Evaluating and Combining Digital Video Shot Boundary Detection Algorithms

Paul Browne, Alan F Smeaton, Noel Murphy, Noel O'Connor, Seán Marlow, Catherine Berrut

Centre for Digital Video Processing
Dublin City University
Dublin 9, Ireland
{ pbrowne, asmeaton }@compapp.dcu.ie

Keywords: Digital Video; Evaluation; Shot Cuts; MPEG-1

Abstract. The development of standards for video encoding coupled with the increased power of computing mean that content-based manipulation of digital video information is now feasible. Shots are a basic structural building block of digital video and the boundaries between shots need to be determined automatically to allow for content-based manipulation. A shot can be thought of as continuous images from one camera at a time. In this paper we examine a variety of automatic techniques for shot boundary detection that we have implemented and evaluated on a baseline of 720,000 frames (8 hours) of broadcast television. This extends our previous work on evaluating a single technique based on comparing colour histograms. A description of each of our three methods currently working is given along with how they are evaluated. It is found that although the different methods have about the same order of magnitude in terms of effectiveness, different shot boundaries are detected by the different methods. We then look at combining the three shot boundary detection methods to produce one output result and the benefits in accuracy and performance that this brought to our system. Each of the methods were changed from using a static threshold value for three unconnected methods to one using three dynamic threshold values for one connected method. In a final summing up we look at the future directions for this work.

1 Introduction

Digital video information consists of a series of 25 frames or images per second, plus an associated and synchronised audio track. The development of standards for video encoding such as the MPEG family coupled with the increased power of computing mean that content-based manipulation of digital video information is now feasible. In order to develop any content-based manipulations on digital video information, this information must first be structured and broken down into components. Shots are the basic structural building block for this and the boundaries between shots need to be determined automatically.

A shot in video information may be defined as continuous images (frames) from a single camera at a time. A shot boundary is the gap between two shots. A shot cut is a type of shot boundary where one shot abruptly changes to another shot. An example of a shot cut is where the last frame in one shot is followed by the first frame in the next. Examples of other types of shot boundary are fades (where the frames of the shot gradually change from or to black), dissolves (where the frames of the first shot are gradually morphed into the frames of the second) or wipes (where the frames of the first shot are moved gradually in a horizontal or vertical direction into the frames of the second).

The main reason why automatic shot boundary detection is difficult is the fact that any kind of shot transition can be easily confused with camera and object motion which occurs in video anyway. A shot with much object motion throughout the frame such as a sports or action shot or a clip from a music video, can cause the false recognition of a shot boundary. To further complicate matters, a camera can have a variety of movements such as panning, tilting, booming, tracking, zooming in or out, or a combination of these. All this can lead to too much visual "noise" making techniques for automatic detection too simplistic.

The reason for wanting to segment video into shots is to allow content-based operations such as browsing and searching. The Físchlár system under development in our research group [Lee *et al*., 2000] is one such example of this and there are others such as [Zhang *et al*., 1995], [Simpson-Young & Yap, 1996] and [Christel & Martin, 1998]. All of these systems depend upon an underlying shot boundary detection algorithm which works accurately in order to allow subsequent operations such as selection of representative frames, browsing, similarity searching and the creation of program transition graphs to be performed.

A variety of techniques have been proposed for shot boundary detection in digital video. Pairwise comparison checks each pixel in one frame with the corresponding pixel in the next frame [Zhang *et al*., 1995]. The Likelihood ratio approach compares blocks of pixel regions [Zhang *et al*., 1995]. Net comparison breaks the frame into base windows [Xiong *et al*., 1996]. The Colour histogram method compares the intensity or colour histograms between adjacent frames [Bouthemy *et al*., 1995]. Model-based comparison uses the video production system as a template [Hampapur *et al*., 1995]. Edge detection segmentation looks for entering and exiting edge pixels [Zabih *et al*., 1995].

The methods above work on uncompressed video, but some other approaches work on the compressed data itself. These methods utilize the compression features of MPEG for shot boundary detection. [Boon-Lock & Liu, 1995] use colour blocks in an MPEG stream to find shots. DCT-based shot boundary detection uses differences in motion encoded in an MPEG stream to find shots [Arman *et al*., 1993].

