Mid-air Pointing on Ultra-Walls

Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, Michel Beaudouin-Lafon

Mid-air Pointing on Ultra-Walls

MATHIEU NANCEL, Université Paris-Sud & CNRS, Inria; University of Canterbury; University of Waterloo

EMMANUEL PIETRIGA, Inria, Université Paris-Sud & CNRS; Inria Chile

OLIVIER CHAPUIS, Université Paris-Sud & CNRS, Inria

MICHEL BEAUDOUIN-LAFON, Université Paris-Sud & CNRS, Inria

Ultra-high-resolution wall-sized displays ("ultra-walls") are effective for presenting large datasets, but their size and resolution make traditional pointing techniques inadequate for precision pointing. We study mid-air pointing techniques that can be combined with other, domain-specific interactions. We first explore the limits of existing single-mode remote pointing techniques and demonstrate theoretically that they do not support high-precision pointing on ultra-walls. We then explore solutions to improve mid-air pointing efficiency: a tunable acceleration function and a framework for dual-precision techniques, both with precise tuning guidelines. We designed novel pointing techniques following these guidelines, several of which outperform existing techniques in controlled experiments that involve pointing difficulties never tested prior to this work. We discuss the strengths and weaknesses of our techniques to help interaction designers choose the best technique according to the task and equipment at hand. Finally, we discuss the cognitive mechanisms that affect pointing performance with these techniques.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces, Input Devices and Strategies.

General Terms: Design, Performance, Experimentation

Additional Key Words and Phrases: pointing, dual-precision, large displays, ultra-high resolution

ACM Reference Format:
Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon, 2015. Mid-air Pointing on Ultra-Walls. *ACM Trans. Comput.-Hum. Interact.* 22, 5, Article 21 (August 2015), 62 pages.
DOI: http://doi.acm.org/10.1145/2766448

1. INTRODUCTION

Ultra-high-resolution wall-sized displays, or *ultra-walls* for short, are a new generation of wall-sized displays made of a mosaic of LCD panels [Nam et al. 2009; Nam et al. 2010]. With about 100 pixels per inch and more than 100 million pixels overall, they have three times the pixel density and ten times the resolution of the previous generation of projection-based, very-high-resolution wall-sized displays. Such high density and resolution support the display of large datasets with a high level of detail while retaining context, and enable the juxtaposition of data of various types [Andrews et al. 2010], including small text that is perfectly legible from up close.

These displays are well suited to the visualization of, e.g., very large maps and complex networks (Figure 1), complex molecule simulations, or astronomy imagery with associated metadata from astronomical catalogs [Beaudouin-Lafon et al. 2012]. The combination of large size and high resolution affords a natural form of multiscale interaction: simply by walking, a user can smoothly transition from an overview of the whole display when standing at a distance to the fine details of a specific area by getting up close [Ball et al. 2007; Yost et al. 2007; Liu et al. 2014; Jakobsen and Hornbæk 2014]. It is thus critical that users be able to point at very small objects on the screen efficiently, whether they are standing far away from the display or within arm's reach [Vogel and Balakrishnan 2005].

This work is supported by the Region Île-de-France/Digiteo grant 2008-25D "WILD" and by the French National Research Agency grant ANR-10-EQPX-26-01 "DIGISCOPE".

Authors' addresses: M. Nancel <mathieu@nancel.net>: University of Waterloo, Canada. E. Pietriga <emmanuel.pietriga@inria.fr> & O. Chapuis <chapuis@lri.fr> & M. Beaudouin-Lafon <mbl@lri.fr>: Université Paris Sud, 91405 ORSAY Cedex, France.

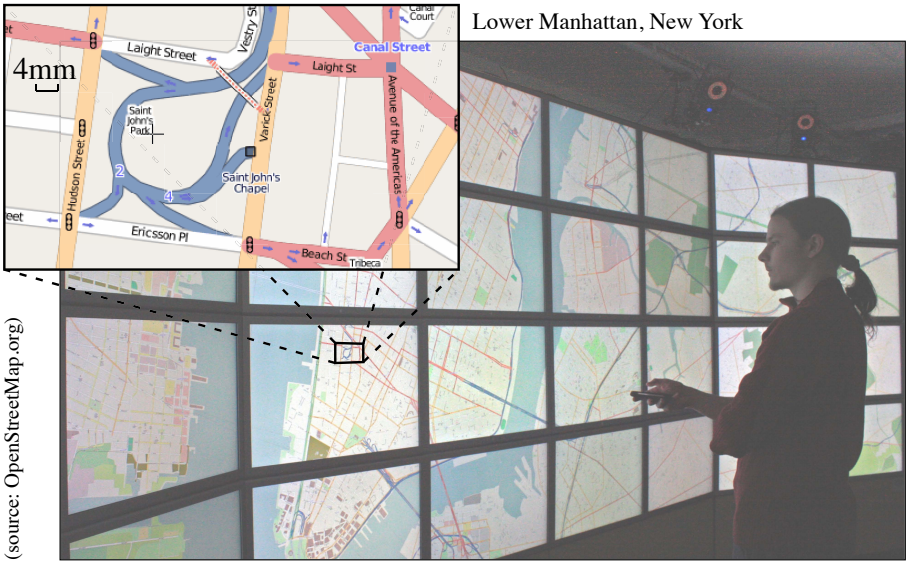


Fig. 1: The WILD ultra-wall used for our studies (5.5m × 1.8m for a resolution of 20 480 × 6 400 = 131 million pixels). Inset: magnification of a 9cm × 5cm area.

In addition to this basic requirement, pointing techniques for ultra-walls must also support concomitant interactions such as panning and zooming [Nancel et al. 2011] or invoking domain-specific commands. They must therefore minimize the *input requirements* both in terms of human motor capabilities and device sensing capabilities. For example, if a pointing requires both hands (motor capability) or the entire surface area of a touch tablet (device sensing capability), it will be more difficult to perform other interactions without introducing modes. Pointing should also avoid dividing the user's visual attention between the wall display and the input device and it should not be tiring even when used for long periods of time. Finally, large displays are well suited to collaborative work [Jakobsen and Hornbæk 2014], therefore several users must be able to point simultaneously and the pointing techniques must not hinder the tasks carried out by other users.

Distant pointing at large displays has been studied in various contexts, ranging from low resolution displays to high-resolution back-projected walls. However, it has not been studied in the context of ultra-walls that can display much smaller visual elements that users must still be able to select.

This paper addresses the problem of target-agnostic[1], mid-air and eyes-free high-precision pointing on ultra-walls: given the very high pixel density and with no a priori knowledge of the target's location, can we design remote pointing techniques that enable users to efficiently select a target that is both small and far away from the cursor with minimal input requirements and without looking away from the display?

We investigate this question by first identifying the limits of modeless devices in a theoretical study. We then consider two families of techniques that vary the Control-to-Display (CD) gain[2] during the pointing movement: Pointer Acceleration [Casiez and Roussel 2011], a technique that adapts the CD gain to the velocity of the input movement and for which we introduce a calibration method that allows very difficult pointing tasks to be performed using very limited movement ranges; and Dual-Precision techniques [Nancel et al. 2013; Nancel et al. 2011; Vogel and Balakrishnan 2005; McCallum and Irani 2009], a family of pointing techniques that feature two levels of precision, a *Coarse* positioning mode to quickly approach the vicinity of the target and a *Precise* pointing mode for acquiring the smallest targets. To address our goal of minimizing input requirements for pointing, we explore two families of Dual-Precision techniques: unimanual techniques that allow other devices to be used simultaneously in the non-dominant hand, and touch-based techniques that can be used on small input devices, e.g. smartphones, or that use a small part of a tablet screen, leaving space for other widgets to be displayed at the same time. We also introduce a method to calibrate the parameters of the *Coarse* and *Precise* modes depending on an estimate of the most difficult pointing task in a given environment, and a model for predicting pointing time for these Dual-Precision techniques.

[1] A target-agnostic pointing technique is a technique that does not need any knowledge of the location and size of *individual* targets; however we assume that the *range* of target locations and sizes can be estimated by the application designers or by domain experts.
[2] Ratio between cursor movement and input device displacement.

We designed a variety of pointing techniques that can be used eyes-free and with minimum input requirements and we evaluated them in four controlled experiments. We find that our calibration methods make it possible to use both Pointer Acceleration and Dual-Precision techniques at a distance for pointing tasks with a high index of difficulty. Given these results we present a set of design guidelines to select which techniques and input methods can be used depending on the available input devices and the required pointing difficulty. Finally we discuss the cognitive costs associated with varying the CD gain during a pointing task.

2. RELATED WORK

A large corpus of techniques and devices have been developped and studied for mid-air pointing. We review them according to the way they map user input to cursor movements. First we briefly review pointing techniques that use velocity control (first-order control). We then consider position control (zeroth-order control) mid-air pointing techniques by distinguishing between absolute and relative mappings, and finally we report on "hybrid" techniques that feature both types of controls.

2.1. Rate-based mid-air pointing techniques

Techniques that map relative device motion to cursor displacements can be based on position control or rate/velocity control (zeroth or first order of control). [Rutledge and Selker 1990] designed and evaluated transfer functions for isometric pointing joysticks at the center of the keyboard and found it faster than a normal mouse for intermixed pointing and keyboard tasks in a desktop environment. However they do not report the mouse gain nor whether it was speed-dependent. More recent studies [Campbell et al. 2008; Natapov et al. 2009] and our own tests show that techniques based on rate control are faster and more comfortable for coarse pointing across large distances, but perform poorly during the final precise pointing phase. [MacKenzie and Jusoh 2001] compared a regular mouse with a gyroscopic mouse[3] held on a table and then in mid-air, and to a handheld isometric joystick. The task was performed 1.52m away from a 15" screen. The joystick and the gyro-mouse held in mid-air performed poorly compared to the mice. More recently, [Casiez and Vogel 2008] evaluated rate-controlled pointing techniques with isometric and elastic devices and with several CD gains, but did not evaluate them against position-controlled techniques.

2.2. Position-based absolute mid-air pointing techniques

Absolute pointing devices or techniques map an input state, e.g. the location and orientation of a hand-held device, to a cursor location on the display. Techniques based on the absolute position of the input device include the family of *ray-casting* techniques, also called "laser pointing" [Myers et al. 2002; Oh and Stürzlinger 2002; Olsen and Nielsen 2001]. These techniques extend the user's finger, arm, or hand-held device with an imaginary ray whose intersection with the display defines the cursor position. [Kopper et al. 2010] and [Jota et al. 2010] formulate angular-based models of pointing time that fit ray-casting performance better than Fitts' law.

While intuitive, ray casting is essentially angle-based and thus degrades quickly with distance to the display because hand tremor and involuntary motion due to fatigue are amplified as the user is farther away from the display surface [Myers et al. 2002; Oh and Stürzlinger 2002]. It is therefore not suitable for small targets on ultra-walls. [Olsen and Nielsen 2001] adapted existing interaction techniques to the limitations of this technology. Both [Chen and Davis 2000] and [Oh and Stürzlinger 2002] designed collaborative pointing devices based on laser pointers, enabling several users to interact with the display simultaneously. The latter also compared a laser pointer with a conventional mouse in a pointing task. The laser pointer performed significantly worse than the mouse on a 1.83m $\times$ 1.22m low-resolution back-projected screen, but was preferred by users.

[Myers et al. 2002] studied the effect of human body limitations on laser pointing. They compared the pointing performance of a laser pointer with that of a regular mouse, a touch-sensitive SmartBoard™ and *Semantic snarfing*. With the latter, users point with a stylus on a handheld device that displays a copy of a region from the main screen. The technique requires users to look at the handheld device, creating a division of attention. Direct input standing in front of the SmartBoard was the most efficient technique, followed by Semantic Snarfing. Laser pointer was the worst technique. Except for the SmartBoard which required direct contact, other conditions were performed seated about 1.52m away from the display.

Another ray-casting technique consists in holding a device at arm-length in front of the eyes so that the target is aligned with the tip of the device [Pierce et al. 1997]. The technique is interesting as it resembles aiming,

[3] A mouse that uses gyroscopic sensors, also called gyro-mouse.

but our own tests [Nancel 2012, App. B] revealed its limitations: it is more tiring and less precise than laser pointing, causes visual occlusion, and requires users to repeatedly switch between two very different focal lengths.

Some techniques use absolute input but focus more on the interaction vocabulary than on pointing performance. For example, VisionWand [Cao and Balakrishnan 2003] tracks the position of a wand in 3D using two low-cost cameras. The two ends of the wand have different colors and can be distinguished by the vision system. While it does not improve distant pointing performance, it enables interactions such as tap, tilt, flip and rotate gestures. Other works use vision-based techniques to enable freehand pointing. [Nickel and Stiefelhagen 2003] recognize pointing gestures with two cameras. They introduce new pointing techniques using information such as head and forearm orientation, but focus on the recognition of relevant gestures among a sequence of arbitrary movements rather than precision of pointing gestures. With Shadow Reaching [Shoemaker et al. 2007], users reach distant objects through the shadow of their body cast on the display surface by a light source. Because of projection perspective, the regions that can be reached depend on both the setup and the user's distance to the display. As with laser pointing, the accuracy of this technique decreases with distance to the display.

Absolute mappings can also be used in combination with a small hand-held device. With the Touch projector [Boring et al. 2010], users manipulate objects located on a distant display using a smartphone (iPhone) through a live video feed showing that display.

Finally, the Wiimote and other game controllers have also been studied as general-purpose pointing devices. [Campbell et al. 2008] evaluated a Wiimote operated as a zero-order or first-order pointing device, and found that participants were roughly 2.5 times faster in the zero-order condition. [Natapov et al. 2009] compared remote pointing with a Wiimote, a classic gamepad's joystick, and a mouse operated on a desk as baseline. They found that the mouse had the best throughput, followed by the Wiimote and the joystick, and reported that hand tremor and small movements greatly affected accuracy in the Wiimote condition for small targets.

2.3. Position-based relative mid-air pointing: Pointer Acceleration

Relative position-based control maps the user's motion to the movements of the cursor, regardless of the original location of that motion. A wide variety of devices and techniques, including the widespread mouse and trackpad for desktop computers, use this technique. Jellinek *et al.* describe and explore a design space of such mappings [Jellinek and Card 1990] for mice on desktop computers. They found that varying gains were preferred but did not improve performance over fixed gains in their experimental setup. In this subsection we only describe relative pointing techniques that use a transfer function between the velocity of the user input and that of the cursor ('Hydromatic-glide' in [Jellinek and Card 1990]) because constant gains have seldom been used in the literature for mid-air pointing. Indeed, as we demonstrate later in this article, constant-gain transfer functions result in insufficient range and precision to be usable on most large displays at a distance.

Pointer Acceleration consists in varying the ratio between the velocities of the cursor and the input movement[4], based on the assumption that users perform faster movements when they want the cursor to move far away and that they perform slower movements when precisely adjusting the cursor's position. Pointer Acceleration is used by all major operating systems. [Casiez and Roussel 2011] have measured the velocity transfer functions of these systems and showed that they varied significantly in shape, but not necessarily in their input and output domains.

Despite the widespread use of Pointer Acceleration, there is very little literature on how to design effective transfer functions for specific input and output systems such as ultra-walls. The PRISM technique [Frees et al. 2007] and its subsequent refinements [König et al. 2009; Gallo and Minutolo 2012] are among the very few that have documented their transfer functions. However, they were designed to support implicit absolute-to-relative transfer functions and, as we show later in this paper, this approach does not scale to the high indices of difficulty that users of ultra-walls are exposed to.

2.4. Position-based "hybrid" mid-air pointing techniques

Some techniques, rather than making the whole range of CD gains accessible through a continuous transfer function, provide users with two pointing modes, usually an absolute one for large-amplitude movements and a relative one for slower, more precise movements. With HybridPointing [Forlines et al. 2006], users can reach distant objects by switching from absolute to relative pointing by pressing with the pen used for

[4]We refer to this ratio as the Control-to-Display (CD) gain.

pointing inside a trailing widget that follows the cursor. The technique requires direct contact of the pen with the display surface, which makes it impossible to operate in mid-air.

ARC-Pad [McCallum and Irani 2009] uses a touch-sensitive mobile device for cursor positioning on large displays. When the user taps the screen, the position of the tap is mapped to the entire display, enabling coarse but fast repositioning of the cursor. When the user drags on the touch surface, the finger movements are interpreted as relative input, allowing precise adjustments to the cursor's position.

[Vogel and Balakrishnan 2005] use a high-precision 3D motion tracking system to develop and evaluate three techniques: pure ray casting, relative pointing with clutching, and ray-to-relative pointing, which combines absolute and relative pointing using two different hand postures. We adapted the latter to our environment and tested it in the experiment described in Section 5.

3. LIMITATIONS OF THE HUMAN SENSORY-MOTOR SYSTEM FOR MID-AIR POINTING

Ultra-walls are powerful interactive platforms both in terms of display capacity and input modalities [Nancel 2012; Beaudouin-Lafon et al. 2012]. Combined with computer clusters and appropriate software, dozens of high-density displays can be tiled so as to be used as a single, unified surface [Pietriga et al. 2011; Schwarz et al. 2012]. At the same time, advances in input sensors make it easier and cheaper to use input channels such as mid-air gestures [Nancel et al. 2011], whole-body input [Wagner et al. 2013] and multi-sensor devices such as smartphones and tablets in a collaborative context [Chapuis et al. 2014].

These technological advances actually surpass the sensory-motor capabilities of the human body. For example, [Aceituno et al. 2013] showed that the input resolution of pointing devices exceeds the motor resolution of most users, and recent display technology such as Apple's Retina displays reach and sometimes go beyond human visual acuity. In this section we analyze the limits of the human visual and motor systems in the context of mid-air interaction on large displays. We later use the results of this analysis in the design of mid-air pointing gestures adapted to ultra-walls.

3.1. Limitations of the human visual system

The largest ultra-walls range from 5-meter to 10-meter wide and 1.8-meter to 3-meter high. These sizes are useful to display large quantities of information, especially when coupled with high pixel densities. Even at a distance, they can be visually larger than desktop screens while allowing fine details, and therefore small targets, to be displayed.

There is a limit distance beyond which a user with normal vision can no longer perceive a target. The angular width of a target seen by a user at a distance D from the display is

$$\beta = \tan^{-1}\left(\frac{A+\frac{w}{2}}{D}\right) - \tan^{-1}\left(\frac{A-\frac{w}{2}}{D}\right) \quad (1)$$

where w is the target's width and A the distance between the target and the orthogonal projection of the user's point of view on the display[5], i.e. the point directly facing the user (point P in Fig. 2-a). According to the theory of visual acuity, β must be greater than or equal to 1' of arc ($\frac{1}{60}^\circ$) for a "normal" user (20/20 vision) to be able to distinguish the target [Ware 2004], and greater than or equal to 5' of arc ($\frac{1}{12}^\circ$) to be readable [Vogel and Balakrishnan 2005]. Solving this inequality for D results in a half disc of diameter $\emptyset = w/\tan(\beta)^\circ$ centered at $\left(A = 0, D = \frac{\emptyset}{2}\right)$.

Fig. 2-b shows the inequality plot for a user with 20/20 vision and targets of 4 mm and 10 mm. The plot shows that the user must stand closer than respectively 13.8 m and 34.4 m from the screen to be able to distinguish these targets directly facing him, and closer than respectively 2.8 m and 6.9 m to read them. Another reading of this plot is that regardless of the user's distance to the display, 4 mm targets further away than 6.9 m from point P (the point facing the user) and 10 mm targets further away than 17.2 m from point P cannot be perceived accurately by users with normal vision. Similarly text labels cannot be read if they are further away than 1.4 m (4mm or 11-point text) and 3.4 m (10mm or 28-point text) from P.

Using the same formula, we can compute the smallest pixel size w that can be accurately perceived at a given distance by a user facing the screen:

$$w = 2 \times D \times \tan\left(\frac{\beta}{2}\right) \quad (2)$$

[5]Reduced to a single point for simplicity.

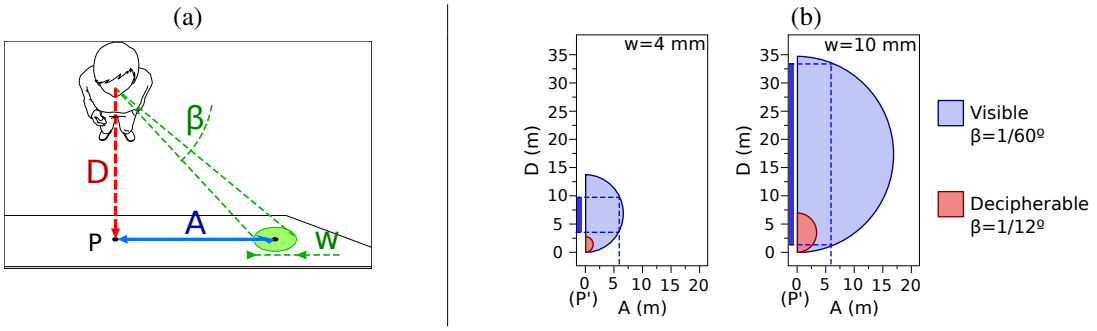


Fig. 2: (a) The user looks at a target of width w at an offset A from the orthogonal projection P of his eyes on the display and from a distance D to the display. (b) The possible values of D for which a 4-mm target (left) and 10-mm target (right) can be accurately perceived (blue) and read (red) for a given distance A. As an example, the thick blue line on the Y axis represents the range of distances at which the user can accurately distinguish pixels that are $A = 6$ m away from his projected point of view P.

Using a conservative minimal distance D of 2 m and the readability limit of $\frac{1}{12}^{\circ}$, this smallest size is $w =$ 2.9 mm (or about 8 points for text). Smaller targets will require users to step forward to be accurately read. In the following, we use $w = 4$ mm as our smallest target width at 2 meter distance in order to evaluate the hardest mid-air pointing tasks that can be performed on ultra-walls. We chose a value slightly larger than 2.9 mm to account for non-perfect vision, even when corrected.

Note that the limits of visual acuity do not invalidate the use of high-resolution large displays. Large displays improve collaboration and performance by allowing users to navigate physically [Ball et al. 2007]. They outperform desktop computers for interactive tasks that can take advantage of the high pixel density of ultra-walls to display numerous small textual elements simultaneously [Liu et al. 2014]. [Yost et al. 2007] showed that the advantages of high pixel density are not limited by visual acuity. In particular, users were more efficient and sometimes more accurate performing information visualization tasks on a display with a pixel density exceeding visual acuity. [Tan et al. 2006] showed that *regardless of visual angles*, physically large displays improve performance and precision for spatial tasks such as mental rotation, 3D navigation and mental map formation and memory. They hypothesize that large displays improve the users' sense of immersion and presence and help them to use more optimal cognitive strategies.

3.2. Limitations of the human motor system

Over the past ten years, a number of physical input devices have been explored for pointing on large displays. As with desktop pointing, some techniques map the absolute position of the input device to the cursor's position, while others use its relative motion to control cursor displacements.

We are specifically interested in techniques that allow users to point from afar while standing or walking in front of the display. Therefore we do not consider direct techniques that use a pen [Guimbretière et al. 2001] or direct touch [Buxton et al. 2000; Streitz et al. 1999] since they require users to stand within physical reach of the display. Similarly, we do not consider systems that can work from afar but that require users to sit at a table. For example, [Malik et al. 2005] introduce a vision-based system for whole-hand gestural interactions performed on a constrained tabletop area. The system supports precise target acquisition on a back-projected wall-sized display from afar using asymmetric interactions, but is designed for users seated at a table. The rest of this section focuses on techniques that can be used in mid-air.

Position-based pointing techniques require a *transfer function* [Nancel et al. 2013] mapping input motion, as captured by the device, to cursor motion. The most common transfer function uses a Control-to-Display (CD) gain, defined as the ratio between cursor movement, generally measured as a linear distance[6] and input variation, expressed in the same unit. Multiplying the magnitude of the input device motion by the CD gain therefore gives the amplitude of cursor motion. Since a gain is by definition without unit, both input and output amplitudes must be expressed in the same units. However, some techniques or devices map input and output of different natures. For example, a gyroscopic mouse transforms angular movements into cursor translations. In such cases the "gain" has a unit, i.e. meters per degree in this example. For the sake of simplicity and since the same computational mechanism is applied whether the multiplier has a unit or not, we also refer to such values as CD gains. In the rest of the paper we indicate the units of gains only when input and output are of different types.

[6]Note however that ray-casting techniques can be modeled as an angular-to-angular mapping of gain 1.

The value of the CD gain is not necessarily constant. Most operating systems vary the CD gain with input velocity, based on the principle that slow input motion occurs when precise cursor movements are intended and that fast input motion occurs when coarse cursor movements are intended. This specific velocity-based relative technique is called Pointer Acceleration [Casiez and Roussel 2011]. However, the literature about Pointer Acceleration functions is rather scarce, and existing functions were not designed for ultra-walls. Jellinek *et al.* argued that Pointer Acceleration (and all types of 'Powermice' in general [Jellinek and Card 1990]) does not improve pointing performance, and is only preferred because it minimizes the device footprint. However their experimental setup was limited to mice and desktop computers. This hypothesis remains to be validated on a wall-sized display.

The set of relative pointing techniques and devices that can work for ultra-walls can be refined by analyzing the devices' characteristics using the framework defined by [Casiez et al. 2008]. This framework provides formulae to compute upper and lower bounds for the CD gain, noted CD_{max} and CD_{min}. A gain below CD_{min} requires clutching, which is believed to decrease performance. A gain above CD_{max} causes precision problems because of hand tremor and/or device quantization. If CD_{min} is greater than CD_{max} (Fig. 3), these problems are compounded: any CD gain will trigger at least one of them, or both.

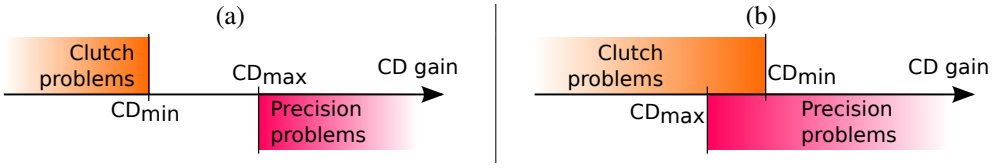


Fig. 3: (a) Problems that arise when the CD gain value is lower than CD_{min} or larger than CD_{max}. (b) When CD_{min} is greater than CD_{max}, no CD gain value can avoid both problems.

