PUBLISHED IN 3RD WORKSHOP ON CONTENT-BASED MULTIMEDIA INDEXING (CBMI)
SEPTEMBER 2003, RENNES, FRANCE.

VIDEO SHOT CLUSTERING USING SPECTRAL METHODS

Jean-Marc Odobez, Daniel Gatica-Perez, and Mael Guillemot

Dalle Molle Institute for Perceptual Intelligence (IDIAP), Martigny, Switzerland

ABSTRACT

The automatic segmentation and structuring of videos present technical challenges due to the large variation of content, spatial layout, and possible lack of storyline. In this paper, we propose a spectral method to group video shots into scenes based on their visual similarity and temporal relations. Spectral methods have been shown to be effective in capturing perceptual organization features. In particular, we investigate the problem of automatic model selection, which is currently an open research issue for spectral methods, and propose measures to assess the validity of a grouping result. The methodology is used to group shots from home videos and soccer games. The results indicate the validity of the proposed approach, both compared to existing techniques as well as to human performance.

1. INTRODUCTION

Segmentation and structuring of videos constitute important functionalities in content-based media analysis [2, 8, 12]. As an exploratory analysis technique, clustering of video entities like shots and scenes allows for unsupervised content organization (possibly at multiple levels), and has a direct application in browsing [12, 8].

Broadly speaking, videos can be characterized according to the specificity of their content and their production models. On one side of the spectrum, home videos are non-produced, and are characterized by unrestricted content and the absence of storyline [2, 3]. They are composed of scenes, each composed of few shots, visually consistent, localized in time, and randomly recorded. On the other extreme, sports videos and news programs are heavily produced, acquired within a specific context, and prior domain knowledge can be employed to extract relevant information [13, 14]. However, there exists structure in both types of content (looser in the first case, much more specific in the second one) that can potentially be discovered by clustering algorithms and used for browsing purposes [2, 8, 12].

Spectral clustering methods [10, 9, 11], which aim at partitioning a graph based on the eigenvectors of its pairwise similarity matrix, have received an increasing interest in the computer vision and machine learning literatures. In practice, these methods have provided some of the best known results for image segmentation [9] and data clustering, but to our knowledge have not been applied to video organization. Furthermore, one key problem in clustering, namely the automatic determination of the number of clusters, has not been fully addressed in previous work.

In this paper, we present a methodology to discover the cluster structure in videos using spectral algorithms. Towards this goal, we first study some measures to assess clustering quality, discussing the balance between the number of clusters and the clustering quality. In particular, we discuss the use of the eigengap for model selection, a measure referred to as a potential tool for clustering evaluation [5], but for which we are not aware of any experimental studies showing its usefulness in practice. Furthermore, we show that the application of spectral clustering to video results in a powerful method, despite the use of basic global features of visual similarity and temporal relations. We exemplify the performance of the methodology with respect to cluster detection and individual shot-cluster assignment using two different types of video material (home videos and soccer games), and show that our approach compares well to people performing the same task, and outperforms other existing automatic techniques. While the features used here are simple, the approach is general and could be extended to deal with more dedicated, domain-specific feature extraction algorithms in these and other types of video content.

The paper is organized as follows. Section 2 describes the spectral clustering algorithm, discussing the use of various clustering quality measures to automatically detect the number of clusters. Section 3 describes the application of the methodology to the extraction of meaningful clusters of home videos and soccer games. Section 4 describes the data sets and the performance measures, and presents results of our approach. Section 5 provides some concluding remarks.

The authors thank the Eastman Kodak Company for providing the Home Video Database, and Napat Triroj (University of Washington) for providing the multiple-subject third-party scene ground-truth. The soccer images are provided by the BBC in the context of the ASSAVID european project. This work was carried out in the framework of the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM)2.

2. THE SPECTRAL CLUSTERING ALGORITHM

First, we briefly describe the spectral algorithm (proposed in [5] and inspired by [9]). Model selection is then discussed, and two measures of assessing clustering quality are presented.

