Projective Windows: Arranging Windows in Space Using Projective Geometry

Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, Seok-Hyung Bae
Department of Industrial Design, KAIST
joonhyub.lee | sang-gyun.an | yongkwan.kim | seokhyung.bae @ kaist.ac.kr

Figure 1. In a mock setup showing the flow of Projective Windows, (a) the user wishing to adjust the scale and position of an AR window (b) grabs the window, (c) moves it, (d) makes it bigger by bringing it closer, and (e) projects it to the desired position.

ABSTRACT
In augmented and virtual reality, there may be many 3D planar windows with 2D texts, images, and videos on them. Projective Windows is a technique using projective geometry to bring any near or distant window instantly to the fingertip and then to scale and position it simultaneously with a single, continuous flow of hand motion.

Author Keywords
Augmented reality; virtual reality; 3D windows management.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces-Interaction styles, Windowing systems

INTRODUCTION
We imagine a rich, situated future of computing, where minimal augmented reality (AR) or virtual reality (VR) gear worn over the eyes brings connected information from the internet and local nodes (the Internet of Things [1]) to the space around the user as interactive virtual elements [10]. Just like desktop PCs in the 80s and ubiquitous smartphones in the 00s [20], this new computing form factor affords exciting challenges to reimagine everyday computing and the user interface (UI) that facilitates it.

In doing so, we focus on the planar window in 3D space [4, 5, 6, 11, 18, 19], a rectangular 3D UI element encapsulating some 2D contents and controls, because, while much of the promise of AR and VR are in immersive 3D contents, many types of contents, such as texts, images, and videos will likely remain 2D. Thus, the window may be an essential building block, even in future UI, and we can imagine many windows of varying sizes and distances surrounding the user [17]. However, adjusting their sizes and arranging them in space can be difficult without a special interaction technique. We present one such technique (Figure 1a-e).

UIST '17 Adjunct, October 22–25, 2017, Quebec City, QC, Canada

ACM ISBN 978-1-4503-5419-6/17/10.
https://doi.org/10.1145/3131785.3131816

PROJECTIVE WINDOWS
Projective Windows is a spatial window management technique that uses projective geometry to enable the user to quickly bring a window at any distance to the fingertip. The user can then easily scale and position it relative to the geometric features of the surrounding environment, all in one continuous flow of hand motion and without the need for dedicated hardware controllers or UI widgets.

Making an Area Cursor
First, an open pinch gesture makes a circular area cursor that activates all windows that cross boundaries with it [7] (Figure 2a). The user narrows the selection by closing the fingers (Figure 2b), and completes the selection by making the tips of the fingers touch (Figure 2c); i.e., a "grab."

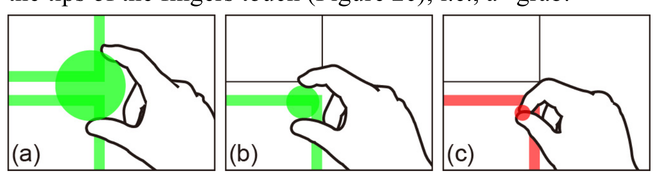


Figure 2. (a) The user makes a big area cursor, (b) specifies a window in a cluttered situation by closing the fingers and making the cursor smaller, and (c) grabs it.

Grabbing a Window
When the window is grabbed, it is instantly brought to the fingertip while maintaining the same *apparent* size, rather than the *absolute* size (Figure 3a, b) by reverse-projecting it to a picture plane defined at the fingertip [16]. Here, visual continuity is maintained, as the window *appears* the same to the user (Figure 3a inset, 3b inset).

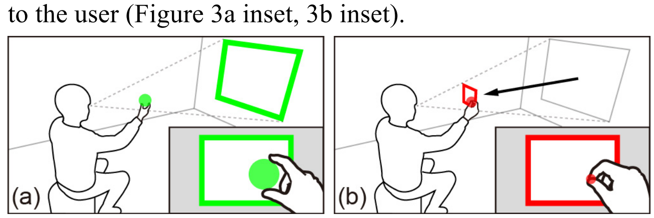


Figure 3. (a) The user makes a grab gesture on a window to (b) projectively bring it to the grabbed point.

Scaling and Positioning the Window

Once grabbed, the window's *absolute* size stays fixed, so the user can make the window *appear* bigger or smaller by bringing it closer to (Figure 4a) or pushing it away from the face (Figure 4b), just as with any physical object. At the same time, the user can move the grabbing hand to choose onto which surface to project the window.

