OpenGL Terminology

- What is OpenGL?
- How are objects represented?
- What’s a Fragment?
- What’s a buffer?
  - How many and name them?
- What’s a texture?

OpenGL Pipeline

Blending

Learn to use the A component in RGBA color for
- Blending for translucent surfaces
- Compositing images
- Antialiasing

Opacity and Transparency

Opaque surfaces permit no light to pass through
- Transparent surfaces permit all light to pass
- Translucent surfaces pass some light
  translucency = 1 – opacity (α)

Physically Correct Translucency

Dealing with translucency in a physically correct manner is difficult due to
- The complexity of the internal interactions of light and matter
- Limitations of fixed-pipeline rendering w/ State Machine
Window Transparency

• Look out a window

• What’s wrong with that?

Screen Door Transparency

• glEnable(GL_POLYGON_STIPPLE(GL_POLYGON_STIPPLE))

Example

• Example 1
• Example 2

- Frame Buffer (assuming 32-bits)
  - Simple color model: R, G, B; 8 bits each
  - α-channel A, another 8 bits
- Alpha determines opacity, pixel-by-pixel
  - \( \alpha = 1 \): opaque
  - \( \alpha = 0 \): transparent
  - \( 0 < \alpha < 1 \): translucent
- Blend translucent objects during rendering
- Achieve other effects (e.g., shadows)
Compositing

- Back to Front
  \[ C_{out} = (1 - \alpha_c)C_{in} + \alpha_c C_c \]
- Front to Back
  \[ C_{out} = C_{in} + \alpha_c \alpha (1 - \alpha_{in}) \]
  \[ \alpha_{out} = \alpha_{in} + \alpha_c (1 - \alpha_{in}) \]

Blending

- Blending operation
  - Source: \( s = [s_r, s_g, s_b, s_a] \)
  - Destination: \( d = [d_r, d_g, d_b, d_a] \)
  - \( b = [b_r, b_g, b_b, b_a] \) source blending factors
  - \( c = [c_r, c_g, c_b, c_a] \) destination blending factors
  - \( d' = \{b_r s_r + c_r d_r, b_g s_g + c_g d_g, b_b s_b + c_b d_b, b_a s_a + c_a d_a\} \)

OpenGL Blending and Compositing

- Must enable blending and pick source and destination factors
  - \( \text{glEnable}(	ext{GL_BLEND}) \)
  - \( \text{glBlendFunc}(\text{source\_factor}, \text{destination\_factor}) \)
- Only certain factors supported
  - \( \text{GL_ZERO, GL_ONE} \)
  - \( \text{GL_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA} \)
  - \( \text{GL_DST\_ALPHA, GL\_ONE\_MINUS\_DST\_ALPHA} \)
  - See Red Book for complete list

Camera Input:

- Combine fragments with pixel values that are already in the framebuffer
  \[ \bar{c}_v = \text{src} \bar{c}_v + \text{dst} \bar{c}_p \]
  - \( \text{glBlendEquation(…)} \)
    - \( \text{GL\_FUNC\_ADD} \)
    - \( \text{GL\_FUNC\_SUBTRACT} \)
    - \( \text{GL\_REVERSE\_SUBTRACT} \)
    - \( \text{GL\_MIN} \)
    - \( \text{GL\_MAX} \)
Blending Errors

- Operations are not commutative (order!)
- Operations are not idempotent
- Limited dynamic range
- Interaction with hidden-surface removal
  - Polygon behind opaque one should be hidden
  - Translucent in front of others should be composited
  - Show Demo of the problem
  - Solution?

Blending Errors

- Interaction with hidden-surface removal
  - Draw Opaque geom first, then semi-transparent
  - Use Alpha test:
    - glAlphaFunc( GL_GREATER, 0.1 )
    - glEnable( GL_ALPHA_TEST )

Blending Errors

- Interaction with hidden-surface removal
  - Disable Z-test?
  - 2 polys: red (front) and blue (behind) on green background, 50% transparency
    1. Render background
    2. Render red poly
    3. Render blue poly
    What happens (z-test enabled)?

