Dhiren Bhatia
Carnegie Mellon University

15-462 Computer Graphics Final Review

Administrative Stuff

- University Course Evaluations available online – Please Fill!
- December 4 : In-class final exam
  - Held during class time
  - All students expected to give final this date
- Alternate final date:
  - December 16th :- 10:00 am-11:30 am
  - Must get Prof. Pollard’s permission by Dec 2nd

Administrative Stuff

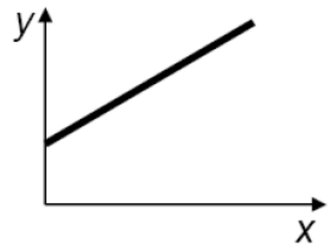
- Project 4 - Due Dec 6th
  - No late days!
- Final grade break-down
  - Midterm – 15%
  - Final – 25%
  - Homework – 20%
  - Projects –
    - P1: 7%
    - P2: 10%
    - P3: 15%
    - P4: 8%

Final Information

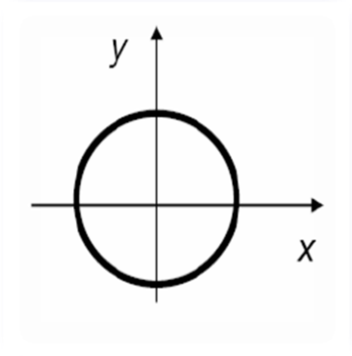
- Closed book
- No cheat sheet
- Everything covered so far in the semester
- Review HWs and Lecture Reviews

