Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Sébastien Lefèvre · Nicole Vincent

Efficient and Robust Shot Change Detection

the date of receipt and acceptance should be inserted later

Abstract In this article, we deal with the problem of shot change detection which is of primary importance when trying to segment and abstract video sequences. Contrary to recent experiments, our aim is to elaborate a robust but very efficient (real-time even with uncompressed data) method to deal with the remaining problems related to shot change detection: illumination changes, context and data independency, and parameter settings. To do so, we have considered some adaptive threshold and derivative measures in a hue-saturation colour space. We illustrate our robust and efficient method by some experiments on news and football broadcast video sequences.

Keywords Shot Change · Hue Saturation Luminance · Illumination Invariance · Context Independency · Real-Time · Parameter Robustness

S. Lefèvre (corresponding author)
LSIIT – Université Louis Pasteur (Strasbourg I)
Parc d'Innovation, Bd Brant, BP 10413, 67412 Illkirch Cedex, France
Tel.: +33-390244570, Fax: +33-390244455, E-mail: lefevre@lsiit.u-strasbg.fr

N. Vincent
CRIP5 – Université René Descartes (Paris V)
45 rue des Saints Pères, 75270 Paris Cedex 06, France
E-mail: nicole.vincent@math-info.univ-paris5.fr

1 Introduction

Shot change detection, also called temporal segmentation, is a crucial task in multimedia applications such as multimedia database management systems, automatic abstracting of TV or movie sequences, or real-time object tracking. The great impact of this process on a global system explains that numerous temporal segmentation methods have been proposed for two decades and original contributions are still appearing.

We believe the process of shot change detection should be both efficient and robust to be widely useful. While the efficiency ensures the process to be performed in real-time (even with uncompressed data or unspecialized hardware), the robustness helps to deal with various artefacts (such as illumination changes) and to process unknown datasets without the need for very accurate parameter settings or dedicated supervised learning step. These two properties are the grounds of the proposed approach, which is based on relative interframe measures and thresholds on low-resolution data in an illumination-invariant colour subspace. Compared to the state-of-the-art, our contribution consists in a method which is both efficient and robust to extract shot changes.

Our paper is organized as follows. First we will recall the problem of shot change detection and underline the importance of efficiency and robustness. Next we will describe the different steps or our algorithm. Results on several datasets will then be given and properties of our method will be discussed.

2 The problem of shot change detection

In this section, we will first give some definitions necessary for the general reader. We will then briefly present related works by indicating early approaches as well as current trends. Finally we will identify the problems that remain open in the field and describe the main objectives of the proposed method.

2.1 Shot change definition

A shot is defined as a set of successive images obtained from a continuous acquisition of a single camera. It is often considered as the base unit in video analysis systems. Each shot is separated from the previous and the next ones by transitions. There exist two kinds of transitions: abrupt and progressive transitions. In an abrupt transition (also called a cut), the last frame of the first shot is directly followed by the first frame of the second shot. No effect has been inserted between the two shots, as shown in the top line of figure 1.

In the case where two shots are connected using a particular effect, the term progressive transition is used. Several kinds of transitions can be introduced in video sequences, the best known being fade (or dissolve) and wipe, as illustrated in figure 1. During a fade, the level of each pixel in the intermediary frames (frames which belong to the progressive transition) is computed using pixel values from the images of the two shots, with an increasing weight given to values from the second shot. During a wipe, each pixel in the intermediary frames has a level which is equal to the level of a pixel with the same spatial coordinates, either in the first shot or in the second one, with an increasing proportion of pixels from the second shot. Production processes are able to build some more complex effects but the previous principles are always taken into consideration.

We will now see briefly the historical approaches to solve the problem of shot change detection and present some more recent approaches in order to underline our contribution.

2.2 Related work: from early days to current trends

The problem of shot change detection has been studied very deeply and numerous techniques have been proposed for two decades. The reader can find in [13,15] exhaustive reviews of the state-of-the-art in this field. The usual way to solve this problem is based on two successive steps: a dissimilarity measure is computed

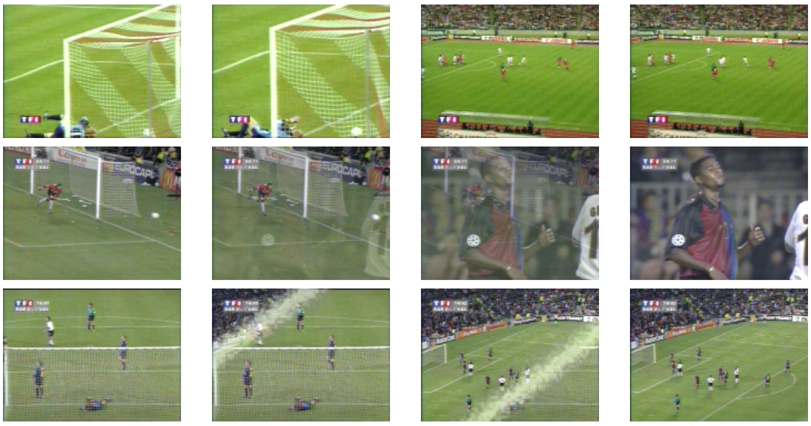

Fig. 1 Examples of main transition types : cut (top), fade (middle), and wipe (bottom).

between successive frames of a video sequence, then this value is compared to a threshold in order to determine whether a shot change is present or not. Relying on this principle, a shot change is detected if the following condition holds:

$$d(I_t, I_{t-1}) > T \tag{1}$$

where I_t denotes the video frame obtained at time t, d is a dissimilarity measure, and T a threshold. Setting the T value is a difficult task. As we will see, some recent approaches have tried to avoid using this early scheme since it is based on a fixed threshold.

The main early approaches to solve this problem were comparing directly pixel values between successive frames which yields to high sensitivity to noise and motion. Then frames were characterized by various features (histograms, local statistics, motion, etc) and interframe comparison was relying on these features. In the compressed domain, the methods were considering most of the time video compression standards [6], such as MPEG and MJPEG (or Motion-JPEG). While these methods ensure a lower computation time since they work directly in the compressed domain, they are very dependent on the compression scheme considered and thus cannot deal with any video data.

