Raytracing II

12 April 2007
CMPT370
Dr. Sean Ho
Trinity Western University

Review last time

- Radiosity: terminology
  - Assumptions
  - Solid angles
  - Radiometry
  - BRDFs
  - Albedo
  - Radiosity equation
    - Form factors

Radiosity equation

- Interaction between a surface element i and all other surface elements j:

$$A_i B_i = A_i E_i + \rho_i \sum_j F_{ji} A_j B_j$$

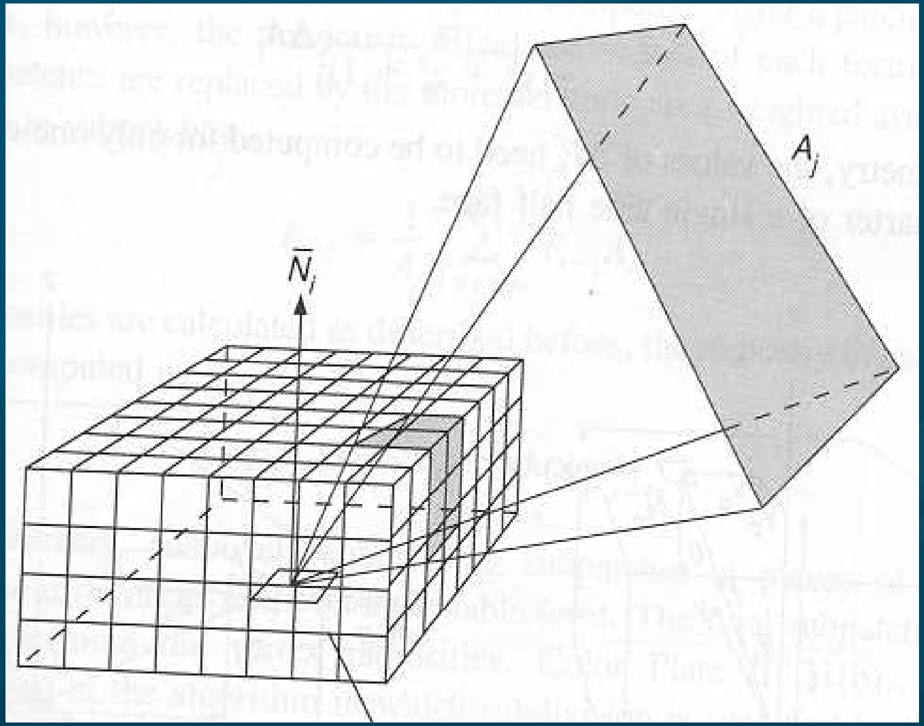
  - A_i: area of element i (known)
  - B_i: radiosity of element i (unknown)
  - E_i: emissivity of element i (given)
  - ρ_i: reflectance of element i (given)
  - F_{ji}: form factor from j to i (computable)

Radiosity pipeline

- Split geometry into elements
- Compute form factors F_{ji} between all elements
- Solve radiosity equation to calculate radiosity of each element
  - Very large linear system
  - Radiosity view-independent
- Render final view with OpenGL
  - Flat-shaded using colours given by radiosity

