Impact of Handedness and Merging in Command Selection Speed

François Guimbretière, Mary Czerwinski
University of Maryland, Microsoft Research
One Microsoft Way, Redmond, WA USA
francois@cs.umd.edu, marycz@microsoft.com

Abstract

We explore the influence of handedness and the way command selection is integrated with direct manipulation on the speed of command selection. Two empirical studies provide converging evidence that it is the merging of command selection and direct manipulation that benefits performance most, and this is especially effective in a single-handed technique. These findings provide empirical evidence of the importance of merging command selection and direct manipulation in menu design, a factor not often taken into consideration in the past. The findings also yield new insight into the relative importance of handedness and merging in the Toolglass technique.

Keywords: control menus, toolbar, Toolglass, handedness, merging, empirical studies;

1. Introduction

As more computers such as PDAs and tablets are designed to be used without a keyboard, the design of efficient direct manipulation interfaces that do not rely on keyboard shortcuts for tool selection is becoming more important. Over the years, many graphical menu systems have been proposed to improve the efficiency of command selection in direct manipulation interfaces, such as traditional pop-up menus, pie menus [Hopkins 1991], marking menus [Kurtenbach 1993], Toolglass menus [Bier et al. 1993], control menus [Pook et al. 2000] and FlowMenus [Guimbretière and Winograd 2000]. Among these techniques, Toolglass menus, a two-handed technique, probably received the most attention. Kabbash et al. [1994] found that Toolglass menus provided a significant performance advantage over the more traditional toolbar and attributed this gain in performance to its two-handed design. Yet, in his analysis, Kabbash set aside another important aspect of the Toolglass design: because the tool is see-through (semi-transparent), Toolglass users can select a command and proceed with direct manipulation in a single stroke. While not considered consequential at the time, this feature is emerging as a new dimension in the design space. For example, both the control menu and FlowMenu techniques are *single-handed* and merge command selection with direct manipulation.

To understand the benefits of handedness and the merging of command selection and direct manipulation, as well as their possible interaction in command selection design, we designed an experiment crossing these two factors as the two main variables of importance, and assigned the following techniques to each cell (Figure 1):

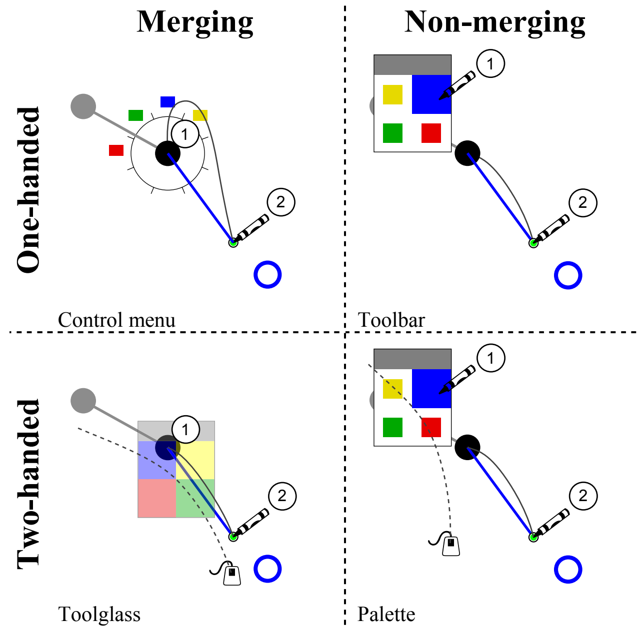

Figure 1 The four menu designs studied in this paper shown in a Handedness × Merging matrix reflecting our experimental design. In each cell, we show how one will use the menu to first select a color (**1**), before creating a colored line between two points (**2**). For each technique, the path of the pen during a connection is shown with a light line. For two-handed techniques (bottom row) the path of the puck is shown with a light dotted line. Toolglass menus use a semi-transparent tool.

- **one-handed, merging**: Control menu, a one-handed technique which lets users select a command and proceed with direct manipulation in the same stroke.
- **two-handed, merging**: Toolglass menus, the technique introduced by Bier et al. [1993]. It was found to be the fastest technique in Kabbash's experiment.
- **one-handed, non-merging**: the familiar toolbar, as commonly seen in any image-oriented or paint software.
- **two-handed, non-merging**: Palette, a two-handed technique used in the original Kabbash experiment. Its mode of operation is similar to the toolbar, but the position of the tool is controlled by the user's non-dominant hand.

Two user studies were carried out. The first user study used a within-subjects design in order to allow all users to use each technique and compare them. The second study used a between-subjects design to prevent any learning or order effects from influencing the data. The results from both studies were consistent with each other and replicated Kabbash's findings that Toolglass menus are faster than both the toolbar and Palette for

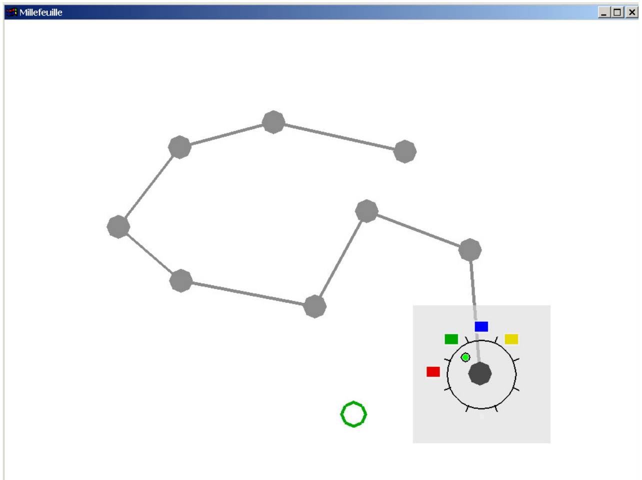

Figure 2 A typical display for our experiment, shown here with the control menu condition. Previously connected dots are shown in gray, while the current dot is shown in black and the target dot is shown open. The 4 possible colors are: red, green, blue, yellow. The gray background has been removed for clarity.

this task. They also showed, however, that the single-handed, merging technique (control menu), is the fastest technique for the task. In contrast to Kabbash's findings our results suggest that **merging**, not two-handedness, is a predominant factor in improving overall speed in the techniques studied. The analysis of our results helps us better understand the relative benefits of handedness and merging in techniques such as Toolglass menus, and highlights the importance of command merging as a design parameter for new interaction design.

