Journal of Electronic Imaging

JElectronicImaging.org

Saliency-based foreground trajectory extraction using multiscale hybrid masks for action recognition

Guoliang Zhang
Songmin Jia
Xiangyin Zhang
Xiuzhi Li

SPIE. IS&T imaging.org

Guoliang Zhang, Songmin Jia, Xiangyin Zhang, Xiuzhi Li, "Saliency-based foreground trajectory extraction using multiscale hybrid masks for action recognition," *J. Electron. Imaging* **27**(5), 053049 (2018), doi: 10.1117/1.JEI.27.5.053049.

Saliency-based foreground trajectory extraction using multiscale hybrid masks for action recognition

Guoliang Zhang,[a,b,*] Songmin Jia,[a,b] Xiangyin Zhang,[a,b] and Xiuzhi Li[a,b]
[a]Beijing University of Technology, Faculty of Information Technology, Beijing, China
[b]Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing, China

Abstract. Action recognition in realistic scenes is a challenging task in the field of computer vision. Although trajectory-based methods have demonstrated promising performance, background trajectories cannot be filtered out effectively, which leads to a reduction in the ratio of valid trajectories. To address this issue, we propose a saliency-based sampling strategy named foreground trajectories on multiscale hybrid masks (HM-FTs). First, the motion boundary images of each frame are calculated to derive the initial masks. According to the characteristics of action videos, image priors and the synchronous updating mechanism based on cellular automata are exploited to generate an optimized weak saliency map, which will be integrated with a strong saliency map obtained via the multiple kernels boosting algorithm. Then, multiscale hybrid masks are achieved through the collaborative optimization strategy and masks intersection. The compensation schemes are designed to extract a set of foreground trajectories that are closely related to human actions. Finally, a hybrid fusion framework for combining trajectory features and pose features is constructed to enhance the recognition performance. The experimental results on two benchmark datasets demonstrate that the proposed method is effective and improves upon most of the state-of-the-art algorithms. [DOI: 10.1117/1.JEI.27.5.053049]

Keywords: action recognition; foreground trajectory; saliency detection; multiscale hybrid masks; features fusion.

Paper 180389 received May 1, 2018; accepted for publication Oct. 11, 2018; published online Oct. 30, 2018.

1 Introduction

Human action recognition from videos is one of the research hotspots in the field of computer vision. As recognition algorithms have been innovated continuously, recognizing fewer categories of actions in a specific environment is not a significant challenge. However, for the videos captured in realistic scenes, where the problems such as camera movement, viewpoint change, and target occlusion are widespread, and the stability of a recognition system needs to be further improved.

In general, the methods used to achieve action recognition are divided into two categories according to different feature types,[1] i.e., handcrafted methods and deep-learned methods. Recently, the handcrafted methods[2–6] using global representations or local representations have achieved promising performance on a variety of datasets.[7,8] For the global representation methods, Bobick and Davis[9] extracted the motion energy image (MEI) and the motion history image (MHI) and then used the HU invariant moments of MEI and MHI as templates to achieve template matching. Yilmaz and Shah[10] exploited contour information to extract the three-dimensional (3-D) spatiotemporal volume (STV), and the peak point, valley point, and saddle point on the surface of STV are treated as the expression of human behaviors. Sadanand and Corso[5] generated cascaded features based on time-space pyramids, which are utilized as action representations to train a variety of templates and construct a behaviors warehouse named action bank. Then, action recognition is achieved by calculating the response of testing video to the templates.

For the local representation methods, the final recognition performance is determined by the strategies of feature extraction and feature encoding. An early approach[2] extracts space-time interest points from videos, then the descriptors of the histogram of oriented gradients (HOG)[11] and the histogram of oriented flow (HOF)[12] are computed at these points. Wang et al.[13] demonstrated that dense sampling is more efficient than all the tested interest point detectors in realistic video settings. Since the dense trajectory (DT)[3,6] method extracts trajectories by densely sampled points across frames and obtains good performance in various experiments, it is frequently employed as a baseline feature to compare with other methods. However, the original DT feature adopts an indiscriminate dense sampling strategy in all regions of each frame, which has some unavoidable drawbacks for the complex action scenes. For example, when there are other moving objects in the background or the camera is in motion, background trajectories are generated extensively because the area of background is usually much larger than that of action subjects. These action-irrelevant trajectories do not contain any information that facilitates action recognition, thereby limiting the performance of trajectory features. To improve the DT features, Wang et al.[7,8] extracted feature point matches between frames using the speeded up robust features (SURF) and dense optical flow. A homography matrix estimated by matches is used to remove the trajectories consistent with homography and cancel out the camera motion from optical flow. Peng et al.[14] proposed a motion boundary (MB)-based sampling strategy, which can effectively filter out background trajectories while retaining the discriminative power of DT features. Yi et al.[15] utilized

*Address all correspondence to: Guoliang Zhang, E-mail: zhangglmxy@foxmail.com

the appearance saliency and motion saliency to classify the dense trajectories into two categories. Then, the salient foreground trajectories are obtained by subtracting the possible background trajectories based on the low-rank property of background motion. For feature encoding, current methods can be roughly classified into three categories,[16] i.e., voting-based encoding,[3] reconstruction-based encoding,[17] and super vector-based encoding.[7,16] As a super vector encoding method, fisher vector (FV) aggregates information using the first- and second-order statistics and performs well on many challenging datasets[7,16,18] when handcrafted features are employed. Another representative encoding method is the vector of locally aggregated descriptors (VLAD),[19] which is an improved variant of FV and only retains the first-order statistics. Despite the high efficiency of VLAD, its recognition accuracy is slightly lower than FV.

Besides, the human pose feature constructed using joints information is typically designed by experts and considered as another form of the handcrafted feature. It is mainly generated in two steps, i.e., pose estimation and pose feature description. Yang and Ramanan[20] achieved pose estimation in static images based on a flexible mixture model, which is used for capturing contextual co-occurrence relations between human parts and extending the conventional spring model that encodes spatial relationships. Jhuang et al.[21] evaluated the pose estimation algorithm in Ref. 20 by using various types of descriptors derived from joint annotations. The result suggests that even though the estimated joint positions are not entirely accurate, the performance of resulting pose features is not inferior to handcrafted features. Nie et al.[22] introduced a spatial–temporal and-or graph (ST-AOG) model, where each action is described as a tree structure composed of poses, ST-parts, and parts, and action recognition and pose estimation benefit from each other in the same framework.

Due to the success of deep learning technology in image classification, many studies on human behavior analysis based on deep architecture have been launched. Ji et al.[23] developed a 3-D convolutional neural networks (3-D CNN) model that constructs features from both spatial and temporal dimensions by performing 3-D convolutions. Karpathy et al.[24] extended CNN connectivity in the time-domain and proposed an architecture that processes input at two spatial resolutions for accelerated training. Simonyan and Zisserman[25] designed a two-stream CNN structure in combination with spatial and temporal networks and verified that CNN trained on multiframe dense optical flow could effectively improve recognition performance. Cheron et al.[26] presented a pose-based CNN (P-CNN) features and demonstrated the importance of a representation extracted from poses. Similarly, an action conditioned pictorial structure based on CNN is proposed in Ref. 27. By utilizing long short-term memory (LSTM), Krishnan et al.[28] proposed a recurrent neural network variant to keep track of joints and train the network on joint information across an ordered sample of several frames from a video. Mavroudi and Tao[29] used deep appearance and motion features extracted from STVs defined along body part trajectories to learn midlevel classifiers. Wang et al.[30] proposed the trajectory-pooled deep-convolutional descriptor (TDD), which combines the advantages of handcrafted features and deep-learned features. Wang et al.[31] combined the ideas of segmentation and sparse sampling into the two-stream network and proposed the temporal segment network. Overall, deep-learned methods have improved the state-of-the-art performance on many datasets;[25,31,32] however, still some handcrafted features [e.g., improved dense trajectory (iDT)[7,8]] are comparable in performance.[18] In fact, the optimal classification results achieved by deep learning methods are usually obtained by combining with trajectory features.[30,32,33]

In this work, we proposed a trajectory feature and constructed an efficient action recognition framework that combines multiple trajectory features and pose features. Although they describe different aspects of human behavior, the two types of features have potential complementarity. In Ref. 34, this property has been revealed by analyzing their combination conducted in both feature level and classifier level. Nie et al.[22] implemented features fusion based on an ST-AOG model, in which dense trajectories are extracted to generate the coarse features, and human poses are estimated to construct the fine-level feature. Peng et al.[16] proved that each fusion method has its pros and cons, and the practical fusion strategy needs to be formulated by analyzing the correlation of descriptors at different processing levels. Iqbal et al.[27] presented a pictorial structure model to incorporate high-level activity information and then combined the pose-based action recognition with FV encoding of iDT using late fusion. Zhang et al.[35] applied an improved score-level fusion to trajectory features and pose features based on the bag-of-visual-words (BoVW)[16] model and Dempster–Shafer evidence theory and demonstrated that score-level fusion is the most effective strategy for the combination of these two types of features. For the trajectory features, inspired by the breakthrough in saliency detection,[36,37] we propose a foreground trajectory extraction method according to the characteristics of video frames. An overview of our method is shown in Fig. 1. Concretely, the MB image derived from the optical flow is processed to obtain a binary image, which is used as an initial mask for dense sampling. Second, the center-bias[38] and dark channel[39] priors are exploited to detect the foreground region, which is optimized by the synchronous updating mechanism based on cellular automata[40] and then treated as a weak saliency map. The strong saliency map is calculated through a superpixel classification model, which is constructed via the multiple kernels boosting (MKB)[41] method. We apply a collaborative optimization strategy to the integration of the two saliency maps and obtain the final foreground detections for each frame. The multiscale hybrid masks are generated by the intersection of the initial mask and the generalized foreground region. Finally, we can extract a set of foreground trajectories that are closely related to human actions by the compensation schemes.