2.1. The algorithm

Let us define a graph $\mathcal{G}$ by (S, A), where S denotes the set of nodes, and A is the affinity matrix encoding the similarity between any two nodes in the set S. We ensure that $A_{ii} = 0$ for all i in S. The affinity A_{ij} is often defined as :

$$A_{ij} = \exp^{-\frac{d^2(i,j)}{2\sigma^2}} \ , \qquad (1)$$

where $d(i, j)$ denotes a distance measure between two nodes, and σ is a scale parameter. The algorithm consists of the following steps :

1. Define $D(A)$ to be the degree matrix of A (i.e. a diagonal matrix such that $D_{ii} = \sum_j A_{ij}$), and construct $L(A)$ by $L(A) = (D(A))^{-1/2} A (D(A))^{-1/2}$.
2. Find $\{x_1, x_2, \ldots, x_k\}$ the k largest eigenvectors of L, [1] and form the matrix $X = [x_1 x_2 \ldots x_k]$ by stacking the eigenvectors in columns.
3. Form the matrix Y from X by renormalizing each row to have unit length. The row Y_i is the new feature associated with node i.
4. Cluster the rows Y_i into k clusters via K-means.
5. Assign to each node i the cluster number corresponding to its row.

When the value of K corresponds to its true value, the rows of Y should cluster in K orthogonal directions. Thus, the K initial centroids $(Y_i^c)_{i=1,\ldots,K}$ in the fourth step of the algorithm can be selected by first identifying the row of Y whose $Ninit$ neighbours form the tightest cluster, and then recursively selecting the row whose inner product to the existing centroids is the smallest, according to :

$$Y_{i+1}^c = \underset{Y_j}{\text{argmin}} \max_{(Y_l^c)_{l=1:i}} (Y_l^c \cdot Y_j) \, .$$

2.2. Algorithm analysis

Fig. 1 shows examples of clustering results that can be obtained with this algorithm. It was shown in [5] that, under the condition that K corresponds to the true number of clusters (whenever such a value exists), the rows of Y should cluster in K orthogonal directions. Given the correct K value, the work in [5] exploited this property by computing a MSE distorsion measure at the end of the K-means step to select the final clustering result from a set of results obtained by varying the scale parameter σ in the affinity matrix computation (Eq. (1)).

[1] chosen to be mutually orthogonal in the case of repeated eigenvalues

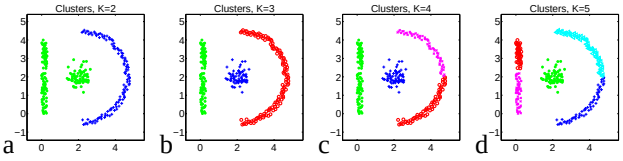

Fig. 1. A clustering example. Result with (a) K=2, (b) K=3, (c) K=4 (d) K=5.

In [6], we analyzed the behaviour of the algorithm for the case when K is different from this ideal number. When $K < K_{ideal}$, we showed that the orthogonal property may not hold in general, so that the MSE distorsion may be low or not in this case. When $K > K_{ideal}$, the orthogonal property does not hold and results in an overclustering of the ideal case (Fig. 1). Therefore, there is no clear indication of how the MSE measure would behave for varying values of K. Note in particular that the distortion measure is computed in spaces of different dimension (the rows Y_j lie in $\mathbb{R}^K$), so that distortion values may not be easily compared.

2.3. Automatic model selection

The selection of the "correct" number of clusters is a difficult task. We have seen in the previous Section that the analysis of the MSE measures for varying K is not trivial. For this reason, we considered other criteria stemming from matrix perturbation and spectral graph theories to perform model selection.

We have adopted the following strategy. The spectral clustering algorithm is employed to provide candidate solutions (one per value of K), and the selection is performed based on the criteria discussed in the following sections.

