Online Object Tracking: A Benchmark

Yi Wu
University of California at Merced
ywu29@ucmerced.edu

Jongwoo Lim
Hanyang University
jlim@hanyang.ac.kr

Ming-Hsuan Yang
University of California at Merced
mhyang@ucmerced.edu

Abstract

Object tracking is one of the most important components in numerous applications of computer vision. While much progress has been made in recent years with efforts on sharing code and datasets, it is of great importance to develop a library and benchmark to gauge the state of the art. After briefly reviewing recent advances of online object tracking, we carry out large scale experiments with various evaluation criteria to understand how these algorithms perform. The test image sequences are annotated with different attributes for performance evaluation and analysis. By analyzing quantitative results, we identify effective approaches for robust tracking and provide potential future research directions in this field.

1. Introduction

Object tracking is one of the most important components in a wide range of applications in computer vision, such as surveillance, human computer interaction, and medical imaging [60, 12]. Given the initialized state (e.g., position and size) of a target object in a frame of a video, the goal of tracking is to estimate the states of the target in the subsequent frames. Although object tracking has been studied for several decades, and much progress has been made in recent years [28, 16, 47, 5, 40, 26, 19], it remains a very challenging problem. Numerous factors affect the performance of a tracking algorithm, such as illumination variation, occlusion, as well as background clutters, and there exists no single tracking approach that can successfully handle *all* scenarios. Therefore, it is crucial to evaluate the performance of state-of-the-art trackers to demonstrate their strength and weakness and help identify future research directions in this field for designing more robust algorithms.

For comprehensive performance evaluation, it is critical to collect a representative dataset. There exist several datasets for visual tracking in the surveillance scenarios, such as the VIVID [13], CAVIAR [21], and PETS databases. However, the target objects are usually humans or cars of small size in these surveillance sequences, and the background is usually static. Although some tracking datasets [47, 5, 33] for generic scenes are annotated with bounding box, most of them are not. For sequences without labeled ground truth, it is difficult to evaluate tracking algorithms as the reported results are based on inconsistently annotated object locations.

Recently, more tracking source codes have been made publicly available, e.g., the OAB [22], IVT [47], MIL [5], L1 [40], and TLD [31] algorithms, which have been commonly used for evaluation. However, the input and output formats of most trackers are different and thus it is inconvenient for large scale performance evaluation. In this work, we build a code library that includes most publicly available trackers and a test dataset with ground-truth annotations to facilitate the evaluation task. Additionally each sequence in the dataset is annotated with attributes that often affect tracking performance, such as occlusion, fast motion, and illumination variation.

One common issue in assessing tracking algorithms is that the results are reported based on just a few sequences with different initial conditions or parameters. Thus, the results do not provide the holistic view of these algorithms. For fair and comprehensive performance evaluation, we propose to perturb the initial state spatially and temporally from the ground-truth target locations. While the robustness to initialization is a well-known problem in the field, it is seldom addressed in the literature. To the best of our knowledge, this is the first comprehensive work to address and analyze the initialization problem of object tracking. We use the precision plots based on location error metric and the success plots based on the overlap metric, to analyze the performance of each algorithm.

The contribution of this work is three-fold:
Dataset. We build a tracking dataset with 50 fully annotated sequences to facilitate tracking evaluation.
Code library. We integrate most publicly available trackers in our code library with uniform input and output formats to facilitate large scale performance evaluation. At present, it includes 29 tracking algorithms.
Robustness evaluation. The initial bounding boxes for tracking are sampled spatially and temporally to evaluate the robustness and characteristics of trackers. Each track-

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.312

er is extensively evaluated by analyzing more than 660,000 bounding box outputs.

This work mainly focuses on the online[1] tracking of single target. The code library, annotated dataset and all the tracking results are available on the website http://visual-tracking.net .

2. Related Work

In this section, we review recent algorithms for object tracking in terms of several main modules: target representation scheme, search mechanism, and model update. In addition, some methods have been proposed that build on combing some trackers or mining context information.

Representation Scheme. Object representation is one of major components in any visual tracker and numerous schemes have been presented [35]. Since the pioneering work of Lucas and Kanade [37, 8], *holistic* templates (raw intensity values) have been widely used for tracking [25, 39, 2]. Subsequently, subspace-based tracking approaches [11, 47] have been proposed to better account for appearance changes. Furthermore, Mei and Ling [40] proposed a tracking approach based on sparse representation to handle the corrupted appearance and recently it has been further improved [41, 57, 64, 10, 55, 42]. In addition to template, many other visual features have been adopted in tracking algorithms, such as color histograms [16], histograms of oriented gradients (HOG) [17, 52], covariance region descriptor [53, 46, 56] and Haar-like features [54, 22]. Recently, the discriminative model has been widely adopted in tracking [15, 4], where a binary classifier is learned online to discriminate the target from the background. Numerous learning methods have been adapted to the tracking problem, such as SVM [3], structured output SVM [26], ranking SVM [7], boosting [4, 22], semi-boosting [23] and multi-instance boosting [5]. To make trackers more robust to pose variation and partial occlusion, an object can be represented by parts where each one is represented by descriptors or histograms. In [1] several *local* histograms are used to represent the object in a pre-defined grid structure. Kwon and Lee [32] propose an approach to automatically update the topology of local patches to handle large pose changes. To better handle appearance variations, some approaches regarding integration of multiple representation schemes have recently been proposed [62, 51, 33].

