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
yz769@cornell.edu

Edward Cutrell
Microsoft Research
Redmond, Washington
cutrell@microsoft.com

Christian Holz
Microsoft Research
Redmond, Washington
cholz@microsoft.com

Meredith Ringel Morris
Microsoft Research
Redmond, Washington
merrie@microsoft.com

Eyal Ofek
Microsoft Research
Redmond, Washington
eyalofek@microsoft.com

Andrew D. Wilson
Microsoft Research
Redmond, Washington
awilson@microsoft.com

ABSTRACT

Current virtual reality applications do not support people who have *low vision*, *i.e.*, vision loss that falls short of complete blindness but is not correctable by glasses. We present *SeeingVR*, a set of 14 tools that enhance a VR application for people with low vision by providing visual and audio augmentations. A user can select, adjust, and combine different tools based on their preferences. Nine of our tools modify an existing VR application *post hoc* via a plugin without developer effort. The rest require simple inputs from developers using a Unity toolkit we created that allows integrating all 14 of our low vision support tools during development. Our evaluation with 11 participants with low vision showed that SeeingVR enabled users to better enjoy VR and complete tasks more quickly and accurately. Developers also found our Unity toolkit easy and convenient to use.

CCS CONCEPTS

• **Human-centered computing → Accessibility technologies**.

KEYWORDS

Virtual reality, low vision, Unity, accessibility

ACM Reference Format:
Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3290605.3300341

1 INTRODUCTION

Virtual reality (VR) is an emerging technology widely applied to different fields, such as entertainment [5, 88], education [41, 72], and accessibility [24, 78]. Mainstream VR mainly relies on visual abilities, making VR largely inaccessible to people with visual impairments [36, 78]. Although researchers have created non-visual VR for people who are blind by leveraging auditory (*e.g.*, [36, 47]) and haptic feedback (*e.g.*, [9, 23, 27, 28, 32, 83]), no prior work on VR has focused on people with *low vision*.

Low vision is a visual impairment that cannot be corrected with eyeglasses or contact lenses [1]. According to the World Health Organization, 217 million people worldwide

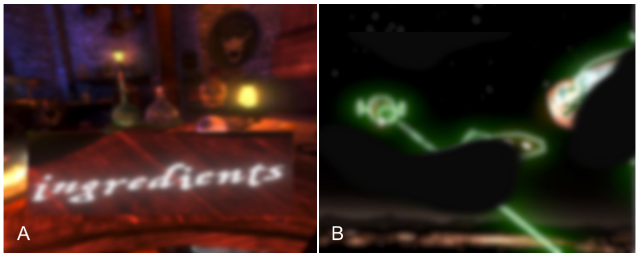

Figure 1: VR apps under simulated low vision augmented by SeeingVR: (A) *Waltz of the Wizard* [Aldin Dynamics, 2016] under diffuse depression of vision, with Bifocal Lens; (B) *Space Pirate Trainer* [I-Illusions, 2017] under blurred vision with blind spots, with Edge Enhancement and Depth Measurement.

[*] Work completed as an intern at Microsoft Research

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2019, May 4–9, 2019, Glasgow, Scotland Uk
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5970-2/19/05...$15.00
https://doi.org/10.1145/3290605.3300341

have low vision [77]. Low vision includes a variety of visual conditions, such as blurred vision, loss of peripheral vision, and extreme light sensitivity [3]. Prior studies have shown that people with low vision use their remaining vision extensively in daily activities, often with the support of low-tech and/or high-tech aids [58, 59, 85].

Using their residual vision, people with low vision could potentially use mainstream, vision-dominant VR, if it were accessibly designed. We conducted a formative study to understand the challenges that people with low vision face when using VR. Guided by this study, 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. We evaluated SeeingVR with 11 people with low vision, and found it enabled them to complete VR tasks more quickly and accurately. We also demonstrate that our plugin can augment popular VR apps *post hoc*, and that developers found our toolkit easy to use.

2 RELATED WORK

VR for People with Visual Impairments

Prior work on accessibility of VR has focused on blindness, creating non-visual VR with audio and haptic feedback. Several researchers have designed auditory VR systems [15, 36, 47, 48, 50, 51, 54, 62, 63]. For example, Picinali *et al.* [47] created an acoustic VR system, generating a 3D soundscape mixing environmental and self-generated sounds (*e.g.*, footsteps) based on users' position in the virtual space. Researchers have also explored haptic interfaces for VR for blind users [23, 29, 32, 52, 55, 66, 67, 83]. For instance, Canetroller [83] is a wearable VR controller that generates force, tactile-vibration, and auditory feedback to simulate untethered white cane interaction in VR. While prior work has focused on non-visual VR for blind users, SeeingVR focuses on people with low vision who are likely to prefer using their vision in daily life [58, 59].

Little work has explored VR accessibility for people with low vision. The Disability Visibility Project and ILMxLAB [80] conducted a survey investigating the VR experiences of 79 people with disabilities; the survey indicated some challenges faced by people with low vision, such that lighting effects (*e.g.*, flashing) hurt some users' eyes, and some users lose depth perception completely in VR. Trewin *et al.* [64, 65] designed several tools to increase the accessibility of a specific 3D game on a 2D display for people with a range of disabilities; in that system, a user with low vision could enlarge the scene or customize the font size and color of text. The GearVR [14] also supports some accessibility features, such as basic zooming and inverted color options for low vision users. Teofilo *et al.* [61] evaluated these features, confirming that users with disabilities showed enthusiasm for them. These prior projects adapted basic low vision features on 2D displays to the virtual space; SeeingVR builds on prior work by considering the 3D space presented by VR, a variety of VR tasks, and a range of low vision conditions.

Approaches to Low Vision Aids

Most low vision aids are designed to help people access detailed information through magnification and contrast enhancement. There are optical tools, such as handheld magnifiers and bioptics for magnification [21, 38, 53]. For digital devices, people use Closed Circuit Televisions (CCTVs), which capture video of the real world from a camera and magnify it onto a digital display [79]. Mainstream computer and smartphone platforms also support built-in applications [13, 16, 19] or standalone software (*e.g.*, MAGic [35]), which provide magnifiers and adjustable contrast and color settings. There have been smartglass products, such as eSight [10], that used similar enhancements to improve users' visual ability. Researchers have also created new assistive technologies on AR glasses using these enhancement methods [22, 39, 85]. For example, ForeSee [85] was a pair of video see-through AR glasses that provided five enhancements and two display modes.

Another approach is increasing visual field for people with limited field of view. One optical device is prism glasses [31], which expand people's vision by optically shifting objects from outside the user's visual field to a functional area. Peli and his team also designed a method on AR glasses called vision multiplexing, where the contours of a minified version of the wide field are overlaid on the user's functional field of view [34, 46, 70].

Other vision enhancement methods are task-specific; for example, visual cues to direct users' attention in a visual search task [26, 86], recoloring scenes to support obstacle detection [12], and indicating object distance with brightness [18]. These approaches informed the design of SeeingVR's tools.

Accessibility Enhancement of Mainstream Technology

We aim to improve the accessibility of mainstream VR for people with low vision instead of creating specialized VR experiences. Mainstream technology has many advantages over specialized technology, such as lower cost, increased availability, and social acceptability. Prior studies found that specialized technology can stigmatize people with disabilities, leading to technology abandonment [45, 49, 56].

One way to improve accessibility is integrating accessibility support at the development level. Basic accessibility features like screen readers and magnifiers have been embedded into computer and smartphone platforms (*e.g.*, [4, 43]). To support specific accessibility features, developers need to follow rules and provide metadata. For example, to support screen readers, developers need to label descriptions for UI components and organize related UI content in one group [2, 75]. Although it would be ideal if all applications were designed to be accessible during development, developers sometimes fail to provide required metadata [7] or follow accessibility guidelines [20].

To repair failures in fundamental accessibility support, researchers have designed *post hoc* modification methods by adding overlays or proxy layers to existing technologies. For instance, many web proxies have been designed to improve web accessibility [7, 8, 60]. Zhang *et al.* introduced interaction proxies [81], which improved the accessibility of mobile apps by adding an overlay that modified the input and output of the underlying app. Researchers have also added physical overlays to digital devices to increase their accessibility by providing tactile feedback [17, 25, 82].