The eigengap

The eigengap is an important measure in spectral methods [10, 5]. The eigengap of a matrix A is defined by $\delta(A) = 1 - \frac{\lambda_2}{\lambda_1}$ where λ_1 and λ_2 are its two largest eigenvalues [10]. In practice, the eigengap is often used to assess the stability of the first eigenvector[2] of a matrix and it can be shown to be related to the *Cheeger constant*, a measure of the tightness of clusters. To clarify this relation, let us define the *cut value* of the partitioning $(\mathcal{I}, \bar{\mathcal{I}})$ of a graph with affinity matrix A by $Cut_A(\mathcal{I}, \bar{\mathcal{I}}) = \sum_{i \in \mathcal{I}} \sum_{j \notin \mathcal{I}} A_{ij}$. We also define the *volume* of the subset $\mathcal{I}$ by $Vol_A(\mathcal{I}) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}} A_{ij}$. Furthermore, the *conductance* ϕ of the partitioning $(\mathcal{I}, \bar{\mathcal{I}})$

[2] Or the first k eigenvectors, in cases where we have a k-repeated, largest eigenvalues.

is defined as

$$\phi_A(\mathcal{I}) = \frac{Cut_A(\mathcal{I}, \bar{\mathcal{I}})}{\min(Vol_A(\mathcal{I}), Vol_A(\bar{\mathcal{I}}))}.$$

The Cheeger constant h_G is defined as $h_G(A) = \min_{\mathcal{I}} \phi_A(\mathcal{I})$ and can be shown to be bounded by the eigengap [5, 10] : $h_G(A) \geq \frac{1}{2}\delta(A)$. The conductance indicates how well $(\mathcal{I}, \bar{\mathcal{I}})$ partitions the set of nodes into two subsets, and the minimum over $\mathcal{I}$ corresponds to the best partition. Therefore, if there exist a partition for which (i) the weights A_{ij} of the graph edges across the partition are small, and (ii) each of the regions in the partition has enough volume, then the Cheeger constant will be small. Starting from $K = 1$, we would like to select the simplest clustering model (i.e., the smallest K) for which the extracted clusters are tight enough (hard to split into two subsets). This is equivalent to request that the Cheeger constant is large enough for each cluster, or to request that the eigengap is large for all clusters. Our first criterion is

$$\delta_K = \min_{i \in 1 \ldots K} \delta(L(A_K^{(ii)})), \quad (2)$$

where $A_K^{(ii)}$ are the submatrices extracted from A according to the model obtained by the spectral algorithm, and L is defined in Section 2.1. The algorithm selects the smallest K for which the eigengap as defined by Eq. (2) exceeds a threshold.

The relative cut

The measure defined by Eq. (2) has a drawback, as it only considers intra-cluster information. When part of the data have no clearly defined clusters, the algorithm may overestimate the number of clusters so that all clusters (possibly reduced to a single element) are tight enough. We thus considered a second criterion that characterizes the overall quality of a clustering. This criterion is defined as the fraction of the total weight of edges not covered by the clusters,

$$\text{rcut}_K = \frac{\sum_{k=1}^{K} \sum_{l=1, l \neq k}^{K} \sum_{i \in S_k} \sum_{j \in S_l} A_{ij}}{\sum_i \sum_j A_{ij}}. \quad (3)$$

The algorithm outputs the largest K for which rcut is below a threshold.

3. SPECTRAL STRUCTURING OF VIDEOS

In this Section we explore the application of the spectral algorithm to the video shot clustering problem, based on general scene appearence models[3], to two type of contents : home videos and soccer games. We first describe the features employed to represent a shot. Then, we discuss the similarity measure between two feature points used to build the affinity matrix. As shots are represented by more than one feature point, an additional shot to cluster assignment step is required, as explained in Subsection 3.3.

[3]Needless to say, multiple valid partitions of the same video exist: clustering a video based on their scenes is clearly a different task than clustering it based on people identities.

