

SmartPlayer: User-Centric Video Fast-Forwarding

Kai-Yin Cheng[†]

Sheng-Jie Luo[†]

Bing-Yu Chen[‡]

Hao-Hua Chu^{*}

National Taiwan University

[†]{keynes,forestking}@cmlab.csie.ntu.edu.tw

[‡]robin@ntu.edu.tw

^{*}hchu@csie.ntu.edu.tw



Figure 1. Our SmartPlayer is adopted by the metaphor of scenic car driving.

ABSTRACT

In this paper we propose a new video interaction model called adaptive fast-forwarding to help people quickly browse videos with predefined semantic rules. This model is designed around the metaphor of “scenic car driving,” in which the driver slows down near areas of interest and speeds through unexciting areas. Results from a preliminary user study of our video player suggest the following: (1) the player should adaptively adjust the current playback speed based on the complexity of the present scene and predefined semantic events; (2) the player should learn user preferences about predefined event types as well as a suitable playback speed; (3) the player should fast-forward the video continuously with a playback rate acceptable to the user to avoid missing any undefined events or areas of interest. Furthermore, our user study results suggest that for certain types of video, our SmartPlayer yields better user experiences in browsing and fast-forwarding videos than existing video players’ interaction models.

Author Keywords

Video playback, adaptive fast-forward, predefined event detection, undefined event preserving.

ACM Classification Keywords

H5.1. Information interfaces and presentation (e.g., HCI): Multimedia Information Systems; H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION

Recent developments in digital technologies have made it easy for people to download, record, and watch videos on a variety of media access devices. Smart media access devices such as the TiVo [24] set-top box also learn users’ preferences about TV programs, and automatically record them for users. Additionally, inexpensive mass storage devices enable people to stock unwatched video content on hard disks. Despite the fact that digital content recording and storing technologies continue to improve over time, video playback systems have not changed much. Commercial video players such as Apple QuickTime Player [1], CyberLink PowerDVD [6], Microsoft Windows Media Player [18], and Real Network RealOnePlayer [21] offer comparable sets of simple controls for playing, pausing, stopping, fast-forwarding, and rewinding/reversing videos. When users have limited patience or time to watch the entire length of a video, they are obliged to manually skim and fast-forward to locate content of interest to watch in fine detail. This often involves tedious work on the users’ part. Hence, smart playback mechanisms are needed to help users efficiently skim through and fast-forward lengthy and boring content while slowing down to watch the good parts in fine detail.

Several video summarization methods [25] have been proposed to enable users to skim through content within a short amount of time. They can be categorized into two approaches: *still-image abstraction* and *video skimming*. The

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4–9, 2009, Boston, MA, USA.

Copyright 2009 ACM 978-1-60558-246-7/08/04...\$5.00

still-image abstraction approach extracts key-frames from a video which are used to compose a brief content summary. For example, key-frames can be played back sequentially as a slideshow or composed into an image mosaic [15]. Although *still-image abstraction* has been shown to be effective in helping people quickly obtain a general understanding of what is contained in a video, it does not provide sufficient information to users who want finer details on the parts they are interested in, either for comprehension or entertainment. In contrast to the *still-image abstraction* approach, the *video skimming* approach uses automated video analysis to extract segments that carry significant information, composing them into a short video summary. Predefined events or rules guide the decision on the significance of different video segments.

When fast-forwarding through new or unfamiliar content, since most users are not able to anticipate precisely where and whether the upcoming content matches their interests, they do not know when they ought to speed up or slow down the playback rate. Even if the key-frames or video summary extracted through automated video analysis are presented to users, they may still wonder or worry about missing any of what they consider the good parts, given that automated video analysis, based on predefined and incomplete event sets, often falls short of good semantic accuracy. Therefore, although our SmartPlayer adopts the *video skimming* approach, it does not skip any parts of a video but enables rapid skimming at a fast playback speed.

Li *et al.* [12] conducted a user study to explore how users benefit from advanced controls such as pause removal, notes, and table of contents. Because their player system preserves the audio pitch, it sets a maximum playback rate at 250% (*i.e.*, 2.5x) of the normal playback speed. Since their player system is designed specifically for lecture-type videos in which the audio is important, it may not be suitable for other motion- or event-centric types of video programs such as sports or surveillance. One noted finding in their studies is that although *video skimming* is applicable to a wide range of videos, people are unwilling to fast-forward video programs such as movies or entertainment shows. This finding was consistent with our user inquiry.

In this paper, we propose a new interaction model based on the metaphor of “scenic car driving.” Drivers adjust their speed according to road conditions as well as the quality of the scenery. When the scenery is monotonous or boring, drivers tend to speed up and skip it. When the scenery is complex and interesting, drivers tend to slow down to get a better look. They may also use GPS navigation and guide devices to inform them of any upcoming POIs (points of interest), and then slow down approaching these areas. When drivers encounter unlabeled POIs that the GPS guide device misses, as they are in control of the car, they can still slow down to get a better look.

Our SmartPlayer is designed based on this “scenic car driving” metaphor with the following features:

- The video player adaptively adjusts playback speed based on the complexity of the current scene and predefined semantic events.
- The video player learns the user’s preferences about predefined semantic event types as well as the user’s favorite playback speed when watching videos matching these event types. Learning user preferences is necessary for adapting video playback speed.
- The video player plays the video continuously with a user acceptable playback rate so as not to miss any undefined events or areas of interest.

