Movie Segmentation into Scenes and Chapters Using Locally Weighted Bag of Visual Words

Vasileios Chasanis
Department of Computer Science
University of Ioannina
GR 45110
vchasani@cs.uoi.gr

Argyris Kalogeratos
Department of Computer Science
University of Ioannina
GR 45110
akaloger@cs.uoi.gr

Aristidis Likas
Department of Computer Science
University of Ioannina
GR 45110
arly@cs.uoi.gr

ABSTRACT

Movies segmentation into semantically correlated units is a quite tedious task due to "semantic gap". Low-level features do not provide useful information about the semantical correlation between shots and usually fail to detect scenes with constantly dynamic content. In the method we propose herein, local invariant descriptors are used to represent the key-frames of video shots and a visual vocabulary is created from these descriptors resulting to a visual words histogram representation (bag of visual words) for each shot. A key aspect of our method is that, based on an idea from text segmentation, the histograms of visual words corresponding to each shot are further smoothed temporally by taking into account the histograms of neighboring shots. In this way, valuable contextual information is preserved. The final scene and chapter boundaries are determined at the local maxima of the difference of successive smoothed histograms for low and high values of the smoothing parameter respectively. Numerical experiments indicate that our method provides high detection rates while preserving a good tradeoff between recall and precision.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing, Abstracting methods

General Terms

Experimentation

Keywords

Scene detection, Chapter Detection, SIFT descriptors, CCH descriptors, Lowbow

1. INTRODUCTION

Indexing of video data is very important nowadays, due to the increasing amount of video data produced every year. More specifically, the efficient segmentation of movies into scenes and larger semantically correlated units could provide easy and quick access to a huge volume of video data. The first level of video segmentation is the *shot* and is defined as an unbroken sequence of frames recorded from the same camera. Each shot is represented by unique frames (*key-frames*) that capture its content. The efficient representation of a shot by its key-frames is very important, because the more information we have about a shot, the better we can associate it with the preceding or following shots.

Proceeding further towards the goal of video indexing and retrieval requires the grouping of shots into scenes. Usually, a *scene* refers to a group of shots that take place in the same physical location (e.g. a dialogue detection in a room) or a group of shots that describe an action or event (e.g a car chase by police cars). A more compact representation/segmentation of a video is the merging of scenes into logical story units. The latter, corresponds to the DVD chapters describing the different sub-themes of a movie.

There are two major problems concerning movie segmentation into scenes and chapters. The first problem concerns the content of the video. In dialogue scenes where the content of video does not change dramatically (change between cameras recording actors speaking), low-level features such as color histograms are quite efficient in detecting the scene boundaries. However, there are scenes where content changes constantly. For example, in a scene describing a car chase, there are different shots taken in different places during the course of the car. Thus, the color distribution of the shots constantly changes making the use of color histograms inefficient. On the other hand, there are some objects or distinctive points that are repeated in consecutive shots during the progress of such an event. Local invariant descriptors provide sufficient description of these interest points and their possible transformations (rotation, scale). These descriptors can be grouped into a large number of clusters. Each cluster is treated as a *visual word* and represents a specific local pattern shared by all the descriptors in the cluster. By mapping the descriptors into visual words we can adopt the bag of words representation, that in the field of image and video processing is known as *bag of visual words*. Thus, each video shot can be represented as a vector containing the weighted count of each visual word in the shot. This sematic representation of each shot with a set of visual words helps to "semantically" correlate two shots and detect possible scene boundaries.

The second problem in movie segmentation is the detec-

CIVR '09 July 8-10, 2009 Santorini, GR.
Copyright 2009 ACM 978-1-60558-480-5/09/07 ...$5.00.

