

# One-Handed 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

Philipp Merkle

Albert-Ludwigs University  
Institute of Computer Science  
Georges-Köhler-Allee, 79110 Freiburg, Germany

philippmerkle@gmx.de

## ABSTRACT

Various situations exist in which people want to use just one hand when interacting with their mobile phones or PDAs. However, most of the few advanced video browsing approaches for such devices only support two handed operation. In this paper, we present an interface design for one-handed mobile video browsing. Using a PDA with touch screen, users can interactively scroll through a list of thumbnails representing the video's content. Thumbnail sizes can be manipulated in order to provide easier access to relevant information. Two evaluations are presented which motivate the design and verify the usability of our solution.

## 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

Video, mobile video, video browsing, interfaces, PDA.

## 1. MOTIVATION

Advanced browsing techniques that enable users to interactively skim a file at different granularities are very important for the usability of video in a mobile context. For example, assume you recorded an interesting documentary. In a discussion with your friends about this topic on the next day, you want to show them one important statement from that show. For this, you need to be able to quickly find the relevant position within the file. Because of the small screen size, only one or two people can watch it at a time. Hence, your friends, who are passing the device around, also need to be able to reset replay to the beginning of this relevant position. A recent study [1] about how people experience mobile video confirmed such intuitive arguments. Many usage scenarios could have been observed which indicate that users of

mobile video would indeed profit from having better possibilities for navigation on such devices. However, common handheld devices hardly support any kind of advanced video browsing technique. Among other reasons (such as a lack of awareness for this necessity [2] and limited processing power) user interface design seems to be one of the most critical issues. Small screen sizes and limited input capabilities prevent us from implementing successful browsing techniques from desktop PCs on smartphones or PDAs.

For this reason, we started developing interfaces to interactively browse and navigate video data on mobile devices, in our case a pen-based PDA. Our work includes the MobileZoomSlider [3] which features several virtual timelines for navigation in a video at different granularity levels, and the ScrollWheel [4] which enables users to scroll through a video making circular movements with the pen on the screen. In our latest work [5], we experimented with elastic panning and flicking, two techniques in which the user dynamically manipulates scrolling speed – in the first case by moving the pen closer or further away from a moving slider thumb, in the second one by flicking the pen over the screen, similarly to a technique used to browse long text lists on the iPhone™. In different evaluations, we proved the usefulness and usability of the proposed designs. Although all of them offer very advanced and useful functionality for video browsing, they also share one common disadvantage. For each design, two hands are needed: One to hold the device, the other to do the actual interaction by moving the pen (or a finger) over the screen.

However, in mobile situations, people often prefer to use just one hand for operating handheld devices. For example, it is not uncommon to experience mobile video for entertainment purposes while eating and drinking when sitting in a coffee place. In addition, there are lots of situations where the second hand might not be free at all, for example, when holding on to a handrail while standing in a bus or when carrying a briefcase while waiting at a station. In [6], Karlson et al. describe a user study about the operation of mobile phones and PDAs with your thumb. Among other things, it also indicated that people often prefer to operate mobile devices with one hand but that current user interfaces normally do not support this kind of interaction.

In this paper, we present a new design for one-handed video browsing on a PDA with a touch sensitive screen. Instead of the pen, which is the common input device for PDAs, users can browse a video's content by making different thumb movements at different positions on the screen while holding the device with the same hand. Based on an initial study which evaluates the possibilities for one-handed thumb-based interaction (Section 2), we introduce the interface design (Section 3). The implementation (Section 4) is evaluated in a first user study in a static and mobile scenario (Section 5).

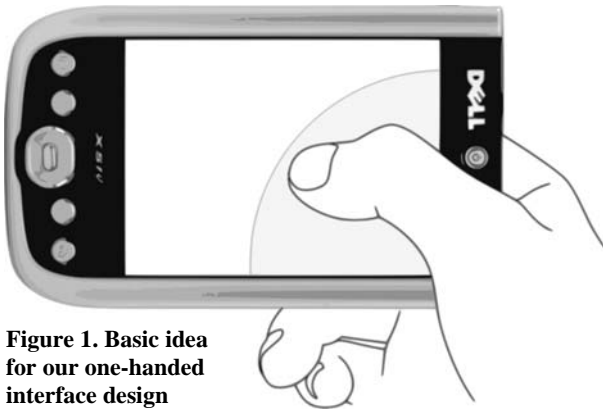
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

uxTV'08, October 22–24, 2008, Silicon Valley, California, USA.

Copyright 2008 ACM 978-1-60558-100-2/08/10...\$5.00.

## 2. THUMB-BASED INTERACTION

Operating mobile, handheld devices with one hand basically means that you hold the device in your hand in a way that the thumb can reach the device's surface in order to press buttons or interact with a touch sensitive screen. Obviously, using your thumb to, for example, hit buttons on a touch screen is different from using a pen. First of all, thumbs are usually larger than the small tip of a pen, a fact which should be considered, for example, when designing the size of buttons. Parhi et al. [7] evaluated the target size for thumb-based interaction and present some recommendations for optimum target sizes. Secondly, not every area of the screen can be reached easily with a thumb when the same hand is used to hold the device. Karlson et al. [6] present a user study indicating that thumb movements are easier to do in the center of the screen than in border areas. In [8], Karlson et al. developed *AppLens* and *LaunchTile*, two interfaces for one-handed operation of PDAs and smartphones. In *AppLens*, gesture-based interaction is used to move some sort of cursor over the screen in order to deal with the problem that some parts of it are not reachable with the thumb. In *LaunchTile*, control elements on the screen are adapted to the size of the thumb.

