Stereoscopic Untethered Video See-Through Augmented Reality

Khoi Le
Stanford University
School of Engineering
ihustle@stanford.edu

Kyle Qian
Stanford University
School of Engineering
kyleqian@stanford.edu

Abstract

Modern augmented reality systems strive to seamlessly blend virtual, computer-generated objects with the real world. However, many current systems have flaws such as limited field of view, not being usable outside, being tethered to a computer, or only having monoscopic vision. We present SUVSTAR: Stereoscopic Untethered Video See-Through Augmented Reality as a potential manifestation of augmented reality headsets in the future.

1. Introduction

Augmented reality (AR) is a new display technology that seeks to integrate digital objects and overlays with the real world, taking advantage of contextual cues to insert relevant objects in spatially coherent locations within a user's field of view (FOV).

Many of today's AR displays, such as the HoloLens and Meta, display images to your eye through a transparent plate, either through reflection or waveguides. This paper will refer to these systems as "optical see-through AR." Optical see-through (OST) AR faces many challenges, such as limited field of view (FOV), indoor usage only, and no fully opaque blacks. This project proposes the use of "stereoscopic untethered video see-through AR" (SUVSTAR): displaying two different camera feeds to a stereoscopic display, which has been implemented in several systems such as: combining a Zed camera with an Oculus Rift, building custom hardware [4] or using standalone camera feeds through webcams [3]. However, the Oculus Rift is tethered to a computer, preventing users from using AR in the real world, where AR can provide immense contextual value. While untethered video see-through (VST) AR systems exist in the Vive Focus [7], Google Lenovo Mirage [6], and Oculus Quest, these camera feeds do not receive color data due to computational efficiency choices for virtual reality (VR). This project seeks to build a standalone stereoscopic untethered pass-through augmented reality system (SUVSTAR) with color.

2. Motivation

Current augmented reality displays face a host of challenges that can be mitigated by stereoscopic untethered video see-through augmented reality systems. The advantages of SUVSTAR systems are outlined below.

1. Wider field of view
2. Outdoor use
3. Dark color occlusion
4. Altering perception

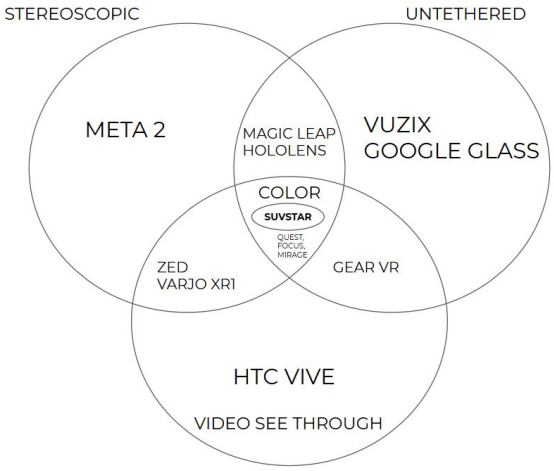

Figure 1. Comparison

2.1. Field of View

The HoloLens 2.0, the cutting edge of $MixedReality^{TM}$ headsets, has a vertical FOV of 29 degrees and horizontal FOV of 43 degrees, for a 52 degree diagonal FOV [5]. The human visual system has a FOV of 200 degrees, so the HoloLens FOV is much lower than that of the human eye. Since pass-through augmented reality systems work similarly to virtual reality headsets with LCD screens, they can achieve similar FOVs to systems like the Vive and Oculus Rift; around 110 degrees. This is more than double that of the HoloLens 2.0.

2.2. Outdoor Use

Although current state-of-the-art AR headsets are untethered, their see-through displays do not work well outdoors. This is because the nature of see-through displays is such that light must be let through behind the digital content. SUVSTAR, on the other hand, would not have this problem as all external light is first captured via a camera and then rendered onto a display. From there, digital content will completely occlude real life content, and the display will take up the user's entire FOV, exactly like a VR headset. SUVSTAR would thus be able to take full advantage of its being untethered, rather than being effectively limited to indoor spaces with friendly lighting conditions.