Search Mechanism. To estimate the state of the target objects, deterministic or stochastic methods have been used. When the tracking problem is posed within an optimization framework, assuming the objective function is differentiable with respect to the motion parameters, gradient descent methods can be used to locate the target efficiently [37, 16, 20, 49]. However, these objective functions are usually nonlinear and contain many local minima. To alleviate this problem, dense sampling methods have been adopted [22, 5, 26] at the expense of high computational load. On the other hand, stochastic search algorithms such as particle filters [28, 44] have been widely used since they are relatively insensitive to local minima and computationally efficient [47, 40, 30].

Model Update. It is crucial to update the target representation or model to account for appearance variations. Matthews *et al.* [39] address the template update problem for the Lucas-Kanade algorithm [37] where the template is updated with the combination of the fixed reference template extracted from the first frame and the result from the most recent frame. Effective update algorithms have also been proposed via online mixture model [29], online boosting [22], and incremental subspace update [47]. For discriminative models, the main issue has been improving the sample collection part to make the online-trained classifier more robust [23, 5, 31, 26]. While much progress has been made, it is still difficult to get an adaptive appearance model to avoid drifts.

Context and Fusion of Trackers. Context information is also very important for tracking. Recently some approaches have been proposed by mining auxiliary objects or local visual information surrounding the target to assist tracking [59, 24, 18]. The context information is especially helpful when the target is fully occluded or leaves the image region [24]. To improve the tracking performance, some tracker fusion methods have been proposed recently. Santner *et al.* [48] proposed an approach that combines static, moderately adaptive and highly adaptive trackers to account for appearance changes. Even multiple trackers [34] or multiple feature sets [61] are maintained and selected in a Bayesian framework to better account for appearance changes.

3. Evaluated Algorithms and Datasets

For fair evaluation, we test the tracking algorithms whose original source or binary codes are publicly available as all implementations inevitably involve technical details and specific parameter settings[2]. Table 1 shows the list of the evaluated tracking algorithms. We also evaluate the trackers in the VIVID testbed [13] including the mean shift (MS-V), template matching (TM-V), ratio shift (RS-V) and peak difference (PD-V) methods.

In recent years, many benchmark datasets have been developed for various vision problems, such as the Berkeley segmentation [38], FERET face recognition [45] and optical flow dataset [9]. There exist some datasets for the tracking in the surveillance scenario, such as the VIVID [13] and CAVIAR [21] datasets. For generic visual tracking, more

[1] Here, the word *online* means during tracking only the information of previous few frames is used for inference at any time instance.

[2] Some source codes [36, 58] are obtained from direct contact, and some methods are implemented on our own [44, 16].

Method	Representation	Search	MU	Code	FPS
CPF [44]	L, IH	PF	N	C	109
LOT [43]	L, color	PF	Y	M	0.70
IVT [47]	H, PCA, GM	PF	Y	MC	33.4
ASLA [30]	L, SR, GM	PF	Y	MC	8.5
SCM [65]	L, SR, GM+DM	PF	Y	MC	0.51
L1APG [10]	H, SR, GM	PF	Y	MC	2.0
MTT [64]	H, SR, GM	PF	Y	M	1.0
VTD [33]	H, SPCA, GM	MCMC	Y	MC-E	5.7
VTS [34]	L, SPCA, GM	MCMC	Y	MC-E	5.7
LSK [36]	L, SR, GM	LOS	Y	M-E	5.5
ORIA [58]	H, T, GM	LOS	Y	M	9.0
DFT [49]	L, T	LOS	Y	M	13.2
KMS [16]	H, IH	LOS	N	C	3,159
SMS [14]	H, IH	LOS	N	C	19.2
VR-V [15]	H, color	LOS	Y	MC	109
Frag [1]	L, IH	DS	N	C	6.3
OAB [22]	H, Haar, DM	DS	Y	C	22.4
SemiT [23]	H, Haar, DM	DS	Y	C	11.2
BSBT [50]	H, Haar, DM	DS	Y	C	7.0
MIL [5]	H, Haar, DM	DS	Y	C	38.1
CT [63]	H, Haar, DM	DS	Y	MC	64.4
TLD [31]	L, BP, DM	DS	Y	MC	28.1
Struck [26]	H, Haar, DM	DS	Y	C	20.2
CSK [27]	H, T, DM	DS	Y	M	362
CXT [18]	H, BP, DM	DS	Y	C	15.3

Table 1. Evaluated tracking algorithms (MU: model update, FPS: frames per second). For representation schemes, L: local, H: holistic, T: template, IH: intensity histogram, BP: binary pattern, PCA: principal component analysis, SPCA: sparse PCA, SR: sparse representation, DM: discriminative model, GM: generative model. For search mechanism, PF: particle filter, MCMC: Markov Chain Monte Carlo, LOS: local optimum search, DS: dense sampling search. For the model update, N: No, Y: Yes. In the Code column, M: Matlab, C:C/C++, MC: Mixture of Matlab and C/C++, suffix E: executable binary code.