VR is a relatively new medium, with no accessibility framework or guidelines yet established. We present two alternatives to applying SeeingVR via both development and *post hoc* modification approaches.

3 FORMATIVE STUDY

We conducted a formative study to understand the challenges people with low vision experience with VR. We recruited six participants with low vision, with a variety of visual conditions (Table 1, L1-L6). During the one-hour study, we observed participants experiencing four VR apps with an HTC Vive headset and two controllers. After a tutorial, participants used the VR apps while thinking aloud, talking about what they saw, what challenges they had, and possible improvements. The four VR apps included: (1) *EscapeVR-HarryPotter* (*EscapeVR*) [33]: an open-source room-escape game where users find clues under objects in a dark room. (2) *Drop* [Drop Software Inc., 2017]: a productivity app, where a user types keywords on a virtual keyboard and views search results on several screens. (3) *Tilt Brush* [Google, 2016]: an app where a user paints in 3D space using the controller as a brush. (4) *Space Pirate Trainer* [I-Illusions, 2017]: a game where the user shoots flying droids with the controllers. L1 did not use the apps due to light sensitivity, instead describing past VR experiences. Participants encountered several challenges using the VR apps:

Seeing Things at a Distance. Participants liked that they had the power to move close to things in VR. With teleportation (*i.e.*, a locomotion technique that allows players to rapidly transition to a target location), they could get very close to distant text or objects in order to see details. However, this interaction is not always supported (only *EscapeVR* in our study supported teleportation). Even with teleportation, participants were limited to the ground level; it was difficult to see text at higher locations. Although participants could still physically walk closer to the text, they were limited by efficiency of traversing large virtual spaces and could not always walk close enough to read. L2 noted, "There's a time limit... so if I'm really playing, I have to be balancing, 'can I read from here or do I want to teleport and read it and go away.'"

Table 1: Demographic information of participants with low vision. Participants labeled "1" participated in the formative study, while those labeled "2" participated in the final evaluation. All the information in the table was self-reported by the participants.

ID	Age/sex	Legally blind	Diagnosis	Visual acuity	Visual field	Light sensitivity	Color vision
$^{1}L1$	34/M	Yes	Septo-optic dysplasia	Left: no vision; right: 20/800	No peripheral vision	Yes	Needs high contrast colors
$^{1,2}L2$	32/M	No	Ocular Albinism	Left: 20/80; right: 20/200	Full	Yes	Good
$^{1,2}L3$	51/M	Yes	Retinal Pigmentosa	Left: 20/200; right: no central vision	Limited in both eyes with blind spots in peripheral vision	No	Needs high contrast colors
$^{1,2}L4$	46/M	Yes	Cataracts; partial glaucoma	Left: partial vision; right: very low	Very narrow in left eye; no peripheral vision in right eye	No	Good
$^{1,2}L5$	23/F	No	Degenerative vision; ocular migraines	Better than 20/200	Narrow in the left eye; have blind spots in both eyes	Yes	Good
$^{1,2}L6$	22/M	Yes	Bilateral microphtalmia	Left: 20/200; right: no vision	Only has peripheral vision in the lower part of left eye	No	Good
$^{2}L7$	33/M	Yes	Cortical blindness	20/400	Full	No	Good
$^{2}L8$	57/F	Yes	Underdeveloped optic nerves; Nystagmus; Colorblindness	20/400	Full	Yes	Colorblind
$^{2}L9$	31/M	Yes	Bilateral retinoblastoma	Left: 20/80; right: no vision	Very limited	Yes	Needs high contrast colors
$^{2}L10$	22/F	Yes	Optic nerve detached on left eye; primarily nearsighted	Left: no vision; right: 20/400	Full in right eye; no vision in left eye	Yes	Good
$^{2}L11$	51/F	No	Pigmented growth on optic nerve	Left: better than 20/200; right: no vision	Limited	Yes	Good
$^{2}L12$	32/M	Yes	Blue cone achromatopsia	Left: 20/400; right: 20/200	Lost some peripheral vision but not all	Yes	Colorblind

Interacting with Virtual Elements. It was challenging for participants to interact with different virtual elements. When using a laser pointer to select a specific menu item, participants with low visual acuity (L3, L6) could not distinguish where the laser pointer was aimed since it got very thin at a distance and sometimes had low contrast with the background. For participants with monocular vision (*e.g.*, L4, L6), it was hard to pick up virtual objects due to challenges judging distance when moving the controller toward an object. Tracking moving targets was especially difficult for participants with limited visual field (*e.g.*, L4).

Dealing with Lighting Effects. Lighting effects caused difficulty for many participants. Some VR scenes were too dark (*EscapeVR*), which made everything too low contrast to see. Some scenes were too bright (*Drop*), which bothered those with light sensitivity (*e.g.*, L2, L5). Moreover, some VR scenes created lighting effects to make the experience more realistic or exciting (*e.g.*, a candle effect in *EscapeVR*, a flash effect in *Tilt Brush*), yet these effects overwhelmed the participants, thus diminishing low vision accessibility.

All participants indicated a need for technologies to make VR more accessible. L6 further emphasized the importance of universality for accessibility feature design for VR: "Whatever [VR] app you're running, you should be able to do a certain gesture... to [invoke an accessibility feature]."

Discussion

Our formative study guided the subsequent design of SeeingVR. First, the challenges participants faced in VR helped us identify the tasks to facilitate when improving VR accessibility, such as seeing details at a distance, perceiving depth information, and searching for and tracking a target. Second, participants' accessibility needs varied based on their visual conditions, lighting effects in the VR scenes, and the different VR tasks, which suggested a need for customizable solutions based on users' abilities and context. Moreover, participants indicated it would be more beneficial if a consistent set of assistive tools were available across all VR apps; since VR apps can have different interaction designs, it is also important to avoid conflicts between the use of the assistive technology and the original app's interaction scheme, so we left the choice of how to invoke the SeeingVR toolkit up to developers, allowing a voice-based Wizard-of-Oz invocation for user testing.

4 SEEINGVR

Guided by the formative study, we designed SeeingVR, a set of 14 low vision tools that can augment a VR application as an overlay with both visual and audio feedback (Figure 2). Users can select, combine, and adjust different tools based on their preferences and needs. Many SeeingVR tools were inspired by low vision technologies in the real world. Nine tools can augment VR apps without requiring the source code or any developer effort, while the other five can further augment the app if the developer provides metadata such as object descriptions. Color choices for all tools are customizable, to account for variation in VR scene color schemes as well as users' color perception capabilities. We use speech input to control SeeingVR (selecting and adjusting each tool) to avoid conflicting with the apps' native interaction design.

Our nine tools that augment VR apps *post hoc* without requiring any developer effort are:

Magnification Lens. Magnification is the most common vision enhancement to enable 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. We attach the Magnification Lens in front of the user's eyes, covering 60 degrees of users' visual field; we did not cover the full peripheral visual field so that users can keep some spatial awareness. This tool is implemented by adding a 2D plane with the texture rendered from a second virtual camera that captures the VR scene from the same position as the main camera in Unity. We adjust the field of view of the second camera to adjust the magnification level.

Bifocal Lens. To reduce the dramatic change magnification may create, we designed a Bifocal Lens by adding a smaller, rectangular-shaped magnifier at the bottom of the user's visual field. This simulates bifocal glasses, in which each lens provides two different refractive powers; the upper part of the lens is used for ordinary or distant vision, while the lower part is for near vision. With our Bifocal Lens, 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. The position of the Bifocal Lens can be adjusted higher or lower according to the position of the user's functional vision. The magnification level ranges from 1 to 10. This tool is implemented in the same way as the Magnification Lens.

Brightness Lens. Since people with low vision have different light sensitivity and some VR scenes can have extreme dark or bright light effects, we provide a Brightness Lens, allowing 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. We provide a Contrast Lens to increase the luminance contrast of the scene by enhancing the brightness difference between each pixel and the average of its adjacent pixels. The contrast intensity ranges from 0 (the original scene) to 1 with 10 increments.

