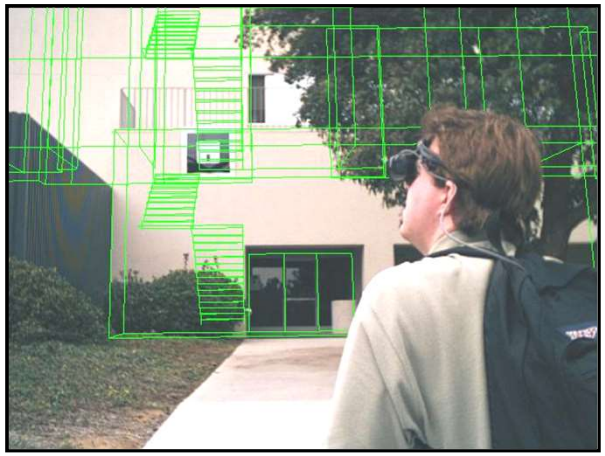
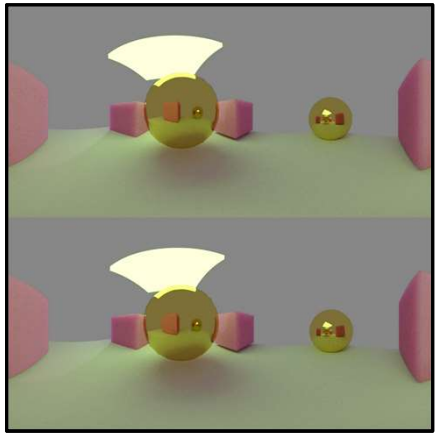
Virtual and Augmented Reality



Oregon State University

Mike Bailey

mjb@cs.oregonstate.edu



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

Oregon State University
Computer Graphics

"Reality, what a concept!"
-- Robin Williams

Virtual Reality Definition

Virtual reality (**VR**) is a simulated experience that can be similar to or completely different from the real world. Applications of virtual reality can include entertainment (i.e. video games) and educational purposes (i.e. medical or military training). Other, distinct types of VR style technology include augmented reality and mixed reality, sometimes referred to as extended reality or XR.

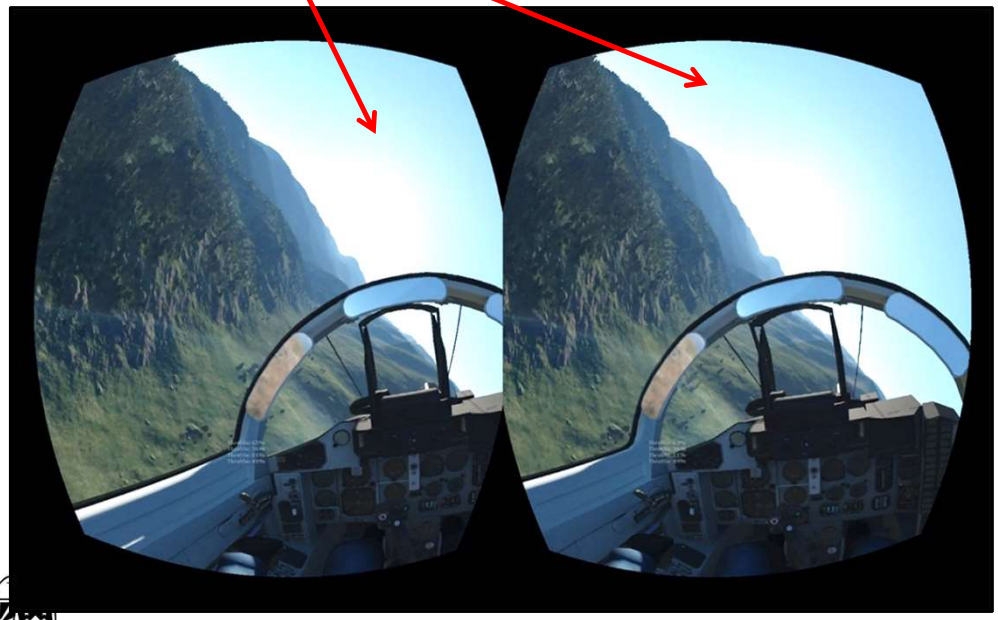
Currently standard virtual reality systems use either virtual reality headsets or multi-projected environments to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual environment. A person using virtual reality equipment is able to look around the artificial world, move around in it, and interact with virtual features or items. The effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes, but can also be created through specially designed rooms with multiple large screens. Virtual reality typically incorporates auditory and video feedback, but may also allow other types of sensory and force feedback through haptic technology.

https://en.wikipedia.org/wiki/Virtual_reality

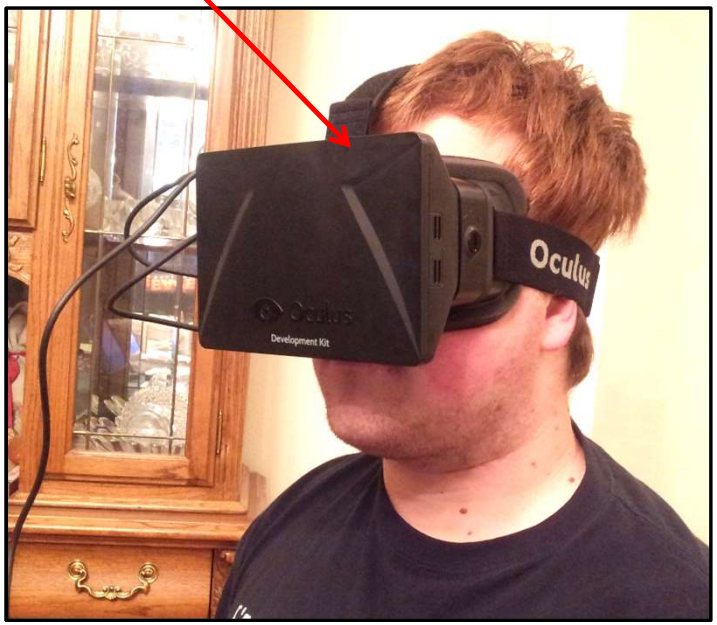
VR Headsets

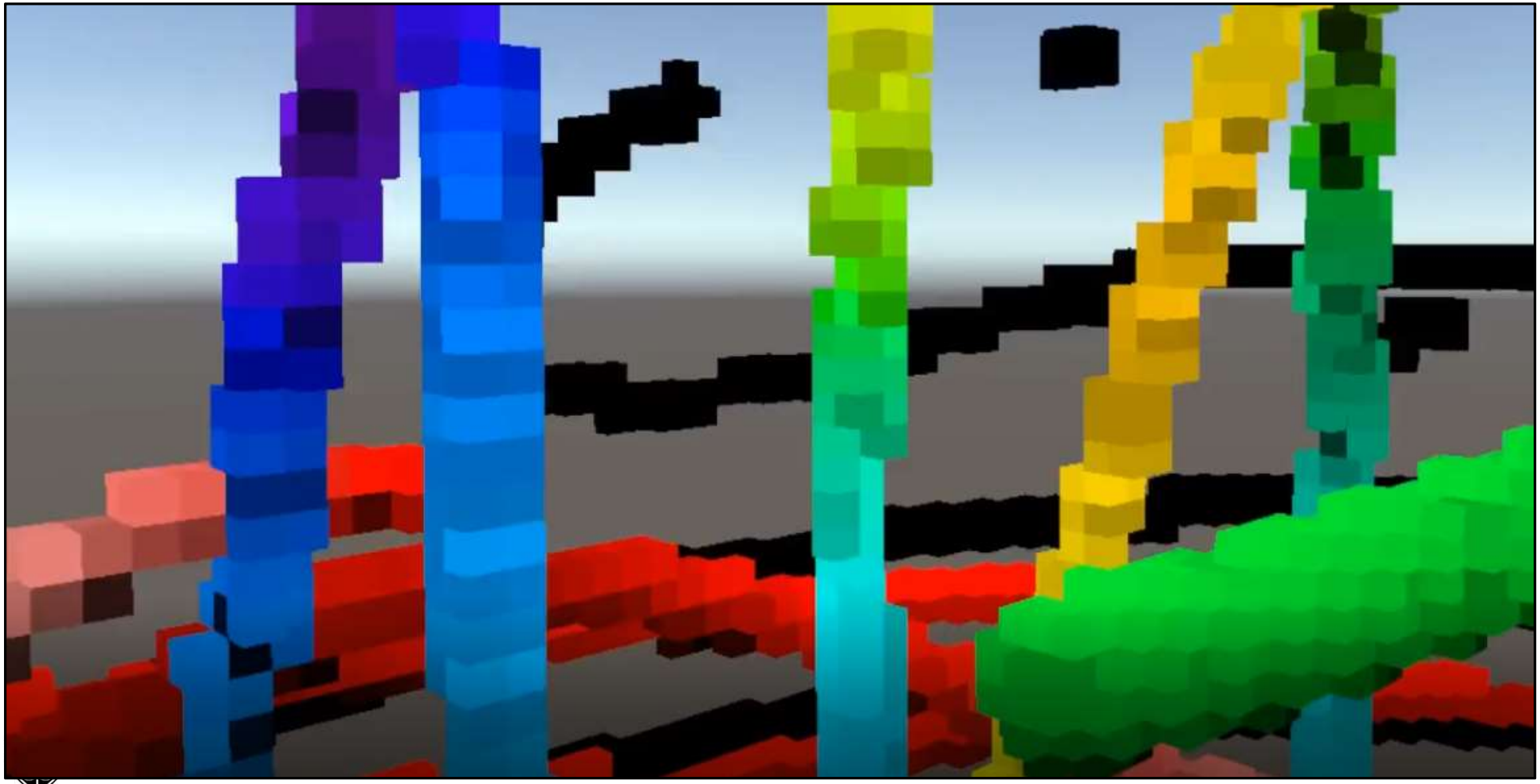
Uses shaders to get the correct non-linear fisheye lens distortion

