ForceExtension: Extending Isotonic Position-controlled Multi-touch Gestures with Rate-controlled Force Sensing for 3D Manipulation

Jia Wang[*]
HIVE Lab
Worcester Polytechnic Institute

Robert W. Lindeman[†]
HIVE Lab
Worcester Polytechnic Institute

ABSTRACT

Recent advances in multi-touch technology have enabled pressure sensing of each touch point on a multi-touch touchpad in addition to position tracking. In this paper we propose two novel approaches for utilizing this extra dimension of input to extend the effect range of position controlled multi-touch gestures. Both *ForceExtension* approaches are only activated when the averaged force of all active fingers reaches a threshold. The first approach, *context-force extension*, tracks the most recent position-control movement as the context and combines it with the force input as an isometric rate-controlled extension. The second approach, *shear-force extension*, scales the micro displacement of the active fingers with the force input to simulate shear-force sensing as a viscoelastic rate-controlled extension. We collected feedback from several users who were asked to perform a 3D search task using variations of these interfaces. A single force sensing multi-touch touchpad was used to control the first-person camera during the search, and the multi-touch gestures to pan, rotate, and zoom the 3D camera were augmented through ForceExtension. Users preferred a medium gain position control combined with the context-force extension.

Keywords: Force sensing touchpad, multi-touch, position control, rate control, isotonic and isometric, hybrid solution.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input Devices and Strategies.

1 INTRODUCTION

Current multi-touch screens and touchpads support rich expressiveness of multi-touch gestures such as swipe, pinch, and rotation. However, multi-touch input is isotonic, and the dominant mapping is for position control. Like the mouse, multi-touch interfaces face the same challenges of greater user fatigue and degraded user performance when frequent clutching – the temporary recalibration to extend the limited input space – is required. On the other hand, rate-control devices such as the IBM TrackPoint [12] eliminate clutching, but suffer from low precision when performing precise movements.

Based on the fact that isotonic position-control devices and isometric/elastic rate control devices afford similar performance [17], research efforts have recently been investigating hybrid approaches, tapping the advantages of both position and rate control by augmenting the normal touchpad with a rate-controlled elastic edge [5] [10]. Empirical studies of 2D pointing tasks have shown improved performance and user experience of such hybrid

solutions in comparison to position control alone. However, these interfaces have limitations. For example, to switch between position and rate control, the finger needs to constantly move back and forth between the center and the edge of the input space to activate different sensors. This discontinuous transition may break the interaction flow of the user and hurt performance even when complementary subtasks are assigned to both sensors [10]. Additionally, as these devices were originally designed to expand the effect range of position-controlled pointing, they are not capable of combining position and rate control for multi-touch gestures such as pinch, rotation, or multi-finger swipe.

Taking advantage of recent advances in multi-touch force sensing technology, we report two novel attempts to combine the precision of position control with the large effect range of rate control. The first approach, *context-force extension*, uses the most recent macro-finger displacement as the context of the extension, preserving the user's flow of interaction through a smooth transition between the two modes. By scaling micro-finger displacement with pressure input, the second approach, *shear-force extension*, successfully simulates shear force sensing, allowing the user to change the direction of rate control without switching back to position control. The possible pros and cons of the two approaches are discussed and an evaluation is reported in which a force sensing multi-touch touchpad alone was used for first-person 3D camera manipulation.

2 RELATED WORK

The two most popular ways to map input to output are position control (zero order) and rate control (first order) [17]. Most pointing devices such as the mouse use position control. Previous studies comparing a mouse and a finger-controlled isometric joystick (the IBM TrackPoint) revealed the movement microstructure of both devices and concluded that the random variations in the velocity of the joystick make it harder to control [12]. However, when the input space of position-control devices is limited compared to the much-larger screen space, frequent clutching can cause low efficiency and high fatigue. Increasing Control-Display gain (or CD gain) can reduce clutching, but a high CD gain can hurt performance [1]. Alternatively, the CD gain can be dynamically adjusted based on the velocity (*i.e.,* the pointer acceleration technique [9]) or the range (*e.g.*, the Go-go interaction technique [13]) of the input. However there is no published research showing the performance benefit of such techniques in comparison to standard position control with clutching [9]. Zhai classified input devices into isotonic, isometric, and elastic, and found that isotonic devices were better suited to position control and isometric and elastic devices should use rate control because of their self-centering properties [17].

