# INTERACTIVE, DYNAMIC VIDEO BROWSING WITH THE ZOOMSLIDER INTERFACE

*Wolfgang Hürst, Philipp Jarvers*

Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany
{huerst, jarvers}@informatik.uni-freiburg.de

## ABSTRACT

We present the ZoomSlider, a new interface for skimming and browsing video content in a flexible and interactive way. It circumvents common problems of existing video browsing approaches, such as their lack of scalability to large document sizes and unpleasant because jerky visual feedback. With the ZoomSlider a user can visually skim through video files at different granularity levels thus providing the power and flexibility needed for highly interactive tasks. First usability tests are presented in order to prove the feasibility of the overall concept.

## 1. INTRODUCTION

Browsing a digital video file is much harder than scanning static documents such as text. The reason for this is the dynamic, time-dependent nature of this medium. One of the most powerful approaches for video browsing is a time-based slider in combination with real-time random access to the video; immediately displaying any visual change in a file while the user moves a slider along a time line has proven to be a very convenient, intuitive, and flexible way to navigate and skim a video's content. However, this approach has one significant problem, i.e. that sliders do not scale to large document sizes. While documents can be arbitrarily long, the length of a slider is restricted by screen resolution and window size. Thus, not every frame of the video can be mapped to the slider's scale (cp. Fig. 1A). This results in a jerky visual feedback when the slider is moved by the user, what is generally considered to be disturbing and irritating. Even more critical is the fact that the corresponding parts of the file are skipped during skimming and therefore particular information might not be accessible directly at all. Solutions to this problem include, for example, the use of special hardware (e.g. two controllers, one for fine and one for coarse skimming) or additional interface elements (e.g. several slider widgets with scales of different granularities, see [1] for example). These approaches generally work well for tasks such as video editing. However, they are often limited to specific scrolling granularities (e.g. by the number of controllers) or require constant alternation between different interface elements (e.g. multiple slider widgets). Hence, they are not well suited for tasks such as information seeking or video annotation where users want to be able to quickly change between coarse and fine navigation without the need to switch between different interfaces or widgets, without the use of special hardware, and without the need to continuously reconfigure the scrolling granularity in a separate interaction. The PVslider introduced by Ramos and Balakrishnan [2] offers this kind of flexibility. However, it supports slider-like navigation only in a very small temporal interval (the "position region") before the slider goes into "velocity mode" where the user can modify the current position within a file only indirectly by variation of the scrolling velocity. In this paper we present the ZoomSlider as a new interactive way for video skimming which enables fine as well as coarse navigation in a document with one single interface. The ZoomSlider solves the problem of missing scalability of regular sliders and enables interactive, flexible browsing at random granularity levels.

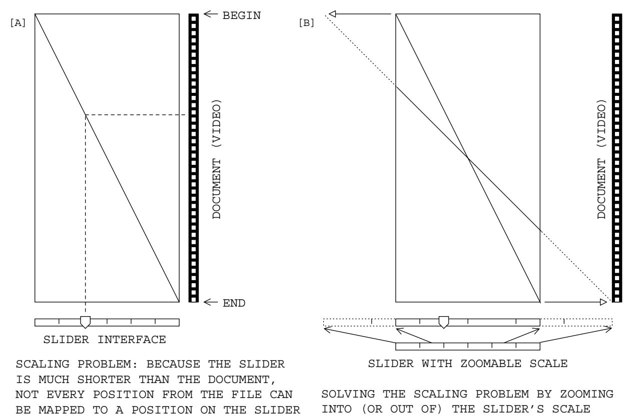


**Figure 1.** The scaling problem.

## 2. THE ZOOMSLIDER INTERFACE

**Basic idea**. One common approach to support fine navigation when a document is much longer than the size of the associated slider is to adapt the slider by zooming into (or out of) the slider's scale (cp. Fig. 1B). However,

