VIDEO SHOT BOUNDARY DETECTION BASED ON COLOR HISTOGRAM

Jordi Mas and Gabriel Fernandez

Digital Television Center (CeTVD)
La Salle School of Engineering, Ramon Llull University
Barcelona, Spain

Abstract

In TREC 2003 Video Track, the Digital Television Center of La Salle School of Engineering, Ramon Llull University, participated in the shot boundary detection task. In this paper we present a method for video shot boundary detection based on color histogram differences. The technique is able to differentiate abrupt shot boundaries by the analysis of color histogram differences and smooth shot boundaries by temporal color variation. The aim of the method is to provide a simple and fast algorithm able to work in real-time with reasonable high performances in a video indexing tool. In this paper we give an overview of the approaches and a summary of the results. The obtained results show that although the computational cost is reduced this fact does not prevent the method from having an overall precision of 84.7% and a recall of 80.6%.

1. Introduction

Video shot boundary detection has been deeply studied in recent years and has found applications in different domains like video indexing, video compression, video access and others. In the framework of a video indexing tool based on the MPEG7 content description standard, a fast and automatic temporal segmentation of video content is required in order to produce an accurate content description of each temporally homogeneous shot.

Video shot boundary detection algorithms have to challenge the difficulty of finding shot boundaries in the presence of camera and object motion and illumination variations. Moreover, different video shot boundaries may present very different appearances like abrupt temporal changes or smooth temporal transitions.

Shot boundary detection has been an area of active research. Many automatic techniques have been developed to detect frame transitions in video sequences. The simplest way of measuring the visual-content discontinuity is to compare the corresponding pixels between two frames [1]. A popular alternative to pixel-based approaches is using grayscale or color histograms as features [2]. Another characteristic feature that proved to be useful in detection shot boundaries is edge changes [3]. Other further methods use predefined models, objects, regions or spatio-temporal frame sub-sampling to detect camera breaks [4]. Hybrids of these techniques have also been investigated [5].

In this paper we present a simple and fast method for detecting video shot boundaries based on color histogram differences [6]. The algorithm firstly extracts abrupt shot discontinuities by the analysis of the color histogram difference, and then it detects possible gradual discontinuities for the following discrimination from motion activity as they behave a smooth discontinuity pattern. The simplicity of the method relies on the low complexity of the computation of the color histogram difference and that the most of the subsequent analysis is performed over the resulting 1-D signal.

The paper is organized as follows. After this introduction, Section 2 will describe the method for detecting abrupt video shot boundaries and will discuss the difficulty of smooth shot boundary detection. This section also will propose a method to overcome the detection of smooth changes. Section 3 gives an overview of results obtained with the methods explained in this paper. In Section 4, we introduce an audiovisual indexing tool developed by our team for cataloguing video content using the presented shot detection methods. Section 5 will comment some conclusions about the results obtained and some future improvements.

2. Shot boundary detection

2.1. Introduction

The basis of any shot boundary detection method in a video sequence consists in detecting visual discontinuities along the time domain. During this detection process, it is required to extract visual features that measure the degree of similarity between frames in a given shot. This measure, denoted as $g(n,n+k)$, is related to the difference or discontinuity between frame n and $n+k$ where $k \geq 1$. There exist different methods for computing the value of $g(n,n+k)$ in a video sequence, being one of the simplest the absolute difference between frames:

$$g(n,n+k)=\sum_{x,y}\left|I_n(x,y)-I_{n+k}(x,y)\right| \tag{2.1}$$

where *I(x,y)* is the image intensity level of the image at *x* and *y* position. Usually, methods based on absolute difference compare this value with a given threshold in order to determine the occurrence of a significant change in the image sequence. However, the measure of discontinuity *g(n,n+k)* computed in this way is very sensitive to luminance changes or object and camera motion, leading to a high ratio of false alarms.
An alternative to compare image values in a pixel level is to compare statistics of global color. Histograms capture the color distribution of an image. In some cases, luminance histogram is a sufficient measure to reach our aim. As we have mentioned before, the shot detection effectiveness depends on the suitable election of the similarity measure between consecutive frames. Unlike techniques based on pixel difference between adjacent frames, color histogram techniques reach a higher degree of independence to object motion within a frame sequence. On the other hand, we consider that the probability of having different consecutive shots with similar color histograms is very low. However, the method based on color histogram remains sensitive to camera motion such as panning or zooming. We will discuss this limitation and will present a method to overcome it later in this paper.

