HMD Display Optics I

Gordon Wetzstein
Stanford University

EE 267 Virtual Reality

Lecture 7

stanford.edu/class/ee267/

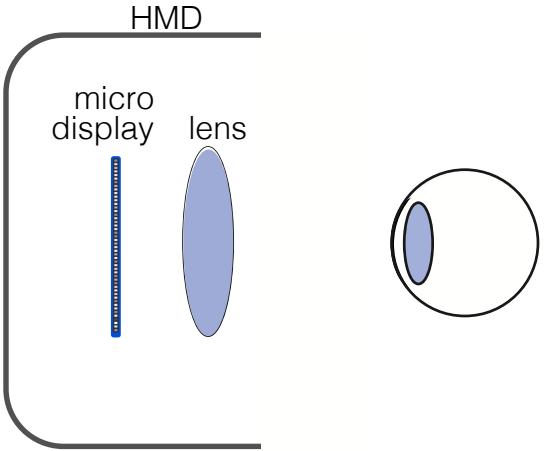
Lecture Overview

1. stereo rendering for HMDs
2. lens distortion correction using GLSL

1. Stereo Rendering for HMDs

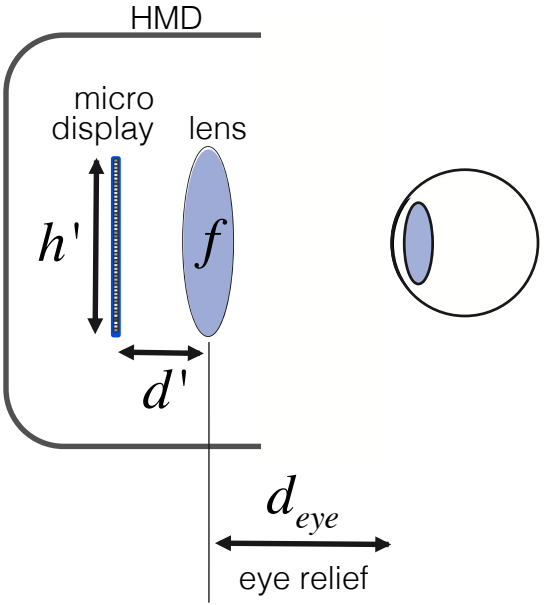
All Current-generation VR HMDs are "Simple Magnifiers"

Image Formation



Side View

Image Formation



Side View

Image Formation

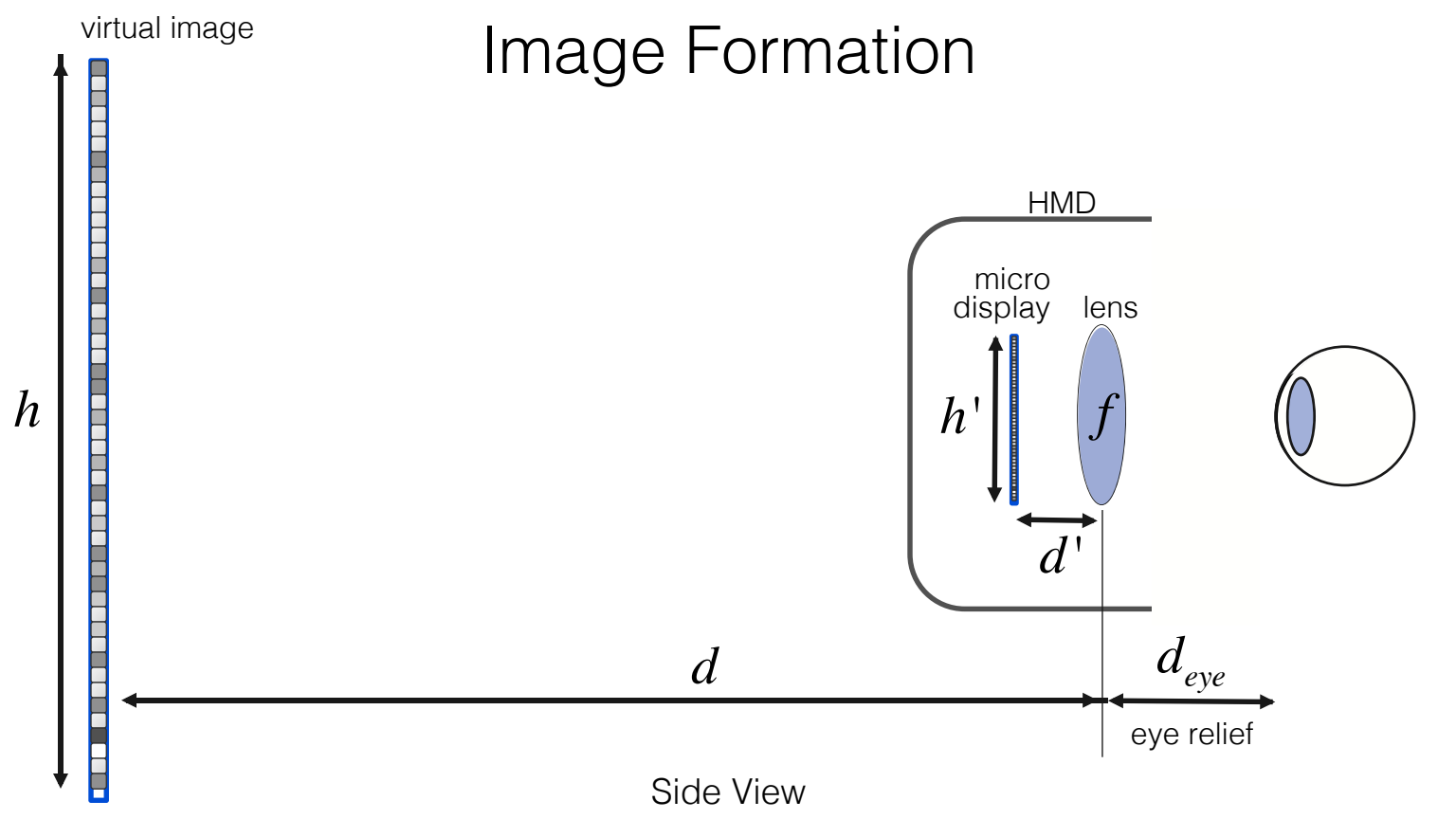
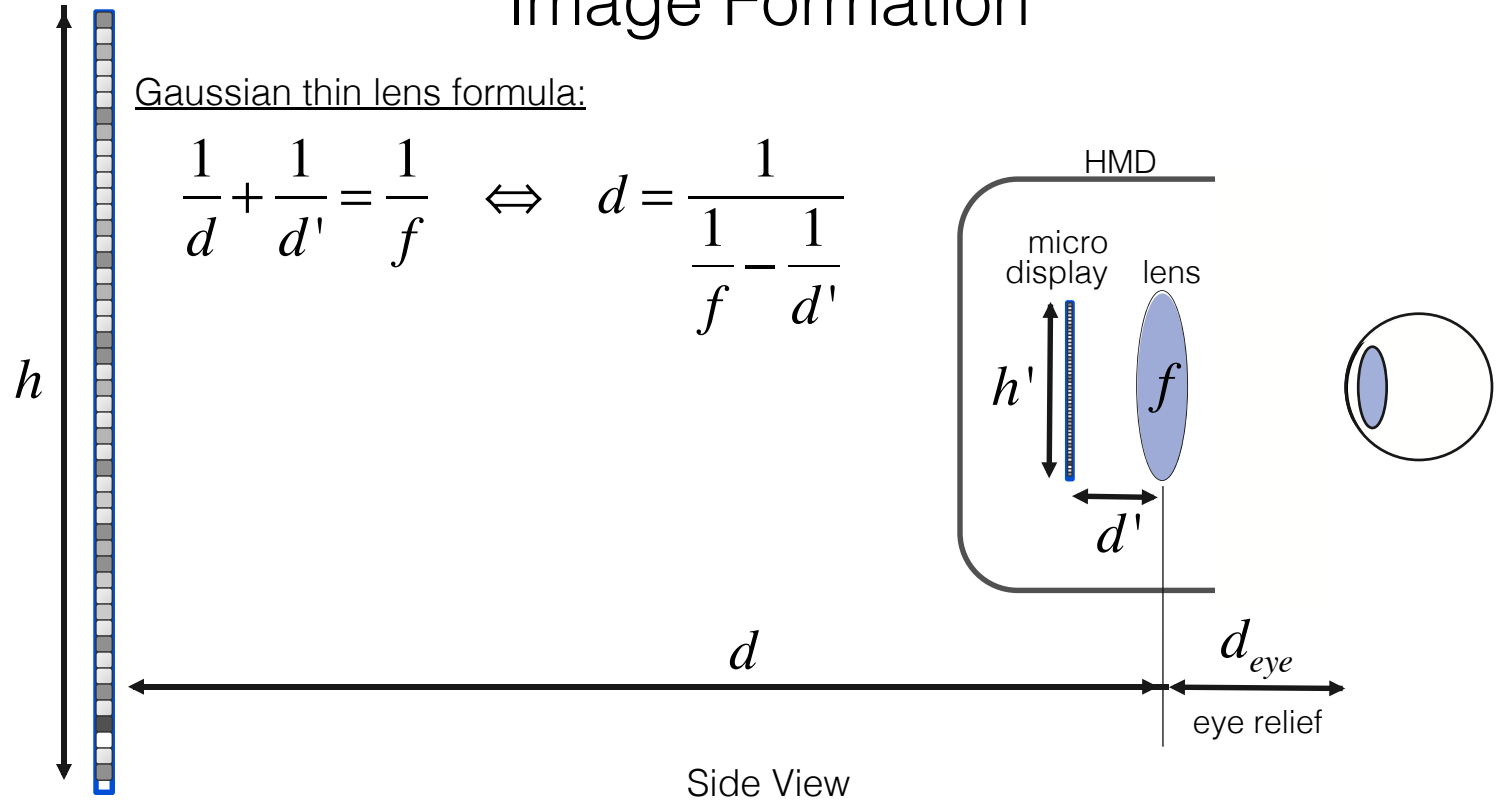


Image Formation

virtual image

Gaussian thin lens formula:

$$\frac{1}{d}+\frac{1}{d'}=\frac{1}{f} \quad \Leftrightarrow \quad d=\frac{1}{\frac{1}{f}-\frac{1}{d'}}$$



Side View

Image Formation

virtual image

Gaussian thin lens formula:

$$\frac{1}{d}+\frac{1}{d'}=\frac{1}{f} \quad \Leftrightarrow \quad d=\frac{1}{\frac{1}{f}-\frac{1}{d'}}$$