Equations

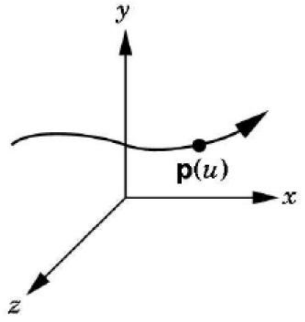
- Explicit:
  - $y=f(x)$, $y=mx+c$



- Implicit:
  - $f(x,y)=0$, $x^2+y^2=r^2$

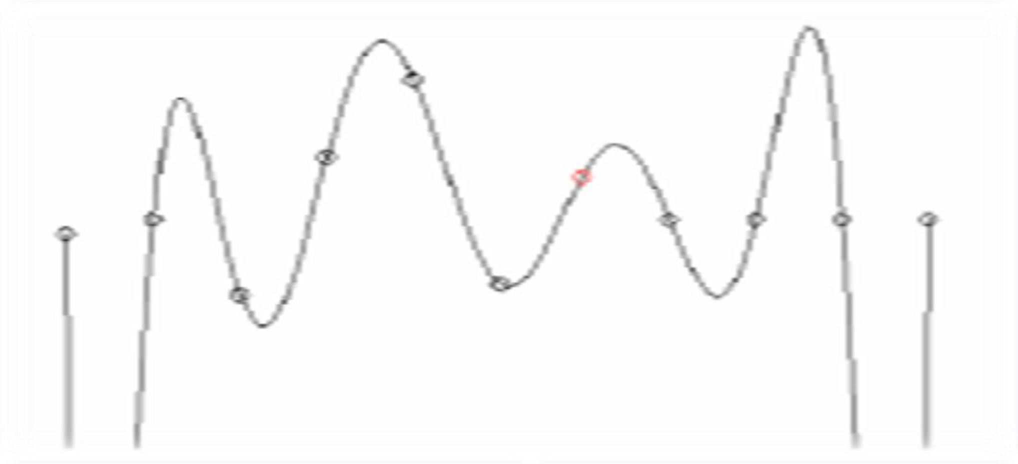


- Parametric:
  - $(x,y)=(f(u), g(u))$
  - $(x,y)=(\cos(u),\sin(u))$



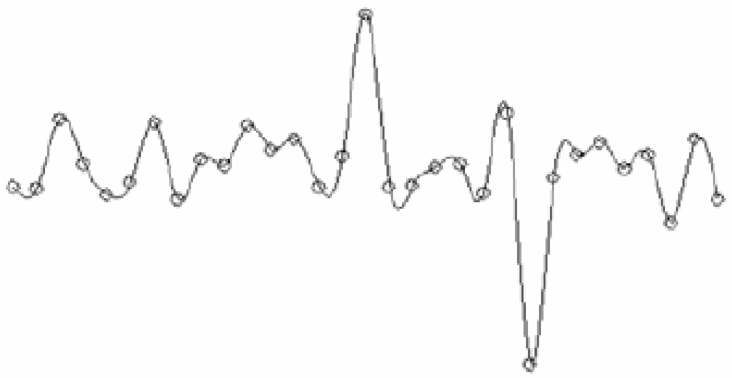
Polynomial Interpolation

- An n-th degree polynomial fits a curve to n+1 points
  - Cons: Change to any control point affects the entire curve



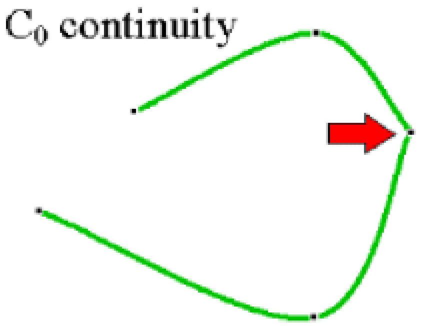
Splines

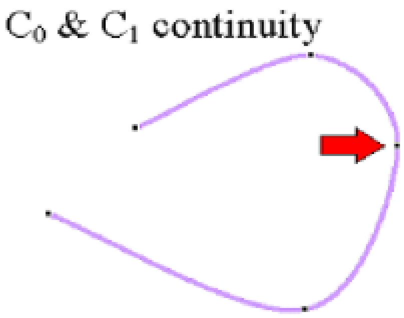
- A spline is a piecewise polynomial – many low degree polynomials are used to interpolate the control points
  - Most common: Cubic piecewise polynomials



Continuity

- C_0, C_1 continuity:
  - C_0 continuity: Continuous in positions
  - C_1 continuity: Continuous in positions and tangent vectors (first differential)





Cubic Polynomial Form

- Each P and C_k is a column vector
- From control information (points or tangents)
- Goal – to determine cubic polynomial form

$$\mathbf{p}(u) = \mathbf{c}_0 + \mathbf{c}_1 u + \mathbf{c}_2 u^2 + \mathbf{c}_3 u^3 = \sum_{k=0}^{3} \mathbf{c}_k u^k$$

Hermite and Bezier Curves

- Hermite: Two points on the curve and two tangents
- Bezier: Two points on the curve and two points outside of the curve

$$\mathbf{M}_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \qquad \mathbf{M}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

Transformations

- Linear Transformations
  - $T(v_1+v_2)=T(v_1)+T(v_2)$
  - $T(\alpha v)=\alpha T(v)$
- Rotation
- Scaling
- Translation

Rendering Equation

$$L(t,p,\omega_0,\lambda) = E(t,p,\omega_0,\lambda) + \int_{\omega} L(t,p,\omega_0,\lambda)\rho(p,\omega,\omega_0,\lambda)\,|n.\omega|\,d\omega$$

