Boomerang: Suspendable Drag-and-Drop Interactions Based on a Throw-and-Catch Metaphor

Masatomo Kobayashi
Department of Computer Science,
The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, Japan
Tel: 81-3-5841-4091
kobayash@is.s.u-tokyo.ac.jp

Takeo Igarashi
Department of Computer Science,
The University of Tokyo / SORST, JST
7-3-1 Hongo, Bunkyo, Tokyo, Japan
Tel: 81-3-5841-4109
takeo@acm.org

ABSTRACT
We present the "boomerang" technique, which makes it possible to suspend and resume drag-and-drop operations. A throwing gesture while dragging an object suspends the operation, anytime and anywhere. A drag-and-drop interaction, enhanced with our technique, allows users to switch windows, invoke commands, and even drag other objects during a drag-and-drop operation without using the keyboard or menus. We explain how a throwing gesture can suspend drag-and-drop operations, and describe other features of our technique, including grouping, copying, and deleting dragged objects. We conclude by presenting prototype implementations and initial feedback on the proposed technique.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Drag-and-Drop, Throw-and-Catch, Gestures

INTRODUCTION

Drag-and-drop interactions are frequently used in modern desktop environments. This technique helps users move various objects, including files, texts, and images, within and across applications. However, many usability issues remain. One of the most serious problems is that once users start dragging an object, very few actions can be taken while holding the object. For example, users cannot switch to the foreground window by clicking the title bar of a background window or a button on the taskbar, although such actions are often necessary to make the drop target visible. Dragicevic [5] addressed this problem by allowing one to temporarily switch the foreground window with a *leafing through* gesture. However, users may also need to scroll a window to a distant target, open a hidden folder in the folder tree, or change the active tab, sheet, and slide to make the destination visible. These operations are difficult or even impossible to perform using existing techniques.

To address this problem, we developed the "boomerang" technique, which allows users to suspend a dragging operation using a throw-and-catch metaphor. Once dragging is suspended, users can perform another operation, including those described above, before dropping the object. Figure 1 illustrates a basic behavior of the technique. First, users can start dragging an object as in a traditional dragging interaction. To perform another operation before dropping, they *throw* the object by releasing the mouse button while continuing to move the pointer. The object flies out of the screen, and its location when it was thrown is marked by an animated, translucent circle. At this time, users can complete any operation to make the hidden target visible such as, for example, scrolling a window. After finishing, users move the mouse pointer to the translucent circle to make the thrown object fly back into the screen, and *catch* it by clicking on the object, and then resume the drag-and-drop operation.

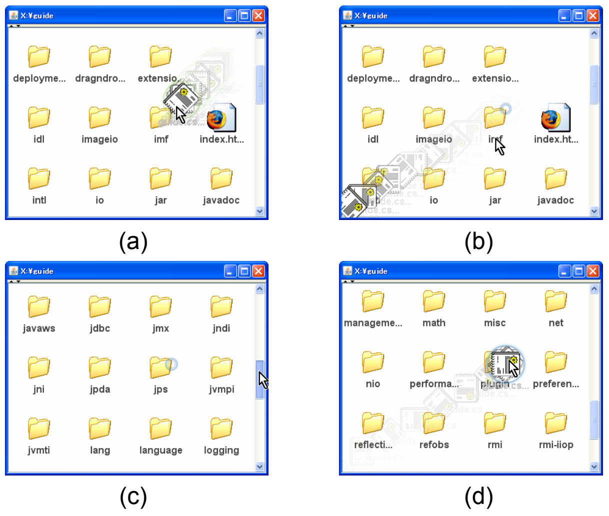

Figure 1: (a) A throwing gesture suspends a dragging operation; (b) the thrown object flies out of the screen; (c) users do something else, e.g., window scrolling; and (d) they catch the thrown object and resume dragging.

The remainder of this paper is organized as follows. First we summarize the related work. Then we provide an overview of the boomerang technique, and propose a number of advanced features, including grouping, copying, and deleting dragged objects. Finally, we present prototype implementations and the initial feedback on our technique.

RELATED WORK

Several techniques have been proposed to improve the usability of drag-and-drop interactions. Drag-and-throw and push-and-throw [6] use a metaphor of throwing and the pantograph, respectively, to extend one's reach while dragging. Drag-and-pop and drag-and-pick [1] temporarily move target objects toward the pointing cursor so that users can easily drop it on the target. Push-and-pop [4] is a combination of push-and-throw and drag-and-pop techniques. Pick-and-drop [7] is a drag-and-drop technique for multiscreen environments. It allows a dragged object to jump to another screen between the picking and dropping gestures. These techniques were designed to improve the efficiency of simple target-pointing tasks with large or distant displays by reducing cursor movement. Therefore, the focus is not on complicated actions such as window flipping. An exception is the fold-and-drop technique introduced by Dragicevic [5]. Using a paper metaphor and crossing-based interactions, this technique seamlessly integrates window flipping into drag-and-drop operations.

