Benefits of Merging Command Selection and Direct Manipulation

FRANCOIS GUIMBRETIERE
University of Maryland
and
ANDREW MARTIN AND TERRY WINOGRAD
Stanford University

Toolglass [Bier et al. 1993], demonstrated a two-handed command selection technique that combined command selection and direct manipulation. The experiments with Toolglass did not show the relative importance of two possible factors in its improved performance: 1) the use of two hands and 2) the merging of command selection and direct manipulation.

To examine this question, we conducted a study comparing the relative benefits of three command selection techniques that merge command selection and direct manipulation: one two-handed technique, Toolglass, and two one-handed techniques, control menus [Pook et al., 2000] and FlowMenu [Guimbretière and Winograd, 2000]. Our results show that control menus and FlowMenu are significantly faster than Toolglass. Further analysis suggests that the merging of command selection and direct manipulation is the most important factor in the performance of all three techniques.

Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation/methodology; interaction styles (e.g., commands, menus, forms, direct manipulation)*

General Terms: Experimentation, Human Factors, Measurement, Performance

Additional Key Words and Phrases: Control menu, FlowMenu, Toolglass, two-handed interfaces, merging command selection and direct manipulation

---

1. INTRODUCTION

Toolglass, a two-handed interaction technique introduced by Bier et al. [1993], is the only command selection technique merging command selection and direct manipulation that has been studied empirically. Kabbash et al. [1994] found that Toolglass provides a significant performance advantage over the more traditional tool palette for a simple color painting "connect the dots" task.

Because there has not been an equivalent study for a one-handed command selection technique that merges command selection and direct manipulation, it has been difficult to

---

Authors' addresses: François Guimbretière Human Computer Interaction Laboratory, Department of Computer Science, University of Maryland, A. V. Williams Building, College Park, Maryland 20742; email: francois@cs.umd.edu; Andrew Martin, andy.martin@stanfordalumni.org; Terry Winograd, Department of Computer Science, Stanford University, William Gates Building, Stanford, California 94305; email: winograd@cs.stanford.edu.
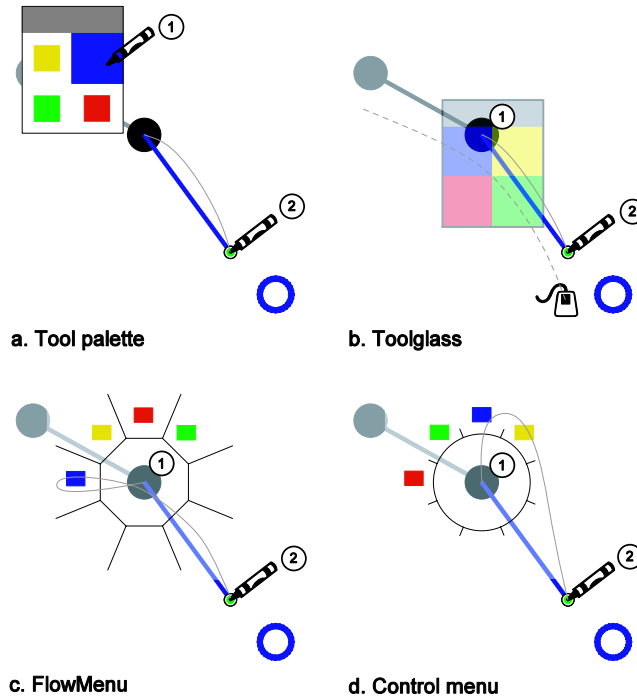
Fig. 1. The four command mechanisms used in this study. Each command mechanism is used to select a color before connecting two dots on the screen. A tool palette requires the user to first click on the appropriate color and then connect the dots. With Toolglass, a two handed technique, users move the semi-transparent Toolglass with a puck and select the color by starting the connection *through* the correct color. Both control menu and FlowMenu are invoked by pressing the pen command button while clicking on the starting dot and then performing a gesture to select the color (see text). The path of the pen on the screen during a connection is shown with a light line. The path of the puck is shown with a light dotted line.

understand the relative importance of two factors in Toolglass's improved performance: 1) the use of two hands and 2) the merging of command selection and direct manipulation.

To understand the relative performance benefits resulting from each of these two factors, we adapted the experiment of Kabbash et al. [1994], using a simple color painting "connect the dots" task to compare the commonly used tool palette, Toolglass, and two recently introduced one-handed techniques that merge command selection and direct manipulation: control menus [Pook et al., 2000] and FlowMenu [Guimbretière and Winograd, 2000] (Figure 1). While we could have used other one-handed techniques that merge command selection and direct manipulation, such as pie menus [Hopkins, 1991] or the extension of marking menus proposed by Kurtenbach and Buxton [1991], we chose control menus and FlowMenu because they are the closest to Toolglass's style of

interaction. Furthermore we judged that these two techniques are different enough to provide good coverage of the current design space, and their inclusion would ensure that our results were not specific to just one technique.

