Adaptive Pointing – Implicit Gain Adaptation for Absolute Pointing Devices

Werner A. König
HCI-Group, University of Konstanz
Box-D73, Universität Konstanz
78457 Konstanz, Germany
werner.koenig@uni-konstanz.de

Jens Gerken
HCI-Group, University of Konstanz
Box-D73, Universität Konstanz
78457 Konstanz, Germany
jens.gerken@uni-konstanz.de

Stefan Dierdorf
HCI-Group, University of Konstanz
Box-D73, Universität Konstanz
78457 Konstanz, Germany
stefan.dierdorf@uni-konstanz.de

Harald Reiterer
HCI-Group, University of Konstanz
Box-D73, Universität Konstanz
78457 Konstanz, Germany
harald.reiterer@uni-konstanz.de

CHI 2009, April 4 – April 9, 2009, Boston, MA, USA
ACM 978-1-60558-247-4/08/04.

Abstract
We present Adaptive Pointing, a novel approach to addressing the common problem of accuracy when using absolute pointing devices for distant interaction. The intention behind this approach is to improve pointing performance for absolute input devices by implicitly adapting the Control-Display gain to the current user's needs without violating users' mental model of absolute-device operation. First evaluation results show that Adaptive Pointing leads to a significant improvement compared with absolute pointing in terms of movement time (19%), error rate (63%), and user satisfaction.

Keywords
Adaptive Pointing, bubble test, pointing precision, hand tremor, control-display gain, distant interaction, laser-pointer.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies, Interaction styles, Evaluation/methodology.

Introduction
With the steadily growing diversity of application domains beyond standard desktop usage, absolute pointing devices are becoming more and more favored. Absolute devices use a position-to-position mapping

(mouse: velocity-to-velocity) as the transfer function between the input device and the display pointer [8]. As a result the user benefits from a more natural and convenient pointing experience [11] and easier hand-eye coordination compared with the decoupling of motor and display spaces and the non-linear pointer acceleration when using relative pointing devices. Due to the direct mapping of absolute pointing devices, the user can easily keep track of the cursor, since it is always in line with the user's finger, stylus, laser pointer or any other absolute device.

Besides home entertainment (e.g. Nintendo Wii), there are various other application domains in, for example, the fields of ubiquitous computing, visual analytics, collaborative environments and interactive exhibitions, where users need the flexibility of absolute pointing devices to interact effectively. Especially in combination with large, high-resolution displays, there is a need for input devices that provide more user mobility, allowing the user to work close to the display with detailed information and also to step back and manipulate the contents of the entire display space [14]. This trend is also reflected in research literature, with several authors proposing solutions for absolute input devices such as freehand pointing [14] or laser-pointer interaction [11],[12].

However, the pointing precision is a common problem shared by all absolute input devices operated from a distance, particularly in combination with high-resolution displays. Myers et al. concluded that "interaction techniques using laser pointers tend to be imprecise, error-prone, and slow" [11]. Vogel et al. reported a similar result for their comparison of absolute, relative and hybrid mapping of hand movements. While the absolute technique was significantly faster than the hybrid and relative ones, the high error rates of the absolute mapping "prevent it from being a practical technique" [14]. Based on previous related work and our experience, we identified two main factors for this serious imprecision of absolute pointing devices used in midair: deviations are caused by **natural hand tremor** and limited **human pointing precision**.

Natural Hand Tremor

The task of maintaining a part of a limb in a constant position produces involuntary muscular contraction with rhythmical oscillations (8-40 Hz) referred to as physiological tremor [13]. When using freehand pointing or absolute pointing devices in midair without a stable rest, such natural tremor causes serious noise, which makes accurate pointing and selection more difficult or even impossible as the distance between display and user increases. A variety of approaches exist to reduce noise and so to steady the cursor, such as discrete or dynamic moving windows (Myers et al. [11], Vogel et al. [14]) or using a Kalman filter (e.g. [12]) to smoothen the pointing behavior. While all approaches seem to increase the accuracy they also introduce a noticeable time lag, which reduces the responsiveness of the pointing device. To date, we are not aware of a systematic investigation that compares and ranks these smoothing approaches. All authors report a general improvement, but eliminating noise for pointing movements without introducing a certain amount of delay or reduction of responsiveness seems to be impossible for such reactive methods. Besides, it is questionable whether even the most perfect jitter compensation would, on its own, provide sufficient pointing accuracy. Another factor has to be considered as well: human pointing precision.

