SLIME: A Tool for Composing Live and Stored Media

Liu Bing*and Ooi Wei Tsang†
Department of Computer Science,
School of Computing,
National University of Singapore

ABSTRACT

One of the three grand challenges in multimedia research highlighted recently is the ability to make authoring complex multimedia content as easy as word processing. To take up this challenge, we designed an easy-to-use video editing tool named SLIME that allow user to compose live and stored video together. It has powerful features on spatial editing and real-time editing. It makes use of Plasma language as its processing engine, and provides a user-friendly GUI by combining many existing interface designs on video editing tools. Those designs and the implementation are presented in this paper.

1. INTRODUCTION

One of the three grand challenges in multimedia research highlighted recently is the ability to make authoring complex multimedia content as easy as word processing (Rowe & Jain, 2005). Due to the development of digital video equipments such as video camera and web camera, digital video is becoming increasingly ubiquitous. As a result, easy-to-use video editing tools are needed by more and more average users. Although some excellent commercial tools exist, such as iMovie and Premiere Pro, video editing is still a difficult, tedious and error-prone activity. Video composition is widely used in multimedia productions such as television programs, academic video, video conferencing, and surveillance. However, we find that most of the tools focus on temporal editing tasks such as the cutting and pasting of video segments. The spatial editing tasks such as video composing and overlay can only be done by complex software which target experts, such as Final Cut Pro. Since there is trade-off between expensiveness and ease of use, those expert-user tools require too much learning and high system performance. Another limitation of traditional video editing tools is that most of them only can process stored media. With the development of IP Multicast, RTP, Internet Mbone Tools (McCanne, 1999), the network media applications used for distant learning, remote collaboration, video on demand system, and interactive game became more and more popular. As a result, there is a need to real-time adapt and manipulate live streaming media. For example, during producing a live webcast for a lecture, several video and audio stream form different cameras and microphones are sent to a central studio. Then the director needs to real-time select and edit those media and stream to the distant audience. In television industry, to perform the above task, it may require expensive equipments and many people to operate the system.

The above challenges motivated us to design and implement **S**tored and **LI**ve **M**edia **E**ditor (SLIME), a prototype of an easy-to-use video editing tool that can real-time compose both live and stored media. One of the contribution of SLIME is that it provides an intuitive graphic user interface that exploits the powerful features in Plasma scripting language (Zhu, 2003), currently actively under development in SoC. The novel user interface of SLIME can reduce the difficulty during video editing and webcast production. It allows user to compose different videos together by freely moving, resizing, cropping, fading and titling video clips. With SLIME even untrained average usercan easily produce high quality videos which only can be produced by professional video editors. Furthermore, SLIME can receive, process, and broadcast heterogeneous streaming media. The efficient way for real-time editing allows producing webcast with common PC and only requires one director to operate. User also can use SLIME for video conferring, video surveillance and even as a multi-channel network TV. Moreover, we propose a novel design of the video timeline which allow user to easily retrieve the information such as shot transitions.

*Student
†Supervisor

In this report, we will review the previous works in next section. Then we presents the features of SLIME in section 3. Section 4 discusses the design and implementation issues. After address the results and limitation in section 5. We discuss future work and conclude.

2. RELATED WORK

2.1. Stored Video Editing

Most modern digital video editing systems are adaptations of classical film editing systems. In general, the interface of commercial video editor consist three components which are browsing panel, preview panel and timeline. Browsing panel is used to browse video clips. Some systems present the clips in a hierarchical structure such as VIVO (Yang, Mu, & Marchionini, 2003). Hitchcock (Girgensohn, Boreczky, Chiu, Doherty, Foote, Golovchinsky, Uchihashi, & Wilcox, 2000) even can cluster all clips by the similarity of their colors. Preview panel is used to preview source video or resultant video. Most of the software support play/pause, stop functions. Premiere also support non-linear forward/backward playback. MyVideo (Wang, Zhao, Zhang, Li, & Zhang, 2002), a home video management system, use a technique called playback highlight. It also called video summary means play a small portion of representative content form a long video sequences. On timeline, user can cut/paste video clips, and add transition effect or titles. Silver (Casares, Long, & et al., 2002) that support a hierarchical timeline which allows user to work at a high level of detail without losing the context. SLIME adapts some of above designs and also propose our CHI designs that bring user more convenience. The details will be described in next section.

