CS174A Lecture 15

Announcements & Reminders

- ***Nov 21: Demo by Prof Demetri on Biometric Human Simulation***
- ***Nov 24: Project assignment #4 due***
- ***Dec 1: projects due***
- ***Dec 3 and 5: project presentations, in class***
  - We will post some guidelines about the order of presentations, how to gather scores from teammates, class, etc.
  - Test your laptops, adapters, etc. beforehand with class projector
- ***Dec 12: final exam, 11:30 AM – 2:30 PM, Place TBD***
- ***Course evaluations: TBD***

TA Session This Friday

- ***Project #4***
- ***Team projects & logistics***

Last Lecture Recap

- ***Hidden Surface Removal***
  - Ray casting
- ***Ray Tracing***

Next Up

- ***Ray Tracing***
  - Issues: speed, shadows, aliasing
  - Stochastic ray tracing
- ***Transparent Objects***
- ***Particle Rendering***
- ***Volume Rendering***

Ray Tracing: Illumination

- ***Ray Tree***
  - Formed of primary and secondary rays
  - Evaluated bottom up
- ***Tree terminates if***
  - No intersection for a ray
  - Tree depth has reached a specified level
  - Intensity of I_r and I_t becomes very low

Ray Tracing: Speed

- ***Efficiency Considerations***
  - Total # of shadow rays spawned = $m(2^n - 1)$
    m = # light sources; n = depth of ray-tree
  - Total # of rays = $(m + 1)(2^n - 1)$
  - Back faces cannot be culled
  - Clipping cannot be done for view volume or behind eye
  - 75%-95% of time is spent in intersection calculations
  - Use bounding box testing or hierarchies (octrees)

Ray Tracing: Issues

- *Self shadowing due to numerical precision (surface acne)*
- *Shadow rays not refracted through transparent medium*
- *Specular illumination on backface polygons*

Ray Tracing: Aliasing

- ***Aliasing in RT***
  - Spatial aliasing
  - Temporal aliasing: for small objects

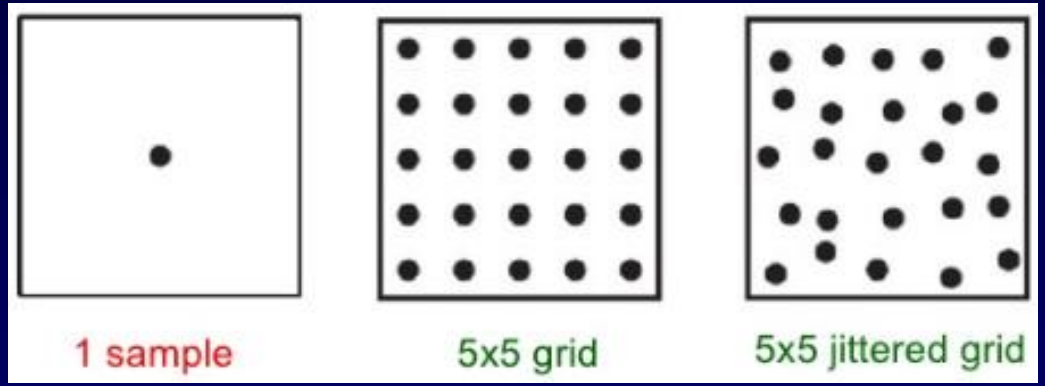
- ***Anti-Aliased Ray Tracing***
  a. Super-sampling
  b. Adaptive super-sampling (along edges of objects)
  c. Statistical super-sampling
  d. Stochastic RT