Our boomerang technique can be considered a gestural alternative to cut-and-paste in the sense that it separates cutting and pasting. However, by the very nature of drag-and-drop, the boomerang technique has several advantages over common cut-and-paste interactions. First, it works without keyboard shortcuts and context menus. Therefore, it is expected to be preferable for small-screen devices without keyboards. Second, it provides visual feedback indicating what is being dragged and what happens when it is dropped, e.g., copying, moving, and creating a shortcut. Third, drag-and-drop has more functionality than cut-and-paste. For example, it can open a file by dropping the file onto an application icon or window, create a shortcut, and perform numerous application-specific operations, such as changing the homepage by dropping a URL onto the "Home" button in the toolbar. Meanwhile, it has been shown that drag-and-drop is as efficient as keyboard-based cut-and-paste operations in terms of task-completion time [3].

THE BOOMERANG TECHNIQUE

In this section, we describe the interactive design of our technique and its visual feedback.

Figure 2 shows the state diagram of our throw-and-catch interaction. The first transition from S_0 to S_1 represents the start of a dragging operation. This transition occurs when users start dragging an object by holding down the mouse button on the object and moving the mouse in the same way as traditional drag-and-drop operations.

A transition from S_1 to S_2 occurs when the speed of mouse movement, v, becomes faster than a threshold, v_1. A reverse transition occurs when the speed becomes slower than a threshold, v_2. As visual feedback to help warn users about these transitions, the dragged object *spins* in the state S_2 (Figure 3-a). The thresholds v_1 and v_2 satisfy the inequality $v_1 > v_2$ to reduce fluctuation in the movement. In our current implementation, the thresholds are manually specified by the users. An adaptive online algorithm to adjust these thresholds might improve the performance of throwing interactions.

While the interaction is in state S_2, users can throw the dragged object by releasing the mouse button. A transition from S_2 to S_3 represents this action. While the interaction is in state S_1, users can drop the object as if it were a normal drag-and-drop operation. A transition from S_1 to S_0 represents this action. This means that the speed of mouse movement at the time users release the mouse button determines whether the dragged object is thrown or dropped. More specifically, a fast movement results in the object being thrown, while a slow movement results in it being dropped. When a throwing event occurs, the object flies out of the screen and a small circle appears at the location where it was thrown as a marker. This circle continuously grows and shrinks to help users easily locate it, but is translucent so as not to obscure other objects (Figure 3-b).

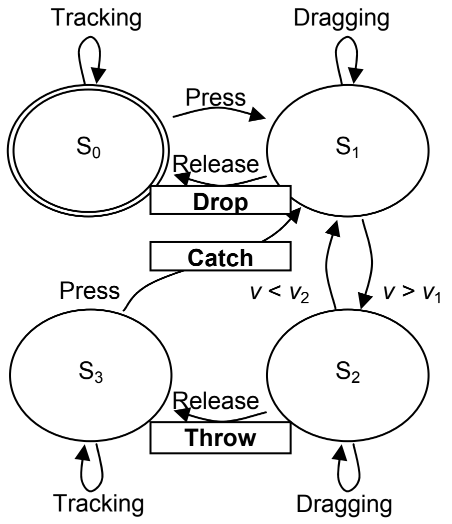

Figure 2: A state diagram of boomerang interactions. The transitions between S_0 and S_1 are the same as normal interactions.

Once the object is thrown, the drag-and-drop operation is suspended, and users can perform any other task, just as if they were not dragging an object (e.g., switch to the foreground window by clicking its title bar and scroll the window using the scroll bar). While modern operating systems such as Windows XP support window switching by hovering on a taskbar button and document scrolling by hovering near the edge of the window, directly switching windows and scrolling documents using standard clicking and dragging are much faster than hover-and-wait operations. In addition, our technique allows users to perform another drag-and-drop operation while one is suspended. The second dragging operation can also be suspended. Thus, users can keep two or more dragging operations suspended concurrently (Figure 4). Each suspended operation can be resumed by catching the object, as previously described. In this sense, the boomerang technique is an alternative to Microsoft Office Clipboard, which allows one to store multiple objects. The catch function (Figure 3-c) is represented by a transition from S_3 to S_1.

