Vision-Based Multiple Interacting Targets Tracking via On-Line Supervised Learning

Xuan Song, Jinshi Cui, Hongbin Zha, and Huijing Zhao

Key Laboratory of Machine Perception (Ministry of Education),
Peking University, China
{songxuan,cjs,zha,zhaohj}@cis.pku.edu.cn

Abstract. Successful multi-target tracking requires locating the targets and labeling their identities. This mission becomes significantly more challenging when many targets frequently interact with each other (present partial or complete occlusions). This paper presents an on-line supervised learning based method for tracking multiple interacting targets. When the targets do not interact with each other, multiple independent trackers are employed for training a classifier for each target. When the targets are in close proximity or present occlusions, the learned classifiers are used to assist in tracking. The tracking and learning supplement each other in the proposed method, which not only deals with tough problems encountered in multi-target tracking, but also ensures the entire process to be completely on-line. Various evaluations have demonstrated that this method performs better than previous methods when the interactions occur, and can maintain the correct tracking under various complex tracking situations, including crossovers, collisions and occlusions.

1 Introduction

Multiple targets tracking plays a vital role in various applications, such as surveillance, sports video analysis, human motion analysis and many others. Multi-target tracking is much easier when the targets are distinctive and do not interact with each other. It can be solved by employing multiple independent trackers. However, for those targets that are similar in appearance, obtaining their correct trajectories becomes significantly more challenging when they are in close proximity or present partial occlusions. Specifically, maintaining the correct tracking seems almost impossible when the well-known "merge/split" condition occurs (some targets occlude others completely, but they split after several frames). Hence, the goals of this research are: 1) to devise a new method that will help obtain a better tracking performance than those obtained from previous methods when the interactions occur; 2) to make a new attempt to solve the "merge/split" problem in the multi-target tracking area.

In this paper, we present an on-line supervised learning based method for tracking a variable number of interacting targets. The essence of this research is that the learning and tracking can be integrated and supplement each other in

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 642–655, 2008.

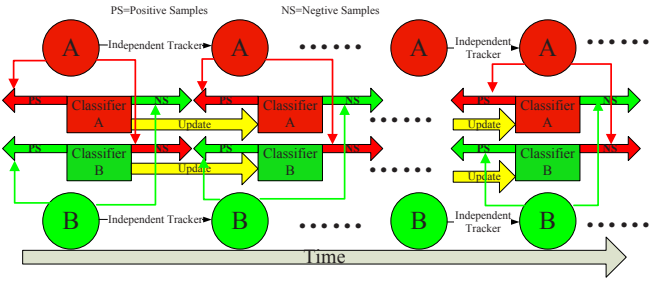


Fig. 1. Tracking for learning: When A and B do not interact with each other, we employ independent trackers to track them and the tracking results are used for learning. For each classifier of targets, the positive samples are dependent on its tracking results, and the negative samples are dependent on the other targets. We should update the classifiers per frame when the new samples are coming in.

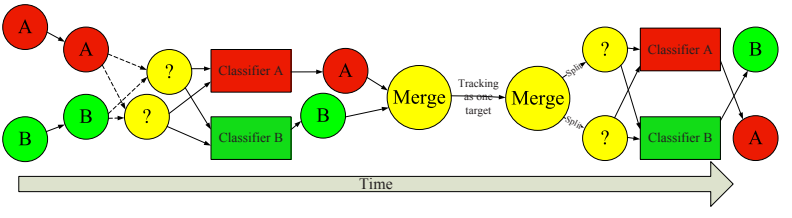


Fig. 2. Learning for tracking: When the targets are in close proximity or a "merge/split" condition occurs, we use these classifiers to assist in the tracking.

one framework to deal with various complex tracking problems. The core idea of our method can be depicted in Fig.1 and Fig.2. For purposes of simplicity, we only track two targets, A and B. When the two targets do not interact with each other (see Fig.1), tracking becomes very easy and multiple independent trackers are employed. Due to the reliability of these tracking results, they are used as positive or negative samples to train a classifier for each target. When the two targets are in close proximity (see Fig.2), the learned classifiers are used to assist in tracking. Specifically, when the two targets merge, we assign a new state space and track this "merging target" as one target. When they split, their classifiers are used again to specify a correct identification.

In this research, we extend some exciting learning based single target tracking method [1,2] into the multi-target tracking area. In this procedure, we solve two crucial problems which is the main contribution of this paper: (1) Tracking for learning: we solve the problem of how to obtain the positive and negative samples with no human-interaction to achieve the on-line supervised learning. (2) Learning for tracking: we solve the problem of how to use these learned classifiers to deal with difficult problems (interaction or merge/split) encountered in multi-target tracking.

