Measuring FlowMenu Performance

This paper evaluates the performance characteristics of FlowMenu, a new type of pop-up menu mixing command and direct manipulation [8]. FlowMenu was compared with marking menus [12] on a simple selection task, and with tool palette selection [11] on a task that combined selection with drawing.

Our results compared with those in previous experiments [11] indicate that Flowmenu offers as much of an advantage over tool palette selection as does Toolglass [2], a two-handed technique. At the same time, it is not significantly slower than marking menus. Additional results show that selection performance depends on distance around the menu, but not on initial position or direction. These results provide a guide for the design of high-performance menu configurations.

Keywords: FlowMenu, marking menu, tool palette, Toolglass, empirical studies;

1. INTRODUCTION

Interaction mechanisms designed for traditional pointer and keyboard interfaces are not optimized for direct interaction with a display surface using either pens or fingers. Many pen-based interaction techniques have been introduced for use with devices ranging from handheld [15] and tablet computers [14, 6] to large wall-mounted high-resolution displays [5, 9]. For example, when standing in front of a large high-resolution display, it is impractical to use conventional GUI widgets such as menu bars and tool bars at the edges of the screen, due to the combination of the large physical size and control by direct pointing. Interaction needs to be focused on an immediate locus of attention, through techniques such as pop-up menus or pen gestures.

A number of pop-up command techniques have been developed for use with pen-on-display systems, including pie menus [10] and marking menus [13]. FlowMenu [8] combines the radial geometry of these techniques with the ability to enter a command and parameter information within a single continuous stroke. By using a "return-to-center" activation path similar in spirit to the QuikWriting text entry system [16], the FlowMenu provides multi-level menuing and a transition from command to direct manipulation that

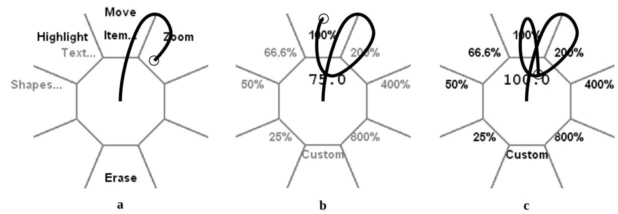

Figure 1: A simple FlowMenu interaction. After calling the menu, to resize an item, the user moves the pen from the rest area into the *Item...* octant. Submenus (*Highlight, Move, Zoom*) appear and the first level menu items not selected are grayed out (**a**). Entering the Zoom octant submenu then moving back to the rest area dismisses the root level menu and brings up the zoom menu (**b**). A new zoom value of 100% is selected by moving into the octant for the desired value and back to the center, at which point the zoom is applied (**c**) The pen track is shown here for illustration.

does not involve picking up the pen, clicking a button, or any other interruption.

Command selection at the locus of attention and fluid transition between menu and direct manipulation should result in faster performance in direct manipulation tasks. To test this hypothesis, we used a simple color painting "connect the dots" task that was used by Kabbash [11] to compare a standard color tool palette and a Toolglass [2]. He found that the Toolglass was significantly faster, and attributed this speed increase to the use of two hands (one for the Toolglass on which colors were selected, the other for the painting). Using a setting similar to Kabbash's, we compared the color tool palette with a FlowMenu configuration that allowed color selection and drawing in the same stroke. In order to test whether the effect was dependent on a particular kind of device (Kabbash used a mouse for cursor control), we performed the experiment with three different device configurations: a mouse, a pen on a tablet, and a pen on screen. Experimental results with the mouse condition showed that our one-handed technique provided a similar advantage to the Toolglass, casting doubt on Kabbash's conjecture that two-handed manipulation provided the advantage. Comparison across conditions showed that mouse and pen-on-tablet were similar, but that direct pen-on-screen was faster than the indirect modes, primarily in the acquisition of a color button in the tool palette.

Our second experiment compared the speed of selection between FlowMenu and marking menus, based on Kurtenbach's study [12] of expert performance with marking menus. We hypothesized that the difference would be dependent on the angular distance between exit and reentry to the menu center. Our results showed that FlowMenu is close in speed to marking menus of similar complexity, and is relatively isotropic.

2. RELATED WORK

FlowMenu [8], was introduced as a new kind of pop-menu well suited for large interactive surfaces. FlowMenu is a radial menu with 8 octants and a central rest area (Figure 1). Upon invocation, the menu pops up centered on the pen. The user selects a top-level menu item by leaving the central area into the corresponding octant. As she does, sub-menus for this menu octant appear. Moving the pen to the desired sub-menu octant and reentering the rest area from that octant will trigger a menu selection. This command mechanism is similar to marking menus [13], but by activating on return-to-center instead of on pen-up, it is possible to continue in the same stroke with parameter entry or direct manipulation. FlowMenu has been used as the primary selection mechanism for a brainstorming tool on a wall-size display [9], but no previous studies of its performance have been reported.

