Tennis video abstraction from audio and visual cues

F. Coldefy[1] P. Bouthemy[1] M. Betser[1] G. Gravier[2]
[1]IRISA/INRIA [2]IRISA/CNRS
Campus de Beaulieu, 35042 Rennes Cedex, France

***Abstract*— We propose a context-based model of video abstraction exploiting both audio and video features and applied to tennis TV programs. We can automatically produce different types of summary of a given video depending on the users' constraints or preferences. We have first designed an efficient and accurate temporal segmentation of the video into segments homogeneous w.r.t. the camera motion. We introduce original visual descriptors related to the dominant and residual image motions. The different summary types are obtained by specifying adapted classification criteria which involve audio features to select the relevant segments to be included in the video abstract. The proposed scheme has been validated on 22 hours of tennis videos.**

I. Introduction and related work

Video abstraction is motivated by the growing need of fast or selective visualization of TV broadcasts or videos. It may be an efficient tool to browse a video by picking out the main points of the program or to preview a video by selecting the meaningful sequences according to the program theme.

We focus our attention on tennis videos. We aim at automatically producing different types of summaries by selecting the relevant sequences according to the user's preferences. The summary might be restricted to the highlights of the game (*e.g.*, best rallies or winning services) or be more comprehensive by including all the rallies for instance. Summaries of different duration or focus may thus be produced. We extract appropriate audio and visual features to characterize relevant events in the video. We define low-level visual descriptors related to the dominant and residual image motions. The different summary types stem from the formulation of the audio and visual criteria for selecting the desired video contents. The proposed scheme has been tested on seven tennis programs (22 hours).

Papers on sports video summarization often focus on soccer and especially on goal detection. Tekalp et al. propose in [2] an automatic goal detection method based on dominant color extraction and shot classification. Leonardi et al. introduce in [4] an audio and visual model exploiting an hidden Markov models (HMM) to classify each pair of successive shots. Hanjalic [3] has proposed a deterministic excitement criterion based on the mean dominant motion magnitude per shot, the density of cuts, and the audio loudness to detect goals in soccer video. In [6], Zhong et al. have developed visual features including color clustering, object segmentation and line detection for classification of tennis and baseball shots. However, model learning involved in HMM or in color-based methods may be a long process since it requires precise and manual indexation of numerous videos. Although the method presented below needs also a learning stage, the training set can be quickly constructed and easily exploited.

II. General description

We have defined a four-step method to create video abstracts. First, we perform a temporal segmentation of the video based on shot change detection and on the dominant image motion analysis. Then, we extract the segments which could be of potential interest for the video abstract. This second step requires a dedicated off-line learning stage. Independently, we segment the soundtrack and achieve the audio event detection ("applause" and "ball hits" detection). Finally, we select on visual and audio criteria the segments to be retained for the different abstract types.

We specify the temporal video segmentation as the detection of the camera motion changes combined with a shot change detection method. This segmentation does not depend on the video type (sports, movie or documentary).

From each video segment, we extract low-level visual features evaluating the spatial distribution of the residual motion. In an off-line learning stage, a K-means clustering of these features is performed over a training set formed with the kind of segments to be involved in the video abstract. Then, we introduce a statistical decision rule to decide whether each segment of the video to be processed resembles the prototypes of the classes of interest, and thus is offering a potential interest for the abstract or not.

Audio event detection is independently carried out in a two-step process. First, the soundtrack is segmented into segments with homogeneous content. Then, the detection of ball hits and applause is performed based on statistical hypothesis testing.

The video abstract results from the selection of specific clusters which allow us to determine the candidate segments. It is finally obtained by retaining clusters of video segments corresponding to the user's preferences.

In Section 3, we describe the temporal video segmentation. Sections 4 and 5 are concerned with the characterization of the visual and audio content respectively. Section 6 deals with the abstract creation step. In Section 7, experimental results are reported and Section 8 contains concluding remarks.