Since these early works, the problem of shot change detection has been widely studied by the very fact of the primary importance of this processing in video analysis and indexing. Some attempts have been made to review existing solutions for shot change detection under a common framework, such as the formal study from Yuan et al [21]. They discuss method optimal criteria and propose to use two support vector machine (SVM) classifiers to detect respectively cuts and gradual transitions. In [4], Cao and Cai use a single SVM classifier to assign each frame one of the three following labels: abrupt transition, progressive transition and no transition. Classification-based approaches tend to gather most of current efforts in the field but are limited to their very high dependency on the learning step.

Alternatively, many methods are dedicated to a specific kind of transition. Grana and Cucchiara [10] introduce an iterative algorithm for linear transition detections which tries to determine optimal transition extremities and length using only two parameters. Li and Lee [14] focus on wipe detection and propose a generic wipe model to be matched with observed sequences in a Bayesian framework. The influence of parameter setting on the results seems to be reasonably low. Qian et al [18] deal with flashlights and fades by accumulating histogram differences. It is also possible to build a set of dedicated detectors, such as the system proposed by Bezerra and Leite [1] who project the video signal along horizontal and vertical dimensions and process the observed sequences with several specialized detectors (cut, wipe, fade) relying on various parameters. These heuristic methods often require several parameters to be hardly set.

In order to increase shot change detection efficiency, several strategies may be considered. Most often, efficiency is reached using compressed data instead of uncompressed data. By limiting the amount of data to be analysed, real time processing can be achieved quite easily [12]. This is particularly true for motion-related techniques which can hardly be performed in real-time [7] despite their interesting accuracy. An alternative strategy to ensure efficiency is to rely on a hardware implementation, such as Boussaid et al [3] who coded an early histogram-based shot change detector on a FPGA-platform. Here we prefer to reach real time by relying on an efficient algorithm rather than on compressed data or dedicated hardware.

Finally, several authors have recently proposed very original approaches relying on principles never used in the shot detection community. Among these principles, we can mention attentive vision [2], information theory [5, 11], fuzzy logic [8], or Markov chain Monte Carlo [22]. These different works offer new theoretical frameworks to deal with shot change detection. However, most of them do not tackle our concerns (i.e. efficiency and robustness).

So far we have described the most common approaches for shot change detection, since the early days to current trends. Before we put forward our contribution which consists in a method both robust and efficient, it might be useful to present the current unresolved problems and the main limits of recent approaches.

2.3 Robustness and efficiency

The problem of shot change detection has been tackled for many years and hundreds of solutions have been proposed. Despite the quality shown by the most recent ones (see for instance the results of the TRECVID 2005 benchmark [17]), there remain several remaining and challenging problems which make the problem of shot change detection an open research problem. We think we can summarize these problems with the statement “robustness and efficiency” which is the leading principle of our contribution.

Certainly the main limitation is related to the process of parameter setting. Most of the methods can reach high quality rates with a set of parameter whose values are dedicated to the dataset on which the method is applied. The sensitivity of parameters is very strong and an expert is necessarily involved to determine the adequate values for the different parameters, depending on the considered dataset. A current way to solve this problem is to replace the thresholds by some supervised classification algorithms. However, despite the benefit brought by the learning procedure, the supervised shot change detector is then characterized by a high sensivity to this learning step which has to be representative enough of the corpus to be considered in order to be relevant. Indeed, if one gives to the detector some video sequences for which there has not been any learning, the results may be very unpredictable. The actual trend consisting in using classifications instead of thresholds may limit this problem, but it does not resolve it completely. We believe a relevant solution would be to elaborate a method with low sensitivity to parameter settings.

Lack of robustness can also be observed when dealing with heterogeneous video sequences. Indeed, the overall quality of the process depends largely on the kind of video corpus which is used to perform evaluation. For instance, processing news video to detect shot changes often leads to better results than more complex data like football video broadcasts. Therefore ensuring a constant good performance of the shot change detection step, whatever the video corpus considered, is required to provide a reliable input to higher-level video indexing and analysis processes. We have decided here not to use the TRECVID dataset which is composed quite exclusively of news video sequences, but to focus on some complex data from the viewpoint of shot change detection in particular football video broadcasts. On these sequences, commonly used histogram-based features are completely irrelevant because of the constant background colour. We have added to this specific dataset a more common corpus composed of news video sequences from the French National Audiovisual Institute in order to take into account various directing styles (with various transition lengths and speeds) in a global and heterogeneous corpus.

Moreover, the diversity of data content may also be noticed at the transition level. As pointed out in TRECVID experiments, transition types are quite unlimited: it is always possible to create complex editing effects or to combine several effects to build new kinds of complex transitions. A shot change detector relying on a predefined formulation of the possible shot changes will then undoubtedly miss these transitions. Besides the transition variability, which has to be taken into account, different artefacts appear in video sequences. These artefacts may be due to camera or important object motions, illumination changes and lighting effects, low signal quality or high noise level following a compression scheme. A majority of the existing approaches do not ensure robustness to all these artefacts, otherwise they require a specific processing to deal explicitely with each of them.

To ensure a satisfactory robustness to the different possibilities of variability, the commonly adopted solution consists in adding more reliable data features or detection algorithms. However, using shape or texture descriptors like in some TRECVID approaches prevent the system from performing in real-time if uncompressed data are used. That is why many approaches described as efficient algorithms consider (only) the compressed data which already contain some motion information. Limiting the video sequences which can be processed to compressed data is also a way to reduce robustness as the uncompressed data cannot be processed anylonger and only considered compression scheme can be tackled.

Thus the main difficulty of the shot change detection problem may be to find a solution which ensures robustness and reliability without discarding efficiency. The method we are proposing here and describing in the next section has been elaborated with the common objectives of robustness and efficiency discussed in this subsection. These two properties together differentiate our contribution from recent trends such as the methods from TRECVID, which are most of the time characterized by high complexity (except when dealing with compressed data) in order to ensure a significant robustness. Contrary to those approaches, our method may be seen as a simple but reliable technique, as we will see in the next section.

