Model-Free Multiple Object Tracking with Shared Proposals

Gao Zhu[1], Fatih Porikli[1,2,3], Hongdong Li[1,3]

Australian National University[1], Data61/CSIRO[2],
ARC Centre of Excellence for Robotic Vision[3]

Abstract. Most previous methods for tracking of multiple objects follow the conventional "tracking by detection" scheme and focus on improving the performance of category-specific object detectors as well as the between-frame tracklet association. These methods are therefore heavily sensitive to the performance of the object detectors, leading to limited application scenarios. In this work, we overcome this issue by a novel model-free framework that incorporates generic category-independent object proposals without the need to pretrain any object detectors. In each frame, our method generates a small number of target object proposals that are shared by multiple objects regardless of their category. This significantly improves the search efficiency in comparison to the traditional dense sampling approach. To further increase the discriminative power of our tracker among targets, we treat all other object proposals as the negative samples, i.e. as "distractors", and update them in an online fashion. For a comprehensive evaluation, we test on the PETS benchmark datasets as well as a new MOOT benchmark dataset that contains more challenging videos. Results show that our method achieves superior performance in terms of both computational speed and tracking accuracy metrics.

1 Introduction

Single object tracking attained considerable success thanks to the advances in "tracking-by-detection" that demonstrated improved performance on standard benchmarks [1,2,3]. Compared to single-object tracking counterpart, multiple-object tracking is a more challenging task due to the frequent occlusions between the target objects [4] and typical similarities in their motion patterns as well as visual appearances. Moreover, the background scenes also tend to be more cluttered due to the presence of other moving objects [5,3].

In model-based tracking-by-detection of multiple objects, an offline trained category-specific object detector, e.g., DPM [6] or R-CNN [7], is applied at every frame to generate high quality object hypotheses, and then graph-based methods such as max-flow [8,9] are used to solve the subsequent multi-frame multi-target association problem. These multiple object tracking methods, however, depend heavily on the performance of category-specific object detectors, which often miss objects or generate false positives that are induced by the discrepancy between the training dataset and the test conditions of individual deployments [10].

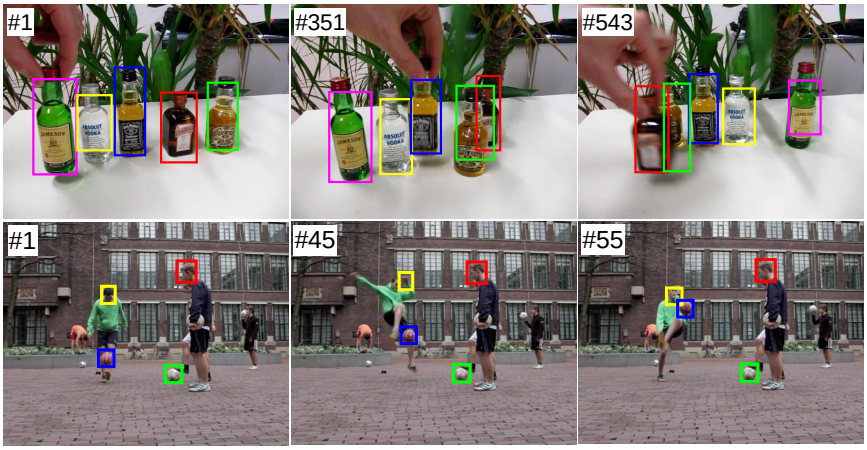


Fig. 1. Results obtained using our model-free multiple object tracking method. Bounding boxes of the same color denote the same tracked object. After initialization, our method tracks each object without any pretrained models.

Being constrained to a specific object class also limits the applicability of the tracker to a certain setting, for example, multiple vehicle tracking in traffic scenes. In practice, however, various applications demand tracking of different types of objects undergoing complex motions as shown in Fig. 1.

On the other end of the spectrum, "model-free" approaches aim to track arbitrary (category-independent) objects [11,12,13,14,15]. They initiate a single bounding box on the target in the first frame and then employ either a generative [16,17,18,19] or a discriminative [20,21,22,23] strategy to train their object models online. These methods are successfully applied for single-object tracking. However, extending "model-free" methods to multiple tracking task is not a straightforward problem due to two major reasons:

- Computational efficiency – Since each tracker searches around the previous location to localize the object, the time cost is proportional to the number of objects.
- Interactions – Objects contact or occlude each other. They often have similar appearances. Blindly and independently applying single-object trackers multiple times for different targets leads to ambiguities and tracking failures.

To overcome the above challenges, we propose a model-free multiple object tracking framework based on generic object proposals. We take advantage of the proposals in both online training and testing of the tracker.

In the testing stage, a small set of object candidates are generated based on simple objectness cues first. Notice, this set is shared by *all* trackers and it provides two benefits: i) a significant reduction of the number of candidates, and ii) tracking accuracy improvement since many false positives can be eliminated

at this stage. The proposals are then assigned to trackers based on the classifier confidence and temporal smoothness measures. The number of proposals can be as many as hundreds while the number of objects might be only a few. We use the Hungarian algorithm [24,23] with appropriate modifications to reduce the computational cost during the data association stage. Other association methods [1,2,3] can also be used, yet we observe that the computationally efficient Hungarian method works favorably when we build discriminative classifiers based on the generated proposals.