Uses an accelerometer and a gyroscope to determine the head position and the head orientation



http://theriftarcade.com





PolyVox, a VR paint program
OSU CS Capstone Project: Richard Cunard, Braxton Cuneo, Chris Bakkom

Inexpensive VR Viewers for your Cell phone

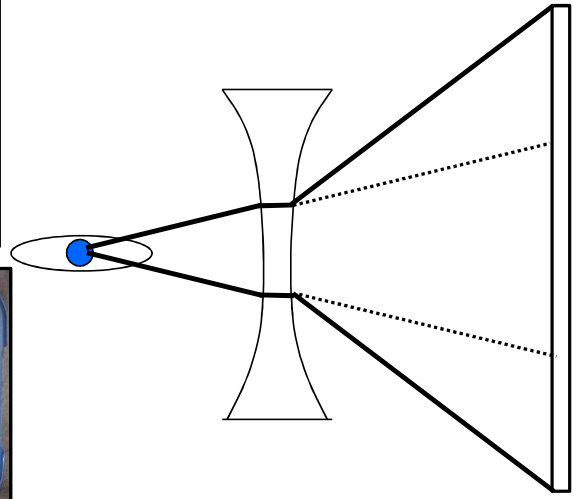
Uses OpenGL-ES shaders to get the correct non-linear fisheye lens distortion

Uses your phone's gyroscope to know the head orientation

Aquarium game, SideKick

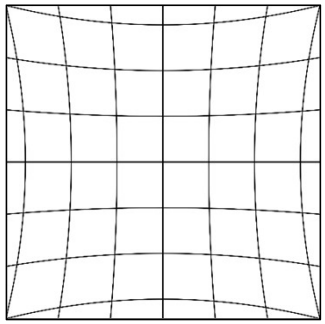
Uses a moving magnet and the phone's digital compass to perform a "left-click"

Correcting Lens Distortion with Shader Un-distortion



VR/AR goggles use **fisheye lenses** so that your eye can be close to the view screen but still see the whole scene.

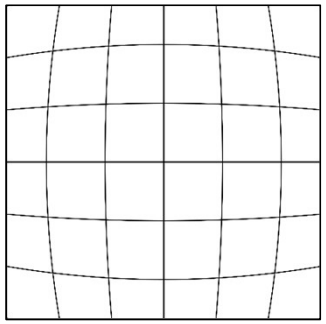
If you displayed a normal CG scene through such a lens, a grid would come out looking like this:



This is known as a **pincushion**.

https://en.wikipedia.org/wiki/Distortion_(optics)

The trick then, is to *distort* the CG image like this so that the lens's pincushion effect will make it look normal:



This is known as a **barrel**.

Correcting Lens Distortion with Shader Un-distortion

Pass #2 vertex shader

```
#version 330 compatibility
out vec2 vST;

void
main( )
{
        vST = gl_MultiTexCoords0.st;
        gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

Pass #2 fragment shader

```
#version 330 compatibility
uniform sampler2D renderedImage;
in vec2 vST;
const float ALPHA = 0.20;

