Rushes Video Segmentation Using Semantic Features

Athina Pappa, Vasileios Chasanis, and Antonis Ioannidis

Department of Computer Science and Engineering, University of Ioannina,
GR 45110, Ioannina, Greece
{apappa, vchasani, aioannid}@cs.uoi.gr

Abstract. In this paper we describe a method for efficient video rushes segmentation. Video rushes are unedited video footage and contain many repetitive information, since the same scene is taken many times until the desired result is produced. Color histograms have difficulty in capturing the scene changes in rushes videos. In the herein approach shot frames are represented by semantic feature vectors extracted from existing semantic concept detectors. Moreover, each shot keyframe is represented by the mean of the semantic feature vectors of its neighborhood, defined as the frames that fall inside a window centered at the keyframe. In this way, if a concept exists in most of the frames of a keyframe's neighborhood, then with high probability it exists on the corresponding keyframe. By comparing consecutive pairs of shots we seek to find changes in groups of similar shots. To improve the performance of our algorithm, we employ a face and body detection algorithm to eliminate false boundaries detected between similar shots. Numerical experiments on TRECVID rushes videos show that our method efficiently segments rushes videos by detecting groups of similar shots.

Keywords: Rushes summarization, semantic concept detectors, face detection.

1 Introduction

Video rushes segmentation and summarization is an important task in video processing. Rushes are unedited video used for movie and documentary editing. The duration of rushes videos is often ten times larger than the duration of the corresponding edited video. Thus, video rushes segmentation is necessary in order to provide fast access to video data to montage editors. The basic problems of rushes videos are three. First, the presence of useless frames such as colorbars, monochrome frames and frames containing clapboards. Second, the repetition of similar segments produced from multiple takes of the same scene and finally, the efficient representation of the original video in the video summary. In this paper, we focus on finding similar segments/shots that are captured under various circumstances, such as different camera positions or luminance conditions, changed background and even characters. Given the fact that similar shots are time ordered, grouping similar shots can be regarded as a video segmentation problem.

In [1], HSV color histograms and a sequence alignment algorithm are employed to cluster similar shots. In [2], histograms of dc-images are employed to compute similarity between extracted keyframes. In [3], hierarchical clustering on keyframes represented by Color layout and edge histograms is employed to group similar shots into

sets. In [4], spectral clustering on HSV color histograms is employed to cluster shots into groups, followed by a sequence alignment similarity metric.

In the method we propose herein, each video is segmented into shots, shots containing colorbars or monochrome frames are removed and for each shot we extract keyframes using the method described in [5]. Then, we define a neighborhood for each keyframe of a video shot. The neighborhood of a key-frame contains the frames that fall inside a window centered at the keyframe. For each frame of the neighborhood we compute semantic feature vectors based on semantic concept detectors available in bibliography ([6], [7]). Finally, each keyframe is represented by the mean of the semantic feature vectors of its neighborhood. A unique characteristic of rushes videos is that similar shots are time ordered, thus when a scene is recorded, a new group of similar shots is formed and ends when a new scene begins. With term "scene" we refer to similar shots produced from multiple takes of the same scenery. In our method, we seek to find when a new group of similar shots is formed, thus we compare successive shots to find scene boundaries. To improve the performance of our method, we employ a face and body detection algorithm.

The contribution of our method is three-fold. Firstly, each frame is represented with semantic feature vectors based on common semantic concept detectors. Color histograms have difficulty in capturing the scene changes in rushes videos, whereas the proposed feature vectors provide reliable segmentation. Secondly, to compute the semantic vector for each keyframe we also consider the semantic feature vectors of its neighboring frames. The neighborhood of a key-frame contains the frames that fall inside a window centered at the keyframe. The reason behind such a representation is that the extracted keyframe may not describe sufficiently the concepts of the video shot due to incorrect keypoints detection and description. In other words, by using the neighborhood of each keyframe, we aim to provide a more reliable representation with respect to the probability that certain concepts exist in a video shot. Finally, we employ a face and body detection algorithm to eliminate false detections of scene boundaries between successive shots that contain similar faces/bodies.