2. Related work

Toolglass [Bier et al. 1993] menus merge command selection and direct manipulation by allowing users to use their non-dominant hand to manipulate a semi-transparent tool and their dominant hand to select commands by clicking through the tool before performing direct manipulation tasks (see Figure 1).

The Palette is a two-handed extension of the common toolbar used by Kabbash for an initial experiment examining the design of the Toolglass technique [Kabbash et al. 1994]. Like the Toolglass, the opaque Palette lets users move the tool using their non-dominant hand, but command selection is similar to the common toolbar, in that it is performed sequentially in time, completely separate from a user's direct manipulation task.

The control menu [Pook et al. 2000] is a radial menu technique, conceptually similar to marking menus [Kurtenbach 1993], that does not require users to lift the pen from the screen to select a command. Instead, control menus use a threshold distance as the triggering mechanism to determine menu selection. This feature allows users to proceed directly from command selection to direct manipulation without interruption.

Toolglass menus have been studied extensively (see Leganchuck et al. [1998] for an overview). Of particular interest is the experiment conducted by Kabbash et al. that is described in [Kabbash et al. 1994]. He tested the performance of different menu systems in a simple, colored, connect-the-dots task. Kabbash compared the Toolglass menu to three other techniques, including the conventional toolbar (called *R-tearoff* by Kabbash) and Palette. Kabbash reported significantly better performance in the Toolglass condition. He attributed this result to the fact that the Toolglass technique mirrors a natural pattern, in which users use their non-dominant hand to set a convenient reference for the dominant hand to perform accurate movements (an “asymmetric dependent” assemblage in Guiard's terminology [Guiard 1987]).

Control menus are newer techniques and have not yet been studied extensively. A notable exception is a study by McGuffin et al. [2002], that compared FlowMenu [Guimbretière and Winograd 2000], control menus and FaSTSlider [McGuffin et al. 2002] for assigning parameter values. The study provides qualitative evidence that FlowMenu is more difficult to learn than the other two techniques and that parameter observation and adjustment can be difficult with control menus. The studies presented here will complement this work by providing quantitative data for the case where control menus are used in direct manipulation tasks, while also contributing to the field's knowledge about the importance of command merging and handedness.

3. Methodological approach

To avoid the difficulty of comparing different methods side by side and leveraging the power of statistical tools such Analysis of Variance using a factorial design, we designed our experiment around two important dimensions of the design space of command selection: handedness and merging.

Handedness has long been identified as an important design parameter for input. While many common techniques in use today are single-handed, many researchers (see Leganchuck et al. [1998] for an overview) have argued that two-handed techniques which rely on an “asymmetric dependent” assemblage are better because they leverage our everyday experience [Guiard 1987]. For example, they leverage how we anchor a piece of paper using our non-dominant hand while writing with our dominant hand.

Merging of command selection and direct manipulation (*merging* for short) has been given less attention in the past as it is only recently that several techniques such as FlowMenus and control menus have taken advantage of it. The difference between a non-merging and a merging selection mechanism is best understood by comparing how one draws a line using a marking menu (non-merging) compared to a control menu (merging). If we suppose that both systems use a mouse, then in both systems, users invoke the menu by pressing the mouse button, and then select a command by moving the pen in a given direction. But each technique uses a very different validation technique. To validate their choice and draw the line, marking menu users will have to release the mouse button, click again at the beginning of the line before dragging toward the end of the line and releasing the mouse button. To validate their choice and draw the line, control menu users will first cross the outside menu boundary and proceed directly to the end of the line and release the mouse button there. Because they let users select a command and proceed with direct manipulation in the same stroke, we believe that merging techniques are inherently faster than other techniques. While merging was not identified as an important factor in the design of command selection per se, it was used in several systems in the past, including ToolGlasses, FlowMenus, control menus and more recently the Tracking menu. Yet its importance has not been explored empirically nor has its influence been compared to other factors such as handedness.

Having identified handedness and merging as the two main factors for these experiments, we now have to pick the technique representing each cell of our factorial design (handedness × merging). Given the relative paucity of two handed techniques it was easy to pick the Palette [Kabbash et al. 1994] to represent the (two-handed, non-merging) cell and the Toolglass technique to represent the (two-handed, merging) cell. For the (one-handed, merging) cell, several solutions were possible, including pie menus [Hopkins 1991], the extension of marking menus proposed by Kurtenbach [Kurtenbach and Buxton 1991], control menus, the more complex FlowMenus [Guimbretière and Winograd 2000], or the recently introduced tracking menus [Fitzmaurice et al. 2003]. Unfortunately the design of tracking menus was not available at the time of our study. Therefore, control menus seem to be the best candidate: they are a good representation of the other techniques and both control menus and ToolGlass menus make available a similar number of commands directly accessible. Choosing the technique to represent the (one-handed, non-merging) technique was difficult since so many techniques belong to this category. At the end, we decided on the toolbar for two main reasons: 1) this technique is widely in use today and we felt that it would ground our study to use this technique as a reference point; 2) we wanted to replicate and extend Kabbash's experimental design so that we could compare our results to his as a check on the validity of our findings.

Our final choice was to select a task. One has to find a balance between a task simple enough so it is amenable to measurement yet complex enough so that it is representative of a wide class of everyday usage. Like Kabbash, (and previously by Dillon et al. [1990]) we decided to use a simple "Connect the Dot" task for which each participant is asked to connect a series of color dots on the screen. While a connect-the-dots task might at first seem artificial, it is in fact quite similar to many interactions in today's graphical user interfaces. To make an area selection on a canvas or to create a new object in a CAD program, users often have to first select a tool and then perform a drag between two points on the screen.

We also limited the number of possible selections to 4 colors because we were concerned that a more complex schema (such as 4 colors and 4 shapes) would introduce confounding effects such as the ability of the participant to remember the location of the correct combination. Finally, like Kabbash, we used contiguous segments so that the travel time between one segment to the next did not introduce noise in our measurements.

We will discuss the implication of our choices on our results in the discussion section.

4. Experimental set up

Subjects were asked to connect a series of colored dots on the screen, using a toolbar, Palette, control menu, or Toolglass menu to select a target color. This task provides a good abstraction of common input behavior, is simple enough to learn and is amenable to accurate measurement.

