Computer Graphics 2018

11. Complex 3D

Hongxin Zhang
State Key Lab of CAD&CG, Zhejiang University

2018-12-12

General spline curves

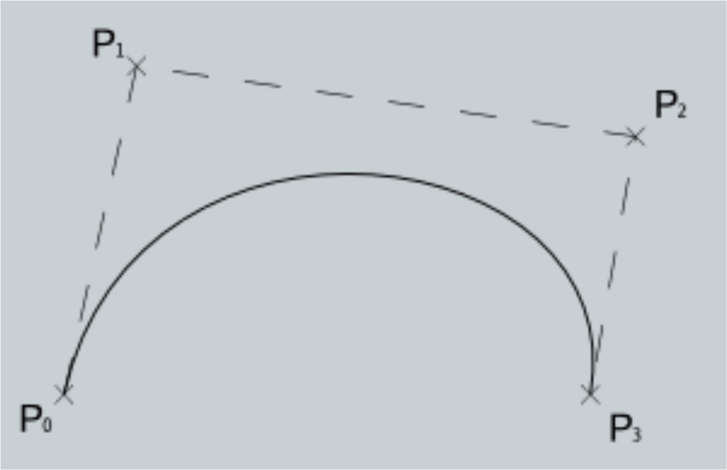
parametric curve

basis functions

$$\mathbf{P}(t) = \sum_i \mathbf{P}_i B_i(t)$$

$t \in [t_0, t_1)$

control pints



Bézier curve

Bézier curve

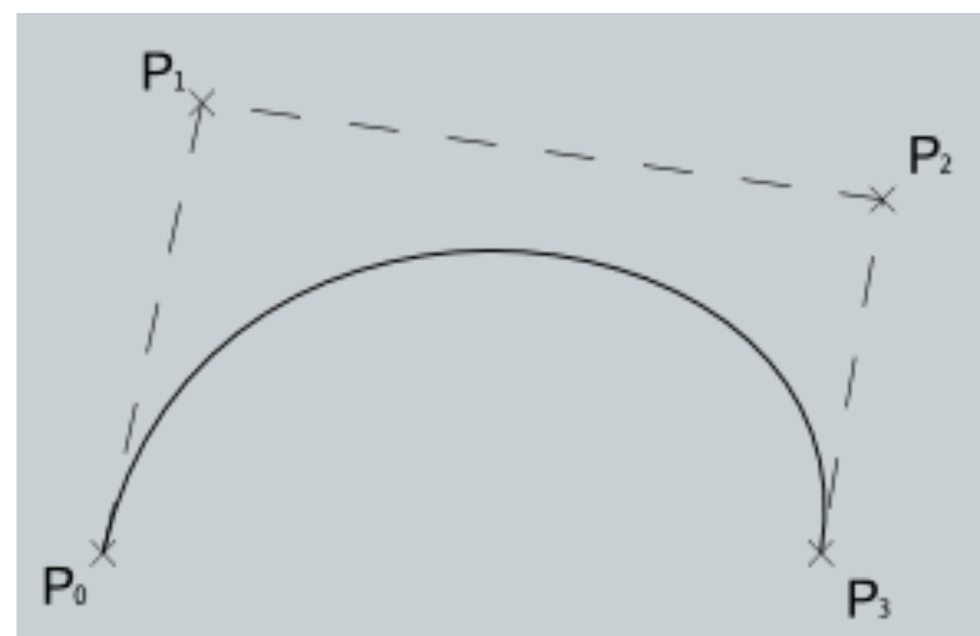
$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \quad t \in [0,1]$$

where, P_i (i=0,1,…,n) are control points.

$$\boxed{B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, t \in [0,1]}$$

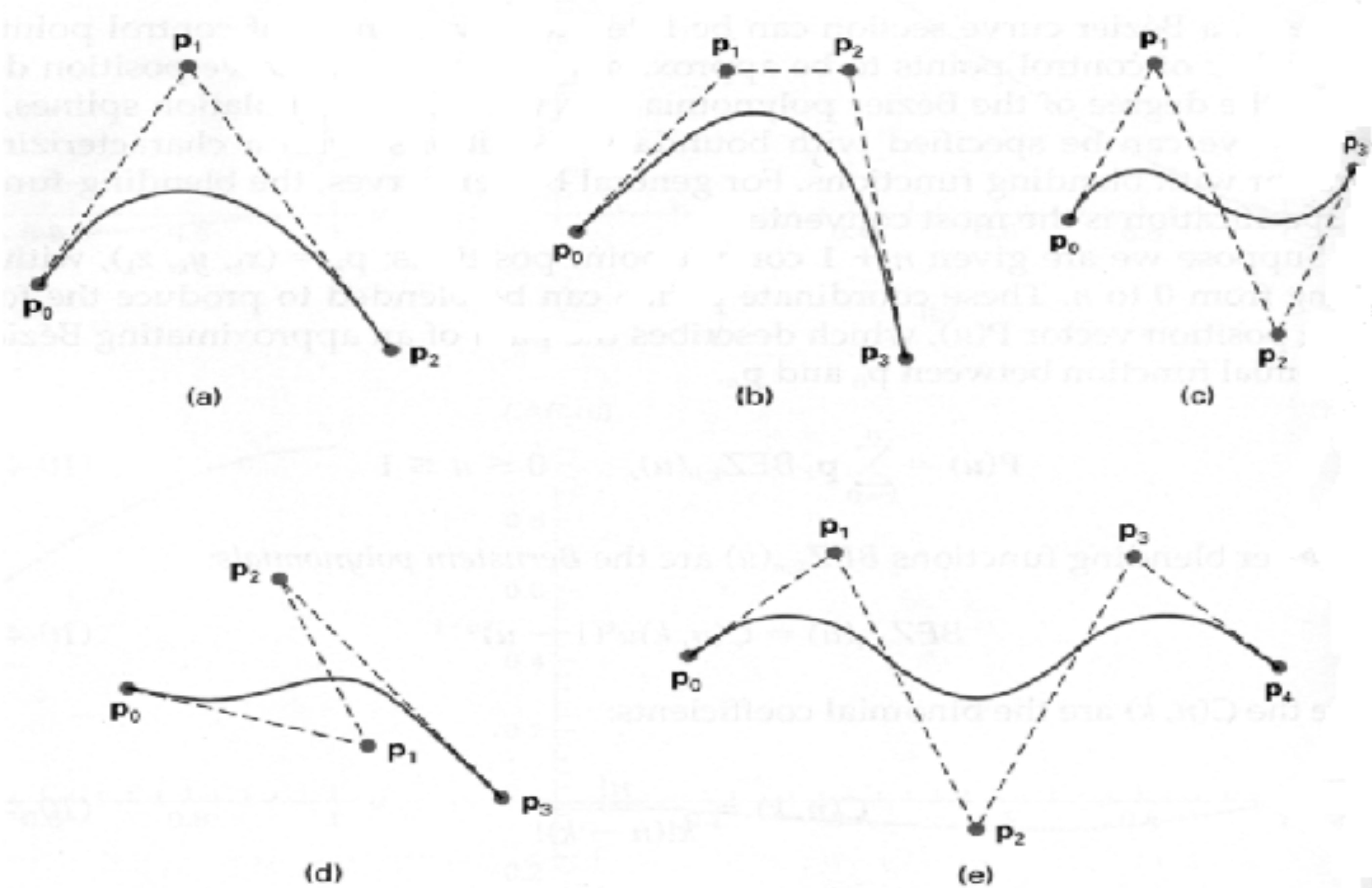
Bernstein basis

$$\begin{cases} X(t) = \sum_{i=0}^{n} x_i B_{i,t}(t) \\ Y(t) = \sum_{i=0}^{n} y_i B_{i,t}(t) \end{cases}$$



$$C(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}, \quad P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

Bézier curve



Bézier curve

Rational Bézier Curve

$$\boldsymbol{R}(t) = \frac{\sum_{i=0}^{n} B_{i,n}(t)\omega_i \boldsymbol{P}_i}{\sum_{i=0}^{n} B_{i,n}(t)\omega_i} = \sum_{i=0}^{n} R_{i,n}(t)\boldsymbol{P}_i$$

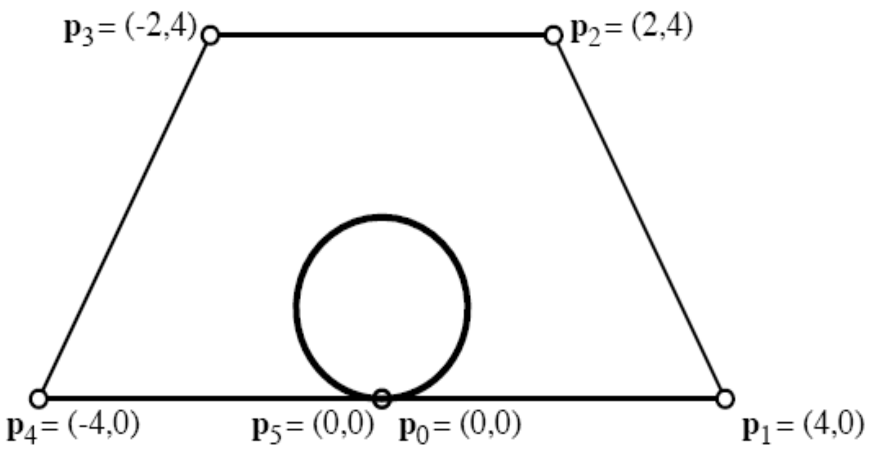


Figure 2.19: Circle as Degree 5 Rational Bézier Curve.

where $B_{i,n}(t)$ is Bernstein basis, ω_i is the weight at p_i .

It's a generalization of Bézier curve, which can express more curves, such as circle.

浙江大学计算机学院
数字媒体与网络技术

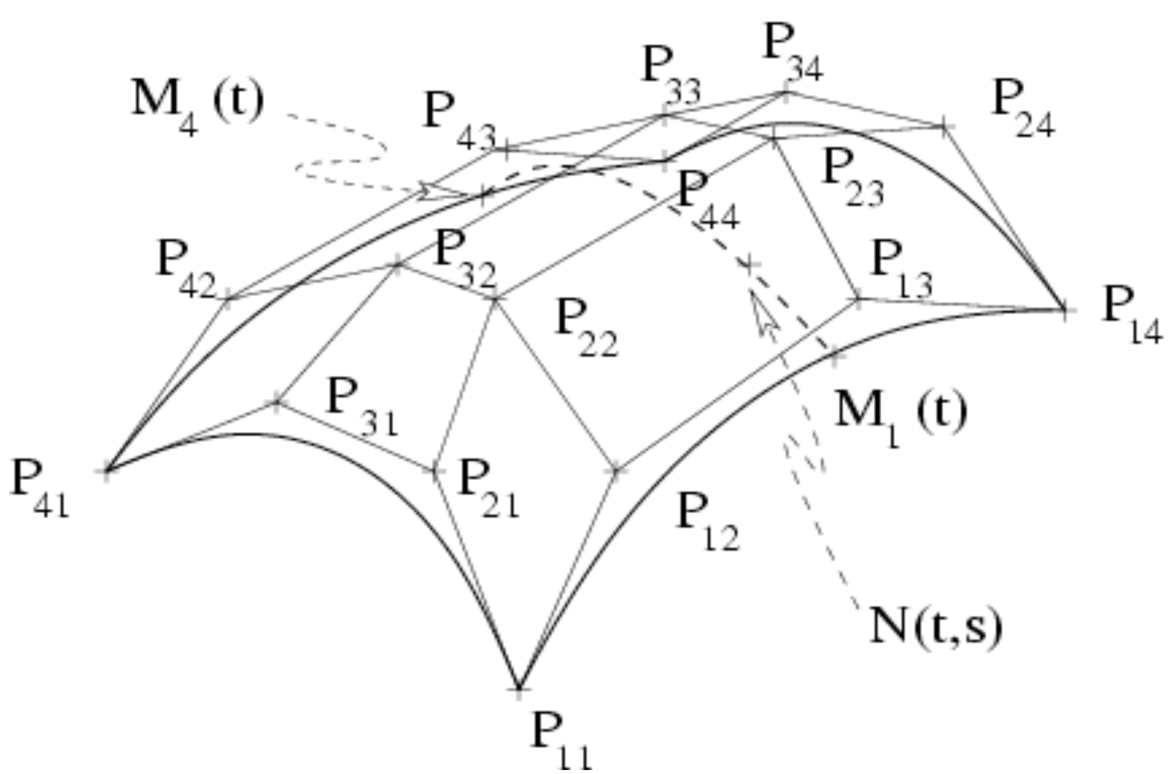
Bézier surface

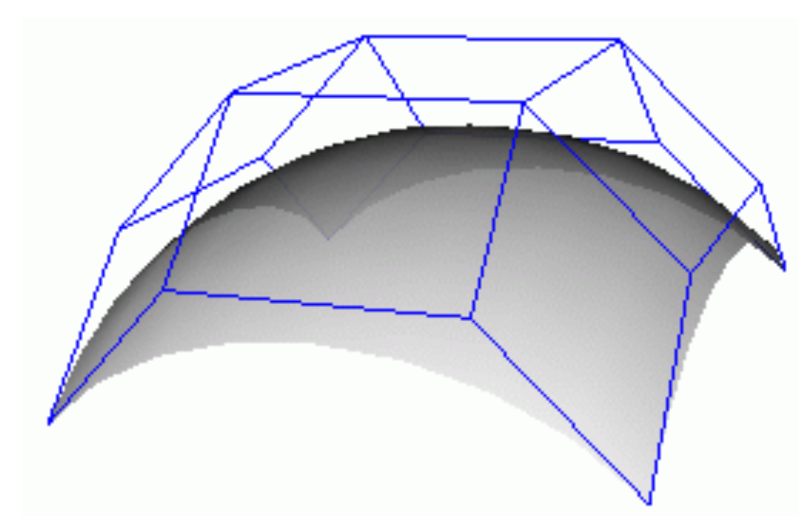
Bézier surface

Bézier surface:

$$\boldsymbol{S}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}\boldsymbol{P}_{ij}B_{i,n}(u)B_{j,m}(v), \qquad 0\le u,v\le 1$$

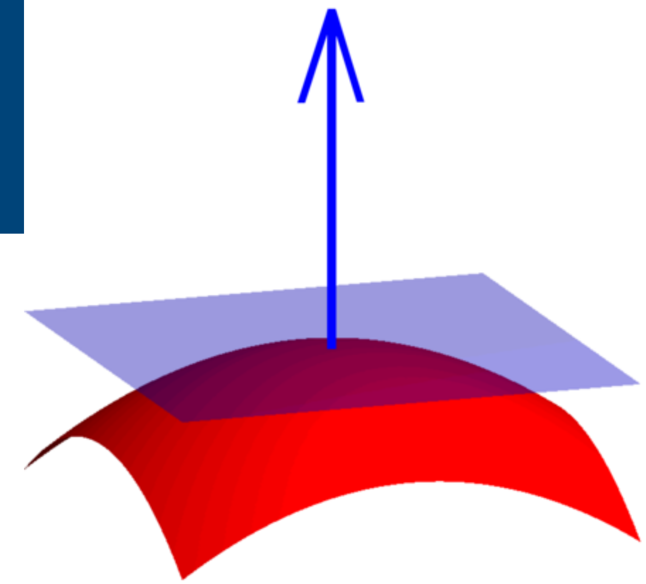
where $B_{i,n}(u)$和$B_{j,m}(v)$ Bernstein basis with n degree and m degree, respectively, $(n+1)\times(m+1)$ $\boldsymbol{P}_{i,j}$(i=0,1,…,n; j=0,1,…,m) construct the control meshes.





Bézier surface

normal vector of Bézier surface



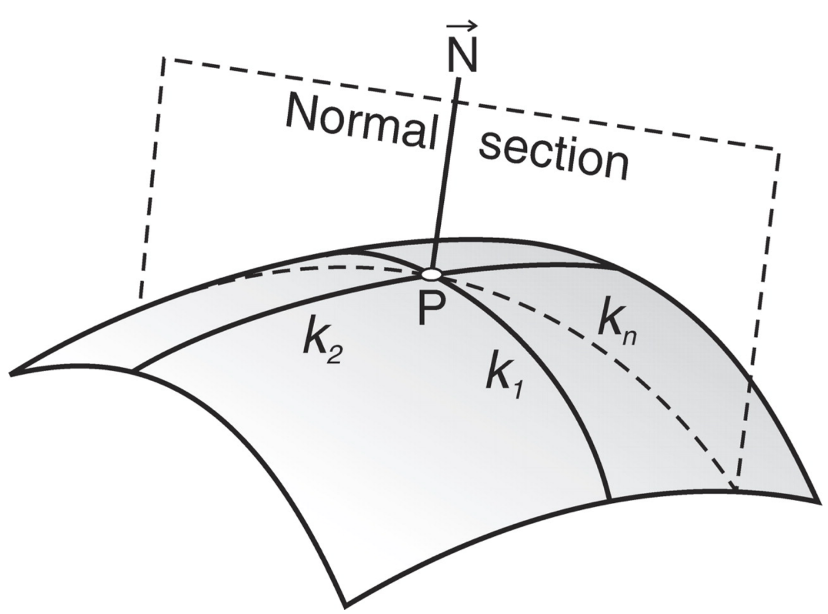
partial derivation of Bézier surface $\boldsymbol{S}(u,v)$:

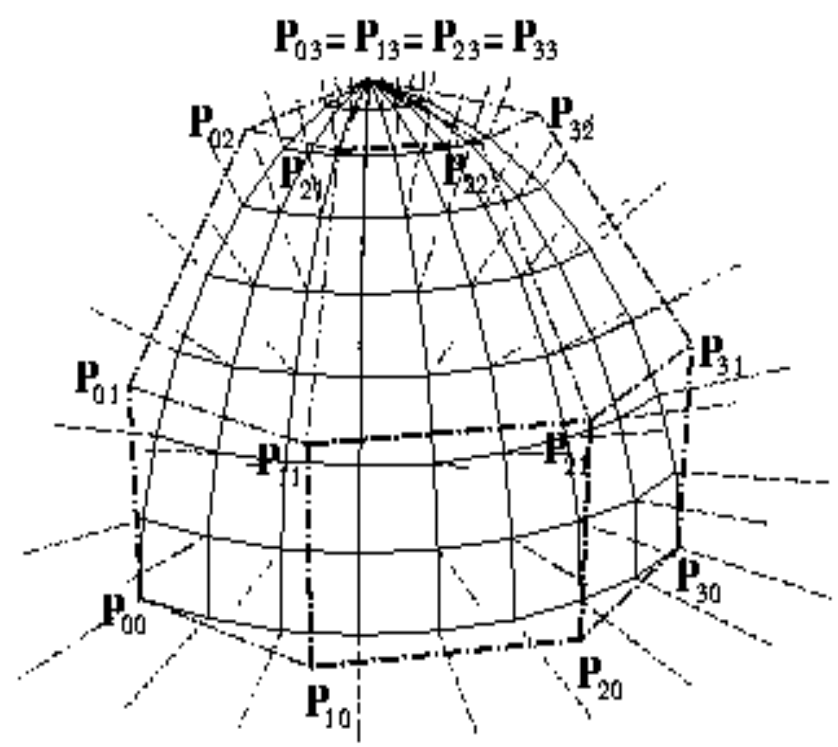
$$\frac{\partial}{\partial u}\boldsymbol{S}(u,v)=\frac{\partial}{\partial u}\sum_{i=0}^{n}\sum_{j=0}^{m}\boldsymbol{P}_{ij}B_{i,n}(u)B_{j,m}(v)=n\sum_{i=0}^{n-1}\sum_{j=0}^{m}(\boldsymbol{P}_{i+1,j}-\boldsymbol{P}_{ij})B_{i,n-1}(u)B_{j,m}(v)$$

$$\frac{\partial}{\partial v}\boldsymbol{S}(u,v)=\frac{\partial}{\partial v}\sum_{i=0}^{n}\sum_{j=0}^{m}\boldsymbol{P}_{ij}B_{i,n}(u)B_{j,m}(v)=m\sum_{i=0}^{n}\sum_{j=0}^{m-1}(\boldsymbol{P}_{i,j+1}-\boldsymbol{P}_{ij})B_{i,n}(u)B_{j,m-1}(v)$$

normal $\boldsymbol{N}(u,v)$:

$$N(u,v)=\frac{\partial \boldsymbol{S}(u,v)}{\partial u}\times\frac{\partial \boldsymbol{S}(u,v)}{\partial v}$$



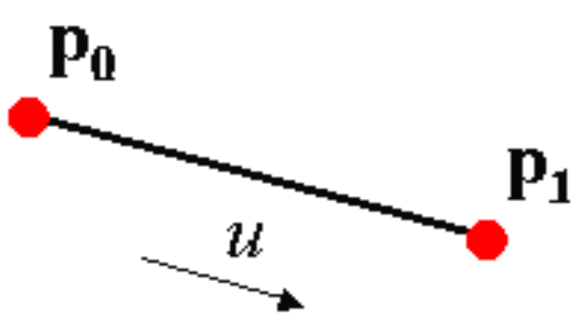


NURBS curve

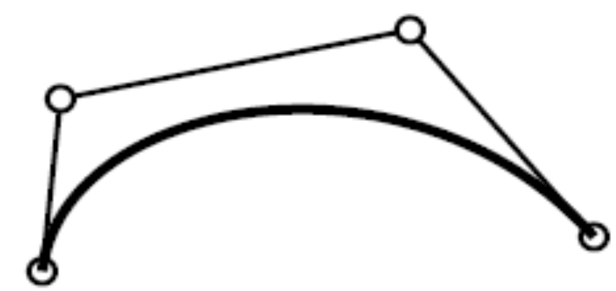
B-spline curve

- disadvantages of Bézier curve:

1. control points determine the degree of the curve. many control points means high degree.
2. It's global. A control point influences the whole curve.