Kabbash tested the performance that could be achieved using different affordances for a simple colored connect-the-dots task [11]. Subjects were asked to connect a series of colored dots appearing on the screen. For each new dot, the subject needed to pick the corresponding color and draw a line from the previous dot to the new dot. Color picking was done using a conventional tool palette (called *R-tearoff* by Kabbash), Toolglass (an *asymmetric bimanual* see-through interaction introduced by Bier [2]), and 2 other techniques not addressed in the current paper. Kabbash reported significantly better performance with the two-handed Toolglass condition, attributing the performance gain to the use of an "asymmetric dependent" technique [7].

Kurtenbach's study of marking menus [12] attempted to evaluate the performance of "experts" by eliminating effects of menu structure learning. He measured and compared the command selection time for different menu configurations using compass directions (such as NW-NE), which mapped directly onto the menu geometry as prompts. To reinforce learning and simulate expert performance, for each menu configuration, subjects were only presented with 3 different menu selections repeated 8 times in randomized order. With this setup, he demonstrated that pen operations were faster than mouse operations and presented an analysis of the relative performance of different menu structures.

3. EXPERIMENTAL DESIGN

Our experimental design adapted the designs from the previous literature in order to facilitate direct comparisons with the previously published results.

3.1 Connect the dots study

This study measured FlowMenu performance in a task in which command and interaction were interleaved. Following the experimental design used by Kabbash [11], subjects were asked to connect a series of colored dots on the screen, using either a tool palette or a FlowMenu to select a color. To assess the influence of the input modality, we compared subjects' performance with an indirect, relative device (mouse), an indirect absolute device (pen on tablet), and a direct absolute device (pen on display). This resulted in 6 possible conditions. The full condition matrix is represented in Figure 2.

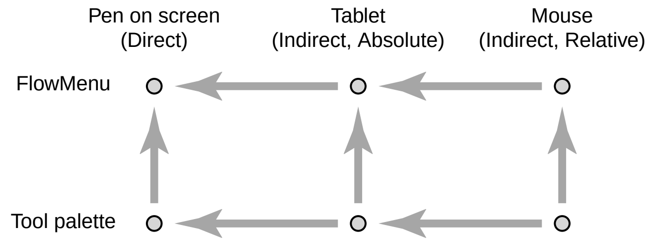

Figure 2: Condition matrix for the connect the dots study. Arrows point to the faster condition, according to hypotheses H1 and H2.

We did not attempt to duplicate the two-handed Toolglass condition in our experimental setup, but we were able to directly compare our results for the tool palette condition with those of Kabbash[11], and started with the actual stimulus sets he employed.

For each condition, subjects were presented with 12 sets of 12 points to connect. In each case, the computer presented a series of colored dots one by one. The subject connected the last dot to the new 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. The "connection time" is 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 dot. After each set, subjects were presented with their aggregate time for the set and their best time so far, to duplicate the feedback conditions in Kabbash's experiment. Subjects could only rest between sets.

Due to constraints on the length of the test for each subject, we used only 12 out of the 24 sets used by Kabbash, selecting the ones with the least chance of occlusion of the new target dot by the subject's hand in the direct interaction condition. To compare our results directly to Kabbash's [11], all 24 sets were run in the "mouse with tool palette" condition, a condition Kabbash called "R-tearoff."

Hypotheses:

H1: FlowMenu will be faster than the tool palette condition in all cases. Because it provides color selection at the locus of attention and provides fluid transition between command selection and drawing, FlowMenu avoids the time needed to reach for the tool palette to select the color.

H2: FlowMenu indirect pen condition will be faster than mouse interaction, and FlowMenu direct pen condition will be faster than indirect condition. This hypothesis is in accordance with results by Albert [1].

H3: FlowMenu will be slower than a two-handed Toolglass condition. Following Kabbash's analysis, the use of two hands leveraging from our everyday experience will provide better performance than single-handed interactions.

