An Intelligent Media Browser using Automatic Multimodal Analysis

Jonathan Foote, John Boreczky, Andreas Girgensohn, and Lynn Wilcox
FX Palo Alto Laboratory
3400 Hillview Avenue, Palo Alto, CA 94304, USA
+1 650 813 {7071, 7585, 7244, 7574}
{foote, johnb, andreasg, wilcox}@pal.xerox.com

1. ABSTRACT

Many techniques can extract information from an multimedia stream, such as speaker identity or shot boundaries. We present a browser that uses this information to navigate through stored media. Because automatically-derived information is not wholly reliable, it is transformed into a time-dependent "confidence score." When presented graphically, confidence scores enable users to make informed decisions about regions of interest in the media, so that non-interesting areas may be skipped. Additionally, index points may be determined automatically for easy navigation, selection, editing, and annotation and will support analysis types other than the speaker identification and shot detection used here.

1.1 Keywords

Content-based retrieval, video, speaker identification, automatic analysis, visualization, skimming

2. INTRODUCTION

Useful information may be automatically derived from multimedia streams. For example, cuts and camera motion can be detected from video, while cues such as applause, silence, and speaker identity can be found from the audio. This paper describes a browser/editor that uses automatically derived data to facilitate the navigation, annotation, and summarization of multimedia data. In a text editor, concepts like pages, paragraphs, and words make it easy to select and edit textual data and navigate a text document. For example, most modern word processors can be configured to select only entire words so that they are not truncated during editing. Similarly, a text file can be advanced by page, paragraph, line, or character. Few such tools exist for audio and video. However, relatively straightforward analyses can yield valuable information about the structure of a multimedia file. Though many analysis techniques have been described in the literature, there are few compelling applications that use the resulting data, especially as it can be unreliable on real-world sources. This paper presents ways to combine analysis results into a "confidence score" and novel ways to visualize and use such a score. Section 7 describes an intelligent browser application that uses audio and video analysis to facilitate browsing real-world corporate meeting videos.

3. PREVIOUS WORK

There has been related work on browsing continuous media query results and presenting confidence values from full-text search results. Bronson [6] describes using time-stamped keyframes and keywords to access portions of a video sequence. Yeung *et al.* [17] cluster keyframes to represent the structure of a video sequence. Arman *et al.* [1] use keyframes supplemented with frame pixels to represent content and motion within a video sequence. In these systems, the keyframes are static and represent fixed points in the video stream. Christal *et al.* [8] describe a system for rapid video playback, where the playback rate changes according to the number of detected index points but is not controllable by the user.

Wilcox *et al.* [16] developed a system that graphically displays speaker segmentation results. Hearst [11] displays term frequency values for text search results graphically. Confidence values for multiple search terms are presented together. Brown *et al.* [7] display a confidence values for text segments based on multiple terms. In this system, a set of confidence values can be selected to start playback of the associated video segment.

4. DETERMINING CONFIDENCE SCORES

Automatically derived information, or metadata, can be generally described as a time-dependent value or values that are synchronous with the source media, although other forms exist. For example, metadata might come from the output of a face-recognition or speaker-identification algorithm.

4.1 Constructing Confidence Information

Because automatic techniques do not always work reliably, and their output is not always easy to interpret, it is useful to translate metadata values into a "confidence score" before presentation to the user. For one-dimensional data, this can be as simple as a linear transform to the range between zero

and one, or as sophisticated as a nonlinear mapping based on *a priori* knowledge about the metadata's reliability. For example, it might be known that values below a certain threshold are insignificant noise, and the corresponding confidence score would be zero. As another example, in Figure 1, the confidence score was smoothed with a low-pass filter before it was rendered graphically. For non-numerical annotation or media types, such as text, closed captions, subtitles, or MIDI streams, confidence scores may be computed using statistical or other mathematical methods. For instance, the "tf-idf" weighting used in information retrieval was used to highlight relevant closed captions in [8]. For multidimensional data, such as the output of a face tracking algorithm, a lower dimensional confidence function may be computed, or image thumbnails, keyframes, icons, or projections may serve as a multidimensional confidence indication. Confidence scores may also be derived from the combination of two or more sensors. For example, combining speech detection and motion detection could yield a confidence measure about gestures. Any function may be used to combine confidence scores; the exact function may be automatically estimated using techniques such as learning Bayesian networks [12].

