This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Object Tracking in Vary Lighting Conditions for Fog based Intelligent Surveillance of Public Spaces

Gaocheng Liu[1, 2], Shuai Liu[1, 2], Khan Muhammad[3], Arun Kumar Sangaiah[4], Faiyaz Doctor[5]

[1]College of Computer Science, Inner Mongolia University, Hohhot, China, 010012
[2] Inner Mongolia Key Laboratory of Social Computing and Data Processing, Hohhot, China, 010012
[3]Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea, 143747
[4]School of Computing Science and Engineering, VIT University, Vellore, India, 632014
[5]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, UK, CO43SQ

Corresponding author: Shuai Liu (e-mail: Cs_liushuai@imu.edu.cn/shuailiu3@acm.org).

ABSTRACT With rapid development of computer vision and artificial intelligence, cities are becoming more and more intelligent. Recently, since intelligent surveillance was applied in all kind of smart city services, object tracking attracted more attention. However, two serious problems blocked development of visual tracking in real applications. The first problem is its lower performance under intense illumination variation while the second issue is its slow speed. This paper addressed these two problems by proposing a correlation filter based tracker. Fog computing platform was deployed to accelerate the proposed tracking approach. The tracker was constructed by multiple positions' detections and alternate templates (MPAT). The detection position was repositioned according to the estimated speed of target by optical flow method, and the alternate template was stored with a template update mechanism, which were all computed at the edge. Experimental results on large-scale public benchmark datasets showed the effectiveness of the proposed method in comparison with state-of-the-art methods.

INDEX TERMS smart city; intelligent surveillance; object tracking; illumination variation; fog computing

I. INTRODUCTION

In the wake of recent terrorist incidents, which have taken place in crowded public places such as Paris and Brussels attacks in 2015 and 2016, as well as attacks in London and Manchester city in 2017, we need more effective systems for monitoring and tracking people and objects in large populated environments in order to keep public safety and security. Therefore, intelligent video surveillance system (IVS) has become an important research domain in ever smarter connected cities [1, 2]. An IVS system can be applied to identify and track targets, detect abnormal circumstances, provide timely alerts and automatic identification in dynamic public environments. The deployment of IVS onto embedded hardware platforms which is supported by edge and fog computing infrastructures is also important, because it can provide real-time tracking, analysis and critical decision support. An IVS generally includes five parts, which are video sequence acquisition [3], target detection, tracking, abnormal detection [4-6] and data analysis [7, 8]. Object tracking is applied to evaluate the object's state (such as position and size) from a complete video when the initial state of the target is given in the first frame of the video. In recent years, visual tracking has attracted widespread attention, and a lot of work in this field has been conducted from various perspectives [9-11].

Over past decades, a variety of effective tracking methods had been widely proposed [12-14], such as methods based on discriminative correlation filter (DCF) [15-17]. Correlation filter (CF) was first introduced by Galton in 1888 [18], after that it had been used to deal with various computer vision difficulties, such as object detection and recognition [19, 20], attitude detection [21], and object tracking [15]. DCF based methods used Fourier transform to solve the target detection and training process in Fourier domain. These techniques can heavily reduce calculations and accelerate the tracking speed. Because of its high computational efficiency, correlation filter had attracted more attention in object tracking applications [22-26].

Although many advances have been made in these methods, there are still many challenges remaining in real tracking application areas. The target or background will be affected by dynamic changes in the tracking process, especially under intense illumination variation. An example is presented in Figure 1. We can see that the tracked object has changed greatly under intense illumination variation. In this condition, the features of the target cannot be extracted accurately which will cause the failure of tracking. Today, most of the DCF based trackers use the histogram of oriented gradient (HOG) feature to improve tracking performance under illumination variation [27]. However, the performance of these trackers is poor when tested under intense illumination variation.

The IVS is widely deployed in urban centers. It is an important part for realizing smart city systems. Fog computing has unique properties in terms of computation and storage. For dynamic city monitoring, fog computing can satisfy the requirements of real-time data fusion. Fog computing, an extension of cloud computing [12], has the ability to solve critical tasks including information fusion, high-speed decision and situation awareness by taking advantage of calculating resources on the edge of the network, such as the embedded and mobile computing devices carried by end users [28]. Fog computing can perform real-time computing tasks because there are many heterogeneous smart devices with computing power at the edge of the network. The smart devices are regarded as the Fog nodes which are able to perform on-site real-time analysis for intelligent surveillance [29]. Therefore, we can use the sufficient resource of fog computing to store captured video streams, transcode and deal with video frames for tasks such as object recognition and object tracking in IVS. After that, notifications, events and depictions are sent to the terminal user, central server, and databases. The main contributions of our work are summarized below:

(1) First, when moving objects such as pedestrians and vehicles appear in the real-time surveillance video, they need to be detected immediately and effectively tracked frame by frame. In the actual scene, occlusions, background clutter and illumination changes may occur during the tracking process. These changes can lead to the failure of tracking. In order to improve the accuracy and robustness of the tracking algorithms under illumination variations, a new strategy by performing multi positions' detection and using alternate templates (MPAT) based on fog computing deployment architecture is proposed in this paper.

(2) Second, we proposed a deployment framework for IVS based on fog computing. Monitoring devices such as cameras are used to capture video information. Then the original video stream is transmitted to the fog computing node where object tracking algorithms are applied. Using a divide and conquer strategy, the sub regions contained the object of interests are determined. After that, the object is sent to the fog node for handling. After handling the sub regions in fog nodes, the results of tracking are returned to the end users.

The rest of this paper is arranged as follows: In Section 2, we outline some of the prior art related to this work. The descriptions and details of our proposed method (MPAT) are introduced in Section 3. The basic deployment structure of the tracking algorithm on fog computing is designed in Section 4. In Section 5, experimental results and analyses are provided. Finally, we conclude our work and provide future directions in Section 6.

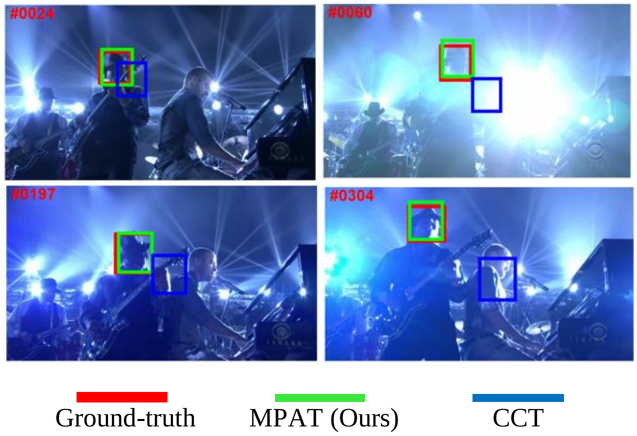

FIGURE 1. The tracking results of CCT (Zhu G et al, BMVC 2015 [34]) and our approach (MPAT) in the challenging situation of intense illumination change.

II. RELATED WORK

A. RESEARCH ON ILLUMINATION VARIATION

For overcoming the impact of intense illumination variation on visual moving target tracking and increasing the robustness of visual tracking algorithm, researchers have proposed a number of methods.

Some researchers selected the features which could best distinguish tracking target and the background to describe the surface model of tracking target [30, 31]. These methods can overcome the influence of target surface changes and background interference in the processes of the tracking. However, it cannot overcome the effects of the rapidly changing characteristics of the target surface, such as the changes of the color distribution and edge feature of targets. Besides, some researchers calculated the target's optical flow field to overcome the effect of severe illumination changes on color characteristics [32]. However, because of the restriction of computing resource, it was difficult to apply this method to do object tracking in actual scenarios. Traditional color histogram was sensitive to illumination change. In order to overcome the shortcomings, some researchers used Fuzzy C-Means clustering method to build the fuzzy color histogram of the target [33]. However, due to parameters setting problem, this approach can weaken the separability of the target model between the foreground and the background. Inappropriate initialization of the parameter may cause the failure of target tracking under the situations of severe illumination changes. Recently, many methods were proposed to improve the performance of vision tracking algorithms under illumination variation. Based on the fuzzy color histogram, a kernel tracking method was proposed to reduce the illumination sensitivity [34]. The HOG feature [27] was widely used in

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

tracking algorithms. The feature was constituted by calculating and counting histogram of gradient direction in a local region of image. The performance of these tracking algorithms was not shown to be excellent under intense illumination.

