CS 326 A: Motion Planning

robotics.stanford.edu/~latombe/cs326/2004/index.htm

Collision Detection and Distance Computation

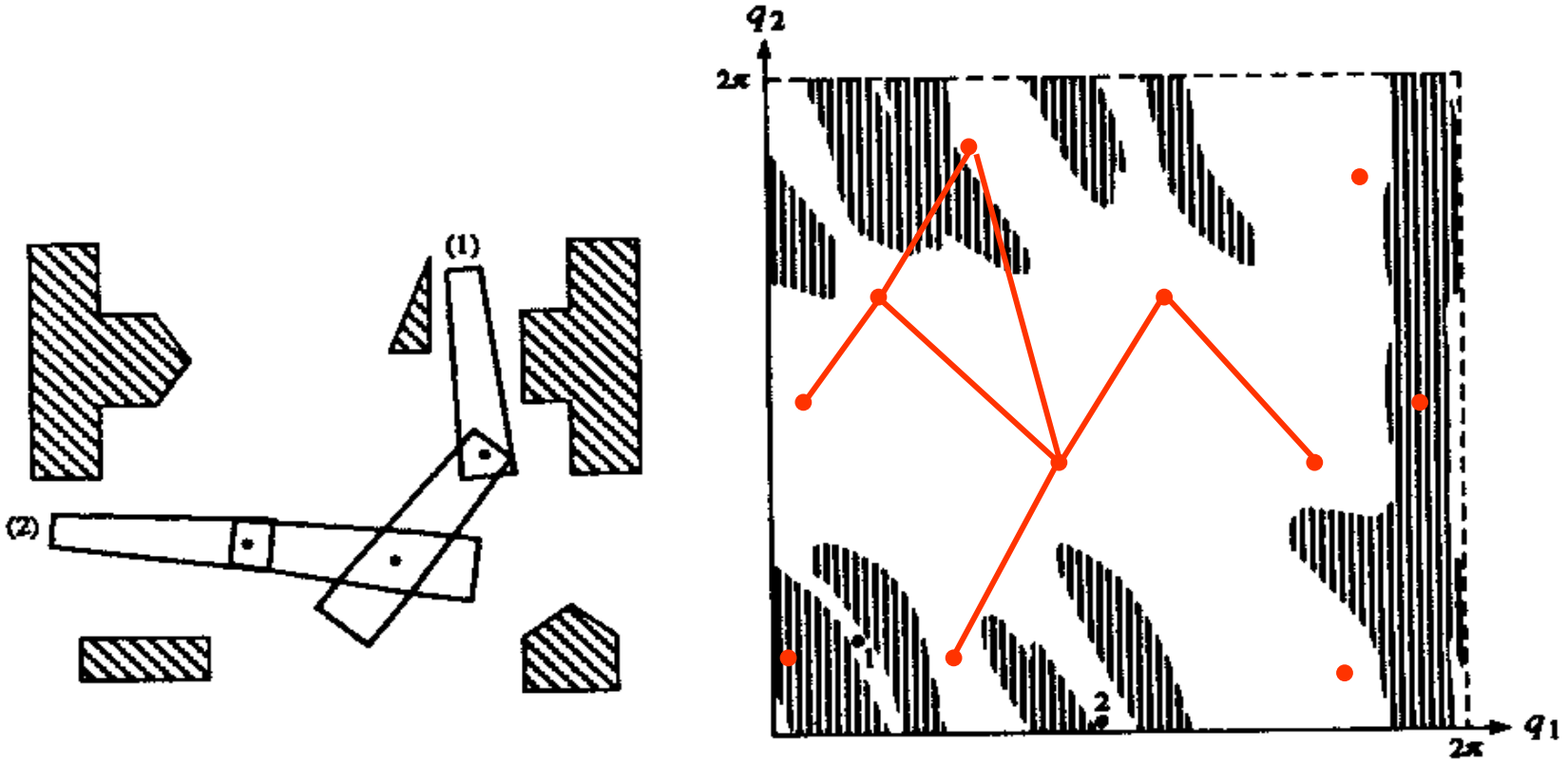
Basic problem

Given two objects A and B, determine whether they collide (overlap), or not

Applications:

