Video shot clustering and summarization through dendrograms

S. Benini, A. Bianchetti, R. Leonardi, and P. Migliorati,

Department of Electronics for Automation - SCL
University of Brescia, Via Branze 38, I-25123, Brescia, Italy

Abstract In the context of analysis of video documents, effective clustering of shots facilitates the access to the content and helps in understanding the associated semantics. This paper introduces a cluster analysis on video shots which employs dendrogram representation to produce hierarchical summaries of the video document. Vector quantization codebooks are used to represent the visual content and to group the shots with similar chromatic consistency. The evaluation of the cluster codebook distortions, and the exploitation of the dependency relationships on the dendrogram, allow to obtain only a few significant summaries of the whole video. Finally the user can navigate through summaries and decide which one best suites his/her needs for eventual post-processing. The effectiveness of the proposed method is demonstrated, on a collection of different video programmes, in term of metrics that measure the content representational value of the summarization technique.

1 Introduction

Recent advances in technology have made terrific amount of multimedia information available to the normal users. Meanwhile, the needs for efficient retrieval of desired information has led to the development of algorithms that enable automated analysis of large multimedia database.

Regarding videos, the segmentation into shots and the key-frame extraction are commonly considered as the prior steps for performing effective content-based indexing, browsing, summarization and retrieval. However, a shot separation often leads to a far too fine segmentation of the sequence. So, building upon this, efforts are invested towards grouping shots into more compact structures sharing common semantic threads. Providing a compact representation of a video sequence, clusters of shots results to be useful for generating static video summaries. Methods dealing with clustering of shots are widely reported in literature. In [11] and [8] approaches based on a time-constrained clustering have been presented. Visual similarity between shot key-frames has been measured by color pixel correlation in [11], or by block matching in [5]. Lately, spectral methods [7] resulted to be effective in capturing perceptual organization features. Video summarization techniques using clusters of shots can be found in [4], while other recent methods use graph theory [1] and curve splitting [2].

The paper aims first to propose a tree-structured vector-quantization codebook as an effective low-level feature for representing each video shot content. Then it is shown how shots with long-term chromatic consistency are grouped together. The proposed distortion measure and the use of dendrogram representation allow to stop the clustering process only on few significant levels. The goal of such analysis is to generate hierarchical summaries of the video, so providing the user with a fast non-linear access to the desired material. The obtained results can be useful for further post-processing, such as semantic annotation and story unit detection [5].

The paper is organized as follows: in section 2 vector quantization on shots is introduced; sections 3 and 4 present an effective shot-clustering algorithm which allows the generation of the hierarchical summaries; finally, in sections 5 and 6 experimental results and conclusions are discussed.

2 Visual Low-level Feature

Starting from an already given shot decomposition, each shot is further analyzed in order to determine its *vector quantization* codebook on color information.

2.1 Tree-Structured Vector Quantization

First the central frame of each shot is chosen, even if the procedure is functionally scalable to the case when more than one frame per shot are needed. Then a *tree-structured vector quantization (TSVQ)* codebook is designed so as to reconstruct each frame with a certain distortion with respect to the original one. In the specific, after having been sub-sampled in both directions at *QCIF* resolution, and filtered with a denoising gaussian filter, every frame is divided into non overlapping blocks of $N \times N$ pixels, scanning the image from left to right and top to bottom. All blocks are then represented using the *LUV* color space and used as the training vectors to a *TSVQ* algorithm [3] by using the *Generalized Lloyd Algorithm (GLA)* for codebooks of size $2^n (n = 0, 1, 2, \ldots)$. Each increase in the size of the codebook is done by splitting codewords from the next smallest codebook (perturbed versions of the old most populated codewords). The *GLA* continues to run until a pre-determined maximum distortion (or a maximum codebook size) is reached. Then, an attempt is made to reduce the number of codewords in the interval $[2^n, 2^{n-1}]$ without exceeding the pre-determined distortion. Finally the algorithm returns the *TSVQ* codebook final dimension for each investigated shot. Note that the dimensions of each codebook could be different among shots. The objective of this approach is to produce codebooks for each key-frame with close distortion values, so as to allow for a further comparison between different codebooks.

