

Creating Map-based Storyboards for Browsing Tour Videos

Suporn Pongnumkul

Computer Science and Engineering
University of Washington
Seattle WA 98195
suporn@cs.washington.edu

Jue Wang

Adobe Systems
801 N. 34th Street
Seattle, WA 98103
juewang@adobe.com

Michael Cohen

Microsoft Research
One Microsoft Way
Redmond, WA 98052
mcohen@microsoft.com

ABSTRACT

Watching a long unedited video is usually a boring experience. In this paper we examine a particular subset of videos, *tour videos*, in which the video is captured by walking about with a running camera with the goal of conveying the essence of some place. We present a system that makes the process of sharing and watching a long tour video easier, less boring, and more informative. To achieve this, we augment the tour video with a map-based storyboard, where the tour path is reconstructed, and coherent shots at different locations are directly visualized on the map. This allows the viewer to navigate the video in the joint location-time space. To create such a storyboard we employ an automatic pre-processing component to parse the video into coherent shots, and an authoring tool to enable the user to tie the shots with landmarks on the map. The browser-based viewing tool allows users to navigate the video in a variety of creative modes with a rich set of controls, giving each viewer a unique, personal viewing experience. Informal evaluation shows that our approach works well for tour videos compared with conventional media players.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Video browsing, Video summarization, Map, Storyboard.

INTRODUCTION

People frequently use camcorders to document their personal events such as weddings, birthday parties, travels, etc. Many videos are kept in storage and never shared with others or watched again. The reasons for this are: (1) the raw video is usually lengthy, containing large portions of low quality shots and thus is not ready to be shared; (2) manual video editing not only is time-consuming, but also requires special skills; and (3) watching an unedited long home video is painful, as it is hard for viewers to find interesting, high-quality parts of the video using a conventional video player.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'08, October 19–22, 2008, Monterey, CA..

Copyright 2008 ACM 1-59593-313-1/06/0010 ...\$5.00.

In this work we focus on ways to enhance one type of video that people often take when visiting new places: tour videos. A tour video is usually long, and casual, with a lot of camera motion. It is thus often unpleasant to watch in its original form. In addition, different shots taken in a tour video depict different locations. Conventional video players have controls related to time but not to location.

In this paper we propose an interactive system to make the process of sharing a long tour video and watching such a video easy and more fun. To help reconstruct the tour documented by the video, we propose a map-based storyboard user interface, on which different video shots taken at different places are directly tied to their corresponding locations on the digital map. Furthermore, a rich set of controls are provided to viewers, which allow them to interactively explore the tour in a variety of ways, for instance, directly jumping to the video shot that is taken at a specific landmark location on the map, or only watching coherent shots highlighted by the author, or even creating a virtual tour path that is different from the original one. Using our system each viewer can navigate the tour video in a unique way.

We also develop an interactive authoring tool which is able to automatically extract salient shots from the raw video using computer vision techniques, and generate a representative keyframe for each shot. The user can then quickly create a storyboard presentation by providing a map, and mapping the keyframes onto their corresponding locations. Additionally the user can easily select additional shots and add them onto the storyboard.

Our work has three key contributions: (1) a novel map-based storyboard as the user interface for browsing long tour videos; (2) an authoring tool to help users efficiently create such a storyboard given an input video; and (3) a rich set of controls on the viewing tool to allow viewers to experience the video in customized ways. User feedback shows that people quickly learn to use our system, and find that using such a system can greatly improve the experience of sharing and watching a tour video.

RELATED WORK

Previous work related to our system falls into three categories: video abstraction, video browsing, and media geotagging. We will briefly discuss representative techniques in each category.

Video Abstraction

Video abstraction visually summarizes a video so that it can be comprehended more rapidly. The output of video abstraction could be either a static image (known as static abstracts, or video summary), or a shorter version of the video (known as dynamic abstracts, or video skimming). Excellent reviews on these techniques can be found in [20, 7, 4].

Static abstracts are usually a small collection of salient images extracted or generated from the input video. Most previous systems extract meaningful keyframes from the input video and display them in a variety of ways. Komlodi and Marchionini [6] compare several types of keyframe display methods and discover improved object identification performance by using side-by-side displays over sequential displays. Tonomura et al. [18] propose two other static abstracts called the “VideoMAP” and the “VideoSpaceIcon”, which not only summarize the content of the video, but also visualize essential features and the spatio-temporal characteristics of the video in an easy-to-perceive manner.

Dynamic abstracts consist of image sequences (sometimes with audio) which are extracted from the original video with a considerably shorter length. Lienhart [8] and Pfeiffer et al. [15] cluster video shots hierarchically into meaningful units. Smith and Kanade [16] characterize video using both image and audio cues, by analyzing motion, language, and detecting face and text overlay. They use these features to identify the most salient sections to retain in their dynamic abstracts. The Hitchcock system [3] measures image unsuitability due to low brightness, or rapid or jerky camera motion. The valleys of the unsuitability function that are longer than a minimum length are defined as shots. The user can manually adjust the length of individual shots, and the unsuitability function is used to re-optimize the in- and out-frames of one or multiple shots.

In summary, dynamic abstracts offer a richer media experience than static abstracts whereas static abstracts can be observed nearly instantaneously. Given the fact that both methods have their own advantages, our system offers a hybrid abstraction by presenting both keyframe-based static abstracts and coherent shot-based dynamic abstracts. Our approach to detect the coherent shots is similar to the similarity matrix used in [1], but we utilize a different distance function.

