

Lecture 25: Rendering Issues in Virtual and Augmented Reality

Yuhao Zhu

<http://yuhaozhu.com>
yzhu@rochester.edu

CSC 259/459, Fall 2024
Computer Imaging & Graphics

The Roadmap

Theoretical Preliminaries

Human Visual Systems

Digital Camera Imaging

Modeling and Rendering



3D Modeling

Rasterization and GPU

Ray Tracing

Shading

Rendering in AR/VR

Logistics

Final project due 12/16, 11:30 AM.

- Provide a write-up describing what you have done, along with the code.
- Submit a Jupyter notebook if you code in Python

Virtual Reality

immersive display; everything you see is emitted from the display.



Oculus Quest



Oculus Rift S



Samsung Gear VR

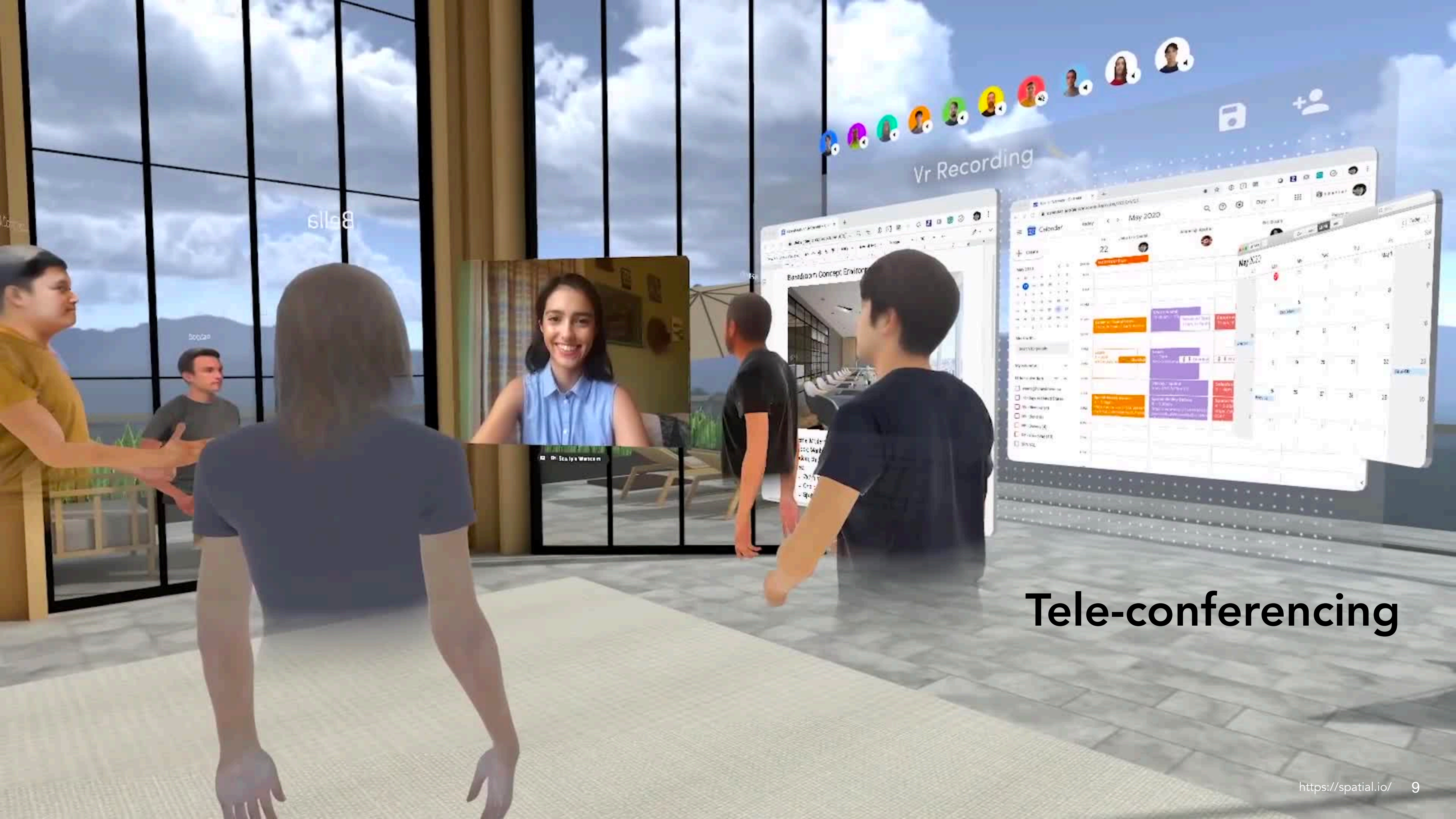


Google Cardboard



Medical Training

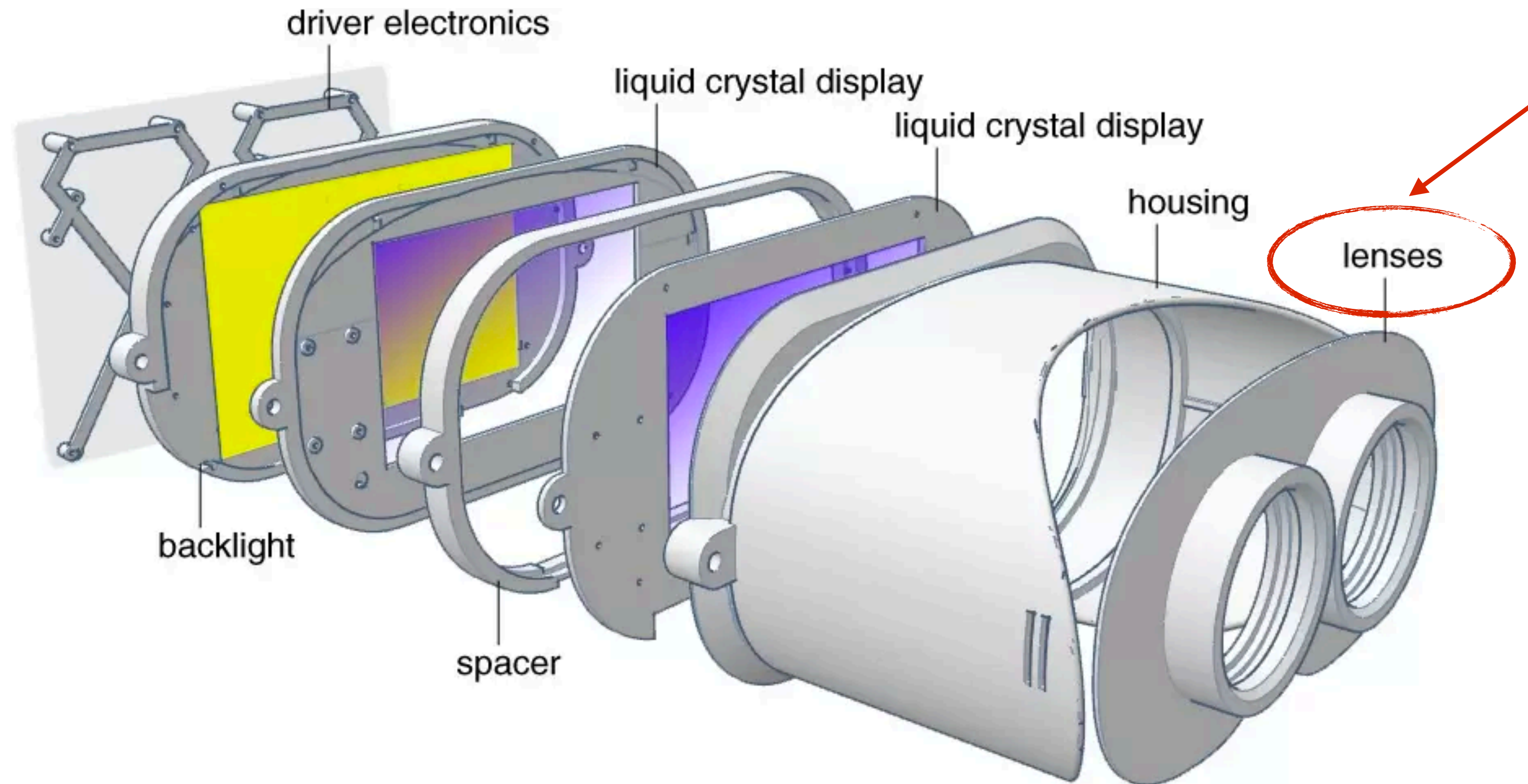




Tele-conferencing

VR Hardware

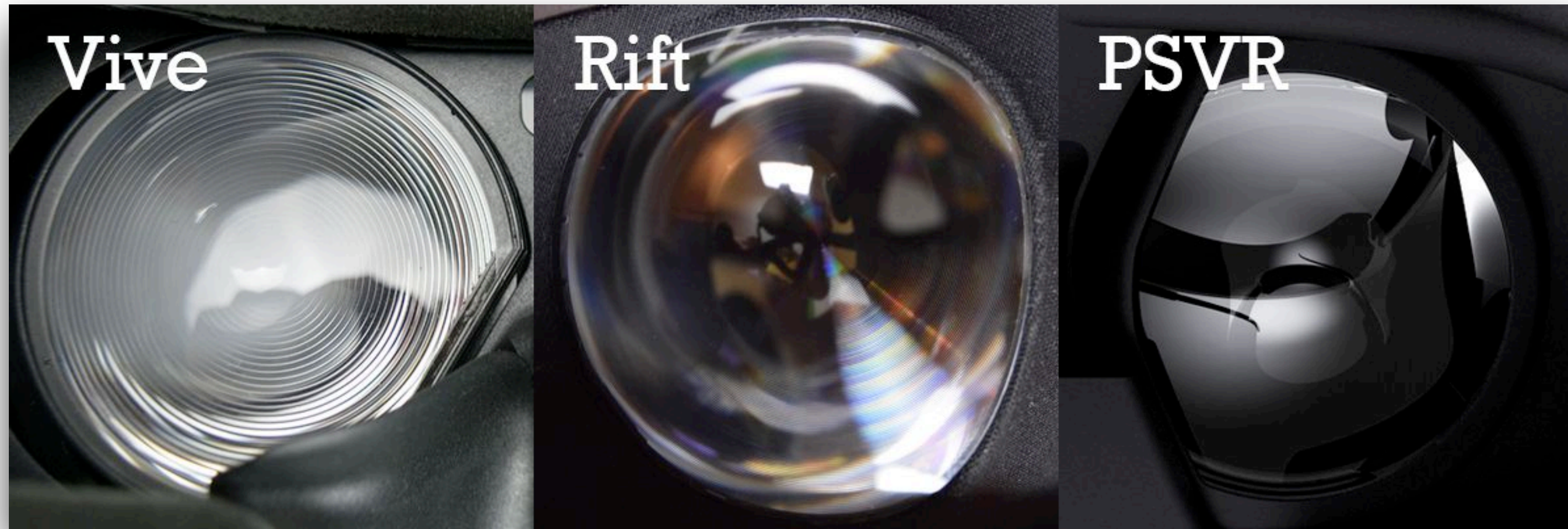
The Most Important Piece: Lenses



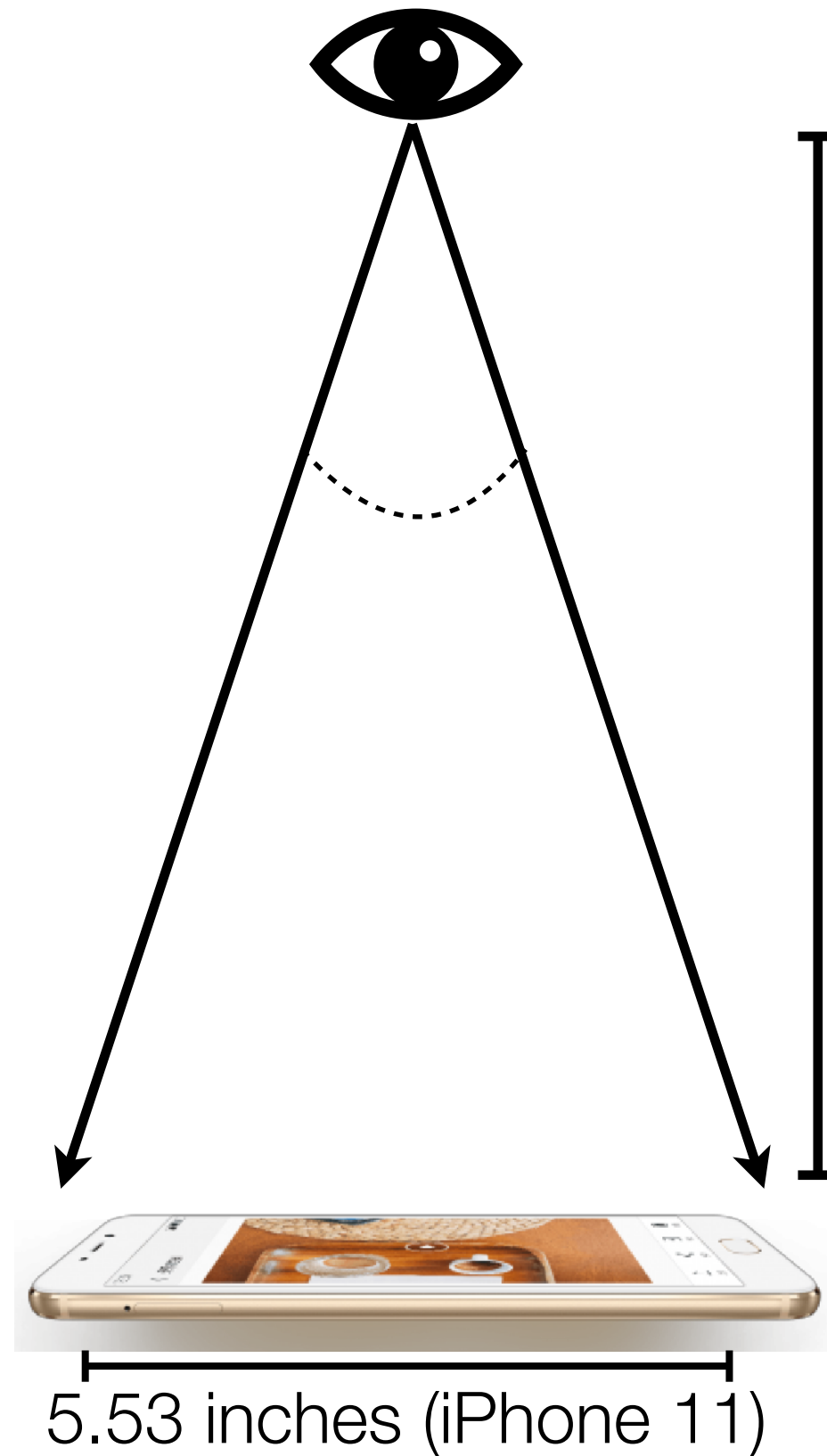
Why do we have to have lenses? Can't we just put a display directly in front of eyes?

The Most Important Piece: Lenses

Also called "eye piece"



Thought Experiment

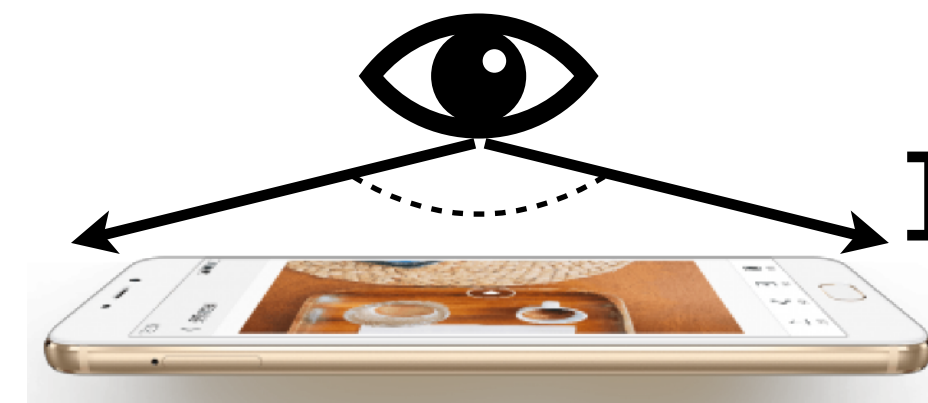


✓ Viewing distance: ~12 inches

Field of view: ~25°

Ideally: combine the best of both worlds:

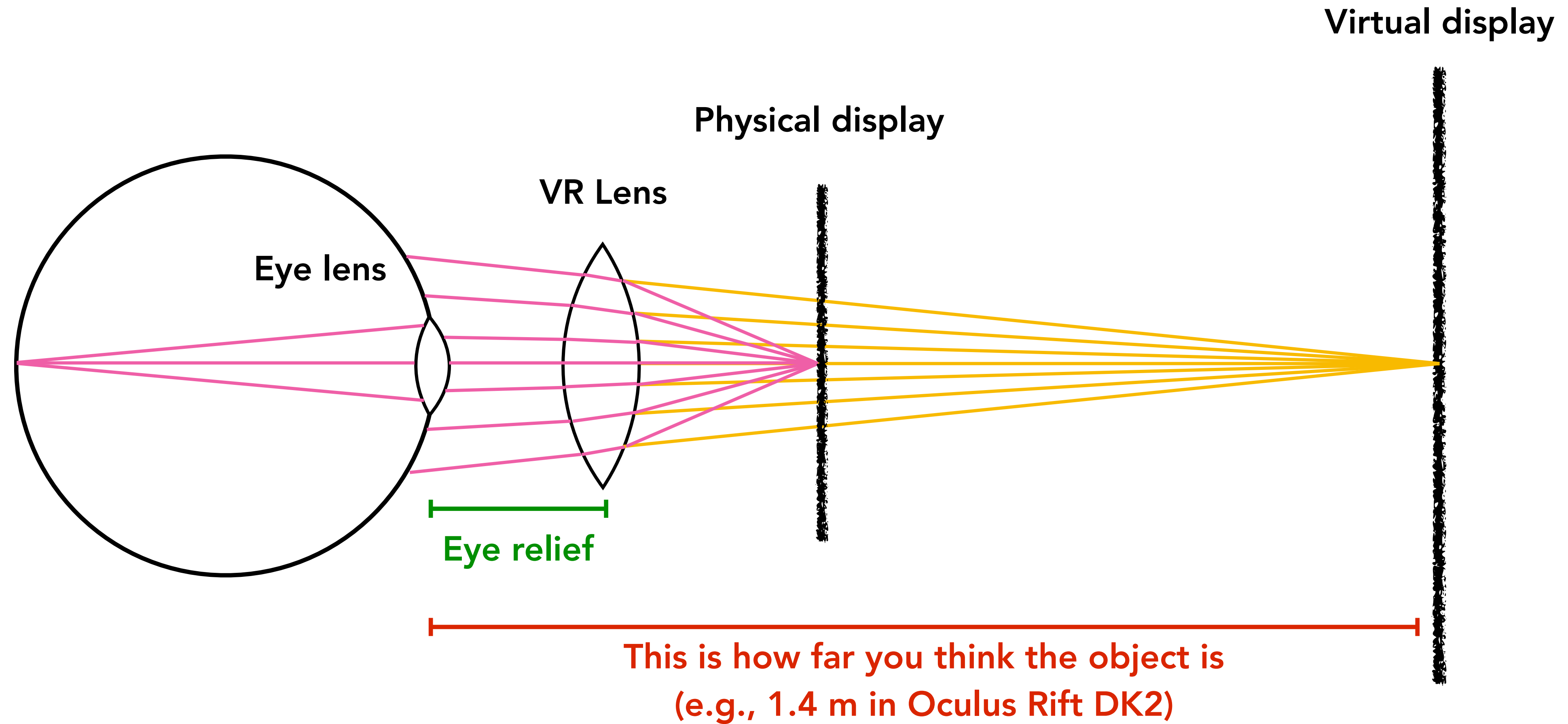
- Large field of view for immersive experience
- Longer viewing distance for eyes to focus



✓ Viewing distance: ~0.53 inches

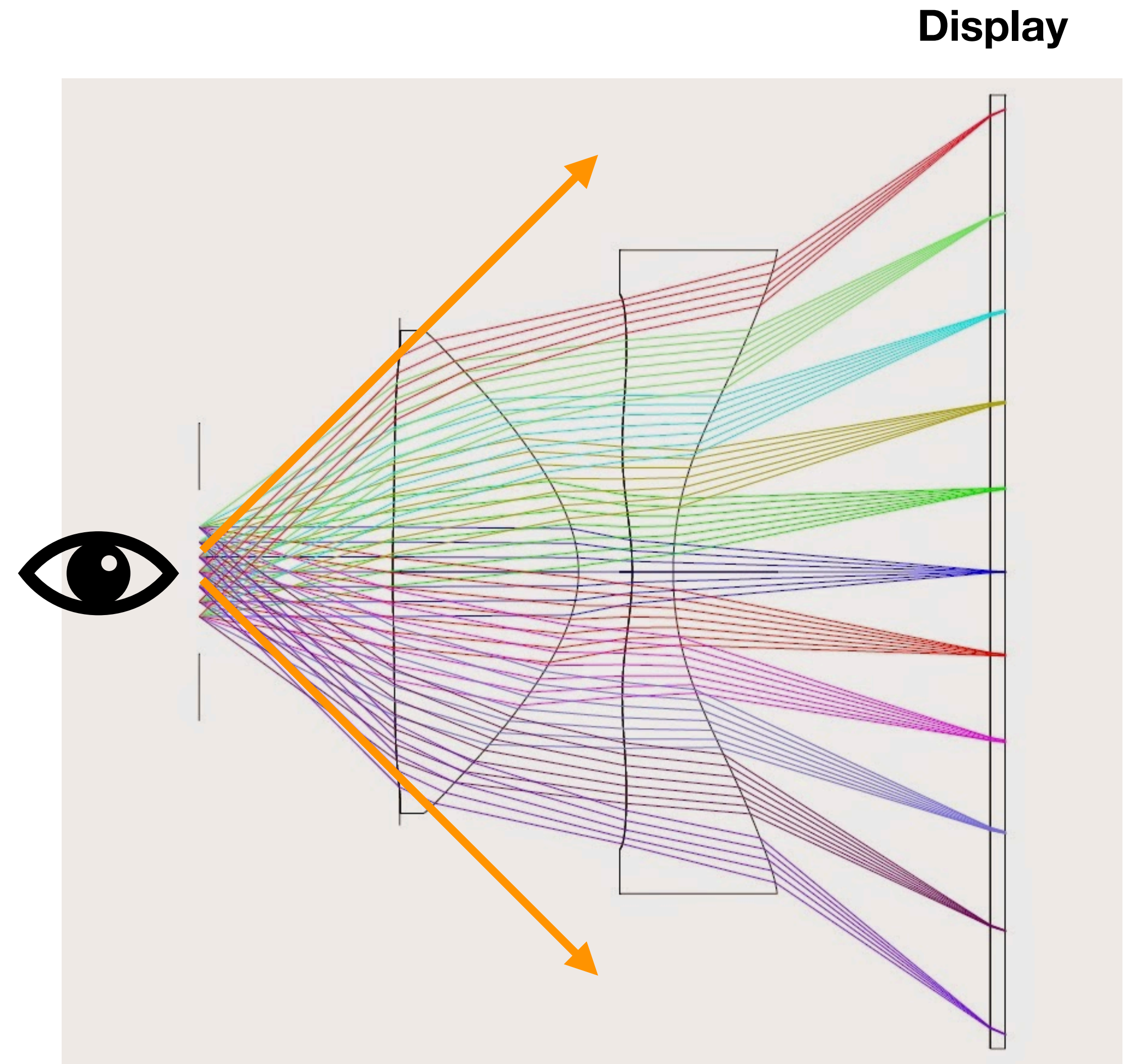
Field of view: ~160°

Lens in Virtual Reality Display



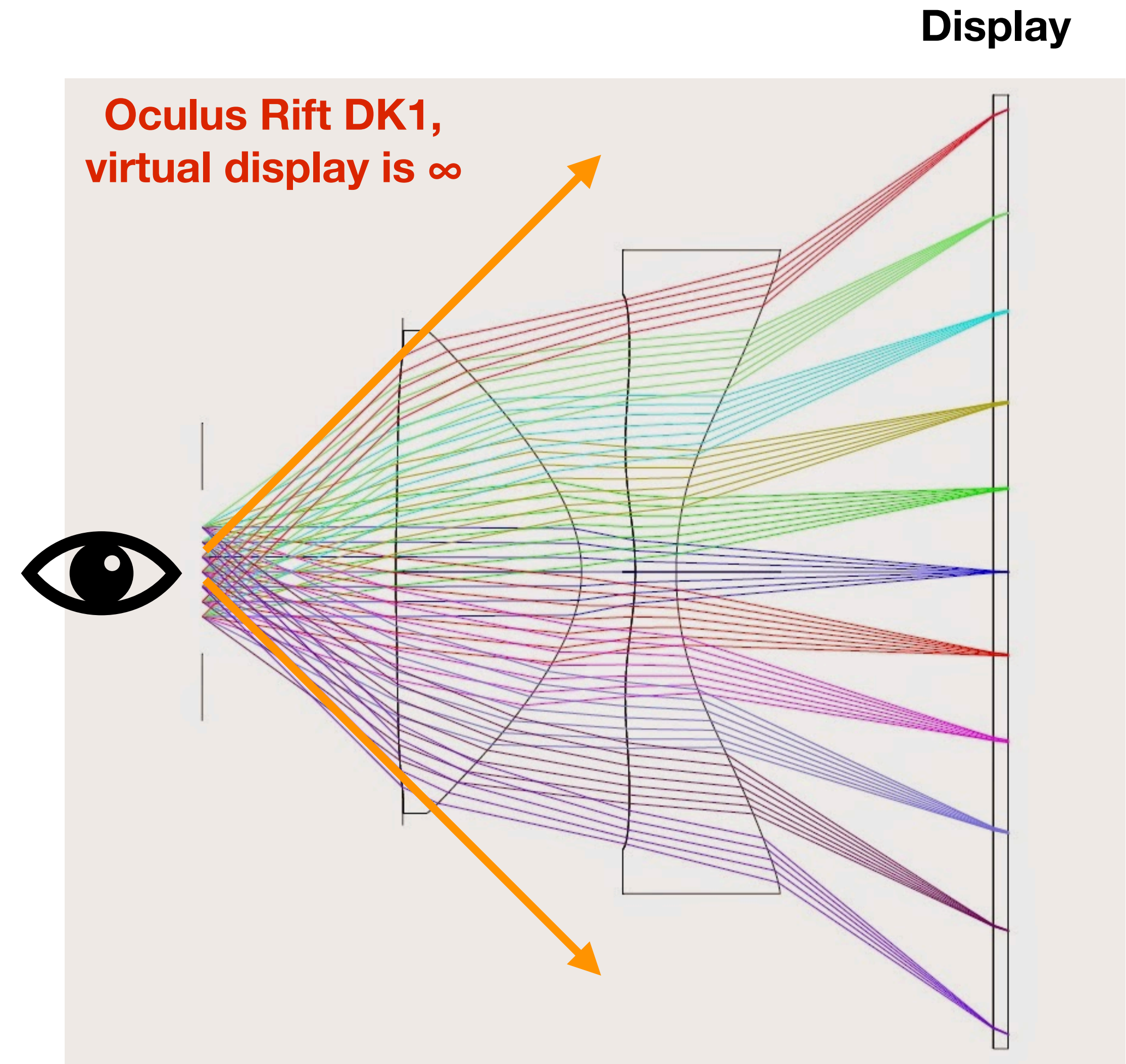
What Do Lenses Do?

- 1: create a wide field of view.
- 2: place the display far away from your eyes.



What Do Lenses Do?

- 1: create a wide field of view.
- 2: place the display far away from your eyes.
 - ...although the actual display is necessarily very close to your eyes.
 - Your eye lenses don't have enough power to focus that close: ~8 cm for teenagers and ~50 cm for elderlies.



VR Rendering

VR Rendering Overview

First, mind the difference between VR and 360° videos, which sometimes are called VR videos. We discussed the latter in earlier lectures.

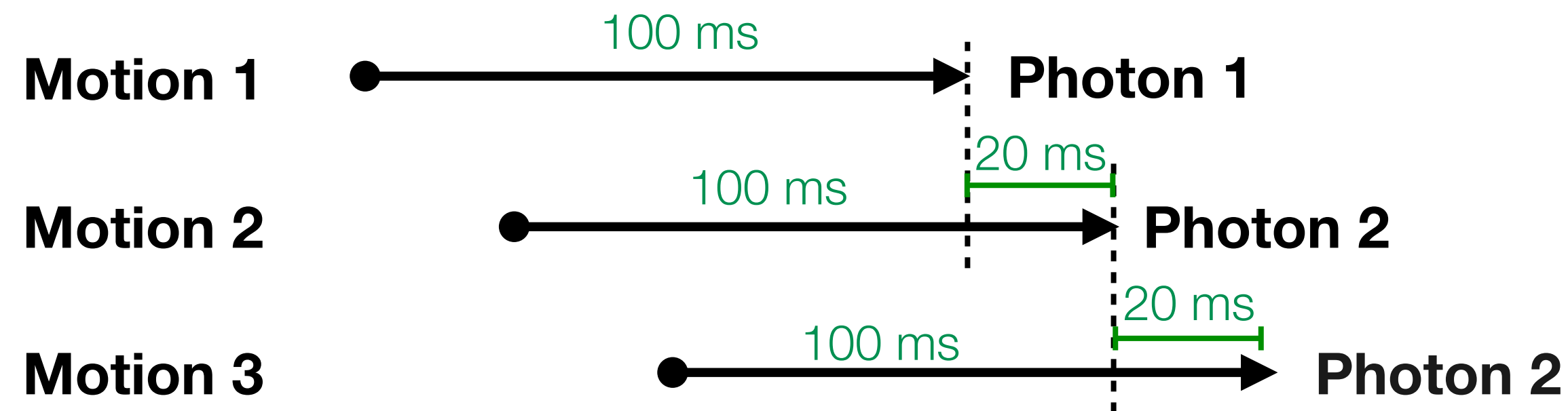
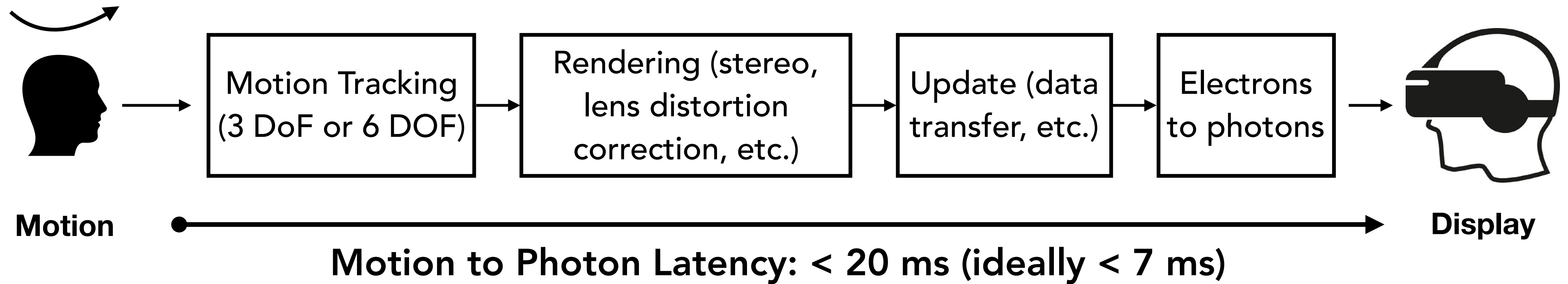
- Creating and rendering VR videos are light-field imaging/rendering in disguise.

Otherwise, rendering for VR is fundamentally no different from conventional rendering (for smartphones, computer displays, etc.)

But a few key differences exist:

- Need to track motion (rotation and/or translation)
- Need to render a stereo pair (vergence-accommodation conflict)
- Need to fix lens distortion
- Need to be aware of the low compute power (foveated rendering, eye tracking)

Grossly Simplified, Not-to-Scale End-to-End Pipeline



Note that pipelining improves throughput but doesn't help reducing the motion to photon latency!

VR Rendering

- **Tracking**

Motion Tracking

Recall: motion = translation + rotation

Responsible for rotation

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \times \begin{bmatrix} T_{00} & T_{01} & T_{02} \\ T_{10} & T_{11} & T_{12} \\ T_{20} & T_{21} & T_{22} \\ T_{30} & T_{31} & T_{32} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & 0_{3 \times 1} \\ T_{1 \times 3} & 1_{1 \times 1} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Responsible for translation

Rotation has 3 Degrees of Freedom, Not 6!

