electronics

MDPI

Article

Motion Video Recognition in Speeded-Up Robust Features Tracking

Jianguang Zhang [1], Yongxia Li [2,*], An Tai [3], Xianbin Wen [4] and Jianmin Jiang [5]

[1] The College of Mathematics and Computer Science, Hengshui University, Hengshui 053000, China
[2] Office of Academic Affaires, Hengshui University, Hengshui 053000, China
[3] The School of Computer Science and Technology, Hainan University, Haikou 570228, China
[4] The School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
[5] The School of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
* Correspondence: lynxzjg@hsnc.edu.cn

Abstract: Motion video recognition has been well explored in applications of computer vision. In this paper, we propose a novel video representation, which enhances motion recognition in videos based on SURF (Speeded-Up Robust Features) and two filters. Firstly, the detector scheme of SURF is used to detect the candidate points of the video because it is an efficient faster local feature detector. Secondly, by using the optical flow field and trajectory, the feature points can be filtered from the candidate points, which enables a robust and efficient extraction of motion feature points. Additionally, we introduce a descriptor, called MoSURF (Motion Speeded-Up Robust Features), based on SURF (Speeded-Up Robust Features), HOG (Histogram of Oriented Gradient), HOF (Histograms of Optical Flow), MBH(Motion Boundary Histograms), and trajectory information, which can effectively describe motion information and are complementary to each other. We evaluate our video representation under action classification on three motion video datasets namely KTH, YouTube, and UCF50. Compared with state-of-the-art methods, the proposed method shows advanced results on all datasets.

Keywords: motion recognition; speeded-up robust features; trajectory; filter

Citation: Zhang, J.; Li, Y.; Tai A.; Wen, X.; Jiang, J. Motion Video Recognition in Speeded-Up Robust Features Tracking. *Electronics* **2022**, *11*, 2959. https://doi.org/10.3390/electronics11182959

Academic Editor: Dah-Jye Lee

Received: 12 August 2022
Accepted: 14 September 2022
Published: 18 September 2022

1. Introduction

With the development of video capture technology and mobile internet, motion video data have grown massively. Motion video analysis has received more and more attention due to its wide applications, such as video data classification [1], video event monitoring [2], video content retrieval [3], surveillance video analysis [4], etc. For effective motion video analysis, motion recognition is a challenging task in video data now.

As for still image recognition, certain types of local features are proposed to characterize the spatial structure of still images or video frames. For example, Scale-Invariant Feature Transform (SIFT) [5] is proposed to characterize the spatial structure of still images. To improve the computational speed, Speeded-Up Robust Feature (SURF) [6] is presented. Meanwhile, SURF can be used to improve the extraction of interest points and the description of feature vectors. In principle, the larger the number of local features, the more the computational complexity and noise. By contrast, if there are too few local features, the discriminative information will be lost. Thus, it is important work to detect and select local features for motion recognition, because local features have the characteristics of repeatability, stability, and robustness when they are used to characterize video frames or still images.

Unlike the motion recognition methods for image analysis, the appearance and structure of the objects in video frames are constantly changing, and the rate of change is variable. Therefore, motion video recognition is not the recognition of a series of continuous motion images. Some methods for motion video recognition are proposed, considering that object

motion in videos is often different than the change in space and time. Histograms of Optical Flow (HOF) [7] and Motion Boundary Histogram (MBH) [8] are successfully applied in motion video recognition. To improve the performance of motion video recognition, these methods are often used in combination with other features, such as the Histogram of Oriented Gradients (HOG) [9]. Thus, the combined features, such as Spatio-Temporal Interest Points (STIP) [10] and Motion Scale Invariant Feature Transform (MoSIFT) [11] are proposed in motion video recognition.

In this paper, we proposed a new local feature descriptor for motion video recognition, namely Motion Speeded-Up Robust Feature (MoSURF). Firtly, a new SURF detection method is used to achieve better real-time performance in MoSURF. Secondly, MoSURF is a combination of multiple features, which uses a special trajectory strategy and filtering strategy to make the advantages of combined features more obvious. To evaluate the proposed video description, we perform action classification with a vector of locally aggregated descriptors (VLADs) [12] and different classifiers, such as Support Vector Machine (SVM) [13] and k-Nearest Neighbor (kNN). Furthermore, we compare different types of descriptors and study the computational complexity. Experiments on three datasets show that MoSURF outperforms several other state-of-the-art descriptors in the performance and computational time, which makes MoSURF more suitable for real-world applications.

Figure 1 is an application example of MoSURF. Each circle in Figure 1b represents a feature point, and the size of the circle represents the intensity of the feature point. When the color of different feature points is the same, it means that these feature points are in the same motion trajectory. The lengths of different motion trajectories are set to $\{2, 3, 4, 5\}$, which have shown empirically to give good results. The time consumption of this setting is 0.1 s in a computer with a 2.75 GHz CPU and 8G memory.

(**a**) (**b**)

Figure 1. Illustration of MoSURF tracking. It shows the No.230 frame captured from the video 'actionclipautoautotrain00385.avi' in the Hollywood2 dataset. (**a**) The initial picture including human movement. (**b**) MoSURF tracking on different trajectories.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The proposed MoSURF is detailed in Section 3. The datasets, the evaluation framework, and experimental analysis are described in Section 4. The conclusion is drawn in Section 5.

2. Related Work

In order to provide a sufficient benchmark and background for introducing our proposed research, we hereby briefly review the representative work on local spatial detectors and descriptors.

For still image recognition, some local spatial detectors and descriptors are proposed. In order to measure similarity between shapes, Belongie et al. proposed the ShapeContext [14] for object recognition. In [5], Lowe et al. proposed the SIFT to reliably match images under different views of an object or scene. In [15], Sande et al. proposed ColorSIFT by studying the invariance properties and the distinctiveness between color descriptors. In [6], SURF was proposed by using a Hessian matrix-based measure for the detector. In [16,17], SURF was compared with plenty of local spatial detectors and descriptors on public benchmark datasets and real-world datasets. The results show that the SURF detector is faster, and the SURF descriptor is more repeatable and more distinctive.

For motion video recognition, Laptev et al. [18] introduced spatial–temporal interest points by extending the Harris detector to the video domain. A few sets of features were detected. The reason is that a time-consuming iterative procedure has to be repeated for each feature candidate separately, and the iterative procedures are often diverged. To reduce the computation time, it is necessary to detect a small number of features. To effectively obtain the interest points, Doll'ar et al. [19] proposed to use local maxima on the space and time of the response function. However, their features are scale-variant. In [11], Chen et al. proposed MoSIFT to detect interest points and describe local features for human action recognition. For MoSIFT, interest point detection is based on spatial appearance and sufficient motions. The feature descriptor of MoSIFT captures both local appearance and motion. Thus, MoSIFT is the combination of HOG and HOF. MoSIFT interest points are scale-invariant in spatial domain. However, MoSIFT has the disadvantage of consuming more time and space, because the detector needs substantial motions, and the descriptor has many dimensions. From MoSIFT, we can see that a proper combination of different descriptors perform better than an individual descriptor.