Video Browsing

Many browsing techniques have been proposed in the literature. Here, we will only mention a few that are closely related to our work. An early map-based video browsing system is the movie-map [9], in which the streets of the city of Aspen were filmed. Videodisc players are used to play the captured images and synthesized images to simulate driving through the city. This work, however, is not designed to be easily edited.

Another interesting browsing technique is presented by Nam and Tewfik [14]. Their system automatically creates a dynamic abstract of an input video using an adaptive nonlinear sampling of the video, where the local sampling rate is directly proportional to the amount of visual activity in localized sub-shot units. A similar technique is used in [2], where

automatically-derived information is used to allow people to browse video nonlinearly by skipping or speeding through uninteresting parts. These are quite similar to one of the advanced viewing modes presented in our system, however our system does not limit the user to a single viewing mode and allows the user to browse the video in a variety of ways.

Geotagging

Geotagging is the addition of geographical identification metadata to media, either through location-aware devices such as GPS, or through manual editing. Here we will briefly summarize geotagging techniques on two closely-related media types: photographs and video.

There has been increasing interest in using geo-location information to facilitate photo browsing. The World-Wide Media Exchange [19] arranges images on an interactive 2D map. The online photo browsing system PhotoCompas [13] clusters images based on both time and location. Kadobayashi and Tanaka [5] present an interface for retrieving images using proximity to a virtual camera. Snaveley et al. propose a photo tourism system [17] for interactively browsing and exploring large unstructured collections of photographs of a scene using a novel 3D interface, where the viewpoint of each photograph is automatically estimated by using advanced computer vision techniques.

Video geotagging could be achieved by coupling a GPS device with the video data to provide complete location information at all times in the video. Dedicated hardware for this purpose is available in the market. For example, VMS-XAI¹ is an accessory that can be clipped on to a video camera to integrate GPS and video data collection. However, these devices are expensive and not suitable for all users. Another limitation of using GPS devices for video geotagging is the limited resolution of GPS signal. For video sequences capturing indoor activities such as the “Paul Allen Center tour” example used in this paper, the location information of each single shot can be missing or inaccurate.

An alternative approach for video geotagging that does not require dedicated hardware is to allow users to interactively tag their videos, and our system falls into this category. Recently Google Maps² allowed users to add various types of content, including Youtube³ videos, to specific geographic locations on its world map. Since then there have been experimental systems on organizing videos based on location, such as the Virtual Video Map⁴. However, users can only tag an entire video to one location rather than tagging different shots to different locations.

We are also aware that some users have explored the idea of geotagging portions of a video using the Google Map service. One example is shown on the web page of Claude Lelouch⁵, where a Ferrari was driven around in Paris with a stabilized video camera attached to the bumper. The video from the ride was coupled with a map which updated the route as the

¹<https://ecommerce.redhensystems.com/pc-30-7-vms-xai.aspx>

²<http://maps.google.com/>

³<http://www.youtube.com/>

⁴<http://www.virtualvideomap.com/>

⁵http://bhendrix.com/wall/Gmaps_GVideo_Mashup_Rendezvous.html

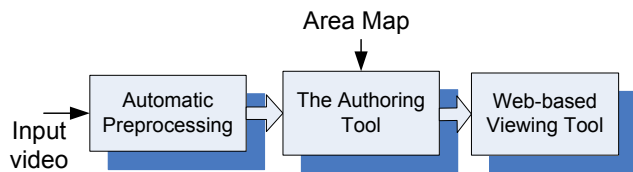


Figure 1: The flow chart of our system.

film progressed. Another similar example with a visualized tour path can be found in the South Bank Walk⁶. However, creating these examples involves a lot of manual work. In contrast, our system provides a much easier way to geotag different shots in a long tour video.

SYSTEM OVERVIEW

Figure 1 shows the flow chart of our system, which contains three major components: a preprocessor, an authoring tool, and a viewing tool. The system inputs are a video and an area map. The map is a digital image which represents the area map of where the video was shot. In order to create a storyboard presentation, the input video is first parsed by the preprocessor, which automatically extracts salient features from video frames, and uses them to identify coherent, high quality shots in the raw video. A representative keyframe is also automatically chosen for each shot as the controlling knob for the user.

In the authoring tool, the pre-processed video along with the digital map are presented to the user to create a storyboard presentation. The user simply drags the keyframes onto the map and pins them on proper landmarks, and the tour path is automatically reconstructed on the map. The user can remove the unwanted pre-chosen shots or additionally pick out new shots and keyframes and use them to refine the tour path until satisfied.

The user’s editing is then passed as an XML configuration file to the web-based viewing tool, which provides a rich set of controls to create a variety of ways to browse the video. Note that unlike existing video editing tools, our authoring tool works in a non-destructive fashion and does not modify the raw video, thus all the information is preserved and viewers have complete control over how much of the video should be played, which parts of the video should be played, and in what order they should be played. In other words, our viewing interface is fully customizable and is able to provide unique and personal user experiences.

