

Hierarchical Brushing in a Collection of Video Data

Dulce Ponceleon and Andreas Dieberger

IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120

dulce@almaden.ibm.com, andreasd@us.ibm.com

Abstract

The amount of digital video and associated metadata being generated and stored is increasing rapidly. Given the complex spatial and temporal structure of video information it is a formidable challenge to provide compact and human-readable representations of such content. The solution has to be versatile in order to satisfy different user needs, such as browsing, zooming (looking for something specific), discovering or recording access patterns, performing partial searches that can be re-used later, etc. We present a new approach that addresses many of the difficulties. Our representation provides, an at-a-glance high-level overview of the video collection and also serves as a navigational tool in that collection. While the user navigates to finer-level representations in the video collection our visualization allows him to maintain a sense of location and context within the collection. We introduce the movieDNA as an abstraction to visualize interesting features in a video. Our compact, yet flexible, representation is applicable to any type of linear data. In order to access even larger amounts of data in one view and to express several levels of granularity and compactness we extend the movieDNA to a hierarchical movieDNA. Hierarchical brushing enables browsing, navigating and visualizing several semantic levels of video content. We discuss hierarchical brushing of video content for various user scenarios, and present examples demonstrating the versatility of our approach.

Keywords

Information visualization, video searching and browsing, video and multimedia visualization, navigation tools, hierarchical video summarization, hierarchical brushing and linking

Introduction

Given recent progress in computer vision, speech recognition, video indexing and retrieval technology, multimedia authoring, video streaming, and Internet, the rate of generation of digital video and associated metadata is increasing at a fast pace. This growth

combined with the plunging prices of storage has resulted in ever increasing amounts of digital video data being archived for increasing periods of time. Providing rapid access and effective representations of such video content is essential, otherwise storing thousands of hours of video is of no use. Today even inexpensive personal computers feature enough hard disk space and processing power to serve as entry-level video processing stations. Falling prices in video capturing devices, digital video cameras, and the wide availability of broadband internet connectivity will all lead to make video a media type as commonly used as static images and sounds (MP3s) are today.

However, navigating large collections of linear data is a big challenge, because it is impossible to directly apprehend a linear media type (video, music) at one glance. Indirectly, such a one-glance assessment of linear data can be achieved through an abstraction of the data. Compare this to typical navigation tools for – say – image data, where it is easily possible to shrink the data itself to a thumbnail image. Many thumbnails can then be displayed side by side and the human visual system is easily capable of gaining an overview of even a large number of such thumbnails. Unfortunately, because of the issues mentioned above, a generalization of this approach to video data or music is not directly possible. We designed and built a visualization and navigation tool for large video collections, called movieDNA, which provides an overview of video data through an abstraction and permits easy navigation in the data as well.

Currently there are not many tools or approaches that support navigation through large amounts of video data that are easy to use and keep the user *within* context. We achieved this by focusing on the three fundamental questions of navigation in any type of information space:

1. Where am I?
2. Where can I go?
3. Where is X?

The movieDNA and hierarchical brushing can serve as navigation tool either for a single long video, or for a

larger collection of videos. The movieDNA approach is general enough so that it can be applied to any kind of linear data, like video, music, event logs etc. In this paper we focus on its application to the domain of video data.

Media Navigation

Although current computers have impressive graphics capabilities, our technological culture is still mostly text-centered. Because of this focus on textual technologies we are trained well to cope with text on displays, to interact with it or to highlight it. We also have a rich tradition in typography to convey subtle meanings (for example serif vs. sans-serif fonts, fonts that convey futuristic or old-fashioned connotations, etc.) Our visual culture of text is well defined and relatively well understood – at least within one culture.

Linear media like video and audio are much newer technologies and we do not have a comparable culture of use of these yet. Even a simple concept like “highlighting” is straightforward in text but not in video. Analogously, a textual summary is a more familiar concept than a video summary. While we are used to navigate in textual information (tables of contents, index pages in a book, footnotes etc.) a similar tradition of navigation in linear data is not in sight. Both gaining a rapid overview of linear data through tools such as visualization or summarization as well as easy navigation of the linear media are still research issues.

