Dense Trajectory Action Recognition Algorithm Based on Improved SURF

Hu Zhao1,*, Jianwu Dang2,3,a, Song Wang1,2,b, Yangping Wang2,3,c and Decheng Gao4, d

1School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China
2Gansu Provincial Engineering Research Center for Artificial Intelligence and Graphics & Image Processing, Lanzhou, China
3Gansu Provincial Key Lab of System Dynamics and Reliability of Rail Transport Equipment, Lanzhou, China
4Gansu Institute of Metrology, Lanzhou, China

*Corresponding author e-mail: zhaohuqxy@163.com, a dangjw@mail.lzjtu.cn, b 99403374@qq.com, c13519311970@163.com, d1451703192@qq.com

Abstract. In order to improve the time-consuming and large error problem of camera motion estimation in dense trajectory feature extraction of video, a dense trajectory action recognition algorithm based on Improved Speeded-Up Robust Features (SURF) is proposed. The algorithm mainly performs dense sampling of video images, and then executes camera motion estimation. In the feature point detection stage, the Gaussian pyramid layer was constructed dynamically to improve the real-time and accuracy of feature point extraction. Based on the SURF algorithm, the brightness center algorithm is used to obtain direction of feature. Binary Robust Independent Elementary Feature (BRIEF) is used to generate feature descriptors to determine matching points and optimized images, then to conduct feature tracking and feature extraction on the images to classify features. The experimental results show that the algorithm performs better in terms of speed when removing camera motion, and improves the real-time performance of feature extraction and the accuracy of action recognition.

1. Introduction

The action recognition technology [1-6] is a process of analyzing and processing a video containing human action through a computer, and realizing an action recognition of a video human body. The workflow can be divided into two parts: feature expression and action recognition. Feature expression is the feature that can extract the key information of human action in video. It is the basis of action classification and recognition. The quality of feature extraction directly affects the classification result.

Wang, Klaser [7] and others first proposed an action recognition algorithm based on dense trajectories. This method directly performs random sampling on dense grids, extracting trajectories and Motion Boundary Histograms (MBH), obtained high accuracy. Later, Wang, Schmid [8] and others proposed an action recognition algorithm based on the improved trajectory. This method improves the camera motion estimation, and uses the Fisher Vector method to encode the features, eliminating background interference and improving coding efficiency. Shao Yanhua [9] and others proposed an infrared human behaviour recognition method based on dense trajectory features. This method combines
dense trajectories with infrared video to solve the limitation of single-scale feature expression behavior. Zhao Xiao Jian [10] and others proposed an action recognition method based on dense optical flow trajectory and sparse coding algorithm. This method can obtain higher accuracy by using sparse feature representation of trajectory and overcome the complexity of traditional extraction features. Lu Tianran [11] and others proposed a human action recognition method based on saliency detection and dense trajectory. This method combines saliency detection and dense trajectory with high complexity.

The above action recognition methods based on dense trajectories [7, 8, 12] are all randomly sampled on dense grids. The features obtained high dimensions, and the computational complexity is large. The extracted trajectory features have more background redundant information, poor real-time performance. Therefore, in the stage of remove background interference information, dynamically construct the Gaussian pyramid layer and use the BRIEF descriptor on the traditional SURF algorithm can improve the real-time of feature extraction and the accuracy of action recognition.

2. Dense trajectory action recognition algorithm based on improved SURF

2.1. Dense sampling
Feature points are densely sampled on image I at multiple scales by meshing. The interval W of feature point sampling is usually taken as 5. Some feature points that are difficult to track are removed. The autocorrelation matrix is calculated for each pixel, and the eigenvalue are obtained. All the feature points below the threshold are removed. The threshold is determined by the following formula (1):

\[ T = 0.001 \times \max_{i \in I} \min(\lambda_1^i, \lambda_2^i) \]  

(1)

Where \((\lambda_1^i, \lambda_2^i)\) is the eigenvalue at pixel i, and 0.001 is an appropriate value experimentally derived. Figure 1 is an example effect of dense sampling of the UCF-101 data set v_TaiChi_g04_c01.