Our experimental results replicated Kabbash's finding that Toolglass is faster than a tool palette for this task and showed that both control menus and FlowMenu are faster than Toolglass. These results seem to imply that the key factor in Toolglass's performance is its merging of command selection and direct manipulation, not its two-handedness. The analysis of our results provides a better understanding of mechanisms at play while using two-handed interactions for command selection. Based on our results, we also propose new directions for the design of efficient command mechanisms.

2. RELATED WORK

Toolglass [Bier et al., 1993] was one of the first interaction mechanisms to merge command selection and direct manipulation. With Toolglass, the user uses his or her non-dominant hand to manipulate a translucent tool palette and his or her dominant hand to select commands and perform direct manipulation tasks (Figure 1a.) To perform an action such as creating a colored line, the user first brings the desired color line tool area on top of the starting point and then issues the command by clicking onto the canvas through the Toolglass. The user can then proceed directly with a direct manipulation action on the line.

Both FlowMenu and Control menu are one-handed techniques merging command selection and direct manipulation. Control menu [Pook et al., 2000] is radial menu with 8 octants. Upon activation, the menu pops up, and the user can make a selection by moving from the center toward one of the menu items (Figure 1d.). When the pen reaches a specified threshold radius, the command is issued and direct manipulation can proceed immediately. Conceptually similar to marking menus [Kurtenbach, 1993], control menus use crossing a threshold instead of lifting the pen as a command selection mechanism. This distinction lets the user proceed directly from command selection to direct manipulation without interruption. Like marking menus, control menus can be cascaded to provide access to more than eight commands.

FlowMenu [Guimbretière and Winograd, 2000] is a radial menu with 8 octants and a central rest area (Figure 1c). Upon invocation, the menu pops up centered on the pen. The user selects a top-level menu item by leaving the central area and entering one of the octants. When he or she leaves the rest area, sub-menus for this menu octant appear. Moving the pen to the desired sub-menu octant and reentering the rest area from that

octant triggers a menu selection and direct manipulation can proceed immediately. Although its principle is similar to control menus, FlowMenu provides additional features such as textual entry and knob interaction (see [Guimbretière and Winograd, 2000] for further details).

Of these three techniques, only Toolglass has been studied extensively. Of particular interest is the experiment conducted by Kabbash et al. [1994], which tested the performance of different affordances in a simple colored "connect-the-dots" task. Kabbash compared Toolglass to three other techniques including the conventional tool palette (called *R-tearoff* by Kabbash) and reported significantly better performance in the Toolglass condition, attributing the performance gain to the use of an "asymmetric dependent" [Guiard, 1987] two-handed technique.

FlowMenu and 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, control menus, and FaSTSlider for assigning parameter values. In that study, users were asked to observe and adjust the numerical value of eight parameters using FlowMenu, control menus, and FaSTSlider. 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.

3. HYPOTHESIS

To explore how the merging of command selection and direct manipulation influences the performance of interaction techniques, we decided to compare the performance of the following interaction techniques:

- Tool palette represents the current *status quo*. It uses a sequential command assemblage [Kabbash et al., 1994] and neither merges command selection and direct manipulation nor relies on two-handed interaction;

- Toolglass uses an asymmetric dependant command assemblage [Kabbash et al., 1994] and merges command selection and direct manipulation by using two hands to perform the task;

- FlowMenu and control menus use a sequential command assemblage and merge command selection and direct manipulation using one hand.

