"Icon Throwing" user interface in tablet PC

Masaatsu Kasukawa, Atsuo Ohki, Yasushi Kuno and Hiroyasu Kakuda

(Recieved: September 25,2006)
(Revised: December 26,2006)

Abstract

We have been proposing "Icon Throwing" as a new primitive for GUI operations. In previous papers, we have used mice as pointing devices for performance measurement, and the results were very positive. We have also found that the task performance is strongly affected by the various aspects of pointing devices. Recently, LCD tablets that use stylus pens are getting popular as pointing devices, especially for mobile PCs; thus task performance on this kind of devices are becoming important. In this paper, we have conducted an experiment to compare the performance of "Icon Throwing" operation between a mouse and a stylus pen in tablet PC, using eight subjects. As the result, a stylus pen outperformed a mouse in average task operation time, while keeping accuracy (error ratio) at the same level.

1 Introduction

Current GUI interfaces such as Microsoft Windows or X-Window operate application programs through the icons, using a mouse as the pointing device, in the two-dimensional bitmap display.

These are called as a WIMP (stands for window, icon, menu and pointing device) interface and this is most common nowadays.

On standard WIMP interfaces, we use pointing devices in two stages. Firstly, we point at one of the icons corresponding to some object which we would like to manipulate. Secondly, we pop-up menus to select operation to be performed upon the object. However, menu selection itself includes multiple stages, namely: (1) display the menu onto the screen, (2) move pointer over menu items, and finally (3) select target item on the menu. This process poses large cognitive burden to users, and cannot be performed quickly. The situation gets even worse when the number of menu items gets large, or one or more levels of sub-menus are used, as we have shown in our previous research [4].

As an alternative to the menu, many UIs also support "drag-and-drop" operation. In drag-and-drop, the user chooses an icon as usual, but at the same time he/she start to drag the icon and drop it on another "target" icon. For example, dragging a file icon to an application icon opens the file by that application, or to the "trash can" removes the file. Drag-and-drop does not require complex menu operations noted above, thus can be more efficient than the menu method in cognitive workloads.

However, we thought that drag-and-drop operations are still enough complicated and can be simplified further.

Namely, a drag-and-drop consists of following three operations.

1. pointing and grabbing an icon
2. dragging the icon to a target icon
3. dropping the icon

In the above, time for both step 1 and step 2 is governed by Fitts' law [2, 3].

In our previous research[4], we have measured time for step 2, and it cost 300-750 milliseconds. This duration is largely the same as time to select a menu item, and as the result, time efficiency of drag-and-drop operation is about the same as that of menu operation.

Then, our idea is as following: if step 2 poses burden on the total operation time, remove the step somehow. Thus, to reduce the drag-and-drop time score, we have proposed "Icon Throwing " method as follows[1]:

1. hold an icon by pressing a button
2. start to drag the icon toward the target
3. release the button along the path
4. icon keeps "flying" to the target
5. action (same as drop operation) takes when the icon hits the target

By omitting the precise positioning associated with dropping, this method can reduce the time cost. In our opinion, this was possible because "throwing" is not constrained by Fitts' law. Note that first stage (selecting the icon to drag) still requires ordinary positioning; we have reduced the number of positioning (governed by Fitt's law) from two to one.

In our previous research[1], we have compared performance of five methods namely: "icon throwing", "drag-and-drop", "operation palette", "pop up menu" and "keyboard shortcut". The performance of "icon throwing" was second fastest in time, and its accuracy (error rate) was also second best (lowest). In both measurements, "keyboard shortcut" won the top. The result confirms effectiveness of keyboard shortcut facility popular in today's menu interfaces.

However, today's mobile PCs are equipped with no keyboard at all, or with small keyboards which cannot be used as efficiently as in ordinary desktop PCs. In those situations, our "Icon throwing" method will be quite promising.

In our previous experiment, we chose the mouse as a pointing device. Mice and trackballs are "indirect" devices in that a movement of the devices is translated (by software) to the corresponding cursor movement on the screen. This translation sometimes included "acceleration" — when the user moves the device quickly, cursor movement is somewhat magnified, so that user can reach the "far end" of the screen more easily. This kind of tricks does enhance usability of those devices, but users must get used to the combination of devices and software to use them effectively.

