Human Action Retrieval via Efficient Feature Matching

Jun Tang
Key Laboratory of IC & SP, Ministry of Education, Anhui University
Huangshan Road, Hefei, China
tangjunahu@gmail.com

Ling Shao, Xiantong Zhen
Department of Electronic and Electric Engineering, The University of Sheffield
Mappin Street, Sheffield, UK
{ling.shao, elr10xz@}sheffield.ac.uk

Abstract

As a large proportion of the available video media concerns humans, human action retrieval is posed as a new topic in the domain of content-based video retrieval. For retrieving complex human actions, measuring the similarity between two videos represented by local features is a critical issue. In this paper, a fast and explicit feature correspondence approach is presented to compute the match cost serving as the similarity metric. Then the proposed similarity metric is embedded into the framework of manifold ranking for action retrieval. In contrast to the Bag-of-Words model and its variants, our method yields an encouraging improvement of accuracy on the KTH and the UCF YouTube datasets with reasonably efficient computation.

1. Introduction

With the ever-increasing growth in the popularity of digital videos, how to search and manage these videos efficiently has become the main challenge, and effective retrieval techniques are in urgent demand. Traditional video retrieval systems are mostly based on keywords, such as those used in YouTube and Google Videos. In these systems, the user query (in the form of keywords) is compared with the textual context associated with a video including the title, manual annotation, tags, web documents, etc. However, the text-based systems suffer easily from many problems such as shortage and ambiguity of textual information and semantic inconsistency between textual and visual data. Therefore, content-based video retrieval, which is based on the automatically extracted visual features, has become a sensible trend to overcome the above difficulties.

As human actions dominate the majority of available videos, recently there has been a boost of interest in investigating the problem of human action retrieval [1, 2]. In contrast to the closed topic of action recognition, action retrieval is a much more difficult task. For action retrieval, the only prior knowledge is usually a single query sample. Under such a circumstance, the amount of training data is extremely limited and only available in the procedure of query, whereas in action recognition, a large amount of positive and negative samples can be drawn upon. In the work of [1, 2], Jones and Shao concentrated on the application of relevance feedback in action retrieval and the similarity metric is mainly based on the traditional Bag-of-Words (BoW) model. It should be noted that the similarity metric, one of the most important elements for a retrieval system, has significant influence on the system performance. Although the BoW model and its extensions, combined with local primitive features, have achieved great success in image/video retrieval, their essential shortage leads to many limitations such as inability to capture enough spatial and temporal structural information, quantization errors and information loss during the generation of the codebook. Addressing these weaknesses, some researchers turn to constructing comparison kernels to evaluate the similarity through feature correspondences, and many promising results have been reported on canonical image datasets [13-17]. Especially in [17], Duchenne et al. demonstrated the superiority of explicit correspondences, which yields competitive performance with the state-of-the-art techniques. However, to the best of our knowledge, little effort has been made to apply the idea of explicit correspondences to video data. We believe the reasons are two-fold: First, the number of local features must be large enough to characterize a video sequence so that it hinders the usage of feature matching algorithms in high time complexity, since the correspondences between a

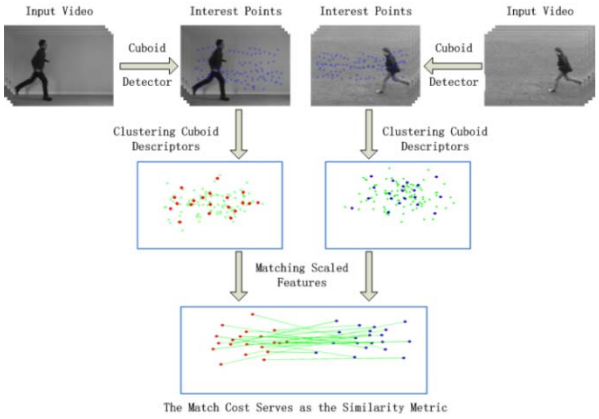

Figure 1: A simplified diagram of the proposed similarity metric between two videos.

query video and all the database videos need to be computed; Second, for image data, many spatial or structural constraints are preserved well, which can be utilized to explore fast approximate solutions to feature matching [16, 17], but they make little sense for video data due to the large intra-class diversity in nature. Based on these observations, our main motivation for the proposed approach is to establish explicit correspondences based similarity metric for video data, which is expected to find a balance between the accuracy benefiting from explicit correspondences and computational efficiency.

