ROBUST VISUAL TRACKING USING CORRELATION RESPONSE MAP

Hao Sheng, Kai Lv, Jiahui Chen, Wei Li

State Key Laboratory of Software Development Environment, School of Computer Science and Engineering
Beihang University, Beijing 100191, P. R. China

ABSTRACT

In this paper, we address the problem of heavy occlusion where the negative samples contaminate the translation model. In this setting, we decompose the task of tracking into translation and scale estimation of objects. We use hierarchical convolutional features to estimate target position and update translation model, and we use HOG features for the scale filter. In addition, we evaluate the translation's reliability according to the correlation responses map which is the result of correlation detection. Then we propose a new method to update model according to the reliability. Experiments are performed on 28 benchmark sequences with significant scale variations, it shows that the proposed algorithm performs favorably against state-of-the-art methods in terms of accuracy and robustness.

***Index Terms*—** heavy occlusion, negative samples, correlation response, correlation filters

Fig. 1. An example of our approach in challenging situations of fast motion, significant deformation and occlusion on the Lemming sequence [8]. At frame 333, our approach dynamically updates the learning rate rather than a constant one. The learning rate is set to 0 using our approach.

1. INTRODUCTION

Visual tracking is of great importance in security, human computer interaction and auto-control systems. A typical scenario of visual tracking is to track an arbitrary object initialized by position and scale in subsequent image frames and get the positions and scales. In this paper, we aim at using hierarchical convolutional features and correlation responses map to address the problem of heavy occlusion where the negative samples contaminate the translation model.

Correlation filter based trackers [1–7] are ranked top in terms of performances. Bolme et al. [1] propose a correlation filter based tracker, named Minimum Output Sum of Squared Error (MOSSE), which produces stable correlation filters when initialized with a single frame. Henriques et al. [4, 5] provide a link to Fourier analysis that opens up the possibility of extremely fast learning and detection with the fast Fourier transforms. They also propose a kernelized correlation filter (KCF) which uses a single kernel and enables fast learning with fast Fourier transforms instead of costly matrix operation, providing the highest tracking speed in benchmark [8]. Danelljan et al. [3] extend the KCF with low-dimensional adaptive color channels and suggest that color attributes provides superior performance for visual tracking. To better solve the partial occlusion issue, Liu et al. [6] propose a novel tracking method which track objects based on parts with multiple correlation filters. Ma et al. [7] exploit features extracted from deep convolutional neural networks trained on object recognition datasets to improve tracking accuracy and robustness. Then they adaptively learned correlation filters on each convolutional layer to encode the target appearance. Danelljan et al. propose the DSST tracker [2] learning adaptive multi-scale correlation filters using HOG features as target representation to handle the problem of scale change. Note that our proposed algorithm differs from existing methods based on correlation filters for the reason that we use hierarchical convolutional features to update translation model and we use HOG features for the scale filter.

As we need feature extractor to represent the target, feature representation is of prime importance in visual object tracking with the goal of discriminating the target from the background context. Feature representation has received considerable attention. In [9], discriminative local patches are selected to compute the target displacement using the Lucas-Kanade method. Similarly, Collins et al. [10] propose an online ranking mechanism for feature selection by measuring the variance ratio between object and background pixels. Grabner et al. use key points to describe the regions containing targets and surrounding context. Several hand-crafted local descriptors, such as SIFT [11], SURF [12] and orientation gradients(HOG) [13], have also been exploited as target representation. However, learning features form raw image pixels on large-scale dataset to deal with computer vision problems has made impressive progress compared with hand-crafted features. In this paper, we use hierarchical convolution features for translation model and HOG features for scale model respectively for the reason of computing efficiency and accuracy. For object visual tracking, the first step is to find out the most reliable positions. On this basis, we can estimate an appropriate scale. Also we find that the HOG feature is robust in scale estimating.