B. CORRELATION FILTERS FOR TRACKING

Since 1980s, correlation filter has been an important method of signal processing [18]. Then it is widely applied to target detection and recognition in the field of computer vision. Recently, high efficiency operations are achieved by replacing a large number of convolution operations with element-wise multiplication in discrete Fourier transform (DFT). As a result, correlation filters have attracted a lot of attention in visual tracking.

Many researchers achieved fruitful results on correlation filter in recent years. Bolme et al. proposed a target tracking algorithm by learning a minimum output sum of squared error (MOSSE) correlation filter on a gray scale image [15]. Henriques et al. proposed a circulant structure of tracking-by-detection with kernels (CSK) tracker. The CSK tracker learned a kernelized regularized least squares classifier of the target appearance by doing dense sampling [35]. It speeded up the learning process by taking advantage of the circular structure of adjoining sub-windows in an image. The CSK tracker used single channel intensity features to describe the target. The Kernelized Correlation Filter (KCF) tracker was an extension of the CSK tracker which can use more robust features (e.g. HOG) [36]. Danelljan et al. proposed an adaptive multi-scale correlation filtering tracking algorithm (DSST), which was used to solve the problem of multi-scale variations of targets [37]. Danelljan et al. used shallow convolutional neural network (CNN) features to effectively represent spatial location information. Then, they proposed a method called learning continuous convolution operators for visual tracking (C-COT). The tracker combined CNN features with correlation filter [38]. Similar to C-COT tracker, CNN derived features were also applied to the correlation filter based algorithms, which improved the adaptability of the algorithms [39-41].

C. DATASETS

As for tracker evaluation, one of the most widely used benchmarks is Online object tracking: A benchmark (OTB2013) [42]. OTB2013 contains 50 videos and has a total of more than 23 thousand frames. It also has clear annotations for videos. The OTB2015 dataset is an extension of the OTB2013 dataset, and contains 100 videos [43]. Two novel performance metrics are used for evaluation in these two datasets. They are the area under curve (AUC) of the overlap rate curve and the central pixel distance curve. The videos in the datasets are annotated with eleven attributes, which describe the challenges that a tracker will face in each sequence. The eleven challenging attributes are illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC) and low resolution (LR). The two datasets use one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) to evaluate the performance of various tracking algorithms. The conventional way to evaluate trackers is to run them throughout a test sequence with initialization from the ground truth position in the first frame. This way is referred as OPE. Doing the initialization on different frames in a sequence to evaluate trackers is referred as TRE. Given one initial frame together with the ground-truth bounding box of target, one tracker is initialized and run to the end of the sequence, i.e., one segment of the entire sequence. The tracking algorithms are evaluated on each segment of the sequence, and the overall statistics are tallied. Using different initialization of the bounding box in the first frame to evaluate trackers is referred as SRE. The bounding box is initialized by shifting or scaling the ground truth in the first frame. The twelve kinds of spatial shifts are used to initialize the bounding box, which are four kinds of center shifts, four corner shifts, and four kinds of scale changes. The performance of the tracking algorithms will be evaluated in different space initialization.

D. FOG COMPUTING

In brief, fog computing is the localization of some cloud services and a platform which can provide rich services and applications on the edge of the network [44, 45]. Fog computing is built on the edge servers between sensor networks and cloud based data centers. It provides limited distributed computing, storage, and network services for users. Fog computing provides local intelligent analysis and feedback services for sensor networks. Fog computing can reduce the application delay, the network bandwidth pressure and the calculation load of data center by providing data preprocessing services for cloud computing [46].

Fog computing expands the network computing model of cloud computing. It can get more extensive use in various services such as surveillance by extending network computing from the network center to the edge of the network. Fog computing is more widely distributed than cloud computing and has a wider range of mobility. These characteristics make it adaptable to today's increasing number of smart devices. Fog computing can be widely used in the field of intelligent analytics because it has a greater advantage for some time-sensitive applications such as real-time and streaming media applications. Fog computing supports mobility and has wide distribution of nodes. Fog computing can be used in broad application areas such as Intelligent Internet of Vehicle (IIOV), Smart Grid, Body Area Network (BAN), Smart City and other fields [47]. For example, Kirak Hong et al. proposed the Mobile Fog and used it to track vehicles and monitor traffic cases in 2013 [48]. Fog computing is suitable for applications with low latency, emergency services, and video streaming and is an important part of intelligent communications.

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

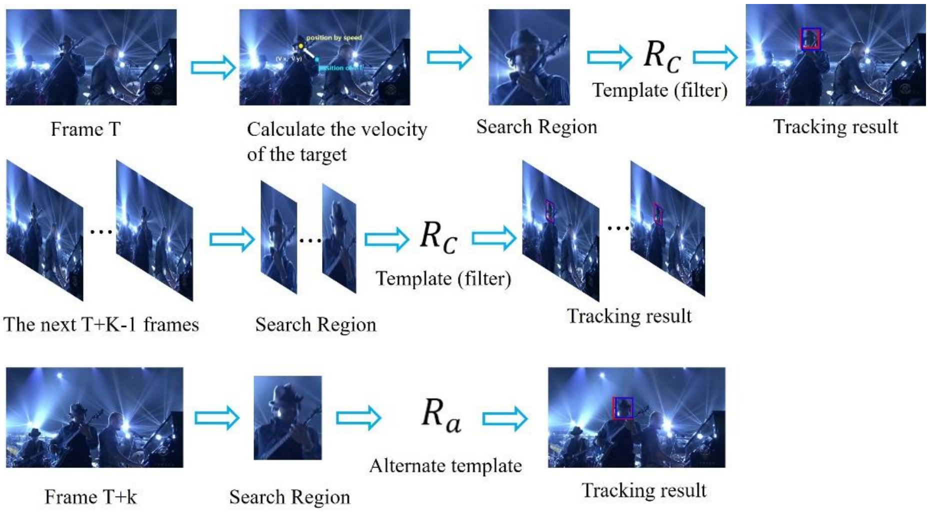

FIGURE 2. **The flowchart of our strategy when the frame is under the intense illumination variation. At t moment, the position of the object is predicted according to the speed of the object and the re-detection is performed in this position. The red box is the ground truth and the blue box is the tracking result of our approach. R_c is the template (context regression model) and it is updated every frame. The R_a is the alternate template which is saved before intense illumination occurs. We use alternate template R_a to detect object and update the template when the intense light disappears.**

III. THE PROPOSED MPAT APPROACH

In a complete sequence of images, the feature information of the target or the background may change in different situations, such as illumination variation, occlusion and scale variation. In these cases, accurate target tracking is a great challenge on complex dynamic scenes. The candidate window which is finally selected as the tracking target in the current frame has very low confidence value under complex dynamic scenes. Therefore, we aim to develop strategies to improve the performance of online tracking algorithms and enable them to be adaptive to complex situations without being prone to drifting. In this paper, we focus on improving the robustness and accuracy of the correlation filter based trackers for handling intense illumination change. A very effective strategy (MPAT) is proposed to solve this problem. The following sections explain the details of our strategy. An overview for our strategy is shown in Algorithm 1 and the flowchart of our approach is displayed in Figure 2.

A. CORRELATION TRACKING

Our strategy is combined with the Discriminative Correlation Filter (DCF) algorithms to improve their robustness under intense illumination conditions. Collaborative Correlation Tracking (CCT) is one of the DCF methods [49]. The DCF algorithm learns a correlation filter w from a series of training samples. It uses an $(M \times N)$ image patch x to train a classifier $f(x) = \langle w, \phi(x)\rangle$ by doing cyclic shift operations on it to produce training samples, where M and N are the length and width of the patch. It is defined that $x = (x_1 \cdots x_n)$ is a one-dimensional patch, and it is the base sample. The $x_1 = (x_n, x_1 \cdots x_{n-1})$ represents one cyclic shift operation of x. The x_i is the i^{th} training sample where $i \in \{0, \dots, M-1\} \times \{0, \dots, N-1\}$. All the training samples which are obtained by performing all cyclic shift operations form a cyclic matrix X:

$$\mathrm{X} = \mathrm{C}(x) = \begin{bmatrix} x_1, x_2 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_2, x_3 & \cdots & x_1 \end{bmatrix} \tag{1}$$

