Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

Module 3A
Ray Tracing Efficiency

Ray Tracing Efficiency

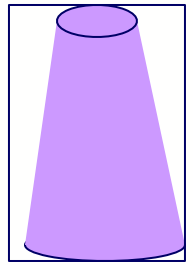
- Aim to reduce the number of intersection tests required
  - Possible techniques
    - Using bounding volumes to speed intersection testing
      - Bounding objects with simple shapes
    - Hierarchy of bounding volumes
      - Discussed in Lecture 1
- First hit speedup
- Adaptive depth control
- Light buffers to improve shadow efficiency
- Spatial coherence methods
  - Spatial partitioning
  - Shadow caching

Bounding Volumes

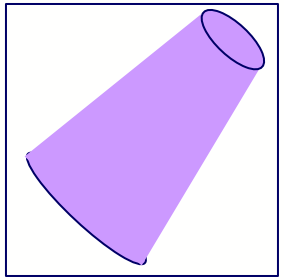
- Whitted used a scheme of bounding volumes to speed intersection calculations
  - If a ray does not intersect the bounding volume the object can be skipped
    - Will not intersect the bounded object
  - Can save time for objects represented as large number of polygons
- Bounding volume efficiency considerations
  - Fast, efficient ray / bounding volume test
  - BV should tightly bound the object
    - Most rays intersecting the volume intersect the object
- We will use spheres and Axis Aligned Bounding Boxes
  - To enclose any polygon mesh objects

Bounding Volumes for Transformed Objects

- BV tests can be done in either world coordinates or local coordinates
- Local coordinates
  - BV stored in local coordinates
  - Transform ray to local coordinates
  - Test against BV
    - BV stored in local coordinates
- World coordinates
  - Form object's BV in world coordinates
    - Can be done during initialization
  - Test ray against the transformed BV
  - Advantage: removes need to transform ray to local coordinates
  - Disadvantage: BVs may not bind the objects as tightly
    - See AABB example at right
- Bounding sphere – test in world coordinates



Local Coordinates



World Coordinates

First Hit Speedup

- First-hit speedup methods aim to more quickly find the intersection of the ray from the eye through the view plane
- Pre-render scene using a modified depth buffer algorithm
  - Store a pointer to the nearest object at each pixel
  - Eliminates intersection calculations for first hit
- F.S. Hill discusses using **projection extents**
  - **Section 12.10.2**
  - Rectangular extent of the projected object onto the view plane
  - Preprocessing step to determine projection extents of all objects
  - Project vertices of the object onto the view plane
    - Form min,max x,y
  - When tracing the initial ray from the eye first check if pixel on view plane is within the projection extent of an object
    - Simple bounds check
    - If not within extent, no need to test for object or its BV intersection