Around X
 $R_x:$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$$

Around Y
 $R_y:$

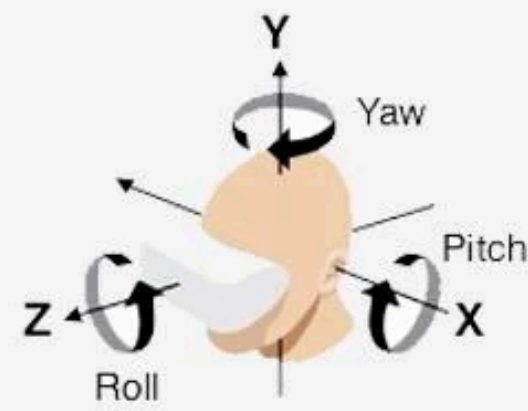
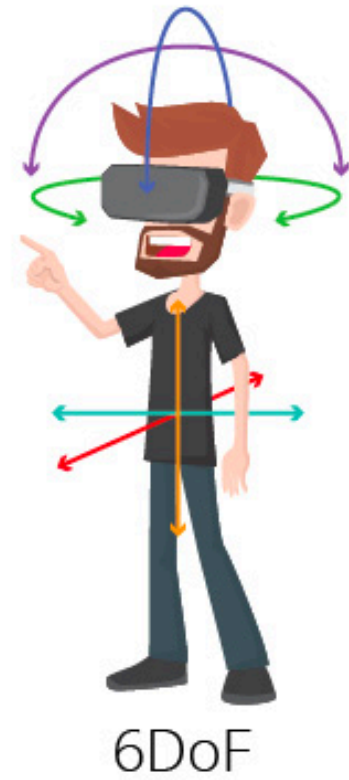
$$\begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix}$$

Around Z
 $R_z:$

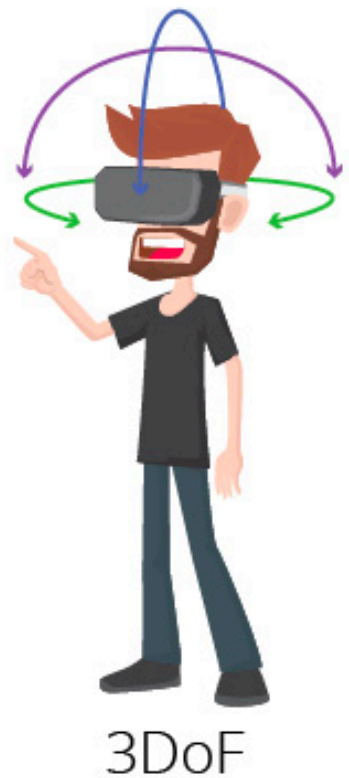
$$\begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix = $R_x \times R_y \times R_z$

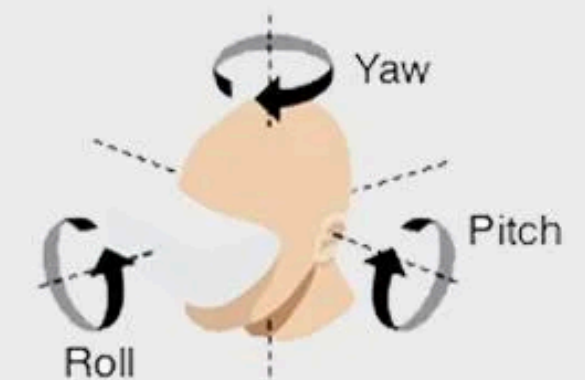
3 DoF vs. 6 DoF



6 DOF (3 DOF + X - T- Z)
 Head orientation + **Walk around**
 Full VR
 Object-based audio



3 DOF (Yaw - Pitch - Roll)
 Head orientation only
 360 Video
 Ambisonics



How to Track?

Can a sensor directly provide the 6 unknowns?

- Inertial Measurement Unit (IMU): gyroscope + accelerometer

If not (or sensor data only is unreliable), resort to compute vision techniques, essentially an optimization problem.

- Inside-out tracking
- Outside-in tracking

Usually combine IMU sensor + computational techniques

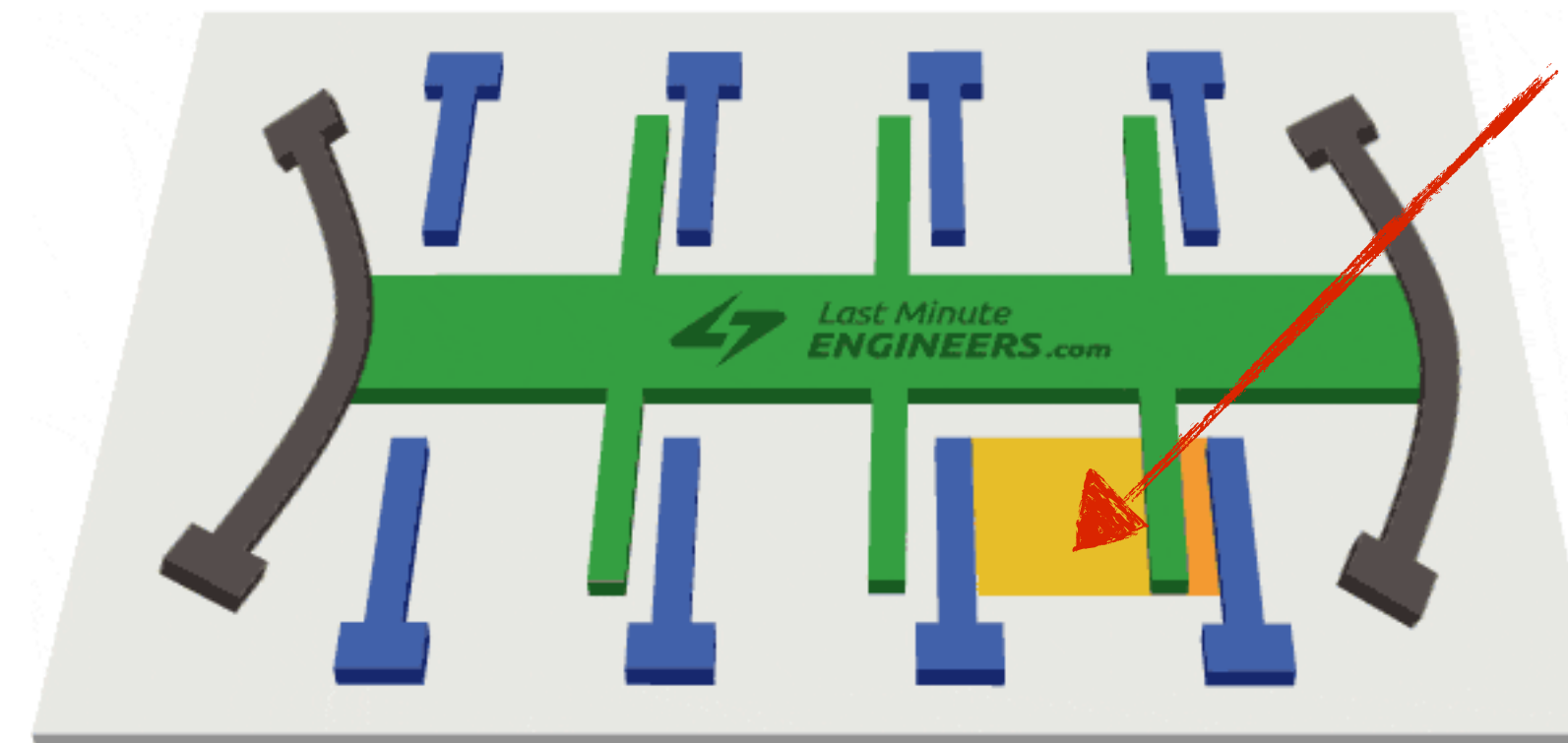
Or, cheat using teleportation!

Gyroscope and Accelerometer

Provides angular velocity



Provides translational acceleration



Change of capacitance is proportional to acceleration

Given the sampling rate of the sensor and the reading per sample, integrate to get the rotation and translation.

Computational Techniques for Tracking

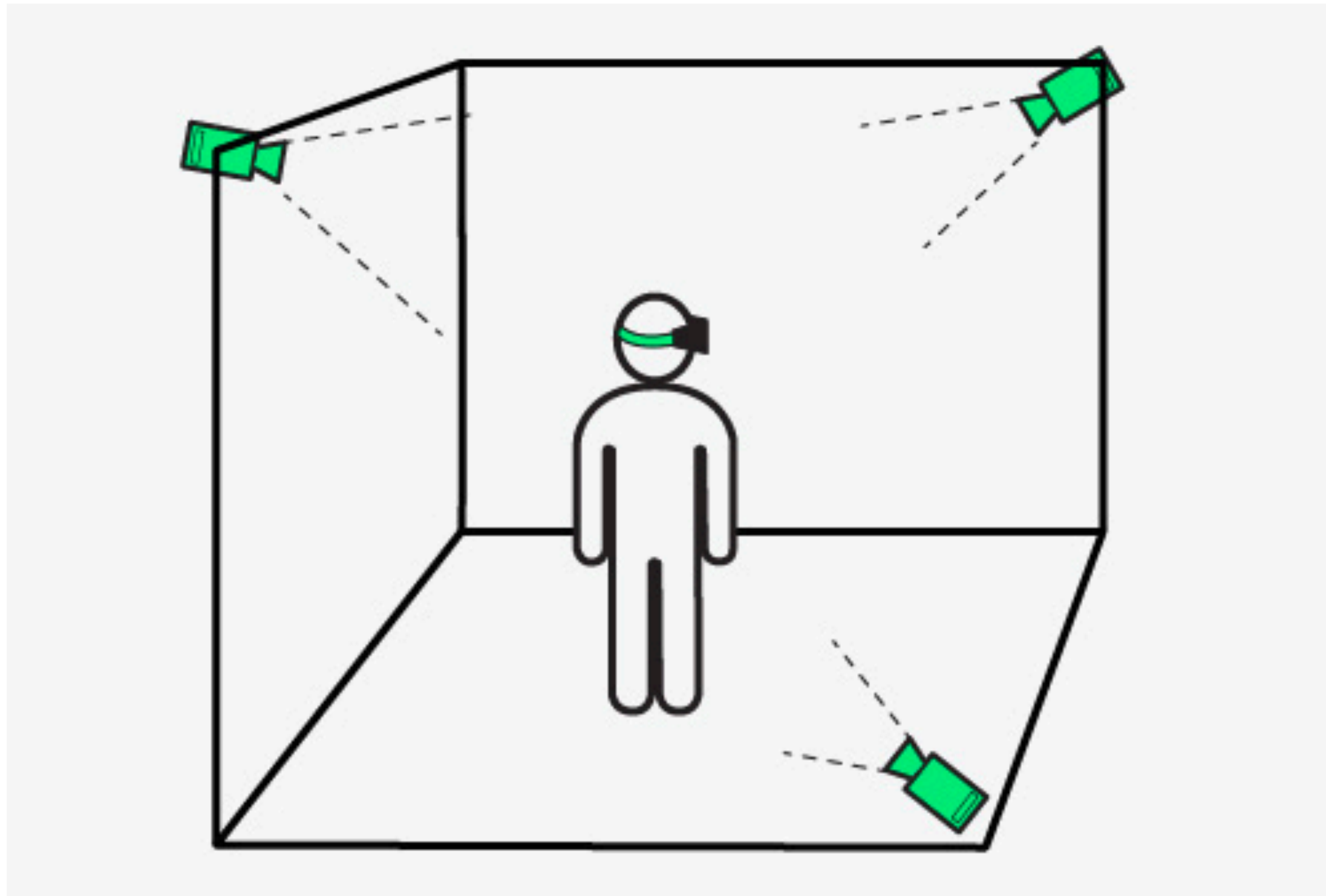
Ideally: solve the system of equations (6 unknowns: $\theta, \varphi, \psi, \Delta x, \Delta y, \Delta z$).

- Recall photogrammetry and SLAM in earlier lectures.
- Find the before/after coordinates of 6 points. Assuming rigid objects.
- In reality: find many points (over-determined) and minimize a loss.
- Either way, key is to find corresponding points in different camera captures.

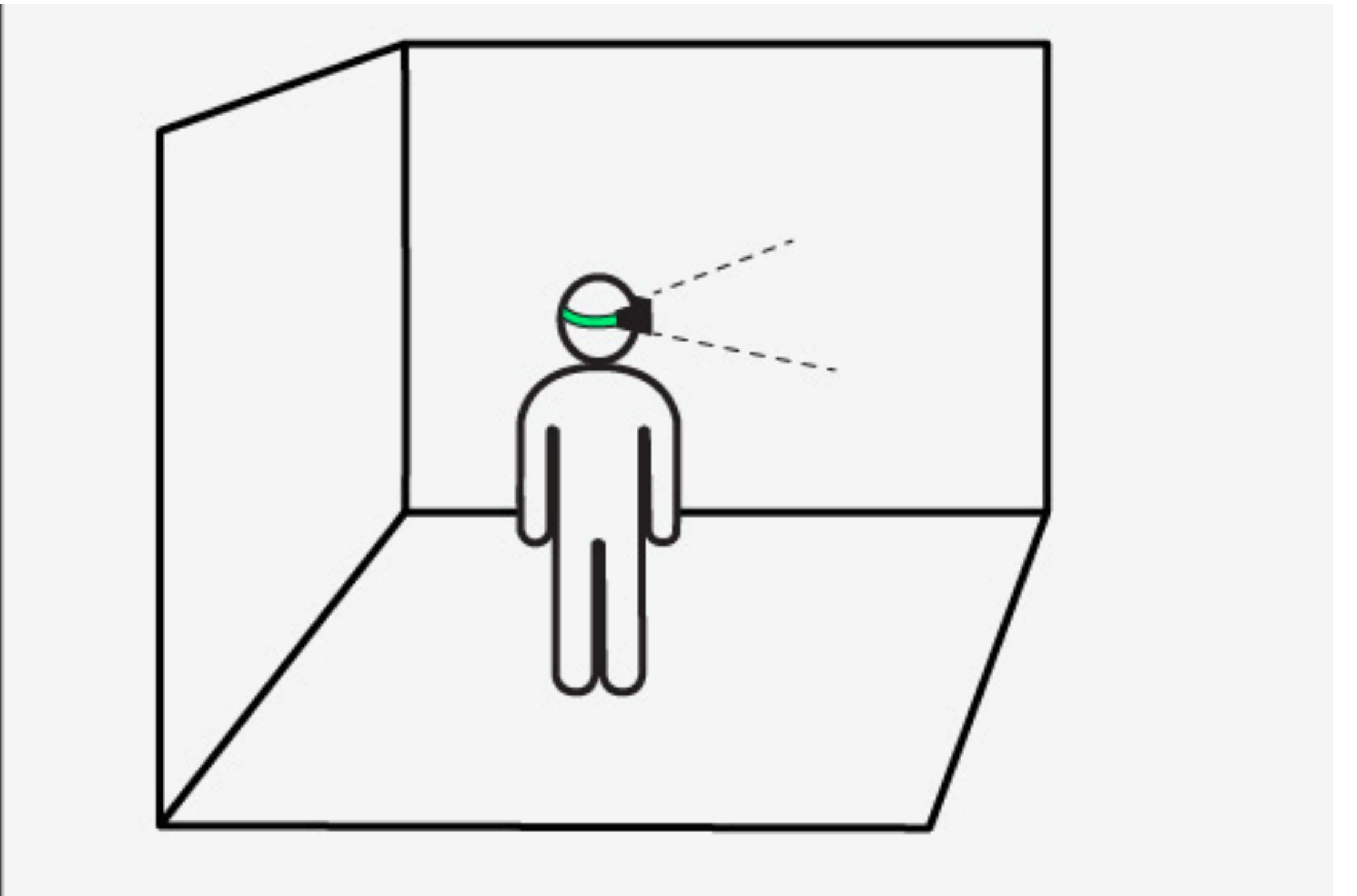
$$[x, y, z, 1] \times \begin{bmatrix} \text{A 3x3 matrix} \\ \text{parameterized} \\ \text{by } \theta, \varphi, \psi \\ \hline \Delta x & \Delta y & \Delta z \\ 0 & 0 & 0 \\ 1 & & \end{bmatrix} = [x', y', z', 1]$$

Outside-In vs. Inside-Out Tracking

Outside-in: cameras are outside, matching points are on the user.



Inside-out: cameras are on the user; matching points are in the scene.



Rift DK2: Outside-In Tracking Using IR Lights



40 IR LEDs provide 60 equations. The fixed pattern makes it easier to find correspondences between camera captures.

Inside-Out Tracking

Windows MR
(2x front cameras)



Oculus Quest
(4x corner cameras)



Oculus Rift S
(5x spaced cameras)

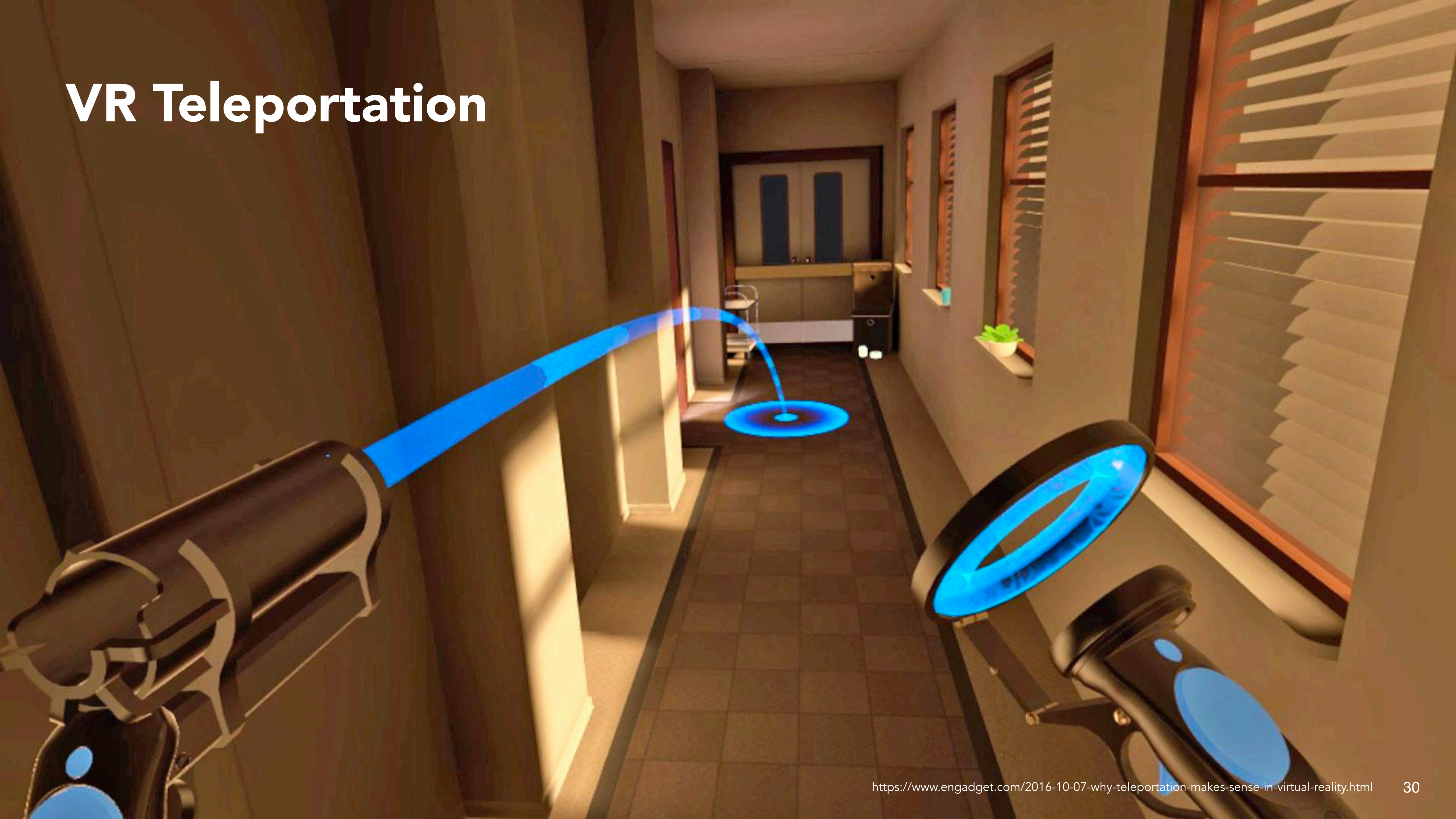


Valve Room, 2014. Use markers to simplify corresponding matching.



PC Perspective

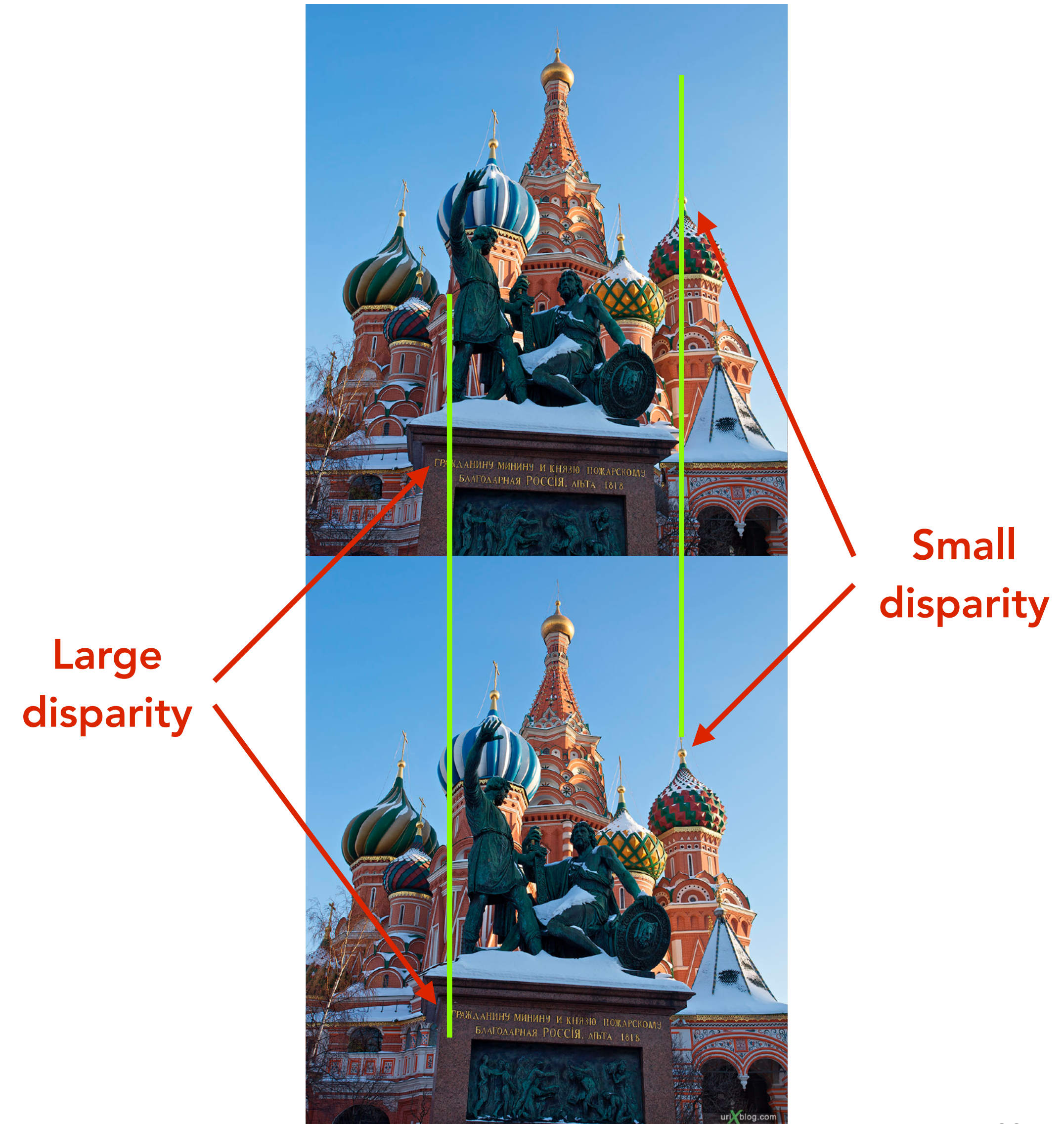
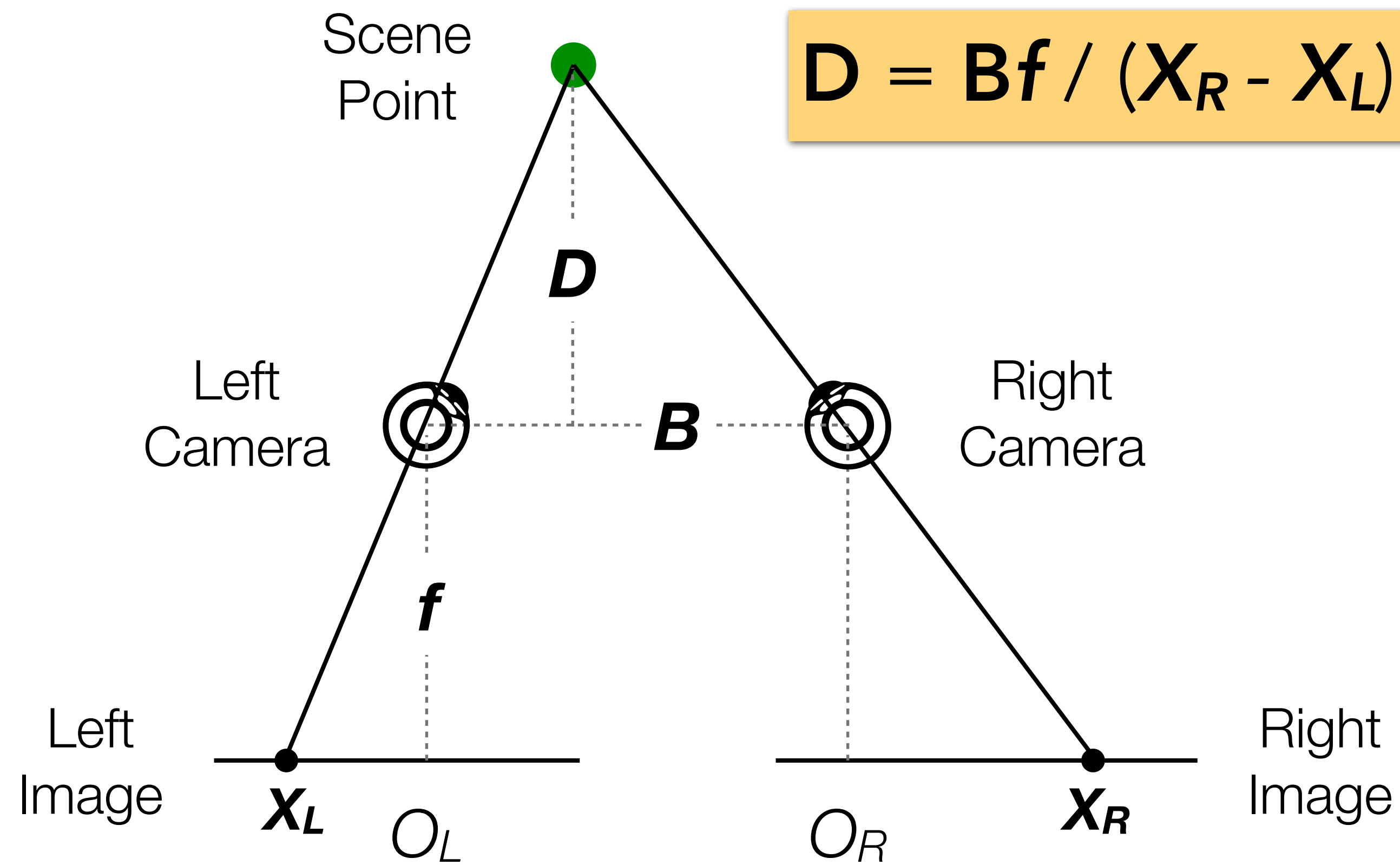
VR Teleportation



VR Rendering

- Tracking
- **Stereo rendering** (vergence-accommodation conflict)

Stereopsis: Depth Perception from Stereo



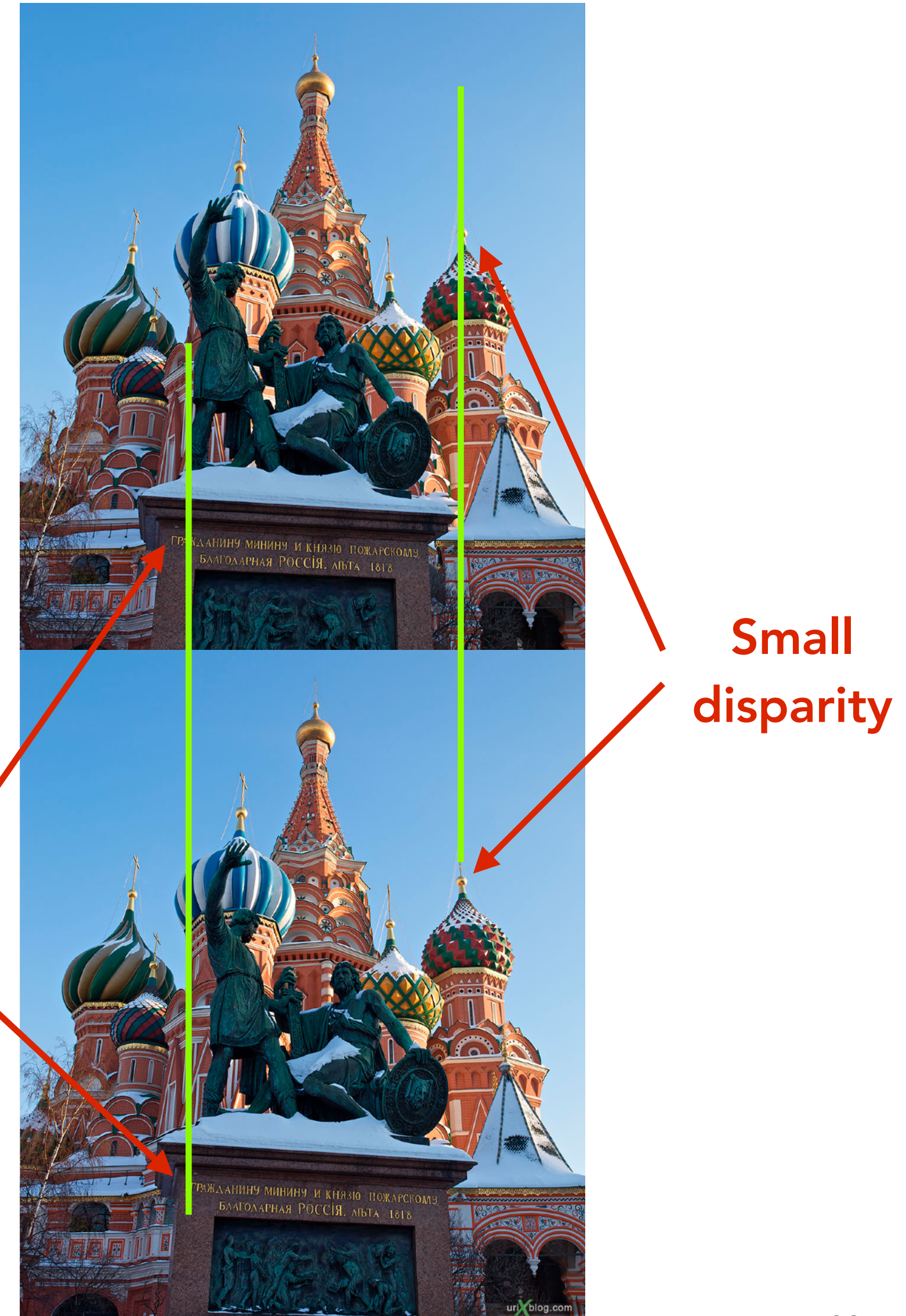
Stereopsis: Depth Perception from Stereo

Object depth is inversely proportional to the disparity on the images.

- This logic is “hard-wired” in your brain (evolution).
- If the disparity of an object is smaller than that of what you currently focus on, you know it’s farther away; vice versa.

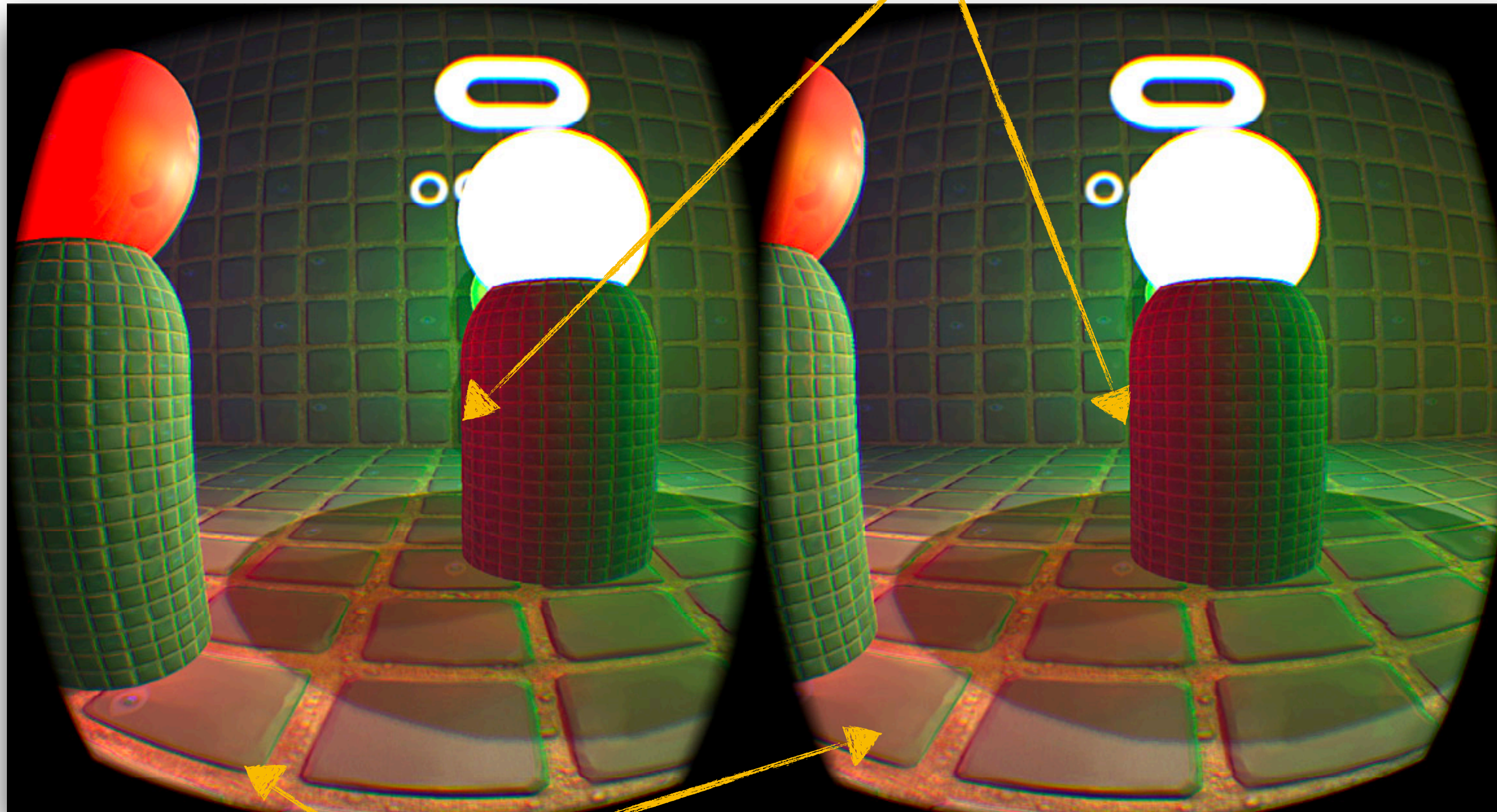
VR rendering ideally should render a stereo pair with the correct pixel disparities to provide the correct sense of depth.