Degree 2



Degree 3

de Boor et al. replaced Bernstein basis with B-spline basis to generate B-spline curve.

NURBS curve

B-spline curve:

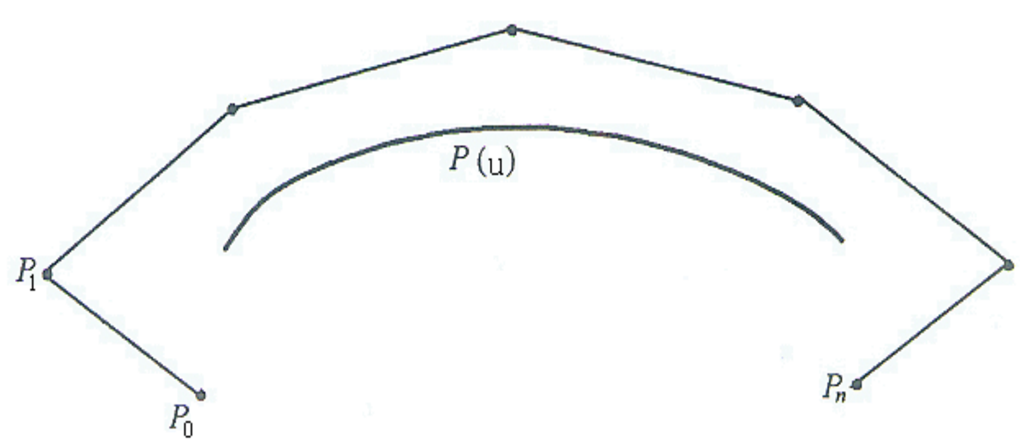
$$C(u)=\sum_{i=0}^{n} P_i N_{i,p}(u) \qquad a \le u \le b$$

Where $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$ are control points, $\mathbf{u}=[u_0=a, u_1, \ldots, u_i, \ldots, u_{n+k+1}=b]$.

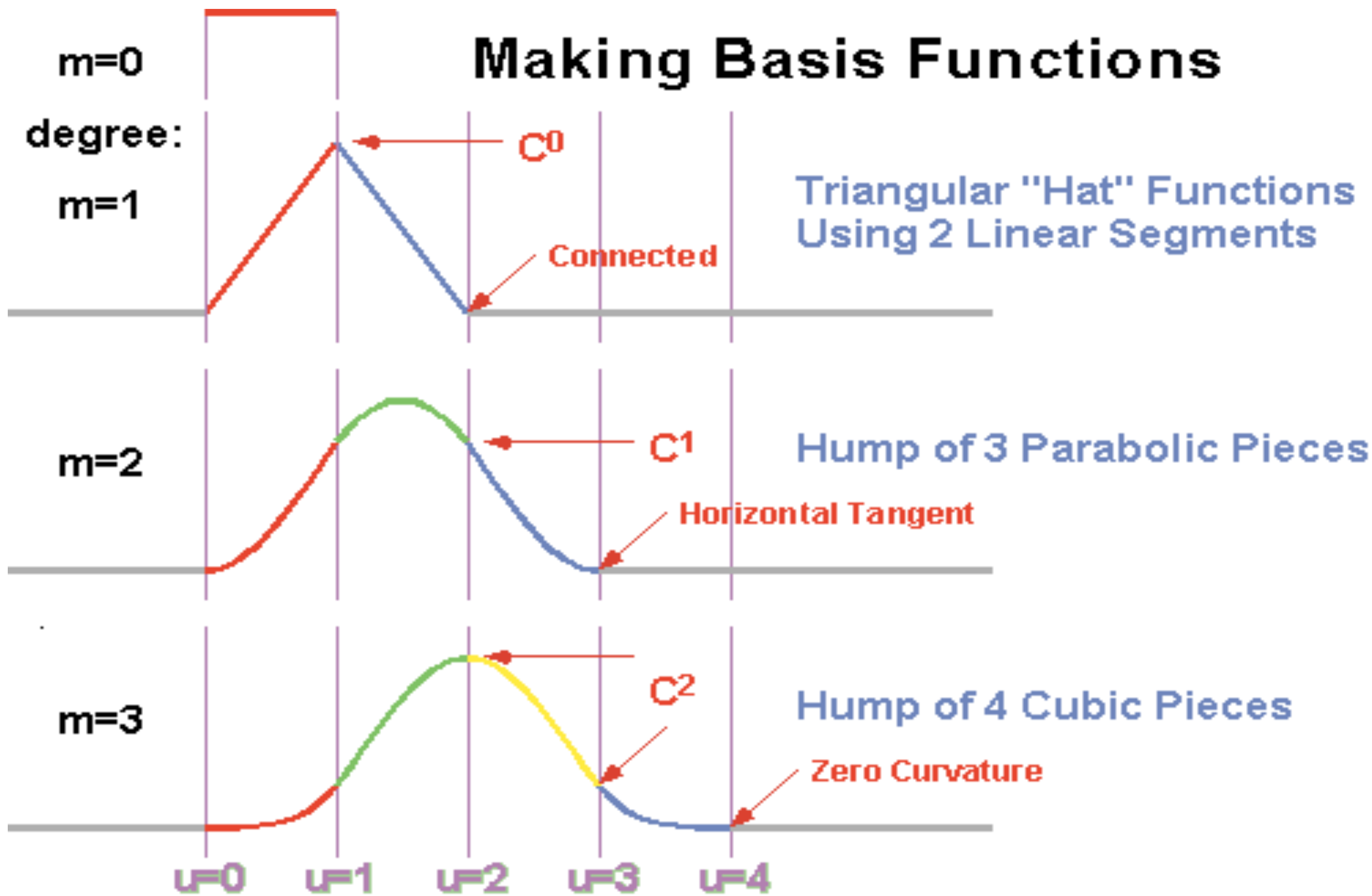
$$N_{i,0}(u)=\begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u),$$

$$\frac{0}{0}=0$$



B-spline basis



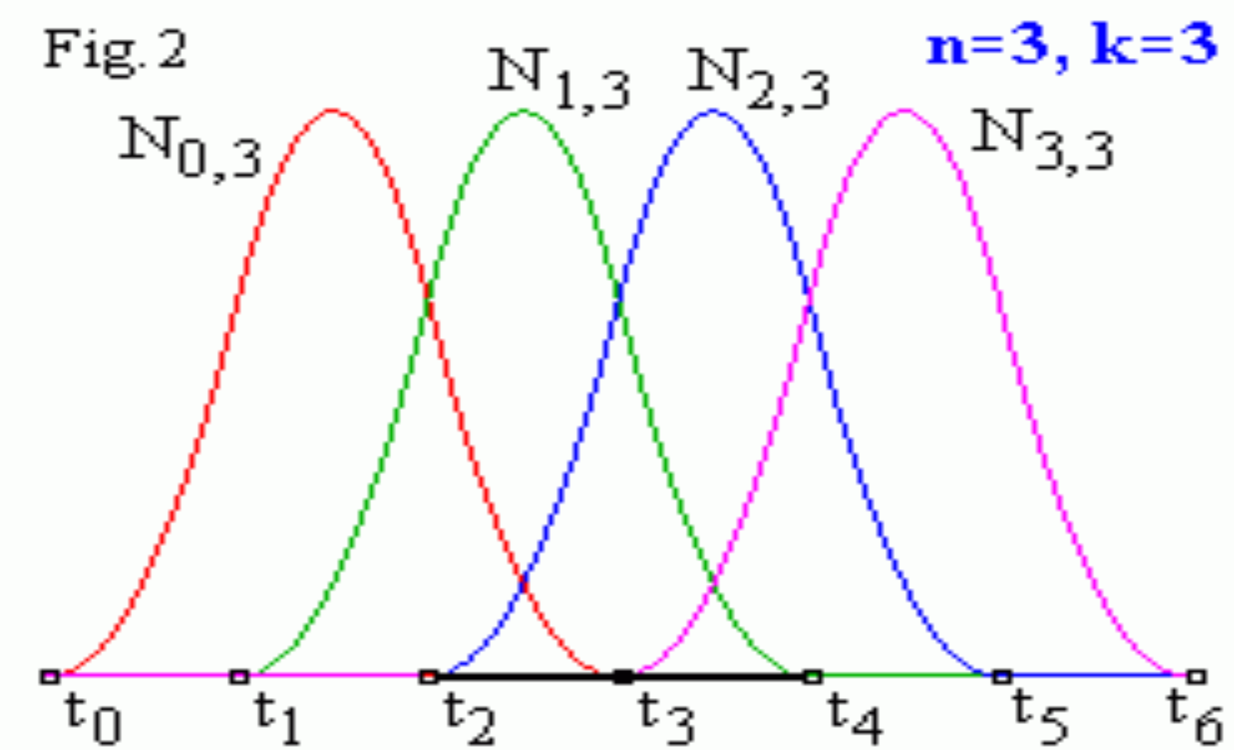
$$U = \{u_i\}_{i=-\infty}^{\infty}$$

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & else \end{cases}$$

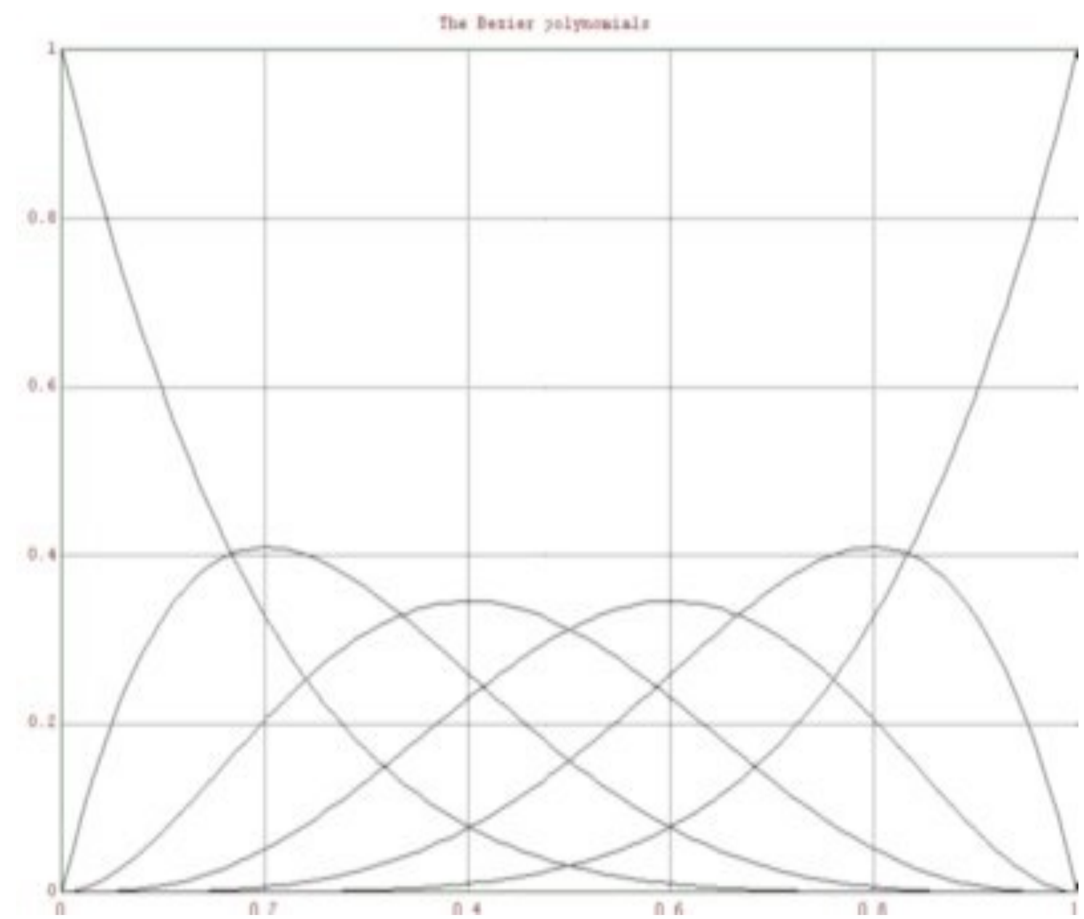
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),$$

$$\frac{0}{0} = 0$$

B-spline basis v.s. Bernstein ~



$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),\ i = 0,1,\ldots,n.$$



$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & else \end{cases}$$

$$U = \{u_i\}_{i=-\infty}^{\infty}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u),$$

$$\frac{0}{0} = 0$$

NURBS curve

properties of B-spline basis

1. localization: $N_{i,p}(u)$ >0 only when $u \in [u_i, u_{i+p+1}]$.

$$N_{i,p}(u) = \begin{cases} > 0, & u_i \le u < u_{i+p+1} \\ = 0, & u < u_i \text{ »Ò} u > u_{i+p+1} \end{cases}$$

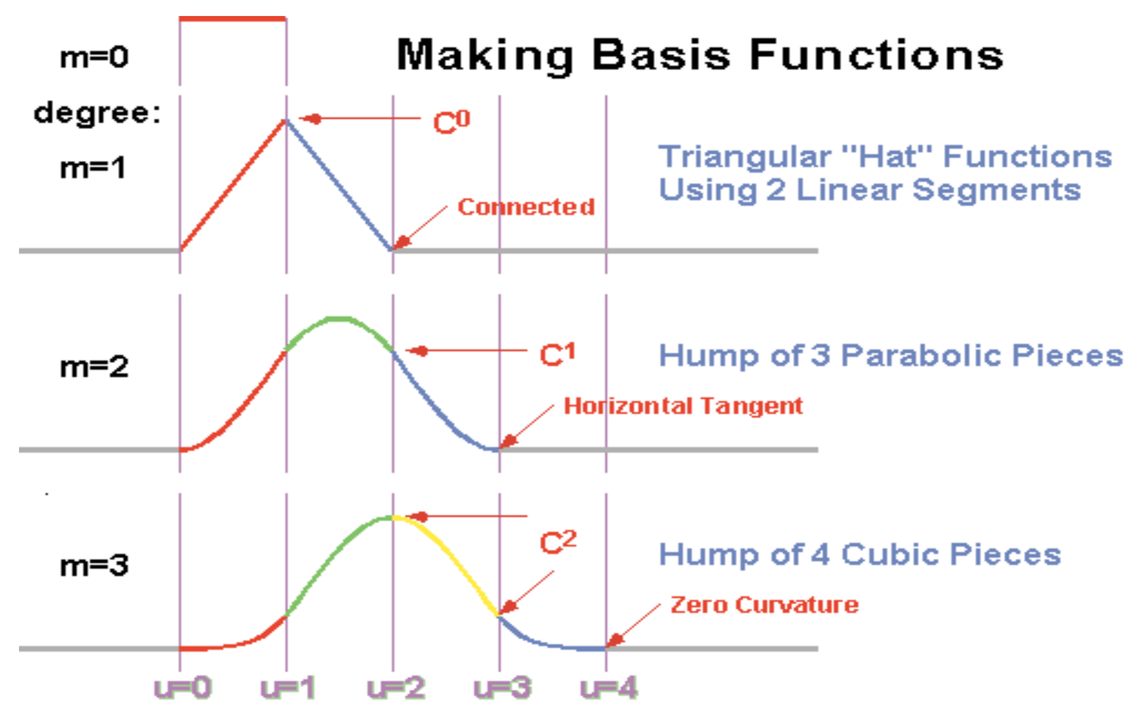
2. normalization:

$$\sum_{j=-\infty}^{\infty} N_{j,p}(u) = \sum_{j=i-p}^{i} N_{j,p}(u) = 1,\ u \in [u_i, u_{i+1})$$

3. piecewise polynomial: $N_{i,p}(u)$ is a polynomial with degree < p, in every $[u_j, u_{j+1})$

4. differential:

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$



浙江大学计算机学院
数字媒体与网络技术

NURBS curve

Properties of B-spline curve:

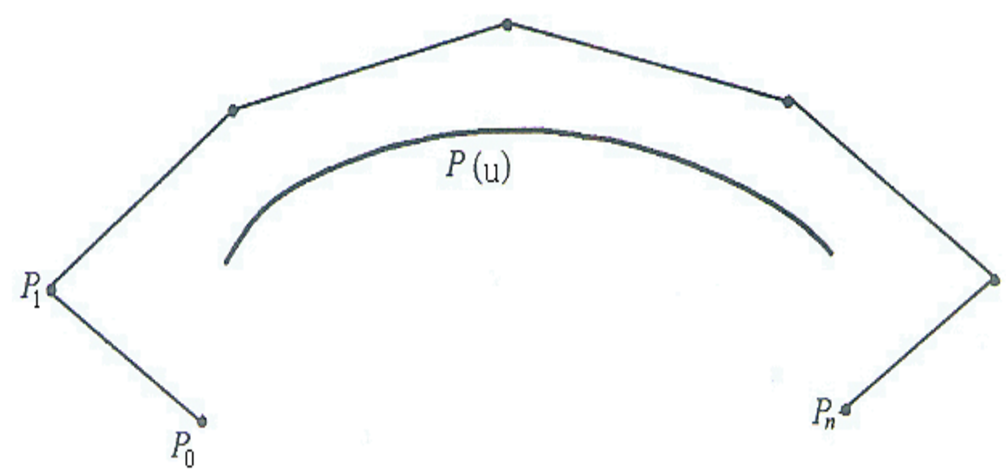
1. Convex Hull Property

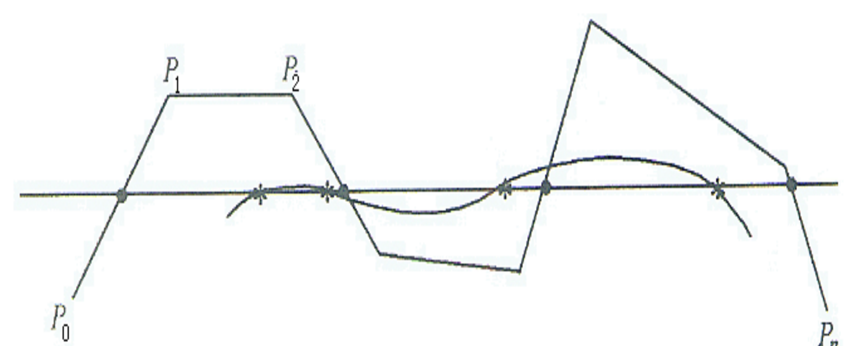
2. variation diminishing property.

