Audio Engineering Society

Convention Paper

Presented at the 126th Convention
2009 May 7–10 Munich, Germany

Enhanced Control of On-Screen Faders with a Computer Mouse

Michael Hlatky[1], Kristian Gohlke[1], David Black[1], and Jörn Loviscach[2]

[1]Hochschule Bremen (University of Applied Sciences)
mhlatky@acm.org, kgohlke@acm.org, dblack@stud.hs-bremen.de
[2]Fachhochschule Bielefeld (University of Applied Sciences)
jl@j3L7h.de

ABSTRACT

Input devices of the audio studio that formerly were physical have mostly been converted into virtual controls on the computer screen. Whereas this transition saves space and cost, it has reduced the performance of these controls, as virtual controls adjusted using the computer mouse do not exhibit the accuracy and accessibility of their physical counterparts. Previous studies show that interaction with scrollable timelines can be enhanced by an intelligent interpretation of the mouse movement. We apply similar techniques to virtual faders as used for audio control, leveraging such approaches as controllable zoom levels and pseudo-haptic interaction. Tests conducted on five such methods provide insight into how to decouple the fader from the mouse movement to improve accuracy without impairing the speed of the interaction.

1. INTRODUCTION

On-screen controls, particularly sliding controls such as faders, constitute principal elements of the interaction with sound tools such as mixing consoles, synthesizers, and other studio equipment. Although physical faders are still important to sound engineers, the ongoing transition from analog consoles and editing desks to digital audio workstations has resulted in virtual faders that are less effective than their physical counterparts. The graphical representation of digital faders results in significantly inferior interaction. Screen real estate is limited, meaning that virtual faders are reduced in size to heights as low as 100 pixels. In standard implementations, this has led to corresponding losses in useable resolution, as the input is quantized into the number of pixels used to display the fader.

In pursuit of a more efficient and accurate approach, different interaction methods were tested that deviated from the direct mapping present in current virtual faders. To this end, we implemented a test setup that allows for a direct comparison between six different control mappings, which include the standard mapping and five modified versions. The software does not employ the default mouse data offered by the operating system, as this would be limited by the screen's resolution. Instead, the *raw* mouse data are interpreted to allow direct manipulation independent of the restrains of the screen. Such a restriction can be observed in Steinberg's Cubase 5[1] music production software, in which heightened fader accuracy can be attained by pressing a dedicated key on the keyboard during fader manipulation. This accuracy is limited, however, because the fader can only be manipulated while the mouse cursor stays within the borders of the screen. Furthermore, the mouse cursor strays from the virtual fader, which distracts the user.

2. RELATED WORK

Precise control of on-screen sliders with sub-pixel accuracy has been previously addressed in various contexts. Our evaluation focuses on faders that are commonly used to adjust parameters in audio software.

Ahlberg et al. [1] suggest a slider design that addresses the problem of quickly and accurately selecting items from a long list of text entries by decoupling the mouse movement from the display position of the slider. The display position of the on-screen slider gives only an approximation of the current position in the list, whereas the mouse movement is used to navigate the list directly. Guiard et al. [7], researched into a user interface that possesses two sliders to control a single value at a "macro" and a "micro" zoom level.

Ayatsuka et al. [4] propose a "popup vernier" slider that visualizes subpixel movement and allows for immediate transitions from coarse to fine movement. They compared this slider to the "Alphaslider" [1] and the "Fineslider" [12]. The results indicate that the "popup vernier" slider yields only a slightly faster performance, although test users highly preferred this method over the other sliders, as it permitted on-the-fly adjustments without hindering interaction.

Pirhonen et al. [13] describe a touchscreen interface for volume control. Faced with the challenge of enabling users to access the full volume range in one sweeping gesture while concurrently allowing fine adjustments, they opt for a solution in which multiple sweeping gestures were necessary to access all available volume levels. However, they deem this solution impractical for faders, but state that it is appropriate for touch-screen interaction.

