Surface Representations
Volumetric Models

Implicit surfaces
- Set of zeros of a function
  \( \{(x,y,z) : f(x,y,z) = 0\} \)
- Good for boolean operations (CSG)
- Difficult to render (ray-tracing)
- Iso-surface
  - Function defined by piecewise function
  - Volumetric mesh
  - 1 function value per vertex
- Iso-surface algorithm
  - Conversion to triangle or polygon mesh representation

Implicit surfaces
- Can be used to represent the probability that a point belongs to a surface
  - Occupancy grid
- Can be used to integrate multiple measurements
- Can be used to merge multiple 3D scans

Implicit Linear Surfaces / Curves

Affine bases / Linear function
\[ p = \lambda_0 p_0 + \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3 \]
\[ \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^{-1} \begin{bmatrix} p \\ 1 \end{bmatrix} \]
\[ f(p) = \lambda_0 f(p_0) + \lambda_1 f(p_1) + \lambda_2 f(p_2) + \lambda_3 f(p_3) \]

Piecewise Linear Functions
- Triangle : Barycentric coordinates
  - Triangle / Tetrahedron / Simplex
- Every point in 3D can be written as a unique affine combination of 4 non-coplanar points (affine basis)
- Every linear function in 3D can be specified by its values at the 4 vertices of an affine basis
- A piecewise-linear function is specified in 3D by its values at the vertices of a tetrahedral mesh (volumetric).
Iso-surfaces on tetrahedral meshes
• Piecewise linear function defined on vertices of tetrahedral mesh \( f(i) \)
• For each edge \((i,j)\) such that \( f(i)f(j)<0 \)
  - create a surface vertex \( v(i,j) \)
• For each tetrahedron \((i,j,k,l)\)
  - Skip if all vertices are positive or negative
  - Else if 3 positive or 3 negative create a triangle
  - Else (if 2 positive and 2 negative) create two triangles
• Output triangle mesh is IndexedFaceSet
• Is it a manifold mesh? Why?

Iso-surfaces on hexahedral meshes
• Function defined on vertices of regular grid
• For each edge \((i,j)\) such that \( f(i)f(j)>0 \)
  - create a surface vertex \( v(i,j) \)
• For each intersecting cube
  - Polygonize intersection
• Output triangle mesh is IndexedFaceSet
• Is it a manifold mesh? Why?
• Main problem: storage
  • Solution: do not represent the mesh explicitly
Interpolation

- Linear interpolation
- Triangle: Barycentric coordinates
  - Triangle
  - Tetrahedron
- Quadrilateral?
  - Bi-linear interpolation
- Cube?
  - Tri-linear interpolation

Marching Cubes Algorithm

1. User specifies threshold value
2. Read four slices into memory
3. Scan middle two slices and create a cell
4. Classify eight vertices, construct index number
5. Use index to look up list of edges
6. Find 3 surf/edge intersections via linear interpolation
7. Calculate unit normal (gradient) at 3 intersections
8. Output the triangle vertices and vertex normals
Extensions

- Iso-surface algorithm assumes smooth surface without singularities
- How to represent ridges?
- Iso-surface algorithm produces regular face sizes even in regions where fewer faces would produce equally good approximation
- Adaptive iso-surfaces?