IDIAP RESEARCH REPORT

On Spectral Methods and the Structuring of Home Videos

Jean-Marc Odobez [*] Daniel Gatica-Perez [*]
Mael Guillemot [*]

IDIAP–RR 02-55

PUBLISHED IN
International Conference on Image and Video Retrieval (CIVR) 2003.

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

* IDIAP, Martigny, Switzerland

IDIAP Research Report 02-55

On Spectral Methods and the Structuring of Home Videos

Jean-Marc Odobez Daniel Gatica-Perez Mael Guillemot

PUBLISHED IN
International Conference on Image and Video Retrieval (CIVR) 2003.

Abstract. Accessing and organizing home videos present technical challenges due to their unrestricted content and 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 exploit the eigenvector decomposition of a pair-wise similarity matrix and can 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. We first analyze the behaviour of the algorithm with respect to variations in the number of clusters, and then propose measures to assess the validity of a grouping result. The methodology is used to group scenes from a six-hour home video database, and is assessed with respect to a ground-truth generated by multiple humans. The results indicate the validity of the proposed approach, both compared to existing techniques as well as the human ground-truth.

1 Introduction

The organization and edition of personal memories contained in home videos constitute a technical challenge due to the lack of efficient tools. The development of browsing and retrieval techniques for home video would open doors to video albuming and other multimedia applications [8], [5]. Unrestricted content and the absence of storyline are the main characteristics of consumer video. Home videos are composed of a set of *scenes*, each composed of one or a few video shots, visually consistent, and randomly recorded along time. Such features make consumer video unsuitable for analysis approaches based on storyline models, and have diverted research on home video analysis until recently, as it was generally assumed that home videos lack of any structure [8, 5]. However, recent studies have revealed that the behaviour of home filmmakers induces certain structure [6, 4], as people implicitly follow certain rules of attention focusing and recording. The structure induced by these filming trends is often semantically meaningful. In particular, the scene structure of home video can be disclosed from such rules [4].

At the same time, there is an increasing interest in computer vision and machine learning towards spectral clustering methods [16, 18, 7, 9], which aim at partitioning a graph based on the eigenvectors of its pair-wise similarity matrix. These methods have provided some of the best known results for image segmentation and data clustering, several relevant issues remain unsolved. One of them is model selection. Several of the current techniques partition a graph in two sets, and are recursively applied to find K clusters [16, 7]. However, it has been experimentally observed that using more eigenvectors and computing directly a K-way partitioning provides better results [1]. In [18], Weiss performed a comparative analysis of four spectral methods employed in computer vision. His analysis led other authors to propose a new algorithm [10] that uses K eigenvectors simultaneously and combines the advantages of two other algorithms [15, 16], demonstrating theoretically why the algorithm works under some conditions. However, in most of these references, the automatic determination of the number of classes has not been fully addressed.

In this paper, we propose a methodology to discover the cluster structure in home videos using spectral algorithms. Our paper has two contributions. In the first place, we present a novel analysis related to the problem of model selection in spectral clustering. We first extend the analysis of the performance of the algorithm of [10] when the number of clusters is not the "correct" one. Then, we study some measures to assess the quality of a partition, and discuss the balance between the number of clusters and the clustering quality. In particular, we discuss the use of the eigengap, a measure often used in matrix perturbation and spectral graph theories [7, 10], and referred to as a potential tool for clustering evaluation [10, 9], but for which we are not aware of any experimental studies showing its usefulness in practice. In the second place, we show that the application of spectral clustering to home video structuring results in a powerful method, despite the use of simple features of visual similarity and temporal relations. The methodology shows good performance with respect to cluster detection and individual shot-cluster assignment, both compared to existing techniques and to humans performing the same task, when evaluated on a six-hour home video database for which a third party ground-truth generated by multiple subjects is available.

The rest of the paper is organized as follows. Section 2 describes in details the spectral clustering algorithm, presenting an analysis of algorithmic performance with respect to model selection, and discussing the use of various measures to assess the validity of a grouping result. Section 3 describes the application of the methodology to structuring of home videos. Section 4 describes the database and the performance measures, and presents results of our approach compared to existing techniques as well as to human performance. Section 5 provides some concluding remarks.

