON2012-B1, Cooperative CCTV Image Based Context-Aware Process Technology]

REFERENCES


Ai Min Li is an associate professor at Information School of Shandong Polytechnic University, China. His research interest includes image processing, computer vision, software engineering, and computer network. He received his M.S. degree in Software Engineering from Shandong University, China.

Pil Seong Park is a professor of Computer Science and the director of Information & Computer Center at Suwon University, Korea. His research interest includes computer vision, especially tracking objects for security and surveillance, Linux clusters, high performance computing, and knowledge-based information systems. He received his Ph.D. in Interdisciplinary Applied Mathematics (with emphasis on Computer Science) from University of Maryland at College Park, M.S. in Applied Mathematics from Old Dominion University, U.S.A., and B.S. in Physical Oceanography from Seoul National University, Korea.