Magnification:

$$M=\frac{f}{f-d'} \quad \Rightarrow \quad h=Mh'$$

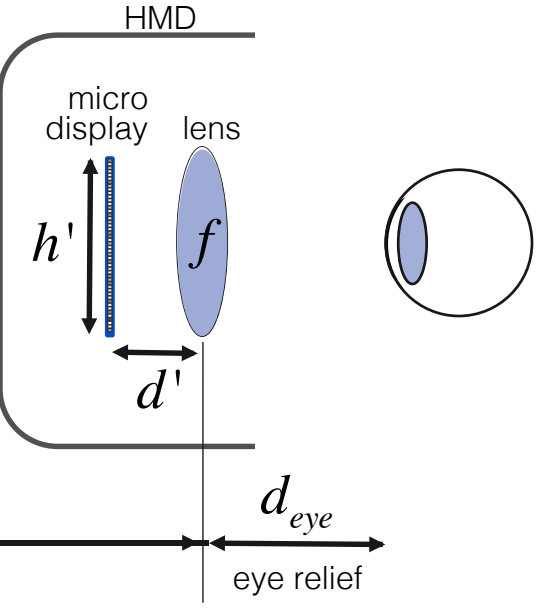


Image Formation

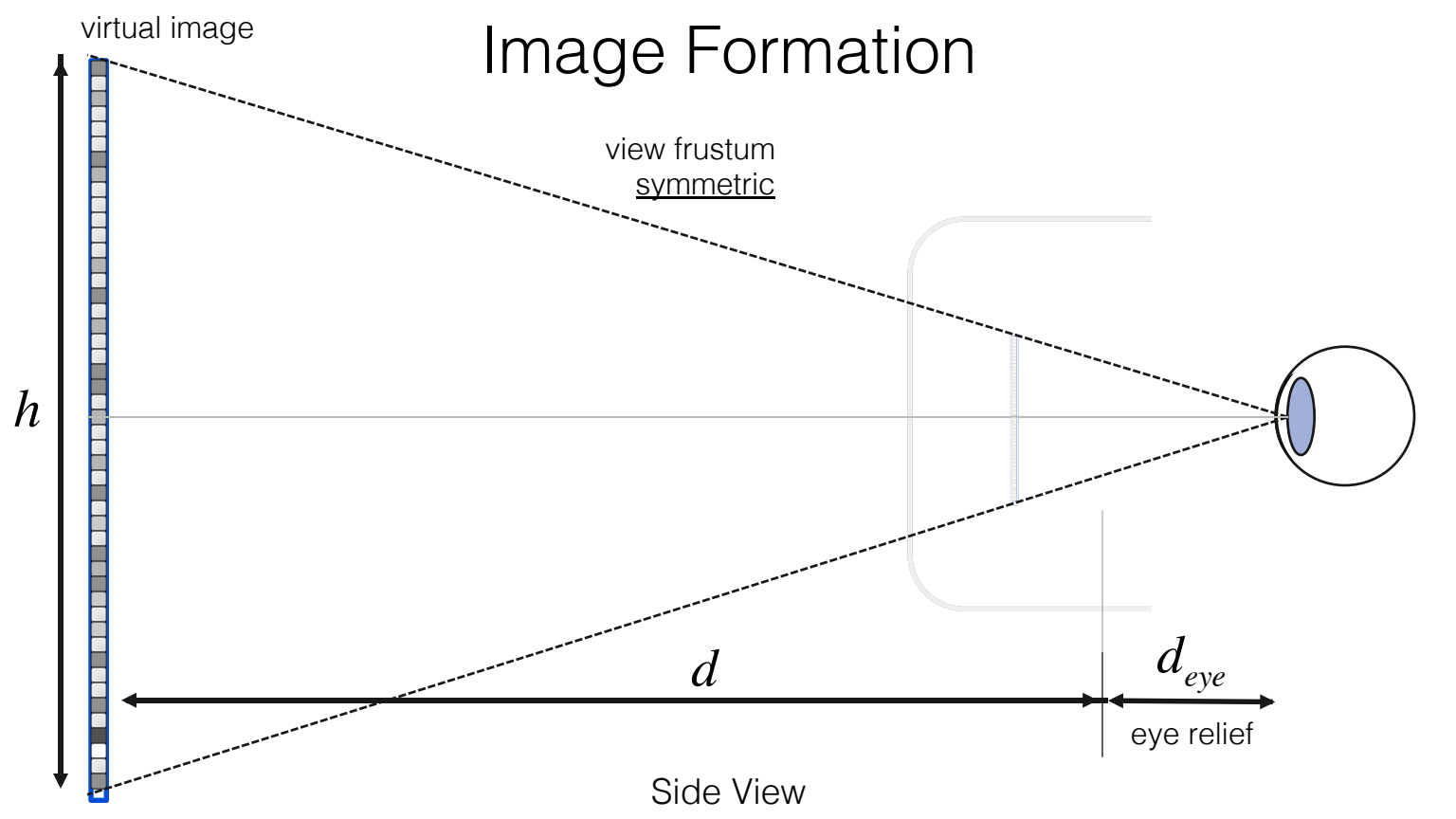


Image Formation

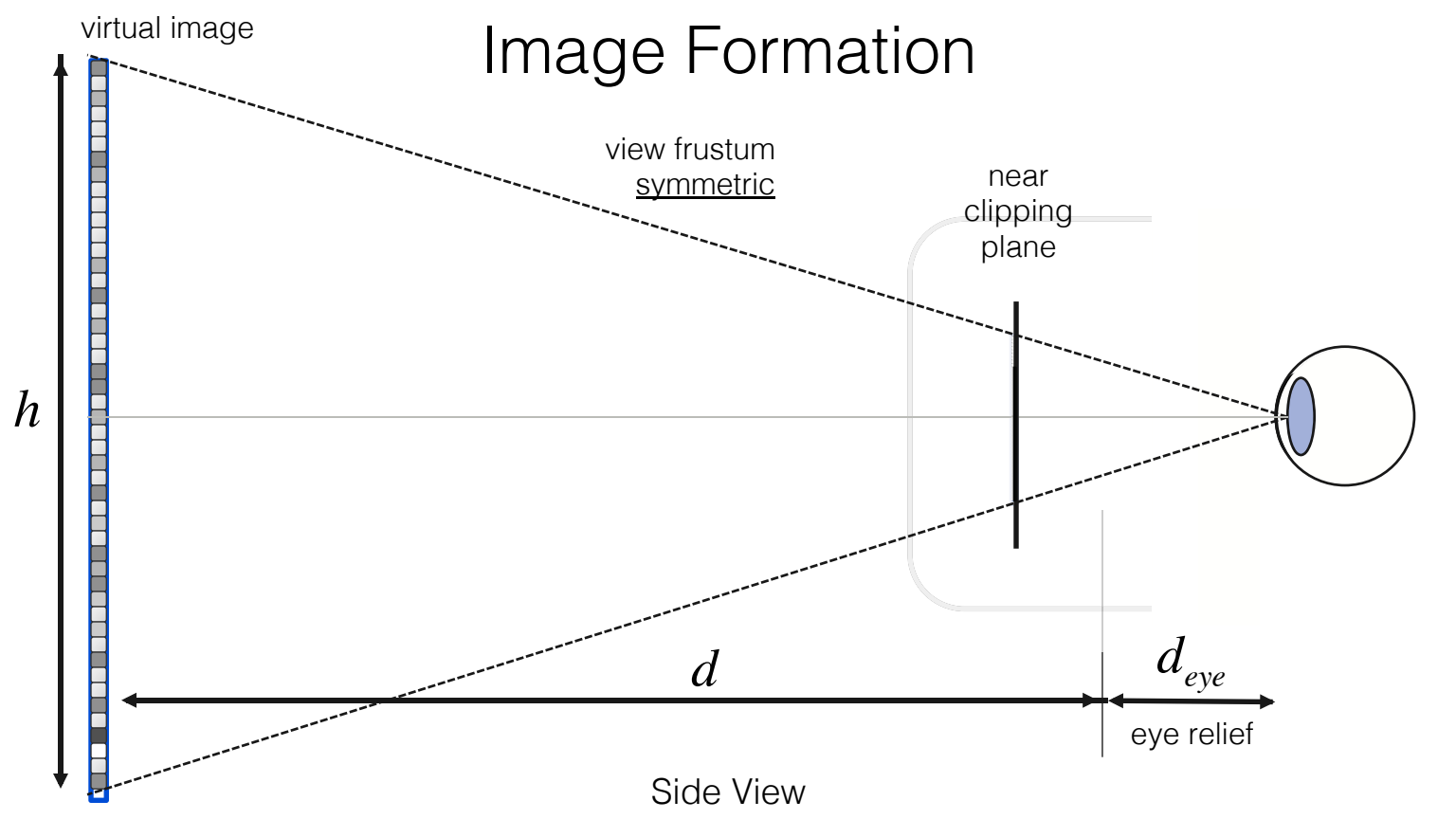


