

# Detection and Classification of Shot Transitions

Sarah Porter and Majid Mirmehdi and Barry Thomas  
Department of Computer Science  
University of Bristol  
Bristol, BS8 1UB, UK  
{porter,majid,barry}@cs.bris.ac.uk

## Abstract

The process of shot break detection is a fundamental component in automatic video indexing, editing and archiving. This paper introduces a novel approach to the detection and classification of shot transitions in video sequences including cuts, fades and dissolves. It uses the average interframe correlation coefficient and block-based motion estimation to track image blocks through the video sequence and to distinguish changes caused by shot transitions from those caused by camera and object motion. We achieve better results compared with two established techniques.

## 1 Introduction

Indexing and annotating large quantities of film and video material is becoming an increasing problem throughout the broadcasting industry, particularly where archived material is concerned. Manually indexing video content is currently the most accurate method but it is a very time consuming process. Considerable amounts of archived data remain unindexed which often leads to the production of new film instead of the reutilisation of existing material. An efficient video indexing technique is to temporally segment a sequence into shots, where a shot is defined as a sequence of frames captured from a single camera operation, and then select representative key-frames to create an indexed database. Hence, a small subset of frames can be used to retrieve information from the video and enable content-based video browsing.

There are two broad classes of transitions defining boundaries between shots: abrupt (discontinuous), also referred to as *cuts*; or gradual (continuous), such as *fades* and *dissolves*. A cut is an instantaneous change from one shot to another. During a fade, a shot gradually appears from, or disappears to, a constant image. A dissolve occurs when the first shot fades out whilst the second shot fades in. Shots unified by a common locale or event are grouped together into scenes. Gradual transitions are often used at scene boundaries to emphasise the change in content of the sequence [1]. Hence, detecting gradual transitions is particularly important for the identification of key-frames.

In the case of shot cuts, the content change is usually large and easier to detect [4, 5]. However, the inter-frame difference during a gradual transition is usually small. This makes it difficult to distinguish changes caused by a continuous edit effect from those caused by object and camera motion without also incurring a large number of false positives. A comparison of recent algorithms shows that the false positive rate when detecting dissolves is usually unacceptably high, indicating that reliable dissolve detection is still

an unsolved problem [4]. In this paper we introduce a novel approach for the detection and classification of the most commonly used shot transitions: cuts, fades and dissolves.

Section 2 presents an algorithm designed explicitly to detect shot cuts using block-based motion compensation. Normalised correlation implemented in the frequency domain is used to estimate the motion for each block. In Section 3, the algorithm for shot cut detection is extended to detect fades and dissolves. The proposed method uses block tracking to differentiate between changes caused by gradual effects from those caused by object and camera motion and it has been designed to handle some of the shortcomings of previous methods [9, 10]. Experimental results confirming the validity of the approach are presented and discussed in Section 4.

## 2 Shot Cut Detection

Most of the existing methods for shot cut detection use some inter-frame difference metric such as those outlined by Yussof et al. [8]. A frame pair where this difference is greater than some predefined threshold is considered to contain a shot cut. Histogram-based methods are the most common approach to shot cut detection in use today, since they offer a good trade-off between accuracy and computational efficiency [1, 5, 6, 10]. Other approaches can be grouped into pixel-based methods [10] and region-based methods [6, 10]. Most of these are weakened by large object and camera motions. Hence, motion-based algorithms have been proposed to distinguish between differences caused by motion and those caused by an abrupt transition [5, 1, 7].

We propose a motion-based algorithm to identify shot cuts which inherently deals with object and camera motion. It uses block-matching motion compensation to generate an inter-frame difference metric. For each block in frame  $n$ , the best match in a neighbourhood around the corresponding block in frame  $n + 1$  is sought. This is achieved by calculating the normalised correlation between blocks and locating the maximum correlation coefficient. Calculating the normalised correlation in the spatial domain is, however, prohibitively expensive unless the blocks are small. Hence, we perform normalised correlations in the frequency domain [2] defined by:

