Improvement on Tracking based on Motion Model and Model Updater

Tong Liu[1], Chao Xu[1] *, Zhaopeng Meng[1], Wanli Xue[2], and Chao Li[3]
{toddlt,xuchao,mengzp,xuewanli}@tju.edu.cn, superlee@mail.tjnu.edu.cn

[1] School of Computer Software, Tianjin University
[2] School of Computer Science and Technology, Tianjin University
[3] College of Computer and Information Engineering, Tianjin Normal University

Abstract. Motion model and model updater are two important components for online visual tracking. On the one hand, an effective motion model needs to strike the right balance between target processing, to account for the target appearance and scene analysis, to describe stable background information. Most conventional trackers focus on one aspect out of the two and hence are not able to achieve the correct balance. On the other hand, the admirable model update needs to consider both the tracking speed and the model drift. Most tracking models are updated on every frame or fixed frames, so it cannot achieve the best state. In this paper, we approach the motion model problem by collaboratively using salient region detection and image segmentation. Particularly, the two methods are for different purposes. In the absence of prior knowledge, the former considers image attributes like color, gradient, edges and boundaries then forms a robust object; the latter aggregates individual pixels into meaningful atomic regions by using the prior knowledge of target and background in the video sequence. Taking advantage of their complementary roles, we construct a more reasonable confidence map. For model update problems, we dynamically update the model by analyzing scene with image similarity, which not only reduces the update frequency of the model but also suppresses the model drift. Finally, we integrate the two components into the pipeline of traditional tracker CT, and experiments demonstrate the effectiveness and robustness of the proposed components.

Keywords: Visual tracking, Motion model, Model updater, Image segmentation, Saliency detection, Image similarity.

1 Introduction

Visual object tracking is part of the fundamental problems in computer vision [10, 17, 24, 25]. It is a task of estimating the trajectory of a target in the video

[*] Chao Xu is the corresponding author, School of Computer Software, Tianjin University, Tianjin, 300354, China.

sequence [22]. The tracker has no prior knowledge of the object to be tracked such as category and shape. Despite extensive research on visual tracking, it remains challenging problems in handling complex object appearance changes caused by illumination, pose, occlusion [5] and motion [14]. According to the research of Wang [18], the tracker is composed of several modules: motion model, feature extractor, observation model, model updater, and ensemble post-processor. In particular, motion model and model updater contain many details that can affect the tracking result, but they are rarely concerned. Therefore, this paper will focus on these two components. More in detail, we try to combine salient region detection and image segmentation in the motion model and use image similarity in model updater.

Our approach is based on two major observations of previous work. First, as we all know, the motion model generates object proposals, it samples from the raw input image to forecast the possible candidate locations so as to confirm the scope of target searching. An effective sample selection mechanism can provide high-quality training samples which make the tracker recovers from failure and estimates appearance changes accurately. Hence, it is important to get more accurate samples in motion model. We develop a collaborative method based on image segmentation and salient region detection to analyze the appearance template consisting of the target object and its surroundings. This method differs significantly from existing motion model, such as the sliding window, which is prone to drifting in fast motion or large deformation video. Specifically, we use simple linear iterative clustering algorithm (SLIC [2]) for image segmentation in Section 3.1 and exploit frequency-tuned saliency analysis algorithm (FT [1]) for salient region detection in Section 3.2.

Second, it is critical to enhance the model updater of a tracker that adopts tracking-by-detection approach. Most of the tracking methods update the observation model in each frame, it reduces efficiency and more critical is that poor tracking results can cause the classifier to be contaminated, thus causing drift. Different from some other tracking paradigms that update model in a fixed manner such as updated every two frames, we formulates a simple and quick method to update observation model dynamically with image similarity. It not only improves the tracking speed, but also increases the accuracy. In Section 3.3, we will introduce a perceptual hash algorithm (pHash) in detail for image similarity.