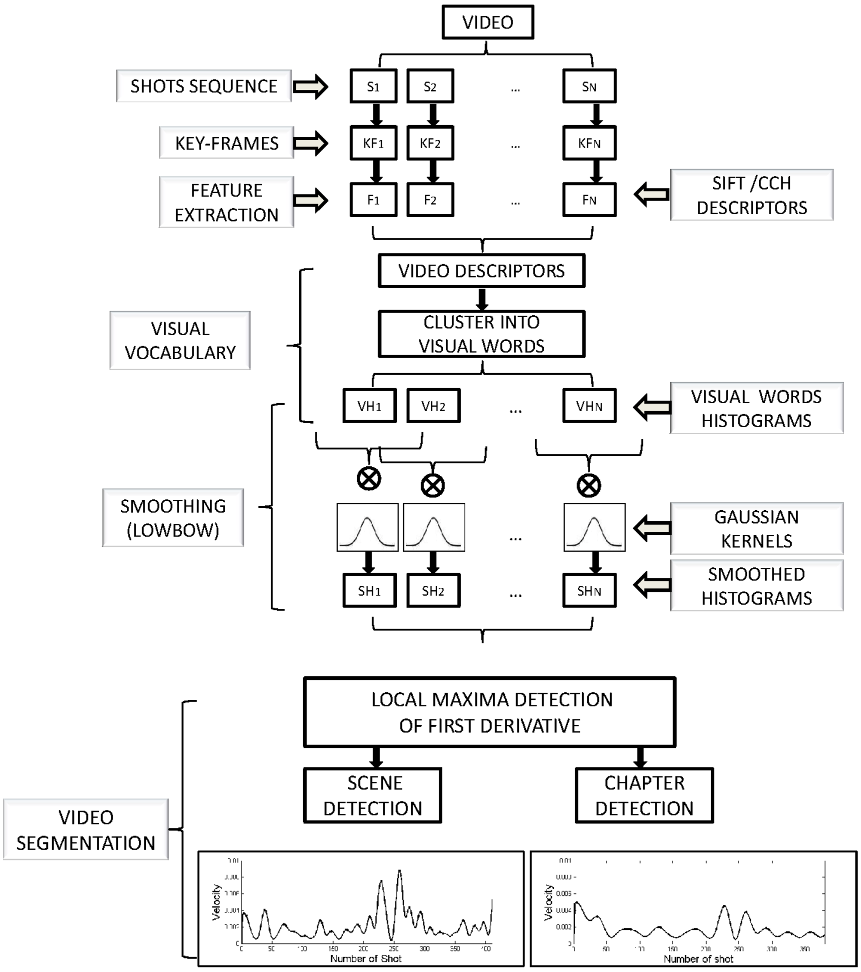

Figure 1: Main steps of our method.

tion of chapters (logical story units). Since a chapter is a group of scenes describing a sub-theme of the movie, it is expected that the color distributions of the corresponding shots will fail to describe the connectivity between them. For example, consider a chapter that comprises of the two following scenes. The first scene describes a thief stealing a car followed by the second scene that describes the chase of the stolen car from the police. Considering that these two events take place in different places, the color distribution of the shots will constantly change. On the other hand, features describing the stolen and the police cars could provide useful information about the semantical connection of these two scenes and their corresponding shots.

Several approaches have been proposed for the scene segmentation problem. In [11], a scene transition graph is constructed to represent the connectivity between the video shots. Then, this graph is divided into connected subgraphs that represent the final scenes. A similar approach is proposed in [9], where the authors transform the scene segmentation problem into a graph partitioning problem. A shot similarity graph is constructed, where each node represents a shot and the edges between shots depict their similarity based on color and motion information. Then, normalized cuts [10] is applied to partition the graph. A different approach is presented in [8] where a two-pass algorithm is proposed. In the first pass shots are clustered by computing backward shot coherence, a similarity measure of a given shot with respect to the previously seen shots, while in the second pass over-segmented scenes are merged based on the computation of motion content in scenes. In [12], the authors propose the use of Markov chain Monte Carlo to determine the scene boundaries. Two processes, diffusions and jumps, are used to update the scene boundaries that are initialized at random positions. Diffusions are the operations that adjust the boundaries between adjacent scenes, while jump operations merge or split existing scenes.

