

# Dual Clustering for Categorization of Action Sequences\*

Joanna Cheng, Liang Wang and Christopher Leckie  
Dept. Computer Science & Software Engineering  
The University of Melbourne, Parkville, Vic 3010, Australia

## Abstract

*This paper proposes a novel algorithm for categorization of action video sequences using unsupervised dual clustering. Given a video database, we extract motion information of actions and perform nonlinear dimensionality reduction for addressing both the high dimensionality of silhouette features and non-linearity of articulated human actions. A  $k$ -means clustering is first performed on frame-wise features in the embedding space to convert each video in the database to a sequence of labels, each of which corresponds to one of  $k$  “key” feature frames. The dissimilarity between any two label sequences is then measured using edit distance. The resulting pairwise dissimilarity matrix is finally input to a spectral clustering algorithm to obtain the category labels of each action video. Experimental results on two recent data sets demonstrate the effectiveness and efficiency of the proposed algorithm.*

## 1. Introduction

Automatic categorization of different human actions in video sequences is receiving growing interest due to a variety of potential applications such as smart surveillance, video summarization and digital library management. However it remains a challenge due to temporal and spatial variations of actions themselves between different subjects, as well as cluttered backgrounds, occlusions and camera motions. Several attempts have been recently made for unsupervised learning of human action categories from images or videos [7, 10, 5, 11]. For example, in [7] a video sequence is represented as a collection of spatial-temporal words by extracting space-time interest points. The algorithm automatically learns the probability distributions of the spatial-temporal words and intermediate topics corresponding to human action categories using a probabilistic Latent

Semantic Analysis model; in [10], the problem of describing an action being performed in *still* images is addressed, in which the coarse shape of the human figure is used to match pairs of images; and in [11] dynamic events are characterized by spatiotemporal gradient features at multiple temporal scales, in which a statistical distance measure between video sequences is designed based on their behavioral content and is used for clustering events within long continuous video sequences.

This paper focuses on automatic categorization of action sequences. Action determination in a single image frame (like [10]) might create ambiguity since different actions may involve several similar poses (or images); shape [5, 10], points of interest [7] or spatiotemporal gradient [11] based feature extraction might be not reliable in cases of different imaging conditions, smooth surfaces, motion singularities and low-quality videos; and the large volume of video data to be processed usually makes the unsupervised categorization algorithms too slow to be of practical use. This paper explores a two-phase method for categorization of action sequences. It essentially consists of two clustering processes, i.e., frame-wise clustering and sequence-wise clustering. Experiments on two state-of-the-art databases show that the proposed approach achieves satisfactory results with computational efficiency.

## 2. Approach

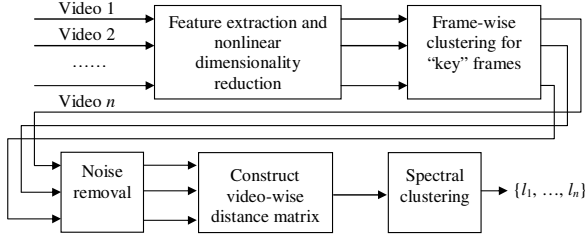
The proposed approach consists of 5 basic steps, as shown in Figure 1, each of which is detailed as follows.

### 2.1. Feature extraction and representation

Given a data set of  $n$  action sequences, i.e.,  $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_n\}$ , we wish to extract useful motion information from original videos to represent the actions performed. Human actions can be implicitly regarded as temporal variations of human silhouettes across image frames, so we use space-time silhouettes, together with kernel-induced subspace analysis,

\*This work is partially supported by Australian Research Council (ARC) Discovery Project (Grant No: DP0663196)

for compact feature extraction and representation.



**Figure 1. Framework of dual clustering for categorization of action sequences**

For each action sequence in the database  $\mathcal{V}$ , denoted by  $\mathbf{V}_i$  consisting of  $T_i$  image frames, i.e.,  $\mathbf{V}_i = \{\mathbf{I}_1^i, \mathbf{I}_2^i, \dots, \mathbf{I}_{T_i}^i\}$  ( $i = 1, 2, \dots, n$ ), we assume that the associated sequence of human movement silhouettes can be extracted by foreground segmentation techniques. Since the size and position of the foreground region vary with the distance of the human to the camera, the size of the human and the performed action, we center and normalize the silhouette images so that the resulting images  $\{\mathbf{S}_1^i, \mathbf{S}_2^i, \dots, \mathbf{S}_{T_i}^i\}$  do not distort the motion shape and are of equal dimensions  $D_r \times D_c$ . If we further represent each silhouette image  $\mathbf{S}_j^i$  ( $j = 1, \dots, T_i$ ) as a vector  $\mathbf{s}_j^i$  in  $\mathbb{R}^{D_r \times D_c}$  in a row-scan manner, the video  $\mathbf{V}_i$  will be accordingly represented as  $\mathbf{V}_i' = \{\mathbf{s}_1^i, \mathbf{s}_2^i, \dots, \mathbf{s}_{T_i}^i\}$ .