In [11] Hearst points out that “aside from icons and highlighting the main information visualization techniques include brushing and linking, panning and zooming, focus-plus-context, magic lenses and the use of animation to retain context and help make occluded information visible”

Media Segmentation and Media Features

The literature shows that typically the first step towards video content analysis is segmentation [2,13,22]. That is, video is divided into manageable semantic units. Automatic shot-boundary detection is commonly used for video segmentation. Through segmentation we map continuous media into discrete units enabling indexing. In turn, indexing and cataloging makes video searchable. Requiring manual annotation of media is extremely costly for most practical applications. Hence, although imprecise, automatic segmentation is essential to any practical solution to manage media. Another important issue to address is identifying *what* is interesting in a video and more specifically, of

interest to the *user*? Automatic feature extraction, although imprecise as well, is key to address this issue. Video segments and video features are the basic elements to build a movieDNA. The segmentation and feature-extraction methods used to generate the sample movieDNAs presented here are part of the CueVideo system [22]. CueVideo is a system for indexing, browsing and searching of video and its associated metadata. Some features of the CueVideo system are shot-boundary detection, speech recognition, phonetic retrieval matching [18], topical segmentation based on the speech transcript, slide matching, event detection, and audio time scale modulation [1]. In this section we review the technologies used in this paper: Video collection categorization and automatic slide matching.

In video collection categorization we first perform term extraction. The term-extraction method is based on the techniques used for query expansion by selecting co-occurring terms. In the query expansion fixed size unit segments from a given speech transcript are clustered and segment cluster labels are generated to extract potentially important multi-word concepts and names that convey the underlying topic structure in the transcript [19]. Where single word labels, out of context are not very meaningful (e.g. “large”, “system”, “index”, “finger” etc.), multi-word noun phrases often convey the underlying information (e.g. “system architecture”, “index finger” etc.)

Slide matching [20] is achieved by modeling the spatial layout of the smaller regions composing the slide. The electronic slide images of a presentation are pre-processed to extract the spatial layout information. Region Hashing is a technique used to discover the alignment of the affine intervals of the video keyframe containing a slide and the slide in –say—a PowerPoint presentation. Only the top 5 matches are displayed in the movieDNA.

The movieDNA

The movieDNA is based on our work on Context Lenses [4]. Context Lenses were inspired by earlier work by Hearst [10]. The term movieDNA was chosen, because our visualization shows some visual similarities to DNA prints.

The movieDNA assumes a segmentation of the video data, ideally based on the video content. If a content-based segmentation is unavailable or not needed, segmentation can be time based. For example, a longer video might be split into segments of 2 minutes length.

Each of these segments is analyzed to determine the presence of features of interest. These features can be visibility of a person, occurrence of background noise, music, or voice-overs, camera angle, or any other kind of metadata imaginable. We successfully used features based on searching for key terms and key n-grams (multi-words) in a textual transcript of video voice-overs. Such a transcript can either be generated by hand, or as in our system, be generated automatically through speech recognition.

A movieDNA is a grid, containing as many rows as there are segments. The vertical arrangement of the DNA display reflects the sequence of segments in the video. Each column in the visualization indicates presence or absence of one the features we are interested in. In its simplest form, the DNA display indicates presence or absence of features by painting cells in the grid in a color distinguishable from the background. Different colors or a gray scale can indicate relevance, scoring, multiple occurrences of features in a segment and such. See Figure 1 for a sample movieDNA for a Boeing training video. It is approximately 15 minutes long and it has been divided into 1-minute segments. The columns of the movieDNA represent the top five multi-words automatically extracted from the speech transcript. Every feature is mapped to a different color.



Figure 1: movieDNA for a Boeing Training Video

Users brush through the video by moving the mouse cursor over the DNA. Every line in the DNA triggers display of a fold-out window to the right of the corresponding line. This window displays detailed metadata on the segment in question. Typically, the window shows a poster frame for that segment as well as (parts of) the transcript. Depending on the current task and user preference additional metadata can be displayed, such as the duration of the segment, annotations, etc. Clicking on the highlighted line in the DNA or on the thumbnail triggers a new window where the video starts playing precisely at the beginning of the corresponding segment (7 minutes into the video for Fig. 1)

Using the movieDNA users can easily gain an overview of the distribution of features in a video, while the fold-out window provides detailed information about single segments. An important feature of the movieDNA is that the fold out window does not obscure the DNA itself. This allows user to study the in-focus information while maintaining a sense of context and location in the overall video.