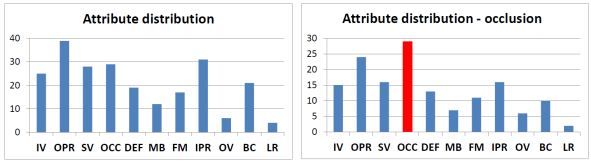
sequences have been used for evaluation [47, 5]. However, most sequences do not have the ground truth annotations, and the quantitative evaluation results may be generated with different initial conditions. To facilitate fair performance evaluation, we have collected and annotated most commonly used tracking sequences. Figure 1 shows the first frame of each sequence where the target object is initialized with a bounding box.

Attributes of a test sequence. Evaluating trackers is difficult because many factors can affect the tracking performance. For better evaluation and analysis of the strength and weakness of tracking approaches, we propose to categorize the sequences by annotating them with the 11 attributes shown in Table 2.

The attribute distribution in our dataset is shown in Figure 2(a). Some attributes occur more frequently, e.g., OPR and IPR, than others. It also shows that one sequence is often annotated with several attributes. Aside from summarizing the performance on the whole dataset, we also construct several subsets corresponding to attributes to report specific challenging conditions. For example, the OCC subset contains 29 sequences which can be used to analyze the

Attr	Description
IV	Illumination Variation - the illumination in the target region is significantly changed.
SV	Scale Variation - the ratio of the bounding boxes of the first frame and the current frame is out of the range $[1/t_s, t_s]$, $t_s > 1$ (t_s=2).
OCC	Occlusion - the target is partially or fully occluded.
DEF	Deformation - non-rigid object deformation.
MB	Motion Blur - the target region is blurred due to the motion of target or camera.
FM	Fast Motion - the motion of the ground truth is larger than t_m pixels (t_m=20).
IPR	In-Plane Rotation - the target rotates in the image plane.
OPR	Out-of-Plane Rotation - the target rotates out of the image plane.
OV	Out-of-View - some portion of the target leaves the view.
BC	Background Clutters - the background near the target has the similar color or texture as the target.
LR	Low Resolution - the number of pixels inside the ground-truth bounding box is less than t_r (t_r=400).

Table 2. List of the attributes annotated to test sequences. The threshold values used in this work are also shown.



(a) (b)

Figure 2. (a) Attribute distribution of the entire testset, and (b) the distribution of the sequences with occlusion (OCC) attribute.

performance of trackers to handle occlusion. The attribute distributions in OCC subset is shown in Figure 2(b) and others are available in the supplemental material.

4. Evaluation Methodology

In this work, we use the precision and success rate for quantitative analysis. In addition, we evaluate the robustness of tracking algorithms in two aspects.

Precision plot. One widely used evaluation metric on tracking precision is the center location error, which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. Then the average center location error over all the frames of one sequence is used to summarize the overall performance for that sequence. However, when the tracker loses the target, the output location can be random and the average error value may not measure the tracking performance correctly [6]. Recently the precision plot [6, 27] has been adopted to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. As the representative precision score for each tracker we use the score for the threshold = 20 pixels [6].

Success plot. Another evaluation metric is the bounding box overlap. Given the tracked bounding box r_t and the

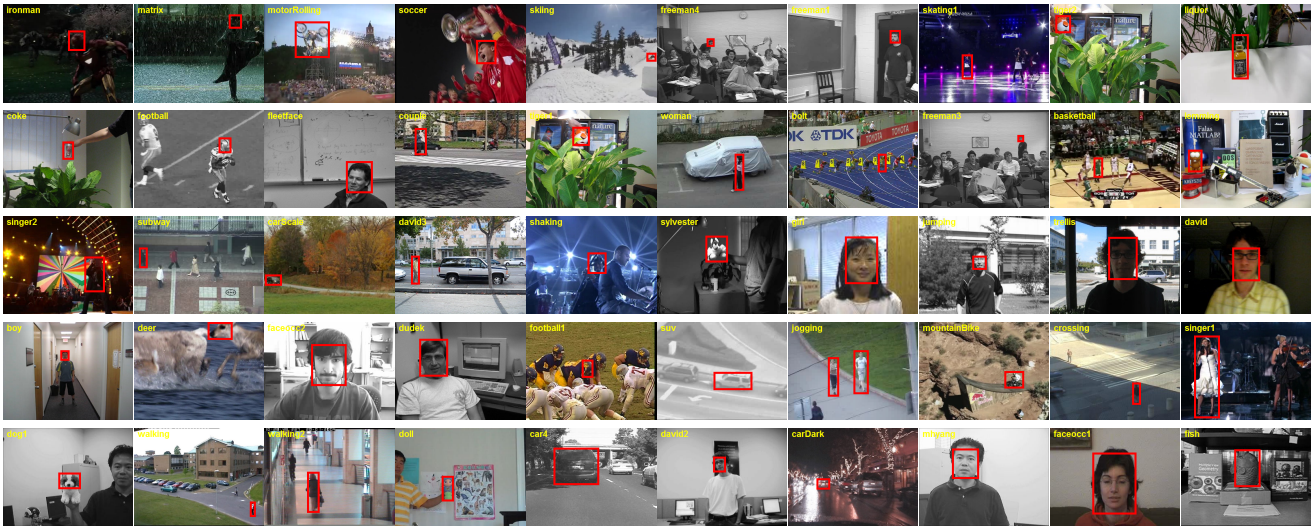


Figure 1. Tracking sequences for evaluation. The first frame with the bounding box of the target object is shown for each sequence. The sequences are ordered based on our ranking results (See supplementary material): the ones on the top left are more difficult for tracking than the ones on the bottom right. Note that we annotated two targets for the *jogging* sequence.

ground truth bounding box r_a, the overlap score is defined as $S = \frac{|\, r_t \bigcap r_a \,|}{|\, r_t \bigcup r_a \,|}$, where $\bigcap$ and $\bigcup$ represent the intersection and union of two regions, respectively, and $|\cdot|$ denotes the number of pixels in the region. To measure the performance on a sequence of frames, we count the number of successful frames whose overlap S is larger than the given threshold t_o. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. Using one success rate value at a specific threshold (e.g. t_o=0.5) for tracker evaluation may not be fair or representative. Instead we use the area under curve (AUC) of each success plot to rank the tracking algorithms.

