Demonstration of SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision

Yuhang Zhao
Cornell Tech, Cornell University
New York, New York, USA
Microsoft Research
Redmond, Washington
yz769@cornell.edu

Edward Cutrell
Christian Holz
Meredith Ringel Morris
Eyal Ofek
Andrew D. Wilson
Microsoft Research
Redmond, Washington
{cutrell,cholz,merrie,eyalofek,awilson}@
microsoft.com

ABSTRACT

Current virtual reality applications do not support people who have *low vision*. We present *SeeingVR* [7], a set of 14 tools that enhance a VR app for people with low vision by providing visual and audio augmentations. A user can select, adjust, and combine different tools based on their preferences. We demonstrate the design of SeeingVR in this paper.

KEYWORDS

Virtual reality, low vision, Unity, accessibility

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3313263

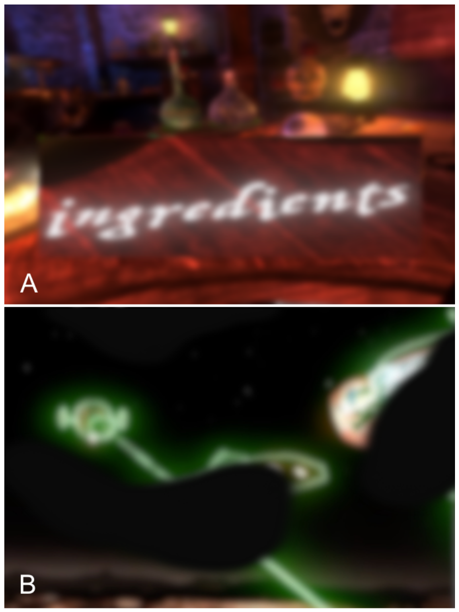

Figure 1: VR apps under simulated low vision augmented by SeeingVR: (A) *Waltz of the Wizard* [Aldin Dynamics, 2016] under diffuse depression of vision, with Bifocal Lens; (B) *Space Pirate Trainer* [I-Illusions, 2017] under blurred vision with blind spots, with Edge Enhancement and Depth Measurement.

INTRODUCTION

Mainstream VR mainly relies on visual abilities, making VR largely inaccessible to people with visual impairments [5]. Although researchers have created non-visual VR for people who are blind by leveraging auditory and haptic feedback (*e.g.*, [6]), no prior work on VR has focused on people with *low vision*, a visual impairment that cannot be corrected with eyeglasses or contact lenses. Since most low vision people have functional vision and use their vision extensively in daily activities [4, 8], they could potentially use mainstream, vision-dominant VR, if it were accessibly designed.

To increase VR accessibility for low vision, we designed *SeeingVR*, a set of low-vision tools that can be applied to a VR application as an overlay, enhancing its scene by providing visual and audio augmentations (Figure 1). We implemented two alternative ways to use SeeingVR: (1) a plugin with nine tools that can inject into any VR app in runtime *post hoc* to the implementation of the app; (2) a Unity toolkit that includes five additional tools requiring simple inputs from developers (14 total), which allows developers to integrate low vision support into their app during development. In this demonstration, we describe the design of SeeingVR and the two alternative ways to use it. Other information, including the formative study that guided our design, the implementation details, and the final evaluation, can be found in our full paper [7].

DESIGN OF SEEINGVR

SeeingVR is a set of 14 low vision tools that can augment a VR app as an overlay (Figure 2). Users can select, combine, and adjust different tools based on their preferences and needs.

Nine tools can augment VR apps without requiring the source code or any developer effort:

Magnification Lens. Magnification enables people with low vision to see details. We created a Magnification Lens, through which the user sees the VR scene with up to 10x magnification.

Bifocal Lens. Bifocal Lens is a smaller, rectangular-shaped magnifier (up to 10x magnification) at the bottom of the user's visual field. The user can look at the magnified region for details and can navigate the original virtual scene by looking through the remaining portion of their visual field.

Brightness Lens. The Brightness Lens allows the user to adjust the brightness of the scene. We use gamma correction to adjust the brightness with the gamma value ranging from 0.5 to 5.

Contrast Lens. The Contrast Lens increases the luminance contrast of the scene. The contrast intensity ranges from 0 (the original scene) to 1 with 10 increments.

Edge Enhancement. This tool adds edges to the whole virtual scene based on the depth and surface normal change to help segment objects from each other.

Peripheral Remapping. For people with peripheral vision loss, our Peripheral Remapping tool overlays the contours of a minified view of a wide field over the center of the user's vision, providing information that was out of the user's visual field [3].

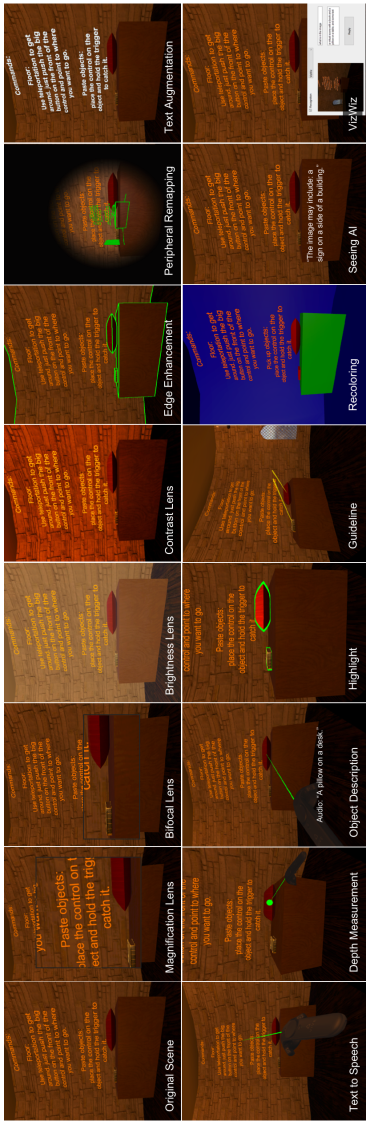

Figure 2: SeeingVR's 14 low vision tools.