Edge Enhancement. Another way to increase contrast is to use Edge Enhancement to segment objects from each other [22]. We add edges to the whole virtual scene based on the depth and surface normal change. It is implemented

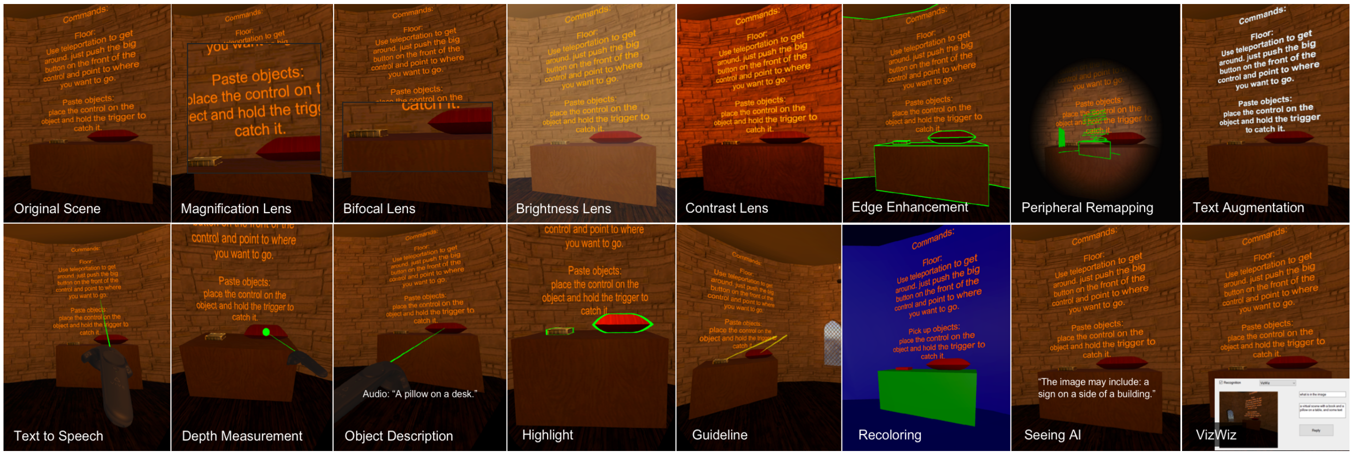

Figure 2: SeeingVR's 14 low vision tools. Tunnel vision in 'Peripheral Remapping' is simulated. The quote in 'Object Description' shows the read-aloud audio description. The quote in 'Seeing AI' shows the read-aloud audio description from the recognition API. The inset in 'VizWiz' shows a screenshot of the question received and the response provided by the human worker.

via the Edge Detection post-processing effect [68] in Unity's Standard Assets Effects package.

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 [46]. Users can adjust the color, size, and position of this overlay.

Text Augmentation. Our Text Augmentation tool improves text contrast by automatically changing its color to black or white (whichever has higher contrast with the scene's background). This ensures text contrast always satisfies the 4.5:1 ratio suggested by web content accessibility guidelines [73]. We also change the font to Arial, which has been found advantageous to people with low vision [37, 84]. The user can also decide whether to make the text bold, and how much to increase the font size.

Text to Speech. We offer a Text to Speech tool for audio augmentation of VR. We attach a virtual laser to the VR controller; when a user points the laser to a piece of text in the virtual scene, this tool will read the text aloud (the audio feedback is an add-on to the VR app's original audio sources). Different from the classic thin laser in most VR apps, our laser was designed to be more visible for people with low vision: it is relatively thick with a 2mm x 2mm cross section and users can also adjust its color for high contrast.

Depth Measurement. Our Depth Measurement tool addresses depth perception challenges by adding a laser (same laser design as the Text to Speech tool) 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.

As the creator of a VR app, developers have knowledge that can be used to further augment the app for low vision users. Our remaining five tools leverage developer input:

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. When exploring a VR scene, a user with low vision may not know where to focus if the scene is visually overwhelming. If the developer labels semantically important objects, our Highlight tool adds contours around these objects to attract the user's attention. The contour is generated by a shader applied to the material of these objects in Unity.

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. If the object is behind the user, the line will curve around the user to their central vision.

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. This tool also simplifies the scene by replacing complex textures. 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 also consider how assistive smartphone apps could be translated from the real world to the virtual world. We demonstrate this concept by reimplementing the popular assistive apps Seeing AI [44] and VizWiz [6] as SeeingVR tools. When the user triggers the "Seeing AI" tool, we mimic the functionality of this computer vision app by capturing a screenshot of the current virtual scene and sending it to the Microsoft Cognitive Services API [42] for recognition; SeeingVR verbally announces the recognized result. When the user triggers the "VizWiz" tool, she speaks aloud a question about the current virtual 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. Unlike our other tools, the 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.

All tools in SeeingVR can be combined based on the users' preferences. The combination order will not affect the combination effect in our current prototype.

Implementation

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. Both approaches currently focus on the Unity platform since it is one of the most widely-used platforms for VR development [71].

Plugin for Post Hoc Modification During Runtime. Our SeeingVR plugin augments Unity-based VR apps during runtime. After the user specifies the path of the VR app to be augmented and runs our plugin, SeeingVR adds itself to the VR app. After starting the app, the user can now select and adjust the various tools of SeeingVR in real time. We used the Illusion Plugin Architecture (IPA) [11], which allows developers to inject code into an existing Unity app in the form of a dynamic-link library (DLL). We first implemented all our low vision tools in Unity and compiled the relevant components (*e.g.*, prefab, shader) into an AssetBundle (*i.e.*, an archive file containing Unity data that can be loaded at runtime). We then built an IPA-compatible DLL that loads all the tools from the AssetBundle into the current VR app. Our plugin then automatically runs the IPA, injecting the DLL into a specific VR app.

Unity Toolkit for Developer. We implemented a Unity toolkit for developers since five of our low vision tools (*e.g.*, Object Description tool) require developers to input semantic information that we cannot easily get by automatic recognition. To enable developers to provide the required metadata, our Unity toolkit extends the GameObject class [69] 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.

The Unity toolkit provides developers access to all of our 14 low vision tools. We created a Unity prefab (*i.e.*, a GameObject template with specific features), "Accessibility Manager," which automatically adds all the tools into the VR app. It provides public entries to control all the SeeingVR tools and their parameters. Developers can simply add this prefab into their VR scene and control the tools through our Accessibility Manager. By following our developer API, they can also customize the tools to harmonize with the design of their VR application.

For the Unity developer toolkit, we used C# extension methods to add our three accessibility features to the GameObject class in Unity. We also customized the Unity editor so that developers can directly edit our metadata. We compiled the GameObject extension file, the editor customization file, and the implementation of all SeeingVR tools into a Unity Asset package. Developers can import this package into their project to use our toolkit.

Control of SeeingVR. Users can control SeeingVR with speech. We implemented this interaction with the Wizard of Oz [40] method in the user study. When a user spoke aloud how she wanted to adjust the tools, a researcher acted as the computer and controlled SeeingVR through a Wizard of Oz interface that communicated with SeeingVR in real time over TCP.

5 EVALUATION

We conducted studies to answer three research questions: (1) How does SeeingVR affect the VR experience for people with low vision? (2) Can SeeingVR be broadly applied to different VR applications? (3) What is the usability of the Unity toolkit for developers?

Study1: Evaluating SeeingVR with Low Vision Users

In this study, we evaluated the effectiveness of 13 tools in SeeingVR with participants with low vision. We did not evaluate the "Assistive Apps" tool, because our initial tests indicated that the Seeing AI clone in VR was not robust, likely due to a lack of VR training data in addition to the real-world images used in building current recognition APIs, and we had no infrastructure to support reliable real-time human worker responses to our VizWiz replication.

