Analysing Mouse and Pen Flick Gestures

Michael Moyle and Andy Cockburn
Human-Computer Interaction Lab
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
+64 3 364 2362 http://www.cosc.canterbury.ac.nz/
{mjm184, andy}@cosc.canterbury.ac.nz

ABSTRACT

Gesture based interfaces promise to increase the efficiency of user input, particularly in mobile computing where standard input devices such as the mouse and keyboard are impractical. This paper describes an investigation into the low-level physical properties of linear 'flick' gestures that users create using mouse and pen input devices. The study was motivated by our need to determine sensible constraints on values such as the magnitude, timing, and angular accuracy of gestures for a marking-menu implementation. The results show that pen gestures are substantially larger than mouse gestures, that angular errors are larger in the left and right directions with the pen, that vertical gestures are 'awkward' with the mouse, and that downwards gestures are approximately 11% slower than other directions.

Keywords

Gesture interfaces, marking menus, 'flick' gestures, pen computing, evaluation.

INTRODUCTION

Gesture based input mechanisms promise two major user interface benefits. First, they can reduce the time taken to issue simple commands. Gesture based 'marking menus' [8], for instance, reduce the Fitts' law [5] time-to-target constraints of normal menus. They do so by allowing users to select menu items by gesturing towards each item's location in a 'pie menu' centred on the user's cursor location. Second, gesture based input methods are readily implemented on mobile devices where mice and keyboards are impractical.

Gesture-based interface mechanisms are becoming increasingly common in commercial systems. Desktop interfaces include the 'Opera' web browser[1], the Alias Wavefront graphics package 'T3' [10], and 'Sensiva'[2] which adds gestural commands to many commercial desktop environments. Gestural systems on mobile devices include the Unistrokes [6] and Graffiti [1] text entry systems.

In order to distinguish between different gestural commands and between other mouse-driven actions, gesture recognition software must set constraints on the timing, direction, and magnitude of gestures. Users must then learn to generate gestures within these constraints. Ideally, however, the software should be designed around the 'natural' properties of human gestural input.

The experiments described in this paper investigate the natural properties of linear 'flick' gestures, such as those used in marking menus. The aim is to empirically determine values that can be used in the design of improved gesture recognition systems. In particular, we investigate three measures: the magnitude (or length) of gestures, the time taken to produce gestures, and the angular error of gestures. These values are determined across two different settings for mouse velocity/acceleration and across mouse and stylus input devices.

The next section describes related work. We then present our experimental design, followed by the results, discussion and conclusions.

BACKGROUND

Marking Menus

Marking menus are a specialisation of pie-menus. Pie menus [2] are pop-up menus that appear immediately under the user's cursor when the mouse button is pressed. The user selects items by dragging the cursor into the appropriate segment of the pie. The motivation for pie-menus is to minimise Fitts' law constraints on time-to-target—in theory, a movement of one-pixel is sufficient to reach any of the menu items, and further movements result in the target effectively becoming larger.

Marking menus [8,9,11] extend the pie-menu concept by allowing users to select items before the menu appears. Expert users can select items with a rapid 'flick' in the appropriate direction. If the user hesitates in their gesture (a delay of more than approximately half a second) then the pie-menu is displayed to assist learning the gesture set.

[1] www.opera.com

[2] www.sensiva.com

Kurtenbach and Buxton [8] found that marking menus were heavily used once users learned the location of commands on the menu. It has also been shown that performance with marking menus deteriorates as the number of items in the menu increases [11].

Early mouse-based marking menus systems used the left mouse button to issue gesture commands. Recent systems such as Opera, however, use the right mouse button to reduce the problems of overloading the interface semantics associated with each button—dragging the left button is normally used for text selection, but right-button dragging is rarely used.

The evaluations reported in this paper include an examination of the differences between gestures that are created using the left and right mouse-buttons.

Non-Linear Gesture Input Schemes

In selecting one item from a single marking menu, the recognition software need only compare the total distance traveled on the X and Y coordinates to determine the direction of the gesture. By extending the marking menu concept to cascading menus, the user can access exponentially larger sets of menu items through gestures that consist of a series of linear edges.

Sophisticated recognition software can distinguish between large numbers of differently shaped gestures. The Unistrokes gestural alphabet, for instance, allows users to express all letters in the Roman alphabet with gestures [6]. Several other character sets have been implemented using similar gesture techniques, for example, T-CUBE [14] and Graffiti [1].

