

Real-Time Multi-scale Tracking via Online RGB-D Multiple Instance Learning

Xianhua Zeng^{1,2*}, Yipeng Gao^{1,2}, Suli Hou^{1,2}, Shuwen Peng^{1,2}

¹ Chongqing Key Lab of Computational Intelligence, Chongqing 400065, China.

² College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

* Corresponding author. Tel.: +86-15823426466; email: xianhuazeng2005@163.com

Manuscript submitted October 24, 2014; accepted August 17, 2015.

doi: 10.17706/jsw.10.11.1235-1244

Abstract: It is still a challenging problem to develop robust target tracking algorithm under various environments. Most of current target tracking algorithms are able to track objects well in controlled environments, but they usually fail in significant variation of the target's scale, pose and plane rotation. One reason for such failure is that these object tracking algorithms employ fixed-size tracking box, and the other is that traditional 2D feature-based tracking algorithms are lack of 3D information. In this paper, we address the two problems by combining the fused 3D features and the bootstrap filter. So a multi-scale RGB-D tracker is proposed. The multi-scale RGB-D tracker has several attractive merits: (1) It exploits multi-instance learning strategy for fusing effective 3D information from the RGB image and the corresponding depth image. (2) It uses the bootstrap filter to solve the problem of target losing, when target changes significantly in scale. (3) Our tracking algorithm also reduces the error accumulation and can obtain a good performance when the candidate target is not so well. Extensive experiments demonstrate the effectiveness of the proposed tracking algorithm in indoor and outdoor environments where the targets undergo large changes in pose, scale, and plane rotation.

Key words: Target tracking, multi-scale, bootstrap filter, multi-instance learning, multi-feature fusion.

1. Introduction

Visual object tracking is one of the core problems of computer vision, with wide-ranging applications including human-computer, surveillance, auto-driving, motion classification and medical imaging [1]. The tracking-by-detection approaches [2]-[8] have become particularly popular recently, which treats the tracking as a binary classification task to separate target from background. Collins and Liu [2] indicated that there exist the most discriminative features which could be learnt to separate target from the background online. Grabner *et al.* [3], [4] have proposed an online boosting algorithm to training features and then relieved the drift by an online semi-supervised boosting method, where they only label the samples of the first frame. Avidan [5] has applied a support vector machine classifier modified by the optical flow approach for tracking. Babenko *et al.* [6] have proposed to handle the ambiguity problem generated by using multiple positive samples and negative ones to update the classifier. The MIL tracker puts the positive and negative bags respectively, and then trains a classifier in an online manner using the bag likelihood function. Zhang *et al.* [7] have proposed an effective tracking algorithm with an appearance model based on

feature extraction in the compressed domain, and several positive and negative samples are used to update the classifier. Recently, K. Zhang has also put forward the WMIL [8] tracker which integrates the sample importance into an efficient online learning procedure by assuming that the most importance sample is known when training the classifier. There have been also some seminal works on tracking using RGBD cameras [9]-[11], but they all focus on tracking the human body.

Fixed-size tracking box is used in the above tracking algorithm based on 2D feature or 3D feature to detect and recognize samples, it will regard the part of the object as target when the target becomes bigger compared with initial scale, and takes the mixture image of the object and background as target when the target become smaller. That leads to loss of target when the target changes significantly in scale. Besides, traditional tracking algorithms based on 2D features are lack of 3D information that makes it easy to fail to track targets stably in pose variation and plane rotation.

In this paper, we propose an effective and efficient multi-scale RGB-D tracking algorithm which introduces an improved discriminative appearance model into the Bootstrap Particle filter to follow the object scale variation. The improved appearance model is based on Haar features and Haar-D features extracted with the multiple instance learning strategy which could fully represent the target, Meanwhile a naive Bayesian classifier is used to generate the observation model of the Bootstrap filter. The basic working flowchart of our multi-scale target tracker is shown in Fig. 1. A observation model based on the sample with confidence and a state equation considering two previous statuses have been used to estimate the current position and scale status for each particle from depth image and color image.

The rest of this paper is organized as follows: In Section 2, we introduce the weighted online multiple instance learning tracking algorithm. In Section 3 we present our tracking algorithm and the main principle of our tracker in detail. Section 4 gives the detailed experiment setup and results. Finally, Section 5 gives some conclusion remarks and the future research in the target tracking field of computer vision.

