Interfaces for Timeline-based Mobile Video Browsing

Wolfgang Hürst
Utrecht University
Department of Information and Computing Sciences
PO Box 80.089, 3508TB Utrecht, The Netherlands
huerst@cs.uu.nl

Konrad Meier
Albert-Ludwigs-University, Freiburg
Institute for Computer Science
Georges-Köhler-Allee, 79110 Freiburg, Germany
KonradMeier@gmx.de

ABSTRACT
Browsing video on mobile devices such as smartphones and PDAs requires new interface designs and interaction concepts because of their small screen sizes. In this paper, we present four different interfaces which enable users to skim video at different replay speed levels: An interface supporting flicking similar to text browsing on an iPhone™, an elastic slider, and two variations which also allow for interactive speed manipulation. Based on a heuristic evaluation with all four designs, revised versions of the two most promising ones have been implemented. A comparative user study proved the usefulness of the proposed designs. Both interfaces showed the same performance (measured in time needed to solve typical browsing tasks) but achieved different results in subjective user assessments.

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, Experimentation, Human Factors.

Keywords
Mobile video, video browsing, iPhone, flicking, elastic interfaces.

1. MOTIVATION
In contrast to earlier mobile devices – which were generally only able to handle small amounts of textual data – modern handheld devices can manage various media types and much larger file sizes. For example, Apple's iPhone™ or state-of-the-art PDAs (Personal Digital Assistants) allow you to read long text files, watch high resolution images, and play video clips as well as high quality videos of motion pictures. However, screen size remains a limiting factor when experiencing such files on handheld devices. Not surprisingly, one of the main reasons for the current hype around the iPhone™ is its innovative interface design, which includes several features enabling users to manipulate and experience larger data sizes despite its small display. For example, by tapping on the screen, users can easily manipulate the zoom level of a web page to a granularity which enables better reading. Multi-finger touch technology enables users to zoom into a high resolution image by just moving two fingers apart. By flicking your finger over the screen, you can browse a long text file or list of items at different speeds (cf. Figure 1).

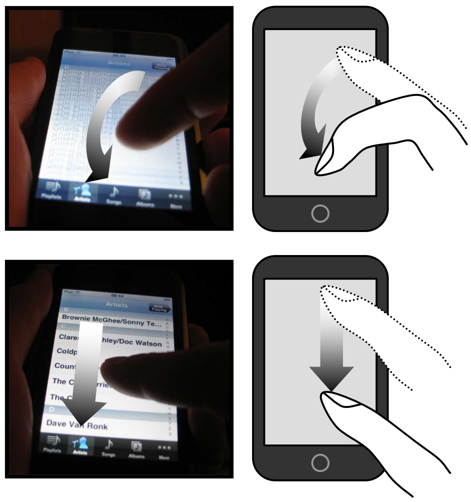

Figure 1. Flicking to browse long lists on the iPhone™.

Unfortunately, all these innovative interface concepts are only used for static media, such as text and images, whereas support for video browsing is limited to a simple navigation along the timeline using a common slider interface (at least at the time of this writing). Yet, especially mobile video – unlike watching TV at home – often asks for interactive manipulation of the timeline. For example, when people are replaying a news show recording while on the bus, they might not have enough time to see the whole show and thus, want to look at single messages in an order that fits their personal interests instead of the given linear order. They might want to skim parts of minor interest only quickly while looking at stories which have more relevance for them on a more detailed level. In such a noisy environment, they might also get easier distracted, resulting in situations where they just want to go back one or two sentences to re-listen to them. O'Hara et al.'s study [1] about how people use mobile video confirms these intuitive arguments. As a consequence, we need interfaces enabling users to navigate a video on a mobile device at different granularity levels in any direction along the timeline (e.g. to quickly skip a larger part of minor interest or to go back just a few seconds in order to re-listen to a sentence they just missed).

In this paper, we present four interface designs for video browsing on a PDA. All interfaces support different browsing tasks by enabling users to skim a file's content at various replay speeds and thus different granularity levels. After presenting some basic design issues (Section 2), we introduce the four interface designs and discuss some implementation details (Section 3). Based on a heuristic evaluation with all four designs, a re-design of the two most promising approaches is described and evaluated in a comparative study (Section 4). The results of this study prove the feasibility of the proposed designs and illustrate typical characteristics of our solutions which also highlight some issues that might have general relevance for user interface development on mobile devices (Section 5).

2. BACKGROUND & DESIGN ISSUES

In the following, we discuss basic problems and describe general issues related to video browsing and interface development for mobile devices. This discussion justifies some basic design decisions we made for the interfaces presented in this paper.

Video browsing approaches. Generally, existing video browsing techniques can roughly be classified into two categories: *structured approaches* which take advantage of a video's structure (e.g. enabling users to jump from scene to scene, displaying storyboards that represent a scene outline, etc.) and *interactive / timeline-based / user controlled approaches* which enable users to interactively skim along the timeline (e.g. by modifying replay speed or scrolling through a file using a slider). Visualization of metadata (if available) can be beneficial in both cases. Often, systems support both kinds of interaction because they complement each other. A simple but good example is a common DVD player: On your remote, you normally have buttons to navigate from scene to scene (structured browsing) and ones to increase and decrease playback speed (timeline-based browsing). Both functionalities are useful and important for video browsing. In this paper however, we restrict ourselves to timeline-based techniques. First, this gives us the possibility to do a more detailed and focused evaluation. Second, we think that user-controlled approaches are more important in a mobile scenario – which is quite different from watching DVDs at home (cf. Section 1). However, we also believe that a perfect interface should support both concepts. Integrating them into one interface is part of our future work which will be discussed in Section 5.

Touch screens vs. hardware solutions. Our goal is to provide an interface which enables users to browse a video by navigating along the timeline in random directions and at various granularity levels and thus different speeds. On the desktop, we can use the keyboard and additional input devices for this. In contrast, handheld devices provide us with limited hardware support for such advanced browsing techniques. The current trend of using touch screens offers an interesting opportunity in this context. First, they allow us to implement several buttons on the display, thus leaving more space for the screen. Second, it gives us full flexibility for implementing different interaction concepts and widgets known from desktop PCs, such as sliders which – if they provide real-time visual feedback – can be used to offer the requested interactive navigation along the timeline.

