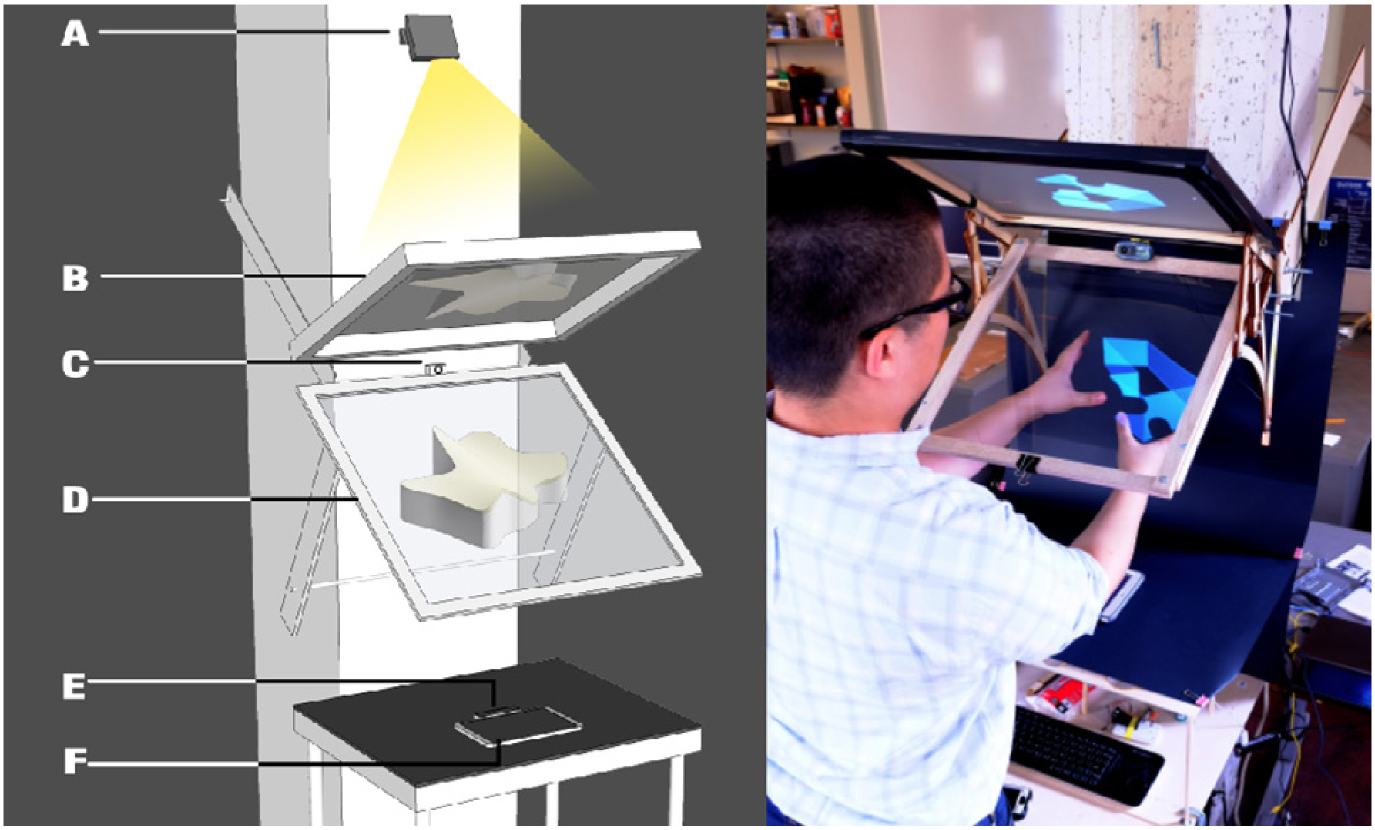
INSPIRE

INTEGRATED SPATIAL GESTURE-BASED DIRECT 3D MODELING AND DISPLAY

Teng Teng
Brian R. Johnson
University of Washington



1 Physical setup of our current InSpire prototype with main components labeled

ABSTRACT

In this paper we introduce *InSpire*, an interactive 3D modeling system combining an optical see-through "holo-display" and video-based motion sensing and head tracking to co-locate 3D model display and user gestures. Users can directly create, edit, and manipulate digital geometry, taking a step towards an intuitive gesture modeling environment that liberates designers' hands from the limitation of 2D mouse input and monitor output and InSpire designer's ideas. In this paper, we describe our goals and the concepts and implementation behind the prototype, on both the software and hardware side. In addition, we present several use case examples that explore potential applications. Finally, based on initial user responses to the prototype, some future development directions are discussed.

INTRODUCTION

The promise of “downstream” use of geometry data in CAD and analysis applications has led to increasing use of digital modeling in architectural schematic design, a phase of design long dominated by physical modeling and loose sketching. With the rapid improvement in computer graphics technology, 3D systems have multiplied, but most of them depend heavily on Window Icon Menu Pointer (WIMP) interfaces, using 2D design symbols to represent 3D design objects and operations. The content of the design, as well as the interaction between the designer and their design tools (mouse and monitor)–all the fundamental elements of design activity–are based on 2D representations. Some designers find these workflows, which derive from paper-based practice, to be awkward, fundamentally contrary to the 3D nature of architectural design.