4.1. Equipment Apparatus

For our experiments, we used a Wacom Intuos 18"x12" tablet as the input device. This tablet can simultaneously track a pen and a puck. The tablet was used in absolute mode, and both pen and puck shared the same active surface (i.e., we used a unified area setting [Balakrishnan and Hinckley 1999]). Pilot studies showed that the most comfortable setting for the pen mapped the screen area to an area 216mm by 162mm located 160mm to the right and 127mm above the lower left corner of the tablet active area. To avoid collisions between the pen and the puck, the puck tracking area was offset 76mm to the right and 33mm below the pen tracking area. This setting was picked for the best Toolglass menu performance according to [Balakrishnan and Hinckley 1999]. The areas of the tablet that mapped to the display were lightly outlined on the tablet itself for the pen and the puck, so users had a general notion of where to place the pen or puck relative to the display objects at the beginning of each trial (this was mostly helpful during the practice trials as participants quickly caught on to how to use the tablet, puck and pen). The gain factor between the tablet and the screen was set to 1.33.

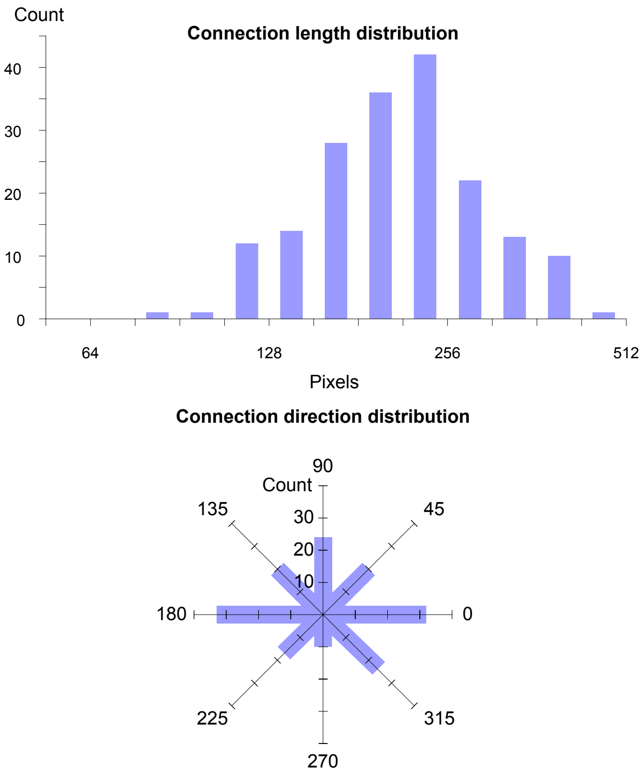

Figure 3 Distance and direction distribution for all connections in our dataset. This dataset was created by picking at random 18 of the 24 patterns used by Kabbash.

The study was run on two identical systems: Compaq EVO Pentium 4 computers with 2.2 GHz CPU and 512 MB RAM were used to drive the experimental software. Wacom Intuos 2 graphics tablets, pucks (4D mouse) and grip pens were used for input during the study. The puck was only used in the 2-handed conditions. The display was a NEC 18" flat panel display running at 1024 x 768 resolution. Custom software logged all the interactions performed by the user. Logged data was only committed to disk between sets to limit timing errors.

4.2. Task and setting

For each condition, participants were presented with 18 sets of 12 points to connect (11 connections per set). We picked at random 18 of the 24 dot patterns used by Kabbash. The connection length and connection direction distributions are shown Figure 3. Patterns were presented in random order.

For each set, the computer presented the series of colored dots one by one. The participant connected the previous dot to the next dot after selecting the correct dot color using the control mechanism. New dots were presented as soon as the participant successfully connected the active dot, and consecutive dots were always of different colors. The "connection time" was computed from the appearance of a new dot to successful completion of the line, including time to correct any errors in picking the color or connecting the dots.

The screen layout is shown in Figure 2. The path created so far is rendered in gray with the exception of the last dot of the path, which is rendered in black. All previous dots in the path are rendered filled. The new target dot is rendered as a circle of the

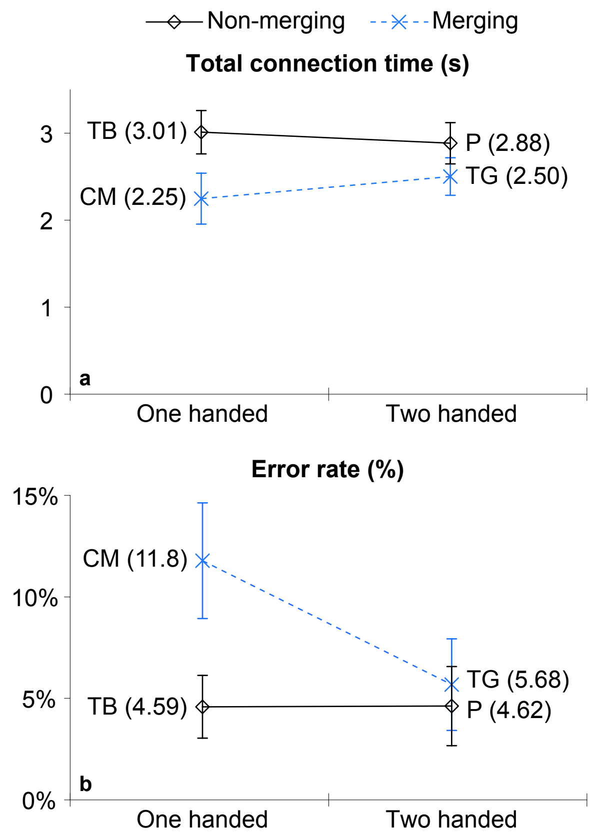

Figure 4 Total connection times (**a**) and error rates (**b**) for Study 1. Total connection time is the average time to perform a connection including time to correct errors. Error bars represent confidence intervals at the 95% level.

requested color. As soon as a line is started, a line of the selected color is shown on the screen as feedback for the rubber band interaction. Each dot radius was 5.4mm and distance between dots varied between 22mm and 123mm.

After each set, participants were presented with their aggregate time for the completed set and their best time so far. If the best time was improved, a rewarding sound was played. The participant could then rest until they were ready to begin the next trial, which they indicated by tapping the pen on a "begin trial" marker. All conditions were run with users interacting with a pen (and a puck for the Toolglass and Palette) on a digital tablet while looking at a monitor (indirect setting).

4.2.1. Toolbar, Palette and Toolglass

In all three conditions, the color tool consisted of 4 buttons, each 16.2mm by 16.2mm, with a header 32.4mm wide and 8.1mm tall at the top. To move the tool, participants could use the header as a handle in the toolbar condition and the puck in the Palette and Toolglass condition. In the Toolglass condition the tool was transparent (40%) so that the dots were visible underneath it.