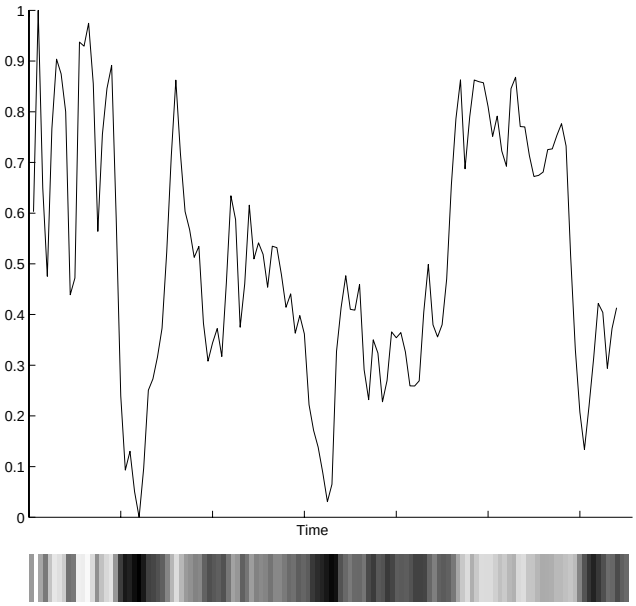

Figure 1. Graphical representation of confidence scores

4.2 Examples

A few examples should illustrate the utility of confidence scores. A simple yet useful application is to constrain an edit region to begin and end only at silences, so that speech is not cut in mid-word. By combining long-silence detection with video cut detection, only those cuts occurring during silence would have a high confidence score. Shots selected using this method are far less likely to interrupt a speaker mid-sentence. A more sophisticated example might combine speaker ID, face recognition, and long-silence detection, such that regions have a high confidence only if they are a complete phrase of a particular person both speaking and appearing on the video.

5. AUTOMATIC ANALYSIS TECHNIQUES

We are primarily using two kinds of automatic analysis: audio similarity for speaker identification and frame differences for shot detection. These provide several dimensions of data for our browsing application of Section 7. It is important to note that we can use confidence information from virtually any source and are not limited to the relatively simple analyses described here. In particular, we intend on integrating face, gesture and motion recognition into our browsing scheme as well as using other more sophisticated analyses, such as the combined video and audio analysis of [5].

5.1 Audio

Much research in diverse fields has been concerned with the extraction of useful information from audio. Besides the obvious speech recognition work, researchers have also used speaker identification [16], silence detection, emphasis detection [3] and even music and sound classification [14] for indexing and segmenting audio. The review paper [9] contains an in-depth discussion of these techniques, especially of speech recognition approaches. The speaker identification method used for the browser presented in Section 7 is based on the supervised-VQ approach of [10]. Though perhaps not as robust as hidden Markov models (HMMs), it has the advantage of needing no HMM decoding step, and gives a distance measure that allows direct interpretation as a confidence score. More general audio-to-text approaches would be desirable, but were judged impractical given the noisy multiple-microphone acoustic environment and informal setting described in Section 7.

5.2 Video

A straightforward video analysis is shot boundary detection. Boundaries are typically found by computing an image-based distance between adjacent (or regularly spaced) frames of the video, and noting when this distance exceeds a certain threshold. The distance between frames can be based on statistical properties of pixels [13], histogram differences [19], compression algorithms [2], edge differences [18], or motion detection [15].

The video features used for this browser are based on histograms and motion detection. Gray-scale histogram differences are used as a shot boundary detection features because they are quick to compute, and perform as well as many more complicated techniques [4]. We also use a motion feature based on block matching to give a confidence value for camera and object motion [5].