It is particularly important to mention that many real-world video datasets have underlying clusters or manifold structures. In such cases, it means that video samples often locate in a much lower dimensional intrinsic space and the Euclidean assumption is only valid locally. Therefore, a well-designed action retrieval scheme should take the intrinsic structure of the video sets into account. Manifold ranking [4] and its extensions have been widely used in information retrieval and demonstrated their good performance and flexibility on various data types, such as text, image and shape. In terms of the underlying structure, manifold ranking assigns each data point a relative ranking score, which is more reasonable to capture the extent of semantic relevance compared to the straightforward pairwise similarity. In this paper, we embed the proposed pairwise similarity into the framework of manifold ranking reported by Bai et al. [5] and achieve a significant improvement on retrieval accuracy.

The rest of the paper is organized as follows. In Section 2, we briefly introduce some related work. In Section 3, we describe the proposed similarity metric in detail and discuss the combination with manifold ranking as well as the approach to adding relevance feedback to the action retrieval system. The experimental results on two canonical action datasets are presented in Section 4. Finally, some conclusions are drawn and the future work is outlined.

2. Related Work

The related work in the literature can be generally divided into three areas.

The first of these concerns the local representation of video sequences. Laptev and Lindeberg [6] introduced space-time interest points (STIPs) by extending the Harris corner detector. Other STIP detectors include detectors based on Gabor filters [7, 8] and on the determinant of the spatio-temporal Hessian matrix [9]. A variety of local descriptors extended from their image versions, such as 3D-SIFT [10], HOG3D [11] and extended SURF, are then applied to describe the cuboids extracted around the detected STIPs.

The second related topic is on the approaches to measuring similarity between images or videos represented by local feature sets. Sivic and Zisserman [3] proposed the pioneer work of the BoW model where the features are quantized to form a bag of words and thus each image is accumulated as a histogram of visual words. Following Sivic and Zisserman [3], Gemert et al. [12] modeled visual words assignment ambiguity by using the soft assignment technique. Additionally, many matching-based approaches were proposed in distinct forms, such as the Mercer match kernel [13], the part-based spatial models [14], and the pyramid match kernel [15]. Fairly speaking, these matching-based methods are only the variants of the BoW model because explicit correspondences are not established in them. The primary reason for this may lie in the fact that the expensive computational cost of explicit matching stands as an obstacle. In [16, 17], the authors used explicit feature matching to construct the image comparison kernel, and what makes explicit feature matching feasible is that spatial consistency represented by the neighborhood relationship is employed to speed up the computation.

The third area is that of manifold ranking. Zhou et al. [18] used the data manifold structure to rank retrieved objects and significantly improved the performance. Liu et al. [19] applied random walk to the tag similarity graph to achieve automatic tag ranking. Bai et al. [5] utilized contextual information for shape retrieval with graph transduction, which is a frequently used approach to manifold ranking. More recently, the tensor product graph was introduced to model data [20].

In a nutshell, little effort has been made to compare videos through explicit correspondences owing to the expensive computational cost, although the expected accuracy is alluring. Our key contribution in this paper is to develop a novel strategy to facilitate the fast computation of establishing explicit correspondences between video pairs, which is basically different from previous works. Moreover, the proposed similarity metric is seamlessly integrated with manifold ranking to learn an intrinsic distance for action retrieval.

3. Feature Matching Based Action Retrieval

3.1. Video Representation

We model our action retrieval scheme on the basis of local spatio-temporal features. And the method of Dollár [8] is adopted to extract STIPs. Compared with the 3D Harris Corner detector [6], it can generate dense features which have been validated to improve the recognition accuracy in many action recognition tasks. As for the descriptor, since the performance of various descriptors is not the focus of this work, we choose the most popular HOG3D [11].

