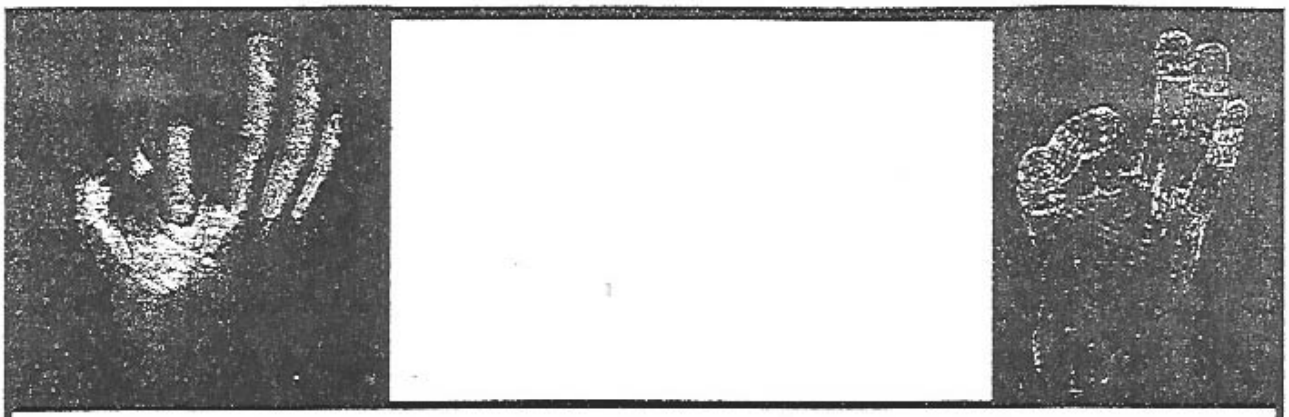
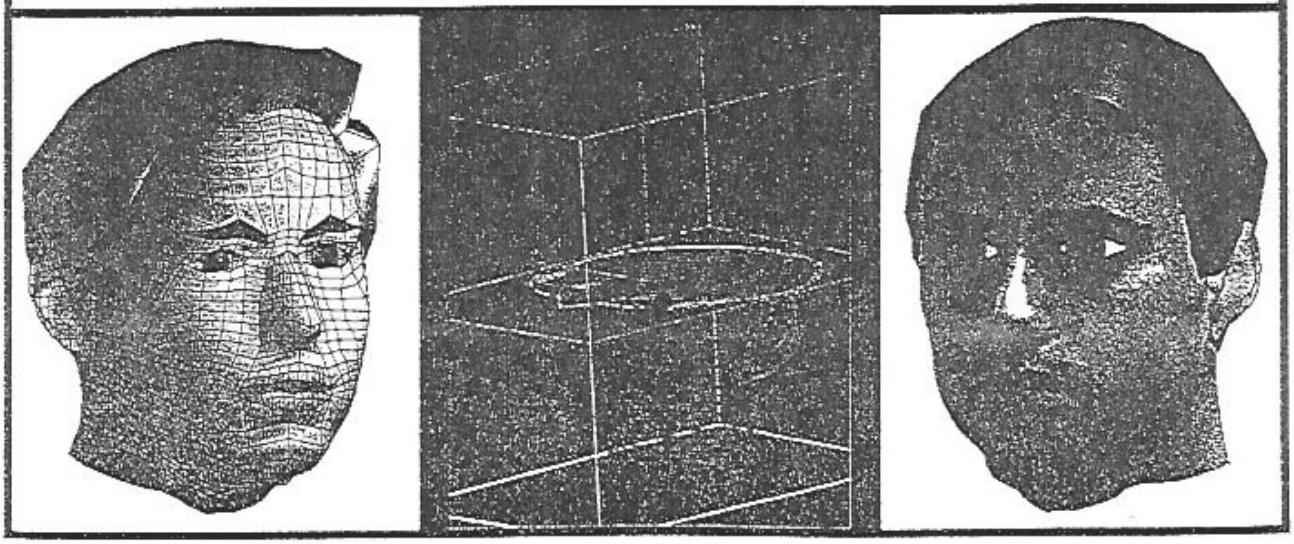


