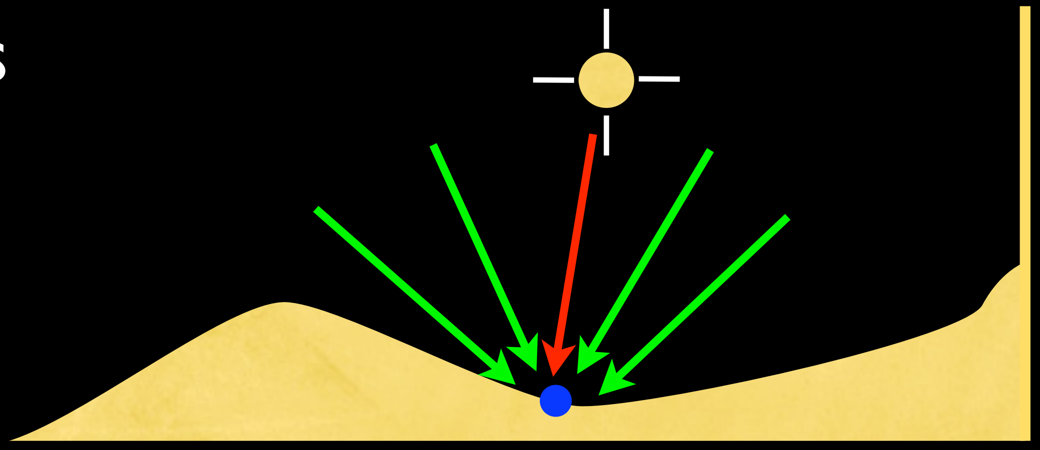
COMPSCI 715 Part 2

Lecture 9 - Global Illumination

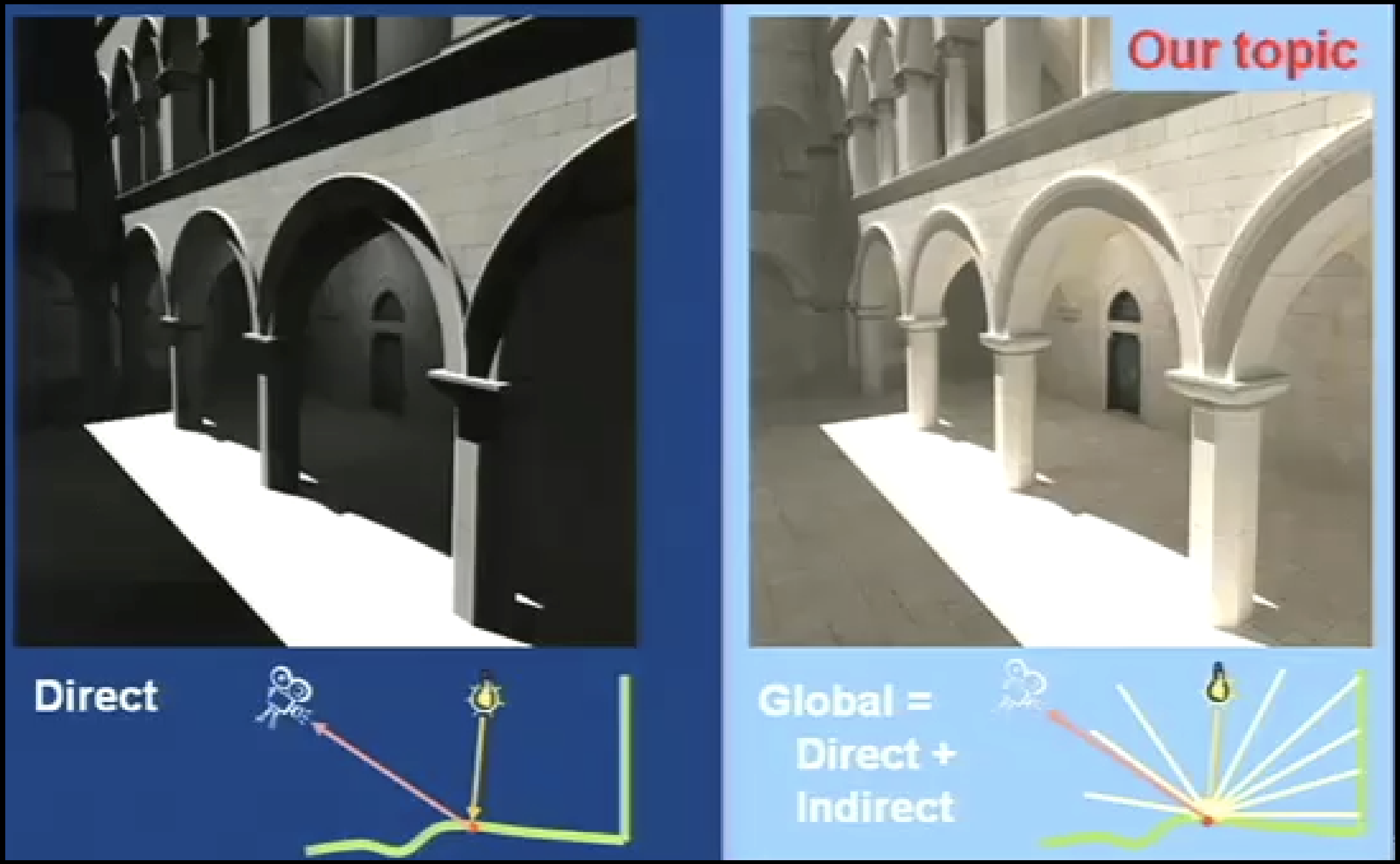
Movie

Global Illumination

- Recursive rendering equation
  - i.e. light illuminates floor, floor illuminates ceiling, ceiling illuminates floor
- Attempt to solve inter-reflection
- Radiosity algorithms: assume all surfaces are diffuse reflectors