2 The spectral clustering algorithm

In this section we briefly describe the spectral algorithm we employ (proposed in [10] and inspired by [16, 15]). The algorithm is then analyzed for both ideal and general cases. The choice of the number

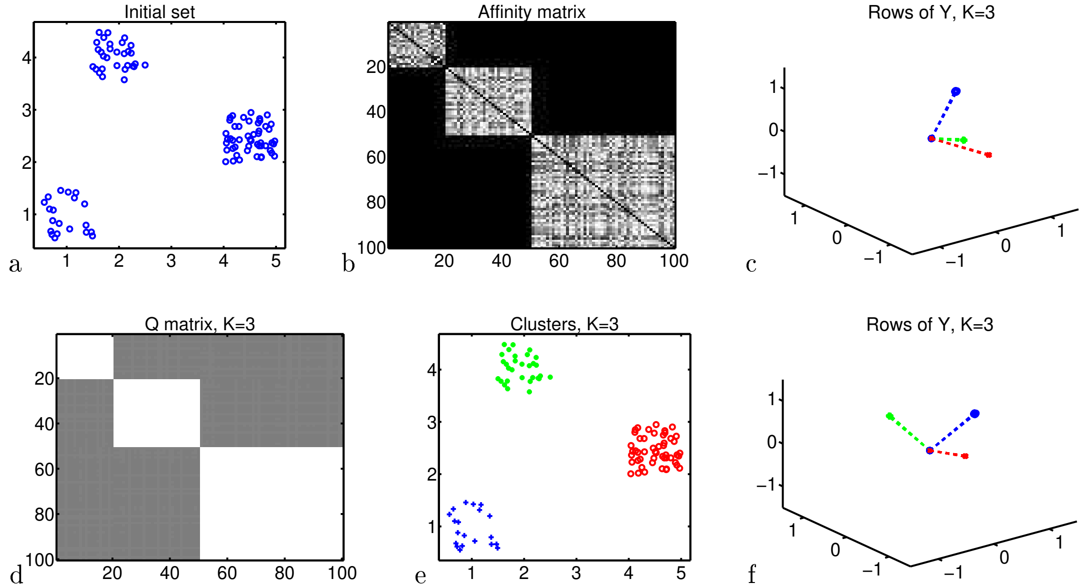

Figure 1: Clustering example : (a) initial points; (b) the affinity matrix; (c) the rows of Y (in $\mathbb{R}^3$) when K=3 and eigensystem solved with *eig* from matlab; (d) the Q matrix; (e) the clustering result; (f) the rows of Y, but with the eigensystem solved with the *eigs* function.

of clusters is discussed, and several 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 value associated with the edges of the graph. $A$is built from the pair-wise similarity defined 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 :

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

where $d(i, j)$ denotes a distance measure between two nodes, and σ is a scale parameter. Following the notation in [10], 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 :

$$L(A) = (D(A))^{-1/2} \, A \, (D(A))^{-1/2} \tag{2}$$

2. Find $\{x_1, x_2, \ldots, x_k\}$ the k largest eigenvectors of L (chosen to be mutually orthogonal in the case of repeated eigenvalues), 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 to the new feature associated with node i.

