

# PressureMove: Pressure Input with Mouse Movement

Kang Shi<sup>1</sup>, Sriram Subramanian<sup>2</sup>, and Pourang Irani<sup>1</sup>

<sup>1</sup>Computer Science Department, University of Manitoba  
Winnipeg, R3T 2N2, Canada

{kangshi, irani}@cs.umanitoba.ca

<sup>2</sup>Computer Science Department, University of Bristol  
Bristol, BS8 1UB, UK  
sriram@cs.bris.ac.uk

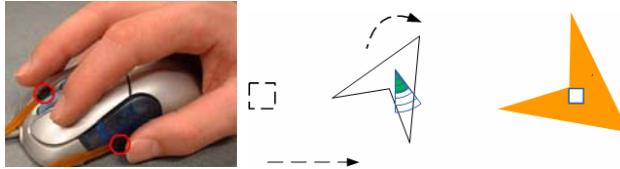
**Abstract.** We present *PressureMove* a pressure based interaction technique that enables simultaneous control of pressure input and mouse movement. Simultaneous control of pressure and mouse movement can support tasks that require control of multiple parameters, like rotation and translation of an object, or pan-and-zoom. We implemented four variations of PressureMove techniques for a 2D position and orientation matching task where pressure manipulations mapped to object orientation and mouse movement to object translation. The *Naive* technique mapped raw pressure-sensor values to the object rotation; the *Rate-based* technique mapped discrete pressure values to speed of rotation and *Hierarchical* and *Hybrid* techniques that use a two-step approach to control orientation using pressure. In user study that compared the four techniques with the default mouse-only technique we found that Rate-based PressureMove was the fastest technique with the least number of crossings and as preferred as the default mouse in terms of user-preference. We discuss the implications of our user study and present several design guidelines.

**Keywords:** Pressure-input, integrality of input dimensions, pressure and movement alternative interaction techniques.

## 1 Introduction

In line with recent incarnations of the mouse, Cechanowicz et al [2] have augmented the mouse with additional pressure input channels, and called this augmentation the PressureMouse. The PressureMouse builds upon the recently published set of guidelines for pressure based interaction [10,14,16]. However, recent studies on pressure interactions primarily provide insight on the strengths and limitations of pressure-based input and offer guidelines for creating pressure augmented interactions. Very little is known on how to fluidly integrate pressure input channels with the basic operations of the input device to which it is being augmented.

Pressure based interaction techniques proposed for the mouse are largely based on users manipulating the pressure channel independently of the movement degrees-of-freedom [10,14,16]. Pressure augmentation could potentially be designed such that the user can manipulate pressure input and cursor movement, enabling users to synchronously perform actions that can otherwise only be accomplished sequentially. For



**Fig. 1.** (a) Mouse with two pressure sensors; (b) Rotating an object with pressure input and displacing it using mouse movement to achieve a common task in several applications. (see <http://www.youtube.com/watch?v=YqyGaOSZhKY> for a video)

example, a mouse could potentially enable users to rotate and translate an object synchronously, a task that is routinely carried out in drawing applications (Figure 1).

Based on results of an early pilot study and prior work (Zliding[14] and Pressure-Marks[15]), we observed that users can simultaneously control pressure and movement, but not all users utilize the simultaneous control in a fluid fashion. In this paper we investigate the design space and the resulting interaction techniques that allow simultaneous control of pressure and movement, referred to as *PressureMove*. To demonstrate the effectiveness of PressureMove, we concentrated on the task of simultaneous rotation and object translation. We designed four PressureMove techniques that provide users the flexibility of using the input dimensions of pressure and movement simultaneously or sequentially. Pressure manipulations controlled object orientation and mouse movement controlled movement. In a 2D rotate and translate task, similar to the tetrahedral docking task in 3D [8,21], we examined the proposed designs for integrating mouse movement and pressure rotation. Our results show that one of our PressureMove designs, the rate-based integration offered best control and performance and was significantly faster than all other techniques including the traditional mouse.

The main contributions of this paper are to: 1) extend the design space of a pressure augmented device (the mouse) to include simultaneous control of pressure and movement; 2) design integral interaction techniques; 3) identify strengths of various strategies for controlling non-competing degrees-of-freedom; and 4) outline design implications that emerge from our systems.

## 2 Related Work

We review the related research on pressure input and integral input channels.

### 2.1 Pressure Based Interaction

Ramos et al. [16] explored the design space of pressure-based interaction with styluses. They proposed a set of pressure widgets that operate based on the users' ability to effectively control a discrete set of pressure values. Ramos et al. [16] identified that adequate control of pressure values is tightly coupled to a fixed number of discrete pressure levels (six maximum levels), the type of selection mechanism and a high degree of visual feedback. However, their investigation does not explore the benefits of simultaneously integrating pressure control with stylus movement.