III. Temporal video segmentation

A. Global motion model

We specify the temporal video segmentation as the detection of the camera motion changes combined with a shot change detection method. We represent the camera motion by a 2D affine motion model [1] (more specifically, this model accounts

for the dominant image motion assumed to be related to the camera motion):

$$w_\theta(p) = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix} \quad (1)$$

where $p = (x, y)$ is an image point whose coordinates are expressed in the image coordinate system, and $\theta = (a_i)_{i=1..6}$ denotes the motion model parameters. This model can correctly handle different global motions corresponding to camera panning, zooming and other complex camera motions. The estimation of θ is performed by the IRLS (Iterative Reweighted Least Squares) algorithm minimizing a robust M-estimator criterion [5].

B. Shot change detection

We have developed an extended version of the method introduced in [1]. The shot changes are detected from ζ_t, the normalized size of the set of pixels for which the estimated dominant motion explains the interframe displacement. The quantity ζ_t is derived from the robust estimation of the global motion model θ at each image instant t. Its value belongs to [0,1]. It is close to 1 when there is no independent moving objects in the scene. When crossing a shot change, ζ_t undergoes a downward jump followed by an upward jump which are more or less sharp whether the shot change is a cut or a progressive transition. These jumps are detected by a downward and an upward Page-Hinkley tests [1].

We have further added the detection of locally abnormal values on ζ_t, performed by an on-line mean test at each instant t: alarm if $|\zeta_t - \mu|/\sigma > g$, where g is a Gaussian threshold set to 3 and μ and σ^2 are the unknown mean and variance of ζ_t estimated on-line. Page-Hinkley and mean tests are performed in parallel and then combined.

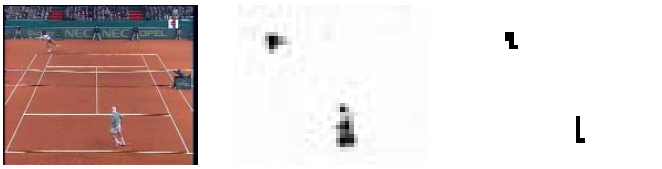
C. Temporal segmentation based on dominant motion changes

Various type of dominant motions as panning, zoom in and out, can be identified by the $(a_i)_{i=1..6}$ parameters of the affine model. We focus our attention on the constant parameters $\tau_\theta(t) = (a_1(t), a_4(t))$ which are involved in most of the dominant motions. Let $\nu(t)$ and $\beta(t)$ be the magnitude and the orientation of the vector $\tau_\theta(t)$ respectively. In order to reduce noise before computing $\nu(t)$ and $\beta(t)$, we apply a temporal median filter of length 9 on $a_1(t)$ and $a_4(t)$. We first proceed to the binary segmentation of $\nu(t)$. We decide the shot is locally static at frame t if $\nu(t) < k$ where k is a given threshold. This binary segmentation is regularized in order to suppress isolated points and to favor longer segments. Within the segments where the dominant motion was stated as not null, we detect the orientation changes of $\tau_\theta(t)$. This is achieved by a Page-Hinkley test on $\beta(t)$ where we fix the tolerated variation value to $\pi/4$. A fast post-processing of the segmentation is performed to suppress short segments (ten or less frames) if they are supplied by the camera motion change detection.

IV. Candidate segment selection on visual cues

A. Visual features

We define several visual features per video segment. We compute ζ^s, the mean value over the segment s of the dominant motion support size ζ_t. It characterizes the ratio of dominant image motion within the video segment.



a) original image *b) $\underline{\omega}(p,t)$ map* *c) valley detection*

Fig. 1. Valley detection in the outlier map to locate independent moving objects in the image

Furthermore, we consider the spatial distribution of the residual motion in the image, associated to the weights map $\omega(p,t)$ computed in the global motion estimation stage. $\omega(p,t)$ is the weight at pixel p for the frame t supplied by the IRLS algorithm once convergence is reached. Its value belongs to [0,1]. It is near 1 when the estimated global motion explains the observed motion at p and close to zero otherwise.

