The Moving Query Window for Shot Boundary Detection at TREC-12

Timo Volkmer 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

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

Abstract

Digital video is widely used in multimedia databases and requires effective retrieval techniques. Retrieval effectiveness relies on content-based indexing, but video is not as self-descriptive as text documents. Shot boundary detection is a common first step in analysing video content. The effective detection of gradual transitions is an especially difficult task. Building upon our past research work, we have designed a novel decision stage for detection of gradual transitions. In this paper, we describe our moving query window method and describe its performance in context of the TREC-12 shot boundary detection task.

1 Introduction

Humans perceive their environment almost entirely by audio-visual means. Video captures both acoustic and visual information, and with the increasing computational power and network bandwidth of modern information systems, applications using digital video have become widespread. We believe that video will continue to gain importance as an information carrier.

Most video applications share the need for efficient retrieval of archived video data. To retrieve video footage effectively, we must know its content and index it. This is commonly performed by annotating sections sequentially with textual information [12]. This is a tedious and expensive process. Moreover, human observation of video is subjective and prone to error. Automatic techniques for video content analysis are required.

The basic semantic element in video is the *shot* [7], formed by a sequence of often similar frames. Selected frames (*key-frames*) can be indexed to represent each shot and allow retrieval [6]. A query using example frames may then return all shots containing similar key-frames. The first step in this process is often *shot boundary detection*, where the video content is separated into distinct shots.

We distinguish between different types of the transitions that delimit shots. These are classified as abrupt transitions or *cuts*, and *gradual transitions*, which include fades, dissolves and spatial edits [9]. Informational video tends to contain more cuts, whereas entertainment material is more likely to be edited using fades, dissolves, and other gradual transitions.

According to Lienhart [14] cuts, dissolves, and fades account for more than 99% of all transitions across all types of video. The TREC-10 and TREC-11 video collections [24, 25] support this observation.

The cut detection quality of existing systems is comparable to the quality of human detection [2]. However, gradual transitions are more difficult to detect using automated systems [17].

1.1 Related work

Research on transition detection in digital video can be categorised into methods that use compressed video and methods that use uncompressed video. An up-to-date overview of existing techniques is provided by Koprinska et al. [13].

Techniques in the compressed domain use one or more features of the encoded footage, such as *Discrete Cosine Transform* (DCT) coefficients, *macro blocks* (MB), or *motion vectors* (MV) [4, 19, 30]. These algorithms are more efficient because the video does not need to be

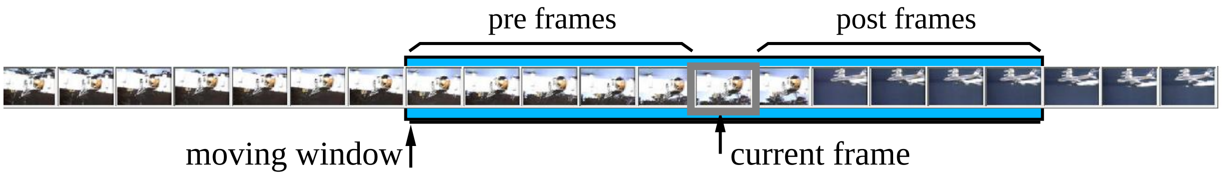

Figure 1: Moving query window with a half window size (HWS) of 5; the five frames before and the five frames after the current frame form a collection on which the current frame is used as a query example.

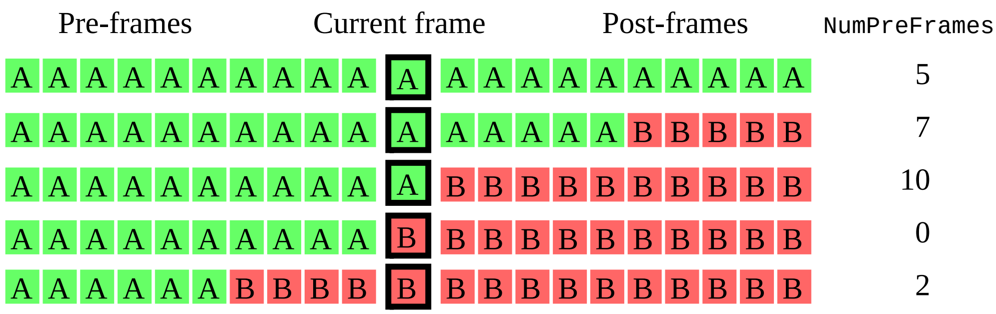

Figure 2: As the moving window traverses an abrupt transition, the number of pre-frames in the $\frac{N}{2}$ frames most similar to the current frame varies significantly. This number (`NumPreFrames`) rises to a maximum just before an abrupt transition, and drops to a minimum immediately afterwards.

fully decoded. However, using the encoded features directly can result in lower precision [5]. The exact transition boundaries may not be determinable, or gradual transitions may not be distinguishable from object movement [13].