These formulae use an estimate of the minimum target width (W_{min}), the maximum distance between targets (A_{max}), the pixel density of the display (Res_{screen}), the device's morphological characteristics —operating range (OR) and input resolution (Res_{device})— and human motor precision (Res_{hand}). All these parameters must be expressed in the same physical unit, e.g. millimeters or inches, but not in pixels because display pixels and device *ticks* (or minimal input) often have different physical sizes: a CD gain used with a 600 dpi mouse on an 80 PPI screen should have the same effect as a 1600 dpi mouse on a 100 PPI screen.

The formulae from [Casiez et al. 2008] use Res_{device} and Res_{screen} expressed as densities in pixels per inches (ppi), which forces all other parameters to be expressed in inches, or to convert units. We use the multiplicative inverse of these measures instead, i.e., the physical size of pixels, or Point Size (PS), so that they can be expressed in any length unit: PS_{device} and PS_{screen} represent the point size of the input and display devices, respectively[7]. Using these conventions, the formulae from [Casiez et al. 2008] are rewritten as follows:

$$CD_{min} = \frac{A_{max}}{OR} \tag{3}$$

$$CD_{max} = \min(CD_{qmax}, CD_{lmax}) \tag{4}$$

$$CD_{lmax} = \frac{W_{min}}{Res_{hand}} \tag{5}$$

$$CD_{qmax} = \frac{Res_{device}}{Res_{screen}} = \frac{PS_{screen}}{PS_{device}} \tag{6}$$

CD_{min} (Eq. (3)) is the ratio between the largest amplitude of a pointing task on a given display (A_{max}) and the largest device input movement (OR); it represents the minimum CD gain below which clutching is bound to occur. If the CD gain is lower than this value, the input movement needed to make the cursor travel a distance of A_{max} is larger than the operating range of the technique / device, resulting in the need to clutch.

Allowing clutching reduces CD_{min} but has a cost. For example, clutching can be an effective pointing strategy on a small touch area [Nancel et al. 2015; Beaudouin-Lafon et al. 2014] but this result might not hold when performing an angular gesture using the whole arm. If we define c as the maximum number of clutches that should happen in a normal pointing task, then the cursor must go through the maximum

[7] Note that Res_{hand} should also be renamed PS_{hand} for it represents a "point size", i.e. the smallest input that users can perform. However since we do not change its definition, as opposed to PS_{screen} and PS_{device}, we keep the same name for consistency with [Casiez et al. 2008].

pointing amplitude A_{max} in less than $c+1$ movements of amplitude OR. We can thus relax Eq. (3) as follows:

$$CD_{min} = \frac{A_{max}}{(c+1)OR} \tag{7}$$

Note that clutching can only occur with relative controls. An absolute control maps OR to at least A_{max}, corresponding to $c = 0$. Thus, *precision* problems can occur but not *clutching* problems.

CD_{lmax} (Eq. (5)), i.e. the maximum CD gain given limb precision, is the ratio between the smallest target size for a given task (W_{min}) and the smallest human input (Res_{hand}); it represents the maximum CD gain beyond which human precision problems start to occur, i.e., the minimal input that can be used to keep the cursor stable within the smallest targets. If the CD gain is higher than this value, acquiring targets of size W_{min} requires more precision than normally available.

CD_{qmax} (Eq. (6)), i.e. the maximum CD gain given device quantization, is the ratio between the smallest display output (PS_{screen}) and the smallest device input (PS_{device}); it represents the maximum CD gain beyond which quantization problems start to occur, i.e., the minimal input that can be used to move the cursor by one pixel. If the CD gain is higher than this value, the smallest device input results in a movement larger than one pixel, thus some pixels become unreachable.

In the context of a ultra-wall and especially at a distance, it is not always necessary, or feasible, to reach every single pixel on screen. In order to relax this constraint on CD_{qmax}, we replace PS_{screen} by W_{min}, the minimum target size. The resulting formula is the maximum CD gain beyond which targets of this size become unreachable:

$$CD_{qmax} = \frac{W_{min}}{PS_{device}} \tag{8}$$

This is equivalent to Casiez *et al.*'s original formula when W_{min} is the size of one pixel. In addition to relaxing the constraint that every pixel must be reachable, our formulation can also be used to model sub-pixel targets, e.g., when using lenses [Appert et al. 2010], by making W_{min} smaller than the pixel size PS_{screen}.

Using equations (4), (5) and (8), we can rewrite CD_{max} as follows:

$$CD_{max} = \frac{W_{min}}{PS_{input}}, \text{ with } PS_{input} = \max(Res_{hand}, PS_{device}) \tag{9}$$

PS_{input} is the general "point size" of the user+device input system: it represent the smallest movement that can be performed by the input device or the user.

Now that we have established formulae for CD_{min} and CD_{max}, we can assess the range of pointing tasks that can be performed on an ultra-wall by computing the maximum Fitts' index of difficulty of such tasks.

Fitts' law [Soukoreff and MacKenzie 2004] is an empirical model that predicts movement time (MT) as a function of movement amplitude (A) and target width (W):

$$MT = a + b \times log_2(1 + A/W)$$

where a, b are determined empirically and depend on factors such as input device and user population, and $log_2(1+A/W)$ is called the index of difficulty (Fitts' ID) of the task and is measured in bits. Using Eq. (7) and (9), we compute a theoretical Fitt's index of difficulty, ID_{max}, beyond which CD_{min} cannot be lower than CD_{max} for a given technique:

$$\begin{aligned} & CD_{min} && < CD_{max} \\ \iff & \frac{A_{max}}{(c+1)OR} && < \frac{W_{min}}{PS_{input}} \\ \iff & \frac{A_{max}}{W_{min}} && < \frac{(c+1)OR}{PS_{input}} && (10) \\ \iff & ID_{max} && < IIC_c && (11) \end{aligned}$$

$$\text{with} \begin{cases} ID_{max} & = \log_2\left(1+\dfrac{A_{max}}{W_{min}}\right) \\ IIC_c & = \log_2\left(1+\dfrac{(c+1)OR}{PS_{input}}\right) \end{cases} \tag{12}$$

IIC_c stands for Index of Input Capability and represents the amount of information (in bits) that users can provide using a given input device and a maximum number of clutches c. It is the counterpart of the Index of Difficulty, which describes the amount of information that must be provided to acquire a target. Eq. (11) provides a theoretical limit for the Fitts' index of difficulty (ID_{max}) of a pointing task [8] beyond which a given technique starts causing precision problems and/or more clutching than expected (Fig. 3).

We now use these results to assess the suitability of a number of input devices and techniques for high-precision pointing on large displays.

3.3. Analysis of existing devices and techniques

The following input devices are position-controlled and can be used in mid-air. Table I summarizes the relevant characteristics of each device and the computed values of ID_{max}, CD_{min} and CD_{max} according to the formulae in the previous section.

(1) *Ray-casting*, i.e. simulated laser pointing, positions the cursor at the intersection of a virtual ray originating from the user's pointing device and the display, with the ray's orientation matching that of the pointing device. Ray-casting uses absolute angular control, so its number of clutches (c) is zero and its gain is 1 deg/deg [Gallo and Minutolo 2012]. Target sizes and amplitudes must be expressed in angular units to account for the fact that off-centered targets require more precision than targets of the same linear size positioned in front of the user. [Nancel et al. 2011] showed that pointing time and error rates increased dramatically with targets smaller than 0.53 degrees ($W = 30$ mm, $A = 3187$ mm, participants standing 2 meters away from the display).

(2) A *gyroscopic mouse* (*GyroMouse*) converts angular movements of a mouse held in mid-air into conventional mouse events. Users can clutch using a button that freezes the cursor. We could not find a gyroscopic mouse without automatic pointer acceleration, so we simulated one using a VICON[9] tracking system in order to control its resolution and CD gain. [Nancel et al. 2011] showed that pointing time and error rates increased dramatically with targets smaller than 0.26 degrees ($W = 15$ mm, $A = 3187$ mm, participants standing 2 meters from the display).

(3) *Soap* [Baudisch et al. 2006] wraps the tracking system of a mouse in a hull made of fabric. Users control the cursor by moving the tracking system inside the fabric, like a piece of soap in the hand. Relative motion of the hull enables both precise positioning of the cursor and moving across large distances. The resolution is that of a regular mouse (600 to 800 PPI), but the operating range without clutching is much smaller (about 3.5 cm).

(4) Some *unimanual trackballs* can be operated in mid-air. Their operating range is rather small[10] The best commercial desktop trackballs have a resolution of 1000 PPI and an operating range of approximately 4 cm.

(5) Cordless *trackpads* can be used in mid-air with two hands, one holding the device and the other interacting with it. The more recent ones are also multi-touch, enabling greater expressiveness for operations other than pointing. The input resolutions of such trackpads can reach 300 PPI and more.

(6) Touch-sensitive devices such as PDAs or *smartphones* can be used as augmented touchpads. They can be operated with one hand using the thumb of the holding hand. They have the additional advantage of being able to display information on their screens as the user interacts with them. The coordinates of touch points are often expressed in screen pixels, so the accessible touch resolution corresponds to the screen resolution. Medium-grade touchscreens can track touch events at 150 PPI while the most advanced ones can reach 325 PPI.

[8] Note that the *max* subscript in ID_{max} represents the maximum index of difficulty for the *technique* being considered, while in A_{max} it represents the estimated highest amplitude of the *pointing task*.
[9] http://www.vicon.com/
[10] Some of these trackballs can be "thrown", i.e., the user can initiate a fast rotation in the direction of the target and let the ball roll until it approaches the target or stops by itself. This use of a trackball is not covered by Casiez et al.'s formulae. We leave the analysis of such throwing techniques for future work.

(7) *Tablets* also provide visual feedback with greater input surfaces than touchpads and smartphones, but must be used with two hands. Their touch resolution is between 100 PPI and more than 250 PPI.

					$c=0$			
Device	Hands	PS_{device}	OR	Res_{hand}	ID_{max}	CD_{min}	CD_{max}	Min c
Soap	1	32 μm (800 PPI)	35 mm	0.2 mm	7.46	91.1	20	4
Trackball	1	32 μm (800 PPI)	40 mm	0.2 mm	7.65	79.7	20	3
GyroMouse	1	15.5 mdeg (65 PPD)[11]	90 deg	0.26 deg	8.42	35.4 $\frac{mm}{deg}$	15.2 $\frac{mm}{deg}$	2
RayCasting	1	15.5 mdeg	77 deg	0.53 deg	7.21	1	0.13	N/A
$Smartphone_m$	1	156 μm (163 PPI)	75 mm	0.2 mm	8.55	42.5	20	2
$Smartphone_t$	1	78 μm (325 PPI)	89 mm	0.2 mm	8.8	35.9	20	1
Trackpad	2	66 μm (387 PPI)	105 mm	0.2 mm	9.04	30.4	20	1
$Tablet_m$	2	192 μm (132 PPI)	197 mm	0.2 mm	9.94	16.2	20	0
$Tablet_t$	2	96 μm (264 PPI)	197 mm	0.2 mm	9.95	16.2	20	0

$$A_{max} = 3187 \text{ mm}, W_{min} = 4 \text{ mm}$$

Table I: Device characteristics for pointing techniques. Entries in red have $CD_{min} > CD_{max}$ and are therefore impractical. Min c represents the minimum number of clutches that would be necessary to ensure that $CD_{min} < CD_{max}$. Subscripts $_m$ and $_t$ indicate *m*edium-grade and *t*op-grade devices. We used the larger dimension for touch devices.

In order to compute CD_{min} and CD_{max} for each of the above devices, we must define their operating range (OR) and their hand and device resolutions (Res_{hand} and PS_{device}). For the *Trackpad*, *Trackball*, *Smartphones*, *Tablets* and *Soap*, we used Casiez *et al.*'s conservative estimate for Res_{hand} (0.2 mm) while for *GyroMouse* we adapted the formulae to obtain CD gains expressed in mm/deg. We used the smallest target size evaluated in Subsection 3.1 ($W_{min} = 4$ mm) and maximum amplitude $A_{max} = 3187$ mm), resulting in a maximum Fitts' index of difficulty of 9.64 bits.

Table I summarizes the CD gain computations for the selected devices. All but the tablets have a CD_{min} larger than their CD_{max}. They are therefore very likely to create clutching and/or precision problems if used with constant CD gains. This is also indicated by their ID_{max}, which is smaller than the maximum Fitts' index of difficulty of the task, and by their minimum number of clutches Min c. We confirmed this assessment by informally evaluating various handheld trackballs, trackpads, smartphones and tablets.

This analysis shows that with the exception of tablets, which must be operated with two hands, fixed CD gains make it difficult at best to perform the more challenging yet realistic mid-air pointing tasks that ultra-walls may require. The users' motor capabilities and the devices' sensing capabilities are too limited to acquire targets that users can visually perceive with normal visual acuity, at least with constant gains. The use of clutching can improve the situation but at the cost of greater movement times, and only for relative techniques. This confirms that ultra-walls are a limit case in Jellinek *et al.*'s hypothesis [Jellinek and Card 1990]: pointing techniques with varying gains become necessary beyond a certain level of pointing difficulty, because the precision required to acquire very small targets leads to high upper bounds on the size of the 'device footprint' (OR in the equations above).

This problem is best described by Eq. 10-12: for a given user, device and acceptable number of clutches, there is a maximum pointing difficulty ID_{max} beyond which a constant CD gain becomes inefficient. Harder pointing tasks, i.e. with higher amplitudes or smaller targets, require techniques that vary their gain, allowing faster movement for distant targets and more precise movements for small targets. In the next section we describe and analyze two families of techniques that provide controllable ranges of CD gains.

4. THEORETICAL MODELS FOR DIFFICULT MID-AIR POINTING TASKS

We established in section 3.1 that we needed to be able to point at targets as small as 4mm from a distance on a 5 meter wide ultra-wall, resulting in an index of difficulty of about 9.3 bits for an amplitude of two thirds of the width of the display. In the remainder of this paper, we call pointing tasks with an ID larger than 9 bits "difficult pointing tasks". We showed in Section 3.3 that apart from tablets, which must be operated with two hands, current input devices cannot reach this level of difficulty using a constant CD gain without clutching. We now explore techniques with varying CD gains.

The literature describes two main ways to increase the range of CD gains of target-agnostic pointing techniques:

[11] mdeg stands for millidegrees, PPD for Points Per Degrees. The values were computed using a 10-camera VICON system.

— *Pointer Acceleration* functions ('hydromatic-glide' in [Jellinek and Card 1990]) dynamically adjust the CD gain according to the velocity of the user's movements, based on the principle that the faster the input movement, the further away the cursor should go. Most operating systems use Pointer Acceleration, both for trackpads and mice. [Casiez and Roussel 2011] reverse-engineered the corresponding transfer functions so that HCI researchers can simulate them accurately. However, there is very little literature about how to design such functions, especially for highly demanding platforms such as ultra-walls.

— *Dual-Precision* techniques provide two pointing modes: a *Coarse* mode to cover large distances with fast movements, and a *Precise* mode to acquire small targets at a short distance. Mode switching is triggered by a user action such as depressing a button [Kopper et al. 2010], changing the hand posture [Vogel and Balakrishnan 2005], or tapping vs dragging on a touch surface [McCallum and Irani 2009]. *Coarse* modes are usually implemented with absolute control and *Precise* modes with relative control. This is similar to the 'manual powermouse' described in [Jellinek and Card 1990].

The main difference between these two approaches is the transition between low and high CD gains. Pointer Acceleration features a continuous, implicit transition based on input movement properties, while Dual-Precision techniques use a discrete mode switch that is explicitly triggered by the user. [Gallo and Minutolo 2012] introduced a hybrid approach that uses two pointing modes, one absolute and one relative, with an implicit, gradual transition based on input velocity similar to the 'automatic powermouse' [Jellinek and Card 1990]). They describe a detailed method to calibrate the corresponding transfer function, but we will show that this function does not meet the constraints of ultra-walls.

We are also interested in pointing techniques that use minimal input resources, so that users can hold other input devices or use the one used for pointing for other purposes. We explored two approaches: unimanual techniques that let users hold other, non-pointing devices in their non-dominant hand if necessary, and touch-based techniques that use a small portion of the touch screen to keep more space for additional widgets.

We first introduce a formal definition and a framework for Dual-Precision techniques and use the formulae from the previous section to define a calibration method for their CD gains and feedback. Then we describe a new transfer function for Pointer Acceleration and describe a simple method to calibrate it. Note that we use the classic formulation of Fitts' law as well as Cartesian coordinates, even though we later apply the method to angular-based techniques for which the models in [Kopper et al. 2010] and [Jota et al. 2010] might provide a better fit. Our goal is to devise general methods that let us compare results directly, with a common definition of task difficulty. Refining these methods for specific types of input is left for future work.

4.1. Dual-Precision pointing techniques

The first phase of a pointing task, especially with large amplitude, is a ballistic phase [Meyer et al. 1988] in which users control the direction and velocity of the cursor but only have limited control over its location. The second phase of a pointing task consists of bringing the cursor within the target and may require high precision when the target is small, thus a finer control over the cursor location and a lower velocity. These two phases require users to focus on two different trade-offs between speed and precision, and the transition between them can be challenging with difficult pointing tasks.

A pointing technique should adapt to the hardware and performance requirements of a task. In particular, it should provide fast cursor movements for large amplitudes and precise control for small targets[12]. While these two aspects are well covered in the literature about pointing techniques in desktop environments, the analysis of the previous section and the results presented in [Nancel et al. 2011] show that existing techniques hit a performance ceiling with high Fitts' IDs such as those required on ultra-walls.

Based on existing designs [Forlines et al. 2006; McCallum and Irani 2009; Vogel and Balakrishnan 2005], we formally define Dual-Precision pointing techniques as techniques that explicitly divide pointing tasks into two phases of different velocities:

— a coarse phase in which users quickly approach the target by traversing most of the task amplitude with no or minimal clutching,

— a precise phase in which they can acquire nearby targets as small as the smallest visible size.

Dual-precision pointing techniques therefore feature two modes, *Coarse* and *Precise*, with *explicit* mode switches.

[12]Other aspects can be considered, such as how a given technique allows accurate steering. Here we only consider pointing difficulty as modeled by Fitts' ID.

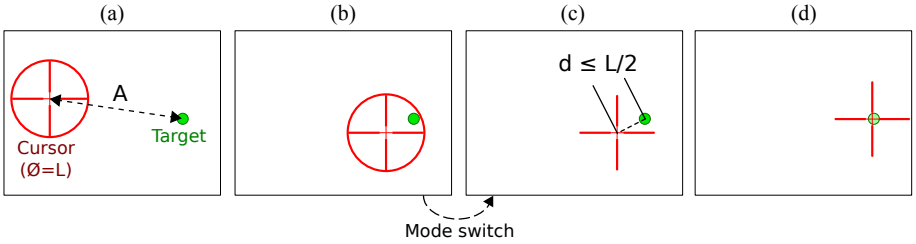


Fig. 4: A Dual-Precision technique: In *Coarse* mode the cursor features a circle indicating the range of the precise mode L (a). The user brings the area cursor over the target (b), i.e. at a distance $d \leq \frac{L}{2}$, then switches to *Precise* mode (c). She finishes the task in *Precise* mode (d).

Coarse and *Precise* modes can each be absolute or relative. Absolute controls are clutch-less and straightforward to use because the previous cursor location does not impact the input movement, i.e. the cursor can be teleported; however they have fixed gains and thus limited precision (see Section 3.2). Pointer Acceleration transfer functions and broader ranges of CD gains can be applied to relative controls, but an offset between the input and cursor locations is likely to build up, leading to clutch actions. Most Dual-Precision techniques in the literature use absolute control for *Coarse* pointing and relative control for *Precise* pointing.

Note that any technique featuring more than one precision mode, such as Smoothed Pointing [Gallo and Minutolo 2012], could be categorized as having double or multiple precision. However, consistent with previous work [Nancel et al. 2013; Nancel et al. 2011], we define Dual-Precision pointing techniques as having an explicit mode switch (Smoothed Pointing's is implicit) and supporting the precision of the smallest targets of a given application.

4.1.1. Theoretical analysis of dual-precision techniques. The design of Dual-Precision techniques relies on four key parameters: the transfer functions used in each phase, the point at which the user is expected to switch modes and the mode switch mechanism.

In order to be optimal, the mode switch should occur when the target is within range of the *Precise* mode, a distance we denote L. Switching mode outside this range causes either clutching or switching back to *Coarse* mode to bring the cursor closer to the target before switching again. With practice, users can learn this distance [McCallum and Irani 2009]. However we propose to visually represent this limited operating range by surrounding the cursor with a circle of diameter L when in *Coarse* pointing mode (Fig. 4-a) to help novice users familiarize with this distance. The coarse pointing phase can then be seen as an area cursor pointing task [Kabbash and Buxton 1995] that consists in bringing the cursor's circle over the target (Fig. 4-b), while the precise pointing phase is a regular target acquisition task with a distance d lower or equal to $L/2$ (Fig. 4-c,d).

Dual-Precision pointing techniques separate difficult pointing tasks into two pointing actions of lower difficulties. As mentioned before, Fitts' law is expressed as a linear model of an Index of Difficulty (Fitts' ID) computed from the task width (W) and amplitude (A):

$$MT = a + b \times log_2(1 + A/W) \tag{13}$$

In our case, the *Coarse* phase corresponds to bringing the cursor close enough to the target in order to be able to use the *Precise* mode. The corresponding index of difficulty is

$$ID_C = log_2(1 + A/L)$$

Similarly, the *Precise* phase consists in bringing the cursor from the location of the mode switch to the target, thus moving a distance d supposedly lower or equal than $L/2$. The corresponding index of difficulty is

$$ID_P = log_2(1 + d/W) \tag{14}$$

We cannot predict the value of d, so we use $L/2$ as a higher bound to compute the Fitts' ID of the *Precise* phase:

$$ID_P = log_2(1 + L/2W) \tag{15}$$

Considered separately, the movement times of the two modes can be expressed as follows (the subscripts C and P indicate values for the *Coarse* and *Precise* modes, respectively):

$$MT_C = a_C + b_C \times \log_2\left(1+\frac{A}{L}\right)$$
$$MT_P = a_P + b_P \times \log_2\left(1+\frac{L}{2W}\right)$$

The total movement time can be modeled as the sum of the movement times of the *Coarse* and *Precise* phases plus an optional time a_S for switching mode:

$$\begin{aligned} MT &= a_S + MT_C + MT_P \\ &= a + b_C \times \log_2\left(1+\frac{A}{L}\right) + b_P \times \log_2\left(1+\frac{L}{2W}\right) \\ \text{with } a &= a_S + a_C + a_P \end{aligned} \tag{16}$$

This formulation corresponds to Eq. (11) in [Shoemaker et al. 2012], with L and $L/2$ as the nominal values A_0 and W_0 and the "Shannon-like 'plus one' term".

The optimal value for L in Eq. (16) is the value that minimizes MT. Solving for L, the single positive value that minimizes MT is:

$$\begin{aligned} L &= \frac{\sqrt{8b_Cb_PAW + (Ab_C - Ab_P)^2} + Ab_C - Ab_P}{2b_P} \\ &= \frac{\sqrt{A(A(r-1)^2 + 8rW)} + A(1-r)}{2r} \\ \text{with } r &= \frac{b_P}{b_C} \end{aligned} \tag{17}$$

We are interested in target-agnostic techniques, so we do not know A and W for each target. However if we use the worst-case values from [Casiez et al. 2008]'s formulae ($W = W_{min}$ and $A = A_{max}$), Eq. (17) becomes:

$$L = \frac{\sqrt{A_{max}(A_{max}(r-1)^2 + 8rW_{min})} + A_{max}(1-r)}{2r} \tag{18}$$

The parameter r is the ratio of b_C and b_P, which are the multiplicative inverse of the throughput in *Coarse* and *Precise* mode. Throughput depends on a number of factors such as the user, the pointing devices, the pointing techniques, etc. If b_C and b_P can be estimated, Eq. (18) can be used. Otherwise, it is reasonable to think that the throughput should be about the same in both phases and we can simplify the computation by taking $r = 1$. The resulting optimal value for L is then:

$$L^* = \sqrt{2A_{max} \times W_{min}} \tag{19}$$

We can now compute the bounds of L for constant CD-gain modes under the above assumptions.

(i) The coarse phase of a pointing task consists in reaching a target of width L at a distance A. Casiez et al.'s (modified) equations for $CD_{min,C}$ and $CD_{max,C}$ are:

$$CD_{min,C} = \frac{A_{max}}{(c_C+1)OR_C} \qquad CD_{max,C} = \min(CD_{qmax,C}, CD_{lmax,C})$$
$$CD_{qmax,C} = \frac{L}{PS_{device,C}} \qquad CD_{lmax,C} = \frac{L}{Res_{hand,C}}$$

We want $CD_{min,C}$ to be lower than $CD_{max,C}$:

$$\begin{aligned} & CD_{min,C} < CD_{max,C} \\ \iff \quad & \frac{A_{max}}{(c_C+1)OR_C} < \frac{L}{\max(PS_{device,C}, Res_{hand,C})} \\ \iff \quad & L > A_{max}\frac{PS_{input,C}}{(c_C+1)OR_C} \text{ , with } PS_{input,C} = \max(Res_{hand,C}, PS_{device,C}) \end{aligned} \tag{20}$$

This lower bound ensures that L is larger than the smallest cursor movement in *Coarse* mode, and that the user can cover a distance of at least A_{max} with at most c_C clutches in *Coarse* mode. If L is smaller than this bound, users may have to either clutch more to reach a target further than A_{max} in *Coarse* mode, or move the cursor by more than $L/2$ in *Precise* mode.

(ii) The precise phase of a pointing task consists in reaching a target of width W from a distance supposedly smaller than $L/2$. Casiez et al.'s (modified) equations for $CD_{min,P}$ and $CD_{max,P}$ are:

$$\begin{aligned} CD_{min,P} &= \frac{L}{2(c_P+1)OR_P} & CD_{max,P} &= \min(CD_{qmax,P}, CD_{lmax,P}) \\ CD_{qmax,P} &= \frac{W_{min}}{PS_{device,P}} & CD_{lmax,P} &= \frac{W_{min}}{Res_{hand,P}} \end{aligned} \tag{21}$$

We want $CD_{min,P}$ to be lower than $CD_{max,P}$, i.e.:

$$\begin{aligned} & CD_{min,P} < CD_{max,P} \\ \iff \quad & \frac{L}{2(c_P+1)OR_P} < \frac{W_{min}}{\max(Res_{hand,P}, PS_{device,P})} \\ \iff \quad & L < W_{min}\frac{2(c_P+1)OR_P}{PS_{input,P}} \text{ , with } PS_{input,P} = \max(Res_{hand,P}, PS_{device,P}) \end{aligned} \tag{22}$$

This higher bound maps the smallest input movement of the *Precise* mode ($PS_{input,P}$) to the smallest target size (W_{min}). It ensures that, in *Precise* mode, a user can reach a target as small as W_{min} within an area of diameter L with c_P or fewer clutches. If L is greater than this bound, users may have to clutch more to reach the target, or the target may be too small to be reachable.