The overview of our approach is illustrated in Fig. 1. The original image is subjected to image segmentation processing and salient region detection respectively, followed by cooperative learning, the result will be sent to CT [23] framework. Our main contribution of this work is to address the problems of tracking-by-detection trackers by effectively ameliorating the motion model and model updater. In order to make more rational use of the visual properties of the image, image segmentation is used to obtain more meaningful atomic regions in the field of color; Salient region detection is used to describe human's visual attention mechanism which involves distance, color, intensity, and texture. We use both methods to handle tracking scenes and targets in motion model thus achieving a more balanced appearance for visual tracking. In addition, we pro-

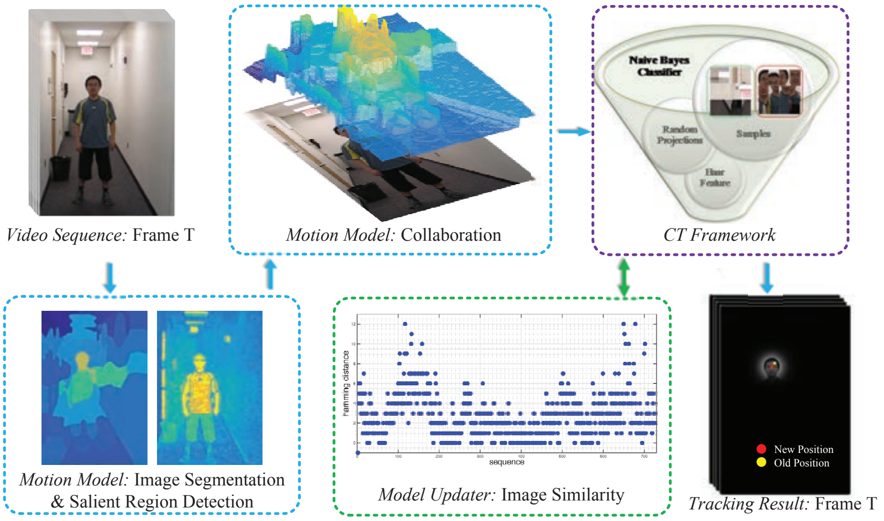

Fig. 1. Tracking Pipeline

pose a novel method to determine whether the estimated target is reliable in the time dimension and make a decision whether to update the observation model by using the hash of image similarity, it is simple and does reduce drift. We evaluate the proposed tracking algorithm on a large-scale benchmark with 50 challenging image sequences [20]. Experimental results show that our algorithm not only has good performance but also makes a significant improvement over the baseline tracker CT [23].

2 Related Work

Most trackers use statistical learning techniques to take charge of constructing robust object descriptors [12,15] and building effective mathematical models for target identification [7,21]. As estimated object position is converted into labeled samples, it is hard to give the accurate estimation of the object position. Hare [6] integrates the labeling positive and negative samples procedure into the learner by using online kernelized structured output support vector machine (Struck). And there are also many tracking algorithms [27] that focus on appearance and motion model definition to deal with the complex scene and avoid drifting. Compressed sensing theory is introduced into visual object tracking by Zhang [23] and he proposes compressive tracking algorithm. CT extracts Haar-like features in the compressed domain as the input characteristics to the classifier. It aims to design an effective appearance model and first compresses sample images of the foreground target and the background using the same sparse measurement matrix to efficiently extract the low-dimensional object descriptors.

In general, tracked results are chosen as the positive samples to update the classifier, noisy samples may often be included since they are not correct enough, which causes the failure of the updating of the classifier. After that, the tracker will drift away from the target. Therefore, sample selection is an important and necessary task for alleviating drift in the motion model. Additionally, massive amounts of training samples would hinder the online updating of the classifier without an appropriate sample selection strategy. Liu [11] designs a sparsity-constrained sample selection strategy to choose some representative support samples from a large number of training samples on the updating stage. It is necessary to integrate the samples contribution into the optimization procedure when observing the appearance of the target.

Most discriminative trackers [13] apply continuous learning strategy, where the observation model is updated rigorously in every frame. Research results show that excessive update strategy will lead to both lower frame-rates and degradation of robustness because of over-fitting in the recent frames. So we refine the strategy of model updater by analyzing the stability of scene.

3 Our Approach

To select high-quality samples, we construct a target-background confidence map according to the similarity of superpixels [19] in the surrounding region of the target between new frame and the first frame. Then it is refined by salient region detection result, the confidence map can facilitate tracker to distinguish the target and the background accurately. Finally, to accelerate the tracker, we control the model updater by judging the stability of scene, computing image similarity between frames.