Design requires the designer to create objects through integration of the mind, the hand and the eye (Lobel 2009). Architectural designing includes the activities *observing*, *thinking*, and *making*. An unexpected mistake of the hand may create a new opportunity for the eye to observe things differently, and cause the mind to find a new design possibility. A useful design aid should improve the association between the designer’s hand, eye, and mind, so that the intra-process communication would be enhanced (Kalay 2004).

In contrast, within most digital modeling methods, the function of hands has been weakened, replaced by the more indirect actions of mouse and keyboard use. We believe that the limited use of the hands reduces the excitation of eyes, and may inhibit an architect’s spatial cognition. Research comparing textual fluency and keyboard versus manual writing indirectly supports this (Richards et al. 2009). Therefore, while many great designs have been accomplished using 2D input, we seek to support a modeling process using direct gestural manipulation.

Several new technologies have appeared in recent years that have focused on enhancing the participation of hands during the entire design process. Multi-touch interfaces truly improve the interaction of hands, but still don’t escape the limitations of 2D input and output, which is the fundamental challenge of fully inspiring designers without transferring limitations.

InSpire combines an optical see-through display and gesture input, enabling users to directly interact with the system using spatial gestures in the same visual volume as the displayed geometry. The actions of input are 3D and output display is virtually 3D.