2. Tracking with Weighted Online Multiple Instance Learning

The classical MIL [6] tracker has demonstrated good performance to handle drift. However, the MIL tracker may detect the positive sample that is less important because it does not discriminately consider the sample importance in its learning procedure. The WMIL [8] tracker integrates the sample importance into an efficient online learning procedure by assuming that the most important sample is known when training the classifier. The classifier $H_k(x)$ is defined as

$$H_k(x) = \ln \left(\frac{p(f(x) | y = 1) p(y = 1)}{p(f(x) | y = 0) p(y = 0)} \right) = \sum_{k=1}^K h_k(x), \quad (1)$$

where

$$h_k(x) = \ln \left(\frac{p(f_k(x) | y = 1)}{p(f_k(x) | y = 0)} \right). \quad (2)$$

The WMIL [8] tracker uses a set of Haar-like features $f(x) = (f_1(x), \dots, f_K(x))^T$ to represent samples x . It assumes that the features in $f(x)$ are independent distributions and the prior information $p(y = 1) = p(y = 0)$ in WMIL tracker [8]. Then, Eq. (4) is used as a discriminative function for constructing a weak classifier in WMIL tracker [8] and our tracker. Then the conditional distributions in classifier $h_k(\cdot)$ are assumed to be Gaussian distributed, that is

$$p(f_k(x_{ij}) | y_i = 1) \sim N(\mu_i, \delta_i) \quad (3)$$

$$p(f_k(x_{ij}) | y_i = 0) \sim N(\mu_0, \delta_0) \quad (4)$$

Then, the update schemes are described for the parameters μ_1 and δ_1 as follows:

$$\mu_1 \leftarrow \eta \mu_1 + (1-\eta) \frac{1}{N} \sum_{j|y_j=1} f_k(x_{ij}) \quad (5)$$

$$\delta_1 \leftarrow \eta \delta_1 + (1-\eta) \sqrt{\frac{1}{N} \sum_{j|y_j=1} (f_k(x_{ij}) - \mu_1)^2}, \quad (6)$$

where η is the learning rate for updating, and N is the number of positive samples. We update μ_0 and δ_0 with similar rules. At the same time, the WMIL [8] tracker also defines two bag probabilities as follows:

$$p(y=1 | X^+) = \sum_{j=0}^{N-1} w_{j0} p(y=1 | x_{1j}) \quad (7)$$

$$\begin{aligned} p(y_0=0 | X^-) &= \sum_{j=N}^{N+L-1} w p(y_0=0 | x_{0j}) \\ &= w \sum_{j=N}^{N+L-1} (1 - p(y_0=1 | x_{0j})) \end{aligned} \quad (8)$$

where the weight w_{j0} is a monotone decreasing function with respect to the Euclidean distance between the location of sample x_{1j} and x_{10} , i.e.

$$w_{j0} = \frac{1}{c} e^{-|l(x_{1j}) - l(x_{10})|}, \quad (9)$$

where c is a normalization constant. Eq. (9) weighs the positive instances according to their importance to the bag probability, i.e., the instances near the tracking location at the current frame have more contribute to the bag probability than those far from the tracking location. Therefore, we can assume that all of negative instances contribute equally to the negative bag.

Similar to MIL [6] tracker, WMIL [8] tracker also maintains a weak classifier pool $\Phi = \{h_1, \dots, h_M\}$, firstly. Then we select K weak classifiers to construct the strong classifier as $H_k = \sum_{k=1}^K h_k$ by using the following criterion:

$$h_k = \arg \max_{h \in \Phi} \langle h, \nabla l(H) \rangle |_{H=H_{k-1}} \quad (10)$$

The strong classifier H_k is applied to the cropped patches in the new frame to determine the one with the maximum confidence, i.e. $x^* = \arg \max_x p(y=1 | x)$ as the new object location.

The weak classifier selection criterion in eq. (10) is much more efficient than the directly maximizes the log-likelihood function used by MIL tracker. However, fixed-size tracking box are still used in [8] to detect and recognize samples, which will regard the part of the object as target when the target becomes bigger compared with initial scale, and it takes the mixture image of object and background as a target when the target becomes smaller. That leads to the loss of target when the object's scale has significant variation. And the WMIL tracker based on 2D features is lack of 3D information, that makes it easy to fail to track targets stably in pose variation or plane rotation. From the above introduction, analysis and inspiration, we will propose our multi-scale RGB-D tracker in next section.

