The Proximity Toolkit and ViconFace: The Video

Rob Diaz-Marino
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, Alberta
Canada T2N 1N4
rob.diazmarino@gmail.com

Saul Greenberg
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, Alberta
Canada T2N 1N4
saul.greenberg@ucalgary.ca

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract
Proximity Toolkit is a toolkit that simplifies the exploration of interaction techniques based on proximity and orientations of people, tools, and large digital surfaces. ViconFace is a playful demonstration application built atop of this toolkit. A cartoon face on a large display tracks a person moving around it, where it visually and verbally responds to that person's proximity, orientation and wand use. The accompanying video illustrates all this in action.

General Terms Design

Keywords Toolkits, proximity, situated interaction

ACM Classification Keywords
H.5.2. [Information Interfaces]: User Interfaces — input devices and strategies; interaction styles; prototyping; user-centered design.

Introduction
The Proximity Toolkit simplifies the exploration of interaction techniques based on proximity and orientations of people, tools, and large digital surfaces. ViconFace is a playful demonstration built atop this toolkit, where a face on a large display tracks and responds to people moving around it. This paper and video summarize and illustrate both systems.

Proximity Toolkit

One of our research goals is to investigate human computer interaction based upon the proximity of people, physical objects, and digital devices located in the everyday world. Our vision is that digital objects would react appropriately to proximal cues, where the interaction experience would build upon people's natural expectations from both a social and physical perspective (e.g., [1,2,3]). This is a very open-ended goal, where particular research directions could easily spawn into a variety of quite different projects and implementation directions.

Consequently, we decided to build a toolkit that would simplify the exploration and iterative design of interaction techniques and applications based on proximity and orientation of people, tools, and large digital surfaces. The result was the Proximity Toolkit. Through the toolkit's API, developers can easily:

- track particular objects of interest, where objects are tools, large displays, and/or people (our current repertoire includes a hat, a wand, an oval-shaped see-through mirror, a large plasma display, and a large digital table (e.g., see Figure 1, top)
- For each object, a developer can track its 3D position, orientation, and velocity (Figure 1, bottom)
- For a set of objects, a developer can track their distance and orientation to one another, whether they are touching or collided, and intersection points for directional objects (via ray-casting) (Figure 1, bottom)

The Proximity Toolkit currently wraps a Vicon motion capture system: a high-end installation that tracks infrared reflective markers affixed to objects. Our API hides the low-level details of the underlying Vicon system: it transforms and returns information about objects in terms of their proximity semantics. No knowledge of 3D math is required to use our toolkit.

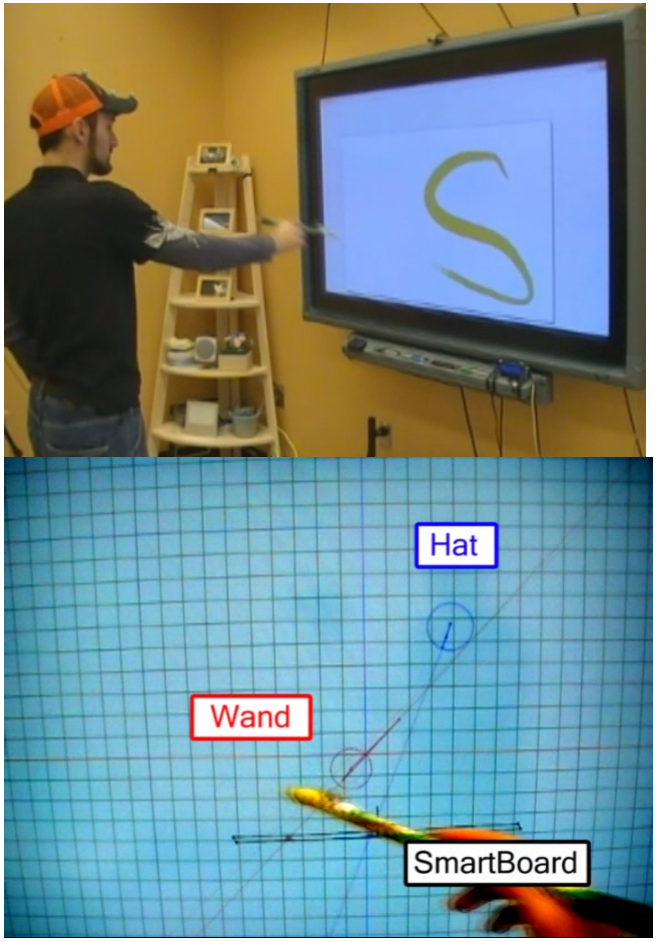


Figure 1. ***Top***: The person's hat, wand, and display all have markers. ***Bottom:*** The Proximity Toolkit tracks the relationship of these two tracked objects and the display.

The actual API is composed of several main objects. A ViconComponent receives, processes and manages raw data from the Vicon System (i.e., it does all the heavy lifting). A ViconSpace provides a high-level framework for monitoring the interaction of objects and displays. A DisplayPlane is a special object that represents a large digital display in 3D space. Tracked objects, such as a wand and hat, represent individual objects.

Initializing all this takes but a few lines of code, where one establishes a connection to the Vicon system via the ViconComponent, initializes and calibrates the display via the DisplayPlane, initializes the various tracked objects, and then adds these objects to the ViconSpace.