Image Formation

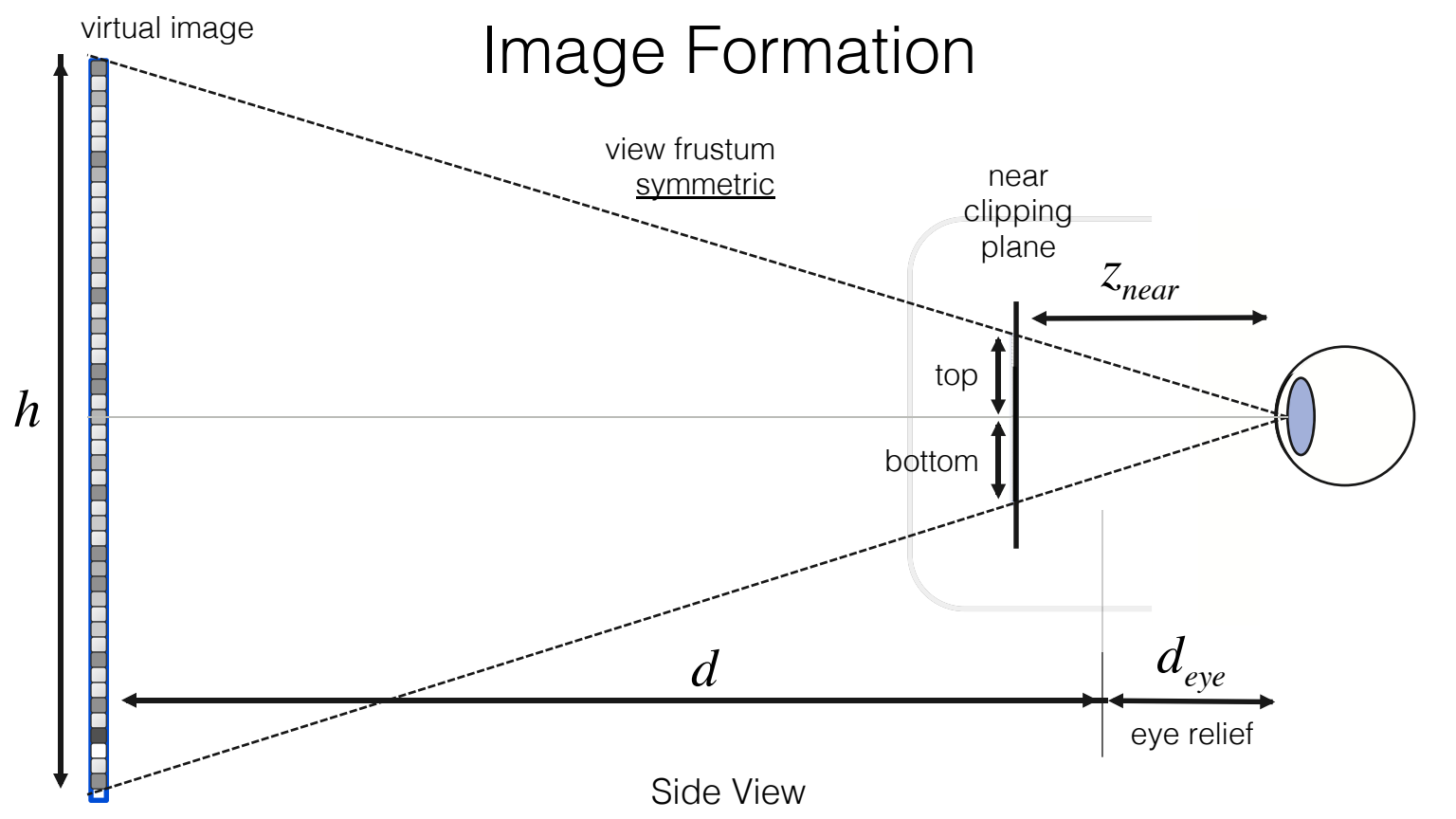


Image Formation

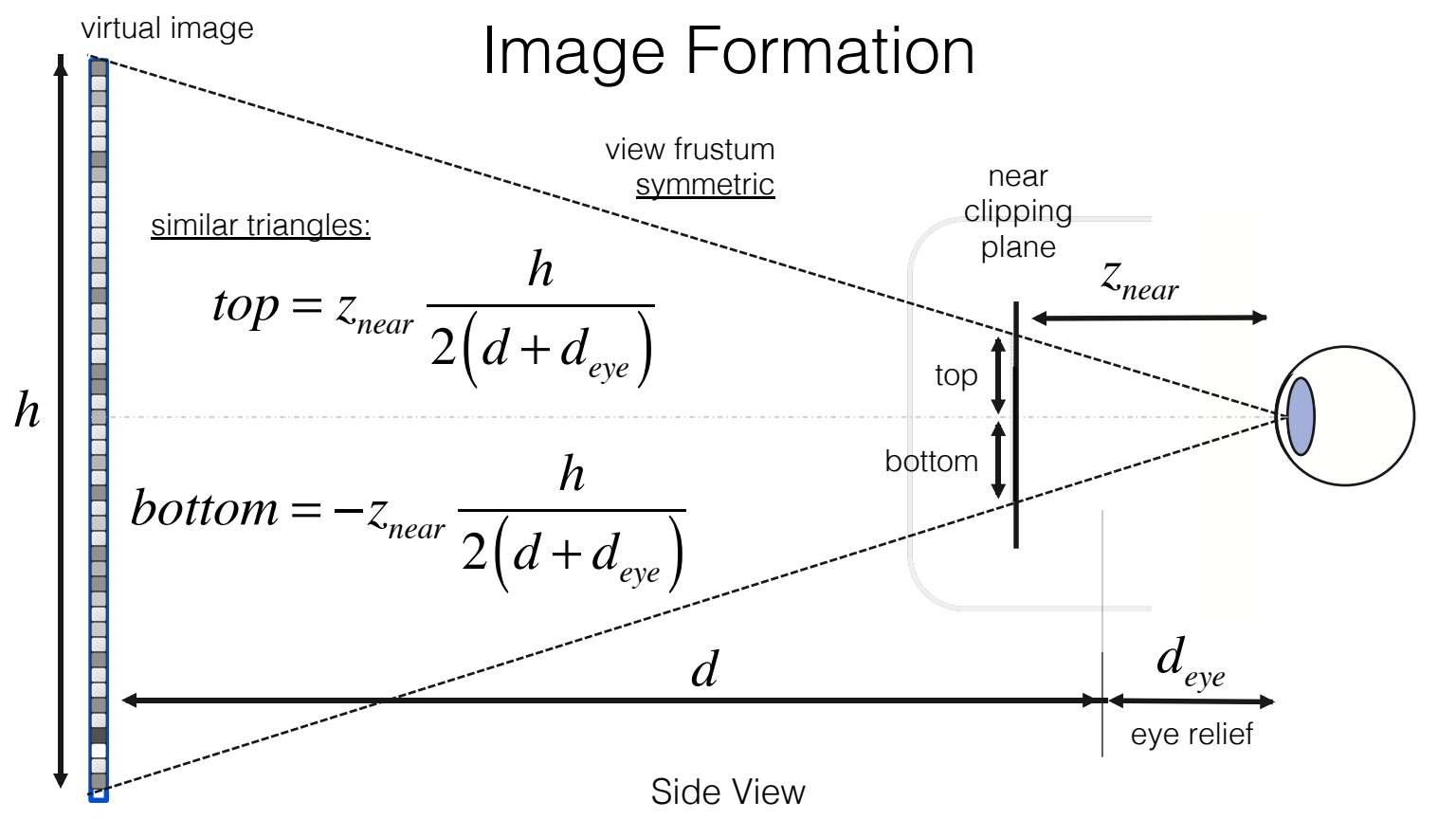
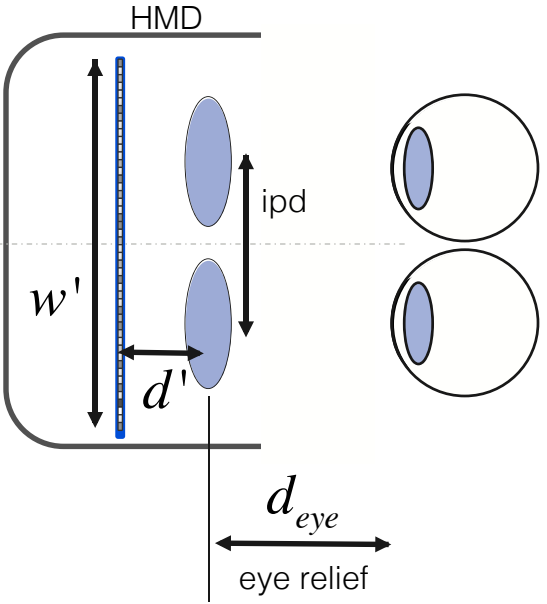
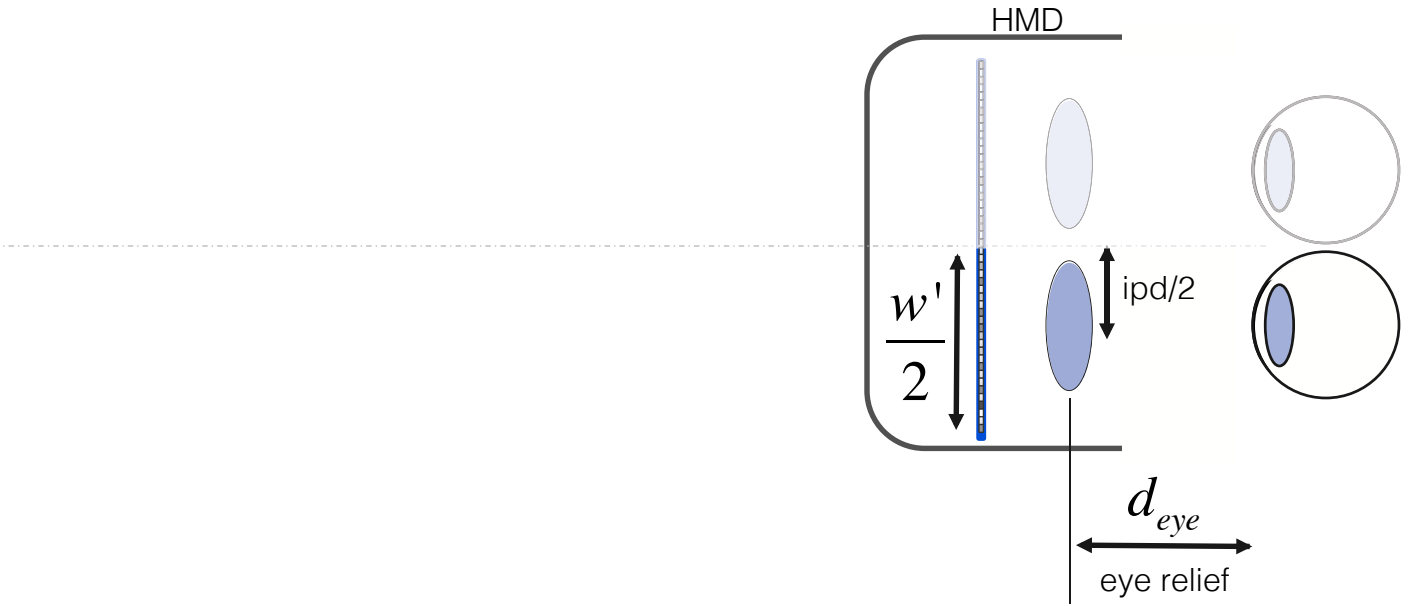


