7

TEMPORAL SEGMENTATION OF VIDEO DATA

Edoardo Ardizzone, and Marco La Cascia
Dipartimento di Ingegneria Informatica
University of Palermo
Palermo, ITALY
{ardizzone,lacascia}@unipa.it

1. INTRODUCTION

Temporal segmentation of video data is the process aimed at the detection and classification of transitions between subsequent sequences of frames semantically homogeneous and characterized by spatiotemporal continuity. These sequences, generally called camera-shots, constitute the basic units of a video indexing system. In fact, from a functional point of view, temporal segmentation of video data can be considered as the first step of the more general process of content-based automatic video indexing. Although the principal application of temporal segmentation is in the generation of content-based video databases, there are other important fields of application. For example in video browsing, automatic summarization of sports video or automatic trailer generation of movies temporal segmentation is implicitly needed. Another field of application of temporal segmentation is the transcoding of MPEG 1 or 2 digital video to the new MPEG-4 standard. As MPEG-4 is based on video objects that are not coded in MPEG 1 or 2, a temporal segmentation step is in general needed to detect and track the objects across the video.

It is important to point out the conceptual difference between the operation of an automatic tool aimed at the detection and classification of the information present in a sequence and the way a human observer analyzes the same sequence. While the human observer usually performs a semantic segmentation starting from the highest conceptual level and then going to the particular, the automatic tool of video analysis, in a dual manner, starts from the lowest level, i.e. the video bitstream, and tries to reconstruct the semantic content. For this reason, operations that are very simple and intuitive for a human may require the implementation of quite complex decision making schemes. For example consider the process of archiving an episode of a tv show. The human observer probably would start annotating the title of the episode and its number, then a general description of the

scenes and finally the detailed description of each scene. On the other side the automatic annotation tool, starting form the bitstream, tries to determine the structure of the video based on the homogeneity of consecutive frames and then can extract semantic information. If we assume a camera-shot is a sequence of semantically homogeneous and spatiotemporally continuous frames then the scene can be considered as an homogeneous sequence of shots and the episode as a homogeneous sequence of scenes. In practice, the first step for the automatic annotation tool is the determination of the structure of the video based on camera-shots, scenes and episodes as depicted in figure 1.

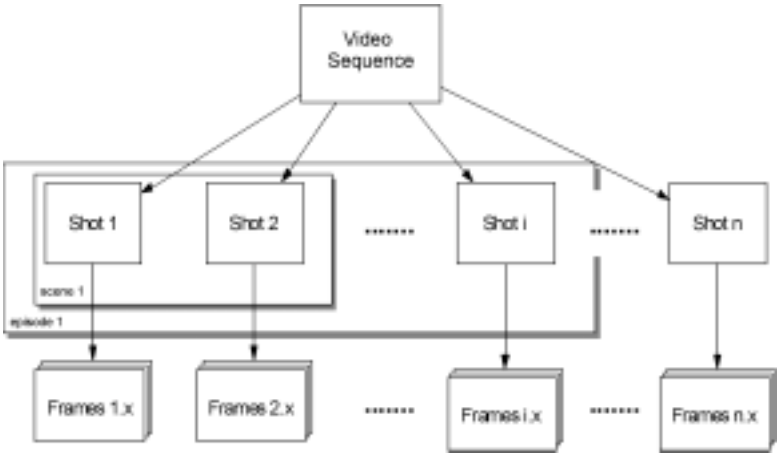

Figure 1. General structure of a video sequence.

In a video sequence the transitions between camera-shots can be of different typologies. Even though in many cases the detection of a transition is sufficient, in some cases it is important at least to determine if the transition is abrupt or gradual. Abrupt transitions, also called *cuts*, involve only two frames (one for each shot), while gradual transitions involve several frames belonging to the two shots processed and mixed together using different spatial and intensity variation effects [4].

It is also possible to further classify gradual transitions [28], [5]. Most common gradual transitions are fades, dissolves and wipes. Fade effects consisting of the transition between the shot and a solid background are called fade-out, while opposite effects are called fade-in. Fade-in and fade-out are often used as beginning and end effect of a video. Dissolve is a very common editing effect and consists of a gradual transition between two shots where the first one slowly disappears and, at the same time and at the same place, the second one slowly appears. Finally, wipes are the family of gradual transitions where the first shot is progressively covered with the second one, following a well defined trajectory. While in the dissolve the superposition of the two shots is obtained changing the intensity of the frames belonging to the shots, in the case of the wipes the superposition is obtained changing the spatial position of the frames belonging to the second shot. In Figure 2 are reported a few frames from a fade-out, a dissolve and a wipe effect.

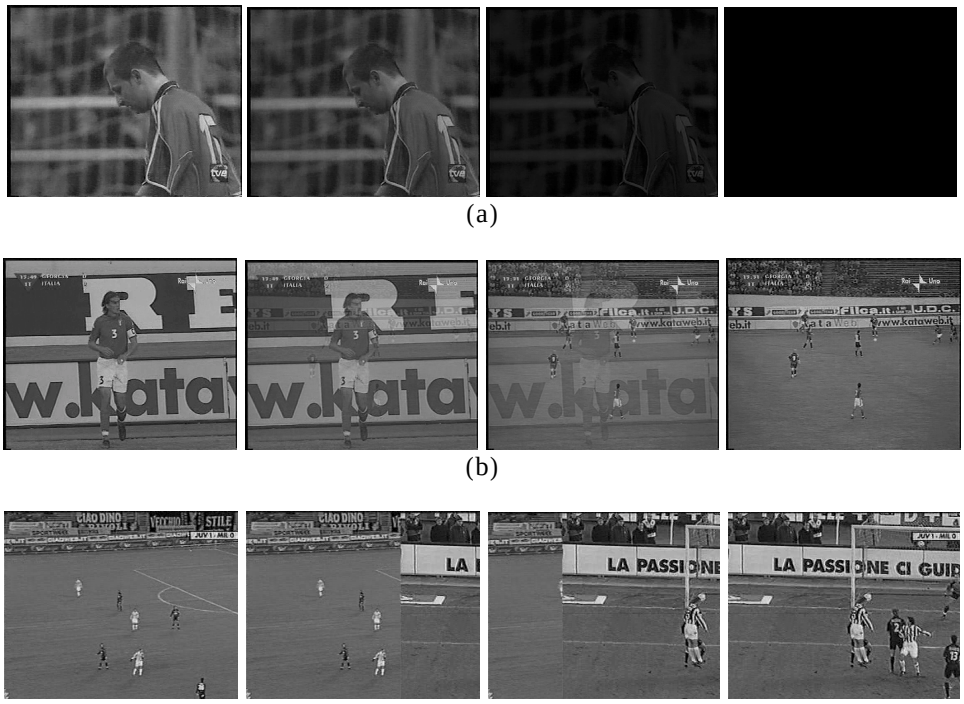

Figure 2. Examples of editing effects. (a) fade-out, (b) dissolve, (c) wipe.

The organization of this chapter is as follows. In Sect. 2 we review most relevant shot boundary detection techniques, starting with basic algorithms in uncompressed and compressed domains and then discussing some more articulated methodologies and tools. In Sect. 3 we propose a new segmentation technique, grounded on a neural network architecture, which does not require explicit threshold values for detection of both abrupt and gradual transitions. Finally, we present in Sect. 4 a test bed for experimental evaluation of the quality of temporal segmentation techniques, employing our proposed technique as an example of use.

2. TEMPORAL SEGMENTATION TECHNIQUES

It has been pointed out that the aim of temporal segmentation is the decomposition of a video in camera-shots. Therefore, the temporal segmentation must primarily allow to exactly locate transitions between consecutive shots. Secondarily, it is of interest the classification of the type of transition. Basically, temporal segmentation algorithms are based on the evaluation of the quantitative differences between successive frames and on some kind of thresholding. In general an effective segmentation technique must combine an interframe metric computationally simple and able to detect video content changes with robust decision criteria.

In subsections 2.1-2.4 we will review some of the most significant approaches proposed in recent years and discuss their strength and limitations. These techniques also constitute the building blocks of more sophisticated segmentation methodologies and tools, such as the ones

illustrated in subsection 2.5, and may be used to evaluate new segmentation techniques.

2.1 TECHNIQUES BASED ON PIXELS DIFFERENCE METRICS

Metrics classified as PDM (pixels difference metrics) are based on the intensity variations of pixels in equal position in consecutive frames. Temporal segmentation techniques using interframe differences based on color are conceptually similar, but they are not very popular as they have a greater computational burden and almost the same accuracy of their intensity based counterpart.

A basic PDM is the sum of the absolute differences of intensity of the pixels of two consecutive frames [8]. In particular, indicating with $Y(x, y, j)$ and $Y(x, y, k)$ the intensity of the pixels at position (x, y) and frames j and k, the metric can be expressed in the following way:

$$\Delta f = \sum_x \sum_y \left| Y(x, y, j) - Y(x, y, k) \right| \tag{1}$$

where the summation is taken all over the frame.

