Pure Path Tracing: the Good and the Bad

- Path tracing concentrates on important paths only
  - Those that hit the eye
  - Those from bright emitters/reflectors
- It treats every pixel independently, and every point on every surface independently
  - Good for specularities, which vary quickly over surfaces
  - Bad for diffuse illumination, which varies very slowly over surfaces (except at sharp shadow boundaries)
- No need for meshing, and general surfaces

Storing Information on Surfaces

- Radiosity algorithms store radiosity (and maybe more) in a mesh
  - Adaptation of the mesh requires new solutions
  - Methods to capture specular transport are expensive and don't incorporate importance
- Monte Carlo methods can store energy on surfaces
  - Tend to be view driven, or light source driven
  - "Mesh" points are generally random
  - Trade off noisy estimates for speed and generality

Ray Tracing Diffuse Transport

- Distribution ray tracing can in principle do diffuse interactions
  - Sample not only specular directions, but also diffuse
  - Cost precludes us, due to the vast number of rays required for adequate sampling
- Where is the expense?
  - Rays are sent to find information that may already have been found, but not recorded
- Solution: Store radiosity as it is found, and re-use

"Radiance" algorithm (Ward)

- When a diffuse contribution is computed, record it for later use
- Some questions:
  - How do we decide whether to use an old value, or compute a new one?
  - How do we store the contribution?
  - How many diffuse bounces to we trace?

Re-Using Values

- When a point is hit by a ray, and a diffuse value is required:
  - If a "good" estimate can be computed using stored results, use it
  - Otherwise, compute a new estimate for this point, and store it
- The quality of an estimate is based on geometric considerations…

Quality of Existing Data

- Examine:
  - Surface curvature between the required point and the existing data
    - diffuse illumination changes with curvature
  - Brightness and the distance to other surfaces
    - Influences how fast incoming illumination can change
  - Assume extreme variation in incoming radiance
- If existing values are used, weigh contribution by quality

Storing Radiosity Estimates

- Store samples in an octree
  - Same tree represents objects and speeds ray tracing
  - Radiosity values are stored at level appropriate to their quality range
    - Good estimates have value over larger regions, so are stored higher in the tree
- Problems?
  - Resolving close surfaces may be very expensive

Implementation Notes

- Compute illumination due to light sources directly
  - Except if the light source is large, in which case treat it like any other surface (with emittance)
- Build up “paths” one bounce at a time
  - Eye rays determine where estimates are needed
  - Schedule rays for next round, and continue
- Exploit numerical gradient information
- How does it relate to Kajiya’s path tracing?
- Which paths does it have trouble with?

Multi-Pass Monte Carlo Methods

- Shoot rays from lights, bounce them off surfaces, and deposit some energy at each bounce
- Options:
  - Which effects to capture with each pass
  - Bounce off all surfaces, or only off specular surfaces
  - Data structures for energy storage

Simplest Version

- Send out packets from light sources, distributed according to a particular model (eg spotlight, area light, …)
- Store some portion of the energy at each surface, on a predefined mesh
- Trace eye rays to render final image
- Problems:
  - No adaptation on receivers
  - View not accounted for in sending out packets

Adaptive Radiosity Textures

- Heckbert 90
- Store intensity in textures attached to the surfaces
  - Works for anything with reasonable texture coordinates
  - Texture is stored as a quadtree, and adapted
- Shoot from lights to add intensity to textures
- Also shoot from bright surfaces to get diffuse-diffuse transport
- Eye ray trace to render

Radiosity Textures Implementation

- Determine maximum texture resolution with an initial eye ray pass
- Paper does not describe complete implementation
  - No initial pass
  - No shooting from non-light sources
  - Only get LS*DS*E
- Difficulties:
  - Light rays must be sampled to fill textures on visible surfaces
  - Subdivision scheme require cooperation between lights and textures

Hybrid Radiosity Method

- Observation: Different techniques work best for different light transport paths
- Distribution ray tracing from the eye to get LDS*E
- Distribution ray tracing from the lights to get LS*DE
- Progressive radiosity and Monte Carlo path tracing to get D* paths
- Compute high frequency information per-pixel, and low frequency using progressive radiosity

Photon Maps

- As much an exercise in careful engineering as a new method
- Basic Idea: Store various "photon maps" on surfaces
  - Maps are balanced kd-trees on surfaces
  - Maps for caustics, diffuse illumination, volumes, …
  - Maps store incoming direction as well as position and energy

Producing the Image

- Use ray tracing to determine the visible points
- Radiance at a point is broken into several components:
  - One-bounce light from sources
  - Light reflected specularly from other points
  - Diffusely reflected caustics
  - Light reflected diffusely multiple times
- Each component is determined separately
  - Accurate method for directly seen light and "difficult" geometry
  - Approximate for diffusely reflected light (low weight)

Computing Contributions

- Direct illumination:
  - Accurate: Photon map gives approx. shadow, cast ray if not certain
  - Approximate: Use diffuse photon map directly
- Specular reflection:
  - Distribution ray tracing with importance
- Caustics:
  - Use caustics photon map directly
- Soft indirect illumination:
  - Accurate: “Radiance” style estimate
  - Approximate: Global photon map

Photon Map Summary

- Combines various techniques depending on the situation
  - Question: Can Veach’s multiple importance sampling be used to improve the results?
- Produces very good results reasonably efficiently
- Geometric problems can still cause bleeding

Global Illumination Summary