Widgets and GUIs vs. on-screen interaction. Slider-like interfaces are good approaches to support interactive video browsing on desktop PCs. However, when used on the screens of handheld devices, they have two major drawbacks. Icons and widgets on a cell phone or PDA are usually harder to target because of their small size and the mobile context. It is much different, to use a pen to interact with a mobile device that you are holding in the air, than moving your mouse over your desk's surface. In addition, the limited screen size normally does not provide enough space for many different GUI components. Thus, on-screen interaction techniques which do not rely on particular widgets but operate directly on the data (such as iPhone-like flicking, see Figure 1) seem to be a good solution in this context.

Scaling problem of sliders. Because of the small screen size, not every position within a video can be accessed directly by dragging the thumb of a slider along the timeline. In fact, for a long video, complete scenes might get skipped while moving the slider's thumb. Having multiple sliders at different granularity levels, as often done in video editing on desktop PCs, uses up too much screen space on small devices. Again, on-screen interaction without explicit widgets might offer a good solution here.

Position- vs. speed-based navigation. Scrollbars have a similar scaling problem and solve it by additional widgets, i.e. small arrows next to the bar for line- or even pixel-based navigation and sometimes clicking onto the bar for page-wise navigation. This doesn't work with video, since we have no comparable, fixed units here. For example, there are no pages in a video and frame-wise navigation in contrast to line-based navigation doesn't make much sense. Hence, manipulation of replay speed might be better suited for interactive navigation in a video.

Related work. Most video players on mobile devices which are available right now only offer very limited browsing and navigation support. Much research has been done on interaction and navigation on mobile devices in relation to static data. However, research related to video browsing is still very limited. This is partly due to the fact, that mobile devices only recently became powerful enough to support advanced browsing functionality. In addition, common timeline-based approaches for advanced interface concepts used on desktop PCs, such as Ramos and Balakrishnan's PVslider [2] often require too much space on the screen and are therefore not suitable for mobile devices (cf. [3]). Our own work in this context includes the MobileZoomSlider [3] and the ScrollWheel design [4]. Both approaches combine speed- with position-based navigation by providing several sliders at different granularity levels and a wheel-shaped interface, respectively. In contrast to this, the designs presented in this paper, focus on pure speed-based navigation. Further evaluations comparing performance and usability of the approaches presented here with our previously implemented interfaces work is part of our agenda for future work (cf. Section 5).

Used technology and limitations. In this paper, we restrict ourselves to pen-based interaction on PDAs. All interfaces presented here have been designed for this particular hardware and the evaluation results can not necessarily be generalized to other devices such as smartphones. Extension to finger-based interaction such as done on the iPhone™ is not considered here but again an issue we would like to address in the nearer future. An image of the player used for the implementations and evaluations presented in this paper can be found in Figure 2. Further details about the implementation, used tools and hardware will be discussed in Section 3.4.

Figure 2. TCPMP player (screen mockup and photo of the actual implementation on a Dell Axim^{TM} PDA).

3. INTERFACE DESIGNS

In this section, we describe the four interface designs that we implemented and evaluated for this paper. For each interface, we introduce its basic concept and discuss different design options as well as implementation variants (Section. 3.1-3.3). Implementation details shared by all designs are described at the end (Section 3.4).

3.1 Dynamic Flicking

Basic idea. Flicking allows users to interactively manipulate the visible part of a document by quickly moving a finger or pen over the screen, thus enabling them to skim the content of a file at different speeds and granularity levels, as illustrated in Figure 1 for navigation in long lists. Initial scrolling speed depends on the speed at which the finger or pen is moved over the screen; a fast moving finger results in faster scrolling, a slower movement gives the file a smaller momentum. After a while, scrolling slows down until the file comes to a complete stop. This kind of navigation is often complemented by a pure shifting of the file's content if the pen or finger is not flicked but continuously moved over the screen (cf. Figure 1).

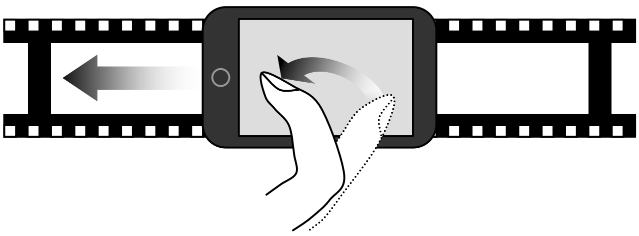

Figure 3. Applying flicking to video browsing.

Flicking for video browsing. Flicking has become popular with the iPhone^{TM} and iPod touch^{TM} where it is mainly used to browse long lists of music or video titles (cf. Figure 1). Studies on flicking for browsing static data include the work done by Aliakseyeu et al. [5]. An example for flicking techniques used in virtual environments can be found in [6]. To the best of our knowledge, our work is the first application of flicking to video browsing. If a video is seen as a sequence of frames which are observed by a user in a linear manner in the same way as letters, words, and lines of a text are read linearly, applying flicking to video browsing is straightforward, as illustrated in Figure 3. However, its usability remains unclear since video is perceived quite differently than a static medium such as text. For example, with text, a certain context (i.e. several lines) is usually visible creating a continuous visual flow during flicking which makes it easy for users to get a feeling for the actual scrolling speed. In case of a video, there might be long scenes with rather few visual changes (e.g. a speaking news anchor) followed by much shorter scenes with lots of visual activity (e.g. a short news clip from a sports event). As a result, users might find it much harder to develop a feeling for the actual scrolling speed during flicking – something we believe is quite essential for the actual usability of this technique.

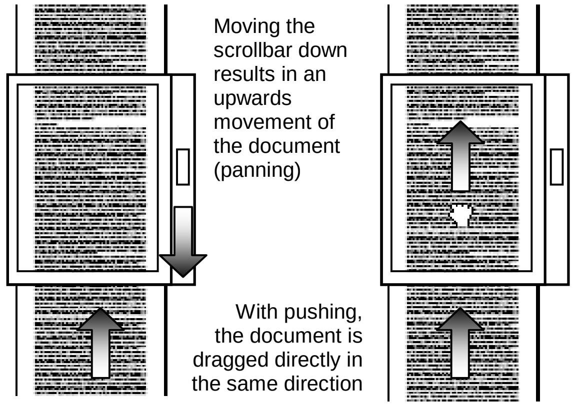

Figure 4. Panning vs. pushing (text documents).