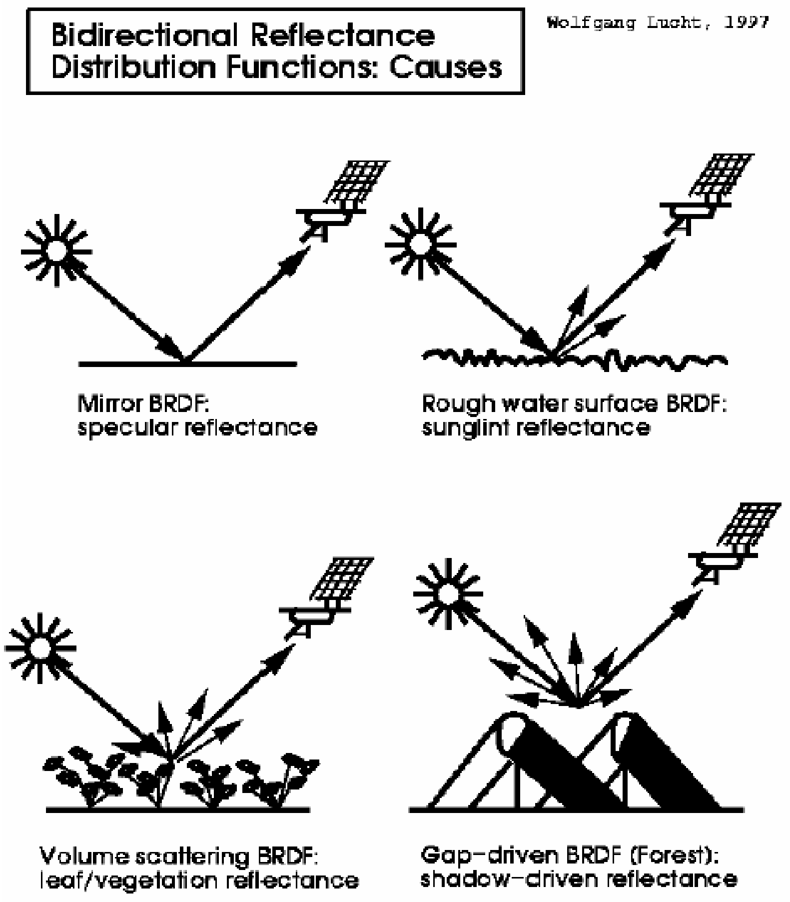
- Key Idea:

[outgoing]=[emitted]+[reflected]+[transmitted]

- Don't forget how to connect this with Radiosity equation

Bidirectional Reflectance Distribution Function (BRDF)

- $\rho(p, \omega, \omega_o, \lambda)$ in the rendering equation is the BRDF, which gives the reflectance of a target as a function of illumination geometry and viewing geometry.



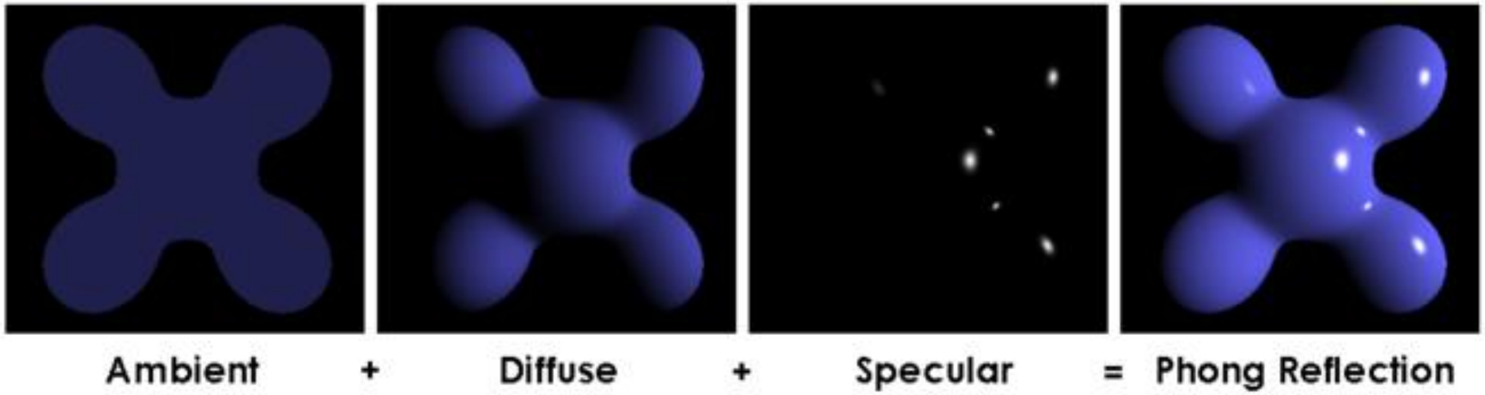
Local vs. Global Rendering Models

- Local rendering models
  - Object illuminations are independent
  - No light scattering between objects
  - No real shadows, reflection, transmission
- Global rendering models
  - Ray tracing (highlights, reflection, transmission)
  - Radiosity (surface interreflections)

Phong Illumination

- Local Illumination model
- Sum of three components:
  - Diffuse reflection + Specular reflection + Ambient term

$$I_p = k_a i_a + \sum_{\text{lights}} (k_d (L \cdot N) i_d + k_s (R \cdot V)^{\alpha} i_s).$$