By combining Eq. (13), (20) and (22) we can compute ID_{max}, the Fitts' index of difficulty above which, in theory, a Dual-Precision technique cannot be used with fewer than c_C clutches in *Coarse* mode and fewer than c_P clutches in *Precise* mode:

$$\begin{aligned} & A_{max}\frac{PS_{input,C}}{(c_C+1)OR_C} < L < W_{min}\frac{2(c_P+1)OR_P}{PS_{input,P}} \\ \iff \quad & \frac{A_{max}}{W_{min}} < \frac{2(c_C+1)(c_P+1)OR_C \times OR_P}{PS_{input,C} \times PS_{input,P}} \\ \iff \quad & ID_{max} = \log_2\left(1 + \frac{2(c_C+1)(c_P+1)OR_C \times OR_P}{PS_{input,C} \times PS_{input,P}}\right) \end{aligned} \tag{23}$$

Again, the *max* subscript in ID_{max} represents the maximum index of difficulty for the *technique* considered, while in A_{max} it represents the estimated largest amplitude of the *pointing task*. Note also that Eq. (20) and (22) are applicable only if the corresponding mode is relative, since they are inferred from CD gain formulae. However, if we consider that absolute control corresponds to a fixed CD gain, e.g. 1 for ray-casting or A_{max}/OR for touch input, and an operating range equal to the size of the display, we can still compute ID_{max} using Eq. (23).

4.1.2. Summary of analysis. For a given technique, Eq. (19), (20) and (22) provide the key parameters while Eq. (23) lets us check that the dual-precision technique meets the constraints of the task.

Note that Eq. (18) only considers the parameters of the task (W_{min} and A_{max}) and the ratio of throughput between *Coarse* and *Precise* modes (r). It does not take into account the characteristics of the input system such as the resolution and range of the user and input device. In particular, values of r significantly different

from 1 may lead to unrealistic values for L. In such cases, Eq. (20) and (22) provide bounds for the value of L that ensure that the technique's expected *Precise* amplitude remains realistic. Note also that Eq. (20), (22) and (23) were based on [Casiez et al. 2008]'s formulae for constant-CD gain techniques and therefore apply primarily to pointing modes using a constant gain.

4.2. Control-Display Transfer Functions

We now turn to the study of variable-gain transfer functions that can work for ultra-walls. Pointer Acceleration consists in applying a transfer function to the Control-Display gain based on the dynamics of the users' movements. As mentioned earlier, optimizing the transfer function that controls Pointer Acceleration for ultra-walls is challenging and previous literature on position-based pointing is scarce[13].

The transfer functions used in major desktop operating systems [Casiez and Roussel 2011] work for single- and multi-monitor display configurations but are not adapted to ultra-walls, which typically have a 200 to 400 inches diagonal and a resolution of, e.g., 20 480 $\times$ 6 400 for our evaluation platform (Fig. 1). Beyond recent work by Casiez *et al.* that provides a general framework but does not address such contexts [Casiez and Roussel 2011; Casiez et al. 2008], the only documented calibration methods are those related to PRISM [Frees et al. 2007] and its refinements [König et al. 2009; Gallo and Minutolo 2012]. These methods, however, were designed to calibrate absolute-to-relative transfer functions for pointing techniques that feature an implicit mode switch between absolute and relative pointing. While we later compare our techniques with those, this work is of little use to calibrate purely relative transfer functions. Finally, the one-dimensional transfer function described in [Cockburn et al. 2012] that adapts to the maximum task amplitude was designed for scrolling long documents but was never evaluated with pointing tasks.

In our context, Pointer Acceleration must be adapted to enable relocation of the cursor across the display (corresponding to amplitudes of 5 meters and more) at high input speeds with minimal clutching. This requires a high CD gain. Conversely, CD gain must be set to a low value at low input speeds so as to enable high-precision cursor control.

We use sigmoid transfer functions similar to those used for rate-based pointing [Rutledge and Selker 1990] and in some operating systems [Casiez et al. 2008; Casiez and Roussel 2011]. They are characterized by a slope that smoothly gets steeper before flattening again, as illustrated in Fig. 5. The flatter slope at the left end enables higher precision at low input velocities. The flatter slope at the right end limits maximum cursor speed.

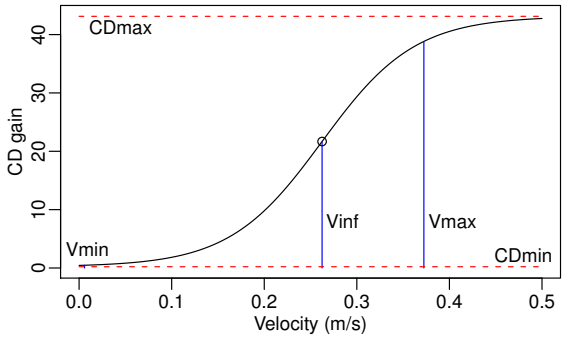


Fig. 5: An example of logistic sigmoid curve (used in condition *RelaLarge* of the experiment reported in Section 7). In this example V_{min}= 6 mm/s, V_{max}= 37 cm/s, $CD_{min} = 0.22$, $CD_{max} = 43.1$, $\lambda = 0.02$ s/mm and $ratio_{inf}$= 0.5.

To model such curves, we use a simple form of the generalized logistic function:

$$CD(x) = \frac{CD_{max} - CD_{min}}{1 + e^{-\lambda(x - V_{inf})}} + CD_{min} \tag{24}$$

$$\text{where } V_{inf} = ratio_{inf}.(V_{max} - V_{min}) + V_{min}.$$

This function can be tuned with six parameters:

[13]A recent study from [Roussel et al. 2012] partially addresses this issue, but was published after the work described here was conducted.

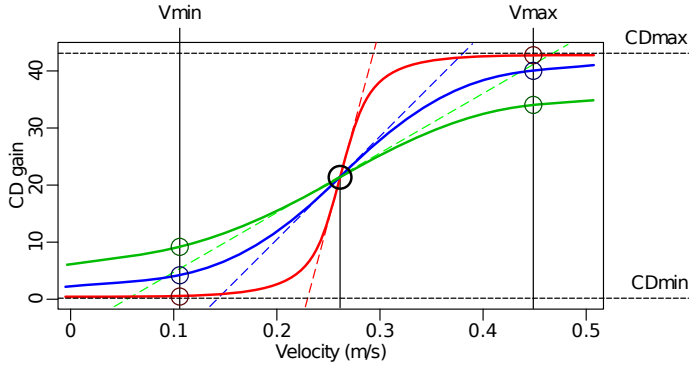


Fig. 6: Examples of curves with the same values for V_{min}, V_{max}, $ratio_{inf}$, CD_{min} and CD_{max}, and a varying λ. The red, blue and green curves correspond to decreasing values of λ. The values of the function at the ends of the input velocity range (small circles) illustrate the fact that lower values of λ reduce the effective output range relatively to the expected output range $[CD_{min} - CD_{max}]$.

— V_{min} and V_{max} are estimates of the lower and upper bounds of input velocity beyond which accurate control becomes difficult. The input function is not constrained to this velocity range: some very fast or very precise movements may exceed these bounds.

— $ratio_{inf}$ controls the position of the inflexion point within this range: 0 sets the inflexion point at V_{min}, 1 sets it at V_{max} and 0.5 sets it in between. We use a ratio rather than an absolute value because it is then independent from the velocity range. V_{inf} is the velocity at the inflextion point.

— CD_{max} and CD_{min} are asymptotic values that bound the output range of the function. They depend on the task (A_{max} and W_{min}) and can be adjusted according to users' motor capabilities.

— λ is related to the slope of the curve at its inflexion point, defined as:

$$CD'(V_{inf}) = \frac{\lambda \times (CD_{max} - CD_{min})}{4} \tag{25}$$

$\lambda = 0$ yields a constant function, $\lambda = +\infty$ a step function. λ is expressed in seconds per input units.

λ is a trade-off between the smoothness of the curve and how close to CD_{min} and CD_{max} the resulting transfer function can go for input velocities in the $[V_{min};V_{max}]$ range. Low values (Fig. 6, green) make for smoother curves but CD_{min} and CD_{max} are harder to reach. High values (Fig. 6, red) increase the effective output range of the function but make it steeper, and thus possibly harder to control around the inflexion point.

We designed the following procedure to calibrate these parameters:

(1) V_{min} and V_{max} are set respectively to the 90^{th} percentile and the median of two corpora of velocities corresponding to voluntarily slow and fast movements collected using the input device[14];

(2) CD_{min} is set to the ratio of the minimum target size to the smallest input movement amplitude ($\min_{input}$) that is considered usable for selecting a target[15]: $CD_{min} = \frac{W_{min}}{\min_{input}}$.
The initial value for CD_{max} is set to the ratio of the maximum pointing amplitude to the corresponding dimension of the dedicated input zone, scaled by the maximum number of pointing movements $(c+1)$ where c is the acceptable number of clutches. Maximum amplitude is A_{max} for mode-less techniques and *Coarse* modes, or L for *Precise* modes. The dimension of the dedicated input zone $\dim_{input}$ is its width for a horizontal movement or its height for a vertical movement, provided that the input zone has the same aspect-ratio than the display.

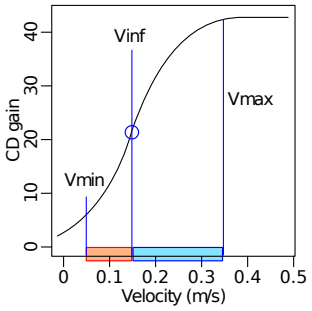
$$CD_{max} = \frac{A_{max} \text{ or } L}{(c+1)\dim_{input}}$$

In the most constrained cases where the curve remains too steep for accurate control after steps 3 and 4, the output CD-gain range must be reduced. Increasing CD_{min} makes smaller targets much harder to select and potentially unreachable. Reducing CD_{max} increases the number of clutches, but effectively

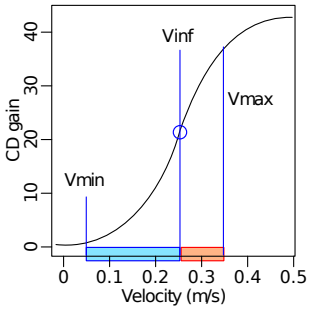
[14]Note that previous work such as [Aceituno et al. 2013] investigate the *limits* to human motor control, while this step defines a comfortable control range.
[15]Note that this method enables subpixel pointing [Roussel et al. 2012] by setting W_{min} to less than the size of one pixel.

reduces the output range of the function. This makes it easier to smoothen the curve (see step 4) but lowers the cursor's maximum velocity.

(3) $ratio_{inf}$ defines the relative position of the inflexion point. As described earlier, fast input motions indicate the user's intention to move the cursor over a long distance while slow input motions indicate the user's intention to perform short-range, precise movements. $ratio_{inf}$ defines the location of the transition between these two behaviors within the input velocity range. $ratio_{inf}$ should initially be set to .5 then adjusted if a different setting improves the trade-off between the two types of control:



Smaller values make the cursor accelerate early, i.e. a large section of the input velocity range (right part, in blue) is dedicated to fast cursor movements, making them more controllable than precise ones; users must therefore be able to finely control their slowest input movements (red section) to select small targets.



Larger values make the cursor accelerate later, i.e. a larger section of the input velocity range (left part, in blue) is dedicated to precise cursor movements, making them more controllable than fast ones; users must therefore be able to perform movements fast enough (red) to travel through large amplitudes.

Note that $ratio_{inf}$ can be negative or greater than 1 if the chosen input range $[V_{min}; V_{max}]$ does not accurately describe the effective velocity range that the users can cover.

(4) λ is directly proportional to the curve's maximum steepness (Eq. (25), Fig. 6). In order to avoid steep changes in velocity, λ should be set to the smallest value while still covering a sufficient part of the CD-gain output range to perform the most difficult pointing tasks with minimum clutching. A conservative initial value for λ is computed from the slope of the line that goes through (V_{min}, CD_{min}) and (V_{max}, CD_{max}), i.e. $\lambda_0 = \frac{4}{V_{max}\text{-}V_{min}}$. This value ensures that 76 % of the output range $[CD_{min}, CD_{max}]$ is reachable from the input range $[V_{min}, V_{max}]$ with minimal steepness. It proved to be a safe lower bound to calibrate the techniques presented later in this paper.

Step 1 in the above procedure results in indicative values for V_{min} and V_{max} that serve as a starting point for the calibration of the other parameters. V_{min} and V_{max} should not be changed after step 1: although V_{inf} is initially defined within this input range, it is used in Eq. 24 as an absolute value, so V_{min} and V_{max} do not actually affect the transfer function once a suitable $ratio_{inf}$ is chosen.

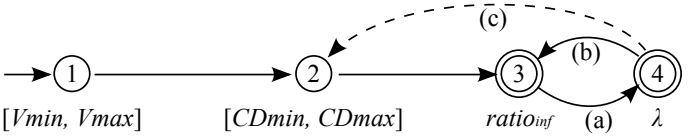


Fig. 7: State machine diagram of the calibration process described above. States correspond to the steps of the calibration process. Step 1 is the initial state, steps 3 and 4 are the two potential final states. Transition (c) occurs when no modification of the couple $\{ratio_{inf}, lambda\}$ provides a satisfying pointing behavior.

A typical calibration process (Fig. 7) consists of performing steps 1 to 4, possibly cycling between steps 3 and 4 until a satisfying behavior is found. If no satisfying values can be found for $ratio_{inf}$ and λ, the output range can be relaxed (transition (c) in Fig 7), e.g. by allowing more clutches or by increasing the minimal target width if the initial W_{min} was loo low. The tuning process is considered over when (1) users can perform the most difficult task (largest amplitude, smallest target width) and (2) no further parameter tweaking can improve pointing performance or the users' subjective perception of smoothness and precision.

This procedure is the result of an extensive iterative design process. We used it successfully to calibrate all the pointing (non-constant) transfer functions for the techniques evaluated in this paper. It was also used successfully for pointing with a device combining elecromyographic and inertial sensors [Haque et al. 2015]. The Smarties toolkit [Chapuis et al. 2014] also uses this transfer function for pointing on wall-display with touch devices, allowing developers to fine tune its parameters. However we leave the formal comparison of this procedure with other calibration methods to future work.

4.3. Validating our theoretical analyses

We presented the two main strategies used in the literature to augment the range of available Control-Display gains for difficult pointing tasks: dual-precision techniques that switch explicitly between a *Coarse* and a *Precise* mode, and Pointer Acceleration transfer functions that adapt the CD gain to the velocity of the input movement. Note that these two approaches are not mutually exclusive: a dual-precision technique could feature Pointer Acceleration functions in its modes. We also presented formulae and methods to calibrate the CD gains of pointing techniques that use these strategies.

The next three sections report on the design and evaluation of pointing techniques that were calibrated with these methods. Considering the large design space for pointing techniques in mid-air, we chose to organize our experimental work according to external interaction constraints, i.e. which other input channel and interaction techniques could be required in a real-world use of our techniques: (i) pointing techniques that require only one hand, leaving the non-dominant hand free for other purposes such as holding a secondary device or inputting gestures; and (ii) dual-precision pointing techniques that require only small areas of a touch-enabled surface, leaving as much of the interactive surface as possible free for other touch-based interactions. The third experiment compares Pointer Acceleration-only techniques that use a small area of a touch-enabled surface with the best techniques of the previous experiments and of the literature.

5. STUDY 1: SINGLE- VS. DUAL-PRECISION UNIMANUAL TECHNIQUES

In this first experiment, we focus on techniques that can be operated with a single hand (the dominant one), in order to allow the other hand to hold and control other input devices. Following the two families of techniques introduced in the previous section, we designed a Pointer Acceleration technique and explored three Dual-Precision pointing techniques.

5.1. Techniques

GyroAcc is a single-mode Pointer Acceleration technique that applies a custom-made transfer function to the rotation of a hand-held wireless mouse to control cursor translation: yaw is mapped to horizontal cursor movements and pitch to vertical cursor movements. The transfer function, detailed later, was calibrated using the procedure described in Section 4.2. The user can click with the left button of the wireless mouse and freeze the cursor with the right button to adjust the offset between cursor position and limb orientation, also known as *clutching*.

We designed three Dual-Precision techniques: *Laser+Position* (Fig. 8), *Laser+Gyro* (Fig. 9) and *Laser+Track* (Fig. 10). All three techniques use *RayCasting* as their *Coarse* mode, which is intuitive, fast, and does not require clutching.

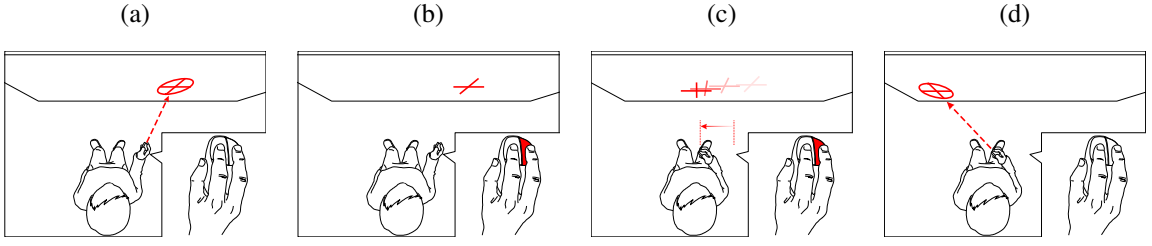


Fig. 8: *Laser+Position. RayCasting* for coarse pointing (a). Pressing a button switches to *Precise* mode (b). Relative translational movements control cursor movements (c). Releasing the button switches back to *Coarse* mode (d). Clicking with the left button selects the target.

Laser+Position (Fig. 8) is an adaptation of Vogel's free-hand *RayToRelative* technique [Vogel and Balakrishnan 2005]. Instead of detecting hand gestures for mode switching and clicking, we use the buttons of a wireless mouse to make recognition more robust. *Laser+Position* combines *RayCasting* for coarse pointing

and relative translational movements of the hand/device for precise pointing. In *Precise* mode, the hand's translation is taken into account in a plane orthogonal to the orientation of the hand-held device when switching mode (Fig. 8-b). *Precise* pointing is activated by keeping a button depressed. A second button is used for clicking. Users can clutch in *Precise* mode by releasing the first button and repositioning their hand quickly: If they press the first button again within less than 600 ms (tuned through pilot testing), the technique doesn't switch back to *Coarse* mode while the button is up. Informal testing showed that an operating range of 300 mm for the *Precise* mode was large enough without causing too much fatigue. The theoretical limit of difficulty (Eq. (23)) for *Laser+Position* is approximately 18.2 bits, i.e., much higher than needed for a task that does not involve zooming or lenses ($A/W \approx 300,000$).

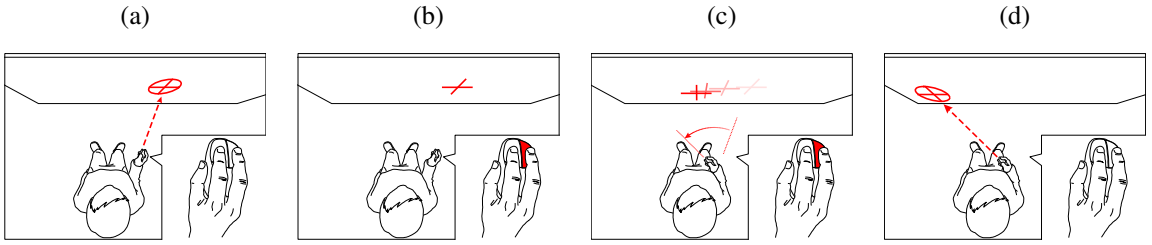


Fig. 9: *Laser+Gyro*. *RayCasting* for coarse pointing (a). Pressing a button switches to *Precise* mode (b). Relative rotational movements control cursor movements (c). Releasing the button switches back to *Coarse* mode (d). Clicking with the left button selects the target.

Laser+Gyro (Fig. 9) combines *RayCasting* for coarse pointing and relative rotational movements for precise pointing. This is somewhat similar to the ARM technique [Kopper et al. 2008; Kopper et al. 2010]. However, the 'precision mode' of ARM is perspective-dependent, since the lower gain is applied to the relative movement of the point at the intersection of the ray and the display surface (as in Smoothed Pointing [Gallo and Minutolo 2012]). In addition, ARM requires both hands. Compared to *Laser+Position*, which mainly involves upper limb segments (forearm to shoulder) in *Precise* mode, *Laser+Gyro* mainly involves the wrist and/or elbow and is potentially less tiring. Clutching, clicking and mode switching are identical to the *Laser+Position* technique. Our tests showed that an operating range of π rad was large enough for the *Precise* mode while not causing too much fatigue. The theoretical limit of difficulty (Eq. (23)) for *Laser+Gyro* is approximately 19 bits.

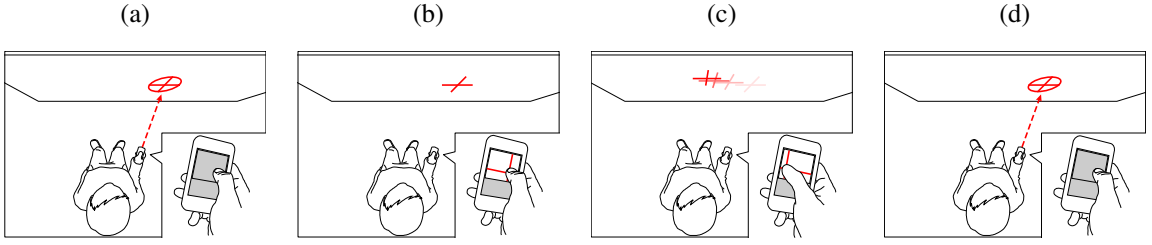


Fig. 10: *Laser+Track*. *RayCasting* for coarse pointing (a). Touching the surface switches to *Precise* mode (b). Moving the thumb over the surface controls cursor movements (c). Lifting the thumb switches back to *Coarse* mode (d). Tapping the bottom part of the screen selects the target (Fig.11).

Laser+Track (Fig. 10) combines *RayCasting* for coarse pointing and relative translational movements of the thumb on a touch-sensitive surface (PDA, smartphone, etc.) for precise pointing. The surface is divided into two zones (Fig. 11): an upper zone (1) for tracking and a lower (smaller) zone (2) for clicking. Switching between the two zones can be achieved easily using proprioceptive information and, after a short learning phase, does not require the user to look at the device. Touching the upper zone switches to *Precise* mode. Switching back to *Coarse* mode only happens 300 ms (tuned through pilot testing) after the thumb has been released, thus enabling clutching. The 300 ms clutching timer is reset each time a click occurs in the lower zone so that users can stay in *Precise* mode if the click was a miss by going back to the upper zone. To compensate for unintended finger movements at release time, we retrieve the coordinates of the click 200 ms before the finger-up event. The theoretical limit of difficulty (Eq. (23)) for *Laser+Track* is approximately 18 bits.

For all three Dual-Precision techniques, the cursor is a crosshair surrounded by a circle (Fig. 8-10). The circle's diameter is equal to the value of L (Eq. (17) for that technique. In *Precise* mode, the circle is decoupled from the crosshair and displayed as a ghost at the position where the cursor will be when the user

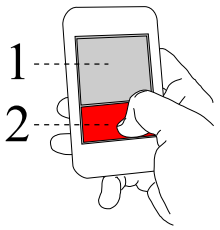


Fig. 11: Zones used in the *Laser+Track* technique: upper zone (1) is for relative pointer movements and lower zone (2) is for clicking.

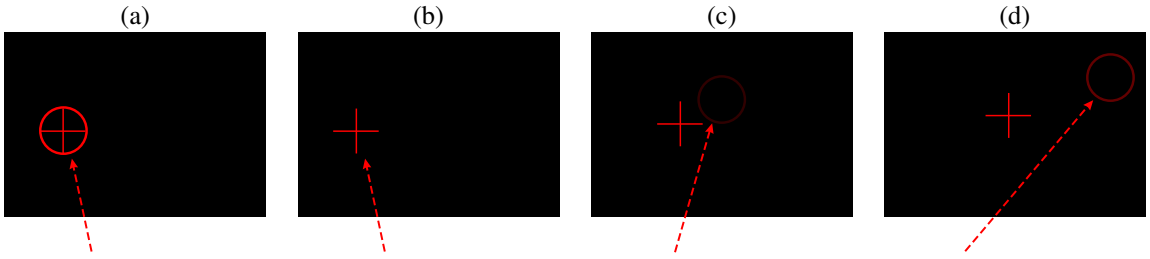


Fig. 12: The dual-precision techniques cursor (a). The circle disappears when the user switches to *Precise* mode (b). In *Precise* mode (b to d), the circle is decoupled from the crosshair to indicate where the *Coarse* mode would be pointing. Opacity increases with distance to the crosshair (c and d).

switches back to *Coarse* mode (Fig. 12). This is because *Coarse* mode is absolute and the cursor jumps when transitioning back from *Precise* mode. The opacity of the ghost is inversely proportional to its distance to the crosshair so as to minimize visual interference, with a maximum value of 25%. A similar double cursor mechanism featuring a more elaborate opacity mechanism has been used in recent work by [Debarba et al. 2012].

All Dual-Precision techniques support clutching in their *Precise* mode (Fig. 13). When the user releases the button or finger that controls *Precise* mode, an animation shows how much time is left until the technique switches to *Coarse* mode: four short strokes perpendicular to the cross' branches move from the center to the end of the cross (Fig. 13.b). The user can use this time to clutch and continue moving in *Precise* mode. When the four strokes reach the end of the cross, clutching times out and the technique switches to *Coarse* mode.

