Videotater: An Approach for Pen-Based Digital Video Segmentation and Tagging

Nicholas Diakopoulos and Irfan Essa
College of Computing (IIC), GVU Center, Georgia Institute of Technology
85 5th Street NW., Atlanta, GA 30332-0760, USA
{nad, irfan}@cc.gatech.edu

ABSTRACT

The continuous growth of media databases necessitates development of novel visualization and interaction techniques to support management of these collections. We present *Videotater*, an experimental tool for a Tablet PC that supports the efficient and intuitive navigation, selection, segmentation, and tagging of video. Our veridical representation immediately signals to the user where appropriate segment boundaries should be placed and allows for rapid review and refinement of manually or automatically generated segments. Finally, we explore a distribution of modalities in the interface by using multiple timeline representations, pressure sensing, and a tag painting/erasing metaphor with the pen.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. – Interaction Styles.

General terms: Design, Human Factors

Keywords: Video Segmentation, Video Tagging

INTRODUCTION

Growing video repositories represent assets that can potentially be mined to provide media for new or different productions. A good example would be to pick a favorite character from a television show and produce a character sketch or montage using material from past episodes. We refer to this process as *video repurposing* and it requires the source material be (1) searchable (using tags or annotations) and (2) represented at an appropriate granularity for repurposing. In this paper we present Videotater, a system designed to facilitate the preparation of video assets for later repurposing. Videotater supports the user in rapid and accurate segmentation and segment trimming at the shot level as well as an intuitive and easy to learn method for tagging segments.

Remixing or modding of existing media, while of benefit to professionals who work on videos, can also support beginners in their efforts, provided efficient interfaces for segmentation and tagging of video databases are available. Remixing video is presently a thriving online activity in which people mash-up media assets and recombine them. One example is the community of people who make anime music videos [16] by taking snippets of many different anime videos and setting them to music. Such remixing behavior can certainly also benefit from the tagged searchable segments of video that Videotater produces.

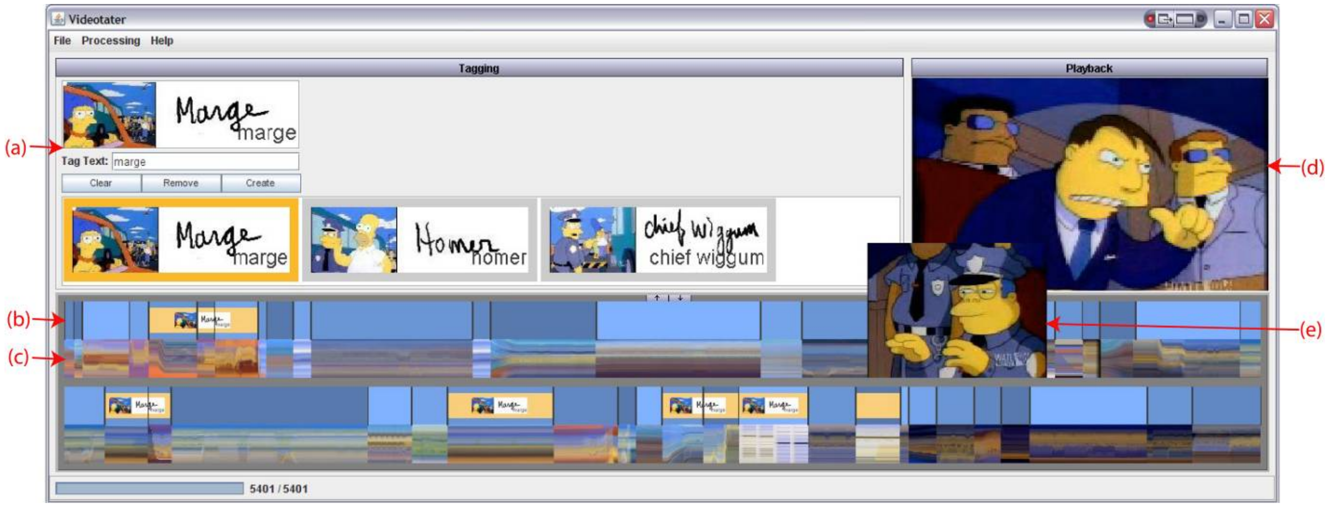
Existing commercial or research video logging tools are inadequate for the tasks of rapid manual segmentation and tagging. Previous research [4, 8] has noted that segmenting and trimming shots is tedious and not well supported in existing interfaces. Videotater introduces visualizations and interactions which support successful navigation, selection, and tagging of video. Our interface incorporates a number of advances and explorations that we see as novel contributions:

