Symmetric Interaction in the User Interface

Celine Latulipe
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
celinelatulipe@gmail.com

ABSTRACT

Symmetric two-handed interaction has been largely unexplored by the user interface community. My research is aimed at investigating this type of interaction, where the two hands work together at similar levels of spatial and temporal detail. This paper briefly describes symPut, a symmetric input system that has been designed to support symmetric interaction applications, and symDraw, a 2D drawing application that uses symPut and acts as a testbed for investigating new symmetric interaction techniques. I conclude with a brief summary of my overall research plan.

Categories and Subject Descriptors: H.5.2 User Interfaces, I.3.6 Methodology and Techniques, H.1.2 User Machine Systems

Additional Keywords and Phrases: bimanual interaction, symmetric interaction, two-handed interaction

INTRODUCTION

Two-handed (bimanual) interaction has been a topic of interest in the user interface community for many years, beginning with the Buxton and Myers landmark study in 1986 [5]. The topic received much attention in the 1990s [1, 2, 4, 6, 7, 12, 13, 14, 15, 18, 20, 23], and we are now seeing bimanual interaction used in a variety of new research systems [3, 9, 19, 22]. Many of these studies have described the benefits of two-handed interaction, such as increases in performance due to parallelism and potential decreases in cognitive load because of simpler task representations. However, most of the bimanual computer interfaces cited above have used asymmetric interaction, following the guidelines set out by Guiard in 1987 [11]. These guidelines promote a stronger role for the dominant hand and relegate the less detailed work to the non-dominant hand (*e.g.*, the left hand pans or zooms while the right hand edits image detail). There is a general belief in the user interface community that these guidelines fit well with human capabilities (*i.e.*, the two hands cannot work together at a similar level of spatial or temporal detail because of interlimb motor interference) and therefore all bimanual interfaces should follow an asymmetric style of interaction. However, a number of recent findings in psychology have shown that the two hands can in fact work together symmetrically; these findings are discussed in detail below.

UIST '04, October 24–27, 2004, Santa Fe, New Mexico, USA
ACM 1-58113-962-4/04/0010

My research aims to supplement the Guiard model of asymmetric bimanual interaction with a thorough investigation of symmetric bimanual interaction. By symmetric interaction, I refer to interaction in which the two hands work at similar levels of spatial and temporal detail, not necessarily interaction in which the two hands make mirrored movements. I believe that Guiard's model of asymmetric interaction is correct and his guidelines extremely useful. However, symmetric interaction styles have been largely unexplored and may also prove useful in some applications. There are tasks people perform everyday that require symmetric use of the two hands, such as tying shoelaces, playing piano, skipping rope, swimming and folding linen or clothing. Symmetric computer interaction is most likely to be useful in applications where:

- there are two different points that have to be specified (such as the beginning and end of a text selection),
- there are deformable objects (such as virtual sculpting or virtual wrapping of cloth) or
- there is a positioning task which involves some combination of translation, rotation and scaling (such as protein docking).

The Twister technique for two-handed deformations of 3D objects [19] is a notable exception where symmetric interaction has been put to good use in the user interface. Much more work needs to be done to explore this area.

For this research, I have reviewed the psychology literature to examine the limits of human bimanual performance. I have also developed a bimanual symmetric input system, symPut, which acts as a testbed for trying out various symmetric interaction techniques. Using symPut, I have created a bimanual symmetric drawing program as well as a prototype "don't spill the water game", both of which require symmetric interaction. A number of preliminary experiments with the symPut system have shown promising results and more thorough experimentation is underway. This paper reviews the relevant previous work in HCI and psychology. A description of the symPut system and applications follow. I conclude with a brief description of my overall research plan and current completion status.

PREVIOUS WORK

Human Bimanual Capabilities

The amount of interlimb interference in two-handed dual target selection was first measured by Kelso *et al.* in 1979 [16]. The authors measured the average total response time of 12 subjects reaching for a target with their left hand, reaching for a target with their right hand, and reaching for two targets using both hands. They found that when the two targets were located equidistant from the starting points, that dual hand target acquisition resulted in mean total response times appoximately 5% slower than the maximum of the response time with their left or right hand. The authors found more interlimb interference when the distances to the two targets varied in the dual target selection condition. These results suggest that for mirrored movements, two targets can be acquired much faster using two hands than if the targets are acquired one at a time by a single hand (*i.e.,* benefits from parallelism are likely to outweigh the small costs associated with interlimb interference).