void
main( )
{
        vec2 xy = 2.*vST – 1.;          // convert [0.,1.] to [-1.,1.]
        xy /= (1. – ALPHA*length(xy) );
        vec2 st = ( xy + 1. ) / 2.;     // convert [-1.,1.] to [0.,1.]
        gl_FragColor = vec4(  texture( renderedImage, st ).rgb, 1.  );
}
```

Use a two-pass rendering algorithm:

Pass #1: Render the image into a texture (not shown here)

Pass #2: Draw a quadrilateral. Normally you would just lookup the rendered texture and map it to the quadrilateral. In this case, we apply a distortion as part of the texture lookup. These are the Pass #2 shaders for drawing the quadrilateral.

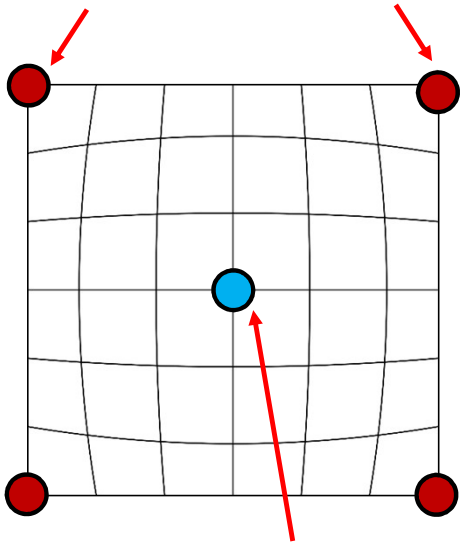
There are many different distortion functions you could use. This is one of the simplest.

$$Distortion\ Scale\ Factor(xy) = \frac{1}{1 - \alpha * \|(xy)\|}$$

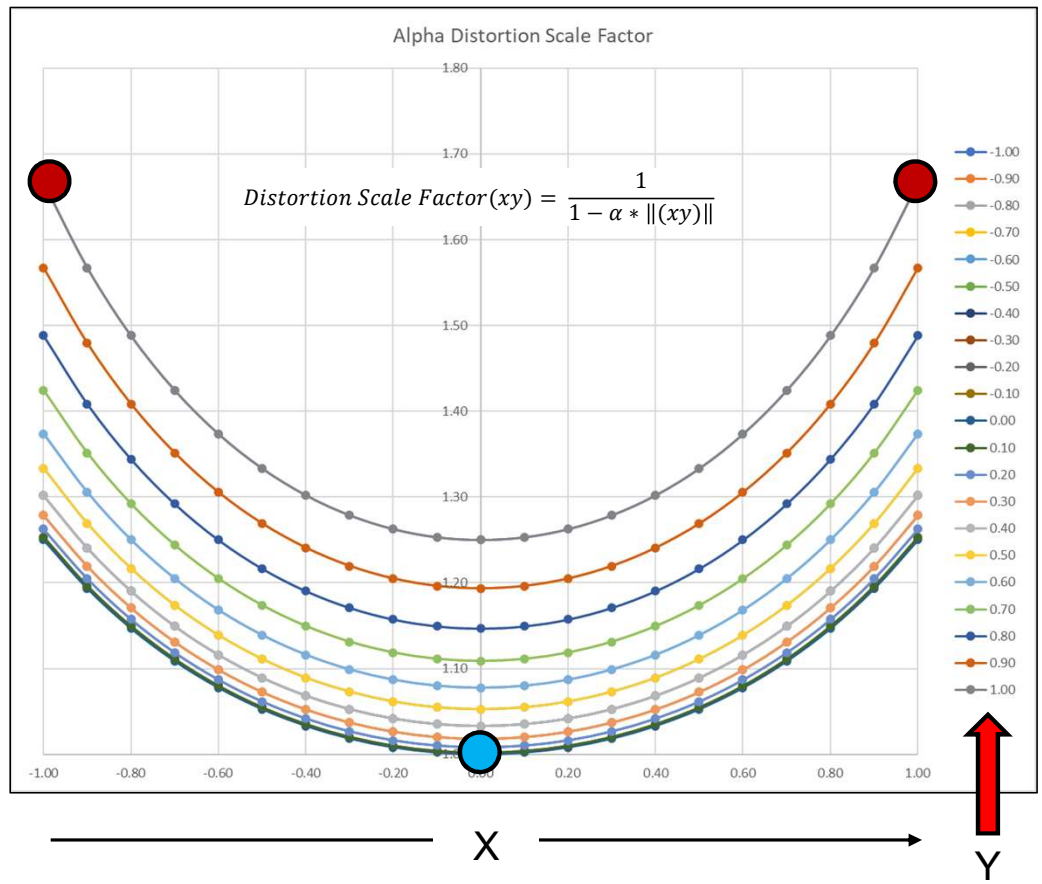
Correcting Lens Distortion with Shader Un-distortion

$$\alpha = 0.20$$

X=±1, Y=±1: most distortion



X=0, Y=0: least distortion

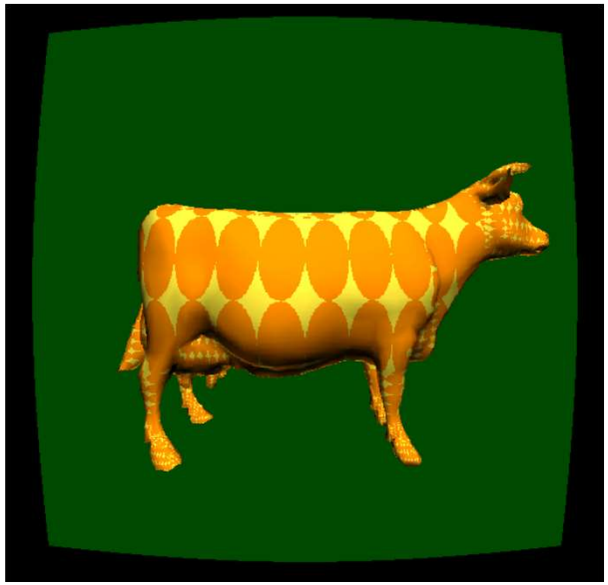


Oregon State University
Computer Graphics

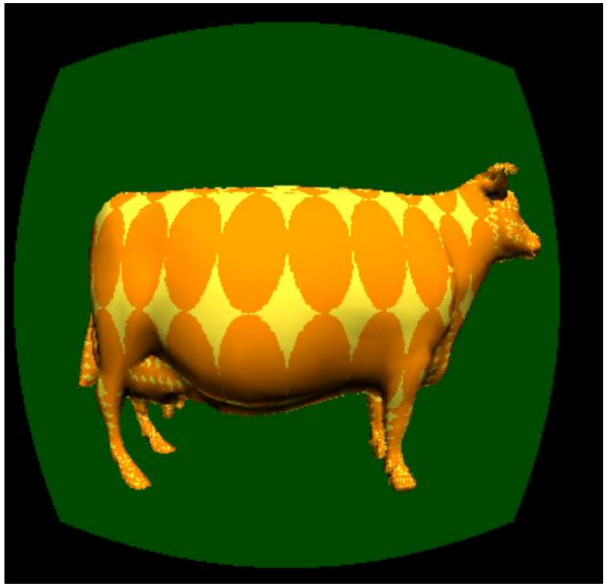
Correcting Lens Distortion with Shader Un-distortion

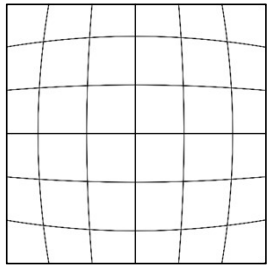
$\alpha = 0.00$

$\alpha = 0.20$



$\alpha = 1.00$

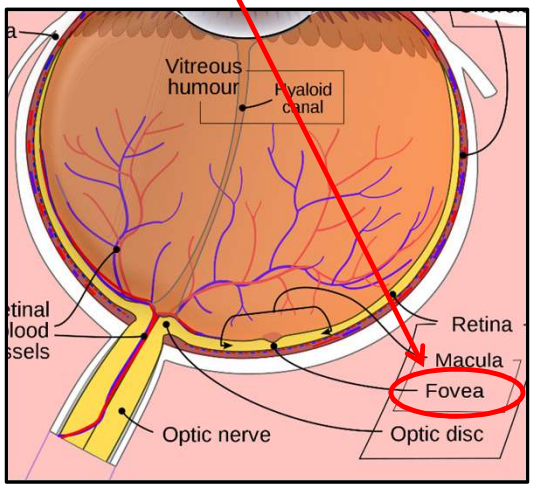




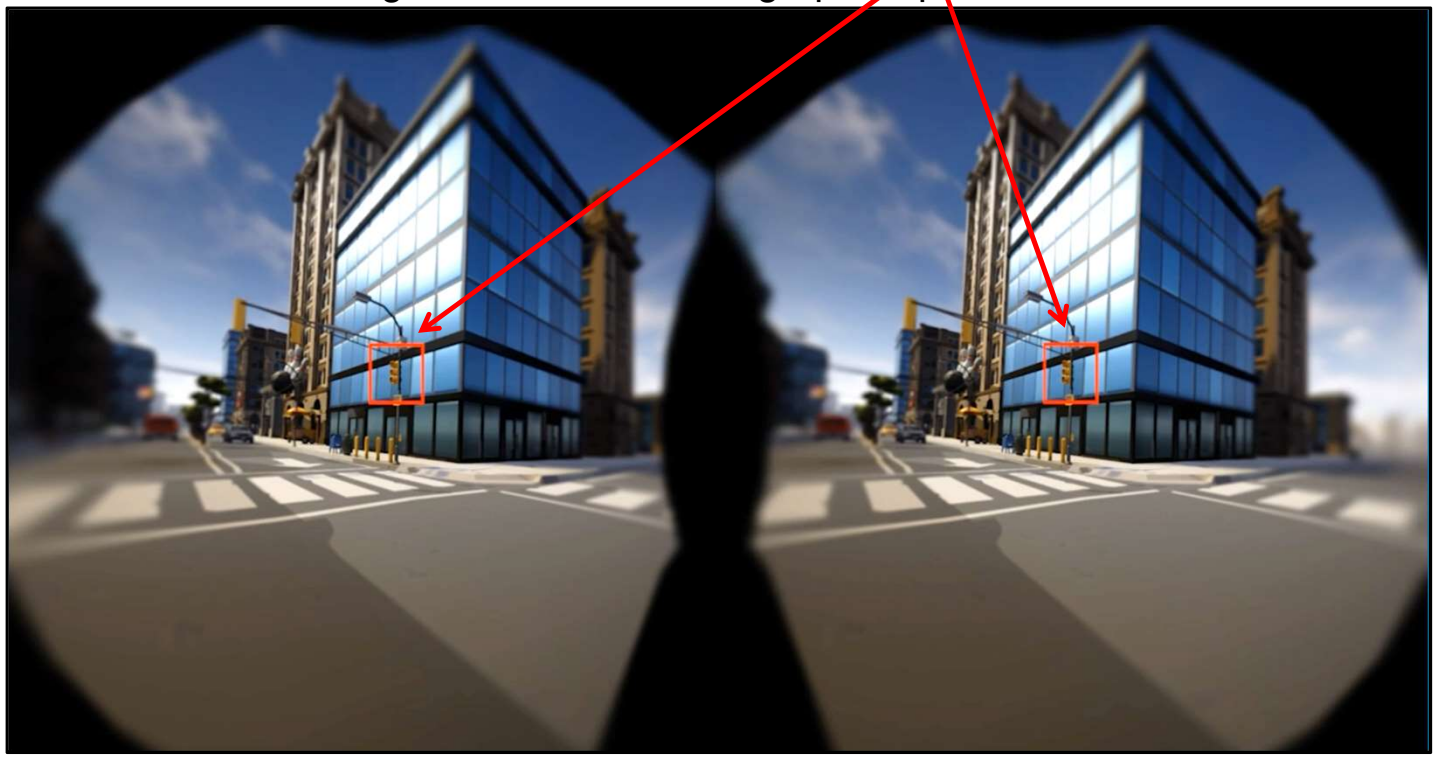
Another Cool VR/AR Rendering Trick – Foveated Rendering

The *fovea* on your retina corresponds to the projection of where you are looking. Of your entire ~220° field-of-view, only the ~2° of the scene projected onto your fovea is really perceived as crisp.

Cool CG trick: track the eye and only render at full resolution the part of the scene where the eye is looking. This saves rendering time and increases graphics performance.



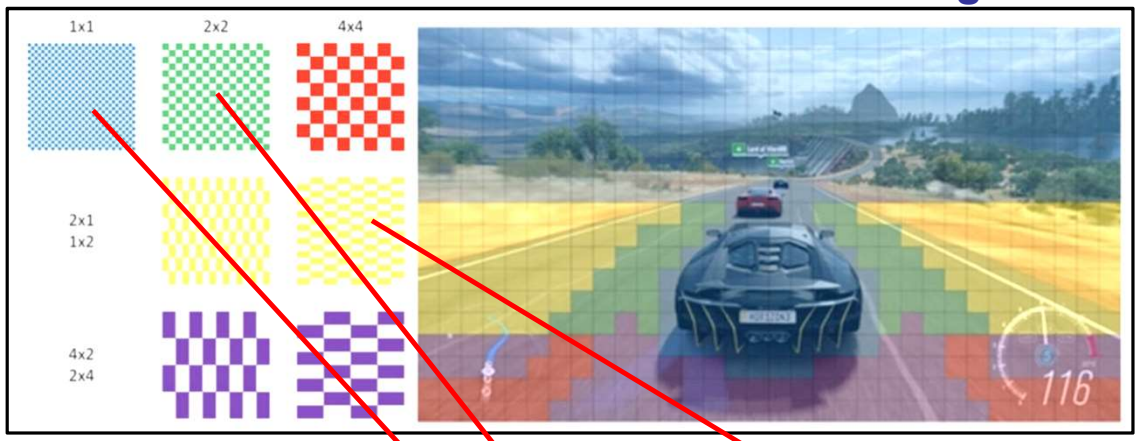
https://en.wikipedia.org/wiki/Fovea_centralis



Oregon State University
Computer Graphics

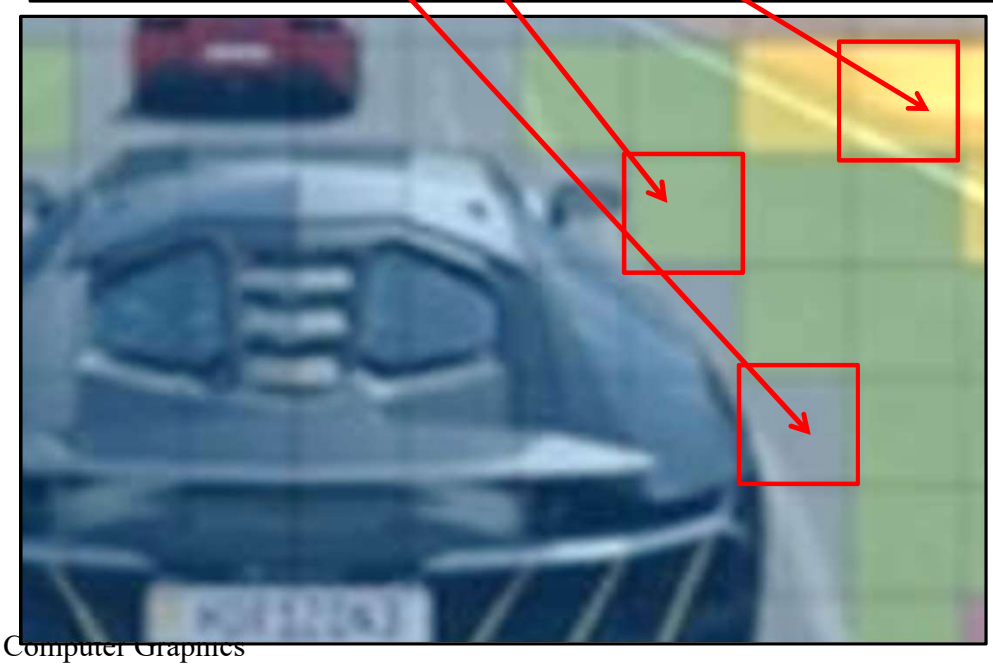
tobii.com

Another Cool VR/AR Rendering Trick – Variable Rate Rendering



By default, there is one fragment shader call per pixel being viewed. The Vulkan API takes foveated rendering and adds more flexibility to it.

Vulkan has a mechanism to enable your program to ask for a fragment shader call to cover more than one pixel. That is, multiple pixels can share the color result from one fragment shader call. This saves rendering time in less important parts of the scene.



Variable Rate Shading can be controlled by specifying parameters for:

1. The entire scene, or,
2. An entire object that is currently being drawn, or,
3. Arbitrary parts of the scene (such as where the eye is looking)

To specify VRS for arbitrary parts of the scene, you create an additional frame buffer and use it to store “flags” in each of its pixels to tell the shading mechanism how many pixels will share the results of a call to that fragment shader.

Inexpensive VR Viewers for your Cell phone

View-Master Virtual Reality

This is the Mattel **View-Master Deluxe VR Viewer.**
It sells for under $25. I trust View-Master to get the mechanical design and the optics right. They've been doing this successfully for decades.

BNEXT Virtual Reality

This is the **BNEXT VR Headset.**