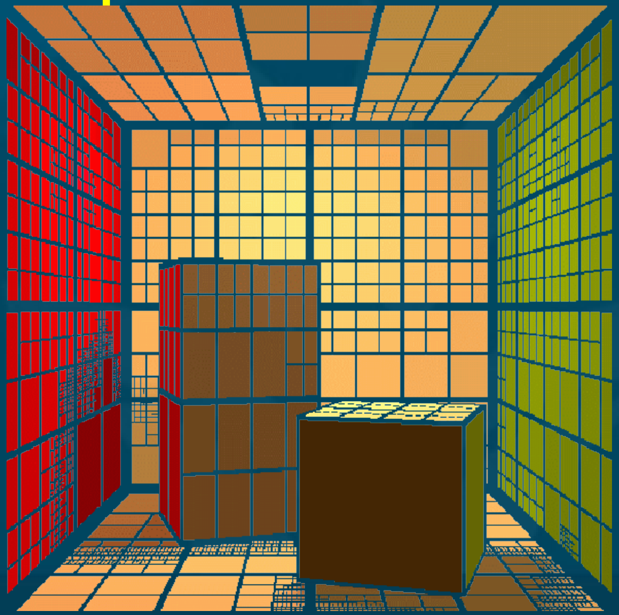
Computing form factors F_{ji}

- Finding visiblity of elements j from element i
- Hemicube algorithm:
  - Render scene onto hemicube about i
  - Instead of colour, use patch id# j
  - Exploits hardware z-buffering to resolve occlusion



Substructuring

- Make distinction between patches and elements:
  - Subdivide patches into elements when needed
    - Near lights, shadow boundaries
  - Skip interaction of elements in same patch
  - Compute F_{ji} from elements to other patches
  - Other patches find a summary F_{ji} to this entire patch
  - Complexity: $O(m\ n)$, where
    - m: # patches
    - n: total # elements



Progressive solution to radiosity

- Iterative solution to global radiosity equation:

$$A_i B_i = A_i E_i + \rho_i \sum_j F_{ji} A_j B_j$$

- Initialize with $B_i = E_i$ for all emitting elements, $B_i = 0$ for all other elements
  - Start with brightest element
  - For all other elements, approx $B_j = \rho_j B_i F_{ji} (A_i / A_j)$
    - Shoot light from this source to scene
  - Do next brightest elt; iterate to convergence
    - Optimization: only iterate over emitting elements

Comparing radiosity + ray tracing

- Kinds of light transport
  - Diffuse to diffuse (done by radiosity)
  - Specular to specular (done by ray tracing)
  - Specular to diffuse
  - Diffuse to specular
- Ray tracing: view-dependent
  - Specular to specular
  - Light sources to diffuse
- Radiosity: view-independent
  - Diffuse to diffuse

Specular radiosity

- Theoretical extension to include specularity
- Diffuse radiosity
  - Lambertian surfaces: light scatters isotropically
  - Only need to know F_{ji} relationship btwn patches
- Specular radiosity
  - Full BRDF $\rho(\omega_i, \omega_o)$ instead of just albedo
  - Global radiosity equation no longer linear
- Retains view independence
- Generally computationally infeasible