The results of Kabbash et al. [1994] suggest that an asymmetric dependant assemblage is faster than a sequential command assemblage therefore we expected Toolglass to be faster than Tool palette, FlowMenu, and control menus.
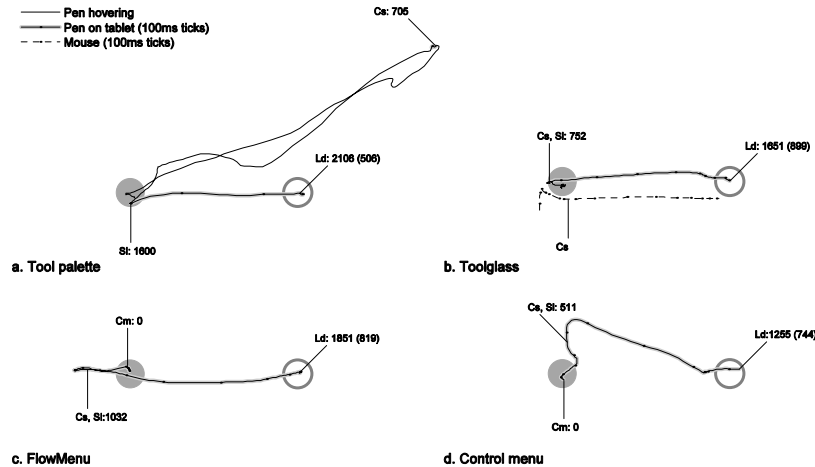
Fig. 2. A typical trace for each technique used in this experiment. Reference points show the time (in milliseconds) since the beginning of the connection and are labeled as follows: Cm : Calling menu; Cs: Color selected; Sl: Started line; Ld: Line done. For Ld, the time in parentheses is the drawing time.

Within the sequential assemblage group, we expect FlowMenu and control menus to be faster than tool palette since they merge command selection and direct manipulation. In doing so, they save users the time it takes to go back and forth between the current locus of attention and the color selection tool.

Predicting the difference between FlowMenu and control menus is somewhat more difficult given the lack of empirical data for both systems. Looking at typical traces (as shown in Figure 2) the most striking difference is the complexity of the overall gesture, with the control menu paths being significantly simpler. Based on this observation, we postulate that control menus should be faster than FlowMenu.

Using the previous analysis, we can now state the hypotheses for our experiment:

Hypothesis 1: The tool palette, a one-handed technique which does not merge command selection and direct manipulation, will be the slowest technique,

Hypothesis 2: Among one-handed mechanisms merging command selection and direct manipulation, control menus, which have a simpler command selection mechanism, will be faster than FlowMenu.

Hypothesis 3: Among techniques merging command selection and direct manipulation, Toolglass, an asymmetric dependent command mechanism, will be faster than both control menus and FlowMenu, which are both sequential command mechanisms.

5

Fig. 3. The experimental setting, consisting of our display, the Wacom tablet, its pen, and its puck. All conditions used this setting.

4. EXPERIMENT

Subjects were asked to connect a series of colored dots on the screen, using a tool palette, a control menu, FlowMenu, or Toolglass to select the color. Like Kabbash et al. [1994] (and before by Dillon et al. [1990]), we use the "connect-the-dots" task because it offers a good abstraction of many typical interactions in today's interfaces (such as making an area selection on a canvas or creating a new object in a CAD program) but is simple enough to be amenable to accurate measurement.

We used a within-subjects design, asking each subject to connect a series of colored dots using all four techniques in turn. The independent variable was the method used to connect dots. The dependent variable was the total task completion time per connection, a good representation of the overall performance since this measure factors in not only the time taken to perform the task but also the time taken to correct errors.

4.1 Subjects

Twelve right-handed, non-colorblind subjects (7 men and 5 women) were recruited from a young adult population (18 to 36 years of age). All subjects had little or no experience using a pen interface other than on a PDA. In addition, subjects had little or no knowledge of control menus, FlowMenu, or Toolglass.

4.2 Experimental Apparatus

We used the setup shown in Figure 3. Interactions were performed on a Wacom Intuos 12"x12" tablet. This tablet can simultaneously track a pen and a puck. The tablet was
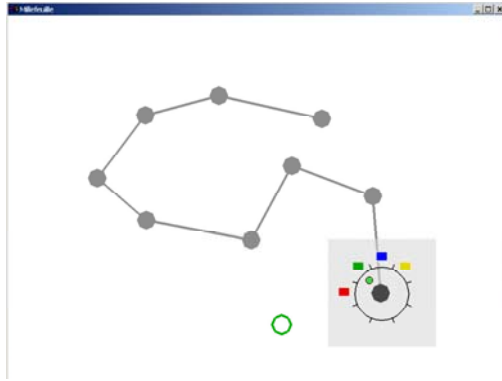
Fig. 4. A typical display for our experiment, shown here with the control menu condition. Previously connected dots are shown in gray, while the current dot is shown in black and the target dot is shown open. The 4 possible colors are: red, green, blue, and yellow. The gray background has been removed for clarity.

used in absolute mode, and both pen and puck shared the same active surface (unified area setting, using the nomenclature of Balakrishnan and Hinckley [1999]). Pilot studies showed that the most comfortable setting mapped the screen area to an area 9" by 6 3/4" starting at 3 1/2" below and 3" to the right of the top left corner of the tablet active area. The gain factor between the tablet and the screen was set to 1.33. Finally, to avoid collisions between the pen and the puck, the puck tracking was offset by 1 1/4". This setting was picked for best Toolglass performance according to Balakrishnan and Hinckley [1999].

