Scale Independence in Marking Menus

Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Don Almeida
Alias
210 King Street East
Toronto Ontario Canada M5A 1J7
{gkurtenbach, gfitzmaurice, akhan, dalmeida} @alias.com

Abstract

This paper describes how marking menus use an explicit recognizer to support menu selection using marks. A key property of the recognizer is scale independence: recognition of marks drawn at any size. We hypothesize that scale independence contributes to fast selections and reduces the effort required. To test this hypothesis, we present an experiment to examine the effects of scale on mark drawing. The results show that smaller marks can be performed more quickly than larger marks but subjects also involuntarily vary the size of the mark, thus giving evidence of the benefits of supporting scale independence. We also present a novel metric for measuring the effort in drawing a mark based on cursor movement dynamics. While this metric is arguably a crude approximation, the experimental analysis agrees with subjective observations.

Keywords

Pop-up menus, marking menus, radial menus, pie menus, gesture recognition, scale independence, accelerators

INTRODUCTION

It is well established that radial menus[1]—pop-up menus based on placing the menu-items in a circle around the cursor—has selection time performance advantages over linear menus [2][7]. This advantage is due to the radial layout of menus items making all menu-items the same, short distance from the cursor.

Marking menus [7] are a combination of pop-up radial menus and gesture recognition (see Figure 1). Marking menus provide two modes of user interaction. If a user presses and holds still for a fraction of a second[2] they enter a "menu-mode" where radial menus pop up and items can be selected in a manner similar to common linear pop-up menus. If the user does not hold but begins dragging immediately, they enter "mark mode" where the cursor leaves an ink trail. In this mode, a selection can be performed by drawing a mark that corresponds to a particular selection path through a menu.

Mark mode is unique in that it utilizes shape recognition techniques to determine a selection. By using shape recognition, a mark only needs to be a shape identifying a particular menu selection not an exact reproduction of how a user would move if they were selecting in menu mode.

In this paper we present the rationale behind this shape recognition approach, describe how the shape recognition is performed, and present an experiment to examine the advantage of this approach.

PREVIOUS WORK

The earliest known work on radial menus dates back to 1969 [11] and is further documented in the popular textbook from 1979 on computer graphics by Newman and Sproull [12]. This work was the first working version of a radial menus system and documented the time-motion advantage of displaying items surrounding the cursor as opposed to displaying item in a linear list. Later work was performed at the University of Waterloo in 1983 experimenting with using pie style menus in a 3D computer graphics modeling system [3]. In 1987

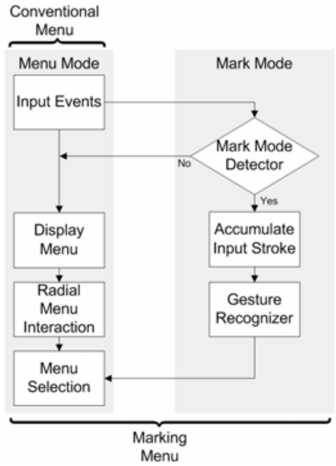

Figure 1: In contrast to conventional pop-up menu systems, marking menus utilize an additional gesture recognizer as shown in the Mark Mode track.

---

[1] We use the term radial menus to describe the general class of menus where items are positioned in a circle around the cursor. This class has many variants: pie menus [5], circular menus [11], oval menus[9], control menus[10], pizza menus, etc

[2] We have found that the hold time can be surprisingly short. Based on our experience of usage in many commercial applications hold time can be as small as 0.15 secs.

researchers at the University of Maryland began investigating radial menus, also exploring many design variations that they named “pie menus” because of their pie chart display style. The work of Callahan et. al. [2] was the first empirical comparison of linear menus and pie menus, which showed that, for single level menus, pie menus were about 15% faster and significantly reduced selection errors.

In terms of the use of gestures, a key observation reported in Callahan et. al. [2] was that items could be selected very quickly because selection could be performed without looking at the menu since direction of movement distinguishes an item. Further work by Hopkins [4, 5] reported that as an artifact of operating system input event buffering, a user could make the mouse movement needed to select from a hierarchic pie menu without having to wait for the system to display each menu. Hopkins believed that this resulted in faster menu selection times (especially when the display of menus is slow). Later work by Hopkins formalized this “mouse ahead” concept by explicitly suppressing the display of menus while dragging.

