# Occlusion Detection in Visual Tracking: A New Framework and A New Benchmark

Xiaoguang Niu[1], Yueyang Gu[1], Zhifeng Lu[2], Zehua Hong[2], Yi Tian[2], Kuan Xu[1], Jie Yang[1], Xingqi Fang[1], and Yu Qiao[1(✉)]

[1] Institute of Image Processing and Pattern Recognition, Department of Automation, Shanghai Jiao Tong University, Shanghai, China
qiaoyu@sjtu.edu.cn
[2] Shanghai Electro-Mechanical Engineering Institute, Shanghai, China

**Abstract.** Occlusion remains being a challenge in visual object tracking. The robustness to occlusion is critical for tracking algorithms, though not much attention has been paid to it. In this paper, we first propose an occlusion detection framework which calculates the proportion of the target that is occluded, hence to decide whether to update the model of target. This framework can be integrated with existing tracking algorithms to increase their robustness to occlusion. Then we introduce a new benchmark which contains sequences where occlusion is the main difficulty. The sequences are chosen from public benchmarks and are fully annotated. The proposed framework is combined with several standard trackers and evaluated on the new benchmark. The experimental results show that our framework can improve the tracking performance, with explicit incorporation of occlusion detection.

**Keywords:** Visual tracking · Occlusion detection · Benchmark

## 1 Introduction

Generic object tracking [1,3,5–7,12–14], where the tracker is not specialized to any specific category of objects, is a popular research field in recent years. Because of the category-agnostic, it is not possible to train a detector offline for a particular type of objects, such as pedestrians or hands. Consequently, occlusion is the most challenging factor for generic object trackers [8], since the trackers usually cannot discriminate the occluders from the targets.

Majority of the work in handling occlusion is to add a sub-module before target model updater to monitor the tracking reliability. In [20], the feedback from tracking results is utilized to decide whether or not to update the target model. However, this strategy still cannot tell what is actually happening, occlusion or target appearance variation, both of which will decrease the tracking confidence.

This research is partly supported by USCAST2015-13, USCAST2016-23, SAST2016008, NSFC (No: 61375048).


L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11304, pp. 579–587, 2018.
https://doi.org/10.1007/978-3-030-04212-7_51

COD (Context-based Occlusion Detection for Tracking) [15–17] is a framework that monitors the background-patches around the target and can identify which of them occlude the target. However, several drawbacks exist. First, the number of background-patches that COD monitors is constant, which contaminates the adaptive ability of the framework. Furthermore, determining the occlusion occurrence simply by the number of occluders over-simplifies the problem and is not guaranteed to be reasonable in all occasions. To solve these issues, we present Adaptive COD, which is adaptive to differently sized targets and able to identify what proportion of the target is affected by occlusion. The number of background-patches is now dependent on the perimeter of the target, hence more background-patches will be allocated to deal with a larger target. After acquiring the positions of the background-patches that occlude the target, we calculate the proportion of the target that is under occlusion. If the proportion is greater than a threshold, model updater will not take any action, avoiding the model being corrupted. The background-patches that occlude the target continues to be monitored, while other background-patches are discarded and new ones will be generated around the new target. As a general framework, Adaptive COD can be integrated with any existing tracking algorithm to address the occlusion problem.

To better evaluate the performance of different trackers and promote the development of tracking algorithms, several benchmarks have been built. OTB [21], VOT [10], and ALOV [19] are the most widely used ones. In OTB [21], each sequence is tagged with 9 attributes, including occlusion, illumination variation and so on, which represent the challenging factors in visual tracking. A sequence will be tagged with attribute 'occlusion' if there are frames in the sequence where occlusion happens. In VOT [10], the attribute annotation is further refined to per-frame level. Later in NUS-PRO [11], the occlusion is classified into three levels: no occlusion, partial occlusion and full occlusion. Recently, attribute-specific benchmarks appear. In [18], a dataset for fast moving objects is collected. A higher frame rate video dataset is proposed in [4]. Although occlusion is one of the attributes in OTB [21] and VOT [10], the frames where occlusion happens only take up a small proportion of the overall sequence. Moreover, before the tracker meets these frames, the tracking results have already drift from the groundtruth, which means that different trackers will have different initialization setups in terms of evaluating their robustness to occlusion. In this paper, we build an attribute-specific benchmark which contains sequences where the target undergoes occlusion. In our proposed dataset, we exclude other attributes and only preserve the frames relevant to occlusion. Each sequence contains three parts: before, during and after occlusion. We evaluate our model updating strategy by integrating it with several mediocre tracking algorithms, including KCF [7], SAMF [14], DSST [3] and Staple [1]. The experimental results show that the Adaptive COD improves the robustness of these tracking algorithms.

