Simple and Robust Hard Cut Detection using Interframe Differences

Alvaro Pardo[1,2]

[1] DIE, Facultad de Ingeniería y Tecnologías, Universidad Católica del Uruguay
[2] IIE, Facultad de Ingeniería, Universidad de la República

Abstract. In this paper we introduce a simple method for the detection of hard cuts using only interframe differences. The method is inspired in the computational gestalt theory. The key idea in this theory is to define a meaningful event as large deviation from the expected background process. That is, an event that has little probability to occur given a probabilistic background model. In our case we will define a hard cut when the interframe differences have little probability to be produced by a given model of interframe differences of non-cut frames. Since we only use interframe differences, there is no need to perform motion estimation, or other type of processing, and the method turns to be very simple with low computational cost. The proposed method outperforms similar methods proposed in the literature.

1 Introduction

Shot boundary detection algorithms are one of the most basic and important methods for video analysis. They allow the segmentation of the original video sequence into basic units called shots that facilitate high level processing and abstraction of the video signal. Although it may seem a simple task, the automatic and reliable extraction of shot boundaries it has some difficulties, mainly due to the different types of video sequences, which still need to be studied. Even for simple shot transitions like hard cuts (abrupt transition between adjacent frames) there is room for improvements. In particular, one of the possible directions of research is to improve the performance of simple methods. We must remember that a video sequence contains a great amount of data, so in general we should avoid unnecessarily complicated methods. Another direction of work is the study of fully automatic methods that permit to process a wide variety of videos. In this work we will present a simple online method with only a few parameters that performs well for a representative set of testing video sequences.

We can distinguish two types of shot transitions: abrupt transitions, called hard cuts, and gradual transitions. A hard cut is an abrupt change in the frame appearance. Gradual transitions, on the other hand, span over a set of frames and are produced by postproduction effects such as fades, dissolves, morphs and wipes. In this work we will concentrate on hard cut detection.

We can divide the existing techniques for shot boundary detection into the following basic categories: pixel, histogram, block matching, object segmentation and tracking and feature tracking based methods. Some methods proposed in the literature combine some of these basic methods to attain better performances.

Pixel based methods usually compute interframe differences between frames (adjacent or not). The frame difference can be computed in several color spaces. The main drawback of pixel-based methods is their sensitivity to camera and object motion and noise. For this reason filtering is usually applied before computed interframne differences [5]. Regarding the measure of difference, we can make a distinction between distance based methods and thresholding ones. The former ones compute a distance between frames such as the absolute difference, while the later ones compute the number of pixels with a difference above a given threshold. Usually these methods are not very reliable and therefore are mostly used as indicators of probable shot boundaries that are the confirmed by more sophisticated methods [6].

Histogram based methods compare the histograms of a pair of frames using a suitable histogram distance [4]. In contrast to pixel based methods, histogram based methods are robust against camera and object motions since the histograms do not contain any spatial information. Unfortunately, the main critic and limitation is that frames of different shot can have similar histograms and in this way these methods will fail. In addition, like pixel-based methods, these methods are not robust against lighting changes.

Block-matching methods divide each frame into blocks and then match a given set of features of blocks (pixel colors, histograms, and so on) between frames. That is, the best match for each block in the source frame is found in the destination frame (This is the methodology applied in MPEG-like video coding techniques) and the similarity of these block is used as an indicator for shot boundary existence [4, 5].

Segmentation and object tracking are typically computational demanding. The underlying idea behind this methods is that frames within a shot have the same objects. Therefore, they used methods for object tracking and segmentation to achieve shot boundary detection.

Feature tracking methods detect shot transitions when there is an abrupt change in the number of features tracked. For example, if the frame edges have strong variations [5]. In [8] the authors propose feature tracking as a measure of frame dissimilarity. Instead of tracking edges, they propose to track fine grained features as corners and textures. Hard cuts are then detected as points with high interframe feature loss.

Nearly all of the previous methods rely on a set of thresholds in order to decide whether there is a shot boundary in a given frame. In the case of the pixel base methods we need a threshold to decide if the interframe distance is enough to declare a shot boundary. For histogram based methods the threshold is applied to the histogram distances. The problem of selection of the right threshold is a key point that has big influence in the overall system performance. Unfortunately it has received little attention in the literature [5] and most of the authors propose heuristics for their selection. Furthermore, it has been demonstrated that global thresholds led to sub optimal methods, with too many false positives or false negatives [5]. To solve this problem adaptive thresholds have been proposed. However, life is never so straight forward, and when using adaptive thresholds we must design an updating rule based on, for example, the statistics of non-boundary frames. This introduces additional problems concerning the correct estimation of this statistical information. Traditionally the problem is solved introducing a learning stage where several video sequences are processed to obtain the desired statistics.