3. Multi-scale RGB-D Tracker

For overcoming the limitation of the aforementioned fixed-size tracking box and using only 2D features, we present a multi-scale RGB-D tracking algorithm in details. Our algorithm flowchart (also named as multi-scale RGB-D tracker) is shown in Fig. 1. Specially we address the above problems by combining the fused 3D features via multiple instance learning strategy and the bootstrap filter.

3.1. RGB-D Fusion Strategy

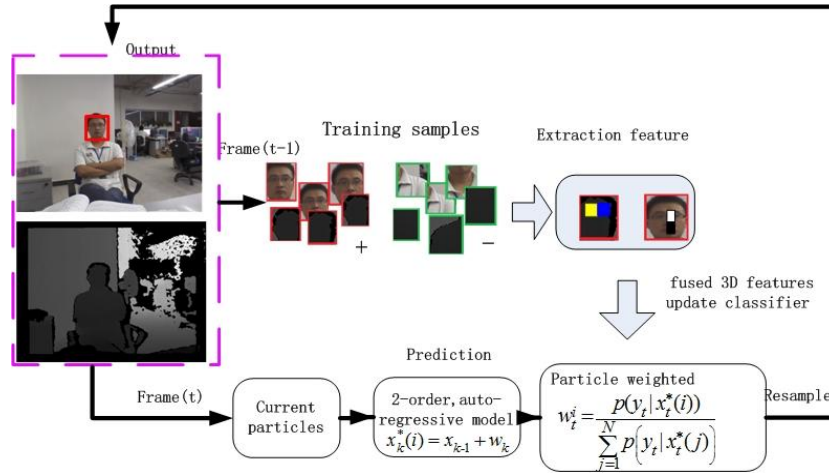


Fig. 1. The flowchart of our multi-scale RGB-D tracker.

We take advantages of the richness of RGB-D data by computing two types of features to train respectively the two weak classifiers: Haar-like features in the intensity image (converted from the RGB values), Haar-D (Haar-like features) features in the depth image. And then the instance probability is modeled by:

$$P_d(y=1|x_{1j}) = \sigma(H_{dk}(x_{1j})), \quad (11)$$

$$P_r(y=1|x_{1j}) = \sigma(H_{rk}(x_{1j})), \quad (12)$$

where $H_{rk}(\cdot), H_{dk}(\cdot)$ (i.e. (1)) is a strong classifier which is constructed by the weak classifier (i.e. (3)) related to each Haar-like feature and Haar-D feature. $\sigma(z) = 1/(1+e^{-z})$ is a sigmoid function and $y \in \{0,1\}$ is a binary label of sample x_{1j} . We obtain the positive bag probability $\{P_{d1}, P_{r1}\}$ and the negative bag probability $\{P_{d0}, P_{r0}\}$ on the depth image and color image in (7), (8). The probability P_{r1} from the Haar-like feature and the probability P_{d1} from the Haar-D feature are fused by an information combination trick:

$$P(y=1|X^+) = (1-k)P_{r1} + kP_{d1}, \quad (13)$$

$$\text{and } k = \frac{\delta_{r1}}{\delta_{r1} + \delta_{d1}}$$

where $\{\delta_{d1}, \delta_{r1}\}$ is respectively the variances from the conditional distributions in two classifiers $H_{rk}(\cdot)$ and $H_{dk}(\cdot)$. For computational simplicity, we compute $P(y=0|X^-)$ with similar rules.

At last, we use an efficient ensemble criterion (10) to select K weak classifiers from the pool $\Phi = \{h_1, \dots, h_M\}$ similar to WMIL [8] tracker. And the strong classifier $H_K = \sum_{k=1}^K h_k$ is constructed by K weak classifiers on the depth image and RGB image respectively.

3.2. Multi-scale RGB-D Tracker

Our particle filter tracker also consists of an initialization of template model and a sequential Monte Carlo implementation of a Bayesian filtering for the stochastic tracking system [14-16]. In each iteration, the particle filtering tracking algorithm includes two steps: the prediction stage and the update stage.

In our algorithm, the state vector of the particle filter is defined as $x_t = (x1_t, x2_t, s_t)$, where $x1_t, x2_t$ represent the location of samples, s_t represents the scale in both $x1$ row direction and $x2$ column direction.

