# Transparent Object Tracking Benchmark

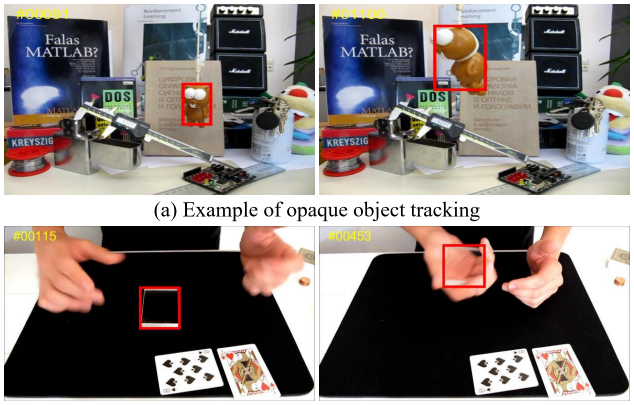
Heng Fan[1] Halady Akhilesha Miththanthaya[2*] Harshit[2*] Siranjiv Ramana Rajan[2*]
Xiaoqiong Liu[2] Zhilin Zou[2] Yuewei Lin[3] Haibin Ling[2†]
[1]Department of Computer Science and Engineering, University of North Texas, Denton, USA
[2]Department of Computer Science, Stony Brook University, Stony Brook, USA
[3]Computational Science Initiative, Brookhaven National Laboratory, Upton, USA
heng.fan@unt.edu hling@cs.stonybrook.edu

## Abstract

*Visual tracking has achieved considerable progress in recent years. However, current research in the field mainly focuses on tracking of opaque objects, while little attention is paid to transparent object tracking. In this paper, we make the first attempt in exploring this problem by proposing a Transparent Object Tracking Benchmark (TOTB). Specifically, TOTB consists of 225 videos (86K frames) from 15 diverse transparent object categories. Each sequence is manually labeled with axis-aligned bounding boxes. To the best of our knowledge, TOTB is the first benchmark dedicated to transparent object tracking. In order to understand how existing trackers perform and to provide comparison for future research on TOTB, we extensively evaluate 25 state-of-the-art tracking algorithms. The evaluation results exhibit that more efforts are needed to improve transparent object tracking. Besides, we observe some nontrivial findings from the evaluation that are discrepant with some common beliefs in opaque object tracking. For example, we find that deeper features are not always good for improvements. Moreover, to encourage future research, we introduce a novel tracker, named TransATOM, which leverages transparency features for tracking and surpasses all 25 evaluated approaches by a large margin. By releasing TOTB, we expect to facilitate future research and application of transparent object tracking in both the academia and industry. The TOTB and evaluation results as well as TransATOM are available at https://hengfan2010.github.io/projects/TOTB/.*

## 1. Introduction

Object tracking is one of the most fundamental problems in computer vision and serves as an important component in numerous applications [37, 50, 60, 36] including robotics, human-machine interaction, video analysis and understanding, *etc*. In recent decades, the tracking community has witnessed remarkable progress. Numerous tracking algorithms have been proposed and significantly pushed the state-of-the-arts. Nevertheless, existing research in the field mainly focuses on opaque object tracking, while *very little* attention is paid to tracking of transparent objects.




(a) Example of opaque object tracking

(b) Example of transparent object tracking

Figure 1. Opaque object tracking (a) and transparent object tracking (b). Compared with opaque object tracking in which target object appearance is more distinguishable from background and consistent over time, tracking of transparent target is more challenging as transparent object appearance is heavily dependent on background. *All figures in this paper are best viewed in color and by zooming in.*

Transparent objects (*e.g.*, *bottle*, *cup*, *bulb*, *jar* and many others made by glass and plastics) are common to see in the real world. Many of them are closely related to human daily life, and tracking of them are crucial for robotic vision and human-machine interaction. For example, a robot may need to know the trajectory of a transparent object in human hand for better action understanding.

Compared with tracking of opaque objects, transparent object tracking is more challenging. Because of the par-

*The three authors make equal contributions.
†Corresponding author.

ticular *transparency* feature, the appearances of transparent objects are relatively *weak* and largely mixed with the surrounding background image (see Figure 1 for an example). As a result, it becomes more difficult to directly leverage appearance information to distinguish the target object from background. In addition, when a target object moves, even slowly, its appearance may change drastically due to background variation, making transparent object tracking harder.

Besides the above technical difficulty, another more important reason that transparent object tracking is untouched is because of lack of a benchmark. Benchmark is crucial for the advancement of tracking. It allows researchers to objectively evaluate and compare their methods as well as design new algorithms for improvement. Currently, there exist various benchmarks (*e.g.*, [57, 45, 20, 47, 54, 30, 14, 25, 42]) for opaque object tracking. However, there is *no* benchmark for transparent object tracking. Although some of benchmarks (*e.g.*, [14, 39]) consist of sequences of transparent objects, they are limited in both number of videos (*e.g.*, less than 10) and object classes (*e.g.*, at most two categories). To facilitate research on transparent object tracking, a dedicated dataset is desired to serve as the testbed for fair evaluation and comparison.

### 1.1. Contribution

In this work, we make the *first* attempt in exploring transparent object tracking by introducing a Transparent Object Tracking benchmark (TOTB), which is our major contribution. TOTB comprises of a diverse selection of 15 common transparent object classes with each containing 15 sequences. In total, TOTB consists of 225 sequences with 87K frames. Each sequence is manually annotated with axis-align bounding boxes and labeled with different attributes. To our best knowledge, TOTB is the *first* benchmark dedicated to the task of transparent object tracking. Figure 4 demonstrates several example sequences in TOTB.