Compared to the traditional multi-target tracking algorithms, our method offers several advantages: Firstly, due to the appearance of each target which is depicted by a classifier with a supervised learning process, the appearance model of the targets become increasingly stronger with the time-lapse and sufficiently exploit targets' history information. Moreover, since the pieces of information from other targets are treated as negative samples for the learning, each classifier considers the other target information and has the strong distinguishability. Through these classifiers, we can deal with challenging tracking situations easily. Secondly, our method can automatically switch tracking and learning, which make them supplement each other in one framework and ensure that the entire process is completely on-line. Lastly, our method is a general method that can be extended in many ways: any better independent tracker, learning algorithm and feature space can be employed in the proposed method.

The remainder of this paper is organized as follows: In the following section, related work is briefly reviewed. Section 3 introduces the switch of learning and tracking in different tracking situations. Section 4 and 5 provide the details about tracking for learning and learning for tracking. Experiments and results are presented in Section 6 and the paper is finally summarized in Section 7.

2 Related Work

Over the last couple of years, a large number of algorithms for multi-target tracking have been proposed. Typically, multi-target tracking can be solved through data association [3]. The nearest neighbor standard filter (NNSF) [3] associates each target with the closest measurement in the target state space. However, this simple procedure prunes away many feasible hypotheses and cannot solve "labeling" problems when the targets are crowed. In this respect, a widely approach to multi-target tracking is achieved by exploiting a joint state space representation which concatenates all of the targets' states together [4,5,6] or inferring this joint data association problem by characterization of all possible associations between the targets and observations, such as joint probabilistic data association filter (JPDAF) [3,7,8], Monte Carlo technique based JPDA algorithms (MC-JPDAF) [9,10] and Markov chain Monte Carlo data association (MCMC-DA) [11,12,13]. However, with the increasing number of tracking targets, the state space becomes increasingly large and obtaining accurate MAP estimation in a large state space becomes quite difficult. Furthermore, the computational complexity of most methods mentioned above grows exponentially with the increasing tracking targets.

Additionally, researchers also propose multiple parallel filters to track multiple targets [14], that is, one filter per target where each one has its own small state space. In spite of this, when the interactions among targets occur, this method encounters difficulty in maintaining the correct tracking. Therefore, modeling the interactions among targets becomes an incredibly important issue. Khan et al. [15] use a Markov random field (MRF) motion prior to model the interactions among targets. Qu et al. [16] proposed a magnetic-inertia potential modeling

to handle the "merge error" problem. Lanz et al. [17] proposed a hybrid joint-separable model to deal with the interactions among targets. Sullivan et al. [18] tracked the isolated targets and the "merging targets" respectively, and then connected these trajectories by a clustering procedure. Nillius et al. [19] employed a track graph to describe when targets are isolated and how they interact. They utilized Bayesian network to associate the identities of the isolated tracks by exploiting the graph. However, most of these methods mentioned above (except [18,19]) consider tracking as a Markov process, which fail to sufficiently exploit target history information and present a strong appearance model of the targets.

Recently, there has been a trend of introducing learning techniques into single target tracking problems, and tracking is viewed as a classification problem in the sense of distinguishing the tracking target from the background. Representative publications include [1,2,20,21]. In this work, we extended this concept into the multi-target tracking to deal with complicated problems encountered in multi-target tracking.

3 Tracking Situation Switch

We have approached different situations in the tracking, such as non-correlated targets tracking, interacting targets (present partial occlusions or merge with each other) tracking, splitting of "merging targets" and appearing or disappearing of targets. We aim to detect these conditions and make the tracking and learning switch automatically from one to another. In this work, we employed a detection-driven strategy to deal with this task.

In each frame, we had to obtain the detections of the targets which aided in judging the tracking situations. There have been a large number of human detection algorithms in recent years [22,23], which can provide reliable and accurate detected results. However, the basic assumption of this research is that the background is static. Hence, we only utilized the simple background subtraction method [24,25] to obtain the detections. After the background subtraction, we utilized Mean-shift [26] to search the centers of detections.