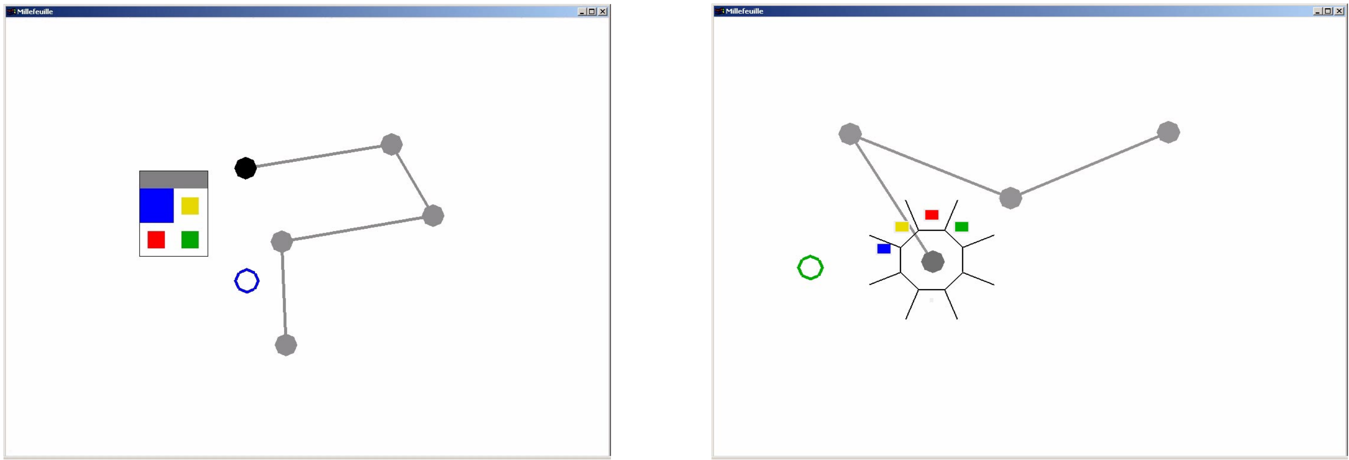

Figure 3: Screen layout for the connect-the-dots study. Using the tool palette (left) or FlowMenu (right) to select the correct connecting color, the user must connect 12 successive dots. This task was performed using 3 different input devices: a mouse (indirect relative), an indirect pen tablet (indirect absolute) and a direct pen on screen (direct).

Figure 2 shows the condition matrix for this experiment as well as the expected outcomes following hypothesis H1-H2.

Setting

The screen layout is shown in Figure 3. The path created so far is rendered in grey 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 dot to connect to is rendered as a circle of the requested color. This setting, slightly different from [11], was used to make the display easier for subjects to parse. Because of the constraint imposed by the side of the Wacom tablet screen and the need for a 1” border while running FlowMenu, the size of our setting was scaled relative to that of [11]. Pilot tests confirmed that this scaling factor did not influence performance.

In the tool palette condition, the color tool palette consisted of 4 buttons, each 9/16” by 9/16”, with a header 1/8” wide and 5/16” tall at the top, which the subject could use as a handle to move the tool palette. In the FlowMenu condition, the average radius of the FlowMenu rest area was 1/2” and the average overall radius was 1 1/8”. All colors were accessible directly from the rest area. Each dot radius was 3/16”.

To perform the task using the tool palette, the subject first had to select the correct color by clicking on the appropriate color button and then click on the last dot of the path and perform a rubber band interaction to connect this dot to the new colored dot. The line of the selected color was shown on the screen as feedback for the rubber band interaction.

To perform the task using FlowMenu, the subject had to call the FlowMenu on top of the last dot of the path by clicking on the command button (pen) or the left button (mouse) while pointing to the dot, then select a color and proceed directly with the rubber band interaction to connect the new colored dot. A line of the selected color was shown on the screen as feedback for the rubber band interaction. In the case of the pen, the command button could be released as soon as the menu appeared.

3.2 Command selection study

In this experiment, we modified the design employed by Kurtenbach [12], in which compass directions were used as the stimuli to specify menu choice. This layout was intended to simulate “expert” performance, in which subjects do not require time to locate the menu choice on the menu. However, since subjects often require some thought to map an abstract symbol such as “NE” onto a direction, Kurtenbach used only 3 different direction pairs in a trial. Each of the 3 selected pairs were repeated 8 times, and the 24 pairs were presented at random to the subject.

