IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 22, 2020, accepted December 16, 2020, date of publication December 23, 2020,
date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046763

Multiple Object Tracking Using Edge Multi-Channel Gradient Model With ORB Feature

JIEYU CHEN[1], ZHENGHAO XI[1], CHI WEI[2], JUNXIN LU[1], YUHUI NIU[1], AND ZHONGFENG LI[3]

[1]School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[2]Department of Electrical and Electronic Engineering, University of Nottingham, Nottingham NG7 2RD, U.K.
[3]College of Electrical Engineering, Yingkou Institute of Technology, Yingkou 115014, China

Corresponding author: Zhenghao Xi (zhenghaoxi@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801286, Grant 61701295, and Grant 61703270.

ABSTRACT Multiple object tracking based on tracking-by-detection is the most common method used in addressing illumination change and occlusion problems. In this paper, we present a tracking algorithm based on Edge Multi-channel Gradient Model. We first use the canny operator to extract the edges of the image, and establish a biologically inspired Multi-channel Gradient Model that integrate the spatio-temporal-spectral information of the edge to detect moving multiple objects. Under this model, the ORB feature is introduced to solve the problem of matching the object with the object library. Therefore, we can achieve object consistency, and the threshold classification method can solve the problem of multiple object occlusion in the process of persistent multiple object tracking. The experimental results show that the proposed method can effectively deal with the problems of occlusion and illumination changes. Compared with other state-of-the-art algorithms, the proposed algorithm achieves better performance on MOTA, MOTP, and IDF1. In particular, it performs best on IDSW on MOT2015 dataset, with an average improvement ratio of 28.99% over the second-place algorithm. In addition, our algorithm has a better performance in running time, achieving a good compromise between the speed and the accuracy.

INDEX TERMS Multiple object tracking, multiple object detection, ORB, E-McGM, tracking by detection.

I. INTRODUCTION

As an important task in computer vision, multiple object tracking (MOT) has extremely important applications in intelligent surveillance [1], [2], autonomous driving [3], medical diagnosis [4], and military vision guidance. Using intelligent technology to solve practical problems is becoming a trend [5]–[7]. However, MOT has to overcome more problems in tracking, such as occlusion and illumination. Occlusion is a common challenge in MOT. In generally, there are two types of occlusion, which are the occlusion of background objects and mutual occlusion between objects. Illumination change is another common challenge in MOT. Both of these challenges will affect the object feature acquisition, resulting in objects loss and objects mistracking.

Over the past decade, MOT focused on modeling and estimation of object trajectories. With the improvement of image feature expression ability, Tracking by Detection can use detection information to characterize the appearance of the object to be tracked, which greatly improves tracking accuracy. It has gradually become the mainstream MOT [8], [9]. According to the trajectory generation mode, Tracking by Detection method can be roughly divided into two categories: offline MOT and online MOT. The offline tracking algorithm can obtain the overall information of the continuous image sequence in advance and hence generally demonstrates excellent performance [10]–[13]. However, the offline MOT method cannot adapt to occasions with high real-time requirements, and can only be used in offline occasions such as video analysis and processing. In contrast, when the online tracking algorithm is running, the tracker can only receive the image information and detection results of the previous frame and the current frame. Due to the small amount of processed data, the online tracking method is suitable for the video systems with real-time requirements [14]–[16].

In addition, most of deep learning-based tracking by detection frameworks can yield accurate tracking results by using object detector with powerful feature representation

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

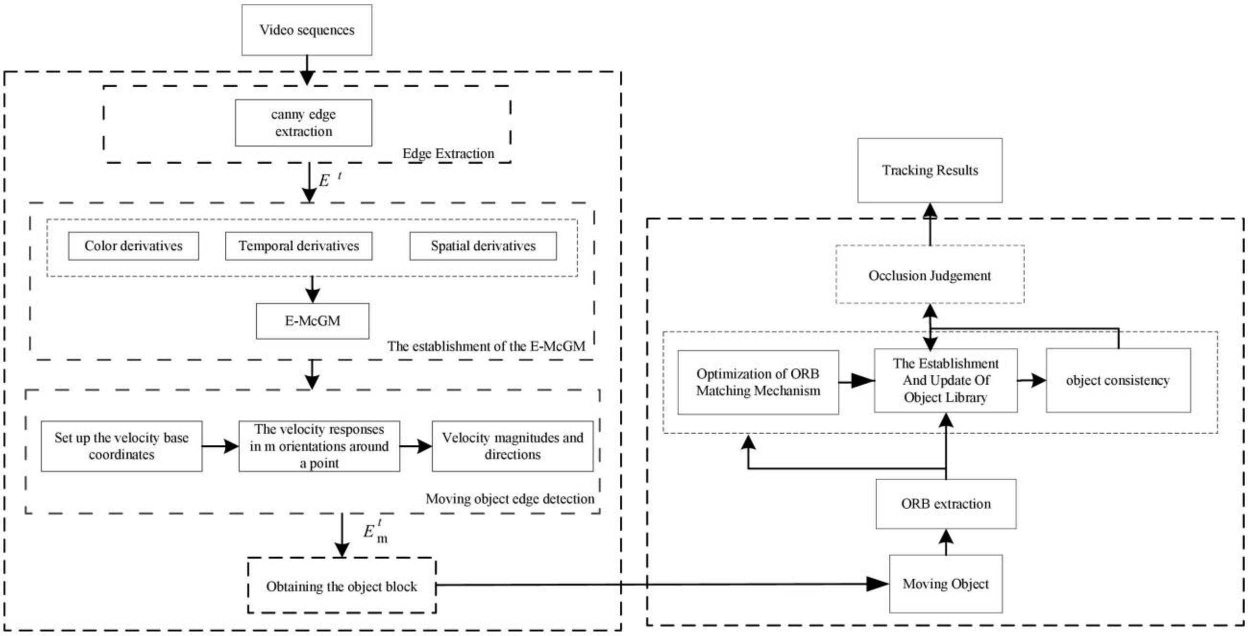

Video sequences
canny edge extraction
Edge Extraction
E^t
Color derivatives
Temporal derivatives
Spatial derivatives
E-McGM
The establishment of the E-McGM
Set up the velocity base coordinates
The velocity responses in m orientations around a point
Velocity magnitudes and directions
Moving object edge detection
E^t_m
Obtaining the object block
Tracking Results
Occlusion Judgement
Optimization of ORB Matching Mechanism
The Establishment And Update Of Object Library
object consistency
ORB extraction
Moving Object

FIGURE 1. **Flow chart of entire tracking process.**

capabilities [17]–[20]. However, these methods are mostly time-consuming and data-intensive. Even though

some deep learning methods can achieve accuracy and real-time performance, but still require dedicated hardware and consume a lot of power to achieve higher operating speeds. Additionally, deep learning methods have poor generalization performance under conditions such as illumination changes, object rotation, and object scaling. Therefore, some scholars began to use sift and surf features for object tracking [42], [43], which improved the generalization performance of tracking, but tracking was time-consuming. Rublee *et al.* [37] proposed ORB, a feature detection algorithm with rotation and scale invariance. While obtaining high-quality feature points, the speed is two orders of magnitude higher than that of sift, which is suitable for applications with higher real-time requirements.

In this paper, an online tracker is proposed for MOT. In the proposed method, a detector based on Edge Multi-channel Gradient Model (E-McGM) is used to detect the object bounding boxes in an image frame and combine ORB keypoints to achieve object consistency between multiple objects. Firstly, the Canny operator is used to obtain the edge information of the object in the video sequence, and the multi-channel gradient model of time, space and color are established on the object edge according to the color constancy of human vision. The model is then used to obtain the motion state description information of the object edge pixel and achieve the separation of the background edge and the edge of the moving object. Further, the discontinuous edges are connected by our edge connection method. Morphological processing of the connected complete contour can segment the object. The ORB keypoints are extracted according to the object block acquired by the object detection, and then the object area motion description information is combined with the ORB keypoints to establish the object library. Additionally, we introduce a method to address the mismatch between the ORB keypoints of the object and the object library during object consistency. Finally, different tracking strategies can be chosen according to different occlusion degree. The contribution of our paper is twofold and can be summarized as follows:

- In our detection framework, we propose the E-McGM model and use it for multiple object detection. To address the problem of incomplete object contour extraction in the process of multi-moving object detection, we use the E-McGM model to design a method for connecting the discontinuous edges of moving objects. The entire tracking process is presented as a flow chart in Figure.1.
- In our tracking framework, a new matching algorithm is proposed to improve the performance of matching between ORB and object library. A threshold classification method is proposed to address occlusion problems in persistent MOT.

The remainder of the paper is organized as follows. We first summarize the related works in Section II. Section III introduces the proposed E-McGM and multiple object detection based on E-McGM in detail. Section IV describes our tracking framework. Section V presents the experimental setup. The results are then described, and the merits of the proposed method are discussed. Finally, we present the conclusions from our work in Section VI.

II. RELATED WORK

According to the way of object initialization, almost all MOT methods can be divided into two categories: Detection-Free Tracking [21] and Tracking by Detection. Tracking by Detection framework is widely used in MOT [20], [22].

TABLE 1. Highlights and limitations of the existing algorithms.

Algorithms	Reference		Highlights	Limitations
offline	[23],[24],[25]		global optimization capabilities excellent performance in tracking	Unadaptable to occations with high real-time requirements
online	deep learning based methods	[17],[18],[20] [26],[27],[28]	high tracking accuracy	time-consuming, data -intensive dedicated hardware requirement
	Non -deep learnig based methods	[12],[29],[30], [31],[32]	No data training required, No hardware requirements	Tracking accuracy is not as good as deep learning methods
	biologically inspired algorithm	[33],[48]	high tracking accuracy	deep learning -based methods

The algorithm takes the detection result as the input hypothesis of the tracker. Because of the introduction of the detection module, the tracking algorithm is adaptable to changes in the number of objects and scale changes, Yu *et al.* [20] propose a pedestrian MOT algorithm (Person of Interest, POI) based on Faster RCNN. Tracking by Detection approach consists of two subgroups: offline tracking and online tracking. Offline tracking has global optimization capabilities. Zhang *et al.* [23] first propose to model the data association problem as a cost network flow with non-overlapping constraints and found the multi-objective global optimal trajectory association by solving the minimum cost flow in the network. Xi *et al.* [24] use the K shortest path to solve the data association. The association problem between trajectory and detection is modeled as a linear programming problem. Andriyenko *et al.* [25] formulate MOT as a continuous energy function minimization problem and find a strong local minimum by the conjugate gradient method to approximate the global optimal correlation solution.