Besides, in order to understand how existing tracking algorithms perform and to provide comparisons for future research on TOTB, we extensively evaluate 25 state-of-the-art trackers. We conduct in-depth analysis on the evaluation results and observe several surprising findings that are discrepant with some popular beliefs in the opaque object tracking. For example, it is widely believed that deeper features are crucial to improve tracking performance, as shown in the existing opaque tracking benchmarks (*e.g.*, [57, 14, 47, 25]). Contrary to this, it turns out that deeper features do not always bring performance gains for transparent object tracking. Instead, it may heavily decrease accuracy. These observations provide better understanding of transparent object tracking and guidance for future improvements.

Furthermore, to facilitate the development of tracking algorithms on TOTB, we introduce a *simple yet effective* tracker by exploiting transparency features for tracking. In particular, considering that transparency is a common attribute of transparent objects, its feature should be generic and transferable for all transparent instances, and also distinguishable from opaque objects. To this end, we train a deep network to learn such transparency feature and apply it for tracking by integrating it into ATOM [8]. Our new tracker, dubbed TransATOM, is assessed on TOTB and significantly outperforms all evaluated algorithms by a large margin. Note that, although TransATOM is simple, it demonstrates the effectiveness of transparency feature in boosting performance. We expect it to provide a reference for facilitating future study.

In summary, we make the following contributions:
**(1)** *We propose TOTB, which is, to the best of knowledge, the first benchmark dedicated for transparent object tracking.*
**(2)** *To assess existing trackers and provide comparisons, we evaluate 25 tracking algorithms with in-depth analysis.*
**(3)** *We introduce a novel transparent object tracker, named TransATOM, to encourage further research on TOTB.*

By releasing TOTB, we hope to facilitate future research and application of transparent object tracking.

The rest of this paper is organized as follows. We discuss related works of this paper in Section 2. Section 3 details the proposed TOTB. Section 4 introduces our proposed tracker TransATOM. Evaluation results are shown in Section 5 with in-depth analysis, followed by conclusion in Section 6.

## 2. Related Work

### 2.1. Tracking Algorithm

As one of core members in the computer vision family, visual tracking has been studied for decades, with a huge past literature whose review is beyond this paper. In this section, we review two popular trends including correlation filter tracking and deep tracking in the field and refer readers to [37, 50, 60, 36] for comprehensive tracking surveys.

Roughly speaking, correlation filter-based tracking algorithms treat tracking as an online regression problem. Correlation filter trackers like [5, 24] demonstrate impressive running speeds of several hundreds frames per second and attract great attention in the tracking community with many inspired extensions for improvements. For example, an additional scale filter is utilized in [38, 10] to deal with the target scale variations. The approaches in [11, 21, 7, 34] leverage regularization techniques to improve robustness. The tracker in [15] integrates correlation filter tracker with an independent verifier to alleviate the drifting problem. The methods of [43, 12, 6] apply deep features to replace handcrafted ones in correlation filter tracking and achieve significant improvements.

Motivated by the tremendous success of deep features in other vision tasks, deep learning-based trackers have been

developed in recent years. Among them, a popular series follows the Siamese trackers [52, 2], which present a simple architecture yet promising performance. Notably, a fully convolutional Siamese network is introduced in [2] with a light structure for tracking, leading to very efficient running performance. Inspired by the balanced accuracy and speed of [2], many other variants [22, 33, 32, 35, 64, 16, 55, 61, 62, 17] have been developed and generated boosted performances. Along another line, some deep trackers [8, 3, 9] decompose tracking into two separate localization and scale estimation tasks, which are respectively solved by an online classifier and an offline intersection-over-union (IoU) network.

### 2.2. Tracking Benchmark

Benchmarks are crucial for the development of tracking. We roughly categorize existing benchmarks into two types: *generic* benchmark and *specific* benchmark.

**Generic Benchmark.** A generic tracking benchmark usually includes sequences for general scenes. OTB-2013 [57] is the first generic dataset with 50 sequences and later extended in larger OTB-2015 [57] by introducing extra videos. TC-128 [39] collects 128 colorful sequences to investigate the impact of color information on tracking performance. VOT [28] introduces a series of tracking competitions with up to 60 sequences. NfS [20] focuses on evaluating trackers on videos with high frame rate. NUS-PRO [31] offers 365 videos with the goal of performance evaluation on rigid objects. TracKlinic [18] provides 2,390 videos with a diagnosis goal of tracking algorithms under various challenges. Recently, to provide training data for developing deep trackers, many large-scale benchmarks have been proposed. OxUvA [54] provides 366 videos with the goal of long-term evaluation. TrackingNet [47] consists of more than 30K sequences for deep tracking. GOT-10k [25] offers 10K videos with rich motion trajectories for tracking. LaSOT [14] comprises 1,400 long-term videos with manual annotations. Later, LaSOT is extended in [13] by introducing 150 new video sequences and a new evaluation protocol for unseen objects with more analysis.

**Specific Benchmark.** Besides generic datasets, there exist other benchmarks for specific goals. UAV and UAV123 [45] consists of 100 and 23 videos captured by unmanned aerial vehicle (UAV). CDTB [42] and PTB [51] aim at assessing tracking performance on RGB-D videos. VOT-TIR [29] is from VOT and focuses on object tacking in RGB-T sequences.

Despite of the availability of the above benchmarks, they mainly focus on opaque object tracking. Tracking of transparent target objects, which widely appear in the real-world, has received *very little* attention. The most important reason is the lack of a dataset for transparent object tracking, which motivates our proposal of TOTB.

