Multimed Tools Appl (2019) 78:1737–1755
https://doi.org/10.1007/s11042-018-6305-x

Interactive video summarization with human intentions

Huaping Liu[1] · **Fuchun Sun**[1] · **Xinyu Zhang**[2] ·
Bin Fang[1]

Received: 4 March 2018 / Revised: 7 June 2018 / Accepted: 22 June 2018 /
Published online: 30 June 2018

Abstract Automatic video summarization, which is a typical cognitive-inspired task and attempts to select a small set of the most representative images or video clips for a specific video sequence, is therefore vital for enabling many tasks. In this work, we develop an interactive Non-negative Matrix Factorization (NMF) method for representative action video discovery. The original video is first evenly segmented into short clips, and the bag-of-words model is used to describe each clip. A temporally consistent NMF model is subsequently used for clustering and action segmentation. Because the clustering and segmentation results may not satisfy user intention, the user-controlled operations MERGE and ADD are developed to permit the user to adjust the results in line with expectations. The newly developed interactive NMF method can therefore generate personalized results.Experimental results on the public Weizman dataset demonstrate that our approach provides satisfactory action discovery and segmentation results.

Keywords Interactive action summarization · Video summarization · Human-machine interaction · Non-negative matrix factorization

1 Introduction

With the popularization of digital cameras and smart phones, far more video clips are being captured and stored than ever before [13, 21]. These large collections of video clips are intrinsically difficult to browse owing to the great number of clips and the inability of computers to effectively characterize their content. Automatic video summarization, which is a

✉ Huaping Liu
hpliu@tsinghua.edu.cn

[1] Department of Computer Science and Technology, Tsinghua University, BNRist, State Key Lab. of Intelligent Technology and Systems, Beijing, China

[2] State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing, China

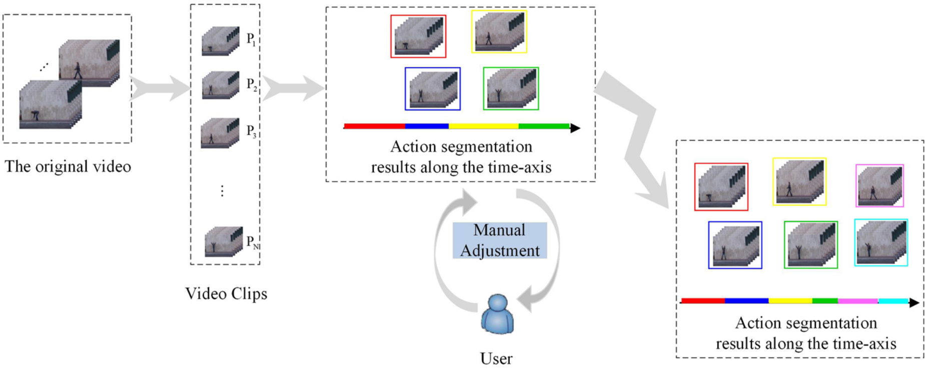
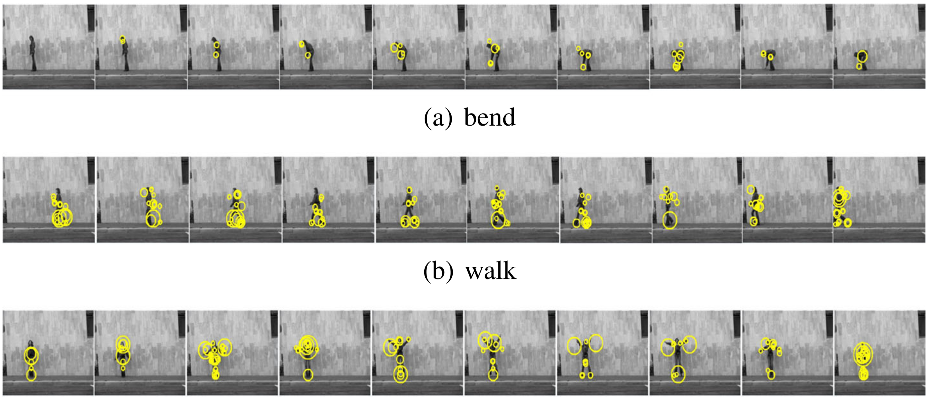

The original video
P1
P2
P3
PN
Video Clips
Action segmentation results along the time-axis
Manual Adjustment
User
Action segmentation results along the time-axis

Fig. 1 Overview of the proposed method

typical cognitive-inspired task [24] and attempts to select a small set of the most representative images or video clips for a specific video sequence, is therefore vital for enabling video retrieval, video browsing, video organization, video surveillance, action recognition [9], and so on. However, the majority of present automatic video summarization work has focused on extracting some key-frames while neglecting the intrinsic temporal consistencies in the video [18]. In many videos, actions form the basic elements and provide important information about the content [16, 27]. Therefore, the action mining of video is a very important component of the problem [1, 14]. In [26] the authors proposed a principled method for online generation of a short video summarizing the most important and interesting content of a long video. Because the method functions online, the number of segments can be automatically determined. However, the method only extracts the first segments obtained after the content changes, and incorporates these into the summarization. As such, the method discovers new actions, but not representative actions, making the method suitable for rapid browsing, but not for analysis, which is addressed in the present work.

Numerous studies of action discovery and segmentation problems have been conducted [7]. In [3], an extended hidden Markov model was proposed for joint action segmentation and classification. However, the model requires estimation using an annotated training set of action sequences. In [11], a matrix factorization method was developed to simultaneously clusters pixel prototypes into signatures and video sequences into action classes. However, this work focused on clustering for different video sequences, and the number of clusters required definition prior to conducting the clustering operation. In [15], a Bayesian non-parametric model of sequential data was adopted to allow for completely unsupervised activity discovery. The authors claim that this work required no predefinition of the relevant behaviors or even the number of behaviors, which were learned directly from data. However, the presented method exhibits the following disadvantages: (1) Due to the complexity of the non-parametric Bayesian method, its time burden is rather substantial. (2) The number of behaviors, though need not be determined by the user, is still sensitive to some parameters of the algorithm (especially the Dirichlet prior parameter). As such, the task of determining the number of behaviors does not diminish, but is replaced by another task to determine a more uninterpreted parameter. (3) The inference algorithm may introduce randomness, which leads to inconsistent results from multiple runs when the human factor is incorporated into the loop. Such a problem has also been discussed elsewhere [10].

Because the process of clustering is highly-related to human intention [20], human intention must be effectively incorporated into the algorithm with the output of the corresponding



(a) bend

(b) walk

(c) jack

Fig. 2 Examples of the spatio-temporal interest points in the Weizman dataset [28]. The radius of circles is proportional to the scale at which change is detected

expected results [12]. In [8], the authors incorporated user-provided constraints into a document clustering problem. The user was therefore able to provide supervised information for document clustering in terms of pair-wise constraints that specify whether some documents either must or cannot be clustered together.