Segmentation of a movie into a more compact representation, such as chapters, has not received much attention yet. In [9], the authors compare the scene segmentation results of their algorithms with the chapters provided in the DVD compilations of known movies. Considering that chapters are more compact representations, there many false detections resulting into a very low precision, thus making inefficient the specific algorithm for the problem of chapter detection.

In the method we propose herein, each video is first segmented into shots. To represent the content of each shot, key-frames are extracted using an improved version of spectral clustering. Then, local invariant descriptors are extracted from all key-frames of the shot. The descriptors of all shots are clustered into a predefined number of visual words and a visual words histogram is constructed for each shot. The histograms of visual words corresponding to each shot are further smoothed temporally by taking into account the histograms of neighboring shots. In this way, valuable contextual information is preserved. The final scene and chapter boundaries are determined at the local maxima of the difference of successive smoothed histograms for low and high values of the smoothing parameter respectively. Thus, by adjusting the smoothing parameter of the gaussian kernel we can segment each video into different levels, scenes or chapters. In *Fig.* 1, we summarize the main steps of our approach.

The rest of the paper is organized as follows: In section 2, the key-frame extraction method, the feature extraction method and the construction of visual vocabulary are described. In section 3, the proposed scene and chapter detection algorithm is presented that is based on temporally smoothed shot histograms. In section 4, we present numerical experiments and compare our method with the method proposed in [9]. Finally in section 5, we conclude our work and provide suggestions for further study.

2. VIDEO REPRESENTATION

In order to proceed with video segmentation into high-level units, the volume of video data to be processed must be reduced. It is required to start with the video segmentation into shots and continue with the efficient representation of shots. In this way, a video comprised of thousands of frames can be efficiently represented using only several hundreds of frames. In our approach, each video is manually segmented into shots and each shot is represented with key-frames extracted by applying the algorithm explained in the following section.

2.1 Key-Frame Extraction

Each shot is represented by distinctive frames, called key-frames, that capture the whole content of each shot. In our approach we use the method proposed in [1]. Each frame is represented by a 3D HSV normalized histogram with 8 bins for hue and 4 bins for each of saturation and value, resulting to 8 × 4 × 4 bins. The frames of the shot are clustered into groups using an improved version of the typical spectral clustering method [7] that uses the fast global k-means algorithm [5] in the clustering stage after the eigenvector computation. Then the medoid of each group, defined as the frame of the group whose average similarity to all other frames of this group is maximal, is characterized as a key-frame.

2.2 Feature Extraction

As already mentioned, color histograms fail to describe connectivity between shots in cases where the visual content rapidly changes. However, the semantic content remains the same because objects or interest points are repeated during the shots of the scenes. A well-known method to describe objects in images are the invariant local descriptors. These descriptors apart from describing an object, also describe its possible transformations (rotation, scale). In our approach we examine two kinds of descriptors that have been proposed in bibliography. The first ones are the SIFT descriptors proposed in [6] and the CCH descriptors proposed in [3].

2.2.1 SIFT Descriptors

In [6], the scale-invariant feature transforms have been proposed that transform image data into scale-invariant coordinates relative to local features. These features are invariant to image scale and rotation. The method described in [6], consists of four major stages: (1) scale-space peak selection; (2) keypoint localization; (3) orientation assignment; (4) keypoint descriptor. In the first stage, the image is scanned over scale and location to detect features (or interest) points. A Gaussian pyramid is constructed and local peaks (keypoints) in a series of difference-of-Gaussian (DoG) images are detected. In the second stage unstable keypoints are eliminated. In the third stage, the dominant orientations for each key-point based on its local image patch are identified. Finally, in the fourth stage, a local image descriptor is built for each keypoint, based upon the image gradient in its local neighborhood. The standard keypoint descriptor used by SIFT is created by sampling the magnitudes and orientations of the image gradient in the patch around the keypoint, and building smoothed orientation histograms to capture the important aspects of the patch. A 4 × 4 array of histograms, each with 8 orientation bins, captures the rough spatial structure of the patch. This 128-element vector (4 × 4 × 8) is then normalized to unit length and thresholded to remove elements with small values.