It sells for under $30. This one is nice because it attaches to your head so that your hands are free. It also has a couple of nice eye-viewing adjustment knobs.

I found both of these on Amazon

Surround-VR: The CAVE

A **Cave Automatic Virtual Environment** (better known by the recursive acronym **CAVE**) is an immersive virtual reality environment where projectors are directed to between three and six of the walls of a room-sized cube.

https://en.wikipedia.org/wiki/Cave_automatic_virtual_environment

https://www.mechdyne.com

Augmented Reality Definition

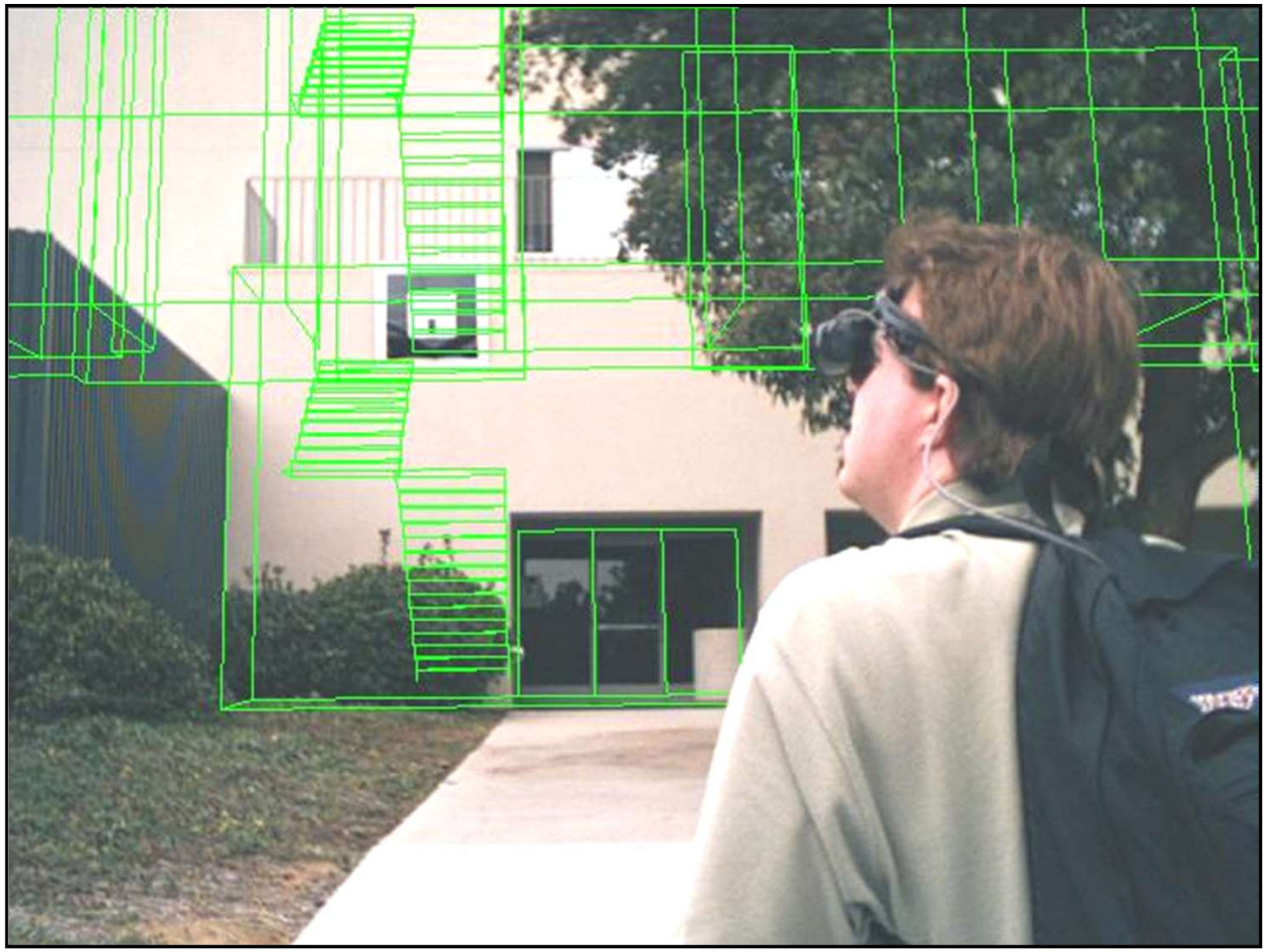
Augmented reality (**AR**) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory. AR can be defined as a system that fulfills three basic features: a combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects. The overlaid sensory information can be constructive (i.e. additive to the natural environment), or destructive (i.e. masking of the natural environment).This experience is seamlessly interwoven with the physical world such that it is perceived as an immersive aspect of the real environment. In this way, augmented reality alters one's ongoing perception of a real-world environment, whereas virtual reality completely replaces the user's real-world environment with a simulated one.

https://en.wikipedia.org/wiki/Augmented_reality

Augmented Reality -- Pokémon Go

Pokemon.com

Architectural Augmented Reality



Matt Clothier

High-end Augmented Reality -- Microsoft Hololens 2

https://www.wired.com/story/microsoft-hololens-2-headset/

Microsoft Hololens 2

The lasers in the HoloLens 2 shine into a set of mirrors that oscillate as quickly as 54,000 times per second so the reflected light can paint a display. Those two pieces together form the basis of a microelectromechanical system (MEMS) display. That's all tricky to make, but the really tricky part for a MEMS display is getting the image that it paints into your eyeball.

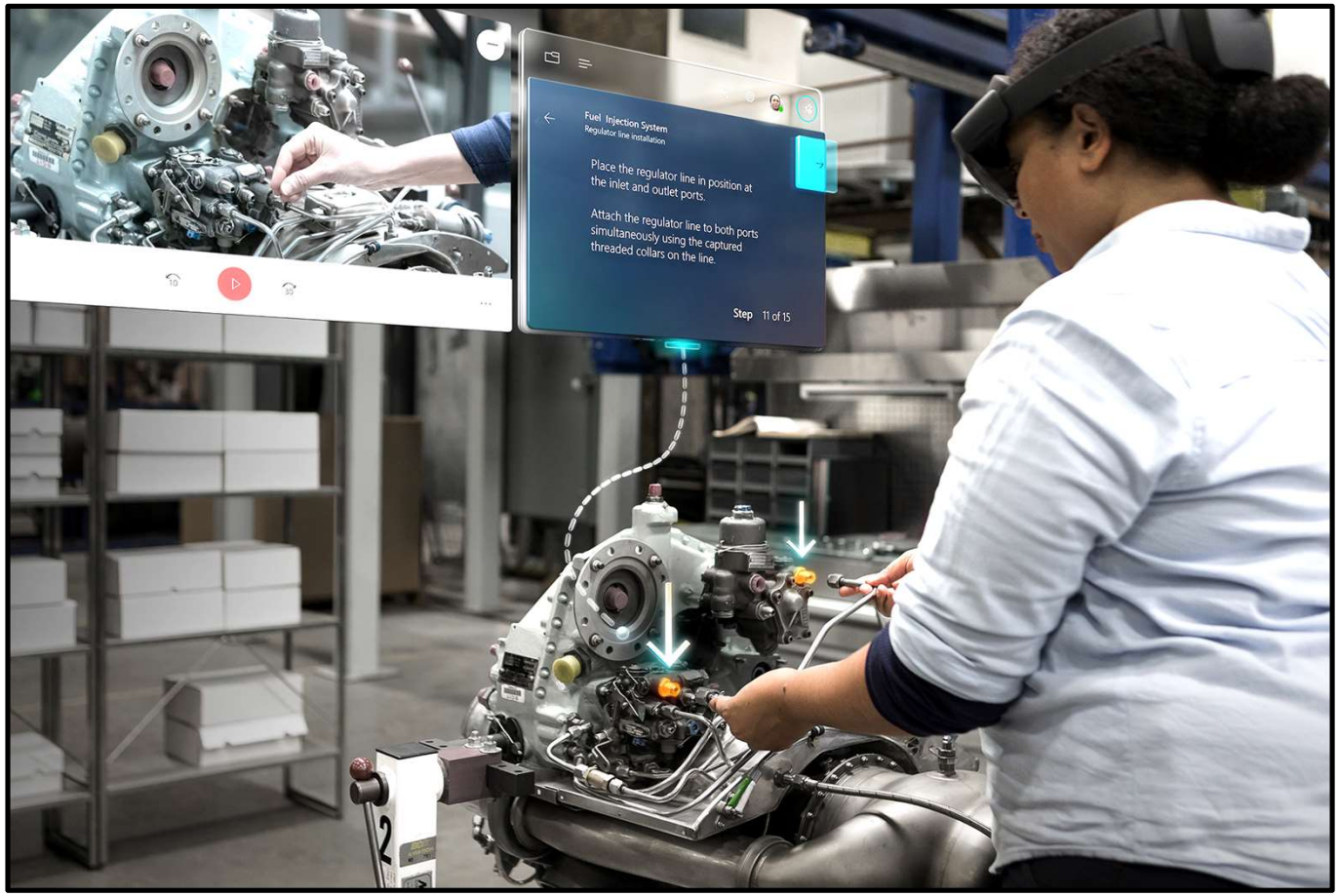
The Hololens uses ***waveguides***, pieces of glass in front of your eye that are carefully etched so they can reflect the 3D displays. When you put the whole system together — the lasers, the mirrors, and the waveguide — you get a bright display with a wide field of view that doesn't have to be precisely aimed into your eyes to work.

The internal processor is an ARM-based Qualcomm Snapdragon 850, which is designed to be very battery-efficient.

Microsoft Hololens 2 Components

Microsoft

Microsoft Hololens 2 used to Guide Mechanical Assembly Operations



Microsoft

Extended Reality

Extended reality (*XR*) is a term referring to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables, where the 'X' represents a variable for any current or future spatial computing technologies. It includes representative forms such as augmented reality (AR), mixed reality (MR), and virtual reality (VR) and the areas interpolated among them. The levels of virtuality range from partially sensory inputs to immersive virtuality, also called VR.