Blending Errors

- Interaction with hidden-surface removal
  - Disable Z-test?
  - 2 polys: red (front) and blue (behind) on green background, 50% transparency
    1. Render background
    2. Render blue poly
    3. Render red poly
    What happens (z-test disabled)?

Blending Errors

- Interaction with hidden-surface removal
  - Disable Z-test?
  - 2 polys: red (front) and blue (behind) on green background, 50% transparency
    1. Render background
    2. Render blue poly
    3. Render red poly
    What happens (z-test disabled)?
Blending Errors

- Interaction with hidden-surface removal
  - Polygon behind opaque one should be hidden
  - Translucent in front of others should be composited
- Solution?
  - Two passes using alpha testing (glAlphaFunc): 1st pass
    - alpha=1 accepted, and 2nd pass alpha<1 accepted
  - make z-buffer read-only for translucent polygons (alpha<1) with
    glDepthMask(GL_FALSE);
  - Demo

Sorting

- General Solution?
  - Just sort polygons
  - Which Space?

Sorting

- General Solution?
  - Just sort polygons
  - Which Space?
  - What About?
    - Depth Peeling

Image Dissolve?

- How to do it?
Depth Peeling Shaders

http://code.google.com/p/cuda-ldi/source/browse/trunk/src/depth_peeling/?r=17

Dual Depth Peeling

• Reduce number of passes by processing both front and back at the same time

![Figure 3. Dual depth peeling advancing fronts.](image)

Antialiasing

• Removing the Jaggies
  
  glEnable( mode )
  - GL_POINT_SMOOTH
  - GL_LINE_SMOOTH
  - GL_POLYGON_SMOOTH
  – alpha value computed by computing sub-pixel coverage
  – available in both RGBA and colormap modes

Antialiasing Revisited

• Single-polygon case first
  • Set \( \alpha \) value of each pixel to covered fraction
  • Use destination factor of \( “1 – \alpha” \)
  • Use source factor of \( \alpha \)
  • This will blend background with foreground
  • Overlaps can lead to blending errors

Antialiasing with Multiple Polygons

• Initially, background color \( C_0 \), \( a_0 = 0 \)
  • Render first polygon; color \( C_1 \) fraction \( \alpha_1 \)
    - \( C_d = (1 - \alpha_1)C_0 + \alpha_1C_1 \)
    - \( a_d = \alpha_1 \)
  • Render second polygon; assume fraction \( \alpha_2 \)
  • If no overlap (case a), then
    - \( C'_d = (1 - \alpha_2)C_0 + \alpha_2C_2 \)
    - \( a'_d = \alpha_1 + \alpha_2 \)

Antialiasing with Multiple Polygons

• Now assume overlap (case b)
  • Average overlap is \( a_1a_2 \)
  • So \( a_2 = a_1 + a_2 - a_1a_2 \)
  • Make front/back decision for color as usual
Antialiasing in OpenGL

- Avoid explicit $\alpha$-calculation in program
- Enable both smoothing and blending
  
  ```
  glEnable(GL_POINT_SMOOTH);
  glEnable(GL_LINE_SMOOTH);
  glEnable(GL_BLEND);
  glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
  ```
- Can also hint about quality vs performance using `glHint(...)`

Depth Cueing and Fog

- Another application of blending
- Use distance-dependent (z) blending
  - Linear dependence: depth cueing effect
  - Exponential dependence: fog effect
  - This is not a physically-based model

Example: Fog

- Fog in RGBA mode:
  \[ C = fC_I + (1-f)C_{I} \]
  - $f$: depth-dependent fog factor

  ```
  GLfloat color4[] = {...};
  glEnable(GL_FOG);
  glFogf(GL_FOG_MODE, GL_EXP);
  glFogf(GL_FOG_DENSITY, 6.5);
  glFogfv(GL_FOG_COLOR, color4);
  ```

Depth Cue via Fog
Example: Depth Cue

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Example

Exa