Tablet PCs have changed the situation largely. In those PCs, the user moves stylus pen along the LCD screen, and the cursor moves exactly by the same amount on the screen, greatly simplifying the situation (and the mental model which users have to deal with). We have noticed above merits during the development of "Icon-throwing Shell[5]." Thus in this paper, to show the effectiveness of the "icon throwing" method on LCD mobile PCs, we compare and evaluate "icon throwing" performance of stylus pen against that of traditional mouse device.

2 Related works

Much work had already been done in comparing performance of mice, trackballs and stylus pens as a pointing device.

MacKenzie *et al.* measured the pointing performance and dragging performance of three kind of pointing devices(mouse, trackball and stylus pen). In their result, a trackball was not good for both pointing and dragging, a stylus pen was the best for the pointing performance and a mouse was the best for the dragging performance[6].

Mizobuchi *et al.* examined the analysis of pen from physical characteristics, and length and weight requirement in PDA stylus pen[7]. In another paper, they reported the relationship between the target size and the number of targets in a stylus pen and cursor control keys[8]. Ono reported the effectiveness of stylus pen and studied a stylus pen performance model[9, 10].

For keyboardless input method using stylus pen, there were Unistroke[11], T-Cube[12]. These two were also implemented in FreeBSD operating system by one of the authors[13].

In [14], Gordon *et al.* compared three devices (mouse, stylus pen and trackball) and found that a stylus pen was the best and a trackball was the worst for a menu operation.

They also reported that the response time and the error ratio both linearly increased according to the number of menu items. Kato *et al.* reported that directions of the pointing target directly influenced the accuracy in icon movement or figure editing tasks using mouse and stylus pen[15].

Hashimoto *et al.* reported that stylus pen was easiest to operate when the pointing target directed to northeast or southwest[16, 17].

3 A stylus pen and a large/small screen

As noted earlier, today's mobile PCs today have no keyboards or small (difficult to use) keyboards. As alternatives for fixed PCs, very large screens equipped with touch-sensors are becoming popular. In those situations, the stylus pen is used for both of the following operations:

- as a pointing device
- as a character input device

Users can input characters without keyboard, point the target to click, and drag the icon in the same stream of operations.

This was the advantage of the stylus pen interface over the keyboard and mouse interface, as users need not move the right/left hand between the mouse and the keyboard.

Alternatively, our "Icon throwing" interface provided full merit in the huge screen, e.g. 10 to 20 times larger pixels (in both width and height) compared to the current standard screen.

On such a huge screen, operations using a mouse encounter serious problems because to archieve both fine control and long movement will become more and more difficult.

To use such a huge screen, users will "walk along" the screen and touch directly on the screen for pointing and dragging,like in case of writing/drawing on a blackboard.

In this case, dragging a long way requires large physical movement and will be very time consuming. In our scheme, the user can naturally "throw" the icon to anywhere he/she wants.

For smaller (portable) devices, stylus pen provides direct operation over the screen and is more natural to the user. This is already conceived as merit compared to other indirect pointing devices as in mice or trackballs.

In those devices, screen is generally small and users have no difficulty in moving pens to the "far end" of the screen. Thus one might suspect that "throwing" will not be necessary for such situation, but we think the contrary. Fitt's law states that time required for positioning is determined by the ratio of diameter of the target against target distance. Therefore, small screen with small icons will require the same positioning time as large screen with large icons. Thus, still the advantage of "throwing" will hold for those portable devices.

4 Method of the experiment

4.1 Purpose of the experiment

In our previous papers[1, 4], we found out that "Icon Throwing" requires some trainings and skills. We suspect that indirectness of mouse devices as noted previously made fine control of pointers difficult, leading to the result.

If computer users manipulate a stylus pen in place of a mouse, as the pen has more direct characteristics, they will be able to control the positioning more easily. So that, they will be able to use "throwing" more easily with lower error rate.