3 Proposed approach

In the previous section we have briefly presented early and recent trends for shot change detection. We have pointed out the interest of a detection method which could ensure both robustness and efficiency. In this section we will present our method which was built in this aim. The proposed algorithm relies on data preprocessing and colour space conversion steps, which will be presented first. Then we will detail the dissimilarity measure to be used and the way our algorithm performs the shot change detection.

3.1 Data preprocessing

In order to ensure a relatively low computation time and to consider uncompressed or compressed data, we propose to introduce a data preprocessing step, the goal of which is to decrease the spatial resolution of the video frames to be analysed. Thus, the number of pixels to be processed will be lower, and so will be the number of required computations. The shot change detection can be performed on uncompressed or compressed video sequences. The only difference is the way to obtain low resolution images, and more precisely images with a size 64 times lower than the original ones. The resolution reached is enough to visually detect the shot changes.

In the case of uncompressed video sequences, each pixel is characterized by a value for each colour component, usually in the RGB (Red Green and Blue) space. We use here the notation $I(x, y, C)$ to design the value of a pixel with spatial coordinates (x, y) and colour component C. We propose to build from each original image (containing $X \times Y$ pixels) a new image composed of $X' \times Y'$ pixels. We set $X' = \frac{X}{8}$ and $Y' = \frac{Y}{8}$ for compatibility of the rest of the method with DCT-based compressed data such as MPEG or M-JPEG. For each block 8×8 (64 pixels) and for each colour component, the block average is computed. The obtained values help to characterize the new pixels of the low resolution image. The computation method can be formulated as :

$$I'_t(x', y', c) = \frac{1}{64} \sum_{x=8(x'-1)+1}^{8x'} \sum_{y=8(y'-1)+1}^{8y'} I_t(x, y, c) \qquad (2)$$

where I'_t represents a low resolution image (with size $X' \times Y'$) obtained from frame I_t (with size $X \times Y$).

In the case of compressed video sequences, it is possible to build low resolution images by avoiding a complete data decoding. Indeed, in video sequences compressed with a DCT scheme (such as M-JPEG, MPEG-1 or MPEG-2 standards), the coefficients resulting from the DCT application on each image block can be used. Following the work from Yeo and Liu [20], we consider here only frames which are coded without any motion information, i.e. all frames from M-JPEG video sequences and I frames from MPEG video sequences. Each 8×8 (64 pixels) block is characterized by a DC coefficient which represents the low frequency information, and by 63 AC coefficients. The DC coefficients of the different image blocks can be used to generate the low resolution image I' in the following way:

$$I'_t(x', y', c) = \frac{1}{8} I_t^{b(x',y')}(0, 0, c) \tag{3}$$

where $b(x', y')$ represents a block of spatial coordinates (x', y') (defined at the block scale) and $I_t^b(0, 0, c)$ the first coefficient (position $(0, 0)$) of the b block, i.e. the DC coefficient, considering the colour component c of the compressed image I_t.

Once the data have been spatially reduced, they go through a colour space conversion.

3.2 Colour space conversion

The colour coding in digital images can be performed using different representation spaces usually called colour spaces. In our method, we have selected a colour space widely known to be related to the human vision system, namely the HSL space which is represented by three components: hue, saturation, and luminance (or value). Whereas the saturation and luminance are coded in a classical way (as scalars), the hue is an angular value. The hue represents the colour perceived (red, yellow, green, etc), the saturation measures the purity of the colour (e.g. for a pink hue, the pink colour is characterized by a lower saturation than the red colour, whereas black, white, and grey colours are characterized by a null saturation), and the luminance represents the grey level, from dark (minimum) to white (maximum). Obviously we could have chosen another colour space but we do not think the choice of the colour space is a very critical point, as long as the colour space used is related to the human vision system and is based on a hue - saturation - luminance triplet.

Figure 2 illustrates these different components. We can observe that saturation and luminance are both represented on a linear scale, saturation on the horizontal direction and luminance on the vertical one, whereas the hue is modelled with an angle. So, it has to be considered when used in various computations [16]. The HSL space brings complementary information through its three components. It is interesting to build solutions which are robust to illumination changes. Indeed, these artefacts affect mainly the luminance component. If only chromatic components (hue and saturation) are taken into account, it is possible to decrease the sensitivity to illumination changes. However hue needs to be analysed very carefully. Indeed, its reliability depends on the saturation level and hue is significant only if saturation is high. Analysis methods which are based on pixel hue values have to check whether those pixels are not achromatic. Another constrainst which is related to the hue comes from its mathematical definition (angular measure) that requires the use of some specific statistical measures [16].

From the main concepts we have recalled here, several authors have proposed their own HSL model. We use here the definition from [19] also called hexagonal cone model for computational reasons as it does not require any trigonometric operation or other floating-point computation. Before defining precisely the HSL space, we introduce the two following notations:

$$I(x,y,\text{min}_{\text{RGB}}) = \min(I(x,y,R), I(x,y,G), I(x,y,B)) \tag{4}$$

$$I(x,y,\text{max}_{\text{RGB}}) = \max(I(x,y,R), I(x,y,G), I(x,y,B)) \tag{5}$$

The coordinates in the HSL domain are then defined by:

$$I(x,y,L) = I(x,y,\text{max}_{\text{RGB}}) \tag{6}$$

This definition is particularly sensitive to noise introduced in the luminance component. Nevertheless, as we will see later in this paper, the method we are proposing relies only on hue and saturation components and therefore is not affected by this limitation. If $I(x,y,L)$ is null, saturation and hue are not defined. Otherwise, we have:

$$I(x,y,L) \neq 0 \quad \text{and} \quad I(x,y,S) = \frac{I(x,y,L) - I(x,y,\text{min}_{\text{RGB}})}{I(x,y,L)} \tag{7}$$

