Scene break detection: a comparison

G. Lupatini, C. Saraceno & R. Leonardi, SCL-DEA
University of Brescia, Brescia I-25123, Italy
E-mail: {lupatini,saraceno,leon}@imago.ing.unibs.it

Abstract

The automatic organization of video databases according to the semantic content of data is a key aspect for efficient indexing and fast retrieval of audio-visual material. In order to generate indices that can be used to access a video database, a description of each video sequence is necessary. The identification of objects present in a frame and the track of their motion and interaction in space and time, is attractive but not yet very robust. For this reason, since the early 90's, attempts have been applied in trying to segment a video in shots. For each shot a representative frame of the shot, called k-frame, is usually chosen and the video can be analyzed through its k-frames.

Even if abrupt scene changes are relatively easy to detected, it is more difficult to identify special effects, such as dissolve, that were operated in the editing stage to merge two shots. Unfortunately, these special effects are normally used to stress the importance of the scene change (from a content point of view), so they are extremely relevant therefore they should not be missed. Beside, it is very important to determine precisely the beginning and the end of the transition in the case of dissolves and fades. In this work, two new parameters are proposed. These characterize the precision of boundaries of special effects when the scene change involves more than two frames. They are combined with the common recall and precision parameters. Three types of algorithms for cut detection are considered: histogram-based, motion-based and contour-based. These algorithms are tested and compared on several video sequences. Results will show that the best performance is achieved by the global histogram-based method which uses color information.

1 Introduction

A shot represents a sequence of frames captured from a unique and continuous record from a camera. Therefore adjacent frames of the same shot exhibit temporal continuity. Once a video sequence is segmented into shots it becomes easy to establish the context of the overall video with only some representative frames (k-frames): for each shot one or more frames can be chosen as representative of the shot, depending on the amount of motion that is present. Besides, the segmentation of the video into shots is extremely important as a first step for content-based segmentation of digital video material. Each shot corresponds to a single continuous action and no change of content can be detected inside a shot. Change of contents always happen at the boundary between two shots. Partitioning a video sequence into shots is also useful for coloration of black and white movies, in fact, each shot has a different associated gray-to-color look-up table. In order to study movie directors' styles it maybe interesting to consider film partitioned into individual shots, allowing to measure the coverage shot length and the type of edit effects that have been used.

The main problem, when segmenting a video sequence into shots, is the ability to distinguish between scene breaks and normal changes that happen in the scene. These changes may be due to the motion of large objects or to the motion of the camera (e.g. zoom, pan, tracking and so on). In case of abrupt changes, the change due to the scene cut is usually very large and easy to detect. When special effects are involved, two shots are merged (in the editing process) using gradual transition: the evolution from one shot to another is few frames long, and each frame, in the gradual transition, differs from the previous one by a small amount. Most used types of gradual transitions are: dissolve, fade in and fade out. Gradual transitions are used less than cuts, but they are often chosen to stress the change in the "semantic" content of the sequence, therefore their detection becomes extremely important.

The aim of this paper is to describe and compare some algorithms that are able to detect cuts and editing effects. Due to the importance of the gradual transition detection, two new parameters will be defined and used together with the classical measure of recall and precision in the evaluation process. These parameters correspond to the precision and recall on the covered portion of the gradual transition [1]. They consider the correct alignment of the boundaries of the gradual transition with respect to the ones that are obtained by the algorithms. Three types of algorithms are chosen: the histogram-based, the motion-based and the contour-based methods. The histogram-based methods have been chosen for their tolerance to the presence of motion in the scene. The motion-based techniques have been chosen for their ability to explicitly handle motion (eliminating the motion, the remaining motion compensated difference between frames should be mainly due to scene breaks). The contour-based technique have been chosen because of their ability in detecting gradual transition. The algorithms are tested on several types of video sequences

[1] see section 4 for details

to find the best method for scene break detection.

In the next section, a brief overview of the existing scene break detection techniques is proposed. Section 3 describes the algorithms we have considered for comparisons. The performance of the proposed algorithms are evaluated and compared in section 4. Conclusions and future developments are finally discussed in the last section.