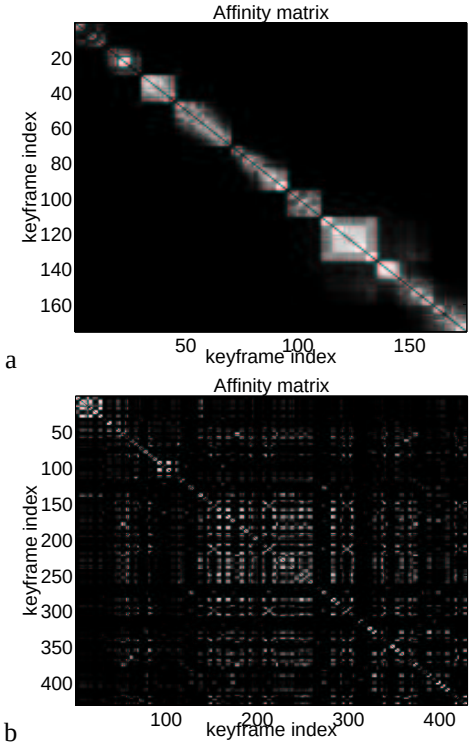

Fig. 2. Affinity matrix : (a) home video case (Video 16) and (b) soccer game case.

3.1. Shot representation and feature extraction

Video shots usually contain more than one appearance, due to camera motion. Consequently, more than one key-frame might be necessary to represent the intra-shot appearance variation. In this paper, a shot is represented by a small fixed number of key-frames, $N_{kf} = 5$. However, we are aware that the number and quality of key-frames could have an impact on clustering performance. Shots are further represented by standard visual features [2]. The $i-th$ key-frame f_i of a video is characterized by a histogram $h_i = \{h_{ij}\}$, where j runs over a set of (r, g, b) uniformly quantized color bins. However, when appropriate, a particular spatial layout of the image can be taken into account. In this case, h_i is a histogram with j running over (r, g, b, p) bins, where p represents a specific region of the image.

3.2. Similarity computation

In our approach, the affinity matrix A is directly built from the set of all key-frames in a video, indexed as a whole, but knowing the correspondence key-frame-shot. The similarity

measure between the key-frames should reflect the knowledge about the specific application domain. In the case of home videos, the content is unrestricted. Thus, similarity measures based on global scene appearence descriptors are a reasonable choice. In sports videos, such as soccer, more specific similarity measures could be defined. However, in the scope of this paper, we restrict ourselves to the use of the same type of global features, to show the sole effect of the clustering algorithm.

Similarity computation for home videos

Home videos contain series of ordered and temporally adjacent shots that can be organized in groups usually related to distinct scenes. Visual similarity and temporal ordering are two of the main criteria that allow people to identify clusters in video collections, when nothing else is known about the content (unlike the filmmaker, who knows details of context). The integration of visual similarity and temporal adjacency in a joint model is a sensible choice. The pairwise affinity matrix A is defined by

$$A_{ij} = A^v_{ij} A^t_{ij}, \text{ with } A^v_{ij} = e^{-\frac{d_v^2(f_i,f_j)}{2\sigma_v^2}}, \text{ and } A^t_{ij} = e^{-\frac{d_t^2(f_i,f_j)}{2\sigma_t^2}},$$

where A_{ij} is the affinity between key-frames f_i and f_j, d_v and d_t are measures of visual and temporal similarity, and σ_v^2 and σ_t^2 are visual and temporal scale parameters.

Visual similarity is computed by the metric based on Bhattacharyya coefficient, which has proven to be robust to compare color distributions [1],

$$d_v(f_i, f_j) = (1 - \rho_{BT}(h_i, h_j))^{1/2}, \quad (4)$$

where the ρ_{BT} denotes the Bhattacharyya coefficient, defined by $\rho_{BT} = \sum_k (h_{ik} h_{jk})^{1/2}$, the sum running over all bins in the histograms.

Temporal similarity exploits the fact that distant shots along the temporal axis are less likely to belong to the same scene, and is defined by $d_t(f_i, f_j) = \frac{||f_j| - |f_i||}{|vc|}$ where $|f_i|$ denote the absolute frame number of f_i in the video, and $|vc|$ denotes the entire video clip duration (in frames). Note that the range for both d_v and d_t is $[0, 1]$.