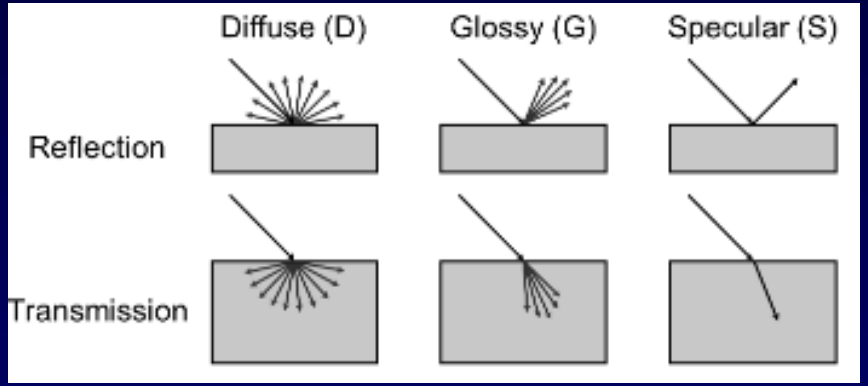
Stochastic Ray Tracing

AKA Distributed Ray Tracing

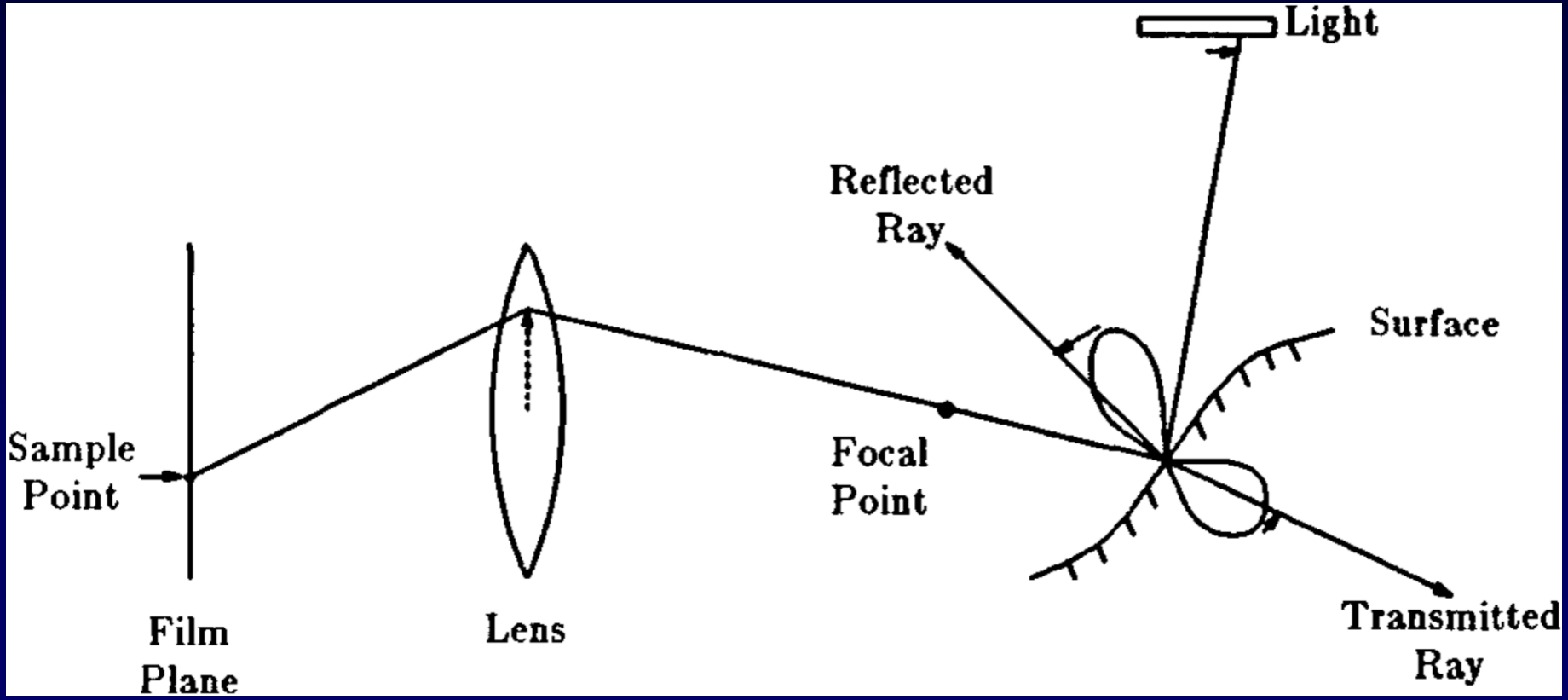
- *Antialiasing: distribute over pixel sampling area*
- *Gloss: distribute reflected ray*
- *Translucency: distribute refracted ray*
- *Penumbra: distribute shadow rays*
- *Depth of Field: distribute over lens diameter*
- *Motion Blur: distribute across frames*