VLBV98

October 8-9, 1998

Beckman Institute for Advanced Science and Technology

University of Illinois at Urbana-Champaign



Identification of Visual Correlations Between Non Consecutive Shots in Digital Image Sequences *

C. Saraceno †‡ R. Leonardi †
† SCL Dept. of Electronic for Automation
University of Brescia, Brescia, Italy I-25123
‡ PRIP Institute for Automation
Vienna University of Technology, Vienna, Austria A-1040

Abstract

A number of automated methods for indexing audio-visual sequences have been developed. Typically, processing starts with a low level segmentation of a sequence of images so as to identify a series of shots (i.e. continuous camera records). To reach a higher level of description, patterns must be identified in the flow of consecutive shots. In this work, three different techniques for measuring visual correlations among non consecutive shots are proposed and compared. Two methods measure the visual correlation among shots by analyzing the respective K-Frames. In particular, they compare K-Frames based either on a low resolution DC JPEG representation or on color and spatial organization of the spatial information. The third technique measures the similarity between shots by comparing their associated codebooks, which are obtained using the Learning Vector Quantization approach. Simulations have shown that the Learning Vector Quantization approach leads to the best performance.

1 Introduction

Distributed processing and access to multimedia databases requires an important effort to organize audio-visual information. Recent efforts have been carried out to suggest that a cross-modal analysis of audio and video informations could bring significant benefits if a semantic level description is desirable [3, 5, 6, 7]. Most approaches usually start to segment the video in shots by applying techniques such as those proposed in [1, 2]. It has been recently suggested to merge consecutive shots exhibiting a similar semantic content, by simply using a characterization of the associated audio information [3, 4, 5, 6]. A more adequate representation is thus obtained when groups of shots can be merged into semantically more coherent entities called scenes. A cross-modal analysis of low-level visual and audio features can therefore bring an intermediate but satisfactory semantic description of the audio-visual sequence.

In this work the focus is placed on establishing visual correlations among non consecutive shots. Three different techniques are compared, the shots being detected using the approach suggested in [7]. Two of them estimate similarities between shots using characteristic K-Frames[1], while the third technique proposes to estimate the degree of correlation by using a codebook associated to each shot.

2 K-frame based analysis

In this section, a similarity measure based on K-frames is proposed For each shot, the central frame is chosen as K-frame. Two shots are considered similar if the associated K-frames are similar. The correlation between two K-frames is estimated using two techniques. The first technique compares a low resolution DC JPEG representation of the K-frames, while the second technique compares K-frames on the basis of color and spatial organization of information.

2.1 Low resolution comparison

In a first attempt to measure the similarity of two K-frames, pixel to pixel difference between the frames are considered. In order to reduce errors due to slight changes of illumination, or object locations, the DC JPEG representation of each luminance frame is used (i.e., 8×8 luminance block average (DC) value). It is well known that such a method is still sensitive to changes in illumination, camera position and object position. Nevertheless, it has still been used to demonstrate that correspondence of simple structural patterns in a movie, can be identified using this type of measure. The mean square difference between DC

*This work has been partially supported by the Italian Ministery of the University and of Scientific and Technological Research (MURST) and by the Austrian Science Foundation under grant S 7000-MAT.

[1] A K-Frame is a frame chosen as representative of the shot.

components is used as similarity measure between the two frames.

$$S_{dc}(F_i, F_j) = \frac{1}{M} \sum_{n=0}^{M-1} |c_{in}[0,0] - c_{jn}[0,0]|^2 \quad (1)$$

where, F_i and F_j are two generic frames to be compared, and $c_{ln}[0,0]$ is the DC coefficient of the n^{th} block of frame F_l.

