

CS4803DGC Design and Programming of Game Consoles

Spring 2011

Prof. Hyesoon Kim



**Georgia
Tech**

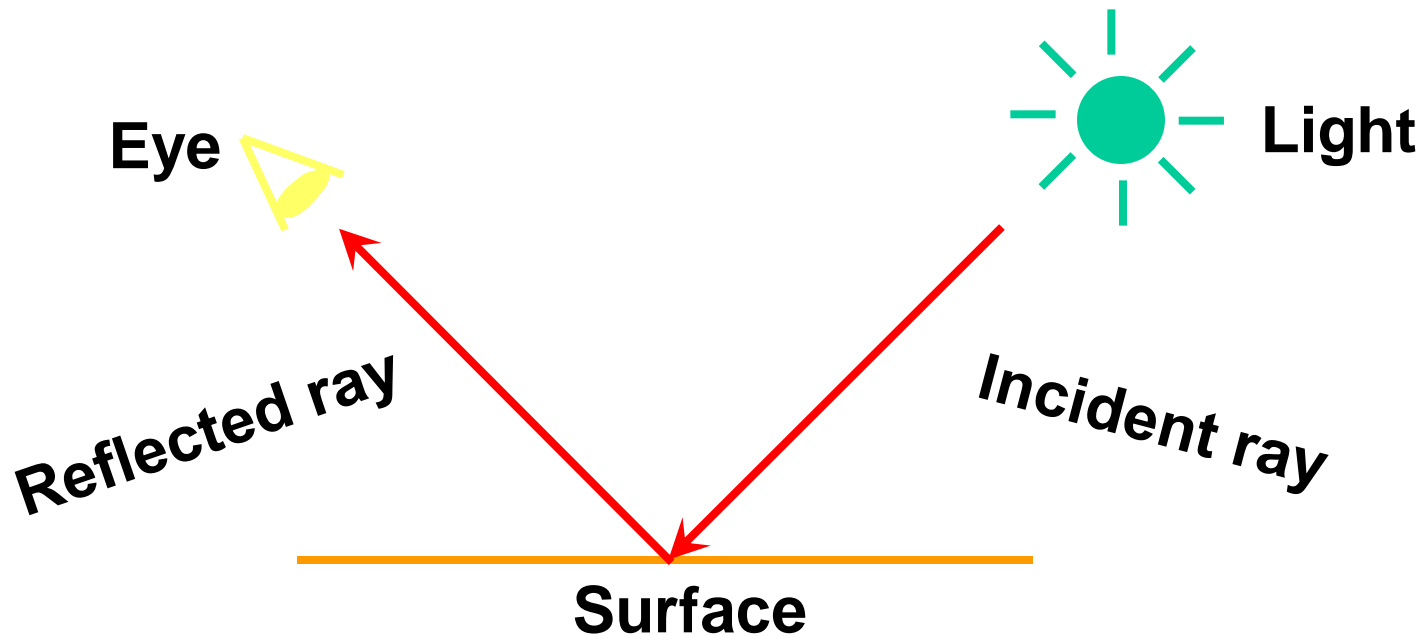


College of
Computing



Ray Tracing Problem