The common characteristics of all this previous work on shot boundary detection are that they have all been evaluated in isolation of each other and on small collections of video (with the exception of the 3 hours used in [Ruiloba *et al*., 1999] and 8 hours used in the early work by [Boreczky *et al*., 1996]. The main reason for this has been the general unavailability of a large video collection which has had shot boundaries marked manually. Very recently, the National Institute for Standards and Technology in the US has put together a 2 hour collection. It has started to make this available to the research community since April 2000 [NIST, 2000], but it will take some time before this or any other collection becomes a *de facto* standard for evaluation.

In our work reported here and elsewhere we have used a single, 8 hour test suite of broadcast TV video for comparing different techniques for shot boundary detection. This collection has been manually annotated to determine shot boundaries and thus acts as a ground truth against which shot boundary detection techniques can be compared. In this paper we report on the performance of three different techniques for shot boundary detection, one based on colour histogram comparison, one based on measuring the differences between edges in frames, and one based on comparing macroblocks across frames.

The structure of the remainder of this paper is as follows. In the next section we describe the collection of video we have used which is now presented in terms of programme sets rather than previous presentations which were in terms of blocks of fixed lengths (20 minutes). We then give an overview of the three techniques which we have implemented. These are described in more detail in [Smeaton *et al*., 1999]. Following that, we present a summary of the performance and effectiveness of each technique on our different program types which presents some new insights not seen before. The real contribution of the paper, however, is in the combination of these three techniques which is presented in section five, and after that we have a conclusion and pointer to further work.

2 Baseline Collection of Video Material

The video collection we use for our work was introduced in [Smeaton *et al*., 1999] and consists of eight hours of continually recorded television from RTE 1, an Irish television station, recorded in June 1998. In its original form it was broken up into 24 MPEG-1 video segments. Each 20 minute segment of video is PAL with a quality similar to that of a home video recording and a size of approximately 235 Mbytes giving an overall size of 5.4 Gbytes. The reason for having 24 separate segments initially was to make the corpus easier to manage and work on. Since our earlier work on this collection we have re-organised it into 13 separate programs based on content type as described in Table 1.

Program Number	Program Name	Program Length	Number of cuts	Average cuts Per min	Content
1	RTE 1pm News	15 min	253	14	Studio news program with outdoor segments
2	Home and Away	30 min	379	15	Australian soap, outdoor segments, beach
3	Cooking program	30 min	528	15	A studio based cooking program
4	Live at Three	1h 40 min	1329	12	Afternoon show, gardening, film review, chat
5	RTE News	20 min	132	9	Studio news, small bulletin segments.
6	Emmerdale	30 min	383	14	English soap, outdoor segments, farming
7	The Lyrics Board	30 min	424	18	Irish quiz, studio based, musical knowledge
8	Shortland Street	30 min	608	21	Australian soap, hospital studio location.
9	RTE Evening News	1h	464	11	Studio news, with outdoor segments, chat
10	Fair City	25 min	467	15	Irish soap, studio and outdoor segments
11	Touched by an Angel	1h	942	17	American drama, action, outdoor and studio.
12	Keeping Appearances	30 min	543	16	Comedy, studio and outdoor segments.
13	RTE 9pm News	20 min	309	17	Studio news
Total		**8h**	**6761**	**14.9 avg**	

Table 1: Content of Video Corpus

The original collection of 8 hours of digital video was manually viewed frame by frame and the type of shot cuts occurring along with a text description of the content was recorded. As our previous work on this data was based on analysis of 24 x 20 minute segments we had been unable to evaluate various shot boundary detection techniques in terms of their performance on a particular program or program type. Evaluation of a shot boundary detection method based on program content and type is important because we suspect that a given method will behave differently depending on the content of the video, for example, a studio news program in comparison with an 'Arnold Schwarzenegger' movie. One shot boundary detection method may not perform well when detecting fades or dissolves and this will show up in the method's performance on certain programs such as cooking programs or some television soaps while it may perform better at other program types such as studio broadcasts of chat shows or news. Knowing the program *type* is thus very important in analysing performance of shot boundary detection methods.

3 Shot Boundary Detection Methods Used

The three shot boundary detection algorithms we used are introduced in [Smeaton *et al*., 1999] and are briefly summarised in this section of the paper

3.1 Colour Histogram (R1)

The approach here takes each (352 * 288) frame of the video and computes a colour histogram using a total of 192 different colour bins where each bin contains the percentage of pixels from the whole frame. When this is done, the colour histogram for each frame is checked against the histogram for the one following and a similarity metric is used to compute the likeness between the two adjacent frames. When the sequence of similarity values is analysed, a shot will show a big change in the sequence where the shot boundary occurs, so by looking for these peaks in the differences, a shot boundary can be detected. The peak in the values is required to be above some minimum value before it will be detected as a shot cut [O'Toole *et al*., 1999].

