Gesture-Based Augmented Reality Annotation

Yun Suk Chang*, Benjamin Nuernberger*, Bo Luan*, Tobias Höllerer*, John O'Donovan*
University of California, Santa Barbara, USA

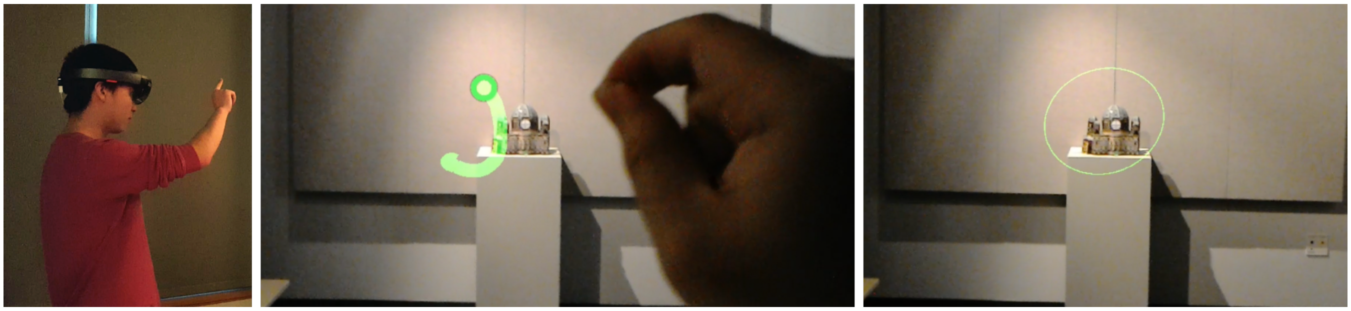

Figure 1: Overview of our annotation drawing procedure. Left: user wearing HoloLens, about to draw an annotation. Middle: "mixed-reality" capture view through the HoloLens while the user is drawing an annotation with the Air-Drawing method (user verging on the finger would actually see background as double image). Right: final beautified result of annotation.

Abstract

Drawing annotations with 3D hand gestures in augmented reality is useful for creating visual and spatial references in the real world, especially when these gestures can be issued from a distance. Different techniques exist for highlighting physical objects with hand-drawn annotations from a distance, assuming an approximate 3D scene model (*e.g.*, as provided by the Microsoft HoloLens). However, little is known about user preference and performance of such methods for annotating real-world 3D environments. To explore and evaluate different 3D hand-gesture-based annotation drawing methods, we have developed an annotation drawing application using the HoloLens augmented reality development platform. The application can be used for highlighting objects at a distance and multi-user collaboration by annotating in the real world.

Keywords: Augmented Reality, Annotations, Spatial Referencing, HoloLens

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 Introduction

Annotating physical objects is a central 3D user interaction task in Augmented Reality (AR) applications [2]. Annotation at a distance is of particular importance, as a main advantage of AR is that users can browse and interact with objects in their field of view without having to move up to those objects. For example, a user can easily draw an annotation on a distant object to highlight it for other observers. Current AR development platforms, such as the Microsoft HoloLens, facilitate drawing gesture interaction by providing stable tracking of head pose and approximate tracking of several finger poses in the user's view. When issuing drawing gestures referring to real-world objects, the question arises as to how the drawing should be done: the paint can be dropped in mid-air, at the user's fingertips, or it can be applied to surfaces in the real world provided that they have been modeled.

Several pros and cons are associated with each of these approaches. Drawing in mid-air at one's fingertip is a very general technique that does not require a world surface model at the time of drawing. Once a gesture has been completed, it can be projected onto a possible scene model or onto strategically chosen (virtual) world planes. A drawback of this technique is the need for vergence switches when aiming at world objects (in Figure 1, middle, the user would see two out-of-focus versions of the pedestal and cathedral model when verging on the fingertip; this is the major reason why the annotation appears misaligned in the image). The vergence switch is hypothesized to lower the accuracy of this technique [1]. Drawing on world-surfaces has its drawbacks, too. For example, if annotation occurs at significant distances, small inaccuracies in fingertip tracking can have a large effect on paint placement in the physical world. Moreover, both of the approaches have the drawback of drawing annotations that are restricted to physical surfaces, unlike freely drawing in mid-air.