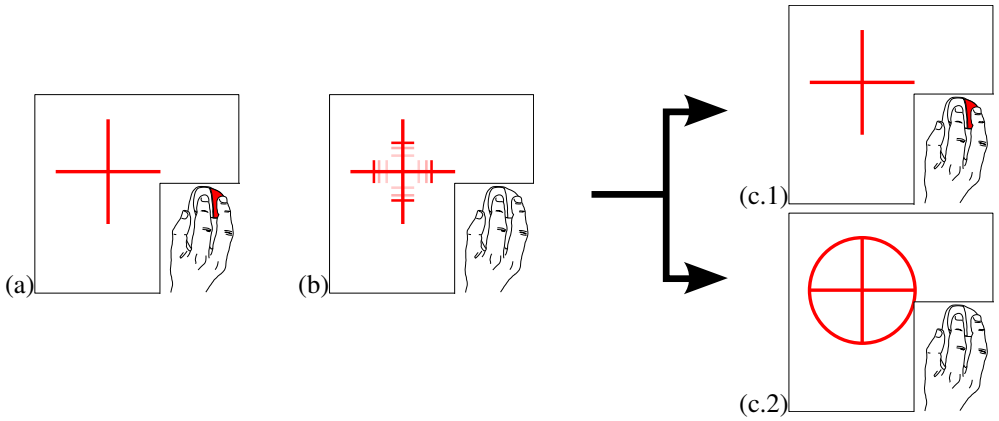


Fig. 13: Visual feedback of the remaining time before a release event triggers a switch from *Precise* to *Coarse* mode. *Precise* mode is active while the user presses a button (a), or the touch surface for *Laser+Track*. When the user lifts the finger (b), moving lines appear on the branches of the cross to indicate the delay. If the user presses the button or touch surface again before the moving lines reach the end of the cross (c.1), the lines disappear and the technique stays in *Precise* mode, effectively allowing the user to clutch. If the lines do reach the end of the cross, the technique switches back to *Coarse* mode (c.2).

5.2. Participants

Twelve unpaid volunteers (2 female), age 24 to 43 (mean 29.5, std dev. 5.76), all right-handed and with normal or corrected-to-normal vision, served in the experiment. All participants were familiar with remote interaction, having previously used at least a WiiMote.

5.3. Apparatus

The experiment was conducted on the WILD ultra-wall [Beaudouin-Lafon et al. 2012] and implemented with jBricks [Pietriga et al. 2011]. The display (Fig. 1) consists of 32 high-resolution 30" screens laid out in an 8×4 matrix (Fig. 1). It is 5.5 meters wide and 1.8 meters high and can display 20480×6400 pixels. A cluster of 16 computers, each with two high-speed nVidia 8800GT graphics cards, communicate via a dedicated high-speed network through a front-end computer. A VICON motion-capture system tracks passive retroreflective markers and provides 3D object coordinates with sub-millimeter accuracy at 200Hz.

RayCasting was implemented using the VICON motion tracker. Both *Laser+Gyro* and *Laser+Position* used a wireless mouse that was attached to a passive device equipped with retroreflective markers. The right button was used for mode switching and clutching, the left button for clicking. A mouse was also used for *GyroAcc*: the right button was used for clutching and the left button for clicking. *Laser+Track* used an Apple iPod Touch as a touch-sensitive surface.

The main factors were TECHNIQUE, AMPLITUDE and target WIDTH. The values of WIDTH were 30, 15, 7 and 4 mm. We checked that all participants could actually see all the targets. The values of AMPLITUDE were 2760 pixels (width of one LCD panel, 0.637 m), 8280 pixels (width of 3 LCD panels, 1.912 m) and 13 800 pixels (width of 5 LCD panels, 3.187 m). The display is 5.5 meters wide, but we wanted to avoid edge pointing effects [Appert et al. 2008] or the cursor leaving the display. The corresponding Index of Difficulty ranged from 4.47 to 9.64 bits.

According to [Casiez et al. 2008], the highest Fitts' Index of Difficulty tested in their review of the literature was 7.6 bits in a desktop context (A= 30 cm, W= 1.5 mm) and 9 bits (A= 4.5 m, W= 9 mm) for a univariate (1D) pointing task on a 25-PPI projected display. Higher difficulties, e.g. 12 bits in [Guiard et al. 2001] and up to 30 bits in [Guiard and Beaudouin-Lafon 2004], have only been studied in the context of multiscale or zoomable interfaces. To our knowledge, the experiments reported in this article are the first ones to feature an index of difficulty as high as 9.64 bits in a single-scale, bivariate task (horizontal movement with 2D circular targets).

We made no assumption about the relative throughputs of the *Coarse* and *Precise* modes ($r = 1$ in Eq. (17)). The value of L is given by Eq. (19) and only depends on task parameters (A_{max} and W_{min}). It is the same for all techniques: $L = 160$ mm, which is within the range defined by equations (20) and (22). For all *Precise* modes we checked that $CD_{min,P} < CD_{max,P}$ (Eq. 21) with no clutching ($c_P = 0$). We chose CD values as low as possible within this range to minimize the needed motor accuracy. To ensure that no clutching was necessary under normal use, we used $CD_P = 2 \times CD_{min,P}$: 3.07 for *Laser+Track*, 0.51 for *Laser+Position* and 0.89 mm/deg for *Laser+Gyro*[16]. The maximum and minimum CD gains for the transfer function of *GyroAcc* were $CD_{max} = 35.41$ mm/deg and $CD_{min} = 0.89$ mm/deg [16]. Applying these values to Equ. (24)) gives the following logistic function, with v and V_{inf} expressed in deg/s:

$$CD(v) = 0.89 + \frac{34.52}{1 + e^{-.016 \times (v - 130)}} \tag{26}$$

5.4. Task and Procedure

The task was a Fitts reciprocal pointing task [Fitts 1954]. Participants were asked to click targets located alternatively left and right from the center of the display. Targets were presented as bright green disks on a black background. When the cursor was inside the target, the target was highlighted in white. An additional, wider green circle appeared (Fig. 14-(4)) so that participants could see the feedback unambiguously even for very small targets.

Before each trial, participants had to move the cursor inside a dwell zone using ray-casting (Fig. 14-(2)) and leave it there for half a second before the target appeared. The goal was to recalibrate the relative position and orientation of the hand-held device and the cursor at each trial so that the successive offsets caused by

[16]The CD gains of *Laser+Gyro* and *GyroAcc* are ratios between angular inputs (expressed in degrees) and linear outputs (in millimeters), hence the unit.

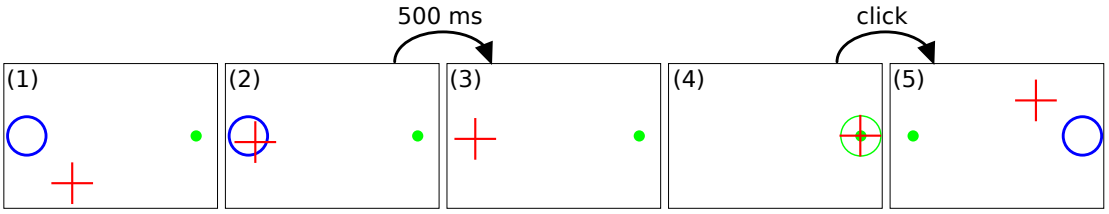


Fig. 14: (1) The target (green dot) and dwell zone (blue circle) are both visible at the beginning of the trial. Participants have to keep the cursor (red cross) in the dwell zone for 500 ms (2) before it disappears (3). Then they click the target (4) and the next, symmetric trial starts (5). The background of the experiment was black but is shown here in white for clarity.

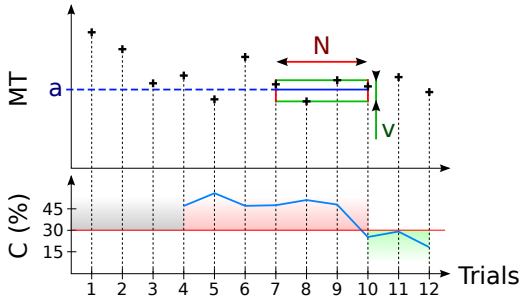


Fig. 15: Criterion for the practice blocks: a user's performance is considered stable if the ratio $C = v/a$ is smaller than 30% over the last N=4 trials. v is the variation in movement time, computed as *Pointing Time*$_{max}$ - *Pointing Time*$_{min}$ over the last N trials. a is the mean *Pointing Time* of the last N trials. In this example, trials 7 to 10 met the criterion ($C < 30\%$) for trial 10) and trials 11 and 12 were performed as additional practice requested by the participant.

the relative techniques would not accumulate, causing undesired clutching. The dwell zone was a 500-pixel-wide circular area centered on the previous target. It disappeared at the end of the dwell time, signaling the start of the trial.

Participants stood at a distance of 2 meters from the display. This distance gave participants a good overview of the display while avoiding problems of visual acuity (see Subsection 3.1).

5.5. Design

We used a $4 \times 4 \times 3$ repeated measures within-subject design with three independent variables. The factors were:

— TECHNIQUE: *GyroAcc*, *Laser+Position*, *Laser+Gyro* and *Laser+Track*;

— WIDTH: 30 mm (130 pixels), 15 mm (65 pixels), 7 mm (30 pixels), and 4 mm (18 pixels);

— AMPLITUDE: 637 mm (2 760 pixels), 1 912 mm (8 280 pixels) and 3 187 mm (13 800 pixels).

For each participant, we grouped trials into 48 blocks of 5 trials, one block per combination of TECHNIQUE, AMPLITUDE and WIDTH. The presentation order for TECHNIQUE, AMPLITUDE and WIDTH was counterbalanced across participants using a Latin square.

Each time a new TECHNIQUE began, participants practiced the new technique with AMPLITUDE = 1912 mm and WIDTH = 7 mm. Practice trials ended as soon as 1) the variation in task performance over the last four trials was less than 30% of the task time average in that window (Fig. 15) and 2) participants felt comfortable with the technique.

We evaluate new pointing techniques with unusually-high Indices of Difficulty. Some tasks might actually be too difficult to perform despite our pilot tests. Therefore, if a given target was not selected after ten seconds, the trial timed out (*TimeOut*) with a quick visual feedback to notify the participant and the next trial was started.

To summarize, we collected 4 TECHNIQUE $\times$ 4 WIDTH $\times$ 3 AMPLITUDE $\times$ 5 replications $\times$ 12 participants = 2880 trials. For each (non-*TimeOut*) trial, we collected the number clicks outside the target (*Misses*), the time until a click in the target (*Pointing Time*), the time to acquire the dwell zone (*Recalibration*), the time to enter the target (*Reaching*), the time to perform the first click (*Clicking*), the time taken to perform the first

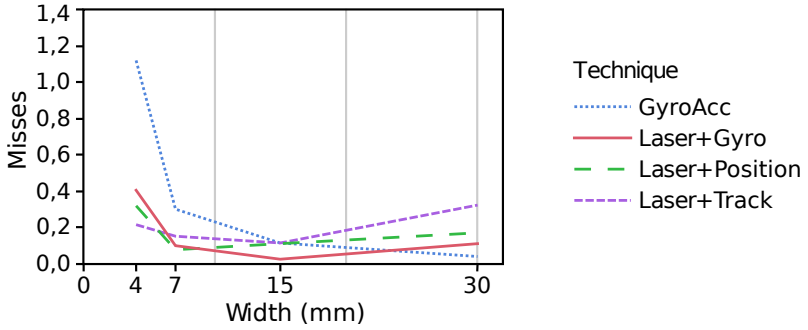


Fig. 16: *Misses* per TECHNIQUE × WIDTH.

switch to *Precise* mode (*Switch Time*), the distance from the first mode switch to the target (*MS Distance*) and the number of times the cursor left the target (*Crossings*).

At the end of the experiment, participants were asked to rank the techniques and rate them for *Mental Effort*, *Accuracy*, *Speed*, *Fatigue*, *Comfort* and *Overall Easiness* on 5-point Likert scales.

5.6. Predictions

The transfer function of *GyroAcc* supports a similar output range in terms of cursor velocity and precision as the Dual-Precision techniques. However, *GyroAcc* only allocates a portion of the velocity input range to precise pointing ($v < V_{inf}$) while Dual-Precision techniques can be "locked" in *Precise* mode, making it easier to control small scale cursor movements without being constrained by the input velocity; we thus expect *GyroAcc* to have lower performance with smaller targets than the three Dual-precision techniques (prediction *P1*).

Conversely, Dual-Precision techniques require switching mode for every pointing task whose target is too small for the *Coarse* mode, adding extra time to the pointing task, while the continuous transfer function of *GyroAcc* supports a smooth transition between coarse and precise cursor movements. However this transition becomes harder to control between very high and low input speeds, as can happen with very large amplitudes and very small targets; we thus expect *GyroAcc* to be faster than the Dual-precision techniques for smaller Indices of Difficulty (prediction *P2*).

Among the Dual-Precision techniques, we note that the *Precise* modes of *Laser+Gyro* and *Laser+Position* differ only in the angular vs. linear control. In particular, they are controlled by the same limbs (forearm, wrist and hand). We expect them to have similar performance (prediction *P3*). However, *Laser+Position* should be more tiring since translations are controlled less naturally than rotations (prediction *P4*).

Also, *Laser+Track* is the only dual-precision technique whose *Precise* mode does not require moving the hand-held device. We expect it to be faster for *Recalibration* (prediction *P5*).

Finally, the value of *L* does not depend on the target WIDTH, so the time spent in *Coarse* mode should not be affected by target WIDTH (prediction *P6*).

5.7. Results

We analyzed the data[17] using multiway ANOVAs, accounting for repeated measures using the REML procedure, and performed Tukey HSD post-hoc tests for pairwise comparisons. We used the mean for *Misses* and *Crossings*. As expected, the distributions of time measurements per condition were left-skewed, so we used the median instead of the mean. We verified that there was no significant effect of TECHNIQUE presentation order and observed that learning and fatigue effects were not significant. All reported results are significant at least at the $p < 0.001$ level unless noted otherwise.

Timeouts. 3% of the trials were *TimeOuts*. There is a significant effect on the number of *TimeOuts* for TECHNIQUE ($F_{3,33} = 13.63$, $p < 0.0001$), AMPLITUDE ($F_{2,22} = 4.67$, $p = 0.0204$), WIDTH ($F_{3,33} = 25.44$, $p < 0.0001$) and TECHNIQUE×WIDTH ($F_{9,99} = 8.44$, $p < 0.0001$). As expected, larger amplitudes and smaller widths cause more *TimeOuts*. *Laser+Track* (mean 0.06) and *GyroAcc* (0.04) cause significantly more *TimeOuts* than *Laser+Position* (0.01) and *Laser+Gyro* (0). The effect increases with smaller widths: for $Width = 4$ mm, *Laser+Track* and *GyroAcc* caused significantly more *TimeOuts* than the two other techniques.

[17] All analyses were performed with SAS JMP.

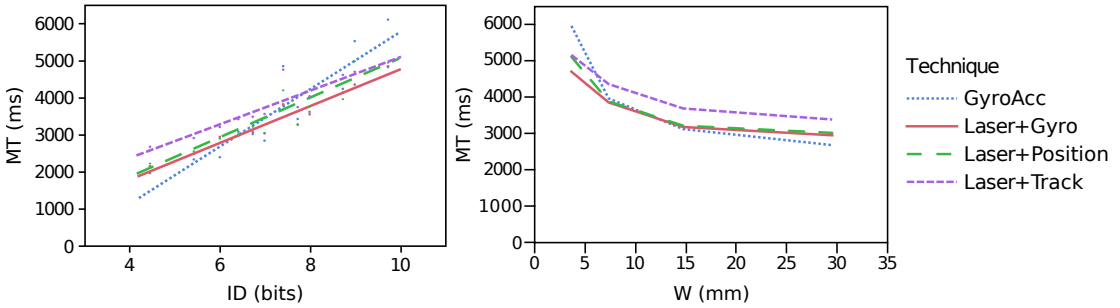


Fig. 17: Fitts' law regressions (left) and movement time per WIDTH (right) for each TECHNIQUE. Note that the abscissa on the left plot is not zero-based for better readability.

Misses. There is a significant effect on *Misses* for TECHNIQUE ($F_{3,33} = 13.19$, $p < 0.0001$), WIDTH ($F_{3,33} = 43.21$, $p < 0.0001$) and TECHNIQUE×WIDTH ($F_{9,99} = 15.47$, $p < 0.0001$). As expected, *Misses* increased with smaller widths. *GyroAcc* (mean 0.44) causes significantly more *Misses* than the other techniques (means from 0.16 to 0.26). The effect was even stronger with WIDTH = 4 mm (mean 1.36), supporting prediction *P1*.

Clutching. We found a significant effect on clutch for TECHNIQUE ($F_{3,33} = 281.01$, $p < 0.0001$), WIDTH ($F_{3,33} = 3.66$, $p = 0.0222$) and WIDTH×AMPLITUDE ($F_{6,66} = 2.24$, $p = 0.0497$). Smaller widths slightly increase the number of clutches (the only significant difference is between WIDTH = 4 and WIDTH = 30 mm). *Laser+Track* (mean 2.27) causes significantly more clutching than *Laser+Gyro* and *Laser+Position* (1.01 and 0.98) which cause significantly more clutching than *GyroAcc* (0.02).

Crossings. There is a significant effect on *Crossings* for TECHNIQUE ($F_{3,33} = 19.57$, $p < 0.0001$), AMPLITUDE ($F_{2,22} = 3.48$, $p = 0.0488$), WIDTH ($F_{3,33} = 79.66$, $p < 0.0001$) and TECHNIQUE×WIDTH ($F_{9,99} = 19.9$, $p < 0.0001$). As expected, smaller widths cause more crossings. The effect of AMPLITUDE is a bit surprising, with more *Crossings* for the medium amplitude. *GyroAcc* and *Laser+Track* (resp. 1.13 and 1.03) cause significantly more *Crossings* than *Laser+Position* and *Laser+Gyro* (resp. 0.7 and 0.61). *GyroAcc* causes almost twice as many *Crossings* than the second worst condition for WIDTH = 4 mm.

Dwell Time. There is a significant effect on *Recalibration* for TECHNIQUE ($F_{3,33} = 7.35$, $p = 0.0007$), AMPLITUDE ($F_{2,22} = 65.42$, $p < 0.0001$), WIDTH ($F_{3,33} = 10.42$, $p < 0.0001$), TECHNIQUE×AMPLITUDE ($F_{6,66} = 3.83$, $p = 0.0024$) and AMPLITUDE×WIDTH ($F_{6,66} = 3.87$, $p = 0.0023$). As expected, *Recalibration* increases with AMPLITUDE. Recalibration takes significantly more time with *GyroAcc* (mean 1722 ms) than with all other techniques (1504 ms to 1417 ms), especially for the larger AMPLITUDES.

Reaching Time. We found a significant effect on *Reaching* for TECHNIQUE ($F_{3,33} = 3.65$, $p = 0.022$), WIDTH ($F_{3,33} = 308.06$, $p < 0.0001$) and AMPLITUDE ($F_{2,22} = 134.3$, $p < 0.0001$). As expected reaching time increases with task difficulty. *Laser+Track* (mean 2710 ms) is significantly slower to reach the target than *GyroAcc* (2428 ms), with the other two techniques in between.

5.8. Pointing Time

There is a significant effect on *Pointing Time* for TECHNIQUE ($F_{3,33} = 13.09$, $p < 0.0001$), AMPLITUDE ($F_{2,22} = 71.14$, $p < 0.0001$), WIDTH ($F_{3,33} = 140.52$, $p < 0.0001$) and TECHNIQUE×WIDTH ($F_{9,99} = 8.56$, $p < 0.0001$). As expected, pointing time increases with task difficulty (Fig. 17, left). *Laser+Track* (mean 4268 ms) is significantly slower than the other techniques (means from 3519 to 3847 ms). For WIDTH = 4 mm, *GyroAcc* is the slowest, supporting prediction *P1*.

Modeling Pointing Time. As described earlier, the task of pointing with a dual-precision technique consists of two phases, coarse and precise, with a mode switch in between. We expect the movement time of each pointing phase (MT_C and MT_P) to follow Fitts' law and assume that the mode switch takes a constant time a_S dependent on the technique. We obtain the following model for the global pointing task time (MT_T):

$$MT_T = MT_C + a_S + MT_P \tag{27}$$

$$= (a_c + a_S + a_p) + b_c \; log\left(1 + \frac{A}{L}\right) + b_p \; log\left(1 + \frac{d}{W}\right) \tag{28}$$

$$= a + b_c \; ID_C + b_p \; ID_P \tag{29}$$

Using our model with $L = \sqrt{2A_{max}W_{min}}$ and $d = L/2$ (Eq. 16), we compute the following regressions and goodness of fit:

Model	Technique	Parameters	r^2	AICc
Our model	*Laser+Gyro*	$-226 + 466 \times ID_C + 633 \times ID_P$	.96	168
	Laser+Position	$-309 + 423 \times ID_C + 745 \times ID_P$	.93	178
	Laser+Track	$601 + 348 \times ID_C + 640 \times ID_P$	.92	175
Fitts' law	*Laser+Gyro*	$-205 + 496 \times ID$	.93	170
	Laser+Position	$-361 + 534 \times ID$	.86	181
	Laser+Track	$540 + 454 \times ID$	.85	178
	GyroAcc	$-1972 + 772 \times ID$	.81	

Table II: Parameters and accuracy of both Fitt's law and our model. *GyroAcc* is a single-mode technique and therefore our model does not apply.

In accordance with the findings in [Shoemaker et al. 2012], the goodness of fit ($r^2 = .96$ for *Laser+Gyro*, .93 for *Laser+Position* and .92 for *Laser+Track*) is consistently better than when modeling the global task time with Fitts' law ($r^2 = .93$ for *Laser+Gyro*, .86 for *Laser+Position* and .85 for *Laser+Track*). Since this could be due to our model having two parameters instead of one, we computed the Akaike Information Criterion with correction (AICc)[18][Akaike 1974]. The results confirm that our model provides a better fit.

Table II shows a wide range of intercepts for the various regressions, including some negative ones. However we cannot interpret these intercepts because we cannot extrapolate these regressions to low IDs. In particular, low-ID tasks can be performed with Dual-Precision techniques using only their *Coarse* mode, which our model does not take into account. Fig. 17 (left) shows the Fitts' law regressions, allowing us to compare all four techniques. We see that, for the range of IDs that we tested, *Laser+Gyro* and *Laser+Track* have comparable throughputs with a consistent disadvantage for *Laser+Track*. *Laser+Position* has a slightly lower throughput and is in between the other two techniques. Finally, *GyroAcc* has the lowest throughput. It is the fastest for low IDs and the slowest with high IDs, supporting both prediction *P1* and prediction *P2*. Fig. 17 (right) shows the average movement time for each target width and is therefore independent of the model we choose. It shows the same pattern: *Laser+Gyro* is, overall, faster, with *GyroAcc*faster for large targets, but much slower than any other technique for small targets.

Focusing on our model, we observe that for the Dual-Precision techniques b_p is consistently higher than b_c, with a ratio $r = b_p/b_c$ of 1.36, 1.76 and 1.84 respectively for *Laser+Gyro*, *Laser+Position* and *Laser+Track*. Users seemed to have more difficulty with the *Precise* mode, maybe due to mental readjustment caused by the change of input modality from *Coarse* to *Precise*. This is inconsistent with our assumption that Fitts' law parameters a and b are the same in both phases (Section 4.1, page 11). Future work should therefore study the effect of L on performance and fine tune the a priori value of r from preliminary evaluations of each mode separately. However, we observe that our approach resulted in a set of techniques that allowed all participants to perform the hardest pointing tasks in less than 6 seconds on average.

5.9. Mode Switches

In order to better understand the effect of the three dual-mode techniques on performance, *Laser+Gyro*, *Laser+Position* and *Laser+Track*, we performed a detailed analysis of mode switches.

Despite our instructions, some participants chose not to switch mode for some trials (5 %) and perform the task in *Coarse* mode only. We observe a significant effect of WIDTH ($F_{3,33} = 4.13$, $p = 0.0136$), AMPLITUDE ($F_{2,22} = 5.8$, $p = 0.0095$) and WIDTH×AMPLITUDE ($F_{6,66} = 5.12$, $p = 0.0002$) on the number of such "coarse-only" trials. Unsurprisingly, participants performed more coarse-only trials with the largest targets WIDTHS (30 mm) than with the the smaller ones (4 and 7 mm). More interestingly, participants performed more

[18]This criterion measures the relative goodness of fit of a statistical model and assesses overfitting, i.e., increasing the number of free parameters to improve the goodness of fit.

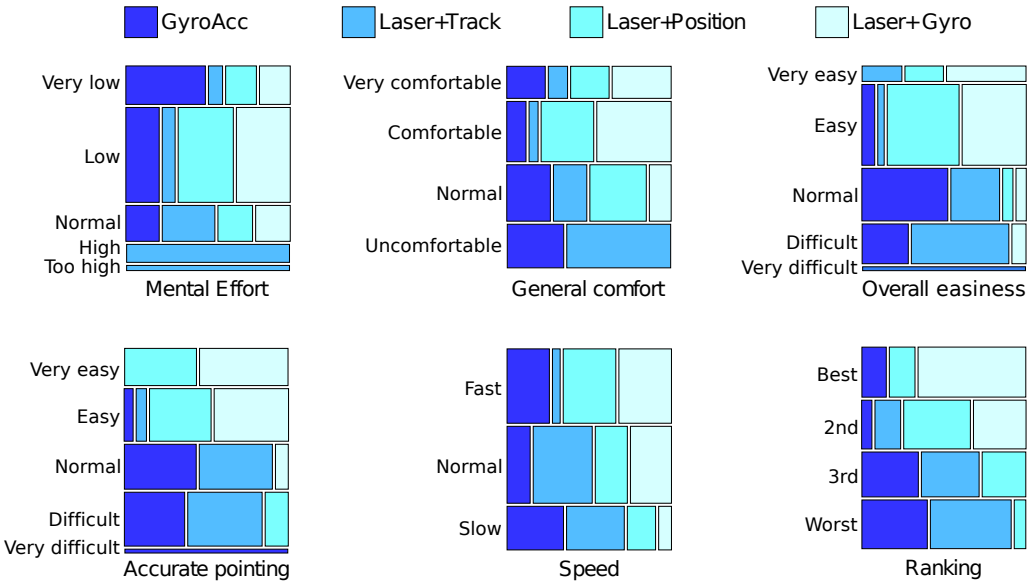


Fig. 18: Mosaic plot of the qualitative results from Exp. 2. The height of a box represents the number of participants that used this grade or rank overall (regardless of technique). The width of a box represents the proportion of a particular grade or rank for a given technique.

coarse-only trials with the smallest AMPLITUDE (637 mm) than with the other two. Finally, the small amplitude $\times$ small target condition caused significantly more coarse-only trials than any other condition. In the following analyses of mode switch occurrences we removed the trials with no mode switch.

