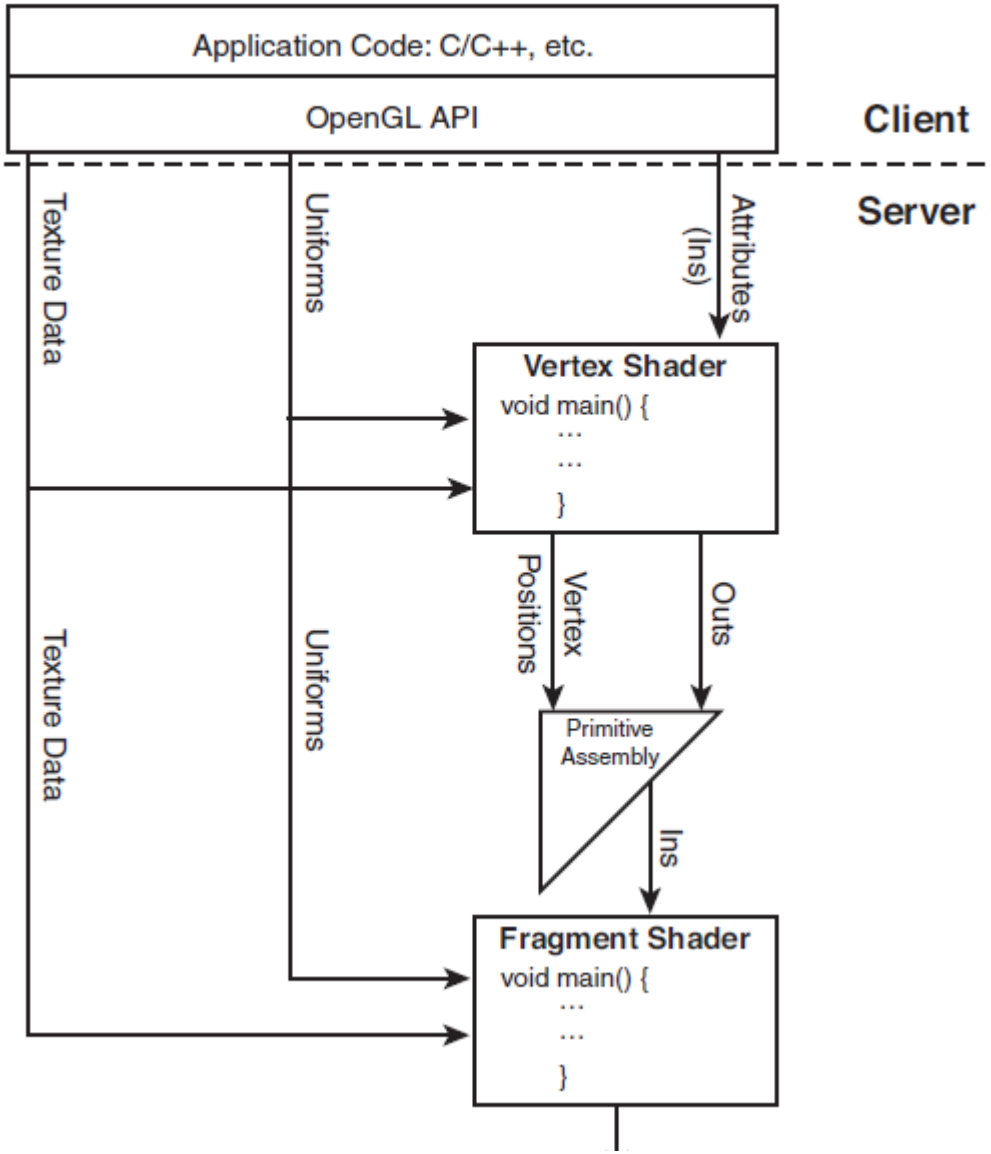
Deferred Shading Technique

Dream

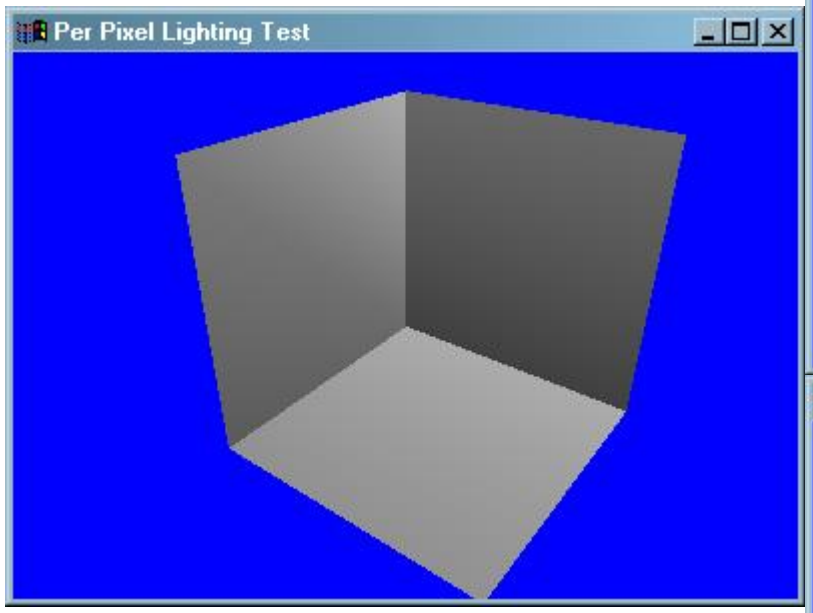