2.2. Histogram features

Luminance histogram considers only the luminance distribution within the image. However, as we are dealing with color images, we should define a color histogram. Besides, the color space and the quantization method chosen will decide the effectiveness of the color histogram descriptor.
Usually, digital images are represented in RGB color space. In our work, we have 24 bits/pixel images (8 bits for R, G and B components respectively). We define the color histogram as an array of *M* elements, where *M* is the number of different possible colors within the color space. If we compute the overall number of possible colors we realize that we rise to a high number of levels (2^{24} bins). Due to the limited response of human visual system we are not able to distinguish the whole levels of possible colors. A simple solution consist in considering only the most significant bits of each component RGB. As we will see later, our shot boundary method uses a color histogram in RGB space, where a quantization is done by eliminating the least significant bits of each component.
Figure 2.1 shows an example of the four most significant bits of each component RGB.

R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
G_7	G_6	G_5	G_4	G_3	G_2	G_1	G_0
B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0

Figure 2.1 Quantization over 8 bits RGB components (gray bits eliminated)

With this quantization method we have grouped all possible colors into 2^{12} different color levels in RGB space, which correspond to 4096 colors. This will be the number of levels of the histogram color array.

In Figure 2.2 and 2.3 we can appreciate the comparison between two images with 24 bits/pixel and 12 bits/pixel respectively. We have eliminated the four least significant bits of every component. Unless the least number of possible colors, we can see that these levels have been grouped according to their similarity.

(a)  (b) 

Figure 2.2 (a) 24 bits/pixel image (b) 12 bits/pixel image

As an alternative, we can evaluate the color histogram in the HSV space due to its similarity and perceptivity characteristics. This color space is defined according to human color perception. Perceptibly similar colors are situated within close quantization levels, otherwise dissimilar colors belong to different far quantization levels. In addition, similarity between two colors can be evaluated by the distance in this color space. Easily, a color can be determined by an user choosing the values of hue, saturation and value. As there is no transformation matrix for RGB/HSV conversion, we have to transform the corresponding values from RGB to HSV space with a conversion algorithm as source images are in RGB space. After this transformation, a number of bins are defined for the color histogram. As in RGB color space, we have a number of quantization levels according to the number of bits of each component. In that case, we are going to distinguish hue component from saturation and value component. The main reason is that visual human system is more sensitive to hue variations than saturation and value variations. For instance, we reach a number of 1024 possible colors by considering 16 levels for hue component, 8 levels for saturation and 8 levels for value component.

2.3. Color histogram differences

The presented method of shot boundary detection is based on the computation of differences of color histograms between frames as a measure of discontinuity. This difference can be computed as the sum of the absolute difference between the bin values,

$$d_{RGB}(X,Y)=\sum_{i=1}^{M}\left|h_x(i)-h_y(i)\right| \tag{2.2}$$

where h_x is the color histogram of image X which contains M different bins.

In our case, the evaluation of color histogram difference in the RGB space with M bins per histogram is considered. As we have mentioned in the section before, we have also the HSV color space to evaluate the color histogram difference. After different preliminary tests with HSV space, we have obtained no significant difference with regard to RGB color space which implied a noticeable improvement. Besides, a space conversion algorithm (RGB to HSV) is required, increasing the computational cost needed.
The shot boundary detection method is based on the difference between color histograms of frames belonging to a video sequence. This difference is computed as

$$HistDif[i]=\sum_{j=1}^{M}\left|h_i(j)-h_{i-1}(j)\right| \tag{2.3}$$

where h_i is the color histogram with M bins of frame i corresponding to the video sequence.
In figures 2.4 and 2.5 it is shown the result of computing the color histogram difference for different video sequences.

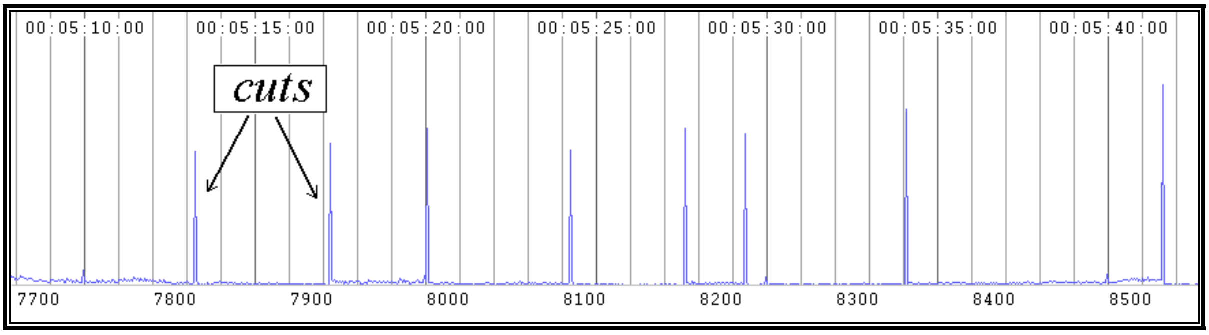

Figure 2.4 Color histogram differences signal of a frame sequence with cut boundaries

