Announcements

- **MP4** released - 4.1 due on Nov 2, 11, 18 @11am

**Final project upcoming deadlines:**

- **Nov 14**, a short video of your progress.
# Object-Order Rending

**Image-order rendering stages:**

1. Ray generation
2. Ray intersection
3. Shading: assign RGB values

**Object-order rendering stages:**

1. Rasterization
2. Depth order
3. Shading: assign RGB values
RGB Mapping: Barycentric Coordinates

\[ R = \alpha_1 R_1 + \alpha_2 R_2 + \alpha_3 R_3 \]
\[ G = \alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3 \]
\[ B = \alpha_1 B_1 + \alpha_2 B_2 + \alpha_3 B_3 \]
\[ z = \alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 \]

Examples:

\( p \)'s \( R, G, B, x, y, z \) ?

\[ p = \frac{2}{5} \]
\[ + \frac{2}{5} \]
\[ + \frac{1}{5} \]

\( p_1 \) (1,0,0)
\( p_2 \) (0,1,0)
\( p_3 \) (1,1,0)
\( p(\frac{2}{5}, \frac{2}{5}, \frac{1}{5}) \)

Red (0,0,0)

Green

Blue (0,0,0)
Texture Mappings

“paint” textures/repeating patterns over triangles using barycentric coordinates
Bump Mapping

\[ L = k_d \cdot I \cdot \max(0, n \cdot l) + k_s \cdot I \cdot \max(0, n \cdot h) + k_a \cdot I_a \]

diffuse

specular

ambient

Without bump map

With bump map

Approximation using bump mapping on a planar surface

Geometry of a bumpy surface
Normal Mapping

Example of a normal map (center) with the scene it was calculated from (left) and the result when applied to a flat surface (right).

Problems in VR!
Problems with Rendering for VR

Shading:

- Highlights need stereo perspective
- Texture maps look like painted cardboard
- Bump/normal maps look fake

(x2 rendering pipeline slowdown)

Why?
Problems with Rendering for VR

Aliasing:
- Need higher resolution than we ever needed before.
- Staircases become **escalators ⇒ more noticeable**
- Stereo causes **mismatched “fuzzy” escalators**
Some Solutions for Aliasing

Antialiasing

- Good but expensive.
- Use MSAA (multi sampling antialiasing)

Solution: "Blend colors of two textures"
The size and aspect ratio of texture may be reduced!
Challenges for Rendering in VR

Aliasing.

Optical distortion.

**Latency:** the amount of time it takes to update the display in response to head orientation and position.

GPU pipeline has been optimized for triangle output, not latency.

Latency contributors:
Problems with Rendering for VR

Optical distortion:
which distortion happens in VR: is it pincushion or barrel distortion?

Why do we need to correct for distortion?

https://www.youtube.com/watch?v=B7qrgrrHry0 3:2

disorienting, lost perception of stationarity
Correcting Optical Distortion

In hardware:

DLP (digital light processing) projects light directly into eye using MEMS.

In software:

Assumption:
distortion is radially symmetric.

Does this assumption always hold?

How would you wear HMD to break it?
Allow IPD/diopter adjustment.
Correcting Optical Distortion: Current Approach

Current approach: correct for optical distortion in software using polynomial approximations.

1. \[(x, y) \rightarrow (r, \theta)\]
   \[r = \sqrt{x^2 + y^2}\]
   \[\theta = \text{atan2}(y, x)\]

2. \[f(r, \theta): r \rightarrow r^d\]
   \[\theta \rightarrow \theta\]
   \[r^d = r + c_1 r^3 + c_2 r^5\]

3. \[f^{-1}(r, \theta): f^{-1}(\theta) = \theta\]
   \[f^{-1}(r) = \text{approximation methods}\]
Correcting Optical Distortion: Current Approach

Apply distortion to $T_{\text{can}}$ **directly**! → not how it is currently done

→ instead standard pipeline used
→ distortion is applied as postrendering step!

Optimizations: Stencil buffer
Multiresolution shading
Correcting Optical Distortion: Current Approach

Optimizations: Multiresolution shading
More Challenges for Rendering in VR

Aliasing

Optical distortion.

**Latency:** the amount of time it takes to update the display in response to head orientation and position.

- GPU pipeline has been optimized for triangle output, not latency.

Some current methods for latency reduction:
1. Lower the complexity of the virtual world.
2. Improve rendering pipeline performance.
3. Remove delays along the path from the rendered image to switching pixels.
4. Use prediction to estimate future viewpoints and world states.
5. Shift or distort the rendered image to compensate for last-moment viewpoint errors.
More Challenges for Rendering in VR

A perfect VR system:

Perfect simulation

Screen update

Shading update for all pixels

Screens are next gen! Fast/high res

Oracle provides values with no delay

Eye moves → Viewpoint update

\[ T_{\text{final}} = T_{\text{eye}} \cdot T_{\text{scene}} \cdot T_{\text{view}} \cdot T_{\text{target}} \]

Chain of transformation update

Recomputation done with no delay
More Challenges for Rendering in VR

A perfect VR system:

Before the Oculus:
60-100ms latency

Currently: 15-25 ms with 0ms "perceived" latency

Screen update

Screen are next gen! fast/high res

Eye moves → viewpoint update

$T = T_{\text{ref}} \cdot T_{\text{can}} \cdot T_{\text{eye}} \cdot T_{\text{rs}}$

Chain of transformation update

Recomputation done with no delay

Shading update for all pixels
More Challenges for Rendering in VR

\[
d = 40 \text{ pixels/°} \\
\omega = 50^\circ / \text{second} \\
\tau = 0.02 \text{ seconds}
\]

Pixel shift: \( d \cdot \omega \cdot \tau = 4 \text{ pixels} \)

What if \( \omega = 300^\circ / \text{seconds} \)? Shift \( \approx \)
Simplifying the Virtual World