Ramos et al. [14] propose the use of a force sensor for adjusting slider precision through the use of a pressure-sensitive pen. Cechanowicz et al. introduce an augmented mouse [5] that employed a pressure sensor to achieve an additional degree of input. They also suggest that for optimal performance, the pressure sensor should be located beneath the thumb or the ring finger.

In previous work [11], a member of our team explored the enhancement of physical faders by transmitting short high-voltage, low-current electrical impulses to a user's skin to both indicate fader markings and display an audio track's short-time frequency spectrum by using five electrodes mounted to the fader's knob.

3. ENHANCED FADERS

In the following text the notion of "zoom" levels refers to the mapping from mouse ticks to on-screen movement. A zoom value of one moves the fader one pixel per mouse tick. A zoom value of 16 moves the fader the 16^{th} fraction of a pixel per mouse tick. In the prototype, the zoom range is limited to the range 1 through 32.

The five enhanced mouse-to-fader mapping methods in our test are as follows:

3.1. Scroll wheel zoom

The control-to-display ratio of the mouse input can be adjusted by the scroll wheel of a standard mouse. With each consecutive level of zoom, the user gains a more precise level of control. Turning the scroll wheel up results in a higher zoom value. Only integer zoom values are allowed. The initial zoom value for this fader is 16.

3.2. Perpendicular zoom

Similar to approaches found in various timeline or linear list controls [3][8][9], This fader's resolution is controlled by the perpendicular distance between the mouse cursor and the fader. The closer the mouse cursor is to the fader, the more finely it can be adjusted. A greater distance leads to a quicker fader movement. This approach offers an adjustable level of accuracy and allows basic mouse gestures to quickly bring the fader to its extreme positions by simply moving the mouse diagonally. In our prototype, this fader starts with the

[1] http://www.steinberg.net/en/home.html

standard zoom factor of one. The adjustment of the accuracy is undertaken with a perpendicular mouse movement to the right-hand side. The maximum accuracy and minimum speed are reached at a distance of 128 pixels, allowing a maximum zoom value of 32.

3.3. Binary zoom

The simple binary zoom approach resembles the implementation found in Steinberg's Cubase 5 software, although the zoom is not invoked by pressing a key on the keyboard, but rather by using a different mouse button. The standard left mouse button sets the zoom value to one, resembling a standard fader, while the right mouse button sets the zoom value to a maximum of 32.

3.4. Pseudo-haptic zoom

A nonlinear mapping simulates a "bump" that represents a zone of heightened accuracy. Previous research [10] in pseudo-haptics has shown that when presented with a variable control-to-display input, users successfully identified certain textures in the interface. In our implementation, the further the fader is moved vertically from its initial position, the faster it moves until a direct mapping between the mouse pointer and the fader takes over. When the mouse button is released, the zone of heightened accuracy is recentered around the current fader position. As an option, the zone can slowly follow the fader position. In our prototype, the pseudo-haptic fader is initialized with a zoom factor of 32 in the initial position, which linearly decreases to a direct mapping over the length of the fader.

3.5. Pressure-sensitive zoom

In this prototype a mouse was fitted with an extra pressure sensor similar to the "EnhancedMouse" by Chechanowicz et al. [5]. The pressure sensor is activated by the thumb, see Fig. 1, and gives the user a third dimension of input. The user can control the zoom value by applying more or less pressure. When no pressure is applied, the zoom value is set to 32, allowing very careful adjustments. If the user applies more pressure, the zoom value decreases to a minimum of one, allowing very fast fader movements. We selected this mode of operation instead of one in which heightened pressure results in a maximum zoom level, based on the assumption that a firmer grip in the physical world would cause more change. This mode also helps to avoid prolonged thumb pressure when using the fader for a long time, assuming that the user will spend more time performing precise editing than large-scale movements.

Fig. 1: The pressure-sensitive zoom mode employs a mouse with an attached thumb-operated sensor. A spare pressure sensor is shown in front of the mouse.