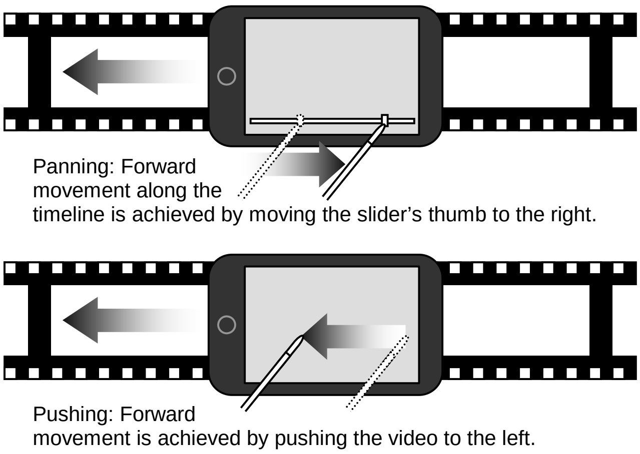

Figure 5. Panning vs. pushing (video).

Panning vs. pushing. There are two ways to manipulate the visible part of a document in a window on the screen. One way is to grab the document and move or push it directly. This approach is used for flicking, where the document moves in the direction in which the finger is flicked (cf. Figure 1 and 3). When scrollbars or sliders are used to manipulate the visible area of the screen, user interaction and document movement behave complimentary, i.e. moving the scrollbar down results in an upwards movement of the file and vice versa (cf. Figure 4). The latter approach is sometimes called panning because it resembles the pan of a camera over a static document. Combining flicking with panning does not seem to make sense for text browsing due to the constant visual flow of the document's content seen by the user. However, in case of video, users do not perceive this constant flow (cf. above) and because of the common way of navigating a video by scrolling along the timeline, panning might actually be more intuitive for flicking of videos (cf. Figure 5). Based on some initial, informal testing, we therefore decided to implement a panning approach for the final interface. Moving the finger over the screen from left to right, results in a forward navigation in the video, similarly to dragging or pushing the slider's thumb along the timeline.

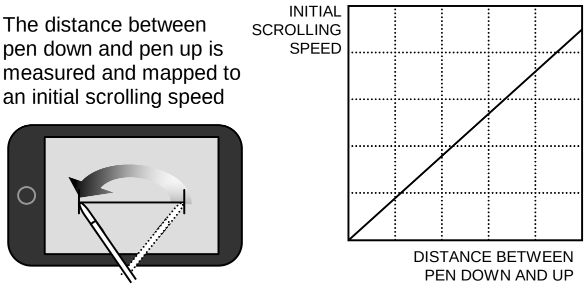

Figure 6. Initial implementation of flicking on our PDA.

Implementation problems and PDA-related issues. Whereas flicking on the iPhone^{TM} is done using your finger, interaction on the PDAs used for our implementation is done with a pen. Hence, we optimized the processing for pen input. Generalization to finger-based input is left for future work. Unfortunately, it turned out that the used hardware was not able to handle the pen-based interaction as easily as we thought. While there is no problem processing events for slow pen movements or single clicks on the screen, quick movements of the pen over the screen's surface did not turn out to be very reliable in terms of actual event processing. As a result, we implemented a different version of flicking. Instead of setting the scrolling speed of the video in relation to the speed at which the pen is moved over the screen (an approach which requires a very reliable event processing), we just mapped the length of the distance covered by the pen on the screen to a scrolling speed (cf. Figure 6). Larger distances result in a faster scrolling speed. Based on some initial tests, we assumed that this approach is sufficient to offer a flicking-like experience to the user when skimming the video.

In order to be consistent with the interfaces we introduce below, we subsequently call this approach ***dynamic flicking***.

3.2 Dynamic Panning

Basic idea. This interface design is based on the idea of elastic interfaces. An elastic slider is not manipulated directly but instead, the slider's thumb is pulled along the timeline with a virtual rubber band which is spanned between the thumb and the mouse pointer (or a pen's tip). The speed at which the thumb follows the pointer's movements depends on the tension on the rubber band, i.e. it increases if the distance between thumb and pointer gets larger, as illustrated in Figure 7. With this, users can browse a video at different speeds (by manipulating this distance, i.e. spanning or loosening the rubber band) and in different directions (by pulling the thumb with the rubber band to the left and right, respectively).

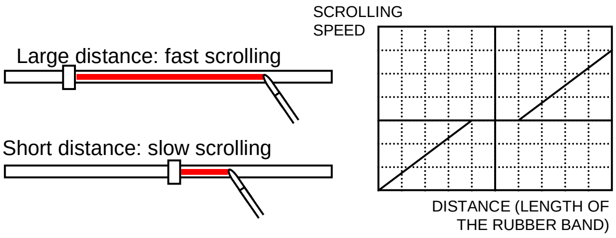

Figure 7. Basic idea of an elastic slider.

Elastic panning for video browsing. Elastic interfaces have initially been introduced by Masui et al. [7] for browsing and navigation in static data, such as text. In our own work [8], we applied the concept of an elastic slider to video browsing on a desktop PC and extended it to the notion of elastic panning. Elastic panning works similar to an elastic slider but does not require any widgets or GUIs. Instead, a user clicks anywhere on the screen (i.e. into the video). Scrolling along the timeline is done by moving the pen or mouse to the left and right (cf. Figure 8). Vertical pointer movements are ignored. This approach has several advantages over the traditional slider integration. For example, interface as well as actual content of the video always stay in the same focus of the user – a characteristic which has proven to be critical in actual usage (for further details we refer to [8]).

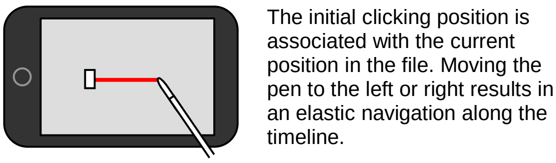

Figure 8. Elastic panning.

Small screen issues. Especially for small devices such as PDAs operated with pen-based interaction, elastic panning should be preferred over elastic sliders. First, it does not require users to target very small icons – a task which is not easy on the small screen of a PDA. Second, the rubber band interaction requires some space at the borders of the timeline in order to be able to move the thumb towards the end or beginning of a file during forward and backward navigation, respectively (cf. Figure 9). This is usually not a problem on desktop PCs, since mouse events can be processed even when the pointer moves out of the window. However, it is impossible to do this in full screen mode using an input device like a pen which manipulates the interface directly.