AUTOMATIC PRE-PROCESSING

Given an input video sequence consisting of T frames I_t , $t \in [1, T]$, we first detect feature points in video frames using SIFT [10], Harris-Affine [12] and MSER [11] feature detectors. The local image information around the detected features is stored using the SIFT feature descriptor. SIFT features are commonly used local image descriptors based on the appearance of the object at particular interest points, which are highly distinctive, and robust to changes in illumination, noise, occlusion and minor changes in viewpoint. They have been extensively used in a variety of computer

⁶<http://homepage.ntlworld.com/keir.clarke/south/southbank.htm>

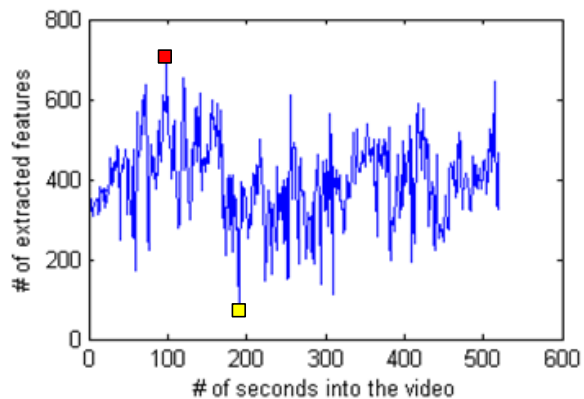


Figure 2: Top: Number of SIFT features extracted at different times of the “market tour” video, one of our example videos. Bottom: The frame with the highest number of features (red) has much higher quality than the frame with the lowest number of features (yellow).

vision and graphics applications, such as object recognition, image retrieval, image stitching, etc.

SIFT feature extraction is more sensitive to sharp, strong image edges and corners than smooth regions. If a video frame is taken with a steady camera, usually a larger number of feature points can be extracted from it. On the contrary, if a video frame is significantly blurred due to severe camera shake, the number of available feature points is usually small. Figure 2 shows the variation of the number of features we extracted from one of our example videos (one frame is selected from each second for feature extraction). It also shows that video frames containing more extractable features usually have much higher visual quality than those containing smaller number of features. Inspired by this observation, we use the number of extracted feature points as the quality measure of video frames. Specifically, for frame I_t , its quality is computed as

$$Q_t = \frac{\max N_f - N(F_t)}{\max N_f - \min N_f}, \quad (1)$$

where F_t is the set of extracted features, $N(F_t)$ is the number of extracted features, and $\max N_f$ and $\min N_f$ are maximum and minimum number of features extracted from a single frame in the whole video.

In our system, the feature points are used not only for quality measurement, but also for identifying coherent shots. If the same image feature appears in multiple frames, then its corresponding SIFT features extracted from these frames can be matched well. For two frames I_{t1} and I_{t2} , we compute the number of their matched features as $N(F_{t1}, F_{t2})$. If the

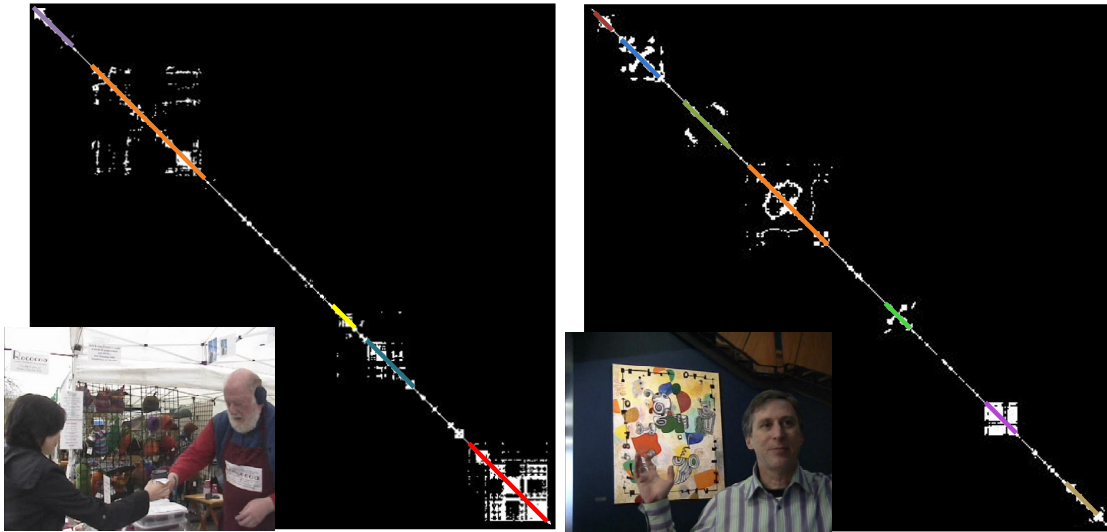


Figure 3: The coherence matrices computed for the “market tour” video (left) and the “Paul Allen Center tour” video (right). Automatically extracted coherent shots are overlaid as lines in different colors.

number is large, then the two frames usually have a large overlap on the scenes they captured. Otherwise if the number is small, the two frames then probably are not captured at the same scene. Using this method we can compute a $T \times T$ coherence matrix for the input video, where the (i, j) th element is $N(F_i, F_j)$, the number of matched features between frame i and j .

Figure 3a visualizes the coherence matrix computed for the “market tour” video, which clearly presents a structure of “chained blocks” along the diagonal axis. The blocks are caused by shooting the same scene with small and steady camera motions over a period of time, resulting in a relatively large time window where video frames are highly correlated with each other. We find that these blocks often directly correspond to the “interesting” parts of the video, where the camera person stopped walking or random panning and carefully focused on some interesting events, activities or scenes when capturing the video. On the other hand, the thin lines connecting these blocks often represent less interesting parts of the video, where the camera motions are large and irregular, resulting in less coherent and low quality shots which are less pleasant to watch.

