Multi-Stage Target Tracking with Drift Correction and Position Prediction

Xin Chen, Keyan Ren and Yibin Hou

Beijing University of Technology, Beijing Engineering Research Center for IoT Software and Systems, Beijing, China.

*keyanren@bjut.edu.cn

Abstract. Most existing tracking methods are hard to combine accuracy and performance, and do not consider the shift between clarity and blur that often occurs. In this paper, we propose a multi-stage tracking framework with two particular modules: position prediction and corrective measure. We conduct tracking based on correlation filter with a corrective measure module to increase both performance and accuracy. Specifically, a convolutional network is used for solving the blur problem in realistic scene, training methodology that training dataset with blur images generated by the three blur algorithms. Then, we propose a position prediction module to reduce the computation cost and make tracker more capable of fast motion. Experimental result shows that our tracking method is more robust compared to others and more accurate on the benchmark sequences.

1. Introduction

VOT (Visual object tracking) is an essential topic in computer vision with many conceptually or methodologically various object-tracking algorithms being proposed recently [1,2,3,4]. The state-of-art target tracking methods are divided into two categories: Convolutional Neural Networks (CNN) [4,5,6] and Correlation Filter [1,2,7,8].

The advantage of correlation filtering method is faster than CNN, but the accuracy of correlation filter is lower. At the same time, the method of general correlation filtering needs to mark the object artificially at the first frame, and the bounding box of target is usually prone to drift due to the changes of object appearance changes. The application method of CNN is general detect each frame of the video in the target tracking field, which lead to a significant decrease in speed. Currently, universal trackers do not consider the blurry situation explicitly, which is general on the realistic scene when the targets move fast or photographic equipment shake unexpectedly.

To address the above practical problems, we proposed a multi-stage tracking framework which conduct tracking by correlation filter with a corrective measure by CNN detection at key frame to increase the performance, accuracy and meet the practical requirements. The whole framework is shown in Figure 1. In this work, to solve the blur situations in the realistic scene and the drift problem, we apply deep CNN for detection and correlation filter for tracking. And we train the VGG16 [9] module provided by SSD [10] with dataset to extracting object proposal with classification and confidence score. The dataset combines VOC (Visual Object Classes) with blurry images that blurred by Gaussian blur, motion blur, and defocus blur to increasing accuracy. We obtain the fi, si and bi from detection and track by discriminative scale space tracking [2]. We also design a corrective
The contributions of our research are: (1) An efficient method proposed for target accurate tracking when visual tracking scene contains clear, blur or illumination variation stages. (2) Different from existing methods, the proposed tracking method has a corrective measure module to relocation the target when tracker drifts. (3) The proposed tracker is capable of scale accommodation in different challenging factors.

We assess the proposed tracking framework on OTB-100 (Visual Tracker Benchmark) [5]. Extensive experimental results on the sequences of OTB-100 show that our framework performs good tracking efficiency and is effective on full scene stage, which can restrain the tracking drift. Thus our framework can increase the performance of the general tracker especially in the situations of scale transformation, motion blur and fast motion.

2. Related work

2.1. Convolutional neural network

Since CNN is applied to the computer vision field, highest level methods of object detection are principally based on deep CNN detection. Girshick et al. [11] proposed a scalable detection algorithm called R-CNN (Regions with Convolutional Neural Networks features) by training deep CNN with extracted region proposals which are localized and segmented by Selective Search [12]. The whole object detection system is decomposed into several stages including extracting region proposals, computing CNN features, classifying with class-specific linear SVMs. At that time, this framework showed prominent performance and many methods had based on it [13,14].

As a result of massive repeated calculations in CNN features computation of extracted region, RCNN faces a serious speed bottleneck. To accelerate the training and detecting process, Fast R-CNN [15] is proposed. A SPP-NET [16] network layer was used in CNN. Each region patch has a fixed dimension feature representation and is classified by the normal softmax instead of wrapping to a
fixed size for CNN features computation. And it skillfully put the bounding box regression into the neural network combine with region classified to a multi-task model. In [17], the selective search algorithm for region proposal was replaced by a Region Proposal Network (RPN) and put into the neural network. The framework can be train as a full convolution network end-to-end that achieved training and update synchronous.

