Shot Detection for Formula 1 Video Digital Libraries

R. Cucchiara, C. Grana, G. Tardini

Department of Information Engineering, University of Modena and Reggio Emilia,
Via Vignolese 905/b, 41100 Modena, Italy, {surname.name}@unimore.it

Abstract. Metadata extraction is one of the first tasks to be performed for automatic Digital Library annotation, and in particular shot detection has been widely explored in literature. While a lot of methods have been proposed for the detection of abrupt cuts, only a small number of them has explicitly addressed the problem of gradual transitions. In this paper we propose an algorithm that exploits a precise model of linear transition. Experimental results on Formula 1 car races videos show the robustness of this method. These test videos are characterized by extreme situations such as fast camera and objects motion and very different kinds of shots. The algorithm is able to estimate the exact length of the transition and an error score is also given as a fitness measure to the linear model, to discriminate true transitions from false detections. The final shot segmentation is delivered as an MPEG7 compliant output.

1 Introduction

The widely growing plethora of systems concerning Universal Multimedia Access aims at defining effective tools for annotation, indexing, information retrieval, adaptation and high performance remote access to large digital libraries. Whenever the Video Digital Library is enriched with editing functions, such as most sport events, news, TV programs, movies and so on, the most common starting step for automatic annotation is shot detection, in order to provide homogeneous sequences of frames that can be used for further tasks. Moreover, for content-based adaptation applications it is necessary to provide the elementary parts that must be further processed for video abstraction (such as key-frame extraction), summarization or, in general, semantic transcoding, both for off line downloading and streaming.

In this paper, we address the problem of automatic annotation of video digital libraries characterized by very high motion, large feature variations and a strong presence of video editing processes (such as cuts, dissolves, wipes, etc.), as it is typical of TV programs of sports competition with high dynamics. In particular, we are addressing the case of *Formula 1 videos* inside a more general system which should also deal with large videos in different storage formats, provide shot detection and indexing, video abstraction and summarization to allow for quick positioning and manual searching and provide shot classification and automatic retrieval.

The shot detection approach, in particular, should be precise in locating transitions, but also fast enough to provide scene selection as soon as possible after the race. For these reasons we defined a combined approach, composed of two steps: the first one selects a set of possible transitions evaluating the color variation ratio in successive frames, while the second one searches for linear transitions around the position suggested by the first one. While the first approach allows for a quick solution, able to miss very few transitions, the second one gets rid of false detected ones giving also an estimate on their length, possibly useful information for later classification tasks.

2 Previous Work

In recent years many techniques have been proposed for abrupt transitions (hence called *cut*) detection, and they have proved to give highly satisfactory results. The most common ones exploit differences of some metrics between adjacent frames [1,2,3,4]. Other research activity has addressed the problem of shot detection in the compressed domain [5,6,7,8]. In these works the only information extracted from the videos are the one directly available from the MPEG streams, that is DCT coefficients, motion vectors and directions of prediction for each block. The absence of the decoding process allows a much faster computation, but it has the drawback of a lower reliability. While this kind of approach has achieved results comparable with the uncompressed domain techniques in cut detection, comparative studies have demonstrated that they perform much worse on gradual transitions [9].

The main problem in *gradual transitions* detection is that comparisons based on adjacent frames is not suitable because changes are too small. On the other hand, the use a larger window (i.e. difference between a frame and the k-th following one) implies an increase of difference values in the same shot, and then makes the discrimination task harder. This is especially true in presence of strong motion scenes. Therefore, latest research on video segmentation has mainly focused on this specific issue, trying to cope with both linear and non linear transitions of whatever length.

In [10] a linear transition model is exploited, but the author doesn’t deal with the choice of the length of the window. The algorithm tries to find a “plateau” in the difference values extracted with a fixed-length window. A more refined approach is proposed in [11], where authors deal with long transitions. A change indicator is derived for each frame and this indicator is claimed to be a ramp during transition and constant in the same shot. They heuristically estimate the slope of the ramp and the standard deviation at the border to find transitions boundaries.