In this demo, we present different annotation drawing techniques for issuing arrow, circle, and free-form annotations in mid-air or at real world surfaces at a distance. Arrow annotations are issued via a simple single line stroke, with the direction of the stroke indicating the placement of an arrow-head (where the stroke ends). Circle annotations are circular or elliptical strokes that end up in the neighborhood of the stroke starting point. Other forms of strokes are drawn as free-form annotations. We also present a mode that makes arrow annotations align with the detected surface's normal vector for clearly indicating the annotated object. The user will be able to freely interact with different drawing modes during this demo. This demo was recently presented at the I/ITSEC conference[1] and received positive feedback from participants.

2 System Design

We designed three different drawing methods on HoloLens: Surface-Drawing (Section 2.1), Air-Drawing (Section 2.2), and Mid-Air-Drawing (Section 2.3). Surface-Drawing begins by drawing directly on the detected surface, while Air-Drawing and Mid-Air-Drawing begin by drawing directly at the user's fingertip. Surface-Drawing and Air-Drawing project their annotations to the real-world surfaces, while Mid-Air-Drawing makes the drawn annotations stay in mid-air where they are drawn.

The drawings are performed via pinch-and-drag gestures and are completed by releasing the pinch gestures. To reduce noise in the

*e-mail: {ychang,bnuernberger,bo_luan,holl,jod}@cs.ucsb.edu

[1] http://www.iitsec.org/Pages/default.aspx

gesture input, we sample the user's drawing positions at 30 Hz and the finished annotation's path points at 1 point per 1 mm. The finished and sampled drawings are then recognized as arrow, circle, or free-form annotations by the $1 gesture recognizer [3]. To simplify the drawing process for users, we defined an arrow annotation gesture as a single-stroke straight line with the first point representing the arrow tail and last point representing the arrow head.

We also provided different annotation settings for defining the finished arrow annotation's action: No-Action and Surface-Normal (Section 2.4). No-Action does not alter the annotation, and Surface-Normal slowly transforms the arrow annotation, over a short period of time, to align with the detected surface's normal vector.

The drawing methods and annotation settings are controlled by single-clicks on the HoloLens clicker. Long presses on the HoloLens clicker clears the drawn annotations.

2.1 Surface-Drawing

Surface-Drawing creates annotations in 3D space by (1) drawing on detected real-world surface data and (2) projecting the finished drawing in an appropriate place depending on the annotation type.

For drawing on the surface, we define the drawing position as the intersection between the surface mesh and a ray cast from user's head through the fingertip position. Consequently, as the user is drawing annotations, the user can easily verge on the object of interest since the annotation is displayed at the detected surface. However, this method also has a drawback: small inaccuracy in finger tracking would result in large inaccuracy in annotation drawing when the distance from the detected surface is far away.

As a user completes the drawings, they are placed in the following manner, which was shown to be effective through pilot studies: For arrow annotations, their heads are anchored at the projected surface and their tails are projected so that the arrow annotation is perpendicular to user's viewing direction. For circle and free-form annotations, first the average depth of each point in the projected drawing path from user's view plane at the time of completing the gesture is calculated. Next the annotations are back-projected to the plane that is orthogonal to the viewing direction and displaced at the calculated average depth.

2.2 Air-Drawing

Air-Drawing (1) starts with drawing annotations at the fingertip and (2) forward-projects the drawings to their appropriate place when the drawing is finished. The projection is performed in the same way as for the Surface-Drawing method.