Kurtenbach [6] further refined the “mouse ahead” artifact by introducing the concept that cursor movement during selection generates a path that identifies a particular menu item. Furthermore, if the cursor left an ink trail when generating one of these paths, a mark that identifies a menu item is created. Given the notion of a mark Kurtenbach then introduced the notion of *scale independence*: a mark can be draw at any size and only the shape of the mark, not its size, identifies the menu-item being selected. Kurtenbach believed that scale independence was a critical property in allowing very fast selection. In earlier pie menu systems, the user, even when selecting using “mouse ahead” still had to be aware of the size of the menus and carefully control their movement. Kurtenbach believed that with this constraint removed, menu selection could be performed even faster and this effect would even be more pronounced in hierarchic menus. Other researchers have since begun utilizing the concept of scale independence [13, 15]. Our work differs from previous work by documenting the shape recognition technique used in marking menus and provides empirical evidence of the advantages of this approach.

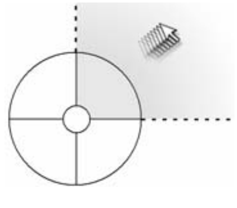

Figure 2: Single level radial menus can allow scale independent gesture through menu-item wedges having infinite size.

Scale Independence

The most important concept in motivating the use of shape recognition is scale independence. Radial layout results in interactive menus where selection is essentially based on direction of movement. A key observation is that direction of movement has the property of scale independence: a movement of any size identifies a menu-item. Thus selection can be performed quickly by either a movement over a short distance, or fast movement over a long distance. Most importantly, the user does not have to monitor or care about size of movement. In effect, movements are ballistic. In contrast, selection by distance of movement, used in linear menus, is inherently size sensitive or scale dependent—a user must not only control the direction they move the cursor, but also the exact end point of that movement, in effect the “scale” of their movement. Thus it can be argued that scale independence is an important property for a radial menu system that aids the selection time advantage of radial menus over linear menus.

Scale Independence in Menu Mode

Can scale independence be attained without the need for explicit gesture recognition? Specifically, can menu mode (interaction with posted menus) be designed to work with scale independent movements so that a simple mouse-ahead technique can be used instead of explicit gesture recognition? For single level or non-hierarchic radial menus, scale independence can be attained by implementing the active area for a menu item as an infinite size “wedge” (see Figure 2). In this case a user can make a movement of any size, just as long as it is bigger than the center “no selection” zone at the center of the menu.

However, attaining scale independence for hierarchic menus is difficult. Ultimately the user needs some way to signal to the system when they are done selecting from the current menu and wish to move onto a submenu. Typically in linear menu systems, and in some radial menu systems, dwell events are used to trigger a submenu display. This technique is used in marking menu’s menu mode and it results in allowing movement through the menus that is scale independent. The drawback of this

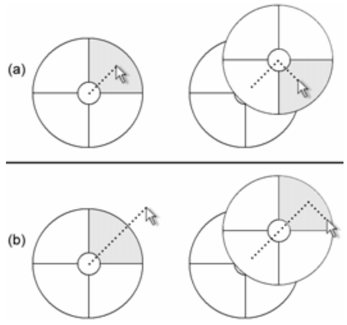

Figure 3: (a) The “A” shaped gesture selects the South-East submenu item. (b) The same shape gesture at a different scale selects a different item.

behavior is that articulation of dwell events slows the user's gesture and therefore cannot be used as a "quick flick" style of gesture input.

There are numerous design variations for radial menu systems and a comprehensive description of the design space is given [7]. The most common method for signaling submenu display is the dwelling technique discussed above or some form of *boundary crossing*. Typically a boundary crossing technique involves displaying a submenu when the cursor is moved close to or past the edge of the current menu. Obviously this results in a scheme where movement through the menus is scale dependent. As shown in Figure 3, changing the size of the movement with a boundary-crossing scheme can produce unintended selections.

A design variation on the boundary crossing technique is used by a pie menu technique called "radial-context", a plug-in for the Mozilla browser [14]. With this design, when the outside border of the parent menu is crossed, like a boundary crossing technique, the submenu is displayed centered at the crossing point. The problem of operating the submenu from a fixed location, shown in Figure 3, is avoided by allowing the submenu to move with the cursor as the borders of the submenu are crossed. This scheme however does not result in complete scale independence. Movements must be at least large enough to cross menu borders thus making it impossible to select from a menu with a movement that is less than the radius of the displayed menu. Also it is not clear how this technique would be applied to menus with more than two levels of submenus without becoming highly scale dependent.

In summary, we know of no design variations of radial menu systems that support truly scale independent movement during menu display. Thus with these types of systems the user must be careful when using mouse-ahead to make movements of a particular size since the input is, or will be, processed in the same manner as if the menus were being displayed.