The same authors propose other metrics based on first and second order statistical moments of the distributions of the levels of intensity of the pixels. Indicating with μ_k and σ_k respectively the values of mean and standard deviation of the intensity of the pixels of the frame k, it is possible to define the following interframe metric between the frames j and k:

$$\lambda = \frac{\left[\frac{\sigma_j + \sigma_k}{2} + \left(\frac{\mu_j - \mu_k}{2} \right)^2 \right]^2}{\sigma_j \sigma_k} \tag{2}$$

This metric has been also used in [7], [30], and is usually named *likelihood ratio*, assuming a uniform second order statistic. Other metrics based on statistical properties of pixels are:

$$\eta_1 = \frac{\left| \mu_j - \mu_k \right| \left\| \sigma_j^2 - \sigma_k^2 \right|}{\sigma_j \sigma_k \left(\frac{\mu_j + \mu_k}{2} \right)} \qquad \eta_1 \geq 0 \tag{3}$$

$$\eta_2 = \frac{\mu_j \sigma_j^2}{\mu_k \sigma_k^2} \qquad \mu_j > \mu_k,\ \sigma_j > \sigma_k \tag{4}$$

$$\eta_3 = \left(\frac{\mu_j \sigma_j}{\mu_k \sigma_k} \right)^2 \qquad \mu_j > \mu_k,\ \sigma_j > \sigma_k \tag{5}$$

Associating a threshold to a metric it is possible to detect a transition whenever the metric value exceeds the threshold value. As it was pointed out in [8], PDMs offer the best performance for the detection of abrupt

transitions. In general all the PDMs are particularly exposed to the effects of noise, camera/object movements or sudden lighting changes, leading to temporally or spatially localized luminance perturbations, and then to a potentially large number of false transitions. From this point of view, slightly better performances are obtained considering block-based comparisons between successive frames, with matching criteria based on values of mean and variance in corresponding blocks. Several block-based PDMs are reviewed in [18].

In order to limit the effects of local perturbations, some authors have developed top-down methods based on mathematical models of video. For example, in [1] a differential model of motion picture is presented, where three factors concur to the intensity variations of pixels of consecutive frames: a small amplitude additive zero-centered Gaussian noise, essentially modelling the noise effects of camera, film and digitizer, the *intrashot* intensity variations due to camera/object motion and focal length or lightness changes; and finally the *intershot* variations due to the presence of abrupt or gradual transitions. In [34], another approach for gradual transitions detection based on a model of intensity changes during fade out, fade in and dissolve effects is presented.

Another interesting approach based on PDMs has been proposed in [6]. In this paper the authors propose the use of *moment invariants* of the image. Properties such as the scale and rotation invariance make them particularly suited to represent the frame. Denoting by $Y(x, y)$ the intensity at the position (x, y), the generic moment of order pq of the image is defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q Y(x,y) \tag{6}$$

Moment invariants are derived from normalized central moments defined as:

$$n_{pq} = \frac{1}{m_{00}^{\gamma}} \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q Y(x,y) \tag{7}$$

where:

$$\gamma = 1 + \frac{(p+q)}{2} \qquad \bar{x} = \frac{m_{10}}{m_{00}} \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{8}$$

Limiting our attention to the first three moment invariants, these are defined as:

$$\begin{aligned} \phi_1 &= n_{20} + n_{02} \\ \phi_2 &= (n_{20} - n_{02})^2 + 4n_{11}^2 \\ \phi_3 &= (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2 \end{aligned} \tag{9}$$

These three numbers may be interpreted as the components of a vector, say $\boldsymbol{\sigma}$, that can be used to represent the image:

$$\boldsymbol{\sigma} = \{\phi_1, \phi_2, \phi_3\} \tag{10}$$

The interframe metric adopted in [6] is the Euclidean distance between the vector $\boldsymbol{\sigma}$ associated to frames j and k:

$$f_{moms}(j,k) = \left|\boldsymbol{\sigma}_j - \boldsymbol{\sigma}_k\right|^2 \tag{11}$$

Although the use of moment invariants can lead to robust segmentation algorithms, with respect to noise and other local perturbations, techniques based on statistical properties generally exhibit a computational load not adequate for specific applications, e.g. real time systems.

2.2 TECHNIQUES BASED ON HISTOGRAM DIFFERENCE METRICS

Metrics classified as HDM (histograms difference metrics) are based on the evaluation of the histograms of one ore more channels of the adopted color space. As it is well known, the histogram of a digital image is a measure that supplies information on the general appearance of the image. With reference to an image represented by three color components, each quantized with 8 bit/pixel, a three-dimensional histogram or three monodimensional histograms can be defined. Although the histogram does not contain any information on the spatial distribution of intensity, the use of interframe metrics based on image histograms is very diffused because it represents a good compromise between the computational complexity and the ability to represent the image content.

In recent years several histogram-based techniques have been proposed [9], [6], [23], [37], some of them are based only on the luminance channel, others on the conversion of 3-D or 2-D histograms to linear histograms [2]. An RGB 24 bit/pixel image would generate an histogram with 16.7 millions of bins and this is not usable in practice. To make manageable histogram-based techniques, a coarse color quantization is needed. For example in [2] the authors use an histogram in the HSV color space using 18 levels for hue and 3 for saturation and value leading to a easily tractable 162 bins histogram. Other approaches are based both on PDM and HDM. For example, in [24] two metrics are combined together: a PDM metric is used to account for spatial variation in the images and a HDM metric to account for color variations. Ideally, a high value in both metrics corresponds to a transition, however the choice of thresholds and weights may be critical.

In what follows some of the most popular HDM metrics are reviewed. In all the equations M and N are respectively the width and height (in pixels) of the image, j and k are the frame indices, L is the number of intensity levels and $H[j, i]$ the value of the histogram for the i-th intensity level at frame j. A commonly used metric [9] is the *bin-to-bin* difference, defined as the sum of the absolute differences between histogram values computed for the two frames:

$$f_{db2b}(j,k) = \frac{1}{2MN}\sum_{i=0}^{L-1}\left|H(j,i) - H(k,i)\right| \tag{12}$$

The metric can easily be extended to the case of color images, computing the difference separately for every color component and weighting the results. For example, for a RGB representation we have:

$$f_{drgb}(j,k) = \frac{r}{s} f_{db2b}(j,k)^{(red)} + \frac{g}{s} f_{db2b}(j,k)^{(green)} + \frac{b}{s} f_{db2b}(j,k)^{(blue)} \quad (13)$$

where *r, g,* and *b* are the average values of the three channels and *s* is:

$$s = \frac{(r+g+b)}{3} \quad (14)$$

Another metric, used for example in [9], [6] is called *intersection* difference and is defined in the following way:

$$f_{d\,\mathrm{int}}(j,k) = 1 - \frac{1}{MN} \sum_{i=0}^{L-1} \min[H(j,i), H(k,i)] \quad (15)$$

In other approaches [28], the *chi-square test* has been used, that is generally accepted as a test useful to detect if two binned distributions are generated from the some source:

$$f_{dchi2}(j,k) = \sum_{i=0}^{L-1} \frac{[H(j,i) - H(k,i)]^2}{[H(j,i) + H(k,i)]^2} \quad (16)$$

Also the correlation between histograms is used:

$$f_{dcorr}(j,k) = 1 - \frac{\mathrm{cov}(j,k)}{\sigma_j \sigma_k} \quad (17)$$

where *cov(i, j)* is the covariance between frame histograms:

$$\mathrm{cov}(j,k) = \frac{1}{L} \sum_{i=0}^{L-1} [H(j,i) - \mu_j][H(k,i) - \mu_k] \quad (18)$$

and μ_j e σ_j represent the mean and the standard deviation, respectively, of the histogram of the frame *j*:

$$\mu_j = \frac{1}{L} \sum_{i=0}^{L-1} H(j,i) \quad (19)$$

$$\sigma_j = \sqrt{\frac{1}{L} \sum_{i=0}^{L-1} [H(j,i) - \mu_j]^2} \quad (20)$$

All the metrics discussed so far are global, i.e. based on the histogram computed over the entire frame. Some authors propose metrics based on histograms computed on subframes. For example, in [5] a rectangular 6x4 frame partitioning is used. Each frame is subdivided into 6 horizontal blocks and 4 vertical blocks. The reason for this asymmetry is that horizontal movements are statistically more frequent than vertical ones. Then an HDM is used to compute difference between corresponding blocks in consecutive frames, after an histogram equalization. The shape of histograms is also taken into account for each block, using histogram moments up to the third order. Region differences are also weighted, using experimentally determined

coefficients, to account for different contribution, in correspondence of a shot transition, of low and high intensity levels. The global metric is finally obtained adding up the contribution of all the sub-regions. In despite of its greater complexity, the performance of this technique remains, in many cases, quite sensitive to the choice of thresholds and weights.

Both PDM and HDM techniques are based on the computation of a similarity measure of two subsequent frames and on comparison of this measure with a threshold. The choice of the threshold is critical, as too low threshold values may lead to false detections and too high threshold values may cause the opposite effect of missed transitions.

To limit the problem of threshold selection, several techniques have been proposed. For example, in [5] a short-term analysis of the sequence of interframe differences is proposed to obtain temporally localized thresholds. The analysis is performed within a temporal window of appropriate size, for example 7 frames. This approach is more robust to local variations of brightness and camera/object motion. In detail, naming D_k the difference metric between the frames k and k-1 and W_k the short-term temporal window centered on the frame k, we have:

$$\rho(k) = \frac{D_k}{M_k} \qquad \text{where} \quad M_k = \max_{k \in W_k} \{D_k\} \tag{21}$$

If $\rho(k)$ exceeds a threshold, computed experimentally, then an abrupt transition at frame k is detected. In a similar approach based on temporal windows and local thresholds [7], the statistical properties of a bin-to-bin histogram metric are evaluated over a 21-frame temporal window and used to detect transitions.