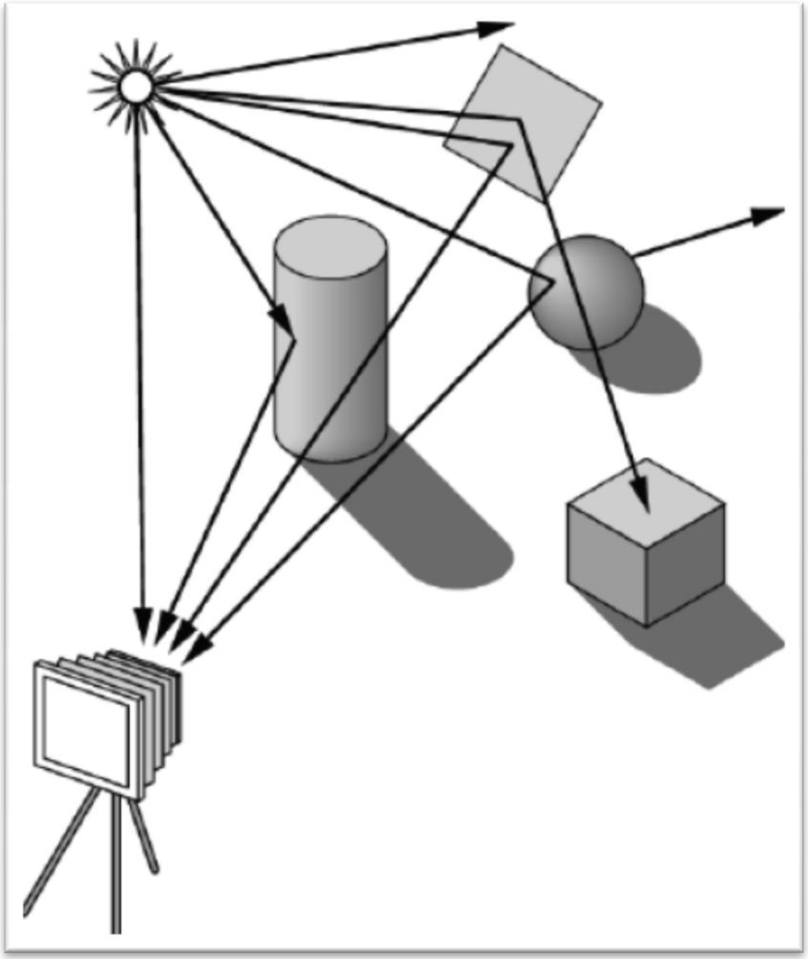


Object Space vs. Image Space

- Graphics pipeline: for each object, render *(online)*
- Ray tracing: for each pixel, determine color *(off-line)*
- Radiosity: for each two surface patches, determine diffuse interreflections *(off-line)*

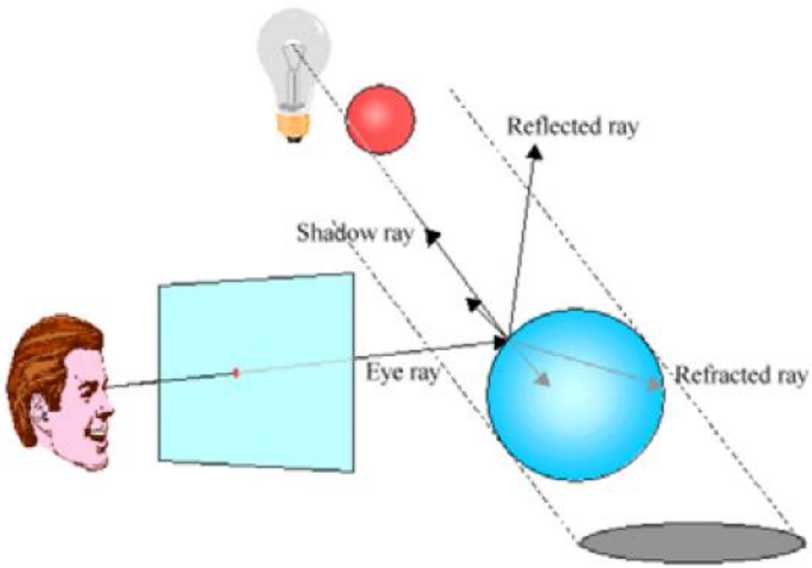
Forward Ray Tracing

- Forward ray tracing follows the photon in direction that light travels from the source
  - Con: Only a tiny fraction of rays actually reach the image



