# A Browser for Summarized Multiple Videos

Saki Kasamatsu Takayuki Itoh

Graduate school for Humanities and Sciences, Ochanomizu University

E-mail:{saki,itot}@itolab.is.ocha.ac.jp

## 1. Overview

Recent digitalization of multimedia contents evolved video recording and streaming technologies, and amount of videos stored in personal computers have been therefore increasing. It is often time-consuming to look for demanded video files, consult the content of each file, or compare the contents of multiple video files, from the large number of stored video files.

There have been many works on video summarization, and visualization of such summarized videos. Some of the works developed browsing tools to visualize the video summaries, including features such as assigning different sizes to each keyframe by calculating importance of scenes, or enabling control of summary lengths by developers. However, most of visualization works did not focus on representation of timing of scenes and total lengths of the video files. Moreover, those works are just available to consult the content of single video file; they are not suitable for visual comparison of multiple video files.

The paper presents a technique for visualization of summarized multiple video files. The technique places keyframes of the summarized video on a horizontal time axis as shown in Figure 1. Also, the technique vertically places the time axis of multiple video files. It makes easier to grasp the story of videos, and compare the contents of multiple videos. The technique provides a user interface for level of detail control, which adaptively adjusts the number of displaying keyframes per unit time. It displays more keyframes with zoom in operation, so that users can understand the detailed story of the video. On the other hand, it displays less keyframes with zoom out operation, so that it can display more number of summaries of video files, or longer stories on one screen.

In our approach, we chiefly focus on quickly getting an overview of video files which the users have seen once, or comparing the contents of accumulated multiple video files. Here, we need to save the screen space that each file takes up, to visualize more video summaries on a screen. To solve the problem, we present a visualization interface that allows us to interactively control the number of displayed keyframes for each video file. We think this approach is particularly effective for TV dramas and films, because they tend to have story changes and events more frequently. Users may easily understand the story of such video files by looking at the less number of impressive keyframes, because contents of frames vary in those kinds of videos.

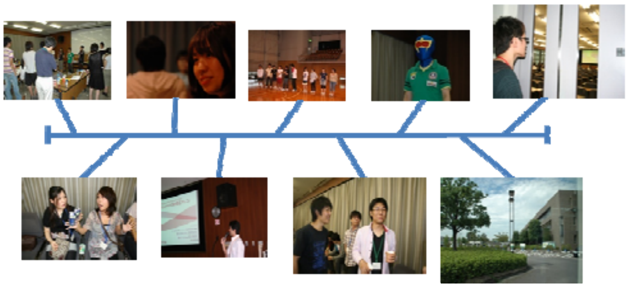

Figure 1: Keyframes on a time axis.

This paper is organized as follows. Section 2 introduces related work. Section 3 describes the architecture of our technique. Lastly, Section 4 describes conclusion and discusses future work.

## 2. Related Work

Video summarization has been an active research topic, due to the saturation of video files. Shot boundary detection is one of the most important technical issues to divide videos into shots in the video summarization techniques. Miyamoto et al. have developed the shot boundary detection system using optical flow, HSV color data, and edge difference [1], in addition to the typical methods based on pixel difference. Other approaches [2][3] have adopted the support vector machines to learn from recognition patterns.

Keyframe selection is another important technical issue to create a concise and meaningful representation of video in the video summarization techniques. While typical keyframe extraction algorithms select keyframes based on clustering of frames [4], Smith et al. have developed the automatic detection system of important keyframes, by calculating importance weight of each video shot based on caption analysis and face detection results [5]. In addition, Hamada et al. have developed system which detects specific motions from the video, to automatically extract the important scenes of a cooking video [6]. Another relevant work has analyzed construction of shots to extract specific scenes from sports video [7]. These approaches have focused on specific kinds of videos, to efficiently obtain good summarization results.

There have been also several works on visualization of summarized videos. Video Manga [8] is one of the works most relevant to ours, which automatically resizes keyframes and packs them into one-page pictorial video summaries, resembling comic books. Another work has accomplished to visualize large-scale news videos, which enables control of sizes and places of the keyframes according to their importance weight [9]. These advanced techniques can be integrated into traditional video browser tools [10] [11]. These visualization techniques mainly focus on representation of single video file. They are useful for understanding the story or finding the specific scenes from the specific video file. However, they are not always sufficient to find the demanded scenes or compare multiple video files in one display space. Also, users are unable to control the length of summary to look minutely or simply, since their techniques do not feature interactive level of detail control.