In this paper, we used the same framework of experiments as in [1]. The experiment method is as follows:

- A pile of source (to-be-thrown) icons and four target icons are displayed on the screen.
- Both source and target icons are labeled with the letter A, B, C or D.
- Subject on-clicks (pen-downs) at the topmost source icon, and start to drag (move while on-clicking/pen-downing) it toward the target icon with the same letter.
- Subject off-clicks (pen-ups) the pointing device while moving; the source icon keeps moving with the same speed and direction.
- When the source icon reaches within pre-specified distance from one of the target icons, the source icon is automatically "dropped" to the target icon.

- When the source icon reaches one of the edges of the window without "dropping," the source icon stays there.

These were the outline of the experiment. Along this outline, we made the following decisions:

- From the feature of "Icon Throwing", we have decided a unit of tasks as "click on an icon, drag it, click off it and click the next icon". "dropping" is not included in the task. This is because after releasing the source icon, the subject has no control over the thrown icon, say "ballistic" movement, and can immediately start picking next source icon while current source icon is still moving.
- The number of goal icons is four. We would like to exclude odd number because "center" target (same horizontal position as in the source icon) might get undesirable effect (unrealistic case, too easy to throw, looks something special to the subject, etc.), and two will be too small, six will be too large (confusing).

The start-up screen of a program for the experiment was shown in figure1. In the following, we called the circles at the left side of figure (source icon) as "an icon", the four squares at the right side as "targets".

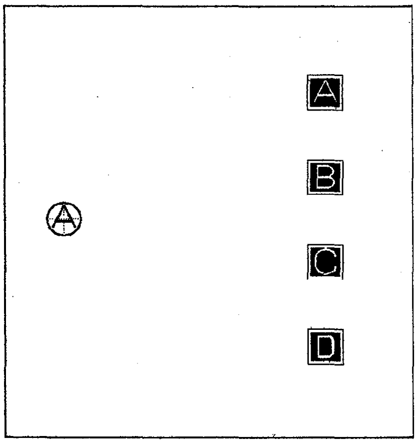

Figure 1: Initial layout of the experiment program

4.2 Experiment environment and subjects

The environment and subjects of the experiments is as the following:

- Computer for the experiments:
  NEC VersaPro VA93J/GL (CPU: mobile PentinumIII 933MHz 256MB with 10.4 inch LCD monitor,1024x768 pixels,24bit color depth)
- When using a mouse, the PC must be attached to an original stand. When using a stylus pen, the PC is released from the stand and held freely by hand.
- We have used FreeBSD 5.0 and XFree86-4.3.0 (X11R6).
- Mouse was Logitech OEM Dell optics 2 button mouse with a wheel, 400counts/inch.
- Stylus pen was the one attached to the computer.
- An icon was a circle of 12 mm in diameter, labeled the letter A,B,C or D.
- A target was a square in 12 mm on a side, also labeled the letter A,B,C or D.

- The average distance between an icon and goal targets was 98 mm. Minimum distance was 93 mm and maximum distance was 103 mm.
- The angle from an icon to goal targets were as follows; to A:26.5 degrees, to B:8.7 degrees, to C: -8.7 degrees, to D:-26.5 degrees.
- Subjects are 6 males, 2 females, aged 22-25 years old. All subjects used computers and mice on a daily basis and all subjects operated a mouse or a stylus pen by right hand.

It is possible to read pointing coordinate even if stylus pen is somewhat away from screen. Device driver has been developed taking advantage of that function. In the followings, we call touch to the screen as a "click".

4.3 Summary of a program for the experiment

The program used for our experiment has the following features.

