Context-based Occlusion Detection for Robust Visual Tracking

Conference Paper · December 2017

DOI: 10.1109/ICIP.2017.8296964

2 authors:

Xiaoguang Niu
Shanghai Jiao Tong University

Yu QIAO
Shanghai Jiao Tong University

CONTEXT-BASED OCCLUSION DETECTION FOR ROBUST VISUAL TRACKING

Xiaoguang Niu, Yu Qiao *

Intelligence Learning Laboratory, Institute of Image Processing and Pattern Recognition,
Department of Automation, Shanghai Jiao Tong University, China

ABSTRACT

Occlusion is one of the most challenging factors in visual tracking. In this paper, we propose a novel context-based occlusion detection algorithm for robust visual tracking. The basic idea of our algorithm is that occlusion indicates that some background points in previous frame move into the target region in current frame. Our algorithm investigates background patches with background trackers. The occlusion is examined by the a occlusion detector. The template updating strategy is that if occlusion is detected, the target template stops updating. Comprehensive experiments in CVPR2013 Online Objecting Tracking Benchmark (OOTB) show that our tracker achieves comparable performance with other state-of-art trackers.

Index Terms— Visual tracking, occlusion detection, background tracker, template update, correlation filter

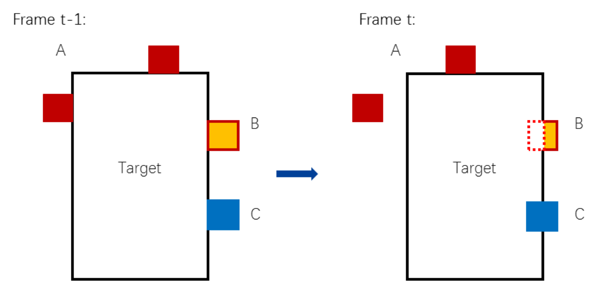

Fig. 1. Three different kinds of background patches. Class #A (red): patches that are left behind; Class #B (orange): patches that are covered by target; Class #C (blue): patches that overlap the target. The number of patches belonging to Class #C indicates whether there is occlusion.

1. INTRODUCTION

Visual object tracking is one of the most widely used techniques in computer vision. Although significant progress has been achieved, partial occlusion, deformation and scale variations, are still difficult problems for accurate and efficient tracking. In this paper, we mainly tackle the problem of partial occlusion and develop an effective approach for occlusion detection based on context information.

In recent years, correlation filter-based tracking (CFTs) algorithms have achieved amazing performance. D. S. Bolme *et al.* [1] proposed Minimum Output Sum of Squared Error (MOSSE), where the appearance of target is modeled by correlation filter and by correlating the trained filter with candidates of target, the position of target in the current frame is the position which has the strongest response. Based on MOSSE, various improvements have been made to handle challenges in tracking. J. F. Henriques *et al.* [2] introduced circulant matrix, kernel method and ridge regression to improve the performance, which is known as Kernelized Correlation Filter (KCF) tracker. M. Danelljan *et al.* [3] introduced Color-Name (CN) attribute to feature extractor. To handle scale variations of target, Y. Li *et al.* [4] introduced scaling pool to find the target in different scales. Z. Hong *et al.* proposed MUlti-Store Tracker (MUSTer) [5], which is inspired by cognitive psychology and consists of short- and long-term memory stores to process target appearance memories. In [6], C. Ma *et al.* use learned CNN features to construct multiple correlation filters on each convolutional layers and hierarchically infer the maximum response of each layer to locate the target.

Part-based tracking methods have favorable property of robustness against partial occlusion. The appearance of the object is modeled by multiple parts. However, how to effectively exploit the confidence scores of individual parts to construct a robust tracker is an open problem. Part Matching Tracker (PMT) [7] simultaneously matches parts in each of multiple frames, which is realized by a locality-constrained low-rank sparse learning method that establishes multi-frame part correspondences through optimization of partial permutation matrices. T. Liu *et al.* [8] adopted correlation filters as part trackers and developed a new criteria to measure the performance of different parts to assign proper weights. Reliable Patch Trackers (RPT) [9] identifies the reliable patches by motion trajectories information and exploits them to estimate the target state.