Robustness Evaluation. The conventional way to evaluate trackers is to run them throughout a test sequence with initialization from the ground truth position in the first frame and report the average precision or success rate. We refer this as one-pass evaluation (**OPE**). However a tracker may be sensitive to the initialization, and its performance with different initialization at a different start frame may become much worse or better. Therefore, we propose two ways to analyze a tracker's robustness to initialization, by perturbing the initialization temporally (i.e., start at different frames) and spatially (i.e., start by different bounding boxes). These tests are referred as temporal robustness evaluation (**TRE**) and spatial robustness evaluation (**SRE**) respectively.

The proposed test scenarios happen a lot in the real-world applications as a tracker is often initialized by an object detector, which is likely to introduce initialization errors in terms of position and scale. In addition, an object detector may be used to re-initialize a tracker at different time instances. By investigating a tracker's characteristic in the robustness evaluation, more thorough understanding and analysis of the tracking algorithm can be carried out.

Temporal Robustness Evaluation. Given one initial frame together with the ground-truth bounding box of target, one tracker is initialized and runs to the end of the sequence, i.e., one segment of the entire sequence. The tracker is evaluated on each segment, and the overall statistics are tallied.

Spatial Robustness Evaluation. We sample the initial bounding box in the first frame by shifting or scaling the ground truth. Here, we use 8 spatial shifts including 4 center shifts and 4 corner shifts, and 4 scale variations (supplement). The amount for shift is 10% of target size, and the scale ratio varys among 0.8, 0.9, 1.1 and 1.2 to the ground truth. Thus, we evaluate each tracker 12 times for SRE.

5. Evaluation Results

For each tracker, the default parameters with the source code are used in all evaluations. Table 1 lists the average FPS of each tracker in OPE running on a PC with Intel i7 3770 CPU (3.4GHz). More detailed speed statistics, such as minimum and maximum, are available in the supplement.

For OPE, each tracker is tested on more than 29,000 frames. For SRE, each tracker is evaluated 12 times on each sequence, where more than 350,000 bounding box results are generated. For TRE, each sequence is partitioned into 20 segments and thus each tracker is performed on around 310,000 frames. To the best of our knowledge, this is the largest scale performance evaluation of visual tracking. We report the most important findings in this manuscript and more details and figures can be found in the supplement.

5.1. Overall Performance

The overall performance for all the trackers is summarized by the success and precision plots as shown in Fig-