2 Previous works

In order to segment a video sequence into shots a dissimilarity measure D(f1,f2) between two frames (f1,f2) must be defined. This measure must return a high value only when the two frames fall in different shots. [1] considers a dissimilarity measure based on a pixel-pixel comparison between two frames. The measure is obtained by counting the number of differences between corresponding pixel luminance that exceed a certain threshold: if this number is large enough, the two processed frames are declared to belong to different shots. This measure is highly sensitive to motion. In fact the motion of an object or of the camera may be confused with a scene break. In order to overcome this problem, [3] and [1] suggest histogram-based methods. They evaluate the difference between the histograms of the two frames of interest, if this difference is high enough a scene break is expected. [6] compares several histogram difference measures, where the histograms are evaluated on the luminance function. The best performance is obtained by the χ^2 test. The use of the luminance information only, may produce false detection due to the presence of strong luminance changes, for this reason [3] suggests to analyze color histograms. In this case, each pixel is represented with a color code obtained by merging the most significant bits of each color component.

Unfortunately, histogram based algorithms fail when frames of different shots have similar histograms. This is due to the global measure represented by the histogram. In fact histograms ignore completely the spatial distribution of the luminance. Consecutive frames which have different spatial distribution of the luminance, but similar histograms, are declared to belong to the same shot. A solution to this problem is proposed by [3] and it consists in splitting each frame into 16 blocks all having the same size (on a 4x4 pattern) and in evaluating the difference between corresponding histograms. This way the method is more tolerant to the motion present in each block, but it is also sensitive to changes of spatial distribution of the luminance over the entire frame. In other words, this region adapted technique combines the advantages given by histogram methods and point-to-point views, such as pixel to pixel differences.

Histogram based methods, described so far, show good performance in case of abrupt scene changes such as cuts. Unfortunately, in presence of dissolve, the difference between consecutive frames may be too low to be misinterpreted as a difference due to motion. For this reason, [1] suggests a method called "twin comparison". This method compares the global histogram differences with respect to two thresholds: a low one (t_l) is used to find possible boundaries of dissolves, while the high one (t_h) is used to confirm the presence of a dissolve (for more detail see section 3.1).

In order to distinguish between changes in the scene due to motion from the ones due to a scene break, different motion based algorithms have been proposed. [4] uses a block-matching algorithm on 12 equal size blocks obtained by splitting the frame on a 4x3 rectangular pattern. The motion-compensated difference values are used to determine scene breaks. Being the main source of difference due to the motion which can be eliminated by motion compensation, the remaining difference is due to other causes such as the occurrence of a scene break. The problem of such method is that block-matching performs well only for a particular type of motion (e.g., translational motion). Therefore the scene-break detection appears very sensitive to motion that the block matching is not able to handle (e.g., rotation, occlusion between objects). [2] works on MPEG compressed video sequences and uses the motion vectors contained in the MPEG file. If the number of motion vectors used is high enough, [2] assumes that it is possible to predict one frame from another simply using motion compensation. On the other hand, if such number is low it means that the two frames may be too different to be predicted one from the other and a scene break may have occurred. This technique is clearly dependent on the quality of the MPEG codec.

Another method oriented to detect dissolves (or, more generally, gradual transitions) is proposed by [5]. It performs a spatial comparison of contours in adjacent frames. More precisely, when the number of contour points that appear or disappear in a frame, with respect to the previous frame, is high enough, then a scene break is expected.

3 Tested algorithms

3.1 Histogram based algorithms

Histogram based algorithms are the most common scene break detection techniques because of their simplicity and the good results they can achieve. In the literature several types of histogram based algorithms exist and they all indicate that the χ^2 test has in general better performance with respect to other measures such as the Yakimovsy likelihood ratio test and the Kolmogorov-Smirnov Test. For this reason, we have used the χ^2 test to compare two histograms (H_1 and H_2) as proposed by [6].

$$\chi^2 = \sum_{i=1}^{k} \begin{cases} \frac{(H_1(i)-H_2(i))^2}{max(H_1(i),H_2(i))} & if(\mathrm{H_1(i)} \neq 0) \vee (\mathrm{H_2(i)} \neq 0) \\ 0 & otherwise \end{cases} \quad (1)$$

where $H_j(i)$ is the bin i value of the histogram of frame j, and k is the overall number of bins.