4. IMPLEMENTATION

The faders with the six different control mappings were implemented in a C# application for Microsoft Windows and appear as common vertical on-screen faders similar to those found in standard audio editing software. All faders' on-screen heights are fixed at 128 pixels to resemble the value range of typical audio applications which often use MIDI data ranges. With each fader, the user can drag the knob to a certain position and release it to set the desired value, or directly click anywhere on the fader to immediately set the knob to that position.

In the user test, setting the fader is accompanied by visual and aural feedback to simulate the actual use in an application. Previous research [2] in feedback modes for pointing tasks with a mouse suggests that aural and visual feedback reduce the final positioning time. In our system, the visual feedback is provided by the fader's position on the screen, the corresponding value displayed in a number box, and seven colored dots similar to the LED arrays found on common guitar tuners, see Fig. 2. In the context of on-screen sliders, color has previously been used successfully as an ambient feedback method by Webb et al. [15].

Three red dots above a green dot indicate "too high", a green dot indicates the value has been reached (a "hit"), and three red dots below the green dot indicate "too low." If the fader's value is too high compared to the desired value, the color saturation of the three upper red dots indicates the distance of the current fader's value to the desired value. The closer the fader knob is brought to the desired position, the more saturated the green dot becomes. If the fader is set precisely to the desired value, the red dots disappear and only the green dot's color is completely saturated. If the set value of the fader becomes lower than the desired value, the lower red dots become saturated in the same fashion as the higher ones.

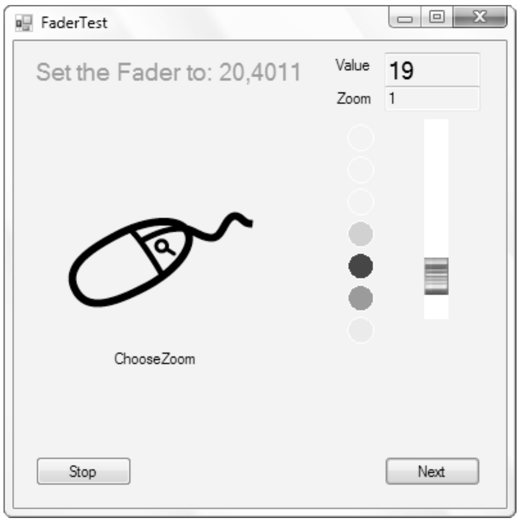

Fig. 2: The users were presented with a pictogram describing the fader's mode and with an arrangement of colored dots resembling a guitar tuner display that provided precise visual feedback about the proximity to the target value.

Audio feedback is implemented to convey the fader's position using a further channel of perception. Two sine tones are monophonically played back to the user via headphones, one tone staying at a constant pitch and the other one instantly reflecting the fader's position with a frequency range of 200 to 5000 Hertz. The transition between the different frequencies is implemented using a smooth progression without noticeably quantized steps. While setting the fader value, the user hears the frequency of the variable sine tone approach that of the steady tone. When nearing the desired value, the user perceives the beat frequency that results from the combination of the two closely-pitched sine tones.

As shown in Fig. 3, the faders are always presented together with a descriptive icon, which allows the user to easily identify which fader mode is currently being used.

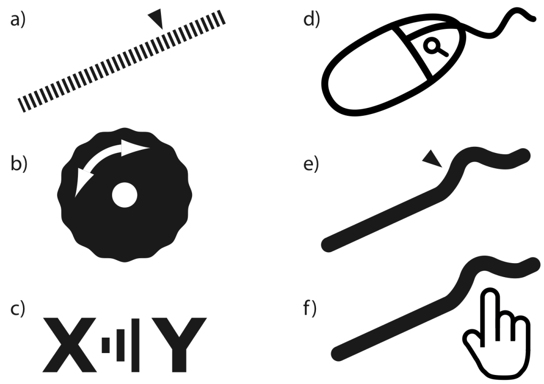

Fig. 3: The included icons designate the current fader mapping mode: a) no-zoom, b) scroll wheel zoom, c) perpendicular zoom, d) binary zoom, e) pseudo-haptic zoom, f) pressure-sensitive zoom.