2.2. Live video editing

BIBS (Rowe, Harley, & Pletcher, 2001) is one of the research of Berkeley Multimedia Research Center. They have proposed Director's Console (Yu, Wu, Meyer-Patel, & Rowe, 2001) which is a live webcast production application based on a general webcast production model composed of three stages. They output two video streams and one audio stream. Another webcast system called Virtual Director (Machnicki & Rowe, 2002) can automate the tasks such as recording equipment control, stream broadcasting, and content decisions. Ooi et al (Ooi, Pletcher, & Rowe, 2001) has developed a middleware for distributed media environment named Indiva. It use a file system metaphor to access resources and provide a unified set of abstractions and high-level operations for hardware devices (e.g. microphone, camera, speaker, etc.), software processes, and media data in a distributed audio and video environment. SLIME adapt dc's webcast production model and provide more spatial effects. Invida can be integrated into SLIME to allow the user to operate devices. This will be described in the future work section.

3. SLIME APPLICAITON

To solve the problems of video editing mentioned in section 1, we developed SLIME, a video editor implemented with Tcl/Tk (Ousterhout, 1994). Currently, SLIME use Plasma beta version 0.2 as processing engine, use Dalí software library (Ooi & Smith, 1999) to retrieve video information and manipulate images, use MPlayer to achieve highlighted playback, and use FFmpeg multimedia system to convert resultant video files from to various popular formats. It can process not only the stored media but also the live RTP/RTSP streaming media on the air. The real-time or static-time manipulations it provides include resizing, cropping, fading, composing (side-by-side or overlay), swapping, etc. These powerful features allow user to easily produce stored videos or live webcast with professional effects.

Browsing panel The Browsing panel serves as a starting point for video editing. It allows user to import, preview or delete stored and live media streams by clicking corresponding button. User can use *Button 2* to import video clips and images with many popular formats (e.g. MPEG, WMV, RM, MOV, JPEG, GIF, PNG). Importing video files one by one is a tedious work. So SLIME supports the feature that allow import multiple files once. Alternatively, user can start to receive a RTP/RTSP media stream by pressing *Button 1* and type in the URL address in the live stream selection dialog. After importing, all the media will be represented by a thumbnail with other information such as duration and file name. For stored video, the thumbnails are generated by extracting the first frame of a clip. For live stream, the thumbnails are represented with a temp image. After the connection was setup, the thumbnails can dynamically update every second. It describes the current frames so that user can know what is going on.

Figure 1. A overview of SLIME user interface

Title panel and Music panel The Title panel and music panel is source panels as the Browsing panel. Text effects (e.g. still, fade in, pup in, and fly in) are represented with animated GIF images. User can specify the duration. To avoid exceptions, those entries has programmed with validity checking functions. After drop a title onto the Editing panel, user can type in the text with in a text box and can move the text box to any position. Moreover, user can choose the font and size of text. The Music panel allows user to select and pre-listen audios as background music. Audios are also represented as thumbnails and user can drag them onto the background music pool.

Editing panel The Editing panel provides a place for editing. User can easily drag and drop source media thumbnails from the Browsing panel. SLIME has WYSIWYG[3] feature so that user can freely move and resize a media and the resultant video will be as the same as the user specified. SLIME adopts the standard GUI design widely used in commercial software (e.g. Microsoft PowerPoint, Adobe Photoshop, etc.) To resize, user can drag and drop the handlers. During resizing, user can keep the width-height ratio by holding *SHIFT* or *CTRL* key. Holding *SHIFT* key also can help user keeping move a clip on a strait line. The *Zoom in/out button* and the pull-down menu can scale of the users' view. This facility allow user to edit at a high level of detail without losing the context. Using *Undo/Redo button*, user can relax and edit as they like without worrying making mistakes. The *Preview button* here can open a Plasma window to show the resultant video. The *Saving button* can open the Make Video Window. Finally, the Streaming button will be used to broadcasting the resultant video out as a RTSP stream. The remote user can receive and play this stream by any software with RTSP support such as FFplay, MPlayer, VLC and so on. During streaming, the Streaming button will be replaced by the *Update button*. Whenever user make changes on the editing panel and press the Update button, those changes will reflect on the output steam at once. With this real-time editing feature, a webcast director can add videos, titles or logos onto the screen, and switch two streams by dropping one's thumbnail onto others.

[3]what you see is what you get

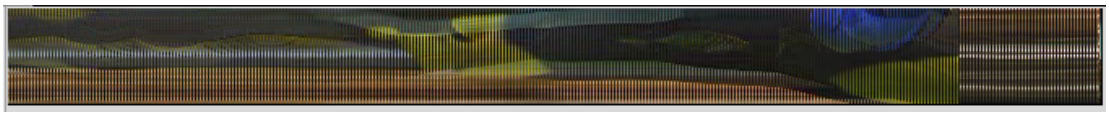

Figure 2. Our timeline bar

Crop panel and Layer panel The Crop panel can help user to crop a video clip or image selected in the Editing panel. Instead of specify the exact coordinates, user can simply draw a crop box by mouse in the Crop panel. Then corresponding part outside that box will be trim out. We achieve this feature using a Dali function and the details will be presented in next chapter. The Layer panel is used to organize the spatial relations of clips on z-axis direction. User can change the order of layer, and adjust the transparency and sound volume. The Eye icon inside each layer is used to hide or show this layer.