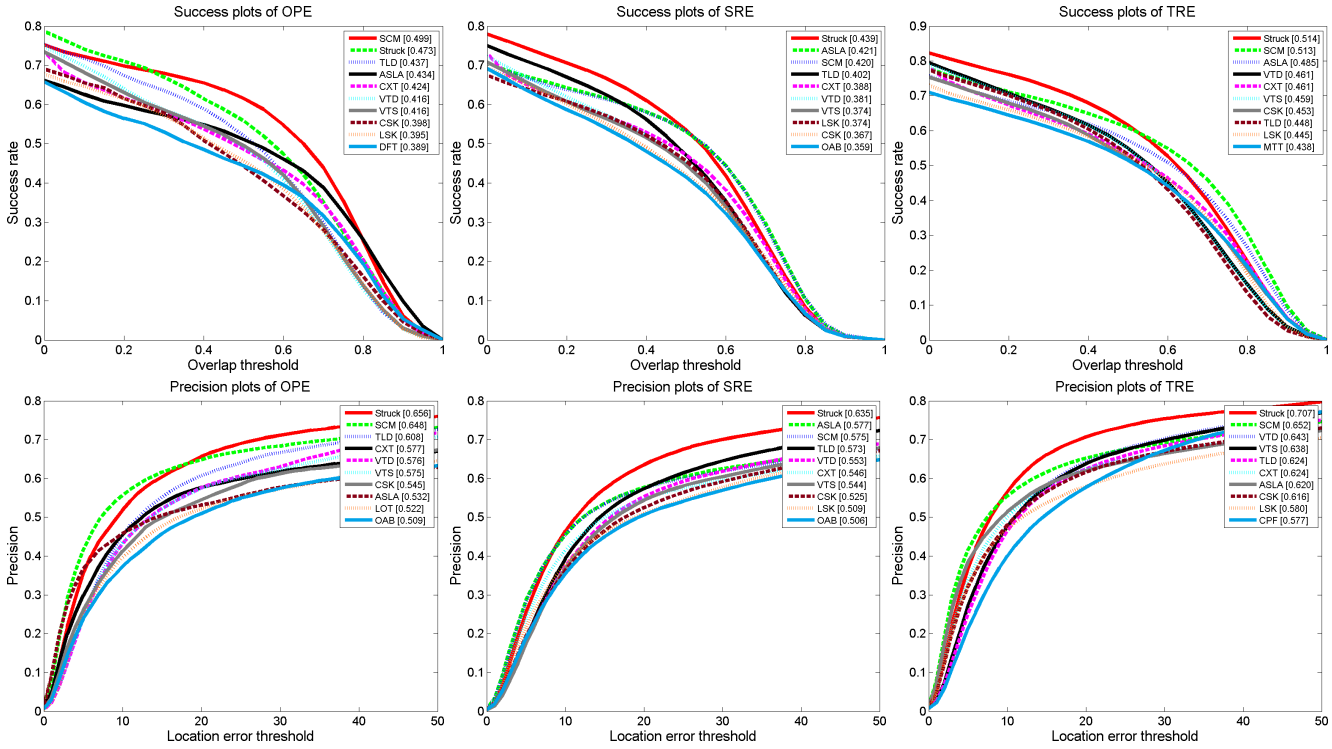


Figure 3. Plots of OPE, SRE, and TRE. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity and complete plots are in the supplementary material (best viewed on high-resolution display).

ure 3 where only the top 10 algorithms are presented for clarity and the complete plots are displayed in the supplementary material. For success plots, we use AUC scores to summarize and rank the trackers, while for precision plots we use the results at error threshold of 20 for ranking. In the precision plots, the rankings of some trackers are slightly different from the rankings in the success plots in that they are based on different metrics which measure different characteristics of trackers. Because the AUC score of success plot measures the overall performance which is more accurate than the score at one threshold of the plot, in the following we mainly analyze the rankings based on success plots but use the precision plots as auxiliary.

The average TRE performance is higher than that of OPE in that the number of frames decreases from the first to last segment of TRE. As the trackers tend to perform well in shorter sequences, the average of all the results in TRE tend to be higher. On the other hand, the average performance of SRE is lower than that of OPE. The initialization errors tend to cause trackers to update with imprecise appearance information, thereby causing gradual drifts.

In the success plots, the top ranked tracker SCM in OPE outperforms Struck by 2.6% but is 1.9% below Struck in SRE. The results also show that OPE is not the best performance indicator as the OPE is one trial of SRE or TRE. The ranking of TLD in TRE is lower than OPE and SRE. This is because TLD performs well in long sequences with a re-detection module while there are numerous short segments in TRE. The success plots of Struck in TRE and SRE show that the success rate of Struck is higher than SCM and ALSA when the overlap threshold is small, but less than SCM and ALSA when the overlap threshold is large. This is because Struck only estimates the location of target and does not handle scale variation.

Sparse representations are used in SCM, ASLA, LSK, MTT and L1APG. These trackers perform well in SRE and TRE, which suggests sparse representations are effective models to account for appearance change (e.g., occlusion). We note that SCM, ASLA and LSK outperform MTT and L1APG. The results suggest that local sparse representations are more effective than the ones with holistic sparse templates. The AUC score of ASLA deceases less than the other top 5 trackers from OPE to SRE and the ranking of ASLA also increases. It indicates the alignment-pooling technique adopted by ASLA is more robust to misalignments and background clutters.

Among the top 10 trackers, CSK has the highest speed where the proposed circulant structure plays a key role. The VTD and VTS methods adopt mixture models to improve the tracking performance. Compared with other higher ranked trackers, the performance bottleneck of them can be attributed to their adopted representation based on sparse

principal component analysis, where the holistic templates are used. Due to the space limitation, the plots of SRE are presented for analysis in the following sections, and more results are included in the supplement.

5.2. Attribute-based Performance Analysis

By annotating the attributes of each sequence, we construct subsets with different dominant attributes which facilitates analyzing the performance of trackers for each challenging factor. Due to space limitations, we only illustrate and analyze the success plots and precision plots of SRE for attributes OCC, SV, and FM as shown in Figure 4, and more results are presented in the supplementary material.

When an object moves fast, dense sampling based trackers (e.g., Struck, TLD and CXT) perform much better than others. One reason is that the search ranges are large and the discriminative models are able to discriminate the targets from the background clutters. However, the stochastic search based trackers with high overall performance (e.g., SCM and ASLA) do not perform well in this subset due to the poor dynamic models. If these parameters are set to large values, more particles are required to make the tracker stable. These trackers can be further improved with dynamic models with more effective particle filters.

On the OCC subset, the Struck, SCM, TLD, LSK and ASLA methods outperform others. The results suggest that structured learning and local sparse representations are effective in dealing with occlusions. On the SV subset, ASLA, SCM and Struck perform best. The results show that trackers with affine motion models (e.g., ASLA and SCM) often handle scale variation better than others that are designed to account for only translational motion with a few exceptions such as Struck.