If $I(x,y,S)$ is null, the hue is undefined. Otherwise, it is computed in radians as:

$$I(x,y,H) = \begin{cases} \frac{\pi}{3}(5 + I(x,y,B')) & \text{if} \quad I(x,y,R) \geq I(x,y,B) \geq I(x,y,G) \\ \frac{\pi}{3}(1 - I(x,y,G')) & \text{if} \quad I(x,y,R) \geq I(x,y,G) \geq I(x,y,B) \\ \frac{\pi}{3}(1 + I(x,y,R')) & \text{if} \quad I(x,y,G) \geq I(x,y,R) \geq I(x,y,B) \\ \frac{\pi}{3}(3 - I(x,y,B')) & \text{if} \quad I(x,y,G) \geq I(x,y,B) \geq I(x,y,R) \\ \frac{\pi}{3}(3 + I(x,y,G')) & \text{if} \quad I(x,y,B) \geq I(x,y,G) \geq I(x,y,R) \\ \frac{\pi}{3}(5 - I(x,y,R')) & \text{if} \quad I(x,y,B) \geq I(x,y,R) \geq I(x,y,G) \end{cases} \tag{8}$$

with $I(x,y,R')$, $I(x,y,G')$, $I(x,y,B')$ computed with the following equation:

$$I(x,y,c') = \frac{I(x,y,\text{max}_{\text{RGB}}) - I(x,y,c)}{I(x,y,\text{max}_{\text{RGB}}) - I(x,y,\text{min}_{\text{RGB}})} \qquad \forall c \in \{R,G,B\} \tag{9}$$

which can also be expressed as:

$$I(x,y,c') = \frac{I(x,y,L) - I(x,y,c)}{I(x,y,S) \times I(x,y,L)} \qquad \forall c \in \{R,G,B\} \tag{10}$$

In order to ensure the robustness against illumination changes but also to reduce computation time, we have decided to limit pixel representation to 2-

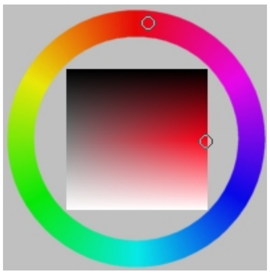

Fig. 2 Visual representation of the HSL colour space: the disc models the hue, whereas the square models the saturation (horizontal direction) and the luminance (vertical direction).

dimensional space composed of hue and saturation only. As we will see in the results section, for all the outdoor scenes which are frequently observed in video sequences, we can notice a significant improvement over the use of the classical RGB space. This result is complementary to those obtained by Gargi et al. in [9] for colour histogram-based shot detection.

Once the data from each frame have been normalised and reduced, a dissimilarity measure will be applied to the successive frames.

3.3 Dissimilarity measure

We propose to measure the dissimilarity between frames in the HS (Hue Saturation) subspace, using the measure d defined as:

$$d(I_{t_1}, I_{t_2}) = \sum_{x=1}^{X} \sum_{y=1}^{Y} I_{t_1}(x,y) \ominus I_{t_2}(x,y) \tag{11}$$

with $\ominus$ an algebraic operator involved in the comparison between two pixels considering the HS subspace. As Hue and Saturation are respectively angular and scalar measures, a specific definition should be given:

$$\begin{aligned} I_{t_1}(x,y) \ominus I_{t_2}(x,y) = \alpha_{H,S}(I_{t_1}(x,y,H) - I_{t_2}(x,y,H)) \pmod{2\pi} \\ + (1-\alpha_{H,S}) \left| I_{t_1}(x,y,S) - I_{t_2}(x,y,S) \right| \end{aligned} \tag{12}$$

where $\alpha_{H,S}$ is a coefficient which helps to give more or less influence to the H and S components. Indeed, in case of achromatic pixels, it is important not to give too much importance to the Hue component since it is not reliable. This coefficient is defined as:

$$\alpha_{H,S} = k \cdot \chi_{S>T_S} \text{ with } \chi_P = \begin{cases} 1 \text{ if } P \text{ holds} \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

with k a constant and T_S a threshold compared to saturation values. This coefficient, computed for each pixel, is thus independent of the type of image considered.

The distance measure d, even if it is relatively simple, enables us correctly estimate the difference between two images in a way which is invariant to illumination. Using a more complex measure could provide a higher information quality but it would also mean higher computational cost. However, the direct use of this dissimilarity measure between two successive frames would require the comparison with a threshold T (see equation (1)). The threshold to be used has often to be set empirically, and depends on the video domain considered (sport, news, etc) or on the kind of shots present in the video sequence. Thus a far shot, where moving objects are small, will be characterized by relatively low d values whereas a close shot, where moving objects represent an important part of the image, will be associated with higher d values. The threshold T should be set accordingly to avoid false positives and negatives. As some video sequences (e.g. sport broadcast video sequences) contain both far and close shots, it is necessary to introduce a more generic method which can adjust to these different kinds of shots. Therefore we propose to consider both an adaptive threshold and a differential measure. The adaptive threshold noted T_d is updated through time to limit the number of false positives and false negatives. For each new frame of the video sequence, we have:

$$T_d(t) = \alpha_{T_d} T_d(t-1) + (1 - \alpha_{T_d}) d(I_t, I_{t-1}) \tag{14}$$

where $T_d(t)$ represents the threshold value T_d at time t. Thus it can adapt automatically, with a given inertia (represented by the coefficient α_{T_d}), to the content of the video sequence analysed, whose value is modified depending on the values $d(I_t, I_{t-1})$. The parameter α_{S_d} has a direct influence on the precision and recall measures, whereas the initial choice of $T_d(0)$ has a very low influence on the results obtained.

