Shot boundary detection using the moving query window

S.M.M. Tahaghoghi James A. Thom Hugh E. Williams

School of Computer Science and Information Technology
RMIT University, GPO Box 2476V
Melbourne, Australia, 3001

{*saied,jat,hugh*}*@cs.rmit.edu.au*

Abstract

The large volume of video content generated each day requires efficient and effective methods of video indexing and retrieval. A common first step in indexing video content is to identify visually and semantically continuous segments or shots. In this paper, we present the moving query window approach to video shot boundary detection. This uses the techniques of query-by-example (QBE) and ranked results, both often used in content-based image retrieval (CBIR). Each frame of the video is used in turn as an example query on the image collection formed by the other frames within a moving window. Transitions are detected by monitoring the relative ranks of these frames in the results list. We show that this is an effective approach for the shot boundary detection task of the TREC-11 video track.

1 Introduction

Video is the next frontier of visual information retrieval: for archived footage to be useful, its contents must be known. Since a video clip has a time dimension, this generally means that the content must be reviewed sequentially, and sections of interest identified. This is costly and tedious to perform manually, and so automatic techniques are required.

A video stream can be considered to be composed of small, coherent sections or *shots*, where adjacent frames are usually similar. A small sample of frames can be selected from each shot and indexed for use in video retrieval [4, 21]. The answer to a video retrieval information need is then a list of shots containing frames similar to the query requirements.

A shot is bounded at each end by a transition. The main types of transition are cuts, fades, dissolves, and spatial edits [5, 16]. The type and frequency of transitions in a video clip is largely dependent on the age of the footage and the nature of the content. Almost all transitions in fast-moving television news footage are cuts, and dissolves are rare. In a documentary, dissolves and fades appear frequently. Cuts, dissolves, and fades account for the majority of transitions; Lienhart [10] reports this is more than 99%, and similar ratios were observed in the TREC-10 and TREC-11 video collections.

Video footage can be segmented into shots by detecting the shot start and end points, as signified by transitions. The difference between adjacent frames of a shot is usually small, but increases during transitions. Most shot boundary detection algorithms identify transitions by monitoring for significant changes in the video frames.

One method to measure this change is to compare frames pixel-by-pixel: transitions are reported if the colour or intensity of a significant number of pixels changes between frames [2]. However, pixel-by-pixel comparison of frames is generally computationally intensive, and sensitive to object motion, noise, camera motion, and changes in camera zoom. Using information produced during video compression is typically efficient [1, 12, 23], but may result in low precision [2]. Computing and comparing statistics of the frames—such as the mean and standard deviation of pixel values [9], or histograms of colour usage [13, 25]—reduces sensitivity but adds computation overhead.

Recent work in this area using colour histograms includes that of Pickering et al. [14]. In their approach, frames are divided into nine blocks, and red, green, and blue (RGB) colour component histograms extracted from each. The Manhattan distance between the histograms of corresponding blocks is calculated, and the

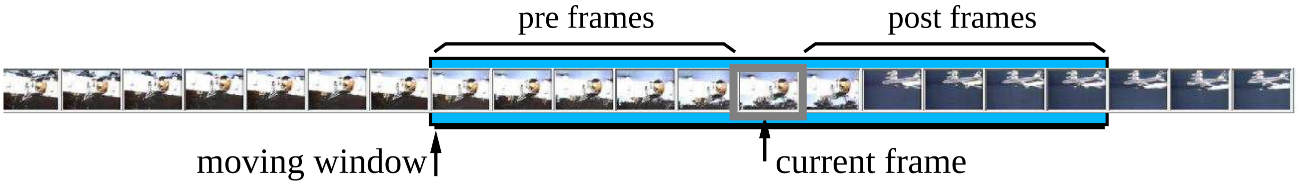


Figure 1: Moving query window with a half window size (HWS) of 5; the five frames before and the five frames after the current frame form a collection on which the current frame is used as a query example.

largest of the three is retained as the distance between the blocks. The median of the nine individual inter-block distances is taken as the inter-frame distance. A transition is reported if this distance is greater than a fixed threshold and also greater than the average distance value for the 32 surrounding frames.

Sun et al. [18] compare the colour histograms of adjacent frames within a moving window; a shot boundary is reported if the distance between the current frame and the immediately preceding one is the largest inter-frame distance in the window, and significantly larger than the second largest inter-frame distance.

The IBM `CueVideo` program uses a sampled three-dimensional RGB colour histogram to measure the distance between pairs of frames [17]. Histograms of recent frames are cached, and statistics are calculated for this moving window. These statistics are used to determine adaptive threshold levels.

In this paper, we present our approach to video segmentation based on the concepts of querying by example image (QBE) and ranked results, both regular features of content-based image retrieval (CBIR). In the next section, we introduce our new approach. Section 3 addresses our choice of features and parameters. In Section 4, we review the performance of our technique on the TREC-11 shot boundary detection task. In Section 5, we conclude and discuss possible areas for improvement.