3. Affine Invariance

4. local

5. piecewise polynomial





NURBS curve

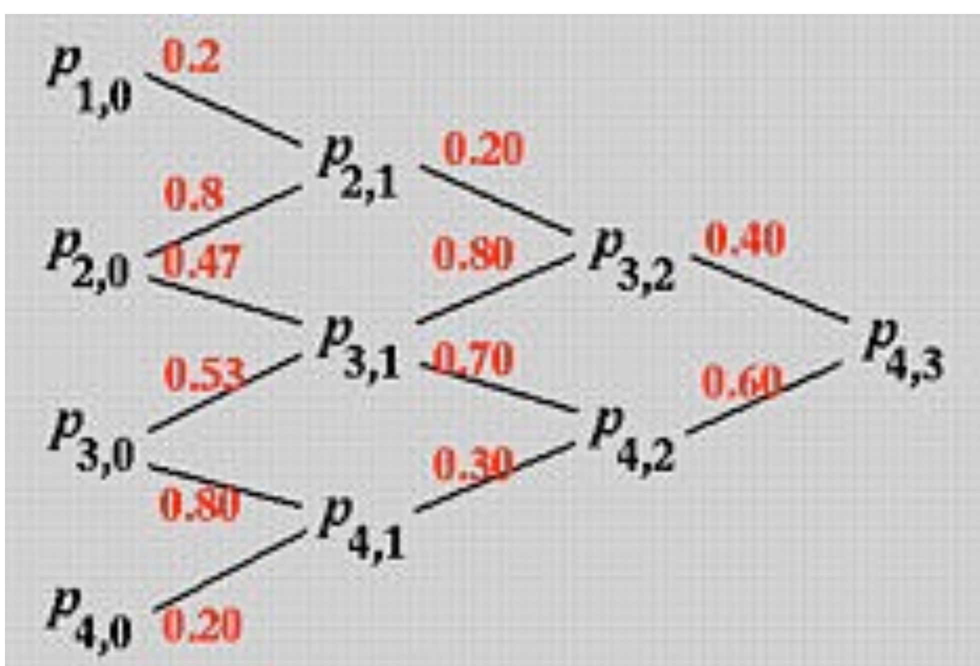
B-spline---de Boor algorithm

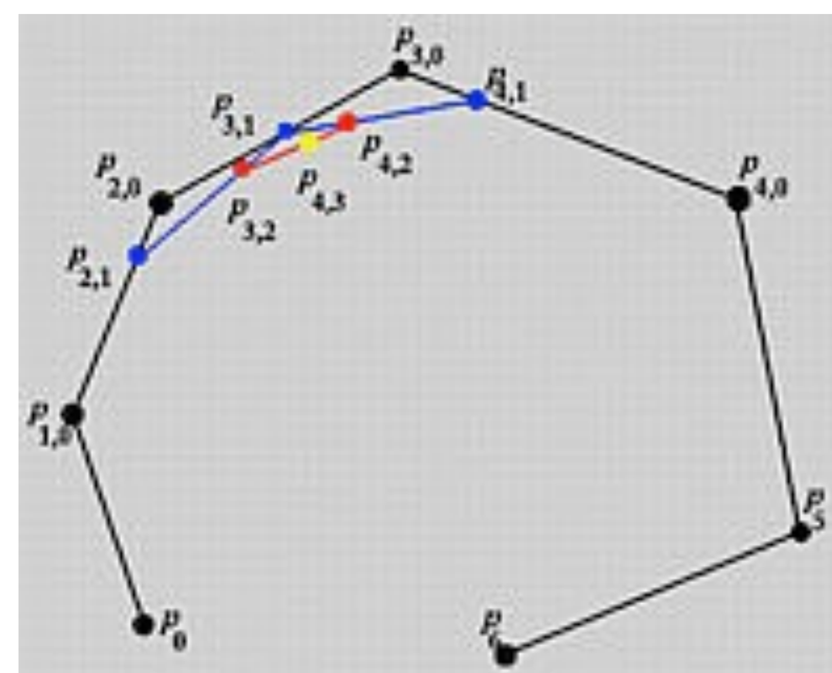
to calculate the point of B-spline curve ***C***(*u*) at *u*:

1. find the interval where *u* lies in : $u \in [u_j, u_{j+1})$;
2. curve in $u \in [u_j, u_{j+1})$ is only determined by $\boldsymbol{P}_{j-p}, \boldsymbol{P}_{j-p+1}, \ldots, \boldsymbol{P}_j$;
3. calculate

$$\boldsymbol{P}_i^r(u) = \begin{cases} \boldsymbol{P}_i & r = 0, i = j-p;\ j-p+1, \cdots, j; \\ \dfrac{u - u_i}{u_{i+k-r} - u_i} \boldsymbol{P}_i^{r-1}(u) + \dfrac{u_{i+k-r} - u}{u_{i+k-r} - u_i} \boldsymbol{P}_{i-1}^{r-1}(u), & r = 1, 2, \cdots k-1;\ i = j-p+r, j-p+r+1, \cdots, j. \end{cases}$$

4. $\boldsymbol{P}_j^{k-1}(u) = C(u)$





NURBS curve

Catmull-Clark and Doo-Sabin subdivision

Start from

$$P^i = \left(\cdots, p_{-1}^i, p_0^i, p_1^i, p_2^i, \cdots\right)$$

Catmull-Clark rules

$$p_{2j}^{i+1} = \frac{1}{8} p_{j-1}^i + \frac{6}{8} p_j^i + \frac{1}{8} p_{j+1}^i$$

$$p_{2j+1}^{i+1} = \frac{4}{8} p_j^i + \frac{4}{8} p_{j+1}^i$$

Doo-Sabin rules:

$$p_{2j}^{i+1} = \frac{3}{4} p_j^i + \frac{1}{4} p_{j+1}^i$$

$$p_{2j+1}^{i+1} = \frac{1}{4} p_j^i + \frac{3}{4} p_{j+1}^i$$

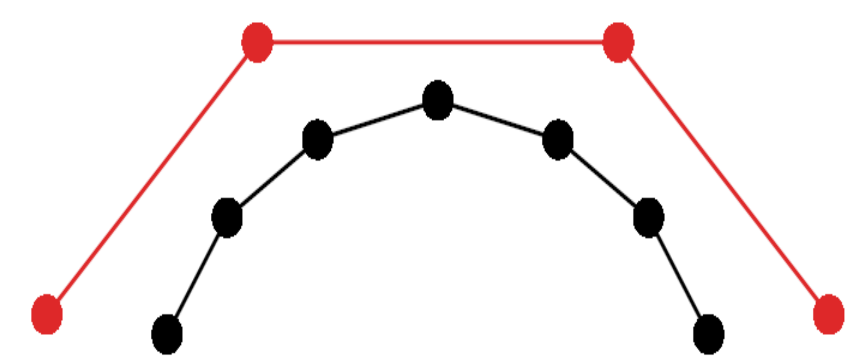


Figure 3: Subdividing an initial set of control points (upper, red) results in additional control points (lower, black), that more closely approximate a curve.

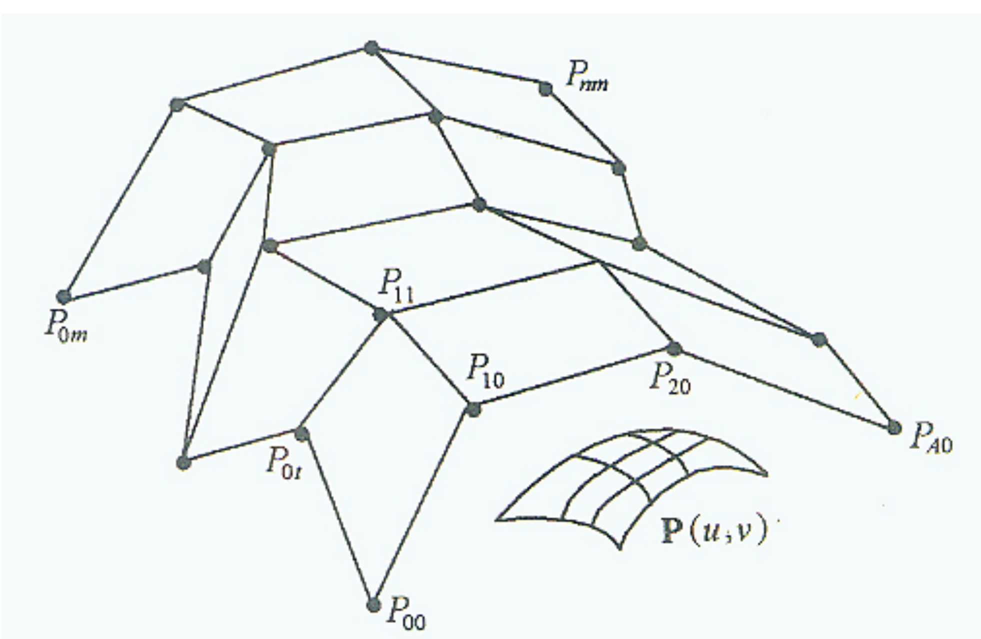
B-spline surface

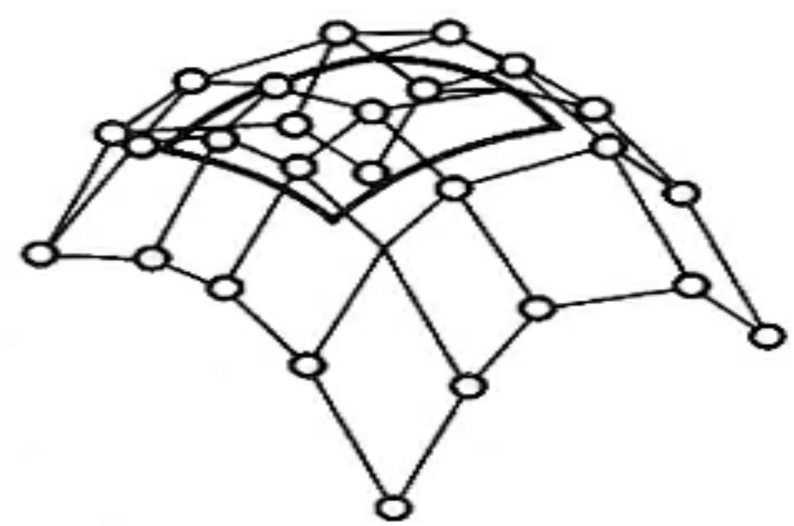
$(n+1)\times(m+1)$ control points: $\boldsymbol{P}_{i,j}$ (Degrees of u, v: p, q);

nodes: $U=[u_0,u_1,\ldots,u_{n+p+1}]$, $V = [v_0,v_1, \ldots, v_{m+q+1}]$,

Then a tensor B-spline surface with degree $p\times q$:

$$\boxed{\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\boldsymbol{P}_{i,j}}$$





NURBS surface

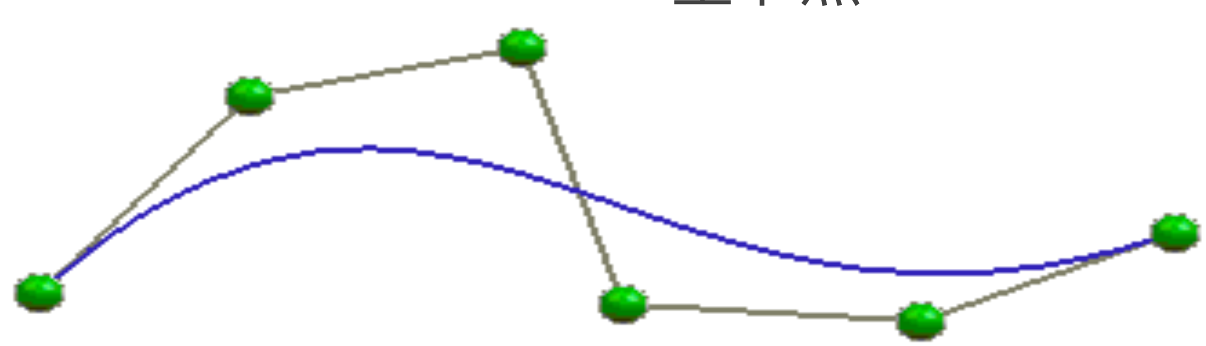
NURBS (Non-uniform **Rational** B-spline)

NURBS curves:

$$\boldsymbol{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\omega_i \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,p}(u)\omega_i}, \quad a \le u \le b$$

$$U = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}\}$$

重节点



NURBS surface

NURBS surface

$$S(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)\omega_{i,j}\boldsymbol{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)\omega_{i,j}}$$

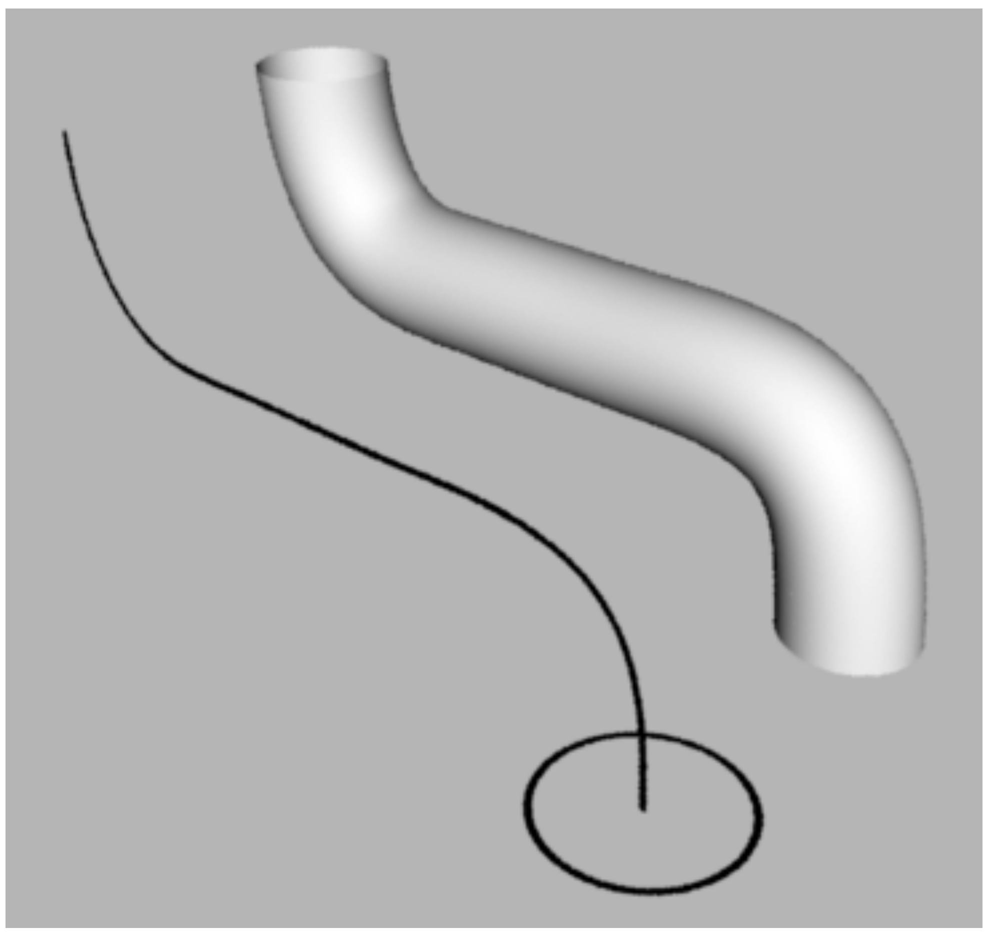
$0 \le u, v \le 1$

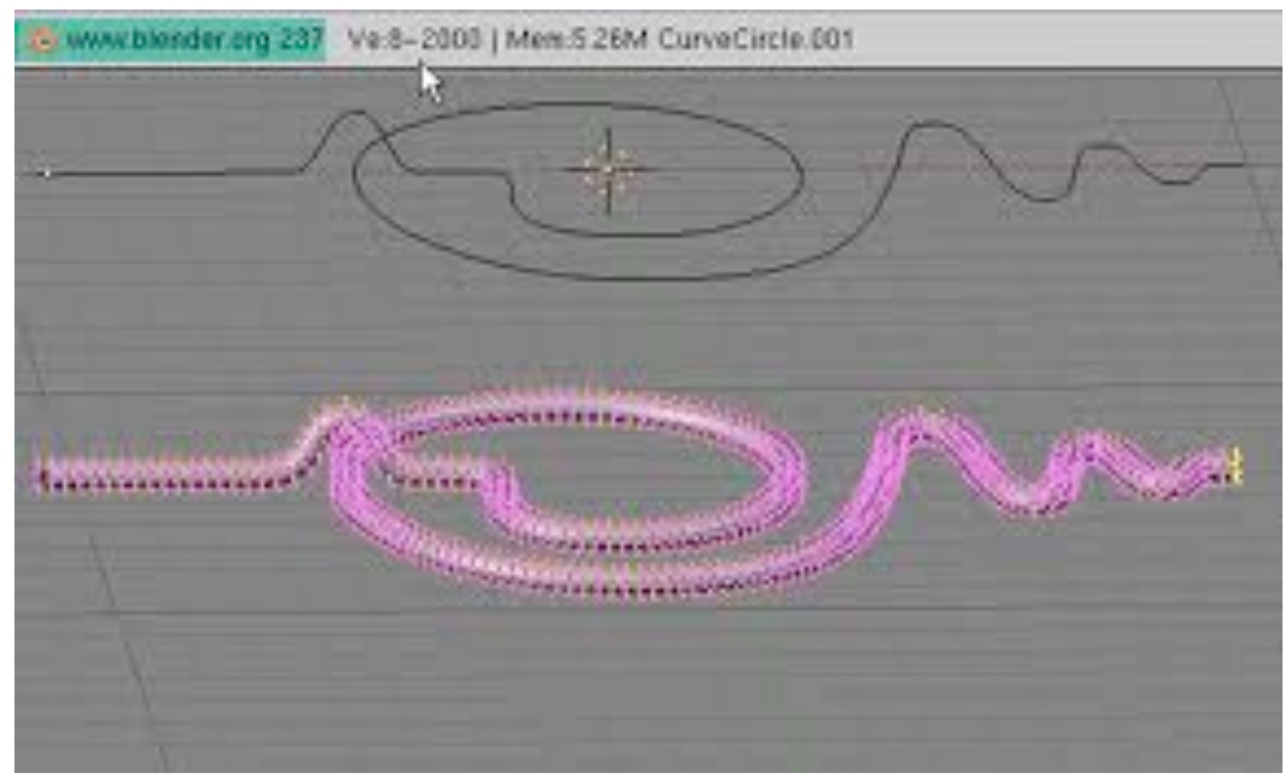
ω_{ij}: weights

$$U=\{\underbrace{0,\ldots,0}_{p+1},u_{p+1},\ldots,u_{r-p-1},\underbrace{1,\ldots,1}_{p+1}\}$$

$$V=\{\underbrace{0,\ldots,0}_{q+1},v_{q+1},\ldots,v_{s-q-1},\underbrace{1,\ldots,1}_{q+1}\}$$

