Action Recognition in Realistic Scenes via Local Spatio-temporal Representation

Qing Lei\textsuperscript{a,b,c}, Shaozi Li\textsuperscript{a,b,*}, Hongbo Zhang\textsuperscript{a,b}

\textsuperscript{a}Cognitive Science Department, Xiamen University, Xiamen 361005, China
\textsuperscript{b}Fujian Key Laboratory of the Brain-like Intelligent Systems, Xiamen 361005, China
\textsuperscript{c}College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

Abstract

In this paper, we consider the problem of human action recognition in realistic scenes. Mainstream methods adopt local spatio-temporal patterns matching strategy that detect STIPs (Spatio-Temporal Interest Points) and compute features from raw video frames, and then classify the features into some predefined action categories. However, these researches have made certain assumptions seldom hold in the real-world environment such as small scale, without viewpoint changes and background motion. To obtain more effective detection and description for human action videos of complex scenes, we improve the performance of STIPs detection in realistic scenes by filtering out noises resulted from low resolution or cluttered background. In addition, we propose a new feature description method that uses 3D local self-similarities to compensate for motion estimation error due to camera motion. BoW (Bag-of-Words) model is applied to learn a codebook in training stage, and a nonlinear SVM classifier is used for action recognition finally. Experimental results show that our approach achieve superior performance on more challenge dataset (Youtube) in comparison to baseline method.

Keywords: Human Action Recognition; Spatio-temporal Interest Points; Local Features; Realistic Scene; Bag-of-Words Action Model

1 Introduction

Automatically and robustly recognizing human actions in image or video sequences is of great importance and has a widely application prospect such as identity authentication, video retrieval, intelligent surveillance and human-computer interaction. However, accurate recognition of actions is a highly challenging task because it is influenced by various aspects such as inter-class variation, background clutter, low resolution, occlusion, variation of view and illumination etc. Most of the

*Project supported by the National Nature Science Foundation of China (No. 61202143), Doctoral Program Foundation of Institutions of Higher Education of China (No. 20090121110032), Shenzhen Science and Technology Research Foundation (No. JC200903180630A, ZYB200907110169A), the Fundamental Research Funds for the Central Universities of Huaqiao University (No. 11QZR04).
*Corresponding author.
Email address: szlig@xmu.edu.cn (Shaozi Li).
current approaches are either attempting to compute effective features from raw video frames [1] or trying to learn a powerful codebook for action representation [2, 3]. These methods follow a two-step approach in which the first step detects and describes human motion features in some effective ways, and the second step learns classifiers to classify these obtained features into some predefined categories [4, 5, 6, 7].

Some methods treat video frames as a collection of 2D templates (e.g., Motion Energy Image (MEI) or Motion History Image (MHI) [8]) or 3D space-time shape volumes [9], and use template matching technique for action recognition. In recent years, local features matching strategy based on Spatial-temporal Interest Points (STIPs) detection is proposed for action recognition [10, 11, 12]. Main works of this strategy include: (1) STIPs detection: design a saliency function and compute on raw video frames to find positions of abrupt changes both in time and space. (2) Feature description: extract effective descriptions of the detected interest points to represent the occurred events. (3) Classification: train a linear or nonlinear classifier for recognition.

Laptev and Lindeberg [13, 14, 15] extend the Harris corner detection algorithm to 3D and use it to detect features in videos, and then compute HOG (histogram of gradient)/HOF (histogram of optical flow) descriptors to represent actions. Finally a SVM classifier is used for recognition. Dollar etc. [16] propose a Gabor and Gaussian mixed filtering detection algorithm and calculate local N-jet features of extracted cuboids, then PCA is applied for dimensionality reduction and use 1-nearest neighbor classifier with 2 distance for classification. Sun [17] present a hierarchical spatio-temporal context modeling for action recognition that extract SIFT features and three-tier contexts of point trajectories, and use multi-channel nonlinear SVM classifiers for recognition.

Although the STIP-based methods have been verified in several scenes, but there are some limitations. Most of these approaches make certain assumptions such as small scale, without viewpoint changes and background motion. However, these assumptions seldom hold in the real-world environment. How to construct a robust human action recognition method to deal with occlusion, illumination variation, camera motion and cluttered background in complex scenes has been an extremely difficult problem and remains unresolved.