Human pointing precision

Absolute pointing devices are characterized by a position-to-position mapping. Hence, the pointer motion in display space is proportional to the movement in motor space. When interacting from a greater distance, for example in a presentation situation or when using a high-density display, the effective pixel size on the display might fall below human pointing precision. In such a case, even if the tremor compensation worked perfectly, the user would not be able to move discretely one pixel at a time because of limited hand-eye coordination, restricted motor precision, and the necessary but unachievable fine control of the muscle groups involved in the movement (see [4] for a more detailed discussion). When using a relative input device such as the mouse, the human precision limit can be overcome by lowering the Control-Display gain (CD gain = $velocity_{Pointer}$ / $velocity_{Device,}$). The CD gain modulates the mapping between the physical input device and the virtual display pointer. With a low-gain transfer function the pointer velocity in display space is several times slower than the actual velocity of the pointing device in motor space. Thus, low CD gain allows for precise targeting even in the case of high-density displays or distant interaction. On the downside, moving long distances is highly inefficient. This speed-accuracy trade-off can be solved by varying the CD gain during interaction. This approach is the basis for several interaction techniques and was also the fundamental design principle of our Adaptive Pointing technique.

We will discuss these different techniques according to a classification scheme we have developed. We thus distinguish between *target-oriented*, *manual-switching*, and *velocity-oriented* approaches. Target-oriented techniques basically use a metaphor approach based on magnetism or stickiness by lowering the CD gain when the pointer either enters a target (e.g. [5]) or when it comes close to a target, thus creating a fisheye effect in motor space (e.g. [1], [3]). As a precondition, however, a semantic knowledge of the environment is required, and having to deal with large numbers of targets can be problematic. The *manual-switching* approaches rely on the user to manually switch between absolute and relative pointing when appropriate (e.g. [7],[14]). Evaluation results showed that error rates are significantly lowered, but sometimes at the price of selection time. A further downside is that the cognitive and physical load of switching explicitly between the two modes remains with the users. The last group, the *velocity-oriented* approaches are motivated by the optimized-submovement model [10], which states that most aimed movements consist of an initial, large and fast movement towards the target followed by a few slower, corrective movements to compensate for over- or undershooting. The movement velocity in motor space indicates in which phase of the movement the user is and which degree of precision or velocity in display space should be beneficial. This is the basis of all relative pointer-acceleration techniques already widely in use, for example by default in Mac OS X and Windows XP. Based on this approach, Frees et al. introduced the PRISM technique which dynamically adjusts the CD-gain between the hand and the controlled object in a virtual 3D environment [6].

(1) $$\hat{v}_x(t) = \begin{cases} 1 & if \quad v_x(t) > v_{max} \\ 0 & if \quad v_x(t) < v_{min} \\ \frac{v_x(t) - v_{min}}{v_{max} - v_{min}} & otherwise \end{cases}$$

(2) $$d_x(t) = x_{mot}(t) - x_{disp}(t-1)$$

(3) $$\hat{d}_x(t) = \begin{cases} 1 & if \quad |d_x(t)| > d_{max} \\ 0 & if \quad |d_x(t)| < d_{min} \\ \frac{|d_x(t)| - d_{min}}{d_{max} - d_{min}} & otherwise \end{cases}$$

(4) $$m_{\mathrm{x}}(\mathrm{t}) = \max\left(\hat{\mathrm{v}}_{\mathrm{x}}(\mathrm{t}), \hat{\mathrm{d}}_{\mathrm{x}}(\mathrm{t})\right)$$

(5) $$g_x(t) = g_{min} + \tfrac{1}{2}\left[\sin\left(m_x(t) \cdot \pi - \tfrac{\pi}{2}\right) + 1\right](g_{max} - g_{min})$$