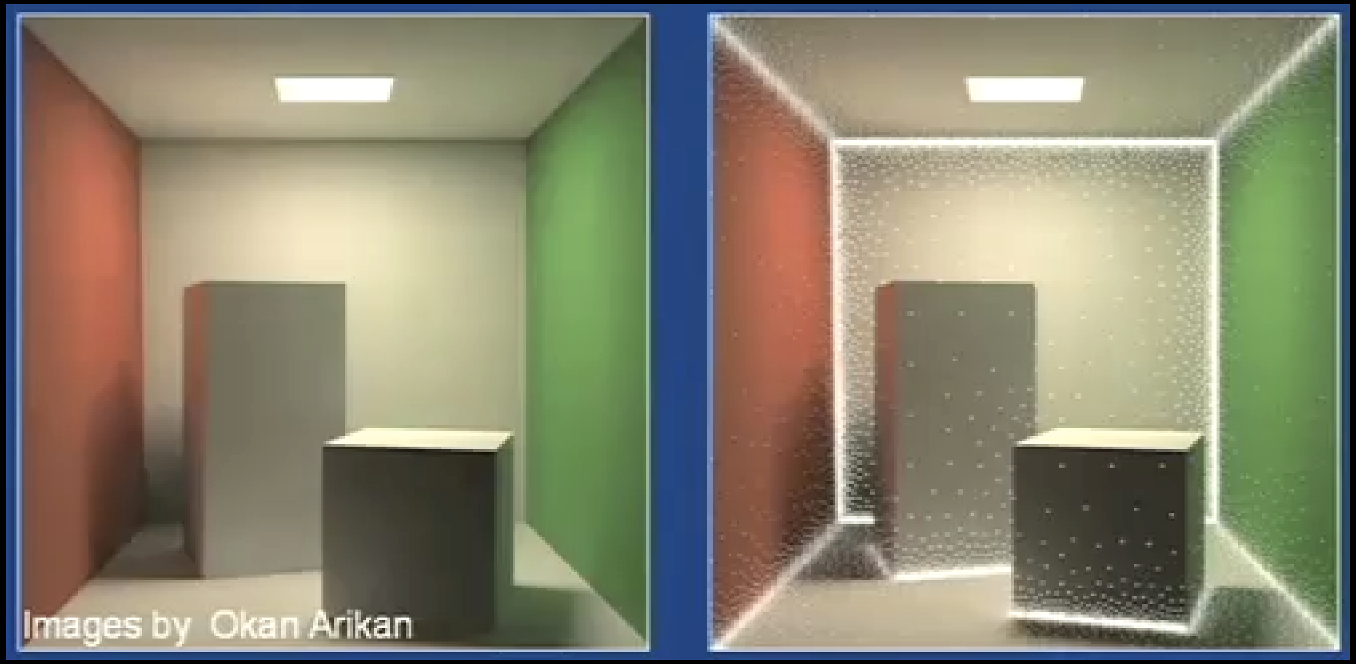
Global Illumination



Source: Krivanek et al

Diffuse Interreflections with Irradiance Caching

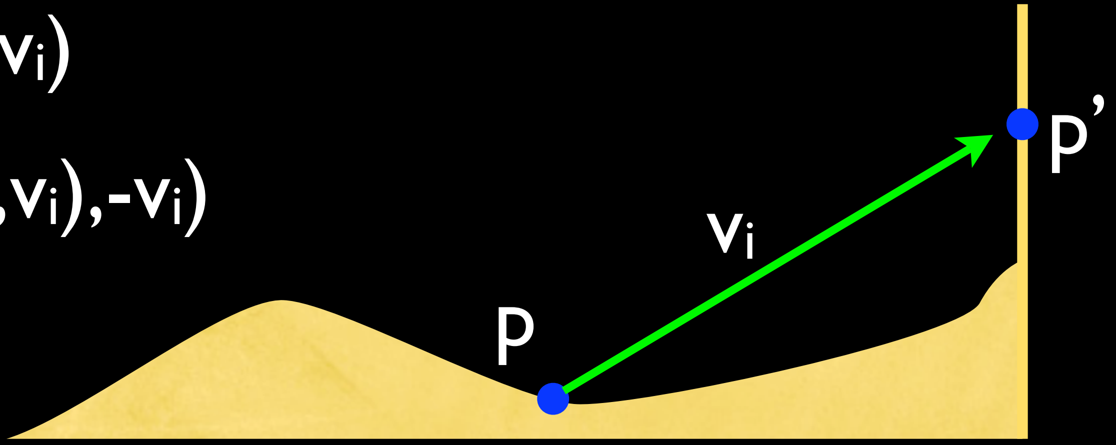
- Technique from 1988 still in wide use today
- Stochastically sample scene through ray tracing and interpolate



Diffuse Interreflections with Irradiance Caching

- At each point:
  - Sum integrals of light from each direction
- From each direction:
  - $L_i(p,v_i) = L_0(p',-v_i)$

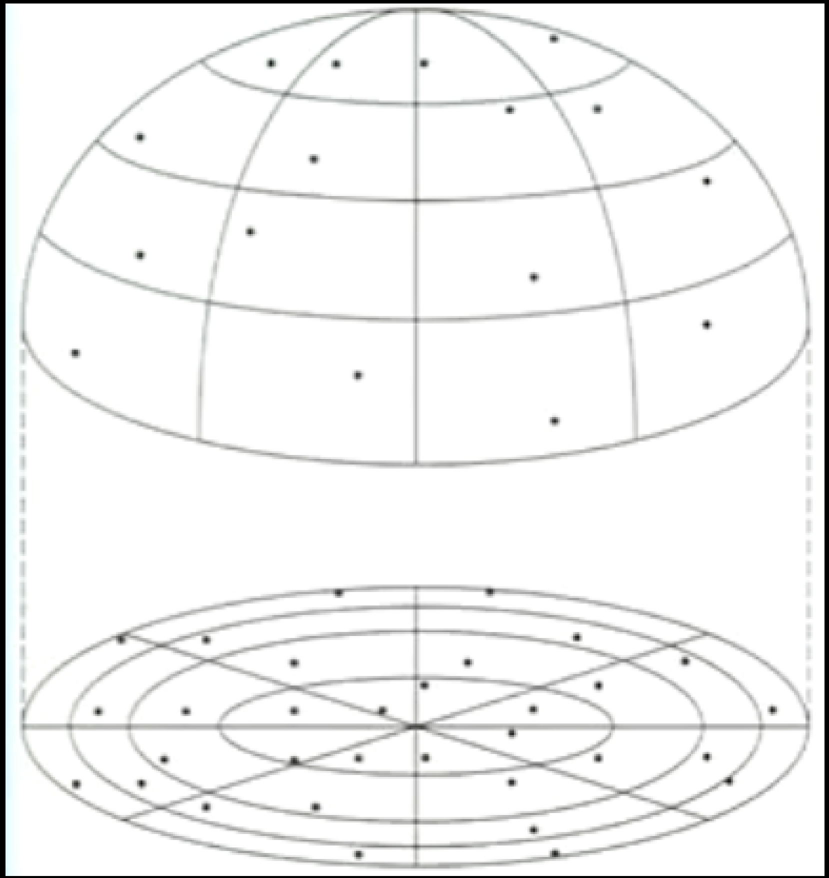
    $= L_0(r(p,v_i),-v_i)$



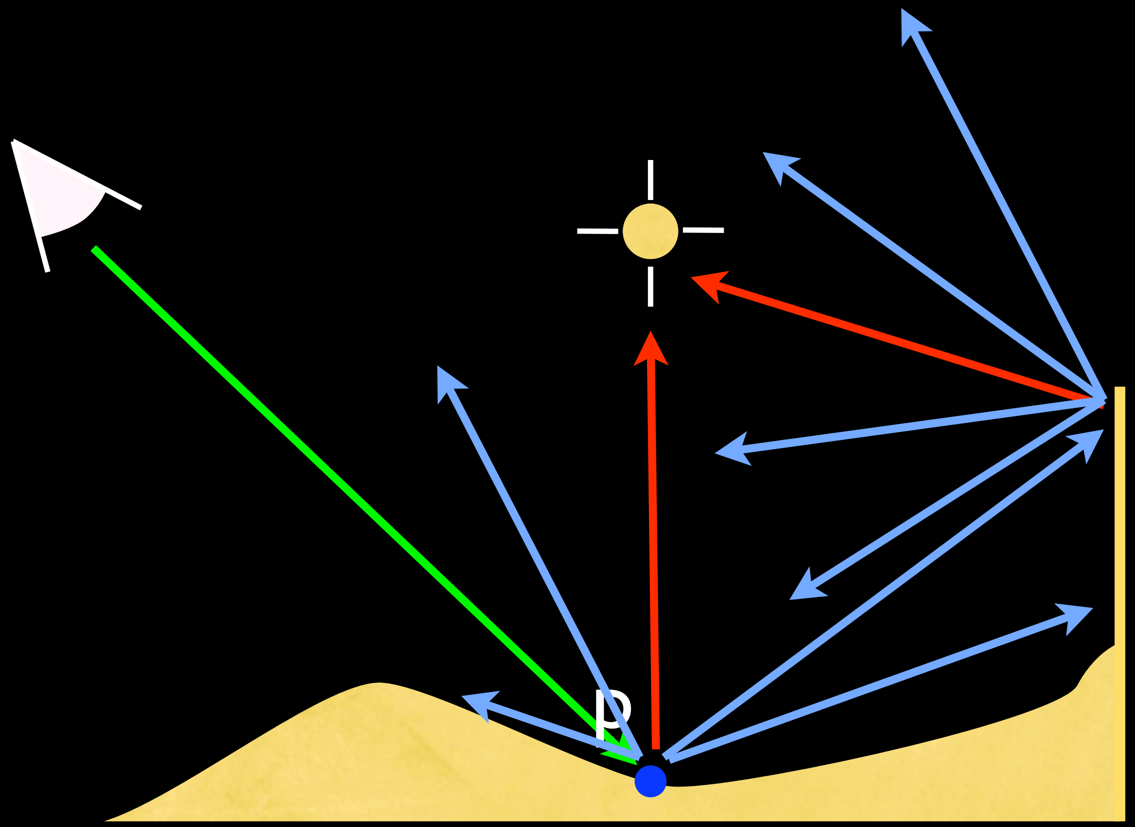
Intensity

- Total light at each point

$$L_0(p,v_0)=L_e(p,v_0)+\int (L_i(p,v_i)f_r(v_i,p,v_0)\cos\theta_i)\ dv_i$$



Recursive Ray Tracing



Recursive Ray Tracing

- Many secondary rays to accumulate global illumination (each of which has many seconday rays...)
- Breaking the recursion:
  - Path Tracing - single GI secondary ray
  - Final Gathering - read from a rough solution (i.e. Photon Mapping)
  - Irradiance Caching - cache lookup