In [12] the author expose a comparative study of most of the metrics used in shot detection approaches, both in compressed and uncompressed domain. He then proposes an algorithm to detect both abrupt and gradual transitions. This system too has a good degree of generality handling all types of transition. The transition is detected using some empirical rules, e.g. the maximum value of the window should be located in the center, and a value which represents the peak with respect to the median of the window. The accuracy of this algorithm is strictly dependent on the length of the window. Thus, if we want to detect transitions of any length, a decision space must be generated for each length. Instead of a fixed-length sliding window, in [13] an increasing length window is used, with a fixed extreme called “seed”. When the window is long enough, difference values are uncorrelated, and a non decreasing ramp in values indicates a transition. A low pass filter is also exploited to detect and remove uncorrelated random noise.

The authors of [14] have developed a learning classifier (a neural network) to detect transitions. The classifier is trained with a dissolve synthetizer which creates artificial dissolves. The classifier detects possible dissolves at multiple temporal scales, and merges the results with a winner-take-all strategy. The algorithm works on contrast-based features, as well as color-based features, and has given good result compared to standard approaches based on edge change ratio. A learning process is required also in [15], where a probabilistic based algorithm is proposed to detect both abrupt and gradual transitions. After obtaining a priori likelihood functions by experiments, they take into account all the relevant knowledge to shot boundary detection, like shot-length distribution and visual discontinuity patterns at shot boundaries.

While these works address gradual transitions of any type, in [16], the authors tried to give an explicit model for a linear transition. The hypothesis exploited here is that in an ergodic process the mean and the variance during a transition have a linear and a quadratic behavior respectively. Therefore the criterion used to determine the presence of a transition is that the ratio of the second derivative of the variance curve to the first derivative of the mean curve should be a constant.

Similarly, our approach is strictly addressed on gradual transitions with a linear behavior, including abrupt transitions. A precise model is exploited allowing achieving more discriminative power than general techniques. We developed an iterative algorithm that, given a frame of possible transition, alternatively tries to find to best center position for the transition and the best length, by minimizing an error function, which measures the fitness of data to the linear model. The error value is then used to distinguish true transitions from camera motion.

3 Basic algorithm

The basic shot detection algorithm relies on a simple distance metric between images: the maximum of the difference between the average values of the three RGB channels of two consecutive frames:

$$D_t = \max_{c \in \{R,G,B\}} \frac{1}{N} \left| \sum_{x \in I_t} c(x) - \sum_{x \in I_{t-1}} c(x) \right| \tag{1}$$

where I_t is the image frame at time t and N is the number of pixels of the image. This metric gives a rough similarity metric between current and previous frame and the base concept is that if we observe a significantly high value this is due to strong changes between the images. To make the decision whether we are observing a scene change we test the *color variation ratio,* defined as follows:

$$R_t = \frac{D_t}{\min\limits_i \sum\limits_{k=t-w}^{t-1} \left| D_j - D_k \right|} \tag{2}$$

If R_t is higher than a defined threshold, that is if the distance between current and previous frames is much higher than the median of the distance on a window of *w* frames before the current one, frame *t* is considered as belonging to a different shot. The assumption made is that we expect the values of a transition to present larger distance values with respect to the previous scene. By setting the threshold value to detect the transitions on difficult video sequences (i.e. where significant motion is present right before or after the transition) allows us to detect changes also in more static situations, since also gradual transitions will present a difference many times higher than the low values of the scene.

As in many other systems, we have a parameter that allows to chose the specificity (or recall ratio) of this algorithm. Because of its function of providing possible candidate transitions to the second stage, it is necessary to set it up so to avoid missing any transition, without of course selecting all available frames. This parameter will be chosen based on detection and speed criteria.

4 Linear Transition Detection

From the results of the basic algorithm the refinement step can start without relying on the same metric. This is done to ensure that different aspects of the scene are taken into consideration. Moreover, since we have to apply this second technique on much fewer frames we can employ a more complex measure.

Often a combination of metrics is needed to achieve good results in shot detection: as in [3] we use a histogram-based metric and a pixel-based one. While the latter is more sensitive to changes in the image but perform badly in high motion sequence, the first is insensitive to motion but also less sensitive to small changes (e.g. the same object viewed from different angle).