The rest of the paper is organized as follows. In Section 2 we describe the computation of semantic feature vectors and the representation of frames and shots. In Section 3 we present rushes segmentation and performance improvement using a face and body detection algorithm. In Section 4 we present numerical experiments. Finally, in Section 5 we conclude our work.

2 Semantic Features

Each video is segmented into shots manually to assess the performance of our method without having any errors introduced from the shot segmentation process. Moreover, for each shot, we extract keyframes using the method described in [5].

2.1 Frame Representation

Semantic concept detectors are employed in order to extract semantic features for each shot frame. Two different databases of semantic concept detectors are employed in the

herein approach. The first one, named Vireo-374 detectors [6], is trained on TRECVID-2005 [8] development data using LSCOM annotation [9]. In order to train these detectors DoG detector and SIFT descriptor [10] are used for keypoint detection and description. The bag of visual words representation [11] is implemented for frame representation. More than 500k SIFT features are clustered into 500 visual words (visual vocabulary). For each frame, its corresponding set of descriptors is mapped into these 500 visual words resulting into a vector containing the normalized count of each visual word in the frame. The soft-weighting scheme [6] is used to weight the significance of each visual word in the frame is used, which has been demonstrated to be more effective than the traditional tf/tf-idf weighting schemes. LibSVM package [12] and Chi-square kernel are used for model training and prediction on test data.

The second database of semantic detectors, named Vireo-Web81 [7], is trained on approximately 160k images taken from social media such as Flickr. 81 concept detectors are trained with settings similar to Vireo-374.

The output of each SVM is a number in the continuous range [0,1], expressing the probability that each frame is related to the corresponding concept. Each shot frame is tested on 374 and 81 semantic detectors of video-374 and web81 databases, respectively. Thus, given K concepts, a frame f is represented from the following semantic vector:

$$v(f) = [c_1(f), c_2(f), \ldots, c_K(f)], \tag{1}$$

where $c_i(f)$ is the probability that frame f is related to concept c_i, $i = 1, \ldots, K$. Thus, vector v is a representation of frame f in the semantic space defined by K concepts. In the herein approach, for each frame f, two semantic feature vectors are computed, $v_{374}(f)$ and $v_{81}(f)$, corresponding to Vireo-374 [6] and Vireo-Web81 [7] detectors, respectively.

2.2 Shot Representation

Each shot is represented by a certain number of keyframes extracted using the method presented in [5] and their corresponding feature vectors. This is the most common approach for shot representation when further processing is required. However, in the herein approach to compute the feature vector of a key-frame we consider not only this frame itself, but also the feature vectors of its neighboring frames. The reason behind such a representation is that the extracted keyframe may not describe sufficiently the concepts of the video shot due to incorrect keypoints detection and description. In other words, by using the neighborhood of each keyframe, we aim to provide a more reliable representation with respect to the probability that certain concepts exist in a video shot. In this way, if a concept c_i exists in most of the frames of the neighborhood of a keyframe kf, then with high probability it exists on the corresponding keyframe, thus it is correctly represented in its semantic feature vectors $v_{374}(kf)$ and $v_{81}(kf)$.

More specifically, given a keyframe kf_i, we choose a set of frames in the neighborhood of kf_i as follows:

$$N_{kf_i} = \{\ldots, f_{i-3d}, f_{i-2d}, f_{i-d}, kf_i, f_{i+d}, f_{i+2d}, f_{i+3d}, \ldots\}, \tag{2}$$

where d is the distance between two frames.

