IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 30, 2017, accepted June 11, 2017, date of publication June 30, 2017, date of current version July 24, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2720746

Multi-Channel Features Spatio-Temporal Context Learning for Visual Tracking

XIAOQIN ZHOU[1,2], XIAOFENG LIU[1,2], (Member, IEEE), CHENGUANG YANG[3], (Senior Member, IEEE), AIMIN JIANG[1,2], (Member, IEEE), AND BIN YAN[4], (Member, IEEE)

[1]Changzhou Key Laboratory of Robotics and Intelligent Technology, College of IoT Engineering, Hohai University, Changzhou 213022, China
[2]Jiangsu Key Laboratory of Special Robots, Hohai University, Changzhou 213022, China
[3]Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea SA1 8EN, U.K.
[4]College of Electronics, Communication and Physics, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Xiaofeng Liu (xfliu@hhu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61471157, in part by the Fundamental Research Funds for the Central Universities of China under Grant 2011B11114, Grant 2012B07314, and Grant 2015B38214, in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20141157 and Grant BK20141159, in part by the Natural Science Foundation of Shandong Province under Grant ZR2014JL044, and in part by the Open Foundation Programs of Changzhou Key Laboratory of Robotics and Intelligent Technology under Grant CZSR2014003.

ABSTRACT Visual tracking is a challenging issue in surveillance, human–computer Interaction, and intelligent robotics, among others. Managing appearance changes of the target object, illumination changes, rotations, non-rigid deformations, partial or full occlusions, background clutter, fast motion, and so forth is generally difficult. Among the numerous existing trackers, the correlation-filter-based tracker can achieve appealing performance with a fast speed for fast Fourier transform. Motivated by this property, the spatio-temporal context (STC) learning algorithm was proposed with the consideration of the information from the context around the target, and this algorithm achieved good results. However, STC only utilizes the overall intensity information. In this paper, we propose a multi-channel features STC learning algorithm with an improved scale-adaptive scheme. Our algorithm integrates powerful features, including Histogram of Oriented Gradients and color naming, using kernel methods on the basis of the STC algorithm to further enhance the overall tracking performance. Extensive experimental results obtained from various benchmark data sets demonstrate that the proposed tracker is promising for various challenging scenarios and maintains real-time performance at an average speed of 78 frames/s. According to the test results, our algorithm outperforms the STC algorithm and achieves performance that is competitive with the state-of-the-art algorithms.

INDEX TERMS Object tracking, kernel methods, correlation filters, fast Fourier transform.

I. INTRODUCTION

Visual tracking is one of the most active research topics due to its wide range of applications, such as motion analysis, activity recognition, surveillance, human-computer interaction and intelligent robotics. Designing a robust visual tracker is a formidable task due to a number of challenging factors, such as illumination changes, appearance changes, pose variations, non-rigid deformations, partial or full occlusions, background clutter, fast motion, and so forth [1], [2].

An object tracker generally consists of four modules: object description, observation model, motion model and model updating scheme. Recently, various types of features, such as HoG [13], [16]–[18] and color naming [14], [17], have also been utilized in object description. Numerous algorithms have been presented that focus on effective observation models, which can be categorized into generative [3]–[5], [10] and discriminative methods [6]–[9], [16]. Many motion models have been proposed to cover the complex motions of a target, such as particle filtering [4], Markov Chain Monte Carlo [10], dense sampling [16]–[19] and combinations of detection and tracking [8], [9].

Among these approaches, the discriminative correlation filter has already been applied in many applications. As described in the convolution theorem [11], the correlation in the time domain corresponds to an element-wise multiplication in the Fourier domain. Thus, the main idea of a correlation filter is that the correlation can be calculated in the Fourier domain, thereby avoiding the time-consuming convolution operation. Bolme *et al.* [11] and Henriques *et al.* [12] introduced the correlation filter into a tracking application.

The circulant structure tracker (CSK) [12] was proposed to explore the circulant structure patch to enhance the classifier by augmenting negative samples, which employs the kernel correlation filter to achieve high efficiency. Based on CSK [12], KCF [16] adopts the HoG feature [13] rather than raw pixels to improve the robustness of the tracker. To further enhance the performance of the CSK tracker, Danelljan et al. [14] adopted the color-naming feature in the object tracking task, which is a powerful feature for colored objects. The scale problem remains unresolved in the aforementioned methods. Although scale-adaptive variants, namely, SAMF [17] and DSST [18], have been proposed, they are not flexible enough due to pre-defined sampling behaviors. For example, these methods encounter difficulties with fast and abrupt scale changes.

The differences between STC and the other introduced correlation filter trackers include the following aspects [2]. First, STC is proposed to model the relationships between the object and its local spatial contexts, whereas common correlation filter trackers model the input appearance using trained filters. Second, the values of the confidence map in STC can be referred to as prior probabilities given the current object, whereas the values in the confidence maps of other correlation filter trackers are correlation scores [2]. Third, the STC algorithm has the ability to estimate scale variations, but the scale estimation is occasionally unstable.

The contributions of this paper are two-fold. First, we integrate powerful features, including HoG and color naming, using kernel methods on the basis of the STC algorithm. Second, we improve the scale-updating scheme to obtain high efficiency. Based on the benchmark protocol and dataset from [22], [23], an experiment is also conducted on the datasets with various challenging attributes. Our tracker reports a better accuracy while running efficiently at an average speed of 78 frames per second (fps).