Backward Ray Tracing

- Backward ray tracing starts the image and follow the ray until the ray finds (or fails to find) a light source

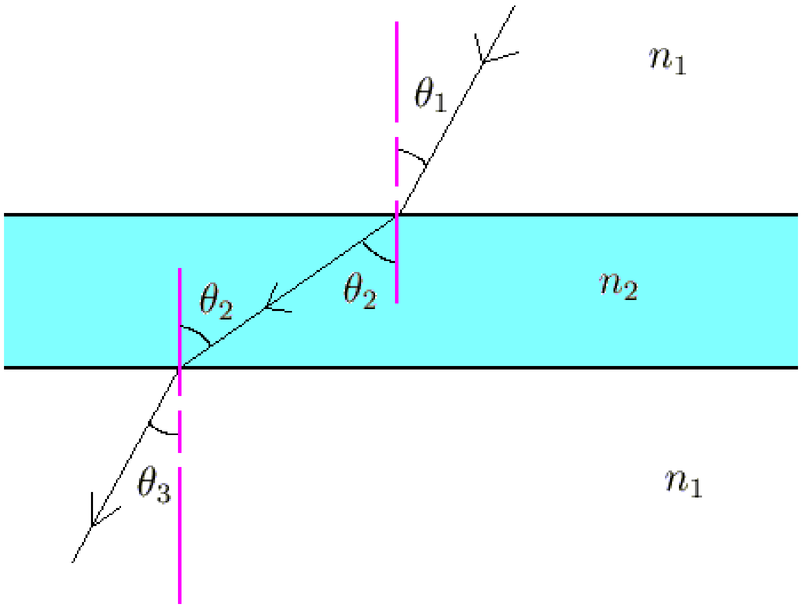


Rays in Ray Tracing

- Eye rays: originate at the eye
- Shadow rays: from surface point to the light source
- Reflection rays: from surface point in mirror direction
- Transmission rays: from surface point in refracted direction

Snell's Law

- n_1 is the refractive index of the first material
- n_2 is the refractive index of the second material
- Key Idea: The ratio of the sinusoids of the angles of incidence and of refraction is a constant depending on the media