Our implementation differs from faders found in standard audio software in that the mouse cursor is hidden when the fader knob is dragged, and re-centered to the middle of the fader knob when the mouse button is released.

5. USER TEST: SETTING

Users were asked to make use of the mouse to set the fader to a given numerical value displayed on the same screen, see Fig. 2. Due to the inherent characteristics of each fader type, this process differed slightly for each type, although all initial distances were close to the integer value of 20 to allow for a comparison between the different types. The first fader was to be set to the value of 20 as it only allowed integer precision, while the enhanced faders were to be set to 20.1337, 20.2674, 20.4011, 20.5348, and 20.6685, respectively.

An instructor guided each user through the complete test procedure, explaining the operational concept and encouraging the user to become familiar with each fader before data was recorded. During the ensuing test period, the user was left alone to set the fader to the given values. Before the test, the user was informed that the test did not predominantly depend on time nor accuracy, but that they should interact as naturally as they would when working with on-screen faders. By first touching the fader to be set, a timer was started and every five milliseconds the current fader position with the accompanying zoom value was written to a file.

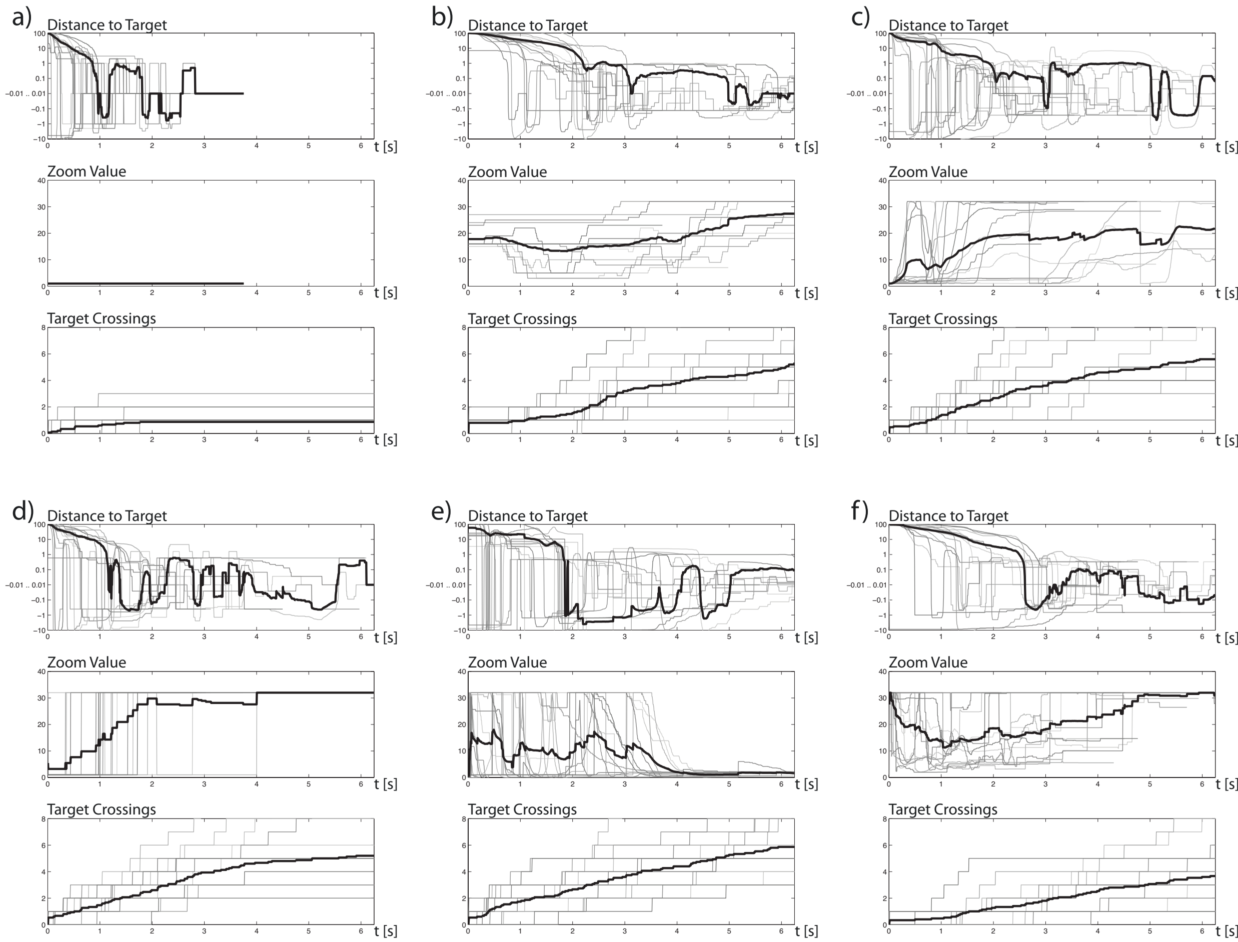

Fig. 4: a) No-zoom, b) scroll wheel zoom, c) perpendicular zoom, d) binary zoom, e) pseudo-haptic zoom, f) pressure-sensitive zoom. The single user's time performance values (thin lines) are shown in the upper subplots; the middle subplots display the zoom value. The lower subplots display the number of target crossings. The thick lines indicate the arithmetic mean.

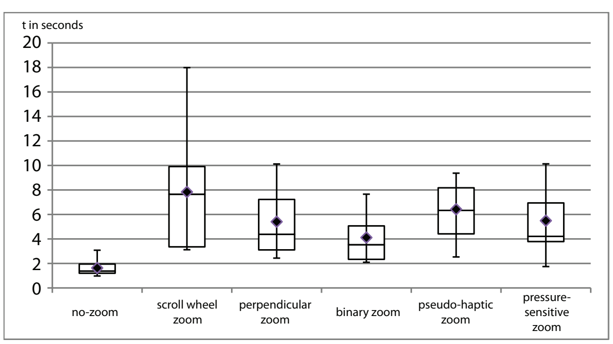

Fig. 5: Box-and-whisker plot of the novice users' performance when setting the fader to the desired values.

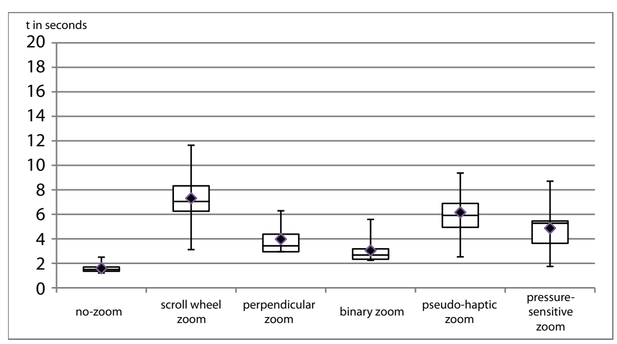

Fig. 6: Box-and-whisker plot of the expert users' performance when setting the fader to the desired values.

Furthermore, data was recorded that indicated when the user released the fader again, or used the “click to jump” functionality. By clicking the “next” button in the right hand corner of the test screen, the user submitted the fader as set.

6. USER TEST: RESULTS

The user test was performed by 16 unpaid volunteers (8 m, 8 f; aged 18 to 50), with normal or corrected vision. Based on the subjects’ experience with digital audio workstations, we considered five participants to be expert users and the remaining eleven participants to be novices or not highly experienced.