Mizobuchi et al. [10] conducted a study to investigate how accurately people control pressure exerted on a pen-based device. Their results show that continuous visual feedback is better than discrete visual feedback, users can better control forces that are smaller than 3N, and 5 to 7 levels of pressure are appropriate for accurate discrimination and control of input values. Since controlling pressure input is challenging, Shi et al [17] recently proposed PressureFish, a technique to discretize the pressure space using fisheye functions. With PressureFish, users are capable of manipulating pressure input with a higher level of control and more efficiently than common discretization functions.

Researchers studied pressure input in the context of multi-level interaction. Zelezniak et al. [19] proposed an additional “pop-through” state to the mechanical operation of the mouse button. Forlines et al. [3] proposed an intermediary “glimpse” state to facilitate various editing tasks. Multi-level input can facilitate navigation, editing or selection tasks but utilize pressure input in a limited way. Such techniques make it challenging to fluidly control another input channel such as mouse movement.

Cechanowicz et al [2] investigated the possibility of facilitating pressure-based input by augmenting a mouse with either one or two pressure sensors. Such an augmentation allows users to control a large number of input modes with minimal displacements of the mouse. Cechanowicz et al [2] developed several pressure mode selection mechanisms and showed that with two pressure sensors users can control over 64 discrete pressure modes. However, Cechanowicz et al [2] did not investigate the possibility of fluidly integrating pressure input with other mouse based operations.

Few results suggest how we can fully integrate pressure with the underlying input mechanisms of the device to which it is augmented. Ramos et al [14] proposed Zliding to control a scaling factor with pressure at the stylus’ tip and manipulating a parameter with the stylus’ x-y position. Similarly, with PressureMarks [15] the user can invoke several states by steering the stylus and simultaneously applying various degrees of pressure. While both these studies highlight the possibility of integrating pressure input with the movement of the device, they have not explored the large design space that results when integrating both input channels.

In general, very few of the reported results have explored the design space of fluidly integrating pressure input with the functional features of the device. Furthermore, little is known about how pressure integrates with the very common task of moving a pointer. Based on this limited knowledge it is challenging to propose applications that can benefit from integrating pressure with multiple input channels.

## 2.2 Fluidly Controlling Multiple Input Channels

There has been a long standing interest in identifying how to integrate and facilitate control of simultaneous input channels. Jacob et al [13] proposed a framework that can facilitate the understanding and categorization of integrality and separability of input devices and interactions afforded by these. Two input dimensions are considered integral if they are perceived as a single dimension or separable if the dimensions seem unrelated [13]. In their study, performance was better when the device matched the tasks in integrality/separability dimensions. In light of their findings, coordinating multiple channels may suggest whether the input device is operating in the same dimension space as the task, i.e. good coordination and performance suggests that the

device and perceptual structure of the task are in the same space. Integrality can be considered to some extent as a coordination measure.

Balakrishnan et al [1] used integrality to demonstrate that subjects could control three degrees of freedom simultaneously with the Rockin'Mouse, a X-Y translational and one Z-rotational DOF. Similarly, MacKenzie et al. [7] investigated the possibility of integrating rotation on the mouse, a device designed primarily for translation and selecting objects. The TwoBall mouse facilitates a number of common tasks, and makes certain application features, such as the rotate tool, redundant.

Studies have also investigated the benefits and possibility of integrating several tasks into one coherent and fluid action. Kruger et al. [5] designed a technique, RNT (Rotate'N Translate), to fluidly integrate rotation and translation. The motivation behind RNT was to provide in one seamless action the ability to rotate and translate an object in a collaborative environment. Results of their study show that RNT is more efficient than separately controlling translation and rotation. RNT further enhances a number of collaborative tasks, including coordination and communication with respect to user orientation.

Fluid integration of multiple input channels was examined in the context controlling an input device with the fingers instead of using the entire arm. In an empirical study, Zhai et al [21] investigated the effectiveness of finger muscle groups in controlling multiple degrees-of-input. Zhai et al [21] gave users two alternative 6DOF input devices, one that controlled a cursor with the movement of the entire arm (glove) and the other with the fingers of a hand (FingerBall [21]). The objective of the study was to compare finger control to arm control in finely rotating and positioning an object in 3D. The task consisted of docking a cursor with the target, both of which were equal size tetrahedral. They found that the finger-based device afforded simultaneous translation and rotation actions with better control.

In developing a metric for measuring the allocation of control in a 6 degree-of-freedom rotation and translation task, Masliah and Milgram [8] studied the interdependence and overlapping actions of the two tasks. They used a 3D virtual docking task, similar to that of Zhai [20] in which subjects were asked to align a tetrahedral shaped cursor onto an identically shaped target. Interestingly, their results showed that users would rarely control all 6 DOFs simultaneously. Instead, users would allocate their control to the rotational and translational DOFs separately. Wang et al [18] carried out a study to investigate the relationship between object transportation and object orientation by the human hand. In their experiment, subjects were asked to align a small wooden block with a graphical target cube. Manipulation tasks were designed that required both object translation and orientation. Their results demonstrate the existence of a parallel and independent structure for object translation and orientation. Their results suggest that object translation and orientation share characteristics of an integral structure according to the notion by Jacob et al [13].