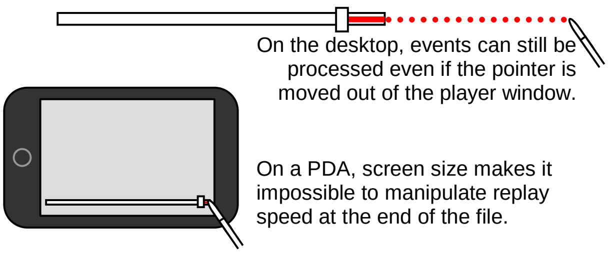

Figure 9. Behavior at the timeline's borders for elastic interfaces.

Variations and actual implementation. The basic realization of elastic panning on the PDA is in general comparable to its counterpart on desktop PCs. However, in the implementation, we had to adapt some of the parameters, for example the mapping function which maps the length of the rubber band to an appropriate scrolling speed (cf. Figure 7). In addition, there are two options for the implementation of the virtual timeline along which the slider's thumb is dragged on the screen, as illustrated in Figure 10: The resolution of the scrollbar can either be fixed or be set depending on the distance of the initial click position and the screen border. The latter case offers the advantage that you can always reach the end of the file. Therefore, it should be preferred on large screens. In the first case, we might need to reposition the pen in order to be able to reach a file's end. However, it gives us more options to modify smaller scrolling speeds which is why we decided to implement this version on the PDA.

In the following we will refer to this approach as ***dynamic panning***.

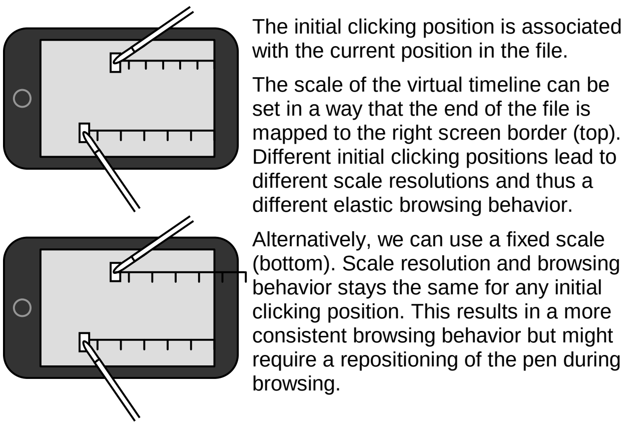

Figure 10. Scale of the virtual timeline.

3.3 Constant Flicking and Constant Panning

Both dynamic flicking and dynamic panning have a physical analogy. For dynamic panning, the rubber band metaphor is used to explain the movements of the slider's thumb and thus, the scrolling behavior of the file. Dynamic flicking can be seen as pushing the video or the slider's thumb along the timeline (depending on the actual implementation, cf. Figure 5 and "Panning vs. pushing" above). Slowing down of the document's movements can be interpreted as frictional loss. In the following, we discuss two further interaction designs which can be interpreted as variations of dynamic flicking and dynamic panning where this physical behavior is turned off. Consequently, we will refer to them as ***constant flicking*** and ***constant panning***, respectively.

Constant flicking. If we turn off the physical behavior of flicking, i.e. the frictional loss effect, scrolling does not slow down but constantly stays at the same level. Because of the problems with event processing mentioned in Subsection 3.1, we only consider left and right movements of the pen but do not take into account the momentum or speed of the actual flicking. For example, moving the pen from left to right results in a discrete increase of scrolling speed. Moving it to the left decreases scrolling speed for the same amount, independently of the length or momentum of the flicking. Scrolling stops at a single pen tip on the screen. This behavior is illustrated in Figure 11.

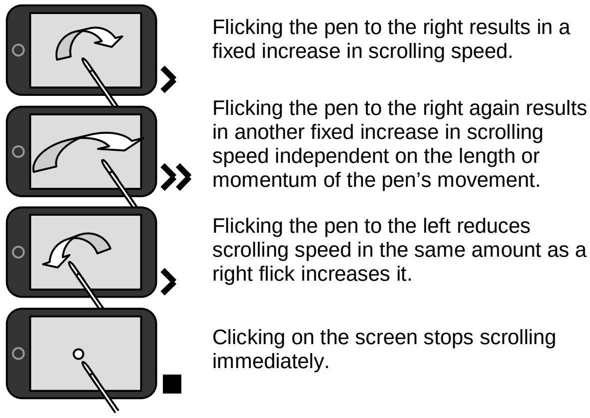

Figure 11. Constant flicking.

Constant panning. If we ignore the rubber band effect in our implementation of elastic panning, we can still increase scrolling speed by moving the pointer or pen away from the initial clicking position, but in contrast to elastic panning, the thumb does not follow and thus scrolling speed stays at a constant level as long as the user does not modify the distance between the pen and the initial clicking position. This behavior is similar to the so called auto panning technique used by some programs to scroll large text documents. It's realization for video browsing is illustrated in Figure 12.

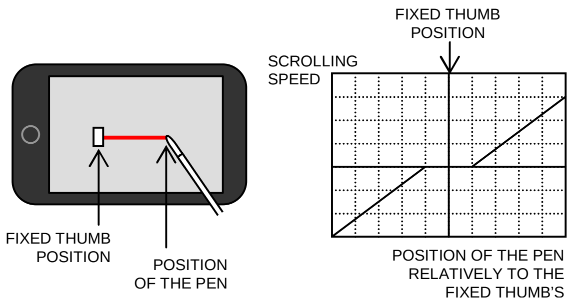

Figure12. Constant panning (auto panning)

3.4 General Implementation Issues

All four interface designs described in the preceding subsections, i.e. dynamic flicking, constant flicking, dynamic panning, and constant panning, have been implemented on a Dell Axim™ X51v PDA. This PDA features an Intel XScal, PXA 270, 624 MHz processor, 64 MB SDRAM, 256 MB FlashROM, and an Intel 2700g co-processor for hardware-side video decoding. It has a touch sensitive 3.7"display with a resolution of 640x480 pixels. Our implementations of the interfaces were made on top of TCPMP (The Core Pocket Media Player) – a high-performance open source video player. Implementation was done in C++ on Microsoft's Windows Mobile 5 platform.

