VoroPoint: Improving Gesture-Based Target Selection on Large Displays

Sven Kratz[1], Jim Vaughan[2], Maribeth Back[1]
FX Palo Alto Laboratory
3174 Porter Drive
Palo Alto, CA, 94304, USA
[1]{lastname}@fxpal.com,[2]jimv@fxpal.com

ABSTRACT

Existing user interfaces for the configuration of large shared displays with multiple inputs and outputs usually do not allow users easy and direct configuration of the display's properties such as window arrangement or scaling. To address this problem, we are exploring a gesture-based technique for manipulating display windows on shared display systems. To aid target selection under noisy tracking conditions, we propose VoroPoint, a modified Voronoi tessellation approach that increases the selectable target area of the display windows. By maximizing the available target area, users can select and interact with display windows with greater ease and precision.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]

1. INTRODUCTION

In environments such as conference rooms or lecture halls, which are equipped with a very large display, for instance a multi-monitor or multi-projector display wall, it is desirable to have multiple inputs as well as outputs. This way, content can be displayed from multiple input devices at multiple locations on the shared display in respective "display windows". Usually, dedicated display processors[1] are used to manage multiple inputs and outputs on a shared display.

However, it is common for these devices to have a relatively poor user interface for managing the assignment of input sources to output windows on the shared display as well as the location and size of output windows. Existing user interfaces usually depict a world-in-miniature representation of the display wall's state and allow the user to manipulate window positions and sizes via mouse input. It is our impression that the interface design is mostly geared towards administrators or technicians to set the configuration of the video wall at single points in time, and not for spontaneous interaction with the shared display by users of conference rooms or lecture halls.

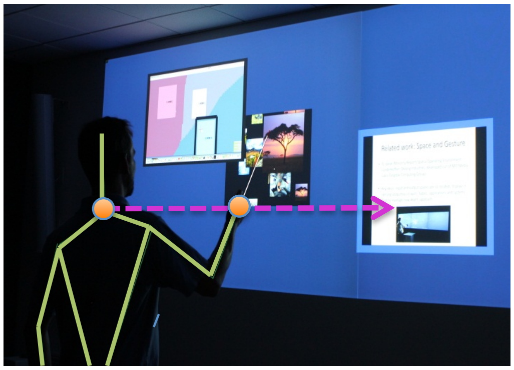

Figure 1: A gestural interface to control the position and size of display windows on a multi-source shared display space, based on skeletal tracking of the user.

In our lab, we have a conference room equipped with a large (3840×1080 pixels; 4×2.5 m size) display wall that consists of two projectors driven by an RGB Spectrum MediaWall display processor (Figure 1). We are in the process of redesigning the user interface for managing the display wall. In an initial step, we intend to improve the usability of repositioning and rescaling the items displayed on the wall in order to allow more direct and spontaneous configuration changes by users of the conference room. In order to achieve this, we include elements of Natural User Interfaces (NUI) [5], allowing direct gestural interactions with display windows on the wall. Our NUI implementation uses multiple Kinect depth imaging sensors with fused skeletal tracking. Due to the inherent noise of skeletal tracking using depth imaging[2], it is at times hard for the users to point precisely at small targets on the display wall. Input precision decreases with any increase in the display size.

In this poster, we present a method for maximizing the target area of elements on the display in order to improve selection under noisy tracking conditions. We assume that, in accordance with Fitts' Law, larger targets decrease the difficulty of manipulating items on the display wall.

---

[1] For instance, the RGB Spectrum MediaWall series `http://www.rgb.com/products/MediaWall4500/`.

[2] Specifically, we are using the skeletal tracking provided by OpenNI.