Given the neighborhood of a shot's keyframe and their corresponding semantic features, we define the following shot representations:

- **Representation $\mathbf{SR^1}$**: For each keyframe kf_i we compute a feature vector as the mean of the semantic features of its neighborhood.

$$SR^1_{kf_i} = \frac{\sum_{j \in N_{kf_i}} v_j}{N_{kf_i}}, \tag{3}$$

  where $v_j, j = 1 \in N_{kf_i}$ is the semantic feature vector of j-th frame in neighborhood N_{kf_i} of keyframe kf_i.
- **Representation $\mathbf{SR^2}$**: Each keyframe is represented by its corresponding semantic feature vector. This is the most common representation in video processing.

$$SR^2_{kf_i} = v_{kf_i}. \tag{4}$$

Summarizing, given a video shot $S = \{kf_1, \ldots, kf_N\}$, with N keyframes, the shot is finally represented by the following feature vectors:

$$SR^r_S = \{SR^r_{kf_i}, i = 1, \ldots, N\}, \tag{5}$$

where r=1, 2.

3 Rushes segmentation

To find groups of repetitive and similar shots we compare successive pair of shots. If two shots are found different, then at the second shot starts a new group of similar shots. Thus, given a series of M videos shots $V = \{S_1, S_2, \ldots, S_M\}, i = 1, \ldots, M$, we seek to find groups of similar shots, or segment the video shot sequence in segments of similar shots.

Suppose we are given two shots i, j and the semantic feature vectors of their corresponding sets of keyframes (or their neighborhood) $S_i = \{SR^i_1, SR^i_2, \ldots, SR^i_{N_i}\}$ and $S_j = \{SR^j_1, SR^j_2, \ldots, SR^j_{N_j}\}$, respectively. N_i and N_j the number of frames that represent shots i, j, respectively. SR can be any of the two representations given from Eq. 3 and Eq. 4. The distance between these two shots is defined as the minimum distance among all possible pairs of their respective representative semantic feature vectors and is given from the following equation:

$$D(S_i, S_j) = \min_{SR^i_k \in S_i, SR^j_n \in S_j} (dist(SR^i_k, SR^j_n)), \tag{6}$$

where $k = 1, \ldots, N_i$, $n = 1, \ldots, N_j$ and $dist$ is the Euclidean distance:

$$dist(x, y) = \sqrt{\sum_h (x_h - y_h)^2}. \tag{7}$$

If distance D is over a predefined threshold t_d, a scene boundary is detected.

3.1 Face and Body Detection

In order to improve the performance of our method, we employ the well-known Viola & Jones algorithm [13] to detect faces and upper-body region, which is defined as the head and shoulders area. We detect faces and upper body regions on each keyframe of every video shot and its corresponding neighborhood. We expect to eliminate false detections, detected by our method, between shots that have similar faces or/and bodies. Face and body detection are performed only in case where scene boundary is detected from our method. Then, after extracting faces and bodies, we calculate the histograms of the detected regions in every frame containing the face/body. The distance between two shots with respect to face/ body histograms is defined as the minimum distance between all possible pairs of the face/body histograms of their respective representative frames. If this distance is below a predefined threshold, these shots are regarded as similar, thus scene boundary is removed and performance is expected to be improved.

4 Experiments

In this Section, we present the video dataset and the performance metrics that have been used in our experiments.

4.1 Datasets and Performance Metrics

We have tested our method on TRECVID 2008 Test Data which was available on the Rushes Summarization task of TRECVID 2008 [8]. The performance of our method was tested on 10 videos. To evaluate the performance of the proposed algorithm and the algorithms under comparison, we have used $F1$ metric provided from the following equation:

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{8}$$

where P and R are Precision and Recall, respectively, and are computed form the following equations:

$$P = \frac{\#\text{correct detections}}{\#\text{correct detections} + \#\text{false detections}}, \tag{9}$$

$$R = \frac{\#\text{correct detections}}{\#\text{correct detections} + \#\text{missed detections}}. \tag{10}$$

4.2 Experimental Results

In Table 1, we present performance results of our method for both shot representations, R^1 and R^2. Four different experiments are presented. In the first two experiments presented as "VIREO - 374" and "VIREO - WEB81", semantic feature vectors $v_{374}(f)$ and $v_{81}(f)$ are employed to represent shot frames. In the third experiment, presented

Table 1: Performance results of the proposed method.