Two-pass radiosity + ray tracing

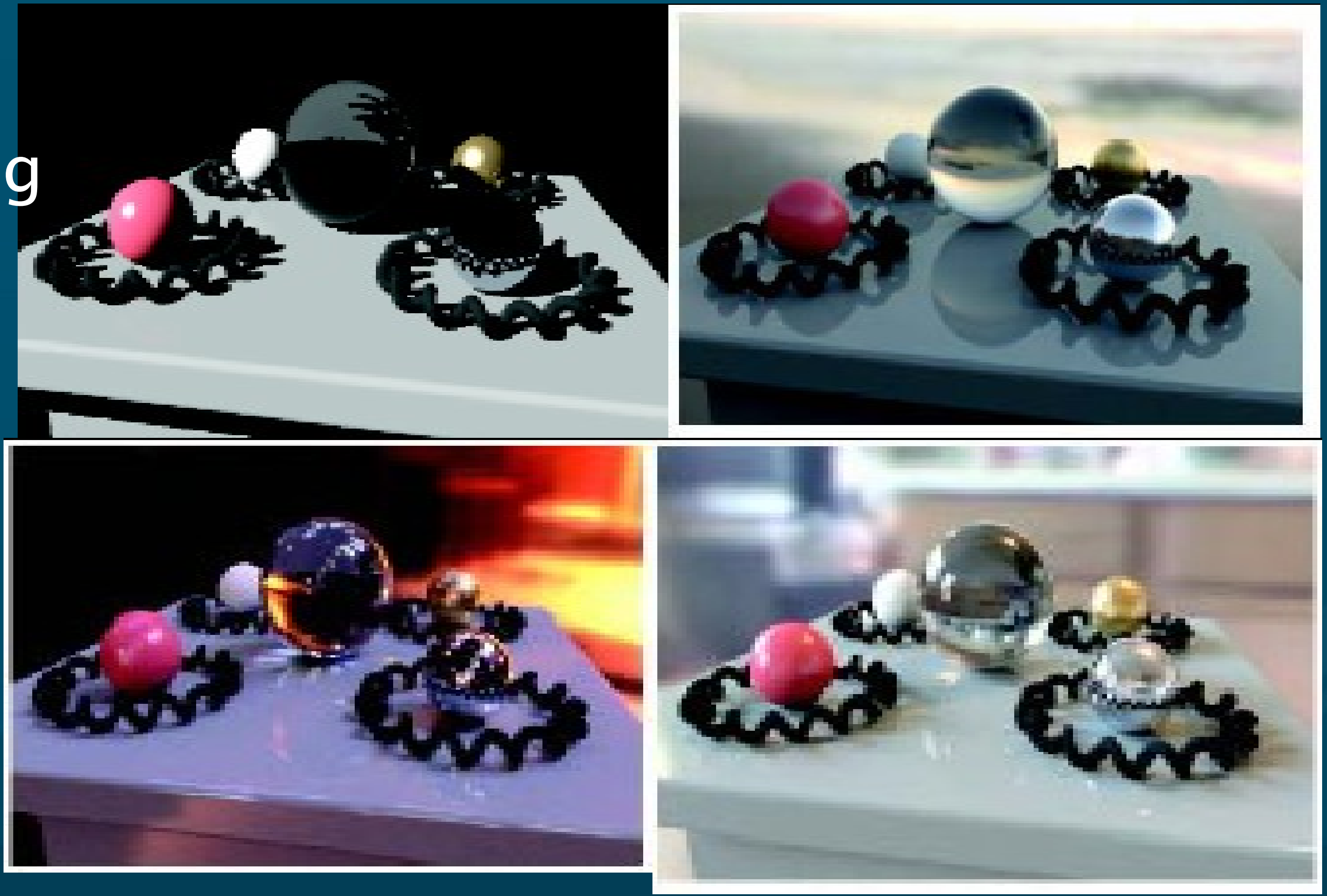
- Radiosity on first pass: view-independent
- Enhancements to classical radiosity:
  - Translucent surfaces: e.g., windows
    - Transmission of both specular and diffuse light
    - Include translucency in "window" form factor
  - Reflective surfaces: e.g., mirrors
    - Create a virtual (mirror-world) environment
    - Mirror becomes a window into the reflected world
    - Use transmission techniques
- Accounts for only one level of reflection