2.3 TECHNIQUES FOR DETECTION OF EDITING EFFECTS

The techniques reported in the previous sections, based on PDMs or HDMs, are mainly suited for the detection of abrupt transitions. Other techniques have been developed to detect gradual transition like fades or wipes. For example, in [5] the authors propose a technique to detect fading effects based on a linear model of the luminance L in the CIE L*u*v color space. Assuming that chrominance components are approximately constant during the fading, the model for a fade-out is:

$$L(x,y,t) = L(x,y,t_0)\left(1 - \frac{t - t_0}{d}\right); \qquad t \in [t_0, t_0 + d] \tag{22}$$

where $L(x, y, t)$ is the luminance of the pixel at position (x, y) and time t, t_0 is the time of beginning of the fading effect and d its duration. Similarly the model for a fade-in is:

$$L(x,y,t) = L(x,y,t_0 + d)\left(\frac{t - t_0}{d}\right); \qquad t \in [t_0, t_0 + d] \tag{23}$$

Even if the behavior of real luminance is not strictly linear during the fading, a technique based on the recognition of pseudo-linearity of L may be used to detect transitions. Even in this case, however, local thresholds have to be considered. Moreover, for some kind of video with very fast dynamic

characteristics (for example TV commercials) this technique cannot be used. Other approaches have been proposed to overcome this limitation. For example, in [12] a production-based model of the most common editing effects is used to detect gradual transitions, in [37] a simple and effective two-thresholds technique based on HDM is reported. The two thresholds respectively detect the beginning and the end of the transition. The method, called of *twin-comparison*, takes into account the cumulative differences between frames of the gradual transition. In the first pass a high threshold T_h is used to detect cuts, in the second pass a lower threshold T_l is employed to detect the potential starting frame F_s of a gradual transition. F_s is then compared to subsequent frames. An accumulated comparison is performed as during a gradual transition this difference value increases. The end frame F_e of the transition is detected when the difference between consecutive frames decreases to less than T_l, while the accumulated comparison has increased to a value higher than T_h. If the consecutive difference falls below T_l before the accumulated difference exceeds T_h, then the potential start frame F_s is dropped and the search continues for other gradual transitions.

Specific techniques have been aimed at the detection of other gradual transitions. For example, in [32]a two-step technique for the detection of wipe effects is proposed. It is based on statistical and structural properties of the video sequence and operates on a partially decompressed MPEG streams. In [26]the authors propose a technique for transition detection and camera motion analysis based on spatiotemporal textures (spatiotemporal images will be further treated in the next subsection 2.5). The analysis of the texture changes can lead to the estimation of shooting conditions and to the detection of some types of wipes.

Finally, techniques based on higher level image features have also been tried. For example, in [35] the analysis of intensity edges between consecutive frames is used. During a cut or a dissolve, new intensity edges appear far from the locations of the old edges. Similarly, old edges disappear far from the location of new edges. Thus, by counting the entering and exiting edge pixels, cuts, fades and dissolves may be detected and classified.

2.4 TECHNIQUES OPERATING ON COMPRESSED VIDEO

Due to the increasingly availability of MPEG [19]compressed digital video, many authors have focused their attention on temporal segmentation techniques operating directly on the compressed domain or on a partially decompressed video sequence. Before discussing some of presented methods, we shortly review the fundamentals of MPEG compression standard.

MPEG uses two basic compression techniques: 16 x 16 macroblock-based motion compensation to reduce temporal redundancy and 8 x 8 Discrete Cosine Transform (DCT) block-based compression to capture spatial redundancy. An MPEG stream consists of three types of pictures, I, P and B, which are combined in a repetitive pattern called group of picture (GOP).

I (*Intra*) frames provide random access points into the compressed data and are coded using only information present in the picture itself. DCT coefficients of each block are quantized and coded using Run Length Encoding (RLE) and entropy coding. The first DCT coefficient is called DC term and is proportional to the average intensity of the respective block.

P (*Predicted*) frames are coded with forward motion compensation using the nearest previous reference (I or P) pictures.

B (*Bi-directional*) pictures are also motion compensated, this time with respect to both past and future reference frames.

Motion compensation is performed finding for each 16 x 16 macroblock of the current frame the best matching block in the respective reference frame(s). The residual error is DCT-encoded and one or two motion vectors are also transmitted.

A well known approach to temporal segmentation in the MPEG compressed domain, useful for detecting both abrupt and gradual transitions, has been proposed in [33] using the DC sequences. A DC sequence is a low resolution version of the video, since it is made up of frames where each pixel is the DC term of a block (see Figure 3). Since this technique uses I, P and B frames, a partial decompression of the video is necessary. The DC terms of I frames are directly available in the MPEG stream, while those of B and P frames must be estimated using the motion vectors and the DCT coefficients of previous I frames. This reconstruction process is computationally very expensive.

Figure 3. Sample frame from a sequence and corresponding DC image.

Differences of DC images are compared and a sliding window is used to set the thresholds for abrupt transitions. Both PDM and HDM metrics are suited as similarity measures, but pixel differences-based metrics give satisfactory results as DC images are already smoothed versions of the corresponding full images. Gradual transitions are detected through a accurate temporal analysis of the metric.

The technique reported in[28] uses only I pictures. It is based on the chi-square test applied to the luminance histogram and to row and column histograms of DC frames. The use of horizontal and vertical projections of the histogram introduces local information that is not available in the global histogram. In particular, for frames of M x N blocks, row and column histograms are defined as:

$$X_i = \frac{1}{M}\sum_{j=1}^{M} b_{0,0}(i,j) \quad ; \qquad i = 1,...,N$$

$$Y_j = \frac{1}{N}\sum_{i=1}^{N} b_{0,0}(i,j) \quad ; \qquad j = 1,...,M \tag{24}$$

where $b_{0,0}(i, j)$ is the DC coefficient of the block (i, j). The three interframe differences computed on the three histograms are then used in a binned decision scheme to detect abrupt or gradual transitions. As only I frames are used, the DC recovering is eliminated. Note that as in a video there are typically two I frames per second, the analysis based on I frames only is adequate to approximately detect abrupt transition. For gradual transitions this coarse temporal sub-sampling of the video may introduce more serious problems.

Another technique operating in the compressed domain is presented in [3], that is based on the correlation of DCT coefficients for M-JPEG compressed sequences. Other authors [38] extended this approach to MPEG sequences.

2.5 OTHER TECHNIQUES

Many authors tried to organize one or more of the previously described basic algorithms within more general frameworks, with the aim of defining and implementing more robust techniques. In particular, most of the approaches discussed so far rely on suitable thresholding of similarity measures between successive frames. However, the thresholds are typically highly sensitive to the type of input video.

In [11], the authors try to overcome this drawback by applying an unsupervised clustering algorithm. In particular, the temporal video segmentation is viewed as a 2-class clustering problem ("scene change" and "no scene change") and the well-known *K*-means algorithm [27] is used to cluster frame dissimilarities. Then the frames from the cluster "scene change" which are temporary adjacent are labeled as belonging to a gradual transition and the other frames from this cluster are considered as cuts. Both chi-square statistics and HDMs were used to measure frame similarity, both in RGB and YUV color spaces. This approach is not able to recognize the type of the gradual transitions, but it exhibits the advantage that it is a generic techniques that not only eliminates the need for threshold setting but also allows multiple features to be used simultaneously to improve the performance. The same limitations and advantages characterize the technique presented at end of this chapter.

Among others, Hanjalick recently proposed a robust statistical shot-boundary detector [14]. The problem of shot-boundary detection is analyzed in detail and a conceptual solution to the shot-boundary detection problem is presented in the form of a statistical detector based on minimization of the average detection-error probability. The idea is that to draw reliable conclusions about the presence or absence of a shot boundary, a clear separation should exist between discontinuity-value ranges for measurements performed within shots and at shot boundaries. Visual-content differences between consecutive frames within the same shot are mainly caused by two factors: object/camera motion and lighting changes. Unfortunately, depending on the magnitude of these factors, the computed

discontinuity values within shots vary and sometimes may cause detection mistakes.

An effective way to reduce the influence of motion and lighting changes on the detection performance is to embed additional information in the shot boundary detector. The main characteristic of this information is that it is not based on the range of discontinuity values but on some other measurements performed on a video. For example, this information may result from the comparison between the measured pattern formed by discontinuity values surrounding the interval of frames taken into consideration and a known template pattern of a shot boundary. Various template patterns, specific for different types of shot boundaries (cuts or fades, wipes and dissolves), may be considered.