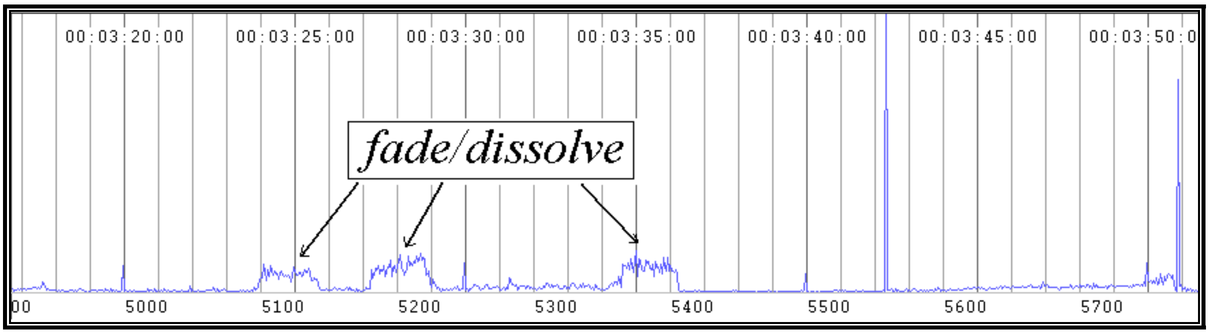

Figure 2.5 Color histogram differences signal of a frame sequence with cut boundaries

As we can see in the figures 2.4 and 2.5, different types of shapes could appear according to the video effect transition applied. Firstly, a peak appears when a large discontinuity occurs between histograms. This peak could be associated to an abrupt transition or cut. These cuts could be easily recognized from other video effects because they always present a big amplitude. Ideally, an abrupt change could be represented as a delta function

$$HistDif_{cut}[i] = \alpha_i \cdot \delta(i - i_{cut}) \tag{2.4}$$

where α_i represent the amplitude of delta function and i_{cut} is the frame number where the cut occurs.
Secondly, gradual transitions, fades and dissolves, appear with a lower level according to the smoothest variation of the color histogram within the sequence, although the amplitude signal is not maintained constant during the whole effect. Ideally, considering the histogram differences being constant within the transition between two shots we will have a rectangular function as follows

$$HistDif_{fade\ and\ dissolve}[i] = \beta_i \cdot rect\left(\frac{i - i_{fade\ and\ dissolve}}{T_{fade\ and\ dissolve}}\right) \tag{2.5}$$

$$rect(x) = \begin{cases} 1, & |x| \le 1/2 \\ 0, & |x| \le 1/2 \end{cases} \tag{2.6}$$

Finally, the presence of objects and camera movement will produce signals whose shape is very close to fades or dissolves. Therefore, we will have to closely analyze them in order to eliminate false alarms.

2.4. Boundary detection

2.4.1. Cut boundary detection

Once we have obtained the color histogram differences signal *HistDif[i]* of a video sequence, the next step consists in convolving this signal with a rectangular window of width *W*.

$$HistDif_{conv}[i] = HistDif[i] * \frac{1}{W} \cdot rect\left(\frac{i}{W}\right) \tag{2.7}$$

With this signal processing operation we are smoothing *HistDif[i]* signal so that small variations due to histogram difference computations are eliminated. Although we maintain a characteristic shape signals of cuts, fades and dissolves for later detection.

After the convolution processing, the cut, fade and dissolve characteristic signals have been modified. In the case of cuts, after convolution we obtain a rectangular shaped signal where the middle of the rectangle is considered the point or frame of abrupt transition.

$$HistDif_{conv}^{cut}[i] = \frac{\alpha_i}{W} \cdot rect\left(\frac{i - i_{cut}}{W/2}\right) \qquad (2.8)$$

In the case of presence of camera/object movement close to a cut, the signal *HistDif*$_{conv}$*[i]*, as we can see in figure 2.6, lose the rectangular shape, increasing the difficulty in its detection.

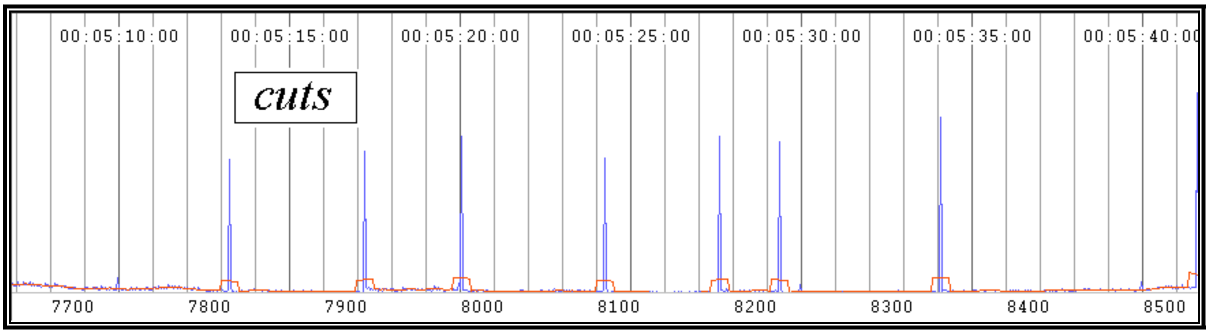

Figure 2.6 Color histogram difference signal (blue) and convolved signal with a rectangular window of *W* width (red)

