IDENTIFYING VIDEO CONTENT CONSISTENCY BY VECTOR QUANTIZATION

Sergio Benini[1], *Li-Qun Xu*[2], *Riccardo Leonardi*[1]

[1] Università di Brescia, DEA, via Branze 38, 25123 Brescia, Italy +39 030 3715 434
[2] BT Research and Venturing, Adastral Park, Ipswich IP5 3RE, UK +44 1473 648608

ABSTRACT

Many post-production videos such as movies and cartoons present well structured story-lines organized in separated visual scenes. Accurate grouping of shots into these logical segments could lead to semantic indexing of scenes for interactive multimedia retrieval and video summaries. In this paper we introduce a novel shot-based analysis approach which aims to cluster together shots with similar visual content. We demonstrate how the use of codebooks of visual codewords (generated by a vector quantization process) represents an effective method to identify clusters containing shots with similar long-term consistency of chromatic compositions. The clusters, obtained by a *single-link* clustering algorithm, allow the further use of the well-known *scene transition graph* framework for *logical story unit* detection and pattern investigation.

1. INTRODUCTION

Shot segmentation and keyframe extraction are commonly considered as the prior steps for performing content-based video indexing, browsing and summarization tasks. Building upon these, recent research activities have been focusing on developing techniques towards higher level semantic content abstraction with respect to a shot-based decomposition. In particular, efforts are invested towards grouping shots into logical units that share a common semantic thread.

Some methods dealing with automatic high-level movie segmentation can be found in literature. In [6], [9], [10] and [11] various approaches based on a time-constrained clustering are presented. In particular in [9] interesting results are provided by first measuring visual similarity between shots by means of pixel-color correlation between their respective keyframes, then followed by the use of predefined memory models for recognizing patterns inside the story. In [3] the concept of *Logical Story Unit* (*LSU*) was introduced as *"a series of temporally contiguous shots, characterized by overlapping links that connect shots with similar visual content."* The dissimilarity between any two shots is estimated by measuring the correlation between keyframes by block matching. Yet another group of algorithms chooses to measure probable scene boundaries by calculating a short term memory-based model of shot-to-shot 'coherence' as proposed in [4] and [7].

In this current work we present a novel and effective method to group shots into *logical story units* (*LSU*) starting from an MPEG-1 video stream, using a "shot-to-shot" similarity measure based on vector quantization (*VQ*) codebooks of the visual content of their keyframes. We use these codebooks so as to effectively represent clusters of shots (as proposed in [6]), allowing then to establish a "cluster-to-cluster" similarity. The proposed algorithm effectively clusters basic video segments into compact and visually coherent structures that lend themselves easily to further processing. In fact, using an already proposed method such as the *scene transition graph (STG)* [9], it is possible to find out the fundamental elements of the semantic structure (*LSU*) and eventually recognize patterns (such as dialogues) and repeats (distant similar shots) along the video.

The paper is organized as follows. The next section discusses shot-based visual content representation and similarity metric. Sections 3 and 4 are devoted to the presentation of the shot clustering algorithm and the technique for further *LSU* detection, respectively. The experimental results are discussed in Section 5, while paper conclusions are drawn in Section 6.

2. VISUAL CONTENT REPRESENTATION

Assuming that the segmented video shots are already given, the video content of each shot is first represented by means of a *Vector Quantization* (*VQ*) method [2]. This means that a visual codebook is built starting from the color information of one or more selected keyframes [6]. In our experiments, the central frame of each shot is chosen, though the procedure is designed to be functionally scalable, being easily adapted to the case when more than one keyframe per shot is needed to gain a proper representation of the visual content.

2.1. VQ codebook for shots

The selected keyframe is decoded as a still 352x288 image and represented in the *LUV* color space. The image

is then sub-sampled by a factor *2* in both directions (*QCIF* resolution) and further subjected to a de-noising Gaussian filtering. The processed keyframe is subdivided into blocks of 4x4 pixels in order to exploit spatial color correlation. For each block, the three *LUV* components of each pixel are concatenated in a raster scan order to form a *D*–dimensional vector (where *D=48=3x16*). These vectors, *1584* in total for a *QCIF* sized image, constitute the training set for the *VQ* codebook generation process. The procedure starts with a randomly generated codebook, which is then iteratively refined by the *Generalized Lloyd Algorithm* [2]. Once the algorithm converges according to a predefined distortion criterion, the centroids (or codewords) $\{\mu_j\}$ of the finally obtained Voronoi partitions represent good candidate features to reproduce the color content of the shot.

The codebook thus generated for each visual shot contains: *i)* the c_j codewords of *D* dimension each, which are the centroids μ_j of each Voronoi partition in the final codebook; *ii)* the variances σ_j of the codewords; *iii)* the normalized weights w_j which estimate the probability mass associated to each region. In our studies, the dimension of the codebook is set to *100* codewords.