Based on the coherence matrix, we demonstrate a simple algorithm to automatically pick out the interesting shots which are represented as blocks in the matrix. We project all the off-diagonal values in the matrix onto the diagonal line, and accumulate them to create a 1-D coherence histogram. We then smooth the histogram using a Gaussian kernel, and apply a threshold (which we set to be 0.3 times the maximum value in the histogram) to generate an initial set of shots. A postprocessing step is then applied to the initial set to remove shots that are too short in time, and to merge shots that are too close to each other into a longer one, to generate the final set of coherent shots.

Finally, in each coherent shot, we pick out the frame that has the strongest correlation with all other frames, and use it as

the keyframe for the entire shot. We also average the quality measures of all frames in a shot and use the average value as the quality measure for this shot.

Feature extraction and matching is a computationally expensive process. To speed up the procedure we down-sample the video both spatially and temporally for pre-processing. We only pick out one frame in every two seconds of the video, and reduce its size by a factor of 2 for feature extraction and matching. In this way the computational time can be dramatically reduced, and we are still able to get similar quality results compared with using the full sequence for pre-processing. In our current system it roughly takes about 1 hour to process a 30 minute video.

THE AUTHORING TOOL

Once the pre-processing is done, the automatically extracted coherent shots, their quality measures and keyframe indexes, along with the 1-D coherence histogram, are passed as an XML configuration file to the authoring tool.

Figure 4a shows the user interface of the authoring tool. The left panel is a video player which shows the input video with a set of common controls. The middle panel shows the list of keyframes suggested by the pre-processor, and the right panel displays the area map that is provided by the user.

To start creating a storyboard presentation, the user simply selects a keyframe in the keyframe list, and drags it over to the map, and places it roughly at the location where the keyframe was shot. As shown in Figure 4b, a pin shape appears on the map, which allows the user to adjust the location of the keyframe at a finer level. A thumbnail of the keyframe is also generated and placed on the map, whose position can also be adjusted for a better layout.

The user iterates in this way to add more keyframes onto the map. When a new keyframe is located on the map, a new section of the tour trajectory is automatically overlaid on the



Figure 4: The authoring tool of our system. (a) The layout of the UI. (b) Adding the first keyframe to the map. (c) Adding the second keyframe to the map automatically creates a piece of tour path.

map to connect the previous landmark with the current one, as shown in Figure 4c. Furthermore, when a new keyframe is added, its corresponding coherent shot is visualized as a colored segment in the timeline bar below the video player in the left panel, and its color matches with the border color of the thumbnail. This gives the user direct visual feedback on how the input video is represented by the coherent shots, and the length and the relative position in time of each shot.

The user is not constrained to only use the automatically generated suggestions. The user can add, modify, delete or merge coherent shots, as well as their representative keyframes. To add a new shot which has not been picked up by the pre-processor but is interesting to the author, he/she can simply use the timeline slider to find a frame that is within the shot, then click a button to tell the system to add a new shot. The system then analyzes the 1D coherence histogram computed at the pre-processing step, finds out a locally coherent shot that contains the user-selected frame, and generates a representative keyframe, which is added to the keyframe list.

Modifications of existing shots can be achieved by directly manipulating the color segments on the timeline bar. A shot can be shifted in time by dragging its color segment and moving it along the timeline bar. Its starting and ending time can be adjusted by dragging the two ends of the segment. If two shots are moved close enough to each other, they will automatically merge into one.

THE VIEWING TOOL

The viewing tool of our system is web-based, and thus can be easily opened in any Javascript-enabled web browser, as shown in Figure 5. Similar to the authoring tool, the left panel is a video player accompanied by a set of user controls at the bottom. The right panel presents the completed map-based storyboard for viewers to interact with. We will first describe the interactive controls provided by the storyboard, and then discuss a few advanced viewing modes in the next

section.

The map is overlaid with four graphical elements: keyframe thumbnails representing coherent shots; pins indicating the locations of these shots; a tour path connecting the shots; and a camera location controller shown as the red dot in Figure 5b. The location of keyframe pins are fixed as they are set by the author in the authoring tool, and the size of the pins are proportional to the length of the coherent shots. The width of the tour trajectory between two pins is proportional to the time interval between the two shots. These two elements give the viewer a clear visual information on how many places the tourist visited and how much time he/she spent at each place, and thus are very helpful for the viewer to mentally generate a complete picture of the whole tour without having to navigate through the whole video first.

The camera location controller allows exploration of locations of scenes presented in the video. When the video is playing in the regular “forward in time” mode, the controller will update its position on the tour path to give the viewer a direct visual instruction of where the current scene is, relative to the whole area. Alternatively the viewer can drag the controller and move it forward or backward along the tour path, and the video will be rewound to the proper times to show the scenes captured around the specific locations selected by the user.