Beyond text input, non-linear gestural input has been used for a wide range of application areas including air traffic control [3]. The GRANDMA toolkit allows gesture recognition to be added to interfaces by having the system developer provide examples of gestures and their associated interface actions [13].

Gesture Studies

Dulberg, Amant and Zettlemoyer [4] compared simple ‘flick’ gestures with normal button clicks and keyboard shortcuts. In tasks that involved flicking towards abstract targets, they showed gestures to be 26% faster than button selection, but not reliably faster than key-bindings. Users also found the gestures easy to learn and accurate with only 4 errors from 3300 trials. In their six-subject informal study of flick gestures for redirecting keyboard focus to items on the Microsoft Windows desktop, subjects reported no problems with learning the technique, and five of the six participants said they would use it if available.

Goldberg and Richardson [6] measured the median times that a single subject took to input various characters using Unistrokes. They found the dot gesture used to specify a space was the fastest at 90 milliseconds, and that the ‘α’ shaped gesture used for ‘q’ was the slowest at approximately 330 milliseconds. Overall, the median stroke time was approximately 150 milliseconds. These values result from observations of only a single user. It remains unclear how these values will differ between users.

Isokoski [7] describes a model for predicting writing time in Unistroke-like gesture alphabets. The motivation is to predict expert performance with gesture alphabets without the necessity of conducting time-consuming empirical studies. Expert performance is particularly hard to empirically evaluate due to the extensive training required.

Mouse Acceleration

Prior to reporting our evaluation, it is necessary to summarise some low-level details of mouse operation.

Mouse motion is normally controlled by one or two user-configurable parameters that determine the mapping (‘control-display gain’ [12]) between movement of the physical mouse and the corresponding movement of the cursor on the screen. These values are normally termed ‘acceleration’ and ‘threshold’. When the mouse moves slowly, a base mapping between physical mouse-motion and cursor movement applies. Normally the default value for base movement is approximately one to four, meaning that the cursor moves four centimeters for each centimeter of physical mouse motion. The acceleration setting determines the maximum mapping between mouse movement and screen distance. This mapping applies during rapid mouse movement, and the normal default value is approximately double the base value, meaning that during rapid motion, each centimeter of mouse motion causes the cursor to move approximately 8cm. The threshold value determines the mouse-movement rate (distance per unit time) that must be reached before the accelerated mouse mapping applies.

The experiments described in this paper were conducted using both accelerated and non-accelerated mouse mappings. Furthermore, in order to aid the generality of the results, we translate screen pixel distances (which vary across hardware platforms) into physical measurements at the mouse.

EVALUATION

Twenty-nine subjects, all right-handed post-graduate Computer Science students, participated in the study. Although these subjects have substantially more computing experience than most, we believe that their general motor skills will be similar to other subject groups.

The subjects were assigned to one of three gesture-input conditions:

1. *Mouse input, no acceleration*. All gestures were created with mouse acceleration turned off (constant control-display gain), providing a constant linear mapping between physical movement of the mouse and corresponding cursor movement.
2. *Pen input*. Gestures were created using a pressure sensitive pen-computer. There was a one-to-one

mapping between physical movement of the pen on the screen surface and the resultant gesture size.

3. *Two-to-one mouse acceleration input, threshold 4*. The gestures were created using a common default setting for mouse motion, with an accelerated mapping termed "two-to-one", and a threshold setting of 'four'. The "two-to-one" setting results in an approximately eight-to-one mapping between physical and cursor motion.

Procedure

The experimental procedure was the same in each of the three conditions. The subjects were informed that we were interested in the natural properties of flick gestures with the mouse (or with the pen). Mouse users were told that 'flicks are quick motions with the mouse, during which the mouse button is pressed and released'. Pen users were told that 'flicks are a quick motion with the pen'. Pen gestures do not require the stylus button to be pressed.

The subjects were asked to practice issuing several gestures in each direction prior to beginning the experiment. Through the experiment, the subjects were asked to generate either twenty-five or fifty gestures in each of four directions: up, right, down, and left. Subjects assigned to one of the two mouse conditions were asked to repeat the set of gestures using the right mouse button (the first set was issued using the left button).

The twenty-five or fifty repetitions in four directions gave us either one- or two-hundred gestures for each pen user, and either two- or four-hundred gestures for each mouse user (half each with the left and right buttons). We changed from fifty to twenty-five gestures approximately half way through the study because some of the subjects mentioned that they started to feel repetitive strain problems with 'stiffness' and 'cramping' in their hands and wrists.