4. Treating each row of Y as a point in $\mathbb{R}^K$, cluster them into k clusters via K-means.

5. Finally, assign to each node of the set S the cluster number corresponding to its row.

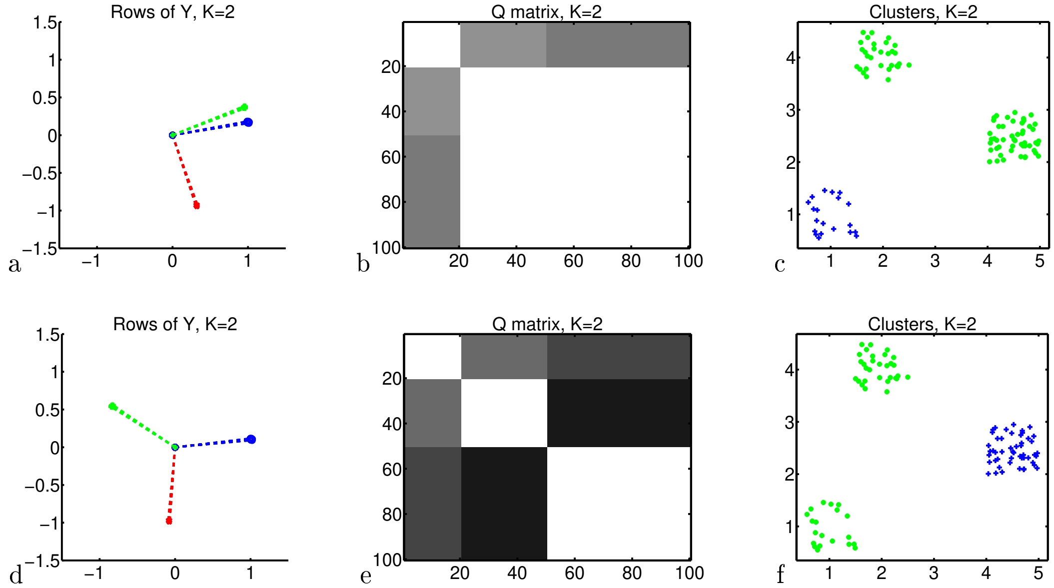

Figure 2: Same data as in figure 1, with K=2 : (a,b,c) when the eigensystem is solved with *eig* on matlab ; (d,e,f) when using *eigs*. (a) (d) denote the rows of Y (in $\mathbb{R}^2$); (b) (e) denote the Q matrix; (c)(f) show the clustering result.

As will be explained in the following section, when the value of K corresponds to its true value, the rows of Y should cluster in K orthogonal directions. Exploiting this property, the K initial centroids $(Y_i^c)_{i=1,\dots,K}$ in the fourth step of the agorithm can be selected from the rows of Y by first finding the row of Y for which the $Ninit$ neighbours form the tightest cluster, and then recursively selecting the row whose inner product to the existing centroids is the smallest according to :

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

where Y_j denotes the j-th row of Y.

2.2 Algorithm analysis

Figure 1(e) and 4(b) show examples of clustering results that can be obtained with this algorithm. It was shown in [10] that the above algorithm is able to find the true clusters under the condition that K corresponds to the true number of clusters (whenever such a value exists). In this section we extend this result by analyzing the behaviour for the case when K is above or below this ideal number. Two cases are considered: the ideal case, when the true clusters are well separated; and the general case, when noise due to inter-cluster similarity exists.

2.2.1 The ideal case

To understand the behaviour of the algorithm, we consider an ideal case in which the different clusters have infinite separation. Without loss of generality, if we additionally suppose that K_{ideal}=3, the set of all node indexes is given by $S = S_1 \cup S_2 \cup S_3$, where S_i denotes the i^{th} cluster of size n_i. We also assume that the node indexes are ordered according to their cluster. An example obeying these assumptions is illustrated in Fig. 1, where the distance employed to define the affinity between two nodes is the usual euclidian distance between the 2D coordinates, and affinity is computed by Eq. 1.

In this case, A (resp. L) is a diagonal matrix composed of 3 blocks $(A^{(ii)})_{i=1,2,3}$ (resp. $(L^{(ii)})_{i=1,2,3}$) which are the intra-cluster affinity matrices for L. It follows that (i) its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks $L^{(ii)}$ (the latter appropriately padded with zeros); (ii) its highest eigenvalue is unity; (iii) unity is a repeated eigenvalue of order 3; (iv) the 4th eigenvalue is stricly less than 1 (assuming $A^{(ii)}_{jk} > 0, j \neq k$) and (v) the resulting eigenspace of the unity eigenvalue has dimension 3, and thus, the eigenvectors provided by a particular decomposition algorithm are not unique. In this case X_3 (where X_K denotes the first K eigenvectors stacked in columns) is of the form :

$$X_3 = \begin{bmatrix} v_1^{(1)} & 0 & 0 \\ 0 & v_1^{(2)} & 0 \\ 0 & 0 & v_1^{(3)} \end{bmatrix} \times R , \qquad \text{with } R = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} ,$$

