RULE-BASED SEMANTIC SUMMARIZATION OF INSTRUCTIONAL VIDEOS

Tiecheng Liu and John R. Kender

Department of Computer Science
Columbia University
New York, NY 10027
{tliu, jrk}@cs.columbia.edu

ABSTRACT

We present a new content-based approach to summarize instructional videos. We first redefine "scene" in instructional videos. Focusing on one dominant scene type, that of handwritten lecture notes, we define semantic content as "ink pixels", and present a low-level retrieval technique to extract this content from each frame with consideration of various occlusion and illumination effects. "Key frames" in this video genre are redefined as set of frames that cover the semantic content, and the fluctuating amount of visible ink is used to drive a heuristic real-time key frame extraction method. A rule-based method is also provided to synchronize key frames with audio. We extend our method to the extraction of key frame hierarchies. We show its application to a 17-minute (30K frames) instructional video sequence, resulting in seven key frames. These techniques create tunable instructional summaries over a wide and dynamically varying range of compression factors.

1. INTRODUCTION

Recent research has led to many approaches[4, 9, 11] on understanding and summarizing highly structured and professionally edited commercial videos. Typically, these videos have rather short camera shots (on the order of four seconds [6]), well-defined scene changes, and visually appealing changes in content and cinematography. These explicit and implicit rules of construction are a great aid in the automated analysis and summary of such videos [5, 7, 10]. However, virtually none of these rules apply to most instructional videos. The "candid classroom" sequences are typically produced by lightly trained staffs who do poor camerawork and almost no editing. Most importantly, shots are very long, and content is best described as "illustrated radio": most of the visual information of a one-hour lecture often is compressible to no more than two dozen slides. Enormous semantic compression of these video sequences is possible. A hierarchy of such key frames, together with extracted gestures and other information, can be described in a language like that suggested by the MPEG 7 standard [2], and transmitted by use of intelligent background prefetching to clients with different connection bandwidths.

The general definition of "shot" as a contiguous segment of video captured by a single camera loses much of its utility in instructional videos. Usually, there are only a small fixed number of cameras in fixed positions and they typically are allowed to dwell long and continuously on the instructor, the board, or other media regardless of their changes in content. Therefore, analysis and summarization of such videos based on semantic content [1] rather than syntactic structure is critical. (Similar scenario exists in presentation videos in which the visual content and audio can be combined together to detect topical events [3].) We conceive an instructional video consists of instructional "scenes", each of which can be summarized by a single key frame. With this definition, we find that there are usually 50-100 instruction scenes in a typical class meeting of one hour length.

We have investigated a variety of instructional videos, including sample videos from five different courses in five different subjects. In instructional videos, scene has been redefined to indicate a clearly identifiable background or foreground difference in content. Our analysis noted nine scene types, among which the real-time handwritten lecture notes scene type accounts for a substantial subset of frames in instructional videos, and is similar in content and problems to that of blackboard work, pre-drawn hand slides, and powerpoint slides. We report on our methods for extracting key frames from this scene type, with the expectation that the other scene types will eventually be shown to be similar. Our design foresees the development of a scene detector to detect the scene type, and of different scene processing modules with each module processing only one type of scene. The key frames of different scenes are finally combined together to generate a comprehensive visual summarization and effective description of the video.

0-7803-7622-6/02/$17.00 ©2002 IEEE

2. LOW-LEVEL CONTENT RETRIEVAL

Rarely is there a clear single view of an entire hand-drawn slide. The first image processing steps focus on minimizing low-level disturbances: we heuristically remove irrelevant areas such as those imaging the table, hand, pen, or their shadows, and extract and enhance the areas that have only paper and ink. Then we calculate the number of ink pixels, which we use as a heuristic measure of the instructional content. Applying this process on every frame in such a video sequence, we get a (noisy) continuous measure of ink, which roughly indicates the change of content. Based on changes of this content curve, we heuristically derive key frames and a key frame hierarchy of the video sequence.