The real-time detection method proposed by W. Liu [10] eliminates the process of bounding boxes and pixel or feature resampling in the middle. It ensures both the speed and the detection accuracy even when the resolution of input image is low by predicting object and box offsets at different levels of feature maps. Training of the convolutional neural networks has become simpler. And a better trade-off is obtained between the detection speed and the detection precision. The mAP of the result can be comparable to the method using region proposals (such as Faster R-CNN).

2.2. Correlation filter-based tracking

The research of object tracking mainly has two categories: generative and discriminant model. The method of the former is to find the best matching window, and the latter is to distinguish the target from the surroundings. In discriminative model, correlation filter performs better that generate high response (correlation peak) to each interested object in the scene and conversely to the background. The main process is mark the object artificially at the first frame, predict the position by the target in initial position.

KCF [8], the typical and high-performance Correlation filter-based tracking algorithm. It used multiple channels instead of the original work, cyclic matrix to acquire samples and ridge regression to train target detector, has reduced the computational complexity and improved the computational speed. Yang, Y. et al [6] proposed a multiple object tracking system based on KCF for tracking and background subtraction for targets detecting. Martin et al [1] designed an efficient tracker based on correlation filter named DSST which designs two consistent correlation filters relatively independent to training or testing with different feature types and computing methods. This algorithm can be transplanted to any algorithm for target location and scale estimation. fDSST [2] and ECO [18] were proposed to reduce the computational cost without sacrificing its robustness and accuracy.

3. Multi-stage target tracking with drift corrective and position prediction

We proposed a multi-stage tracking framework which conduct tracking by correlation filter with a corrective measure by CNN detection at key frame to increase the performance, accuracy and meet the practical requirements. The framework has been designed with two particular modules: Position predicting module and Corrective measure module that will be elaborated in this section. Figure 1 shows the steps of our framework.

3.1. Tracking by correlation filter

The tracker we choose is the fast discriminative scale space tracker from [2], the correlation scores at the search region $Z^t_i$ are computed by $F^{-1} \{ Y_t \}$. [2] shows more robust performance than [18] in our experiment condition by standard translation filter and learned scale filter in the situation of object deformation and scale changes. The correlation filter-based tracker is more suitable and have comparable performance under the consideration of real word tracking conditions, taking into account the performance of deep CNNs-based tracker rely on hardware facilities.

$$Y_t = \frac{\Sigma_{i=1}^{d} A^i_{t-1} z^i_{t}}{B_{t-1} + \lambda}$$

We use the $A^i_{t-1}$ and $B_{t-1}$ to update the filter in the previous frame same as [2], and the search region $Z^t_i$ is generated by equation (2) in section 3.2.

The correlation filter used in object tracking has a drift problem. We have designed a module named corrective measure to handle this problem, in which we use CNN detection to corrective the
position of target when tracker drifts. And the module position predicting is designed for increasing speed to counterpoise the speed drop caused by detection.

3.2. Position prediction module

The main procedure of the tracking method we used is to extract the HOG feature of the previous frame, then to set up a search region to matching characteristics. The search region is obtained from expanding the target size two times with the midpoint of the target as the center. We find that even if search range is already reduced to two times of the target size from the whole image after numerous experiments, there is still a large disinterested area. Therefore, we propose a position prediction method to reduce the redundant cost of computation. The basis of this method is that no matter what kind of movement the object does, in a small time range, it can be regarded as a curve or even a straight line. If we fit the motion trend of the target in current frame according to the previous data, we can predict the moving direction and distance of the target in next frame. The search range $Z_t$ in equation (1) can be saved nearly half in terms of the computational cost and can be better applied into the fast motion situation.