The experimental software was running on a Dell Precision workstation 610 MT with a single Pentium III (550 MHz) and 256MB of memory, using an Nvidia GeForce2 as a display card. The workstation was connected to a Dell UltraScan 1000HS series 17" monitor with a visible area of 15" diagonal, running at a resolution of 1024x768. The software logged all the interactions performed by the user. To limit timing errors, during the experiment, the workstation was disconnected from the network, and logging data was only committed to disk between sets. To verify the accuracy of our timing method, we also compared the timing provided by our program to the timing provided by counting fields in video footage of some pilot experiments. In all cases, the results were in accord.

4.3 Task and Setting

For each condition, subjects were presented with 24 sets of 12 points to connect (11 connections per set). For each set, the computer presented the series of colored dots one by one. The subject 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 subject successfully connected the active dot, and consecutive dots were always of different

colors as in Kabbash et al. [1994]. 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.After each set, subjects 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. Subjects could only rest between sets. All conditions were run with users interacting with a pen (and a puck for the Toolglass) on a digital tablet while looking at a monitor (indirect setting, Figure 3). To improve comparability, subjects went through the same dot patterns used by Kabbash et al. [1994].

The screen layout is shown in Figure 4. The path created so far is rendered in gray with the exception of the last dot of the path, which is rendered in black[1]. All previous dots in the path are rendered filled. The new target dot is rendered as a circle of the 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 is 7/16" and distance between dots varies between 15/16" and 5 15/16".

4.3.1 Tool Palette and Toolglass. In the tool palette condition, the color tool palette consisted of 4 buttons, each 5/8" by 5/8", with a header 1 1/4" wide and 5/16" tall at the top, which the subject could use as a handle to move the tool palette. To perform a connection using the tool palette, the subject 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.

In the Toolglass condition, the color Toolglass consisted of 4 buttons, each 5/8" by 5/8", with a header 1 1/4" wide and 5/16" tall at the top. The Toolglass was set to 40% transparency so that the dots were clearly visible underneath it. The Toolglass was moved with a puck. To perform the task using Toolglass, the subject first had to bring the correct Toolglass color area on top of the last dot in the path using the puck. Then he or she had to click on the last dot of the path with the pen and then proceed directly with the rubber band interaction to connect the new colored dot.

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

---

[1] This differs from the setup used by Kabbash et al., [1994] where the created path was rendered in black. Our setting made the display easier for subjects to parse.
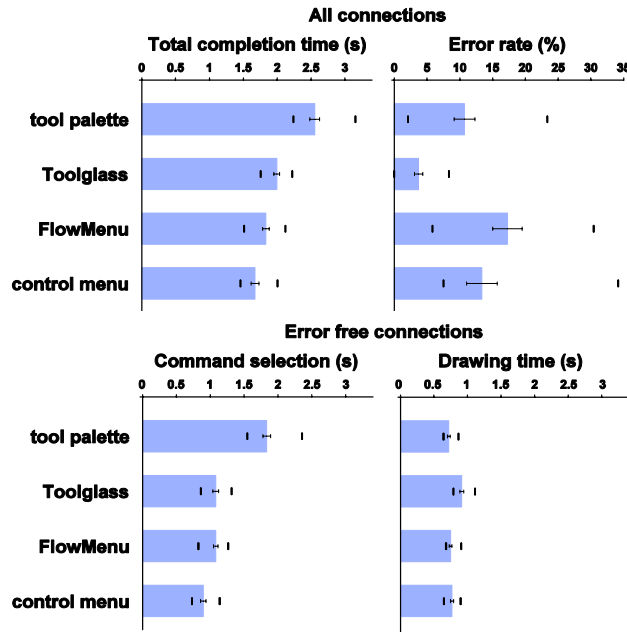
Fig. 5.Quantitative results. For each technique, we show from left to right: the average total connection time, the error rate, the average command selection time for connections without errors, and the average drawing time for connections without errors. Besides average values, the standard error, and the maximum and minimum values for each series are shown. Data used to draw these graphs are tabulated in Table II.

4.3.2 FlowMenu and the Control Menu. In the control menu and FlowMenu conditions, the radius of the menu was 1 3/16". To perform the task using a control menu or FlowMenu, the subject 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 he or she had to select a color by either leaving the rest area through the appropriate color's octant (control menu), or leaving the rest area through the appropriate color's octant then reentering the rest area (FlowMenu). Finally he or she had to proceed directly with the rubber band interaction to connect to the new colored target dot.