To get information on the spatial distribution of the residual motion, we first detect the valleys of $\underline{\omega}(p,t)$, a coarse resolution version of the $\omega(p,t)$ original map in a ratio of eight. The valley points of a $\mathcal{C}^2$ continuous surface $\mathcal{S}$ can be defined as the set of points M for which the highest principal curvature of $\mathcal{S}$ at these points is strictly positive and the associated principal direction is in the (x, y) plane. Since this relation is numerically untractable, we instead compute at each pixel p where the weight value is lower than a threshold r, the eigenvector decomposition of the Hessian of $\underline{\omega}(p,t)$. Pixel p is considered as a valley point if it is a local minimum of $\underline{\omega}(p,t)$ over a discrete segment of length l centered in p and aligned in the direction of the eigenvector associated to the highest curvature (provided it is positive). This detector produces skeletons of the main objects moving in the scene.

In order to remove some temporal noisy detections, we keep a valley point detected at pixel p at frame t if and only if it is near an actually detected valley at p in both next frames $t+1$ and $t+2$ (at most 2 pixels away).

Figure 1 shows an example of this valley detection step. It includes the original frame (352x288 pixels), the $\underline{\omega}(p,t)$ map and the binary detection of the valleys respectively (44x36 pixels). We fix r to 0.6 and l to 11 for all our experiments.

We then define h^s the 1-D histogram computed over the segment s of the horizontal cumulative projections of the valley maps onto the vertical axis. For instance, in segments corresponding to rallies, the tennis players are the unique moving subjects. As a consequence, the histogram presents two peaks corresponding each to a player. The normalized histogram is computed over 36 bins and is convolved with a triangle function before subsampling it to 9 bins. From h^s,

we compute d^s the standard deviation of h^s over s and n^s the mean number of valley detections per frame.

To sum up, we have defined two groups of features per video segment s: h^s, the 1-D normalized histogram of the valley maps, and $x^s = (\zeta^s, d^s, n^s)$ which globally characterizes the dominant and the residual image motions.

B. Learning stage

In a learning stage, we manually identify examples of shots which can be included in the video abstract (serves and rallies for instance). These shots used for the training stage are first automatically segmented as explained above. In order to estimate the different underlying motion components contained in the extracted segments of this training set, we perform a K-means clustering on h^s and x^s. We separately compute clusters on h^s and then on x^s in order to preserve the specificity of each group of features.

C. Segment selection step

We decide that a sequence s of a processed video resembles the elements of the learning set (the serves and rallies in our application) if its features x^s and h^s satisfy the following conditions: 1) among the clusters estimated as explained in subsection IV-B, there exists a cluster c for which each component i of x^s satisfies the statistical mean test: $\frac{|x_i^s - x_i^c|}{\sigma_i^c} < \gamma$, where σ_i^c is the standard deviation of the i-coordinate x_i^c of the cluster c; γ is a given Gaussian threshold; 2) the distance between h^s and at least one histogram among the cluster centroïds computed is lower than a predefined threshold h. We use the L^2 distance as for the K-means algorithm, but this choice is not crucial here.

V. Extraction of audio cues

We want to detect the parts of the soundtrack containing the information relevant to the targeted video abstract. In the application we are dealing with here, we have to detect "applause" and "ball hits". The soundtrack is first segmented into spectrally homogeneous chunks to which are eventually assigned the label "applause" or the label "ball hits" if the corresponding audio event is present.

A. Soundtrack segmentation

Segmenting the soundtrack into spectrally homogeneous units is carried out with the Bayesian information criterion (BIC) using a cepstral representation of the input signal. The BIC is defined as the log-likelihood of a segment y given a model, penalized by the model complexity and the unit length. Formally, for a segment of length T with an associated model Λ, the Bayesian information criterion is defined as $\mathcal{I}(\Lambda) = \ln f(y;\Lambda) - \lambda \frac{\#(\Lambda)}{2} \ln T$, where $\#(\Lambda)$ is the number of free parameters in the model and λ a tunable parameter theoretically equal to one (but not in practice).