Our approach also builds on the observation based on prior work. It is critical to enhance the detection module to dynamically update the translation model and scale model. However, almost all correlation filter based trackers [1–7] simply update the model wiht a learning rate which is invariant during tracking. For model update, therefor, these method with constant learning rate is not adaptive. When the target is occluded or missing, as shown in Figure 1, this method contaminates the translation model and leads to a risk of drifting. We instead update the models depending on the correlation response which is the result of correlation detection. When the target is occluded, the learning rate is set to a small value or even zero.

2. PROPOSED ALGORITHM

As we aim to address the problem of heavy occlusion where the negative samples contaminate the translation model, we decompose the task of tracking into translation and scale estimation of objects. Also, we evaluate the translation's reliability according to the correlation responses map which is the result of a correlation filter (See Figure 2).

2.1. correlation tracking

Here we provide a brief overview of the correlation tracking. A correlation filter is a kind of template to model the appearance of the target. Firstly, an image patch x which is centered around the target is used to extract the original positive sample and the size is $[M, N] = (1 + padding) \times size(target)$. The patch x contains the target and some background. A training sample is defined as $F(x)$, and F is a feature extractor. The tracker uses all cyclic shifts of x : $x_{m,n} = circshift(x, [m, n]), (m, n) \in \{0, ..., M-1\} \times \{0, ..., N-1\}$to extract training samples. Each sample has a expected regression value $y_{m,n}$. For kernelized correlation filter, it is constructed by two parts: one is the feature template $F(x_{m,n})$, the other is coefficient α_i. A typical way to

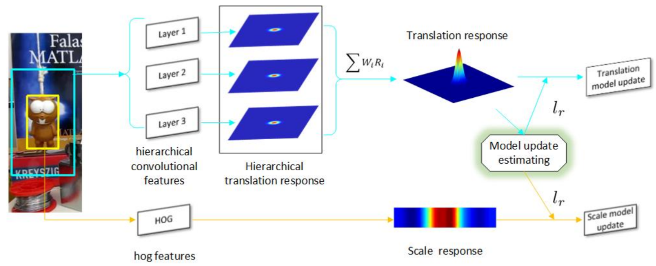

Fig. 2. We use hierarchical convolutional features to estimate target position and update translation model, and we use HOG features for the scale filter. The three convolutional layers extract features and generate responses map respectively. Then the three hierarchical translation responses generate a translation map by (Eq. 9). We dynamically get a learning rate for the translation model and the scale model depending on this translation response.

train a kernelized correlation filter is to get a function $f(z) = \sum_{m,n=1}^{M,N} \alpha_i k(F(z), F(x_{m,n}))$ that minimizes the cost function (Eq. 1) which is a kernelized ridge regression.

$$\min_{\alpha} \sum_{m,n} \left| f\left(F\left(x_{m,n}\right)\right) - y_{m,n} \right|^2 + \lambda \left| \sum_{m,n} \alpha_i \phi(F(x_{m,n})) \right|^2 \quad (1)$$

λ is a regularization parameter. The solution includes two parts: feature template $F(x)$ and coefficient α as

$$A = \mathcal{F}(\alpha) = \frac{\mathcal{F}(y)}{\mathcal{F}(\phi(F(x)) \cdot \phi(F(x))) + \lambda} \quad (2)$$

The $\mathcal{F}$ denotes the discrete Fourier operator. The ϕ is the mapping to the Hilbert space induced by the kernel k, defining the inner product as $\langle \phi(f), \phi(g) \rangle = k(f, g)$. The λ is same as Eq. 1 that controls overfitting. The first detection step is performed by cutting out a patch z of the same size and position in the new frame. The detection scores are calculated as

$$\mathcal{F}(y_{m,n}) = \sum_{i=1}^{n} \alpha_i k(F(z_{m,n}), F(x_i)) \quad (3)$$

where $z_{m,n} = circshift(z, [m, n])$ y. This can be simplified as

$$R = y = \mathcal{F}^{-1}(A \odot \mathcal{F}(\phi(F(z)) \cdot \phi(F(x)))) \quad (4)$$

