Multiplanes: Assisted Freehand VR Drawing

Mayra D. Barrera Machuca[1], Paul Asente[2], Jingwan Lu[2], Byungmoon Kim[2], Wolfgang Stuerzlinger[1]
[1]SIAT, Simon Fraser University, Vancouver, Canada, [2]Adobe Research
[1]{mbarrera, w.s}@sfu.ca, [2]{asente, jlu, bmkim}@adobe.com

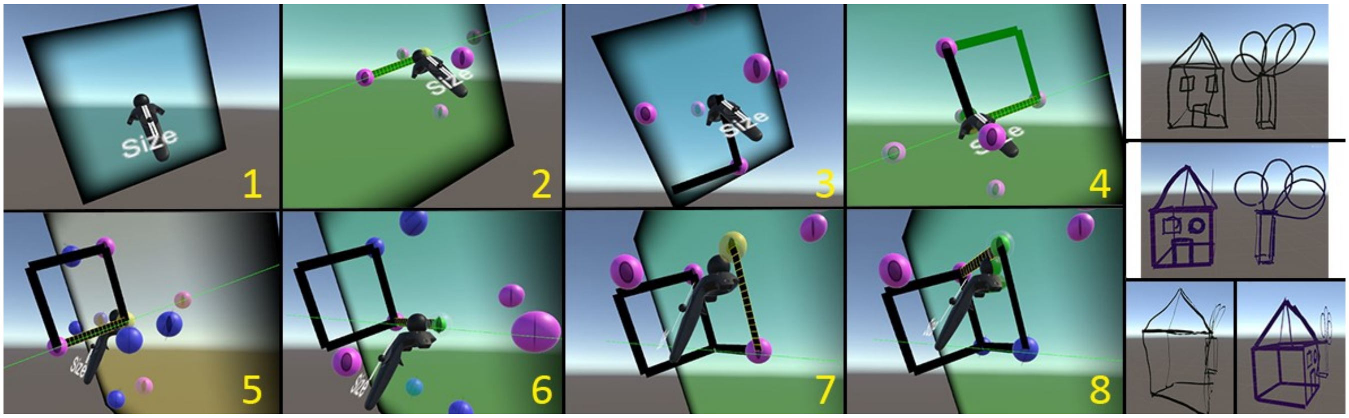

Figure 1: Step-by-step process of drawing two adjacent faces of a cube with the implemented system. To the right are examples of drawings done by participants in the pilot user study. The black house is freehand, the purple house is Multiplanes.

ABSTRACT

Multiplanes is a virtual reality (VR) drawing system that provides users with the flexibility of freehand drawing and the ability to draw perfect shapes. Through the combination of both beautified and 2D drawing, Multiplanes addresses challenges in creating 3D VR drawings. To achieve this, the system beautifies user's strokes based on the most probable, intended shapes while the user is drawing them. It also automatically generates snapping planes and beautification trigger points based on previous and current strokes and the current controller pose. Based on geometrical relationships to previous strokes, beautification trigger points act as guides inside the virtual environment. Users can hit these points to (explicitly) trigger a stroke beautification. In contrast to other systems, when using Multiplanes, users do not need to manually set or do any kind of special gesture to activate, such guides allowing the user to focus on the creative process.

CCS CONCEPTS

H.5.2 [Information Interfaces and Presentation]: User Interfaces – input devices, interaction styles

KEYWORDS

Virtual Reality Drawing; Sketching; 3D User Interfaces

1 INTRODUCTION

High quality virtual reality (VR) devices, such as the HTC Vive and the Oculus Rift, are now easily available. These products have rekindled interest in using VR technologies in the design process by letting users draw or sketch directly in a 3D virtual environment. Examples of commercial 3D drawing systems are Tilt Brush [18] and Gravity Sketch [6]. Most of these tools build on the freehand drawing technique, in which the stroke follows the controller position. Although this technique provides an intuitive and effective method of conceptualizing new ideas, which helps in the creative process [20], prior work showed that 2D sketching is more accurate and efficient than 3D sketching [1, 10]. Some systems solve this problem using novel interactions [2, 4, 7, 11, 13, 16], others beautify the strokes done by the user [5, 12, 14]. Previous work also shows that users have a strong interest in using the 3D space as a medium for conceptual design since they expect VR systems to provide new tools for the sketching process, help them be more creative, and perceive the 3D space better [10, 17]. Therefore, "new tools are needed that provide functionality that goes beyond straight one-to-one mappings of body movements to operations on digital geometry" [10, p.472]. However, unlike current CAD systems, it is important that these new sketching tools do not hinder the creative process or require extensive training to be used productively.