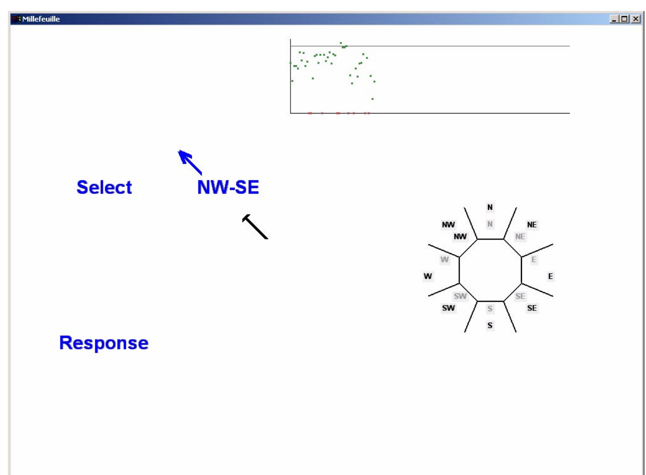

Figure 4: Screen layout for the command selection study. The screen is shown just after the user has pressed the command button and entered the NW octant. The system shows the requested command at the top left (“Select”) and in case of an error the selection performed by the user at the bottom left (“Response”). A blue sharp arrow shows the exit direction and a black flat arrow shows the return direction for the requested selection. These arrows were not shown for the “No-Arrow” setting. The chart on top indicates the selection speed with comparison to a reference selection time. A red dot on the x axis indicates a failed attempt.

In order to be able to effectively test the results of all the different direction pairs (8x8) without requiring impractically large numbers of subjects, we devised a different method for indicating the directions. We augmented the compass direction with 2 arrows showing the subject where to exit the central area and where to reenter it (Figure 4). This approach reduces even further the cognitive load associated with figuring out the shape of the stroke to draw. It allowed us to simulate expert performance using a relatively small number of samples for each of the 64 possible command selection patterns. To be able to compare our results with previous experiments and also to provide the coverage we needed, we ran our experiment using 2 conditions, the "With-Arrows" condition illustrated in Figure 4, and a "No-Arrows" condition that omitted the arrows and was directly comparable to Kurtenbach's experiment.

In the "With Arrows" condition each subject was presented with 128 command selections in randomized order, so that each of the 64 possible menu trajectories was presented twice.

in the "No-Arrows" condition, as in Kurtenbach [12], each subject was presented with 120 command selections in 5 sets of 24 selections. Each set was constructed by picking 3 menu selections, repeating them 8 times and presenting the resulting 24 selections at random. In this design, each subject performed 15 unique menu selections. To get the best possible coverage, we presented each command difficulty (number of octants between the starting point and the ending point) 3 times. Specific instances for a given command difficulty (starting direction and left or right rotation) were picked at random. As in [12] the experiment was self-paced.

In all conditions, the requested menu selection appeared as soon as the subject pressed the command button and selection time was measured from the moment the stimulus appeared until the selection was completed by returning to the center rest area. The selection could be performed anywhere on the screen. In case of an error, a beep was generated and the system showed the selection performed so that the subject could compare it with the requested selection but no attempt was made at correction. As in [12], we continually displayed a graph showing the subject's performance. For each selection, the graph showed whether the selection was successful or not and in case of a successful selection how close the selection time was to a reference time for that menu selection. Failure was represented by a red dot on the x axis.

Each subject performed one "No-Arrows" set, and 2 "With Arrows" set, always finishing with a "With Arrows" set. The order of the first two conditions was alternated between subjects. All interactions were performed using a direct pen input mechanism, a setting similar to Kurtenbach's.

Hypotheses:

H4: Because the gestures used in FlowMenu are more complicated than the ones used with marking menu, FlowMenu will be slower than a marking menu of the same complexity.

Figure 5: Experimental setting showing the mouse (front right), the Wacom PL400 (front) used as a tablet in direct and indirect conditions, and the display (rear) used for indirect conditions.

H5: In return-to-center menu selection, there will be a bias depending on the starting direction (following the result in [4]) and the rotational direction (following [17]).

Setting

The layout of the screen used for the experiment is shown in Figure 4. On the left of the screen, the system displayed the requested selection and in case of an error the performed selection. The graph at the top of the screen displayed the performance of the subject.

To call up the menu, the subject pressed a button on the side of the pen while holding the pen anywhere on the screen. As he or she did so, the next selection to be performed was presented and a timer for this selection was started. We measured both the total time for a selection and the time after departing the center to the selection. The test was self paced in that we did not measure the time between individual trials.

4. PROTOCOL

18 right handed, non color-blind subjects (12 men and 6 women) were recruited from a young adult population. All subjects were skilled using a mouse, but had little or no experience using a pen interface other than on a PDA. All subjects had little or no knowledge of FlowMenu. One additional subject elected to stop before completing the experiment, and this data set was removed from the analysis.

4.1 Equipment

For both experiments, we used the setup shown Figure 5. Direct interactions were performed on a Wacom PL400 tablet. This tablet combines a 1024x768 LCD display with an RF-tracked pen on a graphic tablet. The visible area of the tablet is 13.25" diagonal.