XR is a superset which includes the entire spectrum from "the complete real" to "the complete virtual" in the concept of reality–virtuality continuum ... Still, its connotation lies in the extension of human experiences especially relating to the senses of existence (represented by VR) and the acquisition of cognition (represented by AR). With the continuous development in human–computer interactions, this connotation is still evolving.

https://en.wikipedia.org/wiki/Extended_reality

Definitions of Mixed Reality and Augmented Virtuality

Mixed Reality (**MR**) is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time. Mixed reality does not exclusively take place in either the physical or virtual world, but is a hybrid of reality and virtual reality, encompassing both augmented reality and augmented virtuality via immersive technology.

https://en.wikipedia.org/wiki/Mixed_reality

Augmented Virtuality (**AV**) is a subcategory of mixed reality that refers to the merging of real-world objects into virtual worlds. As an intermediate case in the virtuality continuum, it refers to predominantly virtual spaces, where physical elements (such as physical objects or people) are dynamically integrated into and can interact with the virtual world in real time. This integration is achieved with the use of various techniques, such as streaming video from physical spaces, like through a webcam, or using the 3D digitalization of physical objects.

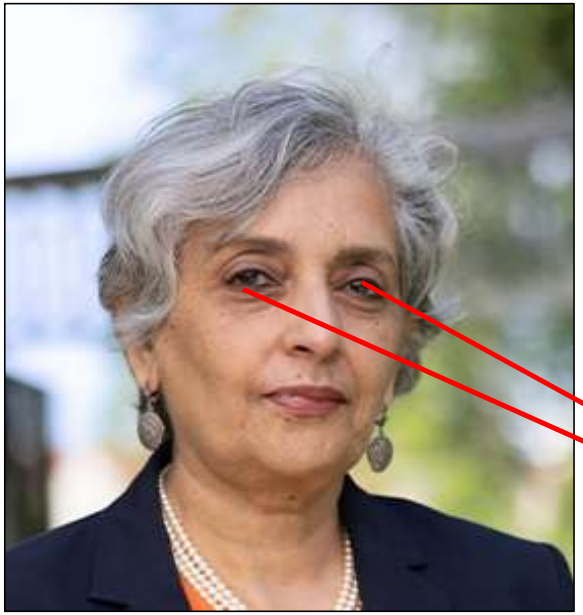
The use of real-world sensor information, such as gyroscopes, to control a virtual environment is an additional form of augmented virtuality, in which external inputs provide context for the virtual view.

https://en.wikipedia.org/wiki/Mixed_reality

VR/AR Usually Involves Binocular Vision, but Doesn't Have To

In everyday living, part of our perception of depth comes from the slight difference in how our two eyes see the world around us. This is known as *binocular vision*.

We care about this, and are discussing it, because stereo computer graphics can be a great help in de-cluttering a complex 3D scene. It can also enhance the feeling of being immersed in a movie.

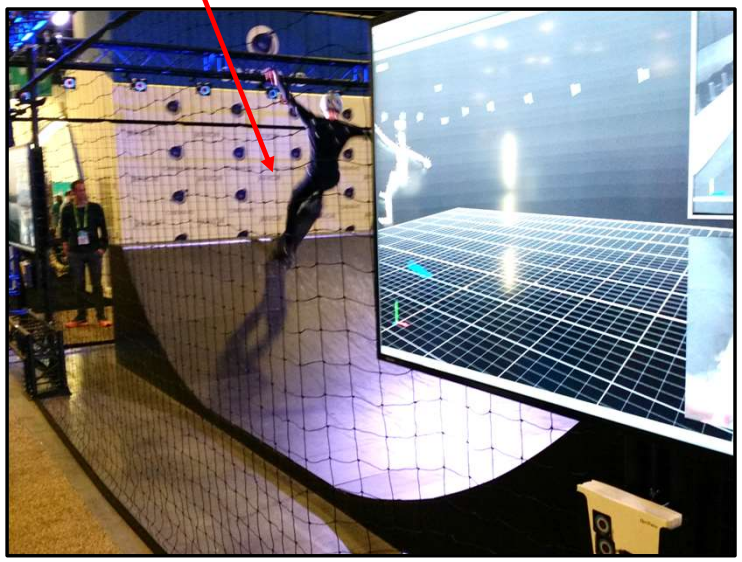


OSU's 16th President Dr. Jayathi Murthy

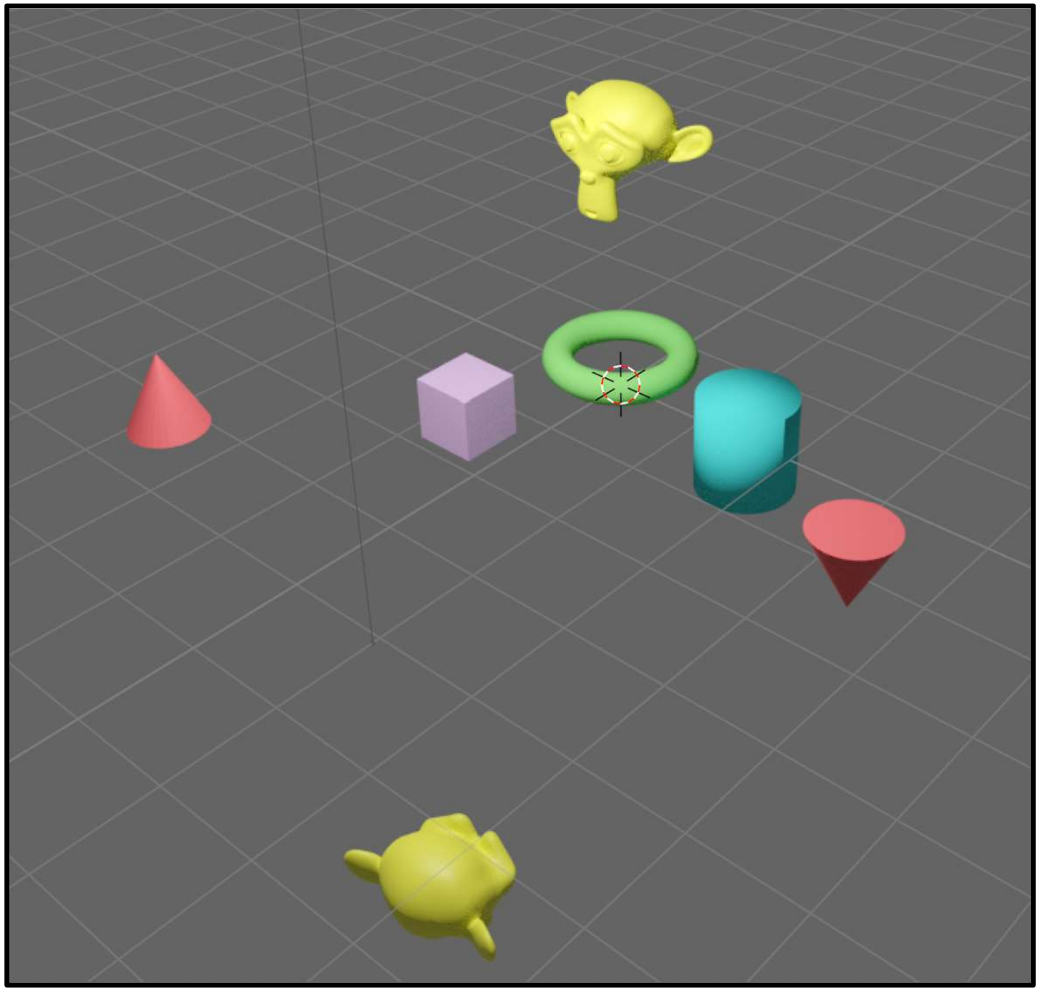
Tracking – Knowing where your Head is and How it is Oriented

3D Tracking Possibilities:

- Mechanical linkages
- Accelerometers and gyroscopes
- Motion Capture ("MoCap")
- Electromagnetic trackers

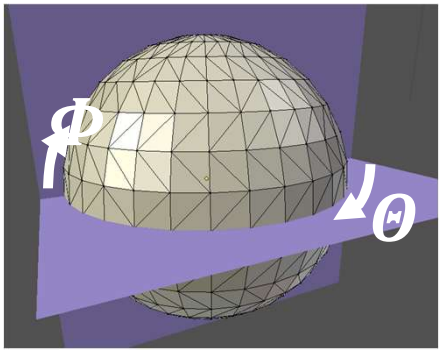


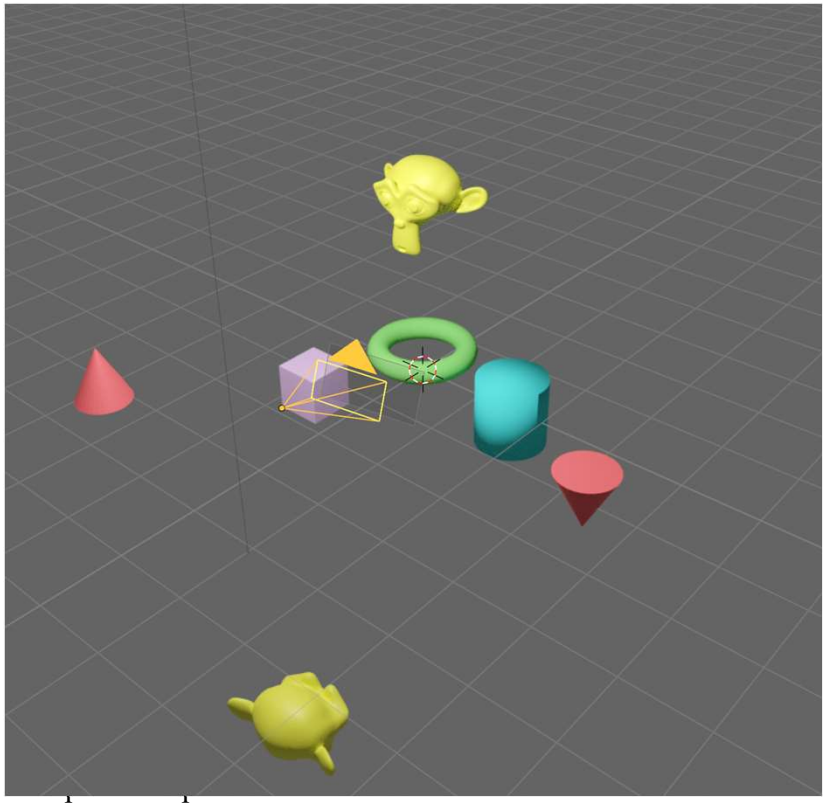
VR with Head Tracking is Good for Walking Through a 3D Scene, Even Without Stereographics



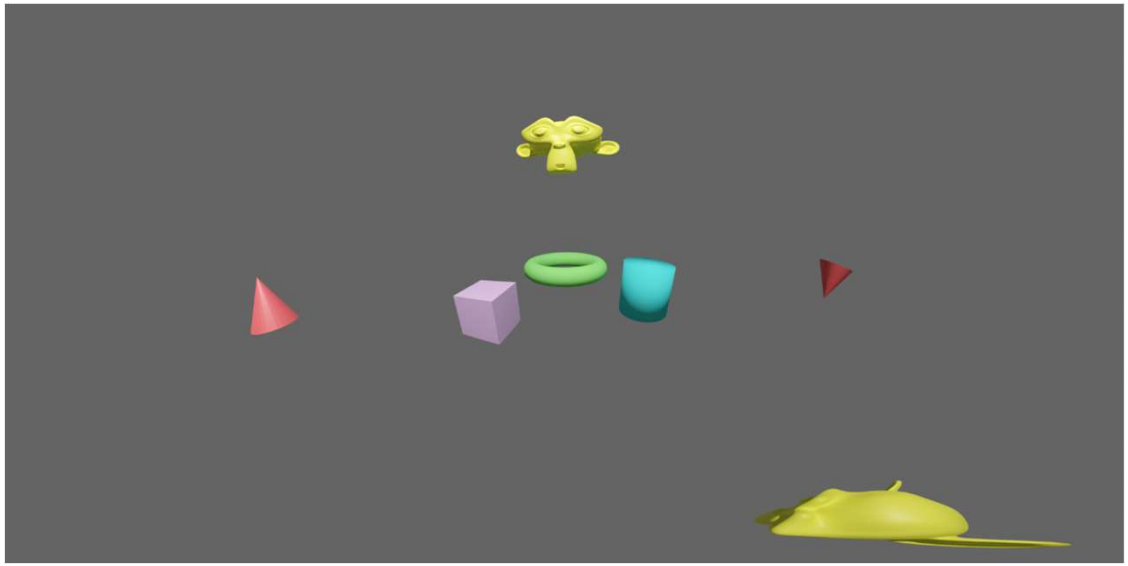
You Can Also Use VR to View a 360° 2D Spherical Image of the Scene

1. Put the eye/camera in the middle of the scene.
2. Render the scene, a vertical strip at a time, onto the inside of a sphere.
3. Save the image, where Θ→s and Φ→t



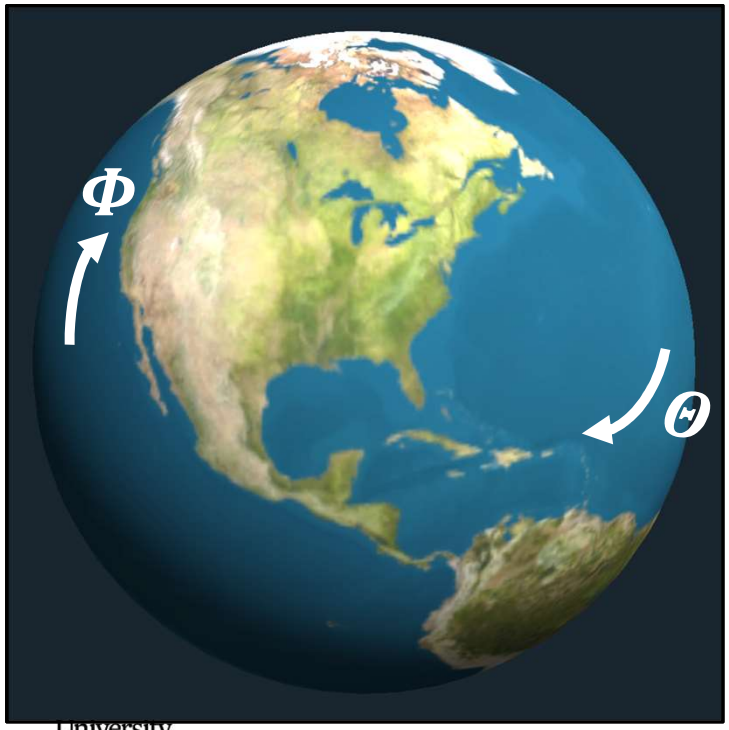


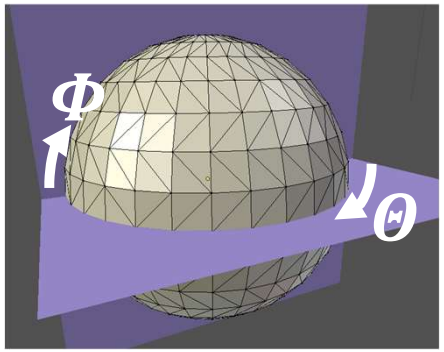
Φ



Θ

It's Roughly the Same Idea as Mapping a Texture Image onto a Sphere, but it's the Reverse Process of Mapping a Sphere onto an Image

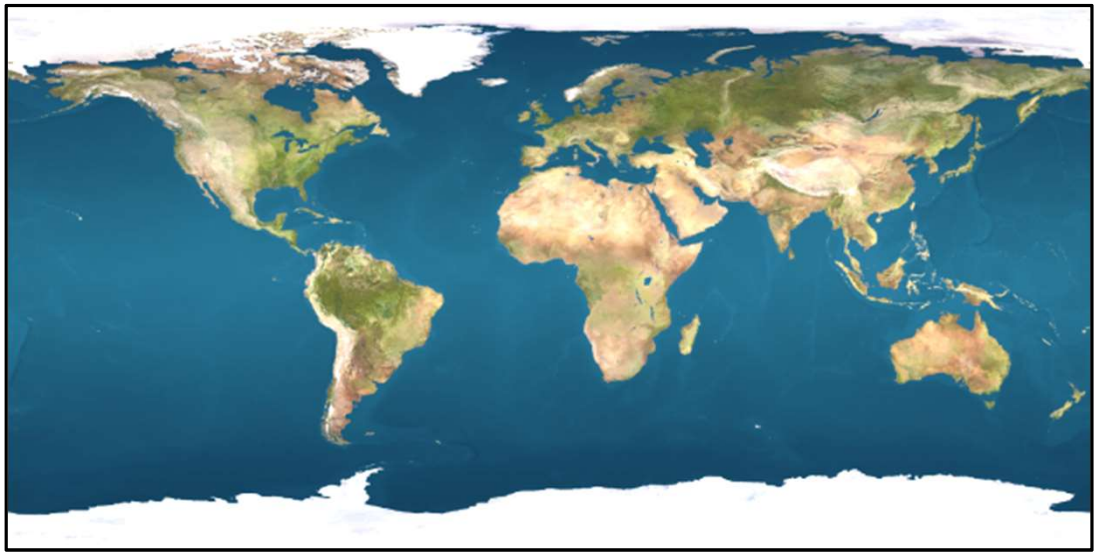




$\Theta \rightarrow s$

$\Phi \rightarrow t$

Φ

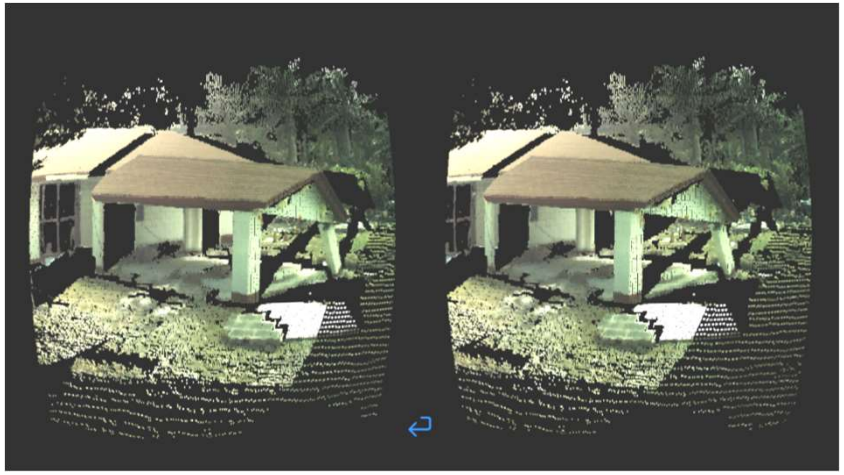


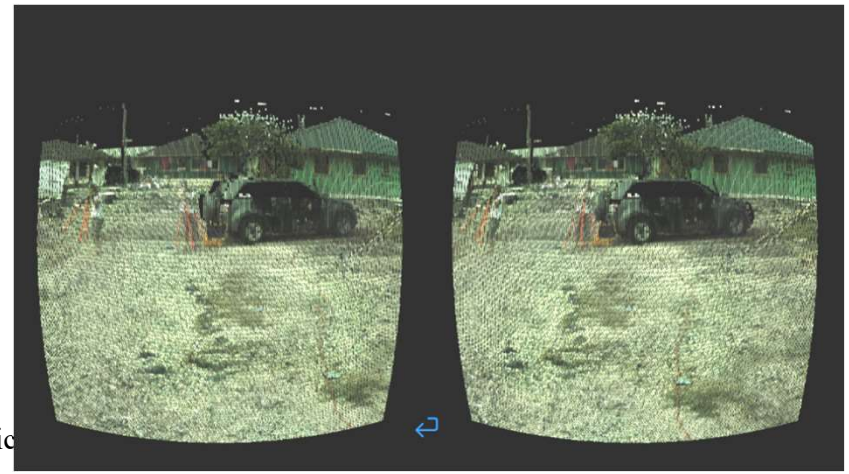
Θ

How to Create a 2D Spherical Image of a 3D Scene