Occlusion may lead to the appearance variation of target, like target deformation and illumination variation. However, the requirement of template updating strategy for occlusion

*Corresponding author: Yu Qiao, qiaoyu@sjtu.edu.cn. This research is partly supported by NSFC, China (No: 61375048).

is opposite from that for target deformation and illumination variation. In cases of occlusion, the target template should be keep unchanged such that unwanted occlusion information may be excluded from the template and will not affect subsequent tracking. On the contrary, in cases of target deformation and illumination variation, the target template should be timely updated to learn the target variation. Therefore the key problem of designing desirable template updating strategy is the discrimination of occlusion from target deformation and illumination variation.

The classical definition of occlusion is that a point of a given image is occluded in the consecutive image if it is not visible by the observer in the latter [10]. However, this definition is not suitable for online tracking because correspondence detection between the points in two frames is difficult and of heavy computational burden. In this paper, we propose a new definition for occlusion: if some background points in previous frames move into the target region in current frame and block parts of target, the target is regarded as being occluded. This occlusion definition focus on the relationship between target and background instead of the target features like visibility or appearance variation.

In order to distinguish occlusion from other challenging factors, we analyze the state of background surrounding the target. If there is no occlusion, the background states may be one of two cases: (A) to be left behind by the target or (B) to be covered by the target. But if there is occlusion, some background points will be the case: (C) to move into the bounding box of target and block the corresponding part of target. Fig. 1. Case (C) happens only in occlusion other than deformation or other cases. Therefore the investigation of case (C) is of key importance for occlusion detection.

Based these observations, we propose a novel occlusion detection algorithm for robust visual tracking. Our algorithm includes the tracker of target and trackers of background. The background around the target is divided into multiple patches. The number of background patches that overlap the target indicates whether there is occlusion. The template updating strategy is accomplished by the template updater based on occlusion information.

2. OUR ALGORITHM

2.1. Overview

In this section, we introduce the proposed visual tracking algorithm based on occlusion detection, which takes into account both spatial and temporal context information.

The framework of our algorithm is shown in Fig. 2. There are two kinds of trackers: the target tracker and the background trackers. The target tracker estimates the bounding box of the target, while the background trackers deal with the background patches surrounding the target. The occlusion detector utilizes the location information about target and background patches for occlusion detection. The template updater makes use of the result of occlusion detector for applying appropriate target template updating strategy. It may prevent the target template from erroneous updating due to occlusion. Therefore our tracking algorithm is robust against occlusion.

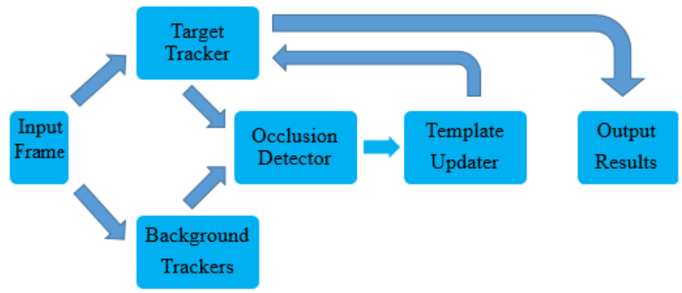

Fig. 2. The framework of our algorithm. Tracking module consists of target tracker and background trackers, whose results are fed into occlusion detector to decide whether to update the template of target.

2.2. Target Tracker

We use a variant of Gaussian KCF [2] as the target tracker . The objective function of linear ridge regression can be formulated as

$$\min \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2, \quad (1)$$

where x_i denotes the i-th input with label y_i, and the function f is $f(x_i) = w^T x_i$. In order to boost the performance, the Gaussian kernel is introduced,

$$k^{xx'} = exp(-\frac{1}{\sigma^2}(\|x\|^2 + \|x'\|^2 - 2F^{-1}(\hat{x}^* \odot \hat{x}'))), \quad (2)$$

where $k^{xx'}$ denotes kernel function, F^{-1} inverse DFT, $*$ complex conjugation.