### 3 PressureMove

We propose PressureMove, a pressure based technique that facilitates simultaneous control of mouse movement and pressure input. We considered two dimensions: controlling pressure input, and visual feedback.

### 3.1 Controlling Pressure

Sensors typically report pressure values between 0 to 1024 levels. Previous studies have suggested that users are not capable of distinguishing the granularity and controlling this range of pressure values [2, 10, 16]. This has led most investigations to discretizing the pressure space into controllable and haptically perceivable units. Ramos et al. [14, 16] revealed that adequate control of pressure values is tightly coupled to a fixed number of discrete pressure levels (maximum of six levels). Cechanowicz et al. [2] suggested that pressure discretization can include 8 to 10 discrete levels, when controlled by the thumb or index finger, on a mouse.

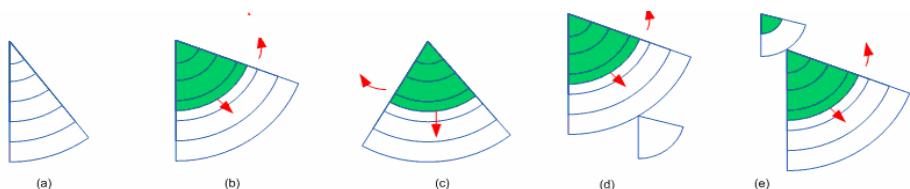
An alternative to discretizing pressure input is to map the raw pressure space (non-discretized referring to the fact that the discrete pressure values reported by the sensor are not further discretized) onto the task parameters. Each unit of pressure in the raw pressure space controls an input parameter, whether it be angular rotation, scalar, or other factor. Raw pressure input is not easily controlled, however facilitates a larger number of mappings.

We can also define a hybrid pressure space that uses continuous and discrete pressure values. With hybrid control, continuous pressure input provides the user with rapid access to a region of interest within the pressure space while switching to discrete control allows finer granularity and control over parameter values. PressureMove includes discrete, raw, and hybrid pressure control techniques.

### 3.2 Visual Feedback

Kinesthetic feedback alone is insufficient for adequately controlling pressure. Visual feedback is a dominant characteristic of most closed-loop pressure based interactions [2, 10, 14, 16]. Different forms of Visual feedback for pressure based input have been explored in PressureWidgets [16]. However, the Visual feedback in PressureMove is inspired by the visual feedback mechanism used by Kittenakare et al [4] and Ramos et al [16]. Since the design of the visual feedback is intricately tied to the task, we describe the feedback designed for the task of simultaneously positioning and orienting an object. We expect that a similar form of visual feedback can be easily adapted for other simultaneous control tasks.

A pressure cursor is used to provide appropriate visual feedback. The default cursor is a solid triangular shaped object (see Figure 2(a)). When the user applies pressure a proportion of this cursor gets highlighted relative to the amount of pressure



**Fig. 2.** Cursor state in the PressureMove techniques; a) standard PressureMove cursor without pressure; b) cursor fills up when pressure is being applied, movement is clockwise for one sensor and counter-clockwise for the other; c) in a hierarchical manner, for first pressure level, The red arrows are not part of the cursor and only used to explain how the cursor moves

being applied as in Figure 2(b) and 2(c). Visual feedback is always continuous, as this form of feedback has shown to enhance performance over non-continuous visual feedback. Additionally, we redundantly encode pressure amount to the aperture of the pressure cursor, i.e. the higher the pressure value, the larger the aperture of the cursor (as is seen in the difference in size of the cursor in Figure 2(a) and 2(b)).

In the case where we used a hybrid pressure space we used a two-step cursor as shown in Figure 2(d) and 2(e). The head-triangle (the triangle that represents the head of the cursor) represents the first pressure space the user can use while the second triangle corresponds to the second pressure space. In Figure 2(d) the user is currently controlling the first pressure space while in Figure 2(e) the user is operating with the second pressure space. In cases where multiple pressure spaces are composed to form the technique, multiple triangles can be concatenated. However, in our design we only used up to two pressure spaces composed to form a single technique.

## 4 PressureMove Techniques

We describe four variations of PressureMove techniques to manipulate mouse movement and pressure input simultaneously. All pressure interaction techniques used the thumb sensor to manipulate the parameter in one direction and the middle finger sensor to manipulate the parameter in the reverse direction.

### 4.1 PressureMove – Naïve

As the name suggests this is a naïve implementation of simultaneous control. In this technique the raw pressure values reported by the pressure sensor are mapped to the object parameter controlled by pressure. Figure 3(a) shows the mapping function - the pressure range is mapped to the complete range of the rotation parameter, i.e. 360° angle. When the user increases pressure the object orientation increases and when they release pressure the orientation reverses i.e., if the initial direction of rotation is clockwise then on releasing pressure the object changes orientation in the counter-clockwise direction. When the user releases the pressure sensor the parameter value returns to the starting position. To fix the value the user can left-click before releasing pressure. When the user presses the thumb sensor the object rotates clockwise and the visual feedback is as shown in Figure 2(b). When the user switches to the sensor located on the middle finger the object rotates counter-clockwise.