In this work, we improve STIP-based method for human action recognition in real-world scenarios. In realistic scenes, there are many STIPs in the background due to camera motion or background motion, but according to a generally known, only the STIPs of the target are useful. So we proposed two filtering rules to refine the STIP detecting results by removing the points belong to background. And the other problem in realistic scenes is the camera capturing video often need motion and change viewpoints. This may lead to the motion estimation error by optical flow algorithm which is widely used for STIP feature representation such as Histogram of Optical Flow (HOF). For solving the problem, we propose a more effective feature description method which combined HNF (histogram of gradient and optical flow) features with 3D self-similarities of local patches. Self-similarities have been proved capable of adapting to geometry deformation, illuminate variation and slightly view variation in object recognition [18]. Finally, we apply a Bag-of-Words (BoW) approach to encode the descriptors and train a SVM classifier for recognition.

The rest of this paper is organized as follows. In Section 2, we introduce the baseline framework of STIP feature detection and BoW action representation. In Section 3, we describe our points refinement method dealing with noises resulted from cluttered background, followed by our feature description based on local self-similarities. The experimental results are presented in Section 4. Finally, we provide concluding remarks and future research in Section 5.
2 Human Action Recognition Based on STIP and BOW Model

2.1 STIP Detection

It is observed that events frequently occurred in positions with abrupt changes both in time and space. How to detect accurate interest points is of vital importance. We use Gabor and Gaussian mixed filtering detection algorithm proposed by Dollar [16] which calculate convolutions separately in spatial domain based on Gaussian filter and in time based on Gabor filtering. The response function has the form:

\[ R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \]  

(1)

where \( g(x, y; \sigma^2) = \frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}} \), and \( g(x, y; \sigma^2) \) is the 2D Gaussian smoothing kernel applied only along the spatial dimensions, and \( h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-\frac{t^2}{\tau^2}} \) and \( h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-\frac{t^2}{\tau^2}} \) are a quadrature pair of 1D Gabor filters applied temporally. The two parameters \( \sigma \) and \( \tau \) correspond to the spatial and temporal scale of the detector.

We use a multi-scale detecting strategy to detect points with levels of \( \sigma_i^2 = 2^{\frac{i+1}{2}}, (i = 1, \cdots, 6) \) and \( \tau_j^2 = 2^{\frac{j}{2}}, (j = 1, 2) \) and \( \omega = \frac{4}{\tau} \). Some detecting results are shown in the following figures. From Fig. 1 and Fig. 2 we can see that accurate detecting results have acquired on Weizmann and KTH datasets due to the clean background and static view point. However, points belong to background also detected on YouTube dataset in Fig. 3. It is mainly caused by low resolution, cluttered background and camera motion on this challenge dataset.

Fig. 1: STIPs detection results of sampled consecutive frames of waving action on Weizmann datasets (interesting point (white circle) selected by satisfying the maximum response values)

2.2 BOW Model for Human Action Recognition

The framework of Bag-of-Words (BoW) model [18, 19] is outlined in Fig. 4. In training stage, accurate STIPs are detected from videos and then effective features are computed to describe these points. Video sequence is represented by the statistic histogram of a set of “visual words”, where clustering method is applied to learn a codebook from all feature vectors, and then Vector-quantization (VQ) is applied to encode the descriptors based on the learned codebook, finally
Fig. 2: STIPs detection results of sampled consecutive frames of running action on KTH datasets (interesting point (white circle) selected by satisfying the maximum response values)

Fig. 3: STIPs detection results of sampled consecutive frames of biking action on YouTube datasets (interesting point (red circle) selected by satisfying the maximum response values)

Fig. 4: Action recognition framework based on BoW modeling and nonlinear classifier
summarizes the distribution of vectors by a set of “visual words”. The codes are average pooled within global region and passed to SVM classifier.

We apply BoW representation for action recognition, which utilize unsupervised k-means algorithm for clustering and learn a codebook from the extracted feature vectors. Specifically, two different ways of codebook learning are used in experiments: learning one unique codebook by clustering on all training feature vectors, and learning a set of action-specific codebooks by clustering on feature vectors for different action categories. We show in experiments (Section 4) that learning a set of action-specific codebooks achieve better accuracy.