2.2. Improved camera motion estimation
After dense sampling of the image, camera motion estimation is needed to eliminate background interference information for better tracking of the trajectory and extraction of features. The improved camera motion estimation method flow is show in Figure 2.
An improved SURF algorithm is proposed to replace the original SURF algorithm. A method of constructing a Gaussian pyramid dynamically is introduced, the brightness center algorithm and the BRIEF descriptors are combined to improve the real-time performance of feature extraction. A schematic diagram of the improved SURF method is shown in Figure 3.

![Schematic diagram of improved SURF](image)

**Figure 3.** Schematic diagram of improved SURF

**2.2.1. Building Hessian Matrix.** When calculating the Hessian matrix [13], Gaussian filtering of the image I is first performed. f(x, y) is a Gaussian filtered function of the image I, and has a second derivative. The calculation formula of the Hessian matrix H is shown in (2):

\[
H(f(x, y)) = \begin{bmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\
\frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2}
\end{bmatrix}
\] (2)

The Hessian matrix discriminant formula is shown in (3):

\[
\det(H) = \frac{\partial^2 f}{\partial x^2} \left( \frac{\partial^2 f}{\partial y^2} \right) - \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2
\] (3)

The Hessian matrix of the image can be calculated by a Gaussian filter. The discriminant symbol can be used to determine whether the pixel is an extremum. When \( \det(H) < 0 \), the point \((x, y)\) is not an extremum. When \( \det(H) > 0 \), the point \((x, y)\) is the extremum.

**2.2.2. Dynamic construction of the Gaussian pyramid.** The Gaussian pyramid can be divided into many layers, each of which is called an Octave, denoted by O. Since the O values constructed by the SURF algorithm are all fixed values, the adaptive effect is not achieved. When the O value is small, the accuracy of the extracted feature points is not high; when the O value is large, feature extraction algorithm has poor real-time performance [14]. Therefore, through experimental verification, the definition of Octave in this paper is as shown in formula (4):

\[
O = \left\lfloor \log_2 \left( \frac{W}{0.5} \right) \right\rfloor - 3
\] (4)

\[
N = \begin{cases} 
G, & G \leq W \\
W, & G > W 
\end{cases}
\] (5)
In equations (4) and (5), G and W are the height and width of the input image, M is the fixed parameter 2, and N is obtained by (5). Since the value obtained in the formula (4) is large, it has been verified by experiments that it is reasonable to reduce the final value by 3. If the value of Octave calculated by equation (4) is within the interval $[3, 5]$, the calculated value is taken; if the value of Octave is greater than 5 or less than 3, the Octave value is taken as 5 or 3.

2.2.3. ORB feature descriptor. The main direction is determined using the brightness center algorithm [15-16] for the detected feature points. Let the centroid of the neighbourhood U be $C = (C_x, C_y)$, and the direction of the feature point to the centroid as main direction of the feature point. The centroid can be calculated by the moment in the neighborhood, $C_x = M_{1,0}/M_{0,0}, C_y = M_{0,1}/M_{0,0}$, so the main direction of the feature point is $\theta = arctan \left( \frac{M_{0,1}}{M_{1,0}} \right)$. The moment of the neighborhood is $M_{p,q} = \sum_{(x,y) \in U} x^p y^q I(x,y), I(x,y)$ is the gray-scale value of the pixel I.

The feature is described using the BRIEF [17] method to obtain a feature descriptor. The test criterion $\tau$ in the pixel neighbourhood $P$ defining the size of the $S \times S$ around the feature point is:

$$\tau(p; x, y) = \begin{cases} 1, & p(x) < p(y) \\ 0, & p(x) \geq p(y) \end{cases} \quad (6)$$

Where $P(x), P(y)$ are the gray-scale values of the pixel point x, y. Select n test point pairs (x,y) in the neighbourhood P, and obtain the n-dimensional binary bit string feature descriptor through the binary test criterion, as show in (7):

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (7)$$

2.2.4. Feature point matching. Since the feature descriptor produced by the BRIEF method is a binary string, the Hamming distance is used to match the extracted feature points.

