Action Recognition Based on Spatial-Temporal Pyramid Sparse Coding

Xiaojing Zhang, Hua Zhang, Xiaochun Cao *
School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
zxj.green@gmail.com, huazhang@tju.edu.cn, xcao@tju.edu.cn

Abstract

This paper introduces a novel video presentation term spatial-temporal pyramid sparse coding (STPSC) which characterizes both the spatial and temporal aspects of the video. Specifically, the co-occurrences of visual words are computed with respect to the spatial layout and the sequencing of the features in the video. The representation captures both the spatial arrangement and the temporal relationship of the words.

Our representation is motivated by the technology spatial pyramid matching (SPM) which is used to recognize scenes in the image. We extend SPM to video analysis combining with sparse coding. Firstly, dense feature points are extracted and represented by displacement information from a dense optical flow field. Then sparse coding is used to quantize the feature descriptors, and the spatial-temporal pyramid is introduced to represent an action. Finally, we use SVM to classify the videos. Experimental results showed improvements over the state-of-the-art techniques on the public action dataset.

1. Introduction

Action recognition from videos is a very hot research topic in computer vision thanks to their many potential applications including surveillance, human-computer interaction, video retrieval. The traditional action recognition framework is based on BoF (bag of features) model [2]. The video descriptors are firstly extracted to construct a dictionary. Then we quantize each descriptor into a histogram to represent the action. Finally, a classifier is used to recognize the actions. Image classification uses a similar framework. There are a lot of common features between action recognition and image classification. Since a video can be considered as a sequence of images, techniques in image recognition field can be used in video analysis. Choi et al. [1] extend image scene classification technique, i.e. spatial pyramid matching [3], to spatial-temporal pyramid. In the recent success [9], Yang et al. introduce a new method based on sparse codes of features instead of the K-means vector quantization in [3]. Inspired by their work [3], we further improve the spatial-temporal pyramid by replacing K-means with sparse coding.

In the related work [1], the authors firstly detect shot boundary, and then split the video into several shots. Two kinds of descriptors are combined to describe the video shot. There are two main differences between their method [1] and ours. First, the application domain is different, we want to recognize actions in the video while their purpose is video retrieval. More importantly, we use the sparse coding [9] instead of K-means for vector quantization considering its lower reconstruction error and better representation.

Our method is also related to [7], which densely samples feature points on each frame, and describes each feature point by the displacement information in the optical flow field. K-means is introduced to quantize the descriptors. A non-linear classifier is utilized to classify

*Corresponding author
the action videos. Although this method uses dense feature points to represent the video, it did not consider the spatial-temporal structure of the video.

To enhance the performance of [7] and construct efficient action recognition system, we introduce a novel action recognition method which is spatial-temporal pyramid sparse coding (STPSC). Firstly, dense feature points are sampled and are described by trajectories [7]. To efficiently construct the dictionary and obtain lower reconstruction errors, sparse coding is utilized. After the dictionary is established, we use a soft spatial pooling method to quantize the feature descriptors. A non-linear SVM classifier is introduced to classify the videos. The process of our method is shown in Fig. 1. In this paper, our main contributions are extending spatial pyramid matching in [9] to video analysis and show how the parameters of our method influence the recognition results.

2. Method

2.1 Feature trajectory extraction

Feature points are extracted based on a grid space by \( W \) pixels. As above-mentioned, we use the displacement of the feature points in optical flow field as our feature descriptors similar to [7]. So each feature point \( P_{t,i}(x_{t,i}, y_{t,i}) \) \( (i = 1, 2, ..., M) \) where \( (x_{t,i}, y_{t,i}) \) is the location of the feature points and \( M \) is the number of feature points at frames \( t \) which is tracked to the next frame \( t + 1 \) by a median filter in the optical flow field \( \Theta_{t}(x, y) = (∆x, ∆y) \)

\[
P_{t+1,i} = P_{t,i} + F \ast \Theta_{t}(x - \bar{x}_{t,i}, y - \bar{y}_{t,i}),
\]