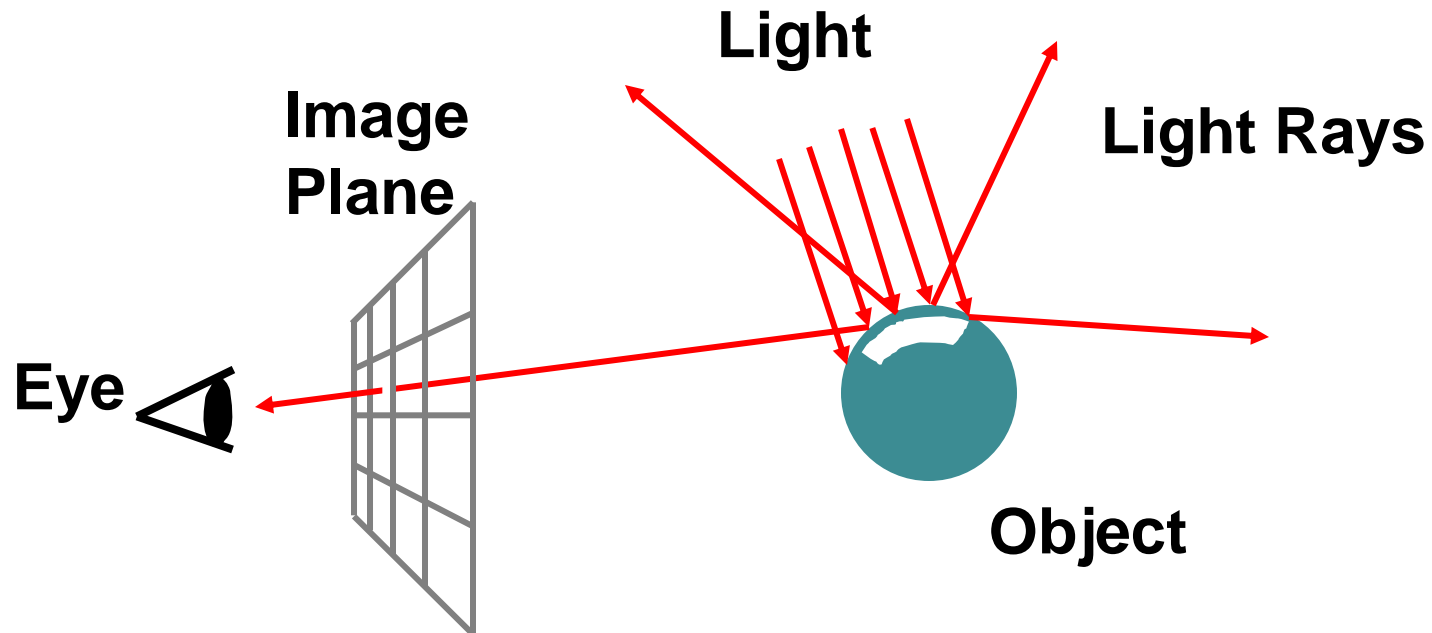
- What is it?
- Simulate light rays from light source to eye





“Forward” Ray Tracing

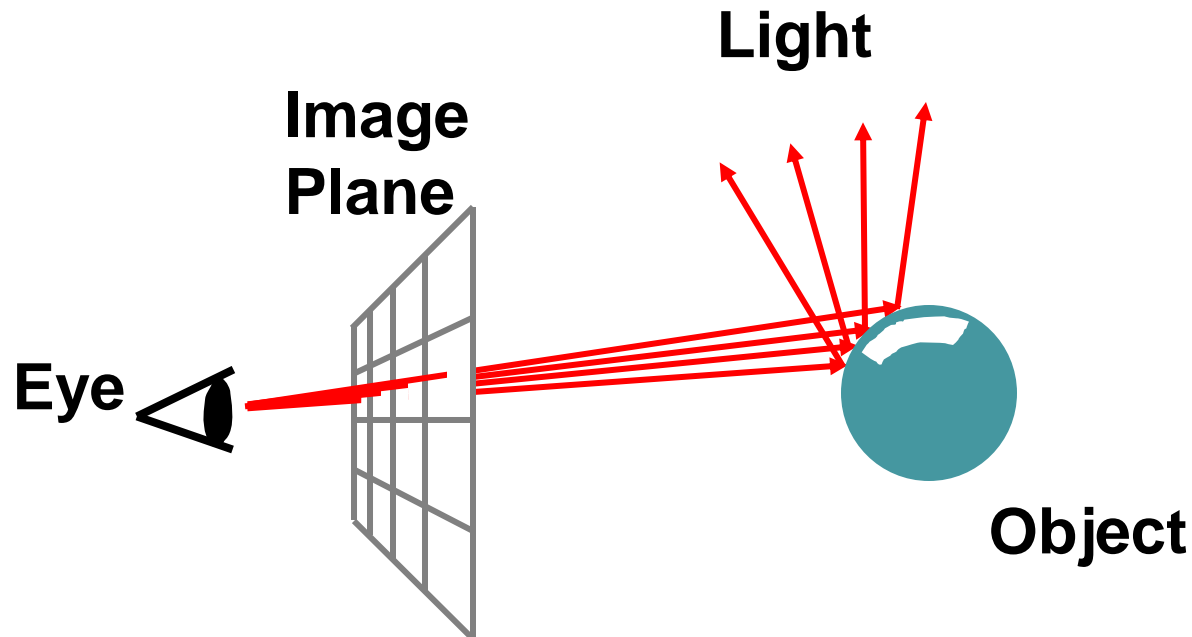
- Trace rays from light
- Lots of work for little return