Moreover, considering the complementarity between multiple features, we design a hybrid fusion framework to integrate the foreground trajectory features, iDT features, and pose features by referring to the correlation between different feature descriptors. The contributions of this paper are as follows:

- To obtain the trajectories closely related to action subject and filter out the trajectories derived from the camera motion and inherent movements in the background, a saliency-based sampling strategy named foreground trajectories on multiscale hybrid masks (HM-FTs) is

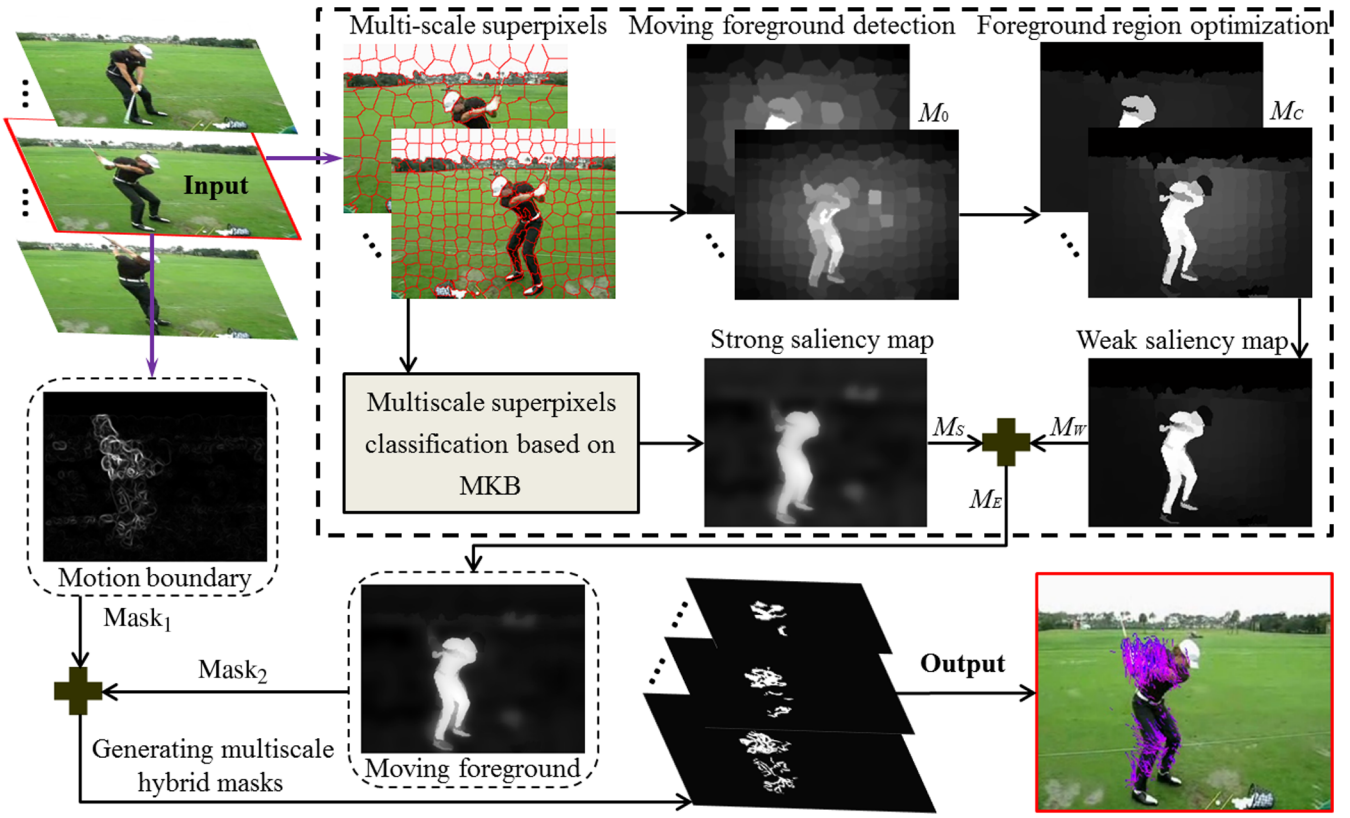

Fig. 1 Flowchart of the foreground trajectory extraction based on multiscale hybrid masks.

proposed. Specifically, according to the characteristics of action videos, a foreground region detection algorithm is presented using the weak saliency map optimized by the synchronous updating mechanism of cellular automata and the strong saliency map achieved through the MKB method.

- The collaborative optimization strategy is formulated to amend the abnormal detection results by exploiting the cooperation between frames. Furthermore, the compensation schemes are designed to improve the robustness of foreground trajectory features.
- A hybrid feature fusion framework, which combines representation- and score-level fusions, is constructed based on the BoVW pipelines. The effectiveness of the HM-FT features and the multifeature fusion method is demonstrated on the Penn Action[42] and sub-JHMDB[21] datasets.

The rest of this paper is organized as follows. In Sec. 2, we describe each extraction step of the proposed HM-FT feature in detail and introduce the iDT feature and an efficient pose feature briefly. A hybrid feature fusion framework for the three types of features is presented in Sec. 3. In Sec. 4, the performance of the HM-FT feature and the features fusion framework is evaluated on the public datasets and compared with state-of-the-art action recognition methods, and this paper is concluded in Sec. 5.

2 Multifeature Extraction Strategy

In this section, the HM-FT feature is presented based on Fig. 1. Different from previous works, we use the optical flow to constrain the sampling points into the MBs to obtain the initial masks. The saliency detection algorithm is improved in the applicability and detection performance based on the characteristics of action videos to generate foreground masks. These masks are integrated to extract trajectory features that are closely related to actions. We also designed the collaborative optimization strategy and the compensation schemes to deal with abnormal and failed detections. Also, we introduce two features that are complementary to HM-FT briefly, and they will be used for features fusion to improve the overall recognition performance.

2.1 *Foreground Trajectory Feature Extraction Based on Multiscale Hybrid Masks*

2.1.1 *Motion boundary detection*

Original DT features need to track densely sampled points on multiple spatial scales of frames, and then generate too many motion trajectories. Although the sampled points in a smooth region are removed when the smaller eigenvalue of its autocorrelation matrix is below a threshold,[3] a large number of points still distribute in the background region. Once there is any moving nontarget human body in these regions, or the camera is shaking, potential background trajectories are generated inevitably, which should significantly reduce the discrimination performance of trajectory features. To solve these problems, we first focus on making the sampled points as much as possible distribute in the boundary of the regions where significant movement occurs in a frame.

The Sobel operator is used to calculate the gradient of horizontal and vertical components of the optical flow to obtain the gradient magnitude images. We compute the

maximum values between the two gradient magnitude images to get a MB image I_B.

The binary image of I_B, which is obtained by the Otsu algorithm[43] and denoted as mask_1, is used as a mask when an image is densely sampled. The dense sampling strategy based on MBs can filter out most of the sampled points in the background, which do not fall in the foreground region of mask_1, so the part of background trajectories generated by camera motion can be removed, as shown in the first two rows of Fig. 2. However, for the regions with rich contours and textures, this method has a poor effect on filtering out background trajectories, as shown in the third row of Fig. 2. The MB of a human can be detected completely, as shown in Fig. 2(b).

2.1.2 *Moving foreground detection*

By researching and summarizing action videos in many datasets, we find that background trajectories are mainly produced by the camera motion and inherent movements in the background. Typical inherent movements in the background include pedestrians passing, vehicles movement, object shaking caused by wind, and so on. To further eliminate the interference of background trajectories, the center-bias and dark channel priors[37–39] are exploited to achieve moving foreground detection in each frame.

The process of shooting an action follows the visual attention mechanism of human eyes, and the purposes of almost all of the intentional camera motions are to lock the moving human in the center of a lens. The dark channel prior[39] is proposed for the image haze removal. It is a statistic-based algorithm summarized by analyzing a large number of foggy images. The observation result shows that those regions that do not include the sky have one or more pixels whose intensity values are approximately equal to zero in one of the RGB color channels. The dark channel of an image is mainly generated by shadow regions and the surface of colored or dark objects, which generally appear in the foreground regions, see Figs. 3(a)–3(c).

Therefore, the dark channel property is exploited as prior information in the process of moving foreground detection. Assuming that the dark channel prior value of pixel p is $S(p)$, which is calculated as

$$S(p) = 1 - \min_{q \in Q} \left\{ \min_{C \in \{R,G,\mathrm{B}\}} [V^C(q)] \right\}, \tag{1}$$

where Q denotes a 5×5 image patch centered on p, and $V^C(q)$ denotes the color value of pixel q in channel C. However, for the images with a brighter foreground or darker background, the dark channel prior may lead to a failure of moving foreground detection as shown in Fig. 3(d). To this end, we calculate the mean value of all $S(p_e)$, where p_e is a specific pixel on the borders in a frame. If it is greater than 0.8, the influence of dark channel prior on a frame will be eliminated, and the value of $S(p)$ will be set to zero.

