AUDIO-VISUAL SEGMENTATION FOR CONTENT-BASED RETRIEVAL

David Pye, Nicholas J. Hollinghurst, Timothy J. Mills and Kenneth R. Wood

Olivetti & Oracle Research Laboratory,
Trumpington Street, Cambridge CB2 1QA, England

dp@orl.co.uk

ABSTRACT

This paper reports recent work at ORL on segmentation of digital audio/video recordings. Firstly, we describe an audio segmentation algorithm that partitions a soundtrack into manageably sized segments for speech recognition. Secondly, we present an algorithm for detecting camera shot-break locations in the video. The output of these two algorithms is combined to produce a semantically meaningful segmentation of audio/video content, appropriate for information retrieval. We report the success of the algorithms in the context of television news retrieval.

1. INTRODUCTION

The DART (Digital Asset Retrieval Technology) project at ORL is concerned with management of digital media such as text and hypertext documents, images, audio and video recordings. DART aims to provide the means to index, annotate, navigate and retrieve from diverse collections of these assets. In part, the project follows on from our successful collaboration with Cambridge University in the Video Mail Retrieval project [10]. Whereas VMR relied solely on speech recognition of the soundtrack, the DART project intends to advance these techniques and combine them with content-based image retrieval and video parsing techniques.

In developing the core DART technologies, we constructed a test-bed application allowing content-based retrieval of television news items. This application required the automatic partitioning of multimedia recordings into units appropriate for information retrieval. The result of a query should ideally be a compact segment encapsulating just the information requested, yet the segments must be self-contained and of sufficient length for effective indexing and retrieval. For example, we might wish to identify individual reports within a news broadcast.

The approach taken determines coincidences between acoustic boundaries and video shot-breaks. The acoustic boundaries are produced by an algorithm originally developed to reduce soundtracks of arbitrary size into manageable portions for speech recognition. The algorithm splits the audio stream at points where the acoustic characteristics change markedly. These points typically signify changes in speaker, microphone and acoustic channel conditions, or the starting or finishing of music. The shot-break locations are generated by our video parser. This was developed to determine the logical structure of the video stream by detecting cuts, fades, dissolves, and camera motion. An iterative algorithm selects acoustic boundaries to form suitably sized segments, favouring boundaries which are close to video breaks.

Given a preliminary audio/video segmentation, an optional post-processing stage uses information retrieval techniques to merge adjacent segments of similar lexical content.

2. AUDIO SEGMENTATION

Advances in speech recognition have seen the focus of research progress from small discrete utterances to “found” material such as broadcast news soundtracks. To process data of this form, an automatic segmentation stage is typically used to partition the soundtrack into acoustically homogenous segments. The approaches to audio segmentation considered loosely fall into two categories:

- Gaussian Mixture Models are suitably trained for acoustic classes (e.g. clean speech, male speech, pure music and noise) and used to classify frames. Contiguous frames similarly classified form a segment. These approaches function very well on suitable data but, in general, cannot discriminate between speakers.
- Local change in acoustic characteristics is estimated frame-by-frame through the soundtrack. Partitioning occurs where this measure peaks. This approach locates events such as speaker/channel changes or music onset.

The second approach was chosen for its ability to separate speakers and for its generality — it works independently of the language and domain. The basic scheme applied speaker change detection techniques similar to Gish & Schmidt [3] to segmentation [5][8]. The audio stream is parameterised at ten millisecond intervals into a sequence of 39-dimensional MFCC feature vectors. Every hundredth of a second, Gaussian means and variances are estimated for the preceding and following short windows (typically two seconds) of acoustics. The Kullback Leibler (KL) distance between the two windows is used as an estimate of acoustic change at that point. Segment boundaries are proposed at points where this distance is a local maximum.

We have extended this in two ways. Firstly, an attempt is made to increase the algorithm’s speed and to prevent segmenting mid-word by proposing boundaries only on pre-determined low energy frames. A threshold is estimated for each frame in terms of the local average and variance of the energy. Only frames whose energy fails to meet this threshold are considered for

splitting. The result of this technique is that typically only 10% of the processor intensive frame-by-frame distance measurements need to be performed.

One weakness of the basic technique is that segment boundaries are proposed using only local acoustic evidence adjacent to the boundary. The second change attempts to reinforce segment boundary hypotheses by considering the acoustic characteristics of the entire segments preceding and following them. This agglomerative algorithm starts by segmenting the soundtrack into small segments typically a few seconds long — rather smaller than those produced by the basic scheme. A Gaussian distribution is estimated for the acoustics of each segment. If the KL distance between any adjacent segments fails to exceed an empirically derived threshold, the boundary is deleted and the two segments merged. The algorithm converges when an iteration completes without merging any segments.