We defined four conditions for each target: non-correlated targets condition, correlated targets condition, merge/split condition and appearing/disappearing of targets. A statistical distance function was employed to aid in detecting these conditions:

$$\frac{(X^*_{t,k} - x)^2}{G^2\sigma_x^2} + \frac{(Y^*_{t,k} - y)^2}{G^2\sigma_y^2} = 1 \tag{1}$$

where $\mathbf{d}^*_{t,k} = (X^*_{t,k}, Y^*_{t,k})$ is the predicted position of target k in frame t, G the threshold, σ_x and σ_y the covariance. We utilized it to search for possible detections of targets and the targets in different conditions are defined as follows:

Non-correlated targets. If a target locates only one detection and this detection is only possessed by this target, then this is a non-correlated target (as shown in Fig.3-b).

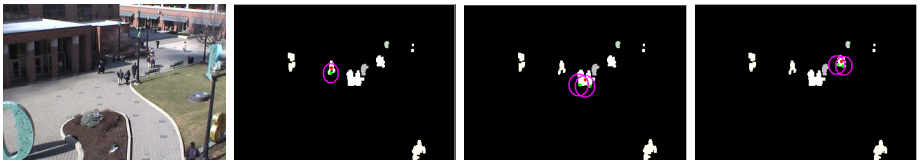


Fig. 3. footnotesize **Different tracking situations.** The red points are centers of detections obtained by Mean-shift; the green points are the predicted position of the targets and the ellipses are the distance function. (b) One target only locates one detection and this detection is only processed by this target; this is a non-correlated target. (c) One detection is shared by two targets and its area is larger than a threshold; the two targets are correlated targets. (d) One detection is also shared by two targets, but its area is smaller than the threshold. Hence, the two targets are merging. We track them as one target, when this merging target locates more than one detection; we believe that these targets split.

Correlated targets. If one detection is shared by more than one targets and the area of this detection is larger than the threshold that depends on the scale parameter, we believe that these targets are in close proximity or present partial occlusions. Hence, they are correlated targets (as shown in Fig.3-c).

Merge/Split condition. If one detection is shared by more than one targets and the area of this detection is smaller than the threshold, we believe that these targets present complete occlusions. We define these targets as merging. Several frames after, should this merging target discover greater than one detections, we believe these targets split. Hence, this is a merge/split condition (as shown in Fig.3-d).

Appearing/Disappearing of targets. If a target on the edge of the coordinate plane cannot locate any detection for some continuous frames, this target may disappear. We save the state of this target and stop to track it. Similarly, if a detection on the edge of the coordinate plane cannot locate any targets for some continuous frames, this should be a new target. We assign a new state space for this target and start to track it.

Therefore, we can detect these conditions easily and automatically switch tracking and learning.

4 Tracking for Learning

When the targets do not interact with each other (non-correlated targets), tracking becomes relatively easier since it can be solved through multiple independent trackers. Specifically, the obtained results are accurate and credible. Consequently, these tracking results can be utilized as samples for a supervised learning. In this section, we provided details about the independent trackers and the on-line supervised learning process.

4.1 Independent Tracker

Our method is a general method that does not depend on the independent tracker. Therefore, any tracker with reasonable performance can be employed, such as Meanshift [27] and CONDENSATION [28].

In this work, we utilized the color-based tracking model [29] and employed the detection-based particle filter [14] to perform the non-correlated targets tracking. The state space $\mathbf{x}_{t,k}$ of target k in frame t are defined by $\mathbf{x}_{t,k} = [\mathbf{d}_t, \mathbf{d}_{t-1}, s_t, s_{t-1}]$, where $\mathbf{d} = (X, Y)$ is the center of bounding box in the image coordinate system, and s is the scale factor. Let $\mathbf{y}_t$ denotes observations. A constant velocity motion model is utilized, which could be best described by a second order autoregressive equation

$$\mathbf{x}_{t,k} = \mathbf{A}\mathbf{x}_{t-1,k} + \mathbf{B}\mathbf{x}_{t-2,k} + \mathbf{C}N(0, \mathbf{\Sigma}) \tag{2}$$

where matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{\Sigma}$ are adjusted manually in the experiments. $N(0, \mathbf{\Sigma})$ is a Gaussian noise with zero mean and standard deviation of 1. The likelihood for the filtering is represented by HSV color histogram similarity [27], which can be written as

$$P(\mathbf{y}_t|\mathbf{x}_{t,k}) \propto e^{-\lambda D^2(K^*, K(\mathbf{d}_{t,k}))} \tag{3}$$

where D is the Bhattacharyya distance [27] on HSV histograms, K^* the reference color model, $K(\mathbf{d}_{t,k})$ the candidate color model, and λ is adjusted to 20 in the experiment After re-weighting and re-sampling in the particle filter, we obtain the new position of each target.