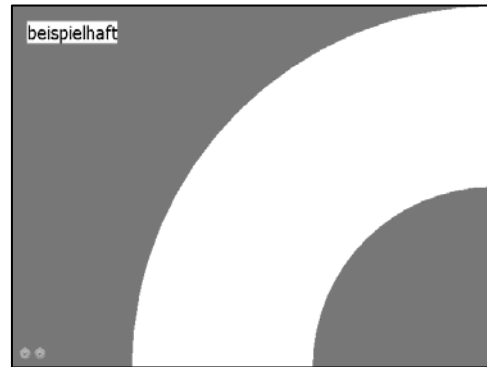


**Figure 1. Basic idea for our one-handed interface design**

**Basic design idea.** Based on some initial tests with different people holding a PDA in one hand, we came up with the idea for a thumb-operated interface design for video browsing which is illustrated in Figure 1. If the PDA is held in landscape mode (which is a common presentation mode for mobile video in order to comply with the aspect ratio of normal video recordings), the thumb can be used to reach a certain area of the screen in order to interact with the application. However, not every part of the screen can be accessed. For example, it is quite obvious that most people will not be able to reach the upper left corner of the screen at all. Based on the way you hold the device, the following three interaction modes appeared to be useful for our purpose: 1. circular thumb movements over the screen, 2. horizontal thumb movements along the lower border of the screen, and 3. vertical thumb movements at the right screen border. The first one seems to be the most natural one because of the normal motion range of a thumb. This statement is confirmed by Karlson et al.'s study [6] which identified that thumb movements on a PDA in the northeast-southwest direction can be done easier and faster than in the northwest-southeast direction. Strict horizontal and vertical thumb movements are much more uncommon which is why we decided not to use them on the screen but only at its lower and right border. On the PDA which we used for our implementation,

the device's case around the embedded screen is actually a little elevated. This kind of physical border around the display should make it much easier for a user to glide their thumb over the screen in the respective way.

**Initial experiment.** People are different and the way in which a few people move their thumb while holding a device can hardly be seen as representative for the heterogeneous range of users we expect for such an interface. Therefore, we set up an initial experiment in order to evaluate what kind of thumb movements people feel comfortable with and what can easily be done when holding the device in the palm of your hand.



**Figure 2. Image on the PDA during the initial experiment.**

**Setup.** A PDA showing the image illustrated in Figure 2 was given to the test subjects. When holding the device in one hand (cf. Figure 1), users were able to scroll through a list of words by moving their thumb along the lower border of the screen, the right border of the screen, and in a circular movement, respectively. The visualization of the white quarter of a circle in the image was supposed to give users an idea of the motion that was requested, but they were informed that it is only for orientation and that they are not required to actually move their thumb within the white area. In order to avoid that the participants adapt their thumb movements to the white circular arc shown in the image, we intentionally made it rather wide. The current entry of the list was shown in the top left corner of the screen. The intention of this text was also to force users to concentrate on the text and thus track their attention away from their actual thumb movements.

**Tasks and participants.** Users were asked to perform four interaction tasks, i.e. scroll through the list of text entries 1) by making circular-like thumb movements over the screen, 2) by making horizontal thumb movements at the lower bottom of the screen, 3) by making vertical thumb movements at the right border of the screen, and 4) by making a combination of all three different interactions in random order. The test was done with 18 subjects, 17 male and 1 female, at ages 20 (1 subject), 21 (1), 22 (5), 23 (5), 24 (1), 25 (4), 48 (1). Four participants had experience with pen-based input on PDAs. One was an experienced user of pen-based operation with TabletPCs.

**Results.** All interactions have been recorded for later analysis. The interactions done by the participants for each of the four tasks are shown in the appendix. Two participants (no. 8 and 14, cf. Appendix) had general problems holding the device. All other subjects generally had no problem performing the circular thumb movements required in task 1. Most participants described this

kind of interaction as intuitive and natural. Two subjects noted that moving the thumb upwards can be done easier than moving it downwards. The horizontal movements required for task 2 appeared to be quite critical: Although most participants were able to perform this kind of interaction successfully, many had the impression that they might drop the device if their thumb is moved too far to the left. For the vertical thumb movements in task 3, only half of the participants made a similar comment, whereas the other half did not have any problem with this kind of interaction.

When looking at the log data visualized in the appendix, it can be noticed that almost all subjects used the whole right screen border for vertical movements, but most of them only made rather short horizontal movements at the lower screen border. This is certainly due to the fact that the width of the screen is larger than its height, but also confirms the users' comments about the unstable feeling when the thumb is moved too far to the left (cf. above). A significant amount of users (especially subject 1, 7, 8, 9, 14, cf. Appendix) only made very short horizontal thumb movements most likely due to a fear of dropping the device. For vertical movements, all but one user (subject 9) used the whole range.

**Consequences.** Based on the observations from this experiment, we draw the following conclusions for the interface design:

- Variations in the circular movements (task 1 and 4) were much lesser than we expected. Also, almost all participants considered this kind of interaction as easy and natural. Hence, we can use it for detailed and extensive interactions.
- All but one participant took advantage of the whole interaction range for vertical thumb movements, but many mentioned that it is harder to handle than circular thumb movements. Hence, this interaction seems to be less intuitive and comfortable, but still useful and manageable. We conclude that an associated functionality should be less intensive and sensitive.
- Horizontal thumb movements are much harder to perform when holding the device in the same hand, which is why only occasionally used functionality should be mapped to this kind of interaction. In addition, the interaction range should be rather limited, i.e. users should not be required to move their thumb horizontally over a larger range on the display.
- Due to the different motion ranges, no association between absolute positions on the screen and the values we want to manipulate should be made. For example, we should not associate the smallest and largest scrolling speed with one corner of the screen but instead, we can associate the initial clicking position with a default value (e.g. current replay speed) and use up and down thumb movements at the right screen border to decrease and increase this value, respectively.
- Two users did not feel comfortable holding and operating the device with one hand at all. In addition, few participants noted that they agree that there are a lot of situations where one-handed interaction is important, but if possible, they would always use two hands. Hence, we conclude that the final interface design should not only be optimized for one handed interaction but enable users to use two hands as well.