“Backward” Ray-Tracing

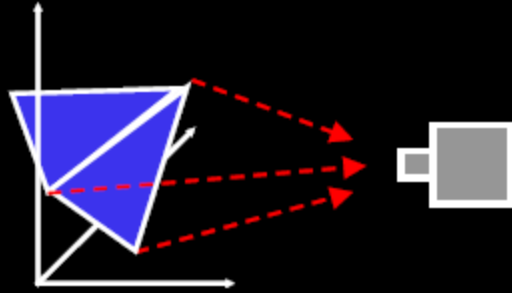
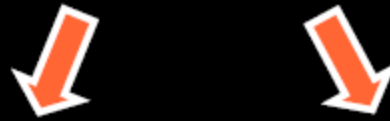
- Trace rays from eye instead
- Do work where it matters



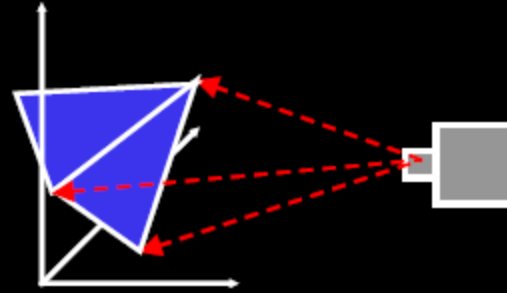
This is what most people mean by "ray tracing".



Rendering in Computer Graphics



Rasterization:
Projection geometry forward

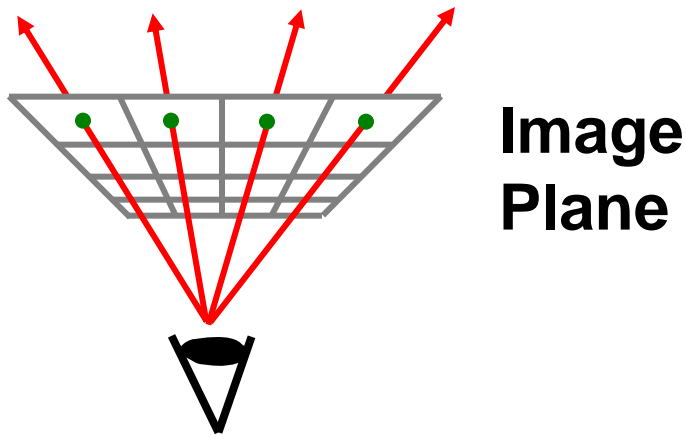


Ray Tracing:
Project image samples backwards



Ray-Tracing Algorithm 101

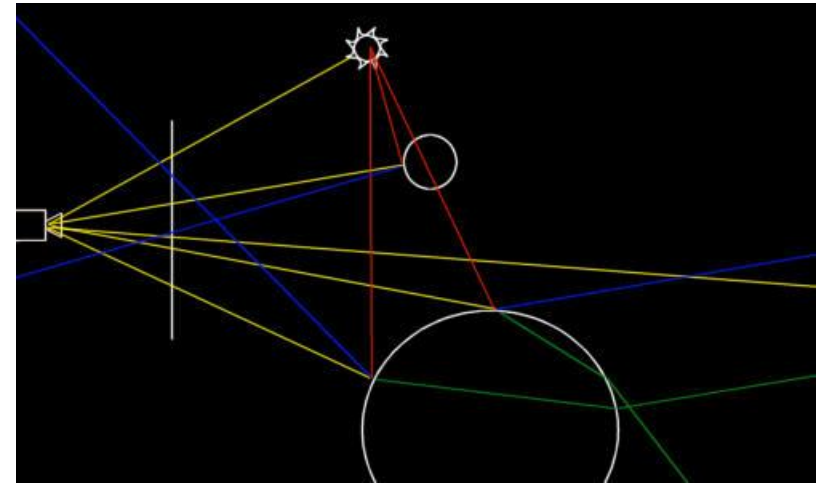
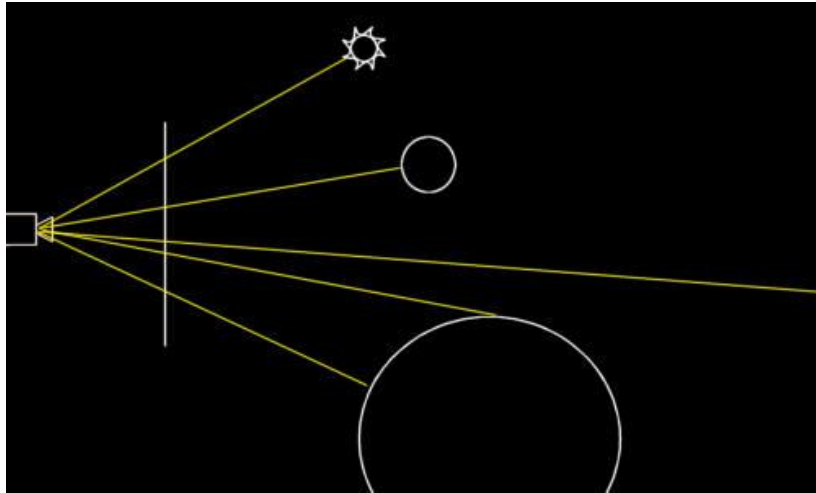
- 1) construct ray from camera through pixel