Instead of clutching, the effect range of position control can be extended by rate control when position input reaches the limit of the input space, such as the Bubble technique [5], the RubberEdge [2], and the GroovePad [10]. Based on the findings of Zhai, these interfaces all use isotonic devices for position input and switch to elastic rate control at the edge of the input space. The Bubble

[*]e-mail: wangjia@wpi.edu
[†]e-mail: gogo@wpi.edu

technique simulates a spherical volume in physical space and visualizes it as a transparent sphere on the display. The movement of the input point is by position control within the volume and by rate control beyond the volume with elastic feedback. The RubberEdge technique identified a flaw in the mapping functions of the Bubble technique that created trajectory and velocity discontinuities at the transition point, and proposed a novel solution inspired by the physical movement of a dinner plate when pulled at the edge with a string. A user study was conducted for a 2D pointing task and results showed that RubberEdge outperformed position control by 20% when there was significant clutching [2]. The GroovePad [10] provides a hardware implementation of RubberEdge and researchers studied its usability in pointing, panning, and dragging tasks. Although results indicated using GroovePad reduced clutching compared to a standard touchpad, performance failed to increase as users spent extra time deciding which mode to activate when using two input sensors for the same functionality. Hybrid position/rate-control techniques have also been used for other purposes. For example, the Magic Barrier Tape uses rate-controlled navigation to extend the walk-able space in virtual reality [4].

Pressure sensors can be added to position-control devices to provide an extra degree of freedom (DOF) for input. Ramos & Balakrishnan studied the human ability to perform discrete target selection tasks by varying a stylus' continuous pressure, with full or partial visual feedback and different ways to confirm selection once the target is acquired [14]. One of the challenges of using pressure input on a touch surface, either through finger or stylus, is the potential interference between spatial x-y movement and pressure channels. In contrast, the movement of the mouse is much more stable and orthogonal to the control of pressure sensors attached to the side of the mouse, allowing users to comfortably control up to 64 modes with a dual-pressure augmented mouse [3]. Shear (tangential) force can also be sensed by pressure sensors attached to the four corners of a touch surface. When applying shear force, the finger does not perceptibly move but the skin of the finger pad shifts position slightly and provides the user with viscoelastic feedback [11]. The potential of shear-force input has been demonstrated for mobile multi-touch devices through single-touch force gestures [7] [8]. A recent user study investigated the user controllability of shear force to reach and maintain target force levels with regard to hand pose and direction of force input, and found that target-acquisition tasks using shear-force input follow Fitts' law [6] and that users have more physical and perceived loads when applying shear force in the lateral direction [11]. Micro thumb-rolls have been found to be discernable from swipes in the definition of a gesture set for mobile input [15]. However, to the best of our knowledge, there is no published solution that senses shear forces of multiple touch points, and the use of rate-controlled shear force to extend the effect range of position control is rarely discussed.
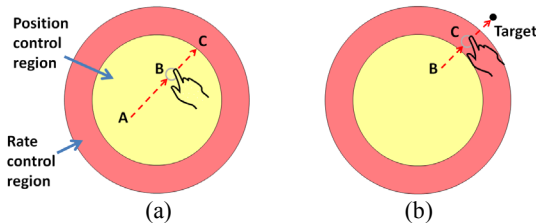

Figure 1: Limits of existing hybrid position/rate-control techniques: (a) can only trigger rate control at edges of position control; (b) need to clutch to use position-control for precise control.

3 METHODOLOGY

Existing hybrid position/rate-control techniques trigger a transition to rate control only at the edge of the position-control region. This design has two limitations. Firstly, as shown in Figure 1(a), since the rate-control region is on the edge of the input space, the user cannot smoothly transition to rate control immediately at point B when he/she realizes the target is out of reach. Instead, an unnecessary finger movement to point C is required. Secondly, as shown in Figure 1(b), assuming the user has performed enough rate-controlled movement in the direction of BC to approach the aforementioned target without overshoot, he/she still has to do at least one clutch to point exactly to the target in order to use the position-control input space.