6. DISPLAYING AND USING CONFIDENCE INFORMATION

In the simplest form, a confidence score may be plotted as a two-dimensional graph where one axis is time and the other the confidence value. Regions of high confidence, corresponding to potentially interesting intervals, can then be visually identified. Different metadata can be differentiated

by rendering in visually distinct ways such as different colors. Other graphical representations may be used, such as coloring objects with hue or intensity proportional to the confidence value along the time axis, as in Figure 1. As with other systems using a timeline-based display [16], the time axis may be "zoomed" in or out to change the time resolution. Figure 2 shows another way of displaying confidence scores associated with *changes* in the media stream. For this example, the integral (cumulative sum), shown with dots, of the histogram difference data is mapped to the color bar on the right. If the time bar is colored proportionally to the cumulative sum, then regions of different video are displayed in different colors, as shown in the figure. The color difference between regions is proportional to the difference in the video signal, and regions of relatively uniform video are displayed in a uniform color. The actual choice of color is not particularly important, as long as it varies perceptibly; and a black-and-white representation (as in the figure) may be useful as well. (Interested readers may wish to view the color version of the figure in the Electronic Proceedings.)

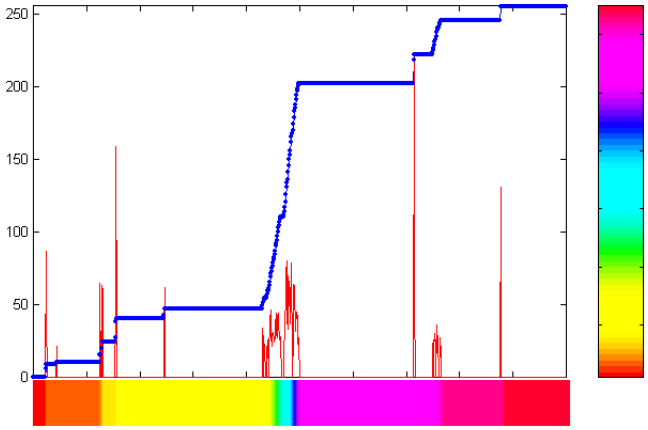

Figure 2. Mapping cumulative video changes to color

Using parallel displays along the same time axis allows a comparison of different metadata streams, or of the same metadata extracted from multiple media sources. The last mode is especially valuable when comparing different media files, for example the ranked output of an automatic search across a large number of files. To reduce screen clutter, user controls can select among confidence scores to display; in addition, multiple scores may be displayed on the same time axis using different graphical representations such as color. This is particularly practical in situations such as speaker identification where typically only one of several indications will be highly confident at any particular time.

6.1 Mapping Confidence to Index Points

Additional information can be extracted from a metadata stream by thresholding; that is, finding the times when the confidence score crosses a certain threshold value. These times can serve as index points for random access; interface buttons change the current playback time to the next (or previous) index point. In another mode of operation, the selection region could be extended or diminished using similar controls. Multiple thresholds can be used: a high threshold yields fewer index points and thus coarser time granularity, while a lower threshold allows finer placement (see Figure 3). A valuable feature is a user-variable threshold controlled by a slider or similar continuous interface. In some applications, a time-variable threshold (such as a moving average of the confidence score) may be appropriate. In this case, thresholding would occur not on the score directly, but some function, such as the difference, of the threshold and the instantaneous confidence score. This scheme allows general functions of one or more inputs, including functions or weights that might change over time given a learning or feedback mechanism.