The thumbnails of keyframes provide an instant visual summarization of coherent shots created by the author. When the viewer mouses over a thumbnail, a larger version of the keyframe will pop up to show a clearer image, along with some additional comments that the user can apply to the selected shot, as shown in Figure 5c. Since a single frame is sometimes not able to represent a shot well, the viewer can use additional forward and backward buttons to review a few other frames that are sparsely sampled from the shot, or double click on the image to play the shot in the main video

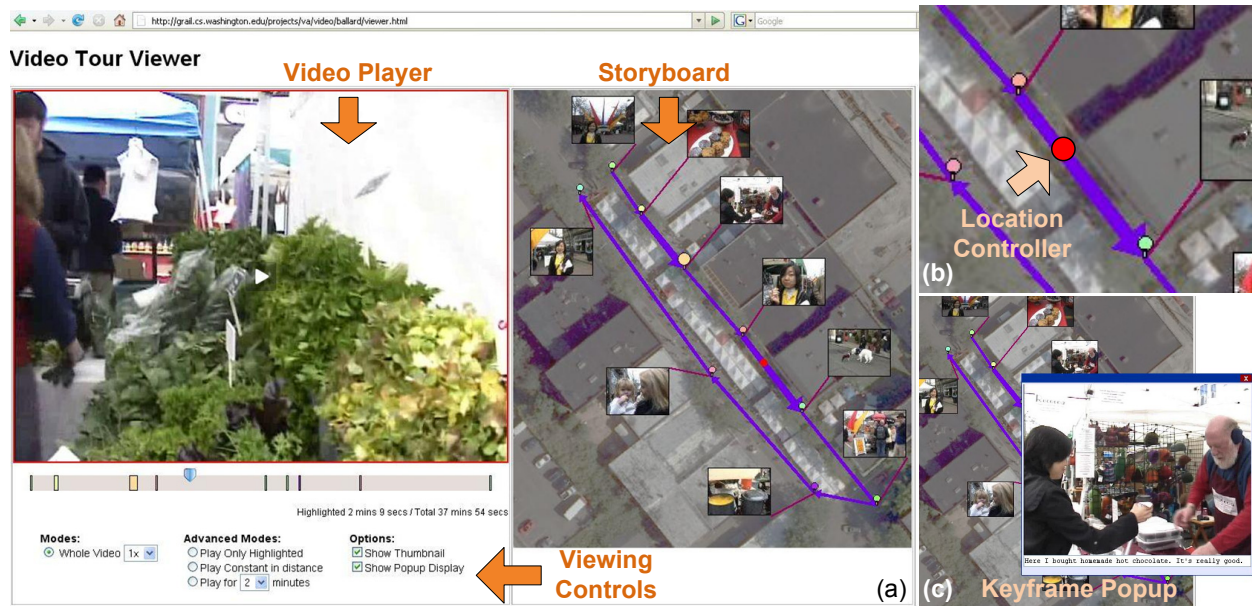


Figure 5: The viewing tool. (a) The layout of the UI. (b) The location controller which identifies the location of the current scene. It either updates automatically when playing the video, or can be controlled by the user for location-based navigation. (c) When moused over a thumbnail on the storyboard, a popup window shows a larger keyframe image along with additional text description.



Figure 6: The viewer created a personal tour path by eliminating some locations. Purple line shows the original path and cyan line shows the path created by the viewer.

player on the left. In this way, the viewer can easily jump to any shot he/she is interested in, at any time.

The viewer can also create a personal virtual tour path from the original path the author provided, by excluding unwanted shots from the map. The viewer can simply click on the “exclude” button on a keyframe to eliminate the corresponding shot from the personal viewing path, and an updated viewing path will be overlaid on the original tour path, as shown in Figure 6. In this way the viewer can easily create a novel

viewing path to visit only a few locations that are interesting to him/her, and thus is able to experience the tour video in a unique way.

ADVANCED VIEWING MODES

We have described two basic viewing modes in our viewing tool: location-based navigation by using the location controller, and coherent shot-based navigation by using keyframes. Leveraging the pre-processing results and the non-destructive video editing infrastructure, we provide three additional novel viewing modes to the viewer to enrich the user experience on browsing the video.

Viewing a High Quality Summarization

This quality-based viewing mode is activated when the viewer chooses to only play highlighted shots. In the pre-processing step, we not only extract coherent shots from the video, but also compute a quality measure for each shot. To play only the highlighted shots, the viewing tool picks out the coherent high quality shots, and only plays these shots while ignoring all other video frames, as shown in Figure 7b. This often results in a short summarization that only contains small or steady camera motions, and thus has a much higher visual quality compared with the original video.

Intelligent Fast Forward

Viewing only high quality shots will cause a large number of video frames to be ignored. This is sometimes undesired if the viewer does not want to skip any parts of the video. In this case the viewer can choose to play the video in the intelligent fast forward mode. In this mode all the highlighted video shots are played at normal speed, while the parts that are between adjacent coherent shots, which we call *transition shots*, are also played, however at faster speeds. The actual speed of a transition shot is determined by (1) the time du-

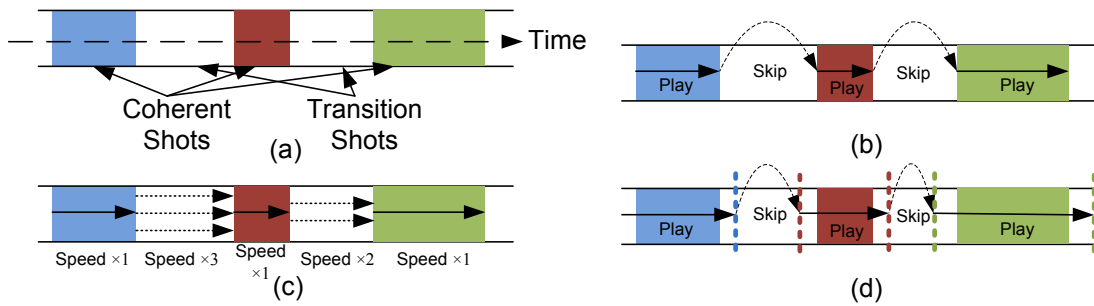


Figure 7: Advanced view modes. (a) The input video is divided into highlighted shots and transition shots. (b) Viewing only high quality shots. (c) Intelligent fast forward. (d) Limited-time abstraction, where the highlighted shots are expanded to meet the user-set time limit.