The remainder of this paper is organized as follows. In section II, we briefly describe the building blocks of our algorithm, including the original STC algorithm, multiple features and kernel methods. Then, the novel MFSTC algorithm and some implementation details are developed in section III. In section IV, the experimental results and discussion are presented. Finally, we conclude the paper in section V.

II. BUILDING BLOCKS

In this section, we first review the STC tracker [19]; then, we introduce the powerful features utilized in our approach: HoG and color naming. To integrate multiple features into the tracker, we utilize kernel methods.

A. THE STC TRACKER

The STC [19] algorithm is an object tracking algorithm that translates the tracking task into a process of locating the object center by calculating a confidence map at every frame. This algorithm is designed to learn a likelihood distribution, which is defined as the prior possibility of locating an object

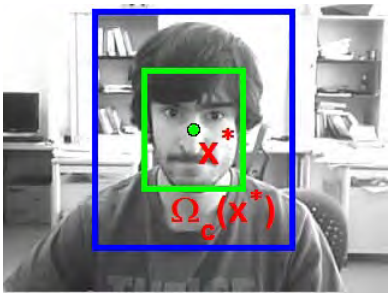


FIGURE 1. Graphical model of the spatial relationship between an object and its dense local context. The object is inside the green rectangle centered at the tracked result $\mathbf{x}^*$. The dense local context $\Omega_c(\mathbf{x}^*)$ is the region inside the blue rectangle, which includes the object region. The image is from a publicly available dataset [22]. http://cvlab.hanyang.ac.kr/tracker_benchmark/

in position $\mathbf{x}$ ($\mathbf{x} \in \mathbb{R}^2$):

$$\ell(\mathbf{x}) = P(\mathbf{x}|o) \tag{1}$$

where $\ell(\cdot)$ means likelihood and o is the object in the scene. Let $\mathbf{x}^*$ (refer to Figure 1) denote the position of the tracked object center, and let $\Omega_c(\mathbf{x}^*)$ denote the neighboring coordinates around $\mathbf{x}^*$. Then, a context feature set can be defined by $\mathbf{X}^c = \{\mathbf{v}(\mathbf{z}) = (\mathbf{I}(\mathbf{z}), \mathbf{z}) | \mathbf{z} \in \Omega_c(\mathbf{x}^*)\}$, where $\mathbf{I}(\mathbf{z})$ denotes the image intensity at location $\mathbf{z}$. By marginalizing the likelihood distribution of $\mathbf{v}(\mathbf{z})$ given o,

$$\begin{aligned} \ell(\mathbf{x}) &= P(\mathbf{x}|o) \\ &= \sum_{\mathbf{v}(\mathbf{z}) \in \mathbf{X^c}} \mathbf{P}(\mathbf{x}, \mathbf{v}(\mathbf{z})|o) \\ &= \sum_{\mathbf{v}(\mathbf{z}) \in \mathbf{X^c}} \mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o)\mathbf{P}(\mathbf{v}(\mathbf{z})|o) \end{aligned} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{z}$ are 2D location coordinates, o is the object, and the spatial context model $\mathbf{P}(\mathbf{x}, \mathbf{v}(\mathbf{z})|o)$ is the joint probability that models the spatial relationship between the object location and its context information. $\mathbf{P}(\mathbf{v}(\mathbf{z})|o)$ is the context prior probability, which models the appearance of the local context. Thus, the main task in the STC algorithm is to learn $\mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o)$ since it is the bridge between the object location and the spatial context.

To obtain the spatial context model $\mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o)$, we must first obtain the context prior model $\mathbf{P}(\mathbf{v}(\mathbf{z})|o)$. In [19], the context prior model was modeled as

$$\mathbf{P}(\mathbf{v}(\mathbf{z})|o) = \mathbf{I}(\mathbf{z})\omega_\sigma(\mathbf{z} - \mathbf{x}^*) \tag{3}$$

where $\mathbf{I}(\cdot)$ is the image intensity and $\omega_\sigma(\cdot)$ denotes a Gaussian-weighted function defined by

$$\omega_\sigma(\mathbf{z} - \mathbf{x}^*) = a \exp\left(-\frac{1}{\sigma^2}(\|\mathbf{z} - \mathbf{x}^*\|^2\right) \tag{4}$$

where a is a normalization constant and σ is a scale parameter. Since there is no direct expression of $\mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o)$, let us define a function to describe it:

$$\mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o) = h(\mathbf{x} - \mathbf{z}) \tag{5}$$

When the tracking object is marked, STC defines the confidence map of the object center as follows:

$$\ell(\mathbf{x}) = P(\mathbf{x}|o) = b\exp\left(-\left\|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right\|^{\beta}\right) \tag{6}$$

where b is also a normalization constant, α is a scale parameter, and β is a shape parameter. Subsequently, by combining equations (2), (5) and (3), we obtain the following:

$$\begin{aligned}\ell(\mathbf{x}) &= \sum_{\mathbf{v}(\mathbf{z})\in\mathbf{X^c}} \mathbf{P}(\mathbf{x}|\mathbf{v}(\mathbf{z}), o)\mathbf{P}(\mathbf{v}(\mathbf{z})|o) \\ &= \sum_{\mathbf{v}(\mathbf{z})\in\mathbf{X^c}} h(\mathbf{x}-\mathbf{z})\mathbf{I}(\mathbf{z})\omega_\sigma(\mathbf{z}-\mathbf{x}^*) \\ &= h(\mathbf{x}) \otimes \mathbf{I}(\mathbf{x})\omega_\sigma(\mathbf{x}-\mathbf{x}^*)\end{aligned} \tag{7}$$