In this paper we introduce a simple method for the detection of hard cuts using only interframe differences. The method is inspired in the works of Computational Gestalt [2, 3]. The key idea in this framework is to define the meaningful event as large deviation from from the expected background process. That is, an event that has little probability to occur given a probabilistic background model. In our case we will define a hard cut when the interframe differences have little probability to be produced by a given model of interframe differences of non-cut frames. Since we only use interframe differences, there is no need to perform motion estimation, or other type of processing, and the methods turns to be very simple with low computational cost.

In the first step of the algorithm we compute a measure of hard cut probability, or meaningfulness. Then in a second stage we apply an adaptive thresholding technique that only uses the information of the video sequence being processed to find the hard cuts. This contrasts with other methods that need a supervised learning step to obtain the thresholds. This makes our methods very simple and fast.

Since we will use only interframe differences for adjacent frames we assume that the videos are contain mainly smooth transitions. From another point of view, we assume a reasonable temporal sampling. As we said above these methods have problems with strong camera or object motions. If a strong motion, or a lightning change occurs, the method may produce a false positive. Even though these restrictions, we will show that the results of the proposed method are very robust and perform well for a wide variety of videos.

2 Proposed Method

Lest suppose we have the probability, $P_\mu = P(e(x) > \mu)$, that the error, $e(x) = |I(x;t) - I(x;t-1)|$ at pixel x, exceeds the threshold μ. Within a video shot segment we expect the frame differences to be small and therefore there would be a small chance for a big number of pixels exceeding a reasonable threshold. Below we will address the threshold selection. If we fix the threshold μ we can compute the error image and the number of pixels, N_μ, exceeding the threshold μ. In order to assess the meaningfulness of this event we must compute its probability of occurrence given the apriori information of interframe differences, P_μ. This can be done computing the probability of at least N_μ pixels exceeding the threshold

μ by using the Binomial distribution:

$$B(N, N_\mu, P_\mu) = \sum_{k=N_\mu}^{N} C_k^N P_\mu^k (1 - P_\mu)^{N-k}$$

Using this probability, and following the ideas of the computational gestalt theory, we say that the previous event is meaningful if its probability is very low given the statistics of past frame differences [3]. This means that we say that the event is meaningful if it is a large deviation of what is expected given past information.

Abrupt changes in interframe differences can be produced by hard cuts, fast motion and deformation, but also by slow motions, freezing or frame repetition. Therefore, we must also detect these events. Applying the same idea, given a threshold λ and the probability $P_\lambda = P(e(x) \leq \lambda) = 1 - P(e(x) > \lambda)$, we compute the probability of at least N_λ pixels being below the threshold.

$$B(N, N_\mu, P_\mu) = \sum_{k=N_\lambda}^{N} C_k^N P_\lambda^k (1 - P_\lambda)^{N-k}$$

So far we have presented the basic method for the assessment of the meaningfulness of the events abrupt change and slow change. Now we are going to explain the selection of the thresholds, the combination of the previous measurements for the detection of hard cuts, and the estimation of the probabilities P_μ and P_λ.

The meaningfulness of each of the events is obtained as the minimal probability over a set of fixed thresholds. That is, the meaningfulness of the event abrupt change is obtained as:

$$M_a = \min_{\mu_i} B(N, N_{\mu_i}, P_{\mu_i})$$

where each term corresponds to a threshold $\mu_i \in \{\mu_1, ..., \mu_n\}$. In the same way, the meaningfulness of a slow change is obtained as:

$$M_s = \min_{\lambda_i} B(N, N_{\lambda_i}, P_{\lambda_i})$$

with $\lambda_i \in \{\lambda_1, ..., \lambda_m\}$. The domain of variation of λ_i is set to detect slow changing frames, hence we set $\lambda_i \in \{1, ..., 10\}$. In the same way, since with the threshold μ_i we expect to detect abrupt changes we set $\mu_i \in \{10, ..., 100\}$. The upper limit is set to a reasonable high value and does not play an important role in the algorithm. The upper limit for λ_i and lower limit of μ_i has been set to 10 as a conservative value. We did several experiments changing these values and

[3] In the computational gestalt theory instead of working only with the probabilities the authors propose to estimate the expectation via multiplying the probability by the number of test performed [3].

we didn't encounter differences in the final results. However, it is still an open problem the tuning of it.

