A Robust Correlation Filtering Tracker with Resampling-Detection and Adaptive Fusion Multi-features

Yong Lu* and Mingbin Wang
Department of Electronic Information Engineering, Beijing Jiaotong University, Beijing, 100044, China

*Corresponding author. Email: 17120025@bjtu.edu.cn

Abstract. Recently, correlation filter is widely used in visual tracking for its robust and accuracy. However, it is still a challenge in tracking with complex situations such as target blurring, occlusion, and scale variation. In this paper, a correlation filter-based tracker with resampling-detection and scale estimation is proposed. We use multiple features with adaptive fusion to describe the target appearance, and resampling-detection module will be performed on the frame which tracking confidence determined by PSR is lower than a threshold. Besides, scale pyramid is introduced to estimate the scale. The extensive experimental evaluates on the OTB benchmark and results show that our approach outperforms the baseline trackers and has excellent performance in accuracy and robust, especially on the challenge of fast motion and motion blur. Additionally, our approach is computationally efficient and suitable for real-time applications.

Keywords: visual tracking, correlation filters, multi-features.

1. Introduction

Visual object tracking is a comprehensive research and has been widely used in intelligent surveillance, human-machine interaction, unmanned driving and remote medical diagnosis[1]. Benefit the accuracy and efficiency, correlation-filter-based trackers have been extremely studied in recent years[2-4].

The core of this method is tracking-by-detection, which can achieve training and detection within the same frame. The[5] transforms the complex spatial calculation into the Fourier domain’s element-wise product calculation, which greatly improves the speed of the algorithm. Then, the CSK[6] simplifies the dense sample sampling by taking advantage of the property of dense circular matrix. After that, the KCF[7] introduces the HOG feature and kernelized into the correlation filters. However, there are still two main limitations that restrict the accuracy of the KCF. First, the fast motion or blur of the foreground may lead to the model learning drift and result in tracking failure. Second, because the scale of the algorithm is fixed, it cannot adapt to the scale change of the object.

In this paper, we considered the problems mentioned above and proposed a novel resampling-detection and adaptive fusion multi-features correlation filter tracker, and the major contributions of this work can be summarized as follows. First, we proposed the multi-sampling strategy in the process of tracking, which introduce the PSR to judge whether current frame has the correct result, and the new position will be reevaluate in these sample regions if necessary. Second, Besides HOG feature, Color Name (CN)[8] features also be adopted to describe the target in the model.
Particularly, we combined them adaptively with a linear equation and the weights is adaptive determined by their response scores. Third, a scale pyramid is introduced to the tracking process to determine the size of the target dynamically. Additionally, we tested our algorithms on the Object Tracking Benchmark (OTB) [9] and got the ideal result compared to the KCF with the speed over 50 FPS.

2. Related work

According to the difference of frameworks and strategies, the visual tracking algorithms can be achieved using two different approaches: generative methods and discriminative methods [10]. Generative tracking is to model the appearance of the target, and then find the most similar region of the model in the subsequent frames as the target position, such as Optical flow[11], MeanShift[12]. However, it does not consider the background information and easy to make tracking errors.

The discriminative tracking algorithm extracts target from background information by constructing a classifier. Struck[13] is one the most representative tracker of them. It employs the structured Support Vector Machine (SVM) to directly link the target’s location space with the training samples and achieves the appealing result in the recent benchmark. Tracking-Learning-Detection (TLD)[14] used the redetection function to improve the robust of tracking thus has a good outperform in the challenge videos. Then [5] proposed a Minimum Output Sum of Squared Error (MOSSE), which transformed the convolution calculation in the spatial domain into the element-wise calculation in the Fourier domain and achieved the fastest tracking speed nearly 670FPS. In view of the efficiency of the kernel function, Henriques et al. proposed a Circulant Structure with Kernels (CSK)[6] and explores the structure of circulant matrix to improve the classification performance of the target appearance with dense sampling. Then they extended the grayscale feature to the HOG feature and proposed kernelized Correlation Filter (KCF). Another way, a powerful feature descriptor called Color Naming (CN)[8] was adopt into tracking, which can enhance the accuracy and robust of tracker with HOG. After that, these two features are widely used in visual tracking, such as DSST, STAPLE[15], SAMF[16]. In these trackers, scale estimate was introduced. After comprehensive consideration, our proposed approach is based on KCF and CN tracker which achieve robust results.