Recently, with the development of object detection and further research on visual object appearance, more attention has been paid to mining strong appearance characteristics which can be obtained by extracting features of objects [44]–[46] and establishing robust appearance similarity measurement in online MOT. Deep learning has been used as the appearance model of MOT [26], [27], and compared with previous learning methods [28], it demonstrates better performance. Karunasekera et al [17] propose a method to address MOT by defining a dissimilarity measure based on object motion, appearance, structure and size. To enhance the performance in MOT, appearance feature and motion information are added in [18]. The POI algorithm combines the retrieval ideas in the pedestrian re-identification application and uses the Google Net network [20] to train on a large-scale pedestrian re-identification data set to obtain extraction parameters for the appearance of pedestrians. The test phase uses cosine distance to measure different pedestrians. The similarity between apparent features effectively improves the performance of MOT.

However, because of the requirement of a sizable amount of training data, deep learning methods cannot be used greatly in practical applications. The use of non-deep learning-based methods for MOT has always attracted the attention of researchers. Sugimura *et al.* [12] use the KLT tracker to track keypoints and then generate the target trajectory; Choi *et al.* [29] use the point features of the KLT tracker as incremental features to estimate the motion of the camera, which greatly improve the tracking performance. Some other methods use color histograms as features, such as Song [30], Mitzel [31]. Furthermore, by utilizing the information of spatial-temporal feature, the MOT framework is more robust and can deal with missed detection [32]. In addition, the object detection method combining biological vision and machine vision has gradually aroused the interest of researchers. For example, the RFBNet [33] algorithm inspired by the human visual perception system uses inception and cavity convolution to simulate the receptive field of human vision and combines the SSD detection network with the receptive field. The detection accuracy and operating efficiency of this algorithm are better than YOLO v3. A biologically inspired appearance model [47] was proposed for robust visual tracking, which achieve high tracking accuracy. Table 1 shows the highlights and limitations of the algorithms mentioned in Section II.

Furthermore, we propose a biologically inspired tracking algorithm without data training and dedicated hardware requirement. We show that the proposed algorithm can achieve better performance in both MOT2015 and MOT2017 datasets with real time running speed.

III. DETECTION FRAMWORK BASED ON E-McGM

Biology believes that the detection of multiple moving objects can be best described as changes in spatio-temporal-spectral gradients of the reflected energy of the object. We define a detection function equivalent to that of [34]. The results of E-McGM solved by Newton-Leibniz will be used as the constraint condition, thereby the obtaining detection of multiple moving objects.

A. THE ESTABLISHMENT OF E-McGM

We consider that different wavelengths of reflected light correspond to different spectral energy values [35]. Hence, we can obtain the color information of image using the spectral energy value. According to the visual optics theory of [35], we can get the description of each color channel of the object edge in the image I as Eq. (1).

$$S_i(\lambda_0) = \int_{\Omega} E(\lambda) G_i(\lambda; \lambda_0, \sigma) d\lambda, \quad i \in (0, 1, 2) \tag{1}$$

where λ is the wavelength. Ω is the range of all relevant wavelengths of the light source. $G_i(\lambda, \lambda_0, \sigma)$ is the sensitivity function with i derivative order, and a Gaussian function with mean $\lambda_0 = 520$nm and standard deviation $\sigma = 55$nm.

By the Hering opponent colors theory of human vision [36], we set $i = 2$ is the highest order value. Obviously, $G_0(\lambda, \lambda_0, \sigma)$, $G_1(\lambda, \lambda_0, \sigma)$ and $G_2(\lambda, \lambda_0, \sigma)$ are similar to the intensity, yellow-blue and red-green weighting functions found in the human visual system. S_0, S_1 and S_2 are interpreted as the responses of intensity, yellow-blue, and red-green receptive fields, respectively. $E(\lambda)$ represents the reflected energy distribution of the incident light. $E(\lambda)$ is defined as follow.

$$E(\lambda) = e(\lambda)R(\lambda) \tag{2}$$

where $e(\lambda)$ is the energy distribution of incident light, and it changes slowly when object moves. $R(\lambda)$ is the reflection coefficient of the object surface, and it does not change with the motion of the object. Therefore, $E(\lambda)$ can be regarded as an invariant, and $S_i(\lambda_0)$ has color constancy. That is, robust to changes in illumination.

S_0 describes the spectral energy of the edge of the object, which can be approximated by the second-order Taylor expansion in the neighborhood of λ_0. It can be written as follow.

$$\begin{aligned}\hat{S}(\lambda_0 + s; E, \sigma) &= S_0(\lambda_0; E, s) + sS_1(\lambda_0; E, \sigma)\\ &\quad + \frac{s^2}{2} S_2(\lambda_0; E, \sigma) + O(s^3)\end{aligned} \tag{3}$$

s is the excursion parameters in spectrum. Eq. (3) indicates a color element in our work. The spectral energy about a point $P(x,y,t,\lambda)$ in the edge of color image can be approximated by Taylor expansion again, which is defined by Eq. (4).

$$\begin{aligned}&\hat{S}(x + p, y + q, t + r, \lambda + s)\\ &\quad = \sum_{i=0}^{e}\sum_{j=0}^{f}\sum_{k=0}^{g}\sum_{l=0}^{h} \frac{p^i q^j r^k s^l}{i!j!k!l!}\\ &\quad \cdot \frac{\partial^{(i+j+k+l)} S(x, y, t, \lambda)}{\partial x^i \partial y^j \partial t^k \partial \lambda^l}, \quad (x, y \in E^t)\end{aligned} \tag{4}$$

where E-McGM corresponding to the image edge pixel $P(x,y,t,\lambda)$ can be established. (p, q, r, s) are the excursion parameters in two orthogonal spatial directions, time and spectrum. This expansion yields a set of gradient measures up to order e, f, g, h in the four dimensions. E^t is the edge image of the t-th frame obtained by the Canny operator. We set $e = 3, f = 2, g = 2, h = 2$. Here, Eq. (4) can be viewed as the local edge geometry due to the combination of spatial and temporal information.

B. EDGE DETECTION OF MULTIPLE MOVING OBJECTS

We define objective speed function which can be described as Eq. (5).

$$\frac{C'}{\sqrt{A'^2 + B'^2}} \tag{5}$$

in which,

$$\begin{aligned}C' &= C + C_s,\\ A' &= A + A_s,\\ B' &= B + B_s,\end{aligned} \tag{6}$$

where C' represents the change in the amount of light over the temporal interval. A' and B' represent the change in the amount of light over some spatial interval. C represents the differences in the total image intensity over the temporal interval. C_s represents the differences in the total image opponent colors over the temporal interval. A and B represent the differences in the total image intensity over the spatial interval. A_s and B_s represent the differences in the total image opponent colors over the spatial interval.

We group the same terms of partial derivative order with respect to space, time, and wavelength in (4) to form a vector k $(x,\ y,\ t,\ \lambda)$, which represents the edge structure of the image. Combining the Newton-Leibniz formula, we can get the image intensity and color differences in temporal interval and spatial interval. We differentiate each component image of the vector $k(x,\ y,\ t,\ \lambda)$ corresponding to a given point $P(x,y,t,\lambda)$, and the result is shown as M.

$$M = \mathrm{D}k(x, y, t, \lambda) = (K_x, K_y, K_t, K_\lambda) \tag{7}$$

where D is the difference operator. We consider the orientations columns found in the primary visual cortex, then we introduce the speed $\widehat{s}$ and the inverse speed $\check{s}$ measures that can be used to compute speed for a range of special orientations. The whole process can be described as follow.

$$M^T M = \begin{bmatrix} K_x \cdot K_x & K_x \cdot K_y & K_x \cdot K_t & K_x \cdot K_\lambda \\ K_y \cdot K_x & K_y \cdot K_y & K_y \cdot K_t & K_y \cdot K_\lambda \\ K_t \cdot K_x & K_t \cdot K_y & K_t \cdot K_t & K_t \cdot K_\lambda \\ K_t \cdot K_x & K_t \cdot K_y & K_t \cdot K_t & K_t \cdot K_\lambda \end{bmatrix} \tag{8}$$

The direction of the element in Eq. (8) depends upon the direction of motion. Eq. (8) is then integrated over a spatio-temporal-spectral region which are the changes in the amount of light over the temporal interval and spatial interval.

$$\begin{aligned}&\int_{-s}^{s}\int_{-r}^{r}\int_{-q}^{q}\int_{-p}^{p} M^T M dxdy\, dtd\lambda\\ &= \begin{bmatrix} X \cdot X & X \cdot Y & X \cdot T & X \cdot \lambda \\ Y \cdot X & Y \cdot Y & Y \cdot T & Y \cdot \lambda \\ T \cdot X & T \cdot Y & T \cdot T & T \cdot \lambda \\ \lambda \cdot X & \lambda \cdot Y & \lambda \cdot T & \lambda \cdot \lambda \end{bmatrix}\\ &= \begin{bmatrix} A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_2 & B_3 & C_3 & D_3 \\ A_4 & B_4 & C & D_4 \end{bmatrix}\end{aligned} \tag{9}$$

The speed with m different orientations θ can be computed by rotating coordinate system. $\widehat{s}$ for a coordinate system with orientation θ is expressed in Eq. (10) below.

$$\widehat{s} = [\frac{C_1 + C_4}{\sqrt{(A_1 + A_4)^2 + (B_1 + B_4)^2}} \cos(\alpha), \frac{C_2 + C_4}{\sqrt{(A_2 + A_4)^2 + (B_2 + B_4)^2}} \sin(\alpha)] \tag{10}$$

where α is orientation of local image contours with respect to the coordinate frame θ. $\widehat{s}$ is computed at m different orientations θ around a point in I. Eq. (10) can be re-written as $\widehat{s} = (\widehat{s}_\parallel, \widehat{s}_\perp)$ in Eq. (11) in a way that allows well-conditioned calculation.

$$\begin{aligned} \widehat{s}_\parallel &= \sqrt{\frac{2}{m}} \left[\frac{C_1 + C_4}{A_1 + A_4} (1 + (\frac{B_1 + B_4}{A_1 + A_4})^2)^{-1} \right] \\ &= \sqrt{\frac{2}{m}} \left[\frac{\lambda \times T + X \times T}{\lambda \times X + X \times X} (1 + (\frac{X \cdot Y + \lambda \cdot Y}{X \cdot X + \lambda \cdot X})^2)^{-1} \right] \\ \widehat{s}_\perp &= \sqrt{\frac{2}{m}} \left[\frac{C_2 + C_4}{B_2 + B_4} (1 + (\frac{A_2 + A_4}{B_2 + B_4})^2)^{-1} \right] \\ &= \sqrt{\frac{2}{m}} \left[\frac{\lambda \times T + Y \times T}{\lambda \times Y + Y \times Y} (1 + (\frac{Y \cdot X + \lambda \cdot X}{Y \cdot Y + \lambda \cdot Y})^2)^{-1} \right] \end{aligned} \tag{11}$$

where $\widehat{s}_\parallel$ and $\widehat{s}_\perp$ are the first and second column of $\widehat{s}$, and they are the parallel and orthogonal to the primary direction, respectively. Similarly, inverse speed can be obtained based on Eq. (12).