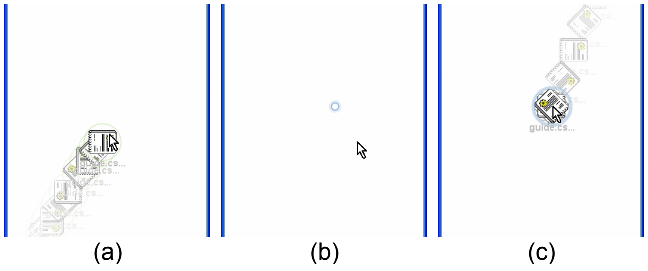

Figure 3: (a) The dragged object spins when the mouse pointer is moved quickly; (b) an animated, translucent circle indicates where an object was thrown; (c) the image of the thrown object returns when the mouse pointer nears the circle.

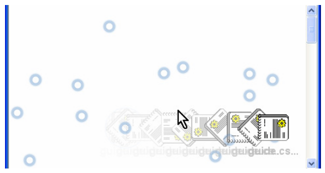

Figure 4: Users can suspend several drag-and-drop operations simultaneously.

ADVANCED FEATURES

In addition to making drag-and-drop interactions suspendable, we propose the following features to enhance dragging operations: grouping, copying, and deleting of the dragged objects.

Grouping

Figure 5 illustrates how to merge suspended operations. To create a group of thrown objects, users drop an object onto the representative circle of another thrown object. Multiple objects can be added, and they are arranged equiangularly around a circle at the center of the group. Once a group is created, users can handle all of the objects at the same time by dragging the circle (Figure 6-a). Users can also separate each object from the group by directly dragging the object (Figure 6-b).

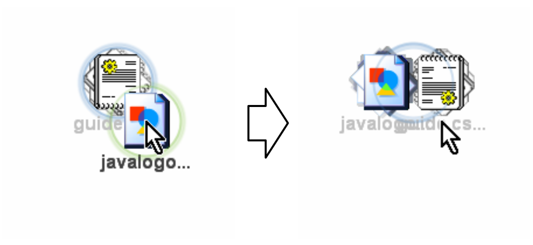

Figure 5: A group of suspended operations is created by dropping an object onto a thrown object.

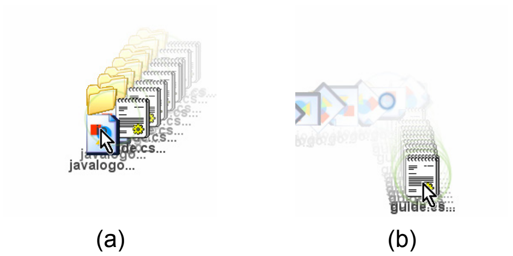

Figure 6: (a) A group can be dragged by the circle at its center. (b) Objects can be separated from the group by dragging them directly.

This grouping feature is expected to improve the efficiency of drag-and-drop operations in some cases. For example, consider a scenario in which users need to move 10 files, each in a different folder, into one shared folder. Without a grouping technique, they would have to drag each file one by one. Moreover, it would be necessary to go back and forth between the source and destination folders. Although two folders could be arranged side by side to avoid such folder flipping, this method is often inefficient because it requires rearranging windows, and wastes screen space by having multiple windows open. However, using the grouping feature, the task can be completed with minimal effort.

Copying and Deleting

Figure 7 shows how to copy and delete a dragged object. While users are dragging an object or group, two visual cues appear near the upper-center and lower-center of the screen to indicate copying and deleting, respectively. When users throw it upward, the dragged object or group is copied. The original objects remain at the original location, and the dragged objects are replaced by their copies. This is equivalent to dragging objects with the control key down in Microsoft Windows. When users throw it downward, the object or group is deleted. This is equivalent to dropping them into the system trash/recycle bin. A confirmation dialog pops up before deleting the objects.

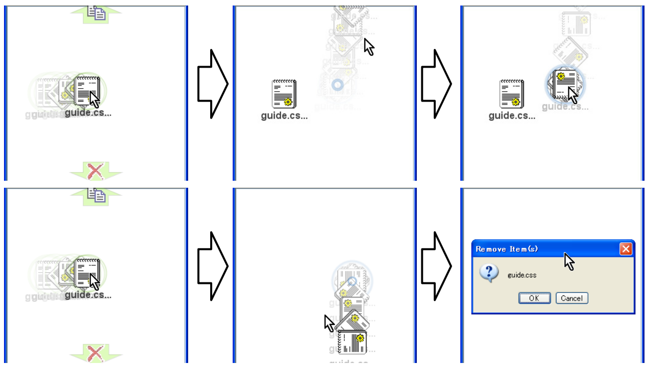

Figure 7: While dragging an object, two visual cues appear. The drag-and-drop operation enters copy mode when the object is thrown upward (top), and delete mode when it is thrown downward (bottom).