As a typical method, optical flow is used to capture temporal features for motion video recognition. Dalal et al. [20] proposed the Motion Boundary Histograms (MBH) descriptor for human detection in videos. MBH can be obtained by computing derivatives separately for the horizontal (MBX) and vertical components (MBY) of the optical flow. MBH can reduce the false rate effectively. With the development of motion video recognition, tracking interest feature points through video sequences is a straightforward choice. Recently, certain methods [21–26] showed good results for action recognition by leveraging the motion information of trajectories.

To progress beyond these aforementioned methods, a novel method called MoSURF is proposed for motion video recognition in this paper. Our contributions are summarized as follows.

(1) The number of candidate points is fewer by using the detector scheme of SURF. Thus, it significantly reduces the time requirements to detect the feature point.

(2) The redundancy and noise can be eliminated by using the optical flow field filter and trajectory filter, which makes the feature more discriminative. Moreover, a novel trajectory strategy is used. So, subtler and more successive motion can be captured by filtering the candidate points and considering the relationship between trajectories.

(3) The descriptor of MoSURF is a combination of different types of descriptors, including trajectory, SURF, HOG, HOF, and MBH. SURF aligned with the trajectory, and HOG and HOF are used to characterize the shape, appearance, and motion, respectively. The MBH descriptor achieves a good performance for real-world videos containing a large amount of camera motion. Thus, these descriptors are complementary and make MoSURF more discriminative for real-world applications.

3. MoSURF

As mentioned above, MoSURF is designed for motion recognition in real-word videos, which includes lots of complex movements. This task is extremely difficult due to several challenges, such as background clutter, camera movement, occlusions, and illumination variations. Each of these challenges is the intractable problem for state-of-the-art computer vision technology. In this paper, we deal with these problems by using four major steps. Step I, local feature points are detected by applying the well-know method of SURF in the spatial domain. Step II, trajectories are used to capture movement by cascading these feature points in the temporal one. Step III, the effective spatial and temporal feature points are selected, and then both kinds of noises are suppressed in order to obtain better effectiveness. Step IV, the generated descriptors are aligned with trajectories. Details of MoSURF are described in the following sections.

3.1. Detection of Spatial Feature Points

In this section, the detector scheme of SURF is discussed briefly. Firstly, the 'Integral Image' has to be introduced before discussing SURF because it can improve the the performance of SURF. The integral image is computed rapidly from an input image. Meanwhile, the integral image is also used to speed up the calculation of box-type convolution filters. Given an input image I and a point $x = (I, J)$, the integral image $I_{\sum(x)}$ can be calculated by the Formula (1).

$$I_{\sum(x)} = \sum_{i=0}^{i \leq I} \sum_{j=0}^{j \leq J} I(i, j) \tag{1}$$

With the integral image, only three additions and four memory accesses are taken to calculate the sum of intensities inside a rectangular region of any size. For example, if we consider a rectangle bounded by vertices A, B, C, and D in Figure 2, the sum of pixel intensities is calculated by Formula (2). Since computation time is invariant to whatever the size of the area is, SURF makes good use of this property to perform convolutions of large-sized box filters at constant time. Thus, the calculation time will not be obviously increased even when a big filter is used.

$$\sum = A + D - (C + B) \tag{2}$$

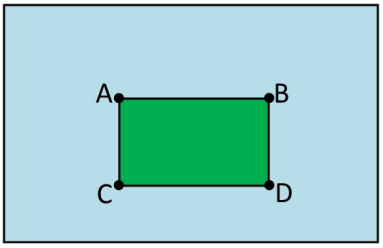


Figure 2. Area computation using integral images.

The other property of SURF is the Hessian matrix. For SURF, the spatial feature points are detected based on Hessian matrix approximation because of its good accuracy. The Hessian matrix $\mathrm{H}(x, \sigma)$ is described by the Formula (3).

$$\mathrm{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{3}$$

where x is a point of an image I, σ is the scale, and $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second–order derivative with image I at point x and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. Figure 3 illustrates the computation.

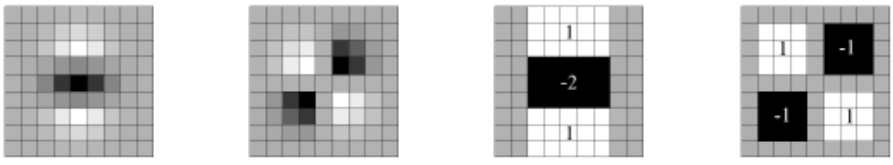


Figure 3. Laplacian of Gaussian approximation. From left to right: the Gaussian second–order partial derivative in the y- and xy-direction (L_{yy} and L_{xy}); weighted box filter approximations for the second–order Gaussian partial derivative in the y- and xy-direction (D_{yy} and D_{xy}).

By using integral images, the approximate second-order Gaussian derivatives in Figure 3 can be evaluated at a very low computational cost. Similarly for SURF, the Hessian determinant can be approximately calculated by using the Formula (4). The relative weight w of the filter responses is used to balance the expression for the Hessian's determinant

and is usually set to 0.9. For SURF, the local spatial feature points are detected by using the maxima of this function.

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (4)$$

where D_{xx} denotes the weighted box filter approximations for the second-order Gaussian partial derivative in the x-direction. Similarly to D_{yy} and D_{xy}, it can be obtained.

A non-maximal suppression method is used to calculate the local maxima. For this, each pixel is compared with its 26 neighbors, among which 8 points are in the native scale and 9 points are in the above or below scale, respectively. Figure 4 illustrates the non-maximal suppression method. At this stage, a set of local spatial features are detected. In order to detect the candidate points rapidly and efficiently, we use the detector scheme of SURF in MoSURF. Compared with other methods, the proposed MoSURF will cost less time and obtain more accuracy.

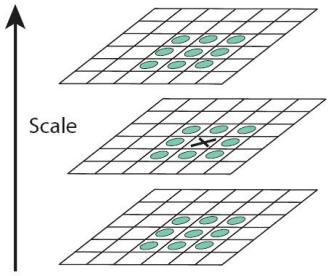


Figure 4. Non-maximal suppression. If it is greater than its 26 neighbors in 3×3 regions at the current and adjacent scales, the pixel marked 'X' is selected as a maxima.

3.2. Motion Capture

The motion in a video can be captured by tracking with trajectories. These trajectories include a series of SURF interest points in space–time domain, which are aligned with the movement in reality. Trajectories make the SURF interest points more meaningful for describing motion because these SURF interest points only characterize the space information. Furthermore, the motion is also captured based on optical flow. The optical flow is too sensitive for movement. Large amounts of optical flows will be generated even if the motion is a subtle movement. Thus, the motion information captured by optical flow is very abundant for us. In order to provide a sufficient benchmark and background for introducing our proposed research, we briefly review the optical flow and trajectory.