2.2.2 CCH Descriptors

A similar local invariant descriptor, called contrast context histogram (CCH) is proposed in [3]. It represents the contrast distributions of a local region around an interest point and serves as a local descriptor for this region. Given an image I, gaussian kernels are applied to smooth this image. Then, a multi-scale Laplacian pyramid is constructed and salient points are extracted by detecting Harris corners. Around each salient point p_c a region R is defined and the contrast of a point in this area is given from the following equation:

$$C(p) = I(p) - I(p_c). \qquad (1)$$

Region R is defined in a quantized log-polar coordinate system (r, θ), where $r_i = 0, \ldots, r,\ r = \lfloor log(\sqrt{2n^2}) \rfloor$ and $\theta_j = \frac{2\pi}{l} m,\ m = 0, \ldots, l-1$. Parameters r, l define the distance and orientation quantization respectively. For each sub-region $R_{ij} = (r_i \theta_j)$, a positive and a negative bin of the contrast values are computed. More specifically, given a salient point p_c and a sub-region R_{ij}, the positive and the negative histogram bins are defined from equations (2) and (3) respectively.

$$H_{R_{ij}^{+}}(p_c) = \frac{\sum\{C(p) | p \in R_{ij}\ and\ C(p) \geq 0\}}{\# R_{ij}^{+}}, \qquad (2)$$

$$H_{R_{ij}^{-}}(p_c) = \frac{\sum\{C(p) | p \in R_{ij}\ and\ C(p) < 0\}}{\# R_{ij}^{-}}, \qquad (3)$$

where $\# R_{ij}{}^{+}$ and $\# R_{ij}{}^{-}$ define the number of positive and negative positive contrast values in R_{ij}. By composing the contrast histograms of all the sub-regions into a single vector, the CCH descriptor of p_c with respect to its local region R is defined as follows:

$$CCHp_c = (H_{R00}^{+}, H_{R00}^{-}, \dots, H_{Rrl}^{+}, H_{Rrl}^{-}). \tag{4}$$

In our approach we used $r = 3$ and $l = 8$ as proposed in [3] resulting to $2 \times 4 \times 8 = 64$ dimensions for each CCH descriptor.

2.3 Visual Words

For each shot a different number of descriptors is computed. These descriptors describe certain objects or interest points in the shots. Suppose we are given a shot s_i and its corresponding set of n key-frames $KF = \{kf_1, \dots, kf_n\}$. For each key-frame $kf_i,\ i = 1, \dots, n$, a set for descriptors D_{kf_i} is extracted ($SIFT$ or CCH) using the algorithms presented in [6] and [3], respectively. Then, all the sets of descriptors are concatenated to represent the whole shot

$$D_{s_i} = D_{kf_1} \bigcup \dots \bigcup D_{kf_n}. \tag{5}$$

2.4 Visual Vocabulary

To extract *visual words* from the descriptors, the set of descriptors of N video shots $D_S = D_{s_1} \bigcup D_{s_2} \bigcup \dots \bigcup D_{s_N}$ is clustered into k groups $\{C_1, C_2, \dots, C_k\}$, where k denotes the total visual words vocabulary size. To construct the visual words histogram (bag of visual words) for each shot, its corresponding set of descriptors D_{s_i} is mapped into the k visual words resulting into a vector containing the weighted count of each visual word in the shot. Thus, given that a shot s_i has D descriptors $d_{p_1}, \dots, d_{p_D}$, the visual words histogram for this shot is defined as:

$$VH_i(l) = \frac{\{d_{p_j} \in C_l,\ j = 1, \dots, D\}}{D},\ l = 1, \dots, k. \tag{6}$$

3. SCENE AND CHAPTER DETECTION

So far, each video shot is represented by a visual words histogram that corresponds to the probability that a specific *visual word* of the video is included in the specific shot. Next, the similarity between shots must be defined in order to detect the scene and chapter boundaries.