Photon Mapping

- Two pass solution:
  1. Photon tracing
     - Rough global illumination solution
  2. Ray tracing
     - Final image rendering

Photon Mapping: Pass 1

- Rough GI solution
- Process:
    - Emit photons from the light source and store hit point in map
    - Each photon that hits is reflected randomly

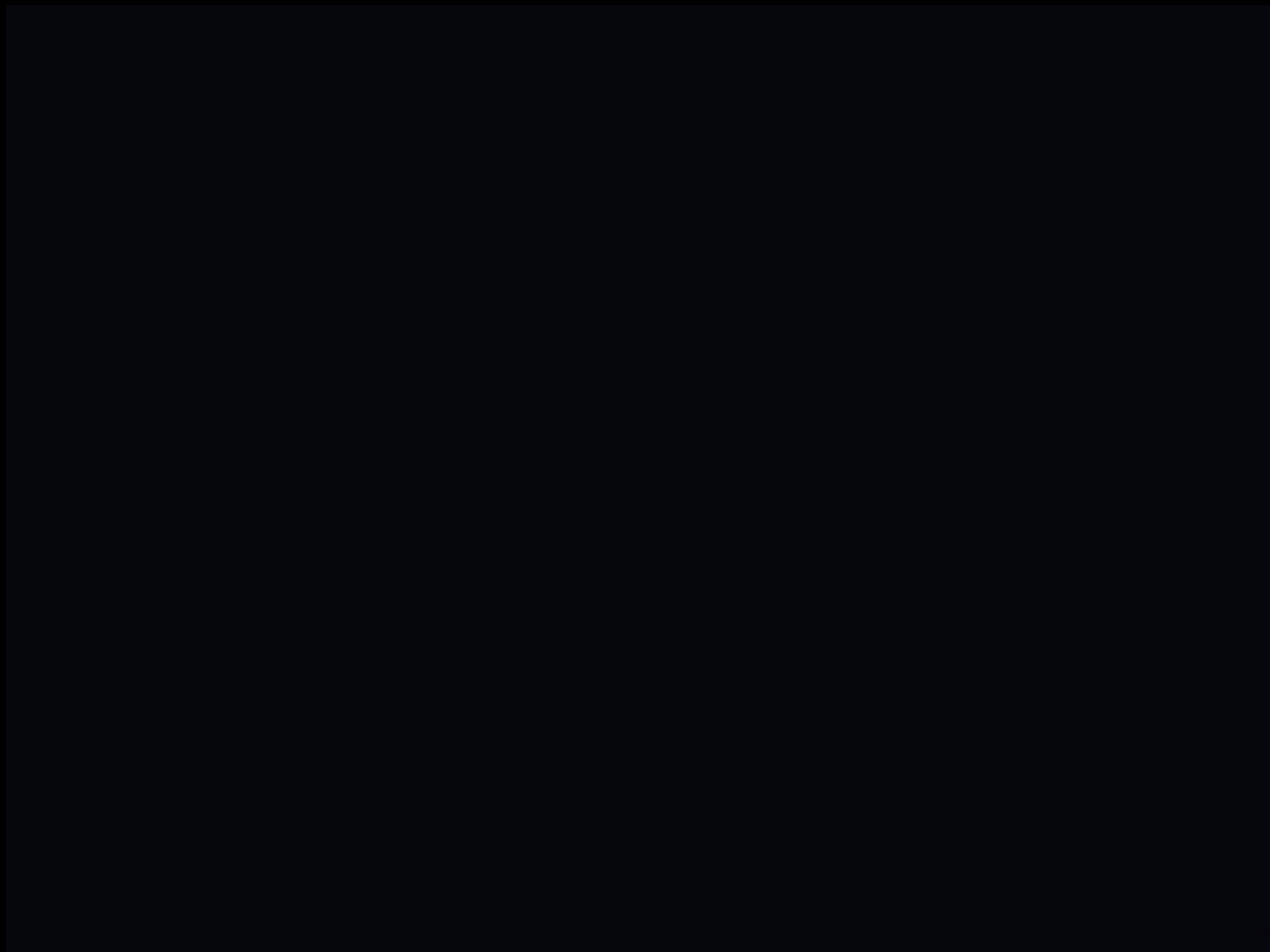
Photon Mapping: Pass 2

- Image Rendering
- Process
  - Use normal ray tracing
  - To obtain intensity do one step of distributed ray tracing (500-5000)
  - Lookup values for each of these from photon map

Photon Mapping - Problems

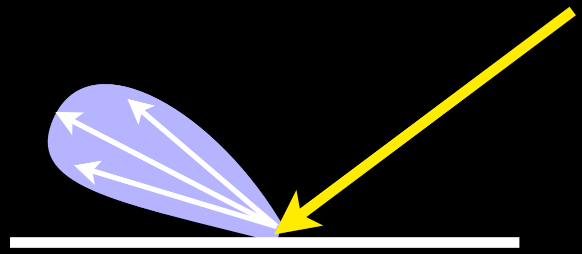
- Slow
- Patchy

Mies House

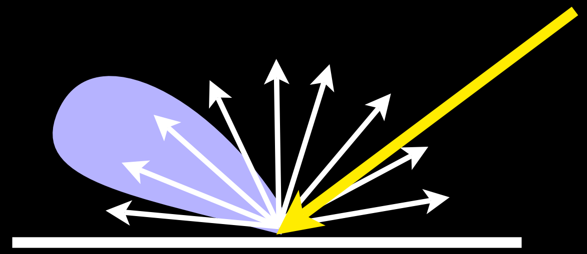


Photon Mapping Importance Sampling

- Depending on BRDF shape can sample in directions most likely to help
- Works well for specular component



Importance Sampling



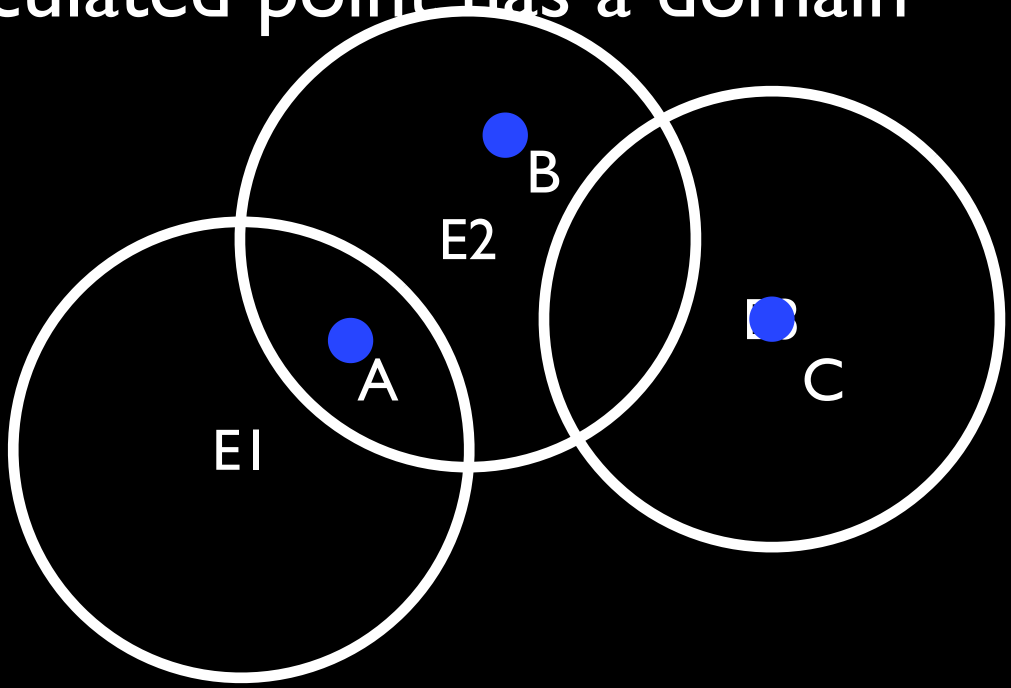
Uniform Sampling

Photon Mapping Interpolation

- Diffuse component
  - View independent
  - Changes slowly over the surface
- Can cache results and interpolate between sample points