$$\rho(\xi) = \frac{\mathcal{F}^{-1}\{\hat{x}_1(\omega) \hat{x}_2^*(\omega)\}}{\sqrt{\int |\hat{x}_1(\omega)|^2 d\omega \cdot \int |\hat{x}_2(\omega)|^2 d\omega}} \quad (1)$$

where  $\xi$  and  $\omega$  are the spatial and spatial frequency coordinate vectors respectively,  $\hat{x}_i(\omega)$  denotes the Fourier transform of block  $x_i(\xi)$ ,  $\mathcal{F}^{-1}$  denotes the inverse Fourier operator and  $*$  is the complex conjugate. A high-pass filter is applied to each image before performing the correlations to accentuate the contributions from higher spatial frequencies, since a correlation field derived from high-pass regions will contain more detectable peaks. Correlation fields derived from low-pass regions will result in a flat correlation field leading to inaccurate peak detection [3]. For this reason, blocks with insufficient energy are not used. A consequence of applying the high-pass filter is that the mean of the image is removed. Hence, the correlation between blocks is invariant to changes in the mean intensity. By normalising the correlation, the method is insensitive to a positive scaling of the image intensities. Most of the previous methods for shot cut detection may falsely detect a shot cut where there are sudden intensity changes within a shot, for example, where

there is a change in the lighting conditions. By applying a high-pass filter and performing normalised correlation our method is robust to changes in the global illumination.

The location of the maximum correlation coefficient can be used to find the offset of each block in frame  $n + 1$  from its position in frame  $n$ . Previous approaches use the estimated motion vectors to calculate the motion-compensated frame difference [7]. In contrast, our proposed approach uses only the value of the maximum correlation coefficient, as a *goodness-of-fit* measure for each block. The value of the goodness-of-fit measure lies between 0 and 1, where a value of 0 indicates a complete mismatch and a value of 1 indicates a perfect match. Between two frames belonging to the same shot, the goodness-of-fit for the majority of the blocks should be large, indicating a good match. A high number of poor matches should suggest the presence of a shot cut.

A similarity metric for each frame pair is derived by combining the goodness-of-fit measure of all the blocks. To achieve this the mean  $\mu$  of the goodness-of-fit measures is computed, defined as

$$\mu = \frac{\sum_{i=1}^B p_i}{B} \quad (2)$$

where  $p_i = \max(\rho(\xi))$  for block  $i$ , and  $B$  is the total number of blocks. During a shot there may be some blocks with poor matches due to occlusion or data that violates the 2-d translational model. To prevent these outliers negatively influencing the similarity metric for a frame pair, those goodness-of-fit measures outside one standard deviation of  $\mu$  are removed. The final similarity metric  $M_n$  for a frame pair  $n$  and  $n + 1$  is the recalculated average of the remaining goodness-of-fit measures. Given  $\overline{M}$  as the average of the previous similarity measures since the last shot cut, defined as

$$\overline{M} = \frac{\sum_{i=1}^{n-1} M_i}{n-1}. \quad (3)$$

then a shot cut is detected if  $\overline{M} - M_n > T_C$ , i.e. if the rate of change from the average similarity measure is greater than some threshold  $T_C$ . Figure 1 shows a plot of  $M_n$  for three different video sequences. It can be seen that during a shot  $M_n$  remains high (close to 1). On the other hand, a shot cut manifests itself as a sudden decrease in  $M_n$ .

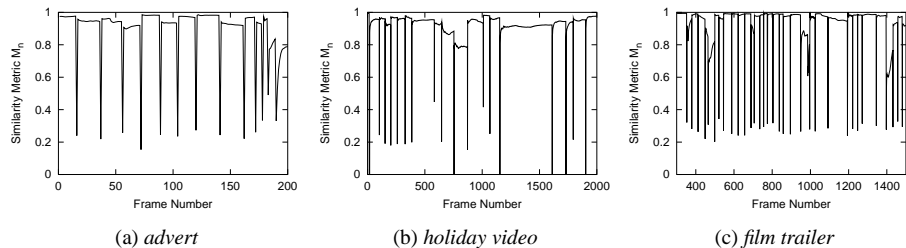


Figure 1: Similarity metric  $M_n$  for three different video sequences.