In 2001, Franz *et al.* [10] showed that when a subject can perceptually integrate two unimanual tasks into a single unified task, interlimb interference is reduced and performance is improved. Additionally, Mechsner *et al.* [21] did a study of mirrored symmetrical and asymmetrical movements of homologous and non-homologous muscle groups, using finger rotations. They found that subjects' movements converged to symmetrical finger rotations, whether the finger muscles being used were homologous or non-homologous. They assert that the interlimb symmetry bias is perceptual, rather than related to muscle group activation, that perceptual goals are important, and that motor activity can be "flexibly tuned in".

Finally, in a recent journal article, Diedrichsen *et al.* [8] presented a number of experiments aimed at locating the source of bimanual interference in target selection tasks. They replicated earlier experiments on target selection and found that interlimb interference arises during the selection or decision process rather than during motor programming. When the targets to be selected were directly cued, rather than abstractly encoded, the interlimb interference virtually disappeared.

These results indicate that users *are* capable of using their two hands together at similar levels of spatial and temporal detail. Problems of interlimb interference appear to be associated with the planning and perceptual organization of the task or tasks being performed and have nothing to do with motor capabilities. This suggests that bimanual symmetric interaction in the computer interface has interesting potential, provided the tasks perceptually unify the two hands.

Symmetric Interaction

There is very little research about interaction symmetry in the user interface, where the two hands work at similar levels of spatial and temporal detail. Typing is an obvious example of symmetric interaction, and people can develop this skill with relative ease, but it is limited to textual input. Spatial symmetric input has received much less attention. Balakrishnan and Hinckley studied two-handed target tracking [2], examining the level of parallelism and interaction symmetry exhibited by subjects using two hands to track two moving targets. They found that increasing the level of visual integration between the two tracked objects led to increases both in the level of interaction symmetry and in the level of parallelism (measured by the percentage of time both hands were moving simultaneously). These observations led to the recommendation that objects being manipulated in a symmetric interaction be visually connected. The amount of interaction symmetry exhibited was not affected by dividing attention between the targets (*i.e.*, separating them further on screen) or by increasing the task speed. However, these two factors did decrease the amount of parallelism exhibited by users. Thus, their results suggest that as long as the objects being manipulated are visually connected, their distance on screen is irrelevant to the ability of users to work symmetrically. However, as the objects move further apart, the parallelism of interaction degrades and users switch hand use in a serial fashion. These results are in accordance with the psychology literature that postulate perceptual rather than motor causes of interlimb interference.

Similarly, the two-handed rectangle drawing study of Casalta *et al.* compared symmetric and asymmetric techniques [6]. Their study showed that the symmetric technique led to higher performance and more parallel interaction. This implies that the symmetric task representation is actually easier to work with and may reduce cognitive load.

Llamas *et al.* presented a technique for two-handed editing of 3D shapes, which allows interaction symmetry using two Polhemus Fastrak trackers [19]. In their system, the symmetric grab-and-drag interaction for deforming shapes was found to be intuitive and effective. When the user is not deforming objects, the system has an asymmetric interface for other types of object editing. Their technique shows that interaction symmetry has great potential. However, the authors do not specifically discuss issues involved in the use of interaction symmetry, nor do they give any guidelines for designing interfaces that afford interaction symmetry.

THE SYMPUT SYSTEM

The symPut system is a library for separately accessing input from two mice. It is written in C for Linux operating system. An application using symPut has access to mouse button and motion events from two mice simultaneously. This data can be used to put two cursors on screen and handle button or scroll wheel input from both mice. The mice can be of any type: optical or ball, with or without a scrollwheel, and with any number of buttons. The only restriction is that both mice must be identical in the number and type of actuators provided and both must be USB mice. The symPut system makes use of the Linux event module, and may eventually evolve from a library into a device driver.

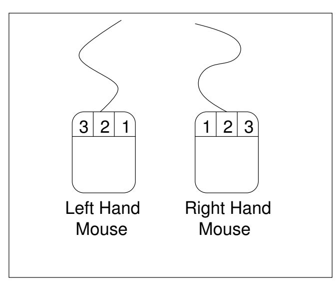

Figure 1: Bilateral Function Symmetry

The purpose of this system is to allow and encourage the application programmer to create a *function-symmetric* interface, where the actuators on the two mice are mapped in a mirrored symmetric fashion across the vertical midline of the body (see Figure 1). symPut allows the programmer to link an action to a specific actuator, such as the main (inner) mouse button or the scroll wheel, and not worry about whether the user invoked that actuator on the left mouse or the right mouse. This simplifies the task of programming two-handed interfaces. Of course, the device that generated the event is always passed to the application program, as it is sometimes necessary to know which mouse generated the event (*e.g.*, if drawing two cursors, an X-motion event must cause the correct cursor to be redrawn in the new position).