As discussed earlier, video programmes will have many different types of shot boundaries, and those where the shot boundary occurs over a number of frames may not be detected by the method above. For example, a fade or dissolve occurring over a 3 second period in, say, a cookery or gardening program will span a total of 75 frames and the incremental difference between adjacent frames in this sequence will be quite small. Furthermore, it is possible that the two separate shots in such a transition may have similar colouring and hence colour histograms, anyway. Such omissions are difficult to avoid using colour histogram based segmentation, but nevertheless this method is very popular and if used correctly it is very reliable and accurate. On the downside it is slow to compute because each frame of the digital video has to be decoded and calculations run on it to extract color values.

3.2 Edge Detection (R2)

This approach looks not at the colour differences, but at the differences between the edges detected in adjacent frames. Each frame in the digital video is turned into a greyscale image and Sobel filtering applied to detect edges. The method looks for similar edges in adjacent frames to detect a shot boundary. This method is described in detail in [Smeaton *et al*., 1999].

The principle behind the edge detection approach is that it can counter problems caused by fades and dissolves and other transitions which are invariant to gradual colour changes. With edge detection, even when there are gradual transitions between shots, there should always be a pair of adjacent frames where an edge is detected in one, and not in the other and identifying an occurrence of this on a large scale locates a shot transition. Like the colour-based method above, this will require a minimum difference between adjacent frames to detect a shot cut but it has the advantage of not being fooled by a large colour change for example when we get a camera flash. In such a case the colour-based methods will register a large difference in colour and as a result it will detect a shot boundary but the edge detection method will be looking for pixel edge differences. The downside of this technique is that it needs to decode each frame and as a result may compute slowly.

3.3 Using Macroblocks (R3)

The third of our methods for shot boundary detection, unlike the others, is based on processing the encoded version of the video. One of the features of MPEG-1 encoding is that each frame is broken into a fixed number of segments called macroblocks and there are three types of these macroblocks, I-, P- and B-. The classification of the different macroblock types is done at the encoder, based on the motion estimation and efficiency of the encoding.

I-macroblocks are encoded independently of other macroblocks and are used when the segment being encoded is very different in appearance from other segments around it in the same frame, or the corresponding segment in previous or following frames. P-macroblocks do not encode the region in the frame but instead have a *motion vector* which indicates the difference between the current block being encoded and the corresponding block in the previous frame. This is refered to as motion compensation and is used in video coding where objects remain stationary or move only a short distance between adjacent frames. B-macroblocks are bi-directionally predicted blocks and use both forward and backward prediction.

Different types of video sequence lead to different uses of I-, B- and P-macroblocks and different types of shot transitions in particular tend to have particular characteristic uses of macroblocks. Our use of macroblocks in this work is to see if we can identify and use these characteristics to determine shot boundaries directly from the compressed form of the video. For example, if a frame type that would be expected to contain backward predicted blocks does not actually have any, then it could be suggested that the future picture changes dramatically and this could indicate a shot boundary. Because it operates on the encoded form directly, it is likely to be more efficient than the other methods we use.

4 Evaluation and Preliminary Results

When evaluating our shot boundary methods we compare our results with a baseline file. The baseline is a text file that contains a listing of the shot cuts, and when and where they occur. In evaluating the results of a shot boundary detection method there are a number of parameters that one should consider, the important ones being:

Ni: Number of false shot boundary detected by the method.
Nd: Number of shot boundary not detected by the method
Nt: Number of actual shot boundary in the baseline.

From the values above most of the important evaluation measures can be calculated and the ones we use on our work are:

$$\text{Recall} = \frac{Nt - Nd}{Nt} \qquad \text{Precision} = \frac{(Nt - Nd)}{((Nt - Nd) + Ni)}$$

The recall measure looks at the percentage of actual shot cuts that the method in question has detected, while the precision measure is a percentage showing how accurate the method is at detecting only baseline shot boundary. Each of our shot boundary detection methods was converted to calculate a difference measure for each frame of the digital video and these were then run on the baseline. For each method a threshold value was determined, independently, based on a series of experimental runs. The threshold is a value above which a difference value will be accepted as a shot cut.