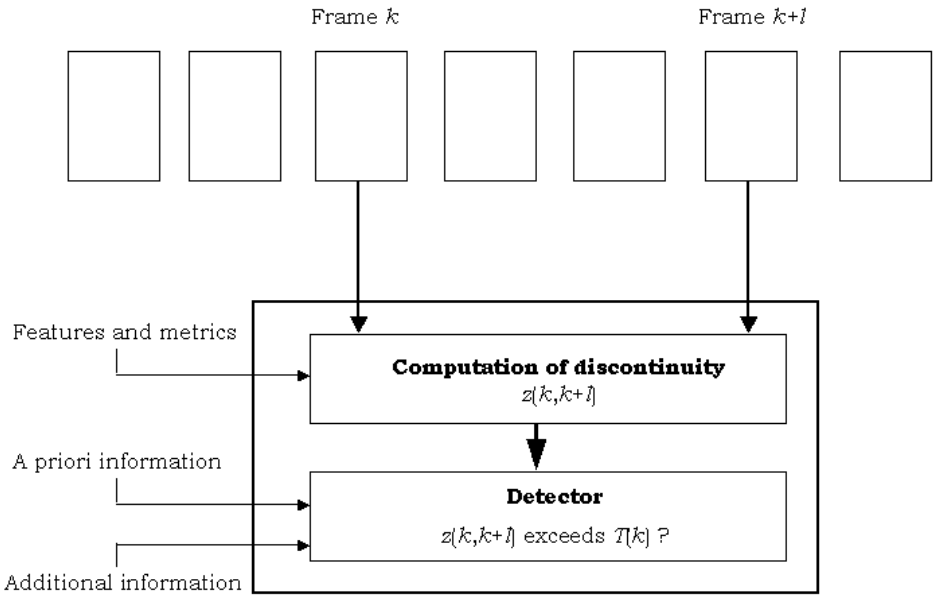

Figure 4. Representation of the shot boundary detector proposed in [14].

Another type of useful additional information may result from observation of the characteristic behavior of some visual features for frames surrounding a shot boundary for the case of gradual transitions. For example, since a dissolve is the result of mixing the visual material from two neighboring shots, it can be expected that variance values measured per frame along a dissolve ideally reveal a downwards-parabolic pattern [22]. Hence, the decision about the presence of a dissolve can be supported by investigating the behavior of the intensity variance in the "suspected" series of frames.

Further improvement of the detection performance can be obtained by taking into account a priori information about the presence or absence of a shot boundary at a certain time stamp along a video. The difference between additional information and a priori information is that the latter is not based on any measurement performed on the video. An example of a priori information is the dependence of the probability for shot boundary occurrence on the number of elapsed frames since the last detected shot boundary. While it can be assumed zero at the beginning of a shot, this

probability grows and converges to the value 0.5 as the number of frames in the shot increases. The main purpose of this probability is to make the detection of one shot boundary immediately after another one practically impossible and so to contribute to a reduction of false detection rate.

Combining measurements of discontinuity values with additional and a priori information may result in a robust shot-boundary detector, e.g. by continuously adapting the detection threshold $T(k)$ for each frame k. Statistical decision theory is employed to obtain $T(k)$ based on the criterion that average probability for detection mistakes is minimized. To this aim, all detection errors (e.g., both missed and false detections) are treated equally, thus the detector performance is determined by the average probability that any of errors occurs. This average probability is expressed in terms of the probability of missed detection and of the probability of false detection. The necessary likelihood functions are pre-computed using training data. Both additional information (for cuts and dissolves) and a priori information are taken into account in the computation of the average probability, while discontinuity values are computed by using a block-matching, motion compensated procedure [31]. The input sequence is assumed to be a MPEG partially decoded sequence, i.e. a DC sequence. This makes possible to limit the block dimensions to 4 x 4 pixels. The system performance is evaluated by using five test sequences (different from those employed for training) belonging to four video categories: movies, soccer game, news, and commercials. The authors report a 100% precision and recall for cuts, 79% precision and 83% recall for dissolve detection[1].

Another probabilistic approach was presented in [21], where Li et al. propose a temporal segmentation method based on spatial–temporal joint probability image (ST-JPI) analysis. Here, joint probabilities are viewed as a similarity estimate between two images (only luminance images are considered in the paper). Given two images $A(x, y)$ and $B(x, y)$, a joint probability image (JPI) is a matrix whose element value $JPI_{A,B}(i_1, i_2)$ is the probability that luminance i_1 and i_2 appear at the same position in image A and image B, respectively. Each element of a JPI corresponds to an intensity pair in two images. The distribution of the values in a JPI maps the correlation between two images: for two identical images, the JPI shows a diagonal line, while for two independent images the JPI consists of a uniform distribution. Because of the high correlation between video frames within a single shot, a JPI derived from two frames belonging to the same shot usually has a narrow distribution along the diagonal line. On the contrary, since narrative and visual content changes between two consecutive shots, then a uniform distribution is expected in the JPI. Thus the JPI behavior may be used to develop transition detection methods.

In particular, a spatial-temporal joint probability image (ST-JPI) is defined as a series of JPIs in chronological order, with all JPIs sharing the same initial image. The ST-JPI reflects the temporal evolution of video contents. For example, if a ST-JPI is derived between frames 0 e T, and a cut happens within this frame interval, the JPIs before the cut have very limited dispersion from the diagonal line, while after the cut uniform JPIs are usually obtained. The shift from narrow dispersion JPIs to uniform JPIs

[1] Recall and precision are popular quality factors whose formal definition is given in the following Sect. 4.

happens instantaneously at the cut position. By estimating the uniformity of JPIs, cuts can be detected and reported.

Detection of gradual transitions is also obtained with this approach, even if in a more complicated way. In particular, two kinds of gradual transition are considered, *cross* transitions and *dither* transitions. During a cross transition, every pixel value gradually changes from one shot to another, while during a dither transition a small portion of pixels abruptly change from pixels values from the first shot to those of the second shot every moment. With time, more and more pixels change until all of the pixels change into the second video shot. Wipes, fades and various types of dissolve may be described in terms of this scheme.

Template ST-JPIs are derived both for cross transitions and dither transitions. The detection of gradual transitions is performed by analyzing the pattern match between model ST-JPIs and the ST-JPI derived for the frame interval under consideration. Experiments performed on several digital videos of various kind gave the following results: 97% (recall) and 96% (precision) for cuts, 82% and 93% for cross transitions, 75% and 81% for dither transitions.

The algorithm proposed in [25] is based on the coherence analysis of temporal slices extracted from the digital video. Temporal slices are extracted from the video by slicing through the sequence of video frames and collecting temporal signatures. Each of these slices contains both spatial and temporal information from which coherent regions are indicative of uninterrupted video partitions separated by camera breaks (cuts, wipes and dissolves). Each spatiotemporal slice is a collection of scans, namely horizontal, vertical or diagonal image stripes, as a function of time. The detection of a shot boundary therefore becomes a problem of spatiotemporal slice segmentation into regions each of a coherent rhythm. Properties could further be extracted from the slice for both the detection and classification of camera breaks. For example, cut and wipes are detected by color-texture properties, while dissolves are detected by mean intensity and variance. The analysis is performed on the DC sequence extracted from a MPEG video. The approach has been tested by experiments on news sequences, documentary films, movies, and TV streams, with the following results: 100% (recall) and 99% (precision) for cuts, 75% and 80% for wipes, 76% and 77% for dissolves.

Finally, a fuzzy theoretic approach for temporal segmentation is presented in [16], with a fusion of various syntactic features to obtain more reliable transition detection. Cuts are detected using histogram intersection, gradual changes are detected using a combination of pixel difference and histogram intersection, while fades are detected using a combination of pixel difference, histogram intersection and edge-pixel-count. Frame-to-frame differences of these properties are considered as the input variable of the problem expressed in fuzzy terms. In particular, the linguistic variable “inter-frame-difference” is fuzzified so that it can be labeled as “negligible”, “small”, “significant”, “large” or “huge”. The values of metric differences are represented as these linguistic terms. To this aim, appropriate class boundaries and membership functions must be selected for each category. This is made by modeling the interframe property difference through the Rayleigh distribution. The appropriateness of this model has been tested by fitting Rayleigh distribution (chi-square test) to interframe difference data for about 300 video sequences of various kind, having 500-5000 frames each.

Fuzzy rules for each property are derived by taking into account the current interframe difference, the previous interframe difference and the next interframe difference.

3. A MLP-BASED TECHNIQUE

As already stated, one of the critical aspects of most of the techniques discussed in previous section is the determination of thresholds or, in general, the definition of criteria of detection and classification of the transitions. Moreover, most of the techniques present in literature are strongly dependent on the kind of sequences analyzed. To cope with these problems we propose the use of a neural network that analyzes the sequence of interframe metric values and is able to detect shot transitions, also producing a coarse classification of the detected transitions. This approach may be considered a generalization of the MLP-based algorithm already proposed in [2].

3.1 SHORT NOTES ON NEURAL NETWORKS

In the last decades, neural networks [29] have been successfully used in many problems of pattern recognition and classification. Briefly, a neural network is a set of units or nodes connected by links or synapses. A numeric weight is associated to each link, the set of weights represents the memory of the network, where knowledge is stored. The determination of these weights is done during the learning phase. There are three basic classes of learning paradigms [15]: supervised learning (i.e. performed under an external supervision), reinforcement learning (i.e. through a trial-and-error process), and unsupervised learning (i.e. performed in a self-organized manner).

The network interacts with the environment in which it is embedded through a set of input nodes and a set of output nodes, respectively. During the learning process, synaptic weights are modified in an orderly fashion so as input-output pairs fit a desired function. Each processing unit is characterized by a set of connecting links to other units, a current activation level and an activation function used to determine the activation level in the next step, given the input weights.

A multilayer perceptron or MLP exhibits a network architecture of the kind shown in fig. 5. It is a multilayer (i.e. the network units are organized in the form of layers) feedforward (i.e., signals propagate through the network in a forward direction) neural network characterized by the presence of one or more hidden layers, whose nodes are correspondingly called hidden units. This network architecture, already proposed in the fifties, has been applied successfully to solve diverse problems after the introduction[17], [29] of the highly popular learning algorithm known as error *back-propagation* algorithm. This supervised learning algorithm is based on the error-correction learning rule.