4.2 Random Image Patches for Learning

Once we obtained the tracking results of each target at time, a set of random image patches [30] are spatially sampled within the image region of each target. We utilized these random image patches as samples for the online supervised learning. In this case, each target is represented by a "bag of patches" model.

Extracting distinguishable features from image patches is relatively important for the learning process. There have been a large number of derived features that can be employed to represent the appearance of an image patch, such as raw RGB intensity vector, texture descriptor (Haralick feature), mean color vector and so on. Suggested by paper [30], we employed the color + texture descriptor to extract features from image patches. We adapted an d-dimensional feature vector to represent each image patch. Therefore, these feature vectors can be utilized as samples for the learning or testing.

4.3 On-Line Supervised Learning

For each target, a strong classifier should be trained, which represents the appearance model of targets. Let each image patch be represented as a d-dimensional feature vector. For target k in frame t, $\{\mathbf{s}^i_{t,k}, l^i_{t,k}\}_{i=1}^N$ denote N samples and their labels, where $\mathbf{s} \in \Re^d$ and $l \in \{-1, +1\}$. The positive samples are the image

patches come from region of target k, while the negative samples are the image patches that come from other targets. In this work, we employed Classification and Regression Trees [31] as weak classifiers. Once the new samples are available, the strong classifier should update synchronously, which would make the classifier stronger and reflect the changes in the object appearance. Therefore, poor weak classifiers are removed and newly trained classifiers are added, which is motivated by *Ensemble Tracking* [1]. The whole learning algorithm is shown below.

Learning Algorithm

Input: Feature vectors of image patches and their labels $\{\mathbf{s}_{t,k}^i, l_{t,k}^i\}_{i=1}^N, t = 1, ..., T$
Output: The strong classifier $H(\mathbf{s}_{t,k})$ of target k at time t
Train a Strong Classifier(for frame 1):

1. Initialize weights $\{w_i\}_{i=1}^N$ to be $1/N$.
2. For $j = 1...M$ (train M weak classifiers)
   (a) Make $\{w_i\}_{i=1}^N$ a distribution.
   (b) Train a weak classifier h_j.
   (c) Set $err = \sum_{i=1}^N w_i |h(\mathbf{s}_{1,k}^i) - l_{1,k}^i|$.
   (d) Set weak classifier weight $\alpha_j = 0.5 \log(1 - err)/err$.
   (e) Update example weights $w_i = w_i e^{\alpha_j |h_j(\mathbf{s}_{1,k}^i) - l_{1,k}^i|}$
3. The strong classifier is given by $sign(H(\mathbf{s}_{1,k}))$, where $H(\mathbf{s}_{1,k}) = \sum_{j=1}^M \alpha_j h_j(\mathbf{s}_{1,k})$.

Update the Strong Classifier (for new frame t is coming in)

1. Initialize weights $\{w_i\}_{i=1}^N$ to be $1/N$.
2. For $j = 1...K$ (choose K best weak classifiers and update their weights)
   (a) Make $\{w_i\}_{i=1}^N$ a distribution.
   (b) Choose $h_j(\mathbf{s}_{t-1,k})$ with minimal err from $\{h_1(\mathbf{s}_{t-1,k}), ...h_M(\mathbf{s}_{t-1,k})\}$.
   (c) Update α_j and $\{w_i\}_{i=1}^N$.
   (d) Remove $h_j(\mathbf{s}_{t-1,k})$ from $\{h_1(\mathbf{s}_{t-1,k}), ...h_j(\mathbf{s}_{t-1,k})\}$.
3. For $j = K + 1...M$(add new weak classifiers)
   (a) Make $\{w_i\}_{i=1}^N$ a distribution.
   (b) Train a weak classifier h_j.
   (c) Compute err and α_j .
   (d) Update examples weights $\{w_i\}_{i=1}^N$.
4. The updated strong classifiers is given by $sign(H(\mathbf{s}_{t,k}))$, where $H(\mathbf{s}_{t,k}) = \sum_{j=1}^M \alpha_j h_j(\mathbf{s}_{t,k})$.

5 Learning for Tracking

When the targets are in close proximity, it is difficult to maintain the correct tracking with the independent trackers. Specifically, when the "merge/split" conditions occur, associating the identities of the targets becomes a significantly challenging problem. In this case, the learned classifiers of the targets can be utilized to assist in tracking. In this section, we have provided details on how to employ these classifiers to deal with difficult problems encountered in the tracking.