Three types of histograms have been considered: color, hue and luminance. Color histograms are obtained by representing each pixel by a color code [3]. The color code is obtained by merging the most significant bits of each RGB component; we have considered a six bit code (2 bits for each component) and a nine bit code (3 bits for each component). Hue histograms

are obtained by considering the hue component of each pixel. Following [7] we have:

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

$$H = tan^{-1}(\frac{V_2}{V_1}) \quad (3)$$

$$S = \sqrt{{V_1}^2 + {V_2}^2} \quad (4)$$

where the I, H and S are the intensity, hue and color saturation components, respectively; and V_1 and V_2 are dummy variables.

Global and local histograms [3] have also been considered. Local histograms are computed on regions obtained by splitting each frame into 16 equal size regions using a 4x4 rectangular lattice. The difference between two frames is evaluated by considering the 16 difference measures of the corresponding histograms (using the χ^2 test) for the two frames. The 8 largest differences are discarded while the remaining 8 allow to detect the scene break. The former are discarded in order to make the method more robust in presence of motion of local objects.

These techniques alone would allow the detection of abrupt scene changes, but they would not allow to detect special effects such as dissolves. The detection of gradual transitions is obtained using the twin comparison technique [1]. This method uses two thresholds, an high threshold (t_h), and a low threshold (t_l), as described in section 2. Whenever the difference measure between consecutive frames exceeds the high threshold a cut is declared; when the difference exceeds only the low threshold the current frame becomes a candidate first frame of a gradual transition. The candidate frame is compared with successive frames and if the histogram difference exceeds the high threshold, before the difference between consecutive frames fall below the low threshold, a gradual transition is declared. This method offers good performance if the motion does not change in a relevant way the context of the scene. To reduce false detection Zhang et al. [1] suggest to perform a motion analysis on the candidate frame. For simplicity purpose, we have discarded this issue.

By combining the histogram methods described so far, we have implemented and tested eight histogram based algorithms using:

1. Global histogram evaluated on luminance information with 256 bins for each histogram, such as that described in [1] (H1).

2. Local histogram evaluated on luminance information with 64 bins for each histogram (H2).

3. Global histogram evaluated on 6-bit color code information (H3).

4. Local histogram evaluated on 6-bit color code information (H4).

5. Global histogram evaluated on 9-bit color code information (H5).

6. Local histogram evaluated on 9-bit color code information (H6).

7. Global histogram evaluated on hue information with 256 bins for each histogram (H7).

8. Local histogram evaluated on hue information with 64 bins for each histogram (H8).

3.2 Motion based algorithms

In presence of significant motion, histogram based algorithms usually detect a scene break, where such a scene break does not exist (in other words causing a false alarm). In order to overcome such a problem, motion based cut detection algorithms have been proposed. A block–matching based algorithm has been considered in our tests (with block size of 16x16) to perform motion compensation.

Inspired by [4] we evaluate the block matching of consecutive frames and add up the motion compensated difference values of each block. If this sum exceeds a predetermined threshold, the two processed frames are declared to belong to a scene break. Since such values are obtained by a pixel-pixel comparison, this method may cause the same problem which is encountered in the pixel-based method (highly sensitive to local motion). In order to overcome this problem, a second technique has been implemented. Instead of considering pixel differences, the average of the luminance function is evaluated for each block. Then the difference is calculated between the average luminance function of each block in the current frame, and the average luminance function of the block in the previous frame, which matches the the current blocks best. In addition, the absolute values of these differences are added up. The algorithm will declare a scene break once this sum exceeds another threshold. In order to increase the algorithm's tolerance towards motion, a third algorithm has been implemented. This algorithm adds up the number of blocks with a motion-compensated difference signal that exceeds a third threshold: if this number exceeds a fourth threshold, then a change in the scene is declared.(For the determination of the thresholds see below.)

In conclusion we have built 3 algorithms based on motion compensation that respectively use the following 3 difference measures:

1. Sum of absolute motion compensated luminance difference values, as proposed in [4].

2. Sum of the absolute difference of the average of the luminance of each block.

3. Number of blocks with absolute motion compensated luminance difference values that exceed a certain threshold, as proposed in [2].