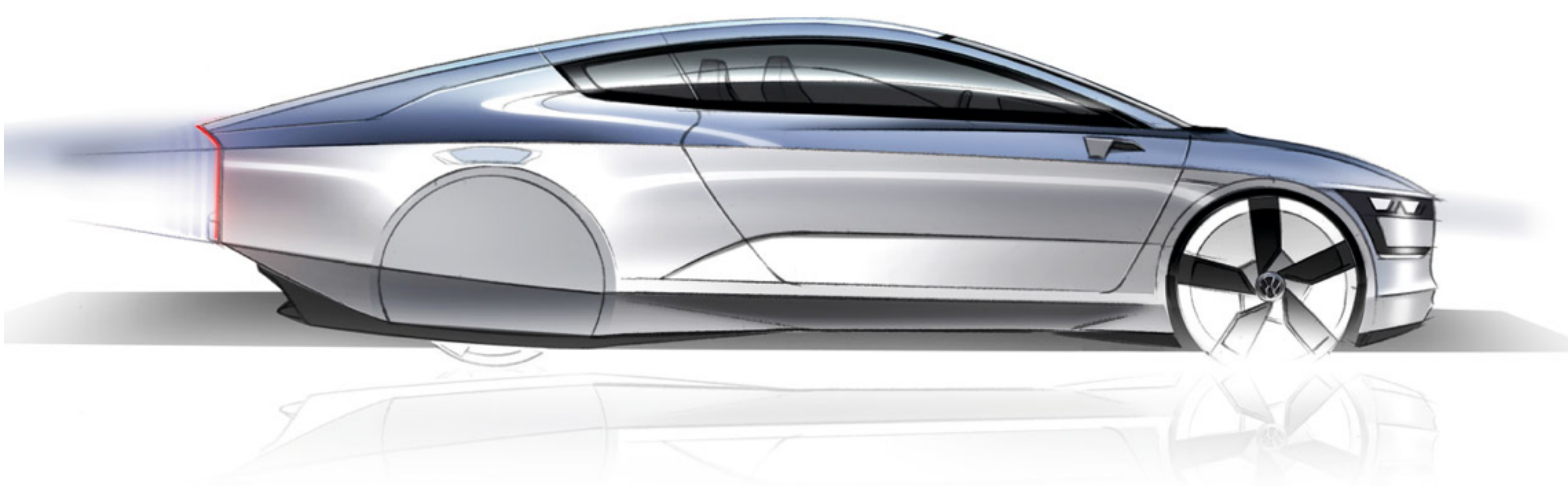
Sweeping





Homework 4

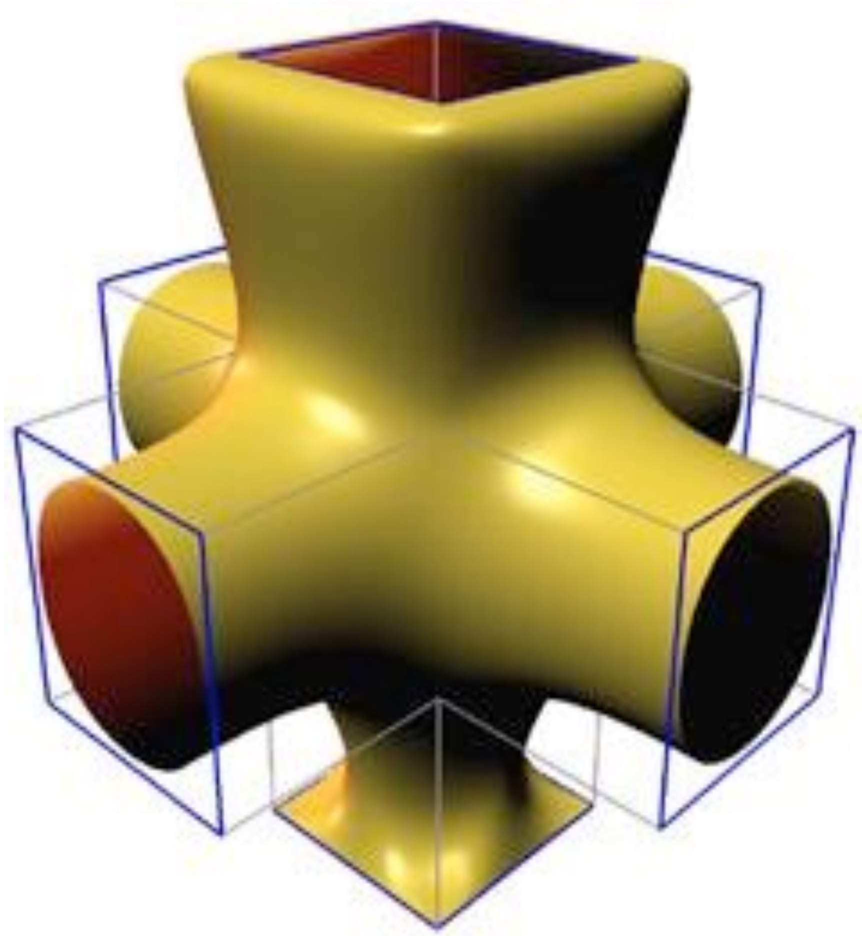
- Draw a circle by Cubic Rational Bezier Curves
  - Please give a detailed implementation in OpenGL/WebGL
  - Key: to understand what is weight in the rational spline representations

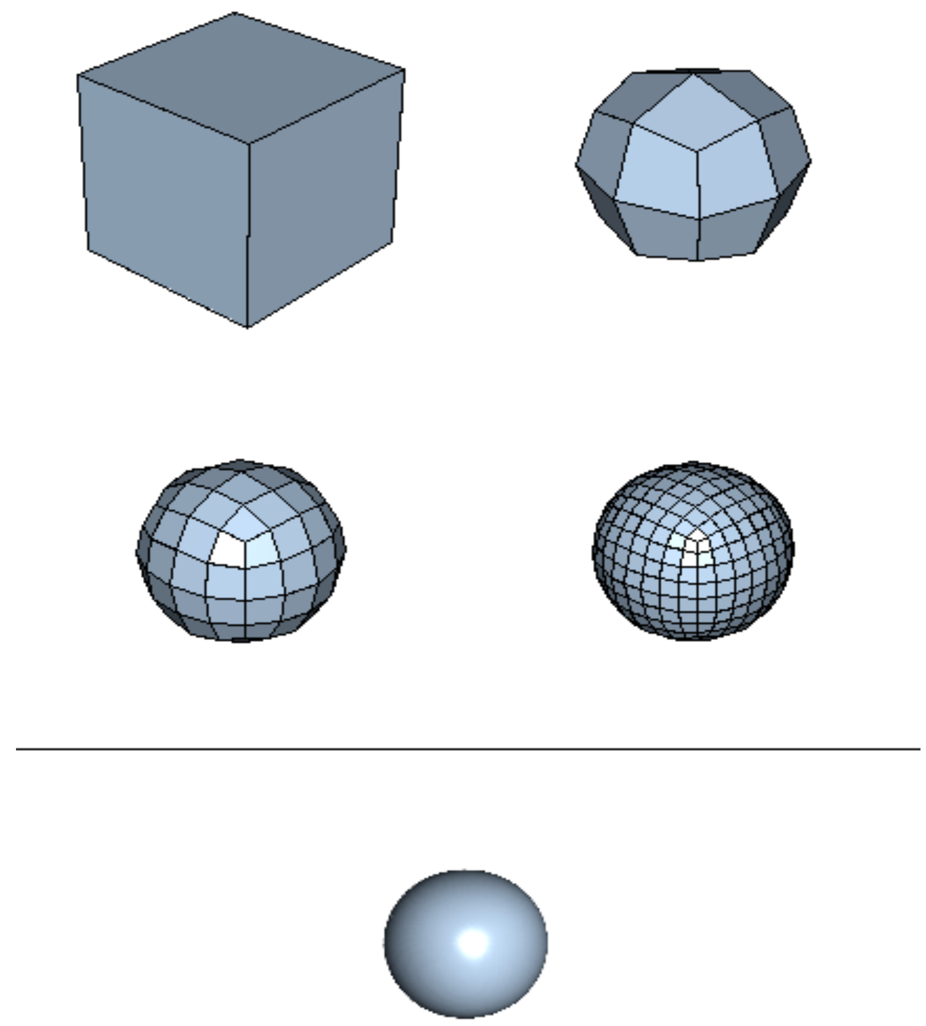


CS3621 Introduction to Computing with Geometry Notes

- http://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/

More complex objects?



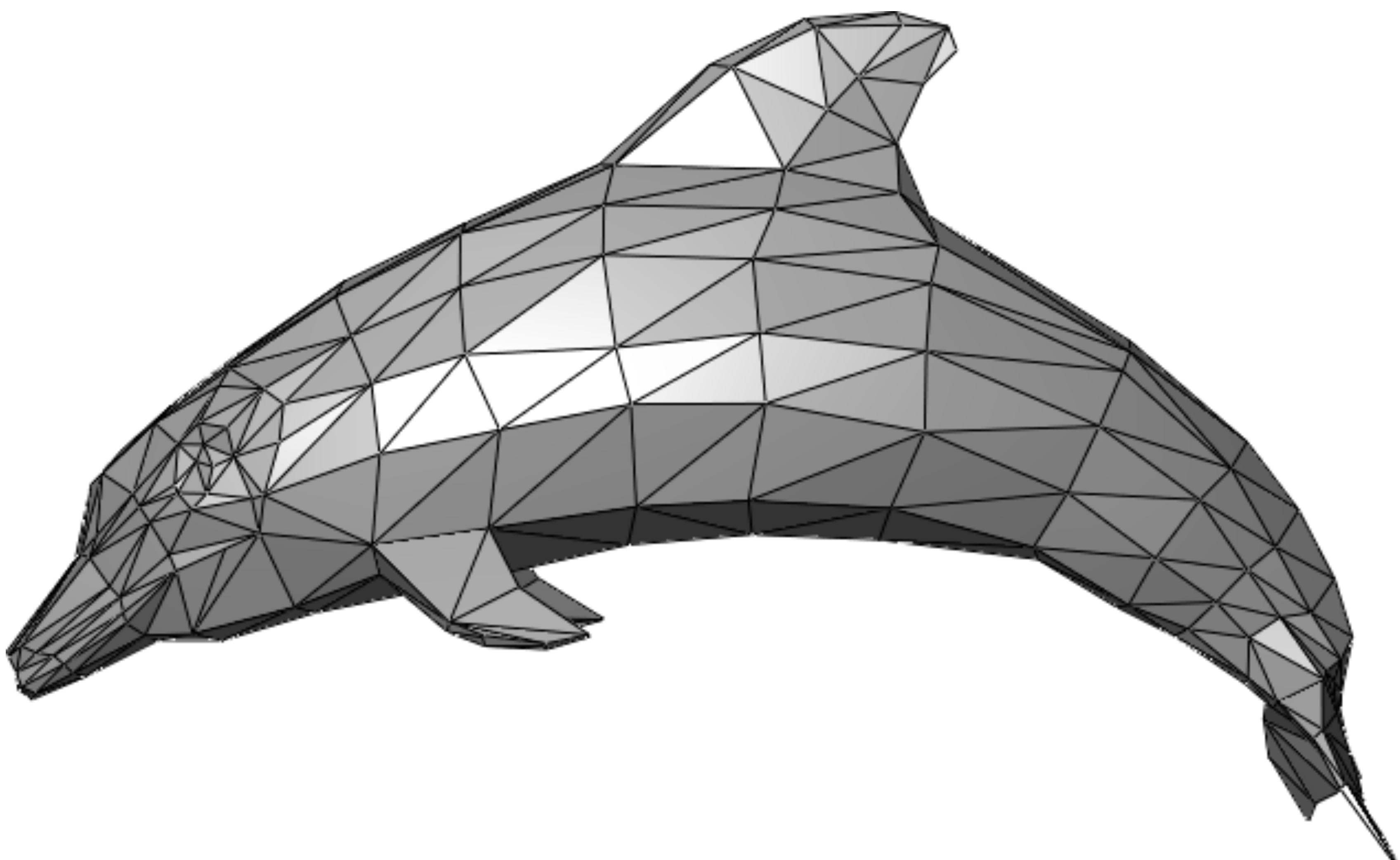


Challenge: surface continuity?

Polygonal Mesh

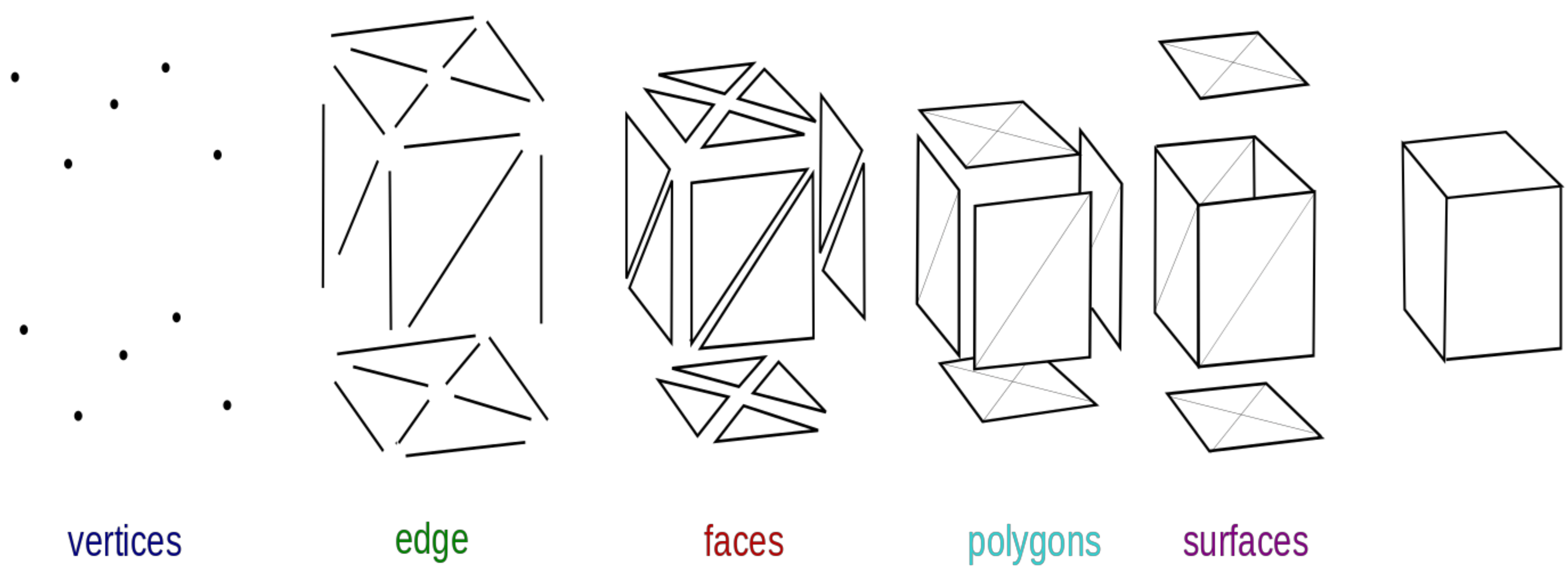
What is polygonal mesh?

- A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling



What is polygonal mesh?

- A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling

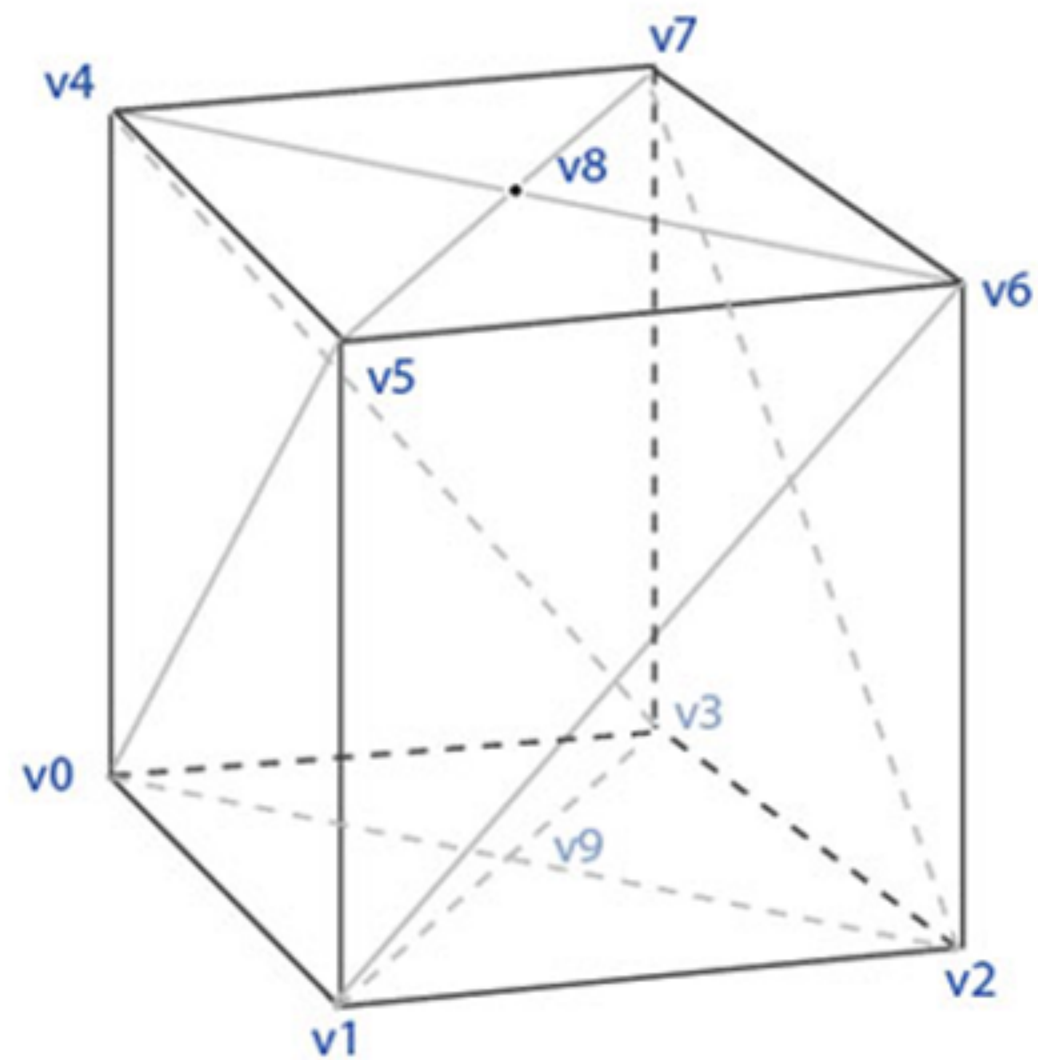


polygonal mesh representations

Vertex-Vertex Meshes (VV)

Vertex List

v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	.5,.5,1	v4 v5 v6 v7
v9	.5,.5,0	v0 v1 v2 v3