where R is a 3×3 rotation matrix composed out of the three row vectors $\mathbf{r}_i$, and $v_l^{(j)}$ denotes the l^{th} eigenvector of the matrix $L^{(jj)}$. Thus, each row of X_3 is of the form $v_{1i}^{(j)} \times \mathbf{r}_j$, where $v_{1i}^{(j)}$ is a scalar (the i^{th} component of $v_1^{(j)}$). Therefore, after renormalizing the rows of X_3 (step 3 of the algorithm), the matrix Y has rows that fulfill $Y_i = \mathbf{r}_j \; \forall i \in S_j$. Fig. 1(c) illustrates this result for the data set of Fig. 1(a). Fig. 1(f) shows the result obtained by changing the matlab function that solves the eigensystem. Note that the three vectors are still orthogonal, but have a different configuration. An alternative formulation defines $Q = Y \times Y^T$ [15]. In the ideal case, we have $Q(i,j) = 1$ for nodes i and j belonging to the same cluster, and $Q(i,j) = 0$ otherwise (see Fig. 1(d)).

2.2.2 variation in the number of clusters in the ideal case

As we are interested in estimating K, let us consider the two cases when $K \neq K_{ideal}$, which have not been studied in [10]:

1. For $K < K_{ideal}$, X_K simply corresponds to the first K columns of $X_{K_{ideal}}$. After normalization, we get a Y matrix whose entries are (in our example, with K=2) :

   $$Y_j = (r_{i1}, r_{i2})/\|(r_{i1}, r_{i2})\| \doteq \mathbf{r}'_i \qquad \forall i \text{ and } \forall j \in S_i .$$

   This simply corresponds to projecting (and normalizing) the initial orthogonal vectors $\mathbf{r}_i$ into a lower dimensional space. Note that this may indeed cause some normalization problem when the projections are near the origin. Since (as pointed out above) the vectors $\mathbf{r}_i$ can be in any orthogonal configuration, there is no general rule about the configuration of their projections $\mathbf{r}'_i$. As an example, Fig. 2 shows these projections in the case of the data of Fig. 1. Note that, depending on the specific eigensystem solver, the projections and the clustering results can differ.

2. For $K > K_{ideal}$, consider for simplicity that $K = 4$. Thus, X_4 consists in the matrix X_3 with the fourth eigenvector appended as an extra column. As mentioned above, this eigenvector originates from one of the $L^{(ii)}$ matrices. More precisely $\lambda_4 = \max_i \lambda_2^{(i)}$. Assume that we choose this fourth eigenvector in the first cluster. We have :

   $$X_4 = \begin{bmatrix} v_1^{(1)}\mathbf{r}_1 & v_2^{(1)} \\ v_1^{(2)}\mathbf{r}_2 & 0 \\ v_1^{(3)}\mathbf{r}_3 & 0 \end{bmatrix} .$$

   After row normalization, it is easy to show that the resulting $Q = Y \times Y^T$ matrix has the following property:

   $$Q(i,j) = 0 , \qquad \forall i \in S_k, \forall j \in S_l, k \neq l.$$

   For example, $\forall (i,j) \in S_1 \times S_2$:

   $$Q(i,j) = Y_i Y_j^T = (v_{1i}^{(1)}\mathbf{r}_1, v_{2i}^{(1)})(v_{1(j-n_1)}^{(2)}\mathbf{r}_2, 0)^T = 0,$$

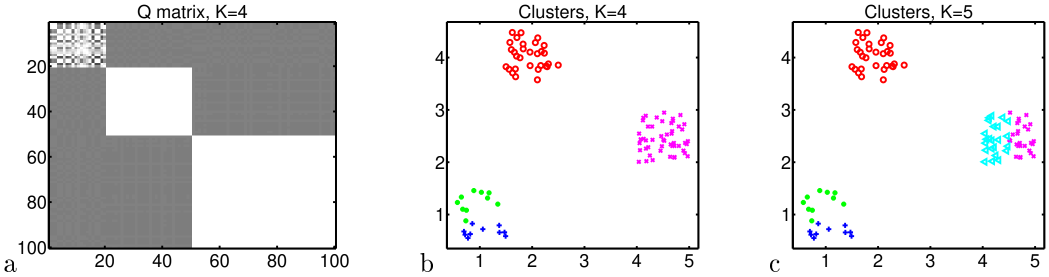

Figure 3: Same data as in Fig. 1. (a) the Q matrix when K=4; (b) the corresponding result; (c) clustering result when K=5.

meaning that the true original clusters remain orthogonal to each other. Furthermore, we have :

$$Q(i,j) = 1, \forall (i,j) \in S_k^2, k = 2 \text{ or } 3,$$

indicating that the second and third cluster remain unchanged. Indeed, only the first cluster is affected, and is divided into two parts. The same kind of reasoning can be applied when moving to higher values of K. To summarize, when $K > K_{ideal}$, the resulting clustering corresponds to an overclustering of the ideal case. This point is illustrated in Fig.s 3 and 4.