To obtain both a compact description and efficient computation, we use Kernel PCA (Principal Component Analysis) [8] to perform nonlinear dimensionality reduction of silhouette features. We use all videos, i.e.,  $\{\mathbf{s}_1^1, \dots, \mathbf{s}_{T_1}^1, \mathbf{s}_1^2, \dots, \mathbf{s}_{T_2}^2, \dots, \mathbf{s}_1^n, \dots, \mathbf{s}_{T_n}^n\}$ , to learn a low-dimensional subspace. The Gaussian kernel function  $K(\mathbf{s}_i, \mathbf{s}_j) = \exp(-\frac{\|\mathbf{s}_i - \mathbf{s}_j\|^2}{2\sigma^2})$  is used in our experiments. After KPCA, each video will be transformed to a  $m$ -dimensional signal form,  $\mathbf{V}_i'' = \{\mathbf{f}_1^i, \mathbf{f}_2^i, \dots, \mathbf{f}_{T_i}^i\}$  in  $\mathbb{R}^m$  ( $m \leq D_r \times D_c$ ).

## 2.2. Frame-wise clustering

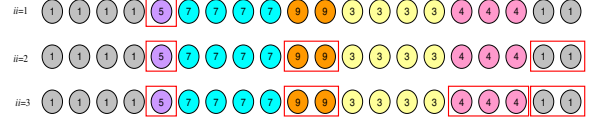
Instead of computing the sequence similarity in feature space like most existing methods, we seek another efficient symbolic representation to map each video into a label sequence. Each label corresponds to one of  $k$  pre-learned representative poses (named “key” frames), based on the premise that certain important poses are sufficient to represent and identify an action. To achieve this, we use a simple and fast  $k$ -means clustering to group all the image frames in the data set in the embedding space to identify such “key” frames.

The number of clusters  $k$  (i.e., the number of “key” frames) is an important consideration. We wish to ob-

tain a good tradeoff between enough representation of the actions to be analyzed and computation efficiency. Let  $\mathcal{K}$  be the set of “key” frames, each of which is labeled using an integer, say  $\{1, 2, \dots, k\}$ . The image frames in each video are thus replaced by the label of the corresponding cluster to which that frame belongs. This transforms each video  $\mathbf{V}_i$  ( $i = 1, \dots, n$ ) into a string of  $T_i$  ordered cluster labels, denoted by  $\mathbf{V}_i'''$ .

## 2.3. Isolated noise removal

The silhouette images could be noisy due to imperfect foreground segmentation. To compensate for this potential influence, we wish to remove such noise from the label sequences. Due to the temporal order relation between frames, we expect that several neighboring frames in the sequence should generally share the same label of “key” frames. We thus regard isolated label(s) as noise, and remove them from the original label sequences. Figure 2 shows examples of how to remove a subsequence of up to  $ii$  successive identically-numbered frames. Note that different values for  $ii$  are used to determine different degrees of (potential) noise. When  $ii = 0$ , the original label sequences are kept. A suitable value of  $ii$  is selected to make sure that it does not affect the accuracy of the final clustering while removing noise.



**Figure 2. Example of which frames of a label sequence would be removed for different values of  $ii$**

## 2.4. Pairwise distance measure

After obtaining (filtered) symbolic video representations, we define a distance function to measure the dissimilarity between any two label sequences. The Levenshtein edit distance [3] is chosen due to its properties of preserving temporal order and the ability to recognize subsequences and cyclic shifting of sequences. This is ideal for the database which probably consists of periodic actions such as walking and running, since the same actions may vary in the starting/ending poses and the action durations between different instances.

We construct the pairwise distance matrix as follows: 1) Let  $\mathbf{M}$  be a  $n \times n$  matrix. For sequences  $i$  and  $j$  ( $i \neq j$ ), calculate the Levenshtein edit distance  $d_{ij}$  between  $\mathbf{V}_i'''$  and  $\mathbf{V}_j'''$ . When  $i = j$ ,  $d_{ij} = 0$ . 2) Set  $M_{ij} = d_{ij} / \max(T_i, T_j)$ , where  $T_i$  is the length of sequence  $i$ .