For indirect manipulation, we used a Sony Multiscan E100 15" monitor with a visible area of 13.5" diagonal as a display. For the pen condition, subjects used the pen on the

		Select (s)	Draw (s)	Total (s)
Tablet (indirect)	FlowMenu	1.21 (.179)	.859 (.152)	2.07 (.287)
	Tool palette	1.84 (.271)	.754 (.0976)	2.59 (.332)
Mouse	FlowMenu	1.16 (.135)	.85 (.0724)	2.01 (.166)
	Tool palette	1.75 (.186)	.746 (.0646)	2.50 (.227)
Tablet (direct)	FlowMenu	1.25 (.100)	.659 (.0925)	1.91 (.160)
	Tool palette	1.52 (.149)	.568 (.0918)	2.09 (.192)
Tablet (direct, no occl.)	FlowMenu	1.12 (.115)	.649 (.0996)	1.77 (.168)
	Tool palette	1.47 (.150)	.557 (.0909)	2.03 (.194)

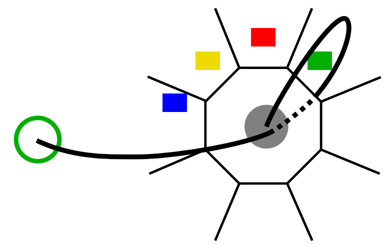

Figure 6: Connect-the-dot experiment results.The table shows the average time in second to complete a connection in the connect-the-dot experiment, for the 12 sets minimizing occlusion (section 3.1), for all 18 subjects. Standard deviation is shown in parenthesis.FlowMenu is significantly faster than the tool palette for all conditions ($p <.01$). The difference in performance appears to come from color selection times, which are significantly different for each condition ($p < .01$). FlowMenu has a greater drawing time in each condition ($p < .05$). The sketch to the right provides some explanation for this difference: in the FlowMenu condition, the drawing time starts as soon as the color is selected rather than when the cursor is above the dot. The path difference is shown as a dotted line. Analysis of video footage suggests that it takes 45 ms to cover this distance. For "Tablet (direct, no occl)", data points leading to occlusion have been remove from the data set.

same Wacom PL400 tablet in absolute mode with no image displayed on the tablet.

For the mouse condition, we used a standard non-optical mouse. Pilot studies showed that subjects preferred the non-optical mouse to an optical mouse, with a corresponding performance advantage.

4.2 Experiment

After having the experiment explained to them, subjects first ran the command selection experiment, then the connect the dot experiment. This allowed subjects to gain some experience using FlowMenu. After completing all trials, subjects completed a questionnaire giving subjective ratings on a scale from 1 (worse) to 7 (best), and information about their previous experiences on similar systems. The total time for both experiments was around 2 hours.

Command selection experiment

Before each set of trials, the setting was explained to the subject including different styles that could be used to perform the task and he or she was given the opportunity to practice as much as he or she wanted (on average subjects practiced with one 20 command selection set).

Connect-the-dots experiment

The setting was explained, including the different styles that could be used to perform the task. The subject was given the opportunity to practice as much as he or she wanted (on average 4 sets of 12 dots).

5. RESULTS AND DISCUSSION

All statistically significant differences given in this section have been computed using a paired-sample two-tailed t-test except when noted otherwise.

5.1 Connect the dots experiment

As shown in Figure 6, FlowMenu was significantly faster ($p < .01$) than tool palette for all input devices, so **we can accept hypothesis 1**.

As expected from Albert's work [1], a direct pen is faster than all other devices. In results from our experiment, this difference is significant for the tool palette ($p < .01$), but is not significant for the FlowMenu. The difference between the indirect relative and the indirect absolute is even smaller. So **we have to reject hypothesis 2**.

Separating drawing time and color selection time, we see that the two techniques have similar drawing times. FlowMenu's advantage comes from a reduced selection time. It is interesting to see that the difference is reduced sharply in the case of direct manipulation on the screen. Figure 7 depicts the timing difference between the direct and indirect

	Indirect to direct variations		
	Menu	Draw	Total
FlowMenu	3.5%	-23.2%	-7.6%
tool palette	-17.1%	-24.7%	-19.3%

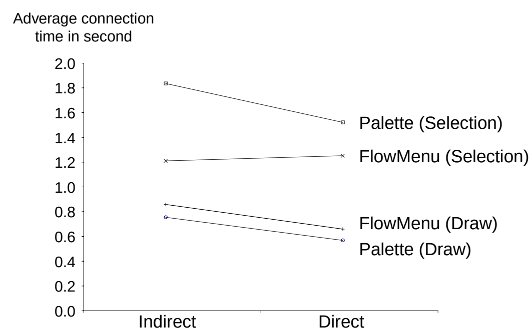

Figure 7: Indirect versus direct timing. While drawing time profits equally from the direct setting, the tool palette menu selection time improves considerably when hand-eye coordination can be used. Note the similar slope between "tool palette (Selection)" and "tool palette (Draw)" suggesting that a similar mechanism is at play.

