International Journal of Innovative
Computing, Information and Control
Volume **14**, Number **4**, August **2018**

ICIC International ©2018 ISSN 1349-4198
pp. **1239–1252**

AN IMPROVED TRACKING METHOD BASED ON KERNELIZED CORRELATION FILTER WITH A UNION FEATURE

Zhengpei Mo[1], Jianjun Ni[1,2,*], Pengfei Shi[1] and Xinnan Fan[1,2]

[1]College of IOT Engineering
[2]Jiangsu Universities and Colleges Key Laboratory of Special Robot Technology
Hohai University
No. 200, North Jinling Road, Changzhou 213022, P. R. China
{ mo1262510125; fanxn519 }@163.com; *Corresponding author: njjhhuc@gmail.com
flyshn@hotmail.com

Received December 2017; revised April 2018

Abstract. *Object tracking is one of the most important topics in the computer vision field. Tracking methods using discriminative models are becoming more and more attractive due to their outperformance compared with the ones using the generative models. The kernelized correlation filter based trackers achieve remarkable performance because of its high speed and accuracy, but there are still some limitations needed to be improved. In this paper, a new kernelized correlation filter based tracker using a new union feature is proposed, which is computed by using two feature matrixes of the modified Histogram of Oriented Gradient (fHOG) feature and Local Binary Pattern (LBP) feature to represent the target. The on-line object tracking benchmark is used to evaluate the method proposed in the paper, and the experiment results show that the proposed method can achieve outstanding performance compared with some state-of-the-art methods and the original kernelized correlation filter method, under the scenario that the target has rich texture features.*

Keywords: Object tracking, Correlation filter, fHOG feature, LBP feature

1. **Introduction.** Visual object tracking is widely used in many applications, such as video surveillance, robotics, unmanned aerial vehicles, intelligent driverless cars and human computer interaction [1, 2, 3]. The goal of visual object tracking is to track the target stably after giving the target position in the first frame, regardless of the occlusion, deformation, scale variation, fast motion, etc. [4]. Therefore, an excellent tracker should handle all of these challenging factors; meanwhile it should have high precision and high tracking speed, so as to be used in the real-time application. Nowadays, a lot of modern trackers have been proposed. Different from the generative ones, they use a discriminative classifier to distinguish the target from the background [5].

Correlation filter based trackers achieve their effectiveness in localization, because the corresponding response peak will be generated when the interest target is encountered in the video scenario during the process of designing the correlation filter, while the response value will be lower in the background. The correlation filter based trackers have drawn a great deal of attention [6, 7]. In these trackers, the correlation filter combined with a self-adaptive training strategy is often used as a detector, which would give a correlation response peak value output when it meets the region of interest, while giving a low output when it meets the background. So these methods above are very easy but efficient to classify the target of interest and the background. Thus, how to represent a target is of great importance in the process of tracking the target of interest [8, 9]. A lot of methods

DOI: 10.24507/ijicic.14.04.1239

have been proposed to deal with this problem. For example, the modified Histogram of Oriented Gradients (fHOG) features and raw pixel are used by the Kernelized Correlation Filter (KCF) tracker to represent the target in the process of target tracking respectively [10, 11]. Sometimes, this representation does not give a good result because the fHOG is usually used to describe the local object appearance and shape within an image, and it is described by the distribution of intensity gradients or edge directions [10]. The color naming feature is also used to describe the target to get a better representation of the target of interest [12].

Lots of research focused on the combinations of various methods for the target representation. For example, Li and Zhu [13] combined the raw pixel, color naming and the Histogram of Oriented Gradients (HOG) together to get a fusion feature to robustly represent the tracking target. Danelljan et al. [14] combined the color attributes into the tracking-by-detection framework. Sometimes, the above features ignore the target texture information and therefore fail to describe the target properly. The Local Binary Pattern (LBP) feature is used to describe the local texture feature within an image, which can also be invariant of gray scale and rotation. The LBP feature and its variants are widely used in the face recognition, pedestrian and so on [15, 16]. Zheng et al. [17] pointed out that the proper utilization of the HOG and LBP feature together would improve the performance of the pedestrian detection. Zhang et al. [18] proposed a boosted local structured HOG-LBP based object detector using a bottom-up paradigm.

In this paper, a new feature matrix is constructed by using the fHOG and the LBP features as they can better represent both the texture and the edge of the target. Here, a novel method is used to construct the new feature matrix, which is different from the existing one, where the fHOG feature is not concatenated with the LBP feature directly [19]. A novel method is proposed to construct a new feature matrix by superimposing the fHOG matrix with the LBP feature matrix with a coefficient. In addition, the procedure to calculate the LBP feature matrix differs from other papers.

This paper is organized as follows. Section 2 gives out the way to extract the fHOG feature and the LBP feature and then the method to construct a new feature matrix is explained in detail. The new feature matrix is combined into the framework of KCF tracker in Section 3. The proposed method is evaluated on the on-line object tracking benchmark in Section 4. Finally, some conclusions are given out in Section 5.

2. Feature Extraction.

The first step to build the tracker is to extract the features from the target of interest, in order to use these features to separate the target from the background. In this paper, the fHOG feature and LBP feature are used to extract both the local textures and the edge information.

2.1. Process of extracting the fHOG feature.

To extract the HOG feature, the orientation $\theta(i,j)$ and magnitude $r(i,j)$ of intensity gradient of every pixel (i,j) should be calculated firstly, using the 1-D centred point discrete derivative mask $[-1,0,1]$ and $[-1,0,1]^T$. Then the gradient orientation is discretized to get the sensitive S_1 and insensitive S_2, which are calculated by

$$S_1(i,j) = round\left(\frac{p \times \theta(i,j)}{2\pi}\right) \bmod p \tag{1}$$

$$S_2(i,j) = round\left(\frac{p \times \theta(i,j)}{\pi}\right) \bmod p \tag{2}$$

where p is the orientation bins number.

We can now get the pixel-level feature map $P(i,j)$ by

$$P(i,j)_s = \begin{cases} r(i,j) & \text{if } (s = S(i,j)) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where S is used to donate S_1 and S_2, $s \in \{0,1,\ldots,p-1\}$, and the size of $P(i,j)$ is $M \times N$.