Method	Step (d) / Neighborhood (N)		$F1$ (in%)				
		0 / 0	1 / 3	1 / 5	1 / 7	3 / 7	5 / 7
VIREO - 374	SR^1	-	82.75	83.76	83.25	85.80	**87.43**
	SR^2	80.11	-	-	-	-	-
VIREO - WEB81	SR^1	-	84.38	84.11	**87.14**	86.21	86.81
	SR^2	79.92	-	-	-	-	-
CONCATENATION	SR^1	-	87.79	88.15	87.09	87.81	**85.59**
	SR^2	83.72	-	-	-	-	-
COMBINATION	SR^1	-	89.69	88.78	87.00	87.29	**92.99**
	SR^2	84.89	-	-	-	-	-

as "Concatenation", semantic feature vectors $v_{374}(f)$ and $v_{81}(f)$ are concatenated to form semantic feature vector $v_{con}(f)$ to represent shot frames. Finally, in the fourth experiment, presented as "Combination", the distance between two shots S_i, S_j is computed as the weighted average of the distances computed when semantic feature vectors $v_{374}(f)$ and $v_{81}(f)$ are employed to represent shot frames. The new weighted distance D_c is given from the following equation:

$$D_c(S_i, S_j) = \alpha D_{374}(S_i, S_j) + (1 - \alpha) D_{81}(S_i, S_j), \tag{11}$$

where $0 \leq \alpha \leq 1$, D_{374} and D_{81} are the distances computed from Eq. 6, when semantic feature vectors $v_{374}(f)$ and $v_{81}(f)$ are employed to represent shot frames, respectively. It is obvious that $\alpha = 0$ corresponds to experiment "VIREO - WEB81", whereas $\alpha = 1$ corresponds to experiment "VIREO - 374". When SR^1 representation is employed, the neighborhood of a keyframe N (Eq. 2), takes value 3, 5 or 7 when distance $d = 1$ and 7 when $d = 3, 5$. When SR^2 representation is employed only the keyframe is used, thus d, N are equal to zero. Threshold t_d is different for each experiment but same for all videos in the same experiment.

It is obvious that when a shot is represented by the neighborhoods of the keyframes the performance is better that using only keyframes. Moreover, better performance is achieved when when the size of the neighborhood and distance from the keyframes increase. In Table 2, we present performance result after refining segmentation using face/body detection. We use thresholds $t_f = 0.01$ for face and $t_b = 0.02$ for body to define whether two shots are similar w.r.t to face/body detection. It is clear that the proposed refinement of scene boundaries increases the performance of the proposed method. In Table 3, we present performance results using only face and/or body detection to compare shots. Performance is poor due to absence of faces/bodies in many shots. Thus, face/body detection results can only serve as a refinement feature.

Table 2: Performance results of the proposed method after refinement with face/body detection.

		$F1$ (in%)					
	Step (d)	0	1	1	1	3	5
Method	Neighborhood (N)	0	3	5	7	7	7
VIREO - 374	SR^1	-	83.41	83.76	83.25	85.80	**87.43**
	SR^2	80.68	-	-	-	-	-
VIREO - WEB81	SR^1	-	84.38	84.11	87.14	86.21	**87.77**
	SR^2	81.02	-	-	-	-	-
CONCATENATION	SR^1	-	87.79	88.15	87.09	87.81	**88.60**
	SR^2	84.49	-	-	-	-	-
COMBINATION	SR^1	-	89.69	88.78	90.18	87.29	**93.30**
	SR^2	85.60	-	-	-	-	-

Table 3: Performance results using only face and body detection.

		$F1$ (in%)					
	Step (d)	0	1	1	1	3	5
Method	Neighborhood (N)	0	3	5	7	7	7
Face	SR^1	-	54.00	55.00	55.00	55.00	55.00
	SR^2	52.00	-	-	-	-	-
Body	SR^1	-	54.00	54.00	54.00	54.00	50.00
	SR^2	52.00	-	-	-	-	-
Face & Body	SR^1	-	54.00	53.00	54.00	54.00	50.00
	SR^2	53.00	-	-	-	-	-

4.3 Comparison

In this Section we present comparative results of our method using HSV color histograms instead of semantic feature vectors, in order to show the superiority of semantic features in rushes segmentation. In Table 4, we present performance results using HSV color histograms. We use 8 bins for hue and 4 bins for each of saturation and value, resulting into a 128 ($8\times4\times4$) dimension feature vector. The main disadvantage of these descriptors is that they only represent the color distribution of an object ignoring its shape and texture. Color histograms are also sensitive to noise, such as lighting changes. It is clear that HSV color histograms have difficulty in capturing the changes between groups of similar shots.

