# Creating an Intuitive and Non-Obstructive Passthrough UI for Sound Localization

Siddhant Gupta
Stanford University
siddg@stanford.edu

Timothy Jacques
Stanford University
tjacques@stanford.edu

## Abstract

*Recent advancements in commercially available virtual and extended reality devices, such as Apple's Vision Pro and Meta's Quest line of headsets, have led to the emergence of more applications designed to augment sensory data available to users and mitigate the effects of certain disabilities. Specifically, visual cues through overlays in the passthrough mode of these headsets can be used to supplement hearing for those who are hard of hearing or unable to localize sound sources. In this work, we explore the development of a user interface (UI) to serve this purpose of helping users localize sound, while being minimally obstructive and intuitive to new users.*

## 1. Introduction

Virtual reality (VR) has rapidly evolved from isolated, fully synthetic environments to hybrid experiences that blend real and virtual content. Modern head-mounted displays (HMDs) now incorporate high-fidelity controllers, room-scale tracking, and increasingly sophisticated optics—all designed to immerse users in computer-generated worlds. Concurrently, extended reality (XR) systems are expanding into augmented reality (AR) and mixed reality (MR) modalities, where computer-rendered imagery coexists with or augments the user's physical surroundings. This spectrum—from fully immersive VR to precisely overlaid AR—enables new applications in gaming, training, design, and accessibility.

One of the most significant advancements in this space is the introduction of passthrough camera access on consumer headsets. Instead of relying on tethered sensors or simple video overlays, passthrough functionality streams real-time, stereoscopic camera feeds directly inside the HMD. Users can see their actual environment—hands, furniture, walls—while the system overlays virtual content or visual effects. Until 2025, passthrough was largely restricted to enterprise-grade MR headsets (e.g., HoloLens), but Meta has recently opened up native passthrough APIs for the Quest 3 and Quest 3S [4]. This democratizes mixed-reality development on a relatively affordable, mass-market device, allowing researchers and indie developers to explore new ways of blending virtual and real-world stimuli.

### 1.1. Cartoon Filter to Overlay Information

Despite these capabilities, most consumer-level AR/VR applications focus on 2D video filters—such as "beautify," "cartoonify," or color-grading effects—applied to smartphone or tablet camera streams. In contrast, there is a clear gap in applying real-time, non-photorealistic stylizations to a 3D passthrough feed. Prior research in stylized augmented reality has demonstrated that painterly or cartoon-like rendering can enhance immersion while reducing computational cost compared to full-resolution, photorealistic rendering. For example, earlier work on stylized AR showed that converting the user's view into a "painting"–style overlay can deliver higher perceived immersion than a low-resolution realistic render, all while saving GPU cycles. In other words, a well-chosen non-photorealistic style—particularly cartoonization—often serves as the "second best" to photorealism: it preserves visual engagement without demanding the same rendering budget.

### 1.2. Targeted Use-Case

Building on this insight, our project explores how a localized cartoon filter can serve an accessibility use case for users who are deaf or hard of hearing. XR headsets include multiple cameras, microphones, and speakers capable of 3D spatial audio. For those who cannot rely solely on auditory cues, visual indicators in AR/VR can effectively compensate. Traditional cues—such as static arrows or colored overlays—often clutter the visual field or demand excessive attentional resources. We hypothesize that a non-photorealistic, cartoon-style overlay—rendered narrowly around a detected sound source—can intuitively guide attention without obstructing most of the user's view.

For individuals who are deaf or hard of hearing—including those with single-sided deafness—locating the origin of everyday sounds can be challenging or impossible. A ringing phone tucked behind a couch, a low-battery chirp from a smoke alarm down the hallway, or someone calling your name from another room: these are scenarios that present safety and convenience concerns. By leveraging pass-through, we can display a localized cartoon overlay precisely at the real-world position of that sound. Instead of relying on static HUD elements, the filter dynamically accentuates the region where the audio is strongest. In doing so, the headset effectively "points" the user toward the sound source without blocking their peripheral vision or requiring manual head sweeping.

To validate this hypothesis, our work is divided into two complementary evaluations: Pure VR and Real-world Passthrough.