To perform a connection using the toolbar or Palette, the participant had to first select the correct color by clicking on the appropriate color button and then had to click on the last dot of the path and perform a rubber band interaction to connect this dot to the new colored target dot.

To perform the task using Toolglass menus, participants had to first bring the correct Toolglass color area on top of the last dot in the path using the puck. Then they had to click with the pen on the last dot of the path through the Toolglass, and then proceed directly with the rubber band interaction to connect the new colored dot.

In all these conditions, dots were successfully connected if the pen was lifted from the tablet on top of the target dot.

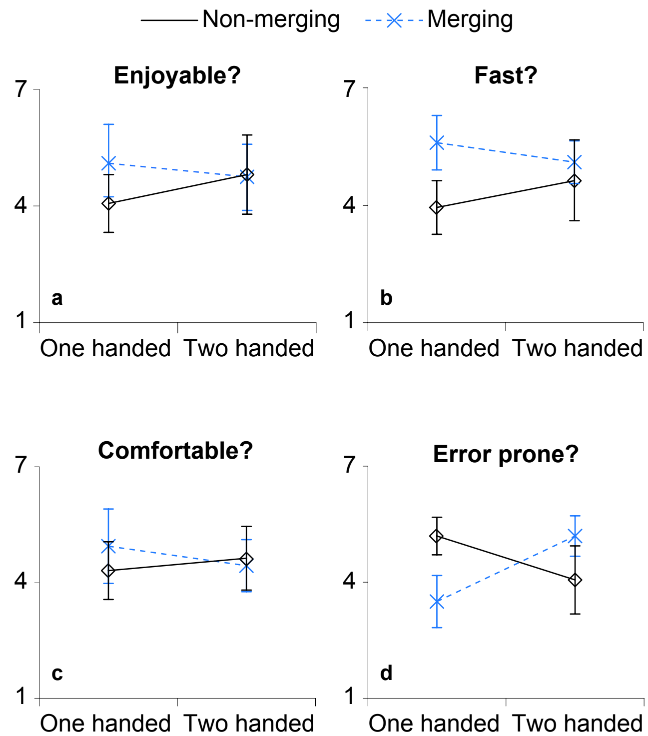

Figure 5 Subjective satisfaction for Study 1. Error bars represent confidence intervals at the 95% level.

4.2.2. Control menu

In the control menu conditions, the radius of the menu was 15.5mm. To perform the task using a control menu, participants had to first invoke the menu on top of the last dot of the path by pressing the pen's command button while pointing to the dot (the command button could be released as soon as the menu appeared). Then they had to select a color by leaving the rest area through the appropriate color's octant. Finally, participants could proceed directly with the rubber band interaction to connect to the new colored target dot. Dots were successfully connected if the pen was lifted from the tablet on top of the target dot or if the command button was pressed while on top of the target dot to start issuing the next command as described in [Guimbretière and Winograd 2000].

For this condition the color location was selected so that it reflected the use of the system in a direct on-screen configuration. In such a configuration the user right-hand masks most of the menu choices in the South-East quadrant, so we group our menu selections in the North-West quadrant. Overall, this choice meant that the dot to be connected was in the same octant as the dot color in 15% of the connections, one octant to the left or to the right in 30% of the connections, and required course reversal in 50% of the connections.

5. Study 1

5.1. Participants

16 participants (9 female), ranging in age from 20 to 50 years old, were recruited for the study. Participants were intermediate to advanced MS Office users, as determined by a well validated internal screening tool. In addition, all participants were required to be right-handed, and have normal or corrected-to-normal vision. Only 3 participants rated themselves as familiar with pen computing, and none rated themselves as expert with either 2-handed control or with radial menus.

5.2. Design

A Handedness (one-hand vs. two-hand) × Task Integration (non-merging vs. merging) within subjects design was chosen for the experiment. Participants were run in pairs, with each pair experiencing the same order of menu techniques during the session. To counterbalance the technique order, and yet limit skill transfer between the same handedness conditions, we picked 8 out of the 24 possible sequences that alternate one-handed and two handed techniques for each participant. Each technique appears at a given order position the same number of times across participants, and each technique precedes or follows a given technique the same number of times as well.

Each menu technique was practiced with 5 patterns picked at random from our set of 18 patterns before beginning the experimental trials for that technique. After the 5 practice trials, the 18 experimental trials were carried out. After the block of 18 trials was completed, a satisfaction questionnaire was filled out. After that, the experimenter came in and demonstrated the next menu technique, allowed the participants to practice the next technique for 5 trials, and the process continued until all four menu techniques had been completed.

At the end of the session the experimenter asked participants for their overall preferences and debriefed them of the study's purpose and main hypotheses. Participants received a software gratuity for their participation. The entire session lasted two hours.

5.3. Results

To correct for strong distributional skewing in reaction time data, and ascertain that assumptions required for ANOVA were respected, timing data were transformed using the Log transform before carrying out any statistical analyses. Data points 3 standard deviations away from the means (per connection × technique) were removed before performing data analysis. This resulted in removing less than 0.6% of all data points prior to analysis.

5.3.1. Task Times

A 2 Handedness (one-hand vs. two-hand) × 2 Task Integration (non-merging vs. merging) Repeated Measures Analysis of Variance (RM-ANOVA) was performed on the average log trial time data. A main effect of merging, $F(1,15)=14.05$, $p<.001$, was observed, but there was no significant effect of handedness, $F(1,15)=4.35$, $p=.054$. The interaction between merging and handedness was, however, significant, $F(1,15)=147.63$, $p<.001$. Merging techniques were reliably faster (2.37 s., on average vs 2.95 s., respectively). In addition, the one-handed, merging technique (control menu) was fastest overall (2.25 s). These results can be seen in Figure 5 a.

5.3.2. Errors

Finally, we analyzed the number of errors occurring in each technique. A 2 × 2 RM-ANOVA following the same pattern as above, was used to reveal significant main effects of handedness, $F(1,15)=10.9$, $p=.004$, merging of commands, $F(1,15)=10.01$, $p=.006$, and the interaction was also significant, $F(1,15)=16.5$, $p<.001$ (Figure 5 b). The 1-handed, merging technique (control menu) was more error-prone (12%) than the other techniques, so there was an indication of a speed-accuracy tradeoff in the data for this particular technique.