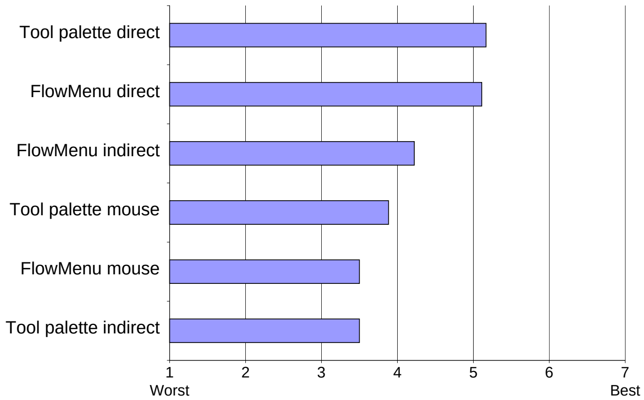

Figure 8: Subjective ratings. Subjects significantly prefer the direct condition over the indirect condition ($p < .01$).

conditions while using a pen. When switching from indirect to direct, both techniques show more than a 23% drop in their drawing time. But while FlowMenu shows a very small change in selection time, the tool palette condition shows a 17% drop in color selection time. In [1], Albert pointed out that eye-hand coordination provides a sharp increase in performance for cursor positioning tasks. This improved performance is demonstrated in the drawing time for both techniques, but FlowMenu selection benefits little from the eye hand coordination since it relies on gesture memory rather than target acquisition. In contrast, tool palette selection benefits from eye-hand coordination since it is a cursor positioning task.

The analysis in Figure 7 has implications for use of FlowMenu as compared to moveable tool palettes on larger surfaces. As the surface gets larger, the cost of using FlowMenu will stay constant since the cost of selection with FlowMenu does not depend on cursor positioning performance. The cost of using a tool palette configuration will become higher either because the subject will have to move the tool palette or take longer to select from it, following Fitts' law.

During testing we observed that occlusions were more of a problem while using the FlowMenu than with the tool palette. Problems occurred when the new dot appeared just below the hand of the subject. Even though we selected sets that minimized occlusion, we still had some occlusions during the tests. The influence of such occlusions is visible in comparing the third and last row in Figure 6. The last row shows the timing data where the samples resulting in occlusion have been removed. In both conditions the difference is significant ($p < .01$), but the difference is small for the tool palette and large for FlowMenu. This result reflects a fundamental difference of design between the two interaction techniques. While lifting the pen is a natural part of interaction using the tool palette, subjects can avoid doing this while using the FlowMenu. As a result, occlusions, which force the subject to lift the pen to discover the next point, have a larger influence.

As explained in section 3.1, our goal was to recreate a setting similar to the one described by Kabbash so that we could compare FlowMenu with Toolglass. Our average completion time in the "tool palette with mouse" condition using all 24 sets provided by Kabbash is 2.50 s (same value as for the 12 sets we picked), significantly lower ($p < .01$) than 2.89 s reported by Kabbash. Pilot studies using a different kind of mouse and different scaling factors did not explain this difference. Since our setup resulted in different times for the same (tool palette using a mouse) condition, we cannot make direct comparisons of our time for FlowMenu and his time for Toolglass. However, we can compare the relative benefit of the Toolglass and FlowMenu conditions compared to the tool palette in both experiments. Kabbash's Toolglass was 15.9% faster then the tool palette. In our experiment we found that FlowMenu is 19.6% faster than the tool palette.

Looking at these results **we have to reject Hypothesis 3**: FlowMenu can achieve better performance than Toolglass relative to a tool palette. Toolglass and FlowMenu are similar in that they both provide a color selection at the locus of attention and provide a simple transition between command and drawing. The main difference is that FlowMenu is a one-handed device while Toolglass is a bimanual technique. Our results seems to suggest that the main advantage of both FlowMenu and Toolglass is providing a selection mechanism at the locus of attention which is seamlessly integrated with direct action. Further studies will be needed to determine the relative role of the asymmetric dependent construction in the Toolglass performance.

Figure 7 shows the results of the survey collected at the end of each session. Subjects indicated a significant preference for the direct condition ($p < .01$). Compared to the tool palette, FlowMenu is perceived equally well or better.

5.2 Menu selection task

Figure 9 shows the average performance for FlowMenu and marking menus of similar complexity (breadth 8, depth 2, in Kurtenbach's nomenclature). FlowMenu is 60ms slower than the reported result by Kurtenbach. Our error rate 9.98% is higher than the one reported for marking menus 6.64%. Neither difference is statistically significant ($p > .5$ and $p > .15$ respectively). As expected because of the lower cognitive load, the "With-Arrow" condition is faster than the "No-Arrow" condition ($p < .05$).