### 1.3. Pure VR Evaluation

We construct a virtual environment in the Unity game engine consisting of a three-room house populated with realistic props, multiple sound-emitting objects, and acoustically realistic reflections. Within this simulated setting, we implement our cartoon shader—combining Sobel-based edge detection, color quantization, Gaussian blur, and gamma correction—around the direction of a target sound. Participants must locate a specified sound under different visual-cue conditions (cartoon overlay, simple crosshair, semi-transparent region). We record task completion times and collect qualitative feedback on how distracting each cue is when attention is divided. We also create some reference UI designs such as a semi-transparent coloring and a small crosshair to compare our method against as a baseline.

### 1.4. Real-World Passthrough Evaluation

Using a Quest 3 headset with live passthrough camera feed, we port the same cartoon filter into the actual mixed-reality pipeline. Users perform everyday tasks—such as picking up objects or walking around—while the cartoon overlay indicates a real-world sound source. We evaluate perceived obstruction, visual quality, and latency through subjective questionnaires and task-performance measures.

All filter effects (cartoon shader, crosshair, and semi-transparent overlays) are implemented at the shader level in high-level shader language (HLSL) and driven by C# scripts in Unity. By comparing performance and user experience across both fully virtual and live passthrough contexts, we aim to demonstrate that localized, stylized rendering can effectively guide attention toward spatial audio cues while minimizing visual interference. This work not only leverages recent passthrough capabilities on consumer headsets but also lays the groundwork for more intuitive, noninvasive accessibility tools in next-generation XR experiences.

## 2. Related Works

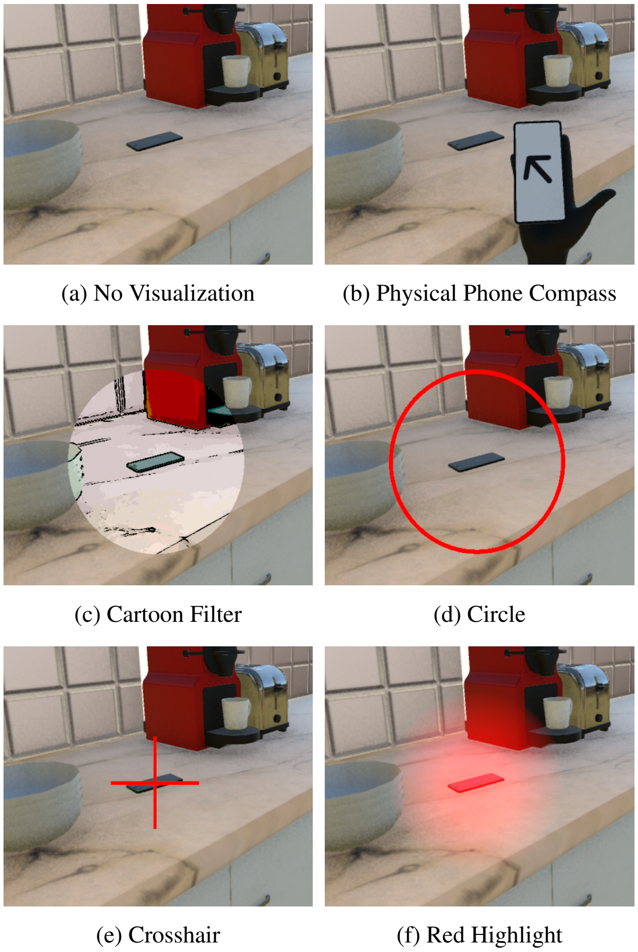
### 2.1. UI and Filters in Passthrough

Prior research has explored non-photorealistic rendering (NPR) techniques in AR/VR to blend virtual content with the real world. For example, Steptoe et al. (ISMAR 2014) applied an edge-detection "cartoon" filter to video see-through AR, finding that stylization increased visual coherence – users had difficulty distinguishing real objects from virtual ones, indicating a more unified and immersive scene. Such stylized AR can also improve task performance by reducing visual distractions. Koshi et al. (IEEE VR 2019) demonstrated an "augmented concentration" approach in which video see-through AR with lowered detail (visual noise reduction) helped users solve math problems faster by filtering out background clutter [2]. These works suggest that applying a painterly or sketch-like filter to passthrough MR can enhance immersion and even user focus, which aligns with our goal of using a cartoon-style filter to integrate guidance cues smoothly into the real world view.