All users demonstrated the best performance with the standard, *no-zoom* fader (see Figs. 4a, 5, and 6). This was expected, because the effective value range that could be accessed using this fader was significantly smaller, as it offered only 128 discrete steps. However, this *no-zoom* fader is not sufficient for fine adjustments utilizing sub-pixel accuracy.

In all tested fader setups, the test subjects first performed a coarse movement of the fader, followed by a finer adjustment to precisely reach the desired target position. Related research [4] has shown similar results. As expected, the finer adjustments were usually made using a higher zoom value. Users tended to preserve higher zoom values once they were set. Most users neglected the middle range of the zoom values and only used the higher or lower values.

The *binary zoom* fader (see Fig. 4d) offered the best overall time performance of the tested subpixel-accurate faders, even though it exhibited a high target-crossing rate. These observations indicate that two zoom modes (coarse and fine) seem to be sufficient for fast, precise parameter adjustment.

Unlike the other faders, the *scroll wheel fader* (see Fig. 4b) was initialized with a medium zoom value. Surprisingly, the average zoom value chosen by the users did not differ much from the initial value. Overall, users did not often switch between zoom values. After setting a high zoom value, this value was often not changed to a significantly lower one. One test subject even selected a high zoom value before moving the fader and did not change the value thereafter.

The *perpendicular zoom* fader (see Fig. 4c) exhibited the second best time performance among the enhanced faders (see Figs. 5 and 6). Similar to the *scroll wheel zoom*, a majority of subjects quickly chose larger zoom values and preserved those until reaching the target position. Many users were confused because the zoom value changed even though they attempted to move the mouse solely along the vertical axis. The current area for setting the zoom value with the perpendicular zoom fader is equivalent to 128 on-screen pixels; a wider area might have reduced this confusion. Nonetheless, due to the design of standard computer mice, physical motion restricted exclusively to only one axis is hardly possible, or would require a significant period of training. Standard graphics software tackles this problem by employing a simple modifier key or button to ignore the input from one mouse axis.

The *pseudo-haptic* fader (see Fig. 4e) seemed to require a learning period. Even after the initial exploration phase, our subjects applied different strategies to handle the fader. Some subjects released the mouse button frequently to reset the center of the *pseudo-haptic* bump, which lead to an effectively lower average individual zoom value and a slower fader movement. Other subjects released the mouse button only when the fader had already moved past the target, frequently resulting in large overshoot.

The *pressure-sensitive zoom* fader (see Fig. 4f) is initialized with a high zoom value. Pressing the sensor gradually lowers the zoom value. Despite its significantly low target-crossing rate, this fader offered a rather poor time performance. Surprisingly, the in-between value range was used by a majority of subjects, while some effectively used the fader in a way similar to an inverted version of the binary zoom fader, applying either a large amount of pressure to the sensor or applying no pressure at all, particularly as the subject approached the target position.

As expected, expert users were generally able to reach the target values faster than novice users. Expert and novice users alike reported that they had at least once used a fader that was comparable to the *no-zoom* fader, perhaps accounting for the better time performance of the *no-zoom* fader.

Both user groups preferred the *binary zoom* fader to the remaining approaches. Some expert users also reported a preference for the *pressure-sensitive zoom* fader. Only one user preferred the *pseudo-haptic zoom* fader to all other faders, but stated that it should have a marker to indicate the current area of heightened accuracy, and suggested that the fader should change its “weight” (i.e. its display size) according to current zoom level.

Overall, our tests confirmed the rationale behind Fitts’s law [6], that a task involving moving or pointing is generally performed coarsely at first and thereafter using finer adjustments.