We set the scale parameters σ_v and σ_t in the following way. Building upon a previous study [2], we fixed the σ_v value to 0.25 which represents a good threshold for separating intra and inter-cluster similarities distributions in home videos. Similarly, it was shown in [2] that in average 70% of home video scenes are composed of four or less shots. Thus, the σ_t value was set to the average temporal separation between four shots in a given video.

Fig. 2(a) shows the affinity matrix computed for one of the home video (corresponding to the video Fig. 4(a)). Bright points correspond to large pairwise similarity. The matrix exhibits a nice block diagonal pattern, due mainly to the fact that similar shots usually correspond to adjacent shots, and to the time-dependent similarity term, which limits the amplitude of the off-diagonal terms.

	H	PHC	SM
SIE_{min}	0.078	0.156	0.116
SIE_{med}	0.275	0.362	0.271
SIE_{max}	0.535	0.532	0.539

Tab. 1. Average of the percentage of shots in error for humans (H), the probabilistic hierarchical clustering algorithm (PHC), and the spectral method (SM)

Similarity computation for soccer games videos

In the context of sports videos, domain knowledge information could be used to analyze their content and temporal structure. In the scope of this paper, we only consider global appearence information, neglecting other useful information (e.g. camera motion, motion activity, detection of specific regions like grass). The only domain knowledge that we use is introduced by splitting each image horizontally into a 2/5 top region and 3/5 bottom region and representing a key-frame by a multidimensional histogram as described in Subsection 3.1. Furthermore, as in this application distant shots can belong to the same scene, we defined the similarity A_{ij} as the visual similarity alone, i.e. $A_{ij} = A^v_{ij}$. The similarity scale value σ_v is kept to a value of 0.25.

Fig. 2(b) shows the affinity matrix obtained in the case of soccer data. The block effect is due to the alternance of wide shots (with grass) and close-up shots. The latter shot category yields less intra cluster similarities and therefore produces less bright blocks (on the diagonal or off-diagonal) in the matrix.

3.3. Shot assignment after spectral clustering

The spectral method is applied as discussed in Section 2. A cluster number is then assigned to each shot using a simple majority rule on the cluster labels of its key-frames. In the case of a tie, the cluster is randomly selected from the possible candidates.

4. EXPERIMENTS AND RESULTS

4.1. Home video experiments

Ground-truth generation

Although shot clustering into scenes is a core function in video content analysis, performance evaluation measures and procedures for this task are not standardized. The objective evaluation of a shot clustering algorithm assumes the existence of a ground-truth (GT) at the scene level. At least two options are conceivable. In the first-party approach,

the GT is generated by the content creator [7] thus incorporating specific context and production model knowledge (location relationships in home videos, or specific camera sources in soccer videos) that cannot be easily extracted by current automatic means. In contrast, a third-party GT is defined by a subject other than the filmmaker (not familiar with the content). In this case, there still exists human context understanding, but limited to what is displayed [2].

One criticism against the latter methodology is the reasonable claim that different people generate distinct GTs, and therefore no single judgement can be reliable. The question of human judgement consistency for scene structuring refers to the general problem of perceptual organization of visual information [4]. One could expect that variations in human judgement arise both from distinct perceptions of a video structure and from different levels of granularity in it [4]. Modeling these variations would be useful to evaluate clustering algorithms against human performance. Similar objectives have been pursued for image segmentation [4] and clustering of still images.

We use a third-party GT based on multiple subject judgement to take into account the fact that different people might generate different results. In the first place, a ground-truth (GT) at the shot level can be generated. In the second place, scenes for each video can be found by a number of people guided by a purposedly general statement about the clustering task (e.g. "group shots together if you believe they belong to the same scene"), with no initial solution. The clustering is made using a GUI that displays a key-frame-based video summary. This methodology has been applied on a six-hour home video database.