- Helping the user to precisely determine and execute *where* to segment. By providing a visual scent on the timeline the user can quickly see where there is a potential segment boundary. We also introduce a semi-automatic component similar to a magnetic lasso in which a user gesture snaps to the nearest likely segment boundary based on color histogram differences.
- Providing explicit support for *refining* or *trimming* rough segmentations that may come from automatic procedures. Though automatic segmentation algorithms have become fairly robust (~80% for fade detection) [11], designing interfaces that better integrate human decision making in automatic processes ensures high precision and meaningful output. Our polyfocal visualization supports the task of refining segmentation using an overview + detail + detail context for fine-tuning the in and out points of a segment.
- An exploration of different modalities of interaction on a pen-based computer. We distribute modalities across space, time, and pressure and provide mode switching in a task specific way. This is meant as an implementation in a more real scenario of some of the mode switching strategies described in [5, 10]. We report user reactions to our choices of modal distribution.

BACKGROUND AND RELATED WORK

During the design process we interviewed three individuals with professional video editing experience to find out some of the shortcomings of existing video editing and logging software. An issue which repeatedly came up was that it

UIST'06, October 15–18, 2006, Montreux, Switzerland.
Copyright 2006 ACM 1-59593-313-1/06/0010...$5.00.



Tag Text: marge

5401 / 5401

Figure 1. GUI showing (a) tag creation, (b) timeline segments, (c) timeline stripe image, (d) playback window, (e) popup frame from timeline stripe image hover. The timeline representations wrap onto the next line to avoid scrolling.

was very tedious to set the in and out points of a video segment. Other important points which also influenced our application were that the dynamics of video clips can be important when annotating them and that the most generally useful granularity of segmentation is the shot level.

There is a wealth of related literature on video annotation, visualization, and interaction techniques that influences our work. One of the earliest systems in the literature to approach video annotation was Mackay's EVA [12]. EVA was a video analysis application which allowed for tagging of events *in real-time* and was developed to allow for behavioral video analysis. Though Videotater could easily be adapted for such a scenario, its visualizations and interactions were designed for video that has already been recorded and is being reviewed by someone interested in repurposing it.

Marquee [18] was another early system developed for video annotation and tagging. Design iterations showed that users also needed a mechanism for applying keywords without the pressure of real-time. Marquee used the notion of segmenting the video into timezones to which keyword "paint" could be applied. Our system is similar to Marquee in that we also allow for segmentation of shots and application of tag paint, however, our visualizations and interaction techniques allow for much more *precise* segmentation and tagging of segments and frames.

We also draw upon recent work on controlled-vocabulary keyframe tagging by Volkmer [17] which indicated that multi-pass tagging with only one concept at a time made the tagging task more efficient. Our work incorporates these results and builds upon them by allowing for review of dynamic aspects of the video rather than assuming that concepts can be tagged using a static keyframe.

The Family Video Archive (FVA) [1] explored the symbiosis between automated and manual techniques for tagging *collections* of videos. Our work also explores automatic/manual symbiosis but is oriented more toward tagging individual videos at a higher granularity. In particular we are interested in providing shot-level tags with the option to include per frame tags, rather than the default in FVA which was to apply a tag to an entire video.