Most approaches working on uncompressed video use frame difference as a measure for shot boundary detection. The difference between successive frames is usually small within one shot. When a sufficient dissimilarity between neighbouring frames can be detected, this is interpreted as a cut. The same scheme is applied cumulatively for gradual transition detection. There are several methods to measure the difference between frames.

Pixel-by-pixel comparison is an obvious approach; here, the number of changing pixels between frames is counted. While this method shows good results [5], it is computationally intensive and sensitive to camera motion, camera zoom, and noise.

The majority of research groups use histograms to represent frame content. Differences between frames are calculated using vector-distance measures [32]. Global histograms suffer from their lack of spatial information. Several researchers try to overcome this by introducing local histograms [18, 26] or adding other techniques such as edge-detection [15, 21].

Approaches that use clustering algorithms [8, 16] classify frames into two categories: *scene change* and *no scene change*. Adjacent frames from the former cluster are then marked as gradual transitions and all others from the same cluster are detected as cuts.

Many researchers have proposed methods, based on transition modelling. These employ mathematical models of video data to represent different types of transitions, and often work without the need for thresholds [10, 31]. Transitions are identified based on similarity to the underlying mathematical model. Koprinska et al. [13] report these approaches to often be sensitive to object and camera motion.

Looking at recent work of TREC participants, histograms seem to be the favoured way to represent feature data. Adams et al. [1] propose a video retrieval system which employs a combination of three-dimensional RGB colour histograms and localised edge gradient histograms for shot boundary detection. Recent frames are held in memory to compute adaptive thresholds.

The system proposed by Hua et al. [11] uses global his-

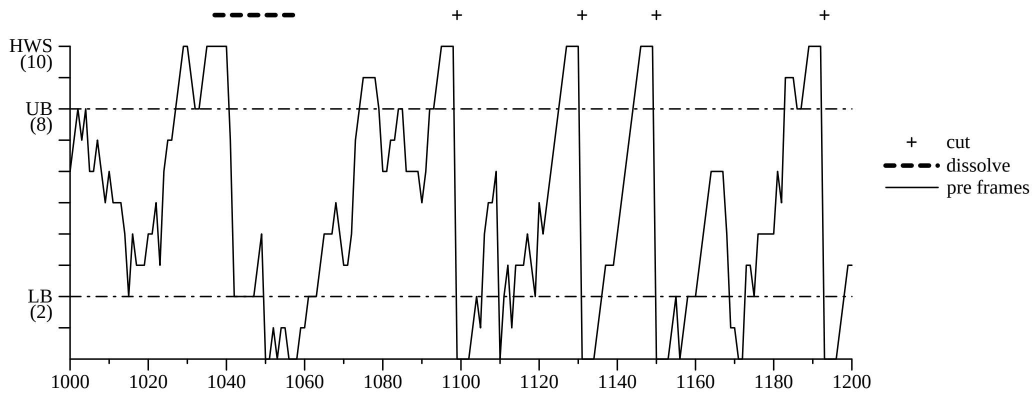

Figure 3: Plot of the number of pre-frames in the top half of the ranked results for a 200-frame interval. The five transitions present in this interval are indicated above the plot. The parameters used for HWS, the upper threshold (UB) and the lower threshold (LB) are listed between parentheses.

tograms in the RGB colour space. Pickering et al. [20] use a detection algorithm which employs localised RGB colour histograms. Each frame is divided into nine blocks and the median between the nine block distances is computed. A transition is detected when the median distance exceeds a fixed threshold.

The system proposed by Quénot et al. [22, 23] uses direct image comparison for cut detection. To reduce false positives, motion compensation is applied before image comparison. A separate flash detection module is used to further reduce false positives. Gradual transitions are detected by checking whether the pixel intensity in adjacent frames roughly follow a linear, non-constant function.

Wu et al. [29] propose a shot boundary detection algorithm which calculates frame-to-frame difference based on luminance information and histogram similarity in the RGB colour space. Flash and motion detectors are used to reduce false positives.