2.2.3 The general case

In the ideal case, we have seen that the Q matrix should only have 0 and 1 entries when the true K is selected, and that there might be other entry values when $K \neq K_{ideal}$ (esp. $K > K_{ideal}$). Indeed, this can be related to the distortion obtained at the end of the K-means algorithm :

$$MSE = \frac{1}{n} \sum_{i=1}^{K} \sum_{j \in cluster_i} \|Y_j - Y_i^c\|, \quad (3)$$

where Y_i^c represent the centroids at the end of the K-means. In the ideal case, and when $K = K_{ideal}$ the distortion should be 0. Furthermore, for the real case, it was shown in [10] that the distortion (computed with respect to the ideal cluster centers $\mathbf{r}_i$) is bounded by some value that depends on the entries of the affinity matrix (related to the clusters' density, the intra-cluster connectivity, etc). Given the correct K value, the authors in [10] use the distortion as a measure to select the clustering result from a set of results obtained by varying the scale parameter σ in the affinity matrix computation (Eq. (1)). However, the actual value of the bound in their experiments is not specified, and there is no indication of how this bound would behave for varying values of K. Note in particular that the distortion measure is computed in spaces of different dimension (Y_j lie in $\mathbb{R}^K$), so 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 different 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.

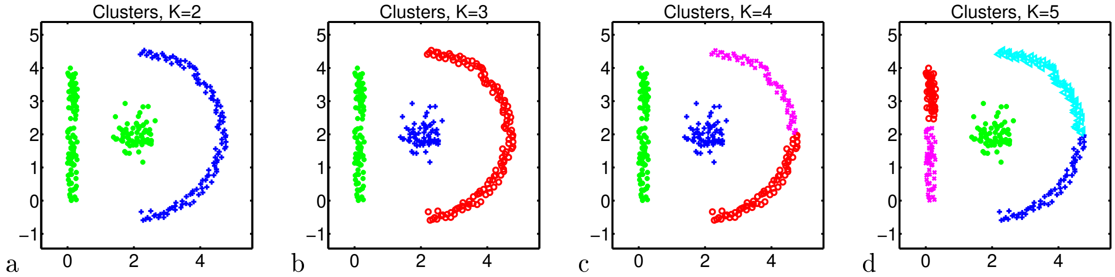

Figure 4: Another example. The clustering result with (a) K=2, (b) K=3, (c) K=4 (d) K=5

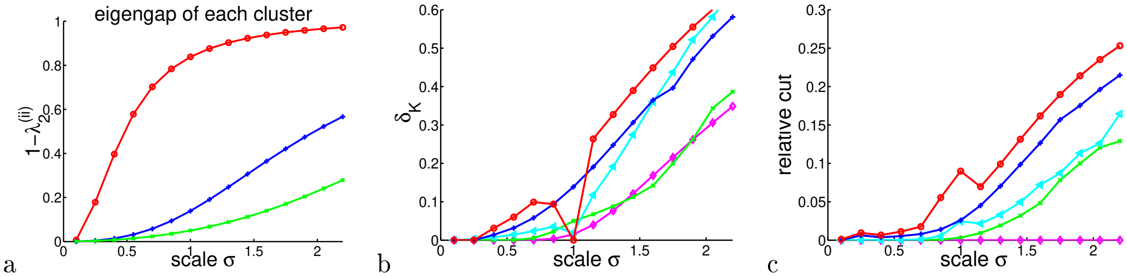

Figure 5: Influence of scale change. (a) eigengap of individual clusters : the red curve at the top corresponds to the "ball like" cluster; the blue curve in the middle corresponds to vertical line cluster; the green curve at the bootom corresponds to the half-circle cluster; (b) eigengap measure δ_K for K=1 (the magenta curve with diamonds), K=2 (the green curve with x's), K=3 (the cyan curve with triangles), K=4 (the blue curve with +'s) and K=5 (the red curve with o's) (c) relative cut rcut$_K$. Same labeling than in (b).