where \( F \) is the median filter kernel and \( (\bar{x}_{t,i}, \bar{y}_{t,i}) \) is the rounded position of \( (x_{t,i}, y_{t,i}) \). Then the trajectory of the feature point can be represented as \( P_{t+i}, P_{t+1+i}, ..., P_{t+n} \). Although we have obtained the trajectory of each feature point in most cases, there are still two challenges to be addressed: the drifting and the flat region in the frame. To reduce the impact of the drifting problem, the length of a trajectory is set to \( L \) (In our experiment, \( L \) is set to 15). This is based on the intuition that the trajectories can accumulate the errors. So we need to set the appropriate length of trajectory. To avoid sampling feature points in a flat region, the autocorrelation matrix of each feature point is computed. If the smaller eigenvalue is less than the threshold, the feature point will be ignored. Finally, the descriptor of the trajectory is represented as \( d_{t,i} = (\Delta P_{t,i}, \Delta P_{t+1,i}, ..., \Delta P_{t+L-1,i}) \) and \( \Delta P_{t,i} = (x_{t+i} - x_{t,i}, y_{t+i} - y_{t,i}) \). We normalize the descriptor as:

\[
d = \frac{(\Delta P_{t}, \Delta P_{t+1}, ..., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} |\Delta P_{j}|},
\]

2.2 Dictionary construction

The traditional dictionary construction method is based on the K-means. The optimization process of K-means is as following: Let \( D \) be the set of the trajectory descriptors in \( H \)-dimensional feature space, i.e. \( D = \{d_{1}, d_{2}, ..., d_{M}\} \in \mathbb{R}^{H \times M} \). The K-means clustering algorithm aims to solve the following objective function:

\[
S(C) = \sum_{m=1}^{M} \min_{k} ||d_{m} - c_{k}||_{2},
\]

where \( C = (c_{1}, c_{2}, ..., c_{K}) \) represents the \( K \) cluster centers and \( || \cdot ||_{2} \) is the L2-norm of the vectors. For each input set of the descriptors \( D \), we want to get the reconstruction coefficient \( R = (r_{1}, r_{2}, ..., r_{M}) \), where \( M \) is the number of feature points. Eq. (3) can be re-formulated in the matrix form:

\[
\min_{c,R} \sum_{m=1}^{M} ||d_{m} - r_{m}C||_{2}^{2},
\]

\[subject to ||r_{m}||_{0} = 1, |r_{m}| = 1, r_{m} \geq 0, \forall m\]

where \( || \cdot ||_{0} \) is the L0-norm, meaning that there is only one nonzero element in \( r_{m} \). The condition of \( |r_{m}| = 1 \) sets the nonzero element to one. The nonzero element indicates the clusters of the current descriptor belonging to.

The constraint of the K-means method in Eq. (4) may be too restrictive, and may lead to limit description to \( d_{m} \) [9]. So sparse coding method [9] is introduced to replace the K-means to construct the dictionary. The sparse coding method releases the constraint of K-means, which has the following form:

\[
\min_{c,R} \sum_{m=1}^{M} ||d_{m} - r_{m}C||_{2}^{2} + \lambda |r_{m}|,
\]

\[subject to ||c_{k}|| \leq 1, \forall k = 1, 2, ..., K\]

where we apply the L2-norm to \( c_{k} \) to avoid the trivial solution [7]. Usually, we will get the overcompleted basis data set \( C \), i.e. \( H < K \). There are two phases in sparse coding, one is the training phase which uses the training data set \( D \) to construct the basis dictionary \( C \). The other is coding phase using the basis dictionary to solve Eq. (5) to get the coefficient \( r_{m} \).

After the basis dictionary and the corresponding coefficient are figured out, we utilize a novel spatial pooling approach which is similar to soft K-means [6]. The proposed method leaves each feature descriptor to have
more than one cluster centers. The difference is that we use the sparse coding method to construct the dictionary. To avoid reconstruction bias we set all the elements in \( r_m \) which are non-zero to one. Specially, one descriptor \( d_m \) now is represented by several visual words.

### 2.3 Spatial-temporal pyramid construction

We can compute a single feature vector based on some statistics of the descriptors to present the videos. For example, if \( R \) is obtained via Eq. (4), a general choice is to compute the histogram

\[
h = \frac{1}{M} \sum_{m=1}^{M} r_m, \tag{6}
\]

A histogram \( h \) for each video is represented by an unordered set of local descriptors. In the SPM approach, the image’s spatial pyramid histogram representation \( h \) is a concatenation of local histograms in various partitions of different scales. We construct the spatial and temporal pyramid based on SPM. In our method, we build a three-level pyramid. For each frame of the video, the image is divided into bins using SPM. For the temporal dimension, we uniformly cut the temporal dimension into three bins. We combine these bins together to construct our spatial-temporal pyramid.

### 2.4 Bag of features and classification

To evaluate the performance of our method, we use a non-linear classifier SVM to recognition actions. To limit the computation complexity, we use a subset of 100,000 randomly selected training features to construct the basis dictionary. To guarantee the precision of our dictionary, we use 100 iterations to find the lowest reconstruction error dictionary as our final dictionary.