This ensures distances are normalised by the length of the sequences being compared, as the upper bound for the edit distance is the length of the longer sequence. 3) Since this distance is symmetric, set  $M_{ji} = M_{ij}$ .

## 2.5. Video-wise clustering

After generating the distance matrix  $\mathbf{M}$ , we use a spectral clustering algorithm [6] to obtain  $c$  final clusters, each of which represents a unique action class. We summarize the steps of spectral clustering as follows:

- Form the affinity matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  defined by  $A_{ij} = \exp(-M_{ij}^2/\tau^2)$  for  $i \neq j$ , and  $A_{ii} = 0$ , where  $\tau$  is a global scale parameter.
- Define  $\mathbf{D}$  to be the diagonal matrix with  $D_{ii} = \sum_{j=1}^n A_{ij}$  (i.e., the  $(i, i)$  element is the sum of  $\mathbf{A}$ 's  $i$ -th row), and construct the normalized affinity matrix  $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ .
- Find  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_c$ , the  $c$  largest eigenvectors of  $\mathbf{L}$  (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix  $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_c] \in \mathbb{R}^{n \times c}$  by stacking the eigenvectors in columns.
- Re-normalize the rows of  $\mathbf{E}$  to have unit length to generate  $\mathbf{G} \in \mathbb{R}^{n \times c}$ , i.e.,  $G_{ij} = E_{ij}/(\sum_j E_{ij}^2)^{1/2}$ .
- For  $i = 1, 2, \dots, n$ , let  $\mathbf{g}_i \in \mathbb{R}^c$  be the vector corresponding to the  $i$ -th row of  $\mathbf{G}$ , cluster them into  $c$  groups  $B_1, \dots, B_c$  via  $c$ -means.
- Assign sequence  $i$  to cluster  $j$  if and only if the corresponding row  $i$  of  $\mathbf{G}$  was assigned to cluster  $j$ , thus obtaining final clusters  $C_1, \dots, C_c$  with  $C_j = \{i | \mathbf{g}_i \in B_j\}$ .

## 3. Experiments

### 3.1. Evaluation datasets

Different instances of the same action may consist of varying relative speeds. Dataset I [9] consists of  $n = 100$  video sequences from 10 different actions performed by one subject (i.e., pick up object, jog in place, push, squash, wave, kick, bend to the side, throw, turn around, and talk on cell phone) and 10 different instances for each action. This dataset is mainly used to examine the effect of the temporal rate of execution on action analysis. Sample images are shown in Figure 3.

There exist inter-person differences between the same actions due to different physical sizes and motion

styles (and speeds). Dataset II [1] consists of 90 low-resolution videos from 9 different people, each performing 10 actions (i.e., bend, jump jack, jump-forward-on-two-legs, jump-in-place-on-two-legs, run, skip, gallop-sideways, walk, wave-one-hand, and wave-two-hands). This dataset provides more realistic data for the test of the method's versatility in terms of variations at both temporal and spatial scales. Sample images are shown in Figure 4.

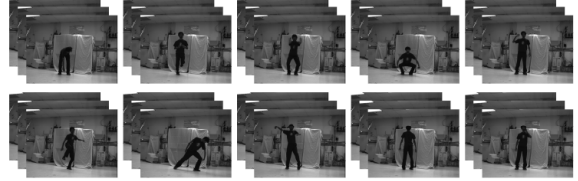


Figure 3. Sample images from Dataset I



Figure 4. Sample images from Dataset II

### 3.2. Clustering accuracy metric

We evaluate our algorithm's performance by comparing the cluster label of each video given by our algorithm with the ground truth (available in our case), as well as the computation time. The accuracy ( $AC$ ) metric has been widely used for clustering performance evaluation [2], which is defined by

$$AC = \frac{\sum_{i=1}^n \delta(l_i^g; \text{map}(l_i^c))}{n}$$

where  $l_i^c$  is the clustering label result of a given example  $i$ ;  $l_i^g$  is the real ground truth label;  $\delta(l_1; l_2)$  is the delta function that equals 1 if and only if  $l_1 = l_2$ , and 0 otherwise; and  $\text{map}$  is the mapping function that permutes clustering labels to match equivalent labels given by the ground truth. The Kuhn-Munkres algorithm [4] is usually used to obtain the best mapping.