The improved SURF feature and the optical flow feature are used to obtain a uniformly distributed matching point pair, and the projection transformation matrix $H$ is calculated by the gray-scale values of two adjacent frames, and the formula is as show in (8):

$$I_{t+1} = H \times I_t \quad (8)$$

Calculate the gray-scale value $I_{t+1}^{warp}$ after the camera motion is removed, as show in equation (9):

$$I_{t+1}^{warp} = H^{-1} \times I_{t+1} \quad (9)$$

Where $I_t$ and $I_{t+1}$ are the gray-scale values of two adjacent frames.

2.3. Feature extraction

Assume that the coordinates of a feature point in the previous step are $P_t = (x_t, y_t)$, the position of the feature point in the next frame image is calculated as show in the following equation (10):

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M \ast \omega_t) x_t, y_t \quad (10)$$

Where $\omega_t = (u_t, v_t)$ represents a dense optical flow field, $u_t$ and $v_t$ are components of the optical flow in the horizontal direction and the vertical direction. M is a 3*3 median filter.

The position of the feature points on the continuous L-frame image constitutes a trajectory $(P_t, P_{t+1}, \cdots, P_{t+k})$, and the subsequent feature extraction proceeds along the respective
trajectories of the feature points. In order to solve the drift phenomenon of the tracking, the feature
points are intensively sampled again every L frames, and tracking is performed again, L is 15.

For the trajectory of length L obtained from the previous step, the shape of the trajectory can
be described by \((\Delta P_1, ..., \Delta P_{t+L-1})\), where \(\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)\), the trajectory
feature descriptor is \(T = \frac{\sum_{t=1}^{L} \|\Delta P_t\|}{\sum_{t=1}^{L} \|\Delta P_t\|}\).

In the time-space body of \(N \times N \times L\) size around the feature point trajectory, a mesh division with
\(n_{\tau}\) in time and \(n_{\sigma}\) in space is performed, and the area of \(n_{\sigma} \times n_{\sigma} \times n_{\tau}\) can be separated. Feature
extraction is performed, setting \(N=32\), \(n_{\sigma}=2\), and \(n_{\tau}=3\). A schematic diagram of extracting HOF, HOG,
and MBH features is show in Figure 4.

![Figure 4. Schematic diagram of feature extraction](image)

![Figure 5. HOG feature extraction](image)

Example of HOG feature extraction for the punching action in the KTH dataset is show in Figure 5.

2.4. Action recognition
The feature is encoded using the Fisher Vector, and the obtained Fisher Vector is sent to the trained
SVM classifier to obtain the result of action recognition.

3. Experimental results and analysis
In order to verify the effectiveness of the improved SURF algorithm in this paper, the environment of
the simulation experiment in this paper is Windows 7 64-bit operating system, the processor is Xeon(R)
E3-1270, the main frequency is 3.60GHz, and the running memory is 16 GB. The programming
environment is Visual Studio 2013.

The SURF, ORB and ours improved SURF algorithm are used to detect the feature points of the
117th frame and the 124th frame of the first video v_Archery_g01_c01.avi of the Archery action in the
UCF-101 dataset, and the results of SURF are show in (a) of Figure 6, the results of ours improved
SURF are show in Figure (b), and the results of ORB are show in Figure (c).

![Figure 6. Algorithm matching results](image)
In order to verify the real-time performance of the improved SURF algorithm, two frames in the first video of Archery, Bowling, Baseball Pitch, Ice Dancing, and Table Tennis Shot in the UCF-101 dataset are selected. The matching time is show in Table 1.

### Table 1. Comparison of matching time

| Group | Ours SURF/ms | SURF/ms | ORB/ms |
|-------|--------------|---------|--------|
| 1     | 103          | 286     | 32     |
| 2     | 95           | 284     | 28     |
| 3     | 100          | 297     | 26     |
| 4     | 94           | 295     | 24     |
| 5     | 92           | 281     | 16     |
| Mean  | 96.8         | 288.6   | 25.2   |

In order to verify the accuracy of the improved SURF algorithm in this paper, two frames in the first video of Archery, Bowling, Baseball Pitch, Ice Dancing, and Table Tennis Shot in the UCF-101 dataset are selected. The matching accuracy is show in Table 2.

