Revisiting Hovering: Interaction Guides for Interactive Surfaces

Victor Cheung
victor.cheung@uwaterloo.ca
Collaborative Systems Laboratory
Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada

Stacey D. Scott
stacey.scott@uwaterloo.ca
Collaborative Systems Laboratory
Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada

Jens Heydekorn
jheyde@isg.cs.uni-magdeburg.de
User Interface & Software Engineering Group
Otto-von-Guericke-Universität
Magdeburg, Germany

Raimund Dachselt
raimund.dachselt@tu-dresden.de
Interactive Media Lab
Technische Universität Dresden
Dresden, Germany

ITS'12, November 11–14, 2012, Cambridge, Massachusetts, USA.
ACM 978-1-4503-1209-7/12/11.

Abstract

Current touch-based interactive surfaces rely heavily on a trial-and-error approach for guiding users through the interaction process. In contrast, the legacy WIMP (Windows, Icons, Menus, Pointer) paradigm employs various methods to provide user assistance. A commonly used strategy is the use of mouse hovering. This research explores how this strategy can be adapted and expanded to user interaction with interactive surfaces to provide user assistance as well as to help address common surface interaction issues, such as precisions. Design dimensions and considerations are discussed, and potential hover interaction techniques are proposed. These techniques emphasize the use of animation to facilitate user engagement and improve the overall user experience.

Author Keywords

Hovering; animation; interactive surfaces; proximate interaction.

ACM Classification Keywords

H.5.2. User Interfaces – Interaction styles.

General Terms

Human Factors; Design.

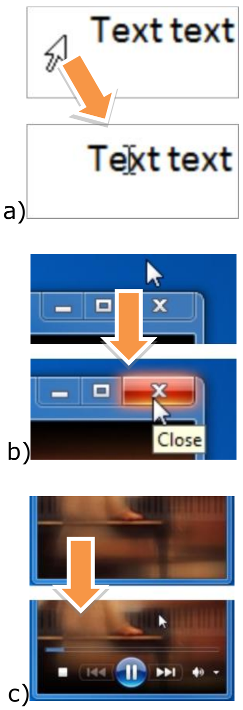

Figure 1. Strategies used with a cursor to provide extra position- and context-dependent information.
(a: shape change;
b: tooltip;
c: revealed controls)

Introduction

Touch-based interactive surfaces respond directly to input from a user's fingers (or hands), without the need for additional input mechanisms, such as a mouse, closely coupling the input and output channels. However, this input model also reduces the state of input to a binary touch or no touch, limiting the expressiveness of a user's input. While various gestures and on-screen menus have been proposed to support user interaction, they require users to learn and memorize steps, often achieved by trial and error.

In the traditional WIMP (Windows, Icons, Menus, Pointer) desktop interaction model, the ability to "point" to an item of interest using a system cursor without activating the item, enables application designers to use position and context information to convey additional information to assist users in their understanding of the interaction process. Such *hovering* (or *mouseover*) capability is commonly used to guide the user through the interaction process by cursor shape change and tooltips (short pieces of text), and timely reveal of controls (see Figure 1).

Hovering allows the system to provide feedback or hints to its user before an action occurs. At the same time, it preserves display space by hiding information or interface components until they are needed. These advantages are widely adopted in various computing applications for both novice and expert users. In contrast, basic touch-based interactive surfaces provide binary input only, and thus, an equivalent hover action is not possible to detect in these systems. However, additional sensors can be added to enable user interaction near the surface, or *proximate interactions*, such as optical sensing [9], or capacitive disruption detection (e.g. Cypress's Real Hover Technology[1]). This proximity sensing can then be used to provide hover-type interaction in surface applications, and provide the potential for designers to leverage the advantages of hover discussed above to improve the overall usability and user experience of touch surface applications.

The aim of this project is to adapt and expand traditional hover interaction design techniques on touch surfaces by taking advantage of proximity sensing capabilities. Moreover, this work focuses on the use of animation as a form of visual feedback to users' proximate interactions. To set the context for this work, we first briefly overview some related work. Next, we discuss our design concepts and future plans for the project.