5.3. Initialization with Different Scale

It has been known that trackers are often sensitive to initialization variations. Figure 5 and Figure 6 show the summarized tracking performance with initialization at different scales. When computing the overlap score, we rescale the tracking results so that the performance summary could be comparable with the original scale, i.e., the plots of OPE in Figure 3. Figure 6 illustrates the average performance of all trackers for each scale which shows the performance often decreases significantly when the scale factor is large (e.g., $\times 1.2$) as many background pixels are inevitably included in the initial representations. The performance of TLD, CXT, DFT and LOT decreases with the increase of initialization scale. This indicates these trackers are more sensitive to background clutters. Some trackers perform better when the scale factor is smaller, such as L1APG, MTT, LOT and CPF. One reason for this in the case of L1APG and MTT is that the templates have to be warped to fit the size of the usually smaller canonical template so that if the initial template is small, more appearance details will be kept in the model. On the other hand, some trackers perform well or even better when the initial bounding box is enlarged, such as Struck, OAB, SemiT, and BSBT. This indicates that the Haar-like features are somewhat robust to background clutters due to the summation operations when computing features. Overall, Struck is less sensitive to scale variation than other well-performing methods.

6. Concluding Remarks

In this paper, we carry out large scale experiments to evaluate the performance of recent online tracking algorithms. Based on our evaluation results and observations, we highlight some tracking components which are essential for improving tracking performance. First, background information is critical for effective tracking. It can be exploited by using advanced learning techniques to encode the background information in the discriminative model implicitly (e.g., Struck), or serving as the tracking context explicitly (e.g., CXT). Second, local models are important for tracking as shown in the performance improvement of local sparse representation (e.g., ASLA and SCM) compared with the holistic sparse representation (e.g., MTT and L1APG). They are particularly useful when the appearance of target is partially changed, such as partial occlusion or deformation. Third, motion model or dynamic model is crucial for object tracking, especially when the motion of target is large or abrupt. However, most of our evaluated trackers do not focus on this component. Good location prediction based on the dynamic model could reduce the search range and thus improve the tracking efficiency and robustness. Improving these components will further advance the state of the art of online object tracking.

The evaluation results show that significant progress in the field of object tracking has been made in the last decade. We propose and demonstrate evaluation metrics for in-depth analysis of tracking algorithms from several perspectives. This large scale performance evaluation facilitates better understanding of the state-of-the-art online object tracking approaches, and provides a platform for gauging new algorithms. Our ongoing work focuses on extending the dataset and code library to include more fully annotated sequences and trackers.

Acknowledgment. We thank the reviewers for valuable comments and suggestions. The work is supported partly by NSF CAREER Grant #1149783 and NSF IIS Grant #1152576. Wu is also with Nanjing University of Information Science and Technology, China and supported partly by NSFC Grant #61005027.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In *CVPR*, 2006.
[2] N. Alt, S. Hinterstoisser, and N. Navab. Rapid Selection of Reliable Templates for Visual Tracking. In *CVPR*, 2010.
[3] S. Avidan. Support Vector Tracking. *PAMI*, 26(8):1064–1072, 2004.
[4] S. Avidan. Ensemble Tracking. *PAMI*, 29(2):261–271, 2008.

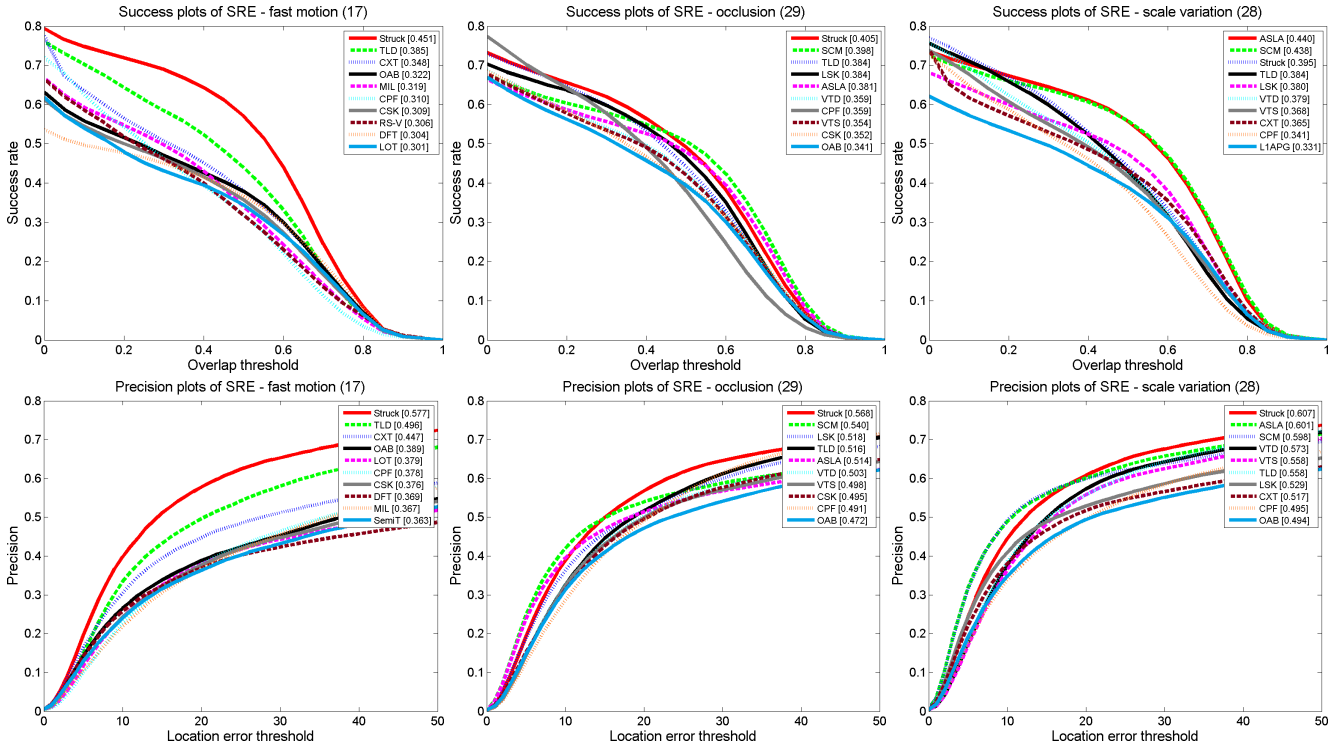


Figure 4. Plots for OCC, SV, and FM subsets. The value appears in the title is the number of sequences in that subset. Only the top 10 trackers are presented for clarity and complete plots are in the supplementary material (best viewed on high-resolution display).