Recently, the incorporation of human intention was addressed for image clustering by introducing some human operations [25], and an interactive non-negative matrix factorization method was employed for document topic discovery [5, 6, 10, 27]. Non-negative Matrix Factorization (NMF), which aims to factorize a matrix into two non-negative matrices whose product reconstructs the original data matrix, has been shown extensive applications in many domains such as signal processing and machine learning [4], and so on. It has been found that such a factorization exhibits many favorable properties such as sparsity and interpretability [19]. In addition, the obtained part-based representation is consistent to the psychological and physiological evidence in human brain [17].

To the authors' best knowledge, no related work has been conducted for video action discovery incorporating human operations, which serves as the primary motivation for the present study. The main task of this work is the discovery of action categories within a video sequence, and, thereby, to identify the actions in the video sequence. The main contributions of the present work are summarized as follows.

1. A new interactive Non-negative Matrix Factorization (NMF) method is designed for representative video action discovery.
2. Two interactive operations designated as MERGE and ADD are developed to incorporate user intention and enhance the video action summarization performance.
3. A practical software system is developed, and extensive experiments are performed to show the effectiveness of the proposed method.

The remainder of this paper is organized as follows. Section 2 provides an overview of the proposed method, and Section 3 discusses the video representation employed. In Section 4, we describe the proposed method in detail, and Section 5 presents the experimental results. Finally, Section 6 concludes the work and offers suggestions for future study.

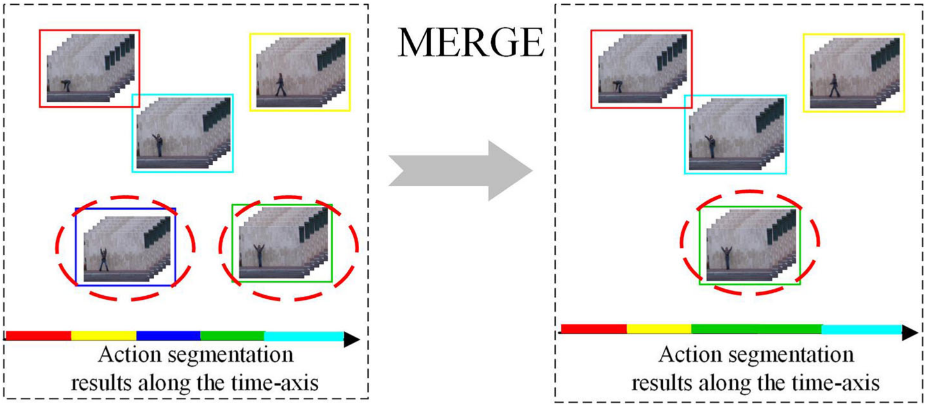

MERGE
Action segmentation results along the time-axis
Action segmentation results along the time-axis

Fig. 3 An illustration of the MERGE operation. The action clips which are surrounded by the dashed red ellipses in the left panel are merged into one single action clip, which is surrounded by the dashed red ellipses in the right panel. After this operation, the value of r and the action segmentation results are updated

2 Overview of the interactive action summarization systems

Figure 1 provides an overview of the developed method. The original video is firstly divided into many short video clips, and a machine learning method is utilized to extract those that are representative of the action. The user can evaluate the results and modify the operation. The user intention is incorporated into the optimization model, and biases the algorithm in outputting the expected results. Finally, a set of satisfactory representative action video, and the corresponding action segmentation results along the time-line, are obtained. Please note that the colored lines in the time axis correspond to the action video clip of the same color.

3 Video representation

The first task for video analysis is to transform the video into some suitably structured form. In this work, we follow the popular Bag-of-Words (BoW) framework, which has been successfully utilized in a number of action analysis studies [22]. To this end, we use Spatio-Temporal Interest Points (STIPs) to detect interest points and obtain Histogram-Of-Gradients (HoG) and Histogram-of-optical Flow (HoF) descriptors. Figure 2 provides several frames from the video data used in the present study and the detected STIPs within these frames. The obtained default descriptors are of $d = 162$ dimensions.

We evenly divide the original video into segments comprised of T frames, where the parameter T is specified by the user. The chosen value should ensure the consistency of action within each segment. In this work, we select $T = 24$ frames, representative of about one second of video. These segments, which are denoted as $\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_N$, represent the basic units of actions, from which the final action summary is constructed. The final action summary should include such segments. N is the number of examples in the video, and is obtained as the ceiling of the total number of frames divided by T.

To give a formal representation of the segments, we first cluster all of the descriptors of this video into K clusters. The parameter K is also a meta-parameter that is specified by the user. A larger K provides better accuracy, but also increases the summarization period. In this work, we empirically selected $K = 128$. The obtained K cluster centers are regarded as code-words. Then, each descriptor is mapped to the nearest code-word, and each segment can be represented as a K-dimensional BoW histogram [22]. We can therefore represent the entire video as $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, where $\mathbf{y}_i$ is the K-dimensional BoW histogram for the i-th segment.

Please note that the above procedure is time consuming (especially for STIP extraction). According to our implementation, the time required for this period is about 3 times that of the normal play time of a video. This is a substantial barrier for interactive video mining. Fortunately, this task can be completed offline. Therefore, in our framework, we store the corresponding feature data of a video for further processing in the form of a matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in R^{K \times N}$.

4 Non-negative matrix factorization for video action discovery

4.1 Basic non-negative matrix factorization

Given the matrix $\mathbf{Y} \in R^{K \times N}$, which includes the low-level action information of the original video, we face the problem of extracting the representative action clips from $\mathbf{Y}$, and then projecting each column to the corresponding representative action clip, which provides the action segmentation results. One method of addressing this problem involves the popular NMF method [17], which solves the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{U}\mathbf{V}||_F^2 \qquad s.t.\ \mathbf{U} \geq 0,\ \mathbf{V} \geq 0, \tag{1}$$

where $\mathbf{U} \in R^{K \times r}$ and $\mathbf{V} \in R^{r \times N}$ are two non-negative matrices. The term-topic matrix $\mathbf{U}$ uncovers the latent topic structure of the actions and r is usually set by the user with a value smaller than K and N.

Once the solutions of $\mathbf{U}$ and $\mathbf{V}$ are obtained, we can subsequently infer the topic presentations of segments, namely the topic-segment matrix $\mathbf{V}$, by projecting all of the segments into the latent topic space. Such a model was originally proposed in [10], and has been used in many fields such as document clustering and image clustering. However, because we deal with continuous video in our work, temporal consistency should be maintained to reflect the continuity of action. Therefore, the model is modified as

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{U}\mathbf{V}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 \\ &s.t.\ \mathbf{U} \geq 0,\ \mathbf{V} \geq 0, \end{aligned} \tag{2}$$

where β is a parameter employed to enforce temporal consistency, and $\mathbf{V}_i$ represents the i-th column of $\mathbf{V}$. In this work, the parameter β is empirically set to 0.1.