The direct use of a dissimilarity measure d between two successive frames is very sensitive to noise and motion present in the video sequence which is analysed. Introducing an adaptive threshold helps to decrease this sensitivity in a certain way, but obviously not completely. Besides we propose to consider a relative measure instead of an absolute one. This relative measure, noted d', helps to increase the robustness against noise and important motions present in the sequence. It is defined by:

$$d'(I_t) = |d(I_t, I_{t-1}) - d(I_{t-1}, I_{t-2})| \tag{15}$$

Thus we have an estimate of the second order derivative of the video signal. Contrary to the measure d, the measure d' is defined in a relative way and its scale depends less on the kind of shot or video sequence analysed. In order to detect a shot change, this measure can then be compared to a threshold $T_{d'}$ set empirically at the beginning of the video sequence. The value of $T_{d'}$ could depend on the kind of video sequence or shot analysed. However experimentally we have observed no need to adapt the threshold value to the different data considered (sport, news, etc), thereby we are ensuring a certain robustness to our approach.

So far we have presented here the dissimilarity measure which is used and we have justified the interest to use the variation of a dissimilarity measure computed between two successive frames instead of the measure itself. We will now present the shot change detection algorithm.

As previously mentioned, a shot change can be abrupt or progressive. Considering a progressive transition as an abrupt one whose effects have been spread on several images, we propose a method to detect abrupt and progressive transitions in a relatively similar way.

In order to detect an abrupt change, we compare directly the value d' with a threshold $T_{d'}$. Indeed, if the value of d' is high, i.e. if the absolute difference between $d(I_t, I_{t-1})$ and $d(I_{t-1}, I_{t-2})$ is significant, then the evolution of the video content from images I_{t-2} and I_{t-1} is not coherent with the evolution observed between frames I_{t-1} and I_t. Therefore an abrupt change is present at time $t-1$.

If no abrupt change has been detected, a progressive transition may yet occur. The value d' cannot be used directly in the case of a progressive transition as it represents the evolution of the dissimilarity measure d only at a given time. The values d' obtained for all the frames composing a progressive transition must be added up in order to obtain a measure of the same scale as the threshold value considered in the case of abrupt transitions.

The detection of progressive transitions is thereby performed in two successive steps. The first step consists in the detection of frames which could be the borders of a progressive transition. In order to detect these frames, we analyse the evolution of the dissimilarity measure d and we compare at each time t the value $d(I_t, I_{t-1})$ with the adaptive threshold $T_d(t)$ defined by equation (14). Using this adaptive threshold enables us to deal with any kind of situation (close or far shot, important or low motion, etc). If the condition defined by $d(I_{t_1}, I_{t_1-1}) > T_d(t_1)$ holds, then a transition may be present in the video sequence from time t_1. The transition ending frame at time t_2 will correspond to the first frame which verifies the condition $d(I_{t_2}, I_{t_2-1}) < T_d(t_2)$ with $t_2 > t_1$ obviously. Once the borders t_1 and t_2 of a possible transition have been determined, it is necessary to analyse the frames t of this temporal interval. To do so, we compute a sum of all d' values on the interval of frames $[t_1, t_2]$ noted $d'_{\text{sum}}(t_1, t_2)$:

$$d'_{\text{sum}}(t_1, t_2) = \sum_{t=t_1}^{t_2} d'(I_t) \tag{16}$$

The comparison between $d'_{\text{sum}}(t_1, t_2)$ and the threshold $T_{d'}$ enables us to validate or not the presence of a progressive shot change, located between frames I_{t_1} and I_{t_2}. This approach by derivation / integration helps to greatly reduce the sensitivity to initial values.

In the following section, we will present the results obtained. They illustrate the robustness and efficiency properties of our method, as claimed in the preliminary sections.

4 Results and discussion

The method presented in the previous section was elaborated to ensure both robustness and efficiency properties. We will hereafter present the corpus considered, give the results obtained by our method and compare them with others, discuss its robustness in particular to parameter settings, and compare the influence of RGB and HSL colour spaces in detection quality.

4.1 Description of the corpus

As specified in the subsection 2.3, we have decided not to use the classical TRECVID corpus since this dataset is composed almost only of news broadcast video sequences. In order to show the robustness of our method to different types of video sequences, we chose to build a specific but diversified corpus.

This corpus is partly composed of football broadcast video sequences as we consider these data as complex in the context of shot change detection. Indeed, these sequences are characterized with both close and far shots, a high quantity and variability of progressive effects, an important variation of camera and object motions, frequent illumination changes due to the lighting system, and finally a certain homogeneity of the video frames even if they belong to different shots. With such data the shot change detection approaches are likely to fail.

In order to make our results comparable to the existing conclusions of the literature, we have considered in an other part of our corpus a more usual news broadcast video dataset. These video sequences were taken from the dataset proposed by the French National Audiovisual Institute to evaluate video analysis algorithms.

The global corpus is composed of 12000 frames equally taken from TV news and football broadcast video sequences and it contains about one hundred shot changes. Despite its content diversity, the football broadcast part of the corpus shares some relatively similar properties with the TRECVID dataset: it contains about 0.75 transitions per 100 frames, and the repartition between cuts and progressive transitions is about one third for the former and two thirds for the latter. Our corpus is much smaller than the TRECVID one (about 1 % the size of the TRECVID dataset). However, we do not think that a more usual corpus (such as the TRECVID one), larger in size but lower in diversity, would yield conclusions different from the ones that will be presented in the rest of this section.

4.2 Evaluation of the quality and efficiency

As previously mentioned, we address in this paper the problem of shot change detection for which we propose an efficient and robust solution. So in order to evaluate the relevance of our contribution in this context, we have decided to measure the quality and efficiency of the proposed approach.

We have used the well-known recall and precision measures (or rates) to quantify the quality of our method and to objectively compare it with others. These measures are respectively defined as:

$$Q_{\text{recall}} = \frac{N_d}{N_d+N_m} \tag{17}$$

$$Q_{\text{precision}} = \frac{N_d}{N_d+N_f} \tag{18}$$

where N_d, N_m, and N_f represent the number of correct detections, missed detections (or false negative), and false detections (or false positive) respectively.