In the next section, we explain our approach to video segmentation. This is an extension of our work first presented at TREC-11 [25]. In Section 3, we discuss features and parameters used. Section 4 reviews the results of our algorithm on the TREC-12 shot boundary evaluation task. We conclude with Section 5, discussing possible improvements and future work.

2 The moving query window technique

Our algorithm applies the concept of query-by-example (QBE), popular in content-based image retrieval [27], to shot boundary detection. The observation that all transitions except cuts stretch over several adjacent frames suggests that we ought to evaluate a set of frames at once. To cater for this, we employ a *moving query window*, consisting of two equal-sized half windows on either side of the current frame.

As shown in Figure 1, the current frame is not part of the actual window. It is used as the query example against which the other frames of the query window can be compared. We refer to the frames forming the preceding half window as *pre-frames*, and to the frames following the current frame as the *post-frames*.

We evaluate frame similarity by employing one-dimensional[1] global colour histograms to represent frame content. We calculate inter-frame distances using the Manhattan—also called cityblock—distance measure [3]. The difference between the current frame and its surrounding frames is usually small. This changes when a transition is passed.

2.1 Abrupt transitions

To detect cuts, we apply a ranking approach as described by Tahaghoghi et al. [27]. We rank the frames within the moving query window based on their similarity to the current frame. The frame most similar to the current frame will be ranked highest.

Figure 2 shows how a cut can be detected using similarity ranking. Shortly before the current frame passes a cut, the half window holding the pre-frames is entirely filled with frames of the previous shot (Shot A). Some

[1] We evaluate only the V colour component in the HSV colour space.

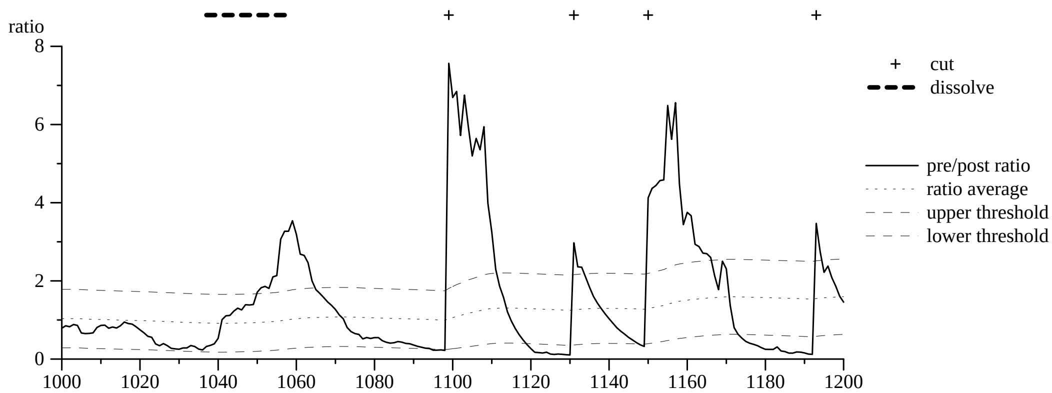

Figure 4: Ratio between the average frame distances to query-example of pre-frames and post-frames (pre/post ratio). Plotted for a 200-frame interval. We apply dynamic upper and lower thresholds, calculated using a moving average and standard deviation.

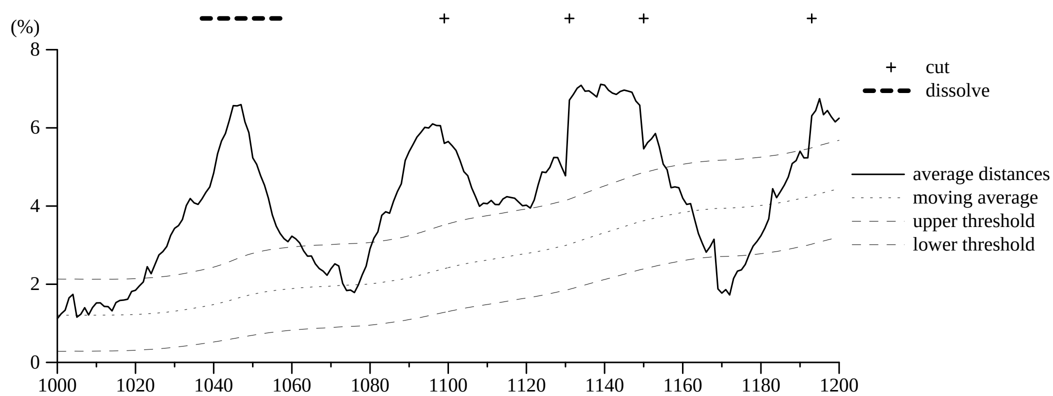

Figure 5: Average frame distances to query example, computed over the entire query window. This is shown for the same 200-frame interval as in Figure 4. The upper and lower thresholds are calculated based on moving average and standard deviation.