In all four implementations, audio feedback is turned off during scrolling. For elastic and constant panning, the player switches back to normal playback or pause mode (depending on its status before the user started scrolling) as soon as the user lifts the pen from the screen. For constant flicking, the file keeps scrolling at a constant speed until the user clicks on the screen. Afterwards, the player goes back into its previous state, i.e. normal playback or paused. For elastic flicking, it switches back to normal playback or pause mode, respectively, as soon as the scrolling has come to a complete stop. This results in an awkward transition of the then paused player to immediate playback when the player was in play mode. Fortunately, this behavior did not have a negative effect on the evaluation described below, since most users navigated through the file while being in pause mode anyhow. However, for future versions, this case should be treated differently, i.e. if in play mode, the file should not slow down completely, but scrolling should smoothly lead into normal playback.

Figure 13 shows the actual implementation of dynamic panning. Constant panning was implemented with a similar visualization but the thumb stayed fixed during scrolling and did not follow the pen. No additional visualization was presented on the screen during dynamic and constant flicking but scrolling behavior only became apparent by the moving video images.

Figure 13. Dynamic panning implementation (screen mockup and actual implementation on the PDA).

4. USER STUDIES AND EVALUATION

Obviously, there are various ways to evaluate the usability of a system or interface design. One common approach which is often used in earlier design stages is heuristic evaluation [9]. Heuristic evaluations have the advantage that they are very effective, i.e. many usability problems can be identified by a relatively small number of evaluators. Laboratory studies involve a larger number of users and are usually done in later phases of the development process. Here, usability problems are identified by data collection in a controlled environment. For mobile systems, such a controlled experiment can also involve, for example, navigation in physical space or some division of attention. However, Kjeldskov and Stage [10] identified that seating test subjects on a table supports identification of significantly more usability problems. Nevertheless, they also confirmed the added value of usability studies in a mobile environment.

In order to evaluate the usability of the four interface designs discussed in the last section, we set up the following two experiments. First, we describe a heuristic evaluation with all four designs (Subsection 4.1). Based on the outcomes, we present revised versions of the two most promising approaches (Subsection 4.2). These revised implementations are evaluated in a laboratory study where some test users have to solve typical interaction tasks with the two revised interface designs (Subsection 4.3). User interface evaluation in mobile scenarios is part of our future research agenda (cf. Section 5).

4.1 Heuristic Evaluation

Setup. The recommended number of evaluators for a heuristic evaluation is three to four user interface experts [9]. Therefore, we decided to use four people (three male, one female) for our initial usability test. All of them were colleagues who have experience in user interface development but were not involved in this project. For the test, we used a within groups design, i.e. each user evaluated each implementation of the four interface designs presented in the preceding section. For each of the interfaces a different news show recording was used. Each file was about 20 minutes long. Associations between file and interface as well as the order in which the designs were presented to the users were shuffled among the participants. Evaluation took place at a table in a room at our department. Users normally hold the device in one hand and operated it using the associated pen.

Procedure and data collection. The evaluators had to perform three different tasks and provide feedback about the usability of each of the interfaces. Feedback was collected based on a list of usability principles. The original list provided by Nielsen [11] was taken into account but had to be modified in order to fit our special needs and conditions. In addition, a think aloud protocol was created, i.e. users were asked to comment on what they are doing and what problems and issues they are facing while performing the tasks and evaluating the designs. Finally, they had to give a rating for each design. Ratings for each interface were given immediately after its evaluation but the participants were allowed to do a final modification of their ratings at the end in order to guarantee a comparative assessment.

Tasks. At the beginning, the device was handed over to the users and they were only told that they can browse the video by moving the pen horizontally over the screen. No further information was given to them. Hence, the *first task* was to figure out the functionality of the interface. Details of the implementation which the participants were not able to identify themselves were explained to them afterwards. The purpose of the *second task* was to simulate a browsing situation in which a user wants to get a quick overview of the content of a file. To do this, the participants were asked to identify the first four topics discussed in the respective news show. The *third task* was to perform a targeted search in the file. Hence, the participants were asked to find a particular scene within the video, i.e. a map of the country which shows the temperatures within the weather forecast.

General feedback for all sliders was quite positive. Therefore, we restrict our discussion in the following mainly to the negative issues and usability problems which have been identified.

Dynamic panning. Overall, feedback for this interface design was very positive. Users were able to easily figure out its functionality and could handle it quite well. One evaluator mentioned that one might need a little longer to get adjusted to it compared to the other designs. Nevertheless, he said that it should not be much of a problem to be able handle it quite quickly. The main problem that was identified with this interface is that it is not easy to cover larger distances within a file and that it takes rather long to reach, for example, the rear part without being forced to constantly having to lift the pen and restart from one side of the screen. In addition, it was not properly visualized in our implementation when the end or beginning of a file was reached during forward and backward scrolling, respectively. This sometimes caused some confusion.

Constant panning. Similar to the elastic version, all users were able to handle this interface design quite easily. However, during scrolling two users often missed the target and had to go back. This resulted in a small but noticeable oscillating navigation around the desired position. Comments included suggestions for a different parameter setting, i.e. a higher maximum replay speed. Despite its different behavior, users generally said its usefulness and usability is comparable with its dynamic counterpart.

Dynamic flicking. None of the evaluators had any experience with the iPhone^{TM} at the time of the evaluation. Only one user had seen a video commercial about it and therefore was roughly familiar with this kind of navigation. Despite the missing experience, all users were able to learn and handle the operation of this interface quite easily. First impressions have been very positive and operating it seemed quite entertaining (e.g. one user immediately said "That's just cool!"). However, when performing the second and third task, most users confirmed our initial concern. They mentioned that it is quite easy to loose the feeling for the speed and thus get disoriented within the file. Frequent scene changes in the video also contributed to this effect. One user intensively tested the relation between the pen movements over the screen and the actual speedup of the file. His conclusion was that it would be much more intuitive and useful to rather do the speedup depending on the momentum of the pen than purely based on the distance. Overshooting the target and the resulting oscillation effect was discovered here as well but to a much lesser degree.

Constant flicking. Again, users were generally able to handle this design quite well. However, smaller problems have been observed when they tried to figure out the interface's functionality. One user wrongly assumed that the length of the input has an influence on how much replay speed is increased. Another one was initially not aware that he has to click on the screen in order to stop scrolling. In addition to the oscillation effect, which has been observed here as well, two users often accidentally stopped scrolling while making a very short flicking gesture on the screen. One user mentioned that having a visualization of the actual scrolling speed during scrolling might improve usability.