Method. We recruited 11 participants with low vision, five of whom had participated in the formative study. Participants had a variety of visual conditions (Table 1, L2-L12). The study lasted ninety minutes. We first interviewed participants about their prior VR experience. We continued with a tutorial, a virtual task session, and an app experience session.

In the tutorial, we introduced the HTC Vive headset and controllers. We then demonstrated the 13 low vision tools (excluding the Assistive Apps tools) one by one in a basic virtual scene with four walls, a table, a stapler and a pen on the table, and a piece of low contrast text on the wall. We presented the tools from more familiar (*e.g.*, Magnification) to more novel (*e.g.*, Recoloring) to ease the learning process. During the tutorial, we asked the participants to think aloud and share their initial reactions to each tool.

In the virtual task session, participants executed three virtual tasks with and without using SeeingVR. The tasks were: navigating a virtual menu in 3D space (Menu Navigation, Figure 3A), looking for a specific virtual object on a desk (Visual Search, Figure 3B), and shooting a moving target (Target Shooting, Figure 3C). We chose these tasks because they are common in VR apps but challenging for people with low vision according to our formative study. The task procedures were as follows:

Menu Navigation: Our menu navigation scene had a semi-transparent blue menu with seven menu items against a dark blue background. The menu was 10 meters away from the user and measured 4.4m x 5m. The user could point and select a menu item with the laser attached to the controller. The menu was hidden at the beginning of each trial. When a trial started, the system verbally announced a specific menu item (randomly chosen). The menu then appeared, and the participant needed to select the announced item. When the item was selected, the menu disappeared. For each trial, the system randomly picked seven of fourteen menu items to display in a random order. We recorded whether participants selected the correct item and the time required.

Visual Search: Our visual search scene was a dark room with a desk and a box. There were 10 virtual objects on the table (*e.g.*, a pen, a stapler, *etc.*). The trial started with an empty desk. When the system verbally announced a specific object (randomly selected), all 10 objects appeared at random, non-overlapping locations on the desk. The participant needed to pick up the object and throw it in the box, using the controller. When an object was thrown in the box, all the objects disappeared again. We recorded selection accuracy and timing.

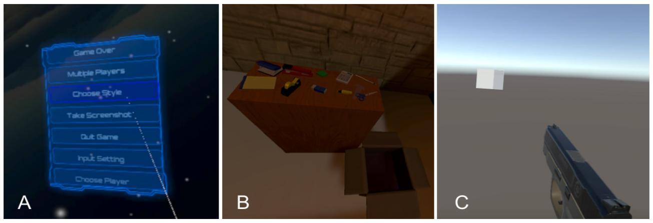

Figure 3: Menu Navigation, Visual Search, Target Shooting.

Target Shooting: Our target shooting scene had 12 white cube-shaped targets evenly distributed at eye-level around a circle with a 12-meter radius. The user stood in the center of the circle and could use the controller to shoot the targets. All 12 targets were hidden at the beginning of a trial; a random target would appear and gradually moved towards the participant. When shot, the target disappeared and another randomly appeared. Participants needed to shoot five targets in each trial. We recorded time and accuracy.

For each task, we first introduced the task to the participants and asked them to try one trial. When they understood the task, we asked them to select and adjust their preferred low vision tools in SeeingVR to assist with that task. When the participants confirmed their selections, they conducted five trials of the task in each of two conditions: using SeeingVR with their preferred tools and without SeeingVR. We asked participants to conduct the tasks as quickly and accurately as possible. We counterbalanced the order of the three tasks and of the two conditions within each task to mitigate order effects.

Finally, in the app experience session, participants freely explored two VR apps with SeeingVR: *EscapeVR* (Figure 4A) and *Drop* (Figure 4B). During the exploration, they thought aloud, talking about which SeeingVR tools they were using and why. While *EscapeVR* is an open source app where we could add all the low vision tools with our Unity toolkit, *Drop* is not open source, so that we could only add the nine tools via the plugin. We counterbalanced the order of the two apps.

We finished the study with an interview, asking participants their general experience with SeeingVR and how they wanted to improve it. Participants also provided scores

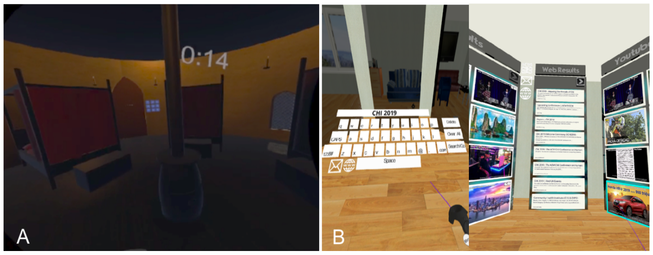

Figure 4: Two study apps: (A) *EscapeVR* and (B) *Drop*.

for the usefulness of each low vision tool from 1 to 7 (with 1 meaning not useful at all, and 7 meaning extremely useful).

Results. ***Effectiveness of SeeingVR.*** All participants completed the three tasks in the study more quickly and accurately when using SeeingVR.

For the Menu Navigation task, all participants completed the task with SeeingVR (*mean*=4.62s, *SD*=4.67s), but only four (L2, L4, L10, L12) were able to complete the task without any augmentations (with augmentation: *mean*=2.22s, *SD*=0.35s; without: *mean*=4.67s, *SD*=1.62s). For these four participants, a paired t-test showed that SeeingVR significantly reduced the time they used in finding and selecting a specific menu item (t_3=-3.53, *p*=0.039, *d*=1.76 by Cohen's d measure). While L10 and L12 sometimes selected the wrong items when not using SeeingVR (accuracy of L10: 40%, L12: 60%), all participants completed the task 100% accurately with SeeingVR.

For the Visual Search task, all participants were able to complete the task both with (*mean*=3.98s, *SD*=2.33s) and without (*mean*=7.51s, *SD*=4.94s) SeeingVR. A paired t-test showed a significant decrease in time when participants used SeeingVR (t_{10}=-3.67, *p*=0.004, *d*=1.11). While L3 and L8 occasionally picked up the wrong object (accuracy of L3: 40%, L8: 60%) when not using augmentations, all participants' accuracy reached 100% with SeeingVR.

For the Target Shooting task, all participants completed the task with (*mean*=16.04s, *SD*=9.60s) and without (*mean*=27.08s, *SD*=13.56s) SeeingVR. They successfully shot all targets in each trial in both conditions. A paired t-test showed that SeeingVR significantly reduced the time participants needed for this task (t_{10}=-5.15, *p*=0.0004, *d*=1.55).

All participants preferred using SeeingVR rather than not using augmentations. Besides the improvement in efficiency and accuracy, participants indicated that SeeingVR benefitted their confidence. For example, in the Visual Search task, some participants (*e.g.*, L3, L4) originally had to get close to the desk or pick up an object to closely examine it. The SeeingVR tools helped them pick up the right object more confidently. "When augmentation was turned off, it was really hard. I can do it but it required more effort. It's creating some stress I just wouldn't have when using the [SeeingVR] tools" (L3).

Preferences among Low Vision Tools. Participants' preferences for the low vision tools in SeeingVR varied based on the tasks and their visual abilities.

Figure 5 shows participants' preferred tool combinations, which varied by task. Besides the Brightness Lens that was widely selected in all tasks, we found that participants generally used Text Augmentation and the Magnification Lens to assist text reading in the menu navigation task, used Highlight to quickly focus on the target object in the visual search task, and used Guideline to track the moving target in the shooting task. Most participants also selected Depth Measurement in the menu navigation and shooting tasks. They used this tool as a laser enhancement tool: by observing the position of the ball at the end of the laser, they could distinguish which menu item the controller was pointing at and whether the gun was aiming at the target.

We also found some tool selection patterns when observing how participants used SeeingVR in the "app experience" portion of the study. Participants usually used Bifocal Lens to navigate a scene to get a general overview (and in the dark, low-contrast *EscapeVR* app most also used Edge Enhancement), but they switched to Magnification Lens when they wanted to focus on details, such as reading text. Participants only preferred Text to Speech when the text was lengthy. For example, in *Drop* 10 participants used only visual augmentations (*e.g.*, Text Augmentation) when typing on the virtual keyboard (Figure 4B, left), while nine participants switched to Text to Speech when reading the search results (Figure 4B, right). Moreover, all participants used Highlight in *EscapeVR* since it helped distinguish which objects they should interact with in the dark and low contrast scene. Although Guideline could also help track objects, participants felt it was distracting in *EscapeVR* because there were many important objects in the scene, which resulted in too many lines.