Text Augmentation. This tool improves text contrast by automatically changing its color to black or white to ensures text contrast always satisfies the 4.5:1 ratio. We also change the font to Arial. The user can decide whether to make the text bold, and how much to increase the font size.

Text to Speech. This tool provides audio augmentation. We attach a virtual laser to the VR controller; when a user points the laser to a piece of text, this tool will read the text aloud.

Depth Measurement. We add a laser to the controller, with a ball appearing at the intersection point when the laser hits an object. The user can point the laser at objects and get a direct sense of distance by looking at the length of the laser between the ball and the controller.

Our remaining five tools leverage developer input to further augment a VR app:

Object Recognition. Similar to alt text for screen readers on a 2D display, if a developer adds descriptions to virtual objects in a VR scene, the Object Recognition tool can read aloud the corresponding description when the user points to a labeled object.

Highlight. If the developer labels semantically important objects, this tool adds contours around these objects to attract the user's attention in a visually overwhelming scene.

Guideline. When an important object is out of the user's visual field, our Guideline tool can direct their attention by connecting a line from the center of an important object (as designated by the developer) to the center of the user's field of view.

Recoloring. To help users segment objects in a low contrast scene, we designed a Recoloring tool, which recolors the whole scene, painting every two objects close to each other in the user's visual field with different colors. To support semantic recoloring, the developer needs to label the object hierarchy, providing information on which components belong to a single object.

Assistive Apps in VR. We demonstrate how assistive smartphone apps could be translated from the real world to the virtual world by reimplementing the popular assistive apps Seeing AI [2] and VizWiz [1] as SeeingVR tools. When the user triggers the "Seeing AI" tool, we mimic this app by capturing a screenshot of the current scene and sending it to the Microsoft Cognitive Services API for recognition; SeeingVR verbally announces the recognized result. When the user triggers the "VizWiz" tool, she speaks aloud a question about the current scene, and we send a screenshot of the scene with the recorded question to a human who composes an answer. When the human answers the question, SeeingVR reads the response aloud. Since these assistive apps need a specific gesture or command to trigger each recognition or human service request, we ask the developer to determine this trigger interaction to avoid any conflict with the original VR app's interaction scheme.

TWO ALTERNATIVES TO USING SEEINGVR

We offer two approaches to using SeeingVR: (1) a plugin with nine tools that augments VR applications *post hoc* without requiring the source code or the developers' input; (2) a Unity toolkit with five additional tools requiring developers' input (14 tools in total), with which developers can provide required metadata and add all low vision support to a VR app during development. To enable developers to provide the required metadata, our Unity toolkit extends the GameObject class in Unity with three accessibility features: (1) *Description*, which describes a virtual object for the Object Description tool; (2) *isSalient*, which labels whether a virtual object is important for the Highlight and Guideline tools; and (3) *isWholeObject*, which provides semantic information on the scene's object hierarchy, indicating whether the current GameObject and its children comprise a single object; the Recoloring tool leverages this for semantic recoloring. We added these features to the Unity editor, so developers can edit the information both in the editor and in code during development.

CONCLUSION

We presented SeeingVR, a set of 14 low vision tools that can be applied to a VR app, enhancing its scenes with visual and audio augmentations. We implemented two approaches to applying SeeingVR: a plugin with nine tools that augments an existing VR app *post hoc*, and a Unity toolkit that allows developers to provide the metadata required by specific tools and integrate all 14 tools during development. While SeeingVR focused on low vision, we hope our work can inspire the design of general accessibility standards for VR.

REFERENCES

[1] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 333–342.

[2] Microsoft. 2018. Seeing AI | Talking camera app for those with a visual impairment. https://www.microsoft.com/en-us/seeing-ai last accessed 21 December 2018.

[3] Eli Peli. 2001. Vision multiplexing: an engineering approach to vision rehabilitation device development. *Optometry and Vision Science* 78, 5 (2001), 304–315.

[4] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. 2016. How people with low vision access computing devices: Understanding challenges and opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 171–180.

[5] Paul N Wilson, Nigel Foreman, and Danaë Stanton. 1997. Virtual reality, disability and rehabilitation. *Disability and rehabilitation* 19, 6 (1997), 213–220.

[6] Yuhang Zhao, Cynthia L Bennett, Hrvoje Benko, Edward Cutrell, Christian Holz, Meredith Ringel Morris, and Mike Sinclair. 2018. Enabling People with Visual Impairments to Navigate Virtual Reality with a Haptic and Auditory Cane Simulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 116.

[7] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and D. Andrew Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM.

[8] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. 2015. Foresee: A customizable head-mounted vision enhancement system for people with low vision. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. ACM, 239–249.