We predict the position $(\hat{X}_t, \hat{Y}_t)$ of the current frame $t$ by fitting the central point of the target in the front $N$ frames. The information to be predicted includes the motion direction (angle) and the motion distance. Because the motion vector of arbitrary angle can be decomposed into sum of two vectors that directions in horizontal and longitudinal, we decompose the position information vector $\vec{d} = (x, y)$ on two dimensions into two one-dimensional information vectors $\vec{a} = x\vec{i} + y\vec{j}$, which reduces the computational complexity. We use a N-th-order difference equation based on Taylor series expansion. To predicting the position $\hat{X}_t$ and $\hat{Y}_t$ of the target at frame $t$ with the center position of the target $X_{t-n+i}$ and $Y_{t-n+i}$ in front $N$ frames, which i $\in$ (0, 1, …, N – 1) by:

$$ (\hat{X}_t, \hat{Y}_t) = \sum_{j=0}^{n-1} \sum_{i=0}^{j} \frac{(-1)^j}{j!} c_j (X_{t-n+i}, Y_{t-n+i}) + \varepsilon $$  \hspace{1cm} (2)

Where $\varepsilon$ is the error offset to reduce the error of the prediction. Then we obtain the motion vectors $\vec{a} = D_x^i \vec{i} + D_y^j \vec{j}$ in which $D_x^i = (\hat{X}_t - X_{t-1}), D_y^j = (\hat{Y}_t - Y_{t-1})$ we use the prediction $\hat{X}_t$ and $\hat{Y}_t$ that obtain from equation (2) to compute the predicted motion direction $\hat{\vec{d}}$ and distance $\hat{d}$, and calculate the size of the search range:

$$ \Lambda = \begin{cases} (D_x^i, D_y^j) & + (w, h) \; ; \text{others} \\ (w, h)/3 & + (w, h) \; ; \text{ } \begin{array}{c} D_x^i, D_y^j \end{array} = 0 \end{cases} $$  \hspace{1cm} (3)

Where $\Lambda$ is the extended length of search range based on target center at frame $t-1$, $w$ and $h$ are the width and height of the target size at frame $t-1$, respectively. $D_x^i$ and $D_y^j$ are the predicted distance of vertical and horizontal direction.

Figure 2 is an example of our position prediction that shows the search region is more accurate when comparing to the real position of the target in frame 1.

![Figure 2. An example of prediction. The image is extracted from sequence named BlurFace in OTB-100 at frame 302](image-url)

Figure 3 is an example of corrective measure. The images are extracted from sequence named Dudek in OTB-100 dataset at frame 925 and 926.

![Figure 3. An example of corrective measure.](image-url)
appropriate. This approach can reduce the computation cost and make tracker more capable even in fast motion situation. When object moves rapidly, there is a real possibility that the search region of old tracker could not combine the real position of target at the next frame, leading to the failure of tracking. Our method effectively solves this problem.

We also give an example in Figure 3(a) about equation (3), On the basis of \((\tilde{X}_t, \tilde{Y}_t)\), the extended width and height is based on predicting motion direction. The \(D_y\) in Figure 3(a) is zero, so the \(\Lambda\) of vertical direction is \((w, h)/3 + (w, h)\).

3.3. Detecting key frames by CNN Net

We use the detection algorithm based on neural network to provide object proposal for the corrective measure module. The algorithm needs to generate a set of parameter: frame number \(f_i\) for matching the frame at tracking module, and confidence score \(s_i\) for showing result and filter negative objects, and bounding box \(b_i\) for locating object.

3.3.1. Dataset preparation. For the selection of dataset, we choose the VOC (Visual Object Classes) and OTB-100 (Visual Tracker Benchmark) [5] dataset. For increasing the performance of algorithm when the image suddenly fuzzy because of the suddenly moving of object or shaking of equipment, the dataset has been adapted by adding moderate blur image in data sample. And we obtained the blur images by change the clear image with three different blur algorithm.

3.3.1.1. Gaussian blur. Assigning the weights with Gauss function and weighted average to calculating the transformation of each pixel in each images. Central pixel receives the highest weights. And the greater the distance to central pixel is, the smaller weight is received. We use Gaussian blur to imitate the lack of detail information of the target in the image due to insufficient resolution or the short distance between target and equipment.