The following table summarises the results of using a fixed, static threshold for each of the methods R1 to R3.

	R1		**R2**		**R3**	
Program	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
1	80.0	76.3	89.9	63.2	81.4	67.6
2	84.0	62.5	82.6	65.2	72.2	71.2
3	71.4	60.6	75.4	54.7	59.9	63.3
4	80.7	84.8	84.7	61.2	62.4	79.1
5	80.6	94.7	82.4	67.4	75.4	95.5
6	83.8	79.6	46.1	72.3	57.1	74.4
7	78.4	91.7	90.5	79.2	82.7	84.7
8	79.0	79.1	87.3	71.7	78.3	75.5
9	88.4	87.1	93.8	80.6	75.5	45.6
10	80.1	73.4	79.9	72.2	48.4	77.1
11	71.2	80.5	70.0	79.7	53.3	84.6
12	85.3	79.2	85.9	76.4	64.2	82.9
13	70.4	75.4	82.7	74.4	72.4	78.0
Avg	**79.5**	**78.8**	**80.9**	**70.6**	**67.9**	**75.3**

Table 2: Results of three shot boundary detection methods with a fixed threshold value.

One of the drawbacks of using a fixed threshold to determine whether a shot change has occurred or not is that it does not take into account the variable programme types we have in our collection, or that occur across any collection of different programme genres. A variation of the fixed threshold method is to use a dynamic or adaptive threshold value where the threshold used will rise and fall depending on the type of programme being analysed. A number of frames with large difference values will have a high threshold value while small difference values will have a low threshold value. We developed an approach to dynamically adjust the threshold by looking at the visual "noise" across a future window of frames and from this raising or lowering the threshold as appropriate [O'Toole *et al*., 1999]. For each of the three methods, data using this approach were computed and the results achieved are better than when using a static threshold for each of the methods but do not change the ranking among them. Table 3 shows the precision and recall values for each of the methods R1 to R3 when using a dynamic threshold.

	R1		**R2**		**R3**	
Program	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
1	90.9	74.7	94.1	63.2	92.6	67.6
2	94.9	64.4	91.7	64.4	90.1	71.2
3	83.8	62.9	83.3	55.7	87.0	63.3
4	91.6	82.9	91.2	61.8	89.2	79.1
5	86.4	91.7	90.5	65.2	90.2	95.5
6	92.1	81.7	72.6	73.4	79.8	74.4
7	91.8	89.9	95.5	80.9	94.9	84.7
8	90.2	78.6	95.2	71.2	93.3	75.5
9	92.9	85.5	97.6	81.0	96.1	45.6
10	89.6	73.7	91.7	70.9	85.2	77.1
11	89.6	79.4	83.7	78.0	59.4	84.6
12	94.1	82.9	93.2	75.1	89.6	82.9
13	86.9	77.0	90.2	71.2	89.3	78.0
Avg	**90.4**	**78.9**	**90.0**	**70.2**	**87.4**	**75.3**

Table 3: Results of three shot boundary detection methods with a dynamic threshold value.

The results of adaptive thresholding show an improvement in precision for all methods, with recall staying about the same. We performed an analysis of the actual shots being detected by the different methods and found a good number of shots detected by all 3 methods (as expected) but also some shots detected by one technique and not others, or not being detected by 1 of the 3 techniques. Figure 1 summarises our findings and these figures confirm that a shot boundary detection method will handle programs differently based on its content and will identify different correct shots. Each of the methods was run on the complete corpus and compared individually with the baseline to find the correct shot cuts. Each of the methods' results was compared with those of the other methods and results given below.

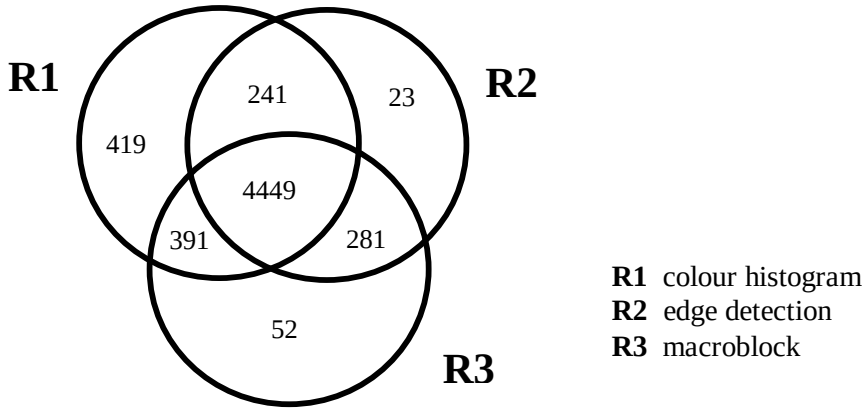

Figure 1: Overlap in correct shot boundaries detected by the methods over the complete corpus.