Then we get a response matrix (Eq. 5). The target position in the new frame is estimated by finding the maximum of the response matrix. For example, given a response matrix

$$R = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & \cdots & r_{0,n-1} \\ r_{1,0} & r_{1,1} & r_{1,2} & \cdots & r_{1,n-1} \\ r_{2,0} & r_{2,1} & r_{2,2} & \cdots & r_{2,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m-1,0} & r_{m-1,1} & r_{m-1,2} & \cdots & r_{m-1,n-1} \end{bmatrix} \quad (5)$$

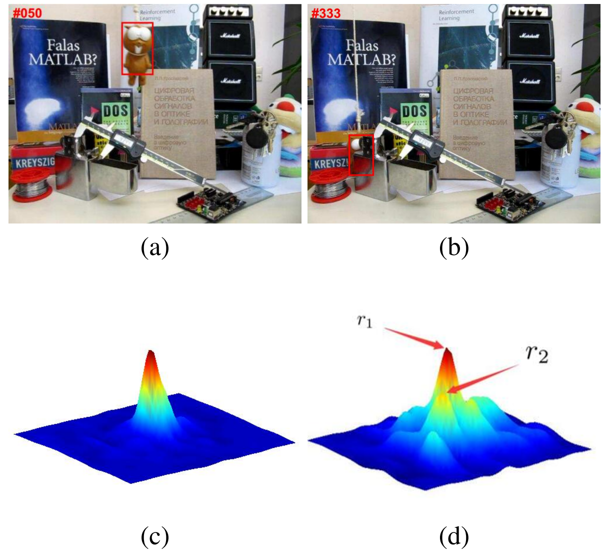

Fig. 3. (a) and (b) are the tracking frames and their corresponding response maps (c) and (d). The target in (b) is heavy occluded and its response performs abnormal: the response at the second highest extreme of response r_2 is half of the maximum of response map.

Suppose $r_{i,j}$ is the maximum of the response matrix. The relative displacement of object in vertical direction and horizontal direction $[vert\_delta, horiz\_delta]$ is defined as

$$vert\_delta = \begin{cases} i & , \text{ if } i < \frac{m-1}{2} \\ i - m + 1 & , \text{ else} \end{cases} \tag{6}$$

$$horiz\_delta = \begin{cases} j & , \text{ if } j < \frac{n-1}{2} \\ j - n + 1 & , \text{ else} \end{cases} \tag{7}$$

2.2. hierarchical convolutional representations

We use hierarchical convolutional features which come from VGG-Net [14] to estimate target position and update translation model. For correlation trackers, target representation must be a matrix retaining spatial resolution. For example, a $M \times N$ patch's feature should be $\frac{M}{4} \times \frac{N}{4} \times C$. As the fully-connected layers show little spatial resolution(i.e.1×1), we use three layers of intermediate representations to encode target appearance.

Due to the pooling and convolution operators used in the CNNs, spatial resolution is much smaller in the deeper convolutional layers. For example, the convolutional feature maps of $pool5$ in the VGG-Net are of spatial size 7×7, which is $\frac{1}{32}$ of the input image size 224×224. To unify different layers' convolutional features to same size and locate targets accurately, we resize each feature map to a fixed larger size with bilinear interpolation. The feature is defined as $F_{l_i}(x), l_i \in \{l_1, l_2, l_3\}$, l_1, l_2, l_3 denote the three layers of intermediate representations.

$$F(x) = [F_{l_1}(x), F_{l_2}(x), F_{l_3}(x)] \tag{8}$$

$$R = \sum_l w_l R_l \tag{9}$$

The interpolation weight w_l depends on the layer and $w_1 > w_2 > w_3$. The R_l (see Eq. 4) is generated by F_l feature respectively and R is finally used to determinate the position.