2.3.1 The eigengap

The eigengap is an important measure in the analysis of spectral methods [7, 9, 10]. Please refer to [2] for basic definitions. The eigengap of a matrix A is defined by $\delta(A) = 1 - \frac{\lambda_2}{\lambda_1}$ where λ_1 and λ_2 are the two largest eigenvalues of A [7]. In practice, the eigengap is often used to assess the stability of the first eigenvector[1] of a matrix and it can be shown to be related to the *Cheeger constant* [2], 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 characterized by its 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 $(\mathcal{I}, \bar{\mathcal{I}})$ 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 is defined as the minimum of the conductance over all the possible bipartitionings of the graph,

$$h_G(A) = \min_{\mathcal{I}} \phi_A(\mathcal{I}).$$

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

It can be shown that the above quantity is bounded by the eigengap [10, 7] :

$$h_G(A) \geq \frac{1}{2}\delta(A). \tag{4}$$

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 due to Eq. (4), to request that the eigengap is large for all clusters. Our first criterion is thus defined by

$$\delta_K = \min_{i \in 1 \dots K} \delta(L(A_K^{(ii)})), \tag{5}$$

where the $A_K^{(ii)}$ are the submatrices extracted from A according to the model obtained by the spectral algorithm, and L is defined by Eq. (2). The selection algorithm selects the smallest K value for which the eigengap as defined by Eq. (5) exceeds a threshold.

2.3.2 The relative cut

The clustering measure defined by Eq. (5) has a drawback. The algorithm could select a clustering with many clusters of relatively low quality so that the minimum in Eq. (5) is above the threshold, and reject a clustering of overall good quality that unfortunately has one cluster of very low quality [7]. We thus considered a second criterion that caracterizes the overall quality of a clustering. This criterion is defined as the fraction of the total weight of edges that are not covered by the clusters,

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

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

2.4 Scale analysis and selection

The selection algorithm described above requires the setting of some threshold, which is itself dependent on the setting of the scale parameters required in the definition of the pairwise affinity between nodes. For instance, let us consider the case of Fig. 4, where the affinity is defined by Eq. (1). Fig. 5(a) plots the eigengap of each of the three clusters considered separately. As expected, as the scale parameter increases, the eigengap increases as well, meaning that clusters become harder to split. Note that this increase is quite regular (there is no 'step' effect in the considered scale range) and that the increase rate is very dependent on the cluster type, whether nodes are concentrated (top red curve) or positioned along a 1-dimensional manifold (lower blue and green curves).

Fig. 5(b,c) displays the evolution of our criteria for different K values. The analysis of the δ_K curves exhibits two trends : before $\sigma = 1$, the nodes can more or less be split into 4 parts ($\delta_1, \delta_2, \delta_3$ are near zero and separated from δ_4 and δ_5), whereas above $\sigma = 1$, there is no evidence for more than one cluster (δ_1 starts increasing rapidly), or at most two clusters ($\delta_1 \simeq \delta_2$). The analysis of the relative cut measure provides similar trends. Note (in Fig. 5(b and 5(c) the behaviour when $K = 5$ around $\sigma = 1$, due to bad K-means initialization and numerical instabilities.

The main conclusion that can be drawn from these curves is that the scale value has a direct influence on the measured criteria. This should not come as a surprise since clustering is inherently a scale-dependent problem.

3 Spectral structuring of home videos

The structure of home video bears similarity to the structure of consumer still pictures [11]: videos contain series of *ordered and temporally adjacent shots* that can be organized in groups that convey semantic meaning, usually related to distinct scenes. Visual similarity and temporal ordering are indeed the two 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). A clustering algorithm that integrates visual similarity and temporal adjacency in a joint model is therefore a sensible choice. Previous formulations in the literature are based on the same idea [17], [12]. We explore the use of spectral clustering, as described in details in the following sections.

