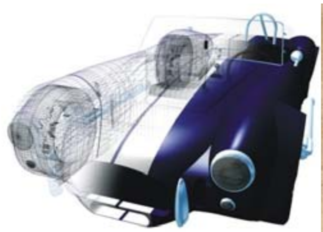
Chapter 14

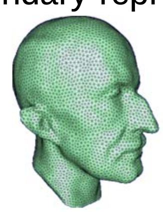
Geometric Modeling - Basics



Geometry

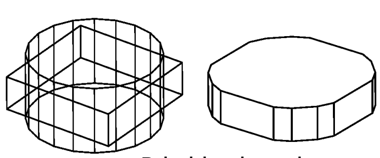
- Mathematical models of real world shapes
  - Most common: Boundary representations

Freeform – smooth surface

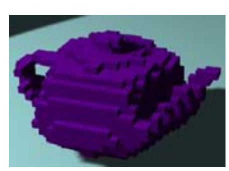


Mesh – polygonal surface

  - Alternative: Volumetric representations



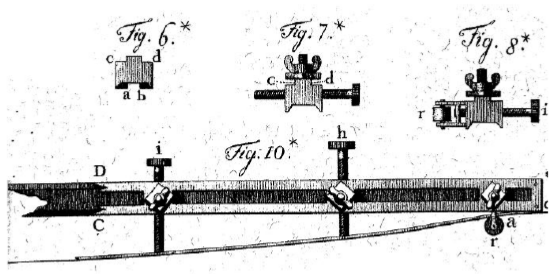
Primitive based



Voxel based

Splines – Free Form Curves

- Geometric meaning of coefficients (base)
  - Approximate/interpolate set of positions, derivatives, etc..



- Will see one example

Splines – Free Form Curves

- Usually parametric
  - $C(t)=[x(t),y(t)]$ or $C(t)=[x(t),y(t),z(t)]$
- Description = basis functions + coefficients

$$C(t)=\sum_{i=0}^{n} P_i B_i(t) = (x(t), y(t))$$

$$x(t)=\sum_{i=0}^{n} P_i^x B_i(t)$$

$$y(t)=\sum_{i=0}^{n} P_i^y B_i(t)$$

  - Same basis functions for all coordinates

Hermite Cubic Basis

- Geometrically-oriented coefficients
  - 2 positions + 2 tangents
- Require $C(0)=P_0$, $C(1) = P_1$, $C'(0)=T_0$, $C'(1)=T_1$
- Define basis function per requirement

$$C(t) = P_0 h_{00}(t) + P_1 h_{01}(t) + T_0 h_{10}(t) + T_1 h_{11}(t)$$

Hermite Basis Functions

$$C(t) = P_0 h_{00}(t) + P_1 h_{01}(t) + T_0 h_{10}(t) + T_1 h_{11}(t)$$

- To enforce $C(0)=P_0$, $C(1) = P_1$, $C'(0)=T_0$, $C'(1)=T_1$ basis should satisfy

$$h_{ij}(t): i, j = 0,1, t \in [0,1]$$

curve	$C(0)$	$C(1)$	$C'(0)$	$C'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

Hermite Cubic Basis

- Can satisfy with cubic polynomials as basis

$$h_{ij}(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

- Obtain - solve 4 linear equations in 4 unknowns for each basis function

$$h_{ij}(t): i, j = 0,1, t \in [0,1]$$

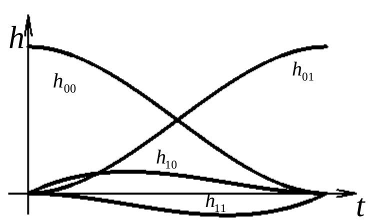
curve	$C(0)$	$C(1)$	$C'(0)$	$C'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

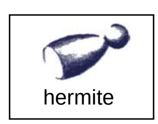
Hermite Cubic Basis

- Four polynomials that satisfy the conditions

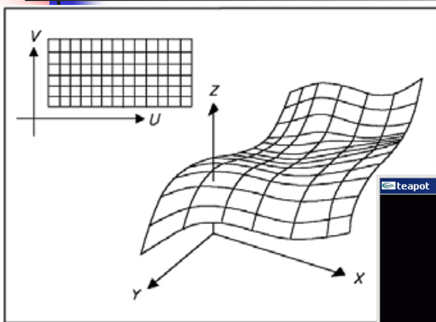
$$h_{00}(t) = t^2(2t-3)+1 \qquad h_{01}(t) = -t^2(2t-3)$$