**Algorithm 1.** (Adaptive) COD

Initialize target tracker and background-patch trackers;
**for** $t = 2$ to $T$ **do**
  Track the target and output target tracking result;
  Track the background-patches and identify occlusion;
  If no occlusion, update target tracker;
  Update background-patch trackers.
**end for**

In summary, the main contributions of this paper are as follows:

1. We improve the occlusion detection framework in [17]. The number of background-patch trackers is adaptive to the size of target. A new model updating strategy is proposed.
2. We establish a new dataset where the sequences contain occlusion for evaluating the robustness of tracking algorithms.
3. Extensive experiments demonstrate the effectiveness of our occlusion detection framework and occlusion benchmark.

## 2 Occlusion Detection Framework

In this section we first briefly review the Context-based Occlusion Detection for Tracking (COD) framework [17]. Then the proposed Adaptive COD is presented.

### 2.1 COD Review

Based on the assumption that both target and background-patches are involved in occlusion, COD [17] pays attention to the background around the target to *actively* detect occlusion. As is shown in Algorithm 1, two kinds of trackers exist in the framework: target tracker and background-patch trackers. Target tracker estimates the bounding box of target in the current frame, while the background-patch trackers provide the position and tracking reliability of every background-patch surrounding the target. Intuitively, if the bounding boxes of a background-patch and the target overlap and that the background-patch has high tracking reliability (hence it is not occluded by the target), then the target is occluded by the background-patch. Please refer to [17] for more details.

However, COD has the following disadvantages. Firstly, the number of background-patches $N_1$ is constant for variously sized targets in different sequences. For small targets, $N_1$ is relatively too large. Therefore, many background-patches overlay with each other, causing the double counting and repeated calculation. For large objects, $N_1$ becomes relatively small, so the background around the target is not fully monitored. Secondly, the target model will be updated online if the number of background-patches that occlude the target, $N$, is greater than a constant threshold $N_{th}$. Similarly, for targets of different sizes, $N$ as merely a counting result cannot properly measure the degree of occlusion.

### 2.2 Adaptive COD

We propose an Adaptive COD to overcome the limitations of COD mentioned in Sect. 2.1. Adaptive COD inherits the structure from COD but differs in two important aspects: the initialization step and the criterion for identifying occlusion. They are shown in Algorithm 1.

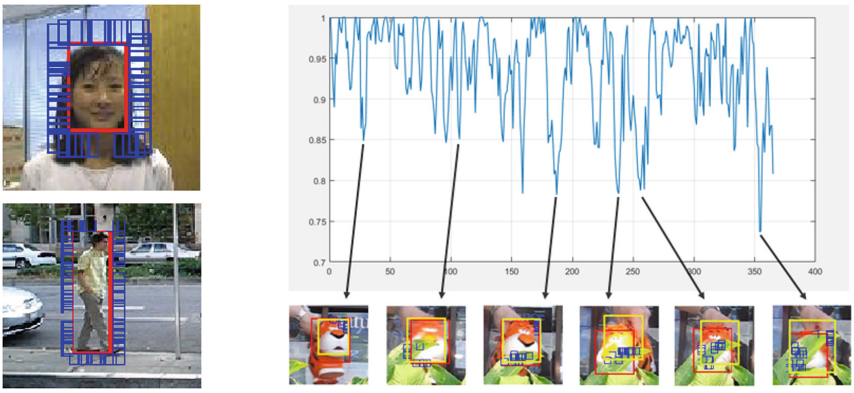


**Fig. 1.** In left, the number of background-patches for sequence *Girl* is 38, while for sequence *David3* it is 83. In right, the curve shows non-occluded proportion of the target for every frame in sequence *Tiger2*, along with the the frames #27,#107,#186,#238,#256,#355, corresponding to local minima of the curve. The blue boxes show where the occlusion happens.

Denote the bounding box of target in frame $t$ as $(x_t, y_t, w_t, h_t)$ for $t = 1, ..., T$, where $(x_t, y_t)$ are the upper-left corner point coordinates and $(w_t, h_t)$ are the width and height. Then we set $N_1 = [\ (w_1 + h_1)/2\ ]$, where $[x]$ will round $x$ to its nearest integer. In this way, the number of background-patches is dependent on the size of target. Unless the scale of target varies heavily, we keep using $N_1$ in the following frames. The results can be seen in Fig. 1.