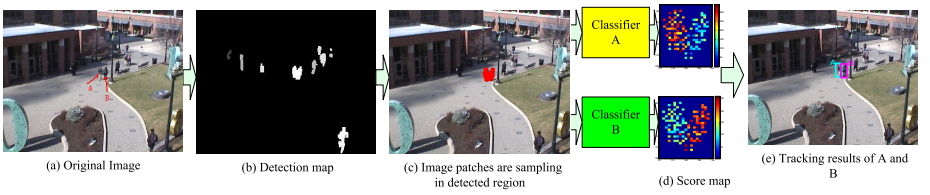


Fig. 4. Correlated targets tracking: We detected that A and B were correlated targets (Fig. a); some random image patches were sampling in their detected region (Fig. c). We used their classifiers to obtain their score maps (Fig. d). After the particle filtering process, we acquired the tracking results (Fig. e).

5.1 Correlated Targets Tracking

As the discussions in section 3, if the targets are in close proximity or present partial occlusions, we conclude that they are correlated targets. When this condition occurs, a set of random image patches are sampled within the interacting region of the detected map, and the feature vectors of these image patches are imputed to the classifiers of interacting targets respectively. The outputs of these classifiers are scores. Hence, we can obtain the score maps of these interacting targets effortlessly.

Once we obtain the score maps of the interacting targets, we employ the particle filter technique [32] to obtain the positions of these targets. The likelihood for the update in the particle filter is

$$P_{scores}(\mathbf{y}_t|\mathbf{x}_{t,k}) = \frac{1}{\sqrt{2\pi/\sigma}} \sum_{i=1}^{N} \beta_i \exp\left(\frac{(\mathbf{d}(\mathbf{x}_{t,k}) - \mathbf{d}_{t,k}^i)^2}{\sigma^2}\right) \tag{4}$$

where β_i is the normalized score of image patch i, $\mathbf{d}(\mathbf{x}_{t,k})$ the center position of candidate target k, $\mathbf{d}_{t,k}^i$ the center position of image patch i, and σ is the covariance which depends on the size of the image patch. For each target, the observation is further peaked around its real position. As a result the particles are much focused around the true target state after each level's re-weighting and re-sampling. Subsequently, we obtain the new position of these interacting targets. The overview of the process is shown in Fig.4.

5.2 Merge/Split Condition

Sometimes, several targets occlude another target completely. Maintaining the correct tracking of targets seems quite impossible. Once this condition occurs, we deal with it as a merge/split condition.

Upon detecting that some targets merge together as discussed in section 3, we initialize the state of the "merging targets" and track it as one target, which is similar to the non-correlated targets tracking depicted in section 4. If we detect that this "merging target" splits and becomes an interacting condition or

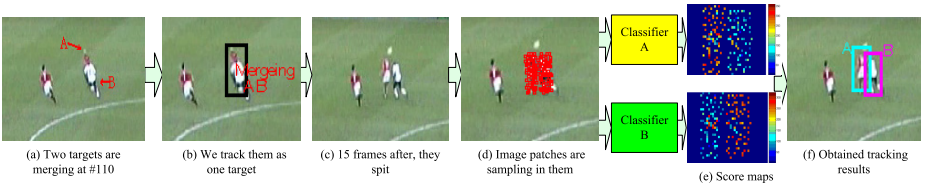


Fig. 5. Merge/Split condition: In frame 110, we detected that A and B were merging (Fig. a); we track A and B as one target (Fig. b). After 15 frames, we detected that they split (Fig. c), and some random image patches were sampling in them (Fig. d). We used their classifiers to obtain their score maps (Fig. e). After the particle filtering process, we obtained the tracking results (Fig. f).

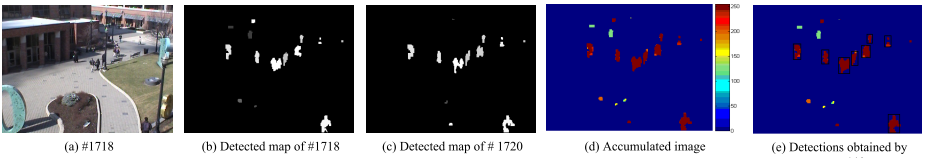


Fig. 6. Disposal of uncertain detections: For the OTCBVS dataset, false alarms frequently took place (Fig. b and Fig. c). We accumulated some continuous frames (Fig. d) and used Mean-shift to obtain the detections (Fig. e).

non-correlated condition, we utilized the classifiers of these targets to identify them (as shown in Fig.5). Hence, we can link the trajectories of these targets without difficulty.