The LEAN system presented by Ramos in [14] is another effort toward novel visualization and interaction techniques for video annotation. That work strove to make video annotation more like annotation on paper with freeform handwriting, which differs considerably from our focus on allowing the user to rapidly add tags from a palette.

An interaction technique that we rely heavily on is that of painting tags onto the timeline. A verb-noun interaction naturally makes sense here because of the large similarity between most adjacent frames in a video. Also, since video represents a special case of a segmented continuous variable it lends itself to a painting interaction in which tags are applied by swathing out areas on the timeline [3].

Finally, we developed a new visualization, the polyfocal visualization, which bears some resemblance to a tracking menu [7]. This dynamic visualization tracks the location of the mouse cursor on the video timeline allowing for a localized visualization and interface for segment trimming. This facilitates low interaction costs in reviewing and refining segment boundaries.

VISUALIZATION

Timeline: We chose a timeline visualization as it is a familiar metaphor for time-based media work. We decided to wrap our timeline in order to avoid the interaction costs of scrolling or panning and to maximize screen space [9]. Our timeline is notable in that it displays two different views of the underlying video (See Figure 1 (b) and (c)). The segment view indicates where segments have been delineated in the timeline. Adjacent segments are demarcated with a vertical line and differing color brightness in order to enhance contrast. The lower half of the timeline shows a stripe image in which each column of pixels represents the row average of a frame in the video. The visual scent of the underlying pixel colors and their evolution on the timeline aids the user in seeing where potential new segments should be inserted. Though a different visualization this is at least similar in spirit to the Video Streamer [6]. The underlying frame pops up when hovering over the stripe image so the user can quickly scan through the video timeline. On top of the segment view a bright orange bar shows the extent of a tag applied to the video. Within this an icon of the tag reminds the user which tag is being shown.

*Polyfocal Visualization***:** The polyfocal visualization (see Figure 2) is meant to aid in the tasks of segmentation and segmentation refinement. The concept of the polyfocal visualization is similar to a bifocal image browser [13], in which a detailed view pops up when something in the overview (timeline) is selected. We took special care in trying to minimize the worldview gap [2], the gap between what is shown and what needs to be shown in order to make the decision about a correct segmentation. This is minimized by showing all relevant information such as what the current in and out frames for the segment are as well as contextual thumbnails showing frames in the vicinity of the in and out frames (± 5 frames). The novelty of the visualization stems from its ability to show both detail of 2 focal points (in and out frames) and detail context (10 frames around each in and out point) while also giving the user a frame of reference in the overview.

Scalability: These visualizations were designed for video lengths typical of American television shows (roughly 22 minutes per episode). Though we do not preclude using longer videos, there are some scalability issues related to screen space and resolution. In particular it may become difficult to accurately select frames or segments on the timeline when they become too small in the visualization. As an alternative we could also allow for the timeline to scroll though we avoided this due to the increased interaction cost.

INTERACTION

There are two primary tasks around which the interactions in Videotater are based; segmentation and tagging. Both of these tasks are accomplished by interacting with the timeline visualization or the polyfocal visualization.

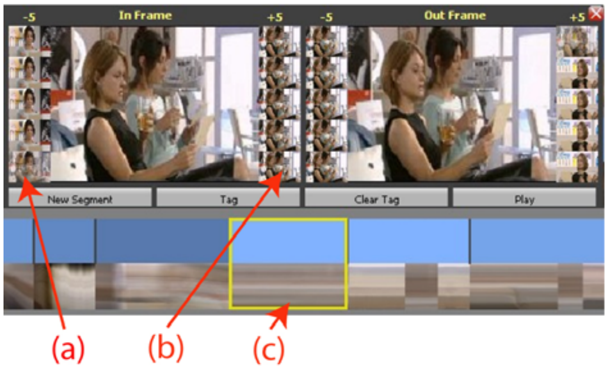


Figure 2. Polyfocal Visualization. (a) 5 frames before in point (b) 5 frames after in point (c) selected segment on timeline

The design strategy for the interaction in Videotater was to distribute modalities across space, time, and pressure and provide for efficient mode switching in a task specific way. The timeline visualization itself defines spatial modalities between the segment and stripe image views. Pressure is used to distinguish between selection and tagging modes during drawing on the timeline. Low pressure indicates selection and high pressure tagging. Using just two levels of pressure allows for ballistic switching to the tagging mode and reduces difficulties and errors in precise pressure control [5, 7, 15]. Finally, we use tangible mode switching when applying tags; the pen tip applies the tag and the pen eraser removes it.