Instead of using the pixel-level feature map, the cell feature map is used, as it will provide invariance to small deformations and reduce the feature map size. Let $l > 0$ be the length of side of a square image patch, then cell feature map is aggregated as $F_C(i,j)$ for $0 \le i \le [(M-1)/l]$ and $0 \le j \le [(N-1)/l]$.

By using the normalization and truncation operation, the 36-dimensional feature vectors are computed by

$$G_{\sigma,\rho}(i,j) = \left(\|F_C(i,j)\|^2 + \|F_C(i+\sigma,j)\|^2 + \|F_C(i,j+\rho)\|^2 + \|F_C(i+\sigma,j+\rho)\|^2\right)^{\frac{1}{2}} \tag{4}$$

$$H(i,j) = \begin{pmatrix} T_\zeta(F_C(i,j)/G_{-1,-1}(i,j)) \\ T_\zeta(F_C(i,j)/G_{+1,-1}(i,j)) \\ T_\zeta(F_C(i,j)/G_{+1,+1}(i,j)) \\ T_\zeta(F_C(i,j)/G_{-1,+1}(i,j)) \end{pmatrix} \tag{5}$$

where $G_{\sigma,\rho}$ is the normalization factors with $\sigma,\rho \in \{-1,1\}$, and $T_\zeta(\nu)$ denotes the component-wise truncation of vector ν by ζ.

The HOG feature map is calculated using only contrast insensitive gradient orientations in general, which makes it less discriminative and less robust to represent the target [10]. The fHOG feature can be used to overcome this drawback for the reason that the fHOG feature uses both contrast insensitive gradient orientations and contrast sensitive gradient orientations. The calculation process of the fHOG feature is as follows: the previous steps to calculate the fHOG feature are similar to HOG. The only difference is that, two cell-based feature maps (donated as $F_{insen}(i,j)$ and $F_{sen}(i,j)$) are calculated by aggregating the pixel-level feature map using contrast insensitive and contrast sensitive gradient orientations respectively in the fHOG feature calculation. However, the HOG feature calculates only one cell-based feature map $F_C(i,j)$ using the contrast insensitive gradient orientations. After the operation of normalization and truncation of $F_{insen}(i,j)$ and $F_{sen}(i,j)$, 108 dimensional feature vectors are obtained. Then an additional operation called analytic projection has been done over the 108 dimensional feature vectors to get the final 31-dimensional feature vectors consisting of 27 gradient orientations channels and 4 texture channels.

2.2. **Process of extracting the LBP feature.** The LBP feature is an effective descriptor of texture, and is widely used in classification due to its highly discriminative ability and invariance of gray scale and rotation. A gray scale and rotation invariant LBP feature is used in this paper [20]. To get a gray scale and rotation invariance, in a 3×3 neighbourhood texture image (see Figure 1), the gray values of c_1, c_3, c_5, c_7 are calculated by using the interpolation operation firstly. The center pixel value is calculated by

$$LBP_8 = \sum_{h=1}^{8} f(c_h - c_0)2^{h-1} \tag{6}$$

where c_0 is the center pixel of the original texture image; $c_h, h \in 1,2,\ldots,8$ is the neighbourhoods of the original texture image; the function f is defined as

$$f(c_h - c_0) = \begin{cases} 1 & \text{if } (c_h - c_0) \ge 0 \\ 0 & \text{else} \end{cases} \tag{7}$$

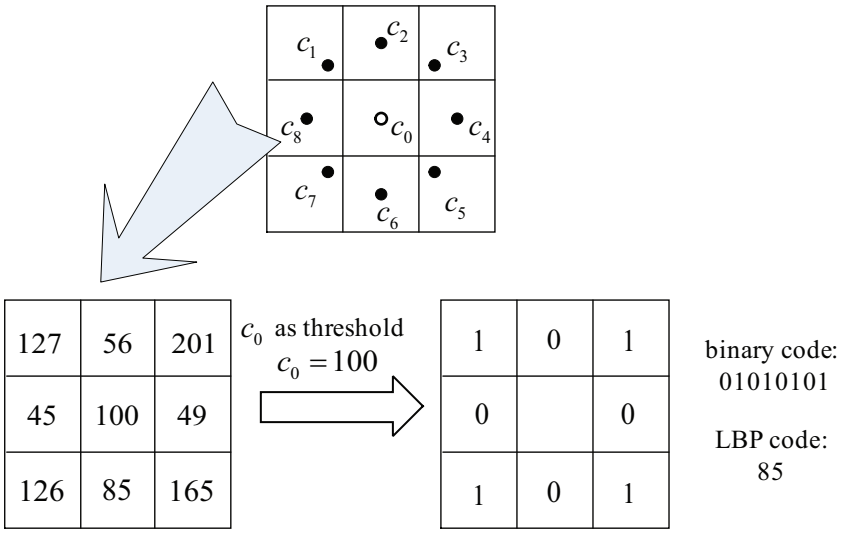

FIGURE 1. The 3×3 neighbourhood of a texture image

Secondly, to make sure that the rotation of a particular binary pattern will not affect the texture image after the operation in the first step, a rotation invariant LBP descriptor can be obtained by

$$LBP_8^{ri36} = \min\{ROR(LBP_8, h) \mid h = 0, 1, \ldots, 7\} \tag{8}$$

where $ROR(LBP_8, h)$ is a circular bit-wise right shift operator.

2.3. **Constructing a new feature.** Firstly, the fHOG feature is extracted from the target, and then a 31-dimensional feature is obtained. At each dimension, its number of columns is approximately the number of columns of the image divided by patch size, and the same as the number of rows. The third dimension spans the feature components. The LBP feature is extracted at the same time. To make sure that the two features are not differed from each other too much, we use the maximum value of fHOG to normalize the LBP feature, namely

$$LBP_{norm} = \frac{LBP - \min(LBP)}{\max(LBP) - \min(LBP)} \times \max(fHOG) \tag{9}$$

where LBP_{norm} is the normalized LBP feature, and the LBP feature LBP is achieved using a circular neighbourhood. To get the circular LBP, the pixels values are calculated using the following method: for the pixels that are just at the center of the grid, the pixels value of the square where the point is located, is taken as its value. For the neighbourhood points that are not in the center of the pixel, that value is determined by bilinear interpolation. To speed up the computation procedure, a mapping table for LBP codes is used in a neighbourhood of n sampling points to calculate the LBP feature. After extracting both the LBP and fHOG, a new feature is constructed by

$$f_{new} = fHOG + \mu \times LBP_{norm} \tag{10}$$

where μ is a coefficient to construct the new feature, and its value is decided by the experience.