3.3.1.2. Motion Blur. Motion blur is used to simulate the blur image as a result of suddenly moving of target or shaking of device, that weighted average to calculating the transformation of each pixel in each images and assign the weights with equation (4) when \(1 - |ycos\theta - xsin\theta| \geq 0:\)

\[
G(x, y) = 1 - |ycos\theta - xsin\theta| ; x \in [0, \cos\theta \cdot \text{len}/2], y \in [0, \sin\theta \cdot \text{len}/2], \theta \in [0, \pi]
\]  \tag{4}

Where \(x\) is the pixel’s horizontal distance to the central pixel, \(y\) is the pixel’s vertical distance to the central, \(\theta\) is the angle in which the picture rotates counter-clockwise, and len is the motion length. The result matrix is symmetric about the central pixel that we can only calculate quarter element of the matrix we decrease the speed. As the central pixel received the highest weight, the further the distance perpendicular to the motion vector and the distance to the central pixel, the lower weight is received.

3.3.1.3. Defocus Blur. Contrary to Gaussian Blur we used defocus blur to simulate the image when the distance from object to device is far relatively and cause the blur. This method designs the weights with a circular template, in which weights is related to the position of the pixel on the circular. The pixels outside the circle receive the weights 0, the pixels inside the circle receive the weights 1, and the pixels on the circle receive the weight with parameter \(m_1\) and \(m_2\) by:

\[
G(x, y) = \frac{r^2}{2} (\arcsin \left( \frac{m_2}{r} \right) - \arcsin \left( \frac{m_1}{r} \right)) + \frac{r^2}{4} (\sin \left( 2\arcsin \left( \frac{m_2}{2} \right) \right) - \sin \left( 2\arcsin \left( \frac{m_1}{2} \right) \right))
\]

\[- (\max(x, y) - 0.5) * (m_2 - m_1) + m_1 - (\min(x, y) - 0.5) \]  \tag{5}

Where \(x\) and \(y\) are the pixel’s horizontal and vertical distance to the central pixel, \(r\) is the radius of circle template, \(m_1\) and \(m_2\) are the vertical distance between the intersection of the circle and the pixel to the central pixel.
3.3.2. Structure of detection network. We use the VOC dataset pre-trained VGG16 [9] model provided by SSD and fine-tuning with the OTB-100 dataset to extracting object with classification and confidence score. There are five classes: person, biker, dog, bird, car, which are enough to tracking at general scene and can be extend for special scene. And removing negative object proposal which the confidence scores of VGG detection and bounding box information are below a certain threshold. We use 0.75 as confidence score threshold. And the objects with confidence score below the threshold are considered as disinterested objects.

3.4. Corrective measure module

The scale change and drift problem is a basic and common problem in correlation filter-based tracking field. When the size of target decreases, the filter will learn a lot form background information. If the target is expanding, the filter will go with the local texture. These two situations are likely to lead to unexpected results, drift and failure. Even fDSST [2] uses learned scale filter, the accuracy of tracking algorithm with the drift problem had been improved relative to filters like KCF [8] without scale update. But eventually cause drift or failure because of the accumulation of small error when the tracking task for a long time.

We design a corrective measure module in our framework shown in Figure 1(f) to solve this problem. Contrasting the results between tracker and detection algorithm at discontinuous key frame to determine whether the drift occurred, and correcting the track result by the detection algorithm while the drift occurring. The problem for that is 2-fold: how to correctly extract the bounding box of the target in detection results at the current frame; how to determine the drift occurred and update the bounding box for a new tracker. The position prediction proposed a search region like Figure 1(c) has narrowed the detection range from whole image to a predicting region. It increases the speed and filtering out the disinterested targets. Then we compare the detection proposal region DPRf with the tracking target Tt and design the corrective measure by overlap region ORf between tracking result and bounding box of detection algorithm. We divide the problem into following situations (with a premise that there is at least one DPRf obtained from detection):

- (situation 1) If the tracking target Tt is not overlapping with any proposal region DPRf it means tracker drifts seriously.
- (situation 2) If the tracking target Tt is only overlapping with one proposal region DPRf.
- (situation 3) If the tracking target Tt is overlapping with two or more proposal regions DPRf.

