Spatial Enumeration models

- solid point set represented by collection of non-overlapping “blocks”
- blocks are “pasted” together to create geometry

Spatial enumeration

- 3D space decomposed into a set of identical cells.
  - Most commonly used geometry for cells is the cube.
- Cells are located by their centers, within a fixed 3-dimensional grid (XYZ space).
- Cells are referred to as volume elements or “voxels”.

Spatial Enumeration Model

Spatial enumeration

- Object geometry is defined by specifying whether a cell within the is empty or full (cell occupancy).
- Provides for simple database:
  - a unique and unambiguous list of occupied cells.

Spatial enumeration

- There exists easy access to positions within the database.
- It is a simple process to determine if a given cell location lies inside or outside of the solid (i.e. membership test)
- Higher model accuracy requires larger number of cells, hence higher data requirement.
Spatial enumeration

- representation used primarily for volume visualization
- simple data structure has made is popular for medical purposes such as
  - CAT scans
  - Magnetic Resonance Imaging (MRI)

Spatial Enumeration Variation:

- Representation scheme that alleviates some of the storage requirements of spatial enumeration.
- Most common is the Octree representation
  - Fundamental idea is that of “divide and conquer”.
  - More easily understood by examining the 2D variant first (quadtrees)

Quadtrees

- Quadtree is a tree data structure of storage used for 2 dimensional cellular decomposition.
- Quadtree is derived by successively sub-dividing a 2D plane in quadrants.
- Each quadrant fully occupied, partially occupied or empty.

Quadtrees

- Graphic representation of quadtree data structure similar to CSG tree except:
  - not binary, but quartic (four branches at node)
  - nodes represent last level of decomposition
  - nodes indicate occupancy level
    - full
    - partial
    - empty

a) 2D spatial enumeration of shape shown.
b) Quadtree representation of the same.
**Quadtree structure from previous object.**

**Quadtree storage example**

**Sample of octree subdivision**

**Octrees**
- Octrees represent the 3 dimensional extension of the quadtree concept.
  - expand quadtree (2D) to 3D space
- Spatial volumes are sub-divided into a set of eight cells or octants.
- Storage tree now has eight branches at each node.

**Octree Representation**
Spatial Enumeration:
Domain and expressive power

- representation is **approximate**
- domain can be considered unlimited providing one excepts the following conditions
  - that it is an approximation and,
  - high storage requirements exist

Validity

- creates valid models if minimal connectivity required
  - each filled cell must have a neighbor
- if connectivity required, validity check is straightforward

Other aspects

- representation is **unambiguous**
- at a **fixed resolution** (i.e. level of decomposition), representation is unique

Octree Format (other uses)

- used in some CAD systems for
  - mass property calculations
  - automated generation of finite element meshes
  - improved display efficiency
  - in each case a secondary model is created in the octree format to simplify calculations

Boundary Representation

- One of the most common solid representation schemes in use.
- Name usually shortened to B-Rep
- Solid “point set” defined by set of surfaces enclosing a volume.

B-Rep model of a component.
(note the connected surfaces)
Boundary Representation

• B-Rep models represent a solid volume by a representation of its bounding surfaces.
  – Solid is a set of surfaces together with topological information which defines the relationships between the surfaces.
• Because B-Rep includes such topological information:
  – a solid is represented as a closed space in 3-D space (surfaces connect without gaps)
  – The boundary of a solid separates points inside from points outside the solid.

Boundary Hierarchy

• A solid object is represented by a collection of faces.
  – Normally a face is a bounded region of a planar, quadratic, toroidal, or sculptured surface.
• The bounded region of the surface that forms the face is represented by a closed curve.
• A single face can have several bounding curves to represent holes in a solid.
• The bounding curves of faces are represented by edges.
  – The edge is delimited represented by two vertices.

B-Rep vs. Surface Modeling

• Surface model:
  – A collection of surface entities which simply enclose a volume lacks the connective data to define a solid (i.e. topology).
• B-Rep model:
  – Technique guarantees that surfaces definitively divide model space into solid and void, even after model modification commands.