where $\otimes$ denotes the convolution operator. Because the Fourier transform of a convolution equals the pixel-wise product of a Fourier transform, we can transform equation (7) into the frequency domain, where the fast Fourier transform (FFT) can be used for fast calculations, i.e.,

$$\begin{aligned}\mathcal{F}(\ell(\mathbf{x})) &= \mathcal{F}(h(\mathbf{x})) \odot \mathcal{F}(\mathbf{I}(\mathbf{x})\omega_\sigma(\mathbf{x}-\mathbf{x}^*)) \\ &= \mathcal{F}\left(b\exp\left(-\left\|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right\|^{\beta}\right)\right)\end{aligned} \tag{8}$$

where $\mathcal{F}$ is the FFT operation and $\odot$ is the pixel-wise product; thus, we can obtain

$$h(\mathbf{x}) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(b\exp\left(-\left\|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right\|^{\beta}\right)\right)}{\mathcal{F}(\mathbf{I}(\mathbf{x})\omega_\sigma(\mathbf{x}-\mathbf{x}^*))}\right) \tag{9}$$

where $\mathcal{F}^{-1}$ denotes the inverse FFT function and division is performed element-wise. The spatial context model $h(\mathbf{x})$ learns the relative spatial relations between different pixels in a Bayesian framework.

After we obtain $h(\mathbf{x})$, the tracking task becomes a detection problem. We assume that we have updated the spatial context model $H_t(\mathbf{x})$ in the $t-th$ frame; then, the object center $\mathbf{x}^*_{t+1}$ in the $(t+1)-th$ frame with the maximum value in $\ell_{t+1}(\mathbf{x})$ can be viewed as the new position of the object. $\ell_{t+1}(\mathbf{x})$ of the new frame can be calculated as follows:

$$\ell_{t+1}(\mathbf{x}) = \mathcal{F}^{-1}(\mathcal{F}(H_t(\mathbf{x})) \odot \mathcal{F}(\mathbf{I}_{t+1}(\mathbf{x})\omega_\sigma(\mathbf{x}-(\mathbf{x}^*_t)))) \tag{10}$$

where $H_t(\mathbf{x})$ is updated online as follows:

$$H_t(\mathbf{x}) = (1-\rho)H_{t-1}(\mathbf{x}) + \rho h_t(\mathbf{x}) \tag{11}$$

where ρ is the learning rate.

B. MULTIPLE FEATURES

HoG is one of the most popular visual features in the field of computer vision since it is very effective in practical applications. The HoG feature extracts the gradient information from a cell, which is a region of pixels. HoG counts the discrete orientations to form the histogram. As in [13], we adopt the 31 gradient orientation bins variant in our approach [16], [17].

In addition to HoG, color naming or color attributes are also believed to be beneficial [14]. Being better than the RGB space, the distance in color name space is more similar to human perception. We transform the RGB space into the color name space, which is an 11-dimensional color representation that includes black, blue, brown and so on. Color names provide the perception of object color with unit length, which typically contains the important information of the target.

C. KERNEL METHODS

Henriques *et al.* [16] proposed the use of the kernel trick to extend correlation filters for very fast tracking. The main reason for its prominent speed is that the tracker exploits the circulant structure that appears from the periodic assumption of the local image patch. A classifier is trained using image patch x of size $M \times N$ that is centered around the object. The tracker considers all cyclic shifts $x_{m,n}$, $(m,n) \in \{0,\ldots,M-1\} \times \{0,\ldots,N-1\}$, as the training examples for the classifier. These are labeled with a Gaussian function y; thus, $y_{m,n}$ is the label for $x_{m,n}$. The classifier is trained by minimizing the cost function over $\mathbf{w}$.

$$\epsilon = \sum_{m,n} \left|\left\langle\varphi(x_{m,n}), \mathbf{w}\right\rangle - y_{m,n}\right|^2 + \lambda\left\langle\mathbf{w},\mathbf{w}\right\rangle \tag{12}$$

where φ is the mapping to the Hilbert space induced by the kernel κ, and the inner product is defined as $\langle\varphi(f),\varphi(g)\rangle = \kappa(f,g)$. The constant λ is a regularization parameter. The cost function is minimized by $\mathbf{w} = \sum_{m,n}\alpha_{m,n}\varphi(x_{m,n})$, where the coefficients α are

$$\mathcal{F}(\alpha) = \hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{x}'\mathbf{x}'} + \lambda} \tag{13}$$

where $\mathcal{F}$ is the DFT (discrete Fourier transform) operator. $\mathbf{k}^{\mathbf{x}'\mathbf{x}'}$ is defined as *kernel correlation* in [16]. We choose the Gaussian RBF kernel, which can be applied to the circulant matrix trick as follows:

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

The detection step is conducted by previously cropping out patch z of size $M \times N$ in the new frame. Patch z is treated as the base sample to calculate the response in the Fourier domain,

$$\hat{\mathbf{f}}(\mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{x}'\mathbf{z}} \odot \hat{\boldsymbol{\alpha}} \tag{15}$$

where matrix $\mathbf{f}$ contains the output of model $f()$ for all cyclic shifts of $\mathbf{z}$. Intuitively, evaluating $f(\mathbf{z})$ can be viewed as a spatial filtering operation over the kernel values $\mathbf{k}^{\mathbf{x}'\mathbf{z}}$. Each $f(\mathbf{z})$ is a linear combination of the neighboring kernel values from $\mathbf{k}^{\mathbf{x}'\mathbf{z}}$ weighted by the learned coefficients $\boldsymbol{\alpha}$. The location of the maximum element in $\mathbf{f}$ corresponds to the cyclic shift of $\mathbf{z}$ most similar to the current target appearance $\mathbf{x}'$. For more details regarding the formulation, please refer to [12] and [16].