Performance measures

We propose to use two performance measures: the number of clusters selected by the algorithm, and the number of shots in error (SIE). For the number of clusters, we report the value we obtain and compare it with the numbers provided by people. For shot in errors, let us denote $GT^i = \{GT^i_j, j \in 1, ..., N_i\}$ the set of human GTs for the video V_i, and $\mathcal{C}^i$ the solution of an algorithm for the same video. The SIE between the clustering result $\mathcal{C}^i$ and a ground-truth GT^i_j is defined as the number of shots whose cluster label in $\mathcal{C}^i$ does not match the label in the GT. This figure is computed between $\mathcal{C}_i$ and each GT^i_j, and the GTs are ranked according to this measure. We then keep three measures : the minimum, the median and the maximum value of the SIE, denoted SIE^i_{min}, SIE^i_{med} and SIE^i_{max} respectively. The minimum value SIE_{min} provides us an indication of how far an automatic clustering is from the nearest segmentation provided by a human. The median value can be considered as a fair measure of how well the algorithm performs, taking into account the majority of the human GTs and excluding the largest errors. These large errors may come from outliers and are taken into account by SIE^i_{max}, which gives an idea of the spread of the measures. For the overall performance measure, we computed the average SIE measures over all the videos, of the percentage of shots in errors w.r.t. the number of shots in each video. Note that this normalization is necessary because the number of clusters and shots might vary considerably from one video to another.

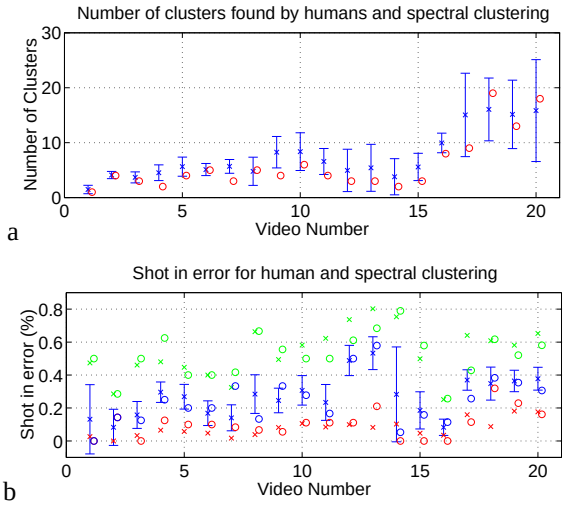

Fig. 3. (a) Determination of number of clusters. (b) Percentage of shot in error. The blue bar indicates the spread of the human performances of the SIE_{med} value.

Data set

The data set consists of 20 20-minute MPEG-1 home videos (430 shots) provided by seven people, which depict vacations, school parties, weddings, etc. The number of shots per video varies considerably (4-62 shots). Following the procedure described in the previous section, scenes for each video were found by about twenty subjects, so the GT consists of nearly 400 human segmentations.

Results

The best result with our method was obtained using the eigengap criterion and a threshold $\delta_K = 0.15$. We compared it with a probabilistic hierarchical clustering method (PHC) [2], which has been shown to perform better results than traditional methods (e..g. K-means), as well as with human performance. The latter was obtained in the following way : for each video, the minimum, median and maximum shots in error were computed for each human GT against all the others. These values were then averaged over all subjects. These averages are plotted in Fig. 3 for each video. Finally we computed the average over all the videos to get the overall performance.

Table 1 summarizes the results. We can first notice from the

Fig. 4. Home video structuring examples.(a) Video 16 (b) Video 7. Only one keyframe of each shot is displayed.

minimum and maximum values that the spread of performances is high, given the performance measure. Secondly, the spectral method is performing better than PHC, as can be seen from the median and minimum value, and approximately as well as people.