To obtain structure information of the moving foreground, the simple linear iterative clustering[44] algorithm is exploited to achieve multiscale superpixel segmentation for an input frame. The numbers of superpixels at different scales are set to 100, 150, 200, and 250, respectively, to avoid the incomplete structure information caused by single-scale superpixel segmentation. Let b_i denote a superpixel, $i = 1, \ldots, N$, where N is the number of superpixels for a scale. The saliency value of b_i is calculated as

$$m(b_i) = g(b_i) \times \exp[S(b_i)] \times \sum_{f \in F} \left[\frac{1}{N_B} \sum_{j=1}^{N_B} d_f(b_i, e_j) \right], \tag{2}$$

where $d_f(b_i, e_i)$ represents the Euclidean distance in f feature space between b_i and a border superpixel e_j, and N_B is the number of superpixels along a frame border. f is the type of features. F includes RGB, CIELab, and LBP features,

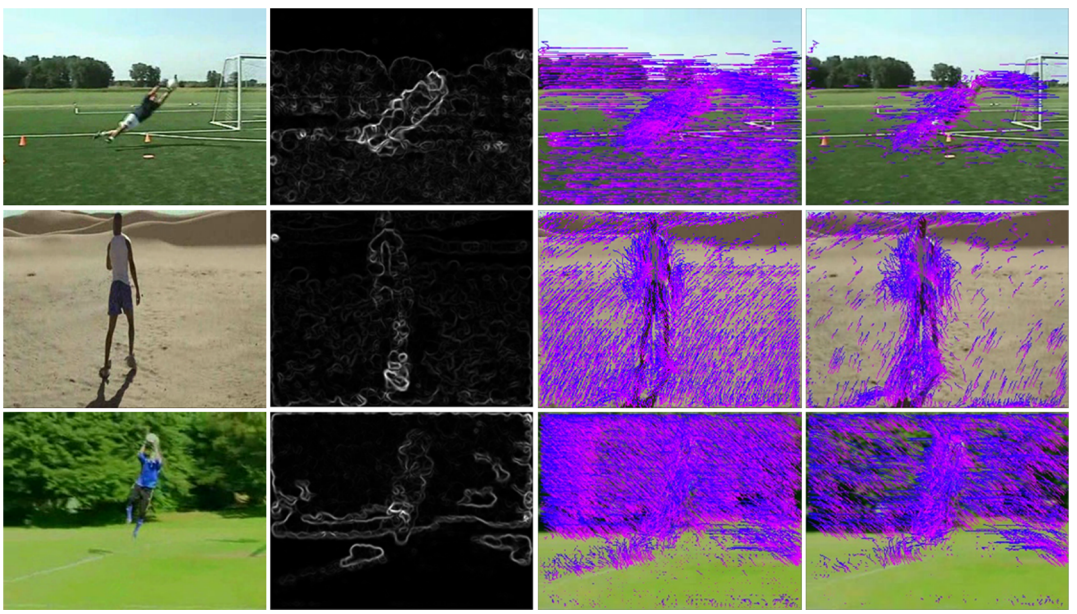

Fig. 2 Comparison of the original DT and the dense trajectories on MB. (a) Video frame, (b) MB image, (c) trajectories by DT, and (d) trajectories by MB.

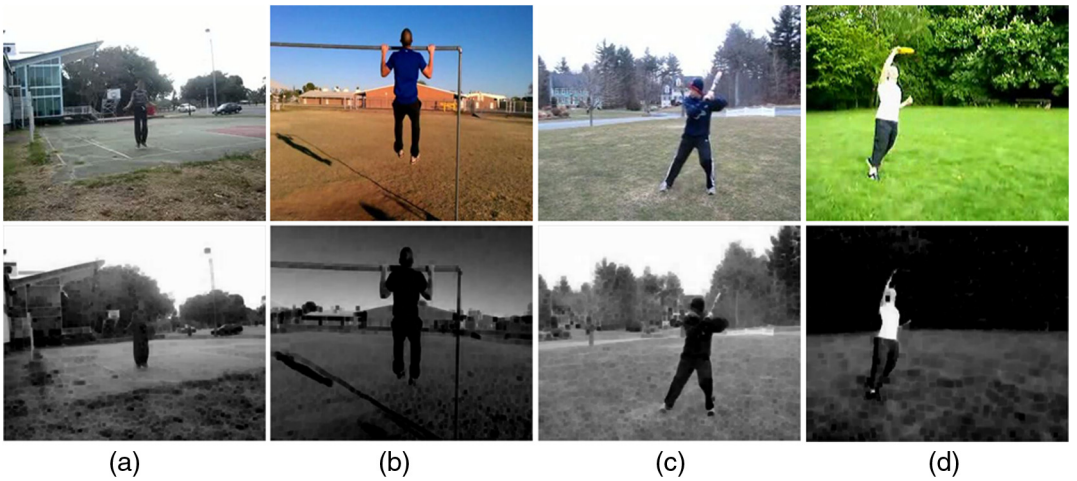

Fig. 3 Examples of the dark channel prior for the frames of action videos. (a)–(c) show the dark channel generated by foreground regions and (d) shows the dark channel generated by the image with a brighter foreground or darker background.

because there is a complementarity between RGB and CIELab, and the color and texture features used simultaneously are more robust to complex backgrounds. $S(b_i)$ is the mean value of all $S(p)$, $p \in b_i$. $g(b_i)$ is the weight of center-bias prior, and its value is equal to the normalized spatial distance between the center of b_i and the frame center. The value of $m(b_i)$ is assigned to all the pixels in the region b_i, and we use Gaussian filtering to generate the moving foreground map M_0.

2.1.3 *Foreground region optimization*

To obtain a more accurate foreground region, all the superpixels at each scale are treated as a set of cells, and the synchronous updating mechanism based on cellular automata[40] is exploited to optimize M_0. Unlike the original cellular automata models, the influences of neighbors to a cell are not fixed. The influence of any pair of cells is closely related to their similarity in CIELab color space. Accordingly, the impact factor z_{ij} between cell b_i and its neighbor b_j is calculated as

$$z_{ij} = \exp\left[\frac{-d(b_i, b_j)}{\mu}\right] \qquad i \neq j, \tag{3}$$

where $d(b_i, b_j)$ represents the Euclidean distance between b_i and b_j. μ is the regulatory factor. We follow Ref. 45 to set the value of μ to 0.1.

The z_{ij} of any pair of adjacent cells is calculated to construct an impact factor matrix $\mathbf{Z} = [z_{ij}]_{N\times N}$ with the main diagonal elements of zero. Note that all the cells along a frame borders are considered to be interconnected because all of them are regarded as background seeds. Then, each row of $\mathbf{Z}$ is normalized by $d_i = \sum_j z_{ij}$, i, $j = 1,2,\ldots,N$. A coherence matrix $\mathbf{T} = diag\{c_1, c_2, \ldots, c_N\}$ is established to make the moving foreground more complete and avoid losing fine structures on a human body, where the calculation equation for c_i is written as

$$c_i = \frac{1}{\max(z_{ij})}. \tag{4}$$

If there is a significant difference between a cell and its neighbors, the state of the next moment of the cell will be determined primarily by itself. On the contrary, if the cell is more similar to a neighbor, it is likely to be assimilated by the neighbor. Considering that the evolution of a cell will produce extreme results when c_i is too high or too low, we follow Ref. 46 to convert the value of c_i to $[\gamma, \gamma + \eta]$ by

$$c_i = \gamma + \eta \cdot \frac{c_i - \min(c_j)}{\max(c_j) - \min(c_j)}, \tag{5}$$

where $j = 1,\ldots,N$. The synchronous updating mechanism for cellular automata is formulated based on the impact factor matrix and the coherence matrix as follows:

$$M_{t+1} = T \cdot M_t + (I - T) \cdot Z \cdot M_t, \tag{6}$$

where I is the identity matrix. When $t = 0$, the initial M_t is the moving foreground map M_0. The optimized saliency map M_C is achieved by iteratively executing the updating mechanism. Note that we use the saliency value of each superpixel as its state, which can describe the relationship between the cells more comprehensively and reasonably. In the iterative process, since the influences of neighbors are changed, cellular automata based on the broader definition of neighborhoods can enhance saliency consistency among similar regions and form a clear boundary between the action subject and the background naturally. Besides, when salient superpixels are selected as the background by mistake, they will automatically increase their saliency values under the influence of the local environment.

The Otsu algorithm is used to calculate the binary image M_B^i of saliency map M_C^i at the i'th superpixel scale. We consider both M_B^i and M_C^i to construct the weak saliency map M_W by Eq. (7), which is written as

$$M_W = \frac{1}{n}\sum_i^n \frac{(M_C^i + M_B^i)}{2}, \tag{7}$$

where n is the number of superpixel scales.

2.1.4 *Multiscale superpixels classification*

We select training samples from every M_W^i, where $M_W^i = \frac{M_C^i + M_B^i}{2}$, and then construct a superpixels classification

model based on the MKB[41] method to obtain the strong saliency map under each superpixel scale. Specifically, if $V_j >= \lambda_{\max} \times V_M$, the j'th superpixel will be regarded as a positive sample. Otherwise, if $V_j < \lambda_{\min}$, the j'th superpixel will be considered as a negative sample. V_M represents the average saliency value of M_W^i, and V_j represents the average saliency value of the j'th superpixel. To control the number of training samples, shorten the training time, and ensure the balance of positive and negative samples, $\lambda_{\max}$ and $\lambda_{\min}$ are set to 1.5 and 0.05, respectively.

The RGB, CIELab, and LBP features extracted from training samples are utilized to train the strong classifier by MKB. The discriminant function of MKB is constructed based on the traditional multiple kernel learning method and shown as follows:

$$f(x) = \sum_{r=1}^{R} \beta_r \left[\sum_{h=1}^{H} \alpha_h y_h k_r(x_h, x) + b_r \right], \tag{8}$$

where H is the number of training samples, R is the number of weak classifiers, x_h denotes a training sample, and $y_h \in \{-1,1\}$ denotes the label corresponding to x_h. Moreover, β_r is the weight of the kernel function $k_r(x_h, x)$, α_h is the Lagrange multiplier, and b_r is a constant term. α_h and b_r of each weak classifier can be obtained by solving the corresponding quadratic programming problem.

We can achieve $3 \times K$ basic classifiers based on the three feature sets and K types of kernel functions. The AdaBoost algorithm is used to solve the weight β_r of each weak classifier iteratively and output weak classifier models. All the superpixels at n superpixel scales derived from a frame are considered as the testing samples. Equation (8) can be used to output the decision values of every superpixel, which will be assigned to all the pixels in the region and then normalized to achieve the strong saliency map at each scale. The Gaussian filtering is used to generate a smoother saliency map M_S^i. M_H^i denotes the binary image of M_S^i. The strong saliency map $\overline{M}_S$ for a frame is constructed as

$$\overline{M}_S = \frac{1}{n} \sum_{i}^{n} \frac{(M_S^i + M_H^i)}{2}. \tag{9}$$

Considering that the guided filter has the properties of preserving strong edges and blurring weak edges, we use it to optimize $\overline{M}_S$. The resulting saliency map is represented as M_S.

2.1.5 *Multiscale hybrid masks acquisition*

The weak saliency map is easier to capture the local structure information of the moving foreground. However, the strong saliency map achieved by the MKB[41] method, which transforms the task of foreground detection into solving a binary classification problem for superpixels, tends to describe the global information of objects. The two saliency maps are integrated by a weighted fusion method to obtain the final result of foreground detection (which is represented as M_E), where the ratio factors are $\omega_S = 0.7$ and $\omega_W = 0.3$.