To conclude the description of the first step of the algorithm we now present the estimation of probabilities P_μ and P_λ. These probabilities are obtained from the error histogram of past frames. To cope with non-stationary statistics, we use a buffer, Buf, of size n of non-cut histograms and a $\alpha - \beta$ filter. The histogram of errors is updated with the following rule:

$$\begin{aligned} h_t &= \text{Histogram}(|I(x;t) - I(x;t-1)|) \\ h &= \alpha \text{mean}(Buf) + (1-\alpha)h_t \end{aligned}$$

with $\alpha = 0.9$ and $n = 12$. The value for n was chosen to hold in the buffer half second of video (assuming 24 fps).

As said before, we the previous rule we track non-cut error histogram. That means that we must have a rule to decide whether a frame is hard cut or not. To do so we use the measure $H = M_a/M_s$. If $H < 1$ the probability of occurrence of an abrupt change given the previous non-cut probability distributions is smaller, more meaningful, than the occurrence of an slow change.

Algorithm

For all frames $t \geq 2$:

1. Compute interframe differences:

$$e(x) = |I(x;t) - I(x;t-1)|$$

2. Find the meaningfulness of the events abrupt and slow change:

$$M_a = \min_{\mu_i} B(N_{\mu_i}, N, P_{\mu_i})$$

$$M_s = \min_{\lambda_i} B(N_{\lambda_i}, N, P_{\lambda_i})$$

The probabilities P_{μ_i} and P_{λ_i} are computed using the histogram h [4].

3. If $M_a < M_s$ (there is a probable hard cut), do not introduce the histogram of $e(x;t)$ into the buffer, else, update the histogram with:

$$\begin{aligned} h_t &= \text{Histogram}(|I(x;t) - I(x;t-1)|) \\ h &= \alpha \text{mean}(Buf) + (1-\alpha)h_t \end{aligned}$$

and introduce h_t in the buffer.

For the computation of the binomial distributions we use the Hoeffding approximations [1] to obtain an upper bound for the logarithm of M_a and M_s using:

$$\log(B(k,n,p)) \leq k \log(pn/k) + n(1-k/n) \log\left(\frac{1-p}{1-k/n}\right) \text{ for } k/n \geq p$$

[4] Initially h is computed using first and second frames.

Since both, M_a and M_s, can attain extremely small values is numerically impossible to work directly with them. For this reason we compute their logarithms and use $\log(H) = \log(M_a) - \log(M_s)$ in our method.

As we said in the introduction we propose an online method, therefore, we must decide the occurrence of a hard cut using only past values. In fact we introduce a delay in the system response in order to consider while judging frame t also the results from frames $t+1, ..., t+4$. In the second step of processing we consider a window, $W = [t-4, ..t+4]$ centered in t. We will sat that there is a hard cut at frame t if the following conditions are fulfilled:

$$\log(H)(t) = \min_{s \in W} \log(H)(s) \tag{1}$$

$$\log(H)(t) < \min_{s \in \{t-4,..,t-1\}} 4 \log(H)(s) \text{ or } \log(H)(t) < \min_{s \in \{t+1,..,t+4\}} 4 \log(H)(s) \tag{2}$$

$$\log(H)(t) < \text{Threshold}(t) \tag{3}$$

where Threshold(t) is an adaptive threshold that is computed using only the accumulated values of $\log(H)$ for non-cuts X [5]:

$$\text{Threshold}(t) = \text{mean}(X) - 5 * \text{std}(X)$$

This is a simple method of template matching to obtain only prominent peaks. We must mention that we are assuming that hard cuts are separated at least four frames (As we will see in next section some video sequences do not fulfill this hypothesis).

For processing color video sequences we apply the previous method by adding up the meaningfulness $\log(H)$ for the three color channels. In this work we use the YUV color space.