5.3.3. User Satisfaction

Ratings were provided using a scale of 1 to 7 with lower values indicating less satisfaction. Handedness (one-hand vs. two-hand) × Task Integration (non-merging vs. merging) × 4 (question) RM-ANOVA was carried out on the average subjective ratings. No significant main effects or interactions were observed in the user satisfaction data (see Figure 5 a-d). In terms of overall preference, 9 preferred the control menu (significant at the .05 level by binomial test), 3 chose the Palette and 4 chose the Toolglass menu technique. Our participants did not consider the higher errors encountered with the control menu to offset its benefits in terms of speed, as evidenced by their ratings and comments.

5.3.4. Analysis

This first within subjects study replicated Kabbash's findings for the toolbar, Palette and Toolglass. But our full factorial design allowed us to isolate the effects of command merging and handedness, and reveal a clear benefit for a single-handed, merging menu technique (i.e., control menu) in terms of significantly reducing overall task times for the connect-the-dots task. In addition, participants significantly preferred control menus over the other techniques. This study suggests that it is the merging of commands, not the use of two hands as has been traditionally thought, that improves performance for techniques like Toolglass menus for this class of tasks, despite an increase in errors.

To look for possible asymmetrical transfer effects, we ran an analysis checking for the effects of training by order of presentation of the menu types to the users. A 2 (handedness) × 2 (task integration) x 8 (order) RM-ANOVA did not reveal a main effect of order. However, a significant 3-way interaction was observed, $F(7,8)=5.15$, $p=.01$, despite our efforts to control for them. In particular it appeared that Palette performance might be influenced by a prior exposure to Toolglass menus. Despite the large main effect observed, showing the beneficial influence of merging in a single-handed menu technique, we were concerned about the influence that order had on the data and decided to attempt to replicate our finding with a new, between subjects design.

6. Study 2

6.1. Participants

48 participants, screened using the exact same criteria as in Study 1, participated in Study 2. None of the participants in this second study had participated in Study 1. Exactly half the sample was female.

6.2. Design

The design was a between subjects design, with participants run in pairs, so each pair of participants experienced only one of the menu techniques and then filled out the user satisfaction questionnaire. Session length was typically under a half hour.

6.3. Results

Data points 3 standard deviations away from the means (per connection × technique) were removed before performing data

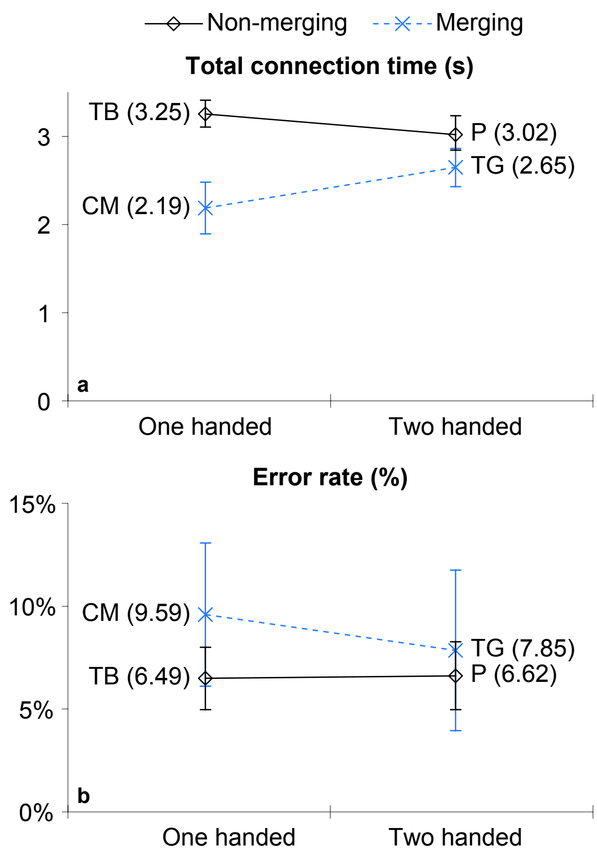

Figure 6 Total connection times (**a**) and error rates (**b**) for Study 2. Total connection time is the average time to perform a connection including time to correct errors. Error bars represent confidence intervals at the 95% level.

analysis. Less than 0.4% of all data points were removed from the analysis. All other aspects of the study were identical to Study 1.

6.3.1. Task Times

A Handedness (one-hand vs. two-hand) × Task Integration (non-merging vs. merging) between subjects ANOVA was performed on the average Log trial time data, as in Study 1. We again observed a significant main effect of merging, $F(1,44)=93.6$, $p<.001$, as well as a significant effect of 1 vs 2 hands, $F(1,44)=4.8$, $p=.03$. As in Study 1, the interaction was also significant, $F(1,44)=25.4$, $p<.001$. Merging benefited performance significantly (2.42 s. vs 3.13 s., on average, respectively). A benefit for single-handed interaction was observed in the merged condition (i.e., control menu), as can be seen in Figure 6 a (and Figure 9), which accounts for the interaction.

6.3.2. Errors

A Handedness (one-hand vs. two-hand) × Task Integration (non-merging vs. merging) between subjects ANOVA was carried out on the error data, but no main effects or interactions were observed at the $p < .05$ level (Figure 6 b), so a speed-accuracy tradeoff is not descriptive of the faster techniques in this study. We are not sure why we observed evidence of a speed-accuracy tradeoff in Study 1, and will examine this further in future work.

6.3.3. User Satisfaction

As in Study 1, subjective ratings were provided using a scale of 1 to 7 with lower values indicating less satisfaction. A Handedness

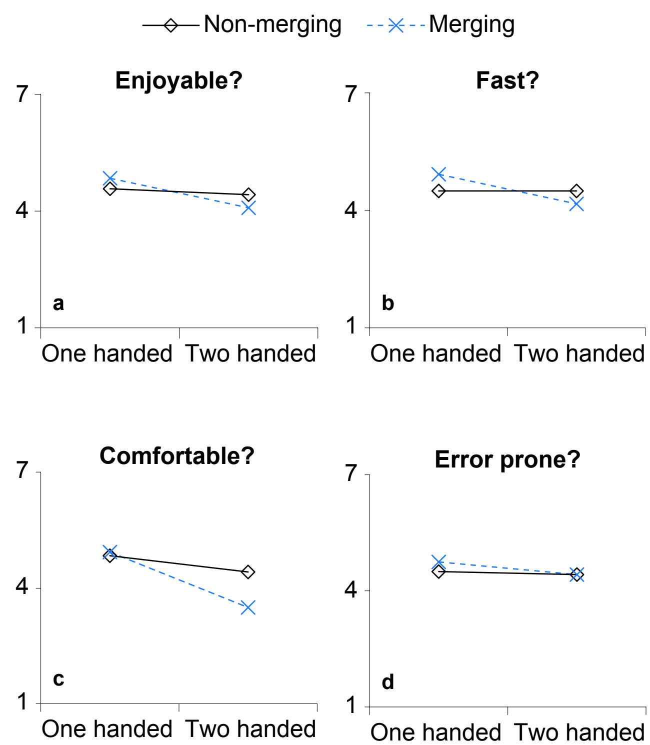

Figure 7 Subjective satisfaction for Study 2. Error bars represent confidence intervals at the 95% level.

