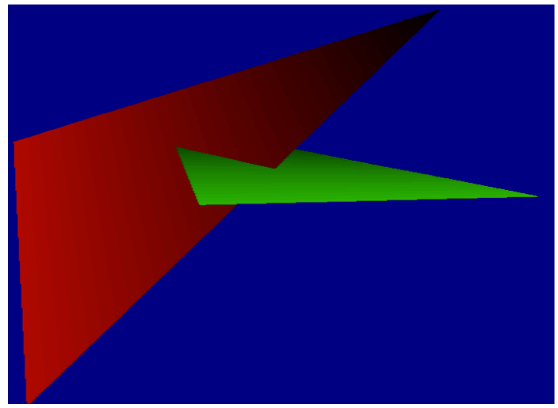
Collision detection [Continued]

Yezhisai Murugesan

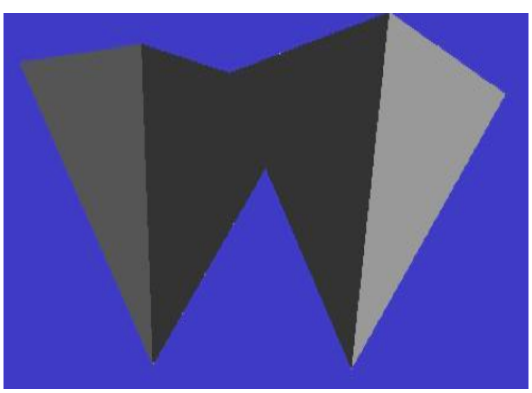
02.28.11

What happens when triangles of two largely triangulated models (say about 10000+ triangles) intersect? What happens when sets of triangles intersect? How do we detect Collision Detection? What is the fundamental problem in Collision Detection? Is there a fast way to enumerate triangles of objects that are already complex?



One simple and obvious way to detect Collision Detection is to check every triangle (from the list of intersecting triangles) of one object against every triangle in the other object. Although we get a valid output, this approach is very slow. This method of detecting collision is done only with models less than 100 triangles. Also, by breaking up objects into triangles, we have NxN triangles (Worst case). Very expensive brute force!

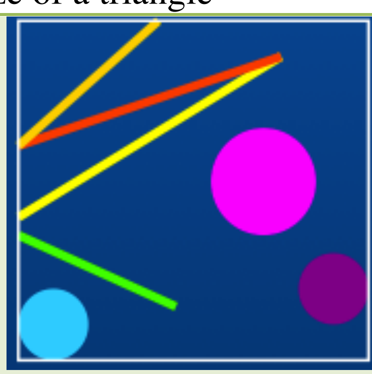
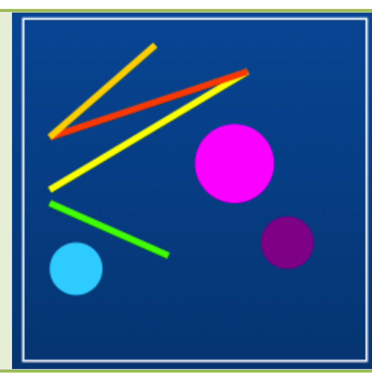
Another alternative approach is to break the object into convex pieces and run specialized techniques for convex v/s convex collision detection. Esoteric and not commonly used.

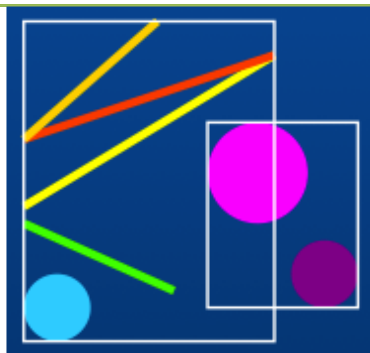
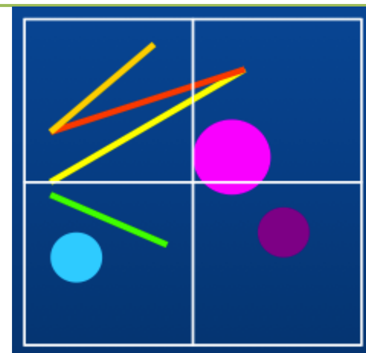
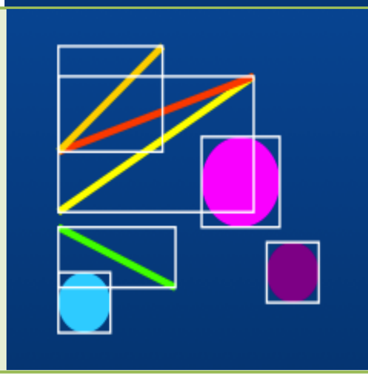
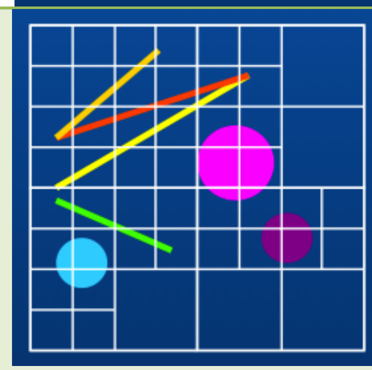