Marking menus have the potential advantage of having a separate and explicit mark mode which can be optimized to support fully scale independent fast gestural input. Menu mode is somewhat scale independent but not as fast as mark input since it requires the use of dwell events. Nevertheless, menu mode serves nicely as a rehearsal mechanism (albeit slower because of the dwells) for the gesture needed to operate in mark mode.

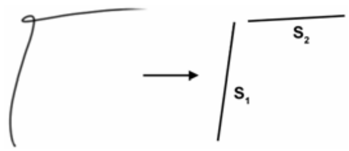

Figure 4: The marking menu recognizer decomposes an input stroke into a series of line segments, s1 and s2, representing selections from a series of radial menus.

Finally, there are other drawbacks in trying to support quick menu selection without scale independence and an explicit mark mode. For example, in the course of doing this research we produced what we thought would be a very typical implementation of a hierarchic radial menu system that did not have scale independence and explicit mark mode. A pie menu style of display was used that did not use the infinite wedge technique shown in Figure 2 and a boundary crossing technique was used to trigger submenu display. We observed that while it was possible to quickly select with gestures from these menus, it was visually irritating to monitor your path through the menus as they were quickly displayed and undisplayed. Furthermore it required careful monitoring of the size of movement in comparison to gesturing with marking menu marks. In contrast, the explicit, stationary, and variable size of the mark of marking menus seemed much less taxing.

Mark Recognition Technique

While the mark recognition is conceptually similar to character recognition, the approach used is very different from typical character recognition systems that match an input symbol against a library of possible symbol types. Essentially, our recognition system divides an input stroke into a series of straight-line segments where each segment corresponds to a menu or submenu selection (Figure 4).

The recognition technique has two major steps. First, the points where the stroke changes direction to indicate a submenu selection (called articulation points) are determined and used to divide the input stroke into a series of line segments. Next, these segments are mapped into a series of menu selections from the menu tree.

Figure 5 shows the basic approach of looking for local maximal angle changes along the input stroke. Essentially the poly-line ABC is used as a virtual protractor that slides along the input stroke. When ABC is positioned at a particular location on the input stroke, point B is varied from L_{AB} to L_{BC} to find the maximum change of angle over this range. Specifically, the

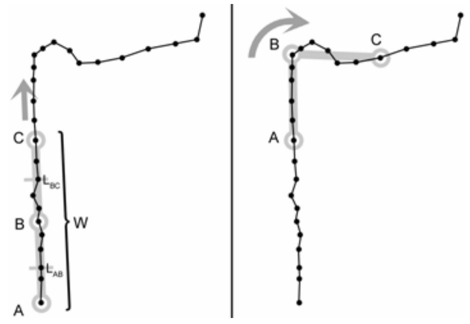

Figure 5: The recognizer works by finding the local maximum angle change along the input stroke. A window (minimum length of w), denoted by line AC, slides along the input points and finds the angle at B between L_{AB} and L_{BC} that has the maximum angle change.

recognizer works as follows:

We first calculate an expected segment length E by dividing the length of input stroke by the depth of the menu structure.

$E = \text{length of input stroke} / \text{menu depth}$

Other values used are:

$W = E * 0.3$

$\text{Sensitivity} = 0.75$

$M = \text{max. breadth of menu}$

$\text{AngleThreshold} = 360 / M / 2 / \text{Sensitivity}$

Next the algorithm to determine the articulation points is performed:

```
A = start of stroke point.
Loop:
    C = the first point at least distance W from A.
    If C cannot be found,
        Exit loop.
    $L_{AB}$ = the first point at least distance W / 8 from A
    $L_{BC}$ = the first point at least distance W / 8 from C
    Find a point B between $L_{AB}$ and $L_{BC}$ that is the
    maximum angle between ABC.
    If maximum angle > angleThreshold
        Save B as an articulation point.
        A = B.
    Else
        A = next point after A.
```

Next the first point of the stroke and last point of the stroke are added to the list of articulation points and from this a list of connected line segments is created. Next, any line segments that are shorter than 1/3 E are deleted.

The final step is to produce a menu selection by mapping the first point of each line segment to the center of a menu. The angle of the line segment then determines which item is selected. This process is repeated down the menu hierarchy.

Resolving ambiguous marks

Our recognition engine not only deals with situations where too many line segments are generated for an input stroke but also for the opposite situation of too few line segments being generated. For example, suppose a user makes a straight line mark to select from a two-level menu (see Figure 6). Since the mark is segmented into a single line, the above recognition process maps this to a selection from the first level menu but does not have another line to map to the second level menu.