In the prediction stage, the samples in the state space are propagated through a state equation [15] in the RGB image and depth image. The dynamics usually is from an auto-regressive process (AR) in [15]. We use the first-order AR process for comparison and simplicity, that is

$$x_t(i) = x(i)_{t-1} + w_t, \quad (14)$$

where $x_t(i)$ denotes the state vector of the i -th particle at time step t and $w_t \sim N(0, \delta)$ i.e. the normal Gaussian distribution.

In the update stage, a common observation model [15] is used to estimate the weights for each sample.

$$p(y_t | x_t^*(i)) \propto \exp(H_K(x^{(i)})) = \sum_{k=1}^K h_k(x^{(i)}), \quad (15)$$

where y_t denotes the label of the tracked target, $H_K(x^{(i)})$ is a strong classifier and then applied to the i -th particle with the discriminative score. The classifier is also updated by maximal likelihood estimation (5)(6).

Then the weights for the i -th particle at time step t can be obtained by the following formula:

$$w_t^i = \frac{p(y_t | x_t^*(i))}{\sum_{j=1}^N p(y_t | x_t^*(j))}. \quad (16)$$

In order to avoid the degeneracy, the prior particle set should be re-sampled according to the weights of the particles obtained by the observation model. The re-sampled process retains particles whose confidence is relative high and the distributions of the particles are reallocated. Since those tracking algorithms based on 2D features only achieve one result box, the error accumulation will be serious when the candidate sample is deviated with target. Moreover, the color and depth images contain different information about the scene, so a more robust tracking algorithm is implemented by combining these information, i.e. Haar-like and Haar-D. These re-sampled particles will work together to find the particle whose confidence are relative highest on the depth image as a result box b . Then, we also obtain another result box c on the RGB image with similar ways. By exploiting the distance between center points, we compute the distance between result box and afore-result box as follows:

$$d(b, c) = \|b(cx, cy), c(cx, cy)\|_2, \quad (17)$$

where b is afore-result box, c is result box which is from depth image or color image, and $\{cx, cy\}$ denote the center coordinates of result box.

As shown in Eq.(17), the new object location is found with a smaller distance d . It reduces the error accumulation and recovers to a good performance when the candidate sample is not so well.

3.3. Summary of the Tracking Algorithm

Summarizing the above analysis, our multi-scale RGB-D tracking algorithm overcomes the limitation of fixed-size tracking box via the multi-scale bootstrap filter, and improves the location accuracy rate and tracking success rate by fusing Haar-like features and Haar-D features with respectively corresponding to RGB image frames and depth image frames. So our algorithm is a robust and effective target tracking method which stably track the specified target with significant variation of the target's scale, pose and plane rotation, shown in our experiment system. In addition, the scale is updated by particle filter (See Step 1 in Algorithm 1), that can lead a rational tradeoff between computational efficiency and effectiveness of handling appearance change caused by fast scale change. In the following, we provide the basic procedures of the proposed multi-scale RGB-D tracking algorithm in Algorithm 1.

Algorithm 1. Multi-Scale Tracking via online RGB-D multiple instance learning (i.e. **Multi-scale RGB-D Tracker**)

Input: t -th video frame

1. Use the 2-order auto-regressive model (14) to renewal all particles information. And then we use the observation model (15) to compute the weight for each particle in (16).
2. Find the particle with maximum confidence on the depth image and RGB image, and compute the distance (17) between afore-result box and candidate result box. Then find the new object location which is the location of the candidate result box with the smaller distance.
3. Re-sample and retain particles whose confidence is larger than the given threshold value.
4. Sample two sets of image patches as positive samples and negative samples on each of the depth image and RGB image.

$$D^{\alpha} = \{z \mid \|l(z) - l_t\| < \alpha\}, \quad D^{\xi, \beta} = \{z \mid \xi \leq \|l(z) - l_t\| < \beta\}, \quad \alpha < \xi < \beta.$$

where α, β are the two search radius of the candidate target patch.

5. Extract the Haar features of these sets and update the classifier parameters according to (5), (6).
6. Compute $P(y_t = 1 \mid X^+)$ and $P(y_t = 0 \mid X^-)$ in (13). And then use an efficient criterion (10) to select K weak classifiers on the depth image and the RGB image respectively.

Output: Tracking location l_t .