The flow diagram of feature extraction is summarized in Figure 2. To speed up the process of LBP feature extraction, an LBP mapping table is pre-computed to map the gray scale value into a corresponding bin.

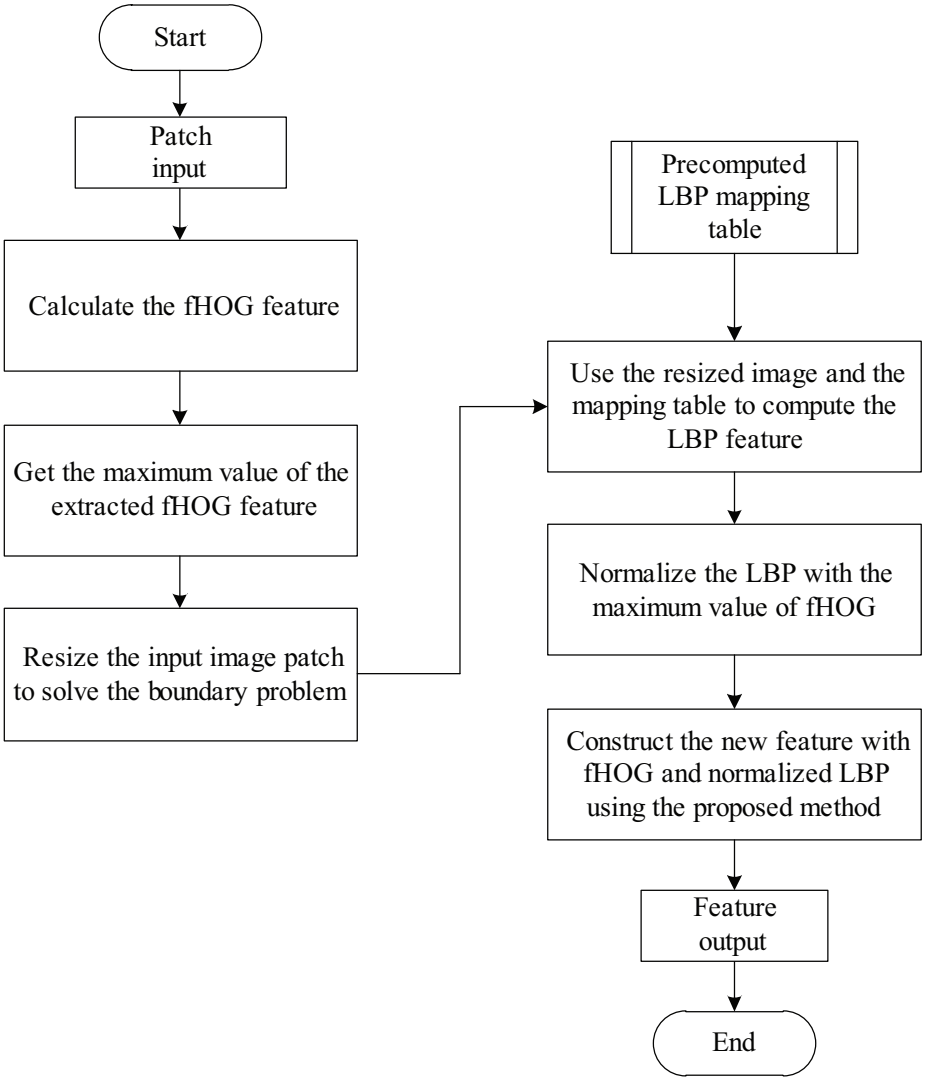

FIGURE 2. The flow diagram of the proposed method to extract the target feature, using an LBP mapping table, which can speed up the process of feature extraction

3. **Combining the New Feature Matrix with KCF.** Recently, trackers based on kernelized correlation filter have drawn great attention and also achieve excellent performance with high speed. In [11], Henriques et al. proved that using most useful kernels like radial basic function kernels, dot-product kernels and additive kernels can preserve the circulant structure, and speed up the tracker. The KCF tracker can be divided into four parts, which will be introduced as follows.

3.1. **Training a classifier.** In the t-th frame, KCF uses an $M \times N$ image patch which is centered around the target to train a classifier as

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle \tag{11}$$

where ϕ is an operation that maps the feature $\mathbf{x}$ into the Hilbert space, which is induced by the kernel κ by using the kernel trick, and the inner product of $\mathbf{x}$ and $\mathbf{x}'$ can be calculated as $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = \kappa(\mathbf{x}, \mathbf{x}')$. The parameter $\mathbf{w}$ is computed by minimizing the squared error over all the training samples $\mathbf{x}_i$ and their regression output $\mathbf{y}_i$, namely

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - \mathbf{y}_i \right)^2 + \lambda \|\mathbf{w}\|^2 \tag{12}$$

where the coefficient λ is used to control the over-fitting problem. In the nonlinear feature space, the solution can be re-written in the format of $\mathbf{w} = \sum_i^n \alpha_i \phi(\mathbf{x}_i)$. Different from the liner regression case, the variable in nonlinear regression that needs to be optimized is α_i, $i \in \{1, 2, \ldots, n\}$. The dual space parameter α can be trained as [11]:

$$\alpha = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\mathbf{k}^{\mathbf{xx}'}) + \lambda}\right) \tag{13}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are denoted the Fourier transform and the inverse Fourier transform, respectively. $\mathbf{k}^{\mathbf{xx}'}$ is the i-th element of the $\kappa(\mathbf{x}, \mathbf{x}')$, which can be calculated by

$$\mathbf{k}^{\mathbf{xx}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}')\right)\right) \tag{14}$$

where σ is the Gaussian kernel parameter, $\hat{\mathbf{x}}^*$ is the complex-conjugate of $\hat{\mathbf{x}}$, the $\odot$ operator is the element-wise product, and the $\wedge$ operator stands for the Fourier transform operation. When multi-channel features are used to represent the target, $\mathbf{k}^{\mathbf{xx}'}$ should be calculated by

$$\mathbf{k}^{\mathbf{xx}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}\left(\sum\nolimits_c \hat{\mathbf{x}}_c^* \odot \hat{\mathbf{x}}_c'\right)\right)\right) \tag{15}$$

where c stands for the channel number.