The choice of optimal block size is an ill-defined problem. A large block is more likely to invalidate the model of a single translational motion per block whereas a small block is less likely to contain enough intensity variation which makes it difficult to measure the motion accurately. In this work a block size of  $32 \times 32$  was chosen empirically as it gives an acceptable trade-off between accuracy and the resolution of the motion field.

### 3 Detecting Fades and Dissolves

The method described above responds well to shot cuts but overlooks other edit effects. We now extend our method to detect fades and dissolves. The difference between a frame pair during a gradual transition is much smaller than that which occurs during a shot cut. Lowering the threshold to detect such small changes would result in many false detections due to the differences caused by camera and object motion. Zhang et al. [10] proposed a “twin comparison” technique which compares the histogram difference with two thresholds. A lower threshold is used to detect small differences that occur for the duration of the gradual transition while a higher threshold is used in the detection of shot cuts and gradual transitions. Zabih et al. [9] proposed another method to detect edit effects by checking the spatial distribution of exiting and entering edge pixels. Both of these techniques are designed to detect cuts, fades and dissolves, so they have been used for comparison with our shot transition detection algorithm.

As in shot cut detection, most of the previous approaches to detecting gradual transitions are also sensitive to object and camera motions. We propose a method that can differentiate between changes caused by a gradual transition from those caused by camera and object motion.

The shot cut detection method described in Section 2 was straight forwardly extended to detect fades. The end of a fade-out and the start of a fade-in is marked by a constant image. A constant image contains very little, if any, high-pass energy. Therefore, correlation of an image with a constant image results in  $M_n = 0$  which can be used to identify the end of a fade-out and the start of a fade-in. However, gradual transitions occur over a number of frames so knowledge of the boundaries of the edit effect is required. A fade is a scaling of the pixel intensities over time which can be observed in the standard deviation of the pixel intensities [4]. If  $M_n$  falls to 0 and the standard deviation of the pixel intensities decreased prior to this, the frame where the standard deviation started to decrease is marked as the first frame of the fade-out. The decrease in the standard deviation must have occurred over more than two frames to distinguish fade-outs from a shot cut to a constant image. Similarly, if the standard deviation of the pixel intensities increases after the similarity metric increases from 0, the frame where the standard deviation becomes constant is marked as the end of the fade-in. Again, this must have occurred over more than two frames. Initially, to compute where the standard deviation becomes constant after a fade-in, the standard deviation of frame  $n$ ,  $\sigma_n$  was compared to the standard deviation of frame  $n - 1$ ,  $\sigma_{n-1}$ . If  $\sigma_n \leq \sigma_{n-1}$  then the end of the fade-in was marked. However, we observed that often the scaling factor is not altered for every frame but only every other frame. Therefore, the end of a fade-in would be marked too early. Hence, the end of the fade-in was marked when  $\sigma_n \leq \frac{\sigma_{n-1} + \sigma_{n-2}}{2}$ . A similar comparison is used to detect the start of a fade-out.

Extending this method to detect dissolves is somewhat more involved. The difference between each frame pair during a dissolve is so small that  $M_n$  does not indicate that a dissolve has occurred. We divide the first frame of a sequence into a regular grid of blocks of size  $32 \times 32$ . A selection of these blocks is then used to represent *regions of interest (ROI)* in the image. A block is selected as a ROI if

$$\sigma_b^2 > \frac{\sigma_l^2}{\ln(\sigma_l^2)}. \quad (4)$$

where  $\sigma_b^2$  is the variance of a block  $b$  and  $\sigma_I^2$  is the variance of the image  $I$ . This is to prevent all of the blocks in an image with low variance being selected as ROI. Figure 2 shows the first frame of two shots and their selected ROI highlighted in white.

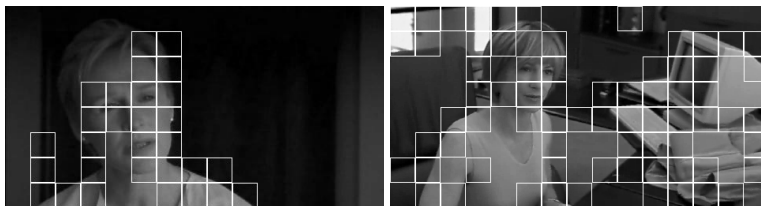


Figure 2: Blocks are selected to be regions of interest (ROI) in the first frame of each shot.