ration T_d of the shot and (2) the spatial distance S_d the shot covers. Mathematically, the playing speed is set to be proportional to T_d/S_d . Intuitively, this design creates such an effect that the camera moves at a constant fast speed on all the transition trajectories (lines connecting highlighted shots) on the tour path, and automatically slows down to the normal speed in each highlighted shot, as depicted in Figure 7c. By automatically varying the playing speed the viewer can not only focus on highlighted shots, but also get a better summarization of the entire video. This is similar to variable rate control, described by [2], but

Limited-Time Abstraction

Another advanced viewing mode is to create a video abstraction of a limited duration, which can be set by the viewer. Let's assume that the total video length is T , and the sum of the lengths of all the highlighted shots is T_c , and the user-set time limit is T_u . If $T_u < T_c$, the viewing tool will reduce the length of each coherent shot proportionally, and chain them together to create a playing video of length T_c . Similarly, if $T_u > T_c$, the highlighted shots will then be expanded proportionally and chained together to create the final abstraction, as illustrated in Figure 7d. In other words, the viewer can set a time limit which he/she plans to spend on the video, and the viewing tool can automatically generate a good abstraction of the video that meets the exact time limit.

EVALUATION

We evaluated the viewing aspect of the system by conducting an exploratory user study with the following four questions in mind:

1. Do users feel that the map-based storyboard is useful and would people use them in their video navigation?
2. Can users easily understand and learn how to use the different navigation techniques offered in the system?
3. What controls are particularly helpful/distracting to the users?
4. Looking forward, what types of operations should be supported to help home video sharing and navigation?

We interviewed thirteen participants, eleven men and two women. All subjects have actively utilized computers on a daily basis in their professional and personal lives. The study

was performed on a Windows XP desktop machine with 3GB RAM and 3.2GHz processor. The machine was connected to the Internet via a well provisioned campus network. Our web-based interfaces were rendered in a Mozilla Firefox 2.0 browser. The participants had individual sessions and each session lasted 45 minutes on average. Each session started with an introduction where an example video was first played in a conventional media player, and the participant was asked to play around the player for a short time. Then the participant was given a brief tutorial of the viewing interface, particularly, the relationship between the storyboard and the video, and the functionalities of different controls on the UI. The viewing controls that had been explicitly described to the participant are:

1. Playing the whole video at different constant speeds (1x, 2x, 4x)
2. Using the storyboard controls (location controller, pin, thumbnail, or pop-up display) to navigate the video based on spatial locations
3. Playing only high quality shots
4. Playing a video abstraction in a limited time (2-5 minutes)
5. Playing the video in the intelligent fast forward mode

The participant was then asked to familiarize him/herself with the interface until he/she felt comfortable to use the tool. In the next phase, the participant was given two example videos to navigate, and he/she could browse the video on the viewing interface for up to 5 minutes without any instructions or interruptions. The first video was a 38-minute market tour video and the second one was a 10-minute building guided tour. While the participant was watching the videos, we collected the usage of each viewing modes. The viewing modes are categorized into smaller subcategories as follows (to better understand each element of our interface):

1. Playing the whole video at different constant speeds
 - (a) 1x
 - (b) 2x
 - (c) 4x
2. Using timeline slider to jump to a specific moment

3. Using the storyboard controls to navigate
 - (a) location controller
 - (b) pin
 - (c) thumbnail
 - (d) pop-up display
4. Playing only highlighted shots
5. Playing a video abstraction in a limited time
6. Playing the video in the intelligent fast forward mode.

After the navigation phase was finished, the participant was asked to write a summary of the content of the video. At the end of the section, the participant was asked to write down comments about the current interface, his/her general feeling about the usability of the system, and suggestions for future improvements.

Observations and feedback

Participants were able to quickly understand and familiarize themselves with our viewing interface. All of them were able to perform both tasks and use various controls and modes to get to the end of both videos. The feedback was generally positive and one participant said “I would love to have it. I have 8-hour of Japan video that I would love to distribute to my friends in this fashion.”

Interface learning; Most participants familiarized themselves with the interface within a short period of time. The introduction and tutorial session took on average 8.23 minutes. The average time that our participants spent familiarizing themselves with the interface was 5.46 minutes. A few comments from our participants were “The playback interface is almost identical to conventional video player so I felt very comfortable in using,” and “easy to understand and use”.

Observed usage of the interface; There are two important observations from the conductor’s point of view during the tryout session: (1) each participant used different strategies to utilize the 5-minute period of time allocated to each example video; and (2) the strategies were consistent for each participant when navigating the two different example videos. Many started from watching the highlights and then later used different features (e.g. map navigation, slide scrubbing) to go back to specific portions of their interests. One participant started with hovering over all thumbnails to see image/read description of each shot and watched the highlighted shots of the video. Another participant started with watching the fast forward of the whole video then went back to the beginning to watch the highlights. The other participant chose to use the “limit to 5 minute” feature to watch a limited-time abstraction. Figure 8 shows the feature usage statistics from our study. It suggests that the top three features that were used the most are (1) playing highlights; (2) slider scrubbing; and (3) location controller scrubbing.