dreamc@acm.org

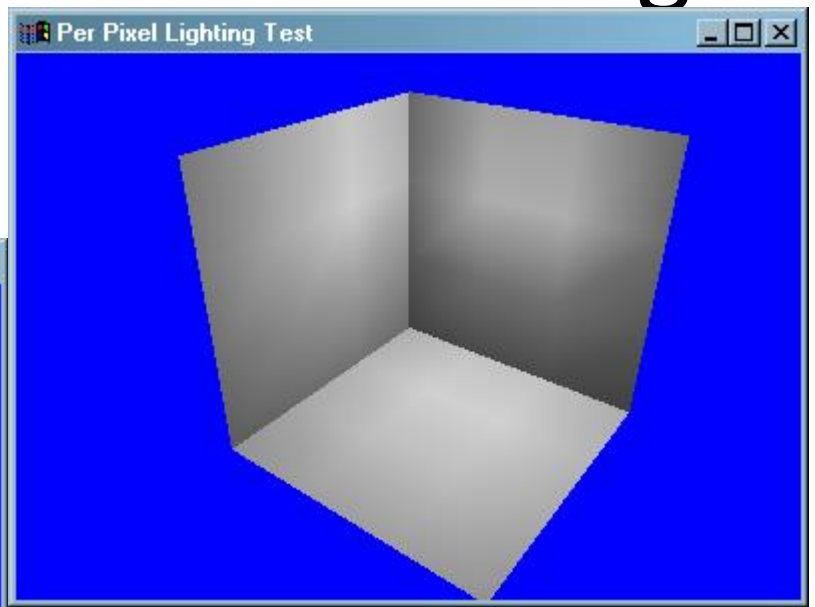
How does raster graphic APIs run?