3.2. **Detecting a target.** After finishing the training procedure, the target at the $(t+1)$-th frame can be detected using the trained classifier, and the possible position can be found by

$$f(\mathbf{z}) = \mathcal{F}^{-1}\left(\left(\hat{\mathbf{k}}^{\mathbf{xz}}\right) \odot \hat{\alpha}\right) \tag{16}$$

The reason is that it would output a maximum value $f(\mathbf{z})_{\max}$, if the classifier correctly finds the target in the $M \times N$ testing input image, and z denotes the base image patch.

3.3. **Updating the tracker.** To detect the target at the following input frames continuously, there should be a mechanism to update the parameter α and the feature $\mathbf{x}$ extracted from the target, namely

$$\begin{cases} \hat{\alpha}_{t+1} = (1-\varphi)\,\hat{\alpha}_t + \varphi\hat{\alpha} \\ \hat{\mathbf{x}}_{t+1} = (1-\varphi)\,\hat{\mathbf{x}}_t + \varphi\hat{\mathbf{x}} \end{cases} \tag{17}$$

3.4. **Target representation.** To show the feature extraction in a more intuitive perspective, the sequence 'Freeman1' is used to explain the procedure of target representation, which is selected from the online object tracking benchmark [4]. The t-th frame of this sequence is shown in Figure 3(a). Firstly, the fHOG feature is extracted (see Figure 3(b)). Secondly, the rotation and gray scale invariant LBP feature is extracted (see Figure 3(c)). Based on the two features, the visible version of the new constructed feature can be obtained, which is shown in Figure 3(d).

By comparing Figure 3(b) and Figure 3(d), the shape and texture information of the picture is clearly described by using the new feature, which has the ability to handle the problem that the original fHOG feature cannot handle, such as the empty holes (see the red rectangles in Figure 3(b) and Figure 3(d)). Also, using the new feature, the structure of the target is more clear to represent.

To show the framework of the proposed method, the pseudo-code of the tracker is summarized and shown in Figure 4.

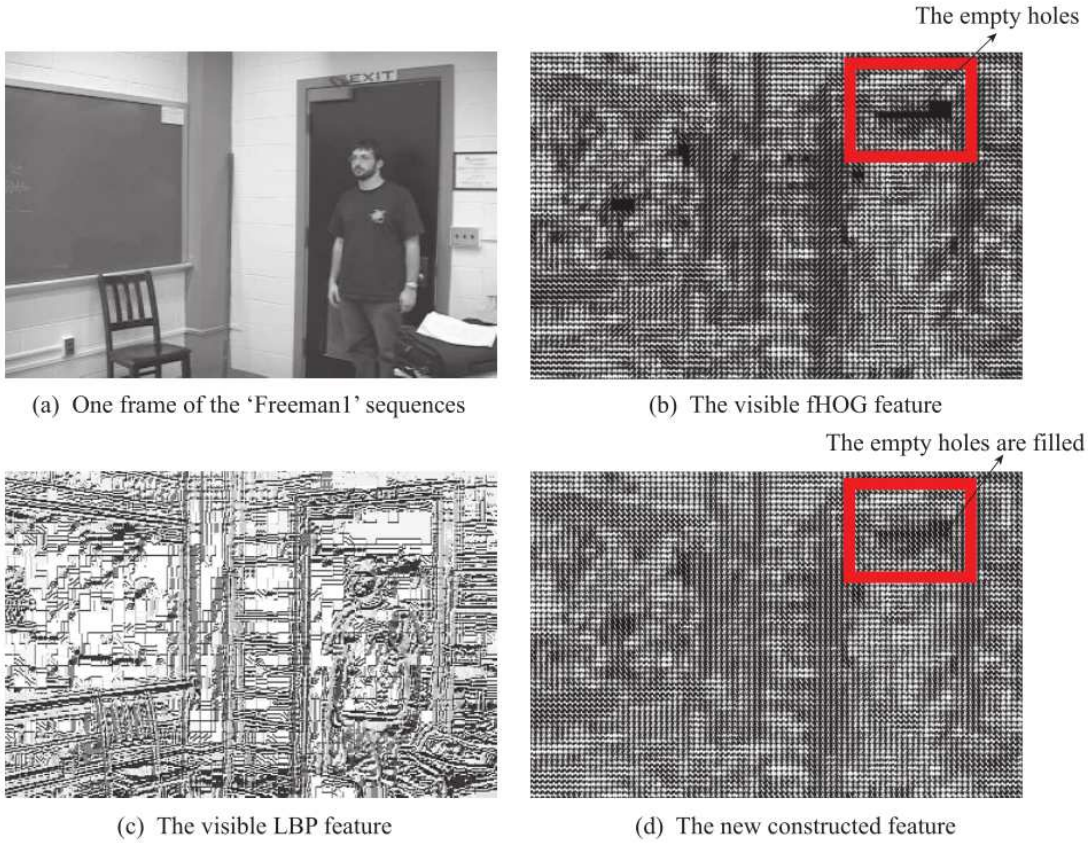

FIGURE 3. The visible extracted features and the constructed feature: (a) the t-th frame selected from the 'Freeman1' sequences; (b) the fHOG feature; (c) the LBP feature; (d) the visible new constructed feature

4. **Evaluation of the Proposed Method.** In this section, the proposed method is evaluated using ten sequences from the object tracking benchmark [4], and these ten sequences have the attributes of illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), and background clutter (BC). The ten sequences used in this paper are listed in Table 1, which are 'Bolt', 'Crossing', 'Dudek', 'Fleetface', 'Freeman1', 'Football', 'Girl', 'Jogging-1', 'Matrix' and 'Skiing'. The reason to choose these ten sequences is that they cover all the challenging factors. The configuration of the computer used to evaluate the method is Intel i5-4200, and 2.8GHz CPU with 8GB RAM. Most of the values of the parameters used in the tracking method are the same as those of [11] and are listed in Table 2.