2.2 A Measure of Shot Similarity

The similarity between two shots can be measured by using the codebooks representing the shots.

Let S_i be a shot, and let K_j be a generic codebook; when a vector $s \in S_i$ is quantized to a vector $k \in K_j$, a quantization error occurs. This error may be measured by the average distortion $D_{K_j}(S_i)$, defined as:

$$D_{K_j}(S_i) = \frac{1}{V_i} \sum_{p=0}^{V_i - 1} \|s_{ip} - k_{jq}\|^2 \qquad (1)$$

where V_i is the number of vectors s_{ip} of S_i (the number of $N \times N$ blocks in the shot), and k_{jq} is the code vector of K_j with the smallest euclidean distance from s_{ip}, *i.e.*:

$$q = \arg\min_z \|s_{ip} - k_{jz}\|^2 \qquad (2)$$

Furthermore, given two codebooks (K_i and K_j), the value $|D_{K_i}(S_i) - D_{K_j}(S_i)|$ can be interpreted as the distance between the two codebooks, when applied to shot S_i. A symmetric form of the similarity measure used in [9] between shot S_i and shot S_j can, thus, be defined as:

$$\phi(S_i, S_j) = |D_{K_j}(S_i) - D_{K_i}(S_i)| + |D_{K_i}(S_j) - D_{K_j}(S_j)| \qquad (3)$$

where $D_{K_i}(S_i)$ is the distortion obtained when shot S_i is quantized using its associated codebook. The smaller ϕ is, the more similar the shots are. It should be noticed that the similarity is based on the cross-effect of the two codebooks on the two shots. In fact, it may happen that the majority of blocks of one shot (for example S_i), can be very well represented by a subset of codewords of codebook K_j representing the other shot. Therefore K_j can represent S_i with a small average distortion, even if the visual content of the two shots is only partly similar. On the other hand, it is possible that codebook K_i doesn't lead to a small distortion when applied to S_j. So cross-effect of codebooks on the two shots is needed.

3 Shot Clustering

Once a shot similarity measure has been defined, the next step is to identify clusters of shots. Suppose we have a sequence with N_s shots. At the beginning of the iterative process each shot belongs to a different cluster (level-N_s). At each step the algorithm merges the two most similar clusters, where similarity between clusters C_i and C_j, $\Phi(C_i, C_j)$, is defined as the average of the similarities between shots belonging to C_i and C_j, *i.e.*:

$$\Phi(C_i, C_j) = \frac{1}{N_i N_j} \sum_{S_i \in C_i} \sum_{S_j \in C_j} \phi(S_i, S_j) \qquad (4)$$

where N_i (N_j) is the number of shots of cluster C_i (C_j).

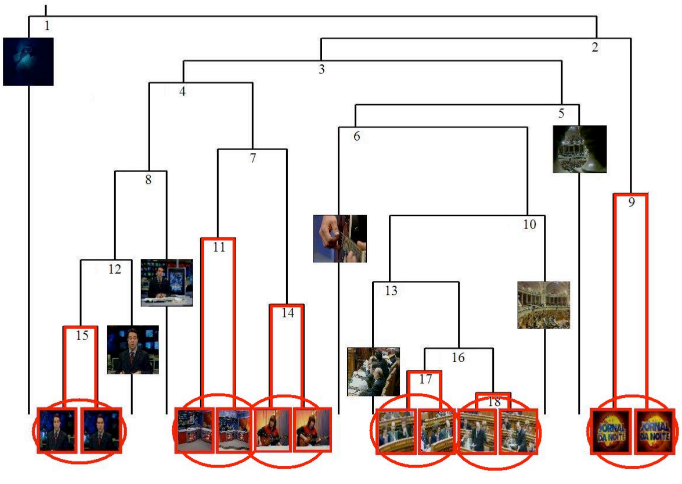

Fig. 1 Dendrogram with the *leading* clusters highlighted.