After obtaining the solutions $\mathbf{U}$ and $\mathbf{V}$, we can easily obtain the discovered representative actions and the temporal action segmentation results. The details are described as follows. For $\mathbf{U}$, each column $\mathbf{U}_i \in R^K$ corresponds a representative action clip. By searching

$$i^* = \operatorname*{argmin}_{j \in [1,N]} \frac{\mathbf{U}_i^T \mathbf{y}_j}{||\mathbf{U}_i||_2 \cdot ||\mathbf{y}_j||_2},$$

we can use the video clip $\mathbf{P}_{i^*}$ as the corresponding representative action clip. On the other hand, we use the column $\mathbf{V}_j \in R^r$ for $j = 1, 2, \cdots, N$ to determine the cluster assignment of the j-th video clip, and thereby realize the action segmentation. Concretely speaking, we search the maximum element in the vector $\mathbf{V}_j$, and use the corresponding index as the clustering assignment result.

Although the above procedure has achieved great success in a number of fields [23], it admits some disadvantages. First, the choice of r by the user is very challenging and even impossible in practice. In addition, it is well known that the formulation of the clustering problem is highly dependent on human intention. Considering a sample of students for example, a human may cluster them according to age, sex, or weight, and a different intention leads to different clustering results. It is therefore necessary to develop an interactive method for incorporating user intention into the clustering algorithm.

4.2 Interactive non-negative matrix factorization

In [10], some interesting operations such as keyword operations were used for interactive topic discovery or refinement. However, such operations cannot be employed for video. The main reason for this is that, in document clustering for example, the dictionary atom is a conventional word (such as *dog*, *apple*, *play*, *eat*, and so on.) which has a semantic meaning, and it is impossible to construct such a dictionary for a video. In our case, the dictionary is learned using a k-means clustering algorithm, and, therefore, the dictionary atoms have no semantic meanings. As such, the keyword-based operation defined in [10] is not applicable. On the other hand, for document clustering, we can use the keyword distribution of each topic to realize the visualization. For the same reason cited above, this manner of visualization is not suitable in our case. To this end, we developed two interaction operations: MERGE and ADD, for user-directed visualized video action discovery. These operations are described below. Please note the some more complicated operations such as DELETE, DIVIDE, and so on, can be flexibly incorporated into this unified framework.

4.2.1 MERGE operation

The MERGE operation attempts to address the problem where similar video segments may be clustered into different topics. This is unavoidable due to at least two reasons. (1) The semantic gap between human understanding and that of the adopted BoW model, which is based on low-level feature descriptors. (2) The results are not consistent with user intention.

To solve this problem, we permit the user to instruct the computer via the MERGE operation to merge selected video clips into the same topic during the next iteration. This interaction also provides very important supervised information that we can exploit to enhance our model. In fact, a set of representative action clips are given by the video segments $\mathbf{P}_{t_1}, \mathbf{P}_{t_2}, \cdots, \mathbf{P}_{t_r}$. Without loss of generality, we denote the selected segments to be merged as $\mathbf{P}_i$ and $\mathbf{P}_j$, and then add this pair into a set $\mathcal{M} = \mathcal{M} \cup \{(i, j)\}$, whereupon the following optimization problem is solved in the next iteration:

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{U}\mathbf{V}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 + \gamma \sum_{(i,j)\in\mathcal{M}} ||\mathbf{V}_i - \mathbf{V}_j||_F^2 \\ &s.t. \, \mathbf{U} \geq 0, \;\; \mathbf{V} \geq 0.\end{aligned} \tag{3}$$

The main characteristic of this model is the third term, which biases the i-th and j-th segments to share a similar topic pattern, and γ is a trade-off parameter which is empirically set to 0.1. Please note that the pair set $\mathcal{M}$ is empty for the first iteration, and, once $\mathcal{M}$ is provided with some pair elements, it always plays a role in subsequent iterations. At the next iteration, the value of r is decreased by 1, and only $r - 1$ representative actions will be discovered. In Fig. 3 we show the illustration of this operation.

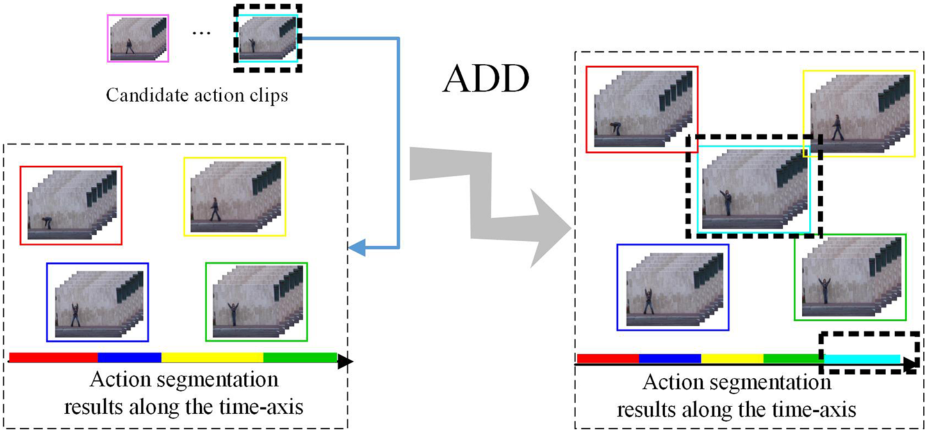
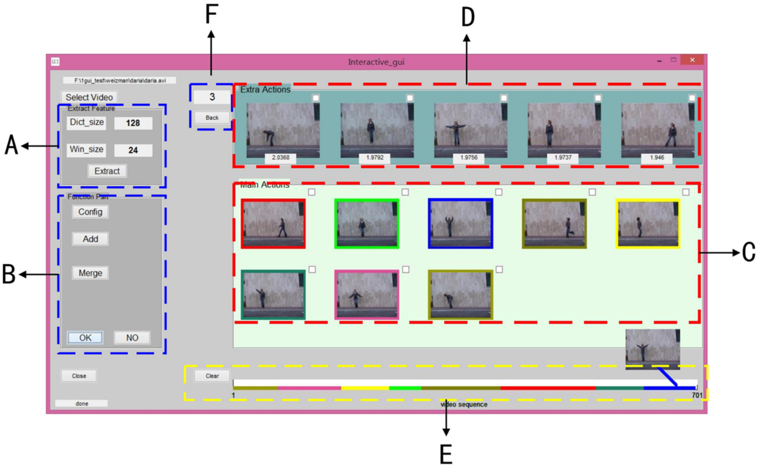

Candidate action clips
...
ADD
Action segmentation results along the time-axis
Action segmentation results along the time-axis