With the help of the classifiers, our method is able to deal with various complex situations in the tracking. In addition, the tracking and learning supplement each other in the proposed method, consequently becoming an adaptive loop, which ensures all the process to be completely on-line.

6 Experiments and Results

We evaluated the proposed method in the different kinds of videos, such as SCEPTRE Dataset [33], OTCBVS Benchmark Dataset [34] and our surveillance videos. The selected data used for testing were five different clips in which complex interactions frequently took place. All the algorithms mentioned in the experiments were implemented by the non-optimized MATLAB code. The results and comparisons are detailed in this section.

6.1 Disposal of Uncertain Detections

For the SCEPTRE Dataset and surveillance video, we achieved reliable detections by using background subtraction, since their background was simple or the targets in the image were large. However, for the OTCBVS Benchmark Dataset, the detections obtained by the background subtractions were unreliable: False

Fig. 7. Disposal of interactions or "merge/split" among targets: The first row is the tracking results of multiple independent color-based trackers [29]. The second row is the results of multiple independent Ensemble Trackers [1] and the the third is our tracking results.

alarms or ruptured human bodies frequently occurred (as shown in Fig.6-b,c), which sometimes had influenced on the tracking. Hence, we employed a practical method to deal with this problem in the experiments. We accumulated some continuous detection maps and utilized Mean-shift to obtain the centre of the new detections (as shown in Fig.6-e). We discovered that this simple strategy could deal with most false alarms or non-connected human bodies, ensuring the robustness of the tracking.

6.2 Tracking Results

Fig.7 displayed the efficacy of the proposed method to deal with interactions or "merge/split" among multiple targets. In this experiment, we utilized our method, multiple independent color-based trackers [29] or Ensemble Trackers [1] only to perform the tracking respectively. We can see that our method can deal with "merge/split" problem easily and maintain the correct identifications of targets when they split (at frame 323), which is difficult to just utilize the two kinds of independent trackers.

Tracking results of different datasets under complex tracking situations were displayed in Fig.8, Fig.9 and Fig.10. More tracking results can be seen in our supplementary video.

6.3 Quantitative Comparison

We conducted two groups of comparisons to show the tracking performance of different methods under interacting situations. The first comparison was among some methods which can track a variable number of targets, and the second was among the methods which can track a fixed number of targets. The selected dataset for testing was SCEPTRE which had the most complex interactions in

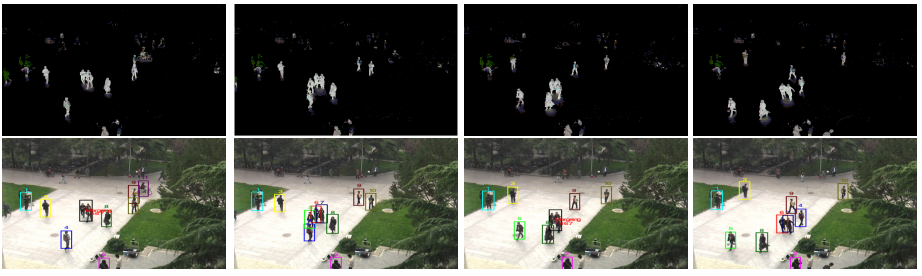


Fig. 8. Tracking results of surveillance video: The first row is the detection of targets obtained by background subtraction; the second row is the tracking results of our method. Note that targets 5, 6 and 7 were merging in frame 231 and targets 4, 6 and 7 were merging in frame 305; when they split, we were still able to maintain their correct tracking. Please see our supplementary video for more details.

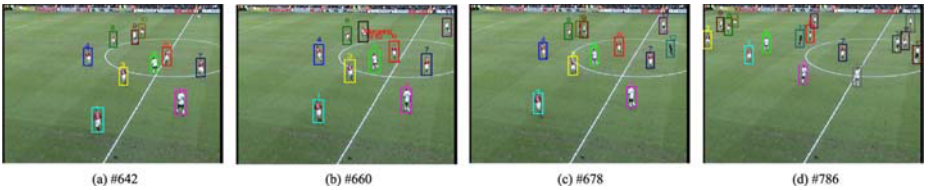


Fig. 9. Tracking results of SCEPTRE dataset: This dataset is very challenging, where complex interactions frequently occurred. This is an example of our results. Note that there was an interaction among targets 9 and 10 in frame 660; they split in the frame 678. In the frame 786, we still maintained their correct tracking. Please see our supplementary video for more details.