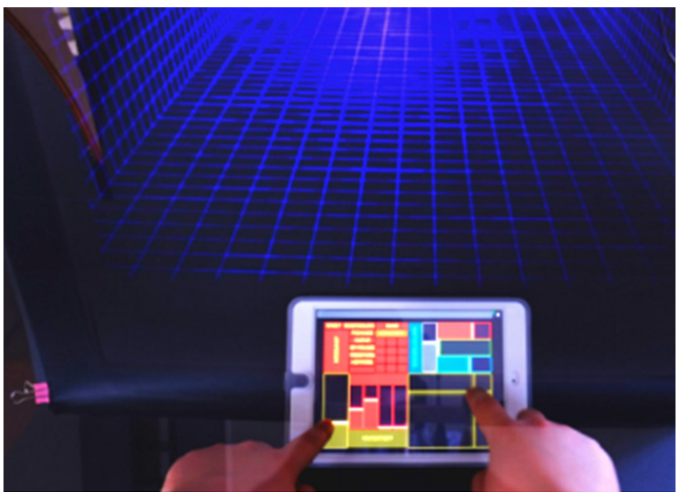
RELATED WORK: SPATIAL GESTURE INPUT

Compared with input methods such as mouse or touchpad, spatial gesture input provides users a more interesting interaction to manipulate digital devices. Recent research projects that helped us develop our project: Grandhi et al. (2011) discuss the definition of natural and intuitive gestures. They found (a) that a dynamic representation of motion makes interaction easier than static hand poses, (b) that containing any of the body tools is not recommended in the natural and intuitive gesture setting, (c) that two-handed manipulation provides a better sense of virtual space than one-handed, and (d) that the manipulation should be egocentric, as this will help users to ignore the awkwardness between users and system. Other research has focused more heavily on sensing devices, such as Wii Remote, Kinect, and Leap Motion. Fritz et al. (2009) demonstrated use of wireless motion-sensing objects, including the Wii Remote, to control navigation of 3D architectural models. Francese et al. (2012) present several projects using the Wii Remote and Kinect to navigate a map in a 3D digital environment.

However, navigation as a single function will not be enough for designers to do design job with spatial gesture. Hilliges et al. (2012) developed “HoloDesk,” a system that allows users to use their hands to pick up, move and even shoot virtual 3D objects, in addition to which the system recognizes and responds to the presence of inanimate real world objects such as a piece of paper or an upturned cup. HoloDesk is a good attempt to change the way that human beings interact with a computer, but it is still oriented to the needs of a regular user rather than a designer.

Other valuable attempts that explored the possibility that designers can get support from gesture input: Yi et al. (2009) presented a novel method of gesture modeling supporting architectural design. They define various components of architecture based on a set of signs. However, their gesture recognition depends heavily on the application of colored marker to the user’s hand and numerous motion capture devices. It is not a simple and inexpensive system. Gross and Kemp (2001) used video-based gesture recognition utilizing two simple orthogonal webcams to capture the 3D. Their configuration requires a white background and use of a black glove so that the location and shape of hand is accurately captured. It is relatively strict limiting users’ experience, and the system is limited to a few simple gestures, limiting application of their gesture modeling system to simple functions such as distorting a mesh or moving existing objects. Kumaragurubaran (2012) presented a hands-free prototype of gestural support for design using a Kinect to perform manipulation of a Grasshopper

2 Left: Glasses with two mounted LEDs. Right: Observing the same object from different angles



3 The tablet provides a control panel with different command buttons

(a Rhinoceros plug-in) interface and parametric model, but manipulating Grasshopper components is not directly interacting with digital geometry.

3D DISPLAY

When designing in 3D, designers always welcome a 3D display. Still, multiple orthogonal 2D representations (plan and elevation) are commonly used to unambiguously represent 3D objects for editing. One notable feature of the HoloDesk project is its display (Hilliges et al. 2012). It uses a semi-reflective horizontal optical film onto which a 2D image is projected from above. Users put their hands under the semi-reflective surface, where they see a combination of reflected and transmitted imagery. An RGBD camera tracks the user's hand positions within the 3D design volume. By coordinating display and gesture recognition, a unified visual experience is presented to the user wherein models can be directly manipulated using hand gestures as well as other physical objects. Because the display occurs between the user's eyes and hands, the holographic objects shown on the screen can be picked up and stacked on top of real world ones, and real hands can juggle virtual balls or play with a virtual smartphone. Another interesting approach, SpaceTop, has been developed by Lee et al. (2013). Using a configuration similar to the HoloDesk's, the prototype device uses a modified OLED screen as a monitor, with the keyboard placed behind the screen. Users can type on the keyboard by placing their hands behind the screen, where they can also manipulate digital objects on the transparent monitor from behind. Depth sensitive cameras enable the manipulation to be in real 3D, and users can raise their hands to catch the displayed windows and arrange them in the 3D space. Like HoloDesk, SpaceTop is oriented to regular users rather than designers. Lee also developed a motion-sensitive 3D display system (2008) by using the infrared camera in the Wii remote and a head mounted sensor bar. The system can accurately track the location of a user's head and render appropriate view-dependent (monocular) images to the screen. The display reacts to head and body movement as if it were a real window, creating a realistic illusion of depth and space and effectively transforming the user's display into a portal into a virtual world.