The rest of this paper is organized as follows. We first review related work, and then describe a preliminary user inquiry. Based on the observations from this user inquiry, we then show how our SmartPlayer system is designed. Afterward, we outline the user study tasks that were conducted to evaluate our SmartPlayer. Finally, we discuss findings that can be generalized to other designs, as well as future work.

RELATED WORK

Many video interaction methods, described in [25], have been proposed to help users browse video content. Some interaction models, though less relevant to video browsing, have inspired our work. For example, Igarashi *et al.* propose speed-dependent automatic zooming (SDAZ) [10] in document browsing: when the scroll rate of a document is increased, it automatically zooms out semantically to allow for continuous reading. Ishak *et al.* propose content-aware scrolling [11] in which a document or an image is analyzed to derive its semantic scrolling path. Similarly, our SmartPlayer finds the semantic properties of a video and adapts the playback rate accordingly.

Other work involves novel control slider bars to improve the video browsing experience. Hürst *et al.* [9] incorporate the elastic graphical interface [17] into the video slider bar. The browsing speed is adjusted based on the distance between the mouse pointer and the thumb on the scrollbar. However, this mechanism does not lead to intuitive browsing speed control because the thumb moves continuously during video playback, and users must keep moving their cursor to maintain a certain browsing speed. Dragicevic *et al.* [8] developed a novel method for video interaction based on direct manipulation, in which video playback is controlled by directly manipulating an object in a video. Although playing with this direct video manipulation is fun for short video segments, it is not suitable for browsing long videos.

Other work uses *still-image abstraction* to construct content summaries, enabling users to quickly obtain a high-level understanding of what is contained in a video. Liu *et al.* [15] apply video abstraction techniques to identify important key-frames from a video and compose these key-frames into a “video collage”. One representation of a video collage is a mosaic, which also serves as a catalog with entry points to different video segments. Other forms of video

collages are possible, such as stained glass [4] or video manga [26]. Various work uses automated video analysis techniques to construct content summaries from interesting video segments and events [25]. Interesting events are identified by analyzing a variety of image features, including color, contrast, speech, closed captions, camera motion, and human faces. Domain-specific knowledge is often necessary to improve the detection accuracy of interesting video segments. For example, videos of baseball [5], tennis [23], weddings [3], movies [2], and news [22] call for analysis of different feature sets corresponding to what are considered interesting semantic concepts in the different video domains. To enable our SmartPlayer to work on diverse video types, we separate the video analyzers (the *semantic layer* in Figure 3) from the main player, such that it is an independent, pluggable unit. For example, when a news video is being played, the semantic layer loads the news domain-specific video analyzer that identifies interesting events for news.

Accelerating video playback enables users to efficiently browse videos. Pekar *et al.* [19][20] developed a method to accelerate playback speed according to motion activity in the video to maintain a “constant pace”. We adopt a similar method. However, unlike their approach, we do not arbitrarily choose a threshold to accelerate playback speed. Since results of our user study indicate that different users have different preferences and tolerances with respect to playback speed acceleration, we adopt a learning mechanism that adapts playback speed according to user preferences.

Other work has developed methods for constructing personalized video summarizations. Lie and Hsu [13] propose a method to generate personalized video summarizations by asking users to fill out questionnaires on their preferences. Compared to their approach, our SmartPlayer learns user preferences by observing how users change and override the playback speed set by the system, and thus does not require users to fill out forms. For example, if a user indicates his or her dislike for certain types of events by consistently speeding up to skip past them, the SmartPlayer will learn this user preference and increase the playback speed for these types of events.

USER BEHAVIOR OBSERVATION AND INQUIRY

Prior to designing our SmartPlayer, we performed a preliminary user inquiry on how users watch videos with the fast-forwarding mechanism.

There were 10 unpaid participants in this user inquiry: 5 males and 5 females. All participants were computer-savvy users with experience watching videos on computers. We listed many types of video programs, including short video clips, lectures, home videos, sports videos, movies, cartoons, news programs, travel videos, and surveillance videos, and asked the participants why and what types of video programs they prefer to fast-forward. Our findings are as follows. All 10 participants fast-forward surveillance videos, because surveillance videos are very boring, with relatively

simple events. 9 out of the 10 participants fast-forward sports videos, because sports videos have predefined patterns and rules which they can use to predict what they are interested in. No participant fast-forwards movies, because they expect good movies to be enjoyable from start to finish. 2 out of the 10 participants fast-forward lecture videos when slides are shown in the video program. The other 8 participants consider audio critical for understanding lecture videos, and speech comprehension requires time to think and reflect. Therefore, they think that lecture videos should not be fast-forwarded.

Thus, videos with predefined rules or simple events, such as surveillance and sports videos, are suitable for fast-forwarding. We then investigated *how* users fast-forward these videos. We prepared five types of video programs: surveillance, baseball, tennis, golf, and wedding videos. The length of each video clip was about twenty minutes. Participants were asked to watch these five videos as quickly as possible. Before testing, for each type of video, we also prepared training videos to familiarize the participants with the rules or event patterns. The testing and training videos are separated into different sets.