III. PROPOSED MULTIPLE-SCALE MFSTC TRACKER

The original STC algorithm adopts an update scheme by considering all previous frames when calculating the current model. However, this scheme is only applied to one-dimensional features and linear kernels. To integrate multiple features, we propose a multiple-scale MFSTC tracker. The block diagram of the proposed MFSTC tracker is presented in Figure 2. The multi-dimensional features integration and target position update are described in section III(A). In section III(B), we introduce an improved scale update strategy.

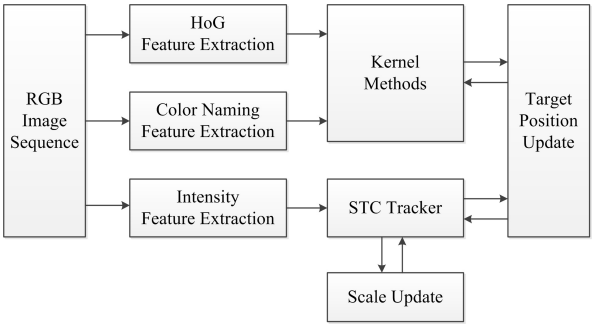


FIGURE 2. Block diagram of the proposed MFSTC tracker.

A. INTEGRATION OF MULTIPLE FEATURES

The HoG feature is primarily applied for analyzing the image gradients, whereas the color-naming feature focuses on color representations. These two features are complementary to each other. Based on the efficiency of integrating multi-channel data, both the HoG feature and color-naming feature can be fused together to promote robust tracking [17].

Since the kernel correlation function only needs to calculate the dot product and vector norm, multiple channels can be applied for the image features. Assume that the multiple channels of the data representation are concatenated into a vector $\mathbf{x}' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_c]$. Equation (14) can be rewritten as follows:

$$\mathbf{k}^{\mathbf{x}'\mathbf{x}''} = \exp\left(-\frac{1}{\sigma^2}(\|\mathbf{x}'\|^2 + \|\mathbf{x}''\|^2 - 2\mathcal{F}^{-1}(\sum_c \hat{\mathbf{x}}'^*_c \odot \hat{\mathbf{x}}''_c))\right) \tag{16}$$

which allows us to employ the stronger features rather than the raw gray pixels. There are three types of features applied in our proposed tracker. In addition to the raw gray pixel of the original image, we adopt two features that are commonly used in visual tasks: HoG and color naming.

The final location of the tracked object is determined by two response values: one from the result of equation (10) and the other from the output of equation (15) containing multiple-feature-based kernel methods. We set the weights for both response values. In practice, we average the weights to arrive at the final tracked result, which is a common technique for online learning.

B. UPDATE OF SCALE

STC has its own scheme for managing scale variations. Suppose that the new estimated center of the object is $\mathbf{x}^*$ and that $\ell(\mathbf{x}^*)$ is its calculated confidence score. Then, the scales can be estimated as follows:

$$s'_t = \sqrt{\frac{\ell_t(\mathbf{x}^*_t)}{\ell_{t-1}(\mathbf{x}^*_{t-1})}} \tag{17}$$

where s'_t is the predicted scale at time t. To smooth the predictions, the estimated scales are averaged over n consecutive frames, and linear interpolation is utilized for prediction:

$$\begin{cases} \bar{s}_t = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} s'_{t-i} \\ s_{t+1} = (1-\lambda)s_t + \lambda \bar{s}_t \end{cases} \tag{18}$$

where λ is a fixed parameter. With the estimated size of the object, the parameter σ of the weight function in equation (4) is also required to be updated:

$$\sigma_{t+1} = s_t \sigma_t \tag{19}$$

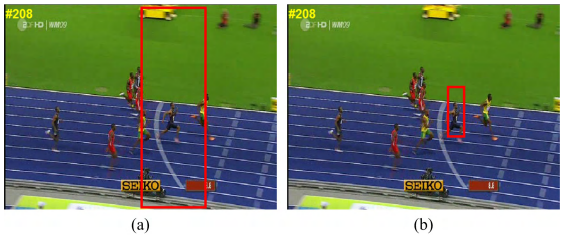
When conducting the experiments, we found that the estimation in STC may occasionally be unstable because the calculation can be extremely large if the denominator of equation (17) is close to zero. For this case, we have proposed an improved scale update strategy in our paper. We first set s'_t to 1 in equation (17). Then we introduce a penalty term $p(s)$ in equation (20) because the scale variation between two consecutive frames is continuous and small, which prevents an abrupt change in scale factor s.

$$p(s) = \begin{cases} -\log_2(s), & \left|\log_2(s)\right| \leqslant d \quad (20a) \\ d, & \log_2(s) < -d \quad (20b) \\ -d, & \log_2(s) > d \quad (20c) \end{cases}$$

Thus, the updated scale value is formulated as follows:

$$s'_{t+1} = s_{t+1} + p(s_{t+1}) \tag{21}$$

Figure 3 shows the scale comparison between the original and the improved scale update strategy on the *Bolt* sequence.