To make it more persuasive, the proposed method (denoted by KCF_LBP) is compared with some state-of-the-art tracking methods. The experiment shows that, in the situation where the objects under tracking have rich texture feature, the proposed method can perform better than the MUlti-Store Tracker (MUSTer) [21], Scale Adaptive with Multiple Features tracker (SAMF) [13], Spatially Regularized Discriminative Correlation Filters tracker (SRDCF) [22], Discriminative Scale Space Tracking tracker (DSST) [23], KCF tracker [11], Tracking-Learning-Detection tracker (TLD) [24] and Structured Output Tracking with Kernels tracker (Struck) [25]. In this paper, two performance standards are chosen to evaluate the tracker: one is the precision, which shows the percentage of currently tracked frames for a range of distance thresholds and a higher precision at low

Algorithm 1: The framework of the KCF tracker

```
Load the initial parameters
Get the regression target output y
for frame = 1:N do
    if frame > 1 then
        get patch from frame with last predicted position pos_frame
        get feature z_frame from patch using the proposed method
        calculate response with α_frame+1 and z_frame and x_frame+1
        get new position pos_frame+1 from maximum response
        update position with pos_frame = pos_frame+1
    if frame ==1 then
        get patch from image with initial position pos
    else
        get patch from image with last predicted position pos_frame
    get feature x_frame from image patch using the proposed method
    train α_frame with y and x_frame
    if frame == 1 then
        set initial α_frame+1 and x_frame+1 with α_frame and x_frame
    else
        update α_frame+1 and x_frame+1 with α_frame and x_frame
```

FIGURE 4. The framework of KCF tracker using the new constructed feature

TABLE 1. The data sequences used to evaluate the proposed method

The sequences name	The attributes
Bolt	OCC, DEF, IPR, OPR
Crossing	SV, DEF, FM, OPR, BC
Dudek	SV, OCC, DEF, FM,IPR, OPR, OV, BC
Fleetface	SV, DEF, MB, FM, IPR, OPR
Football	OCC, IPR, OPR, BC
Freeman1	SV, IPR, OPR
Girl	SV, OCC, IPR, OPR
Jogging-1	OCC, DEF, OPR
Matrix	IV, SV, OCC, FM, IPR, OPR, BC
Skiing	IV, SV, DEF, IPR, OPR

TABLE 2. Some parameters we used in the paper

The parameters	The values	The parameter description
n	8	The neighborhoods of sampling points
r	1	The radius to sample
μ	10	The coefficient to construct the feature
ξ	1.5	The extra area surrounding the target
λ	1e-4	The regularization parameter
φ	0.02	The factor to update the model
σ	0.5	The Gaussian kernel bandwidth
l	4	The cell size of fHOG
p	9	The orientations of fHOG

thresholds means the more accurate results. The threshold is chosen at 20 pixels to represent precision score for comparing with different trackers. This usually shows as precision curves, which shows the average precision of the threshold in a certain range, and all tracking algorithms are ordered by the mean precision value of the Center Location Error (CLE) threshold at 20 pixels. The other one is the success plot rate, which expresses the percentage of successfully tracked sequences. The successfully tracked means that the overlap of the tracking bounding box is bigger than a set threshold value, which is usually set to 0.5. Pascal VOC Overlap Ratio (VOR), which is defined as

$$\mathrm{VOR} = \frac{Area\left(B_T \cap B_G\right)}{Area\left(B_T \cup B_G\right)} \tag{18}$$

where B_T is the tracking bounding box, and B_G is the ground truth bounding box; the operation $\cap$ stands for the overlap part of tracking bounding box and ground truth bounding box; the operation $\cup$ stands for the total coverage area of the two bounding box, and $Area\left(\cdot\right)$ stands for the area. The performances of trackers using the success plot are ranked by the Area Under the Curve (AUC). The better the tracker is, the larger the AUC is. The performances of the trackers are ranked at the right-top of the figure in both the precision graph and the success graph.

The results in Figure 5 show the precision graph and the success graph of the eight tracking methods over the benchmark. From Figure 5, we can draw a conclusion that our new feature based tracker performs better than other trackers. The precision of the One Pass Evaluation (OPE) of the proposed method at the 20 pixels is 0.766, and ranks first. In the success plots of OPE, the NO.1 method is the SAMF, with an AUC of 0.698, and the proposed method ranks the second with an AUC of 0.678. It means that, the proposed tracker works better than other trackers.

The tracking speed is a very important standard to evaluate a tracker, if a tracker can reach a speed of 25 Frames Per Second (FPS), we can say that this tracker can run in real time. The results in Table 3 show the tracking speed of different trackers. Due to the reason that the DSST and the SAMF use a technique to predict the scale change, and MUSTer uses a similar Atkinson-Shiffrin memory model to represent the target, and SRDCF uses a bigger padding to extract the spatial information, the speed of these trackers is slower than the original KCF tracker and the proposed method. Seen from Table 3, the proposed method is slower than the general KCF, but faster than the other trackers.

The results in Table 4 and Table 5 show the precision of the eight compared trackers over a specific sequence at the threshold of 20 pixels and the success rate of the eight

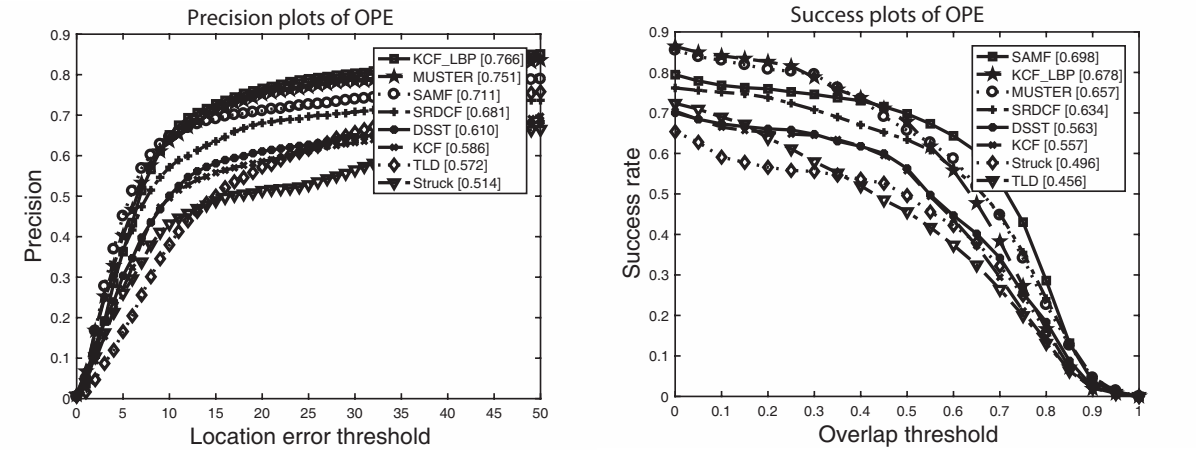

FIGURE 5. Quantitative comparison in OTB. The performance score for each tracker is shown in the legend (this is also the rank). For each figure, only the eight compared trackers are presented.