Hierarchical Brushing

This simple (single level) movieDNA is perfectly suited for moderate length linear data, for example to navigate a video of 1-to-2 hours duration. To navigate longer videos or collections of videos, this approach is not sufficient, though. Therefore we generalized the single-level movieDNA to a hierarchical movieDNA.

To achieve this, two key issues had to be tackled. Firstly, a hierarchical movieDNA requires a segmentation of the entire video data into sections of manageable size. Secondly, we need a more compact representation or abstraction of the data contained in a movieDNA to save screen real estate.

In case of a video collection where each video contains – say – 1 hour of video, the segmentation is taken care of by the video length already (this is a special case of time segmentation). In the case of one very long video it is necessary to split the video into segments of manageable size, ideally into sections that are topical units or into sections of similar length.

For each of these sections an abstraction of the DNA display has to be generated, that is smaller than the DNA itself. One possibility to achieve this is to simply shrink the DNA display. However, as the DNA can contain quite a large amount of data and the resolution of current display devices is relatively low, shrinking would obscure information.

Instead we generate an “aggregated DNA”, which does not contain active lines, like the single level movieDNA does. Instead each aggregated DNA acts as one active segment in a movieDNA. Several such aggregated DNAs are stacked on top of each other, to form a higher level movieDNA: as users roll the mouse cursor over one of these aggregated DNAs, a fold-out window opens, which then contains a full movieDNA for the corresponding video section.

This second level movieDNA is a fully functional movieDNA as described above such that users can apply brushing to scan the video. Using this hierarchical movieDNA users can first brush inside the collection of videos and then brush inside each video in more detail.

Critical for the utility of a hierarchical visualization of this kind is that the top level of the visualization is compact and space saving. At the same time, though, this compact representation needs to convey enough information about the video segment under consideration to allow deciding where to explore further. To achieve this, the aggregated DNAs at the top level needs to convey what Information Foraging theory defines as "information scent" [16]. Information scent is a noticeable trace of information that is otherwise not directly visible. Our current hierarchical movieDNA contains only 2 levels, but movieDNAs with more than 2 levels are possible and in such systems information scent will be even more critical to avoid aimless poking around in the information space. For example, one could use automatic hierarchical segmentation of video [13,17], where every aggregated level correspond to a level in the hierarchy.

The aggregated DNAs of the hierarchical movieDNA aim to convey as much information about the movieDNA of the second level as possible. Currently our aggregation function is relatively straightforward: we aggregate several (5) DNA lines into one line by counting occurrences of features and visualizing the count as gray levels. This way, if a feature occurs in several of the 5 aggregated lines of the DNA this feature shows up strongly in the top level DNA (meaning: it will show strong information scent). Logical extensions of this model would consider weights of features, relevance rankings etc. In addition to the aggregation step, the aggregated DNAs are drawn at a smaller scale. As users do not need to interact with single lines in the aggregated DNAs the rows in this abstraction can be much smaller. See Figures 2 and 3.

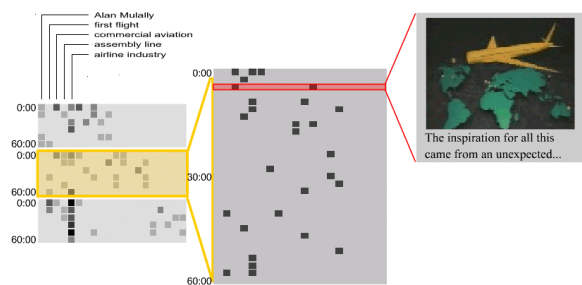


Figure 2: hierarchical movieDNA.

We mentioned that a movieDNA displays an arbitrary number of features for the entire video. In our current implementation the aggregated DNAs show the same number of features as the second level DNAs. However, this is not an iron cast rule. Just as we aggregated several segments into one, it might be feasible to aggregate several related features into one,

in order to achieve an even more compact representation on the first level. Such a design has to ascertain that the first level contains enough information scent to allow users to navigate.