Boundary Representation

• System must validate topology of created solids.
  • A B-Rep representation has to fulfill certain conditions to disallow self-intersecting and open objects.
  – These conditions would include:
    • Each edge should adjoin exactly two faces and have a vertex at each end.
    • Vertices are geometrically described by point coordinates
  (Continued next slide)

Validity of a B-Rep model:

• Validity also checked through mathematical evaluation.
  • Modeler performs an inventory of all model elements at each operation. (vertices, edges, surfaces, etc.)
  • An evaluation involving simple integer math is then performed upon that inventory
  – Evaluation is based upon Euler’s Law

Boundary Representation

• At least three edges must meet at each vertex.
• Faces are described by surface equations
• The set of faces forms a complete skin of the solid with no missing parts.
• Each face is bordered by an ordered set of edges forming a closed loop.
• Faces must only intersect at common edges or vertices.
  – Each edge has two neighboring faces (loops)
  – Within these two loops, the edge is contained in opposite directions
• Faces must not interpenetrate.
  – The boundaries of faces do not intersect themselves.
Euler’s Law:

- Defines an invariant relationship among the vertices, edges, and face loops of a polyhedron.
- Valid for simple polyhedra (continuous, no hole)
- \( V - E + F = 2 \)

\( V \) (# vertices) \( E \) (# edges) \( F \) (# face loops)

Example of Euler evaluation

This is known as the Euler-Poincare Law:

- \( V - E + F - L = 2(S - G) \)
- \( L \) = # of inner face loops (a loop contained entirely within another face loop)
- \( S \) = # of shell bodies (sometimes “C”)
- \( G \) = # thru holes, A.K.A. genus ( # of passage features)

Euler-Poincare example

\[ V - E + F - L = 2(S - G) \]
\[ 24 - 36 + 15 - 3 = 2(1 - 1) \]

Euler Operators

- Model is topologically valid if satisfies Euler-Poincare
- Validity relationship allows for definition of a set of operators:
  - known as Euler Operators
  - allow faces, edges and vertices to be added and removed from model while retaining validity
Sample Euler Operators

<table>
<thead>
<tr>
<th>Operation</th>
<th>Coding</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvfs</td>
<td>1 0 1 0 1 0</td>
<td>make vertex, face, body</td>
</tr>
<tr>
<td>mev</td>
<td>1 1 0 0 0 0</td>
<td>make edge, vertex</td>
</tr>
<tr>
<td>mef</td>
<td>0 1 1 0 0 0</td>
<td>make edge, face</td>
</tr>
<tr>
<td>kemr</td>
<td>0-1 0 0 0 1</td>
<td>kill edge, make ring</td>
</tr>
<tr>
<td>kev</td>
<td>-1-1 0 0 0 0</td>
<td>kill edge, vertex</td>
</tr>
<tr>
<td>kef</td>
<td>0-1-1 0 0 0</td>
<td>kill edge, face</td>
</tr>
<tr>
<td>mekr</td>
<td>0 1 0 0 0-1</td>
<td>make edge, kill ring</td>
</tr>
</tbody>
</table>

Sample Euler Operation Build

**MBV** (Make body, vertex)

**MEV** (Make edge, vertex)

**MEF** (Make edge, face)

Euler Operators

- Operators
  - Are complex
  - Operate at a low level (computationally)
  - Difficult for designers to use
- Euler operators usually not available to user at interface, only exist internal to software

B-Rep Construction methods

- B-Rep systems use various techniques for model construction.
  - **Sweeping** and **lofting** operations are the most common of these techniques.
  - The resultant volumes from these operations may be joined or removed from one another to create a desired shape.
Validity
- Can be difficult to ensure while still permitting user design freedom
- B-Rep modelers more prone to validity failure than CSG
- Validity dependent upon individual package's algorithms

Ambiguity and Uniqueness
- valid B-Reps are unambiguous
- not fully unique, but much more so than CSG
- potential difference exists in division of
  - surfaces into faces
  - curves into edges

B-Rep conciseness
- Within the database, each topological component linked with its geometric counterpart.
- Exact mathematical equations are stored with associated surfaces and curves.

B-Rep conciseness
- Data structure
  - highly explicit and evaluated
  - this makes the database very large and lacking in conciseness.
- Some B-Rep modelers:
  - model display limited to planar faces and linear edges
    - complex curves and surfaces only approximated
    - display is a faceted model (polyhedral)