4. Experiments

This section evaluates the effectiveness and efficiency of the proposed Multi-scale RGB-D Tracker. We have implemented it in MATLAB and carry out a series of test experiments on a PC with an AMD 3.10GHz GPU and 4GB RAM. Experimental results are compared it with 4 state-of-the-art trackers on 6 benchmark video sequences (<http://vision.cs.princeton.edu/projects/2013/tracking/dataset.html>), which contain some challenging factors (drastic illumination change, pose and scale variation, heavy occlusion and background clutter). These video data were captured by using a standard Microsoft Kinect. The compared 4 trackers are the Compressive tracker (CT) [7], the Structured Output Tracking (Struck) [18], the Multiple Instance Learning tracker (MIL) [6], the Online Discriminative Feature Selection tracker (OFDS) [17].

4.1. Experimental Setup

Given a initial target location at the current frame, the search radius for drawing positive samples is set to $\alpha = 4 \sim 7$ which generates 45-190 positive samples. Some sequences in challenging as target is heavily occluded and pose variation, we found the results are robust with different α . Especially when the appearance changes very fast, we need a large α to capture more positive samples. The inner and outer radius for the set $X^{\xi, \beta}$ that generates negative samples are set to $\xi = 2\alpha$ and $\beta = 38$. These setting can crop negative samples with less overlap with positive samples. Then, we select 42-100 negative samples. We found the results are robust with different numbers of selected negative samples. The initial standard deviation for the status components $x1_t, x2_t, s_t$ of 200 particles is 6.0, 1.5, 0.04. Our multi-scale RGB-D tracker select $k = 15 \sim 30$ features to design the classifier, and the learning parameter can be set as $\eta = 0.76 \sim 0.92$. From (5) and (6), we can see that a smaller learning rate can make the tracker quickly adapts to the fast appearance and larger rate can reduce the likelihood that the tracker drifts off the target.

4.2. Experimental Results

Two evaluation criterions are used in our experiments: center location error and tracking success rate. The center location error is defined as the Euclidean distance from the detected object center to the ground truth center at each frame. The maximum, mean and standard deviation of the center location error per frame are summarized in Table 1. And the success rate is defined as $score = area(B_T \cap G_T) / area(B_T \cup G_T)$, where B_T and B_G are the bounding boxes of the tracker and of the ground-truth, respectively. When the

overlap ratio is larger than 0.5, the tracking result of the current frame is considered as a success. Table 2 presents the tracking results in terms of success rate. Fig. 2 plots the frame-by-frame center location errors obtained by the 5 trackers for the 6 video sequences. Table 1, Tabel 2 and Fig. 2 show that our Multi-scale RGB-D tracking algorithm achieves the best or second best performance in most sequences in terms of success rate and center location error. Furthermore, the proposed Multi-scale RGB-D tracker performs well in terms of speed (only slightly slower than CT and STC method) among all the evaluated algorithms on the same machine even though other trackers. Figure 3 shows screenshots of some tracking results.