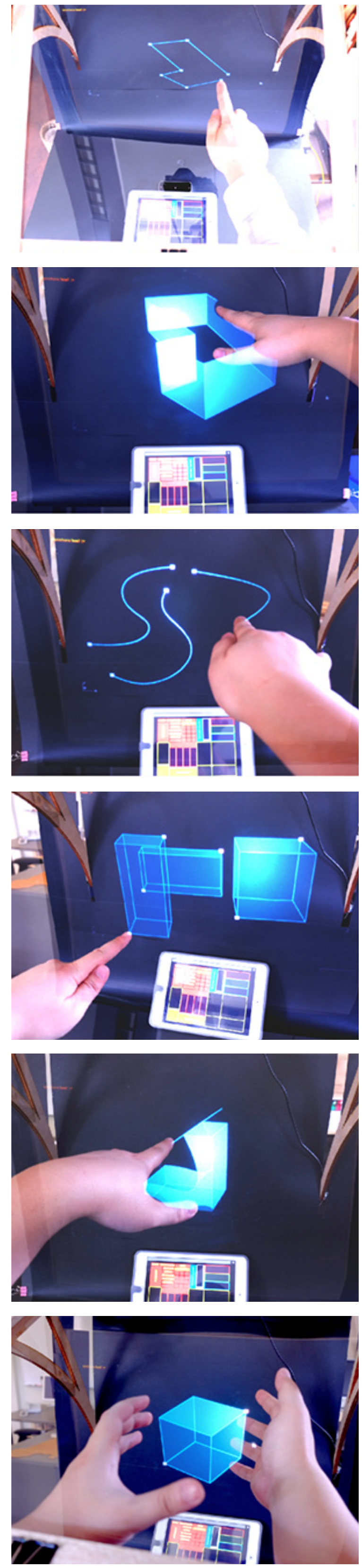
Building on these systems, our goal with InSpire is to bring 3D gesture input and a motion-sensitive monocular see-through 3D display together to support architectural design, allowing designers to interact directly with a 3D model using spatial gestures to create, edit, manipulate, and observe in real time.

INSPIRE OVERVIEW

The current physical configuration for *InSpire* is shown in (Figure 1). It is a single user installation consisting of a CPU (not shown), a mini projector (A), a rear projection surface (B), an RGB webcam (C), an adjustable semi-reflective plastic surface (D), a Leap Motion sensor (E) and a tablet (F). The virtual geometry is displayed on the semi-reflective surface, which acts as a see-through holographic screen. The size of the display screen and the volume below it roughly match the sensing range of the Leap Motion sensor. This configuration allows users to put their hands below the display screen without obscuring content, allowing their hands to directly (if virtually) "touch" the 3D model. The Leap Motion sensor placed on the bottom surface tracks the hand. The tablet placed next to the Leap Motion provides a control panel to switch between different functions. The semi-reflective surface reflects only the light parts of the projected 2D image; dark areas allow the user to see through to the space below the surface.

During project development we found that although the mirrored image on the display provided users some sense of depth, the absence of motion-parallax was a problem. In the physical world, when a viewer is observing an object, what they see changes with head position. In order to achieve this effect, we added an RGB webcam and head-mounted LEDs to provide head tracking. Visually, the user interacts with the mirrored shape in the display. The RGB webcam tracks the user's head position and automatically updates the viewpoint used to produce the model display. Since the RGB webcam is a 2D input device, head tracking requires a tracking target with extent, which we provided using two LEDs mounted at the hinges of a pair of glasses (Figure 2). Based on the angular separation between the two LEDs, the system calculates the distance to the head and generates 3D position data.

The tablet (Figure 3) in the system has two purposes. The Leap Motion is based on structured light in the IR spectrum, captured by two built-in monochrome infrared cameras. During testing we found that sometimes it did not seem sensitive enough to capture hands in dark conditions, but the extra light provided by the tablet has proven helpful. In addition, the tablet acts as a control panel for the system. Hand gestures are used to perform geometry manipulation—creating, scaling and rotating geometric elements—while more abstract operations with the object, such as file save, object delete and setting textures, which lack natural gesture mappings, are conveniently carried out on the embedded tablet. The tablet sends Open Sound Control (OSC) messages to the CPU via a Wi-Fi connection.