2.2 Color based comparison considering spatial information

The idea of comparing frames based on color information has been widely utilized to retrieve images in a database. This is usually performed by evaluating the histogram of the images and comparing such histograms using the χ^2 test. Unfortunately, histogram based techniques do not take into account the spatial organization of information. Thus images with different content, but similar color information, are often erroneously declared as similar. In order to improve the effectiveness of these histogram-based methods, a measure of the correspondence of the spatial arrangement of color information has been proposed [8]. Accordingly, each K-frame is described using an adjacency graph. Each node in the graph represents a cluster of pixels exhibiting similar values. A similarity measure between graphs associated to two different images is obtained by considering color distances between pairs of matching nodes, differences in the adjacency information of such nodes, and differences in the relative amount of pixels being represented by such nodes. Matching nodes are determined by using the average color value of each pixel it represents. Thus, the final similarity value uses a combination of color and adjacency information.

The algorithm to identify pairs of matching nodes, between the two graphs associated to two K-frames F_1 and F_2, uses their color distance. A 2D correspondence matrix is built for this purpose, where each entry specifies the color distance between any two nodes. The distance between a node A in frame F_1 and a node B in frame F_2 is evaluated as follows:

$$\Delta_{(A,B)} = \sqrt{(L_A - L_B)^2 + (u_A - u_B)^2 + (v_A - v_B)^2} \quad (2)$$

where, (L_A, u_A, v_A) is the average color of node A and (L_B, u_B, v_B) is the average color of node B. The smallest distance in the correspondence matrix identifies a pair of matching nodes. This pair is inserted in a list while the corresponding row and column are removed from the correspondence matrix. The next pair is then searched by scanning the correspondence matrix to identify the next smallest distance. The process is iterated until there are no more elements in the correspondence matrix, or the color distance exceeds a predetermined threshold. It can happen that nodes from a given K-frame do not have a match in the other K-frame. For each pair (N_1, N_2) of matching nodes an adjustment is made to take into account the size of the two nodes:

$$\hat{W}_{(N_1,N_2)} = size(N_1) + size(N_2) + \frac{size(N_1)}{size(N_2)} \quad (3)$$

where $size(N_1)$ denotes the size (in terms of number of pixels) of the smaller node normalized to the total size of the image (and $size(N_2)$ is the normalized size of the other node). The more similar the sizes, the higher the weighting coefficient. Once all $\hat{W}_{(N_1,N_2)}$'s are determined, their sum S is considered. $S = \sum_{(N_1,N_2)\in P} \hat{W}_{(N_1,N_2)}$ The color similarity is evaluated by considering all pairs of matching nodes, using the following expression:

$$color\_similarity(F_1, F_2) = \sum_{(N_1,N_2)\in P} W_{(N_1,N_2)} * \Delta_{(N_1,N_2)} \quad (4)$$

where F_1 and F_2 are the two processed frames and $W_{(N_1,N_2)} = \frac{\hat{W}_{(N_1,N_2)}}{S}$ is the normalized weight, and P is the list of pairs of matching nodes.

The adjacency similarity is obtained by evaluating the difference between the node links, adjusted by the same weight used for the color distance. For each node pair, all combinations of links are considered. The adjacency similarity between frames F_1 and F_2 is defined as follows:

$$adj\_similarity(F_1, F_2) = \sum_{(e_{(A,D)},e_{(B,C)})\in E} \omega_{(e_{(A,D)},e_{(B,C)})} \sqrt{(d_{(A,D)} - d_{(B,C)})^2} \quad (5)$$

where E is the set of node pairs having common links, $d_{(i,j)}$ is the adjacency measure associated with the link connecting i and j, and $\omega_{(e_{(A,D)},e_{(B,C)})} = W_{(A,B)}$

Finally the total matching value is obtained as follows:

$$S_{col}(F_1, F_2) = 0.6 * color\_similarity(F_1, F_2) + +0.4 * adj\_similarity(F_1, F_2) \quad (6)$$

The smaller S_{col}, the more similar the two frames.

3 Vector codebook based analysis

So far, assuming that the video has been temporally segmented in shots, the correlation among shots was exploited utilizing the K-frames of the shots. This technique has one important side effect: the number and the position of the K-frames in each shot can affect the performance of the proposed technique to estimate the shot correspondence, when just K-frames are compared. In fact, major problems occur when the compared two K-frames are recorded under different conditions. Indeed, even when shots compare well, the corresponding K-frames are often taken from different viewpoints and/or with different object orientations, with a resulting change of shadowing, shading and lighting conditions. In addition, situation such as occlusion or disclosure of objects should also be taken into account. A simple but more robust way to analyze correlations existing among shots can be achieved through a vector quantization based approach.