$$n_1 \sin(\vartheta_1) = n_2 \sin(\vartheta_2)$$

Different Intersection Tests

- Ray-Sphere intersection
- Ray-Polygon intersection
- Ray-Quadric intersection

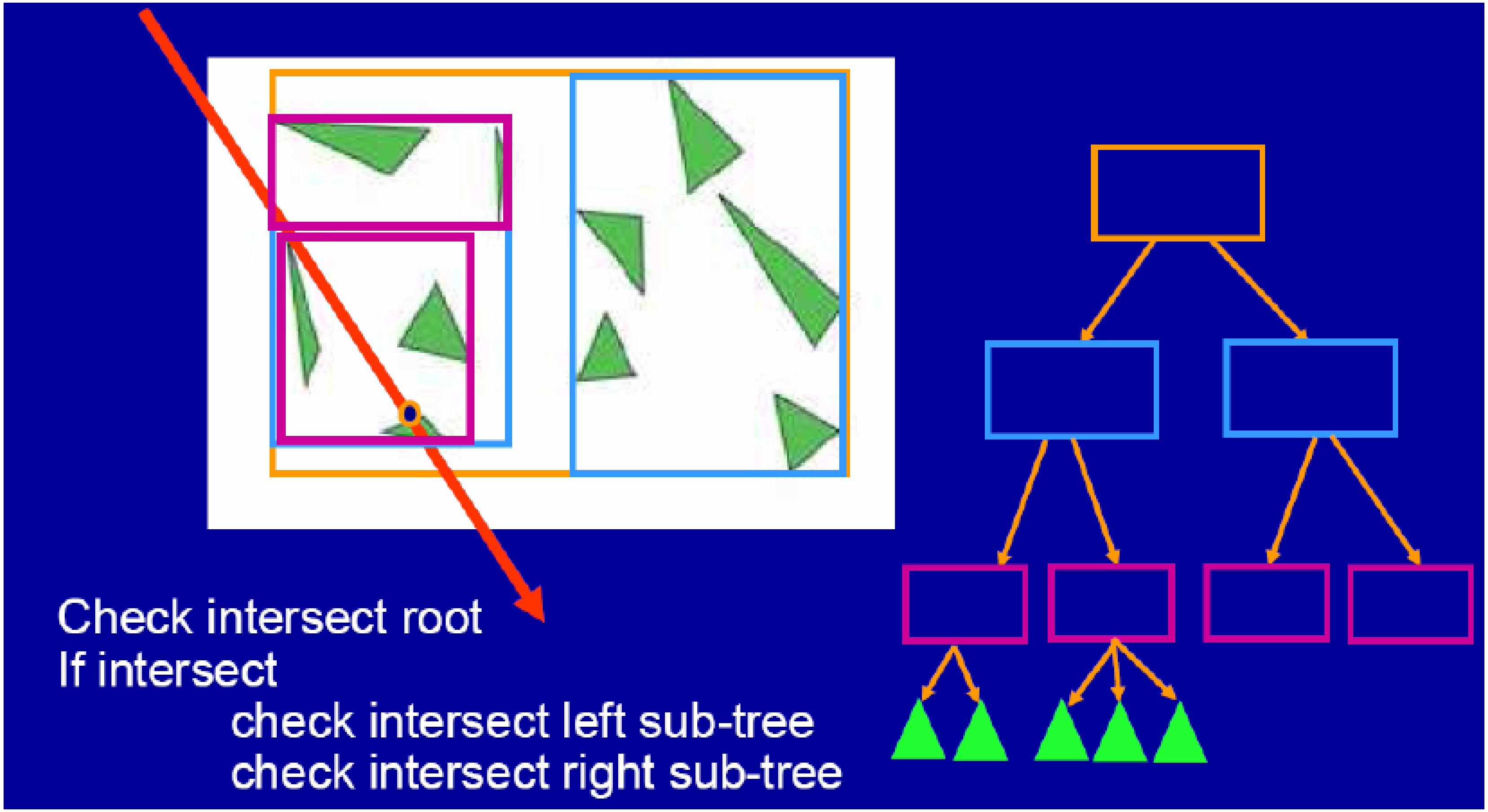
Hierarchical Data Structures

- Good data structures greatly speed up ray tracing
- Up to 10x or even 100x!
- You need to know:
  - Hierarchical bounding volumes
  - Grids
  - Octrees
  - K-d trees and BSP trees