TABLE 3. Tracking speed of different trackers (FPS)

Sequences	Methods							
	KCF	DSST	SAMF	Struck	TLD	MUSTer	SRDCF	KCF_LBP
Bolt	220.53	35.51	13.48	18.50	43.87	2.45	4.18	81.42
Crossing	363.59	67.74	23.22	30.32	30.85	8.69	9.85	84.62
Dudek	71.19	4.81	12.10	15.60	8.79	2.04	3.26	33.57
Fleetface	91.69	3.57	15.14	16.34	11.27	2.32	3.30	42.47
Football	193.94	54.01	22.03	14.85	16.00	3.22	4.42	78.85
Rreeman1	445.52	89.84	44.78	17.53	30.23	9.78	7.49	141.04
Girl	225.95	56.17	26.70	22.19	22.97	13.14	6.37	85.28
Jogging-1	174.13	24.45	16.68	26.21	23.84	5.29	3.57	64.44
Matrix	209.99	52.70	12.67	16.01	15.86	1.89	5.25	81.19
Skiing	467.49	71.65	16.65	29.31	26.97	2.63	6.10	123.17

TABLE 4. The tracking precision of the eight compared trackers over the specific sequences (at 20 pixels)

Methods	Sequences									
	Bolt	Crossing	Dudek	Fleetface	Freeman1	Football	Girl	Jogging-1	Matrix	Skiing
DSST	1.000	1.000	0.828	0.620	0.383	0.798	0.928	0.231	0.180	0.136
KCF	0.989	1.000	0.877	0.460	0.393	0.796	0.864	0.235	0.170	0.074
SAMF	1.000	1.000	0.887	0.631	0.390	0.801	1.000	0.974	0.350	0.074
Struck	0.026	0.391	0.831	0.532	0.534	0.691	1.000	0.977	0.100	0.062
TLD	0.320	0.575	0.643	0.481	0.715	0.790	0.940	0.974	0.160	0.124
MUSTer	1.000	1.000	0.833	0.608	0.893	0.798	0.990	0.954	0.350	0.086
SRDCF	0.017	1.000	0.833	0.597	0.948	1.000	0.994	0.974	0.370	0.074
KCF_LBP	1.000	1.000	0.900	0.536	0.942	0.796	0.928	0.974	0.400	0.185

compared trackers over a specific sequences at a threshold of VOR = 0.5 respectively. From the result, we can see that the proposed tracker using a new feature performs better than some of the state-of-the-art trackers, and at some sequences, it is the best among all the trackers we used to compare.

TABLE 5. The success rate of the eight compared trackers over the specific sequences (at VOR = 0.5)

Methods	Sequences									
	Bolt	Crossing	Dudek	Fleetface	Freeman1	Football	Girl	Jogging-1	Matrix	Skiing
DSST	1.000	0.967	0.969	0.679	0.150	0.727	0.670	0.225	0.180	0.062
KCF	0.943	0.950	0.976	0.669	0.163	0.702	0.742	0.225	0.130	0.074
SAMF	0.997	1.000	0.982	0.703	0.282	0.677	1.000	0.967	0.320	0.049
Struck	0.020	0.392	0.962	0.675	0.203	0.674	0.964	0.909	0.100	0.062
TLD	0.177	0.458	0.670	0.441	0.233	0.749	0.726	0.964	0.070	0.074
MUSTer	1.000	1.000	0.765	0.645	0.623	0.627	0.580	0.948	0.330	0.049
SRDCF	0.014	1.000	0.992	0.663	0.626	0.879	0.776	0.971	0.370	0.049
KCF_LBP	0.911	0.925	0.982	0.728	0.199	0.790	0.828	0.967	0.330	0.124

Figure 6 shows the precision graphs of OPE at the effect of different challenging factors. The performances of the trackers are ranked at the legend at the right-top of the image, and they are ranked by the precision. We can see that in the attributes of deformation, in-plane rotation, illumination, out-of-plane rotation, scale variation, the proposed method ranks the first among the compared methods. While, in the attributes of occlusion, background clutter and fast motion, our tracker ranks the second; however, it is satisfied for the simple mechanism of feature reconstruction.

To show the tracking result apparently, some tracking results are shown in Figure 7. For all the tracking sequences, the bounding box marked with different colors stands for different tracking results of different methods, which is explained in the title of the figure in detail. As shown in Figure 7, in the sequences of 'Bolt', 'Freeman1', 'Girl', and 'Jogging-1', the target has a rich texture information, although they suffer from out-of-plane rotation, it means the target would rotate out of the image plane during tracking; however, with the new constructed feature consisting of LBP and fHOG, the proposed method can handle this challenging factor. The new constructed feature can not only capture the shape information of the target but also the texture information, although the target suffers from an out-of-plane rotation. In addition, the 'Girl' and the 'Jogging-1' sequences also suffer from the partial occlusion, with different textures between the target and the background, the proposed method can also handle this challenging factor, like other state-of-the-art methods, but the mechanism inside the proposed method is easier and understandable, making it more efficient and timesaving.