The principle of the segmentation algorithm is to move two adjacent windows on the audio signal. For each position of the windows, one can compare whether it is more appropriate, in terms of BIC, to model the two windows separately with two different Gaussian distributions, say Λ_1 and Λ_2, or with a single one, say Λ_0. If $\mathcal{I}(\Lambda_0) - \mathcal{I}(\Lambda_1) - \mathcal{I}(\Lambda_2)$ is negative, then the two-segment model is chosen and a segment boundary is placed at the boundary between the two windows.

B. Audio classification

The presence or absence of the considered audio label has to be stated in every segments extracted from the previous soundtrack segmentation. This detection problem can be solved using a two-hypothesis test, where H_0 (resp., H_1) is the hypothesis that the event considered is (resp. is not) present in the segment. Assuming a model for the distribution of y is available under both hypotheses, the decision on the presence of an event is taken by comparing the log-likelihood ratio $\mathrm{R}(y) = \ln f(y;H_0) - \ln f(y;H_1)$ to a threshold δ, where $R(y) > \delta$ means that the event is detected in segment y.

In practice, $f(y;H_0) = f(y;M)$ where M is a Gaussian mixture model whose parameters were estimated from training data containing the event of interest (whether alone or superimposed with other events). Similarly, $f(y;H_1)$ is approximated using a "non-event" model $\overline{M}$ whose parameters were estimated on training data where the event considered is not present. The decision threshold δ was determined experimentally on training data.

Using this approach, the rate of correct audio event detection is around 88%, most of the errors coming from inaccuracies in the audio segmentation step.

VI. Tennis video abstraction

In subsection IV-C, we have explained how to select relevant segments of a processed video on visual cues. We now introduce the audio information by computing, for each preselected segment s, b^s the percentage of time during which ball hits are detected in s and a^s the percentage of the applause duration in the next segment $s+1$. We thus define a vector $y^s = (log(T^s), a^s, b^s)$ where T^s is the duration of the segment s. Other descriptors could be added too as a motion activity measure for instance. We apply the K-means algorithm on these data and obtain clusters which represent different shot types, the number of which has been empirically set to 8. We have in general two clusters of "applause" and "ball hits" segments corresponding to short shots and long shots (winning services and rallies respectively). A third cluster with "ball hits" without any "applause" segments corresponds to first serves. The five other clusters correspond to rallies with no particular interest, false visual detections (warm up shots, false detection), and missed audio detections.

Different types of abstracts may be obtained by choosing different decision rules. If one aims at getting the best rallies and serves of the game, one may only keep into the abstract the elements of the clusters corresponding to the "ball hits" with "applause" segments. Thus, we select the segments of the clusters c for which a^c and b^c, the "applause and "ball hits" coordinates of its centroid, are greater than some pre-defined thresholds (set to 0.8 and 0.6 respectively).

Summaries of fixed duration may also be produced by scoring the "ball hits" segments according to their "applause level" and selecting the n most applauded segments for the desired duration. Extensive summary can also be created by keeping all the relevant segments selected on visual cues. As one can see, many decision rules can be easily designed.

VII. Experimental results

A. Temporal segmentation

We have successfully tested the temporal segmentation algorithm on 33 hours of different video types including tennis, soccer, figure skating, athletics and documentaries. For all our experiments, we set the Hinkley parameters and the Gaussian threshold g to the same values. Threshold k on the norm $\nu(t)$ is chosen equal to 2 pixels for a 352x288 image. All the cuts and almost all the camera motion changes were detected. False cuts may occur when the camera is tracking a moving object or people. Special transitions (dissolve, fade, wipe) are most of the time properly detected if no independent moving object occupies a too large part of the viewed scene, otherwise the algorithm may detect a cut approximatively in the middle of the transition.