To analyze and record user playback behavior, we designed a prototype player with acceleration and deceleration buttons, for which the maximum playback speed was 16x normal playback speed. We also provided a hotkey to allow them to jump to the normal speed (1x) immediately, analogous to an emergency brake in a car. The prototype player recorded all participants’ button clicks and video-watching behavior for analysis. We also asked participants some questions to understand their video-watching behavior.

From our analysis, we discerned the following principles to guide the design of our SmartPlayer system. Participants had varying tolerance on the fast-forward speeds for different video types. The user-acceptable fast-forward speed for complex, motion-rich videos (*i.e.*, baseball) was much lower than that of slow videos (*i.e.*, golf). This is because in golf, progress is relatively simple and motion-less, the scene is simple, and the camera view does not move or pan too much. Based on this observation, we designed the learning mechanism to adjust playback speed according to user preferences.

Participants in general maintained a constant playback speed within one video shot (Figure 2) and seldom changed the playback speed dramatically. They preferred to gradually increase the playback speed, allowing their eyes to accommodate to a higher playback speed. When asked about their preferences about skipping in-between, non-event parts of videos, participants said that they prefer not to skip any parts on the first pass, because these in-between video segments provide “context to help them understand what’s going on” and enabled them to “watch the video more enjoyably”. Based on these findings, we designed our new video browsing interaction model into the SmartPlayer.

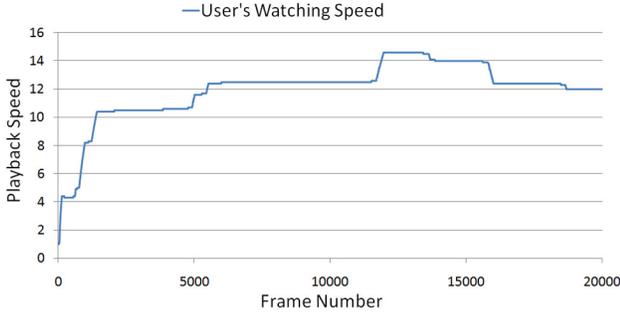


Figure 2. One user’s watching pattern for a baseball video.

USER-CENTRIC VIDEO FAST-FORWARDING

The goal of the SmartPlayer is to provide a better user experience when watching videos in fast-forward mode. Based on the metaphor of the “scenic car driving”, SmartPlayer automatically adjusts playback speed according to the complexity of the current scene and predefined events. In addition, SmartPlayer allows users to manually adjust or override the playback speed set by the system, thus allowing the system to learn individual user preferences for different events of interest as well as preferred playback speeds.

When first playing a video, SmartPlayer starts out in an automatic playing mode. This is similar to the autopilot in an airplane. In automatic playing mode, the video playback speed is automatically increased or decreased according to the current scene. We design a skimming model (discussed later) to formulate these principals. Findings from the preliminary user inquiry (described in the previous section) that are valuable for designing the SmartPlayer are summarized below.

- Users tend to maintain a constant playback speed within a video shot when they still want to know what is happening in the video.
- Users prefer gradual rather than sudden or dramatic increases of playback speed.
- Users set the playback rate based on several minutes of recently viewed shots.

Based on the observations, adjusting the playback speed frame by frame would not be desirable. Instead, SmartPlayer cuts the video into a number of segments and then adjusts the playback speed gradually across segment boundaries. Additionally, because users change the playback speed based on recently viewed content, the speed of the upcoming content should take into account not only the motion complexity of the upcoming content but also the playback speed of the previous content.

Instead of providing the control of playback speed with limited and discontinuous choices (*i.e.*, playback speed can only be set to 1x, 2x, 4x, ...) like most existing video players, SmartPlayer allows a seemingly continuous playback speed control at a fine increment of 0.1x up to the maximum speed of 16x. There is no frame dropping during fast-

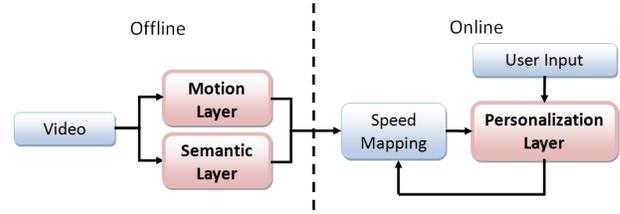


Figure 3. Flow of the user centric fast-forwarding mechanism.

forwarding so as to preserve the experience of continuous video watching.

By default, SmartPlayer automatically changes speed according to scene complexity. If users dislike the current playback speed under the automatic playing mode, they can manually reset the playback speed. In manual mode, the player adjusts playback speed only according to user input. In the following sections, we will describe each of the features in SmartPlayer as well as the underlying technologies.

Skimming Model

To allow for automated playback speed adjustment, three software engines (Figure 3) have been developed. These engines correspond to the *motion layer*, the *semantic layer*, and the *personalization layer*. The *motion layer* adapts the default playback rate according to detected motion between frames, in which higher motion maps to a lower speed, and vice versa. The *semantic layer* detects predefined semantic events in the video, and the *personalization layer* learns user preferences by analyzing the user’s previous video browsing behavior. The design and implementation of these three engines is described in the following sections.

Motion Layer

In order to support adaptive fast-forwarding, it is essential to gauge the similarity between scenes. We use two low-level features for this: color and motion. Calculating color histogram differences between frames [14] allows us to detect shot boundaries in a video.