*0-7803-9332-5/05/$20.00 ©2005 IEEE*

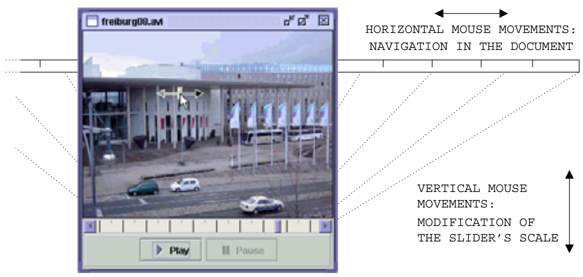


**Figure 2.** Basic idea of the ZoomSlider interface.

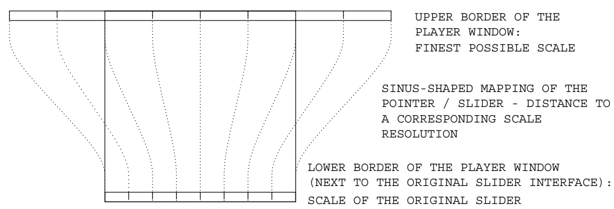


**Figure 3.** Modifying the scale resolution.

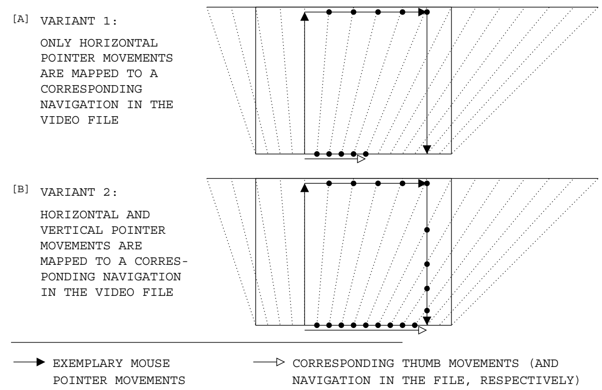


**Figure 4.** Navigation in the file.

generally the selection of an appropriate scale is done using different buttons or requires other, separate interactions by the user, thus reducing flexibility. In addition, separating the modification of the scale from the actual browsing process can be critical since the "best" scale for a specific situation and file is generally unknown. The basic idea of the ZoomSlider is to uncouple the scrolling from the actual slider widget and to use the second dimension which is orthogonal to the slider's orientation in order to continuously modify its scale. For example, with a horizontally mounted slider, horizontal mouse movements are mapped to navigation in the document while vertical movements are used to modify the scale. Moving the pointer away from the original slider interface results in a finer scale, as illustrated in Figure 2. This way, the ZoomSlider combines the possibility to continuously modify the scrolling granularity with the actual scrolling interaction.

**Modifying the scale resolution**. In a first implementation of this idea, the modification of the scale was done linearly depending on the distance between mouse pointer and original scale. We performed a heuristic usability evaluation with this prototype which identified that such a mapping of pointer position to slider scale results in a (subjective) abrupt scale change at the beginning which might irritate users and therefore be hard to handle. Hence, we changed the mapping of pointer position to slider scale as illustrated in Figure 3: The part of the player window on the opposite side of the slider is mapped to the finest reasonable slider scale resolution, i.e. one pixel on the scale is mapped onto one frame of the corresponding video. In between, a sinus-shaped mapping is performed, thus allowing the user to modify the scale in a continuous way between the finest possible navigation (near the window border) to the granularity implicitly defined by slider and document size (near the original slider). However, a user does not have to move the pointer up or down in order to modify the scrolling scale, but can click at any area of the window at any time and start scrolling with the corresponding granularity by moving the pointer horizontally. When clicking in the window, the current pointer position is always associated with the actual position in the document, while the corresponding slider scale is set depending on the pointer's vertical position.

**Navigation in the file**. Two options for the mapping of pointer movements to the actual navigation in the file were implemented and tested: In the first one, only the horizontal parts of the pointer movements are mapped to actual movements in the video (cp. Fig. 4A). In the second version, vertical pointer movements are not only used to adapt the scale resolution but to update the actual position within the file as well (cp. Fig. 4B). The first variant has the little negative side effect that pointer and thumb may drift apart (again, cp. the example illustrated in Fig. 4A). However, the second one turned out to be very confusing and harder to handle: Manipulating two parameters, i.e. scale resolution as well as document position, with one action turned out to put too much cognitive load on the users and therefore should be avoided. Having a strict separation of actions (horizontal = skimming, vertical = scaling) proved to be more intuitive and much easier to handle.

**Visualization**. During scrolling, an icon of the slider thumb is glued to the pointer and the scrolling direction is illustrated through two arrows next to this icon (cp. Fig. 5A). The length of the arrows depends on the current resolution of the scale. While this "virtual" thumb is dragged with the pointer (and therefore moves along the actual scale), the thumb of the original slider still represents the relative position of the actual frame within the whole video file. As a consequence, the original thumb moves very slowly if the virtual thumb is dragged in an area with a finer scale, thus giving little visual feedback to