General observations and comparative ratings. Overall, all interface designs appeared to be easy to understand, quickly to learn, and useful when performing actual browsing tasks. However, the constant flicking approach caused some problems during operation. This also becomes apparent by the comparative rating which is summarized in Table 1. Although only four users participated in this heuristic study, the ratings also reflect that there are different preferences and perceptions by different users. In addition, we also observed that even with the same interface, the evaluators used different strategies to solve the browsing tasks (e.g. slow navigation through the whole video vs. quickly skimming it to get a high level overview followed by a finer navigation within the most promising areas of the file).

Table 1. Comparative ratings of the interface designs with grades according to the German rating system (i.e. 1 = very good, 2 = good, 3 = satisfactory, 4 = sufficient, 5 = insufficient, 6 = unsatisfactory). Similarly to the German school ratings, steps of the size 0.25 were allowed to do a more detailed specification.

Interface	User 1	User 2	User 3	User 4	Avg.
Dynamic panning	1.75	2	2	3	2.19
Constant panning	2	1.75	2.5	2	2.06
Dynamic flicking	1.5	1.5	3.25	1	1.81
Constant flicking	2	2	3.25	3	2.56

Based on the feedback from this study, we decided to pick the two most promising approaches from the four designs, revise the implementation in order to deal with the identified usability problems, and run a comparative usability study in a laboratory setting with several users. With one exception, dynamic flicking got the highest ratings and therefore was a natural choice as one of the two interfaces for this evaluation. For constant flicking, we believe that we can modify the implementation in a way that avoids most of the identified problems (e.g. accidental stops because of short pen movements during flicking). Nevertheless, we decided not to continue to pursue this approach because it appeared to be the least promising one. Based on the ratings and the evaluators' comments, both dynamic and constant panning appeared to be equally suited candidates for the second interface. Because of its similarity to dynamic flicking, we decided to use dynamic panning for the final evaluation.

4.2 Revised Interface Designs

Although both interface designs appeared to work quite well, some usability problems have been discovered in the heuristic study. Therefore, we implemented revised versions which deal with the identified problems.

Dynamic Flicking. Feedback from the evaluators in the heuristic study suggests that dynamic flicking might be the most entertaining one of all the tested interface designs. However, the feeling to loose control over the speed when flicking the pen over the screen turned out to result in a significant usability problem. We concluded that there are two main reasons for this.

First, there is no indication about the actual scrolling speed and the visual feedback from the video is too irregular and therefore rather adds to this problem than actually solving it. In order to deal with this issue, we decided to illustrate the replay speed at the top of the screen during scrolling. As soon as the user clicks on the screen, a bar appears at the top of the display which illustrates all possible scrolling speeds. Once the user starts flicking the pen over the screen, the actual speed level is illustrated by a lighter bar which is displayed within the speed bar. The speed level is constantly updated. The bar disappears as soon as scrolling stops and the player switches back into its previous mode (pause or normal playback, respectively). Figure 14 illustrates the final implementation.

The second problem was related to the way in which we implemented the actual flicking behavior. As said in Section 3.1, we initially assumed that just using the distance covered by the pen instead of its actual momentum is sufficient to simulate a reasonable flicking behavior. Apparently, this assumption was wrong. Hence, we re-evaluated the event processing and came up with a solution that allowed us to implement a flicking behavior which is much more comparable to the one realized on the iPhone™ for text browsing. By modifying the event processing and filtering outliers, scrolling speed now directly depends on the speed at which a user moves the pen over the screen, instead of just on the actual distance covered by the pen.

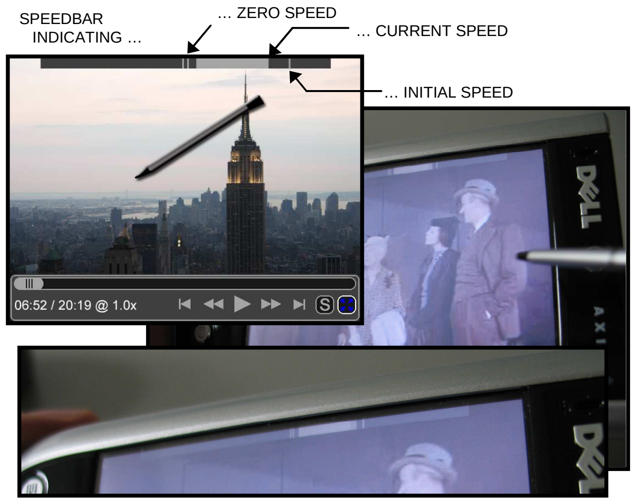

Figure 14. Redesign of the dynamic flicking interface.

Dynamic Panning. The most critical problem mentioned in relation to dynamic panning was that it took too long to skim larger distances in the file. Therefore, we experimented with different parameter settings for the mapping of the distance between the pen and the slider thumb (i.e. the length of the rubber band) to the associated scrolling speed. Increasing the maximum speed level three times as much as in the original implementation proved to work quite well. However, having a larger variety of different speed levels at lower scrolling speeds seems important as well. Therefore, we replaced the linear distance-to-speed mapping function (cf. Figure 7) with a nonlinear version that fulfills these requirements, i.e. provides a maximum scrolling speed which is three times as fast while still offering a large variety of different speed levels for slower scrolling speeds (i.e. shorter rubber band lengths).

Another issue mentioned in the evaluation was an improper visualization when the user reached the end or beginning of the video during forward and backward scrolling, respectively. We were able to eliminate this problem with a simple change in the visualization of the interface.

4.3 Comparative User Study

We showed the two implementations of the re-designed interfaces to the four evaluators from the heuristic evaluation and all of them agreed that the changes significantly increased usability. Therefore, we used them for our comparative user study.

Participants. As said before, heuristic evaluations are normally used during the design process to identify usability problems. The goal of the laboratory study that we set up with the two resulting interface designs was therefore to compare the final designs with each other rather than to identify remaining usability problems. In contrast to heuristic evaluations, where three to five users are normally enough to identify most of the usability problems, such a comparative study requires much more participants. For example, 24 users is a good number if you want to collect statistically reliable data.