Remark 4.1. *In the motion of blur sequence 'Matrix' as shown at the last row of Figure 7, all the listed trackers fail to track the target, because of the target region that can blur due to the fast motion of the target or the camera, which is very difficult for the tracker to extract features, not only the shape and texture, but also the spatial information. So, this would be a future research direction of target tracking in the fast motion application.*

5. **Conclusions.** In this paper, a new kernelized correlation filter based tracker using a new feature is proposed. By using the fHOG feature and the LBP feature of the tracking object, a new feature is computed by using the proposed method to represent the target. The online object tracking benchmark is used to evaluate the method proposed in the paper. The experimental results show that, in the scenario that the target has rich texture features between the background, constructing the target texture and the shape information can make the target more robust to be represented. Using this new constructed feature matrix, the proposed method can achieve outstanding performance compared with some state-of-the-art methods and the original kernelized correlation filter

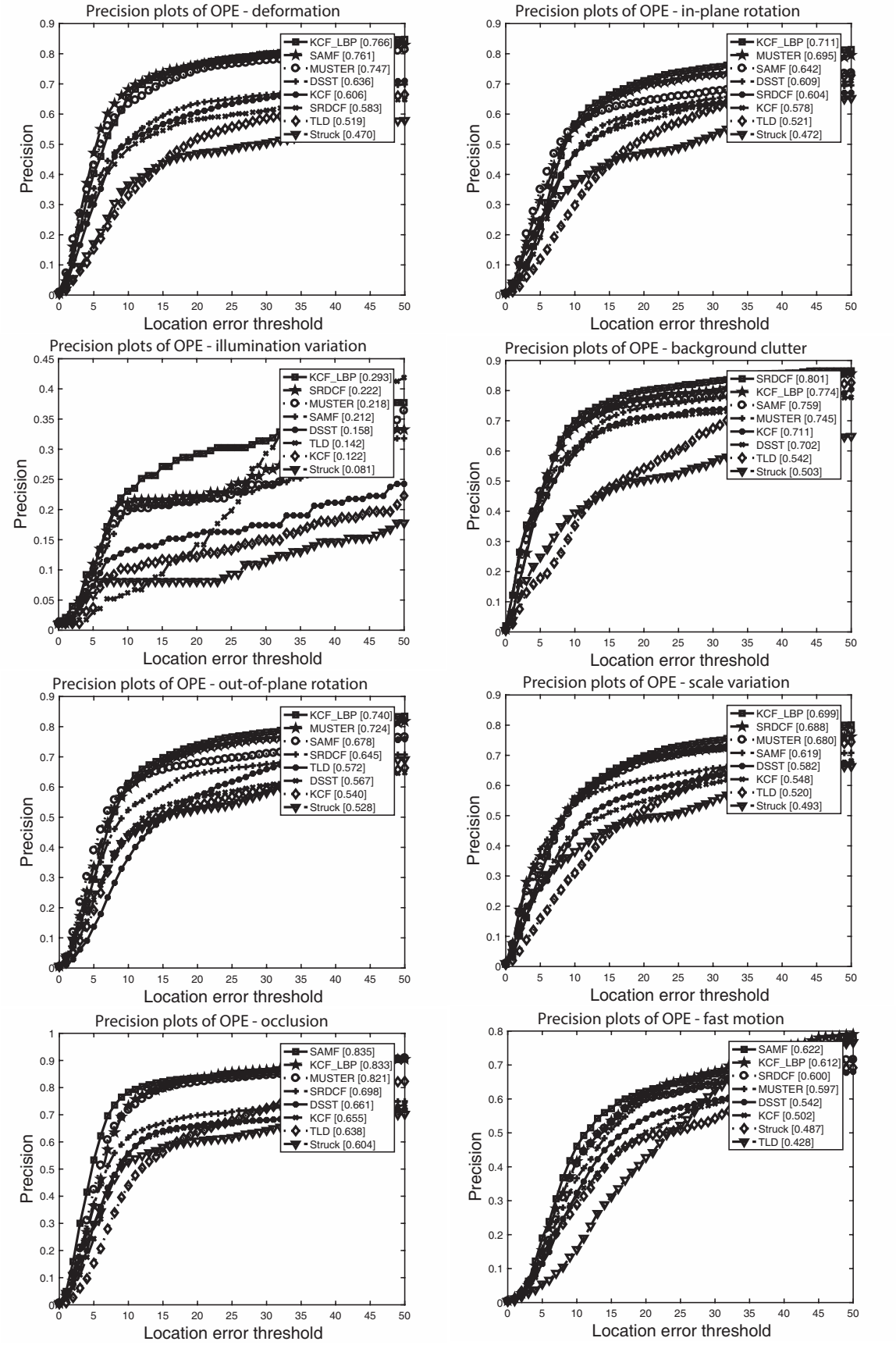

FIGURE 6. Attribute-based analysis of various approaches on the OTB dataset, where precision graphs are shown for eight attributes and only eight trackers for each attribute are used and displayed