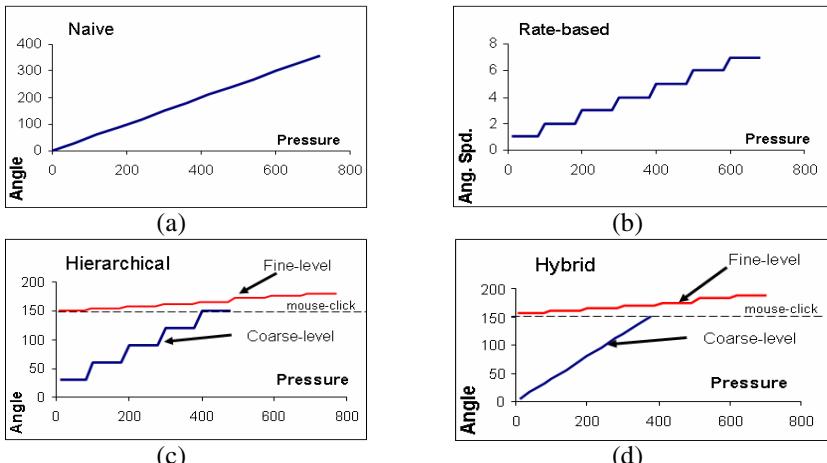
### 4.2 PressureMove – Rate-Based

In this technique each level of the discrete pressure space maps to the speed of rotation of the object as shown in Figure 3(b). When the user maintains pressure at discrete level 1 the object rotates by 1° at each timer event. To move the object faster the user moves higher up within the pressure levels. At level  $n$  the object rotates at  $n$  degrees per timer event. This mechanism provides the additional benefit of maintaining a given orientation when the user releases the pressure sensor, thus incorporating

a clutching mechanism that is not available with the naïve technique. At discrete level 0 the user can tap the pressure sensor to nudge the object by  $1^\circ$  per tap. This gives the user additional fine control when honing in on the target. This tapping was inspired from the Tap-and-Refine technique in [2]. The visual feedback used was the same as for the Naïve implementation.

### 4.3 PressureMove – Hierarchical

PressureMove-Hierarchical allows users to control rotation in two steps – a coarse-step and a fine-step. The coarse and fine movement is controlled by a discrete pressure mapping. In the coarse-step moving to a pressure level 1 results in rotating the object by  $24^\circ$  (one step is  $360^\circ/15\text{levels} = 24^\circ$ ) and moving up successive levels rotated the object by  $24^\circ$  per level  $n$  ( $n \in [0,15]$ ,  $n$  is the coarse-step pressure level). Thus at any pressure level the object is rotated by  $n \times 24^\circ$ ; while in the fine-step moving up each pressure level rotates the object by  $1^\circ$  starting from  $n$ . The object rotates from  $n$  to  $n \times 24 - 15$  using one sensor and from  $n$  up to  $n \times 24 + 15$  using the other sensor where  $n$  is the point in the coarse-control when the user switches to fine-control. The user can toggle between coarse- and fine-step by using the left click button. Figure 3(c) shows the pressure vs angle profile for this technique. The dotted line at about  $150^\circ$  indicates the moment at which the user moved from coarse to fine control using left-click. Figure 2(d) and 2(e) show the visual feedback that was provided to the user when using the thumb sensor (so object rotates clockwise). The top triangle of the cursor changes with pressure when the user is performing a coarse-level action (as in Figure 2(d)) and the bottom triangle changes with pressure when the user is performing a fine-level action (as in Figure 2(e)).



**Fig. 3.** The pressure mapping functions for each of the PressureMove techniques. a) Naïve implementation b) Rate-based technique c) Hierarchical technique; d) Hybrid technique. The dotted horizontal line in (c) and (d) at Angle = 150 indicates a left-click action. The red line is the fine-level control and the blue line is for the coarse-level.

#### 4.4 PressureMove - Hybrid

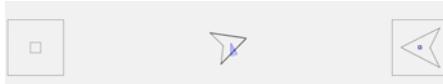
Hybrid combines the simplicity available with Naïve with the fine control provided by Hierarchical. The coarse-step of Hierarchical is replaced by the continuous rotation control used in Naïve (see the bottom left part of Figure 3(d)). This enables the user to quickly rotate the object to approximately the desired orientation and then use finer step control to perform a more precise orientation. The fine-control step and the visual feedback mechanism worked exactly as in Hierarchical.

### 5 Experiment

The goal of this experiment was to evaluate PressureMove as a viable concept for simultaneous control of pressure input and mouse movement.

#### 5.1 Task and Stimuli

The task, shown in Figure 4, required the user to reposition and reorient to a target location and orientation a small object (100×100 pixels) which initially appeared upright and in the left end of the screen. The target, of a slightly larger size than the object appeared to the right of the object. The size, the distance to the object and the orientation of the target were changed as part of the experimental design.