(6) $$s_{\mathrm{x}}(\mathrm{t}) = x_{mot}(t) - x_{mot}(t-1)$$

(7)
$$\hat{g}_x(t) = \begin{cases} 1 - (\, g_x(t) - 1\,) & if \quad g_x(t) > 1 \ AND \ d_x(t) > 0 \ AND \ s_{\mathrm{x}}(\mathrm{t}) < 0 \\ 1 - (\, g_x(t) - 1\,) & if \quad g_x(t) > 1 \ AND \ d_x(t) < 0 \ AND \ s_{\mathrm{x}}(\mathrm{t}) > 0 \\ g_x(t) & otherwise \end{cases}$$

(8) $$x_{disp}(t) = x_{disp}(t-1) + \hat{g}_x(t) \cdot s_x(t)$$

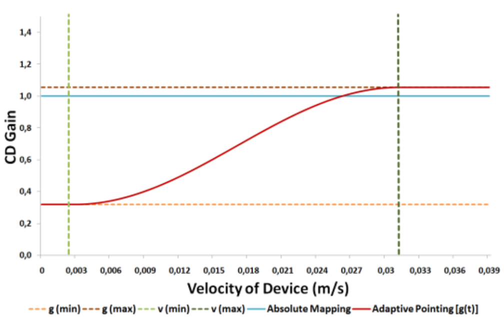

Figure 1: Smooth transition between relative and absolute CD-gain of Adaptive Pointing

Adaptive Pointing

We introduce the Adaptive Pointing technique, which can also be classified as a velocity-oriented approach, relying on the optimized-submovement model of Meyer et al. [10] discussed above. However it differs from similar concepts such as PRISM by simulating absolute pointing behavior. The basic idea is to improve pointing performance for absolute input devices by implicitly adapting the CD-gain to the current user's needs without violating the users' mental model of absolute-device operation. Users expect a 1:1 mapping between their device movement in motor space and the resulting pointer movement in display space when using an absolute pointing device. Adaptive Pointing appears to provide this pure absolute behavior but imperceptibly lowers the CD-gain when higher precision is needed.

While PRISM works very well in the dedicated virtual environment for professional users, it has some obvious drawbacks when applied to a more general setting of (simulating) absolute pointing devices. Since the system visualizes the offset between display space and motor space movement, the device does no longer seem to be an absolute pointing device to the user. This also reduces the intuitiveness and ease of use of the device, as the user has to understand at first how this gap between motor space and display space arises and how to deal with it. The absolute pointing behavior is furthermore flawed by the necessary offset reduction. PRISM increases the CD-gain by the amount that is needed so that the offset is nullified within a period of about one second. This, however, should result in a noticeable "jumping" which would lead to an unnatural and unexpected behavior. Furthermore in case of movement direction changes, it might be that the pointer in display space is actually "in front" of the motor space movement. In such a case PRISM lets the users catch up the offset by themselves, which results in a non-movement of the pointer in display space. Again, this behavior results in a reduced ease of use and intuitiveness of the technique when applied to the more generic setting of an absolute pointing device.

Comparing the Adaptive pointing with the manual-switching approaches, for example [14], [7], the user is not explicitly involved in the gain variation and thus does not need to decide which technique would be most suitable for the next task. Unlike target-oriented approaches such as [1] and [3], Adaptive Pointing does not need any knowledge of the displayed information or active elements. However, it can be easily combined with visual interaction techniques such as expanding targets [9] or Drag-and-Pop [2], as well as hand-tremor compensations (e.g. Kalman filter) if further pointing and selection improvement is desired.