2.3. Dark Color Transparency (No Fully Opaque Blacks)

Because see-through AR systems can only shoot light into the eye, the displayable color spectrum is anchored to the real world's color and can only be additively augmented. This means that see-through AR systems cannot render full opaque blacks [13]. Dark colored digital mixed reality objects that should occlude physical objects will appear transparent and not fully occlude the real world.

2.4. Altering Perception

See-through AR can only add things to the world. However, pass-through AR receives all image data from the world, allowing the system to manipulate it before displaying it to the user. This allows developers to alter visual perception in (near) real time. For example, a user could see the real world in inverted colors for a novel perceptual experience. Novel manipulation of social situations would also be possible, such as "control[ling] ... interpersonal distance by changing the size of [other people] in the HMD view" [9].

3. Related Work

Work related to our SUVSTAR system involves current video see-through AR systems, real-time stereo camera systems, and visual perception experiments.

3.1. Current Video See-Through Systems

There exist several consumer standalone VR headsets that have "see-through" modes. As mentioned above, the HTC Vive Focus, Google Lenovo Mirage Solo, and Oculus Quest all have "see-through" modes. These "see-through" modes are implemented through multiple front-mounted cameras. However, these cameras do not capture color, or at least the color data is not streamed to the display. Most likely, forgo-

ing color capabilities makes the camera components cheaper. Similarly, black and white is much cheaper to stream at low latency. With SUVSTAR, we will be forgoing these cost cuts in order to explore the potential of color stereoscopy.

3.2. Real-Time Stereo Camera Systems

Work has been done with camera-based VR as an attempt to replicate video see-through AR. One solution is to use a ready-made stereo camera, such as the Zed, attach it to a PC VR headset, such as the Oculus Rift, and stream the camera feeds into each eye [10]. Some PC VR headsets have built-in cameras, such as the HTC Vive. However, relying on a PC VR headset tethers you to the computer, and you cannot use it outside unless you wear a laptop backpack. Another option is to use webcams, two of them mounted like a Zed, to capture stereo images which are streamed to each eye [3]. However, most webcams are built to stream data through USB to a computer; and most are fairly large and clunky. Other options are to use two cameras with two remote transmitters [14]. However, the analog transmission implemented by Wu, though low-latency, results in lower resolution and visual artifacts.

3.3. Visual Perception Experiments

Finally, the types of simple experiences possible with SUVSTAR would mirror related work in psychology. Interesting work has been done in the field of altering visual inputs in social situations, such as seeing enhanced smiles on virtual avatars, which improves mood [11]. Similarly, one could also control the size of other people in the view [9].

4. Implementation Details

4.1. Hardware

The SUVSTAR system is based off of the Samsung GearVR pass-through mode, with a phone in a VR housing with an exposed back so the camera has a clear view of the world. The phone's rear camera is used to capture image data, which is then streamed to each eye in a barrel distorted form seen through lenses. In order to achieve stereoscopic vision, a second phone was used rather than a second camera. We chose to use a second phone for simplicity.

To build the system, two Samsung Galaxy S9 phones were used with a modified plastic VR housing made for cell phone VR use. The Samsung Galaxy S9's were selected because of the central location of the back-facing camera. This minimizes the spatial offset of the camera perspective from the center of the lens where the image is displayed to the eye. The VR housing was selected for the presence of a small ridge at the bottom of the headset, so the phones could be supported from the bottom. An acrylic brace was laid across the front of the headset to keep the phones in place.

