USE OF SUPPORT VECTOR MACHINES BASED ON COLOR AND MOTION FEATURES FOR SHOT BOUNDARY DETECTION

C. Dalatsi, S. Krinidis, S. Tsekeridou and I.Pitas

Department of Informatics, Aristotle Univ. of Thessaloniki,
BOX 451, 54006 Thessaloniki, Greece
Tel.:+3031996304, Fax: +3031998419
e-mail: pitas@zeus.csd.auth.gr

ABSTRACT

In this paper we present a shot cut detection technique based on the fusion of information provided by three individual methods of analyzing both the static and the dynamic features of video sequences. The presented video analysis focuses on extracting multiple low-level content descriptors that once combined, enhance the shot boundary detection results achieved when each descriptor is considered individually. Static feature analysis provides the system with the bin-wise differences of color vector histograms and also with the color frame differences, both computed between successive frames. Furthermore, dynamic feature analysis yields bin-wise differences of motion vector histograms calculated on successive frames too. Static and dynamic features are combined to form 4D feature vectors characterizing a frame in the temporal dimension. Shot cuts are detected via the implementation of a supervised Support Vector Machine classifier so that feature vectors are assigned to two classes, one representing shot cuts and another representing in-shot information. The method proves to behave satisfactorily leading to remarkable recall and precision values.

Keywords: shot boundary detection, dynamic feature analysis, low-level content descriptors, data fusion

1. INTRODUCTION

During the last decades the amount of video data produced, stored, edited and communicated has been increasing rapidly thus introducing a growing need for tools enabling fast, efficient and automatic indexing, querying, browsing and delivery of audio-visual data. Content-based video indexing has evolved into a major research topic involving among other temporal video segmentation into elementary units which are further used for efficient video retrieval or browsing.

Extended research has thus been carried out on video shot cut detection techniques. The term *shot* denotes a sequence of successive frames that corresponds to a single camera start and end session. The major techniques that have been used for shot boundary detection are pair-wise frame differences [1], statistical differences [2], histogram comparisons [2], [3], edge differences [2] and different source integration techniques [4]-[6], most of which involve the a-priori or adaptive determination of an appropriate threshold [7], [3].

This work has been supported by the European Project RTN1-1999-00177 MOUMIR

Instead of employing thresholding methods in order to identify shot cuts, the problem may be reformulated as a pure classification problem concerning two main clusters: one cluster representing shot boundaries and one containing in-shot information. The approach presented here uses the Support Vector Machine (SVM) clustering algorithm to achieve static and dynamic feature-based data fusion. The shot cut detection system described in this paper is presented in Figure 1. After static and dynamic feature analysis (Section 2), the extracted features are transformed into 4-D feature vectors (Section 3.1) which are then classified in order to detect shot boundaries (Section 3.2). Experimental results are presented in Section 4 and conclusions are drawn in Section 5.

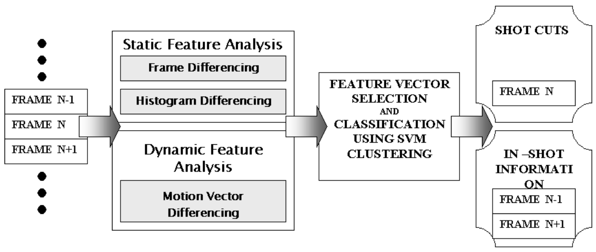

Figure 1: Block diagram of the shot cut detection algorithm.

2. STATIC AND DYNAMIC FEATURE ANALYSIS

After having been properly processed, the low-level content descriptors introduced by the static and dynamic feature analysis on every frame will form the components of the feature vectors to be clustered for shot cut detection.