Let $M \times N$ be the target size in the first frame and S indicate the scales of target. We extract a scale patch P_s of size $sM \times sN$ centered around the estimated location. Here s denotes one scale of S and we resize all patches with size $M \times N$. In this paper, we use HOG features to construct the scale feature pyramid. Let R_s denote the correlation response of the target regressor to patch P_s, the optimal scale s of target is

$$s = \underset{s_i}{\operatorname{argmax}} \{R_{s_1}, R_{s_2}, R_{s_3} \cdots\}, s_i \in S \tag{10}$$

2.3. model update

Adaptability is important for the regression model to estimate the target positions. Unlike prior work which simply update model with constant learning rate, we propose a method to evaluate the translation's reliability according to the correlation responses (Eq. 5)(Eq. 9). According to this approach, we can find out whether the target is occluded or missing (see Figure 3). Then we dynamically adjust the learning rate. Therefore, our approach effectively adapts to appearance change and alleviates the risk of drifting.

In this paper, the peak-to-sidelobe ratio (PSR) (Eq. 11) is used to quantify the sharpness of the response peak and so evaluate the translation's reliability. The higher PSR value means more confident detection. Therefore, the PSR can be adopted to our approach.

$$PSR_i = \frac{r_i - \mu_i}{\sigma_i} \tag{11}$$

In this paper, we define the the highest extreme value as r_1 and the second highest extreme value r_2 (see Figure 3). Then the response matrix 11×11 which is centered around the extreme value is used to calculate the PSR. Thus, the PSR can describe the highest extreme value PSR_1 as well as the second highest extreme value PSR_2. μ_i and σ_i are the mean and the standard deviation of the 11×11 response matrix. z We then dynamically get the learning rate l_r and update model as

$$l_r = \begin{cases} \eta r_1 & , \text{ if } psr > \tau \text{ and } r_1 > wr_2 \\ 0 & , \text{ else} \end{cases} \tag{12}$$

$$M^t = (1 - l_r)M^{t-1} + l_r T^t \tag{13}$$

	boy	car4	carScale	couple	crossing	david	dog1	doll	dudek	fleetface	freeman1	freeman3	freeman4	girl
Ours	**100**	97.9	56	**81.4**	98.3	72.1	**100**	**99.6**	**100**	64.8	**54.9**	29.6	**45.9**	**100**
DSST	**100**	**100**	**84.5**	10.7	**100**	**100**	**100**	**99.6**	98.1	66.5	36.8	31.3	41.7	24.2
DLT	**100**	**100**	70.6	28.6	99.2	27	88.4	96	97.8	42.1	33.4	**85.2**	15.5	66.6
KCF	99.2	36.7	44.4	24.3	92.5	62.2	65.3	55.2	97.6	66.9	16	27.4	18.4	75.6
Struck	97.5	39.9	43.3	60.7	95.8	23.6	65.2	68.9	98.1	**78.1**	20.2	17.6	18.7	97
LSHT	50.7	27.6	44.8	9.29	40	28.2	54.3	23	89.9	65.5	18.4	15.7	20.1	14.4
TLD	82.9	24	68.7	22.9	45.8	61.1	75.6	69.3	67	44.1	23.3	64.6	21.6	72.6

	ironman	lemming	liquor	matrix	mRolling	shaking	singer1	skating1	skiing	soccer	trellis	walking	walking2	woman
Ours	**55.4**	**94.2**	52.5	**25**	**42.7**	98.9	**100**	30.3	9.88	34.2	92.8	83.5	**100**	93.1
DSST	13.3	26.9	40.9	18	6.71	**100**	**100**	**54.8**	4.94	**52.8**	**96.8**	**99.8**	**100**	93.3
DLT	6	24.3	36.3	1	7.3	92.6	**100**	48.5	**11.1**	13.8	31.8	46.4	**100**	80.2
KCF	15.7	43.1	**98.2**	13	7.9	1.4	27.6	36.3	6.2	39	84	51.5	37.8	**93.6**
Struck	4.82	63.8	40.5	11	15.2	52.9	29.9	31.3	4.94	18.1	54	52.4	43	93.5
LSHT	2.41	40.7	60.1	2	9.15	69.9	27.6	18.3	3.7	9.18	44.3	54.4	38.4	83.9
TLD	8.43	63.4	75.5	7	17.7	3.29	98.6	9	7.41	11.2	40.6	35.2	20.8	32.8