Pass 2: enhanced ray tracing

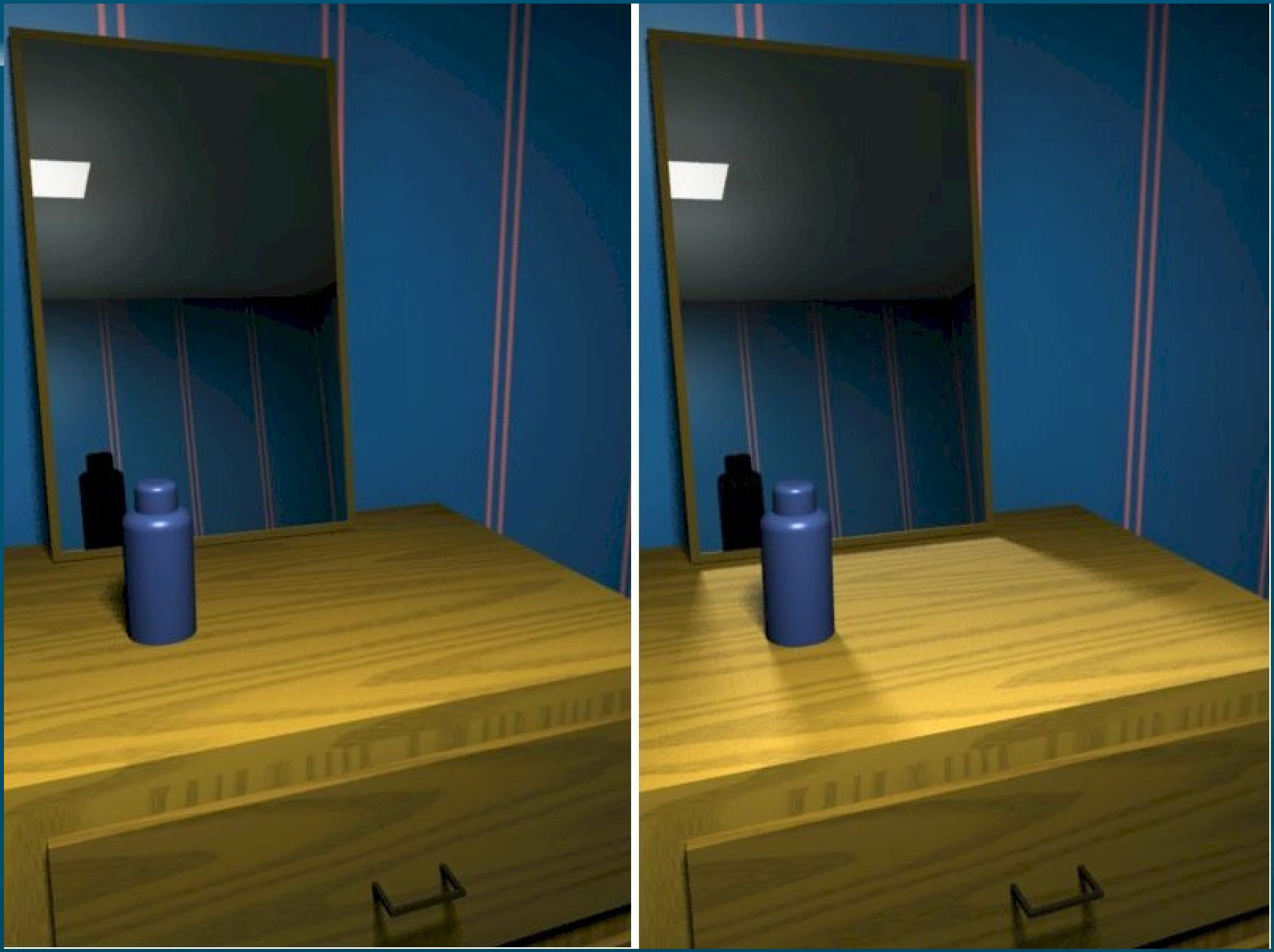
- Ray tracing on second pass: view-dependent
- Classical ray tracing only specular-to-specular:
  - Reflection ray
  - Refraction ray
  - Diffuse local illumination
- Specular-to-diffuse light transport:
  - Diffuse illumination should integrate all incoming light over a hemisphere
  - Approximate using a cone about reflection ray
  - If surface visible in cone is specular, recurse

