Long-term tracking with fast scale estimation and efficient re-detection

Zhang Zengshuo 1,2, Tang Linbo 1,2, Han Yuqi 1,2, Nan Jinghong 1,2, Zhao Baojun 1,2

1 School of Information and Electronic, Beijing Institute of Technology, People’s Republic of China
2 Beijing Key laboratory of Embedded Real-time Information Processing Technology, Beijing Institute of Technology, Beijing 100081, People’s Republic of China

Abstract: In long-term tracking applications, occlusion and scale variation are common attributes which cause performance degradation. Existing solutions use heavy calculation to deal with these problems, without considering the real-time implementation. Therefore, the authors propose a novel long-term tracker with fast scale estimation and efficient re-detection scheme to maintain real-time speed and favourable accuracy. Specifically, the authors integrate a distance metric method into correlation filter-based tracker to realise fast translation calculation and scale estimation. In addition, the authors advocate a keypoint-matching based confidence indicator to verify the tracking result and activate the re-detection module when the occlusion happens. The authors test our approach on challenging sequences with scale variation and occlusion. Experiments demonstrate that our proposed tracker procures preferable effect than state-of-the-art methods in the aspect of both speed and accuracy.

1 Introduction

Visual object tracking has numerous applications in the field of computer vision, such as video surveillance and robotics. The purpose of long-term tracking is to detect the target in the subsequent frames as long as possible, after given the position of a target in the first frame [1, 2]. Despite a large quantity of tracking methods have been proposed in this years, most of these trackers suffer a degradation in performance due to illumination changes, and so on. In this paper, we mainly tackle two attributes in long-term visual tracking, i.e. occlusion and scale variation.

Occlusion is an enormous challenge for visual tracking, as it would temporary hide the holistic appearance of the target. To resolve this dilemma, Kalal et al. decompose a tracker into three modules, namely, tracking, learning and detection (TLD) [3]. In this framework the tracking and the detection facilitate each other, the results from the tracker provide training data to update the online random forest detector, and this detector will re-initialises to find target when the tracker fails. The long-term correlation tracker (LCT) [4] augments a standard correlation filter tracker with an confidence estimation using the additional correlation filter and an online support vector machine (SVM) to re-detected the target. However, these methods rely on online training slide-window detector with weak features (e.g. grey, Haar-like), which have complexity calculation yet hardly satisfy the robust and real-time requirement for visual object tracking, especially under the circumstance of limited computing resources.

On the other hand, accurate scale estimation is also a hard difficulty in the visual object tracking. Among these popular correlation filter-based trackers [1–6], existing scale estimation solutions can be classified into two categories. One way is reproducing the same translation calculation in different image scale but require to bear the cost of massive repeating low-value calculation (e.g. scale adaptive multiple feature tracking (SAMF) [5]). The other solution is training a scale correlation filter to calculate the most similar scale using the multi-scale pyramid, while this method demand exhaustive scale layers for training (e.g. discriminative scale space tracking (DSST) [6]). Overall, such existing scale-aware approaches could address such issues but undertake a heavy inefficiency computational burden.

In this article, we propose a novel robust long-term tracker based upon a stabilised and fast short-term tracker, combining an efficient re-detection scheme to address such limitations mentioned above. Besides the correlation filter-based translation estimating in short-term tracking, we design a distance metric method using Histogram of Oriented Gradients (HOG) features to estimate the scale variation with less computational magnitude. In addition, we advocate a target confidence indicator to monitor the result of tracking and activate the re-detection module when the occlusion happens. This module detect the target based on star model by means of keypoint matching. Since target keypoints contain the relative position and orientation of the target, we show that these keypoints possess the power of re-detecting the target effectively after long-term occlusion. Experimental results indicated that the proposed method outperforms the most compared methods.

2 Proposed method

For the objective of long-term tracking, we equip the online short-term tracker with a robust re-detection scheme in case of occlusion and out-of-view. We decompose the short-term tracking into the two-dimensional translation calculation and the one-dimensional scale estimation, where the translation is estimated using the correlation filter as baseline and the scale estimation is carried out using distance metric in the multi-scale pyramid. Moreover, we propose an effective target confidence indicator to monitor the result of tracker and re-detect the target by keypoint matching method when the tracking failure happens (Fig. 1).