The action scenes and appearance of a moving human in all frames of the same video are usually highly consistent, so the detections for these frames are similar, especially for the adjacent frames. Although the subtle local changes occur in the foreground region due to the lens movement and human-pose adjustment, there is strong cooperation between the detections of frames. The collaborative optimization strategy is proposed to amend the abnormal detections based on the above analysis. The specific steps are as follows.

First, we concatenate the saliency values of all pixels in the normalized M_E to generate a feature vector f_i. Assuming that there is an action video with m frames, the Euclidean distance between any two saliency feature vectors is $d(f_i, f_j)$, where $i, j = 1,2,\ldots,m$, and $i \neq j$. The sum of $d(f_i, f_{i+1})$ and $d(f_i, f_{i-1})$ is denoted as φ_i. Then, abnormal frames are selected out as

$$\xi = \sigma \cdot \frac{\sum_{i=2}^{m-1} \varphi_i}{m-2}, \tag{10}$$

where the scale factor σ is set as 1.5. If $\varphi_i >= \xi$, the i'th frame will be regarded as an abnormal frame. Otherwise, it will be defined as a keyframe. Finally, the saliency values of abnormal frames, which are not adjacent to each other, are reset to the average of saliency values of the previous and subsequent keyframes. If the abnormal frames are continuous, we calculate the abnormity degree for each of them by $\sum_{j=1, j\neq i}^{m} d(f_i, f_j)$ and regard the frame with minimum abnormity degree as a relative keyframe ψ. Let ψ_c be the nearest keyframe of ψ, the saliency values of the frames between ψ and ψ_c are reset to the average of ψ and ψ_c.

We use two iterations of the morphological dilation on the binary image of M_E to generate a robust foreground mask denoted as mask_2. To make a human body covered by mask_2 more complete and overcome the problem that the extremities and head are lost in detection due to the low image resolution, mask_2 is generalized as follows: if the area of the foreground region in mask_2 is less than or equal to 0.08 of the image area, the bounding box of foreground will be constructed using the maximum and minimum values of its pixel coordinates in the horizontal and vertical directions. The distances between the center of a bounding box and its four borders are respectively increased by 3 pixels, which decrease as the spatial scales of a frame decrease progressively, to obtain a generalized mask_2.

We calculate the intersection of mask_1 and mask_2 on each spatial scale of a frame separately to achieve the multiscale hybrid masks for the moving foreground.

2.1.6 *Foreground trajectory features extraction*

Two-dimensional grids are constructed for each frame with a sampling step size of 5 pixels on eight spatial scales spaced by a factor of $1/\sqrt{2}$ to extract foreground trajectories. The multiscale hybrid masks are used to refine the sampled points. Specifically, when the sampled points do not fall in the foreground region of the mask, it will be removed. Foreground trajectories are generated by tracking the remaining points on multiple spatial scales of a frame. To fully mine the motion information from foreground trajectories, multiple descriptors [i.e., trajectory shape, HOG, HOF, and motion boundary histogram (MBH)[12]] within a space-time volume around trajectory are computed. We use the identical settings to Ref. 3, so the final dimensions of the descriptors are 30 for TS, 96 for HOG, 108 for HOF, and 192 for MBH.

When the variations between adjacent frames are too subtle to generate a large number of MBs, sampled points

extracted by the proposed method are relatively few. However, when hybrid masks of each frame can completely cover the moving foreground, the frames with fewer sampled points are only a small part of a video. In extreme cases, since the colors of background and foreground are highly consistent, or there are some objects in the background that are more salient than the moving foreground, the detection will deviate from the foreground region. These deviations cause too few foreground trajectories related to the human motion to describe actions adequately, thereby reducing the discrimination power of the trajectory features. To solve the above problems, we have formulated two compensation schemes:

- Sampled points are extracted from each frame using the hybrid masks, and the number of frames in which the number of points in the first layer of the image pyramid is not larger than τ_1 is counted as m_f. If $m_f/(m-1) \geq 0.5$, the hybrid mask will be replaced by mask_1, and the trajectory features for the video will be re-extracted. Since the number of sampled points is proportional to image resolution, τ_1 is adjusted by $\tau_1 = \overline{p} \cdot \tau_2$ adaptively, where $\overline{p}$ is the baseline number of points and τ_2 is the scaling factor for resolution.
- When only mask_1 is employed, the real background trajectories are sparser than the real foreground trajectories, so failed foreground detection will lead to a decrease in the number of trajectories. We denote the number of trajectories for a video as N_{ts}. If $N_{ts}/m \leq \tau_3$, where $\tau_3 = \overline{d} \cdot \tau_2$, $\overline{d}$ is the baseline number of trajectories, then we will re-extract the original DT features for the video.

Note that the first scheme is given priority. Trajectory features processed by the first scheme will be judged and corrected again by the second scheme. In the end, the HM-FTs can be obtained. Figure 4 shows the visualization of different stages of foreground trajectory extraction, including the trajectories by DT, MB image, trajectories by MB, foreground detection results, and our proposed HM-FTs. From Fig. 4, we find that the MBs can filter out most of the trajectories in the smooth region of background. However, for the background regions with rich contours and textures, the method cannot remove background trajectories generated by camera motion, as shown in the first-, third-, and sixth-rows of

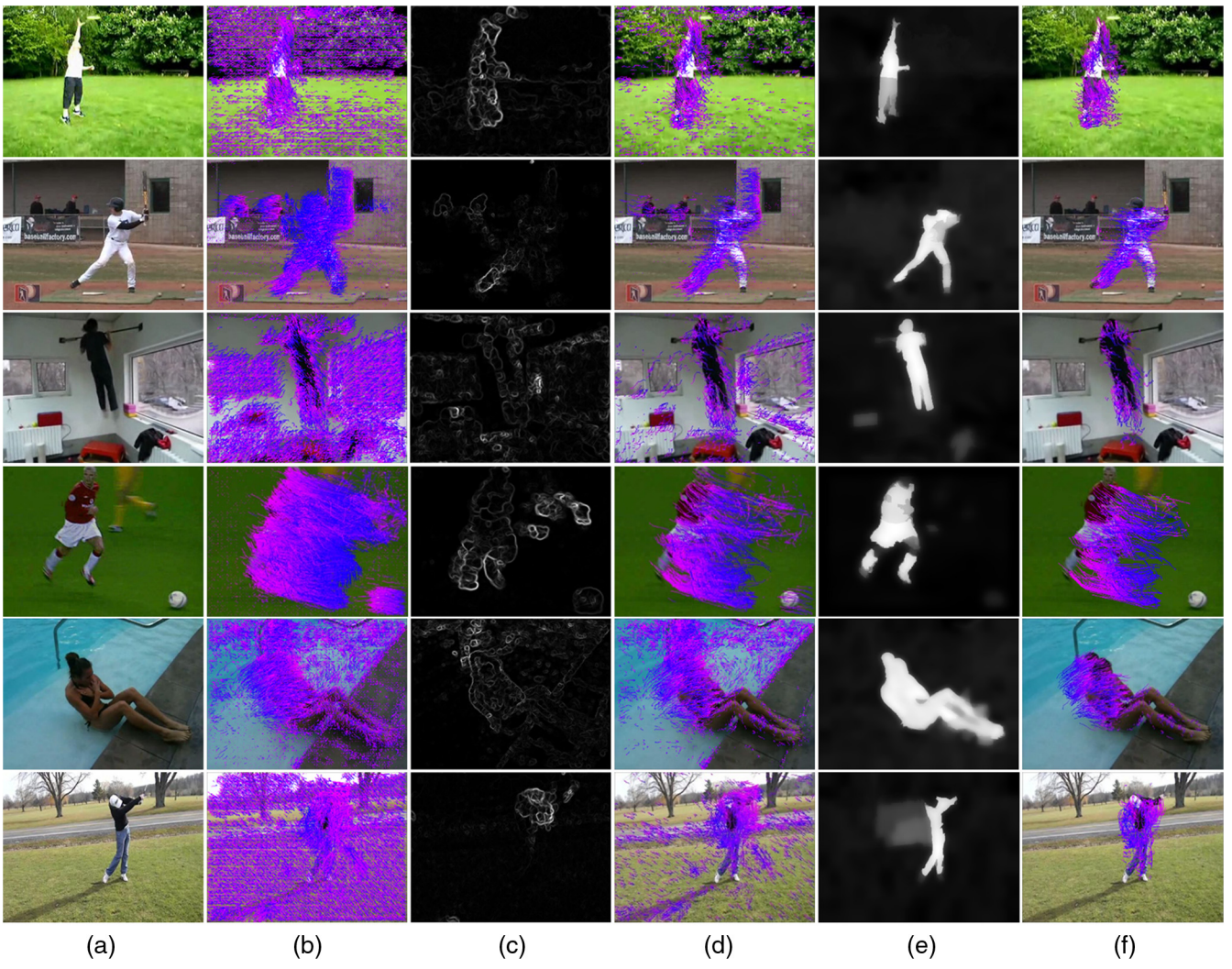

Fig. 4 Visualizations of trajectories in different stages of the proposed method. (a) Video frame, (b) trajectories by DT, (c) MB image, (d) trajectories by MB, (e) foreground detection results, and (f) trajectories by HM-FT.

Fig. 4(d). For the trajectories generated by inherent movements in the background (e.g., pedestrians passing, nontarget object movement, water surface fluctuations, etc.), the method does not have any effective removal mechanism, as shown in the second-, fourth-, and fifth-rows of Fig. 4(d). The hybrid masks achieved by exploiting MB detection and foreground detection are utilized to refine the densely sampled points. The resulting foreground trajectories not only suppress the influence of background trajectories on the discrimination power of features but also contain abundant information for human action, which makes the trajectory features more expressive, as shown in Fig. 4(f).