Basically, the back-propagation process consists of two passes through the different network layers, a *forward* pass and a *backward* pass. In the forward pass, an input vector (training pattern) is applied to the input nodes, and its effect propagates through the network, layer by layer, so as to produce a set of outputs as the actual response of the network. In this phase the synaptic weights are all fixed. During the backward pass, the synaptic weights are all

adjusted in accordance with the error-correction rule. Specifically, the actual response of the network is subtracted from the desired response to produce an error signal. This error signal is then propagated backward through the network, and the synaptic weights are adjusted so as to make the actual response of the network move closer to the desired response. The process is then iterated until the synaptic weights stabilize and the error converge to some minimum, or acceptably small, value. In practical applications, learning results from the many presentations of a prescribed set of training examples to the network. One complete presentation of the entire training set is called an *epoch*. It is common practice to randomize the order of presentation of training examples from one epoch to the next.

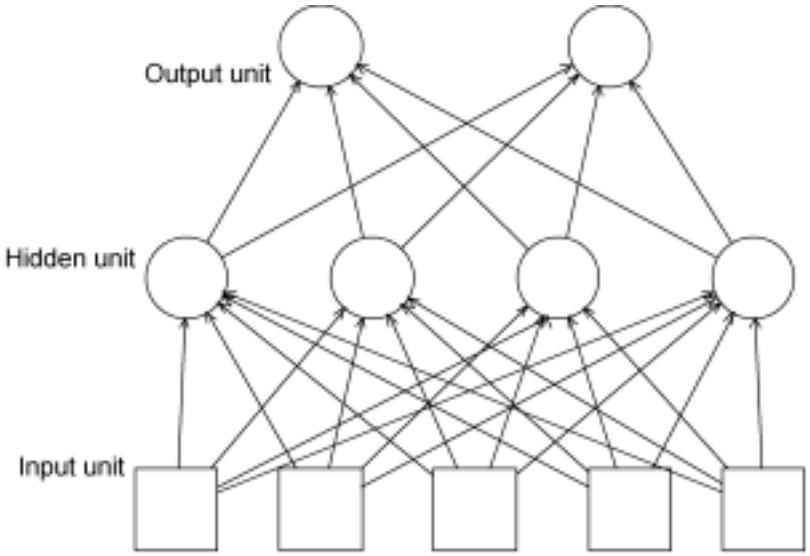

Figure 5. An example of multilayer perceptron with one hidden layer.

3.2 USE OF MLPs IN TEMPORAL SEGMENTATION

We propose the use of a MLP with an input layer, an hidden layer and an output layer, whose input vector is a set of interframe metric difference values. The training set is made up by examples extracted from sequences containing abrupt transitions, gradual transitions or no transition at all. We adopted the bin-to-bin luminance histogram difference as interframe metric. This choice is due to the simpleness of this metric, to its sufficient representativity with respect to both abrupt and gradual transitions, and to its ability to provide a simple interpretation model of the general video content evolution. As an example, figure 6 illustrates the evolution of the bin-to-bin metric within a frame interval of 1000 frames, extracted from a soccer video sequence. Five cuts and four dissolves are present in the sequence, and all these shot boundaries are clearly present in the figure. In the same figure it is also evident as gradual transitions are sometimes very difficult to distinguish from intrashot sequences (e.g. compare A and B in the figure).

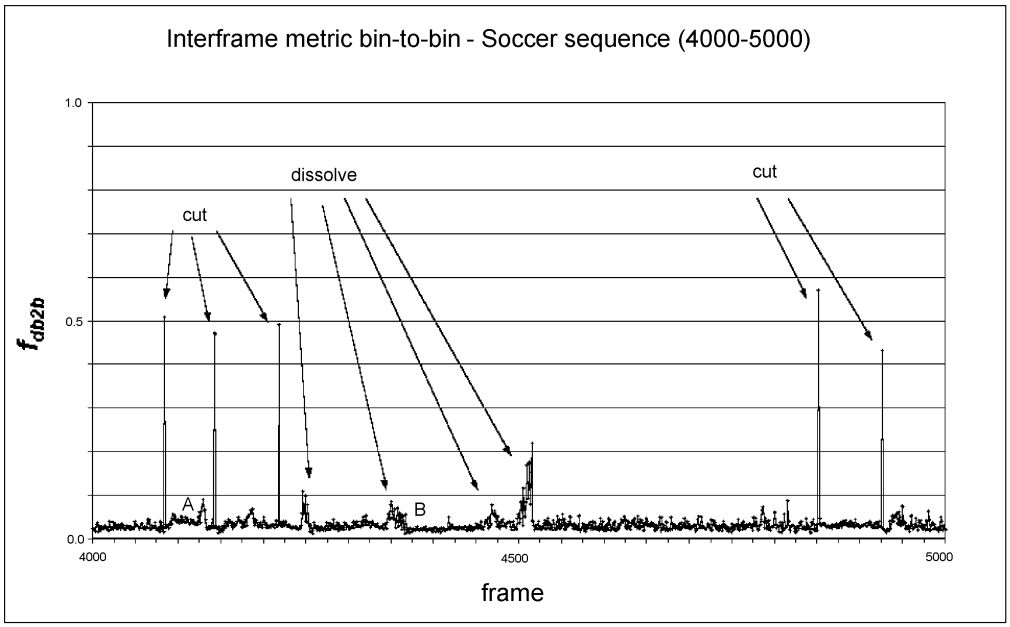

Figure 6. Interframe bin-to-bin metric for a soccer sequence.

The MLP's input units are fed with the values of the HDM computed within a temporal windows of several frames (the details about the choice of the number of input and hidden units will be provided in the next section). The output units are three, each of them representing one of three distinct possibilities: abrupt transition, gradual transition or no break at all. Output units assume real values in the range 0 – 1. Table 1 shows the output values used during the training process, in correspondence to the following input configurations: abrupt transition at the center of the temporal window, gradual transition with its maximum located near the center of the temporal window, no-transition.

Table 1. Output values of the MLP during the training phase

	O_1	O_2	O_3
Abrupt transition	0.99	0.01	0.01
Gradual transition	0.01	0.99	0.01
No transition	0.01	0.01	0.99

During the learning phase, the position of each transition in the input video sequence is known, thus defining the correct triple of output values for each position of the input temporal window, i.e. for each input pattern in the training set. The above described back-propagation process therefore allows for the weights adjustment. When the trained network analyzes an unknown input sequence, at each step of the analysis the highest value of the output triple determines the detection of an abrupt or gradual transition, or the absence of transitions, in the analyzed temporal window. As a consequence, explicit threshold values are not required for the decision.

4. PERFORMANCE EVALUATION OF TEMPORAL SEGMENTATION TECHNIQUES

In this section we give some guidelines for the performance evaluation of temporal segmentation techniques. In particular, we define a test bed made of two techniques, respectively named T1 and T2, used as comparison terms and suitable for detection of abrupt and abrupt/gradual transitions, respectively. Both T1 and T2 techniques require a manual or semi-automatic work for adapting thresholds and other parameters to the feature of input sequences, in order to obtain performances near to the optimum for both of them. It is with these optimal or near-to-optimal performances that a new temporal segmentation technique can be compared. As an example, we will evaluate the performance of our MLP-based approach. The design of this evaluation framework is part of a project on video temporal segmentation currently under development at the *Computer Science and Artificial Intelligence* lab of the University of Palermo.

We also describe the dataset we used for the evaluation. We used four video sequences representative of different types of video. In particular, we considered two sport sequences (soccer and F1 world championship), one news sequence and one naturalistic documentary. Details on test sequences are reported in table 2.

Table 2. Characteristics of dataset.

Sequence	**Frame rate (fps)**	**# of frames**	**# of abrupt transitions**	**# of gradual transitions**	**Total # of transitions**
Soccer	25	19260	115	51	166
F1	25	20083	106	14	120
News	25	20642	122	6	128
Nature	25	25016	150	12	162
	Total	85001	493	83	576

Most of the gradual transitions are dissolves. A few wipes are present and they are mainly in the F1 sequence. As previously stated both the T2 technique and the MLP-based one are aimed to the detection of a gradual transition but cannot discriminate between different kinds of gradual transitions.

In order to evaluate the performance of a segmentation algorithm we should primarily consider correctly detected transitions, false detections and missed transitions. Then we should analyze the performance with respect to gradual transition classification accuracy.

Recall and precision factors are normally used to evaluate performance [28] of transition detection techniques. These quality factors are defined as follows:

$$P_c = \frac{n_c}{n_c + n_f}; \qquad R_c = \frac{n_c}{n_c + n_m} \tag{25}$$

where n_c is the number of transitions correctly detected, n_f is the number of false positives and n_m is the number of missed detections. Ideally, $n_f = n_m = 0$

so that both the factors are 1. In our test bed we found useful a new quality factor defined in the following way:

$$Qr = \frac{n_c}{n_c + n_m + n_f} = \frac{n_c}{n_c + n_{tot} - n_c + n_f} = \frac{n_c}{n_{tot} + n_f} \qquad (26)$$

where n_{tot} is the total number of transitions and $n_m = n_{tot} - n_c$ .This quality factor takes simultaneously into account the ability to avoid both false and missed detections.