$$\begin{aligned} \check{s}_\parallel &= \sqrt{\frac{2}{m}} \left[\frac{A_3 + A_4}{C_3 + C_4} \right] = \sqrt{\frac{2}{m}} \left[\frac{T \cdot X + \lambda \cdot X}{T \cdot T + \lambda \cdot T} \right] \\ \check{s}_\perp &= \sqrt{\frac{2}{m}} \left[\frac{B_3 + B_4}{C_3 + C_4} \right] = \sqrt{\frac{2}{m}} \left[\frac{T \cdot Y + \lambda \cdot Y}{T \cdot T + \lambda \cdot T} \right] \end{aligned} \tag{12}$$

Substituting (11) and (12) into (13), the speed of $P(x,y,t,\lambda)$ can be obtained:

$$S^2 = \frac{\begin{vmatrix} \widehat{s}_\parallel \cdot \cos\theta & \widehat{s}_\parallel \cdot \sin\theta \\ \widehat{s}_\perp \cdot \cos\theta & \widehat{s}_\perp \cdot \sin\theta \end{vmatrix}}{\begin{vmatrix} \widehat{s}_\parallel \cdot \check{s}_\parallel & \widehat{s}_\parallel \cdot \check{s}_\perp \\ \widehat{s}_\perp \cdot \check{s}_\parallel & \widehat{s}_\perp \cdot \check{s}_\perp \end{vmatrix}} \tag{13}$$

Substituting (11) and (12) into (14), the orientation of $P(x,y,t,\lambda)$ can be shown as follow:

$$\Theta = \arctan \frac{(\widehat{s}_\parallel + \check{s}_\parallel) \cdot \sin\theta + (\widehat{s}_\perp + \check{s}_\perp) \cos\theta}{(\widehat{s}_\parallel + \check{s}_\parallel) \cdot \cos\theta + (\widehat{s}_\perp + \check{s}_\perp) \sin\theta} \tag{14}$$

From (13) and (14), we can obtain the speed description of $P(x,y,t,\lambda)$. Further, our method achieves to extract the edge of the moving objects and obtain the moving objects edge map E_m^t.

C. EDGE CONNECTION

The edges of the image are discontinuous in E_m^t, and these affect the acquisition of the objects block. Based on the proposed E-McGM model mentioned in section II.A, we design a discontinuous edge connection method through the following steps:

1) We use mark w to record the state of the point. When the point is dynamic, the mark w is 1, otherwise is 0. $\boldsymbol{P}^t = \left\{ p^t_{\mathrm{x1,y1}}, p^t_{\mathrm{x2y2}}, \ldots, p^t_{\mathrm{xc,yc}} \right\}$ is a set of edge point in E_m^t, and we initialize all w corresponding to elements in P^t to 1.

2) We pick the n-th point $p^t_{xn,y\,n}$ in the set $\boldsymbol{P}^t$, and create a dynamic array $\zeta^t_{xn,yn}$ belonging to $p^t_{xn,y\,n}$, and store $p^t_{xn,y\,n}$ in the array as the first element:

$$\xi^t_{xn,yn} = \{\xi_0\} \tag{15}$$

We set a variable, *count* which denotes the initial value of the number of points in $\zeta^t_{xn,y\,n}$, and let it start from 1; We set $MS^t_{xn,y\,n}$ and $M\theta^t_{xn,yn}$ are the average speed and average direction of all points in the $\zeta\, t\; xn,yn$, respectively. The initial values of $MS^t_{xn,y\,n}$ and $M\theta^t_{xn,yn}$ are the speed and direction of ζ_0.

3) We set ζ_0 in $\zeta^t_{xn,y\,n}$ on the Canny edge map E^t, and establish a 3×3 neighborhood with ζ_0 as center, and we obtain eight neighborhood points e_i, $i \in [0, 7]$. If the neighborhood point satisfies $e_i \in E^t$ and $e_i \notin E^t_m$, we will determine whether the point is a missing dynamic edge point by Eq. (16).

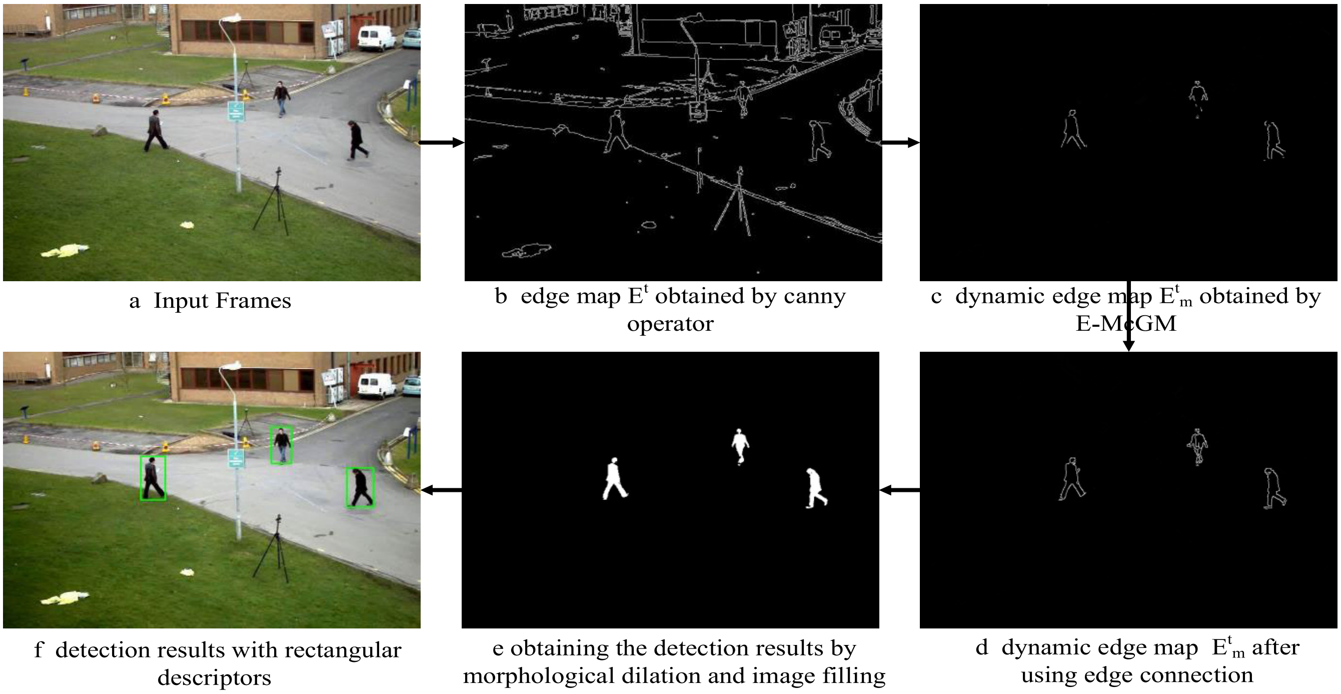
$$w = \begin{cases} 1, & if \left| S^t_{e_i} - MS^t_{xn,yn} \right| < T_s \text{and} \left| \Theta^t_{e_i} - M\theta^t_{xn,yn} \right| < T_\theta \\ 0, & \text{else} \end{cases} \tag{16}$$

where $S^t_{e_i}$ is the speed of e_i, and Θte_i is the direction of e_i. T_s and T_θ are the threshold of speed and direction, respectively. We set T_s =0.18, $T_\theta = \pi/36$. If $w = 1$, e_i is considered as a point in dynamic edge, and added to $\zeta^t_{xn,y\,n}$ and E^t_m, and $count = count + 1$.

4) We update $MS^t_{xn,yn}$ and $M\theta^t_{xn,yn}$ by Eq. (17) and delete $p^t_{xn,y\,n}$ and $count = count$-1. If $count \neq 0$, then we extract a new ξ_0 from $\zeta^t_{xn,y\,n}$ and return to step 3). Otherwise, let $n = n + 1$ and repeat step 2) until all elements in the set P^t are extracted. A summary of the complete algorithm is given as Algorithm 1 below in the pseudocode.

$$\begin{aligned} MS^t_{xn,yn} &= \frac{\sum_{i=1}^{count} S^t_{e_i}}{count} \\ M\theta^t_{xn,yn} &= \frac{\sum_{i=1}^{count} \theta^t_{e_i}}{count} \end{aligned} \tag{17}$$

Through the above steps, we can get the complete contour of the moving object and detection results by the morphological dilation and image filling the complete contour. Furthermore, we use different bounding box to mark the different object block, respectively. The proposed entire moving object detection algorithm process is illustrated in Figure 2.



a Input Frames

b edge map E^t obtained by canny operator

c dynamic edge map E^t_m obtained by E-McGM

d dynamic edge map E^t_m after using edge connection

e obtaining the detection results by morphological dilation and image filling

f detection results with rectangular descriptors

FIGURE 2. Flow chart of target detection algorithm.

Algorithm 1 Edge Connection Method in E-McGM

```
Algorithm: discontinuous edge connection
Input. $Et\ m$, $E^t$, $T_s = 0.18$, $T_\theta = \frac{\pi}{36}$, $S$, $\theta$
Output. updated $E^t_m$
 1. for $Pt\ xn,yn$ in $\boldsymbol{P}^t$ do:
 2. $\xi^t_{xn,yn} = \{\xi_n\}$; $\xi_n = P^t_{xn,yn}$; $count$++;
 3. $MS^t_{xn,yn} = \frac{\sum_{i=1}^{count} S^t_{e_i}}{count}$, $M\theta^t_{xn,yn} = \frac{\sum_{i=1}^{count} \theta^t_{e_i}}{count}$
 4. for $e_i$ in $E^t$ 3×3 neighbors($Pt\ xn,yn$) do:
 5. if $e_i \notin E^t_{\mathrm{m}}$&& $\left|S^t_{e_i} - MS^t_{xn,yn}\right| < T_s$&& $\left|\Theta^t_{e_i} - M\theta^t_{xn,yn}\right| < T_\theta$ then:
 6. $P^t$ insert($e_i$), $W_p = 1$; $\xi^t_{xn,yn}$ insert($e_i$) $count$++;
 7. end
 8. delete $Pt\ xn,yn$ in $P^t$; delete $\xi_n$ in $\xi^t_{xn,yn}$; $count$–;
 9. if $count = 0$ then: $n = n + 1$;continue;
10. else $Pt\ xn,yn = \xi_0$ do; goto 3;
11. end
```

IV. TRACKING FRAMEWORK

In this paper, we propose a MOT algorithm using ORB keypoints [37], which can be extracted from the object block in section III. In this algorithm, we establish an object library. The matching rate function is used to realize the object consistency between the object library and the object, and the multiple object occlusion problem is solved by the classification threshold method.