3.1 Dendrogram Representation of Clustering Process

The results of the clustering process can be graphically rendered by a dendrogram plot. A dendrogram consists of many ⊓-shaped lines connecting objects in a binary tree. For our scope, a dendrogram represents the whole clustering process of N_s shots, from the level-N_s (each cluster containing one single shot) up to level-1, where a single cluster contains all the shots of the sequence (as in Figure 1). Moreover the height of each ⊓-branch represents the similarity between the two clusters being connected, so that low (high) connections correspond to similar (dissimilar) merged clusters. Through a dendrogram, it is therefore possible to follow the clustering process at each iteration step, every level providing a different representation of the video sequence.

4 Hierarchical Summaries

Observing the clustering process at each level is almost of no use for a multimedia content consumer, due to the large number of levels. Our principal aim is to automatically determine few significant levels (among all generated ones) able to offer the user semantically significant *summaries* of the observed video sequence.

A news program, for example, can be summarized at various levels of granularity, but only very few of them are *semantically* significant. In our example, the top level *summary* can be the whole programme; on a lower level, it may be helpful to discriminate between the "studio" shots and the "reports"; then, inside the "studio", to distinguish between the "anchorman" shots and ones with the "guest", and so on. With such a hierarchic scheme, the video content can be expressed progressively, from top to bottom in increasing levels of granularity.

4.1 Leading *Cluster Analysis on Dendrogram*

Looking at the bottom of the dendrogram, it is easy to single out the *leading* clusters as the ones originally formed by the fusion of two single shots (see Figure 1). Each time a *leading* cluster merges with another one, it propagates its property of being a *leading* cluster to the new formed one. Since at each merging step at least one of the two merged clusters is a *leading* cluster, only by tracking the evolution of the *leading* clusters it is possible to perform a complete analysis of the dendrogram.

Let C_k^* be a *leading* cluster, and let us call $C_k^*(i)$ the cluster at level-i, where $i \in I = \{N_s, N_s - 1, \ldots, 1\}$. Tracking the evolution of C_k^* from level-N_s to level-1 it is possible to evaluate the cluster's internal distortion introduced as the cluster grows bigger. In particular, let $I_k^* = \{i_1, i_2, \ldots, i_n\} \subseteq I$ be the sub-set of levels of I in which $C_k^*(i)$ actually takes part in a merging operation, the internal distortion of cluster C_k^* at level i_j can be expressed as:

$$\Psi(C_k^*(i_j)) = \Phi(C_k^*(i_{j-1}), C_h) \tag{5}$$

where C_h is the cluster (which can be *leading* or not) merged with C_k^* at level-i_j (*i.e.* the internal distortion is given by the cluster similarity between the two clusters being merged).

4.2 Summaries

Following the internal distortion of each *leading* cluster C_k^* on each level belonging to I_k^*, it is possible to automatically determine which few levels are semantically significant to be considered *summaries*. Observing the internal distortion of each *leading* cluster, $\Psi(C_k^*)$, and setting a threshold on its discrete derivative

$$\Psi'(C_k^*(i_j)) = \Psi(C_k^*(i_j)) - \Psi(C_k^*(i_{j-1})) \tag{6}$$

the user is able to stop the *leading* cluster C_k^* growth at levels $D_k^* = \{i_{d_1}, i_{d_2}, \ldots, ..., i_{d_n}\} \subseteq I_k^*$. These levels indicate meaningful moments in the growth evolution of C_k^* (*i.e.* when the height of the ⊓-branch of the dendrogram varies significantly with respect to the previous steps). Once computed all the sets D_k^* for each C_k^*, all the significant *summaries* for the investigated sequence can be obtained. The number of the available *summaries* is given by $w = \max_k |D_k^*|$, where w is the maximum cardinality among sets D_k^*. If we want to obtain the m^{th} *summary* ($m = 1, 2, \ldots, w$), the algorithm lets each *leading* cluster C_k^* grow until $C_k^*(i_{d_m})$. Since at each level $i_j^k \in I_k^*$ with $i_1^k \leq i_j^k \leq i_{d_m}^k$ the cluster C_k^* merges with another cluster C_h, if C_h is a *leading* cluster, the condition $i_{d_m}^h \leq i_j^k$ must be met. This condition verifies the dependency condition between the merging clusters, *i.e.* the case when the cluster C_h has been already arrested at a previous level with regard to that of the merging with C_k^*. If the condition is not fulfilled, the growth of C_k^* must be stopped iteratively at level $i_{(j-1)}^k$ until the dependency condition is verified. The resulting set of all the obtained clusters determines the m^{th} *summary* of the video.