- If not you will still get some sense of depth but weaker (other depth cues: occlusion, size, blur, etc.)



Stereo Rendering

Small
disparity

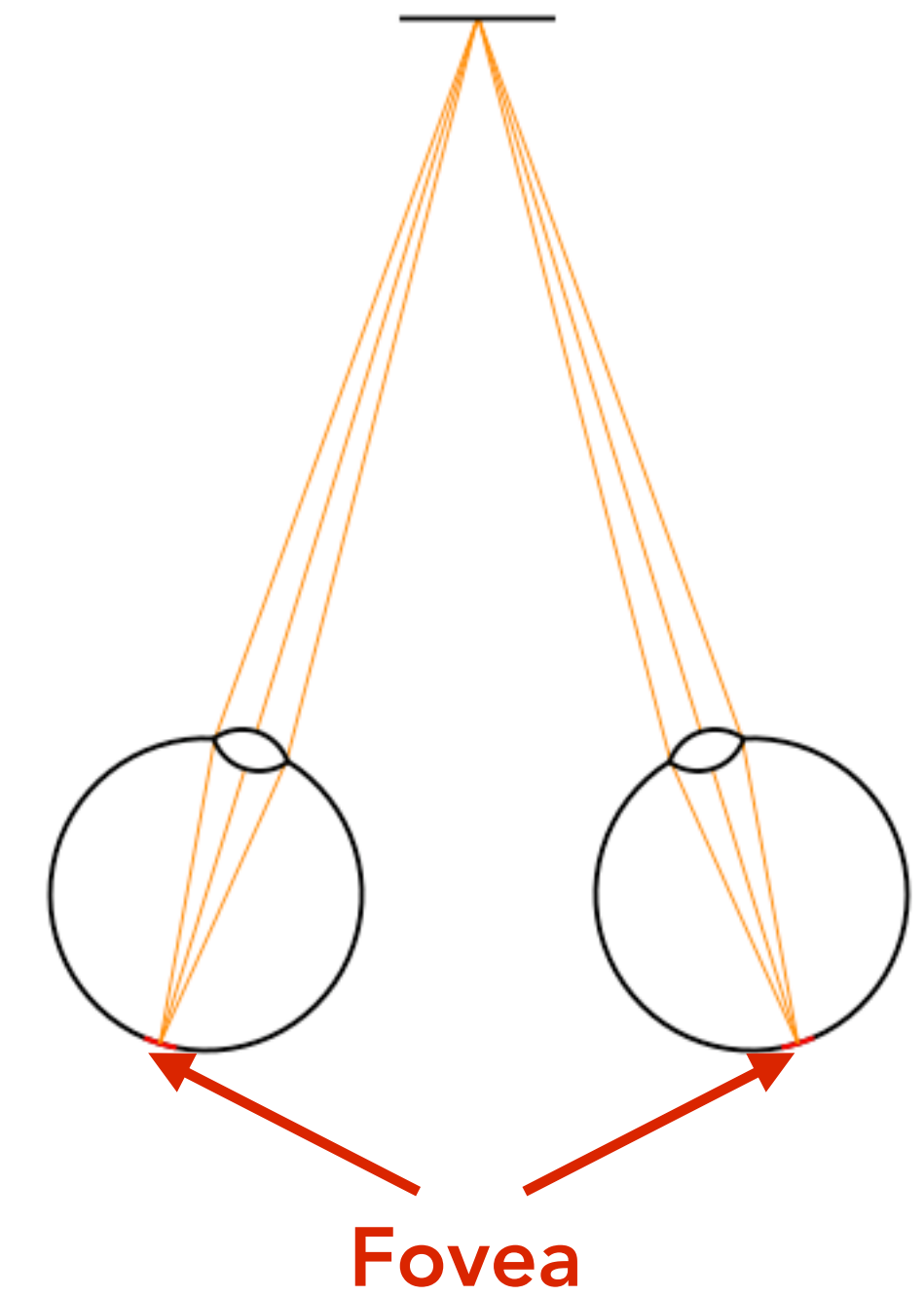
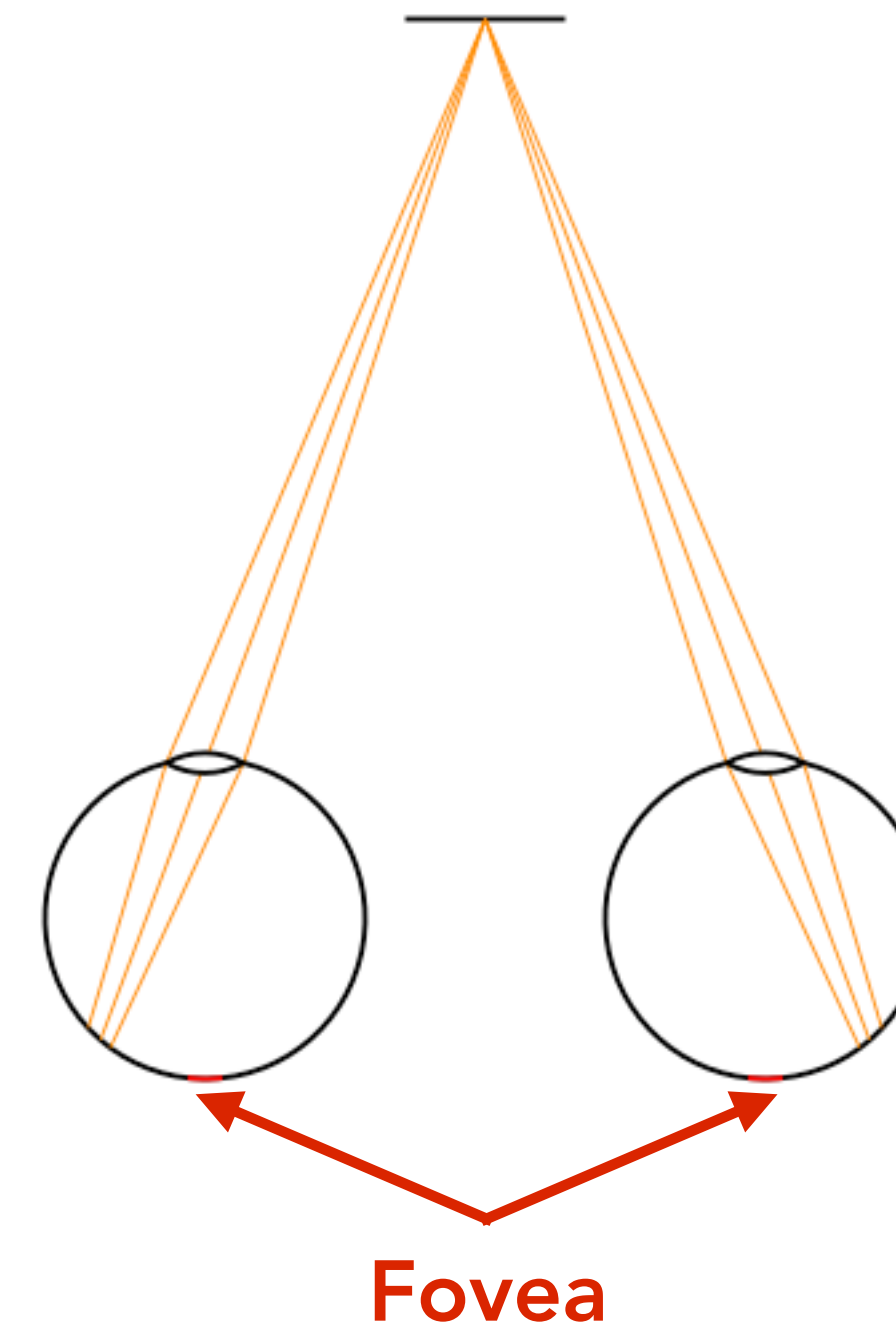
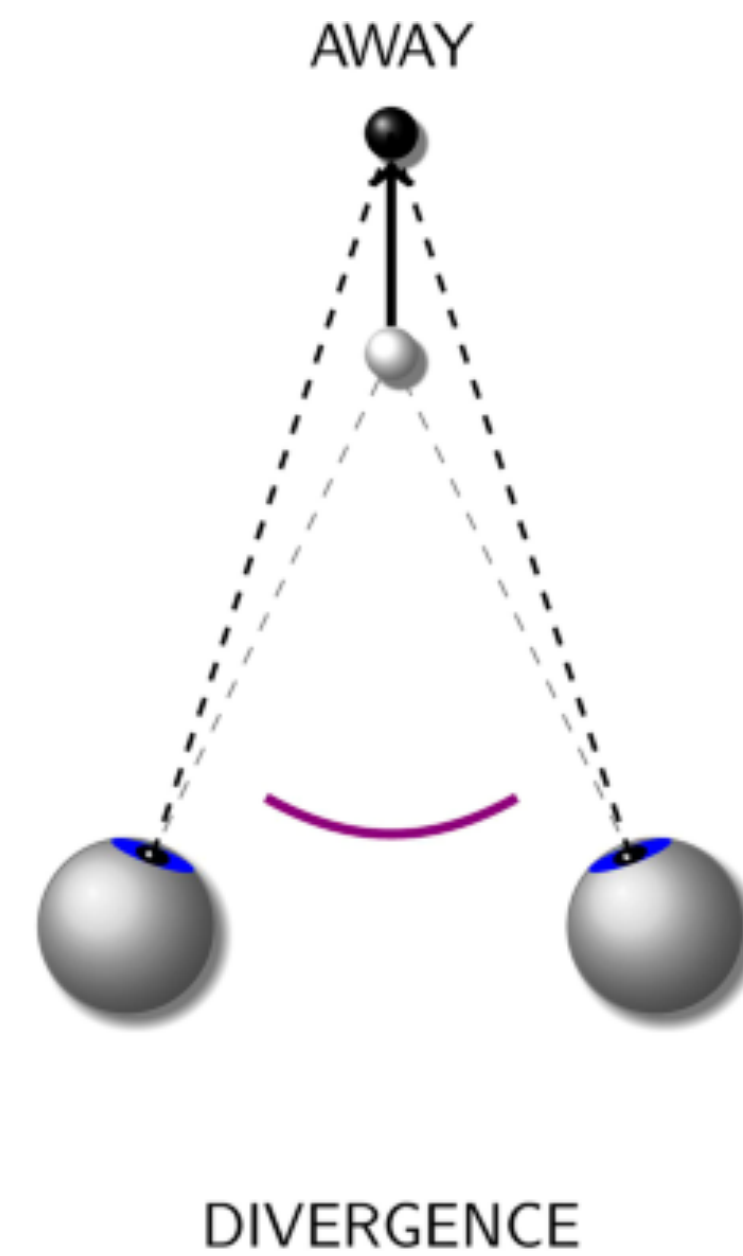
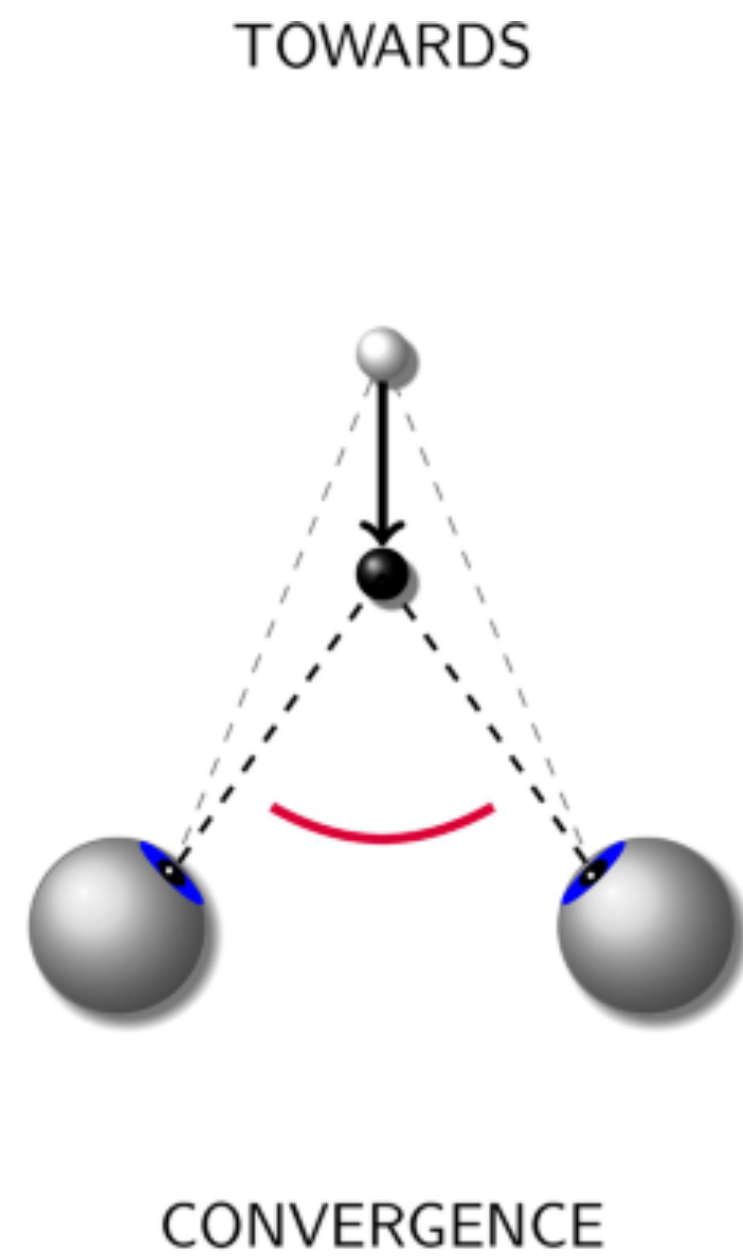


Large
disparity

The stereo disparity drives the vergence state of the eyes.

Vergence

Rotating eye balls so that images are formed on the fovea (highest acuity).



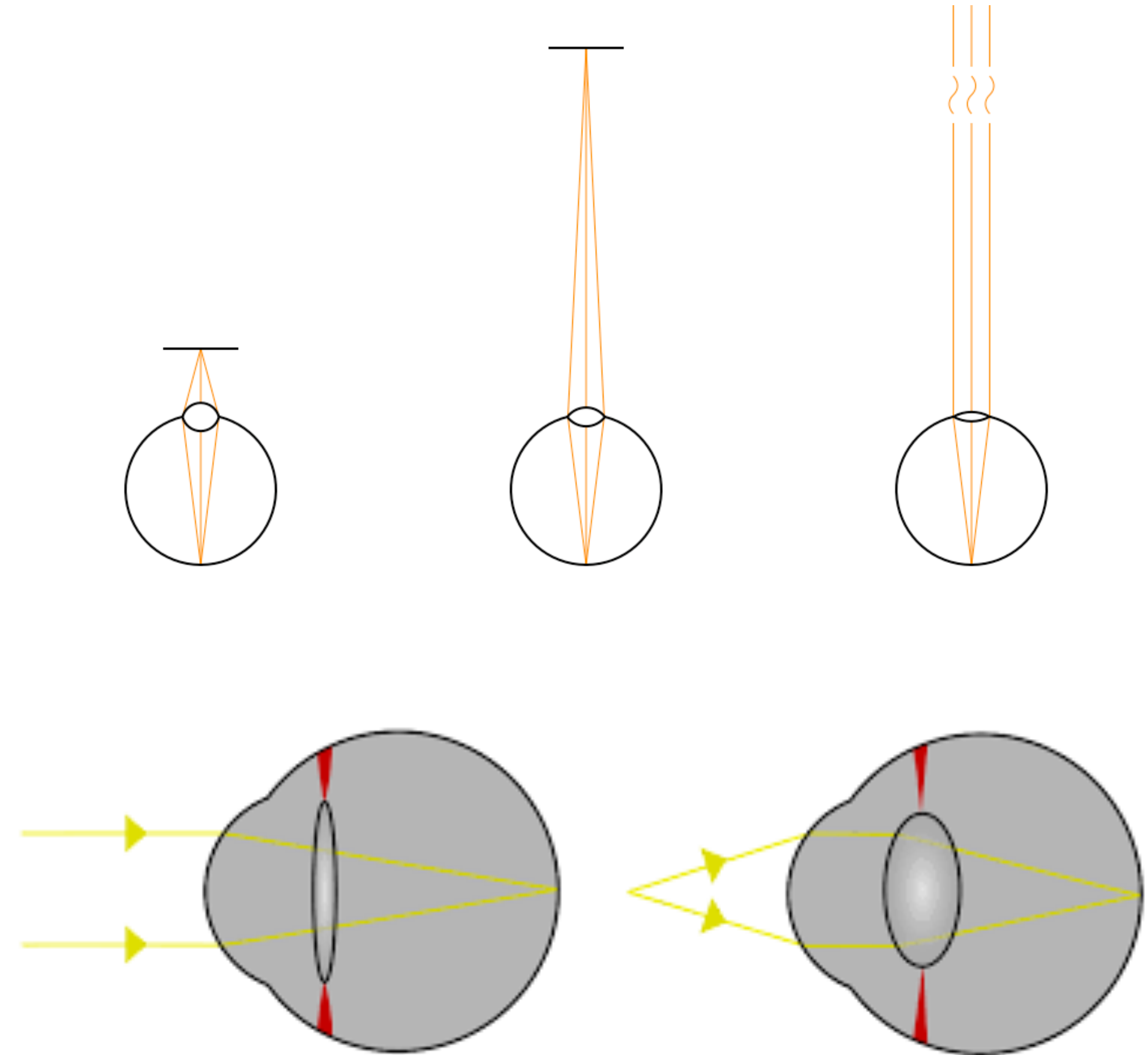
Accommodation

Vergence itself isn't enough. The object on the fovea might be blurred because it's out-of-focus.

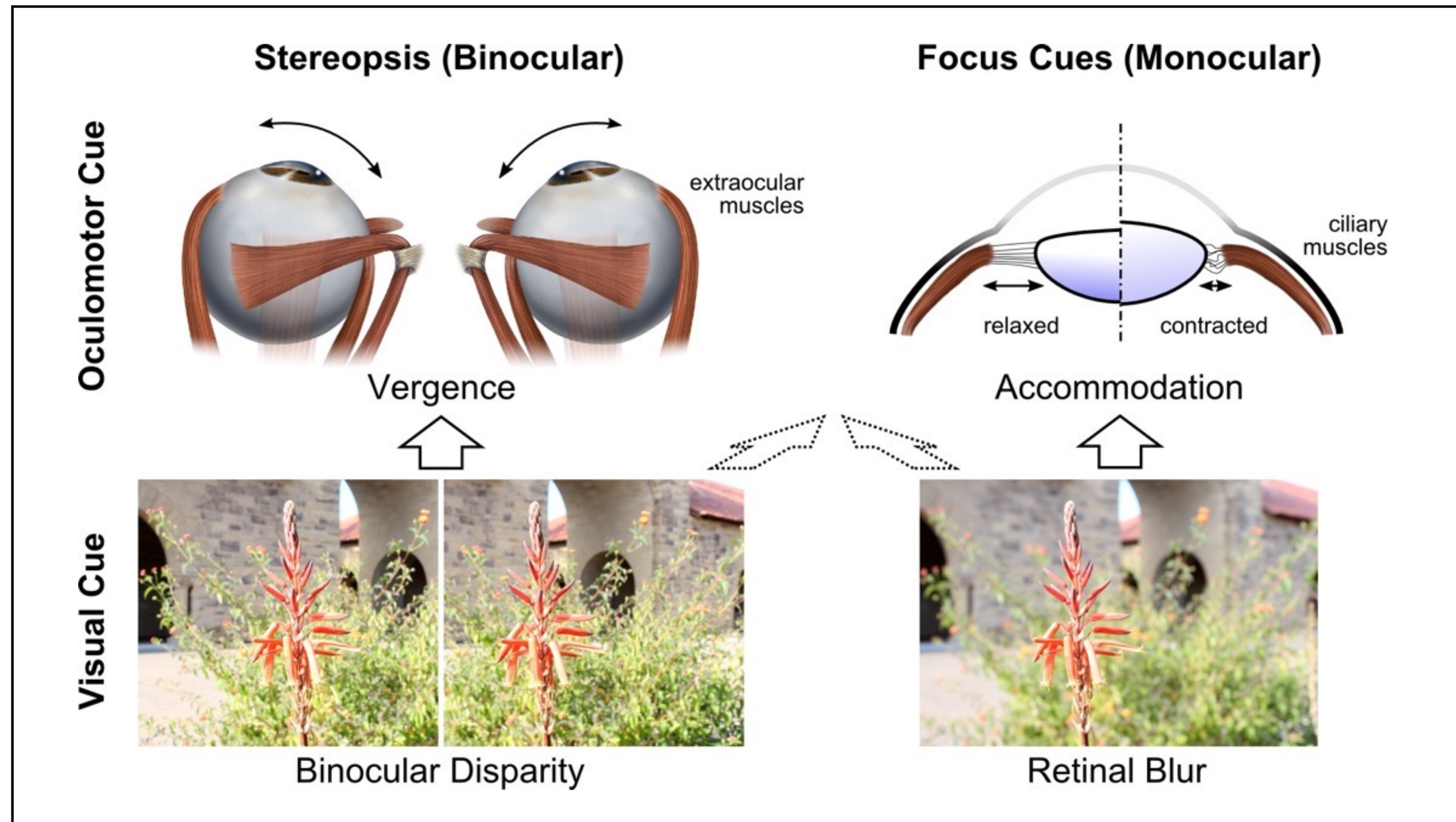
Change eye len's convexity (refractive power) so that the object on the fovea is sharp.

- Controlled by ciliary muscle inside eye.

Retinal (foveal) blur drives accommodation.



Vergence and Accommodation are Coupled



Gordon Wetzstein

Focus on far objects:

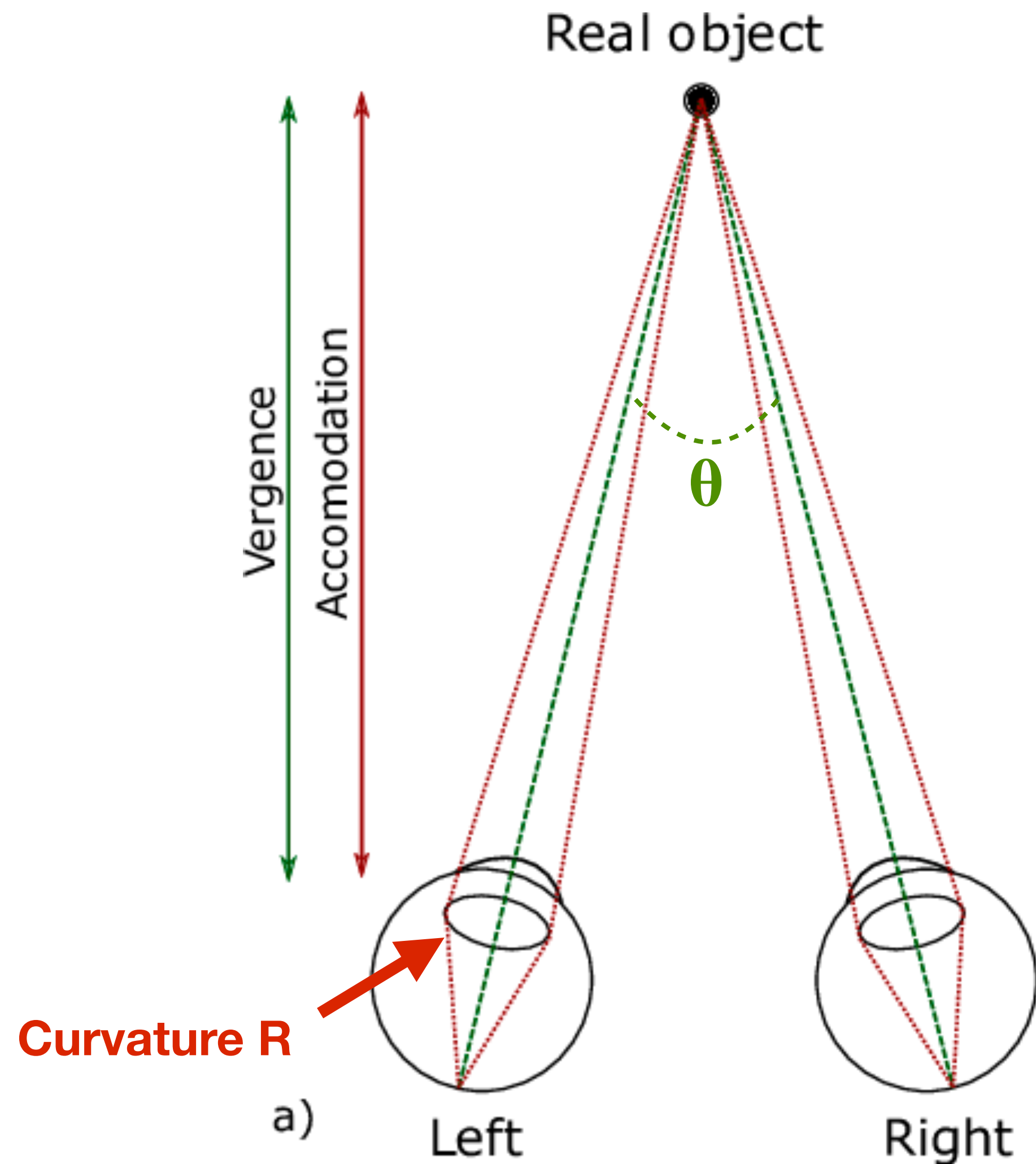
- Diverge
- Relax lenses

Focus on near objects:

- Converge
- Contract lenses

It's a reflex action
(accommodation-
convergence reflex)

Vergence and Accommodation are Coupled



Object distance dictates:

- the stereo disparity, which drives the vergence state θ .
- the retinal blur, which drives the accommodate state **R**.
- θ and **R** are correlated.

Distance decreases (increases):

- θ increases (decreases)
- **R** increases (decreases)

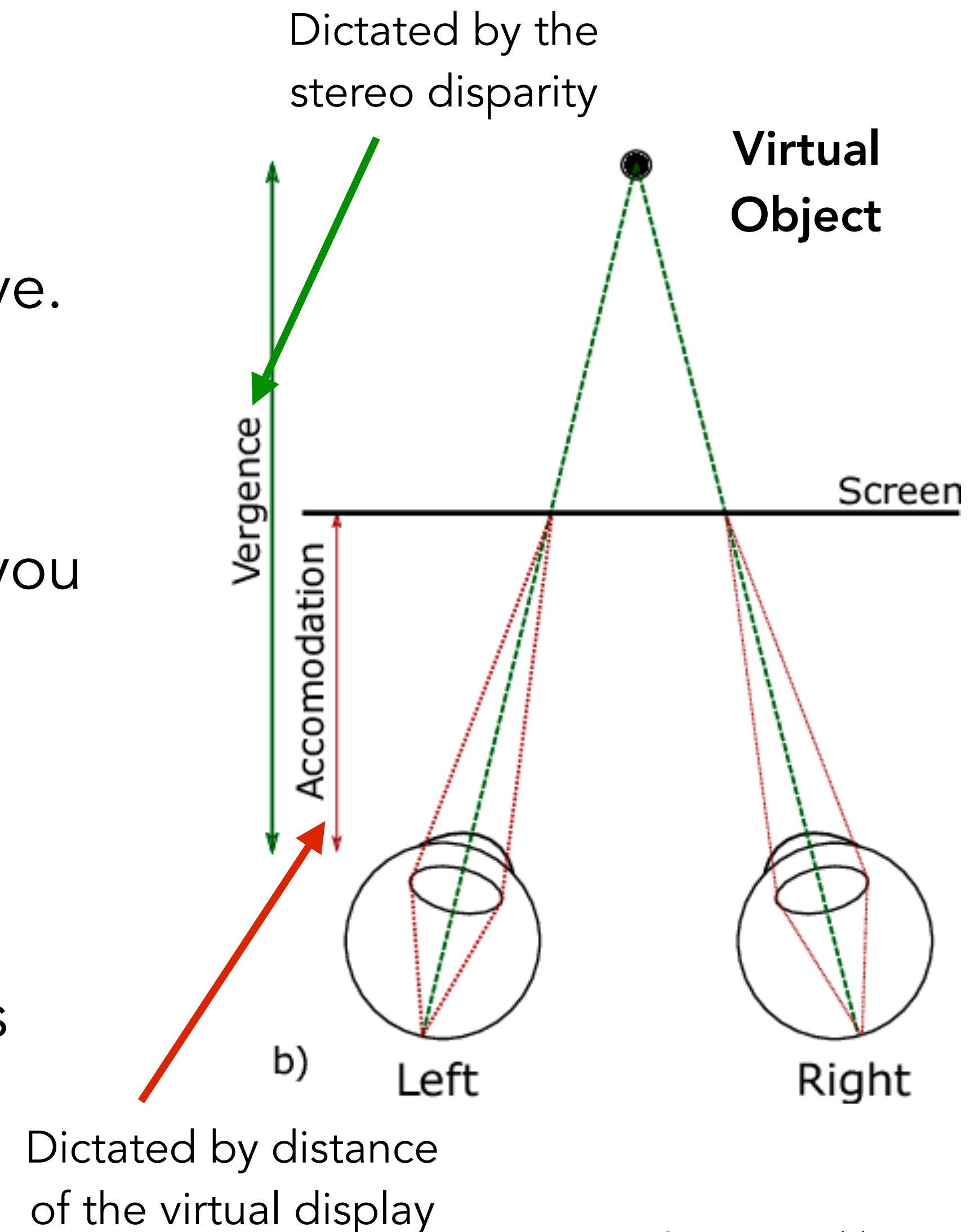
Problem with VR: Vergence-Accommodation Conflict

Virtual objects at different distances are emitted from the same fixed distance!

- The physical display and the lenses don't move.
- The accommodation distance (e.g., 1.4 m in Oculus Rift DK2) is fixed.
- So the accommodation state **won't** change as you focus on different objects at different depths.

But the stereo disparities of different objects do change with the depths.

- Stereo rendering generates correct disparities
- So your vergence state **does** change.