In this article we use as histogram metric the χ^2 test, which has been proved to perform better than other measures [13]:

$$d_{\chi^2}\left(I_a, I_b\right) = \sum_{c \in \{R,G,B\}} \sum_{j=1}^{N} \frac{\left(H_{c_a}(j) - H_{c_b}(j)\right)^2}{\max\left(H_{c_a}(j), H_{c_b}(j)\right)} \tag{3}$$

where $H_{c_a}\left(j\right)$ is the *j*-th bin value of the histogram of color plan *c* of the image I_a, and N is the number of bins. We then use the sum of squared differences as a pixel-based metric:

$$d_\delta\left(I_a, I_b\right) = \sum_{(x,y) \in I_{a,b}} \left[I_a\left(x,y\right) - I_b\left(x,y\right)\right]^2 \tag{4}$$

and we combine the two metrics in a single *difference metric:*

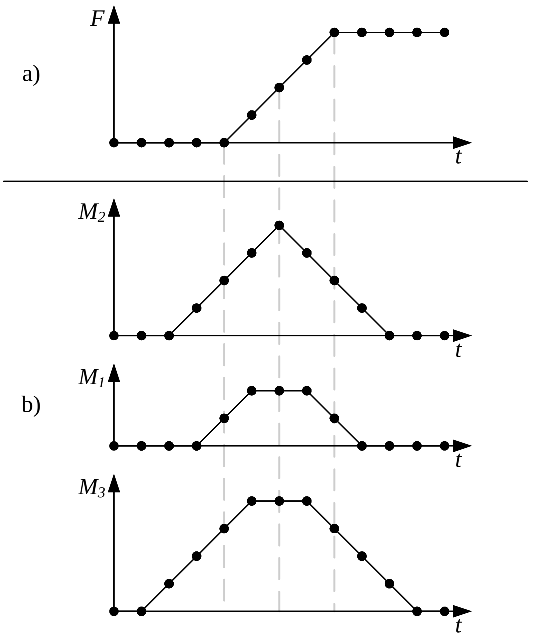

Fig. 1. Feature variation and distance measures obtained with different window sizes.

$$M\left(I_a, I_b\right) = c_{\chi^2} d_{\chi^2} + c_\delta d_\delta \tag{5}$$

The distance analysis is performed on a block of frames centered on the reference position provided by the base algorithm. The block size is set as large as needed, to include at least the largest transition of the video.

One of the parameter that we wish to estimate is the transition length, but it is possible to observe that the behavior of the distance metric is strongly dependent on the time distance between the frames that are compared. We define to this aim a window centered on a frame *t*, and compare the values of frames at $t \pm w$ under the hypothesis of an ideal situation, in which the scene before and after the transition are composed of static images, so that a constant but different feature value *F(t)* is obtained. If a *w* value equal to half of the transition length is used (so that the length is equal to *2w+1*), in case of a linear dissolve, a triangular shaped variation of the distance measure would be obtained. For example, a transition of 5 frames is depicted in in Fig. 1a, while in Fig. 1b the difference metric M_w with *w*=2, *w*=1 and *w*=3 is plotted. We can see that, if a smaller or larger window size is selected (always with respect to the real transition variation), the distance measure shape changes to trapezoidal, staying centered on the middle of the transition (Fig. 1). In case of smaller window, the trapezoid height lowers, instead, the sides' tilt doesn't change, since it only depends on the transition shape and length. For all these reasons, the refinement algorithm is constructed of two iteratively repeated steps: the first one searches for the center position assuming a fixed window size, which is then optimized by the second step, without changing the center position found.

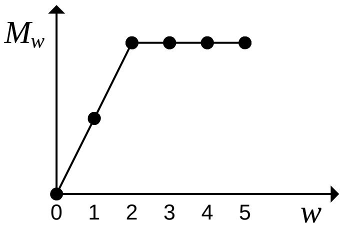

Fig. 2. Distance value at changing window sizes, with fixed center.

The search for the center position begins with a small window $\tilde{w}$, and in this step, the "best" trapezoid is searched centered on all frames, trying different lengths for the lower base. In fact the tilt of the sides is constantly computed from the size of the window and from the current center position. For each couple of positions and minor base the following measure is computed:

$$F_t^\delta = \sum_{i=t-\delta}^{t+\delta} \min\left(M_{\tilde{w}}(i), \psi_{\tilde{w}}^t(i)\right) - \sum_{i=t-\delta}^{t+\delta} \left|M_{\tilde{w}}(i) - \psi_{\tilde{w}}^t\right| \tag{6}$$

$M_{\tilde{w}}(i)$ is the difference metric between frames $i+\tilde{w}$ and $i-\tilde{w}$, $\psi_{\tilde{w}}^t(i)$ is the value of current trapezoid at point *i* and δ is the current window size $\tilde{w}$ plus the half minor base. In the fitting measure equation (6) two components are evident: the first one tends to maximize the area under the trapezoid, while the second component describes the correspondence of our linear hypothesis with the data. It is very important to include both components, since we expect the distance measure to give a trapezoidal shape (the second term in Eq. 6), but we also request its relevance, i.e. the amount of difference between the left and right scenes, to be significant.