5 Experimental Results

Applying the scheme for example to the *Portuguese News* sequence, *summaries* can be parsed into a hierarchical structure, each level containing a compact overview of the video at different granularity. In Figure 2, the top (5^{th}) *summary* is a unique cluster containing all the shots; the 4^{th} *summary* distinguishes between the "news programme" cluster and the opening and closing "jingle". Then, the 3^{rd} *summary* presents the "report" shots, the "studio", and the "jingle" ones in separated clusters. Then, the hierarchical decomposition continues on lower *summaries* at increasing levels of granularity, allowing the user to evaluate the quality of the decomposition with respects to his/her own desires. After that, he/she can recursively descend the hierarchy until a satisfactory result is achieved. This structure provides a more reliable and fast access to video content material: it is better than the manual manner in terms of efficiency, and more accurate than a completely automatic scheme.

In order to objectively evaluate the cluster decomposition accuracy, we carried out some experiments using video segments from one news programme, three feature movies, two soap operas, one music show, one miscellaneous programme and one cartoon for a total of about 4

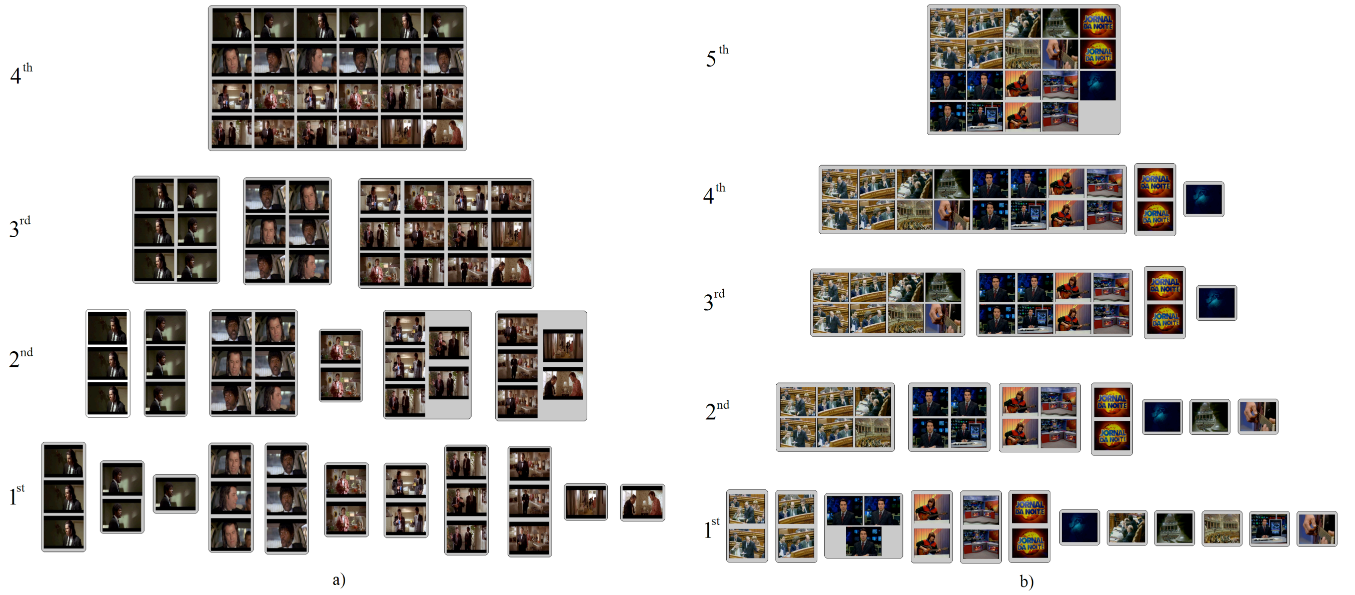

Fig. 2 Hierarchical summaries for a) *Pulp Fiction* movie and the b) *Portuguese News* programme.