3.2.1. Optical Flow Field

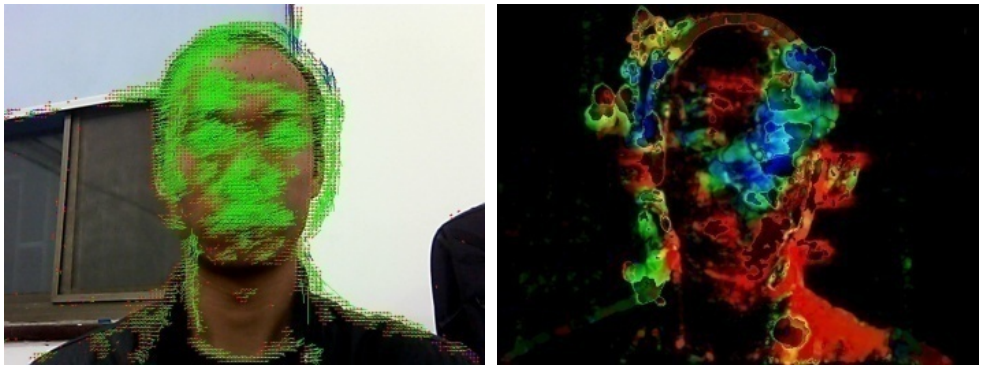
As mentioned above, optical flow is used as the basis of motion capture, and it represents the movement difference at each pixel where movement takes place between two consecutive frames. The optical flow can be caused by some kinds of movements, such as foreground object movement and background movement caused by camera moving. The optical flow field can be composed by all the optical flows of a frame. Figure 5 shows an example of 157 optical flow fields. From Figure 5, we can see that the optical flow is described by various color lines in the optical flow field. The optical flows are showed by every five-line interval at the horizontal and vertical directions, respectively. A large amount of optical flows are caused by both human running and shot moving.

To clearly describe the optical flow field, the other illustration of optical flow field is shown in Figure 6, in which the directions and the magnitudes of optical flows can be more easily observed. In Figure 6, the head of the human slightly moves to the upper left. In Figure 6a, the optical flows are shown by the green line. The red point represents the start, and the length of the green line represents the magnitude. In Figure 6b, the optical flows are shown by different colors. The red and green colors, respectively, represent the

left and right orientation, and the blue color represents the vertical orientation. The depth of color represents the magnitude. Form Figure 6, we can see that a large amount of optical flows are caused by slight human movement. In MoSURF, the optical flow field is used to initially filter the SURF points. This is introduced in Section 3.3.

(**a**) (**b**) (**c**)

Figure 5. Example of optical flow field. (**a**) Former frame. (**b**) Latter frame. (**c**) Optical flow field.



(**a**) (**b**)

Figure 6. Illustration of optical flow field.

3.2.2. Trajectory

A motion usually continues in several frames. To capture the motion, trajectories can be used to track the motion along several frames. Trajectories include both spatial and temporal information of a motion. Figure 7 shows an example of trajectories. In Figure 7, the rectangles represent frames, and the curves represent trajectories. Frames and trajectories are described with different colors, which is distinctive to the adjacent one. The black points represent the SURF points in each trajectory. A trajectory cannot grow any longer if it has no successive SURF point.

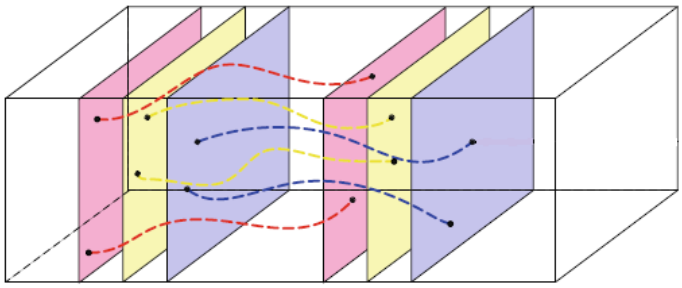


Figure 7. Trajectories between successive frames.

In [21], there are two disadvantages. First, the successive points are input to the the back of a trajectory directly. The redundancy and noise will be included in the input. Additionally, the computational complexity will be increased. Second, the descriptor information aligned with the trajectory output when the trajectory reaches its maximum size. This process is finished without considering the relationship between trajectories. Therefore, some motion information will be lost. In order to enhance the reasonability of trajectory and maintain the subtle motion information, we proposed new strategies for trajectory as follows:

(1) Input strategy.

We filter the candidate successive SURF points before inputting them to the back of a trajectory. We use ST to denote the SURF point at the back of a trajectory and SP to denote the candidate SURF point.

Firstly, we set a distance constrain condition in the range of [minFlowValue, maxFlow Value] to evaluate whether SP has a reasonable distance to ST, because a trajectory represents a time series of motions, and the motion distance between two successive frames is finite. Secondly, we set a gradient orientation constrain condition at the range of [−maxVarOrientation, +maxVarOrientation] to evaluate whether the gradient orientation of SP is within the range where maxVarOrientation is set to 120. This means that the characters of the two points must be similar. Thirdly, we set a moving direction constrain condition in the range of [−maxVarDirection, +maxVarDirection] to evaluate whether SP is reasonable by the moving direction where maxVarDirection is set to 90. The second and third constrain conditions are necessary. The main reason is that we can pick out the more appropriate succeeding SURF point from the severe intensive optical flows with abundant SURF points, especially when overlapping or occluded.

The second and third constrain conditions are shown in Figures 8 and 9, respectively. In Figure 8, the black arrow indicates the gradient orientation of each SURF point. The red two arrows indicate the criticality gradient orientations compared with that of the former one. The green line indicates the trajectory. In Figure 9, the black arrow indicates the instantaneous direction of the optical flow at each SURF point. The green arrow indicates the reasonable successive motion direction, while the red two arrows indicate the criticality directions compared with that of the former one. The value of maxVarDirection is set to 90 because the deviation of direction between two succeeding optical flows can hardly surpass this value even if the object moves with high velocity. Figure 10 illustrates an example of SURF trajectories. In Figure 10c, the running motion captured by SURF trajectories is shown. The SURF trajectories are described with various colors.

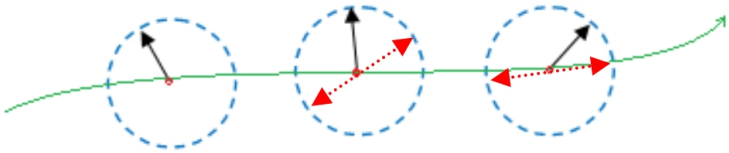


Figure 8. The example of the second constrain conditions.

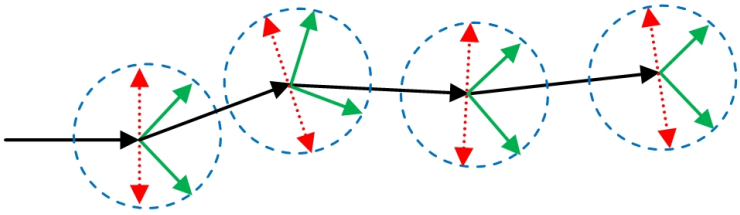


Figure 9. The example of the third constrain conditions.

(**a**) (**b**) (**c**)

Figure 10. Illustration of SURF trajectory. (**a**) The former frame. (**b**) The next frame. (**c**) Motion captured by SURF trajectories.

(2) Output strategy.