3.1 Shot representation and feature extraction

Home video shots usually contain more than one appearance, due to hand-held camera motion. Consequently, more than one key-frame might be necessary to represent the inter-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 [4]. The $i-th$ key-frame f_i of a video is characterized by a color histogram h_i in the RGB space (uniformly quantized to $8 \times 8 \times 8$ bins).

3.2 Similarity computation

The pair-wise 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) by defining

$$A_{ij} = e^{-\left(\frac{d_v^2(f_i,f_j)}{2\sigma_v^2} + \frac{d_t^2(f_i,f_j)}{2\sigma_t^2}\right)}, \tag{6}$$

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 [3],

$$d_v(f_i, f_j) = (1 - \rho_{BT}(h_i, h_j))^{1/2}, \tag{7}$$

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

$$d_t(f_i, f_j) = \frac{|\,|f_j| - |f_i|\,|}{|v|} \tag{8}$$

where $|f_i|$ denote the absolute frame number of f_i in the video, and $|v|$ denotes the entire video clip duration (in frames). Similar features have been used by other authors in different formulations [17]. Note that the range for both d_v and d_t is $[0, 1]$.

Taking into account the discusion in Subsection 2.4, we set the scale parameters σ_v and σ_t in the following way. Building upon a previous study of home videos in [4], 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 [4] that almost 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.

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

In this Section we first describe the dataset and the performance measures that we use for evaluation. We then compare the best result obtained by the spectral method with respect to the performance of humans as well as with a probabilistic hierarchical clustering method [4]. The third subsection is devoted to the comparison between the different criterions used for the selection of K.

4.1 Data set and ground-truth

The data set consists of 20 MPEG-1 home videos, digitized from VHS tapes provided by seven different people, and with approximate individual duration of 20 minutes. The videos depict vacations, school parties, weddings, and children playing in indoor/outdoor scenarios. A ground-truth (GT) at the shot level was semi-automatically generated, resulting in a total of 430 shots. The number of shots per video varies considerably (between 4 and 62 shots).

There are two typical options to define the GT at the scene level. In the first-party approach, the GT is generated by the video creator [11]. This method incorporates specific context knowledge about the content (e.g., family links, and location relationships) that cannot be automatically extracted by current automatic means. In contrast, a third-party GT is defined by a subject not familiar with the content. In this case, there still exists human context understanding, but limited to what is displayed to a subject. This "blind context" makes third-party GTs a fairer benchmark strategy for automatic algorithms [13].

In this paper, we use a third-party GT based on multiple-subject judgement that takes into account the fact that different people might generate different results. Scenes for each video were found by approximately twenty subjects using a GUI that displayed a key-frame-based video summary (no real videos were displayed). A very general statement about the clustering task, and no initial solution were provided to the subjects at the beginning of the process. The final GT set consists of about 400 human segmentations.

4.2 Performance measures

The performance measures that we consider are (i) the number of clusters selected by the algorithm and (ii) the shots in errors (SIE). For the number of clusters, we report the value we obtain and compare it with the numbers provided by humans. 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) varies considerably from one video to another.

	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

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

4.3 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) described in [4], as well as with the performance of humans. 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. 6 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 minimum and maximum values that the spread of performances is very high, given the performance measure. Secondly, the spectral method is performing better than the PHC, as can be seen from the median and minimum value, and approximately as well as the humans.
Fig. 6 displays the results obtained for each video. First, in Fig. 6(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 ground-truth. The spread of the cluster numbers in the ground-truth is represented by the blue bar (plus or minus one standard deviation). Note that the video have been ordered according to their number of shots. The detected cluster numbers are in good accordance with the GT, though slightly underestimated. Fig. 6(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 differents measures (minimum in red, median in blue, and maximum in green). The median performance of our algorithm is better than the average human in 8 cases and worse in 6 cases. Notice that in 25% of the cases, our algorithm provides a segmentation that also exists in the ground-truth.

Two examples of the generated clusters are shown in Fig. 7. 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.

4.4 Comparison of the different criteria

Fig. 8 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 relatively large range of thresholds (in any case, better than the probabilistic hierarchical clustering algorithm). We also used the MSE (cf Eq. (3)) as a criterion, but could not obtain good results with it.

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. We have shown in our experiments that the eigengap measure could indeed be used to estimate this number. The algorithm was applied to a six-hour home video database, and the results are favorably compared to existing techniques as well as human performance.