3.3 Contour based algorithms

Like in [5] we use edge information to detect scene breaks. We aim that during a dissolve the number of edge points that appears or disappears (with respect to the previous frame) is large enough to produce a good difference measure for dissolve detection. Our algorithm works as follows:

1. It performs a motion compensation on the two frames of interest using the block- matching technique already used for the motion based algorithms described previously (while [5] uses a global motion compensation);
2. It filters the frames with a low-pass spatial filter to eliminate noise and to smooth the boundary of the block used during the block matching;
3. It extracts the edges of the frames using the Sobel operator [7];
4. It determines the number of edge points that appears in a given frame with respect to the previous one. In order to handle non translational motion (e.g., rotations) that could not have been compensated by step 1, the edges of the previous frame are dilated. Dilation is obtained considering a mask with a squared shape of side l; the mask is centered on each edge point, and all the points covered by the mask are set to be edge points in the dilated version. Appeared edge points are by definition all the edge points of the actual frame which are not edge points in the dilated edge version of the previous frame.
5. In a dual fashion the disappeared edge points are determined: this time the edges of the actual frame are dilated and compared with the edges of the previous frame.
6. The difference measure is represented by the maximum value between the fraction of edge points that appear and the ones that disappear.

4 Simulation results

4.1 Sequences features

The implemented algorithms are tested on a set of video sequences. These sequences are captured from VHS at a frame rate of 15 frm/sec with a 192x144 frame size and compressed in MVC1 format using an INDIGO2 SGI computer using Galileo 601 and Cosmocompress. The sequence contains 33 minutes of a film, 33 minutes of a documentary, 33 minutes of news and 16 minutes of advertising material. Table 1 describes each type of processed visual material in terms of number of frames, duration, number of cuts and dissolves.

4.2 Performance parameters

Usually the performance of a cut detection algorithm is expressed in terms of recall and precision. The recall parameter defines the percentage of true detection (performed by the detection algorithm) with respect to the overall events (scene breaks) actually present in the sequence. In a dual fashion the precision is the percentage of true detection with respect to the overall declared event. The equation for recall and precision are the following:

Sequence	Frame Number	Length		Scene Break	
		min	sec	cut	diss
Film	29982	33	18	247	10
Documentary	30421	33	48	308	7
News	30360	33	44	339	23
Advertising	14601	16	13	518	116
Total	105364	117	4	1412	156

Table 1: Features of the processed video-sequences

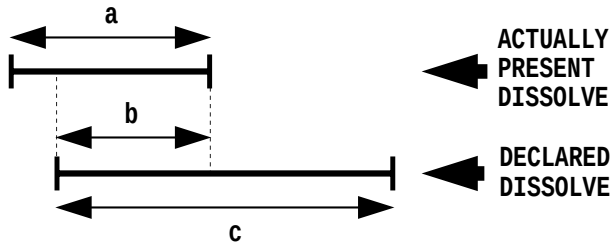

Figure 1: A possible dissolve detection. The upper segment represents the actual dissolve present in the video sequence, instead the lower is the dissolve declared by the detection algorithms.

$$Recall = \frac{Nc}{Nc + Nm} * 100\% \quad (5)$$

$$Precision = \frac{Nc}{Nc + Nf} * 100\% \quad (6)$$

where, Nc is the number of correct detections, Nm is the number of missed detections, Nf is the number of false detections, $Nc + Nm$ is the number of the existing events and $Nc + Nf$ is the number of overall declarations.

Recall and precision are usually the only two parameters used to evaluate the performance of a shot cut detector. In case of dissolves, these two parameters do not take into account the precision of the detection. The detected dissolve does not always coincide with the real dissolve, sometimes it is included in the real dissolve. Sometimes it ends a few frames later. In order to consider such "partial" mistakes, two new parameters have been defined: *cover recall* and *cover precision*.

The cover recall is defined as the percentage of covered length of correct detected dissolve with respect to the length of the real dissolve; in a dual fashion we define the cover precision.

Eq.(7) and (8) define the recall and precision cover parameters for the situation shown in Fig.1:

$$Recall_{cover} = \frac{b}{a} * 100\% \quad (7)$$

$$Precision_{cover} = \frac{b}{c} * 100\% \quad (8)$$

where a is the length of the real dissolve, c is the length of the declared dissolve and b is the length of the real dissolve covered by the declared dissolve.