To estimate the motion magnitude between two frames, we extract optical flows between frames using the Lucas-Kanade method [16], which is a widely-used motion estimation approach. The motion magnitude between two frames is computed using the following equation:

$$M_f = \frac{\sum_{i=0}^{N-1} \|\mathbf{V}_i\|}{N \cdot \max(\|\mathbf{V}_i\|)}$$

where i is a pixel in the current frame f , \mathbf{V}_i is the motion vector to the next frame $f+1$ of pixel i , and N is number of pixels in one frame. If $M_f \approx 0$, the playback rate is set to the maximum (*i.e.*, 16x) according to our user inquiry. If $M_f \approx \max(M_f)$, the playback rate is set to the normal (*i.e.*, 1x), where $\max(M_f)$ is trained from a huge amount of video clips. Otherwise, the playback rate is set proportionally to a value between the maximum and normal speed based on the motion magnitude M_f .

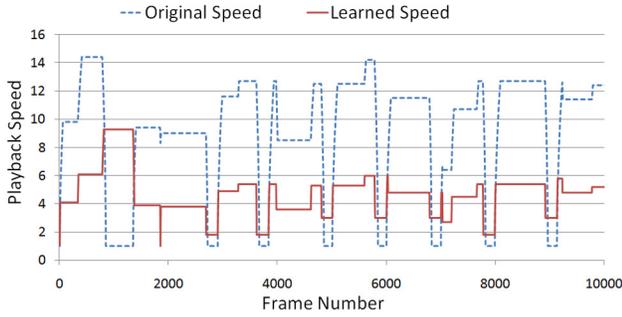


Figure 4. The user’s behavior and preferences (red line) are learned by the SmartPlayer and used to adjust the original speed (blue dotted line).

Semantic Layer

The *semantic layer* extracts semantic event points in a video. To effectively extract these event points, predefined domain-specific inference rules are required, for instance those for sports [5][7][23] and weddings [3]. As the *semantic layer* is domain-specific, it uses a plug-in framework in which different inference rules can be inserted to process different domain-specific videos.

Since our system focuses on how to adjust the fast-forward speed, we used manually annotated semantic events in the testing video clips. Note that such manual annotations can be replaced by an automated event detector such as MagicSport [7] for baseball videos.

Personalization Layer

The *personalization layer* is used to learn user preferences. In SmartPlayer, users can adjust the playback speed if they dislike the current playback speed set by the automatic playing mode. By learning from user input, SmartPlayer updates user preferences with respect to video playback speed. We calculate the new video playback speed by linearly interpolating the original playback speed and the user’s input speed as $S'_e = \alpha S_e + (1 - \alpha) S_e^u$, where S'_e , S_e , and S_e^u are the updated, original, and user input playback speeds for the predefined event type e ; the weight α is set to 0.95 based to user feedbacks. A video segment with no predefined event is treated as event type e_{none} , corresponding to a “none” event.

To learn user preferences for various event types, the default playback speeds for all predefined events are initially set to the normal speed (1x). If a user dislikes one specific event type, he or she will accelerate the playback speed through this specific event. The SmartPlayer thus learns to adjust the playback speed when the same event type is encountered in the future. Figure 4 shows one of the learning results. The blue line shows the default playback speed as generated by taking into account each scene’s motion complexity and the detected predefined events. The red line shows the learned speed.

User Interface

Figure 5 shows the SmartPlayer user interface. In addition to the basic control buttons (play, pause, and stop), the playback speed is shown at the center of the control panel dashboard to match the “scenic car driving” design metaphor. When the playback speed changes, the needle swivels to the current speed. The numeric playback speed is also shown to the right.

Visualizing the scene complexity and semantic events in a video helps users grasp the temporal locations of potential interesting events. We designed an improved seeker bar (Figure 5), shown near the bottom of the SmartPlayer control panel. This bar is similar to the scented widgets proposed by Willett *et al.* [27], which use embedded visualization to enhance the graphical user interface controls. Our visual scent on the video seeker bar is encoded by the amount of saturation on the red color. If a video segment has a relatively high amount of motion, its red color saturation value on the seeker bar will be higher than those of other video segments. This indicates that the SmartPlayer will likely slow down when playing this motion-rich video segment.

USER TESTING

To assess how well the SmartPlayer improves the user’s experience for browsing video, we recruited test subjects and asked them to perform the following two tasks, during which we collected their video watching data. The first task involved using the SmartPlayer to browse through several selected videos of the target types, from which user data was collected to analyze the functional usability of the adaptive fast-forwarding mechanism. The second task involved browsing selected videos using the SmartPlayer and other video players, from which user data was collected to compare the effectiveness and user satisfactory of the SmartPlayer with that of the traditional player, such as Apple QuickTime Player [1] and Microsoft Windows Media Player [18], and the event-based player, which plays only system-detected, predefined events and skips other video segments.

Apparatus and Participants

We recruited 20 unpaid participants including 13 males and 7 females. They were all computer-savvy users with experience watching videos on the computer. Our prototype player was run on a desktop PC with an Intel Core 2 Duo 2.4GHz CPU with 2GB RAM running Microsoft Windows XP Professional SP3.