In [21], motion descriptors such as HOG, HOF, and MBH are encoded by a 3D space–time volume aligned with a trajectory. The structure of the volume includes $N \times N$ pixels around each feature point at the space domain and is maxTrackLength-frames-long at the time domain. We use this structure but carry out a different strategy. Firstly, the motion descriptors are output when the length of the trajectory comes to an output threshold value T_{out}. This process plays an important role for determining the degree of filtering, which is explained in Section 3.3. Secondly, when the trajectory reaches its maximum size, we need to perform two things: 1. We firstly delete this trajectory after output descriptors aligned with it and create a new trajectory, even if the successive SURF point comes. 2. We keep and clean this trajectory by reserving at least a remainder of one point at the back of the trajectory and deleting all the former points. Then, the remainder points are moved to the front of the trajectory and prepared for the next circulation. Figure 11 shows an example of this procedure. The size of remainder can be computed by the value of maxTrackLength and the value of output stride T_{step}. Figure 11a shows the waiting state before the size of the trajectory reaches the output threshold T_{out}. Figure 11b shows the outputting state with the stride of T_{step}. Figure 11c shows the remainder of the trajectory, which is represented by the blank space at the right of the trajectory. Figure 11d shows the fourth state, before which the trajectory is cleaned and prepared for next circulation.

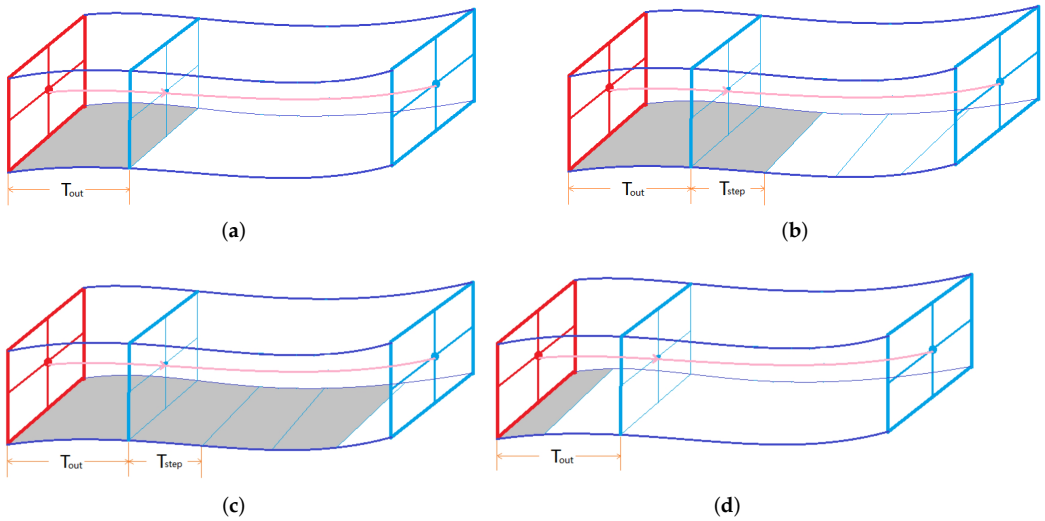


Figure 11. Illustration of output procedure for descriptors embedded in a 3D space–time volume aligned with a trajectory.

The output strategy makes the trajectory more reasonable, and it can keep the consistency of successiveness with that of motion in reality. The trajectory meets its end when there is no more successive SURF point to its last one.

3.3. Filter

To achieve the goal of MoSURF, something must firstly be carried out to make it have the capability to detect and track motion effectively. This requires that the detected features are actually able to describe the semantics of the videos. MoSURF can be achieved by using three filters.

(1) SURF points filter by optical flow field

For MoSURF, the optical flow field can be used to initially filter the candidate SURF points. The motion information is important for motion video recognition. Thus, the features must be relevant to the motion. We select the feature points from the candidate SURF points by two criteria about this initial filter. Firstly, there must be some optical flows in the space area around this feature point. Secondly, all the magnitudes of these optical flows need to be in a reasonable range [minFlowValue, maxFlowValue], where

we set the minFlowValue to 0.4, maxFlowValue to 50, and the size of the filter area to 5. Figure 12 shows the example of this filter. Figure 12a shows the initially detected SURF points. Figure 12b shows the optical flow field, and the interval is 4 pixels for x- and y-directions, respectively. Figure 12c shows the the SURF points filtered by optical flow field. The number of SURF points in Figure 12a is 810, while the number in Figure 12c is 187.

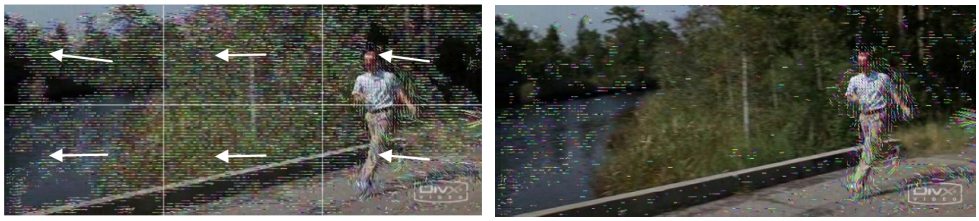
(**a**) (**b**) (**c**)

Figure 12. Illustration of the initial filter of SURF points by optical flow field. (**a**) Initial SURF points. (**b**) Initial optical flow field. (**c**) Initial filter by optical flow field.

(2) Optical flow field filter

The amount of optical flow between two successive frames may be huge. However,some of the optical flows may be unuseful or noise because they are caused by background movements, shot moving or illumination variation. It is significant to use the filter for optical flow field because the number of noises in the optical flow field may be far more than that of the foreground objects, especially when the shot is not stationary. In [10,11], the noises are only detected at the local area for optical flow. The points would be considered as noises when they are around a feature point with common characters, such as gradient orientation and magnitude, or common movement orientation. This procedure may eliminate the feature points when the foreground object moves in one direction integrally.

The filter method of MoSURF takes the whole factors into account and combines the whole and local movement circumstances. Thus, the optical flow is considered as noise when its direction and magnitude are simultaneously the same as that of most optical flows between the two successive frames. It is easy for humans to judge the optical flow noises by a glance, but this is not the case for computers because computers only discriminate the location and size for each optical flow. So, we use a statistics method to accomplish this task. Firstly, we divide a frame into four areas or six grids according to the frame size. The size of all divided grids are the same. Secondly, it is easy to count the average direction and its variance. If the variance is smaller than a threshold T_{var}, where T_{var} is set to 7, the direction may be caused by shot moving, and the range of its magnitude is that of optical flow noises. Figure 13 illustrates the optical flow field filter, and Figure 14 illustrates the SURF points after optimizing optical flow field.



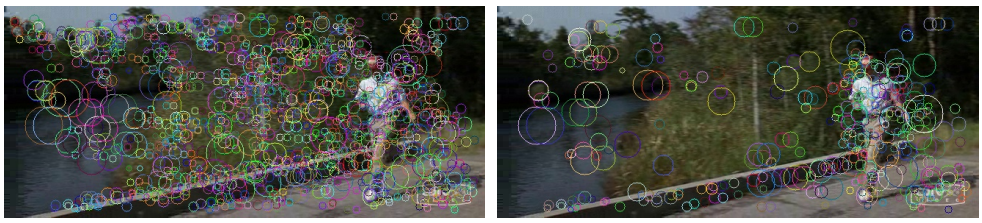
(**a**) (**b**)

Figure 13. Illustration of the optimized optical flow field. (**a**) Initial optical flow field. (**b**) Optimized optical flow field.