Each example x_i achieves a label $y_i \in [0,1]$. The value is generated by a Gaussian function based on the shifted distance. By minimizing the regression error, the classifier can be trained as follows:

$$\mathrm{w} = \arg\min_w \sum_i (\langle w, \phi(x)\rangle - y_i)^2 + \lambda \|w\|^2 \tag{2}$$

where $\phi(x)$ is the mapping to the Hilbert Space, $\lambda \geq 0$ is the regularization parameter which dominates the simplicity of model. Efficient training and detection can be achieved by exploiting the fast Fourier transform (FFT). When a translation invariant kernel is adopted, the solution of Eq. (2) can be expressed in the following form because of special nature of Circulant Matrices:

$$\mathrm{w} = X^T \alpha \tag{3a}$$

$$\alpha = (\mathrm{X}X^T + \lambda)^{-1} \tag{3b}$$

where X^T is the transpose matrix of X, and α is the dual variable. We can quickly obtain $\hat{\alpha}$ as follows by exploiting the special nature of Circulant Matrices.

$$\hat{\alpha} = \frac{\hat{y}}{\mathcal{F}(\phi^T(x)\phi(x)) + \lambda} \tag{4}$$

where the $\phi^T(x)$ is the transpose of $\phi(x)$, $\hat{\alpha}$ and $\hat{y}$ are discrete Fourier transform (DFT) of α and y. The $\mathcal{F}$ denotes the discrete Fourier operator.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

An $(M \times N)$ candidate image patch z is used as the search area in the process of translation estimation. The responses of all the cyclic patches can be evaluated by Eq. (5).

$$f(z) = \mathcal{F}^{-1}(\mathcal{F}(\phi^T(x)\phi(z)) \odot \hat{\alpha}) \qquad (5)$$

where $f(z)$ is filtered responses (Confidence) for all the cyclic shifts of z, and the $\odot$ is the Hadamard product. $\mathcal{F}^{-1}$ represents Inverse Fourier transform. Then, the patch which with the highest response is estimated as the target in current frame.

Algorithm 1: The strategy of improved tracking algorithm

Input: The first frame of the video sequence, the ground truth of the initial target, the threshold $\mathcal{T}_a$ and the parameter K

Output: Estimate object position $X_t = (x_t, y_t)$ where (x_t, y_t) are the center point coordinates of the target bounding box in current frame and context regression model R_c.

repeat

  Detect: According to the target position of the last frame (x_{t-1}, y_{t-1}) to crop out the searching window at current frame t and extract the target features; Compute the correlation responses using R_c and Eq. 5. The candidate window with highest response y_t is chosen as the target in current frame and estimate the target position (x_t, y_t) according it.

  If $y_t < \mathcal{T}_a$ **then**

    Use Eq. 7 to calculate speed $(\mathcal{V}_x, \mathcal{V}_y)$ of object and calculate the possible position (x_{tp}, y_{tp}) according to Eq. 8

    Do **Detect** according to (x_{tp}, y_{tp})

    Save the R_c as the alternate template R_a.

    Trained: Extract the target features according to (x_t, y_t) in current frame to train the template (context regression model) R_c by Eq. 4.

    Update the R_c according to the Eq. 9.

    If $t <$ frame $< t + K$ **then**

      Do **Detect,** using the R_c to compute the correlation map y_t

      Update the R_c according to the Eq. 9.

    else-if frame $= t + K$ **then**

      When do **Detect**, using the R_a to compute the correlation map y_t

      Let $R_c = R_a$

      Update the R_c according to the Eq. 9.

    End

  End

until End of video sequences;

B. MULTI POSITIONS DETECTION UNDER ILLUMINATION VARIATION

During the detection process, correlation filter based trackers detect the object by performing cyclic shift operations at the position of the previous frame which can generate a series of candidate windows. Then these candidate windows perform the correlation calculation with the filter which has been trained on the previous frame. After that, the candidate window with the highest confidence is chosen as the target. The appearance model is an important part of tracking algorithms, which mainly consists of two parts: visual representation and statistical modeling [50]. Visual representation uses different types of visual features to describe objects. Statistical modeling exploits statistical learning techniques to build the object models. The appearance model of the tracking object can be changed greatly under intense illumination variation conditions. As a result, the original features of the target are lost and the confidence of the target is lower than a threshold $\mathcal{T}_a$. The tracking results are inaccurate in this case. Figure 3 shows that the most of DCF algorithms use an image patch which contains the object and the background information surrounding the object to obtain the appearance model. In order to calculate the threshold $\mathcal{T}_a$, we first calculate the similarity value $\mathcal{C}$ of two image patches which have the same target but the background information is different by Eq. 5. The target's confidence is lower than the similarity value $\mathcal{C}$ when the tracking failure occurs under intense illumination. In this case, the results of tracking are unreliable and need to do re-detection. The threshold value $\mathcal{T}_a$ can be calculated by Eq. 6, where β is the confidence parameter.

$$\mathcal{T}_a = \frac{\mathcal{C}}{\beta} \qquad (6)$$

In order to make an accurate tracking of the object under intense illumination, the position of the object is predicted according to the speed of the object and the re-detection is performed in this position. The speed of the object can provide a precise location where the object is located. We use optical flow method to calculate the target speed between two continuous frames, see Eqs. 7a-7b.

$$\mathcal{V}_x = \sum_{i=1}^{m+n} \frac{v_x^i}{m+n} \qquad (7a)$$

$$\mathcal{V}_y = \sum_{i=1}^{m+n} \frac{v_y^i}{m+n} \qquad (7b)$$

where m and n are the size of the target box, v_x^i and v_y^i represent the speed of each pixel i in the target bounding box. v_x is the speed in the horizontal direction and v_y is the vertical speed. The velocity can be obtained by the Lucas-Kanade method [51] which uses the changes of pixels in the image sequence and the correlation between adjacent frames. Therefore, the method can find the corresponding relation between the previous frame and the current frame. Then, it calculates the motion of an object between adjacent frames. $\mathcal{V}_x$ and $\mathcal{V}_y$ are the velocity of the object in the horizontal and vertical directions.

The possible position of the object in the next frame can be predicted on the basis of its speed by Eqs. 8a-8b.

$$P_x^t = P_x^{t-1} + \mathcal{V}_x \tag{8a}$$

$$P_y^t = P_y^{t-1} + \mathcal{V}_y \tag{8b}$$

where P_x^{t-1} and P_y^{t-1} are the central position of the target in the horizontal and vertical directions at the $t-1$ moment. P_x^t and P_y^t are the central position of the target at the t moment. Dense sampling is carried out around the location and the confidence values of these samples are computed by Eq. 5. The highest one is chosen as the object.

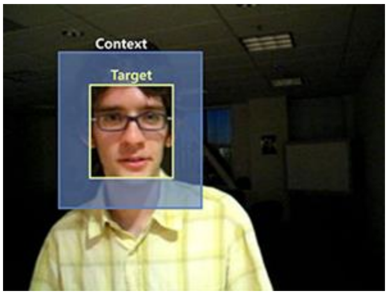

FIGURE 3. The image patch which contains the target and the background is used to get the appearance model. The target is in the Yellow rectangular and the Blue area is the background context surrounding the target.

C. ALTERNATE TEMPLATE

Our template update strategy is introduced in this section. Most of the correlation filter based trackers adopt the real-time template (filter) updates. In the process of tracking, each frame produces a correlation filter. Trackers update the object template by combining the filter with existing trained filter. The template is updated according to Eqs. 9a-9b, where the t represents current frame and θ is interpolation weights. The $\hat{x}^t$ is the object appearance model trained by the current frame. The $\hat{\alpha}^t$ is the dual variable which can be obtained by Eq. 4 in the current frame. This update method can provide the tracker with some memory of the target information in the previous frame by linearly interpolating the obtained values of α̂ and x̂ with the ones from the previous frame. As a result, it can improve the robustness of the tracker. When the target is under intense illumination conditions, the template cannot describe the target well. If we continue to adopt this update method, it can cause the failure of the tracking.

In order to solve this problem, we make some improvements to the template update process. In the current frame, when the confidence value of the selected target is lower than the threshold $\mathcal{T}_a$, we save the template (context regression model) which has been trained from the previous frame as an alternate template R_a. Based on the speed of object, the target which is obtained by performing multi positions' detection in the current frame, is used to train another template (context regression model) R_c. Because an intense lighting event can last for a period of time. In the next K frames, the template R_c is used to detect the target and updated with the existing trained filter in each frame. Meanwhile the template R_a is in reserve and not be updated. After K frames, we use alternate template R_a to detect object and update the template.

$$\hat{X}^t = (1-\theta)\hat{X}^{t-1} + \theta\hat{x}^t \tag{9a}$$

$$\hat{A}^t = (1-\theta)\hat{A}^{t-1} + \theta\hat{\alpha}^t \tag{9b}$$

IV. FOG BASED DEPLOYMENT INFRASTRUCTURE

Fog computing is a virtualization platform distributed between the end user and network cloud data. It can reduce latency, energy consumption and data traffic.