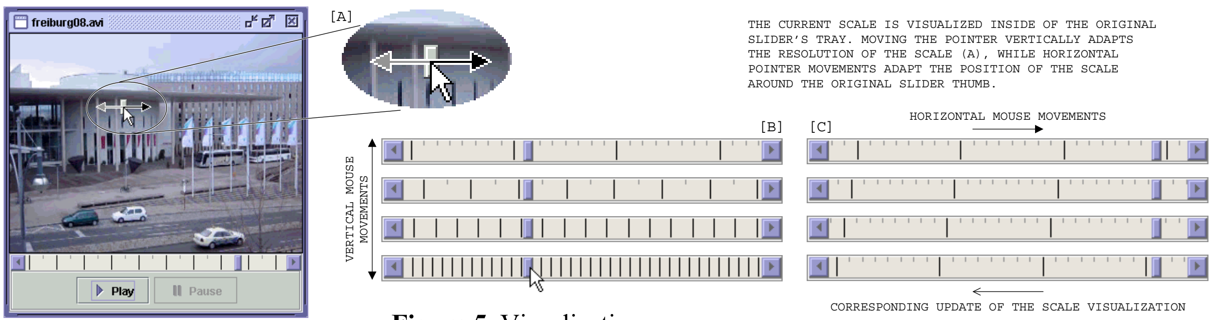


**Figure 5.** Visualization.

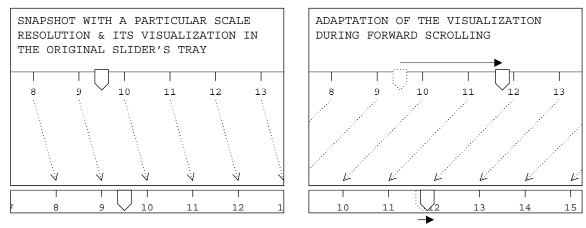


**Figure 6.** Example for the update of the scale's visualization.

the user. This was recognized as irritating during the heuristic evaluation mentioned above because it makes it hard for users to estimate the actual scrolling speed especially for parts of the file with low activity in the visual signal. Therefore we extended the visualization in the following way: The current scale resolution is illustrated through some ticks within the tray of the original slider. Moving the pointer up or down continuously adapts the distance between these ticks in an appropriate way (cp. Fig. 5B). Similarly, the positions of the ticks are continuously updated when a user scrolls through the file (i.e. when the pointer is moved horizontally; cp. Fig. 5C). With this adaptation the illusion of a slowly moving thumb is created, because it makes the ticks move opposed to the current scrolling direction (cp. the exemplary illustration in Fig. 6). Creating the illusion of movement of an object by moving its environment in the opposite direction is an approach which has been used successfully in other applications as well. For example, Ayatsuka et al. [3] use it in relation to skimming of static, time-independent documents. Their evaluations with such visualizations proved the usefulness and advantage of this approach. Informal tests with our implementation could confirm their arguments. It turned out that the continuous updates of the scale's visualization in the original slider's tray give a good impression for the current scale resolution and scrolling speed and therefore improve usability.

## 3. FURTHER USABILITY TESTS

As already mentioned, we developed a first prototype and did a heuristic usability study. This study was based on a well established framework proposed by Nielsen [4] for the evaluation of user interfaces, especially in early design stages. Such an evaluation is usually done with usability experts. Their comments identified several problems which lead to a revised interface design as described above. However, combining two separate actions, i.e. rescaling and scrolling, into one single interface might put a higher cognitive load on potential users. Therefore, we set up an additional evaluation with the revised interface design and a group of twelve users including seven male and five female members (ages: 17, 52, 54, all others between 20 and 30). None of them has seen or used the ZoomSlider interface before. The goal was to see if regular, "non-usability expert" users are able to handle the additional cognitive workload and if the interface can be used by them in a reasonable way.

**Setup**. After a short introduction, all users tested the ZoomSlider interface and compared it to a regular slider which did not have the possibility to modify the scale. During the test, the participants had to perform a specific search task where they had to find a particular scene in a video based on a given still image (one frame) of that scene. They had to solve this task with both, the ZoomSlider as well as a standard slider interface. Two short video clips were used in the evaluation (lengths 2:08 and 6:16 min). The size of the window for the video clips was 320x240 pixels. Screen size was 14'' (resolution 1024x768 pixels). As input device a regular mouse was used. The order of the sliders and the assignment of document to slider type were equally balanced among the participants to avoid any ordering effects. Data was gathered via the common think aloud technique, a questionnaire, and an informal interview with each participant at the end of the tests. In addition, all interactions were logged.

**Results**. Although the used video files were relatively short, the scaling problem was clearly observable and remarked by the participants as jerky, irritating visual feedback when the regular slider was used. All users were able to handle the ZoomSlider interface very well although they had no training before the evaluation. Overall feedback was very positive. Most users agreed that the ZoomSlider offered them more possibilities and adequate flexibility in browsing the files. Some participants noted that they had the impression that it took them a while to understand and get used to the operation of the ZoomSlider. But overall, all users agreed that after a short time the handling of the ZoomSlider became easy and intuitive.