The reduction in number of gestures has a minimal impact on our data analysis, because only one mean sample time in each direction is calculated per user.

Apparatus

All of the gestures were issued in a large window (500x400 pixels) created by a Tcl/Tk program. The program was equipped to log coordinates and time when the gesture began and finished. For mouse actions, the gesture began when the left/right mouse button was pressed, and finished when the button was released. For pen gestures, the gesture began when the mouse touched the screen surface and finished when it left the surface.

The mouse experiments were run on a 32x24cm display, with a 1280x1024-pixel resolution, giving 40 horizontal pixels per cm. The pen experiments ran on a stylus computer with a 19x14.25cm display, running at a 640x480-pixel resolution, giving 33.7 horizontal pixels per cm. The pen computer only allowed time to be measured to the nearest 55 milliseconds. For this reason, time information for the pen gestures was discarded.

The number of pixels per centimeter for each device allows us to translate the logged magnitude of gestures, which were logged in display pixels, to millimeter motions of the physical device.

Experimental Design

The logged data was analysed in three different experimental designs, described below. The three dependent variables measured were as follows:

Gesture magnitude—the distance between the physical location of the mouse or pen when the gesture begins and when it finishes. The data values were logged in pixel coordinates, but were also translated into the corresponding millimeter motion values at the physical device.

Gesture timing—the time, measured in milliseconds, between the start and finish of each gesture. Timing values were not measured for the pen condition in experiment one because of the low timing granularity supported by the pen computer.

Angular error—the per-gesture offset between the intended gesture direction (up, right, down or left) and the actual direction. Measuring angular error allows us to detect stereotypical biases towards particular angular errors for each gesture direction for each input device.

Experiment one

The first experiment examines gesture magnitude and angular error in a mixed 2x4 analysis of variance (ANOVA) for factors 'input device' and 'gesture direction'. Input device is a between-subjects factor with two levels: pen (stylus) and non-accelerating mouse. 'Gesture direction' is a within-subjects factor with four levels: up, right, down, and left. This experiment allows us to examine the differences between pen and mouse gestures, and between gestures in different directions. If reliable differences exist between input devices or between directions, then gesture recognition software should account for them.

Experiment two

The second experiment examines the three dependent variables—magnitude, time, and angular error—across gestures created with the left and right mouse-buttons. A non-accelerating mouse was used. The design is a repeated measures 2x4 ANOVA, with the second factor enabling us to inspect differences between gesture directions. This experiment allows us to detect differences between gestures created with the left and right mouse buttons.

Experiment three

The third experiment compares gestures created with a non-accelerating mouse with those created with a 'standard' acceleration setting. The experimental design is a 2x4 mixed ANOVA for factors 'mouse acceleration' (zero or two-to-one) and 'gesture direction'.

RESULTS

All subjects completed the full set of gestures extremely rapidly, with all training and two hundred gestures typically

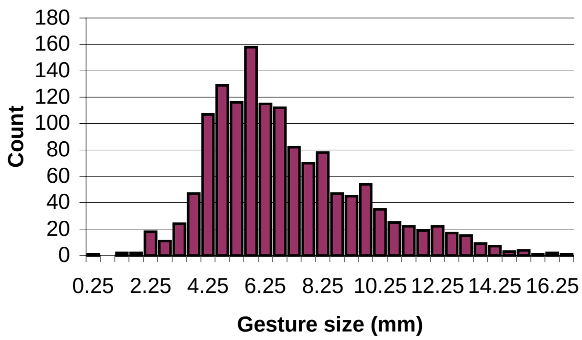


Figure 1: Size distribution of 1400 non-accelerating mouse gestures (half-millimeter increments).

taking less than five minutes to complete. Considering that the subjects were given minimal training and instruction, there was surprisingly little variation in the magnitude, timing, and angular error of gestures produced by each subject and between subjects. Across the total 5200-gesture set, the mean gesture size (distance from mouse-button/pen down position to mouse-button/pen up position) was 6.6mm (σ 3.2), the mean gesture time was 151 milliseconds (σ 53), and the mean angular error was 4.2 degrees (σ 5.3). Figure 1 shows the distribution of gesture magnitudes, segregated into half-millimeter intervals, across 1400 gestures created using a non-accelerating mouse. The relatively normal shape of this graph is typical of the data gathered for each of the three dependent variables.