Task 1: Personalized Adaptive Fast-forwarding

Procedure and Measures

Participants were asked to watch videos using the SmartPlayer. Five types of videos were selected: surveillance, baseball, news, drama, and wedding videos. Each type of video included one training video and five testing videos. The training video was used to familiarize participants with the SmartPlayer user interface. Five testing videos were used in the actual user testing. Each video clip was

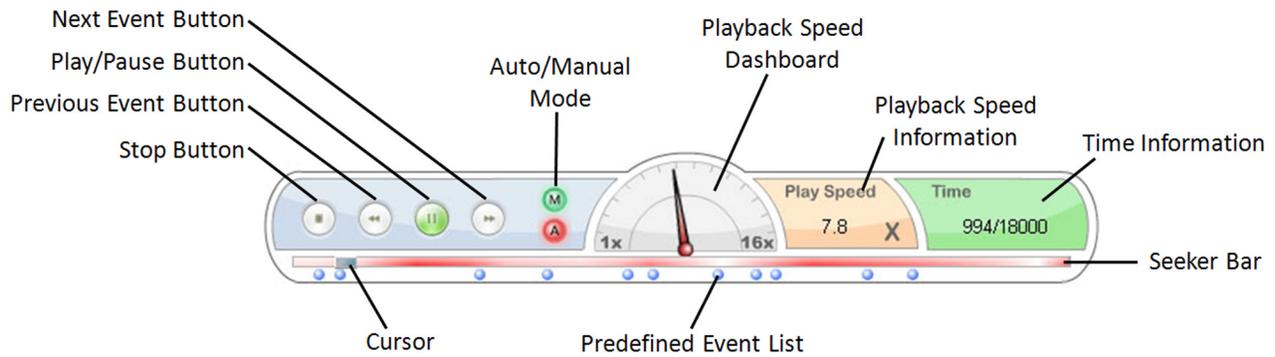


Figure 5. The user interface and functions of the SmartPlayer.

around 10 minutes long. Prior to the user testing, we used a short 5-minute briefing to explain the functions of the SmartPlayer. Then we asked participants to watch the videos as fast as they could while trying to understand the content. After participants watched the videos, we interviewed them (1) to assess how much actual content they comprehended and (2) to understand their fast-forwarding strategies using the SmartPlayer's functions. Additionally, the program also recorded the participants' manual fast-forwarding behaviors for later analysis.

Events in baseball videos were defined and classified according to well-known baseball rules, such as pitch, hit, homerun, *etc.* Events in surveillance videos were defined and classified based on the appearance of pedestrians, cars, and bicycles, *etc.* Similarly, events in wedding videos were defined and classified according to the formal wedding procedure. Events in news reports were categorized into political, financial, life, and international event types. Since it was difficult to define events in drama videos, no event was defined.

Results

Figure 6 shows the average number of manual adjustments for the 20 participants who used the SmartPlayer to watch five videos in each of five video types (*i.e.*, baseball, surveillance, news, drama, and weddings). Our three main findings include the following. (1) In all five video types, the average number of manual adjustments exhibited a decreasing trend from the showing of the first video to the fifth. This suggests that as each participant watched more clips of the same type of video, the SmartPlayer learned more about his/her preference, thus resulting in a reduced number of manual adjustments. (2) The SmartPlayer's learning mechanism was more effective for certain video types, such as surveillance, baseball, and wedding videos, than other types, such as news and drama videos. The results for surveillance, baseball, and wedding videos were expected because surveillance videos have explicit events, and wedding and baseball videos have explicitly defined rules. We found two participants who did not make any adjustment to the automated playback speed when watching baseball videos. These two participants remarked that the automated playback speed was appropriate and that the sys-

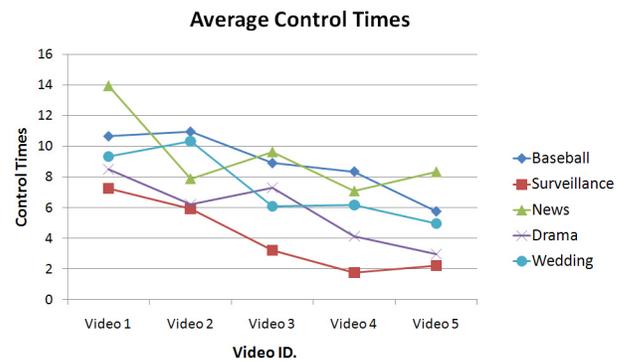


Figure 6. Average manual adjustment times of the tested five types of video.

tem-extracted events matched their interests. (3) An unexpected result was that the SmartPlayer's learning mechanism also proved somewhat effective for drama videos. From the analysis of the manual fast-forwarding behaviors and the user interviews, participants adjusted playback speeds to be no higher than the speed at which they could follow the subtitles in the drama videos. A similar phenomenon was also observed for news videos. Although subtitles effectively improved the learning mechanism of fast-forwarding, the bottleneck becomes the playback speed at which viewers can follow the subtitles (approximately 2x to 5x normal playback speed). We learned that effectively leveraging subtitles can also help users fast-forward videos.

Discussion

Participants had specific preferences for different categories of news. For example, some participants were not interested in political news, and hence consistently fast-forwarded such videos. However, when we asked these participants if they wanted to skip all political news completely, they answered no because they still wanted to know the political news for that day, which they indicated to be the reason for watching news.