Using these two criteria, we can compute some more complex quality measures such as the average precision measure or the F1 measure. We have chosen the latter to merge the correlated recall and precision values into a single measure. The F1 measure is defined as :

$$Q_{\text{F1}} = \frac{2 \times N_d}{2 \times N_d + N_m + N_f} \tag{19}$$

or equivalently as :

$$Q_{\text{F1}} = \frac{2 \times Q_{\text{recall}} \times Q_{\text{precision}}}{Q_{\text{recall}} + Q_{\text{precision}}} \tag{20}$$

For the sake of simplicity, we do not present here the recall-precision curves but instead we examine the highest quality measure which can be reached by a given method, assuming an optimal set of parameters is chosen, and considering the F1 quality criterion. To choose this optimal parameter set, we have evaluated the results obtained by the tested methods using varying sets of parameters. Following the remarks made in section 2.3, we have not involved any learning process to set the actual parameters. As we will see further, our method is quite robust to parameter settings and its evaluation does not require two distinct datasets (training and testing) but only one.

In table 1, we have compared our results with those given by two classical detectors [13]: the first is based on a pixel wise difference and illustrates the incompatibility of this kind of approach with scenes acquired with a moving camera, whereas the second is based on histogram difference and illustrates the difficulty to process scenes with a globally uniform background. For each triplet (corpus, transition type, method) we have computed the best F1 measure which could be reached with varying parameters. The goal of this table is to show the maximal quality that can be ensured by the different methods on the two corpora, assuming optimal parameters are known. It can notice that processing news data seems easier than processing football data. Our method however presents slightly similar results for these two datasets contrary to classical approaches, and it consequently ensures higher quality. But as we will see in the next subsection, the main aspect of our method may not be that it yields the best results but that it yields satisfactory results most of the time.

Table 1 Best quality (F1 measure) obtained by our method and classical pixel-based and histogram-based approaches.

Corpus	Transition	Proposed method	Pixel method	Histogram method
News	Cuts	1.00	0.99	0.98
News	Gradual	0.90	0.86	0.83
News	Global	0.98	0.96	0.96
Football	Cuts	0.96	0.92	0.86
Football	Gradual	0.84	0.75	0.76
Football	Global	0.88	0.83	0.77

The main limitation of the method which has been proposed here is its sensitivity to the motion present in the sequence. This observable motion may result from an abrupt acceleration of the camera or by the motion of an object which is represented by the major part of the image in a close shot. Therefore a compromise must be found between false detections due to motion and missed detections of effects like fades. We think we have reached reasonable limits under the real-time constraint by introducing a second order difference and an adaptive threshold. Moreover, it has been noticed that F1 measure is higher than 95 % on motion-based dataset if abrupt transitions only are considered.

After the spatial reduction step (ratio equal to 8×8), the images to be processed in the football dataset contain only 20×15 pixels instead of 160×120 pixels, as shown in figure 3. For this reduced image size, the required computation time is equal to 0.4 milliseconds per frame with a Java-based implementation on a PC Pentium IV 3 GHz 512 MB. In other words, our system is able to process 2500 frames per second and fully respect the real-time constraint.

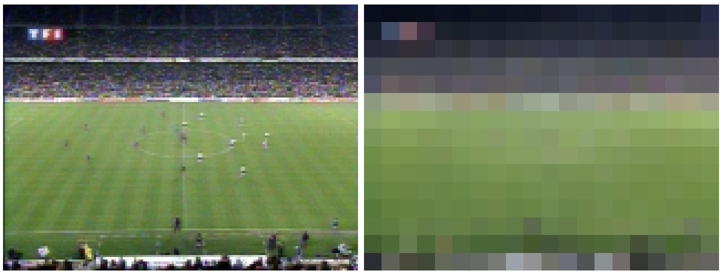

Fig. 3 Reduction of spatial resolution with a 1:64 ratio: original image (left) and reduced image (right).

4.3 Robustness to parameter settings

The results presented here have been obtained after an automatic evaluation of all possible parameter settings in order to determine the optimal one. There are neither manual or expert settings, nor learning procedure. Nevertheless, as an example, we give in table 2, two samples of parameter settings that enable our method to obtain the best quality (measured with the F1 criterion). In the first case, the optimal set is searched among 1296 possibilities, whereas in the second case the search space is enlarged to 14641 possibilities, thus leading to a very low quality improvement of 0.9%. The weights of hue versus saturation are respectively 0.4 and 0.5, the minimum saturation level to consider the associate hue relevant is equal to 0.2, the inertia of the adaptive threshold is either 0.4 or 0.7, and the invariant threshold is set to 8 or 10. Let us note however that these settings are just one possibility among many others, as our method can ensure a certain robustness to parameter settings. This will be discussed in the rest of this section.

Table 2 Two samples of parameters used in the detection process.

Parameter	Description	best value	best value
		among 1296	among 14641
k	Constant related to $\alpha_{H,S}$	0.4	0.5
T_S	Hue reliability threshold	0.2	0.2
α_{T_d}	Inertia of adaptive threshold T_d	0.4	0.7
$T_{d'}$	Invariant threshold	8	10

The sensitivity to parameters has been evaluated in different ways, either numerically or graphically. In tables 3 and 4 respectively dedicated to the football and the news dataset, we have measured for each method the number of parameter sets which help to reach a given quality considering the F1 measure (each F1 measure being computed for a given parameter set). Our method with its 1296 possible parameters has been compared to the classic pixel-based and histogram-based approaches with 1001 parameter sets. We have also experimented our method with a space of 14641 parameter sets, but the improvement was relatively small.

From these tables, we can observe that our method will ensure a reasonable quality with a large set of parameters: quality higher than 80 % (resp. 70 %) is reached with 30 % (resp. 67 %) of the possible parameter values on the football dataset. News are easier to process since already 62 % of the parameters yield a quality higher than 90 %. In comparison, the other approaches are far more sensitive to parameter values: with both methods, only 10 % of the parameters give a quality higher than 90 % for the news dataset, whereas on the football dataset, 2 % or less of the parameters ensure a quality at least equal to 80 %.