2.1. Static feature analysis

Static feature analysis involves the calculation of two color distance metrics. The *first method* estimates color frame differences $FD(t)$ between successive frames by means of the formula [1]:

$$FD_t = \frac{1}{3N_X \times N_Y} \sum_{\mathbf{x}} \|\mathbf{I}(\mathbf{x};t) - \mathbf{I}(\mathbf{x};t-1)\|_1 \qquad (1)$$

where $\mathbf{I}((\mathbf{x};t) = [I_r(\mathbf{x};t)I_g(\mathbf{x};t)I_b(\mathbf{x};t)]^T$ denotes the vectorvalued pixel intensity function composed of the three color components: $I_r(\mathbf{x};t)$, $I_g(\mathbf{x};t)$ and $I_b(\mathbf{x};t)$. By $\|\cdot\|$ we denote the L_1-vector norm whereas $\mathbf{x} = (x, y)$ and t span the spatial (each frame is of size $N_X \times N_Y$) and the temporal dimension of the video sequence respectively. This metric is of significant value in cases of abrupt shot cuts where the number of pixels whose color is modified increases significantly. This is not the case in smooth shot transitions, nevertheless this type of static information aids in clarifying some false detections caused by noisy results in dynamic feature analysis.

The *second method* evaluates color vector histograms, calculates their bin-wise differences and then performs summation over all bins in order to conclude to a single scalar distance value for each frame. Histograms in general remain invariant under camera/object motion, translation and rotation about the view axis. Furthermore, robustness against shot lightening changes can be attained by using the HLS color space and ignoring luminance information. Histogram vector bins are constructed by considering all possible pairs of the scalar hue and saturation bins $\mathbf{hs}_k = (h_i, s_j)$, where h_i, $i = 1 \ldots 32$ denotes the 32 equally-spaced hue bins and s_j, $j = 1 \ldots 8$ denotes the 8 equally-spaced saturation bins, leading thus to a total number of 256 vector bins. The color vector bin-wise histogram $H(\mathbf{hs}_k;t)$ for any frame indexed with t is computed by counting all pixels having hue and saturation values lying inside the considered vector bin $\mathbf{hs}_k$ and dividing by the total number of frame pixels. The histogram differences are then computed for every pair of frames $(t-1, t)$ as follows [3]:

$$HD_t = \frac{1}{N_X \times N_Y} \sum_{k=1}^{256} \ln(\|H(\mathbf{hs}_k;t) - H(\mathbf{hs}_k;t-1)\|_1) \quad (2)$$

where k denotes the vector bin index and t is the temporal spatial dimension of the video sequence. Each frame is of size $N_X \times N_Y$. This histogram-based metric provides us with the second type of static information to be combined in the shot boundary detection procedure to follow. Shot boundaries characterized by abrupt changes in frame content are directly related with significant values of histogram differences.

Exclusive use of visual static features in shot detection techniques as the ones presented here yield satisfactory detection results for shot boundaries characterized by easily recognizable visual content changes. In order to enhance the detection capability of the presented method, dynamic feature analysis is also proposed.

2.2. Dynamic feature analysis

Dynamic feature analysis involves the calculation of motion vector histogram differences following the procedure described in Section 2.1. To begin with, motion field estimation is performed by a block matching approach between pairs of successive frames, capable of detecting pure translational motion only. For each block of size $N \times N$ centered around pixel $\mathbf{n} = (n_1, n_2)$ of a frame indexed by t in the spatiotemporal dimension, we are seeking the corresponding motion vector of the form $\mathbf{d}(\mathbf{n}, t) = [d_x, d_y]$ where d_x and d_y denote the translation in the horizontal and the vertical axis respectively [8]. Given that block-matching-based approaches often result in noisy motion fields and in blocking artifacts, a motion field smoothing step is further adopted.

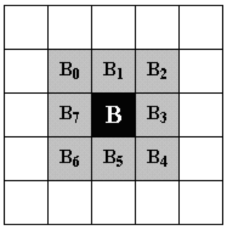

Figure 2: Part of video frame divided in blocks, containing the 3×3-sized neighborhood around block B whose motion vector is to be smoothed.

The motion field smoothing procedure presented here is based on vector rational interpolation, which employs a 3×3 sized motion vector neighborhood centered around the motion vector of interest as depicted in Figure 2. If we define by $\mathbf{m}_i$, $i = 0 \ldots 7$, the motion vector corresponding to block B_i in Figure 2, then by:

$$\mathbf{S} = \{\{\mathbf{m}_0, \mathbf{m}_4\}, \{\mathbf{m}_1, \mathbf{m}_5\}, \{\mathbf{m}_2, \mathbf{m}_6\}, \{\mathbf{m}_3, \mathbf{m}_7\}\} \quad (3)$$