In both of these conditions, 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].

4.4 Protocol

After a brief description of the experiment, subjects were familiarized with the operation of the testing apparatus. For each condition, the correct way to perform the task was
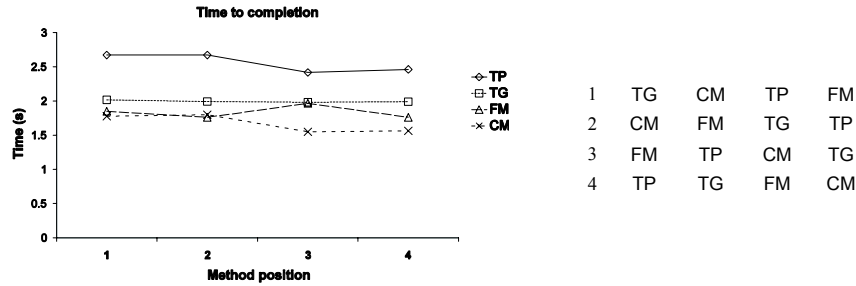
9

Fig. 6. Skill transfer analysis. For each technique, the graph presents the average task completion time depending on the technique position during each test. The differences are not statistically significant. For reference the Latin square we used is presented to the right.

explained to the subject, and each subject was given the opportunity to practice on 5 sets of 12 dots. The order of the experimental conditions for each subject was counterbalanced using a Latin square to limit order effects. Furthermore, the color layouts of the control menu and FlowMenu were arranged in different orders, as were those of Toolglass and the tool palette, in an effort to limit carryover effects.

After completing all trials, subjects completed a questionnaire giving subjective ratings of aspects of each technique on a scale from 1 (worst) to 7 (best) and providing information about their previous experience with similar systems. The total time for the experiment was about 1.5 hours.

5. RESULTS

As in Kabbash et al. [1994], the first connection in each set was removed from the data. As a result, we recorded 240 connections in each of the four conditions for each user. A box plot showed subject S9 (who used a double click to invoke the control menu) as an outlier for error rate in the control menu condition and subjective enjoyability in the control menu condition, so all S9 data points were removed from those two analyses. The results are shown in Figure 5 and Figure 8 with the numerical data tabulated in Table II.

To check for skill transfers, we examined the interaction between Technique and Order [Poulton, 1966]. The corresponding data is represented in Figure 6. As expected, the graph shows an overall improvement with position consistent with symmetrical skill transfer. The graph also shows an anomaly when the FlowMenu technique was performed as the third condition or in our case just after the Toolglass technique. These anomalies might suggest a possible asymmetric skill transfer between these two conditions. An ANOVA determined these interactions between Technique and Order $(F(9,32) = .75, p = .66)$ were not significant.

10

A repeated-measures ANOVA determined that means for total completion times were significantly different ($F(3,33) = 73.4$, $p < .0005$). All pairwise comparison significance levels use a Bonferonni correction for multiple comparisons.

5.1 Hypothesis 1

As shown in Figure 5, the tool palette was significantly slower than the control menu, FlowMenu, and Toolglass ($p < .0001$), so **our results support hypothesis 1**. We found slightly more relative improvement between Toolglass and the tool palette compared to Kabbash et al. [1994] (22% compared to the 16% they reported). Recorded traces (Figure 2) further illustrate that longer path length is the main reason for the inferior performance of the tool palette.

5.2 Hypothesis 2

The control menu condition was faster than the FlowMenu condition, but this difference was not significant ($p = .2$), so **our results do not support hypothesis 2**. Furthermore as pointed out in the beginning of the RESULTS section, the skill transfer analysis suggests that an interaction might exist between the Toolglass and FlowMenu when they are performed back to back. This interaction might artificially increase the difference between control menu and FlowMenu. To understand the possible effect of this interaction we ran our statistical analysis removing the three subjects for whom FlowMenu was used after Toolglass. For the group that remained, a repeated-measures ANOVA determined that means for total completion times were significantly different ($F(3,33) = 73.4$, $p < .0005$) and that the difference between FlowMenu and control menu was not significant ($p = 1.0$). These results further suggest that hypothesis 2 has to be rejected.