These comments are confirmed through the results from the questionnaires which contained ten questions of three different categories: three about the general impression and overall reactions (category 1), three about the visualization (category 2), and four related to learning and general operation of the interface (category 3). Questions had to be answered on a scale from 0 (= worst) to 9 (= best). Users generally gave high ratings for the ZoomSlider interface. A significant difference ($p < 0.1$, Wilcoxon test) could be identified for the average ratings of all questions belonging to category 1 (average rating 7.28 for ZoomSlider vs. 5.42 for standard slider). Although being more complex to handle and having a less familiar visualization, users did not rate the ZoomSlider lower than the standard slider in relation to these issues, i.e. no significant difference could be identified between the average ratings in category 2 (7.39 for ZoomSlider vs. 7.11 for standard slider) and category 3 (7.19 for ZoomSlider vs. 6.75 for standard slider), respectively.

When solving the given tasks, all users performed a linear search strategy independent of the used interface, i.e. they navigated through the file from the beginning towards the end till the target position was found (with occasionally going back a little in order to check a previously skimmed position). Regarding the success in solving the tasks, it is generally less likely to find a target position with the standard slider interface (because it might be skipped due to the scaling problem illustrated in Fig. 1A). However, if a target position is accessible with a standard slider interface, we can expect the search time to be lower than with the ZoomSlider, because navigating at a coarser granularity means that fewer positions have to be accessed in the file before the target position is reached. Since it makes no sense to compare both interfaces for tasks which cannot be solved with one of them, we have chosen the tasks in a way that they were always solvable not only with the ZoomSlider but also with the standard slider, although such an assumption is unrealistic in a real-world setting. Based on this, it is noteworthy that the standard slider did not significantly outperform the ZoomSlider in terms of search time (average search time of 15.67 sec with the ZoomSlider vs. 12.89 sec with the standard slider, $p > 0.1$, t-test).

**Conclusion**. When comparing the results from the ZoomSlider with the standard slider interface, it is important to consider that to some degree it is obvious that the ZoomSlider gets a better user rating (because it extends the standard slider's functionality and offers more possibilities) and that the standard slider works faster (because of the unrealistic, artificial setup of this evaluation). However, the goal of this evaluation was not to compare both sliders with each other but to gather usability information from the users, i.e. to figure out if the users can handle the ZoomSlider at all and if they like it and see it as a reasonable and good extension for video browsing. Based on our initial usability study, both questions can be answered positively. Further evaluation and investigation is part of our future research.

## 4. SUMMARY

Time-based sliders offer a very good way for video browsing but lack scalability to large document sizes. The ZoomSlider tries to overcome this problem by using the vertical dimension in order to modify the scrolling granularity while navigating through the document in the other, horizontal dimension. The basic idea of this approach is simple and sounds straight forward; however, there are a lot of options and different degrees of freedom for an actual implementation. Therefore, we performed a heuristic evaluation with different variants which could identify specific problems and gave clues to the successful design of an interface with high usability. An initial usability study with the revised interface design and twelve random test users proved the feasibility of the overall approach. The ZoomSlider was identified as an interface which offers high flexibility and the ability to quickly browse a document at different granularity levels. A further advantage of the ZoomSlider is that it is not connected to any widget, what means that it enables users to browse a video even if it is watched in fullscreen mode. In addition, the separation from a particular widget might be an advantage in pen-based computing, since widgets such as regular sliders are often hard to target with a pen, especially on small devices. Therefore, evaluation of the ZoomSlider with different input devices is one key area of our future research.

## 8. REFERENCES

[1] Richter, H., Brotherton, J., Abowd, G.D., Truong, K. A.: Multi-Scale Timeline Slider for Stream Visualization and Control, GVU Center, Georgia Institute of Technology, Technical Report GIT-GVU-99-30. June 1999.

[2] Ramos, G., Balakrishnan, R.: Fluid interaction techniques for the control and annotation of digital video, Proceedings of ACM UIST 2003, p. 105-114.

[3] Ayatsuka, Y., Rekimoto, J., Matsuoka, S.: Popup vernier: A tool for sub-pixel-pitch dragging with smooth mode transition, Proceedings of ACM UIST 1998, p. 39-48.

[4] Nielsen, J. and Mack, R. L. (eds.) Usability Inspection Methods, J. Wiley & Sons, New York, NY, USA, 1994

**Acknowledgments**. This work was supported by the German Research Foundation's (DFG) research initiative "Distributed Processing and Exchange of Digital Documents (V3D2)".