In the training stage, we collect the proposals as *hard* negative samples instead of manual selecting around positive samples. These proposals are expected to contain the other targets and object-like background clutter. Mining explicitly for such hard negative samples and employing hard negatives in the training of individual object models significantly improves the discriminative power of the object models. We update the classifiers at certain time intervals in an online fashion to compensate for object appearances changes over time and incorporate *new* distractors. A few local candidates sampled around the previous object locations are included in the negative set to further improve tracking precision.

We focus on a challenging scenario of multi-object tracking where each object may move **very fast** in an **irregular** fashion. To our knowledge, this challenge has not been widely researched and there are only a few benchmarks (e.g. PETS [5]) available for investigation. Therefore, we collected an extensive set of challenging video sequences from various sources and manually labeled the ground-truth object locations for a comprehensive experimental evaluation.

Our method is conceptually simple, easy to implement, and most importantly, achieves superior performance in comparison to several state-of-the-art techniques in terms of both tracking accuracy metrics and computational efficiency.

2 Related Work

Here we give a brief review to previous methods for multi-object tracking that are most related to this paper. For more comprehensive literature surveys the reader is referred to [11,12,13,3].

Multiple Target Tracking

As aforementioned in Section 1, most multiple object tracking methods focus on the data association problem, assuming sufficiently long and accurate tracklets are provided by using advanced object detectors [3]. For example, [25] considers motion dynamics as the major cue to distinguish different targets with similar appearance. It solves the problem as generalized linear assignment (GLA) of tracklets, which are incrementally joined forming longer trajectories based on their similar dynamics. The work in [1] observes that motion cues are not always reliable for this task, due to for example abrupt camera movement. As a remedy a structured motion constraint between objects is therefore proposed to address this issue.

Tracker in [2] proposes an online discriminative appearance learning approach to handle similar appearances of different objects in tracklet association. This method is similar to our method to be described in this paper; however, in their work those negative training samples are only collected around the tracklets, while ours pivots on the hard negative ones.

Model-Free Object Tracking

Model-free object tracking algorithms are proposed primarily for solving single object tracking applications [11,12]. The work in [26] tries to improve the identification of a single target object by also tracking stable features in the background, thereby improving the location prior for the target object. [27] proposes a context-aware tracker which considers a set of auxiliary objects as the contextual information for the foreground. These auxiliary objects must satisfy conditions such as having persistent co-occurrence with the foreground and consistent motion correlation.

The tracker in [28] is probably the most closely related work to ours. However, they assume spatial relationship between objects. For instance, nearby objects tend to move along the same direction. The appearance models of all the objects and the structural constraints between these objects are jointly trained in an online structured support vector machine framework. Our framework has no such an assumption and can track arbitrarily moving objects.

Object Proposals for Visual Tracking

As reported in [29,30], using object proposal improves the object detection benchmark along with the convolutional neural nets. Since, a subset of high-quality candidates are used for detection, object proposal methods boost not only the speed but also the accuracy by reducing false positives. The top performing detection methods [31,32] for PASCAL VOC [33] use detection proposals. Among the existed proposal methods, the EdgeBox method [30] proposes object candidates based on the observation that the number of contours wholly enclosed by a bounding box is an indicator of the likelihood of the box containing an object. It is designed as a fast algorithm to balance between speed and proposal recall, comparing to BING [34] and region proposal network (RPN) [7].

Many work exist adopting the object proposals for the model-free single object tracking. A straightforward strategy based on linear combination of the original tracking confidence and an adaptive objectness score obtained by BING is employed in [35]. In [36], a detection proposal scheme is applied as a post-processing step, mainly to improve the tracker's adaptability to scale and aspect ratio changes. EBT [37] employs the EdgeBoxes method to globally track the object, disregarding potentially fast or drastic object motion. In contrast, our work utilizes the shared proposals for efficient handling of multiple trackers. [38] deals with generic object tracking for street scenes by generating multi-scale candidates from the point-density map. Tracking is performed using the pseudo-Boolean optimization (QPBO) method. In comparison, our method is applied to more generic object categories rather than street scenes. Besides, our object models is built taking advantage of the proposals, while [38] adopts a generative model using RGB feature distance.

