Head Mounted Display Optics I

Gordon Wetzstein
Stanford University
EE 267 Virtual Reality
Lecture 7
stanford.edu/class/ee267/
Logistics

- HW3 is probably the longest homework, so get started asap if you have not done so already

- all hardware is shipped and should be with you already
Lecture Overview

1. stereo rendering for HMDs
2. field of view and visual field
3. lens distortion correction using GLSL
4. overview of microdisplay technology
Stereo Rendering for HMDs

All Current-generation VR HMDs are “Simple Magnifiers”
Image Formation

Side View

HMD

micro display

lens
Image Formation

HMD

HMD

micro display

lens

\( h' \)

\( f \)

\( d' \)

\( d_{\text{eye}} \)

eye relief

Side View
world origin is in the center of the virtual image!
Image Formation

Gaussian thin lens formula:

\[
\frac{1}{d} + \frac{1}{d'} = \frac{1}{f} \iff d = \left| \frac{1}{\frac{1}{f} - \frac{1}{d'}} \right|
\]

virtual image

h

d

d'

HMD

micro display

lens

f

h'

d eye

d eye relief

Side View
Image Formation

Gaussian thin lens formula:

\[
\frac{1}{d} + \frac{1}{d'} = \frac{1}{f} \quad \Leftrightarrow \quad d = \left| \frac{1}{\frac{1}{f} - \frac{1}{d'}} \right|
\]

Magnification:

\[
M = \frac{f}{f - d'} \quad \Rightarrow \quad h = M h'
\]
Image Formation

- virtual image
- view frustum symmetric
- world origin is in the center of the virtual image!
- eye relief $d_{\text{eye}}$
- Side View
Image Formation

virtual image

view frustum symmetric

d

near clipping plane

d_{eye}

eye relief

d

Side View

h

y

z
Image Formation

- virtual image
- view frustum symmetric
- near clipping plane
- top
- bottom
- $z_{\text{near}}$
- $d$
- $d_{\text{eye}}$
- eye relief
- Side View
Image Formation

similar triangles:

\[
\text{top} = z_{\text{near}} \frac{h}{2(d + d_{\text{eye}})}
\]

\[
\text{bottom} = -z_{\text{near}} \frac{h}{2(d + d_{\text{eye}})}
\]
Image Formation

Top View

\[ d \]
\[ d' \]
\[ w' \]
\[ ipd \]
\[ d_{\text{eye}} \]

HMD

eye relief
Image Formation – Left Eye

Virtual Image

Top View

$w_1$ $w_2$

$x$ $z$

$d$

$HMD$

$ipd/2$

$d_{eye}$

Eye Relief

Image Formation – Left Eye

Virtual Image

Top View

$w_1$ $w_2$

$x$ $z$

$d$

$HMD$

$ipd/2$

$d_{eye}$

Eye Relief
Image Formation – Left Eye

\[ w_1 = M \frac{ipd}{2} \]

\[ w_2 = M \left( \frac{w' - ipd}{2} \right) \]

d

d_{eye}

eye relief

Top View
Image Formation – Left Eye

- Virtual image
- View frustum asymmetric
- Near clipping plane
- $z_{\text{near}}$
- $w_1$
- $w_2$
- $x$
- $z$
- $d$
- $d_{\text{eye}}$
- Eye relief

Top View
Image Formation – Left Eye

- Virtual image
- Top View
- View frustum asymmetric
- Near clipping plane
- Right clipping plane
- Left clipping plane
- $w_1$, $w_2$
- $x$, $z$
- $d$
- $d_{eye}$
- Eye relief

Mathematical notation:

- $z_{near}$

Diagram elements:

- Imagery formation diagram
- Eye relief measurement
similar triangles:

right = \frac{w_1}{z_{near}} \frac{w_1}{d + d_{eye}}

left = -\frac{w_2}{z_{near}} \frac{w_2}{d + d_{eye}}

virtual image

Top View
Image Formation – Right Eye

\[ w_2 = M \left( \frac{w' - ipd}{2} \right) \]

\[ w_1 = M \frac{ipd}{2} \]

virtual image

view frustum
asymmetric

Top View

\( d \)

\( d_{\text{eye}} \)

eye relief
Image Formation – Right Eye

similar triangles:

\[
\text{right} = z_{\text{near}} \frac{w_2}{d + d_{\text{eye}}}
\]

\[
\text{left} = -z_{\text{near}} \frac{w_1}{d + d_{\text{eye}}}
\]

virtual image

Top View
View Matrix - Lookat

center point (right eye)

center point (left eye)

d

eye position (right eye)

d_{eye}

eye position (left eye)

eye relief
Prototype Specs – View-Master Deluxe VR Viewer

• roughly follows Google Cardboard 2.0:
  • lenses focal length: 40 mm
  • lenses diameter: 34 mm
  • inter-lens distance: 64 mm
  • screen to lens distance: 39 mm
  • eye relief: 18 mm

• Topfoison 6” LCD: width 132.5 mm, height 74.5 mm; 1920x1080 px OR
• Topfoison 5.5” LCD: width 120.96 mm, height 68.03 mm; 1920x1080 px
Image Formation

- use these formulas to compute the perspective matrix in WebGL

  
  - you can use:

    ```javascript
    THREE.Matrix4().makePerspective(left, right, top, bottom, near, far)
    
    THREE.Matrix4().lookAt(eye, center, up) — attention: this only does rotation, not the translation, which is required in addition to the rotation!
    