Although copying and deleting are fundamental operations in object management, traditional systems require menus, toolbar buttons, or keyboard shortcuts to complete these operations. Our technique integrates these operations seamlessly into throw-and-catch interactions using directional lists such as pie menus [2].

PROTOTYPE IMPLEMENTATION

We developed prototype applications to demonstrate and evaluate the boomerang interaction.

File Manager

Figure 1 is a screen shot of the file manager using the technique. While a dragging operation is suspended, users can scroll the folder view, open target subfolders, and start an application on which the dragged file is to be dropped. These operations are difficult to perform with traditional drag-and-drop systems.

Text Editor

A text editor is another prospective application for which the boomerang technique will be useful. It helps users cut, copy, paste, and delete multiple snippets of documents without using context menus, toolbar buttons, or keyboard shortcuts. This will facilitate the creation and editing of documents.

Scrapbook

Figure 8 shows a Web scrapbooking tool based on the boomerang technique. This tool allows users to copy snippets of Web pages by selecting a region and dragging it. The snippets can be pasted onto blank pages to make a scrapbook. As our technique allows multiple dragging operations to be kept suspended simultaneously, users can copy all excerpts in which they are interested before pasting them into a scrapbook. This separation of document browsing and scrap arrangement prevents one's concentration from being broken by repeatedly switching modes.

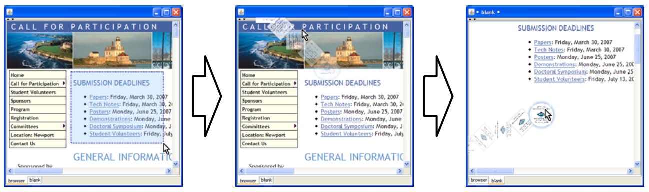

Figure 8: Users select, drag, and throw snippets of documents, then arrange the scraps on a blank page.

Initial Feedback

We asked four people to use the prototype applications described above. One was a non-technical user and the others were computer science students. All of them were frequent users of traditional drag-and-drop operations.

They all understood how to use our boomerang technique, including the advanced features, after a few minutes of instruction and practice. One commented that the throw-and-catch interaction was exciting. Another noted that the grouping feature would be very useful in practical situations. However, three of the users, including the non-technical one, indicated that they needed more experience to master the throwing gesture, while all of the users commented that the grouping function was easy to use. In addition, one user complained that the visual effects, especially the flying animations, were too distracting. The non-technical user indicated that the boomerang technique was best suited for expert users, because, in general, novice users rarely want to perform such a complicated drag-and-drop operation, regardless of whether it is suspendable or not.

DISCUSSION AND FUTURE WORK

Our boomerang technique makes drag-and-drop operations suspendable, allowing users to perform other tasks between cutting and pasting. In other words, it makes drag-and-drop an alternative to clipboard-based cut-and-paste interactions, while preserving the benefits of drag-and-drop. With the proposed technique, users can make the hidden destination visible in various ways before dropping the dragged object. In addition, our technique offers grouping, copying, and deleting without any context menus, toolbar buttons, or keyboard shortcuts. Our throw-and-catch interactions require only a pointing device, such as a mouse or stylus. Thus, the technique might also be suitable for keyboard-less and limited-screen environments, including tablet PCs, large screens, and handheld devices. We plan to conduct a user study to assess the performance of our technique in greater detail and address the problems noted by the initial users.

REFERENCES

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. In *Proceedings of INTERACT'03*, 2003, pp.57-64.

2. Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of CHI'88*, 1988, pp.95-100.

3. Chapuis, O. and Roussel, N. Copy-and-Paste between Overlapping Windows, In *Proceedings of CHI'07*, 2007, pp.201-210.

4. Collomb, M., Hascoët, M., Baudisch, P., and Lee, B. Improving Drag-and-Drop on Wall-Size Displays. In *Proceedings of the 2005 Conference on Graphics Interface*, 2005, pp.25-32.

5. Dragicevic, P. Combining Crossing-based and Paper-based Interaction Paradigms for Dragging and Dropping between Overlapping Windows. In *Proceedings of UIST'04*, 2004, pp.193-196.

6. Hascoët, M. Throwing Models for Large Displays. *HCI'2003*, *Designing for Society*, Volume 2, pp.73-77, British HCI Group.

7. Rekimoto, J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, 1997, pp.31-39.