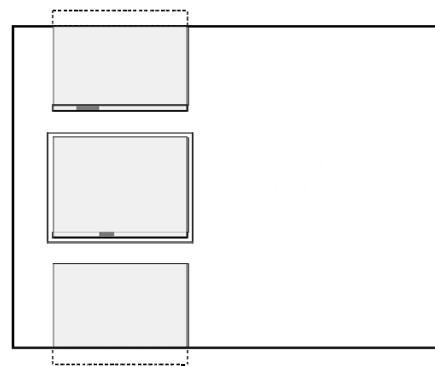
In the following, we present a user interface design which complies with the requirements specified above.

### 3. INTERFACE DESIGN

There are different options for browsing and navigating a video. In our previous work [3,4,5], we created interfaces enabling users to interactively manipulate the current position in the file along the timeline of a video. For the one handed design, we decided to use scene-based thumbnails to support user browsing. From each scene, a representative thumbnail is selected and shown to the user. In interfaces for video browsing on desktop PCs, such thumbnails are usually compiled to a storyboard representing the structure and content of a video in a static format (cf., for example, [9]). By looking at these static representations, users can get an idea of the content of the associated scenes and directly access them by, for example, clicking at the respective thumbnail.

**Initial design decisions.** Storyboards have been proven to be very useful for video browsing. However, they lack from a detail vs. overview tradeoff: representing too many thumbnails results in a loss of detail (because of the reduced size of individual thumbnails) whereas representing fewer thumbnails results in a loss of overview (because less content from the video is visualized). This problem obviously becomes even more critical on PDAs because their small screen size generally prevents us from representing whole storyboards. Therefore, we made the following design decisions for our one-handed user interface:

- Only a few thumbnails are shown at a time: Initially, the one representing the current scene as well as a few of the preceding and following scenes. Users should be able to browse through the whole content of the video by manipulating which thumbnails are currently shown.
- Since the optimum thumbnail size depends on the content of a particular scene as well as the actual browsing task, there is no best, one-size-fits-all solution here. Hence, users should be able to manipulate thumbnail sizes.
- Because the interactions evaluated in the initial experiment should be used, thumbnail representations are placed on the left side of the screen in a way that the user's interactions do not interfere with the visualization of the thumbnails.

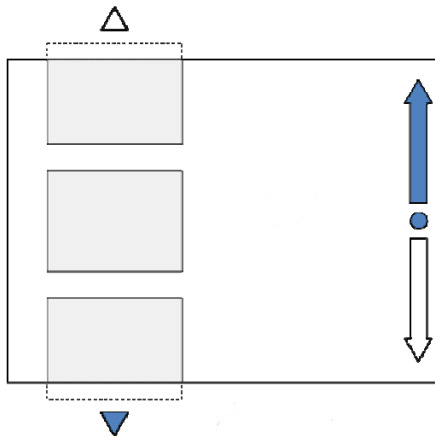


**Figure 3. Thumbnail representation on the PDA's screen.**

**Thumbnail visualization.** After experimenting with different placement options, we decided to represent the thumbnails as a linear list on the left side of the screen, as illustrated in Figure 3. This way, the thumbnail images are not blocked by any thumb movement. Each thumbnail also has a small bar at the bottom indicating the length and position of the associated scene relative

to the whole file. Since the thumb can not reach the images in order to activate an associated scene, we indicate the middle thumbnail as active using a small border. Making a double click (i.e. tipping anywhere on the screen twice within a short time slice) causes the player to jump to the associated scene in the video.

**Interaction concepts.** When scrolling through a list of images, there are generally two options: Manipulation of scrolling speed (i.e. how fast or slow the images are moved) and moving the currently visible part of the list, for example, by dragging the thumb of a scrollbar. Both interaction concepts can be useful. Speed manipulation is often better if you want to continuously browse a list – for example, when using fast forward to quickly get an idea of the whole content. When moving the currently visible part of the list, users can not sit back and just look at the moving content. Instead, they can interactively manipulate at which time (and therefore at which level of detail) the content is presented to them. Hence, this kind of interaction offers more flexibility but also requires more interaction. Since we believe that both kinds of interactions are equally important when browsing through the list of thumbnails in our interface, we decided that both of them should be supported.