For a testing sample, we detect accurate STIPs \( P = \{p_1, p_2, \cdots, p_s\} \) and compute the descriptions to obtain the feature sets \( F_i = \{f_1, f_2, \cdots, f_m\} \), then classify the feature vectors based on Euclidian distance into visual words of visual codebook \( V = \{w_1, w_2, \cdots, w_N\} \), where each visual word is expressed as \( w_i = \{f_1, f_2, \cdots, f_m\} \). Afterwards, average pooling is applied to encode features by computing statistic histogram \( H = \{h_1, h_2, \cdots, h_N\} \) where \( K \) is the dimension of codebook, \( h_i \) is the ith word’s occurrence frequency \( (i = 1 \cdots N) \). Finally, codes are passed into a trained SVM classifier with \( \chi^2 \) kernel as shown in Eq. (2) for recognition.

\[
K(H_i, H_j) = \exp \left( -\frac{1}{2A} \sum_{m=1}^{N} \frac{(h_{im} - h_{jm})^2}{h_{im} + h_{jm}} \right)
\]  

(2)

where \( H_i = \{h_{i1}, h_{i2}, \cdots, h_{in}\} \) and \( H_j = \{h_{j1}, h_{j2}, \cdots, h_{jn}\} \) are the center occurrence frequency histograms, \( N \) is the dimensions of codebook, \( A \) is the average distance of all training examples.

To improve the performance of STIPs detection in realistic scenes, we propose a points-refinement method to remove noise belongs to background, and construct a new feature descriptor to deal with challenges about human appearance deformation and camera motion. The details are introduced in the following.

3 Our Approach

3.1 Points Refinement

STIPs detection method has a major defect that it computes saliency within local regions rather than global region, thus a large number of points belongs to background are also detected. The first work of our research is to develop a points-refinement method to improve the performance of STIPs detection on videos of realistic scenes. Our points-refinement algorithm is based on two observations: (1) from the appearance characteristic of image, we remove points of straight lines exist in venue, housing walls and other similar positions of background. (2) From the motion characteristic of 3D patches, if a spatial-temporal patch is small enough, all pixels within a patch move with a single uniform direction. While few patches located at motion discontinuities or contained an abrupt change in motion direction or velocity disobey this phenomenon. We preserve points as a center of these non-uniform movement patches. According to these two observations, we propose our points-refinement algorithm:

Algorithm 1: remove straight line points

Input: \( STIPs = \{p(x_i, y_i, t_i, \sigma_i, \tau_i, H)\} \), \( x_i, y_i, t_i \) are the corresponding coordinates of ith point in three dimensions. \( \sigma_i \) and \( \tau_i \) denote the scale factors of gaussian filtering in space and gabor filtering in time. Video sequences: V.
Process: For each point $p(x_i, y_i, t_i) \in STIPs$ do

$L_{xx} = (V[x_i+1][y_i][t_i] - V[x_i][y_i][t_i]) - (V[x_i][y_i][t_i] - V[x_i-1][y_i][t_i]);$
$L_y = V[x_i][y_i+1][t_i] - V[x_i][y_i][t_i];$

if ($L_{xx} \times L_y > \epsilon$) then STIPs = STIPs - {p};

End if

End For

Return (STIPs).

Algorithm 2: remove points within an uniform-movement patch
Input: $STIPs = \{p(x_i, y_i, t_i, \sigma_i, \tau_i, H)\}$. Video sequences: $V$. Cuboids size: $w$. Optical flow algorithm: cvCalcOpticalFlowLK.

Process:

For each point $p(x_i, y_i, t_i) \in STIPs$ do

flag=0;

build a cuboid cube $[w][w][w]$ centered at $(x_i, y_i, t_i)$;

ang = cvCalcOpticalFlowLK $(x_i, y_i, t_i)$;

For each pixel cube $[w_1][w_2][w_3]$ do

calculate the optical flow orientation: ang'←cvCalcOpticalFlowLK;

if (ang!=ang') then flag=1; break;
end if

End For

if !flag then then STIPs = STIPs - {p};

End if

End for

Return (STIPs).

3.2 Local Self-similarities Descriptor

Self-similarities of visual entity that refers to the repetition of local intensity pattern in nearby region have been proved to be stable on local intensity pattern variation, small affine deformation and view changes [20, 21]. The similarity of images relies on similar local layout of local intensity patterns rather than common low-level visual properties such as colors, edges or gradients. We utilize the robust property of self-similarities to compensate for the motion estimation error by optical flow algorithm which is widely used for STIP feature representation such as Histogram of Optical Flow (HOF). The method treats a video as a collection of 3D local space-time patches, and extracts a set of 3D local blocks centered at the obtained interest points. Then self-similarities descriptors of 3D blocks are computed according to the procedures described in algorithm 3. The generation process of self-similarities descriptor is outlined in Fig. 5.