	FlowMenu (arrows)		FlowMenu (no arrows)		Marking Menu (no arrow)	
Time (s)	1.28	(.245)	1.47	(.285)	1.41	(.298)
Error rate (%)	10.8	(5.14)	9.98	(5.70)	6.64	(6.56)

Figure 9: Comparison of timing and error rate for the 3 settings: As expected, the "With-Arrow" protocol is faster. For reference we show marking menu results from Kurtenbach [13]. FlowMenu performance is similar to marking menus. The high error rate in our setting may have been influenced by the characteristics of the pen we used during the experiment.

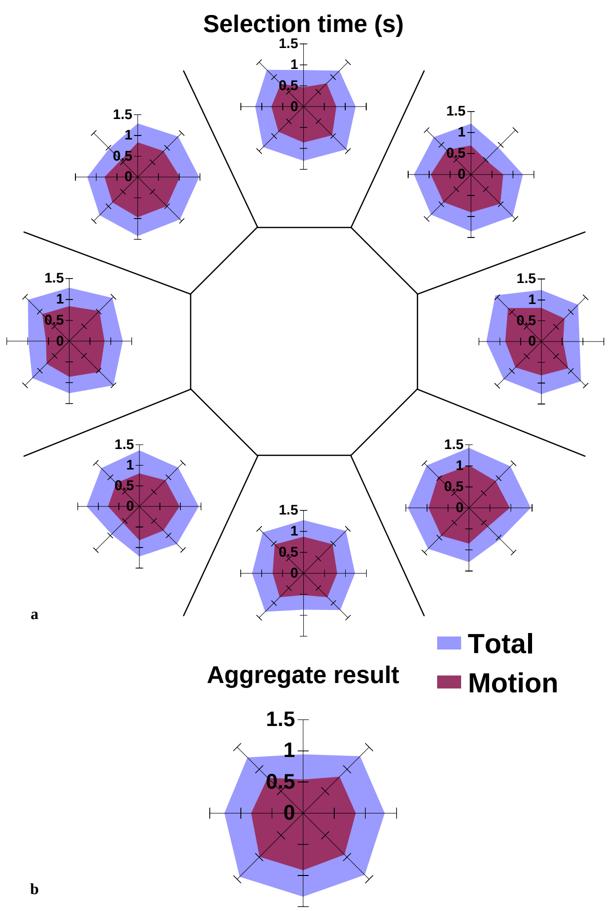

Figure 10: Selection speed for FlowMenu using the "With Arrow" setting. (**a**) shows for each starting octant the time it takes to reach a given octant. Light blue shows total time, while the dark red center shows time to completion after motion is begun. No starting octant is faster than any other octant and there is no bias between clockwise and counter-clockwise rotation. As expected, the further apart the starting and ending octant are, the longer the selection time. (**b**) shows the average of all normalized samples. The average selection time is 1.28s.

In Hypothesis 4, we conjectured that FlowMenu will be slower than marking menus given the apparent complexity of the former. According to our result, there is only a small difference (60ms or 4%) between the 2 techniques. This small difference is easily counterbalanced by the advantage of being able to mix command selection and direct manipulation. Furthermore, since command selection time depends only on the distance between the menu octant and the submenu octant, a careful design of the menu structure could group often-used submenus close to their primary menu, delivering a better average result. **We can neither reject nor accept hypothesis 4**.

FlowMenu has a higher error rate than marking menus. As shown in Figure 10, most of the errors are accumulated during the selection of far-away submenus, implying that a careful design of the menu structure will improve accuracy. It is also possible that part of the difference was due to the slippery interface between the pen and the screen, which was a problem reported by some of our subjects.