We consider a scene break (a cut or a dissolve) correctly detected if at least one of its frames has been detected as a scene break. Sometimes, when two dissolves are close to each other, just a single dissolve is detected. In this case, we aim that just the first dissolve has been detected, while the second one has been missed.

5 Performance evaluation

The thresholds for each algorithm are chosen testing 10 minutes of advertising material while achieving whenever possible, a recall of 90% for cut and 70% for dissolve. Tables 2, 3, 4 and 5 report the performance of the evaluated algorithms for each type of video material, Table 6 reports the performance evaluated on the overall considered video sequence.

Each table reports the precision and recall parameters for cut, dissolve and cover. H1, H3, H5, H7 analyze global histograms evaluated on luminance, 6-bit color code, 9-bit color code, and hue information respectively. H2, H4, H6, H8 use local histograms evaluated on luminance, 6-bit color code, 9-bit color code, and hue information respectively. Motion based algorithms are labeled as M1, M2 and M3 which respectively consider the sum of the absolute motion compensated difference values, the sum of the absolute differences of average luminance values and the number of blocks presenting a high average motion compensated difference value. The edge-based algorithm is labeled as E.

The algorithms in each table are listed according to their performance, and grouped in three classes: high, middle and low performance (each class is separated by a blank row). The order is made on a subjective scale oriented to find the best trade-off between the parameters. A scale of importance among the 6 parameters has been defined as follows: 1 $Recall_{diss}$, 2 $Recall_{cut}$, 3 $Precision_{diss}$, 4 $Precision_{cut}$, 5 $Recall_{cover}$, 6 $Precision_{cover}$.

No order has been defined inside each class.

Looking at the tables 2 to 6, the cover precision parameters frequently assume very low values. In fact, sometimes the algorithms declare a dissolve when a cut occurs (usually, this is due to the presence of motion before and after the cut). In such a case only one frame of the declared dissolve is covered by the actual cut. The best performance is achieved by the histogram-based algorithms. This is cause by the use of the twin comparison method that allows to utilize a low threshold without significantly increasing the number of false transitions. Unfortunately, twin comparison combined with the χ^2 test, based on histogram comparison, fails when a continuous motion changes completely the content of the scene in terms of histogram representation (this situation may be produced by long tracking or pan actions). In other words, considering the group of frames from the beginning to the end of a continuous motion, the twin comparison fails when the histogram corresponding to the first frame of this group differs completely from the histogram of the last frame.

Good results are obtained in particular by the histogram based technique using the hue component and the color code (H3, H4, H5, H6, H7, H8). Thus, the use of color code and hue component actually makes the method more robust to luminance changes. On the other hand, histograms evaluated on luminance information (H1, H2) may fail in presence of strong changes in illuminance (such as a flash), but also small illuminance changes (such as partial occlusion of an illumination source) may cause false detection. Looking at the abrupt change (cut) and gradual transition (diss.) parameters, there are no big differences between the performance of algorithms using local features (H2, H4, H6, H8) and those using global features (H1, H3, H5, H7). Global histograms show a good tolerance towards motion due to the small resolution of the processed frames, while subdividing a small size frame into regions may not be significant. The cover parameters demonstrate that the global histograms are more accurate in the detection of the boundary position of gradual transitions. On the other hand, local histograms may fail in the correct determination of such a boundary, when the average luminance of the next shot is similar to the previous. During the dissolve, the difference between consecutive frames could be localized just in some regions, whereas the local histogram methods may reject these very regions causing a missing in the detection.

The motion-based algorithms (M1, M2, M3) usually exhibit lower performance, except for the one based on the absolute difference of the average luminance in blocks (M2). These algorithms occupy a middle position, because the motion compensation cannot handle every type of motion. In particular, the M1 algorithm, performing a pixel-pixel comparison, is extremely unreliable when blocks contain moving objects. A better performance is achieved by the M2 algorithm, because the local motion present in the block is ignored by averaging the luminance values of the block. M2 achieves good performance in detecting cuts in the film sequence (99.19% for Recall and 84.56% for Precision). For complex video sequences, such as advertising, M2 exhibits lower performance, due to the relevant presence of motion: high speed of the objects and large use of camera motion. Algorithm M3 cannot be used for the detection of gradual transition (very poor results are obtained in case of Film and Documentary), because of the small dynamic of the measure of difference used. In fact, each frame is split in 12x9 blocks. Therefore the measure can assume just 108 distinct values.