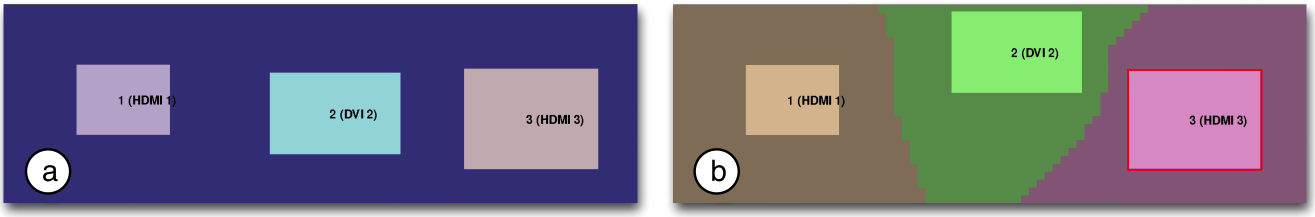

Figure 2: A standard world-in-miniature representation of a display wall showing the output of multiple image sources (a) offers only a limited target area for manipulation of display windows. VoroPoint (b) makes the entire display wall interactive by assigning screen pixels to display windows according to the distance from the window rectangles.

Previous work has covered how to improve pointing tasks for mouse-based interfaces and large screens [2, 4]. Variants of Voronoi tessellation for target selection have been proposed, e.g. [1] focuses on the problems associated with a high number of densely collocated target points, a scenario which is not directly applicable to our current prototype. Jota et al. analyzed pointing strategies for large displays; we used the best body-based technique they found [3].

2. LARGER INTERACTIVE AREA

We have implemented three basic operations for interacting with the display wall: window selection, window translation and window scaling. The users accomplish all three operations through pointing gestures captured via skeletal tracking. Selection is realized by pointing at the desired window. A window is selected after a certain dwell time. Once a window has been selected, and the keeps pointing within its bounds, the window can be translated on the display. Rapid pointing motions out of the window's bounds will cause a deselection of the window. Scaling windows is accomplished by keeping the window selected with the right hand and simultaneously raising or lowering the user's left hand above or below the display wall's horizontal centerline for enlarging or shrinking windows, respectively. We determine pointing location on the display wall by calculating the intersection of the neck-arm vector (obtained from the user's skeleton) and the surface of the display.

A standard world-in-miniature representation of the display wall's state is a set of rectangles, which represent the output windows inputs to the display wall, and a larger boundary rectangle, which represents the bounds of the media wall. In a naïve approach to target manipulation, each window can be used as a target area for manipulation operations, in particular to measure the dwell time for selection. We obtained very poor results when informally testing this technique, because the target area of the windows was too small considering the high noise in the tracking data obtained from the skeletal tracking.

To increase the tracking area, we propose a modified Voronoi-based tessellation scheme, *VoroPoint*. Our scheme divides up the entire screen space into selectable areas (Figure 2). Each targetable area is defined by the distance to the closest window rectangle. We modify the basic Voronoi approach by using the distance to a given window rectangle, rather than the distance to a point. This significantly increases the interaction area for each selectable target (i.e., window rectangle) and thus mitigates some of the problems incurred from high tracking noise.

The amount of simultaneous output windows is usually limited by the capabilities of the display wall hardware, and usually falls between 4 and 12 video input sources that can be displayed simultaneously. Thus, high-density clustering of targetable objects, as addressed in [1], is not a significant factor for our user interface.

3. DISCUSSION AND FUTURE WORK

Informal tests indicate that the method we propose has improved usability over a naïve approach. However, one possible issue is inadvertent selection, as every pixel on the display wall is made interactive through our method. This issue is mitigated by the fact that only pointing gestures that intersect the display are taken into account and that the dwell time is sufficient to avoid instant selection of the display windows. In the future, we plan to further refine our method to solve some special-case issues, such as exact overlaps between windows. Moreover, we intend to conduct formal user studies to asses with more precision how our approach increases usability. Finally, we want to add further gestures control further functions of the display wall, such as enabling or disabling display windows, or quickly maximizing or minimizing existing display windows.

4. REFERENCES

[1] P. Baudisch, A. Zotov, E. Cutrell, and K. Hinckley. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proc. AVI*, pages 129–137. ACM, 2008.

[2] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI*, pages 281–290. ACM, 2005.

[3] R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg. A comparison of ray pointing techniques for very large displays. In *Proc. Graphics Interface*, pages 269–276. Canadian Information Processing Society, 2010.

[4] M. Kobayashi and T. Igarashi. Ninja cursors: using multiple cursors to assist target acquisition on large screens. In *Proc. CHI*, pages 949–958. ACM, 2008.

[5] D. Wigdor and D. Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Morgan Kaufmann, 2011.