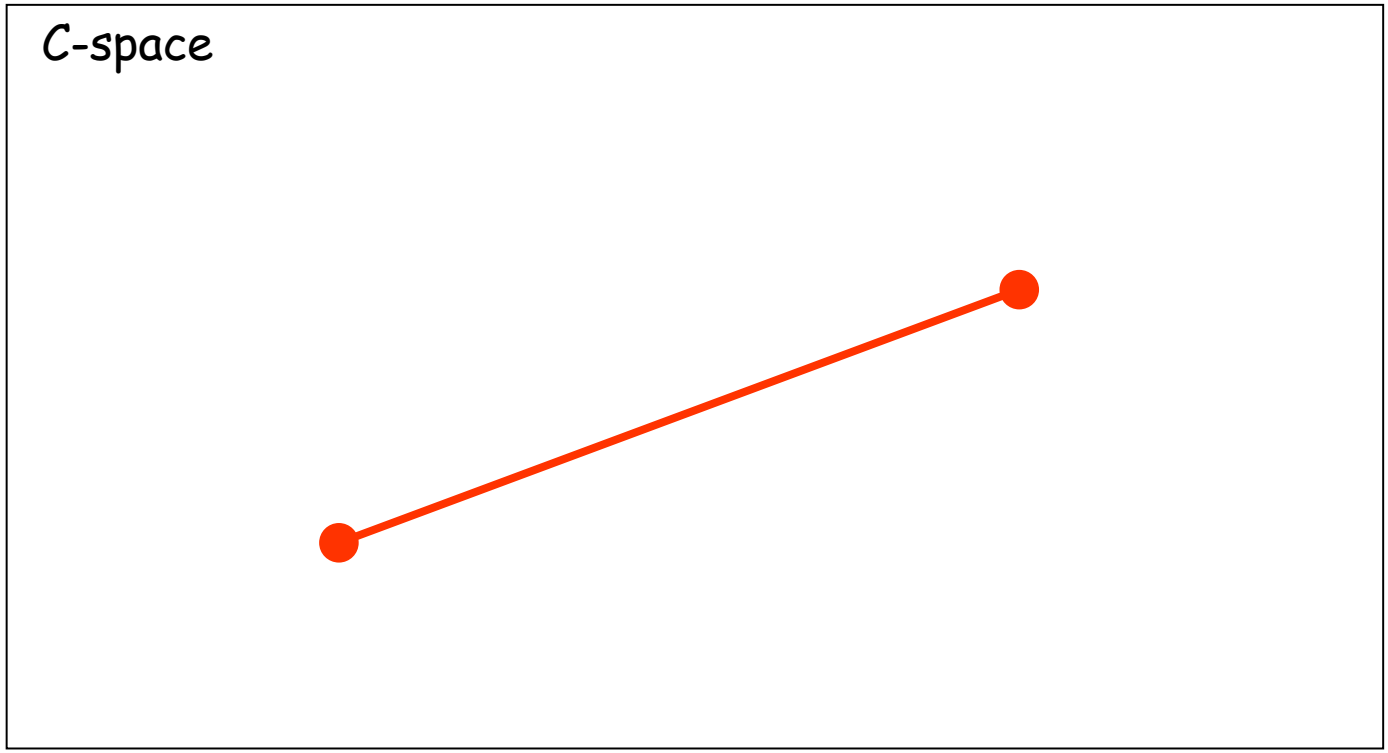
- Computer graphics
- Simulation, e.g., surgical simulation
- Robotics, motion planning