The edge based algorithms exhibit high performance in the case of films. In all other types of video, the performance can be considered mediocre. This method, like the model-based method, is very sensitive to motion. The translation is handled by the block-matching algorithm while the local motion is handled

	RECALL			PRECISION		
	Cut	Diss	Cover	Cut	Diss	Cover
H8	**99.19%**	80%	17.69%	90.87%	33.01%	31.01%
H1	97.98%	**100%**	**76.58%**	96.13%	**48.36%**	30.66%
E	96.36%	70%	33.05%	65.52%	8.43%	5.54%
H2	98.79%	60%	16.66%	95.14%	28.3%	20.14%
H7	95.55%	**100%**	26.58%	87.55%	56%	**49.00%**
M2	**99.19%**	30%	44.18%	84.56%	27.88%	11.22%
H6	**99.19%**	30%	57.14%	91.97%	27.14%	26.31%
H5	90.28%	70%	36.36%	100%	50%	33.02%
H3	90.28%	60%	41.22%	98.5%	36.36%	36.68%
H4	93.12%	30%	20.27%	**99%**	35.05%	14.6%
M3	97.57%	10%	7.14%	82.31%	21.21%	4.17%
M1	98.79%	0%	0%	93.2%	32.35%	13.86%

Table 2: Performance on Film sequence

	RECALL			PRECISION		
	Cut	Diss	Cover	Cut	Diss	Cover
H7	**98.38%**	**100%**	41.66%	85.83%	48.17%	26.13%
M2	97.73%	85.71%	61.29%	72.62%	29.83%	6.67%
H6	93.83%	**100%**	**81.94%**	98.26%	57.99%	7.546%
H4	92.21%	**100%**	75%	**99.35%**	56.56%	8.75%
H1	94.16%	85.71%	64.51%	96.14%	60.63%	19.09%
H3	93.51%	85.71%	75.8%	97.42%	41.61%	**26.66%**
H8	93.89%	71.43%	50%	96.15%	42.59%	9.97%
H5	92.21%	85.71%	77.41%	99.08%	**67.89%**	24.46%
M1	95.45%	28.57%	40%	69.94%	30.35%	4.7%
E	91.88%	42.86%	41.66%	60.52%	36.94%	3.28%
H2	89.94%	57.14%	62.22%	97.02%	46.09%	14.63%
M3	93.83%	0%	0%	68.55%	27.88%	5.04%

Table 3: Performance on Documentary sequence

	RECALL			PRECISION		
	Cut	Diss	Cover	Cut	Diss	Cover
H5	95.87%	**100%**	90.21%	**98.72%**	**71.17%**	16.62%
H3	**96.76%**	95.65%	85.87%	97.27%	32.73%	**22.21%**
H2	95.28%	95.65%	78.8%	94.84%	56.99%	11.45%
H4	92.33%	86.96%	91.97%	96.98%	54.05%	9.23%
M2	94.4%	78.26%	73.1%	66.76%	42.44%	7.48%
H6	90.56%	82.61%	**93.71%**	96.77%	58.87%	7.51%
E	95.58%	73.91%	36.56%	76.1%	38.1%	7.91%
H7	88.79%	79.91%	77.24%	89.75%	61.54%	4.55%
H8	90.86%	73.91%	68.49%	91.7%	56.1%	6.2%
H1	86.43%	73.91%	92.59%	97.57%	63.01%	5.34%
M1	96.76%	43.48%	46.15%	83.04%	42.86%	6.23%
M3	93.51%	47.83%	37.8%	71.84%	37.68%	4.16%