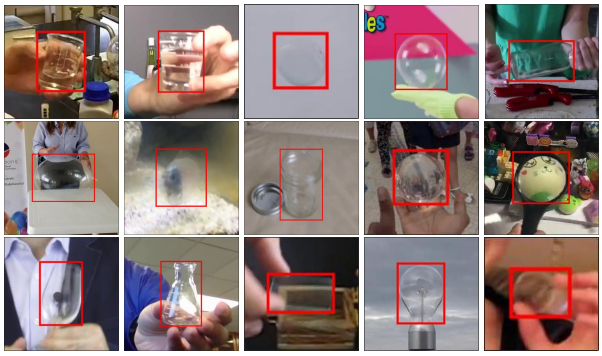


Figure 2. Samples from 15 transparent object categories. First row: Beaker, GlassCup, WubbleBubble, JuggleBubble and GlassBottle. Second row: BubbleBalloon, TransparentAnimal, GlassJar, GlassBall and MagnifyingGlass. Third row: WineGlass, Flask, GlassSlab, Bulb and ShotGlass. The tracking targets are shown in the red bounding boxes.

### 2.3. Dealing with Transparent Object in Vision

Transparent objects are common to see in the real-world, and a significant amount of research has been devoted to deal with them. For example, the methods of [19, 44] investigate the problem of transparent object recognition. The approach of [27] explores the time of flight (ToF) camera to detect and reconstruct transparent objects. The approach of [40] proposes to estimate keypoints of transparent objects in RGB-D images. The work of [49] studies the problem of 3D shape estimation for transparent objects in RGB-D images. The methods of [59, 26, 58] handle the task of segmenting transparent objects from an image. Especially, the work of [58] presents a large-scale benchmark for transparent object segmentation.

Our work is related to [40, 49, 58] but different in: (1) TOTB focuses on 2D object tracking, while other works on 3D shape estimation [49], 3D labeling and keypoint estimation [40] and 2D object segmentation [58]. (2) TOTB deals with transparent objects in videos, while [40, 49, 58] in static images.

## 3. Transparent Object Tracking Benchmark

We aim to construct a dedicated transparent object tracking benchmark (TOTB). When developing TOTB, we cover a diverse selection of transparent object classes and provide manual annotations for each video, as detailed later.

### 3.1. Video Collection

In TOTB, we select 15 transparent object categories consisting of *Beaker*, *GlassCup*, *WubbleBubble*, *JuggleBubble*, *GlassBottle*, *BubbleBalloon*, *TransparentAnimal*, *GlassJar*, *GlassBall*, *MagnifyingGlass*, *WineGlass*, *Flask*, *GlassSlab*, *Bulb* and *ShotGlass*. Note that, the transparent *window* and *door* widely appear in the real-world, nevertheless, the ob-

Table 1. Summary of statistics of the proposed TOTB. OV: out-of-view; FOC: full occlusion.

| | | | |
|---|---|---|---|
| Number of videos | 225 | Avg. duration | 12.7s |
| Total frames | 86K | Frame rate | 30 *fps* |
| Max frames | 500 | Absent labels | OV, FOC |
| Min frames | 126 | Object categories | 15 |
| Avg. frames | 381 | Number of att. | 12 |

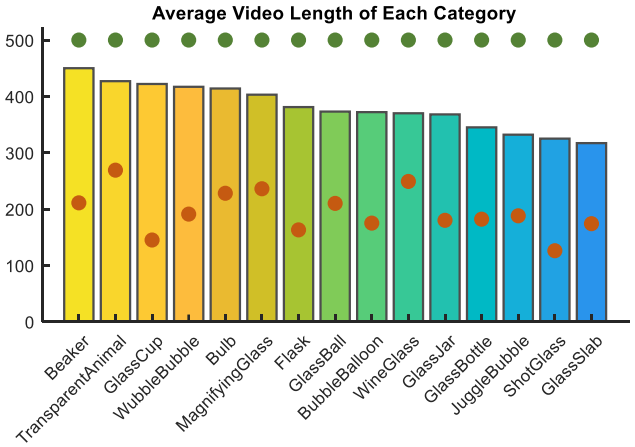



Figure 3. Average video length for each object class in TOTB. The green and brown dots represent the maximum and minimum frame numbers of each category.


Figure 4. Example sequences of transparent object tracking in our novel TOTB. Each sequence is annotated with axis-aligned bounding boxes.

jects of these two categories are usually static, and therefore not suitable for tracking task. Figure 2 demonstrates the samples from these 15 categories.

After determining object categories, we search for raw sequences of each class from YouTube[1], as it is the largest public video platform and motivates many tracking benchmarks (*e.g.*, LaSOT [13], TrackingNet [47], GOT-10k [25] and OxUvA [54]). Initially, we have collected at least 30 raw videos for each class and gathered more than 600 sequences in total. Then, we carefully inspect each sequence for its availability for tracking and choose 15 sequences for each category. We verify the content of each raw sequence and remove the irrelevant parts to acquire a video clip that is suitable for tracking. We limit the number of frames in each video up to 500, which is enough for testing a tracker's performance on transparent objects, while being manageable for annotation. Eventually, TOTB consist of 225 sequences from 15 transparent object classes with 86K frames. Table 1 summarizes TOTB, and Figure 3 demonstrates the average video length of each object category in TOTB.

### 3.2. Annotation

We follow the same principle as in [14] for sequence annotation: given the initial target in a video, for each frame, if the target appears, the annotator draws/edits an axis-aligned bounding box as the tightest one to fit any visible part of the target object; otherwise, an absence label, either *full occlusion* (FOC) or *out-of-view* (OV), is assigned to this frame.