3 Results and Evaluation

We are going to test our algorithm against a set of videos used in [8]. In figures 3 and 3 we show the first frame of each video together with $\log(H)$. As we can see there are set of well defined peaks that correspond to the hard cuts. In table 3 we present the results for all the sequences together with the numerical results obtained in [8]. As in [8] we measure the performance of out method using precision (Prec), recall (Rec) and F1 defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F1} = \frac{2 \times \text{Precision } \times \text{ Recall}}{\text{Precision} + \text{Recall}}$$

The proposed method outperforms on average the precision the feature tracking method and the pixel based one, while performs worse than the histogram

	Proposed Method			Feature tracking [8]			Pixel based [8]			Histogram based [7]		
Seq	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
A	1	1	1	1	1	1	1	1	1	1	1	1
B	.800	1	.889	1	1	1	.825	.825	.825	1	.375	.545
C	.941	.906	.923	.595	.870	.707	.764	.778	.771	.936	.536	.682
D	1	1	1	1	1	1	1	1	1	1	.941	.969
E	1	.840	.913	.938	1	.968	.867	.867	.867	.955	.700	.808
F	1	1	1	1	1	1	0	0	0	1	1	1
G	.882	.938	.909	.810	.944	.872	.708	.994	.809	1	.666	.800
H	.760	.950	.844	.895	.895	.895	.927	1	.962	.971	.895	.932
I	1	1	1	1	1	1	1	1	1	1	.500	.667
Average	.932	.959	.942	.915	.968	.938	.788	.829	.804	.985	.735	.823
Variance	.009	.003	.004	.019	.003	.010	.099	.104	.099	.001	.055	.027
Std dev	.095	.057	.059	.137	.052	.100	.314	.323	.315	.025	.234	.165

Table 1. Results obtained for sequences in figures 3 and 3

based one. It has similar recall capabilities than the feature tracking based method. From these number we can conclude that the proposed methods has less false positives than the other three reported methods while achieving similar number of false negatives with respect with the feature tracking method. Summing up the F1 measure is the best among the four methods tested.

Looking at the individual sequences, the proposed method outperforms the feature tracking method three cases while loses precession in two cases (B and H). This is mainly due to strong motions that are not satisfactory resolved in the proposed method. Also, in the case of sequence C, it contains very close hard cuts that are missed due to our restriction of cuts separated in time at least four frames. This sequence has a poor temporal sampling rate On the other hand the proposed method has always better recall perform that then feature tracking one.

Finally, at the bottom of the table 1 we present the average, variance and standard deviation of the results to show that the results are stable.

To show the advantages of the proposed method against other well-known interframe difference methods we are going to compare the output of our method against the output of standard frame difference in the YUV space. For the comparison we normalize both results dividing each one by the maximum difference. The results are presented in figure 3 for videos A (Lisa) and B (Jamie). As we can see the results are less noisy and the peaks at hard cut positions are clearly separated from non-cut ones. This contrast with results obtained with traditional frame difference methods. However, we can also see, especially for the results on Jamie sequence, that the peaks have strong variations. Nevertheless, from this plots we can conclude that an offline hard cut detection would be much easier using $\log(H)$ than the traditional pixel differences as the hard cut peaks are clearly separated from the non-cut ones.