Due to the high fast-forwarding speed we muted the audio. Participants found that the lack of audio for certain video types, such as news and wedding videos, degraded the viewing experience because the vocal content was important for comprehension. Therefore, subtitles might be help-

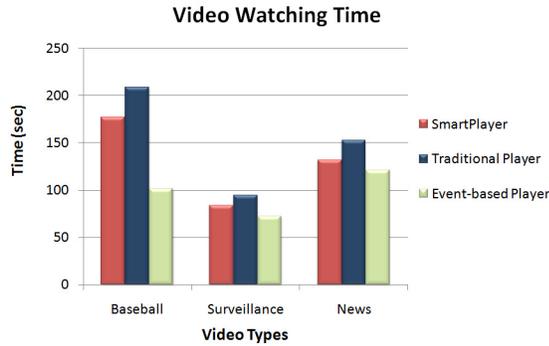


Figure 7. Average video watching time.

ful. Although most participants did not like watching videos without any audio, they also did not want high-pitched audio from high-speed fast-forwarding. Providing audio in fast-forwarding mode is a future challenge.

Task 2: Comparisons of Different Video Players

Procedure and Measures

Participants were asked to watch videos using three video players, which are the SmartPlayer, the traditional player, and the event-based player. Three different video clips were prepared for each video type so that each participant would not watch the same video clip repetitively on the three different video players. Additionally, the playback order for the three video clips was set randomly on the three video players for the different participants, thus reducing the ordering effect on the video clips. Each video clip was approximately 10-minute in length when played at regular speed. For user behavior analysis, our system recorded the total watching times for each video clip on each of three video players by each participant. After watching the video clips, participants were asked to fill out questionnaires containing five true/false questions to assess their comprehension of the video contents. After completing the questions, the participants also filled out qualitative questionnaires about their preferences with respect to the three video players and their experiences fast-forwarding different types of videos.

Results

Figure 7 shows the average video watching time for the 20 participants who used the SmartPlayer, the traditional player, and the event-based player to watch baseball, surveillance, and news videos. On average, participants spent more watching time using the traditional player than that using the SmartPlayer and the event-based player. The event-based player had the least amount of watching time because it skipped all of the non-event segments. Since some undefined events were embedded in the skipped segments, participants missed important information. We believe that there are two main reasons for the traditional player's requiring more time than the SmartPlayer. First, the SmartPlayer provides an event detection mechanism and marks detected events on the seeker bar as shown in Figure 5, these marks can be seen as good hints to adjust the play-

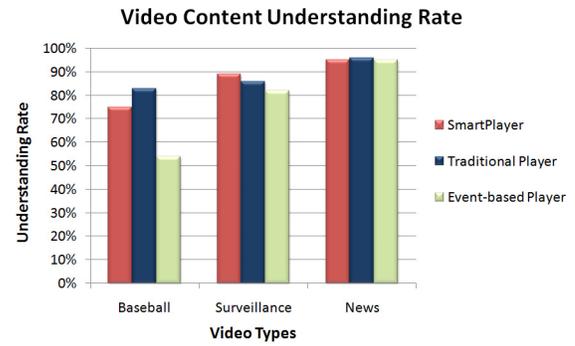


Figure 8. Average video content understanding rate.

back speed while playing videos. Second, the SmartPlayer adjusts playback speed according to scene complexity and detected events. Hence, when using the traditional player, users do not have any information about what will happen next, and therefore watch the video with relatively slow speeds. This is like driving on an unfamiliar road; we tend to slow down when we do not know enough about the surrounding environment.

Figure 8 shows the average video comprehension levels from the 20 participants who watched baseball, surveillance and news video clips on the SmartPlayer, the traditional player and the event-based player. Two main findings were described as follows. (1) On average, participants had better content comprehension using the traditional player than when using the SmartPlayer and the event-based player. The average comprehension level for the SmartPlayer was similar to that of the traditional player; this means that while using the SmartPlayer, users can still effectively understand the video contents. (2) No significant difference was observed for news videos. This is likely because users usually can understand a news story by its title.

Figure 9 shows the average ratings (a higher score means better preference), calculated from the results of questionnaires filled out by 20 participants, for each of the three video players in watching baseball, surveillance, and news videos. For baseball and news videos, participants preferred the SmartPlayer over the other two video players. For surveillance videos, participants preferred the event-based player over the other two players because surveillance videos are extremely boring and non-event segments are usually meaningless to viewers. Note that participants missed some important undefined events using the event-based player. In comparison, if the non-event segment provided meaning to the viewers, they preferred the SmartPlayer, because the SmartPlayer preserved in-between video segments to help them comprehend what was going on, and because the fast-forwarded video segments also contained interesting yet undefined events. For an example, during discussion with participants about the contents of a baseball video, many participants noticed many interesting yet undefined events, such as coaches coming on the field to negotiate with the referee, audience played waving, bats broking, etc.

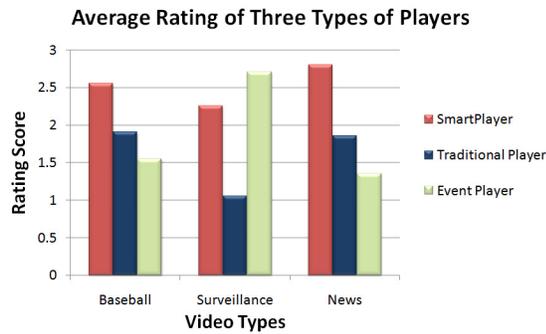


Figure 9. Average rating of three types of video players.