of the post-frames belong to the second shot (Shot B). Since the current frame still belongs to Shot A, frames of Shot B will be ranked lower than those of Shot A. When the last frame of Shot A becomes the current frame, all pre-frames will be Shot A frames, whereas all post-frames will be from Shot B. As a result, the number of pre-frames ranked in the top half window reaches a maximum.

This effect will be reversed when the query window advances by one frame and the current frame is the first frame of Shot B. Here, the number of pre-frames ranked in the top half window drops significantly to near zero.

The graph in Figure 3 shows the variation in the number of pre-frames ranked in the top half of the query window. The diagram shows a 200-frame interval; four cuts and one gradual transition are marked above the graph. Cuts are clearly indicated by a rapid decrease in the number of pre-frames from above the *Upper Bound* (UB) to a value below the *Lower Bound* (LB).

2.2 Gradual transitions

As can be seen from Figure 3, a gradual transition can not be as clearly identified by the ranking approach as a cut. This is mainly because gradual transitions stretch over several adjacent frames. This observation led us to develop a different approach for detecting gradual transitions.

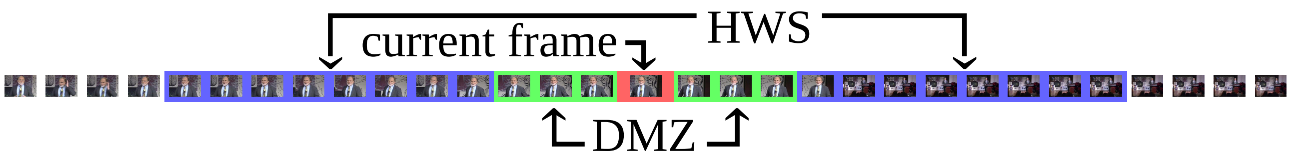

Figure 6: Moving query window with a half window size (HWS) of 8, and a demilitarised zone (DMZ) of three frames on either side of the current frame; the eight frames preceding and the eight frames following the current frame form a collection, against which the current frame is used as a query example.

We monitor the average distance of frames within the query window on either side of the current frame. These values are used to build the ratio of differences between pre-frames and post-frames (pre/post ratio). Figure 4 shows the pre/post ratio for the same 200-frame interval as previously used in Figure 3.

Gradual transitions are indicated by a peak in the pre/post ratio, usually at the end of the transition. The slopes of these peaks are often moderately steep, as opposed to a very quick rise found for cuts. We also calculate the average sum of distances in the entire query window, the *average frame distance*.

Figure 5 shows the average frame distance. The curve has a maximum in the middle of the gradual transition, which is typical for a dissolve. We can identify gradual transitions by monitoring for these patterns using peak detection and plateau detection algorithms.

The dashed lines in Figure 3 and Figure 5 mark the adaptive lower and upper thresholds we use for decision making.

Our algorithm allows a number of frames bordering the current frame to be omitted from the collection. This results in a gap on either side of the current frame, which we refer to as the *Demilitarised Zone* (DMZ). As shown in Figure 6, the DMZ may be applied to blur the distinction between frames. Although the frames which form the demilitarised zone are disregarded in the evaluation, we consider them to be part of the query window. The *Query Window Size* (QWS) is the number of frames contained in both half windows, the frames of the demilitarised zone plus the *Current Frame* (CF).

2.3 Algorithm details

We describe more details of our algorithm in this section. We first define the major algorithm parameters, and then discuss the detection steps for cuts and gradual transitions.

Lower Bound (LB): This is the lower threshold used for cut detection. As illustrated in Figure 3, a possible cut is indicated when the number of pre-frames falls below this threshold.

Upper Bound (UB): This upper threshold is used for cut detection in connection with LB. Whenever the number of pre-frames rises above UB, a possible cut is detected.

Half Window Size (HWS): This is the number of frames within the query window on either side of the current frame. The number of frames contained in the full query window is therefore $2 \times HWS$, as shown in Figure 6.