A. THE FRAMEWORK OF FOG COMPUTING

Fog computing architecture mainly includes cloud computing layer, fog computing layer and end user layer. It can be represented as a three level hierarchy [52]. Figure 4 shows that each smart device (cell phone, office building sensors, car, monitoring device, etc.) is connected to nearby fog device controllers which are connected to the cloud server. Fog components such as local servers or embedded computers are mainly deployed on the edge of the network. Fog computing has the following characteristics [53]:

1. The location of the fog computing is on the edge of the network. Fog computing has the characteristics of location awareness and adaptive mobility which allows intelligent devices to communicate or transmit data directly through wireless networks. As a result, the same user can quickly switch the response in different areas.
2. Low latency. Fog computing is suitable for some applications which need low latency, such as online games, video transmission and emergency response.
3. The geographical distribution is very extensive. Fog equipment is close to the user and is distributed in various places such as car system and roadside.

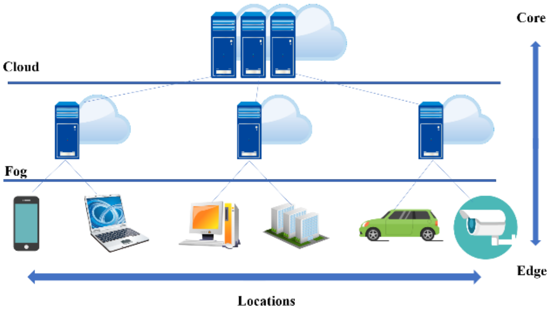

FIGURE 4. The framework of fog computing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

B. DEPLOYMENT INFRASTRUCTURE OF INTELLIGENT SURVEILLANCE SYSTEMS

In an IVS, fog computing can utilize the technologies such as radio frequency identification (RFID), sensors, wireless communications, satellite positioning, etc. to collect and annotate video information [54]. With the help of fog computing framework, the collected multi-target information can be analyzed on the service platform. After that, an IVS can provide real-time monitoring, positioning and tracking to stakeholders such as municipal authorities, police and security services. The deployment infrastructure of IVS is shown in Figure 5.

An IVS architecture consists of three layers, including computing layer, surveillance application layer and fog computing layer. The fog computing layer, which is constituted by a variety of smart on-site devices, plays a crucial role in the IVS. The smart devices such as smart tablets, smart phones and computers not only perform data processing but also serve as fog computing nodes [55]. When the original video stream is collected, they are allocated to near-site fog computing equipment to do processing rather than devolving them to the cloud based remote data centers.

In IVS, the sensors such as a drone or cameras in the city, are used to monitor the public occasions. When the video data is collected, the original video stream is returned to ground controllers and displayed on the monitor screens. The personnel can select the object of interest to do further tracking in real-time video. The tracking algorithms are implemented on the near-site fog computing nodes. Every frame in the surveillance videos can be handled on the fog computing nodes. It is worth noting that the fog computing nodes not only offer computing power, but also provide storage space. The original video data can be pre-processed and stored in fog computing nodes. Then it can be sent to the remote cloud center to do longer duration analysis for IVS. For example, we can do a long-term analysis of traffic conditions in smart cities to solve urban congestion problems.

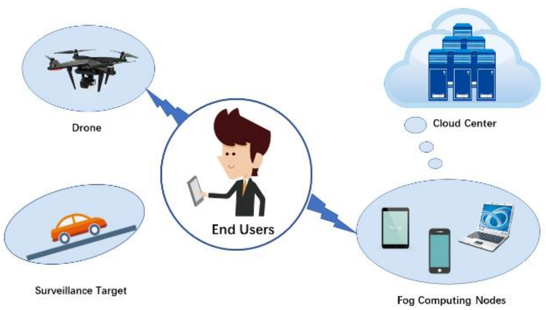

FIGURE 5. The deployment infrastructure of intelligent surveillance systems.

V. EXPERIMENTAL RESULTS AND ANALYSIS

We use two well-known benchmark datasets which are Online object tracking: A benchmark (OTB2013) [42] and Object Tracking Benchmark (OTB2015) [43] to evaluate our proposed algorithm. We also compare our method with other well-known algorithms. The OTB2013 benchmark dataset contains 50 challenging videos. The OTB2015 benchmark contains 100 challenging videos which is expanded by the OTB2013. All tracking algorithms do one-pass evaluation (OPE). The way to evaluate trackers is to run them throughout a test sequence with initialization from the ground truth position in the first frame. All the tracking methods are evaluated by three metrics.

a). Distance precision (DP). It illustrates the percentage of frames whose estimated location is within the given threshold distance of the ground truth.

b). Overlap success rate (OS). It is defined as the percentage of frames where bounding box overlap more than a threshold.

c). Center location error (CLE). It represents the mean Euclidean distance between the ground truth and the estimated center position.

Setup parameters: The confidence parameter of Eq. (6) is set to $\beta = 1.5$. The value $\mathcal{C}$ which can be obtained by the method mentioned in section 3.2 is set to 0.4. Based on Eq. 6, the threshold $\mathcal{T}_a$ is set to 0.25. When the confidence of the target is lower than 0.25, the tracking results are not reliable and need to do re-detection. For the update of the template, we set $\mathrm{K} = 4$.

A. EVALUATION OF IMPROVED DCF ALGORITHM

The precision plots and success plots are used to show the performance of tracking algorithms. The precision plot exhibits the percentage of frames whose estimated location is within the given threshold distance of ground truth. As a representative precision score for each tracker we use the score for the threshold = 20 pixels [56]. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. Using one success rate value at a specific threshold (e.g. to=0.5) for tracker evaluation may not be fair or representative. Instead we use the area under curve (AUC) of each success plot to rank the tracking algorithms.

B. OVERALL PERFORMANCE

The tracking results of our proposed approach and the baseline tracker for OPE are shown in Figure 6. MPAT obtains Distance precision (DP) of 82.4% and Success rate (OS) of 61.1% on OTB2013. MPAT improves the DP and OS scores of the baseline tracker (CCT) which obtains DP of 79.5% and OS of 59.4% on OTB2013. Result of MPAT's performance on OTB2015 is shown in Figure 6 with DP score (79.7%) and OS score (58.3%), which is higher than CCT's score (DP 73.5% and OS 54.9%). The results prove that our approach is able to ameliorate the performance of the underlying algorithm significantly. The speed of MPAT is 35.4 frames per second (FPS) and CCT runs at 50 (FPS). Because calculating the speed of each pixel by using Lucas-Kanade method requires large computation, MPAT is slower than the CCT tracker. Both of our approach and CCT can operate in real-time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

We not only compared our method with basic tracker (CCT) but also evaluated the proposed approach with other seven state-of-the-art trackers which are DCFNet [57], Staple [58], BIT [59], fDSST [60], SiamFC_3s [61], RPT [62] and CFNet-conv3 [63]. The DCFNet tracker presented an end-to-end lightweight network architecture to learn the convolutional features and performed the correlation tracking process simultaneously. The Staple tracker did a simple combination of a Correlation Filter (using HOG features) and a global color histogram. The BIT tracker cascaded four units including appearance model (S1 and C1 units) and tracking model (S2 and C2 units): S1 unit extracted texture and color information; C1 unit collected texture and color features and combined them by using complex response maps; S2 unit learned view-turned feature; C2 unit used a full-connection neural network for task-dependent learning. The fDSST tracker was a novel scale adaptive tracking method. By learning separate DCFs, the translation and scale estimation were carried out. The tracker was a novel fully-convolutional Siamese network which was trained end-to-end. The neural network only had convolution layers and pooling layers. The RPT tracker utilized reliable patches to build a novel sequential Monte Carlo framework. It studied how to use visual information to calculate the likelihood of tracking reliability for a patch and estimate likelihood of the patch-on-object from motion information. CFNet-conv3 tracker used correlation filtering as a layer of convolutional neural networks on the basis of SiamFC algorithm [61].