(a) (b)

FIGURE 3. A visual comparison of the original and improved scale update schemes in the STC tracker: (a) original scale update, (b) improved scale update.

TABLE 1. Average P20 (%) and average frames per second (FPS). The total number of evaluated frames is 8,100.

Sequence	CT	FCT	TLD	IVT	DFT	CXT	CSK	KCF	DSST	SAMF	STC	CN	MFSTC
mhyang	81.9	56.1	97.8	**100.0**	76.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	96.2	**100.0**
shaking	4.7	37.5	40.5	1.1	83.0	12.6	56.4	2.5	**99.7**	2.7	97.8	70.7	**99.7**
singer2	0.5	22.1	7.1	3.6	59.6	6.3	3.6	94.8	**99.7**	3.6	57.1	3.6	92.6
coke	11.3	12.4	68.4	13.1	8.6	65.3	87.3	83.8	91.8	**93.5**	15.5	61.5	85.9
crossing	**100.0**	**100.0**	61.7	**100.0**	68.3	62.5	**100.0**	**100.0**	**100.0**	**100.0**	53.3	**100.0**	**100.0**
girl	60.8	66.8	91.8	44.4	29.6	76.8	55.4	86.4	92.8	**100.0**	59.4	86.4	**100.0**
walking	**100.0**	**100.0**	96.4	**100.0**	**100.0**	23.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
dog1	95.0	88.1	**100.0**	98.0	62.4	**100.0**	**100.0**	**100.0**	**100.0**	99.9	70.0	**100.0**	**100.0**
mountainBike	17.5	67.5	25.9	99.6	35.1	28.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
faceocc2	68.1	90.8	85.6	99.3	**100.0**	**100.0**	**100.0**	97.2	**100.0**	90.8	97.4	62.3	88.4
football	79.8	80.1	80.4	79.3	**84.3**	79.6	79.8	79.6	79.8	79.6	80.1	79.8	79.8
Coupon	61.5	26.3	39.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
walking_occ_long	4.5	4.5	30.7	5.5	4.0	7.0	7.5	6.0	21.1	**93.0**	87.9	6.5	91.0
wangyong	73.3	98.7	49.0	33.7	20.7	8.0	74.3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
pigeon_rgb	25.2	2.6	8.5	2.6	28.4	6.2	5.0	2.1	2.9	2.1	1.8	**28.2**	2.6
pigeon_depth	10.9	19.6	3.5	1.2	1.5	1.2	5.3	21.1	2.1	19.6	13.8	6.5	**25.5**
mouse_black2	96.3	93.9	51.7	72.0	32.8	12.8	**100.0**	**100.0**	**100.0**	**100.0**	99.7	**100.0**	**100.0**
Average P20	52.4	56.9	55.2	56.1	52.6	46.5	69.1	74.9	81.8	75.6	72.6	70.7	**86.2**
Average fps	44	130	29	28	10	15	204	160	35	16	**475**	157	78

IV. EXPERIMENTS AND RESULTS

We evaluate the proposed MFSTC tracking algorithm using 17 video sequences with many challenging attributes, including drastic illumination changes, heavy occlusions, pose and scale variations, rotations, non-rigid deformations, and background clutter. Among these 17 video sequences, 12 sequences (*mhyang*, *shaking*, *singer*2, *coke*, *crossing*, *girl*, *walking*, *dog*1, *mountainBike*, *faceocc*2, *football*, and *Coupon*) are from benchmark [22], 1 sequence (*walking_occ_long*) is from benchmark [23], and 4 sequences (*wangyong*, *pigeon_rgb*, *pigeon_depth*, and *mouse_black*2) are from our own test videos. We compare the proposed MFSTC tracker with 12 state-of-the-art methods in which the FFT-based trackers [12], [14], [16]–[19] are included. We compare with the following 12 trackers: compressive tracker (CT) [6], fast compressive tracker (FCT) [7], tracking-learning-detection (TLD) method [9], incremental visual tracking (IVT) method [5], distribution field tracker (DFT) [21], CXT [15], circulant structure tracker (CSK) [12], kernelized correlation filters (KCF) [16], DSST [18], SAMF [17], STC [19] and CN [14]. All the experiments are implemented in MATLAB, and our tracker runs at 78 fps on an i5-4460S CPU (2.90 GHz) PC with 8.0 GB of memory.

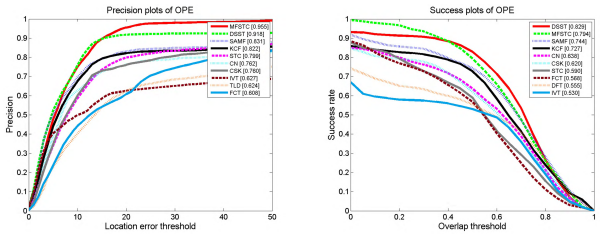


FIGURE 4. Average precision plot (left) and average success plot (right) on all our test sequences (only the top 10 trackers are presented for clarity).

TABLE 2. Average AUC (%). The total number of evaluated frames is 8,100.