Table 1 shows the segmentation performance on a half hour TV news broadcast using hand segmented audio boundaries as a reference. The first line shows the basic scheme, which correctly identifies 70% of boundaries with a false alarm rate of 40%. These results are comparable with published figures by CMU [8] and BBN [5] on US broadcast news material, with all systems exhibiting a high false alarm rate. The results using the agglomerative algorithm form the second line, with an improved 81% correctly identified boundaries at a similar false alarm rate.

	Correct	False Alarms
Basic	**0.70**	**0.40**
Agglomerative	**0.81**	**0.41**

Table 1: Results of basic and agglomerative audio segmentation schemes on 30 minutes of a TV news recording.

3. VIDEO SEGMENTATION

We are concerned with dividing a video into semantically meaningful segments. One way in which this may be done is by detecting shot breaks in edited videos, although we are also investigating other cues such as camera motion and captions.

Many methods have been proposed for detecting edits in digital video; several have been evaluated by Boreczky [2].

- MPEG or JPEG-specific statistics can be used to measure frame differences [1,13]; this is fast but must be tailored to a particular compression scheme.
- Area statistics such as colour histograms [12] or regional grey-level histograms [9] are widely applicable and give good results, though they have trouble distinguishing between dissolves and motion.
- Feature-based methods track objects between frames of a moving sequence and can distinguish cuts, fades, dissolves and wipes by the pattern of feature appearance or disappearance [11]. This is around 50–100 times slower than histogram comparison.

We have implemented a regional colour histogram method, which represents a good tradeoff between speed and accuracy. We introduce a novel multi-timescale filter bank to detect and distinguish between cut, dissolve and fade effects whilst rejecting transients such as flashes.

3.1 Characterising the image

From each video frame we derive an array of 216 integers: the image is divided into nine blocks, and 8-element histograms are calculated for the Y, C_B and C_R colour components. These characterise the intensity and colour content of each region.

The distance between corresponding blocks in two images is the largest of the three L_1 histogram distances.

The distance between entire images is defined to be the median of the nine block distances. This disregards local motion or change, whilst giving a strong response to transitions which affect more than half of the picture.

3.2 Multi-timescale break detection

Let $D(i,j)$ be the median block distance between frames numbered i and j. The difference between successive frames at position t is thus:

$$d_1(t) = D(t, t-1)$$

This gives a strong response to cuts, but is less responsive to slow transitions such as dissolves, and can be sensitive to transients. We therefore define difference measures at longer timescales:

$$d_n(t) = \frac{1}{n}\sum_{i=0}^{n-1} D(t+i, t-n+i)$$

The moving average provides a degree of smoothing appropriate to the timescale, and elicits a peak response centred about each transition. In general, a given d_n detects transitions of duration less than or equal to n, and will reject transients of less than n frames. The peak response to a cut between static scenes will be independent of n. We consider four timescales: d_2, d_4, d_8 and d_{16}.

We define a *peak* to be a value of d_n which exceeds a certain threshold, and is greater than its 16 neighbours before and 16 neighbours after. Our algorithm looks for near-simultaneous peaks of d_{16} and d_8 — these transitions are localised in time and are likely to correspond to shot breaks. We also detect if they coincide with d_4 and d_2 peaks. In each case the d_n peak must be within a factor of $\sqrt{2}$ and a time within $\pm n/2$ of the corresponding d_{2n} peak, to be considered coincident.

Breaks which include a d_2 peak are interpreted as cuts; each longer timescale break is classified as a dissolve, fade-in or fade-out by considering the gradient of average brightness

across the transition. Motion will typically not produce coincident peaks.

3.3 Results

The multi-timescale break detector was tested on two half-hour TV news broadcasts containing many breaks of different types, graphic effects, camera motions and flashbulbs. Video was in MPEG-1 format. A threshold of 0.66 (33% histogram difference) was used throughout.

Results are summarised below. 95% of breaks were detected; only 7% of those reported were false. Most of the spurious breaks coincide with sudden camera motions. The video break detector runs in real time (including a software MPEG decoder) on a 167MHz Sun UltraSparc processor.

	Missing	$\tau = 2$	$\tau = 4$	$\tau = 8$
Spurious	–	**5**	**6**	**21**
Cut	**16**	**413**	**5**	**0**
Dissolve/Fade	**4**	**7**	**12**	**8**
Wipe/Other	**5**	**0**	**0**	**0**

Table 2: Numbers of breaks detected by the algorithm, tabulated by timescale τ, against hand classification of break type.