we denote the set of motion vector pairs of blocks diametrically opposite with reference to block B, that are used in the smoothing process. The resulting smoothed motion vector $\mathbf{m}$ of block B is calculated by:

$$\mathbf{m} = \frac{\sum_{\{\mathbf{u},\mathbf{v}\} \in \mathbf{S}} w_{\mathbf{uv}}(\mathbf{u} + \mathbf{v})}{2 \sum_{\{\mathbf{u},\mathbf{v}\} \in \mathbf{S}} w_{\mathbf{uv}}} \quad (4)$$

where $w_{\mathbf{uv}}$ are the rational weights defined by:

$$w_{\mathbf{uv}} = \frac{1}{1 + 2.5\|\mathbf{u} - \mathbf{v}\|} \quad (5)$$

This smoothing process leads to uniform motion vectors within moving objects and to an overall motion field deprived of outliers caused by block matching method failures.

Motion vector histograms are evaluated by considering all possible pairs of the estimated scalar translations in the horizontal and in the vertical axis $\mathbf{d_x d_y}_k$, where $\mathbf{d}_{\mathbf{x}_i}$, $\mathbf{d}_{\mathbf{y}_j}$ for $i = 1 \ldots 9$, $j = 1 \ldots 9$, denote the 9 equally-spaced horizontal and vertical translation bins, respectively. Considering thus 81 motion vector bins, the motion vector bin-wise histogram $H(\mathbf{d_x d_y}_k;t)$ for any frame indexed with t in the spatiotemporal dimension is computed by counting all blocks having horizontal and vertical translations in the ranges defined by the considered motion vector bin $\mathbf{d_x d_y}_k$ and by dividing this number by the total number of blocks per frame. The motion vector histogram differences are then computed for every pair of frames $(t-1, t)$ as follows

$$MD_t = \frac{N \times N}{N_X \times N_Y} \sum_{k=1}^{81} \ln(\|H(\mathbf{d_x d_y}_k;t) - H(\mathbf{d_x d_y}_k;t-1)\|_1) \quad (6)$$

where k denotes the vector bin index. Motion estimation-based approaches like the one adopted here can assist any color-based technique in detecting shot boundaries deprived of significant chromatic changes. Even in cases of dissolves or other gradual transitions where other metrics fail at least partially, this motion field-based approach outputs significant values within a temporal window including the shot cut.

3. FEATURE VECTOR SELECTION AND FRAME CLASSIFICATION

Instead of thresholding the difference metrics, as usually done to accomplish shot cut detection, we have chosen to combine them through data fusion techniques, in order to form 4D feature vectors. These are inputted to an SVM classifier which outputs two clusters, one of which represents shot cuts.

3.1. Feature vector formulation

Having three scalar values in our disposal representing the visual static and dynamic features of each frame ((1), (2), (6)), we proceed in properly transforming them into feature vectors characterizing each video frame. The feature vector for frame t is formulated as follows:

$$\mathbf{v}_t = \begin{bmatrix} HD_t \\ MD_t \\ MD_t + MD_{t+1} + MD_{t+2} \\ FD_t \end{bmatrix} \quad (7)$$

These features are selected based on observations regarding in-shot visual content related with a shot boundary. To begin with, color frame differencing and color histogram differencing metrics take relatively low values in frames inside shots while they present peaks at abrupt shot cuts. The reason for employing both techniques is that frame differencing considers spatial along with chromatic information in estimating content similarity between pairs of frames. What is noted is that, despite the nature of the shot boundary, there is a temporal neighborhood in its vicinity where the values of the motion estimation-based metric are great, in relation with the previous in-shot values. That is why the outcome of summation of the motion histogram difference metric over a temporal window of size 3 is used as the 3rd component of the feature vector. This third component transforms the local neighborhood of increased values corresponding to a shot cut into a single peak situated at the last frame of a shot (see Figure 3).

3.2. Shot boundary detection using a Support Vector Machine classifier

After having performed static/dynamic feature analysis and feature selection on a set of observations consisting of l frames, we have to formulate the appropriate machine providing us with the matching $\mathbf{v}_i \longrightarrow y_i$, where $i = 1 \ldots l$, $\mathbf{v}_i$ the feature vector corresponding to the i-th frame of the video sequence and y_i its associated "truth" as defined in [9]. In our case, y_i can only be set equal to the values $-1, 1$, the first denoting that a frame is situated in the internal of a shot while the second is assigned to frames situated at the end of a shot i.e. to those introducing a shot boundary. Practically, the learning machine is defined through a set of possible matchings $\mathbf{v}_i \longrightarrow f(\mathbf{v}_i)$.