Fig. 4 An illustration of the ADD operation: In the left panel, the system discovers 4 representative actions, and provides a list of 5 candidate actions at the top. The user may select one candidate (surrounded by the dashed-line box) and add it into the discovered action list. The results are shown in the right panel. Please note that after this operation, the value of r and the action segmentation results are updated

4.2.2 ADD operation

Though the above model can successfully discover most of the representative actions from a video, it remains possible that some important action clips will not be discovered automatically. To this end, a candidate list of potentially representative action clips are presented to the user for performing the ADD operation. Ideally, such a list should be short and contain


F:\1gui_test\weizman\daria\daria.avi

Dict_size 128

Win_size 24

Fig. 5 The overview of the software system. **a** The parameters to control the feature extraction. Before we conduct the operations, the features for the video are extracted and stored. **b** Discovery operation part. Two buttons are visualized to conduct the ADD or MERGE operation. The expecting number of clusters is a default value or set by the user. **c** Discovery results of representative actions view. Video action clips are visualized in this part. **d** Candidate list of new action clips view. A list of candidate action clips are provided according to the entropy values in (4). **e** Action segmentation results. Different segmentations are distinguished by various colors. **f** Iteration number view

only actions that are independent of the list of discovered representative actions. In other words, the candidate list should not be well reconstructed by the discovered representative actions. Figure 4 shows this procedure. Based on the above considerations, we designed a performance index to evaluate the novelty of each action clip. For each video segment, we define its confidence regarding the topic assignment. To this end, we regard $\bar{\mathbf{V}}_i$ as the L_1 normalized vector i-th column of $\mathbf{V}_i$, and then adopt its entropy function as:

$$En(\mathbf{V}_i) = -\sum_{j=1}^{r} \bar{\mathbf{V}}_i(j) \log \bar{\mathbf{V}}_i(j). \tag{4}$$

Obviously, when only a single nonzero element of $\bar{\mathbf{V}}_i$ exists and is equal to one, the entropy is zero and the confidence score is maximum. On the other hand, when all elements of $\bar{\mathbf{V}}_i$ are equal to $1/r$, the entropy is equal to $\log_2 r$ and the confidence score is minimum. Therefore, it is very convenient to adopt the entropy when selecting the most novel video segments for the ADD operation. In this work, we sort the entropies of all previously undiscovered video segments in descending order, and then present the top N_a segments in a specifically designed manner that allows for rapid browsing by the user, and that can be selected by the user using the ADD operation for addition to the discovered representative actions in the next iteration. The number N_a cannot be too large, otherwise the number of segments presented to the user will be excessive. In this paper, it is set to 5; therefore, at each iteration stage, 5 of the most novel video segments are provided to the user for possible candidates of the ADD operation. After solving this optimization problem, the value of $\mathbf{V}$ can be used to illustrate the action segmentation results.

Once some action $\mathbf{y}_i$ is selected to be added, r must be increased by 1 in the next operation, and some further adjustments must be made. Concretely speaking, we augment the topic matrix as $\bar{\mathbf{U}} = [\mathbf{U}\,\mathbf{y}_i] \in R^{N\times(r+1)}$. The optimization problem then becomes the following.

$$\begin{aligned} &\min_{\mathbf{V}} ||\mathbf{Y} - \bar{\mathbf{U}}\mathbf{V}||_F^2 + \beta \sum_{i=1}^{N-1} ||\mathbf{V}_{i+1} - \mathbf{V}_i||_F^2 + \gamma \sum_{(i,j)\in\mathcal{M}} ||\mathbf{V}_i - \mathbf{V}_j||_F^2 \\ &s.t.\ \mathbf{V} \geq 0 \end{aligned} \tag{5}$$

Note that in the above model, $\bar{\mathbf{U}}$ is known and only $\mathbf{V}$ should be calculated.

4.3 Optimization method

All of the models given in (2), (3) and (5) can be efficiently solved by the regularized NMF method proposed in [4]. To this end, we construct a nearest neighbor graph to encode the consistency information of the data points. Consider a graph with vertices, where each vertex corresponds to a data point. Define the edge weight matrix $\mathbf{W} \in R^{N\times N}$ as follows:

$$\mathbf{W}_{ij} = \begin{cases} \beta, & \text{if } |i-j| = 1 \\ \gamma, & \text{if } \{i,j\} \in \mathcal{M} \text{ and } |i-j| \neq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

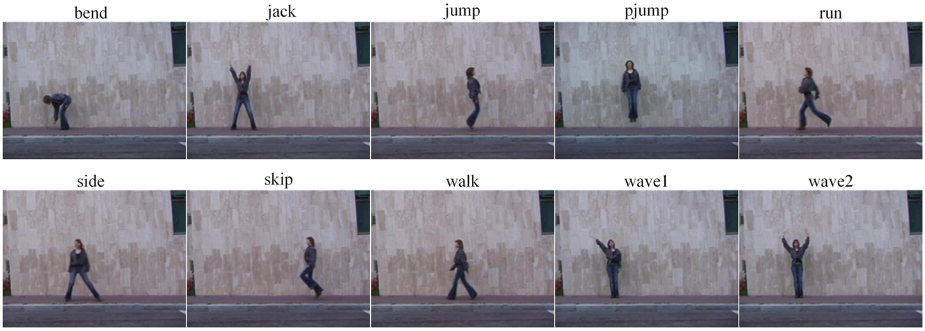


Fig. 6 Typical frames from the Weizman dataset

Define a diagonal matrix $\mathbf{D}$, whose entries are column sums of $\mathbf{W}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{W}_{ij}$. Then, the reformulated optimization problem leads to the following two update rules [4].

$$\mathbf{U}_{ij} \leftarrow \mathbf{U}_{ij} \frac{\left(\mathbf{Y}\mathbf{V}^T\right)_{ij}}{\left(\mathbf{U}\mathbf{V}\mathbf{V}^T\right)_{ij}}, \tag{7}$$

$$\mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij} \frac{\left(\mathbf{U}^T\mathbf{V} + \mathbf{V}\mathbf{W}\right)_{ij}}{\left(\mathbf{U}^T\mathbf{U}\mathbf{V} + \mathbf{V}\mathbf{D}\right)_{ij}}. \tag{8}$$

Here, the subscript ij represents the ij-th element in the corresponding matrix. A detailed algorithm flow and convergence analysis can be found in [4].