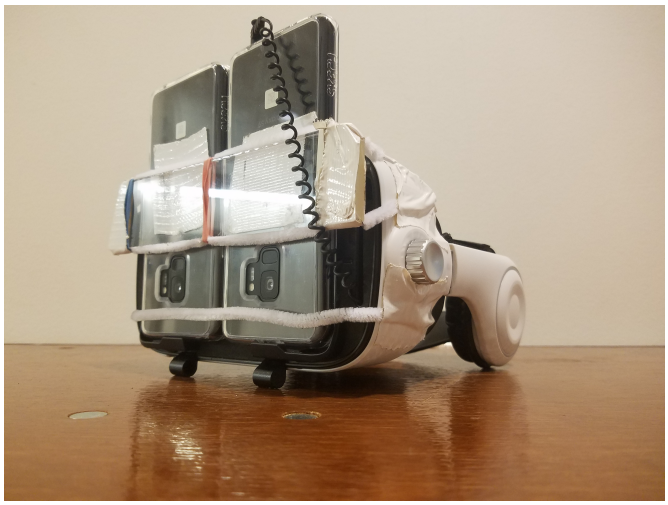

Figure 2. SUVSTAR Physical Housing with Phones Inserted

4.2. Rendering Video See Through

The phones were loaded with software that we built in Unity 2019.1.4f1. We used the Google ARCore SDK (v1.9) to display virtual objects anchored in space with the camera feed of the real world in the background. This view was written to a Unity RenderTexture and then displayed on a

quad. Another camera views this quad, but from a distance, simulating a virtual reprojection to compensate for the spatial offset of the cameras. The offset distance was gauged by calibrating the virtual image seen through the lenses to match the size of real objects seen with the naked eye. This

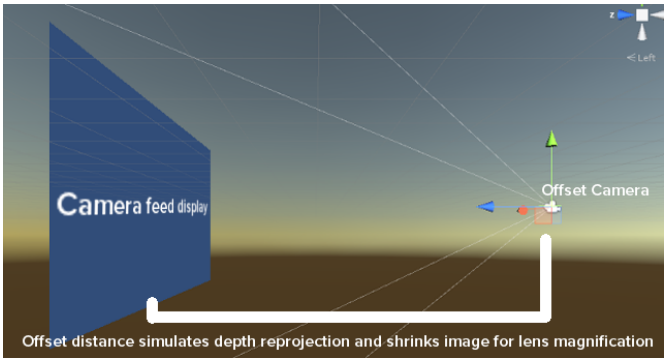

Figure 3. Camera with offset

camera writes to another RenderTexture, which our post-rendering script barrel distorts and displays on the final screen shown to the eye through the lens. Our post-rendering script is based on the Unity Stereo SDK provided by EE267 staff, which is based on the Cardboard v1 SDK.

4.3. Image Recognition and Tracking

In a siloed software system, where each phone is running independently, it is difficult to internally sync the systems and have virtual objects correctly line up. Thus, we used an external anchor that both phones could reference: an image. To showcase useful augmented reality content, we developed five potential use cases where AR could provide useful information overlaid on the real world. These were: live translation of a foreign language, seeing notifications of messages from friends, playing audio and providing context for images, displaying details of an artwork, and seeing reviews of a restaurant or product. For each of these use cases, we used a 2D image printed on paper to represent the situation. Each of the 2D images was registered in an ARCore AugmentedImages Database. ARCore automatically can recognize and track images registered to an AugmentedImages Database, so we simply created the virtual content matching each image.

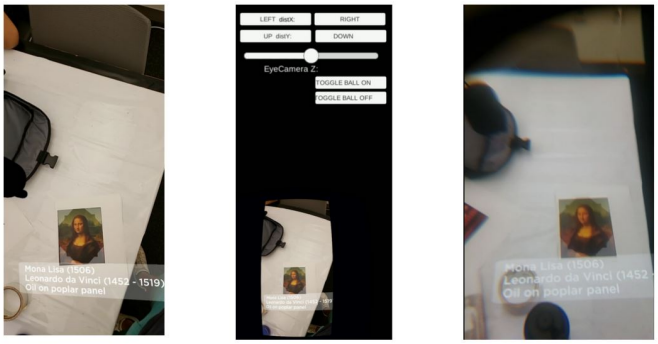

Figure 4. Image Tracking of the Mona Lisa: ARCore Feed to Distortion to View through Lens

4.4. 3DOF HUD Display

We created a 3 degree-of-freedom (DOF) heads-up display (HUD) to show the current time and upcoming appointments, similar to the functionality given by a smartwatch. In order to sync the positions across the phones internally with severe positional tracking jitter in ARCore, the HUD is locked to the position of the tracked camera device. Rotational tracking in ARCore is much more accurate and reliable across long sessions. We lock the rotation of the HUD so that its relative position and rotation from the camera is standard across the two phones. Then, we offset the HUD for each eye to give accurate stereo effects. Because the phones start in the same orientation in the headsets, the HUD's relative virtual position is the same across devices; simulating real-world anchoring in 3DOF space.