Moreover, since SIFT descriptors are employed to compute semantic features, we provide performance results using the number of matching descriptors between shots as a shot similarity metric. For each keyframe and its neighborhood (Eq. 2) we extract SIFT descriptors. The number of matching descriptors [10] between two shots serves as the shot similarity metric. More specifically, the maximum number of matching descriptors between all possible pairs of their respective representative frames is the final similarity value. Two shots belong to different groups, thus they are not similar, if their respective similarity value is below a threshold set to 0.04 in our experiments.

Table 4: Performance results using HSV color histograms and SIFT descriptors.

		$F1$ (in%)					
	Step (d)	0	1	1	1	3	5
Method	Neighborhood (N)	0	3	5	7	7	7
HSV	SR^1	-	79.06	79.04	79.38	78.71	78.74
	SR^2	79.46	-	-	-	-	-
SIFT	SR^1	-	78.76	78.18	76.05	73.76	73.46
	SR^2	77.28	-	-	-	-	-
SIFT - IMPROVED	SR^1	-	82.84	83.54	83.40	83.13	81.53
	SR^2	80.14	-	-	-	-	-

Fig. 1: Subset of the matching descriptors between two "similar" frames, before and after imposing spatial constraint.

In Table 4, we present performance results using SIFT descriptors. It can be observed that matching SIFT descriptors does not provide good results. A main reason for this is that the same actors/object/setting may appear in two shots that belong to different groups/scenes. For this reason a spatial constraint on matching descriptors is employed. Given the coordinates (x, y) of a descriptor, we seek to find a matching descriptor with coordinates in area $(x \pm s, y \pm s)$, where s is set to 20 in our experiments. In Fig. 1 we present a subset of the matching descriptors between two "similar" frames, before and after imposing spatial constraint. Performance results are presented in Table 4. It is clear that performance is improved. However, semantic features vectors still provide the best performance.

In another experiment we carried out, we reduce the number of concepts employed to form semantic feature vectors. More specifically, for each concept we compute the mean probability of occurrence across all videos. If this mean value is below a prede-

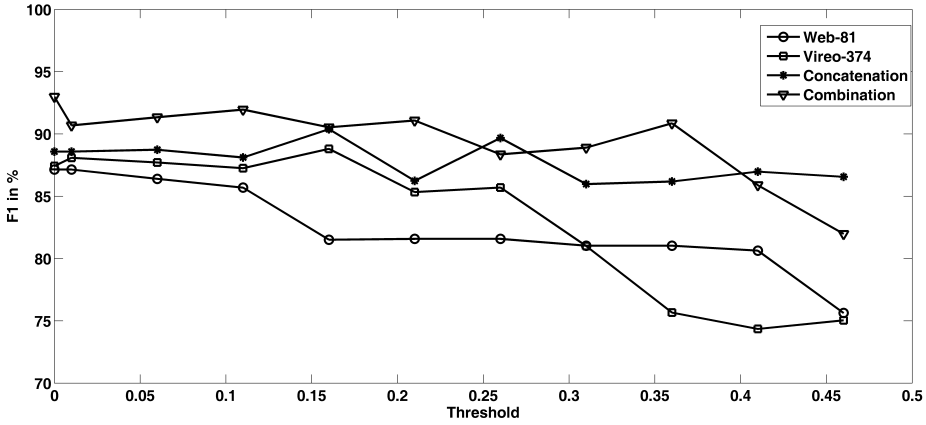

Fig. 2: Performance results of our method using d=5 and N=7 on a subset of concepts.

fined threshold t_c, the corresponding concept is not taken into consideration. In Fig. 2 and Fig. 3, we present performance results and the corresponding number of concepts with respect to threshold t_c, respectively. It can be observed that even with a subset of concepts, our method yields very good results.