With the above principle, we adopt a three-step strategy for annotation, including *manual labeling*, *visual inspection* and *box refinement*. In the first stage, each video is labeled by an expert, *i.e.*, a graduate student who works on tacking. Since there may exist unavoidable annotation errors or inconsistencies in the first stage, a visual inspection is performed in the second stage to verify the annotation. The inspection of annotation for each video is conducted by a validation team. If the annotation result is not unanimously agreed by the members of validation team, it will be sent back to the original annotator for refinement in the third step. Such three-step strategy ensures high-quality annotation boxes for transparent objects in TOTB. Some examples for box annotation of TOTB can be found in Figure 4. We show more statistics in *supplementary material*.

### 3.3. Attributes

Further in-depth analysis of tracking algorithms is important for researchers to understand trackers' strengths and limitations. Thus motivated, we select twelve attributes that widely exist in video tasks and annotate each sequence with these attributes, including (1) illumination variation (IV), (2) partial occlusion (POC), (3) deformation (DEF), (4) motion blur (MB), (5) rotation (ROT), (6) background clutter (BC), (7) scale variation (SV), which is assigned when the ratio of bounding box is outside the range [0.5, 2], (8) full

[1]Each video is collected under the Creative Commons license.

Table 2. Distribution of twelve attributes on the TOTB. The diagonal (shown in **bold**) corresponds to the distribution over the entire benchmark, and each row or column presents the joint distribution for the attribute subset.

| | IV | POC | DEF | MB | ROT | BC | SV | FOC | FM | OV | LR | ARC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IV | **69** | 24 | 7 | 16 | 43 | 5 | 20 | 2 | 10 | 2 | 3 | 16 |
| POC | 24 | **110** | 18 | 38 | 59 | 23 | 48 | 9 | 26 | 7 | 12 | 40 |
| DEF | 7 | 18 | **42** | 6 | 6 | 8 | 24 | 0 | 7 | 0 | 1 | 20 |
| MB | 16 | 38 | 6 | **69** | 50 | 16 | 29 | 7 | 18 | 6 | 5 | 27 |
| ROT | 43 | 59 | 6 | 50 | **123** | 21 | 59 | 7 | 27 | 6 | 9 | 61 |
| BC | 5 | 23 | 8 | 16 | 21 | **42** | 17 | 3 | 5 | 1 | 0 | 11 |
| SV | 20 | 48 | 24 | 29 | 59 | 17 | **95** | 0 | 33 | 0 | 14 | 68 |
| FOC | 2 | 9 | 0 | 7 | 7 | 3 | 0 | **10** | 0 | 3 | 0 | 0 |
| FM | 10 | 26 | 7 | 18 | 27 | 5 | 33 | 0 | **44** | 0 | 11 | 29 |
| OV | 2 | 7 | 0 | 6 | 6 | 1 | 0 | 3 | 0 | **9** | 0 | 0 |
| LR | 3 | 12 | 1 | 5 | 9 | 0 | 14 | 0 | 11 | 0 | **18** | 11 |
| ARC | 16 | 40 | 20 | 27 | 61 | 11 | 68 | 0 | 29 | 0 | 11 | **82** |

occlusion (FOC), (9) fast motion (FM), which is assigned when the target center moves by at least 50% of its size in last frame, (10) out-of-view (OV), (11) low resolution (LR), which is assigned when the region of the target is less than 900 pixels, and (12) aspect ratio change (ARC), which is assigned when the ratio of bounding box aspect ratio is outside the range [0.5, 2]. For each video, a 12D binary vector is provided to indicate the presence of an attribute (*i.e.*, "1" denotes the presence of a certain attribute, "0" otherwise.)

The distribution of these attributes on TOTB is presented in Table 2. We can observe that the most common challenge in TOTB is *rotation* (including in-place and out-plane rotations), which may cause serious feature misalignment and lead to tracking failure. In addition, the *scale variation* and *partial occlusion* frequently occur in videos of TOTB.

## 4. A New Baseline: TransATOM

As mentioned early, the technical difficulty of transparent object tracking is the weak appearance caused by transparency. To address this issue, we exploit transparency feature for transparent object tracking. Specifically, considering that the transparency is a *common* attribute of transparent objects, its feature should be *generic* and *transferable* for different transparent instances, and *differentiable* from opaque objects.

Inspired by [58], we learn such transparency feature with a deep segmentation network that classifies each pixel belonging to transparent regions. Different from [58] adopting a complex network, we utilize a much simpler FCN architecture [41] with ResNet-18 [23] for efficient inference. The images used for training our segmentation are borrowed from the training set in [58]. Note that, in our task, we only segment small and movable transparent objects. Thus, there are 2,844 static images for training. The details of the segmentation network for our task and its training are shown in *supplementary material* due to limited space.

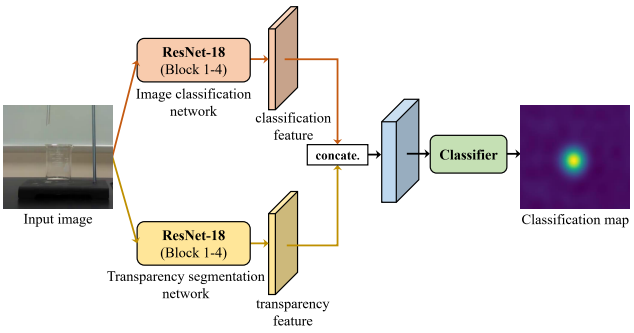



Figure 5. Illustration of architecture of TransATOM that integrates conventional classification feature and our proposed transparency feature for target localization.