### Table 2. Comparison of matching accuracy

| Group | Ours SURF /% | SURF /% | ORB /% |
|-------|--------------|---------|--------|
| 1     | 88.6         | 85.6    | 64.1   |
| 2     | 89.4         | 85.8    | 62.9   |
| 3     | 90.2         | 86.1    | 63.8   |
| 4     | 89.7         | 87.2    | 63.2   |
| 5     | 88.9         | 86.5    | 62.8   |
| Mean  | 89.36        | 86.24   | 63.36  |

It can be seen from Table 1 that the average matching time of the improved SURF is 96.8ms, although it is 71.6ms slower than the ORB; but it is 191.8ms faster than the SURF, indicating the dynamic construction of the Gaussian pyramid and the use of the ORB feature descriptor can improve the speed of feature extraction. Due to the establishment of Gaussian pyramid, the proposed algorithm has scale invariance. The use of ORB feature descriptors makes the features obtained in this paper also have rotation invariance. It can be seen from Table 2 that the improved SURF matching accuracy is 3.12% higher than that of SURF and also higher than the ORB. Therefore, it can be concluded from the experiment that the algorithm improves the speed and accuracy of feature extraction.

The improved SURF accelerates the speed of feature extraction when eliminating camera motion. This paper experiments on the first video of boxing in the KTH dataset. In Figure 7, (a), (b) is the result of eliminating the camera motion, (c), (d) is the experimental result after eliminating camera movement.

![Figure 7. Comparison of camera motion results](image)

The Red Cross in the figure represents the densely sampled feature points, and the green graphic represents the calculated optical flow trajectory. It can be clearly seen from the above figure (a) and (b)
that there are some green optical flow trajectory around the human body due to camera motion. Figure (c) and (d) has no optical flow trajectory around the human body, so our algorithm can eliminate the trajectory interference caused by camera motion to a certain extent.

In order to verify the real-time performance of the dense trajectory action recognition algorithm based on improved SURF, the first video of boxing in the KTH dataset was selected for experiment. The time of feature extraction is show in Table 3.

**Table 3. Feature extraction time comparison**

| Group | Wang et al.[8] /s | Ours /s |
|-------|------------------|--------|
| 1     | 4.19             | 3.91   |
| 2     | 4.04             | 3.89   |
| 3     | 4.08             | 4.02   |
| 4     | 3.98             | 3.85   |
| 5     | 4.26             | 3.14   |
| Mean  | 4.110            | 3.762  |

In order to verify the accuracy of the dense trajectory action recognition algorithm based on improved SURF, experiments were carried out on the KTH, Hollywood2, and UCF Sports datasets. The recognition rate of each feature and the recognition rate after fusion of each feature are show in Table 4.

**Table 4. Comparison of feature recognition accuracy**

| Datasets       | Trajectory | HOG  | HOF  | MBH  | Wang et al.[7] | Wang et al.[8] | Ours |
|----------------|------------|------|------|------|----------------|----------------|------|
| KTH            | 90.2       | 86.5 | 93.2 | 95.0 | 93.4           | 94.2           | 94.8 |
| Hollywood2     | 47.7       | 41.5 | 50.8 | 54.2 | 54.6           | 58.3           | 59.2 |
| UCF Sports     | 75.2       | 83.8 | 77.6 | 84.8 | 82.1           | 88.2           | 89.3 |

It can be seen from Table 3 that the average time of feature extraction for applying the improved SURF algorithm to the dense trajectory is 3.762 s, which is 0.348 s faster than the dense trajectory using the SURF algorithm, indicating that ours improved action recognition algorithm has better real-time. It can be concluded from Table 4 that the recognition rate of the method on each dataset is 0.6, 0.9 and 1.1 percentage points higher than the literature [8], and the performance is better. We can see from the above two experiments that ours improved algorithm improves the speed of feature extraction in a certain sense, and improves the real-time and accuracy of action recognition.