A. OBJECT LIBRARY

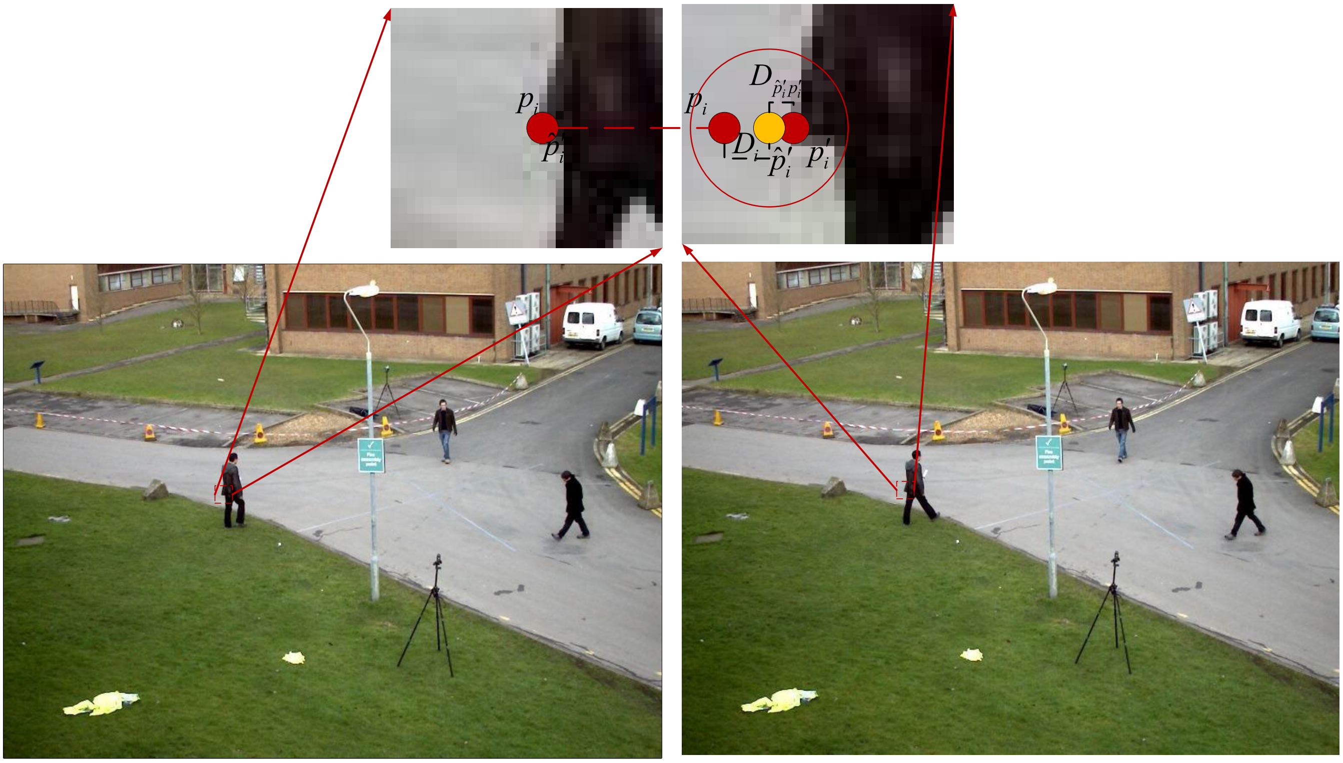
We define an object library set $\boldsymbol{H}$ as presented in Eq. (18). The keypoints in every object block matched with $\boldsymbol{H}$. We update $\boldsymbol{H}$ by limiting the threshold of $\boldsymbol{K}_j$ in each frame.

$$\boldsymbol{H} = \{C_i, S_i, \omega_i, V_{\omega_i}, P_j, V_j, D_j, K_j\} \tag{18}$$

in which:

$$\begin{aligned}
&\boldsymbol{C}_i = \{C_1, C_2, \cdots, C_i \cdots, C_z\}\\
&\boldsymbol{S}_i(W,H) = \{S_1(W,H), S_2(W,H), \cdots,\\
&\qquad\qquad S_i(W,H), \cdots, S_z(W,H)\}\\
&\boldsymbol{\omega}_i(x_i, y_i) = \{\omega_1(x_1, y_1), \omega_2(x_2, y_2), \cdots,\\
&\qquad\qquad \omega_i(x_i, y_i), \cdots, \omega_z(x_z, y_z)\}\\
&\boldsymbol{V}_{\omega_i} = \left\{V_{\omega_1}, V_{\omega_2}, \cdots, V_{\omega_i}, \cdots, V_{\omega_z}\right\}\\
&\boldsymbol{P}_j = \left\{P_1, P_2, \cdots, P_j, \cdots, P_q\right\},\\
&\qquad (i \in [1, 2, \ldots, z], z \in \mathbf{Z}^+)\\
&\boldsymbol{V}_j = \left\{V_1, V_2, \cdots, V_j, \cdots, V_q\right\}\\
&\qquad (j \in [1, 2, \ldots, q], q \in \mathbf{Z}^+)\\
&\boldsymbol{D}_j = \left\{D_1, D_2, \cdots, D_j, \cdots, D_q\right\}\\
&\boldsymbol{K}_j = (\tau_1, \tau_2, \cdots, \tau_j, \cdots, \tau_q)
\end{aligned} \tag{19}$$

where i is the number of objects in the object library and j is the number of keypoints of each object block. C_i is ID of i-th object. S_i is the size of object block. $\omega_i(x_i, y_i)$ is centroid position of each object block in the previous frame. $\boldsymbol{V}_{\omega i}$ is the speed set of $\omega_i(x_i, y_i)$. $\boldsymbol{P}_j$ is a keypoints set extracted from object $\boldsymbol{C}_i$. $\boldsymbol{V}_i$ and $\boldsymbol{D}_j$ are speed and derection set of the set $\boldsymbol{P}_j$, respectively. $\boldsymbol{K}_j$ is the update levels for each keypoint in $\boldsymbol{P}_j$. When keypoint in current frame does not match the keypoint in $\boldsymbol{P}_j$ for τ_j consecutive frames, we will update the keypoint in $\boldsymbol{P}_j$. In this paper, we let $\tau_j = 3$.



(a)

(b)

FIGURE 3. Matching schematic (a) The keypoint p_i in the previous frame; (b) the matching point p_i of p_i in the current frame.

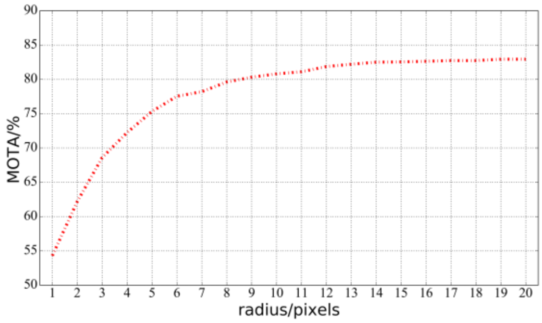


FIGURE 4. The relationship between different values of Ψ and MOTA.

B. E-McGM-BASED MATCHING METHOD

We obtain stable tracking by the ORB features, and we propose a method to match results the object with the object library. Experiments show that this method can significantly reduce mismatches compared with the previous method. This method steps are as follows:

1) We define a point $p_i(x_i, y_i)$ from P_j and a point $p_i'(x_i', y_i')$ from object O, which is a pair of matching points with the smallest Hamming distance.

2) We can calculate speed V_i of the keypoint p_i in the object library by Eq. (11). Distance between two points from two adjacent frames is given in Eq. (21), where $\hat{p}_i'(\hat{x}_i', \hat{y}_i')$ is the predicted position of $p_i(x_i, y_i)$ in the current frame.

$$\hat{p}_i'(\hat{x}_i', \hat{y}_i') = (x_i + dx_i, y_i + dy_i) \tag{20}$$

where

$$D_i(dx_i, dy_i) = V_i \cdot T_{interframe} \tag{21}$$

In Eq. (21), $T_{interframe}$ is time of interframe.

3) The distance D_d between $\hat{p}'$ and p_i' is calculated using the Euclidean distance equation. Considering the rotation and deformation of the object, we make a circle with $\hat{p}'$ as the center and ψ as radius. Function υ is used to determine whether $p_i(x_i, y_i)$ and $p_i'(x_i', y_i')$ are a pair of accurate matching points:

$$\upsilon = \begin{cases} 1, & \text{if}(D_d < \psi) \\ 0, & \text{if}(D_d > \psi) \end{cases} \tag{22}$$

As shown in Figure.3, when point p_i' is in the circle, $p_i(x_i, y_i)$ and $p_i'(x_i', y_i')$ are considered to be a pair of correct matching points. However, there are a few mismatches, and it does not have a great impact for our tracking. We let $\Psi = 14$ due to the relationship between Ψ and MOTA, which can be visualized by Figure.4 and Table 2. According to Figure.4 and Table 2, when $\Psi = 14$, the value of MOTA will not change and the number of ORB matching points have little change.

TABLE 2. Video sequence parameters.

Ψ	1	2	3	4	5	6	7	8	9	10
ORB matching point	304	345	377	393	403	416	430	446	452	453
MOTA	54.30%	62.19%	68.60%	72.25%	75.33%	77.54%	78.21%	79.62%	80.32%	80.81%
Ψ	11	12	13	14	15	16	17	18	19	20
ORB matching point	453	454	455	455	456	457	457	457	457	457
MOTA	81.11%	81.87%	82.21%	82.55%	82.55%	82.55%	82.55%	82.56%	82.56%	82.56%

Figure.5 illustrate a comparison between original method and matching method based on E-McGM.

C. OBJECT CONSISTENCY

We define the set $\boldsymbol{S} = \{M_{c1}, M_{c2},\ldots, M_{ci},\ldots, M_{cl}\}$, $i \in\{1,2,\ldots, l\}$, $l \in \mathbf{Z}^{+}$. M_{ci} denote the number of that the keypoints extracted from object O match the keypoints of object C_i. The matching rate function is defined as follows.

$$F = \frac{M_{C_i}}{O_{\text{all}}} \tag{23}$$

where O_{all} is the number of keypoints extracted from the O in the current frame. We choose C_i with the highest F value as the ID of O.

D. OCCLUSION HANDLING

Occlusion is a common problem in MOT. In this paper, we solve this problem by threshold classification processing method.

According to the velocity of centroid in the last frame of the moving object obtained by E-McGM, we can calculate the predicted position $(\hat{x}_i, \hat{y}_i)$ of the centroid in the current frame, and the observed position $(\tilde{x}_i, \tilde{y}_i)$ of the centroid is calculated by the centroid formula. Then we use Eq. (24) to determine whether the object is occluded.

$$r_i = \sqrt{(\tilde{x}_i - \hat{x}_i)^2 + (\tilde{y}_i - \hat{y}_i)^2} \tag{24}$$

We define R_c is the size of the object search window in the current frame, and R_0 is the size of the search window when the object first appears. The occlusion rate Λ is defined as Eq. (25):

$$\Lambda = \frac{R_c}{R_o} \tag{25}$$

In the proposed method, ζ is the threshold of r_i. Λ_h and Λ_l are the maximum occlusion probability and the minimum occlusion probability, respectively. Combining Eq. (24) and (25), the threshold classification processing steps are as follows:

Case1: When $r_i < \zeta$ and $\Lambda_h \leq \Lambda \leq 1$, it means that slight occlusion may be encountered in the process of tracking, but it has no effect on the tracking result. We can obtain the object ID in data association.