We propose a new criterion for identifying occlusion. For target with parameter $(x_t, y_t, w_t, h_t)$, we build a mask $M_t$ as follows:

$$M_t(x, y) = \begin{cases} 1, if\ x \in [x_t, x_t + w_t]\ \&\&\ y \in [y_t, y_t + h_t] \\ 0, otherwise \end{cases} \tag{1}$$

I.e., $M_t$ has the same size of frame and the region representing the target is set as 1. The area of target region is $A_t = \sum M_t$. Similarly, for a background-patch with parameter $(bx_t^i, by_t^i, bw_t^i, bh_t^i)$ for $i = 1, 2, ..., N_1$, we build a mask $m_t^i$. Denoting the tracking reliability of background-patch $i$ as $r_t^i$ which is usually calculated as Peak-to-Sidelobe Ratio [2], we update $M_t$ as

$$M_t = \begin{cases} M_t\ -\ m_t^i, & if\ r_t^i\ >\ r_{th} \\ M_t, & otherwise \end{cases} \tag{2}$$

where $r_{th}$ is the threshold. After inspecting every background-patch and updating $M_t$, the area of target that is not occluded is $S_t = \sum M_t$. We use $\gamma_t = S_t \,/\, A_t$ as the measurement of occlusion, as is demonstrated in Fig. 1. Compared with using $N$ as the indicator of occlusion in COD, the new area-based adaptive criterion makes sense for targets of any size.

After identifying occlusion, the algorithm makes decision on whether to update the target tracker. The background-patches that are identified as occluders will continue to be monitored. Meanwhile, the algorithm will not pay attention to the other background patches which does not occlude the target and new background patches around the target in current frame will be added in the monitoring set.

## 3 Occlusion Benchmark

In this section, we present a new specialized benchmark for evaluating the robustness of tracking algorithms to occlusion. The benchmark is available at https://pan.baidu.com/s/1qZ0KeoW.

Although occlusion is one of the attributes in OTB [21], VOT [10] and NUS-PRO [11], these benchmarks still cannot accurately reflect the robustness of tracking algorithms to occlusion, due to the following reason. Each sequence usually has multiple challenging factors. Suppose a sequence $s$ with frames (#1,...,#$t_1$,...,#$t_2$,...,#$T$), where the occlusion happens in frames between #$t_1$ and #$t_2$. Since all the trackers start tracking in frame #1, they will have different tracking outputs before the occlusion occurs in frame #$t_1$, which means that the performance on frames between #$t_1$ and #$t_2$ is heavily influenced by the previous frames. As a recent study [9] shows, performance measures computed on a sequence are significantly biased to the dominant attribute of the sequence. Moreover, besides occlusion, there may exist other challenging factors in frames between #$t_1$ and #$t_2$, which makes the evaluation more unreliable.

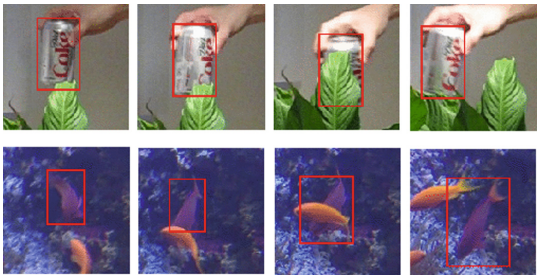

**Fig. 2.** Sequences in our occlusion benchmark can be divided into three parts. The first column shows the first frames of sequences *Coke_1* and *fish2_1*. The second and third columns show the targets being occluded. The last column shows targets after occlusion.

**Table 1.** Statistics about our occlusion benchmark.

| Sequence sources | # | Target categories | # |
|---|---|---|---|
| From OTB | 31 | Person | 19 |
| From VOT | 12 | Object | 15 |
| From NUS-PRO | 8 | Animal | 4 |
| Total sequences | 51 | Face | 8 |
| Total frames | 2628 | Other | 5 |

Based on these observations, we propose an occlusion benchmark that has the following characteristics:

1. Each sequence $s$ with frames (#1,...,#$t_1$,...,#$t_2$,...,#$T$) can be divided into 3 sub-sequences. In the first sub-sequence with frames (#1,...,#$t_1$), neither occlusion nor other challenging factor occur, so the target model can be initialized. In the second sub-sequence with frames (#$t_1$,...,#$t_2$), the target is occluded. In the last sub-sequence with frames (#$t_2$,...,#$T$), occlusion disappears so we can identify if the tracking succeeds. See Fig. 2 for explanation.
2. In frames (#$t_1$,...,#$t_2$), we exclude other attributes such as deformation, so that the only difficulty for tracking is to handle occlusion. However, it is a common scenario that the occluders are of the same category as the targets and have similar appearance, so we keep these sequences in the benchmark.
3. The sequences are selected from OTB [21], VOT [10] and NUS-PRO [11] with diversity and richness. The statistics is shown in Table 1.