5.3 Hypothesis 3

Both FlowMenu and the control menu were significantly faster than Toolglass ($p = .01$, and $p < .0005$, respectively), so **our results do not support hypothesis 3**. This result is surprising given the extended body of work supporting the superiority of two-handed techniques over one-handed techniques (see Leganchuk et al. [1998] for an overview). Contrary to Kabbash's interpretation, our results suggest that Toolglass's advantage over standard techniques might not come from its two-handedness but from its merging of command selection and direct manipulation

While our main results include errors as part of the connection time, it is interesting to look at connections performed without errors more carefully. In the bottom half of
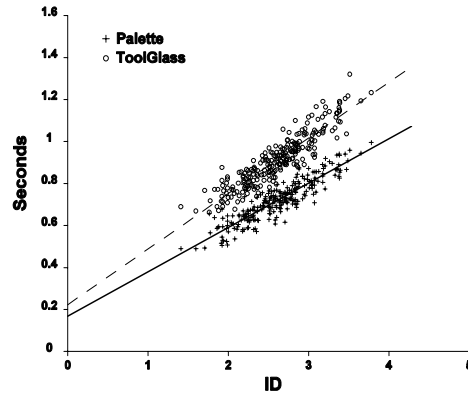
11

Fig. 7. Relationship between the indices of difficulty and the average drawing time (across users) for the 240 connections of our set. ID was computed as: $ID = \log_2(D/S + .5)$ [Welford, 1971].

Figure 5, we present the command selection time (time until the color is selected) and the drawing time for error-free connections. In both case, a repeated-measures ANOVA determined that mean times were significantly different, ($F(3,33) = 194$, $p < .0005$ and $F(3,33) = 17.8$, $p < .0005$, respectively). As expected, the tool palette had the slowest command selection time ($p < .0001$), and FlowMenu, which uses a more complex selection gesture, was slower than the control menu ($p < .0005$). FlowMenu and Toolglass had similar command selection times.

Toolglass drawing time was slower than that of the other techniques ($p < .005$). To explain this difference, we will focus on the difference between Toolglass and the tool palette, since their drawing tasks are almost identical. We plotted the average connection time for all users against the index of difficulty (ID) for our 240 connections. The result is plotted in Figure 7. As expected, there is a good linear correlation between ID and time to perform the connections ($r^2 = .83$ for Toolglass and $r^2 = .82$ for tool palette), but the slopes for each population are significantly different ($t_{\alpha(2),476} = 35.26$, $p < .0005$) with Toolglass connections being more "difficult" to handle. The most commonly observed way users carried out the task with the Toolglass 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 slows down the tracing part of the task, possibly because the user needs to attend to two tasks at the same time.

5.4 Error Rates

A repeated-measures ANOVA determined that means for error rates were significantly different ($F(3,30) = 13.2$, $p < .0005$, S9 removed). Toolglass was significantly less error
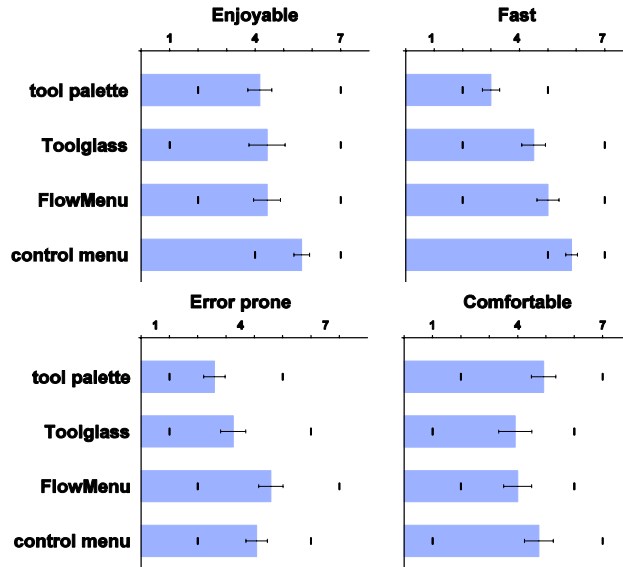
Fig. 8. Subjective ratings. For each technique we show from left to right: how much users enjoyed using the technique; how fast they felt it was; how error prone they felt it was; and how comfortable they felt it was. Data were collected using a scale from 1 (worst) to 7 (best). Besides average values, the standard error, the maximum and minimum values for each series are shown. Data used to draw this graphs are tabulated in Table II.

prone than any other technique ($p \leq .01$). One subject was even able to connect all 240 points without any errors using Toolglass. No other differences were significant.

5.5 Subjective Ratings

While a repeated-measures ANOVA determined that means for the "fast" and "error prone" subjective variables were significantly different ($F(3,33) = 11.1$, $p < .0005$, $F(3,33) = 4.8$, $p < .01$, respectively), the only significant pairwise comparisons were found for the "fast" variable. Overall subjective ratings for these two variables were in accord with the measured speed and error rate. More precisely, both FlowMenu and the control menu were perceived as being faster than the tool palette ($p < .01$), but only the control menu was perceived as being faster than Toolglass ($p < .05$).