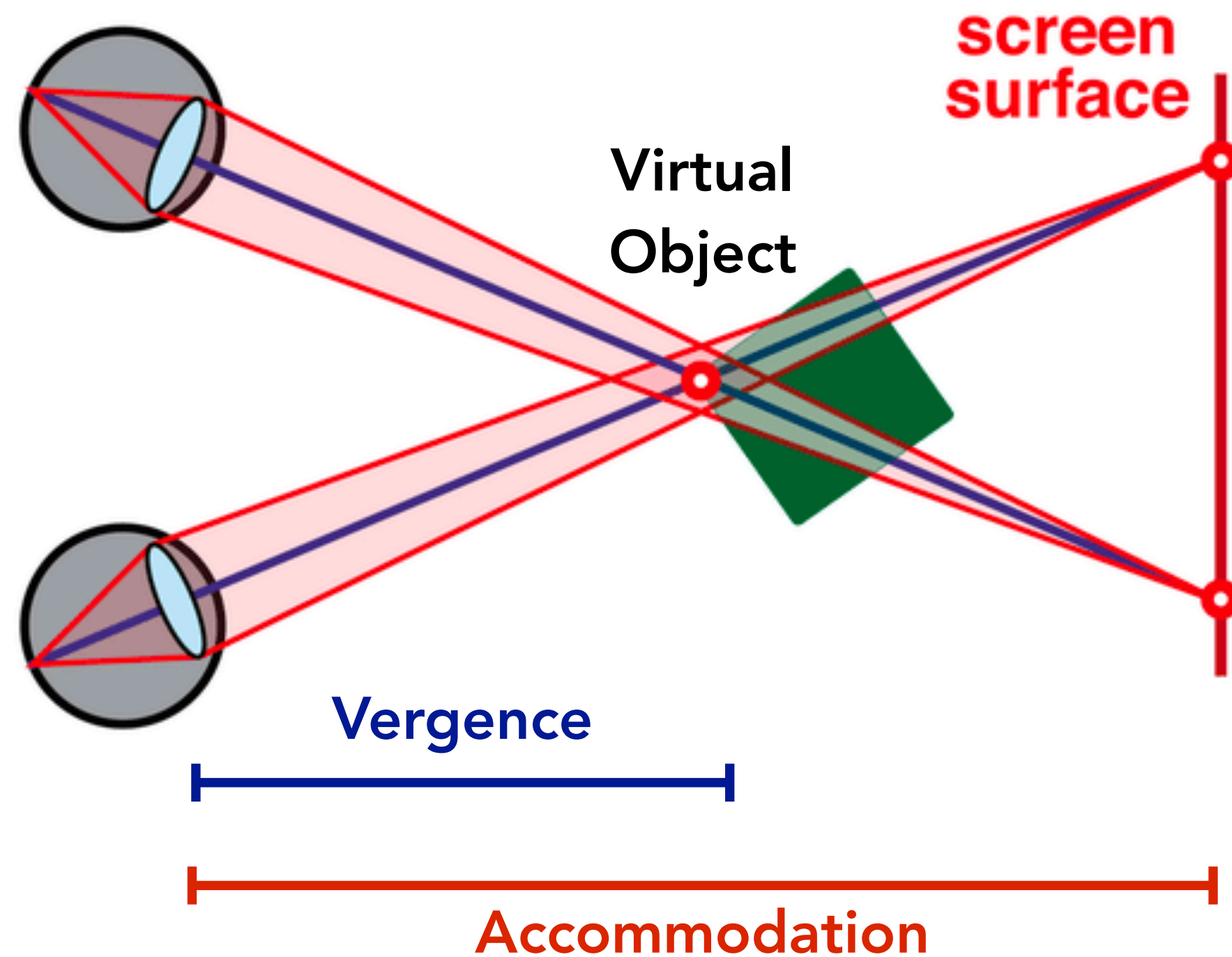


Problem with VR: Vergence-Accommodation Conflict

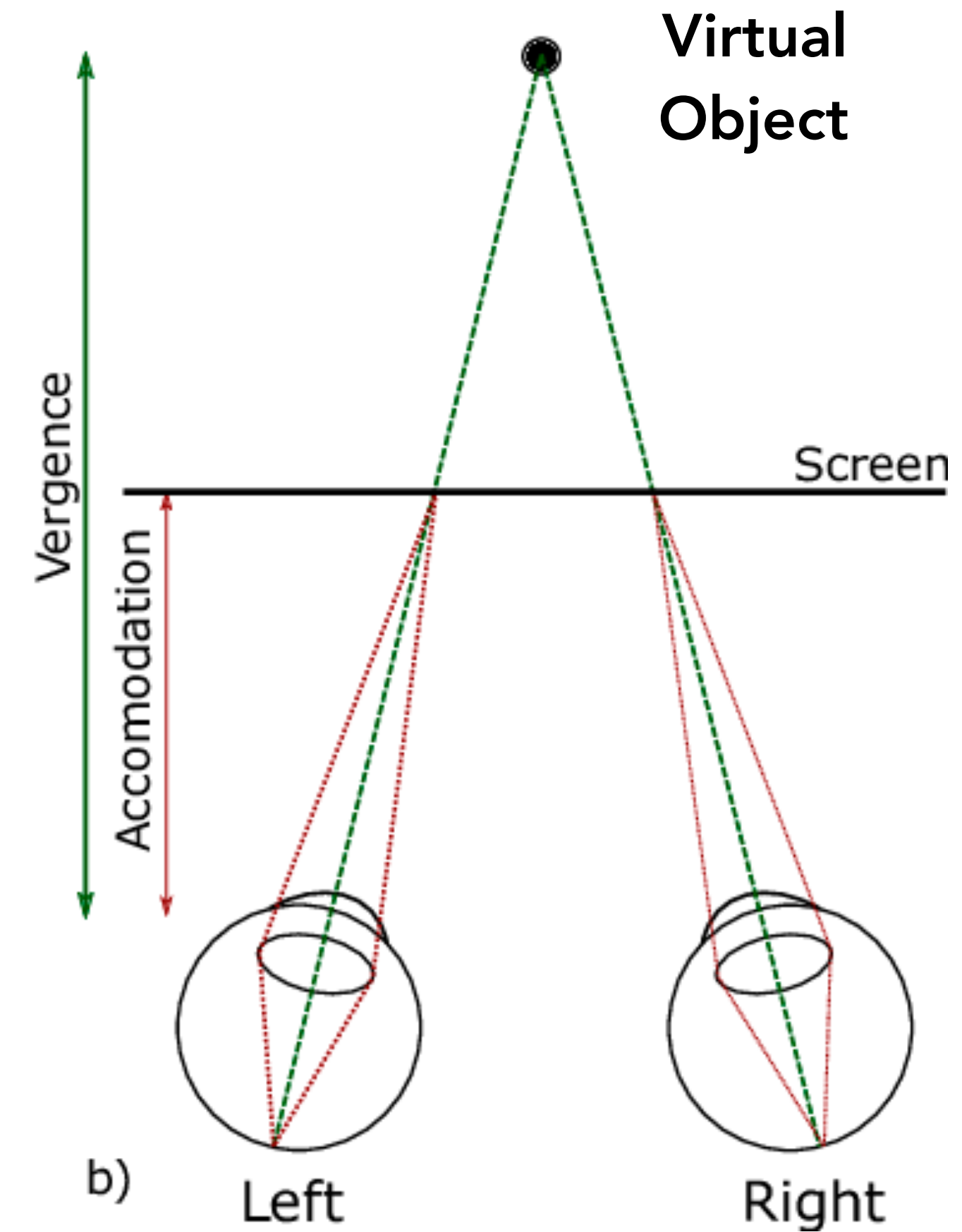
- Given the rendered stereo disparity of an object you want to focus on, your eyes will settle for a vergence state, θ .
- Your brain will try to impose the accommodate state R coupled with θ .
- But that R won't focus on the virtual object. The object is now actually blurred. Fighting the VAC creates discomfort and fatigue.
- Given enough time, your brain will learn to decouple vergence with accommodation, but when going back to the real world, you get the discomfort again.

Two Cases

Accommodation Distance > Vergence Distance.
Looking at object close up. More severe.



Vergence Distance > Accommodation Distance. VAC
less of an issue, because accommodation cue is less important for farther objects.



Light-Field Display

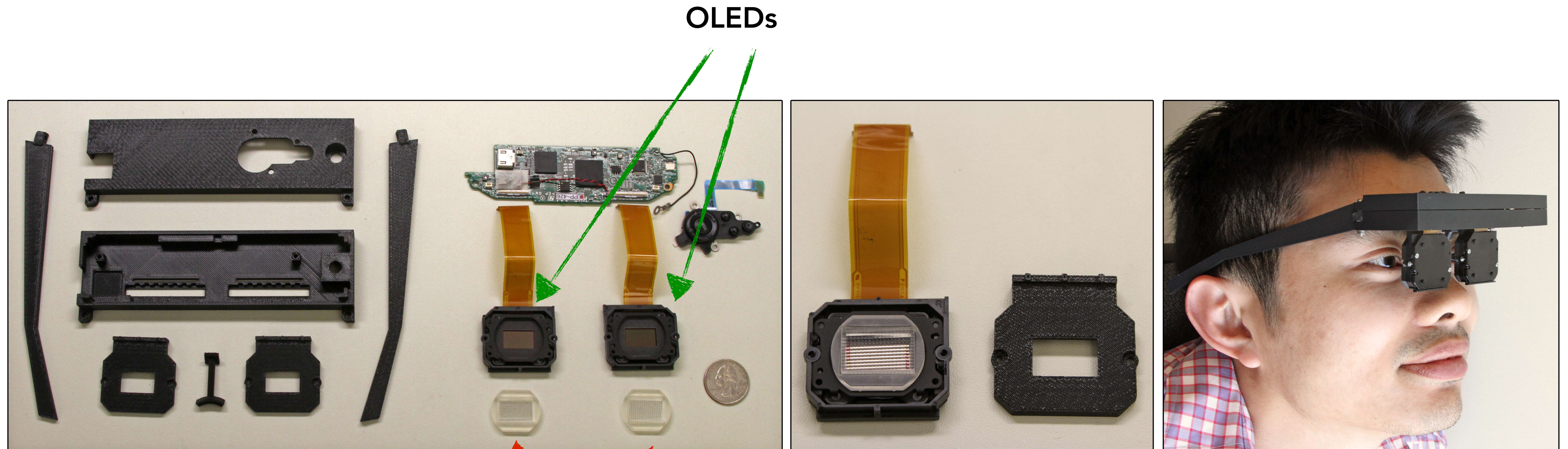
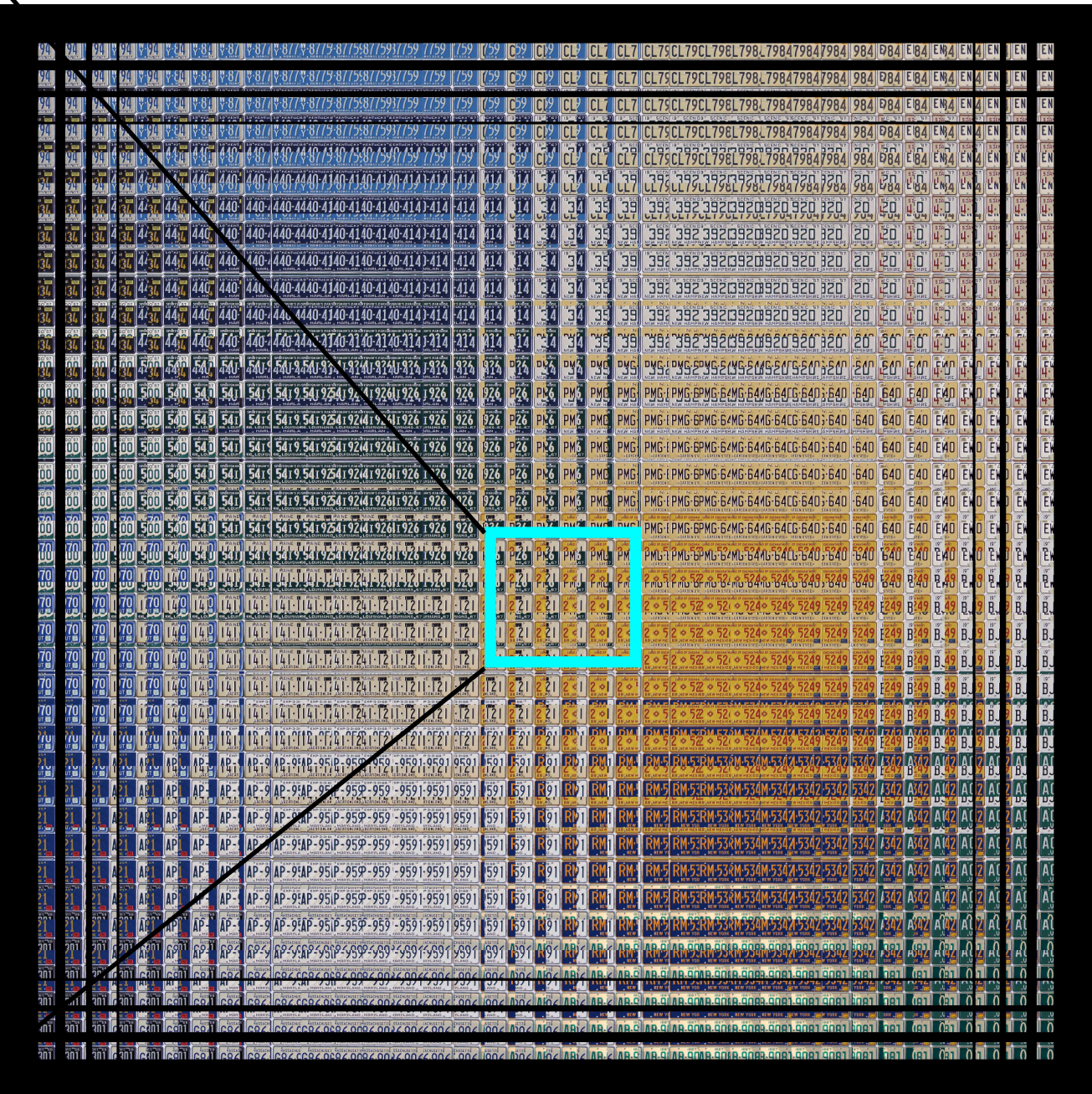
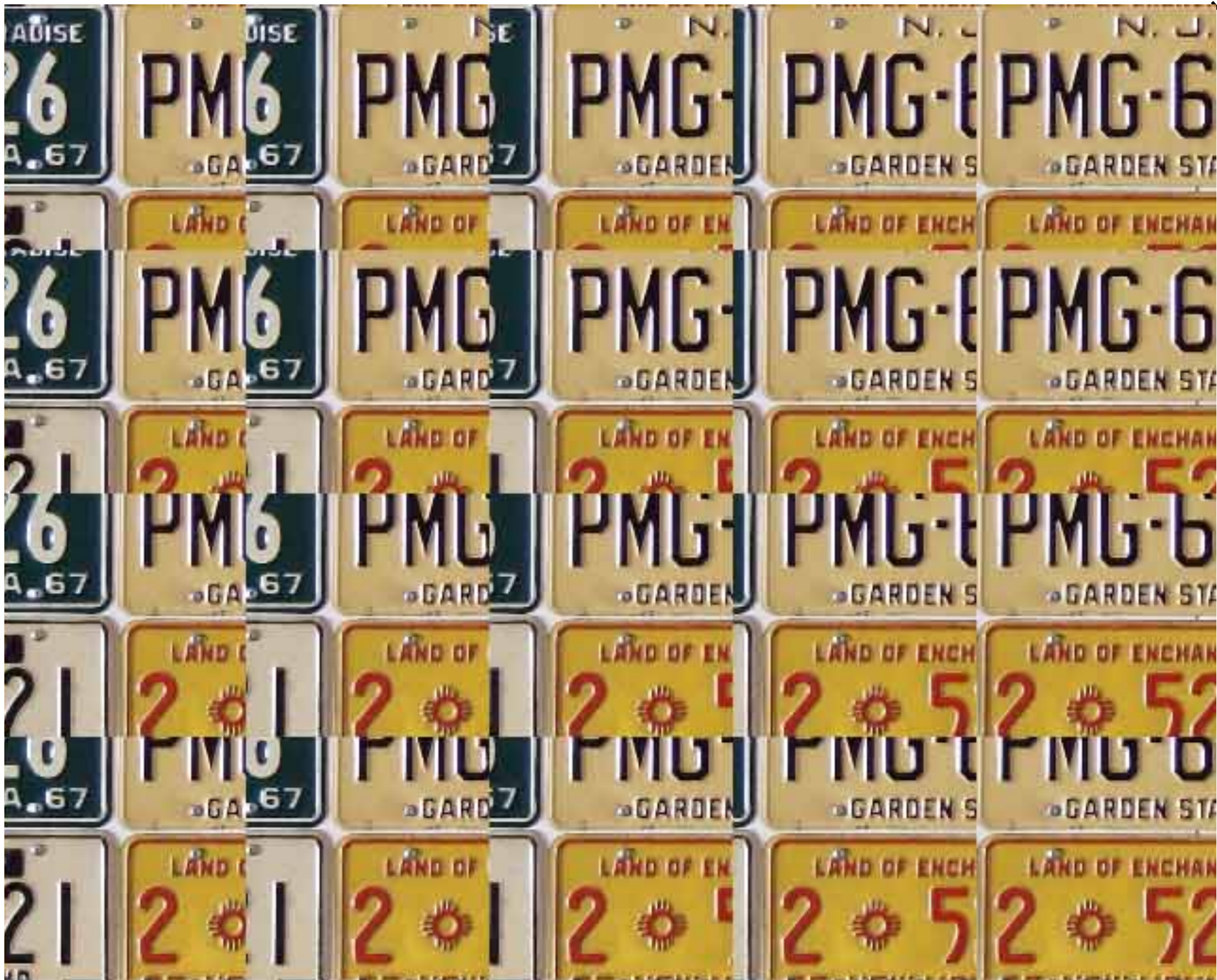
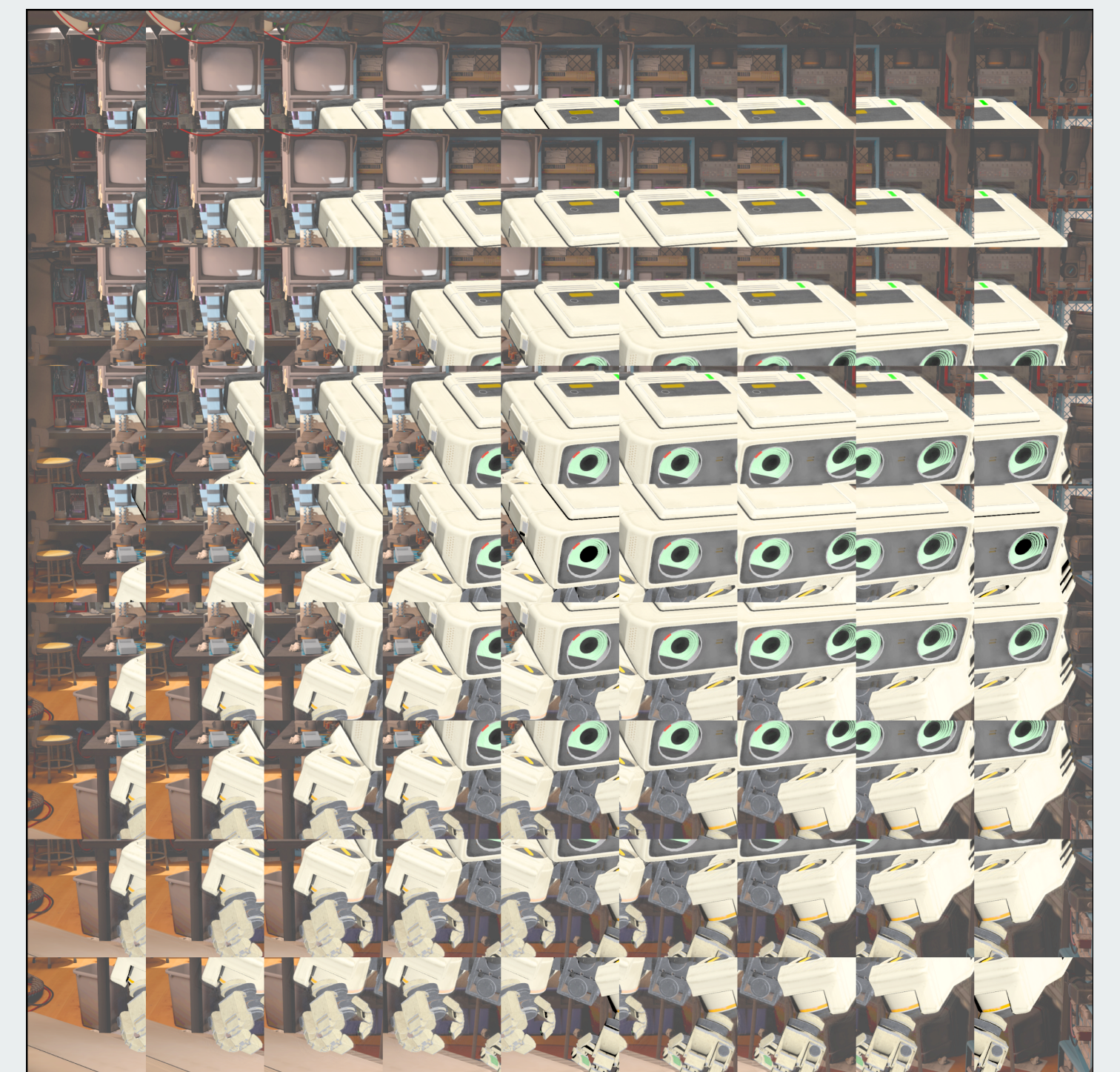


Figure 7: Constructing a near-eye light field HMD. (Left) A custom head-mounted enclosure, comprising the plastic parts shown on the left, was fabricated to hold the Sony HMZ-T1 driver electronics and our modified eyepieces, shown on the right. (Middle) Each modified eyepiece contains a Fresnel Technologies #630 microlens array, mounted in front of a Sony ECX332A OLED microdisplay. (Right) A user wearing the assembled HMD. (See Appendix A of the supplementary material for an extended discussion of the prototype construction.)

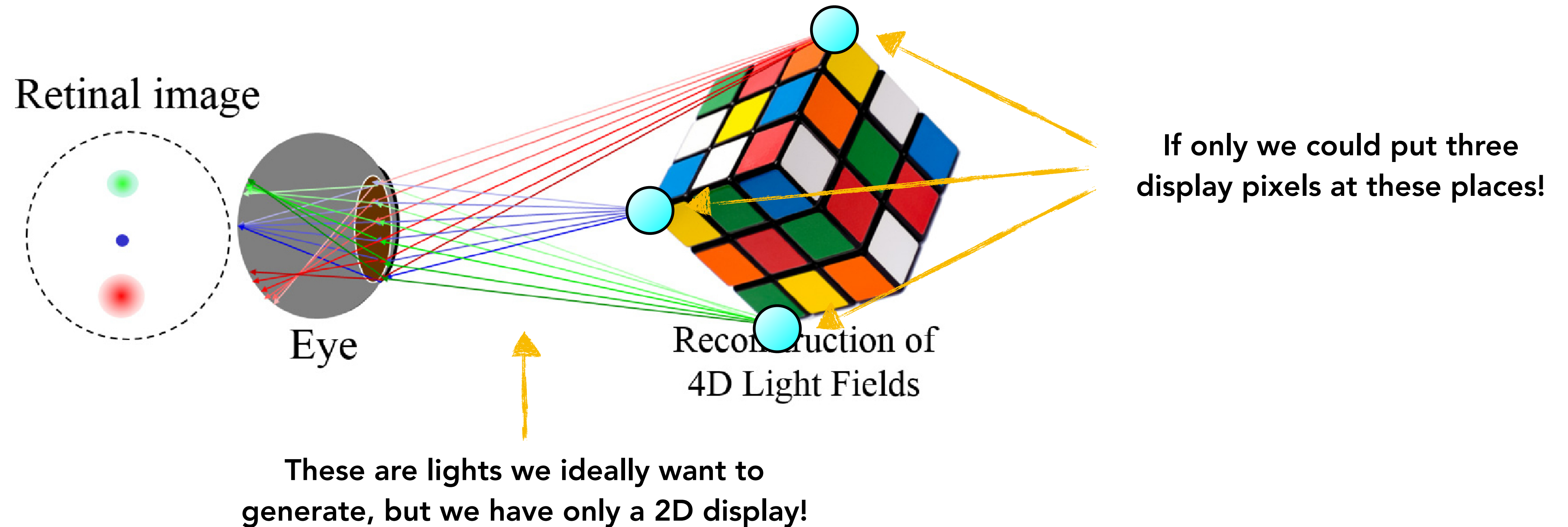
Light-Field Display



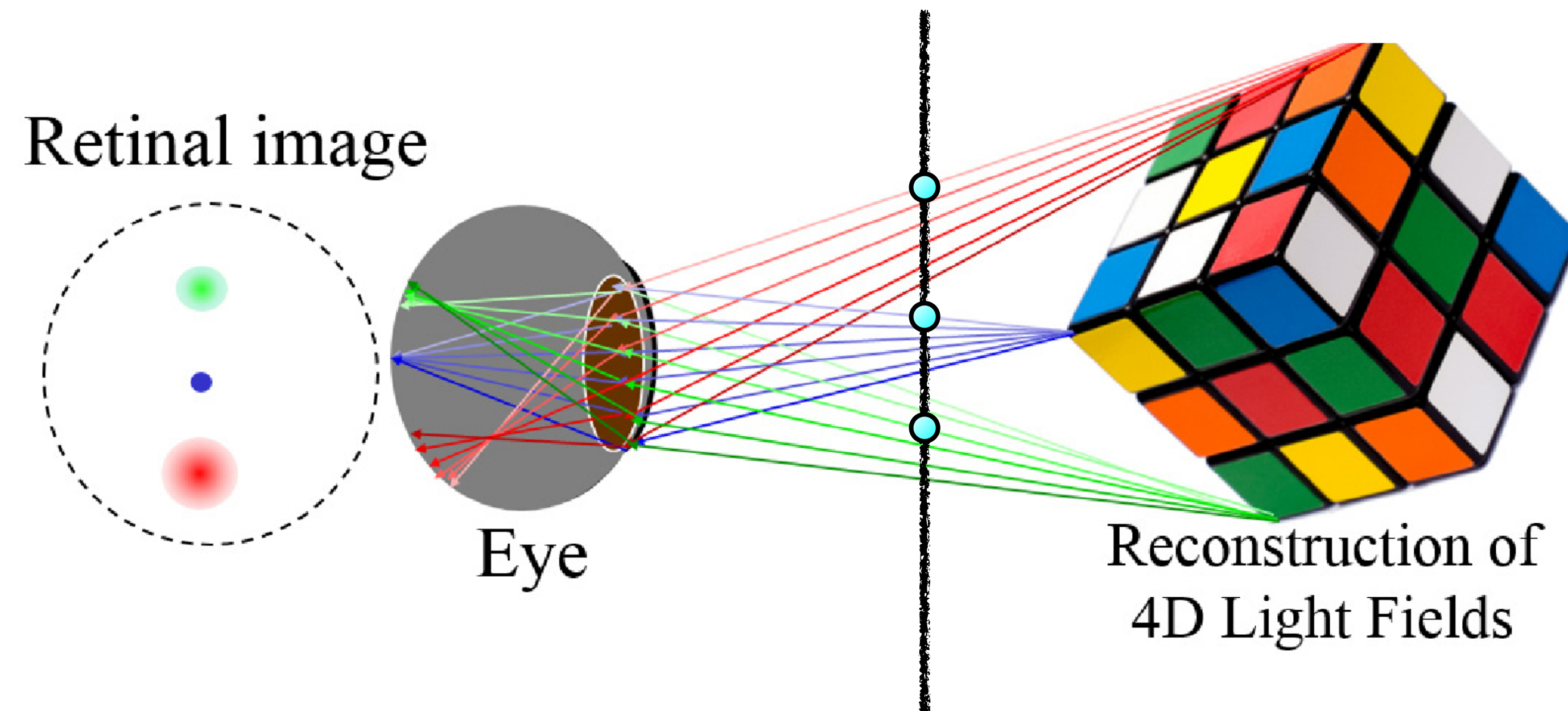
Light-Field Display



Goal: Reproducing 4D Light Field from a 2D Display



Goal: Reproducing 4D Light Field from a 2D Display

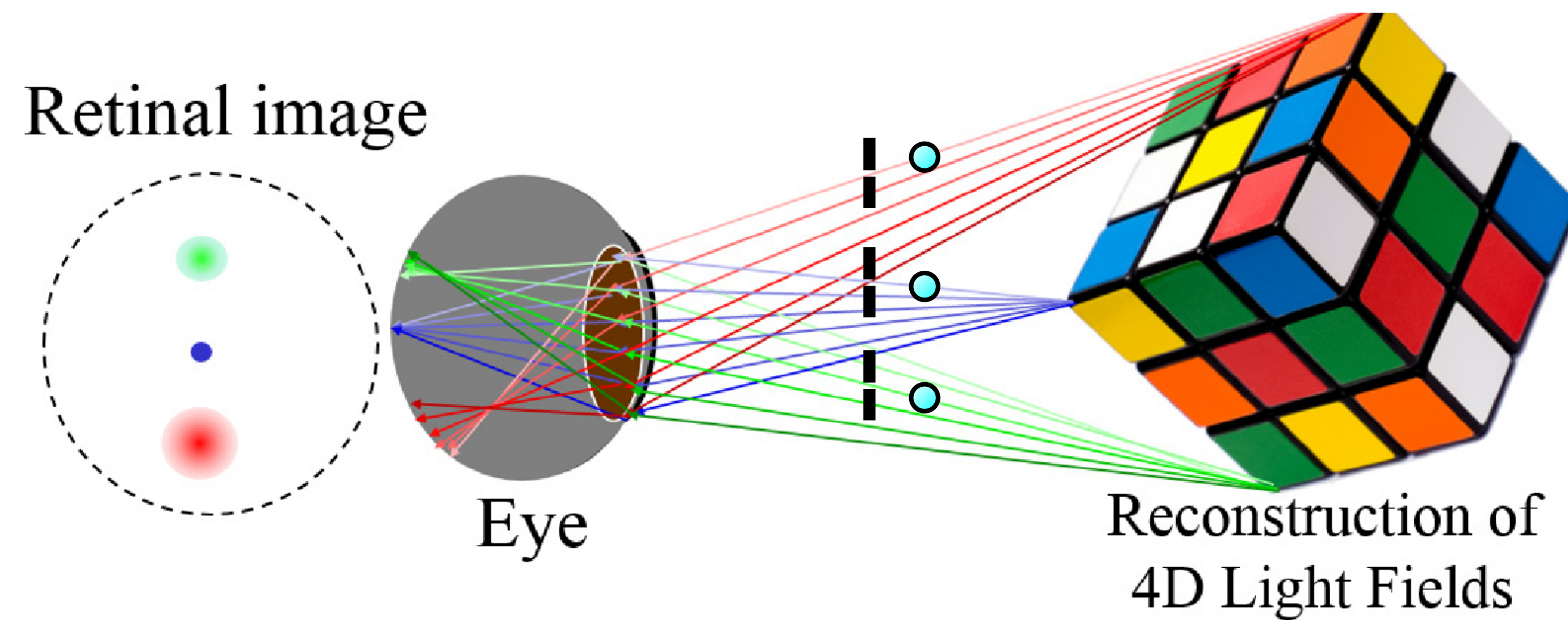


As long as each display pixel emits lights in the desired direction, we can reproduce the lightfield.

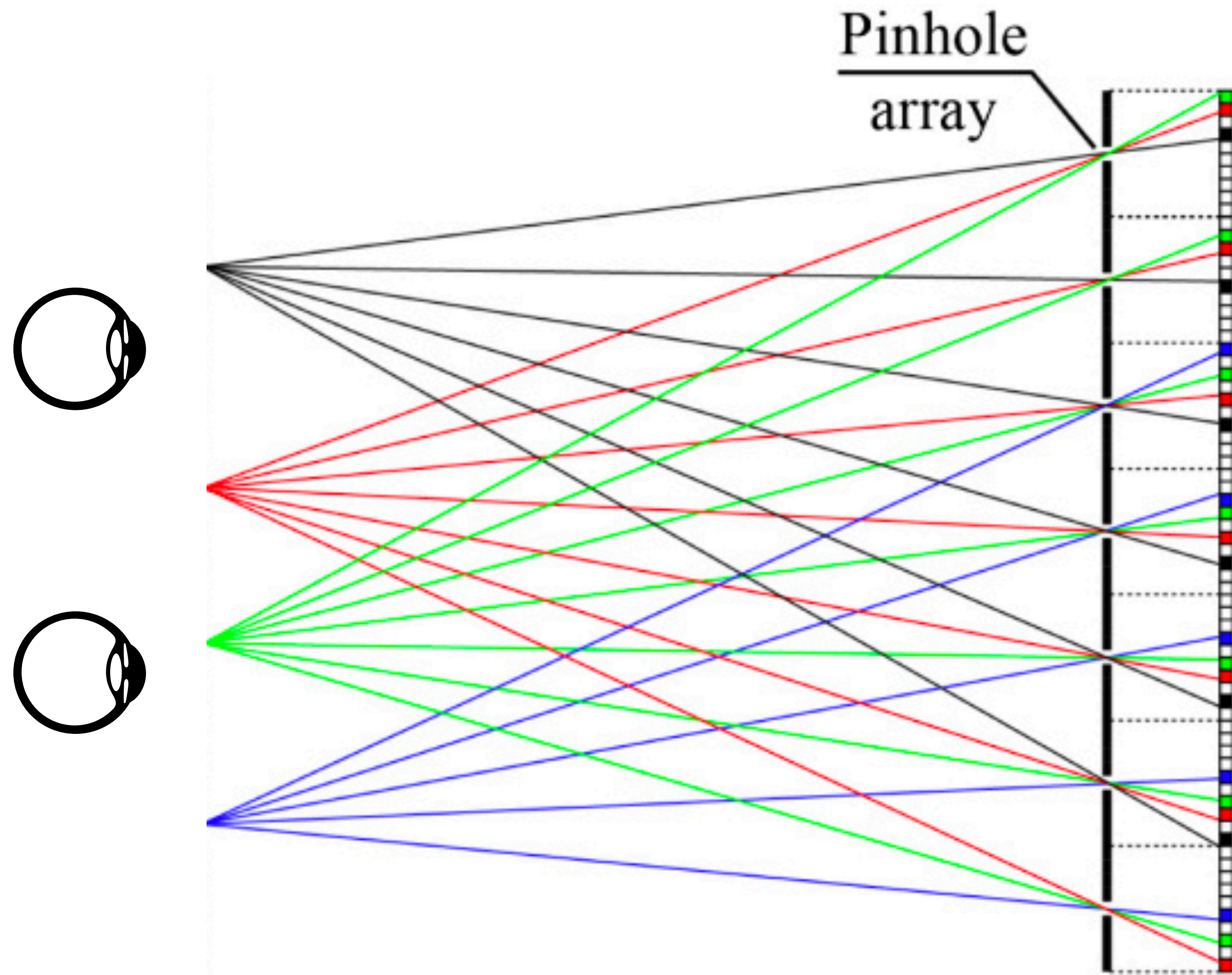
But each pixel emits lights to the hemisphere. How can we control the direction of each pixel?

Generating Light Field Using Pinhole Array

Put a pinhole array in front of the display!