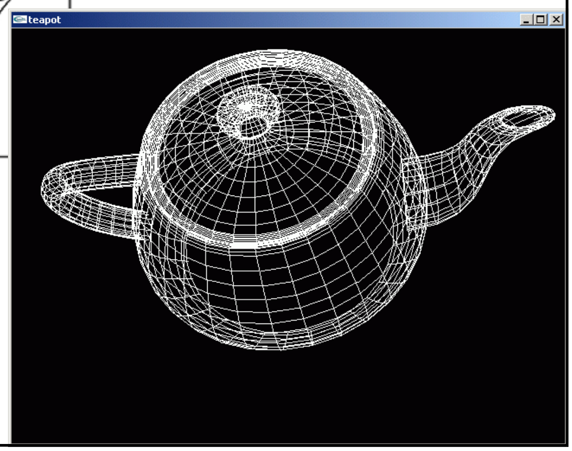
$$h_{10}(t) = t(t-1)^2 \qquad h_{11}(t) = t^2(t-1)$$





Tensor Spline Surfaces





From Curves to Surfaces: Tensor Splines

- Curve is expressed as inner product of P_i coefficients and basis functions

$$C(u) = \sum_{i=0}^{n} P_i B_i(u)$$

- To extend curves to surfaces - treat surface as a curve of curves
- Assume P_i is not constant, but a function of second parameter v: $P_i(v) = \sum_{j=0}^{m} Q_{ij} B_j(v)$

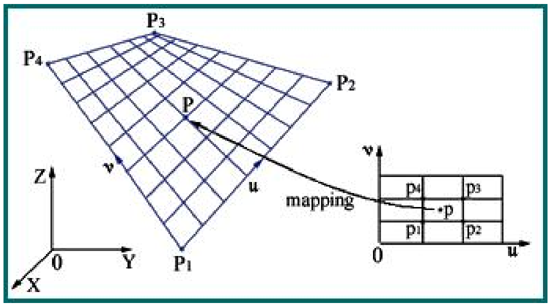
$$\boxed{C(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} Q_{ij} B_j(v) B_i(u)}$$

Bilinear Patches

- Bilinear interpolation of 4 3D points

$$P_{00}, P_{01}, P_{10}, P_{11}$$

- surface analog of line segment curve



Bilinear Patches

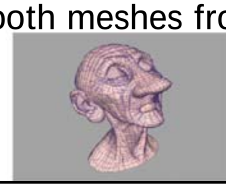
- Given $P_{00}, P_{01}, P_{10}, P_{11}$ associated parametric bilinear surface for $u, v \in [0,1]$ is:

$$P(u,v) = (1-u)(1-v)P_{00} + (1-u)vP_{01} + u(1-v)P_{10} + uvP_{11}$$

- Questions:
  - What does an isoparametric curve of a bilinear patch look like ?
  - When is a bilinear patch planar?

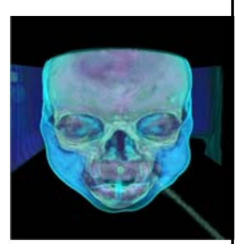
Geometry Creation (Meshes)

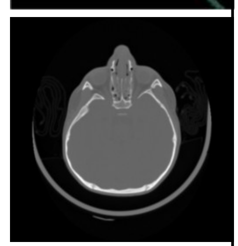
- Reconstruction: Capture real life shapes & convert to mesh
  - Inputs:
    - Points (laser scanner)
    - 3D images
- Modeling (user driven)
- Will see two examples
  - Marching Cubes – reconstruction from images
  - Subdivision – generating smooth meshes from coarse user-given “cages”