    ```

- 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
Field of View and Visual Field
Example Calculations for Field of View

- use Google Cardboard 2 lenses (f=40mm, d’=39mm, interpupillary/interlens distance = 64mm, eye relief = 18mm)
- Topfoison 6” LCD panel (132.5 x 74.5 mm)
Example Calculations for Field of View

magnification: \[ M = \frac{f}{f - d'} = 40 \]

distance lens-virtual image:
\[ d_v = \frac{1}{\frac{1}{f} - \frac{1}{d'}} = 1560\, mm \]

distance eye-virtual image:
\[ d = d_v + d_{eye} = 1578\, mm \]
Example Calculations for Field of View

horizonal field of view:

\[
fov_h = \tan^{-1}\left(\frac{M \text{ipd}}{2d}\right) + \tan^{-1}\left(\frac{M(w - \text{ipd})}{2d}\right)
\]

\[
= 39^\circ + 41^\circ = 80^\circ
\]

80° horizontal field of view is approx. 50% of the horizontal visual field of a single eye (160° total)
Example Calculations for Field of View

Virtual image of right display side

Vertical field of view:

\[
\text{fov}_v = \text{fov}_{v\text{ (superior)}} + \text{fov}_{v\text{ (inferior)}}
\]

\[
= 2 \tan^{-1} \left( \frac{M \frac{h}{2}}{d} \right) = 87^\circ
\]

87° vertical field of view is approx. 64% of the vertical visual field of a single eye (135° total)
Example Calculations for Field of View

total monocular field of view of both eyes:

\[ f_{ov_{h}}^{(total)} = 2 f_{ov_{h}}^{(temporal)} = 82° \]

82° monocular field of view is approx. 41% of the full monocular visual field of both eyes (200° total)

binocular field of view of both eyes:

\[ f_{ov_{h}}^{(total)} = 2 f_{ov_{h}}^{(nasal)} = 78° \]

78° binocular field of view is approx. 65% of the binocular visual field of both eyes (120° total)
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!
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

- $x_u, y_u$ undistorted point
Lens Distortion

- $x_u, y_u$ undistorted point
- $x_d \approx x_u \left( 1 + K_1 r^2 + K_2 r^4 \right)$
  
  $y_d \approx y_u \left( 1 + K_1 r^2 + K_2 r^4 \right)$

$x_d, y_d$ distorted point coordinates

$K_1, K_2$ distortion coefficients

$r$ normalized distance from center

$x_c, y_c$ center of optical axis

→ this is the origin, i.e. all other points are defined relative to this

Barrel Distortion
digital correction
Lens Distortion

- \( x_u, y_u \) undistorted point
- \( x_d \approx x_u \left(1 + K_1 r^2 + K_2 r^4\right) \)
- \( y_d \approx y_u \left(1 + K_1 r^2 + K_2 r^4\right) \)

\( x_d, y_d \) distorted point coordinates
\( K_1, K_2 \) distortion coefficients
\( r \) normalized distance from center
\( x_c, y_c \) center of optical axis

→ this is the origin, i.e. all other points are defined relative to this

NOTES:
- center is assumed to be the center point (on optical axis) on screen
- distortion is radially symmetric around center point
- easy to get confused!
- can implement in fragment shader (not super efficient, but easier for us)
Normalizing $r$

- $x_u, y_u$ undistorted point
- $x_d \approx x_u \left(1 + K_1 r^2 + K_2 r^4\right)$
  
  
  
  
  $y_d \approx y_u \left(1 + K_1 r^2 + K_2 r^4\right)$

un-normalized radial distance from center:

\[
\tilde{r}^2 = (x_u - x_c)^2 + (y_u - y_c)^2
\]

$x_c, y_c$ center

Calculate $\tilde{r}$ in metric units, e.g. mm. Need physical size of the pixels of your screen for this!
Normalizing $r$

$$r = \frac{\tilde{r}}{d} \text{ in } \left[ \frac{mm}{mm} \right]$$

- $\tilde{r}$: distance of a pixel from center point in mm
- $d$: distance to lens in mm

$r$ normalized, unit-less distance that we use for distortion!
Lens Distortion – Center Point!

- $x_c, y_c$: Right eye coordinates
- $x_c, y_c$: Left eye coordinates
- $h$: Height
- ipd: interpupillary distance
- $d$: Distance
- $d_{eye}$: Eye relief

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
Overview of Microdisplays
Liquid Crystal Display (LCD) - Subpixels

- TN subpixels
- IPS
- S-IPS
- IPS
LCD Backlight

- **Nanoimprinted Light Guide**
  - Light Direction

- **0.5 mm**
  - Front Light
Liquid Crystal on Silicon (LCoS)

- basically a reflective LCD
- standard component in projectors and head mounted displays
- used e.g. in Google Glass
Organic Light Emitting Diodes (OLED)

- Self emissive
- Lower persistence (can turn on and off faster than LCD/LCoS, which is great for VR)
- used e.g. VR-compatible phones, like Google’s Pixel
Digital Micromirror Device (DMD)

- developed by TI
- MEMS device
- binary states (e.g. +/- 10 degrees)
- gray-level through pulse width modulation (PWM)
- Super-fast (10-20 kHz) binary display
- More light efficient than LCD/LCoS!
Next Lecture: HMD Displays Optics II

- advanced VR & AR optics
drawing from Google Glass patent