Ray tracing: image-based light

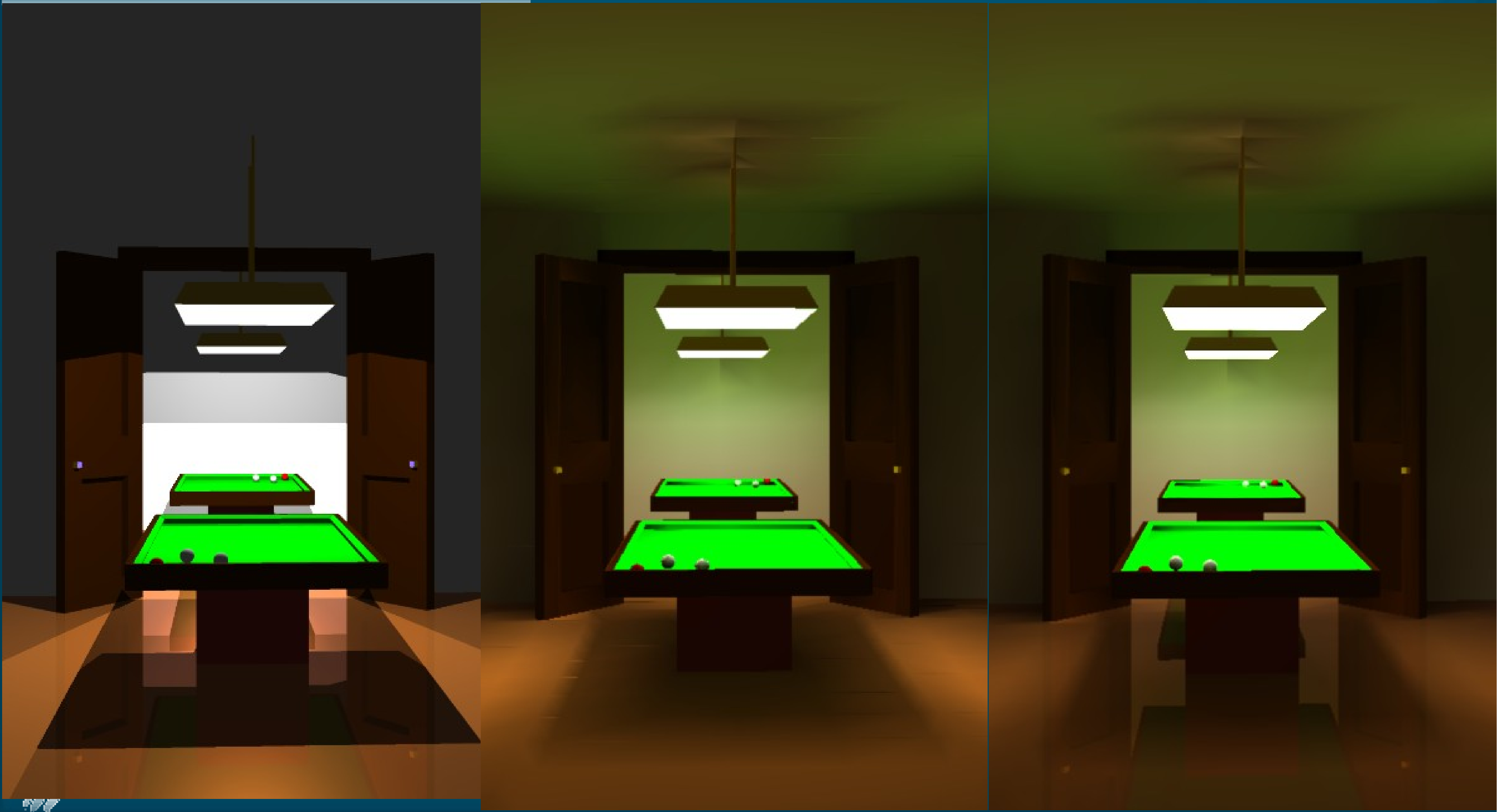
- Rays that go to infinity map to sky sphere
- Get colour from surrounding image
- Often used with HDR images



Classical radiosity vs. 2-pass



Raytrace vs. radiosity vs. 2-pass



tuwien Austria

TODO

- Lab5 due tonight
  - Virtual world
    - Creative, interesting scene
    - Lights and materials
    - Texture map
    - Bezier evaluator or NURBS
    - Pick objects
  - Final deadline for late labs: Thu 19Apr