3.1 The Synaptics ForcePad

Both limitations can be resolved using a force-sensing touchpad, as pressure-based rate control can be triggered at any time and place on the touchpad. Also, because rate control can start anywhere, the extra space around the finger can be used to finish the final touch when the target is approached. The force-sensing touchpad used in this work is the Synaptics ForcePad [15]. As shown in Figure 2, it is a multi-touch force-sensing touchpad that detects up to five fingers of variable pressure, at 6-bit resolution and up to 1000g of force. By sensing the vertical force of each active finger, the ForcePad adds an extra DOF input to the position input of traditional touchpads, which can be utilized to preserve the fluidity of multi-touch gestures.
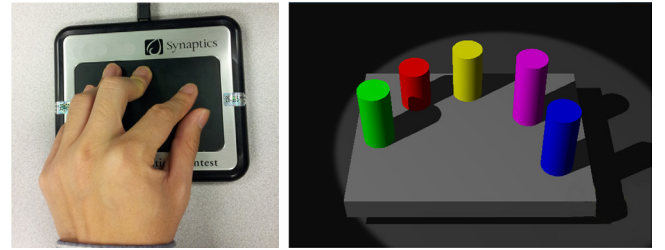

Figure 2: The Synaptics ForcePad detects up to five fingers of variable pressure (snapshot from Synaptics demo software)

3.2 Context-Force Extension

The transition between position and rate control is triggered by the force input crossing a threshold, which is set to 20 percent (200g) of the maximum force in our implementation. Taking single-finger cursor control as an example, Equation 1 and Figure 3 illustrate the transfer functions and the position/rate-control transitions of the context-force extension mode step by step. When a touch-down gesture is detected, the base point is updated to P_0, and all subsequent movement of the finger (the vector from the base point to the current point, $P_1 - P_0$) is scaled by a constant CD gain (c in Equation 1) to move the cursor by position control as long as the current force F_1 is below the threshold F_t. When F_1 goes beyond F_t, rate control is triggered and the speed vector is calculated by scaling the current position control vector (P_0P_1, as the *context*) with the force input beyond the threshold ($F_1 - F_t$). To prevent adding noise to the speed vector, the finger position tremors after the rate control activation (P_2) are ignored [14]. The transition back to position control is triggered when F_2 decreases to below F_t, while the base point gets updated to P_2 so that the direction of future rate-controlled movement can be changed without lifting the finger, allowing smooth transitions between position and rate control within one touch session.

$$\overrightarrow{\Delta P} = c * (P_i - P_0), \text{when } F_i < F_t \qquad \text{Equation 1}$$

$$\vec{v} = \overrightarrow{\Delta P} * (F_i - F_t), \text{when } F_i \geq F_t$$

Step 1:
- $F_1 < F_t$ (threshold)
- Position control
- $P_1 - P_0$ moves cursor on the display by constant CD gain

Step 2:
- $F_1 \geq F_t$ triggers mode switch
- Rate control
- Speed vector:
$$\vec{v} = \overrightarrow{P_0 P_1} * (F_1 - F_t)$$

Step 3:
- $F_2 \geq F_t$ keeps current mode
- Rate control
- P_2 is ignored to avoid finger tremor
- Speed vector:
$$\vec{v} = \overrightarrow{P_0 P_1} * (F_2 - F_t)$$

Step 4:
- $F_2 < F_t$ triggers mode switch again
- Position control
- Set base point to P_2

