Rolling-Menu: Rapid Command Selection in Toolbars Using Roll Gestures with a Multi-DoF Mouse

Emmanuel Dubois, Marcos Serrano, Mathieu Raynal
University of Toulouse - IRIT
Toulouse, France
{first_name.last_name}@irit.fr

ABSTRACT
This paper presents Rolling-Menu, a technique for selecting toolbar items, based on the use of roll gestures with a multidimensional device, the Roly-Poly Mouse (RPM). Rolling-Menu reduces object-command transition, resulting in a better integration between command selection and direct manipulation of application objects. Selecting a toolbar item with Rolling-Menu requires rolling RPM in a predefined direction corresponding to the item. We propose a design space of Rolling-Menu that includes different roll mapping and validation modes. A first user's study, with a simple toolbar containing up to 14 items, establishes that the best version of Rolling-Menu takes, on average, up to 29% less time than the Mouse to select a toolbar item. Moreover accuracy of the selection with Rolling-Menu is above 90%. Both the validation mode and the mapping between roll direction and toolbar items influence the performance of Rolling-Menus. A second study compares the three best versions of Rolling-Menu with the Mouse to select an item in two types of multidimensional toolbars: a toolbar containing dropdown lists, and a grid toolbar. Results confirm the advantage of Rolling-Menu over a Mouse.

Author Keywords
Toolbar; Command selection; Multidimensional device; Roly-Poly Mouse; hemispherical mouse.

ACM Classification Keywords
H.5.2. Information interfaces and presentation: Interaction

INTRODUCTION
Most desktop applications include some type of toolbars. The simplest version of toolbar is an array of icons, usually arranged in a horizontal way on top of the application window. More advanced forms of toolbars can arrange multiple items in a grid that structures a set of items into different subsets (like the Microsoft Ribbon); they can also include dropdown lists (such as the list to select a font size in Word). The usual interaction with toolbars is carried with the mouse through a classical pointing gesture. However, it requires the user to move the mouse from the object of interest (text in a text editor, drawing area in a graphic design application) to the toolbar, and then come back once the command is selected to pursue the main task. This object-command transition breaks the interaction flow [3].

To reduce such disruption, multiple solutions have been proposed in the literature. Keyboard shortcuts [8] enable immediate access to commands but need to be memorized, and therefore are usually applied only to a few frequent commands [9]. Contextual menus, such as marking menus, allow rapid access to a relatively large set of commands, but their use is still limited in real applications because they are hidden, change with the context, may occlude some of the underlying application and do not cover the complete set of commands. As a result improvements are still required to better take advantage of the benefits of toolbars, i.e. offering a constant and always available set of commands, and advantages of contextual menus or keyboard shortcuts, i.e. rapid access to commands with minimal interruption of the interaction flow.

In this work, we explore a novel way of interacting with toolbars by using a mouse with multiple degrees of freedom, the Roly-Poly Mouse (RPM) [17] illustrated in Figure 1. The RPM has a hemispherical bottom, allowing roll and rotation gestures in addition to the regular translation. We propose to exploit roll gestures to select toolbar commands: selecting a toolbar item with Rolling-Menu requires rolling RPM in a predefined direction corresponding to the item, while translation of the RPM is used to control the application pointer. The resulting Rolling-Menu thus presents the advantage of keeping the application pointer in the working area, and therefore contributes to limit work flow interruption.

We first identify and describe the design dimensions of Rolling-Menu. These design dimensions include different mappings between roll gestures and command selection, as well as different activation and validation mechanisms. We implemented eight versions of Rolling-Menu based on different combinations of our design dimensions.

We then explore the performance of these techniques. We first experimentally establish that using Rolling-Menu require less time than using the Mouse to select items in a toolbar containing up to 14 items. We then study how Rolling-Menus can be used to support a more efficient

The definitive version will be published in CHI 2018, April 21–26, 2018, Montreal, QC, Canada

interaction with two different types of complex toolbars: "dropdown toolbars", containing dropdown lists, and "grid toolbar", i.e. a grid of items. We discuss the advantages and limitations of Rolling-Menus in such complex toolbars.

Our contributions are: 1) an exploration of the design space of the use of roll gestures to select toolbar items; 2) a study showing that Rolling-Menus take less time than the regular Mouse for selecting up to 14 items in a simple toolbar; and 3) a second study showing that Rolling-Menus also perform better than the Mouse for more complex types of toolbars.

STATE OF THE ART

The goal of Rolling-Menu is to reduce the object-command transition. We summarize the process of command selection and review previous work on rapid access to commands.

From object of interest to command selection

In most interactive situations, command selection is not the main task, and the user focuses on other objects of interest before and after selecting a command (i.e. text on a text editor, a graphical element on a graphic design application, etc.). Dillon et al. [6] identified two main stages in the process of command selection: object-to-command transition (or command selection), and command-to-object transition (back to object of interest). According to authors, the total cost of command selection should include both stages. Bailly at al. [3] described two additional stages: command-to-command transition (i.e. when a user applies several consecutive commands) and command-to-value transition (i.e. specifying parameters). In our work, we focus on the two main stages identified by Dillon. Our goal is to minimize the total time of object-to-command and back-to-object transitions.

Contextual menus

A major approach to minimize object-to-command transition is to bring the menu closer to the object of interest by using contextual menus. Contextual menus are most of the time invoked through a mouse right-click and can take different forms, from linear menus to circular or pie menus [11,12,20,22,29]. All these different forms of contextual menus share the same fundamental limitations: they only include a subset of the menus, they are context dependent (i.e. menu items vary for each invocation), are difficult to use when the mouse is close to the window border, and they are hidden while performing the main task. These limitations contrast with the properties that ensure the success of toolbars, used on most desktop applications: toolbars are always visible, and their content and position are stable over time. Our goal is to combine both approaches, i.e. minimizing the selection time as ensured by contextual menus and interacting with the popular toolbars.

Multimodal command selection

Another approach to reduce command selection time is to use additional input modalities in parallel with pointing. For instance, Hover Widgets [10], TiltMenu [24] and PushMenu [14] use stylus input dimensions to select commands: hovering, orientation and pressure respectively.

Multitouch input has also been used to offer fast command selection. Most of multitouch work has been carried in the context of tabletop or mobile touchscreens, such as Multi-Touch Menu [2] or Microroll gestures [21]. Recent work has also explored using multitouch on a laptop touchpad: MarkPad [7] consists of several gestural touchpad shortcuts to select commands. However our goal is to offer rapid access to toolbars on regular desktop computers, i.e. with a keyboard and a mouse.

Multi-DOF mice

Adding degrees of freedom (DoF) to the traditional mouse extends its command selection capabilities, while preserving the normal pointing interaction. Two main approaches have been used to add DoFs to the regular mouse. The first approach consists in combining the regular mouse with other input modalities through device composition [18]. For instance, LensMouse [27] consists of a mouse augmented with a touchscreen, and Inflatable Mouse [15] includes a pressure sensor.