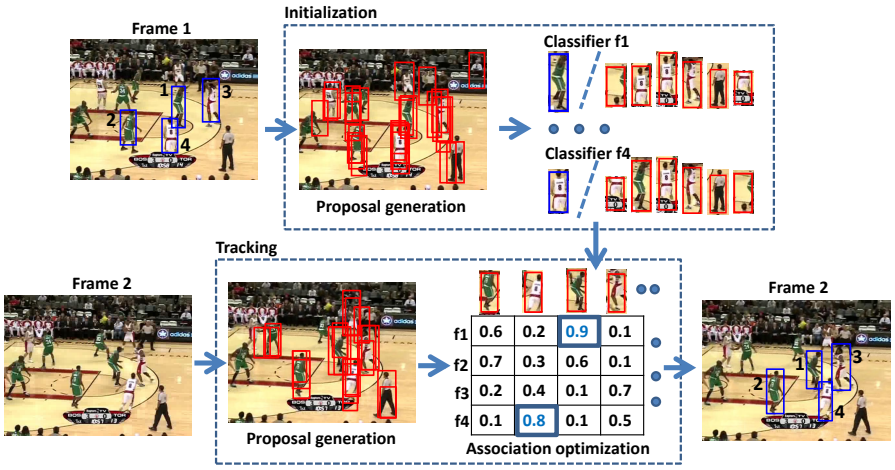


Fig. 2. The structure of our model-free multiple object tracker. The only input is the bounding boxes at the first frame. Our method then initializes multiple classifiers for each object taking advantage of a small set of object proposals generated from the frame. In the next frame, these classifiers are used to assign confidence scores for the candidate proposals. The final trajectories are obtained after solving the optimal association problem. Note that, we also apply the proposals to online update the classifiers to make them more robust to distractors.

3 Multiple Object Tracking with Proposals

As illustrated in Figure 2, our framework starts with a few manually initialized bounding boxes on the target objects to be tracked in the first frame of the video. This is similar to the single object online visual tracking task [11,12,13]. Given these initial bounding boxes, denoted as $\{B^i_{t=1}\}$, $i = 1, \ldots, N_o$, where N_o is the total number of objects, the multiple object tracking problem then aims to find the locations and bounding boxes of the multiple objects in the remainder of the video while maintaining the correct identity of each object.

Following the tracking-by-detection framework, we train the object appearance models for each object. We have an option to use either the generative or discriminative learning strategy. Recent literature on object tracking resort to the discriminative learning to maximize the inter-class separability between the object and background regions and report improved performance as the discriminative learning is more robust to distractions from the background. This property is especially important in multiple object tracking [2,39] where the objects exhibit similar appearance and interact frequently, as depicted in Fig-2.

As explained in Section 1, we do not independently initialize N_o classifiers by collecting locally and densely sampled negative patches as training samples, a scheme that conventional online single object trackers typically employ.

Instead, we incorporate object proposals [29,30] to generate a small number of shared object candidates. Notice that, we are not simply using the original object proposals either, since the sizes of the objects usually change during the tracking. We impose the proposal bounding box sizes to be within a certain range of the object sizes. More details about this can be found in Section 3.1.

Suppose the object proposal bounding boxes are $\{\hat{B}^j_{t=1}\}$, $j = 1, \ldots, N_p^{t=1}$, where $N_p^{t=1}$ is the total number of proposals in the first frame. We train the classifiers with the corresponding positive samples $B^i_{t=1}$ that are not in the common negative set $\{\hat{B}^j_{t=1}\}$. The initialized classifiers are denoted as

$$f^i_{t=1}(B), \quad i = 1, \ldots, N_o, \tag{1}$$

We additionally select a small set of local candidates sampled around the object to further improve the discriminative power, thus the localization precision, of the classifier as [37].

In the consecutive frame, we generate a set of proposals $\{\hat{B}^j_{t=2}\}, j = 1, \ldots, N_p^{t=2}$, to be shared and tested by all classifiers $\{f^i_{t=1}(B)\}$. Considering the temporal smoothness between the object $B^i_{t=1}$ and the proposal $\hat{B}^j_{t=2}$, (spatial distance between them), we build an association matrix that will be efficiently optimized by a modified Hungarian algorithm [24,23]. The new object locations are then determined as the optimal solution of this association problem. More details about it can be found in Section 3.2.

To adapt the object appearance changes as well as to increase the discriminative power of the classifiers against newly appeared distractors, we incrementally update the classifiers by treating the estimated bounding box in current frame as the positive sample and object proposals as the negative samples as we did in the first frame. More information is in Section 3.3.

3.1 Object Proposal Generation

As mentioned in Section 2, various object proposal algorithms exist. We employ EdgeBox [30] as it strikes a good balance between recall and speed. In our experimental analysis, we also test other proposal methods such as BING [34] and region proposal network (RPN) [7].

Two important factors should be noticed here. The first one is the about the sizes of the generated object proposals, termed as size adaption ratio and denoted as $\alpha \in [0, 1]$. We allow the size of the proposals maximally differ the target with a bounding box intersection-over-union (IoU) [33] of ratio α. To be specific, we consider $\hat{B}^j_t$ only when

$$\max_i(\text{IoU}(\hat{B}^j_t, B^i_{t-1},)) > \alpha, \quad i \in [1, \ldots, N_o] \tag{2}$$

This setting significantly reduces the number of proposals while permitting the object window to adapt the target size changes at the same time. We use $\alpha = 0.8$ and test other values in the experimental part.

The other factor is the maximal number of object proposals generated. EdgeBox does not output a fixed number of proposals. The number of proposals could be any depending on the threshold of the "objectness" score (set as 0.01 as recommended). An appropriate maximal number of proposals needs to be used as its lower values may result in missing the object window in the proposal set while its higher values would cause an extensive number of distractors. We set this number at 500 for all experiments. We also run test other values of the maximal number of proposals in Section 4.2.