2.2 *Improved Dense Trajectory Feature Extraction*

Although the above method can effectively filter out background trajectories, the offset of foreground trajectory caused by camera motion lacks necessary amendments. Therefore, we combine the foreground trajectory features with the iDT features to make up for the deficiency. iDT feature is an improved version from DT, which makes reasonable estimation and effective utilization for the information of camera motion so that the trajectory feature is more focused on describing the subject of an action. Specifically, iDT assumes that there is a homography transformation between adjacent frames because the changes between them are relatively slight. Then, camera motion estimation can be solved by calculating a homography matrix between adjacent frames. The SURF and dense optical flow are used to achieve frames matching and obtain the matching point pairs. The global homography matrix is calculated by the random sample consensus algorithm[47] based on these point pairs. The original DT will be amended by the camera motion information.

2.3 *Pose-Based Feature Extraction*

Trajectory features are used to describe the apparent structure and motion state around trajectories. However, pose features focus on describing the distribution and coupling relationship of human joints. These two types of features are highly complementary.[35]

The popular methods[20,22] for pose estimation usually describe human joints as a tree-structured graph and use the dynamic programming algorithm to deduce the positions of every joint. Considering that the framework mentioned in Ref. 20 is representative, it is employed to achieve pose estimation. Some pose estimation results for the full body with 26 human joints are shown in Fig. 5. The pose descriptors are designed from both time- and space-level based on the results of pose estimation. To remove redundant joints, we follow Ref. 21 to retain 15 joints for describing a full body. When the frame step is set to s, if the joint coordinate is (x, y), the coordinate displacements are d_x and d_y, and the angle of the space-time displacement vector is $\arctan(d_x/d_y)$.

To improve the pose features, a weakening factor R is used to attenuate the effect of joints information in the initial and last frames, because the motion amplitude in middle frames is usually more apparent. For a video with m frames, the multiple sets of descriptors at time-level can be obtained by constructing the coordinate displacement matrix P_{tr} and vectorial angle matrix P_{an}, which are shown as

$$P_{tr} = \begin{bmatrix} f_1 - f_{1+s} & \cdots & f_{m-Rs} - f_{m-(R-1)s} \\ \vdots & \ddots & \vdots \\ f_{1+(R-1)s} - f_{1+Rs} & \cdots & f_{m-s} - f_m \end{bmatrix}, \tag{11}$$

$$P_{tr} = \begin{bmatrix} (f_1, f_{1+s}) & \cdots & (f_{m-Rs}, f_{m-(R-1)s}) \\ \vdots & \ddots & \vdots \\ (f_{1+(R-1)s}, f_{1+Rs}) & \cdots & (f_{m-s}, f_m) \end{bmatrix}, \tag{12}$$

where f represents all the joint coordinate data for a frame, and its subscript indicates the frame number. (f_1, f_{1+s}) is a column vector consisting of 15 angles of space-time displacement vectors. Each type of time-level descriptors for a video is composed of the data in the same dimension of all elements in P_{tr} or P_{an}. Therefore, we can obtain 75 types of descriptors derived from human joints.

3 Features Fusion and Classification

In this work, different pipelines of BoVW are employed to construct the video-level representation from a set of descriptors. The trajectory features focus on describing the appearance structures, motion states, and MBs of a video. The pose features focus on describing the changes for the position and

Fig. 5 Some pose estimation results for the full body with 26 human joints on datasets.

movement of joints at both temporal and spatial hierarchies. To make full use of the complementarity between these two features and exert their respective advantages, we designed a simple and effective features fusion method.

For the trajectory features, we extract two sets of trajectories from a given video, namely, HM-FTs and iDTs. For iDTs, we follow the framework of Ref. 7 to compute multiple descriptors under the default settings. For each set of trajectories, four types of descriptors are encoded separately to obtain the video-level representations. The global representation derived from a specific type of descriptors can be obtained by concatenating the same type of video-level representations. Due to the strong correlation between different descriptors, representation-level fusion is exploited, which has been proved to be the best choice in Ref. 16. Generally speaking, different global representations are concatenated as a final representation for a video. The principal component analysis (PCA) is employed to reduce the dimension of descriptors to half of the original dimension. The whitening technique is combined with PCA to ensure that each dimension of the dimensionality-reduced vector has the same variance. We randomly select 256,000 descriptors from each descriptor set to train a Gaussian mixture model with 256 components respectively. FV is utilized to encode the processed descriptors by the implementation of VLFeat Toolbox[48] and then normalized by the L2 and power normalization. A linear SVM with fixed $C = 100$ is used for classification because it has been proven to be more efficient in combination with FV.[7] The decision matrix M_{tr} of the combination of HM-FT and iDT for all testing samples is calculated by the one-against-rest approach.

For the various descriptors of pose features, we employ the representation-level fusion to obtain a global representation for a video. All the training samples of a particular descriptor type are exploited to generate a codebook of size 20 by the k-means[49] algorithm. We use the vector quantization to encode these descriptors, which are then normalized and concatenated to generate a 1500-dimensional pose feature for a video. An SVM with RBF kernel is selected for classification, where the fivefold cross-validation is used to calculate the optimal parameters. The one-against-rest approach is utilized to calculate the decision matrix M_{po} for testing samples.

Since trajectory features and pose features are independent of each other, the score-level fusion[35] is chosen to achieve their integration. Let the final decision matrix be $Z_f = M_{tr} + M_{\mathrm{po}}$, the prediction with the highest score of each row in Z_f is selected as the classification result.

4 Experiments

4.1 *Datasets*

Our method is compared and analyzed on two benchmark action datasets including Penn Action[42] and sub-JHMDB.[21] These two datasets are utilized to evaluate the algorithms for pose estimation and action recognition, and the research object is full body. The Penn Action dataset contains 2326 video clips that belong to 15 action categories. They are "baseball pitch," "baseball swing," "bench press," "bowling," "clean and jerk," "golf swing," "jump rope," "jumping jacks," "pull up," "push up," "sit up," "squats," "strumming guitar," "tennis forehand," and "tennis serve." To achieve the extraction of pose features, we remove the action "strumming guitar" and several samples according to Ref. 22 because most of the human body in those data is invisible. The pruned dataset contains 1206 training samples and 1017 testing samples, and its average accuracy is reported by utilizing the train/test split provided in Ref. 42.

As a subset of HMDB51, the sub-JHMDB dataset[21] contains 316 video clips that belong to 12 action categories. They are "catching," "climbing stairs," "golfing," "jumping," "kicking ball," "picking," "pulling up," "pushing," "running," "shooting ball," "swinging baseball," and "walking." We test the sub-JHMDB dataset by using the threefold cross-validation presented in Ref. 21 and report the average accuracy of three splits.

We have considered using the complete HMDB51 dataset, but we found that the dataset is not suitable as a benchmark for the performance evaluation of our method. The pose features cannot be extracted from a large number of samples in the HMDB51 dataset, because they do not contain any fully visible person. Some actions only contain head and shoulder, such as "smile," "chew," "laugh," "talk," "drink," "kiss," "eat," and "smoke." Moreover, the pose estimation algorithm presented in this paper also cannot be applied to the actions where more than 1/2 or even 2/3 of a person is invisible, such as "shake hands," "sit down," "brush hair," "pour," "hug," and "clap hands." To this end, all the performance evaluations and comparisons are achieved on the sub-JHMDB dataset.[21] Some sample frames from Penn Action, sub-JHMDB, and HMDB51 are shown in Fig. 6.

4.2 *Experimental Results and Discussions*

4.2.1 *Basic performance evaluation for HM-FT feature*

The effectiveness of HM-FT feature is demonstrated by testing it on the two public datasets for human action recognition. All the experiments are conducted on a lab computer running Windows 10 with 3.50 GHz Intel Core i7-5930K CPU and 64 GB of RAM. We have used Matlab R2015a and Visual Studio 2013 for implementation purposes.

For the HM-FT feature, the range of c_i in the coherence matrix during the process of foreground detection is set to [0.2,0.8]. If η is fixed to 0.6, the results are virtually unchanged when γ varies from 0.1 to 0.3. In the stage of superpixels classification, three kinds of kernel functions, including linear kernel function, polynomial kernel function, and RBF kernel function, are utilized to train basic classifiers. Moreover, considering that the image size for all videos in sub-JHMDB is 320×240 pixels, it is set as the baseline resolution. The influence of different parameters $\overline{p}$ and $\overline{d}$ on recognition results will be discussed in Sec. 4.2.4. Here, we report the best performance of HM-FT feature with $\overline{p} = 8$ and $\overline{d} = 5$. With the HM-FT feature, the average recognition rates on Penn Action and sub-JHMDB are 88.91% and 63.19%, respectively. Note that although the average accuracy is reported both for the two datasets, we follow Ref. 21 to calculate the per-video accuracy for sub-JHMDB, which does differ from the per-class accuracy employed in Penn Action.[42] The confusion matrices on the two datasets are shown in Fig. 7.