The second approach consists in modifying the shape of the mouse to allow supplemental physical manipulations. Rockin'Mouse [4] and VideoMouse [13] are similar to a regular mouse, but they have a rounded bottom allowing tilt gestures. The Roly-Poly Mouse (RPM) [17] is a hemispherical mouse offering a larger amplitude of rolls gestures. For this reason, we chose to study the use of roll gestures with RPM to select toolbar commands. To our knowledge, no previous work on multi-DoF mice has evaluated a solution for shortening object-command transitions on toolbars.

ROLLING-MENU

We present Rolling-Menu, a new technique for selecting a toolbar item with a roll gesture using the Roly-Poly Mouse (RPM) [17]. After describing the technique, we detail four design dimensions that lead to different versions of the technique: roll range, size of roll sectors, activation and validation mechanisms.

Roly-Poly Mouse: roll direction and amplitude

The Roly-Poly Mouse (RPM) is a multidimensional hemispherical mouse can be translated, rotated and rolled.

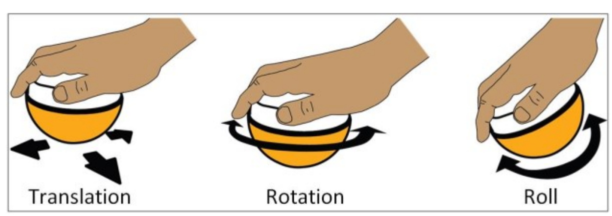

Figure 1. Physical manipulations that can be applied to the Roly-Poly Mouse independently or in a combined way.

Based on previous studies with this device, we decided to explore its rolling capabilities: translations are already used to control the application pointer, while rotations have a limited range and are not very comfortable [17]. A roll gesture is composed of two independent parameters:

- *Roll direction*: From a top-down point of view on RPM, it indicates the direction in which the user tilts RPM (Figure 3, left). Direction can range from -90° (tilt left) to +90° (tilt right), 0° corresponding to a tilt forward.
- *Roll amplitude*: From a side view, it corresponds to the inclination angle of RPM, once it has been rolled (Figure 3, centre). The amplitude is 0° when RPM is in its initial upright position, 90° when RPM is perpendicular to the surface on which RPM is used.

Selecting a toolbar item with Rolling-Menu

In this work we limit our explorations to horizontal toolbars, positioned on top of the window, similar to most applications toolbars. More specifically, we explore three variants of horizontal toolbars (see Figure 2):

- Simple toolbar: a single set of top-level items, where each item triggers a command;
- Dropdown toolbar: a toolbar where each top-level item is associated with a dropdown vertical list. Each sub-item from this list triggers a command;
- Grid toolbar: a toolbar structured into several blocks (hereafter referred to as the top-level items) that contain a grid of sub-items, each one associated with a command.

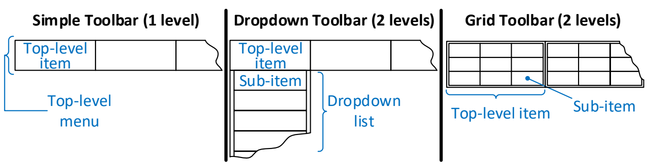

Figure 2. Three toolbar variants: a simple toolbar, a toolbar with dropdown lists and a grid-structured toolbar.

Using a Rolling-Menu to select a toolbar item relies on the activation of a quasimode [19], a specific mode of the application which ends automatically once the selection is performed, i.e. that does not require any dedicated action to exit. This quasimode is activated when the roll amplitude reaches a predefined threshold. In this quasimode the application pointer is frozen and the top-level menu is associated to a *range* of possible rolls (Figure 3, right). This range is divided into *sectors*, so that each top-level item is associated to one distinct sector. To select an item, the user rolls RPM in the direction of the corresponding sector.

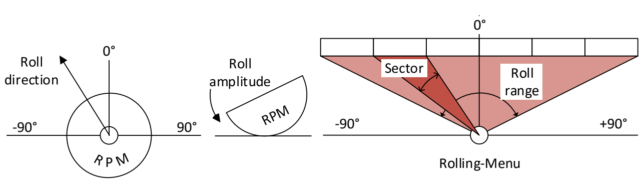

Figure 3. RPM roll direction (left: top-down view) and amplitude (centre: side view); Roll range and one sector of a Rolling-Menu (right: top-down view).

When the user validates the selection, it will execute the command corresponding to the top-level item (simple toolbar), open the dropdown list (dropdown toolbar) or select a grid block (grid toolbar). We explore in subsequent user studies different RPM manipulations to interact with these three variants of toolbars.

The benefits of a Rolling-Menu are twofold. First, it strongly decouples the menu navigation from the application pointer, reducing workflow interruptions due to object-command transitions (see first section of State of the art). Second, each item corresponds to a specific gesture and can therefore be encoded through muscle memory. This could potentially lead to a more effective expert mode [3].

Design dimensions

Four design dimensions result from the previous general principle. Two of them are related to the roll mapping (type of roll range and size of the roll sectors). The two other dimensions are related to the use of the Rolling-Menu quasimode, which needs to be activated, before the item selection is validated. Therefore we consider different activation and validation methods in our design space.

Roll Range

We envisioned two types of roll range:

- The *Direct-range* ensures a coherent mapping between the physical roll directions of RPM and the width of the toolbar (Figure 4 – left). In this case, the roll range is a triangle, the summits being the left and right extremities of the toolbar, and the origin of the Rolling-Menu.
- The *180-range* maximizes the roll range by mapping 180° of RPM physical roll directions (from -90° to +90°) to the toolbar width (Figure 4 – right). The roll range is here trapezoidal.

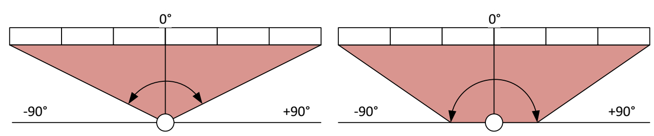

Figure 4. Direct-range (left) vs. 180-range (right).

We exclude roll directions above the 180° range because informal pre-tests revealed that the resulting mapping is hard to understand (as a user performs a roll back to select an item on the opposite direction), and previous studies showed that performing such rolls is less efficient [17].

Size of roll sectors

Independently of the roll range, we identified two different ways for establishing a mapping between a roll sector and a toolbar item, based on the size of the sectors:

- *Variable* size: Each sector is defined through a direct mapping between a toolbar item and the origin of the Rolling-Menu (Figure 5 – line 1). As a consequence, the size of each sector is different from each other and depends on the position of the toolbar item: sectors on the extremities of the toolbar are smaller than those near the centre. This ensures a straight mapping between the RPM roll direction and the position of the toolbar item, but can make some items more difficult to select.

- *Fixed* size: The roll range is divided into the same number of sectors than the number of toolbar items. As a consequence, the size of each sector is identical (Figure 5 – line 2), which offers the same selection difficulty for every item, but may lead to a mismatch between the roll direction and the item direction.