5 Performance evaluation

5.1 System description

For performance validation, we developed the graphic user interface shown in Fig. 5 to permit the user to perform the interactive operations supporting adjustments in the action discovery and segmentations so as to enable the user to fine-tune the results to satisfy their specific requirements. It also provides for visualization of the discovered action clips and the

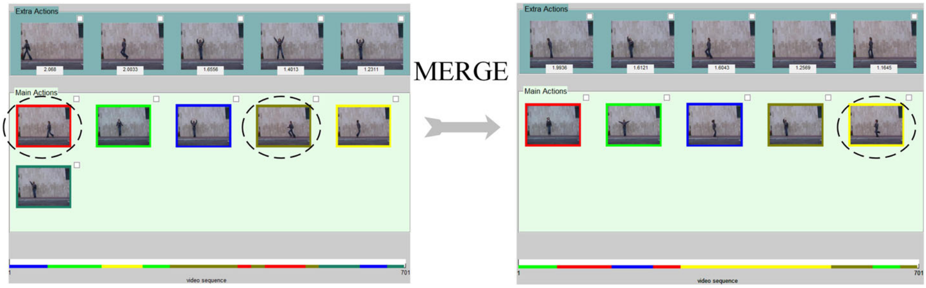


Fig. 7 Demonstration of the MERGE operation

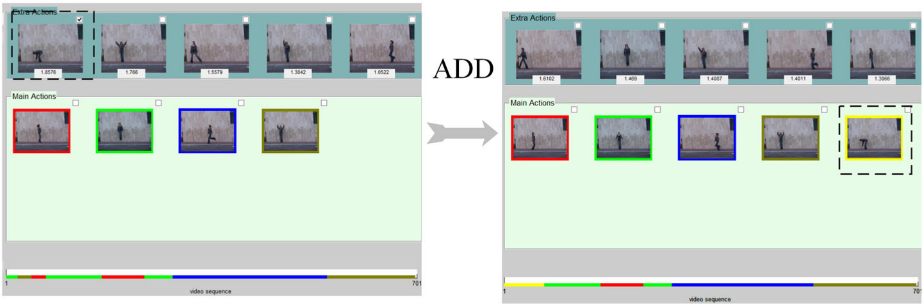


Fig. 8 Demonstration of the ADD operation

action segmentation results. In addition, the software supports save and restore operations to enable the user to return to previous analysis results.

Given a video clip, this software encodes the video as a matrix using the BoW features, as described in Section 2. For illustration, we divided the interface in Fig. 5 into 6 panels. Panel A permits the user to set some parameters that control the local motion feature extraction process. Panel B provides some interactive buttons such as those for the ADD and MERGE operations. Panel C presents the discovered action video clips. Panel D shows the 5 action video clips with the largest novelty values (i.e., the candidate list). Panel E shows the action segmentation results using different colored lines. The colors in Panel E are consistent with the colored boxes in Panel C. Finally, Panel F provides information regarding the iterations.

5.2 Dataset

Although the developed software can be used to process arbitrary action video, use of a standard dataset facilitates the analysis of the performance. To this end, we use the well-known Weizman dataset [2] of 90 low-resolution (180×144, deinterlaced 50 fps) video sequences

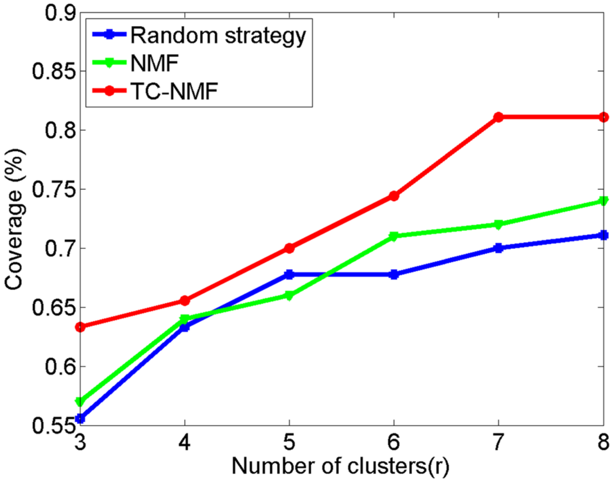


Fig. 9 Coverage curves versus the value of *r*

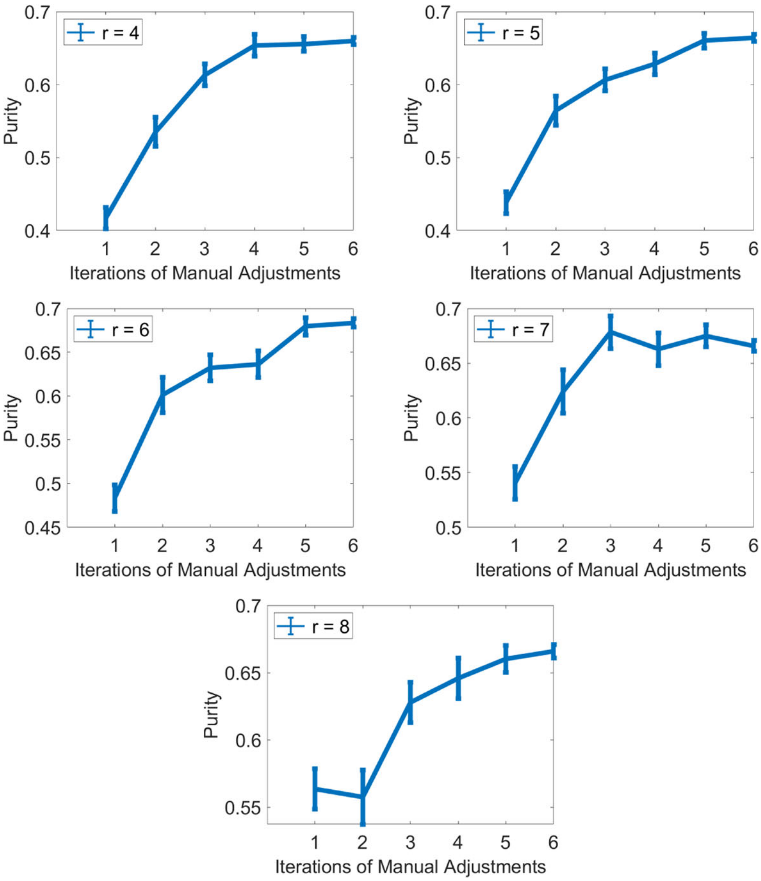


Fig. 10 The purity versus the iteration number

showing 9 different people, each performing 10 natural actions such as *run*, *walk*, *skip*, *jumping-jack* (denoted by *jack*), *jump-forward-on-two-legs*(denoted by *jump*), *jump-in-place-on-two-legs* (denoted by *pjump*), *gallop-sideways*(denoted by *side*), *wave-two-hands* (denoted by *wave2*), *wave-one-hand* (denoted by *wave1*), or *bend*. Figure 6 displays several typical frames comprised of a single person extracted from the dataset. Each video sequence in this dataset consists of a single action only. To evaluate the performance of our interactive method, we created a stitched version of the Weizman dataset comprised of uninterrupted sequences. Each sequence depicts a single person performing the 10 actions for a total duration of approximately 700 frames, and each frame has been resized to 320 × 240 to extract the features. By so doing, we can naturally obtain ground-truth label information regarding the actions.