We also recorded the users' subjective liking and level of comfort for each technique (Figure 8), but we did not find any significant differences.

6. DISCUSSION

The results presented above highlight the importance of techniques that can fluidly mix command selection and direct manipulation. Our results suggest that the main

13

Table I. Techniques used by Kabbash et al. [1994] and their reported times. In this Table we are using Kabbash's notation. In particular the tool palette is called R-tearoff by Kabbash.

| | | Command selection and Direct manipulation | |
		Not merged	Merged
Interaction	One handed	R-tearoff (2.89s)	
	Two-handed	L-tearoff (2.96s) Palette (2.90s)	Toolglass (2.43s)

advantage of the two-handed Toolglass comes from the fact that it lets the user smoothly merge command selection and direct manipulation.

It is important to note that our findings are consistent with Kabbash's results. To see this, we present the results of Kabbash et al. [1994] in Table I. From Table I, it is clear that even though Kabbash tested three two-handed techniques he observed a significant performance increase only when the technique merged command selection and direct manipulation (Toolglass). These observations reinforce our belief that merging command selection and direct manipulation is the key aspect of Toolglass's superior performance over the standard tool palette technique.

Our study was unable to show the expected significant difference between FlowMenu and control menus. We believe that a separate study and a finer-grained analysis (perhaps using the steering law proposed by Accot and Zhai [2002]) will be needed to determine the performance difference between FlowMenu and control menus.

6.1 Interaction Design Considerations

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

6.1.1 Versatility. Control menus, FlowMenu, and Toolglass 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 has a distinctive advantage because the Toolglass "see-through" metaphor lets the user start the interaction at the point where the command was invoked. Neither control menus nor FlowMenu provide this flexibility because they require the user to cross a specific boundary away from the point where the menu was called. Note that pie menus [Hopkins, 1991] and the extension of marking menus proposed by Kurtenbach and

14

Buxton [1991] 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.

6.1.2 Design Considerations. Our results open new avenues in the design of efficient interfaces in situations where two-handed interactions are not practical. For example, for a PDA or Tablet computer, it might be difficult to use both hands at the same time or too expensive to install two tracking devices. Another area that might benefit from our finding is that of interaction design for large interactive surfaces. Large interactive surfaces are becoming more and more prevalent and are designed with a direct interaction mode in mind. For those purposes, we believe that a two-handed interface may still have benefits. However, instead of using the non-dominant hand to help select the current command, as with Toolglass, one might use it to set the frame of reference for the work or to manipulate tools, as in T3 [Kurtenbach et al., 1997]. Guiard [1987] has advocated this setting and our results suggest that this approach might deliver significantly better interfaces.

6.1.3 Scalability. Another important point in selecting command mechanisms is scalability. While it is true that the basic operation in our "connect the dots" task resembles a variety of different direct manipulation actions, 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:

$$CT = CST + DT \text{ (for control menu, FlowMenu and Toolglass)} \tag{1}$$

$$CT = \frac{CST}{n} + DT \text{ (for the tool palette since the color is only selected once)} \tag{2}$$

Equating (1) and (2) and the using numerical values for *CST* and *DT* provided in Table II, one can see that performing as few as two operations per tool selection might be enough to make the tool palette faster than Toolglass and FlowMenu (it would take three to be faster than a control menu). These results reflect error free connections, but of course a complete analysis should also take into account the fact that users might be more likely to forget to select a color after a long streak of dots from one color.

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

15

accommodate a larger set of commands (see for example the T3 system of 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. Future research needs to address these issues.

7. FUTURE WORK

It is obviously difficult to extrapolate quantitative findings in a controlled experiment to more realistic situations. In our case, we were unable to model how users amortized the cost of command selections with a palette by performing several successive actions with one tool. To solve this problem we are exploring methods to gather quantitative data from real usage behavior for tool palettes in drawing and CAD tools.

We would also like to broaden our understanding of how two-handed techniques can improve other operations. Recent results [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.

Finally, we would like to explore the new design avenues opened by our results. In particular, we would like to pursue interface designs that either remove the current limitation of control menus and FlowMenu to non-freehand applications or propose a new style of two-handed interaction that lets the non-dominant hand orient the work while the dominant hand performs the work in the frame of reference set by the non-dominant hand.