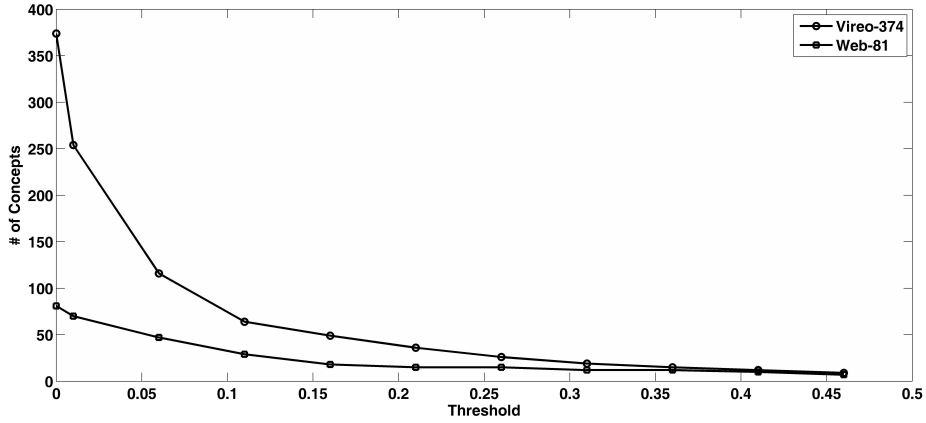

Fig. 3: Number of concepts w.r.t threshold t_c.

5 Conclusions

In this paper a rushes video segmentation method is proposed. Contrary to existing approaches, shot frames are represented by semantic feature vectors extracted using

common semantic concept detectors. Moreover, each keyframe is represented by the mean of the semantic features vectors of its neighborhood, defined as the frames that fall inside a window centered at the keyframe. Next, successive shots are compared to find boundaries between groups of similar shots. Face and body detection is employed to improve the performance of our method by eliminating false boundaries detected between shots with similar faces/bodies. Numerical experiments show that the proposed method can efficient segment rushes videos in groups of similar shots.

References

1. Dumont, E., Merialdo, B.: Rushes video parsing using video sequence alignment. In: Seventh International Workshop on Content-Based Multimedia Indexing, CBMI '09. (2009) 44–49
2. Ren, J., Jiang, J.: Hierarchical modeling and adaptive clustering for real-time summarization of rush videos. IEEE Transactions on Multimedia **11**(5) (2009) 906–917
3. Rossi, E., Benini, S., Leonardi, R., Mansencal, B., Benois-Pineau, J.: Clustering of scene repeats for essential rushes preview. In: 10th Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS '09. (2009) 234–237
4. Chasanis, V., Likas, A., Galatsanos, N.: Video rushes summarization using spectral clustering and sequence alignment. In: TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop, Vancouver, British Columbia, Canada (2008) 75–79
5. Chasanis, V., Likas, A., Galatsanos, N.: Scene detection in videos using shot clustering and sequence alignment. IEEE Transactions on Multimedia **11**(1) (January 2009) 89–100
6. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval. CIVR '07 (2007) 494–501
7. Zhu, S., Wang, G., Ngo, C.W., Jiang, Y.G.: On the sampling of web images for learning visual concept classifiers. In: Proceeding of the ACM International Conference on Image and Video Retrieval (CIVR 2010). (2010) 50–57
8. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and trecvid. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval. (2006) 321–330
9. Kennedy, L., Hauptmann, A.: Lscom lexicon definitions and annotations version 1.0, dto challenge workshop on large scale concept ontology for multimedia. Technical report, Columbia University (March 2006)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
11. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In: Proceedings of the international workshop on Workshop on multimedia information retrieval. MIR '07 (2007) 197–206
12. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27 Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001. Volume 1. (2001) 511–518