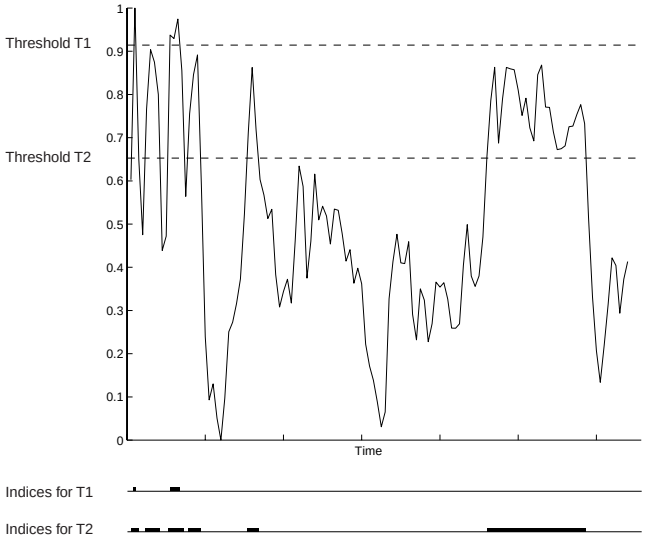

Figure 3. Effect of varying threshold on available index points

Among other non-linear functions, thresholding is especially valuable for displaying index points determined from a continuously variable confidence score. Visual indications or icons indicate regions where the confidence score exceeds a certain threshold. By varying the threshold, the user can vary the number of available index points. For certain confidence types, such as the image change data of Figure 2, good thresholds may be determined (or learned) in advance and used to generate index points without requiring user intervention. In this case, no graphical display is even necessary: interface buttons can advance or retreat to the next index point. Like indexes, keyframes may be produced depending on a confidence score; again any combination of modes may be used. For example, a keyframe-detector based on video frame differences may be conditioned on a thresholded confidence score from the speaker identification, thus leading to keyframes including only the desired speaker. Thresholds can also be used in concert with confidence scores to produce a summary or excerpt of the video: only those regions above the threshold may be included in the summary, as in the "video skims" of [8]. Thus thresholds, though perhaps the simplest of non-linear operations, can be very valuable especially when thresholds are not precomputed but rather left to the user to adjust as desired.

7. APPLICATION: AN INTELLIGENT BROWSER

We have prototyped a browser that uses many of the above ideas, towards the application of browsing and reviewing corporate meetings, presentations, and informal video. At FX Palo Alto Laboratory, weekly staff meetings as well as other seminars and presentations are held in a conference room outfitted with 3 controllable cameras, an omnidirectional podium microphone as well as six ceiling microphones with automatic gain control, and a large back-projection display. All formal meetings and most presentations are videotaped, MPEG-encoded, and made available to staff via the FXPAL intranet. The intelligent browser is a first step towards intelligent retrieval and indexing of this material for corporate memory purposes. We currently have upwards of 50 hours of video and are developing automatic annotation and search tools. The browser will also be used to review automatic search results as well as in a stand-alone mode.

For audio and speech recognition purposes, the environment is less than ideal. The conferencing microphones are sensitive enough to capture remarks from the back of the room; however, they also capture unwanted sounds such as paper rustles, door closings, coughs, murmurs, and the ventilation system. In addition, automatic gain control changes microphone levels unpredictably. For speaker identification, two meetings were chosen as training material, and training data for 3 male speakers was hand-segmented from the audio, as well as examples of silence, laughter and applause. Each speaker had from 80 seconds to several minutes of training data available; using this data, signatures were computed using the MMI quantization tree methods of [10]. These were used to generate indexes for the browser. Note that these are only preliminary attempts at indexing and we continue to investigate other sources of metadata, such as the combination of audio and video features described in [5].

7.1 A Prototype Browser

Figure 4 shows the user interface of our browser prototype. To the top left are the usual video playback window and controls. On the middle right are menu controls that select which confidence scores to display on the bottom timebar. Confidence scores are displayed time-synchronously with the video slider bar. In the figure, a confidence score based on color histogram difference is shown. This results in bright lines at times corresponding to video cuts. Also available are confidence scores based on frame difference (to detect camera motion), speaker identification for three common speakers, slide/graphic detection, and music detection, though not all scores are available for all videos.

The threshold slider at middle right controls how index points are derived from the confidence scores. Index points are shown as bright lines in the upper region of the time bar. (This is primarily for the B/W reproduction of this paper: we find index points may be better displayed using contrasting colors.) In this example, the slider is set relatively high, and thus only five index points are available. The “index