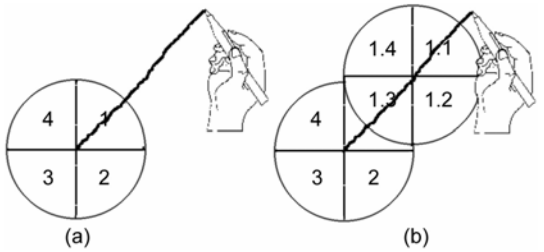

Figure 6: Two possible interpretations of mark. (a) the interpretation of selected menu item 1. (b) an interpretation of selecting item 1.1.

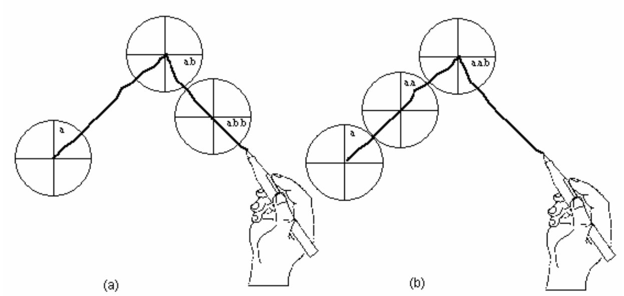

Figure 7: Selection ambiguity when menu hierarchies are greater that 2 levels deep.

From the range of approaches to solve this problem, reported in [7], we have adopted the technique that is based on the assumption that a user does not need to select non-terminal menu items. Specifically, a mark is interpreted as a path that goes as deep as possible into a menu tree. The algorithm to support this involves sub-dividing the input line until the number of lines is equal to the depth of the menu structure, thus ensuring a line segment for each level of the menu.

For two level menus, this approach is sufficient. However, if menus are three levels deep, or more, ambiguity can still occur (see Figure 7). A technique could be used where line segment lengths are assumed to correspond to the radius of the radial menu (boundary crossing described earlier). However, this results in the interpretation of a mark not being scale independent. To ensure scale independence, we utilize a technique which assumes if a stroke is of length L, and the menu structure is of depth D, then the line segments should be approximately L/D in length. This results in long line segments being divided such that all the resulting line segments in a mark are approximately the same length.

Our recognition approach has several limitations. First, it fails to produce any articulation points if a corner of a mark is very smoothly curved (Figure 8a). Second, the recognition technique is based on using local information to find turns in the stroke so it fails to interpret larger patterns correctly (Figure 8b). Additional improvements in stroke decomposition might be attained by utilizing information about each individual submenu's breadth and

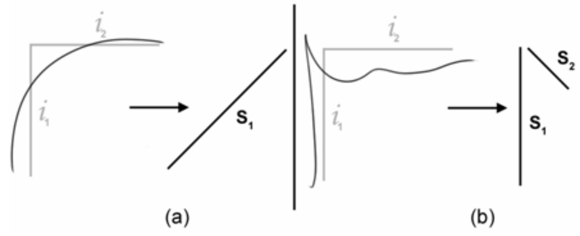

Figure 8: The recognizer fails when (a) there are no distinct local corners or (b) global features are not considered. $i_{1,2}$ segments show the intended shape while $S_{1,2}$ show the recognizer results.

depth as opposed to the current method using only maximum menu breadth and maximum depth of the entire menu structure.

Experiment

The recognition system used by marking menus is an additional implementation cost and complication when compared to a menu system which simple uses "mouse ahead" to support gesture input. The question is: is this cost worth the benefit? To begin to understand if a real benefit exists we conducted an experiment to study the effects of drawing marks at different sizes on human performance. Our own personal experiences with using marking menu marks was that smaller marks seemed faster especially with a pen input device. This was reinforced by the logic that smaller marks required less distance to be traveled and therefore could be performed more quickly. This rationale seemed to hold for mouse input but perhaps not as strongly.

Given these observations we formed the following hypothesis:

H1: As the length of a mark increases, time to draw the mark will increase.

H2: Marks can be performed more quickly with the pen than that mouse.

Design

We recruited 8 subjects for the experiment. All were right handed males between ages 20 to 30, experienced computer users with vary degrees of experience using pen-based systems.

Like other experiments with marking menu marks [8], the task was constructed to simulate expert performance of drawing marks. Trials were blocked by mark shape and size such that a subject would repeat drawing the same mark 10 times in a row thus producing a very practiced articulation of the mark to simulate expert performance. A block of trials worked as follows. The subject was shown a "target mark" of particular size and shape on the left size of the display monitor. The subject's job was to draw the same mark on the right side of the monitor at approximately the same size ten times. A drawn mark would remain on the screen until the user began drawing the next mark in the block of trials. When a subject completed a block, a dialog box would be displayed to inform them they were being given another target mark.