Participants' tool selections were also related to their visual abilities. In the menu navigation task, participants with better visual acuity (L2, L4) used Text Augmentation without magnification tools. L9, who had relatively good visual acuity, used Magnification Lens but with a low magnification level. In the visual search task, although all the objects were put on the desk within a small area, some participants with limited visual field (L6, L9, L11) still used Guideline to locate the target object. Participants with limited but not extremely low visual field also preferred using Peripheral Remapping: L6 and L10 used it in the shooting task and L12 used it when exploring *EscapeVR*. However, participants whose visual field was extremely limited (*e.g.*, L4, L9) did not use this tool since it covered their full functional vision and prevented them from seeing the original scene. Moreover, some participants with good color vision (L6, L7) used Recoloring in the shooting task.

Although there were some preference patterns related to visual abilities, participants with similar visual abilities sometimes had distinct preferences. For example, L2 and L5 both had relatively good vision (not legally blind). L5 liked Depth Measurement and used it in two tasks and the EscapeVR app, but L2 did not use it at all. According to Figure 5, no two participants had exactly the same selections for the three tasks. This aligned with the results in Zhao *et al.* [85], who found that people with low vision have different preferences and customization is important.

	Magnification Lens	Bifocal Lens	Brightness Lens	Contrast Lens	Edge Enhancement	Peripheral Remapping	Text Augmentation	Text to Speech	Depth Measurement	Object Description	Highlight	Guideline	Recoloring
L2			○ ■				✓				○	■	
L3	✓				○ ■		✓		○ ■	○	○	■	
L4			■				✓		✓ ■		○	■	
L5	✓ ■		✓ ○ ■	✓			✓		✓ ■		○	■	
L6	✓		✓ ○ ■	✓ ○		■	✓		✓ ■		○	○	■
L7			✓ ○ ■					✓	✓ ■		○ ■		■
L8	✓		✓ ○ ■				✓		✓ ■		○ ■	■	
L9	✓		✓ ○ ■	○ ■			✓		✓ ■		○ ■	○ ■	
L10	✓		✓ ○	✓		■	✓		■		○		
L11	✓		■				✓		✓ ■		○	○ ■	
L12		✓	✓ ○ ■	✓	○		✓		✓ ■		○ ■	■	
Mean	5.5	4.8	5.3	4.5	4.7	3.4	6.9	5.1	6.1	4.7	6.1	4.9	3.1

Figure 5: Participants' tool choices: "✓" indicates tools used in the Menu Navigation task; "○" indicates tools used in the Visual Search task; and "■" indicates tools used in the Target Shooting task. The last row shows the average usefulness rating (on a 7-point scale) for each tool.

The last row in Figure 5 shows the average utility scores for each tool on a seven-point scale. Text Augmentation, Depth Measurement, and Highlight were the three highest-rated tools (rated above 6). Most tools had an average score between 4.5 and 6 since they were useful for specific scenarios (*e.g.*, Edge Enhancement for low contrast scenes, Guideline for tracking moving objects). However, Peripheral Remapping and Recoloring had low scores below 4. This is because Peripheral Remapping was only effective for participants with limited but not extremely low visual field (*e.g.*, L6, L10), and Recoloring made too much aesthetic change, removing the scene's original textures.

Participants showed great interest in Depth Measurement. Besides using it to perceive depth (*e.g.*, L10 used it as a 'visual' cane to detect the boundary of the virtual space), participants used it to enhance the laser pointer and aim a target. Moreover, L5 used this tool to estimate the place she would teleport to in *EscapeVR*. Participants suggested improving Depth Measurement with gradient color along the laser to indicate different distances.

Impact and Potential of SeeingVR. Participants felt that SeeingVR made VR "more friendly" (L10) and "enjoyable beyond just usable" (L5). They also appreciated the variety of tools and the flexibility to customize tools' parameters. However, some participants (L2, L4, L7) had concerns that some tools (*e.g.*, Recoloring, Edge Enhancement) could reduce the aesthetics of the original design, diminishing immersion. Moreover, they felt some tools (*e.g.*, Guideline) could reduce the challenge level of the original app if it was applied to a VR game, which "feels like cheating" (L4). However, most participants weighted accessibility and equity more than aesthetics, feeling that SeeingVR democratized the opportunity to experience VR as sighted people do. L9 described his prior experience:

"I stopped playing FIFA because it became more realistic... they changed the visuals suddenly... I was not able to play that game after enjoying it for 10 years... At least [SeeingVR] gives me the option to be able to see. It may not be comparable or directly equal to what you may experience if you don't have a visual impairment, but it's an equitable experience, so that I'm still able to... participate in that game."

Participants believed that SeeingVR could be a "game changer" (L2) in advancing VR accessibility for low vision. As L6 said, "If this ever becomes standard in all VR headsets, I'd definitely invest in a VR [headset]."

Study2: Applying SeeingVR Plugin to Top 10 VR Apps

We examined the generalizability of the SeeingVR plugin by testing it on 10 popular VR applications.

Method. We used the VR app rank on Steam Spy [57], a website that provides statistics about Steam apps by collecting and analyzing data from Steam users' profiles. It ranked all Steam apps based on different features, such as price, number of owners, and playtime. We filtered the rank with the "VR+Only" tag to generate a rank specifically for VR apps, and sorted by number of owners (*i.e.*, popularity). Since our plugin focuses on Unity apps, we looked for the top 10 Unity apps (Table 2) on the VR app rank list by testing each app with the Illusion Plugin Architecture. We ran the SeeingVR plugin on these VR apps, and tested each of our 9 *post hoc* low vision tools to examine whether they could successfully modify the app.

Results. When collecting Unity apps on the VR app rank list, we found that nine out of the top 10 VR apps were

Table 2: SeeingVR plugin success for the Top 10 Unity apps.

VR apps	Ineffective tools in SeeingVR
The Lab	Brightness Lens; Contrast Lens; Edge Enhancement
Rec Room	None
Waltz of the Wizard	None
Beat Saber	None
SUPERHOT VR	None
Arizona Sunshine	None
Job Simulator	Text Augmentations; Text to Speech (TTS)
Space Pirate Trainer	None
Tilt Brush	Edge Enhancement
Audioshield	Edge Enhancement; Text Augmentation; TTS

built with Unity. The top 10 Unity apps were ranked within the top 13 on the original list for all VR apps. This verified that Unity is the major platform for VR app development; Unity's popularity is the reason why our SeeingVR plugin and developer toolkit focus on this platform.

In general, our plugin could be successfully applied to all 10 Unity VR apps. For 60% of the apps, all of the nine low vision tools were effective. Table 2 lists the tools that were ineffective for each app. We investigated what caused the failure of some tools and found two main reasons: (1) The tool implementation conflicted with the original app implementation: the tools implemented by post-processing shaders (*e.g.*, Edge Enhancement) conflicted with the special rendering method [76] in *The Lab* and the original post-processing shaders in *Tilt Brush* and *Audioshield*. (2) The app does not have any text component (*Job Simulator*) or the text component was obsolete (*Audioshield*), so that Text Augmentation and Text to Speech were ineffective.

Study3: Evaluating the Unity Toolkit with Developers

To evaluate the usability of the Unity toolkit we obtained developers' feedback on the three accessibility features and the impact of the toolkit on their development process.

Method. We recruited six Unity developers (D1-D6, 2 female) with Unity experience of more than 5 months, whose age ranged from 23 to 47 (*mean*=31.8). They had all worked on AR or VR projects that are (or will be) publicly released. The study lasted one hour. We began by discussing their demographics, Unity development experience, and whether they had developed any accessibility features for VR.