2 The Moving Query Window Technique

A content-based image retrieval or CBIR system aims to satisfy the information need of a user by selecting images from the collection that best meet the user's requirements. With many CBIR systems, users convey their requirements by selecting features such as colour and texture from a palette [3], sketching a representation of the desired image [8], or providing an example image that captures the qualities of the target image [7, 19]. The last two methods are categorised as query-by-example, or QBE.

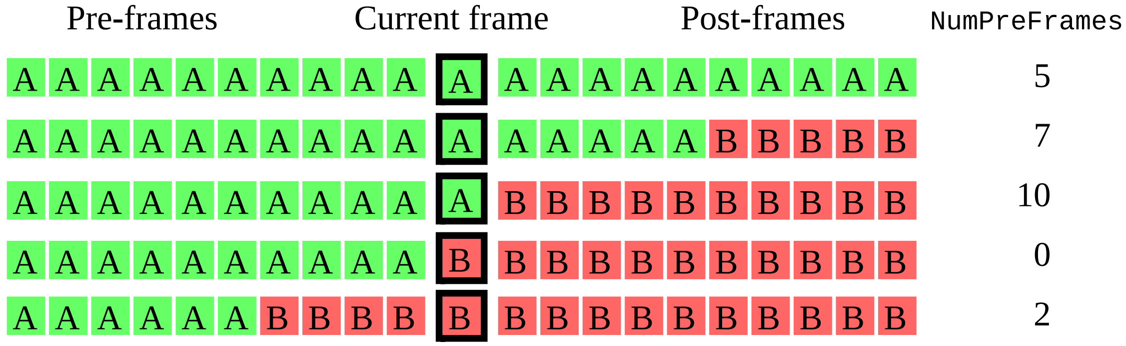
In CBIR, a summary is produced for each image in the collection that captures visual aspects such as colour and texture distributions, and the shape and location of objects in the image. When using QBE, a corresponding summary is produced for the query. These summaries are compared, and collection images are ranked by similarity to the query. The user is then presented a list of all the images in the collection, ranked from most- to least-similar.

We have applied the concepts of QBE and ranking to the video segmentation problem. We were motivated to explore this approach by the observation that frames preceding a transition are similar in content but are usually dissimilar to those following the transition.

We define a moving window of size N extending equally on either side of the current frame, but not including the current frame itself. The number $\frac{N}{2}$ is referred to as the half window size (HWS). We refer to the $\frac{N}{2}$ frames preceding the current frame as the *pre-frames*. Similarly, the $\frac{N}{2}$ window frames following the current frame are *post-frames*. Figure 1 shows a moving window of ten frames, with five pre- and post-frames on either side of the current frame.

We use the current frame as a query on the collection of frames inside this moving window, that is, to the pre- and post-frames. This QBE orders the N collection frames by decreasing similarity to the query frame, with the most similar frame first, and the most dissimilar frame last.

The difference between the current frame—which is used as the query example—and the frames before and after it will usually be near-symmetrical. Thus, the pre- and post-frames will be interspersed throughout the ordered list of window frames, and the number of pre- and post-frames in the top $\frac{N}{2}$ results will be approximately equal. However, this changes in the vicinity of a transition.



Pre-frames										Current frame	Post-frames										NumPreFrames
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	5
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	B	B	B	B	7
A	A	A	A	A	A	A	A	A	A	A	B	B	B	B	B	B	B	B	B	B	10
A	A	A	A	A	A	A	A	A	A	B	B	B	B	B	B	B	B	B	B	B	0
A	A	A	A	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	2

Figure 2: As the moving window traverses an abrupt transition, the number of pre-frames in the $\frac{N}{2}$ frames most similar to the current frame varies significantly. This number (`NumPreFrames`*) rises to a maximum just before an abrupt transition, and drops to a minimum immediately afterwards.*