polygonal mesh representations

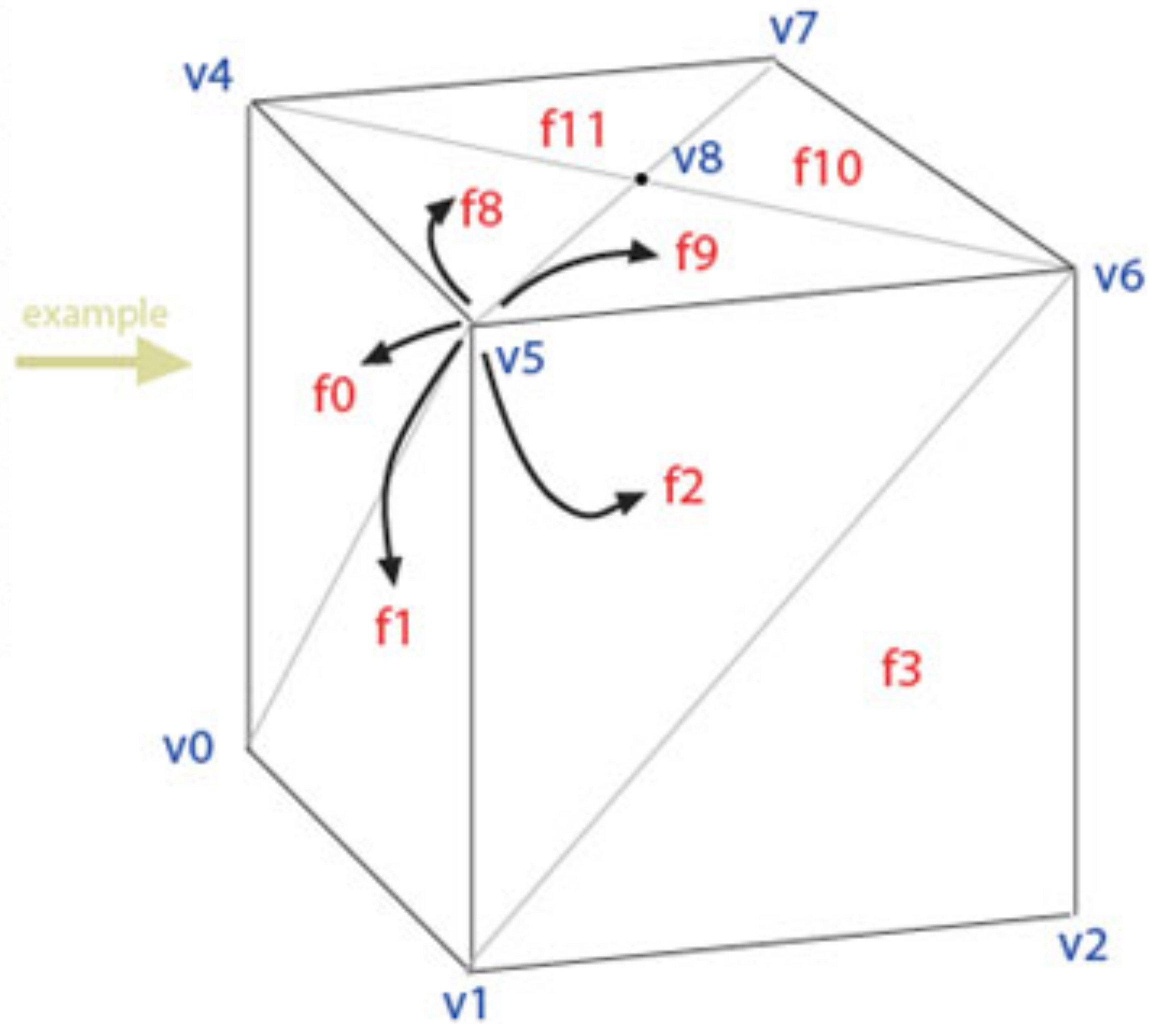
Face-Vertex Meshes

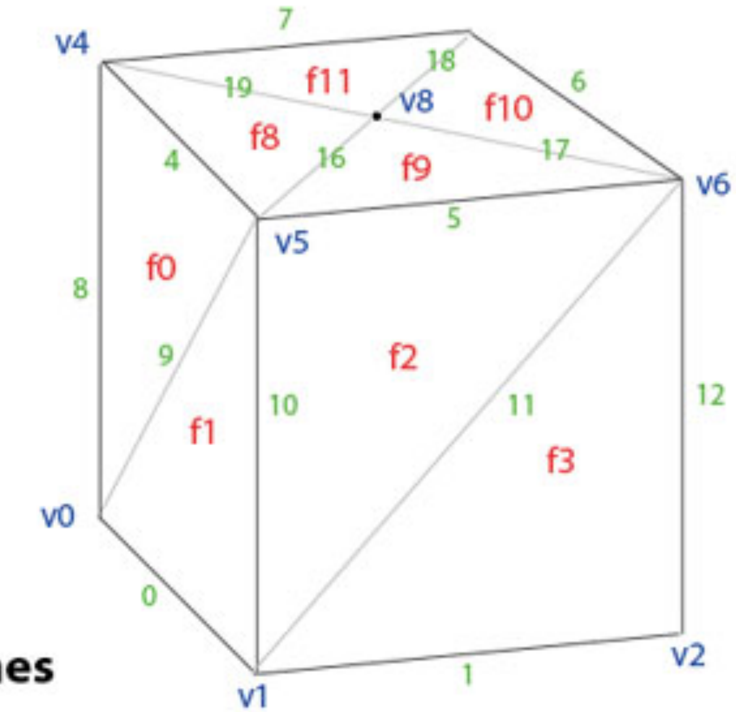
Face List

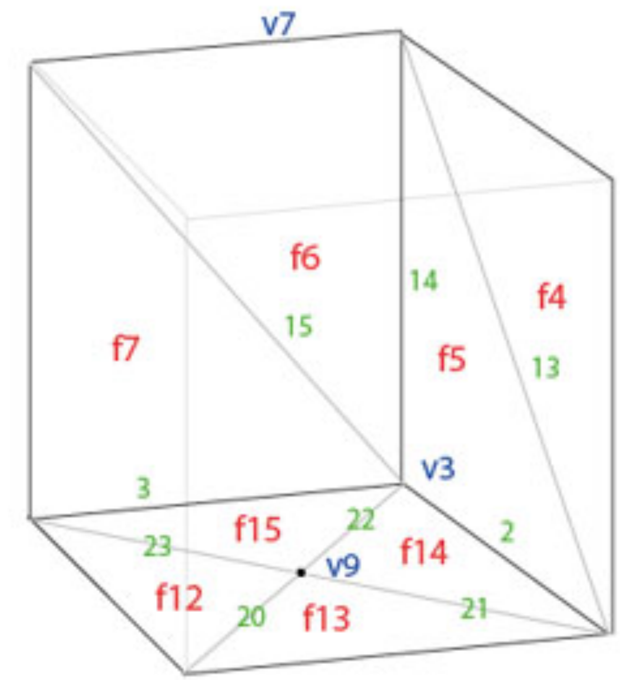
f0	v0 v4 v5
f1	v0 v5 v1
f2	v1 v5 v6
f3	v1 v6 v2
f4	v2 v6 v7
f5	v2 v7 v3
f6	v3 v7 v4
f7	v3 v4 v0
f8	v8 v5 v4
f9	v8 v6 v5
f10	v8 v7 v6
f11	v8 v4 v7
f12	v9 v5 v4
f13	v9 v6 v5
f14	v9 v7 v6
f15	v9 v4 v7

Vertex List

v0	0,0,0	f0 f1 f12 f15 f7
v1	1,0,0	f2 f3 f13 f12 f1
v2	1,1,0	f4 f5 f14 f13 f3
v3	0,1,0	f6 f7 f15 f14 f5
v4	0,0,1	f6 f7 f0 f8 f11
v5	1,0,1	f0 f1 f2 f9 f8
v6	1,1,1	f2 f3 f4 f10 f9
v7	0,1,1	f4 f5 f6 f11 f10
v8	.5,.5,0	f8 f9 f10 f11
v9	.5,.5,1	f12 13 14 15







Winged-Edge Meshes

Face List

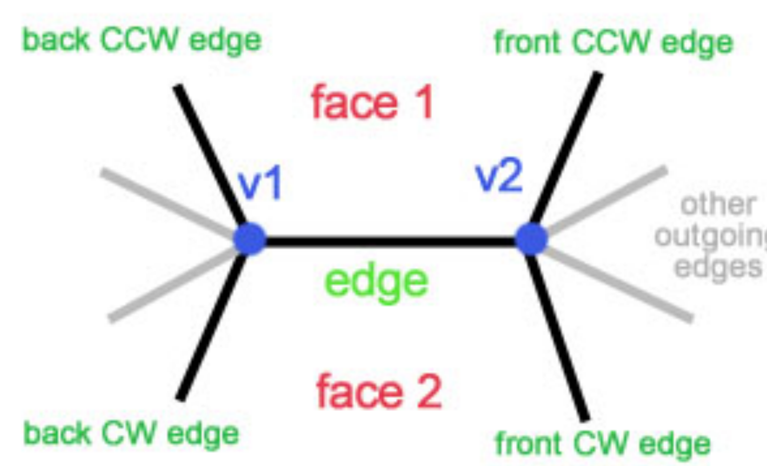
f0	4 8 9
f1	0 10 9
f2	5 10 11
f3	1 12 11
f4	6 12 13
f5	2 14 13
f6	7 14 15
f7	3 8 15
f8	4 16 19
f9	5 17 16
f10	6 18 17
f11	7 19 18
f12	0 23 20
f13	1 20 21
f14	2 21 22
f15	3 22 23

Edge List

e0	v0 v1	f1 f12	9 23 10 20
e1	v1 v2	f3 f13	11 20 12 21
e2	v2 v3	f5 f14	13 21 14 22
e3	v3 v0	f7 f15	15 22 8 23
e4	v4 v5	f0 f8	19 8 16 9
e5	v5 v6	f2 f9	16 10 17 11
e6	v6 v7	f4 f10	17 12 18 13
e7	v7 v4	f6 f11	18 14 19 15
e8	v0 v4	f7 f0	3 9 7 4
e9	v0 v5	f0 f1	8 0 4 10
e10	v1 v5	f1 f2	0 11 9 5
e11	v1 v6	f2 f3	10 1 5 12
e12	v2 v6	f3 f4	1 13 11 6
e13	v2 v7	f4 f5	12 2 6 14
e14	v3 v7	f5 f6	2 15 13 7
e15	v3 v4	f6 f7	14 3 7 15
e16	v5 v8	f8 f9	4 5 19 17
e17	v6 v8	f9 f10	5 6 16 18
e18	v7 v8	f10 f11	6 7 17 19
e19	v4 v8	f11 f8	7 4 18 16
e20	v1 v9	f12 f13	0 1 23 21
e21	v2 v9	f13 f14	1 2 20 22
e22	v3 v9	f14 f15	2 3 21 23
e23	v0 v9	f15 f12	3 0 22 20

Vertex List

v0	0,0,0	8 9 0 23 3
v1	1,0,0	10 11 1 20 0
v2	1,1,0	12 13 2 21 1
v3	0,1,0	14 15 3 22 2
v4	0,0,1	8 15 7 19 4
v5	1,0,1	10 9 4 16 5
v6	1,1,1	12 11 5 17 6
v7	0,1,1	14 13 6 18 7
v8	.5,.5,0	16 17 18 19
v9	.5,.5,1	20 21 22 23



Winged Edge Structure

Wavefront .obj file

```
# List of Vertices, with (x,y,z[,w]) coordinates, w is optional and defaults to 1.0.
  v 0.123 0.234 0.345 1.0
  v ...
  ...
  # Texture coordinates, in (u ,v [,w]) coordinates, these will vary between 0 and 1, w is optional and
default to 0.
  vt 0.500 1 [0]
  vt ...
  ...
  # Normals in (x,y,z) form; normals might not be unit.
.
  vn 0.707 0.000 0.707
  vn ...
  ...
  # Parameter space vertices in ( u [,v] [,w] ) form; free form geometry statement ( see below )
  vp 0.310000 3.210000 2.100000
  vp ...
  ...
  # Face Definitions (see below)
  f 1 2 3
  f 3/1 4/2 5/3
  f 6/4/1 3/5/3 7/6/5
  f ...
  ...
```

Wavefront .obj file

- Vertex positions
  - v, vt, vn
- Face definitions
  - f v1 v2 v3 v4 ...
  - f v1/vt1 v2/vt2 v3/vt3 ...
  - f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 ...
  - f v1//vn1 v2//vn2 v3//vn3 ...
- Referencing materials
  - mtllib [external .mtl file name]
  - usemtl [material name]

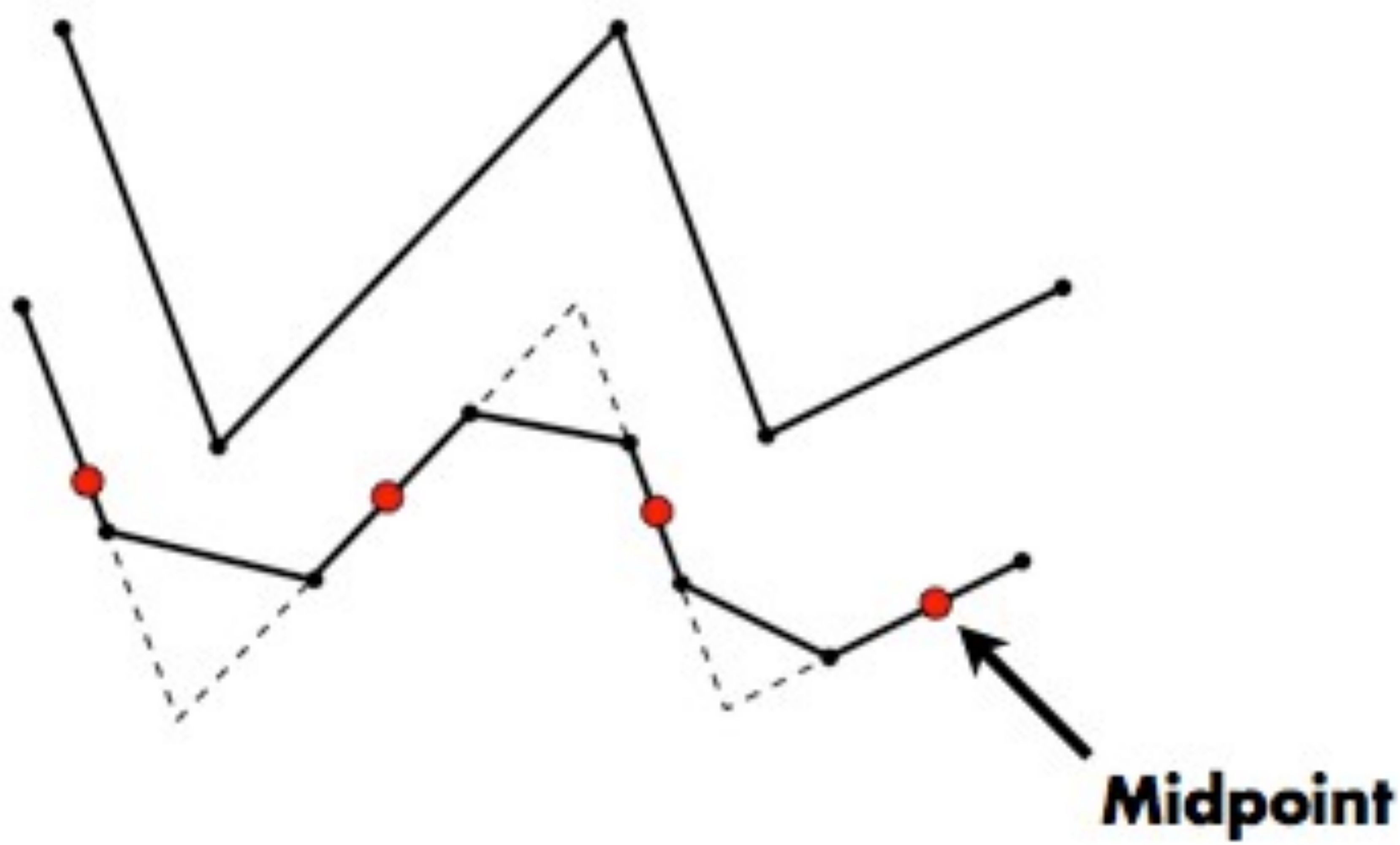
Wavefront .obj file

- Named objects and polygon groups are specified via the following tags.
  - o [object name]
  - g [group name]

- Smooth shading across polygons is enabled by smoothing groups.
  - s 1
  - ...
  - # Smooth shading can be disabled as well.
  - s off
  - ...

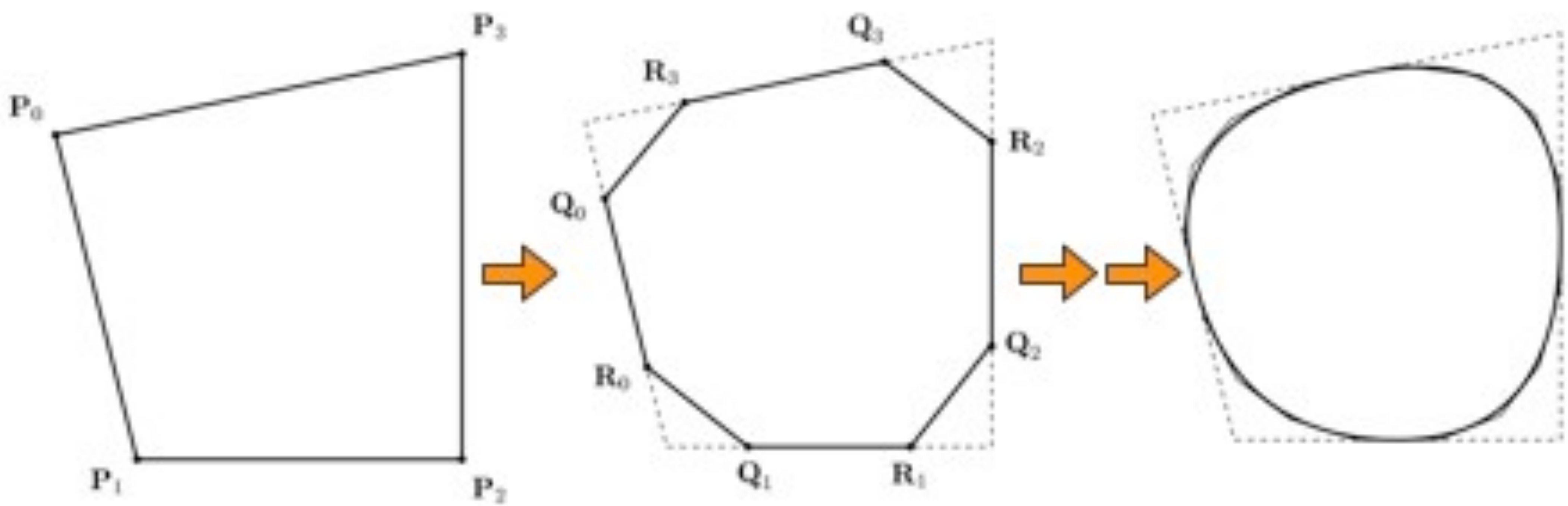
Subdivision surfaces

Corner Cutting Algorithm



Chaiken (1974)

Procedural Curve



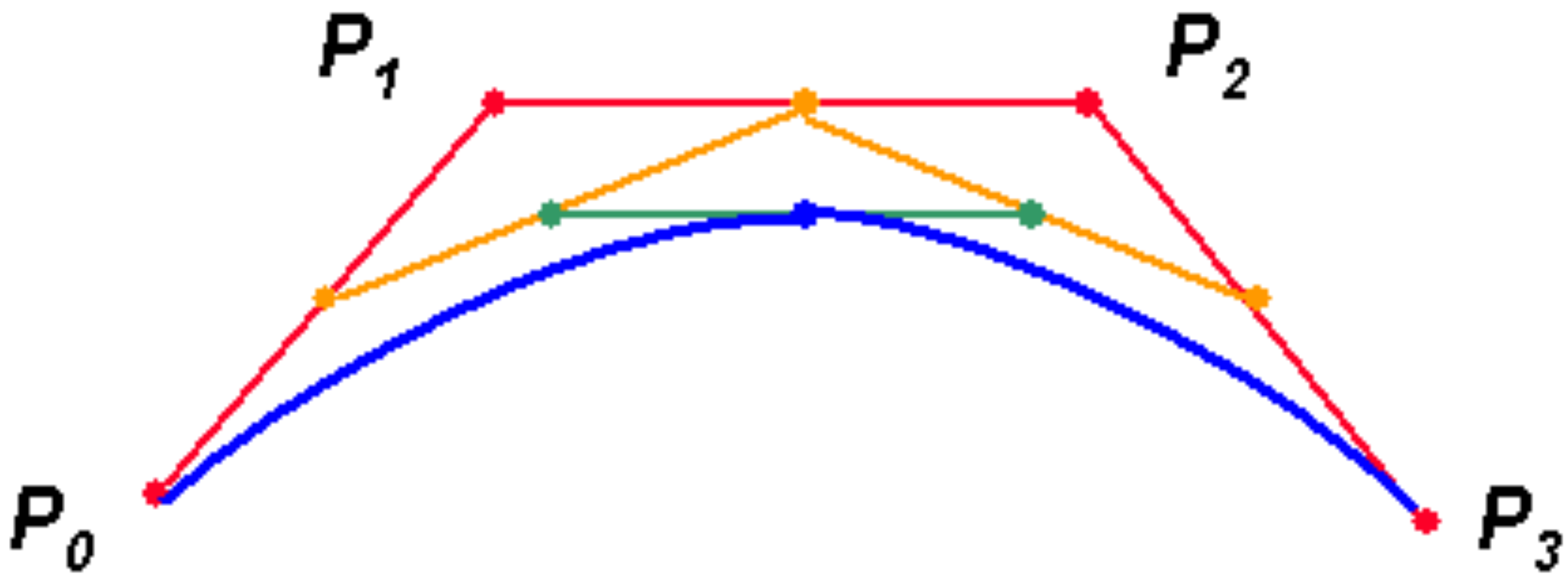
Repeatedly cutting corners generates a limit curve