```
void  DrawAndWriteSegments( )
{
        unsigned char array[3*PIXELS_PER_SEG*HEIGHT];
        glViewport( 0, 0, PIXELS_PER_SEG, HEIGHT );
        glMatrixMode( GL_PROJECTION );
        glLoadIdentity( );
        gluPerspective( PHIDEG, ASPECT_Y_OVER_X, ZNEAR, ZFAR );
        int col = 0;                    // column in the full array
        for( int lookDeg = -90.; lookDeg < 270; lookDeg += PHIDEG )
        {
                glMatrixMode( GL_MODELVIEW );
                glLoadIdentity( );
                float lx = Sind( (float)lookDeg )  +  EX;
                float ly = 0. + EY;
                float lz = Cosd( (float)lookDeg )  +  EZ;
                gluLookAt( EX, EY, EZ,  lx, ly, lz,  0., 1., 0. );
                glCallList( LidarList );
                glFlush( );
                glutSwapBuffers( );
                glFinish( );

                glPixelStorei( GL_PACK_ALIGNMENT, 1 );
                glReadPixels( 0, 0, PIXELS_PER_SEG, HEIGHT, GL_RGB, GL_UNSIGNED_BYTE, array );

                for( int y = 0; y < HEIGHT; y++ )
                {
                        memcpy( &FullArray[3*col*HEIGHT+y], &array[3*y*PIXELS_PER_SEG], 3*PIXELS_PER_SEG );
                }
                col += PIXELS_PER_SEG;
        }