5.3 Operation process

Figures 7 and 8 illustrate the MERGE and ADD operations, respectively. In Fig. 7, the user determines that the action video clips surrounded by the red and breen boxes are similar (see the left panel of the figure) and seeks to merge them. Selecting both action clips and clicking the Merge button leads to the results shown in the right panel of the figure, where the actions in the red and breen boxes at left are merged into a single action within a yellow box at right. From these results, we see that the two actions are indeed merged, where the number of discovered actions is decreased by 1 and the action segmentation results are also updated.

On the other hand, if the user determines that the number of discovered action clips is insufficient, the candidate action list can be browsed, as shown in Fig. 8. Here, the user selects the first candidate video clip that is surrounded by the dashed-line box, and clicks the Add button. Such an operation leads to the results shown in the right panel of the figure, where the number of the discovered action clips are observed to be increased by 1 and the action segmentation results are also updated.

5.4 Performance evaluation on the novel action discovery

To illustrate the role of the candidate action discovery module integral to the ADD operation, we designed a *Coverage* index that measures the extent to which the union set of the discovered representative action set $\mathcal{D}$ and the selected candidate action set $\mathcal{N}$ covers the ground truth action set $\mathcal{G}$. Its formal definition is

$$Coverage = \frac{|(\mathcal{D} \cup \mathcal{N}) \cap \mathcal{G}|}{|(\mathcal{D} \cup \mathcal{N}) \cup \mathcal{G}|}.$$

As a comparison, we also designed a module that randomly selects actions for the candidate list rather than selection based on the extent of novelty according to (4). In addition, we set $\beta = 0$ in (3) to form the conventional NMF model with interaction actions.

We performed experiments by changing r from 4 to 8 and then calculating the average values of *Coverage* across the 9 videos. Figure 9 shows the results, where TC-NMF represents the proposed Temporal Consistent NMF method. Both the random strategy and NMF method perform consistently inferior to the proposed TC-NMF method. This indicates that the proposed method provides rather robust results for discovering candidate action clips.

5.5 Performance evaluation on the interactive operations

To evaluate the action discovery segmentation results in each iteration, we adopt the Purity and Normalized Mutual Information (NMI) which are popular in clustering performance evaluation. To compute purity [28], each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned segments and dividing by N which is the total number of the whole video segments. For convenience, we denoting the ground-truth action clustering results as $\Omega = \left\{\omega_1, \omega_2, \ldots, \omega_{N_g}\right\}$, where N_g is the number of the cluster and ω_i represents the i-th set which includes the samples which belong to the same cluster. Similarly, we denote the semination cluster results as $\mathbb{C} = \{c_1, c_2, \ldots, c_r\}$, where r is the number of the segmentations and c_i represents the i-th set which includes the samples belonging to the same cluster. Then we can calculate the purity as

$$Purity\,(\Omega, \mathbb{C}) = \frac{1}{N_g} \sum_{k=1}^{N_g} \max_j \left|\omega_k \cap c_j\right|.$$

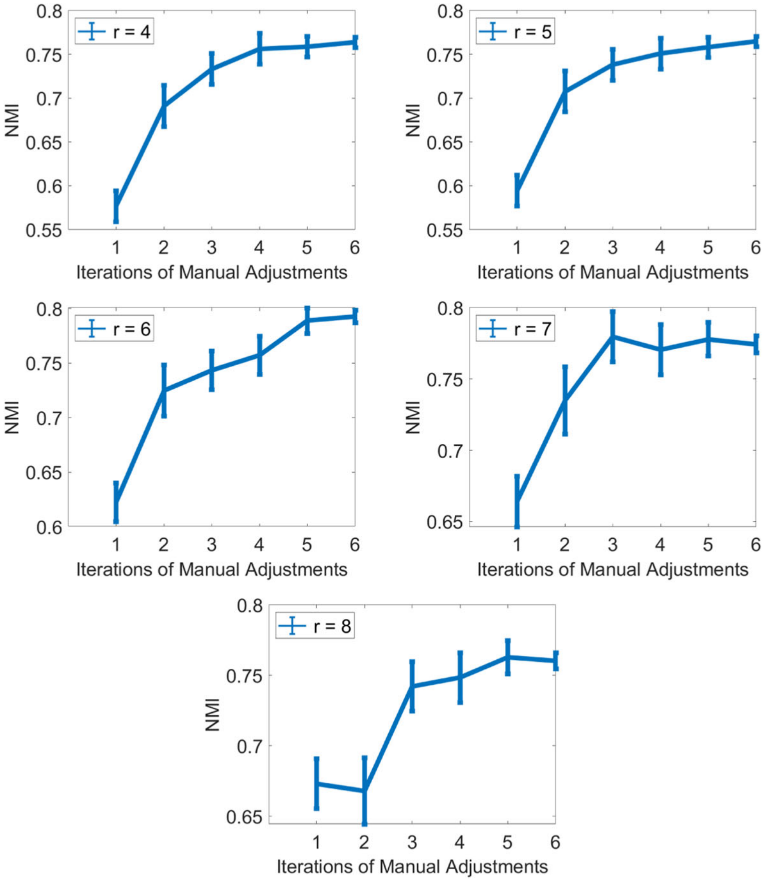


Fig. 11 The NMI versus the iteration number

On the other hand, Purity has limitations for evaluation since high purity value is easy to be achieved when the number of clusters is large. Particularly, the value of *Purity* is 1 if each segment gets its own cluster. We therefore resort NMI [28] to evaluate the performance. NMI measures the mutual dependence of the label and the cluster assignment. The minimum of the NMI is 0 if the clustering assignment is independent to the label and 1 if they are perfectly aligned. The value of NMI is calculated as the mutual information normalized by the average entropy of label distribution and cluster assignment distribution:

$$\mathbf{NMI}\,(\Omega, \mathbb{C}) = \frac{I\,(\Omega, \mathbb{C})}{[H\,(\Omega) + H\,(\mathbb{C})]/2},$$

where I is the mutual information and $H(\cdot)$ represents the entropy. The mutual information $I(\Omega, \mathbb{C})$ is defined by $I(\Omega, \mathbb{C}) = \sum_{k=1}^{N_g} \sum_{j=1}^{r} P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_k)}$. High NMI is also easy to achieve when the number of clusters is small - in particular, NMI is 1 if all data are assigned to a single cluster, which is the opposite of purity.