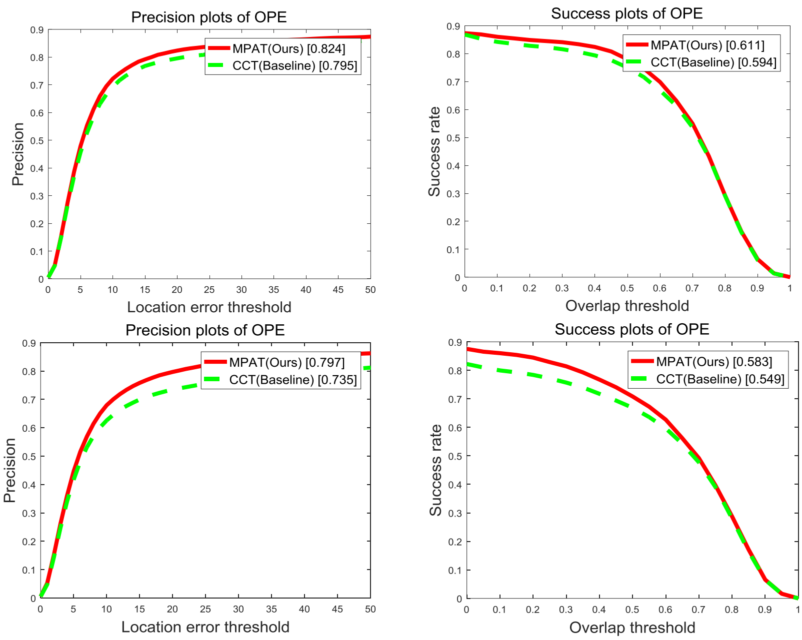

FIGURE 6. **The distance precision and overlap success plots of OPE on OTB2013 and OTB2015. The first row is the results on OTB2013 and the second row is the results on OTB2015. The improved algorithm (MPAT) outperforms the basic algorithm (CCT) [49].**

TABLE 1
THE RESULTS OF OUR METHOD AND OTHER SEVEN STATE-OF-THE-ART TRACKERS ON OTB2013

	DCFNet [57]	Staple [58]	BIT [59]	fDSST [60]	SiamFC_3s [61]	RPT [62]	CFNet-conv3 [63]	MPAT (ours)
DP (%)	79.5	79.3	81.6	80.2	80.9	81.0	82.2	82.4
OS (%)	62.2	60.0	59.3	59.5	60.8	57.8	61.0	61.1
Speed (fps)	27.3	50.8	22.6	39.5	13.1	2.2	11.1	35.4

TABLE 2
THE RESULTS OF OUR METHOD AND OTHER SEVEN STATE-OF-THE-ART TRACKERS ON OTB2015

	DCFNet [57]	Staple [58]	BIT [59]	fDSST [60]	SiamFC_3s [61]	RPT [62]	CFNet-conv3 [63]	MPAT (ours)
DP (%)	75.1	78.4	73.1	72.4	77.1	75.6	77.7	79.7
OS (%)	58.0	58.1	51.5	55.2	58.2	53.5	58.9	58.3
Speed (fps)	27.3	50.8	22.6	39.5	13.1	2.2	11.1	35.4

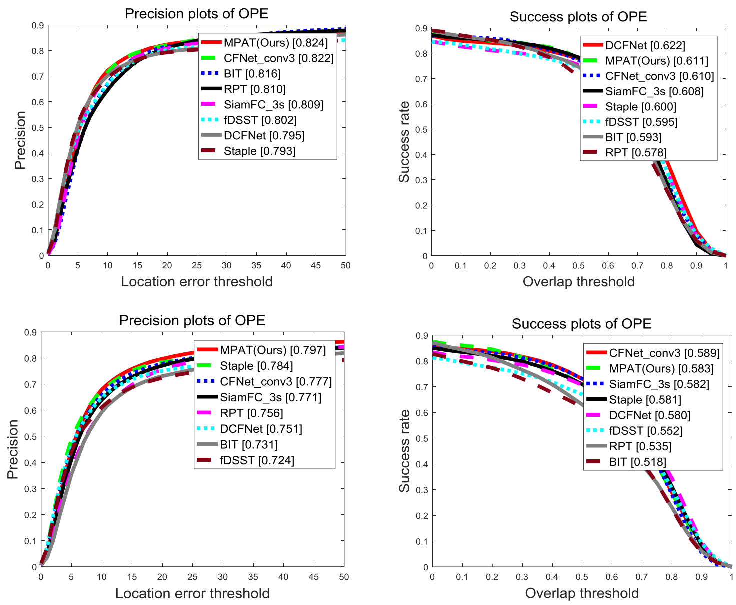

FIGURE 7. Precision and success plots using OPE on OTB2013 and OTB2015. The first row is the results on OTB2013 and the second row is the results on OTB2015. The numbers in the legend indicate the representative precisions at 20 pixels for precision plots, and the AUC scores for success plots. Our approach MPAT performs favorably against the state-of-the-art trackers.

The above seven algorithms can be divided into two typical tracking algorithms, (1) One is based on CF framework without using deep learning methods (Staple, BIT, fDSST, and RPT). These trackers are based on the CFs without using the deep leaning methods to do training and extract features. (2) Another one is tracking by using CF methods and deep learning methods (DCFNet, SiamFC_3s and CFNet-con3). These trackers combine deep learning methods with correlation filtering, such as convolutional neural networks (CNN).

The results are illustrated in Figure 7, Table 1 and Table 2. The top two trackers are marked in red and blue, respectively. In precision plots, MPAT obtains the best performance both on OTB2013 and OTB2015 with DP score of 82.4% and 79.7%. In success plots, MPAT is the second-best algorithm with OS score of 61.1% on OTB2013. DCFNet which uses the deep learning methods to represent object appearance is higher than MPAT's score (OS 62.2%). CFNet-conv3 obtains the best performance with OS score of 58.9% on OTB2015. MPAT can perform better than the trackers both in category (1), like (Staple and fDSST) and category (2), like (SiamFC_3s). For overall performance, MPAT can perform well both on the OTB2013 and OTB2015. This is different from several other trackers, which only achieve good performance in one of the datasets. For example, DCFNet tracker obtains the best OS score on OTB2013 but performs very poor on OTB2015. The experimental results demonstrate that our method is effective compared to other methods under consideration. The speed of each tracker is shown in Table 1. All results are run on a 3.5GHz Intel Core i7 CPU with 32 GB memory. The deep learning trackers except DCFNet run at around 10 frames per second (FPS). Their speed is very slow which cannot meet the requirements of real–time processing. Staple tracker is the fastest algorithm with the speed of 50.8 fps. The speed of MPAT is 39.5 fps. Both the Staple and MPAT can meet the real-time requirements.

C. THE PERFORMANCE ON THE SUBSETS WITH DIFFERENT CHALLENGING ATTRIBUTES

To analyze the performance of tracking algorithms in different scenarios, the videos in the benchmark are annotated with eleven challenging attributes, which are illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of plane rotation (OPR), out-of-view (OV), background clutters (BC) and low resolution (LR) respectively. Each attribute represents a specific challenging factor in object tracking. One video may be annotated with many attributes. For each challenging attribute, we evaluated the performance of trackers on the subsets with the challenging attributes in the benchmark. For example, if we want to evaluate the performance of the trackers on the challenging attribute of scale variation, we can use the 29 videos tagged with this challenging attribute in the benchmark. The main purpose of our paper is to deal with tracking problems under intense illumination variation conditions. Result of MPAT's performance under intense illumination variation conditions of OTB2013 is shown in Figure 8 with DP score (79.3%) and OS score (59.6%), which is higher than CCT's score (DP 74.3% and OS 56.1%). In OTB2015, MPAT also gets excellent performance under illumination variation. The results show that MPAT can better deal with the challenges of illumination variation in the tracking process.

TABLE 3
THE RESULTS OF OUR METHOD AND OTHER SEVEN STATE-OF-THE-ART TRACKERS ON OTB2013

	FM	BC	MB	DEF	IV	OCC	OPR	OV	SV	LR	IPR
DCFNet [57]	53.4	57.9	51.5	60.6	59.6	64.5	61.2	69.0	61.9	49.6	57.2
Staple [58]	50.8	57.6	54.1	61.8	56.8	59.3	57.5	54.7	55.1	43.8	58.0
BIT[59]	50.4	56.6	52.1	61.0	56.0	62.8	59.5	55.2	56.6	32.4	55.7
fDSST [60]	55.3	61.7	59.1	56.2	59.1	55.4	56.6	55.6	56.4	39.5	57.7
SiamFC_3s [61]	54.4	54.8	51.6	54.4	53.6	59.8	59.0	63.5	60.1	49.9	57.0
RPT [62]	56.1	61.4	57.6	53.7	55.9	53.3	55.5	57.6	53.8	36.3	56.9
CFNet-conv3 [63]	52.0	56.8	53.5	58.1	53.1	56.6	58.3	42.3	58.4	43.4	56.5
MPAT (ours)	55.4	58.9	53.8	63.9	59.6	60.5	59.6	62.6	57.7	41.0	55.3