### 2.2. Visual Cues to Localize Sound

*XR Accessibility for Deaf/Hard-of-Hearing Users:* There is a growing body of HCI research on conveying audio information visually in XR to assist users with hearing impairments. In the VR domain, Li et al. (ASSETS 2022) introduced SoundVizVR, a system that visualizes spatial audio cues in immersive VR [3]. SoundVizVR combines on-object visual indicators (e.g. an icon or vibration effect on the sound-emitting virtual object) with heads-up mini-maps to indicate each sound's location, loudness, and duration. In user studies with Deaf and hard-of-hearing (DHH) participants, the authors found that a full-field mini-map plus on-object indicator was most effective, enabling DHH users to quickly locate sound sources in a virtual scene. These VR-based solutions illustrate how visual augmentation of audio events can convey crucial spatial cues to users who cannot hear them, a strategy our project extends into a passthrough AR context.

*Passthrough AR and Spatial Audio for Accessibility*: Recent work has started translating these ideas to augmented reality using passthrough devices and smart glasses. Asakura (Sensors 2023) developed an AR system for everyday sound awareness, aimed at DHH users in home environments [1]. The system uses a wearable passthrough AR display to detect and classify household sounds (e.g. doorbells, alarms) and then presents visual icons for each sound alongside a real-time animated spectrogram. In a field evaluation, Asakura found that combining machine-learned sound type icons with a dynamic audio spectrogram significantly improved DHH users' comfort and situational awareness in their daily lives [1]. This demonstrates the



(a) No Visualization

(b) Physical Phone Compass

(c) Cartoon Filter

(d) Circle

(e) Crosshair

(f) Red Highlight

Figure 1: All evaluated visualizations indicating a tracked object on a kitchen counter. (b) serves as a baseline that is feasible without XR entirely.

feasibility and benefit of overlaying visual sound cues onto one's real-world view.

Our project builds on these insights by using a Meta Quest 3 passthrough MR headset with a stylized (cartoon) filter to provide spatial audio guidance. In essence, we integrate the visual clarity and coherence afforded by stylized rendering with proven audio-visual cue techniques from prior accessibility research. By doing so, we aim to create an MR experience where important sounds (e.g. a speaking person or an alert) are highlighted visually in the cartoon-filtered scene, allowing DHH users to perceive and localize audio events that would otherwise be missed – a direct extension of the motivations and findings of the related work above.

## 3. Methodologies

The process of our work can be broadly divided into the categories of filter/UI design, sound and environment setup, shader implementation, and evaluation. In filter/UI design, we iterate to develop a stylized cartoon filter, working on a reference image first and then making adjustments and parameterizations to adapt it to virtual reality. Since we did not have access to an ambisonic microphone for precise sound localization, we split our project evaluation into two environments - a virtual environment to simulate how the UI would respond to sounds in a semi-realistic environment, and a stereo passthrough environment implementing the cartoon filter to evaluate the effect on realworld environments. Next, we discuss the process of creating a room and modeling sounds for the VR world. Lastly, we discuss the methodology of implementing our filters in Unity with the Meta XR plugin, as well as the setup leading to our evaluations.

### 3.1. Filter Designs

#### 3.1.1 Cartoon Filter Implementation

The first step of generating our stylized cartoon filter was prototyping on a single image. We selected a relatively bright image and wrote a unlit 2-D textured shader in HLSL, applying it to a cube with the material set to our input image for our first iteration. In this rudimentary implementation, we had only color quantization and Sobel filtering with parameterized edge thresholding and color levels. The purpose of color quantization was to flatten the colors of the image to a smaller amount of colors and effectively smooth out textures, while the purpose of Sobel filtering was to enhance the visibility of "major" edges in the image. In order to target an output that was both clearly stylized and still preserved detail for maximum immersion, we sought a graphic novel style appearance.

After our initial prototype, we noticed in large patches of the image with a gradient color, like the sky, color quantization added a noisy effect to the output image where some pixels were slightly different hue than the surrounding ones. In order to mitigate this, we added a gaussian blur kernel before the color quantization step.