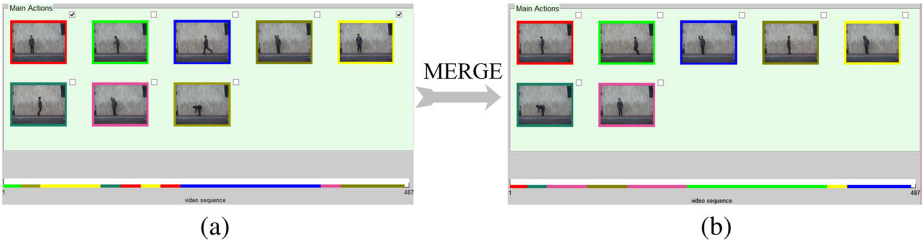


Fig. 12 Illustration of the MERGE operation. In this case, the values of *Purity* and *NMI* may decrease because the number of the clusters is decreased

In our experiments the interactive action discovery and segmentation performance was tested on the same data set by 5 users who were not aware of the video content. The user was allowed to make at most 6 manual adjustment operations (including MERGE and ADD) to achieve satisfactory results. Figures 10 and 11 illustrate the averaged performance using the proposed interactive method with different initial values of r. The results indicate that the clustering performance is significantly improved by adopting manual interactive adjustments.

A strange phenomenon occurs when $r = 8$. In this case, we find that the values of *Purity* and *NMI* decrease at the first manual adjustment iteration. A similar phenomenon occurred in the fourth-iteration when $r = 7$. The reason is that, when r is large, similar representative actions will be extracted, and users first employ the MERGE operation. In practice, the MERGE operation tends to decrease the values of *Purity* and *NMI* because the number of clusters is decreased. An example is shown in Fig. 12. However, these values are generally increased during subsequent iterations using ADD operations.

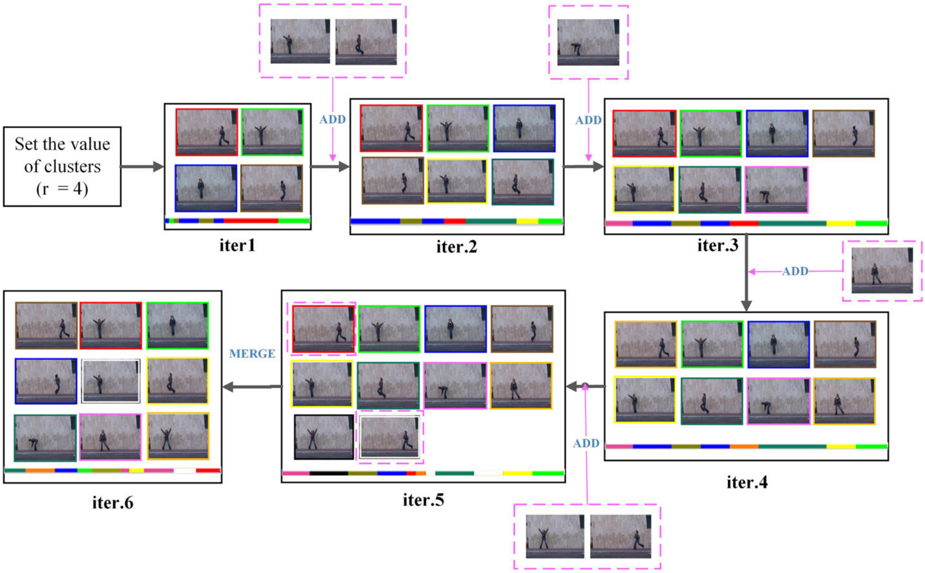


Fig. 13 An illustration of the user study process

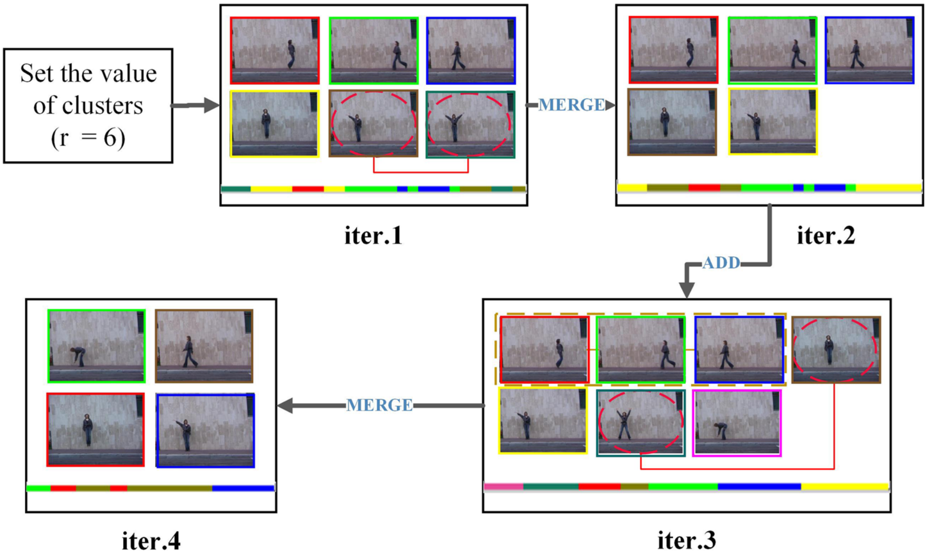


Fig. 14 An illustration of the user study process incorporating user intention

5.6 User study

Using the developed system, the user is able to discover as many actions as possible after performing several specific operations, as illustrated in Fig. 13, where the corresponding frames at each iteration are shown. Because the user is unaware of the exact number of actions in a video sequence, the value of *r* is initially set to 4 in this experiment. In this way, 4 representative actions are obtained, and, simultaneously, a list of 5 candidate actions are presented to the user on the interface. By performing the ADD operation, two novel actions are added into the clustering results, and the candidate actions are updated. By repeating such operations, ten representative actions are eventually discovered. However, minor mistakes are inevitable due to the high similarity of *walk* and *run*. To solve the problem, the user merges these two actions, and the 9 remaining representative actions are finally shown in Fig. 13.

Another merit of the developed system is that it can easily incorporate user intention to obtain personalized action discovery and segmentation results. Figure 14 shows such an example. In this experiment, the user intention is incorporated into the optimization model and biases the algorithm to output the expected results shown in the figure. Because the default value of *r* is set to 6, the user initially obtains 6 representative actions including *jump*, *run*, *walk*, *pjump*, *wave1*, *wave2*. After a quick browse of the discovery results, the user determines that *wave1* and *wave2* are similar and performs the MERGE operation with them. The system then outputs 5 representative actions. The user then determines that some candidate actions such as *pjump*, *jack* should be added and the ADD operation is performed. The system then outputs 7 actions. Finally, the user chooses to merge *run*, *walk* and *jump*, and the system outputs 4 representative actions labeled *Bend*, *Walk*, *Jump*, *Wave*. Other actions in the groundtruth label are clustered into these 4 categories. This reflects the user's intention, whose concern is with only 4 action categories: (1)*Jump*, which includes *pjump* and *jack*; (2)*Wave*, which includes *wave1* and *wave2*; (3)*Walk*, which includes *run*, *walk*, *jump*, *side*, and *skip*; (4)*Bend*, which includes *bend*.