2.2. VQ codebook distance metric

After the codebook for each shot is created, a "shot-to-shot" similarity measure between shot *A* and *B* can be defined in terms of the distance between their respective codebooks. An effective method can be derived from the fast earth mover's distance computation found in [1]. First, each codebook vector of shot *A*, $y_A=\{y_1,...,y_C\}$, is compared with those of shot *B*, $z_B=\{z_1,...,z_C\}$, to produce a distance matrix $d_{i,j}=d\{y_i,z_j\}$:

$$d_{i,j}=\frac{1}{D}\left[\sum_{h=1}^{D}\frac{1}{2}(\mu_{ih}-\mu_{jh})^2+\frac{1}{2}(\sigma_{ih}-\sigma_{jh})^2\right]$$

with *i,j=1,...,C*. Then, the *VQ codebook distance metric* between shots *A* and *B* can be defined as:

$$VQ\_Dist(A,B)=\sum_{i=1}^{C} w\cdot\min_j(d_{i,j})$$

where the weight *w* is determined as $w=max(w_i,w_j)$.

3. SHOT CLUSTERING ALGORITHM

Having defined the "shot-to-shot" distance metric, the next step in our proposed content processing chain is to group similar shots into compact clusters.

3.1. Time unconstrained analysis

First, the clustering is conducted over a global view of the whole video segment, relying only on the shot visual content without considering any temporal constraints. The objective of such a time-unconstrained approach is twofold: on one hand, it does not set an *a priori* time limit for an *LSU* extent (which is an issue in [9]) and on the other hand, it can be useful for certain retrieval purposes, like user defined queries searching for repeats (e.g., a viewer may like to see all the shots set at one specific location in a movie, even if distant in time).

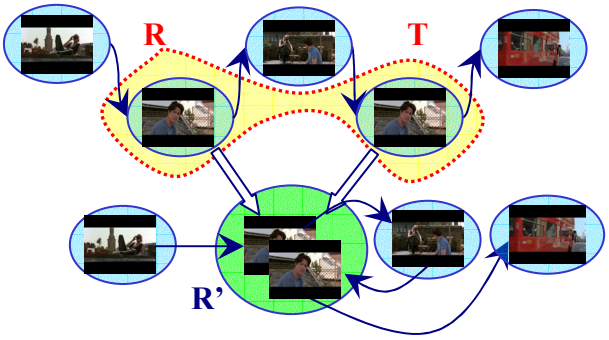

Figure 1: Illustration of a merge operation in clustering

At the start, each cluster only contains a single shot, so forming the *Original Shot Graph (OSG)*, in which nodes correspond to single shots, and edges indicate the transitions between shots. The *VQ codebook "shot-to-shot" distances* can be computed between all shots along the timeline, so as to explore exhaustively the visual similarities of the entire video.

At each step, the algorithm merges a reference cluster *R* with its visually most similar test cluster *T* (*i.e.* the one having the minimal *VQ distance metric*) to form a new cluster *R'*. All the shots belonging, respectively, to *R* and *T*, then become the shots of the new cluster *R'*, and all transitions from/to *R* and *T* are properly updated to keep the correct temporal flow of the story (Figure 1 shows an example of such a merge operation). For the new combined cluster *R'*, a new *VQ* codebook is now needed to represent its visual content.

3.2. VQ codebook update

As a simple extension of the case for a single shot, a *VQ codebook* can also be created to represent a cluster of shots. The same codebook generation process used before is applied, but since the new cluster contains more than one shot, the keyframes of all the shots belonging to the cluster contribute equally to the training set.

Once the codebook of the cluster *R'* is generated, the "cluster-to-cluster" distances can now be computed between *R'* and all the other clusters before proceeding to the next iteration.

3.3. Online statistical analysis on VQ distortion

It should be noted that while the *VQ codebook* generated at the *OSG* step is specific for each shot, the *VQ codebook*, which represents the visual content of a cluster with multiple shots, usually becomes less specific. This

means that, for each shot, the best representing *VQ codebook* is the one prior to the first merging. From then on, the distortion of representing the original shot with its cluster *VQ codebook* is likely to increase.