As mentioned earlier, the subjects began to suffer repetitive strain symptoms very rapidly. We doubt that commercial use of gesture commands will cause similar problems because the experiment required the subjects to generate artificially large numbers of gestures in an extremely short period of time. It is unlikely that commercial use would require equivalently dense patterns of use.

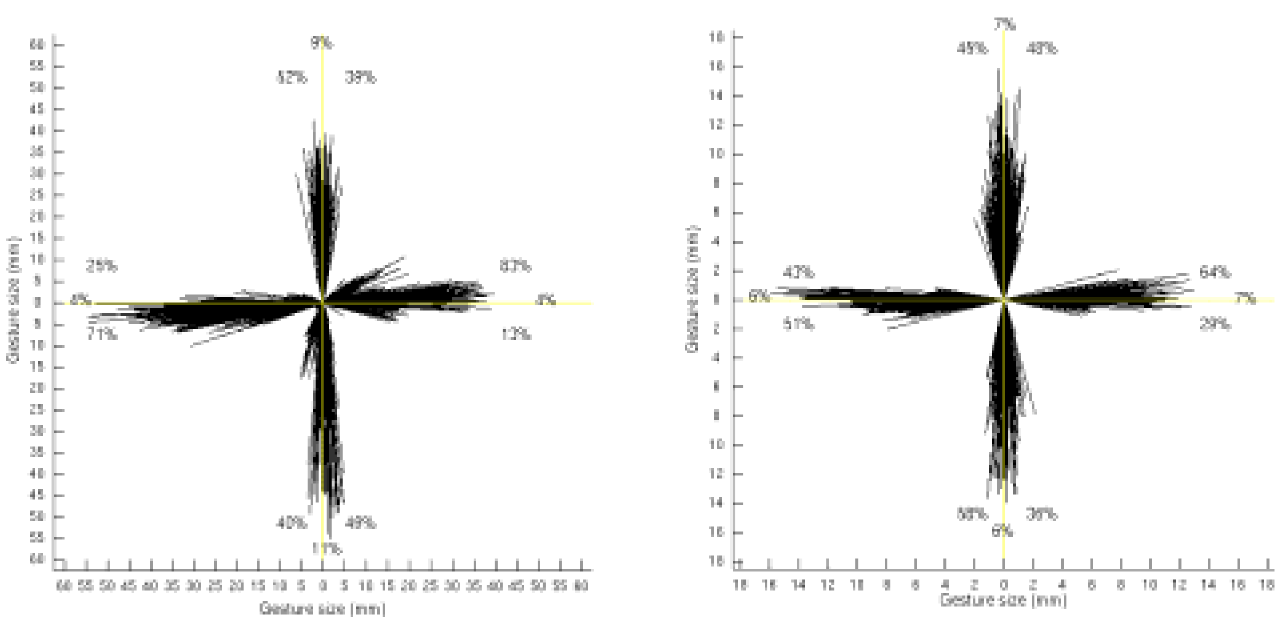
Experiment one: pen versus non-accelerating mouse

Pen gestures were substantially larger than mouse gestures, with the mean magnitudes of physical movement for the pen and mouse of 18.9mm (σ 9.5) and 7.0mm (σ 2.2). This is a reliable difference: $F(1, 13)=10.84$, $p < .01$. The gesture magnitudes in the four directions (left, right, up and down) were not reliably different: $F(3, 39)=1.47$, $p=.24$.

The difference between the magnitude of pen and mouse gestures is unsurprising. Pen gestures begin as soon as the pen makes contact with the pen-computer display: the gesture size is directly equivalent to that created with a pencil on paper. With the mouse, however, the gesture size is the subset of the physical mouse movement that occurs between the button being depressed and released.

Analysing angular errors reveals an interesting contrast between gestures created with the pen and the mouse. Figures 2a and 2b show the magnitude and direction of gestures created in the four directions using the pen (left) and the mouse (right). The 'brush-like' effect when using the pen (Figure 2a), particularly for the rightwards gesture, indicates a higher degree of angular error.