The target tracker outputs the bounding box of target in current frame and will be updated by the target template updater according to the result of occlusion detector.

2.3. Background Tracker

Most of the existing part-based tracking methods divide the *target* into several parts and observe whether these parts can be tracked with high confidence. However, if there is appearance variation, the corresponding tracking confidence will also drop. So the tracker may falsely detect non-existent occlusion and miss the template update, which may lead to the failure of future tracking.

According to our definition for occlusion, occlusion indicates that some parts of background overlap the target bounding box. Therefore, unlike part-based tracking methods, we utilize the *background* information for occlusion detection. We divide the background surrounding the target into a number of patches and check whether there are background patches overlapping the target. In this way, occlusion and target appearance variation can be accurately distinguished.

Since there is no prior knowledge about occlusion, we select the patches uniformly and randomly along all four sides of the target bounding box. We apply linear KCF tracker as the background tracker since it is more efficient than Gaussian KCF, with

$$k^{xx'} = F^{-1}(\hat{x}^* \odot \hat{x'}). \tag{3}$$

The background tracker outputs the location of individual background patch in current frame.

2.4. Occlusion Detector

The occlusion detector works with the the location information about target and background patches obtained from the target tracker and background trackers respectively. All background patches are examined based on their location relationship with the target. If a patch does not overlap the target bounding box, it does not influence the occlusion detection. If it overlaps the the target bounding box, this patch may be occluded by the target or may occlude the target. We use the the peak-to-sidelobe ratio (PSR) as the tracking confidence to identify these two situations:

$$PSR = \frac{max(R) - avg(R)}{\sigma(R)}, \tag{4}$$

where R denotes the response map in KCF (for details about the response map, please refer to [2]), and σ the standard deviation. If the tracking confidence of a patch is lower than a threshold, it may be blocked by the target. Here we make some simplifications, ignoring the fact that the low confidence may be attributed to appearance change or other situations. If a patch overlaps the target bounding box with PSR above the threshold, it very likely blocks the target.

The occlusion detector classifies the background patches into 3 classes with the classification criterion as follows:

1. If a patch does not overlap the target bounding box, then it belongs to Class #A.

2. If a patch overlaps the target bounding box with low tracking confidence, it belongs to Class #B.

3. If a patch overlaps the target bounding box with high tracking confidence, it belongs to Class #C.

Obviously, the background patches in Class #C represent the occlusion information to some extent, which may be helpful for target template updating.

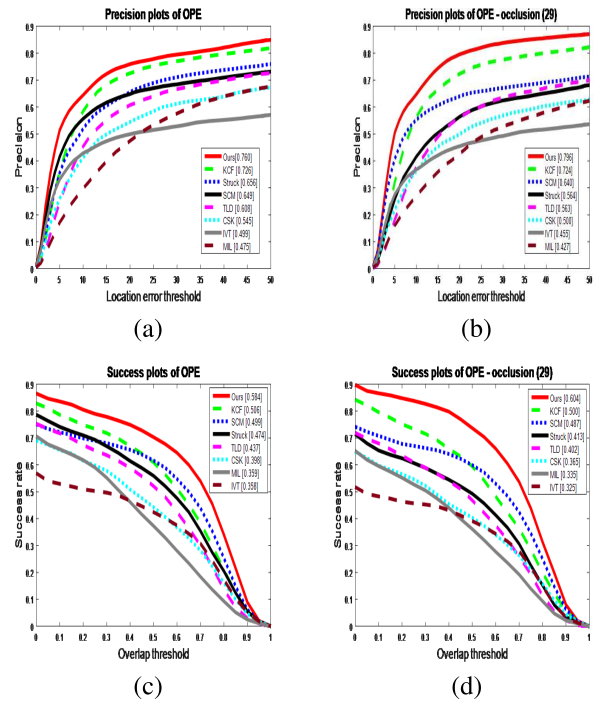

Fig. 3. Quantitative Evaluation. (a) Precision plot of trackers on OOTB. (b) Precision plot of trackers on *OCC* sequences. (c) Success plot of trackers on OOTB. (d) Success plot of trackers on *OCC* sequences.