To evaluate the performance of transition classification techniques a different quality factor is needed. Assume n_{cc} as the number of abrupt transitions detected and correctly classified, n_{cg} as the number of gradual transitions detected and correctly classified and n_{sw} the number of transitions detected but misclassified. We can define the following quality factor:

$$I_{sw} = \frac{n_{cc} + n_{cg}}{n_{cc} + n_{cg} + n_{sw}} \qquad (27)$$

This quality factor is 1 if $n_{sw}=0$. Note that, as previously stated, I_{sw} is only a measure of transition classification accuracy and then cannot replace the quality factor Q_r that is a measure of transition detection performance. The two factors should always be used together to evaluate the performance of a system.

4.1 PERFORMANCE EVALUATION OF THE TECHNIQUE T1

We used the interframe metrics bin-to-bin, chi-square and histogram correlation, and called **T1** the technique whose result coincides, in each analyzed case, with the output of the best performing of the three methods. We evaluated the quality factor Q_r for the different HDM used varying the value of the threshold.

Using the annotated dataset described in the previous section we were able to determine for each sequence the optimal threshold. For example, in figures 7 and 8 is reported the quality factor Q_r as a function of the threshold using the bin-to-bin metric, respectively for soccer and nature videos. Obviously, as the threshold increases the number of false detections decreases but the number of missed detections increases. From figures 7 and 8 it is possible to see as Q_r for the nature video is higher, meaning a better performance, than Q_r for the soccer video. This is due to the larger presence of gradual transitions in soccer video that are misdetected.

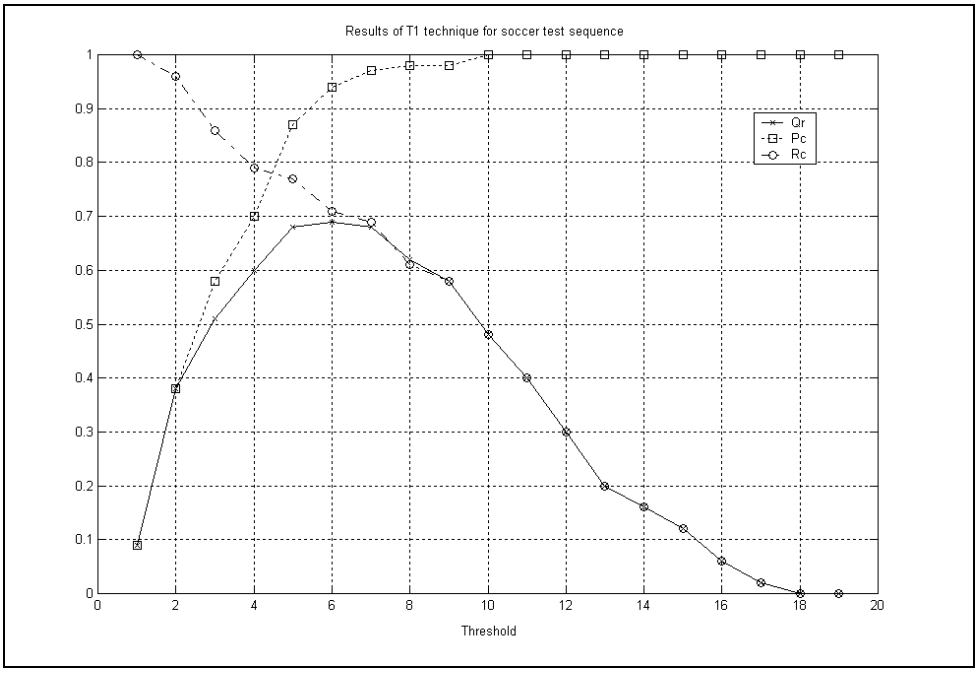

Figure 7. Performance of abrupt detection technique T1 for the *soccer* test sequence.

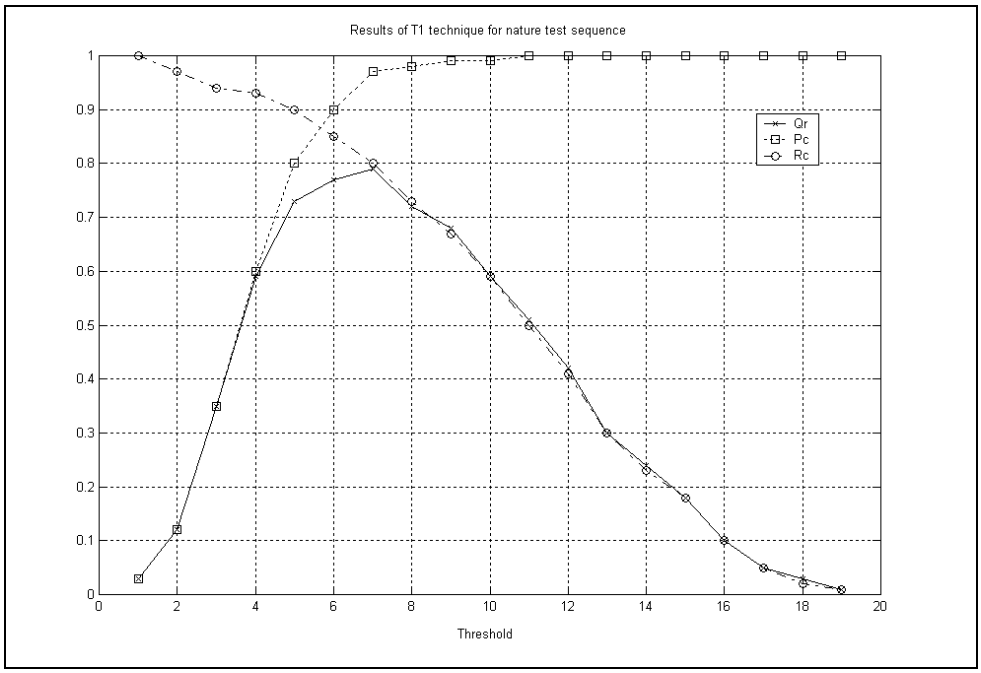

Figure 8. Performance of abrupt detection technique T1 for the *nature* test sequence.

For the other sequences and for the other metrics we obtained very similar shapes of the curves of figures 7 and 8. In tables 3-5 it is summarized the

performance of this technique on the four different test video and using three different histogram metrics in correspondence to the optimal value of the threshold:

Table 3. Performance using the b2b metric

Sequence	**Optimal threshold**	***Qr***
Soccer	0.30	0.68
News	0.30	0.76
F1	0.30	0.72
Nature	0.35	0.79

Table 4. Performance using the chi-square metric

Sequence	**Optimal threshold**	***Qr***
Soccer	0.35	0.62
News	0.35	0.74
F1	0.40	0.69
Nature	0.40	0.75

Table 5. Performance using the histogram correlation metric.

Sequence	**Optimal threshold**	***Qr***
Soccer	0.20	0.64
News	0.25	0.74
F1	0.25	0.71
Nature	0.25	0.77

It should be noted that the bin-to-bin metric, despite of its simpleness, exhibits the best behavior for the proposed input sequences.

4.2 PERFORMANCE EVALUATION OF THE TECHNIQUE T2

To evaluate the performance of a temporal segmentation technique in presence of gradual transitions, as a term of comparison we implemented a multi-threshold technique, called **T2**, inspired to the algorithm presented in [7].The technique T2, similarly to other techniques present in literature, is based on the observation that the video properties relevant for the shot transition detection are intrinsically local, i.e. depending on the behavior of the interframe metric within a temporal window of a few decades of frames. Within the window, a shot boundary may be detected and classified analyzing statistical factors like the mean and the standard deviation of the sequence of interframe metric values. In particular, the local standard deviation calculated when an abrupt transition is present within the window is very different from those obtained in case of a gradual transition, due to the more distributed changes of the interframe metric values in latter case.
Table 6 shows the definition and the meaning of thresholds and other factors intervening in the transition detection process.

Table 6. Factors intervening in transition detection (technique T2)

Parameter	**Description**
ω	Temporal window of analysis
$T_{hc} = \mu_\omega + \alpha_c\sigma_\omega$	High threshold for abrupt transitions
$T_{hg} = \mu_\omega + \alpha_g\sigma_\omega$	High threshold for gradual transitions
$T_{lc} = \beta_c\mu_T$	Low threshold for abrupt transitions
$T_{lg} = \beta_g\mu_T$	Low threshold for gradual transitions

In the above definitions, μ_ω and σ_ω respectively are the mean and the standard deviation of the interframe metric values within the temporal window ω, while μ_T is the global mean of the interframe metric. The coefficients α_c, α_g, β_c and β_g are determined experimentally, as well as the optimal size of the window ω.

The technique T2 operates in the following way. Shot boundary occurrences are detected in correspondence to the central pair of frames within the current temporal window. To detect a cut, the corresponding interframe metric value must be the maximum of values belonging to the current window, and also greater than the local thresholds T_{hc} and T_{hg}. Moreover, it must also be greater than the global threshold T_{lc}, whose value is depending, through the coefficient β_c, on the global mean of the interframe metric. Similarly, a gradual transition is detected in correspondence to the central pair of frames within the current temporal window if the corresponding interframe metric value is a local maximum, and if it is greater than T_{hg} and T_{lg}, but not greater than T_{hc}.

It should be noted that verifying the presence of a local maximum before declaring a transition is a condition necessary to avoid the detection of more than one transition within the same temporal window. Moreover, two different local thresholds, T_{lc} and T_{lg}, have been introduced to overcome the problem of local perturbation effects, that could cause exceeding the threshold T_{hg}, as it has also been experimentally verified. T_{lc} and T_{lg} must be different, because the average metric values related to abrupt transitions are very different from the average metric values related to gradual transitions.