Table 4: Performance on News sequence

	RECALL			PRECISION		
	Cut	Diss	Cover	Cut	Diss	Cover
H7	**91.7%**	73.28%	54.76%	93.92%	81.35%	**36.75%**
H2	90.35%	75.86%	65.24%	96.33%	65.28%	20.45%
H3	89.19%	**77.59%**	63.95%	98.09%	83.28%	36.37%
H8	87.84%	65.52%	48.71%	94.51%	83.27%	23.05%
H1	87.07%	65.52%	74.74%	96.12%	88.89%	25.8%
E	87.07%	64.66%	38.19%	90.46%	85.84%	18.45%
H5	83.98%	68.97%	71.57%	**99.22%**	89.73%	26.69%
M1	84.75%	46.55%	**91.81%**	91.81%	88.18%	14.61%
M2	80.89%	65.52%	59.12%	87.71%	83.8%	17.87%
H4	79.34%	61.21%	66.26%	98.58%	64.04%	12.67%
M3	80.89%	39.66%	34.04%	86.94%	81.36%	11.9%
H6	74.71%	62.07%	67.68%	99.04%	**90%**	13.49%

Table 5: Performance on Advertising sequence

	RECALL			PRECISION		
	Cut	Diss	Cover	Cut	Diss	Cover
H7	**93.13%**	76.28%	52.91%	89.81%	64.79%	16.05%
H2	92.92%	76.92%	62.21%	96.33%	65.25%	20.45%
H3	92.14%	**79.49%**	65.71%	97.78%	63.5%	**31.05%**
H5	89.73%	74.36%	71.13%	**99.23%**	**76.9%**	23.49%
H1	90.37%	69.87%	**76.73%**	96.5%	69.03%	15.66%
H8	91.86%	67.95%	47.64%	93.46%	59.02%	12.46%
M2	91.01%	66.03%	60.74%	76.86%	51.14%	11.93%
E	91.78%	65.38%	37.39%	74.14%	31.35%	8.21%
M1	92.42%	42.31%	40.3%	83.94%	54.08%	10.05%
H4	87.68%	64.74%	67.69%	98.58%	64.04%	12.67%
H6	86.97%	64.74%	72.78%	96.2%	66.59%	10.32%
M3	89.66%	37.18%	33.92%	76.96%	47.7%	7.42%

Table 6: Overall Performance

considering the edge dilated version. These techniques are not able to deal with complex movements, such as rotations, and effects, such as occlusions. An object that overlaps a second object causes the disappearance of a certain number of edge points. This event may be confused by the detection algorithm as a fade out. In particular, the low performance in the documentary sequence is due to the features of the sequence. The elements present in this type of scenes have not a homogeneous distribution of colors (e.g., an animal's fur or a desert), the number of detected edge points is very high, making it hard to distinguish between edge points that appear and disappear. A solution to this problem would be to smooth the frame, but at this point the method becomes insensitive to little differences caused by a dissolve.

6 Conclusions

Twelve scene break detection algorithms have been considered inspired by 3 base methodologies: histogram-based, motion-based and contour-based. The performance of these algorithms has been evaluated on 4 types of video sequences: films, news, documentaries and advertising material, for an overall 2 hours duration. The better performance was achieved by histogram-based algorithm that uses color or hue information, exhibiting a recall of 93% for cut and 76% for dissolve and a precision of 89% for cut and 65% for dissolve (see Table 6 row H7) on the overall sequences (including advertising).

Future works should include global camera motion compensation (zoom, pan and tilt) to obtain better identification of gradual transitions, when such type of motion occurs. Emphasis will be placed in making such methods quite simple.

References

[1] H.J. Zhang, A. Kankanhalli, S.W. Smoliar: "Automatic Partitioning of Full-motion Video" *Multimedia Systems 1993*, Vol.1, No.1, pp. 10-28

[2] H.J. Zhang, A. Kankanhalli, S.W. Smoliar: "Video Parsing and Browsing using compressed data", *Multimedia Tools and applications 1995*, pp. 89-111

[3] A. Nagasaka, Y. Tanaka: "Automatic Video Indexing and Full-video Search for Object Appearances", *Visual database Systems II 1992*, pp. 113-127

[4] B. Shahraray: "Scene Change Detection and Content-Based Sampling of Video Sequences", *Digital video compression: algorithms and technologies 1995*, pp. 2-13

[5] R. Zabhi, J. Miller, K. Mai: "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", *Proc. ACM Multimedia 95*, pp. 189-200

[6] I. K. Sethi, Nilesh Patel: "A Statistical Approach to Scene Change Detection", *SPIE Vol. 2420 1995*, pp. 329-338

[7] W. K. Pratt: "Digital Image Processing" *2nd ed., John Wiley & sons, in., 1991* , pp. 62-78, pp.491-556.