3.1 Similarities Between Video and Text Documents

Video and text documents exhibit many analogies. Videos can be segmented into shots, scenes and logical story units (DVD chapters). In a similar way, text documents can be segmented into words, paragraphs and logical story units (book chapters). In [4], the Locally Weighted Bag of Words (Lowbow) framework has been proposed for document representation. The main idea of this algorithm is to use a local smoothing kernel to smooth the original word sequence temporally. In other words, instead of using the typical bag of words representation, the presence of a word at a certain location in the document is borrowed to a neighboring location but discount its contribution depending on the temporal distance between the two locations. This representation captures sequential content at a certain resolution determined by a given local smoothing operator.

A similar approach can be adopted in video documents. Video shots can be regarded as the words of a document that compose a paragraph (scene) that further compose a book chapter (DVD chapter) that describes a specific theme. Thus, in each shot a visual word histogram representation can be adopted and computed as described in Section 2.

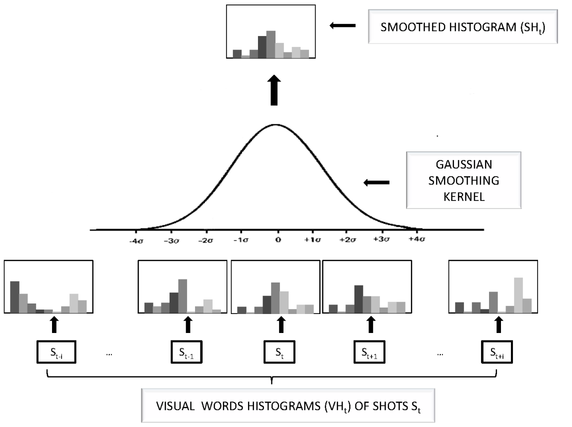
SMOOTHED HISTOGRAM (SH$_t$)
GAUSSIAN SMOOTHING KERNEL
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
S$_{t-i}$
...
S$_{t-1}$
S$_t$
S$_{t+1}$
...
S$_{t+i}$
VISUAL WORDS HISTOGRAMS (VH$_t$) OF SHOTS S$_t$

Figure 2: Temporal smoothing of visual words histograms representing the video shots, using a gaussian smoothing kernel.

3.2 Scene Segmentation

In a similar way to Lowbow framework described in [4], a local smoothing kernel can be used to smooth temporally the visual words histogram of a shot with respect to the histograms of neighboring shots. The smoothed histogram SH_t of a visual words histogram VH_t of a shot S_t (where t denotes the time index of the shot) is given from the following equation:

$$SH_t = \sum_{n=-\infty}^{\infty} VH_{t-n} \cdot K_\sigma(t-n), \tag{7}$$

where K_σ is the gaussian kernel with zero mean and standard deviation σ. In *Fig.* 2 a visual representation of the smoothing process is given. First, the visual words histogram (VH_t) for each shot are computed (bottom level). Then, the visual words histogram of shot S_t is smoothed temporally with the neighbor visual words histograms using a gaussian kernel, resulting to the smoothed histogram (SH_t) of the shot (upper level). The number of neighboring histograms that contribute to smoothing is defined by the value of the smoothing parameter σ. By adjusting the value of σ we can preserve contextual information in different time scales. A low value of σ can preserve contextual information concerning scenes, whereas a higher value of σ can preserve contextual information concerning chapters.

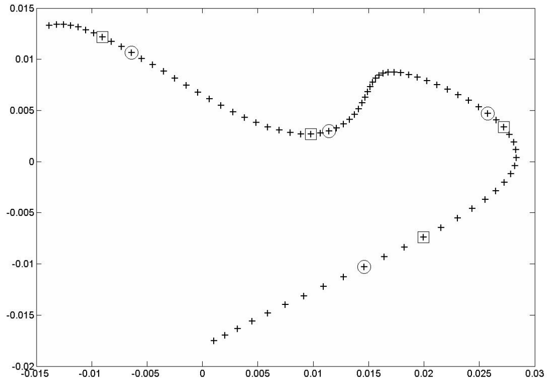

Figure 3: 2D embedding (using PCA) of the TSVWH curve representing a video shot sequence.