2.1 Correlation tracking

Correlation Filter-based trackers take advantage of the densely sampled examples and high-dimensional handcraft feature in high-speed utilising the Fourier domain transformation, which recently have achieved the better performance. Accordingly, we use the correlation-based tracker to process the frame without occlusion, providing a stable and fast base tracker. Let $f$ be a $d$-dimensional rectangular feature map, which calculated from an image patch centred around the target. We denote dimension of feature index $l \in \{1, \ldots, d\}$ of $f$ by $f_l$. The target is to training an expected
correlation filter $h$, which containing every filter $h_i$ in each feature dimension. The cost function will be minimised as:

$$\varepsilon = \| \sum_{i=1}^{d} h_i \odot f_i - g \| + \lambda \| \sum_{i=1}^{d} h_i \|^2.$$  \hspace{1cm} (1)

Here, $g$ is represented as the desired Gaussian output. The parameter $\lambda \geq 0$ stand for the ability of regularisation. The fast Fourier transformation (FFT) is applied to minimise the cost function, and the solution is formulated as:

$$H_i^l = \frac{G_i^l}{\sum_{i=1}^{d} F_i^l + \lambda}.$$  \hspace{1cm} (2)

This above capital letters marks the FFT of the corresponding functions, and the overline represents the complex conjugation operator.

To achieve the robust approximation, we update parameters of the correlation filter with linear increase method. Given an image patch $z$ in the next frame, the correlation scores $y$ will be computed as $y = F^{-1}(\sum H^l Z^l)$, where $F^{-1}$ represent the inverse FFT calculation. The target location in this new frame can be detected at the most greatest correlation score of $y$. For more details, we refer readers to DSST [6].

2.2 Fast scale estimation

During the visual tracking application, the scale difference between the two adjacent frames is usually smaller which compared with translation. Therefore, we calculate the scale variation at the target location which estimated by the translation estimation. Let $R \times C$ represent the size of target in the current frame and $N$ stand for the scales number. For each $s \in S = \{a^d \cdot [(N - 1)/2], \ldots, [(N - 1)/2]\}$, $s$ denotes the scale factor between feature layers. We collect every image patch $J_i$ according to the size $sR \times sC$ around the target, and then uniformly resize this patches into the size $R \times C$ again and use HOG features to establish the scale feature pyramid. Let $H_i^l$ denote HOG features of the target in the training frame and $H_i$ indicate this features of sample at the $i$th scale layer. The HOG Distance between the target and samples is defined as follows:

$$D(H_i, H_m) = \text{Tr}((H_i - H_m)^T(H_i - H_m)).$$  \hspace{1cm} (3)

where $\text{Tr}(\cdot)$ represents the trace of a matrix. By minimising this HOG distance, the target optimum scale will be estimated by minimised the function as:

$$\hat{s} = \arg \min_i \{D(H_i, H_m)\}.$$  \hspace{1cm} (4)

After scale estimation, the scale model $H_m$ is updated using the new target sample by means of linear increase. Up to this point, translation and scale have been estimated while being much more computationally efficient and effective.

2.3 Target confidence indicator

When tracking a target, the short-term tracker may cause a deterministic failure once the target is severely occluded or missing from the current frame entirely. To maintain the tracker stability, we use a confidence indicator verifying the result of tracker. When the target is occluded, the confidence coefficient will decline rapidly. Therefore, target confidence indicator will activate the target re-detection when tracking failure happens and determines whether or not adopt the re-detected results. When the confidence coefficient divided by its historical average lower than the predefined threshold $T_\theta$, then the re-detection module will activate to locate the position of target. During the re-detection state, until this confidence ratio greater than the above threshold again, then the detector will re-initialises the tracker.