1. Interpolates midpoints
2. Tangent preserved at midpoints

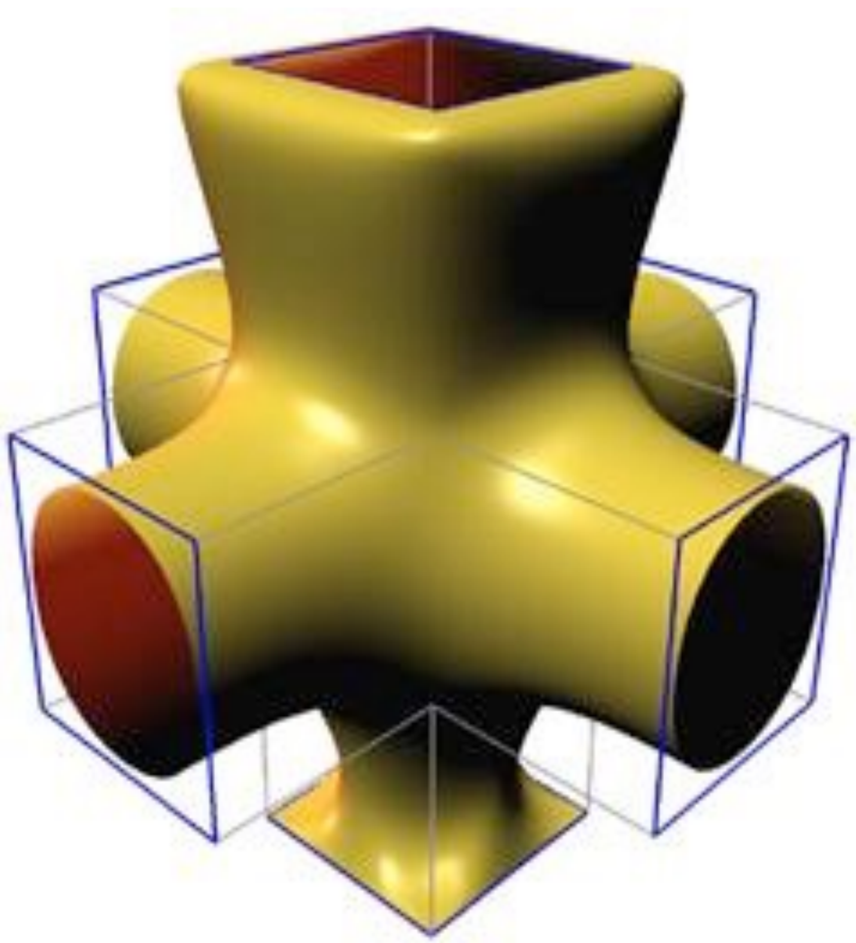
subdivision surface

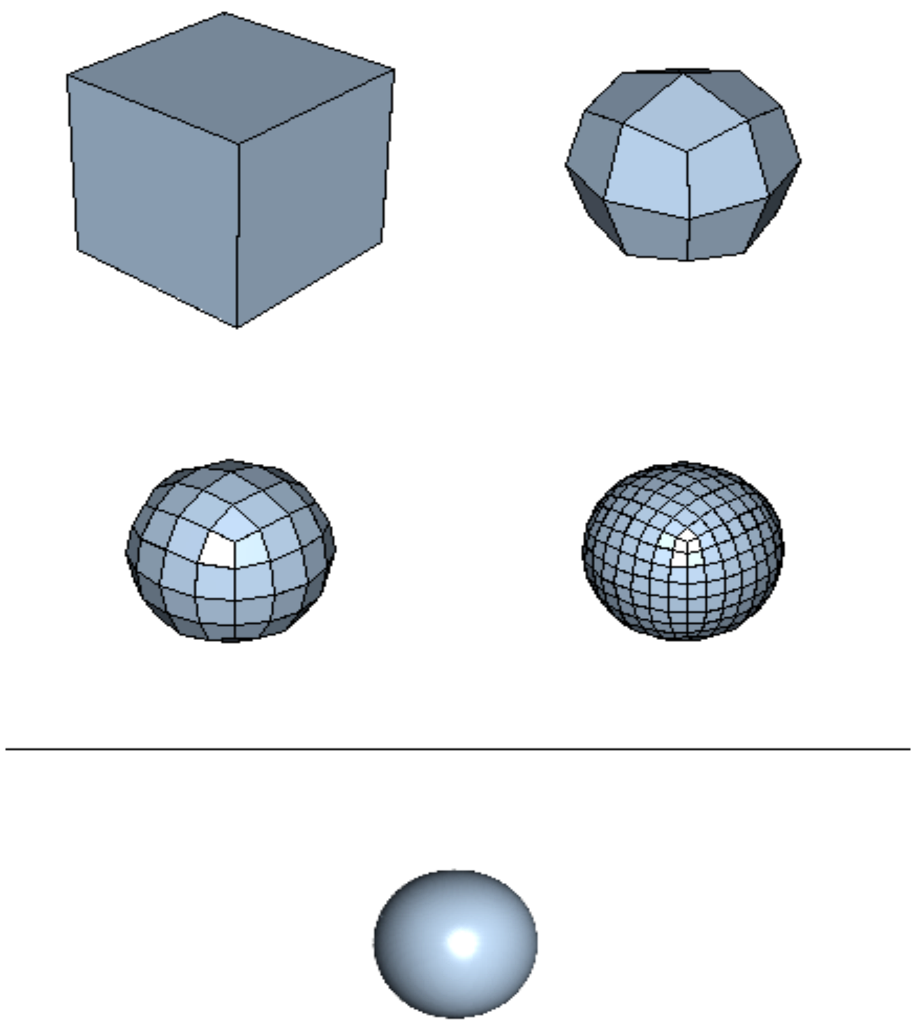
subdivision curves:

- starting from a set of points, generate new points in every step under some rules, when such step goes on infinitely, the points will be convergent to a smooth curve.



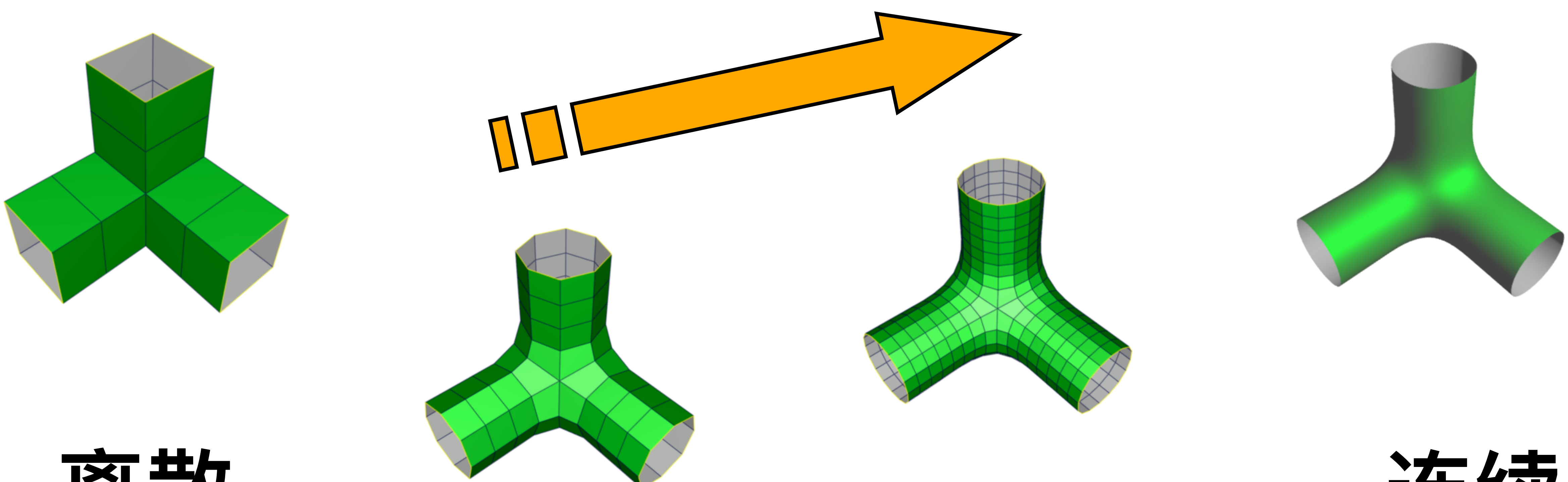
subdivision surface





细分曲面的基本概念

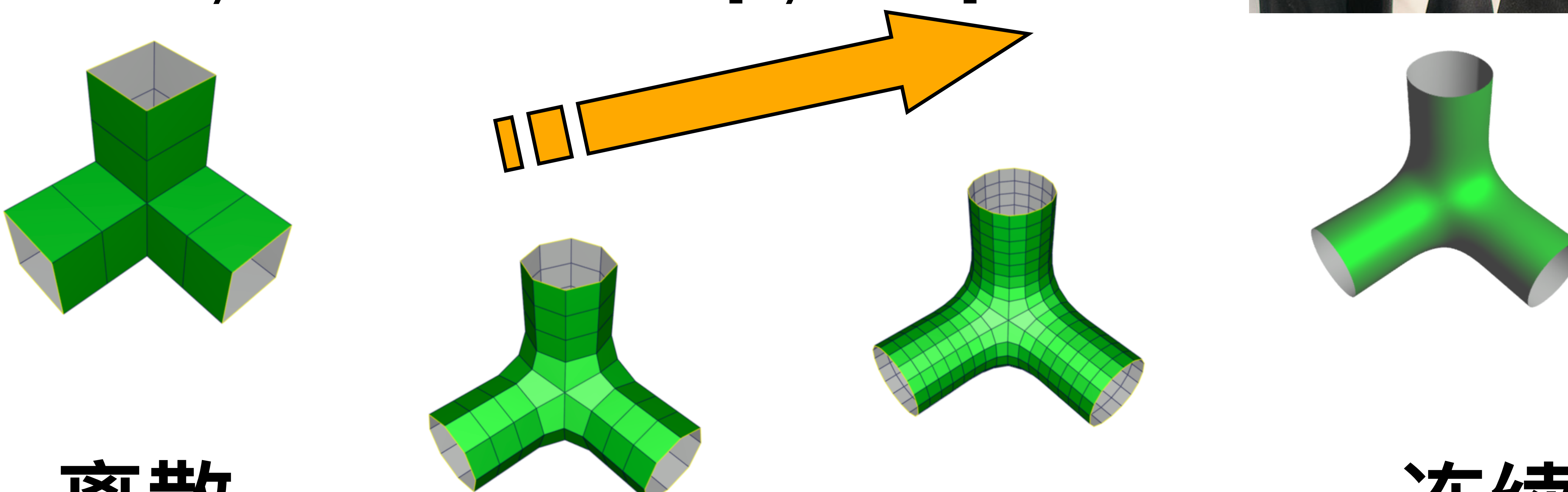
- 是一种过程式（迭代）的曲面构造方法
  - 输入： 多边形网格 => 控制网格
  - 输出： a. 加密网格 => 有限次迭代 [用于绘制]
    b. 连续曲面 => 极限曲面 [用于分析]



离散 连续

Classical subdivision schemes

- Catmull-Clark surface. [Catmull 1978]
- Doo-Sabin surface. [Doo 1978]
- Loop's subdivision schemes. [Loop 1987]
- Butterfly subdivision schemes. [Dyn 1990]



离散 连续

advantage of subdivision surface

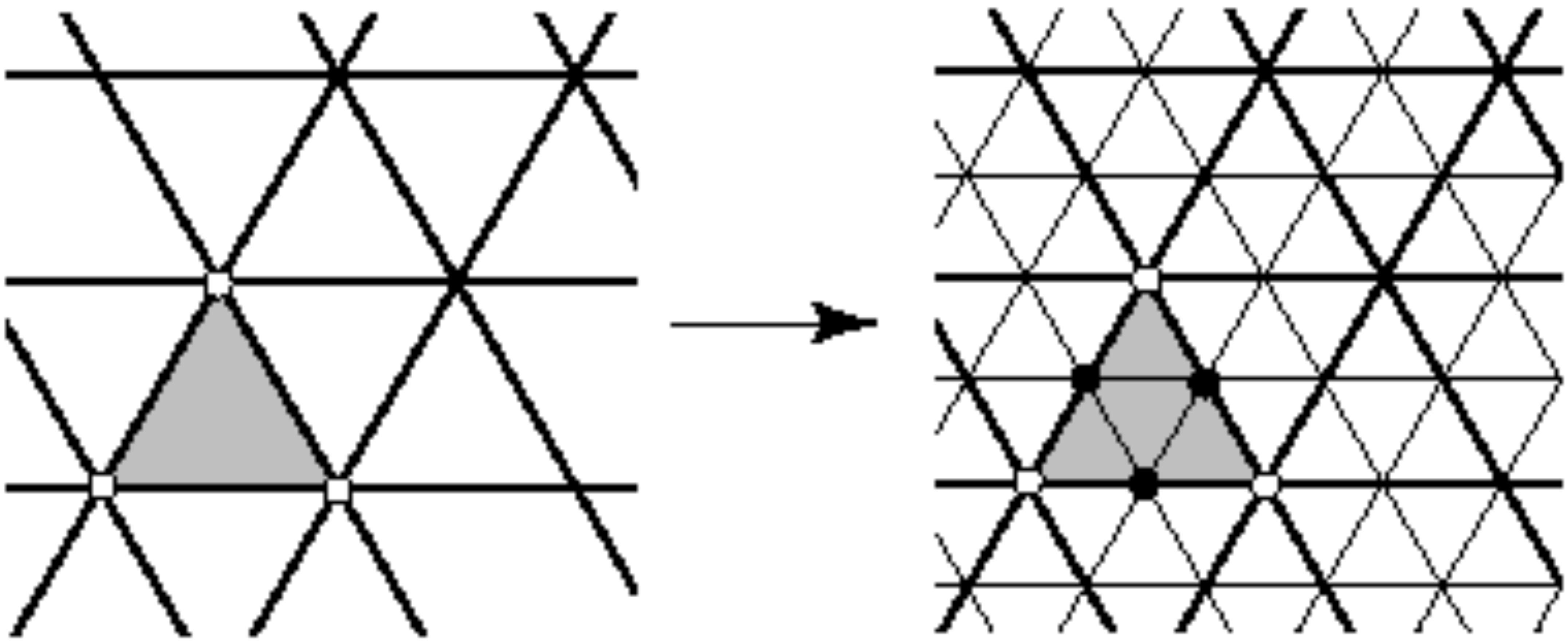
- **topologically complex shape**
- **stable;**
- **easy to implement;**
- **no need to merge between th surfaces**
- **LOD**

细分曲面的基本步骤

- 细分操作：
  - –拓扑规则：加密采样，重新构造网格
  - –几何规则：光顺网格

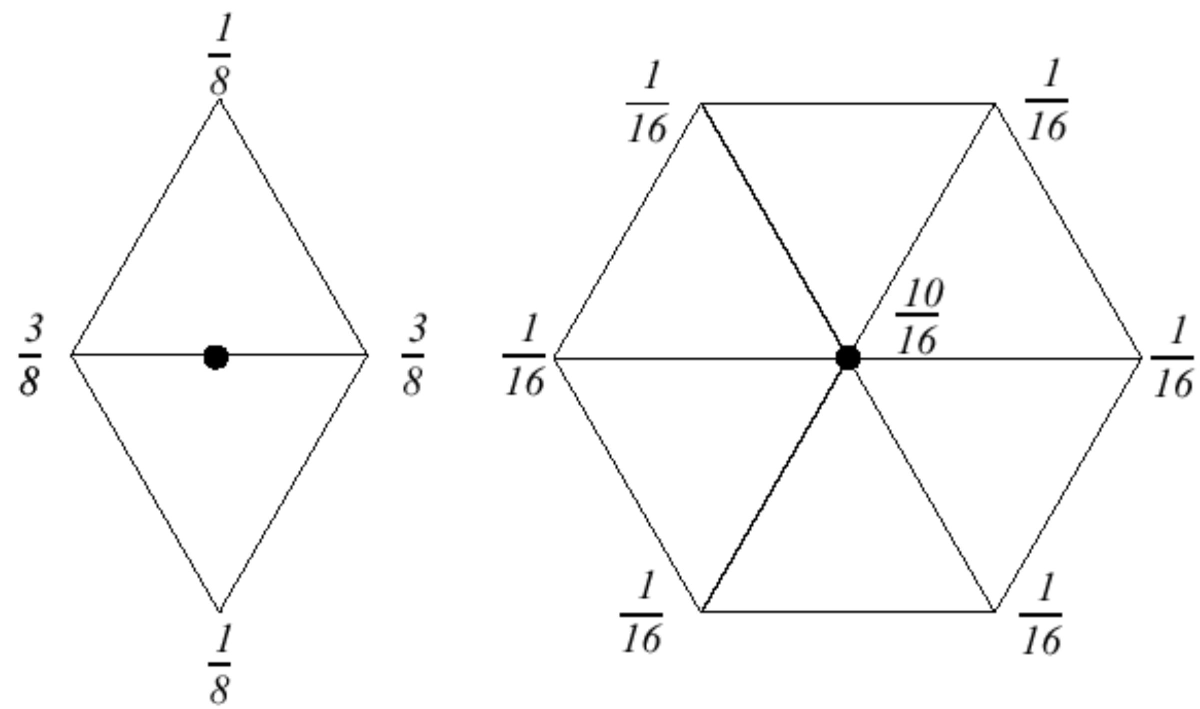
Loop subdivision surface

- 拓扑规则：怎样加密三角网格?
  - –对分每条边，并连接新顶点
  - –将每个三角形剖分成四个更小的三角形



Loop subdivision surface

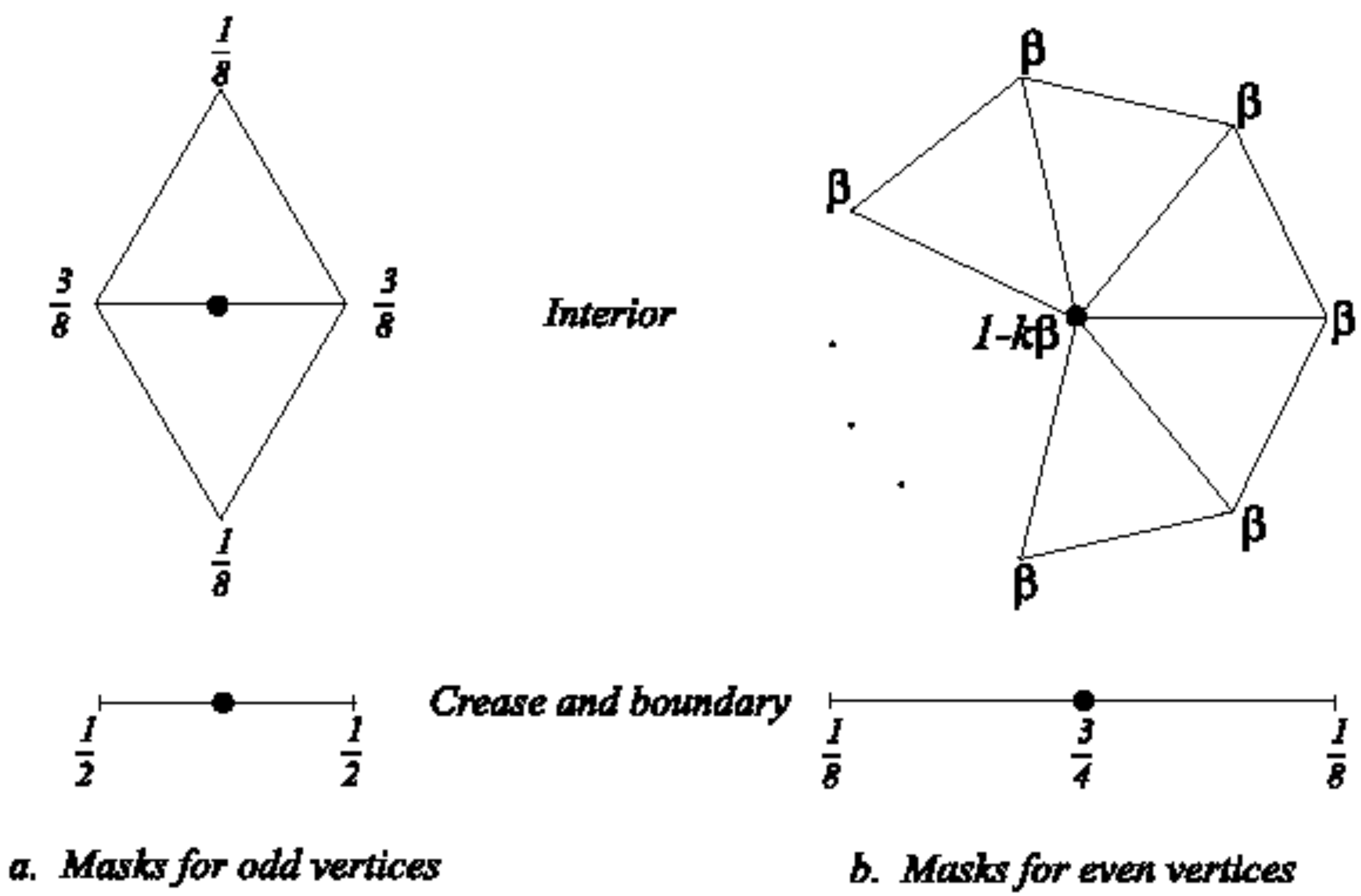
- 几何规则：怎样放置新顶点的位置?
  - –利用原始网格中与新顶点相邻的顶点来做**加权平均**



如果顶点的连接度不是6该怎么办?