Figure 7(a) shows that on Penn Action dataset, we achieve high accuracies on most of the actions, such as "clean and

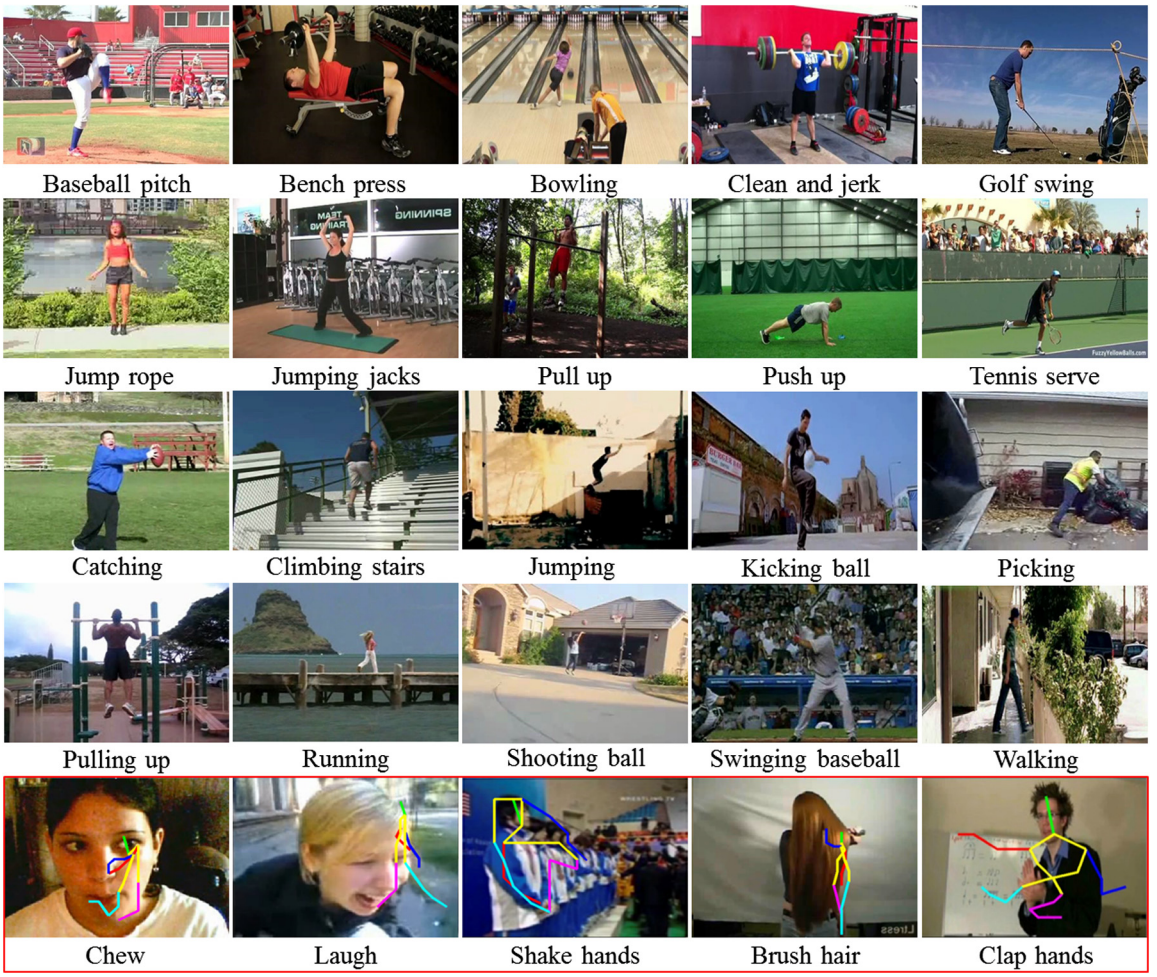

Fig. 6 Sample frames from Penn Action, sub-JHMDB, and HMDB51. The frames in the first four rows are from Penn Action and sub-JHMDB, and the last row shows the frames with failure pose estimations form HMDB51.

jerk," "jump rope," and "bowling." However, "bench press" has the lowest recognition accuracy with 0.71. In most cases, its testing samples are incorrectly recognized as "push up" because both of the actions only include the up-and-down motion of arms, and their movement ranges are similar.

As for sub-JHMDB in Figs. 7(b)–7(d), although the numbers of testing samples for the same action are different in three splits, the proposed HM-FT feature performs well on the actions such as "golf" and "shoot ball" in general. Moreover, we find that "climb stairs" and "walk" are easily confused with each other. Compared to the latter, there is an upward trend in the trajectory of "climb stairs," but the camera always adjusts objects to the center of the visual field, which make this difference insignificant.

From the four confusion matrices, we can infer that even though the extracted HM-FT features have effectively suppressed the interference of background trajectories to action recognition, it is not entirely robust to the action classes with highly similar motion patterns. The future work will focus on identifying motion-related objects in the scene to provide necessary semantic information for different actions, which is considered as an auxiliary discrimination basis to improve the discrimination power of HM-FT features.

4.2.2 *Overall recognition performance*

For comparison, the recognition performance of different features, including trajectory features, pose features, and their combinations, are evaluated on the two public datasets. To ensure the objectivity of results, we apply the same BoVW pipeline to different types of trajectory features. The specific settings of each step of BoVW (i.e., feature preprocessing, codebook generation, feature encoding, and normalization) and the selection of classifiers are determined by referencing to Sec. 3.

For the pose features, we set both the frame step s and the weakening factor R to 3. Unlike the 3225 types of descriptors shown in Ref. 21, the optimized pose features only contain 75 types of descriptors, but they can significantly reduce running time and preserve discriminative power (note that the accuracy achieved by the combination of 3225 descriptor types and DT on the sub-JHMDB dataset is 52.9%).[21] For example, when the video contains 42 frames with a resolution of 320×240 pixels, the running time of the optimized pose features is about 0.0058 s, which is far less than 6.17 s consumed by the 3225 descriptor types.

Table 1 presents the comparison of the average accuracies achieved by different methods, where Comb. 1 is the combination of HM-FT and iDT, and Comb. 2 is the combination

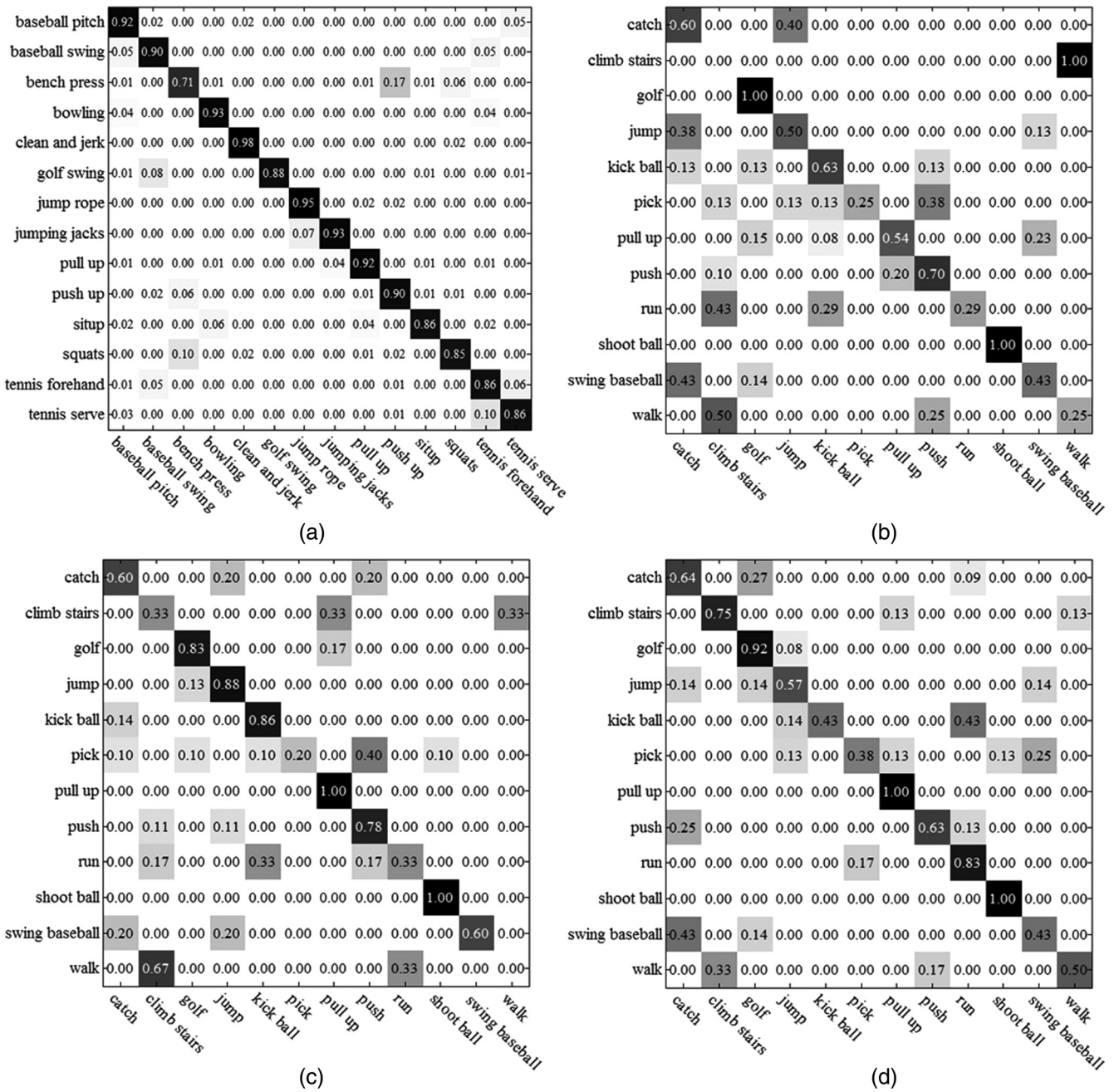

Fig. 7 The confusion matrices on the two datasets: (a) for Penn Action dataset; (b), (c), and (d) for three splits of the sub-JHMDB dataset.

of HM-FT, iDT, and pose. We observe that the iDT feature demonstrates higher accuracies than other trajectory features on both sub-JHMDB and Penn Action datasets, which outperforms HM-FT by 2.3% and 3.4%, respectively. As an improved version of DT, the accuracies of HM-FT on the two datasets are improved by 10.1% and 6.2%, which shows that the discrimination performance of original DT has been significantly enhanced after filtering out background trajectories. We also use two state-of-the-art saliency detection methods presented in Refs. 36 and 37 to generate masks individually and test the recognition performance on the two datasets. However, their recognition accuracies are significantly inferior to that of HM-FT, where the multiscale hybrid masks are exploited. It could be attributed to the failed saliency detections. Actually, due to the inherent challenges of saliency detection and the characteristics of action videos, where a frame does not necessarily contain a salient motion subject, saliency detection methods are not sufficient to provide reliable prior information for trajectory features without any auxiliary strategy.

Furthermore, the combination of HM-FT and iDT always performs better than each set of trajectories, but worse than the combination of trajectory features and pose features, which has achieved the best accuracies of 72.4% and 95.2%. Thus, we conclude that the proposed feature fusion framework can effectively exploit the complementarity

Table 1 Overall recognition performance of different methods for the sub-JHMDB and Penn Action datasets.