When porting our proof-of-concept implementation into a full-screen shader for passthrough and virtual evaluation, we needed to implement a few more changes. First, since our prototype shader was developed for a bright image, we did not compensate for the appearance in a dimmer environment like passthrough in a bedroom - we were able to add exposure and gamma adjustment to mitigate this. Second, we added parameters to localize our filter to a region instead of the entire screen. Our post-process rendering shader would take in the world-space coordinates of the sound source, size, and intensity (weighted blend between cartoon filter and original image) as parameters from our Unity environment. In order to compare our implementation with others in a user evaluation, we created a baseline visualization of using a phone to find sound, as well as three other passthrough UIs.

### 3.1.2 Reference UIs

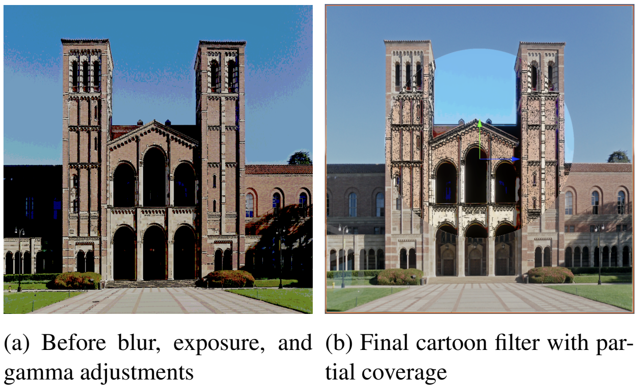
To evaluate our cartoon filter independently, several reference UIs were designed to compare to for our user study in the VR evaluation. As displayed in Fig 1, there were six total options:

- *No Visualization*: Users rely on their hearing to locate the object.
- *Physical Phone Compass*: A phone held in hand that points directly to the object's location. This serves as a non-XR reference.
- *Circle*: An XR-based red circle around the location of the object in the user's view.
- *Highlight*: An XR-based red highlight around the location of the object in the user's view.
- *Crosshair*: An XR-based red crosshair that is centered on the object in the user's view.
- *Cartoon*: An XR-based cartoon filter centered on the object in the user's view.

Since the phone is able to point to sound objects off screen as well, each of our passthrough filters had an off-screen indicator - if a sound is off screen, there will be a demarcation towards the edge of a screen in the nearest direction to the sound.

## 3.2. Shader Implementation

In VR, each eye's view-projection matrix is used to transform a 3D world-space point (passed in as the _RegionCenter uniform) into normalized device coordinates (NDC) that range from 0 to 1 in UV space. At the start of the fragment shader, a built-in macro ensures that the combined view-projection matrix corresponds to the correct eye. The shader then calls a function—ComputeNormalizedDeviceCoordinatesWithZ—with _RegionCenter.xyz and the per-eye view-projection matrix. This function performs a complete world→clip→NDC conversion (including the perspective divide by w). Unity automatically remaps the usual NDC range (–1 to +1) into UV coordinates between 0 and 1, so the X and Y outputs become direct screen-space UV coordinates inside that eye's render texture, and the Z output indicates whether the point is in front of (positive) or behind (negative) the camera. Because this projection happens inside a full-screen post-process pass that samples from the already-rendered scene texture, no new 3D geometry is drawn; instead, the shader simply knows where that world-space point would have appeared on the final color buffer.


(a) Before blur, exposure, and gamma adjustments

(b) Final cartoon filter with partial coverage

Figure 2: Cartoon filter implementation steps

After computing those UV coordinates, the shader checks whether the projected point is off-screen—either because its depth is negative (behind the camera) or because its UV coordinates lie outside the [0, 1] range in X or Y. If the point is on-screen, the shader uses those UV coordinates directly as the center of a circular mask, with a radius set by the _RegionRadius parameter. If the point is off-screen, the shader computes the direction from the center of the screen (0.5, 0.5) toward the extrapolated UV point and places a larger "indicator" circle at a fixed distance in that direction, ensuring part of the circle overlaps the edge of the viewport. In both cases, any pixel whose UV coordinate is outside the defined circle simply samples the original scene color and returns it unchanged, preserving the unaltered view outside that mask.