After training the segmentation network, we apply it for extracting transparent features for transparent objects. We integrate such feature into state-of-the-art ATOM [8] to develop our new tracker, dubbed TransATOM. In particular, TransATOM consists of two feature branches. One branch is the pre-trained ResNet-18 for classification as in [8], and the other one is our trained segmentation network for transparency feature extraction. In both two branches, we extract features after block 4 and concatenate them for more robust feature representation. After that, we adopt a classification network to locate the target object. Figure 5 shows the classification architecture of TransATOM.

Similar to [8], the classification network consists of two convolutional layers and is formulated as follows,

$$f(X;\mathbf{w}) = \phi_2(w_2 * \phi_1(w_1 * X)) \tag{1}$$

where $\mathbf{w} = \{w_1, w_2\}$ represent parameters of network and $\phi_1$ and $\phi_2$ are activation functions after each convolutional layer. $X$ is input feature to the classifier and obtained by combining both pre-trained image classification feature $x_{\text{cls}}$ and transparency feature $x_{\text{trs}}$ (see Figure 5) as follows,

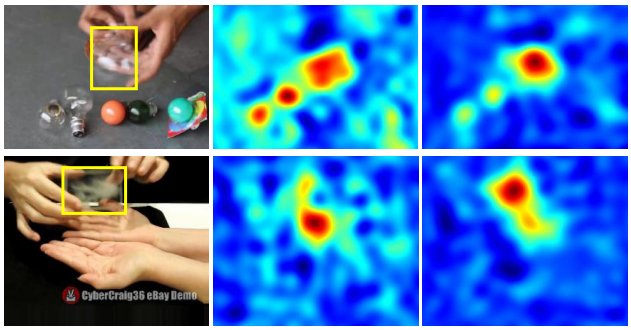
$$X = x_{\text{cls}} \| x_{\text{trs}} \tag{2}$$

where $\|$ denotes concatenation operation.

We use the $L2$ loss to learn the classifier via

$$\ell_{\mathbf{w}} = \sum_{j=1}^{M} \gamma_j \|f(X_j;\mathbf{w}) - Y_j\|^2 + \sum_k \lambda_k \|w_k\|^2 \tag{3}$$

where $X_j$ is the $j$-th training sample and $Y_j$ is its Gaussian label centered at target location; $\gamma_j$ and $\lambda_k$ control the sample weight and the regularization amount, respectively. We use the same optimization method as in [8] for learning and updating the classifier. For target scale estimation, we adopt IoU-Net as in [8]. Note that, in addition to the transparency




(a) Input image (b) Classification result of ATOM (c) Classification result of TransATOM

Figure 6. Classification results of ATOM and TransATOM. We can observe that TransATOM shows better classification results for locating transparent target objects. The yellow boxes in input images are groundtruth.

feature branch, the rest of TransATOM, including classification feature branch and IoU-Net, is directly borrowed from the baseline ATOM [8]. Please refer to [8] for more details.

Notice that, different from ATOM [8], TransATOM aims to explore additional transparency feature to improve localization of objects. Figure 6 shows target localization results of the two methods. We observe that TransATOM better locates the objects with the help of transparency features. Furthermore, our TransATOM runs in real-time at 26 *fps*.

It is worth mentioning that, the proposed transparency feature in TransATOM is *generic* and *transferable* to other trackers (*e.g.*, DiMP [3] and KYS [4]) for improvements as shown in our ablation study in Section 5.4.

# 5. Evaluation

## 5.1. Evaluation Methodology

Following [14, 47], we use one-pass evaluation (OPE) and measure each tracker using *precision*, *normalized precision* and *success*. The precision (PRE) measures the distance between centers of tracking results and groundtruth boxes in pixels. Different algorithms are ranked by their PRE score at a threshold (*e.g.*, 20 pixels). To eliminate the influence of different scales, normalized precision (NPRE) is adopted by performing normalization with target areas. Success (SUC) compares the intersection over union (IoU) of tracking results and groundtruth boxes, and SUC score is computed by the percentage of tracking results whose IoU is larger than 0.5.

## 5.2. Evaluated Trackers

We evaluate 25 state-of-the-art trackers on TOTB and provide basis for future comparison. These algorithms can be roughly categorized into three types: correlation filter trackers, Siamese trackers and other deep trackers.

Correlation filter tracking approaches include KCF [24], SRDCF [11], HCFT [43], Staple [1], ECOhc [7], ECO [7], STRCF [34], StapleCA[46], CFNet [53], BACF [21] and ASRCF [6]. The Siamese trackers consist of SiamFC [2], SiamRPN [33], DaSiamRPN [64], C-RPN [16], SPM [55], SiamRPN++ [32], SiamDW [61] and SiamMask [56]. For other trackers, we use MDNet [48], ATOM [8], DiMP [3], PrDiMP [9], DCFST [63] and KYS [4].