3. Proposed method

In this section, we first have a brief review about the Kernelized Correlation Filter (KCF), and then introduce the scale estimation with scale pyramid. Next, multi-features adaptive fusion strategy is present. Moreover, the multi-sampling strategy we proposed is described in the end.

3.1. The Kernelized correlation filter with scale estimation

KCF trackers based on the framework of correlation filter. As basic, ridge regression is used to train the filter and the convolution operation in time domain is transformed into the element-wise product successfully. Particularity, cyclic shifting was proposed to make dense sampling and reduce the computational quantity enormously in the Fourier domain. Assuming that a patch with an object of interest is denoted with a vector \( \mathbf{x} = [x_1, x_2, x_3, ... x_n] \), cyclic shift of \( \mathbf{x} \) is \( \mathbf{P}\mathbf{x} = [x_n, x_1, x_2, ..., x_{n-1}] \). Therefore, \( \{\mathbf{P}^u\mathbf{x} | u = 0...n-1\} \) contains all the cyclic shift visual samples, which can be drawn as a circulant matrix \( \mathbf{X} = C(\mathbf{x}) \). It has an intriguing property[7]: \( \mathbf{X} = \mathbf{F}\text{diag}(\hat{\mathbf{x}})\mathbf{F}^H \). Where \( \hat{\mathbf{x}} \) denotes the DFT transform of \( \tilde{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \), \( F \) is known as the DFT matrix, and \( \mathbf{F}^H \) is the Hermitian transpose of \( \mathbf{F} \). In the KCF, this property is used to simplify the solution of linear ridge regression:

\[
\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2
\]  

(1)

where \( \lambda \) is a regularization parameter that controls overfitting, and the function \( f \) can be rewrite as:

\( f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \). With the “intriguing property” , we can have the solution of (1): \( \mathbf{w} = \frac{\hat{\mathbf{x}}^* \circ \mathbf{y}}{\hat{\mathbf{x}}^* \circ \hat{\mathbf{x}} + \lambda} \). Where \( \circ \)
denotes the element-wise product in the Fourier domain, $\hat{x}^*$ represents the complex-conjugate of $\hat{x}$. With the non-linear kernel trick: $f(z) = w^T z = \sum_{i=1}^{n} \alpha_i k(z, x_i)$, we can derive the solution of $\alpha$:

$$\hat{\alpha} = \frac{\hat{y}}{k^{xx} + \lambda}$$  \hfill (2)

Here $\alpha_i$ is the linear dual space coefficient matrix what we used to detect and $\hat{k}^{xx}$ is defined as kernel correlation. The Gaussian kernel is adopted: $k^{xx} = \exp\left(-\frac{1}{\sigma^2} \|x\|^2 + \|x\|^2 - 2\mathcal{F}^{-1}(\hat{x}^* \circ \hat{x})\)$.

So, the current frame $z$ response value of sample patch can be computed conveniently:

$$f(z) = \hat{k}^{xx} \circ \hat{\alpha}$$  \hfill (3)

where $x$ is the trained model in the previous frame, and $\hat{k}^{xx}$ is the kernel of $x$ and $z$. The maximum value’s location is the target position of current frame. In order to adapt the change of target appearance, the model is updated per-frame with learning rate $\gamma$:

$$\hat{x}_{t} = (1-\gamma)\hat{x}_{t-1} + \gamma \hat{x}_{t}$$  
$$\hat{\alpha}_{t} = (1-\gamma)\hat{\alpha}_{t-1} + \gamma \hat{\alpha}_{t}$$  \hfill (4)

Here $\hat{x}_{t-1}$ and $\hat{\alpha}_{t-1}$ is the model of last frame, $\hat{x}_{t}$ and $\hat{\alpha}_{t}$ is the current model update with the learning rate. In this way, we only need to pay attention on the previous frame rather than all of the sequence.