4 Modeling mode. Top to bottom: drawing polyline, extruding polyline, drawing curves, creating 3D volume, hotwire mode cutting object, moving and rotating an object

5 Top: Manipulating camera via trigger-mode

6 Bottom: Modeling mode gestures from left to right: push, click, drag, select and move & rotate

APPLICATION SCENARIOS

We have used the InSpire system to develop support for two designer-oriented applications and several interaction possibilities. The most mature and important application so far is the modeling function. The function leads to interesting natural modeling possibilities. A designer could build a 3D model with hand gestures just like sculptor and observe it from many different angles. Currently InSpire recognizes modeling mode gestures for drawing lines, polylines and curves, as well as creating single or multiple simple and complex surfaces and 3D volumes (Figure 4). After the geometry is created, users can edit an existing object via "hot wire" cutting gestures and control point "tweaks" to further develop the freeform geometry, as well as moving, scaling and rotating objects. As an architectural design tool, InSpire also allows users to set the geometry of different architectural elements on different layers.

The InSpire system can also be utilized to navigate a model. Users can control the camera, flying or walking through the entire project. For displaying an architecture project, the model view, which is generated by InSpire, could be shared on a large display, giving a larger audience access to the project. The navigation mode utilizes a different set of gestures from the modeling mode, and is called *trigger mode*. It can be used to view both the outside and the inside of a model, an improvement over the traditional physical model. (Figure 5) shows a user manipulating a camera flying through an urban model and focusing on a specific object.

SYSTEM IMPLEMENTATION

The InSpire system prototype is mainly built on top of Rhinoceros 3D and its parametric plug-in, Grasshopper. In order to support all the application and tentative ideas mentioned above, some other platforms and software are also used.

INPUT

As described above, the Leap Motion sensor captures the position, trajectory, and speed of fingertips and palm, producing a compact "hand" data structure. We transfer Leap Motion data into Grasshopper by using User Datagram Protocol (UDP) and then use a python script component in Grasshopper to extract the coordinates of the hand. Within Grasshopper, manipulation command gestures were defined-based on the number of fingers and palm, the speed of fingertips, and the vectors. For the head tracking feature, images are acquired from the RGB camera. After lowering the gamma value of the video frames, the LEDs are easily seen. Based on the 2D positions (as XY values) and the distance between the LEDs (as Z value), we compute a 3D head position, which is used to control Rhino's scene rendering. Grasshopper also receives commands from the tablet via Open Sound Control (OSC) messages, filtering the information through different definitions to activate modeling commands.