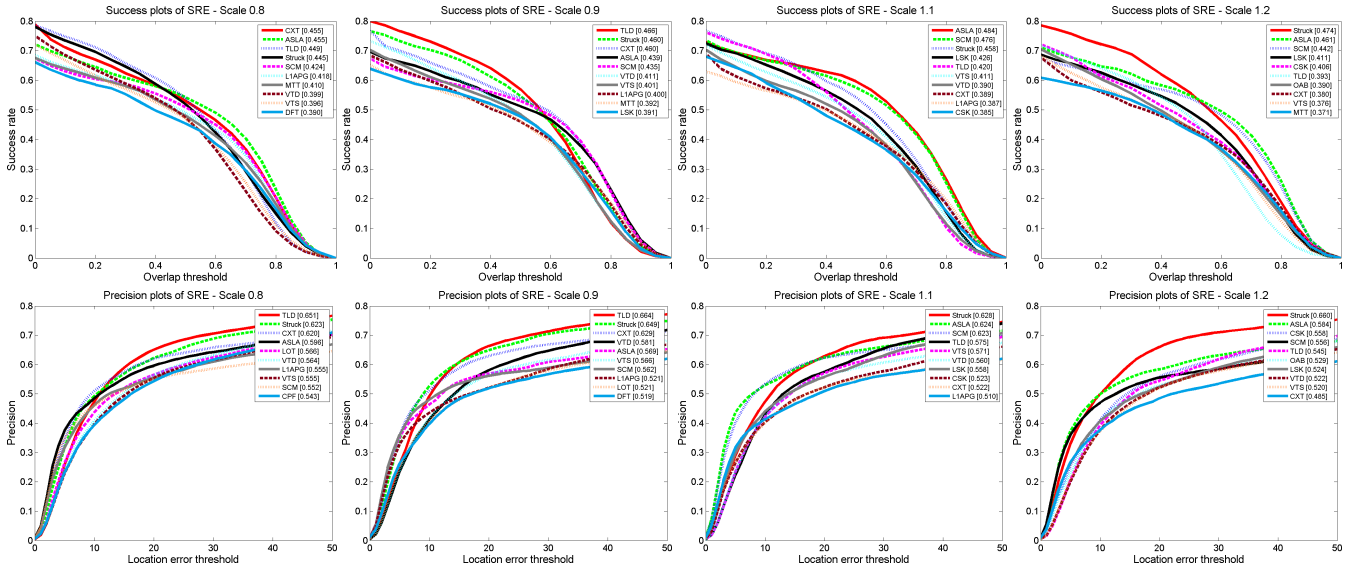


Figure 5. SRE of the trackers initialized with different bounding box sizes. The value on top of each figure is the scale factor. For each figure, the top 10 trackers are presented for clarity and complete plots are in the supplementary material.

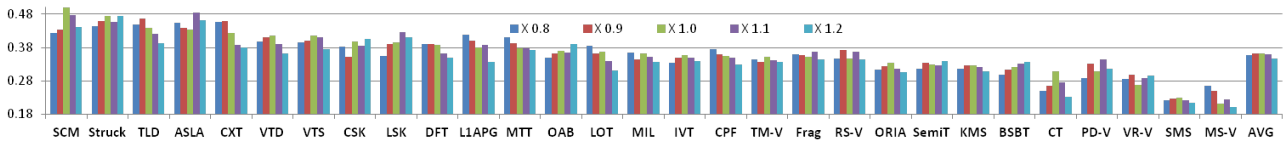


Figure 6. Performance summary for the trackers initialized with different size of bounding box. AVG (the last one) illustrates the average performance over all trackers for each scale.

[5] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[6] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *PAMI*, 33(7):1619–1632, 2011.

[7] Y. Bai and M. Tang. Robust Tracking via Weakly Supervised Ranking SVM. In *CVPR*, 2012.

[8] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 56(3):221–255, 2004.

[9] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. In *ICCV*, 2007.

[10] C. Bao, Y. Wu, H. Ling, and H. Ji. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. In *CVPR*, 2012.

[11] M. J. Black. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *IJCV*, 26(1):63–84, 1998.

[12] K. Cannons. A Review of Visual Tracking. Technical Report CSE-2008-07, York University, Canada, 2008.

[13] R. Collins, X. Zhou, and S. K. Teh. An Open Source Tracking Testbed and Evaluation Web Site. In *PETS*, 2005.

[14] R. T. Collins. Mean-shift Blob Tracking through Scale Space. In *CVPR*, 2003.

[15] R. T. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631–1643, 2005.