Loop subdivision surface

- 几何规则：怎样放置新顶点的位置?
  - –奇异顶点处的规则



Loop subdivision surface

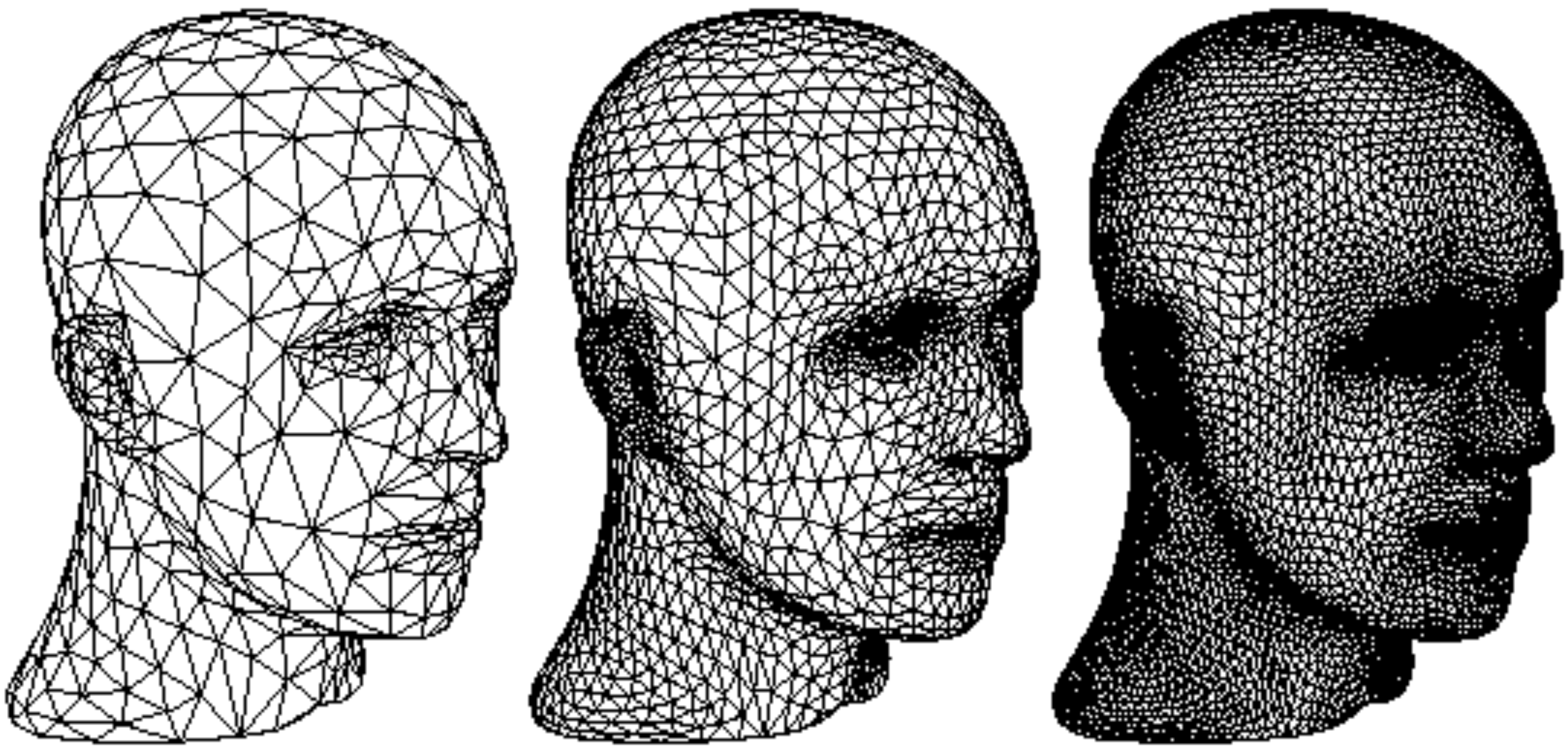
- 怎样选择系数Beta?
  - –分析极限曲面性质
  - –与曲面的连续性和光滑性相关
  - –需要计算相关（细分系数）矩阵的特征结构
    - » Original Loop

$$\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right)$$

    - » Warren

$$\beta = \begin{cases} \frac{3}{8n} & n > 3 \\ \frac{3}{16} & n = 3 \end{cases}$$

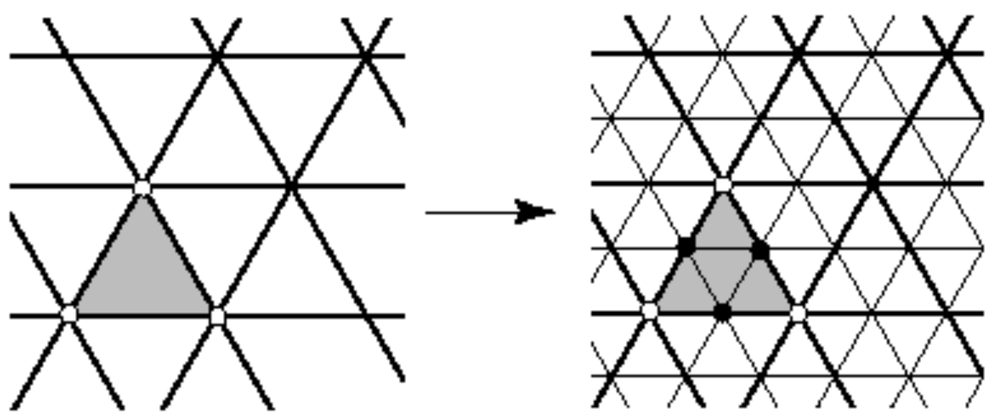
Loop subdivision surface



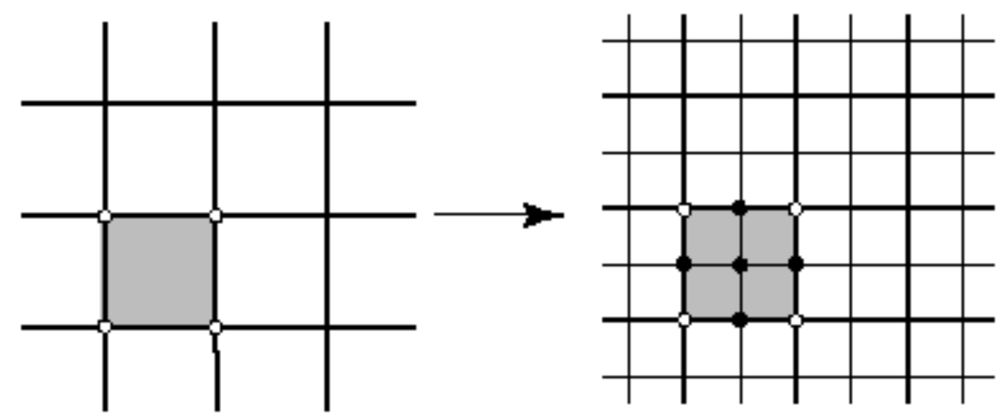
细分结果可达到较好的连续性光顺性！

Subdivision schemes

- 各种不同的细分方法
  - 不同的处理拓扑加密方法
  - 不同的布置顶点位置方法
    - 逼近 vs. 插值



Face split for triangles

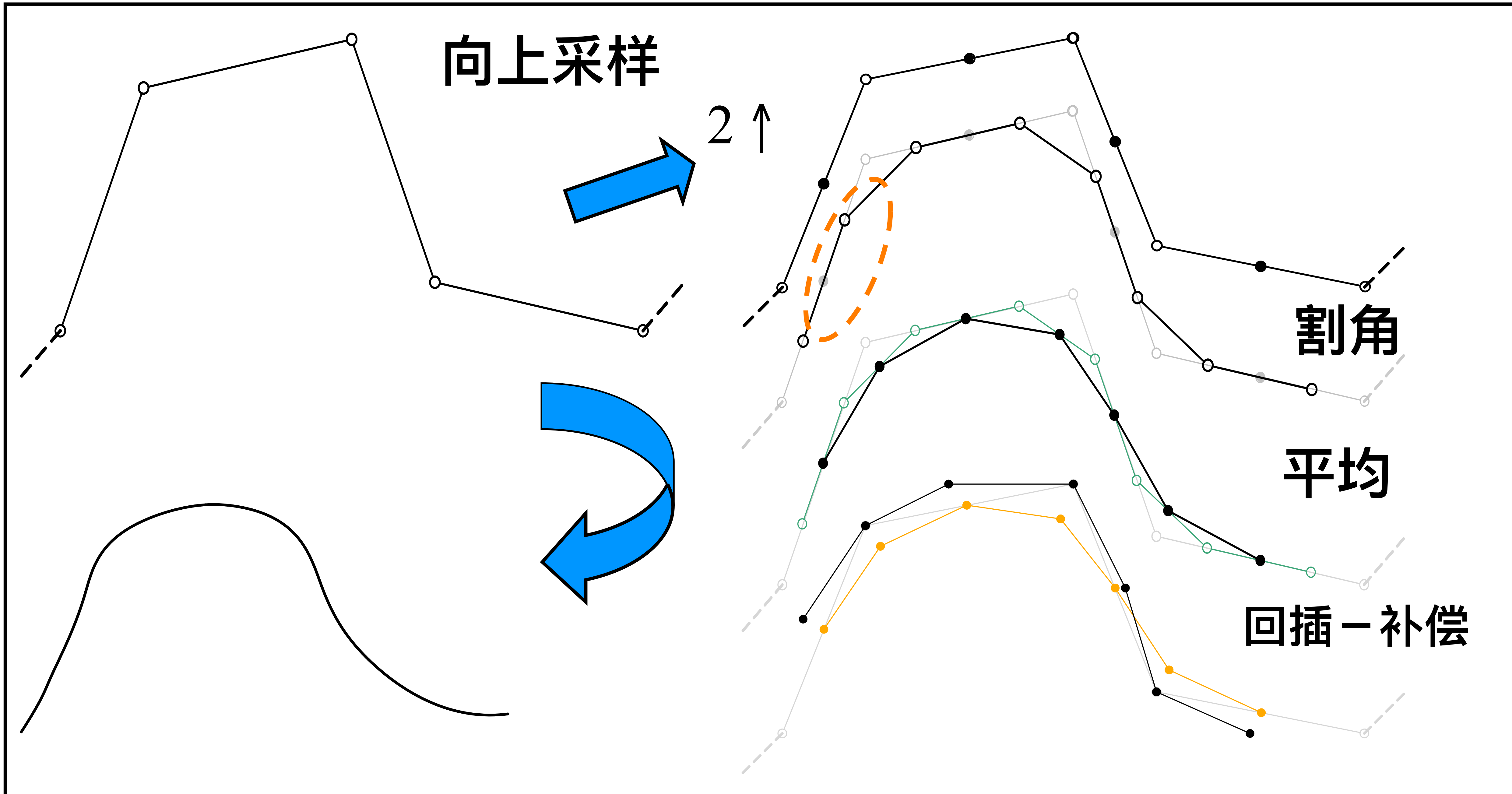


Face split for quads

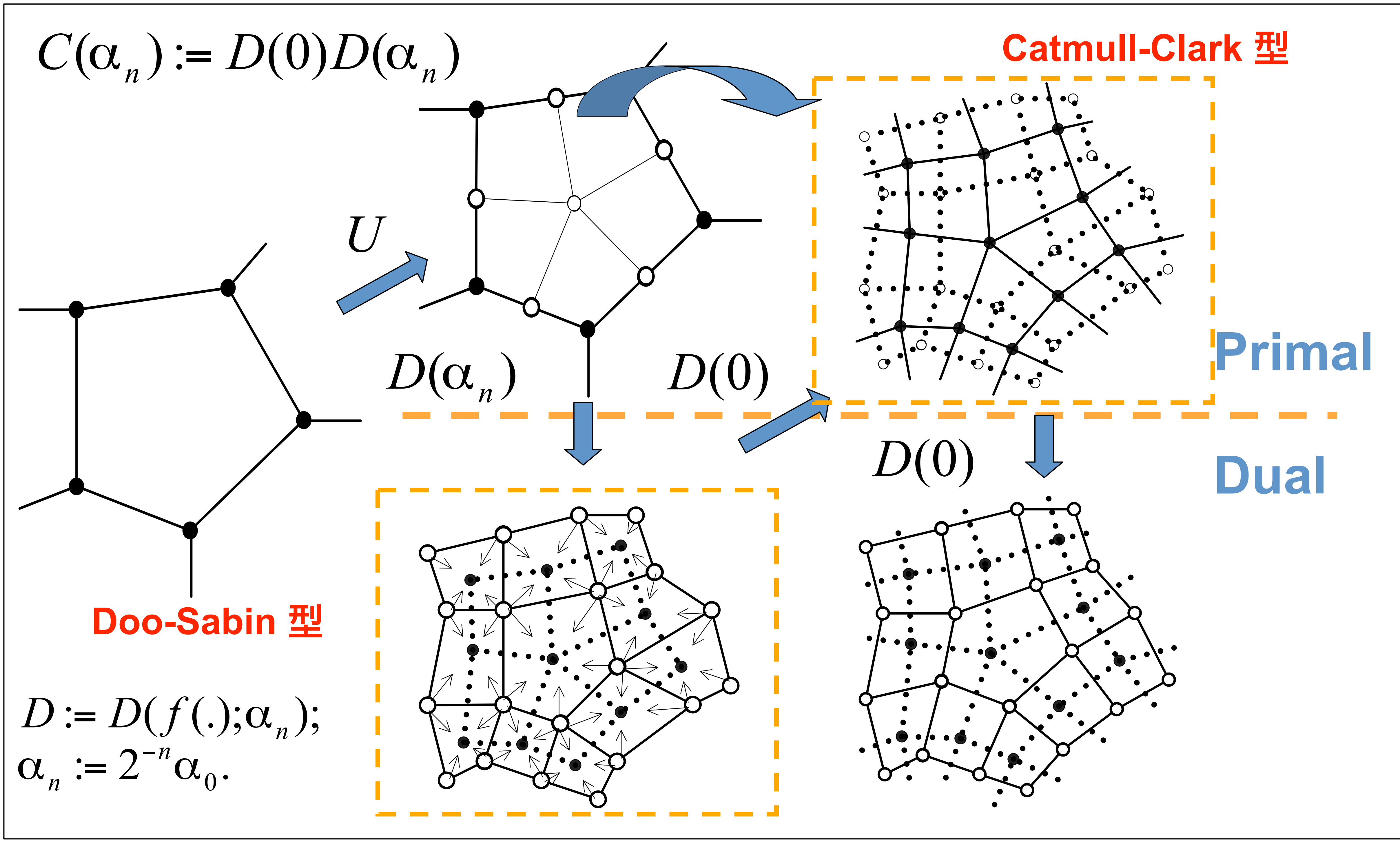
Face split		
	Triangular meshes	*Quad. meshes*
Approximating	Loop (C^2)	Catmull-Clark (C^2)
Interpolating	Mod. Butterfly (C^1)	Kobbelt (C^1)

Vertex split
Doo-Sabin, Midedge (C^1)
Biquartic (C^2)

细分曲面的基本思想

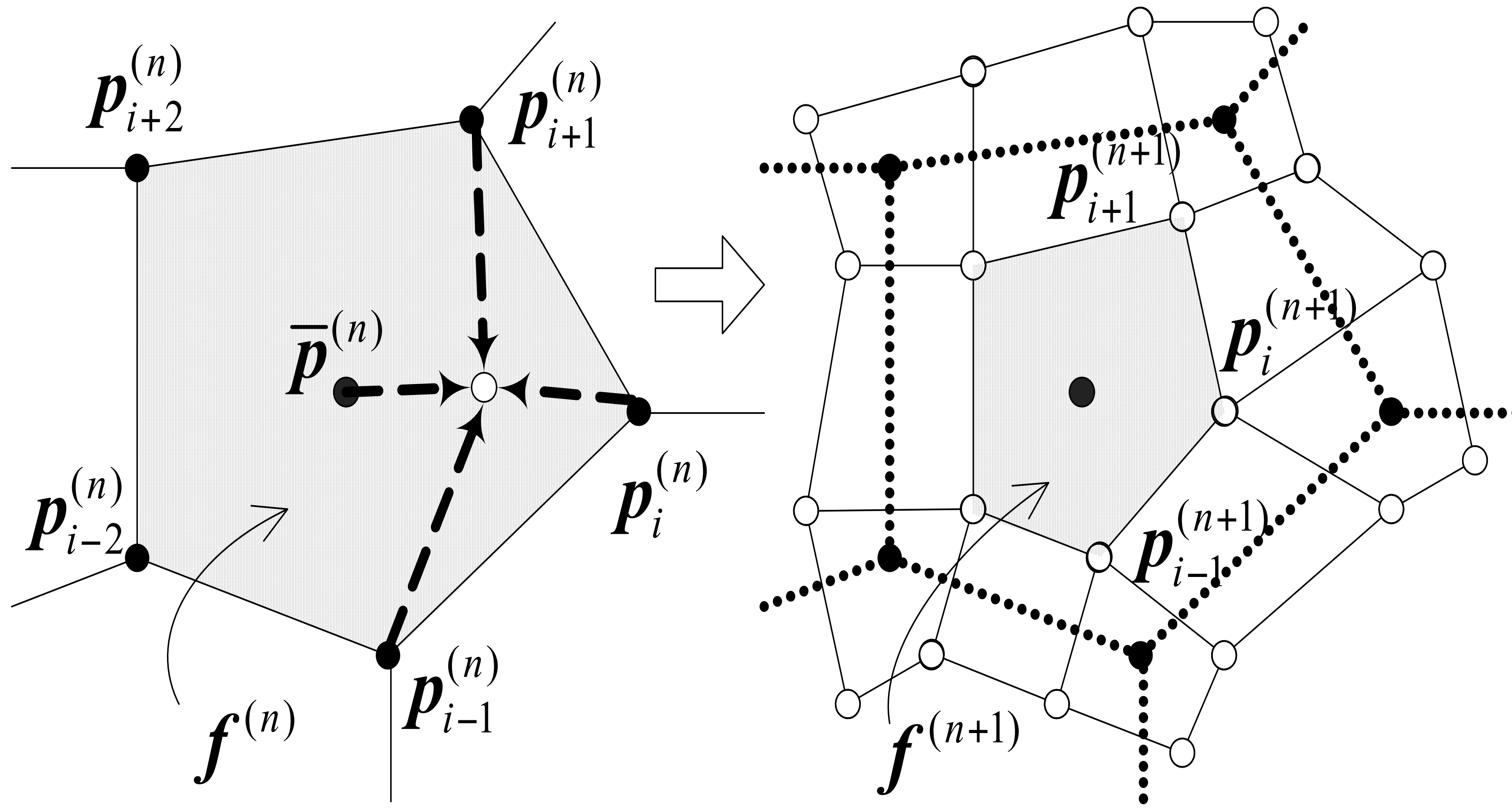


四边网格上的拓扑规则



$C(\alpha_n) := D(0)D(\alpha_n)$
Catmull-Clark 型
U
$D(\alpha_n)$
$D(0)$
Primal
Dual
$D(0)$
Doo-Sabin 型
$D := D(f(.);\alpha_n);$
$\alpha_n := 2^{-n}\alpha_0.$

Doo-Sabin型细分曲面



$$p_i^{(n+1)} = \frac{f_n}{(1+f_n)} p_i^{(n)} + \frac{f_n}{2(1+f_n)^2}\left(p_{i-1}^{(n)} + p_{i+1}^{(n)}\right) + \frac{1}{N(1+f_n)^2}\sum_{i=1}^{N} p_i^{(n)}.$$

subdivision surface

Catmull-Clark and Doo-Sabin subdivision

start from

$$P^i = \left(\cdots, p_{-1}^i, p_0^i, p_1^i, p_2^i, \cdots\right)$$

Catmull-Clark rules

$$p_{2j}^{i+1} = \frac{1}{8} p_{j-1}^i + \frac{6}{8} p_j^i + \frac{1}{8} p_{j+1}^i$$

$$p_{2j+1}^{i+1} = \frac{4}{8} p_j^i + \frac{4}{8} p_{j+1}^i$$

Doo-Sabin rules:

$$p_{2j}^{i+1} = \frac{3}{4} p_j^i + \frac{1}{4} p_{j+1}^i$$

$$p_{2j+1}^{i+1} = \frac{1}{4} p_j^i + \frac{3}{4} p_{j+1}^i$$

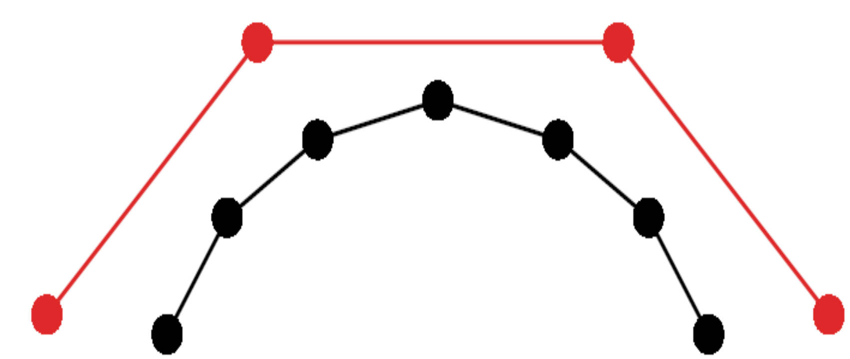
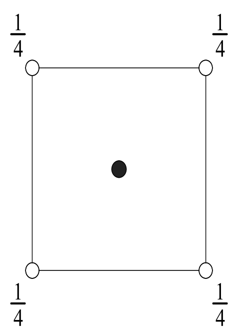
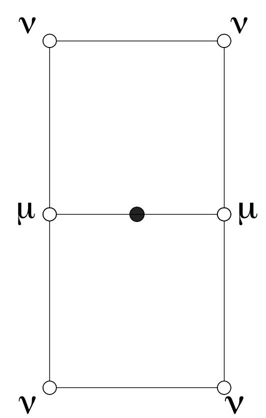


Figure 3: Subdividing an initial set of control points (upper, red) results in additional control points (lower, black), that more closely approximate a curve.