Image Formation



Top View

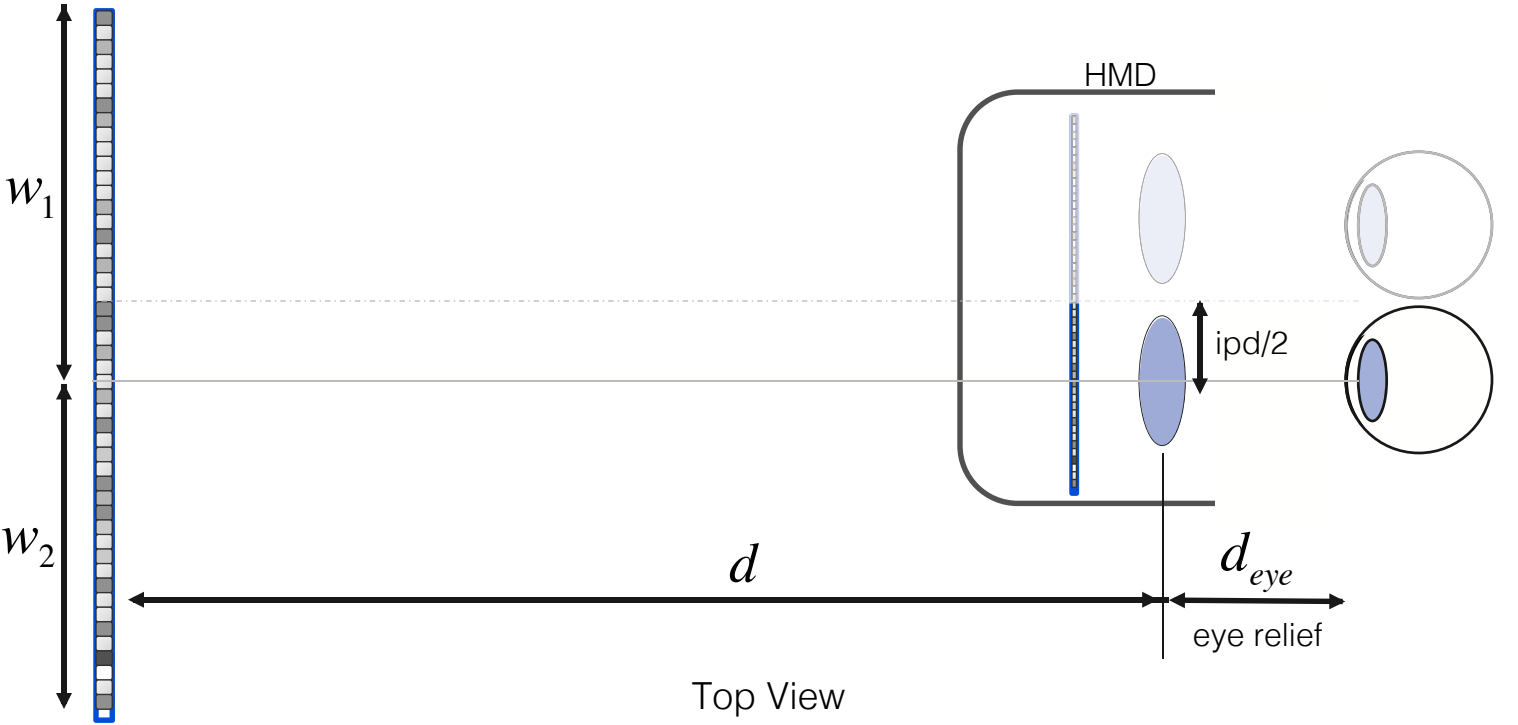
Image Formation – Left Eye



Top View

virtual image

Image Formation – Left Eye



virtual image

Image Formation – Left Eye

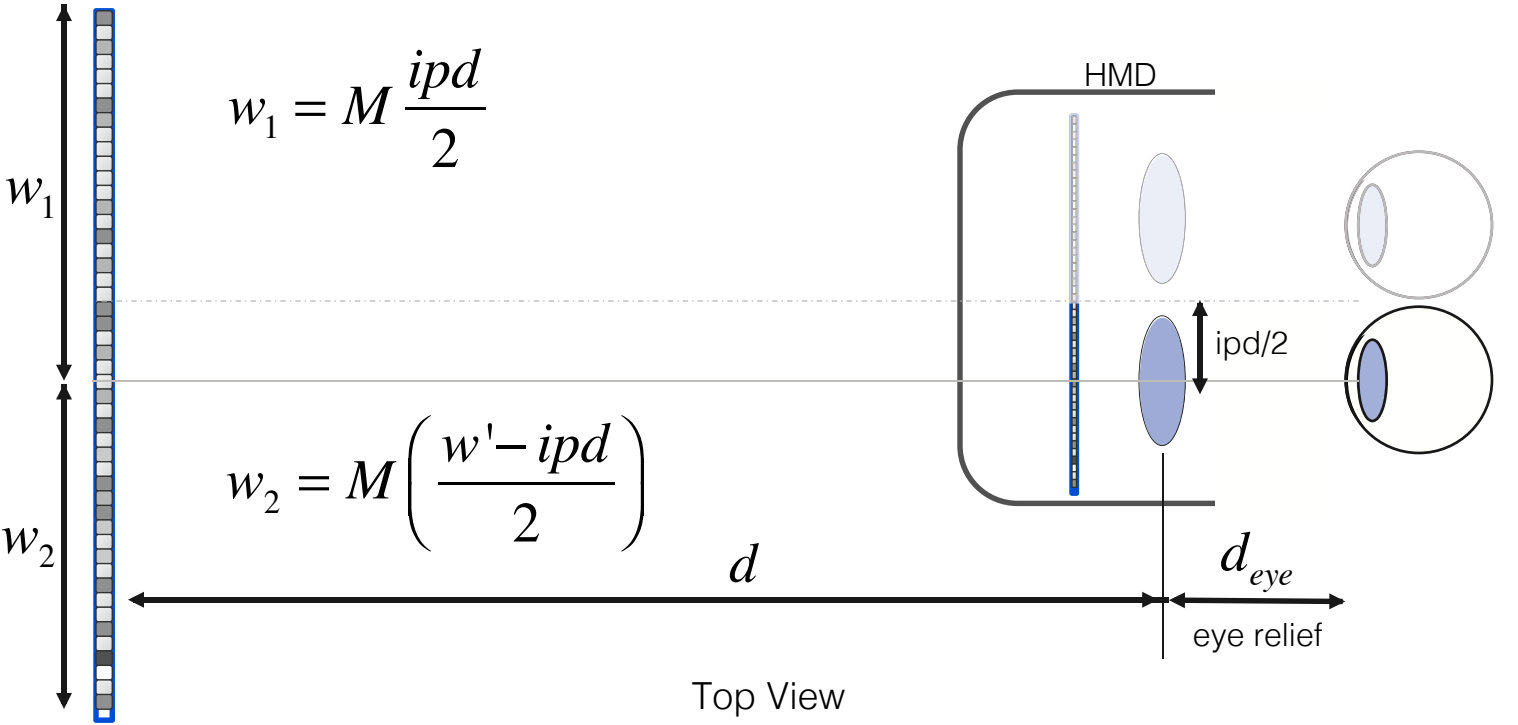


Image Formation – Left Eye

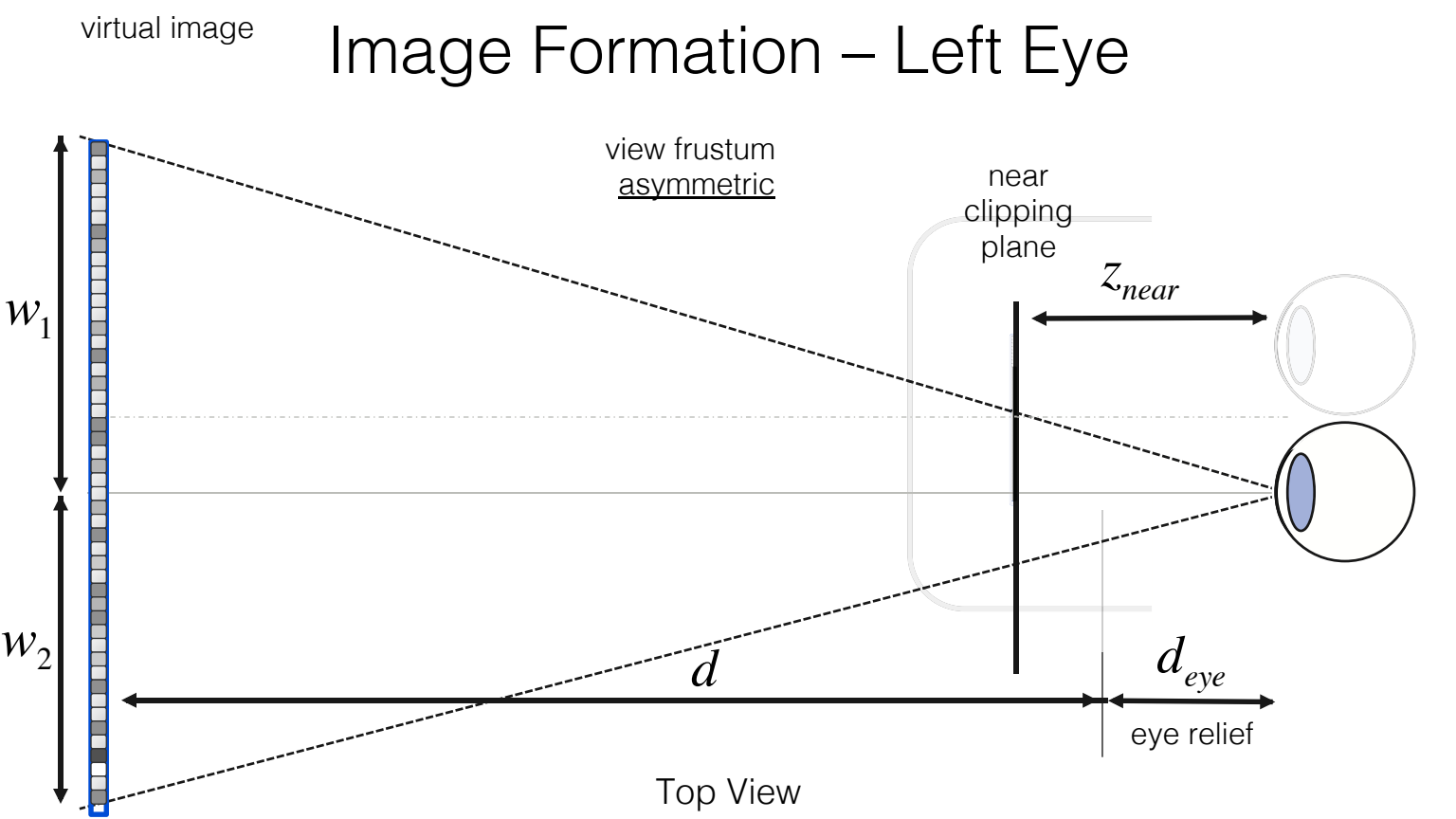


Image Formation – Left Eye

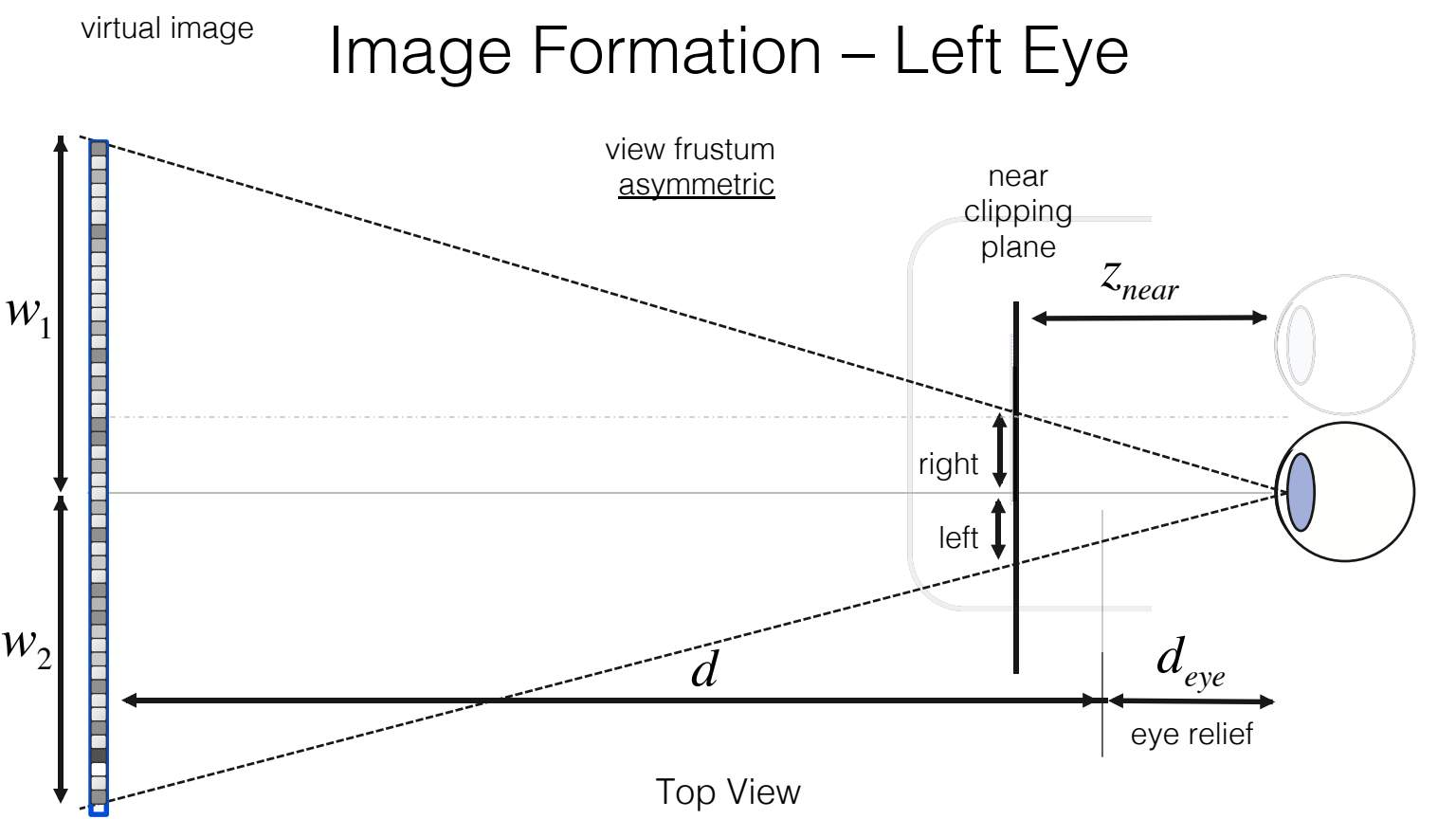


Image Formation – Left Eye

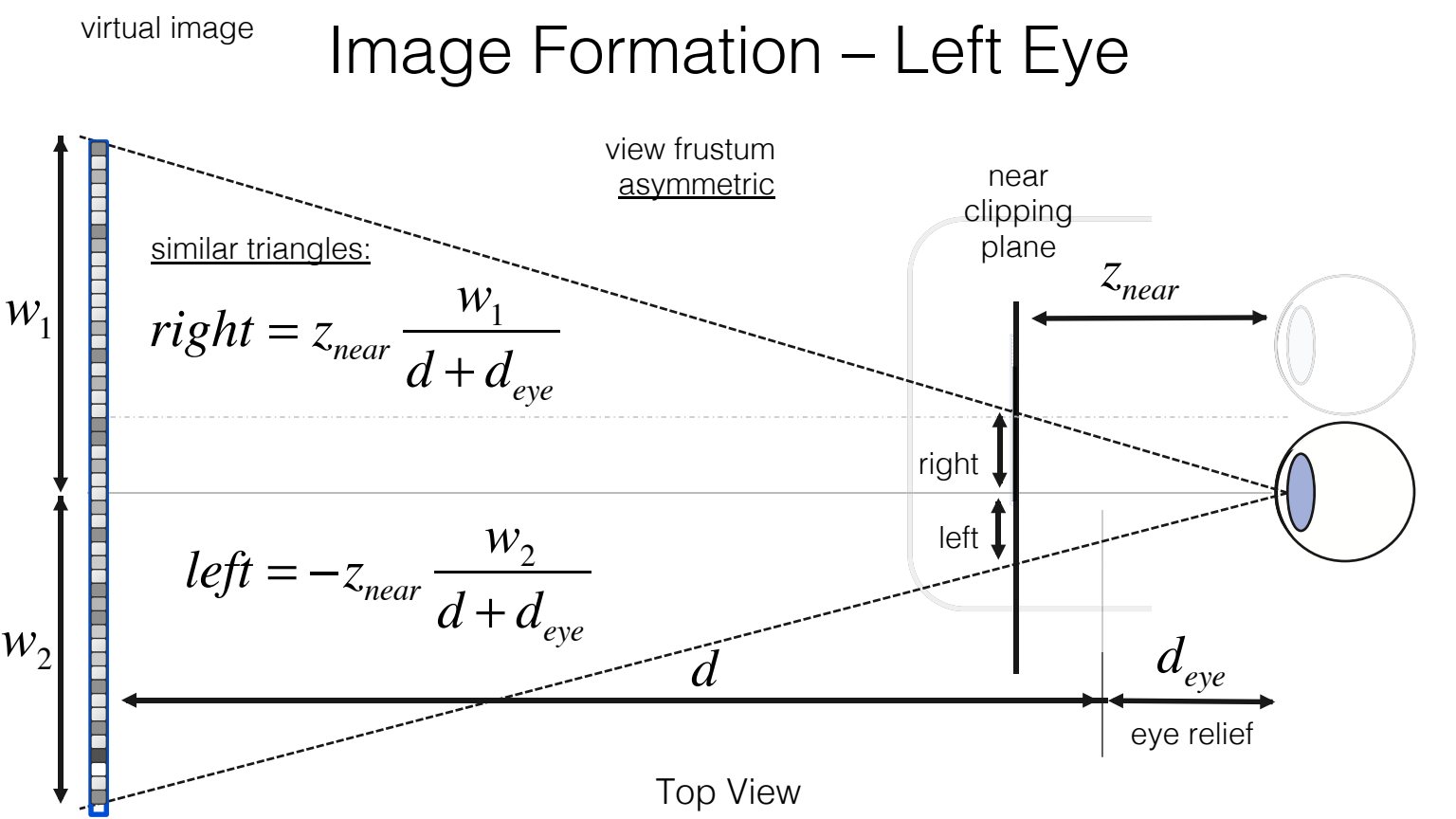


Image Formation – Right Eye

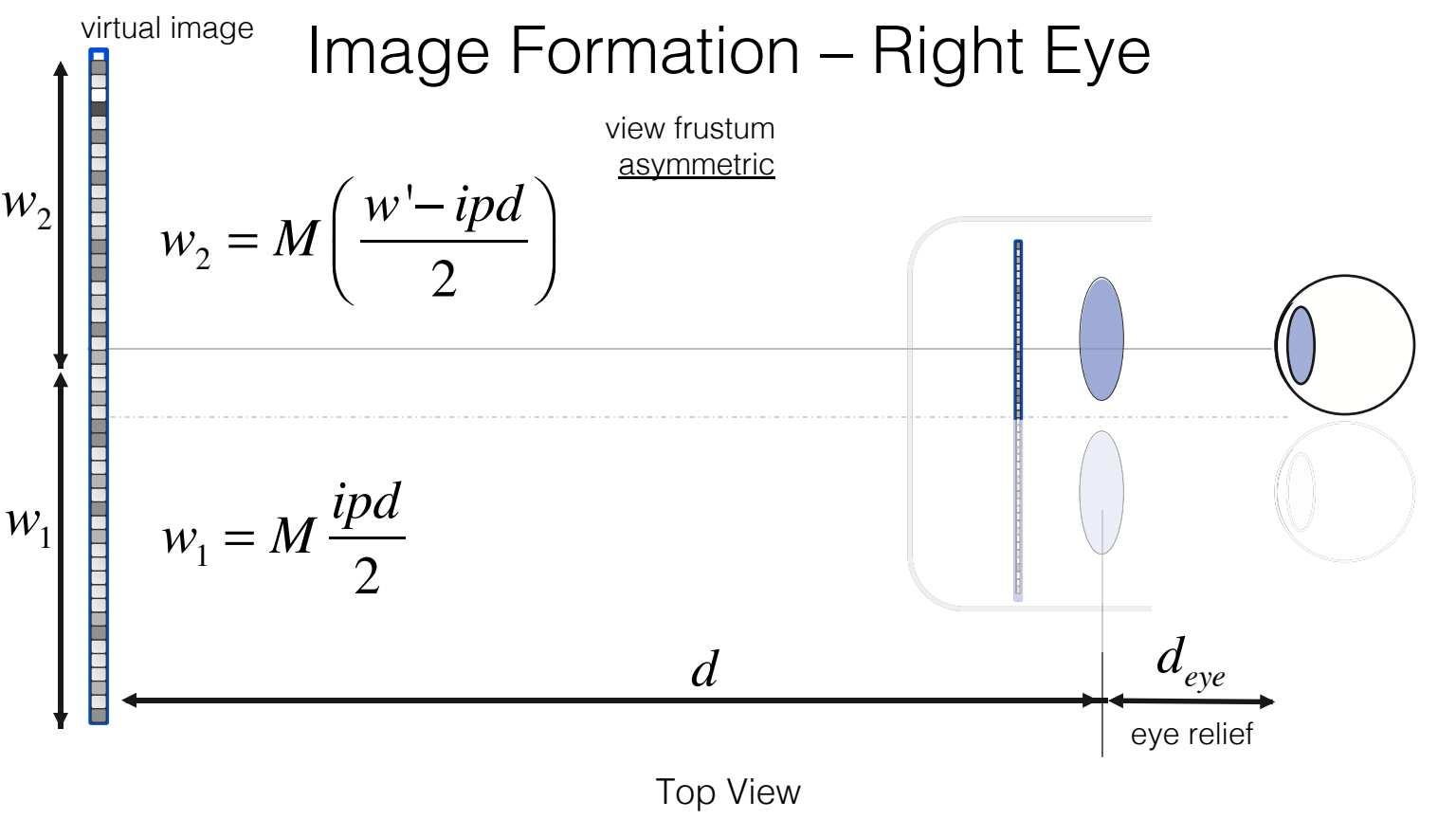
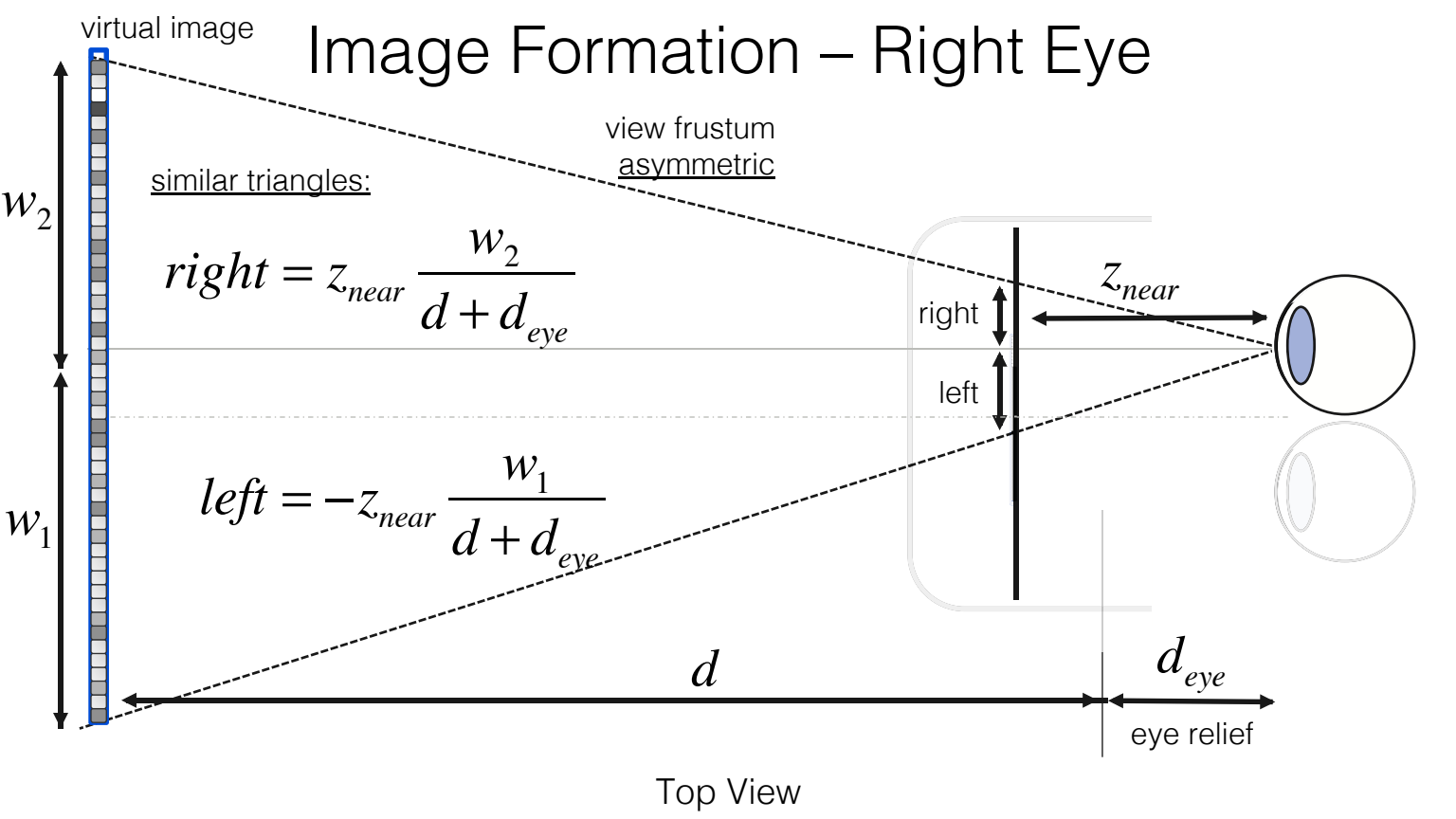
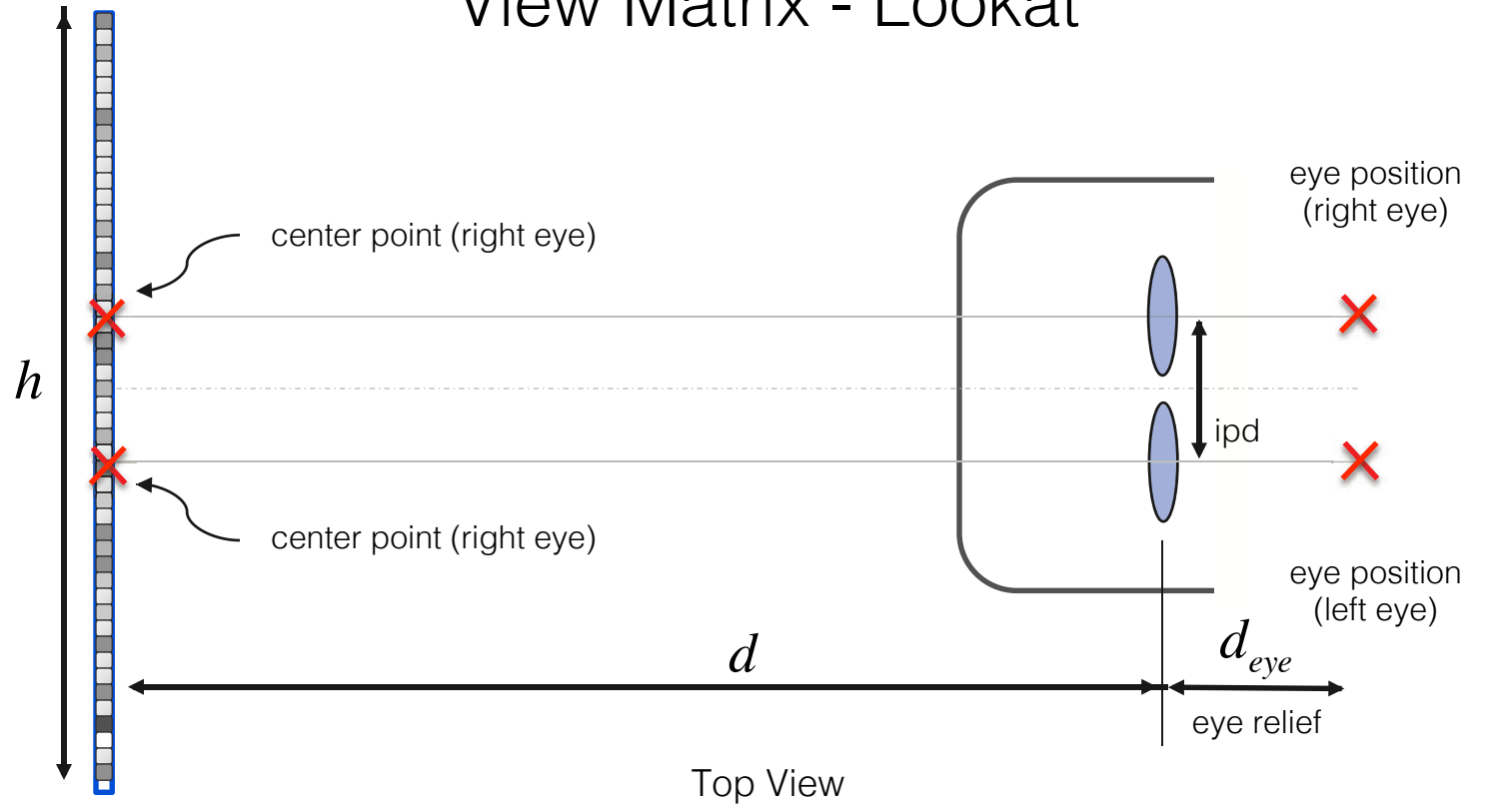


Image Formation – Right Eye



View Matrix - Lookat



Prototype Specs