Reconstruction from Volume Data

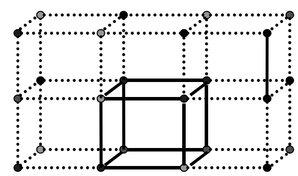
- Volume data – view as voxel grid with values at vertices
  - Defines implicit function in 3D
    - interpolate grid values
- Shape defined by isosurface
  - isosurface = set of points with constant isovalue α
  - separates values above α from values below
- Reconstruction – Extract triangulation approximating isosurface





Voxels

- Voxel – cube with values at eight corners
  - Each value is above or below isovalue α
- $2^8=256$ possible configurations (per voxel)
  - reduced to 15 (symmetry and rotations)
- Each voxel is either:
  - Entirely inside isosurface
  - Entirely outside isosurface
  - Intersected by isosurface



- MC main observation: Can extract triangulation independently per voxel

Basic MC Algorithm

- For each voxel produce set of triangles
  - Based on above/below corner configuration
  - Empty for non-intersecting voxels
  - Approximate surface inside voxel

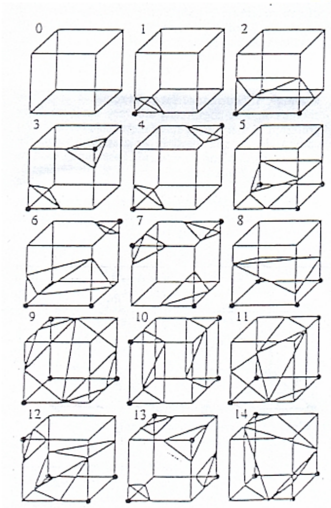
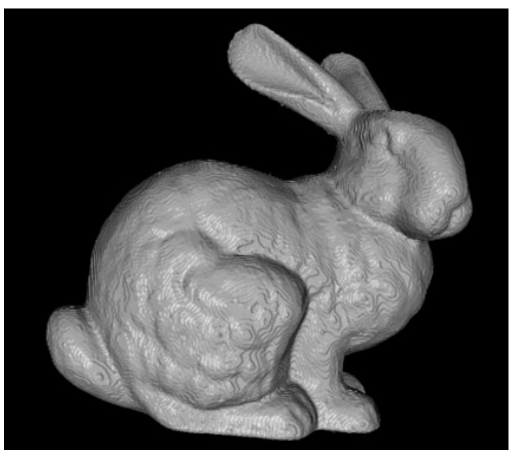


Figure 2. Configurations.

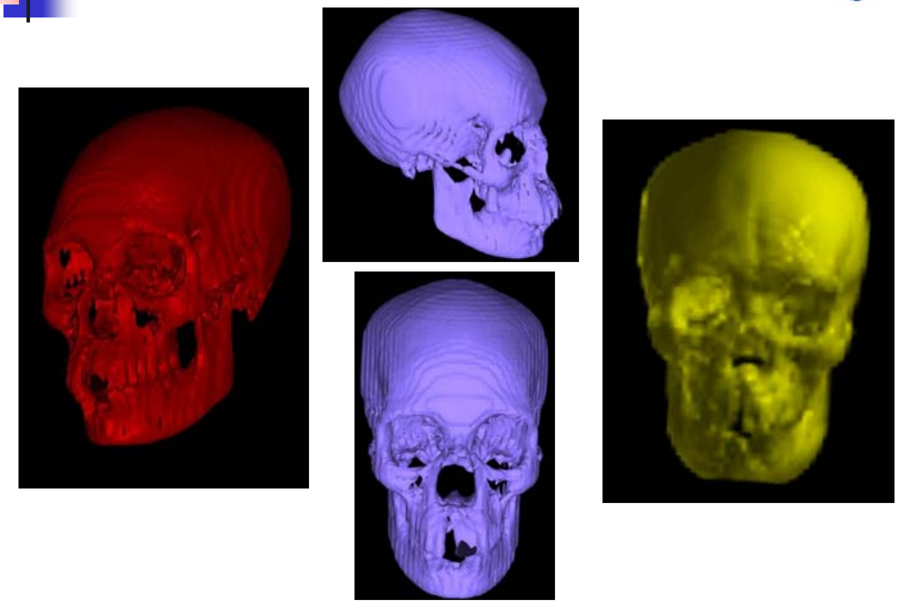
Configurations

- For each configuration add 1-4 triangles to isosurface
- Isosurface vertices computed by:
  - Interpolation along edges (according to grid values)
    - better shading, smoother surfaces
  - Default – mid-edges