Similar to [37], we generate a fixed number of bounding boxes, $\{\tilde{B}_t^k\}_{t-1}^i$, $k = 1, \ldots, N_s$, by sampling only around the previous object location B_{t-1}^i for each object (as in traditional methods). This set $\{\tilde{B}_t^k\}_{t-1}^i$ is only tested by the corresponding classifier $f_{t-1}^i(B)$ and they are useful to smoothen the trajectory as the object proposal component works independently at each frame, which may result in temporally inconsistent proposals. Thus, a combined set of $\{\hat{B}_t^j\} \cup \{\tilde{B}_t^k\}_{t-1}^i$ is used during the test stage for the classifier $f_{t-1}^i(B)$. However, we only update the classifier when the estimated one comes from the proposal set $\{\hat{B}_t^j\}$ to attain resistance to potential corruptions. We sample $N_s = 80$ patches uniformly within a 30-pixels radius. More details are in Section 3.3.

3.2 Optimal Target Association

Given N_o targets and $(N_p^t + N_s \times N_o)$ candidates, the target association stage therefore aims to find the optimal non-repetitive N_o candidates for the N_o targets, such that the overall *gain* is maximized. Note that, the candidates $\{\tilde{B}_t^k\}_{t-1}^i$ are only allowed to link with target i, thus we set the *gain values* of linking them to other targets to zero.

The gain value $P(B_t, i)$ of linking a candidate B_t to target i is designed base on both classifier confidence score and temporal smoothness,

$$P(B_t, i) = f_{t-1}^i(B_t) + s(B_t, B_{t-1}^i). \tag{3}$$

$s(B_t, B_{t-1}^i)$ is a term representing the temporal smoothness between the previous target bounding box B_{t-1}^i and the candidate box B_t. We use a simple function in this paper: $s(B_t, B_{t-1}^i) = \exp(-\frac{1}{2\sigma^2}\|c(B_t) - c(B_{t-1}^i)\|^2)$, where $c(B_t)$ is the center of bounding box B_t and σ is a value controlling the impact of the temporal smoothness term. We set $\sigma = R_i$, where R_i is half of the diagonal length of the initialized bounding box B_1^i. We also test other values as in Section 4.2.

Once the gain values are set, the standard Hungarian algorithm [24,23] can be modified to optimally solve the association problem. As $(N_p^t + N_s \times N_o)$ is usually much larger than N_o (a few hundreds vs. a few), available fast implementation [40] is too slow to be applied directly. We thus firstly find top N_o candidates for each target i locally and separately. As the global optimal assignment for that target i must be one of them, we then combine those found local candidates into a small matrix in which the optimal solution is exact the same global optimal solution to the original association problem. Notice that, the standard Hungarian algorithm solves the minimization problem, thus a simple modification is required before feeding the small matrix to it.

3.3 Online Updating with Proposals

To update the classifier, $f_{t-1}^i \rightarrow f_t^i$, we also generate a few local samples, $\{\tilde{B}_t^k\}_t^i$, $k = 1, \ldots, N_s$, around the estimated object location B_t^i. They are helpful to increase the discriminative power of the classifier, as the object proposals alone represent other good "object-like" regions and training with them increases the discriminative power among "objects-like" candidates, while the negative sample space contains a lot more other negative samples, thus more negative samples help. The updating procedure is applied every 5 frames to balance computational time and minimize potential drift.

As mentioned in the last paragraph of Section 3.1, we treat the estimated result B_t^i as an indication for model updating. This is to say, when $B_t^i \in \{\tilde{B}_t^k\}_{t-1}^i$, we assume that there is no good object proposal and the current estimation is a compromise for trajectory smoothness, thus skipping the model updating. If $B_t^i \in \{\hat{B}_t^j\}$, then it suggests a good estimation which has both desirable classifier response and high "objectness", then we update the object model $f_{t-1}^i(B)$ immediately.

3.4 Proposed Tracker: PMOT

Various object models can be integrated into our framework. We choose a popular structured support vector machine (SSVM) method [41], as it shows good performance on several benchmarks [11,12]. The tracker is denoted as PMOT to reflect the concepts of shared proposals and multiple object tracking.

Denote the support vector set trained in the SSVM as $\mathcal{V}_{t-1}$, the classification function can then be expressed as a weighted sum of affinities between the candidate bounding box and the support vectors [42,41]:

$$f_{t-1}^i(B_t) = \sum_{\bar{B}^m \in \mathcal{V}_{t-1}} w^m k(\bar{B}^m, B_t), \quad m = 1, \ldots, |\mathcal{V}_{t-1}| \tag{4}$$

where w^m is a scalar weight associated with the support vector $\bar{B}^m$. Kernel function $k(\bar{B}^m, B_t)$ calculates the affinity between two feature vectors extracted from $\bar{B}^m$ and B_t respectively. The classifier is updated in an online fashion using [43,44] with a budget [45]. Intersection kernel is used and other parameters are set same as [41]. We use histogram features obtained by concatenating 16-bin intensity histograms from a spatial pyramid of 5 levels and RGB channels separately. At each level L, the patch is divided into $L \times L$ cells, resulting in a 2640-D feature vector.

