

### Background

Pen-based interfaces aim to provide a natural way for users to create diagrams and take free-form notes. However, the integration of diagram creation and diagram editing remains a barrier to the widespread use of these interfaces. Because the user must use the pen for both drawing and editing, a core challenge for pen-based computing is to construct an interface that allows users to switch easily between the two tasks, while allowing the system to unambiguously interpret a given pen stroke as drawing or editing. Traditional solutions require the user to enter "edit mode," usually by pressing a software button. However, users often forget which mode they are in, attempting to edit while in drawing mode or vice versa. We aim to create a more natural and intuitive transition between drawing and editing.

### Current Approach – Edit mode and Lasso Select

Lasso select is the state of the art for selection, where the user clicks on a starting location and drags the cursor around the objects she wants to select. Windows Journal, as illustrated below, is a typical example of this.



In order to select anything, the user must click on the lasso button to enter selection mode. Lasso select needs this **modal change** because its selection strokes are hard to distinguish from drawing strokes. The problem with modal change is that it distracts the user from the current task and makes it difficult for her to remember which state she is currently in. Thus, Lasso can be cumbersome to use.

### Key Problems

- How can the user smoothly transition between drawing and editing using only the pen?
- How can they reliably perform drawing and editing actions without explicitly changing modes?



HoverCross extensively uses the Hover space, which is the area directly above the Tablet screen that tracks the pen's position as it hovers over the screen. The Hover space is a naturally separated area to unambiguously interpret selection strokes.

### Introduction to HoverCross

In many modal pen-based interfaces, it is selection, not editing in general, that is typically relegated to its own mode. Once the user selects an object, she can then edit it via a menu or direct manipulation. An explicit selection mode seems necessary because any apparent selection stroke just as easily could be a drawing stroke.

The key insight behind HoverCross is that relegating the selection to the hover space allows users to switch seamlessly between drawing and selecting without any extra effort. Furthermore, because the user performs only selection in the hover space, it suffices to use a simple crossing interface, where the user just crosses over the objects of interest. The system then leverages the context of the selection to give the user additional power through a gesture-invoked context menu or direct interaction with the selected objects.

### HoverCross Highlights

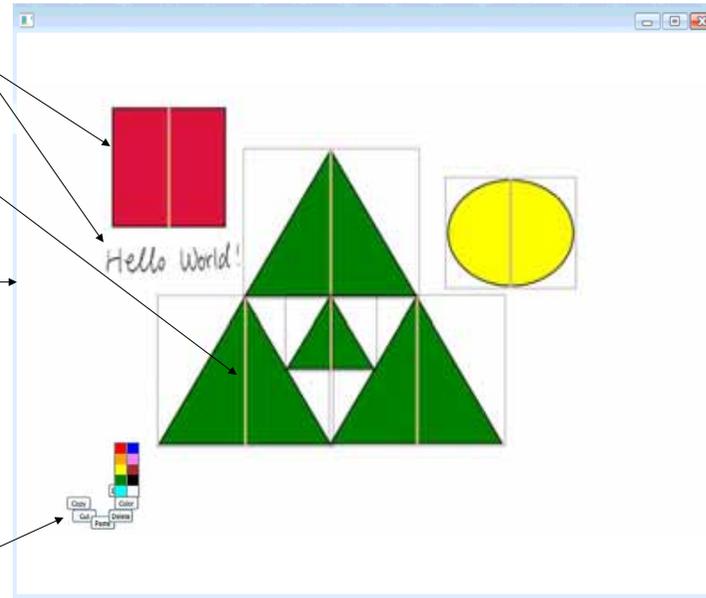
The user draws normally on the screen to create diagrams containing simple shapes and text. HoverCross could work with both recognized shapes and raw ink.

When the user hovers over the screen, vertical selection handles appear. Crossing these handles in the hover space selects and deselects objects.

The user can move the pen in hover space off the edge of the screen to clear selection.

When the user wants to select some shapes separated by others, she can easily cross over one shape, lift her pen, skip over the ones she does not want, and cross over the next one.

The user can move selected shapes by dragging them with her pen on the screen. For more complicated editing tasks, the user can draw a simple half circle to the right to bring up a ring-shaped context menu around the tip of the stylus. The content of the menu depends on the type of objects selected. Another short movement allows the user to quickly choose from the given options to edit the selection.



With little practice, the user can easily coordinate drawing, selecting, and editing in one fluid motion.

### User Feedback

We conducted a small user study to evaluate HoverCross.

- Twelve out of twenty users preferred HoverCross to Lasso.
- Overall, users felt that it was easy and fast to select objects with HoverCross.
- They enjoyed the modeless transition HoverCross offers between sketching and selecting, without the hassle of pressing a button or other explicit indication.
- HoverCross provided an easy way to select a small group of objects or objects spread out across the screen. Lasso is more cumbersome for these tasks.
- When objects are closely placed together, lasso is easier to use than individually selecting all the objects.
- Users can learn HoverCross really quickly. A few minutes of tutorial is all it takes for the users to start using HoverCross.

### Design Implication / Areas to Explore

Based on the constructive feedback the users provided, there are several things we want to do to make HoverCross even more user friendly and intuitive to use.

- We want to optimize the specific placement of crossing handles to deal with small objects and mostly occluded objects. HoverCross does not currently work well with overlapping objects whose crossing handles are covered up by larger shapes
- We want an interface that integrates HoverCross and Lasso. Since they each perform better on certain tasks, an interface that provides the flexibility to use either could be even more user-friendly.
- We want to further investigate the optimal time lag that occurs between when the pen enters hover space and when selection is triggered. User feedback on this varies, and it is worth carrying out more studies to see what the best time lag is, if any, and to allow the user to change its length.

### Acknowledgements:

I'd like to thank my research advisor, Prof. Christine Alvarado, and the HMC students who participated in the user study.

This work is supported in part by an NSF CAREER award (IIS-0546809) and by the Baker Foundation.