Property panel and History panel The Property panel can show the detail information of each selected clip, and it also allow use to specify the size, position, color and transition effects (e.g. fade in, slide in, curtain in, fly in, etc.). User can change these parameters to make more precise editing. Furthermore, user can specify a background color of the whole project. Our system also allow user to select one appearance effect for each media object. The History panel records all actions user performed so that user can recover to any pervious point as they like. The history panel allow user to recover many actions once. Clicking this box will recover the system to that state.

Preview window and Make movie window Currently we use MPlayer for our Preview window. User can press *P key* to pause or play the video and press *Q key* to exit. For random access, user can press arrow keys. *Right arrow key* and *Left arrow key* can forward or backward video for 10 seconds, and *Up arrow key* and *Down arrow key* can forward or backward video for 10 minutes. This feature can be consider as a kind of manually highlight playback (Wang et al., 2002). During playback, user can play use arrow key to dynamically adjust the highlight portion. This video summary technique is very useful for user to quickly go through the video. Finally user may want to save the result video to a file. Make movie dialog allow user to select the location and the format (e.g. MEPG, DivX, WMV, RM, MJPEG, H.263+, etc.). Then the result file will be produced after clicking start button.

Timeline Timeline is very useful for stored video temporal editing. Our timeline has two levels. At high level, the video sequence is a series Scenes. Each Scene is a composition of many video clips. In the low level view, each video clips in a Scene has its own timeline. However, in stead of using a bar to represent timeline as traditional approach, our timeline bar is formed of thousands of colored slice. Each slice represents a video frame at particular time point. We extract the dominant colors from a frame and form a slice. Then from the whole bar, user can get the general information such as shot transition (See Figure 2).

4. DESIGN AND IMPLEMENTATION

Currently SLIME is implemented as a Tcl/Tk 8.4.7 application with approximately 8000 lines of code. TkZinc extension is used to achieve the features such as transparency and editable text. Exacting frame, information from video clip and images manipulation rely on Dalí library. Our processing engine is Plasma beta 0.2. Since it support limited encoding codecs, we transcode the resultant video using Mencoder.

4.1. Importing

Different types of source require different import method. For stored video, we use Dalí code to parse video header to get information about width height and framerate. Then we count the frames to get the duration. We also exact RGB bytes of the first frame and image resize method to generate a thumbnail. Lastly, we write the RGB bytes of first frame to a temporal PPM file for future use. For live RTSP stream, to get the dynamically updated frames, we open a connection to the source stream and start decoding using Plasma. With the method of Plasma, we successfully make the thumbnails in Browsing panel and Editing panel to be live. To preview a stored media ,we current use MPlayer to achieve the highlight playback and resizable preview window features. For random access, user can use the arrow keys to forward or backward.

4.2. Editing

Drag and drop functions can be easily done with Tk code. However, resizes image task is hard to perform with Tk. So we write a Dalí function for resizing Tk photo. Moreover, another Dalí function is written to perform crop task. Both Dalí and Tk cannot make a Tk photo transparent. We solve this problem using TkZinc. The Editing panel and Crop panel was totally written in TkZinc. With the support of TkZinc, we can also write function to make text in *canvas* to be editable. Since the main contribution of SLIME is about real-time spatial editing of a live stream, we have focused our research on the spatial editing and network part. For temporal editing part, due to the time constraint, we are still improving our algorithm for represent the frame as a slice.

4.3. Processing

We use Plasma beta 0.2 as processing engine. For producing stored video we generate a plasma script and then execute it in a different process. For producing live media supporting real-time editing, we have to execute plasma command inside our application. To generate plasma script or command, we abstract medie objects and store them in a database. Then we use several algorithms to parse the database and generate plasma script or command.

5. EVALUATION

5.1. User Study

Since SLIME is design for making video editing as easy as word processing, we conduct an informal user study on three of our friends. We first give them and demonstration. Then we let them perform three tasks with different levels. Finally we collect the results video and begin a discussion to get their feedback. All the three students roughly finish the three tasks and give positive comments during the disscusion such as "It's very easy to master." The result shows that SLIME successfully make video editing as easy as word processing and the friendly user interface encourages the participants to explore new features.

5.2. Comparison

Most of the current commercial video editor which targets to average users only cannot perform complex spatial editing task. Although expert software can perform the same task, it requires user to spend many time to learn and make video editing to be a tedious work. For example, to five video together with a picture-in-picture effect, user have to input the precise size and position for all the five clips. Some research project can perform spatial editing task in a relative easy way. For example Hyper-Hitchcock (Shipman, Girgensohn, & Wilcox, 2003) proposed by Shipman use a tree structure to organize the composite videos. To construct the sample pattern, a 4-level tree is needed. So it is not enough easy to use for novice. Comparing with the above tools, SLIME has powerful feature and a fast and easy-to-use user interface.