We designed the confidence indicator through matching the keypoints between the result of tracker in current frame and history target keypoints. Using fast keypoint detectors and binary descriptors to make our implementation have the capacity to run in real-time, first, we detect Oriented FAST and Rotated BRIEF (ORB) keypoints and its corresponding binary descriptors in the image of target area

$$K_p = \{ (p, \theta, m) \}^N.$$  \hspace{1cm} (5)

Here, $K_p$ denotes the pool of target keypoints, and these keypoints will be updated in the queue form during the tracking state. We initialise $K_p$ by detecting and describing keypoints in the first frame, where each keypoint denotes a location $p$ in absolute image coordinates, an orientation $\theta$ and a binary descriptor $m$. For every binary descriptor, we will calculate the Hamming distance as a metric of keypoint matching. After the translation and scale estimation in the new arriving frame, we detect the keypoints $K_i$ in the area of target, and match these to the keypoint in $K_p$ by requiring that the minimum distance lower than threshold $T_m$. The confidence indicator refers to the number of matching keypoints (Fig. 2)

$$C_i = \text{NUM}(\text{dis}(K_p, K_i) < \lambda_\theta).$$  \hspace{1cm} (6)

Owing to matching keypoints are represented for the similarity between current and historical target images. Therefore the number of matching keypoints is designed as the confidence indicator for estimating the stability of tracker. The confidence indicator change smoothly when appearance changes slowly, but will significantly decrease once the target is occluded or missing. Overall, the confidence indicator can be used to discriminate the occlusion and make a decision whether to adopt the re-detection result.
In our experiment, the regularisation parameter is set to $\lambda = 0.5$.

### 3.1 Experimental parameter setup

In our experiment, the regularisation parameter is set to $\lambda = 0.5$ in (1). The filter size is set to twice the initial target size. In the scale estimation section, we set the HOG cell size is $4 \times 4$, and the bin number of orientation is 9. We only use $N = 3$ scales number and the scale factor is fixed as 1.02. The learning rate is set to $\eta = 0.025$ for our method. For handling occlusion, the determine threshold is set to $T_a = 0.5$, and the keypoint Hamming matching threshold is $T_m = 75$. For fair comparison, we fixed the same parameter values during testing all the sequences, and other trackers results are generated by the source code provided by the authors on their websites.

### 2.4 Effective re-detection

In order to re-detect target fast and effectively, we utilise the local keypoint collected in $K_p$ in times of tracking, and locate the target based on Star Model during re-detection period. After the activation by confidence indicator, we will match keypoints in $K_p$ to keypoints extracted in the new frame. Making full use of the angle difference between the orientation of keypoint and the direction of target's centre position, the proposed method will vote and cluster the target centre point by matching keypoints to achieve the re-detection.

When tracking the target, we constantly collect local keypoints extracted from the target region. Let $\theta_i$ is the orientation of a target keypoint. $\Delta x$ and $\Delta y$ denote coordinate difference between keypoint and target centre position in horizontal and vertical directions, then the angle difference $\beta_d$ can be calculated as follows:

$$\beta_d = \text{atan2}(\Delta y/\Delta x) - \theta_i,$$  

where $\text{atan2}$ is the quadrant-aware version of arctan. After matching new keypoints $K_n$ to historical keypoints in $K_p$, the matching keypoint will add the angle difference $\beta_d$ on the basis of the keypoint orientation to get the direction of target position. Utilising the relative position and polar coordinates, these matching keypoints will vote the target centre position efficiency. However, due to the error matching and image noise, this voting points still contain outliers in the process of matching keypoints. We utilise the hierarchical clustering method to cluster the target centre in the image domain.

## 3 Experiments

In this section, we empirically validated our proposed tracker on 11 challenging video sequences, which annotated with the scale variation and occlusion attributes in the OTB100 benchmark [8] and UAV dataset [7]. Furthermore, we compare our method with five recent and excellent trackers which can be divided into two categories: (i) excellent short-term tracker DSST [6] and (ii) long-term trackers equipped with re-detection method such as LCT [4], TLD [3] and MUSTER [9]. Our proposed method in this paper is realised in Matlab, and ORB keypoints are extracted by OpenCV using MEX function. We perform the experiments on an Intel Core i5-4570S 2.90 GHz CPU with 8 GB RAM and Samsung 850 EVO SSD.