So, at each iteration k the newly formed cluster R_k' introduces a *VQ distortion* with respect to the *OSG* step, where each cluster contains only a single shot. This distortion can be computed, using the proposed *VQ distance,* as the sum of the distances between the codebook of the newly formed clusters R_k' and the codebooks of its shots computed at the *OSG* step. The accumulated distortion *VQ_Err* introduced from the beginning of the process until step k is:

$$VQ\_Err(k) = \sum_{j=1}^{k} \sum_{s_i \in R_j} VQ\_Dist(s_i, R_j')$$

With the increase in a cluster's size, its *VQ* codebook is at the risk of losing the specificity in representing any of its constituent shots. To prevent this degenerative process, a statistical analysis is performed on the distortion generated by the latest merging step. At each iteration k, we compute the first derivative of the *VQ* distortion as,

$$\Delta VQ\_Err(k) = [VQ\_Err(k) - VQ\_Err(k-1)] \quad ,$$

together with its mean μ and standard deviation σ with respect to the previous steps. Specifically, observing that after some initial steps (in which very similar shots collapse in the same clusters) the ΔVQ_Err tends to be gaussianly distributed, if:

$$\Delta VQ\_Err(k) > \mu_j\{\Delta VQ\_Err(j)\} + thrs \cdot \sigma_j\{\Delta VQ\_Err(j)\}$$

with $j=1,2,...k-1$, and $thrs \in [2,3]$, this means that the distortion introduced by the newly formed cluster is too large and that the *VQ* codebook fails to represent all the cluster components. In such a case, this latest merging between cluster R and T is discarded. Besides, these clusters are since locked, being excluded from any future clustering process and a new reference and test cluster, currently unlocked, are selected according to the next minimal value of *VQ distance metric*. The above iteration process is repeated until no more unlocked clusters are available for merging. Experimental trials demonstrated that in order to prevent a cluster codebook from growing too much and losing specificity, it is useful to lock the cluster at a certain size (*e.g.,* 12-15 shots in practice).

3.4. Intra-cluster time analysis

The clusters generated from the time-unconstrained analysis provide the first level representation of video content hierarchy. Each cluster now contains shots with high visual similarity to each other even if they may not be temporally adjacent, which, as discussed before, is already desirable for certain retrieval purposes.

However, in order to separate one *LSU* from another, it is necessary that further analysis looks into the temporal locality of shots within each cluster so as to introduce a second level representation in the cluster hierarchy. To do this, a temporal analysis is performed with a view to splitting each cluster into a few temporally consistent sub-clusters (see Figure 2) according to the following criterion: inside each cluster, two subsequent shots are considered to belong to the same sub-cluster if the temporal distance between them is less than a predefined interval D_W (in terms of number of shots).

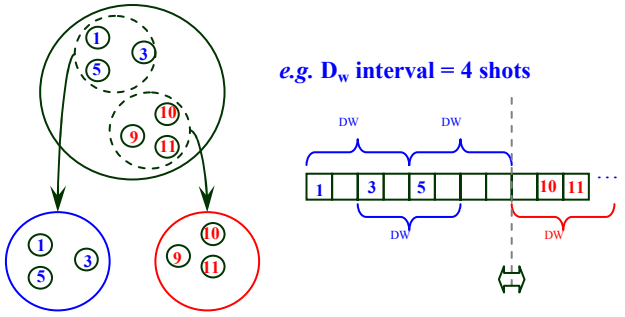

Figure 2: Cluster hierarchy: time local sub-clusters

The length of D_W=8 shots chosen for our tests is not a critical value: it only determines the maximally explored range for a shot with similar visual content (e.g., in a pattern like **A**BCDEFG**A**…, it allows the two shots **A** to be included in the same sub-cluster), whereas similar shots usually occur quite closely to each other.

4. *LSU* SEGMENTATION WITH *STG*

Based on the outcome of processing in Section 3, the *Scene Transition Graph (STG)* framework proposed in [9] can be conveniently employed to detect the *LSU* and extract the story structure without *a priori* knowledge of the video. In this case, the clusters and transitions form a directed graph: it comprises nodes (local sub-clusters) that contain a number of visually similar and temporally close shots, and the edges between nodes (transitions between shots) that represent the time evolution of the story. A special type of transition between two nodes is the so-called 'cut-edge', which, if removed, will lead to the decomposition of the graph into two disconnected sub-graphs. Therefore, a cut edge defines an *LSU* boundary. As shown in Figure 3, each connected sub-graph after the removal of cut-edges represents an *LSU* while the collection of all cut-edges represent all the transitions from one *LSU* to the next, thus reflecting the natural evolution of the video flow.

4.1. Pattern investigation inside *logical story units*

The *STG* as discussed above allows further analysis of different shot-patterns inside the detected *LSUs* and to distinguish between three main types of actions [7]: the

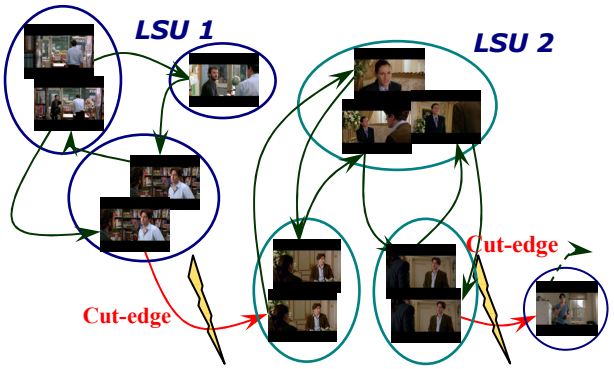

Figure 3: LSU detection through cut-edges

dialogues (e.g., *ABAB* or *ABCABC*), progressive actions (e.g., *ABCD*) and a hybrid of the two (*ABCBCD*).