(3) SURF points filter by trajectory

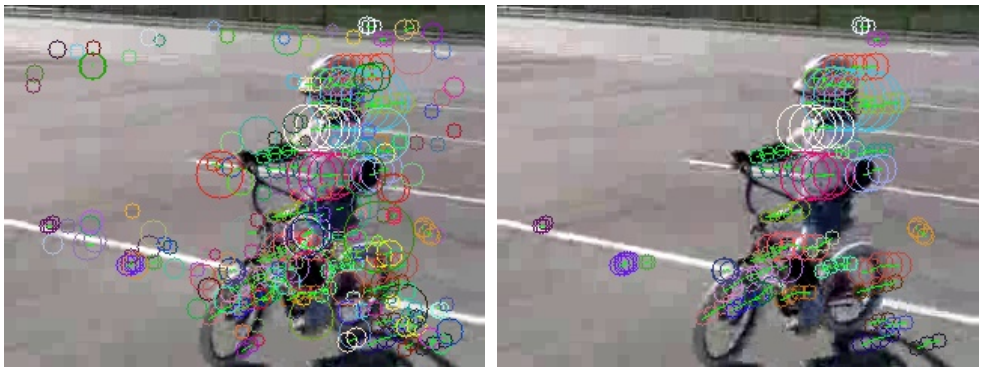
The SURF points in a trajectory should be effectively relevant to a motion, which means that the motion should have a successive SURF point in the next frame because motion cannot appear only in one frame. When a new frame comes, we firstly detect the SURF points by the initial filter introduced above. Then, we judge whether these points are added to trajectories as the successive feature points or create a new trajectory for each

of them by the trajectory input strategy introduced in Section 3.2.2. Thus, the length of trajectory is very important because it is used to decide whether the SURF point is or is not in a trajectory. For example, if the length of a trajectory is not more than 1, it may be regarded as noise caused by optical flow noise. Thus, the capacity of the filter is relevant to the set of the threshold value of trajectory length, and the accuracy of motion recognition is relevant to the output threshold value T_{out}. The two thresholds are the same because they are so tightly associated. Figure 15 illustrates the SURF points' filter by the trajectory filter.



(**a**) (**b**)

Figure 14. Illustration of SURF points after optimizing optical flow field. (**a**) Initial SURF points. (**b**) SURF points after optimizing optical flow filter.



(**a**) (**b**)

Figure 15. Illustration of SURF points filtered by only trajectories filtering. (The frame is the NO. 70 frame from the video "v_biking_01_01.avi" in the Hollywood2 dataset.) (**a**) Trajectory length from 1 to 5. (**b**) Trajectory length from 3 to 5.

3.4. Descriptors

In order to effectively described the motion information, we proposed a combined descriptor. The composition of the proposed descriptor includes SURF, trajectory, HOG, HOF, and MBH, which are complementary to each other.

(1) SURF

When we obtain the initial SURF points in every frame, there are some parameters, such as spatial octaves, intervals, and minimum strength threshold. In our experiment, spatial octaves are set to 3, intervals are set to 2, and the minimum strength threshold is set to 0.0001. We reduce the requirement of the first parameter because we pay less attention to the spatial effort, while we improve the requirements of the latter two parameters in order to obtain more subtle SURF points. The length of the SURF descriptor is 64.

(2) Trajectory

We set the maxTrackLength of trajectory to 15 frames fixed, and the volume around each SURF point at the space domain is set to $N \times N$ fixed optical flows, where N is set to 32. T_{out} and T_{step} are set to 2 and 1, respectively. The length of the descriptors of trajectory at each SURF point is fixed to 4. The descriptors are, respectively, X-coordinate, Y-coordinate, optical flow direction, and optical flow magnitude at each SURF point. The value of coordinates can be normalized by the frame size. The value of direction can be normalized to 360. The value of magnitude can be normalized to the maximum computed by maxFlowValue.

(3) HOG, HOF, and MBH

At each SURF point, we combine its descriptor with HOG, HOF, and MBH in an $N \times N$ spatial area around it, along with the trajectory. The area is subdivided into a grid of the size $n_\delta \times n_\delta$. The default value of N is set to 32, n_δ is set to 2, and the bin sizes of HOG, HOF, and MBH are set to 8, 9, and 8, respectively, as shown in [21]. Then, the size of the three descriptors are $2 \times 2 \times 8$ for HOG, $2 \times 2 \times 9$ for HOF, and $2 \times 2 \times 8$ for MBHX and MBHY, respectively. Additionally, the output size of descriptors is relevant to the value of the T_{step} in temporal dimension.

4. Experiment

4.1. Datasets

To evaluate the performances of motion video recognition, MoSURF is applied to video action recognition. For video action recognition, experiments are carried out by using three public available data sets, such as the KTH dataset, YouTube dataset, and UCF50 dataset, as shown in Figure 16. These datasets are collected from various sources, e.g., controlled experimental settings, Web videos, etc. Thus, the performance of our approach is investigated on diverse datasets with different resolutions, viewpoints, illumination changes, occlusion, background clutter, irregular motion, etc.

A. The KTH dataset [27]: The KTH database includes 600 videos in 6 action classes and 4 different scenarios. Additionally, 25 subjects perform several times for each action. The KTH dataset is a benchmark dataset.

B. The YouTube dataset [28]: The YouTube dataset contains a total of 1168 videos in 11 action categories. Most of the videos in this dataset are personal videos produced by users daily. Thus, YouTube is a real-world dataset.

C. The UCF50 dataset [29]: The UCF50 dataset contains 6618 video clips in 50 action categories. The action videos are daily life exercises and downloaded from the YouTube website. The UCF50 dataset can be considered as a big data set because the class numbers and video numbers are more than KTH and YouTube.