Gesturing on the timeline supports the task of segmentation and allows for very rapid splitting and merging of segments. Drawing a straight vertical line splits the current segment along the mean x-value of that line or snaps to the nearest likely boundary within a given window width according to inter-frame color histogram intersection. This interaction is most akin to the magnetic lasso in Photoshop in which the lasso snaps to the nearest boundary according to image gradients. A merge gesture is affected by drawing a line between segments on the timeline, including between segments in potentially different rows. All segments between the first and last segments intersected by the gesture are merged.

A painting interaction metaphor is used to accomplish the tagging task, whereby tag paint is selected from the tagging view (see Figure 1(a)) and drawn over the timeline wherever it should be applied. When drawn on a segment the tag is applied to the entire segment whereas drawing on the stripe image applies the tag only to the frames touched. Thus both shot (segment) level tagging and finer frame-based tagging are supported. Removing tags from segments or frames is accomplished by drawing over the timeline user the eraser end of the stylus.

Segmentation, segment tweaking, and tagging can also be invoked using the polyfocal visualization (see Figure 2). During mouse or pen hovering the polyfocal visualization is modeless. When a concrete selection is made through clicking or dragging it becomes modal until dismissed or the selection unlocked by clicking again. During mouse hovering, the polyfocal visualization tracks the segment that the mouse is over, similar to the tracking menu in [7]. This facilitates rapid review of segment boundaries. Hovering is temporarily disengaged when the user moves the cursor over the polyfocal visualization to interact with the menu and re-engaged when the cursor again leaves the polyfocal window. Within the polyfocal visualization, hovering over a context image shows it at a larger size and clicking on a context image makes it the new in or out point of the segment. There are also buttons for creating a new segment, tagging/un-tagging a segment, and playing a segment.

Interaction with the tagging view (see Figure 1(a)) is quite simple. New tags can be drawn, key frames selected from the timeline to go with the tag, and a textual description entered. Clicking a tag highlights sections of video containing that tag on the timeline and makes it the active tag, which can then be painted onto the timeline.

IMPLEMENTATION

Videotater was implemented in Java Swing using the latest beta release (mustang, V1.6) in order to take advantage of improved hardware acceleration. Video preprocessing, which amounts to roughly 20% of video real-time on a 2.1GHz Pentium M processor, was also necessary in order to maintain interactive frame rates during program use. It consists of capturing thumbnail images of each frame of a video in order to avoid slow video decompression during run-time. These frames were JPEG compressed to reduce the memory footprint.

USER FEEDBACK

Due to the lack of other systems with feature parity for comparison, our preliminary evaluation involved asking 3 experienced video editors to use our interface to segment and tag a two minute length of video while thinking aloud. Users were given a demonstration and allowed to explore and become comfortable with the interface before the think aloud began.

All users felt that the timeline visualization was compelling and that the hovering popup frames and colors in the stripe image made the task of scanning the video to determine *where* to segment and tag quite fast and easy. Though intuitive, they did however think the split gesture *felt* a bit sloppy, despite snapping to the nearest boundary, and didn't necessarily like the idea of having to go back and trim segments later. The segment and stripe image visualizations on the timeline seemed to accommodate the task of segmentation well. Informal timings of segmentation efficiency indicates it takes roughly 2x real-time while maintaining high precision.