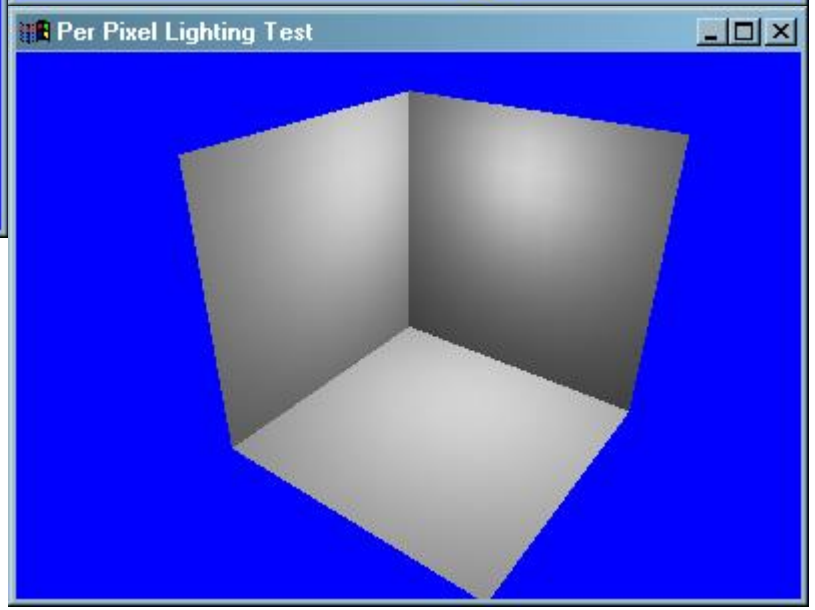


Traditional forward shading

- Per vertex
- Per pixel







Drawbacks of forward shading

For Each Light:

For Each Object Affected By Light:

framebuffer += brdf(object, light)

Algorithm complexity : O(object * light)

Deferred Shading

For Each Object:

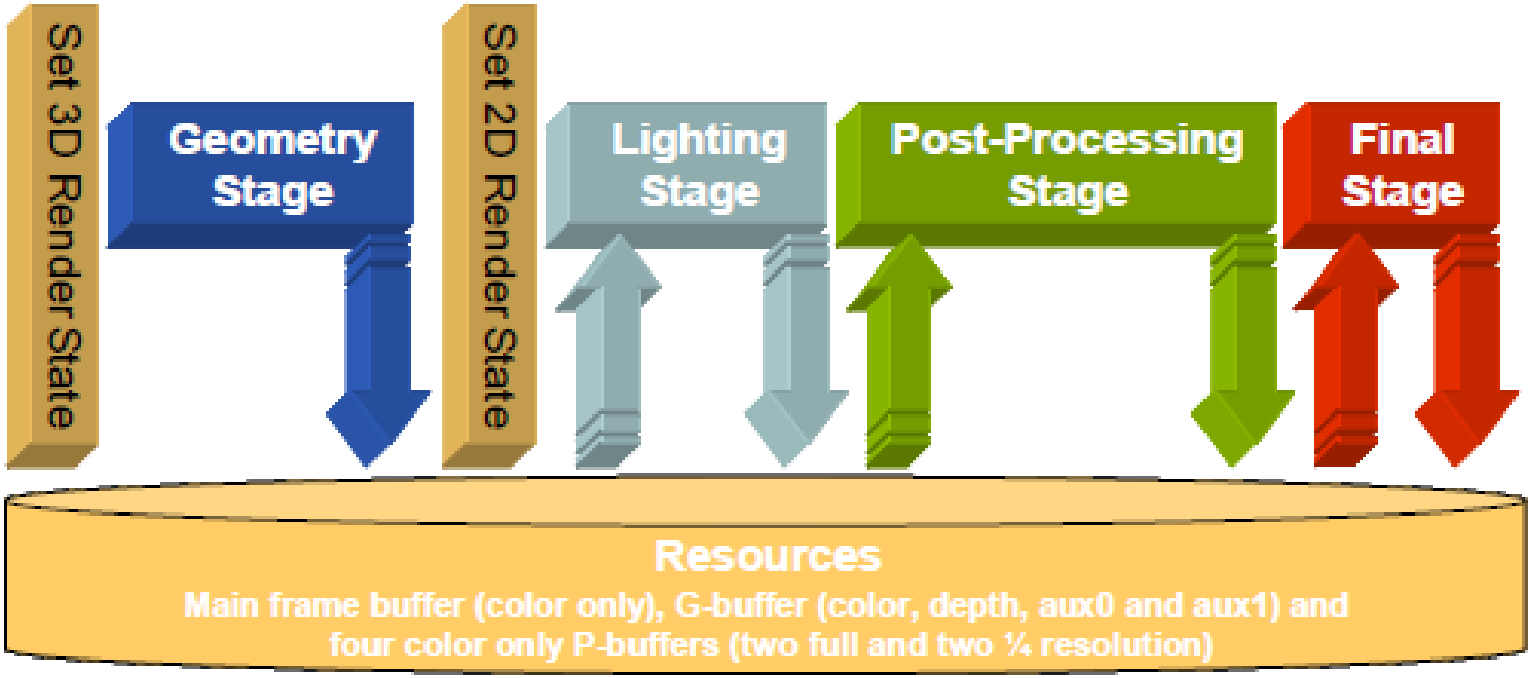
Render lighting properties to “G-buffer”