After finding the trapezoid which maximizes F_t^δ, we found a candidate transition center $\tilde{t}$. The second step of the algorithm gives an estimate of the transition length considering the variation of the difference between frames at growing distances with respect to the transition center $\tilde{t}$. In Fig. 2 the difference values between frames $\tilde{t}-w$ and $\tilde{t}+w$ are plotted. We can see that in the ideal case the difference linearly grows up to the transition size and successively it is stable, leading to an horizontal straight line in the graph. This observation holds without regard to window size we chose to use in the first step of the algorithm, so we try to estimate the change in tilt of the distance graph by optimizing the function

$$Z_{\tilde{t}}^w = \sum_{i=0}^{w} \left|M_i(\tilde{t}) - \frac{M_w(\tilde{t})}{w} i\right| + \sum_{i=w+1}^{W} \left|M_i(\tilde{t}) - M_w(\tilde{t})\right|, \tag{7}$$

where W is the maximum size that a transition can assume. The w value that minimizes $Z_{\tilde{t}}^w$ is then used for the next iteration of the refinement procedure. In simple cases the algorithm progressively narrows the trapezoid minor base leading to the expected triangular shape. Unfortunately this narrowing is not always quick enough and this can lead to unacceptable times. For this reason, at each iteration the trapezoid minor base length is requested to become smaller than before, giving a deterministic maximum convergence time.

Given the length $\tilde{w}$ of the transition and its center $\tilde{t}$, as detected by the algorithm, we must compare the obtained real data with the linear transition model. This model is described by a triangle T_w^t centered in t and defined by the three points:

$$\begin{aligned} &T_w(t-w) = M_w(t-w) \\ &T_w(t+w) = M_w(t+w) \\ &T_w(t) = M_w(t) \end{aligned} \tag{8}$$

In case of transition between two frames with no motion at all, we should obtain $T_w(t-w) = T_w(t+w)$.

We define the error measure as

$$e_w^t = \frac{1}{2w} \sum_{i=-w}^{w} \left| M_w(i+t) - T_w^t(i) \right| \tag{9}$$

which measures the distance of real data from the triangle model, for the given position and length. The error sum is divided by the triangle's base to obtain a measure independent from the transition length.

We then define the ratio

$$r_w^t = \frac{Peak_w^t}{e_w^t} \tag{10}$$

where

$$Peak_w^t = M_w(t) - \min\left(T_w^t(t-w), T_w^t(t+w)\right). \tag{11}$$

The Peak value measures the height of the center value with respect to the lowest extreme of the triangle providing information on the transition significance, while the ratio provides a normalized insight on the similarity of the sequence to a linear change. These two values are employed to discriminate true and false transitions.

Finally, a not negligible improvement (about 1-2% in precision and recall) in video with high motion and pan movements adding to the error expression a marginal error component, as in the following equations:

$$me_w^t = \frac{\sum_{i=-w-\lambda}^{-w-1} \left(M_w(i+t) - T_w^t(-w+t) \right) + \sum_{i=w+1}^{w+\lambda} \left(M_w(i+t) - T_w^t(w+t) \right)}{2w}. \tag{12}$$

The error component measures the constancy of difference values at the borders of the transitions, defined by a window of arbitrary size λ (in our experiments $\lambda = 10$ frames). Typically a transition will have more stable values than a panning sequence.

The exposed approach is tied to employ windows, and thus transitions, with an odd number of frames. To allow also even dimensions and managing cuts with slightly more tolerance, we interpolated the distance value between consecutive frames, enabling an half frame positioning of each transition.

With this algorithm we are able to complete the description of the transitions hinted by the base shot detection technique providing transition length, refining the center position and describing the quality of fit with the real data.