We then asked the participants to apply our Unity toolkit to a Unity project. We first gave a tutorial about our toolkit, introducing how to use the 14 low vision tools and the three accessibility features. We also prepared an API document, explaining the use of each low vision tool and its parameters. We asked participants to apply the toolkit to a Unity project by adding all the low vision tools and filling in the metadata required by the three accessibility features. Participants could either modify their own Unity project, or use a sample project (*EscapeVR*) that we provided; D4 (*age=47*, *gender=m*) and D5 (*23*, *f*) used their own projects. We provided an HTC Vive headset and controllers to enable participants to test their project. During the study, participants thought aloud, talking about how they used this toolkit and giving their feedback on the low vision tools.

We ended the study with an interview, asking participants' general experience with the Unity toolkit and the three accessibility features. We also asked how likely they would be to use the toolkit in their real development process, and whether they would rather use the toolkit or allow our plugin to modify their app *post hoc.*

Findings. ***Lack of Accessibility Guidelines for VR.*** No participants had worked on any accessibility features when developing a VR project. Besides the fact that accessibility was not their development priority, they were unaware of any accessibility guidelines they could follow to make a VR product accessible. As D1 (*31*, *m*) described: "Sometimes people just assume accessibility in VR is the same as accessibility [of a standard GUI] on a 2D screen, which is not really right. We sometimes got asked by [the] accessibility team [at our company], 'You need to be accessible.' But they don't really understand what accessibility is in the VR context. You guys are the first that actually look this deeply into this problem."

Experience with the Unity Toolkit. All participants agreed that our Unity toolkit was easy to learn and use. They all understood the three extended accessibility features and used these features to support the tools. Except for D2 (*35*, *m*), who had concerns that filling in the accessibility features (especially Description) was too time-consuming, participants believed that the toolkit would not add much workload and they would probably use it especially when building something to ship. Beyond accessibility purposes, some developers (*e.g.*, D5, D6) felt that the tools could even be useful for sighted users. "This [magnification] is nice. I would use it myself" (D6, *26*, *m*).

As opposed to letting an external plugin modify the applications *post hoc*, all developers preferred using the toolkit so that they could fully control their app design: they wanted to customize the tools and implement suitable interactions for them to provide users the best experience. For example, D3 (*29*, *f*) attached the Magnification Lens to the controller, using it as a handheld magnifier. D4 (*47*, *m*) considered designing an accessible version of texture for the Recoloring tool, to avoid diminishing the immersion of the original scene. D1 (*31*, *m*) noted, "You should give developers the flexibility to optimize [the tools] for the original VR experience. Because even for people with disability, you still want them to enjoy the original game, which is handcrafted for the best experience, right?"

6 DISCUSSION

Our evaluation demonstrated that SeeingVR is effective for people with a variety of low vision conditions. Using SeeingVR's tools, participants with low vision were able to optimize their visual experience, thus completing tasks in VR more quickly and accurately. Moreover, users reported finding VR more enjoyable when using SeeingVR. Our study also confirmed the feasibility of the two approaches to applying SeeingVR: the plugin can successfully augment the 10 most popular Unity VR applications, and developers felt our toolkit could improve VR accessibility with reasonable workload.

To our knowledge, SeeingVR is the first exploration on VR accessibility for people with low vision. Different from websites and apps on 2D displays, which have relatively mature accessibility guidelines [2, 74, 75], there are currently no VR-specific guidelines that designers or developers can follow to make 3D virtual spaces more accessible. We sought to advance VR accessibility by exploring the design space from different perspectives. Besides supporting various low vision conditions and many potential types of interactions in VR, we also considered the difference between 2D and 3D, and the transition from the real to virtual world. In addition to the tools that could be used on 2D displays (*e.g.*, Magnification, Brightness Lens), we designed tools that leveraged and enhanced 3D information. For example, Edge Enhancement segments the environment based on depth change, and Depth Measurement was designed to enhance depth perception. Since VR simulates the real world, we also transferred real-world low vision technologies to VR. For optical devices, we created Bifocal Lens to simulate bifocals. In terms of high-tech, we mimicked assistive smartphone apps to enable people to benefit from these technologies in VR. While our initial toolkit focused on low vision, we hope SeeingVR will stimulate a discussion on VR accessibility approaches and standards for many disabilities.

Compared with the plugin that modifies a VR application *post hoc*, the Unity toolkit provided more accessibility support but also required more developer effort. Although accessibility was not a priority that developers in our study had considered when creating VR, they still wanted to have control of the tools in SeeingVR, customizing the design and interaction of the tools to better convey their original design intent. Participants with low vision also indicated that they cared about the original VR experience, having concerns that some tools may reduce immersion and aesthetics. These findings demonstrate the importance and complexity of balancing tradeoffs among the original VR experience, accessibility, and developers' effort when designing accessibility guidelines for VR.

With so many low vision support options, it is necessary to design interaction techniques to allow users to select and adjust SeeingVR tools to best match their abilities and context. To avoid conflict with the original interaction in the VR apps, we used Wizard of Oz voice control for tool selection. However, we also asked the low vision participants and the developers how they would prefer to control the tools. While all participants with low vision wanted to adjust the tools in context via a shortcut command (*e.g.*, a voice command or a button on the controller), they still need a menu that includes all the tools and is accessible enough for them to navigate. Developers also mentioned that they would have to implement a menu to expose the tools to users; they suggested the Unity toolkit provide this menu, so that they could integrate the menu to their project directly. Some developers were willing to design and implement shortcut gestures for the most frequently used tools, such as the gesture of "pulling the magnifier out from a particular part of their body" (D3). Besides letting users manually adjust the tools, some participants with low vision (*e.g.*, L9) also suggested making the tool selection adaptive to the current scene by automatically detecting the characteristics of the scene. Future work should consider designing suitable interaction techniques to enable convenient control of SeeingVR.

In addition to its use in VR applications, our SeeingVR tools can be extended to non-VR 3D applications that run on regular 2D displays (*e.g.*, *Second Life*). While current operating systems come with assistive technologies, such as screen magnifiers, these basic tools target 2D interfaces and cannot help users perceive the 3D information presented in the virtual world. Moreover, many 3D apps may not even support such built-in assistive technologies because of their use of specific rendering methods. Since SeeingVR was designed to enhance 3D renderings, we see SeeingVR's potential to make (non-VR) 3D interfaces on 2D displays more accessible for people with low vision. Testing SeeingVR on this type of applications to evaluate its generalizability is one direction for future work.

Our design of SeeingVR demonstrated the benefits of translating low vision technologies from the real to virtual world. Future work could investigate the potential of translating low vision tools we designed for SeeingVR to the real world. For example, the Depth Measurement tool could be adapted into the real world via projection technology: with a handheld projector and a depth camera, a user could project a virtual ball into the environment, indicating depth information with different colors. Moreover, as newer-generation VR headsets will likely support video see-through AR [87] (*e.g.*, Vive Pro integrates stereo camera to support AR capability [30]), all low vision tools in SeeingVR can be potentially implemented through

AR to augment the real world based on some semantic understanding of the real environment via computer vision and other sensory technology.

7 CONCLUSION

In this paper, 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. We evaluated SeeingVR with 11 participants with low vision, showing that SeeingVR enabled users to better enjoy VR and complete tasks more quickly and accurately. Our testing with 10 popular VR apps demonstrated the robustness of our plugin, while the study with six developers showed our Unity toolkit was easy to use. While SeeingVR focused on low vision, we hope our work can inspire the design of general accessibility standards for VR.

ACKNOWLEDGMENTS

We thank all our participants. We also thank Mar Gonzalez Franco, Ken Hinckley, Jackie Yang, and Jeremy Hartmann for their feedback.

REFERENCES

[1] PF Adams and PM Barnes. 2006. Summary health statistics for the US population: National Health Interview Survey, 2004. *National Center for Health Statistics* 10, 229 (2006), 1–104.

[2] Android. 2018. Make apps more accessible | Android Developers. https://developer.android.com/guide/topics/ui/accessibility/apps Last accessed 21 December 2018.