- When an icon is released while in movement, the program moves it with the same speed and direction at the moment of releasing.
- If a moving icon meets the edge of the screen, it is stuck there and can not be operated anymore and "error" counter is incremented.
- If a moving icon reaches one of the targets, the icon is stuck to the target and is immediately moved underneath the target (slightly offset toward lower-right, so that series of stuck icons are visible in a row), and can not be operate upon anymore.
- While an icon is moving, the subject can re-throw it to a new direction.
- Each icon moves independently and they can be thrown simultaneously.
- Movements of the icons are independent from each other.
- Target collision range is a circle of 180 pixels in diameter. When the center of an icon enters within the circle, the icon is considered to reach the target.
- Whether the label on the target is the same as the label on the icon or not does not affect the "dropping".
- After dropping, if those labels match, "correct" counter is incremented; otherwise "error" counter is incremented.
- The ordering of icons is specified in the external file, which we have prepared for the experiment sessions based on the random numbers.
- Each kind of icons appears five times in each session.
- When an icon reaches a wrong target or the edge of screen, a retry icon (which have the same letter as the failed one) will be added at the end of source icon pile. This is to collect sufficient successful throwing data.
- As the special case, when the last icon in the pile has failed, a dummy icon is added first to the list, and then the retry icon is added after that. This is to prevent practice effect by throwing to the same target repeatedly. Letters labeled for the dummy targets is chosen cyclically as in A,B,C,D,A.

20 successful throws (excluding first and last ones, which are dummies) deserve single session. Within each session, we measured the time of each throws, an average time of 20 throws, and error frequency of the session.

Note that **the time of throw** is not a time from a start of a throw action to the time icon reached a target, but from the start of throw action to the next start of throw action. The reason for the choice is that it is possible to throw next icon (start next task) while previous icon is still moving.

4.4 Procedure of the experiment

In the following, we express one mouse session as m, one stylus pen session as p. We call an experiment of succeeding 5 sessions of the same pointing device as a set and express a set of mouse ($mmmmm$) as M and a set of stylus pen ($ppppp$) as P.

We conducted the experiment according to the following procedure:

1. Explanation
   - We explained subjects about icon throwing and purpose of experiment (to compare a mean time of stylus pen throwing against a mean time of mouse throwing).
   - We demonstrated how one could throw icons.
   - Subjects have practiced until they were satisfied with throwing with both mouse and pen.
2. Experiment
   - Subjects were formed into two groups; one started with a mouse and the other started with a stylus pen.
   - all subjects performed 30 sessions in 6 sets $MPMPMP$ or $PMPMPM$ according to the group.
   - We ignored leading 10 sessions and used remaining 20 sessions as valid data.

To exclude the learning effect, we performed the experiment repeatedly, namely once per day. From the second day, t-test for difference between the current and the previous day's average performance is conducted for each of the subjects and pointing devices (a mouse or a stylus pen). If there are statistically significant difference on either of the pointing devices, the experiment for that subject continues; otherwise the experiment is over and the previous day's result is adopted as that subject's performance measure.

5 Results and discussions

5.1 Time of a throw

We assumed the normal distribution of unequal variance between a mouse trial and a stylus pen trial, and used the Welch's t-test. Table 1 is the result of average time for one throw, its standard deviation, and t-test value for a mouse and a stylus pen. In that, eight subjects' result are all statistically significant($p < 0.01$). And the ratios of the stylus pen S.D. to the mouse S.D.lay between 2.1(subject 2) to 4.5(subject 6).

Table 1: Mean throwing time using a stylus pen/mouse

subject	mouse average in sec.(*S.D.*)	stylus pen average in sec.(*S.D.*)	difference	t	*d.f.*
1	0.669(0.192)	0.593(0.081)	0.076	5.15	267.3
2	0.739(0.249)	0.590(0.118)	0.149	7.65	283.6
3	0.600(0.183)	0.467(0.067)	0.133	9.63	252.2
4	0.605(0.197)	0.535(0.057)	0.070	4.82	232.5
5	0.569(0.132)	0.498(0.056)	0.071	7.03	269.8
6	0.860(0.335)	0.528(0.074)	0.332	13.67	218.4
7	0.683(0.194)	0.588(0.083)	0.095	6.40	269.3
8	0.617(0.209)	0.519(0.055)	0.098	6.38	225.9
S.D. means a standard deviation, *d.f.* means s degree of freedom.					

Table2 shows the required trials for the experiment completed. One trial consists of six sets or 30 sessions. This table shows somewhat extra trials are required in a stylus pen. But the difference is not so large. Subject 3 required 5 more trials but the others required only one or two trials at most.