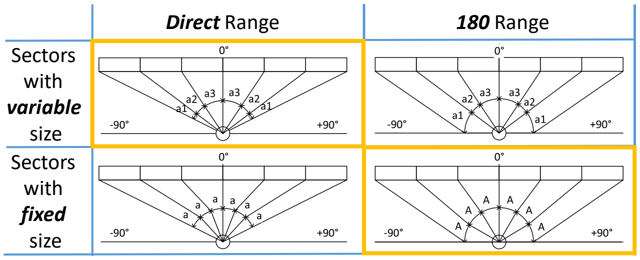

Figure 5. Sectors with variable size (line 1) or fixed size (line 2) used with a direct roll range (left column) or a range of 180° (right column).

Activation method of the Rolling-Menu

To activate the quasimode without affecting the user's interaction flow, we avoid using a modifier key. Instead, we exploit the roll amplitude and defined an activation threshold. We designed two alternatives:

- *Bottom* activation: The user activates the quasimode as soon as the roll amplitude is greater than a predefined small threshold. We know that unintentional rolls occur when translating the RPM, preventing the use of any roll below 12° [17]. Therefore, rolls over 12° activate the quasimode (see Figure 6-left).
- *Top* activation: To activate the quasimode, the user needs to perform a roll with an amplitude greater than a predefined big threshold. Since rolls are considered comfortable up to 37° [17], we adopt this value as the big threshold. The advantage of this method is that the sector arcs are wider as roll amplitude increases, offering more precision to select an item (see Figure 6-right).

Validation mechanism

We considered two mechanisms to validate the selection of a toolbar item and trigger the execution of a command or the opening of the sub-menu.

- *Tap*: The user can perform a finger's tap (on any tactile surface underneath RPM or on the keyboard), to select a sector when the roll amplitude of RPM is over the activation threshold.
- *Roll*: it is based on a validation threshold depending of the activation method. For a Bottom activation, validation occurs when roll amplitude is greater than a validation threshold of 37°. A complete activation and validation gesture in this case consists on a straight roll. For a Top activation, validation occurs when roll amplitude becomes smaller than a validation threshold of 12°. A complete gesture in this case consists on a roll forward to activate, then a roll backward to validate. Validation thresholds (37° with Bottom activation, and 12° for the Top activation) were chosen so that the same angle exists between the activation and validation thresholds, whatever the activation method.

IMPLEMENTING ROLLING-MENUS

Our four design dimensions produce a large number of design combinations (16 possible Rolling-Menus). We decided to implement and study a subset of 8 Rolling-Menus after an analytical and empirical exclusion of the others. We carried an iterative process to design the visual feedback for these versions and implement the input apparatus.

Choice of Rolling-Menu versions

Among the four possible combinations of Roll range and Size of sectors, we selected two of them, hereafter referred to as RMDirect and RM180:

- RMDirect (Figure 5, top left) combines a **Direct range** with a **variable** sector size. It offers the most direct mapping between roll actions on RPM and the position of toolbar items: the roll range corresponds to the width of the toolbar and the roll direction corresponds to the direction in which the toolbar item is.
- RM180 (Figure 5, bottom right) combines a **180 range** with a **fixed** sector size. It minimizes the accuracy required by maximizing the size of sectors: the roll range is extended to 180° and each sector has the same size, which depends on the number of toolbar items.

We implemented these two designs with the two methods of validation (tap or roll) and activation (bottom or top), resulting in 8 different interaction techniques, hereafter referred to as the Rolling-Menus (RM).

Implementing the validation mechanisms

We tried to insert various forms of button on top of RPM. However our pre-tests showed that using a physical button on RPM altered the device handling gesture and brought a number of technical issues (button position, etc.). Instead, we considered the use of a tactile surface underneath RPM to detect a user's finger tap: the user can employ any finger of the same hand that manipulates RPM to tap on the surface, although participants seemed to prefer the thumb. An algorithm associates the first touch on the tablet to the RPM position, and triggers a tap event only when detecting a second touch. Alternatively, the user can press a key on the keyboard with the non-dominant hand: as the user's main task is probably involving keyboard input, this bimanual setting offers a fluid interaction compatible with regular keyboard input (the keyboard is only used as a validation once the Rolling-Menu quasimode is activated).

Regarding the roll validation, we also decided to test an additional threshold to increase the robustness of the validation: the selection threshold. Between this selection threshold and the validation threshold, a modification of the roll direction does not change the sector selected (i.e. the sector is locked). We empirically established that an angle of 7° between the selection and validation thresholds was the most appropriate. Selection thresholds are therefore 30° with

Bottom activation method and 19° with Top activation method (see Figure 6).

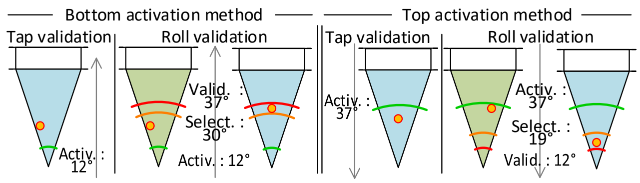

Figure 6. Graphical feedback provided in the 4 possible combinations of validation and activation methods.

Visual feedback

After the activation of the Rolling-Menu quasimode (through a roll), a visual feedback shows the roll sectors, from the centre of the screen to the toolbar items. In addition a coloured circle (10px diameter) is displayed: its position conveys the RPM roll direction and amplitude (see Figure 6). Finally the selected sector is coloured: it becomes blue as soon as the selection can be validated.

RPM Input apparatus

To interact with the Rolling-Menu, we enhanced the original Roly-Poly Mouse [17]. As the original one, our version consists of a sphere with a diameter of 8 cm, which includes a Bluetooth enable Inertial Measurement Unit (xIMU by xIO Tech – sensor rate: 512 Hz, angular accuracy: 1°). In comparison to the original RPM, our version was placed on a Wacom Intuos 3D tablet (216x135 mm, resolution: 2540 lpi). As the tablet is multitouch, it can detect the translation of RPM and finger taps (Figure 7-right). We therefore covered the RPM surface with a graphite lacquer (Graphit 33 – Kontakt Chemie) to give the device a conductive coating (Figure 7-left and centre).

Figure 7. Overview of the RPM input apparatus (left): rolling RPM (centre) and tapping on the tablet to validate (right).

PRELIMINARY STUDY: ADJUSTING ROLLING-MENU SETTINGS

We performed a preliminary user study to fine tune our design dimensions (in particular the roll thresholds). The goal was also to assess the ability of Rolling-Menu to tackle a large set of items and to study the impact of the activation point (i.e. the centre of the roll sectors).

Design and procedure

The task consisted in selecting items from a simple toolbar positioned on the top of the screen and containing 4 to 10 items. The activation point was placed vertically at the middle of the screen, and horizontally in one of three different positions: left, centre or right of the screen. Toolbar items were 25px height and their width varied from 90 to 225px according to the number of items.