The window is projected parallel to a vertical surface (Figure 4a, b) and erected perpendicular to the user's gaze on a horizontal surface to enforce the best viewing angle (Figure 4c). When the user releases the grab, the window is projected toward the surface with the same *apparent* size, which would have a larger or smaller *absolute* size compared to before the scaling operation (Figure 4a, b).

The different binding behavior follows a physical metaphor: a picture frame is hung parallel to a wall and erected on a desk. But, we may enforce different projection behaviors as needed; e.g., it may make no sense to erect a window against the surface of a tablet device, even when it lays horizontal.

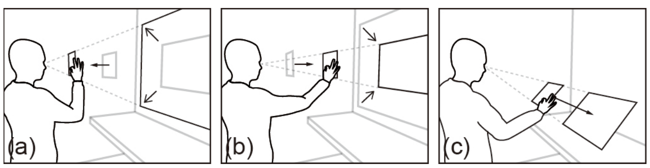
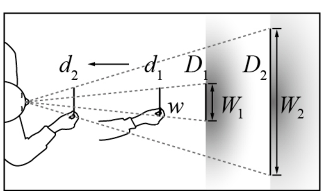


Figure 4. (a) The user makes the window appear larger by bringing it closer to the face and (b) smaller by pushing it away. The user can project a window (a, b) parallel to a vertical surface or (c) make it stand on a horizontal surface.

Zoom Factor

We estimate how much scaling a single grab–move–release can maximally produce. In the simple case where the user is directly facing a wall and grabs a window of width W_1 attached to a wall at distance D_1 using the hand at d_1, thereby reducing it to a fixed width w at the hand, moves it to d_2 and releases it to another wall at D_2 (Figure 5), zoom, defined as final W_2 divided by W_1, can be expressed as:



$$\text{zoom} = \frac{W_2}{W_1} = \frac{D_2 d_1}{D_1 d_2}$$

Figure 5. When the user grabs a window from a wall, moves it relative to the face and releases it to another wall, zoom can be expressed in terms of D_1, D_2, d_1 and d_2.

By substituting reasonable values for D (1 m: a wall just out of reach; 4 m: a distant wall) and d (0.1 m: closest to face; 0.4 m: reasonable arm extension) in the equation, we see that the window can be scaled by a factor of 16 through one grab–move–release sequence, demonstrating the benefit of projective geometry: The same zoom can be more tedious with techniques that operate in absolute sizes.

IMPLEMENTATION & USER SCENARIOS

We used an HTC Vive VR headset, a Leap Motion sensor, and Unity for a proof-of-concept implementation (Figure 6a, b), and prototyped user scenarios of Projective Windows in everyday computing. (The accompanying video to this abstract better captures the gist of the interaction scenarios.)

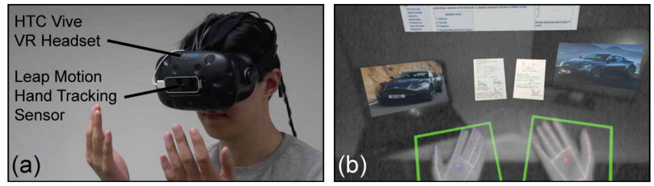


Figure 6. (a) Implementation hardware. (b) The hands, real and virtual objects in the user's view.

- *Scale & position anywhere*: In a design studio scenario (Figure 7a), the user can pull picture windows out of a laptop screen and easily scale and place them anywhere on the nearby walls for visual reference, just like sticky notes, but with the ability to freely change the size.
- *Cross-device jumps*: Also in the design studio (Figure 7a), the user can pick up a window from a laptop screen and place it on a tablet device to quickly change input from typing to drawing without having to swap applications.
- *Cloning physical objects*: In a study scenario (Figure 7b), the user can perform the grab gesture to instantly scan a notebook page and generate a projective window from it to scale and place it anywhere for reference.
- *Using proximity & geometry*: In a living room scenario (Figure 7c), the user can pick up a small movie window from a nearby table and play the preview of the movie by bringing it closer to the face [2] and then start playing the movie by projecting it on a vertical wall.
- *AR- and VR-compatibility*: In VR (Figure 7d), the user can utilize the entire scene, not bound by the physical room, as a workspace, even projecting across very large distances.