Related Work

Several techniques have been proposed to bring back hovering to interactive surfaces by offsetting the touch location [3,7], or using an additional input level [4]. Though, they are limited to traditional behaviors of a mouse cursor without adapting to interactive surfaces.

Some recent work has investigated interactions close to a touch surface and generalizes them as *near touch interaction* [6,8]. In contrast to these interactions that rely on gestures or precise positioning of the hands, we focus on interface design that facilitates the fundamental touch input via animation, and the guiding ability it possesses.

[1] http://www.cypress.com/touch/hover.cfm

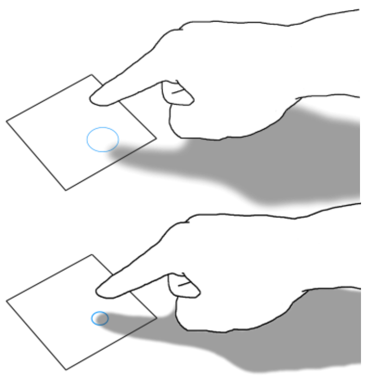

Figure 2. Touch position estimated by the system is shown during hovering as a shrinking halo (multi-state continuous hovering).

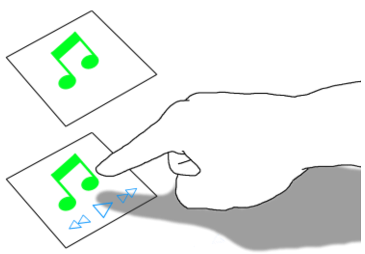

Figure 3. Playback controls appear as the finger approaches (2-state discrete hovering).

Design Dimensions and Considerations

With the additional input modality of proximity in an interactive surface system, an extended set of design dimensions is available with the following parameters: *context* (application status), *position* (planar finger position relative to surface), *proximity* (finger to surface distance), *granularity* (discrete or continuous input), *speed* (rate of approach or depart), and *angle*.

While several modalities of output are possible, this project focuses on the use of animation to provide visual feedback regarding proximate interactions. Previous research has shown that proper use of animation improves decision-making [5] and makes the system more enjoyable and comprehensible [1]. Thus, animation seems an appropriate choice for designing hover interaction on a touch surface that helps to guide user interaction and improve the overall user experience. Various animation parameters are being explored to understand their potential to support hover assistance, including but not limited to *movement* (positional change, rate) and other *appearance* attributes (size, brightness, color, transparency).

When designing animations as feedback to proximate interactions, the following design issues must also be considered:

- *Proximity range.* Proximity sensing provides an additional near and far dimensional range that can be leveraged to provide more sophisticated hover interaction, e.g., multi-state feedback. Also, as hovering is invoked and maintained by the *proximity* of the finger, the invoking height has to be carefully selected: too large creates unnecessary hovering detection and can confuse the user; too small increases the chances of unintentional touches.
- *Feedback complexity*. Duration of each hovering is likely to be short. Thus the visual response from the system has to be immediate and understandable: lengthy words and multiple steps should be avoided.
- *Feedback location*. The animation should take place at the immediate vicinity of the hovering *position*, where the user is currently focusing their attention.
- *Relationship between animation content and proximity*. The relationship between animation content and *proximity* should be consistent with the amount of information needed in the particular context. For instance, increased amounts of animated details could be provided as proximity increases. *Continuity*, *speed* and *angle* could also be considered as determining factors of the user experience context when appropriate.

Interaction Techniques

The project aims to leverage the directness of touch-based systems by adapting traditional hover techniques while providing visual responses for suggestive actions via the use of animation. The animation techniques are grouped based on the *eight uses of animation* proposed by Baecker and Small [1], as described below:

Informative. Informative techniques combine the purpose of *identification* and *choice*. When a hovering event is invoked, the estimated position of touch can be displayed, which is animated in proportion to the finger-surface distance for further cues (see Figure 2). This addresses the occlusion problem by making the estimated point of contact visible before being occluded. Also, some of the hovering techniques in the WIMP paradigm, e.g., appearing controls, can be used, allowing the user to decide ahead if touching at that position leads to the desired outcome (see Figure 3).