B. Candidate segment selection on visual cues

To create our learning set, we have extracted the first fifty rallies of three different videos, each from a different tournament (one outdoor on hard-packed surface and two indoors on slick and hard-packed surfaces, see Fig. 2). It corresponds to fifteen minutes of rally sequences for one hour and an half of video. We have performed the K-means clustering on x^s and h^s separately. We have empirically fixed the number of clusters to 15.

We applied the rally detection based on the defined visual cues on the rest of the three videos. We add four other videos from the same tournaments but recorded at different years. Altogether, we have 22 hours of videos. Each video contains between twenty and forty minutes of studio, interviews and commercials in addition to the tennis match.

We set h, the histogram distance threshold, to 0.2 and the Gaussian threshold γ to 2.25. For the preselection of candidate segments on visual cues, we obtain a precision rate which is the relevant objective evaluation measure when considering video abstraction, is steady around 95% (93% and 95.7% for the worst and best score respectively). False alarms correspond to static shots where some residual motion is present at the bottom or at the top of the frame (a speaker moving his head and his hands for instance). The preselected segments correspond to 12% up to 20% of the video.

Since outdoors matches on hard-packed surface and indoors matches on slick surfaces differ in game (long rallies versus short rallies) and in hall configuration (in small halls, the camera field of view is reduced and a panning is often necessary to shoot a rally), performances slightly drop when processing a video from a tournament no video of which is included in the learning set.

Fig. 2. Videos of the learning set for rallies and serves

C. Final video abstraction

We focus on the creation of a selective summary containing the best winning serves and rallies of the match. As described in Section VI, we perform an on-line clustering of the p-reselected video segments on visual cues and we retain the segments belonging to the "applause" and "ball hits" clusters. The final summaries corresponds to 6% up to 13% of the videos. Very few false alarms remain in the abstract as the audio features increase the precision rate (actually one false alarm for seven video summaries). We may produce even shorter summaries of limited duration by keeping into the abstract the most applauded segments only.

VIII. Conclusion

We have described an efficient method for flexible video abstraction based on audio and visual features. When applied to tennis videos, we are able to accurately identify first serves, winning serves and rallies in order to create summaries of various duration and focus according to the user's preferences or interests. Such a goal can be attained if the video has been first properly segmented and a relevant pre-selection of segments based on spatio-temporal visual cues has been achieved. All our experiments have been carried out with the same set of parameter values which are not critical to fix.The computation is almost real time (22.5 Hz instead of 25 Hz).

Acknowledgment

This work was partly supported by the French Ministery of Industry with the RNTL Domus Videum project. The videos are provided by INA.

References

[1] P. Bouthemy, M. Gelgon and F. Ganansia, "An unified approach to shot change detection and camera motion characterization" *IEEE Trans. on Circuits and Systems for Video Technology*, vol 9, pp 1030-1044, 1999.

[2] A. Ekin, A. M. Tekalp and R. Mehrotra "Automatic soccer video analysis and summarization" *IEEE Trans. on Image Processing*, vol.12, pp 796-807, July 2003.

[3] A. Hanjalic, "Generic approach to highlights extraction from a sport video", *ICIP'2003*, Barcelona, Sept. 2003.

[4] R. Leonardi, P. Migliorati and M. Prandini, "Annotation, Retrieval, and Relevance Feedback - Semantic Indexing of Soccer Audio-Visual Sequences: A Multimodal Approach Based on Controlled Markov Chains", *IEEE Trans. on Circuits and Systems for Video Technology* , vol 14, n^0 5, p634-643, May 2004 .

[5] J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models", *Jal of Visual Comm. and Image Repr.*, vol. 6(4): 348-365, Dec. 1995.

[6] D. Zhong and S.F. Chang, "Structure analysis of sports video using domain models", *ICME'2001*, Tokyo, Aug. 2001.