Signals corresponding to fade and dissolve have been also modified. They appear as triangular shaped signals which belong to gradual transitions in a video sequence. The real shape of these signals can be seen in figure 2.7.

$$HistDif_{conv}^{fade/dissolve}[i] = \frac{\beta_i}{W} \cdot tri\left(\frac{i - i_{fade/dissolve}}{W/2}\right) \qquad (2.9)$$

$$tri(x) = \begin{cases} 1 - |x|, & |x| \le 1 \\ 0, & others \end{cases} \qquad (2.10)$$

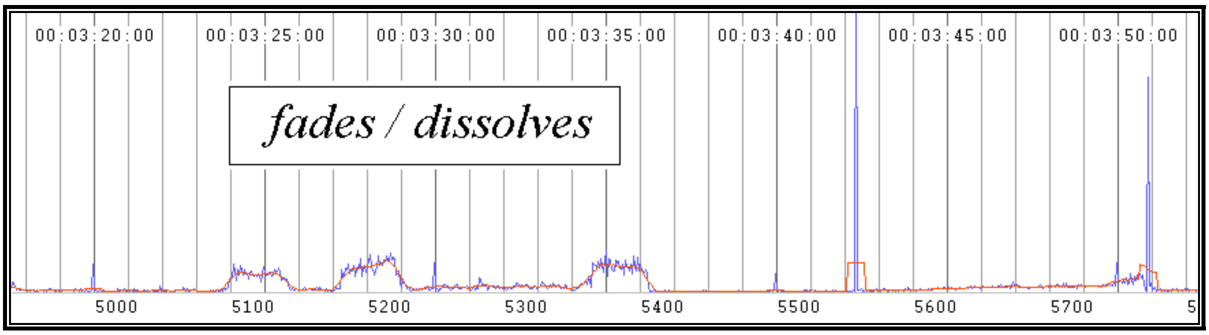

Figure 2.7 Color histogram difference signal (blue) and convolved signal with a rectangular window of *W* width (red)

Once defined the possible type of signals that can be found in the convolved signal of the color histogram differences, the next step consists in detecting cuts, fades and dissolves.
For the cut detection, we aim to detect and identify signals with rectangular pattern in the convolved signal of the color histogram difference *HistDif*$_{conv}$*[i]*. An effective method that reduces the distortion produced by camera/object motion is to apply the first derivative to *HistDif*$_{conv}$*[i]* signal.

$$HistDif_{conv}^{deriv}[i] = [1,-1] * HistDif_{conv}[i] \qquad (2.11)$$

As we can see in figure 2.8, applying the first derivative a positive peak followed by a negative peak are obtained. The distance between the two peaks is *W*, the width of the rectangular window used during the convolution.

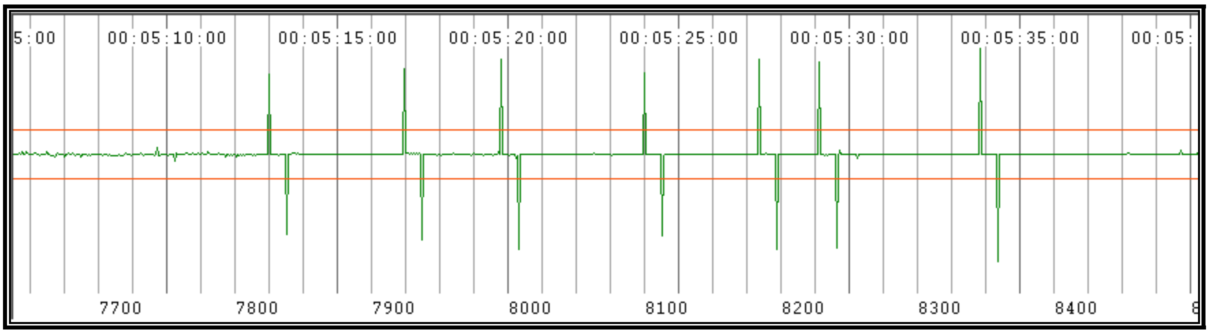

Figure 2.8 First derivative of convolved color histogram differences signal (green) and cut threshold (red)

A frame that fulfill this condition, a positive peak and a negative peak shifted *W* samples, as well as peak values exceeding a certain threshold is considered as a cut within the video sequence. Specifically, the middle frame between positive and negative peaks is taken as the beginning of a new shot, which coincides with the center of the rectangular shape signal found in the convolution signal. The selection of threshold level is not a critical process because cut signals appear well-distinguished from noise. The threshold value for cut detection used in the first derivative signal is obtained from the percentage of the maximum possible value of the color histogram difference. In this way, a cut is only detected when an abrupt variation overcome a certain percentage of color change in a video sequence.
The advantage of the presented cut detector method based on color histogram is that it takes into account the global variation of the image, being therefore less sensitive to camera or object movement.