4 Experiments

4.1 Full Benchmark Evaluations

To evaluate the performance of the proposed multiple object tracking method, we collect 10 videos from various sources, including TB50[15], OTB [11] and VOT2015 [13]. We denote this dataset as MOOT (Multiple Object Online Tracking) and a few samples can be seen in Figure 5. The number of targets in these

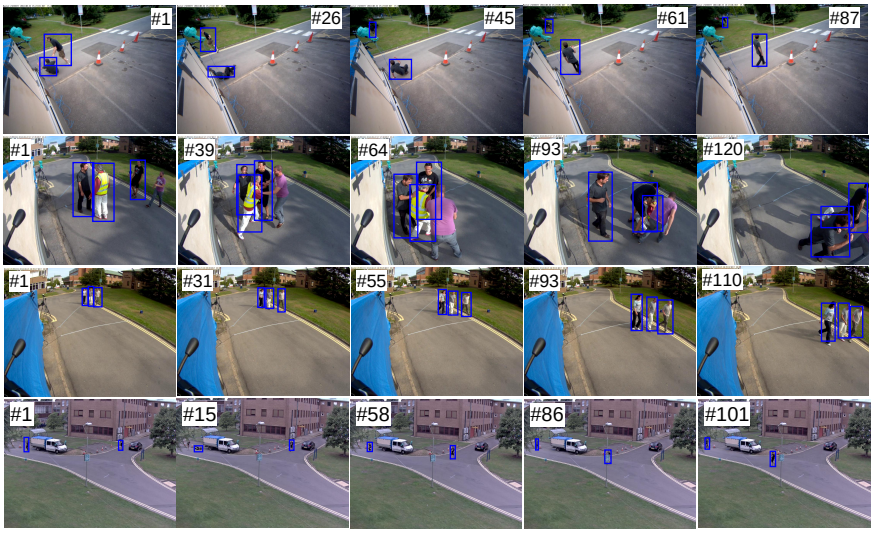


Fig. 3. Sample sequences from the PETS benchmark dataset [5] with ground truth object windows (blue).

videos ranges from 2 to 5. This dataset contains extremely challenging scenarios, including repetitive mutual occlusion (videos "liquor" and "skating2") and similar appearance among the targets (videos "bolt1", "bolt2", "football" and "basketball").

We also evaluate the proposed method on the video sequences from Performance Evaluation of Tracking and Surveillance (PETS) 2015 [5]. These videos are from surveillance cameras and all targets are humans. We list the details of the four sequences in Table 1 with corresponding challenges featured. As we can see, all sequences contain challenging aspects, while video "A1_ARENA-15_06_TRK_RGB_2" (row 2 in Figure 3) is the most difficult one containing both deformation and occlusion challenges.

Compared Trackers and Evaluation Metrics. Our method (PMOT) is compared with several state-of-the-art methods. Specifically, we compare our method with SPOT [28] which addresses a similar task as ours and it deploys a structure preserving model. We also compare with several single online object trackers to corroborate the point that by sharing and building discriminative classifiers based on proposals, our method is more robust to drifting. MEEM [20], KCF [22] and Struck [41] are three top-ranked trackers in recent large benchmarks [11,12,46,15] for single online object tracking. For all the trackers, we use their default settings and separately initialize on each object for each video. We also modify the PMOT for the single object case, denoted as PMOTsingle. This allows us to precisely analyze the improvement of adopting the proposal sharing scheme, in term of both the tracking metrics and computational efficiency.

Table 1. Attributes of the four video sequences from the PETS dataset.

Video	#humans	#frames	Challenge
N1_ARENA-01_02_TRK_RGB_2	3	115	Size change
W1_ARENA-11_03_ENV_RGB_3	2	107	Body deformation
W1_ARENA-11_03_TRK_RGB_1	2	101	Body deformation
A1_ARENA-15_06_TRK_RGB_2	3	121	Occlusion and body deformation

Table 2. Area Under Curve (AUC) of *success plot* and *precision score* (PS) with 20 pixels threshold on the MOOT dataset for the one-pass evaluation (OPE). Cell values: AUC/PS

MOOT	PMOT	PMOTsingle	SPOT [28]	MEEM [20]	KCF [22]	Struck [41]
ball1	**66.2**/99.0	66.0/**99.3**	30.6/67.4	51.3/74.5	48.5/83.1	52.7/86.0
basketball	**61.5**/**84.0**	60.2/81.7	11.6/8.6	46.2/70.9	51.3/59.8	38.5/50.3
bolt1	**47.4**/**93.8**	36.6/71.6	0.5/0.5	23.5/50.6	34.3/70.6	33.9/73.8
bolt2	50.8/89.0	38.6/69.9	0.6/0.8	47.3/90.4	50.9/93.6	**57.4**/**97.7**
football	**62.0**/94.6	57.8/88.9	23.4/41.5	60.7/**97.0**	49.5/69.1	57.5/79.7
human4	60.7/93.5	34.5/48.5	**61.5**/**99.5**	57.4/91.2	50.2/75.7	62.7/94.7
jogging	**67.4**/**97.6**	63.8/89.7	12.3/13.5	60.6/88.4	15.5/19.9	15.0/19.7
liquor	**61.0**/**79.8**	41.6/51.0	32.8/38.2	10.6/16.8	18.8/24.6	7.2/8.9
skating1	56.5/71.2	46.5/55.4	55.5/78.4	62.2/**92.3**	**62.8**/89.6	35.9/50.0
skating2	**50.8**/**44.9**	48.1/43.7	34.6/25.8	35.9/28.4	33.7/37.1	26.7/18.2
Mean	**58.5**/**86.2**	49.5/71.5	23.7/34.1	41.7/67.7	40.5/61.6	37.5/61.4