Inside the circular region, the shader applies a multi-step "toon" pipeline. First, it performs a 3 × 3 bilateral blur to smooth each pixel's color while preserving sharp color transitions. To do this, it samples nine neighboring texels (offset by one texel in each direction) and weights each sample by an exponential function of the squared color difference to the original pixel; this ensures pixels with similar colors contribute most heavily. The weighted average yields a blurred color. Next, the shader multiplies that blurred color by an exposure factor and then applies an inverse gamma correction, brightening the result and adjusting for perceptual luminance. After that, it computes the pixel's luminance, clamps it to a minimum threshold, and quantizes it into a fixed number of discrete bands (determined by the _Levels parameter). By scaling the entire color so its luminance matches one of those quantized values and then snapping each channel to the same discrete step, the output within the circle gains flat, cel-shaded regions.

Meanwhile, in parallel, the shader samples eight neighboring texels' luminance from the unblurred scene and applies a Sobel filter to compute each pixel's gradient magnitude. If that gradient exceeds an edge-threshold parameter, the shader forces the pixel to black; otherwise, it keeps the quantized color. Finally, it blends between the original (blurred) color and this "toon" color according to a _FilterIntensity parameter, so that each pixel inside the circle transitions smoothly from the unmodified scene to the fully stylized result. The final output is a crisp cartoon effect—complete with flat shading and bold black outlines—around the projected object (or its off-screen indicator), while the rest of the VR scene remains exactly as it was.

## 3.3. Virtual Environment

### 3.3.1 Virtual Home Design

To provide a realistic environment for evaluation, a few hundred free assets from the Unity Asset Store were hand placed into a virtual home. Props that would reasonably be grabbable in a real environment were assigned physics colliders and allowed for user interaction through grabbing and throwing. While the visual fidelity of the environment was fairly low due to limited XR performance requirements, the diversity of items in the environment was high, delivering a reasonable amount of immersion.

### 3.3.2 Acoustic Simulation

For a audio-based study, realistic simulation of room and sound source acoustics is essential to ensure the baseline is valid. For this, we utilized Meta's XR Audio SDK for Unity package, which provides mesh-based ray tracing for audio reverb and reflections, material acoustic modeling, and head-related transfer function (HRTF) support. With this package, the majority of the work is already complete to enable extremely realistic audio simulation, however we needed to individually mark every item in the world with proper acoustic properties and tune the overall system before its performance was satisfactory.

One of the goals for this project was to properly visualize ray-traced audio reflections, as these would be present with an ambisonic microphone in the real world. While the Meta Audio SDK performs reflection ray tracing and related calculations, there is no exposed API to access the internal values. Therefore, we designed a separate ray-tracing system to visualize sound when reflected off walls and doors for a sound source in another room. This ray-casting method worked, however due to inefficiencies in Unity's scripting system, the performance was too poor for reasonable use in our user study. Because these issues would not be present with a real world ambisonic microphone source, we decided against continuing development.

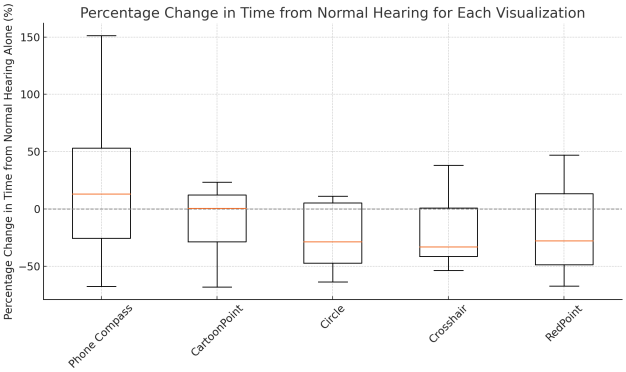


Figure 3: Boxplot showing performance of each visualization when the user is deafened as percentage change in discovery time. Negative percentage means faster discovery time, which is better.

# 4. Evaluation & Results

To evaluate the performance and feasibility of our filter design, we ran a user study among 8 student participants, acquiring quantitative data from the VR simulation (160 individual time trials), and acquiring qualitative experience data by providing the participants with a questionnaire after the VR simulation. While other referenced works in this space compare a specific technology (wearable vs VR) in this task, we choose to do a comparison to a virtual phone as the baseline but also include a comparison of different passthrough filters themselves to see the impact of the UI. Our work also differs in that we produce two distinct evaluations: pure virtual reality, and a proof of concept passthrough, in order to address our constraint of not having an ambisonic microphone.