Algorithm 3: local self-similarities descriptor computation algorithm
Input: $STIPs = \{p(x_i, y_i, t_i, \sigma_i, \tau_i, H)\}$. Video sequences: $V$. related window radius: $W$. refer-
ence volume size: $Q (Q \ll W)$

**Output**: local self-similarities descriptor $H = \{h_i\}$

**Process**: For each point $q(x_q, y_q, t_q) \in STIPs$ do

**Step 1** Extract the query volume centered at $q$ and size is $W$.

**Step 2** Build a reference volume centered at $q$ with window size $Q$.

**Step 3** Calculate the sum of squared difference of a given position $(x, y, t)$ in query volume according to Eq. (3), then generate the distance volume $SSD_q$.

$$SSD_q(x, y, t) = \sum_{i=-Q}^{Q} \sum_{j=-Q}^{Q} \sum_{k=-Q}^{Q} (v(x_q+i, y_q+j, t_q+k) - v(x_q-x+i, y_q-y+j, t_q-t+k))^2$$ (3)

where $-W \leq x \leq W, -W \leq y \leq W, -W \leq z \leq W$.

**Step 4** Normalization: The distance volume is normalized and transforms into related volume $C_q$ according to Eq. (4).

$$C_q(x, y, t) = \exp \left( \frac{-SSD_q(x, y, t)}{\max(\text{var}_\text{noise}, \text{var}_\text{auto}(q))} \right)$$ (4)

where $\text{var}_\text{auto}(q) = SSD_{\text{max}}$ is the probable maximum intensity variation accumulated on all pixels in the query volume, $\text{var}_\text{noise} = \text{patchAreanChannels}SSD_{\text{max}}$ is the maximal variation of intensity of all pixels in this volume.

**Step 5** Divide the 3D volume into a set of 3d blocks expressed as $(n_x, n_y, n_t)$.

**Step 6** Calculate the spatial-temporal gradient orientation $\theta$ and $\phi$ for each pixel and interpolate voted into 2-dimension histogram $\text{hist}(i_\theta, i_\phi)$ according to Eq. (5, 6, 7).

$$m_{3D}(x, y, t) = \sqrt{L_x^2 + L_y^2 + L_t^2}$$ (5)

$$\theta(x, y, t) = \arctan \left( \frac{L_y}{L_x} \right)$$ (6)

$$\phi(x, y, t) = \arctan \left( \frac{L_t}{\sqrt{L_x^2 + L_y^2}} \right)$$ (7)
\[ \text{hist}(i_{\theta_1}, i_{\phi_1}) + = \omega_1, \text{hist}(i_{\theta_2}, i_{\phi_2}) + = \omega_2, \] where \( L_x = V(x + 1, y, t) - V(x - 1, y, t), L_y = V(x, y + 1, t) - V(x, y - 1, t), L_t = V(x, y, t + 1) - V(x, y, t - 1), \) \( \omega_1 \) and \( \omega_2 \) are weights assigned for the nearest two-bins after interpolate.

Step 7 Concatenate the histograms of all blocks to form the final descriptor \( H_q \) of point \( q \).

End For

Return \(( H = h_i )\).

4 Experimental Results

4.1 Human Action Datasets

We evaluate our approach on two benchmark datasets: KTH and YouTube [22]. KTH contains 600 videos of 6 action categories including walking, jogging, running, boxing, handwaving, handclapping performed by 25 subjects in four scenes (outdoors, outdoors with scale variation, outdoors with different clothes and indoors). The average length of videos is 4 minutes with resolution \( 160 \times 120 \), and frame rate 25 fps. UCF YouTube Action Dataset contains 25 subjects and 11 action categories including basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog.

4.2 Performance Evaluation of Points Refinement

A detecting results comparison between STIPs and our points refinement algorithm is shown in Fig. 6. We can find that points belongs to background resulted from low resolution (biking), camera movement (diving) and cluttered backgrounds (spiking and walk_dog) have been successfully removed in our methods.