**Fig. 4.** The experimental task consisted of docking a triangular shaped object over a target. Rotation is controlled using pressure, and displacement controlled with mouse movement.

Users see the object and the target before the beginning of each trial. The trial begins when the user moved the cursor onto the object and pressed the left mouse-click. They reposition and reorient the object to the target location using the different interaction techniques. When the object position and orientation match the target position and orientation, the target bounding rectangle changes to a green color. The user then has to maintain the matching position and orientation for 1 second before the trial is completed. We did this to prevent users from accidentally matching the position and orientation. If the user moves the object away from the matched position, the 1 second timer is reset. The object position and orientation were considered to match those of the target if the difference in pixels and orientation was within the target-fit parameter controlled as factor of the experiment. When the trial is completed the target bounding rectangle briefly turns red and the next trial loads.

#### 5.2 Hardware Configuration and Techniques

Our study used an optical mouse with pressure sensors mounted on its rim (Figure 1). The sensors (model #IESF-R-5L from CUI Inc.) could measure a maximum pressure value of 1.5Ns. Each sensor provided 1024 pressure levels. The application was

developed in C# and the sensor was controlled using the Phidgets library [11]. The experiments were conducted in full-screen mode at 1280×800 pixels on an Intel T5600 1.83GHz, Windows Vista OS. Two sensors were mounted on the mouse such that they could be easily accessed by the thumb or the middle finger (as shown in Figure 1). All pressure interaction techniques used the thumb sensor to rotate the object clockwise and the middle finger sensor to rotate the object counter clockwise.

For all the discrete pressure based techniques we used the PressureFish discretization function [17] with 15 pressure levels. For the continuous pressure cases we only used pressure values between 0 and 720 as previous research has shown that users find it difficult to maintain pressures at higher values.

### 5.3 Procedure and Design

The study used a  $5 \times 2 \times 3 \times 2$  within-participants factorial design. The factors were: Technique (Naive, Rate-based, Hierarchical, Hybrid, Mouse-only), Distance (500 pixels, 1100 pixels), Orientation (60, 135, 270) and Target Fit (tight, loose). A tight target-fit meant that the users had to position the center of the object within  $\pm 4$  pixels of the target center and the object orientation has to be within  $\pm 5^\circ$  of the target orientation. For loose target-fit these figures were  $\pm 12$  pixels and  $\pm 8^\circ$  respectively.

The order of presentation first controlled for technique and then for distance followed by orientation and target-fit. We explained the techniques and participants were given ample time to practice the techniques at the beginning of the experiment. The experiment consisted of three blocks with each block comprising of two repetitions for each condition. With 5 techniques, 2 distances, 3 orientations, 2 target-fits, 3 blocks, and 2 trials, the system recorded a total of  $(5 \times 2 \times 3 \times 2 \times 3 \times 2)$  360 trials per participant. The experiment took approximately 60 minutes per participant.

### 5.4 Performance Measure and Participants

The experimental software recorded trial completion time, and number of crossings as dependent variables. Trial completion time (MT) is defined as the total time taken for the user to position and orient the object within the target. The number of crossings (NC) is defined as the number of times the object enters and leaves the target position or orientation for a particular trial. Users were not able to proceed to the next trial without successfully completing the task and so there were no errors for the software to record. Participants were also asked in an exit questionnaire to rank the different pressure control techniques in terms of mental demand, physical demand, effort, overall performance and frustration. Thirteen participants (11 males and 2 females) between the ages of 19 and 40 were recruited from a local university. All participants had previous experience with graphical interfaces and used the mouse in their right hand. None of the participants had worked with a pressure based input device before.

### 5.5 Results

We used the univariate ANOVA test with participant number as a random factor and Tamhane post-hoc pair-wise tests (unequal variances) for all our analyses.

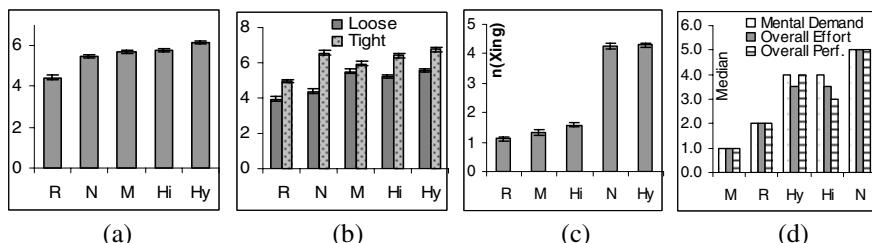
**Completion Time.** The average trial completion time was 6.1s (standard deviation 4.9s). Out of a total of 4680 trials 73 outliers (more than 3.5 standard deviations from the group mean) were excluded from further analysis. There was a significant effect of interaction technique ( $F_{(4,48)} = 11.15$ ,  $p < 0.001$ ), target-fit ( $F_{(1,12)} = 102.9$ ,  $p < 0.001$ ), distance ( $F_{(1,12)} = 7.5$ ,  $p < 0.02$ ), Orientation ( $F_{(2,24)} = 15.9$ ,  $p < 0.001$ ) and block-number ( $F_{(2,24)} = 43.4$ ,  $p < 0.001$ ) on MT. Figure 5 shows the mean trial completion time for each technique and target-fit. Overall, Rate-based was the fastest technique followed by Naive, Mouse, Hierarchical, and Hybrid.