Sequence	CT	FCT	TLD	IVT	DFT	CXT	CSK	KCF	DSST	SAMF	STC	CN	MFSTC
mhyang	73.0	50.2	89.3	**100.0**	77.5	**100.0**	**100.0**	**100.0**	99.9	**100.0**	86.0	91.7	**100.0**
shaking	4.1	34.0	40.0	1.1	82.5	10.7	58.1	1.4	**100.0**	1.4	83.6	67.4	92.1
singer2	1.1	39.1	10.1	3.8	69.7	3.8	3.6	97.0	**100.0**	3.6	45.9	3.6	95.4
coke	9.3	11.7	28.9	13.1	8.6	59.1	73.9	72.2	**86.3**	83.2	8.9	47.8	57.4
crossing	98.3	97.5	51.7	24.2	64.2	34.2	31.7	92.5	97.5	**100.0**	17.5	96.7	99.2
girl	17.8	31.8	76.4	18.6	25.2	64.2	39.8	75.6	62.8	91.6	30.2	46.2	**95.8**
walking	50.2	54.6	38.3	**99.8**	55.1	21.8	51.9	51.5	54.9	**99.8**	72.1	45.9	45.6
dog1	65.2	63.9	67.3	86.0	52.1	**99.8**	65.3	65.3	64.6	82.7	57.3	65.3	64.8
mountainBike	17.1	66.2	25.9	98.2	35.1	28.1	**100.0**	98.7	**100.0**	96.9	86.8	**100.0**	99.1
faceocc2	74.4	91.3	82.9	91.4	99.5	94.6	**100.0**	99.6	99.9	98.4	98.0	62.4	83.7
football	78.5	75.7	41.2	71.5	**84.3**	65.2	65.7	68.2	72.7	77.3	61.9	65.7	64.4
Coupon	88.7	93.6	38.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
walking_occ_long	4.5	4.5	11.1	5.0	3.5	11.1	7.5	5.5	23.1	16.6	**27.6**	6.5	22.1
wangyong	33.0	81.7	45.0	29.7	19.3	1.0	70.3	91.0	**99.0**	89.7	50.3	93.7	92.0
pigeon_rgb	26.4	2.3	6.7	1.8	29.0	13.5	12.9	6.5	7.3	2.1	1.5	**29.3**	2.9
pigeon_depth	23.8	26.4	1.5	1.5	2.9	1.8	6.5	24.0	1.2	27.6	15.0	11.1	**50.7**
mouse_black2	54.1	32.8	24.7	52.4	29.4	5.1	88.2	**97.3**	88.5	91.6	67.2	89.5	93.6
Average AUC	42.3	50.4	40.0	46.9	49.3	42.0	57.4	67.4	**74.0**	68.4	53.5	60.2	**74.0**