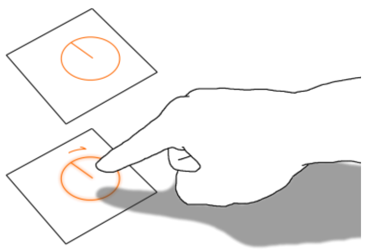

Figure 4. Control guides (glow and arrow) appear upon approach.

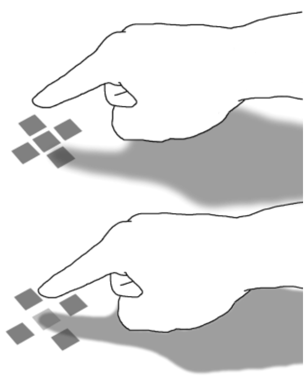

Figure 5. Items move away (and the to-be-selected item lights up) upon approach to prevent selection error.

Guidance. In a more elaborate manner, animation can be used for *demonstration*, providing *feedback*, and showing necessary steps as *guidance* to bring the user through the interaction process. For example, a control that requires adjustment can become increasingly animated (e.g., lights up) upon approach, and displays available options (e.g., arrows, preview of suggested movement) (see Figure 4). In addition, "feedforward" [2] can be used as a preview of an action's outcome.

Preventive. Instead of reporting a user-made error, animation can be used to proactively prevent the error from happening. The key idea is to visually inform the user ahead with the undesirable options, and with smooth transitioning (e.g., slow-in-slow-out) so the user will not be startled. For example, unavailable nearby items can move away slightly during hovering to prevent selection error (see Figure 5). This mitigates the precision problem by both lowering the chances of wrong selection and increasing that of the correct one.

Conclusion & Future Work

Like any other systems, touch-based interactive surfaces require ways to guide their users through the interaction process. We propose the use of hovering and animation to address this issue in a visual and engaging way. In particular, we propose using animation in response to proximate interaction to provide suggestive guides, as well as to deter infeasible interaction to improve the overall interaction process.

In the future we plan to evaluate the design concepts proposed here, with a focus on the subtleties of the implementation parameters, such as threshold of hovering and relationship between animation content and proximity. We will also explore the extension of hovering beyond a single finger. Finally, we intend to investigate how animation could be used during and after the touch event to guide the user further.

ACKNOWLEDGEMENTS

This work was funded by the NSERC SurfNet Network.

References

1. Baecker, R. and Small, I. (1990). Animation at the Interface. In L. Brenda (ed.) *The Art of Human-Computer Interface Design*. Boston, Addison-Wesley Longman Publishing Co., Inc., 251-267.
2. Djajadiningrat, T., Overbeeke, K., and Wensveen, S. (2002). But how, Donald, tell us how?: on the creation of meaning in interaction design through feedforward and inherent feedback. *Proc DIS 2002*, 285-291.
3. Esenther, A. and Ryall, K. (2006). Fluid DTMouse: better mouse support for touch-based interactions. *Proc. AVI 2006*, 112-115.
4. Forlines, C., Shen, C. and Buxton, B. (2005). Glimpse: A Novel Input Model for Mutli-level Devices. *Ext. Abstracts CHI 2005*, 1375-1378.
5. Gonzalez, C. (1996). Does animation in user interfaces improve decision making? *Proc. CHI 1996*, 27-34.
6. Moghaddam, A. B., Svendsen, J., Tory, M., and Albu, A., B. (2011). Integrating touch and near touch interactions for information visualizations. *Ext. Abstracts CHI 2011*, 2347-2352.
7. Potter, R. L., Weldon, L. J., and Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI 1988*, 27-32.
8. Takeoka, Y., Miyaki, T., and Rekimoto, J. (2010). Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. *Proc. ITS 2010*, 91-94.
9. Wilson, A. D. (2005). PlayAnywhere: A Compact Interactive Tabletop Projection-vision System. *Proc. UIST 2005*, 83-92.