C-Space Sampling

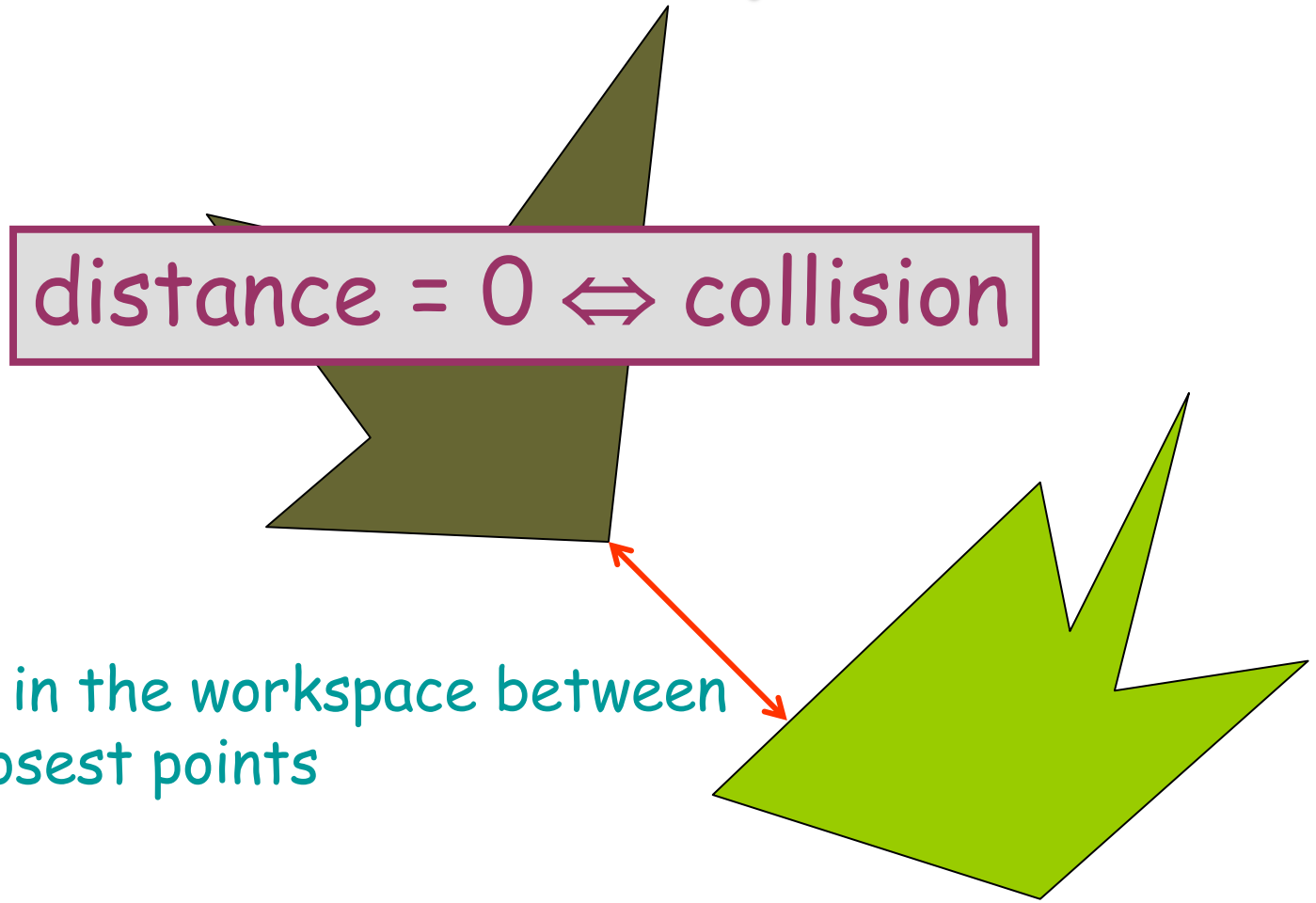


→ Need for efficient collision-checking algorithms

Static vs. Dynamic Collision Checking

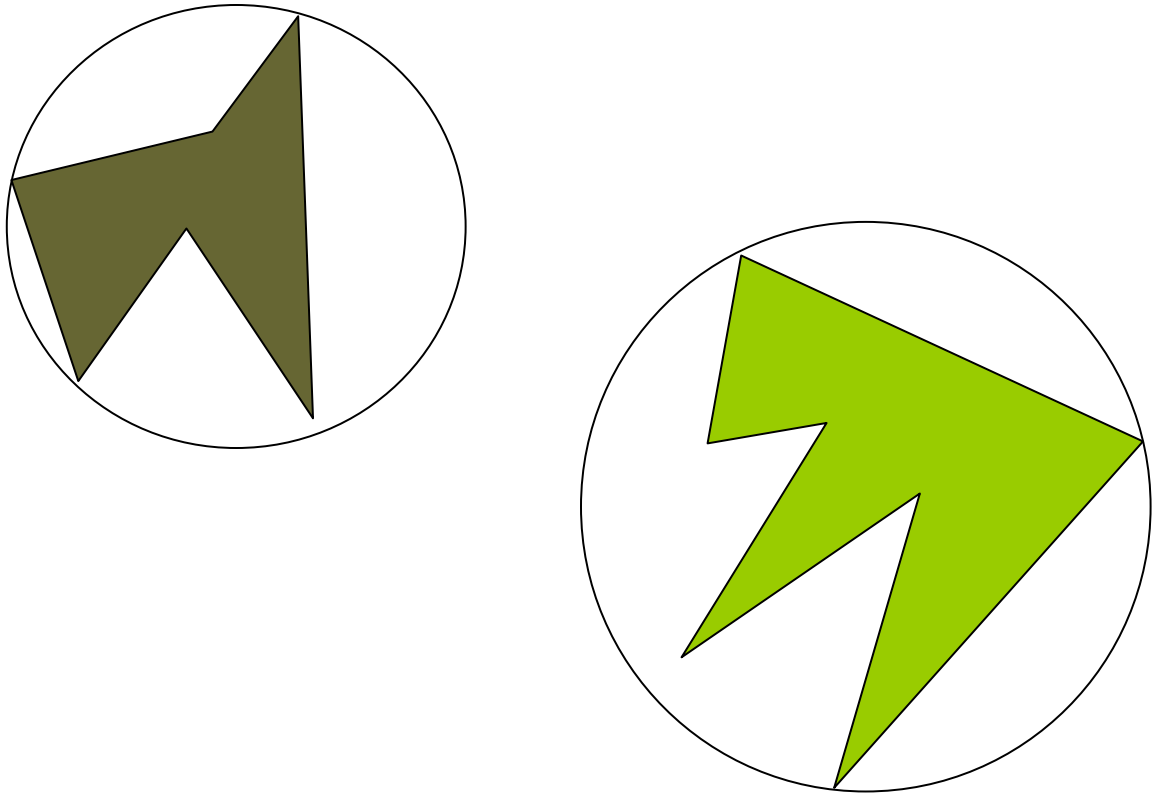


Collision Checking vs. Distance Computation



Distance is in the workspace between the two closest points

It may be easier to check collision than to compute distance



… but (approximate) distance may provide useful additional information

Collision Detection for:

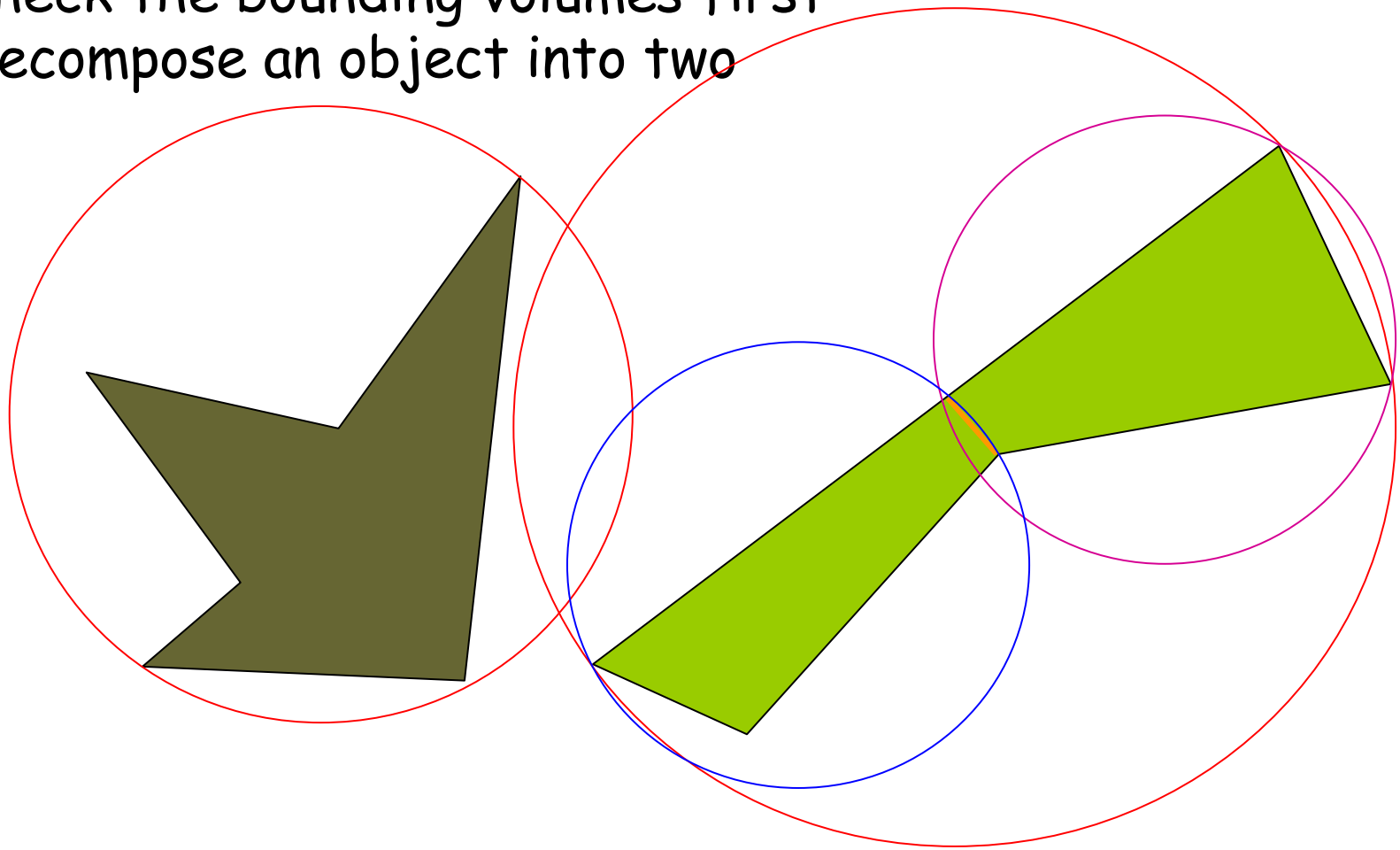
- Two objects:
  - ✓ convex objects
  - ✓ arbitrarily shaped objects
- Collection of objects, e.g., articulated robots + moving obstacles + …
- Deformable objects
- Self-collision

Collision Detection for:

- Two objects:
  - ✓ convex objects
  - ✓ arbitrarily shaped objects
- Collection of objects, e.g., articulated robots + moving obstacles + …
- Deformable objects
- Self-collision