2.4.2. Fade and dissolve boundary detection

The presented method in the previous section is able to effectively detect abrupt shot boundaries in video sequences. However, the method is not able to detect smooth shot boundaries as fades and dissolves. As mentioned previously, these sort of shot changes present a triangular-like shape in the convolved color histogram difference signal $HistDif_{conv}[i]$, due to the linear transition between shots that video editing effects as fades and dissolves present.
Unfortunately, not all the smooth shot boundaries present a perfect triangular shape, as this depends on the length and linearity of the effect and the presence of motion during the transition. Moreover, similar triangular shapes appear in the presence of camera motion and large moving object within the scene. The goal of the present section is to effectively identify both cases in order to discriminate smooth shot boundaries in front of other artifacts.
As first step, the convolved color histogram difference signal $HistDif_{conv}[i]$ is processed in order to locate local maxima. For this purpose, we apply Mathematical Morphology operators to the convolved signal to find the beginning and end of a possible smooth shot boundary. The Mathematical Morphology offers a group of nonlinear signal processing techniques based on minimum and maximum operations. The morphological opening and closing create a simpler function than the original, smoothing in a non linear way. The opening removes positive peaks that are thinner than the structuring element. Otherwise, the closing removes negative peaks that are thinner than the structuring element. Besides, while the opening remains below the original function, the closing remains above. In particular, we compute a morphological opening to $HistDif_{conv}[i]$ signal in order to detect the possible gradual transitions. The opening evaluation consists in applying a dilation of the erosion of the original signal, in our case $HistDif_{conv}[i]$, by a flat structuring element.
The dilation of a function by a flat structuring element can be defined as the dilation of each set $X_f(\lambda)$ by a set *B*. The erosion of a function by a flat structuring element can be defined as the erosion of each set $X_f(\lambda)$ by a set *B*. These definitions lead to the following formulas:

$$\varepsilon_B(\mathrm{f}(\mathrm{x})) = \inf_{y \in B} [\mathrm{f}(\mathrm{x-y})] \qquad (2.12)$$

$$\delta_B(\mathrm{f}(\mathrm{x})) = \sup_{y \in B} [\mathrm{f}(\mathrm{x-y})] \qquad (2.13)$$

As defined before, the morphological opening of $HistDif_{conv}[i]$ can be computed as follows

$$Opening = \delta_B[\varepsilon_B(\mathrm{f}(\mathrm{x}))] \qquad (2.14)$$

where f(x) is the convolved signal.

Figure 2.9 shows the result of computing the morphological opening to $HistDif_{conv}[i]$ where peaks that are thinner than the structuring element have been eliminated.

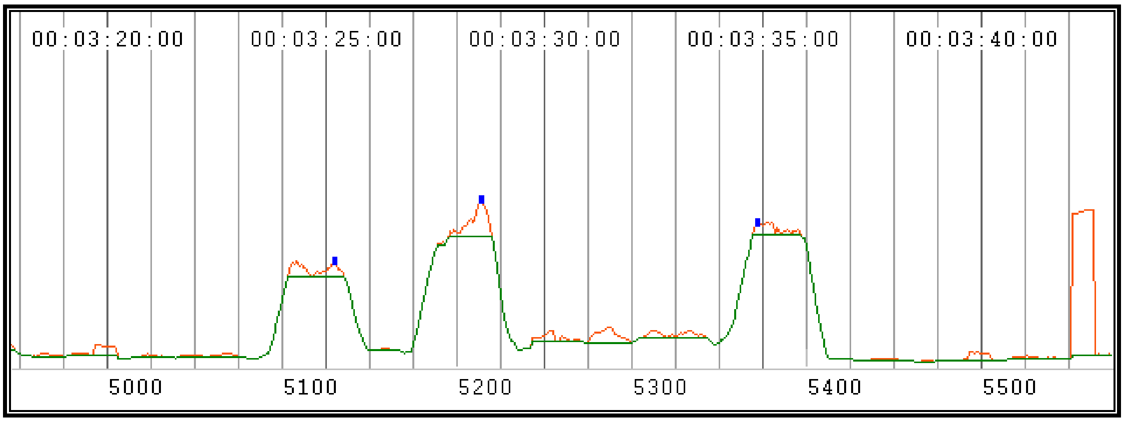

Figure 2.9 $HistDif_{conv}[i]$ signal (red), Opening of $HistDif_{conv}[i]$ signal (green)and local maxima points (blue)