Demilitarised Zone depth (DMZ): This specifies the number of frames on each side of the current frame which are not evaluated as part of the query window. Figure 6 shows an example.

Threshold History Size (THS): This is the length of the buffer that we use to calculate the adaptive thresholds for gradual transition detection. We multiply the query window size by this factor.

The decision-making process for gradual transition detection differs from the process used for cut detection. This reflects the different nature of gradual transitions and abrupt transitions. We can employ a smaller part of the moving query window for detecting cuts. The parameters UB and LB are only used for cuts, whereas THS is only used for gradual transition detection. Half window size and DMZ are used in both decision stages, but are set to different values. Our algorithm performs both cut and gradual transition detection within a single pass.

Detection of cuts

As we advance through the video, we monitor the number of pre-frames that are ranked in the top half of all frames in the moving query window. We refer to this number as `NumPreFrames`. We also calculate the slope

Run	Vector length	HWS used for cuts/gradual transitions	Lower Bound (LB)	Upper Bound (UB)	Threshold History (THS)	DMZ used for cuts/gradual transitions
1	48	6/20	2	5	$3 \times QWS$	0/2
2	48	6/26	2	5	$3 \times QWS$	0/2
3	96	6/20	2	5	$4 \times QWS$	0/2
4	96	6/26	2	5	$8 \times QWS$	0/2
5	192	6/20	2	5	$3 \times QWS$	0/2
6	192	6/26	2	5	$4 \times QWS$	0/2

Table 1: Parameters used for each submitted run. Global colour histograms in HSV coulour space have been used for all runs. We employ only a subset of the entire query window for cut detection.

of the `NumPreFrames` curve. When we near an abrupt transition, `NumPreFrames` rises above the upper threshold (UB). After we pass a cut, `NumPreFrames` generally drops below the lower bound (LB). This is also reflected in the slope of the `NumPreFrames` curve. This slope takes on a large positive value before a cut, and a large negative value just after passing it.

In many video clips, variations in lighting conditions lead to false cut detection. This may occur, for example, when the camera focus follows an object from the shade into bright sunlight. To avoid such false positives, we evaluate the average distances of the top-ranked half-window and the bottom-ranked half-window to the current frame. The top-ranked frames must have less than half the average distance to the current frame than the bottom ranked frames.

Training experiments using the TREC-10 and TREC-11 shot boundary evaluation sets has shown that we can reduce the number of false positives further by requiring a significant difference between the last pre-frame and the first post-frame. We introduce an absolute threshold at 25% of the maximum possible frame distance to express this.

In accordance with the TRECVid definition that a cut may stretch over up to six frames [28], we allow a fuzziness of four consecutive frames for the above criteria to be satisfied. In summary, our algorithm reports a cut when the following criteria are fulfilled at any point within four consecutive frames:

1. The slope of the `NumPreFrames` curve has a large negative value,
2. The top ranked half window frames have less than half the average distance to the query frame than the bottom ranked frames, and
3. The last pre-frame is more than 25% different from the first post-frame.

If all these conditions are satisfied, we report a cut with the current frame being the first frame of the new shot.

Detection of gradual transitions

We have focused on applying the ranking approach to gradual transition detection in our recent experiments. We found it hard to determine a ranking pattern suitable for the effective detection of gradual transitions. This led us to develop a new approach for gradual transition detection, that can be employed within our existing algorithm.

Here, we monitor the pre/post ratio, as described in Section 2.2, and use a peak detection algorithm to find the local maximum in the curve whenever the upper threshold is exceeded, as shown in Figure 4. We hold the pre/post ratios of the past 60 frames in a history to detect the local minimum preceding the peak. We record this minimum as the start of a possible transition, and the local maximum as its end.

Gradual transition detection is performed after cut detection within a single pass. We check that no cut has previously been detected within the suspected gradual transition. Two heuristics are employed to further reduce false hits. We compute the area between the pre/post ratio curve and its upper threshold, as well as the area between the average frame distance curve and the upper threshold. Both values must exceed a certain fixed threshold. We heuristically determined a suitable value for this threshold by training on the TREC-10 and TREC-11 test sets. Peaks in the curves covering smaller areas are usually caused by normal scene activity.