Starting from the detected *LSUs* and the derived action patterns therein, a hierarchical organization of the story structure of the video can be easily created. Besides, a hierarchical segment decomposition MPEG-7 compliant is automatically produced, which can be useful for further developing effective browsing tools for navigation or summarization [5] of the video document.

5. EXPERIMENTAL RESULTS

A prototype software has been realized based on the preceding discussion, and experiments carried out using video segments from two feature movies and two cartoons with quite distinct story structures: a 50 min excerpt from the comedy "Notting Hill", a 17 min one from "A Beautiful Mind", and two 20 min segments taken from the cartoons "Don Quixote" and "Lucky Luke".

Although some *LSU* segmentation methods have been reported in literature, a common accepted ground-truth data set for comparison is still missing. To evaluate our system appropriately we choose to use the criteria discussed in the latest and most complete work on *logical story unit* segmentation evaluation [8], which proposes some measures on the effectiveness of segmentation results more meaningful and reliable than the counts of false positive and false negative. In particular, measures of *Coverage* χ and *Overflow* Ω on *LSU*s are proposed; *Coverage* χ is the fraction of shots of automatically generated *LSU* λ_j that overlap the most with the ground-truth *LSU* Λ_t, while *Overlap* Ω measures the overlap of automatically generated *LSUs* λ_j covering Λ_t, with the previous and subsequent ground-truth *LSUs* (see [8] for more details). The two measurements, averaged over the entire test video sequence for every candidate *LSU*, are presented in Table 1, where very low values of *Overflow* and the high scores of *Coverage* reveal the good performance of the proposed shot clustering algorithm that precedes the final *STG* analysis.

Table 1: Detected *LSU* in term of *Coverage* χ and *Overflow* Ω

Video	*# shots*	*# Λ_t*	*# λ_i*	*χ (%)*	*Ω (%)*
N-H	521	23	45	**78.9** %	**3.9** %
A-B-M	203	9	13	**91.1** %	**0.0** %
D-Q	167	13	17	**86.7** %	**2.3** %
L-L	197	15	22	**84.2 %**	**2.7 %**

6. CONCLUSIONS

In this work we have presented a novel content-based video analysis method for *logical story unit* detection based on visual *VQ* shot representation. The method aims to address three key issues: a time-unconstrained shot clustering, a method to automatically determine the number of clusters, and a procedure to analyze the temporal consistency of the visual content associated to each cluster. The output clusters and temporal links lend themselves easily to *STG* analysis to obtain desired *LSUs*. Experiments on feature movies and cartoons have demonstrated the promising results of this approach.

7. REFERENCES

[1] Adami N., Leonardi R., Wang Y., "Evaluation of different descriptors for identifying similar video shots," Proc. of IEEE ICME'2001, Tokyo.

[2] Gersho A., Gray R.M., Vector Quantization and Signal Compression, Kluwer Academic Publishers, Jan 1992.

[3] Hanjalic A., Lagendijk R., Biemond J., "Automated high-level movie segmentation for advanced video retrieval systems," IEEE Trans on CSVT., Vol.9, No.4, Jun 99.

[4] Kender J.R.,Yeo B-L.,"Video scene segmentation via continuous video coherence," CVPR-98, pp 367-373.

[5] Lee S.H., Yeh C.H., Kuo C.C.J., "Automatic movie skimming system with story units via general tempo analysis," Proc. SPIE, Vol. 5307, pp. 396-407, 2004.

[6] Saraceno C., Leonardi R., "Identification of story units in audio-visual sequences by joint audio and video processing," Proc. of ICIP, I: 363-367, Oct. 1998.

[7] Sundaram H., Chang S.F., "Determining computable scenes in films and their structures using audio-visual memory models," Proc. of ACM 2000, pp. 95-104.

[8] Vendrig J., Worring M., "Systematic evaluation of logical story unit segmentation," IEEE Trans. on Multimedia, Vol. 4, No. 4, Dec 2002.

[9] Yeung M., Yeo B-L., Liu B., "Segmentation of video by clustering and graph analysis," *Comput. Vis. Image Understand.*, vol. 71, no. 1, pp 94–109, 1998.

[10] Lienhart R., Pfeiffer S., Effelsberg W., "Scene determination based on video and audio features," IEEE Int. Conf. on Multim. Systems, vol. 1, pp. 685–690, 1999.

[11] E. Sahouria, A. Zakhor, "Content analysis of video using principal components," IEEE Trans. on CSVT, vol. 9, pp. 1290–1298, Dec. 1999.