## 3. Processing Flow

This paper presents a technique for visualization of summarized videos, by placing keyframes on a horizontal time axis. This section describes the architecture of our technique in detail.

Our current study applies typical keyframe selection algorithm, and we mainly focus on interactive visualization of multiple video summaries. We assume our technique is especially suitable for videos which have frequent scene changes, such as dramas and films.

### 3.1. Video Summarization

The technique first summarizes the video data by keyframe selection. Our implementation is based on an existing work presented in [8]. This process first divides a video into "shots" that are consecutive series of frames in the video constituting a unit of action. It divides the video into shots by detecting shot boundaries, where the difference of pixel values between two adjacent frames is large enough. It then selects representative frames of each shot. Our current implementation abstracts the temporally central frame in a shot as its representative frame. The technique then generates clusters of similar representative frames of temporally adjacent shots, by calculating averages of $YC_bC_r$ color values for each frame. We call the clusters "scenes", which denotes larger semantic series of frames in a video. It finally selects the representative frames for each cluster, as "keyframes" of the video, which have the closest $YC_bC_r$ color value to the averages of each cluster.

### 3.2. Importance Score of Keyframes

The technique then calculates importance score of each keyframes. While several existing video summarization techniques calculate the importance score based on the semantics, our implementation considers the appearance of the images as well. We think that frames are important if it can be clearly recognized even they are zoomed out. Following are the elements used to calculate importance score of the keyframes:

**Face detection results.** We treat that a frame is important if faces of persons appear in the frame. Our implementation assigns higher scores when the number of the face detected in the frame increases. Furthermore, our implementation also takes sizes of faces into account for importance calculation. Close-ups are also important because they are comprehensive even when the frames are zoomed out.

**Saturation.** We presume that a frame is important if saturation average of the image is higher. Such frames are still comprehensive even when they are zoomed out. Our implementation calculates averages of saturation of keyframes from $YC_bC_r$ color values used in Section 3.1.

**Length of a scene.** Our implementation supposes that length of a scene is proportional to size of a cluster which the keyframe belongs to. We treat that a frame is important if the scene of the frame is long. We do not want to miss such scenes from the visualization results.

**Time Position of a frame.** Importance score also varies depending on the time position of the keyframe in the whole video. It is an application-specific matter: we need to adjust the importance calculation using the time position according to the genre of videos. For example, we think users may want to focus on latter parts of drama videos because they often include climaxes of the stories. In this case users can adjust to make the importance score higher. In contrast, we think former parts of videos tend to contain less important scenes. In this case we can decrease the importance score for the frames which belong to former parts of a video.

### 3.3. Sizing of Keyframes

The technique then calculates sizes of each keyframe according to their importance scores, to emphasize important keyframes by assigning larger sizes. Our implementation calculates the sizes of the keyframes from the proportions of their importance score to the maximum score of the whole video, so we can normalize the importance score among multiple video files. It then assigns one of several different sizes to a given keyframe from the calculated proportion. Our implementation uses fixed proportion thresholds to determine the sizes.

Figure 2 shows an example of the visualization result of multiple video files. Various sizes are assigned to the keyframes according to their importance score, so larger frames take users' attention to the important scenes. A set of the keyframes lined up horizontally along a time axis represents a summary of one video file. Since each one summary takes smaller space comparing to the existing browsers for video summaries, our implementation can line up and show more than one summary of video file at the same time.

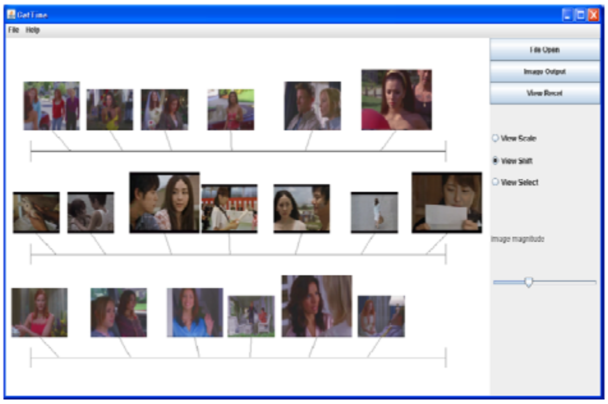


Figure 2: Size-controlled keyframes of multiple video files.