Table 3 Statistical quality representation on the football dataset in the available space of parameter sets for our method, and the classical histogram-based and pixel-based methods. Each column represents the proportion of parameter sets which ensures the F1 quality measure.

quality	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
our method	0	0.30	0.67	0.79	0.83	0.83	0.91	0.92	0.94	1
histogram	0	0	0.21	0.39	0.48	0.62	0.69	0.75	0.82	1
pixel	0	0.02	0.11	0.13	0.16	0.20	0.27	0.32	0.35	0.36

Table 4 Statistical quality representation on the news dataset in the available space of parameter sets for our method, and the classical histogram-based and pixel-based methods. Each column represents the proportion of parameter sets which ensures the F1 quality measure.

quality	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
our method	0.62	0.79	0.82	0.83	0.83	0.83	0.83	0.83	0.90	1
histogram	0.09	0.22	0.35	0.53	0.57	0.59	0.63	0.71	0.81	0.99
pixel	0.11	0.14	0.20	0.22	0.24	0.28	0.31	0.32	0.34	0.37

In complement to these tables, figure 4 shows a graphical representation of the quality ensured by all the possible sets of parameters taken from the complete space considering the F1 quality measure. Once again we can observe that our method is far more robust to parameter settings than other approaches.

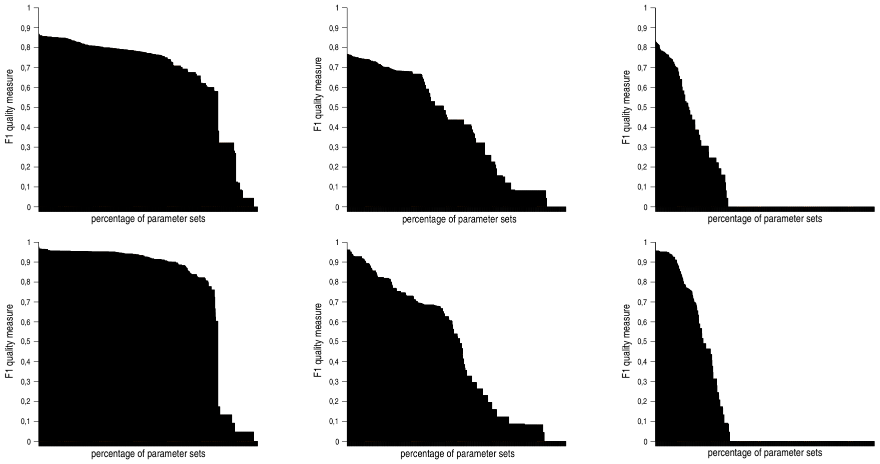

Fig. 4 Visual representation of the F1 measure in the available space of parameter sets for our method (left) and the classical histogram-based (middle) and pixel-based (right) methods. The corpus considered is the football dataset (top) and the news dataset (bottom).

4.4 Comparison between the RGB and HSL colour spaces

In our method, we have decided to use the HSL colour space (or more precisely the HS subspace) instead of the classical RGB space. We present here some experiments that validate our choice.

Figure 5 shows the evolution of measures d, d', and d'_{sum} for video sequences containing different types of transitions, considering both the HSL and the RGB spaces. In the first example (top line, two cuts), using the RGB space results in many false detections. In the second example (middle line, a wipe), the results obtained with the two spaces are relatively similar. Finally, in the last example (bottom line, a fade), the transition is not detected in the RGB space whereas false detections are identified. Thus we can observe in the three representative examples that the contrast of local extremum values is more significant in the HSL space than in the RGB space. This colour space choice ensures a higher robustness of the method, and confirms our theoretical assumption. We recall the reader is referred to [9] for a complementary study on colour histogram-based shot detection.

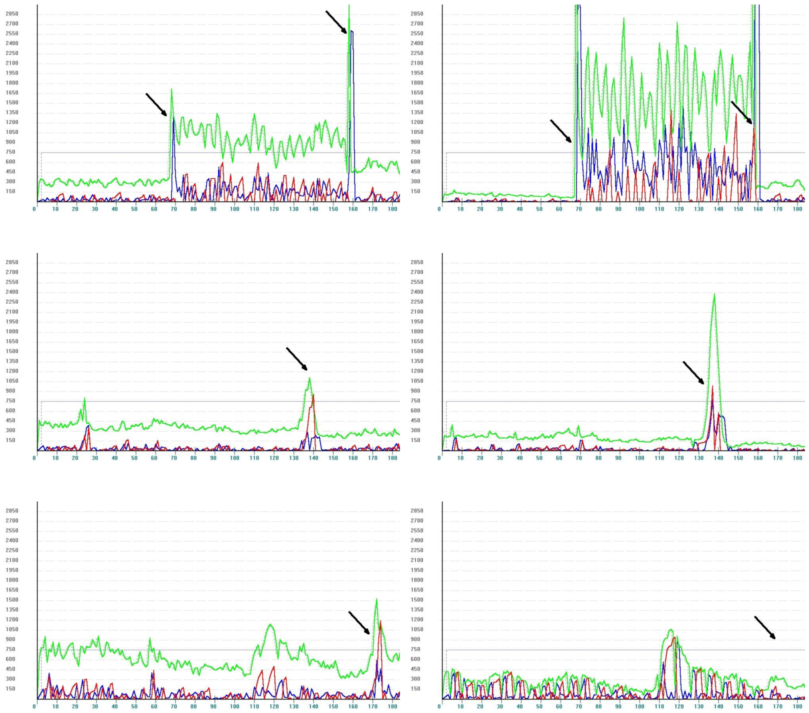

Fig. 5 Temporal evolution of measures d (in green), d' (in red), and d'_{sum} (in blue) based on the HSL (left) or the RGB (right) space, for a sequence containing two abrupt transitions (top), a progressive transition of type wipe (middle) and a progressive transition of type fade (bottom) indicated by the black arrows.