To extract the text area from video frames efficiently, we use a block processing approach. Before extracting content from video, we first sample several 32x32 pixel blocks in different positions of a frame to get the mean and variance of color of the paper background, which is always characterized by very small variance around "white". Then we divide a frame into blocks of size 16x16 and classify these blocks into three categories – paper blocks, irrelevant blocks, and uncertain blocks based on the portion of paper background color pixels in a block (see figure 1). Paper blocks have only paper and/or text, and irrelevant blocks have images of the instructor, pen, or table. Uncertain blocks are those that fail this initial coarse filter, and they are further processed by more sophisticated and expensive techniques, such as using a region growing technique to extend the blocks around a pen to cover the whole pen. For each frame, we get rid of irrelevant areas and calculate the number of ink pixels. These form a content curve (as shown in figure 3), which measures content change of an instructional video.

3. KEY FRAMES IN INSTRUCTIONAL VIDEOS

Key frames are usually defined as a set of frames that represent the original video within an acceptable degree of accuracy. In instructional videos, we redefine key frames in this video genre as a set of frames which cover the semantic content of the video. There are three requirements for a frame being a key frame. Firstly, a key frame must be a clean frame: that is, it must have minimal instructor occlusion and not in motion. Secondly, the content of a key frame must not be included in the content of its successor clean frame. Finally, a key frame must contain significantly different content than its predecessor clean frame.

3.1. Extraction of key frames

First we introduce some terms to describe our key frame extraction procedure. Let $T(\cdot)$ represent our low-level content retrieval operation. $T(f) = f^*$ means the transformation of the original frame f to f^* by getting rid of irrelevant areas

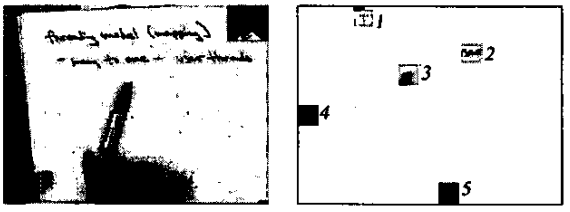

Figure 1: Block types in handwritten lecture notes video sequence. The right image shows some blocks sampled from the left image. Block 2 is a paper block; 1 and 3 are uncertain blocks;4 and 5 are irrelevant blocks.

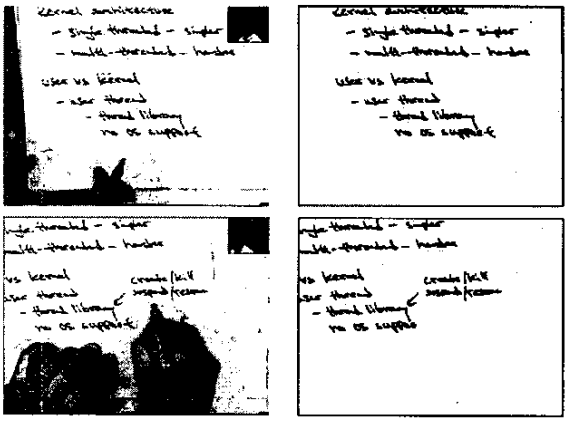

Figure 2: Low level content retrieval. The right images are transformed from left images by getting rid of irrelevant areas and enhancing the images.

of f. Let $\| \cdot \|$ be the operation of counting the number of ink pixels. Using the number of ink pixels as a heuristic measure of the semantic content, $\| T(f) \|$ indicates the amount of content in a frame f. $\| T(f_i) \ominus T(f_j) \|$ is defined as the symmetric difference of the content of f_i and f_j:

$$\| T(f_i) \ominus T(f_j) \| = \| T(f_i) \setminus T(f_j) \| + \| T(f_j) \setminus T(f_i) \|$$

where $\| T(f_i) \setminus T(f_j) \|$ is the number of ink pixels that are in frame f_i but not in frame f_j. Considering noise in real videos, we select a small threshold ϵ, if $\| T(f_i) \setminus T(f_j) \| < \epsilon$, the content of f_i is considered to be part of that of f_j; if $\| T(f_i) \ominus T(f_j) \| < \epsilon$, f_i and f_j are considered having same semantic content. Let $\| T(f_i) \cap T(f_j) \|$ represent the common semantic content of f_i and f_j. It is obvious that

$$\| T(f_i) \setminus T(f_j) \| = \| T(f_i) \| - \| T(f_i) \cap T(f_j) \|$$

Because one slide may not be exactly in the same camera position in two frames, we calculate the common semantic content (number of ink pixels) of f_i and f_j by translating and rotating these two frames to maximize the number of

common ink pixels. Combined with $T(\cdot)$ and $\| \cdot \|$ operators, we get $\| T(f_i) \setminus T(f_j) \|$ and $\| T(f_i) \ominus T(f_j) \|$.

