EXTRACTION OF SIGNIFICANT VIDEO SUMMARIES BY DENDROGRAM ANALYSIS

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

DEA-SCL, University of Brescia, Via Branze 38, I-25123, Brescia, Italy

ABSTRACT

In the current video analysis scenario, effective clustering of shots facilitates the access to the content and helps in understanding the associated semantics. This paper introduces a cluster analysis on 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 dendrograms, 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 by testing it on a collection of video-data from different kinds of programmes. Results are evaluated in terms of metrics that measure the content representational value of the summarization technique.

Index Terms— Vector quantization, clustering methods

1. INTRODUCTION

As long as we are entering the multimedia era, tremendous amounts of video information have been made 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 video databases.

If the field of video analysis, the segmentation into shots and the key-frame extraction are now 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.

Clustering methods based on a time-constrained approach have been presented in [9] and [7]. Visual similarity between shots has been measured between key-frames by means of color pixel correlation in [9], or by block matching in [5]. Lately, spectral methods [6] resulted to be effective in capturing perceptual organization features. Video summarization techniques using clusters of shots can be found in [4], while other recent summarization methods use graph theory [1] and curve splitting [2].

The principal aim of the paper is to first 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 document, which provide the user with a fast non-linear access to the desired visual 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. LOW-LEVEL FEATURE

Supposing that the video has been already decomposed into shots, a further low-level feature analysis is performed, determining each shot *vector quantization* codebook on color.

2.1. Vector Quantization Codebook

The central frame of each shot is first chosen, even if the procedure is functionally scalable to the case when more than one frame per shot are needed. Then, for each extracted frame, 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 pop-

ulated 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 limit. Finally the algorithm returns the code vectors and the *TSVQ* codebook final dimension for each investigated shot. Note that the dimensions of each codebook could be different for each single shot. 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. Shot Similarity

The similarity between two shots can be measured by using the codebooks computed on respective 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 quantization 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 shot 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.*:

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

where $k_{jz} \in K_j$. 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 [8] 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 to obtain a sound similarity measure.

3. CLUSTERING OF VIDEO SHOTS

Now that is possible to evaluate similarity between shots, 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 new iteration, the algorithm sets to merge 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).

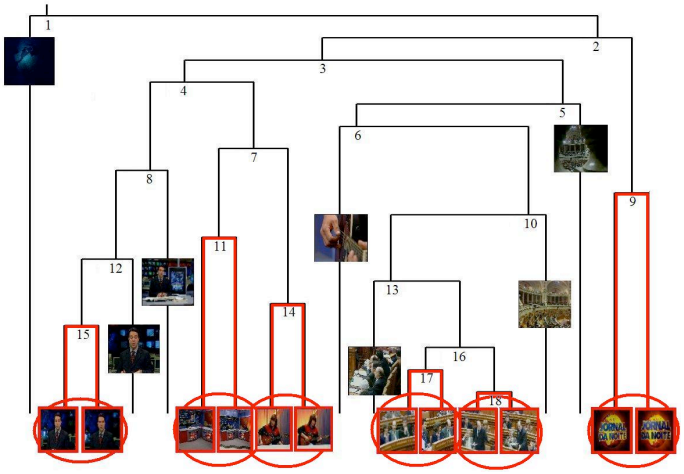

Fig. 1. Dendrogram with the *leading* clusters highlighted.

3.1. Dendrograms

The ouput resulting from a 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 SUMMARIZATION

Due to the large number of merging steps, observing the clustering process at each level is almost of no use for a multimedia content consumer. 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 movie, for example, can be summarized at various levels of granularity, but only very few of them are *semantically* significant. For example, the top level *summary* can be the whole programme; on a lower level, it may be helpful to discriminate between the "outdoor" and the "indoor" shots; then, inside the "indoor", to distinguish among the different settings, 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* Clusters in Dendrogram

Observing the bottom level of a 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). In the upper branches of the dendrogram, 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 following 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 $i \in I = \{N_s, N_s - 1, \ldots, 1\}$. Following 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. *Extracted Summaries*

Looking at the evolution of 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 evolution of C_k^*, *i.e.* when its visual content significantly changes (and the height of the $\sqcap$-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.

Fig. 2. Hierarchical summaries for the movie *Pulp Fiction*.