Firstly, separable linear filters are applied to the video sequence to get a response function for each point. The response function is denoted by:

$$\boldsymbol{R} = (\boldsymbol{I} * g * h_{ev})^2 + (\boldsymbol{I} * g * h_{od})^2 \qquad (1)$$

where $\boldsymbol{I}(x, y, t)$ indicates a video sequence, $g(x, y; \sigma)$ is a 2D Gaussian smoothing kernel applied to the spatial dimensions, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, and defined as:

$$h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)\mathrm{e}^{-t^2/\tau^2} \quad (2)$$

$$h_{ev}(t; \tau, \omega) = -\sin(2\pi t\omega)\mathrm{e}^{-t^2/\tau^2} \quad (3)$$

where parameters τ and ω correspond to the spatial and temporal scales, respectively. We follow the setting of in $\omega = 4/\tau$ in [8].

To describe a local STIP, a spatio-temporal cuboid is extracted around it. Then the gradients along the x, y and t axes are calculated. Based on these gradients, a HOG3D descriptor is obtained through orientation quantization and histogram computation. Details of HOG3D can be found in [11].

3.2. Proposed Similarity Metric

A diagram describing the proposed similarity metric is shown in Fig. 1. Given a video set $\boldsymbol{V} = \{v_1, v_2, \cdots, v_n\}$, without loss of generality, we assume v_1 is a query video and the remaining ones are the database videos. For any $v_i \in \boldsymbol{V}$, we extract m local features and the corresponding feature set is denoted by $\boldsymbol{F}_i = \{\boldsymbol{f}_t^i | t = 1,2,\cdots,m\}$. And the feature dimensionality is denoted by l.

As we mentioned earlier, the main obstacle for applying explicit matching is the large size of local feature sets representing videos. Since it is not easy to reduce the time complexity of feature matching algorithms, we turn to employing the idea of ***scaling data*** to achieve efficient computation. Somewhat similar ideas can be found in [21], where a video sequence is divided into many subvolumes and the features representing each subvolume are put together to characterize the video. Consequently, the amount of local features is reduced. However, the features of subvolumes are still inherited from the BoW model so that the problems brought from it are not essentially solved. Here we adopt an entirely different approach. In the BoW model, the histogram form implies that features assigned to a same histogram bin are similar, and hence we conjecture that a small number of dominant features derived from the original feature set may be sufficient enough to represent the whole set. With such an assumption, we apply the k-means algorithm to each feature set to group the features into a few clusters, and the cluster centers are treated as the ***scaled features***. That is, for each $\boldsymbol{F}_i$ associated with $v_i \in \boldsymbol{V}$, the scaled feature set is denoted by $\boldsymbol{H}_i = \{h_s^i | s = 1,2,\cdots,q\}$, where q is the number of clusters. And $\boldsymbol{h}_s^i$ is defined as:

$$\boldsymbol{h}_s^i = \sum_{k \in \boldsymbol{\Omega}_s} \boldsymbol{f}_k^i / |\boldsymbol{\Omega}_s| \quad (4)$$

where $\boldsymbol{\Omega}_s$ is the sth cluster in $\boldsymbol{F}_i$.

With the scaled feature sets available, the set of correspondences between $\boldsymbol{H}_i$ and $\boldsymbol{H}_j$ is defined as $\phi: \boldsymbol{H}_i \longrightarrow \boldsymbol{H}_j$. The optimal correspondences are obtained by minimizing the total cost of matching:

$$\phi^* = \mathrm{argmin}_\phi(\sum_s ||\boldsymbol{h}_s^i - \boldsymbol{h}_{\phi(s)}^j||) \quad (5)$$

subject to the one-to-one correspondence constraint. Obviously, it is a typical instance of the weighted bipartite graph matching problem, which can be solved in cubic polynomial time by using the Hungarian algorithm. And the corresponding match cost is defined as:

$$C(v_i, v_j) = C(\boldsymbol{H}_i, \boldsymbol{H}_j) = \frac{1}{q}\sum_s ||\boldsymbol{h}_s^i - \boldsymbol{h}_{\phi^*(s)}^j|| \quad (6)$$

Thereupon the similarity $sim(v_i, v_j)$ between v_i and v_j can be given by using a Gaussian function:

$$sim(v_i, v_j) = exp(-C^2(v_i, v_j)/2\beta^2) \quad (7)$$

where β is a smoothing coefficient. In our experiments, the fast iterative k-means algorithm of Cai [22] is adopted to speed up the computation.