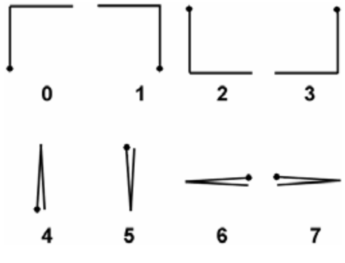

Figure 9. Marking Menu shapes used in experiment

To keep the subject drawing marks close to target size, the system beeped and displayed a warning message when the input mark was less than three-fifths, or greater than five-thirds of the target length. If a subject drew a mark of the wrong shape (or the recognizer did not recognize the right shape) the systems beeped and displayed an error message.

Eight different shapes of marks were used. We limited the types of marks to be shapes from the set of sixteen marks needed for a two level hierarchic menu with four items in the parent and submenus (see Figure 9).

Five different scales of mark were used 1x, 2x, 3x, 4x, and 6x, with a 1x mark having a length on the display of 60 pixels (18mm) and a 6x mark having a length of 360 pixels.

A Wacom PL-550 Cintiq combination display and digitizer tablet was used connected to a dual processor 800MHz PIII Dell Workstation with a Nvidia Quadra FX1000 graphics card. Subjects used the Wacom pen on the PL-550 display to draw marks with the pen directly on the display. To draw marks with a mouse a Logitech Mx500 USB mouse was used with the medium mouse speed setting for Windows 2000 professional. A subject was seated at a table with the display placed on it. The tablet/display was tilted 20 degrees from horizontal to make drawing with the pen comfortable.

A within-subject design was used with each subject using both the pen and mouse as input devices and performing all combinations of mark shape and size. This resulted in 8 subjects X 2 input devices X 8 shapes X 5 size X 10 trials = 6400 data points for the entire experiment. Trails were grouped by input device and counter balanced with ½ of the subjects using the pen first. Within a group, target marks of particular shape and size were chosen randomly without replacement until all combinations were performed.

For every trail we logged the time to draw the mark, specifically, the time from pen down to pen up, the length of the drawn mark, and drawing velocity information (more on this later). Trials were tagged as errors if the wrong shape was recognized.

After completing the experiments subjects were given a short questionnaire to determine their preferred size of mark, input device and any other comments they might have.

Results and Discussion

Mark Performance

Our data confirms the first hypothesis, as the length of a mark increases, time to draw the mark will increase (see Figure 10). An analysis of variance shows that the effect of mark scale is significant $F(4, 28) = 224.5$, $p < 0.0001$. The effect of input device is also significant $F(1,7)=239.1$,

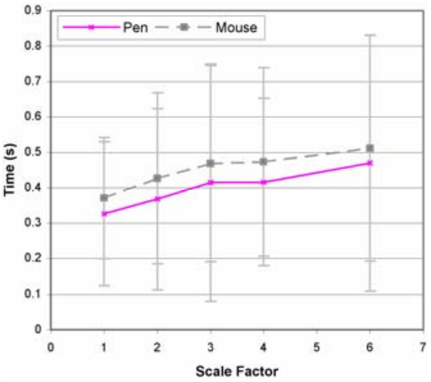

Figure 10. Mean time to draw mark at a given scale.

$p < .0001$ with the data showing that the pen performs better than the mouse (see Figure 10). This confirms our second hypothesis.

To further reveal the effects of our first hypothesis, we refined and regrouped our scale observations. Instead of having the 5 scale factors (1x, 2x, 3x, 4x, 6x), we took all of the data and assigned it a bucket from scale factor 1 to 8. For example, if we requested the subject to articulate a mark of scale 6x but they actually made it at scale 3x, we placed it in bucket 3. An analysis of variance with this new "bucketization" data grouping shows a significant effect for performance $F(8, 56) = 106.8$, $p < 0.0001$. Again, the effect of input device is also significant $F(8,56) = 6.1$, $p < .0001$. Figure 12 shows the results of our "bucketization" for both the pen and mouse. Here you can see a more pronounced performance effect of articulating a 1x mark with the pen (0.33 sec.) compared to an 8x mark (0.53 sec.). Thus a 0.2 second difference was observed with pen users. It is interesting to note that for bucket scale factors 7x an 8x, the pen takes longer to articulate the mark compared to the mouse. We believe the performance difference lies in the fact that the mouse is a relative device compared to the absolute positioning of the pen. Note that for the remaining analysis, we use the original scale factors grouping.