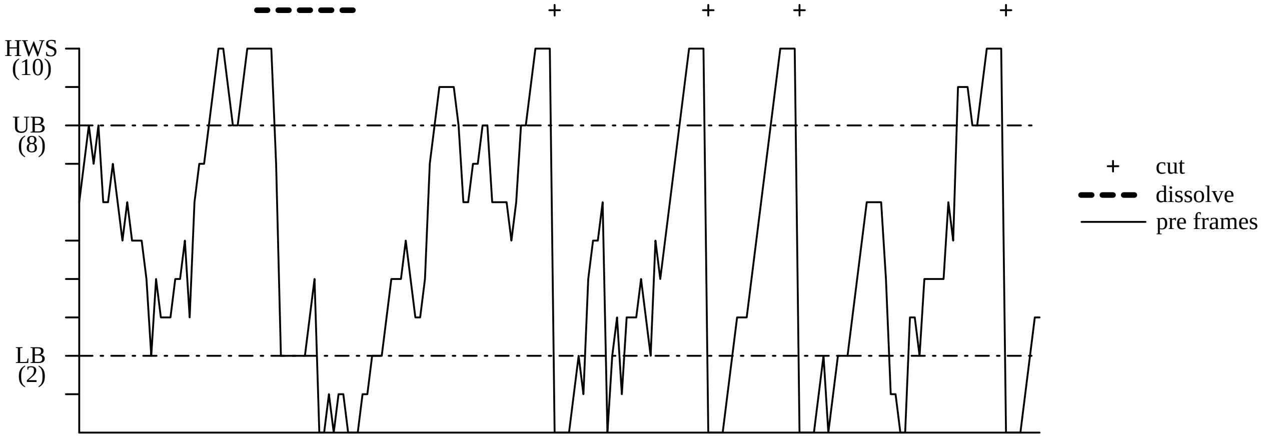


Figure 3: Plot of the number of pre-frames in the top half of the ranked results for a 200-frame interval. The five transitions present in this interval are indicated above the plot. The parameters used for HWS, the upper threshold (UB) and the lower threshold (LB) are listed between parentheses.

2.1 Abrupt Transitions

As the current frame approaches a cut, frames from the second shot enter the window. All the pre-frames are from the first shot (shot A), while some of the post-frames belong to the second shot (shot B). However, the current frame is still from shot A, so after computing the similarity to the query, we generally find the shot B frames ranked lower than the shot A frames. As a result, there is a rise in the number of pre-frames in the top $\frac{N}{2}$.

When the current frame is the last frame of shot A, all pre-frames are from shot A, and all post-frames are from shot B. At this point, the number of pre-frames in the top $\frac{N}{2}$ reaches a maximum, since the shot A frames will all be ranked above the shot B frames. This can be seen in Figure 2. Once the current frame moves into the next shot, the example image will belong to shot B, so the situation is reversed: the number of post-frames in the top $\frac{N}{2}$ exhibits a sharp rise, while the number of pre-frames drops to near zero.

In Figure 3, the variation in the number of pre-frames in the top $\frac{N}{2}$ results is shown over 200 frames of a clip. The location of the four cuts and one dissolve in this interval are shown at the top of the figure. Cuts are accompanied by a sharp drop in the number of pre-frames at the top of the ranked list.

2.2 Gradual Transitions

The first transition in Figure 3 is a gradual transition. We see that frame ranks within the moving window are also affected by this transition, although to a lesser degree than by the cuts. Our technique can be modified to additionally detect gradual transitions. When the moving window traverses a gradual transition, we observe three phases:

1. **Post-frames enter transition, but the current frame is not yet in transition:** The number of pre-frames ranked in the top $\frac{N}{2}$ rises, since the transition frames are less similar to the example frame than the non-transition frames.
2. **Current frame in transition:** The number of pre-frames ranked in the top $\frac{N}{2}$ slowly decreases.
3. **Current frame exits transition:** The number of pre-frames ranked in the top $\frac{N}{2}$ falls significantly, since the pre-frames—which are still within in the transition—are less similar to the example than the post-frames.

The three phases of this transition can be seen from the plot at the top of Figure 3. Considering the number of pre-frames in the top $\frac{N}{2}$ results, we see that this number increases towards the peak as we approach the start of a transition. During the transition, the number returns to moderate values. As the current frame exits the transition, the number of pre-frames drops to a minimum; the value gradually increases again as the transition frames leave the half-window preceding the current frame. We can detect gradual transitions by monitoring for this characteristic pattern.

In general, detection of gradual transitions is more difficult than detection of abrupt transitions. In contrast to cuts, gradual transitions do not have a sharp division between shots, and adjacent frames within a gradual transition usually differ by a small amount. To accentuate the differences between the frames, we could sample the stream at a lower rate. This would, however, reduce our precision: if we use every nth frame, we can only resolve the shot boundary to within n frames.

In our experiments, we use all frames, employing each in turn as a query example. However, we omit the closest few frames bordering the current frame from the collection. This leaves a gap, which we refer to as the *Demilitarised Zone* (DMZ) on either side of the current frame, as illustrated in Figure 4. The DMZ effectively determines the difference between the example frame and the most similar frame from the window; a large value for the DMZ will blur the distinction between frames of shot A and frames of shot B.

2.3 Algorithm Details

In this section, we describe the details of our shot boundary detection scheme. We begin by defining the algorithm parameters, and continue with a description of the detection steps for transitions.
In our discussion, we refer to four primary parameters:

Lower Bound (LB): This is the lower threshold. Once the number of pre-frames falls below this level, a possible transition is detected as shown in Figure 3.

Upper Bound (UB): This is the upper threshold. Once the number of pre-frames rises above this level, a possible transition is detected as shown in Figure 3.