Aiming at the problem of scale variation in the process of target movement, [17] proposed an accurate and effective method for scale estimation. Similar to the (1), the scale estimation is used for position estimation and achieved by minimizing the cost function:

$$\min_{w} \sum_{l} \left( \sum_{d=1}^{d} w^l * f^l - g^l \right)^2 + \lambda \sum_{l=1}^{d} \|w^l\|^2$$  \hfill (5)

Here, $f^l$ is an image patch of size $d'M \times d'N$ extracted from the centre of target, where $M$ and $N$ denote the target size in the current frame, $l \in \left\{ \left\lfloor -\frac{S-1}{2} \right\rfloor, \ldots, \left\lceil \frac{S-1}{2} \right\rceil \right\}$ and $S$ is the size of the scale filter. $g$ is the desired correlation output of the training example $f$. $w^l$ is parameter of scale correlation filters. We can get the solution of (6) computing in the Fourier domain:

$$w^l = \frac{G \mathcal{F}^l}{\sum_{l=1}^{d} \mathcal{F}^l \mathcal{F}^l + \lambda} = \frac{A^l}{B^l}$$  \hfill (6)

The numerator and denominator are updated separately, similar to (4):

$$A^l = (1-\gamma)A^l_{t-1} + \gamma G \mathcal{F}^l$$  
$$B^l = (1-\gamma)B^l_{t-1} + \gamma \sum_{l=1}^{d} \mathcal{F}^l \mathcal{F}^l$$  \hfill (7)

Then the new target scale is found by maximizing the score $\hat{y}_s = \mathcal{F}^{-1}\left\{ \frac{\sum_{l=1}^{d} A^l}{B^l + \lambda} \right\}$. In this way, we can find the appropriate scale while determining the target position.

### 3.2. Adaptive fusion of multi-features

Actually, single feature is usually not enough to describe the target accurately. So, we introduce two complementary features in our tracker, which contain both the shape and color information of the target.

#### Histogram of Gradient

HOG is a popular feature descriptor used for object detection in computer vision and image processing. It constitutes features by calculating and counting the histogram of gradient direction in the local area of the image. The image is first divided into small connected areas
called cell. Then the orientation histogram of the gradient or edge of each pixel in the cell is collected. Finally, these histograms can be combined to form a feature descriptor. As in [7], we use the 31 gradient orientation bins variant replace the traditional hog feature.

**Color-Naming.** Color Naming (CN, or Color Attributes) is several language tags used by humans to describe colors. [18] proposed a method for automatically learning eleven basic colors in the English language from Google Images, that is, the RGB values of the image maps into 11 predefined colors: red, orange, yellow, green, black, blue, purple, brown, gray, white, and pink. [8] uses the CN feature for correlated filter tracking, and evaluates it with other mainstream color descriptors (such as RGB, LAB, HSV, HUE, etc.). Experimental results show that CN feature descriptors have obvious advantages in tracking accuracy.

**Adaptive fusion.** Here we introduce the PSR to determine the contribution of the two features:

\[
PSR = \frac{\mu - g_{\text{max}}}{\sigma}
\]

(8)

where \(\mu\) and \(\sigma\) represent the mean and standard deviation of the response \(F^{-1}\{\hat{f}(z)\}\), and \(g_{\text{max}}\) is the maximum value inside. Then we can express the weight of two features as

\[
w_{\text{HOG}} = \frac{PSR_{\text{HOG}}}{PSR_{\text{HOG}} + PSR_{\text{CN}}} \quad \text{and} \quad w_{\text{CN}} = \frac{PSR_{\text{CN}}}{PSR_{\text{HOG}} + PSR_{\text{CN}}}
\]

(9)

Thus, the target position in the current frame can be derived as

\[
P = w_{\text{HOG}} \times P_{\text{HOG}} + w_{\text{CN}} \times P_{\text{CN}}
\]

(10)

where \(P_{\text{HOG}}\) and \(P_{\text{CN}}\) are positions obtained separately from two features’ response.

![Figure 1. PSR and PSR\text{ratio} of Tiger1 (frames:23,31,60,84).](image1)

![Figure 2. Resampling redetection (blurOwl: frame109).](image2)

### 3.3. Model update with resampling-detection

In this section, we build a judgment mechanism based on the tracking confidence of the current frame. A threshold \(T\) is introduced to judge whether tracking results are accuracy. The quality of the tracking results also can be evaluated by PSR. As shown in figure 1, we can see that when the target is occlusion (frames 23,60), deformed (frame 31), and blurred (frame 84), the PSR curves drop...
dramatically. But we noticed that the maximum value of PSR will be different when object patches with the different size and features. Therefore, we take the \( PSR^j \) as the judgment criterion:

\[
PSR^j = \frac{PSR(j)}{PSR(1)} \tag{11}
\]

where \( j \in \{1,2, \ldots, n\} \), is the label of the current frame. \( PSR(j) \) and \( PSR(1) \) are the PSRs of the \( j \)-th frame and the first frame.