        WriteArray( (char *)“Middle.bmp", FullArray );
}
```

You Can Also Do the Spherical Image Projection *with* Stereographics





Louis Panton

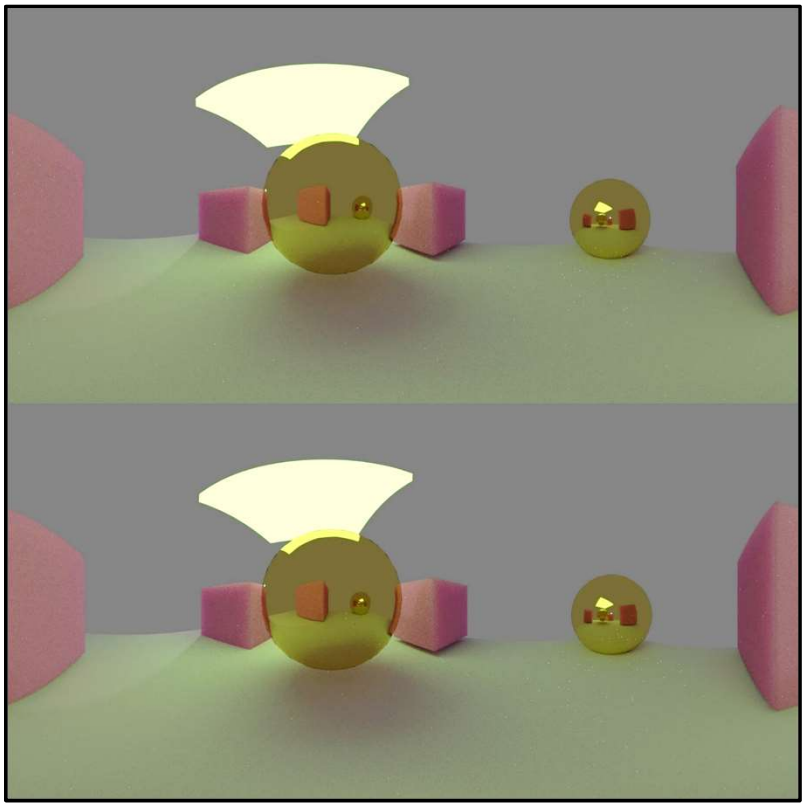
How to Create Two 2D Spherical Stereographics Images of a 3D Scene