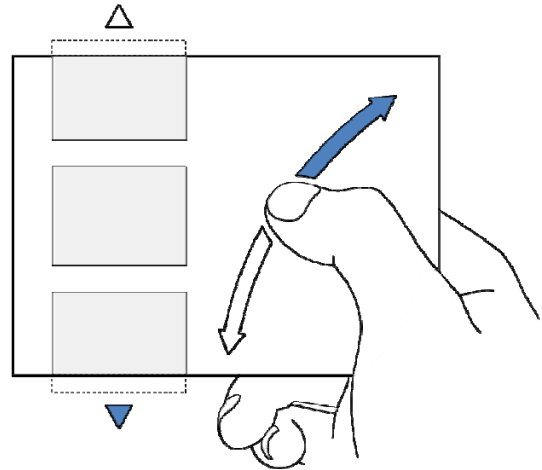


**Figure 4. Manipulation of scrolling speed.**

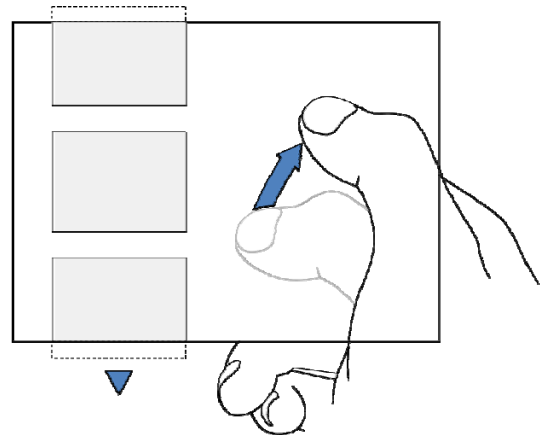
**Browsing thumbnails by speed manipulation.** When putting your thumb in an area close to the right screen border, this initial clicking position is associated with a zero speed. When moving the thumb downwards or upwards along the screen’s border, the thumbnails start scrolling in backward and forward direction, respectively, as illustrated in Figure 4. The association between up/down movements and forward/backward movement was motivated by a similar scrolling behavior when scrollbars are used to browse a text document. Scrolling speed increases the further the thumb is moved away from the initial clicking position, i.e. the initial reference point representing zero speed. Scrolling of the thumbnails stops as soon as the thumb is released. Double clicking anywhere on the screen starts replay at the scene which is represented by the middle thumbnail.

**Browsing thumbnails by position manipulation.** As said before, moving of the currently visible part of a long list of thumbnails, i.e. by manipulating the position of the list in relation to the screen usually requires much more interaction than modifying scrolling speed. Hence, we associate this functionality with circular thumb movements on the screen which – according to our

initial experiment – are more natural, intuitive, and easier to handle. Clicking anywhere into the center of the screen associates this initial clicking position with the current thumbnail representation. Moving the thumb up or down causes the thumbnails to move down or up, as illustrated in Figure 5. In contrast to speed manipulation at the right screen border, the thumbnails’ positions are controlled directly, i.e. they are moved at the distance that the thumb is moved over the screen.



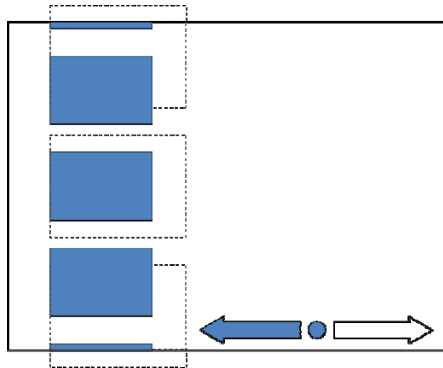
**Figure 5. Manipulation of the position of the currently visible part of the list of thumbnails.**



**Figure 6. Browsing by flicking.**

**Browsing thumbnails by flicking.** Modifying the currently visible thumbnails by making circular movements over the screen as illustrated above allows users to do a more exact thumbnail selection than with scrolling speed manipulation. However, the possible range for manipulation is limited. If users want scroll larger distances, they continuously have to reposition their thumb once the border of the screen is reached. In order to give users an opportunity to skip ahead for a larger part of the file, we added the functionality of flicking: If a user quickly flicks the thumb over the screen, the thumbnails start scrolling at a speed which is proportional to the momentum at which the thumb is flicked over the display (cf. Figure 6). Scrolling slows down after a while, again at a speed which depends on the momentum of the flick. Such an interaction is well known from Apple’s iPhone™, where

it is used to scroll large lists of text entries (e.g. songs in a music collection). The main advantage of flicking is that it complements normal position manipulation very well. Users can browse a certain range of thumbnails by moving their thumb over the screen in a circular movement. If they realize that the current area of the file is of no interest to them, they can quickly skip ahead by flicking their thumb with a similar, i.e. circular thumb movement. Although skimming through the thumbnails at a higher speed now, visual feedback is always provided. The further the thumbnails advance in the file, the more likely it is that another relevant area is reached, making it reasonable to automatically slow down scrolling speed.



**Figure 7. Manipulating thumbnail sizes.**

**Manipulating thumbnail sizes.** Since the optimum thumbnail size for browsing depends on the actual context, for example, the content of a video or the browsing task, users should be able to easily and quickly manipulate this value. However, it might not be necessary to provide every possible size. In addition, we do not expect users to change thumbnail sizes very often. Hence, horizontal thumb movements on the lower part of the screen seem to be a reasonable choice for realizing this kind of functionality. When clicking on the lower part of the screen, this initial clicking position is taken as a reference point representing the current thumbnail size. Moving the thumb to the left or right decreases and increases thumbnail size, respectively, as illustrated in Figure 7. Because many users seem to have difficulties moving their thumb horizontally without having a feeling of instability in holding the device, we decided to provide only four different thumbnail sizes: 100x75 pixels (overview), 200x150 pixels (default), 250x180 pixels, and 300x225 pixels (detailed view). These sizes are mapped to a certain range on the lower part of the screen at a position relative to the initial reference point.

**Discussion.** All of the design decisions discussed above comply with the requirements resulting from our initial experiment:

- All interactions require users to move their thumb in a particular area, but not to hit a specific position such as a slider thumb or button. They all require one click on the screen followed by a thumb movement in two opposite directions, thus adapting to the individual thumb sizes and motion ranges of different users.
- The length of the required movement for navigation is proportional to the common motion range identified in the initial experiment. For example, modifying the thumbnail size

only requires very limited movements because only few parameters can be selected. Hence, it is mapped to the lower screen border. Manipulation of the visible thumbnail range usually takes into account a much larger area and also requires a continuous change of parameters, which is why it is associated with the most intuitive and natural movements.