Switch Time and *MS Distance* measure respectively the time and distance to target of the first mode switch. We consider that the mode switch was an error when performed at a distance greater than trials $3 \times L$. Such trials (13 %) were excluded from the analyses.

We observed no significant effect of TECHNIQUE on *Switch Time*. Unsurprisingly, *Switch Time* increases significantly with AMPLITUDE ($F_{2,20.86} = 147.96$, $p < 0.0001$). However we also found a significant effect of WIDTH ($F_{3,32.33} = 8.84$, $p = 0.0002$), in contradiction with prediction *P6*. This is surprising since the size of the *Coarse* mode "target", L, does not depend on the actual target WIDTH. The post-hoc test reveals that users switch modes later for smaller targets (WIDTHS = 4 and 7 mm) than for larger ones (15 and 30 mm).

Similarly, we did not observe any effect of TECHNIQUE on *MS Distance*. We note that the actual distance between the first mode switch and the target is always lower than our conservative estimate of $L/2 = 80$ mm (mean 27.7 mm, stdev 23.2). This is to be expected since we calculated L as a higher bound for *MS Distance*, to define the visual size of the cursor. We did observe, however, a significant effect of TECHNIQUE for WIDTH ($F_{3,32.69} = 4.72$, $p = 0.0076$) and AMPLITUDE ($F_{2,21.99} = 9$, $p = 0.0014$). Participants switched mode further away from the target with larger AMPLITUDE (3187 and 1912 mm) than with the smaller one (637 mm). Participants also switched mode further away with the largest target (30 mm) than with the smaller ones (4 and 7 mm).

We computed the time spent in *Precise* mode as *Precise Time* = *Pointing Time* - *Switch Time* and found a significant effect of TECHNIQUE ($F_{2,19.19} = 11.57$, $p = 0.0005$) and WIDTH ($F_{3,28.33} = 81.48$, $p < 0.0001$). Unsurprisingly, *Precise Time* with targets of 4 mm were higher than targets of 7 mm, which were higher than targets of 15 and 30 mm. More interestingly, *Laser+Gyro* caused shorter *Precise Time* than *Laser+Position* and *Laser+Track*.

Finally, we observed that *Switch Time* follows Fitts' law with the model $Switch\,Time = a + b \times \log_2\left(1 + \frac{Amplitude}{L}\right)$ with goodnesses of fit equal[19] to 1 for *Laser+Gyro* and *Laser+Position* and 0.99 for *Laser+Track*.

5.10. Qualitative results

A Pearson χ^2 test shows that there is a significant effect on *Mental Effort*, *Accuracy*, *Comfort*, *Easiness* and *Ranking* for TECHNIQUE. 6 participants graded mental effort for *Laser+Track* as *High* or *Too high*,

[19]Rounded at two decimals.

and all 12 graded all other techniques *Normal* (3) or below. 6 participants graded precise pointing with *Laser+Track* and *GyroAcc* as *Difficult* or *Very difficult* and 9 graded it *Easy* or *Very easy* with *Laser+Gyro* and *Laser+Position*.

10 participants graded *Laser+Track Uncomfortable* or *Very uncomfortable*. 10 and 7 participants respectively graded *Laser+Gyro* and *Laser+Position* as *Comfortable* or *Very comfortable*, partially supporting prediction *P4* . 10 participants graded *Laser+Position* and *Laser+Gyro* as *Easy* or *Very easy* and 10 graded *Laser+Track* as *Normal* (3) or below. Finally, 8 participants preferred *Laser+Gyro* overall, 2 preferred *Laser+Position*, 2 preferred *GyroAcc* and none preferred *Laser+Track*. Overall, these results are consistent with the quantitative analysis.

5.11. Discussion

5.11.1. Coarse and Precise phases. Based on these results, we hypothesize that users adapt their *Coarse* pointing phase according to the expected difficulty of the *Precise* phase in order to optimize general pointing time. Indeed, despite being functionally unrelated to target WIDTH, the *Coarse* phase lasted longer with, and ended closer to, the smaller targets. We presume that this behavior was intended to make smaller targets easier to acquire by lowering the amplitude of the *Precise* phase. Participants also switched modes closer to the target when the AMPLITUDE was smaller, possibly capitalizing on a normally shorter *Coarse* phase to switch mode closer to the target.

Despite comparable *Switch Time* and *MS Distance*, participants spent less time in *Precise* mode with *Laser+Gyro* than with the other two Dual-Precision techniques. We hypothesize that this is due to the angular control of the *Precise* mode of *Laser+Gyro* being similar to that of the *Coarse* mode (ray-casting), whereas the *Precise* mode of *Laser+Position* and *Laser+Track* both involves linear control and different limbs. Also, participants sometimes avoided switching to *Precise* mode for small AMPLITUDE and large target WIDTH. We hypothesize that they tried to minimize pointing time by saving the cost of both switching mode and adapting their input to the *Precise* phase control.

Technique	*GyroAcc*	*Laser+Track*	*Laser+Gyro*	*Laser+Position*
Timeouts	–	–	++	+
Misses	–	+	+	+
Clutching	++	–	+	+
Crossings	–	-	+	+
Dwell	–	+	+	+
Reaching	–	++	o	o
Pointing	+ lowID - HighID	–	o	o
Mental Effort		–		
Precision	–	–	++	++
Comfort		–		
Easy		o	++	++
Preference	+	–	++	+

Table III: Overall assessment of the techniques on a five-point scale ("–", "-", "o", "+", "++").

5.11.2. Techniques. Table III summarizes the quantitative and qualitative results by ranking the techniques on a five-point scale.

GyroAcc was generally perceived as imprecise, which is consistent with the qualitative results. More precisely, despite the fact that it was the fastest technique to reach the target, it caused more crossings and misses than the other techniques, especially with very small targets. This indicates that users could easily perform fast, coarse pointing but that stabilizing the cursor over a small target was difficult. However, *GyroAcc* performed well in both speed and accuracy for WIDTHS 7 mm and larger, indicating that pointing tasks of this difficulty can be performed without using a Dual-Precision technique to explicitly trade speed for precision.

Laser+Track was the slowest and least preferred technique overall. Despite having the same *Coarse* mode as the other dual-precision techniques, it was the slowest for *Reaching*; it also caused many *Clutching* and *Crossings*, meaning that the *Precise* mode is neither precise nor fast enough to compete with other dual-precision techniques. In addition, it was not better for recalibration time, contradicting prediction *P5* (we expected that its *Precise* mode would cause smaller displacements of the absolute cursor). These results are consistent with the participants' opinion: hard to use, imprecise and uncomfortable. This may be due in part to the following problem: 5 participants reported that lifting the finger from the tracking zone of the device

(zone 1 in Fig. 11) caused a loss in precision despite our finger-release adjustment, and that they would have preferred a physical button. An improvement could be to implement the *Precise* mode of *Laser+Track* with an input device that couples finger tracking with physical actuators, such as Apple's Magic Mouse or Microsoft's Touch Mouse.

Laser+Gyro and *Laser+Position* had very similar results, supporting prediction *P3*. They caused almost no *TimeOuts*, meaning that they were able to withstand very difficult tasks such as those in the experiment. They were also the most stable (best for *Crossings*) and fastest techniques overall. They were both perceived as precise, comfortable and easy to use. *Laser+Gyro* was the preferred technique and had a slight advantage when pointing at very small targets, which is consistent with prediction *P2*.

It is interesting to note the sudden decrease in performance of *GyroAcc* for the smallest targets. We hypothesize that there is a limit to the output range of continuous transfer functions above which the steepness of the curve makes it difficult to control the transition between fast and precise movements. While this particular theory should be evaluated further, we can at least observe that Dual-Precision techniques do not have such a limitation since the transition between modes is discrete: there is no need for a smooth transition between the ranges of CD gains used in *Coarse* and *Precise* modes. Eq. (23) shows that, in most cases, the theoretical limit of difficulty of dual-precision techniques (18.97 for *Laser+Gyro*, i.e., targets of 0.01 mm on a 5.5-meter ultra-wall) is higher than human visual acuity.

Finally, it is worth noting that despite their relative differences, *all* techniques enabled participants to reliably perform pointing tasks with IDs up to 9.6 bits, a difficulty never evaluated before in a single-scale pointing study. This demonstrates the effectiveness of our design approch and theoretical analysis of unimanual Dual-Precision and Pointer Acceleration pointing techniques. The next section applies a similar method to design and evaluate touch-based pointing techniques that minimize input resources.

6. STUDY 2: SINGLE- VS. DUAL-CHANNEL TECHNIQUES USING TOUCH DEVICES

Fig. 19: Interacting with an ultra-wall using a tablet that features both control widgets and an area dedicated to high-precision remote pointing (top-left).

In this section we focus on techniques that can be used on a small portion of a touchscreen so that other widgets can be displayed simultaneously. We consider contexts of use where the primary purpose of the handheld device is to accommodate widgets for the advanced control of objects selected via pointing. The physical area dedicated to pointing on the device, or *pointing zone*, should therefore not be too large. However, if too small, pointing will be inefficient. Our goal is to identify the best trade-off between screen real-estate allocation and good pointing performance.

In Section 3.2 we demonstrated that a smartphone screen does not provide enough input resolution to acquire small (visible) targets on an ultra-wall with a constant CD gain, and that the full size of a tablet screen would have to be used for this purpose. However we would like to use the smallest area possible for input on the device (tablet or smartphone) to allow for other interactions. Thus, more than one CD gain is required. To address this problem we explore two ways to augment the input expressiveness of a touch area with Dual-Precision techniques (Section 4.1): using an additional input channel, and distinguishing between two modes when touching the handheld.

6.1. Techniques

We showed in the previous sections that the main challenge when designing Dual-Precision techniques is to seamlessly integrate the two modes so that the mode switch minimizes cognitive and/or motor costs. All the techniques presented in this section assign the *Precise* mode to single-finger drag gestures performed with the dominant hand in the pointing zone of the handheld device. Since the non-dominant hand typically holds the device [Wagner et al. 2012], this leaves two main options for the *Coarse* mode:

— Use the dominant hand in both modes but in two different configurations that are easily distinguishable by the system and the user;

— Use a different body part for the *Coarse* mode, that does not impair the control of the *Precise* mode.

We experimented with several options and identified two viable assignments for the *Coarse* mode through iterative design, prototyping, piloting and tuning: double-finger drag gestures with the dominant hand in the handheld's pointing zone, and tracking the head orientation. Both approaches are detailed below.

6.1.1. Two-finger pad-based coarse pointing. This approach is inspired by ARC-Pad [McCallum and Irani 2009], a pointing technique that provides users with an absolute and a relative pointing mode. A typical ARC-Pad pointing task is composed of a tap (press-release) on a touch-sensitive handheld device, followed by a dragging gesture. The tap gesture coarsely positions the cursor on the large display according to an absolute mapping of the handheld device's surface to the large display. The following drag gesture is interpreted as relative movements of the cursor to adjust its position. Tapping vs. dragging is a practical way of differentiating the user's intention between coarse and precise cursor control, and was proven efficient on a 52" screen with a resolution of 1360×768 pixels. However, this approach does not scale: when mapping a 7 cm input device to a 5 meter display, a 1 millimeter error in the tap location corresponds to a 7.5 cm error on the display, resulting in multiple dragging gestures to reach the target. ARC-Pad proved very difficult to use on ultra-walls, mainly because in that context the absolute mapping is far too imprecise and often requires either several attempts (taps) in *Coarse* mode to move the cursor close enough to the target, or several relative-mode drag gestures to adjust the cursor position, causing much clutching in the second phase. We thus adapted the original technique so that users can adjust the location of their cursor in *Coarse* mode. We call this variation of the original technique ARC-Pad2[20].

ARC-Pad2 distinguishes between *Coarse* (absolute) and *Precise* (relative) pointing by the number of fingers involved in performing the pointing gestures rather than by the type of gesture. A single-finger drag gesture controls the cursor in *Precise* mode (as before); a drag gesture performed with two conjoined fingers [Kin et al. 2011] is interpreted as absolute positioning of the cursor. As opposed to the original method that relied on tap gestures for absolute mapping, users can now adjust the cursor position in *Coarse* mode as well by dragging the two fingers, and then switch to *Precise* mode for more precise, relative adjustments of the cursor position. The switch from *Coarse* to *Precise* mode is triggered whenever at least one of the fingers is lifted from the pad's surface. This means that users can either lift a single finger at the current location and continue dragging, or they can lift both fingers, adjust their hand position relative to the pad, and touch anywhere on the surface with a single finger. This second option can be very useful when pointing at a target near the edges of the display, as it lets users initiate relative drag gestures from the center of the pointing zone, in any direction.

We also considered the symmetric approach, i.e. one finger for coarse pointing and two for precise pointing, but preliminary tests showed that lowering a second finger in order to switch to *Precise* mode may cause the second finger to fall out of the input area when the target is near the display borders. This has either no effect if the finger did not reach an interaction widget outside the pointing area, or it can trigger unwanted commands. In both cases users may wonder why lowering their finger did not have the expected effect and look at the input device, thus losing their focus. Leaving the input area when using two fingers is less likely to happen in *Coarse* mode since it uses an absolute mapping: users always have a feedback of their fingers' location on the input area through the position of the cursor on the display.

6.1.2. Head-based coarse pointing. The second viable approach that we identified for *Coarse* control uses the natural head movements that occur when remotely pointing at targets on a wall.

Object selection is often preceded by a visual search when the target is not located in the user's immediate field of view. Head orientation provides a good approximation of where a user is looking [Stahl 1999; Freedman and Sparks 2000; Nickel and Stiefelhagen 2003]. Head movements can also be exploited in conjunction with any positioning device used in the environment [Kitajima et al. 2001; Nickel and Stiefelhagen 2003]

[20] ARC-Pad$_{2F}$ in [Nancel et al. 2013]

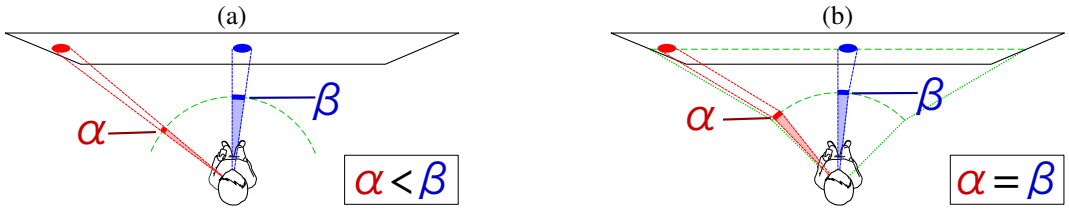


Fig. 20: Varying angular motor size with ray-casting (a) can be fixed using an indirect absolute mapping (b).

and has been shown to support a variety of interaction techniques [Kitajima et al. 2001; Morency and Darrell 2006; Kollerl et al. 1996]. Our approach integrates head motion with cursor selection. It was inspired by [Ashdown et al. 2005], who use head orientation for positioning a cursor on the monitor of interest, thereby reducing mouse trips in a multi-monitor setup. We provide a framework for designing target selection techniques that use head orientation, based on the results of a study of head and body orientation when acquiring targets in mid-air on an ultra-wall [Nancel 2012, App. C].

[Stellmach and Dachselt 2012] took a similar approach using gaze instead of head orientation. They introduce four techniques that follow a common design guideline: *gaze suggests, touch confirms*. While their results are promising, they did not compare their design with existing techniques, and only evaluated it in desktop environment. Wall-sized displays feature much greater sizes than desktop screens, regardless of pixel density. While desktop screens can often be skimmed through with minimal movement other than the eyes', wall-sized displays may require much larger head rotations, as discussed in Section 3.1. Confirming prior work [Jacob 1991; Zhai et al. 1999], [Nancel 2012, App. C] observed that users consistently stabilize their gaze and lock it on the target while acquiring it. While tracking the users' gaze is not a practical option in our context, we found head orientation to be a good coarse indirect indicator of where users are looking[21], and thus a good predictor of the rough location of the next target on the display.

Our technique controls the cursor's location in *Coarse* mode through an absolute but indirect mapping of head orientation to cursor position on the wall. We favored an indirect mapping over direct ray-casting (or laser pointing) for two reasons. First, being based on angular movements, ray-casting is perspective-dependent and would have caused targets of the same size on the display to have noticeably different *motor* sizes depending on their location with respect to the user's physical position, as illustrated in Fig. 20-a (this effect gets amplified as users get closer to the display). We also wanted to optimize the users' input operating range, within the limits of comfortable neck and eyes positions.

As illustrated in Fig. 20-b, we addressed these issues by mapping a location-independent, fixed-size angular operating range centered on the orthogonal projection of the user's location on the flat display surface. This ensures that when users move in front of the display, looking straight ahead always sets the cursor exactly in front of them. As they look away, the cursor is progressively offset from the direction of the head (up to 12.5°, see Fig. 21), accounting for the extra rotation of the eyes. This offset makes it possible to point at targets on the sides of the wall comfortably while maximizing accuracy in the central area.

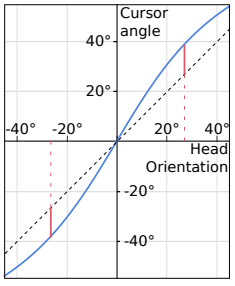


Fig. 21: Angular abscissa of the cursor as a function of the horizontal orientation of the head. The maximum offset (plain red line), i.e. the difference between the head orientation and the cursor angle, is 12.5°.

This is an absolute yet indirect technique with a constant transfer function: all targets with the same display size have the same motor size. Its precision directly depends on the operating range of the input channel, which is limited by the angular range of head orientation that users can reach without stretching their neck

[21]When a target is off-centered relative to the default head position, users rotates their head in order to minimize the effort put on ocular muscles [Freedman and Sparks 2000].

too much and with minimal difference between eye and head orientation. Increasing the angular operating range makes small targets easier to acquire, but amplitudes become larger. This technique facilitates mid-air interactions when the user moves around the room, since target motor size does not depend on the distance to the wall.

6.1.3. Quantized coarse pointing. In the first study (Section 5) we observed that users changed their behavior in *Coarse* mode depending on the apparent relative difficulty of the *Precise* phase. Participants typically spent more time adjusting the location of the mode switch when the *Precise* phase looked challenging (i.e. smaller targets) or when the *Coarse* phase looked easy (i.e. small pointing amplitude). In the easiest cases they even skipped the *Precise* phase and achieved the selection in *Coarse* mode only. However this optimization behavior occurred in only 5 % of the trials and did not significantly improve performance.

The *Coarse* phase was originally designed to embody the initial, ballistic phase of a pointing task. While fast, unconstrained *Coarse* movements showed signs of accuracy optimization and thus of additional cognitive cost. This raises a question: beyond minimizing the motor cost of switching between modes, is it possible to minimize the *cognitive* cost associated with making the decision to switch between the two modes?

We explored an approach where the cognitive load is transferred to the perceptual system, hypothesizing that this would significantly reduce the switching cost. We designed variations of the techniques described above that artificially limit the precision of the *Coarse* pointing mode: users approach the target fast and know when to switch to *Precise* mode simply because there is no other option. We *discretize* the display according to a set of *cells*, in our case the 8×4 matrix of 30" LCD panels, so that in *Coarse* mode users can only jump from cell to cell. Adopting the physical subdivision of the wall into display tiles ensures that cells do not intersect the screen edges; Subdividing the screens further would either lead to cells with aspect-ratios different from that of the input zone, or too small to be accurately acquired in *Coarse* mode. We call this process *discretization* to emphasize the fact that the pointing resolution in *Coarse* mode is artificially degraded while keeping the same physical input resolution.

The resulting technique is reminiscent of the Rake cursor [Blanch and Ortega 2009; Räihä and Špakov 2009] and similar multi-cursor desktop pointing techniques that use eye gaze to select the active cursor in a matrix of cursors all controlled with the same input device. However our technique always relocate the cursor at the center of the currently head-pointed cell when in *Coarse* mode, as opposed to keeping its last *Precise* location relative to the cell: Although we expected users to remember where they left the cursor when they last switched to *Coarse* mode, our tests showed otherwise. We found that having the *Precise* cursor start in the middle of the cell yields better performance and is less demanding. It also results in a constant average distance between the cursor and the targets in that cell. Once the cell containing the target is selected, users have to switch to *Precise* mode to reposition the cursor within that cell.

6.1.4. Four Dual-Precision pointing techniques. We eventually narrowed our design down to four pointing techniques by combining the two coarse input control techniques (pad-based vs. head-based) with two *Coarse* mode input precisions (*Continuous* vs. *Discrete*). As mentioned earlier, all four techniques use the handheld device's pointing zone to control the cursor in *Precise* mode using an optimized CD gain transfer function.

— ARC-Pad2 + *Continuous* = *ContPad*: two conjoined fingers in the handheld's pointing zone activate *Coarse* mode (absolute mapping of the zone to the wall); a single finger activates *Precise* mode (relative cursor displacements).

— ARC-Pad2 + *Discrete* = *DiscPad*: same as *ContPad* but pointing precision in *Coarse* mode is artificially restricted: the cursor can only move to the center of a cell, requiring a switch to *Precise* mode for further cursor adjustments.

— *Head* + *Continuous* = *ContHead*: head orientation provides *Coarse* control of the cursor. Touching the pointing zone on the handheld automatically switches to *Precise* mode (relative cursor displacements).

— *Head* + *Discrete* = *DiscHead*: same as *ContHead*, but pointing precision in *Coarse* mode is artificially restricted. Head orientation can only make the cursor jump to the center of a cell, as for *DiscPad*.

With the two ARC-Pad2 techniques, the cursor only moves if one or two fingers touch the screen, allowing users to easily perform clutch and tap gestures. For the two *Head*-based techniques, *Coarse* mode is the default mode: as long as the user does not touch the handheld screen, the cursor follows the user's head movements. *Head*-based techniques also feature a 500 ms delay after the finger is released from the handheld screen before switching back to *Coarse* mode, letting users perform clutch and tap gestures in *Precise* mode.

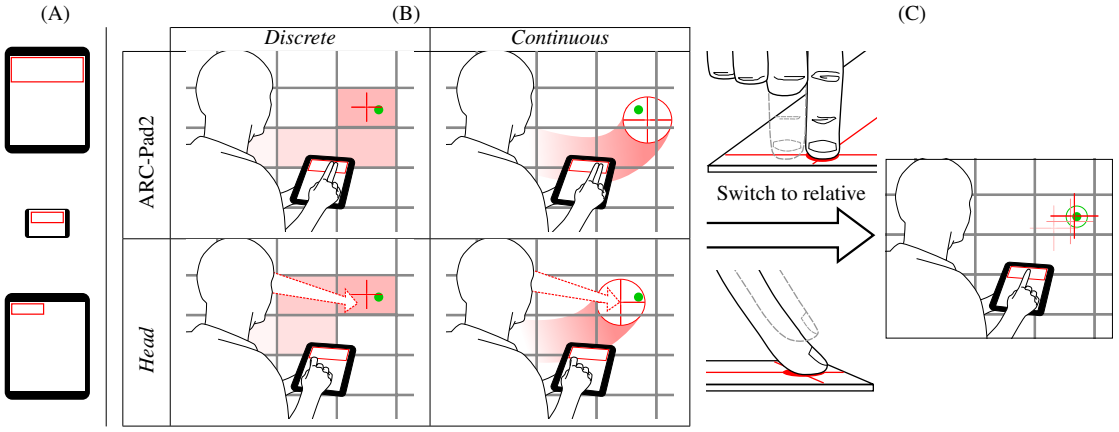


Fig. 22: The four Dual-Precision techniques. (A) The three combinations of device and pointing zone used in our experiments: tablet with large zone (Experiments 1 and 3), smartphone with small zone (Experiment 2) and tablet with small zone (Exp. 3). (B) The four coarse (absolute) modes, combinations of *Head* vs. ARC-Pad2 and *Discrete* vs. *Continuous*. (C) In all cases, using a single finger in the pointing area switches to precise (relative) mode.

To calibrate the transfer functions, we followed the procedure described in Subsection 8.2 and asked three volunteers to test the techniques in an informal iterative design process of 800 pointing trials per user on average. Table IV gives the parameter values that we obtained after extensive pilot testing for the two pointing zones that served in the experiments reported below: a large zone that fits a tablet screen in portrait mode and a small zone that fits a smartphone screen in landscape mode.

Size	Technique	CD_{max}	λ (s/mm)	$ratio_{inf}$
Large zone 148 × 49 (mm)	$ContHead_L$	9.9	.02	0.4
	$ContPad_L$	6.6	.018	0.4
	$DiscHead_L$	12.9	.018	0.4
	$DiscPad_L$			
Small zone 75 × 26 (mm)	$ContHead_S$	6.3	.027	0.4
	$ContPad_S$			
	$DiscHead_S$	16.4	.027	0.7
	$DiscPad_S$			

Large zone: $V_{min} = 0.006m/s$, $V_{max} = 0.37m/s$, $CD\ gain_{min} = 0.22$
Small zone: $V_{min} = 0.003m/s$, $V_{max} = 0.19m/s$, $CD\ gain_{min} = 0.27$

Table IV: Transfer function parameter values.

V_{min} and V_{max} are lower for the small zone because users have less physical (motor) space at their disposal for pointing, and thus less amplitude to accelerate. Similarly, the small zone features a higher $CD\ gain_{max}$, enabling faster cursor movements to compensate for the smaller operating range. *Discrete* techniques feature a higher $CD\ gain_{max}$ than *Continuous* ones to compensate for the larger distance between the target and the point where mode switching is performed.

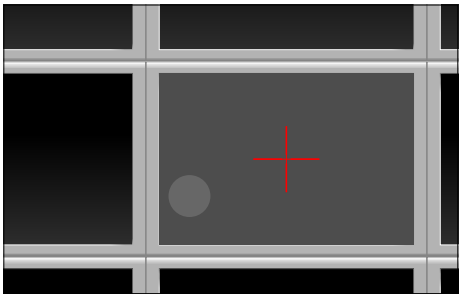


Fig. 23: Cursor of the *Discrete* techniques in *Coarse* mode. The precise cursor (red cross) is located at the center of the screen currently pointed at, which is highlighted (white, 5% translucency). A translucent proxy (white circle, 2% additional translucency) indicates the location of the continuous pointing input in order to enable users to anticipate when the pointed screen will change.

6.1.5. Cursor Feedback. We use a red cross for the cursor and additional feedback in *Coarse* mode. For the *Continuous* techniques we surround the cursor with a red circle. As in the experiment reported in Section 5, the size of the circle is computed using Eq. (19) and gives users an indication of when to switch to *Precise* mode.