4. AUDIO-VISUAL SEGMENTATION

The goal of this stage is to produce semantically meaningful segments suitable for information retrieval. We index each *retrieval unit* by constructing a word frequency vector from the automatically transcribed speech. Segments should therefore be sufficiently large to successfully exploit word frequency statistics, yet small enough to enable precise searching.

4.1 The algorithm

In order to produce the retrieval unit segmentation, we combined evidence of audio and video break locations. Experiments showed that where an audio and video break coincide, evidence exists of a significant semantic change. A decision was taken to segment only on audio boundaries, using video evidence to decide an optimal subset of audio breaks to retain. This decision was taken for both theoretical and pragmatic reasons. Firstly, audio breaks are generally of higher semantic relevance than video breaks, which are often purely stylistic events. Furthermore, humans are more sensitive to disruption in the audio stream than in the video. The pragmatic reason was the desire to avoid the need to split (audio segment based) recognised speech transcripts.

A score is computed for each audio boundary in terms of the distance to the nearest video break (and the video break type), and the acoustic similarity of the bordering audio segments. Audio breaks associated with lower scores are the best candidates for splitting. The top-down algorithm iteratively splits the recording by considering audio breaks in order of increasing score. A split is performed if heuristics to encourage good segment lengths are satisfied.

An optional refinement we have investigated merges segments closely related by lexical content. This uses the recognised transcripts to build a word frequency vector for each retrieval unit. Vectors are compared using a cosine distance metric [7]. If the distance between the vectors of adjacent segments falls below a threshold, the two segments are merged. For instance, this will merge segments corresponding to an interview, despite acoustic differences and video breaks between the speakers, if the content includes common vocabulary.

4.2 Appraisal

While the concept of information retrieval units is rather abstract and thus difficult to evaluate, empirical evidence suggests that the algorithm is highly effective. Figure 1 shows the results of audio, video and retrieval unit segmentation respectively for the first 250 seconds of a typical TV news broadcast. In this context, a human might choose to segment a news story in its entirety, or instead select the individual components of it (such as the introduction by the newsreader, an outside report and an interview) as useful retrieval units. For short stories, our algorithm will typically identify the whole story, while longer stories are segmented into their constituent components.

The algorithm has also performed well on other domains including our laboratory project videos, and is applicable to any domain with regular acoustic events to detect. In the absence of acoustic events, emphasis can be placed on video breaks with silence detection used to reinforce them. However, this will not help for unedited material such as a video lecture with only a single speaker and no acoustic events or shot breaks.

5. SUMMARY

This paper describes segmentation algorithms that have been developed as part of a content-based retrieval system. An audio segmenter effectively partitions the soundtrack by determining acoustic events such as speaker and channel changes or music onset. The algorithm is fast and performs well without making any *a priori* assumptions about the language and domain of the recording. A video segmenter has been developed which accurately and efficiently locates shot breaks in edited video such as cuts, dissolves and fades. The output of these two algorithms is used in combination to produce a large grained, semantically coherent segmentation across a variety of domains. A TV news retrieval system developed at ORL successfully uses segments derived in this manner as the basic units of retrieval.

The DART project is simultaneously developing image and video parsing techniques that will allow high-level matching and retrieval of video segments based on picture content. These complement the speech-based indexing system described above. We intend to combine them to produce an integrated multimedia retrieval system, in domains such as broadcast news, educational hypertext, films, home videos and audio-annotated photographs.