The program then receives events about tracked objects (called a *subject*) through a number of events, e.g., SubjectMove. When relationships between particular objects are of interest, the programmer creates a SubjectRelation (for two subjects) and/or a PlaneSubjectRelation (for a subject and a display plane). Events are then raised specific to these relations. For instance, if a subject (such as a person or a wand) is pointing at the display, a PointMove event is raised, where it returns parameters such as the distance of the subject from the display, intersection of the point on the display in screen coordinates, the angle of the subject to the display (i.e., its orientation relative to the display), whether the subject is actually touching the display, and so on.

ViconFace

ViconFace is a playful application built atop this toolkit, where its purpose is to demonstrate the toolkit capabilities (See Figure 2). ViconFace lives on a wall display. Its animated cartoon face tracks a person (via the person's hat) walking around it, and a wand that the person is holding. It continuously reacts as follows.

- *Position*: its eyes follow the moving hat and wand.
- *Orientation*: its face is happier when the person is looking at it, and sad when the person is turned away.
- *Distance*: it has a larger smile as the person comes closer, although it starts to frown if the person comes too close.
- *Velocity*: it is startled when the person moves quickly, and is fearful of sudden motion of the wand.
- *Pointing*: it visually fixates on the wand's beam when the wand is pointed at it.
- *Touch*: It becomes angry when it is poked in the eye by the wand.
- *Presence:* It notices when people enter/leave the scene, where it becomes happier or lonelier.

ViconFace also talks, where its speech reflects its reactions. While all attributes are triggered independently, the combination effect gives ViconFace a surprisingly realistic and compelling behavior.

Figure 2 illustrates several scenes from the video, where ViconFace is reacting to particular situations.

Conclusion

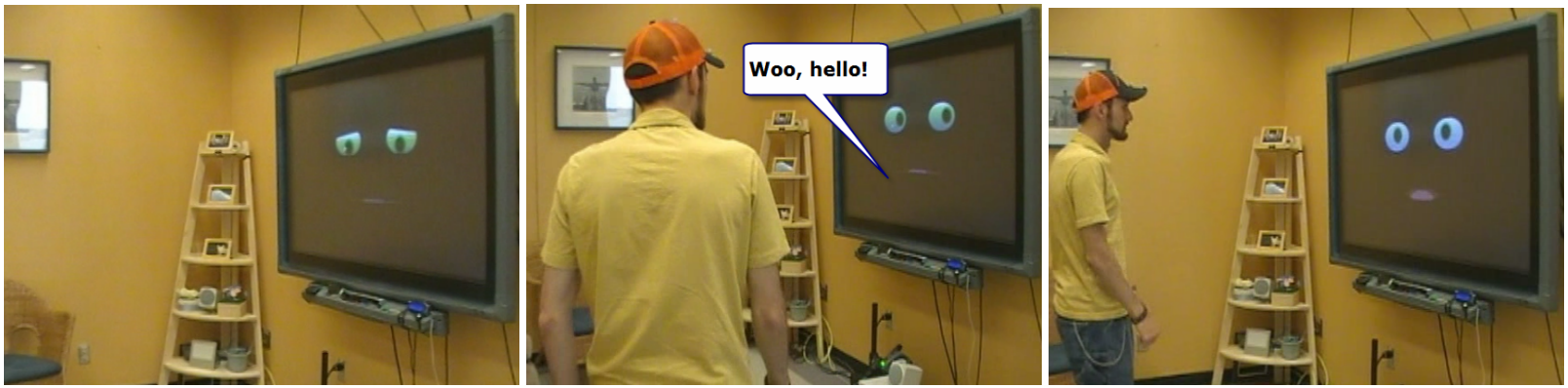
The ProximityToolkit is a first attempt at creating an easy to use toolkit that continuously tracks proximity and orientation of people, objects, and displays. While it is currently based on Vicon hardware, we believe its API will generalize to other tracking systems. One issue is how the information captured and delivered by our toolkit can degrade gracefully based on limitations of these other systems. Another issue is how we can design applications that react continuously as a function of proximity (as seen in the ViconFace), rather

than discretely across distance boundaries e.g., near, middle, far.

ViconFace is just a demonstration. We have created other demonstrations, such as a paint program whose drawing tools react differently based on a person's distance and orientation from the display, and a video viewer that displays videos and associated information as a function of proximity. This is just the beginning.

Acknowledgements

This work was partially supported by the



a) ViconFace; lonely when no one is around.
b) It sees Rob come in, and greets him
c) Looks at Rob when Rob looks at him,
d) but is saddened when Rob looking away.
e)Initially fascinated by the flashlight beam, it is annoyed when Rob pokes him in the eye
f) Rob is a bit too close for comfort.

Figure 2. ViconFace in action. It continuously detects and responds to the person and the wand he is holding as a function of orientation, proximity and touch.

NSERC/iCORE/Smart Chair in Interactive Technologies. Thanks to Nicolai Marquardt, Ricardo Jota and Miguel Nacenta who helped develop applications for it.

Citations

[1] Vogel, D. and Balakrishnan, R. (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. Proceedings of UIST 2004 – the ACM Symposium on User Interface Software and Technology. 137-146.

[2] Ju, W., Lee, B. A., and Klemmer, S. R. (2008). Range: exploring implicit interaction through electronic whiteboard design. Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work (San Diego, CA, USA, November 08 - 12, 2008). CSCW '08. ACM, New York, NY, 17-26.

[3] Kortuem, G., Kray, C., Gellersen, H. (2005) Sensing and visualizing spatial relations of mobile devices. Proceedings of UIST 2005 – the ACM Symposium on User Interface Software and Technology, ACM, New York, NY, 93–102.