Case2: When $r_i > \zeta$ and $\Lambda_l \leq \Lambda < \Lambda_h$, the detection of mutually obscuring objects needs to be further segmented for this situation. Steps are shown below.

1) Extracting the ORB keypoints $P_o, o \in(0, r)$ in the object block, and matching it with the object library to obtain the identification of each keypoint.

2) The motion description corresponding to the classified ORB keypoints P_i can be obtained through E-McGM. Further use least square regression to get the speed S_{Ci} and θ_{Ci} direction of each category.

3) If the edge points of object in set $P^e = \{p_1^e, p_2^e, \ldots, p_i^e, \ldots, p_1^e, p_2^e, \ldots, p_n^e, \ldots\}$, $i \in(0, n)$ obtained in the detection stage satisfie $\left|S^{P_i^e} - S^{C_i}\right| < T^s \&\& \left|\theta^{P_i^e} - \theta^{C_i}\right| < T^\theta$, p_i^e and P_i belong to the same object. Among them, $S_{P_i^e}$ and $\theta_{P_i^e}$ are the speed and direction of P_i^e. T^s and T^θ are the thresholds and their values are $T^s = 0.9$ and $T^\theta = \pi/42$.

After the segmentation of the outer contour of the object block is completed, the minimal outer rectangle is made to generate the final object tracking area.

Case3: When $r_i \geq \zeta$ and $0 \leq \Lambda < \Lambda_l$, we use the previous frame of motion description information obtained by E-McGM to predict and track the object in the current frame. We assume that the object is in a state of uniform motion, and the object is not updated until the end of the heavy occlusion state. The specific tracking process is as follows:

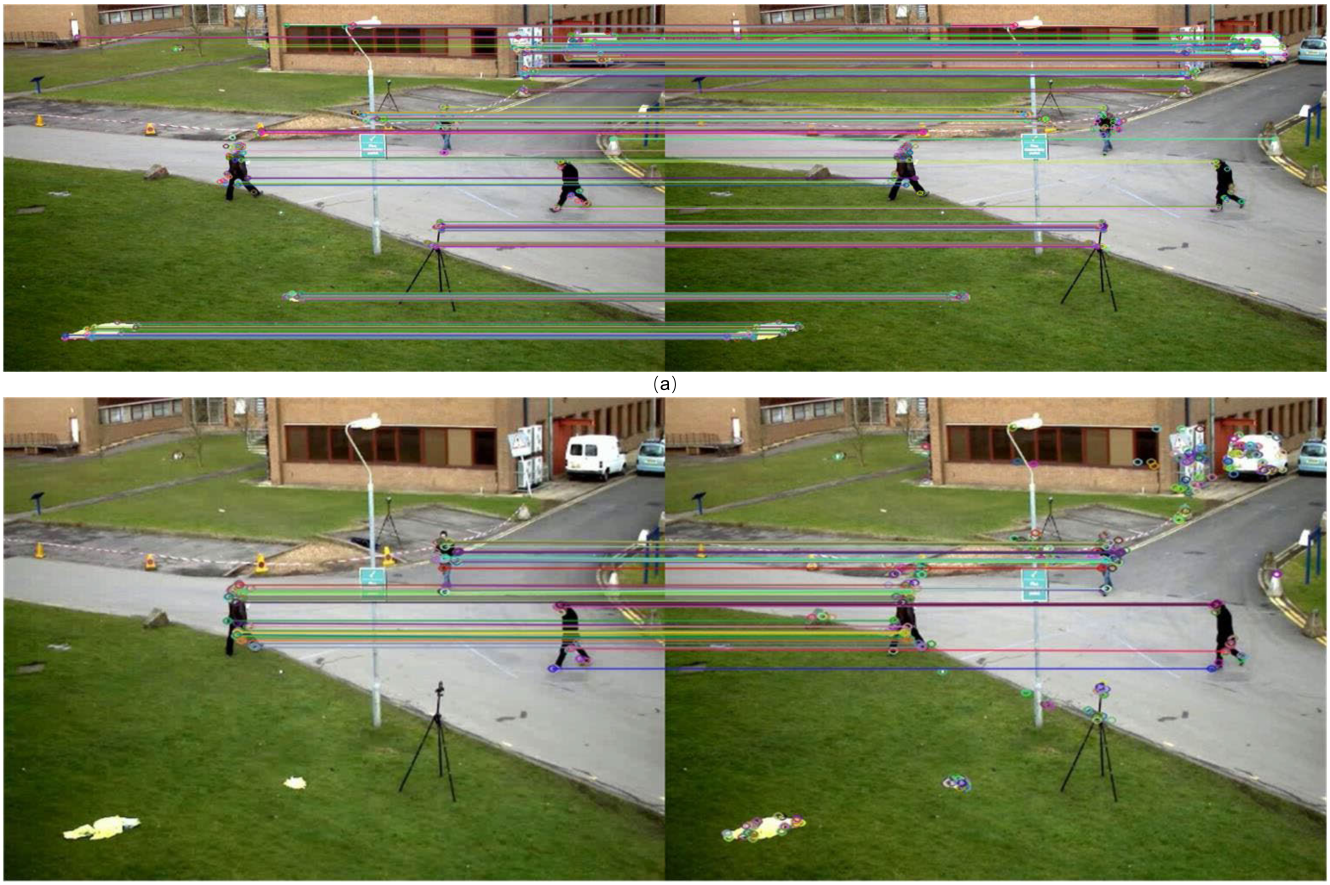
1) $D_i = V_{\omega i} \cdot T$ is used to estimate the centroid position of the object in the current frame.

2) The center of the object box in the previous frame is moved to the centroid position of the current frame. Identification of the object is the same as the previous frame.

We let $\zeta = 10$, $\Lambda_h = 95\%$, and $\Lambda_l = 5\%$.

V. EXPERIMENTAL RESULTS

The proposed method is implemented using C++ and OpenCV library and all the cases presented in this section are done in a desktop with 64-bit Ubuntu16.04 operating system, Intel Core Xeon e5-2603 CPU processor, 16GB memory of the personal computer. In this section, firstly, E-McGM is


(a)

(b)

FIGURE 5. Matching results of ORB (a) is the matching result without our method (b) is the result using our proposed method.

evaluated on the image verification to demonstrate the performance of detection. Secondly, a tracking framework based on E-McGM is applied to the publicly available MOT15 and MOT17 benchmark for comparison with various state-of-the-art algorithms.

A. DETECTION

The aim of the experiments was to test if the E-McGM achieves competitive performance. The blizzard and PETS2006 video sequences of the CDnet2014 dataset [38] and the David sequence of the visual tracker benchmark are selected to verify the effectiveness of the algorithm in this paper. As listed in Table 3, the challenges of them and video parameters are summarized.

In our experiments, we compare our detector with other state-of-the-art methods, which are FCB [39], MoGG [40], and SGF [41], respectively. To conduct a quantitative comparison between the proposed method and other approaches, we use the evaluation metrics provided by the CDnet dataset, which are Recall (R), Precision (P) and F- measure (F_M). The comparison results are visualized in Table 4 and Figure 6.

Note that the proposed method does not lead to the best precision and recall in sequence PETS2006, but it effectively balances the false positive rate and false negative rate. Good performance was achieved in sequences blizzard and David with complex backgrounds, reflecting the accuracy of algorithm detection.

TABLE 3. Relationship between Ψ and MOTA.

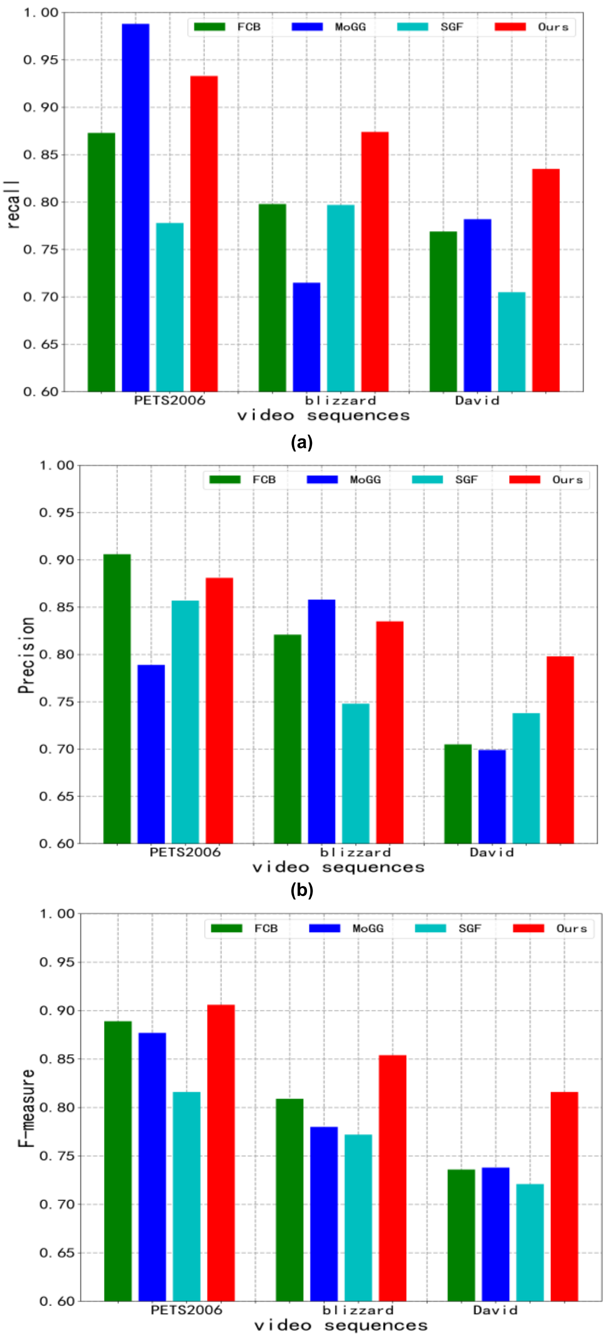
Video sequence	Frames	Size	Occ	Scaling	Ill	Dynamic background
PETS2006	1200	720×576	√	√	×	×
blizzard	7000	720×480	×	√	×	√
David	770	320×240	×	√	√	×

TABLE 4. Performance comparison of different algorithms.