7. CONCLUSION AND OUTLOOK

In our test, users performed better with a simpler mouse-to-fader mapping, preferring this setup to more complex varieties. The user comments and the test results both indicate that two zoom levels suffice for fast and precise editing of parameters. The normal zoom value should not differ from the operating system’s mouse-to-pointer translation, and the heightened zoom value should be set according to the user’s preference, although this should not be higher than the threshold of just-noticeable differences in the parameter being adjusted. Furthermore, it was clear that when the fader is decoupled from the mouse pointer movement, hiding the mouse pointer while adjusting the fader results in a better performance, as the user is presented with less visual obfuscation. Moreover, the *raw* mouse data should be employed. Although this results in more complex computation, the screen’s borders no longer limit the user’s mouse movement, thus enabling a more fluent interaction with on-screen faders.

8. REFERENCES

[1] Ahlberg, C. and Shneiderman, B. The Alphaslider: a compact and rapid selector. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence (Boston, MA, USA, April 24-28, 1994). B. Adelson, S. Dumais, and J. Olson, Eds. CHI ’94. 365-371. 1994.

[2] Akamastu, M., MacKenzie, I.S. Hasbrouq, T. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. Ergonomics, 38, 816-827. 1995.

[3] Appert, C. and Fekete, J. OrthoZoom scroller: 1D multi-scale navigation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montreal, Quebec, CA, April 22-27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI ’06. 21-30. 2006.

[4] Ayatsuka, Y., Rekimoto, J., and Matsuoka, S. Popup vernier: a tool for sub-pixel-pitch dragging with smooth mode transition. In Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (San Francisco, CA, USA, November 01 - 04, 1998). UIST ’98. 39-48. 1998.

[5] Cechanowicz, J., Irani, P., and Subramanian, S. Augmenting the mouse with pressure sensitive input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007). CHI ’07. 1385-1394. 2007.

[6] Fitts, P.M., and Peterson, J.R. Information capacity of discrete motor responses. Journal of Experimental Psychology. Vol. 67(2), February 1964, 103-112. 1964.

[7] Guiard, Y., Beaudouin-Lafon, M., and Mottet, D. Navigation as multiscale pointing: extending Fitts’ model to very high precision tasks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit (Pittsburgh, PA, USA, May 15-20, 1999). CHI ’99. 450-457. 1999.

[8] Hürst, W., Jarvers, P. Interactive, dynamic video browsing with the zoomslider interface. In Proceedings of the IEEE International Conference on Multimedia and Expo (Amsterdam, NL, July 6-8, 2005). ICME 2005. 2005.

[9] Hürst, W. Interactive audio-visual video browsing. In Proceedings of the 14th Annual ACM international Conference on Multimedia (Santa Barbara, CA, USA, October 23-27, 2006). MULTIMEDIA ’06. 675-678. 2006.

[10] Lécuyer, A., Burkhardt, J., and Etienne, L. Feeling bumps and holes without a haptic interface: the perception of pseudo-haptic textures. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vienna, AT, April 24-29, 2004). CHI ’04. 239-246. 2004.

[11] Loviscach, J. Music at Your Fingertips: An Electrotactile Fader. Presented at the 123rd AES Convention. (New York, NY, USA. October 5-8, 2007). 2007.

[12] Masui, T., Kashiwagi, K., and Borden, G. R. Elastic graphical interfaces to precise data manipulation. In Conference Companion on Human Factors in Computing Systems (Denver, CO, USA, May 07-11, 1995). I. Katz, R. Mack, and L. Marks, Eds. CHI ’95. 143-144. 1995.

[13] Pirhonen, A., Isomäki, H., Roast, C., Saariluoma, P. (Eds.), Future Interaction Design, Springer, New York, NY, USA. 119-120. 2005.

[14] Ramos, G. and Balakrishnan, R. Zliding: fluid zooming and sliding for high precision parameter manipulation. In Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (Seattle, WA, USA, October 23-26, 2005). UIST 2005. 143-152. 2005.

[15] Webb, A. and Kerne, A. The in-context slider: a fluid interface component for visualization and adjustment of values while authoring. In Proceedings of the Working Conference on Advanced Visual interfaces (Napoli, IT, May 28-30, 2008). AVI ’08. 91-99. 2008.