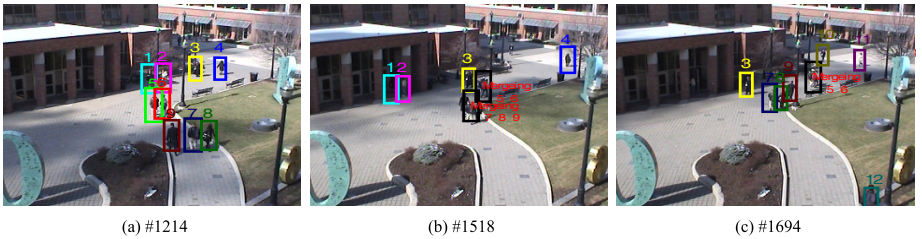
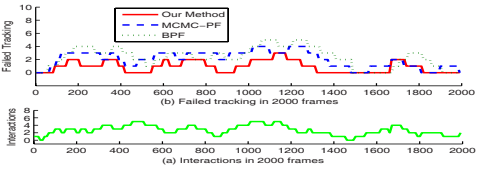


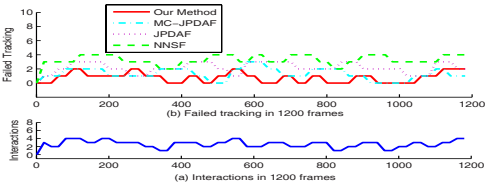
Fig. 10. Tracking results of OTCBVS dataset: Note that target 7, 8, 9 and target 5, 6 were merging in frame 1518. Please see our supplementary video for more details.

the three kinds of video. The ground truth was obtained by software *ViPER-GT* [35], and the failed tracking were including target missed, false location and identity switch.



Algorithm	Success Rate
BPF	68.66%
MCMC-PF	75.63%
Ours	**83.75%**

Fig. 11. Quantitative comparison under interacting situations among three methods which can track a variable number of targets



Algorithm	Success Rate
NNSF	62.17%
JPDAF	73.13%
MC-JPDAF	82.75%
Ours	**86.20%**

Fig. 12. Quantitative comparison under interacting situations among four methods which can track a fixed number of targets

In the first experiment, we performed a quantitative comparison with two famous multi-target tracking algorithms: Boosted Particle Filter (BPF) [4] and MCMC-based Particle Filter (MCMC-PF) [15]. For the BPF, AdaBoost detections were displaced by background subtraction. We conducted a statistical survey of 2000 continuous frames to evaluate the tracking performance of these methods under interacting situations. Fig.11 illustrates the quantitative evaluation of three methods and the success rate of these methods is shown in the right table.

In the second experiment, we conducted a comparison with several classical data association algorithms: Joint Probabilistic Data Association Filter (JPDAF), Monte Carlo Joint Probabilistic Data Association Filter (MC-JPDAF) and Nearest Neighbor Standard Filter (NNSF). Because JPDAF and MC-JPDAF can only track a fixed number of targets, in which the tracking targets must stay in the image all along. Therefore, we tracked seven targets which stayed in the image for 1200 continuous frames. A quantitative evaluation of the four methods is shown in Fig.12.

Although the proposed method obtained better tracking performance than above methods under interacting situations and could deal with "merge/split" easily, our method also has some limitations. For the SCEPTRE dataset, we discovered that when some similar appearance players merged or split, our method might fail due to the similar score map obtained by the classifiers. Majority of the failed tracking in this dataset was caused by this condition. In the future, a more powerful feature space (including other cues, such as motion or shape) should be exploited to solve this problem.

7 Conclusion

In this paper, a novel on-line supervised learning based method is presented for tracking a variable number of interacting targets. Different evaluations describe the superior tracking performance of the proposed method under complex situations. Our method is a general method that can be extended in many ways: It is a much robust independent tracker; more powerful feature space; reliable detection algorithm and faster supervised learning algorithm. For the present testing datasets, we concluded that a better feature space that can distinguish targets with similar appearance and a much robust detection algorithm can further improve the tracking performance significantly. This task can be achieved in future.

Acknowledgments. This work was supported in part by the NKBRPC (No.2006CB303100), NSFC Grant (No.60333010), NSFC Grant (No.60605001), the NHTRDP 863 Grant (No.2006AA01Z302) and (No.2007AA11Z225). We specially thank Kenji Okuma and Yizheng Cai for providing their code on the web. We also thank Vinay Sharma and James W. Davis for providing us their detected results.