An analysis of variance shows a significant interaction between device and mark shape $F(7, 49) = 7.46$, $p < 0.0001$. We observed and subjects reported that "V" marks were more difficult with the mouse.

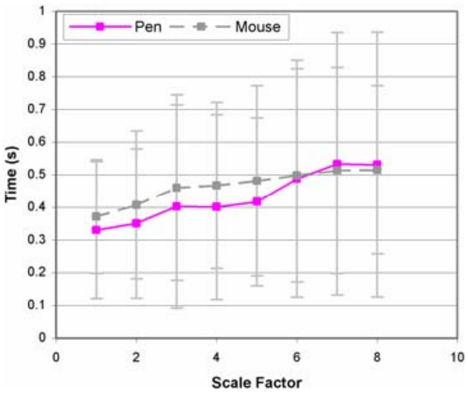

Figure 12. Mean performance time for "bucketized" data based on stroke length drawn.

Error Rates

We consider two types of input errors: scale and shape. *Scale errors* occur when the user draws a mark that is a different size from the scale requested. Scale errors in the input can be taken as instances of other scales and can be regrouped into their actual scale range and have been dealt with earlier using "bucketization". *Shape errors* occur when the recognizer interprets the input stroke to be a different shape from the requested target shape.

Shape error rate was almost significantly affected by scale $F(4,28) = 2.24$ $p < .0626$. We found that the smallest mark had a shape error rate of 4.2% while all the other sizes had about two-thirds that rate, averaging 2.78%.

The device had a significant effect on shape error $F(1,7)=19.46$, $p < 0.0001$. The shape error rate for the pen was lower (2.8%) than the mouse (3.9%). Figure 11 and 13 show shape errors versus experimental factors. The combination of device, scale, and shape significantly affected shape error rate $F(17,119)=5.83$ $p < .0001$. The affect of shape combined with device was significant $F(7,49)=4.15$ $p < .0001$. Figure 13 shows that this effect seems due to the mouse having problems with "V" shaped marks.

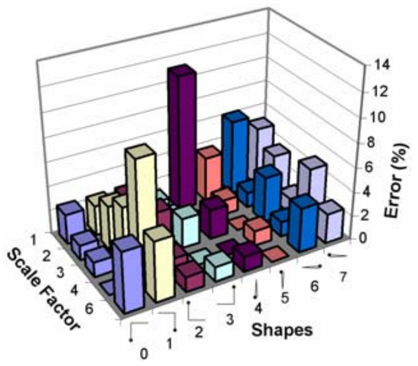

Figure 11: Error rate by shape and scale for the pen. Average is 2.8%.

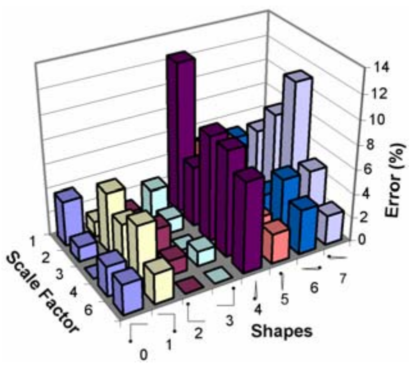

Figure 13 Error rate by shape and scale for the mouse. Average is 3.9%.

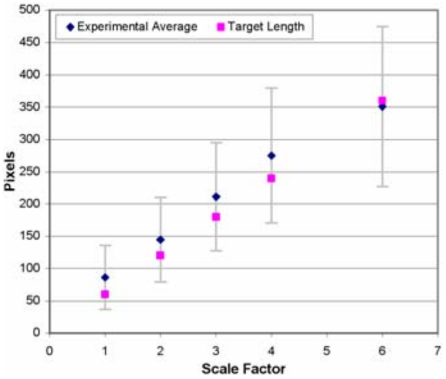

Figure 14: Stroke Deviation.

Scale Deviation
In general, subjects demonstrated appreciable deviation from the requested target stroke length. Figure 14 shows the standard deviation versus scale of mark. Larger marks seem to result in larger deviations than with smaller marks. In general these results indicate that allowing for marks of varying size will be a good strategy.