When the \( PSR_{ratio} \) of current frame is lower than \( T \), the current frame tracking should be considered as error. Then the tracking algorithm is used in the sampling regions (with adaptive offset pixel \( M \)) around the target position tracked in current frame, as shown in figure 2.

As mentioned in Section 3.2, we use two features to determine the location of the tracking target independently. Therefore, we can adaptively determine the target position according to the tracking confidence of each feature after adding the resampling detection strategy as equation (12).

\[
P^i = \begin{cases} 
  w_{HOG} \times P^i_{HOG} + w_{CN} \times P^i_{CN}, & \text{if } PSR^{HOG}_{ratio} > T \& PSR^{CN}_{ratio} > T \\
  P^i_{HOG}, & \text{if } PSR^{HOG}_{ratio} > T \& PSR^{CN}_{ratio} < T \\
  P^i_{CN}, & \text{if } PSR^{HOG}_{ratio} < T \& PSR^{CN}_{ratio} > T \\
  P^i, & \text{else}
\end{cases} \tag{12}
\]

Here \( PSR_{ratio} \) is the results of resampling detection of two features.

3.4. Whole process of our approach

The whole process of tracking method that we proposed can be described as table 1.

| Table 1. The pseudo-codes of our proposed method. |
|---|

**Algorithm 1.**

**Input:** Current frame \( I_t \),
- Resampling threshold \( T \).
- Target position and scale of last frame: \( P_{t-1}, S_{t-1} \).
- Previous model of translation \( \hat{x}_{t-1}, \hat{\alpha}_{t-1} \) and scale estimation \( A_{t-1}^l, B_{t-1}^l \)

**Output:** New position \( P_t \) and new scale \( S_t \).
- Updated model of translation \( \hat{x}_t, \hat{\alpha}_t \) and scale estimation \( A_t^l, B_t^l \)

**Position estimation**

1: Extract the patch \( z_{trans} \) from \( I_t \) at \( P_{t-1} \) with \( S_{t-1} \).
2: Compute the translation correlation with \( \hat{x}_{t-1}, \hat{\alpha}_{t-1} \).
3: get the new position \( P_{t,HOG} \) and \( P_{t,CN} \) at the maximum of responses \( \hat{f}(z)_{HOG} \) and \( \hat{f}(z)_{CN} \) using two features, using (12).
4: Compute the \( PSR_{ratio-HOG} \) and \( PSR_{ratio-CN} \). If lower than \( T \) (both or one of them), perform resampling-detection module.
5: Determine the new position \( P_t \) by equation (12).

**Resampling-detection module**

R-1: Resampling patches \( z_{HOG} \) and \( z_{CN} \) from \( I_t \) at \( P_{t,HOG} \) and \( P_{t,CN} \) with \( S_{t-1} \), offset pixel \( M \).
R-2: Perform position estimation module on each patch, return the new position \( P_{t,HOG} \) and \( P_{t,CN} \) of two features at the maximum of its all responses respectively.

**Scale estimation**

6: Extract the patch \( z_{scale} \) from \( I_t \) at \( P_{t-1} \) with \( S_{t-1} \).
7: Compute the scale correlation with \( A_{t-1}^l, B_{t-1}^l \).
8: Get the new scale \( S_t \) at the maximum of responses \( \hat{y}_s \) using HOG feature.
Model update

9: Extract the patch $z_{\text{scale}}$ and $z_{\text{trans}}$ from $I_t$ at $P_t$ with $S_t$.
10: Update the translation model $\mathbf{x}_t, \mathbf{a}_t$ by equation (4).
11: Update the scale model $A^t_l, B^t_l$ by equation (7).

4. Experiments

In this section, we first show the performance that replacing the HOG features with adaptive fusion multi-features and resampling-detection strategy in the tracker. Then our approach is test and compared with several out-of-arts correlation filter trackers and we show the quantitative results achieved on the OTB-50 benchmark. Finally, we provide several qualitative results in detail.