- The expected intensity of interactions is directly related to the intuitiveness and ease of the related interaction. For example, we expect that thumbnail sizes are not changed that often, thus mapping the least natural thumb movement to this kind of functionality. Speed and position manipulation are equally likely to be used, but the latter one requires more interaction, which is why circular movements are associated with it.
- The required sensitivity of the parameter manipulation is also in line with how easy and exact the thumb movements can be done. Position manipulation requires an exact, continuous change of parameters. Otherwise, the resulting feedback would become jerky and thus, irritating and difficult to follow. Speed manipulation also requires continuous parameter manipulation but usually it is not that important to do a very accurate parameter selection here. For example, hardly any users will recognize the difference between a scrolling speed of 1.4 and 1.45. Finally, horizontal thumb movements are the most difficult which is why they are mapped to only a few discrete parameters represented by larger ranges on the screen.

It should be noted that the proposed design can easily be operated with two hands as well, which is important for situations in which users actually want to use both hands or generally feel uncomfortable when using just one hand (cf. Section 2).

## 4. IMPLEMENTATION

The interface design introduced in the previous section has been implemented on a Dell Axim™ X51v PDA which runs Windows Mobile 5 and features a touch sensitive screen that is usually operated via pen input. We decided to use this kind of device because of its high performance and the flexible SDK which gave us enough control and power to do our implementation. The interface was realized on top of TCPMP (The Core Pocket Media Player), an open source video player which (among other characteristics) is also optimized for high quality and high performance replay on mobile devices. Figure 8 shows examples of our implementation on the actual device.

**Left- vs. right-handed operation.** In the above description of the interface design, we always assumed that people are holding the device in their right hand. Because the interactions are quite easy to do, we expect that right as well as left handed persons can use it this way. But also both of these user groups might want to switch between both hands every now and then and hold the device in their non dominant hand. Obviously the user interface should be adapted to this, i.e. the thumbnail representations and interaction areas should be mirrored for left-handed operation. Due to time restrictions, this functionality has not been implemented yet in the current version but is part of our plans for future work. The switch from left- to right-handed and vice versa could be recognized automatically, for example, based on the position and direction of the initial input.

**Figure 8. Different interactions with the implementation on a Dell Axim™ PDA: position (left), speed (right), and thumbnail size manipulation**



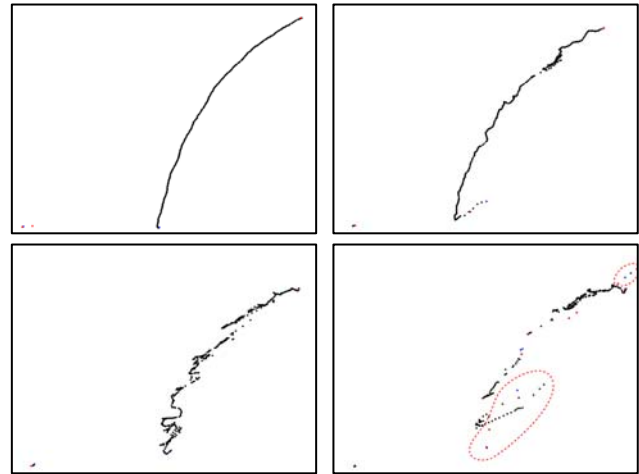
**Visualization.** In the final implementation (cf. Fig. 9), when clicking anywhere on the screen with your thumb, the video is paused, the image is blacked out a little, and the thumbnails representing different scenes are shown on the left side of the screen with the middle thumbnail representing the current scene. Small icons on the screen illustrate the different interaction areas and give a hint about the expected interaction, i.e. left/right, up/down, or circular.



**Figure 9. Visualization during browsing.**

**Interaction processing.** The implementation of thumb motion tracking turned out to be quite challenging since this PDA is normally operated with a pen. Hence, the touch technology used in this device is optimized for this kind of input and not for finger-based operation. Technically, when a pen touches the screen, the position of the pen's tip is returned to the system as an event containing the x/y position and a timestamp. Due to the small size of the pen's tip, this information is very accurate. If your finger moves over the screen, it never just touches one pixel but always covers a smaller area of pixels. The problem for the implementation is that *any* of the pixels covered by this area can be returned as an event. This turned out to be a critical issue when users are moving their thumb over the screen at a very low pressure. Figure 10 illustrates four examples where a pen movement and three thumb movements at different pressure levels have been recorded. The last image clearly shows some significant outliers in the recorded thumb movements which might be critical when scrolling through the thumbnails. In order to deal with these outliers and get a more robust estimation of the actual thumb movements, we developed an error resistant motion detection algorithm. Its basic idea is to divide the screen in different piles representing a larger input area. Based on some experiments, we identified that there are many outliers such as the ones illustrated in the example in Figure 10 if low pressure is

used, but also that the distance between two single outliers is usually rather large. Hence, we are able to use a threshold in order to cluster subsequent events and create a motion vector between different piles as soon as the number of events within one pile exceeds this threshold. Using a pile size of 16x16 pixels and a threshold of three single events turned out to produce an approximation for the thumb's movement which is robust enough for our purposes.



**Figure 10. Events recorded with pen input (top left) vs. thumb input with strong (top right), medium (bottom left) and weak pressure (bottom right, outliers marked by red dashes).**

**Segmentation.** Our interface assumes that a reasonable video segmentation exists. Alternatively, if no such information is available, we could just use periodic time frames, for example one frame every two minutes. Because this work is focusing on interface design, we excluded this problem here and used a manually created segmentation for the evaluation in the next section. Integrating a reliable automatic segmentation is part of our plans for future work. In the current implementation, once a file is loaded, the player takes a textual description which contains timestamps which in turn indicate the beginning of new scenes or segments. Each segment is then represented by one thumbnail.