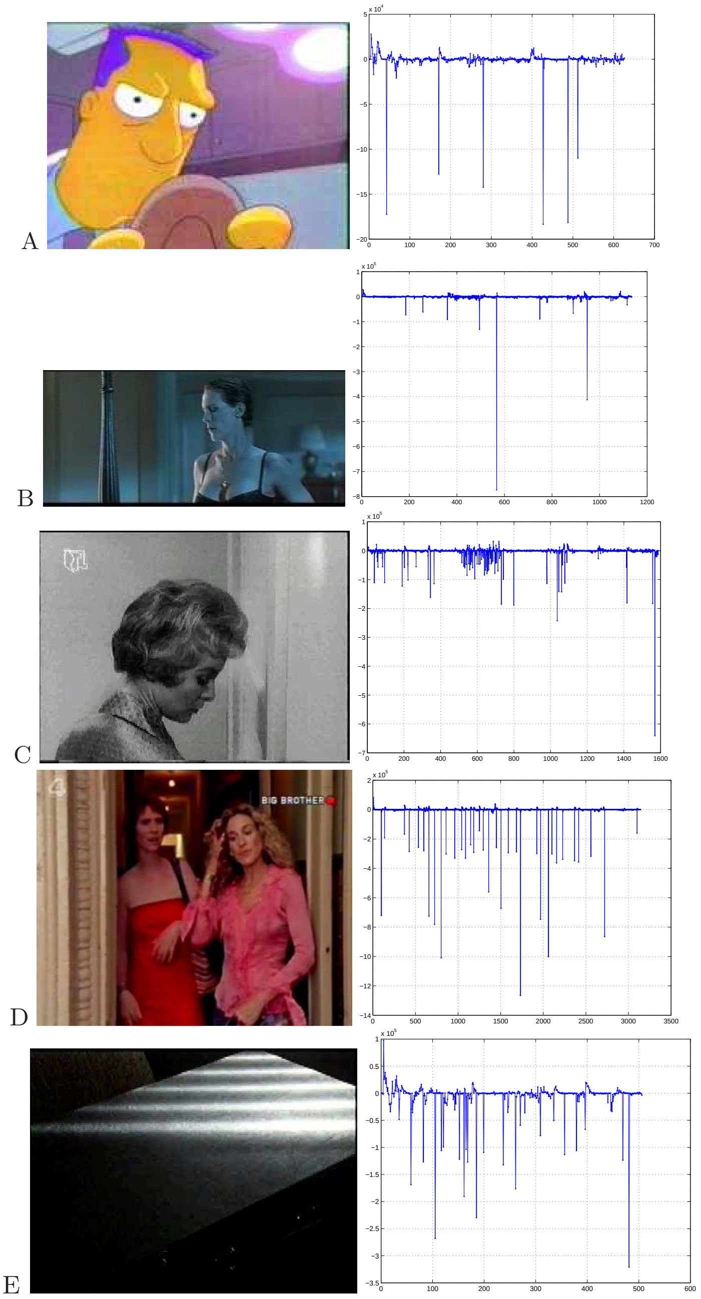

Fig. 1. Left: First frame from the sequence. Right: $\log(H)$ for the sequence. A(Lisa): Cartoon video with substantial object motion. B(Jamie): Strong motions. C(Psycho): Black and white movie with substantial action and motions and many close hard cuts. D(Sex in the city): High quality digitalization TV show. E(Highlander): Low quality digitalization of TV show

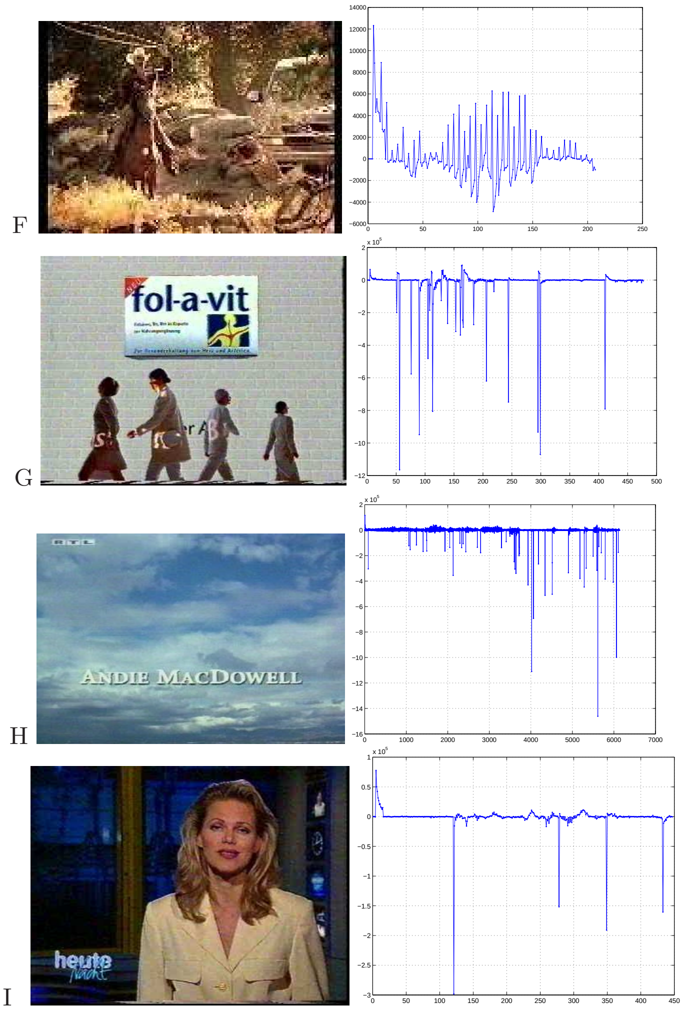

Fig. 2. Left: First frame from the sequence. Right: $\log(H)$ for the sequence. F(Commercial2): Contains no cuts but it has a low of postproductions effects that can be misclassified as cuts. G (Comemrcial1): Commercial sequence. H(Video): Its contains passages of strong motions. I (News): TV news.