5.3. Limitation

The main limitation of SLIME is the on its network part. Due to the limitation of Plasma, SLIME only supports the standard RTSP sessions and dose not support more popular Real Network. When receiving stream from a live streamer, more streams require more time to connec. Using video-on-demand servers instead of live streamers can solve this problem. However, since video-on-demand server is not live, we cannot use it to produce live webcast. To solve this problem, we can use capture device (e.g. TV card).

6. CONCLUSION

We have presented SLIME, an easy-to-use and video editing tool that allow user to real-time compose live and stored video together. In summary, our SLIME has two main features which are powerful spatial editing feature and live stream real-time editing feature. Moreover, the user-friendly GUI design of SLIME makes video editing as easy as word processing. In this report, we begin with introducing some software tools. We also exploited the powerful spatial editing features in Plasma, and review the existing CHI designs on video editor user interface. After we described the intuitive GUI of SLIME, we discussed the design and implementation issue of a prototype. Then we show and analysis the results our informal user study and give a comparison between SLIME and other related works. After address the limitations, we will finally recommend the future work.

There are several features could be added to SLIME such equipment control and QoS event handling. In this section, we will recommend our future works in following aspects. First and foremost, we will complete

our GUI with those features we haven't implemented. For the timeline, we will improve our algorithm to get better performance, and then integrate it into the main interface to support temporal editing such as cutting and concatenating. We will conduct a formal user study and get more feedbacks on the interface. We will integrate Indiva into SLIME so that our application can manage resource and control the equipments in the lecture rooms. Plasma has powerful feature on event handling. We will exploit other event features such as QoS event and remote controlling. For example, it will detect the packet loss and allow user to reduce the frame rate of a video steam.

7. REFERENCES

[1] Casares, J., Long, A., & et al. (2002). Simplifying video editing using metadata. *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques* (pp. 157–166), London, England, June, 2002.

[2] Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., & Wilcox, L. (2000). A semi-automatic approach to home video editing. *Proceedings of the 13th annual ACM symposium on User interface software and technology* (pp. 81–89), San Diego, California, November, 2000.

[3] Machnicki, E., & Rowe, L. A. (2002). Virtual director: Automating a Webcast. *Proceedings of 2002 SPIE Multimedia Computing and Networking* (pp. 56–58), San Jose, California, January, 2002.

[4] McCanne, S. (1999). Scalable multimedia communication with internet multicast, light-weight sessions, and the mbone. *IEEE Internet Computing*, *3(2)*, jun, 1999, 33–45.

[5] Ooi, W. T., Pletcher, P., & Rowe, L. A. (2001). Indiva: A middleware for managing distributed media environment. *Proceedings of Multimedia Computing and Networking*, San Clara, California, January, 2001.

[6] Ooi, W. T., & Smith, B. (1999). Dali: A mutimedia software livrary. *Proceedings of 1999 SPIE Multimedia Computing and Networking* (pp. 25–27), San Jose, California, January, 1999.

[7] Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*. Addison Wesley.

[8] Rowe, L., Harley, D., & Pletcher, P. (2001). *BIBS: A lecture webcasting system* (Technical report). Berkeley Multimedia Research Center.

[9] Rowe, L., & Jain, R. (2005). ACM SIGMM Retreat report on future directions in multimedia research. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, *1*, February, 2005, 3–13.

[10] Shipman, F., Girgensohn, A., & Wilcox, L. (2003). Hyper-hitchcock: Towards the easy authoring of interactive video. *Proceedings of the Human-Computer Interaction INTERACT '03* (pp. 33–40), Zurich, Switzerland, September, 2003.

[11] Wang, Y., Zhao, P., Zhang, D., Li, M., & Zhang, H. (2002). Myvideos: a system for home video management. *Proceedings of the 10th ACM international conference on Multimedia* (pp. 412–413), Juan-les-Pins, France, December, 2002.

[12] Yang, M., Mu, X., & Marchionini, G. (2003). Vivo: a video indexing and visualization organizer. *Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries* (pp. 403–403), Houston, Texas, May, 2003.

[13] Yu, T. P., Wu, D., Meyer-Patel, K., & Rowe, L. A. (2001). dc: A live webcast control system. *Proceedings of 2001 SPIE Multimedia Computing and Networking* (pp. 111–112), San Jose, California, January, 2001.

[14] Zhu, T. (2003). Design of a scripting language for processing streaming media. *Proceedings of NUROP Conference*, Singapore, October, 2003.