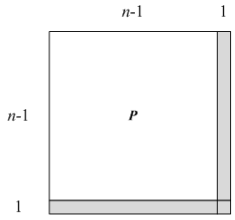

Figure 2: Extend matrix $\boldsymbol{P}$ in the grey regions for a query ($\boldsymbol{P}$ is defined in Section 3.3).

Data for Matching	Scaling Features	Constructing Distance Matrix	Matching Features
Original Features	/	$O(nm^2l)$	$O(nm^3)$
Scaled Features	$O(pqml)$	$O(nq^2l)$	$O(nq^3)$

Table 1 Time performance of different methods.

Now we turn our attention to the time performance of the proposed similarity metric. Since the similarity between database videos can be computed offline, we mainly investigate the time consumption on computing similarity between a query video and database videos (see Fig. 2). Suppose the iterative number of the k-means algorithm is p, thus we need $O(pqml)$ to scale feature sets. Besides, we need $O((n-1)q^2l)$, i.e., $O(nq^2l)$, to construct the distance matrix when using the Hungarian algorithm. Similarly, $O(nq^3)$ time is needed to solve the correspondences. We summarize the above analysis and compare it with the matching on original features in Table 1. It is worth noting that p and q are often chosen to be much smaller than m in practice. Obviously, the time consumption is reduced to a great extent in the case of using scaled features.

3.3. Manifold Ranking

With the pairwise similarity at hand, we discuss the details of using Bai's manifold ranking method for action

retrieval. We define $w_{ij} \equiv w(v_i, v_j) = sim(v_i, v_j)$, for $i, j = 1,2,\cdots,n$, and the elements of the probabilistic matrix $\boldsymbol{P}$ with the dimension $n \times n$ is given by:

$$\boldsymbol{P}_{ij} = w_{ij} / \sum_{k=1}^{n} w_{ik} \quad (8)$$

Our aim is to seek the intrinsic distance measure of $\{v_2, \cdots, v_n\}$ with respect to v_1. We simplify it as $g(v_i)$ for $i, j = 1,2,\cdots,n$, and the self-similarity (i.e., $g(v_1)$) is set to 1. The ranking process is as follows:

Step 1: Initially, assign $g_1(v_1) = 1$ and $g_1(v_i) = 1$ for $i = 2,\cdots,n$.

Step 2: Set $g_{t+1}(v_i) = \sum_{j=1}^{n} \boldsymbol{P}_{ij} g_t(v_j)$ for $i = 1,\cdots,n$.

Step 3: Set $g_{t+1}(v_i) = 1$.

Step 4: Repeat steps 2 and 3 until the maximum iteration number is reached.

After such an iterative procedure, we can determine the similarity between each dataset video and a given query and return the user the most similar videos in terms of the learned distance.

3.4. Relevance Feedback

Relevance feedback provides a powerful interactive facility to improve the performance of information retrieval systems. Incorporating the user feedback on relevant and/or irrelevant information of the retrieved results, a new and often more accurate query can be captured. Relevance feedback can be achieved by some complicated techniques such as SVM and manifold learning. As we concentrate on the similarity measurement between videos here, a simple strategy is employed for relevance feedback: in step 3 of the ranking process, we set $g_{t+1}(v_i) = 1$, if v_i is relevant to the query; otherwise, we set $g_{t+1}(v_i) = 0$.