TABLE 4
THE RESULTS OF OUR METHOD AND OTHER SEVEN STATE-OF-THE-ART TRACKERS ON OTB2015

	FM	BC	MB	DEF	IV	OCC	OPR	OV	SV	LR	IPR
DCFNet [57]	54.1	56.9	54.4	49.7	58.1	57.3	57.5	55.7	56.5	53.3	55.7
Staple [58]	53.7	57.4	54.6	55.4	59.8	54.8	53.4	48.1	52.5	41.8	55.2
BIT[59]	50.1	52.1	48.3	47.1	51.3	52.2	51.4	43.0	46.1	32.1	51.6
fDSST [60]	55.3	58.5	54.6	46.7	56.4	48.1	49.7	45.8	50.2	46.0	54.5
SiamFC_3s [61]	56.8	52.3	55.0	50.6	56.8	54.3	55.8	50.6	55.2	59.2	55.7
RPT [62]	54.3	57.5	51.7	49.2	53.7	48.1	50.3	47.5	48.1	35.9	52.3
CFNet-conv3 [63]	55.5	56.1	54.0	52.5	54.3	52.7	55.6	45.6	54.7	55.2	57.1
MPAT (ours)	56.2	60.2	54.4	54.7	60.5	56.5	56.0	51.2	55.6	49.7	54.4

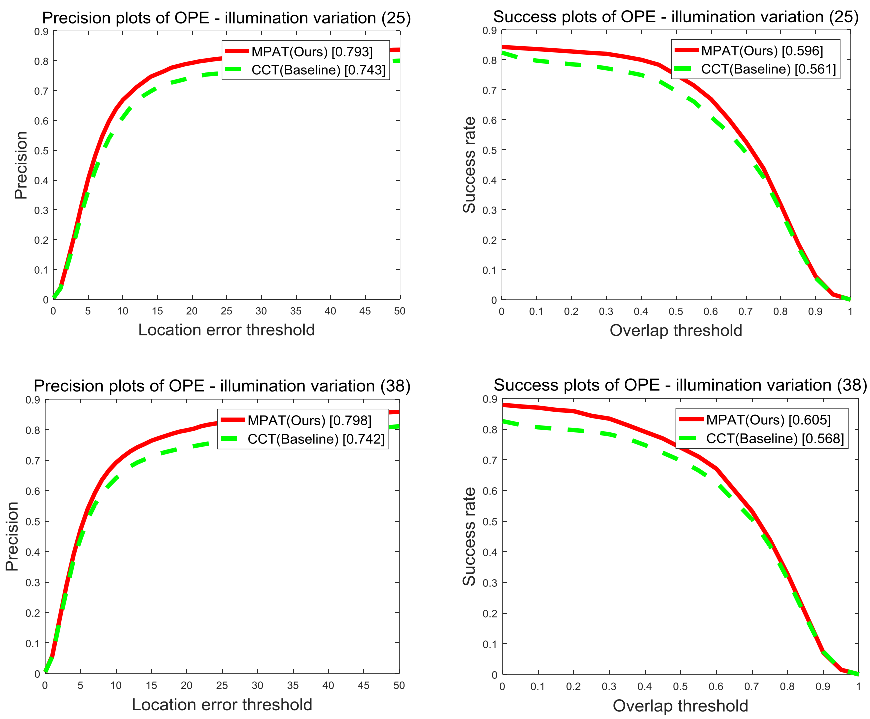

FIGURE 8. The precision plots and success plots over tracking challenge of illumination variation. The first row is the results on OTB2013 and the second row is the results on OTB2015. Our improved algorithm (MPAT) has outstanding performance compared to basic algorithms both on OTB2013 and OTB2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

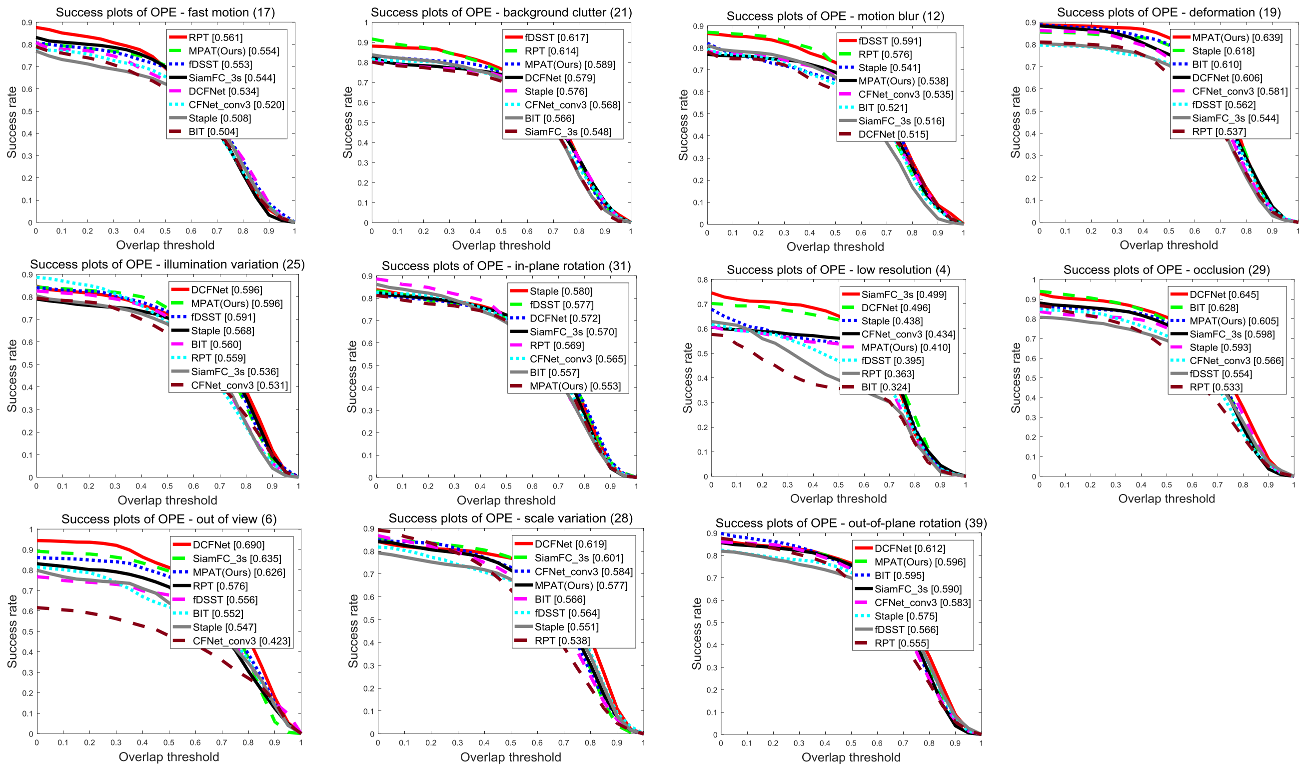

FIGURE 9. The success plots over eleven tracking challenging attributes of FM, BC, MB, OPR, DEF, SV, IV, OCC, OV, LR and IPR on OTB2013. The AUC values of each tracker are included in the legend. Our improved algorithm gets outstanding performance compared to seven excellent algorithms especially in scenarios with illumination variation (IV).

We also compared our improved algorithm with above seven state-of-the-art trackers on the eleven challenging attributes. Figure 9 shows the results on these eleven attributes on OTB2013. Details of the attribute-based performance are shown in Table 3. In Table 3, the top three trackers are marked in red, blue, and green for each challenging attribute, respectively. Our MPAT method achieves competitive performances among all the seven state-of-the-art trackers in the videos which contain illumination variation. The results clearly show that MPAT method is more robust under intensive illumination variation. Furthermore, Table 3 shows that MPAT method also can achieve superior performance and robustness on other six challenging attributes, which are fast motion, background clutters, deformation, out-of plane rotation, out-of-view and occlusion. The details of the attribute-based performance on OTB2015 are shown in Table 4, which denotes that our method has excellent robustness to illumination changes.

D. THE TRACKING VISUALIZATION RESULTS ON CHALLENGING SEQUENCES

We compared our algorithm with the basic tracker in some challenging sequences under intense illumination changes. The compared results are shown in Figure 10. When intense illumination changes occur, the baseline tracker (CCT) obviously cannot keep up with the target, but our MPAT method can deal with this scenario very well. The results in the figure demonstrate that our method can improve the performance of trackers under intense illumination changes.