While asked about fast-forwarding the three types of video, all of them wanted to fast-forward surveillance videos, 18 out of 20 wanted to fast-forward news videos, and 17 wanted to fast-forward baseball videos. Thus a fast-forwarding mechanism for such kinds of videos is appropriate and desirable.

Discussion

Table 1 shows the results from our usability test that compares the SmartPlayer with the traditional player and the event-based player. Our findings suggest that the SmartPlayer helps participants watch videos in fast-forward mode, reduces watching time from that of the traditional player, maintains a pleasurable viewing experience, and unlike the event-based player, does not cause participants to miss any interesting content.

Player/Features	Smart Player	Traditional Player	Event-based Player
Total Watch Time	medium	Long	short
Predefined Event Understanding	O	O	O
Undefined Event Understanding	O	O	X
Personalization	O	X	X

Table 1. The comparisons of three types of players.

Compared to the traditional player, participants do not need to manually adjust the playback speed all the time, because the SmartPlayer can learn their preferences with respect to playback speed. Moreover, providing seemingly continuous playback speed control at fine increments (0.1x) may be more suitable for a large range of users. While using the traditional player, users are limited to playback speeds of 1x, 2x, 4x, etc., and thus cannot fine-tune the playback speed according to their true preference. Hence, providing fine increments also follows the spirit of the universal design principle.

In comparison to the traditional player and the event-based player, the SmartPlayer also provides personalization to help participants browse videos effectively at their preferred playback speed. Note that the SmartPlayer is not suitable

for browsing all types of videos. From our findings in the user inquiry, we found that although certain video types such as sports and surveillance videos were suitable for automated fast-forwarding, but other video types such as movies and lectures were not. Video types such as news programs may or may not be suitable, depending on whether they have clear patterns or rely on audio information for understanding.

To generalize our design concepts, our fast-forwarding mechanism is suitable for videos with the following characteristics:

- The audio parts of the videos are of secondary importance for understanding the videos. For example, lecture talks are not suitable for high-speed fast-forwarding because the user’s attention is mainly focused on understanding what the speaker is saying.
- The motion in the videos can be easily interpreted and understood by viewers. For example, viewers can often guess that a baseball player has hit a homerun from the player’s body movement and the subsequent celebratory scene.
- The videos are long and/or boring. From our findings, the two main reasons for fast-forwarding a video are that (1) viewers do not have enough time to watch the entire video and (2) viewers perceive the video as boring.
- The events in the videos follow predefined and known rules. For example, each sport has its own game-play rules, which enable our system to automatically recognize and learn events that are of interest/disinterest to users.
- The content of the videos follows standard procedures and patterns. For example, wedding videos often have formal procedures, and news videos often follow formal patterns with interlaced news gathering at the scene and the announcer’s report.

CONCLUSIONS & FUTURE WORK

In this paper, we propose a new interactive video browsing model, in which the design concept adopts the metaphor of “scenic car driving”. From observations in the user inquiry, our SmartPlayer automatically adapts its playback speed according to the scene complexity, any predefined events of interest, and the user’s preferences with respect to playback speed. Additionally, the SmartPlayer also learns the user’s preferred event types and the preferred playback speeds for these event types from the user’s manual adjustments. Our user study shows that as a user watches videos over time, the SmartPlayer effectively learns his or her preferences to make more accurate playback speed adjustment. Moreover, not skipping any video segments (*i.e.*, rapidly fast-forwarding through the less-interesting segments that precede the more interesting segments) maintains a sense of context and enhances the user experience in browsing and comprehension.

Future work will improve upon the limitations of the SmartPlayer found in our user studies. The predefined event

points on the seeker bar as shown in Figure 5 provide semantic meaning to the users. For example, if users are familiar with the rules of the baseball game, they can guess how the game will progress according to the distribution of the events. If two adjacent events are separated by a long period of time, users may guess that our system has missed events of interest between these two events. To help users make accurate guesses, advanced visualization techniques can be provided on the event slider bar such as coloring various types of events with unique colors.

The current SmartPlayer mutes the audio during fast-forwarding. If the videos are event- and motion-centric, users can still understand the content. However, lack of background audio degrades the user watching experience. We hope to provide quality audio during fast-forwarding to accompany the video.

Our learning function weights the previous playback speed and the user's input speed to provide an updated video playback speed. If a user seeks to change the video playback speed to his or her preferred value, he or she must input the new value several times. Therefore, in the future we might alter the learning function to consider not only the frequency but also the duration of user's input. In addition, we found that users sometimes accidentally accelerate events. However, the length of some event types is so short that users cannot train the SmartPlayer to reduce the speed again after they have made this mistake. To handle this problem, we might provide a semantic acceleration mechanism such that if the event's length is too short, the SmartPlayer would adjust its speed only slightly according to users' input speed.

Though our interactive video browsing model is designed for browsing video with predefined rules, such as sports or surveillance videos, there may be the potential to extend our design concepts to different types of videos and apply it to different use scenarios, which we can explore more in the future.