## 5. EVALUATION

In order to evaluate the proposed design, we set up a user study which is based on a heuristic user interface evaluation. Heuristic evaluations are often used in early design stages where three to five interface experts evaluate a new user interface design based

on some common guidelines and rules. Because the technique of heuristic evaluation has originally been developed for desktop PC interfaces, we only used the related guidelines as a rough orientation but adapted them to the mobile scenario we are targeting. We also asked the participants to perform certain browsing tasks because in our experience with previous user interface evaluations we had the impression that performing such tasks helps participants in making better judgments. The tasks had to be solved in a mobile scenario. In [10], Kjeldskov and Stage verified, that evaluations done in a static setting are able to identify most of the critical design issues even for mobile interfaces. Nevertheless it can be useful to do an additional evaluation in a mobile context. Based on the approaches proposed in [10], our heuristic evaluation was therefore followed by a second part, where the users had to use the interface and solve browsing tasks while walking around.

**Setup of static experiment.** Because three to five interface experts are normally recommended for a reliable heuristic evaluation, we asked four evaluators to participate in this study. All of them were right-handed males at ages 23 (2 subjects), 27, and 28. Two of them had usage experience with PDAs, one with a smartphone, and one with a Nintendo DS. All of them have been involved in some sort of interface development and therefore qualify as reasonable candidates for a heuristic evaluation. The video was a recording of a popular German TV news show. It was segmented into 17 segments each of which represented a content-related unit. The average segment length was 1.5 minutes. Timing information about these segments was used by the player software to create the related thumbnails. After a short general introduction, each part of the interface was evaluated separately. Users were given some examples for situations in which the according functionality might be useful. A neutral observer noted problems which were identified when the participants used the respective functionality and wrote down comments they made. In addition, a list of questions which was motivated by the ones typically used in heuristic evaluations was used to interview the evaluators. After testing all different functions individually, they were asked to use them all together and give some final comments on the overall design and integration.

**Setup of mobile experiment.** Obviously, the usage of a device in a mobile context can be quite different from using it in a static environment. Interacting with a device while you are walking, riding a bus, or even just standing in a noisy environment certainly has an influence on the overall user experience. Kjeldskov and Stage [10] noted that a distraction of the user while operating a device can be helpful to identify problems in the layout of an interface. Motivated by their recommendations for the evaluation of interfaces in a mobile context, we set up an experiment where the participants from the heuristic evaluation had to use the device with our new interface design while they were walking. It should be noted that situations where people are watching video while they are walking certainly exist, but most likely are rather limited. The purpose for this experiment was therefore rather based on the idea of creating some kind of distraction instead of representing a really realistic usage scenario. For this second part of the experiment we used a different but similar video, i.e. a recording from the same news show on a different day. Users had to walk a given path: Out of the room in which the heuristic evaluation was done, to the end of the corridor, down the stairs to the subjacent floor, along the

corresponding corridor, up the stairs again at the other end of this corridor, and back to the evaluation room. In order to guarantee that they were only using one hand during the experiment, they were given a closed bottle of liquid which they had to carry in their left hand. Additional distraction was created by an observer who noted observations, comments from the participants, and told them which tasks they have to solve. Task 1 was to find the weather forecast and the map showing the temperature for the next day. Beginning from the start of the video, this task required a longer navigation to the end of the file in order to find the weather forecast and then an exact positioning in order to identify the correct temperature. Task 2 asked for the name of a company which was mentioned in one news clip. In order to solve this task, users had to either enlarge the thumbnails or start replay of the video. Task 3 asked for the content of the first news message, requiring users to go back to the beginning of the file. Task 4 asked for the most important topics of this news show. The purpose of this task was to get a rough overview of its general content.

**Observations and discussion.** Generally, all evaluators were able to handle the interface very well, confirmed the usefulness of the proposed design, and attested a high usability. However, certain negative aspects have been noted and interesting observations have been made which we summarize in the following.

For the navigation using circular movements in the center of the screen, one user noted that upward movements are easier to make than downward ones. We noted that he actually made different tracks in both cases – a behavior which we did not observe in the initial experiment (cf. Section 2). All four evaluators agreed with the scrolling direction, but three noted that some users might prefer it the other way around, i.e. that upwards thumb movements result in upwards movements of the thumbnails. One user was irritated that repeated flicking did not increase scrolling speed. In fact, we did not implement this behavior but are planning to do so in the future since it would be very useful for browsing longer distances. There was a little delay between the thumb movements on the screen and the resulting motion of the thumbnails because of the motion estimation we had to do in the implementation to cope with the outliers (cf. Section 4). However, none of the evaluators considered this to be critical. In fact, one of them even noted that it might actually be a useful feature in order to avoid accidental input which can happen more often in a mobile context.

When using the speed manipulation on the right side of the screen it became apparent that we have set the area covering this functionality too small. Two users actually had some small problems because of this. One participant noted this functionality as particular useful for scrolling longer distances because not much interaction is needed. Being able to modify the thumbnail size was generally considered to be very useful as well. One evaluator noted that the best size in a particular situation can also depend on the mobile context, for example, sitting at the bus station vs. standing in a moving bus between several people.

Generally, the evaluators confirmed that depending on the interaction, there are situations in which they feel that the device does fit less comfortably in the palm of your hand, but none of them had any problems operating it or the feeling that holding it became unstable when doing certain interactions. Interestingly, one participant noted that he feels horizontal movements are more

comfortable than vertical ones – a comment which does not conform to our observations in the initial experiment (cf. Section 2). None of the evaluators had any problems performing horizontal thumb movements or did consider them as uncomfortable. Although the limited number of participants does not allow us to draw a general conclusion here, we consider this as an evidence for the success of our design and implementation.