Generating Light Field Using Pinhole Array



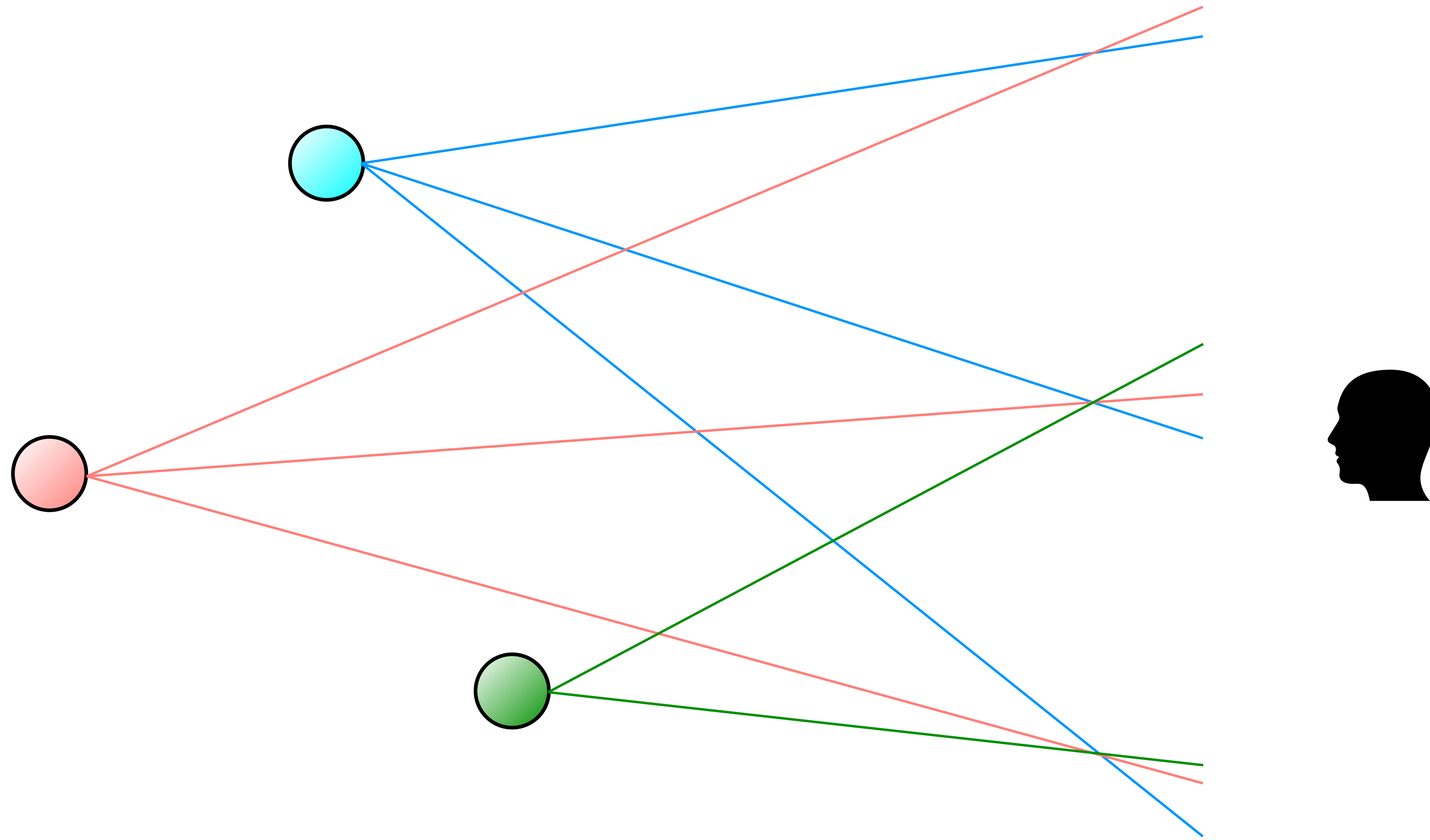
Put a pinhole array in front of the display!

Two steps:

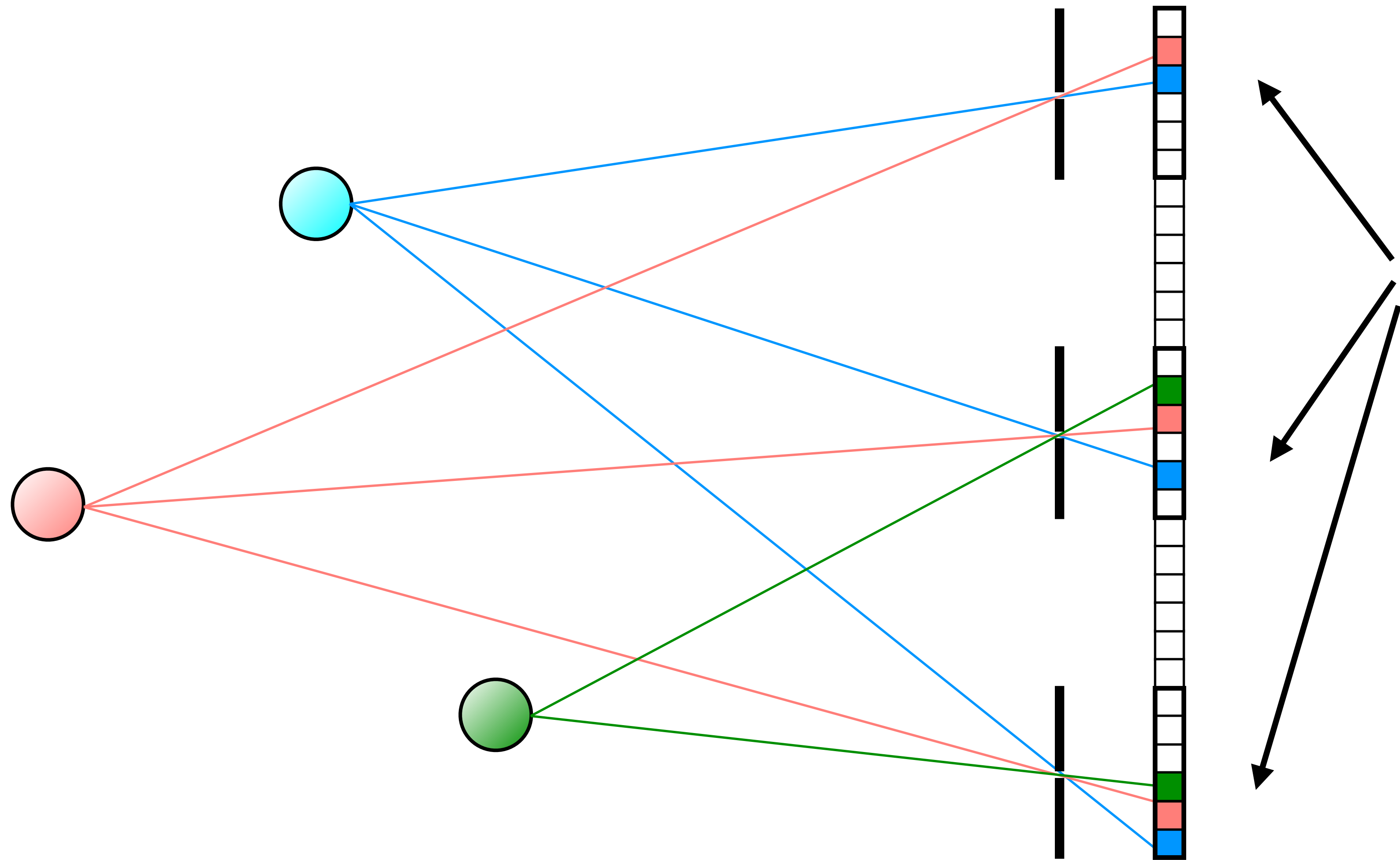
- Record the light field.
- Reproduce the light field.

Both can be done using the pinhole array.

Step 1: Recording Light Field Using Pinhole Array



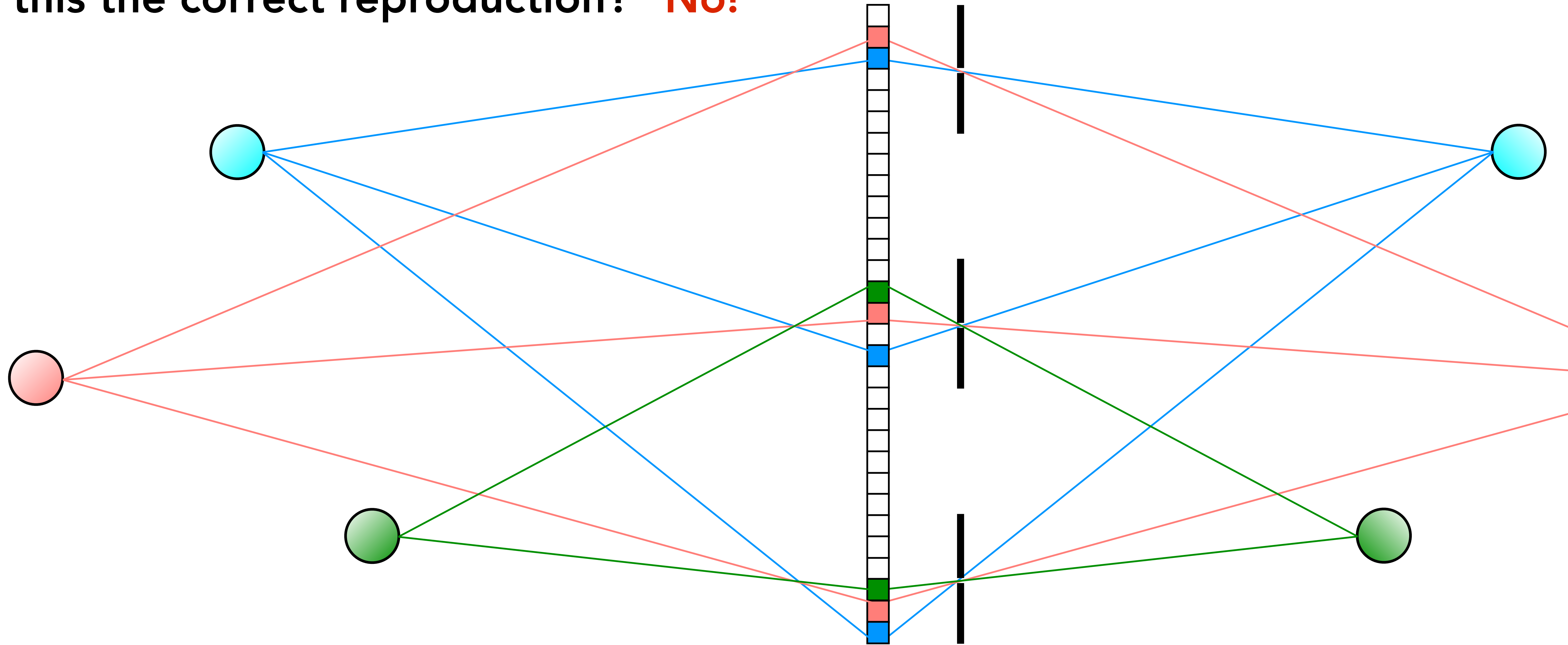
Step 1: Recording Light Field Using Pinhole Array



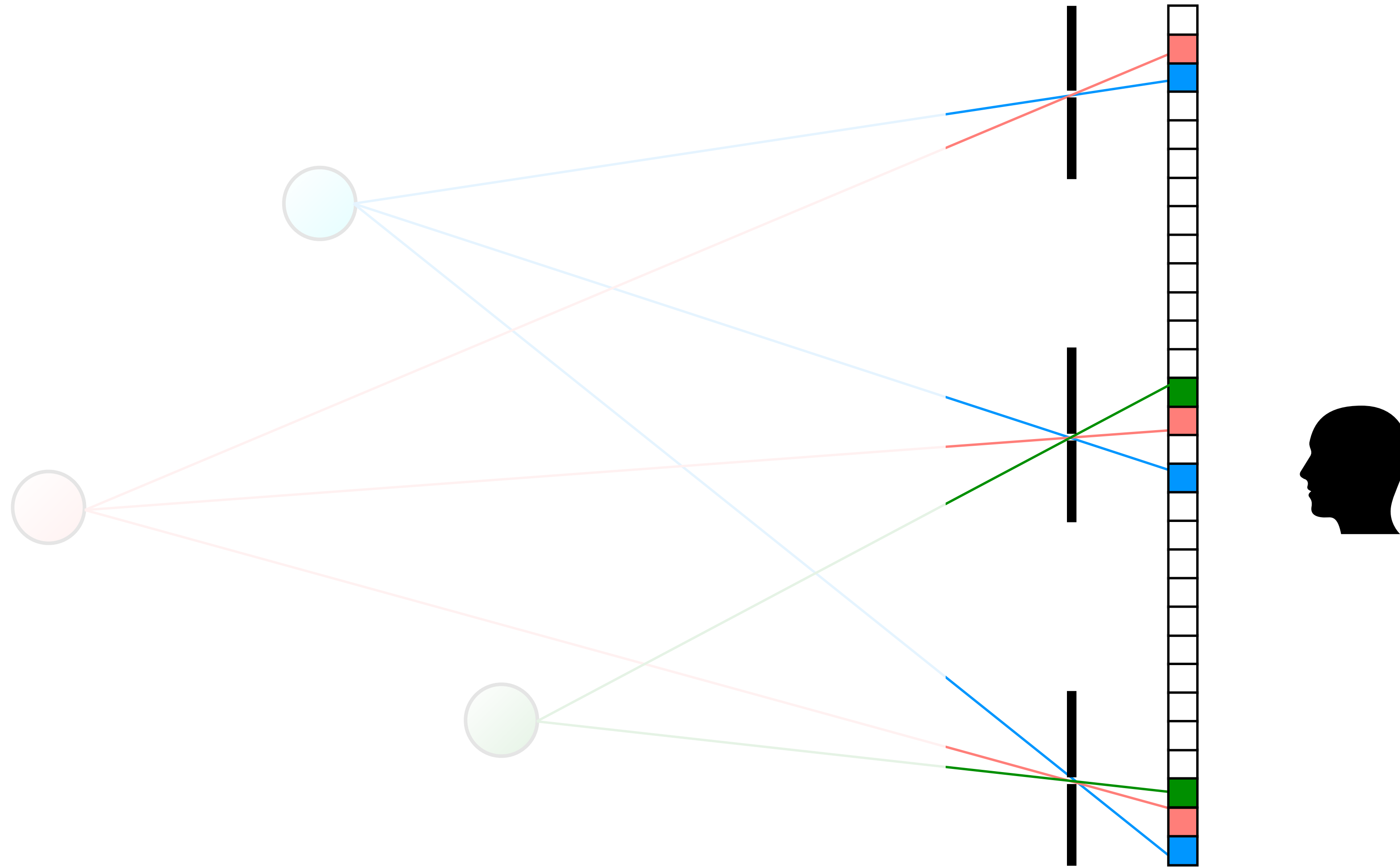
These "elemental images" are essentially photos of the scene taken at different perspectives. Note how these elemental images are not exactly the same.

Step 2: Reproducing Light Field Using Pinhole Array

Is this the correct reproduction? **No!**

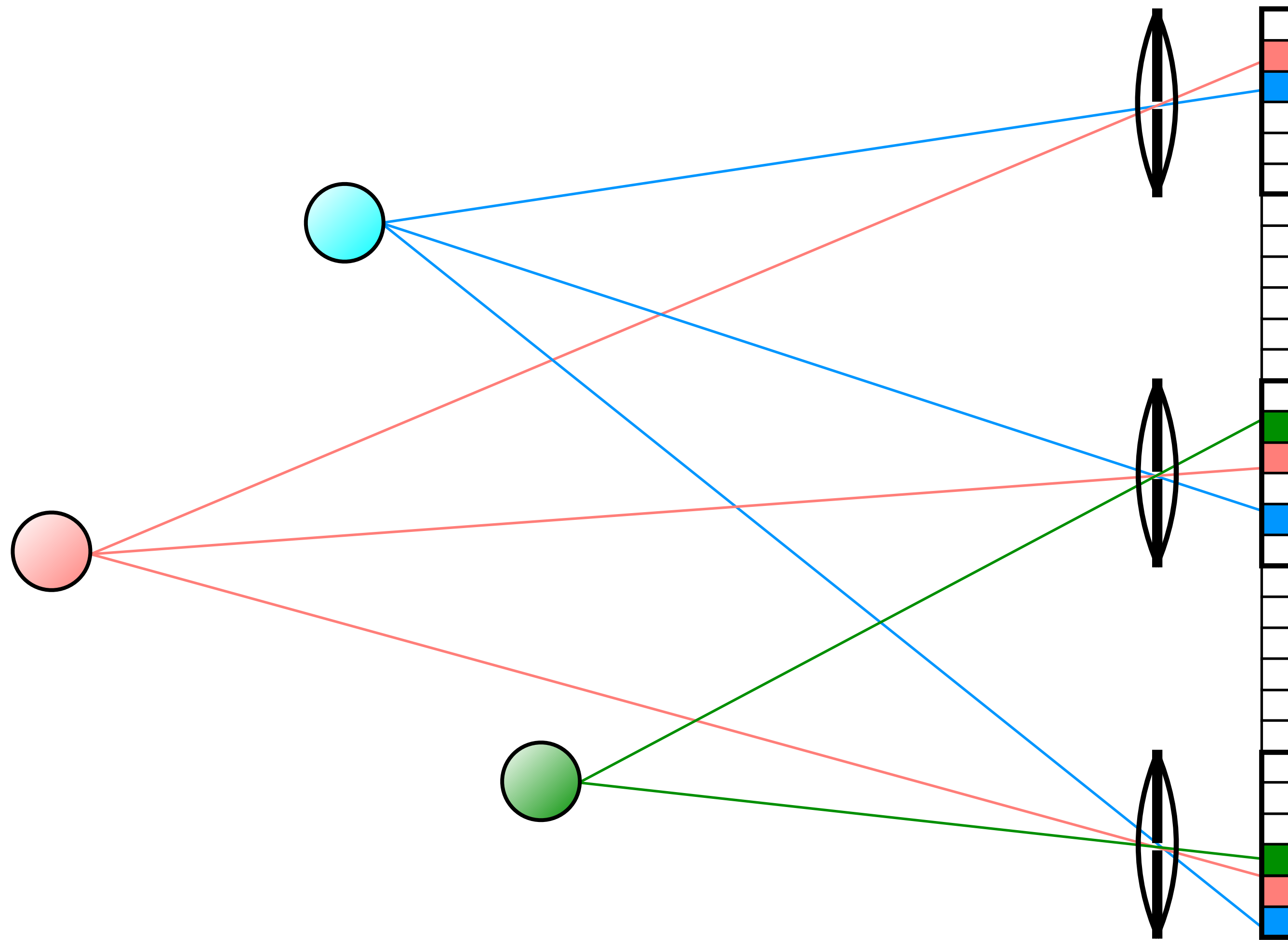


Step 2: Reproducing Light Field Using Pinhole Array



**Correct
reproduction!**

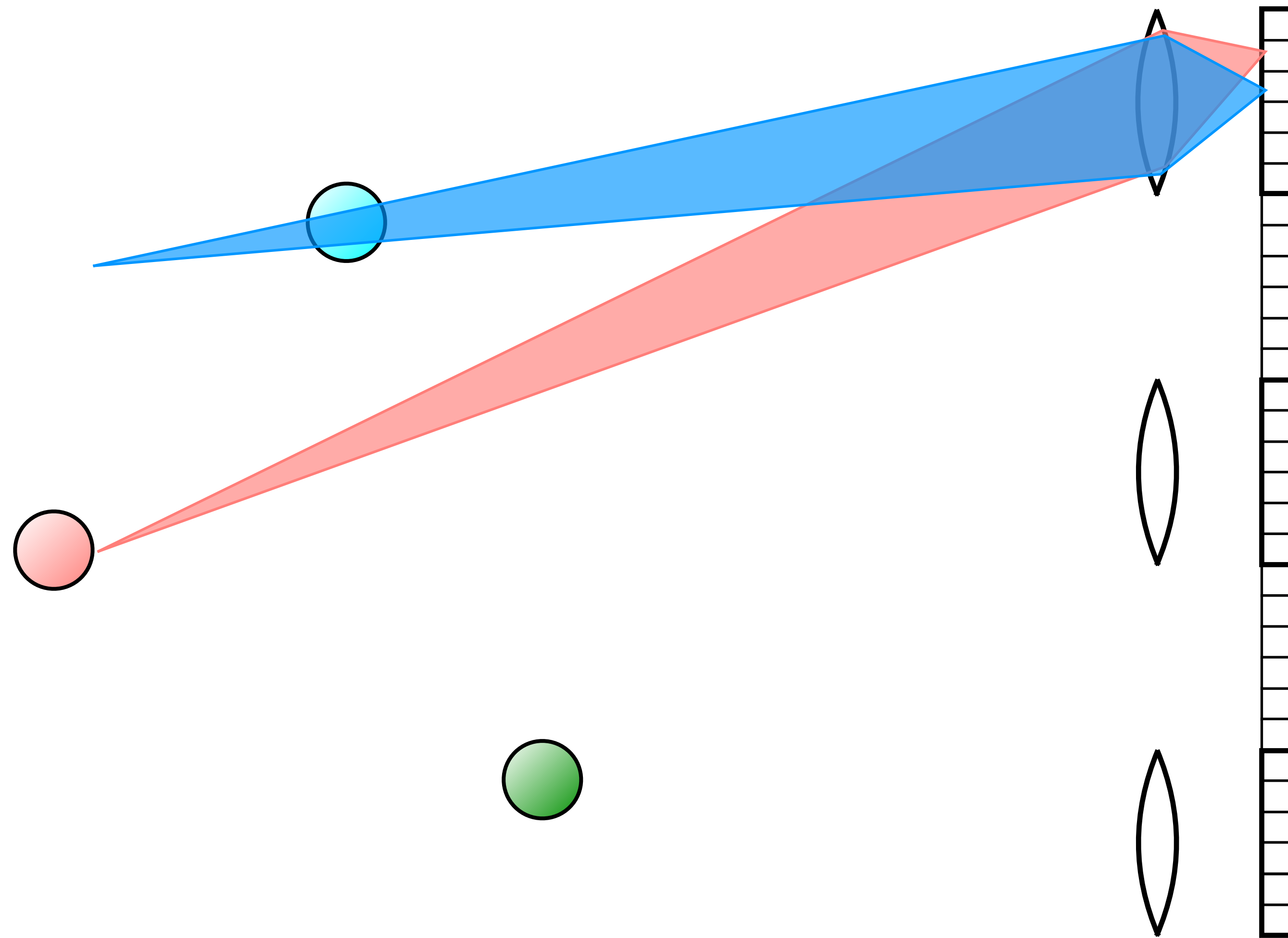
Recording Light Field Using Microlens Array



Disadvantages: Pinholes are very small, which increases noise and gives rise to diffraction effects.

Replace each pinhole with a microlens.

Recording Light Field Using Microlens Array



Disadvantages: Pinholes are very small, which increases noise and gives rise to diffraction effects.

Replace each pinhole with a microlens.

Downside: the lightfield sampling has lower resolution.

Other Ways for Depth Cues for Accommodation

- Light-field displays
- Varifocal displays
- Multi-focal displays
- Holographic displays

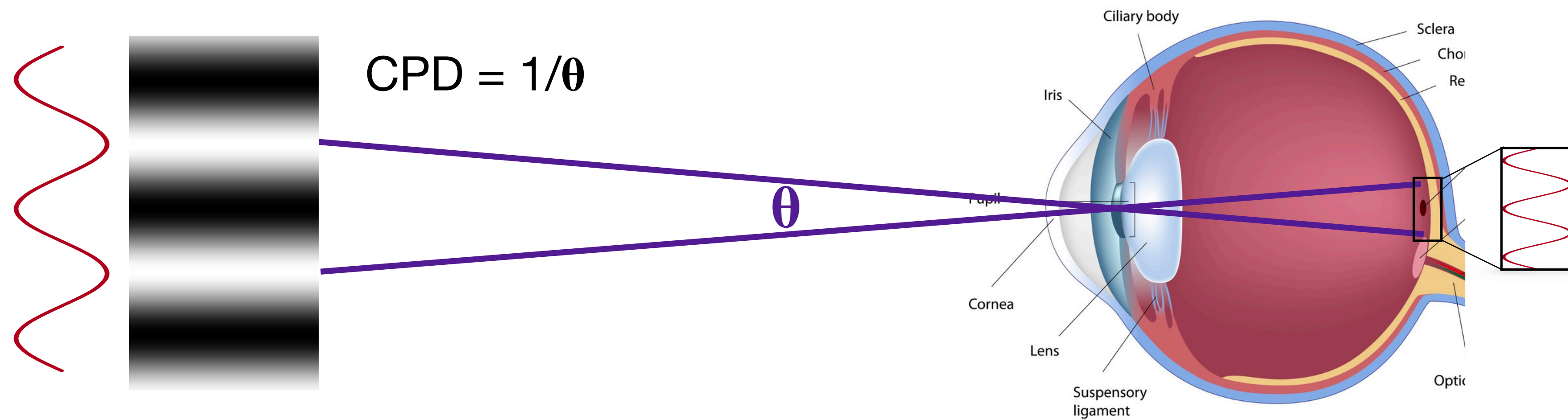
*Each eye's display is equipped with an actuation system to alter its depth position. **Stepper motors** with lead screws provide the linear motion, paired with crossed roller bearing slides as the guidance mechanism.*



VR Rendering

- Tracking
- Stereo rendering (vergence-accommodation conflict)
- **Foveated rendering (+eye tracking)**

Recall: Spatial Resolution of Human Visual System

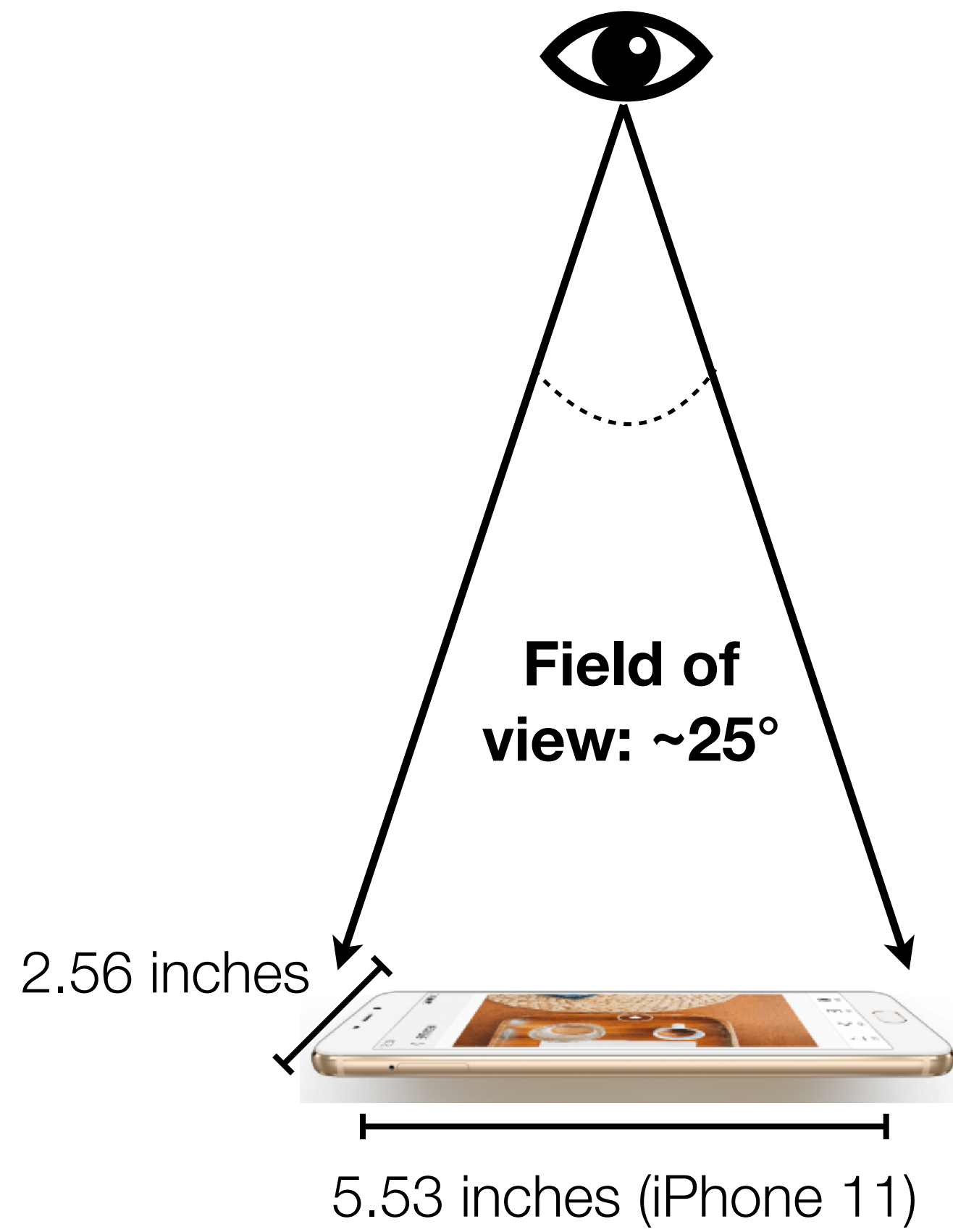


Assuming the display displays alternating black and white pixels. One pair of black and white is one cycle.

Cycle per degree (CPD) quantifies the spatial resolution in the scene.

Human perception limit is 60 CPD (based on the sampling theory).

How Many Pixels to Render?



To approach human spatial resolution limit, the CPD should be at least 60.

Vertical FOV at a viewing distance of 12" is 25.0°

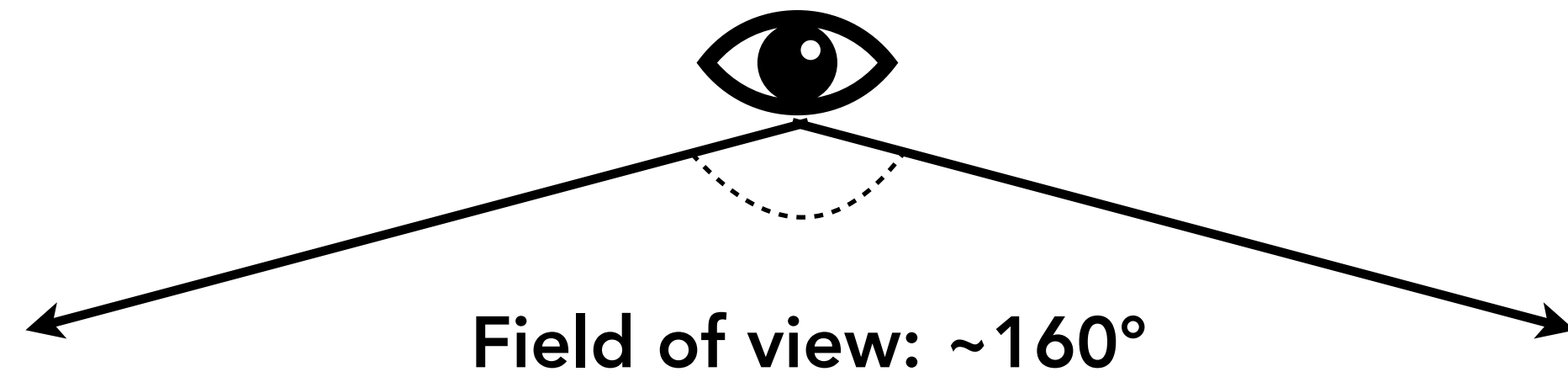
- $CPD = 25 * 60 = 1,500$
- # of pixels (height) = 3,000

Horizontal FOV is 12.2°

- $CPD = 12.2 * 60 = 732$
- # of pixels (width) = 1464

Actual iPhone 11 resolution: 1792 x 828

How Many Pixels to Render?



To maintain retinal resolution:

- assuming horizontal/vertical FOV is 160°
- # of pixels = $(160 * 60)^2 = 9600 \times 9600$
- If FOVs is 200°, # of pixels = 110K x 110K
- This is per eye!

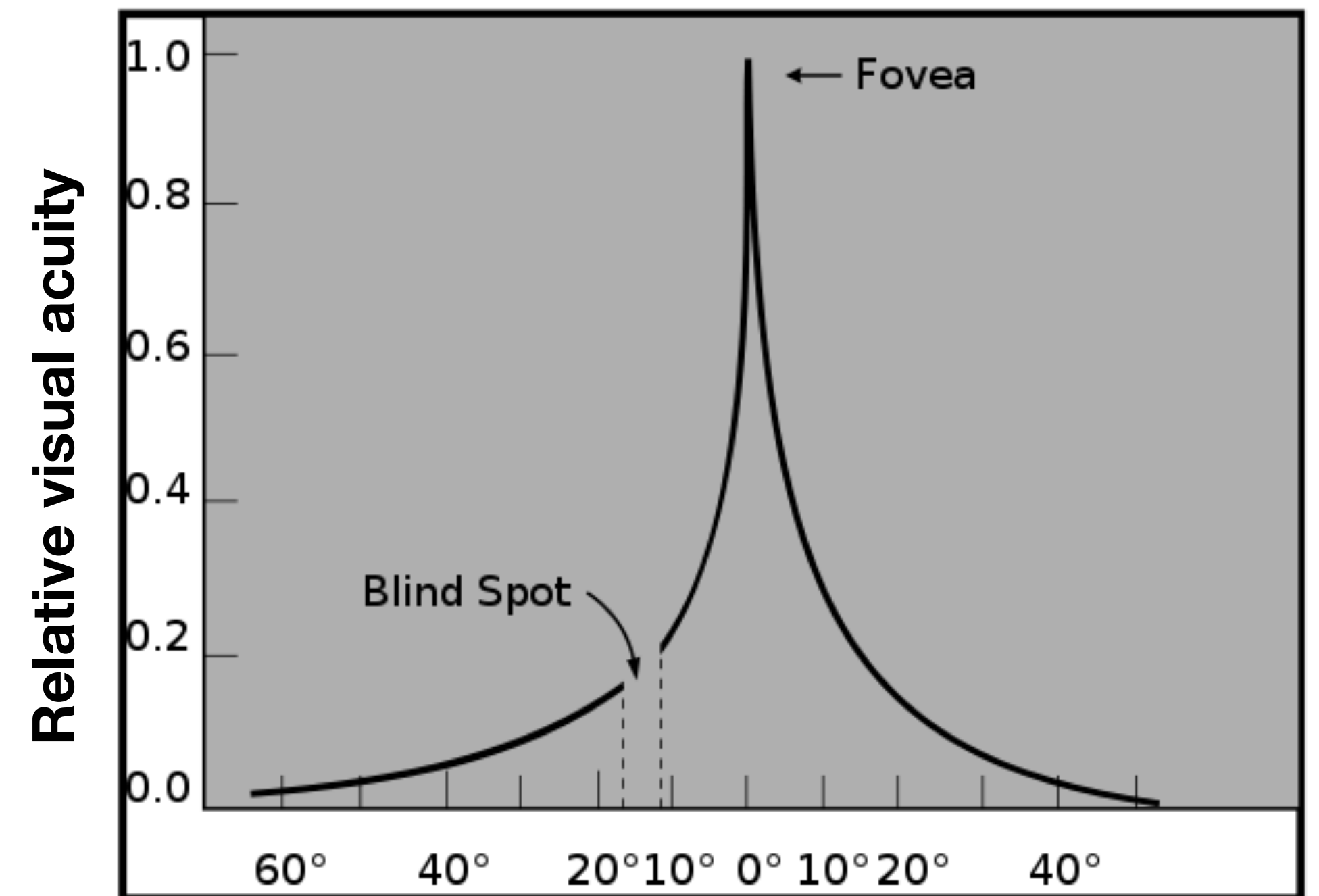
Today's VR headset resolution per eye:

- Oculus Quest 2: 1832 x 1920
- Valve Index: 1440 x 1600

Recall: Photoreceptor Density

The area on the retina with the highest cone density is called *fovea*.