5. EXPERIMENTAL RESULTS

Applying this scheme, for example to a short *Pulp Fiction* sequence, *summaries* can be parsed into a hierarchical structure, each level containing a compact overview of the video at different granularity. Looking at Figure 2, the top (4^{th}) *summary* is a unique cluster containing all the shots; the 3^{th} *summary* distinguishes among three different settings. Then, the hierarchical decomposition continues on lower *summaries* at increasing levels of granularity, allowing the user to evaluate the quality of the decomposition with respect to his/her own desires. After that, he/she can recursively descend the hierarchy until a satisfactory result is achieved.

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 miscellaneous programme and one cartoon for a total time of about 4 hours of video. To judge the quality

Video	Summary	*C (%)*	*P (%)*	Video	Summary	*C (%)*	*P (%)*
Portuguese News (news)	1^{st} *(217 clusters)*	54.4	86.1	***Camilo & Filho (soap)***	1^{st} *(68 clusters)*	51.4	95.6
476 shots	2^{nd} *(117 clusters)*	75.4	68.3	***140 shots***	2^{nd} *(35 clusters)*	75.0	80.0
47:21	3^{rd} *(81 clusters)*	82.9	49.3	***38:12***	3^{rd} *(23 clusters)*	83.6	73.9
Notting Hill (movie)	1^{st} *(201 clusters)*	53.1	88.1	***Riscos (soap)***	1^{st} *(192 clusters)*	54.6	88.0
429 shots	2^{nd} *(111 clusters)*	74.1	81.1	***423 shots***	2^{nd} *(107 clusters)*	74.7	75.7
30:00	3^{rd} *(69 clusters)*	83.9	73.9	***27:37***	3^{rd} *(65 clusters)*	84.6	69.2
A Beautiful Mind (movie)	1^{st} *(98 clusters)*	53.4	93.1	***Misc. (basket/soap/quiz)***	1^{st} *(94 clusters)*	51.8	93.6
210 shots	2^{nd} *(60 clusters)*	71.4	81.1	***195 shots***	2^{nd} *(47 clusters)*	75.9	82.9
17:42	3^{rd} *(41 clusters)*	80.4	69.1	***38:30***	3^{rd} *(30 clusters)*	84.6	80.0
Pulp Fiction (movie)	1^{st} *(91 clusters)*	48.3	94.5	***Don Quixotte (cartoon)***	1^{st} *(96 clusters)*	48.9	83.3
176 shots	2^{nd} *(54 clusters)*	69.3	87.0	***188 shots***	2^{nd} *(42 clusters)*	77.7	54.8
20:30	3^{rd} *(35 clusters)*	80.1	71.4	***15:26***	3^{rd} *(19 clusters)*	89.9	31.6

Table 1. For each video the first three *summaries* are presented in terms of *Compression C* and *Precision P*.

of the detected results, the following rule is applied:
*"A cluster belonging to the m^{th} summary is judged to be correctly detected if and only if **all** shots in the current cluster share a common semantic meaning. Otherwise the current cluster is judged to be falsely detected"*.

Clustering *Precision* P is used for performance evaluation, where P is defined as:

$$P = \frac{\text{\# rightly detected clusters}}{\text{\# detected clusters}} \tag{7}$$

For example in the 1^{st} summary of *Pulp Fiction* (see Figure 2) we have a cluster containing only shots sharing the semantics "J. Travolta in a car". In this case the cluster is considered as correctly detected. It has to be pointed out that, if we look to a specific shot in the final hierarchy, the semantics of the clusters containing the shot changes depending on the summarization level. In our example, if we climb, on a higher level of abstraction, to the 2^{nd} summary, the shots with J. Travolta in the car are clustered together with those showing S.L. Jackson in the same car, so that the shared semantics among all shots would be described as a more general "Man in a car".

Clearly, at the top level summary (all shots belonging to one cluster), the cluster detection precision would be 100%. And the same happens if we treat each shot as a cluster. Hence, in order to discriminate the representative power of a given summary, another measure is needed to express the *Compression* factor of the summary, i.e.:

$$C = 1 - \frac{\text{\# detected cluster}}{\text{\# shot in the video}} \tag{8}$$

The experimental results of cluster detection at different summarization levels for all our video data set are given in Table 1 in terms of *Precision P* and *Compression C*.

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.

7. 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 Academic 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] J-M. Odobez, D. Gatica-Perez and M. Guillemot, "Video shot clustering using spectral methods", CBMI'03, Rennes, France, Sept 2003.

[7] E. Sahouria and A. Zakhor, "Content analysis of video using principal components," IEEE Trans. CSVT, Vol. 9, No. 8, pp. 1290-1298, 1999.

[8] 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.

[9] 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.