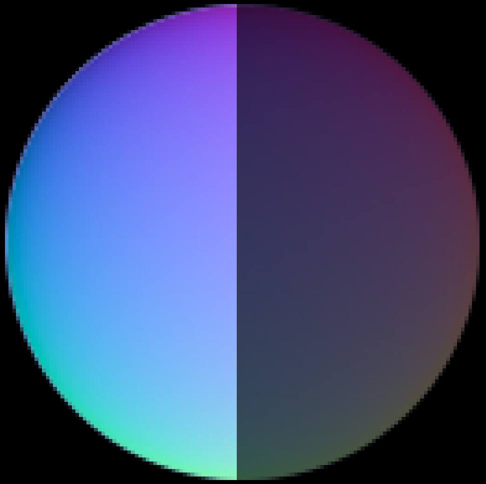
Irradiance Caching

- Uses the properties from the previous slide
- Only evaluates points as necessary
- Each calculated point has a domain



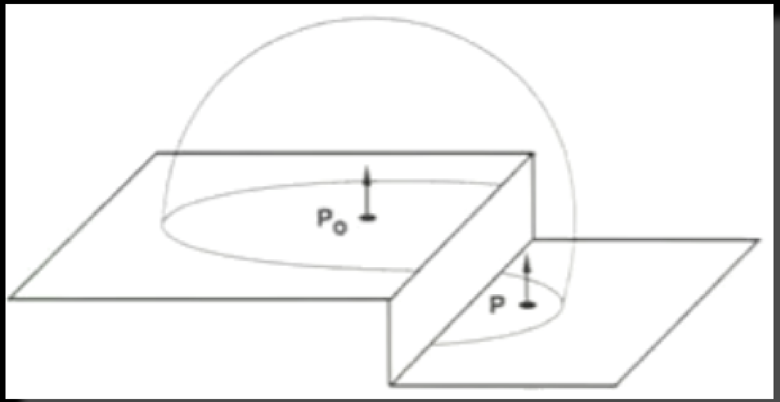
Point Selection

- Split-Sphere Approximation
  - Worst case scenario:



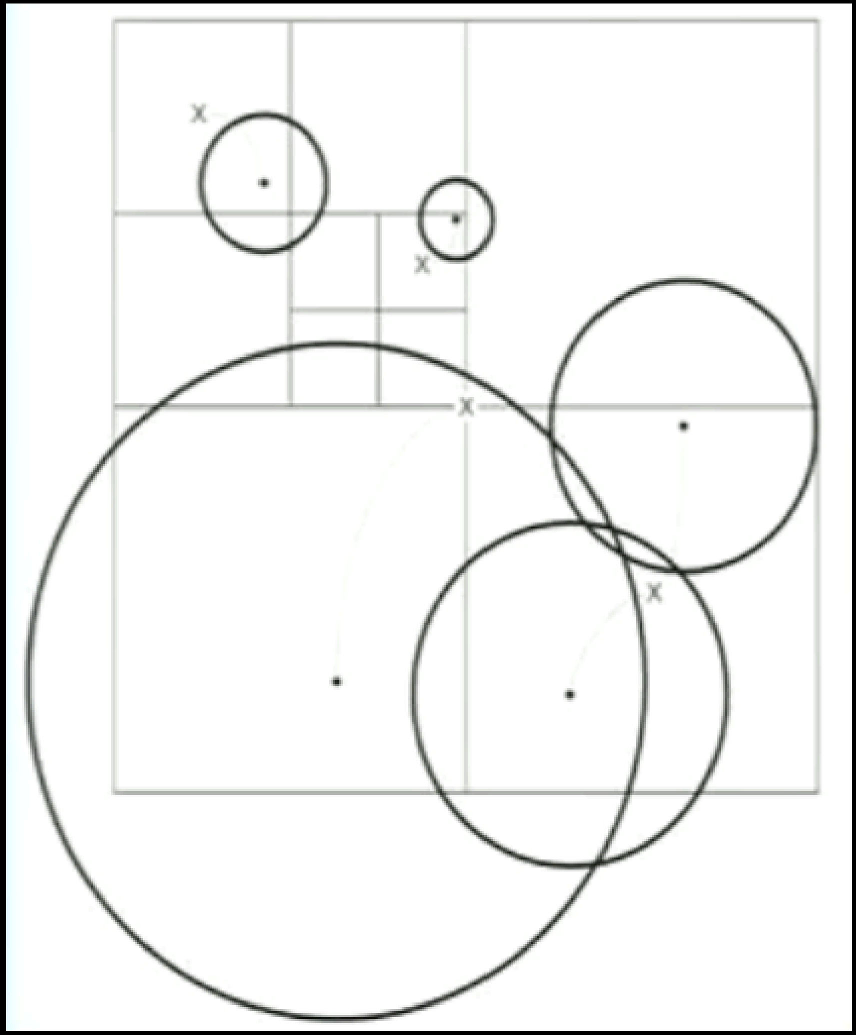
- 'Worse than sphere' constraint

$$(P\text{-}P_i) \cdot \frac{(N_P + N_{P_i})}{2} \geq 0$$

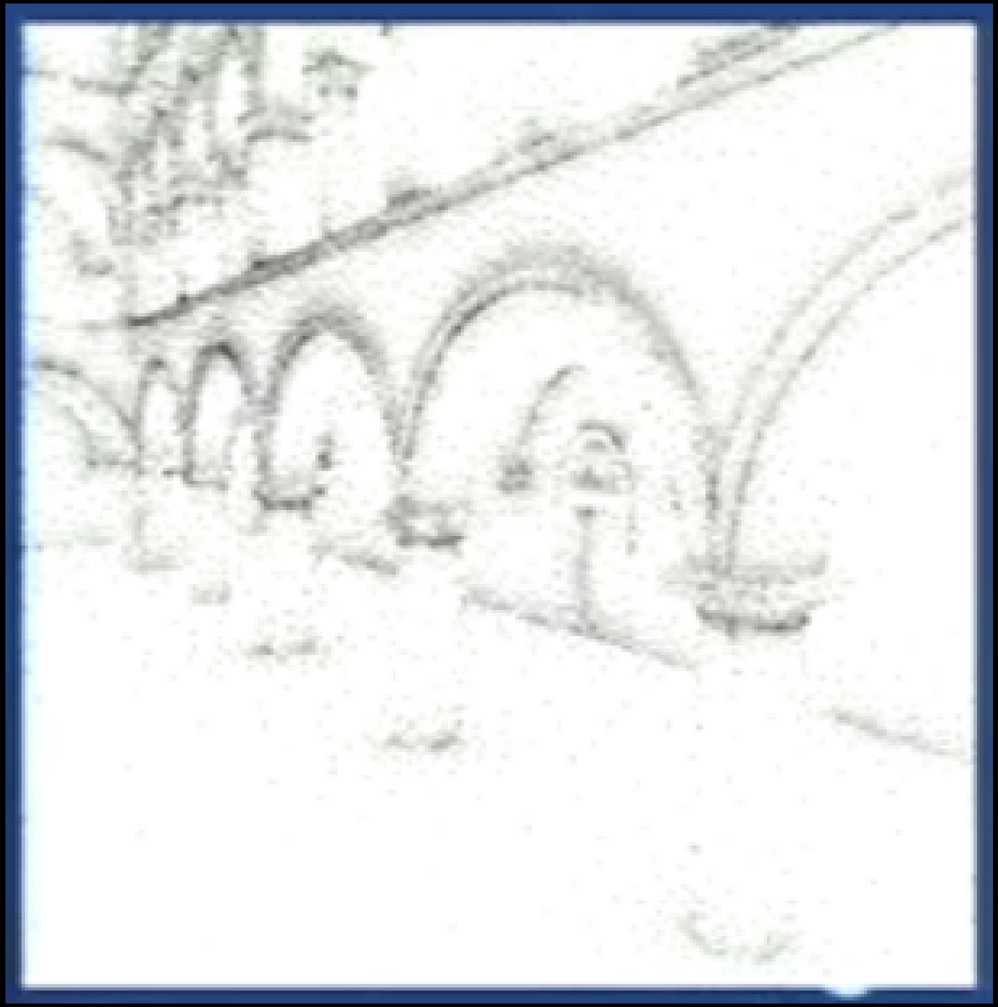


How to store domains

- Octree
  - Each node is ~ same size as sample point's domain
  - On lookup: recurse until all nodes that are influenced by a point are found