(one-hand vs. two-hand) × Task Integration (non-merging vs. merging) × 4 (question) RM-ANOVA was carried out on the user satisfaction data, with only the question variable within subjects. No significant main effects or interactions were observed. All of the average satisfaction ratings are shown in Figure 7 a-d.

6.3.4. Detailed Analysis

As in study 1, our results replicated those reported by Kabbash for the three techniques shared by both designs: Palettes provided a small advantage over a standard toolbar, while the Toolglass technique provided a significant advantage over both techniques. Like Kabbash, we examined the tool position with respect to the starting point for each tool based technique (Figure 8 left) and found that the simultaneous use of both hands significantly reduced the distance between the center of the tool and the starting point ($F(2,6444) = 3835.1$, $p < .0001$). On average, participants placed the Toolglass 85 pixels away from the initiating dot, the Palette 155 pixels away and Toolbar 318 pixels away. Like Kabbash, we also found a difference in drawing time between techniques. A 2 (Handedness) × 2 (Task Integration) between subjects ANOVA on the drawing time data revealed that drawing time was longer for merging techniques than for non-merging techniques ($F(1,44)=14.1$, $p=.001$), and the interaction between handedness and merging was borderline significant, $F(1,44)=3.9$, $p=.056$. It is of course difficult to analyze the real drawing time for the control menu condition, given that "drawing" does not start on the starting dot but at the periphery of the menu (Figure 1). So we will only focus on the average differences between Toolglass (1.20 s.), toolbar (1.01 s.), and Palette (.98 s.) since they correspond to a very similar movement. Figure 8 (middle), shows that two-handed merging techniques were slower than non-merging techniques during the drawing phase ($p < .001$ in both cases, using the Bonferonni correction).

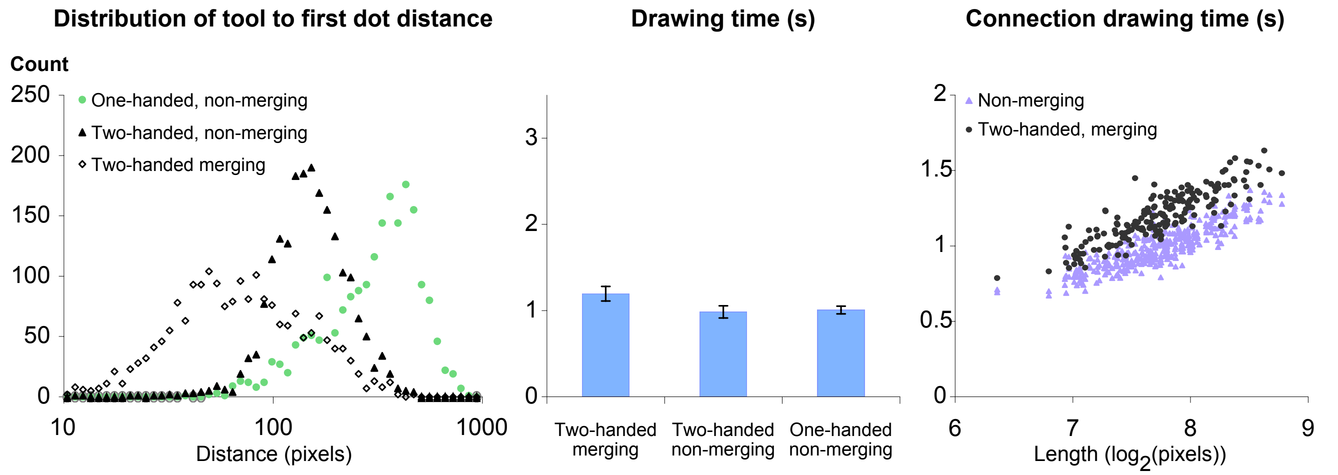

Figure 8 How two-handedness influences the drawing task**. Left**: Distribution of the distance between the tool and the first connection dot for the three tool based techniques. The use of the second hand helps to reduce the distance between the tool and the first dot of a connection. **Middle**: Drawing time for the three tool based techniques showing that ToolGlass (two-handed merging) is slower than other techniques. **Right**: Relationship between the connection length and the average drawing time for the 180 connections of our set. Connections performed with ToolGlass form a different cluster and seem more "difficult". All data from Study 2, using only error free connections.

While Kabbash concluded that there was no evidence that an increase in left hand use slows down drawing, we disagree. We analyzed the average connection time for all users against the connection length for our 180 connections. The result is plotted in Figure 8 on the right. It shows that connections made with Toolglass menus were part of a different cluster than connections made with either toolbar or Palette menus ($F(2,537) = 99.8$, $p < .001$), with Toolglass connections being more "difficult" to handle (i.e., taking longer to make). From our logs we distinguished that the most common way users carried out the task with the Toolglass menu was by moving the Toolglass at the same time as a connection was being made. Thus, it seems that moving the Toolglass with the left hand slowed down the tracing part of the task, probably because the user needed to attend to the two tasks at the same time.

As in Study 1, our factorial design lets us understand the role of merging, a hidden factor in Kabbash' study. Study 2 replicated the main findings of Study 1, in that a significant benefit was once again obtained for merging techniques, but this time **without** a speed-accuracy tradeoff. We also saw a significant interaction, which can be best explained as the control menu (1 handed and with merging) driving the benefits of single-handed interaction. In other words, both studies have provided converging evidence that it is not primarily the 2-handed interaction that improves performance in connect-the-dot kinds of computer tasks, but the merging of command selection and direct manipulation within the menu technique. Since this second study was a between subjects design, we could not ask participants which technique they preferred overall.