Fovea angle is about 2° . Peripheral vision has very low acuity.





Fovea

Periphery

Periphery

Fovea

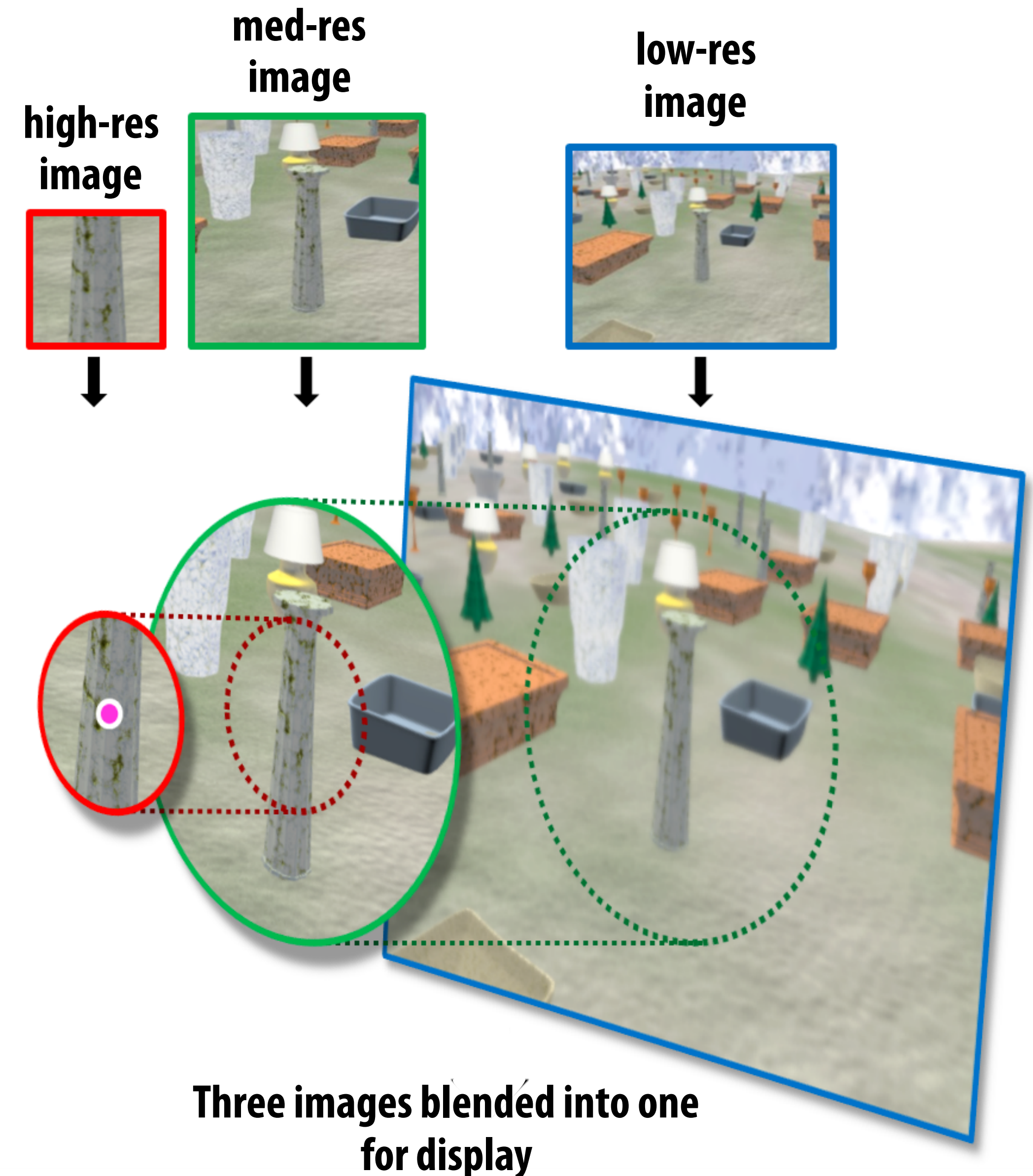


How?

Intuition: peripheral rendering can be of low quality (e.g., low resolution).

One common approach:

- Render with different resolutions.
- Stack and blend them together.
- Patch boundaries will have clear edge artifacts. Use low-pass filter.
- Enhance contrast as a final pass.
Empirically peripheral vision needs high contrast.



What to Render for Peripheral?

nature neuroscience

Explore content ▾ About the journal ▾ Publish with us ▾

nature > nature neuroscience > articles > article

Published: 14 August 2011

Metamers of the ventral stream

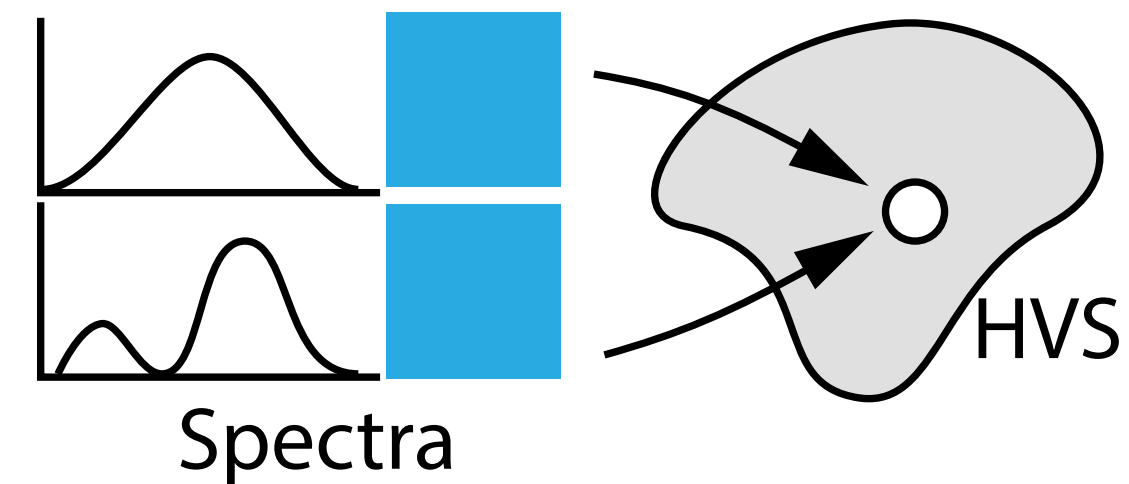
Jeremy Freeman [✉](#) & Eero P. Simoncelli

Nature Neuroscience 14, 1195–1201 (2011) | [Cite this article](#)

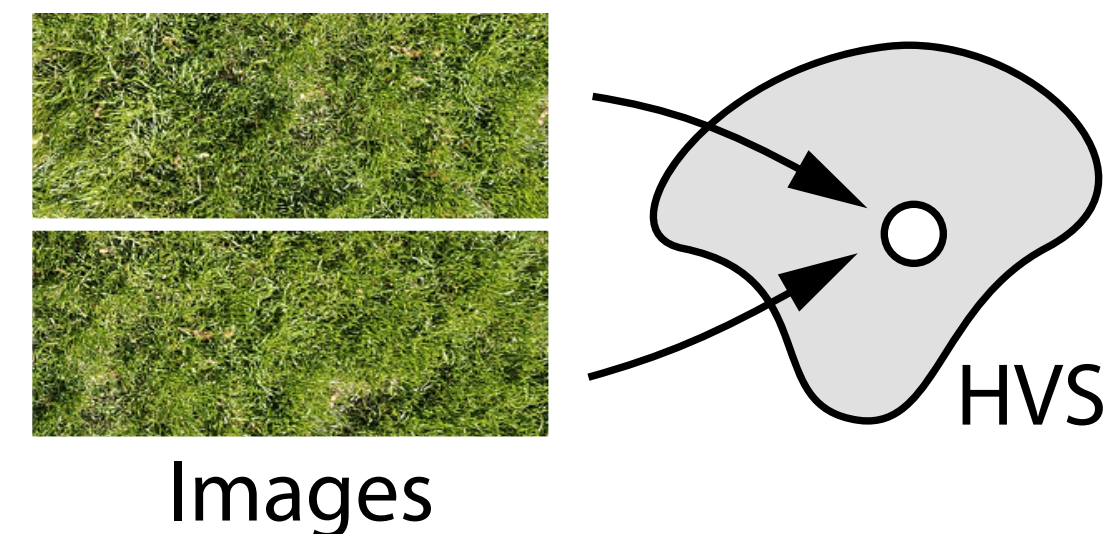
10k Accesses | 313 Citations | 17 Altmetric | [Metrics](#)

Abstract

The human capacity to recognize complex visual patterns emerges in a sequence of brain areas known as the ventral stream, beginning with primary visual cortex (V1). We developed a population model for mid-ventral processing, in which nonlinear combinations of V1 responses are averaged in receptive fields that grow with eccentricity. To test the model, we generated novel forms of visual metamers, stimuli that differ physically but look the same. We developed a behavioral protocol that uses metameric stimuli to estimate the receptive field sizes in which the model features are represented. Because receptive field sizes change along the ventral stream, our behavioral results can identify the visual area corresponding to the representation. Measurements in human observers implicate visual area V2, providing a new functional account of neurons in this area. The model also explains deficits of peripheral vision known as crowding, and provides a quantitative framework for assessing the capabilities and limitations of everyday vision.



Color metarism



Ventral metarism

Ventral Metarism



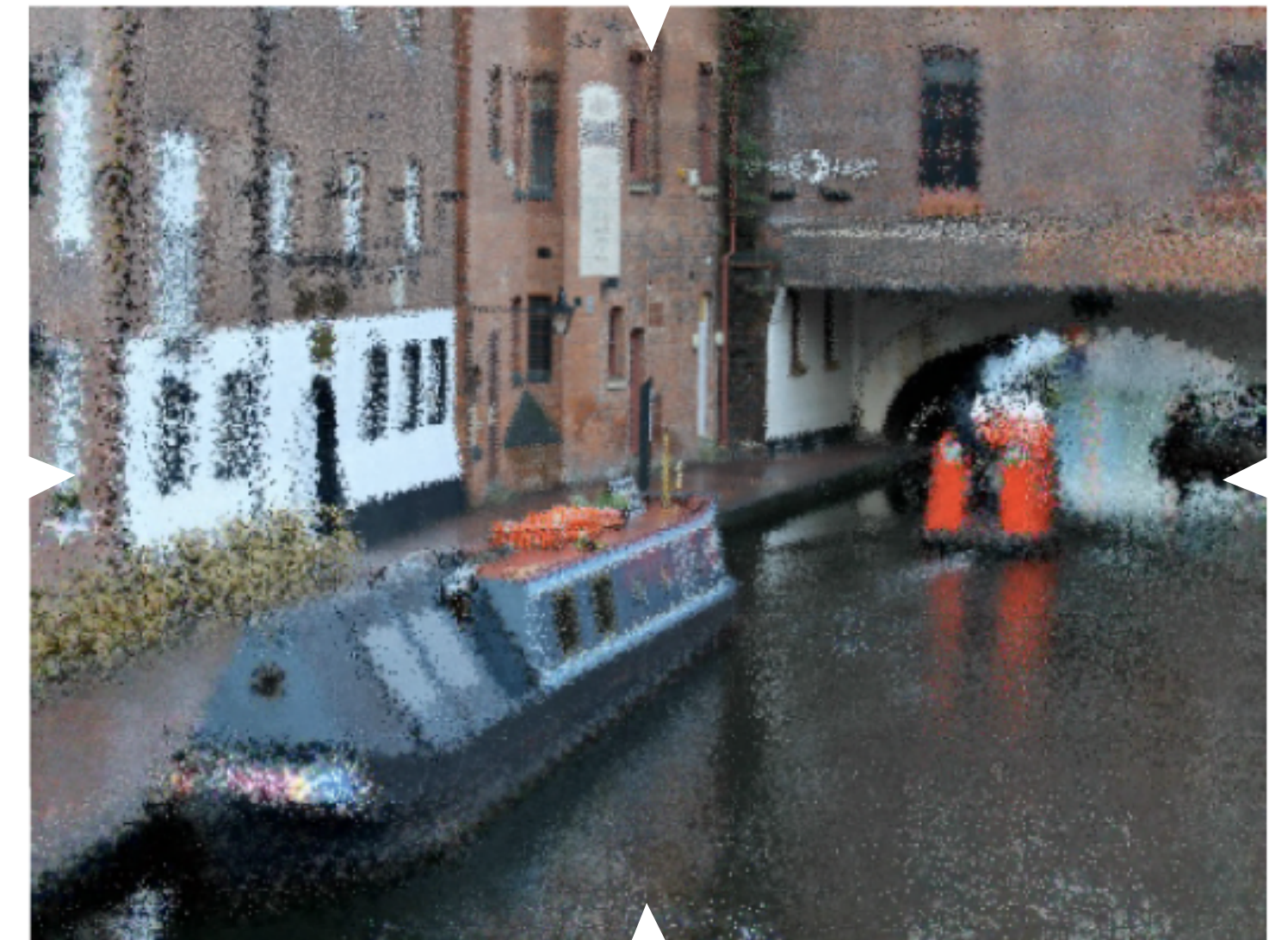
Ventral Metarism



Ground truth



Acuity-only [0.5 ms]

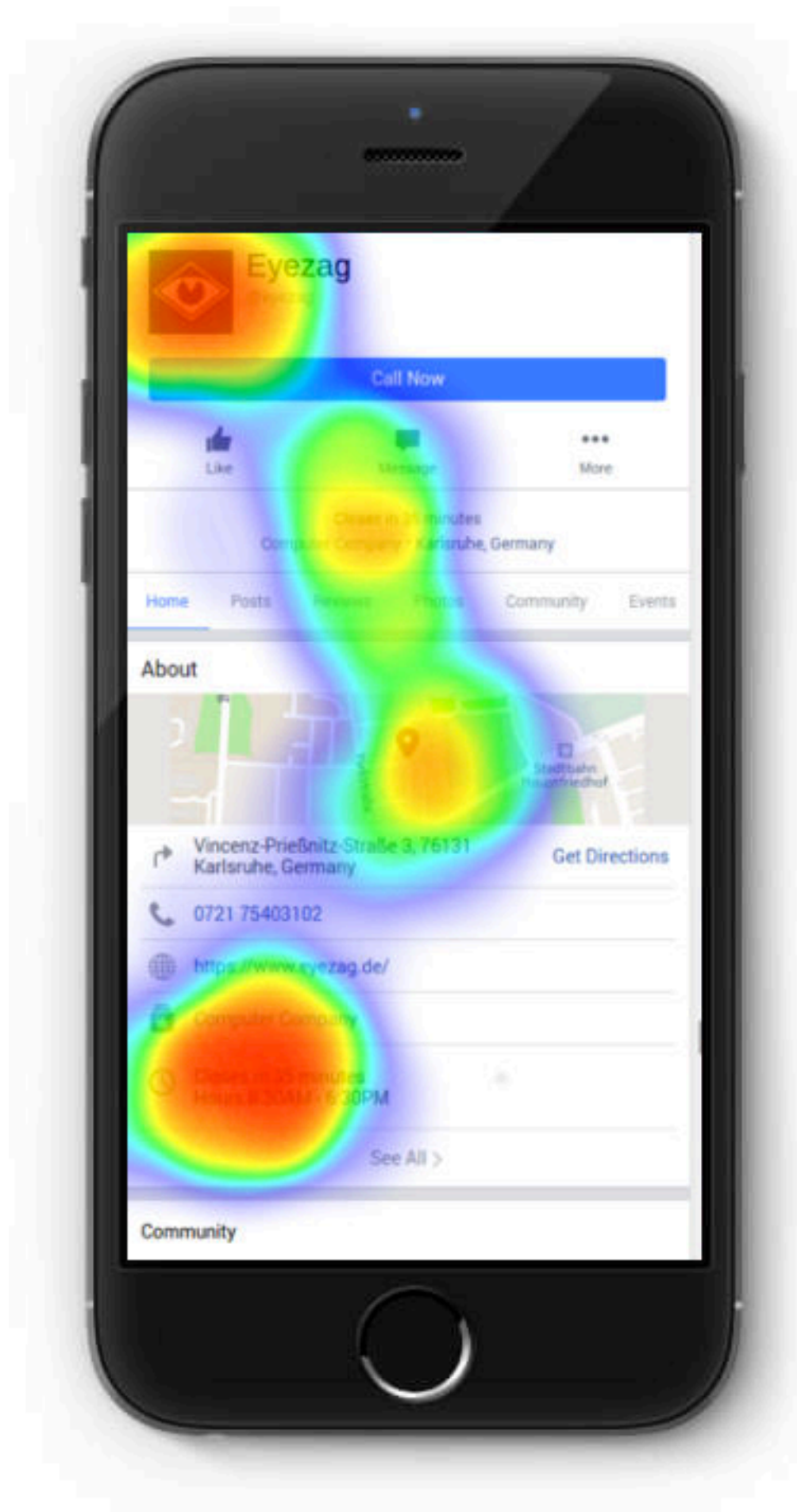
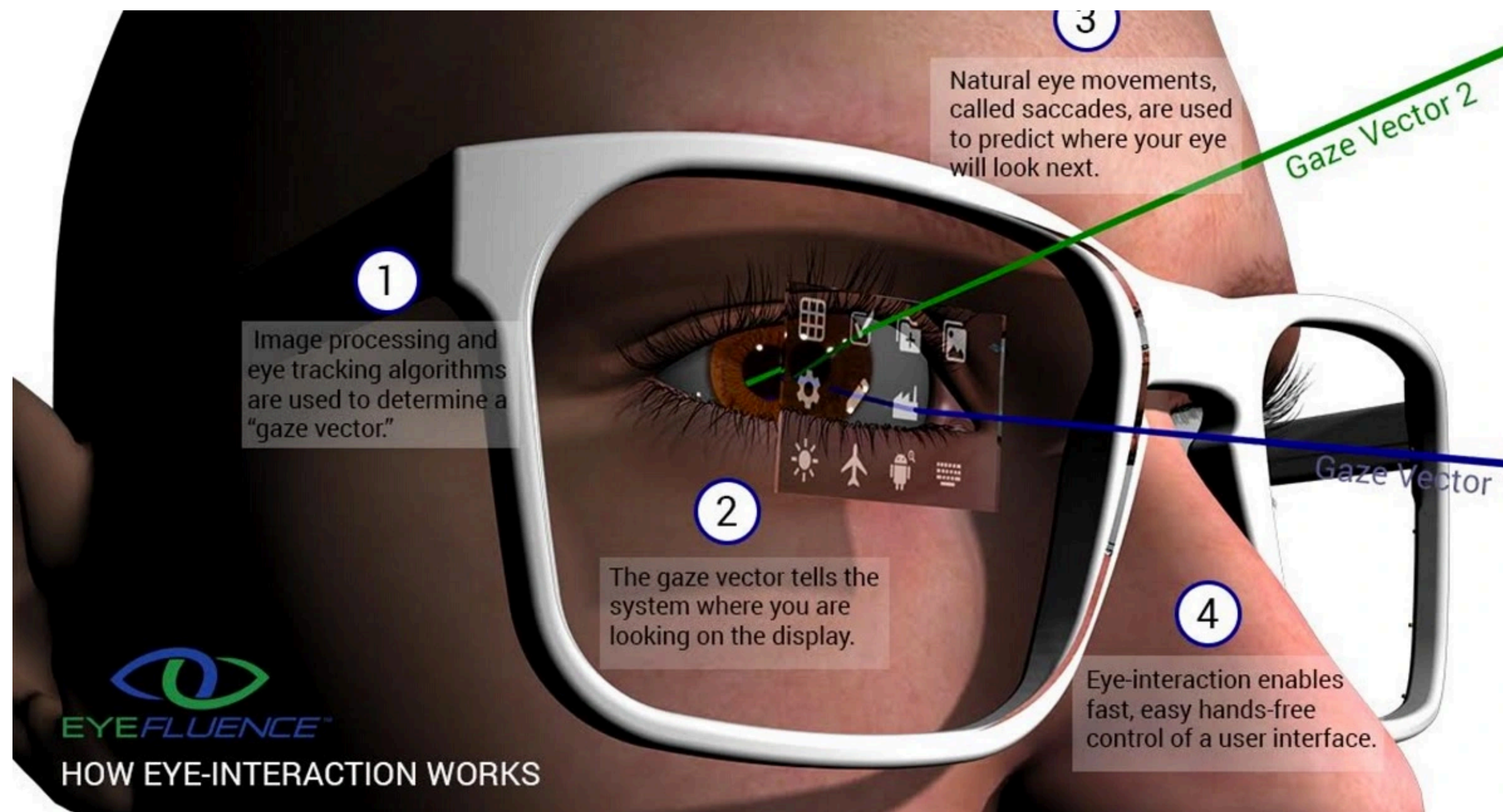


Metamer (Ours) [0.7 ms]

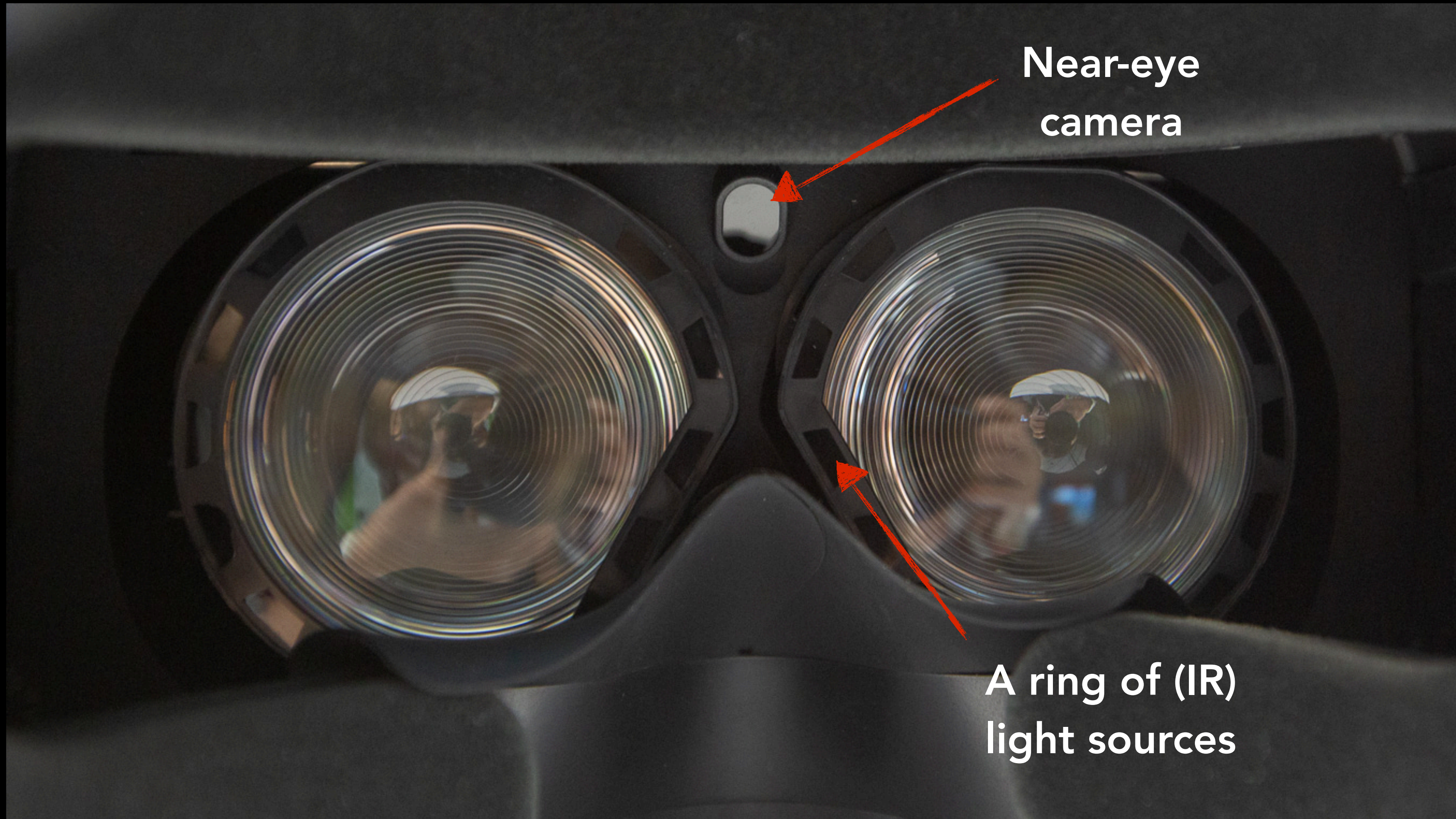
Gaze Tracking

Foveated rendering requires tracking gaze.

Gaze tracking is just in general very useful (in AR/VR)

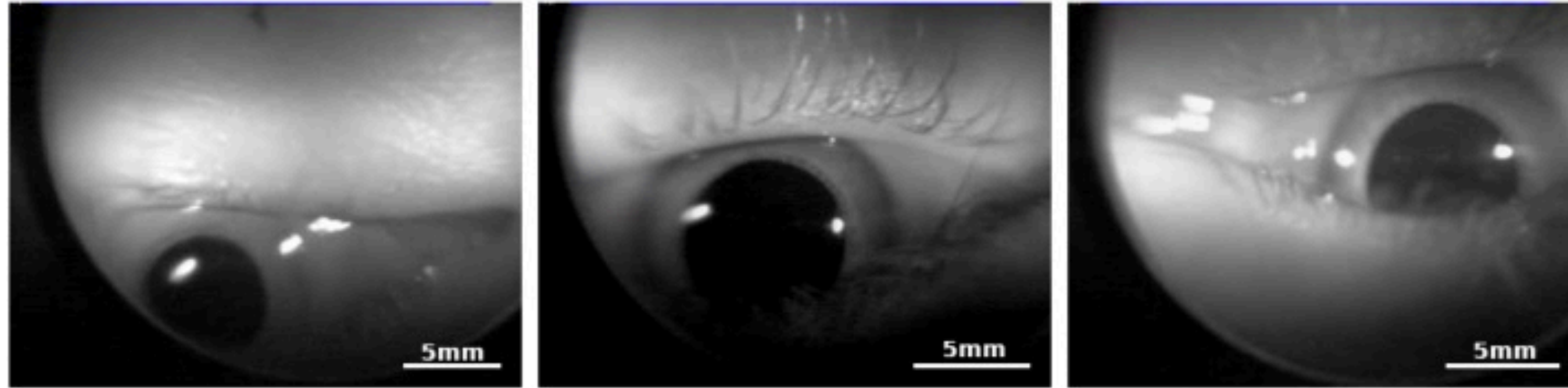


Gaze Tracking Hardware



VIVE Pro Eyes

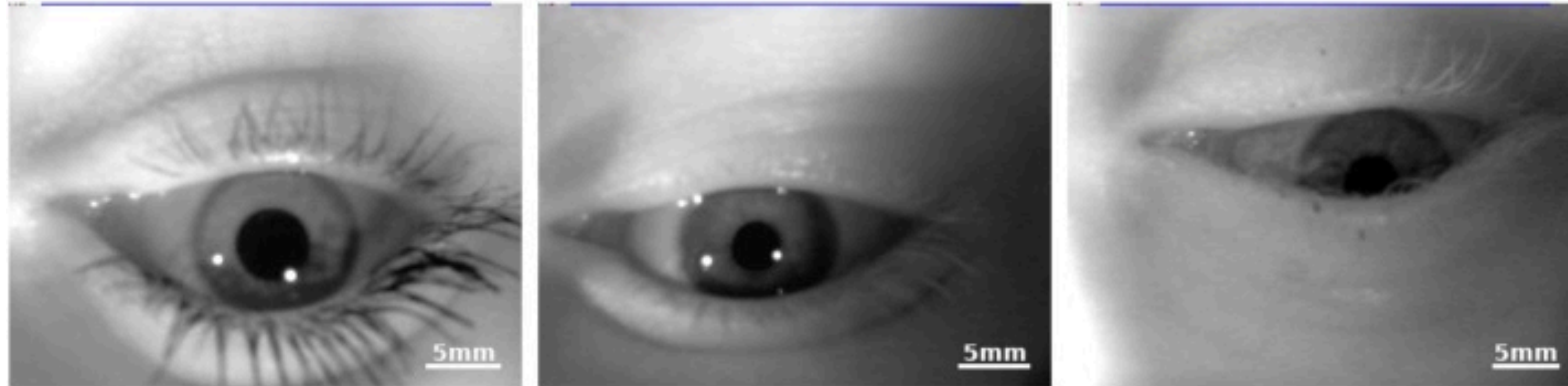
Near-Eye Images



(a)

(b)

(c)



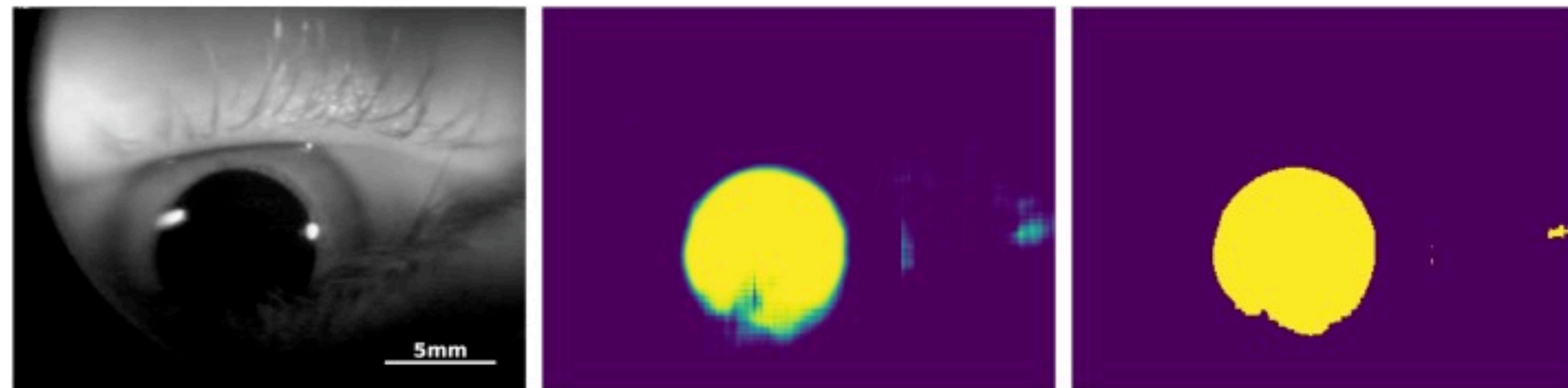
(d)

(e)

(f)

Gaze Tracking Algorithm

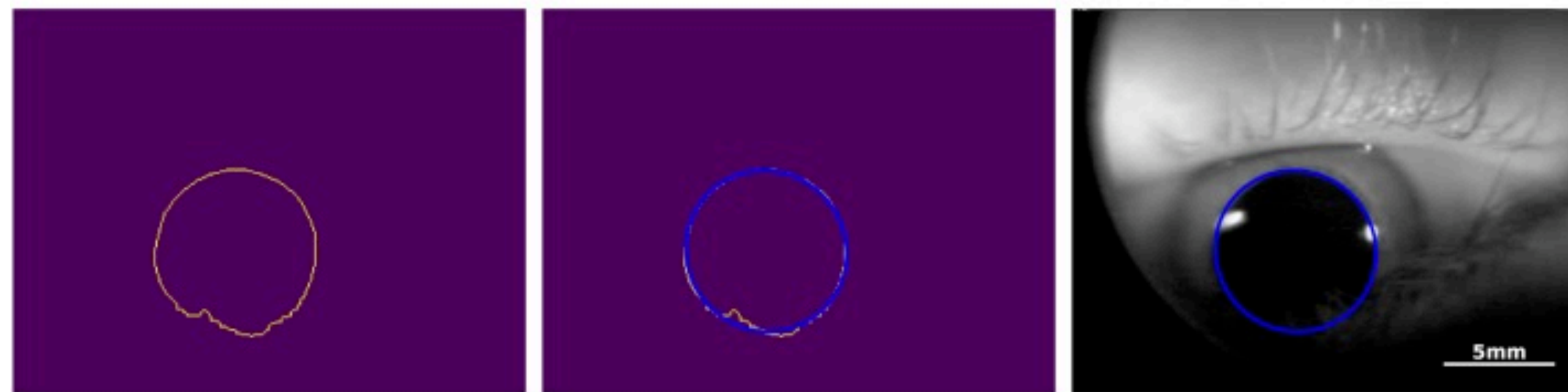
Segment the pupil from the eye



(a)

(b)

(c)