- 2) Find first primitive hit by ray
- 3) Determine color at intersection point
 - Simple diffuse shading,
- 4) draw color



Primary vs. Secondary Rays



Blue: reflected rays

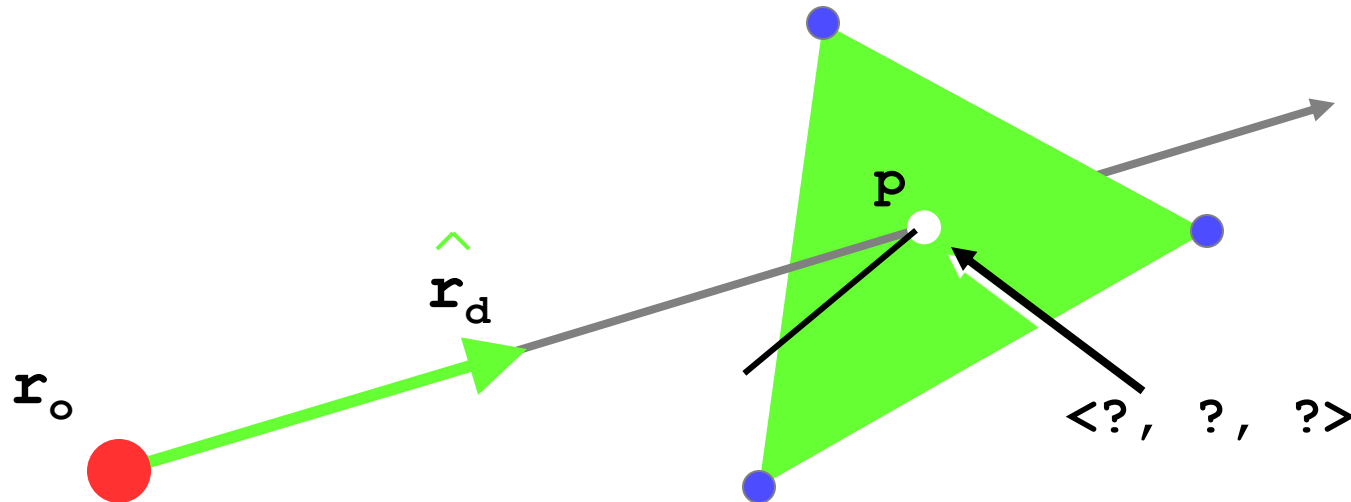
Green: refracted rays

Red: light source (useful for occlusion, shadow)