The eight Rolling-Menus we implemented (see section "Choice of Rolling-Menu versions") were compared to the Mouse. With the Mouse, the task consisted in clicking on the starting point (a circle with a diameter of 100px), selecting the highlighted top-level item and returning to the starting point. Completion time was measured between the two mouse-press events. This task simulates the object-to-command and command-to-object transitions, which are all part of the total command selection time, as explained in the related work section. With Rolling-Menus, the application pointer is separated from the roll interaction, and selecting a toolbar item consists simply in rolling the RPM in the direction of the toolbar item to select.

The toolbar and feedback were displayed on a 17" screen (1280px by 1024px). Participants were sitting in front of it. We used the keyboard spacebar as validation method. Twelve participants (aged 29.4 on average; SD=9.5) took part in this experiment. All of them were University students or researchers. After a training period, each participant performed 9 techniques x (4+6+8+10) top-level items x 3 starting points = 756 trials. They filled a SUS questionnaire for each technique.

Data analysis

Regarding the data analysis, we chose to rely on estimation techniques with 95% confidence intervals and ratio analysis as recommended by the APA [25]. Ratio is an intra-subject measurement that expresses the effect size (pair-wise comparison) and is computed between each of the geometric means. All CIs are 95% BCa bootstrap confidence interval. For the reader more used to interpret the p-values, a parallel might be drawn with results obtained through the estimation technique and CIs reports (see Figure 3 in [16]). Scripts used to compute the geometric mean and confidence intervals were used in [26] and are available online [1].

Results

Results established that on average Rolling-Menus using the tap validation always takes less time (1322ms CI[1155;1563]) than the Mouse (1719ms CI[1562,1918]) to select a top-level item, even when removing the time of the final mouse press (74ms on average). They also established that all Rolling-Menus with 4 or 6 items always require less time than the Mouse (from 14.2%-CI [7.9%, 22.8%] to 34.0%-CI [24.3%, 43.4%]) to select an item. The accuracy ranges from 94% to 97.5% for the 9 techniques. We thus decided to consider larger toolbars, with up to 14 items.

Results also highlighted that with the Mouse, the time taken from the menu selection and back to the starting point represent on average 47% of the total time, thus justifying the need to avoid the object-to-command transition. In addition it appeared that selecting an item requires on average 17.7% less time when the starting point is on the centre than on the left or right of the screen. Using always the same origin for the Rolling-Menu would in addition offer a more stable interface and could facilitate command memorization since each toolbar item would be associated to a unique roll direction. For the following experiments we therefore considered the central starting point only.

We finally observed that, for the tap validation method, the average roll amplitude when the spacebar is pressed is 30.9°, CI[27.0°; 34.7°]. Informal comments also stated that the roll amplitude of the validation threshold was too large. In addition, the SUS scores of Rolling-Menus using roll validation mechanism were the lowest (72 on average, vs. 85 for the other RM and 90 for the Mouse). From these results it appears necessary to reduce the roll amplitude required when validating (bottom activation) or activating (top activation). This modification also affects the selection threshold. We thus empirically defined new thresholds for a Bottom activation (activation at 15°, selection at 26° and validation at 31°) and for a Top activation (activation at 31°, selection at 29° and validation at 24°).

STUDY 1: ITEM SELECTION IN A SIMPLE TOOLBAR

The goal of this study is to explore the impact of the four design dimensions (roll range, sector size, activation method and validation method) on the selection of a simple toolbar item using the eight Rolling-Menus, compared to a baseline (Mouse). We applied the settings derived from the preliminary study. The goal is also to study the impact of the number of toolbar items on completion time, accuracy and user preference.

Study description

Task and instruction

The task consisted in selecting items in simple toolbar displayed on top of the screen. The toolbar included 8, 10, 12 or 14 items. To limit the length of the experiment, we preliminary identified, in each toolbar, 8 items uniformly spread over the width of the toolbar: these items were then the only items targeted during the experiment. The height of the toolbar items was 25px and their width ranged from 102 to 180px depending on the number of items in the toolbar. We asked participants to perform the selection task as quickly as possible and with accuracy.

Apparatus

We used the same screen and mouse than in the preliminary experiment. We used the RPM apparatus described in the Implementation section. Tap validation was performed on the underlying touch surface with the hand holding RPM, in order to validate the use of a more integrated RPM version.

Participants

We recruited 16 participants (8 female, 2 left-handed), aged 27.9 years on average (SD=9.9). All of them were University students or researchers. Six of them took part in the preliminary study. The experiment lasted 75min on average.

Design

This experiment followed a 4x9 within-subjects design with Number of top-level items (8, 10, 12 and 14) and Interaction techniques (Mouse and eight Rolling-Menus) as factors. Each session was divided into 9 blocks, each block corresponding to one Interaction technique (the Mouse or one of the eight Rolling-Menus). Half of the participants used the Mouse prior to the eight Rolling-Menus, while the other half used the Mouse after them. The eight Rolling-Menu blocks were counterbalanced across participants by means of a 4x4 Latin Square.

Procedure

Each participant completed nine blocks. Each block contains a training and an experimental session. The training consists in selecting 8 top-level items in four successive toolbars containing 8, 10, 12 or 14 items. The training session contains 8 items x 4 toolbar sizes = 32 trials. The experimental session is similar to the training session but with two repetition for each toolbar size. It contains 8 items x 2 repetitions x 4 toolbar sizes = 64 trials. Participants had the possibility to take a break between each trial. Overall, each participant performed 9 blocks x 64 trials = 576 trials (without training). In total we collected 16 participants * 576 trials = 9216 trials.

Collected data and data analysis

We logged all tracking data (RPM rolls, translations and taps) and measured completion time from stimulus onset. Participants also had to fill in a SUS questionnaire after each block (i.e. for each technique). Data analysis is performed with the same approach than in the preliminary study. As underlined in [28], many studies on menu techniques have focused on selection time, accuracy and learnability. In our case we did not address learnability and focused on time and accuracy, since our goal is to reduce object to command transitions [3].

Results

We first report quantitative results and then discuss qualitative results. We use the following naming convention to refer to the eight implemented Rolling-Menus: RM_"Activation"_"Range"_"Validation" where:

- "Activation" is T for Top or B for Bottom;
- "Range" is 180 for large or Dir for direct roll range;
- "Validation" is Roll or Tap.

Quantitative results

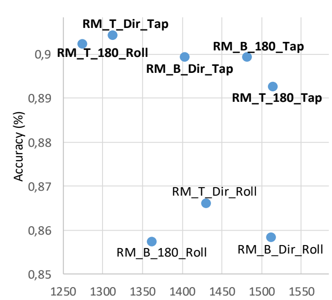

Figure 8. Cross-analysis of selection time (ms) and accuracy (%).