Half Window Size (HWS): The number of frames from either side of the current frame that are contained within the moving window. Since we examine the top $\frac{N}{2}$-ranking frames, we use this number as the main parameter, rather than the full window size (N) itself. This is shown in Figure 4.

Demilitarised zone depth (DMZ): This is the size of the gap between the current frame and the nearest frame that is part of the moving window. See Figure 4 for an example.

We continue next with a discussion of how abrupt transitions are detected using the moving window and these parameters.

Detection of Cuts

To detect abrupt transitions, we monitor the number of pre-frames in the top $\frac{N}{2}$ results as each frame is examined. We refer to this number as `NumPreFrames`. We also measure the slope of the `NumPreFrames` curve. This is normally small, that is, around two.

As we near an abrupt transition, `NumPreFrames` rises quickly and passes the upper bound (UB). Once we pass the transition, `NumPreFrames` falls sharply below the lower bound (LB). The slope reflects this by taking on a large positive value, followed quickly by a large negative value. This behaviour can be observed in

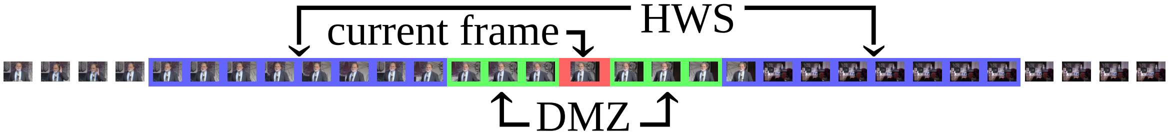


Figure 4: Moving query window with a half window size (HWS) of 8, and a demilitarised zone (DMZ) of three frames on either side of the current frame; the eight frames preceding and the eight frames following the current frame form a collection, against which the current frame is used as a query example.

Figure 3. We report a possible cut if `NumPreFrames` exceeds UB, then falls below LB in the space of two frames.

In some cases, the slope condition may be satisfied inside a shot where no transition exists. This may occur where, for example, a traffic light changes from red to green; all "red" frames will be ranked together and separately from all "green" frames, causing the slope to exhibit the behaviour seen for cuts. To avoid incorrectly declaring a cut in such cases, we impose the condition that there must be a large difference between the pre- and post-frames. This is achieved by requiring the average distance of the top $\frac{N}{2}$ frames to the query image to be less than half the average distance of the bottom $\frac{N}{2}$ frames from the same query image.

All comparisons so far have been relative. To further reduce the occurrence of false positives, we introduce an absolute threshold for the distance between the last pre-frame and the first post-frame. This is expressed as a proportion of the maximum distance possible between two frames using the current feature and histogram representation. We fixed this threshold at 25% of the maximum possible distance.

To summarise, a cut is reported if the following conditions are satisfied:

1. The `NumPreFrames` slope takes on a large negative value;
2. The top $\frac{N}{2}$ frames are significantly different from the bottom $\frac{N}{2}$ frames; and,
3. The last pre-frame and the first post-frame are significantly different.

Since these conditions are not synchronous, we allow them to be satisfied at any point within an interval of four frames. For example, the first condition may be met at frame n, and the second condition may be met at frame $n+2$. If all three conditions hold, we record a cut with the current frame being the first frame of the new shot.

Detection of Gradual Transitions

Detection of gradual transitions is more difficult than detection of abrupt transitions, and we need to employ more heuristics. We experimented only briefly with gradual transition detection in this work and, as we show later, our detection of gradual transitions is relatively ineffective. We plan further experiments to determine the variation of parameters required for improved detection of such transitions.

We noted in Section 2.2 that during a gradual transition, `NumPreframes` often rises to high levels, then drops to low values, and remains there for a some time before rising to return to typical levels. We are alerted to a possible gradual transition when we detect that `NumPreframes` has remained low for several frames. We regard the current frame as marking the end of the transition. To identify the beginning of the transition, we look back to find the location of the first phase of the gradual transition, that is, the point where `NumPreFrames` first rises to a high level designated by the upper bound (UB). Finally, we measure how long `NumPreframes` remains high. If this is more than a threshold value, we declare a gradual transition.

In summary, a gradual transition is reported if the following conditions are met:

1. The `NumPreFrames` slope remains low for several frames, and
2. before this, `NumPreFrames` increases to a high level, and remains consistently high over several frames.