3.1 Motion Model with Image Segmentation

Image segmentation process clusters pixels by the similarity of their feature and divides the raw image into several specific regions that may correspond to the tracked object. Superpixel is a kind of image segmentation algorithm, which provides a convenient primitive to compute local image features. There is a popular superpixel algorithm named SLIC (Simple Linear Iterative Clustering). SLIC [2] is fast, easy to use, and produces high-quality segmentations.

We segment the first frame into N superpixels. A color histogram is extracted as the feature vector f_i for each superpixel $sp(i)(i = 1, ..., N)$. We choose mean shift to cluster these superpixels. We can obtain n_{sp} ($n_{sp} < N$) different clusters and each cluster center $f_c(j)(j = 1, ..., n)$ by employing mean shift algorithm on the feature pool $F = \{f_i | i = 1, ..., N\}$.

Each cluster $clst(j)(j = 1, ..., n_{sp})$ is a set, and each cluster has its own members $\{f_i | f_i \in clst(j)\}$. Therefore, each cluster is represented by $f_c(j)$ and $clst(j)$ in the feature space. The members of each cluster $clst(j)$ corresponds to different superpixels in the image region. The weight $w(j)(j = 1, ..., n_{sp})$ is assigned to each cluster center $f_c(j)$ by exploiting a prior knowledge of the targets bounding box in the first frame, which indicates the likelihood that superpixel members of $clst(j)$ belong to the target area. We count two scores for each cluster $clst(j)$: $s^+(j)$ and $s^-(j)$. The former denotes the size of the overlapping area that all superpixel members of each cluster $clst(j)$ cover the bounding box, and accordingly the latter denotes the size of all superpixel members outside the target area. The weight is normalized between -1 and 1 and calculated as follows:

$$w(j) = \frac{s^+(j) - s^-(j)}{s^+(j) + s^-(j)}, \forall j = 1, ..., n_{sp} \tag{1}$$

where larger positive values indicate high confidence to assign the cluster center $f_c(j)$ to target and vice versa. To obtain a confidence map for the t-th frame, we first segment a surrounding region of the target into n_{sp} superpixels and then compute every superpixel confidence value. The surrounding region of the target is a square area, and its side length is $\eta\sqrt{S}$, where η is the constant parameter to control the size of this surrounding area and S is the area size of the target. The confidence value of a superpixel depends on two factors: the distance between this superpixel and the cluster center in the feature space, and the weight of the corresponding cluster center. $C_t(i)$ is the confidence value for superpixel i at the t-th frame. The confidence value of each superpixel is computed as:

$$C_t(i) = argmax_{1 \leq j \leq n}\{e^{||f_t(i) - f_c(j)||} \times w(j)\} \; \forall i = 1, ..., n_{sp} \tag{2}$$

where $f_t(i)$ and $f_c(j)$ denote the feature vector of the i-th superpixel in the t-th frame and the j-th cluster center in the first frame, respectively. Intuitively, the nearer the feature of a superpixel $f_t(i)$ is close to the targets cluster center $f_c(j)$, the more likely this superpixel belongs to the target area.

Each pixel in the i-th superpixel in the t-th frame shares the same confidence value $C_t(i)$. The surrounding area of the target is scanned with a sliding window that has the same size as the bounding box. At each position, the sum of the confidence value in this sliding window is computed, which demonstrates evidence for separating the target from the background. Then, the location of sliding window with the maximum value response will be selected as the new candidate location.

3.2 Motion Model with Salient Region Detection

Saliency is intentionally regarded as visual attention, it is determined as the local contrast of an image region with respect to its neighborhood at various scales, using one or more features of intensity, color, and orientation. The study of saliency detection comes from biological research. It is utilized to interpret complex scenes now. Scene analysis technique is integrated into visual tracking pipeline will significantly improve the performance, because it can separate the target from the background using high-quality saliency maps.