Among the eight techniques compared to the Mouse, a group of five Rolling-Menus offered the best accuracy, with selection times similar to the three remaining ones (five Rolling-Menus bolded in Figure 8). We focus on these five techniques and compare them in detail with the Mouse in terms of selection time and accuracy.

Selection time analysis. Computing averages and 95% confidence intervals for the selection time of the Rolling-

Menus establishes that selecting a toolbar item with any Rolling-Menu (average of all Rolling-Menus: 1410ms, CI[1222; 1658]) is faster than with the Mouse (1790ms, CI[1705; 1886]), as detailed in Figure 9-left.

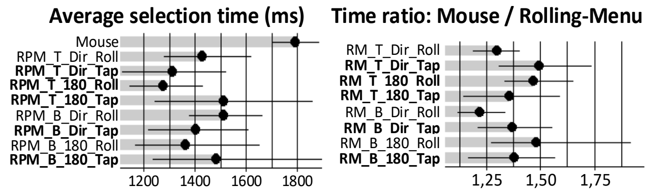

Figure 9. Average selection time in ms (with 95% CIs) (left), and time ratio (with 95% CIs) of the Mouse / Rolling-Menu.

When focusing on the five Rolling-Menus identified in the cross analysis (selection time X accuracy), three of them are definitely faster (RM_T_Dir_Tap: 1312ms, CI[1116;1523]; RM_T_180_Roll: 1274ms, CI[1146;1432], RM_B_Dir_Tap : 1403ms, CI[1217; 1610]) than the Mouse to select a top-level item. Due to the large confidence intervals of the 2 other Rolling-Menus (RM_T_180_Tap and RM_B_180_Tap), which intersect with the Mouse confidence interval, the difference is less obvious. The intra subject analysis based on the time ratio (Mouse/Rolling-Menu) confirms the differences already highlighted between these different techniques (Figure 9-right): the ratio is above 1 (meaning that from an intra-subject point of view, the selection time with the mouse is greater than the selection time with the Rolling-Menu) and the corresponding CIs do not intersect with the value 1.

Accuracy analysis. Now, regarding the average accuracy of each interaction technique, the same analysis establishes that the Mouse is more accurate (95.1%, CI [93.6; 96.3]) than any of the Rolling-Menus (average of all Rolling-Menus: 89.0%, CI[85.0%; 92.0%], see details in Figure 10-left); the analysis of the accuracy ratio (Mouse/Rolling-Menu) confirm these conclusions (see Figure 10-left). The five Rolling-Menus identified in the cross-analysis are very similar in terms of accuracy (average of the five techniques: 90.0%, CI [86.9; 92.3], see details in Figure 10-left) and they are clearly but only slightly more accurate than the three other Rolling-Menus (average of the three techniques: 86.0%, CI[79.2%; 90.1%], see details in Figure 10-left). The ratio analysis also confirms the validity of this result (Ratio>1 and CIs not intersecting the value 1¸ see Figure 10-right).

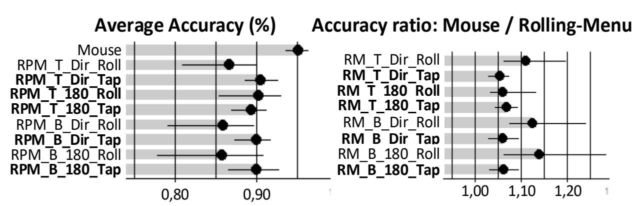

Figure 10. Average accuracy percentage (with 95% CIs) for each technique (left), and accuracy ratio (with 95% CIs) of the Mouse / Rolling-Menu.

These results in terms of selection time and accuracy still hold for any toolbar size (8, 10, 12 or 14 items): selecting an item in a simple toolbar with any Rolling-Menu is always faster than with the Mouse (Figure 11 – left), but with less accuracy (Figure 11 – right). As expected, we also notice that selection time increases and accuracy decreases when the toolbar size augments (Figure 11 – left).

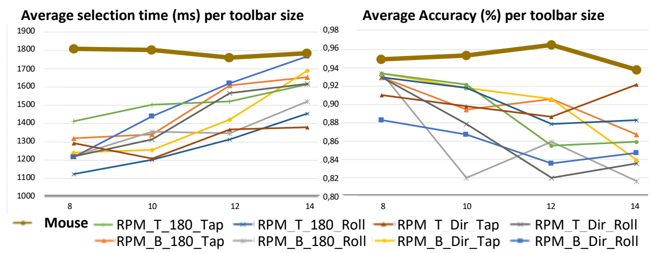

Figure 11. Average time (ms) and accuracy (%) for each technique with the four toolbar sizes (Mouse line is thicker).

Complementary analysis. Further data analysis, focusing on the design dimensions independently, did not reveal any major and clear distinction in terms of **activation method** (Top: 1382ms, CI[1219;1546]; Bottom: 1439ms, CI[1293;1636]), in terms of **validation method** (Roll: 1394ms, CI[1287;1551]; Tap: 1427ms, CI[1238;1643] or in terms of **roll range** (180: 1407ms, CI[1234;1626]; Direct:1414ms, CI[1291;1549]). We can therefore conclude that it is the combination of these design dimensions that affects the overall performance of the technique.

Finally, results establish that the average selection time of the first and second repetitions are very similar (repetition1: 1891ms, CI[1756; 2047]; repetition2: 1858ms, CI[1716; 2014]). We conclude that the training session was sufficient and that it is easy to learn how to use the Rolling-Menus.

Qualitative results

We computed SUS scores to assess the usability of the techniques. The Mouse reaches an average score of 80.8 while the average SUS score of the 8 Rolling-Menus is 72.0, which corresponds to a "good" usability level [5]. SUS scores obtained by the three best Rolling-Menus are above this average (RM_T_Dir_Tap: 76.0]; RM_T_180_Roll: 73.3; RM_B_Dir_Tap: 71.3).

Summary

Results of the first experiment strongly identify three of the eight Rolling-Menus as the best techniques for selecting an item in a simple toolbar:

- RM_T_Dir_Tap: Top activation, direct roll range, validation with tap,
- RM_T_180_Roll: Top activation, 180° roll range, validation via RPM roll,
- RM_B_Dir_Tap: Bottom activation, direct roll range, validation with tap.

They take on average 25% less time than the Mouse, with an accuracy above 90%, to select a simple toolbar item. These three Rolling-Menus therefore represent the best combinations of our design dimensions for selecting an item in a simple toolbar with up to 14 items.

STUDY 2: DROPDOWN AND GRID TOOLBARS

The goal of this experiment is to compare the three best Rolling-Menus from the first study with the Mouse, for selecting a sub-item in two different toolbar variants: the dropdown and grid toolbars (cf. Figure 2).