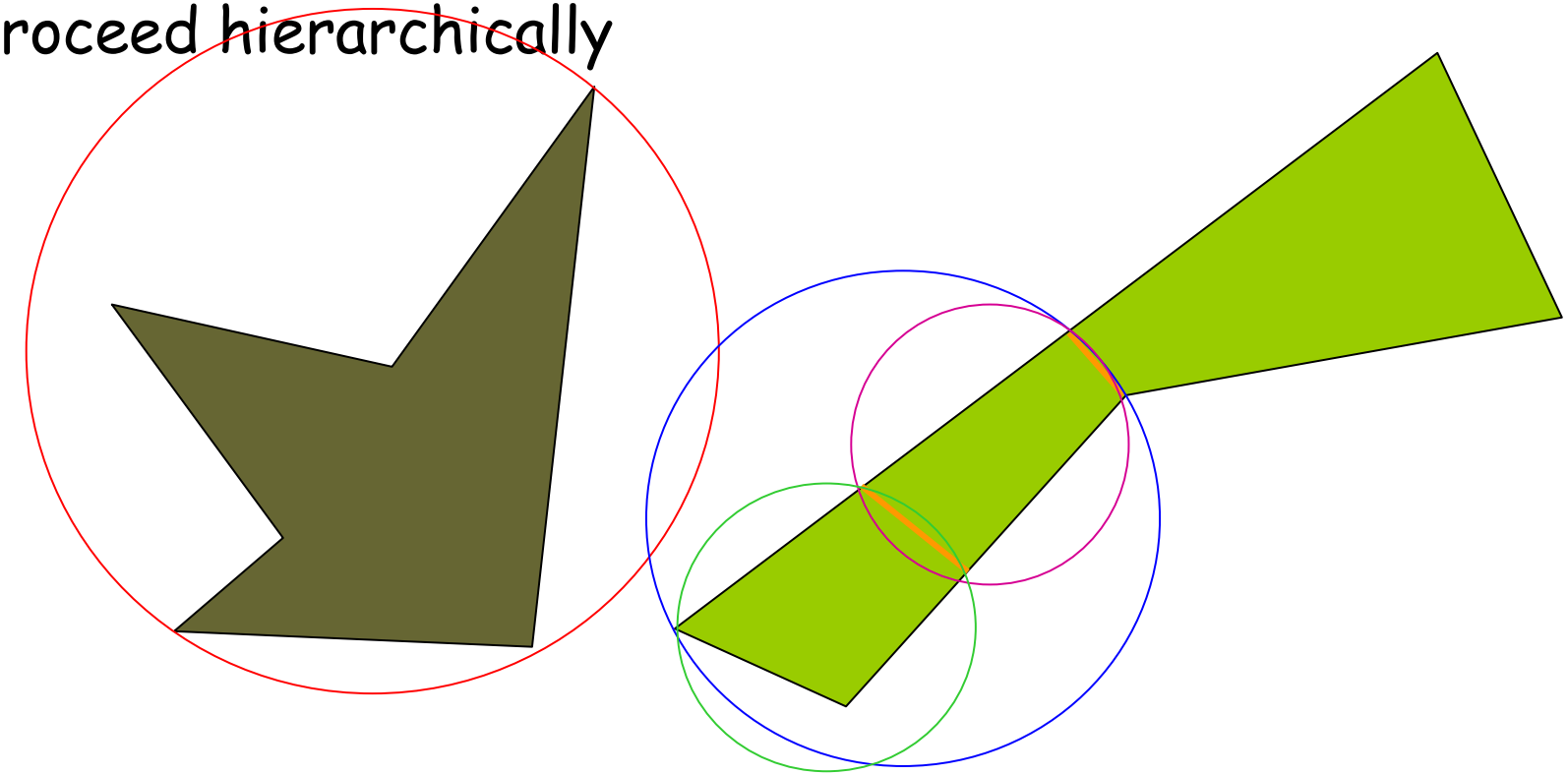
Bounding Volume Hierarchy Method

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two