The vector quantization utilized in this work is the *Learning Vector Quantization* (LVQ) proposed by Kohonen [9]. This is used to determine a codebook for each shot. The measure of similarity between two shots utilizes their associated codebook.

In our application, the LVQ has been utilized as a one-class problem. The video is split in shots, and, for each shot, a set of reference vectors, representing the shot is identified using the LVQ design procedure.

Every frame of a shot is divided into non overlapping blocks of size NXM, scanning the image from left to right and top to bottom. Each block is, indeed, a signal vector $\mathbf{x} \in \Re^{NXM}$. All blocks of all frames belonging to the shot are labeled as belonging to the same class. These vectors are then passed to the LVQ algorithm. The number of reference vectors, which represent the dimensionality of the class, is chosen equal to the number of blocks in a frame. Therefore, it does not depend on the length of the shot, but on the size of frames. The reference vectors returned by the LVQ algorithm, also called codevectors, will constitute the codebook associated to that shot.

Once a codebook has been associated to a shot, a similarity measure between two shots, based on the codevectors representing the shots, can be defined.

Let S_i be a shot and let C_j be a codebook, when a vector $\mathbf{x}_{il} \in S_i$ is quantized to a vector $\mathbf{v}_{jk} \in C_j$, a quantization error occurs. This quantization error may be measured by the average distortion

$$D_j(S_i) = \frac{1}{M} \sum_{l=0}^{M-1} \|\mathbf{x}_{il} - \mathbf{v}_{jk}\| \qquad (7)$$

where M is the number of vectors $\mathbf{x}_{il}$ of shot S_i (i.e. the number of blocks forming all its frames), and $\mathbf{v}_{jk}$ is the codevector of C_j with the smallest distance with respect to $\mathbf{x}_{il}$, i.e. $k = \arg\min_u \|\mathbf{x}_{il} - \mathbf{v}_{ju}\|$ with $\mathbf{v}_{ju} \in C_j$. Furthermore, $\|D_j(S_i) - D_k(S_i)\|$ can be reasonably interpreted as the distance between two codebooks (C_j and C_k) when applied to shot S_i.

A similarity measure between two shots can, thus, be defined as follows:

$$S_{vq}(S_i, S_j) = \|D_j(S_i) - D_i(S_i)\| + \|D_i(S_j) - D_j(S_j)\| \qquad (8)$$

where $D_i(S_i)$ is the distortion obtained when shot S_i is quantized using its associated codebook. The smaller S_{vq} the more similar the shots are. It is to be noticed that the similarity is based on the cross-effects of codebooks on each shot. In fact, if only one of the two elements of the sum in Eq. (8) is used, it is more likely that the resulting value be low, although the two shots are different. This happens, for example, when the majority of blocks of one shot, e.g. S_j is similar to a subset of blocks of the other shot, S_i. Therefore, the codebook associated to S_i, can represent S_j with a resulting small average distortion. On the other hand, the codebook associated with S_j cannot keep the same distortion level when applied to S_i.

4 Simulation results

300 K-frames were extracted from movies and news reports, and compared using S_{dc} and S_{col} (as defined in Eq(1) and Eq(6), respectively); whereas, the corresponding shots, associated with these K-Frames, were processed utilizing the S_{vq} similarity measure (defined in Eq(8)).

Both S_{dc} and S_{col} showed very low values whenever two similar frames came from news reports. This is caused by the fact that, in the case of news, the content of a picture, usually, does not change significantly: the anchor remains in the same location, the background is often still, etc. In the case of movies instead, the camera usually changes viewpoint, or simply, the objects in the picture move around. Choosing a same threshold (on S_{dc} or S_{col} values) when comparing K-frames extracted from news material and those extracted from movies, would penalize either news or movies. On the one hand, a low threshold would give acceptable performance in the case of news, but it would result in a high percentage of misses in the case of movies. On the other hand, a high threshold would declare similar a higher number of K-frames, but it would result in a higher percentage of errors, especially in the case of news, where, for example, two frames having different anchors could be recognized

	Detections	Misses	False Alarms
S_{dc}	59.7%	40.3%	6.67%
S_{col}	69.3%	30.7%	20%
S_{vq}	69.7%	30.3%	8.86%

Table 1: Comparison of similarity measures

as similar. Based on this consideration, two different thresholds were selected when using S_{dc} and S_{col}, similarity measures, depending on the type of source material. A lower threshold was set for news material with respect to movies.