5. Experiment

We wanted to see how important stereoscopic vision was in perofrming hand-eye coordination tasks. We devised a hand-eye coordination task to compare performance in our system under mono-

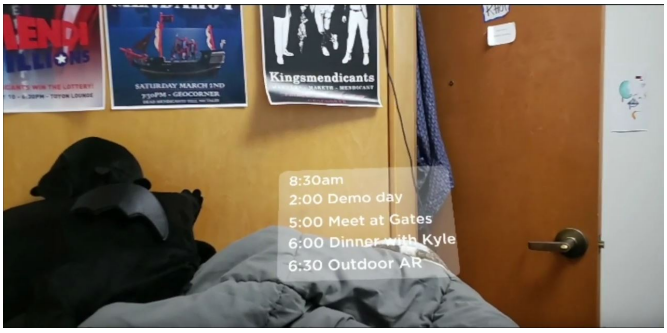

Figure 5. Heads-Up Display Showing Schedule

scopic and stereoscopic conditions. Our task consisted of nine cups randomly arranged on a table with nine plastic forks in front of the participant (fig. 7). Participants were asked to put one fork in each cup as quickly as they could while being timed. There were no restrictions on placement style, which should be implemented in the future. We ran the informal experiment on 15 participants across three conditions: no SUVSTAR glasses, stereoscopic SUVSTAR glasses, and monoscopic SUVSTAR glasses. Time was recorded using a stopwatch on a Samsung Galaxy S7, starting upon completion of saying "Ready, set, go!" and ending when the last fork touched the bottom of the last cup. Participants in the monoscopic condition

Figure 6. Hand-Eye Coordination Task

were asked which of their eyes felt stronger, and the phone in the other eye was turned off, showing black. Participants in the stereoscopic condition were shown both images from both phones. We wanted to see if the stereo image from both eyes would improve time of completion. The arrangement of cups was randomized so that users could not use relative spacing to their advantage; they had to rely on their vision. The cups were Handy Paint Cups with a diameter of 6 inches arranged within arm's length radius of the participant. The forks were lined up on the table in front of the participant.

6. Results

In 30 informal trials with 15 participants across three conditions, stereoscopic SUVSTAR was 25% faster than monoscopic SUVSTAR, but still 3 times slower than no glasses. The median time-of-completion for participants in the monoscopic condition was 17.9 seconds, compared to 13.71 seconds for participants with stereoscopic SUVSTAR. As a baseline, the median time-of-completion for participants without SUVSTAR glasses was 4.615 seconds.

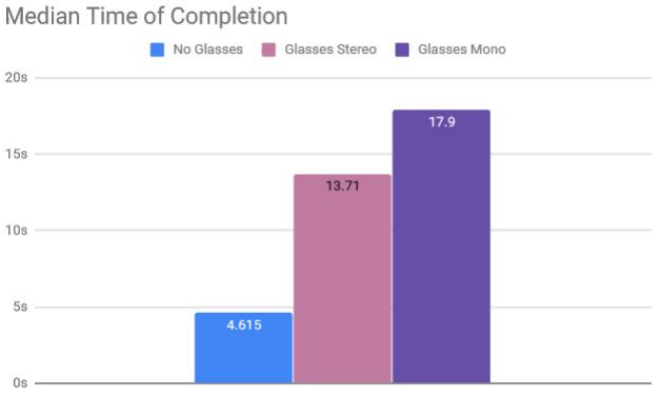

Figure 7. Informal Experiment Results

7. Discussion

Based on the results of the informal experiment, stereoscopic vision is important to the completion of hand-eye coordination tasks involving depth and proprioception such as using your

hands to place objects in cups at varying distances. Many participants reported struggling with the latency, often overshooting the cups because their visual system would not show their hands above the cups until later, making them believe they hadn't reached it yet and causing them to overextend. Moreover, the position of the cups would move with a delay in their frame of reference as they looked around. The latency for the cup and hand positions combined made it very difficult for a lot of participants to move quickly. Several participants that did exceptionally well discovered that holding their heads still would prevent the cups from moving, which helped them quickly adapt to the latency, since they only had to adjust for the visually altered movement speed of their hands. While participants without glasses outperformed participants wearing glasses in this task, there was no true augmentation or digital display occurring in the SUVSTAR headset in these trials. We foresee tasks where having digital information displayed to you in a headset would compensate for the slower hand-eye coordination. These tasks include: assembling furniture with a digital handbook vs. a physical one, sorting warehouse packages with digital overlays vs. reading the physical labels, and navigating a city with an AR map vs. with a 2D phone map.