We use frequency-tuned saliency analysis algorithm (FT) to obtain the saliency map. This method can emphasize the largest salient objects and uniformly highlight whole salient regions. In order to have well-defined boundaries, the FT algorithm retains high frequencies from the original image. The frequency-tuned saliency analysis is formulated as

$$S_t(x,y) = ||I_\mu - I_{\omega hc}(x,y)|| \tag{3}$$

I_μ is the mean image feature vector of color and luminance, $I_{\omega hc}(x,y)$ is the corresponding image pixel vector value in the Gaussian blurred version (using a 5×5 separable binomial kernel) of the original image, and $|| \cdot ||$ is the L2 norm. Here we use the Lab color space, each pixel location is an $[L,a,b]^T$ vector, and the L2 norm is the Euclidean distance.

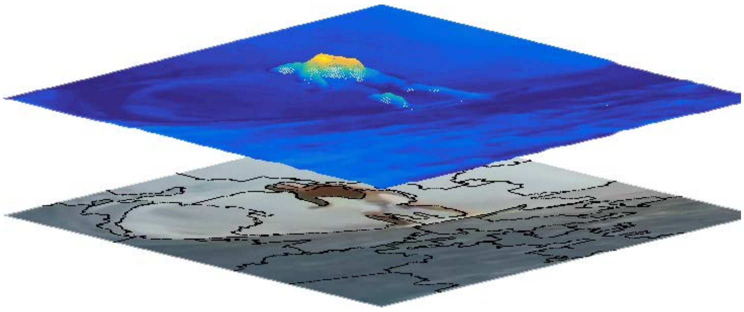

Fig. 2. saliency map

As we can see, the superpixels result is not stable. It only provides a coarse over-segmented image. To get the likelihood that superpixel members whether belong to the target area, we still need the prior knowledge of the targets bounding box in the first frame. The figure shows that salient region detection provides the probability of each pixel belonging to the foreground target, the result can be used to refine confidence map, it is formulated as:

$$Cmap_t = \phi C_t(i) + (1-\phi)S_t \tag{4}$$

Here, each pixel of the i-th superpixel in the t-th frame shares the same confidence value $C_t(i)$, and each pixel has a sailency value in map S_t. We normalize the superpixel confidence value and FT saliency value to $[0,1]$, and then fuse them to get the final confidence map. The parameter ϕ controls fusion of these two confidence value.

3.3 Model Updater with Image Similarity

We have integrated superpixels segmentation and salient region detection into CT, these procedure improves the performance of the base model. However, there is a computational overhead, it will slow down the base model. So we refine the strategy of model update to accelerate our tracker, the classifier will only be updated when the scene is not stable (background significantly changes).

We analysis the stability of scene by comparing similarity of incoming frame with previous frames. Here we use a perceptual hash algorithm pHash [16] to get the fingerprint of image, which has several properties: images can be scaled larger or smaller, have different aspect ratios, and even minor coloring differences (contrast, brightness, etc.) and they will still match similar images. The fingerprint result will not vary as long as the overall structure of the image remains the same. This can survive color histogram adjustments.

Algorithm 1 pHash Algorithm

1. Reduce size: Resize the input image to 32x32, pHash starts with a small image to simplify the DCT computation.
2. Reduce color: Reduce the image to grayscale to further simplify computation.
3. Compute the DCT: Separate the image into a collection of frequencies and scalars.
4. Reduce the DCT: Output of DCT is 32x32, just keep the top-left 8x8, which represents the lowest frequencies.
5. Further reduce the DCT: Compute the mean DCT value, and set the 64 hash bits to 0 or 1 depending on whether DCT values is above or below the average value.
6. Construct the hash: Set the 64 bits into a vector, it is the hash of image, we can compare the difference of images by computing hamming distance of their hash vector.

3.4 Tracking Framework

Compressive tracking (CT) aims to design an effective appearance model, which first compresses sample images of the foreground target and the background using the same sparse measurement matrix to efficiently extract the low-dimensional object descriptors [8], and then apply naive Bayes classifier with online update to classify the extracted features for object identification.

CT extracts haar-like features in the compressed domain as the input characteristics into classifier. The sparse measurement matrix facilitates efficient projection from the image feature space to a low-dimensional compressed subspace. The random measurement matrix is not a traditional Gaussian matrix, because it is dense with expensive memory space and computation. Finally, the maximal response classified by the classifier represents the tracking location in the current frame.