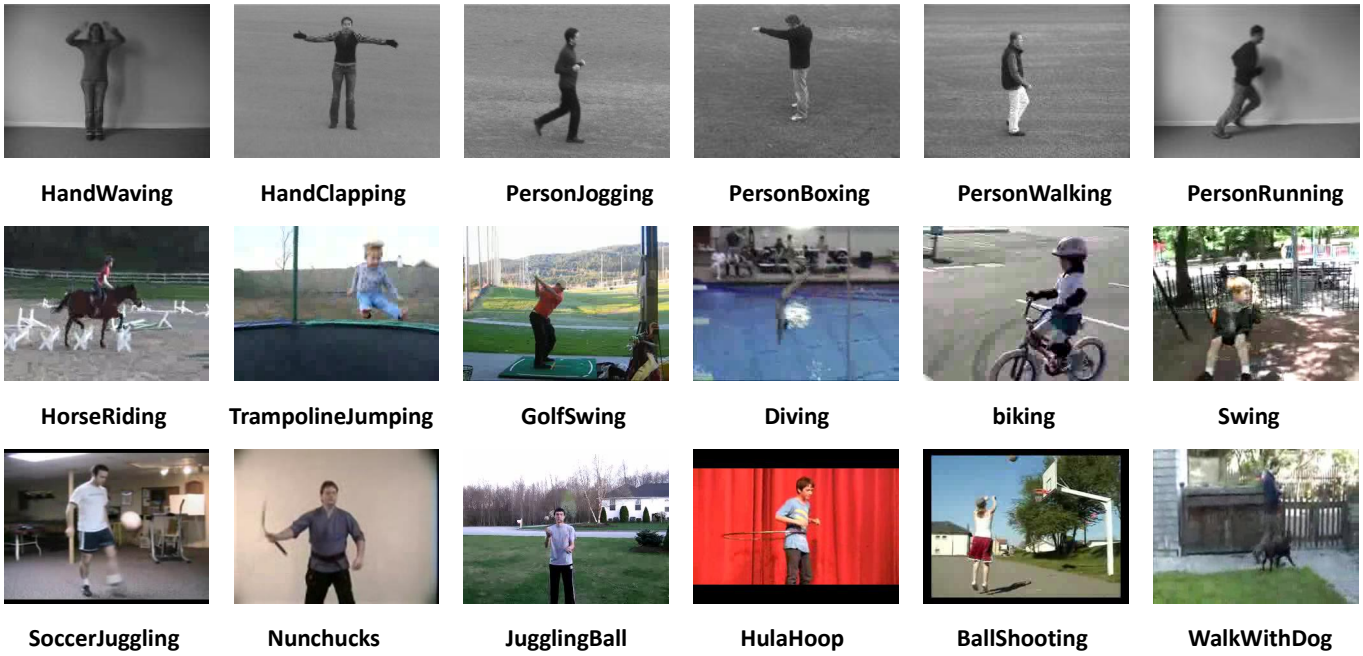


Figure 16. Sample frames from the three action recognition datasets in our experiments. From top to bottom: KTH, YouTube, and UCF50.

4.2. Evaluation Framework

The VLAD (Vector of Locally Aggregated Descriptors) [21] method is used to encode each descriptor, so as to form the feature representations of videos. Firstly, the codebook of each descriptor is constructed. The number of visual words per descriptor is set to 32, which has shown to empirically give good results for a wide range of datasets. To limit the complexity, we cluster a subset of 100,000 randomly selected training descriptors using k-means. To increase accuracy, we initialize k-means 10 times and keep the result with the lowest error. Descriptors are assigned to their closest vocabulary word using Euclidean distance. The resulting histograms of visual word occurrences are used as video representations.

For video action recognition, we use the non-linear SVM with a χ^2 kernel [13]. We use Average Accuracy (AA) over all action categories as the evaluation metric, which is defined as:

$$\mathrm{AA} = \frac{\sum_{k=1}^{c_v} acc_k}{c_v} \tag{5}$$

where c_v is the number of action classes. acc_k is the accuracy for the kth class. The best results are highlighted in bold.

4.3. Experimental Result

To evaluate the performance of MoSURF, MoSURF is compared with some types of descriptors, such as STIP, MoSIFT, and the variances of different descriptors of MoSURF (i.e., SURF, HOG, HOF, MBH, and TraSURF, where TraSURF is the combination descriptor of trajectory and SURF). For each dataset, $\{20, 30, 40, 50, 60\}$ videos are sampled from each class for training, and the remaining images are taken as the testing data. The sampling is repeated 10 times. With different numbers of sampling (NoS), the mean and standard derivation (std) values of the AA results are shown in Tables 1–3.

Table 1. Experimental results for different descriptors on KTH.

NoS	STIP	MoSIFT	SURF	TraSURF	HoG	HoF	MBH	MoSURF
20	85.46 ± 1.17%	78.78 ± 1.86%	62.08 ± 1.68%	69.50 ± 1.68%	73.17 ± 2.21%	84.75 ± 1.26%	84.39 ± 1.68%	**89.04 ± 1.29%**
30	88.95 ± 1.08%	80.48 ± 1.21%	66.62 ± 1.74%	73.17 ± 1.75%	75.52 ± 1.57%	87.93 ± 1.69%	87.88 ± 1.61%	**92.02 ± 1.05%**
40	89.92 ± 1.76%	82.89 ± 1.29%	67.97 ± 2.29%	74.42 ± 2.29%	77.58 ± 1.63%	89.86 ± 1.64%	89.82 ± 1.56%	**93.86 ± 1.17%**
50	91.58 ± 1.56%	84.17 ± 1.23%	69.83 ± 1.68%	75.80 ± 1.68%	78.40 ± 2.45%	90.40 ± 1.83%	90.33 ± 1.75%	**94.73 ± 1.31%**
60	91.77 ± 1.13%	86.50 ± 2.21%	70.71 ± 2.43%	77.00 ± 1.72%	81.79 ± 3.07%	91.08 ± 1.20%	90.91 ± 1.28%	**94.75 ± 1.29%**

Table 2. Experimental results for different descriptors on YouTube.

NoS	STIP	MoSIFT	SURF	TraSURF	HoG	HoF	MBH	MoSURF
20	54.60 ± 0.85%	57.81 ± 2.47%	57.11 ± 1.66%	60.73 ± 1.45%	60.59 ± 1.79%	54.09 ± 2.35%	54.76 ± 3.16%	**68.86 ± 2.63%**
30	57.74 ± 2.38%	61.53 ± 2.21%	61.50 ± 1.79%	65.25 ± 1.72%	64.62 ± 1.39%	56.27 ± 1.71%	56.98 ± 2.87%	**73.62 ± 1.48%**
40	62.10 ± 1.14%	65.62 ± 1.50%	64.72 ± 1.73%	68.68 ± 1.54%	68.03 ± 1.03%	59.53 ± 1.23%	60.17 ± 2.42%	**76.52 ± 0.90%**
50	64.98 ± 1.04%	68.43 ± 1.51%	69.09 ± 2.03%	72.09 ± 1.25%	71.83 ± 2.27%	61.44 ± 1.37%	62.10 ± 2.56%	**78.85 ± 0.86%**
60	67.21 ± 1.65%	70.85 ± 1.34%	71.06 ± 0.81%	74.33 ± 0.86%	74.38 ± 1.32%	62.45 ± 1.52%	62.98 ± 2.28%	**80.86 ± 1.38%**

Table 3. Experimental results for different descriptors on UCF50.

NoS	STIP	MoSIFT	SURF	TraSURF	HoG	HoF	MBH	MoSURF
20	54.78 ± 0.87%	58.43 ± 0.83%	53.56 ± 0.65%	56.98 ± 0.85%	57.52 ± 0.70%	49.10 ± 0.61%	50.21 ± 3.73%	**67.92 ± 0.74%**
30	59.79 ± 0.65%	63.14 ± 0.85%	59.32 ± 1.13%	62.71 ± 0.47%	63.22 ± 0.83%	54.59 ± 0.67%	55.53 ± 3.16%	**73.04 ± 0.80%**
40	63.00 ± 0.89%	66.52 ± 0.80%	62.61 ± 0.66%	65.64 ± 0.77%	66.85 ± 0.57%	56.34 ± 0.57%	57.34 ± 3.37%	**76.01 ± 0.90%**
50	66.08 ± 0.69%	69.09 ± 0.97%	65.83 ± 0.69%	68.85 ± 0.58%	70.26 ± 1.01%	58.97 ± 0.89%	59.79 ± 2.85%	**78.87 ± 0.97%**
60	67.48 ± 1.21%	70.83 ± 0.92%	68.02 ± 1.41%	70.86 ± 0.58%	72.12 ± 1.33%	60.23 ± 0.47%	60.99 ± 2.57%	**80.22 ± 0.58%**

From the experimental results in Tables 1–3, we see that:

(1) For all cases, our proposed MoSURF gains the best performances. It is noticed that MoSURF still achieves the highest accuracy even though the number of training samples is

small. Therefore, MoSURF can be used to effectively solve real-world problems because the labeled samples are often rare in the real world.