Table 1 show that our points refinement method improve the average precision on Youtube about 6\%. But average accuracy is decreased on KTH probably since this datasets have a relatively clean background and points refinement method has not obvious effect in this circumstance when some useful features also had been filtered out by such rules.

<table>
<thead>
<tr>
<th>dataset</th>
<th>STIP (HNF features)</th>
<th>STIP (HNF features)+Point Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH</td>
<td>91.83%</td>
<td>85.5%</td>
</tr>
<tr>
<td>YouTube</td>
<td>53.91%</td>
<td>60.18%</td>
</tr>
</tbody>
</table>

4.3 Performance Evaluation of Feature Description

During all experiments on two datasets we followed the common setup that divide the datasets into training and testing samples according to different subjects. Each experiment we choose
four videos of $n_1$ subjects for training and actions of remaining $n_2 - n_1$ ($n_2$ is the total number of subjects) subjects for testing. Then the training sets contains $n_1 \times 4$ videos and the testing sets contains $(n_2 - n_1) \times 4$ videos. The performance is measured using average accuracy over all subjects. We compared our result with Laptev’s HNF descriptor [15]. In Laptev’s method, HNF feature is used which concatenate the HOG feature and the HOF feature. HOG feature is computed in the neighborhood of detected points. The size of each volume is related to the detection scales. Each volume is subdivided into a $3 \times 3 \times 2$ grid of patches. Orientations of the gradient are divided in 4 bins, and it generates a 72-dimensional feature vector to describe each interesting point. HOF feature is also computed, while orientations of motion flow are divided into 5 bins, so it generates a 90-dimensional feature vector for each point. Finally, HNF feature concatenates two vectors and becomes a 162-dimensional feature vector.

In Table 2 we list our action classification accuracy in comparison with Laptev’s HNF descriptor. As seen from Table 2, our descriptor can achieve similar performance on KTH dataset. Moreover, a significant performance improvement is achieved on more challenge YouTube dataset. Confusion matrixes on two datasets are shown in Fig. 7 and Fig. 8. Observed on confusion matrix, it can be seen that the average accuracy have been significant increased about 32%, 23% and 25% respectively for juggle, shooting and walk_dog, while a relative low improvement on jumping, spiking and riding as 10%, 13% and 15%. It’s probably a consequence of our points refinement method that remove points belongs to cluttered background, also some useful information are also removed. Specifically, hundreds of interest points detected for action contained relatively clean background such as jumping and riding, and almost all of these points are useful for classification. However, some of these useful points are filtered out by points refinement rules. When detecting on videos of juggle, shooting and walk_dog, thousands of or more interest points are detected for each video. Most of these points are generated from low resolution or camera motion and useless for classification. In this circumstance our approach is obvious more useful.
Table 2: Average Precisions of two action descriptions on KTH and Youtube datasets

<table>
<thead>
<tr>
<th>dataset</th>
<th>KTH</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>HNF (one codebook)</td>
<td>88.83%</td>
<td>53.91%</td>
</tr>
<tr>
<td>HNF (action-specific codebooks)</td>
<td>91.83%</td>
<td>55.80%</td>
</tr>
<tr>
<td>HNF+Self, Similarities (one codebook)</td>
<td>88.33%</td>
<td>70.24%</td>
</tr>
<tr>
<td>HNF+Self, Similarities (action-specific codebooks)</td>
<td>90.5%</td>
<td>72.28%</td>
</tr>
</tbody>
</table>

Fig. 7: The confusion matrixes of STIP (left) and our approach (right) on KTH

Fig. 8: The confusion matrixes of STIP (left) and our approach (right) on YouTube

5 Conclusion

This paper explores the effectiveness of local spatio-temporal pattern matching obtained by STIP detection and BoW representation for human action recognition in realistic scenes. We develop points refinement method to improve STIPs detection performance in realistic scene. Then we propose a new feature description that uses 3D local self-similarity to complement HNF features to deal with camera motion and view changes. We evaluate our approach on the KTH and the Youtube datasets. Results show that our method outperforms compared methods on the Youtube data and achieves competitive performance on the KTH data, demonstrating its superior performance in real-world environments.

The BoW approach discards the spatial order of local descriptors, which severely limits the descriptive power of the action representation. However, correlations of local features are useful for interactive human-human action or human-object actions. It would be reasonable to construct hierarchical model to encode these important information. By overcoming this problem, we will explore other hierarchical action models in the future.

References