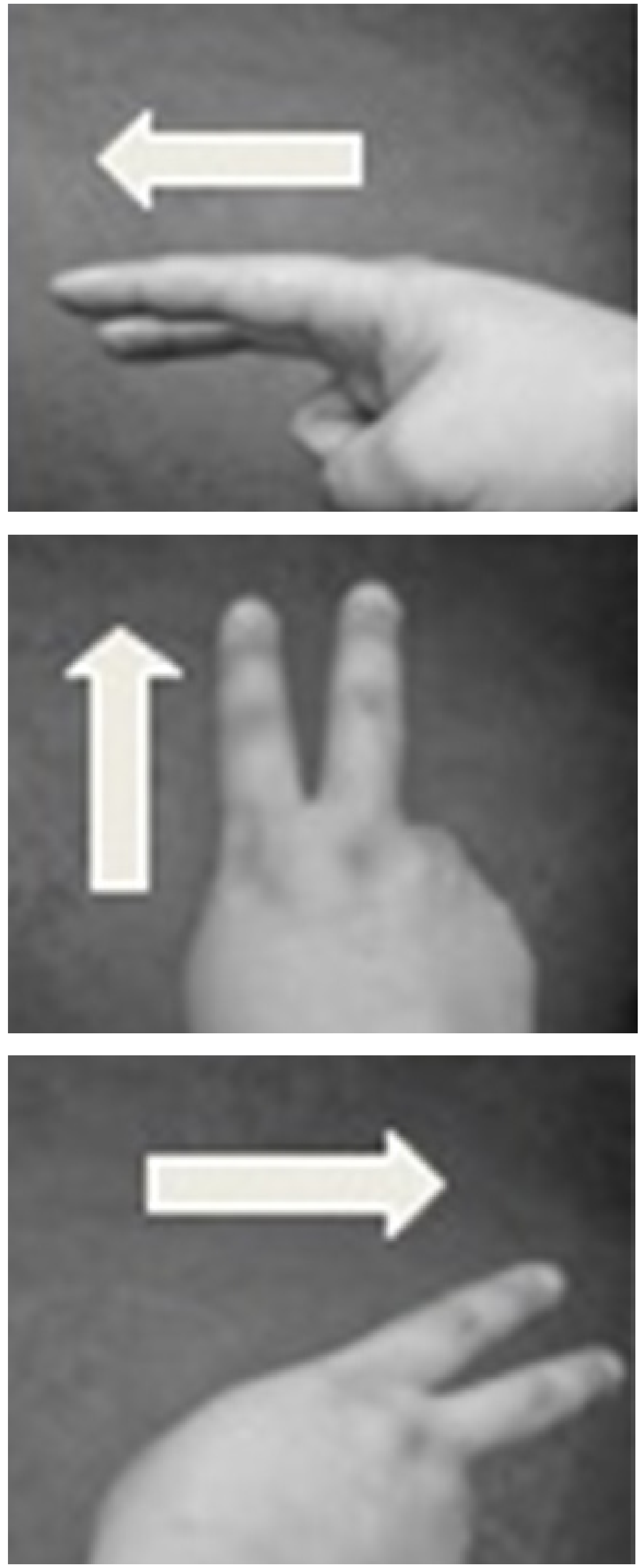
GESTURE RECOGNITION

The InSpire system uses two types of gesture, as mentioned above: modeling mode gestures and trigger mode gestures. The modeling mode gestures are mainly used to support direct manipulation of geometry in the modeling applications—including continuous drawing, scaling, rotating and selecting. Compared with the trigger mode gestures, the modeling mode gestures are more natural; users just need to open or close thumbs and palms (Figure 6). Trigger mode gestures (Figure 7) are used in the navigation application. Swiping two fingers defines a vector which the system applies as an adjustment to the camera view vector.

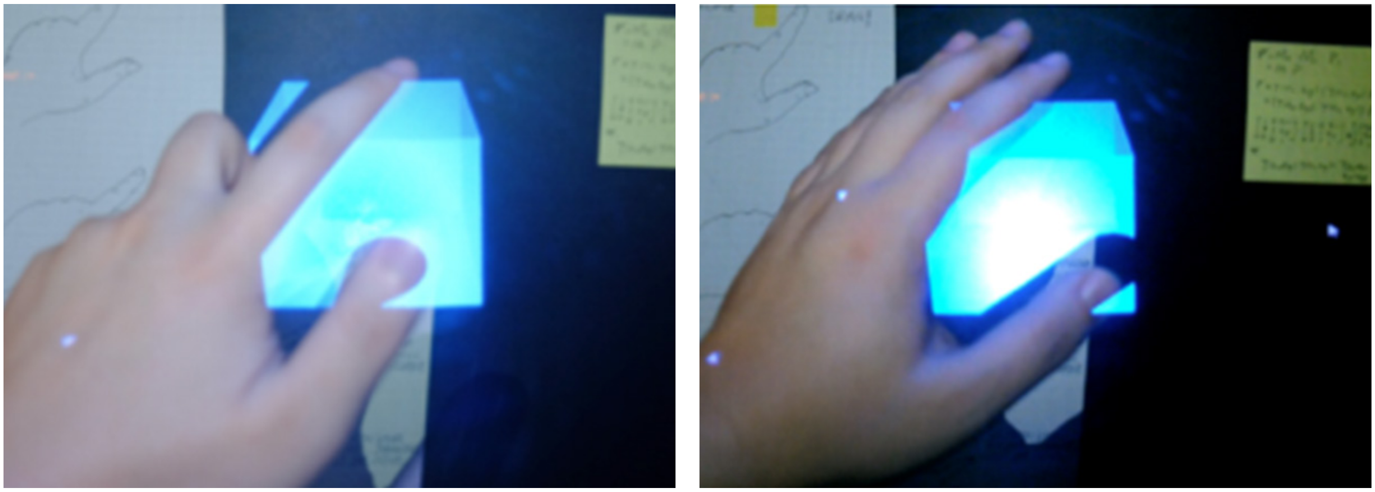
All the gestures are preset in the Grasshopper definition, based on features such as the number of fingers and palms and the movement of fingertips. Once a gesture's feature gets confirmed, a command is activated to manipulate the geometry. Using Grasshopper definition is possible to create NURBS surfaces and 3D volumes. When a model is complete, the user can use the tablet for saving files, setting layers or deleting objects.