We compute dynamic thresholds for peak detection using a moving average, calculated over a number of frames held in a *Threshold History*. We can set the size of this history, dependent on the query window size, by specifying the THS value. As mentioned in Section 2.3 the query window size is multiplied by this factor and

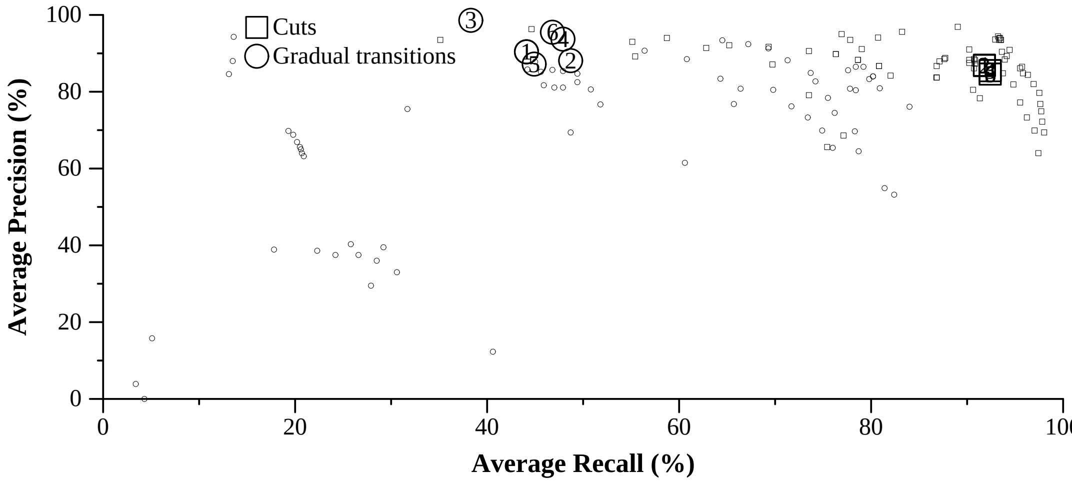

Figure 7: Performance of the moving query window for cuts and gradual transitions on the TREC-12 shot boundary detection task, measured by Recall and Precision.

the result taken as the threshold history size.
The actual thresholds are computed using the standard deviation of the curve. We have briefly experimented with incorporating the lower threshold to increase accuracy, but without significant success. We currently base the decision making on only the upper threshold.

3 Selection of features and parameters

We have tested our algorithm on the shot boundary evaluation sets of TREC-10 and TREC-11 [24, 25]. For submission to TREC-12, we used the parameters shown in Table 1.
The effectiveness of the segmentation process is evaluated using the standard information retrieval measures of recall and precision. Recall measures the fraction of all reference transitions that are correctly detected:

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

Precision represents the fraction of detected transitions that match the reference data:

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

These measures can be used for both abrupt and gradual transitions. With TREC-11 two additional measures were introduced to evaluate how well reported gradual transitions overlap with reference transitions. These are, *Frame Precision* (FP) and *Frame Recall* (FR), defined as:

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

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

3.1 Features

In the runs submitted to TREC-12, we have used one-dimensional global HSV colour histograms to represent frames.
The best results we have achieved so far use a feature derived from the Daubechies wavelet transform [27]. Despite this, we decided to further investigate using the HSV feature, as HSV feature data can be extracted at relatively small computational expense.
Table 1 shows all relevant parameter combinations used for the six runs submitted to TREC-12. Working towards our goal of a universal shot boundary detection algorithm, we have tried to keep parameter variations as small as possible.

3.2 Other Parameters

The main parameter settings used for the six submitted runs are shown in Table 1. The Lower bound and the upper bound are used only for cut detection. We have found that our approach performs best when setting HWS to 6, LB to 2 and UB to 5 for cut detection. The