7. Discussion

The results presented above highlight the importance of techniques that can fluidly mix command selection and direct manipulation. As explained above, such techniques present a significant advantage over the standard toolbar solution in an analog of commonly performed tasks.

7.1. Two-handedness versus merging

As our analysis shows, merging is the most significant factor in improving users' menu selection performance. By merging command selection and direct manipulation, a command selection system alleviates the need for the user to go back and forth between the tool and the task locus of attention. As expected, this results in a significant time savings for both the Toolglass and the control menu. Yet both systems reach this goal in a very different way. By using a two-handed setting, the Toolglass technique still requires the user to accurately position the tool; hence its speed is limited by Fitts' law for area cursors [Kabbash and Buxton 1995]. In contrast, control menus, like marking menus, rely on a gesture for command selection. This gesture corresponds to a low difficulty steering task [Accot and Zhai 1997] that might not require direct visual feedback for experienced users. While the selection mechanism increases the average total drawing distance (and therefore the drawing time), the difference is small (28% in our case) and overall control menu design was the fastest in our studies.

We can now present our overall analysis of our results as shown in Figure 9. We decomposed the steps required by each technique. The Palette technique presents the same interaction structure as the traditional toolbar (two aiming tasks for color selection and one for drawing) but benefits from two-handedness by allowing users to reduce the distance between the tool and the target. The Toolglass technique benefits from two-handedness by letting users reduce the distance between the tool and targets even further. Its performance benefits even more from merging command selection and direct manipulation by requiring only one target acquisition task per color selection. Some of the advantage of this technique is lost by the added difficulty of moving the tool and the pen at the same time while drawing. Control menus, by avoiding the use of a tool altogether, reap the full benefit of merging the command selection and direct manipulation with an overall command structure composed by one gesture and one aiming task.

One surprising aspect of our results was the interaction between merging and handedness. We hypothesize that this might be caused in part by the choice of the toolbar as the technique to represent the (one-handed, non-merging) cell. Because users seldom adjusted the position of the toolbar, a significant time was spent going back and forth between the task area and the

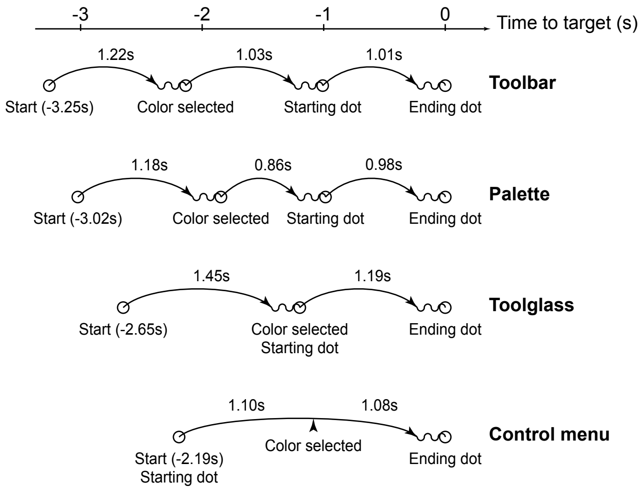

Figure 9 Command structure of the different techniques studied here showing the average time for each phase of the task. By limiting the number of aiming task (represented here as wavy lines) merging provides a significant speed advantage. All data from Study 2, using only error free connections, and assuming the pen is over the starting point at the beginning of the task.

command selection tool. In retrospect other techniques such as a pie menu, or marking menu might have been a better choice for this cell. Yet for this initial round of exploration our choice made sense as it helped us compare our experimental approach to Kabbash's and because the toolbar is so commonly used in existing software applications. Given other corroborating evidence of the importance of merging in the command selection process, we believe that the presence of this interaction does not weaken our main conclusion that merging is a key factor in the design of efficient command selection mechanisms.

7.2. Influence of training

Given the short exposure time each participant had to each technique (with the exception of the toolbar) one might wonder how our results might apply to expert users. While we did not perform a longitudinal study, we looked at how participants' performance evolved over trials during our first study. As shown in Figure 10, the relative position of each technique remains similar as users improve. From this observation, we believe that our results will hold over even longer periods of usage.

7.3. Interaction design considerations

Our work suggests that control menus might be faster than Toolglass menus for tasks like area selection and vector drawing in CAD and illustration programs. Nevertheless, speed is not the only criterion for designing a user interface interaction mechanism. Therefore, it is important to understand some of the fundamental differences between these techniques.

7.3.1. Versatility

Control menus and the Toolglass menus might be functionally equivalent in many situations, such as for drawing geometric shapes or for selecting and transforming objects. However, if the application requires freeform drawing, then Toolglass menus have a distinctive advantage because of the Toolglass menu's "see-through" metaphor that allows the user to start the interaction at the point where the command was invoked. Control menus do not provide this flexibility because they require the user to cross a specific boundary away from the point where the menu was

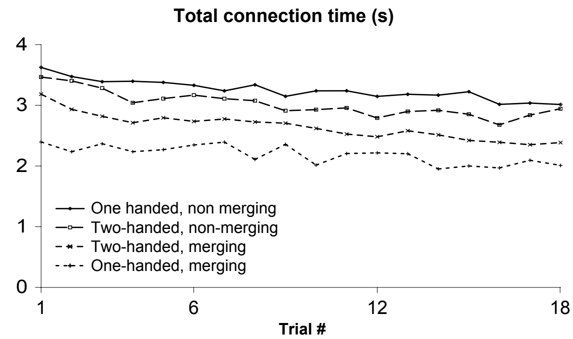

Figure 10 Learning effect for Study 2.

called. Note that pie menus [Hopkins 1991], the extension of marking menus discussed in [Kurtenbach and Buxton 1991] and FlowMenus [Guimbretière and Winograd 2000] would face a similar limitation since the selection stroke has to be part of the drawing, or the drawing has to be started at some point away from the initial location where the menu was called.

7.3.2. Amortizing command cost

While it is true that the basic operation in our "connect the dots" task resembles a variety of different direct manipulation actions, as pointed by Mackay [Mackay 2002], users often perform similar operations in succession. It is interesting to see how the cost of performing a connection evolves if several points of the same color are presented in succession. In that case the amortized connection time (CT) can be expressed as a function of the number n of successive points with the same color, the average color selection time (CST) for a technique and the average drawing time (DT) as:

$$CT = CST + DT \quad \text{(for merging techniques) (1)}$$

$$CT = \frac{CST}{n} + DT \quad \text{(for non-merging techniques) (2)}$$

Equating (1) and (2) we can compute how many operations per tool selection might be enough to make the toolbar technique faster than any merging technique. From our data (using error free connections), we found that 1.4 operations will be necessary for Toolglass and 2.3 operations for control menus. As shown by Mackay's study of CPN2000 [Mackay 2002], some tasks such as making a copy of a Petri net naturally lead to such a high amortization (3.3 commands per command selection for the toolbar). This might help to explain why the toolbar is still such a popular menu technique.