## 3.4. Level of Detail Control

One of the main features of the visualization interface on our implementation is realizing level of detail control by users' zooming operations. Here we define the level of detail of video summary as the number of displayed keyframes per unit time. The technique adjusts the number of displayed keyframes by horizontal mouse operation. The number of displayed keyframes increases according to zoom in operation, as shown in Figure 3(b), so that users can look the detail of the video contents. On the other hand, the number of keyframes decreases as users zoom out the summaries, so that they can look the simplified summary of the video.

While zooming out, the technique reduces the displaying keyframes by eliminating less important frames. The technique automatically deal with the process, by comparing two adjoined keyframes that are timely close, and eliminating the frame which has the less importance score. While zooming in, the technique revives the eliminated keyframes.

As well as horizontal zoom operation, the technique supports vertical zoom operation. As users zoom out the summaries vertically, a set of keyframes per each video file is alternately lined up in two rows, above and under the time axes, as shown in Figure 3(c). Placing keyframes in two rows achieves an efficient visualization even when the number of keyframes increases. Also, vertical zoom out operation simultaneously controls the number of displaying video file.

Summaries of various videos often yield different numbers of keyframes. To avoid gaps of the numbers of displayed keyframes between different videos, the technique unifies the number of keyframes per one video among multiple files, by fixing it according to the length of file. If it selects larger number of keyframes from a specific video, it eliminates less important keyframes to unify the number of displayed keyframes with other videos.

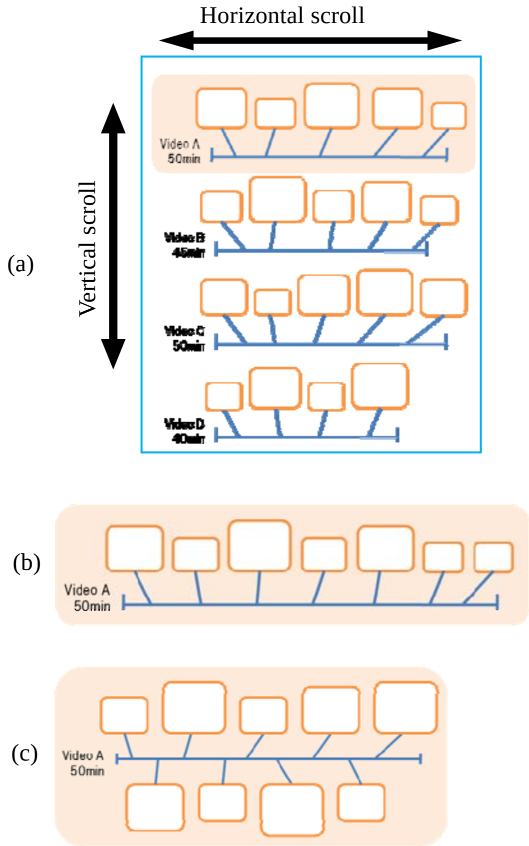


Figure 3: Level of Detail Control: Keyframes of each video(a) increases as (b) when they are zoomed in horizontally. Keyframes are rearranged as (c) when they are zoomed in vertically.

## 4. Example

Figures 2 and 4 are the results of implementation of the proposed technique described in Section 3. As a user has a vertical zoom in operation, in the condition of Figure 2, keyframes are rearranged to form two rows per video, as shown in Figure 4. This interactive interface enables users to get a glimpse at contents of multiple videos, or to consult the detailed content of a video, depending on users' purpose.

## 5. Conclusion and Future Work

In this paper we have presented a visualization interface for summarized multiple video files, which realizes interactive level of detail control. We think the technique can assist users to easily understand or explore the contents of accumulated video files. Our study differs from many of existing visualization techniques which focused on summarization of single video file, by concentrating on visualization of multiple video files.