As second step, we aim to detect the group of consecutive susceptible frames to belong to a fade or dissolve. Note that areas where cuts have been previously detected are excluded of this analysis. For that purpose, we look for a number of consecutive values within *Opening* signal that exceed the structuring element duration. With this processing, we detect the shape signals of a certain duration that are over a level, then we save the beginning and the end of the detected group of frames as the *start* and *end* frame of the video effect.
Once detected the *start* and *end* frame of the possible gradual shot boundary, we have to discriminate between real video effects, fades and dissolves, and false alarms caused by camera motion or large moving objects in the scene. The method for discriminating smooth shot boundaries is based on a simple fact that occurs when a gradual transition is done. First, we consider that the main characteristic of a fade or dissolve is the gradual and global color change of frames during the video effect. Then, we consider the frame sequence as a linear combination of the *start* and *end* frame that evolves during the transition. At the beginning of the transition the resulting image is a combination of the whole percentage of the *start* image and the null percentage of the *end*. Hypothetically, during the transition this percentages linearly increase/decrease to the end state where the resulting image is a combination of the null percentage of the *start* image and the whole percentage of the *end* one. Our aim consists in detecting the frame sequence which satisfies this behavior. So that, as the first step of the detection process, we compute the mean block images of all the frames belonging to the possible smooth transition. These images $I_m^{MeanBlock}$, shown in figure 2.10, are obtained as a result of dividing the frames in blocks of 8x8 pixels and computing for each block the mean of their RGB components of all pixels. As we consider these mean block images we are less sensitive to single pixel variation in the frame sequence. This fact allows the presence of little movement within the images where the gradual transition is done.

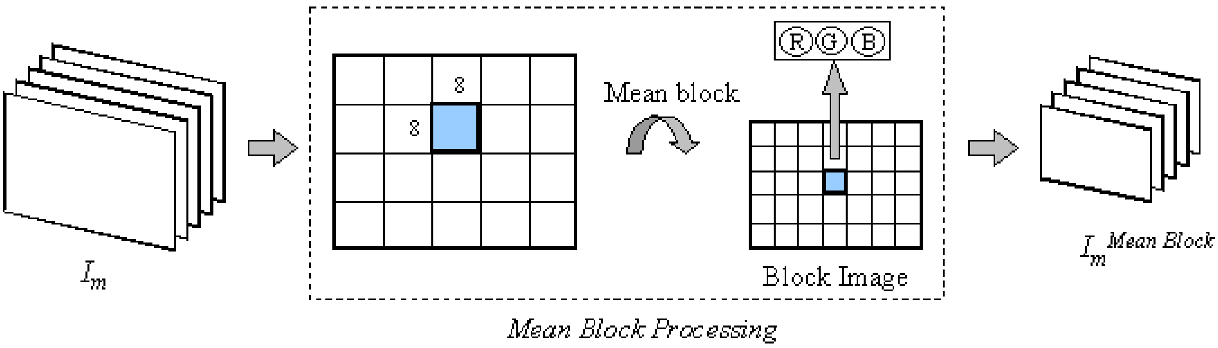

Figure 2.10 Mean Block Processing

In the next step, we suppose that in a gradual transition we will be able to identify a certain image pattern sequence. We define a pattern sequence as a group of synthetic images obtained as follows

$$I_n^{synth} = \alpha_n \cdot I_{start}^{Mean\,Block} + (1-\alpha_n)\cdot I_{end}^{Mean\,Block}; \qquad \alpha_n > \alpha_{n+1} \tag{2.15}$$

where $I_{start}^{Mean\,Block}$ and $I_{end}^{Mean\,Block}$ are the beginning and end frame of the possible gradual boundary after the mean block processing and α_n, with a 0 to 1 range, is the weight of the *n* image. Once defined the synthetic images of the pattern sequence, the proposed method search for each synthetic image computed I_n^{synth} the *m* position which satisfies that the image difference $d\left(I_m^{Mean\,Block}, I_n^{synth}\right)$, defined as

$$d(X,Y) = \sum_{i=1}^{B} EuclideanDistance_{RGB}(X_i, Y_i) \tag{2.16}$$

where X_i and Y_i are the mean blocks values of X and Y block images respectively and *B* the whole number of image blocks, is minimum.
After that evaluation, we obtain the image position of each I_n^{synth} within the gradual transition according to their similarity to $I_m^{Mean\,Block}$ images. With the analysis of these image positions we are able to conclude wether the studied frame sequence belongs to a smooth transition or a false alarm caused by camera/object movement.
Specifically, in our work, we have defined a simple pattern sequence of three synthetic images as a result of a lineal combination of the start and end frame, with $\alpha_0 = 0.75$, $\alpha_1 = 0.5$ and $\alpha_2 = 0.25$ respectively. Then we check if the synthetic images I_n^{synth} positions fulfill a distribution as follows

$$0 \le I_0^{synth} < \frac{M}{2} - p; \qquad \frac{M}{2} - p \le I_1^{synth} \le \frac{M}{2} + p; \qquad \frac{M}{2} - p < I_2^{synth} < M$$
$$0 \le p < \frac{M}{2} \tag{2.17}$$

where *M* is the number of frames of the frame sequence and *p* gives a margin of distribution of synthetic images during the transition. Figure 2.11 shows an example of this criteria. If this condition holds we consider that the analyzed sequence belongs to a gradual shot boundary.