- ViewMaster VR Starter Pack (same specs as Google Cardboard 1):
    - lenses focal length: 45 mm
    - lenses diameter: 25 mm
    - interpupillary distance: 60 mm
    - distance between lenses and screen: 42 mm
- Topfoison 6” LCD: width 132.5 mm, height 74.5 mm (1920x1080 pixels)

Image Formation

- use these formulas to compute the perspective matrix in WebGL

- you can use:

```
THREE.Matrix4().makePerspective(left,right,top,bottom,near,far)
THREE.Matrix4().lookAt(eye,center,up)
```

- that's all you need to render stereo images on the HMD

Image Formation for More Complex Optics

- especially important in free-form optics, off-axis optical configurations & AR

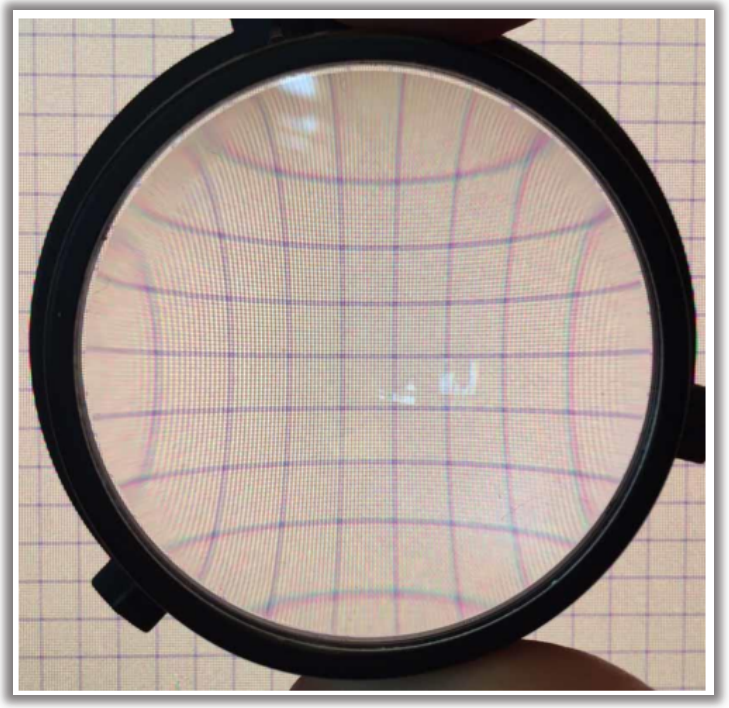
- use ray tracing – some nonlinear mapping from view frustum to microdisplay pixels

- much more computationally challenging & sensitive to precise calibration; our HMD and most magnifier-based designs will work with what we discussed so far

2. Lens Distortion Correction

All lenses introduce image distortion, chromatic aberrations, and other artifacts – we need to correct for them as best as we can in software!

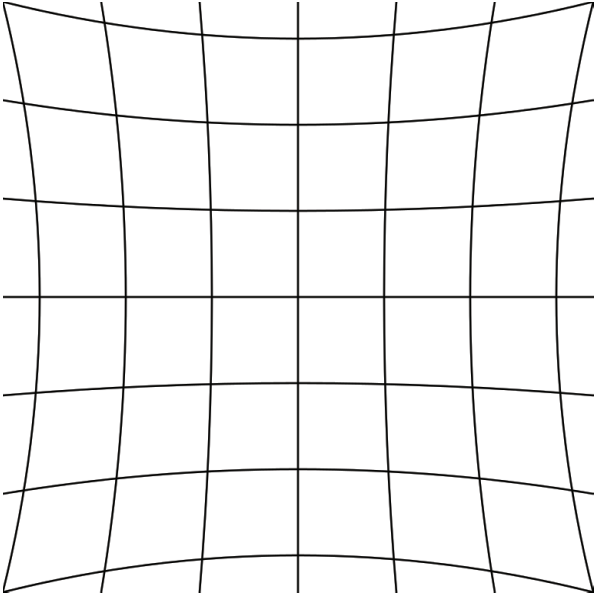
Lens Distortion



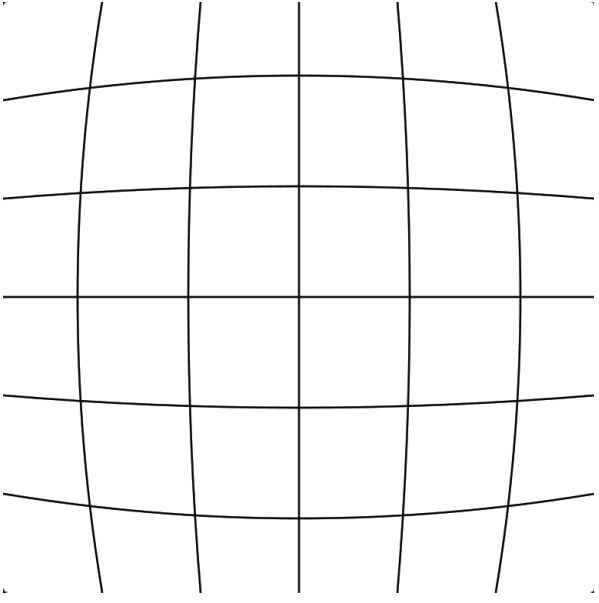
- grid seen through HMD lens
- lateral (xy) distortion of the image
- chromatic aberrations: distortion is wavelength dependent!