Sample Densities



Irradiance Caching Limitations

- Cached values over different scales
  - May cause light leaks
- ‘Hairy’ geometry
  - Grass, trees etc
  - Cost too high to warrant caching
- Bias in super sampling

Extension for glossy surfaces

- Is a view dependent phenomena
  - When reusing the cache we need to use different parts of distribution
  - Can store and reuse like before, using gradients to interpolate
- Only works for low frequency BRDFs

Irradiance Cache on the GPU

CPU

- Raytracing
- Tree Storage
- Spatial Queries

GPGPU →

Reformulation →

GPU

- Rasterization
- Texture Storage
- Texture lookups

Reformulation

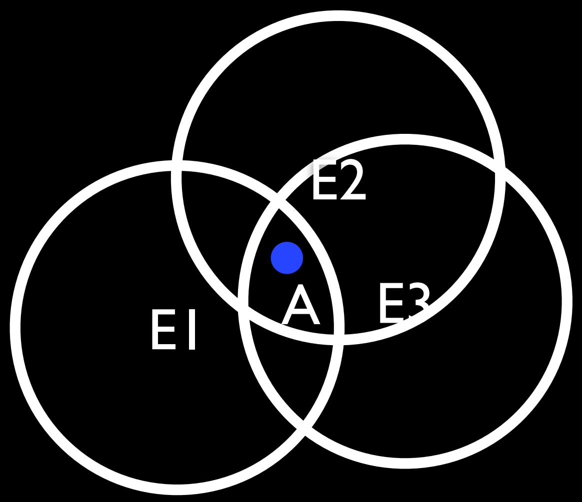
- To use OpenGL
- Make use of native features and optimizations
- Replace data structures with simple alternatives

Reformulation

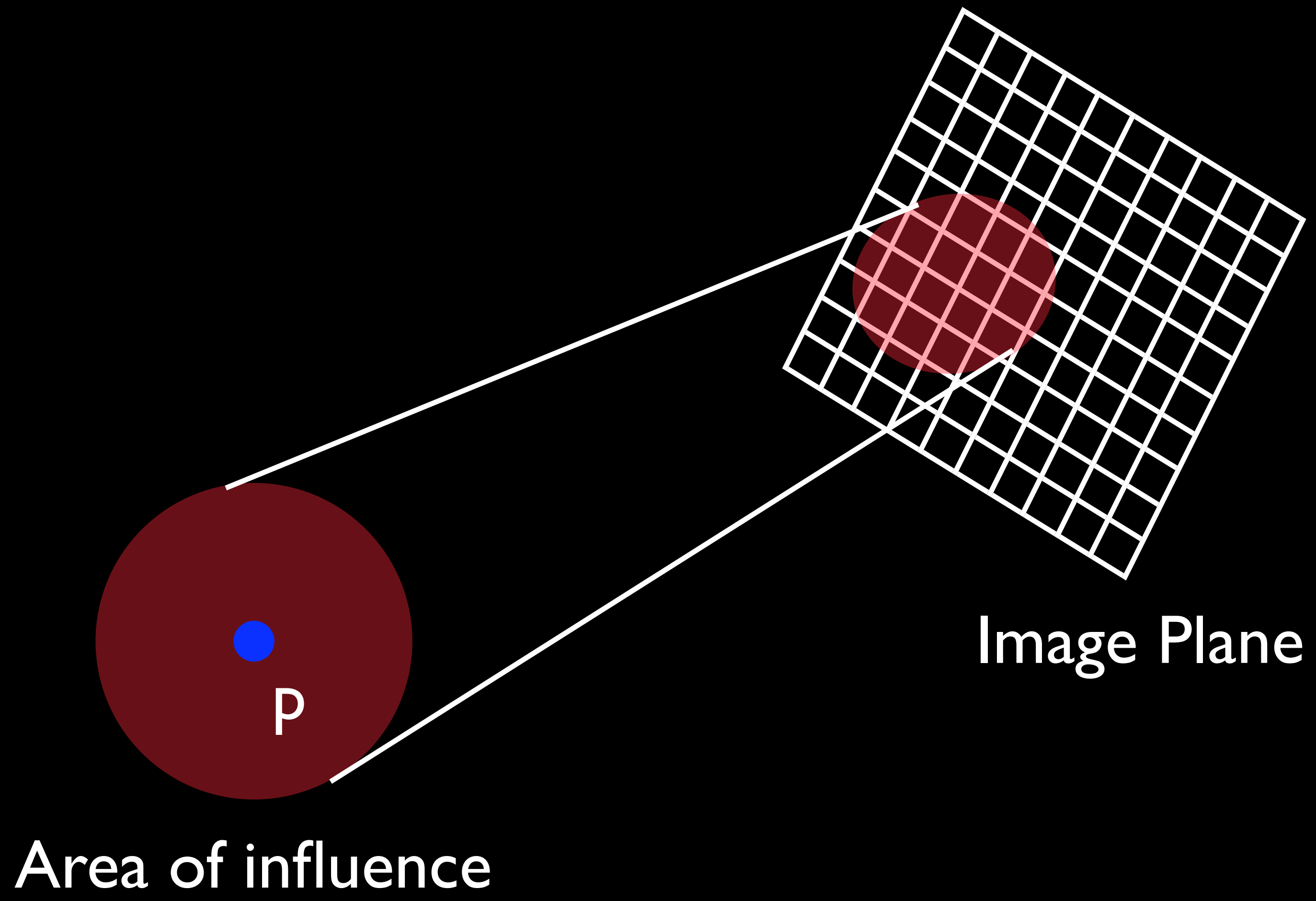
- Octrees => Splatting
- Raytracing => Rasterization

Reformulation - Splatting

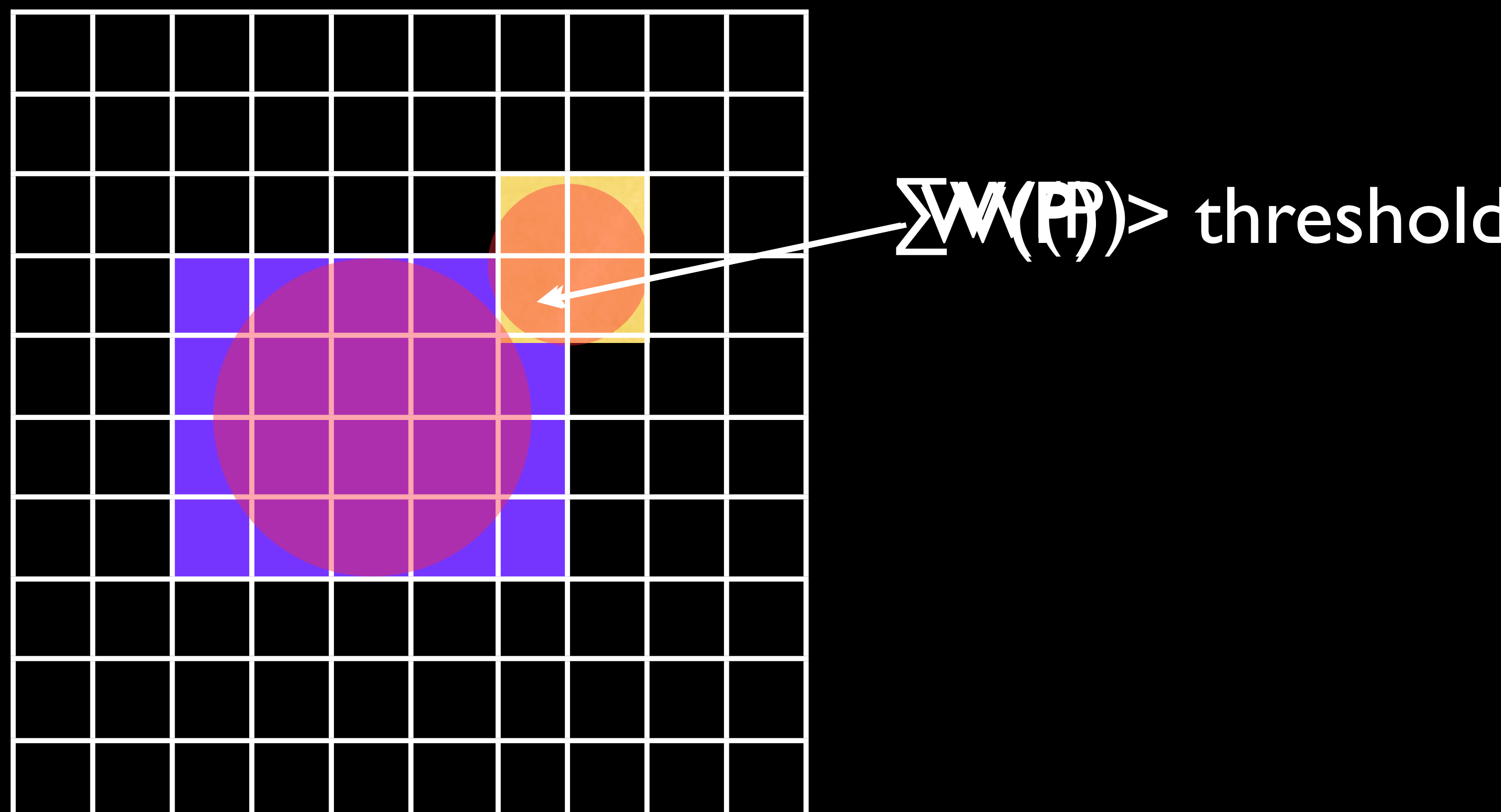
- Octree interpolation:
  - Weighted average of all applicable points
- Splatting:
  - Each splat adds its contribution for what will be each pixel (no lookup required)