Stroke Work
One result that we found surprising was that while the effect of scale is significant, the absolute difference between the time to draw a 1x mark and an 8x mark was smaller than we anticipated (0.2 second). Logically, one would estimate that the relationship between length of mark and time to draw the mark would be linearly related, for example, a 6x mark would take about 6 times as long as a 1x mark. However, as the data shows this was not the case. Initially in our pilot tests of the experiment this result surprised us because the small marks seemed so much better to us. Moreover, if we considered what type of mark we would prefer if we had to make the mark all day long, it was unanimous that the smaller mark was preferred because it required less effort. Thus we developed the hypothesis that while the small marks and large marks were very close in articulation time, there was a significant difference in effort involved.

With this in mind we embarked on trying to define some metric for effort. In our own use of marks we observed that large marks could be drawn almost as quickly as small mark by increasing physical effort which in turn allowed a user to travel the distance of the longer mark in close to the same amount of time as a smaller mark. Essentially, large marks seemed to require more work.

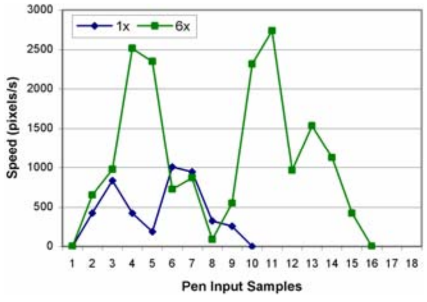

Figure 15. Stroke velocity at 1X and 6X scale.

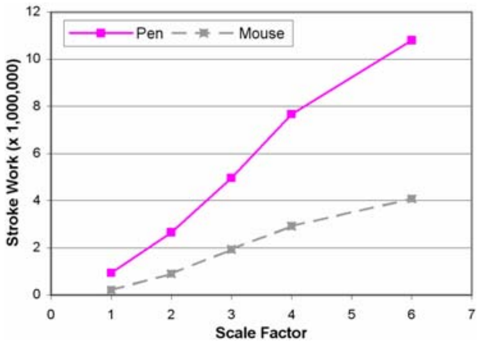

Figure 16. Stroke work increases as scale increases.

Given this observation we decided to try to apply Newtonian mechanics to mark drawing and consider the pen as a mass that is being accelerated and decelerated over time. Figure 15 shows a graphic of pen velocity versus time for a small mark and a large mark. Note that there are much larger changes in velocity when drawing the large mark.

Furthermore, we speculate that a rough estimate of the effort in drawing a mark can be calculated by applying the Newtonian definition of work where work = force over a distance (where the forces are what the user applies to accelerate and decelerate the pen). Since force is proportional to changes in velocity over time, from the type of data shown in Figure 15 we can derive a force measure for each input event. A measure of the work done between two input events can be calculated if the force measure for the first input event is multiplied by the distance traveled between input events. Repeating this calculation for all input events in a stroke results in a measure of the total work performed for a stroke.

Figure 16 shows the application of this calculation to the experimental data. We conjecture that in general it shows that subjects indeed increased their effort with the larger marks to attain faster performance.

Subjective Preference
In our questionnaire, subjects were asked, assuming they were using mark shapes to issue commands in an application, what size of mark they would prefer to draw if they had to do it frequently with a pen, and with a recognizer performing accurately. They indicated their preference by choosing from a listing of the 5 sizes of marks used in the experiment. Most subjects preferred smaller marks: (out of 8 subjects) 3 preferred scale 2, 3 preferred scale 3, and scale 4 and 6 were each preferred by a subject. We suspected that despite asking them to assume a perfect recognizer subjects did not pick the smallest mark since this size had the highest recognition error rate. Subjects strongly preferred the pen over the mouse when asked which required less effort. On a scale from 1 (strongly disagree) to 7 (strongly agree) the average score was 6.4, with 5 being the lowest rating.

The last question was an open-ended request for comments. Two subjects commented that small marks were hard to get right and required more precision. One subject noted that upward movements with the mouse felt awkward. Two subjects had RSI concerns and thought that larger marks would reduce this risk.

Both the experimental data and subjective data indicate that there is a lower bound on the size of marks. Specifically, in the experiment, the smallest scale marks were tending towards higher error rates. It could be argued that these marks were so small that articulation of them was difficult, however, we suspect that this size was not approaching the limits of human articulation (the smallest mark was 9mm in height still very easy to draw with a pencil). We suspect that digitizer accuracy and shape recognizer accuracy were the true source of errors for the smallest marks. With very small and extremely quick marks the digitizer tablet has problems recognizing true pen down events. This results in noisy input that in turn creates problems for the shape recognizer. Refinements to the recognizer to address these problems are a topic for future research.