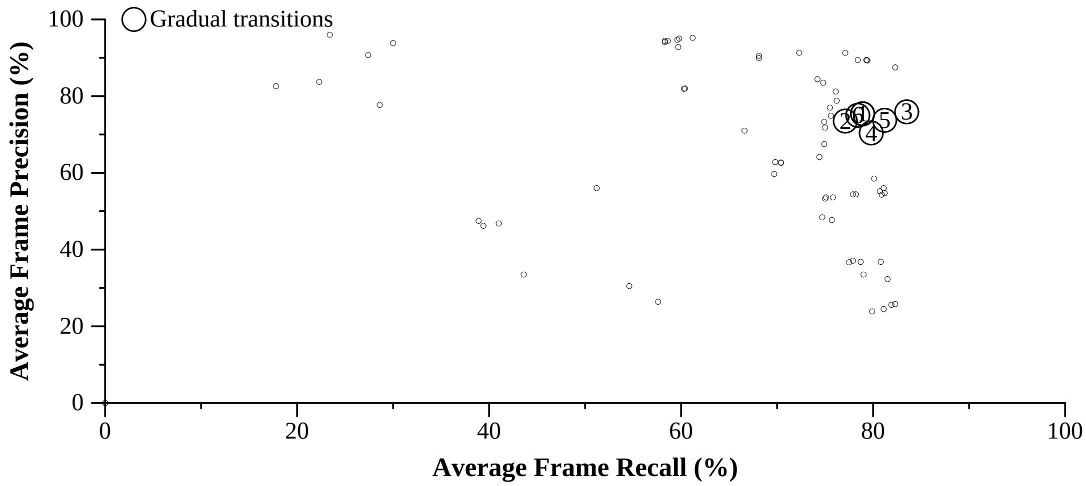

Figure 8: Performance of the moving query window for gradual transitions on the TREC-12 shot boundary detection task, as measured by Frame Recall and Frame Precision.

demilitarised zone was set to 0 for cut detection in all runs.

For gradual transition detection we applied different settings for HWS and DMZ. The lower bound and upper bound are not used in the decision stage for gradual transitions. To compute adaptive thresholds we maintain moving averages of frame distances and the pre/post ratio. The moving average is calculated over a larger number of consecutive frames. The number of frames is specified by the Threshold History Size (THS) value.

A larger value for THS causes the moving average to be smoother. We found that a small value for THS works well for video clips of lower quality which contain more noise. Otherwise this setting may result in a lower recall.

Frames that are part of the demilitarised zone are not considered when calculating average frame distance and the pre/post ratio. In contrast to cut detection, a small value of up to 3 for DMZ produces better precision in gradual transition detection.

4 Results

Recall and precision for cut detection are in the same area as last year. While we achieved slightly better performance in cut detection using the wavelet feature, we have obtained similar performance with much less computational effort.

Our results for gradual transition detection in TREC-11 were not competitive. Our new approach for this year shows promising results. We have good control over the choice of parameters, and an acceptable number of false positives, resulting in good precision. However, recall for gradual transitions is average and too low for practical use. Frame recall and frame precision of our system were among the best.

5 Conclusion and Future Work

In this paper, we have presented our enhanced moving query window method that we have applied to the TREC-12 shot boundary detection task. Separate decision stages for abrupt and gradual transitions are applied during a single pass through the video clip.

Recall and precision for all transitions are within reach of the best performing groups. The ranking approach works well for cut detection, but we see much room for improvement in gradual transition detection.

The fixed thresholds that we still apply should be replaced by dynamic thresholds. We found that fixed thresholds have a strong impact on recall.

When applying our algorithm to the TREC-11 shot boundary detection task, we experienced many false positives. We believe that this was partially due to the lower video quality which resulted in a very noisy slope of the average frame distance curve. It might be reasonable to employ pre-filtering stages, or a second feature, such as edge-tracking.

We will continue to focus on improvements for gradual transition detection. We plan to replace all fixed thresholds by adaptive methods to increase recall and make the system more applicable to different types of video.

We will also explore the use of three-dimensional histograms and spatial information, such as localised histograms, to reduce false positives. We plan to experiment with different feature spaces, and to explore the application of wavelet transform features for gradual transition detection.

Future refinements may consist of the application of filter stages and additional features, such as edge-tracking to reduce false detections in difficult video clips.

References

[1] B. Adams, A. Amir, C. Dorai, S. Ghosal, G. Iyengar, A. Jaimes, C. Lang, C. yung Lin, A. Natsev, M. Naphade, C. Neti, H. J. Nock, H. H. Permuter, R. Singh, J. R. Smith, S. Srinivasan, B. L. Tseng, Ashwin T. V., and D. Zhang. IBM Research TREC-2002 video retrieval system. In *NIST Special Publication 500-251: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, USA, 19–22 November 2002.

[2] P. Aigrain, H. J. Zhang, and D. Petkovic. Content-based representation and retrieval of visual media: A state-of-the-art review. *Multimedia Tools and Applications*, 3(3):179–202, 1996.

[3] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos. Distance measures for color image retrieval. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'98)*, volume 2, pages 770–774, Chicago, Illinois, USA, 4–7 October 1998.