Triangle Intersection

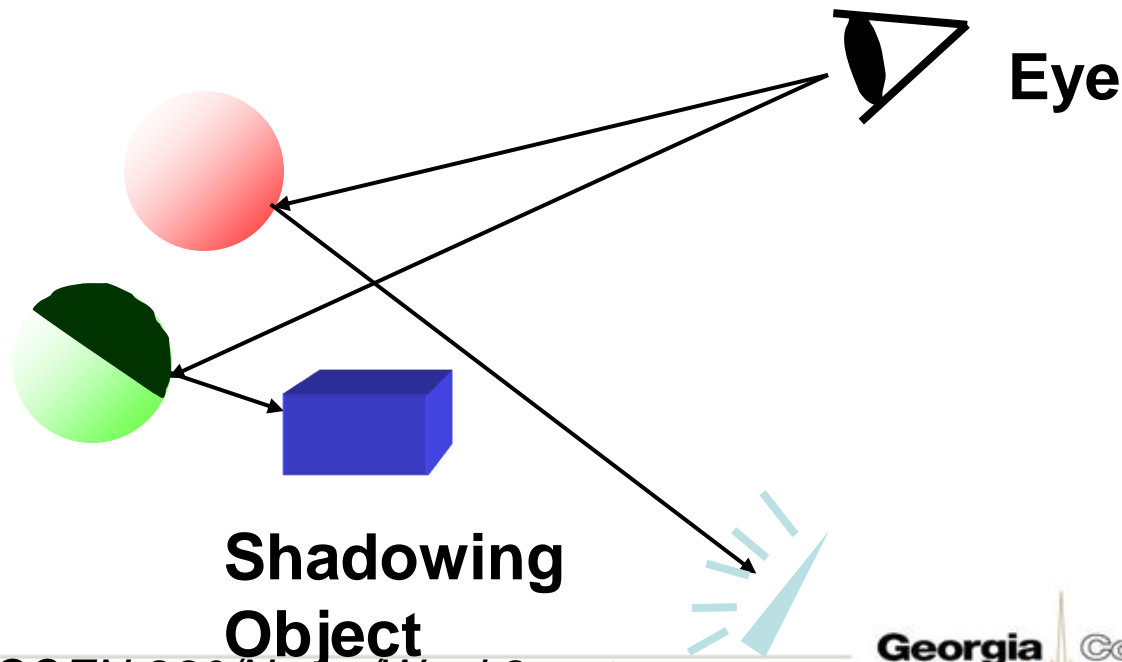
- Want to know: at what *point* (p) does ray intersect triangle?
- Compute lighting, reflected rays, shadowing *from that point*