In Section 2, the method for shot cut detection discarded the motion vector estimated from block matching. However, motion estimation between frame pairs is now used to track blocks over time in the video sequence. Between each frame pair  $n$  and  $n + 1$ ,  $M_n$  is still computed to detect shot cuts. In addition, each ROI is correlated with its new location in frame  $n + 1$ ,  $n + 2$  etc., as shown in Figure 3(a–c), until the end of the next edit effect or until the block is removed. The value of the correlation peak,  $\max(\rho(\xi))$ , is used as a goodness-of-fit measure for each ROI over time. A single similarity metric  $F_n$  for the set of ROI is calculated in the same way as  $M_n$  i.e. the recalculated average of the goodness-of-fit measures for the set of ROI after removing outliers.

Whilst tracking, object or camera motion may cause blocks to become overlapped as shown in Figure 3(c). Once this occurs the block tracking is no longer reliable because block-matching can not resolve occlusion. Therefore, blocks that are overlapping or have begun to move outside the image are removed as shown in Figure 3(d). If any of the removed blocks were a ROI they are also removed from the current set of ROI. This will leave areas of the image uncovered, contents of which still need to be tracked. For this reason we try to reintroduce new blocks in the uncovered areas. This is achieved by comparing the current positions of the remaining blocks to a regular spatial grid. Any blocks in this regular grid that are not covered by the current set of blocks are added as shown in Figure 3(e). If a new block satisfies (4) it is added to the current set of ROI. Once this is complete the algorithm continues to track the blocks into the next frame (Figure 3(f)). The addition and removal of blocks allows the set of ROI to be updated for changes due to camera and object motion.

During a shot,  $F_n$  should remain high indicating that the contents of each ROI has not changed significantly. During a dissolve, the content of each ROI gradually changes and  $F_n$  will decrease until it reaches its lowest value at the end of the dissolve. During a shot  $M_n$  and  $F_n$  should be approximately equivalent. Rather than compare the value of  $F_n$  to a threshold, we want to compare how much it has changed with respect to  $M_n$ . Hence, we define the ratio  $R_n$  as

$$R_n = \frac{M_n}{F_n}. \quad (5)$$

If  $R_n$  is greater than a threshold  $T_D$  then the end of the dissolve is marked once  $R_n$  reaches its maximum. The start of the dissolve is marked where  $R_n$  started to increase.

Figure 4 illustrates  $F_n$  and  $R_n$  during three consecutive dissolves in a video sequence. It can be seen that  $F_n$  decreases during a dissolve and reaches its minimum at the end.

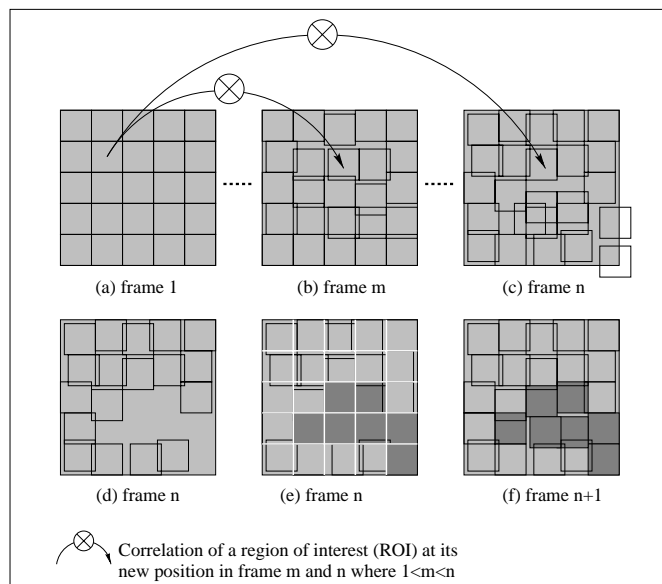


Figure 3: (a-c) Blocks are tracked over time and may become overlapped, (d) overlapped blocks removed, (e) blocks added in uncovered area, (f) blocks continue to be tracked.