How does one check if two triangles intersect each other (in 3D)? Use **Separating Axes Theorem**. Test limited number of directions. If there is an overlap in the direction of the projected triangles, then collision.

Tri-tri Overlap function involves passing 3 vertices, from each of the triangles(18 input parameters in total). The output is a boolean – true if colliding, false if not.

Two approaches in detecting collision for complex objects:

Bounding Volume Hierarchy	**Spatial Decomposition**
Most common data structure	Less Common Data Structure
Object Centric	Space Centric
The idea is to enclose the bunny within a box/sphere, which is the root of the hierarchy (tree). Smaller, specialized boxes form next level of the hierarchy. Each of these boxes is responsible for covering part of the model.	When objects are far apart from each other, it is not necessary to check each of the boxes. Checking if bounding boxes of each object overlap. By bringing the objects closer, there may be a collision between boxes but not the actual objects or there may be a collision between objects.
Partitions the Object. Root that covers the entire model has k children (~k boxes). Union of k boxes forms the large model. Recursively decompose each of the k boxes into smaller boxes. Recursion stops when the size of the box equals the size of a triangle	Partitions the Space. Store integer identifier for every object as an array index. List of objects that are inside/intersecting the cell. Create 2/3/4D lists for every box and then traverse through all cells (analogous to scan conversion technique)
	

	
	
Trees of spheres, Ellipses, Cubes, Axis-aligned bounding boxes (AABB), Oriented Bounding boxes (OBB), k-DOP, SSV etc.	BSP, K-d trees, octrees, MSP trees, R-trees, grids/cells, space-time bounds etc.
Space Redundancy	Object Redundancy

Thus, Running time depends on or is proportional to how difficult the case of collision detection is based on say, proximity of the two objects. Double Helix and boxes (with epsilon difference/gap) are examples of a difficult case of collision detection.

Negative 3D image

- Another example of a difficult case of Collision Detection.
- Gigantic Stanford bunny penetrated through the wall will leave a negative image (3D bunny hole) of the bunny in the wall.
- Shrinking the bunny by a scale of 0.999 and re-penetrating it with an epsilon gap will vary the results.

<u>Continuous Collision Detection</u>

To detect if two objects collided 'between' two consecutive, discrete time steps. Although there was no collision at say, t=100, there might have been a collision at t=101 and we missed collision in between.

Swept volume

- Check if a cylindrical capsule and parallelepiped (obtained by sweeping a sphere and box along a line in 3D) are in contact with each other.
- If they are not intersecting, then collision hasn't occurred. If they are, then collision 'might' have happened (Potential Collision).
- Volumes may intersect but didn't meet in time.
- Recurse in time. Break the interval in half and perform CCD on first and second half of the time intervals. This way we can resolve continuous collision detections.
- This is necessary in cases when two objects with high velocities collide against each other.

<u>Bounding Volume Hierarchies in detail</u>

Model Hierarchy

- Each node has a simple volume that bounds a set of triangles
- Children contain volumes that each bound a different portion of parent's triangles
- The leaves of hierarchy usually contain individual triangles.

