Lecture 12: Modelling Surfaces

Surface Modelling
- Implicit Surfaces
  \( F(x, y, z) = 0 \)
- Parametric Surfaces
  \( P(u, v) = (x(u, v), y(u, v), z(u, v)) \)

Parameter Space
Surface in Euclidean Space

Parametric Surfaces
- Surfaces of Revolution
  - Obtained by rotating a 2D generating curve around an axis

Parametric Surfaces
- Surfaces of Revolution
  - A Sphere can be generated by rotating a semi circle around an axis
    - Sphere
      \( Q(\theta, \phi) = (x, y \cos \phi, y \sin \phi) \)
      \( = (r \cos \theta, r \sin \theta \cos \phi, r \sin \theta \sin \phi) \)
      - Similarly we can generate
        - Ellipsoides
        - Paraboloides
        - Cones

Parametric Surfaces
- Surfaces of Revolution
  - A Torus is generated by rotating a circle in the XY plane around an axis, but whose center is not on the axis.
    - Torus
      \( Q(\theta, \phi) = (h + r \cos \phi, k + r \sin \phi, (h + r \cos \phi) \cos \theta, (h + r \cos \phi) \sin \theta) \)

Point
Circle
Line
Cylinder

Circle
Torus
Semi Circle
Sphere
Cone

Cylinder
Torus
Semi Circle
Parametric Surfaces

- Surfaces of Revolution
  - Matrix form for the rotation about X axis
    $$\begin{bmatrix}
      1 & 0 & 0 & 0 \\
      0 & \cos \phi & 0 & 0 \\
      0 & 0 & \sin \phi & 0 \\
      0 & 0 & 0 & 1
    \end{bmatrix}$$
  - Parametric curve
    $$P(t) = \begin{pmatrix} x \cos \phi, y \sin \phi \end{pmatrix}$$
  - Parametric Surface
    $$\mathbf{Q}(t, \phi) = S \mathbf{P}$$

- Sweep Surfaces
  - Sweep surfaces are generated by moving a 2D curve along a path
    $$\mathbf{Q}(t, s) = T(s) \mathbf{P}(t)$$
  - $T(s)$ is called the sweep transformation.
  - A translation sweep of a circle generates a cylinder.
  - A translation sweep of a circle with a scaling generates a cone.

- Normal to the polygon or closed curve can be kept fixed or it can be made the instantaneous tangent of the curve of sweep.

- Bilinear Surfaces
  - Linear interpolation fits the simplest curve between two points
    $$P(t) = (1-t)P_1 + tP_2$$
  - Bilinear interpolation fits the simplest surface to four corner points

- Bilinear Surfaces
  - Linear interpolation fits the simplest surface to four corner points
Parametric Surfaces

- **Bilinear Surfaces**
  - Bilinear interpolation fits the simplest surface to four corner points

\[
\begin{align*}
\beta^0_0(v) &= (1-v)\beta_{00} + v \beta_{01} \\
\beta^1_0(v) &= (1-v)\beta_{10} + v \beta_{11} \\
X(u, v) &= (1-u)\beta_{00} + u \beta_{01} + v \beta_{10} + (1-v)\beta_{11}
\end{align*}
\]

\[
X(u, v) = \sum_{i=0}^{1} \sum_{j=0}^{1} b_{ij} B_i^1(u) B_j^1(v)
\]

Parametric Surfaces

- **Ruled Surface**
  - Given two space curves \(C_1\) and \(C_2\) defined in the parametric range \([0, 1]\).
  - Find a surface \(X\) that contains both the curves as boundary curves.

\[
\begin{align*}
X(u, 0) &= C_1(u) \\
X(u, 1) &= C_2(u)
\end{align*}
\]

For a constant \(u\), the iso-parametric curve is a straight line.

Parametric Surfaces

- **Coon's Patch**
  - Given for boundary curves \(C_1, C_2, D_1, D_2\)

\[
\begin{align*}
X(u, 0) &= C_1(u) \\
X(u, 1) &= C_2(u) \\
X(0, v) &= D_1(v) \\
X(1, v) &= D_2(v)
\end{align*}
\]

Parametric Surfaces

- **Ruled Surface**
  - We do linear interpolation in \(v\)

\[
\begin{align*}
X(u, v) &= (1-v)C_1(u) + vC_2(u) \\
X(u, 0) &= C_1(u) \\
X(u, 1) &= C_2(u)
\end{align*}
\]

For a constant \(u\), the iso-parametric curve is a straight line.
Parametric Surfaces

But something is extra here and that is?