Since Air-Drawing is focused on the fingertip and not the object of interest, we hypothesized that it may be harder to aim because the user has to focus on either the drawing or the target object. However, the Air-Drawing method has the advantage of starting the drawing annotations from a 3D position that is not ray-casted; as a result, unlike Surface-Drawing, Air-Drawing does not require there to be known surfaces to begin the drawing. In addition, there are no jumps or discontinuities in depth when drawing on a closer object with distant background objects unlike for Surface-Drawing.

2.3 Mid-Air-Drawing

Unlike Air-Drawing and Surface-Drawing, Mid-Air-Drawing does not project drawings to detected surfaces; rather, it makes the drawings stay in mid-air where they are drawn.

Mid-Air-Drawing has an advantage in accuracy of drawing since it is unaffected by distance from the detected surface and problem of vergence. In addition, Mid-Air-Drawing can create more complex annotations since annotations stay in mid-air, unrestricted to the detected surfaces. Users may use this feature for creating complex 3D art. However, users must get close to the object if they wish to annotate the object using Mid-Air-Drawing method.

2.4 Surface-Normal

When the Surface-Normal setting is on, arrow annotations drawn by Surface-Drawing or Air-Drawing will transform itself to align with the normal vector of the detected surface as shown in Figure 2. The equation for finding the arrow direction **d** at time t as follows:

(a) Before the transform (b) During the transform (c) After the transform

Figure 2: Snapshots of Surface-Normal setting's arrow direction transformation.

$$\mathbf{d} = -\left(\frac{t}{t_{total}} \cdot \mathbf{n} + \left(1 - \frac{t}{t_{total}}\right) \cdot \frac{\mathbf{head} - \mathbf{tail}}{\|\mathbf{head} - \mathbf{tail}\|}\right),\ t \le t_{total} \quad (1)$$

Where **n** is the surface normal vector; t_{total} is the total time for the transform animation; **head** is a 3D arrow head point; and **tail** is a 3D arrow tail point. As t reaches t_{total}, the arrow direction becomes closer to the surface normal vector. The Surface-Normal setting has the advantage of creating arrow annotations that clearly point at the object from many different viewing angles. However, the arrow annotations will lose the original contextual information since their directions have been modified.

3 Demonstration Procedure

The demo will take place in a booth with some objects to be annotated. The user will wear the HoloLens and receive a basic tutorial on HoloLens gestures. The user will also learn how to draw arrow, circle, and free-form annotations. After the tutorial, the user will have freedom to draw annotations on any objects in any of following modes: (1) Mid-Air-Drawing, (2) Surface-Drawing, (3) Air-Drawing, (4) Surface-Drawing-Surface-Normal, and (5) Air-Drawing-Surface-Normal. The modes can be switched by single-clicking the HoloLens clicker, which will be given to the user. Annotations can be cleared by the demo presenter upon request by the user. The user may also try other HoloLens app for variety. The user's "mixed reality" capture view through HoloLens will be live-streamed to a monitor so that others can share the experience.

4 Conclusion

In this demo, we present five different drawing techniques for issuing annotations at the finger-tip or the real-world surface via our HoloLens application. Mid-Air-Drawing makes annotations in mid-air where it is drawn; Surface-Drawing directly draws onto real-world surfaces; Air-Drawing draws at the user's fingertip and projects the drawings onto the world upon release. The Surface-Normal setting makes arrow annotations of Surface-Drawing and Air-Drawing align with the real-world surface's normal vector. The demo application has been used by our lab for evaluating 3D hand-drawing interfaces for issuing or drawing annotations on distant objects [1]; this is useful for AR collaboration.

Acknowledgements

This work was supported by NSF grant IIS-1423676 and ONR grants N00014-14-1-0133 and N00014-16-1-3002.

References

[1] Y. S. Chang, B. Nuernberger, B. Luan, and T. Höllerer. Evaluating gesture-based augmented reality annotation (in submission). In *IEEE Symposium on 3D User Interfaces (3DUI)*, 3DUI '17, 2017.

[2] J. Wither, S. DiVerdi, and T. Höllerer. Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6):679–689, 2009.

[3] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 159–168, New York, NY, USA, 2007. ACM.