VI. CONCLUSION

The demand of IVS for various kinds of security monitoring and tracking applications in a smart city had grown. However, an IVS was faced with a number of challenging scenarios such as illumination variations, occlusion and fast moving objects. In order to track target smoothly under different complex conditions, this paper provided a novel strategy which performed multi positions' detection and used alternate templates (MPAT) for object tracking. Further, a deployment framework was designed for IVS for intelligent surveillance systems based on fog computing. Extensive experiments were conducted and results were compared with other state of the art tracking algorithms. The tracking results showed that the proposed approach had outstanding of adaptability and robustness. Moreover, our strategy (MPAT) can be combined with other strategies. For example, we can track multiple objects based on the single object tracking algorithms by taking advantage of divide-and-conquer strategy. The proposed framework was able to provide a unifying platform. It is rich enough to deliver many emerging services and develops new urban surveillance and monitoring applications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2834916, IEEE Access

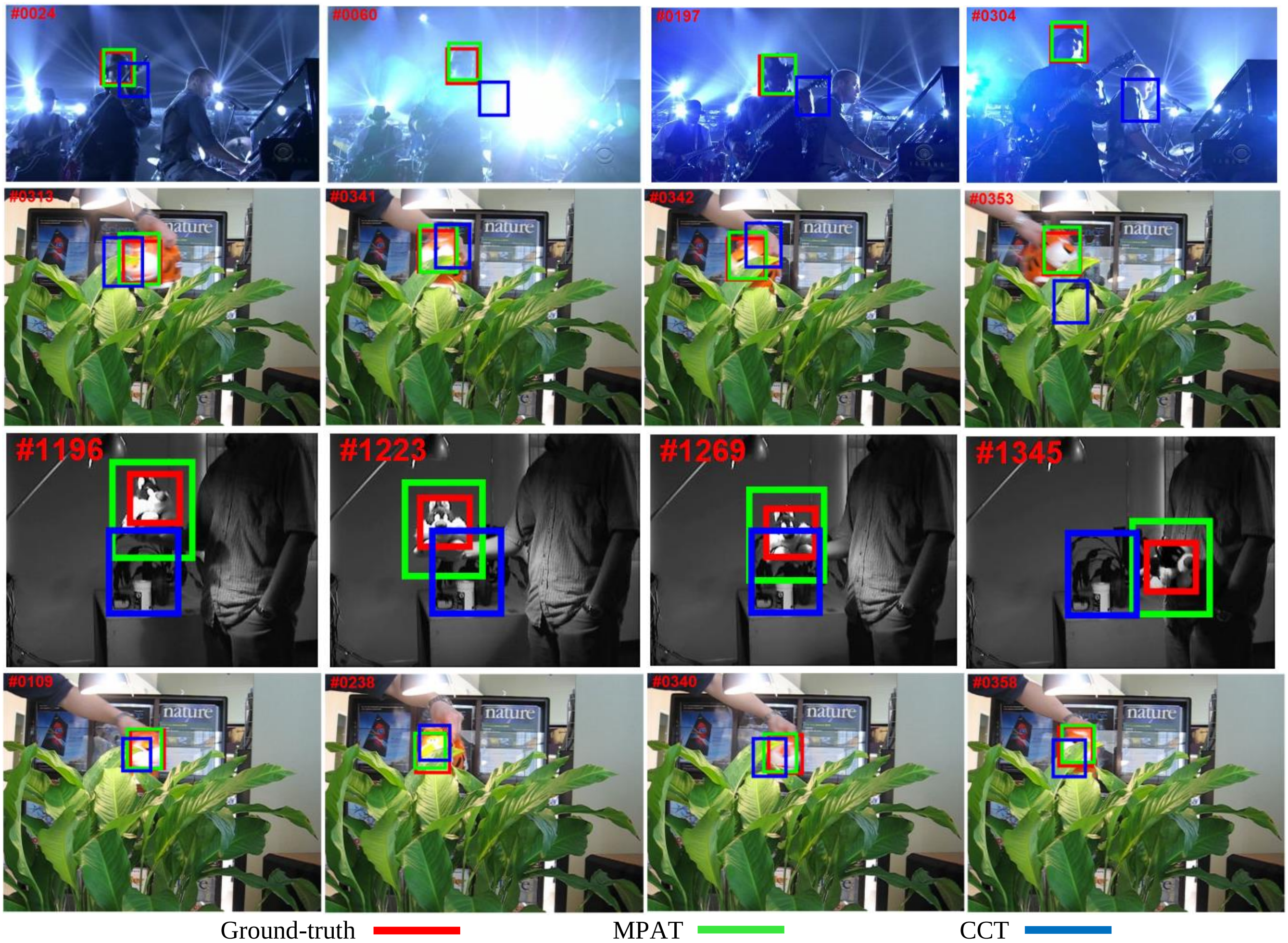

FIGURE 10. Tracking results of our method (MPAT) and CCT (Baseline) on four challenging sequences Shaking, Tiger1, Sylvester and Tiger2 (from top to bottom).

Moreover, due to lack of many related technologies such as resource management, security, privacy etc., the design and deployment of object tracking solutions based on fog computing still face many challenges today. Our future work will aim to address these challenges in order to deploy a prototype of our proposed system in an urban surveillance context.

ACKNOWLEDGMENT

The research is supported by National Natural Science Foundation of China (Grant No. 61502254) and Program for Yong Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (Grant No.NJYT-18-B10).

REFERENCES

[1] Han G, Guizani M, Lloret J, et al. Emerging Trends, issues, and Challenges in Big Data and Its Implementation toward Future Smart Cities[J]. IEEE Communications Magazine, 2017, 55(12):16-17

[2] Muhammad K, Hamza R, Ahmad J, et al. Secure Surveillance Framework for IoT systems using Probabilistic Image Encryption[J]. IEEE Transactions on Industrial Informatics, 2018.

[3] Chang X, Yu Y L, Yang Y, et al. Semantic Pooling for Complex Event Analysis in Untrimmed Videos[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(8):1617-1632.

[4] Muhammad K, Ahmad J, Baik S W. Early Fire Detection using Convolutional Neural Networks during Surveillance for Effective Disaster Management[J]. Neurocomputing, 2017.

[5] Chang X, Ma Z, Lin M, et al. Feature Interaction Augmented Sparse Learning for Fast Kinect Motion Detection.[J]. IEEE Trans Image Process, 2017, 26(8):3911-3920.

[6] Chang X, Ma Z, Yang Y, et al. Bi-Level Semantic Representation Analysis for Multimedia Event Detection[J]. IEEE Trans Cybern, 2017, 47(5):1180-1197.

[7] Chang X, Yang Y. Semisupervised Feature Analysis by Mining Correlations Among Multiple Tasks[J]. IEEE Trans Neural Netw Learn Syst, 2017, 28(10):2294-2305.

[8] Li Z, Nie F, Chang X, et al. Beyond Trace Ratio: Weighted Harmonic Mean of Trace Ratios for Multiclass Discriminant Analysis[J]. IEEE Transactions on Knowledge & Data Engineering, 2017, PP(99):2100-2110.

[9] Yilmaz A, Javed O, Shah M. Object tracking: A survey[J]. Acm computing surveys (CSUR), 2006, 38(4): 13.

[10] Pan Z, Liu S, Fu W. A review of visual moving target tracking[J]. Multimedia Tools and Applications, 2016: 1-30.

[11] Chen Z, Hong Z, Tao D. An experimental survey on correlation filter-based tracking[J]. arXiv preprint arXiv:1509.05520, 2015.

[12] Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, 34(7):1409-1422.

[13] Hare S, Saffari A, Torr P H S. Struck: Structured output tracking with kernels[C]// IEEE International Conference on Computer Vision. IEEE, 2012:263-270.

[14] Muhammad K, Sajjad M, Mehmood I, et al. A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image [J]. Multimedia Tools and Applications, 2016, 75(22): 14867-14893.

[15] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]// Computer Vision and Pattern Recognition. IEEE, 2010:2544-2550.

[16] Danelljan M, Robinson A, Khan F S, et al. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking[C]// European Conference on Computer Vision. Springer, Cham, 2016:472-488.

[17] Chen C, Liu X, Qiu T, et al. Latency estimation based on traffic density for video streaming in the internet of vehicles[J]. Computer Communications, 2017, 111: 176-186.

[18] Kumar B V K V, Mahalanobis A, Juday R D. Correlation pattern recognition[M]. Cambridge University Press, 2005.

[19] Henriques J F, Carreira J, Caseiro R, et al. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition[C]//proceedings of the IEEE International Conference on Computer Vision. 2013: 2760-2767.

[20] Kumar B V K V, Mahalanobis A, Juday R D. Correlation pattern recognition[M].Cambridge University Press, 2005.