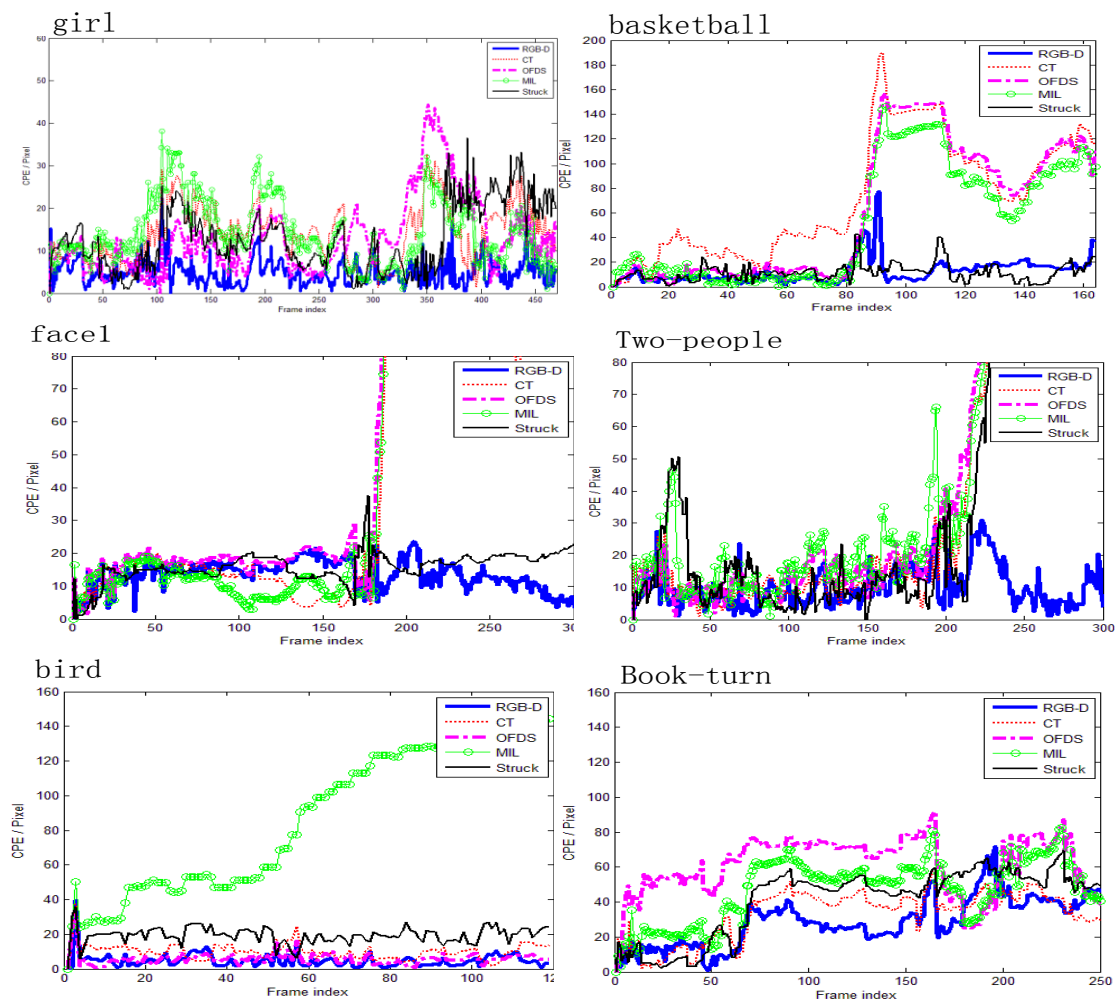


Fig. 2. Quantitative comparisons of the 5 trackers with the center location error on the 6 video sequences.

Scale, pose and illumination variation. Compare with other tracker, our tracker is able to handle scale and orientation change well due to the use of Haar-like feature, Haar-D feature and Bootstrap filter (PF). We evaluate sequences with different kind of scale and illumination variations. The targets in *girl* (#209, #357, #426 in Fig. 3) and *basketball* (#9, #87, #157 in Fig. 3) sequences undergo large appearance change due to scale and pose variation. Our tracker achieves the best or second best performance in most sequences. The Struck method is able to adapt the scale and pose variation well, but does not perform well in *girl* sequences due to illumination variations. The CT, OFDS and MIL method gradually drift away due to large appearance change caused by occlusion, scale and pose variation in *basketball* sequences.

Occlusion and Pose variation. The targets are partially occluded in the *face1* and *Two-people* sequences. The CT, MIL and OFDS method do not perform well as shown in Fig. 3 due to large appearance change caused by occlusion (#38, #176, #184 in Fig. 3). The *Two-people* sequence is challenging as target is heavily

occluded and pose variation (#33, #187, #258 in Fig. 3). Only the proposed multi-scale RGB-D tracking algorithm is able to relocate the target after heavy occlusion and pose variation as it use RGB-D features. As some regions are updated at any time instance by the proposed method, the tracking drift problem can be better handled where heavy occlusion occurs.

Table 1. The Average Center Location Errors of the 6 Video Sequences

Method Location error Video	RGB-D Tracker	CT [7]	OFDS[18]	MIL [6]	Struck[17]
face1	12.9	64	73.8	64.2	16.01
Two-people	9.27	83	86.9	89.3	81.9
Bird	2.8	13.4	5.5	143	23
Book-turn	22	32	56	43.4	38
Basketball	12.8	70.3	60.6	51.6	10.01
Girl	4.9	13.8	13	14.2	12
Avg-CPE	14	46	52	69	30.1

Table 2. The Success Rate (%) and the Average Frames Per Second (FPS) of the 6 Sequences

Method Success rate Video	RGB-D Tracker	CT [7]	OFDS [18]	MIL [6]	Struck [17]
face1	65	34	28	32	60
Two-people	72	23	21	15	38
Bird	82	54	73	0	30
Book-turn	74	51	37.2	45	48.2
Basketball	70	28	39	42	71
Girl	82	70	71	68.2	72
Avg-CPE	72	44	43	34	55



Fig. 3. Tracking results of 5 trackers (denoted in different colors) on 6 video sequences. Frame numbers are overlaid in yellow.