Methods	Sub-JHMDB (%)				Penn Action (%)
	Split 1	Split 2	Split 3	Average	
HM-FT	58.4	63.8	67.4	63.2	88.9
iDT	59.6	67.5	69.6	65.5	92.3
Pose	49.4	52.5	56.5	52.8	71.4
DT	52.8	48.8	57.6	53.1	82.7
Cheng et al.[36]	49.4	43.8	48.9	47.4	74.1
Lu et al.[37]	53.4	53.8	61.2	56.1	83.5
Comb. 1	66.3	70.0	68.5	68.3	93.3
Comb. 2	69.7	72.5	75.0	72.4	95.2

among the two types of features, thereby boosting the overall recognition performance.

4.2.3 *Comparison of the performance of different trajectory features*

The recognition results of each class based on different trajectory features are computed on the two datasets. For the sub-JHMDB dataset, to show comparison results intuitively, the recognition accuracy for a class is defined as the quotient of the number of samples that have been correctly classified and the total number of testing samples in all three splits.

As shown in Fig. 8, HM-FT achieves higher accuracies for 7 out of 12 classes on sub-JHMDB compared to DT, while the same results are obtained on "pick" and "run." In particular, the accuracy of "shoot ball" achieved by HM-FT is 100%, which outperforms DT by 75%. Moreover, HM-FT +iDT is better than HM-FT alone by 13.51% on average for seven classes of actions and only worse than it on "kick ball" and "swing baseball."

From Fig. 9, HM-FT+iDT achieves the highest recognition accuracies for almost all classes on Penn Action dataset,

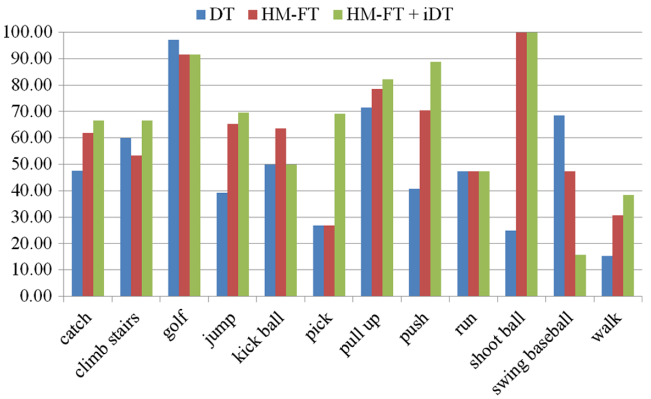

Fig. 8 Accuracy comparison of each class by DT, HM-FT, and HM-FT +iDT on sub-JHMDB.

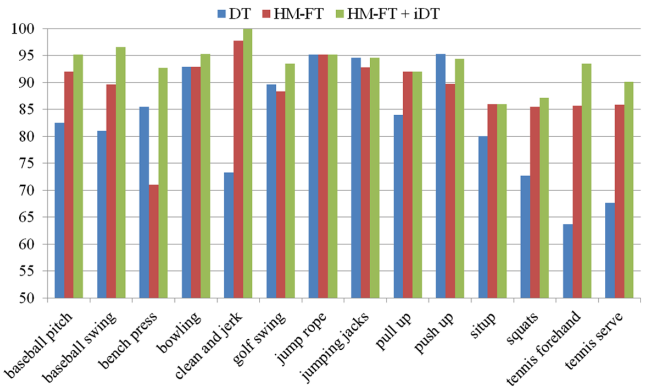

Fig. 9 Accuracy comparison of each class by DT, HM-FT, and HM-FT +iDT on Penn Action.

especially on the easily confusing "bench press," "tennis forehand," and "tennis serve." In addition, HM-FT is greater than or equal to DT on 10 classes, but gets much lower accuracy than DT on "bench press." By comparing the different trajectory features, we conclude that although the detections deviate from the foreground region in a few cases and lead to a decline in accuracy, HM-FT is more efficient than DT for most actions. In the vast majority of cases, the combination of HM-FT and iDT can always improve the classification power of HM-FT.

To visualize the computational cost of the proposed HM-FT, we compare its performance with three trajectory feature extraction methods in different aspects, including the time taken to process a video frame, the average number of trajectories per video clip, and the recognition accuracy. We randomly select 12 video clips from the sub-JHMDB dataset with a resolution of 320×240. There are 14 videos selected from the Penn Action dataset with a minimum resolution of 480×270 and a maximum resolution of 480×393.

From Table 2, since the computational cost of tracking sampled points decreases significantly by using DT-MB,[14] its time taken to process a video frame is the lowest. However, its recognition accuracy has not improved compared to DT. iDT-RCB[50] is an improved strategy based on DT, where the warped optical flow is exploited to adjust the interest points sampling to remove subtle motions. Although the recognition accuracy of DT is enhanced, its computational cost is higher than DT, which should be attributed to the calculation of optical flow and saliency detection. The proposed HM-FT further filters out invalid points by the multiscale hybrid masks to produce a minimum number of trajectories, which further reduces the computational cost of tracking points compared to DT-MB. However, since the moving foreground detection in the proposed scheme requires additional computational cost, the final computational cost of HM-FT is more than DT on the two datasets. This disadvantage will be decreased with the increasing of image resolution because more invalid points are removed, as we can see from Table 2. HM-FT has significantly improved the recognition accuracy of DT. Indeed, a limited reduction and efficient selection tend to improve the accuracy with minor computational cost. Taking into account the subsequent recognition procedure, fewer trajectories also lead to faster video encoding process.

Table 2 Comparison of HM-FT with other trajectory feature extraction methods.

Methods	Penn action			sub-JHMDB		
	Trajectories/clip	ms/frame	Accuracy	Trajectories/clip	ms/frame	Accuracy
DT[3]	12,198	378.79	82.7	6,487	278.74	53.1
DT-MB[14]	4601	290.70	83.0	1,584	210.08	52.4
iDT-RCB[50]	10,727	424.02	87.5	2,931	329.40	59.7
HM-FT	3071	389.64	88.9	1,249	306.71	63.2

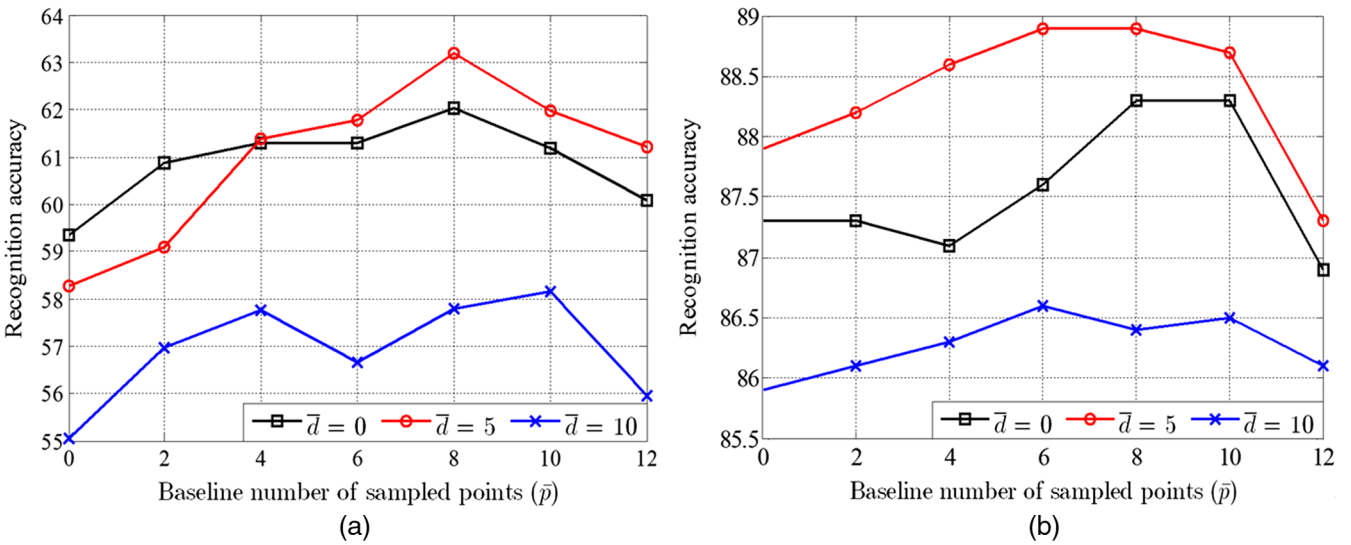

Fig. 10 Performance of HM-FT as a function of $\bar{p}$ and $\bar{d}$ on (a) sub-JHMDB and (b) Penn Action.

4.2.4 *Evaluation of the parameters for compensation schemes*

We evaluate the impact of the compensation scheme parameters on recognition performance. The relationships between the performance of HM-FT and the two parameters of the compensation schemes ($\bar{p}$ and $\bar{d}$) are respectively shown in Fig. 10. Overall, increasing the baseline number of trajectories from 0 to 5 on both datasets can improve performance. Instancing the baseline number of trajectories (from 5 to 10) yields significant performance degradation. In the cases of $\bar{p} \leq 8$ and $\bar{d} \leq 5$, increasing the baseline number of sampled points improves performance, likely because the samples with foreground detection deviation are corrected. We find that $\bar{p} = 8$ with $\bar{d} = 5$ provides a good tradeoff of performance versus computation.

4.2.5 *Comparison with the state-of-the-art*

The recognition accuracies achieved by our method are compared with the state-of-the-art methods on Penn Action and sub-JHMDB datasets, as shown in Table 3, where *F*-level indicates feature level fusion, and *S*-level indicates score level fusion. For Penn Action, the average accuracy achieved by this work is 95.2%, which has improved the state-of-the-art methods. For sub-JHMDB, only the work in Ref. 27, which combines iDT and a pose feature based on CNN, produces a better result than ours. However, if we replace the pose estimation results with the ground truth (GT) provided by the datasets, the recognition rate of "Comb. 2 (Pose-GT)" is 81.3%, which means that the insufficient of pose estimation does not affect our contributions in improving trajectory features and designing the hybrid multifeature fusion framework.

