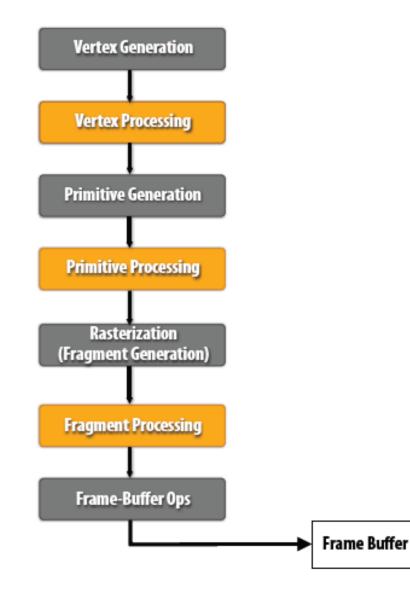
Deferred Shading

The graphics pipeline



"Forward rendering"

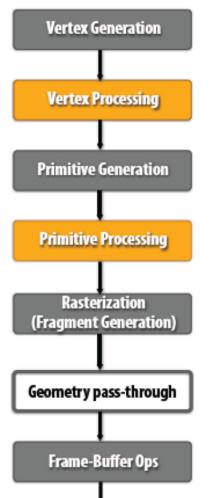
Forward rendering

- Traditional method
- Single pass
 - For each object
 - Find all lights affecting object
 - Render all lighting and material in a single shader
 - Shaderfor each material vs. light setup combination
 - Wasted shadercycles
 - Invisible surfaces / overdraw
 - Triangles outside light influence

Forward rendering

- Solution to material/light combination issue
- Multi-pass
 - For each light
 - For each object
 - Add lighting from single light to frame buffer
 - Shaderfor each material and light type
 - Wasted shadercycles
 - Invisible surfaces / overdraw
 - Triangles outside light influence
 - Lots of repeated work
 - Full vertex shaders, texture filtering

Deferred Shading Pipeline

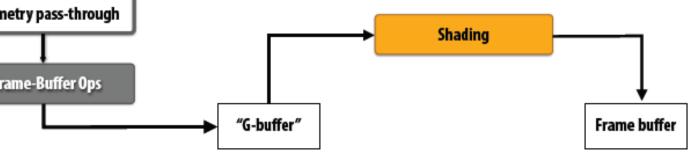


Fragment shader outputs surface properties (e.g., position, normal, material diffuse color, specular color)

Traditional pipeline does not output RGB image. Output is a 2D buffer representing information about the surface geometry visible at each pixel (a.k.a. "g-buffer")

After all geometry has been rendered, shader is executed for each sample in the G-buffer, yielding RGB values

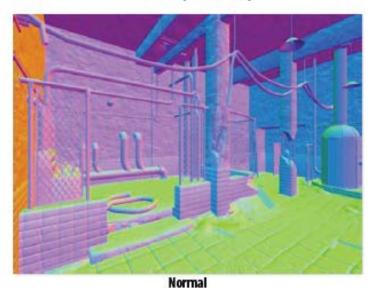
(shading is <u>deferred</u> until all geometry processing -- including all occlusion computations -- is complete)



G-buffer = geometry buffer



Albedo (Reflectance)





Depth



Specular

Deferred Rendering: procedure

- 1. For each object
 - Render surface properties into the G-Buffer
- 2. For each light and lit pixel
 - Use G-Buffer to compute lighting
 - Add result to frame buffer
- 3. Render Transparent Stuff (using forward rendering)