In order to locate the object of interest in the current frame, we introduce a confidence map based on scene analysis in the process of motion estimation and extend the coarse-to-fine sliding window search strategy. Firstly, the coarse-grained sliding window search is performed based on the previous object location within a large search radius γ_c to find the coarse location $\mathbf{l}_t$. Secondly, the sliding window obtains the tracking location $\hat{\mathbf{l}}_t$ with the maximal value response by shifting the window in the confidence map. Above two procedures not only consider the different shape or texture between the target and the background, but also take full advantage of discriminative color descriptors as a guidance. After that, the detected object location is close to the accurate object location. Thirdly, the fine-grained sliding window is carried out within a small search radius γ_f and then we can obtain the final object location $\mathbf{l}_t$ in the the t-th frame. This strategy is more effective than previous work because of constructing the confidence map from color cues.

4 Experiment

In this section, we validate our tracking algorithm on OTB datasets, and compare our method with several trackers to demonstrate the effectiveness and robustness of our method. Our tracker is implemented in MATLAB on a 2.6 GHz Intel Core i5 CPU with 8GB memory.

4.1 Qualitative Comparison

Comparison to the Baseline Tracker To evaluate the impact of scene analysis, we compare our model with the standard CT on OTB [20], and run the One-Pass Evaluation (OPE) to verify the performance. As we can see, our method significantly ourperforms the baseline tracker CT in success rate and precision, the score has increased by about 50 percent. Therefore, the experiment results clearly demonstrate the importance and necessity of scene analysis. It helps the base tracker to handle target deformation and illumination change. The robustness of our method validates the important role of scene analysis component in visual tracking pipeline.

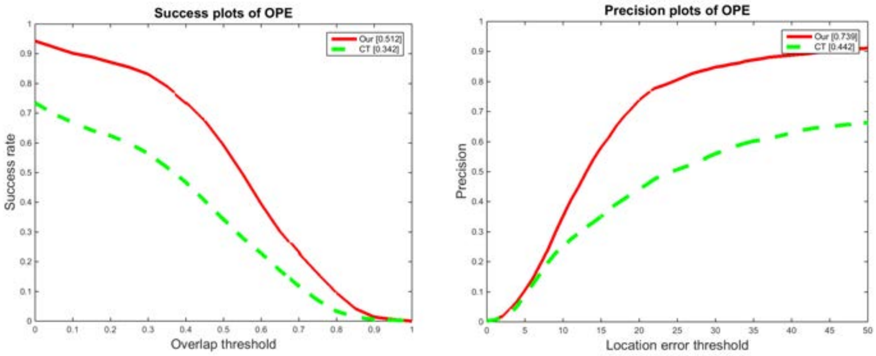

Fig. 3. OPE plot on OTB dataset, comparing to baseline CT

Comparison to the Basic Trackers We compare our tracker with 7 tracking algorithms on dataset OTB over all 50 videos, these trackers are proposed almost the same period with CT. They are: CSK [4], SCM [26], Struck [6], ASLA [8], TLD [9], MIL [3] and CT. The results in Fig. 4 show that our method achieves almost the best performance using both the metrics. Our method is robust when the target deformation, illumination variatation and background clutter, as we can see, our method achieves higher score in the benchmark than other methods.

4.2 Quantitative Comparison

The robustness of our method is pretty obvious when there are target deformation, illumination variatation and background clutter. Fig. 5 shows that our method has the advantages of dealing with complex scene, such as the squence Matrix and Cola. Most trackers are confused by background clutter because they don’t have an efficient motion model to identify high-quality samples.