[21] Henriques J F, Martins P, Caseiro R F, et al. Fast training of pose detectors in the fourier domain[C]//Advances in neural information processing systems. 2014: 3050-3058.

[22] Danelljan M, Khan F S, Felsberg M, et al. Adaptive Color Attributes for Real-Time Visual Tracking[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1090-1097.

[23] Montero A S, Lang J, Laganiere R. Scalable kernel correlation filter with sparse feature integration[C]//Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on. IEEE, 2015: 587-594.

[24] Samuel O W, Asogbon G M, Sangaiah A K, et al. An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction [J]. Expert Systems with Applications, 2017, 68: 163-172.

[25] Bibi A, Mueller M, Ghanem B. Target response adaptation for correlation filter tracking[C]//European Conference on Computer Vision. Springer International Publishing, 2016: 419-433.

[26] Danelljan M, Hager G, Shahbaz Khan F, et al. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1430-1438.

[27] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.

[28] Chen N, Chen Y, You Y, et al. Dynamic Urban Surveillance Video Stream Processing Using Fog Computing[C]// IEEE Second International Conference on Multimedia Big Data. IEEE, 2016:105-112.

[29] Yi S, Li C, Li Q. A survey of fog computing: concepts, applications and issues[C]//Proceedings of the 2015 Workshop on Mobile Big Data. ACM, 2015: 37-42.

[30] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking[J]. IEEE Transactions on pattern analysis and machine intelligence, 2003, 25(5): 564-577.

[31] Collins R T, Liu Y, Leordeanu M. Online selection of discriminative tracking features[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(10): 1631-1643.

[32] Freedman D, Turek M W. Illumination-invariant tracking via graph cuts[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 2: 10-17.

[33] Han J, Ma K K. Fuzzy color histogram and its use in color image retrieval[J]. IEEE Transactions on image Processing, 2002, 11(8): 944-952.

[34] Wang Y, Pan Q, Zhao C, et al. Fuzzy color histogram based kernel tracking under varying illumination[C]//Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on. IEEE, 2007, 4: 235-239.

[35] Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2012: 702-715.

[36] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[37] Danelljan M, Häger G, Khan F, et al. Accurate scale estimation for robust visual tracking[C]//British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press, 2014.

[38] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking[C]//European Conference on Computer Vision. Springer International Publishing, 2016: 472-488.

[39] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015: 58-66.

[40] Ma C, Huang J B, Yang X, et al. Hierarchical convolutional features for visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 3074-3082.

[41] Choi J, Chang H J, Yun S, et al. Attentional Correlation Filter Network for Adaptive Visual Tracking[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017.

[42] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2411-2418.

[43] Wu Y, Lim J, Yang M H. Object Tracking Benchmark[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1834-1848.

[44] Bonomi F. Connected vehicles, the internet of things, and fog computing[C]//The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA. 2011: 13-15.

[45] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]//Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012: 13-16.

[46] Shi Y, Ding G, Wang H, et al. The fog computing service for healthcare[C]//Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), 2015 2nd International Symposium on. IEEE, 2015: 1-5.

[47] Stojmenovic I, Wen S. The fog computing paradigm: Scenarios and security issues[C]//Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on. IEEE, 2014: 1-8.

[48] Hong K, Lillethun D, Ramachandran U, et al. Mobile fog: A programming model for large-scale applications on the internet of things[C]//Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing. ACM, 2013: 15-20.

[49] Zhu G, Wang J, Wu Y, et al. Collaborative Correlation Tracking[C]//BMVC. 2015: 184.1-184.12.

[50] Li X, Hu W, Shen C, et al. A survey of appearance models in visual object tracking[J]. Acm Transactions on Intelligent Systems & Technology, 2013, 4(4):58.

[51] Baker S, Matthews I. Lucas-kanade 20 years on: A unifying framework[J]. International journal of computer vision, 2004, 56(3): 221-255.

[52] Bonomi F, Milito R, Natarajan P, et al. Fog computing: A platform for internet of things and analytics[M]//Big Data and Internet of Things: A Roadmap for Smart Environments. Springer International Publishing, 2014: 169-186.

[53] Zhu J, Chan D S, Prabhu M S, et al. Improving web sites performance using edge servers in fog computing architecture[C]//Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on. IEEE, 2013: 320-323.

[54] Muhammad K, Sajjad M, Mehmood I, et al. Image steganography using uncorrelated color space and its application for security of visual contents in online social networks[J]. Future Generation Computer Systems, 2016. 10.1016/j.future.2016.11.029.

[55] Chen N, Chen Y, Song S, et al. Poster Abstract: Smart Urban Surveillance Using Fog Computing[C]// Edge Computing. IEEE, 2016:95-96.

[56] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. PAMI, 33(8):1619–1632,

2010.
[57] Wang Q, Gao J, Xing J, et al. DCFNet: Discriminant Correlation Filters Network for Visual Tracking[J]. arXiv preprint arXiv:1704.04057, 2017.
[58] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary learners for real-time tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1401-1409.
[59] Cai B, Xu X, Xing X, et al. BIT: Biologically inspired tracker[J]. IEEE Transactions on Image Processing, 2016, 25(3): 1327-1339.
[60] Danelljan M, Häger G, Khan F S, et al. Discriminative scale space tracking[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(8): 1561-1575.
[61] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European Conference on Computer Vision. Springer International Publishing, 2016: 850-865.
[62] Li Y, Zhu J, Hoi S C H. Reliable patch trackers: Robust visual tracking by exploiting reliable patches[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 353-361.
[63] Valmadre J, Bertinetto L, Henriques J F, et al. End-to-end representation learning for Correlation Filter based tracking[J]. arXiv preprint arXiv:1704.06036, 2017.

Gaocheng Liu is a graduate student studying at College of Computer Science, Inner Mongolia University, Hohhot, China. His main research areas are image processing and computer vision.

Shuai Liu received his bachelor's (2002), master's (2004) and doctoral (2011) degrees from the College of Computer Science and Technology, Jilin University. He now works as an Associate Professor in the College of Computer Science, Inner Mongolia University. He has published more than 20 papers in Elsevier, Springer, Oxford and other publishers. He is now interested in image processing, computer vision and fractals. He now acts as an (associate/guest) editor in IEEE Access, Mobile Networks and Application and so on. He also acts as a reviewer of IEEE Transactions on Fuzzy Systems, IEEE Transactions on Image Processing, Information Sciences, Applied Mathematics and Computation, IEEE Acces, and so.

Khan Muhammad (S'16) received his BS degree in computer science from Islamia College, Peshawar, Pakistan with research in information security. Currently, he is a research associate at Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea. He is working as a researcher at Intelligent Media Laboratory (IM Lab). His research interests include image and video processing, wireless networks, information security, image and video steganography, video summarization, diagnostic hysteroscopy, wireless capsule endoscopy, computer vision, IoT and CCTV video analysis. He has published over 45 papers in peer-reviewed international journals and conferences in these research areas. He is an active reviewer of more than 40 reputed journals and is involved in editing of several special issues.

Arun Kumar Sangaiah has received his Doctor of Philosophy (PhD) degree in Computer Science and Engineering from the VIT University, Vellore, India. He is presently working as an Associate Professor in School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. He has authored more than 100 publications in different journals and conference of national and international repute. His current research work includes global software development, wireless ad hoc and sensor networks, machine learning, cognitive networks and advances in mobile computing and communications. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Prof. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international journals.

Faiyaz Doctor is a Senior Lecturer in the Faculty of Engineering and Computing and associate head of the Intelligent Information Modelling and Retrieval research group at Coventry University. He has previously worked jointly in industry and academia to develop novel artificial intelligence solutions for addressing real world problems related to smart environments, energy optimization, predictive analytics and decision support. His work has resulted in high profile innovation awards (Best KTP Regional Finalist 2011, Load Stafford Award for Innovation) and an international patent on improved approaches for data analysis and decision-making using hybrid neuro-fuzzy and type-2 fuzzy systems: WO/2009/141631. His research interests include ambient intelligence, pervasive computing and applications of affective computing, computational intelligence with an emphasis on fuzzy logic, applications of type-2 fuzzy logic and hybrid systems. Dr. Doctor has published over 30 papers in peer reviewed international journals, conferences and workshops. He is co-organizer of the International Workshop on Applications of Affective Computing in Intelligent Environments (ACIE) in conjunction with the International Conference on Intelligent Environments and is the current vice chair of the IEEE Computational Intelligence Society's Emergent Technologies 'Affective Computing' Task Force. He is also a member of the IEEE and IEEE Computational Intelligence Society.