Figure 9 shows the behavior of the interframe metric bin-to-bin for the soccer sequence, outlining the variation of local thresholds T_{hc} and T_{hg} (the values $\omega = 21$, $\alpha_c = 4$, and $\alpha_g = 1.5$ have been assumed). It should be noted that both the thresholds are exceeded in correspondence to abrupt transitions, while T_{hg} (but not T_{hc}) is sometimes exceeded when searching for gradual transitions.

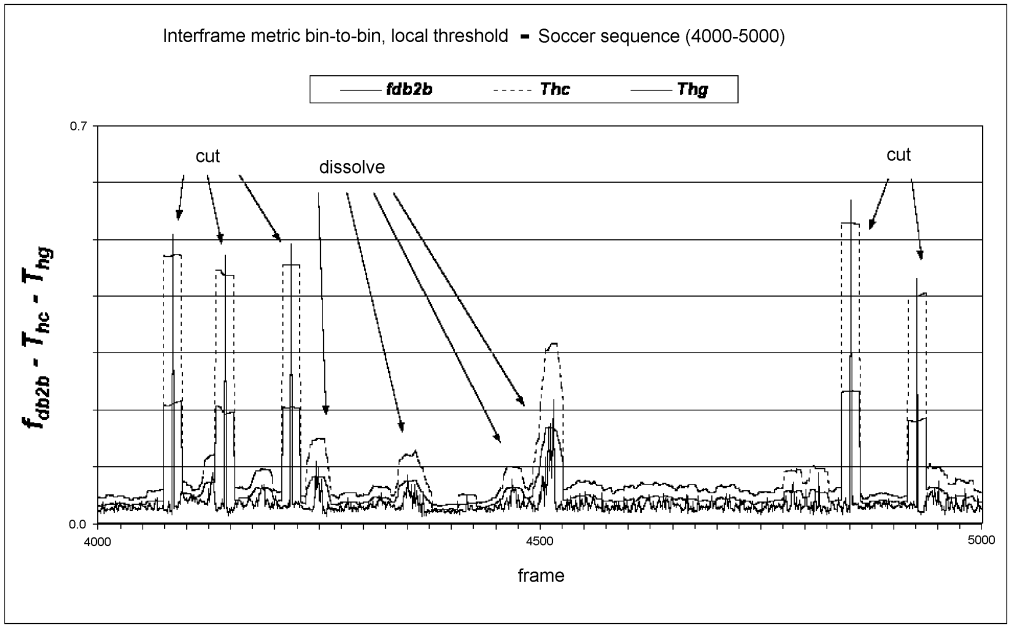

Figure 9. Interframe bin-to-bin metric and local thresholds for a soccer sequence.

These aspects are even more evident in Figure 10, where the metric bin-to-bin is shown for a shorter sequence of 250 frames extracted from the same video sequence. In this figure two cuts of different value are reported, along with a gradual transition (dissolve) characterized by prominent fluctuations.

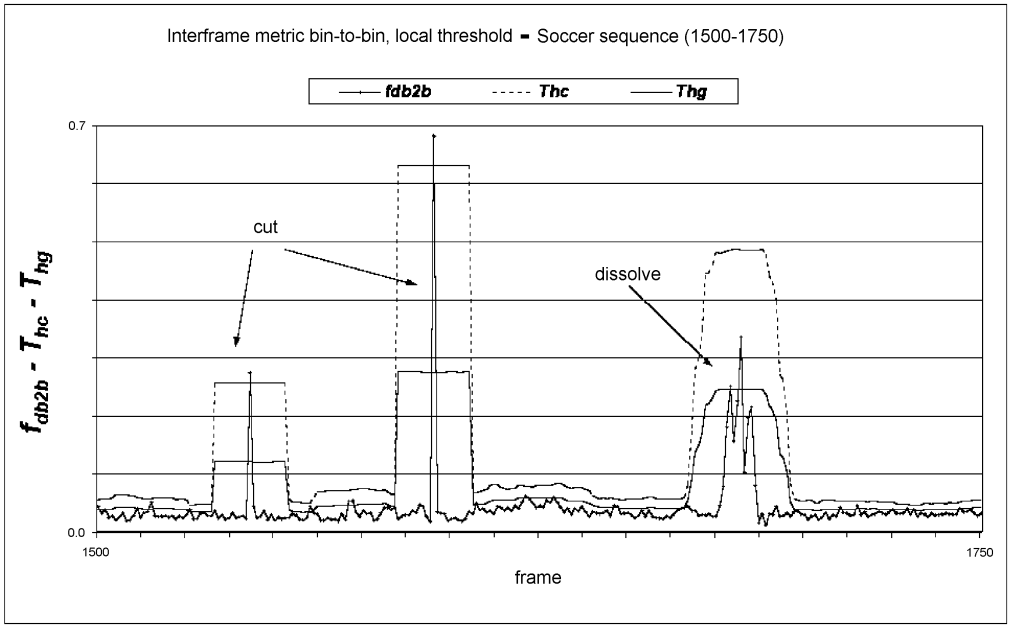

Figure 10. Interframe bin-to-bin metric and local thresholds for a soccer sequence.

As the technique T2 is largely dependent on various parameters and thresholds, we carefully evaluated its performance by letting the parameters vary in quite large intervals, with small variation steps (intervals and steps were determined on the basis of preliminary experiments):

Table 7. Range of parameters used for experiments.

Parameter	Range	Step
w	15 – 49	2
α_c	2.0 – 5.0	0.5
α_g	1.0 – 3.0	0.5
β_c	2.0 – 5.0	0.5
β_g	1.0 – 4.0	0.5

In Tables 8-10 we report the best result obtained for each metric and each sequence and the set of parameters that led to the result:

Table 8. Optimal result using bin-to-bin metric

Sequence	Qr	I_{sw}	W	α_c	α_g	β_c	β_g
Soccer	0.84	0.94	21	4.0	1.5	3.0	2.0
F1	0.82	0.92	25	3.5	1.5	3.0	2.0
News	0.83	0.97	21	3.5	2.0	3.0	2.0
Nature	0.83	0.96	21	3.5	2.0	3.0	2.0

Table 9. Optimal result using chi-squares metric

Sequence	Qr	I_{sw}	W	α_c	α_g	β_c	β_g
Soccer	0.72	0.80	29	3.5	2.0	3.5	3.0
F1	0.75	0.76	27	3.0	2.0	3.0	2.5
News	0.78	0.86	27	3.5	2.5	3.5	2.5
Nature	0.71	0.88	25	3.0	2.5	3.5	3.0

Table 10. Optimal result using correlation metric

Sequence	Qr	I_{sw}	W	α_c	α_g	β_c	β_g
Soccer	0.76	0.90	25	4.0	2.5	2.0	1.5
F1	0.77	0.92	29	4.0	2.5	2.5	1.5
News	0.78	0.86	27	4.0	2.5	2.0	1.5
Nature	0.75	0.79	25	4.5	2.5	2.5	2.0

4.3 PERFORMANCE EVALUATION OF THE MLP-BASED TECHNIQUE

To evaluate the performance of the proposed MLP-based technique we used the soccer and news video as training set and the nature and F1 video as test set. After each training epoch, the MLP performance is evaluated both on the training and test set.

The MLP architecture has been determined experimentally. The number of input units is obviously related to the probable number of frames belonging to a gradual transition. From video data available in our dataset we observed that the duration of gradual transitions is usually between 10 and 20 frames, i.e. below one second at 25 fps, a result in accord with modern electronic editing tools. We then tried several networks with 21 or 31 input units, a choice allowing to capture an entire transition without the risk of enclosing more than one transition within an unique temporal window. A negative consequence of this choice is the impossibility of dealing with very short shot transitions, like those ones typical of some commercial or musical sequences. However, we assumed our dataset more representative of kinds of video sequences of interest for video segmentation.
We tried network architectures with different numbers of hidden units. If the number of input units is intuitively related to the duration of gradual transitions, the choice of the number of hidden units is critical, in the sense that a limited number of hidden units can limit the ability of the network to generalize, wile an excessive number of hidden units can induce problems of "overtraining". Even this aspect has been investigated experimentally.

In Table 11 the performance of different MLP architectures after 50 training epochs is reported. The network architecture is indicated with three numbers indicating, respectively, the number of input, hidden and output units. The performance is expressed in terms of the quality factors previously defined, both on training set and testing set:

Table 11. Performance of MLPs of different architecture.

	(21,10,3)		**(21,40,3)**		**(21,100,3)**		**(31,40,3)**	
	Q_r	I_{sw}	Q_r	I_{sw}	Q_r	I_{sw}	Q_r	I_{sw}
Soccer	0.87	0.92	0.95	0.99	0.86	0.90	0.99	0.99
News	0.84	0.94	0.94	1.00	0.84	0.94	0.94	1.00
F1	0.84	0.90	0.89	0.97	0.82	0.91	0.88	0.97
Nature	0.80	0.91	0.91	0.99	0.77	0.88	0.89	0.96

From inspection of Table 11 it is evident as the best performance is obtained using a network with 21 input units and 40 hidden units. As expected, the network with 10 or 100 hidden units does not perform well. Since the use of 31 input units does not give significant performance improvement on our dataset, the (21-40-3) network has been selected for its lower computational load and for its better ability to detect temporally close transitions.

4.4 COMPARISON OF DIFFERENT TECHNIQUES

In this section we compare the three techniques we described in the previous sections. In Table 12 a summary of most important results is reported.