Step 5:
- $F_3 \geq F_t$ triggers mode switch again
- Rate control
- Speed vector:
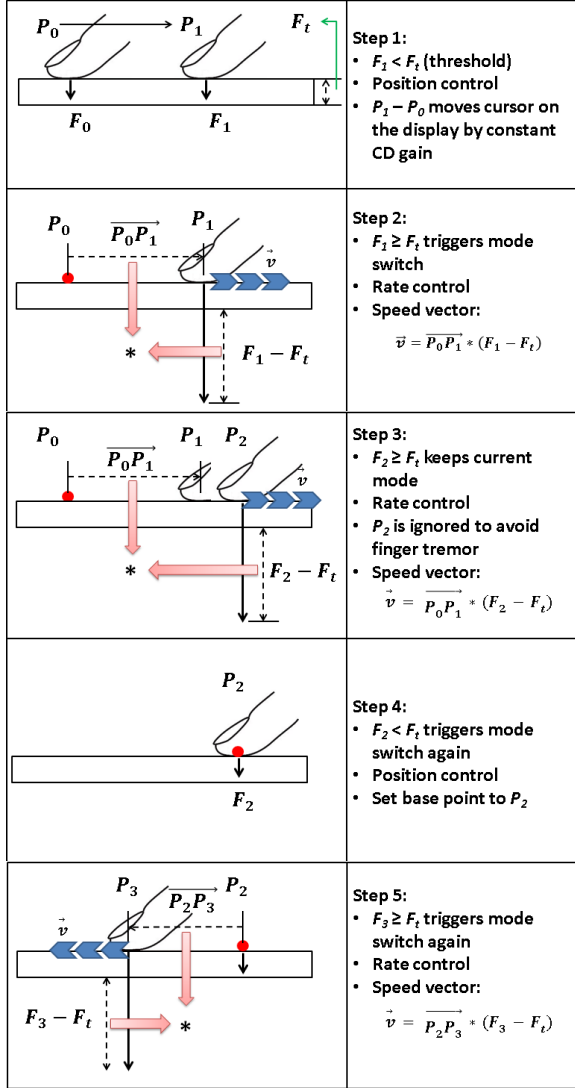$$\vec{v} = \overrightarrow{P_2 P_3} * (F_3 - F_t)$$

Figure 3: Step by step illustration of context-force extension

3.3 Shear-Force Extension

In Figure 3, the finger displacement during rate control is ignored at the third step to avoid noise from finger tremor. During the iterative implementation of the context-force extension mode, we discovered that when applying shear force on the touchpad, the detected position of the fingers shift slightly with the force variance. This "micro-finger displacement" provides a direction vector which can be combined with pressure input as the magnitude to simulate shear-force sensing. As shown in Figure 4, the base point is set to P_1 when F_1 exceeds F_t and instead of scaling the previous position-control movement (P_0P_1) as the context of rate control, the shear-force extension mode tracks the micro-finger displacement (P_1P_2) and scales it with the force input beyond the threshold ($F_2 - F_t$) and a constant factor c (to scale up the micro-finger displacement, this was set to 10.0 in our implementation) to calculate the speed vector. Preliminary tests confirmed that this mechanism was able to realistically simulate shear-force sensing of multiple active fingers in all directions.

Because the base point is updated every time F_i exceeds F_t, the mechanism is also very tolerant of different use patterns, as some users tend to release the force when changing the shear-force direction while others do not. Nevertheless, it should also be mentioned that a potential problem exists at the third step. If the finger movement from P_1 to P_2 is inadvertently more than a micro displacement, the transfer function could produce a velocity much greater than expected. The speed vector can be clamped at a maximum but per finger calibration is necessary to sample the possible range of the user's micro-finger displacement.

Step 2:
- $F_1 \geq F_t$ triggers mode switch
- Rate control
- Set base point to P_1
- No speed yet

Step 3:
- $F_2 \geq F_t$ keeps current mode
- Rate control
- Speed vector:
$$\vec{v} = c * \overrightarrow{P_1 P_2} * (F_2 - F_t)$$
- Scaled micro displacement by c to simulate shear force sensing

Step 4:
- $F_2 < F_t$ triggers mode switch again
- Position control
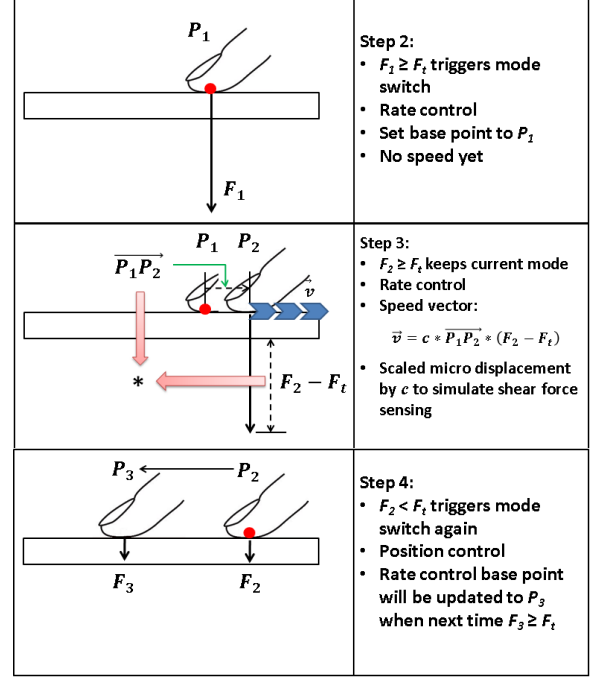- Rate control base point will be updated to P_3 when next time $F_3 \geq F_t$

