Shadows in Ray Tracing

- Shadows are important for realism

- Basic idea: figure out whether a point on an object is illuminated by a light source

- Easy for ray tracers

- Just trace a shadow ray from the point to the light source
  - If the shadow ray hits something before the light, point is in shadow
Shadows in Ray Tracing
Shadows in Ray Tracing

- Problems?
- They’re too dark
- They’re too sharp
- How do we do this with a rasterizer?
Outline

- Shadows in Ray Tracing
- Shadows in Rasterization
- Soft Shadows
- Ambient Occlusion
Shadows in Rasterization

- Basic goal: determine if a fragment is in shadow
- Using some form of rasterization
- Shadows are essentially related to visibility
  - In a rasterizer, visibility information is in the depth buffer
  - So we’ll have to use the depth buffer for shadows, somehow
Shadow Mapping

- When is a point in shadow?

- In ray tracing:
  - When the shadow ray is occluded

- In general:
  - When the light source can’t see the point

- What if we render from the light source?
Shadow Mapping

- Render the scene from the light source

- The depth buffer contains distances to points visible to the light
  - All points behind the visible points are in shadow
  - This is the shadow map

- Now render from the (usual) camera
  - Transform each fragment to the light’s coordinate frame
  - Compare to the shadow map value to check if in shadow
Shadow Mapping

eye depth + eye view = final output

light depth = shadow
Shadow Mapping

- In ray tracing, shadow ray origins were offset
  - To avoid self-intersections due to precision issues

- This can happen in shadow maps too!
  - Precision issues when projecting into light coordinates

- Just render back-faces for the shadow map
  - Depth differences should be large enough
  - Works with closed objects only
Shadow Mapping

- Shadow map is, after all, a sampled image
- If sampled too coarsely, shadows appear blocky
- Can workaround by:
  - Adaptively sampling shadow maps
  - Warping shadow maps to fit camera view
Shadow Mapping
Shadow Mapping

- How often is a shadow map evaluated?
- What shader stage would this happen in?
- Deferred shading can improve performance!
Shadow Volumes

- Construct 3D geometry representing shadow volume

- Compute shadow volumes from light source
  - Shadow volumes are behind objects from light view

- Render from (usual) camera

- A fragment is in shadow if it lies in a shadow volume
Shadow Volumes

- Sharp shadows, unlike shadow maps
- Difficult to create shadow volumes
- Shadow volume containment test is tricky
Shadow Volume Testing

- Render scene from camera, save depth

- Render shadow volume front faces
  - Increment if depth test passes (shadow front face visible)

- Render shadow volume back faces
  - Decrement if depth test passes (shadow back face visible)

- Pixels with non-zero values are in shadow
Shadow Volume Testing
Shadow Volume Testing

- What if camera is in shadow volume?
- Need an alternate approach
Shadow Volume Testing

- Render scene from camera, save depth

- Render shadow volume back faces
  - Increment if depth test fails (shadow back face occluded)

- Render shadow volume front faces
  - Decrement if depth test fails (shadow front face occluded)

- Pixels with non-zero values are in shadow

- Need to cap shadow volumes
Shadow Volume Testing
Outline

- Shadows in Ray Tracing
- Shadows in Rasterization
- Soft Shadows
- Ambient Occlusion
Soft Shadows

- Real shadows don’t have hard boundaries

- This is because real lights are not point lights
  - More generally, because at any point, light comes from multiple directions
Soft Shadows

- Consider area light sources
- Shadow has two parts
- Umbra: points hidden from all of the light
- Penumbra: points that can see part of the light
How to model soft shadows in ray tracing?

At any point, need to know how much of the light source is visible

Trace shadow rays to multiple points on the light source
  - Randomly sampled

Average the results
Soft Shadows
Outline

- Shadows in Ray Tracing
- Shadows in Rasterization
- Soft Shadows
- Ambient Occlusion
Ambient Occlusion

- Shadows cast by *ambient* light
- A crude approximation for global illumination
- Similar approach to soft shadows
Ambient Occlusion

- At each surface point, trace rays out in all directions

- Calculate the fractions of rays that don’t hit anything
  - Until a certain distance, perhaps

- Scale lighting value by this fraction
  - Points surrounded by other objects appear darker
Ambient Occlusion
Ambient Occlusion