Reformulation - Splatting

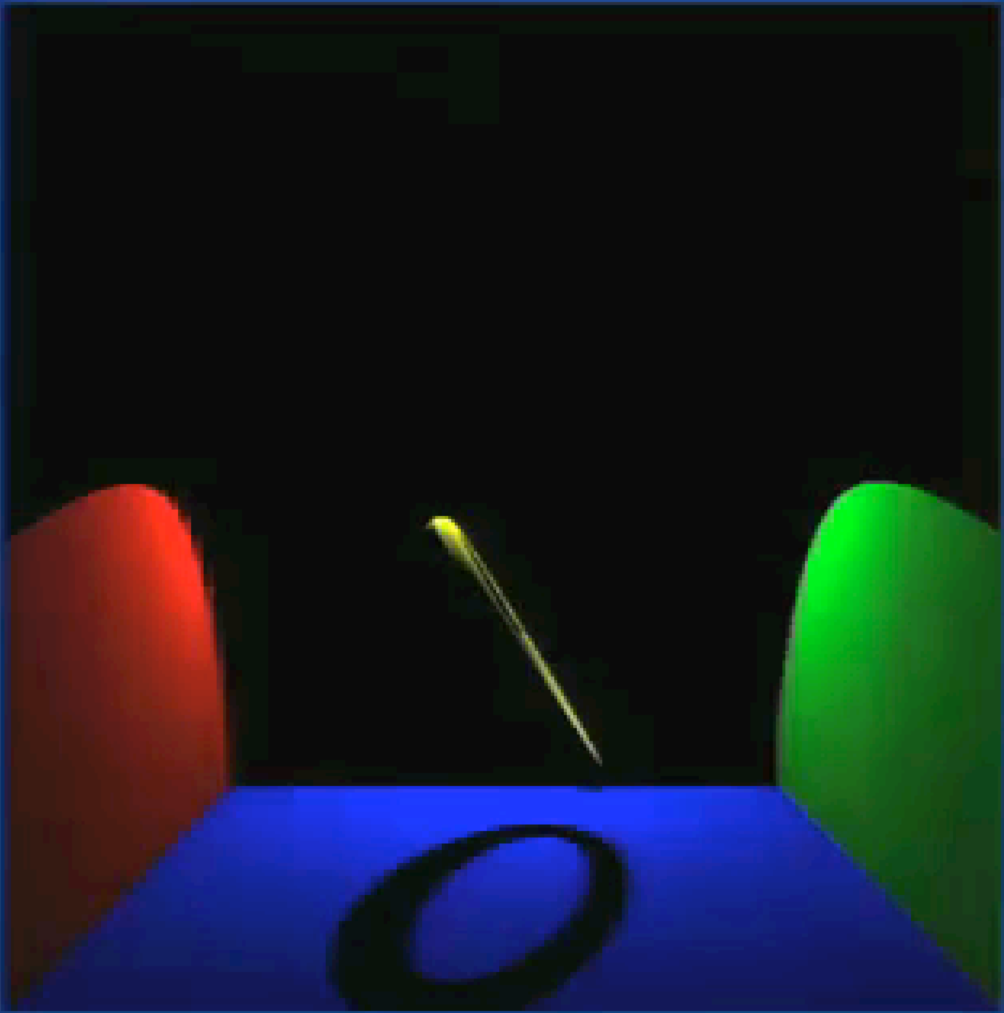


Reformulation - Splatting

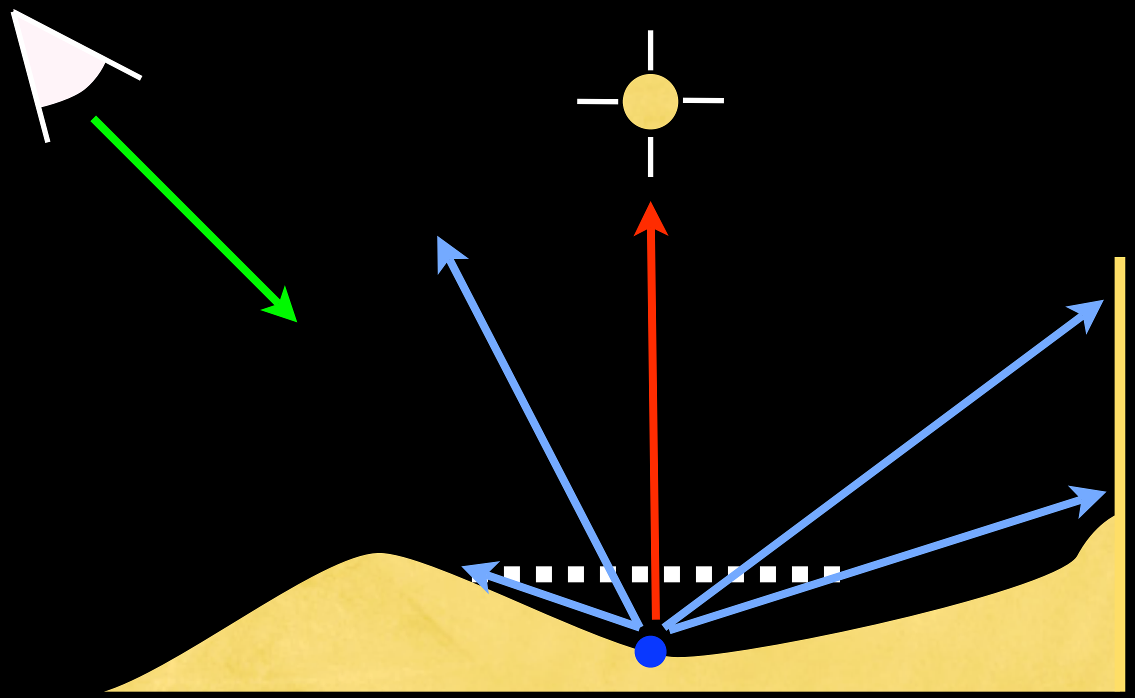


From Octree to Splatting

Example



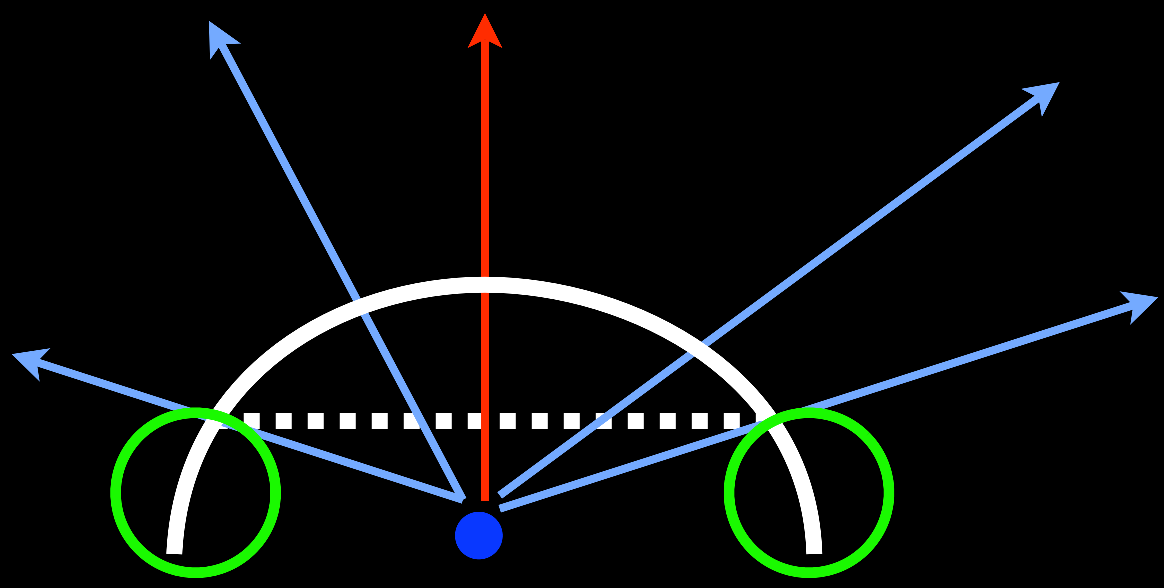
Reformulation - Rasterization



- Use a rendering pass to sample incoming radiance

Reformulation - Rasterization

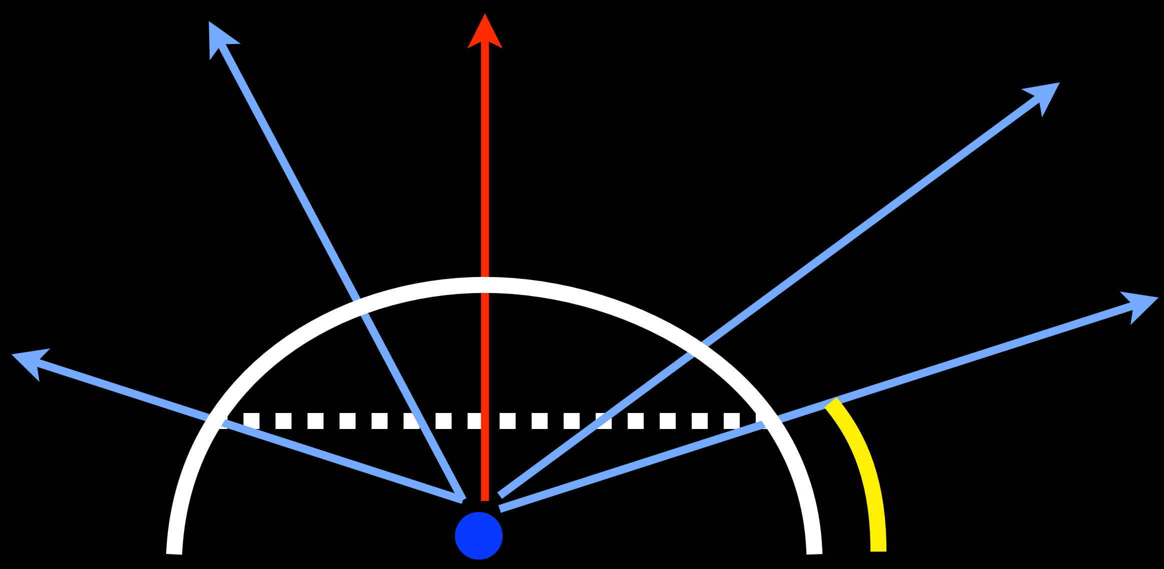