Post-hoc analysis showed that all pairs were significantly different except (Naive, Mouse), and (Mouse, Hierarchical). Block 3 was significantly faster than Block 2 which was significantly faster than Block 1. Users were significantly slower in completing the trials when the target-fit was tight (as opposed to loose); when targets were farther (1100 pixels followed by 500) and when the orientation of target was greater (all combinations significantly different with  $270^\circ > 135^\circ > 60^\circ$ ).

**Crossings.** The average number of crossings per trial across all conditions was 2.5 (standard error = 0.048). There was a significant effect of interaction technique ( $F_{(4,48)} = 55.15$ ,  $p < 0.001$ ), target-fit ( $F_{(1,12)} = 68.1$ ,  $p < 0.001$ ), distance ( $F_{(1,12)} = 7.5$ ,  $p < 0.02$ ), Orientation ( $F_{(2,24)} = 19.8$ ,  $p < 0.001$ ) and block-number ( $F_{(2,24)} = 13.7$ ,  $p < 0.001$ ) on MT. We found no effect of target distance on number of crossings. Figure 5(c) shows the mean crossings for each technique. Overall Rate-based had the least number of crossings, followed by Mouse, Hierarchical, Naive and Hybrid.

Post-hoc comparisons showed that there was a significant difference between all pairs except the (Rate-based, Mouse), (Mouse, Hierarchical) and (Naive, Hybrid). Block 3 had significantly fewer crossings (mean 2.1) than Block2 (mean 2.4) which in turn had significantly fewer crossings than Block1 (mean 2.9). Users had significantly fewer crossings in the loose target-fit condition (mean 2.1) than in the tight target-fit condition (mean 2.9). Users had significantly fewer crossings when the target orientation was  $270^\circ$  (mean = 2) when compared to  $60^\circ$  (mean = 2.7) or  $135^\circ$  (mean = 2.9). We found no statistical difference in number of crossings between  $60^\circ$  and  $135^\circ$ .

**Subjective Ranking.** Users ranked the Mouse as the best technique followed by Rate-based, Hybrid, Hierarchical and Naive. Anova test on the overall performance revealed a significant difference in terms of user ranking between the different



**Fig. 5.** Mean trial completion time (along the Y-axis in seconds) with standard error bars (a) for each technique and (b) for each technique and target-fit. (c) Mean scores for different techniques. (d) Median user-ranking of different techniques in terms of Mental Demand, Overall Effort and Performance.

techniques ( $F(4,64) = 16.6$ ,  $p<0.001$ ). Post-hoc analysis did not reveal any significant differences between (Mouse, Rate-based) and (Hierarchical, Hybrid) pairs. But all other pairs were significantly different. We found similar rank-ordering of the techniques in terms of Overall Effort, Mental Demand, Physical Demand, and Frustration (see Figure 5(d)).

## 6 Discussion

Users were constantly improving their performance over the three blocks for both trial completion times and number of crossings. The average MT in Block 3 was 5.0s compared to 5.3s for Block 2 and 6.1s for Block 1. However, the univariate analysis we used in the previous section did not reveal any significant interaction between technique and block number for both MT and Crossings. Thus while users improved their performance over each block the overall order of the different techniques did not change. Observing improvement over blocks is in line with prior work suggesting that practice allows users to allocate better control to the simultaneous operation of different input dimensions [8].

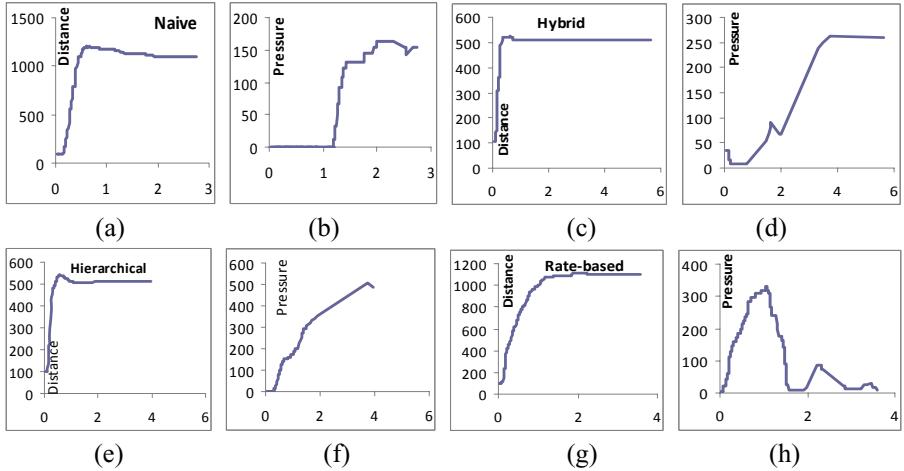
### 6.1 Simultaneous Control

As part of the experimental log we collected continuous data of mouse movement and pressure values for each trial. Figure 6 shows typical movement and pressure profiles for the four pressure-based techniques. Each left-right pair is distance and pressure profile for the same trial of a user. However, each technique is from a different user, selected randomly to highlight that the movement profiles shown in the figures are stereotypical. In the left images when the distance is not changing the user has positioned the object near the target whereas the same interpretation is not true for all pressure based techniques. The Rate-based technique being a relative input technique, users don't need to maintain constant pressure to complete the task.