Table 12. Performance comparison among the described techniques.

	T1		**T2**		**MLP**	
	Q_r	I_{sw}	Q_r	I_{sw}	Q_r	I_{sw}
Soccer	0.68	---	0.84	0.94	0.95	0.99
News	0.76	---	0.82	0.92	0.94	1.00
F1	0.72	---	0.83	0.97	0.89	0.97

Nature	0.79	---	0.83	0,96	0.91	0.99

Note that the results of techniques T1 and T2 were obtained using a choice of optimal parameters (for T1 and T2). Similarly the results of the MLP-based technique for the sequences soccer and news are computed on the training set. These results are then to be considered optimal and, in practice, we should expect worse performance.
On the other side, it is important to note that the performance of the MLP-based classifier on the sequences F1 and Nature, that are outside of the training set, is not based on the knowledge of the ground truth and then may be considered representative of the algorithm behavior. If we used the optimal parameters computed for techniques T1 and T2 for the sequences soccer and news to analyze the sequences F1 and Nature, we would obtain a worse performance.
Finally, although there is no theoretical justification, we note that, independently of the decision technique, best results are obtained using bin-to-bin metric.

5. CONCLUSIONS

in the last decade, with the increasing availability of digital video material, advanced applications based on storage and transmission of video data (digital libraries, video communication over the networks, digital TV, video-on-demand, etc...) began to appear. Their full utilization depends on the capability of information providers to organize and structure video data, as well as on the possibility of intermediate and final users to access the information.

From this point of view, automatic video temporal segmentation is a necessary task. As we showed in this chapter, much work has been done to individuate algorithms and methods in this area, but results are often too much depending on the peculiarity of input data. Research effort is still necessary to find generally effective and computationally manageable solutions to the problem.

REFERENCES

[1] P. Aigrain, and P. Joly, “The Automatic Real Time Analysis of Film Editing and Transition Effects and its Applications”, Computer & Graphics, vol. 18, n. 1, pp. 93-103, 1994.

[2] E. Ardizzone, and M. La Cascia, “Automatic Video Database Indexing and Retrieval”, Multimedia Tools and Applications, vol. 4, pp. 29-56, 1997.

[3] F. Arman, A. Hsu, and M. Y. Chiu, “Feature Management for Large Video Databases”, Proceedings IS&T/SPIE Conf. Storage and Retrieval for Image and Video Databases I, vol. SPIE 1908, pp. 2-12, 1993.

[4] Y. A. Aslandogan, and C. T. Yu, “Techniques and Systems for Image and Video Retrieval”, IEEE Transactions On Knowledge and Data Engineering, vol. 11, n. 1, pp. 56-63, 1999.

[5] J. M. Corridoni, and A. Del Bimbo, “Structured Representation and Automatic Indexing of Movie Information Content”, Pattern Recognition, vol. 31, n. 12, pp. 2027-2045, 1998.

[6] A. Dailianas, R. B. Allen, and P. England, “Comparison of Automatic Video Segmentation Algorithms”, Proceedings of SPIE Photonics West, 1995.SPIE, vol. 2615, pp. 2-16, Philadelphia, 1995.

[7] R. Dugad, K. Ratakonda, and N. Ahuja, “Robust Video Shot Change Detection”, IEEE Second Workshop on Multimedia Signal Processing, pp. 376-381, Redondo Beach, California, 1998.

[8] R. M. Ford, C. Robson, D. Tample, and M. Gerlach, “Metrics for Scene Change Detection in Digital Video Sequences”, IEEE International Conference on Multimedia Computing and Systems (ICMCS '97), pp. 610-611, Ottawa, Canada, 1997.

[9] U. Gargi, R. Kasturi, and S. H. Strayer, “Performance Characterization of Video-Shot-Change Detection Methods”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, n. 1, pp. 1-13, 2000.

[10] R. C. Gonzalez, and R. E. Woods, “Digital Image Processing”, Addison-Wesley, 1992.

[11] B. Gunsel, A.M. Ferman, and A.M. Tekalp, “Temporal video segmentation using unsupervised clustering and semantic object tracking”, Journal of Electronic Imaging vol. 7, n. 3, pp 592-604, 1998

[12] A. Hampapur, R. Jain, and T. Weymouth, “Production Model Based in Digital Video Segmentation”, Multimedia Tools and Applications, vol. 1, n. 1, pp. 9-46, 1995.

[13] A. Hampapur, R. Jain, and T. Weymouth, “Indexing in video Databases”, IS&T SPIE Proceedings: Storage and Retrieval for Image and Video Databases III, vol. 2420, pp. 292-306, San Jose, 1995.

[14] A. Hanjalick, “Shot Boundary Detection: Unraveled and Resolved?”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, n. 2, pp. 90-105, 2002.

[15] S. Haykin, “Neural Networks – A Comprehensive Foundation”, MacMillan College Publishing Company, 1994.

[16] R.S. Jadon, S. Chaudhury, K.K. Biswas, “A fuzzy theoretic approach for video segmentation using syntactic features”, Pattern Recognition Letters, vol. 22, pp. 1359-1369, 2001.

[17] T. Khanna, “Foundation of Neural Networks”, Addison-Wesley, 1990.

[18] I. Koprinska, and S. Carrato, “Temporal video segmentation: A Survey”, Signal Processing: Image Communication, vol. 16, pp. 477–500, 2001.

[19] D. Le Gall, “A Video Compression Standard for Multimedia Applications”, Commun. Of the ACM, vol. 34, n. 4, pp. 46-58, 1991.

[20] M. S. Lee, Y. M. Yang, S. W. Lee, “Automatic Video Parsing Using Shot Boundary Detection and Camera Operation Analysis”, Pattern Recognition, vol. 34, pp. 711-719, 2001.

[21] Z.-N. Li, X. Zhong, and M.S. Drew, “Spatial-temporal joint probability images for video segmentation”, Pattern Recognition, vol. 35, pp. 1847-1867, 2002.

[22] R. Lienhart, “Reliable transition detection in videos: A survey and practitioner's guide.” Int. J. Image Graph., vol. 1, n.3, pp. 469-486, Aug. 2001.

[23] A. Nagasaka, and Y. Tanaka, “Automatic Video Indexing and Full-Video Search for Objects Appearances”, Visual Databases Systems II, E. Knuth and L. M. Wegner (eds.), Elsevier Science Publ., pp. 113-127, 1992.

[24] M. R. Naphade, R. Mehrotra, A.M. Ferman, J. Warnick, T. S. Huang, and A. M. Tekalp, ”A High-Performance Shot Boundary Detection Algorithm Using Multiple Cues”, IEEE International Conference on Image Processing (ICIP ’98), vol. 1, pp. 884-887, Chicago, 1998.

[25] C.-W. Ngo, T.-C. Pong, and R.T. Chin, “Video Partitioning by Temporal Slice Coherency”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, n. 8, pp. 941-953, 2001.

[26] C. W. Ngo, T. C. Pong, and R. T. Chin, “Camera Breaks Detection by Partitioning of 2D Spatio-temporal Images in MPEG Domain”, IEEE International Conference on Multimedia Systems (ICMCS '99), vol. 1, pp. 750-755, Italy, 1999.

[27] T.N. Pappas, An adaptive clustering algorithm for image segmentation, IEEE Transaction on Signal Processing vol. 40 pp. 901-914, 1992

[28] N. V. Pathel, and I. K. Sethi, “Video Shot Detection and Characterization for Video Databases”, Pattern Recognition, Special Issue on Multimedia, vol. 30, pp. 583-592, 1997.

[29] S. J. Russell, and P. Norvig, “Artificial Intelligence – A Modern Approach”, Prentice Hall, 1995.

[30] K. Sethi, and N. V. Patel, “A Statistical Approach to Scene Change Detection”, IS&T SPIE Proceedings: Storage and Retrieval for Image and Video Databases III, vol. 2420, pp. 329-339, San Jose, 1995.

[31] B. Shahraray, “Scene change detection and content-based sampling of video sequences,” in Proc. IS&T/SPIE, vol. 2419, Feb. 1995, pp. 2–13.

[32] M. Wu, W. Wolf, and B. Liu, “An Algorithm for Wipe Detection”, IEEE International Conference on Image Processing (ICIP ’98), vol. 1, pp. 893-897, Chicago, 1998.

[33] L. Yeo, and B. Liu, “Rapid Scene Analysis on Compressed Video”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, n. 6, pp. 533-544, 1995.

[34] H. Yu, G. Bozdagi, and S. Harrington, “Feature-based hierarchical video segmentation”, IEEE International Conference on Image Processing (ICIP'97), Santa Barbara, 1997, pp. 498-501.

[35] R. Zabih, J. Miller, and K. Mai, “A Feature-Based Algorithm for Detecting and Classifying Scene Breaks”, Proceedings of ACM Multimedia ‘95, pp. 189-200, San Francisco, 1995.

[36] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, “An Integrated System for Content-Based Video Retrieval and Browsing”, Pattern Recognition, vol. 30, n. 4, pp. 643-658, 1997.

[37] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, “Automatic Partitioning of full-motion Video”, Multimedia Systems, vol. 1, n. 1, pp. 10-28, 1993.

[38] H. J. Zhang, C. Y. Low, and S. W. Smoliar, “Video Parsing Using Compressed Data”, Proceedings IS&T/SPIE, Image and Video Processing II, pp. 142-149, 1994.

INDEX