For the DNA display as we described it, a segmentation of the video is essential. However, it is possible to combine the segmented representation of the video with non-segmented metadata in one representation. For example it is possible to show a visual representation of the soundtrack of the video (an amplitude graph), indicating the volume of the soundtrack alongside of the movieDNA. Another useful feature to visualize this way is to show the number of accesses to this particular video segment, indicated – for example – through a color spectrum drawn alongside the DNA. In a video that can be accessed by a large number of people such a feature would implement a social navigation tool, indicating popularity of sections of the video.

System Implementation

One component of our system, the movieDNA and the hierarchical movieDNA are written in Java. The applications read data from text files. In a hierarchical movieDNA there is one such text file for every video. These text files contain both the data for the DNAs themselves, as well as associated textual metadata, if used. Transcript or top keywords of a video segment are examples of such metadata. Furthermore, the data files can contain a URL for every segment, which provides access to a streaming version of that particular video segment. Most of the data files are generated automatically by our segmentation and feature extraction component. This component uses CueVideo technology and a set of utilities written in C and C++. In the current implementation batch files tie the two parts of the system together.

Application Scenarios

The movieDNA is a useful tool to navigate one video or a static collection of video data. However, the concept is very flexible and can accommodate a wide variety of diverse application scenarios. In this section we present a number of possible applications of this technology.

Figure 3 shows the hierarchical movieDNA for a collection of three videos. Each video is approximately one hour long. We have included this as an example to show the kinds of features that might be of interest to users. In this example the columns show the presence several audio events (such as music, and background noise) detected in the corresponding close caption.

From this movieDNA one can easily find segments in the video where several features co-occur and where the audio quality was not good enough to generate an accurate caption.

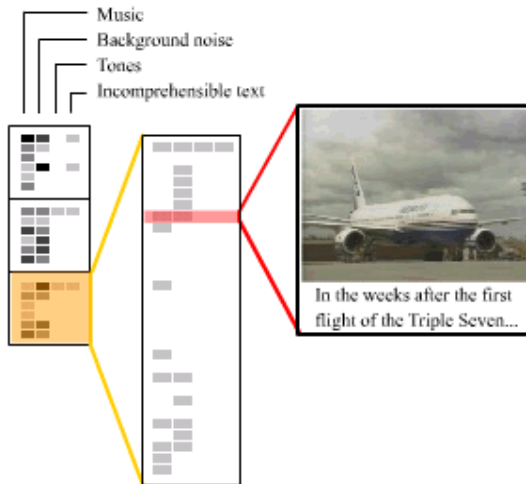


Figure 3: hierarchical movieDNA highlighting features in the audio track.

Although the term “linear data” seems to indicate that data represented by a movieDNA needs to be continuous, this is not necessarily so. For example, a slide presentation is (mostly) linear, but it is naturally segmented into slides. A slide presentation, along with a recording of a live lecture is an ideal application of the ideas realized in the movieDNA. Figure 4 shows an example of such an application. Note that the fold out window now shows two images: the slide itself, as well as the poster frame of the corresponding section of the lecture video. The segmentation appears to be trivial in this case. However lecture videos sometimes show the slide, or a lecturer refers to an earlier slide which makes an optimal segmentation more difficult than it might appear at first. Clearly this particular application of the movieDNA by itself provides ample opportunity for additional research.

A movieDNA like display can also represent streaming data or an event log for ongoing processes, be that log data from a Web server, from a security system or from a patient monitor in an emergency room. In this case the “current” segment of the DNA is the bottom line (representing “now”) and the DNA grows upward (higher portions of the DNA represent earlier points in time). Segmentation would occur either time based, or based on certain “events”. Users of this display can easily monitor the recent history of a monitored process and quickly drag the pointer to a location of interest to see what happened at that point.

For example, an emergency room surveillance system could use a movieDNA to show vital data about a patient: heart rate, blood pressure, breathing rate etc. The fold out window shows stills from a surveillance camera mounted in the emergency room. Segmentation is based on time and, additionally, on key events, like a major change in the heart rate. If the DNA shows a sudden surge in heart rate, it is a simple matter to check that -- for example -- the patient turned around at that point in time.



Figure 4: movieDNA of a talk using slide matching.