For the *Discrete* techniques, in *Coarse* mode the visible cursor is a red cross always located at the center of the pointed cell, which is slightly highlighted (Fig. 23). Preliminary tests showed that users also need to know how close they are to switching cells, otherwise flickering may happen when the cursor is moving back and forth over the edge between two cells. We added a translucent disc (Fig. 23) to provide feedback about the exact, continuous location pointed to by the *Coarse* technique so that users know how close they are to the edge of the current cell. We tried other solutions such as highlighting the border of the cell when the user is about to change cell, but they require more attention to be understood than a simple proxy of the pointed location. To reduce clutter and since they are not critical to the pointing task, the screen highlight and the proxy of the continuous pointing location are displayed with low opacity (5% for the screen, combined with 2% for the proxy).

6.2. Comparing the Techniques

We conducted two experiments to evaluate the performance of the four Dual-Precision pointing techniques introduced above, *ContHead*, *ContPad*, *DiscHead* and *DiscPad*, and to assess the cost of mode switching. The two experiments followed the same design but involved two different devices and input sizes: we used a tablet with a large pointing zone in the first experiment, and a smartphone with a small pointing zone in the second one. For the sake of clarity, we use subscripts $_L$ (*Large*) and $_S$ (*Small*) when referring specifically to the tablet vs. smartphone conditions, e.g. $ContHead_L$ and $ContHead_S$.

Our hypotheses are as follows:

(H1) *Discrete* techniques reduce the cognitive cost of mode switching by leaving no choice to users about when to switch mode, leading to a shorter *Coarse* pointing phase than *Continuous* techniques.

(H2) *Head*-based techniques make mode switching cognitively less demanding, as they use different body parts to control the two modes. *Pad*-based techniques use the fingers in both modes and for mode switching. With *Head*-based techniques, a mode switch is triggered when the finger comes into contact with the pointing zone.

Related to (H1), we expect an effect of forcing the mode switch with *Discrete* techniques on the time spent in the *Precise* phase, since this often entails engaging the *Precise* mode significantly farther away from the target than with *Continuous* techniques.

We also expect to observe relative differences between the two experiments due to the smaller pointing zone used in the second one: we expect *Head*-based techniques to be at an advantage in the second experiment, as the smaller size of the pointing zone will hinder performance of *Pad*-based techniques in *Coarse* mode.

6.2.1. Participants. The same 12 participants (2 female; 24 to 38 year-old, avg. 29.6, med. 28.5) served in both experiments. All had normal or corrected-to-normal vision and were right-handed. All were daily users of personal computers and smartphones. Only two used tablets regularly.

6.2.2. Apparatus. Both experiments were conducted on the same WILD platform as the first experiment. We used the VICON motion-capture system to track passive IR retroreflective markers and provide the 3D coordinates of the participant's head. Participants stood up, 2 meters away from the display. Given this position and the size of the wall, the operating range of the head was $\pi/2$ rad horizontally $\times\, \pi/5$ rad vertically. For the *Discrete* techniques, the cells correspond to the physical screens of the display (8×4 cells of $2\,760 \times 1\,800$ pixels or 637 mm $\times$ 415 mm).

6.2.3. Trials. The task consists of acquiring circular targets of two WIDTHS: 4 mm (18 pixels) and 18 mm (80 pixels). Participants first move the cursor in *Coarse* mode to a dedicated zone (a cell for the *Discrete* techniques, a 500-pixel circle for the *Continuous* techniques) symmetric to the target location from the center of the display (see Fig. 24) and dwell for half a second. Then they acquire the target, located at one of three AMPLITUDES: 2 760 pixels (width of one LCD panel, 0.637 m), 8 280 pixels (width of 3 LCD panels, 1.912 m), 13 800 pixels (width of 5 LCD panels, 3.187 m). The corresponding Index of Difficulty ranges from 5.19 to 9.64 bits.

For each AMPLITUDE, we use six combinations of starting and ending screens around the center of the display (Fig. 24) so that the dwell zone is never located in the same screen as the previous target. Targets

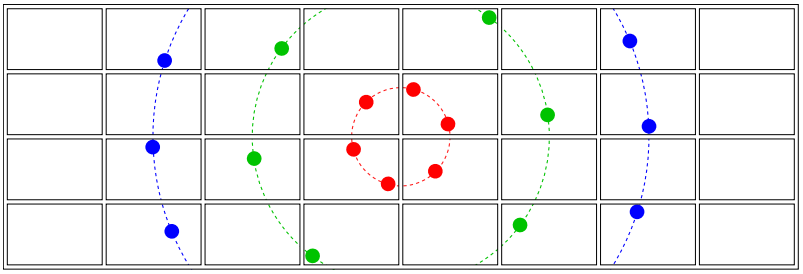


Fig. 24: Location of the six targets in all three experiments for each AMPLITUDE. Colors were changed for readability: the experiment background was black and all targets were green. Targets are also represented much larger than they were in the experiment.

are pseudo-randomly positioned within a screen so that the average distance from the center of that screen to the target is about 600 pixels (150 mm). We chose this value so as to neither advantage nor disadvantage *Discrete* techniques (relative to the size of a screen), given that those techniques position the cursor at the center of the most recently visited LCD panel.

6.2.4. Design and Procedure. Both experiments use a $4 \times 2 \times 3$ within-subject design with three factors:

(1) TECHNIQUE: *ContHead*, *ContPad*, *DiscHead* and *DiscPad*

(2) WIDTH: 4 mm (18 pixels) and 18 mm (80 pixels);

(3) AMPLITUDE: 637 mm (2 760 pixels), 1 912 mm (8 280 pixels) and 3 187 mm (13 800 pixels)

The trials are blocked by TECHNIQUE and the presentation order of techniques among participants is balanced using a Latin square. At the beginning of each block of a *Head* technique, participants are asked to stand still and stare at the center of the display for 5 seconds to calibrate the center of their head's operating range.

Each new TECHNIQUE starts with a training session composed of two parts. In the first part, WIDTH and AMPLITUDE are set to their largest values. The operator explains the technique and asks participant to practice until their performance has stabilized. Performance is considered stable when the participant performs six consecutive trials with a movement time variation below 40 % of their average, similar to the first experiment (Section 5). Participants were allowed to practice longer if they wanted to. Then the second part of the training session starts, consisting of three blocks of trials covering the experimental conditions: largest WIDTH and largest AMPLITUDE, smallest WIDTH and medium AMPLITUDE, smallest WIDTH and smallest AMPLITUDE. After the training session, the next six TECHNIQUE blocks are measured, each consisting of six replications of a WIDTH $\times$ AMPLITUDE condition.

For both experiments, sessions lasted 40 minutes on average. At the end of a session, participants answered a questionnaire about their preferences and were encouraged to make comments.

6.2.5. Measures. To summarize, we collected 4 TECHNIQUE $\times$ 2 WIDTH $\times$ 3 AMPLITUDE $\times$ 6 replications $\times$ 12 participants = 1728 measured trials. We measured movement time *MT* – from the moment participants stop dwelling to the first successful press event on the target – and error rate, the number of trials where the first selection is not inside the target. We split *MT* into movement time of the *Coarse* phase, *CMT*, and movement time of the *Precise* phase, *PMT*, according to the time of the *last* switch to *Precise* mode (in order to ignore erroneous mode switches). To evaluate the cost of mode switching, we also measured *VPT*, the time between the last event in *Coarse* mode and the peak in velocity in the subsequent *Precise* phase. Time measurements are detailed in Fig. 25.

6.3. Tablet Experiment

In this experiment, we used a tablet with a resolution of 768×1024 (Apple iPad, weight: 680 g, dimensions: $19 \times 24.3 \times 1.3$ cm, screen diagonal: 24.6 cm). Participants had to hold the tablet in portrait mode. The pointing zone used the top 768×256 pixels (148×49 mm) yielding an input resolution of 5.2 dot/mm. The transfer functions are shown in Fig. 26.

6.3.1. Quantitative Results. We removed 0.69 % of the trials, which had an unreasonable residual/predicted ratio. These outliers were mainly due to Wi-Fi transmission problems between the input device

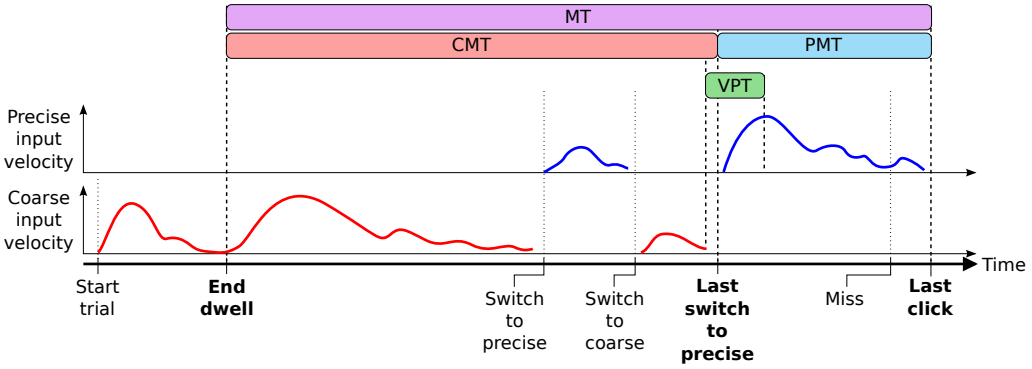


Fig. 25: Events and corresponding measurements made in the experiment. Dotted lines indicate events that are not taken into account: erroneous clicks ("misses") and mode switches other than the last switch to *Precise* mode.

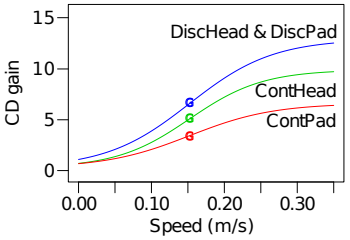


Fig. 26: Transfer functions used for the *Precise* modes in the tablet experiment (see Table IV for parameter values).

and the computer running the study. As expected, *MT* distributions per condition are skewed. We thus perform our analyses using median values, per participant, on the model TECHNIQUE × WIDTH × AMPLITUDE × Rand(Participant). Fig. 27-a graphs *MT* for each technique[22].

Movement Time. A multiway ANOVA reveals a significant effect of TECHNIQUE on *MT* ($F_{3,33} = 11.4$, $p < 0.0001$). A post-hoc t-test with Bonferroni correction shows that $ContHead_L$ is significantly faster than all other techniques (all p's < 0.001): 9 % faster than $ContPad_L$, 15 % faster than $DiscPad_L$, and 14 % faster than $DiscHead_L$. The only other significant difference is between $ContPad_L$ and $DiscPad_L$, the former being 6 % faster.

We also observe, as expected, significant effects of target width WIDTH ($F_{1,11} = 459$, $p < 0.0001$) and movement amplitude AMPLITUDE ($F_{2,22} = 43.8$, $p < 0.0001$) on *MT*, participants being faster with the larger target and the smallest distances. We do not observe any interaction effect TECHNIQUE × WIDTH or TECHNIQUE× AMPLITUDE.

Fitts vs. our model. We ran regressions for Fitts' law and for our model presented in Section 4.1. For *Continuous* techniques we used $d = L/2$ in Eq. (14). For *Discrete* techniques, since the *Coarse* cursor feedback consisted only of highlighting (screen) cells, we used the average between the width and height of the screens in order to account for the various angles of the pointing trajectories (see Fig. 24). Overall, the values of $r = b_p/b_c$ are generally closer to 1 than in the first experiment and they are lower for *Discrete* techniques (1.01 and 1.16) than for *Continuous* ones (1.36 and 1.63). This suggests that the techniques in this experiment, in particular the *Discrete* ones, better balance the use of the input system between the two modes. However, as we will see, this does not result in a performance advantage for *Discrete* techniques.

As in the first experiment, we observe that our model (Eq. (16)) has a better fit than Fitts' law (Table V), although the difference is less pronounced than in the unimanual experiment. $DiscHead_L$ has a low r^2 score in both models, which may indicate that factors other than WIDTH and AMPLITUDE have a strong effect on pointing time when using this technique on a tablet.

We observe that all techniques except $ContHead_L$ show similar performance at the lower end of the ID range, but have different slopes: both *Discrete* techniques exhibit lower throughputs than $ContPad_L$ in both models, indicating that participants were generally less efficient using them. More precisely, *Continuous* techniques show much lower b_c values, which could indicate that *Discrete Coarse* modes were harder to

[22]In all barcharts, the mean is taken over the medians of each experimental condition (including Participant). Error bars represent the corresponding 95 % confidence interval.

Model	Technique	Parameters	r^2
Our model	$ContHead_L$	$596 + 390 \times ID_C + 531 \times ID_P$	0.92
	$ContPad_L$	$1410 + 296 \times ID_C + 482 \times ID_P$	0.98
	$DiscHead_L$	$-80 + 590 \times ID_C + 686 \times ID_P$	0.72
	$DiscPad_L$	$938 + 481 \times ID_C + 486 \times ID_P$	0.98
Fitts' law	$ContHead_L$	$581 + 421 \times ID$	0.90
	$ContPad_L$	$1369 + 359 \times ID$	0.93
	$DiscHead_L$	$276 + 569 \times ID$	0.72
	$DiscPad_L$	$1227 + 416 \times ID$	0.96

Table V: Parameters and accuracy of both Fitt's law and our model.

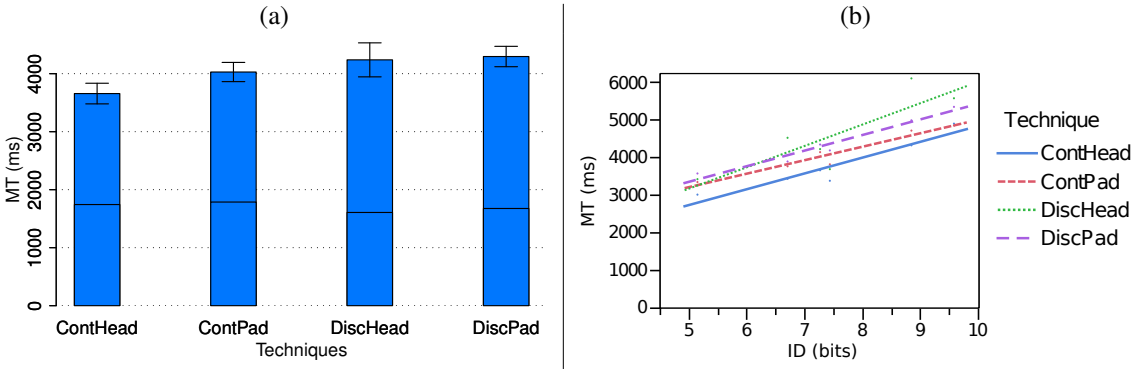


Fig. 27: (a) *MT* for each TECHNIQUE in the tablet experiment. The black intermediate lines in the bars show the time to the last mode switch to *Precise* mode. (b) Fitt's law regressions for all techniques in the tablet experiment.

use or that participants were unsure when to switch to *Precise* mode despite our *discretization*. Interestingly, $ContPad_L$ has the highest througput in both models, indicating that participants were quite efficient using it; however, it was slower than $ContHead_L$, at least for the Indices of Difficulty tested in this experiment.

Error rate. The overall error rate is 5.9 %. A multiway ANOVA reveals no effect of TECHNIQUE on the error rate ($F_{3,33} = 0.61$, $p = 0.6151$) and no significant interaction. Again, as expected, we find a significant effect of WIDTH on error rate ($F_{1,11} = 11.5$, $p = 0.0061$): 8.4 % for the small target size and 3.4 % for the larger one. We also measure a significant effect of AMPLITUDE, although the effect size is very small ($F_{2,22} = 0.69$, $p = 0.0240$): 4.9 % for the largest amplitude, 6.9 % for the medium one and 5.9 % for the short one.

Coarse vs Precise mode. As illustrated in Fig. 27-a, the time spent in the *Coarse* phase is slightly shorter with *Discrete* techniques than with *Continuous* techniques. However, this difference is not statistically significant – (H1) is not supported – and is not large enough to make *Discrete* techniques more efficient than *Continuous* ones. Indeed, the time spent in *Precise* mode is systematically longer with the *Discrete* techniques. We attribute this to the fact that the last mode switch is performed 150 mm away from the target on average with *Discrete* techniques, compared to 67 mm with $ContHead_L$ and 71 mm with $ContPad_L$.

$ContHead_L$ and $ContPad_L$ feature very similar *Coarse* pointing times (*CMT*) and distance-to-target at mode-switch time. $ContHead_L$'s shorter task time is mainly due to better performance in the *Precise* pointing phase. We observe that peaks in velocity occur earlier with $ContHead_L$ than with $ContPad_L$ (average *VPT* of 462 ms vs. 735 ms), although the velocities are not significantly different (0.23 vs. 0.30 $m.s^{-1}$). The distances to the target when switching mode are similar for *Head*- and *Pad*-based techniques, and their acceleration curves have the same input characteristics (V_{min}, V_{max} and $ratio_{inf}$). We thus expected the velocity peaks of $ContHead_L$ and $ContPad_L$ to occur at similar times, yet participants required more time to go "full-speed" with $ContPad_L$. This could be caused by the cognitive cost of switching between two very different control-display ratios while using the same input device. This supports (H2), suggesting that the cost of the mode switch is indeed lower for *Head*-based techniques than for *Pad*-based ones.

Note also that with the *Pad* techniques, participants removed two fingers for mode switching (vs. removing only one finger and continuing to point) in about 54.5 % of all cases (52.8 % for *ContPad* and 56.1 % for *DiscPad*). This might also explain the higher cost of the mode switch for *Pad*-based techniques.

6.3.2. Qualitative Results. Overall, participants preferred to use *Head*-based techniques (10 out of 12) and *Continuous* techniques (8 out of 12). 7 participants ranked $ContHead_L$ first, 4 ranked $DiscHead_L$ first

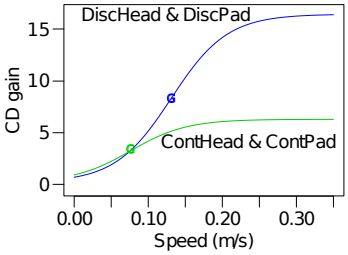


Fig. 28: Transfer functions used for the *Precise* modes in the smartphone experiment (parameters values in Table IV).

and 2 ranked $ContPad_L$ first. However, there were no strong complaints about any particular technique, except for one participant who clearly stated that he disliked *Discrete* techniques.

Several participants complained about the lack of tactile feedback, making it difficult to know when the fingers where leaving the pointing zone. They expressed the need for some sort of physical border, such as the border of a touchpad on a laptop. Only one participant answered that holding the tablet for 40 minutes caused fatigue when we inquired about this potential issue.

6.4. Smartphone Experiment

This experiment used a smartphone with a resolution of 480×320 (Apple iPod Touch, 115 g, $11 \times 6.2 \times 0.8$ cm, screen diagonal: 8.9 cm). Participants held the device in landscape orientation. The pointing zone used the top 480×166 pixels (75×26 mm), yielding an input resolution of 6.4 dot/mm. The transfer functions are shown in Fig. 28.

We took the feedback about the lack of tactile feedback by the participants in the previous experiment into account. We attached a thick tape delimiting the pointing zone (3 mm wide, 0.8 mm thick, see Fig. 31) to the device. Preliminary tests showed that the tape made it much easier to find the input zone eyes-free and to know in advance when the finger is about to leave the input zone while pointing.

Results are very similar to those of the tablet experiment. We get similar error rates, and an overall subjective preference for $ContHead_S$.

Movement time. The ANOVA reveals a significant effect of TECHNIQUE on *MT* ($F_{3,33} = 12.1$, $p < 0.0001$) and, as in the first experiment, no interaction of TECHNIQUE with WIDTH, AMPLITUDE and WIDTH $\times$ AMPLITUDE. A post-hoc t-test with Bonferonni correction shows that *ContHead* is significantly faster than all the other techniques (all p's < 0.0001), with a speed up of 13 % against *ContPad*, 12 % against *DiscHead* and 16 % against *DiscPad*. Regarding WIDTH and AMPLITUDE we find results similar to the smartphone experiment for *MT*. Movement time, split between the coarse and precise phases, is shown in Fig. 29-a using the same scale as in Fig. 27-a.

Fitts' law vs. our model. We used the same values for d (Eq. (14)) as in the previous experiment. Fitts' law fits the data better than in the tablet experiment, though our model (Eq. (28)) still produces higher r^2 scores (Table VI).

Model	Technique	Parameters	r^2
Our model	$ContHead_S$	$472 + 403 \times ID_C + 547 \times ID_P$	0.97
	$ContPad_S$	$969 + 423 \times ID_C + 535 \times ID_P$	0.97
	$DiscHead_S$	$879 + 463 \times ID_C + 472 \times ID_P$	0.97
	$DiscPad_S$	$1428 + 378 \times ID_C + 440 \times ID_P$	0.95
Fitts' law	$ContHead_S$	$464 + 434 \times ID$	0.95
	$ContPad_S$	$966 + 437 \times ID$	0.96
	$DiscHead_S$	$1162 + 402 \times ID$	0.94
	$DiscPad_S$	$1702 + 359 \times ID$	0.91

Table VI: Parameters and accuracy of both Fitt's law and our model.

The regression lines have relatively similar slopes, much more so than in the tablet experiment. The values of $r = b_p/b_c$ are similar to the tablet experiment: 1.02 and 1.16 for *Discrete* techniques, 1.26 and 1.36 for *Continuous* techniques.

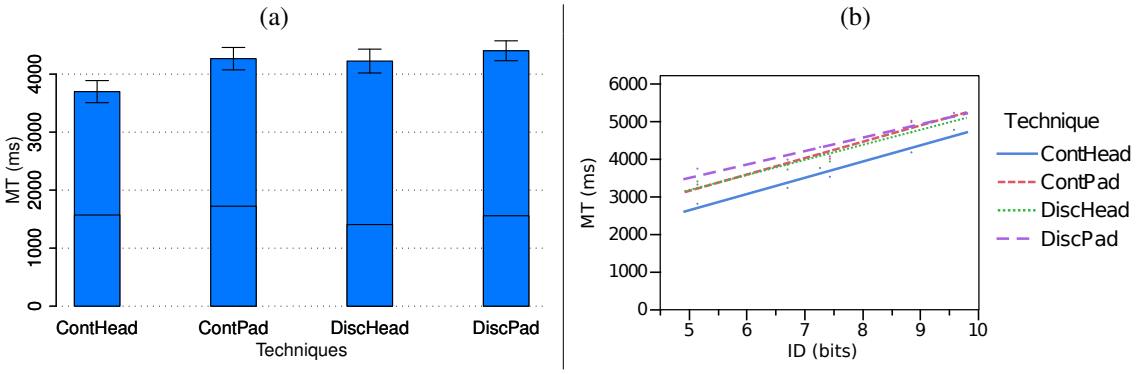


Fig. 29: (a) *MT* for each TECHNIQUE in the smartphone experiment. The black intermediate lines in the bars show the time to the last mode switch to *Precise* mode. (b) Fitt's law regressions for all techniques in the smartphone experiment.

Coarse vs. Precise mode. Regarding the coarse pointing phase, this time we observe a significant effect of TECHNIQUE ($F_{3,33} = 5.39$, $p = 0.0039$). $ContHead_S$ is significantly faster than all other techniques, and $DiscPad_S$ and $ContHead_S$ are significantly faster than $ContPad_S$ (when considering the *Coarse* phase only). With the smaller pointing zone used in this experiment, we do observe the hypothesized advantage for discrete techniques (H1). However, the time gained during the coarse phase is again not sufficient to make *Discrete* techniques faster than *Continuous* ones.

Regarding the precise (and post mode-switch) phase, we note that the distance to the target at the last mode switch is 83 mm and 88 mm for *ContHead* and *ContPad*, and 149 mm and 179 mm for *DiscHead* and *DiscPad*. The difference between continuous and discrete techniques shows that the time saved in the coarse phase by the discrete technique does not lead to better performance for the full pointing task.

In the case of *ContHead* and *ContPad* the above distances are very similar, but *ContHead* is faster than *ContPad* (even though they use the same transfer function), widening the gap that already exists in the *Coarse* phase. As in the tablet experiment, it seems that the cost of the mode switch is higher for *ContPad* than for *ContHead*. Indeed, as in the tablet experiment, the velocity peak comes earlier in the precise phase with *ContHead* than with *ContPad* (485 ms vs 773 ms for peaks of 0.33 and 0.29 $m.s^{-1}$). Therefore, as in the tablet experiment, (H2) is supported.

6.4.1. Subjective Results. In this experiment 8 participants (out of 12) ranked the *ContHead* technique first, 3 participants ranked *DiscHead* first and one participant ranked *DiscPad* first. This is similar to the tablet experiment, however no participant preferred $ContPad_S$ in this experiment while two of them ranked $ContPad_L$ first in the tablet experiment (where *DiscPad* was not ranked first). Only one participant (not the same as in the tablet) found that holding the smartphone during 40 minutes caused fatigue.

Surprisingly, this experiment was considered more tiring by the participants. Despite the smartphone being lighter and smaller than the tablet, it required a different grip and, unlike the tablet, the body could not be used to support the weight of the device.

6.5. Discussion

The first result of these studies is that beyond the relative differences in performance and user preference, the four techniques that we designed enabled participants to acquire very small targets from a distance relatively easily, with an average pointing time under 4.5 seconds and average error rate under 6 %. This is very encouraging considering the small size of the touch zones (72.5 and 19.5 cm^2). These studies also show that leveraging natural head movements yields very good performance. When the orientation of the head is not available, dividing difficult pointing tasks into two subtasks performed on the same input surface also leads to good performance, confirming the findings of our first experiment (Section 5).

6.5.1. Comparing Tablet and Smartphone Input. Since we did not counterbalance the order of pointing zone sizes (all participants performed the tablet experiment first), and since we added tactile feedback using tape to delimit the zone in the second experiment, we cannot formally compare overall performance across the results of the two experiments. However, we can make four informal observations:

(1) The movement time difference between *ContHead* and *ContPad* is larger for the smartphone (13 %) than for the tablet (9 %), which could be due to (i) the additional haptic feedback provided by the tape

surrounding the touch area in the Smartphone experiment and/or (ii) the lower size of the input zone causing smaller repositioning movements between the *Coarse* and *Precise* phases.