Fig. 3 displays the results obtained for each video. First, in Fig. 3(a), we show the number of detected clusters (the red circles) as predicted by the algorithm and compare them to the mean of the number of clusters in the GT. The spread of the cluster numbers in the GT is represented by the blue bar (plus or minus one standard deviation). Note that the videos have been ordered according to their number of shots. The detected cluster numbers are in good accordance with the GT, though slightly underestimated. Fig. 3(b) displays the values of the shot in error measures in comparison to the average of human performance. The circles depict the measures obtained with our method and the crosses denote human performance. The color represents the different measures (minimum in red, median in blue, and maximum in green). The median performance of our algorithm is better than the average human in eight cases and worse in six cases. Notice that in 25% of the cases, our algorithm provides a segmentation that also exists in the GT.

Two examples of the generated clusters are shown in Fig. 4. Each cluster is displayed as a row of shots, which in turn are represented by one keyframe (labeled e). Qualitatively, the method provides sensible results.

Fig. 5 shows the obtained results using the two criteria. The selection with the eigengap criterion slightly outperforms the results obtained with the relative cut. We can also notice that the results are quite consistent over a range of thresholds (in any case, better than PHC). We also considered the MSE distorsion measure [5] as a criterion, but we could not obtain good results with it.

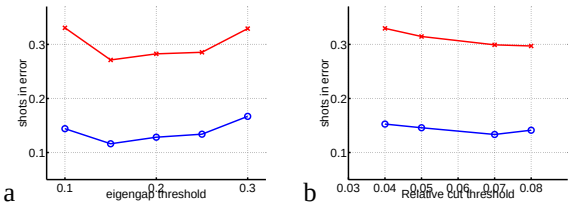

Fig. 5. Variation of the average of percentage of shots in error (average of the median in red, of the min in blue) for different criteria as function of their threshold: (a) eigengap; (range: (0.1,0.3)); (b) relative cut (range: (0.04,0.08)).

4.2. Soccer games results

We applied the spectral clustering (SC) algorithm on one 10-minute soccer game video clip (86 shots). The selection algorithm based on the eigengap and a threshold of $\delta_K = 0.15$ applied to the soccer data gave a value of K equal to 10 [4].

Fig. 6 displays the 10 clusters obtained with the method. Remember that only one keyframe is displayed per shot. For instance, some wide shots may correspond to play actions taking place on both side of the playfield, leading to keyframes with and without crowd at the top. The obtained clusters are visually quite consistent. The first cluster correspond to the display of the players names; the second mainly to wide field of views with the crowd at the top of the image, including shots taken from the playfield side; the third one to close-ups of players with a crowd background. The fourth cluster is a mixture of close-ups with grass background and medium field of view shots. The fith one contains mainly a dark grass, while the sixth cluster contains shots with lighter grass and more black and white pixels. The four last clusters exhibits single shots with distinctive color signatures.

For comparison purposes, in Fig.7 we present the result obtained with the standard K-means (KC) algorithm[5] applied on the multidimensional histogram data. The clusters present more inconsistencies than with the spectral method. For instance, the third and fourth clusters seems to correspond to the same class. In the fifth cluster, two close-up shots of players with grass don't seem to belong to this clus-

[4]Note that, due to the absence of time-dependent similarity term, there is more inter-cluster similarity than in the home video case. Thus, a threshold value of around 0.15 would be necessary to select the same number of cluster with the relative-cut criterion.

[5]More precisely, 10 runs of the K-means algorithm are performed and the result with the lowest MSE is kept.

ter. Close-up shots of white players with grass are disseminated in 5 different clusters. Close-ups of red players in the ninth cluster are similar to the first shot of cluster 5.