Shadow Test

- Check against other objects to see if point is shadowed





Lab #3: Improving Ray Tracing

- <http://cg.alexandra.dk/2009/08/10/riers-cuda-ray-tracing-tutorial/>

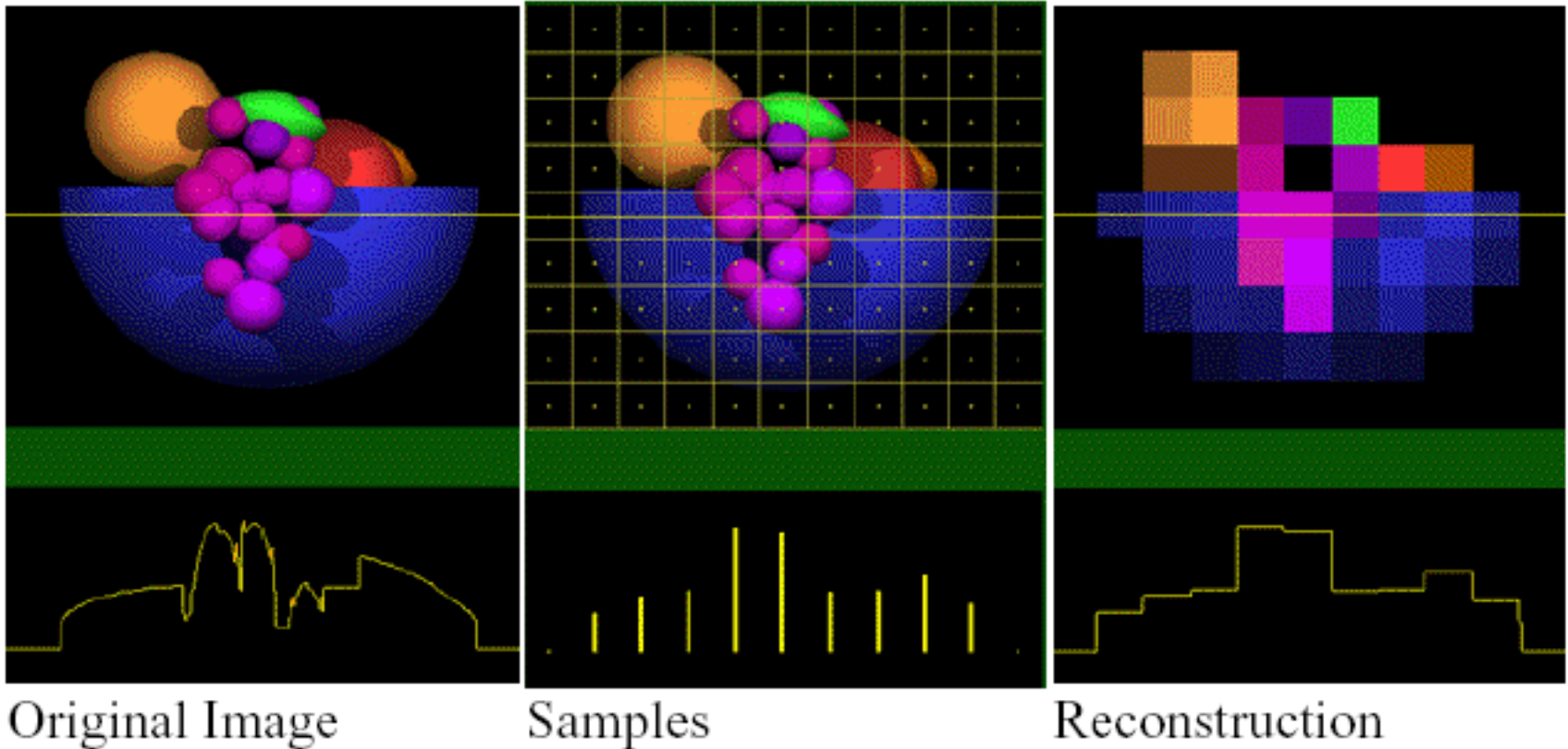


Display with CUDA

- CUDA + OpenGL
- “postProcessGL” SDK example shows a good example
- Simply,
 - Create a PBO (Pixel buffer object) using OpenGL
 - Render the scene to the framebuffer
 - Copy the image to a PBO
 - Map PBO so that it’s accessible from CUDA
 - CUDA process data
 - Copy PBO to a texture
 - Display the texture.