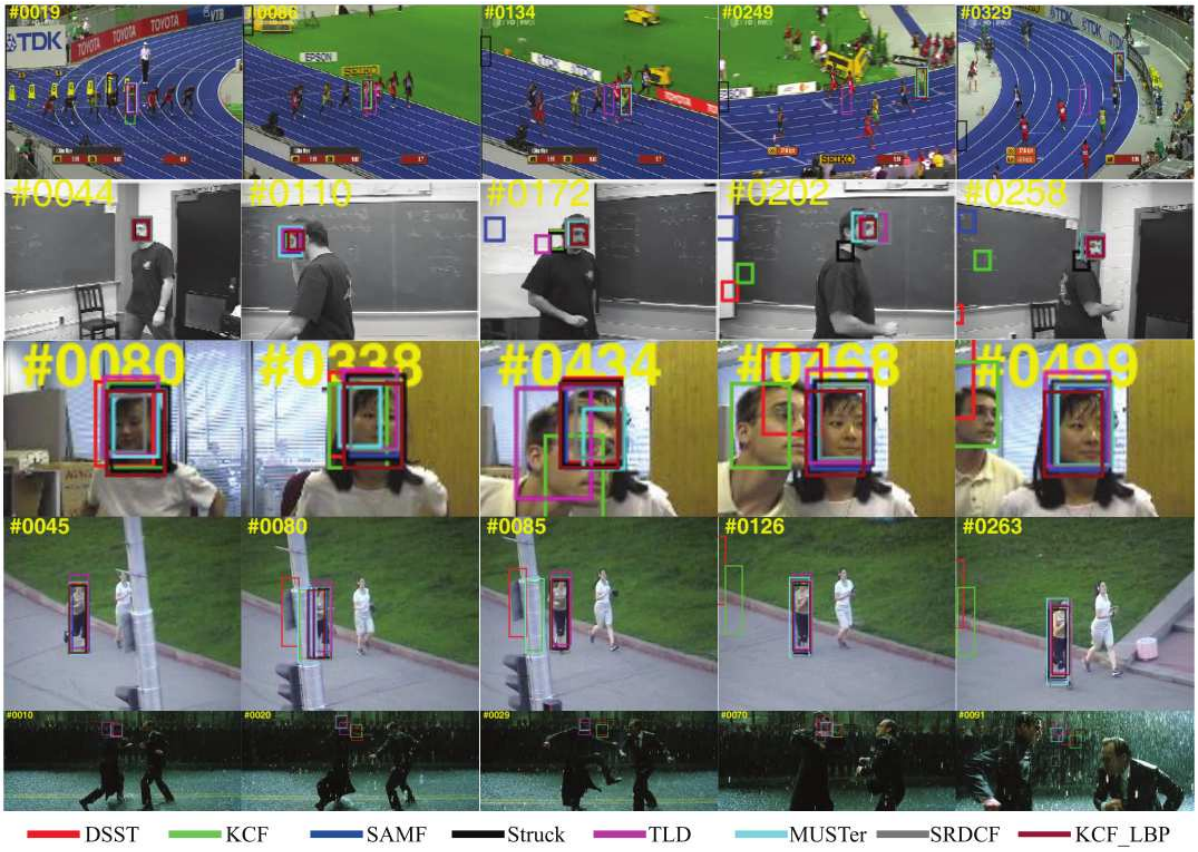

FIGURE 7. (color online) Tracking results of selected trackers in representative frames. The examples are from sequences 'Bolt', 'Freeman1', 'Girl', 'Jogging-1', and 'Matrix' (from top to down).

method. The future work will focus on a new tracking framework to consider the problem in the fast motion applications.

Acknowledgments. The authors would like to thank the National Natural Science Foundation of China (61203365, 61573128) and the Fundamental Research Funds for the Central Universities (2018B23214) for their support of this paper.

REFERENCES

[1] C.-C. Chiang, M.-C. Ho, H.-S. Liao, A. Pratama and W.-C. Syu, Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts, *International Journal of Innovative Computing, Information and Control*, vol.7, no.12, pp.6919-6934, 2011.

[2] D.-I. D. Cho and T.-J. Lee, A review of bioinspired vision sensors and their applications, *Sensors and Materials*, vol.27, no.6, pp.447-463, 2015.

[3] J. Ni, L. Yang, L. Wu and X. Fan, An improved spinal neural system-based approach for heterogeneous AUVs cooperative hunting, *International Journal of Fuzzy Systems*, vol.20, no.2, pp.672-686, 2018.

[4] Y. Wu, J. Lim and M. H. Yang, Object tracking benchmark, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.37, no.9, pp.1834-1848, 2015.

[5] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan and M. Shah, Visual tracking: An experimental survey, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.36, no.7, pp.1442-1468, 2014.

[6] Y. Xu, Y. Li, J. Wang, Z. Miao, H. Li and Y. Zhang, Feature adaptive correlation tracking, *IEICE Trans. Information and Systems*, vol.E100D, no.3, pp.594-597, 2017.

[7] S. Jeong, G. Kim and S. Lee, Effective visual tracking using multi-block and scale space based on kernelized correlation filters, *Sensors*, vol.17, no.3, 2017.
[8] S. Banerji, A. Sinha and C. Liu, New image descriptors based on color, texture, shape, and wavelets for object and scene image classification, *Neurocomputing*, vol.117, pp.173-185, 2013.
[9] W. Hu, W. Li, X. Zhang and S. Maybank, Single and multiple object tracking using a multi-feature joint sparse representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.37, no.4, pp.816-833, 2015.
[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, Object detection with discriminatively trained part-based models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.32, no.9, pp.1627-1645, 2010.
[11] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.37, no.3, pp.583-596, 2015.
[12] D. L. Cosmo, E. O. T. Salles and P. M. Ciarelli, Pedestrian detection utilizing gradient orientation histograms and color self similarities descriptors, *IEEE LATIN America Transactions*, vol.13, no.7, pp.2416-2422, 2015.
[13] Y. Li and J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, *Computer Vision – ECCV 2014 Workshops, PT II*, Berlin, Germany, vol.8926, pp.254-265, 2015.
[14] M. Danelljan, F. S. Khan, M. Felsberg and J. van de Weijer, Adaptive color attributes for real-time visual tracking, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1090-1097, 2014.
[15] D. Konstantinidis, T. Stathaki, V. Argyriou and N. Grammalidis, Building detection using enhanced HOG-LBP features and region refinement processes, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.10, no.3, pp.888-905, 2017.
[16] S. M. M. Ahsan, J. K. Tan, H. Kim and S. Ishikawa, Spatiotemporal LBP and shape feature for human activity representation and recognition, *International Journal of Innovative Computing, Information and Control*, vol.12, no.1, pp.1-13, 2016.
[17] C.-H. Zheng, W.-J. Pei, Q. Yan and Y.-W. Chong, Pedestrian detection based on gradient and texture feature integration, *Neurocomputing*, vol.228, no.SI, pp.71-78, 2017.
[18] J. Zhang, K. Huang, Y. Yu and T. Tan, Boosted local structured HOG-LBP for object localization, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1393-1400, 2011.
[19] X. Wang, T. X. Han and S. Yan, An HOG-LBP human detector with partial occlusion handling, *IEEE the 12th International Conference on Computer Vision (ICCV)*, pp.32-39, 2009.
[20] C. Zhu and R. Wang, Local multiple patterns based multiresolution gray-scale and rotation invariant texture classification, *Information Sciences*, vol.187, pp.93-108, 2012.
[21] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov and D. Tao, Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, pp.749-758, 2015.
[22] M. Danelljan, G. Hager, F. S. Khan and M. Felsberg, Learning spatially regularized correlation filters for visual tracking, *IEEE International Conference on Computer Vision (ICCV)*, New York, USA, pp.4310-4318, 2015.
[23] M. Danelljan, G. Häger, F. Khan and M. Felsberg, Accurate scale estimation for robust visual tracking, *Proc. of the British Machine Vision Conference*, 2014.
[24] Z. Kalal, K. Mikolajczyk and J. Matas, Tracking-learning-detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.34, no.7, pp.1409-1422, 2012.
[25] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks and P. H. S. Torr, Struck: Structured output tracking with kernels, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.38, no.10, pp.2096-2109, 2016.