RCD

The bilinear patch between the vertices!

Coon’s Patch = RCD

The bilinear patch between the vertices!

Coon’s Patch

\[ Q(u, v) = R_x(u, v) + R_y(u, v) - R_{CD}(u, v) \]

\[ X(u, v) = R_x(u, v) + R_y(u, v) - R_{CD}(u, v) \]

\[ \begin{bmatrix} u \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \]

● Coon’s Bicubic Surface
  - Each boundary curve is a normalized cubic spline
  - Blending function used is also cubic

\[ P_i(u) = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \]

\[ F_i(v) = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ F_{21} & F_{22} & F_{23} & F_{24} \\ F_{31} & F_{32} & F_{33} & F_{34} \\ F_{41} & F_{42} & F_{43} & F_{44} \end{bmatrix} \]

\[ \begin{bmatrix} Q_{00} & Q_{01} & Q_{02} & Q_{03} \\ Q_{10} & Q_{11} & Q_{12} & Q_{13} \\ Q_{20} & Q_{21} & Q_{22} & Q_{23} \\ Q_{30} & Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \]

\[ F_{ij}(v) = \frac{v}{(1 - v)} \begin{bmatrix} Q_{00} & Q_{01} & Q_{02} & Q_{03} \\ Q_{10} & Q_{11} & Q_{12} & Q_{13} \\ Q_{20} & Q_{21} & Q_{22} & Q_{23} \\ Q_{30} & Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \]

● Bézier Surface
  - Given the control points, find the surface
Parametric Surfaces

- **Bézier Curves Recap**
  - Mathematically: \( P(t) = \sum b_i B_n^i(x) \) with \( 0 \leq t \leq 1 \)
  - Where \( B_n^i(x) = \binom{n}{i} x^i (1-x)^{n-i} \) and is called the Bernstein basis.

- **Bilinear Surfaces Recap**
  - Bilinear interpolation fits the simplest surface to four corner points

- **Bézier Surfaces – The De Casteljau Algorithm**
  - \( B_n^1 = (1-t)b_1 + tb_2 \)
  - \( B_n^0 = (1-t)b_0 + tb_1 \)
  - \( P(x) = B_n^1 + db_2^0 \)
  - \( P(x) = B_n^0 + db_3^0 \)
  - \( X(u, v) = (1-u) B_n^0 + u B_n^1 \)
  - \( X(u, v) = (1-v) B_m^0 + v B_m^1 \)
  - \( X = (1 - u) \begin{bmatrix} B_n^0 & B_n^1 \end{bmatrix} (1 - v) \begin{bmatrix} B_m^0 & B_m^1 \end{bmatrix} \)
  - \( X = \sum_{i=0}^{n} \sum_{j=0}^{m} B_n^i B_m^j (u)(v) \)
Parametric Surfaces

- Bézier Surfaces – The De Casteljau Algorithm

\[ B_{i,j}^{m,n}(u,v) = \sum_{r=0}^{m+n} \binom{m+n}{r} B_i^{m}(u) B_j^{n}(v) \]

What if degree of the Bézier in \( u \) and \( v \) is different?

Let degree along \( u \) be \( m \) and along \( v \) be \( n \), with \( m < n \).

- How does this work?

Point on the surface can still be found out. But solution is no longer symmetrical.
Parametric Surfaces

- Bézier Surfaces – The Tensor Product
  - A surface can be thought of as being swept out by a moving and deforming curve.
  - Let the sweep curve be \( b^m(u) = \sum_{i=0}^{m} b_i J^m_i(u) \)
  - Each control point traverses the curve \( b_i = \sum_{j=0}^{n} b_{ij} J^n_j(v) \)
  - Combining, we get the surface as:
    \[
    b^{m,n}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} b_{ij} J^m_i(u) J^n_j(v)
    \]

- The tensor product technique constructs surfaces by “multiplying” two curves.
- Surfaces generated this way are called tensor product surfaces.
- Bézier surfaces or B-spline surfaces are all tensor product surfaces.
Parametric Surfaces

deCasteljau( \( m+1 \times n+1 \) control points, \( u, v \))
{
    for \( i := 0 \) to \( n \) do
        apply deCasteljau's algorithm to the \( i \)-th column of control points with \( v \);
        let the point obtained be \( q_i(v) \);
        apply deCasteljau's algorithm to \( q_0(v), q_1(v), ..., q_n(v) \) with \( u \);
        the point obtained is \( p(u,v) \);
}