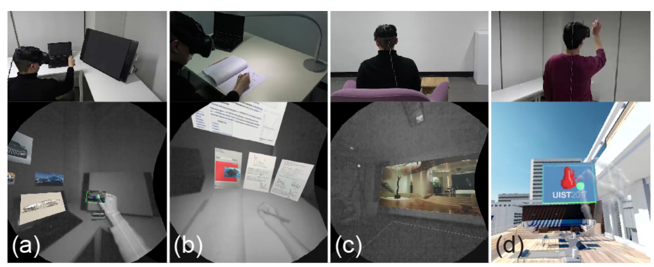


Figure 7. User scenarios of Projective Windows in (a) a design studio, (b) study, (c) living room and (d) a VR scene.

CONCLUSION & FUTURE WORK

We proposed a technique for managing planar windows in space, which is, thanks to projective geometry, minimal, direct and intuitive. We demonstrated the relevance and usefulness of projective geometry in AR and VR UI. Some speculate that AR and VR might replace all screen-based devices in the future [15]. Toward seamless interaction with 2D contents inside immersive 3D experiences and ensuring that Projective Windows is a part of that future, more work is needed on systematic use of surrounding geometries, such as wall edges and ceilings; thorough usability evaluation; and integration with other 3D window techniques [4, 5, 6, 11, 18, 19] and relevant spatial techniques [3, 8, 9, 12, 13, 14].

REFERENCES

1. Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: a survey. *Computer Networks*, 54(15), 2787-2805.
2. Till Ballendat, Nicolai Marquardt, and Saul Greenberg. 2010. Proxemic interaction: designing for a proximity and orientation-aware environment. In *Proc. ITS'10*, 121-130.
3. Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. 2010. Touching the void: direct-touch interaction for intangible displays. In *Proc. CHI'10*, 2625-2634.
4. Barrett Ens, Rory Finnegan, and Pourang Irani. 2014. The personal cockpit: a spatial interface for effective task switching on head-worn displays. In *Proc. CHI'14*, 3171-3180.
5. Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. 2014. Ethereal planes: a design framework for 2D information space in 3D mixed reality environments. In *Proc. SUI'14*, 2-12.
6. Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. 1993. Windows on the world: 2D windows for 3D augmented reality. In *Proc. UIST'93*, 145-155.
7. Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI'05*, 281-290.
8. Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: enabling touch interaction in display fixed UIs for mobile virtual reality. In *Proc. UIST'16*, 49-60.
9. Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proc. UIST'11*, 441-450.
10. Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Adj. Proc. UbiComp'13*, 307-310.
11. Jinha Lee, Alex Olwal, Hiroshi Ishii, and Cati Boulanger. 2013. SpaceTop: integrating 2D and spatial 3D interactions in a see-through desktop environment. In *Proc. CHI'13*, 189-192.
12. Joon Hyub Lee, Seok-Hyung Bae, Jinyung Jung, Hayan Choi. 2012. Transparent display interaction without binocular parallax. In *Adj. Proc. UIST'12*, 97-98.
13. Joon Hyub Lee and Seok-Hyung Bae. 2013. Binocular cursor: enabling selection on transparent displays troubled by binocular parallax. In *Proc. CHI'13*, 3169-3172.
14. Frank Chun Yat Li, David Dearman, and Khai N. Truong. 2009. Virtual shelves: interactions with orientation aware devices. In *Proc. UIST'09*, 125-128.
15. Kyle Orland. 2014. Will VR make flat panels obsolete? Oculus' founder gives it 20 years. *Ars Technica.* Retrieved July 12, 2017 from https://arstechnica.com /gaming/2014/04/will-vr-make-flat-panels-obsolete-oculus-founder-gives-it-20-years/
16. Jeffrey S. Pierce, Andrew Forsberg, Matthew J. Conway, Seung Hong, Robert Zeleznik, and Mark R. Mine. 1997. Image plane interaction techniques in 3D immersive environments. In *Proc. I3D'97*, 39-43.
17. George Robertson, Maarten van Dantzich, Daniel Robbins, Mary Czerwinski, Ken Hinckley, Kirsten Risden, David Thiel, and Vadim Gorokhovsky. 2000. The task gallery: a 3D window manager. In *Proc. CHI'00*, 494-501.
18. Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Gluey: developing a head-worn display interface to unify the interaction experience in distributed display environments. In *Proc. MobileHCI'15*, 161-170.
19. Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Desktop-Gluey: augmenting desktop environments with wearable devices. In *Adj. Proc. MobileHCI'15*, 1175-1178.
20. Mark Weiser. 1991. The computer for the 21st century. *Scientific American*, 265(3), 94-104.