	Video 1			Video 2			Video 3		
	R(%)	P(%)	F_M(%)	R(%)	P(%)	F_M(%)	R(%)	P(%)	F_M(%)
FCB [39]	87.3	**90.6**	88.9	79.8	82.1	80.9	76.9	70.5	73.6
MoGG [40]	**98.8**	78.9	87.7	71.5	**85.8**	78.0	78.2	69.9	73.8
SGF [41]	77.8	85.7	81.6	79.7	74.8	77.2	70.5	73.8	72.1
Ours	93.3	88.1	**90.6**	**87.4**	83.5	**85.4**	**83.5**	**79.8**	**81.6**

B. TRACKING METHOD BASED ON E-McGM

We conducted all these experiments in MOT15, MOT17, Urban Tracker and Visual Tracker Benchmark and its



(a)

(b)

(c)

FIGURE 6. Experimental results of different algorithms. (a) Recall; (b) Precision; (c) F-measure.

performance was compared to MOT methods which are MOT_LST [32], MASS [17], POI [20] and OMOT_JD [19], FPSN [18].

1) DATASETS AND EVALUATION METRICS

In order to verify the effectiveness of the algorithm, the test video sequences of the four datasets include different scenarios indoor and outdoor, as well as rigid and non-rigid targets. The video sequences on the datasets MOT15 and MOT17 mainly track pedestrians in outdoor scenes. The video sequences in the dataset Urban Tracker can be used to track indoor and outdoor pedestrians and vehicles. The video sequences selected in the dataset Visual Tracker mainly test the effectiveness of the algorithm in this paper when the illumination change. All video sequences and parameters are shown in Table 5.

TABLE 5. All video sequences and basic parameters.

Video Databases	Video Sequences	Frame Size	Frame Count
CDnet2014	PETS2006	720×576	1200
	blizzard	720×480	7000
	David	320×240	770
MOT15	TUD-Stadtmitte	640×480	179
	PES09-S2L1	768×576	795
	TUD-Campus	640×480	71
	ADL-Rundle-6	1920×1080	525
	KITTI-17	1224×370	145
MOT17	MOT17-01-DPM	1920×1080	450
	MOT17-08-DPM	1920×1080	625
Visual Tracker Benchmark	Man	240×192	134
	Car_1	640×360	913

To provide a concrete comparison, the results of all the competing methods are summarized in terms of the same metrics suggested in Eq. (26), (27) and (28) [48].

$$MOTA = 1 - \frac{FN + FP + IDSW}{GT} \tag{26}$$

where FN denotes the total number of missed objects in the video. FP denotes the total number of falsely detected objects. IDSW denotes the number of target identity switches, and GT denotes the total number of true objects.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{27}$$

where $d_{t,i}$ is the bounding box overlap between the hypothesis I with its assigned ground truth object. c_t denotes the number of matches in frame t.

$$\begin{aligned} \mathrm{IDF}_1 &= \frac{2\mathrm{IDTP}}{2\mathrm{IDTP}+\mathrm{IDFP}+\mathrm{IDFN}} \\ \mathrm{IDTP} &= \sum\nolimits_t len - \mathrm{IDFN} = \sum\nolimits_c len - \mathrm{IDFP} \end{aligned} \tag{28}$$

where IDFP is the total number of objects that the ground truth trajectories and the calculated trajectories do not match. IDFP is the total number of objects not matched between the calculated trajectory and the ground truth trajectories. $\sum_t len$ is the total number of real objects in all video frames. $\sum_c len$ is the total number of targets in all video frames. MOTP focuses on the quality of detection, MOTA focuses on the quality of tracking, and IDF1 focuses on the comparison between the calculated trajectory and the ground truth trajectory.

TABLE 6. Comparison of results on different tracking methods on MOT15.

Sequence	Method	MOTA (%)↑	MOTP (%)↑	IDF1 (%)↑	FN↓	FP↓	IDSW ↓
TUD-staditimtte	MOT_LST [32]	66.3	50.8	68.4	206	168	16
	MASS [17]	60.7	76.5	63.9	256	185	13
	POI [20]	**80.6**	**87.4**	**88.1**	**162**	**48**	14
	OMOT_JD [19]	66.9	70.3	70.3	231	133	19
	ours	76.8	80.7	78.4	196	65	**7**
PES09-S2L1	MOT_LST [32]	82.86	70.8	84.6	442	284	48
	MASS [17]	78.4	87.1	82.3	675	248	44
	POI [20]	**87.5**	**91.6**	**89.5**	**369**	**163**	28
	OMOT_JD [19]	74.2	81.8	77.7	705	363	87
	ours	83.8	88.1	86.5	487	230	**8**
TUD-campus	MOT_LST [32]	58.4	42.9	60.9	94	47	8
	MASS [17]	58.3	74.7	61.8	106	39	5
	POI [20]	**76.2**	**83.5**	**77.4**	**56**	**26**	**3**
	OMOT_JD [19]	60.6	66.2	63.6	103	32	6
	ours	73.4	72.7	77.0	61	31	**3**
ADL-Roudle-6	MOT_LST [32]	68.2	54.5	70.4	1015	493	85
	MASS [17]	77.9	80.3	80.1	886	154	67
	POI [20]	**84.6**	**89.2**	**86.5**	**592**	**136**	**43**
	OMOT_JD [19]	64.7	72.4	67.2	1345	329	94
	ours	76.8	82.5	78.5	850	214	98
KITTI-17	MOT_LST [32]	61.7	50.9	64.6	170	79	13
	MASS [17]	70.0	85.5	86.8	140	55	10
	POI [20]	**86.4**	**90.1**	**88.3**	**70**	17	**6**
	OMOT_JD [19]	68.1	76.5	70.5	159	46	13
	ours	83.1	88.3	87.9	93	**15**	7

2) COMPARISON WITH OTHERS

Table 6 shows the comparison results of tracking performance data on the MOT15 dataset between the proposed algorithm and other algorithms.

As shown in Table 6, in terms of IDF1, our algorithm is the next best to POI, which is based on deep learning and significantly outperforms the majority of referenceable methods on most evaluation metrics including MOTA, MOTP and IDF1. The reason is that POI explores the high-performance detection and deep learning-based appearance feature which play the key role in data association-based MOT. The results also show that our method does not reach top performance on other sequences but is the second best among all methods except in ADL-Roudle-6. However, in TUD-staditimtte, PES09-S2L1 and KITTI-17, the IDF1 performance of our method is 1.1%-8.1% higher than the third best method.

In terms of IDSW, the proposed method achieves the best performance in PES09-S2L1, TUD-staditimtte and TUD-campus, which indicate the effectiveness of our algorithm for object consistency. In ADL-Roudle-6, the proposed method is ranked in fifth place due to long-term severe occlusion, ORB keypoints cannot be updated for a long time.

Although FN is the next best to POI, the number is very close to the POI. The proposed algorithm in this paper is for moving object tracking, but when the objects suddenly stop, the result of FN will be affected, such as PES09-S2L1.

Table 6 also record that a lowest number of FP in the KITTI-17 sequence and a low number of FP in other sequences, which can be explained by the effectiveness of our algorithm.

TABLE 7. Comparison of results on different tracking methods on MOT17.

Sequence	Method	MOTA (%)↑	MOTP (%)↑	IDF1 (%)↑	FN↓	FP↓	IDSW↓
MOT17-01-DPM	MOT_LST [32]	40.6	54.8	46.7	2862	843	126
	MASS [17]	47.8	78.1	53.6	2400	875	92
	POI [20]	53.4	**68.5**	55.9	**2397**	522	87
	OMOT_JD [19]	36.2	44.9	39.2	3592	369	154
	FPSN [18]	25.0	-	31.7	4307	476	**54**
	ours	**54.8**	67.6	**59.4**	2557	**279**	79
MOT17-08-DPM	MOT_LST [32]	37.8	48.2	42.5	11300	1534	305
	MASS [17]	45.9	75.4	48.9	10132	1120	176
	POI [20]	**58.2**	**70.5**	**60.1**	**7699**	967	**164**
	OMOT_JD [19]	30.5	40.8	37.5	12525	1792	364
	FPSN [18]	22.3	-	28.2	15496	666	241
	ours	50.9	69.3	54.2	9277	**765**	330

Further, we also verify our algorithm in MOT17-01-DPM and MOT17-08-DPM sequences without camera shaking on the MOT17 dataset. MOT17 dataset is more challenging than the MOT15 dataset, which is why there is a performance drop compared to results on results stated in Table 7.

As shown in Table 7, obviously, the proposed method achieves the best performance in MOTA, IDF1 and FP in MOT17-01-DPM sequence, which means that it can successfully deal with lower illumination. Compared to POI and MASS, the proposed method has higher number of FN because the target is too small and too fuzzy to extract feature points.

Meanwhile, the MOT17-08-DPM sequence is more difficult cases than the MOT17-01-DPM. In this sequence, although our algorithm does not achieve the best results, it is better than several other state-of-the-art methods. In terms of IDF1, results are close to POI and it is 5.4% higher than the third-placed algorithm MASS. The proposed method achieves a good performance in FP. Evaluation results on Table 6 and Table 7 are visualized in Figure.7.

3) TEST FOR OCCLUSION

The performance of occlusion handing was tested by video sequences ADL-Roudle-6, TUD-staditimtte and Car_1, as shown in Figure.8. Figure.8 shows the tracking results of our method under partial occlusion and severe occlusion. When tracking under partial occlusion, we can see that person 24 was tracked from frames 453 to 512 in line 2 and car 2 was tracked correctl from frames 258 to 318 in line 4. From frames 252 in line 1 to 352 in line 2. frames 29 to frames 90 in line 3, and line 4, we can note that the proposed algorithm still has good tracking performance under short term severe occlusion. As shown in line3, person 4 was successfully tracked when he was detected from frames 29 to 90. However, car 9 changed its identity when it was tracked from frames 198 to 378 in line 4. It shows that our method cannot overcome long term severe occlusion tracking because speed is unable to update and position errors will be accumulated. As shown in frames