In our occlusion benchmark, we propose a new metric called Normalized Center Location Error (NCLE) for evaluating performance. For tracking result $(cx_1, cy_1, w_1, h_1)$ and ground-truth $(cx, cy, w, h)$ where $(cx_1, cy_1)$ and $(cx, cy)$ are center locations, the traditional CLE adopted by OTB [21] is defined as

$$CLE = \sqrt{(cx_1 - cx)^2 + (cy_1 - cy)^2}. \tag{3}$$

A constant number, 20-pixel, is used for ranking trackers. However, for differently shaped and sized targets, 20-pixel deviation may have distinct meanings. For example, the width of a pedestrian target is usually smaller than the height, so the deviation is more serious if it is in the horizontal direction. In NCLE, we normalize the CLE by the width and height of target:

$$NCLE = min\{\ max\{\frac{|cx_1 - cx|}{w}, \frac{|cy_1 - cy|}{h}\},\ 1\ \}. \tag{4}$$

NCLE = 1 means a tracking failure. We utilize NCLE-based Precision Plot and Success Plot [21] as performance measurements in our occlusion benchmark.

## 4 Experiments

In this section, we present the experimental results of several recent tracking algorithms evaluated on our occlusion benchmark, including KCF [7],

SAMF [14], DSST [3] and Staple [1]. Meanwhile, we integrate these trackers into our adaptive COD framework to validate its effectiveness. All the code is available at https://github.com/xgniu/Occlusion-Benchmark.

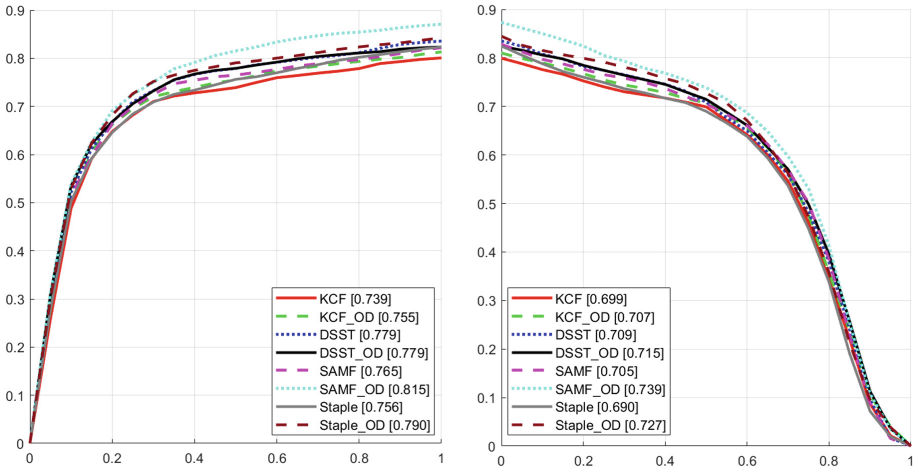


**Fig. 3.** The quantitative evaluation results. Left: NCLE-based Precision Plot. The numbers in brackets are the proportion of frames that have NCLE less than 0.5. Right: Success Plot.

**Table 2.** Different $\gamma$ for different tracking algorithms. Our framework is not sensitive to the value of $\gamma$.

| Precision | baseline | $\gamma$=0.90 | $\gamma$=0.85 | $\gamma$=0.8 |
|---|---|---|---|---|
| KCF | 0.739 | 0.755 | 0.758 | 0.760 |
| DSST | 0.779 | 0.779 | 0.779 | 0.795 |
| SAMF | 0.765 | 0.815 | 0.809 | 0.803 |
| Staple | 0.756 | 0.790 | 0.783 | 0.783 |

| Success | baseline | $\gamma$=0.90 | $\gamma$=0.85 | $\gamma$=0.8 |
|---|---|---|---|---|
| KCF | 0.699 | 0.707 | 0.714 | 0.720 |
| DSST | 0.709 | 0.715 | 0.716 | 0.732 |
| SAMF | 0.705 | 0.739 | 0.724 | 0.723 |
| Staple | 0.690 | 0.727 | 0.721 | 0.719 |

### 4.1 Quantitative Evaluation

The quantitative evaluation results are shown in Fig. 3 in the form of Precision Plot and Success Plot. All the four trackers gain improvements in performance after being integrated into our adaptive occlusion detection framework. Moreover, we find that though different tracking algorithms require differently valued $\gamma$ for best performance, a wide range of $\gamma$ can provide comparable results (Table 2). The other thresholds are the same as in COD [17].