We find that the multifeature fusion strategies in Refs. 16, 21, 22, 27, 34, and 35 can always improve the recognition performance of single feature by integrating more abundant human motion information. Our method that benefits from the proposed HM-FT features and the appropriate fusion strategy has improved upon most of the similar algorithms. Moreover, although deep-learned methods have improved the state-of-the-art performance on many datasets based on the massive video data and large-scale training, they have no significant advantage over the handcrafted methods when the two datasets have less training data. The comparisons of deep-learned methods (i.e., P-CNN,[26] Pose,[27] iDT +Pose,[27] ARRNET,[28] and Deep Moving Poselets[29]) have confirmed this conclusion. From Ref. 27, we find that the

Table 3 Comparison of our method with the state-of-the-art methods.

Methods	Year	Penn Action	sub-JHMDB
Dense[21]	2013	—	46.0
Pose[27]	2017	79.0	61.5
Pose[51]	2016	—	55.4
iDT-FV[7]	2016	92.0	60.9
Dense + pose[21]	2013	—	52.9
PS-M + DT (*F*-level)[34]	2014	83.8	54.6
iDT + pose (*S*-level)[16]	2016	93.2	66.3
iDT + pose[27]	2017	92.9	74.6
WSF-DS[35]	2018	94.5	71.0
MST[52]	2014	74.0	45.3
ST-AOG[22]	2015	85.5	61.2
P-CNN[26]	2015	—	66.8
ARRNET[28]	2016	87.2	63.7
Deep moving poselets[29]	2017	—	70.2
Comb. 2 (pose-GT)		96.3	81.3
Comb. 2 (ours)		95.2	72.4

fusion strategy for the deep-learned features and handcrafted features at different levels is a promising research direction to improve recognition performance.

5 Conclusion

In this paper, a saliency-based sampling strategy (i.e., HM-FTs) and a hybrid multifeature fusion framework are proposed to improve the action recognition rate in realistic scenes efficiently. To obtain the trajectories closely related to action subject and filter out the trajectories derived from camera motion and inherent movements in the background, multiscale hybrid masks, which are generated by the weak saliency map optimized by the synchronous updating mechanism of cellular automata and the strong saliency map achieved through the MKB method, are utilized to refine the original dense sampling points. The collaborative optimization strategy is used to ensure that the foreground detection results are more reasonable and effective. The compensation schemes are employed to improve the fault tolerance of the proposed features. The experimental results show that the HM-FT feature has effectively improved the recognition performance of the original DT. Furthermore, the discriminative power of the overall recognition framework can be enhanced significantly using the hybrid feature fusion strategy.

However, during the experiments, we found that when the motion patterns and amplitudes of two types of actions are highly similar, neither trajectory features nor pose features can satisfactorily solve the confusion between their testing samples. In the future, we will focus on identifying critical objects in the scene that provide auxiliary discrimination information for action classification and trying to incorporate deep learning methods into the proposed framework to improve the recognition accuracy in realistic scenes.

Acknowledgments

This research was financially supported by the 2017 BJUT United Grand Scientific Research Program on Intelligent Manufacturing (No. 040000546317552) and the National Natural Science Foundation of China (Grant Nos. 61175087 and 61703012). The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. X. Tian and J. Fan, "Joints kinetic and relational features for action recognition," *Signal Process.* **142**, 412–422 (2018).
2. I. Laptev, "On space-time interest points," *Int. J. Comput. Vision* **64**(2), 107–123 (2005).
3. H. Wang et al., "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vision* **103**(1), 60–79 (2013).
4. H. Luo et al., "Human action recognition with group lasso regularized-support vector machine," *J. Electron. Imaging* **25**(3), 033015 (2016).
5. S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1234–1241 (2012).
6. H. Wang et al., "Action recognition by dense trajectories," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3169–3176 (2011).
7. H. Wang et al., "A robust and efficient video representation for action recognition," *Int. J. Comput. Vision* **119**(3), 219–238 (2016).
8. H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. of Int. Conf. on Computer Vision*, pp. 3551–3558 (2013).
9. A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(3), 257–267 (2001).
10. A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 984–989 (2005).
11. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEE Conf. on Computer Vision and Pattern Recognition*, pp. 886–893 (2005).
12. N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," *Lect. Notes Comput. Sci.* **3952**, 428–441 (2006).
13. H. Wang et al., "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conf.*, pp. 124.1–124.11 (2009).
14. X. J. Peng, Y. Qiao, and Q. Peng, "Motion boundary based sampling and 3-D co-occurrence descriptors for action recognition," *Image Vision Comput.* **32**(9), 616–628 (2014).
15. Y. Yi, Z. Zheng, and M. Lin, "Realistic action recognition with salient foreground trajectories," *Expert Syst. Appl.* **75**, 44–55 (2017).
16. X. Peng et al., "Bag of visual words and fusion methods for action recognition: comprehensive study and good practice," *Comput. Vision Image Understanding* **150**(C), 109–125 (2016).
17. J. Wang et al., "Locality-constrained linear coding for image classification," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3360–3367 (2010).
18. K. Matsui et al., "Trajectory-set feature for action recognition," *IEICE Trans. Inf. Syst.* **E100.D**(8), 1922–1924 (2017).
19. H. Jegou et al., "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(9), 1704–1716 (2012).
20. Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(12), 2878–2890 (2013).
21. H. Jhuang et al., "Towards understanding action recognition," in *Proc. of Int. Conf. on Computer Vision*, pp. 3192–3199 (2013).
22. B. X. Nie, C. Xiong, and S. C. Zhu, "Joint action recognition and pose estimation from video," in *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1293–1301 (2015).
23. S. Ji, M. Yang, and K. Yu, "3-D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **35** (1), 221–231 (2013).

24. A. Karpathy et al., "Large-scale video classification with convolutional neural networks," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1725–1732 (2014).
25. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Int. Conf. on Neural Information Processing Systems*, pp. 568–576 (2014).
26. G. Cheron, I. Laptev, and C. Schmid, "P-CNN: pose-based CNN features for action recognition," in *Proc. of Int. Conf. on Computer Vision*, pp. 3218–3226 (2015).
27. U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," in *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 438–445 (2017).
28. K. Krishnan, N. Prabhu, and R. V. Babu, "ARRNET: action recognition through recurrent neural networks," in *IEEE Int. Conf. on Signal Processing and Communications*, pp. 1–5 (2016).
29. E. Mavroudi and L. Tao, "Deep moving poselets for video based action recognition," in *IEEE Winter Conf. on Applications of Computer Vision*, pp. 111–120 (2017).
30. L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4305–4314 (2015).
31. L. Wang et al., "Temporal segment networks: towards good practices for deep action recognition," in *European Conf. on Computer Vision*, pp. 20–36 (2016).
32. C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1933–1941 (2016).
33. T. Du et al., "Learning spatiotemporal features with 3-D convolutional networks," in *Proc. of Int. Conf. on Computer Vision*, pp. 4489–4497 (2015).
34. L. Pishchulin, M. Andriluka, and B. Schiele, "Fine-grained activity recognition with holistic and pose based features," in *German Conf. on Pattern Recognition*, pp. 678–689 (2014).
35. G. L. Zhang, S. M. Jia, and X. Z. Li, "Weighted score-level feature fusion based on Dempster-Shafer evidence theory for action recognition," *J. Electron. Imaging* **27**(1), 013021 (2018).
36. M. M. Cheng et al., "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 569–582 (2015).
37. H. Lu et al., "Co-bootstrapping saliency," *IEEE Trans. Image Process.* **26**(1), 414–425 (2017).
38. N. Tong, H. Lu, and X. Ruan, "Salient object detection via bootstrap learning," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1884–1892 (2015).
39. K. M. He, J. Sun, and X. O. Tang, "Single image haze removal using dark channel prior," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1956–1963 (2009).
40. J. Von Neumann, "The general and logical theory of automata," in *Papers of John Von Neumann on Computing and Computer Theory*, Vol. 1, pp. 1–41 (1951).
41. F. Yang, H. Lu, and M. H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Trans. Circuits Syst. Video Technol.* **24**(2), 242–254 (2014).
42. W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: a strongly-supervised representation for detailed action understanding," in *Proc. of Int. Conf. on Computer Vision*, pp. 2248–2255 (2013).
43. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979).
44. R. Achanta, A. Shaji, and K. Smith, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012).
45. C. Yang et al., "Saliency detection via graph-based manifold ranking," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3166–3173 (2013).
46. Y. Qin et al., "Saliency detection via cellular automata," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 110–119 (2015).
47. S. M. Jia et al., "A novel improved probability-guided RANSAC algorithm for robot 3-D map building," *J. Sens.* **2016**(1), 1–18 (2016).
48. A. Vedaldi and B. Fulkerson, "VLFeat: an open and portable library of computer vision algorithms," in *Int. Conf. on Multimedia*, pp. 1469–1472 (2010).
49. S. Mathavan et al., "Fast segmentation of industrial quality pavement images using laws texture energy measures and k-means clustering," *J. Electron. Imaging* **25**(5), 053010 (2016).
50. Z. Xu et al., "Action recognition by saliency-based dense sampling," *Neurocomputing* **236**, 82–92 (2017).
51. S. Cao, K. Chen, and R. Nevatia, "Activity recognition and prediction with pose based discriminative patch model," in *IEEE Winter Conf. on Applications of Computer Vision*, pp. 1–9 (2016).
52. J. Wang et al., "Cross-view action modeling, learning and recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2649–2656 (2014).

Guoliang Zhang is currently a PhD candidate at the Faculty of Information Technology, Beijing University of Technology, China. His research interests include action recognition, machine learning, and man–machine interaction system of robots.

Songmin Jia received her PhD from the University of Electro-Communications, Japan, in 2002. Currently, she is a professor at the Faculty of Information Technology, Beijing University of Technology. Her research interests include distributed robotics, machine learning, visual computation, and image processing.

Xiangyin Zhang received his PhD from Beihang University, China, in 2016. Currently, he is a lecturer at the Faculty of Information Technology, Beijing University of Technology. His research interests include bionic intelligent computing theory and application, machine vision, and robot control theory.

Xiuzhi Li received his PhD from Beihang University, China, in 2008. Currently, he is an associate professor at the Faculty of Information Technology, Beijing University of Technology. His research interests include computer vision, 3-D image reconstruction, and mobile robot control and navigation.