(2) For recognition on the KTH dataset, STIP descriptors outperform the MoSIFT descriptors because the KTH dataset has a clean background, and STIP carries more discriminative information. STIP does not perform well on YouTube and UCF50, as complex cluttered backgrounds degrade its discriminative power. MoSURF performs better than STIP and MoSIFT in most cases, no matter what the number of sampling is. Because the descriptor, detector, and filter are optimized, MoSURF can carry more discriminative information and eliminate more noise.

(3) For all cases on different datasets, TraSURF performs better than SURF. This indicates that trajectory adds the temporal information into TraSURF, which can describe the video motion more effectively.

(4) MoSURF, based on the combination descriptors, consistently achieves better performances than the individual components (i.e., SURF, HOG, HOF or MBH) of MoSURF on all the datasets. This indicates that MoSURF can improve performance by exploiting the complementary information among different descriptors.

The dense representation method has improved performances for motion video recognition. The representative existing method is based on dense trajectories and motion boundary descriptors, such as DenseTrajectories [21]. To evaluate the effectiveness of MoSURF, we compare MoSURF with STIP, MoSIFT, and DenseTrajectories on both the KTH dataset and YouTube dataset. To study the performance variances when the numbers of labeled data are different, the numbers of training videos are set to $\{20, 30, 40, 50, 60\}$ per class. The sampling is repeated 10 times, and the mean values of the AA results are reported as the evaluation results. The video recognition is achieved by performing the kNN (k = 10) classifier. The results are plotted in Figure 17. Figure 17 shows that the performance of MoSURF is generally better than that of STIP, MoSIFT, and DenseTrajectories for all the numbers of training videos per class. For the KTH dataset, DenseTrajectories achieves better results when many training samples are provided. However, MoSURF still achieves higher accuracy, especially when only few training samples are available. This advantage is especially desirable for real-world problems, as precisely annotated videos are scarce. For the YouTube dataset, the videos come from daily life. Additionally, our proposed MoSURF achieves the highest recognition accuracy in all cases, which indicates that MoSURF is more suitable for real-world applications.

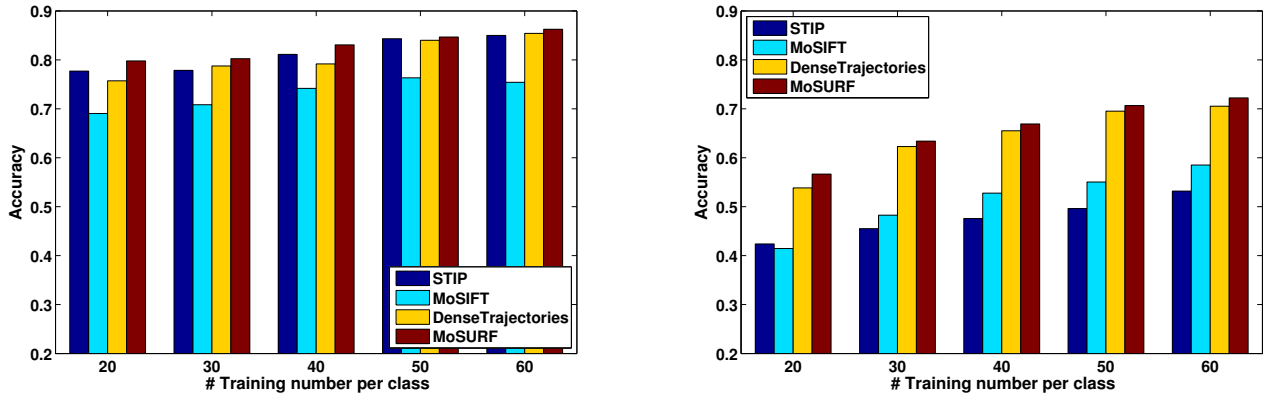


Figure 17. Performance comparison of STIP, MoSIFT, DenseTrajectories, and MoSURF when the numbers of training videos per class are set to 20, 30, 40, 50, and 60, respectively. For the two datasets, the results are obtained by performing the kNN (k = 10) classifier. (**a**) KTH dataset. (**b**) YouTube dataset.

To further investigate the effectiveness of the proposed MoSURF, the comparison accuracy of STIP, MoSIFT, and MoSURF for each class on UCF50 is additionally reported in Figure 18 when the number of sampling is 60. Figure 18 shows that the performance

of MoSURF is more stable, and it always gains good performance for different classes. It is also worth mentioning that the overall average performance is significantly low on certain categories of actions in the UCF50 dataset, e.g., the "BallShooting", "NunChucks", and "WalkWithDog" action classes. The main reason is that these actions are easily confused with other actions in UCF50. However, the proposed MoSURF has better results for the"BallShooting", "NunChucks", and "WalkWithDog" classes. This indicates that MoSURF can be effectively used to extract the discriminative spatial–time information from video data and enhance the performance of action recognition.

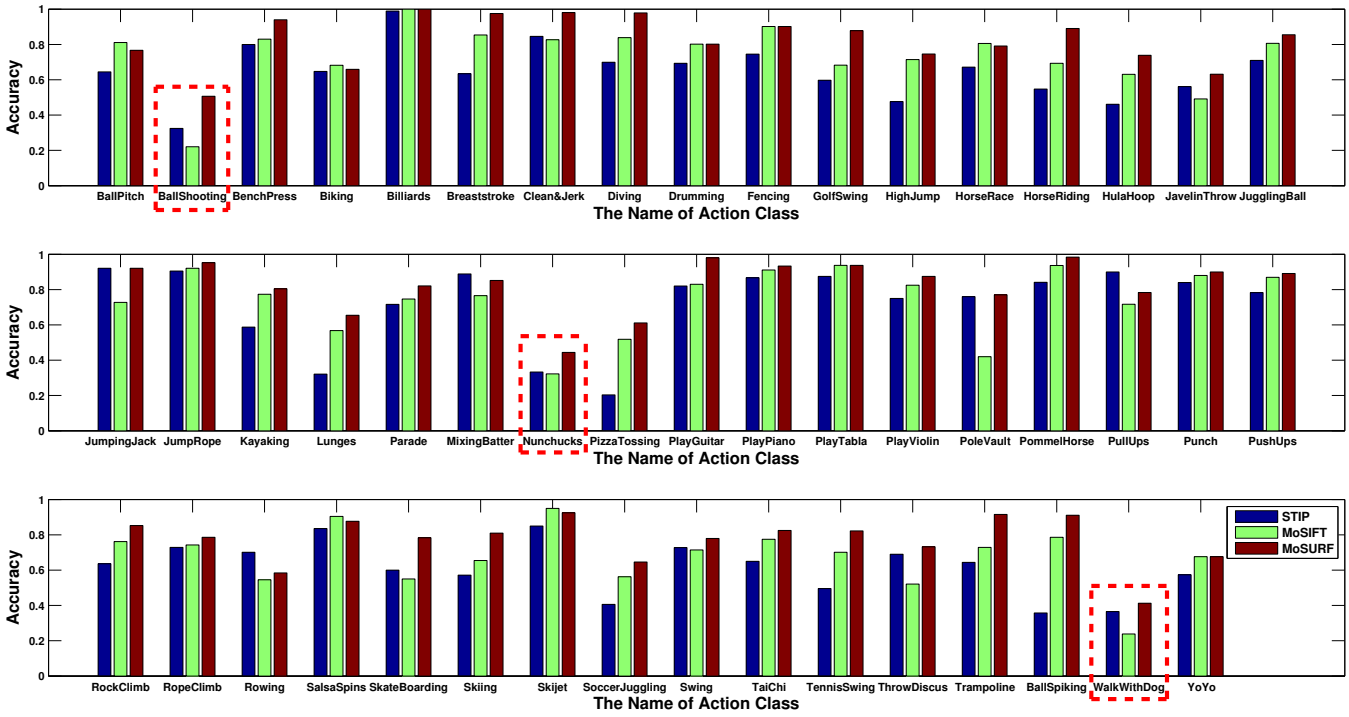


Figure 18. Recognition results of different descriptors. This figure shows the accuracy corresponding to each class of the UCF50 datasets in our experiments.