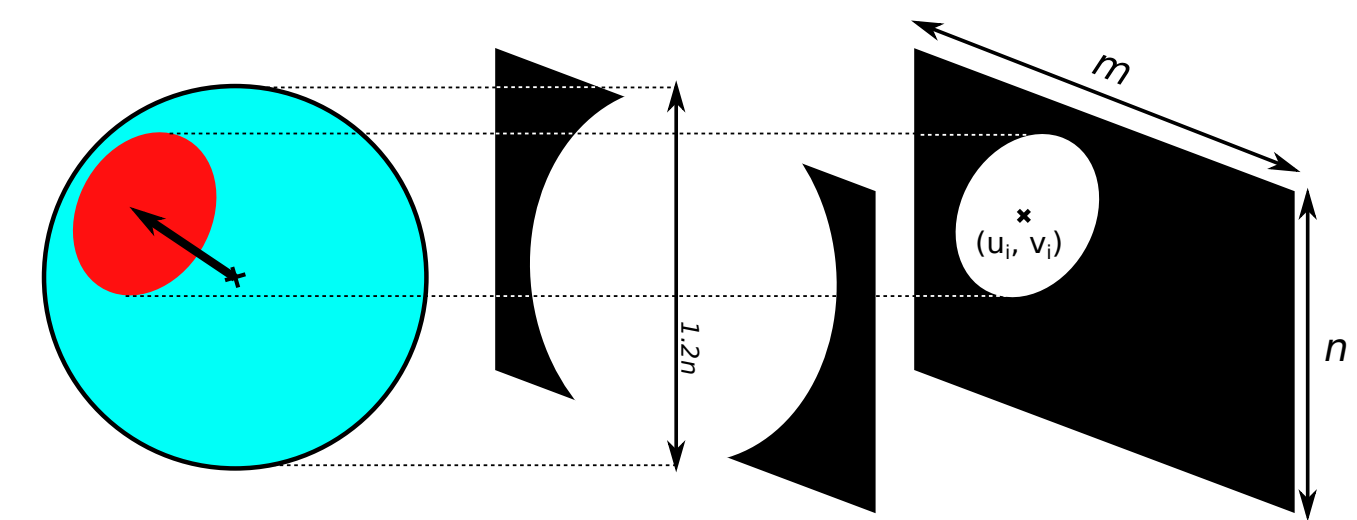


(d)

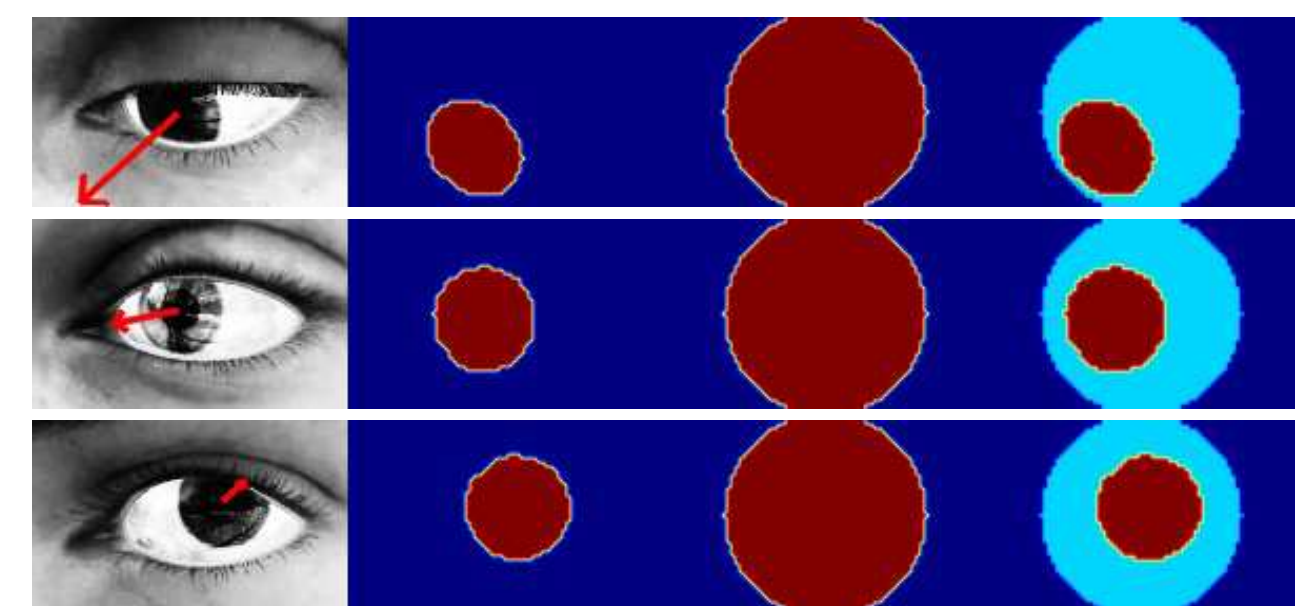
(e)

(f)

Estimating the orientation (gaze)



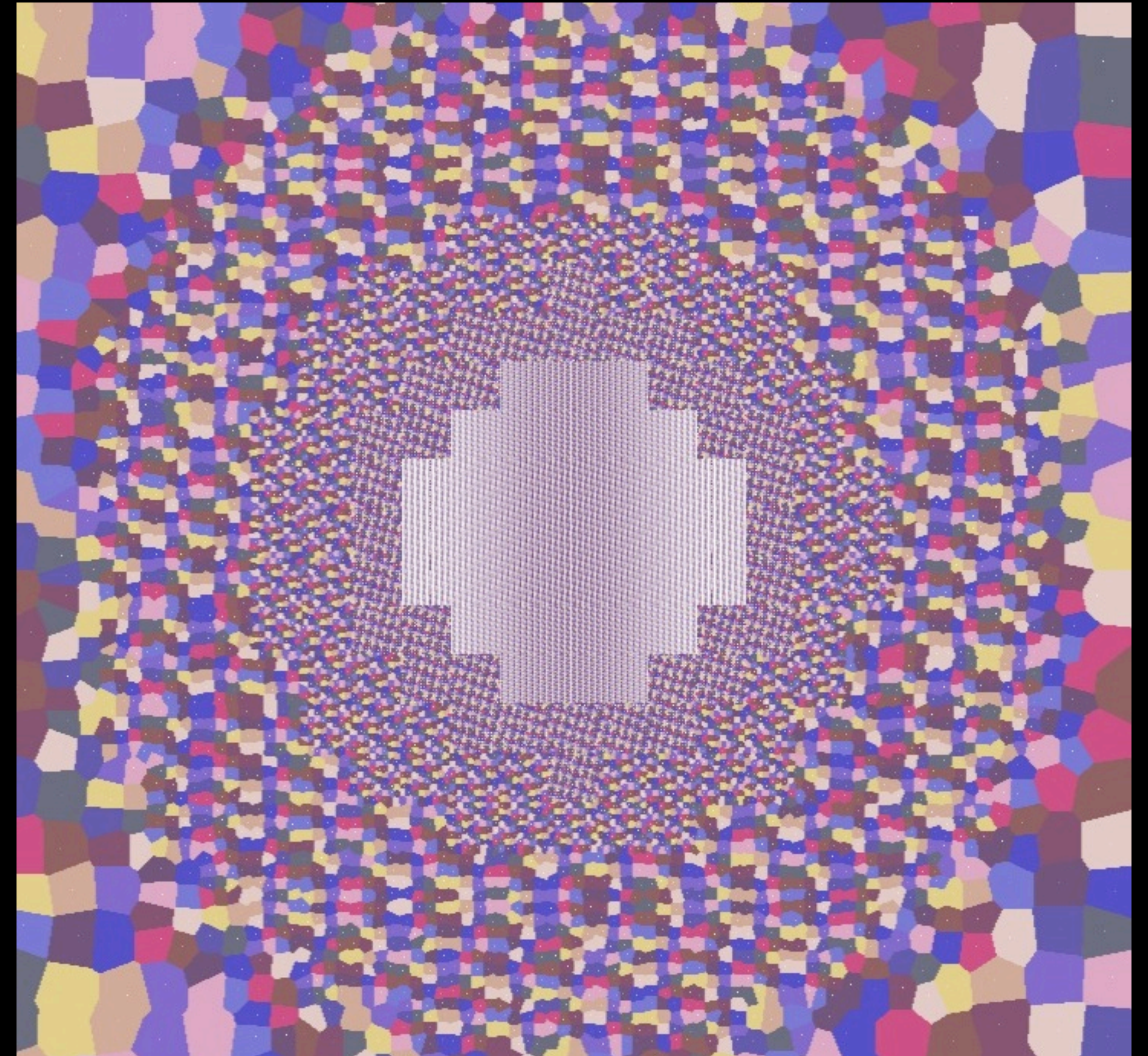
(a)



(b) Example gazemaps from UnityEyes

FOVEATED RENDERING

- ▲ We can only see clearly where we are looking at
- ▲ Shading at full rate everywhere is a waste of computation
- ▲ Steps
 - Create a density map
 - Ray trace 1 sample for each area
 - Reconstruct full resolution image



FOVEATED RENDERING

- ▲ We can only see clearly where we are looking at
- ▲ Shading at full rate everywhere is a waste of computation
- ▲ Steps
 - Create a density map
 - Ray trace 1 sample for each area
 - Reconstruct full resolution image

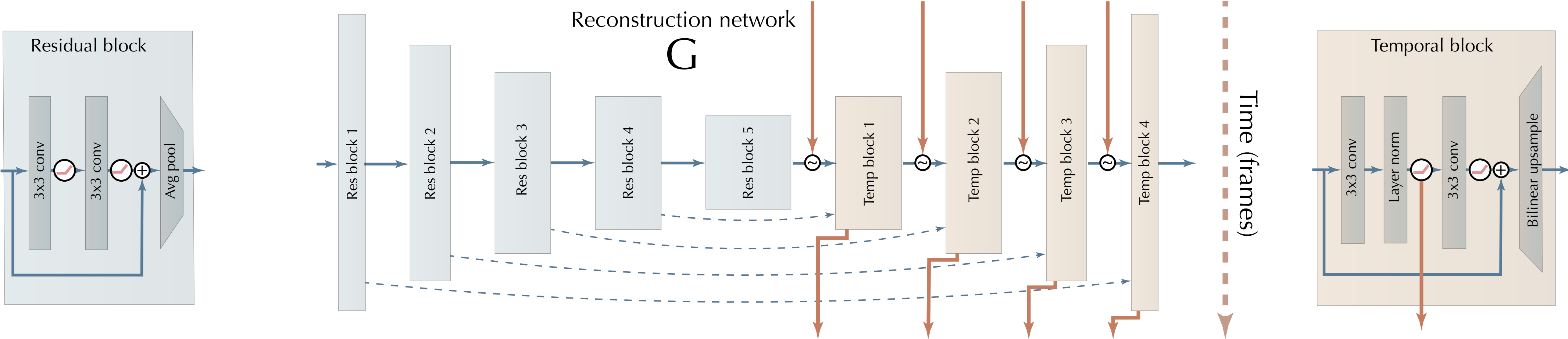


FOVEATED RENDERING

- ▲ We can only see clearly where we are looking at
- ▲ Shading at full rate everywhere is a waste of computation
- ▲ Steps
 - Create a density map
 - Ray trace 1 sample for each area
 - Reconstruct full resolution image



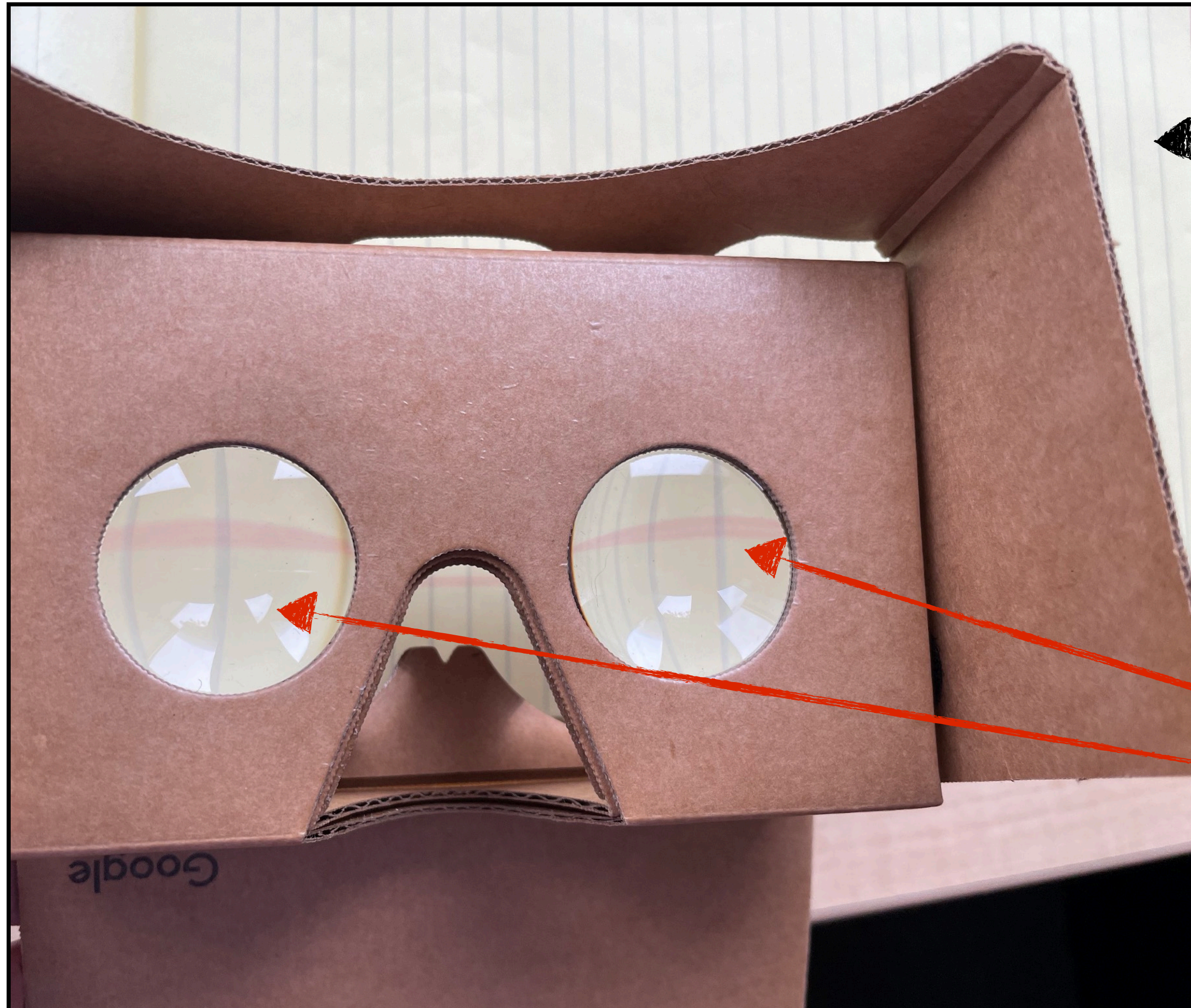
Neural Reconstruction



VR Rendering

- Tracking
- Stereo rendering (vergence-accommodation conflict)
- Foveated rendering (+eye tracking)
- **Lens distortion correction**

Lens Distortion (in VR)

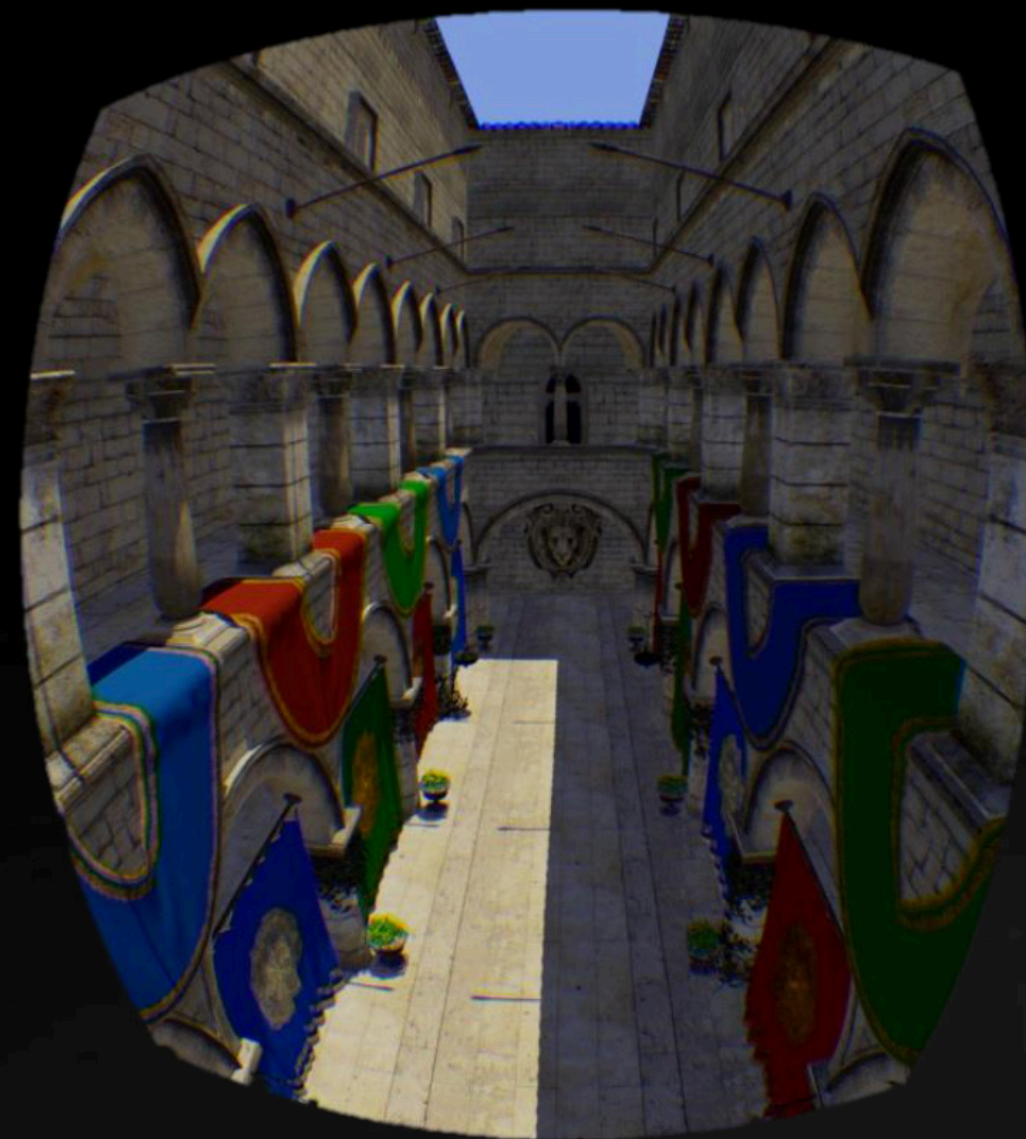


Straight lines

Distorted

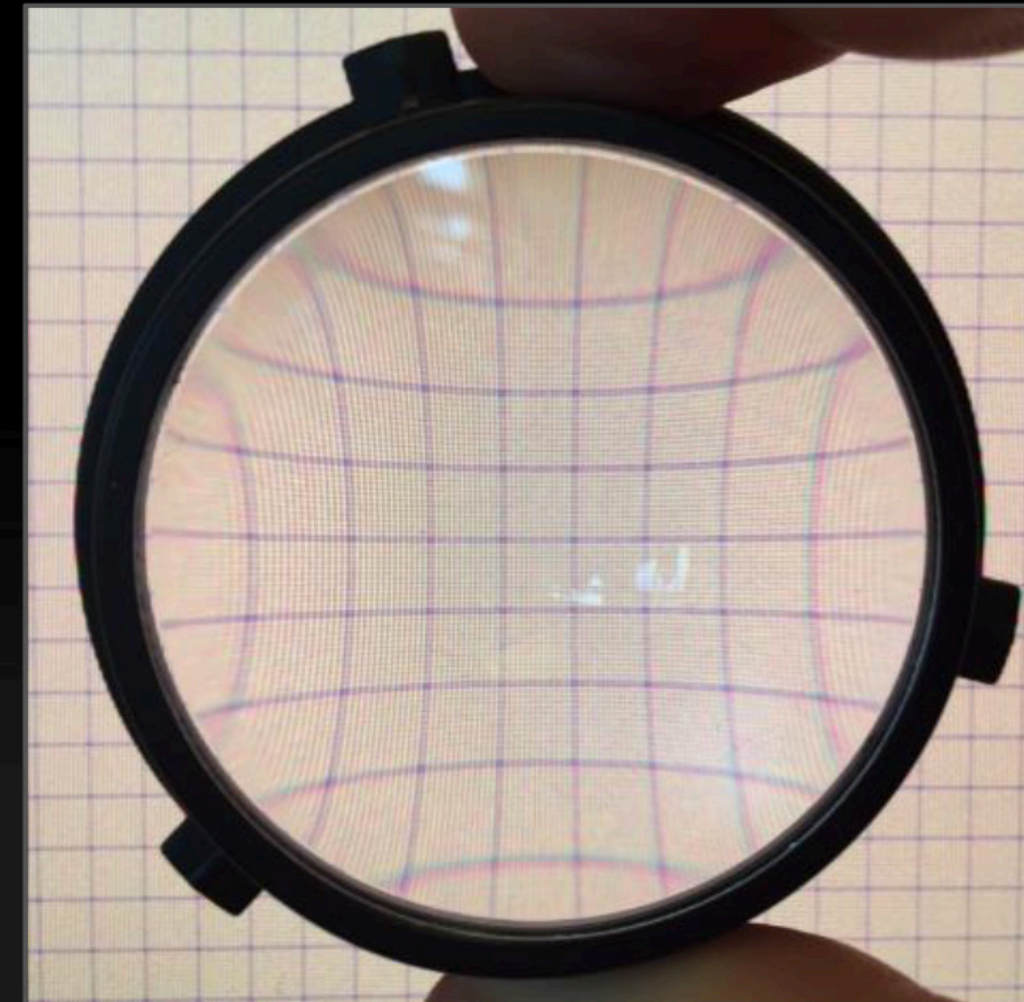
A photo of my cardboard

Lens Distortion Correction



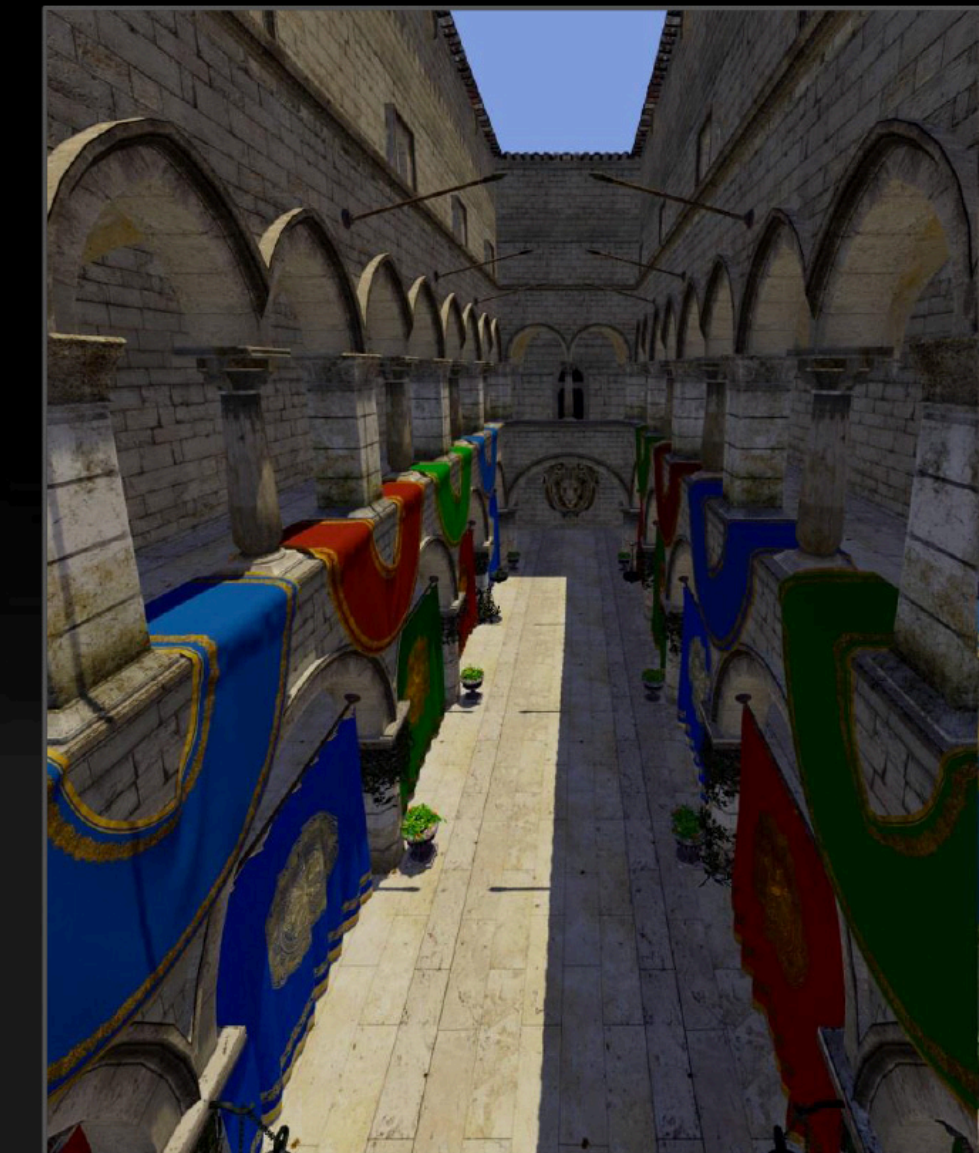
LCD Display

Warped “fisheye”-like image required to match optics - enlarged in the center and compressed in the periphery



Optics

Transforms light from display to a wide field of view focused on the eye

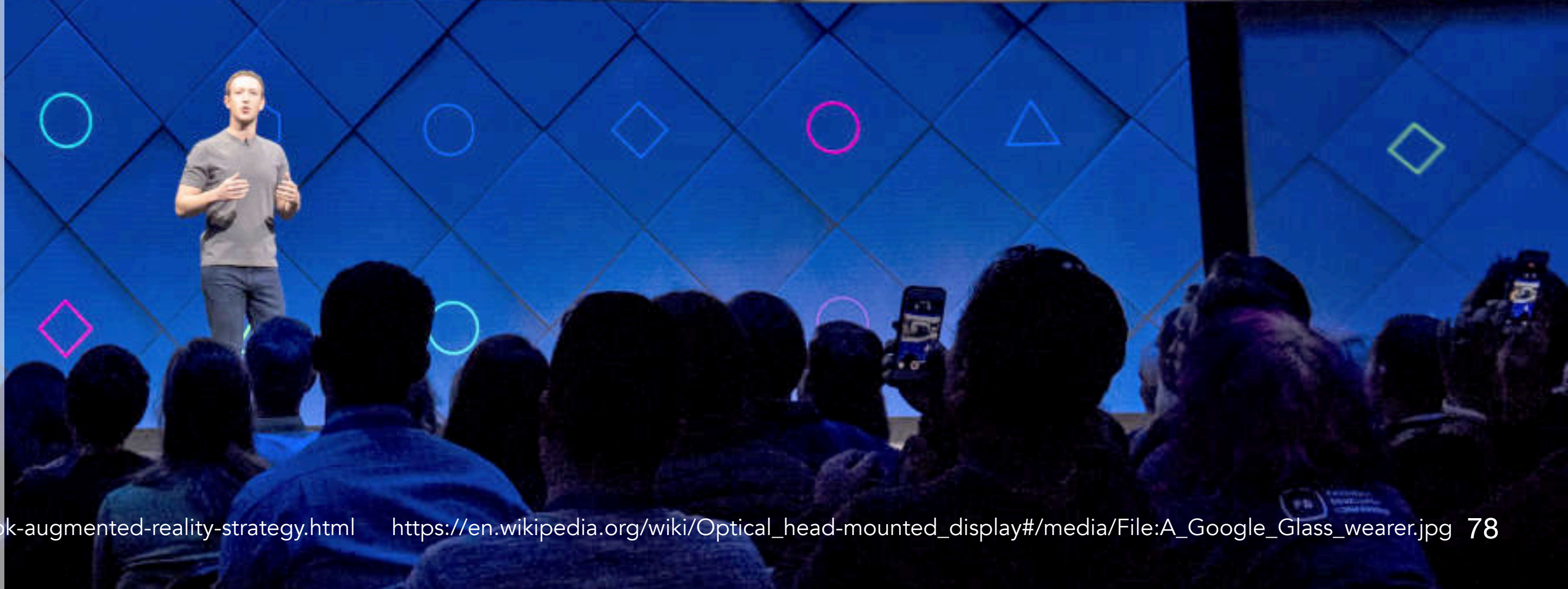
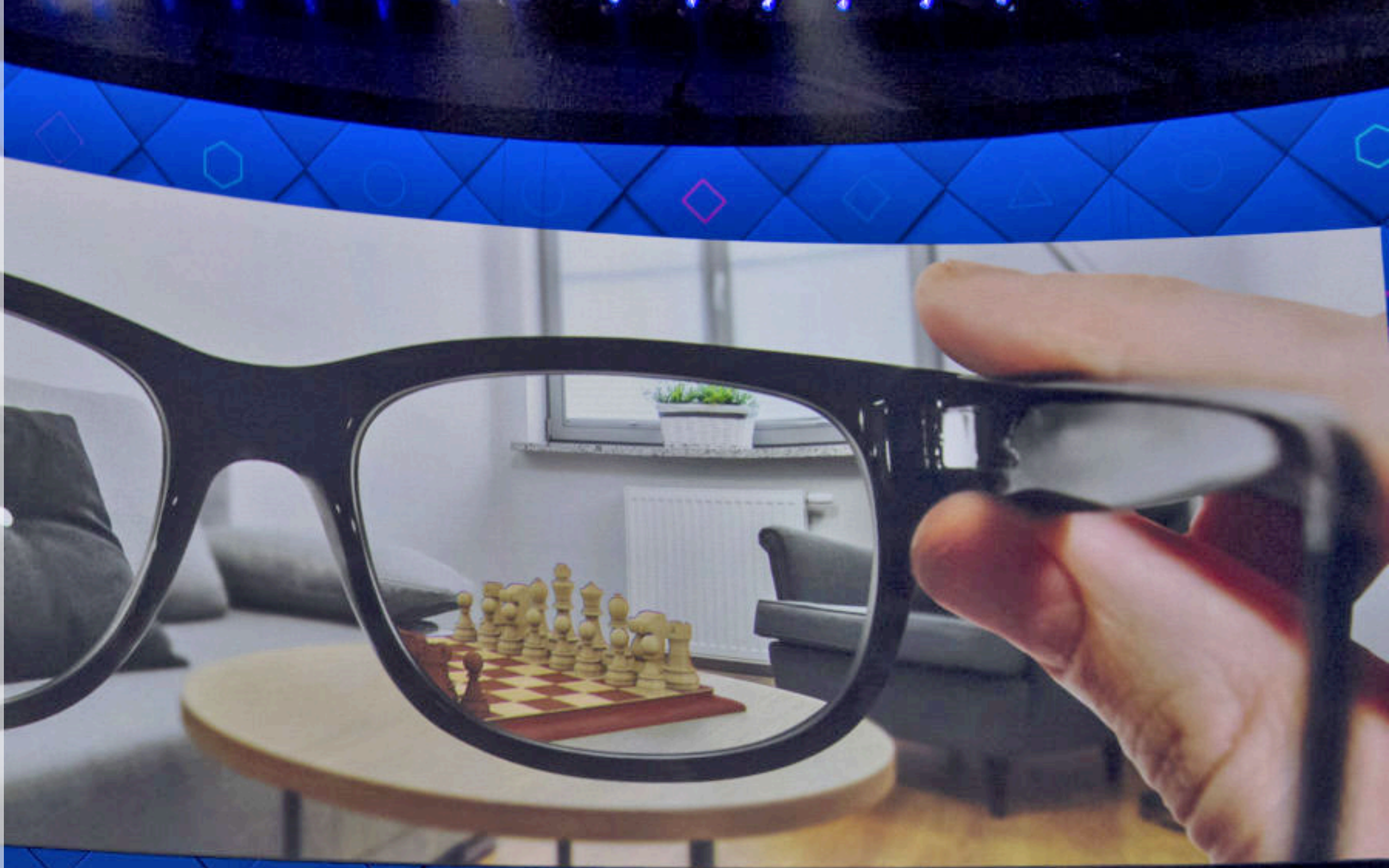