Our model associates each shot $S_t,\ t = 1, \ldots, N$ with a smoothed histogram $SH_t,\ t = 1, \ldots, N$ and a point in the multinomial simplex $\mathbb{P}_{k-1}$, where k is the vocabulary size. The multinomial simplex $\mathbb{P}_{k-1}$ for k > 1 is the k-dimensional subset of $\mathbb{R}^k$ of all histograms over k objects

$$\mathbb{P}_{k-1} = \{\theta \in \mathbb{R}^k : \forall i\ \theta_i \geq 0,\ \sum_{j=1}^{k} \theta_j = 1\}. \tag{8}$$

The sequence $SH_t,\ t = 1, \ldots, N$ of smoothed histograms represents the video shot sequence with a curve in $\mathbb{P}_{k-1}$ called Temporally Smoothed Visual Words Histograms or TSVWH curve. *Fig.* 3, illustrates an example of a video shot sequence, whose TSVWH curve representation is projected from $\mathbb{P}_{k-1}$ to $\mathbb{P}_2$ using Principal Component Analysis (PCA).

The boundaries between different video segments separate video parts containing different visual words distributions. In the context of the TSVWH curve produced by the smoothed histograms this would correspond to sudden shifts in the curve location. Due to the continuity of TSVWH curve, such sudden shifts may be discovered by considering the local maxima of the Euclidean distance between successive smoothed histograms:

$$V_i = \sqrt{\sum_{h=1}^{k} (SH_i(h) - SH_{i+1}(h))^2},\ i = 1, \ldots, N-1, \tag{9}$$

where k denotes the number of visual words (histogram bins). In *Fig.* 4 and 5, we present the values of the difference of smoothed histograms using smoothing parameter $\sigma = 8$ (for scene detection) and $\sigma = 16$ (for chapter detection) respectively. In both cases, circles correspond to detected boundaries and stars correspond to true boundaries.

4. EXPERIMENTS

In this section we present numerical experiments for the scene and chapter detection problem, and we also compare our method to an existing approach [9].

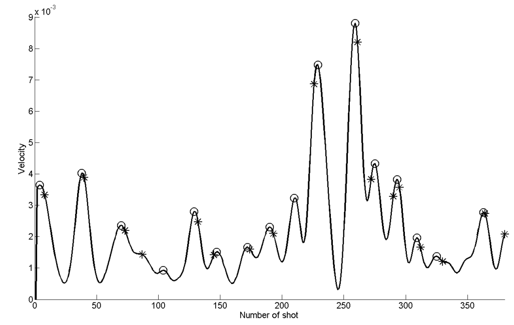

Figure 4: Difference values of the smoothed histograms using $\sigma = 8$ (scene detection).

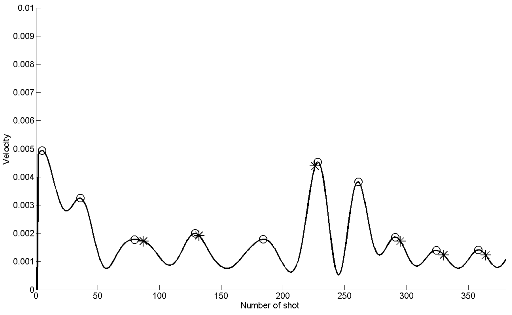

Figure 5: Difference values of the smoothed histograms using $\sigma = 16$ (chapter detection).

4.1 Data and Performance Criteria

To evaluate the performance of our detection algorithm we use three movies that belong to different genres. The characteristics of these movies are shown in Table 1.

To evaluate the performance of our method we used the following commonly used criteria [2]:

$$\text{Recall} = \frac{N_c}{N_c+N_m}, \quad \text{Precision} = \frac{N_c}{N_c+N_f}, \qquad F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall+Precision}}, \tag{10}$$

where N_c stands for the number of correctly detected scene boundaries (true positive), N_m for the number of missed ones (false positive) and N_f the number of false detections (false negative). Two human observers identified the scene boundaries and the ground truth was defined as the intersection of the two opinions. The boundaries of the chapters were extracted from the menu of the DVD compilations of the corresponding movies.