[3] American Optometric Association (AOA). 2018. Common Types of Low Vision. https://www.aoa.org/patients-and-public/caring-for-your-vision/low-vision/common-types-of-low-vision last accessed 7 July 2015.

[4] Apple. 2018. Vision Accessibility - iPhone. https://www.apple.com/accessibility/iphone/vision/ Last accessed 21 December 2018.

[5] Joseph Bates. 1992. Virtual reality, art, and entertainment. *Presence: Teleoperators & Virtual Environments* 1, 1 (1992), 133–138.

[6] 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.

[7] Jeffrey P Bigham, Ryan S Kaminsky, Richard E Ladner, Oscar M Danielsson, and Gordon L Hempton. 2006. WebInSight:: making web images accessible. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 181–188.

[8] Jeffrey P Bigham, Craig M Prince, and Richard E Ladner. 2008. WebAnywhere: a screen reader on-the-go. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*. ACM, 73–82.

[9] Chetz Colwell, Helen Petrie, Diana Kornbrot, Andrew Hardwick, and Stephen Furner. 1998. Haptic virtual reality for blind computer users. In *Proceedings of the third international ACM conference on Assistive technologies*. ACM, 92–99.

[10] eSight. 2018. eSight Electronic Glasses. https://www.esighteyewear.com/homex last accessed 21 December 2018.

[11] Eusth. 2016. Illusion Plugin Architecture. https://github.com/Eusth/IPA Last accessed 21 December 2018.

[12] MR Everingham, BT Thomas, T Troscianko, et al. 1999. Head-mounted mobility aid for low vision using scene classification techniques. *The International Journal of Virtual Reality* 3, 4 (1999), 3.

[13] American Foundation for the Blind (AFB). 2018. Screen Readers and Screen Magnifiers: An Introduction to Computer Accessibility Software. http://suo.im/5rv11h

[14] GearVR. 2018. GearVR Framework. http://www.gearvrf.org/ Last accessed 21 December 2018.

[15] José Luis González-Mora, A Rodriguez-Hernandez, Enrique Burunat, F Martin, and Miguel A Castellano. 2006. Seeing the world by hearing: Virtual Acoustic Space (VAS) a new space perception system for blind people.. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, Vol. 1. IEEE, 837–842.

[16] Google. 2018. Android accessibility overview - Android Accessibility Help. https://support.google.com/accessibility/android/answer/6006564?hl=en last accessed 21 December 2018.

[17] Anhong Guo, Jeeeun Kim, Xiang'Anthony' Chen, Tom Yeh, Scott E Hudson, Jennifer Mankoff, and Jeffrey P Bigham. 2017. Facade: Auto-generating tactile interfaces to appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5826–5838.

[18] Stephen L Hicks, Iain Wilson, Louwai Muhammed, John Worsfold, Susan M Downes, and Christopher Kennard. 2013. A depth-based head-mounted visual display to aid navigation in partially sighted individuals. *PloS one* 8, 7 (2013), e67695.

[19] Bill Holton. 2014. A Review of iOS Access for All: Your Comprehensive Guide to Accessibility for iPad, iPhone, and iPod Touch, by Shelly Brisbin. *AccessWorld Magazine* 15, 7 (2014), aw150706.

[20] Amy Hurst, Scott E Hudson, and Jennifer Mankoff. 2010. Automatically identifying targets users interact with during real world tasks. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 11–20.

[21] Chuck Huss and Anne Corn. 2004. Low Vision Driving with Bioptics: An Overview. *Journal of Visual Impairment and Blindness* 98, 10 (2004), 641–653.

[22] Alex D Hwang and Eli Peli. 2014. An augmented-reality edge enhancement application for Google Glass. *Optometry and vision science: official publication of the American Academy of Optometry* 91, 8 (2014), 1021.

[23] Gunnar Jansson, H Petrie, C Colwell, D Kornbrot, J Fänger, H König, K Billberger, Andrew Hardwick, and Stephen Furner. 1999. Haptic virtual environments for blind people: Exploratory experiments with two devices. *International journal of virtual reality* 4, 1 (1999), 10–20.

[24] Katerina Kalyvioti and Tassos A Mikropoulos. 2014. Virtual Environments and Dyslexia: A literature review. *Procedia Computer Science* 27 (2014), 138–147.

[25] Shaun K Kane, Meredith Ringel Morris, and Jacob O Wobbrock. 2013. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 22.

[26] Richard L Kline and Ephraim P Glinert. 1995. Improving GUI accessibility for people with low vision. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 114–121.

[27] Andreas Kunz, Klaus Miesenberger, Limin Zeng, and Gerhard Weber. 2018. Virtual Navigation Environment for Blind and Low Vision People.

In *International Conference on Computers Helping People with Special Needs*. Springer, 114–122.

[28] Orly Lahav, Hadas Gedalevitz, Steven Battersby, David Brown, Lindsay Evett, and Patrick Merritt. 2018. Virtual environment navigation with look-around mode to explore new real spaces by people who are blind. *Disability and rehabilitation* 40, 9 (2018), 1072–1084.

[29] Orly Lahav and David Mioduser. 2001. Multisensory virtual environment for supporting blind persons' acquisition of spatial cognitive mapping–a case study. In *EdMedia: World Conference on Educational Media and Technology*. Association for the Advancement of Computing in Education (AACE), 1046–1051.

[30] Ben Lang. 2018. Vive Pro Gets AR Capabilities Thanks to HTC's New Tools for Front-facing Cameras. http://suo.im/4PYzCi/. Last accessed 21 December 2018.

[31] Mary Leach. 2013. Researchers Show Peripheral Prism Glasses to be a Simple, Inexpensive Rehabilitation Tool. https://www.masseyeandear.org/news/press-releases/2013/11/2013-prism-glasses last accessed 21 December 2018.

[32] Anatole Lécuyer, Pascal Mobuchon, Christine Mégard, Jérôme Perret, Claude Andriot, and J-P Colinot. 2003. HOMERE: a multimodal system for visually impaired people to explore virtual environments. In *Virtual Reality, 2003. Proceedings. IEEE*. IEEE, 251–258.

[33] lucassa3. 2018. EscapeVR-HarryPotter. https://github.com/lucassa3/EscapeVR-HarryPotter Last accessed 21 December 2018.

[34] Gang Luo and Eli Peli. 2006. Use of an augmented-vision device for visual search by patients with tunnel vision. *Investigative ophthalmology & visual science* 47, 9 (2006), 4152–4159.

[35] MAGic. 2018. The best screen magnifier - MAGic. https://www.freedomscientific.com/Products/LowVision/MAGic last accessed 21 December 2018.

[36] Shachar Maidenbaum, Shelly Levy-Tzedek, Daniel-Robert Chebat, and Amir Amedi. 2013. Increasing accessibility to the blind of virtual environments, using a virtual mobility aid based on the" EyeCane": Feasibility study. *PloS one* 8, 8 (2013), e72555.

[37] J Stephen Mansfield, Gordon E Legge, and Mark C Bane. 1996. Psychophysics of reading. XV: Font effects in normal and low vision. *Investigative Ophthalmology & Visual Science* 37, 8 (1996), 1492–1501.

[38] Tom H Margrain. 2000. Helping blind and partially sighted people to read: the effectiveness of low vision aids. *British Journal of Ophthalmology* 84, 8 (2000), 919–921.

[39] Robert W Massof and Douglas L Rickman. 1992. Obstacles encountered in the development of the low vision enhancement system. *Optometry and vision science: official publication of the American Academy of Optometry* 69, 1 (1992), 32–41.

[40] David Maulsby, Saul Greenberg, and Richard Mander. 1993. Prototyping an intelligent agent through Wizard of Oz. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 277–284.

[41] Zahira Merchant, Ernest T Goetz, Lauren Cifuentes, Wendy Keeney-Kennicutt, and Trina J Davis. 2014. Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education* 70 (2014), 29–40.

[42] Microsoft. 2018. Cognitive Services | Microsoft Azure. https://azure.microsoft.com/en-us/services/cognitive-services/?v=18.44b Last accessed 21 December 2018.