The approach adopted here is that of Support Vector Machines, which seek for the proper hyperplane $g(\mathbf{v}_t) = 0$ to separate the training set $\{(\mathbf{v}_1, y_1), (\mathbf{v}_2, y_2) \ldots (\mathbf{v}_l, y_l)\}$ in two classes, using the restriction that $g(\mathbf{v}_t) \geq 1$ if $y_t = 1$ and $g(\mathbf{v}_t) \leq -1$ if $y_t = -1$. Training is performed in a supervised fashion, meaning that a relatively small portion of the feature vectors $\mathbf{v}_i$ is assigned a-priori with the appropriate matching value y_i. More specifically, the decision function is a non-linear function of the data, such that

$$f(\mathbf{v}) = \sum_i^l a_i y_i K(\mathbf{v}_i, \mathbf{v}_j) + b \quad (8)$$

where $K(\mathbf{v}_i, \mathbf{v}_j)$ denotes a "kernel function" [9]. The most useful kernel functions for this pattern recognition problem are presented below

$$K(\mathbf{v}_i, \mathbf{v}_j) = (\mathbf{v}_i \cdot \mathbf{v}_j + 1)^p \quad (9)$$

$$K(\mathbf{v}_i, \mathbf{v}_j) = e^{\gamma \|\mathbf{v}_i - \mathbf{v}_j\|^2} \quad (10)$$

where (9) denotes the polynomial kernel of degree p and (10) the RBF kernel function.

4. EXPERIMENTAL RESULTS

In our initial experiments, we used a 6 minute TV news video sequence containing mainly abrupt shot cuts. The video sequence has been digitized with a frame rate of 25 fps and with a resolution of 184x136 pixels.

Static feature analysis follows the procedure described in Section 2.1. Dynamic feature analysis was tested for different sets of block-matching method parameters to discover the best ones. The process concluded to the parameters depicted in Table 1.

Block Size	$N \times N$	8×8
Search Window Size	$N + 2M \times N + 2M$	16×16
Block Overlapping	NO	

Table 1: Block matching method parameters

Thus, we end up with 9 possible horizontal and vertical translation values $-4 \ldots 4$ and with 81 total motion vector bins as mentioned in Section 2.2.

In Figure 3 a typical abrupt shot boundary is depicted, by means of the values of the three presented metrics, calculated in a temporal window of size 25 centered at the frame of interest (i.e. at the last frame of the shot, where the shot boundary is to be detected). The characteristics of abrupt shot cuts are easy to observe.

The shot boundary detection procedure presented in Section 3.2 is performed via supervised clustering. The first 2 min of the video sequence are used in the training phase, while the remaining ones are used for testing. In order to evaluate the performance of this method, the following performance criteria are used [2]:

$$Recall = \frac{relevant\ correctly\ retrieved\ shots}{all\ relevant\ shots} = \frac{N_c}{N_c + N_m} \quad (11)$$

$$Precision = \frac{relevant\ correctly\ retrieved\ shots}{all\ retrieved\ shots} = \frac{N_c}{N_c + N_f} \quad (12)$$

where N_c denotes the number of correctly detected shot boundaries, N_m the number of shot boundaries that are not detected (missed) and N_f the number of falsely detected ones. For comparison purposes, the results obtained using three different kernel

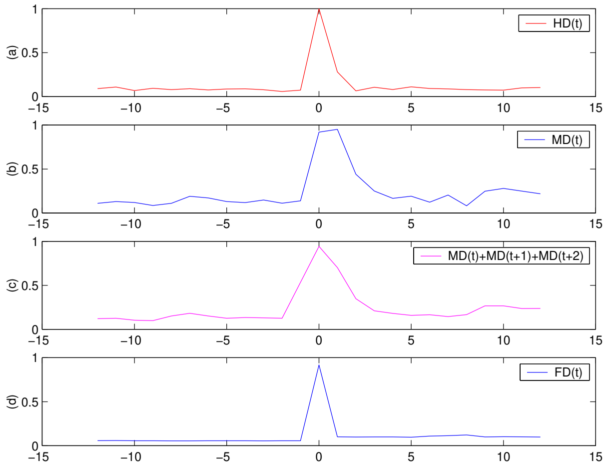

Figure 3: Features value behavior around a typical shot cut. (a)-(d) correspond to the 4 features forming the feature vector.

functions in the SVMs are presented in Table 2. The kernel parameter values correspond to the ones described in (9) and (10). It is noted that the linear kernel function is no other than the polynomial kernel function with parameter $p = 1$.