ACKNOWLEDGMENTS

We gratefully acknowledge helpful comments and suggestions from the Associate Chair, and the anonymous reviewers. We would also like to thank to Shang Chou, Ming-Yang Yu, and Ken-Yi Lee for their help, and to the users who perform the usability testing and provide significant insight comments on our SmartPlayer. This paper was partially supported by the National Science Council of Taiwan under NSC97-2622-E-002-010 and also by the Excellent Research Projects of the National Taiwan University under NTU97R0062-04.

REFERENCES

1. Apple QuickTime Player, Apple Corporation Inc., <http://www.apple.com/quicktime/>
2. Chen, H.-W., Kuo, J.-H., Chu, W.-T., and Wu, J.-L. Action movies segmentation and summarization based on tempo analysis. In *Proc. ACM SIGMM International Workshop on Multimedia Information Retrieval 2004*, (2004), 251-258.
3. Cheng, W.-H., Chuang, Y.-Y., Lin, Y.-T., Hsieh, C.-C., Fang, S.-Y., Chen, B.-Y., and Wu, J.-L. Semantic analysis for automatic event recognition and segmentation of wedding ceremony videos. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 11 (2008), 1639-1650.
4. Chiu, P., Girgensohn, A., and Liu, Q. Stained-glass visualization for highly condensed video summaries. In *Proc. IEEE International Conference on Multimedia and Expo 2004*, (2004), 2059-2062.
5. Chu, W.-T. and Wu, J.-L. Explicit semantic events detection and development of realistic applications for broadcasting baseball videos. *Multimedia Tools and Applications* 38, 1 (2008), 27-50.
6. CyberLink PowerDVD, CyberLink Corporation Inc., http://www.cyberlink.com/multi/products/main_1_ENU.html
7. CyberLink MagicSports, CyberLink Corporation, Inc. http://www.cyberlink.com/multi/products/main_75_EN_U.html
8. Dragicevic, P., Ramos, G., Bibliowicz, J., Nowrouzezahrai, D., Balakrishnan, R., and Singh, K. Video browsing by direct manipulation. In *Proc. ACM CHI 2008*, (2008), 237-246.
9. Hürst, W., Götz, G., and Jarvers, P. Advanced user interfaces for dynamic video browsing. In *Proc. ACM Multimedia 2004*, (2004), 742-743.
10. Igarashi, T. and Hinckley, K.. Speed-dependent automatic zooming for browsing large documents. In *Proc. ACM Symposium on User Interface Software and Technology 2000*, (2000), 139-148.
11. Ishak, E. W. and Feiner, S. K. Content-aware scrolling. In *Proc. ACM Symposium on User Interface Software and Technology 2006*, (2006), 155-158.
12. Li, F. C., Gupta, A., Sanocki, E., He, L.-W., and Rui, Y. Browsing digital video. In *Proc. ACM CHI 2000*, (2000), 169-176.
13. Lie, W.-N. and Hsu, K.-C. Video summarization based on semantic feature analysis and user preference. In *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 2008*, (2008), 486-491.
14. Lienhart, R. Comparison of automatic shot boundary detection algorithms. *SPIE Storage and Retrieval for Image and Video Databases VII 3656*, (1999), 290-301.
15. Liu, X., Mei, T., Hua, X.-S., Yang, B., and Zhou, H.-Q. Video collage. In *Proc. ACM Multimedia 2007*, (2007), 461-462.
16. Lucas, B. D. and Kanade, T. An iterative image registration technique with an application to stereo vision. In *Proc. Imaging Understanding Workshop 1981*, (1981), 121-130.

17. Masui, T., Kashiwagi, K., and Borden, G. R. Elastic graphical interfaces for precise data manipulation. In *Proc. ACM CHI 1995*, (1995), 143-144.
18. Microsoft Windows Media, Microsoft Corporation Inc., <http://www.microsoft.com/windows/windowsmedia/default.msp>
19. Peker, K. A. and Divakaran, A. An extended framework for adaptive playback-based video summarization. *SPIE Internet Multimedia Management Systems IV 5242*, (2003), 26-33.
20. Peker, K. A., Divakaran, A., and Sun, H. Constant pace skimming and temporal sub-sampling of video using motion activity. In *Proc. IEEE International Conference on Image Processing 2001*, (2001) 414-417.
21. Real Networks RealOne Player, <http://www.real.com/>
22. Sundaram, H. and Chang, S.-F. Video skims: Taxonomies and an optimal generation framework. In *Proc. IEEE International Conference on Image Processing 2002*, (2002), 21-24.
23. Tien, M.-C., Wang, Y.-T., Chou, C.-W., Hsieh, K.-Y., Chu, W.-T., and Wu, J.-L. Event detection in tennis matches based on video data mining. In *Proc. IEEE International Conference on Multimedia and Expo 2008*, (2008), 1477-1480.
24. TiVo Inc. <http://www.tivo.com/>
25. Truong, B. T. and Venkatesh, S. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications and Applications 3*, 1 (2007), 3.
26. Uchiashi, S., Foote, J., Girgensohn, A., and Boreczky, J. Video manga: Generating semantically meaningful video summaries. In *Proc. ACM Multimedia 1999*, (1999), 383-392.
27. Willett, W., Heer, J., and Agrawala, M. Scented widgets: Improving navigation cues with embedded visualizations. In *Proc. ACM CHI 2007*, (2007), 51-58.