We use the single online object tracking metrics to measure the tracking performance, similar to [28]. Evaluation metrics and code are provided by the benchmark [11,15]. We employ the one-pass evaluation (OPE) and use two metrics: *precision plot* and *success plot*. The former one calculates the percentage (*precision score*, PS) of frames whose center location is within a certain threshold distance with the ground truth. A commonly used threshold is 20 pixels. The latter one calculates a same percentage but based on bounding box overlap threshold. We utilize the area under curve (AUC) as an indicative measurement for it.

Experimental Setting. Our tracker is implemented using C++ and MATLAB, on an i7-2600 3.40 GHz desktop with a 8 GB RAM. For the EdgeBox proposal method and SSVM applied, we use the default setting recommended by the authors, except those specified otherwise. We further discuss some parameters in Section 4.2

Benchmark Results. The results are summarized in Figure 4 and Table 2. We can see that the SPOT tracker achieves undesirable results, significantly lagging behind other compared methods. In term of the PS metric, it is 27.3% worse than Struck, the second worst tracker. It is not particularly surprising though, as can be seen in Figure 5, where we draw the visual comparison between the

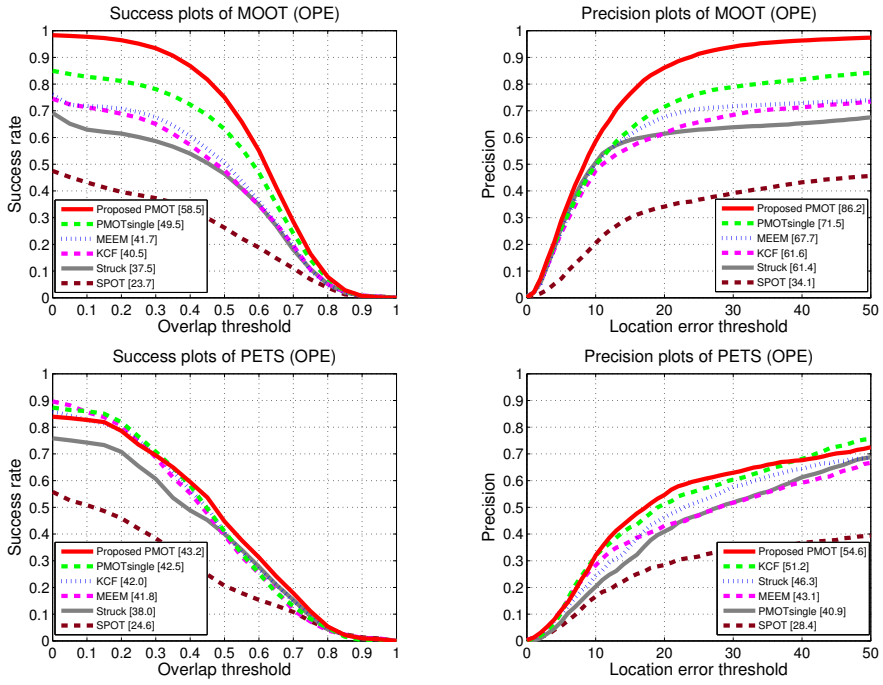


Fig. 4. *Success plot* and *precison plot* on two datasets: MOOT and PETS. Algorithms are ranked by the area under the curve (AUC) and the *precision score* (20 pixels threshold, PS). Our method achieves consistently superior performance, especially on the more challenging MOOT dataset.

proposed PMOT and SPOT. It clearly demonstrates that the SPOT tracker presumes a strong spatial structure exhibited among the objects, while it does not always hold. As shown in the video "bolt1" (row 1 in Figure 5), the four dash-line windows (SPOT) still maintain the relative positions while drifting away the true objects. In contrast, our method robustly and consistently tracks the objects even they are not moving coherently.

When comparing to the single object online tracking methods, the improvement is clearly shown. On the challenging MOOT dataset, our PMOT tracker outperforms the second best tracker by a large margin, with 9% and 14.7% in term of AUC and PS respectively. We can also see the clear advantage of applying the proposal based approach. Even the single object tracking variant, PMOTsingle, outperforms the best non-proposal tracker, MEEM, by 7.8% and 3.8% in AUC and PS respectively. This is partly contributed by the online updating strategy of collecting the proposals as hard negative samples to improve the discriminative power of the classifier, hence is robust to the distractions from other objects as well as potential distractors in the background.

For the PETS dataset, we can see that the improvement of PMOT is not great, outperforming the second best tracker, by 3.4% and 0.7% in the PS and AUC metrics, respectively. This is partly due to the fact that there is no significant interactions presented among the objects on PETS, except the video "A1_ARENA-15_06_TRK_RGB_2". Therefore, our proposed multiple object tracking system is unable to take a strong advantage of the proposal sharing benefit.