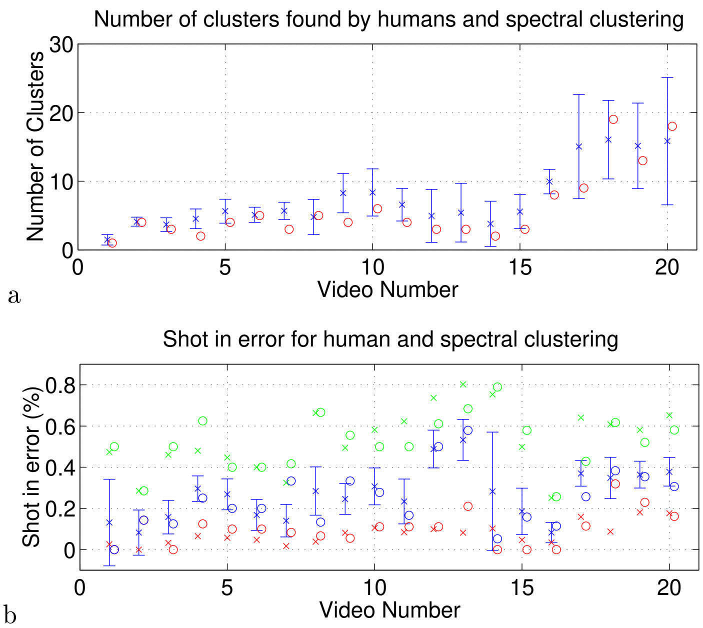

Figure 6: (a) Determination of number of clusters. (b) Percentage of shot in error. The blue bar indicates the spread of the human performances of the SIE$_{med}$ value.

Figure 7: Example of shot clustering (a) Video 16 (b) Video 8. Only one keyframe of each shot is displayed.

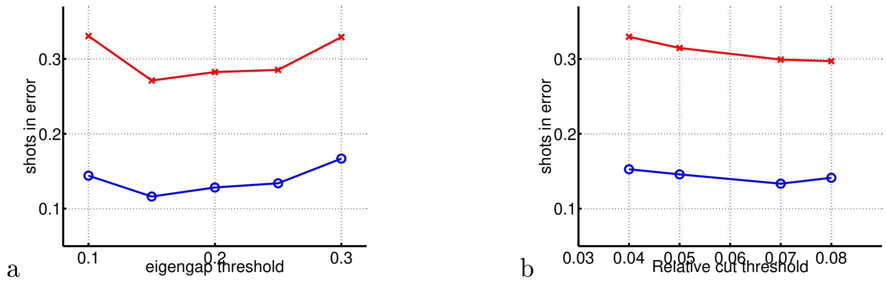

Figure 8: Variation of the average of percentage of shots in error (average of the median in red, of the min in blue) for the different criterion in function of their threshold a) the eigengap threshold (ranging from 0.1 to 0.3) b) the relative cut threshold (ranging from 0.04 to 0.08).

6 Acknowledgments

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. 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.

References

[1] C. Alpert, A. Kahng, and S.Z Yao, "Spectral partitioning : The more eigenvectors, the better," *Discrete Applied Math*, , no. 90, pp. 3–26, 1999.

[2] F. R.K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

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

[4] 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*, Tokyo, Aug. 2001.

[5] G. Iyengar and A. Lippman, "Content-based browsing and edition of unstructured video," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, New York City, Aug. 2000.

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

[7] 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.

[8] R. Lienhart, "Abstracting Home Video Automatically," in *Proc. ACM Multimedia Conf.*, Orlando, Oct. 1999. pp. 37-41.

[9] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *Proc. AISTATS*, Florida, 2001.

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

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

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

[13] A. Savakis, S. Etz, and A. Loui, "Evaluation of image appeal in consumer photography," in *Proc. SPIE/IS&T Conf. on Human Vision and Electronic Imaging V*, San Jose, CA, Jan. 2000.

[14] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, , no. 10, pp. 1299–1319, 1998.

[15] G.L. Scott and H.C. Longuet-Higgins, "Feature grouping by relocalisation of eigenvectors of the proximity matrix," in *British Machine Vision Conf.*, 1990, pp. 103–108.

[16] 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.

[17] 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.

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