Figure 4: Step by step illustration of shear-force extension

Both force-extension modes have also been applied to multi-touch gestures. For example, by averaging the pressure of multiple fingers and replacing single finger position (P) with two-finger centroid, separation, or rotation, multi-touch controls such as camera pan, pinch zoom, or camera orbit can be augmented using the same mechanism demonstrated in Figures 3 and 4.

4 USER EVALUATION

We conducted an informal user evaluation with six users to collect feedback using a single Synaptics ForcePad to perform 3D camera manipulation. A rough terrain was constructed with eight objects scattered around, each one with a three-digit number attached to it (Figure 5). The task was for the user to move to each of the objects and report the number verbally. Each user was instructed on the various interface variations, and was given time to explore and ask questions during the search process. No time limit was imposed, and the users were encouraged to think aloud as they searched. After each trial, users were asked for any comments, and after all the trials, they were asked to rank the interfaces. Five interface variations were constructed: three position-control-only interfaces with c (CD gain) in Equation 1 equal to 200, 400, or 800, (called "p200", "p400", and "p800" here), context-force extension with c=200 ("c200") and shear-force extension with c=200 ("s200").

Single-finger gestures were used to pan the camera, two-finger pinch gestures were used to zoom the camera in and out, and three-finger swipe gestures were used to orbit the camera around a point on the surface at the center of the view. All interface

variants used position control with their respective values for gain (*c*). In addition, c200 and s200 incorporated the proposed force extensions as described previously for panning, zooming, and orbiting the camera. The application was developed in the Unity3D game engine using TUIO to communicate to the Synaptics ForcePad driver.
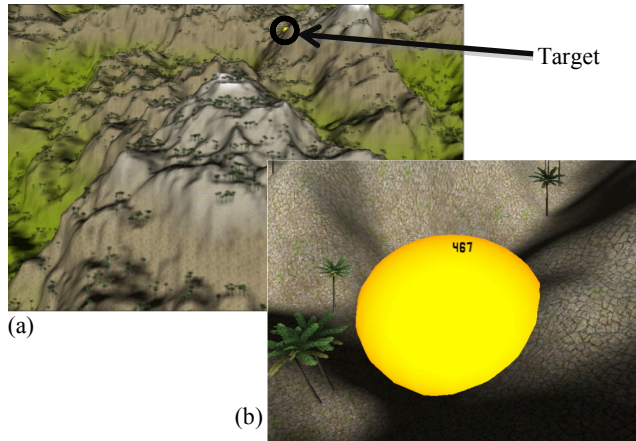

Figure 5: (a) Terrain scene and (b) target object close up

Of the position-control-only approaches, p400 was preferred most often, with p200 reported as requiring too much clutching, and p800 leading to too much overshoot. For all five interfaces, two users preferred c200, three preferred p400, and one preferred p800. Two users ranked c200 as the least preferable, but these users also commented that more practice might change their answers. Two users commented that they thought the rate-controlled movement was counter to what they expected, meaning that they expected the *scene* to move in the direction of finger movement instead of the *camera* to move in the direction of finger movement. This is the well-known cognitive problem of viewport vs. content scrolling that is present in many tablet interfaces. Providing a settable user preference for this is one solution used in many applications. Three users suggested the strategy of using force extension modes for large scale navigation, and low (200) to medium (400) gain position control when close to a target, and commented that force extension was most efficient when zooming out to the highest level. Finally, two users suggested that the c200 should have a higher gain (*c*) value, and that the force should be scaled using a fixed rate, rather than a rate based on the length of the recent position-control gesture. This might remove some of the confusion, and improve usability.