A similar application is to use this tool to navigate a surveillance or security system. In that application the DNA could represent movement sensors in various hallways and instead of one thumbnail, the fold out would show a larger number of thumbnails, each representing a key frame from one of several surveillance cameras. When reviewing the security data it is easily visible whether a door was opened recently. Moving the mouse cursor over that bar in the DNA shows the thumbnails from all surveillance cameras at that moment in time. Another related application would be to use a movieDNA to navigate a recorded theater performance with several camera angles. In this example the features in the DNA could indicate when each of the main characters is on stage, etc.

Applications for the hierarchical movieDNA for video collections are very varied as well. For example, it supports exploratory browsing in a collection, where there is little a priori knowledge of the content. It supports conducting searches in the video based on combinations of features, or on the transcript and to zoom in on matching segments while maintaining a sense of location and context in the overall collection. For this reason we imagine that hierarchical movieDNAs will provide especially useful for general video cataloging and indexing.

Many applications of a (hierarchical) movieDNA depend on the type of linear data visualized. For example in videos of lectures the movieDNA is an ideal tool to locate speaker changes or audience questions. We expect many more such applications to emerge as we apply the movieDNA concept to different domains.

Related Work

There is a wide range of approaches to video navigation, browsing and summarization. They exhibit various types of interfaces ranging from plain text, over images, storyboards, sophisticated storyboards, animations to derived video or 3D representations.

Traditional video storyboards [1,9] consist of one-dimensional or two-dimensional arrays of images sorted in time. Even for short videos storyboards require scrolling and flipping through pages. For long videos or video collections they are definitely not suitable as navigation tools. There are also more sophisticated visualizations of video content beyond storyboards [5,12,14,21,23] However, unlike our tools, they neither work for both videos and collections of videos nor do they feature compactness and versatility.. There are more sophisticated approaches to media visualization some of which we describe below.

Informedia [25] has the one of the most advanced interfaces of the “integrated” solutions of video cataloging; indexing and retrieval, but still no scalable unified representation. The Informedia project combines speech recognition, image processing, and natural language understanding techniques to process video automatically in a digital library system. A basic element of their interface design is the provision of alternate browsing options in response to a query. Headlines, thumbnails and filmstrips are viewed statically whereas skims are played back to communicate the content of the video. The filmstrip view reduces the need to view each “video paragraph” in its entirety by providing a storyboard for quick viewing.

A Videogram [3] provides a fine temporal resolution of the dynamics of the video content. It is generated by concatenating strips (typically from the center) of every video frame.

Cuboid shaped video icons constitute another category of visualizations. Video Streamer [8] belongs to this category. It represents video as a three-dimensional block of images, flowing in distance and time. Such a cube reveals temporal aspects of video. The flowing

patterns that appear on the sides of the cube can capture editing rhythm, shot boundaries and various camera motions. It is useful for reviewing and selecting video segments, and to gain perspective in time.

The SeeSoft [7] visualization was originally designed for software development. It uses a strip visualization where each row represents *one line* of text and the column represents, in a compact way, all the characters in a line. Such compact strip column originally showed programming features, such as, which programmers modified which lines of code. This strip visualization was subsequently extended to display text in documents. However, it has not been applied to video and multimedia content.

Scene Transition Graphs [24] attempt to capture the semantics of a video by displaying the underlying story structure. However, it has not been extended to video collections. In [15] the authors use a cone-tree-like visualization of a keyframe hierarchy. This approach aims to reduce browsing time per individual video and emphasizing fast sequential and hierarchical navigation.

Our visualization approach is novel in that it provides a means of navigating and representing any linear data collection; we have specifically focused on multimedia collections and its associated metadata.

Summary

We presented a method to visualize and navigate video data, called the movieDNA, which builds both on our work on Context Lenses as well as on our work with the CueVideo system. The movieDNA assumes a segmentation of video data and extraction of features of interest. The DNA itself is a segment based visualization of the occurrence of the extracted features. It is interactive: by rolling the mouse cursor over the DNA, users can brush through the video, pulling up detailed meta information on each segment. While the movieDNA is well suited to navigate moderate length videos, it is not sufficient to visualize and navigate very long videos or large collections of videos. Therefore, we generalized the movieDNA to a hierarchical movieDNA that permits brushing through larger collections of video data. The key to this generalization is a segmentation of the long video into large-scale segments, for each of which a movieDNA is generated. The data of these movieDNAs is reduced in screen size by generating aggregated DNAs, which are then used in the higher levels of the hierarchical movieDNA. This approach works only if this aggregation step is able to convey enough information

about the lower levels in the navigation tool to make an informed decision as to where to explore further. We employ the concept of "information scent" from information foraging theory to achieve this requirement.