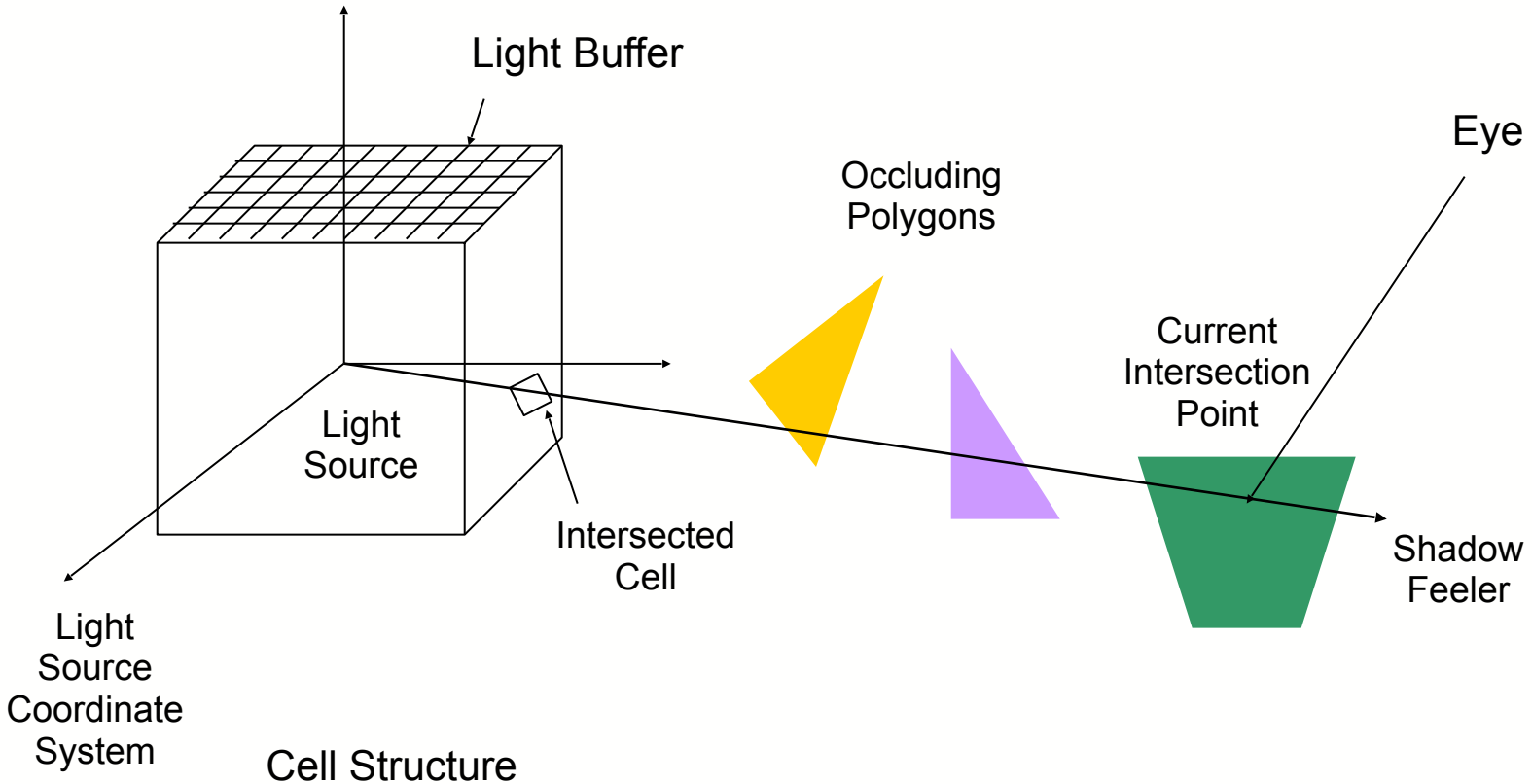
Adaptive Depth Control

- Inefficient to trace to a maximum depth
  - Especially if majority of objects are not highly reflective
- Adaptive depth control traces rays until global reflectance drops below a threshold
  - Contribution of ray depends on the product of global reflectivity coefficients of the materials the ray has intersected
    - $k_1k_2k_3\ldots k_n$
    - Once below some threshold, further traced rays will contribute minimal color changes
    - Three attenuation coefficients: R,G,B
- Also can account for transmitted rays
- Hall and Greenberg test
  - Using a max. depth of 15 they reported an average depth traced to be 1.71
    - Savings depends on the types of object materials and the distribution of objects

Light Buffer to Improve Shadow Performance

- Haines and Greenberg (1986) use a **light buffer** to accelerate shadow testing
  - Reduces shadow testing time by a factor of between 4 and 30
- Light buffer for each light source
  - Two dimensional arrays of 6 cube faces surrounding the light source
  - Scene polygons are cast onto the face of each cube
    - Light source is center of projection
  - Each cell contains a list of polygons that can be seen from the light source
    - Polygons sorted in ascending depth order (distance from the light source)
- For each shadow ray - the light buffer provides a list of polygons that may shadow the intersect point in question
- High storage requirements depending on number of light sources and resolution of the light buffer

Light Buffer



Cell Structure

Object ID	Polygon ID	Depth

Shadow Caching

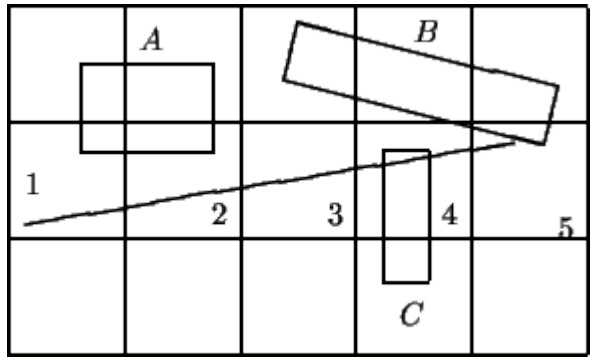
- Shadow caching was suggested by Haines and Greenburg in 1986
- Assumption: neighboring points in a shadow are probably shadowed by the same object
  - This is a from of spatial coherence
  - Shadow feelers can be a large part of the cost of rendering a scene
- Method
  - Each light source stores a reference to the object that has most recently cast a shadow from this light source
    - Null if last shadow ray to this light source did not intersect any objects
  - When checking a shadow ray to this light source, first check the object in the shadow cache
- Relatively easy to implement

Spatial Partitioning and Spatial Coherence

- Spatial partitioning: adds additional structure to the objects to constrain the search
  - Space occupied by the scene is subdivided into labeled regions
    - Label with a list of objects contained within
  - Pre-processing stage
- Spatial coherence methods make use of spatial partitioning
  - Find the region of the ray origin
  - Check for intersections with any objects within this region
    - If any are found this will include the closest intersection point
  - If none are found, ray enters the “next” region along its path
  - Limits unnecessary intersection checking
- Spatial coherence methods “focus” the search

Uniform Subdivision

- Divides scene/world into uniform 3D cells or grids
  - Often called SEADS (Spatially Enumerated Auxiliary Data Structure)
- Advantages
  - Simplicity
  - Rays can be traversed through the cells very efficiently
    - By a 3D-DDA calculation
  - Speed of traversal often offsets the disadvantages of the uniform space subdivision
- Disadvantages
  - Takes more space than octrees or BSP trees
  - Often a large number of empty cells



Spatial Partitioning

- Non-uniform subdivision techniques include octrees and BSP trees
  - Advantages:
    - Space efficient
    - Allow rays to efficiently pass through large empty areas in a scene
  - Disadvantages
    - Harder to create
    - More costly to compute next cell along a ray
- Octree
  - First used in ray tracing by Glassner
  - Requires less space
  - Subdivision takes time
- Binary Space Partition (BSP) Tree
  - Trees more balanced
  - Partitioning more flexible
- Will discuss each in a little more detail in Module 5
  - Without detail of how they may be used in ray tracing