At the situation 1, because the tracking method of correlation filter is based on target information in prior frame Tt-1 and evaluate the target Tt in current frame with filter, we chose the DPRf as the result of current frame t and build a new filter for tracking. At the situation 2, Since drift may happen, we set up a threshold Thr to determine whether to correct the filter result. When the overlapping of the DPRf and the Tt is bigger than Thr, we consider the tracking performance is good enough. Because generally the detection of bounding box will contain some background information, we use the tracking results, and use the detection results contrarily. And at the situation 3, we choose the DPRf which has the maximum overlapping area with the target Tf and compare with Thr like situation 2.

\[
T = \begin{cases} 
  \text{DPR}_f ; & n = 0 (\text{situation 1}) \\
  \text{DPR}_f ; & n = 1 (\text{situation 2}), \text{OR}_f \leq \text{Thr} \\
  T^f ; & n = 1 (\text{situation 2}), \text{OR}_f > \text{Thr} 
\end{cases} 
\]

(6)

Where n is the number of the overlap between tracking target T and proposal region DPRf, T is the result of the tracker in current frame t.
Figure 3 shows an example of the corrective measure processing that target position at frame t-1 and prediction result of frame t in Figure 3(a). And the green box in Figure 3(b) is the search region’s result of detection and the red box is the filter result.

4. Experiments

In this chapter, we provide the experiment results to verify the performance of our tracking framework. Comparing with nine current and state-of-the-art tracker including MCPF [7], HDT [19], CF2 [20], DeepSRDCF [21], CNN-SVM [22], KCF [8], DSST [1], fDSST [2] and CCOT [23]. The tested fifty sequences we choose are extracted from OTB-100 mainly including person, animal and car as target.

We analyse the performance based on OTB-100 sequences in terms of 11 challenging factors, and show the representative results. The results showed in Figure 4 and Figure 5 using the distance precision and overlap success rate, illustrate the mean distance and overlap precision over all fifty sequences. The figures contain the AUC (area-under-the-curve) score and precision score of average distance for each tracker. Our tracker performs better compared with all nine compared trackers in both precision and success plots. Column (b) in Figure 4 and 5 illustrate precision and success plots of background clutter, column (c) illustrates the results of illumination variation and the column (d) illustrates the results of deformation. The results of success plots show that our method is capable of scale accommodation in different challenging factors.

Column (a) in Figure 4 and Figure 5 illustrate the success and precision plots on 10 trackers of fifty sequences on the OTB-100 Dataset. The top method MCPF [7] runs on real-time with an AUC score of 84.7%. Ours tracking method runs with score of 90.8%, significantly outperforming MCPF [7] by 6.1%. The overlapping rate is best without sacrificing too much location precision, which shows the more robust when size and appearance changing. Summarily, the precision plot proves that our framework is outperform in robustness compared with others, and the success plot demonstrates that our tracker tracks scale more accurately.
We also provide a comparison of our framework with several current trackers MCPF [7], CCOT [23], fDSST [2] and DSST [1] on representative video sequences named Gym, MotorRolling, Bolt2, Bird1 and CarScale. Shows the comparative result in various situation in Figure 6.

Figure 6(a), (b) and (e) illustrate sequences with significant scale variations and deformation. MCPF are capable in-plane rotation, but incompetent to significant scale changes. MCPF is bigger and others is smaller than real scale. Our tracker performance better on both the scale and position in spite of these challenges.

Figure 6(c) and (d) illustrate the results on sequences with partial occlusions by similar objects named Bolt2 and Bird1. Especially, the target is completely blocked for a while in the sequence Bird1 showed in Figure 6(d). The compared trackers are incompetent to the significant clutter and occlusions in the both two sequences. Figure 6(b) and (c) illustrate the results on sequences with fast motion and Figure 6(b) and (e) illustrate the results on sequences with blur motion. Our tracker demonstrates the best performance in all five representative sequences. As speed of tracking framework, our tracker shows average 30.7 fps with visualization in our experimental environment.

5. Conclusion
In this paper, we propose a multi-stage tracking framework that tracking by correlation filter. Our method consists of a corrective measure based on CNN detection at key frame to increase the accuracy and meet the practical requirements. To handle the shift between clarity and blur problem, we design the blur dataset based on VOC and OTB-100 generated by three blur methods to simulate the blur caused by different reasons. Experiments are executed by the challenging benchmark sequences with 11 attributes such as scale variation, deformation, motion blur, fast motion, and background clutter. Compared with above state-of-the-art tracking scheme, the results clearly certify that our method outperforms them on both success and precision plot of OPE. Since our detection and tracking methods are respectively independent that can be utilized in any multi-stage tracking framework for different realistic scenes.