Results are shown in Table 1. 67% of K-frames and shots was extracted from movies, while the remaining 33% belonged to news. The S_{dc} similarity measure achieves an overall detection of 59.7% (40.3% of misses) and a false alarm rate of 6.67%. Here misses are usually due to changes of illumination, viewpoint or object positions. However, the method seems to be robust to errors (i.e. false detection of similar frames). The S_{col} measure achieves an overall detection of 69.3% (30.7% misses) and a false alarm rate of about 20%. In this case, errors are usually due to frames having similar colors and similar spatial organization, whereas, misses are due to changes of viewpoint, or appearance/disappearance of spatially large objects. The S_{vq} measure seems to be more robust to errors when compared with the S_{col} measure. The detection capability is similar to the one obtained with the S_{col} similarity measure. Misses usually occur when one of the two shots can be considered a subset of the other shot. For example, if shot S_j can be divided into two parts, one part similar to S_i, and another part different, then $\|D_j(S_i) - D_i(S_i)\|$ may result in a low value, whereas the codebook associated to S_i may be not adequate to represent shot S_j, producing a high value in the term $\|D_i(S_j) - D_j(S_j)\|$.

5 Conclusions

In this work we have proposed three techniques in order to evaluate the visual correlation existing between non consecutive shots. Results have shown that the Vector Quantization based approach leads to better performance due to its ability to summarize information of the entire shots. By dividing further shots into microsegments exhibiting consistent visual content, it is likely that a more detailed correspondence of the sequence of frames can be achieved, leading thus to better performance overall. In a related work, we propose or combine this result with an audio processing technique for a segmentation of an audio-visual sequence which leads to a more semantic level description of information content [7].

References

[1] I. K. Sethi & N. Patel, "A Statistical Approach to Scene Change Detection", *Storage and Retrieval for Image and Video Databases III*, SPIE Vol. 2420: 329-338, Feb. 1995.

[2] H. Zhang, C. Y. Low and S. W. Smoliar, "Video Parsing and Browsing Using Compressed Data", *Multimedia Tools and Application*, Kluwer Academic Publishers, Boston, Vol. 1: 89-111, 1995.

[3] C. Saraceno & R. Leonardi, "Identification of Successive Correlated Camera Shots using Audio and Video Information", in *Proc. of ICIP'97*, Santa Barbara, CA (U.S.A.), III:166-169, Oct. 97.

[4] N. Patel and I.K. Sethi, "Video Classification Using Speaker Identification", *Proc. of EI'97: Storage and Retrieval for Image and Video Databases V*, Vol. SPIE-3022, pp. 218-225, Feb. 1997.

[5] J. Nam and A. H. Tewfik, "Combined Audio and Visual Streams Analysis for Video Sequence Segmentation," *Proc. of ICASSP'97*. Vol.3, pp.2665-2668, 1997.

[6] Y. Wang, J. Huang, Z. Liu and T. Chen, "Multimedia Content Classification using Motion and Audio Information," *Proc. of IEEE ISCAS'97*, Vol.2, pp.1488-1491, 1997.

[7] C. Saraceno & R. Leonardi, "Identification of Story Units in Audio-Visual Sequences by Joint Audio and Video Processing," to appear in *Proc. of ICIP'98*, Chicago, Illinois, Oct. 4-7, 1998.

[8] I. Tastl and C. Wolf, "Color Based Image Retrieval Considering Spatial Information," to appear in *Proc. of Electronic Image Capture and Publishing Conferences*, Zurich, 1998.

[9] T. Kohonen, "The self-organizing map," *Proc. of the IEEE* 78(9):1464-1480, 1990