Stochastic Ray Tracing



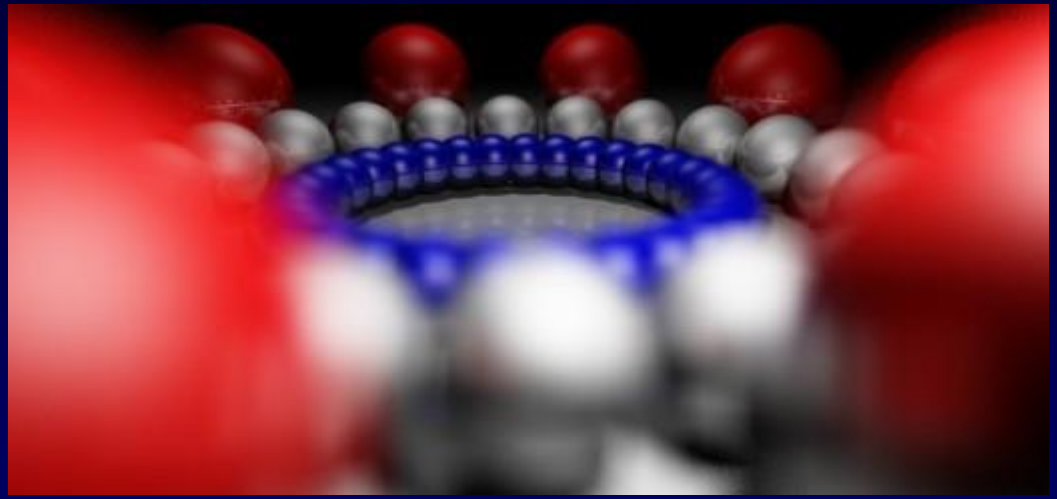


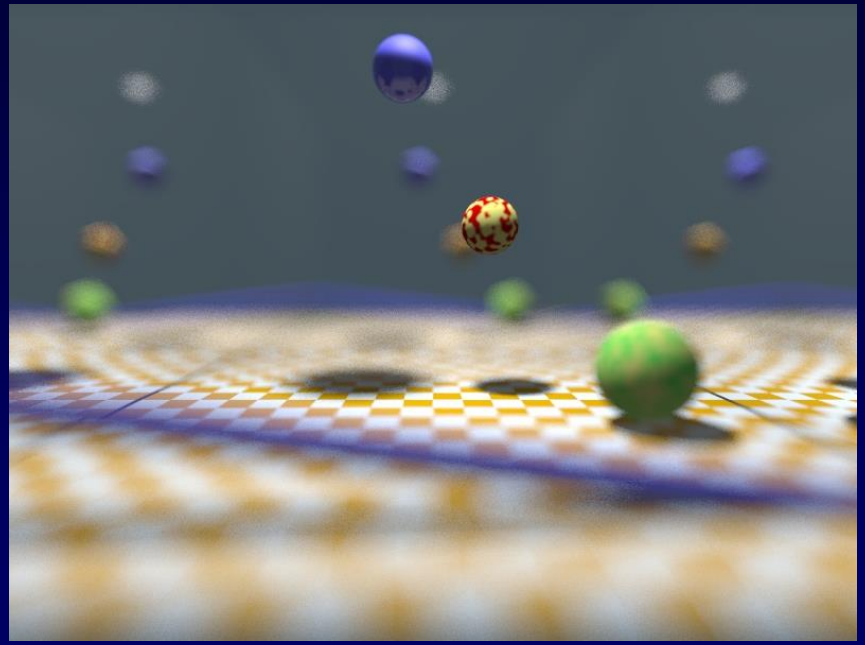
Stochastic Ray Tracing



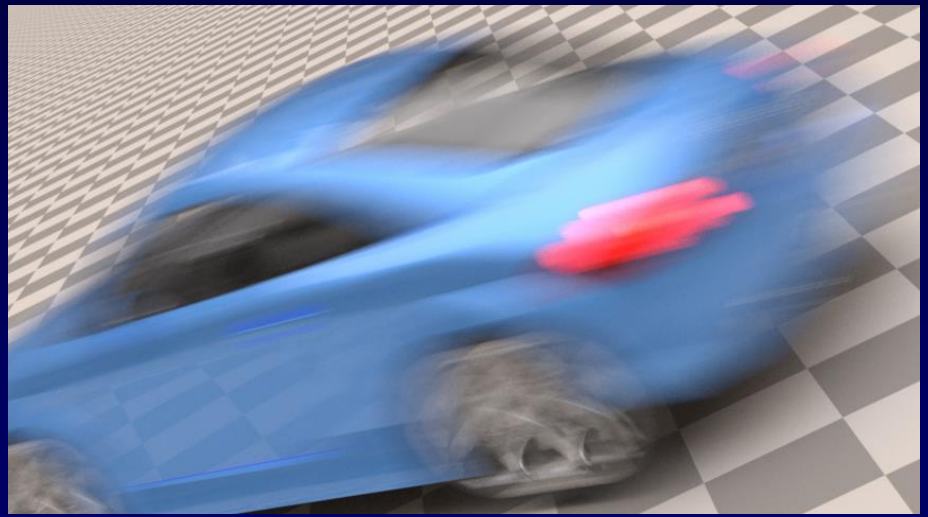
Stochastic Ray Tracing

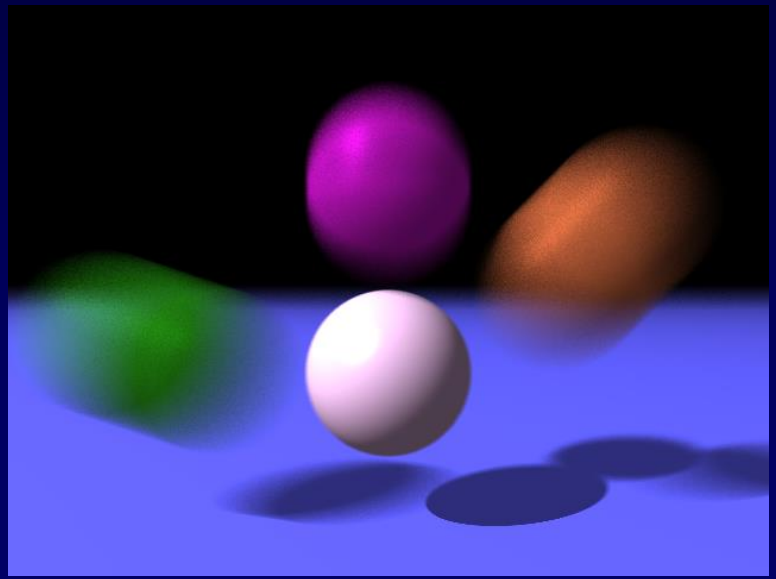
Stochastic Ray Tracing: Depth of Field





Stochastic Ray Tracing: Motion Blur





Transparency/Opacity

- ***Matte: coverage info***
- ***Add a 4^{th} channel to color: α = opacity (RGBA), range [0..1]***
- ***α = 0 ⇒ fully transparent; α = 1 ⇒ fully opaque***
- ***Transparency = 1 – Opacity***
- ***Applications***
  - Image compositing, e.g., combining computer-rendered images with live footage
  - Particle rendering, e.g., smoke, snow, fire
  - Volume rendering

Transparency/Opacity: Blending

- ***Pre-multiplied vs straight alpha***
- ***Operators: over, in, out, atop, xor***
- ***Alpha blending or alpha compositing (over operator)***
  - Straight

    $$C_0 = \frac{\boldsymbol{C_f}\alpha_f + \boldsymbol{Cb}\alpha_b(1-\alpha_f)}{\alpha_f + \alpha_b(1-\alpha_f)}$$

  - Pre-multiplied

    $$c_0 = c_f + c_b(1-\alpha_f)$$