- Problem:
  - Lack information for behind the viewing plane



Areas of lost Irradiance

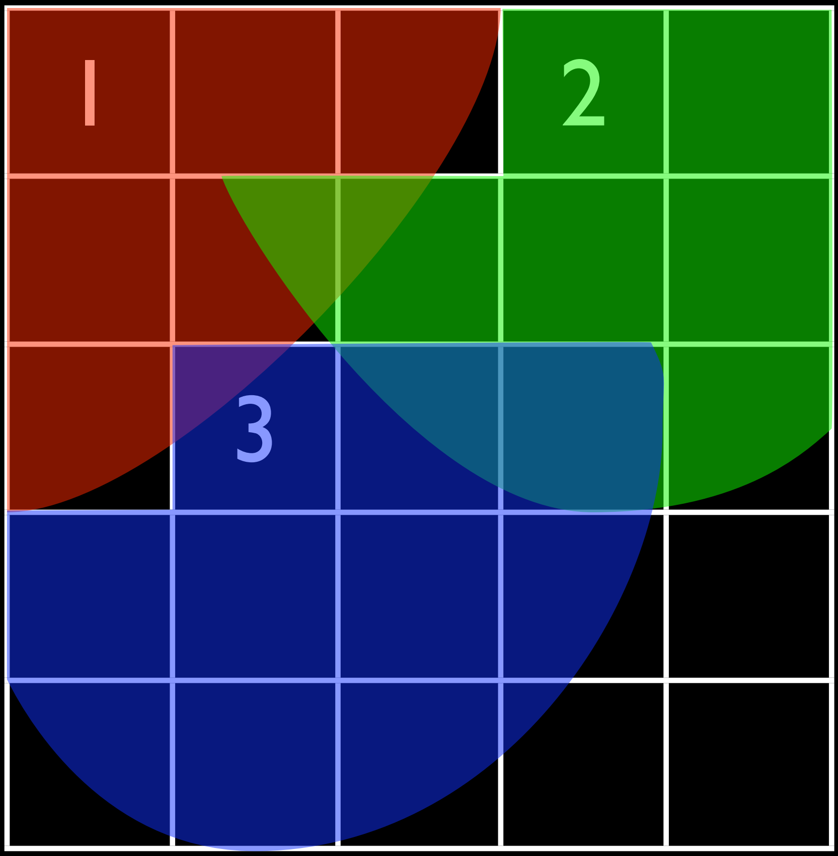
Reformulation - Rasterization

- Solution:
  - Assume colour at edge continues



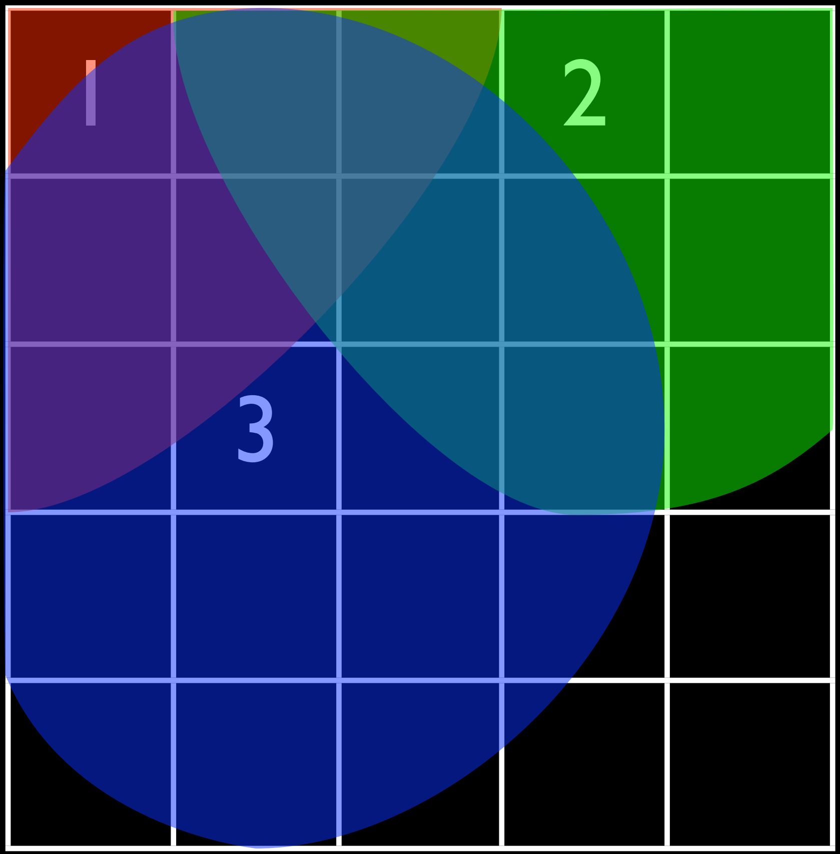
Octree based artifacts

- Scanline algorithm:
  - Can’t go back and recompute using new weightings without using 2 passes