For Each Light:

framebuffer += brdf(G-buffer, light)

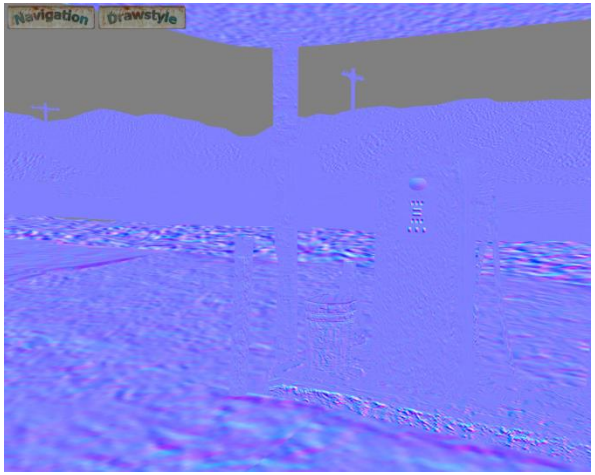
Algorithm complexity : O(object + light)

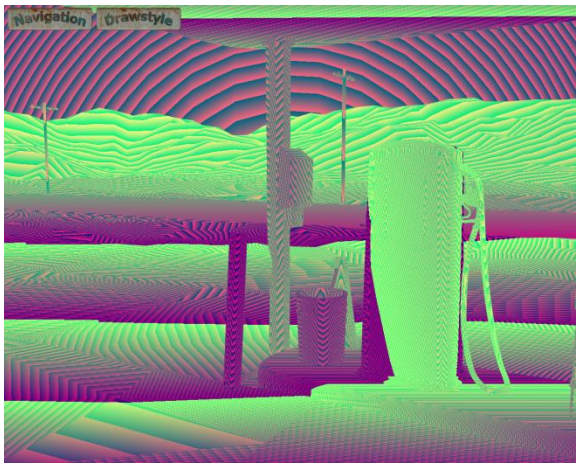
What’s Deferred shading



Geometry stage

- **G-Buffer = All necessary per-pixel lighting terms**
  - **Normal**
  - **Position**
  - **Diffuse / Specular Albedo, other attributes**
  - **Limits lighting to a small number of parameters!**





Lighting stage

- **Blend contribution from each light into accumulation buffer**
  - **Keep diffuse and specular separate**

  For each light:

  diffuse += diffuse(G-buff.N, L))

  specular += G-buff.spec *

  specular(G-buff.N, G-buff.P, L)
- **A final full-screen pass modulates diffuse color:**

  framebuffer = diffuse * G-buff.diffuse + specular

Advantages

- **Better when you have lots of local lights**
  - **Ideal case is non-overlapping lights**
  - **Shading complexity O(R)**
  - **Nighttime scenes with many dynamic lights!**

Disadvantages

- AA and alpha blending are troubles

Still need works

- High performance AA algorithm
- Translucent & transparent object
- Data compression(Bandwidth issue)