DISPLAY OCCLUSION ADJUSTMENTS

Although the shape seen on the screen is a 3D projection, when users places their hands below the screen their sense of space may still be confused if they see the geometry displayed on top of their hand even when the geometry is located farther from their eyes than their hands. In the real world, parts of user's hand should block, or occlude, the geometry of the model, depending on the distance of hand and geometry from user's eyes. We address this issue in the way we render the hands. The Leap



7 Trigger mode gestures from left to right: turn left, moving forward, turn right



8 Occlusion display

Motion performs onboard image analysis and joint detection, so the data stream does not include a point cloud or depth mesh, but does report fingertip positions and the palm's center. Using the hand data, InSpire generates a 3D hand shape in the model and applies a flat black texture to it. When any part of the hand is in front of the holo-geometry, the rendered geometry will be partly or completely blocked by the hand's black shape. However, because users see through this part of the display (black is the background color and not reflected), they see their own hand in correct relation to the 3D model, enhancing their sense of immersion (Figure 8).

LIMITATIONS AND FUTURE WORK

The InSpire system is an initial prototype, with a number of issues and limitations. Some of these arise out of the hardware configuration we have chosen, while others are more fundamental. The fundamental difficulty is that architects must provide small-scale size and position control of large objects. Although the 3D "holo-interface" enables manipulations with one more dimension than available with a mouse, vision-based positioning is approximate. In the traditional CAD system, object snap features are available to make the modeling job more accurate. Without object snaps, InSpire remains useful in massing studies which occur at a very early phase in the entire design process, but is less helpful when working with complex models or detailed geometry. Related to this issue, the interface enables one to directly manipulate the model geometry, but under the current scenarios, it is difficult to measure the depth dimension accurately. Features such as head tracking and hand-model occlusion partially solve the problem, but another strategy, such as adding a projected outline of hands to both of side and bottom in the display, or voice-activated positioning constraints, might further enhance the experience.

Another fundamental problem is the lack of tactile feedback. In the physical world, when people are manipulating an object, there are at least two senses that help them feel in control of the object, visual and haptic. The InSpire system is now focusing on the visual interaction. With further development, the holo-geometry could be perceived not only visually but also tactilely. A touch sense could help to enhance the participation of hands. A potential strategy could be wearing tactile feedback gloves.

Finally, we note that while architectural design is frequently collaborative in nature, the current system is built for a single user. Duplicate workstations might share a model (see Hilliges 2012) allowing multiple modeling participants, but true collaboration requires a shared space of discussion as well as a shared model (Belcher and Johnson 2008), which will be difficult to achieve with the current approach.

CONCLUSION

In this paper we have introduced an interactive 3D architectural schematic design system called InSpire that allows users to directly interact with co-located virtual 3D objects using hands gestures. We described the system implementation, focusing on the workflow for supporting gesture interactions (visualization and gesture recognition). We illustrated several interactions that InSpire supports, and described application scenarios and limitations. We summarize our contribution as follows: developing a system allowing for freeform 3D geometry being created and manipulated through interactions based on user gestures, while leveraging the affordances of a semi-reflective holographic display and good hand position data to present appropriate spatial position cues through occlusion. This project highlights that it is possible and helpful for designers working in a total 3D environment.