2 MULTIPLANES

We present a system called Multiplanes, which assists freehand VR drawing by automatically identifying drawing planes and guides based on previous and current strokes and the controller position and rotation. We use 3D freehand drawing because this technique is easy to learn and everybody can operate these systems regardless of their expertise with VR systems [20]. Moreover, 3D freehand drawing is powerful enough that even experienced users find their drawings satisfactory [15]. Figure 1 shows a working example of a house created in our system.

3D Stroke Beautification: In Multiplanes, every stroke is projected onto a plane. This approach encourages users to focus on the 2D

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

ACM ISBN 978-1-4503-5419-6/17/10.
https://doi.org/10.1145/3131785.3131794

movement of their arm, as changes in the third dimension will not affect the final stroke. Even after removing one DOF through projection, 3D drawing in the air is still difficult. To improve user accuracy, Multiplanes automatically beautifies strokes *in real-time* as the user is drawing them. It recognizes the stroke as a line, arc, or circle based upon the geometric relationship between controller positions. If the geometric relationships do not resemble any known shape, the system leaves the stroke as a general curve that follows the hand movement.

Beautification Trigger points (BTPs): Our system presents a new type of guides called beautification trigger points (BTP). They trigger actions only when the user starts or ends a stroke in them, and are embedded inside the 3D environment, so users benefit from multiple depth cues. There are different categories and types of BTPs depending on the respective parent stroke type (Figure 2). Our BTPs improve on previous work [3, 15] by showing geometrical relationships instead of just end points or intersections. Moreover, to create BTPs, we consider not only the closest existing stroke, but also the best drawing plane and the current controller pose. When users are not drawing, BTPs work as normal snapping points and effectively define starting positions on specific features of previous strokes and points that are in relevant geometric relationships to previous strokes. While the user is drawing, BTPs show the position(s) where ending a new stroke will snap the drawn content to that position and create a stroke with a defined geometric relationship, relative to previous strokes. To interact with BTPs users need to “hit” them with the controller. This explicit action helps the user correct mistakes, such as selecting a wrong snapping plane or performing a careless hand movement, by automatically orienting the drawing plane to map the current coordinate system to the local coordinate system of the BTP. Hitting a BTP also triggers an automatic beautification event that beautifies the current stroke section based on the BTP type. Finally, if a BTP from a line is hit, we create six standard BTPs, one in each axis of the local coordinate system, so users can benefit from a local reference frame, if desired.

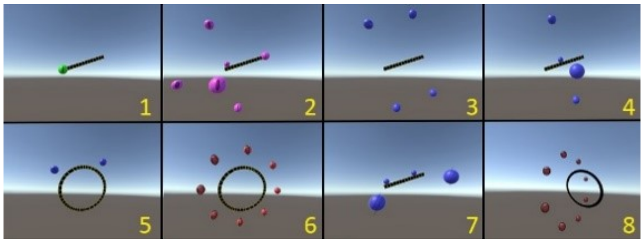

Figure 2: Different BTPs generated using different stroke shapes and the relationship between the plane and controller normals. 1) Vertex, 2) Local coordinate system, 3) Vertices forming lines parallel to the snapping stroke, 4) Vertices perpendicular to the snapping stroke, 5) Vertices forming tangent line to the circle, 6) Circle offset vertices, 7) Vertices lying on a plane parallel to the snapping stroke, 8) Circle offset for snapping plane.

Automatic Plane Generation: Instead of using fixed drawing guides [8] or mapping the plane to the view [19], Multiplanes uses the current controller pose and previously used planes to define the new plane. This removes the need for the users to press any buttons or do any gesture to define such a plane. In our system users see a visual representation of the drawing plane before starting to draw. Such visualizations have been shown to improve user accuracy in 3D drawing [8, 9]. The automatic generation of planes consist of two states. The first state is a feedback plane that is created while the user moves the controller in the virtual environment. The second state is the drawing plane, where the system displays a static planar visual guide to show the plane in which the stroke will be created. Every new plane created in the drawing plane state is stored so the system can use it later in the feedback plane state. A “Quickflip” interaction lets users change to a plane that is orthogonal to the one that was currently active if they move the controller far enough away from the current drawing plane (on either side) while drawing.

Other User Interactions: Multiplanes also allow users to change the stroke size and color. Stroke size is adjusted by moving the thumb vertically on the right-hand controller’s touchpad. Stroke color is adjusted by moving the left thumb on the left-hand controller’s touchpad. The color space is a HSL cylinder with saturation on the vertical axis and hue on the horizontal one.

3 Evaluation

We evaluated our prototype through a pilot user study. The objective was to solicit opinions of VR users about our new interaction method and to identify how they compare against the freehand drawing technique. In general, the results generated with Multiplanes look visually more appealing than the freehand results (Figure 1). Most participants liked the freedom of freehand drawing. Yet, the results also show that they appreciated the benefit of higher accuracy achievable with Multiplanes. Moreover, they rated the ease of interaction, automatic and unobtrusive beautification, and the BTP functionality high (average of 5.5 or better on a 7-point Likert scale).