8. Future Work

8.1. Future Experiments

There are several more experiments that we could explore to validate core advantages of SUVSTAR. We believe that color is key to the experience of augmented reality, and we would like to explore the role that color plays in certain tasks such as warehouse sorting, assembling parts or operating switchboards. We would also like to compare performance on certain hand-eye coordination tasks between SUVSTAR stereo and similar colorless stereo headsets like the Oculus Quest. Finally, we'd like to compare SUVSTAR stereo to similar colorless stereo headsets like the Oculus Quest on resolution-based tasks such as reading text.

8.2. Latency

The biggest obstacle for SUVSTAR is latency. Currently, the system inefficiently bounces back and forth between the CPU and GPU. The system has to retrieve the image from the GPU, display it and write it to a RenderTexture, display it again to the offset camera and write that image again, then distorting it on the CPU and displaying it again. We hope to explore improving the graphics rendering pipeline by moving more of the processing into the GPU as well as pushing some computational efforts into the edge cloud [15]. In a custom-built embedded system, more hefty processors could be used.

8.3. Depth Reprojection

Reprojection is using data from a camera feed and re-rendering it as if the camera is in a different position. Currently, SUVSTAR estimates a virtual reprojection by moving the camera backwards away from the camera feed. This makes the image appear approximately the same size as it would be in real life. However, this method results in a lower effective FOV, as moving the camera back shrinks the image size in the viewport. No data is collected about the outside environment outside of the initial camera feed. Given more cameras and depth sensors, a complete image and depth map necessary for reprojection can be made of a scene.

8.4. Additional Features

There are a few additional features that would be powerful improvements to the utility of the SUVSTAR system. The most important would be some form of input. Current SDKs and APIs exist for both hand-gesture tracking and voice commands.

8.4.1 Hand Tracking

Hand tracking is a natural and intuitive form of interaction. Design principles are key to avoid "gorilla arm," a type of fatigue that occurs when interacting with hand-based user interfaces at upper arm level [2]. Implementing hand tracking would afford interaction with virtual objects, such as tapping the schedule to pull up additional information. Existing work has been developed for augmented reality systems utilizing Leap Motion for hand tracking [8]. Other systems exist for finger tracking [1] and hand tracking [12] using the single RGB camera feed.

8.4.2 Voice Commands

Voice input is a natural and simple way for users to express contextual commands by speaking. Voice can be difficult to execute in loud areas, since the excess noise will disturb the algorithms accuracy. Users may be averse to using voice, especially if they are sensitive to speaking to their device in social situations with others around. Google Speech, IBM Watson, and Microsoft Speech are all powerful natural language processing (NLP) APIs that can enable voice commands in the future.

8.4.3 Eye Tracking

Another potential route for input is eye-tracking, where the system knows where the user is looking and can react accordingly. The current SUVSTAR system has head-gaze-based tracking, where looking at the HUD expands the schedule. These types of interactions can be augmented by the introduction of eye-tracking. Companies like Tobii and 7invensun have existing eye-tracking solutions that can work in an HMD.

8.5. Challenges

Several challenges were faced in the creation of this project. First, the ARCore image is cropped and correctly mapped to the resolution and aspect ratio of the hardware device. It is also slightly zoomed in so that the image appears more seamlessly blended into the world. This makes it difficult to barrel distort and show through a lens, since it makes everything look very big and close once magnified. Getting a larger camera feed is possible through putting a Unity WebCamTexture on a large quad at the far clipping plane of the camera, but the latency on this proves to be much slower than ARCore's camera feed. We also tried to access the ARCore raw texture, which was not well documented and gave us a large red square texture when we tried. Another approach was to get the raw image bytes from the GPU and modify them on the CPU, but this proved to have significant overhead, since the bytes are saved in YUV, and converting each pixel to RGB proved to be inefficient on the CPU. Moreover, the YUV is saved as a flipped and rotated image, and a re-mapping to the correct orientation would also be expensive on the CPU.