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

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

[6] R. Brunelli, O. Mich, and C. M. Modena. A survey of the automatic indexing of video data. *Journal of Visual Communication and Image Representation 10*, 1047(3203):78–112, 1999.

[7] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, California, USA, San Francisco, California, 2001.

[8] B. Günsel, A. M. Ferman, and A. M. Tekalp. Temporal video segmentation using unsupervised clustering and sematic object tracking. *Journal of Electronic Imaging*, 7(3):592–604, 1998.

[9] A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *Proceedings of the ACM International Conference on Multimedia*, pages 357–364, San Francisco, CA, USA, October 1994.

[10] A. Hampapur, R. Jain, and T. E. Weymouth. Production model based digital video segmentation. *Journal of Multimedia Tools and Applications*, 1(1):9–46, 1995.

[11] X.-S. Hua, P. Yin, H. Wang, J. Chen, L. Lu, M. Li, and H.-J. Zhang. MSR-Asia at TREC-11 video track. Technical report, Media Computing Group, Microsoft Research Asia, 2002.

[12] F. M. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, June 1997.

[13] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Journal of Signal Processing: Image Communication*, 16(5):477–500, 2001.

[14] 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.

[15] R. W. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics (IJIG)*, 1(3):469–486, July 2001.

[16] C.-C. Lo and S.-J. Wang. Video segmentation using a histogram-based fuzzy C-Means clustering algorithm. *Journal of Computer Standards and Interfaces*, 23:429–438, 2001.

[17] S. Marchand-Maillet. Content-based video retrieval: An overview. Technical Report 00.06, CUI - University of Geneva, Geneva, Switzerland, 2000.

[18] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. *Journal of Visual Database Systems II*, 33(4):113–127, 1995.

[19] J. Nam and A. H. Tewfik. Dissolve transition detection using B-Splines interpolation. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 1349–1352, New York, NY, USA, 30 July – 2 August 2000.

[20] M. J. Pickering, D. Heesch, R. O'Callaghan, S. Rüger, and D. Bull. Video retrieval using global features in keyframes. Technical report, Department of Computing, Imperial College London, 180 Queen's Gate, London, UK, 2002.

[21] 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.
URL: `http://clips.imag.fr/mrim/georges.quenot/articles/cbmi99a.ps`.

[22] G. M. Quénot. TREC-10 shot boundary detection task: CLIPS system description and evaluation. In *NIST Special Publication 500-250: Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 142–151, Gaithersburg, MD, USA, 13–16 November 2001.

[23] G. M. Quénot, D. Moraru, L. Besacier, and P. Mulhem. CLIPS at TREC-11: Experiments in video retrieval. In *NIST Special Publication 500-251: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, USA, 19–22 November 2002.
URL: `http://trec.nist.gov/pubs/trec11/papers/clips.video.pdf`.

[24] 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, MD, USA, 13–16 November 2001.
URL: `http://trec.nist.gov/pubs/trec10/papers/trec10video_proc_report.pdf`.

[25] A. F. Smeaton and P. Over. The TREC-2002 video track report. In *NIST Special Publication 500-251: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, USA, 19–22 November 2002.
URL: `http://trec.nist.gov/pubs/trec11/papers/VIDEO.OVER.pdf`.

[26] D. Swanberg, C.-F. Shu, and R. Jain. Knowledge guided parsing in video databases. *Proceedings of the SPIE; Image Storage and Retrieval Systems*, 1908:13–21, February 1993.

[27] S. M. M. Tahaghoghi, J. A. Thom, and H. E. Williams. Shot boundary detection using the moving query window. In *NIST Special Publication 500-251: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, USA, 19–22 November 2002.
URL: `http://trec.nist.gov/pubs/trec11/papers/rmit.ps.gz`.

[28] Text REtrieval Conference – Video Track Evaluation Workshop, National Institute of Standards and Technology, Gaithersburg, MD, USA.
URL: `http://www-nlpir.nist.gov/projects/tv2003/tv2003.html`.

[29] L. Wu, X. Huang, J. Niu, Y. Xia, Z. Feng, and Y. Zhou. FDU at TREC2002: Filtering, Q&A, web and video tracks. Technical report, Fudan University, Shanghai, China, 2002.

[30] 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.

[31] D. Zhang, W. Qi, and H. J. Zhang. A new shot boundary detection algorithm. *Lecture Notes in Computer Science*, 2195:63–70, 2001.

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