4.1. Parameters

The regularization parameter $\lambda$ is 0.0001, the learning rate $\gamma$ of HOG feature is 0.02, the CN feature is 0.075. we set the scale factor $a=1.02$ and $S=33$. The threshold $T$ is set to 0.4, and adaptive offset pixel $M$ is set to equal to a quarter of the current tracking window size. Other parameters are same as shown in [7](with HOG feature) [8](with CN feature).

4.2. Experimental Setup

The algorithm proposed in this paper is implemented in MATLAB. Our tracker is running on an Intel Core i7-8750H 2.20 GHz CPU with 8G RAM.

Datasets. The algorithm is test on the OTB benchmark, which has 50 sequences and manually divided into 11 attributes include illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), low resolution (LR), out of view (OV), background clutter (BC), motion blur (MB), in-plane rotation (IPR), out of plane rotation (OPR) and fast motion (FM) [29,30] to represent challenging aspects of visual tracking.

Evaluation Methodology. We following the protocol used in [9] to evaluate the trackers. The metrics that we use to present the results are centre location error (CLE), distance precision (DP), and overlap precision (OP). The first metric CLE is the average Euclidean distance between the centre position of the tracker estimated and its ground-truth. The second metric, DP, is compute the number of frames whose CLE is less than a certain threshold as a percentage of all frames. The third metric, OP, is defined as the overlap rate between the tracking result (usually represented by a bounding box) and the ground-truth. With DP, we can get the precision plots using the threshold within a range and the average DP at 20 pixels will be reported in the plots’ legend. With OP, we can get the curve of overlap rate as the success plots, and the area under the curve (AUC) of OP is included in the legend of a plot.

4.3. Compared with several representative trackers

In this section, the trackers included in the comparison are Struck [13], CSK[6], KCF [7], CN[8], SAMF[16], Staple [15]. The results are shown in table 2 and figure 4.

From Table 2 and figure 4, we can see that our algorithm achieves the highest performance on mean DP (in 20 pixels) compared to other trackers, and improves the tracking accuracy by about 5% compared to KCF. In the OP, the algorithm proposed in this paper also reaches the sub-optimal level, second only to the Staple, which is about 3% higher than the KCF algorithm, and 14.7% higher than the CN. In addition, the running speed of the algorithm reaches 55fps, which is lower than KCF and CN, but fully meets the requirements of real-time performance. Therefore, it can be considered that the algorithm proposed in this paper is robust, accurate and real-time.
4.4. Experiment with different challenge attributes

Table 3 shows the results that our methods and other trackers run with the challenge attributes. It can be seen that our proposed algorithm achieves the excellent results in five cases among 11 attributes, namely motion blur (MB), illumination variation (IV), scale variation (SV), fast motion (FM), in-plane rotation (IPR). And reaches the sub-optimal in two cases, which is out of plane rotation (OPR), background clutter (BC).

The details of some of the challenges are shown in Figure 5. The number in the title brackets represent the number of sequences containing such attributes. In general, the algorithm proposed in this paper has an excellent performance on the situation of moving blur, fast movement, target out-of-field and scale changes.

| Trackers performance in different challenge attributes (Mean DP (20 px)). |
|---|---|---|---|---|---|---|---|---|---|
| (%) | IV | SV | OCC | DEF | MB | FM | IPR | OPR | OV | BC | LR |
| ours | 74.5 | 78.8 | 75.6 | 68.9 | 74.5 | 71.0 | 80.1 | 76.7 | 70.1 | 76.1 | 62.9 |
| Staple | 72.9 | 73.3 | 78.2 | 76.3 | 71.9 | 67.3 | 78.7 | 76.2 | 77.6 | 74.8 | 49.2 |
| SAMF | 68.4 | 74.0 | 85.9 | 78.8 | 61.4 | 64.5 | 72.3 | 76.8 | 76.0 | 69.3 | 64.3 |
| KCF | 73.5 | 69.1 | 75.8 | 71.0 | 68.9 | 62.6 | 73.4 | 72.6 | 73.7 | 76.8 | 43.5 |
| CN | 58.7 | 61.3 | 63.3 | 57.7 | 58.6 | 52.0 | 69.2 | 64.8 | 49.2 | 66.8 | 49.2 |
| CSK | 46.4 | 51.1 | 52.5 | 47.1 | 36.1 | 39.6 | 57.9 | 54.9 | 42.9 | 58.7 | 50.4 |
| Struck | 55.4 | 66.0 | 60.0 | 52.2 | 59.1 | 63.4 | 65.5 | 61.4 | 62.4 | 58.7 | 68.9 |
4.5. Qualitative evaluation on several specific sequences

In this subsection, we select 4 different video sequences to test 4 trackers. Figure 6 shows the detail that different trackers run with different challenges in a particular sequence. As can be seen in figure 6, our tracker has good performance when the background illumination changes dramatically (6a), the tracked object is blurred (6b), the background color is relatively complex and similar to the foreground (6c) and the target is occluded (6d).