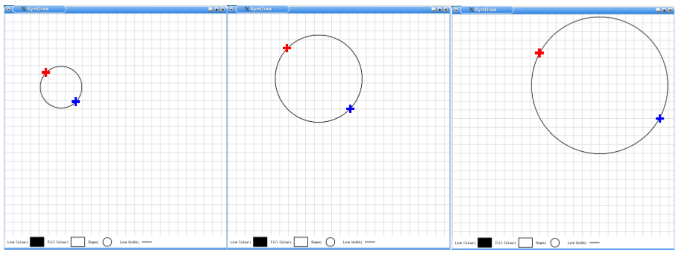

Figure 2: *symDraw* Symmetric object creation affords scaling and translating without mode switching

symDraw

symDraw is a bimanual drawing program that has a completely symmetric interface. In symDraw, creation of objects requires the use of both hands: objects are created by clicking Button 1 on both mice and dragging (see Figure 2). When the object is sized and positioned correctly, the user releases both buttons. This technique has the benefit of allowing sizing and positioning to be performed in a single, intuitive gesture, rather than requiring mode switching, which is the standard technique in drawing programs.

Currently symDraw is a simple 2D drawing program. Lines and shapes can be drawn and manipulated, but there is no support yet for adding general polygonal shapes or text. A basic layering model is used, and the usual cut, copy, paste, undo and redo operations are available. Objects can be snapped to a grid. In addition, a bimanual colour selection tool is included [17]. The symDraw interface is very plain: all of the program functionality is accessed via transparent, popup, radial menus. This enables an interface that is presentation symmetric: there are no menus or buttonbars in fixed positions in the interface, as these create an asymmetry and lead to a hand-use preference for choosing menu items or clicking buttons. Because all program functionality is available through popup menus, and the menus can be instantiated by either hand, the interface is equally accessible across the handedness spectrum, so users to not have to adjust any settings for their hand preference.

symDrive

The next logical step beyond simultaneous scaling and translation is simultaneous rotation and translation. This is what symDrive accomplishes. symDrive is currently embedded in the symDraw application, but the intent is to create a standalone library that will allow symDrive to be included in any two-handed symPut application. symDrive works by creating a center of rotation at the midpoint between the two cursors. Moving the two cursors in the same direction causes the selected object to be pushed in that direction. However, as soon as one of the mice starts to move in diverging directions or with diverging velocities, the selected object will rotate. If both mice are moving and turning together, the object is translated and rotated simultaneously (see Figure 3). This can be better visualized if the reader imagines putting their two index fingers on opposite corners of a piece of paper on a desk. Then the two fingers can be used to push and turn the paper simultaneously until it is at the opposite end of the desk and rotated $90°$. As with the simultaneous scale and translation, simultaneous rotation and translation is an intuitive way to position objects in the plane. symDrive mimics the way people naturally move objects in real life and removes the need to switch between translation and rotation modes.

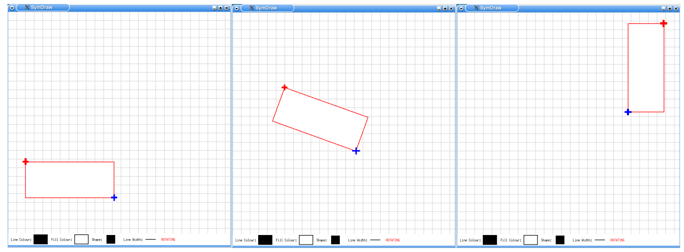

Figure 3: *symDrive* Symmetric interation affords rotating and translating in a single gesture

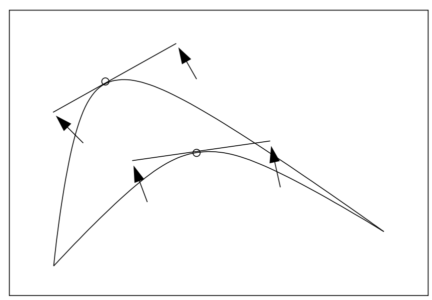

Figure 4: *symSpline* The two mice manipulate the ends of the control point tangent

symSpline

I am currently implementing a symmetric control point manipulation technique for splines. symSpline will allow a user to control the two ends of a control point tangent using two mice. If the user moves the two cursors such that the tangent is translated in the plane, the control point is translated. If the user moves the two cursors such that the slope of the tangent changes, the spline curve is appropriately altered. Most importantly, the symmetric two-handed method allows the user to translate and rotate the tangent in a single fluid gesture, allowing a spline curve to be easily controlled by the two hands (see Figure 4). Additionally, various symmetric techniques for initial spline creation are being investigated. symSpline will be incorporated into symDraw when it is complete.