Table 2: Required trials to converge the average in mouse and stylus pen

subjects	mouse	stylus pen	difference
1	3	3	0
2	2	3	-1
3	2	7	-5
4	4	4	0
5	3	4	-1
6	3	5	-2
7	3	3	0
8	3	4	-1

5.2 Error frequency

Table3 shows an error frequency for single session. The result is that an error frequency for a mouse and a stylus pen have no difference(non significant).

Table 3: Mean errors in 20 throws(stylus pen and mouse)

subject	mouse(S.D.)	pen(S.D.)	difference of mean	t
1	1.0(1.69)	1.0(1.41)	0.0	0.00
2	1.4(1.08)	0.5(0.97)	0.9	1.96
3	0.6(0.70)	1.3(0.95)	-0.7	-1.88
4	0.9(0.88)	1.7(1.06)	-0.8	-1.84
5	1.0(0.67)	0.4(0.70)	0.6	1.96
6	1.8(1.23)	0.9(1.20)	0.9	1.66
7	2.3(2.16)	2.5(1.96)	-0.2	-0.22
8	2.5(1.72)	1.9(1.37)	0.6	0.86
$d.f. = 18$				

5.3 Comparison of a stylus pen and a mouse

In our experiment program, a thrown icon continues to keep speed and direction at the moment of releasing. Physical distance needed to move the cursor by 1 pixel differs between a mouse and a stylus pen, and this difference might affect the ease of throwing.

In mouse devices, "cpi"(count per inch) is used to express count of pulses the mouse sends out while 1 inch movement. All mice used in our experiments have the cpi value of 400. Also, in X-Window users can modify "acceleration factor" using "xset" command, and subjects have chosen the factor from 4 to 6.

On the other hand, stylus pen does not have those parameters; to move 1 pixel the user must move the device by the same amount.

In the case of traversing whole screen horizontally, following computation holds:

- Mouse — when the acceleration factor of 4 is used, $1024/4 = 256$ counts is necessary, which corresponds to $256/400 = 0.64$in. physical movement.
- Pen — Screen width was 8.32 inches, thus 8.32in. physical movement is necessary.

Thus, when the physical movement by itself is compared, stylus pen requires about 10 times larger movement. However, the experiment result shows that the pen is faster. This fact shows that the difference of pointing time does not come from the distance of movement in these pointing devices. Superiority of stylus pen might come from the difference of indirect and direct positioning device noted above.

5.4 Result of the experiment

Final result is that subjects can "throw" by a stylus pen better than a mouse in speed, and error ratio for those two are indifferent, so a stylus pen is appropriate device for "Icon Throwing" method.

6 Future works

After the experiment, we have reviewed our experiment design and listed up several points which we should consider further experiments, as in the following:

- Direction of throwing and speed.

  Some researchers[15],[16],[17] pointed out that moving pointer toward upper-right seems to have better performance compared to moving pointer toward upper-left. Our throwing method might have a same kind of "direction dependency," and examining the effect might lead to better interface design. And in this experiment, the mouse acceleration factor was determined from the subjects' ease of pointings. we want to see the result if the both were tested in the same physical movement factor.

- Separate cognitive errors and slips.

  During the experiment, two kind of errors are observed. One is the cognitive errors, in which target label is miss-recognized and subject throws to a wrong target "correctly." Another is slip, in which target is correctly recognized but subject's muscles are irregularly activated and the icon is thrown to a wrong direction. We would like to separate those two and analyze them more thoroughly.

- Icon throw and Fitts' law.

  In the above experiment, the total throwing time consists of

  - Time to point the target.
  - Time to drag the target and release it.

  These two operations are executed overlapped but it will be possible to examine the each of the execution time and evaluate the convolution. We would like to reveal the contribution of each time and variance.

Finally, we would like to investigate more on the effect of subjects' background (how much subject is accustomed to each pointing devices), subjects' physical positioning, use of multiple screens, use of fingers (instead of stylus pen).

References

[1] Yasushi Kuno, Hiroyasu Kakuda, Atsuo Ohki, Masaatsu Kasukawa: Are Icons Throwable?, Programming Symposium Vol.35, *Information Processing Society of Japan*, pp. 121–132 (1994)

[2] Fitts, P.: The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, Vol. 47, pp. 381–391 (1954).