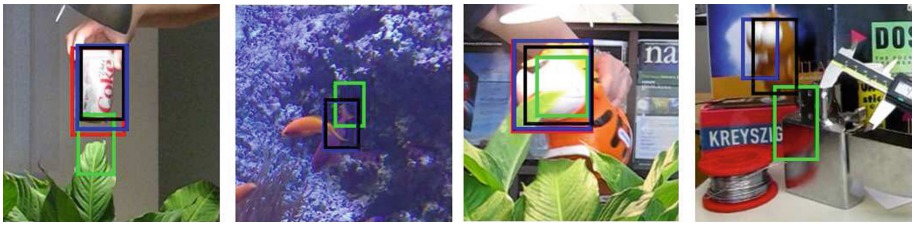


**Fig. 4.** The qualitative evaluation results. Red: SAMF. Blue: SAMF_OD. Green: Staple. Black: Staple_OD. The four sequences are *Coke*, *fish*, *Tiger2* and *Lemming* (Color figure online).

### 4.2 Qualitative Evaluation

Figure 4 visualizes several sequences from our occlusion benchmark along with the tracking results of different algorithms. Only the tracking results of SAMF, SAMF_OD, Staple and Staple_OD are shown for clarity, where the suffix '_OD' stands for being integrated into our occlusion detection framework. As the figure shows, when occlusion occurs, SAMF_OD and Staple_OD outperform their baselines.

## 5 Conclusion

Based on COD [17], we propose an adaptive occlusion detection framework which calculates the proportion of target that is not occluded. To better evaluate the robustness of tracking algorithms to occlusion, we propose an occlusion benchmark that excludes other challenging factors. In our benchmark, normalized center location error is adopted as the performance measure. Much work is needed in future to solve the occlusion problem for robust visual object tracking.

## References

1. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1401–1409 (2016)
2. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2544–2550. IEEE (2010)
3. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference (BMVC). BMVA Press, Nottingham, 1–5 September 2014
4. Galoogahi, H.K., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: a benchmark for higher frame rate object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1134–1143 (2017)

5. Gu, K., Zhou, T., Liu, F., Yang, J., Qiao, Y.: Correlation filter tracking via bootstrap learning. In: IEEE International Conference on Image Processing, pp. 459–463 (2016)
6. Gu, K., Zhou, T., Liu, F., Yang, J., Qiao, Y.: Patch-based object tracking via locality-constrained linear coding. In: Proceedings of the 35th Chinese Control Conference, pp. 7015–7020 (2016)
7. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
8. Kristan, M., Leonardis, A., Matas, J., Felsberg, M.: The visual object tracking VOT2017 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, pp. 1949–1972 (2017)
9. Kristan, M., et al.: A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2137–2155 (2016)
10. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L.: The visual object tracking vot2013 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, pp. 564–586, December 2013
11. Li, A., Lin, M., Wu, Y., Yang, M., Yan, S.: Nus-pro: a new visual tracking challenge. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 335–349 (2016)
12. Li, Q., Qiao, Y., Yang, J.: Robust visual tracking based on local kernelized representation. In: IEEE International Conference on Robiotics and Biomimetics, pp. 2523–2528 (2014)
13. Li, Q., Qiao, Y., Yang, J., Bai, L.: Robust visual tracking based on online learning of joint sparse dictionary. In: International Conference on Machine Vision (2013)
14. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8926, pp. 254–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_18
15. Niu, X., Cui, Z., Geng, S., Yang, J., Qiao, Y.: Robust visual tracking via occlusion detection based on depth-layer information. In: International Conference on Neural Information Processing, pp. 44–53 (2017)
16. Niu, X., Fang, X., Qiao, Y.: Robust visual tracking via occlusion detection based on staple algorithm. In: Asian Control Conference, pp. 1051–1056 (2017)
17. Niu, X., Qiao, Y.: Context-based occlusion detection for robust visual tracking. In: IEEE International Conference on Image Processing, pp. 3655–3659 (2017)
18. Rozumnyi, D., Kotera, J., Sroubek, F., Novotn, L., Matas, J.: The world of fast moving objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
19. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
20. Wang, M., Liu, Y., Huang, Z.: Large margin object tracking with circulant feature maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21–26 (2017)
21. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418 (2013)