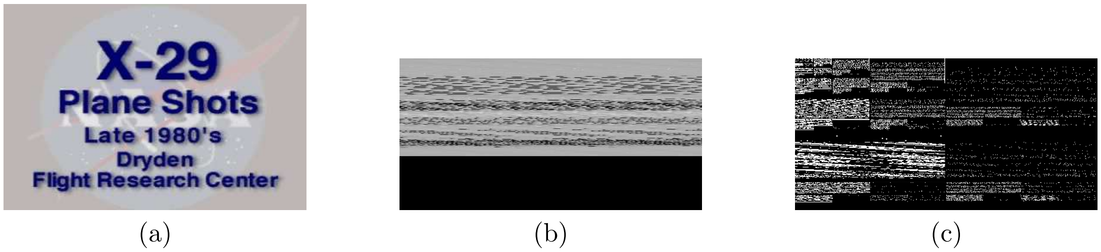
If both conditions are met, we record a gradual transition starting at the point `NumPreFrames` first exceeds the upper bound, and ending at the current frame.

3 Selection of Features and Parameters

To compare different features and identify suitable parameters, the moving query window algorithm was ap-

Parameter	Acronym	Range start	Range end	Step size
Half window size ($\frac{N}{2}$)	HWS	6	30	2
Lower bound	LB	1	4	1
Upper bound	UB	$HWS - 4$	$HWS - 1$	2
De-militarised zone	DMZ	0	10	2

Table 1: The ranges of values used for the parameters of the shot boundary detection algorithm.



(a) (b) (c)

Figure 5: (a) Input frame of dimensions 352×240. (b) Frame Y (brightness) data placed in a super-frame of dimensions 512×256, with the unused portion of the super-frame being set to black. (c) Transformed super-frame; the data corresponding to the unused portion of the super-frame does not contain any information, and is discarded.

plied to detect shot boundaries on a subset of the TREC-10 evaluation set comprising eleven clips, containing a total of 996 cuts and 406 gradual transitions. Each feature was evaluated using parameters in the ranges shown in Table 1.

The effectiveness of the segmentation operation is evaluated using the standard information retrieval measures of recall and precision. Precision represents the fraction of detected transitions that match the reference data:

$$P = \frac{\text{Transitions correctly reported}}{\text{Total transitions reported}}$$

Recall measures the fraction of all reference transitions that are correctly detected:

$$R = \frac{\text{Transitions correctly reported}}{\text{Total reference transitions}}$$

These two measures can be used for both abrupt and gradual transitions. To evaluate how well reported gradual transitions overlap with reference transitions, TREC-11 introduced the measures *Frame Precision* (FP) and *Frame Recall* (FR).

$$FP = \frac{\text{Frames correctly reported in detected transition}}{\text{Frames reported in detected transition}}$$

$$FR = \frac{\text{Frames correctly reported in detected transition}}{\text{Frames in reference data for detected transition}}$$

3.1 Features

We used one-dimensional global histograms using the HSV, Lab, and Luv colour spaces, and a fourth feature derived from the Daubechies wavelet transform of the frames. Preliminary experiments using three-dimensional colour histograms have produced slightly better results but we do not describe them here.

The native colour space of the MPEG compressed video stream is $\mathrm{YC_bC_r}$. The wavelet-based feature for each frame was generated by computing the six-tap Daubechies wavelet transform coefficients from the $\mathrm{YC_bC_r}$ colour data. When calculating the wavelet transform using the Mallat algorithm, the data dimensions are halved after each pass [11, 22]. Thus, we can perform four passes on frames with dimensions 352×240, ending at 22×15, which cannot be transformed further. Frames with dimensions 320×240 can also be transformed four times (ending at 20×15), while frames with dimensions 352×288 can be transformed five times (ending at 11×9).

All clips used in TREC-11 had dimensions 352×240; nevertheless, we should cater for different frame sizes. To allow comparison of equivalent wavelet scales for different-size frames without the expense of resizing, we rearrange the frame data to fit into a *super-frame* with dimensions that are a power of two. For example, the pixel data from a 352×240 frame is inserted into a super-frame of dimension 512×256, as shown in Figure 5. The unused portion of the super-frame

Abrupt	Bins/subbands	HWS	LB	UB	DMZ	**Gradual**	Bins/subbands	HWS	LB	UB	DMZ
HSV	384	20	4	18	0	HSV	48	20	4	18	4
Lab	1536	26	3	24	4	Lab	192	22	3	20	4
Luv	1536	10	4	8	0	Luv	1536	22	3	20	4
RWav	5	10	3	8	4	RWav	4	20	3	18	4

Table 2: The best set of parameters varies by feature and transition type; gradual transitions are generally best detected with a DMZ of four. The effect of varying the DMZ is less pronounced for cut detection. While in some cases the best results are obtained with non-zero DMZ, the difference with the DMZ=0 results is insignificant.

is zero-filled, and the transform data for this portion is later discarded. With the new frame dimensions, eight transform passes are possible, ending with the data dimensions 2×1. We call this feature the wavelet transform on re-ordered data or RWav.

Of the feature combinations tried, RWav proved to be the most effective for detecting cuts, and Luv was the best feature to use for detecting gradual transitions. The simple HSV feature also proved to be effective, with recall and precision comparable to those of the best features. The amount of processing required to extract the HSV data from the video stream is much less than the other features. This low extraction cost may make HSV the most practical choice of feature for a commercial system.