[16] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *PAMI*, 25(5):564–577, 2003.

[17] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[18] T. B. Dinh, N. Vo, and G. Medioni. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments. In *CVPR*, 2011.

[19] J. Fan, X. Shen, and Y. Wu. Scribble Tracker: A Matting-based Approach for Robust Tracking. *PAMI*, 34(8):1633–1644, 2012.

[20] J. Fan, Y. Wu, and S. Dai. Discriminative Spatial Attention for Robust Tracking. In *ECCV*, 2010.

[21] R. B. Fisher. The PETS04 Surveillance Ground-Truth Data Sets. In *PETS*, 2004.

[22] H. Grabner, M. Grabner, and H. Bischof. Real-Time Tracking via On-line Boosting. In *BMVC*, 2006.

[23] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-Line Boosting for Robust Tracking. In *ECCV*, 2008.

[24] H. Grabner, J. Matas, L. V. Gool, and P. Cattin. Tracking the Invisible: Learning Where the Object Might be. In *CVPR*, 2010.

[25] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *PAMI*, 20(10):1025–1039, 1998.

[26] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured Output Tracking with Kernels. In *ICCV*, 2011.

[27] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *ECCV*, 2012.

[28] M. Isard and A. Blake. CONDENSATION–Conditional Density Propagation for Visual Tracking. *IJCV*, 29(1):5–28, 1998.

[29] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust Online Appearance Models for Visual Tracking. *PAMI*, 25(10):1296–1311, 2003.

[30] X. Jia, H. Lu, and M.-H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. In *CVPR*, 2012.

[31] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. In *CVPR*, 2010.

[32] J. Kwon and K. M. Lee. Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling. In *CVPR*, 2009.

[33] J. Kwon and K. M. Lee. Visual Tracking Decomposition. In *CVPR*, 2010.

[34] J. Kwon and K. M. Lee. Tracking by Sampling Trackers. In *ICCV*, 2011.

[35] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Hengel. A Survey of Appearance Models in Visual Object Tracking. *TIST*, 2013, in press.

[36] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust Tracking using Local Sparse Appearance Model and K-Selection. In *CVPR*, 2011.

[37] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with An Application to Stereo Vision. In *IJCAI*, 1981.

[38] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *PAMI*, 26(5):530–49, 2004.

[39] I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem. *PAMI*, 26(6):810–815, 2004.

[40] X. Mei and H. Ling. Robust Visual Tracking using L1 Minimization. In *ICCV*, 2009.

[41] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection. In *CVPR*, 2011.

[42] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Efficient Minimum Error Bounded Particle Resampling L1 Tracker with Occlusion Detection. *TIP*, 2013, in press.

[43] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally Orderless Tracking. In *CVPR*, 2012.

[44] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *ECCV*, 2002.

[45] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *PAMI*, 22(10):1090–1104, 2000.

[46] F. Porikli, O. Tuzel, and P. Meer. Covariance Tracking using Model Update based on Lie Algebra. In *CVPR*, 2006.

[47] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, 77(1):125–141, 2008.

[48] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel Robust Online Simple Tracking. In *CVPR*, 2010.

[49] L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. In *CVPR*, 2012.

[50] S. Stalder, H. Grabner, and L. van Gool. Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition. In *ICCV Workshop*, 2009.

[51] B. Stenger, T. Woodley, and R. Cipolla. Learning to Track with Multiple Observers. In *CVPR*, 2009.

[52] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-Tracking Using Semi-Supervised Support Vector Machines. In *CVPR*, 2007.

[53] O. Tuzel, F. Porikli, and P. Meer. Region Covariance: A Fast Descriptor for Detection and Classification. In *ECCV*, 2006.

[54] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *IJCV*, 57(2):137–154, 2004.

[55] D. Wang, H. Lu, and M.-H. Yang. Online Object Tracking with Sparse Prototypes. *TIP*, 22(1):314–325, 2013.

[56] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai. Real-time Probabilistic Covariance Tracking with Efficient Model Update. *TIP*, 21(5):2824–2837, 2012.

[57] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred Target Tracking by Blur-driven Tracker. In *ICCV*, 2011.

[58] Y. Wu, B. Shen, and H. Ling. Online Robust Image Alignment via Iterative Convex Optimization. In *CVPR*, 2012.

[59] M. Yang, Y. Wu, and G. Hua. Context-Aware Visual Tracking. *PAMI*, 31(7):1195–1209, 2008.

[60] A. Yilmaz, O. Javed, and M. Shah. Object Tracking: A Survey. *ACM Computing Surveys*, 38(4):1–45, 2006.

[61] J. H. Yoon, D. Y. Kim, and K.-J. Yoon. Visual Tracking via Adaptive Tracker Selection with Multiple Features. In *ECCV*, 2012.

[62] L. Yuan, A. Haizhou, T. Yamashita, L. Shihong, and M. Kawade. Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans. *PAMI*, 30(10):1728–1740, 2008.

[63] K. Zhang, L. Zhang, and M.-H. Yang. Real-time Compressive Tracking. In *ECCV*, 2012.

[64] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust Visual Tracking via Multi-task Sparse Learning. In *CVPR*, 2012.

[65] W. Zhong, H. Lu, and M.-H. Yang. Robust Object Tracking via Sparsity-based Collaborative Model. In *CVPR*, 2012.