Kernel	N_c	N_m	N_f	**Recall**	**Precision**
linear	20	0	9	1	0.6897
polynomial $p = 2$	20	0	1	1	0.9524
RBF $\gamma = 0.001$	20	0	1	1	0.9524

Table 2: Performance results for the news sequence

Even though the presented method does not yet account for gradual shot transitions, we have tested it also for a part of a TV serial sequence with many commercials in-between, containing many shots, characterized by significant edit effects like zoom-ins and dissolves, abrupt camera movement and significant motion inside single shots. The video sequence is of approximately 12 min duration and it has been digitized with the same frame rate as the news sequence. Regarding the shot boundary detection procedure, the first 2.5 min of the video sequence have been used for training and the rest for testing. It is noted that the edit effects appear at the last half of the sequence. This means that the training process does not account for edit effects, which is the reason why the performance results now (Table 3) appear inferior to the previous ones.

Kernel	N_c	N_m	N_f	**Recall**	**Precision**
linear	120	16	15	0.8824	0.8889
polynomial $p = 2$	119	17	13	0.8750	0.9015
RBF $\gamma = 0.001$	127	9	15	0.9338	0.8944

Table 3: Shot boundary detection results for the TV serial sequence

5. CONCLUSIONS

In this paper, a shot cut detection technique based on the fusion of visual static and dynamic feature analysis results is presented. The novelty of this method lies in both the use of properly formulated feature vectors to represent video frame content and in the implementation of SVM clustering during the shot cut detection procedure in order to divide the derived feature vectors in two clusters, one characterizing shot cuts and another representing in-shot visual content. The method proves to perform satisfactorily even in cases of gradual shot transitions, although no attempt to make the method robust against them has been made. In future, we intend to broaden its field of application to account for more complicated shot transitions.

6. REFERENCES

[1] Sofia Tsekeridou and Ioannis Pitas, "Audio-visual content analysis for content-based video indexing," in *Proc. of 1998 IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, 1999.

[2] A. Del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, Inc, San Francisco, California, 1999.

[3] S. Krinidis S. Tsekeridou and I. Pitas, "Scene change detection based on audio-visual analysis and interaction," in *Proc. of 2000 Multi-Image Search and Analysis Workshop*, Schloss Dagstuhl, Germany, 12-17 March 2000, to appear in Lecture Notes in Computer Science, Springer Verlag, R. Klette, Th. Huang, G. Gimel'farb (eds.), 2001.

[4] Zhu Liu Jincheng Huang and Yao Wang, "Integration of audio and visual infromation for content-based video segmentation," in *Proc. of 1998 IEEE International Conference on Image Processing*, Chicago, Illinois, USA, 1997.

[5] C. Saraceno and R. Leonardi, "Identification of successive correlated camera shots using ausio and video information," in *Proc. of 1997 IEEE International Conference on Image Processing*, 1997.

[6] M. Naphade R. Mehrotra A.M. Ferman J. Warnick T.S. Huang A.M. Tekalp, "A high-performance shot boundary detection algorithm using multiple cues," in *Proc. of 1998 IEEE International Conference on Image Processing*, Chicago, Illinois, USA, 1998.

[7] Bilge Gunsel and A. Murat Tekalp, "Content-based video abstraction," in *Proc. of 1998 IEEE International Conference on Image Processing*, Chicago, Illinois, USA, 1998.

[8] Murat Tekalp A, *Digital Video Processing*, Prentice-Hall, Inc, Upper Saddle River, 1995.

[9] Christopher J.C. Burges, "A tutorial on support vector machines for pattern recognition," Tech. Rep., Bell Laboratories, Lucent Technologies, 1998.