In order to get a representative cross section of potential users, our plan was to get about 30% of users at the ages of 14-20 years, 50% of users at the ages 21-30 years, and 20% at the ages of 31-50 years. Unfortunately, not that many people were available for our study in the later group. In the actual study we had 8, 14, and 2 users in the age groups 14-20, 21-30, and 31-50, respectively. The male-female distribution in each of these groups was 6:2, 12:2, and 0:2, respectively. In the final evaluation, we were not able to discover any differences neither among the different age groups nor among female and male participants. Only nine users had pre-experience with pen based devices but again, no differences among them and the other participants could be observed in the evaluation with one single exception, which we discuss below.

Tasks. In order to be able to compare the performance and usability of the two designs, participants were asked to solve four tasks with each of the interfaces. Similar to the heuristic evaluation, the first task was to make you familiar with the slider and practice its usage. The remaining tasks were also quite similar to the ones used in the heuristic evaluation. However, since the goal of this study was not to identify usability problems but to compare the actual usage of the two approaches, they were presented to the users in a specific context:

Context for task 2: Assume you are at the station waiting for the bus. On your PDA you have the latest news show recording and you want to quickly browse it in order to see if there is any message that is interesting to you.

Actual task: What are the topics of the first four news messages?

Context for task 3: Assume you are on the bus now and you want to have a look at the second new message since it appears to be the most interesting for your.

Actual task: Navigate to the beginning of this message.

Context for task 4: Assume you almost reached your destination but before leaving the bus you want to have a quick look at tomorrow's weather forecast.

Actual task: Go to the weather forecast and find the map showing tomorrow's temperatures.

It should be noted, that for the third task only a rough positioning to the beginning of the scene was required. Hence, these tasks represent three different browsing situations, i.e. quick navigation to get an overview (task 2), navigation to roughly find a particular point of interest (task 3), and exact positioning (task 4).

Setup and procedure. Similarly to the heuristic evaluation, the study was done in a room with the participants holding the PDA in one hand and the pen in the other one. One person was sitting next to them, guiding them through the evaluation, and taking notes of interesting observations and comments given by the participants.

All users evaluated both interfaces (within group design). A different news show video was used for each test. Association between interface and video file as well as the order in which the interfaces were presented to the participants was equally balanced in order to avoid any influences on the results.

For the first task, users were given a maximum of 5 minutes. Time to solve each task was measured. Since the participants were asked to perform each task as fast as possible, no think aloud protocol technique was used in this study. Instead, users were interviewed and encouraged to give comments between tasks and at the end. Before a new task started, all files were put in a defined state (i.e. at the same position) in order to be able to compare the results with each other. At the end, users had to give a rating for each of the two interfaces.

Time to solve each task. Figure 15 illustrates the average time it took for the users to solve each of the tasks. Interestingly, no significant difference for any of the actual browsing tasks (i.e. tasks 2-4) could be observed. Based on the observations from the heuristic evaluation, we would have expected dynamic panning to perform better here, because of the loss of control observed with dynamic flicking. Apparently, it seems that this problem was limited in cause of the modifications in the design, i.e. the visualization of the actual scrolling speed and the different implementation of flicking. The only significant difference we could observe in terms of duration was related to the time people took to familiarize themselves with the interfaces (t-test, $t = -3.871$, $p = 0.00038$). On average, people spent more time with task 1 for dynamic flicking than they did for dynamic panning.

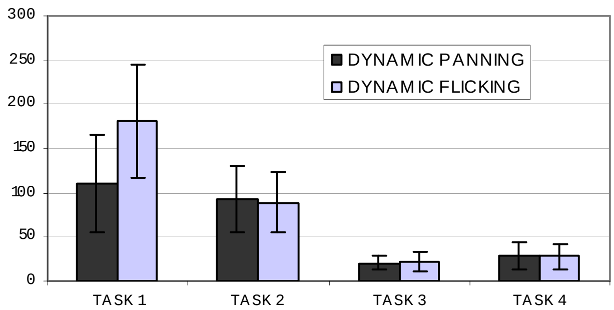

Figure 15. Average time (in seconds) to complete each task.

One possible reason for this is that dynamic flicking is more complex and thus requires more learning time, for example, in order to get a better feeling for the relation between scrolling speed and the way in which the pen is moved over the screen. However, in the heuristic evaluation we also had the subjective impression that this interface had the highest fun factor. Hence, people might just have spent more time here because they enjoyed playing with it. Unfortunately, we were not able to figure out which of the two options is more likely based on the observed data and the comments given by the users. However, the participants' ratings illustrated below give a hint that actually both reasons might apply but for different user groups.

Subjective user ratings. The average grading given by the users for the two interfaces was 1.917 for dynamic panning and 1.927 for dynamic flicking (on the same scale as used in Table 1). Obviously, the very small difference did not turn out to be statistically significant (t-test, $t = -0.062$, $p = 0.951$). However, there is a large difference in the standard deviation which was 0.303 for dynamic panning and 0.705 for dynamic flicking. This also becomes apparent when looking at the distribution of the grades which is illustrated in Figure 16.

Since both of the interfaces performed equally well in terms of time needed to solve the browsing tasks, we assume that the ratings generally reflect more personal preferences and dislikes. When asked about their ratings, most of the users who gave dynamic flicking a lower grade than dynamic panning argued with the loss of control problem that we mentioned above and which was already identified in the heuristic evaluation. Although our changes made to the interface seem to have limited this problem, it looks like they were not able to eliminate it completely. We were not able to identify a similar strong argument for their preference from the users who preferred dynamic flicking over dynamic panning. Being more fun might be a possible explanation but we were neither able to prove or disprove this hypothesis. We could not discover any correlation between the ratings and the time spent with task 1, i.e. the time needed to familiarize with the interface does not seem to have influenced the final ratings.

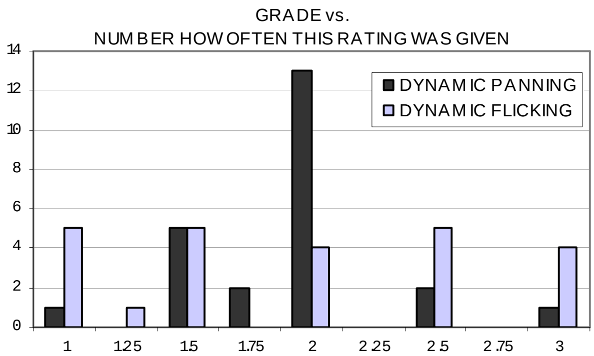

Figure 16. Comparative ratings of the interface designs with grades according to the German rating system (cf. Table 1).