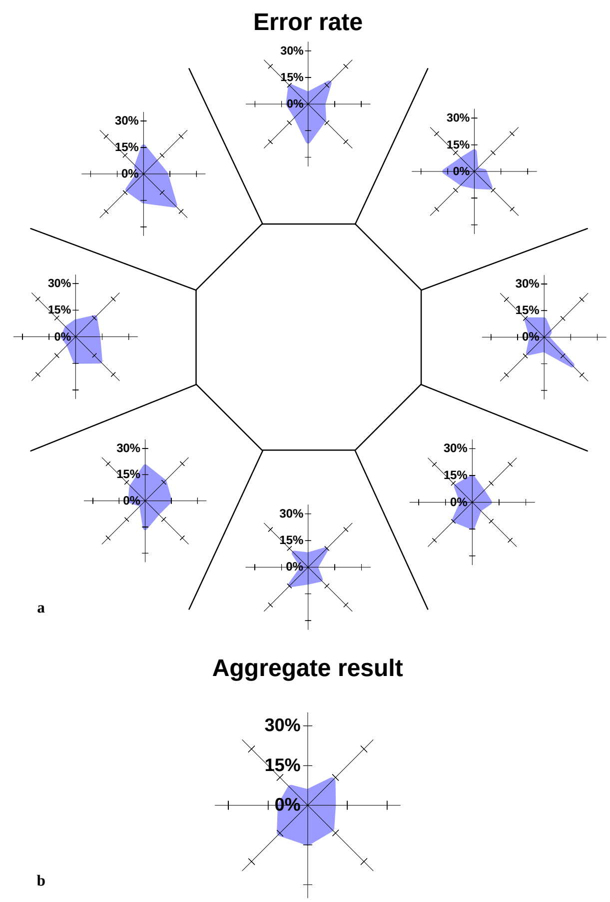

Figure 11: Error rate for FlowMenu using the "With Arrow" setting. (**a**) shows for each starting octant the error rate while reaching a given octant. As expected, the further apart the starting and ending octants, the higher the error rate. (**b**) shows the average error rate for all normalized samples. The overall average error rate is 10.8%.

Figure 10 and Figure 11 show the data collected using the "With-Arrow" condition to study the performance of FlowMenu depending on the starting and ending direction. No starting direction is faster than another, and there is no substantial asymmetry between accessing commands left of the starting point and right of the starting point. As expected, the further away the submenu is from the starting octant the longer it takes to make a selection and the more mistakes are made. So **we have to reject hypothesis 5**.

6. CONCLUSION

FlowMenu combines the general advantages of pop-up with the ability to integrate command selection and direct manipulation. This feature gives a significant performance advantage for tasks like the connect the dots, in which selection and action are integrated. It gives as much of an advantage as a two-handed technique for the same task. It appears that integrating command selection and direct action may be the key gain of both FlowMenu and Toolglass but further experiments will be needed to provide a quantitative basis for this claim. At the same time, FlowMenu is not significantly slower than marking menu techniques, which have been shown to be the most effective for command selection alone.

FlowMenu, which was designed initially for wall size displays, proves to be a versatile and efficient command mechanism for more commonly used input device configurations (mouse, pen tablet) achieving an improvement over conventional techniques which is greater than the improvement reported for a two-handed technique.

7. ACKNOWLEDGEMENTS

Removed for anonymous review

8. REFERENCES

1. Albert, A.E. The Effect of Graphic Input Devices on Performance in a Cursor Positioning Task. *In Proc. Human Factors Society'82*, pp. 54-58.
2. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: The see-through interface. *In Proc SIGGRAPH'93*. pp. 73-80.
3. Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An Empirical Comparison of Pie vs. Linear Menus. *In Proc. CHI'88*, pp. 95-100.
4. Corrigan, R. E., and Brogden, W. J. The effect of angle on precision of linear pursuit movements. *American Journal of Psychology*, 61, pp. 502-510.
5. Funkhouser, T., and Li K., (eds.). Special issue of *IEEE Computer Graphics and Applications.* Onto the Wall: Large Displays, 20(4) (Jul/Aug 2000).
6. Carr, R., and Shafer, D. The Power of Penpoint (1991).
7. Guiard, Y. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *J. Motor Behaviour, 19(4)* (Dec 1*987)*, pp. 486-517.
8. Guimbretière, F., and Winograd, T. FlowMenu: Combining Command, Text, and Parameter Entry, In *Proc. UIST'00*, pp. 213-216.
9. Guimbretière, F., Stone, M. and Winograd, T. Fluid Interaction with High-resolution Wall-size Displays, In *Proc. UIST'01*.
10. Hopkins, D. The Design and Implementation of Pie-Menus. *Dr. Dobb's Journa*l, 16(12), pp.16–26.
11. Kabbash, P., Buxton, W., and Sellen, A. Two-handed Input in a Compound Task. *In Proc. CHI'94*, pp. 417-423.
12. Kurtenbach, G., and Buxton, W. The Limits of Expert Performance Using Hierarchic Marking Menus. *In Proc. INTERCHI'93*, pp. 482-487.
13. Kurtenbach, G. *The Design and Evaluation of Marking-Menu*s. PhD thesis, University of Toronto, 1993.
14. Momenta User's Reference Manual. Momenta.
15. Palm OS. http://www.palmos.com/
16. Perlin, K. QuikWriting: Continuous Stylus-Based Text Entry. *In Proc. UIST '98*, pp.215-216.
17. Van Sommers, P. Drawing and cognition: descriptive and experimental studies of graphic production process. Cambridge University Press, 1984.