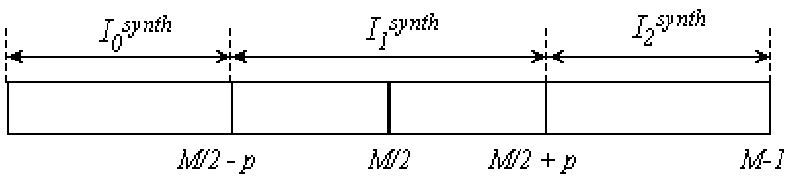

Figure 2.11 Example of I_n^{synth} distribution in the frame sequence

3. Experimental results

Experimental results of the presented method were obtained from the processing of TRECVID shot boundary test data. First of all we applied the cut boundary detection method. After the computation of the color histogram differences and convolution with a rectangular window of *W=13* samples, we reduced the effect of camera and object motion over the detection of abrupt shot boundaries. The obtained results are shown in Table 3.1, where we show the first run, with a certain threshold level for cut detection, is the best result. The following runs have been obtained by relaxing the threshold

level. Note that as we decrease this cut threshold we improve recall results, but, at the same time, we obtain worst precision results due to false alarms detected.
We have also applied the proposed method for smooth shot boundary detection for the same video data. In our experiment we have applied a morphological opening with a structuring element of 13 samples of width to detect the possible gradual transitions sequences. Then we have checked a simple pattern sequence of gradual change of three synthetic images within the a frame transition. As it is also shown in Table 3.1, due to the fact that we are only considering fades and dissolves as smooth transitions the obtained recall results for these transitions are worse than the ones obtained for cut detection. Currently, our method do not take into account transitions as wipes or other video effects. For this reason we have got a recall of approximately 50% for gradual transitions, but the precision results obtained are good enough, as the method distinguishes between real smooth transitions and false alarms caused by motion.

	Cut		Gradual		Global	
Run	Recall	Precision	Recall	Precision	Recall	Precision
1	0.940	0.855	0.479	0.811	**0.806**	**0.847**
2	0.965	0.740	0.479	0.854	0.823	0.757
3	0.951	0.826	0.470	0.811	0.811	0.824
4	0.973	0.706	0.468	0.857	0.825	0.727
5	0.958	0.779	0.459	0.817	0.812	0.785
6	0.977	0.647	0.442	0.858	0.821	0.673

Table 3.1 Results on shot boundary detection

With the proposed method we obtain good overall results for shot boundary detection, as it is shown in Table 3.1. The limitation of the method remains in the detection of gradual transition due to easy method used. Otherwise, the presence of camera motion or object motion during smooth shot boundaries also increases the difficulty of shot detection. This situation, although not being very critical, should be considered in further refinements of the method.
Another point to improve is the behavior of the method in the presence of motion and low variation of the color in the scene which leads to false alarms. It must be said that the low visual quality of videos definitely affected the results.

4. Audiovisual Indexing Tool

Apart from detecting shot boundary methods, an audiovisual indexing tool has been developed in order to catalogue video content according to MPEG-7 standard criteria. This software let us describe a certain audiovisual content by general features, content features and temporal description based on scenes.

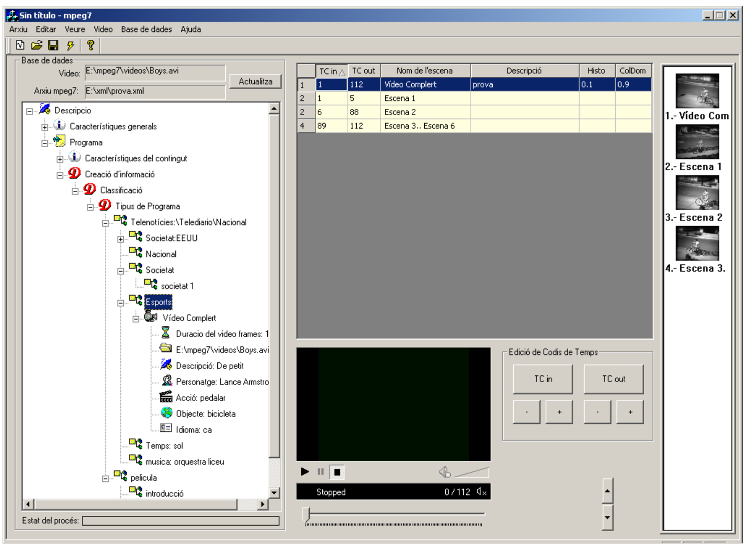

Figure 4.1 Audiovisual Indexing Tool

The main description element of an audiovisual content is the generic description scheme. The cataloguing tool let us create new description schemes for all kind of content, for example news, advertisements, sports and so on, that contain the structure of the whole general descriptors and content features. A Mpeg7Wizard tool is used to generate these descriptor structures helping us in the classification scheme definitions according to the kind of content.

