Geometry

2021年7月30日 14:36

§1 Introduction

1. represent geometry

Implicit
- points that satisfy specific relationship $F(x,y,z)=0$ Inside/Outside is easy
- Constructive Solid Geometry (Boolean operations) $A \cup B$; $A \cap B$; $A \setminus B$
- Distance Functions
- Level Set Methods Sampling is easy
- Fractals 分形

Explicit
- given directly
- $\begin{cases} \text{parameter mapping} \\ f: R^2 \to R^3 \quad (u,v) \to (x,y,z) \end{cases}$

§2 Curves and Surfaces

1. Point Cloud (Explicit)
   list of points
   Useful for LARGE datasets
   often converted into meshes

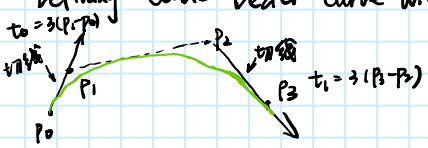
2. Polygon Mesh
   Easy to process / sample / simulate.
   More complicated data structures

3. How to represent an object.
   The Wavefront Object File (.obj) Format
   a text file that specifies vertices, normals, texture coordinates and connectivities.
   How vertices are connected with each other.

4. Curves

Bézier Curve.

Defining Cubic Bézier Curve with tangents

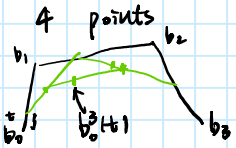


de Casteljau Algorithm.

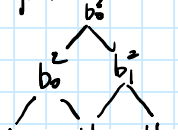
3 points (quadratic Bezier)

For $t \in (0,1)$:
Find $b_0^2(t)$
And connect all these points
⇒ Bezier curve.

4 points



Algebraic Formula

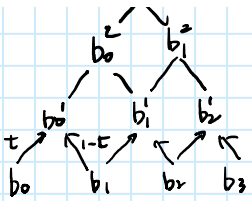


$$b^n(t) = b_0^n(t) = \sum_{j=0}^{n} b_j B_j^n(t)$$

$b^n(t)$: Bezier curve order n

b_j: Control

$B_j^n(t)$: Bernstein polynomial



Bezier curve order n

$\sum_{j=0}^{n}$

control points

Bernstein polynomial 二项展开

$= C_n^j t^j (1-t)^{n-j}$

Properties of Bezier curve.

cubic: $b(0) = b_0$, $b(1) = b_3$

$b'(0) = 3(b_1 - b_0)$, $b'(1) = 3(b_3 - b_2)$

transform curve by transforming control points

(Affine transformation property) Note! projection 不是 Affine trans.)

Curve is within convex hull of control points

(convex property)

C^0 continuity :

C^1 continuity : $f'(x)$ is continuous

5. Other types of splines 样条

- B-splines
  - basic splines
  - satisfy all important properties that Bezier curves have.
  - refer to 胡事民's course on CG.

6. Surfaces

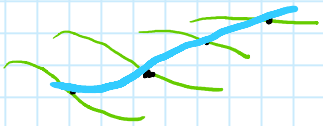
extend Bezier curve to surfaces

4 × 4 array of control points

每行4个点插值得到一个点，共4个点

对此4点做 Bezier 曲线

对不同 t，该曲线扫过面积为曲面



Evaluating Bezier surfaces

7. Mesh operations = Geometry Processing

① Mesh subdivision upsampling

② Mesh simplification downsampling

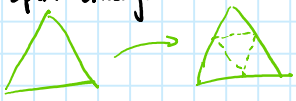
③ Mesh regularization. = modify sample distribution

3

1. Subdivision

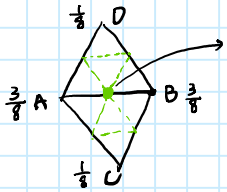
— Loop subdivision

① Split triangles



② Assign new vertex positions according to weights

For new vertices



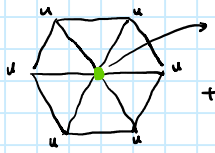
Update to

$\frac{3}{8}(A+B) + \frac{1}{8}(C+D)$

For old vertices

$\frac{1}{8}$

For old vertices



Update to:

$(1-n\cdot u)*$ original_position

$+\ u *$ neighbor_position_sum

n = vertex degree

$u = \frac{3}{16}$ if $n=3$, $\frac{3}{8n}$ otherwise

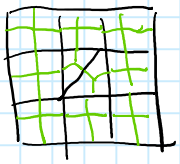
- Catmull-Clark Subdivision General Mesh.

  - Subdivision Step:

    add vertex in each face

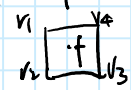
    add midpoint on each edge

    Connect all new vertices



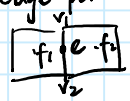
  - FYI: Vertex Update Rules (Quad Mesh)

Face point:



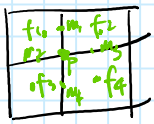
$f = \frac{v_1+v_2+v_3+v_4}{4}$

Edge point



$e = \frac{v_1+v_2+f_1+f_2}{4}$

Vertex point



$v = \frac{f_1+f_2+f_3+f_4+2(m_1+m_2+m_3+m_4)+4p}{16}$

2. Mesh Simplification

① edge collapse

Quadric Error metric

new vertex should minimize L2 distance to previously related triangle planes

Idea: compute edge midpoint, measure quadric error

Iteratively collapse edges

Assign score with quadric error metric.

- approximate distance to surface as sum of distances to planes containing triangles
- iteratively collapse edge with smallest score
- greedy algorithm.