4.2 Further Remarks

Temporal Smoothness. The smoothness term $s(B_t, B^i_{t-1})$ (3) discussed in Section 3.2 controls the temporal consistency of the trajectory. This is especially important in our formulation as the object proposals are generated independently in each frame, which results temporal inconsistencies inevitably. We test different σ values and include the results in Table 3. We observe that a small σ leads to a strong smoothness constraint, which harms the performance when objects are occluded, while a large σ tends to result in unstable trajectories.

Size Adaption Ratio. The size adaption ratio α in (2) allows the target window to adapt the object size changes naturally once set properly. A smaller α leads to a larger set of object proposals with a more significant size variance, which harms both the computational efficiency and trajectory stability. We validate it with different values and results is in Table 3. It corroborates that a larger value is preferable, but the performance drops when $\alpha = 0.9$, as it constrains the sizes of object proposals too tight that it fails to adapt the object size changes.

Maximal Number of Object Proposals. We test 5 variants with the maximal object proposal number set at 200, 350, 500, 750 and 1000, respectively. The results are reported in term of AUC/PS metrics as included in Figure 6. As discussed in Section 3.1, using insufficient number of proposal leads to a bad coverage of the false positives as well as the object, while using a large number of proposals attracts spurious candidates.

Alternative Object Proposal Methods. We evaluate using other two popular object proposal methods, BING [34] and region proposal network (RPN) [7], instead of EdgeBox for proposals. Results are in Figure 6. Both performances are worse than the EdgeBox method. This is expected. As shown [29,30], BING results in a relatively low recall of the objects, while RPN performs undesirably for small-size objects.

Table 3. Area Under Curve (AUC) of *success plot* and *precision score* (20 pixels threshold) results of PMOT with different temporal smoothness constraints and size adaption ratios.

	Temporal Smoothness			Size Adaption Ratio		
	$\sigma = 0.5R_i$	$\sigma = R_i$	$\sigma = 2R_i$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
AUC	51.0	58.5	56.2	49.5	58.5	57.9
PS	72.3	86.2	84.1	70.5	86.2	84.9

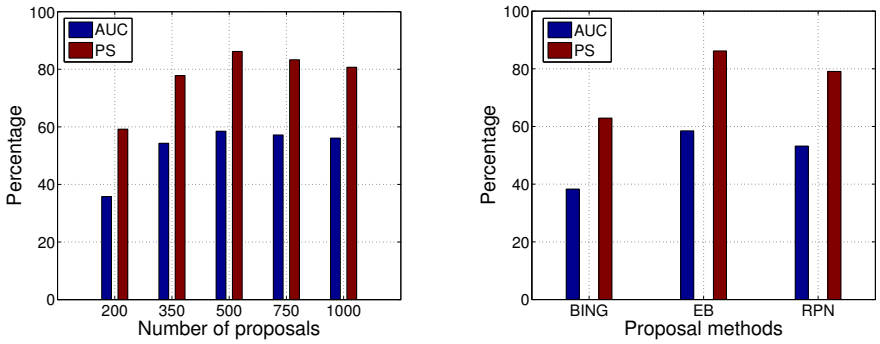
Fig. 5. Qualitative comparisons with the proposed PMOT tracker (solid lines) against the SPOT tracker (dash lines) on videos "bolt1", "ball1", "liquor", "bolt2", "football", "skating2" and "jogging" from MOOT dataset (from top to bottom). Our method exhibits robustness in challenging scenarios such as repetitive mutual occlusions and similar target appearances.

Failure Mode. Our method may not find every single object in every frame since we use object proposals as object candidates. Thus it may miss the object under, for example, extreme conditions (severe blur, distortion). Such miss detections, however, do not occur all the time. A temporary failure does not harm the overall performance since the model is incrementally and selectively updated.

Table 4. Processing times (frames per second, fps) of PMOT on videos containing different number of objects.

	PMOT				PMOTsingle
# of targets N_o	2	3	4	5	1
fps	4.1	3.3	2.6	1.9	5.3

Fig. 6. Area Under Curve (AUC) of *success plot* and *precision score* (20 pixels threshold) results of PMOT with different maximal numbers of proposals and various proposal methods.



Computational Efficiency. Since the object proposals are shared among the classifiers of multiple targets, we reduce the computational load by not repeating the proposal generation and feature extraction for each target. Table 4 shows the processing times (frames per second, fps) for different number of targets. We categorize the test videos according to the number of targets in them. For PMOTsingle, the number of targets is always 1. As we can see, our system is computationally efficient.

5 Conclusion

We proposed a computationally efficient and accurate model-free multiple object tracking method. It takes the advantage of the object proposals and generates a small and shared set of object hypotheses in the frame. Then it initializes multiple classifiers for each target using the shared set. In consecutive frames, the application and update of the classifiers are also achieved by using the detected proposals. We evaluated our method on both PETS and a newly introduced dataset. The results show superior performance against the state-of-the-art.

Acknowledgement. This work was supported under the Australian Research Councils Discovery Projects funding scheme (project DP150104645, DP120103896), Linkage Projects funding scheme (LP100100588), ARC Centre of Excellence on Robotic Vision (CE140100016).