Further studies beyond this work include improving the accuracy of position prediction and test the performance of our method in occlusion. And improve our tracker for multi target tracking.

Acknowledgments
This work was supported by The Aeronautical Science Foundations of China (No.20161375002) through a grants for our project.
References
[1] Danelljan M, Häger G, Khan F S and Felsberg M 2014 Accurate scale estimation for robust visual tracking *Proceedings of the British Machine Vision Conference* (Nottingham)
[2] Danelljan M, Häger G, Khan F S and Felsberg M 2017 Discriminative scale space tracking *IEEE Transactions on Pattern Analysis & Machine Intelligence* (vol 39 n 8) pp 1561-75
[3] Held D, Thrun S and Savarese S 2016 *Proc. European Conf. on Computer Vision* (vol 9905 LNCS) pp 749-65
[4] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: unified, real-time object detection *IEEE Computer Vision and Pattern Recognition* pp 779-88
[5] Wu Y, Lim J and Yang M H 2013 Online object tracking: a benchmark *IEEE Conf. on Computer Vision and Pattern Recognition* (vol 9) pp 2411-8
[6] Yang Y and Bilodeau G A 2016 Multiple object tracking with kernelized correlation filters in urban mixed traffic
[7] Zhang T Z, Xu C S and Yang M H 2017 Multi-task correlation particle filter for robust visual tracking *Computer Vision and Pattern Recognition.*
[8] Henriques J F, Caseiro R, Martins P and Batista J 2015 High-speed tracking with kernelized correlation filters *IEEE Transactions on Pattern Analysis and Machine Intelligence* (vol 37 n 3) pp 583-96
[9] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition *Computer Science*
[10] Liu W et al. 2016 SSD: Single shot multibox detector *Proc. European Conf. on Computer Vision* (vol 9905 LNCS) pp 21-37
[11] Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation *Proc. Computer Society Conf. Computer Vision Pattern Recognit* pp 580-7
[12] Uijlings J R R, Sande K E A V D, Gevers T and Smeulders A W M 2013 Selective search for object recognition *International Journal of Computer Vision* (vol 104 n 2) pp 154-71
[13] Szegedy C et al. 2015 Going deeper with convolutions *Proc. IEEE Computer Society Conf. Computer Vision Pattern Recognition* (vol 07-12-June-2015) pp 1-9
[14] Ouyang W et al. 2016 DeepID-Net: Deformable deep convolutional neural networks for object detection *Proc. Computer Society Conf. Computer Vision Pattern Recognition* pp 2403-12
[15] Girshick R 2015 Fast R-CNN *Int. Conf. on Computer Vision* pp 1440-8
[16] He K, Zhang X, Ren S and Sun J 2014 Spatial pyramid pooling in deep convolutional networks for visual recognition *European Conf. on Computer Vision* (vol 8691 LNCS) pp 346-61
[17] Ren S, He K, Girshick R and Sun J 2015 Faster R-CNN: towards real-time object detection with region proposal networks *IEEE Transactions Pattern Analysis Machine Intelligence* (vol 39 n 6) pp 1137-49
[18] Danelljan M, Bhat G, Khan F S and Felsberg M 2017 Efficient convolution operators for tracking *Proc. IEEE Computer Conf. and Pattern Recognition*
[19] Qi Y et al. 2016 Hedged deep tracking *Proc. IEEE Computer Society Conf. Computer Vision Pattern Recognition* pp 4303-11
[20] Ma C, Huang J B, Yang X and Yang M H 2015 Hierarchical convolutional features for visual tracking *Proc. IEEE Int. Conf. Computer Vision* pp 3074-82
[21] Danelljan M, Häger G, Khan F S and Felsberg M 2015 Convolutional features for correlation filter based visual tracking *Proc. Int. Conf. Computer Vision* (vol 2015-February) pp 621-9
[22] Hong S, You T, Kwak S and Han B 2015 Online tracking by learning discriminative saliency map with convolutional neural network *International Conference on Machine Learning* (vol 1) pp 597-606
[23] Danelljan M, Robinson A, Khan F and Felsberg M 2016 Beyond correlation filters: learning continuous convolution operators for visual tracking *European Conf. on Computer Vision*