```
void  DrawAndWriteSegments( )
{
        unsigned char array[3*PIXELS_PER_SEG*HEIGHT];
        for( int eye = 0; eye <= 1; eye++ )
        {
                glViewport( 0, 0, PIXELS_PER_SEG, HEIGHT );
                glMatrixMode( GL_PROJECTION );
                glLoadIdentity( );
                StereoPersp( PHIDEG, ASPECT_Y_OVER_X, ZNEAR, ZFAR, Z0P, eye == 0 ? -EYESEP : EYESEP );
                unsigned char *FullArray = ( eye == 0 ? Left : Right );
                int col = 0;                    // column in the full array
                for( int lookDeg = -90.; lookDeg < 270; lookDeg += PHIDEG )
                {
                        glMatrixMode( GL_MODELVIEW );
                        glLoadIdentity( );
                        float lx = Sind( (float)lookDeg )  +  EX;
                        float ly = 0. + EY;
                        float lz = Cosd( (float)lookDeg )  +  EZ;
                        gluLookAt( EX, EY, EZ,  lx, ly, lz,  0., 1., 0. );
                        glCallList( LidarList );
                        glFlush( );
                        glutSwapBuffers( );
                        glFinish( );

                        glPixelStorei( GL_PACK_ALIGNMENT, 1 );
                        glReadPixels( 0, 0, PIXELS_PER_SEG, HEIGHT, GL_RGB, GL_UNSIGNED_BYTE, array );

                        for( int y = 0; y < HEIGHT; y++ )
                        {
                                memcpy( &FullArray[3*col*HEIGHT+y], &array[3*y*PIXELS_PER_SEG], 3*PIXELS_PER_SEG );
                        }
                        col += PIXELS_PER_SEG;
                }

                WriteArray( eye == 0 ? (char *)"Left.bmp" : (char *)"Right.bmp", FullArray );
        }
}
```

Examples of Spherical Stereographics Images, Suitable for Displaying in a VR Headset

Left Eye View

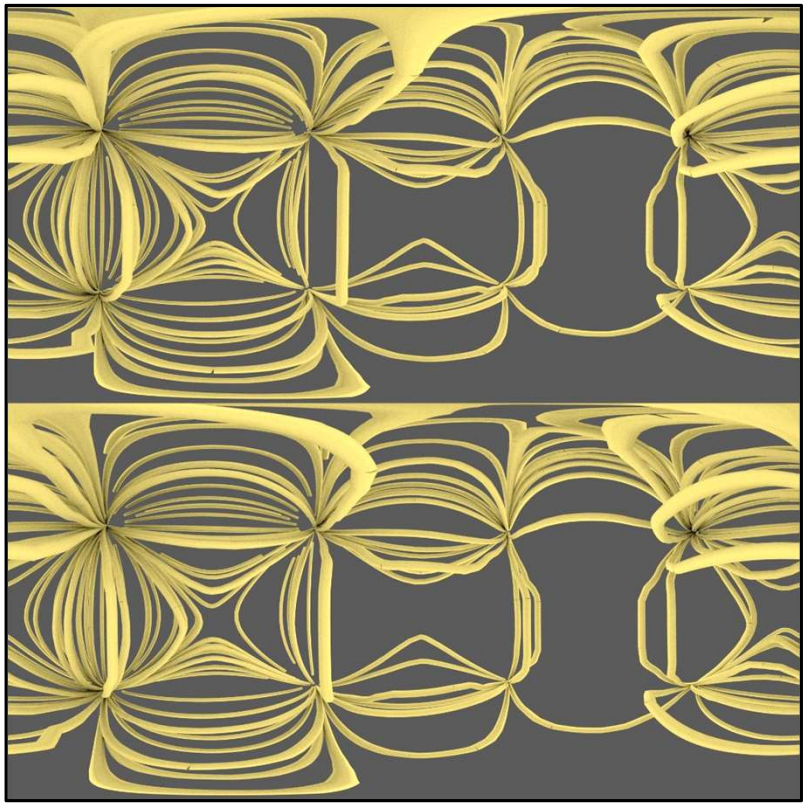
Right Eye View



Blender Scene

Left Eye View

Right Eye View



ParaView Fluid Flow Streamtubes

Examples of Spherical Stereographics Images, Suitable for Displaying in a VR Headset

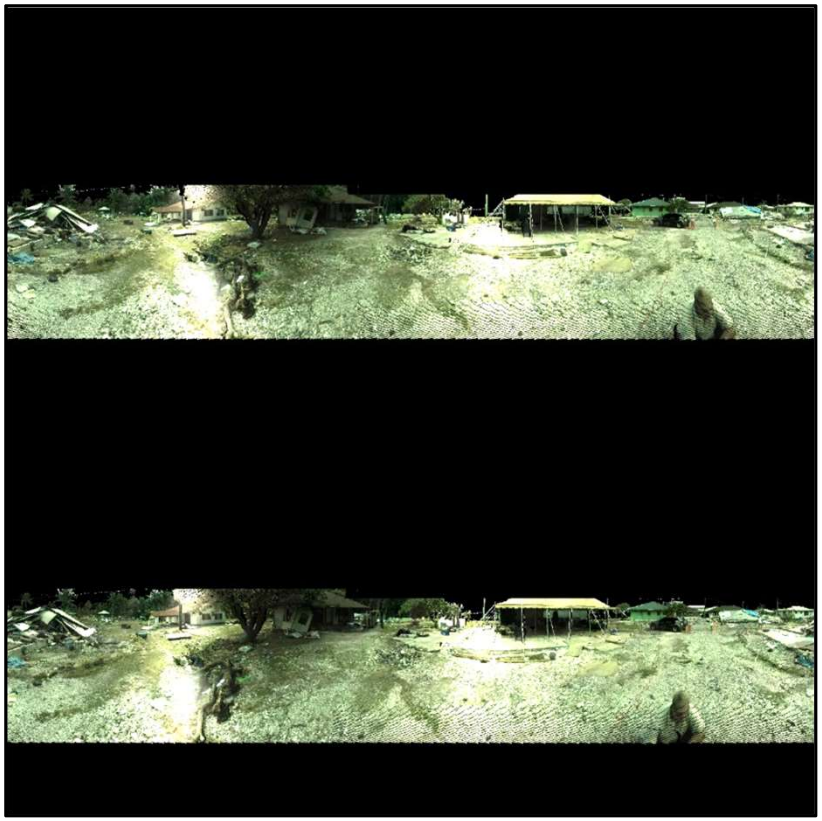
Left Eye View

Right Eye View

My Side Yard
(using a Garmin VIRB 360° camera)

Left Eye View

Right Eye View



Lidar Scene

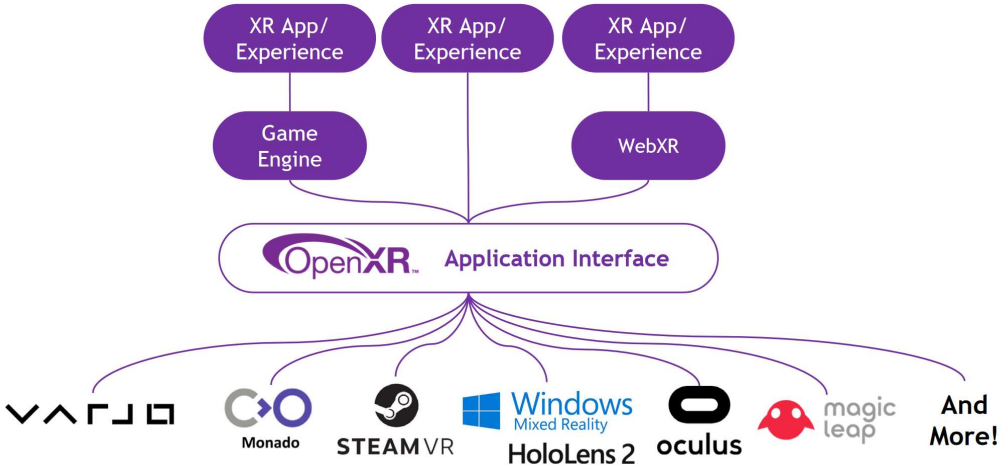
Data from Dr. Michael Olsen, OSU CCE

Something to Cut Through the VR/AR "Clutter" -- The Khronos Group's OpenXR API

OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices.

https://www.khronos.org/openxr/

OpenXR Provides a Common API Interface for XR the same way that OpenGL does for 3D Graphics



https://www.khronos.org/openxr/

Who's Been Involved with Creating OpenXR?

https://www.khronos.org/openxr/