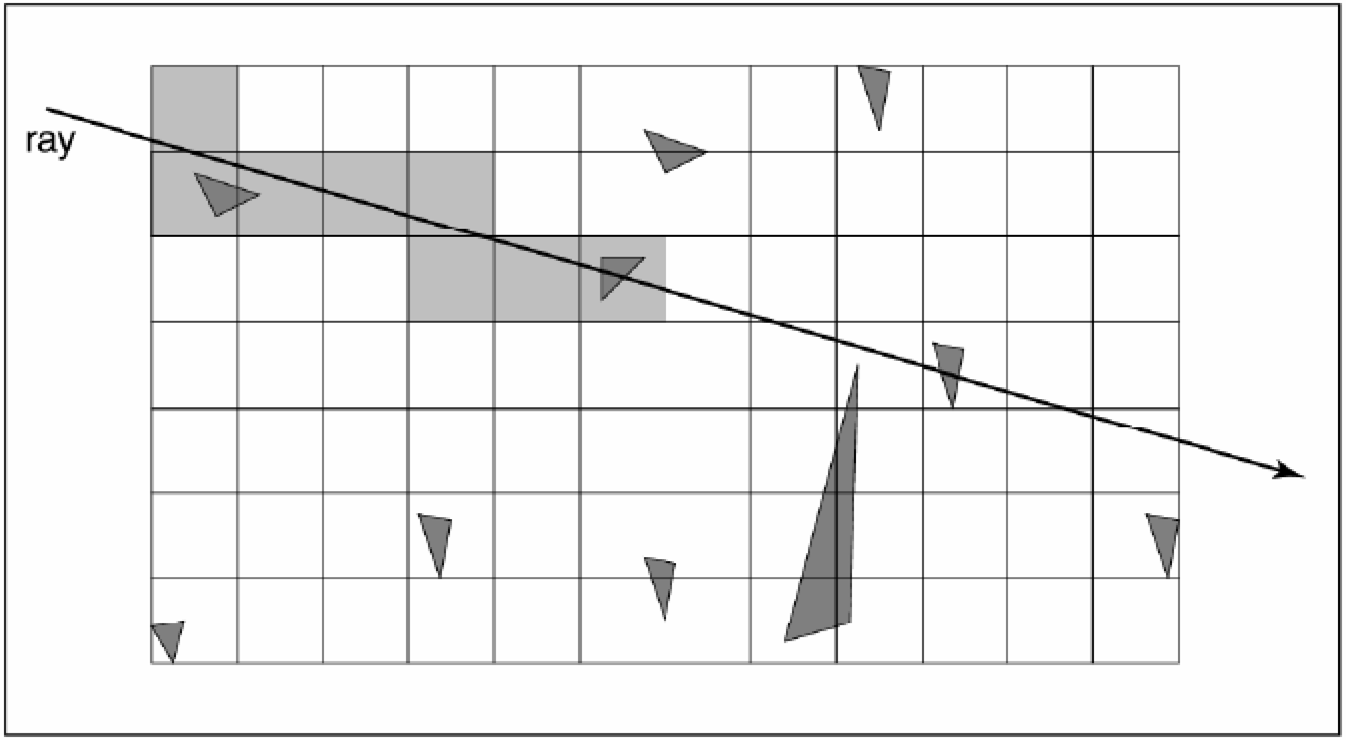
Bounding Volumes

- Wrap complex objects in simple ones
- Does ray intersect with the bounding box?
  - Yes: Calculate intersection with enclosed objects
  - No: Does not intersect enclosed objects
- Effectiveness depends on:
  - Probability that ray hits bounding volume, but not enclosed objects
  - Expense to calculate intersections with bounding volume and enclosed objects

Hierarchical Bounding Volumes

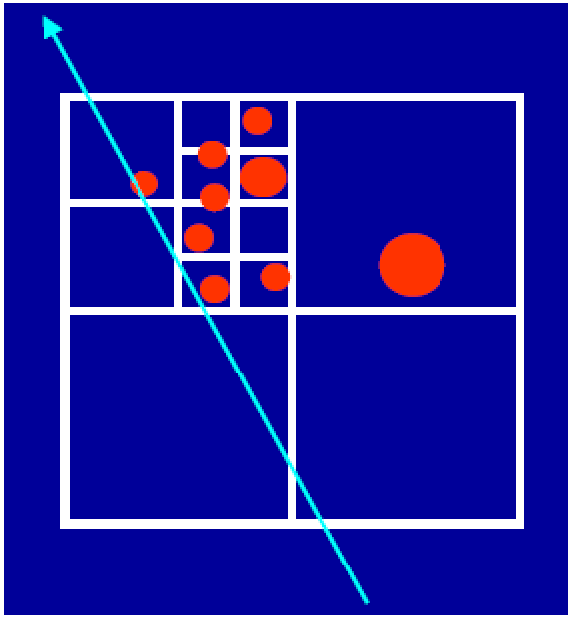


Grid

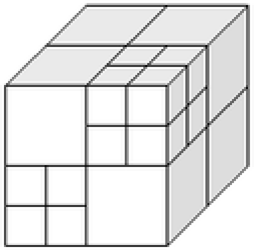


Octrees

- Quadtree is the 2-D generalization of binary trees
  - Node is a square
  - Recursively split into four equal sub-squares