## 5.3. Evaluation Results

**Overall performance.** We extensively evaluate 25 tracking algorithms and our proposed TransATOM on 225 sequences in TOTB. Notice that, existing trackers are used without any modifications for evaluation. In order to avoid randomness, we run each tracker three times and average the results for its final performance. The evaluation results are reported in OPE using precision (PRE), normalized precision (NPRE) and success (SUC). Figure 7 displays the performance of 15 trackers and our TransATOM and we refer readers to the *supplementary material* for full results of all trackers. As demonstrated in Figure 7. TransATOM achieves the best results with 0.668 PRE, 0.747 NPRE and 0.641 SUC. SiamRPN++ obtains the second best PRE score of 0.647, SiamMask the second best NPRE score of 0.724 and PrDiMP the second best SUC score of 0.633. In comparison with these trackers, TransATOM achieves improvements of 2.1%, 2.3% and 0.8% in terms of PRE, NPRE and SUC, respectively. ATOM, which serves as the baseline of TransATOM, shows the results of 0.641 PRE, 0.717 NPRE, and 0.641 SUC. Compared to ATOM, TransATOM obtains significant performance gains of 4.1%, 3.0% and 2.7%, respectively, which evidences the effectiveness and advantage of transparency feature for transparent object tracking.

**Attribute-based performance.** In order to further analyze and understand the performance of different tracking algorithms, we conduct performance evaluation under twelve attributes. We demonstrate the results for the three most frequent challenges, including *rotation*, *partial occlusion* and *scale variation*, in Figure 8, and refer readers to *supplementary material* for full results.

We observe that TransATOM performs the best on partial occlusion and scale variation. Specifically, TransATOM achieves the SUC scores of 0.635 and 0.604 on partial occlusion and scale variation, outperforming the second best PrDiMP with SCU scores of 0.621 and 0.598 by 1.4% and 0.6%. On the challenge of rotation, PrDiMP shows the best result with 0.592 SUC score. TransATOM ranks the second with 0.591 SUC score, which is competitive compared with PrDiMP. It is worth noticing that, PrDiMP leverages deeper ResNet-50 for feature extraction, while TransATOM adopts ResNet-18. Despite this, TransATOM shows better or competitive performance in comparison with PrDiMP owing to the effective transparent features. Besides, on all three attributes, TransATOM significantly outperforms ATOM with

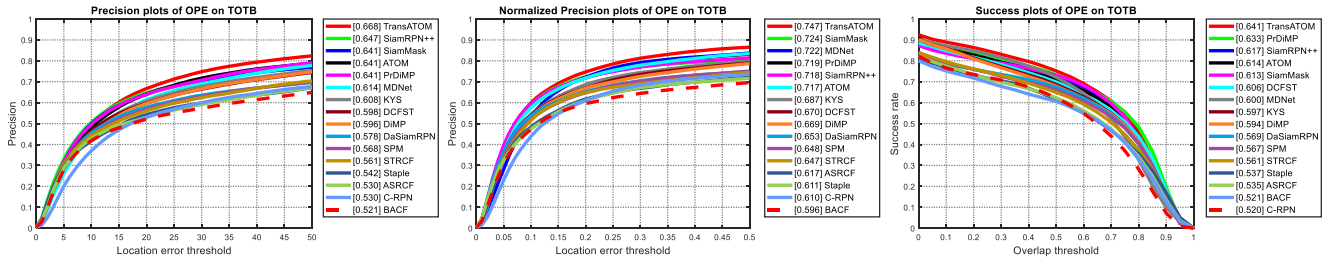



Figure 7. Tracking performance of 15 state-of-the-art trackers and TransATOM in terms of precision, normalized precision and success (please check the full results of all trackers in *supplementary material*). Our TransATOM achieves the best results with all three metrics.

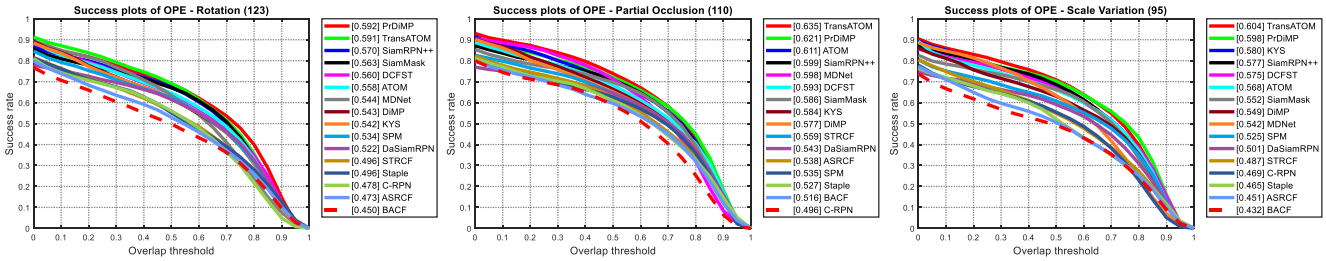



Figure 8. Tracking performance of different tracking algorithms on the three most common attributes in TOTB including *rotation*, *partial occlusion* and *scale variation* using success (please check the full results and comparisons of all trackers in *supplementary material*).