    $$\alpha_0 = \alpha_f + \alpha_b(1-\alpha_f)$$

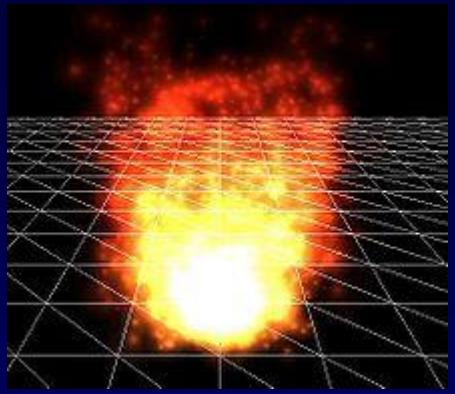
Transparency/Opacity

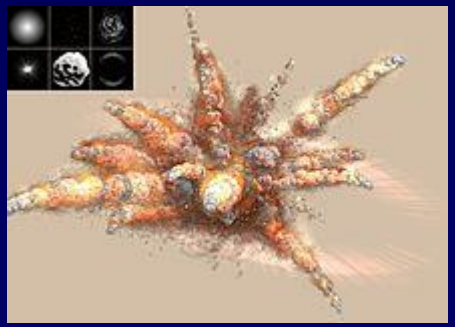
- ***Example: Water Flowmap***
- ***Other good examples:***

  marchingcubes, *translucency*, *water / flowmap*, *lod*, *nearestneighbour*, *youtube*, *orientation / transform* (quaternion math), *reflectivity*, *manualmipmap*, *multiple elements*, *shadowmap viewer*

Particle Rendering

- ***Used for simulating fuzzy phenomena***
- ***Examples: chaotic, natural or chemical systems***
- ***Like fire, smoke, explosions, snow, dust, etc.***
- ***Modeled as an emitter, like sphere or box***
- ***Particle behavior params***
  - Spawning rate
  - Initial velocity vector
  - Lifetime
  - Color, etc.
  - Collision?





Particle Rendering

- ***Rendered usually as textured bill-boarded quads***
- ***In games, as a single pixel***
- ***Examples:***
  - https://www.youtube.com/watch?v=tfghiimtgyY&list=PLAwxTw4SYaPlaHwnoGxJE7NFhEWRCIyet&index=391
  - https://www.youtube.com/watch?v=jsPfaQ7aMqk&list=PLAwxTw4SYaPlaHwnoGxJE7NFhEWRCIyet&index=499