Considering the visualization, all evaluators made very positive comments about the overall look and feel and described the design as pleasant and intuitive. One exception was the icon used to illustrate the circular movements which did not represent the related functionality very well. One participant initially assumed that the functionality on the screen borders is only available on top of the corresponding icons, which did not expand across the whole screen. One user noted that it would be better to explicitly visualize the beginning and end of a file (i.e. the first and last thumbnail) instead of just relying on the position indicator placed below each thumbnail (cf. Figure 3 and 9). All users found it very positive that the visualization of the thumbnails was separated from the area where you interact with the device, thus limiting the risk of covering important content with your thumb. However, one participant noted that this is not true for the largest thumbnail size and gave this as a reason why he did not use them in the mobile evaluation. One user mentioned positively about the lack of buttons because first, this limits the chance to accidentally hit the wrong one. Second, using a double click to start replay can be done anywhere on the screen and does not require you to go back to a specific position.

In the mobile experiment, all users were able to solve the given tasks in an easy and straightforward way despite the significant amount of distraction being created. Table 1 illustrates which interaction and functionality were used for which task. Interestingly, one participant mentioned in the heuristic evaluation that he would not need speed manipulation in such an interface, but actually used it to solve the first task. Overall, it is quite satisfying to see that the users took advantage of different interaction styles in different situations. We interpret this observation as an indication for a good integration and the usefulness of the various functionalities offered by our interface.

**Table 1. Usage of different interactions in the mobile experiment (one \* indicates that this interaction was used by one person at least one time in this task).**

TASK	POSITION MANIPLATION	FLICKING	SPEED MANIPULATION	THUMBNAIL SIZE.
1	*		***	*
2	**		**	**
3	*	**	**	*
4	***		*	*

## 6. CONCLUSION AND FUTURE WORK

The work we presented in this paper was motivated by the argument that there are situations in which users want operate their mobile devices by just using one hand. Based on an initial experiment we identified comfortable and useful thumb motions for one-handed interaction with a touch sensitive screen and specified certain design restrictions. We presented an interface design which fulfills these requirements and implemented it on a

common PDA. It can be operated with one as well as two hands. The usability of the proposed solution was verified with a heuristic study. An additional evaluation proved the usefulness of the device in a mobile context. Minor design flaws, such as the appearance of the icon for circle shaped movements and the width of the area for speed manipulation, have been identified and will be corrected in the next implementation.

Heuristic studies are done with a low number of experienced interface experts and as such can only verify the usefulness and usability of a design. Further evaluations with a larger group of inexperienced users would certainly be worthwhile and might provide further insight into the general problem of one-handed interaction with mobile devices. In addition to such studies, we are planning to integrate automatic segmentation and an adaptation to left- and right-handed operation, as already mentioned in the preceding sections. In addition, we want to experiment with different thumbnail representations, such as always showing an increased version of the active thumbnail, comparable to expanding targets used for direct manipulation.

## 7. 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*, San Jose, CA, USA
- [2] W. Hürst (2008) Mobile Video - The Next Killer Application or Not? Workshop on *Mobile Multimedia – Content Creation and Use* at MobileHCI 2008, Amsterdam, Netherlands.
- [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] W. Hürst, K. Meier (2008) Mobile Video Browsing Interfaces *Proceedings of ACM Multimedia 2008*, Vancouver, BC, Canada.
- [6] A. Karlson, B. Bederson, J.L. Contreras-Vidal (2006) Studies in One-Handed Mobile Design: Habit, Desire and Agility. *Tech report HCIL-2006-02*, Computer Science Dept, University of Maryland, College Park, MD.
- [7] P. Parhi, A. Karlson, B. Bederson (2006) Target size study for one-handed thumb use on small touchscreen devices. *Proceedings of the 8<sup>th</sup> Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI 2006, Espoo, Finland.
- [8] A. Karlson, B. Bederson, J. SanGiovanni (2005) AppLens and LaunchTile: Two designs for one-handed thumb use on small devices. *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM CHI 2005.
- [9] A. Girgensohn, J. Boreczky, L. Wilcox (2001) Keyframe-based user interfaces for digital video. *IEEE Computer*, Vol. 34(9), pp. 61-67, 2001.
- [10] J. Kjeldskov, J. Stage (2004) New Techniques for Usability Evaluation of Mobile Systems. *International Journal of Human-Computer Studies*, Vol. 60, 2004.