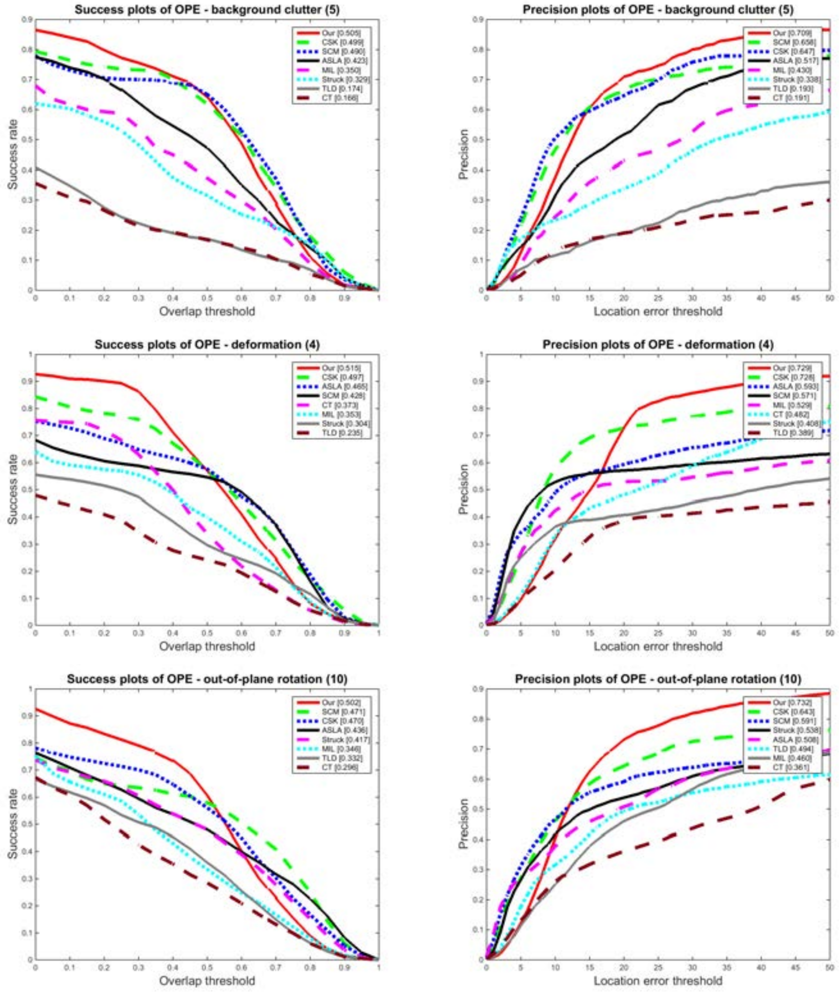

Fig. 4. OPE result with deformation, out-of-plane rotation and background clutter

5 Conclusion

In this paper, we propose an effective algorithm for conventional visual tracking in motion model and model updater. Our method is more comprehensively considers the visual spatial attention factors in the appearance template, such as color, distance, intensity, and texture. Through the cooperation between salient region detection and image segmentation, we get an effective motion model which

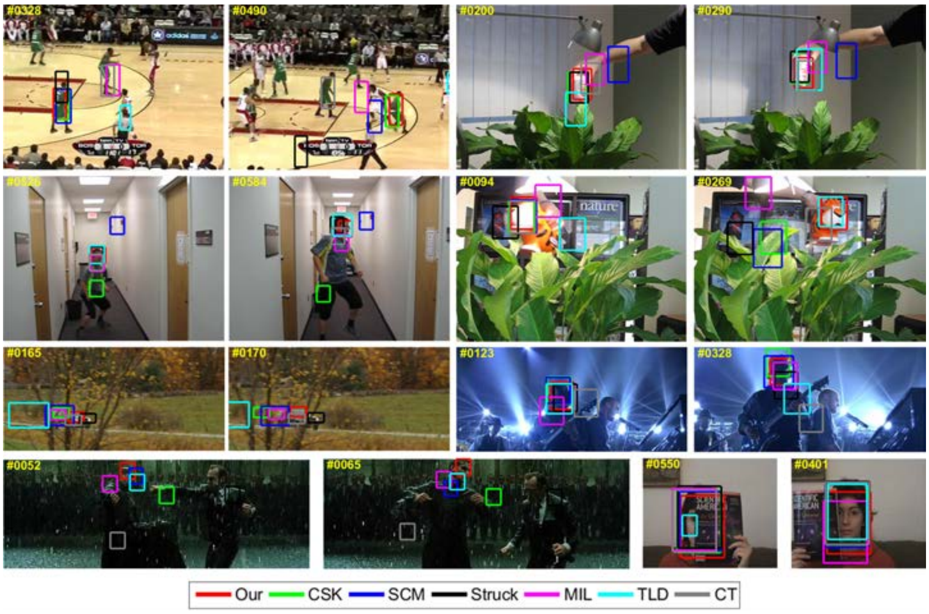

Fig. 5. Tracking snapshot in several sequence

has the right balance between target processing and scene analysis. We further develop an effective online model updater using fast image similarity to measure the rationality of the estimated target in the time dimension, and it will reduce the frequency of the model update and improve the tracking accuracy. Extensive experimental results show that the proposed algorithm performs favorably against the baseline trackers and basic tracker in terms of efficiency, accuracy, and robustness.