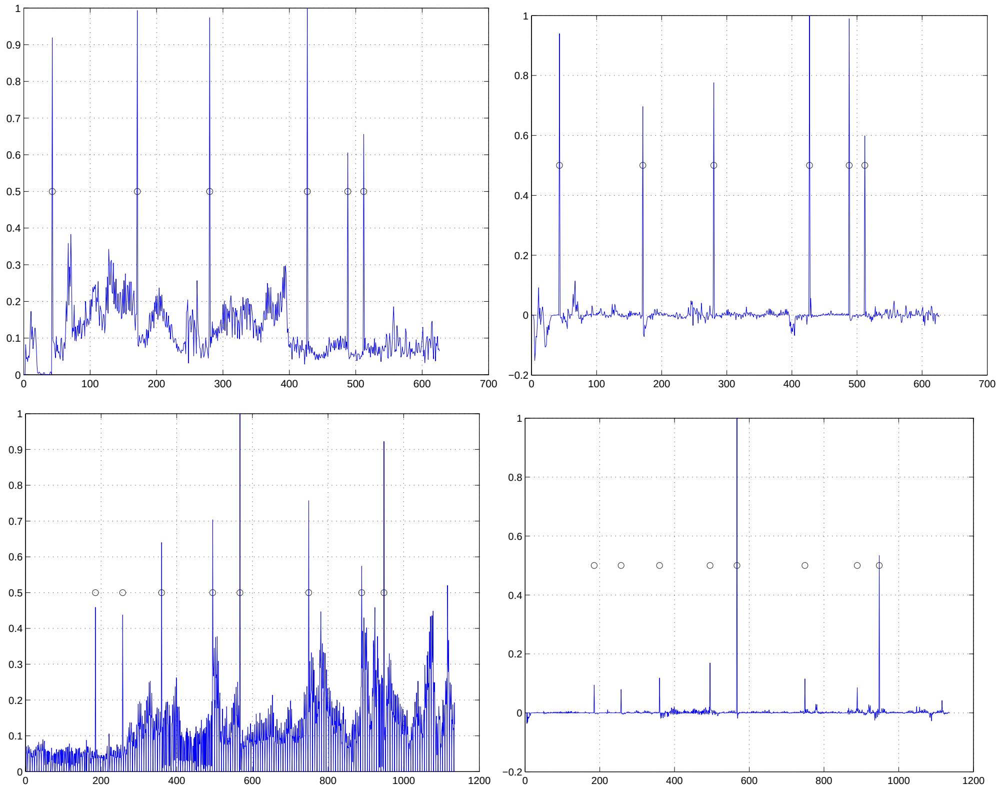

Fig. 3. Left: Results of a tradition pixel base difference method. Right: Results of the proposed algorithm. The black dots indicate the true hard cuts. Top: Results for Lisa sequences. Bottom: Results for Jamie sequence.

4 Conclusions and Future Work

We have presented a simple method that uses only interframe differences that improves the results of previously reported methods. The method obtains a measure for hard cut meaningfulness with clear peaks at hard cut positions. This allows for simpler adaptive threshold and offline detection methods.

We formulated the problem inspired in the computational gestalt theory and presented a novel method to compute hard cuts based on simple interframe differences. We believe this direction of work can provide better results and particularly more formal methods with less heuristics behind them.

In future work we will address the limitation of the method with respect to strong motions and lightning changes, and also we will try to obtain bounds on M_a and M_s to improve the adaptive thresholding technique. This will be important to normalize the peaks in the response ($\log(H)$).

References

1. A. Desolneux. *Evnements significatifs et applications l'analyse d'images.* PhD thesis, ENS-Cachan, France, 2000.
2. A. Desolneux, L. Moisan, and J.-M-Morel. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–512, April 2003.
3. A. Desolneux, L. Moisan, and J.-M-Morel. Maximal meaningful events and applications to image analysis. *The Annals of Statistics*, 31(6):1822–1851, December 2003.
4. Ullas Gargi, Rangachar Kasturi, and Susan H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 2000.
5. Alan Hanjalic. Shot-boundary detection: Unraveled and resolved. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002.
6. Chung-Lin Huang and Bing Yao Liao. A robust scene-change detection method for video segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001.
7. S. Pfeiffer, R. Leinhart, G. Kuhne, and W. Effelserberg. The MoCa Project - Movie Content Analysis REsearch at the University of Mannheim.
8. A. Whitehead, P. Bose, and R. Laganiere. Feature based cut detection with automatic threshold selection. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 410–418, 2004.