8. CONCLUSION

In this paper we have presented new evidence for the benefits of mixing command selection and direct manipulation in commonly performed direct manipulation tasks of modern interfaces. Our results show that these benefits can be obtained not only by using two-handed interaction techniques such as Toolglass but also by using one-handed techniques such as control menus and FlowMenu. Our analysis of these results should aid in the understanding of the advantages of two-handed command selection and in weighing them against possible drawbacks such as more complex hardware. We have also highlighted the need for further analysis to better understand the real advantage of these techniques for actual patterns of use as observed in large industrial-strength applications.

Table II. Means and standard error (shown in parenthesis) for each variable in our data set.

	Tool palette	Toolglass	FlowMenu	Control menu
Total connection time (s)	2.56 (0.0725)	1.99 (0.0439)	1.83 (0.0486)	1.67 (0.0567)
Error rate (%)	10.7 (1.57)	3.72 (0.64)	17.3 (2.24)	13.4 (2.33)
Command time for error-free connections (s)	1.74 (0.0221)	1.02 (0.0276)	0.955 (0.0202)	0.724 (0.0234)
Drawing time for error-free connections (s)	0.720 (0.0603)	0.914 (0.0369)	0.747 (0.0311)	0.769 (0.0290)
Enjoyable (1-7)?	4.2 (0.42)	4.4 (0.63)	4.4 (0.47)	5.6 (0.28)
Fast (1-7)?	3 (0.30)	4.5 (0.42)	5 (0.39)	5.8 (0.21)
Error prone (1-7)?	2.6 (0.38)	3.3 (0.45)	4.6 (0.43)	4.1 (0.38)
Comfortable (1-7)?	4.9 (0.43)	3.9 (0.58)	4 (0.49)	4.8 (0.51)

9. ACKNOWLEDGEMENTS

10. REFERENCES

Accot, J., and Zhai, S. 2002. More than dotting the i's – Foundations for crossing-based interfaces. In *Proc. CHI '02*, Minneapolis, MN, ACM Press, 73-80.

Balakrishnan, R. and Hinckley, K. 1999. The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. In *Proc. UIST '99*, Asheville, NC, November 1999, ACM Press, 171-178.

Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. 1993. Toolglass and magic lenses: The see-through interface. In *Proc SIGGRAPH '93*, Anaheim, CA, August 1993, ACM Press, 73-80.

Bourgeois, F., and Guiard Y. 2002. Multiscale Pointing: Facilitating Pan-Zoom Coordination. In *Extended Abstracts CHI '02*, Minneapolis, MN, April 2002, ACM Press, 758-759.

Dillon, R.F., Eday, J.D., and Tombaugh, J.W. 1990. Measuring the True Cost of Command Selection: Techniques and Results. In *Proc. CHI '90*, Seattle, WA, April-May 1990, ACM Press, 19-25.

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

Guimbretière, F., and Winograd, T. 2000. FlowMenu: Combining Command, Text, and Parameter Entry. In *Proc. UIST '00*, San Diego, CA, November 2000, ACM Press, 213-216.

Hopkins, D. 1991. The Design and Implementation of PieMenus. *Dr. Dobb's Journal 16(12)*, 16–26.

Igarashi, T. and Hinckley, K. 2000. Speed-dependent automatic zooming for browsing large documents. In *Proc. UIST '00*, San Diego, CA, November 2000, ACM Press, 139-148.

Kabbash, P., Buxton, W., and Sellen, A. 1994. Two-handed Input in a Compound Task. In *Proc. CHI '94*, Boston, MA, April 1994, ACM Press, 417-423.

Kurtenbach, G. 1993. The Design and Evaluation of MarkingMenus. PhD thesis, University of Toronto.

Kurtenbach, G., and Buxton, W. 1991. Integrating mark-up and direct manipulation techniques. In *Proc. UIST '91*, Hilton Head, SC, November 1991, ACM Press, 137 - 144.

Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, W. 1997. The Design of a GUI Paradigm Based on Tablets, Two-Hands, and Transparency. In *Proc. CHI '97*, Atlanta, GA, March 1997, ACM Press, 35-42.

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

McGuffin, M., Burtnyk, N., and Kurtenbach, G. 2002. FaST   Sliders: Integrating Marking Menus and the Adjustment of Continuous   Values. In *Proc. of Graphics Interface '02*, Calgary, Canada, May 2002.

Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. 2000. Control Menu: Execution and Control in a Single Interactor. In *Extended Abstracts CHI '00*, The Hague, The Netherlands, April 2000, ACM Press, 263-264.

Poulton, E. C. 1966. Unwanted asymmetrical transfer effects with balanced experimental designs. *Psychological Bulletin 66 (1)*, 1-8.

Welford, A. T. 1971. *Fundamentals of Skill*. Methuen, London.