5 CONCLUSION AND FUTURE WORK

With the advance of multi-touch technology, next generation laptops will be equipped with multi-touch touchpads that can accurately sense the pressure of each individual finger. To take advantage of this new technology, we have presented two novel approaches to smoothly combine position and rate control for multi-touch gesture input. The transition to rate control is triggered whenever the averaged force input of all active fingers exceeds a threshold. The context-force extension approach extends the most recent position-control movement using rate control, and the shear-force extension approach utilizes the fingers' micro displacement after passing the threshold to simulate shear-force based rate control. By granting position control context to the rate-control extension, the former approach intensifies the user's recent memory of the current interaction state, promising not only physically but also cognitively smooth transitions between position and rate control. On the other hand, the latter approach builds on the intuitiveness and rich

expressiveness of shear-force input, allowing the direction of rate-control extension to be altered without switching back to position control. A carefully designed and formally conducted user study is necessary to draw comparative results of the two approaches and is regarded as future work. Furthermore, an analysis of existing hybrid position/rate-control techniques revealed two limitations that may break the user's flow of interaction, making our approach more advantageous. However, it has been argued that transitioning from horizontal movement to vertical pressure for rate control may not be intuitive [2], which poses a challenge to ForceExtension. Therefore it is also of our interest to formally compare the two ForceExtension approaches with existing techniques such as the GroovePad [10] to explore under what circumstances each technique is preferred.

REFERENCES

[1] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human-Computer Interaction*, vol. 23(3), pp. 215–250, 2008.

[2] G. Casiez, D. Vogel, Q. Pan, and C. Chaillou. RubberEdge: Reducing Clutching by Combining Position and Rate Control with Elastic Feedback. In *Proc. of ACM UIST'07*, pp. 129–138, 2007.

[3] J. Cechanowicz, P. Irani, and S. Subramanian. Augmenting the Mouse with Pressure Sensitive Input. In *Proc. of ACM SIGCHI'07*, pp. 1385–1394, 2007.

[4] G. Cirio, M. Marchal, T. Regia-Corte, and A. Lécuyer. The Magic Barrier Tape: a Novel Metaphor for Infinite Navigation in Virtual Worlds with a Restricted Walking Workspace. In *Proc. of ACM VRST'09*, pp. 155–162, 2009.

[5] L. Dominjon, A. Lécuyer, J. Burkhardt, G. Andrade-Barroso, and S. Richir. The "Bubble" Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace. In *Proc. of IEEE World Haptics*, pp. 639–640, 2005.

[6] P. Fitts. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *J. of Exp. Psychology*, vol. 47(6), pp. 381–391, 1954.

[7] C. Harrison and S. Hudson. Using Shear as a Supplemental Two-dimensional Input Channel for Rich Touchscreen Interaction. In *Proc. of ACM SIGCHI'12*, pp. 3149–3152, 2012.

[8] S. Heo and G. Lee. Force Gestures: Augmenting Touch Screen Gestures with Normal and Tangential Forces. In *Proc. of ACM UIST'11*, pp. 621–626, 2011.

[9] H. D. Jellinek and S. K. Card. Powermice and User Performance. In *Proc. of ACM SIGCHI'90*, pp. 213–220, 1990.

[10] A. Kulik, A. Kunert, A. Huckauf, and B. Froehlich. The Groovepad: Ergonomic Integration of Isotonic and Elastic Input for Efficient Control of Complementary Subtasks. In *Proc. of ACM APCHI'12*, pp. 75–84, 2012.

[11] B. Lee, H. Lee, S. Lim, H. Lee, S. Han, and J. Park. Evaluation of Human Tangential Force Input Performance. In *Proc. of ACM SIGCHI'12*, pp. 3121–3130, 2012.

[12] A. K. Mithal and S. A. Douglas. Differences in Movement Microstructure of the Mouse and the Finger-controlled Isometric Joystick. In *Proc. of ACM SIGCHI'96*, pp. 300–307, 1996.

[13] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *Proc. of ACM UIST'96*, pp. 79–80, 1996.

[14] G. Ramos, M. Boulos, and R. Balakrishnan. Pressure Widgets. In *Proc. of ACM SIGCHI'04*, pp. 487–494, 2004.

[15] A. Roudaut, E. Lecolinet, and Y. Guiard. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proc. of ACM SIGCHI'09,* pp. 927–936, 2009.

[16] Synaptics ForcePad: http://www.synaptics.com/solutions/products/forcepad

[17] S. Zhai. Human Performance in Six Degree of Freedom Input Control. *Ph.D. Thesis*, University of Toronto, 1995.