We found that while using only the luminance component of the colour data trebles processing speed, detection effectiveness is significantly reduced. An exception is the RWav feature, where the effectiveness in detecting cuts with only luminance (Y) information is relatively unchanged from the full $\mathrm{YC_bC_r}$ version.

Although the global colour features generally produced good results, they often failed to detect cuts between two shots of the same scene where the camera followed an object moving rapidly against a noisy background. This type of cut is often easily detected by the wavelet (RWav) feature, which preserves spatial layout information.

Conversely, the wavelet feature is sensitive to small changes in the frame content and performs relatively poorly at finding gradual transitions. However, the high-frequency data—corresponding to detail in the image—plays an important part in cut detection; we observe the best results when using the first four or five transform sub-bands. Further increasing the number of sub-bands inserts too much detail, and adversely affects performance. The volume of feature data stored per frame also quadruples for each additional sub-band, and so a performance penalty is also incurred.

3.2 Other Parameters

The best choice of algorithm parameters varied for different features and for the two transition types; these are listed in Table 2.

We found that transitions are best detected with a half window size (HWS) of approximately 18 or 20 frames. It is likely that the optimal value for HWS will vary depending on the content of the footage being examined; long, slow transitions will favour larger values of HWS. We have not performed in-depth experiments to test this supposition.

The lower bound (LB) and upper bound (UB) determine the relative priorities of recall and precision. Decreasing LB towards zero generally increases precision at the cost of recall. This effect is relatively minor for cut detection, since in most cases, `NumPreFrames` actually reaches zero at the cut boundary. Detection of gradual transitions is sensitive to the LB parameter, and our best preliminary results were obtained with an LB of 3 or 4.

There is a close relationship between the best choice of frame gap (DMZ) and the type of transition to be detected. Cuts are generally best detected with no gap at all (DMZ=0), while gradual transitions are best found with a small gap (DMZ=4). As with HWS, we believe the best value is somewhat dependent on the type of video footage being processed and we plan further experiments to verify this.

Histogram distances were calculated using the Manhattan, cumulative Manhattan, histogram intersection, and Euclidean measures. The Manhattan distance measure produced the best results. The relatively high computation cost of the Euclidean distance measure makes it unattractive for use in video.

4 TREC-11 Results

In TREC-11, groups were permitted to submit a maximum of ten runs. The evaluation set consisted of eighteen video clips, with 1 466 cuts and 624 gradual transi-

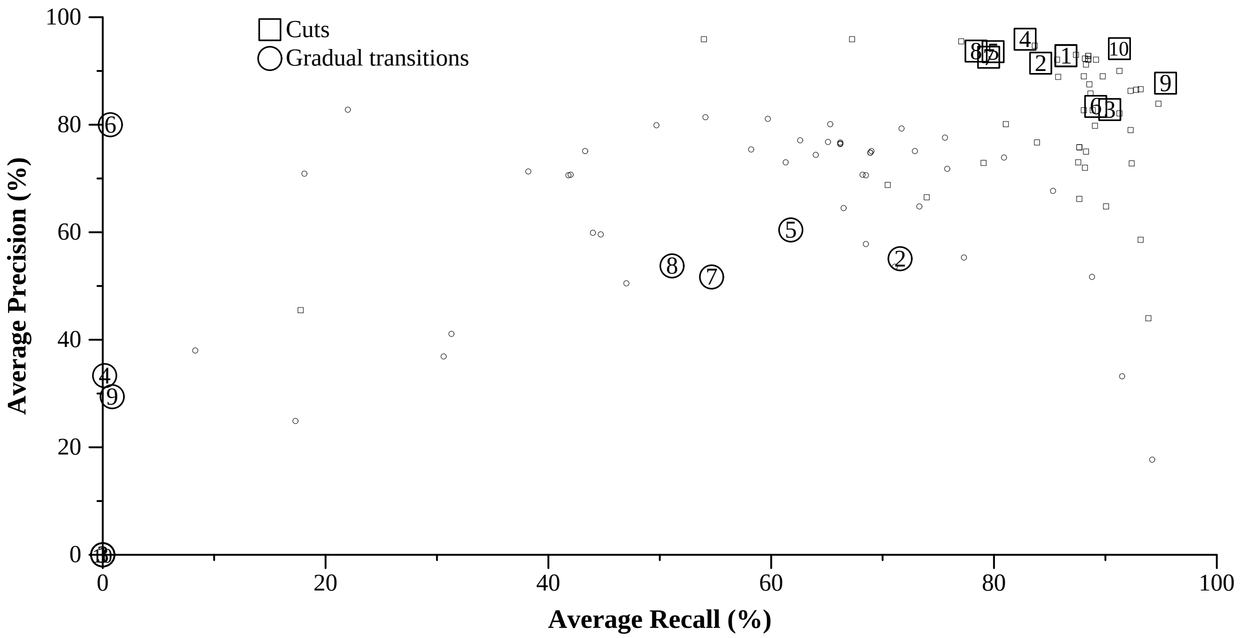


Figure 6: Performance of the moving query window for cuts and gradual transitions on the TREC-11 shot boundary detection task.

