A New Interface for Video Browsing on PDAs

Wolfgang Hürst
Institute of Computer Science
Albert-Ludwigs-University Freiburg
Georges-Köhler-Allee, Building 51
79110 Freiburg, Germany
huerst@acm.org

Georg Götz
Institute of Computer Science
Albert-Ludwigs-University Freiburg
Georges-Köhler-Allee, Building 51
79110 Freiburg, Germany
mail@georg-goetz.de

Martina Welte
Institute of Computer Science
Albert-Ludwigs-University Freiburg
Georges-Köhler-Allee, Building 51
79110 Freiburg, Germany
welte@informatik.uni-freiburg.de

ABSTRACT
We present an interface for interactive video browsing on pen-based handheld devices. Our solution enables users to navigate through a video along the timeline at different granularity levels. In addition, one can skim a file's content by continuous modification of replay speed. Both interaction concepts are smoothly integrated into a single interface that takes full advantage of the limited screen space being available.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g. HCI)**]: User Interfaces – *Graphical user interfaces (GUI), input devices and strategies, interaction styles, screen design*

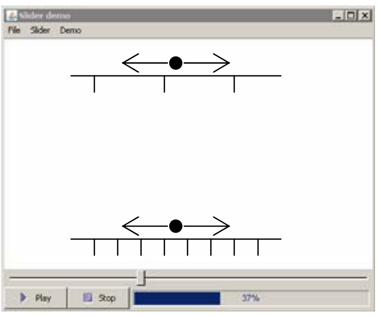
General Terms
Design, Human Factors.

Keywords
Handheld devices, pen-based computing, mobile computing, interface design, interaction, video browsing, video navigation.

1. MOTIVATION
The term "handheld" implies that screen space is (and always will be) a limiting resource for interface development on such devices. As a consequence, common approaches for video browsing (e.g. storyboards) can often not be applied to handheld devices such as PDAs. In addition, the available screen space for GUI widgets commonly used for video browsing (e.g. sliders to modify replay speed, etc.) is limited. Finally, a time-based slider (which is often used for navigation and browsing along a video's timeline) is less useful here because the limited screen size does not allow associating every frame from a video with a position on the slider's timeline. Hence, single frames or sometimes even whole scenes are skipped when moving the slider's thumb along the timeline.

In this demo, we present the *MobileZoomSlider* interface which realizes a slider-like navigation within a video at different granularities, thus enabling users to interactively skim a video's content at different levels of detail. In addition to such a *position-based* navigation, where a user directly accesses different positions along the timeline, the MobileZoomSlider also integrates *speed-based* navigation at the right window border, enabling the user to browse the file's content by modifying the replay speed.

2. THE MOBILEZOOMSLIDER
The basic idea of our MobileZoomSlider is to navigate through a video along the timeline by just clicking anywhere on the screen (i.e. without the usage of explicit GUI components, cf. Figure 1): If a user clicks somewhere close to the bottom of the screen and moves the pen to the left or right, he/she navigates through the video backwards and forwards, respectively, along a "virtual" timeline which has the same resolution as the original slider. The resolution of the virtual timeline's scale becomes continuously finer with increasing vertical position of the pen; the finest resolution (i.e. one pixel on the screen corresponds to one frame in the video) is associated with the upper border of the screen. This way, a user has immediate access to several scales for skimming at different granularity levels; coarser at the bottom (e.g. to get a rough overview of a file's content, quickly reach a specific position on the timeline), and finer at the top (e.g. to do an exact positioning, access singular frames, or explore single scenes in more detail).



Moving horizontally on the screen results in a navigation along a virtual timeline. The timeline's scale depends on the vertical pen position;
top: finest resolution (= detailed navigation),
bottom: coarsest resolution (= faster scrolling),
in between: linear interpolation between these two scales

Figure 1. Position-based navigation with the MobileZoom-Slider.

In addition to this position-based navigation, replay speed can be modified by moving the pen up or down in an area close to the right side of the screen (cf. Figure 2). Again, the browsing level depends on the vertical position of the pen. At the bottom, replay speed is faster, enabling users to fast forward through a file's content. At the top, a slow motion is realized, enabling users to take a more detailed look at, for example, a particular scene, access single frames directly, etc.

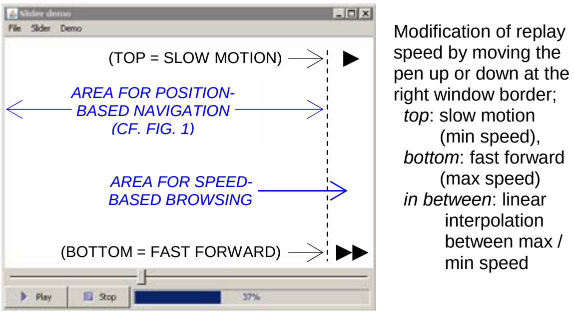


Figure 2. Speed-based navigation with the MobileZoomSlider.

Our approach has several advantages for video skimming on handheld devices. First, it provides powerful browsing functionalities (speed- and position-based at different speeds and granularity levels, respectively) without the need for any GUI components, thus taking full advantage of the available screen size without limiting the space for presentation of the actual video content. Second, by mapping slow speed / fine scales and fast speed / coarse scales to the top and bottom of the window, respectively, it smoothly integrates position- and speed-based navigation. Switching between both interaction modes can easily be done and integrates nicely into the overall interaction. Third, the interface is easy to understand and intuitive to operate, as we confirmed in an evaluation where different users had to solve several video browsing tasks using our implementation on a PDA.