5 Results

Performance evaluation of shot detection can be provided at frame or at event level, that is we can test whether we can identify every frame as belonging to a transition or not, or we can see if we shall consider a transition as correctly found. The main rule we chose is, implicitly, the assumption made by many works in literature [9]: an intersection is made between the transition frames as detected by the algorithm and by a human operator (ground truth). True positives (TP) are the transitions in the ground truth that present at least one frame in the intersection, false negatives (FN) those which do not have any. We can also measure false positives, by counting all the transitions that our algorithm detected, but which didn't have any frame in the intersection. This comparison was performed at event level to provide a hint on the user satisfaction, since losing a transition would be certainly worse than missing many transition frames, but still differentiating shots.

As usual in literature, since we did not have the availability of true negatives (TN), we do not use a *Specificity* measure, but the concept of *Precision*, together with the measure of *Recall* (analogue to *Sensitivity*):

$$Recall = \frac{TP}{TP + FN}, Precision = \frac{TP}{TP + FP} \tag{13}$$

During the algorithm development, we used a series of Formula 1 selected sequences as training examples for the choice of thresholds and as well for the histogram and spatial metrics linear combination coefficients. Later we tested the algorithm on a single complete F1 race video, composed of about 125000 frames. Here, the base algorithm selected 1754 possible transitions, including 175 real linear transitions and 539 cuts.

In the analysis we excluded 16 non linear transitions, namely special editing effects, since these are too far from the linear model to produce meaningful results.

We give separate results for gradual transitions and cut: precision and recall values in function of the threshold on (10) are plotted in Fig. 3. As it can be seen from the figure, the recall value never reaches 100%: this is due to an amount of transitions and cuts that are discarded by the minimum threshold on the peak value (11). In other

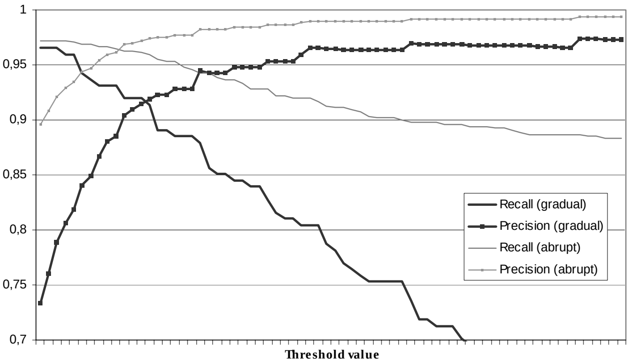

Fig. 3. Recall and precision plots

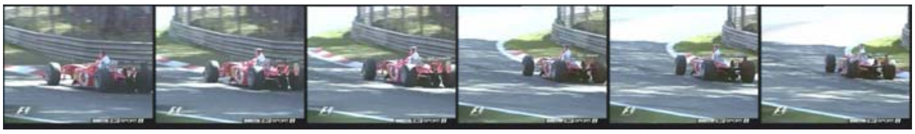

Fig. 4. False positive detection, caused by quick camera panning motion.

words, the scenes are too similar to each other to be detected as different with the employed metric.

The main cause for false positives is camera panning: if the start and end position of the pan are different enough and the camera motion is constant between the two points, this can be seen as a linear transition. The reason is that we used metrics that take into account only global changes in the frame. In real transitions, two different zones of the frame should show the same behavior, which is not true in case of motion. In Fig. 4, it is possible to see an example of a false detected transition. The panning of the camera and the subsequent significant change of color and lightness shows a linear behavior that is erroneously interpreted as a transition. In Fig. 5, the difference values for two scenes are reported: the triangular shape is clearly visible. In Fig. 6 an example of possible miss detection is depicted. The fast camera movement on the right side of the transition destroys the triangular shape of the differences, increasing the error. The detection of such a case depends on the threshold on the peak/error ratio: if we move it in a lower than the optimal point, most of these situations can be correctly identified, since at least one side of the triangle is visible.

6 Conclusions

A novel linear transition detector that was implemented in order to refine scene changes detection in high-motion videos was presented. The detection scheme integrates indications from a first basic algorithm with the results of a two step optimization process on different feature spaces. Transition missed at the first step cannot be recovered in the second stage so a conservative setting for the first algorithm must be ensured. The method was tested on Formula 1 car races videos and results are very promising. For this specific test additional work can be done, using domain specific knowledge, in order to secure higher detection results and their integration into a broader scene classification and video management framework.