The movieDNA and the hierarchical movieDNA are tools that can be used to visualize any kind of linear data, not just video data. We describe a number of applications we foresee for this tool or that we are currently experimenting with. The hierarchical movieDNA permits users to get detailed information about small segments of linear data while at the same time maintaining information on context and location in the overall collection. This means users can focus on details while keeping an overview of the bigger picture which distinguishes the hierarchical movieDNA from most other approaches.

Acknowledgments

We would like to thank the WBI team at the Almaden Research Center for their support and feedback while developing our new visualization. Special thanks go to Dan Russell for his support, discussions and encouragement to pursue this work.

References

- [1] Amir, A., Ponceleon, D., Blanchard, B., Petkovic, P., Srinivasan, S., and Cohen, G., "Using Audio Time Scale Modification for Video Browsing", *Proc. of Hawaii Int. Conf. on System Sciences, HICSS-33*, HI, Jan. 2000.
- [2] Arman, F., Depommier, R., Hsu, A. and Chiu, M.-Y., Content-based Browsing of Video Sequences. *Proceedings of ACM Multimedia*, 1994, p. 97-103.
- [3] Davis, M., "Media Streams: An Iconic Visual Language for Video Annotation." In: *Proceedings of 1993 IEEE Symposium on Visual Languages*, Bergen, Norway, IEEE Computer Society Press, 196-202, 1993.
- [4] Dieberger, A. and Russell, D.M., Context Lenses – Document Visualization and Navigation Tools for Rapid Access to Detail, submitted to ACM CHI 2000.
- [5] Ding, W., Marchionini, G., and Tse, T., Previewing Video Data: Browsing Key Frames at High Rates Using a Video slide Show Interface. In *Proceedings of the International Symposium on Research, Development and Practice in Digital Libraries (ISDL'97)*, p. 425-426, 1997.
- [7] Eick, S.G., Graphically displaying text, *Journal of Computational and Graphical Statistics*, 3(2):127-142, June 1994.
- [8] Elliot, E., and Davenport, G., Video Streamer, in *Proceedings of CHI'94*, Vol. 2, p. 65-66, Boston, MA, 1994.
- [9] Gupta, A. and Jain, R. Visual information retrieval. *In Communications of the ACM* 40, 5 (May. 1997), p. 70 - 79. Also see URL at <http://www.virage.com/>.
- [10] Hearst, M.A., TileBars: Visualization of terms distribution information in full text information access. *In Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems*, p. 59-66, Denver, CO, May 1995.
- [11] Hearst, M.A., User Interfaces and Visualization, edited by Baeza-Yates, R., and Ribeiro-Neto, B., *Modern Information Retrieval*, Addison-Wesley, ACM Press, New York, January 1999.
- [12] Kobla, V., Doermann, D., and Faloutsos, C., VideoTrails: Representing and Visualizing Structure in Video Sequences. *Proceedings of ACM Multimedia*, 1997, p. 335-356, 1997.
- [13] Komlodi, A., and Marchionini, G., Key frame preview techniques for video browsing. *Digital Librarys 98* Pittsburgh, PA , p. 118-126.
- [14] Lienhart, L., Pfeiffer, S. and Effelsberg, W. Video Abstracting, in *Comm ACM*, Dec 1997, pp. 55-62.
- [15] Manske, K., Video Browsing Using 3D Video Content Trees, *New Paradigms in Information Visualization, NPIV'98*, Bethesda, MD, ACM p. 20-24.
- [16] Pirolli, P., Computational Models of Information Scent-Following in a Very Large Browsable Text Collection, in *CHI'97*. 1997, ACM Press: Atlanta, GA. p. 3-10.
- [17] Ratakonda, K., Sezan, M.I., and Crinon, R., "Hierarchical Video Summarization," in *Visual Communications and Image Processing '99*, Proc. of SPIE vol. 3653, San Jose, CA, 1999.
- [18] Srinivasan, S. and Petkovic, D., Phonetic Confusion Matrix Based Spoken Document Retrieval. *In Proceedings of SIGIR-2000*. July 2000, Greece.
- [19] Srinivasan, S., Petkovic, D., Ponceleon, D. and Viswanathan, M. Query Expansion for Imperfect Speech: Applications in Distributed Learning. In *CBAIVL-2000*, IEEE Workshop on Content-based Access of Image and Video Libraries, Hilton Head Island, SC.
- [20] Syeda-Mahmood, T. and Srinivasan, S. Detecting Topical Events in Audio. To appear in *Proceedings of ACM Multimedia 2000*, LA, CA.