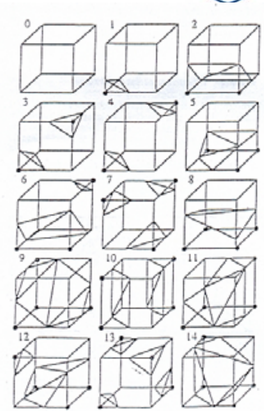
Example



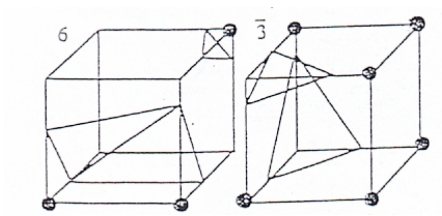
Example



Consistency Problem



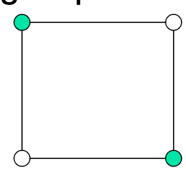
- Can produce non-manifold
  - Isovalue surfaces with “holes”
- Example:



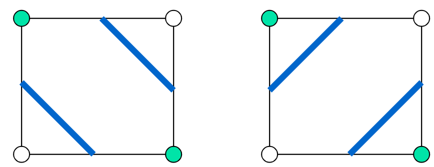
  - Voxel with configuration 6 sharing face with complement of configuration 3

Ambiguous Faces

- Face containing two diagonally opposite marked grid points and two unmarked ones



- Two locally valid interpretations



- Source of MC consistency problem

Consistency

- Problem:
  - Connection of isosurface points on shared face done one way on one face & another way on the other
- Need consistency → use different triangulations
- If choices are consistent get topologically correct surface

Solution

- For each problematic configuration have more than one triangulation
- Distinguish different cases by choosing pairwise connections of four vertices on common face
  - Example:

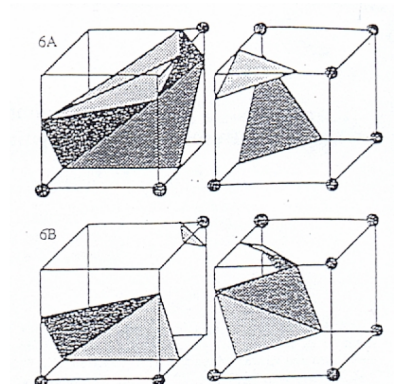
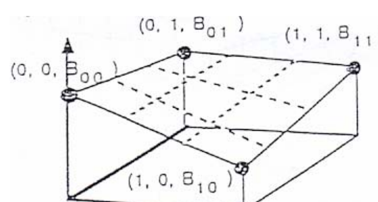


Figure 4. Two possible triangulations which yield a topologically correct isovalue surface.

2.0 Asymptotic Decider

Asymptotic Decider

- "Guess" value at quad center
- Use bilinear interpolation to obtain



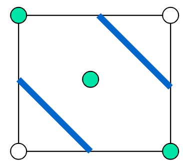
$$B(s,t) = \begin{pmatrix} 1-s & s \end{pmatrix} \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix} \begin{pmatrix} 1-t \\ t \end{pmatrix}$$

$$\{(s,t): 0 \le s \le 1, \quad 0 \le t \le 1\}$$

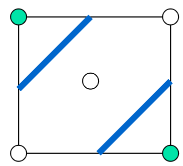
- B_{ij} - isovalues at face corners

Ambiguous Faces

- If center value closer to "green" choose



- Else



Various Cases

- Some configurations have no ambiguous faces → no modifications
- Other configurations need modifications according to number of ambiguous faces
  - Apply decoder to each face to decide on triangulation template

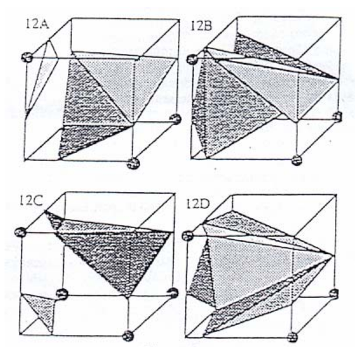


Figure 11.