According to our definition of key frames, we have three rules for a frame (f_j) being a key frame:

1. (Local change) $\| T(f_m) \ominus T(f_j) \| < \epsilon$, for $j - t \leq m < j$, where t is a threshold. A frame f_j is a clean frame if there is no significant content change for a designated period of time.

2. (Content Uniqueness) $\| T(f_j) \setminus T(f_k) \| > \epsilon$, where f_k is f_j's successor candidate key frame.

3. (Significant Change) $\| T(f_j) \ominus T(f_i) \| > \delta$, where f_i is f_j's predecessor clean frame, δ is a threshold of content change.

We combine the methods for measuring content and extracting key frames into a one-pass real-time algorithm. For each new frame, we calculate the number of ink pixels and apply the first rule for clean frame detection. If the new frame is not a clean frame, we continue to process the next frame. Otherwise we insert the new clean frame into a video buffer which hold a number of previous clean frames, then apply the other two rules to further detect key frames. Assume f_i, f_j and f_k are three contiguous clean frames in the video buffer. If f_j satisfies the second and the third conditions, we choose it as a key frame. If f_j fails, it indicates that either the content of f_j is part of that of other key frames, or the content change is not significant enough; in this case f_j can not be a key frame and it is removed from the video buffer. Using this technique, a key frame is more likely to occur in a temporal stable segment of a local maximum content value, which always corresponds to the end of one slide just before starting a new slide, or before an instructor moves a slide out of the captured screen (as shown in figure 3).

It is possible to create a hierarchy of key frames by choosing different content change threshold δ in the process above. We have incorporated the low-level instructional video content extraction technique into an existing time-constrained video buffer model [8] to achieve a tree-like key frame indexing structure, which provides compressed videos at different ratios to clients through dynamically changing bandwidth connections.

3.2. Synchronizing key frames with audio

When displaying key frames as a version of compressed instructional videos, we need to detect the appropriate time to display a key frame with respect to the uncompressed audio stream. Here we use a rule-based model to detect the appropriate temporal boundaries between key frames. Assume frame f_i corresponds to time t_i in original videos. The temporal boundary between two adjacent key frames f_i and f_j is calculated based on the following testing:

1. (New Slide) If there exists a frame f_k between f_i and f_j such that $\| T(f_k) \| < \epsilon$ and $\| T(f_m) \| > \epsilon$ for all $i \leq m < k$, f_k is more likely to be the starting frame of a new slide. In this case, we start to display key frame f_j at time t_k.

2. (Developing Slide) If the "content curve" is almost increasing between f_i and f_j, that is, if the number of frames that increase semantic content is much more than that of decreasing content, this segment of "content curve" more likely indicates the development of one slide. In such case, we display key frame f_j as early as possible (at time t_{i+1}).

3. (Other cases) We use a criterion of minimum content sampling error to determine appropriate temporal boundaries. Define the content sampling error of displaying key frame f_j at time t_m ($t_i < t_m < t_j$) as:

$$err(i, j, m) = \sum_{p=i+1}^{m-1} \| f_i \ominus f_p \| + \sum_{p=m}^{j-1} \| f_j \ominus f_p \|$$

We display key frame f_j at the time t_k such that

$$err(i, j, k) = \min_{i<m<j} \{err(i, j, m)\}$$

In figure 3, the temporal boundary between key frames b and c belongs to the first case and the temporal boundary between key frames a and b belongs to the third case.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

We developed a software tool (shown in figure 4) for real-time content retrieval and key frame extraction of instructional videos which have handwritten slide scene type. Our algorithm is one pass and we achieved a performance of 20 frames/second on a Pentium III 500 computer. Experiments on a 17-minute 30,000 frame video sequence result in 7 key frames, which cover the whole instruction content of this part of video. We show four of the seven key frames and the corresponding part of ink pixel curve in figure 3. We also developed a key frame hierarchy, which was used both for indexing and compression. Subjective evaluation of the synchronization of key frames with audio gave satisfactory results.

As demonstrated, server-based semantic compression followed by client-based image reconstruction can lead to a further three orders of magnitude (1000 times) compression over standard MPEG signal-based compression. Future work would include the completion of the other special-purpose scene content modules, and their integration into a working system that not only has a server-side semantic compressor, but also has multiple client-side key frame and gesture interpreters.