3. IMPLEMENTATION

We implemented the interface concept described above on a Dell Axim^{TM} X51v [2]. This PDA model has an Intel XScale, PXA 270, 624 MHz processor, 64 MB SDRAM, 256 MB Flash-ROM and a screen size of 3.7’’ with a resolution of 640x480 pixels. Our implementation uses TCPMP (The Core Pocket Media Player), a high performance open source video player [1]. Hence, it is divided into two parts: The first one consists of TCPMP and its codec libraries. The other one is our custom interface for TCPMP, i.e. the implementation of the MobileZoomSlider (together with our reimplementation of standard player components such as play/stop buttons, etc.). Apart from having various codec libraries available, the main advantage of TCPMP is its performance even on mobile computer systems: When navigating in a video along the timeline, the visual representation of that video gets updated instantly. This is a major premise for realizing the MobileZoomSlider concept.

TCPMP can be compiled for several platforms. Among these platforms are Pocket PCs and Smartphones equipped with Windows Mobile as well as – with slight modifications – platforms equipped with desktop Windows systems such as Windows XP. The compilation process for Palm OS is yet to be tested.

Our custom interface is implemented based on the Win32 API using GDI (Graphics Device Interface) for rendering the graphics. Thus, we keep the source code portable among the different versions of Windows. The integration of TCPMP into our new interface is rather straightforward because it relies on the same APIs. TCPMP has its own build-in rendering routines optimized for video playback either based on GDI, DirectDraw, or GAPI (Game API) that is specific to Windows Mobile platforms.

4. DEMONSTRATION

In the following, we give a detailed description of the actual interface of our implementation. Figure 3 compares the original TCPMP’s interface with our reimplementation. While still keeping the same size for the video’s content, most GUI components are made larger and therefore, easier to target with the pen.

Figure 3. TCPMP’s original interface (left) compared to our implementation (right).

The MobileZoomSlider functionality can be accessed once the player is put to full screen mode, as illustrated in Figure 4. In this mode, standard interface components (play/stop button, progress bar, etc.) are faded in once a user clicks on the screen.

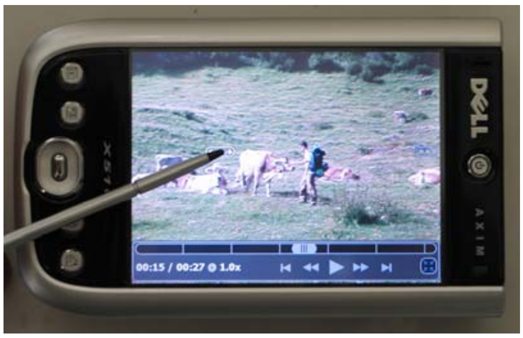


Figure 4. Full screen mode. The task bar with play/pause button, progress bar, etc. is faded in as soon as a user clicks anywhere on the screen and automatically disappear a few seconds after the user lifts the pen.

As described in the previous section, the right area of the screen is reserved for speed-based scrolling. This behavior is illustrated on the screen by different icons representing the various replay speeds as shown in Figure 5. In our evaluations it turned out that this kind of interaction is very comfortable and easy due to the natural, physical border around the screen: Users can just place the pen there and slide it up or down along the elevated part of the PDA’s case in order to modify replay speed (cf. Figure 6). Once the pen is released, replay continues at normal speed.

Figure 5. Speed-based skimming at the player's right border illustrated with different icons for different replay speeds.

Figure 6. The physical border resulting from the slightly elevated case around the PDA's screen makes sliding of the pen at the right window border easy and comfortable.

The whole rest of the screen space which is not reserved for speed-based scrolling and the standard interface elements is used for position-based navigation, as illustrated in Figure 7.

Figure 7. Illustration of regions with different functionality: Speed-based navigation (right), standard GUI components (bottom), position-based navigation (whole rest of the available screen space).

Figure 8 shows the visualization of different scale resolutions of the timeline: If a user moves the pen horizontally at the bottom of the screen, a coarse scale is used. If the pen is moved in the upper part of the screen, the timeline's scale gets finer, thus enabling a slower, more detailed navigation.

Figure 8. Different scale resolutions for position-based navigation depending on the vertical position of the pen.

5. CONCLUSION

In this document we illustrated the basic concept of our MobileZoomSlider approach for interactive navigation in videos, described its implementation on a PDA, and presented a detailed look at its interface. Our approach integrates position- as well as speed-based navigation at different granularities and replay speeds, respectively. The feasibility of our concept for video browsing on traditional desktop PCs has been evaluated in [3] and [4]. An evaluation of the PDA version described here can be found in [5]. An initial user study with 20 participants confirmed our statement that operation of the interface is easy and intuitive despite its complexity and the variety of offered interaction styles.

6. REFERENCES

[1] The Core Pocket Media Player: see http://tcpmp.corecodec.org

[2] Dell Handhelds, see http://www1.euro.dell.com/content/products/category.aspx/handheld

[3] W. Hürst and P. Jarvers: Interactive, Dynamic Video Browsing with the ZoomSlider Interface. *Proceedings of the IEEE International Conference on Multimedia & Expo*, Amsterdam, The Netherlands, 2005.

[4] W. Hürst: Interactive audio-visual video browsing. *Proceedings of the 14th Annual ACM International Conference on Multimedia*, Santa Barbara, CA, USA, 2006.

[5] W. Hürst, G. Götz, and M. Welte: Interactive video browsing on mobile devices. *Proceedings of the 15th Annual ACM International Conference on Multimedia,* Augsburg, Germany, 2007.