Users also felt that using high pressure to apply tags was very easy to get used to and was quicker than hitting the button on the polyfocal visualization once they learned how pressure mapping behaved. Changing to the tag erasing mode by using the eraser end of the stylus was natural and easy to learn, but wasn't always preferred to hitting a button due to the time involved in flipping the stylus.

We did receive some negative user feedback on the visual design of the polyfocal visualization. It was difficult to parse all the visual information at once and the vertical layout of contextual thumbnails seemed unnatural to the users. The tracking nature of the visualization also made some users feel it was in the way and that it belonged in a stationary window. Of course this represents a tradeoff in the amount of time necessary to move the cursor and interact with the visualization. Overall, users found the polyfocal visualization useful for trimming segments, but the visual design may need to be improved and retested.

CONCLUSIONS

We have presented an application comprised of some novel visualizations and interactions which facilitate the preparation of video assets for later video repurposing. User feedback indicates that our timeline visualization and gestural interactions were successful and helpful for finding and executing where to segment a video. The mapping of low pressure to selection and high pressure to tagging was intuitive and effective for rapid tagging. The polyfocal visualization provided an overview + detail + detail context view of a segment and was useful for trimming segment boundaries, though it could benefit from more intuitive visual design. In future work we would like to refine the polyfocal visualization as well as incorporate an audio track into the interface.

REFERENCES

1. Abowd, G.D., Gauger, M. and Lachenmann, A. The Family Video Archive: an annotation and browsing environment for home movies *5th ACM SIGMM workshop on Multimedia information retrieval*, 2003.
2. Amar, R. and Stasko, J. Knowledge Precepts for Design and Evaluation of Information Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, *11* (4). 432-442.
3. Baudisch, P. Using a Painting Metaphor to Rate Large Numbers of Objects *Proceedings of HCI International*, 1999.
4. Casares, J., Long, A.C., *et al*. Simplifying video editing using metadata *Proceedings of DIS*, 2002.
5. Deming, K. and Lank, E., Managing Ambiguous Intention in Mode Inferencing. in *AAAI Fall Symposium Series: Making Pen-based Interaction Intelligent and Natural*, (2004), 49-54.
6. Elliot, E. and Davenport, G., Video Streamer. in *CHI extended abstracts*, 65-66. 1994.
7. Fitzmaurice, G., Khan, A., Piek, R., Buxton, B. and Kurtenbach, G. Tracking menus *Proceedings of UIST*, 2003.
8. Girgensohn, A., Boreczky, J. *et al*. A semi-automatic approach to home video editing *Proceedings of UIST*, 2000.
9. Huynh, D.F., Drucker, S.M., Baudisch, P. and Wong, C. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections *CHI extended abstracts*, 2005.
10. Li, Y., Hinckley, K., Guan, Z. and Landay, J.A. Experimental analysis of mode switching techniques in pen-based user interfaces *Proceedings of CHI*, 2005.
11. Lienhart, R. Reliable Transition Detection In Videos: A Survey and Practitioner's Guide. *International Journal of Image and Graphics (IJIG)*, *1* (3). 469-486.
12. Mackay, W.E. EVA: an experimental video annotator for symbolic analysis of video data. *SIGCHI Bull.*, *21* (2). 68-71.
13. Plaisant, C., Carr, D. and Shneiderman, B. Image-Browser Taxonomy and Guidlines for Designers. *IEEE Software*, *12* (2). 21-32.
14. Ramos, G. and Balakrishnan, R. Fluid interaction techniques for the control and annotation of digital video *Proceedings of UIST*, 2003.
15. Ramos, G., Boulos, M. and Balakrishnan, R. Pressure widgets *Proceedings of CHI*, 2004.
16. Shaw, R. and Davis, M. Toward emergent representations for video *Proceedings of ACM Multimedia*, 2005.
17. Volkmer, T., Smith, J.R. and Natsev, A. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study *Proceedings of ACM Multimedia*, 2005.
18. Weber, K. and Poon, A. Marquee: a tool for real-time video logging *Proceedings of CHI*, 1994.