5. Conclusion

In this paper, we propose a novel multi-scale tracking algorithm which combines the fused 3D features and the bootstrap filter. Multiple instance learning strategy is adopted to fuse 3D information. The proposed scale invariant box was adopted to characterize the target object of different scales to be tracked, which is effective, simple and robust for scale variation or in-variation targets. We extensively analyze the performance of our tracker on real-word video sequences and show that it outperforms 4 state-of-the-art tracking methods. We can see that our tracker is robust for scale change, plane rotation and partial occlusion. Our future work will focus on applications of the developed algorithm for object detection and

recognition under abrupt motion. In addition, we will explore efficient detection models for persistent tracking (where objects disappear and reappear after a long period of time).

Acknowledgments

The authors would like to thank the anonymous reviewers for their help. This work was supported by the State Key Program of National Natural Science of China(Grant No. U1401252), the National Natural Science Foundation of China (Grant no. 61379114, 61203308) and the Chongqing Natural Science Foundation (No. cstc2015jcyjA40036).

References

- [1] Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4), 1-45.
- [2] Collins, R. T., Liu, Y., & Leordeanu, M. (2005). Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1631-1643.
- [3] Grabner, H., & Bischof, H. (2006). On-line boosting and vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 260-267).
- [4] Grabner, H., Grabner, M., & Bischof, H. (2006). Real-time tracking via on-line boosting. *BMVC*.
- [5] Avidan, S. (2005). Ensemble tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 494-501).
- [6] Babenko, B., Yang, M. H., & Belongie, S. (2011). Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1619-1632..
- [7] Zhang, K. H., Zhang, L., & Yang, M. H. (2012). Real-time compressive tracking. *Proceedings of the European Conference on Computer Vision* (pp. 864-877).
- [8] Zhang, K. H., & Song, H. (2013). Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*, 46(1), 397-411.
- [9] Luber, M., Spinello, L., & Arras, K. O. (2011). People tracking in RGB-D data with on-line boosted target models. *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3844-3849).
- [10] Spinello, L., & Arras, K. O. (2011). People detection in RGB-D data. *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3838-3843).
- [11] Spinello, L., Luber, M., & Arras, K. O. (2011). Tracking people in 3d using a bottom-up top-down detector. *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1304-1310).
- [12] Wang, J., Cao, J., & Wu, D. *et al.* (2012). An object tracking algorithm based on the current statistical model and the multi-feature fusion. *Journal of Software*, 7(9), 2000-2008.
- [13] Zhou, Z., Wu D., & Peng, X. *et al.* (2014). Object tracking based on camshift with multi-feature fusion. *Journal of Software*, 9(1), 147-153.
- [14] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 1-511).
- [15] Meng, J., Liu, J., & Wang, J. *et al.* (2013). Target tracking based on optimized particle filter algorithm. *Journal of Software*, 8(5), 1140-1144.
- [16] Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2), 107-113.

- [17] Zhang, K. H., Zhang, L., & Yang, M. (2013). Real-time object tracking via online discriminative feature selection. *IEEE Transactions on Image Processin*, 22(12), 4664 – 4677.
- [18] Hare, S., Saffari, A. & Torr, P. H. S. (2011). Struck: Structured output tracking with kernels. *Proceedings of the International Conference on Computer Vision* (pp. 263-270).



Xianhua Zeng was born in 1973, Sichuan province, China. He received his PhD degree in computer software and theory from Beijing Jiaotong University in July, 2009. He is currently an associate professor and master supervisor at the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, in China. His main research interests include computer vision and machine learning.



Yipeng Gao was born in 1989, Gansu province, China. He is an enrolled postgraduate of Chongqing University of Posts and Telecommunications. His main research interests is target tracking in computer vision.

Suli Hou was born in 1987, He'nan province, China. She is an enrolled postgraduate of Chongqing University of Posts and Telecommunications. Her main research interests is intelligent information processing.

ShuWen Peng was born in 1987, Chongqing province, China. She is an enrolled postgraduate of chongqing university of Posts and Telecommunications. Her main research interests include machine learning and image super-resolution.