Examples of Aliasing

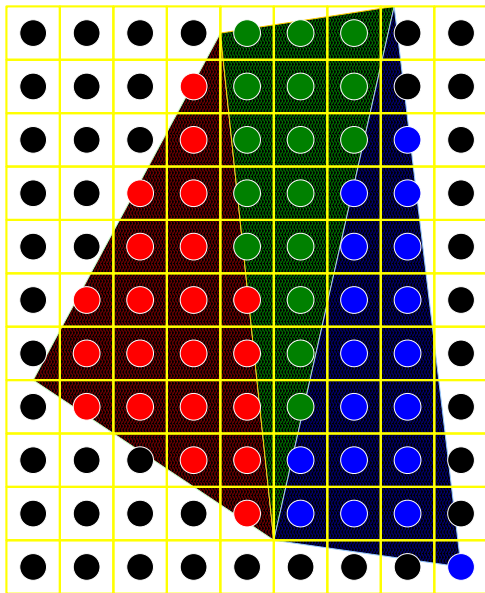


- Aliasing occurs because of *sampling* and *reconstruction*

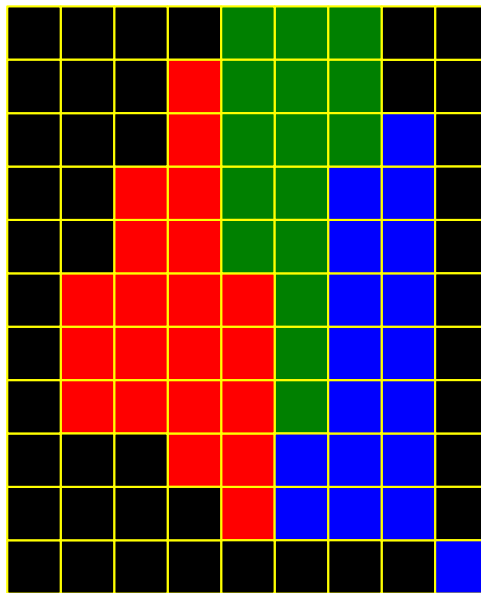


Anti-Aliasing

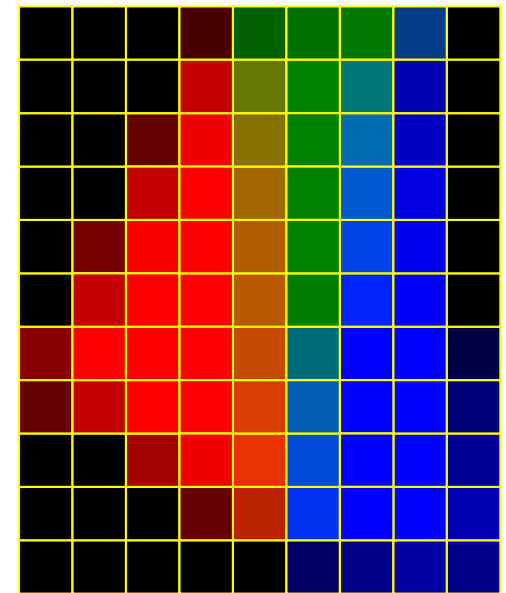
- Aliased rendering: color sample at pixel center is the color of the whole pixel
- Anti-aliasing accounts for the contribution of all the primitives that intersect the pixel



Triangle Geometry



Aliased



Anti-Aliased



Assignments

A group project: 2 people

Provide an example code of ray tracing

(1) implement one anti-aliasing algorithm and compare the results

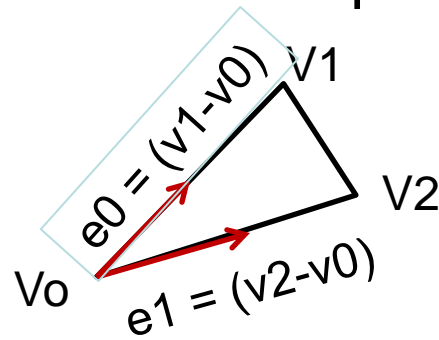
- supersampling

(2) To-be-announced



main.cpp

- Load images into Texture memory
 - Stores edges instead of vertex points to save some calculations



- OpenGL setup
- PBO (Pixel buffer object)
- I/O (setting up camera position, light source etc.)
 - AntTweakBar library provides the APIs



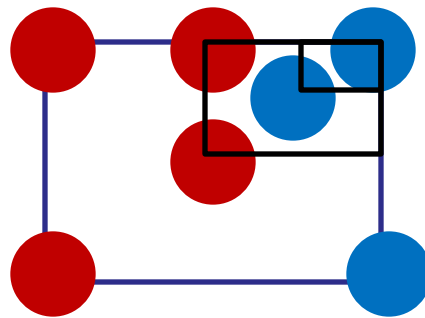
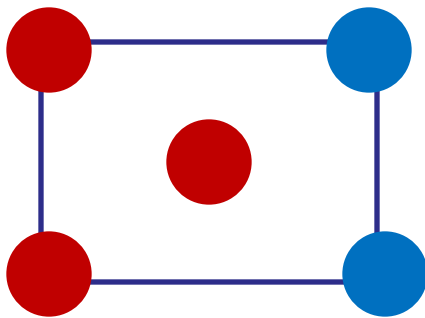
raytracer.cu

- RayTraceImage: interface with C
 - blocksize (8,8,1)
- raytrace
 - Each thread (different image pixel)
 - Calculate ray (origin, direction)
 - Ray sphere intersection, (search the nearest hit point) : traverse all triangle
 - Continue the light (depth level 4) until the light is out of the screen
 - Calculate simple diffuse and specular light
 - Shadow light
 - Create a shadow ray
 - Check whether there is a blocker on the path or not, it is then make it darker
- HitRecord contains color and normal vector and store `out_data` (PBO pointer)

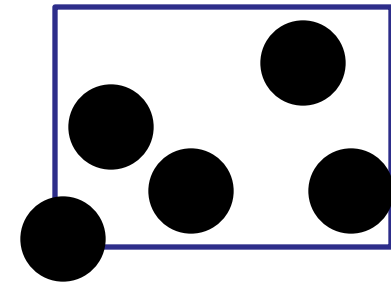


Anti-aliasing with Super Sampling

- More than one array per pixel
- Average the results



adaptive



Stochastic

- Adaptive sampling (increase the resolution based on the color differences)
- Stochastic sampling: Instead of regular grid, random subgrid



Suggesting Anti-aliasing Implementation

- Create more rays
 - How? (naïve super sampling, adaptive, stochastic)
 - Recalculate x_f , y_f
 - Repeat all the computation
 - Average the values before `out_data`



Performance Issues

- Ray tracing pipeline
- Ray generation → Intersection → closest hit shade
- Traverse requires finding closest primitives
- Source of performance degradations
- Optimizations:
 - Better data structures, (KD-tree, BVH (Bounding Volume Hierarchy Data Structure))
 - Monte Carlo simulations
- **Branch...**