These results depend heavily on the task. Like previously published research, we used a task with contiguous lines. This reduced the noise introduced by the travel time from one segment to the next, but this is probably not the most common pattern of use. It is certainly difficult to understand the influence of travel time since it might overlap with "thinking time" during which participants mentally prepare for the next command selection. At this point, we feel that we are lacking data to predict this effect. Yet we hope that our experience will encourage future experimenters to include this variable in their designs.

7.3.3. Scalability

Finally, many applications have far more than four possible commands. All one-handed techniques studied here as well as Toolglass can easily be extended to accommodate a larger set of commands (see for example the T3 system [Kurtenbach et al. 1997]). The analysis of such systems is made more difficult when dealing with large command sets because of possible confounding effects, such as having to remember how to access any particular command. In this respect we believe that our choice to set aside this issue in these experiments was sound and led to easier interpretation of our results. This is not to say that this issue is not important. On the contrary, we believe that the paucity of empirical results in this area makes it difficult for designers to make informed choices. Yet this difficult issue needs to be analyzed with a study specifically designed for this purpose.

8. Future work

We would like to replicate our findings using other techniques to represent the different cells in our design. In particular we would like to run a new study to compare the new Tracking menu with other techniques. This one-handed technique has a selection pattern which is very similar to the ToolGlass menu, making it an interesting candidate for the (one-handed, merging) cell.

We would like to broaden our understanding of how one-handed techniques can improve other novel operations. Recent results by Bourgeois [Bourgeois and Guiard 2002] show the benefits of two-handed interaction for zooming and panning. Using techniques such as speed-dependent zooming [Igarashi and Hinckley 2000], we would like to see if similar performance can be obtained using one hand so that the non-dominant hand can be used for other purposes.

We would also like to explore the new design avenues suggested by our results. We intend to study a new style of two-handed interfaces where, instead of using the non-dominant hand to select commands as in the Toolglass menu, one might use it to orient the drawing canvas or masking tool as in T3 [Kurtenbach et al. 1997]. Guiard has advocated this setting [Guiard 1987], and our results suggest that this approach might deliver significantly better performance.

9. Conclusion

The results presented in this paper provide a better understanding of the relative role of handedness and merging in command selection task speed. Our analysis from two separate studies shows that the impact of merging on user performance has not been fully appreciated by previous studies of command techniques, including both one-handed techniques and two-handed techniques such as Toolglass menus [Kabbash et al. 1994].

Our results also suggest that two-handed command selection techniques seemingly suffer from an inherent limitation, since they force users to split attention between movement of the tool in the non-dominant hand and movement of the pen in the dominant hand. This suggests that one-handed merging techniques may enjoy a performance advantage that a two-handed technique cannot emulate, at least in the context of the connect-the-dots experimental task.

Far from questioning the advantages of two-handed techniques, our results help to understand how handedness and merging interplay in the case of Toolglass menus. This might help with the design of better future one- and two-handed command selection techniques.

10. Acknowledgements

TBD.

References

Accot, J. and S. Zhai (1997). Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. *Proceedings of CHI'97*, 295 - 302.

Balakrishnan, R. and K. Hinckley (1999). The role of kinesthetic reference frames in two-handed input performance. *Proceedings of UIST'99*, 171 - 178.

Bier, E. A., M. C. Stone, K. Pier, W. Buxton and T. D. DeRose (1993). Toolglass and magic lenses: the see-through interface. *Proceedings of SigGraph'93*, 73 - 80.

Bourgeois, F. and Y. Guiard (2002). Multiscale pointing: facilitating pan-zoom coordination. *Proceedings of CHI '02, extended abstracts*, 758 - 759.

Dillon, R. F., J. D. Edey and J. W. Tombaugh (1990). Measuring the true cost of command selection: techniques and results. *Proceedings of CHI'90*, 19 - 26.

Fitzmaurice, G., A. Khan, R. Pieké, B. Buxton and G. Kurtenbach (2003). Tracking Menus. *Proceedings of UIST'03*, 71 - 80.

Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behaviour* **19**(4): 486 - 517.

Guimbretière, F. and T. Winograd (2000). FlowMenu: combining command, text, and data entry. *Proceedings of UIST'00*, 213 - 216.

Hopkins, D. (1991). The Design and Implementation of Pie-Menus. *Dr. Dobb's Journal* **16**(12): 16 - 26.

Igarashi, T. and K. Hinckley (2000). Speed-dependent automatic zooming for browsing large documents. *Proceedings of UIST'00*, 139 - 148.

Kabbash, P., W. Buxton and A. Sellen (1994). Two-handed input in a compound task. *Proceedings of CHI'94*, 417 - 423.

Kabbash, P. and W. A. S. Buxton (1995). The “prince” technique: Fitts' law and selection using area cursors. *Proceedings of CHI'95*, 273 - 279.

Kurtenbach, G. (1993). *The design and Evaluation of Marking Menus*. PhD thesis, University of Toronto.

Kurtenbach, G. and W. Buxton (1991). Issues in combining marking and direct manipulation techniques. *Proceedings of UIST'91*, 137 - 144.

Kurtenbach, G., G. Fitzmaurice, T. Baudel and B. Buxton (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *Proceedings of CHI'97*, 35 - 42.

Leganchuk, A., S. Zhai and W. Buxton (1998). Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study. *Transactions on Human-Computer Interaction* **5**(4): 326-359.

Mackay, W. E. (2002). Which Interaction Technique Works When? Floating Palettes, Marking Menus and Toolglasses support different task strategies. *Proceedings of AVI'02.*

McGuffin, M., N. Burtnyk and G. Kurtenbach (2002). FaST Sliders: Integrating Marking Menus and the Adjustment of Continuous Values. *Proceedings of GI'02*, 35 - 42.

Pook, S., E. Lecolinet, G. Vaysseix and E. Barillot (2000). Control menus: excecution and control in a single interactor. *Proceedings of CHI'00 Extended Abstracts*, 263 - 264.