While our metric to measure effort (Stroke Work) may be speculative we believe there is some merit in this analysis. It definitely conforms to our observations that larger scale marks seemed more effortful than small ones. However, note that Figure 16 shows the mouse requiring less effort than the pen and that this is contrary to comments from subjects. We believe this anomaly is partly due to our assumption that the mass of the pen and mouse are equal in our Newtonian work calculations. It can also be argued that the additional effort applied in controlling the mouse is not converted as efficiently into stroke work as with the pen. Ultimately, our stroke work metric does not capture the different muscle groups used to draw small marks (the fingers) and large marks (wrist and arm) and the true energy used [1]. Nor does it capture cognitive aspects on monitoring movement and drawing. Taking these factors into account could result in the more realistic measure of effort and is a topic for future research.

CONCLUSIONS & FUTURE WORK

In this paper we have three main contributions. First, we present the description of the specialized recognition algorithm used by marking menus (so other non-menu researchers can easily implement the technique). In particular, we discuss how users may benefit from using gesture recognition as opposed to simple pie menus with mouse ahead. Second, a full experiment was conducted to quantify the nature of the effect of scale independence on performance. Third, we defined a new metric for measuring the effort in drawing a mark based on cursor movement dynamics.

At the beginning of this paper we argued that scale independence would significantly contribute to better selection performance. The results from the experiment provide evidence of this. Small marks can be performed significantly faster than large marks and are also subjectively preferred by users. In addition, the results indicate that a gesture input scheme that supports marks of varying size will suit users' behavior: The experimental data showed that users involuntarily vary the size of their marks and the subjective data indicates a range of mark sizes is preferred, although there is a tendency to prefer smaller marks.

The concept of scale independence is not specific to marking menus. Indeed the concept can be applied to any type of motor movement involved in human computer interaction. For example, Zhai states that his research of marking over a graphical keyboard [13] was inspired by marking menus and the issue of scale independence. Gesture scale independence issues and gesture effort are important characteristics for designers to consider when building interactive systems that work across different display and input scale form factors ranging from PDAs, tabletPCs and large whiteboard displays. The upshot is that the results in this paper shed light on a fundamental aspect of input in human computer interaction.

REFERENCES

1. Balakrishnan R., & MacKenzie I. S., (1997). Performance differences in the fingers, wrist, and forearm in computer input control. *Proceedings of CHI 1997*, 303-310, ACM New York
2. Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988) An empirical comparison of pie vs. linear menus. *Proceedings of the CHI '88*, 95-100, New York: ACM.
3. Forsey D., (1984) Harmony in Transposition: A Tocatta for Vax and M68000, *M.Sc. Thesis, University of Waterloo, Canada*
4. Hopkins, D. (1987) Direction selection is easy as pie menus! *login: The USENIX Association Newsletter*, 12(5), 31-32.
5. Hopkins, D. (1991) The design and implementation of pie menus. *Dr. Dobb's Journal*, 16(12), 16-26.
6. Kurtenbach, G. & Buxton W. (1991) Issues in combining marking and direct manipulation techniques. *Proceedings of UIST '91 Conference*, 137-144, New York: ACM.
7. Kurtenbach, G. (1993) The Design and Evaluation of Marking Menus. *Ph.D. Thesis University of Toronto*.
8. Kurtenbach, G. & Buxton, W. (1993) The limits of expert performance using hierarchical marking menus. *Proceedings of the CHI '93*, New York: ACM.
9. Lyons P.J., Pitchforth M., Page D., Given T. & Apperley M.D., (1996) The Oval Menu - Evolution and Evaluation of a Widget *Proceeding of the OZCHI '96 Conference*, 252-259, IEEE Computer Society Press
10. Pook S., Lecolinet E., Vaysseix G., & Barillot E., (2000) Control Menus: Execution and Control in a Single Interactor, *CHI 2000 Extended Abstracts*, 262-264, ACM Press
11. Wiseman, N. E., Lemke, H. U., & Hiles, J. O. (1969) PIXIE: A New Approach to Graphical Man-machine Communication. *Proceedings of 1969 CAD Conference Southhampton*, 463, IEEE Conference Publication 51.
12. Newman, W. M. and Sproull, R. F., (1979) *Principles of Interactive Computer Graphics*, (2nd ed.) pages 224-225, McGraw-Hill, New York
13. Zhai S. & Kristensson P. (2003) Shorthand Writing on Stylus Keyboard. *Proceedings of CHI 2003*, 97-104, ACM New York
14. http://www.radialthinking.de/radialcontext/
15. Zhao, S. and Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. *Proceedings of UIST 2004*, 33-42.