(2) $ContPad_S$ is not significantly faster than $DiscPad_S$, as opposed to $ContPad_L$ vs. $DiscPad_L$. This is partly explained by the fact that in the range of Fitts' IDs that we evaluated, the fitting curves of *ContPad* and *DiscPad* tend to diverge in the Tablet experiment and converge in the Smartphone experiment (Fig. 27-b and 29-b).

(3) Fig. 27-a and 29-a suggest that *DiscHead* performed better with the smartphone than with the tablet, relative to *ContPad* and *DiscPad*. These observations suggest that as we had anticipated, *Head*-based techniques are at an advantage with smaller pointing zones.

(4) *Continuous* techniques have higher slopes with the smartphone, as opposed to *Discrete* techniques which show lower slopes. If this effect holds at higher IDs, e.g. when performing subpixel pointing with lenses, *Discrete* techniques could outperform *Continuous* techniques.

6.5.2. Cost of Switching in Discrete Techniques. In Section 5, we hypothesized that participants tended to switch mode closer to the target than needed when either (1) the *Precise* phase seemed difficult (small target size), or (2) the *Coarse* phase was short enough (small amplitude) to spend extra time in *Coarse* mode and switch to *Precise* closer to the target. We expected this decision process to be cognitively demanding and time consuming.

To address this problem we investigated *Discrete* techniques that simplify the decision by leaving the user little or no choice about the switch location. However in both the Tablet and Smartphone experiments the effect of discretization on Dual-Precision pointing performance was either neutral or negative. This suggests that the mode-switching strategy observed in Section 5 is more efficient than forcing the location and time of the mode switch, at least within the range of IDs that we tested.

We studied only one size for the discretization: the cells corresponded to the screens of our ultra-wall. Future work will investigate the effect of the size and number of the cells, similar to the theoretical analysis of L in Section 4.1.

7. STUDY 3: POINTER ACCELERATION VS. DUAL-PRECISION TECHNIQUES

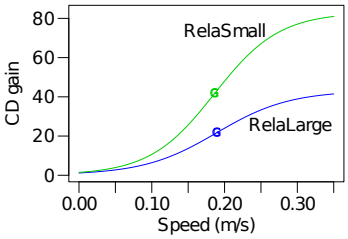


Fig. 30: Transfer functions used for the third study (see Table VII for parameter values).

In Section 5 we studied the design of unimanual mid-air pointing techniques that can accommodate very difficult pointing tasks. We explored Dual-Precision techniques, following the calibration methods defined in Section 4.1, and applied the Pointer Acceleration transfer function presented in Section 4.2 to angular mid-air input. In Section 6 we explored the design of Dual-Precision mid-air pointing techniques that used minimal portions of a touch-screen, either by using head orientation as an additional input channel or by varying the number of fingers in contact with the touch area. These designs and results provide useful data on which technique to use depending on the other tasks that must be performed during or after pointing and on the available input devices and channels of a given platform.

We now apply our Pointer Acceleration calibration method to mode-less, touch-based relative pointing, and compare it with the best techniques of our previous experiments, *Laser+Gyro* and *ContHead*, and to one state-of-the-art baseline technique for mid-air pointing, Smoothed Pointing (*SmoothPoint*).

7.1. Techniques

We implemented *RelaSmall* and *RelaLarge*, two purely relative, trackpad-like techniques that differ only in the size of the input area and the transfer function, optimized for this size (Fig. 30). Compared to the functions used in the two experiments of Section 6, these two functions feature a much higher CD_{max}

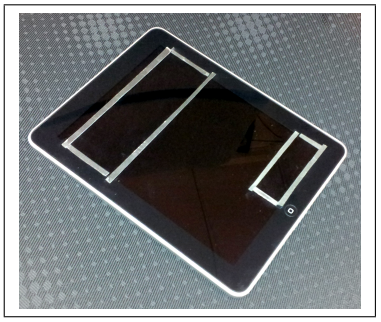


Fig. 31: The two input zones delimited by thick removable tape.

that allows traveling much larger distances without too much clutching, but also drastically increases their maximum slope.

For *ContHead*, we decided to use $ContHead_S$ (Fig. 22-A) on a tablet instead of a smartphone. This avoids introducing a confound due to the device, and the performance in the two previous experiments was similar to $ContHead_L$. This also makes sense beyond this laboratory study, since using a smaller pointing zone on a tablet means that the device can accommodate more widgets for other purposes.

For *LaserGyro*, unlike in Section 5, we computed a transfer function for the *Precise* mode using the procedure described in Section 4.2. Its parameter values are reported in Table VII, together with those of *RelaLarge* and *RelaSmall*.

Technique	V_{min}	V_{max}	CD_{min}	CD_{max}	λ	$ratio_{inf}$
RelaLarge $_{148 \times 49 \text{ (mm)}}$	0.006 m/s	0.37 m/s	0.22	43.1	.02 s/mm	0.5
RelaSmall $_{75 \times 26 \text{ (mm)}}$	0.003 m/s	0.19 m/s	0.27	83.0	.023 s/mm	1.0
LaserGyro	9.17 deg/s	258.98 deg/s	0 m/deg	3.09 mm/deg	.017 s/deg	0.4

Table VII: Transfer function parameter values.

Like *GyroAcc* and *ContHead*, *SmoothPoint* [Gallo and Minutolo 2012] also combines ray-casting and relative pointing. However, the transition between the two modes is progressive, based on a transfer function that depends on input velocity. The authors present a method to tune this function, but pilot tests in our environment revealed that this method does not scale to pixel densities such as those typically encountered on ultra-walls: the large difference between the minimum and maximum CD gain values causes the slope of the function to be very steep, making the resulting technique far too jerky to select small targets such as those considered here. We transposed the calibration method described in [Gallo and Minutolo 2012] to our context[23] to the best of our abilities, iterating until the technique eventually enabled us to achieve the pointing tasks featured in our experiments.

7.2. Experiment

In this experiment, we tested five techniques (TECHNIQUE): $ContHead_S$, *RelaSmall*, *RelaLarge*, *SmoothPoint* and *LaserGyro*. The apparatus, design, and procedure were exactly the same as in the experiments of Section 6. We added a physical border around the large and small pointing zones on the tablet to limit the need to look at the input device by providing tactile feedback when the fingers were about to leave the zone. We used a 5×5 Latin square to balance the techniques. 15 participants served in the experiment. 10 of them had participated in the experiments of Section 6, 5 were new and assigned to the same Latin square.

7.2.1. Quantitative Results. As in the previous experiments we removed a few outliers (1.28%).

Movement time. Fig. 32 shows movement time *MT* by WIDTH and AMPLITUDE. For $ContHead_S$ the results are very close to those of the previous experiments. We observe that *SmoothPoint* performs poorly compared to all the other techniques (significantly so, for each WIDTH and AMPLITUDE condition). For brevity, we do not report the results of *SmoothPoint* in post-hoc tests, even though it was of course included.

An ANOVA reveals a significant effect of TECHNIQUE on *MT* ($F_{4,56} = 20.5$, $p < 0.0001$). A post-hoc t-test with Bonferonni correction shows that $ContHead_S$, *LaserGyro* and *RelaLarge* are all significantly faster than

[23] We did not use our logistic function because the function described in [Gallo and Minutolo 2012] is part of its contribution.

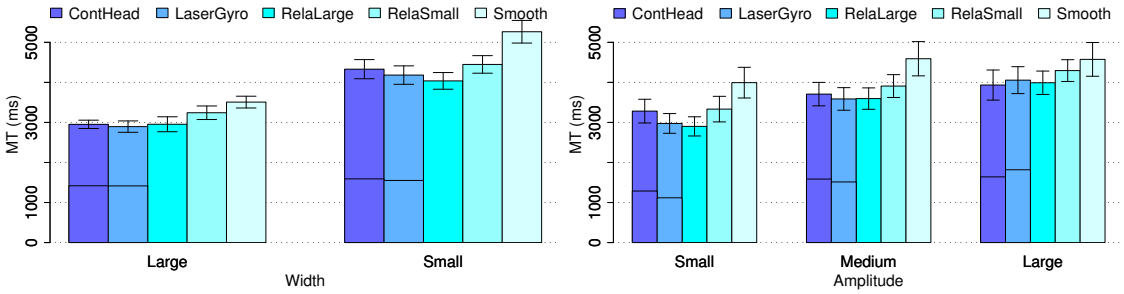


Fig. 32: *MT* for each TECHNIQUE by WIDTH and by AMPLITUDE for the last experiment.

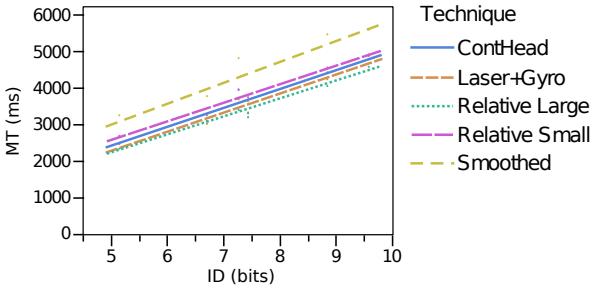


Fig. 33: Fitt's law regressions for all techniques in the baselines experiment.

RelaSmall (with speed-ups of 5.4%, 8% and 9%, respectively). We find no significant difference between $ContHead_S$, *LaserGyro* and *RelaLarge*.

This time, the ANOVA reveals significant interactions TECHNIQUE × WIDTH ($F_{4,56} = 5.90$, $p = 0.0005$) and TECHNIQUE × AMPLITUDE ($F_{8,112} = 2.60$, $p = 0.0122$) on *MT*. One cause for these interactions is *SmoothPoint*, which is slower for small targets than for larger ones, and faster for the largest amplitude. We also observe (Fig. 32) that the two relative techniques, *RelaSmall* and *RelaLarge*, are faster for the small width and the small amplitude. Indeed, post-hoc tests show that (i) for small targets, the only significant difference is that *RelaLarge* is faster than *RelaSmall*, and for large targets, $ContHead_S$ and *LaserGyro* are also significantly faster than *RelaSmall*; (ii) for the small amplitude the only significant difference is that *RelaLarge* is faster than *RelaSmall* and $ContHead_S$, while for the large amplitude the only significant difference is that both *RelaLarge* and $ContHead_S$ are significantly faster than *RelaSmall*.

Fitts' Law vs our model. Table VIII shows the regressions for Fitts' Law and our model, when applicable. We ran our model on *SmoothPoint* since it uses two pointing modes, even though the switch occurs progressively and implicitly.

Technique	Fitts' law	r^2	Our model	r^2
ContHead	$-177 + 518 \times ID$	0.88	$-68 + 354 \times ID_C + 754 \times ID_P$	0.98
LaserGyro	$-342 + 523 \times ID$	0.95	$-310 + 458 \times ID_C + 681 \times ID_P$	0.99
RelaLarge	$-234 + 493 \times ID$	0.99		
RelaSmall	$24 + 509 \times ID$	0.97		
SmoothPoint	$122 + 573 \times ID$	0.8	$318 + 301 \times ID_C + 905 \times ID_P$	0.98

Table VIII: Parameters and accuracy of both Fitt's law and our model.

Our model fits data better than Fitts' law for techniques using two modes, especially *ContHead* and *SmoothPoint*. We believe this is because the single slope parameter in Fitts' law (Eq. (13), p. 12) cannot convey precise information about both pointing phases. For example, we observe that *ContHead* has a throughput twice as high in *Coarse* mode than in *Precise* mode ($r = 2.13$ in Eq. (18), p. 13) and that using head orientation for coarse pointing is more efficient than laser pointing ($b_C = 354$ vs. 458 ms/b for *ContHead* vs. *LaserGyro*). We also observe a high throughput in *Coarse* mode (low b_c) for *SmoothPoint*, indicating that it is very efficient for coarse pointing; however, as hinted by its low throughput in *Precise* mode (high b_p), it is far too slow to acquire small targets. This corresponds to our own observations: *SmoothPoint* was very hard to control at low speeds.

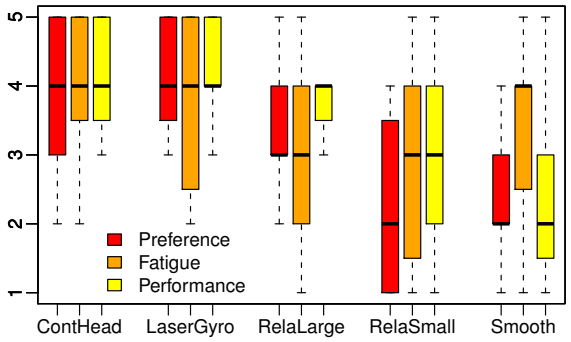


Fig. 34: Preference, fatigue and (self-reported) perceived performance for each TECHNIQUE on a five-point Likert scale (5 is best, 1 is worst). Bold lines show the median, boxes show the lower and upper quartiles and the whiskers show the 1.5 × inter-quartile range.

Error rate. The overall error rate is 9.6%. *ContHead$_S$*, *RelaSmall* and *RelaLarge* feature low error rates (3.90%, 3.52% and 4.04%) with only marginal differences between large and small targets (2.5% vs 5.17%). *LaserGyro* and *SmoothPoint* also feature low error rates for large targets (0.70% and 5.60%), but the error rate rises dramatically for small targets: 25.6% and 43.0%. Based on our pilot studies, we did not expect such an increase. The problem comes from the fact that clicking with the handheld wireless mouse causes the cursor to sometimes leave the target. This led users to click multiple times in quick succession to acquire the target. However, the high error rate of *LaserGyro* did not cause a big penalty in pointing time compared to the touch-based techniques, as opposed to *SmoothPoint*. *SmoothPoint* caused twice as many *Misses* on average (mean 0.87, stdev 1.5) as *LaserGyro* (mean 0.4, stdev 0.75), and 74 % of the errors with *LaserGyro* were caused by a single *Miss* per trial vs. 49 % for *SmoothPoint*. This suggests that *LaserGyro* was easier to control with small targets.

7.2.2. Qualitative Results. At the end of the experiment, we asked participants to rank the techniques on a five-point Likert scale in terms of preference, fatigue and perceived performance. Fig. 34 summarizes the results in a boxplot.

Kruskal-Wallis tests revealed a significant effect of TECHNIQUE on Preference ($\chi^2_4 = 28.5$, $p < 0.0001$) and Performance ($\chi^2_4 = 23.4$, $p = 0.0001$), but not on Fatigue. Post-hoc tests using Mann-Whitney tests with Bonferroni correction show that (i) *ContHead$_S$* and *LaserGyro* were preferred to both *RelaSmall* and *SmoothPoint*; (ii) *RelaLarge* was preferred to *SmoothPoint*; (iii) *ContHead$_S$*, *LaserGyro* and *RelaLarge* were perceived as faster than *SmoothPoint*; and (iv) *LaserGyro* (and *ContHead$_S$*) was perceived as (marginally) faster than *RelaLarge*.

7.3. Discussion and Design Guidelines

Overall, the results show that our techniques, designed according to the guidelines introduced in Section 4, fare better than state-of-the-art mid-air pointing techniques and enable users to perform pointing tasks with high a high Index of Difficulty. More precisely, the three techniques that perform best in terms of movement time and preference are *RelaLarge*, *ContHead* and *LaserGyro*. While there is no significant difference in movement time among them, each technique has its own strengths and limitations, making it more suitable to specific contexts of use.

While relative pointing is not novel, making it work efficiently in such challenging contexts is an interesting result. Indeed, existing functions, even elaborate ones such as that of *SmoothPoint*, were designed for lower-resolution environments, and fare poorly with the high Fitts' IDs considered here. We suspect that this is because *SmoothPoint* uses a sine-based function that does not allow control of the abscissa and slope of its inflexion point: for a given input and output range, the method provides a single function that can be very steep when the output range is much larger than the input range. Indeed, the cursor was much jerkier with *SmoothPoint* than with any other technique evaluated in this study. *RelaLarge*, based on the transfer function and calibration method introduced in Section 4.2, provides pointing performance that matches that of more elaborate techniques. It is straightforward to implement and does not require special equipment to track the 3D position and orientation of the device or user.

RelaSmall provides enough precision to perform bivariate pointing tasks of difficulty up to 9.5 bits with a pointing zone of 20 cm^2 only (approximately $1/4^{th}$ of *RelaLarge*'s surface area). We did not expect such good performance. Given that the size of the pointing zone of *RelaLarge* preclude its use on smartphones,

RelaSmall can be an appealing option. Indeed, the technique will only incur a 5-to-10-percent performance cost when compared to more efficient techniques, which can be considered an acceptable tradeoff when only small handheld devices are available, or when a large portion of the handheld's screen real-estate must be allocated to additional interface widgets.

This decrease in performance can be avoided by using *ContHead*, which achieves the same level of performance as *RelaLarge* but on a much smaller input area, equivalent to that of *RelaSmall*. *ContHead* should be considered when the task and context of use require many additional interface widgets or when only smartphones are available, provided that tracking the location and orientation of the head is possible.

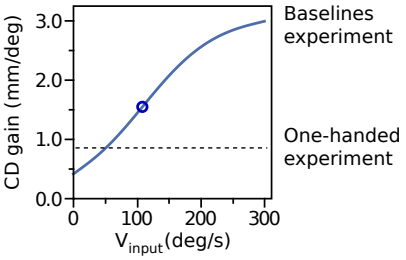


Fig. 35: Transfer functions used for the *Precise* mode of *LaserGyro* in the unimanual experiment (dashed) and in the baselines experiment (plain blue). The blue circle represents the inflexion point of the transfer function.

Finally, our implementation of *LaserGyro* causes many more errors than the other techniques for small values of WIDTH, because the tremor caused by pressing a physical button is sufficient to miss small targets given the current transfer function. *LaserGyro* did not cause that many errors in the unimanual experiment (Section 5) because we used a low constant CD gain (.89 mm/deg) compared to the range used in the baselines experiment (mean 1.9 mm/deg in the $[V_{min}, V_{max}]$ interval, Fig. 35). This raises the question of whether the calibration of the transfer function used in *Precise* mode should also be adapted to the type of input device and to the clicking mechanism, for exemple by setting PS_{device} to the amplitude of the cursor movement corresponding to the aforementioned tremor in Equations (9), (20) and (22). For touch-based techniques, using a tap for clicking did not affect the cursor location.

8. IMPLICATIONS FOR DESIGN

In this work we started to map the design space of pointing techniques for ultra-walls. We studied two categories of techniques: Pointer Acceleration (PA) and Dual-Precision (DP). Tables IX and X describe the subset of the design space that we explored in terms of the *input dimensions* that were sensed and their *mapping* to cursor movements. For Dual-Precision techniques, we also list the transition mechanism.

Study	Technique	Input	Mapping
1	*GyroAcc*	Relative arm rotation	Variable gain
3	*RelaSmall*	Relative touch (small area)	Variable gain
	RelaLarge	Relative touch (large area)	Variable gain
	SmoothPoint	Ray-casting (arm)	Direct + Variable gain

Table IX: Design space of Pointing Acceleration techniques (see text for explanations).

Study	Technique	Coarse input	Coarse mapping	Transition	Precise input	Precise mapping
1	*LaserGyro* (1)	Ray-casting (arm)	Direct	Button	Rel. arm rotation	Constant gain
	Laser+Position	Ray-casting (arm)	Direct	Button	Rel. arm translation	Constant gain
	Laser+Track	Ray-casting (arm)	Direct	Touch	Rel. touch	Constant gain
2	$ContHead_L$	Abs. head rotation	Direct[24]	Touch	Rel. touch (large area)	Variable gain
	$ContHead_S$	Abs. head rotation	Direct[24]	Touch	Rel. touch (small area)	Variable gain
	$ContPad_L$	Abs. touch (large area)	Indirect	#fingers	Rel. touch (large area)	Variable gain
	$ContPad_S$	Abs. touch (small area)	Indirect	#fingers	Rel. touch (small area)	Variable gain
3	*LaserGyro* (3)	Ray-casting (arm)	Direct	Button	Rel. arm rotation	Variable gain

Table X: Design space of Dual Precision techniques (Abs. stands for Absolute, Rel. for Relative - see text for additional explanations). The precise mapping of *LaserGyro* changed between studies (1) and (3).

[24]The *Coarse* mode of the *-Head* techniques is a linear mapping between absolute orientation input and absolute cursor location, therefore not strictly a direct mapping (see Section 6.1.2 p. 29). However, we tuned it so that it was *perceived* as one.

When input is absolute, the mapping can be either *Direct*, when pointing at the real target, or *Indirect*, when pointing on a separate surface such as a handheld tablet. When input is relative, the mapping is specified by the type of gain applied to the sensed motion: *Constant* or *Variable*. Ray-casting requires both absolute position and rotation sensing of the hand. *SmoothPoint* is a special case that applies a variable CD-gain to absolute ray-casting input in its precision mode.

Generally speaking, sensing an absolute dimension (absolute input) is more demanding than sensing its changes (relative input). For example, tracking the mid-air position and orientation of the arm and hand typically requires an external motion tracking system while tracking the variations of its position and orientation can be done with an accelerometer held in the hand or attached to the arm.

The rest of this section builds on this design space and on the results of our studies to provide guidelines on how to choose a mid-air pointing technique given the input capabilities of the interactive platform (e.g. with or without motion tracking) and the interaction requirements of its users (e.g. tablet-based interactions). We then discuss two high-level psycho-motor factors that seem to affect the usability and performance of pointing techniques and that can suggest new directions to design more efficient pointing techniques.

8.1. Choosing among pointing techniques

			Input requirements				**Subjective evaluation**
Study	Technique	Type	Involved limbs	Touch area	Angular sensing	Position sensing	Ranking
1	*LaserGyro*	DP	1 hand	n/a	Absolute	Absolute	1/4
	Laser+Position	DP	1 hand	n/a	Absolute	Absolute	2/4
	GyroAcc	PA	1 hand	n/a	Relative	n/a	3/4
	Laser+Track	DP	1 hand	Small	Absolute	Absolute	3.5/4
2	$ContHead_L$	DP	2 hands + head	Large	Absolute	Absolute	1/4
	$ContHead_S$	DP	2 hands + head	Small	Absolute	Absolute	1/4
	$ContPad_L$	DP	2 hands	Large	n/a	n/a	3/4
	$ContPad_S$	DP	2 hands	Small	n/a	n/a	3/4
3	*LaserGyro*	DP	1 hand	n/a	Absolute	Absolute	2/5
	$ContHead_S$	DP	2 hands + head	Small	Absolute	Absolute	2/5
	RelaLarge	PA	2 hands	Large	n/a	n/a	3/5
	RelaSmall	PA	2 hands	Small	n/a	n/a	4/5
	SmoothPoint	PA	1 hand	n/a	Absolute	Absolute	4/5

Table XI: Main input requirements and subjective ranking of the techniques evaluated in Sections 5, 6 and 7. *DP* stands for Dual-Precision, *PA* for Pointer Acceleration. *Angular* and *Position* sensing are the input requirements to capture the angle and position of the limbs. *Ranking* is the median preference score of the technique in the experiment (1 is best).

Beyond their relative performance ranking, all the techniques that we designed enabled users to select very small targets at a distance with good performance, i.e. less than 5 seconds for most techniques, with different input requirements. Table XI summarizes these input requirements and Fig. 36 summarizes performance results. For movement time, we show the Fitts' law regressions[25]. For error rate, since in most cases target width had a significant effect on error rate (as opposed to the Fitts' ID or amplitude), we represent error rates for each target width.

Using these tables, interaction designers can select a technique depending on the input requirements of the task and the input capabilities of the platform, as well as the maximum pointing difficulty and user preferences. For example, in situations without absolute real-time motion tracking, *GyroAcc* can be used for easier tasks ($ID < 8$ bits, WIDTH$>$ 15 mm) and provide good performance while leaving one hand free for other interactions performed with another device; *RelaSmall* is the only technique allowing precise pointing on a small touch sensitive surface without absolute motion tracking; for wide ranges of IDs, *LaserGyro* and *ContHead* provide the best performance and user preference overall, but require absolute motion tracking; finally, despite its lower performance, *Laser+Track* is the only technique allowing unimanual interaction *and* additional touch-based input.

[25]As noted earlier, we may have obtained better fits with angle-based models [Kopper et al. 2010; Jota et al. 2010] for angular-based pointing techniques (*SmoothPoint*, *GyroAcc*) and modes (*Laser-* and *-Head* techniques). However, this would have prevented us from comparing these techniques to the other ones based on a common definition of task difficulty.

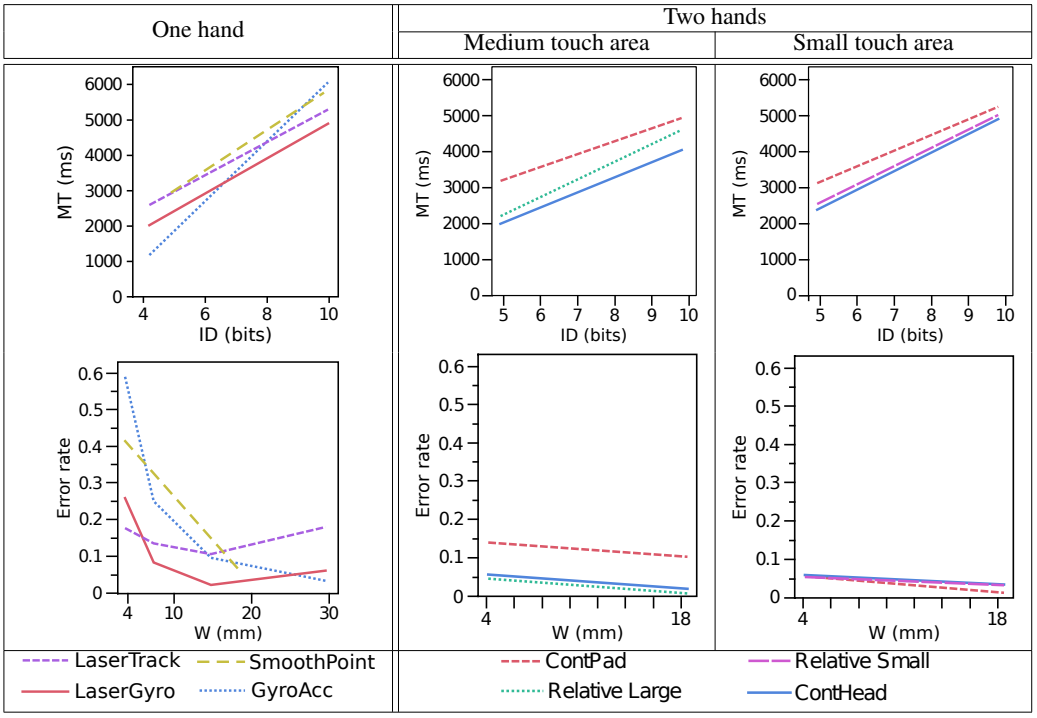


Fig. 36: Fitts' law regressions and error rates per target widths for our techniques, organized by input requirements. *Laser+Position* was discarded because of its similarity with *Laser+Gyro*; we only represent *LaserGyro* in the first study due to its low precision in the last one; *SmoothPoint* was added for comparison.