hours of video. Using the cluster validity analysis as in [6] the optimal *summary* is chosen, and building upon this, a segmentation into *Logical Story Unit (LSU)* [5] is obtained. To evaluate our performance appropriately we use the *Coverage* and *Overflow* criteria proposed in [10]. The measurements are presented in Table 1, where low values of *Overflow* and the high scores of *Coverage* reveal the good performance of the proposed clustering algorithm at the optimal *summary* level.

Table 1 Detected *LSU* in terms of *Coverage* and *Overflow*.

Video (genre)	Length	*Cover.*	*Overf.*
Portuguese News (news)	47:21	65.1%	9.4%
Notting Hill (movie)	30:00	82.8%	0.0%
A Beautiful Mind (movie)	17:42	92.2%	0.0%
Pulp Fiction (movie)	20:30	85.6%	2.3%
Camilo & Filho (soap)	38:12	90.0%	0.0%
Riscos (soap)	27:37	74.9%	4.7%
Music Show	10:00	66.1%	10.7%
Misc. (basket/soap/quiz)	38:30	99.4%	0.0%
Don Quixotte (cartoon)	15:26	67.5%	7.4%

6 Conclusions

This work describes the issue of clustering shots by using a tree-structured vector quantization and a dendrogram representation for clusters. The proposed hierarchical scheme is suitable for expressing video content progressively at increasing levels of granularity. Resulting summaries, obtained from a large test set, provide the user with a compact representation of video content and a fast access to the desired video material for eventual post-processing.

References

1. H. S. Chang, S. S. Sull and S. U. Lee, “Efficient video indexing scheme for content based retrieval,” IEEE Trans. on CSVT, Vol. 9, No. 8 , Dec 1999.
2. D. DeMenthon, V. Kobla and D. Doermann, “Video Summarization by curve simplification,” CVPR’98, Santa Barbara, USA, 1998.
3. A. Gersho and R. M. Gray, “Vector Quantization and Signal Compression”, Kluwer Acad. Publishers, 1992.
4. Y. Gong and X. Liu, “Video summarization and retrieval using Singular Value Decomposition,” ACM MM Systems Journal, Vol. 9, No. 2, pp. 157-168, Aug 2003.
5. A. Hanjalic and R. L. Lagendijk, “Automated high-level movie segmentation for advanced video retrieval systems,” IEEE Trans. on CSVT, Vol. 9, No. 4, June 1999.
6. A. Hanjalic and H. J. Zhang, “An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis,” IEEE Trans. on CSVT, Vol. 9, No. 8, Dec 1999.
7. J-M. Odobez, D. Gatica-Perez and M. Guillemot, “Video shot clustering using spectral methods”, CBMI’03, Rennes, France, Sept 2003.
8. E. Sahouria and A. Zakhor, “Content analysis of video using principal components,” IEEE Trans. CSVT, Vol. 9, No. 8, pp. 1290-1298, 1999.
9. C. Saraceno and R. Leonardi, “Indexing audio-visual databases through a joint audio and video processing”, Int. Journal of Imaging Systems and Technology, Vol. 9, No. 5, pp. 320-331, Oct 1998.
10. J. Vendrig and M. Worring, “Systematic evaluation of logical story unit segmentation,” IEEE Trans. on Multimedia, Vol. 4, No. 4, Dec 2002.
11. M. M. Yeung and B.-L. Yeo, “Time-constrained clustering for segmentation of video into story units,” ICPR’96, Vol.III-Vol.7276, p.375, Vienna, Austria, Aug 1996.