To evaluate the difference between the two algorithms (SC and KC), we performed the following 'user' experiment. Video summaries similar to the ones in Fig. 6 were produced for three 10-minute soccer video clips (with resp. 61, 71 and 85 shots), and printed on paper[6]. Then, 10 people were asked to perform the following tasks : "First, observe one summary carefully and then report for each cluster those shots that do not belong to such cluster or that fit better in a different cluster. Then, do the same for the summary obtained with the other algorithm on the same video clip[7]. Secondly, looking at both summaries, tell which one is better and why."
For the first task, we obtained the following results : an average of 8.7, 8.1 and 10.6 shots in error were counted in the three KC summaries, making a total average of 9.13. The results of the SC summaries are the following : 8.7, 6.5 and 10.1 with a total average of 8.43. For the second task, the numbers were the following : 23 times the SC summary was said to be better, 2 times the KC, and in 5 cases the person thought they were equivalent. The results for the second task are suprisingly good for SC given the small and unsignificant difference between the numbers of reported shot in error. Here, human perception does not match the numbers. The main comments made by the people can be summarized as follows : the spectral clustering algorithm produces more homogeneous clusters that make more sense, with mistakes that are acceptable. On the contrary, the K-means algorithm was said to produce too many similar sized clusters with a small number of errors everywhere.

5. CONCLUSION

In this paper we have described a method for clustering video shots using a spectral method. In particular, we investigated the automatic selection of the number of clusters, which is currently an open research issue for spectral methods. The algorithm was applied to a six-hour home video database and to soccer data, and the results are favorably compared to existing techniques as well as human performance.
The improvement of the methodology can be achieved by designing better similarity distances between shots or images. This can be done by using other cues such as motion or texture. However, good ways of combining these cues into one similarity matrix (and the effect on the clustering algorithm) is still an open issue. Nevertheless, in the context of a specific application, dedicated similarity distances could be defined and are expected to lead to more precise and finer clustering results.

[6]We printed one summary per page. Thus, we were able to represent each shot with three key-frames. Furthermore, to avoid a perceptual layout bias between summaries, clusters were reorganized such that the grass clusters were displayed on top of the summaries.

[7]Half of the subjects looked at the SC summary first ; the other half viewed the KC summary first.

6. REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," in *Proc. IEEE CVPR.*, June 2000.

[2] D. Gatica-Perez, M.-T. Sun, and A. Loui, "Consumer Video Structuring by Probabilistic Merging of Video Segments," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Aug. 2001.

[3] J.R. Kender and B.L. Yeo, "On the Structure and Analysis of Home Videos," in *Proc. Asian Conf. Comp. Vision*, Jan. 2000.

[4] D. Martin, C. Fowlkes, D. Tal, J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," in *IEEE ICCV*, 2001.

[5] A. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Proc. NIPS*, Dec 2001.

[6] J.-M. Odobez, D. Gatica-Perez and M. Guillemot, "On Spectral Methods and Structuring of Home Videos," IDIAP Technical Report, IDIAP-RR-55, Nov. 2002.

[7] J. Platt "AutoAlbum: Clustering Digital Photographs using Probablisitic Model Merging," in *Proc. IEEE Workshop on Content-Based Access to Image and Video Libraries*,2000.

[8] Y. Rui and T. Huang, "A Unified Framework for Video Browsing and Retrieval," in A. Bovik, Ed., *Image and Video Processing Handbook*, Academic Press, pp.705-715, 2000.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[10] S. Vempala R. Kannan and A. Vetta, "On clusterings - good, bad and spectral," in *Proc. 41st Symposium on the Foundation of Computer Science, FOCS*, 2000.

[11] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *IEEE ICCV*, 1999.

[12] M. Yeung, B.L. Yeo, and B. Liu, "Segmentation of Video by Clustering and Graph Analysis," *Computer Vision and Image Understanding*, Vol. 71, No. 1, pp. 94-109, July 1998.

[13] H.-J. Zhang, S.Y. Tan, S.W. Smoliar, and G. Yihong, "Automatic parsing and indexing of news video" *Multimedia Systems* , 2(6):256–266, 1995.

[14] Di Zhong and Shih-Fu Chang, "Structure Analysis of Sports Video Using Domain Models" *IEEE ICME*, Aug. 2001.

Fig. 6. Soccer game clustering result with Spectral method. Only one keyframe of each shot is displayed.

Fig. 7. Soccer game clustering result with Kmeans method. Only one keyframe of each shot is displayed.