REFERENCES

Belcher, D., and B. R. Johnson. 2008. "ARchitectureView: An Augmented Reality Interface for Viewing 3D Building Information Models." Paper presented at annual meeting of eCAADe.

Francese, R., I. Passero, and G. Tortora. 2012. "Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI." Paper presented at the ACM International Working Conference on Advanced Visual Interfaces.

Fritz, R., C. Hsiao, and B. R. Johnson. 2009. "Gizmo & Wiiview: Tangible User Interfaces Enabling Architectural Presentations." Paper presented at annual meeting of ACADIA.

Grandhi, S., G. Joue, and I. Mittelberg. 2011. "Understanding Naturalness and Intuitiveness in Gesture Production: Insights for Touchless Gestural Interfaces." Paper presented at the annual meeting of the ACM Conference on Human Factors in Computing Systems.

Gross, M. D., and A. J. Kemp. 2001. "Gesture Modelling: Using Video to Capture Freehand Modeling Commands." Paper presented at CAAD Futures 2001, Eindhoven.

Hilliges, O., D. Kim, S.Izadi, M. Weiss, and A. D. Wilson. 2012. "HoloDesk: Direct 3D Interactions with a Situated See-Through Display." Paper presented at the annual meeting of the ACM SIGCHI Conference on Human Factors in Computing Systems.

Kalay, Y. E. 2004. "The Roles of Communication." In *Architecture's New Media*. Cambridge, The MIT Press.

Kumaragurubaran, V. 2012. "A Multi-user Collaborative Space for Architectural Design Reviews." Published in extended abstracts of the ACM annual conference on Human Factors in Computing Systems.

Lee, J., A. Olwal, H.Ishii, and C. Boulanger. 2013. "SpaceTop: Integrating 2D and Spatial 3D Interactions in a See-through Desktop Environment." Paper presented at annual conference of ACM SIGCHI Conference on Human Factors in Computing Systems.

Lee, J. C. 2008. "Hacking the Nintendo Wii Remote." *Pervasive Computing* 7 (3).

Lobel, Josh. 2009. "Computer Aids to Design." In *Computational Constructs: Architectural Design, Logic, and Theory*, edited by S. Chen, and J. Lobel, Beijing: China Architecture and Building Press.

Richards, T., Berninger, V., Stock, P., Altemeier, L., Trivedi, P., and Maravilla, K. 2009. "Functional magnetic resonance imaging sequential-finger movement activation differentiating good and poor writers." *Journal of Clinical and Experimental Neuropsychology* 31(8): 967–983.

Yi, X., S. Qin, J. Kang. 2009. "Generating 3D architectural models based on hand motion and gesture." *Computers in Industry* 60(9): 677–685.

IMAGE CREDITS

All image credits to Teng Teng (2014)

TENG TENG is an MS in Architecture candidate in Design Computing at the University of Washington since 2012. Teng's interests include parametric design and practice, 3D modeling, responsive environments and robotics design. He is particularly interested in HCI issues in design, focusing on providing a more intuitive and instructive interaction between designers and design tools. Teng is also a founding member of NCF studio, the largest computational design platform in China. They seek to popularize the use of algorithm and parametric design in Chinese design firms and universities. NCF provides design firms with technical support and professional training, holds workshops and courses with universities and produces online teaching materials.

BRIAN R. JOHNSON is an Associate Professor of Architecture at the University of Washington. Johnson has been involved in design computing for over thirty years, in research, software development, pedagogy and theory. He served ACADIA as Webmaster and Steering Committee member for many years, in addition to service as Site Coordinator (1995) and President (1999–2000). He was given the ACADIA Award of Excellence in Research (2002), as well the ACADIA Society Award (2010). He is Director of the UW's Design Machine Group, and Director of the UW's MS in Architecture Program in Design Computing. His interests include computer graphics, small group collaboration, human computer interaction, web technologies, scripting, BIM, and responsive environments.