We can see from the figure (Figure 6(a) and 6(b)) that the Naïve implementation does not really encourage simultaneous control of pressure and movement. Users use the first second to complete positioning the object before applying pressure to change orientation. We observe a similar trend with the Hybrid technique.

In the case of the Hierarchical technique, users start applying pressure about the same time that they start moving (see Figure 6(e) and 6(f)). But in the first part of their motion (between 0 and 2s) they mostly focus on moving the object to the right location and then switch attention to orienting (between 2 and 4s) the object.

But in the case of Rate-based technique, users start applying pressure to change orientation at the same time as they are moving the object to position it. In Figures 6(g) and 6(h) we see that in the first 2s the user is both positioning the object while at the same time as applying pressure. However, unlike the Hierarchical technique, they have completed most of the positioning and orienting within the first 2 s and between 2 and 4s they are merely fine-tuning the object. We believe that the open-loop motion for both positioning and orienting coincide making rate-based



**Fig. 6.** Traces of a typical user control when using the four PressureMove techniques. The patterns reveal the degree of simultaneity employed in each of the techniques, ranging from low simultaneity with the Naïve technique to high simultaneity with Rate-based.

technique a powerful PressureMove technique. However, we did not test this hypothesis with our data.

We observed similar profiles across all users and believe that PressureMove technique that's based on a rate-based mapping encourages users to simultaneously control both movement and pressure.

## 6.2 PressureMove – Rated-Based

PressureMove – Rate-based outperforms all other PressureMove designs; several unique properties of the technique contribute to its superior performance. This technique is based on discrete pressure control. A high degree of control is required to hold and maintain the pressure at given discrete levels; this is facilitated by a small number of discrete pressure levels and the use of discrete fisheye function.

Additionally, since each pressure level is assigned an angular velocity, pressure level 0 brings the rotating object to a halt, at the last applied orientation. The implicit clutching mechanism in the rate-based technique allows smaller close-loop movements than the other techniques. Finally, the technique allows fine adjustments at level 0, by nudging the object by  $1^\circ$  every tap. The fine grain control over angular displacement and the fluidity of this technique facilitates a higher degree of simultaneous control than any of the other systems.

Results similar to ours which show that rate-based technique improves performance in certain types of input devices, have been observed in 3D positioning tasks. Zhai [20] points out that using isometric devices (such as a joystick that self-centers) to operate in a position control mode (or zero-order) results in poorer performance than when operating the device in a rate-based mode (or a first-order).

### 6.3 Applications

PressureMove can enhance the interactive performance in a number of different applications. In all of the following applications, the simultaneous control of more than one input parameter would ease the task of the operator.

**Zoomable User Interfaces.** Zoomable user interfaces can largely benefit from the simultaneous control of several parameters. PressureMove can control various parameters by applying pressure to a scalar value and movement to direction of the zooming operation. For instance moving the mouse left or right could zoom in or out respectively, while pressure controls the resolution factor of the zooming operation. The rate-based technique would change the resolution of the zoom operation by one step at each level of angular velocity. Similarly, on a map the mouse movement would pan the document while pressure input zooms in or out. Unlike, most ZUI implementations where the center point of reference is defined by the position of the cursor or cross-hair before transitioning into the zoom, with PressureMove, the position of the cursor can be updated dynamically during zoom transitions, thereby facilitating a larger degree of freedom in moving around a workspace while zooming.

**Drawing Applications.** Drawing applications facilitate a large number of object positioning tasks with operations that involve rotating elements, scaling and/or skewing. Here, operations requiring coarse movements (such as scaling or skewing an object) could be relegated to the pressure input and precise positioning could be assigned to the mouse movement.

**Dynamic Control-Gain.** PressureMove could be utilized to dynamically manipulate control-gain ratios. Such manipulation is particularly useful on high resolution, large display interactions on which users operate with fine and coarse resolution.

### 6.4 Design Recommendations

There are several lessons that designers can take from our investigation:

- Pressure input can be appropriately integrated with mouse movement, such that both dimensions are operated simultaneously. This should result in higher performance gains than operating with either channel separately.
- PressureMove – Rate-based should be the first and preferred implementation of any PressureMove application. The discrete pressure control, fine grain pressure mapping and inherent clutching mechanisms in this technique are favorable properties that could be borrowed to implement other variations.
- Allowing users to gain experience with PressureMove is important and may be necessary in some cases, i.e. new implementation of a PressureMove technique should not be discarded without first giving consideration to proper training.