4. Conclusion

In this paper, the improved SURF algorithm is used to extract matching points and eliminate camera motion. That can remove background track interference, and apply dense optical flow method to extract features of video, then to action recognition. The improved SURF algorithm uses dynamic methods to construct Gaussian pyramids, the brightness center algorithm and BRIEF are used to determine the main directions and descriptors of feature points. Therefore, the speed of feature extraction and the accuracy of action recognition can be accelerated to some extent in the action recognition.

Acknowledgments

This work was financially supported by National Natural Science Foundation of China (No. 61162016, 61562057), the Science and Technology Project of Gansu Colleges and Universities (No. 2017D-08) fund and Science and Technology Plan Project of Gansu (No. 18JR3RA104, 1504FKCA038) fund.

References

[1] Q. Hu, L. Qin, Q. M. Huang. A Survey on Vision Visual Human Action Recognition.Chinese Journal of Computers, 2013, 36 (12): 2512 - 2524.
[2] L. Qin, Q. Hu, Q. M. Huang, et al. Action Recognition Using Trajectories of Spatio-Temporal Feature Point. Chinese Journal of Computers, 2014, 37 (6): 1281 - 1288.

[3] R. F. Li, L. L. Wang, K. Wang. A Survey of Human Body Action Recognition. Pattern Recognition and Artificial Intelligence, 2014, 27 (1): 35 - 48.

[4] Q. Lei, D. S. Chen, S. Z. Li. Advances on Human Action Recognition in Realistic Scenes. Computer Science, 2014, 41 (12): 1 - 7.

[5] K. Q. Huang, X. T. Chen, Y. F. Kang, et al. Intelligent Visual Surveillance: A Review. Chinese Journal of Computers, 2015, 38 (6): 1093 - 1118.

[6] J. G. Feng, J. Xiao. View-invariant action recognition: a survey. Journal of Image and Graphics, 2013, 18 (2): 157 - 168.

[7] H. Wang, A. Klaser, C. Schmid, et al. Action Recognition by Dense Trajectories. IEEE Conference on Computer Vision & Pattern Recognition, Colorado Springs, United States: IEEE, 2011: 3169 - 3176.

[8] H. Wang, C. Schmid. Action Recognition with Improved Trajectories. ICCV-IEEE International Conference on Computer Vision, Sydney Australia: IEEE, 2013: 3551 - 3558.

[9] Y. H. Shao, Y. C. Guo, C. Gao. Infrared human action recognition using dense trajectories-based feature. Journal of Optoelectronics-Laster, 2015, 26 (04): 758 - 763.

[10] X. J. Zhao, X. Q. Zen. Action recognition method based on dense optical flow trajectory and sparse coding algorithm. Journal of Computer Application, 2016, 36 (1): 181 - 187.

[11] T. R. Lu, F. Q. Yu, H. Z. Yang, et al. Human action recognition based on dense trajectories with saliency detection. Computer Engineering and Application, 2018, 54 (14): 163 - 167.

[12] H. Wang, A. Klaser, C. Schmid, et al. Dense trajectories and motion boundary descriptors for action recognition [J]. International Journal of Computer Vision, 2013, 103 (1): 60 – 79.

[13] H. Bay, A. Ess, T. Tuytelaars, et al. Speeded-up Robust Features (SURF). Computer Vision and Image Understanding, 2008, 110 (3): 346 - 359.

[14] H. B. Li, Y. Y. Zhao, Y. Wu, et al. Improved SURF Algorithm Based on Distance Constraint. Journal of System Simulation, 2014, 26 (12): 2944 - 2956.

[15] B. X. Bai, J. Che, X. K. Mou, et al. Improved feature points matching algorithm based on speed-up robust feature and oriented fast and rotated brief. Journal of Computer Applications, 2016, 36 (7): 1923 - 1926.

[16] E. Rublee, V. Rabaud, K. Konolige, et al. ORB: An Efficient Alternative to SIFT or SURF. ICCV-IEEE International Conference on Computer Vision, USA: IEEE, 2011: 2564 - 2571.

[17] M. Calonder, V. Lepetit, C. Strecha, et al. BRIEF: Binary Robust Independent Elementary Feature. In European Conference on Computer Vision. Germany: Springer Berlin Heidelberg, 2010: 778 - 792.