4. Experiments

4.1. Datasets and Experimental Setup

We conduct our experiments on the KTH dataset and the unconstrained UCF YouTube dataset. The KTH dataset is a commonly used canonical action dataset comprising 598 samples, which contains six action classes performed by 25 subjects. Each action is performed in four different scenarios: outdoors, outdoors with scale variation, out-doors with different clothes and indoors. The UCF YouTube dataset is composed of 1160 videos in 11 action categories drawn from existing YouTube videos. Action retrieval from the UCF YouTube dataset is more challenging because the scenario is more realistic. In the experiments on the KTH dataset, the number of STIPs is set to 100. Due to the complexity of the UCF YouTube dataset, we set the number of STIPs to 400 in order to capture most of the variations.

For the purpose of conducting a comprehensive evaluation on the datasets, a set of round-robin tests are performed: each video clip is treated as a query whilst the remainder of the dataset is regarded as the queried database. As for the metric of performance evaluation, we follow the configuration in [2]: The precision of a query is computed as the percentage of the top $M (M = ceil\left(\frac{N}{5}\right))$ results belonging to the query's action category (N is the number of all the database videos holding the same category as the query). The BoW model [3], the BoW-Soft model [12] and the match kernel [13] are chosen as the baseline methods to verify the effectiveness of the proposed similarity metric. Meanwhile, ranking in terms of the original distance is also performed to provide a reference for evaluating the effectiveness of manifold ranking. All the methods are implemented with Matlab, and the following running time is obtained by computing with a single 2.8 GHz Xeon CPU core.

4.2. Results

We first investigate the performance on the KTH dataset. In Table 2, we compare the overall average precisions obtained from matching scaled features, BoW, BoW-Soft and match kernel, where the number associated with each method denotes the number of scaled features or the size of codebook, respectively. The performance of BoW and BoW-Soft is tightly related to the size of codebook, and we only report the best tuned results for brevity. It is easy to see that the proposed explicit matching scheme comprehensively outperforms the alternatives in this dataset, no matter in the case of original distance or learned distance. The experimental results validate that the proposed computational model is essentially reasonable and effective. Meanwhile, the retrieval precisions are improved to a great extent by leveraging manifold ranking. There is a more than 10% margin of improvement for all the compared methods. Fig. 3 shows the average precisions with respect to the original distance and the learned distance in the case of varied numbers of scaled features, where the number of scaled features is chosen as 30, 40 or 50, respectively. Interestingly, the performance is very close for different numbers of scaled features, which indicates the proposed similarity metric possesses the advantage of parameter insensitivity. In addition, we also show the per-class comparison on precisions when the number of scaled features is 40 in Fig. 4. The precisions of learned distance exceed those of original distance in most classes.

Metric	Original Distance (%)	Learned Distance (%)
Matching Scaled Features (40)	**60.2**	**76.3**
BoW (1000)	57.0	70.1
BoW-Soft (3000)	46.9	63.2
Match Kernel	47.3	64.8

Table 2 Retrieval precisions of the compared methods on the KTH dataset.

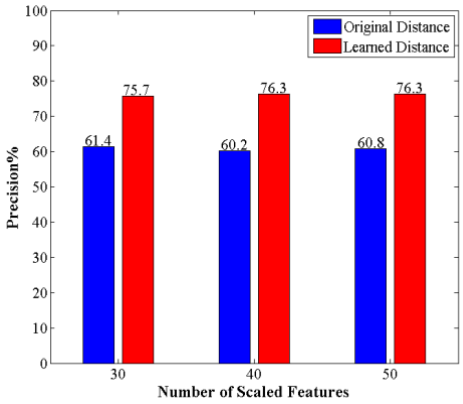

Figure 3: Retrieval precision versus number of scaled features on the KTH dataset.

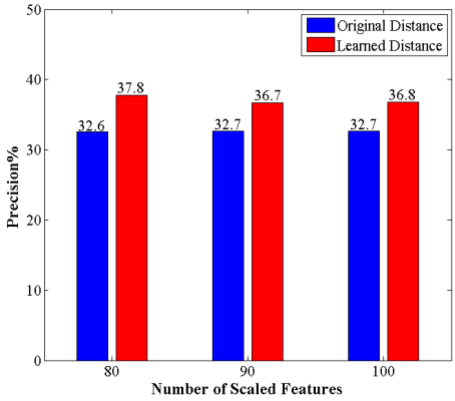

Figure 5: Retrieval precision versus number of scaled features on the UCF YouTube dataset.