Preferred modes; From the closing questionnaire, one question asked was “which advanced playing mode do you like most?”, and the participants could choose one or multiple modes, or not choose any at all. Interestingly, instead of a single mode, most answers were a combination of two

modes. The top two modes that were mentioned are “Playing Highlight” (from 10 participants) and “Map Navigation” (from 5 participants). A quote from a participant who liked “Playing Highlight” mode, was that “It saves time to see the whole story.” One participant found the highlights of the videos to be analogous to abstracts of articles. A quote from a participant whose favorite mode was “Map Navigation” was “It provides more straightforward information on how/where to navigate.”

The participants who favored “Map Navigation” are usually those who were very excited when the map-based storyboard was first shown to them. However, “Map Navigation” was not favored by other participants for a variety of reasons, and some of them are directly related to the limitations of our study setup. One quote from the feedback was “I think the map gives you a nice overview at the first glance. However, due to the nature of the survey, I just concentrated on the video.”. Another one was “If I had more time, I think I would use the location controller to try to go back to a spot.” These comments suggest that the map-based storyboard can be more useful in other settings, especially when users have more time to navigate the video.

Some of our participants found the less complicated storyboard (market tour) to be useful, while the more complicated storyboard (building tour) to be a little confusing. This suggests that even though our approach has the benefit that all the stories could be seen immediately in a single image, in the more complicated paths, visualizations that show only a small part of the storyboard or an actual 3D-style layout with some animation might be easier to follow and focus their attention.

Participants’ content summaries; The brief video summaries written by participants varied from just a few sentences to a rather detailed description of the whole tour. Although it is quite difficult to quantify the understanding of the videos from the summaries, since the summaries probably reflect the participants’ writing styles and memories, we found that participants generally covered the most important scenes/shots that the authors wanted to highlight when creating the storyboards. In the market tour video there were four key elements: market, food, dogs and babies, and all summaries mentioned market and food, 11 out of 13 summaries mentioned dogs, and 6 out of 13 mentioned babies. The summaries of the building tour were more elaborate, and except for two, all the other summaries followed closely the tour path of the video, resulting in really long summaries.

Suggestions Our participants were eager to give suggestions about what they would like to see on our interface. Some suggested different and customizable placement of the maps with respect to the video player. Some suggested different colors used on the layout and better graphics (e.g. vector graphics).

Feedback The overall feedback about the interface was very positive. In response to the question “what is your general feeling about the interface?”, many mentioned that they were impressed by the variety of interactive modes the interface provides, and also the integration between the video and the

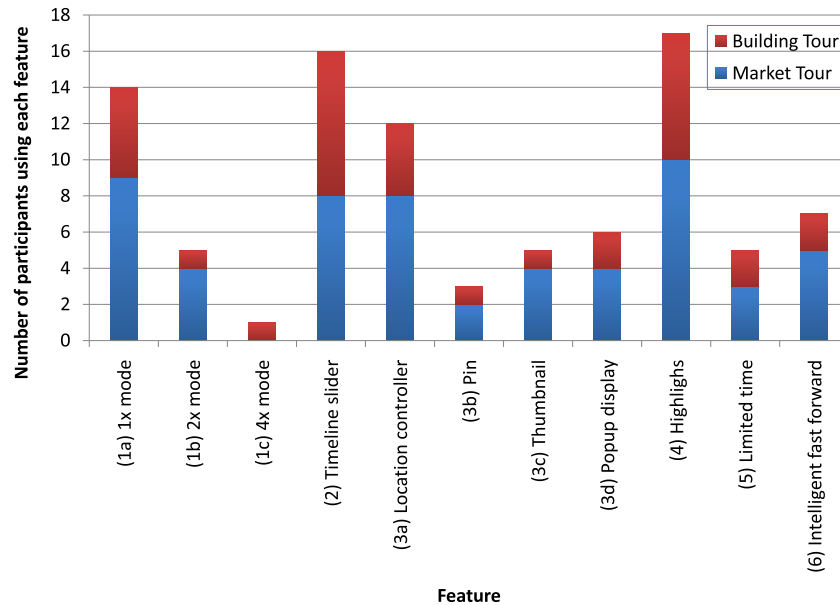


Figure 8: The feature usage statistics collected from our user study. A feature is marked as used by a participant if the participant used it at least once in the 5-minute video browsing period. The y-axis is the number of participants that used the feature.

map, as one comment said “[the interface was] good, interactive, and helped me see where the video is in the time line”.

DISCUSSION

Even though the preprocessing is a part of our system, its purpose is to assist users in picking keyframes. A user can bypass the process and the rest of the tools will still work fine. However, the user will need to spend more time on picking interesting keyframes, and all frames will be considered to have equal quality.