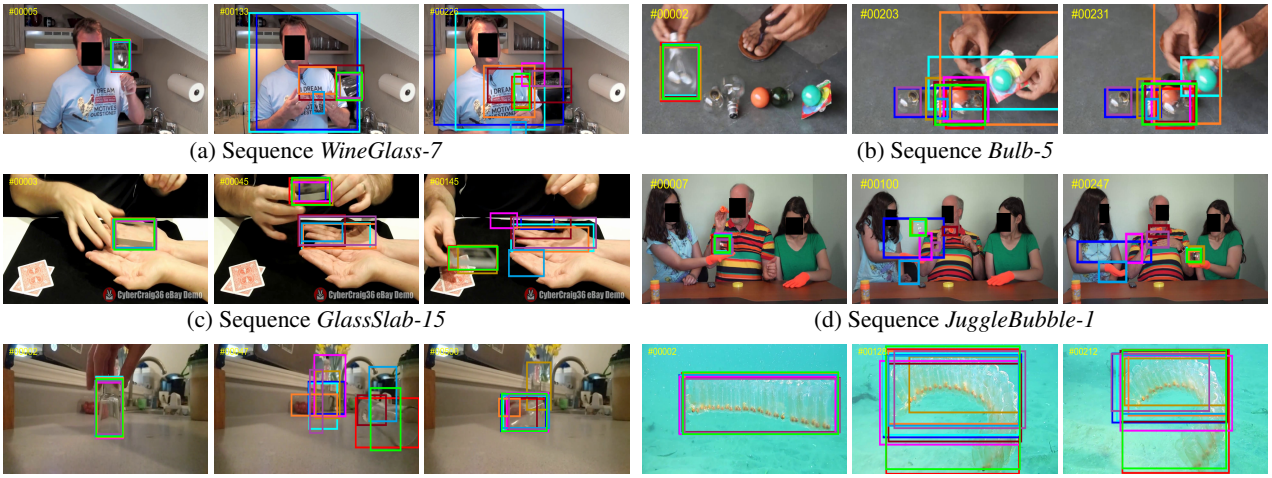



Figure 9. Qualitative results of nine trackers in six typical difficult challenges: *WineGlass-7* with rotation, *Bulb-5* with background clutter, *GlassSlab-15* with aspect ratio change, *JuggleBubble-1* with partial occlusion, *ShotGlass-10* with motion blur and *TransparentAnimal-11* with scale variation. The proposed TransATOM robustly locates target objects under various challenges owing to transparency feature.

SUC scores of 0.558, 0.611 and 0.568, showing the importance of transparency feature.

**Qualitative evaluation.** To better understand each tracking algorithm, we demonstrate qualitative tracking results of the top trackers, including TransATOM, PrDiMP, SiamRPN++, ATOM, SiamMask, DCFST, MDNet, KYS and DiMP, in six typical challenges consisting of *rotation*, *background clutter*, *aspect ratio change*, *partial occlusion*, *motion blur* and *scale variation* in Figure 9. From Figure 9, we observe that other trackers are able to deal with only one or several challenges. For example, PrDiMP performs well in dealing with aspect ratio change in *GlassSlab-15* but fails in other challenges. SiamRPN++ can locate the target in *ShotGlass-10* with motion blur while is prone to drift in *Bulb-5* with background clutter. MDNet works robustly in *WineGlass-7* with rotation but loses the target in *Bulb-5* with background clutter and *ShotGlass-10* with motion blur. Similar observations are found for other trackers. Different from these methods,

Table 3. Analysis of different backbones for tracking performance on TOTB using SUC score. The better one is shown in red font.

| | ResNet-18 | ResNet-50 |
|---|---|---|
| ATOM [8] | 0.614 | 0.608 |
| DiMP [3] | 0.605 | 0.594 |
| PrDiMP [9] | 0.639 | 0.633 |
| SiamRPN++ [32] | 0.585 | 0.617 |
| TransATOM (ours) | 0.641 | 0.632 |

TransATOM well handles all challenges for robust target localization owing to the transparency features. More qualitative results can be found at the project website.

### 5.4. Ablation Study

**Depth of backbone.** Deep neural network has significantly improved tracking performance. In opaque object tracking, many recently proposed deep trackers using ResNet-50 as backbone significantly outperform those using ResNet-18 as backbone because of deeper features. Nevertheless, when tracking transparent objects, deeper features do *not* always bring performance gains. In particular, we compare four representative state-of-art trackers including ATOM, DiMP, PrDiMP and SiamRPN++ on TOTB. Table 3 lists the comparison results using SUC scores. As displayed in Table 3, we observe that, when using deeper ResNet-50 as backbone, the SUC scores of ATOM, DiMP and PrDiMP are decreased from 0.614, 0.605, 0.639 to 0.608, 0.594 and 0.633, respectively. This indicates that the deeper features may hurt tracking performance for ATOM and DiMP. For SiamRPN++, when using deeper ResNet-50 as backbone, the SUC score is significantly improved from 0.585 to 0.617, showing the effectiveness of deeper features in Siamese tracker for transparent object tracking. Likewise, we conduct experiments of our tracker TransATOM using two backbones. As shown in Table 3, the performance of TransATOM with deeper ResNet-50 backbone is decreased in comparison with TransATOM with ResNet-18 backbone. By analyzing the impact of different backbones on tracking performance, we find that deeper features are *not* always beneficial for tracking of transparent objects. We hope this finding can provide a reference for transparent object tracker design in future.

**Transparency feature.** To facilitate development of tracking algorithm on TOTB, we propose TransATOM by integrating transparency feature, which is a generic characteristic for transparent object learned explicitly, into state-of-the-art ATOM. In order to analyze the effect of transparency feature, we compare three tracking algorithms including ATOM, TransATOM-V and TransATOM. TransATOM-V is implemented by removing visual classification feature branch from TransATOM. Except for features, all other settings are the same for three trackers. Table 4 shows the comparison results. Compared to ATOM with 0.614 SUC score, TransATOM-V obtains 0.625 SUC score with 1.1% absolute gain, demonstrating the effectiveness of transparency feature in boosting performance. Moreover, TransATOM, which combines visual and transparency features, further pushes the performance to 0.641 and still runs in real-time.