References

1. Avidan, S.: Ensemble tracking. IEEE Trans. PAMI 29, 261–271 (2007)
2. Le, L., Gregory, D.: A nonparametric treatment for location/segmentation based visual tracking. In: Proc. IEEE CVPR, pp. 261–268 (2007)
3. Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Academic Press, New York (1998)
4. Okuma, K., Taleghani, A., Freitas, N.D., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
5. Vermaak, J., Doucet, A., Perez, P.: Maintaining multi-modality through mixture tracking. In: Proc. IEEE ICCV, pp. 1110–1116 (2003)
6. Zhao, T., Nevatia, R.: Tracking multiple humans in complex situations. IEEE Trans. PAMI 7, 1208–1221 (2004)
7. Rasmussen, C., Hager, G.: Probabilistic data association methods for tracking complex visual objects. IEEE Trans. PAMI 23, 560–576 (2001)
8. Gennari, G., Hager, G.: Probabilistic data association methods in visual tracking of groups. In: Proc. IEEE CVPR, pp. 876–881 (2004)
9. Vermaak, J., Godsill, S.J., Perez, P.: Monte carlo filtering for multi target tracking and data association. IEEE Trans. Aerospace and Electronic Systems 41, 309–332 (2005)
10. Schulz, D., Burgard, W., Fox, D., Cremers, A.: People tracking with a mobile robot using sample-based joint probabilistic data association filters. International Journal of Robotics Research 22, 99–116 (2003)
11. Oh, S., Russell, S., Sastry, S.: Markov chain monte carlo data association for general multiple target tracking problems. In: Proc. IEEE Conf. Decision and Control, pp. 735–742 (2004)
12. Khan, Z., Balch, T., Dellaert, F.: Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. IEEE Trans. PAMI 28, 1960–1972 (2006)

13. Yu, Q., Medioni, G., Cohen, I.: Multiple target tracking using spatio-temporal markov chain monte carlo data association. In: Proc. IEEE CVPR, pp. 642–649 (2007)
14. Cai, Y., Freitas, N.D., Little, J.J.: Robust visual tracking for multiple targets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 125–135. Springer, Heidelberg (2006)
15. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. IEEE Trans. PAMI 27, 1805–1819 (2005)
16. Qu, W., Schonfeld, D., Mohamed, M.: Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model. In: Proc. IEEE ICCV, pp. 535–540 (2005)
17. Lanz, O., Manduchi, R.: Hybrid joint-separable multibody tracking. In: Proc. IEEE CVPR, pp. 413–420 (2005)
18. Sullivan, J., Carlsson, S.: Tracking and labeling of interacting multiple targets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 661–675. Springer, Heidelberg (2006)
19. Nillius, P., Sullivan, J., Carlsson, S.: Multi-target tracking - linking identities using bayesian network inference. In: Proc. IEEE CVPR, pp. 2187–2194 (2006)
20. Li, Y., Ai, H.Z., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life-spans. In: Proc. IEEE CVPR, pp. 1–8 (2007)
21. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proc. IEEE CVPR, pp. 260–267 (2006)
22. Zhe, L., Larry, S.D., David, D., Daniel, D.: Hierarchical part-template matching for human detection and segmentation. In: Proc. IEEE ICCV, pp. 351–358 (2007)
23. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: Proc. IEEE CVPR, pp. 661–668 (2005)
24. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE ICCV, pp. 37–63 (1999)
25. Davis, J., Sharma, V.: Fusion-based background-subtraction using contour saliency. In: Proc. IEEE CVPR, pp. 20–26 (2005)
26. Comaniciu, D., Visvanathan, R., Meer, P.: Kernel-based object tracking. IEEE Trans. PAMI 25, 564–575 (2003)
27. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. IEEE CVPR, pp. 142–149 (2000)
28. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 28, 5–28 (1998)
29. Perez, P., Hue, C., Vermaak, J.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
30. Lu, L., Hager, G.: Dynamic foreground/background extraction from images and videos using random patches. In: Proc. NIPS, pp. 351–358 (2006)
31. Breiman, L., Friedman, J.H., Olshen, R., Stone, C.J.: Classification and regression trees. Wadsworth, Chapman Hall, New York (1984)
32. Doucet, A., Godsill, S.J., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing 10, 197–208 (2000)
33. SCEPTRE-Dataset, `http://sceptre.king.ac.uk/sceptre/default.html`
34. Davis, J., Sharma, V.: Otcbvs benchmark dataset 03, `http://www.cse.ohio-state.edu/otcbvs-bench/`
35. ViPER-GT, `http://viper-toolkit.sourceforge.net/products/gt`