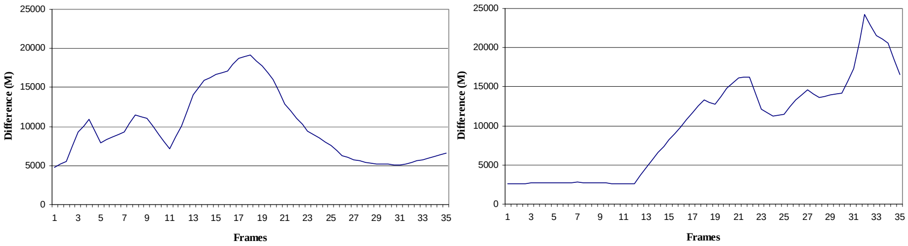

Fig. 5. Differences of the panning and miss detection sequence.

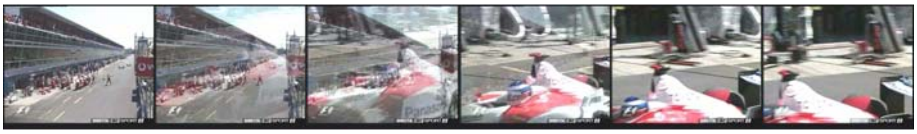

Fig. 6. Possible miss detection due to high camera motion in proximity of the shot change.

Acknowledgements

The project is funded by the European Network of Excellence DELOS of the VI Framework Program. We also would like to thank Ferrari S.p.A. for kindly providing the video database.

References

1. Gong, Y.: An Accurate and Robust Method for Detecting Video Shot Boundaries. IEEE Int. Conf. Multimedia Comput. Syst. **1** (1999) 850-854
2. Kumar, R., Devatha, V.: Statistical approach to robust video temporal segmentation. ICVGIP (2002) 91-96
3. Naphade, M.R., Mehrotra, R., Ferman, A.M., et al.: A high-performance shot boundary detection algorithm using multiple cues. Int. Conf. Image Process. **1** (1998) 884-887
4. Zhang, D., Qi, W., Zhang, H.J.: A New Shot Boundary Detection Algorithm. IEEE Pacific Rim Conf. Multimedia (2001) 63-70
5. Gamaz, N., Huang, X., Panchanathan, S.: Scene change detection in MPEG domain. IEEE Southwest Sympos. Image Anal. Interpr. (1998) 12-17
6. Pei, S.C., Chou, Y.Z.: Efficient MPEG Compressed Video Analysis Using Macroblock Type Information. IEEE Trans. Multimedia **1** (1999) 321-333
7. Sugano, M., Isaksson, R., Nakajima, Y., Yanagihara, H.: Shot genre classification using compressed audio-visual features. IEEE Int. Conf. Image Process. **2** (2003) 17-20
8. Taskiran, C., Delp, E. J.: Video scene change detection using the generalized sequence trace. IEEE Int. Conf. Acoust. Speech Signal Process. (1998) 2961-2964
9. Gargi, U., Kasturi, R., Strayer, S.H.: Performance Characterization of Video-Shot-Change Detection Methods. IEEE Trans. Circuits Syst. Video Technol. **10** (2000) 1-13
10. Yeo, B.L., Liu, B.: Rapid Scene Analysis on Compressed Video. IEEE Trans. Circuits Syst. Video Technol. **5** (1995) 533-544
11. Heng, W.J., Ngan, K.N.: Long transition analysis for digital video sequences. Circuits Syst. Signal Process. **20** (2001) 113-141
12. Bescos, J.: Real-Time Shot Change Detection Over Online MPEG-2 Video. IEEE Trans. Circuits Syst. Video Technol. **14** (2004) 475-484
13. Huang, C.L., Liao, B.-Y.: A robust scene-change detection method for video segmentation. IEEE Trans. Circuits Syst. Video Technol. **11** (2001) 1281-1288
14. Lienhart, R., Zaccarin, A., A System for Reliable Dissolve Detection in Videos. IEEE Int. Conf. Image Process. (2001) 406-409
15. Hanjalic, A.: Shot-boundary detection: unraveled and resolved? IEEE Trans. Circuits Syst. Video Technol. **12** (2002) 90-105.
16. Fernando, W.A.C., Canagarajah, C.N., Bull, D.R.: Scene change detection algorithms for content-based video indexing and retrieval. Electron. Commun. Eng. J. **13** (2001) 117-126