## 4.1. Quantitative Evaluation

To quantitatively determine the performance of each visualization method, we created a game within the VR simulation that randomly chooses an object in the world to emit a sound. The player is then instructed to find the object and grab it as fast as possible. With this in mind, the visualization method that yields the fastest grab times for each object can be considered the most effective. The users perform the same actions with full hearing, with no hearing + phone compass, and with no hearing + random AR visualizations. Using full hearing alone is considered a baseline. With the results in figure 3, we can extract the following:

**The performance of each of the AR visualizations is about the same.** This is somewhat expected, as the visual differences between each do not strongly change how easily the user can find the object. The main difference between the AR visualizations is expected to be intuitiveness and obstructiveness, both of which are discussed in the qualitative evaluation.

**Using the phone compass alone can be much slower**

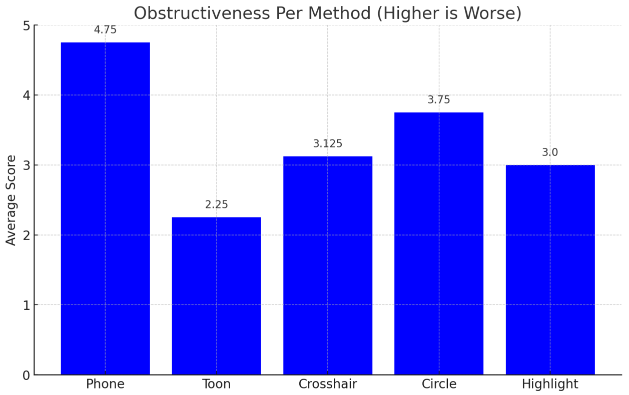


Figure 4: Qualitative user results on obstructiveness of each visualization

**than the AR visualizations.** This is suspected to mainly be due to the 2D arrow on the phone's display versus a direct object annotation. Given that full hearing is 3D, it makes sense this is slower than normal hearing as well.

**The performance of each of the AR visualizations meets or exceeds that of hearing alone.** This is fairly surprising, as it implies that AR sound visualizations would augment even users that are not DHH. However, visualizations do give a pinpoint location rather than the relatively vague nature of hearing, so this makes sense.

**The crosshair visualization performed best, while the cartoon filter performed worst.** We speculate that this could be due to the higher precision of the crosshair filter, while the cartoon filter is less noticeable at first glance. However, the cartoon filter still performs on par with normal hearing performance.

## 4.2. Qualitative Evaluation

Our qualitative evaluation includes feedback from 8 volunteers to answer questions about the immersiveness, intuitiveness, obstructiveness, and overall favorability of the different options presented.

### 4.2.1 Intuitiveness in Localizing Sound

87.5% of users found passthrough to be more intuitive than using the phone to find a sound source, and found it better than just sound alone. For preference (intuition-wise) the aggregate overall ranking was Highlight >Toon >Crosshair >Circle >Phone. However, individual preferences seemed to vary - some preferring the Toon filter and crosshair due to not drawing too much attention on-screen, while some preferred the other UIs like the highlight since they better drew attention. Interestingly, participants mentioned for tests with no sound, simulating deafness, they preferred more visible and obvious demarcations, while for cases like mono hearing or smaller losses they preferred UIs that were more subtle.

### 4.2.2 Obstructiveness to the Rest of the FOV

Users were asked to rate the visualizations if they were focusing on something else while the effect is displayed. Figure 4 displays a bar chart visualizing the data from their survey. Note: For the phone it is stipulated you need to get it out every time you want to find a sound source.

Common User Comments: "For less hearing, I want more obstructive and visual indicators, but for closer to full hearing, something more subtle is better, since I won't always be using the visual information for what I'm doing."

From the perspective of designing a UI that is less obstructive and keeps immersion rather than becoming a distraction, it seems our cartoon filter was quite successful. While it was not the favorite in terms of intuitiveness in finding an object to compensate for severe hearing loss, users found the filter much less distracting when focusing on other tasks.