Adaptive Gain

The Adaptive Pointing technique dynamically adjusts the CD-gain depending on the movement velocity and the current offset between the motor-space position and display-space position. Figure 1 shows the behavior for the velocity factor. As soon as a predefined minimal velocity threshold is met the CD-gain is smoothly decreased. We describe this behavior in the following equations, but only for the horizontal case indicated by the index *x*. Vertical movement is calculated likewise. In figure 2, a flow chart illustrates how the equations are combined in the end. The first step of the iterative position mapping between motor and display space is the normalization of the velocity, which serves as an indicator of the users' need and as the main controlling

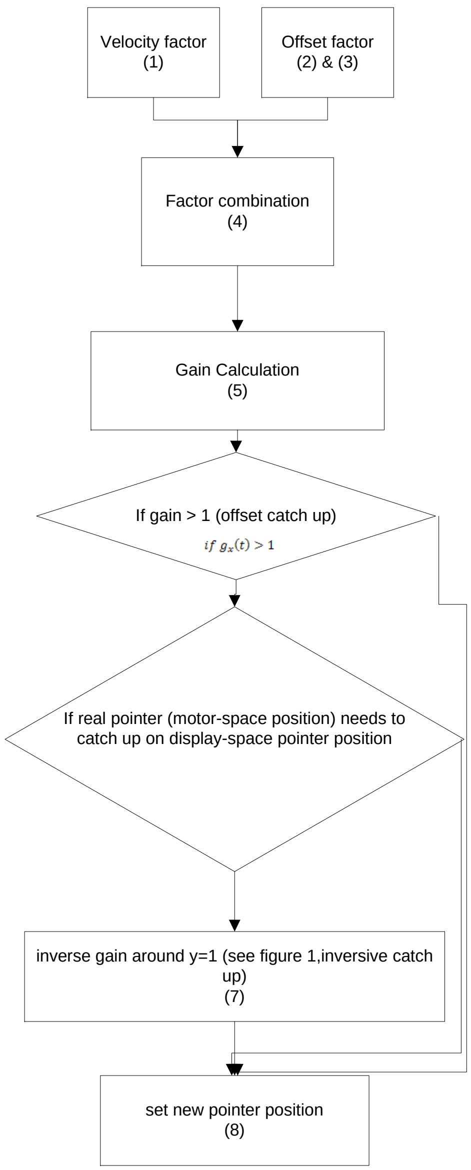

Figure 2: Flow-chart of the Adaptive Pointing algorithm

factor (Eq. 1). The upper limit is $\mathrm{v_{max}}$, which marks the threshold from where the CD gain decreases until the lower limit $\mathrm{v_{min}}$ is reached. Velocities below $\mathrm{v_{min}}$ and above $\mathrm{v_{max}}$ are also limited to a value range of 0 to 1 (Eq. 1). Since we want to ensure an absolute pointing behavior, it is important that the offset between the position in motor space and in display space is considered as well. Eq. 2 describes the offset calculation and the normalization is done likewise to the velocity normalization (Eq. 3). For further calculations we use the larger one of these two factors (Eq. 4). Since we want to avoid abrupt switches during the transition from constant gain (absolute mapping) to the varying gain (relative mapping), we use a modulated sine wave as damping function (Eq. 5). When the user decreases speed to aim at a target, the CD gain is smoothly adapted by the modulated sine wave until the minimum gain is reached or the user increases the movement speed again. When the CD gain is lowered, however, the pointer moves more slowly in display space than the input device in motor space. This results in an offset between the detected pointing position and the modulated pointer position. In case of either a high velocity or a large offset, the gain calculation reaches values above 1 and up to a predefined maximum. In case that the pointer position in display-space trails behind the position in motor space this results in a smooth catch-up. For the opposite case that the position in display space is "in front" of the position in motor space (e.g. due to a change of direction) we flip the part of the sine wave for which applies CD-gain>1 at the CD-gain=1.0 axis (Eq. 7). Thereby we reach a gain value slightly below 1 which allows a reverse catch-up of the offset. The new pointer position in display space is then calculated by applying the current CD gain $\mathrm{g(t)}$ as a factor to the last movement in motor space (Eq. 6) and adding this to the last position $\mathrm{x_{disp}(t-1)}$ in display space (Eq. 8). This approach allows a smooth and continuous pointer movement that is regulated by parameters for the maximum and minimum values for the CD-gain, the movement velocity, and the offset between display- and motor-space. As pointed out before, this is an important difference to approaches like the PRISM technique, which furthermore does not consider the size of the offset but only the velocity of the movement.