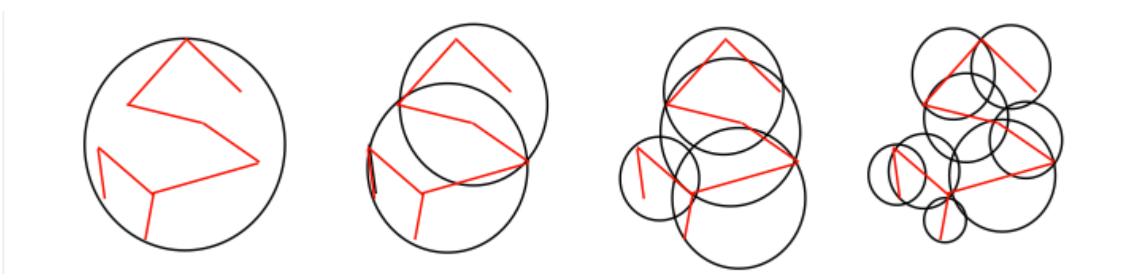
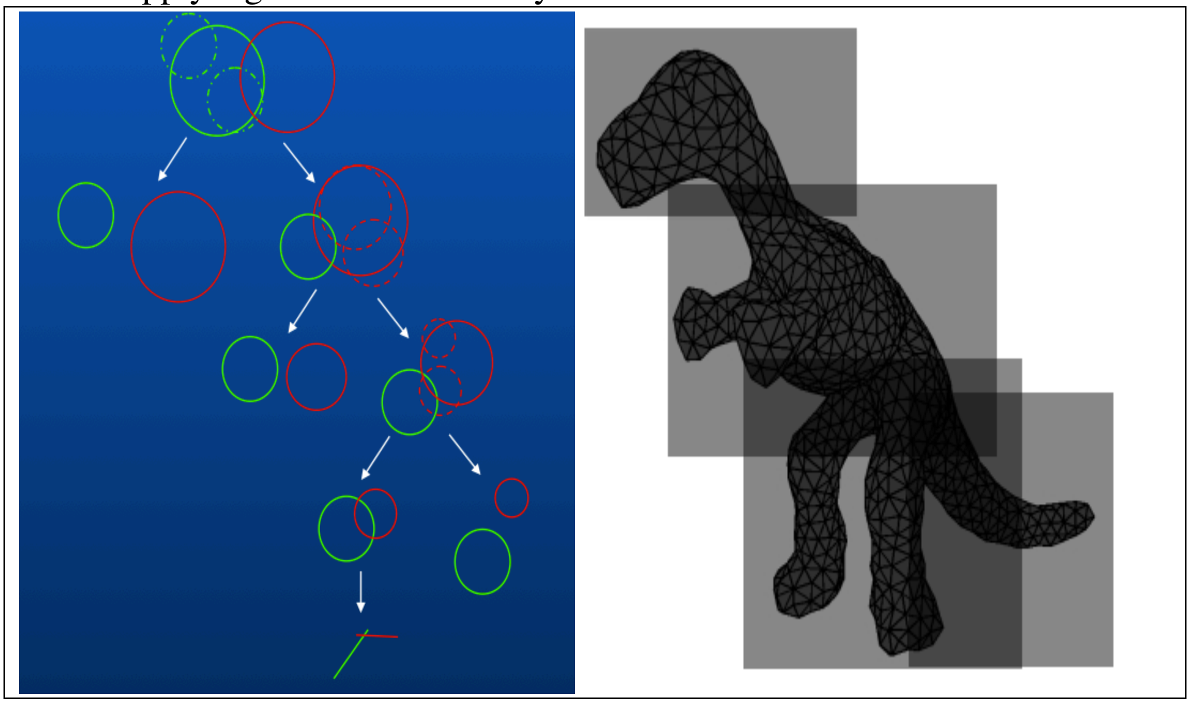


Figure above shows a binary bounding volume hierarchy.

BVH using Dino Mesh

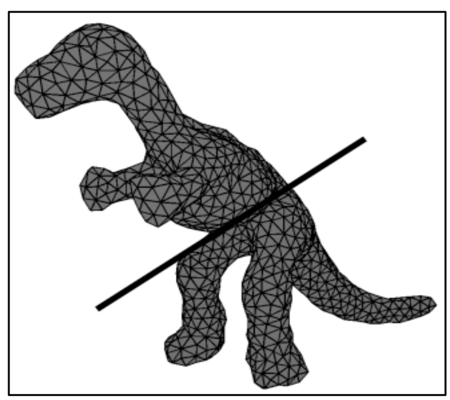
- Small, tight boxes that fit into a triangle are achieved by minimum-maximum.
- For every triangle in the dinosaur, they belong to one of the four boxes.
- All sub regions are of the same size but the 4 (as shown in figure below) boxes need not have same size!
- Assign colors to each bounding box.

- Apply algorithms recursively.