4.2 Scene detection

In the experiments we carried out, we have tested the performance of our algorithm for different number of visual words, adjusted by setting different values for parameter k (see section 2.4). More specifically we have tested a visual vocabulary of 10, 20, 50, 100, 200, 500 visual words. In Tables 2 and 3, we present the recall, precision and F_1 values

Table 1: Movies characteristics.

Video	A Beautiful Mind (M1)	Sex and the City(M2)	Gone in 60 seconds(M3)
Duration(min)	36	70	80
Shots	421	1217	1788
Scenes	18	45	74
DVD Chapters	7	19	23
Genre	Biography | Drama	Comedy | Romance	Action | Crime | Thriller

Table 2: Performance of our movie scene segmentation method.

	Movie M1			Movie M2			Movie M3		
Method	Recall	Precision	F_1	Recall	Precision	F_1	Recall	Precision	F_1
sift10	83.33	83.33	83.33	86.67	83.33	78.79	77.03	71.25	74.03
sift20	83.33	83.33	83.33	77.78	83.33	72.92	81.08	68.97	74.53
sift50	88.89	84.21	86.49	82.22	84.21	75.51	81.08	70.59	75.47
sift100	88.89	88.89	88.89	82.22	88.89	78.72	82.43	69.32	75.31
sift200	83.33	88.24	85.71	80.00	88.24	78.26	87.84	73.03	79.75
sift500	88.89	88.89	88.89	91.11	88.89	87.23	82.43	72.62	77.22
cch10	66.67	75.00	70.59	68.89	58.49	63.27	70.62	59.77	64.60
cch20	72.22	76.47	74.29	80.00	67.92	73.47	71.03	62.35	66.67
cch50	77.78	82.35	80.00	80.00	72.00	75.79	77.03	66.28	71.25
cch100	83.33	78.95	81.08	80.00	72.00	75.79	77.38	66.28	71.25
cch200	77.78	77.78	77.78	80.00	73.47	76.60	78.03	67.44	72.50
cch500	83.33	88.24	85.71	80.00	73.47	76.60	77.62	69.51	73.08
[9]	83.33	72.22	69.77	73.33	53.23	61.68	74.32	55.56	63.58

Table 3: Performance of our movie chapter segmentation method.

	Movie M1			Movie M2			Movie M3		
Method	Recall	Precision	F_1	Recall	Precision	F_1	Recall	Precision	F_1
sift10	100.00	41.67	52.63	73.68	53.85	62.22	73.91	39.54	51.52
sift20	100.00	53.85	70.00	89.47	60.71	72.34	73.91	41.46	53.13
sift50	100.00	58.33	73.68	94.74	60.00	73.47	73.91	44.74	55.74
sift100	100.00	58.33	73.68	84.21	61.54	71.11	73.91	43.59	54.84
sift200	100.00	58.33	73.68	94.74	64.29	76.60	78.26	46.15	58.07
sift500	100.00	58.33	77.78	94.74	69.23	80.00	78.26	45.00	57.14
cch10	85.71	63.64	70.59	73.68	53.85	62.22	78.26	42.00	55.39
cch20	85.71	60.00	70.59	73.68	53.85	62.22	82.61	44.19	57.58
cch50	85.71	60.00	70.59	73.68	53.85	62.22	82.61	43.18	56.72
cch100	85.71	60.00	70.59	84.21	59.26	69.57	78.26	42.86	55.39
cch200	85.71	60.00	70.59	84.21	59.26	69.57	78.26	42.86	55.39
cch500	85.71	60.00	70.59	84.21	59.26	69.57	82.61	43.18	56.72
[9]	100.00	28.00	43.75	94.74	29.03	44.44	95.65	22.22	36.07

for the three movies of our dataset for the scene and chapter segmentation problem respectively. In all experiments, for scene detection we use $\sigma = 8$ and for chapter detection $\sigma = 16$. It can be observed that our approach achieves high correct detection rate while keeping small the number of false detections. Even for the difficult problem of chapter detection, our algorithm yields very good results. It is worth mentioning that the performance of the detection algorithm improves, as the number of visual words k increases. This is expected, since more visual worlds provide more information about the semantic content of the shots. Thus, the more we know about the semantic correlation between shots, the better we can detect scene and chapter boundaries.