### 4.2.3 Quality of the Filter in Passthrough

Most users (63.5%) reported that they could still scroll, read, or perform other detail-oriented tasks while the toon filter was active, whereas 37.5% said they could not; of those who said "no," two-thirds (66%) attributed their difficulty not to the filter itself but to passthrough limitations (e.g., low resolution). When asked whether the visualization—ignoring inherent passthrough issues like resolution or framerate—was distracting or could be mentally filtered out over time, 75% of respondents felt they could adapt to or ignore it.

Despite this relative comfort with the filter, only 37.5% said they would be comfortable using the system for extended periods under the current Quest 3 passthrough's framerate and latency. Half of the participants noted motion sickness, vergence–accommodation conflict, or other VR-related discomfort when using passthrough for longer than thirty minutes. However, 87.5% of users indicated that if the same filter were implemented on higher-fidelity, low-latency AR glasses (rather than the Quest 3), they would feel far more comfortable using it over time. In other words, while the toon filter itself does not significantly degrade users' ability to perform tasks or adapt visually, the hardware's passthrough performance remains the primary barrier to prolonged use.

## 5. Discussion & Conclusion

The claims we were able to prove or demonstrate are as follows:

Because we did not have access to an ambisonic microphone, we created a virtual environment to perform our user study. Most users agreed our sound setup and ray tracing contributed to a realistic environment. In other words, we successfully overcame our technical challenge not having a directional microphone. Furthermore, users found our virtual room's construction and level of detail appropriate for this evaluation.

The claims we were able to prove or demonstrate are as follows:

- Passthrough is a more intuitive and helpful way of localizing sound than existing methods (like a phone or baseline) - our quantitative times from the time trial and survey supports this.
- The Toon filter is less obstructive than most of our reference designs. Notably, whether or not this is a good thing seems to depend on the user's preferences
- Our Toon filter works in stereo in passthrough, and allows users to perform their normal tasks. However, users report some discomfort, which is largely attributed to the headset, and not the filter.

The claims we learned more about and can now qualify or limit in scope: While the toon filter is broadly considered less obstructive, we can't definitively say that it is more intuitive than other UIs (simply that in point 3 above, all passthrough UIs show better results than reference/baseline). In fact many users did not like the toon filter if it was the primary method of guiding the user to sound. However, if the user had a high degree of hearing and specifically only used the Toon filter to help localize the sounds they could hear (mono hearing), the Toon filter became a favorite. In short, the Toon filter is not more intuitive than the others in general because it is more subtle, but if the system is augmenting good hearing or mono hearing, it is an intuitive method that works without cluttering up the screen.

As to limitations of our work, we acknowledge that despite our efforts, a virtual scene to evaluate sound localization is not as robust as a in-person real-life study. There are many factors that could change the quantitative data, such as familiarity with VR and its controls, difficulties interacting with the virtual world, and imperfect sound modeling. Qualitatively, the lighting in the room didn't vary much, the textures were not photo-real or very high resolution, and we didn't add any ambient or extra noise sources which could skew our participants' perception of the systems.

In the future, if we wanted to continue this study, it would best be done with an ambisonic microphone in passthrough only. By doing so, we can remove all the ambiguity in which factors could be influenced by the test environment being virtual and also have a better representation of the responsiveness and appearance of the UI with real life passthrough.

However, we find our work sufficient to conclude that passthrough has incredible potential to augment hearing and address accessibility gaps, and specifically that there is a place for designing filters like cartoon filters that can reduce the obstruction and distraction and lead to a more comfortable and immersive experience.

## References

[1] T. Asakura. Augmented-reality presentation of household sounds for deaf and hard-of-hearing people. *Sensors*, 23(17), 2023.

[2] M. Koshi, N. Sakata, and K. Kiyokawa. Augmented concentration: Concentration improvement by visual noise reduction with a video see-through hmd. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1030–1031, 2019.

[3] Z. Li, S. Connell, W. Dannels, and R. Peiris. Soundvizvr: Sound indicators for accessible sounds in virtual reality for deaf or hard-of-hearing users. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '22, New York, NY, USA, 2022. Association for Computing Machinery.

[4] Meta. Explore a new era of mixed reality with the passthrough camera api, Apr 2025.