Study description

Task and instruction

The task consisted in selecting one top-level item, and then one sub-item in a Dropdown toolbar and a Grid toolbar. Since experiment 1 revealed that accuracy tends to decrease when the size of the toolbar increases, we limit this experiment to toolbars containing 8 top-level items. As illustrated in Figure 2, each top-level item is associated to 1) a dropdown list containing 8 sub-items for the Dropdown toolbar, or 2) a 3x3 grid of sub-items for the Grid toolbar. We asked participants to perform each selection as quickly and accurately as possible.

Selecting all items from both levels would make the overall experiment too long, so we decided to predefine target positions, balancing their difficulty. For the top-level selection, we predefined 4 positions (2, 4, 5 and 7, from left to right) to cover the left, middle and right of the toolbar. In the dropdown lists, we predefined 3 positions (2, 4 and 7, from top to bottom) to cover the top, middle and bottom of the list. In the Grid toolbar, we randomly selected 3 positions among the 3x3 possible sub-item positions to cover all possible directions.

Apparatus

We used the same apparatus than in the first experiment. The same feedback was provided during the first phase of the task, i.e. the selection of the top-level item. For Rolling-Menus using the Tap validation method, we asked participants to use a bimanual approach (pressing a keyboard key with the non-dominant hand): our pre-tests showed that while taping on the underlying tablet with the hand holding RPM worked well to select a top-level item, it disrupted the fluidity of a two-level selection. Indeed, a sub-item selection requires a rapid sequence of rolls / translations of RPM and the pre-tests revealed that moving the thumb to tap on the tablet is difficult to perform during this rapid combination of RPM gestures. As explained earlier, this bimanual validation does not specifically requires to move the non-dominant hand, which is usually already on the keyboard. Further, it does not interfere with the regular use of the keyboard since the key validation only works when the user activates the menu quasimode.

With the dropdown toolbar, rolling RPM towards the targeted top-level item and validating it opens the dropdown list. To navigate through the list of sub-items, the user rolls RPM forward or backward. A final tap validates the sub-item selection.

With the grid toolbar, rolling RPM towards the target top-level item and validating it highlights the central sub-item of the 3x3 grid. To select one of the 8 surrounding sub-items on the grid, the user translates RPM in the corresponding direction (similar to a Marking Menu). Validating the central item selection was based on a tap input. While using two different validation methods according to the sub-item position (tap for the central item, translation for the others) could seem to be cognitively complex, our pre-tests showed that participants did not mix up both types of validation.

Participants

We recruited 12 participants (5 female, 2 left-handed), aged 28.8 years on average (SD=8.8). All of them were University students or researchers. Eight of them took part in the preliminary study and/or the first experiment. The experiment lasted one hour on average.

Design and procedure

This experiment followed a 4x2 within-subjects design with Interaction techniques (Mouse and three Rolling-Menus) and Toolbar variant (Dropdown toolbar and Grid toolbar) as factors. Each participant completed 2 phases, each one corresponding to one toolbar variant. We used the same order for all participants (Dropdown then Grid), as we did not want to compare the toolbars between them. Each phase was composed of 4 blocks corresponding to each interaction technique. Blocks were counterbalanced across participants by means of a 4x4 Latin Square.

For each technique, participants first had a training session composed of 2 series: each series consisted of selecting 3 sub-items in the 4 predefined top-level items (i.e. 2x3x4=24 training trials). The experimental session then followed the same procedure and was composed of 6 series, corresponding to 6x3x4=72 trials. Participants had the possibility to take a break between each trial.

Overall each participant performed 2 phases (toolbar variant) x 4 blocks (techniques) x 72 trials = 576 trials (without training). Over the 12 participants we collected 6912 trials in total.

Collected data and data analysis

We logged all tracking data and measured completion time from stimulus onset. We also asked participants to fill in a SUS questionnaire after each block (i.e. for each combination of technique and toolbar variant). At the end of each phase (i.e. toolbar variant), participants were requested to rank the four techniques according to their preference and were invited to comment about them. Data analysis is performed according to the same approach than in study 1.

Results

We first discuss selection time and accuracy for both types of toolbars before reporting qualitative feedback.

Quantitative results

Selection time. The average selection time per technique tends to establish that on a Dropdown toolbar (Figure 12-Top), Rolling-Menus with Tap validation require less time than the Mouse and the other Rolling-Menu for selecting a sub-item (RM_T_Dir_Tap: 2099ms, CI[1862;2527]; RM_B_Dir_Tap: 2263ms, CI[1940;2658]); Mouse: 2462ms, CI[2273;2628]; RM_T_180_Roll: 2465ms, CI[2230;2830]). The intra-subject analysis based on the time ratio Mouse/Rolling-Menu (Figure 12-Top right) strongly confirms that Rolling-Menus with Tap validation took up to 18% less time than the Mouse (ratio=1.22 and 95% CIs not intersecting the value 1.0). This ratio analysis also shows that no obvious difference can be established between RM_T_180_Roll and the Mouse.

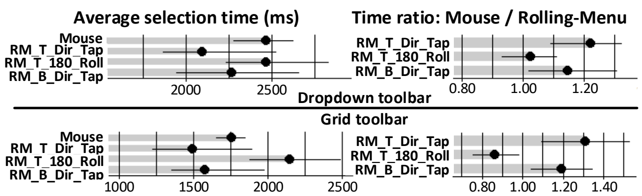

Figure 12. Sub-item selection time in a Dropdown toolbar (top) and Grid toolbar (bottom). On the left, the average selection time in ms and, on the right, time ratio of the Mouse / Rolling-Menu (all bars represent 95% CIs).

Regarding the selection of a sub-item in a Grid toolbar (Figure 12-bottom), Rolling-Menus with Tap validation also require less time than the Mouse and the other Rolling-Menu (RM_T_Dir_Tap: 1493ms, CI[1222;1894]; RM_B_Dir_Tap: 1577ms, CI[1348;1980]); Mouse: 1753ms, CI[1653;1853]; RM_T_180_Roll: 2146ms, CI[1872; 2491]). From the ratio analysis, we can strongly conclude that these two Rolling-Menus takes up to 24% less time than the Mouse (ratio=1.31 and 95% CIs not intersecting the value 1.0). The ratio analysis also firmly establishes that RM_T_180_Roll takes 14% more time than the Mouse.

Accuracy. When selecting a sub-item in a Dropdown toolbar, the accuracy of the three Rolling-Menus is on average 95.6%, CI[94.1%; 97.5%], very similar to the accuracy of the Mouse (97.3%, CI[96.1%; 98.4%]). Results are detailed in Figure 13-Top.

When using a Grid toolbar the accuracy obtained with Rolling-Menus using the Tap validation reaches on average 84.2%, CI[76.5%; 88.0%]. Accuracy with the Rolling-Menu using a Roll validation technique (RM_T_180_Roll) is below 60% while the Mouse shows an accuracy of 94.7%, CI[89.4%; 97.1%]). These results are further detailed in Figure 13-Bottom. The ratio analysis strongly confirm that the Roll version is less accurate than the Mouse. The intra-subject difference between the Mouse and the two others is also confirmed: the Mouse is on average 8.2% more accurate than RM_T_Dir_Tap and 15.9% than RM_B_Dir_Tap. Two reasons might explain this drop of accuracy: the gestures with the mouse (translation) and the RPM (roll) are different, and the user's expertise is optimized for the mouse but only at novice level with RPM.