Next, because the image is cropped and shaped to the phone, the image becomes a vertical portrait when the phone is oriented in portrait upside-down mode. This means the horizontal FOV is lowered in the hardware configuration that we chose. This can be overcome by turning the phone sideways, which we elected not to do because of ridges on our plastic housing were in the way. We attempted a software fix by rotating the image received by the feed, but then it appears rotated in the headset as well.

Finally, the Cardboard SDK plugins are incompatible with the ARCore plugins. Building a version to the phones with both plugins installed lead to crashes on launch. Moreover, the Cardboard SDK prevents any application from using Unity UI elements, which we needed to configure our camera parameters. Thus, we had to strip the project of the Cardboard plugins and adapt the Cardboard code to our needs without the availability of plugins.

9. Conclusion

For the future, video see-through augmented reality is gated by spatial, due to the camera position, and temporal, due to the latency, offsets. Both of these issues can be addressed using techniques outlined above such as reprojection to overcome spatial offsets and improving processing time to overcome temporal offsets. These solutions are compute-heavy, so attaching a computer can help. Already, companies like Varjo have excellent video see-through AR while tethered to a computer. We believe that more formal studies should be conducted comparing video see-through and optical see-through augmented reality systems in various tasks.

References

[1]

[2] J. H. aand C. Peng et. al. Dispelling the gorilla arm syndrome: The viability of prolonged gesture interactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 505–520, 07 2017.

[3] J. Becker and T. Ngo. Mitigating visually-induced motion sickness in virtual reality. *EE267: Virtual Reality*, 2016.

[4] V. Ferrari, F. Cutolo, and E. M. Calabr. Hmd video see though ar with unfixed cameras vergence. *IEEE International Symposium on Mixed and Augmented Reality 2014*, 2014.

[5] L. Goode. Microsofts hololens 2 puts a full-fledged computer on your face. *Wired*, 2019. https://www.wired.com/story/microsoft-hololens-2-headset/.

[6] D. Heaney. Lenovo mirage solo update adds camera passthrough to the daydream standalone. *UploadVR*, 2019. https://uploadvr.com/mirage-solo-passthrough/.

[7] B. Lang. Vive focus 2.0 update brings bevy of new features: Phone mirroring, pass-through video more. *Road to VR*, 2018. https://www.roadtovr.com/vive-focus-system-update-2-0-phone-mirroring-pass-through-video-and-more/.

[8] H. Le and J. Kim. An augmented reality application with hand gestures for learning 3d geometry. pages 34–41, 02 2017.

[9] M. Maeda and N. Sakata. Maintaining appropriate interpersonal distance using virtual body size. *2015 IEEE International Symposium on Mixed and Augmented Reality*, 2015.

[10] J. Meng. v.os. https://www.youtube.com/watch?v=oBQ3uweB2-Yfeature=youtu.be.

[11] S. Oh and J. Bailenson. Let the avatar brighten your smile: Effects of enhancing facial expressions in virtual environments. *PLoS ONE 11(9)*, 2016. https://vhil.stanford.edu/pubs/2016/brighten-your-smile/.

[12] T. Palladino. Manomotion updates its hand tracking sdk for smartphones with skeleton tracking. https://mobile-ar.reality.news/news/manomotion-updates-its-hand-tracking-sdk-for-smartphones-with-skeleton-tracking-0185034/.

[13] L. Rizzotto. Ar glasses, step back: Vr/ar hybrids will lead mainstream ar, 2019. https://medium.com/futurepi/ar-glasses-step-back-vr-hybrids-will-be-the-first-mainstream-form-of-ar-e28ba2528c72.

[14] E. Wu. Fly it like you mean it mitigating motion sickness in first-person-view drones. *EE267: Virtual Reality*, 2017.

[15] W. Zhang, B. Han, and P. Hui. Jaguar: Low latency mobile augmented reality with flexible tracking. *Proceedings of ACM Multimedia*, 2018.