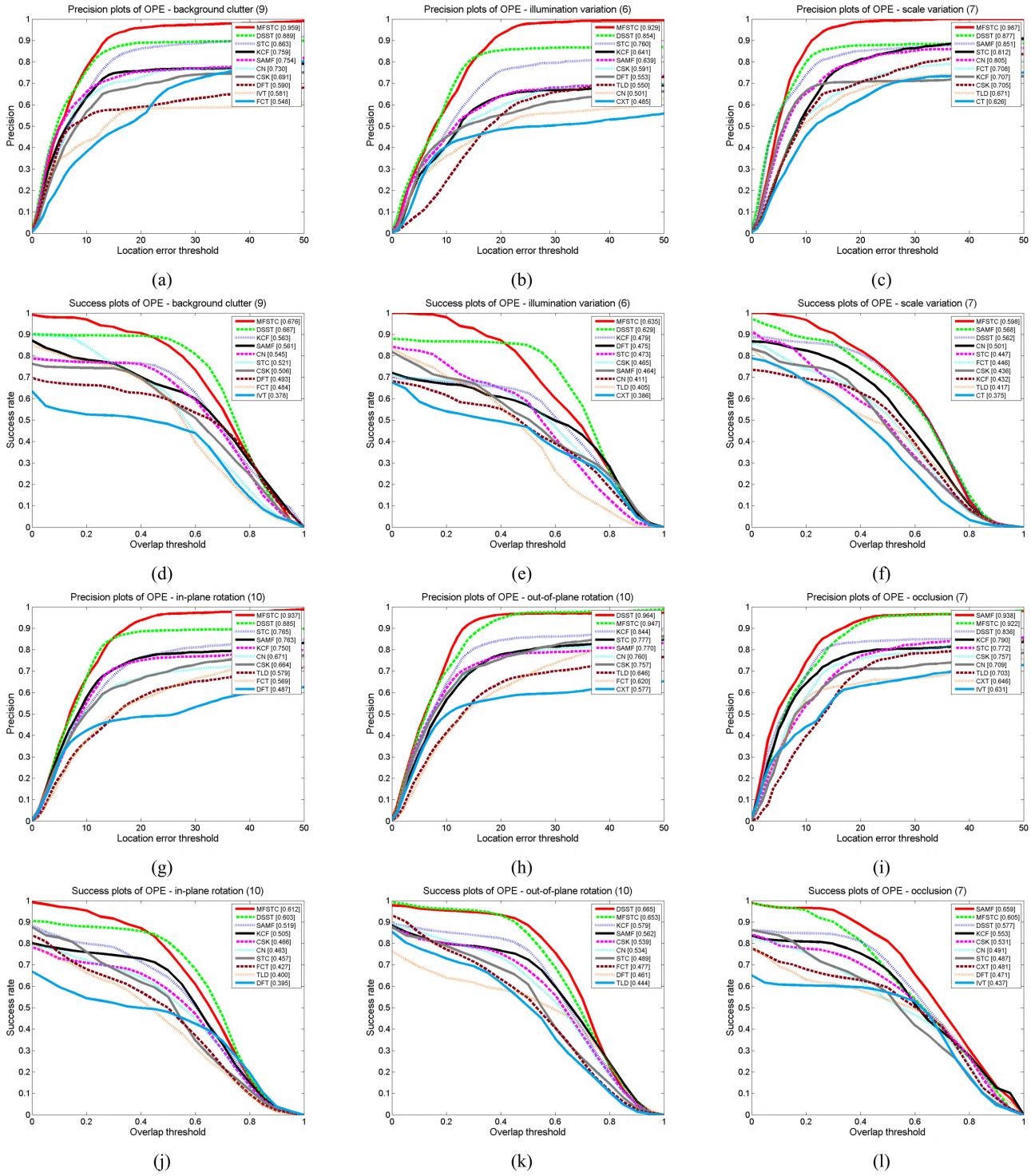


FIGURE 5. **The plot curves for the proposed tracker MFSTC compared with the other state-of-the-art trackers on the datasets (only the top 10 trackers are presented for clarity).**

A. EXPERIMENTAL SETUP

The parameters of the proposed algorithm are fixed for all the experiments. The size of the context region is set to be twice the size of the target object [19]. The parameter σ_t of equation (4) is initially set to $\sigma_1 = \frac{r_h + r_w}{2}$, where r_h and r_w are the height and width of the initial tracking rectangle, respectively. The parameters of the map function in equation (6) are set to $\alpha = 2.25$ and $\beta = 1$. The learning parameter is $\rho = 0.075$ in equation (11). The scale parameter s_t is initialized to $s_1 = 1$, and the filter parameter is $\lambda = 0.25$ in equation (18). In the kernel methods, σ used in the Gaussian function in equation (14) is set to 0.5. The cell size of HoG is

FIGURE 6. A visual comparison of our tracker with five state-of-the-art correlation filter trackers.

4×4, and the orientation bin number of HoG is 9. The penalty constant d in equation (20) is set to 0.02 empirically.

B. EXPERIMENTAL RESULTS

We provide two types of plots, precision and success plots [22], to evaluate the 13 trackers. Precision plots are obtained by computing the percentage of frames from which the location error is below a certain threshold. In TABLE 1, we select the threshold equal to 20 pixels (P20), as proposed in [22]. Success plots measure the bounding box overlap between the tracked object and the ground-truth. These plots provide the percentage of successful frames where the overlap is larger than a threshold as it is varied from 0 to 1. In TABLE 2, we select the threshold equal to 0.5, as proposed in [19]. Figure 4 shows the average precision plot (left) and the average success plot (right) on all 17 of our test sequences. Our proposed MFSTC tracker obtains the best precision results on average in Figure 4 (left) and the second best success rate (AUC) on average in Figure 4 (right).

The total number of evaluated frames is 8,100. The proposed MFSTC tracker obtains better performance both in terms of P20 of 86.2% and AUC of 74.0%, as shown in TABLE 1 and TABLE 2.

Figure 5 shows the detailed report of MFSTC compared with the other trackers: CT [6], FCT [7], TLD [9], IVT [5], DFT [21], CXT [15], CSK [12], KCF [16], DSST [18], SAMF [17], STC [19] and CN [14]. Although MFSTC is not specifically designed for background clutter and illumination and scale variations, amazingly, the proposed tracker achieved appealing performances on these challenging video sequences (refer to Figure 5(a)-(f)). These promising results suggest that the effective features are more effective than the complicated models for background clutter and illumination and scale variations.

An intuitive visual comparison on four very challenging sequences is presented in Figure 6, which shows that our tracker can preferentially track the object. Six correlation filter trackers are included in the comparison, namely, STC [19], CN [14], DSST [18], KCF [16], SAMF [17] and our proposed MFSTC, as shown in Figure 6.

1) BACKGROUND CLUTTER AND ILLUMINATION VARIATIONS

In the *shaking* sequence, as shown in Figure 6(a), the texture in the background is very similar to that of the target. The KCF and SAMF trackers drift to the background, whereas

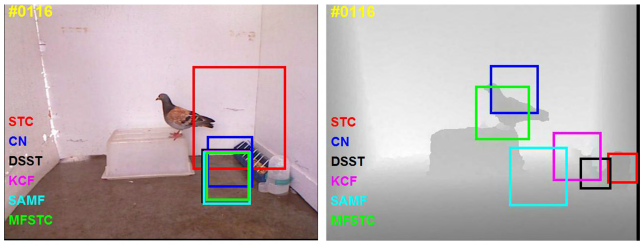
our proposed MFSTC algorithm achieves a better tracking result. There are large illumination variations and background clutter in the *singer*2 sequence, as shown in Figure 6(b). This is a dynamic scenario. The STC tracker can track at the ♯75, ♯181, ♯208 frames, but it also drifts to the background at the ♯298, ♯349 frames. The DSST and MFSTC methods provide stable tracking in the *shaking* and *singer*2 sequences.

2) POSE AND SCALE VARIATIONS

The objects in the *shaking*, *singer*2, and *walking_occ_long* sequences also undergo gradual pose and scale variations in Figure 6(a, b, d), which make the tracking tasks difficult. Our proposed algorithm is able to successfully track the objects in most frames of these sequences.