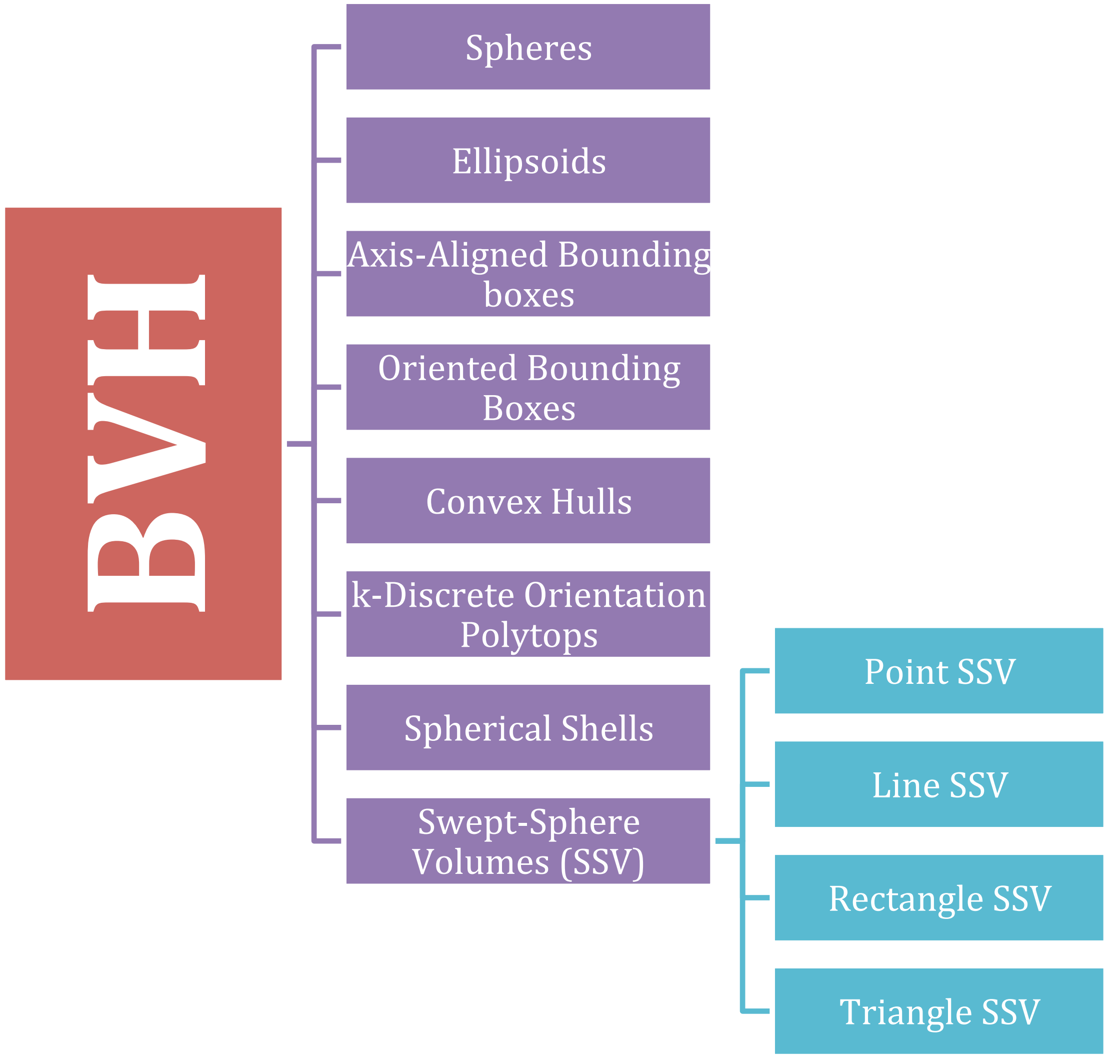
So, how do we break the box into chunks?

- Find center of object - add them up!
- Cut object with a plane that goes through the center point.
- Parameterize all possible planes by giving the unit vector.
- Pick one good cutting plane that separates the objects nicely into two pieces. A good cutting plane in 3d is found by computing variance in the normal direction of the plane in any chosen direction (Found by performing Eigen analysis on the co-variance matrix).



- Compute largest extent (farthest vertex in each direction) along the normal.
- Obtain range between two planes.
- Once we know the direction, if branching factor is tree is two, each side goes into its own individual set.

Types of BVH



General Notes:

- Contrast Axes Aligned Bounding Boxes (Normal to spaces are aligned with coordinated system) with Oriented Bounding Boxes (Box which can be rotated).
- OBB's are better than AABB's as it matches the geometry of model better.
- The disadvantage of OBB's: Difficult to transform box in world space.
- Checking OBB's is more expensive than AABB's.
- Polyhedron – Intersection of finite number of half spaces.
- AABB: Special kind of k-DOPs.

- Convex-hulls: Exotic
- K-DOPs: Powerful.

The equation below explains how ***expensive*** Collision Detection is in terms of CPU Time:

Cost Function:

$$F = N_u \times C_u + N_{bv} \times C_{bv} + N_p \times C_p$$

F: total cost function for interference detection

N_u: no. of bounding volumes updated

C_u: cost of updating a bounding volume,

N_{bv}: no. of bounding volume pair overlap tests

C_{bv}: cost of overlap test between 2 BVs

N_p: no. of primitive pairs tested for interference

C_p: cost of testing 2 primitives for interference

*****