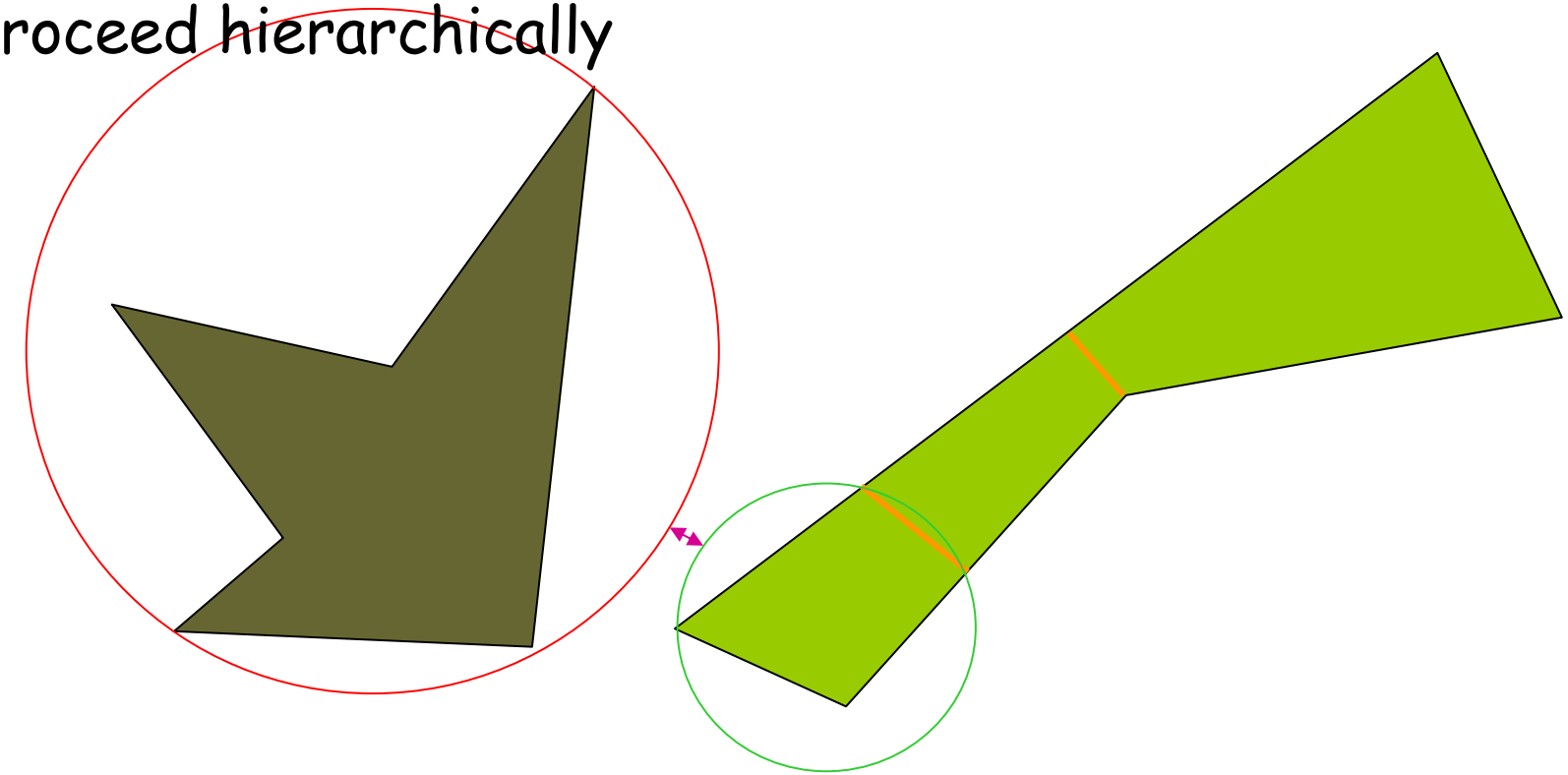
Bounding Volume Hierarchy Method

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two
- Proceed hierarchically