Table 4. Analysis of transparency feature on tracking performance in terms of accuracy and speed.

| | Visual feature | Transparency feature | SUC | Speed |
|---|---|---|---|---|
| ATOM [8] | ✓ | | 0.614 | 37 *fps* |
| TransATOM-V | | ✓ | 0.625 | 37 *fps* |
| TransATOM | ✓ | ✓ | 0.641 | 26 *fps* |

Table 5. Analysis of transferability of transparency feature.

| Trackers | SUC |
|---|---|
| ATOM [8] | 0.614 |
| TransATOM | 0.641 (↑2.7%) |
| DiMP [3] | 0.594 |
| TransDiMP | 0.613 (↑1.9%) |
| KYS [4] | 0.597 |
| TransKYS | 0.619 (↑2.2%) |

**Transferability of transparency feature.** Transparency is a *common* attribute of transparent objects, and transparency feature should be *generic* and *transferable*. To analyze its transferability, we integrate transparency feature into different trackers as shown Table 5, similar to TransATOM. We observe that, TransDiMP and TransKYS respectively improves their baseline DiMP and KYS by 1.9% and 2.2% gains, evidencing the transferability of transparency feature.

## 6. Conclusion

In this paper, we explore a new tracking task, *i.e.*, *transparent object tracking*. In particular, we propose the TOTB, which is the first benchmark for transparent object tracking, to our best knowledge. In addition, in order to understand the performance of existing trackers and to provide baseline for future comparison, we extensively evaluate 25 state-of-the-art tracking algorithms with in-depth analysis. Furthermore, we propose a novel tracker, named TransATOM, by leveraging transparency features of transparent objects. TransATOM significantly outperforms existing state-of-the-art tracking algorithms by a clear margin. We believe that, the benchmark, evaluation and the baseline tracker will inspire and facilitate more future research and application on transparent object tracking.

**Acknowledgment.** This work is supported in part by NSF Grant IIS-2006665 and IIS-1814745.

## References


[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016. 6

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 3, 6

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 3, 6, 8

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020. 6, 8

[5] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 2

[6] Kenan Dai, Dong Wang, Huchuan Lu, Chong Sun, and Jianhua Li. Visual tracking via adaptive spatially-regularized correlation filters. In *CVPR*, 2019. 2, 6

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 2, 6

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 2, 3, 5, 6, 8

[9] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 3, 6, 8

[10] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. 2

[11] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015. 2, 6

[12] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 2

[13] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *IJCV*, 129(2):439–461, 2021. 3, 4

[14] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 2, 3, 4, 6

[15] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017. 2

[16] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019. 3, 6

[17] Heng Fan and Haibin Ling. Cract: Cascaded regression-align-classification for robust visual tracking. In *IROS*, 2021. 3

[18] Heng Fan, Fan Yang, Peng Chu, Lin Yuan, and Haibin Ling. Tracklinic: Diagnosis of challenge factors in visual tracking. In *WACV*, 2021. 3

[19] Mario Fritz, Gary Bradski, Sergey Karayev, Trevor Darrell, and Michael J Black. An additive latent feature model for transparent object recognition. In *NIPS*, 2009. 3

[20] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017. 2, 3

[21] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017. 2, 6

[22] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, 2018. 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[24] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015. 2, 6

[25] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019. 2, 3, 4

[26] Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *CVPR*, 2020. 3

[27] Ulrich Klank, Daniel Carton, and Michael Beetz. Transparent object detection and reconstruction on a mobile platform. In *ICRA*, 2011. 3

[28] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *TPAMI*, 38(11):2137–2155, 2016. 3

[29] Matej Kristan et al. The visual object tracking vot2017 challenge results. In *ICCVW*, 2017. 3

[30] Matej Kristan et al. The visual object tracking vot2018 challenge results. In *ECCVW*, 2018. 2

[31] Annan Li, Min Lin, Yi Wu, Ming-Hsuan Yang, and Shuicheng Yan. Nus-pro: A new visual tracking challenge. *TPAMI*, 38(2):335–349, 2016. 3

[32] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 3, 6, 8

[33] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 3, 6

[34] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, 2018. 2, 6

[35] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *ICCV*, 2019. 3

[36] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *PR*, 76:323–338, 2018. 1, 2

[37] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM TIST*, 4(4):58, 2013. 1, 2

[38] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCVW*, 2014. 2

[39] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 24(12):5630–5644, 2015. 2, 3

[40] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *CVPR*, 2020. 3

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 5

[42] Alan Lukezic, Ugur Kart, Jani Kapyla, Ahmed Durmush, Joni-Kristian Kamarainen, Jiri Matas, and Matej Kristan. Cdtb: A color and depth visual object tracking dataset and benchmark. In *ICCV*, 2019. 2, 3

[43] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 2, 6

[44] Kazuki Maeno, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Light field distortion feature for transparent object recognition. In *CVPR*, 2013. 3

[45] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016. 2, 3

[46] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *CVPR*, 2017. 6

[47] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 2, 3, 4, 6

[48] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 6

[49] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *ICRA*, 2020. 3

[50] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2014. 1, 2

[51] Shuran Song and Jianxiong Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *ICCV*, 2013. 3

[52] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. 3

[53] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017. 6

[54] Jack Valmadre, Luca Bertinetto, João F Henriques, Ran Tao, Andrea Vedaldi, Arnold Smeulders, Philip Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *ECCV*, 2018. 2, 3, 4

[55] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019. 3, 6

[56] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 6

[57] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 2, 3

[58] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *ECCV*, 2020. 3, 5

[59] Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *ICCV*, 2015. 3

[60] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM CSUR*, 38(4):13, 2006. 1, 2

[61] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019. 3, 6

[62] Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 3

[63] Linyu Zheng, Ming Tang, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Learning feature embeddings for discriminant model based tracking. In *ECCV*, 2020. 6

[64] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018. 3, 6