Run	Feature type	Colour space	Vector length	Half window size (HWS)	Lower Bound (LB)	Upper Bound (UB)	Demilitarised Zone (DMZ)
1	Colour histogram	HSV	384	20	4	18	0
2	Colour histogram	HSV	96	20	3	16	4
3	Colour histogram	Lab	1536	12	6	10	0
4	Colour histogram	Lab	1536	26	3	24	0
5	Colour histogram	Lab	1536	26	3	24	4
6	Colour histogram	Luv	1536	10	4	8	0
7	Colour histogram	Luv	1536	22	3	20	4
8	Colour histogram	Luv	1536	26	3	24	4
9	Wavelet (5 scales)	$\mathrm{YC_bC_r}$	1176	10	3	8	0
10	Wavelet (5 scales)	$\mathrm{YC_bC_r}$	1176	20	3	18	0

Table 3: Parameters used for each submitted run.

tions. We submitted runs using the parameters shown in Table 3.

The recall and precision levels for cuts and gradual transitions are shown in Figure 6. The numbered squares and numbered circles correspond to moving query window results for abrupt and gradual transitions respectively. Results submitted to TREC-11 by other groups are indicated by the small squares and circles. Similarly, Figure 7 shows the performance of our approach and that of other systems when detecting gradual transitions, as measured by Frame Recall and Frame Precision.

The moving query window showed good results for cut detection and poor results for gradual transitions. Algorithm parameters that performed well on abrupt transitions performed poorly on gradual transitions and vice-versa. Run ten produced the best results for detection of abrupt transitions, but failed to detect any gradual transitions.

5 Summary

We have introduced a new moving query window approach that applies the CBIR concepts of querying

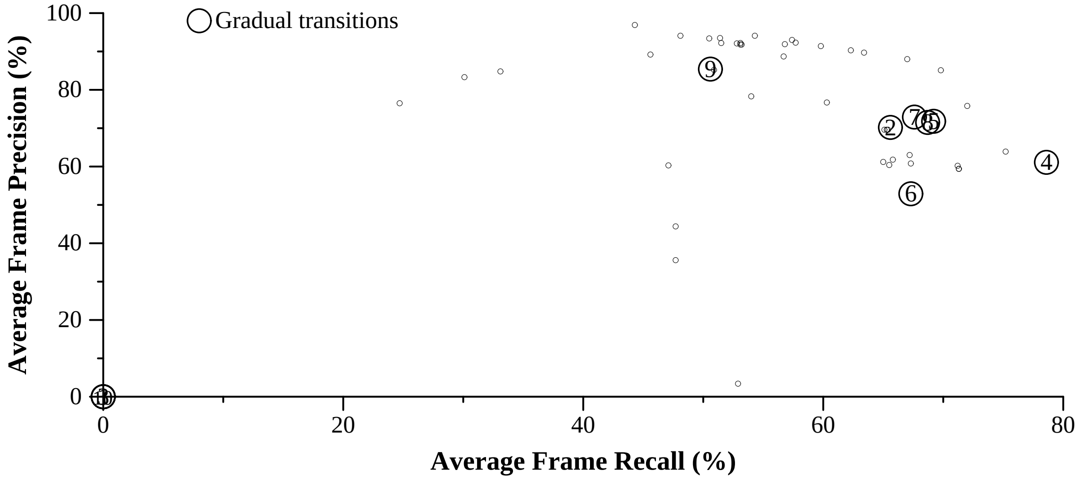


Figure 7: Performance of the moving query window for gradual transitions on the TREC-11 shot boundary detection task, as measured by Frame Recall and Frame Precision.

by example image and ranked results to detect shot boundaries in video. We have described the parameters of the algorithm, and discussed the steps used to determine the presence of transitions.

We have identified several areas where modifications could lead to improved efficiency and effectiveness. One improvement could be to preserve some information about the spatial colour distribution in the colour features; this can be done by using local rather than global colour histograms.

Our algorithm is sensitive to sudden changes in the video brightness level, photographic flashes, and the appearance and disappearance of textual captions. This sensitivity can be reduced by integrating existing work on detectors for such phenomena [15, 24].

Populating the window requires that the algorithm begin operation from the $\frac{N}{2}$th frame, and end $\frac{N}{2}$ frames before the end. Transitions occurring within the excluded regions cannot be detected. Other methods must be used to handle the approximately half-second of footage at the extremities of each clip.