image from: https://www.slideshare.net/Mark_Kilgard/nvidia-opengl-in-2016

Lens Distortion

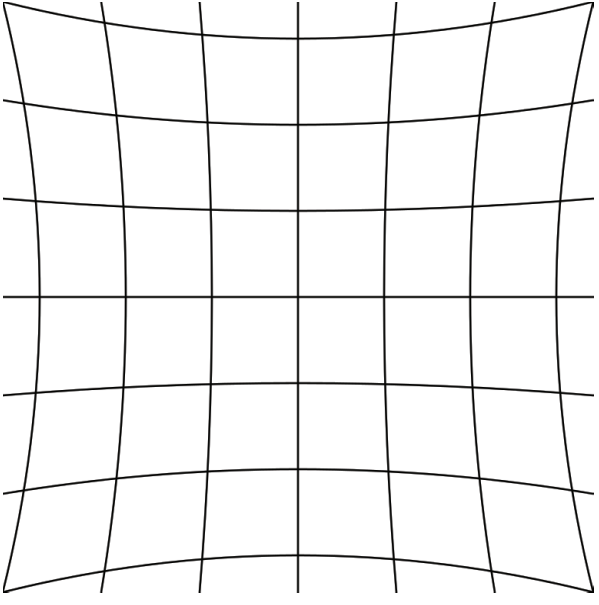


Pincussion Distortion



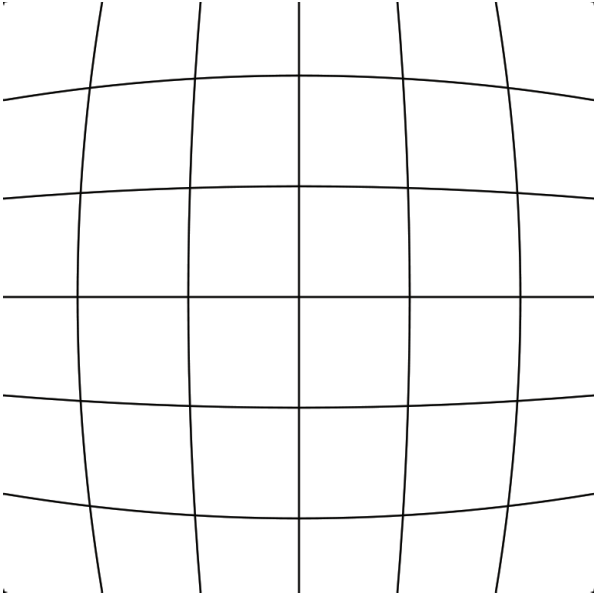
Barrel Distortion

Lens Distortion



Pincussion Distortion

optical



Barrel Distortion

digital correction

Lens Distortion

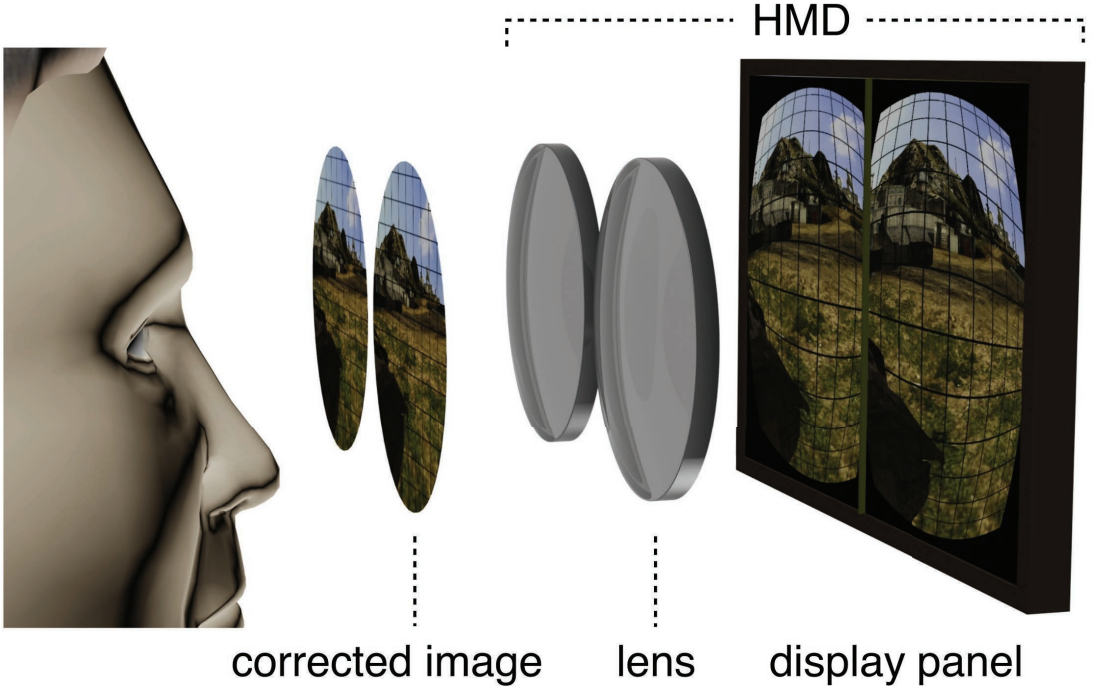
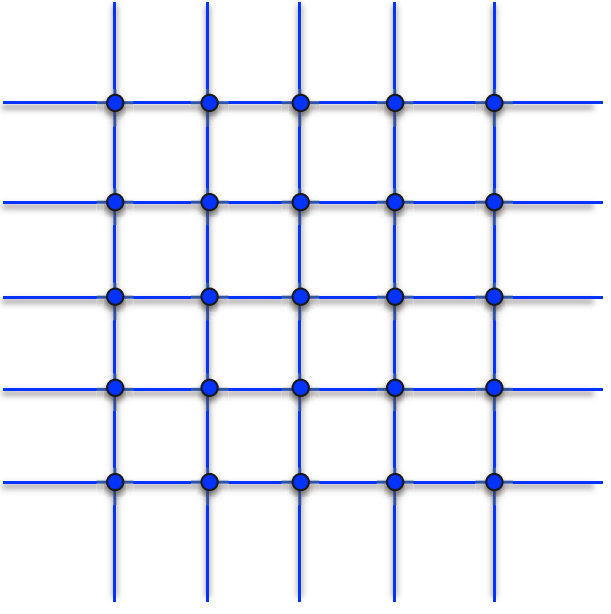


image from: https://www.slideshare.net/Mark_Kilgard/nvidia-opengl-in-2016

Lens Distortion

- x_u, y_u



Barrel Distortion

digital correction

Lens Distortion

- x_u, y_u undistorted point

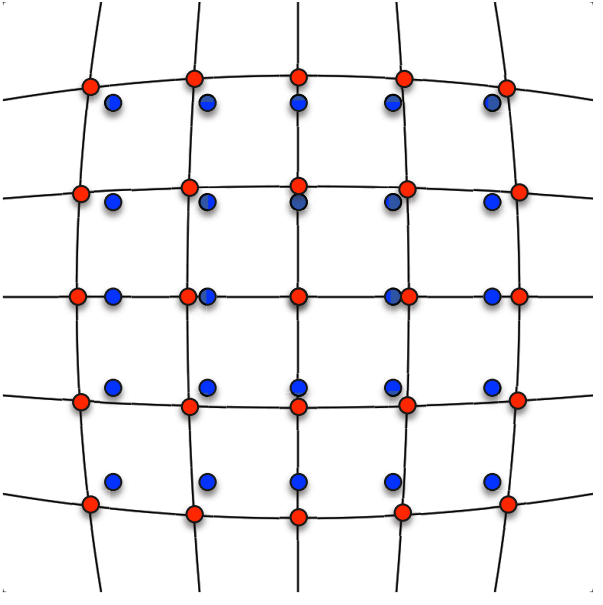
- $x_d \approx x_u \left(1 + K_1 r^2 + K_2 r^4\right)$

  $y_d \approx y_u \left(1 + K_1 r^2 + K_2 r^4\right)$

radial distance from center

$$r^2 = \left(x_u - x_c\right)^2 + \left(y_u - y_c\right)^2$$

x_c, y_c center



Barrel Distortion

digital correction

Lens Distortion

- x_u, y_u undistorted point

- $x_d \approx x_u \left(1 + K_1 r^2 + K_2 r^4\right)$

  $y_d \approx y_u \left(1 + K_1 r^2 + K_2 r^4\right)$

radial distance from center

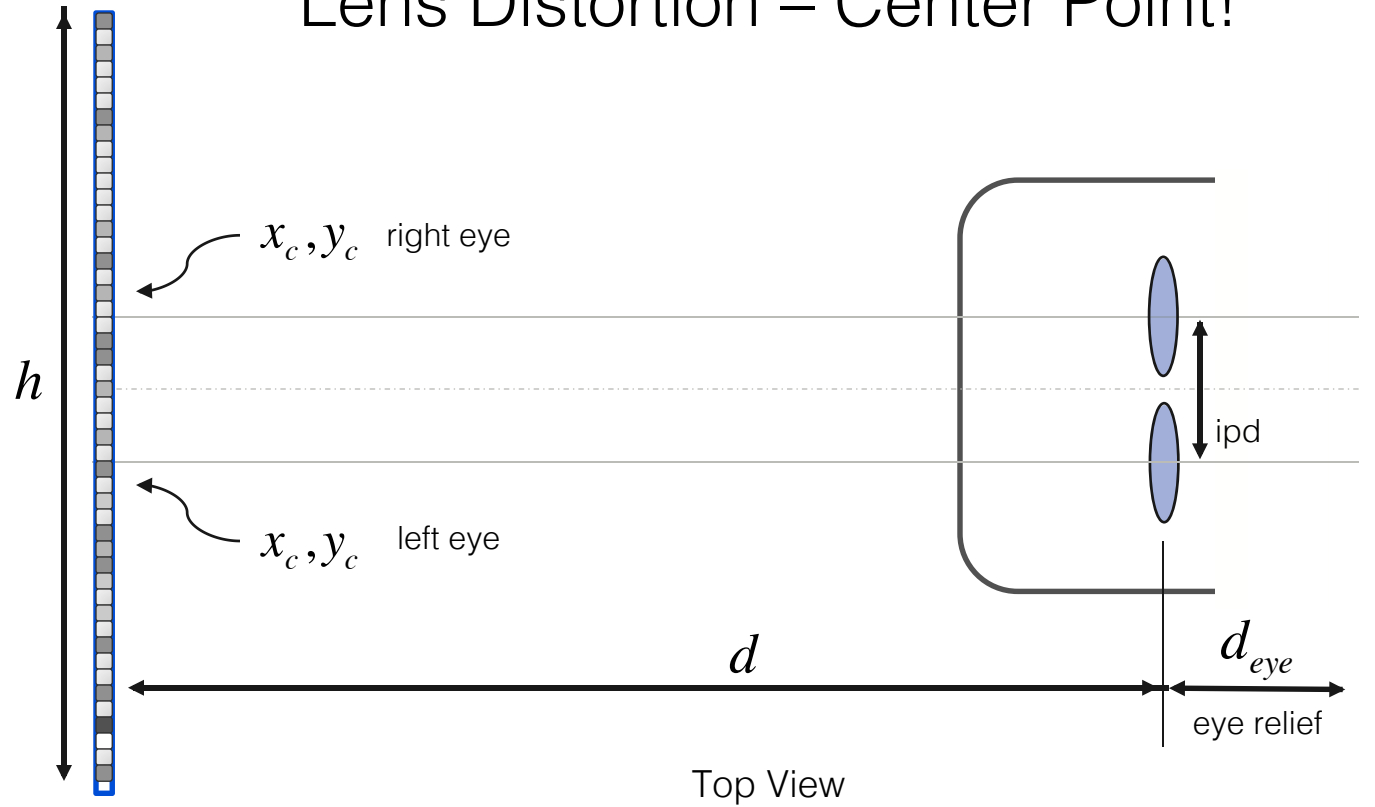
$$r^2 = \left(x_u - x_c\right)^2 + \left(y_u - y_c\right)^2$$

x_c, y_c center

NOTES:

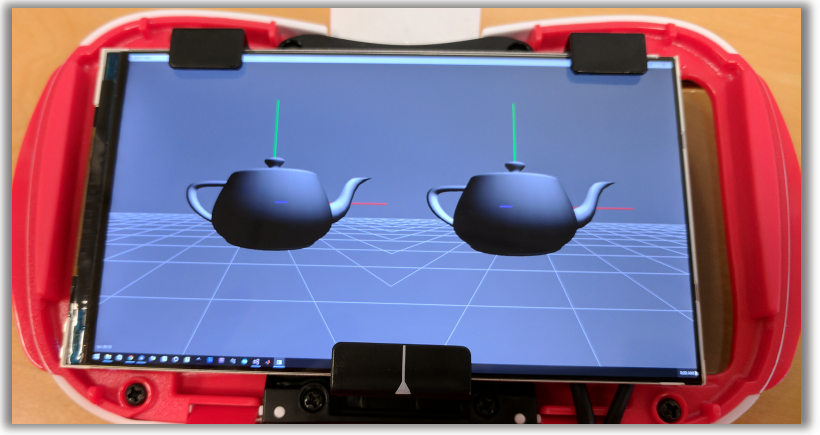
- center is assumed to be the center point (on optical axis) on screen (same as lookat center)
- distortion is radially symmetric around center point = (0,0)
- easy to get confused!
- typically implemented in fragment shader

Lens Distortion – Center Point!

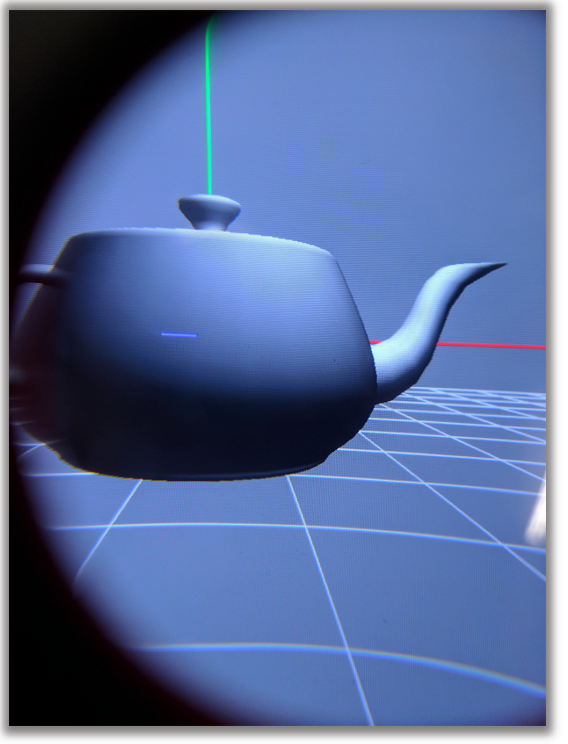


Top View

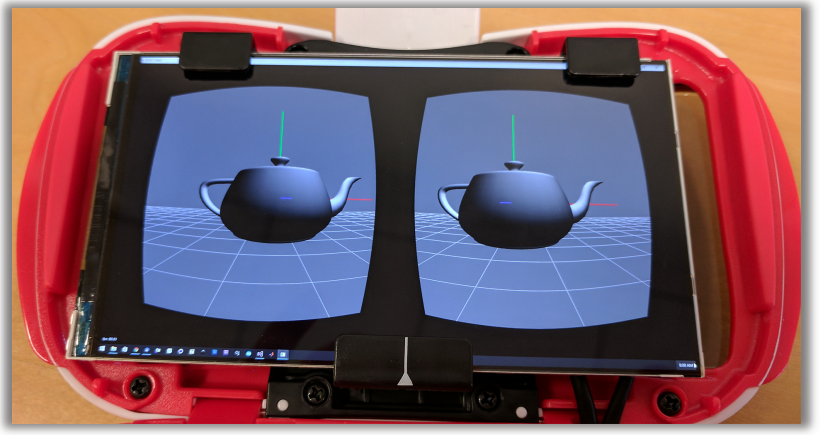
Lens Distortion Correction Example



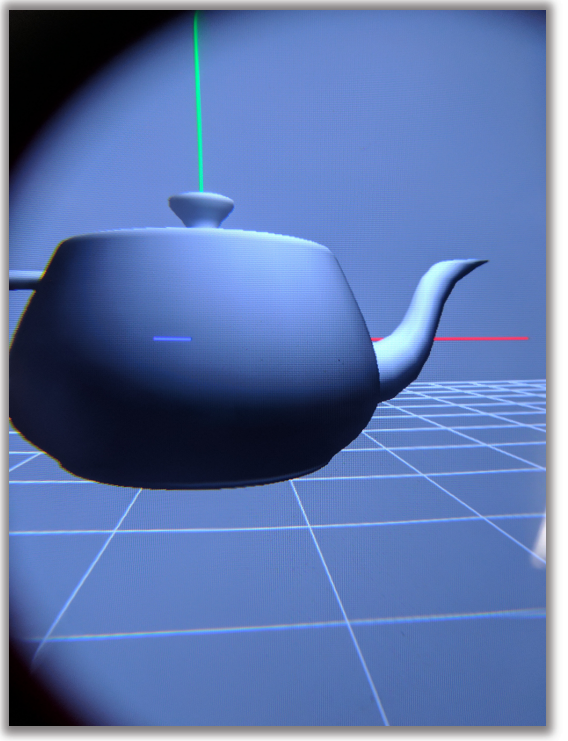
stereo rendering without lens distortion correction



Lens Distortion Correction Example

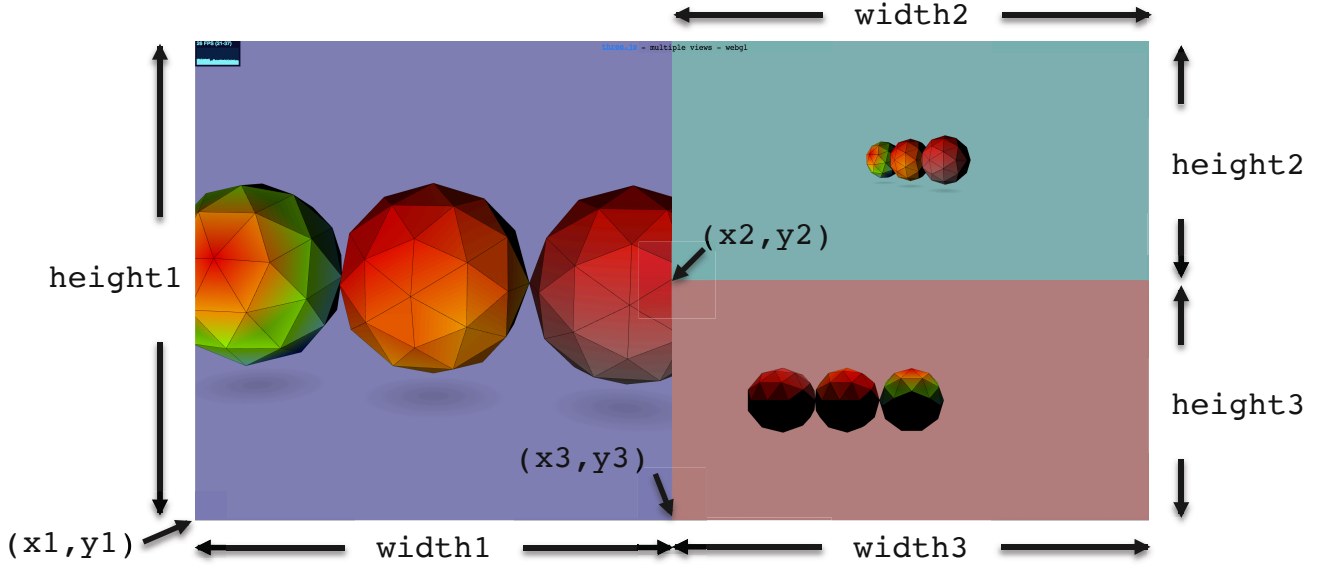


stereo rendering with lens distortion correction



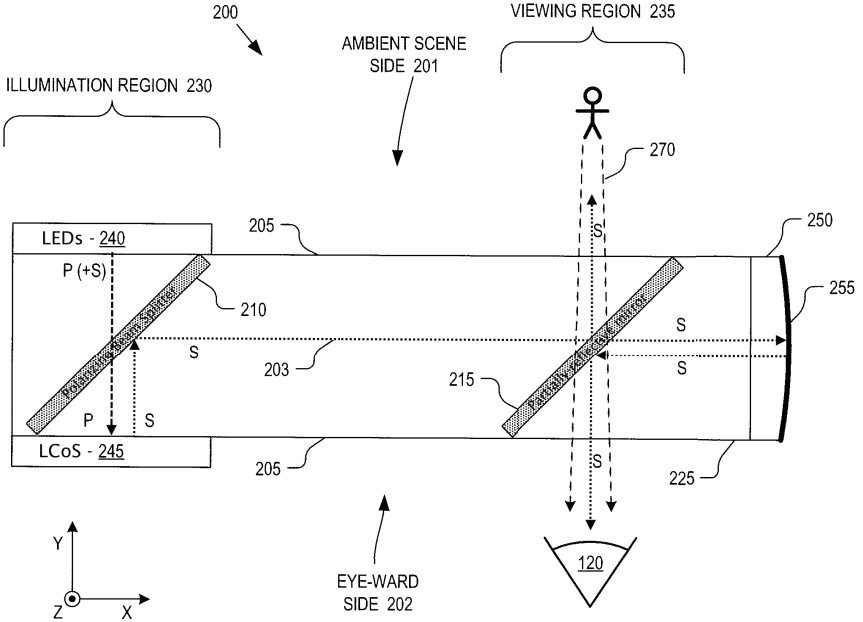
How to Render into Different Parts of the Window?

- `WebGLRenderer.setViewport(x,y,width,height)`
- `x,y` lower left corner; width, height viewport size



Next Lecture: HMD Displays Optics II

- advanced VR & AR optics
- microdisplays



drawing from Google Glass patent