Few participants used the thumbnails on the storyboard to navigate the video. We informally talked with a few participants after the study and asked why they did not use thumbnails for navigation. The general feedback was that the users would rather directly watch different parts of the video instead of looking at the thumbnails first. We believe this is more related to our experimental setup, since the experimental machine is connected to a high speed campus network thus video streaming is nearly instant. Since the participants had access to the whole video from the very beginning, the additional information presented by thumbnails was not appreciated. However in other settings such a video will take a much longer time to be streamed over the Internet, and the thumbnails will be more valuable to the user since they provide an instant summarization of the whole sequence, and allow the user to choose which part of the video should be streamed first.

In the user study, we did not conduct any experiments on the authoring tool, since the primary goal of the study is to explore the value of the map-based storyboard. Conducting a formal user study on the authoring tool requires the users to provide tour videos they captured themselves so they know the geographic locations of the shots in the video, which is

too much of a burden for the participants. We plan to post our system online so anyone who is interested in the system can use the authoring tool to create his/her own storyboards, and an online user study can be conducted later when enough examples have been created.

CONCLUSIONS AND FUTURE WORK

We present a map-based storyboard as a novel user interface for browsing a long tour video. The storyboard allows users to navigate a video based on both time and location information of different shots, yielding a set of novel controls. An authoring tool is also presented to make the process of creating such a storyboard easy. The web-based viewing tool allows effective video sharing and our informal user study indicates that the viewing tool gives the users advantages over regular media players.

A number of components of the current system can be further improved to enhance the user experience. In the preprocessing step, more features, such as audio features can be extracted for a better analysis of the video content. This will enable the system to capture interesting shots based on sound, or avoid a shot cut in the middle of a person’s speech. On the authoring tool, the extracted keyframes can be compared with other photographs online which have geo-location information attached, so that the keyframes can be automatically placed at the proper locations on the map.

On the viewing tool, the static thumbnails can be replaced with short dynamic thumbnails for a better summarization. Different tour videos captured by different people at the same place could be coupled together on the same storyboard so the viewers can experience a more complete virtual tour of the area. Viewer comments, user-added photographs and videos will also enable fun networking opportunities. There

are many opportunities to make the system better, more intelligent, and more fun to use.

Although in this paper we focus on a single type of video, our system is definitely not constrained to only work in one case. As the most important future work we plan to expand the system to deal with other types of video, by replacing map-based storyboard with agenda-based storyboard, or other types of visualization tools, with the hope that our system can be used as a universal interface for online video browsing. As mentioned in the discussion, we plan to post our system online later so that anyone who is interested can use our system to easily process their unedited videos and share the results with others.

ACKNOWLEDGMENTS

The viewing tool uses the open source work from Jeroen Wijering⁷ and Walter Zorn⁸. The automatic preprocessing used the code by Pravin Bhat. We would like to thank our study participants for spending time with our system and providing useful feedback, Hank Levy for giving us the Paul Allen center tour at the University of Washington, Pahnit Seriburi for starring in our market tour, which are used as examples in our study, and Laura Effinger-Dean for the help with our demo video. Funding and research facilities were provided by the University of Washington GRAIL lab and Adobe Systems Incorporated.

REFERENCES

1. M. Cooper and J. Foote. Scene boundary detection via video self-similarity analysis. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 3:378–381 vol.3, 2001.
2. J. Foote, J. Boreczky, A. Girgensohn, and L. Wilcox. An intelligent media browser using automatic multimodal analysis. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 375–380, New York, NY, USA, 1998. ACM.
3. A. Girgensohn, J. Boreczky, P. Chiu, J. Doherty, J. Foote, G. Golovchinsky, S. Uchihashi, and L. Wilcox. A semi-automatic approach to home video editing. In *Proceedings of UIST '00*, pages 81–89, 2000.
4. D. R. Goldman. *A Framework for Video Annotation, Visualization, and Interaction*. PhD thesis, University of Washington, 2007. Section 2.4.
5. R. Kadobayashi and K. Tanaka. 3d viewpoint-based photo search and information browsing. In *Proceedings of SIGIR '05*, pages 621–622, 2005.
6. A. Komlodi and G. Marchionini. Key frame preview techniques for video browsing. In *Proc. of DL '98*, pages 118–125, New York, NY, USA, 1998. ACM.
7. Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical Report HPL-2001-191, HP Laboratory, July 2001.
8. R. Lienhart. Abstracting home video automatically. In *Proc. of MULTIMEDIA '99*, pages 37–40, New York, NY, USA, 1999. ACM.
9. A. Lippman. Movie-maps: An application of the optical videodisc to computer graphics. *SIGGRAPH Comput. Graph.*, 14(3):32–42, 1980.
10. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
11. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of the British Machine Vision Conference*, pages 384–393, 2002.
12. K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
13. M. Naaman, Y. J. Song, A. Paepcke, and H. Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *Proceedings of JCDL '04*, pages 53–62, 2004.
14. J. Nam and A. H. Tewfik. Video abstract of video. In *Proc. of IEEE 3rd Workshop on Multimedia Signal Processing*, pages 117–122, 1999.
15. S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting digital movies automatically. Technical Report TR-96-005, 1, 1996.
16. M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *CAIVD*, pages 61–70, 1998.
17. N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc. of ACM SIGGRAPH 2006*, pages 835–846, 2006.
18. Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. Videomap and videospaceicon: tools for anatomizing video content. In *Proceedings of CHI '93*, pages 131–136, 1993.
19. K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In *Proceedings of MULTIMEDIA '03*, pages 156–166, 2003.
20. B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):3, 2007.

⁷<http://www.jeroenwijering.com>

⁸<http://www.walterzorn.com>