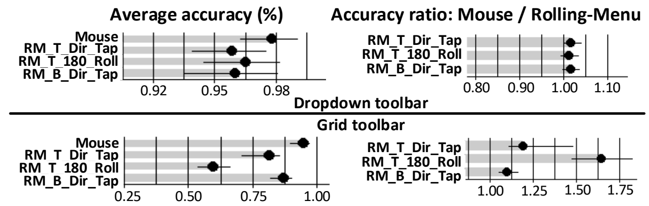

Figure 13. Accuracy when selecting a sub-item in a Dropdown toolbar (top) and Grid toolbar (bottom). On the left, average accuracy in % and, on the right, accuracy ratio of the Mouse / Rolling-Menu (all bars represent 95% CIs).

Complementary analysis. We observed for both types of toolbars that selection time during the training phase was on average 12% longer than during the experiment. We found no evolution of selection time during the six series of the experimental phase. The training phase was thus sufficient for participants to familiarize with the Rolling-Menus.

We also observed that the results discussed above still apply when considering every predefined sub-item independently. Roll direction and amplitude do not affect the user's interaction efficiency.

Qualitative results

We computed the SUS score for each technique with every toolbar. Overall the usability of the Mouse is rated "excellent" (average SUS: 91.6). With the Dropdown toolbar, the usability of Rolling-Menus using a Tap validation method is "good" (75.5 on average). However the usability of Rolling-Menus using a Roll validation method (RM_T_180_Roll) is "highly marginal" (average SUS: 61.0). With the grid toolbar, Rolling-Menus using a Tap validation method obtains a score of 67.8 on average and almost corresponds to a "good" acceptability, while the one with roll validation obtains 37.3, which is below an acceptable usability. We propose some perspectives to address this issue in the discussion section. As already mentioned, the roll validation requires a high precision, which might be annoying and thus justify the low score of the Rolling-Menu with Roll validation.

Summary

The results of this second experiment establish that, with Dropdown and Grid toolbars, two versions of Rolling-Menu are faster than the Mouse while offering a very good (Dropdown toolbar: 95.6%) or good accuracy (Grid toolbar: 84.2%). Rolling-Menus are thus an efficient solution for roll-based command selection in toolbars.

DISCUSSIONS AND PERSPECTIVES

Factors influencing Rolling-Menu performance

In this work, we established that the use of rolling gestures with a multi-DOF Mouse is an efficient solution for interacting with different types of toolbars. Through a comparison of eight Rolling-Menus, we determined that their efficiency is influenced by the combination of various design dimensions (activation, roll range and validation method), rather than by each design dimension on its own. In particular three configurations emerge as being more efficient than the others for interaction with a simple toolbar (RM_T_Dir_Tap, RM_T_180_Roll, RM_B_Dir_Tap).

Among them, those based on the Tap validation method outperform the one based on Roll validation in complex toolbars (dropdown and grid toolbars) containing 64 to 72 sub-items. The Tap validation method appears to be particularly well suited for use with Direct roll mapping, whatever the activation method (Top or Bottom) and the type of toolbar (simple, Dropdown or Grid). Roll validation method seems more appropriate for 180 roll range as the sector size is larger.

While Top activation requires longer gestures to activate the quasimode, we did not observe clear differences in selection time and accuracy with the Bottom activation method in our second study. Intuitively, the Top activation with Tap validation should be a good solution for newcomers, as it uses the widest part of the sectors, while experts could benefit from the shorter gestures using a Bottom activation. Further experiments focusing on the benefits of this version for experts are planned.

Rolling-Menus in existing applications

We confronted Rolling-Menu to different types of toolbars that correspond to concrete usages in common applications: simple toolbars with up to 14 items are for example used in Thunderbird (get mail, write, address, etc.); dropdown toolbars with up to 8x8 items are for instance used in Gimp; a grid-toolbar with up to 9x9 items corresponds to the tool palette in Photoshop. Rolling-Menu can therefore easily be introduced in everyday applications.

Toolbars in existing applications usually result from a combination of the three types of toolbar explored in our work: they include lists as in our Dropdown toolbar, buttons in a single row as in our Simple toolbar, or in multiple rows as in our Grid-toolbars. Since Rolling-Menu performs better than the Mouse in these different cases individually, it will theoretically still perform better on a more complex toolbar. Another difference with the toolbars of our studies is that toolbar items do not have necessarily the same width. In this case, the activation of our quasimode could momentarily spread the items over the width of the window to facilitate their selection by maximizing their size.

Further studies will focus on the use of Rolling-Menu in different contexts: vertical toolbars induce wrist biomechanical constraints that will probably affect left-right rolls differently than up-down rolls; Microsoft menu combine Tab-panes with Ribbons and thus requires to adapt and assess the use of Rolling-Menu; and finally a valuable follow-up will involve domain experts to compare memorized keyboard shortcuts to muscular memory with Rolling-Menu.

Technical limitations

The current implementations of Rolling-Menu are affected by the lack of resolution of the Inertial Measurement Unit. In addition, the Wacom tablet detects residual translations during rolls, which disrupts the correct detection of the direction and length of translation gestures. These aspects induced limitations in the ability to detect accurate or compound gestures (i.e. roll + translate). For example, when selecting a sub-item in a grid toolbar (study 2), a RPM roll followed by a RPM translation are used to select and validate the appropriate sub-item: in this situation we observed that the Rolling-Menu accuracy decreased to 84%. We believe that adopting a more accurate technology would lead to better results and richer usages of Rolling-Menu: the mouse laser would be the ideal solution, but its adaptation to a spherical device remains challenging; the detection of RPM micro-rolls might be another promising solution.

Using a tap uni-manually on the underlying tablet, or bimanually on the keyboard, ensures a consistent interaction: the user is moving the mouse with one hand, and most often leaving the other hand on the keyboard to continue typing afterwards. However, the best solution would be a more integrated validation mechanism activated by the interactive hand. To this end we envision integrating a touch or tactile sensor on the RPM itself, instead of a button in our first tests. Such an approach has already been explored in TDome [23] but only in combination with physical gestures. The challenge in this case is how to place the sensor on the device without changing the grip, losing comfort or triggering accidental touch events.

Addressing these technical limitations will allow to carry longitudinal studies to evaluate the ergonomic aspects of RPM and Rolling-Menu.

CONCLUSION

In this work we presented different roll-based techniques for command selection in toolbars. The aim of these techniques is to minimize disruptive transitions between the working area and the toolbar. To do so, Rolling-Menus rely on the detection of roll gestures performed in the direction of the toolbar items to select. Based on different design dimensions, we proposed 8 versions of Rolling-Menu that we compared to the Mouse for selecting a top-level item in a Simple toolbar, or sub-items in more complex toolbars, i.e. a Dropdown and a Grid toolbar. Our user studies demonstrated that two Rolling-Menus reduce the selection time for a top-level item or sub-item, while keeping a good or very good accuracy.