## 8. APPENDIX

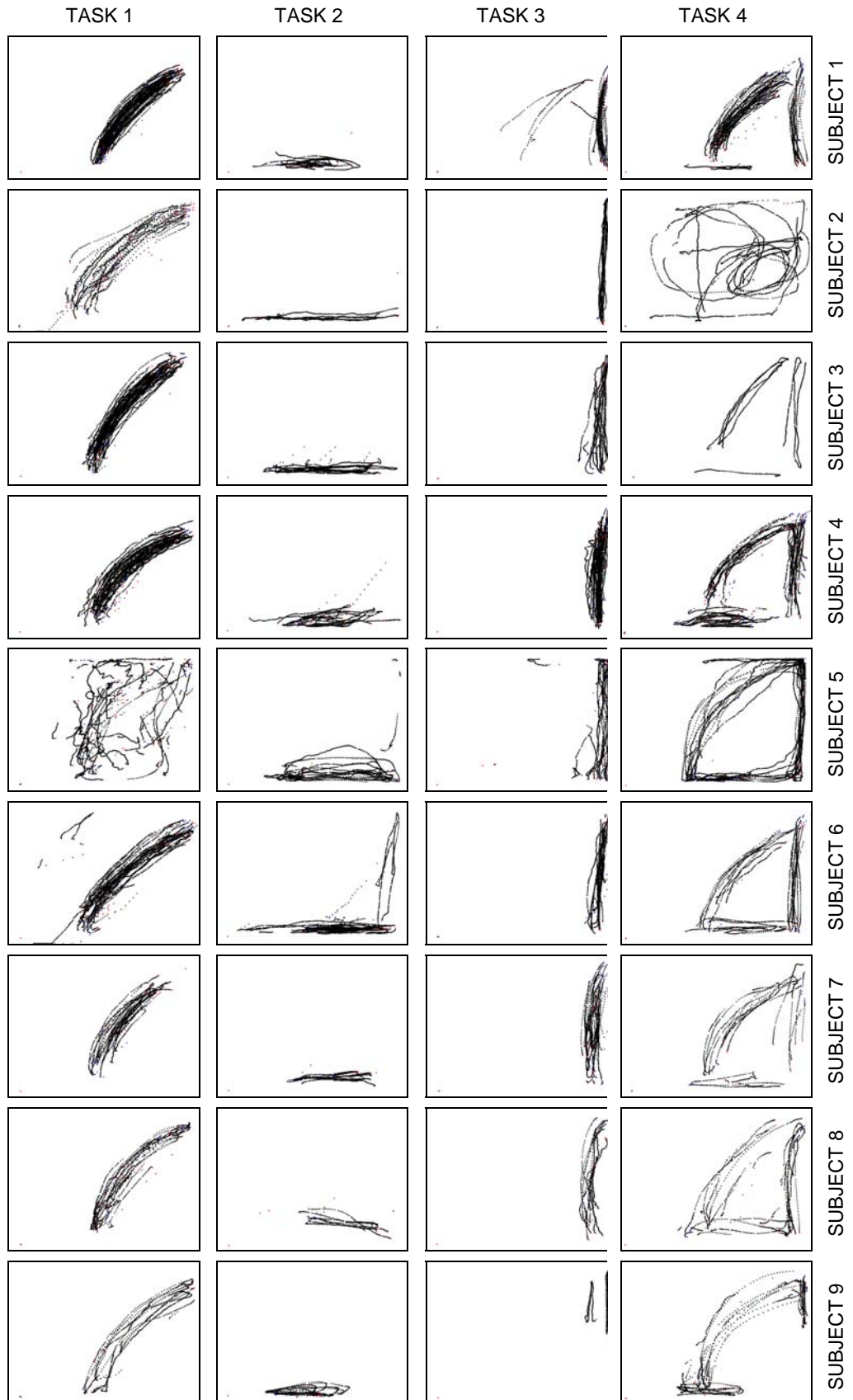
Visualization of the data that was captured in the first experiment (cf. Section 2). Each square represents the screen area and contains the thumb movements that were made by the subjects for each task.

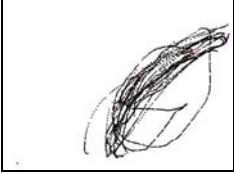
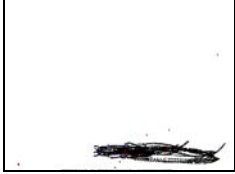

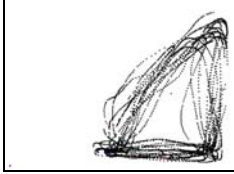

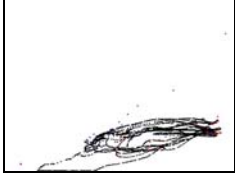

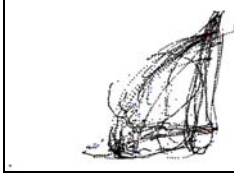

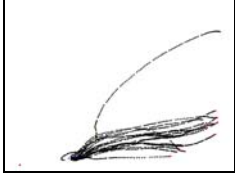

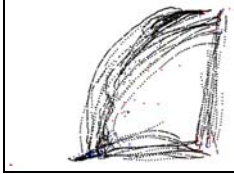





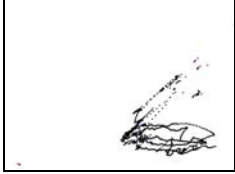
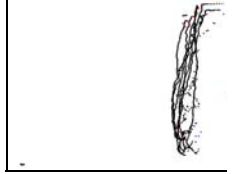
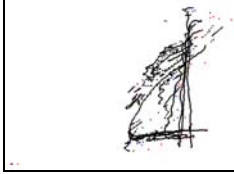
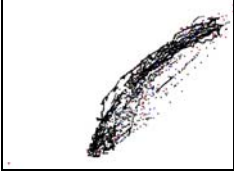


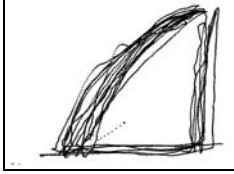




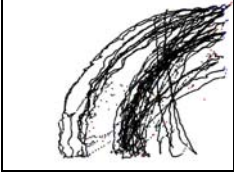
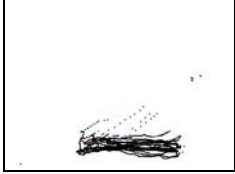
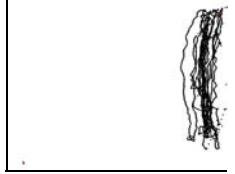
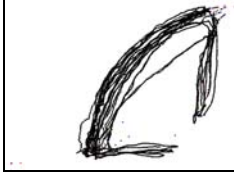
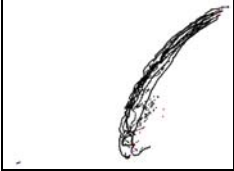


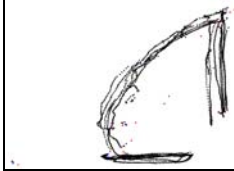
1st column: Task 1 (circular thumb movements in the center of the screen)

2nd column: Task 2 (horiz. thumb movements at the lower screen border)

3rd column: Task 3 (vertical thumb movements at the right screen border)

4th column: Task 4 (combination of all three interactions in random order)



	TASK 1	TASK 2	TASK 3	TASK 4	
					SUBJECT 10
					SUBJECT 11
					SUBJECT 12
					SUBJECT 13
					SUBJECT 14
					SUBJECT 15
					SUBJECT 16
					SUBJECT 17
					SUBJECT 18