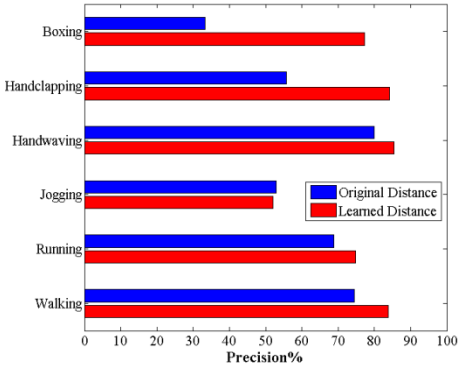

Figure 4: Per-class precisions on the KTH dataset when the number of scaled features is 40.

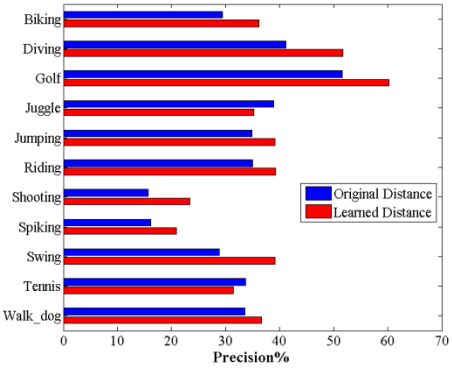

Figure 6: Per-class precisions on the UCF YouTube dataset when the number of scaled features is 80.

Data for Matching	Running Time (s)
Scaled Features (30)	**0.58**
Scaled Features (40)	1.32
Scaled Features (50)	3.04
Original Features	9.31

Table 3 Summary of running time on the KTH dataset.

In Table 3, we summarize the running time for computing the similarity between a query and database videos by matching original features and scaled features on the KTH dataset, where the number denotes the number of scaled features. The considerable difference in magnitude indicates that the proposed method can greatly reduce computational time. Putting the above precision data together, we can conclude that matching with scaled features find a balance between precision and computational efficiency.

To take our study one step further, we conduct experiments on the realistic UCF YouTube dataset. Table 4 summarizes the overall average precisions of all the mentioned methods. Since the scenario in the YouTube dataset is more challenging, the performance degrades to some extent. However, the proposed approach still consistently yields better results than the compared methods. In Fig. 5, we show the precisions corresponding to different numbers of scaled features. Although the improvement derived from manifold ranking drops to 5% or so, the advantage of parameter insensitivity is verified once again. The per-class comparison on precisions is plotted in Fig. 6, when the number of scaled features is 80. We can observe that the precisions for 9 out of the 11 classes benefit from manifold ranking. The summary of running time on the UCF YouTube dataset is detailed in Table 5, from which we can see the usage of scaled features greatly improves computational efficiency.

Metric	Original Distance (%)	Learned Distance (%)
Matching Scaled Features (80)	**32.6**	**37.8**
BoW (1000)	25.9	31.7
BoW-Soft (2000)	24.4	29.7
Match Kernel	23.8	29.2

Table 4 Ranking accuracies of the compared methods on the UCF YouTube dataset.

Finally, we turn to the usage of relevance feedback. We simulate relevance feedback by using ground truth data from the dataset: for each query, the labels of the top M retrieved results are employed as relevance feedback. We assume the users do not have much patience and the process is performed for two rounds. And $M/2$ videos are labeled in each round. The cases of 40 scaled features for the KTH dataset and 80 scaled features for the UCF YouTube dataset are chosen to verify the effectiveness of relevance feedback. Fig. 7 shows the variation of the average precision through relevance feedback. Although the adopted relevance feedback approach is extremely simple, we still receive a remarkable improvement.

Data for Matching	Running Time (s)
Scaled Features (80)	**8.68**
Scaled Features (90)	9.67
Scaled Features (100)	10.49
Original Features	1968.43

Table 5 Summary of running time on the UCF YouTube dataset.

