Solid Modeling

Outline
- Why solid modeling
- Sweep representation
- Boundary representation
- Spatial-partitioning representation
  - Spatial-occupancy enumerations
  - Octrees
  - Binary space partitioning trees
- Constructive solid geometry

Why Solid Modeling
- Triangle mesh & surface patch models may not be watertight
- No information about inside & outside
- Some applications require the knowledge of the volume
- Collision detection
- Weight calculation for a mechanical part
- Ray tracing with refraction

Sweep Representation
- Sweep a shape along a trajectory gives a solid
- Translational sweep:
  - Produce by extrusion
  - Material is pushed or drawn through a die of desired cross-section
- Rotational sweep:
  - Produce by turning
  - Rotate material to cut the desired shape
- Generalized sweep

Boundary Representation (B-Rep)
- Use a collection of connected surface elements to define the boundary between inside & out
- Involved surfaces are closed & oriented 2-manifolds embedded in 3-space
  - A surface is 2-manifolds if a small neighborhood around every point is topologically equal to a disk
  - A manifold surface is oriented if any path on the manifold maintains the orientation of the normal
  - An oriented manifold surface is closed if it partitions 3-space into points inside, on, & outside the surface

Invalid B-Rep
- Non-manifold surfaces:
  - Surface around a point divides the space into more than 2 parts
- Non-oriented manifolds:
  - The 2 sides of the same point are connected by a path on surface
- Examples: Moebius strip & Klein bottle
**Topology Information**

- **Vertex (V):**
  - A unique point in space

- **Edge (E):**
  - A finite, directed, & non-selfintersecting space curve bounded by 2 vertices

- **Loop (L):**
  - An ordered alternating sequence of vertices & edges defining closed & non-selfintersecting space curve

- **Face (F):**
  - A finite, connected, & non-selfintersecting region of a surface bounded by one or more loops

- **Body (B):**
  - A single, connected, & closed volume bounded by a set of faces

- **Genus (G):**
  - A hole or handle.

**Validity Check for B-Rep**

- The Euler's formula states that a polyhedra satisfies:
  - \( V - E + F = 2 \)

- A general solid (may has holes) satisfies:
  - \( V - E + F - H = 2(B - G) \)

  - \( H \) (hole) is the number of inner loops

**Solid cube:**
- \( V=8, E=12, F=6, H=0 \)
- \( B=1, G=0 \)

**Cube w/ dent:**
- \( V=16, E=24, F=11, H=2, B=1, G=0 \)

**Cube w/ hole:**
- \( V=16, E=24, F=10, H=2, B=1, G=1 \)

**Spatial-Partitioning Representation**

- Decompose space into a collection of regions that are labeled as being inside or outside the solid being modeled
  - Different regions are adjoining & nonintersecting
  - A family of approaches exist
    - Spatial-occupancy enumerations
    - Octrees
    - Binary space partitioning trees

**Spatial-Occupancy Enumerations**

- Partition 3D space using uniform grid & refer each cell as a voxel
  - Keep voxels inside the solid object
  - Or store transparency or density information at all voxels

  - Pros & cons:
    - Easy to process
    - Approximate true shape
    - High storage cost

**Octrees**

- Partition 3D space adaptively, instead of uniformly
  - If a cell is completely inside or outside of an object, no need to further split it

  - Pros & cons:
    - Lower storage cost
    - More efficient & concise
    - Harder to process

**Binary Space Partitioning Trees**

- Recursively partition 3D space using arbitrary planes, instead of only orthogonal ones
  - Mark leaf cells as inside or outside

  - Pros & cons:
    - Elegant & precise
    - Representation for an object is not unique
    - Efficiency depends on tree depth
Constructive Solid Geometry (CSG)

- Represent a solid as a result of regularized Boolean operations over primitives
- Output is guaranteed to be a solid if all primitives are solids
- Easy to adjust the final model by changing the transformations or the Boolean operators
- Useful for modeling mechanical parts

Hierarchy for Boolean Operations

- The root is the final complex model
- Internal nodes are Boolean operators or transformations
  - Union, intersection, & difference
  - Translation, rotation...
- Leaf nodes are solid primitives
  - Sphere, cylinder, cone, torus
  - Cube, prism, pyramid

A Union B

If \( a_{in} < a_{out} < b_{in} < b_{out} \)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)

A Intersection B

If \( a_{in} < a_{out} < b_{in} < b_{out} \)
- \(null\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)
- \([b_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([b_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([b_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([b_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([b_{in}, a_{out}] \cap [b_{in}, b_{out}]\)

A Difference B

If \( a_{in} < a_{out} < b_{in} < b_{out} \)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, b_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)
- \([a_{in}, a_{out}] \cap [b_{in}, b_{out}]\)

Comparison of Representations

- Accuracy:
  - Spatial-partitioning approaches produce only approximations for many objects
- Domain:
  - The domain of objects can be represented by sweeps is limited
- Uniqueness:
  - Only octree & spatial-occupancy enumeration guarantee the uniqueness of a representation
- Validity:
  - The validity of B-rep is the most difficult one