During these three dissolves  $M_n$  remained approximately equal to 1, causing  $R_n$  to increase during each dissolve. The three dissolves are therefore easily detected.

After every detected shot transition, the first frame of the next shot is divided into a regular grid and a new set of ROI are selected to be tracked for the detection of the next edit effect.

## 4 Comparative Results

To evaluate the performance of this algorithm we compared it with the performance of two other methods, one histogram-based and one feature-based. The histogram-based method was chosen since it is a well established technique and has been shown to perform well in detecting edit effects [5]. The feature-based method was chosen for its ability to detect gradual transitions, although its performance was reported on limited test sequences [9].

The histogram-based method we used is similar to the method with the best performance in the comparative investigation by Lupatini et al. [5]. This approach uses the  $\chi^2$  value to define the difference between two global colour histograms which is compared against two thresholds,  $T_H$  and  $T_L$ . Whenever the histogram difference between two consecutive frames is greater than  $T_H$ , a shot cut is detected. If the difference lies between the two thresholds the frame is marked as the potential start of a gradual transition. Successive frames are then compared with the first frame of the transition and if the difference exceeds  $T_H$ , a gradual transition is detected. The end of the gradual transition is marked once the difference between frame pairs drops below  $T_L$  for two frame pairs.

The feature-based method used was by Zabih et al. [9] who have made the code for this algorithm available enabling us to apply exactly the same implementation. They base

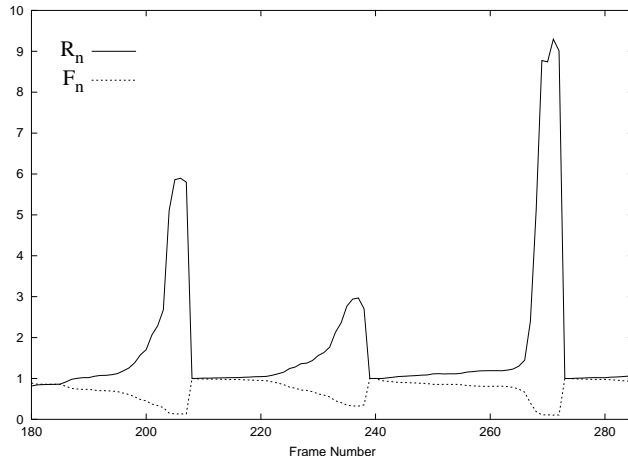


Figure 4: Feature similarity metric  $F_n$  and ratio  $R_n$  during three consecutive dissolves.

their approach on the idea that during a cut or a dissolve, new edges appear far from the locations of disappearing, older edges. By comparing the relative values of entering and exiting edge pixels the method classifies cuts, fades and dissolves. A registration technique is used to compensate for global motion between two frames. To compensate for small object motions edge pixels in one image within a small distance of edge pixels in the other image are not considered to be entering or exiting edges.

To test these methods we used 10 different movie trailers. We found these were a good source of data since they tend to contain many shot transitions over a short sequence. The locations and types of these transitions were hand-labelled for comparison, a subset of which can be seen in Table 1. The distribution of shot transitions in the complete set of test data is shown in the last row.

Sequence	Cuts	Fade-ins	Fade-outs	Dissolves
1	19	26	25	10
2	77	0	0	11
3	11	4	4	6
4	18	6	6	2
5	81	6	8	13
<b>Total for 1-10</b>	<b>450</b>	<b>79</b>	<b>74</b>	<b>114</b>

Table 1: Number and types of edit effects contained in 5 sample sequences and the total test data set (10 sequences)

For our proposed approach and the histogram-based method we experimented with a small set of training data to select thresholds which gave the “best” performance and then ran them over the complete data set using the same thresholds. For the feature-based method [9] we used the thresholds reported by the authors. Two comparative studies [5, 1] reported the histogram-based method to perform well. However, these were not concerned with the direct classification of edit effects into cuts, fades and dissolves. Therefore, if there were several shot cut boundaries detected within a gradual transition the first one was

marked as correct and the method was not penalised for reporting multiple consecutive frames to reduce the number of false positives. However, a novel aspect of our work is that we do want to make a distinction between abrupt and different gradual transitions. This made selecting thresholds for the histogram-based method of [5] more difficult: during several gradual effects (particularly short fades) the difference between frame pairs was greater than  $T_H$ . In such a case we considered all of the shot cuts to be false and the gradual transition to be undetected. By increasing  $T_H$  to prevent the misclassification of gradual transitions, several more edit effects would be undetected since the difference would not exceed  $T_H$ . For our experiments we chose a high value for  $T_H$  to decrease the number of incorrect classifications at the expense of missing some other effects.