6 Conclusion and future work

This paper proposed an interactive method to detect representative actions within streaming or archival video. By incorporating user intention, the expected results have been obtained. The main limitation of this work is that the performed operations are basic elements and therefore the complicated video structure cannot be sufficiently exploited. In the future, extensive further investigation is required. Firstly, we wish to extend the work conducted on single videos to video sets, and discover more interpretable behavior patterns for end users. Secondly, we hope to develop a more flexible interface, and incorporate a greater extent of high-level knowledge related to human perception into the model. Finally, we wish to discover the hierarchical structure of the video action in a coarse-to-fine manner.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant U1613212, Grant 61673238, in part by the Beijing Municipal Science and Technology Commission under Grant D171100005017002, and in part by the National High Technology Research and Development Program of China under Grant 2016YFB0100903.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Amato FF, Castiglione A, Moscato V et al (2018) Multimedia summarization using social media content[J]. Multimed Tools Appl, 1–25
2. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Proceedings of international conference on computer vision (ICCV), pp 1395–1402
3. Borzeshi E, Concha O, Xu R, Piccardi M (2013) Joint action segmentation and classification by an extended Hidden Markov model. IEEE Signal Process Lett, 1207–1210
4. Cai D, He X, Wu X, Han J (2008) Non-negative matrix factorization on manifold. In: Proceedings of international conference in data mining (ICDM), pp 63–72
5. Chang XX, Yang Y (2017) Semisupervised feature analysis by mining correlations among multiple tasks[J]. IEEE Trans Neural Netw Learn Syst 28(10):2294–2305
6. Chang X, Nie F, Wang S et al (2016) Compound rank-*k* projections for bilinear analysis[J]. IEEE Trans Neural Netw Learn Syst 27(7):1502–1513
7. Chang X, Yu Y, Yang Y et al (2017) Semantic pooling for complex event analysis in untrimmed videos[J]. IEEE Trans Pattern Anal Mach Intell 39(8):1617–1632
8. Chen Y, Rege M, Dong M, Hua J (2007) Incorporating user provided constraints into document clustering. In: Proceedings of international conference on data mining (ICDM), pp 103–112
9. Chen S, Xin Y, Luo B (2016) Action-based pedestrian identification via hierarchical matching pursuit and order preserving sparse coding. Cognitive Computation
10. Choo J, Lee C, Reddy C, Park H (2013) Utopian: user-driven topic modeling based on interactive nonnegative matrix factorization. IEEE Trans Visual Comput Graph 19(12):1992–2001
11. Cui P, Wang F, Sun L, Zhang J, Yang S (2012) A matrix-based approach to unsupervised human action categorization. IEEE Trans Multimed, 102–110
12. Hossain M, Ojili P, Grimm C, Muller R, Watson L, Ramakrishnan N (2012) Scatter/gather clsutering: flexibly incorporating user feedback to steer clustering results. IEEE Trans Visual Comput Graph 18(12):2829–2838
13. Hu T, Zhu X, Guo W et al (2018) Human action recognition based on scene semantics[J]. Multimed Tools Appl, 1–22

14. Huang H, Fu S, Cai Z et al (2018) Video abstract system based on spatial-temporal neighborhood trajectory analysis algorithm[J]. Multimed Tools Appl, 1–18
15. Hughes M, Sudderth E (2012) Nonparametric discovery of activity patterns from video collections. In: Proceedings of computer vision and pattern recognition workshops (CVPRW), pp 25–32
16. Kumaran N, Vadivel A, Kumar S (2018) Recognition of human actions using CNN-GWO: a novel modeling of CNN for enhancement of classification performance[J]. Multimed Tools Appl, 1–33
17. Lee D, Seung H (2001) Algorithms for non-negative matrix factorization. Adv Neural Inf Process Syst, 556–562
18. Liu H, Liu Y, Yu Y, Sun F (2014) Diversified key-frame selection using structured $L_{2,1}$ optimization. IEEE Trans Indus Inform 10(3):1736–1745
19. Liu H, Liu H, Sun F, Fang B (In press) Kernel regularized nonlinear dictionary learning for sparse coding. IEEE Trans Syst Man Cybern Syst. https://doi.org/10.1109/TSMC.2017.2736248
20. Luo M, Nie F, Chang XX et al (2017) Adaptive unsupervised feature selection with structure regularization[J]. IEEE Transactions on Neural Networks and Learning Systems
21. Ma Z, Chang X, Xu Z et al (2017) Joint attributes and event analysis for multimedia event detection[J]. IEEE Transactions on Neural Networks and Learning Systems
22. Shao L, Jones S, Li X (2014) Efficient search and localization of human actions in video databases. IEEE Trans Circ Syst Video Technol 24(3):504–512
23. Tang J, Lewis P (2008) Non-negative matrix factorization for object class discovery and image auto-annotation. In: Proceedings of international conference on content-based image and video retrieval (CIVR), pp 105–112
24. Tu Z, Abel A, Zhang L, Luo B, Hussain A (2016) A new spatio-temporal saliency-based video object segmentation. Cognitive Computation
25. Wang M, Ji D, Tian Q, Hua X (2012) Intelligent photo clustering with user interaction and distance metric learning. Pattern Recogn Lett, 462–470
26. Zhao B, Xing E (2014) Quasi real-time summarization for consumer videos. In: Proceedings of computer vision and pattern recognition (CVPR), pp 2513–2520
27. Zhao G, Qin S, Wang D (2018) Interactive segmentation of texture image based on active contour model with local inverse difference moment feature. Multimed Tools Appl, 1–28
28. Evaluation of clustering: http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Huaping Liu is an associate professor in Department of Computer Science and Technology, Tsinghua University. He serves as Associate Editor of some journals including IEEE Robotics & Automation Letters, Neurocomputing, Cognitive Computation, and some conferences including ICRA and IROS. His research interests include robot perception and learning.

Fuchun Sun is a full professor in Department of Computer Science and Technology, Tsinghua University. He is the recipient of National Science Fund for Distinguished Young Scholars. He serves as Associate Editor of a series of international journals including IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON SYSMTES, MAN AND CYBERNETICS: SYSTEMS, Mechatronics, Robotics and Autonomous Systems. His research interests include intelligent control and robotics.

Xinyu Zhang is an associate professor in State Key Laboratory of Automotive Safety and Energy, Tsinghua University. He serves as vice general secretary of Chinese Association for Artificial Intelligence and a visiting scholar at Cambridge. His research interests include unmanned system platform and robotics.

Bin Fang received the Ph.D. degree from Beihang University, Beijing, China, in 2014. He is now an assistant professor in Department of Computer Science and Technology, Tsinghua University, Beijing. His research interests include robotic perception and interaction.