R2 and R3 do not uniquely discover many correct shot bounds, but collectively add another 356 correct shots to R1, so as different shot boundary detection methods have different strengths, it makes sense to use all the methods instead of relying on the results from just one method. For this reason we will look at combining the three methods into one method in the next section of this paper.

5 Combining Shot Boundary Detection Algorithms

There are a variety of ways in the literature in which data fusion, or the combination of more than one source of evidence for a decision, can be achieved. A simple form of combining *rankings* of objects is to add rank positions or similarity scores as we have done previously when combining the output of several document ranking approaches [Smeaton, 1998]. In our case here we wish to combine the similarity values between adjacent frames as measured using our three approaches in order to reach a decision as to whether there has been a shot transition between the two frames in question. In light of the results achieved by the methods working independently, and because of the fact that each worked better when an adaptive threshold was used, we decided not to directly combine the scores of the different techniques but to use a weighted Boolean logic to combine the methods. This approach will favour the results of the R1 colour histogram method, which is the best in terms of performance, but will select one of the other two if its difference value is above a certain low value.

The combined method detects a shot cut using the following logic:

Method R1 is above its adaptive threshold (T1)
Or
Method R2 is above its adaptive threshold (T2) and method R1 is above a minimum value (T4)
Or
Method R3 is above its adaptive threshold (T3) and method R1 is above a minimum value (T4)

We ran the combined technique on the full data set and the results are shown in Table 4. The method uses a dynamic threshold to select possible shot cuts for each of the methods and uses the weighted Boolean logic above to decide if the overall result is a shot cut. These results show a 4% increase in the Recall value and a 1% decrease in the Precision value compared with the R1 strategy presented earlier.

	Recall	Precision	Recall	Precision
Threshold	Static		Dynamic	
R1	78.8	79.3	79.0	90.2
R2	70.6	80.8	70.1	90.0
R3	75.0	68.0	74.7	87.4
Dynamic Combined			83.0	89.2

Table 4: Initial results of combined shot detection method averaged over all program types

Table 5 shows the Precision and Recall performance of the dynamic combined and R1 method (which performed best) over the corpus. As we can see we have an increase in Recall for every program except 7. The Precision figures are lower but they have not affected the overall Precision value greatly.

	R1		Combined	
Program	**Precision**	**Recall**	**Precision**	**Recall**
1	90.9	74.7	79.3	92.5
2	94.9	64.4	94.7	70.7
3	83.8	62.9	83.0	66.7
4	91.6	82.9	90.9	83.9
5	86.4	91.7	86.6	93.2
6	92.1	81.7	91.9	83.3
7	91.8	89.9	92.3	89.9
8	90.2	78.6	90.0	81.3
9	92.9	85.5	92.9	86.5
10	89.6	73.7	88.8	78.2
11	89.6	79.4	89.6	82.0
12	94.1	82.9	93.8	86.2
13	86.9	77.0	86.9	81.2
Avg	**90.4**	**78.9**	**89.3**	**82.7**

Table 5: Combined and R1 Methods, Dynamic Threshold

One of the aspects of this work that was of interest was to identify what programs or program genres give the best and the worst performance. Clearly, the combined strategy has a negligible effect on precision for all programs except program 1, which is RTE 1pm News, while it has a positive effect on recall, the greatest impact being on that same RTE 1pm News program. Program three, which is an Australian soap called 'Home and Away' remains the poorest performing program though with a precision figure in the mid-80s this is still respectable.

6 Conclusions and Impact

In this paper we have presented a summary of our experimental results on three different shot boundary detection algorithms on a variety of video program genres, and an investigation into how these techniques could be combined. Our figures show a mixed bag of results with some techniques doing better than others on different shot transition types and program genres. Our investigation into combining strategies showed a small improvement in performance.