In a nutshell, the algorithm proposed in this paper combines the advantages of HOG feature and CN feature tracking to overcome the tracking drift caused by insufficient description of single feature (as shown in Figures 6a, 6c). And combined with resampling detection strategies, when the target is blurred and occluded (such as 6b, frame364, 6d, frame41), the correct target can be found quickly again in subsequent frames, which effectively reduces the error accumulation caused by tracking failure.

5. Conclusion

In this paper, we propose a robust correlation filtering tracker with resampling-detection and adaptive fusion multi-features. Our method is based on the traditional correlation filter trackers and replace the object feature by adaptive fuse the HOG and CN. Additionally, we add the resampling-detection module
to the trackers, which enhance the tracker performance when tracking the fast motion targets and face the challenge of motion blur.

Experiments are performed on the OTB benchmark, qualitative and quantitative results are used to validate our method. From the results, we can draw a conclusion that our approach is robust and suitable for real-time applications.

References
[1] Q. Hu, Y. Guo, Z. Lin, A. Wei, H. Cheng, Robust and Real-Time Object Tracking Using Scale-Adaptive Correlation Filters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (2014) 1012-1025.
[2] H. Wang, S. Zhang, Mutual kernelized correlation filters with elastic net constraint for visual tracking, EURASIP Journal on Image and Video Processing, 2019 (2019).
[3] H. Masood, S. Rehman, A. Khan, F. Riaz, A. Hassan, M. Abbas, Approximate Proximal Gradient-Based Correlation Filter for Target Tracking in Videos: A Unified Approach, Arabian Journal for Science and Engineering, 44 (2019).
[4] Y. Fang, S. Ko, G.-S. Jo, Visual tracking based on a unified tracking-and-detection framework with spatial-temporal consistency filtering, Computers and Electrical Engineering, 80 (2019).
[5] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, 2010.
[6] J.F. Henriques, C. Rui, P. Martins, J. Batista, Exploiting the Circulant Structure of Tracking-by-Detection with Kernels, in 2012.
[7] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-Speed Tracking with Kernelized Correlation Filters, IEEE Transactions on Pattern Analysis & Machine Intelligence, 37 (2015) 583-596.
[8] M. Danelljan, F.S. Khan, M. Felsberg, J.V.D. Weijer, Adaptive Color Attributes for Real-Time Visual Tracking, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[9] Y. Wu, J. Lim, M. Yang, Object Tracking Benchmark, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37 (2015) 1834-1848.
[10] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual Tracking: An Experimental Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (2014) 1442-1468.
[11] S. Baker, I. Matthews, Lucas-Kanade 20 Years On: A Unifying Framework, International Journal of Computer Vision, 56 (2004) 221-255.
[12] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Computer Vision and Pattern Recognition, 2000, 2000.
[13] S. Hare, A. Saffari, P. Torr, Struck: Structured output tracking with kernels, in: IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, 2011.
[14] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-Learning-Detection, IEEE transactions on pattern analysis and machine intelligence, 34 (2011).
[15] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P.H.S. Torr, Staple: Complementary Learners for Real-Time Tracking, in: International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[16] Y. Li, J. Zhu, A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration, in: L. Agapito, M.M. Bronstein, C. Rother (Eds.) Computer Vision - ECCV 2014 Workshops, Springer International Publishing, Cham, 2015, pp. 254-265.
[17] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, M. Krístan, Discriminative Correlation Filter with Channel and Spatial Reliability, International Journal of Computer Vision, 126 (2016) 671-688.
[18] d.W. Van, J., C. Schmid, J. Verbeek, D. Larlus, Learning Color Names for Real-World Applications, IEEE Transactions on Image Processing, 18 (2009) 1512-1523.