[21] Taniguchi, Y., Akutsu, A. and Tonomura, Y. PanoramaExcerpts: Extracting and Packing Panoramas for Video Browsing, in *Proceedings of MM'97*, pp. 427, ACM Press, November 1997.

[22] The CueVideo Toolkit Package Version 2.0. See URL. www.almaden.ibm.com/cs/cuevideo

[23] Ueda, H., Miyatake, T., Sumino, S., and Nagasaka, A., Automatic Structure Visualization for Video Editing. INTERCHI 93, April 1993, p 137-141.

[24] Yeung, M., Yeo, B-L., and Liu, B, Video Browsing Using Clustering and Scene Transitions on Compressed Sequences. In *Proceedings, IS&T/SPIE Multimedia Computing and Networking*, p. 399-413, 1995.

[25] Wactlar, H., Christel, M., Gong, Y. and Hauptmann, A. Lessons Learned from Building a Terabyte Digital Video Library. In *IEEE Computer*, Vol. 32, Number 2, Feb 1999.

Appendix:

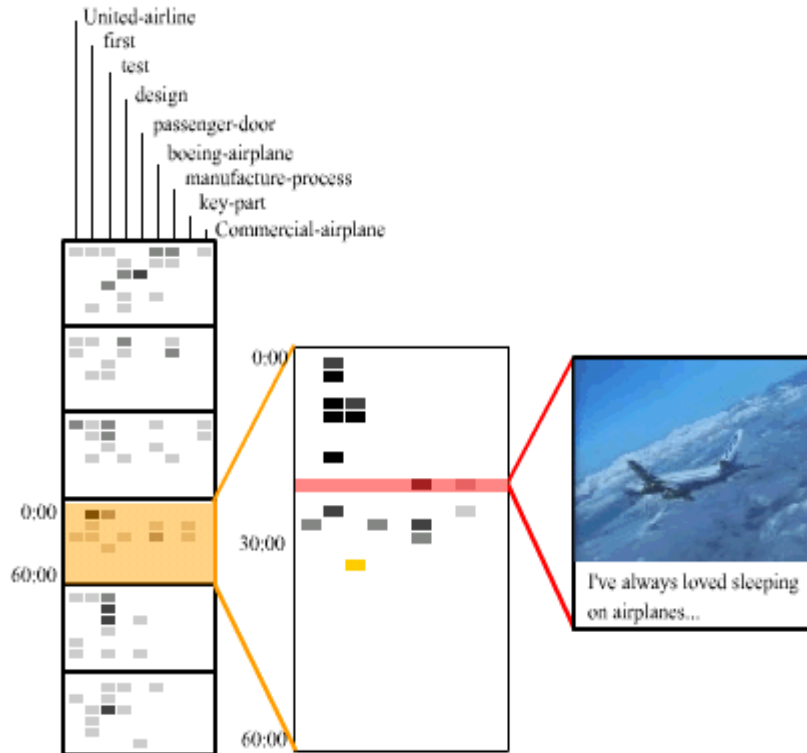


Figure 5: This hierarchical movieDNA uses automatically extracted top n-grams from the speech transcript. In this case, the aggregated DNA on the left (top level) shows aggregated incidence matrices for each of the 6 videos. The full incidence matrix for each video, generated through phonetic matching of features with the speech transcript, is shown as movieDNA on level 2.