Two parameters often used to compare the performance of shot boundary detection methods are recall and precision [1], defined as

$$\text{Recall} = \frac{N_C}{N_C + N_M} \quad \text{Precision} = \frac{N_C}{N_C + N_F} \quad (6)$$

where  $N_C$ ,  $N_M$ , and  $N_F$  are the number of correctly detected, the number of missed, and the number of falsely detected shot transitions respectively. In other words, recall is the percentage of true transitions detected, and precision is the percentage of detected transitions that are actually correct. The performance of our motion-based algorithm (MB) compared with those of the histogram-based (HB) and feature-based methods (FB) for shot cut detection only can be seen in Table 2. A comparison of the performance of the algorithms for the detection of gradual transitions can be seen in Tables 3 and 4. In these three tables we report the results for the first 5 sequences and then the total across all 10 sequences in the data set. It should be noted that while we classify gradual transitions into fade-ins, fade-outs, and dissolves, FB only classifies into fades and dissolves and HB does not make a distinction at all. For these experiments, if an edit effect was detected but classified incorrectly, it was considered a false detection and the actual edit effect was labelled undetected. From Tables 2-4 it is simple to notice the better performance of our method compared with the other two techniques.

	MB			HB			FB		
	Cuts								
Sequence	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$
1	19	0	1	10	9	19	19	0	114
2	77	0	1	48	29	1	73	4	8
3	9	2	0	5	6	8	7	4	8
4	18	0	0	13	5	0	17	1	3
5	81	0	4	72	9	20	70	11	13
<b>Total for 1-10</b>	<b>410</b>	<b>40</b>	<b>48</b>	<b>301</b>	<b>149</b>	<b>190</b>	<b>329</b>	<b>121</b>	<b>224</b>

Table 2: Detection and classification of shot cuts for 5 sample sequences and the total test data set (10 sequences)

Table 5 summarises the performance of the algorithms by comparing the recall and precision of each one (after combining the results for the gradual transitions for FB and MB). FB’s performance was disappointing as it detected many false gradual transitions and few of the actual gradual transitions, reflected by the low precision and recall values

Sequence	MB								
	Fade-ins			Fade-outs			Dissolves		
	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$
1	26	0	0	25	0	0	9	1	1
2	0	0	0	0	0	0	11	0	1
3	4	0	0	4	0	0	6	0	2
4	6	0	0	6	0	0	6	0	0
5	3	3	0	8	0	0	11	2	6
<b>Total for 1-10</b>	<b>64</b>	<b>15</b>	<b>1</b>	<b>71</b>	<b>3</b>	<b>6</b>	<b>99</b>	<b>15</b>	<b>63</b>

Table 3: Detection and classification of gradual effects for our proposed method MB for 5 sample sequences and the total test data set (10 sequences).

Sequence	HB			FB					
	Gradual			Fades			Dissolves		
	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$	$N_C$	$N_M$	$N_F$
1	42	19	1	0	51	2	0	10	0
2	9	2	4	0	0	2	7	4	20
3	9	5	1	5	3	9	2	4	4
4	11	3	0	8	4	1	2	0	1
5	14	13	4	8	6	7	10	3	38
<b>Total for 1-10</b>	<b>155</b>	<b>112</b>	<b>27</b>	<b>66</b>	<b>87</b>	<b>86</b>	<b>55</b>	<b>59</b>	<b>164</b>

Table 4: Detection and classification of gradual effects for HB and FB for 5 sample sequences and the total test data set (10 sequences).