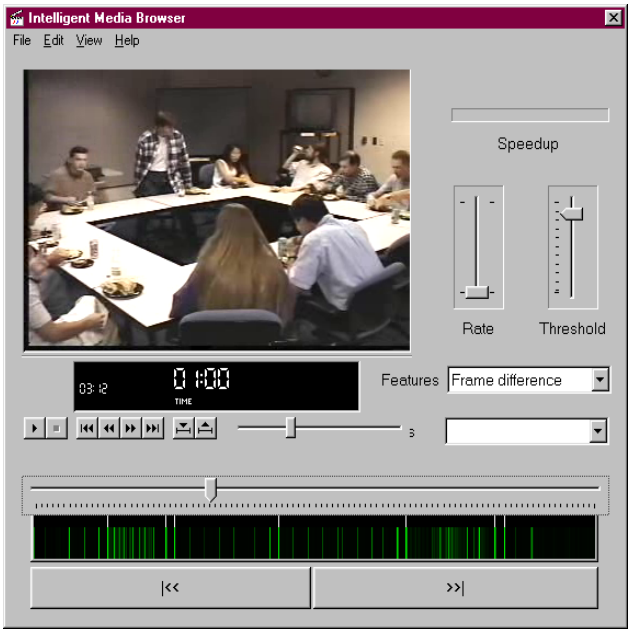

Figure 4. Intelligent Browser Prototype

buttons” beneath the time bar automatically advance the playback point to the next or previous index point. This is shown at the bottom of Figure 4, where the buttons labeled “|<<” and “>>|” move the playback point to the next index point, as determined from the threshold. Figure 5 shows that lowering the threshold results in many more available index points. In an area of large confidence variation (many index points), the user can select the most significant indication by increasing the threshold. In a region of small confidence scores the user can still find index points within the region by reducing the threshold, though they may be less reliable.

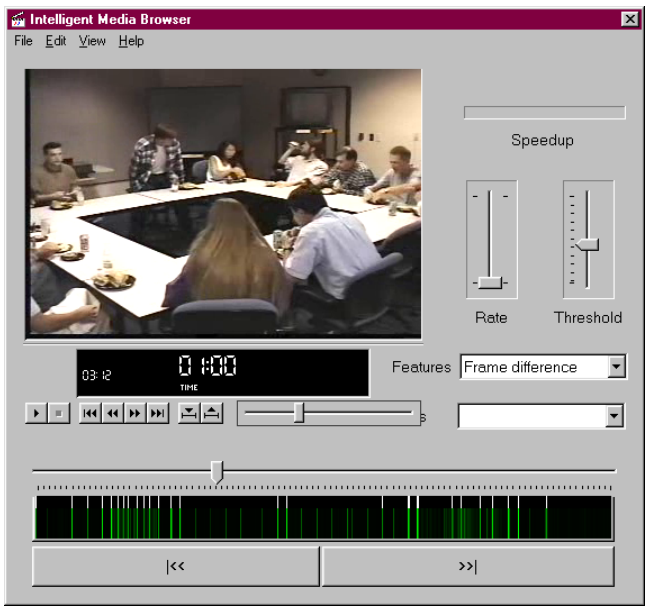

Figure 5. Index points at a lower threshold

7.2 Variable Rate Control

We also use the confidence scores to control the video playback rate. A user-controlled rate multiplier allows areas of low confidence to be played back at high speed while areas of high confidence are played closer to normal speed (see the "Rate" slider and the "Speedup" display on the right of Figure 5). If M is the maximum rate set by the user ($M > 1$), and C is the [0, 1] confidence score at any instant, then the instantaneous playback rate R is determined by

$$R = C + M(1 - C)$$

This relation varies linearly between the normal playback rate ($R = 1$) in regions of high confidence ($C = 1$), and the maximum speedup ($R = M$) in regions of low confidence ($C = 0$). Note that the speedup feature is easily disabled by setting $M = 1$. This is similar to the search function of a VCR where the fast-forward rate is automatically controlled by the confidence score. Because of the limitations of the Microsoft video player (which pauses briefly after every control change), we change the speed incrementally rather than continuously. We also smooth the instantaneous rate R with a running average to both avoid over-frequent speed changes and to flatten gaps or spikes in the confidence score.

The variable speed mechanisms allow users to rapidly skim areas of little interest while preserving a feel for the context. Areas of greater interest are presented at normal speed to convey the relevant information. We found a particularly good application of this technique was skipping musical interludes in a corporate video. Because no information is actually discarded, this may preserve context better than the "video skims" of Christel *et al.* [8].

8. CONCLUSIONS

As might be gathered from Section 3, much work has been done on automatic analysis of multimedia content. Useful applications using such analysis are not as easy to find. We have presented some ways to make error-prone and noisy automatic estimates more useful to the user. Exactly how useful will require user studies and certainly more experimentation. This paper presents only a preliminary attempt at a solution, but one that should serve as a good testbed for exploring additional analysis methods and interface representations. This approach is suitable for all time-dependent media, such as closed captions, digital ink, text annotations, and MIDI streams, in addition to audio and video. There are several promising avenues to be explored, such as combining confidence from several modes. For example, a keyframe-detector based on video frame differences may be gated with a thresholded confidence score from a speaker identifier, thus leading to keyframes including only the desired speaker. Thresholds can also be used in concert with confidence scores to produce a summary or excerpt of video or audio data; only those regions above the threshold may be included in the summary.