References

1. Yoon, J.H., Lee, C.R., Yang, M.H., Yoon, K.: Online multi-object tracking via structural constraint event aggregation. In: CVPR. (2016) 1, 3
2. Bae, S.H., Yoon, K.J.: Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: CVPR. (2014) 1, 3, 4, 5
3. Milan, A., Leal-Taixé, L., Reid, I.D., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. CoRR (2016) 1, 3
4. Zhang, X., Hu, W., Qu, W., Maybank, S.: Multiple object tracking via species-based particle swarm optimization. IEEE Transactions on Circuits and Systems for Video Technology **20** (2010) 1590–1602 1
5. Li, L., Nawaz, T., Ferryman, J.: PETS 2015: Datasets and challenge. In: AVSS. (2015) 1, 3, 9
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010) 1
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015) 1, 4, 6, 12
8. Leal-Taixe, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: CVPR. (2014) 1
9. Milan, A., Roth, S., Schindler, K.: Continuous energy minimization for multitarget tracking. TPAMI (2014) 1
10. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR. (2011) 1
11. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR. (2013) 2, 3, 4, 5, 8, 9, 10
12. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. TPMAI (2014) 2, 3, 4, 5, 8, 9
13. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking VOT2015 challenge results. In: ICCVW. (2015) 2, 3, 5, 8
14. Felsberg, M., Berg, A., Hager, G., Ahlberg, J., Kristan, M., Matas, J., Leonardis, A., Cehovin, L., Fernandez, G., Vojir, T., et al.: The thermal infrared visual object tracking VOT-TIR2015 challenge results. In: ICCVW. (2015) 2
15. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. TPAMI (2015) 2, 8, 9, 10
16. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV (2008) 2
17. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: ICCV. (2009) 2
18. Li, H., Shen, C., Shi, Q.: Real-time visual tracking using compressive sensing. In: CVPR. (2011) 2
19. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR. (2012) 2
20. Zhang, J., Ma, S., Sclaroff, S.: MEEM: Robust tracking via multiple experts using entropy minimization. In: ECCV. (2014) 2, 9, 10
21. Zhu, G., Porikli, F., Ming, Y., Li, H.: Lie-Struck: Affine tracking on Lie groups using structured SVM. In: WACV. (2015) 2
22. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. TPAMI (2015) 2, 9, 10
23. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. TPAMI (2011) 2, 3, 6, 7

24. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society of Industrial and Applied Mathematics (1957) 3, 6, 7
25. Dicle, C., Camps, O.I., Sznaier, M.: The way they move: Tracking multiple targets with similar appearance. In: ICCV. (2013) 3
26. Duan, G., Ai, H., Cao, S., Lao, S.: Group tracking: Exploring mutual relations for multiple object tracking. In: ECCV. (2012) 4
27. Yang, M., Wu, Y., Hua, G.: Context-aware visual tracking. TPAMI (2009) 4
28. Zhang, L., van der Maaten, L.: Preserving structure in model-free tracking. TPAMI (2014) 4, 9, 10
29. Hosang, J., Benenson, R., Schiele, B.: How good are detection proposals, really? In: BMVC. (2014) 4, 6, 12
30. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. (2014) 4, 6, 12
31. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014) 4
32. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: ICCV. (2013) 4
33. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The Pascal visual object classes challenge: A retrospective. IJCV (2015) 4, 6
34. Cheng, M., Zhang, Z., Lin, W., Torr, P.H.S.: BING: binarized normed gradients for objectness estimation at 300fps. In: CVPR. (2014) 4, 6, 12
35. Liang, P., Liao, C., Mei, X., Ling, H.: Adaptive objectness for object tracking. CoRR (2015) 4
36. Huang, D., Luo, L., Wen, M., Chen, Z., Zhang, C.: Enable scale and aspect ratio adaptability in visual tracking with detection proposals. In: BMVC. (2015) 4
37. Zhu, G., Porikli, F., Li, H.: Beyond local search: Tracking objects everywhere with instance-specific proposals. In: CVPR. (2016,) 4, 6, 7
38. Ošep, A., Hermans, A., Engelmann, F., Klostermann, D., , Mathias, M., Leibe, B.: Multi-scale object candidates for generic object tracking in street scenes. In: ICRA. (2016) 4
39. Possegger, H., Mauthner, T., Bischof, H.: In defense of color-based model-free tracking. In: CVPR. (2015) 5
40. Cao, Y.: Hungarian algorithm for linear assignment problems (V2.3). http://www.mathworks.com/ (2008) 7
41. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV. (2011) 8, 9, 10
42. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: ECCV. (2008) 8
43. Bordes, A., Bottou, L., Gallinari, P., Weston, J.: Solving multiclass support vector machines with LaRank. In: ICML. (2007) 8
44. Bordes, A., Usunier, N., Bottou, L.: Sequence labelling SVMs trained in one pass. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD). (2008) 8
45. Wang, Z., Crammer, K., Vucetic, S.: Multi-class Pegasos on a budget. In: ICML. (2010) 8
46. M. Kristan et al.: The visual object tracking VOT2014 challenge results. In: ECCV Workshop. (2014) 9