Splatting based solution

- Splatted contributions mean contributions added immediately in same pass



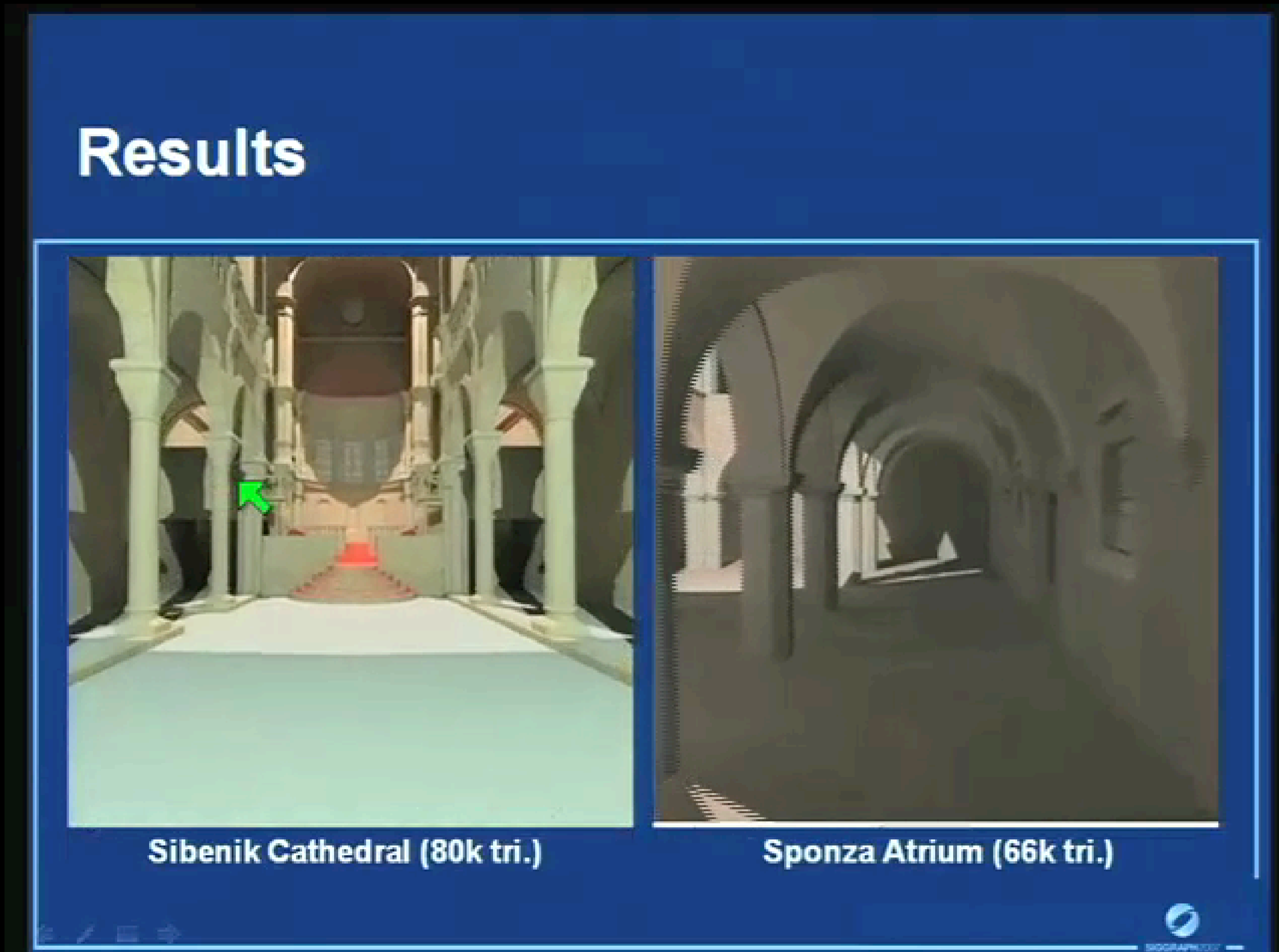
Reformulated Algorithm

- For each sample point:
  - Create bounding box of influence (VS)
  - Compute weighted contribution per pixel (FS)
  - Final rendering pass to obtain final image (FS)

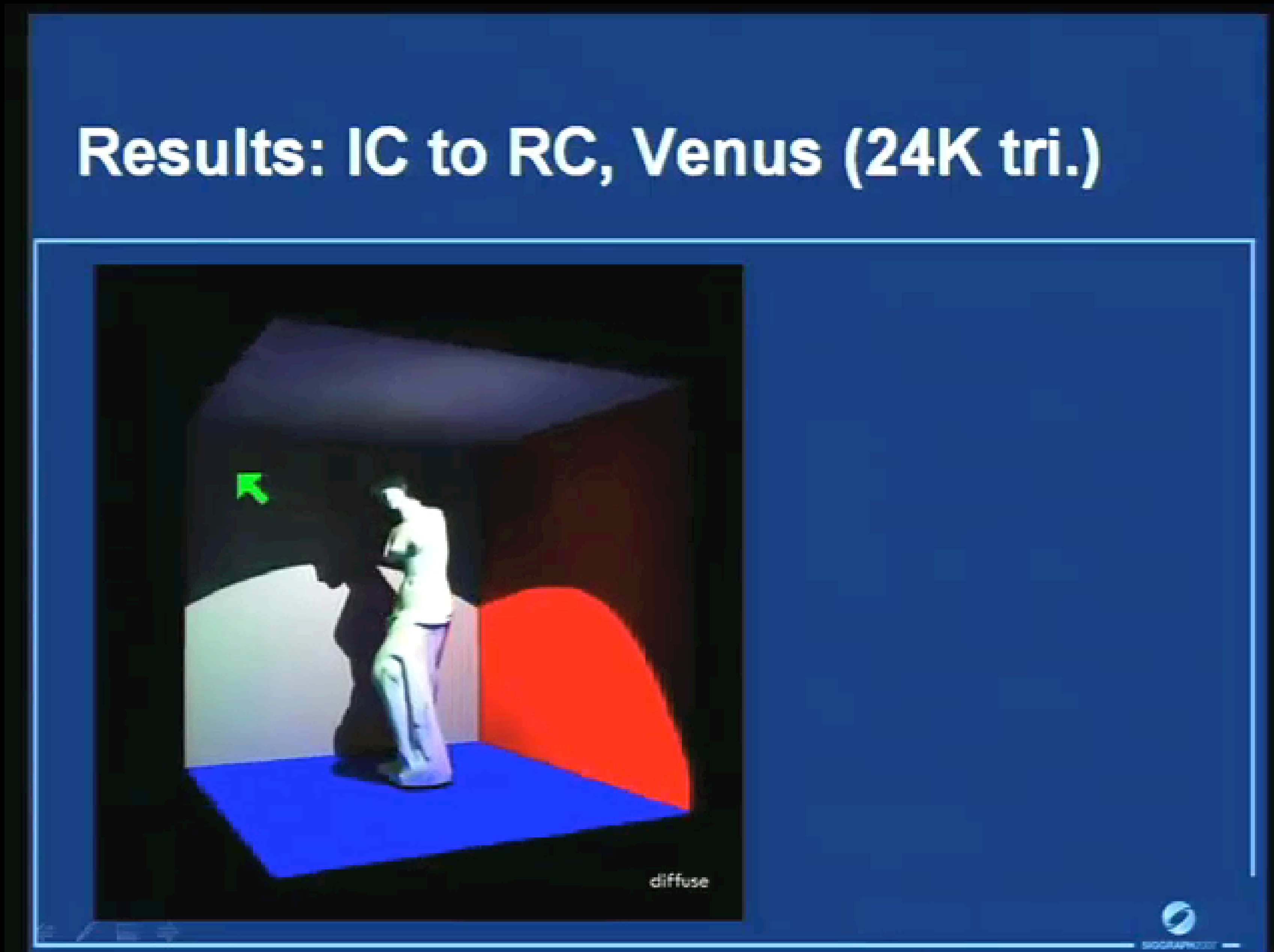
Reformulation - Summary

- No spatial based data structures
- Spatial queries replaced by splatting
- Interpolation *exactly* the same (no quality loss)
- Implementation uses standard GPU features
- ~30-50 times speed increase

Reformulation - Results



Reformulation - RC



Sources

- Krivanek, J et al (2007). Practical Global Illumination with Irradiance Caching. SIGGRAPH 2007 Course Notes