At this point in our work we must ask ourselves whether we have reached a point of diminishing returns. The computational task involved in each of the techniques is quite considerable. In our work we use a SUN Enterprise server which is dedicated to performing this task only, and yet our work runs in approximately real time *for each technique*. The reason for performing a shot boundary detection process is so that a video can be structured into higher order units (shots) for indexing, browsing, search, summarisation and other content-based operations. It may be that because these are ultimately delivered to humans for human analysis, errors in the shot boundary detection process may be tolerable. Certainly, we have found a good deal of user tolerance when it comes to browsing video on the Físchlár system for which this work has been developed. On the other hand, as our work progresses from video browsing (which we currently do) through to video indexing and retrieval, video summarisation and video filtering,

user tolerance of basic errors in identifying shot boundaries may disappear and we will find it more worthwhile incorporating the work presented here into our operational video navigation system.

References

Arman, F., Hsu, A and Chiu, M. *Feature management for large video databases*. In Storage and Retrieval for Image and Video Databases, pages 2-12, 1993.

Boon-Lock, Y., Liu, B. *A unified approach to temporal segmentation of motion jpeg and mpeg compressed video*. International Conference on Multimedia Computing and Systems, pages 81-83. IEEE, Los Alamitos, Ca, USA, 1995.

Boreczky, J.S., Rowe, L.A. *A Comparison of Video Shot Boundary Detection Techniques*. Storage & Retrieval for Image and Video Databases IV, I.K. SPIE 2670, pages 170-179, 1996.

Christel, M. and Martin, D. *Information visualization within a digital video library*. Journal of Intelligent Information Systems, vol 6, pages 235-257, 1998.

Hampapur, A., Jain, R and Weymouth, T.E. *Production model based digital video segmentation*. Multimedia Tools and Applications, pages 9-46, 1995.

Lee H., Smeaton AF., Berrut C., Murphy N., Marlow S and. O'Connor NE. *Analysis and implementations of several keyframe-based browsing of digital video*, 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000), Lisbon, Portugal.

Lee, H., Smeaton AF., O'Toole, C., Murphy, N., Marlow S and O'Connor, N.E. *The Físchlár Digital video Recording, Analysis, and Browsing System*. RIAO '2000: Content-Based Multimedia Information Access, Paris, France, April 12-14, 2000.

O'Toole C., Smeaton AF., Murphy N and Marlow S. *Evaluation of Automatic Shot Boundary Detection on a Large Video Test Suite*, Challenges for Image Retrieval, Brighton 1999.

Ruiloba, R., Joly, P., Marchand-Millet, S., Quénot, G. *Towards a Standard Protocol for the Evaluation of Video-to-Shots Segmentation Algorithms*. CBMI 1999 Proceedings of the European workshop on content-based multimedia indexing, Toulouse France, October 15-27, 1999.

Simpson-Young, B and Yap, K. *FRANK: trialing a system for remote navigation of film archives*. SPIE International Symposium on Voice, Video and Data Communications, Boston MA, November 1996.

Smeaton AF., Gilvarry J., Gormley G., Tobin B., Marlow S and Murphy M. *An Evaluation of Alternative Techniques for Automatic Detection of Shot Boundaries in digital video*. Irish Machine Vision and Image Processing Conference Dublin, Ireland, 8-9 September 1999.

Smeaton AF. *Independence of Contributing Retrieval Strategies in Data Fusion for Effective Information Retrieval*. Proceedings of the 20th BCS-IRSG Colloquium, Grenoble, France, Springer-Verlag Workshops in Computing, April 1998.

The Multimedia and digital video Technologies Group, NIST website, http://www.antd.nist.gov/madvtg/

Xiong, W., Lee C.J and Ip, M. *Net comparison: a fast and effective method for classifying image sequences*. In Niblack and Jain , pages 318-326, 1995.

Zabih, R., Miller, J and Mai, K. *A feature-based algorithm for detecting and classifying scene breaks*. In 3rd International Multimedia Conference and Exhibition, Multimedia Systems, pages 189-200, San Francisco, California 1995.

Zhang, H. Smoliar, S.W and Wu, H.J. *Content-based video browsing tools*. In Multimedia Computing and Networking, pages 389-398, 1995.

Bouthemy, P., Garcia, C., Ronfard, R., Tziritas, G. Scene segmentation and image feature extraction for video indexing and retrieval. In Visual Information and Information Systems, pages 245-252, 1996.

Niblack, W., Jain C.R. editors, Storage and retrieval for image and video databases 3, February 1995.

Multimedia Computing and Networking, Proc SPIE, Society for Optical Engineering, February 1996.