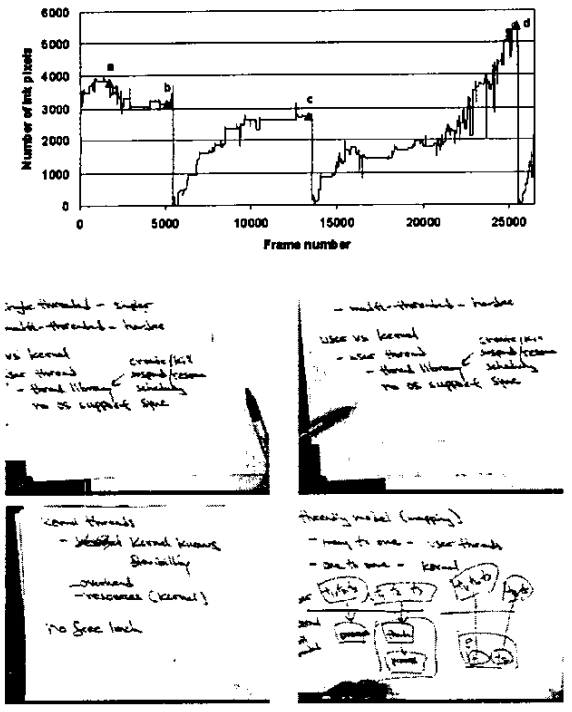

Figure 3: Instructional video key frame selection. The figure above shows the number of ink pixels of a segment of an instructional video. The four marks "a"-"d" are the positions of the four key frames extracted from this segment of the video.

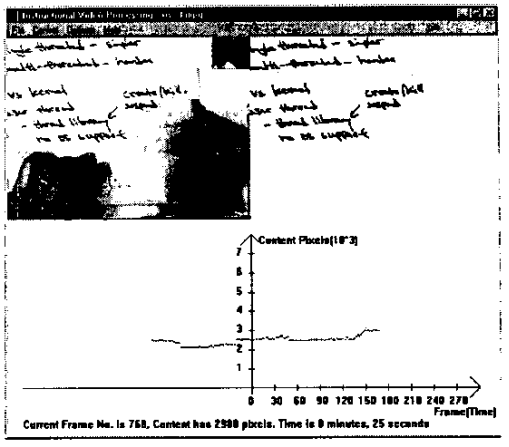

Figure 4: Research tool for real-time content retrieval of the scene of handwritten slide in instructional video. The coordinate below indicates the number of ink pixel of current frame, which is used as a heuristic measure for instructional video summarization.

5. REFERENCES

[1] Edoardo Ardizzone and Mohand-Said Hacid. A Semantic Modeling Approach for Video Retrieval by Content. In *IEEE International Conference on Multimedia Computing and Systems*, pages 158–162, 1999.

[2] Shih-Fu Chang, Thomas Sikora, and Atul Puri. Overview of the mpeg-7 standard. In *IEEE Trans. on Circuits and Systems for Video Technology*, June 2001.

[3] Tanveer FathimaSyeda-Mahmood and S. Srinivasan. Detecting topical events in digital video. In *Proceedings of ACM Multimedia 2000*, 2000.

[4] S. Sull H. S. Chang and Sang Uk Lee. Efficient video indexing scheme for content-based retrieval. In *IEEE Trans. on Circuits and Systems for Video Technology*, pages 1269–1279, Dec. 1999.

[5] J.R. Kender and B.L. Yeo. Video scene segmentation via continuous video coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 367–373, Jun. 1998.

[6] J.R. Kender and B.L. Yeo. On the structure and analysis of home videos. In *Proceedings of the Asian Conference on Computer Vision*, Jan. 2000.

[7] T. Liu and J.R. Kender. A hidden markov model approach to the structure of documentaries. In *Proceedings of the IEEE International Worksop on Content-based Access of Image and Video Databases*, Jun. 2000.

[8] T. Liu and J.R. Kender. Time-constrained dynamic semantic compression for video indexing and interactive searching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Dec. 2001.

[9] M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *Proceedings of the IEEE International Worksop on Content-based Access of Image and Video Databases (ICCV'98)*, 1998.

[10] M. Yeung and B.L. Yeo. Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition*, pages 375–380, 1996.

[11] Thomas S. Huang Yueting Zhuang, Yong Rui and Sharad Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *IEEE International Conference on Image Processing*, pages 866–870, 1998.