Following are our potential future works. Firstly, we would like to evaluate and discuss effectiveness of our importance calculation technique, and consider new elements. Our current implementation calculates importance score only from the elements of the video itself, but the system will get more effective visualization result if it can calculate while comparing with other videos to concern differences and similarities between them.

We are also working on improvement on visualizing keyframes effectively with less space. Current our system places keyframes simply on their corresponding positions on the time axis; however, the space utilization will be better if the system adaptively modifies the positions on the time axis.

Also, we would like to enhance the presented browser as a video player. For example, it is useful to support click operations so that users can start the streaming from the clicked position of the videos.

Another important issue of our implementation would be applying other kinds of videos. We are now working on visualizing a biological simulation video of protein movement.

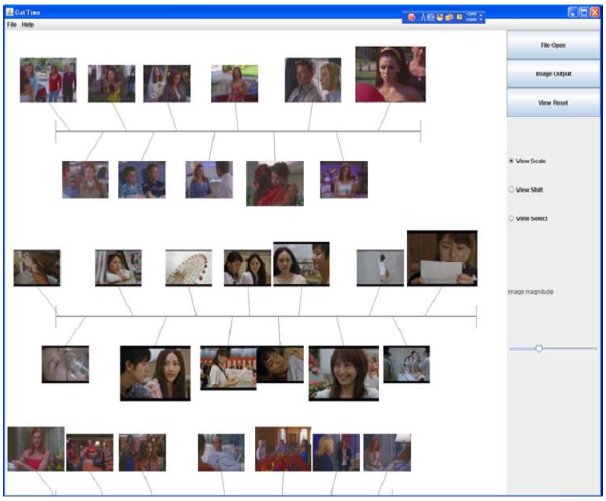

Figure 4: Keyframes lined up in two rows per video.

## References


[1] Y. Miyamura, T. Nakamura, K. Shinoda, S. Furui, “Shot Boundary Detection Using Multi-stage SVM”, Meeting on Image Recognition and Understanding 2007 (MIRU 2007), IS-2-19, pp.815-820, 2007.

[2] K. Matsumoto, M. Naito, K. Hoashi, and F. Sugaya, “SVM-Based Shot Boundary Detection with a Novel Feature”. ICME 2006, pp. 1837-1840, 2006.

[3] Arnon Amir. The IBM Shot Boundary Detection System at TRECVID 2003.
http://www-nlpir.nist.gov/projects/tvpubs/tvpapers03/IBM_TREC03_SBD.slides.pd

[4] Y. Zhuang, Y. Rui, T.S.Huang, and S. Mehrotra, “Adaptive Key Frame Extraction Using Unsupervised Clustering,” Proceedings of ICIP ’98, Vol. I, pp. 866-870, Oct 1998.

[5] M. Smith, and T. Kanade, “Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), pp. 775-781, 1997.

[6] R. Hamada, K. Miura, I. Ide, S. Satoh, S. Sakai, H. Tanaka, “A Motion Based Automatic Abstraction of Cooking Videos”, Meeting on Image Recognition and Understanding 2002 (MIRU 2002), Vol. 2, pp.203-208, 2002.

[7] B. Li, and M. I. Sezan, “Event Detection and Summarization in Sports Video,” Content-Based Access of Image and Video Libraries, 2001. (CBAIVL 2001). IEEE Workshop, Dec. 2001, pp. 132-138.

[8] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, “Video Manga,” Proceedings of the Seventh ACM International Conference on Multimedia, pp.383-392, Orlando, United States, 1999.

[9] H. Luo, J. Fan, J. Yang, W. Ribarsky, S. Satoh, “Exploring Large-Scale Video News via Interactive Visualization”, IEEE Symposium On Visual Analytics Science And Technology (VAST2006), pp. 75-82, 2006.

[10] H. Zhang, C. Y. Low and S. W. Smoliar, “Video Parsing and Browsing Using Compressed Data,” Multimedia Tools and Applications, Vol. 1, Issue 1, pp. 89-111, March 1995.

[11] B. Yeo, and M. M. Yeung, “Retrieving and visualizing video,” Communications of the ACM, Vol. 40, Issue 12, pp. 43-52, Dec 1997.