4 CONCLUSIONS

We presented Multiplanes, a VR freehand drawing assistant that incorporates novel interaction techniques that help users be more accurate. For the stroke and current controller pose, Multiplanes automatically identifies an appropriate drawing plane. The system then also displays guides, called beautification trigger points, based on previous strokes, which show geometrical relationships to previous strokes and snapping points. Multiplanes also automatically beautifies a stroke in real-time while the user is drawing it or when users hit a BTP. Our pilot study identified that participants liked the system and appreciated the increased accuracy they could achieve with it. Especially the unobtrusive beautification and the trigger points were perceived as especially helpful.

ACKNOWLEDGMENTS

This research began as an internship at Adobe Research and was supported by CONACYT and NSERC.

REFERENCES

[1] Rahul Arora, Rubaiat H. Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 5643-5654. DOI: https://doi.org/10.1145/3025453.3025474

[2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST'2008). ACM, 151– 160.

[3] Eric. A. Bier. 1990. Snap-Dragging in three Dimensions. Computer Graphics, Symposium on Interactive 30 Graphics '90, 24(2), March 1990, pp. 193-204.

[4] Jonathan M. Cohen, Lee Markosian, Robert C. Zeleznik, John F. Hughes, and Ronen Barzel. 1999. An interface for sketching 3D curves. In *Proceedings of the 1999 symposium on Interactive 3D graphics* (I3D '99). ACM, New York, NY, USA, 17-21.

[5] Jakub Fišer, Paul Asente, and Daniel Sýkora. 2015. ShipShape: a drawing beautification assistant. In *Proceedings of the workshop on Sketch-Based Interfaces and Modeling* (SBIM '15). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 49-57.

[6] Gravity Sketch, https://www.gravitysketch.com/

[7] Cindy Grimm and Pushkar Joshi. 2012. Just DrawIt: A 3D sketching system. In Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling. Eurographics Association, Aire-la-Ville, Switzerland, 121–130.

[8] Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George W. Fitzmaurice, Azam Khan, and Bill Buxton. 2002. Creating principal 3D curves with digital tape drawing. ACM In *Proceedings* of the 2002 CHI Conference on Human Factors in Computing Systems. 121-128.

[9] Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George W. Fitzmaurice, Azam Khan, and Bill Buxton. 2001. Interaction Techniques for 3D Modeling on Large Displays. I3D, 17-23

[10] Johann H. Israel, Eva Wiese, Magdalena Mateescu, C. Zöllner, and Rainer Stark. 2009. Investigating three-dimensional sketching for early conceptual design—Results from expert discussions and user studies. Computers & Graphics 33, 4: 462–473. http://doi.org/10.1016/j.cag.2009.05.005

[11] Bret Jackson and Daniel F. Keefe. 2016. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. IEEE Transactions on Visualization and Computer Graphics 22, 4: 1442–1451.

[12] Michele Fiorentino, Giuseppe Monno, Pietro A. Renzulli, and Antonio E. Uva. 2007. 3D Sketch Stroke Segmentation and Fitting in Virtual Reality. In IEEE Transactions on Visualization and Computer Graphics

[13] Daniel F. Keefe, Robert C. Zeleznik, and David H. Laidlaw. 2007. Drawing on air: Input techniques for controlled 3d line illustration. IEEE Trans. Vis. Computer Graph., 13(5):1067–1081.

[14] Dominik. Rausch, Ingo Assenmacher, and Torsten Kuhlen. 2010. 3D Sketch Recognition for Interaction in Virtual Environments. Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2010)

[15] Ryan Schmidt, Azam Khan, Gord Kurtenbach, and Karan Singh. 2009. On Expert Performance in 3D Curve-drawing Tasks. Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling (SBIM '09), ACM, 133–140.

[16] Steven Schkolne, Michael Pruett, and Peter Schroeder. 2001. Surface drawing: Creating organic 3D shapes with the hand and tangible tools. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)
ACM, 261-268.

[17] Ryan Schmidt, Azam Khan, Karan Singh, and Gordon Kurtenbach. 2009. Analytic drawing of 3D scaffolds. ACM Transactions on Graphics, 28, 5.

[18] Tilt brush https://www.tiltbrush.com/

[19] Steve Tsang, Ravin Balakrishnan, Karan Singh, and Abhishek Ranjan. 2004. A Suggestive Interface for Image Guided 3D Sketching. In Proceedings of the Conference on Human Factors in Computing Systems (CHI'04), ACM, 591-598.

[20] Gerold Wesche and Hans-Peter Seidel. 2001. FreeDrawer: a free-form sketching system on the responsive workbench. ACM Press, 167. http://doi.org/10.1145/505008.505041