## 7 Conclusion

PressureMove is a novel technique that facilitates the simultaneous control of various input parameters. We designed PressureMove to specifically facilitate object rotation with pressure input and object movement with the mouse displacement. We designed

and implemented four PressureMove techniques, based on existing pressure-based interactions [2, 14]. Our PressureMove techniques cover the wide spectrum of possibilities with pressure control and mouse displacement mappings. In a study, the Rate-based PressureMove technique, which maps pressure input to angular velocity allowed the maximum amount of simultaneous control of pressure with mouse movement. Users were able to perform a docking task more efficiently and with fewer crossings with the rate-based implementation. We have demonstrated the possibility of simultaneous control of pressure input and mouse movement. We believe other similar interactions involving simultaneous pressure and movement are possible and will enhance the interactive performance on tasks with multiple input dimensions.

We are considering several lines of investigation for future work. We want to apply PressureMove to a number of other applications and tasks. For example, we believe new PressureMove techniques can be developed for navigation interfaces where the user is controlling several parameters simultaneously. We will also investigate the possibility of integrating PressureMove on other devices such as the stylus. While TabletPC interactions already provide pressure as a form of input, the pressure applied can interfere with movement. Therefore additional investigation is required to evaluate the possibility of PressureMove on the stylus.

## References

1. Balakrishnan, R., Baudel, T., Kurtenbach, G., Fitzmaurice, G.: The Rockin'Mouse: integral 3D manipulation on a plane. In: Proc. CHI 1997, pp. 311–318 (1997)
2. Cechanowicz, J., Irani, P., Subramanian, S.: Augmenting the mouse with pressure sensitive input. In: Proc. CHI 2007, pp. 1385–1394 (2007)
3. Forlines, C., Shen, C., Buxton, B.: Glimpse: a novel input model for multi-level devices. In: Proc. CHI 2005 Ext. Abstracts, pp. 1375–1378 (2005)
4. Kattinakere, R.S., Grossman, T., Subramanian, S.: Modeling steering within above-the-surface interaction layers. In: Proc. CHI 2007, pp. 317–326 (2007)
5. Kruger, R., Carpendale, S., Scott, S.D., Tang, A.: Fluid integration of rotation and translation. In: Proc. CHI 2005, pp. 601–610 (2005)
6. Kurtenbach, G., Buxton, W.: User learning and performance with marking menus. In: Proc. CHI 1994, pp. 258–264 (1994)
7. MacKenzie, I.S., Soukoreff, R.W., Pal, C.: A two-ball mouse affords three degrees of freedom. In: Proc. CHI 1997 Ext. Abstracts, pp. 303–304 (1997)
8. Masliah, M.R., Milgram, P.: Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In: Proc. CHI 2000, pp. 25–32 (2000)
9. MightyMouse<sup>TM</sup>, <http://www.apple.com/mightymouse/>
10. Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryyynanen, M., Silfverberg, M.: Making an impression: force-controlled pen input for handheld devices. In: Proc. CHI 2005 Ext. Abstracts, pp. 1661–1664 (2005)
11. Greenberg, S., Fitchett, C.: Phidgets: easy development of physical interfaces through physical widgets. In: Proc. UIST, pp. 209–218 (2001)
12. Jacob, R.J., Sibert, L.E.: The perceptual structure of multidimensional input device selection. In: Proc. CHI 1992, pp. 211–218 (1992)
13. Jacob, R.J., Sibert, L.E., McFarlane, D.C., Mullen, M.P.: Integrality and separability of input devices. ACM Trans. Comput.-Hum. Interact. 1(1), 3–26 (1994)

14. Ramos, G., Balakrishnan, R.: Zliding: fluid zooming and sliding for high precision parameter manipulation. In: Proc. UIST 2005, pp. 143–152 (2005)
15. Ramos, G.A., Balakrishnan, R.: Pressure marks. In: Proc. CHI 2007, pp. 1375–1384 (2007)
16. Ramos, G., Boulos, M., Balakrishnan, R.: Pressure widgets. In: Proc. CHI 2004, pp. 487–494 (2004)
17. Shi, K., Irani, P., Gustafson, S., Subramanian, S.: PressureFish: A Method to Improve Control of Discrete Pressure-based Input. In: Proc. CHI 2008, pp. 1295–1298 (2008)
18. Wang, Y., MacKenzie, C.L., Summers, V.A., Booth, K.S.: The structure of object transportation and orientation in human-computer interaction. In: Proc. CHI 1998, pp. 312–319 (1998)
19. Zeleznik, R., Miller, T., Forsberg, A.: Pop through mouse button interactions. In: Proc. UIST 2001, pp. 195–196 (2001)
20. Zhai, S.: User Performance in Relation to 3D Input Device Design. Computer Graphics 32(4), 50–54 (1998)
21. Zhai, S., Milgram, P., Buxton, W.: The influence of muscle groups on performance of multiple degree-of-freedom input. In: Proc. CHI 1996, pp. 308–315 (1996)