Table 1. Per-video overlap precision (OP) in percent on the 28 sequences. The best results are reported in bold. Our approach performs favourably compared to existing trackers.

where τ is the threshold and w in this paper is set to 4. $r_1 > wr_2$ means that r_1 is four times more than r_2. t is the index of the current frame. M denotes the translation model and scale model and T^t denotes the template in frame T which is used to update models, such as A (Eq. 2) and $\mathcal{F}(F(x))$. Contrary to traditional correlation filter based trackers, due to our adaptive model update method, even when the target is occluded at one frame, our approach can still maintain the accuracy of the classifier by slightly updating the model. Hence, it is able to relocate the occluded target when it appears in the following frames.

3. EXPERIMENTS

3.1. experimental settings

We employ all the 28 sequences annotated with the scale variation attribute in the benchmark [8]. We compare our approach with 6 state-of-art trackers, including DSST [2], DLT [15], KCF [5], Struck [16], LSHT [17], TLD [18], in benchmark [8]. We choose VGG-Net to extract hierarchical convolutional features and HOG descriptors to extract scale features. We use distance precision (DP) at a threshold of 0.5 to evaluate trackers' performance.

Our regularization parameter is set to λ =0.0001 in Eq. 1. As shown in Figure 2, the filter size is set to 2.5 times the initial target size. We use S = 33 number of scales with a scale factor of a = 1.02. The base learning rate is set to 0.02 for the translation model an 0.025 for the scale model. We use the same parameter values for all the sequences. The convolutional features are extracted from VGG-Net's layer 37, layer 28 and layer 19.

3.2. robust estimation with occlusion

In this section, we evaluate our approach in challenging situations of occlusion, such as Lemming, walking2 and girl. Table 1 shows the results of our approach. It is worth mentioning that for the most challenging Lemming sequence, none of the other 6 state-of-the-art methods are able to track targets well whereas our method achieves the distance precision rate of 94.2%. For the reason that the model update in this paper using dynamical learning rate, our approach can be highly adaptive. The model is updated in a comprehensive way, so our tracker achieves the best performance in these sequences.

3.3. overall performance evaluation

We evaluate the overall performance on the 28 sequences. Tabel 1 provides a per-video comparison with the top 6 existing trackers in our evaluation. The per-video results are presented using overlap precision (OP). Our approach provides better or competitive performance on 14 out of the 28 sequences. The 14 sequences are reported in bold. Our method provides a OP of 71.9% compared to 64.3% obtained by the method DSST [2]. Compared to DSST and the other trackers, our approach makes a significant improvement on overlap precision.

4. CONCLUSION

In this paper, we propose an effective algorithm for visual object tracking. Our approach decomposes the task of tracking into translation and scale estimation of objects. Also, we indicate that a dynamicallly updated model is comprehensive and adaptive for challenging situations of occlusion. The proposed tracker based on these findings achieves high performance on a large tracking benchmark. In future, we plan to further explore the potential of response map to other tracking difficulties.

5. REFERENCES

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

[2] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

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

[4] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision–ECCV 2012*, pp. 702–715. Springer, 2012.

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

[6] Ting Liu, Gang Wang, and Qingxiong Yang, "Real-time part-based visual tracking via adaptive correlation filters," *Intelligence*, p. 2345390, 2015.

[7] 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.

[8] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2411–2418.

[9] Jianbo Shi and Carlo Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[10] Robert T Collins, Yanxi Liu, and Marius Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1631–1643, 2005.

[11] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] Duy-Nguyen Ta, Wei-Chao Chen, Natasha Gelfand, and Kari Pulli, "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2937–2944.

[13] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[14] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] Naiyan Wang and Dit-Yan Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, 2013, pp. 809–817.

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

[17] Shengfeng He, Qingxiong Yang, Rynson WH Lau, Jiang Wang, and Ming-Hsuan Yang, "Visual tracking via locality sensitive histograms," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2427–2434.

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