### 3.3. Data processing

These two data sets are provided with silhouette masks. We center and normalize silhouette images into  $64 \times 48$  resolution, leading to 3072-dimensional silhouette representation. We use KPCA for nonlinear dimensionality reduction, in which the Gaussian kernel

function’s argument  $\sigma$  is set to be 1000. The reduced dimension  $m$  is set to 25. In the 25-dimensional embedding space, we perform  $k$ -means clustering ( $k = 20, 25$  and  $30$ ) to convert videos to corresponding label sequences, based on which we construct a pairwise distance matrix for spectral clustering. We set the number of clusters  $c = 10$  (i.e., the number of real physical classes in these two datasets). As a baseline, to establish how much the temporal information in a sequence affects accuracy, we also propose a purely frequency-based algorithm, in which the frequency of each numbered frame is calculated for each sequence.

### 3.4. Results and discussion

For each database, we perform our algorithm 50 times, and reported results in terms of average accuracy  $a$  and average computation time  $t$  in Tables 1 and 2. The results show that the accuracy is not very sensitive to the choice of the number of key frames between 20 and 30 (corresponding to 2-3 frames per action class). This suggests that 2-3 representative poses may be enough to represent and identify the actions in these datasets.

**Table 1. The results on Dataset I**

	# “key” frames in $k$ -means clustering					
	20		25		30	
	$a(\%)$	$t(s)$	$a(\%)$	$t(s)$	$a(\%)$	$t(s)$
$ii = 0$	99.3	65.7	99.3	65.8	99.5	66.3
$ii = 1$	99.3	59.7	99.4	59.3	99.5	59.0
$ii = 2$	98.2	54.3	98.3	53.5	98.3	52.2
$ii = 3$	97.0	49.0	97.1	47.4	96.5	45.7
freq	97.6	0.06	99.0	0.06	99.1	0.06

**Table 2. The results on Dataset II**

	# “key” frames in $k$ -means clustering					
	20		25		30	
	$a(\%)$	$t(s)$	$a(\%)$	$t(s)$	$a(\%)$	$t(s)$
$ii = 0$	80.4	22.4	82.4	22.2	79.0	22.3
$ii = 1$	78.6	17.9	77.7	17.4	77.2	17.0
$ii = 2$	74.6	13.3	75.0	12.5	74.4	11.9
$ii = 3$	70.5	9.62	70.2	8.54	70.0	7.95
freq	79.7	0.07	76.6	0.07	74.6	0.08

Overall, better clustering results are obtained when the sequence of frames is left unchanged (i.e.,  $ii = 0$ ), or only single frames are removed (i.e.,  $ii = 1$  for Dataset I). This is probably because the silhouette masks in these two data sets are not very noisy. In that case, as  $ii$  increases, the accuracy naturally decreases since more useful information is lost, though in less computation time.

The original label sequence (i.e.,  $ii = 0$ ) generally outperforms the frequency-based approach, showing that the temporal information of motions does help improve the clustering accuracy. Although the frequency-based approach outperforms the subsequence-removal approach in an overall view (i.e.,  $ii \geq 1$ ), this is mainly due to the fact that removing longer subsequences will lose more useful information, especially when the input sequences are not overly noisy. Note that a benefit of the frequency-based approach is computation time. The majority of the computation time of our algorithm is in the generation of the pairwise distance matrix using the Levenshtein edit distance which is essentially based on dynamic programming.

### 4. Conclusion

We have presented a computationally efficient dual clustering approach to unsupervised categorization of action sequences. Empirical results on two recent video databases show that our method can effectively discover the action categories in video data, and that by exploiting temporal order information, we can achieve better accuracy than a purely frequency-based approach.

### References

- [1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Action as space-time shapes. *ICCV*, 2005.
- [2] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Trans. Knowledge and Data Engineering*, 17(2):1624–2637, 2005.
- [3] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.
- [4] L. Lovasz and M. Plummer. Matching theory. *North Holland, Budapest: Akademiai Kiado*, 1986.
- [5] G. Loy, J. Sullivan, and S. Carlsson. Pose-based clustering in action sequences. *ICCV Workshop on Higher-level Knowledge in 3D Modelling and Motion Analysis*, 2003.
- [6] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. *NIPS*, 2001.
- [7] J. Niebles, H. Wang, and F. F. Li. Unsupervised learning of human action categories using spatial-temporal words. *BMVC*, 2006.
- [8] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [9] A. Veerarghavan, R. Chellappa, and A. Roy-Chowdhury. The function space of an activity. *CVPR*, 2006.
- [10] Y. Wang, H. Jiang, M. Drew, Z. Li, and G. Mori. Unsupervised discovery of action classes. *CVPR*, 2006.
- [11] L. Zelnik-Manor and M. Irani. Event-based video analysis. *CVPR*, 2001.