6 Acknowledgments

This work is supported by National Key Technology R&D Program (No. 2015BAH52F00) and National Natural Science Foundation of China (No. 61304262).

References

1. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. pp. 1597–1604. IEEE (Jun 2009)
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence 34(11), 2274–2282 (2012)
3. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 983–990. IEEE

4. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1090–1097 (2014)
5. Dong, X., Shen, J., Yu, D., Wang, W., Liu, J., Huang, H.: Occlusion-aware real-time object tracking. IEEE Transactions on Multimedia 19(4), 763–771 (2017)
6. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: 2011 International Conference on Computer Vision. pp. 263–270. IEEE
7. Hong, X., Chang, H., Shan, S., Zhong, B., Chen, X., Gao, W.: Sigma set based implicit online learning for object tracking. IEEE Signal Processing Letters 17(9), 807–810 (2010)
8. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. pp. 1822–1829. IEEE (2012)
9. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelligence 34(7), 1409–1422 (2012)
10. Lin, L., Lu, Y., Li, C., Cheng, H., Zuo, W.: Detection-free multiobject tracking by reconfigurable inference with bundle representations. IEEE transactions on cybernetics 46(11), 2447–2458 (2016)
11. Liu, Q., Ma, X., Ou, W., Zhou, Q.: Visual object tracking with online sample selection via lasso regularization. Signal, Image and Video Processing pp. 1–8 (2017)
12. Ma, B., Hu, H., Shen, J., Liu, Y., Shao, L.: Generalized pooling for robust object tracking. IEEE Transactions on Image Processing 25(9), 4199–4208 (2016)
13. Ma, B., Huang, L., Shen, J., Shao, L.: Discriminative tracking using tensor pooling. IEEE transactions on cybernetics 46(11), 2411–2422 (2016)
14. Ma, B., Huang, L., Shen, J., Shao, L., Yang, M.H., Porikli, F.: Visual tracking under motion blur. IEEE Transactions on Image Processing 25(12), 5867–5876 (2016)
15. Ma, B., Shen, J., Liu, Y., Hu, H., Shao, L., Li, X.: Visual tracking using strong classifier and structural local sparse descriptors. IEEE Transactions on Multimedia 17(10), 1818–1828 (2015)
16. Mıhçak, M.K., Venkatesan, R.: New iterative geometric methods for robust perceptual image hashing. In: ACM Workshop on Digital Rights Management. pp. 13–21. Springer (2001)
17. Tian, C., Gao, X., Wei, W., Zheng, H.: Visual tracking based on the adaptive color attention tuned sparse generative object model. IEEE Transactions on Image Processing 24(12), 5236–5248 (2015)
18. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3101–3109 (2015)
19. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 1323–1330. IEEE (2011)
20. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2411–2418 (2013)
21. Xu, C., Tao, W., Meng, Z., Feng, Z.: Robust visual tracking via online multiple instance learning with Fisher information. Pattern Recognition 48(12), 3917–3926 (2015)
22. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing 74(18), 3823–3831 (2011)
23. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: European Conference on Computer Vision. pp. 864–877. Springer (2012)

24. Zhao, L., Gao, X., Tao, D., Li, X.: Learning a tracking and estimation integrated graphical model for human pose tracking. IEEE transactions on neural networks and learning systems 26(12), 3176–3186 (2015)
25. Zhao, L., Gao, X., Tao, D., Li, X.: Tracking human pose using max-margin markov models. IEEE Transactions on Image Processing 24(12), 5274–5287 (2015)
26. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. pp. 1838–1845. IEEE (2012)
27. Zuo, W., Wu, X., Lin, L., Zhang, L., Yang, M.H.: Learning support correlation filters for visual tracking. arXiv preprint arXiv:1601.06032 (2016)