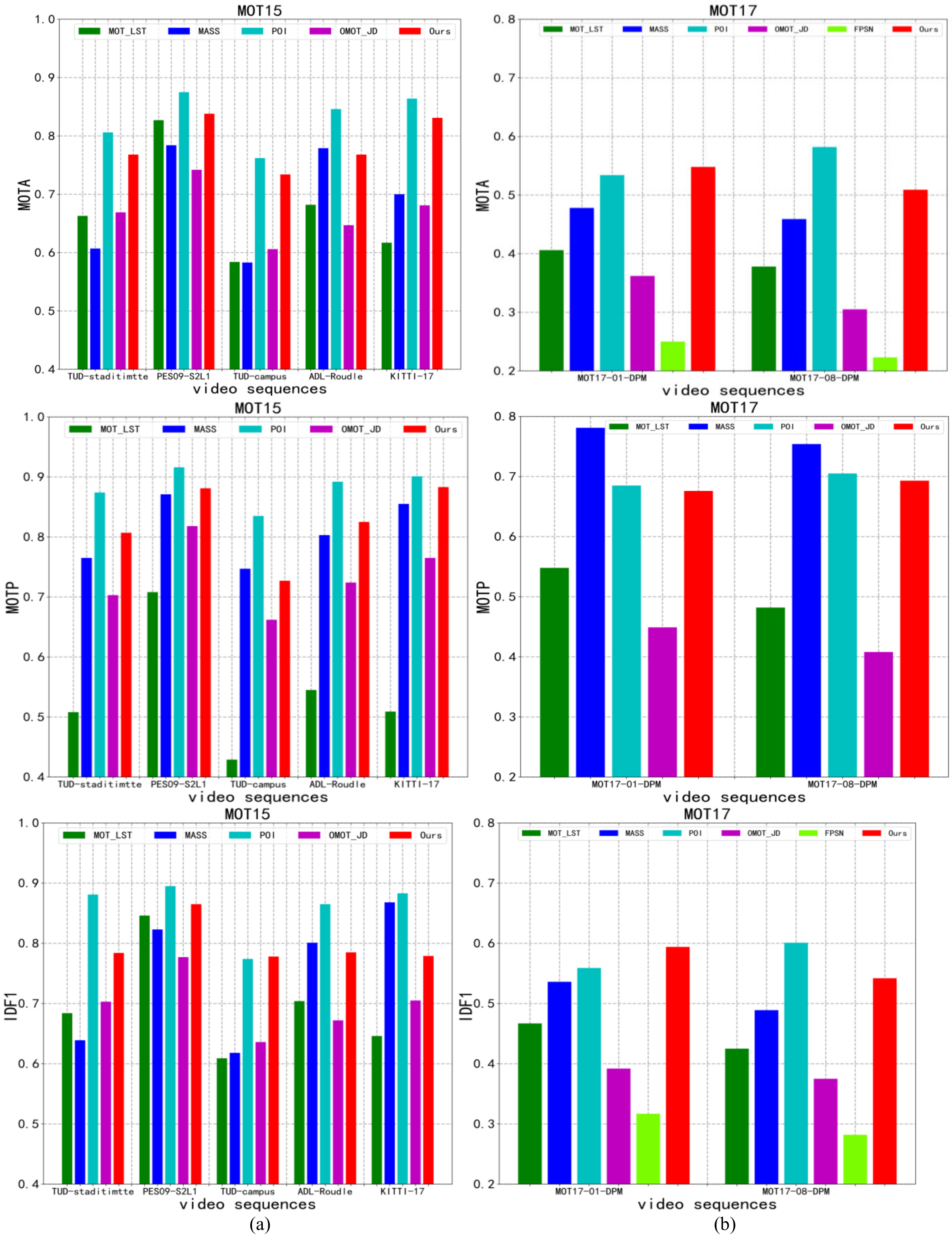


FIGURE 7. Comparison results of MOTA, MOTP and IDF1 in MOT2015 and MOT17. (a). comparison results on MOT2015; (b). comparison results on MOT2017.

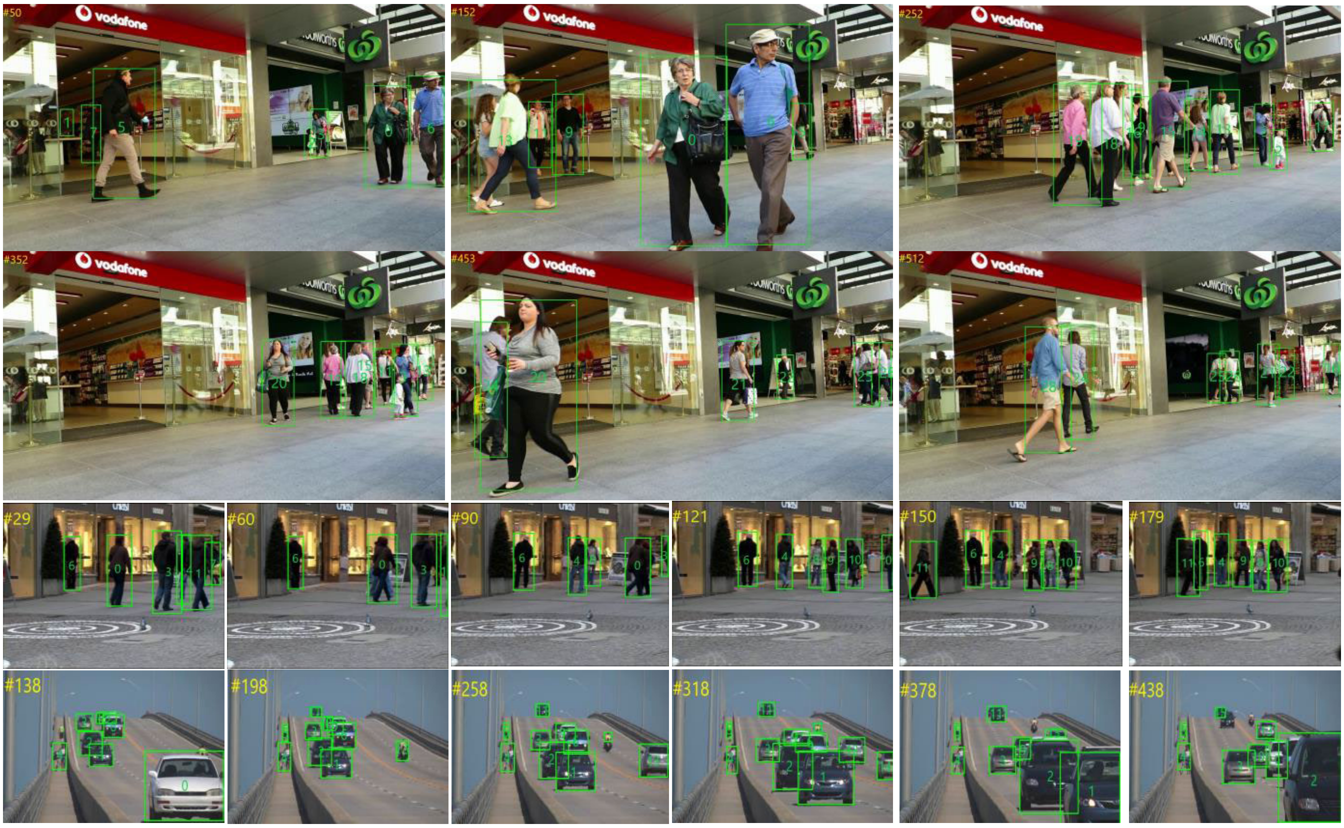
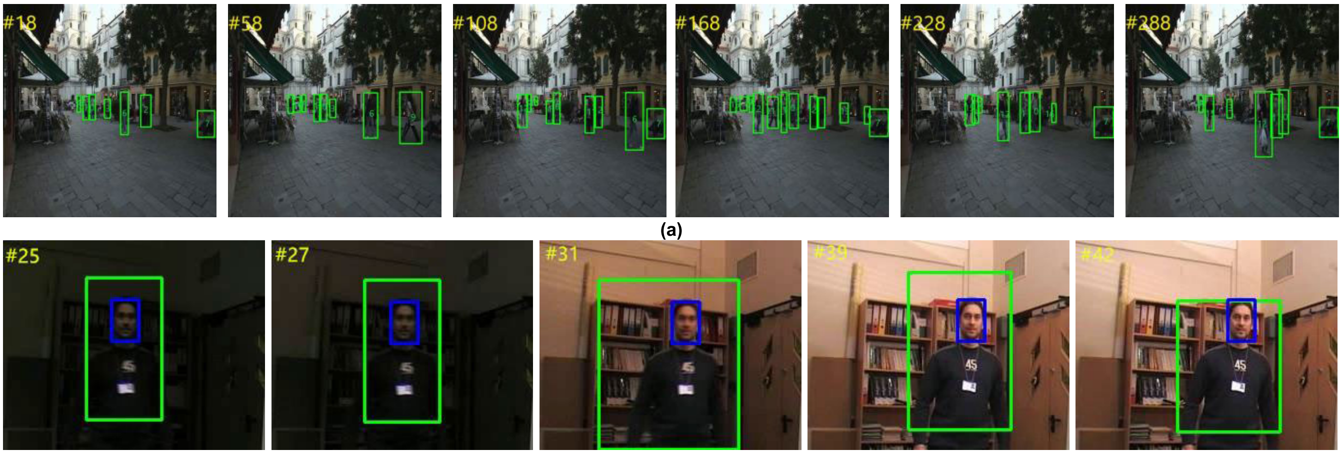


FIGURE 8. Sampling results with occlusion.



(a)

(b)

FIGURE 9. (a) is the sampling results when images darken. (b) is the sampling results when illumination change.

from 198 to 258 in line4, person 14 cannot be identified due to the failure of extracting keypoints.

4) TEST FOR ILLUMINATION

Most of object tracking algorithms are not robust to varying illumination. In this paper, both ORB keypoints and EMcGM are robust to varying illumination and can continue to track effectively when illumination change. This experiment is used to test the tracking performance of the proposed algorithm in the case of illumination change. The video sequences tested are MOT17-01-DPM and Man. The tracking results are shown in Figure 9.

The illumination in Figure.9(a) is low, and our algorithm still obtains good tracking results. In Figure.9(b), the green box is the tracking result of our algorithm, and the blue box is the ground truth. Because the data set only tracks the

TABLE 8. Memory consuming and computational demanding stage.

Memory Consuming and Computational Demanding Stage	Data storage
The establishment of the E-McGM	$(N-len)\times nx\times ny$ $\times nCF\times nTF\times nSF\times m\theta$
Computing $A^T A$	$(N-len)\times nx\times ny\times m\theta\times 6$
The velocity responses in m directions around a point	$(N-len)\times nx\times ny\times m\theta\times 4$
Computing Velocity magnitudes and directions	$(N-len)\times nx\times ny$

TABLE 9. Computation times for 7 video sequences.

	TUD-staditimtte (Hz)	PES09-S2L1 (Hz)	TUD-campus (Hz)	ADL-Roudle-6 (Hz)	KITTI-17 (Hz)	MOT17-01-DPM (Hz)	MOT17-08-DPM (Hz)
MOT_LST [32]	36.7	43.1	49.6	18.2	40.3	13.8	6.2
MASS [17]	44	57.2	**70.7**	**38**	60.8	25.2	**10.8**
POI [20]	42.9	54.4	59.8	32.1	54.9	21.3	9
OMOT_JD [19]	21.3	24.8	27.2	14.6	29.5	9.7	4.1
FPSN [18]	41.2	47.8	52.5	28.2	56.9	18.7	7.9
ours	**45.5**	**60.4**	69.3	37.6	**61.5**	**25.8**	10.4

face, our algorithm can effectively track the entire moving target. During the 27th to 31st frame of the second line, the tracking frame becomes larger. This is because the illumination changes are more obvious, and some background ORB keypoints are not completely filtered, resulting in a larger tracking frame. However, from the 31st frame to the 42st frame, the tracking frame is gradually shrinking, indicating that the algorithm will slightly shake when the illumination changes, but the algorithm will automatically adjust to the optimal value.