Analysis of variance showed no significant difference between the mean angular errors using the pen (6.5 degrees, σ 6.7) and the mouse (3.6 degrees, σ 1.4): $F(1,13)=2.38$, $p=.15$. It also showed no significant difference between the angular errors in the four gesture directions: $F(3,39)=1.64$, $p=.2$. There is, however, a significant and surprising interaction between the input device and direction: $F(3,39)=3.15$, $p<.05$. Figure 3 reveals the cause of the



(a) Pen gestures. **(b) Mouse gestures.**

Figure 2: Magnitude and angular errors of gestures (left, up, right, down) using the pen and mouse.

interaction. It shows the mean angular errors for the pen and mouse gestures across the four directions, and reveals that angular errors are relatively large in the left and right directions when using the pen but small when using the mouse. Conversely, the angular error for the down gesture is relatively large when using the mouse but small when using the pen.

The directional biases of angular errors are shown as percentage values on either side of each direction in Figure 2. These values show that when using the pen to make horizontal (left or right) gestures, there is a strong tendency to err upwards on right gestures (83%), and downwards on left gestures (71%). The bias works in the same direction for mouse gestures, but the effect is much less pronounced, with 64% of right gestures erring up, and 51% of left gestures erring down. There were no clear biases in angular errors for the up and down gestures for either the pen or the mouse. The physical mechanisms used to create gestures partially explain these stereotypical error biases, as discussed in the 'Observations and Comments' section.

All of the subjects were right-handed. We suspect that the direction of the angular bias partially depends on the orientation of the user's body to the device, and would be reversed for left-handers.

Experiment two: left versus right mouse button

Everyday user interfaces frequently require dragging the mouse with the left button depressed (normally with the index finger), but rarely require mouse movement with the right button depressed (normally with the middle or third finger). We therefore suspected that there could be differences between the magnitude, timing, and angular errors of gesture commands when issued with the left and right mouse buttons.

No significant differences between the means for the two buttons were detected for any of the three dependent variables (magnitude, timing, and angular error). Furthermore, the means and standard deviations for each of the values indicated that use of different mouse buttons has a negligible impact on the performance of gesture input. The subjects' comments, however, revealed a marked preference for creating gestures with the left button (discussed below).

Unexpectedly, there was a marginally significant difference between the mean time to create gestures in the four directions: $F(3,18)=2.57$, $p=.09$. This appears to be due to a comparatively high mean for the down gesture (180 milliseconds). This is further analysed in experiment three.

Experiment three: impact of mouse acceleration

We expected that mouse acceleration would have a dramatic effect on the gesture magnitude measured in screen pixels.

In comparing gesture magnitude between a non-accelerating mouse and one accelerating with a default setting, there was a marginally significant difference between the mean pixel

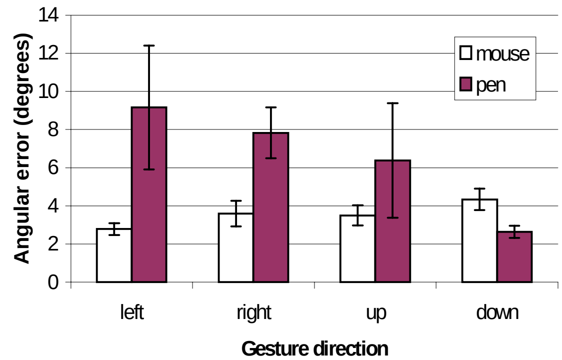


Figure 3: Mean angular errors in each of the four directions using the pen and mouse devices. Error bars show one standard error above and below the mean.

distances: $F(1,19)=3.26$, $p=.087$. The mean pixel distance for the non-accelerating mouse was 111 (σ 34.9) pixels, which equates to a physical mouse movement of 6.9mm. The mean pixel distance for the accelerating mouse was 144.0 (σ 45). The pixel distance for the accelerating mouse cannot be easily translated to a physical mouse movement because of the non-linear mapping between mouse and cursor movement.

The small difference between the magnitude of gestures created with non-accelerating and accelerating mouse settings is almost certainly due to the short duration of the gestures. Mouse motion causes accelerated movement only when the 'threshold' movement per unit time is exceeded. When the threshold time is not substantially larger than the gesture time, the accelerated mapping applies for only a small subset of the gesture.

ANOVA revealed an unexpected significant difference between the mean times taken to issue gestures in different directions: $F(3,57)=7.16$, $p<.001$. The mean gesture times in the left, right, up and down directions were 159 (σ 43), 164 (σ 45), 160 (40) and 179 (σ 50) milliseconds. The down gesture therefore took approximately 11% longer than the others to issue.