2.5. Template Updater

Our template updater is robust against occlusion because it applies a template updating strategy according to the information from occlusion detector. The updating strategy is as follows:

1. If the number of occluding background candidates is greater than a predefined threshold, the occlusion happens and the target template stops updating.

2. Otherwise, the occlusion is not observed and the target template will update in usual interpolation way.

By detecting the occurrence of occlusion, the erroneous target template updating can be effectively avoided. It is worth noting that the information of occlusion detector may be further used for more accurate occlusion detection, such as the relative position between target and occlusion.

3. EXPERIMENTAL RESULTS

To evaluate our proposed occlusion detection algorithm, we have implemented a prototype and conducted both quantitative and qualitative experiments on large scale benchmarks. In our implementation, the number of background patches is

Fig. 4. Qualitative results of our tracker in sequences with partial occlusion. The blue bounding boxes denote ground truth. The yellow bounding boxes denote tracking results. The red squares show the background patches that occlude the target. From the results it can be seen that our algorithm can not only accurately detect out the occurrence of occlusion, but also point out where the occlusion happens.

set as 50. If more than 6 of them overlaps the target, the template of target stops updating. Other parameters are the same as KCF[2] and DSST[3]. All the test sequences in our evaluation come from Online Object Tracking Benchmark (OOTB) [11], which contains 51 different tracking tasks with fully annotated attributes, among which we are most interested in is partial occlusion.

3.1. Quantitative Evaluation

Precision plot and success plot are used to present tracking results. We choose KCF [2], Struck [12], SCM [13], TLD [14], CSK [15], IVT [16] and MIL [17] for comparison.

Fig. 3 shows the results of one-pass evaluation (OPE), both for all sequences and the sequences with attribute partial occlusion. From the comparisons we can see that when it comes to sequences with partial occlusion, our tracker shows far more amazing performance with respect to other trackers. The large margin is especially evident in success plot, which is preferred to evaluate a trackers performance than precision plot. Whats interesting is that although almost all trackers have better performance in usual sequences than partial occlusion sequences, our tracker achieves opposite results.

3.2. Qualitative Evaluation

We choose some sequences with partial occlusion to show what our tracker will do when facing with occlusion in Fig. 4. With each sequence, several frames will be presented since the occlusion is a dynamic process. In order to show the details, we just cut the target and the surrounding background out and ignore the other background with no relevance with tracking.

The first row is sequence *Tiger1*, from frame #342 to frame #349. The tiger moves and is covered by the leaves. Thanks to the occlusion detection scheme, in these frame, our tracker is aware of the occlusion and prevents itself from pollution by the background. The second row is sequence *Coke*, which is undergoing similar situation, where the coke can is occluded by the plant. The following two rows show more examples. In sequences *Basketball*, player in green is occluded by players in white; in *CarScale*, the car are occluded by trees; in *FaceOCC1* , the woman is occluded by books; in *Football*, the player is occluded by other players. All the examples prove that our method takes effect.

4. CONCLUSIONS

In this paper, we propose a new definition of occlusion which is useful for occlusion detection. In our method, we track the background patches surrounding the target and examine their relationship with the bounding box of target for occlusion detection. Extensive experiments demonstrate the effectiveness of our method.

However, our method has some disadvantages, such as low efficiency due to the amount of background patches or the fast motion of background. More future work is required.

5. REFERENCES

[1] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2544–2550.

[2] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[3] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097.

[4] Yang Li and Jianke Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conference on Computer Vision*. Springer, 2014, pp. 254–265.

[5] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758.

[6] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[7] Tianzhu Zhang, Kui Jia, Changsheng Xu, Yi Ma, and Narendra Ahuja, "Partial occlusion handling for visual tracking via robust part matching," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1258–1265.

[8] Ting Liu, Gang Wang, and Qingxiong Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912.

[9] Yang Li, Jianke Zhu, and Steven CH Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 353–361.

[10] Juan-Manuel Perez-Rua, Tomas Crivelli, Patrick Bouthemy, and Patrick Perez, "Determining occlusions from space and time image reconstructions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418.

[12] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 263–270.

[13] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1838–1845.

[14] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[15] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[16] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[17] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.