We used Adaptive Pointing in combination with an infrared laser-pointer interaction system at a 221″ large-high resolution display (8.9 megapixels Powerwall) to explore the potential as well as the constraints of the novel interaction technique. This is obviously a very demanding setting for absolute pointing techniques, since the user has to point at, select and manipulate very small objects from a distance of several meters (e.g. the Windows start button is only 22mm in height on such a display). During iterative testing and configuration we found the following parameters most beneficial for this setting: $\mathrm{v_{min}} = 0.0028\,{}^{\mathrm{m}}/_{\mathrm{s}}$, $\mathrm{v_{max}} = 0.0312\,{}^{\mathrm{m}}/_{\mathrm{s}}$, $d_{min} = 47\mathrm{px}$, $d_{max} = 232\mathrm{px}$, $\mathrm{g_{min}} = 0.32$, and $\mathrm{g_{max}} = 1.055$. Figure 1 illustrates the resulting CD gain with respect to the velocity of the input device in motor space for the parameter set used.

Evaluation & Conclusion

We compared the Adaptive Pointing technique to a state of the art Kalman filter enhanced absolute pointing in an experiment with 24 participants. We used a multi-directional tapping task similar to the ones commonly used in Fitts' Law studies. The experiment provided some clear-cut results. In every single aspect, the Adaptive Pointing technique proved to be significantly better than the absolute pointing. We observed a mean reduction in error rate (effectiveness) of about 63%, as well as more efficient usage in terms of movement time (19% mean difference). While many former approaches suffered a clear speed-accuracy trade-off [14], the Adaptive Pointing performed better in both aspects. Besides, although people did recognize a change in behavior, and seven out of 21 ascribed this change to the laser pointer, no user felt that the laser pointer behaved unnaturally. Our participants clearly ascribed positive characteristics to the Adaptive Pointing technique and rated it as significantly better compared with absolute pointing. For future work we would like to analyze more in detail, how Adaptive Pointing changes the pointing behavior, e.g. if strategies might change to approach and hit a target in the long run. In such a longitudinal study we would also be interested to analyze and compare learning effects of the Adaptive Pointing technique compared to other pointing enhancement techniques. Furthermore, we found out that the Fitts' Law as used for other pointing devices does not hold for the Adaptive Pointing technique. Thereby, we would be interested to find a suitable enhancement to Fitts' Law for such pointing techniques, especially in combination with very small targets.

Citations

[1] P. Baudisch, E. Cutrell, K. Hinckley, and A. Eversole. Snap-and-go: helping users align objects without the modality of traditional snapping. In *Proc. CHI '05*, pages 301–310, 2005.

[2] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. T, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. Interact '03*, pages 57–64, 2003.

[3] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proc. CHI '04*, 2004.

[4] S. K. Card, J. D. Mackinlay, and G. G. Robertson. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, pages 99–122, 1991.

[5] A. Cockburn and A. Firth. Improving the acquisition of small targets. In *Proc. British Computer Society Conference on HCI '03*, pages 181–196, 2003.

[6] Frees, S., Kessler, G. D., and Kay, E. PRISM interaction for enhancing control in immersive virtual environments. ACM Trans. Comput.-Hum. Interact. 14, 1 (May. 2007), 2.

[7] C. Forlines, D. Vogel, R. Balakrishnan. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST '06*, 2006.

[8] K. Hinckley. Input technologies and techniques. *Handbook of Human-Computer Interaction*, pages 161–176, 2008.

[9] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *Proc. CHI '02*, 2002.

[10] Meyer, R. A. Abrams, et al.. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 1988.

[11] B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proc. CHI '02*, 2002.

[12] J. Oh and W. Stuerzlinger. Laser pointers as collaborative pointing devices. In *Proc: GI '02*, 2002.

[13] E. Vaillancourt and K. M. Newell. Amplitude changes in the 8–12, 20–25, and 40 hz oscillations in finger tremor. *Clinical neurophysiology*, 111(10):1792–1801, 2000.

[14] Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST '05*, pages 33–42, 2005.