Another important aspect which might actually have influenced these results might be that we made the mistake of asking people about their experience with the iPhone™ before the evaluation instead of afterwards. Although most people where unaware of the technique of flicking and all but one has never touched an iPhone™ at the time of this study, it was clear to most of them that the flicking technique was unique to this device. The fact that the iPhone™ (or the Apple/MAC vs. Microsoft/Windows debate) polarizes many people might indeed have had an influence on these subjective ratings.

Additional observations. As said before, for users who had some experience with mobile, pen-based devices, no differences could be observed; neither in the time measurements, nor in the users' ratings or comments. This is even true for the single user who owned an iPhone™ (which has just been released shortly before this study was done). However, he was the only one who mentioned that there is an alternative option to implement the flicking behavior (cf. "Panning vs. Pushing" in Section 3.1) and he preferred the other one because of his experience with the iPhone™. Nevertheless, he gave a higher rating for dynamic flicking than for dynamic panning.

5. CONCLUSION AND FUTURE WORK

The most important result from our evaluation is that both proposed interface designs worked very well. Users were able to understand their functionality and handle them easily without significant training time, explanation, or external support. The designs enabled the participants of the study to efficiently solve typical browsing tasks. In addition, users seem to like and enjoy using them. Both interfaces were rated with an average grade of 2 which is the second highest possible grade in the rating system used at German schools and universities (on a scale from 1 to 6 where you need at least a 4 in order to pass, cf. Table 1).

However, the detailed look at the ratings revealed another important observation. Individual preferences exist and are not necessarily related to the actual performance of a system. Although this statement seems intuitively clear, in our experience things such as user perception, experience, and fun are often issues that are underestimated in the academic world. However, our data demonstrates that this is a mistake and that the multimedia research community is well advised to take these aspects more seriously.

Another important issue resulting from the heuristic evaluation is that parameter optimization is very important. Again, this seems like a rather trivial statement, but it is clearly justified by our observation. Despite intensive testing by us, both initial implementations had usability problems which we have not been able to discover ourselves but which only became apparent to us as a result of the heuristic evaluation. Although they resulted in only small changes in the actual implementation, i.e. a different event handling for dynamic flicking and a different distance-to-speed mapping for dynamic panning, we are sure that the ratings and performance of both interfaces would have dropped significantly in the comparative study if we had not made these changes.

Given the high relevance of parameter optimization and usability evaluation, naturally, further user studies are one of our main interests for future work – for example, comparative evaluations with our previous work [3, 4] and tests in true mobile scenarios, where people use the interface designs in an actual mobile situation, (e.g., while riding the bus or when walking), as already mentioned in Section 2 and 4, respectively. In the experiments presented here, participants had to solve different problems that represent typical browsing task, i.e. navigation in a file at different granularity levels (fast scrolling to get an overview, and approximate as well as exact positioning). It would be very interesting to do a long-term study under real-world conditions where participants use the interfaces in their everyday life.

In addition, some people might prefer structured browsing approaches for some of the tasks used in our evaluation. For example, scene-based navigation or storyboards might be very useful for identifying the first topics of a news show (task 2). We already mentioned that we believe that a perfect interface should support both kinds of interaction because they complement each other very well. In addition, there are personal preferences which differ significantly among users. However, how to implement an interface that offers this kind of functionality is an open and difficult problem given the limited screen size and reduced possibilities for interaction. Similarly, visualization of metadata can be very helpful but at the same time can increase complexity, reduce clarity, and thus complicate ease of use. An interface design that adapts to different scenarios, contexts, and contents might be a promising direction.

Another major field of investigation should be the usage of special hardware and controls. Smartphones and PDAs usually have additional buttons or keypads which can be used for browsing in addition to the on-screen interaction realized in our interfaces. For example, using two keys for scene based navigation would be an easy but most likely very effective way to integrate structured browsing into our designs. Devices such as the iPhone^{TM} do not use additional buttons for interaction but feature other advanced approaches such as multi-finger touch which are also worth further exploration

6. REFERENCES

[1] O'Hara, K., Mitchell, A.S., Vorbau, A. (2007) Consuming video on mobile devices. *Proceedings of the SIGCHI conference on Human factors in computing systems*, (CHI 2007), San Jose, CA, USA, p. 857-86

[2] G. Ramos, R. Balakrishnan. (2003). Fluid interaction techniques for the control and annotation of digital video. *Proceedings of UIST 2003 – the ACM Symposium on User Interface Software and Technology*. p. 105-114

[3] W. Hürst, G. Götz, M. Welte (2007) Interactive video browsing on mobile devices. *Proceedings of ACM Multimedia 2007*, Augsburg, Germany

[4] W. Hürst, G. Götz (2008) Interface Designs for Pen-Based Mobile Video Browsing. *Proceedings of DIS 2008 – Designing Interactive Systems*, Cape Town, South Africa.

[5] D. Alikseyeu, P. Irani, A. Lucero, S. Subramanian (2008) Multi-flick: An evaluation of flick-based scrolling techniques for pen interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems*, (CHI 2008), Florence, Italy.

[6] A. Reetz, C. Gutwin, T. Stach, M. Nacenta, S. Subramanian (2006) Superflick: A natural end efficient technique for long-distance object placement on digital tables. *Proceedings of Graphics Interface '06*, p. 163-170

[7] T. Masui , K. Kashiwagi , G. R. Borden, IV (1995) Elastic graphical interfaces to precise data manipulation, *Conference companion on Human factors in computing systems*, p.143-144, May 07-11, 1995, Denver, Colorado, United States

[8] W. Hürst, G. Götz, T. Lauer (2004) New Methods for Visual Information Seeking Through Video Browsing. *Proceedings of the 8th International Conference on Information Visualisation*, London, UK, July 2004.

[9] J. Nielsen (2008) How to Conduct a Heuristic Evaluation *http://www.useit.com/papers/heuristic/heuristic_evaluation.html*

[10] J. Kjeldskov, J. Stage (2004) New Techniques for Usability Evaluation of Mobile Systems. *International Journal of Human-Computer Studies*, 60, 2004, 599--620.

[11] J. Nielsen (2008) Ten Usability Heuristics *http://www.useit.com/papers/heuristic/heuristic_list.html*