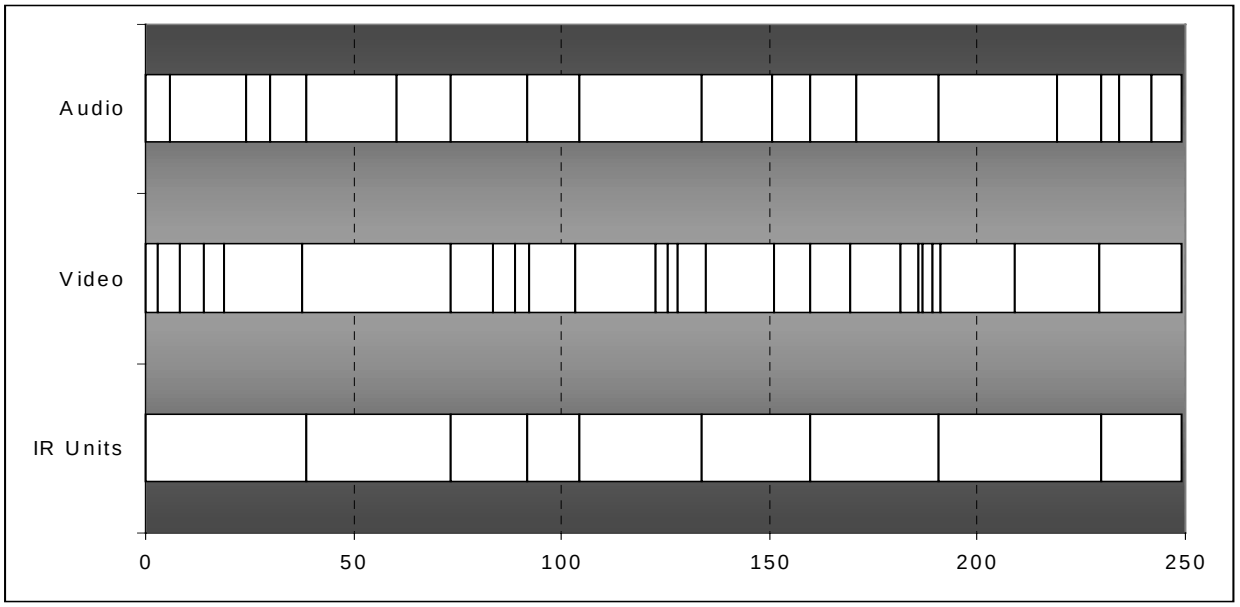

Figure 1: The partitioning of the first two hundred and fifty seconds of a TV news recording into audio, video and combined information retrieval units respectively.

REFERENCES

1. Arman, F., Hsu, A. and Chiu, M.Y., *Image Processing on Encoded Video Sequences*, Multimedia Systems, Volume 1, Number 5, page 211-219. 1994

2. Boreczky, J.S. and Rowe, L.A., *Comparison of Video Shot Boundary Detection Techniques*, Proc. SPIE Storage and Retrieval for Image and Video Databases IV, San Jose, CA, 1996.

3. Gish, H. and Schmidt, N. *Text-Independent Speaker Identification*, IEEE Signal Processing Magazine, 1994.

4. Hain, T., Johnson, S.E., Tuerk, A., Woodland P.C. and Young S.J. *Segment Generation and Clustering in the HTK Broadcast News Transcription System*. Proc. DARPA Broadcast News Transcription and Understanding Workshop, 1998.

5. Kubala, F., Anastasakos, T., Jin H., Makhoul, J., Nguyen, L., Schwartz, R. and Yuan, N. *Toward Automatic Recognition of Broadcast News.* Proc. DARPA Speech Recognition Workshop, New York, 1996.

6. Little T.D.C., Ahanger, G., Folz, R.J., Gibbon, J.F., Reeve F.W., Schelleng, D.H. and Venkesh, D., *A Digital On-Demand Video Service Supporting Content-Based Queries*, Proc. ACM Multimedia, Anaheim, CA, 1993.

7. Salton, G., Yang, C.S. and Wong, A. *A Vector Space Model for Automatic Indexing*. Communications of the ACM, 18(11):613–620, November 1975.

8. Siegler, M.A., Jain, U., Raj, B. and Stern, R.M. *Automatic Segmentation, Classification and Clustering of Broadcast News Audio*, Proc. DARPA Speech Recognition Workshop, Chantilly, VA, 1997

9. Swanberg, D., Shu, C.F. and Jain, R. *Knowledge Guided Parsing and Retrieval in Video Databases*, Proc. SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, 1993.

10. Young, S.J., Brown M.G., Foote, J.T., Jones G.J.F. and Spärck Jones, K. *Acoustic Indexing for Multimedia Retrieval and Browsing*. Proc. Intl. Conf. Acoustics, Speech and Signal Processing ICASSP 97, 1:199–202. Munich, April 1997.

11. Zabih, R., Miller J., and Mai, K., *A Feature-Based Algorithm for Detecting and Classifying Scene Breaks*, Proc. ACM Multimedia 95, San Francisco, CA, 1995.

12. Zhang, H.J., Kankanhalli, A. and Smoliar, S.W. *Automatic Partitioning of Full-Motion Video.* Multimedia Systems 1(1): 10–28, 1993.

13. Zhang, H.J., Low C.Y., Gong, Y. and Smoliar, S.W. *Video Parsing using Compressed Data*, Proc. SPIE Image and Video Processing II, San Jose, CA, 1994