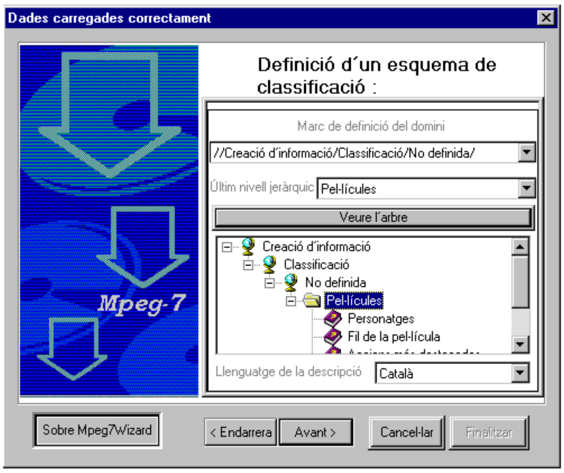

Figure 4.2 Mpeg7Wizard Tool

These description schemes can be created for every audiovisual content that is going to be indexed or it can be used with generic schemes already created for a specific content (for example a news description scheme). The definition of the classification scheme in our audiovisual cataloguing tool is totally flexible as we can add, eliminate or modify any description element predefined.

Once described the whole general characteristics of the video content, a temporal description can be done based on shots. We have integrated our shot boundary method into the cataloguing tool in order to obtain automatically the different video shots. A representative keyframe is assigned to every shot and is shown on the right of the indexing tool. At that point, the user is able to group or divide the detected shots according to their content. Besides, two image processing tools have been developed to manage the detected shots: an histogram matching and a dominant color matching. These tools help the user in the cataloguing process associating shots with similar histogram or dominant color.

Once all shots have been segmented we are able to independently describe each shot. This process is developed in a simple *drag&drop* operation over the description scheme, and the Mpeg7Wizard tool help us in the description of the basic elements of a shot, as the main character, the action, the object and a free description. Then, we are able to modify any of these description elements related to a shot over the description structure.

All description information, general description elements and detailed shot descriptions of a video content, is stored in xml files and can be edited later to add, eliminate or modify any of the description elements associated to the content.

The huge increase of the digital audiovisual contents leads to the need of cataloguing and indexing tools. The result information of indexing process of multimedia data are going to be stored in data bases. The main purpose of the cataloguing process is to access to audiovisual content as fast as possible, and allow to query for a specific video content based on content descriptors.

5. Conclusions

In this paper we have shown our shot boundary detection method for the TREC 2003 Video Track in our first participation using the color histogram difference as the main feature of analysis. The detection of cut boundaries is simple and robust in front of camera and moving objects. With respect to the smooth shot boundaries as fades and dissolves we have presented a complementary method that takes into account the linear color variation of images along the time and removes the presence of moving objects that distort this measure using mean blocks computation.

In the future, we count on using more accurate smooth shot boundary detection and motion analysis methods to overcome missed shots and discard false alarms. From the application point of view, we will integrate the method in a more general framework of a semi-automatic audiovisual indexing tool for broadband and broadcast personalized content delivery.

6. References

[1] B. Yeo and B. Liu, “Rapid Scene Analysis on Compressed Video”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, No.6,pp.533-544 December 1995.
[2] S.H. Han, K.J. Yoon, I.S. Kweon, “A New Technique for Shot Detection and Key Frames Selection in Histogram Space”, Robotics & Computer Vision Lab., Republic of Korea
[3] A Hanjalic, “Shot-Boundary Detection: Unraveled and Resolved?”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No.2,pp.90-104 February 2002.
[4] C.W. Ngo, T.C. Pong and R.T. Chin, “Detection of Gradual Transitions through Temporal Slice Analysis”, Department of Computer Science, The Hong Kong University of Sciene & Technology, Hong Kong.
[5] J. Oh, K. A. Hua, N. Liang, “A Content-based Scene Change Detection and Classification Technique using Background Tracking”, School of Computer Science, University of Central Florida, Orlando.
[6] S.H. Han, I.S. Kweon, “Shot detection combining Bayesian and structural information”, Robotics & Computer Vision Lab., Department of Electrical Engineering & Computer Science, Republic of Korea.