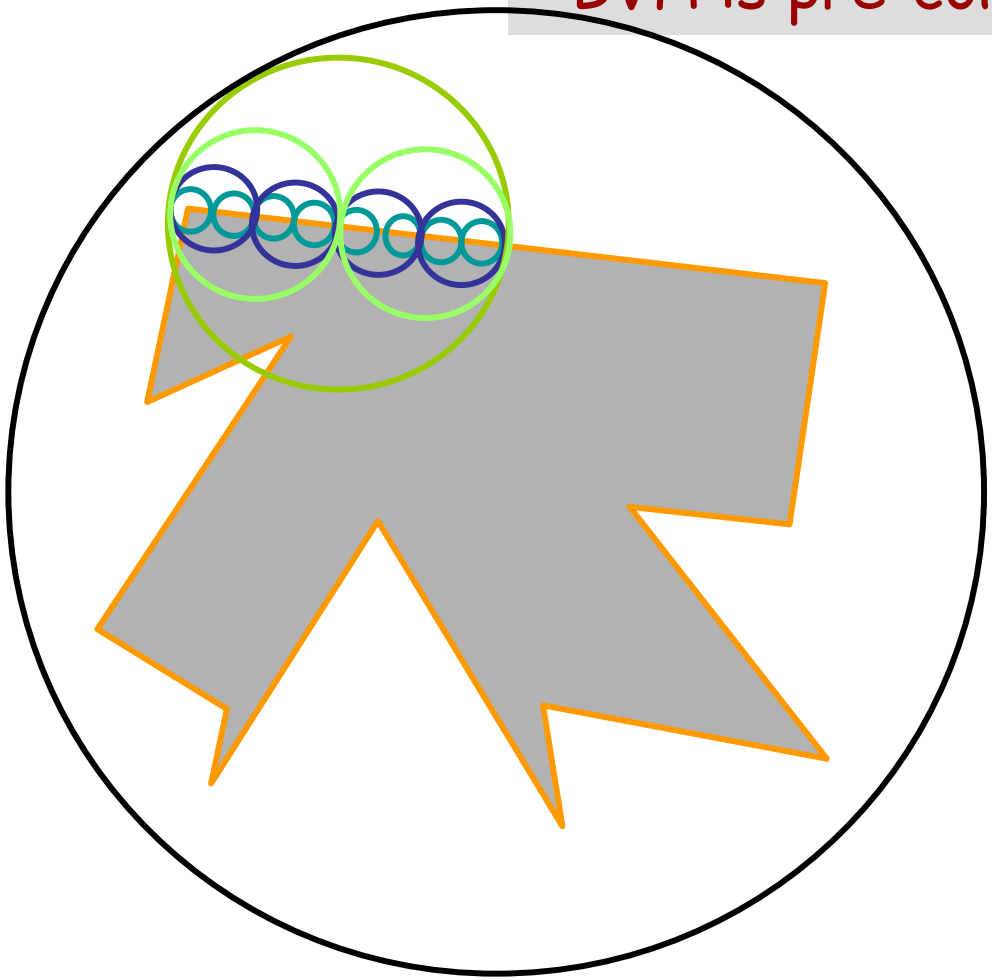
Bounding Volume Hierarchy Method

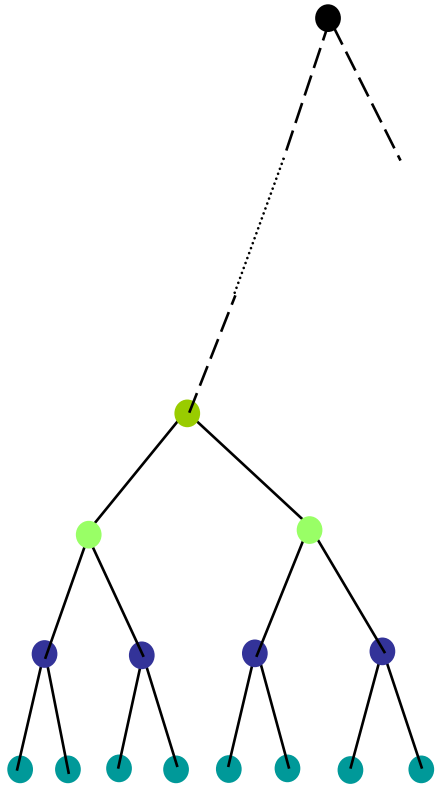
- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two
- Proceed hierarchically



Bounding Volume Hierarchy Method

- BVH is pre-computed for each object





Main Approaches

- Hierarchical bounding volume hierarchies (pre-computation)

- Feature tracking (pairs of closest features)