For classification we use a non-linear SVM with a \( \chi^2 \) kernel [2].

\[
K(v_i, v_j) = \exp(D(v_i, v_j)), \tag{7}
\]

where \( D(v_i, v_j) \) is the \( \chi^2 \) distance between video \( v_i \) and \( v_j \). Since our action recognition is the multi-class classification, we use a one-against-the-rest approach and determine the class with the highest confidence score.

### 3. Experimental results

Firstly, a brief introduction of the action recognition dataset is given. Then the comparison between our proposed approach and other methods [2], [7] and [8] are shown. Finally, we evaluate the main parameters of our method and demonstrate how those parameters influence the results.

### 3.1 Datasets

The KTH dataset [5] contains six human action classes: walking, jogging, running, boxing, waving and clapping. Each action category is performed several times by 25 subjects in four different scenarios. In our experimental setup, we follow the original experimental setup of [5], e.g., 9 subjects (2, 3, 5, 6, 7, 8, 9, 10 and 22) are treated as the test set while the remaining 16 subjects are training set. We evaluate a multi-class classifier and report average accuracy over all classes as performance measure.

The Hollywood human actions dataset (HOHA) [2] consists of 8 classes of realistic human actions: answer phone, get out car, hand shake, hug person, kiss, sit down, sit up, stand up. The training set is sampled from 12 movies and the testing is sampled from 20 different movies.

### 3.2 Results

In this part, we first show the comparison with [7] on the KTH action dataset, which utilizes K-means to construct the dictionary. We set identical parameters to build descriptor as in [7], e.g., \( L = 15 \), \( W = 5 \). We set dictionary size as 600 visual words, and the parameter \( \lambda \) in Eq. (5) is 0.1. Table 1 compares the average class accuracy of our method with results reported by other researchers [7], [2] and [8]. Compared with the existing approaches, our method shows significantly better performance, outperforming the state-of-the-art [7] in the same setup.

<table>
<thead>
<tr>
<th>methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>our method</td>
<td>92.59%</td>
</tr>
<tr>
<td>Wang et al [7]</td>
<td>90.2%</td>
</tr>
<tr>
<td>STIP [2]</td>
<td>91.8%</td>
</tr>
<tr>
<td>Wong et al [8]</td>
<td>86.7%</td>
</tr>
</tbody>
</table>

Our method achieves about 2.5% improvement comparing with [7] on KTH dataset. Our experimental result is also better than [2] and [8] which used the spatiotemporal gradient information as their descriptors. The confusion matrix for our method are given in Figure 2. There are two advantages comparing with [7]. First, spatial-temporal pyramid is introduced which can save the spatial and temporal characteristic of the actions. Second, we use sparse coding method which can get a lower reconstruction errors. There are also some other technologies on action recognition in recent years which give us impressive results. But they are mostly
using leave-one-out setting, we only compare to those using the standard setting.

Hollywood action dataset is also used to test our method. The evaluation criterion is computing the average precision (AP) for each class and reporting the mean AP over all classes (mAP) as in [4]. We get 24.63% using trajectories as the feature descriptors comparing to [4] which is 21.1%. While in this experiments, the dictionary size is set to 1600, $\lambda$ is 0.1 and the linear kernel is used.

3.3 Parameter evaluation

In this subsection, we analyze the parameters of our method $\lambda$ in Eq. (5) and the size of dictionary size $K$. The regularization parameter $\lambda$ controls the weighting of each visual words in the representation. Figure 3 shows how each parameter influences the recognition results. Without surprise, the recognition accuracy is slightly growing when the number of the dictionary increases. The larger number of the dictionary size will make the representation much more discriminative. Here, $\lambda$ is given three values (0.1, 0.2, 0.3). We can find that a larger regularization $\lambda$ will cause a lower recognition accuracy since a larger regularization will reduce the number of the nonzero elements in coefficient, which will lose the discrimination.

4. Conclusion

In this paper, we proposed a novel approach (STPSC) to recognize actions in the video. STPSC extended the existing method [9] to action recognition in the video. Our method was evaluated on the public action dataset, and obtained better performance than the earlier peers. The main parameters of our method are also discussed to determine how they influence the recognition results. In the future work, we will dig the relationship of the visual words and align the videos to expand the application domain of our approach.

5. Acknowledgments

This work was supported by National Natural Science Foundation of China (No.60905019), Tianjin Key Technologies R&D program (No.11ZCKFGX00800), Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, SKL of CG&CAD, and Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030601).

References