9. REFERENCES

[1] Arman, F., Depommier, R., Hsu, A., Chiu, M.-Y., "Content-based Browsing of Video Sequences,", In *Proc. ACM Multimedia 94*, San Francisco, October 1994, pp. 97-103.

[2] Arman, F., Hsu, A., Chiu, M-Y., "Image Processing on Encoded Video Sequences", *Multimedia Systems* (1994) Vol. 1, No. 5, pp. 211-219.

[3] Arons, B., SpeechSkimmer: "A System for Interactively Skimming Recorded Speech". *ACM Trans. on Computer Human Interaction*, Vol. 4, No. 1, pp. 3-38, March 1997.

[4] Boreczky, J. and Rowe, L., "Comparison of Video Shot Boundary Detection Techniques", *Proc.SPIE Conference on Storage and Retrieval for Still Image and Video Databases IV*, San Jose, CA, February, 1996, pp. 170-179.

[5] Boreczky, J., and Wilcox, L.,"A hidden Markov model framework for video segmentation using audio and image features." *Proc. ICASSP '98*, Vol. 6, 1998, pp. 3741-3744, May 1998.

[6] Bronson, B., Hewlett-Packard Company, Palo Alto, CA "Method and apparatus for indexing and retrieving audio-video data." US Patent 5136655: Aug. 4, 1992

[7] Brown, M., Foote, J., Jones, G., Spärck Jones, K., and Young, S., "Automatic Content-Based Retrieval of Broadcast News." In *Proc. ACM Multimedia 95*, San Francisco, November 1995.

[8] Christal, M., Smith, M., Taylor, C., Winkler, D., "Evolving Video Skims into Useful Multimedia Abstractions," in *Human Factors in Computing Systems, CHI 94 Conference Proceedings* (Los Angeles, CA), New York: ACM, pp. 171-178, 1998

[9] Foote, J., "An Overview of Audio Information Retrieval" *ACM-Springer Multimedia Systems*, in press.

[10] Foote, J., "Rapid Speaker Identification using Discrete MMI Feature Quantisation," *Expert Systems with Applications*, (1998) Vol. 13, No. 4.

[11] Hearst, M., "TileBars: Visualization of Term Distribution Information in Full Text Information Access." In *Proc. ACM SIGCHI*, May, 1995.

[12] Heckerman, D., "A Tutorial on Learning with Bayesian Networks,", Microsoft Technical Report MSR-TR-95-06, March 1995.

[13] Kasturi, R., Jain, R., "Dynamic Vision", in *Computer Vision: Principles*, Kasturi R., Jain R., Editors, IEEE Computer Society Press, Washington, 1991.

[14] Pfeiffer, S., Fischer, S., and Effelsberg, W., Automatic Audio Content Analysis. *Proc. ACM Multimedia 96*, Boston, MA, November, 1996, pp. 21-30

[15] Shahraray, B., "Scene Change Detection and Content-Based Sampling of Video Sequences", in *Digital Video Compression: Algorithms and Technologies*, Rod-

riguez, Safranek, Delp, Eds., Proc. SPIE 2419, Feb 1995, pp. 2-13.

[16] Wilcox, L., Chen, F., and Balasubramanian, V., "Segmentation of Speech using Speaker Identification." In *Proc. ICASSP 94*, Vol. S1, pp. 161-164, April 1994.

[17] M.M. Yeung, B.L. Yeo, W. Wolf and B. Liu, "Video Browsing using Clustering and Scene Transitions on Compressed Sequences", in SPIE Vol. 2417 Multimedia Computing and Networking 1995, pp. 399-413, Feb. 1995.

[18] Zabih, R., Miller, J., Mai, K., "A Feature-based Algorithm for Detecting and Classifying Scene Breaks", *Proc. ACM Multimedia 95*, San Francisco, CA, November, 1995, pp. 189-200.

[19] Zhang, H.J., Kankanhalli, A., Smoliar, S.W., "Automatic Partitioning of Full-motion Video", *Multimedia Systems* (1993) Vol. 1, No. 1, pp. 10-28.