### 3.1 Experimental parameter setup

In our experiment, the regularisation parameter is set to $\lambda = 0.01$ in (1). The filter size is set to twice the initial target size. In the scale estimation section, we set the HOG cell size is $4 \times 4$, and the bin number of orientation is 9. We only use $N = 3$ scales number and the scale factor is fixed as 1.02. The learning rate is set to $\eta = 0.025$ for our method. For handling occlusion, the determine threshold is set to $T_a = 0.5$, and the keypoint Hamming matching threshold is $T_m = 75$. For fair comparison, we fixed the same parameter values during testing all the sequences, and other trackers results are generated by the source code provided by the authors on their websites.

### 3.2 Quantitative and qualitative results

Table 1 shows the quantitative results and average speed of experiment sequences. To rank these algorithms, we evaluated all trackers by adopting the average overlap rate. This criteria is defined as the average percentage of frames where the bounding box overlap surpasses a threshold from zero to one. The overlap is defined as the intersection divided by the union between ground truth and tracking results, which proposed in [8]. In these experiment sequences, Dog1, Car1_1 and Car5 are annotated with the scale variation, while others all possess the attribute of occlusion. Compared with other tracking algorithms, the proposed method mostly achieve the first or second performance. We can see that MUSTER could yield a competitive performance, but it shoulder a heavy computational burden. It is worthy to mention that our approach is significantly faster than the other trackers, which is expected to be a fully real-time tracking and favourable accuracy. Specifically, our method is ~9 times faster than TLD and LCT, ~3 times faster than DSST.

Fig. 3 illustrates qualitative results compared with other trackers on scale variation and occlusion sequences. In Jogging2 and Car1_2, our approach favourably re-detect the target position while compared trackers fail to handle the occlusion. In the sequence of Car5, our tracker accurately estimates the scale and position better than LCT and STAPLE, indicating the effective of our scale estimation method. In the other sequences, these short-term trackers DSST and STAPLE hardly deal with the occlusion problem. However, our method manages to track the target demonstrating the favourable re-detection performance.

### 4 Conclusion

In this paper, we propose a novel tracker with effective scale estimation and fast re-detection method satisfying the real-time long-term tracking requirement. The translation is estimated by modelling a correlation filter-based tracker, and the scale estimation can be quickly realised through the HOG distance metric. To solve the occlusion problem, we propose an effective indicator to monitor the result of tracker and re-detect the target by matching keypoints when the tracking failure happens. Experiments demonstrate that the proposed method could outperform the state-of-the-art method in terms of both speed and accuracy. It is obvious that our tracker is a logical choice for applications that require computational simplicity.

### 5 Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2017YFC0804700). The work described in this paper is
supported by Chang Jiang Scholars Programme (Grant No. T2012122), 111 Project of Chain under Grant B14010.

6 References

[1] Yilmaz, A., Javed, O., Shah, M.: ‘Object tracking: a survey’, ACM Comput. Surv., 2006, 38, (4), Article 13, pp. 1–45

[2] Chen, Z., Hong, Z., Tao, D.: ‘An experimental survey on correlation filter-based tracking’. Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 1–13

[3] Kalal, Z., Mikolajczyk, K., Matas, J.: ‘Tracking-learning-detection’, IEEE Trans. Pattern Anal. Mach. Intell., 2012, 34, (7), pp. 1409–1422

[4] Ma, C., Yang, X., Zhang, C., et al.: ‘Long-term correlation tracking’. Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 5388–5396

[5] Li, Y., Zhu, J.: ‘A scale adaptive kernel correlation filter tracker with feature integration’. European Conf. on Computer Vision, Zurich, Switzerland, 2014, pp. 254–265

[6] Danelljan, M., Hager, G., Khan, F.S., et al.: ‘Accurate scale estimation for robust visual tracking’. British Machine Vision Conf., 2014

[7] Mueller, M., Smith, N., Ghanem, B.: ‘A benchmark and simulator for uav tracking’. European Conf. on Computer Vision, 2016, pp. 445–461

[8] Wu, Y., Lim, J., Yang, M.: ‘Object tracking benchmark’. IEEE Trans. Pattern Anal. Mach. Intell., 2015, 37, (9), pp. 1834–1848

[9] Hong, Z., Chen, Z., Wang, C., et al.: ‘Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking’. Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 749–758