Since computational efficiency is very important for real applications, we show the results for the time of feature extraction. We compare MoSURF with STIP and MoSIFT since they have different processes of feature extraction. With the implementation of MATLAB, the extraction is made on a 3.4 GHZ Windows machine. The computational time of different methods on three datasets is listed in Table 4. Table 4 shows that our proposed MoSURF consumes much less time than STIP and MoSIFT on three datasets. The proposed MoSURF reduces the computational costs by 60.49%, 38.96%, and 41.11%, respectively. The main reason is that the detector scheme of SURF can effectively reduce computing time. The other reason is that the candidate points are filtered before inputting them to the back of a trajectory, which can reduce the computational complexity.

Table 4. Computation times on different datasets.

	KTH	YouTube	UCF50
STIP	14,273.16 s	48,937.94 s	276,365.23 s
MoSIFT	18,478.34 s	104,786.60 s	551,711.12 s
MoSURF	5638.83 s	29,869.96 s	162,759.33 s

5. Conclusions

In order to achieve good performance for motion video recognition, we propose a new feature called MoSURF, which can detect the candidate points faster based on the detector scheme of SURF. Meanwhile, it can effectively filter the noise and redundant

points by utilizing the optical flow field and trajectory. Especially, the more discriminative and successive motion can be captured by using a novel trajectory strategy. Finally, the descriptor of MoSURF is an effective combination of five typical descriptors. From the experimental results, we can see that the proposed MoSURF obtains better performance compared with other state-of-the-art methods.

Author Contributions: Conceptualization, J.Z. and Y.L.; methodology, Y.L.; software, J.Z. and A.T.; validation, J.Z.; writing—original draft preparation, J.Z. and Y.L.; writing—review and editing, X.W. and J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Chinese Natural Science Foundation (CNSF) (under Grant 61702165, Grant 61472278). This work was supported in part by the S&T Program of Hebei, China (under Grant F2020111001). This work was supported in part by the Scientific Research Project of Hengshui University (under Grant 2021GC17, Grant 2021yj18, Grant 2022SSW02). This work was supported in part by the Major project of Tianjin under Grant (under Grant 18ZXZNGX00150). This work was supported in part by the Natural Science Foundation of Tianjin (under Grant 18JCY-BJC84800).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sasithradevi, A.; Roomi, M.M. Video classification and retrieval through spatio-temporal Radon features. *Pattern Recognit.* **2019**, *99*, 107099. [CrossRef]
2. Wan, S.; Ding, S.; Chen, C. Edge Computing Enabled Video Segmentation for Real-Time Traffic Monitoring in Internet of Vehicles. *Pattern Recognit.* **2021**, *121*, 108146. [CrossRef]
3. Jing, C.; Dong, Z.; Pei, M.; Jia, Y. Heterogeneous Hashing Network for Face Retrieval Across Image and Video Domains. *IEEE Trans. Multimed.* **2019**, *21*, 782–794. [CrossRef]
4. Zhang, Z.; Nie, Y.; Sun, H.; Zhang, Q.; Xiao, M. Multi-View Video Synopsis via Simultaneous Object-Shifting and View-Switching Optimization. *IEEE Trans. Image Process.* **2019**, *29*, 971–985. [CrossRef] [PubMed]
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
6. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
7. Colque, R.; Caetano, C.; Andrade, M.; Schwartz, W.R. Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 673–682. [CrossRef]
8. Carmona, J.M.; Climent, J. Human action recognition by means of subtensor projections and dense trajectories. *Pattern Recognit.* **2018**, *81*, 443–455. [CrossRef]
9. Zuo, Z.; Yang, L.; Liu, Y.; Chao, F.; Song, R.; Qu, Y. Histogram of Fuzzy Local Spatio-Temporal Descriptors for Video Action Recognition. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4059–4067. [CrossRef]
10. Wang, H.; Ullah, M.M.; Klaser, A.; Laptev, I.; Schmid, C. Evaluation of local spatio-temporal features for action recognition. In Proceedings of the British Machine Vision Conference, Dundee, UK, 29 August–2 September 2011.
11. Chen, M.-Y.; Hauptmann, A. *MoSIFT: Recognizing Human Actions in Surveillance Videos*; Carnegie Mellon University: Pittsburgh, PA, USA, 2009.
12. Sun, Q.; Hong, L.; Ma, L.; Zhang, T. A novel hierarchical Bag-of-Words model for compact action representation. *Neurocomputing* **2015**, *174*, 722–732. [CrossRef]
13. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning Realistic Human Actions from Movies. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008.
14. Belongie, S.; Malik, J.; Puzicha, J. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [CrossRef]
15. van de Sande, K.; Gevers, T.; Snoek, C. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1582–1596. [CrossRef] [PubMed]
16. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005.
17. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *27*, 1615–1630. [CrossRef] [PubMed]
18. Lindeberg, L.T. Velocity adaptation of spatio-temporal receptive fields for direct recognition of activities: An experimental study. *Image Vis. Comput.* **2004**, *22*, 105–116.

19. Dollar, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005.
20. Dalal, N.; Triggs, B.; Schmid, C. Human Detection Using Oriented Histograms of Flow and Appearance. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006.
21. Wang, H.; Klser, A.; Schmid, C.; Liu, C.L. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
22. Matikainen, P.; Hebert, M.; Sukthankar, R. Trajectons: Action recognition through the motion analysis of tracked features. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 27 September–4 October 2009.
23. Messing, R.; Pal, C.; Kautz, H. Activity recognition using the velocity histories of tracked keypoints. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
24. Ju, S.; Xiao, W.; Yan, S.; Cheong, L.F.; Li, J. Hierarchical spatio-temporal context modeling for action recognition. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009.
25. Sun, J.; Mu, Y.; Yan, S.; Cheong, L.F. Activity recognition using dense long-duration trajectories. In Proceedings of the 2010 IEEE International Conference on Multimedia and Expo (ICME), Singapore, 19–23 July 2010.
26. Wang, H.; Schmid, C.; Liu, C.L. Action Recognition by Dense Trajectories. In Proceedings of the Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
27. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 26–26 August 2004.
28. Liu, J.; Luo, J.; Shah, M. Recognizing realistic actions from videos in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, Miami, FL, USA, 20–25 June 2009.
29. Reddy, K.K.; Shah, M. Recognizing 50 human action categories of web videos. *Mach. Vis. Appl.* **2012**, *24*, 971–981. [CrossRef]