[43] Microsoft. 2018. Hear text read aloud with Narrator. https://support.microsoft.com/en-us/help/17173/windows-10-hear-text-read-aloud Last accessed 21 December 2018.

[44] 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.

[45] Phil Parette and Marcia Scherer. 2004. Assistive technology use and stigma. *Education and Training in Developmental Disabilities* (2004), 217–226.

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

[47] Lorenzo Picinali, Amandine Afonso, Michel Denis, and Brian FG Katz. 2014. Exploration of architectural spaces by blind people using auditory virtual reality for the construction of spatial knowledge. *International Journal of Human-Computer Studies* 72, 4 (2014), 393–407.

[48] Sara Reardon. 2011. Playing by ear.

[49] Marti L Riemer-Reiss and Robbyn R Wacker. 2000. Factors associated with assistive technology discontinuance among individuals with disabilities. *Journal of Rehabilitation* 66, 3 (2000).

[50] Jaime Sánchez and Mauricio Lumbreras. 1999. Virtual environment interaction through 3D audio by blind children. *CyberPsychology & Behavior* 2, 2 (1999), 101–111.

[51] JH Sánchez and MA Sáenz. 2006. Assisting the mobilization through subway networks by users with visual disabilities. In *Proc. 6th Intl Conf. Disability, Virtual Reality & Assoc. Tech., Esbjerg, Denmark*. 183–190.

[52] David W Schloerb, Orly Lahav, Joseph G Desloge, and Mandayam A Srinivasan. 2010. BlindAid: Virtual environment system for self-reliant trip planning and orientation and mobility training. In *Haptics Symposium, 2010 IEEE*. IEEE, 363–370.

[53] Liz Segre. 2018. Low Vision Aids For Reading. https://www.allaboutvision.com/lowvision/reading.htm Last accessed 21 December 2018.

[54] Yoshikazu Seki and Tetsuji Sato. 2011. A training system of orientation and mobility for blind people using acoustic virtual reality. *IEEE Transactions on neural systems and rehabilitation engineering* 19, 1 (2011), 95–104.

[55] SK Semwal. 2001. MoVE: Mobiltiy training in haptic virtual environment. *piglet. uccs. edu/˜ semwal/NSF2001PS. pdf* (2001).

[56] Kristen Shinohara and Jacob O Wobbrock. 2011. In the shadow of misperception: assistive technology use and social interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 705–714.

[57] steamspy. 2018. SteamSpy - All the data and stats about Steam games. https://steamspy.com/tag/VR+Only Last accessed 21 December 2018.

[58] Sarit Szpiro, Yuhang Zhao, and Shiri Azenkot. 2016. Finding a store, searching for a product: a study of daily challenges of low vision people. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 61–72.

[59] 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.

[60] Hironobu Takagi and Chieko Asakawa. 2000. Transcoding proxy for nonvisual web access. In *Proceedings of the fourth international ACM conference on Assistive technologies*. ACM, 164–171.

[61] Mauro Teófilo, Vicente F Lucena, Josiane Nascimento, Taynah Miyagawa, and Francimar Maciel. 2018. Evaluating accessibility features designed for virtual reality context. In *Consumer Electronics (ICCE), 2018 IEEE International Conference on*. IEEE, 1–6.

[62] Virgil Tiponut, Zoltan Haraszy, Daniel Ianchis, and Ioan Lie. 2008. Acoustic virtual reality performing man-machine interfacing of the blind. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. WSEAS.

[63] MA Torres-Gil, O Casanova-Gonzalez, and José Luis González-Mora. 2010. Applications of virtual reality for visually impaired people. *WSEAS transactions on computers* 9, 2 (2010), 184–193.

[64] Shari Trewin, Vicki L Hanson, Mark R Laff, and Anna Cavender. 2008. PowerUp: an accessible virtual world. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 177–184.

[65] Shari Trewin, Mark Laff, Vicki Hanson, and Anna Cavender. 2009. Exploring visual and motor accessibility in navigating a virtual world. *ACM Transactions on Accessible Computing (TACCESS)* 2, 2 (2009), 11.

[66] Dimitrios Tzovaras, Konstantinos Moustakas, Georgios Nikolakis, and Michael G Strintzis. 2009. Interactive mixed reality white cane simulation for the training of the blind and the visually impaired. *Personal and Ubiquitous Computing* 13, 1 (2009), 51–58.

[67] Dimitrios Tzovaras, Georgios Nikolakis, George Fergadis, Stratos Malasiotis, and Modestos Stavrakis. 2002. Design and implementation of virtual environments training of the visually impaire. In *Proceedings of the fifth international ACM conference on Assistive technologies*. ACM, 41–48.

[68] Unity. 2016. Edge Detection - Unity. https://docs.unity3d.com/540/Documentation/Manual/script-EdgeDetectEffect.html Last accessed 21 December 2018.

[69] Unity. 2018. Game Objects and Components - Unity. https://unity3d.com/learn/tutorials/topics/interface-essentials/game-objects-and-components Last accessed 21 December 2018.

[70] Fernando Vargas-Martín and Eli Peli. 2001. P-16: Augmented View for Tunnel Vision: Device Testing by Patients in Real Environments. In *SID Symposium Digest of Technical Papers*, Vol. 32. Wiley Online Library, 602–605.

[71] VENTUREBEAT. 2017. 59% of VR Developers Use Unity, But Devs Make More Money With Unreal. https://uploadvr.com/vr-developers-unity-unreal/ Last accessed 21 December 2018.

[72] Maria Virvou and George Katsionis. 2008. On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE. *Computers & Education* 50, 1 (2008), 154–178.

[73] W3C. 2018. Contrast (Minimum): Understanding Success Criterion 1.4.3. https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html Last accessed 21 December 2018.

[74] W3C. 2018. Web Content Accessibility Guidelines (WCAG) Overview. https://www.w3.org/WAI/standards-guidelines/wcag/ Last accessed 21 December 2018.

[75] WebAIM. 2017. Designing for Screen Reader Compatibility. https://webaim.org/techniques/screenreader/ Last accessed 21 December 2018.

[76] Jason Weimann. 2016. Using the Valves - Lab Renderer for VR. https://unity3d.college/2016/08/02/using-valves-lab-renderer-vr/ Last accessed 21 December 2018.

[77] World Health Organization (WHO). 2018. Blindness and Vision Impairment. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment Last accessed 21 December 2018.

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

[79] James S Wolffsohn and Rachael C Peterson. 2003. A review of current knowledge on Electronic Vision Enhancement Systems for the visually impaired. *Ophthalmic and Physiological Optics* 23, 1 (2003), 35–42.

[80] Alice Wong, Hannah Gillis, and Ben Peck. 2018. VR Accessibility Survey: Survey for People with Disabilities. https://drive.google.com/file/d/0B0VwTVwReMqLMFIzdzVVaVdaTFk/view Last accessed 21 December 2018.

[81] Xiaoyi Zhang, Anne Spencer Ross, Anat Caspi, James Fogarty, and Jacob O Wobbrock. 2017. Interaction Proxies for Runtime Repair and Enhancement of Mobile Application Accessibility. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 6024–6037.

[82] Xiaoyi Zhang, Tracy Tran, Yuqian Sun, Ian Culhane, Shobhit Jain, James Fogarty, and Jennifer Mankoff. 2018. Interactiles: 3D Printed Tactile Interfaces to Enhance Mobile Touchscreen Accessibility. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 131–142.

[83] 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.

[84] Yuhang Zhao, Michele Hu, Shafeka Hashash, and Shiri Azenkot. 2017. Understanding Low Vision People's Visual Perception on Commercial Augmented Reality Glasses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4170–4181.

[85] 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.

[86] Yuhang Zhao, Sarit Szpiro, Jonathan Knighten, and Shiri Azenkot. 2016. CueSee: exploring visual cues for people with low vision to facilitate a visual search task. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 73–84.

[87] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. 2008. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 193–202.

[88] Michael Zyda. 2005. From visual simulation to virtual reality to games. *Computer* 38, 9 (2005), 25–32.