5) RUN TIMES

The running time of the algorithm mostly depends on the complexity of the scene in image rather than the size. The more complex image scene, the more calculation time will take. In the proposed algorithm, detector consumes most of the time. Table 8 is the amount of data that needs to be calculated and stored for each stage in E-McGM.

In Table 8, N is the total number of video frames. len is the number of unprocessed video frames. $nx \times ny$ is the total number of pixels in the t-th frame. nCF, nTF, and nSF are the number of color filters, temporal filters, and the spatial filters, respectively. $m\theta$ is the number of different orientations θ around a point in the image. It can be seen from the table that the number of $nx \times ny$ affects the data calculation and memory consumption of all stages in the E-McGM. Therefore, the number of $nx \times ny$ can be reduced by extracting the edge of the image to reduce the operation time and space consumption.

In terms of the worst-case complexity, compared with FPSN, POI and MOT_JD algorithms based on deep learning, our algorithm has an obvious advantage. In the non-deep learning method, the worst-case complexity of MOT_LST is $O(K^{\#}M^{\#}N^{\#}G^{\#})$, where $K^{\#}$ represents the frame numbers of object during the trajectory. $M^{\#}$ represents the numbers of trajectories with low confidence. $N^{\#}$ represents the numbers of trajectories with high confidence. $G^{\#}$ represents the numbers of controlling parameter. The worst-case complexity of MASS is $O(K^{*}N^{*2})$, where K^{*} is the objects number in frame t. N^{*} is the tracks numbers in frame t-1.

The worst-case complexity of our method is $O(K'N'G')$, where K' is the numbers of temporal filters. N' is the numbers of pixels in frame t. G' is the numbers of spatial filters frame t. Generally, we let $K' = 3$. The proposed method and MASS have a better performance in the worst-case complexity. Although the worst complexity of the proposed algorithm and MASS are the same order of magnitude, the overall complexity of the proposed algorithm is better than A due to $K' = 3$.

Table 9 shows the running time of the proposed method and other methods on the 7 video sequences. Although MASS performed better than our tracking method in TUD-campus and ADL-Roudle-6, our work leads to much better results in MOTA, MOTP, IDF1, FN, FP and IDSW.

Compared to POI, our method has lower MOTA, MOTP, IDF1 but higher running speeds in PES09-S2L1, TUD-campus, ADL-Roudle-6, MOT17-01-DPM and MOT17-08-DPM sequences. Hence, our method achieves a good compromise between the speed and the accuracy, which makes it suitable to be used in real- time.

VI. CONCLUSION

This paper proposes a multi-object tracking algorithm based on E-McGM and ORB feature information, which can effectively deal with the problems of occlusion and illumination changes in the tracking process. In the proposed E-McGM, we use Taylor expansion to integrate spatiotemporal- spectral gradients and solve the model with Newton Leibniz's formula. This novel algorithm achieves good performance in detection. Furthermore, in our tracking framework, we use the matching rate function to describe the matching relationship between the object library and the object. We then complete the object consistency by the result of the match rate function and have a good performance in handing with the occlusion problem exiting in persistent MOT.

Experimental results indicate that compared with other state-of-the-art algorithms, the proposed algorithm can effectively deal with the problems of occlusion and illumination changes in multi-object tracking, and can satisfy real-time measurement requirements. For long-term severely occluded situation, the speed cannot be updated in time, which will accumulate position errors and cause mistracking. Persistently tracking in the long-term severe occlusion will be the focus of our future research.

REFERENCES

[1] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[2] Y. Liang, X. Lu, Z. He, and Y. Zheng, "Multiple object tracking by reliable tracklets," *Signal, Image Video Process.*, vol. 13, no. 4, pp. 823–831, Jun. 2019.

[3] Z. X. Li, J. M. Liu, S. Li, D. Y. Bai, and P. Ni, "Group targets tracking algorithm based on box particle filter," *Acta Automatica Sinica*, vol. 41, no. 4, pp. 785–798, 2015.

[4] D. E. O. Dewi, M. M. Fadzil, A. A. M. Faudzi, E. Supriyanto, and K. W. Lai, "Position tracking systems for ultrasound imaging: A survey," in *Medical Imaging Technology*. Singapore: Springer, 2015.

[5] Z. Xi, X. Kan, L. Cao, H. Liu, G. Manogaran, G. Mastorakis, and C. X. Mavromoustakis, "Research on underwater wireless sensor network and MAC protocol and location algorithm," *IEEE Access*, vol. 7, pp. 56606–56616, 2019.

[6] W. Viriyasitavat, L. D. Xu, Z. Bi, and V. Pungpapong, "Blockchain and Internet of Things for modern business process in digital economy—The state of the art," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1420–1432, Dec. 2019.

[7] X. Zhenghao, Y. Niu, J. Chen, X. Kan, and H. Liu, "Facial expression recognition of industrial Internet of Things by parallel neural networks combining texture features," *IEEE Trans. Ind. Informat.*, early access, Jul. 7, 2020, doi: 10.1109/TII.2020.3007629.

[8] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, Jan. 2016.

[9] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear LSTM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 200–215.

[10] A. A. Butt and R. T. Collins, "Multi-target tracking by lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1846–1853.

[11] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[12] D. Sugimura, K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1467–1474.

[13] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 100–111.

[14] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1392–1400.

[15] S. Sharma, J. A. Ansari, J. Krishna Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3508–3515.

[16] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1218–1225.

[17] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104423–104434, 2019.

[18] S. Lee and E. Kim, "Multiple object tracking via feature pyramid siamese networks," *IEEE Access*, vol. 7, pp. 8181–8194, 2019.

[19] H. Kieritz, W. Hubner, and M. Arens, "Joint detection and online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1459–1467.

[20] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 36–42.

[21] L. Zhang and L. van der Maaten, "Structure preserving object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1838–1845.

[22] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[23] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[24] Z. Xi, H. Liu, H. Liu, and Y. Zheng, "Dynamic shortest path association for multiple object tracking in video sequence," *J. Electron. Imag.*, vol. 24, no. 1, Jan. 2015, Art. no. 013009.

[25] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, vol. 2, no. 6, pp. 1265–1272.

[26] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[27] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.

[28] C. Huang, Y. Li, and R. Nevatia, "Multiple target tracking by learning-based hierarchical association of detection responses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 898–910, Apr. 2013.

[29] W. Choi and S. Savarese, "Multiple target tracking in world coordinate with single, minimally calibrated camera," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 553–567.

[30] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury, "A stochastic graph evolution framework for robust multi-target tracking," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 605–619.

[31] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, "Multi-person tracking with sparse detection and continuous segmentation," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 397–410.

[32] J. Wei, M. Yang, and F. Liu, "Learning spatio-temporal information for multi-object tracking," *IEEE Access*, vol. 5, pp. 3869–3877, 2017.

[33] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 385–400.

[34] X. Liang, P. W. McOwan, and A. Johnston, "Biologically inspired framework for spatial and spectral velocity estimations," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 28, no. 4, pp. 713–723, 2011.

[35] P. Golland and A. M. Bruckstein, "Motion from color," *Comput. Vis. Image Understand.*, vol. 68, no. 3, pp. 346–362, Dec. 1997.

[36] R. B. MacLeod, E. Hering, L. M. Hurvich, and D. Jamieson, "Outlines of a theory of the light sense," *Amer. J. Psychol.*, vol. 80, no. 1, p. 163, Mar. 1967, doi: 10.2307/1420566.

[37] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[38] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 387–394.

[39] G.-H. Liu and J.-Y. Yang, "Exploiting color volume and color difference for salient region detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 6–16, Jan. 2019.

[40] A. Boulmerka and M. S. Allili, "Foreground segmentation in videos combining general Gaussian mixture modeling and spatial information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1330–1345, Jun. 2018.

[41] Y. Guo, Z. Li, Y. Liu, G. Yan, and M. Yu, "Video object extraction based on spatiotemporal consistency saliency detection," *IEEE Access*, vol. 6, pp. 35171–35181, 2018.

[42] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

[43] D. M. Chu and A. W. M. Smeulders, "Color invariant surf in discriminative object tracking," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 62–75.

[44] A. Sasithradevi and S. M. M. Roomi, "A new pyramidal opponent color-shape model based video shot boundary detection," *J. Vis. Commun. Image Represent.*, vol. 67, Feb. 2020, Art. no. 102754.

[45] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, T. Baker, W. N. Flayyih, and W. A. Jassim, "A fast feature extraction algorithm for image and video processing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

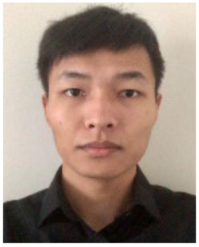
[46] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmmod, and A. Hussien, "Fast temporal video segmentation based on Krawtchouk-Tchebichef moments," *IEEE Access*, vol. 8, pp. 72347–72359, 2020.

[47] S. Zhang, X. Lan, H. Yao, H. Zhou, D. Tao, and X. Li, "A biologically inspired appearance model for robust visual tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2357–2370, Oct. 2017.

[48] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020.

JIEYU CHEN was born in Guangdong, China, in 1995. She received the bachelor's degree in automation from the Shanghai University of Engineering Science, in 2018, where she is currently pursuing the master's degree. Her research interests include digital intelligent systems and computer vision.

ZHENGHAO XI was born in Shanghai, China, in 1981. He received the bachelor's degree in communication engineering from the South-Central University For Nationalities, in 2003, the master's degree in control science and engineering from the University of Science and Technology Liaoning, in 2008, and the Ph.D. degree in control science and engineering from the University of Science and Technology Beijing, in 2015. He is currently a Lecturer with the Shanghai University of Engineering Science. His research interests include digital image processing and computer vision.



CHI WEI received the bachelor's degree in automation from the Shanghai University of Engineering Science, in 2019. He is currently pursuing the master's degree with the University of Nottingham. His research interests include data analyze, machine learning, and digital image processing.

JUNXIN LU was born in Fujian, China, in 1996. He received the bachelor's degree in automation from Jiangsu Ocean University, in 2018. He is currently pursuing the master's degree with the Shanghai University of Engineering Science. His research interests include digital image processing and slam.

YUHUI NIU was born in Shanxi, China, in 1995. She received the bachelor's degree in automation from the Shanghai University of Engineering Science, in 2018, where she is currently pursuing the master's degree. Her research interests include digital image processing and computer vision.

ZHONGFENG LI was born in Liaoning, China, in 1981. He received the bachelor's degree in automation from the University of Science and Technology Liaoning, in 2008. He is currently a Senior Engineer with the Yingkou Institute of Technology. His research interest includes industrial process control.