5 Conclusion

In this paper we adressed shot change detection and presented a solution which ensures robustness and reliability without discarding efficiency even with uncompressed video data. Contrary to other recent approaches requiring either higher computational cost or higher specificity (i.e. limited to a given compression scheme) to reach satisfactory results, our method may be seen as a simple but reliable technique.

We have decided to decrease the amount of data to be processed by reducing the number of image pixels and the number of colour components to be used. We have chosen to represent the pixels by two chromatic parameters (Hue and Saturation), in order to deal correctly with artefacts related to illumination changes. The data reduction step is independent of the nature of video sequences, and enables the process to consider either uncompressed or compressed video sequences. Motion effects are very important as they may induce a high number of errors during the detection step. In order to consider this constraint and to limit the error rate, we have introduced an adaptive threshold but also an original way of computing the dissimilarity measures between frames. More precisely, we have based our process on the study of the evolution of these measures instead of the measures themselves. We have validated our approach in two different contexts (TV news and football videos) to emphasize the robustness of our method, in particular to the type of video sequences to be considered and to the parameter settings.

We are now considering a validation on a very large scale, using for instance the database made available by the TREC Video community [17]. Moreover, we are thinking of integrating our detector into a multimedia information system to let the user build some complex requests on video data. Finally, artefacts due to motion could be better taken into account, once a precise identification of the effects of motion on the dissimilarity measure has been achieved.

References

1. Bezzera, F., Leite, N.: Using string matching to detect video transitions. Pattern Analysis and Applications **10**, 45–54 (2007)
2. Boccignone, G., Chianese, A., Moscato, V., Picariello, A.: Foveated shot detection for video segmentation. IEEE Transactions on Circuits and Systems for Video Technology **15**(3), 365–377 (2005)

3. Boussaid, L., Mtibaa, A., Abid, M., Paindavoine, M.: A real-time shot cut detector: hardware implementation. Computer Standards and Interfaces **29**(3), 335–342 (2007)
4. Cao, J., Cai, A.: A robust shot transition detection method based on support vector machine in compressed domain. Pattern Recognition Letters **28**, 1534–1540 (2007)
5. Cernekova, Z., Pitas, I., Nikou, C.: Information theory-based shot cut/fade detection and video summarization. IEEE Transactions on Circuits and Systems for Video Technology **16**(1), 82–91 (2006)
6. Chen, C.: Video compression: Standards and applications. Journal of Visual Communication and Image Representation **4**(2), 103–111 (1993)
7. Cheng, S., Wu, T.: Scene-adaptive video partitioning by semantic object tracking. Journal of Visual Communication and Image Representation **17**, 72–97 (2006)
8. Fang, H., Jiang, J., Feng, Y.: A fuzzy logic approach for detection of video shot boundaries. Pattern Recognition **39**(11), 2092–2100 (2006)
9. Gargi, U., Kasturi, R., Strayer, S.: Performance characterization of video-shot-change detection methods. IEEE Transactions on Circuits and Systems for Video Technology **10**(1), 1–13 (2000)
10. Grana, C., Cucchiara, R.: Linear transition detection as a unified shot detection approach. IEEE Transactions on Circuits and Systems for Video Technology **17**(4), 483–489 (2007)
11. Janvier, B., Bruno, E., Pun, T., Marchand-Maillet, S.: Information-theoric temporal segmentation of video and applications: multiscale keyframes selection and shot boundaries detection. International Journal of Multimedia Tools and Applications **30**, 273–288 (2006)
12. Joyce, R., Liu, B.: Temporal segmentation of video using frame and histogram space. IEEE Transactions on Multimedia **8**(1), 130–140 (2006)
13. Lefèvre, S., Holler, J., Vincent, N.: A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. Real-Time Imaging **9**(1), 73–98 (2003)
14. Li, S., Lee, M.: Effective detection of various wipe transitions. IEEE Transactions on Circuits and Systems for Video Technology **17**(6), 663–673 (2007)
15. Mandal, M., Idris, F., Panchanathan, S.: A critical evaluation of image and video indexing techniques in the compressed domain. Image and Vision Computing **17**(7), 513–529 (1999)
16. Mardia, K., Jupp, P.: Directional Statistics. Wiley & Sons Ltd. (2000)
17. Over, P., Ianeva, T., Kraaij, W., Smeaton, A. (eds.): Proceedings of TRECVID 2005 (2005)
18. Qian, X., Liu, G., Su, R.: Effective fades and flashlight detection based on accumulating histogram difference. IEEE Transactions on Circuits and Systems for Video Technology **16**(10), 1245–1258 (2006)
19. Travis, D.: Effective Color Displays. Theory and Practice. Academic Press (1991)

20. Yeo, B., Liu, B.: Rapid scene analysis on compressed video. IEEE Transactions on Circuits and Systems for Video Technology **5**(6), 533–544 (1995)
21. Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., Lin, F., Zhang, B.: A formal study of shot boundary detection. IEEE Transactions on Circuits and Systems for Video Technology **17**(2), 168–186 (2007)
22. Zhai, Y., Shah, M.: Video scene segmentation using markov chain monte carlo. IEEE Transactions on Multimedia **8**(4), 686–697 (2006)

S. Lefèvre was granted in 1999 the M.S and Eng. degrees from the University of Technology of Compiègne in Computering Engineering, and in 2002 a Ph.D. degree in Computer Sciences from the University of Tours. From 1999 to 2002 he was with AtosOrigin as a Research and Development Engineer. He is currently an Assistant Professor in the Department of Computer Sciences and the LSIIT (Models, Images and Vision team), University Louis Pasteur, Strasbourg. His research interests are in image and video processing, multimedia analysis and indexing, and mathematical morphology.

N. Vincent defended a Ph.D. in Computer Sciences in 1988 from the Insa of Lyon after studying at Ecole Normale Supérieure obtaining the agregation in Mathematics. She became full Professor at the University of Tours in 1996 and moved to Paris Descartes University in 2003, where she heads the Centre of Research in Computer Science (CRIP5) and the team Systèmes Intelligents de Perception (SIP). She is specialised in pattern recognition, signal and image processing and video analysis.