ACKNOWLEDGMENTS

This work is partly funded by the AP2 project (research grant AP2 ANR-15-CE23-0001). Authors wish to thank Mathias Faure for his early design contributions to Rolling-Menu and the first implementations of the techniques.

REFERENCES

1. AVIZ Group. R Macros for data analysis. www.aviz.fr/reliefshearing.
2. Bailly, G., Demeure, A., Lecolinet, E., and Nigay, L. MultiTouch menu (MTM). Proceedings of the 20th

International Conference of the Association Francophone d'Interaction Homme-Machine on - IHM '08, ACM Press (2008), 165. DOI=http://dx.doi.org/10.1145/1512714.1512746

3. Bailly, G., Lecolinet, E., and Nigay, L. Visual Menu Techniques. ACM Computing Surveys 49, 4 (2016), 1–41. DOI: https://doi.org/10.1145/3002171
4. Balakrishnan, R., Baude, T., Kurtenbach, G., and Fitzmaurice, G. The Rockin'Mouse: Integral 3D Manipulation on a Plane. Proceedings ACM SIGCHI Conference on Human Factors in Computing Systems, (1997), 311–318. DOI: https://doi.org/10.1145/258549.258778
5. Bangor, A., Kortum, P.T., and Miller, J.T. An empirical evaluation of the System Usability Scale. International Journal of Human-Computer Interaction 24, 6 (2008), 574–594.
6. Dillon, R.F., Edey, J.D., and Tombaugh, J.W. Measuring the true cost of command selection: techniques and results. Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90, April (1990), 19–26. DOI: https://doi.org/10.1145/97243.97247
7. Fruchard, B., Lecolinet, E., and Chapuis, O. MarkPad: Augmenting Touchpads for Command Selection. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17, ACM Press (2017), 5630–5642. DOI: https://doi.org/10.1145/3025453.3025486
8. Giannisakis, E., Bailly, G., Malacria, S., and Chevalier, F. IconHK: Using Toolbar button Icons to Communicate Keyboard Shortcuts. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17, ACM Press (2017), 4715–4726. DOI: https://doi.org/10.1145/3025453.3025595
9. Grossman, T., Dragicevic, P., and Balakrishnan, R. Strategies for accelerating on-line learning of hotkeys. Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07, ACM Press (2007), 1591. DOI: https://doi.org/10.1145/1240624.1240865
10. Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., and Balakrishnan, R. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices. CHI 2006, ACM Press (2006), 861–870. DOI: https://doi.org/10.1145/1124772.1124898
11. Guimbretiére, F. and Winograd, T. FlowMenu - Combining Command, Text, and Data Entry. Proceedings of the 13th International Symposium on User Interface Software and Technology (UIST'00) 2, (2000), 213–216. DOI: https://doi.org/10.1145/354401.354778
12. Hascoët, M. Throwing models for large displays. Proceedings of HCI2003, British HCI Group (2003), 73–77.
13. Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., and Conway, M. The VideoMouse: : a camera-based multi-degree-of-freedom input device. Proceedings of the 12th annual ACM symposium on User interface software and technology - UIST '99, ACM Press (1999), 103–112. DOI: https://doi.org/10.1145/320719.322591
14. Huot, S., Nancel, M., and Beaudouin-Lafon, M. PushMenu: Extending Marking Menus for Pressure-Enabled Input Devices, Research Report LRI-RR1502. Paris, 2008.
15. Kim, S., Kim, H., Lee, B., Nam, T.-J., and Lee, W. Inflatable mouse: volume-adjustable mouse with air-pressure-sensitive input and haptic feedback. Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08, ACM Press (2008), 211–224. DOI: https://doi.org/10.1145/1357054.1357090
16. Krzywinski, M. and Altman, N. Points of Significance: Error bars. Nature Methods 10, 10 (2013), 921–922.
17. Perelman, G., Serrano, M., Raynal, M., Picard, C., Derras, M., and Dubois, E. The Roly-Poly Mouse: Designing a Rolling Input Device Unifying 2D and 3D Interaction. Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems 1, (2015), 327–336. DOI: https://doi.org/10.1145/2702123.2702244
18. Perelman, G., Serrano, M., Raynal, M., Picard, C., Derras, M., and Dubois, E. DECO: A Design Space for Device Composition. Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16, ACM Press (2016), 435–446. DOI: https://doi.org/10.1145/2901790.2901893
19. Raskin, J. and Jef. The humane interface : new directions for designing interactive systems. Addison-Wesley, 2000.
20. Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. IFIP TC 13 International Conference on Human-Computer Interaction - Interact'99, Springer, Berlin, Heidelberg (2009), 616–619. DOI: https://doi.org/10.1007/978-3-642-03655-2_69
21. Roudaut, A., Lecolinet, E., and Guiard, Y. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. Proceedings of the 27th international conference on Human factors in computing systems - CHI 09, ACM Press (2009), 927. DOI: https://doi.org/10.1145/1518701.1518843

22. Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E., and Eagan, J. Augmented letters. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13, ACM Press (2013), 2325. DOI: https://doi.org/10.1145/2470654.2481321
23. Houssem Saidi, Marcos Serrano, Pourang Irani, and Emmanuel Dubois. 2017. TDome: A Touch-Enabled 6DOF Interactive Device for Multi-Display Environments. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 5892-5904. DOI: https://doi.org/10.1145/3025453.3025661.
24. Tian, F., Xu, L., Wang, H., et al. Tilt Menu : Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-based User Interfaces. Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems, (2008), 1371–1380. DOI: https://doi.org/10.1145/1357054.1357269
25. VandenBos, G.R., ed. Publication Manual of the American Psychological Association (6th ed.). American Psychological Association, 2009.
26. Willett, W., Jenny, B., Isenberg, T., et al. Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps. Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems (CHI 2015), (2015). DOI: https://doi.org/10.1145/2702123.2702172
27. Yang, X.-D., Mak, E., McCallum, D., Irani, P., Cao, X., and Izadi, S. LensMouse: augmenting the mouse with an interactive touch display. Proceedings of the 28th international conference on Human factors in computing systems - CHI ’10, ACM Press (2010), 2431. DOI: https://doi.org/10.1145/1753326.1753695
28. Zhao, S., Agrawala, M., and Hinckley, K. Zone and polygon menus: Using Relative Position to Increase the Breadth of Multi-Stroke Marking Menus. Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI ’06, ACM Press (2006), 1077. DOI: https://doi.org/10.1145/1124772.1124933
29. Zhao, S. and Balakrishnan, R. Simple vs. compound mark hierarchical marking menus. Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST ’04, ACM Press (2004), 33. DOI: https://doi.org/10.1145/1029632.1029639