The routines for detection of cuts and gradual transitions are independent, and may interfere destructively. Since the conditions to be met for cuts are stricter than those for gradual transitions, we made a decision to give precedence to cuts; if a cut has already been detected in the transition interval, the gradual transition is not reported. In addition, we have not experimented in detail with the detection of gradual transitions; we plan future work on selecting heuristics for this domain.

Overall, we have shown that our method produces competitive results. In particular, we have shown that the RWav feature, derived from the Daubechies wavelet transform of the frame data, produces excellent cut detection results. Our parameters were based on experiments using a subset of the TREC-10 evaluation set, and are therefore not necessarily optimal for the TREC-11 evaluation set. We expect that results can be improved through experimentation with dynamic thresholds and other adaptive parameters.

References

[1] F. Arman, A. Hsu, and M.-Y. Chiu. Image processing on encoded video sequences. *Multimedia Systems*, 1(5):211–219. Springer-Verlag, March 1994.

[2] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, 5(2):122–128. SPIE, April 1996.

[3] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262. Kluwer, July 1994.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, and D. Steele. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32. September 1995.

[5] A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *Proceedings of the ACM In-*

ternational Conference on Multimedia, pages 357–364, San Francisco, California, USA, 15–20 October 1994.

[6] D. Harman. Overview of the Second Text REtrieval Conference (TREC-2). *Information Processing & Management*, 31(3):271–289. Elsevier, May/June 1995.

[7] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'98)*, pages 218–227, New York, USA, 24-27 August 1998.

[8] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of the ACM-SIGMOD International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'95)*, pages 277–286, Los Angeles, California, USA, 6–11 August 1995.

[9] R. Kasturi and R. Jain. *Dynamic Vision*, In *Computer Vision: Principles*, pages 469–480. IEEE Computer Society Press, 1991.

[10] R. W. Lienhart. Comparison of automatic shot boundary detection algorithms. *Proceedings of the SPIE; Storage and Retrieval for Still Image and Video Databases VII*, 3656:290–301. December 1998.

[11] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693. July 1989.

[12] J. Meng, Y. Juan, and S.-F. Chang. Scene change detection in a MPEG compressed video sequence. *Proceedings of the SPIE; Digital Video Compression: Algorithms and Technologies*, 2419:14–25. April 1995.

[13] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-search for video appearances. *Visual Database Systems*, 2:113–127. Elsevier, 1992.

[14] M. Pickering and S. M. Rüger. Multi-timescale video shot-change detection. In *NIST Special Publication 500-250: Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 275–278, Gaithersburg, Maryland, USA, 13–16 November 2001.

[15] G. Quénot and P. Mulhem. Two systems for temporal video segmentation. In *Proceedings of the European Workshop on Content Based Multimedia Indexing (CBMI'99)*, pages 187–194, Toulouse, France, 25–27 October 1999.

[16] A. Smeaton, P. Over, and R. Taban. The TREC-2001 video track report. In *NIST Special Publication 500-250: Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 52–60, Gaithersburg, Maryland, USA, 13–16 November 2001.

[17] J. R. Smith, S. Srinivasan, A. Amir, S. Basu, G. Iyengar, C. Y. Lin, M. R. Naphade, D. B. Ponceleon, and B. L. Tseng. Integrating features, models, and semantics for TREC video retrieval. In *NIST Special Publication 500-250: Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 240–249, Gaithersburg, Maryland, USA, 13–16 November 2001.

[18] J. Sun, S. Cui, X. Xu, and Y. Luo. Automatic video shot detection and characterization for content-based video retrieval. *Proceedings of the SPIE; Visualization and Optimisation Techniques*, 4553:313–320. September 2001.

[19] S. M. M. Tahaghoghi, J. A. Thom, and H. E. Williams. Multiple-example queries in content-based image retrieval. *Proceedings of the Ninth International Symposium on String Processing and Information Retrieval (SPIRE'2002)*, pages 227–240, Lisbon, Portugal, September 2002.

[20] Text REtrieval Conference (TREC), National Institute of Standards and Technology, Gaithersburg, Maryland, USA.
URL: `http://trec.nist.gov`.

[21] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. Video manga: Generating semantically meaningful video summaries. In *Proceedings of the ACM International Conference on Multimedia*, pages 383–392, Orlando, Florida, USA, 30 October – 5 November 1999.

[22] J. R. Williams and K. Amaratunga. Introduction to wavelets in engineering. *International Journal for Numerical Methods in Engineering*, 37(14):2365–2388. John Wiley & Sons, 1994.

[23] B. L. Yeo and B. Liu. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533–544. December 1995.

[24] D. Zhang, W. Qi, and H. J. Zhang. A new shot boundary detection algorithm. *Lecture Notes in Computer Science; Proceedings of the Second IEEE Pacific Rim Conference on Multimedia (PCM'2001)*, 2195:63–70. Beijing, China, 24–26 October 2001.

[25] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28. Springer-Verlag, June 1993.