User Studies

User studies of the symmetric interaction techniques being investigated are currently being planned. The techniques discussed here will be tested against asymmetric bimanual techniques as well as more typical single-handed techniques. Both performance and user satisfaction will be evaluated. It is hoped that the symmetric techniqes will demonstrate superior performance in the experimental tasks.

DOCTORAL PROJECT

The main objective of my doctoral research is to investigate symmetric bimanual interaction. To do this I have created a symmetric input system and the test application described above. The remaining work to be done is described below.

- Complete the symSpline implementation. This implementation has just begun.
- Complete the symDraw implemation: although symDraw is very close to being complete, there are likely to be other interesting symmetric techniques that can be implemented.
- Run user studies of symmetric interaction techniques.
- Create of a set of guidelines for the design of symmetric bimanual interfaces. This requires experimental validation of the symmetric techniques described in this paper.

This research project was approved by my doctoral committee in 2003.

ACKNOWLEDGMENTS

I would like to acknowledge Bill Cowan, Charlie Clarke, Craig Kaplan, Tiberiu Popa, Stefanus Du Toit, Kevin Moule, Selina Siu, Elodie Fourquet, Erin Lester and the other members of the UW Computer Graphics Lab for their support.

REFERENCES

1. R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and W. Buxton. Digital tape drawing. In *UIST'99 Proceedings*, 1999.
2. R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *CHI 2000 Proceedings*, 2000.
3. I. Barakonyi, T. Fahmy, and D. Schmalstieg. Remote collaboration using augmented reality videoconferencing. In *Graphics Interface 2004 Proceedings*, 2004.
4. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: The see-through interface. In *Siggraph'93 Proceedings*, 1993.
5. W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–326. ACM Press, 1986.
6. D. Casalta, Y. Guiard, and M. Beaudouin-Lafon. Evaluating two-handed input techniques: Rectangle editing and navigation. *ACM CHI'99 Conference (Extended Abstracts)*, pages 236–237, 1999.
7. S. Chatty. Issues and experience in designing two-handed interaction. In *CHI'94 Proceedings*, 1994.
8. J. Diedrichsen, R. B. Ivry, E. Hazeltine, S. Kennerley, and A. Cohen. Bimanual interference associated with the selection of target locations. *Journal of Experimental Psychology: Human Perception and Performance*, 29(1):64–77, 2003.
9. M. J. Flider and B. P. Bailey. An evaluation of techniques for controlling focus+context screens. In *Graphics Interface 2004 Proceedings*, pages 135–144, 2004.
10. E. A. Franz, H. N. Zelaznik, S. Swinnen, and C. Walter. Spatial conceptual influences on the coordination of bimanual actions: When a dual task becomes a single task. *Journal of Motor Behaviour*, 33(1):103–112, 2001.
11. Y. Guiard. Asymmetric division of labour in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behaviour*, pages 486–517, 1987.
12. K. Hinckley, M. Czerwinski, and M. Sinclair. Interaction and modeling techniques for desktop two-handed input. In *UIST'98 Proceedings*, 1998.
13. K. Hinckley, R. Pausch, D. Proffitt, J. Patten, and N. Kassell. Cooperative bimanual action. In *CHI'97 Proceedings*, 1997.
14. P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. In *CHI'94 Proceedings*, 1994.
15. P. Kabbash, I. S. MacKenzie, and W. Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *InterCHI'93 Proceedings*, 1993.
16. J. Kelso, D. L. Southard, and D. Goodman. On the coordination of two-handed movements. *Journal of Experimental Psychology: Human Perception and Performance*, 5(2):229–238, 1979.
17. C. Latulipe, E. Fourquet, and W. Cowan. Two-handed colour selection. In *Graphics Interface '03 extended abstracts*, 2003.
18. A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive benefits of two-handed input. *ACM Transactions on Computer-Human Interaction*, 5(4):326–359, 1998.
19. I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw. Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Trans. Graph.*, 22(3):663–668, 2003.
20. I. S. MacKenzie and Y. Guiard. The two-handed desktop interface: are we there yet? In *CHI '01 extended abstracts on Human factors in computer systems*, pages 351–352. ACM Press, 2001.
21. F. Mechsner, D. Kerzel, G. Knoblich, and W. Prinz. Perceptual basis of bimanual coordination. *Nature*, 414:69–73, 2001.
22. D. L. Odell, R. C. Davis, A. Smith, and P. K. Wright. Toolglasses, marking menus, and hotkeys: A comparison of one and two-handed command selection techniques. In *Graphics Interface 2004 Proceedings*, pages 17–24. A.K. Peters, 2004.
23. C. Shaw and M. Green. Thred: A two-handed design system. *Multimedia Systems*, 5:126–139, 1997.