[3] Card, S.K. Moran, T. and Newell, A.: *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum (1983).

[4] Yasushi Kuno, Atsuo Ohki, Hiroyasu Kakuda, Masaatsu Kasukawa: An "Icon Throwing" User Interface, Computer Software, Vol. 13, no. 3, pp. 38–48 (1996).

[5] Yasushi Kuno, Hiroyasu Kakuda, Atsuo Ohki, Masaatsu Kasukawa: An "Icon Throwing" Shell: Programming Symposium Vol.38, *Information Processing Society of Japan*, pp. 65–75 (1997)

[6] Scott MacKenzie, Abigail Sellen, William Buxton: A Comparison of Input Devices i n Elemental Pointing and Dragging Tasks, *Proceedings of the CHI'91 Conference of Human Factors in Computing Systems*, ACM, pp. 161–166 (1991).

[7] Sachi Mizobuchi, Xiangshi Ren, Michiaki Yasumura: Long Pen, Short Pen or Index Finger? Investigating Pointing Performance on Handheld Devices, Interaction 2003, *Information Processing Society of Japan*, pp. 147–154(2003)

[8] Sachi Mizobuchi, Xiangshi Ren, Michiaki Yasumura: An Empirical Study of the Minimum Required Size and the Number of Targets with a Pen and with a Cursor Key on a Small Display,*IPSJ Journal* , Vol. 43,no 12, pp. 3733–3743(2002)

[9] Makoto Ono: An experimental study of a pen's user interface,41st Human Interface Workshop Report,*Information Processing Society of Japan*,pp. 82–99 (1992).

[10] Makoto Ono: Performance modeling for pen input,47th Human Interface Workshop Report, *Information Processing Society of Japan*,pp. 1–6 (1993).

[11] David Goldberg, C. R.: Touch-Typing With a Stylus, *Proc. of ACM Conference on Human Factors in Computing Systems (CHI'93)*, ACM, pp. 80–87 (1993).

[12] Dan Venolia, F. N.: T-Cube: A Fast,Self-Disclosing Pen-Based Alphabet, *Proc. of ACM Conference on Human Factors in Computing Systems (CHI'94)*, ACM, pp. 265–270 (1994).

[13] Atsuo Ohki: Introduction to FreeBSD kernel,ISBN 4756117422,ASCII Publishing(1998).

[14] Gordon P. Kurtenbach, Abigail J. Sellen, William Buxton.: An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus, *Human Computer Interaction*, Vol. 8, pp. 1–23 (1993).

[15] Naoki Kato, Masaki Nakagawa: An Investigation into the Usability of a Pen for Designing Pen User Interface, *IPSJ Journal*, Vol. 39,no. 5,pp. 1536–1546(1998).

[16] Minako Hashimoto, Umpei Nagashima, Masatomo Togasi, Haruo Hosoya: A Virtual Oval Keyboard and a Vector Input Method for Pen-based Computing, *IPSJ Journal*, Vol. 37,no. 11,pp. 2105–2115 (1996).

[17] Minako Hashimoto, Masatomo Togasi: A Virtual Oval Keyboard and a Vector Input Me thod for Pen-based Character Input, *Proc. of ACM Conference on Human Factors in Computing Systems (CHI'95)*, ACM, pp. 254–255 (1995).

Masaatsu Kasukawa
Affiliation: Department of Information Sciences, Faculty of Science, Ochanomizu University.
E-mail: kasukawa@is.ocha.ac.jp
Atsuo Ohki
Affiliation: Graduate School of Systems Management, University of Tsukuba, Tokyo.
E-mail: ohki@gssm.otsuka.tsukuba.ac.jp
Yasushi Kuno
Affiliation: Graduate School of Systems Management, University of Tsukuba, Tokyo.
E-mail: kuno@gssm.otsuka.tsukuba.ac.jp
Hiroyasu Kakuda
Affiliation: Department of Computer Science, The University of Electro-Communications.
E-mail: kakuda@cs.uec.ac.jp