4.3 Comparison

To compare the effectiveness of our approach, we have also implemented the method proposed in [9]. This method computes both color and motion similarity between shots and the final similarity value is weighted by a decreasing function of the temporal distance between shots given by the following equation:

$$w_t(i,j) = e^{-\frac{1}{d}|\frac{m_i-m_j}{\sigma}|^2} , \qquad (11)$$

where m_i and m_j are the time indices of the middle frames of the two shots under consideration and σ the standard deviation of the shots' duration in the entire video. The parameter d plays a critical role in the final number of scenes produced by the algorithm. The final shot similarity matrix defines a weighted undirected graph where each node represents a shot and the edges are the elements of the matrix.

To partition the video into scenes, an iterative application of Normalized cuts method [10] was used that divides the graph into subgraphs. It must be noted that the implementation of the Normalized cuts method in this approach does not require the computation of eigenvectors, because scenes are composed of shots which are time continuous. Thus a cut can be made along the diagonal of the shot similarity matrix. The $Ncut$ algorithm is applied recursively as long as the $Ncut$ value is below some stopping threshold T. The recall, precision and the F_1 values of the experiments of this method are presented in Tables 2 and 3. It is clear that our algorithm provides the best F_1 value for all movies, and in general our method outperforms the other approach. Moreover, concerning the chapter detection problem, our method is by far more efficient, whereas the method in comparison fails to provide a sensible and accurate segmentation since it produces many false negatives.

5. CONCLUSIONS

In this paper a new high-level video segmentation has been proposed based on the temporal smoothing of visual word histograms of video shots. For each video shot a number of key-frames is extracted and local invariant descriptors are computed for each key-frame. All the descriptors are clustered into visual words and for each shot a visual words histogram is computed. Next, to preserve valuable contextual information, by adopting the Lowbow framework proposed for text processing, these histograms of visual words are smoothed temporally by taking into account the histograms of neighboring shots. By adjusting the smoothing parameter of the gaussian kernel we can detect both scene and chapter boundaries of each video, determined at the local maxima of the difference of successive smoothed histograms. The presented experimental results on three movies indicate that the proposed method accurately detects most scene and chapter boundaries, while providing a good trade off between recall and precision. In future work, we will try to extend the use of video representation with temporally smoothed shot histograms to other video based applications, such as video retrieval and surveillance.

6. ACKNOWLEDGMENTS

This research project (PENED) is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development- GSRT (25%).

7. REFERENCES

[1] V. Chasanis, A. Likas, and N. Galatsanos. Efficient video shot summarization using an enhanced spectral clustering approach. In *ICANN '08: Proceedings of the 18th international conference on Artificial Neural Networks, Part I*, pages 847–856, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] A. Del Bimbo. *Visual information retrieval.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[3] C.-R. Huang, C.-S. Chen, and P.-C. Chung. Contrast context histogram - a discriminating local descriptor for image matching. *Pattern Recognition, International Conference on*, 4:53–56, 2006.

[4] G. Lebanon, Y. Mao, and J. Dillon. The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, 8:2405–2441, 2007.

[5] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36:451–461, 2003.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[8] Z. Rasheed and M. Shah. Scene detection in hollywood movies and tv shows. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2:II–343–8 vol.2, June 2003.

[9] Z. Rasheed and M. Shah. Detection and representation of scenes in videos. *IEEE Transactions on Multimedia*, 7(6):1097–1105, Dec. 2005.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[11] M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *Comput. Vis. Image Underst.*, 71(1):94–109, 1998.

[12] Y. Zhai and M. Shah. Video scene segmentation using markov chain monte carlo. *IEEE Transactions on Multimedia*, 8(4):686–697, Aug. 2006.