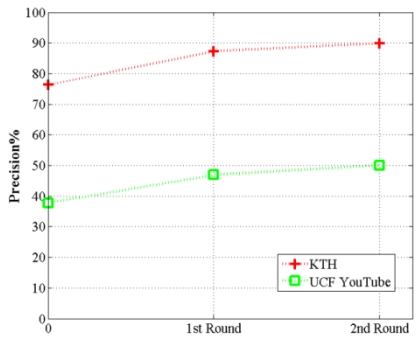

Figure 7: Retrieval precision after relevance feedback.

5. Conclusion and Future Work

In this paper, we have proposed an efficient and explicit matching method to compute the similarity between video pairs. We apply this method to a human action retrieval application and test on two benchmark action datasets. Compared to the BoW model and its variants, the proposed method produces a considerable margin of improvement with efficient computation. In our ongoing research, we will investigate combining our method with multiple features, expecting to achieve better results on realistic datasets.

Acknowledgements

Jun Tang would like to acknowledge the support of China Scholarship Council and Natural Science Foundation of Anhui Provincial Education Department (Grant No. 2011KJ A008).

References

[1] S. Jones, L. Shao. Content-based retrieval of human actions from realistic video databases. Information Sciences, 236:56-65, 2013.
[2] S. Jones, L. Shao, J. Zhang, Y. Liu. Relevance feedback for real-world human action retrieval. Patten Recognition Letters, 33(4):446-452, 2012.
[3] J. Sivic, A. Zisserman. Video google: A text retrieval approach to object matching in videos. In ICCV, 2003.
[4] J. He, M. Li, M. Zhang, et al. Manifold ranking based image retrieval. In ACM Multimedia, 2004.
[5] X. Bai, X. Yang, L.J. Lateki, et al. Learning context sensitive shape similarity by graph transduction. TPAMI, 32(5):861-874, 2010.
[6] I. Laptev, T. Lindeberg. Space-time interest points. In ICCV, 2003.
[7] M. Bregonzio, S. Gong, T. Xiang. Recognizing action as clouds of space-time interest points. In CVPR, 2009.
[8] P. Dollár, V. Rabaud, G. Cottrell, et al. Behavior recognition via sparse spatio-temporal features. In VSPETS, 2005.
[9] G. Willems, T. Tuytelaars, L.V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In ECCV, 2008.
[10] P. Scovanner, S. Ali, M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In ACM Multimedia, 2007.
[11] A. Kläser, M. Marszałek, C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In BMVC, 2008.
[12] J.V. Gemert, C. Veenman, A. Smeulders, et al. Visual word ambiguity. TPAMI, 32(7):1271-1283, 2010.
[13] S. Lyu. Mercer kernels for object recognition with local features. In CVPR, 2005.
[14] R. Fergus, P. Perona, A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. IJCV, 71(3):273-303, 2006.
[15] K. Grauman, T. Darrell. Pyramid match hashing: sub-linear time indexing over partial correspondences. In CVPR, 2007.
[16] M. Parsana, S. Bhattacharya, C. Bhattacharyya, K.R. Ramakrishnan. Kernels on attributed point sets with Applications. In NIPS 2007.
[17] O. Duchenne, A. Joulin, J. Ponce. A graph-matching kernel for object categorization. In ICCV, 2011.
[18] D. Zhou, J. Weston, A. Gretton, et al. Ranking on data manifolds. In NIPS, 2004.
[19] D. Liu, X. Hua, L. Yang, et al. Tag ranking. In WWW, 2009.
[20] X. Yang, L. Prasad, L.J. Lateki. Affinity learning with diffusion on tensor product graph. TPAMI, 35(1):28-38, 2013.
[21] M. Sapienza, F. Cuzzolin, P. H.S. Torr. Learning discriminative space-time actions from weakly labeled videos. In BMVC, 2012.
[22] D. Cai. Litekmeans: the fastest matlab implementation of kmeans. Software available at: http://www.zjucadcg.cn/dengcai/Data/Clustering.html,2011.