Parametric Surfaces

- Bézier Surfaces – Properties
  - Affine Invariance
  - Convex Hull
  - Boundary curve and end-point interpolation
  - Non-local control
  - Degree elevation

Parametric Surfaces

\[
\frac{\partial}{\partial u} b^m_n(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \Delta^m b_{ij}(u,v) J_{m-i,j}(v)
\]

\[
\frac{\partial}{\partial v} b^m_n(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \Delta^n b_{ij}(u,v) J_{n-j,i}(u)
\]
Parametric Surfaces

- Bézier Surfaces - Derivatives

\[ \frac{\partial}{\partial u} b^m_n(u, v) = \sum_{j=0}^{m} \frac{\partial}{\partial u} b_j n_j(u) M_j^n(v) \]

Cross Boundary Derivatives

\[ \frac{\partial}{\partial u} b^m_n(u, v) = \sum_{j=0}^{m} \frac{\partial}{\partial u} b_j n_j(u) M_j^n(v) \]

What can we say about the geometric/parametric continuity of these patches?

Parametric Surfaces

- Bézier Surfaces - Composing Patches

Parametric Surfaces

- Bézier Surfaces - The Utah Teapot

Parametric Surfaces

- Bézier Surfaces - Surface Normals

\[ \mathbf{n}(u, v) = \frac{\frac{\partial}{\partial u} b^m_n(u, v) \times \frac{\partial}{\partial v} b^m_n(u, v)}{|\frac{\partial}{\partial u} b^m_n(u, v) \times \frac{\partial}{\partial v} b^m_n(u, v)|} \]

Parametric Surfaces

- Bézier Surfaces - The Twist Vector

\[ \Delta_n^m = \sum_{j=0}^{m} \frac{\partial}{\partial n} b_j n_j(u) M_j^n(v) \]

Parametric Surfaces

- B-Splines - Recap

\[ P(t) = \sum_{i=0}^{n} B_i N_i(t) \]

With \( t \geq t_{e-1} \) and \( 2 \leq k \leq n+1 \)

\( B_i \) - position vectors of the \( n+1 \) vertices of the control polygon

\( N_i(t) \) - normalized B-spline basis functions

\[ N_i(t) = \begin{cases} 1, & \text{if } x_{i-1} < t < x_i \\ 0, & \text{otherwise} \end{cases} \]

\[ N_i(t) = \frac{(t-x_j)N_{i,j-1}(t)}{x_i-x_j} + \frac{(x_{i+1}-t)N_{i+1,j}(t)}{x_{i+1}-x_i} \]

Cox-deBoor Basis
Parametric Surfaces

- B-Splines Surfaces
  \[ b(u, v) = \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} B_{ij}(u) N_i(u) N_j(v) \]

- Properties
  - Affine Invariance
  - Strong Convex hull
  - Local Control

\[ b(u, v) = \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} B_{ij}(u) N_i(u) N_j(v) \]

Polygonal Meshes

- The object is often rendered as a collection of polygons.
  - Collection of edges and vertices.
  - Each edge is shared by at most two polygons
  - How to store a mesh?
    - Explicit Representation
    - A set of vertices \((x_i, y_i, z_i)\)...
    - Edges connect successive vertices

Explicit Representation

- A set of vertices \((x_i, y_i, z_i)\)...
- Vertices stored in order of traversal.
- Edges connect successive vertices.
- Difficult to manipulate
- Multiple storage of points

Explicit Representation

- A set of vertices \((x_i, y_i, z_i)\)...
- Vertices stored once in a list \(V\):
  \[ V = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\} \]
- Edge: \(E = (V_i, V_j, P)\)
- Polygon: \(P = (E_1, E_2, E_3, E_4)\)

Explicit Representation

- A set of vertices \((x_i, y_i, z_i)\)...
- Vertices stored once in a list \(V\):
  \[ V = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\} \]
- Edge: \(E = (V_i, V_j, P)\)
- Polygon: \(P = (E_1, E_2, E_3, E_4)\)
Polygonal Meshes

- Winged Edge Data Structure

<table>
<thead>
<tr>
<th>Edge</th>
<th>Vertices</th>
<th>Faces</th>
<th>Left Parent</th>
<th>Right Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>e</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>