Observations and comments

Many subjects commented that gestures felt 'awkward' and tiring with the mouse. Equivalent statements were not made about the pen interface. Several subjects using the mouse commented that vertical (up and down) gestures were "no where near as natural" as horizontal (left, right) ones. In comparing left and right gestures, those that expressed a preference preferred the leftward direction. Although mouse-users found horizontal movement preferable, pen subjects generally found up and down gestures to be more natural than left and right. The preferences for different directions when using the pen and mouse can be attributed to the physical characteristics of gesture generation, as described below.

There was a strong preference for creating gestures using the left mouse button rather than the right one. All but one of the subjects used their middle finger to press the button when generating flicks with the right mouse button. The remaining subject used his third-finger. The left finger was always used on the left mouse button. The subjects' lack of experience with dragging using the right mouse button is the most likely explanation for the left-button preference.

Physical characteristics of gestures

The motor movement used to generate gestures helps to explain some of the observed performance and preference differences.

When using the mouse to make left and right gestures, the hand was moved by laterally flexing the wrist with almost no finger movement. In contrast, when making left and right gestures with the pen, lateral and rotational wrist movement was combined with small amounts of finger extension (left) or contraction (right) was used.

The subjects used two different methods to create vertical gestures with the mouse. The less commonly used technique was to move the whole arm with minimal movement in the fingers or wrist. More often, however, the subjects kept their hand and wrist still, and moved the mouse by extending (up) or contracting (down) their thumb and fourth/little fingers. With the pen, vertical gestures were made by extending and contracting the fingers and thumb.

CONCLUSIONS

Marking menus and other forms of gesture input are being used in commercial desktop systems and in mobile devices with increasing frequency. Systems that support gesture input must place constraints on various gesture parameters. This paper reported on an empirical analysis of the size, timing and accuracy of linear 'flick' gestures such as those used in marking menus. The aim is to guide the selection of appropriate values for some of these parameters.

As well as reporting concrete mean values for these measures, based on a pool of 5200 gestures, the experiment revealed some surprising results. These include the high levels of angular error when using a pen-based input device to generate rightwards and leftwards gestures, and the relatively long time taken to generate downwards gestures using the mouse.

In further work, we will investigate the leftwards and rightwards 'sloppiness' observed with the pen. We would like to understand how, if at all, this effect impacts on pen-based marking menu interfaces, and whether the effect is reversed for left-handed users.

REFERENCES

1. 3Com. *PalmPilot Handbook*, 1997, www.palm.com/support/handbooks/palmpilothb.pdf.
2. Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An Empirical Comparison of Pie versus Linear Menus. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*. 1988. 95–100.
3. Chatty, S. and Lecoanet, P. Pen Computing for Air Traffic Control. In *Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems*. 1996. 87–94.
4. Dulberg, M.S., Amant, R.S., and Zettlemoyer, L. An Imprecise Mouse Gesture for the Fast Activation of Controls. In *Proceedings of INTERACT'99*. 1999. 375–382.
5. Fitts, PM. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology 47*, 381–391, 1954.
6. Goldberg, D. and Richardson, C. Touch-Typing with a Stylus. In *Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems*. 1993. 80–87.
7. Isokoski, P. Model for Unistroke Writing Time. In *Proceedings of ACM CHI'2001 Conference on Human Factors in Computing Systems*. 2001. 357–364.
8. Kurtenbach, G. and Buxton, W. Issues in Combining Marking and Direct Manipulation Techniques. In *Proceedings of ACM UIST'91*. 1991. 137–144.
9. Kurtenbach, G. and Buxton, W. User Learning and Performance with Marking Menus. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*. 1994. 258–264.
10. Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, W. The Design of the GUI Paradigm Based on Tablets, Two-hands and Transparency. In *Proceedings ACM CHI'97 Conference on Human Factors in Computing Systems*. 1997. 35–42.
11. Kurtenbach, G., Sellen, A. and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction 8*, 1, 1–23, 1993.
12. MacKenzie, IS. and Riddersma, S. Effects of Output Display and Control-Display Gain on Human Performance in Interactive Systems. *Behaviour and Information Technology 13*, 5, 328–337, 1994.
13. Rubine, D. Specifying Gestures by Example. In *Proceedings ACM SIGGRAPH'95 Conference on Computer Graphics*. 1995. 329-337
14. Venolia, D., and Neiberg, F. T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. In *Proceedings ACM CHI'94 Conference on Human Factors in Computing Systems*. 1994. 265–270.