	MB		HB		FB	
	Cuts	Gradual	Cuts	Gradual	Cuts	Gradual
Recall (%)	91	87	67	58	73	45
Precision (%)	90	77	30	85	60	33

Table 5: Recall and precision for each method over all the cuts and gradual transitions.

(33% and 45% respectively). There are several reasons for this. The algorithm compensates only for translational motion, hence zooms are a cause of false detections. Also, the registration technique only computes the dominant motion so multiple object motions within the frame are another source of false detections. Furthermore, if there are strong motions before or after a cut, the cut is misclassified as a dissolve and cuts to or from a constant image are misclassified as fades.

The results for the HB method were a considerable improvement on the FB approach. The biggest drawback is that although a high value for  $T_H$  was chosen, many gradual transitions are still misclassified as shot cuts, resulting in a low recall value for gradual transitions (58%) and a low precision value for shot cuts (30%). One reason why the recall values for HB are low is that it misses edit effects between shots with similar colour distributions. Another reason is that if a gradual transition is closely followed by another, then HB often detects this as a single transition so the first one is detected and the second is considered undetected. Finally, another source of false detections was camera and object

motion that created changes similar to that caused by a gradual transition.

Our proposed motion-based algorithm gives the most favourable results with high recall and precision values for both cuts and gradual transitions. Moreover, our algorithm is able to classify fade-ins, fade-outs, and dissolves. The precision value for gradual transitions is higher in HB (85%) than in ours (77%) because  $T_H$  was set high to reduce the number of gradual transitions detected as shot cuts; this resulted in fewer false detections  $N_F$  since the amount of change caused by camera and object motion rarely exceeded  $T_H$ . The main cause of false detections of dissolves in our technique was due to the contents of a ROI changing, not in the presence of a dissolve, but for example due to motion blur or a light source that saturates a large part of the image. Also, if a shot cut is undetected, then the set of ROI are not updated and they are tracked into the next shot resulting in a misclassification as a dissolve.

## 5 Conclusions

We have presented a novel, unified approach that classifies shot boundaries with a better resolution into cuts, fade-ins, fade-outs and dissolves. The recall and precision values show either a significant improvement on other approaches or are easily comparable given that all shot transitions are separately resolved.

A weakness of our method is that it will track the most dominant motion if there are multiple motions within a block. This can cause the contents of a ROI to change and result in a decrease in  $F_n$  leading to a false detection of a dissolve. Such problems might be improved by using a multiresolution model to estimate the motion [3].

## References

- [1] J Boreczky and L Rowe. Comparison of video shot boundary detection techniques. In *SPIE Conf. Storage & Retrieval for Image & Video Databases*, volume 2670, pages 170–179, 1996.
- [2] A D Calway, H Knutsson, and R Wilson. Multiresolution estimation of 2-d disparity using a frequency domain approach. In *British Machine Vision Conference*, pages 227–236, 1992.
- [3] S Kruger. *Motion Analysis and Estimation using Multiresolution Affine Models*. PhD thesis, Department of Computer Science, University of Bristol, October 1998.
- [4] R Lienhart. Comparison of automatic shot boundary detection algorithms. In *SPIE Conf. on Storage and Retrieval for Image & Video Databases VII*, volume 3656, pages 290–301, 1999.
- [5] G Lupatini, C Saraceno, and R Leonardi. Scene break detection: a comparison. In *8th International Workshop on Research Issues in Data Engineering*, pages 34–41, 1998.
- [6] A Nagasaka and Y Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Database Systems*, volume 2, pages 113–127, 1992.
- [7] B. Shahraray. Scene change detection and content-based sampling of video sequences. In *Digital Video Compression: Algorithms and Technologies*, volume 2419, pages 2–13, 1995.
- [8] Y Yusoff, W Christmas, and J Kittler. A study on automatic shot change detection. In *3rd European Conf. on Multimedia Applications, Services and Techniques*, pages 177–189, 1998.
- [9] R Zabih, J Miller, and K Mai. A feature-based algorithm for detecting and classifying scene breaks. In *ACM Multimedia '95 Proceedings*, pages 189–200. ACM Press, 1995.
- [10] H Zhang, A Kankanhalli, and S W Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, January 1993.