3) INTERMITTENT OCCLUSIONS

The target in the *coke* sequence is partially occluded at times (refer to ♯42 in Figure 6(c)). The *walking_occ_long* sequence has heavy occlusion (see ♯15, ♯51 in Figure 6(d)). The KCF, CN and DSST trackers fail to successfully track the object. Our MFSTC algorithm has a better adaptation than the original STC method (see ♯168, ♯187 in Figure 6(d)).



(a) (b)

FIGURE 7. A visual comparison of our tracker with five state-of-the-art correlation filter trackers on the *pigeon_rgb* and *pigeon_depth* sequences. (a) ♯116 color frame of the *pigeon_rgb* sequence. (b)♯116 depth frame of the *pigeon_depth* sequence corresponding to (a).

On our test sequences, the values of P20 and AUC in the *pigeon_rgb* and *pigeon_depth* sequences are lower than those in other sequences. The pigeon has a wide range of activities in the scene shown in Figure 7(a). It can jump onto a high platform and walk beside a water bottle and a feeder, which serve as the background. Because of the fast and abrupt pose changes of the pigeon, all algorithms fail to track in most of the frames. In the *pigeon_depth* sequence, our proposed algorithm has achieved slightly better results, as shown in Figure 7(b). In the future, we will focus on redetection technology and on utilizing the corresponding depth images, which will improve the tracking algorithm to achieve better results.

V. CONCLUSION

This paper developed an effective tracker based on the STC framework. We improved the scale-adaptive scheme by adding a penalty term rather than pre-defined sampling behaviors. Moreover, the powerful features, including HoG and color naming, are fused together to further enhance the overall performance for the proposed tracker. The extensive empirical evaluations on the test sequences demonstrate that the proposed method is promising for various challenging scenarios.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006, Art. no. 13.

[2] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," *Comput. Sci.*, vol. 53, no. 6025, pp. 68–83, 2015.

[3] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1808–1814.

[4] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[5] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[6] K. Zhang, L. Zhang, and M.-H. Yang, *Real-Time Compressive Tracking*. Berlin, Germany: Springer, 2012, pp. 864–877.

[7] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

[8] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 49–56.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[10] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1269–1276.

[11] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2544–2550.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, *Exploiting the Circulant Structure of Tracking-by-Detection With Kernels*. Berlin, Germany: Springer, 2012, pp. 702–715

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2005, pp. 886–893.

[14] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.

[15] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1177–1184.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[17] Y. Li and J. Zhu, *A Scale Adaptive Kernel Correlation Filter Tracker With Feature Integration*. Cham, Switzerland: Springer, 2015, pp. 254–265.

[18] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[19] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, *Fast Visual Tracking Via Dense Spatio-temporal Context Learning*. Cham, Switzerland: Springer, 2014, pp. 127–141.

[20] J. Xu, Y. Lu, and J. Liu, "Robust tracking via weighted spatio-temporal context learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 413–416.

[21] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1910–1917.

[22] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.
[23] S. Song and J. Xiao, "Tracking revisited using RGBD camera: Unified benchmark and baselines," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 233–240.

XIAOQIN ZHOU received the B.S. degree in computer science and education from Soochow University, China, in 2000, and the M.S. degree in computer science and technology from Hohai University, China, in 2012, where she is currently pursuing the Ph.D. degree in information and communication engineering. Her research interests include computer vision, signal processing, and human–robot interactions.

XIAOFENG LIU received the B.S. degree in electronics engineering and the M.S. degree in computer science from the Taiyuan University of Technology in 1997 and 1999, respectively, and the Ph.D. degree in biomedical engineering from Xi'an Jiaotong University in 2006. He is currently a Professor with the Department of Telecommunications, Hohai University, China, where he is also the Leader of the Cognition and Robotics Laboratory and the Director of the joint Laboratory of Aldebaran Robotics and Hohai University. He has over 11 grants as a PI and over 12 grants as a Researcher, including the National High-Tech R&D Program (863) and the National Basic Research Program (973). He has been granted 15 patents and authored 20 accredited journal papers. His current research interests focus on human–robot interactions, social robotics, and neural engineering.

CHENGUANG YANG (S'07–M'10–SM'16) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xi'an, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He received the postdoctoral training at the Imperial College London, U.K. He is currently with the Zienkiewicz Centre for Computational Engineering, Swansea University, U.K., as a Senior Lecturer. His research interests lie in robotics, automation, and computational intelligence.

AIMIN JIANG (S'07–M'11) received the B.E. and M.E. degrees from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001 and 2004, respectively, and the Ph.D. degree from the University of Windsor, Windsor, Canada, in 2010, all in electrical engineering. He is currently with the College of Internet of Things Engineering, Hohai University, Changzhou, China. His research interests include mathematical optimization and its applications to digital signal processing and communications. He has served as a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society.

BIN YAN (M'15) received the B.S. degree in applied physics from Qingdao University, China, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the Harbin Institute of Technology, China, in 2002 and 2007, respectively. From 1996 to 1999, he was an Engineer with the Goma Company Group. From 2007 to 2012, he was a Lecturer with the Shandong University of Science and Technology. From 2015 to 2016, he was a Visiting Scholar with Deakin University, Australia. Since 2013, he has been an Associate Professor with the Communication Engineering Department, Shandong University of Science and Technology. His research interests include multimedia signal processing, and security and signal processing for 3-D and virtual reality videos.