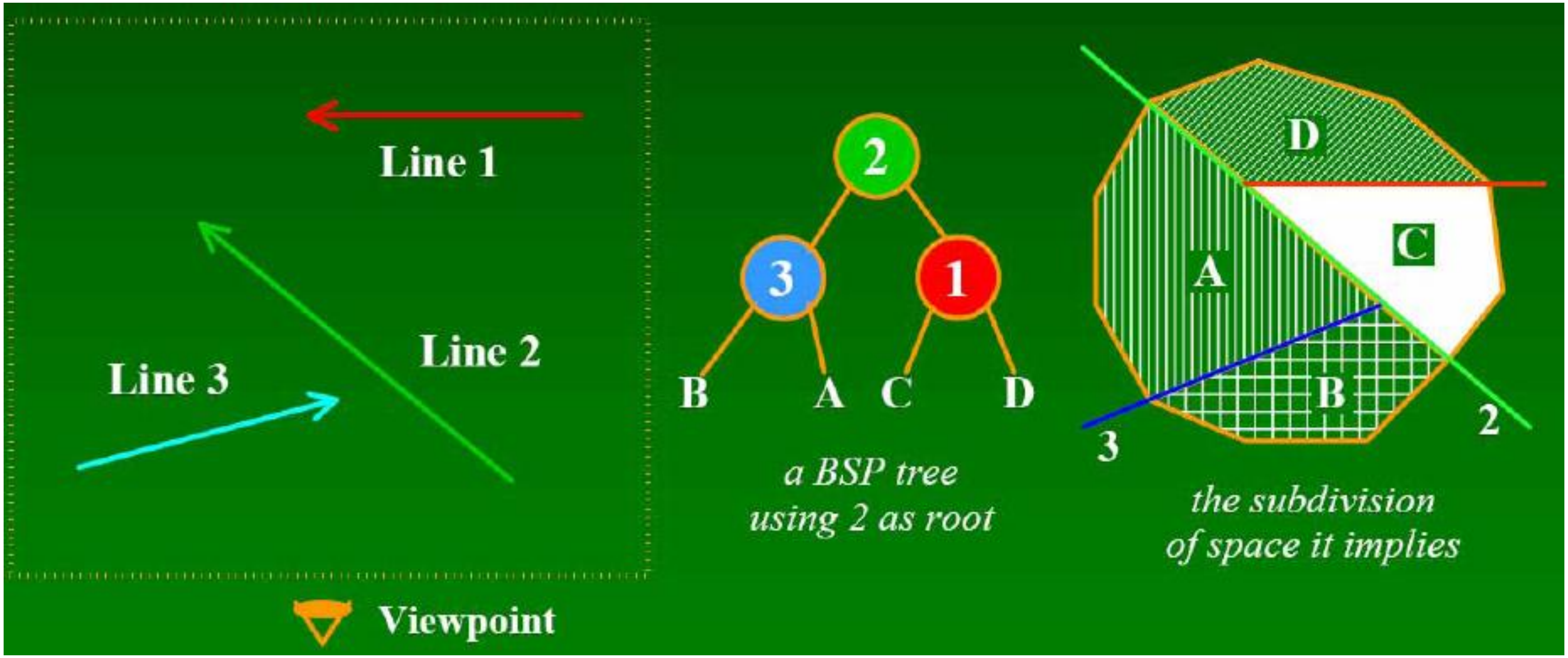
- Octree is 3-D analog of Quadtree



K-d Trees and BSP Trees

- K-Dimensional (K-D tree)
  - Split only one dimension at a time along the coordinate axis
- Binary Space Partitioning (BSP) tree
  - Permits splits with any line
  - In 2-D space split with lines
  - 3-D space split with places
  - K-D space split with k-1 dimensional hyperplanes

Example of BSP tree



Ray Tracing vs. Radiosity

- Both are global illumination compared with the Phong model (local illumination)
- Ray Tracing: Realistic specular reflection/transmission
- Radiosity: Realistic diffuse reflection
- Know the differences

Radiosity Energy Balance Equation

- Unknown: radiosity B_i
- Known: emission E_i, form factor F_{ij}, reflectivity R_i
- Know how to calculate form factor
- Know the matrix formulation to solve equation

$$B_i = E_i + R_i \int_j B_j F_{ij}$$

Global Illumination

- Photon mapping
  - First pass: construction of the photon map
  - Second pass: rendering equation is used to estimate the radiance of every pixel.

Lastly

- Overview not comprehensive!
  - Must study all the material outside these notes
- Good luck for finals week!