8.2. Psycho-motor aspects of high-performance pointing: Decision and Adaptation

The results of the experiments reported in Sections 5, 6 and 7 highlight two cognitive behaviors induced by the changing transfer function during pointing: *Decision* and *Adaptation*.

Decision is the cognitive cost incurred by explicitly breaking the continuity of a pointing task, e.g. when switching to a different transfer function. It occurs when users have to decide *whether*, *when* and *where* to switch transfer function for a given pointing task (defined by its width and amplitude) and pointing technique.

For example, we observed in Section 5 that when using a Dual-Precision technique, the width and amplitude of the pointing task had an effect on the duration of its *Coarse* phase and on the distance of the mode switch to the target. In some of the easier conditions, participants even tried to select the target in *Coarse* mode only, without switching mode.

Our hypothesis, introduced in Section 5, is that the process of deciding where and when to switch modes was part of a higher-level, subconscious process of optimizing time and effort. Participants took more time and were more careful about the location of the mode switch (i) when the *Coarse* phase seemed easy, "saving" time that could be used to ease the *Precise* phase, or (ii) when the *Precise* phase seemed difficult, i.e. when the target was small: more time was spent in *Coarse* mode to lower the amplitude of the *Precise* phase. The no-switch behavior is an extreme case of this optimization process, in which the *Coarse* mode is able to bring the cursor so close to the target that the time needed to switch modes and then acquire the target with the slower *Precise* mode might appear greater than the time needed to acquire the target with the less precise *Coarse* mode.

Similar effects have been observed in previous studies. For example, [Mandryk and Lough 2011] studied the effect of the intended use of the target on the performance of pointing tasks with the same Fitts' ID. The task consisted of acquiring a target, then performing a second subtask that could be either click, click and acquire a second target, press and "throw" the target towards an edge of the screen, or drag the target onto another target. The authors observed that the changes in movement time could be attributed to the differences in the secondary, precise pointing phase, and not the initial ballistic phase. In accordance with our results, the authors hypothesize that the movement time increases with the precision of the objective, and that motor planning and control of the second subtask is encapsulated within the second phase of the first subtask. [Quinn et al. 2011] observed a similar effect when studying multiple trajectory pointing

methods. Participants did not always perform better with methods that shorten the trajectory to a target, such as cursor wrapping [Huot et al. 2011] or Ninja cursors [Kobayashi and Igarashi 2008]. The authors suggest that selecting the best trajectory among several possibilities caused search and decision times that eventually cancelled out the potential performance advantage.

Adaptation is the cognitive and motor adjustment to a change in transfer function. It occurs after a sudden change that requires the user to adapt to a transfer function whose parameters are significantly different from that of the previous one. This change can be due to a change of input, e.g. switching between two input devices or modes to control cursor displacements, or a change of control, e.g. transitioning between very different CD gains.

From our experimental results we hypothesize three possible causes of *Adaptation*:

(1) *Steeper slopes in transfer functions require more cognitive and motor adjustments between their lower and higher levels* – We observed in Section 7 that steeper slopes lead to poorer performance. Indeed, *SmoothPoint* was the worst technique for all measures, followed by *RelaSmall* for movement time.

(2) *Different transfer functions are better controlled with different limbs* – When controlled with one hand, switching from one transfer function to another constrains the starting point of the latter to the last point of the former. Users thus have a limited operating range and must adapt their body movements accordingly, e.g. by clutching. In the *Tablet* experiment in Section 6, participants spent more time in *Precise* mode with the *Pad*-based techniques than with the *Head*-based techniques, despite the morphological similarity between the *Coarse* and *Precise* control of the *Pad*-based techniques. In both experiments of that section, participants also took more time to reach their peak velocity in *Precise* mode with *ContHead* than with *ContPad*, despite the similar proximity of the cursor to the target at mode-switch time.

(3) *When controlled with the same limb, different transfer functions are better controlled with similar input channels* – For example, switching from angular to linear control is more cognitively demanding than if the whole pointing movement is angular or linear. We observed in Section 5 that users spent less time in *Precise* mode with *Laser+Gyro* than with *Laser+Position* and *Laser+Track*, even though the average amplitudes of the *Precise* sub-task were not significantly different. Indeed, angular control (*Precise* mode of *Laser+Gyro*) is morphologically closer to ray-casting (*Coarse* mode of all three techniques) than mid-air position control and touch input (*Precise* modes of *Laser+Position* and *Laser+Track* respectively).

The controlled experiments in Sections 5, 6 and 7 were intended to evaluate the performance and the morphological aspects of input techniques, not to investigate cognition and dynamic motor control. Future work is needed to assess these hypotheses.

9. CONCLUSION

In this paper we investigated pointing techniques for location-independent interaction on ultra-walls. These environments enable very large datasets to be visualized and manipulated by several users simultaneously. Their large physical size and high pixel density create new constraints for interaction: users may need to acquire very small targets from relatively far away, and other interactions must remain possible with the input devices at hand.

In Section 3, we explored the limits of existing pointing techniques and devices relative to the output capabilities of ultra-walls. We used Casiez *et al.*'s formulae to develop a theoretical framework for techniques with constant CD gain [Casiez et al. 2008]. As discussed in Section 3.2, the core problem of constant CD gain techniques is that the input expressiveness of either the device or user movements fails to match the range of pointing task difficulty (Eq. (11), p. 8). Pointing techniques therefore need to vary their CD gain, as demonstrated by the improved performance of Pointer Acceleration. In Section 4.1, we described *Dual-Precision techniques*, a family of techniques that provide two modes, each with a specific range of CD gains, for coarse and precise control. Switching between these modes requires an explicit action so that users can fully control the cursor movements, as opposed to Pointer Acceleration where this transition is implicit. We provided a theoretical analysis of such techniques as well as a method to calibrate the CD gain of each mode. We also developed a new method for calibrating the transfer functions of Pointer Acceleration techniques, based on the capabilities of both the user and the input device(s) as well as the characteristics of the pointing task.

In Sections 5, 6 and 7, we evaluated this framework by designing and implementing a number of mid-air pointing techniques with different gain-variation methods. These techniques were designed to perform very difficult pointing tasks with limited input requirements such as unimanual pointing or limited size of touch input area, which are common for mid-air interaction with large displays. Through a series of controlled experiments we showed that our techniques and their calibration methods meet these goals. Dual-Precision

techniques were always among the fastest techniques in all our studies and were generally preferred by participants. Our Pointing Acceleration transfer functions showed similar performance to Dual-Precision techniques, and tended to perform better for the easiest tasks, probably because of the constant mode-switch time and associated cognitive load of Dual-Precision techniques.

Rather than designating one technique as the "best" one, we provide a classification of these techniques according to their performance, input requirements and user preference in order for interaction designers to more easily select the one that best fits their needs. While the techniques introduced in this work can already be used in a variety of situations, our theoretical framework and the guidelines that we drew from the experimental results can be used by other researchers and interaction designers to design new pointing techniques adapted to their needs. Finally, from our analyses and studies, we gathered general insights about the sensorimotor aspects of highly-varying CD gains, which we summarized and discussed in Section 8.2.

Future work includes a more systematic validation of the models that we developed in Sections 3.2 and 4.1, and of the calibration methods presented in Sections 4.1 and 4.2. In particular, we will investigate how to better take clutching into account in the calibration of the transfer functions, taking into account the fact that the resulting increase in CD-gain range can counterbalance the additional time spent clutching [Nancel et al. 2015]. We will also evaluate our hypotheses on the effects of the Decision and Adaptation factors presented in Section 8.2, and on how they affect the design space of pointing techniques.

REFERENCES

Jonathan Aceituno, Géry Casiez, and Nicolas Roussel. 2013. How low can you go?: Human limits in small unidirectional mouse movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 1383–1386. DOI: http://dx.doi.org/10.1145/2470654.2466182 5, 16

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Trans. Automat. Control* 19, 6 (1974), 716 – 723. DOI: http://dx.doi.org/10.1109/TAC.1974.1100705 25

Christopher Andrews, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, 55–64. DOI: http://dx.doi.org/10.1145/1753326.1753336 1

Caroline Appert, Olivier Chapuis, and Michel Beaudouin-Lafon. 2008. Evaluation of pointing performance on screen edges. In *Proceedings of the working conference on Advanced Visual Interfaces (AVI '08)*. ACM, 119–126. DOI: http://dx.doi.org/10.1145/1385569.1385590 21

Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. 2010. High-precision magnification lenses. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, 273–282. DOI: http://dx.doi.org/10.1145/1753326.1753366 8

Mark Ashdown, Kenji Oka, and Yoichi Sato. 2005. Combining head tracking and mouse input for a GUI on multiple monitors. In *Extended abstracts on Human factors in computing systems (CHI EA '05)*. ACM, 1188–1191. DOI: http://dx.doi.org/10.1145/1056808.1056873 30

Robert Ball, Chris North, and Doug Bowman. 2007. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*. ACM, 191–200. DOI: http://dx.doi.org/10.1145/1753326.1753336 1, 6

Patrick Baudisch, Mike Sinclair, and Andrew Wilson. 2006. Soap: a pointing device that works in mid-air. In *Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06)*. ACM, 43–46. DOI: http://dx.doi.org/10.1145/1166253.1166261 9

Michel Beaudouin-Lafon, Olivier Chapuis, James Eagan, Tony Gjerlufsen, Stéphane Huot, Clemens Klokmose, Wendy Mackay, Mathieu Nancel, Emmanuel Pietriga, Clément Pillias, Romain Primet, and Julie Wagner. 2012. Multi-surface interaction in the WILD room. *IEEE Computer* 45, 4 (2012), 48–56. DOI: http://dx.doi.org/10.1109/MC.2012.110 1, 5, 21

Michel Beaudouin-Lafon, Stéphane Huot, Halla Olafsdottir, and Pierre Dragicevic. 2014. GlideCursor: Pointing with an inertial cursor. In *Proceedings of the working conference on Advanced Visual Interfaces (AVI '14)*. ACM, 49–56. DOI: http://dx.doi.org/10.1145/2598153.2598166 7

Renaud Blanch and Michaël Ortega. 2009. Rake cursor: improving pointing performance with concurrent input channels. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)*. ACM, 1415–1418. DOI: http://dx.doi.org/10.1145/1518701.1518914 31

Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch projector: mobile interaction through video. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, 2287–2296. DOI: http://dx.doi.org/10.1145/1753326.1753671 4

William Buxton, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. 2000. Large displays in automotive design. *IEEE Computer Graphics and Applications* 20, 4 (2000), 68–75. DOI: http://dx.doi.org/10.1109/38.851753 6

Bryan A. Campbell, Katharine R. O'Brien, Michael D. Byrne, and Benjamin J. Bachman. 2008. Fitts' law predictions with an alternative pointing device (wiimote®). *Human Factors and Ergonomics Society Annual Meeting Proceedings* 52, 19 (2008), 1321–1325. DOI: http://dx.doi.org/10.1177/154193120805201904 3, 4

Xiang Cao and Ravin Balakrishnan. 2003. Visionwand: interaction techniques for large displays using a passive wand tracked in 3D. In *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*. ACM, 173–182. DOI: http://dx.doi.org/10.1145/964696.964716 4

Géry Casiez and Nicolas Roussel. 2011. No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM, 603–614. DOI: http://dx.doi.org/10.1145/2047196.2047276 2, 4, 7, 11, 15

Géry Casiez and Daniel Vogel. 2008. The effect of spring stiffness and control gain with an elastic rate control pointing device. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '08)*. ACM, 1709–1718. DOI: http://dx.doi.org/10.1145/1357054.1357321 3

Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction* 23, 3 (2008), 215–250. DOI: http://dx.doi.org/10.1080/07370020802278163 7, 13, 15, 21, 46

Olivier Chapuis, Anastasia Bezerianos, and Stelios Frantzeskakis. 2014. Smarties: An input system for wall display development. In *Proceedings of the 32nd international conference on Human factors in computing systems (CHI '14)*. ACM, 2763–2772. DOI: http://dx.doi.org/10.1145/2556288.2556956 5, 18

Xing Chen and James Davis. 2000. Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays* 23, 5 (2000), 205–211. DOI: http://dx.doi.org/10.1016/S0141-9382(02)00039-2 3

Andy Cockburn, Philip Quinn, Carl Gutwin, and Stephen Fitchett. 2012. Improving scrolling devices with document length dependent gain. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 267–276. DOI: http://dx.doi.org/10.1145/2207676.2207714 15

Henrique Debarba, Luciana Nedel, and Anderson Maciel. 2012. Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision. In *IEEE Symposium on 3D User Interfaces (3DUI '12)*. IEEE, 125 –132. DOI: http://dx.doi.org/10.1109/3DUI.2012.6184196 20

Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47 (1954), 381–391. DOI: http://dx.doi.org/10.1037/h0055392 21

Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. 2006. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06)*. ACM, 211–220. DOI: http://dx.doi.org/10.1145/1166253.1166286 4, 11

Edward G. Freedman and David L. Sparks. 2000. Coordination of the eyes and head: movement kinematics. *Experimental Brain Research* 131, 1 (2000), 22–32. DOI: http://dx.doi.org/10.1007/s002219900296 29, 30

Scott Frees, G. Drew Kessler, and Edwin Kay. 2007. Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction* 14, 1 (2007), 2. DOI: http://dx.doi.org/10.1145/1229855.1229857 4, 15

Luigi Gallo and Aniello Minutolo. 2012. Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique. *International Journal of Human Computer Studies* 70, 4 (2012), 287–300. DOI: http://dx.doi.org/10.1016/j.ijhcs.2011.12.001 4, 9, 11, 12, 15, 19, 40

Yves Guiard and Michel Beaudouin-Lafon. 2004. Target acquisition in multiscale electronic worlds. *International Journal of Human Computer Studies* 61, 6 (2004), 875–905. http://dx.doi.org/10.1016/j.ijhcs.2004.09.005 21

Yves Guiard, Frédéric Bourgeois, Denis Mottet, and Michel Beaudouin-Lafon. 2001. Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. In *People and Computers XV - Interaction without frontiers (Joint proceedings of HCI 2001 and IHM 2001)*. Springer Verlag, 573–587. DOI: http://dx.doi.org/10.1007/978-1-4471-0353-0_36 21

François Guimbretière, Maureen Stone, and Terry Winograd. 2001. Fluid interaction with high-resolution wall-size displays. In *Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST '01)*. ACM, 21–30. DOI: http://dx.doi.org/10.1145/502348.502353 6

Faizan Haque, Mathieu Nancel, and Daniel Vogel. 2015. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 3653–3656. DOI: http://dx.doi.org/10.1145/2702123.2702133 18

Stéphane Huot, Olivier Chapuis, and Pierre Dragicevic. 2011. Torusdesktop: pointing via the backdoor is sometimes shorter. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '11)*. ACM, 829–838. DOI: http://dx.doi.org/10.1145/1978942.1979064 46

Robert Jacob. 1991. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems* 9, 2 (1991), 152–169. DOI: http://dx.doi.org/10.1145/123078.128728 30

Mikkel R. Jakobsen and Kasper Hornbæk. 2014. Up close and personal: Collaborative work on a high-resolution multitouch wall display. *ACM Transactions on Computer-Human Interaction* 21, 2, Article 11 (2014), 34 pages. DOI: http://dx.doi.org/10.1145/2576099 1, 2

Herbert D. Jellinek and Stuart K. Card. 1990. Powermice and user performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM, 213–220. DOI: http://dx.doi.org/10.1145/97243.97276 4, 7, 10, 11

Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. 2010. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface (GI '10)*. CIPS, 269–276. http://dl.acm.org/citation.cfm?id=1839214.1839261 3, 11, 44

Paul Kabbash and William A. S. Buxton. 1995. The "prince" technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '95)*. ACM & Addison-Wesley, 273–279. DOI: http://dx.doi.org/10.1145/223904.223939 12

Kenrick Kin, Tom Miller, Björn Bollensdorff, Tony DeRose, Björn Hartmann, and Maneesh Agrawala. 2011. Eden: a professional multitouch tool for constructing virtual organic environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '11)*. ACM, 1343–1352. DOI: http://dx.doi.org/10.1145/1978942.1979141 29

Kotaro Kitajima, Yoichi Sato, and Hideki Koike. 2001. Vision-based face tracking system for window interface: prototype application and empirical studies. In *Extended Abstracts on Human factors in computing systems (CHI EA '01)*. ACM, 359–360. DOI: http://dx.doi.org/10.1145/634067.634279 29, 30

Masatomo Kobayashi and Takeo Igarashi. 2008. Ninja cursors: using multiple cursors to assist target acquisition on large screens. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '08)*. ACM, 949–958. DOI: http://dx.doi.org/10.1145/1357054.1357201 46

David R. Kollerl, Mark R. Mine, and Scott E. Hudson. 1996. Head-tracked orbital viewing: an interaction technique for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)*. ACM, 81–82. DOI: http://dx.doi.org/10.1145/237091.237103 30

Werner A. König, Jens Gerken, Stefan Dierdorf, and Harald Reiterer. 2009. Adaptive pointing: implicit gain adaptation for absolute pointing devices. In *Extended abstracts on Human factors in computing systems (CHI EA '09)*. ACM, 4171–4176. DOI: http://dx.doi.org/10.1145/1520340.1520635 4, 15

Regis Kopper, Doug A. Bowman, Mara G. Silva, and Ryan P. McMahan. 2010. A human motor behavior model for distal pointing tasks. *International Journal of Human Computer Studies* 68, 10 (2010), 603–615. DOI: http://dx.doi.org/10.1016/j.ijhcs.2010.05.001 3, 11, 19, 44

Regis Kopper, Mara G. Silva, Ryan P. McMahan, and Doug A. Bowman. 2008. Increasing the precision of distant pointing for large high-resolution displays. In *Technical Report TR-08-17, Computer Science, Virginia Tech* (2008-01-01). 19

Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, Eric Lecolinet, and Wendy Mackay. 2014. Effects of display size and navigation type on a classification task. In *Proceedings of the 32nd international conference on Human factors in computing systems (CHI '14)*. ACM, 4147–4156. DOI: http://dx.doi.org/10.1145/2556288.2557020 1, 6

Scott MacKenzie and Shaidah Jusoh. 2001. An evaluation of two input devices for remote pointing. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction (EHCI '01)*. Springer, 235–250. DOI: http://dx.doi.org/10.1007/3-540-45348-2_21 3

Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. 2005. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05)*. ACM, 43–52. DOI: http://dx.doi.org/10.1145/1095034.1095042 6

Regan L. Mandryk and Calvin Lough. 2011. The effects of intended use on target acquisition. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '11)*. ACM, 1649–1652. DOI: http://dx.doi.org/10.1145/1978942.1979182 45

David C. McCallum and Pourang Irani. 2009. ARC-Pad: Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09)*. ACM, 153–156. DOI: http://dx.doi.org/10.1145/1622176.1622205 2, 5, 11, 12, 29

David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review* 95 (1988), 340–370. DOI: http://dx.doi.org/10.1037/0033-295X.95.3.340 11

L.-P. Morency and Trevor Darrell. 2006. Head gesture recognition in intelligent interfaces: the role of context in improving recognition. In *Proceedings of the 11th international conference on Intelligent user interfaces (IUI '06)*. ACM, 32–38. DOI: http://dx.doi.org/10.1145/1111449.1111464 30

Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. 2002. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '02)*. ACM, 33–40. DOI: http://dx.doi.org/10.1145/503376.503383 3

Sungwon Nam, Sachin Deshpande, Venkatram Vishwanath, Byungil Jeong, Luc Renambot, and Jason Leigh. 2010. Multi-application inter-tile synchronization on ultra-high-resolution display walls. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems (MMSys '10)*. ACM, 145–156. DOI: http://dx.doi.org/10.1145/1730836.1730854 1

Sungwon Nam, Byungil Jeong, Luc Renambot, Andrew Johnson, Kelly Gaither, and Jason Leigh. 2009. Remote visualization of large scale data for ultra-high resolution display environments. In *Proceedings of the 2009 Workshop on Ultrascale Visualization (UltraVis '09)*. ACM, 42–44. DOI: http://dx.doi.org/10.1145/1838544.1838550 1

Mathieu Nancel. 2012. *Designing and combining interaction techniques in large display environments*. Ph.D. Dissertation. École Doctorale d'Informatique de l'Université Paris-Sud, France. 3, 5, 30

Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. 2013. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 831–840. DOI: http://dx.doi.org/10.1145/2470654.2470773 2, 6, 12, 29

Mathieu Nancel, Emmanuel Pietriga, and Michel Beaudouin-Lafon. 2011. *Precision Pointing for Ultra-High-Resolution Wall Displays*. Technical Report RR-7624. INRIA. 8 pages. 2, 9, 11, 12

Mathieu Nancel, Daniel Vogel, and Edward Lank. 2015. Clutching is not (necessarily) the enemy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 4199–4202. DOI: http://dx.doi.org/10.1145/2702123.2702134 7, 47

Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '11)*. ACM, 177–186. DOI: http://dx.doi.org/10.1145/1978942.1978969 1, 5

Daniel Natapov, Steven J. Castellucci, and I. Scott MacKenzie. 2009. Iso 9241-9 evaluation of video game controllers. In *Proceedings of Graphics Interface (GI '09)*. CIPS, 223–230. http://dl.acm.org/citation.cfm?id=1555880.1555930 3, 4

Kai Nickel and Rainer Stiefelhagen. 2003. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on Multimodal interfaces (ICMI '03)*. ACM, 140–146. DOI: http://dx.doi.org/10.1145/958432.958460 4, 29

Ji-Young Oh and Wolfgang Stürzlinger. 2002. Laser pointers as collaborative pointing devices. In *Proceedings of Graphics Interface (GI '02)*. 141–150. 3

Dan R. Olsen, Jr. and Travis Nielsen. 2001. Laser pointer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '01)*. ACM, 17–22. DOI: http://dx.doi.org/10.1145/365024.365030 3

Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. 1997. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics (SI3D '97)*. ACM, 39–42. DOI: http://dx.doi.org/10.1145/253284.253303 3

Emmanuel Pietriga, Stéphane Huot, Mathieu Nancel, and Romain Primet. 2011. Rapid development of user interfaces on cluster-driven wall displays with jbricks. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. ACM, 185–190. DOI: http://dx.doi.org/10.1145/1996461.1996518 5, 21

Philip Quinn, Andy Cockburn, Kari-Jouko Räihä, and Jérôme Delamarche. 2011. On the costs of multiple trajectory pointing methods. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '11)*. ACM, 859–862. DOI: http://dx.doi.org/10.1145/1978942.1979067 45

Kari-Jouko Räihä and Oleg Špakov. 2009. Disambiguating ninja cursors with eye gaze. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)*. ACM, 1411–1414. DOI: http://dx.doi.org/10.1145/1518701.1518913 31

Nicolas Roussel, Géry Casiez, Jonathan Aceituno, and Daniel Vogel. 2012. Giving a hand to the eyes: leveraging input accuracy for subpixel interaction. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. ACM, 351–358. DOI: http://dx.doi.org/10.1145/2380116.2380162 15, 16

Joseph D. Rutledge and Ted Selker. 1990. Force-to-motion functions for pointing. In *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction (INTERACT '90)*. North-Holland, 701–706. http://dl.acm.org/citation.cfm?id=647402.725310 3, 15

Julia Schwarz, David Klionsky, Chris Harrison, Paul Dietz, and Andrew Wilson. 2012. Phone as a pixel: enabling ad-hoc, large-scale displays using mobile devices. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2235–2238. DOI: http://dx.doi.org/10.1145/2208276.2208378 5

Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow reaching: a new perspective on interaction for large displays. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*. ACM, 53–56. DOI: http://dx.doi.org/10.1145/1294211.1294221 4

Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, and Kellogg S. Booth. 2012. Two-part models capture the impact of gain on pointing performance. *ACM Transactions on Computer-Human Interaction* 19, 4, Article 28 (Dec. 2012), 34 pages. DOI: http://dx.doi.org/10.1145/2395131.2395135 13, 25

William Soukoreff and Scott MacKenzie. 2004. Towards a standard for pointing device evaluation: Perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human Computer Studies* 61, 6 (2004), 751–789. DOI: http://dx.doi.org/10.1016/j.ijhcs.2004.09.001 8

John S. Stahl. 1999. Amplitude of human head movements associated with horizontal saccades. *Experimental Brain Research* 126, 1 (1999), 41–54. DOI: http://dx.doi.org/10.1007/s002210050715 29

Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2981–2990. DOI: http://dx.doi.org/10.1145/2208636.2208709 30

Norbert A. Streitz, Jörg Geissler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. 1999. i-land: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '99)*. ACM, 120–127. DOI: http://dx.doi.org/10.1145/302979.303010 6

Desney S. Tan, Darren Gergle, Peter Scupelli, and Randy Pausch. 2006. Physically large displays improve performance on spatial tasks. *ACM Transactions on Computer-Human Interaction* 13, 1 (2006), 71–99. DOI: http://dx.doi.org/10.1145/1143518.1143521 6

Daniel Vogel and Ravin Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05)*. ACM, 33–42. DOI: http://dx.doi.org/10.1145/1095034.1095041 1, 2, 5, 11, 18

Julie Wagner, Stéphane Huot, and Wendy Mackay. 2012. BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2317–2326. DOI: http://dx.doi.org/10.1145/2208276.2208391 29

Julie Wagner, Mathieu Nancel, Sean G. Gustafson, Stephane Huot, and Wendy E. Mackay. 2013. Body-centric design space for multi-surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 1299–1308. DOI: http://dx.doi.org/10.1145/2470654.2466170 5

Colin Ware. 2004. *Information visualization: perception for design*. Morgan Kaufmann, San Francisco, CA, USA. 5

Beth Yost, Yonca Haciahmetoglu, and Chris North. 2007. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*. ACM, 101–110. DOI: http://dx.doi.org/10.1145/1240624.1240639 1, 6

Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '99)*. ACM, 246–253. DOI: http://dx.doi.org/10.1145/302979.303053 30

Received July 2012; revised March 2015; accepted April 2015