User's view


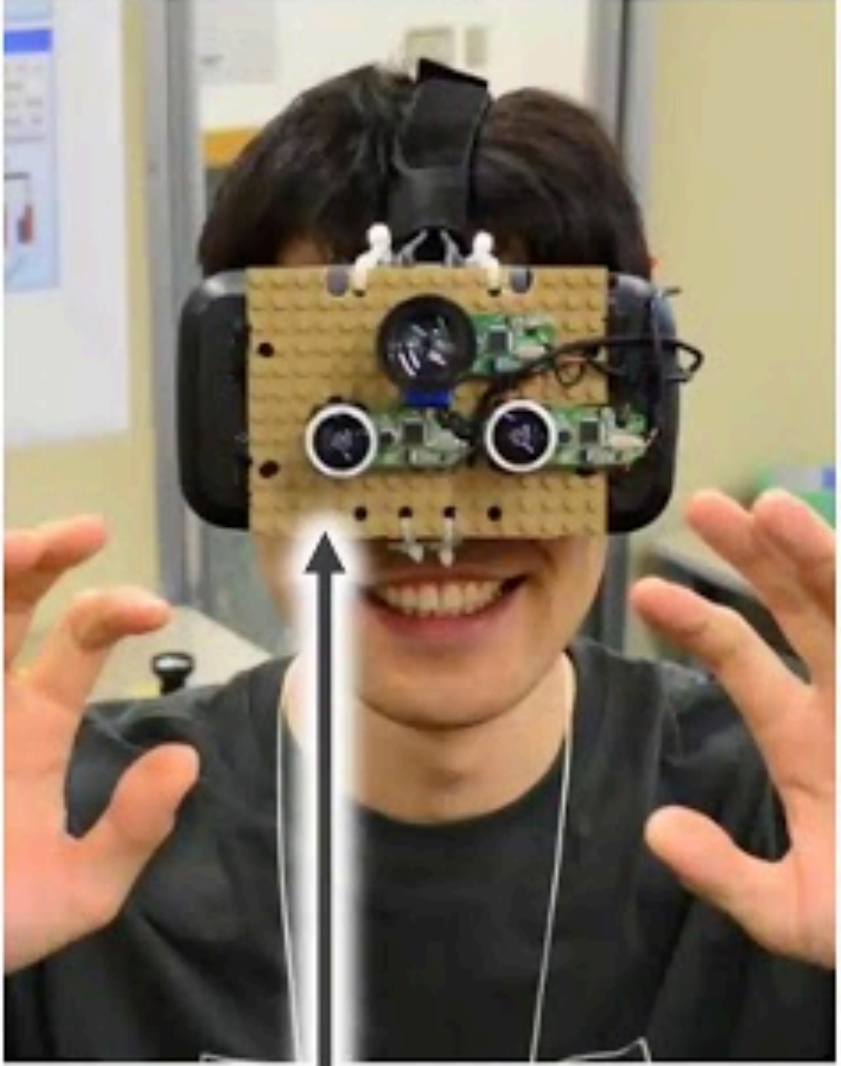
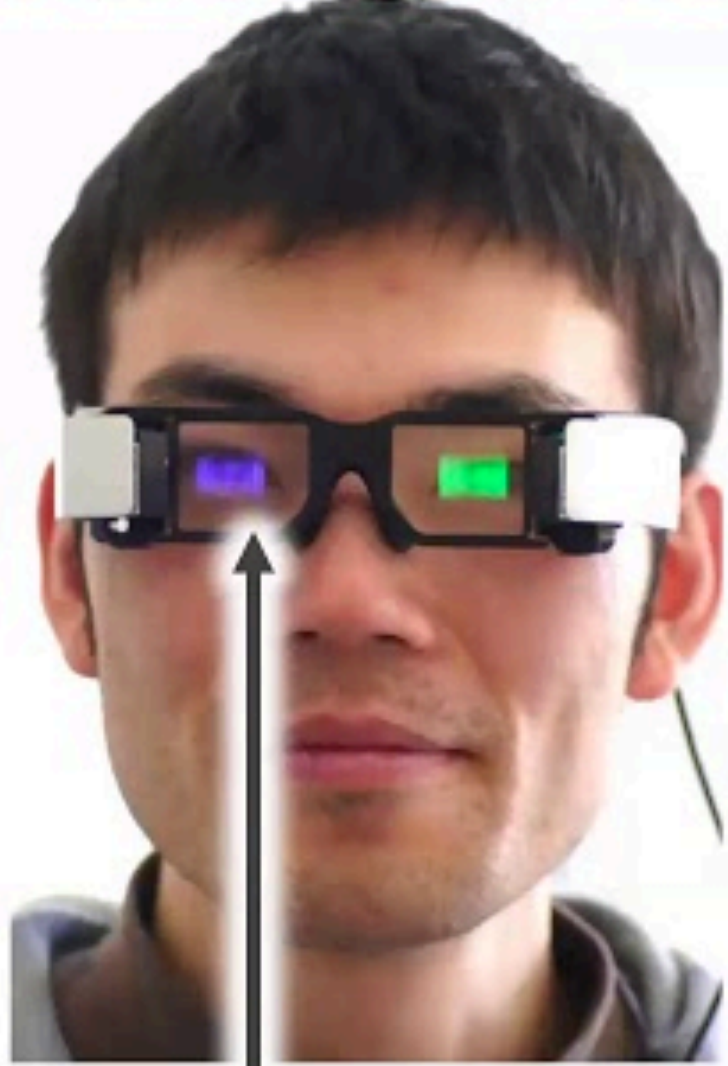
User sees correctly proportioned (not fisheye) scene with wide field of view

Augmented Reality

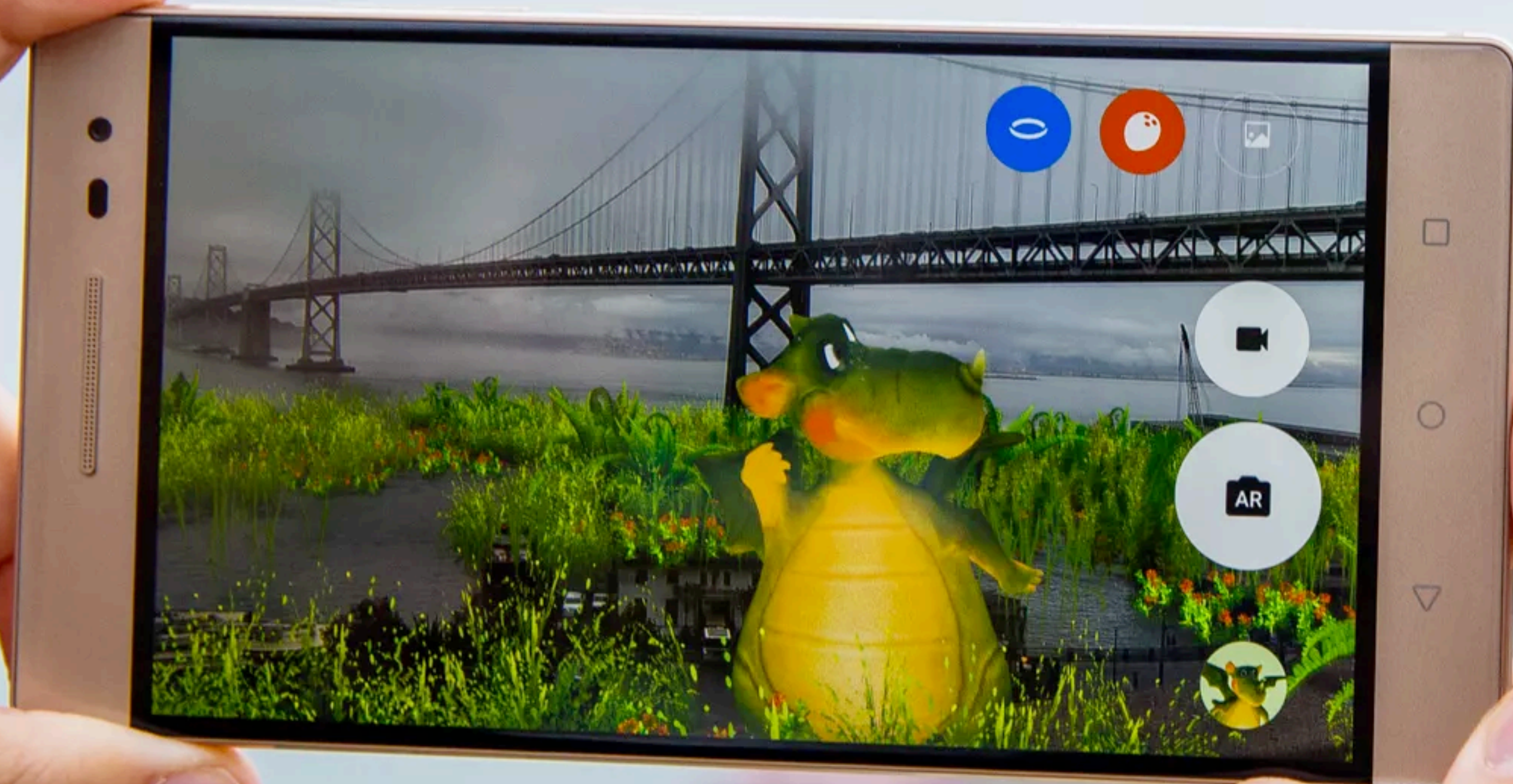
what you see = real world light +
light projected from display.



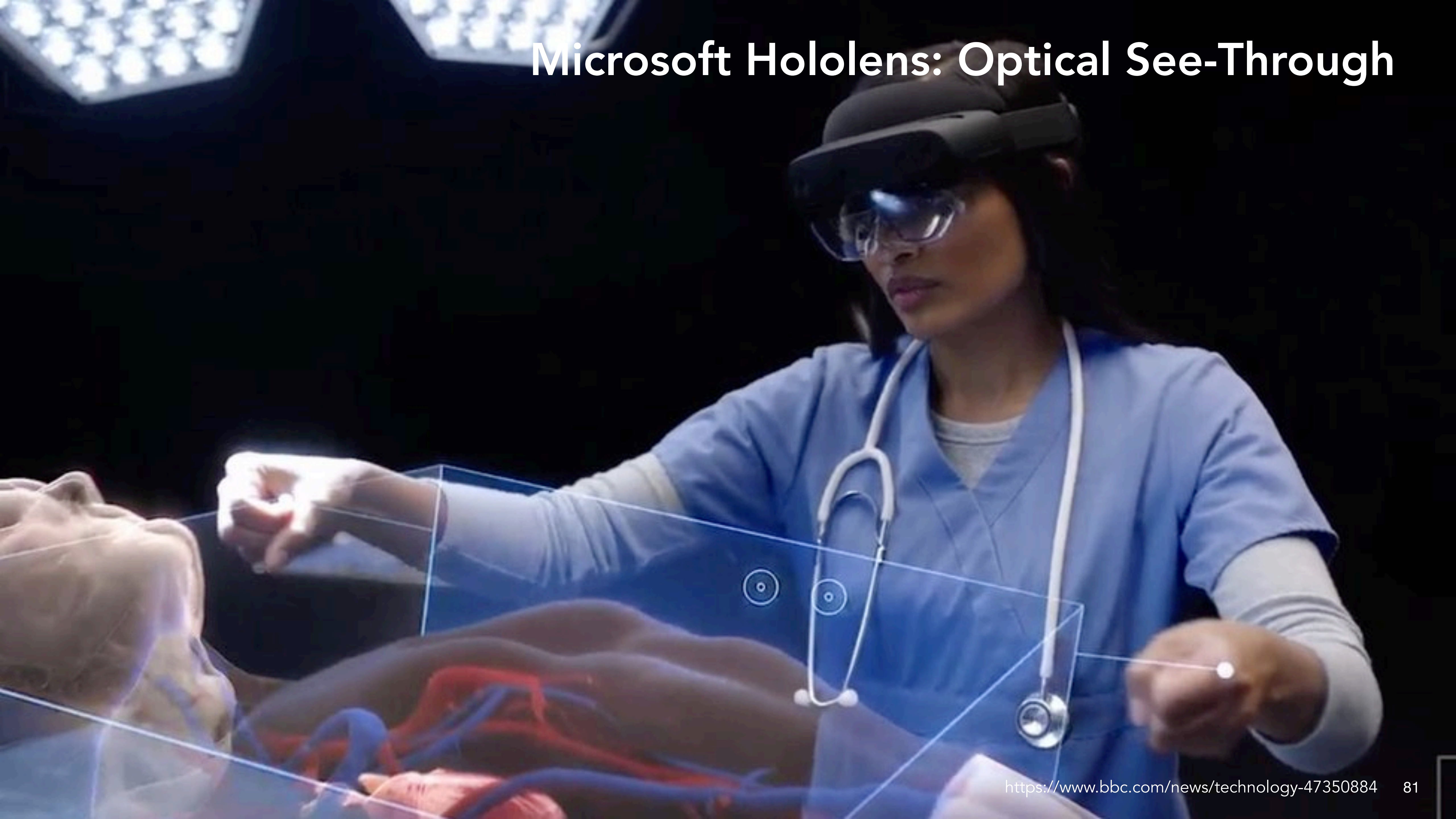
Optical See-Through vs. Video See-Through

<p>VR HMDs</p> 	<p>Video See-Through</p> 	<p>Optical See-Through</p> 
<p>Opaque display</p>	<p>Scene camera</p>	<p>See- through</p>

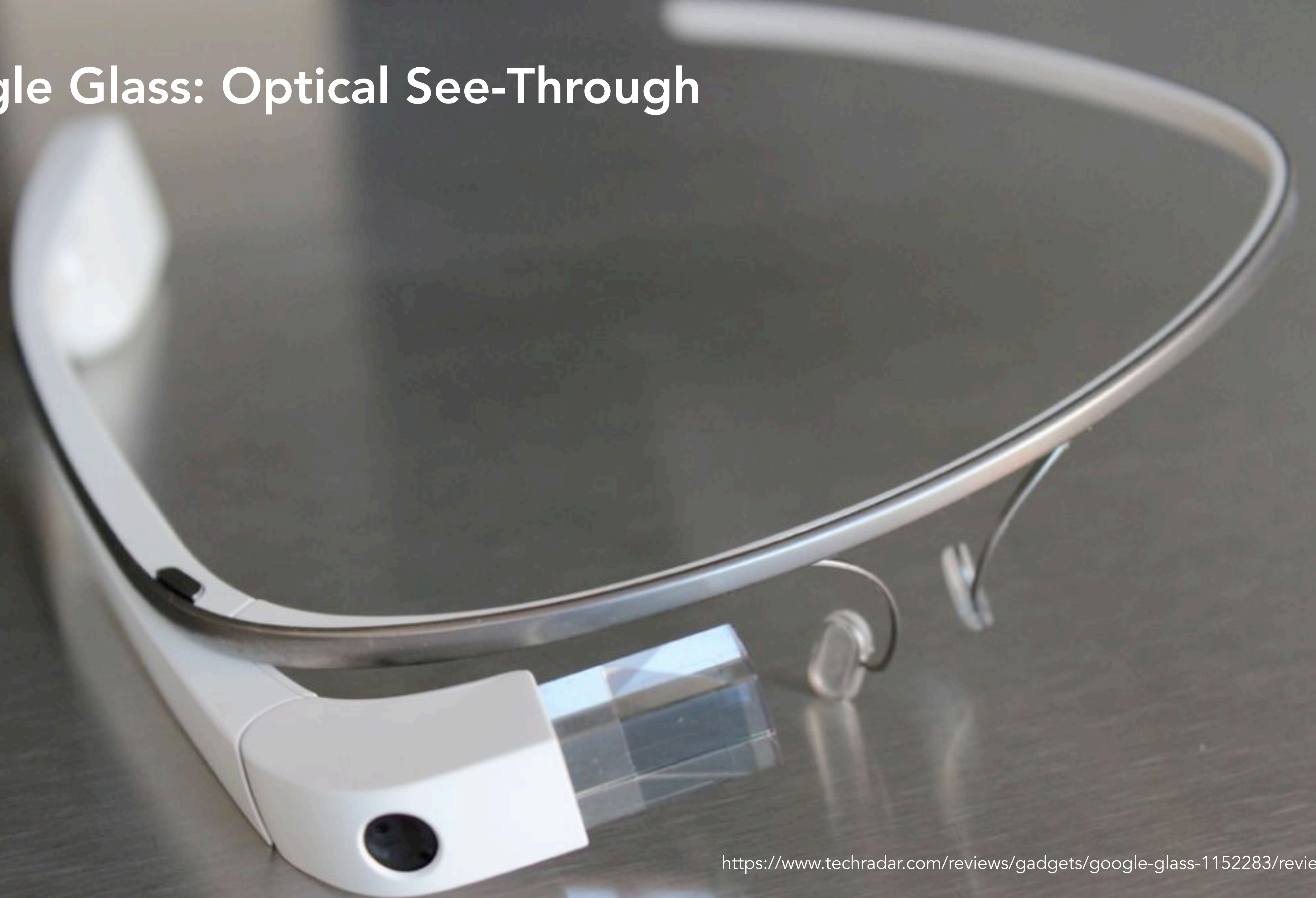
AR on Smartphone: Video See-Through



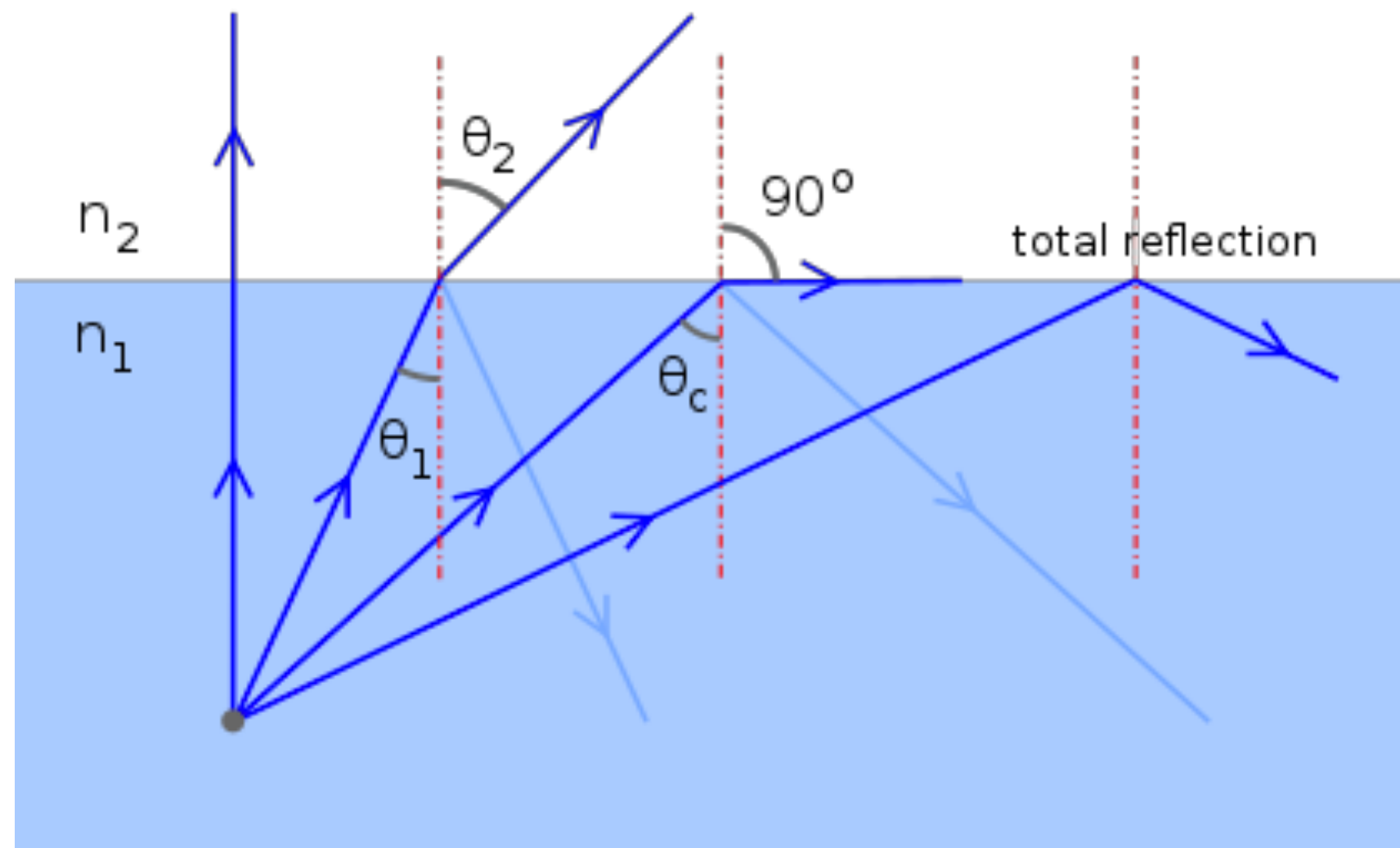
Microsoft HoloLens: Optical See-Through



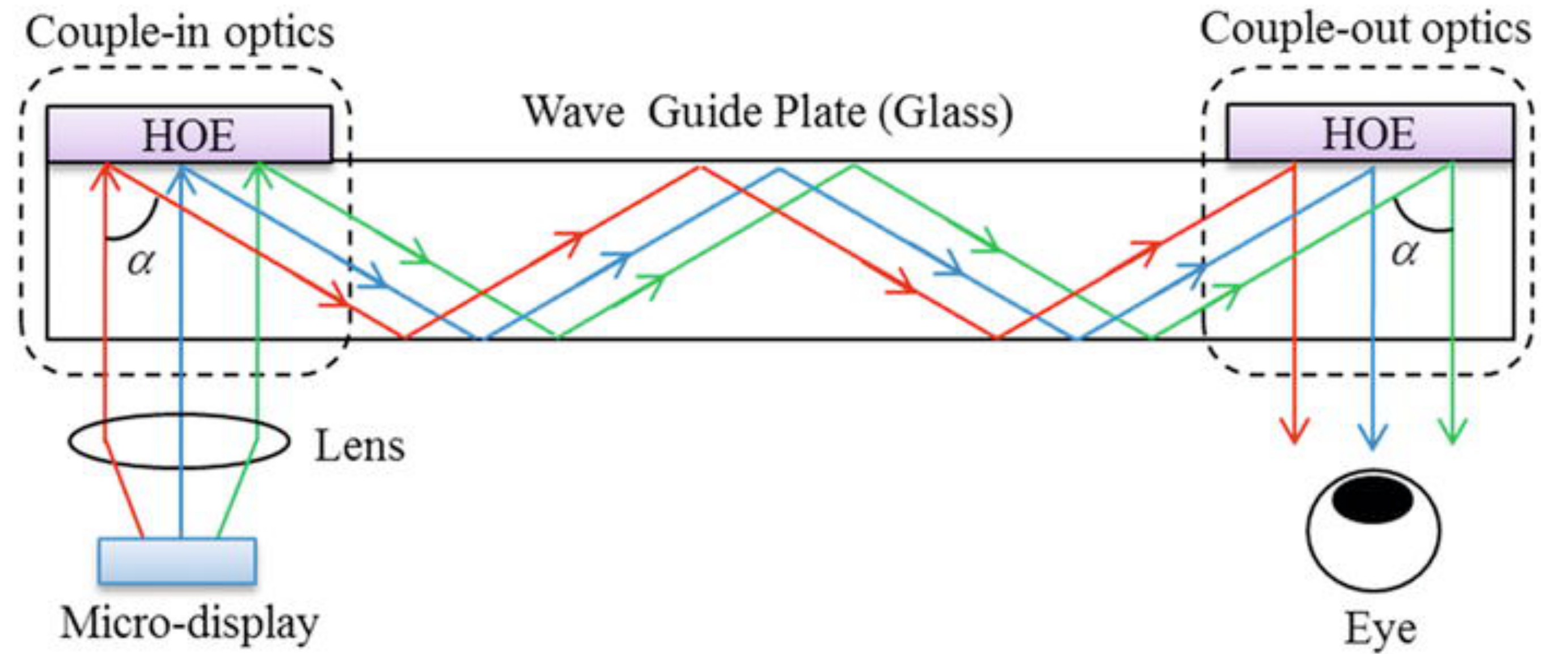
Google Glass: Optical See-Through



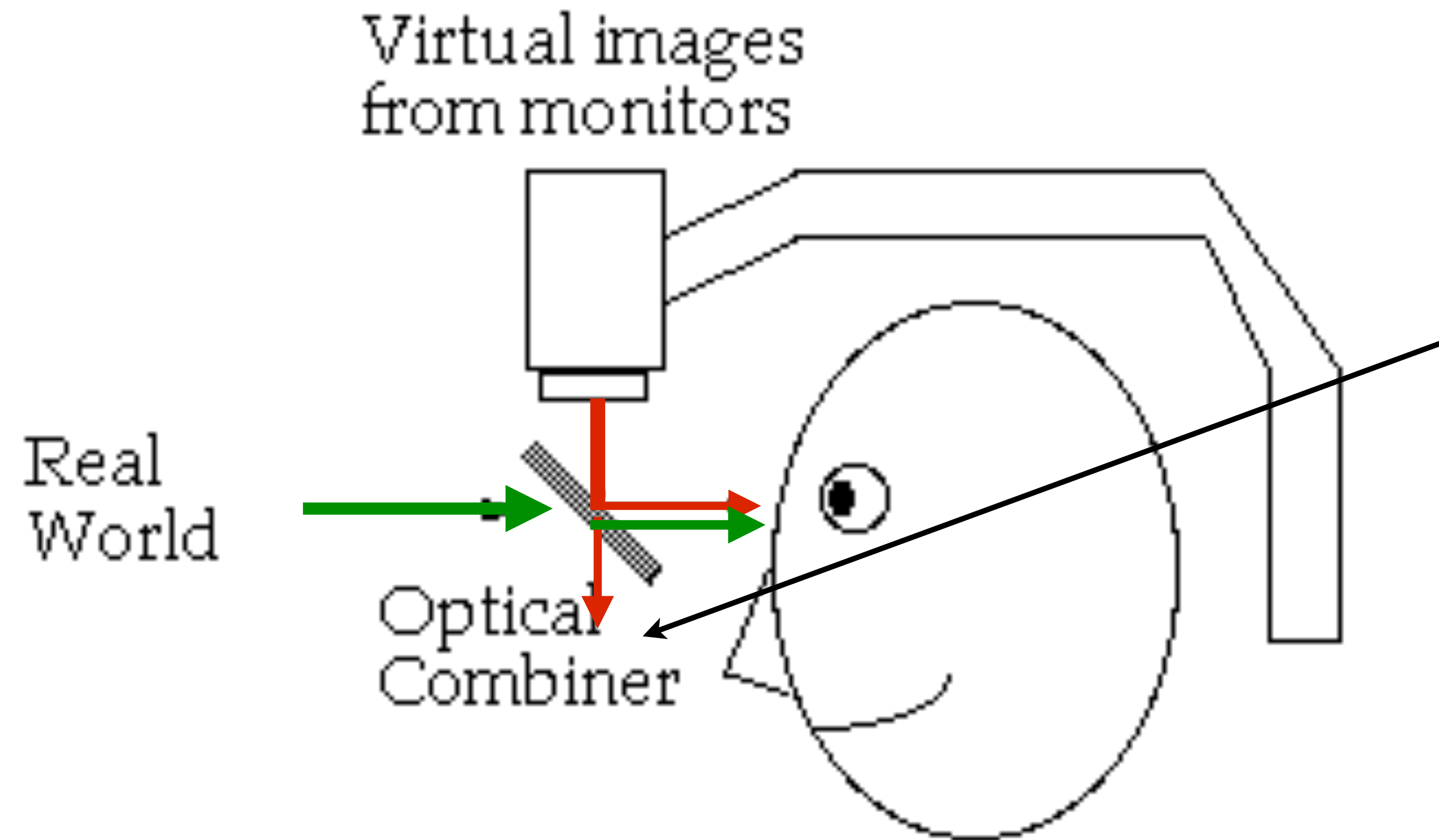
Waveguide



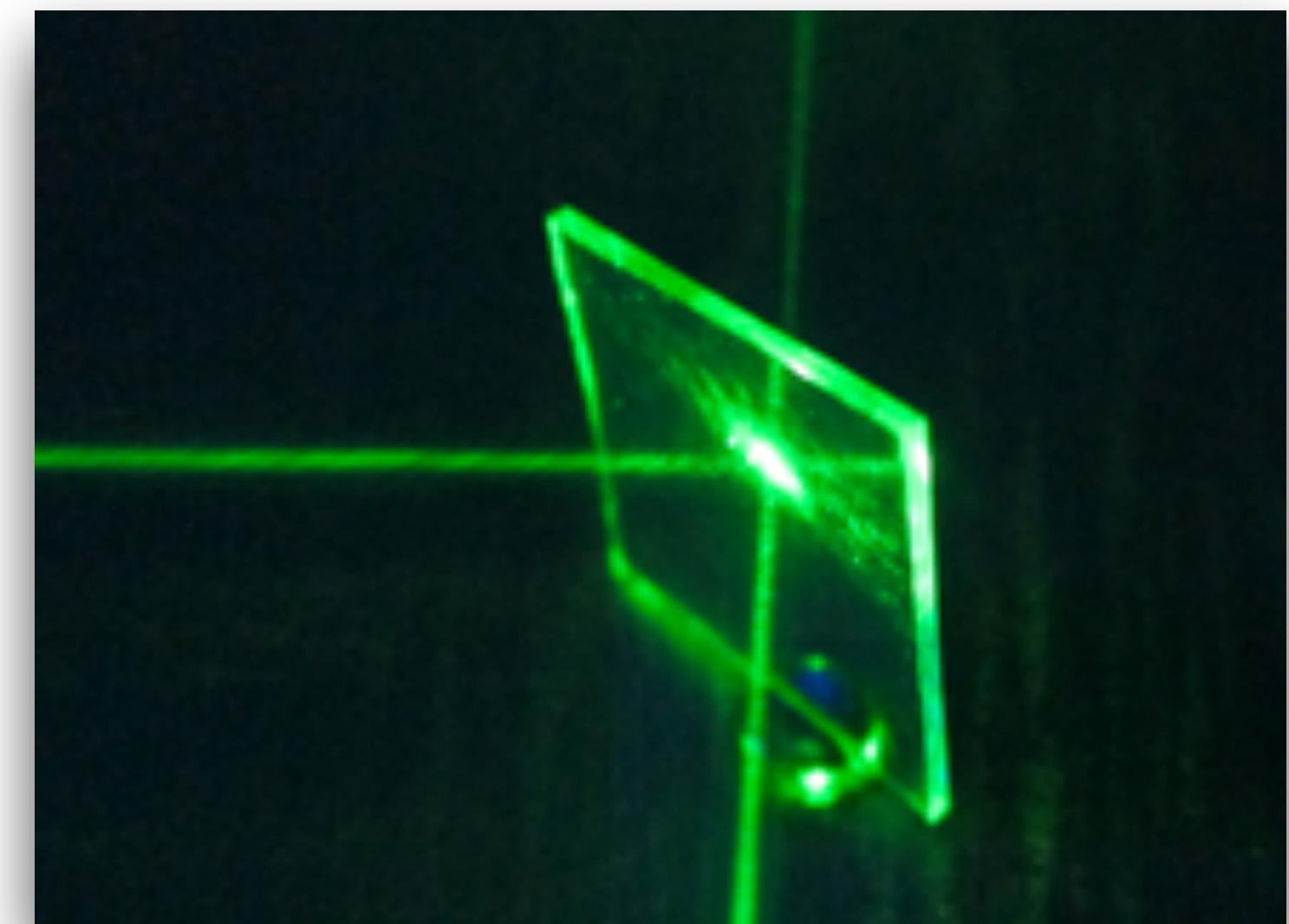
Total Internal Reflection



Optical See-Through Schematic



Beam splitter is often used as a combiner

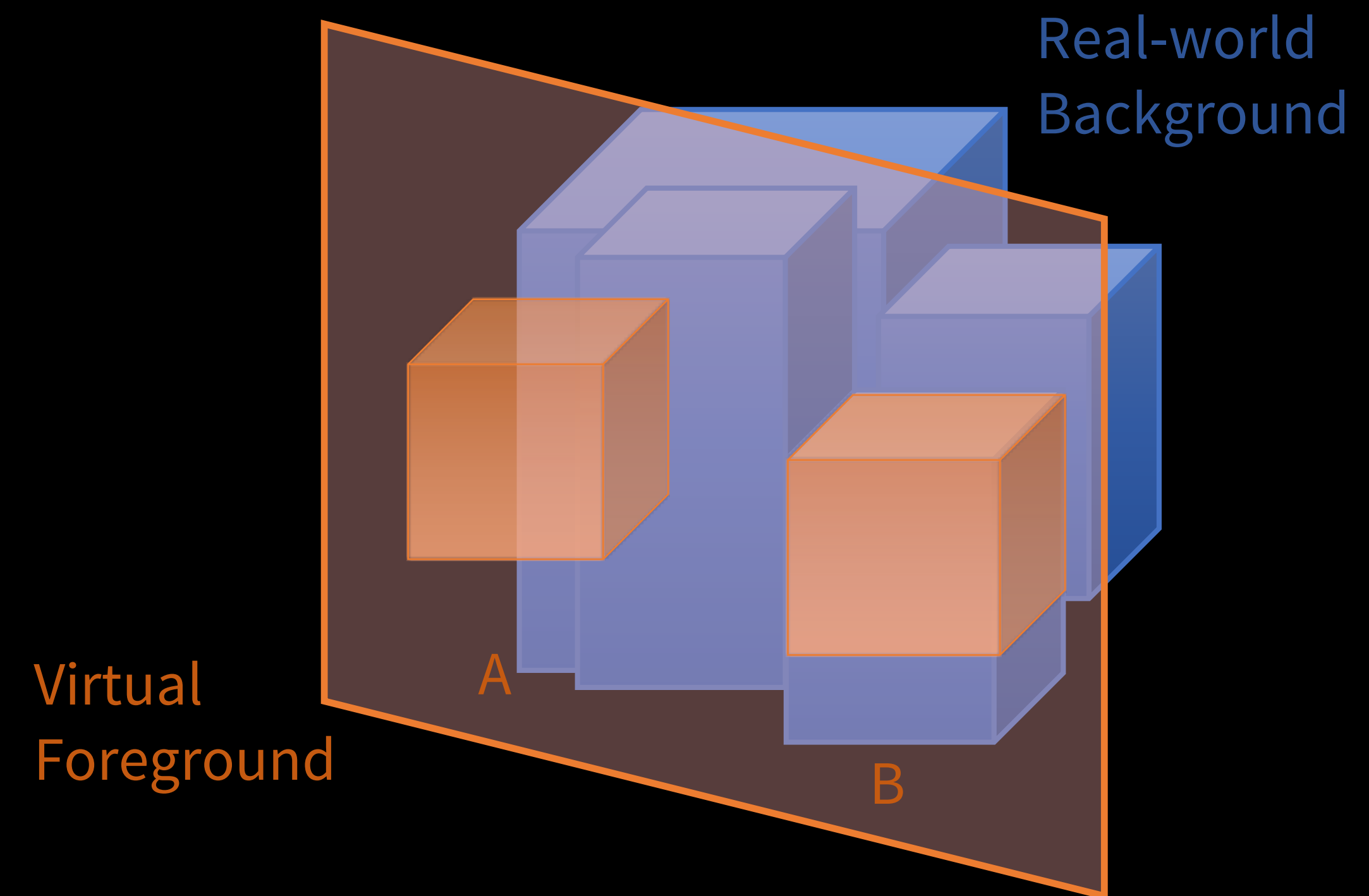




FG/BG Discounting Model

- Physics: sum of FG & BG:

$$XYZ = XYZ_{FG} + XYZ_{BG}$$



FG/BG Discounting Model

- Perceptual: weighted sum of FG & BG: (Physical $\alpha = \beta = 1$)

$$XYZ_{effective} = \alpha XYZ_{FG} + \beta XYZ_{BG}$$

- α & β depend on task, complexity, and luminance
- $\alpha > \beta$ for FG color matching
- $\alpha < \beta$ for BG brightness matching