Catmull-Clark subdivision

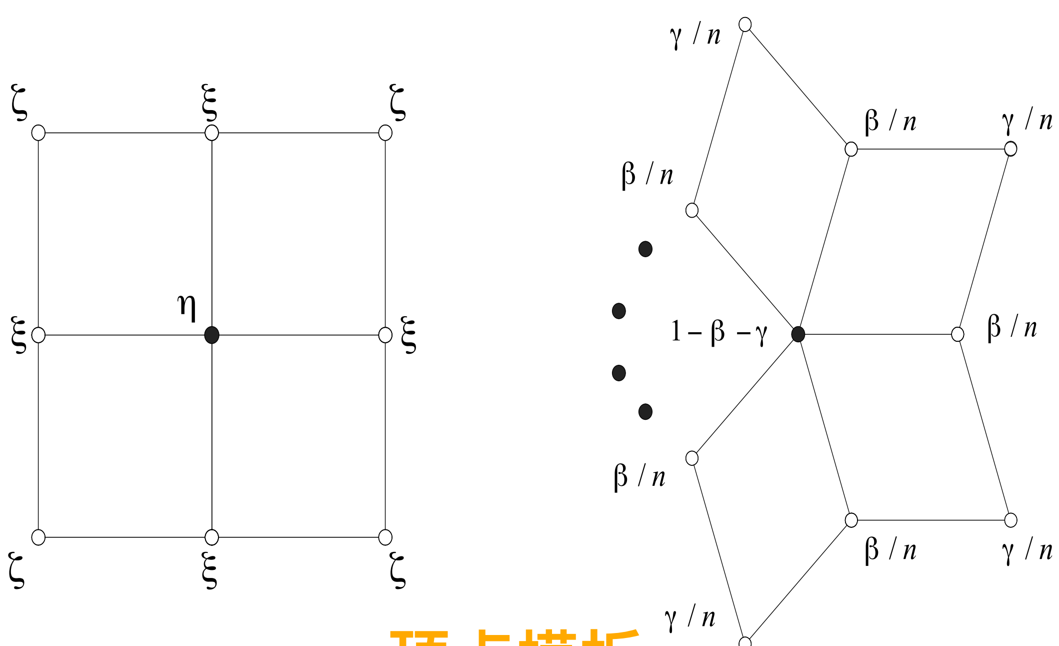


面点模板



边点模板

$$\begin{cases} f_j = f(2^{-j-1}\alpha), & \\ \mu_j = \dfrac{2f_j+1}{4f_j+4}, & \nu_j = \dfrac{1}{8f_j+8}, \\ \eta_j = \dfrac{(2f+1)^2}{(2f+2)^2}, & \xi_j = \dfrac{4f+2}{(4f+4)^2}, \quad \zeta_j = \dfrac{1}{(4f+4)^2}, \\ \beta_j = 4\xi_j, & \gamma_j = 4\zeta_j. \end{cases}$$



顶点模板

subdivision surface

Catmull-Clark subdivision surface:

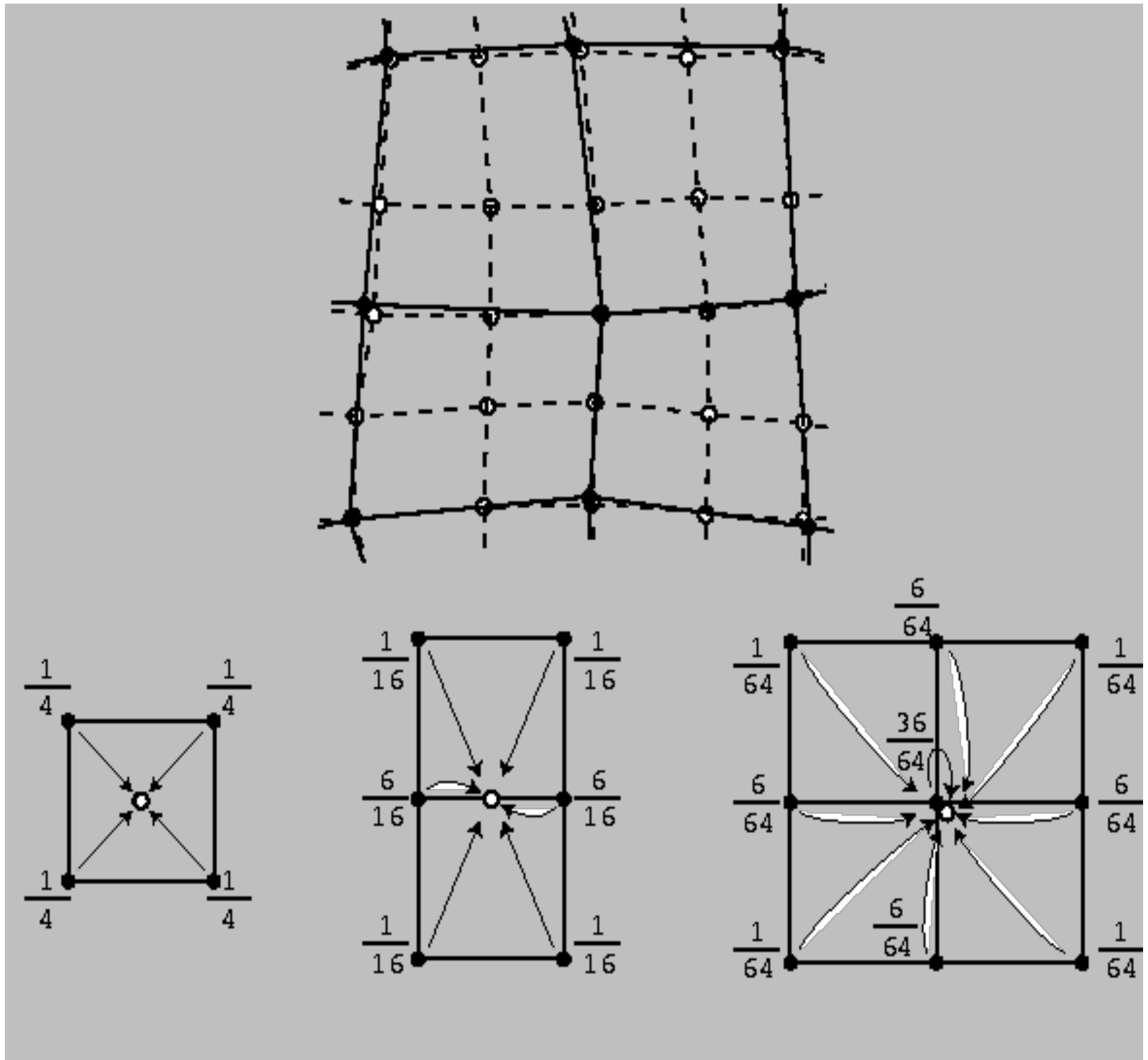
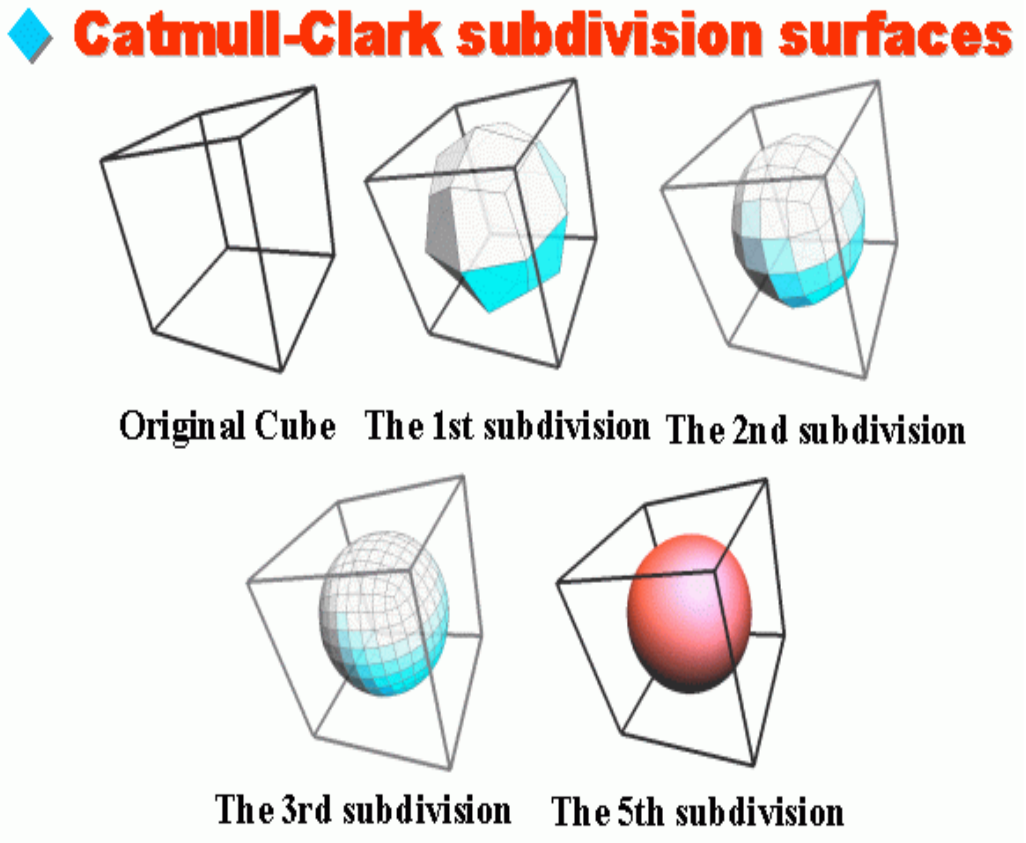


Figure 3: Catmull-Clark subdivision. Above a mesh (solid dots and solid lines) that has been refined (open dots and dashed lines). Below the weights used to generated each type of refined vertex: centre, edge, and modified old vertex.



subdivision surface

Doo-Sabin surface

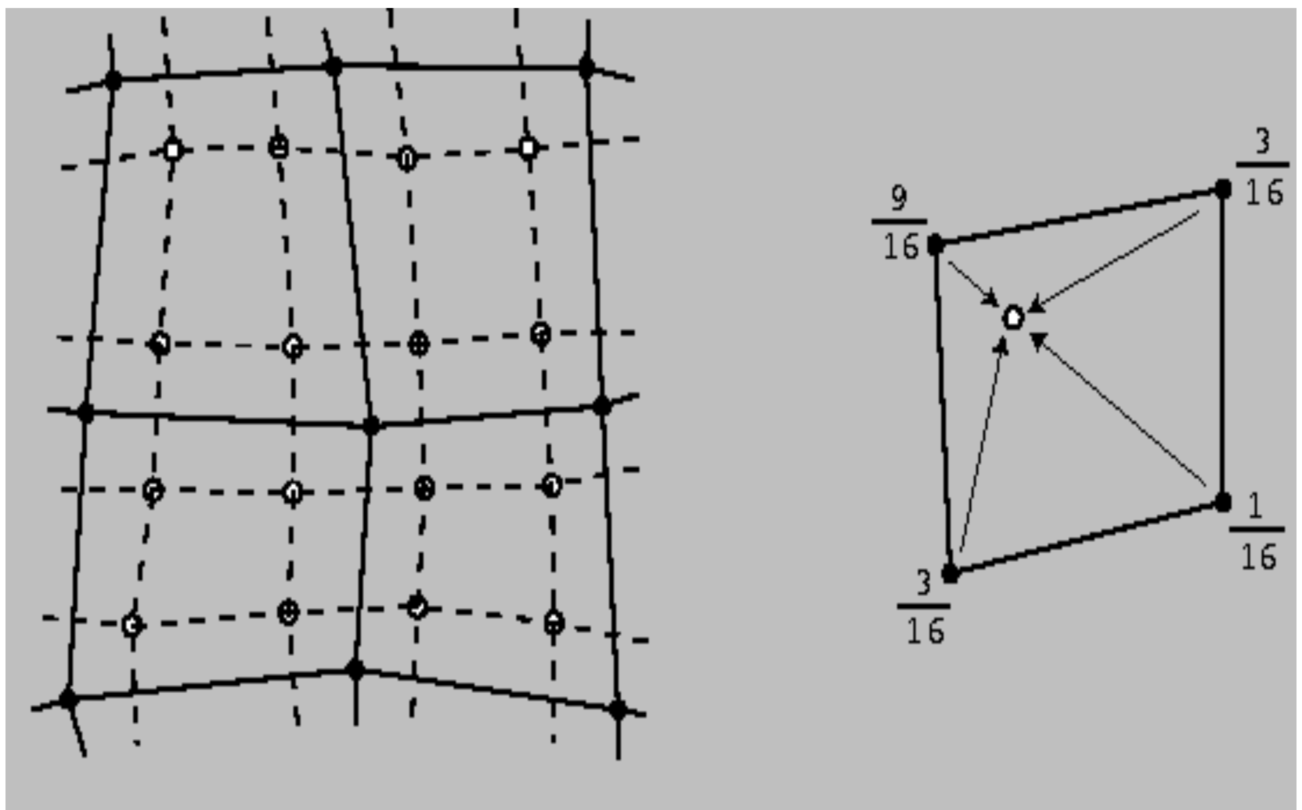
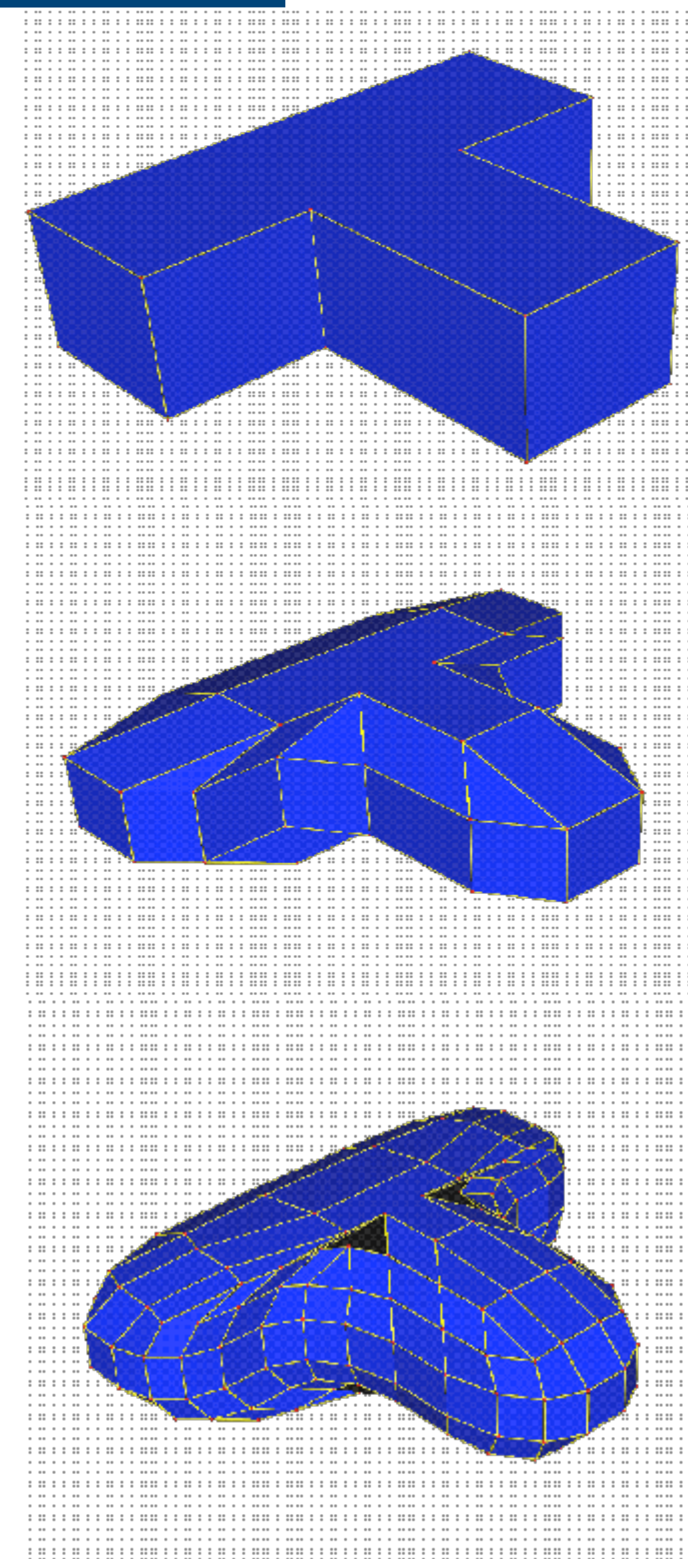
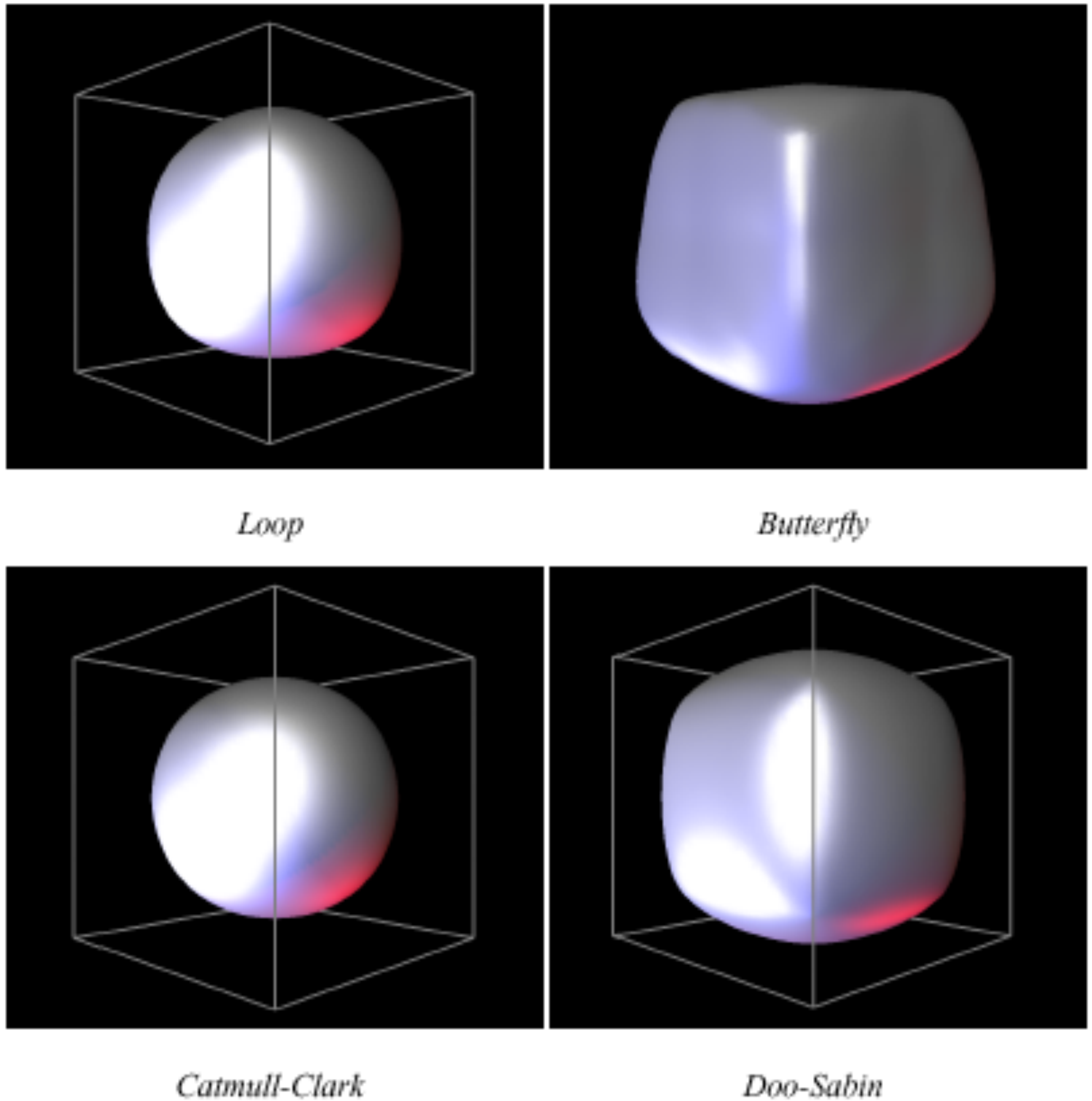


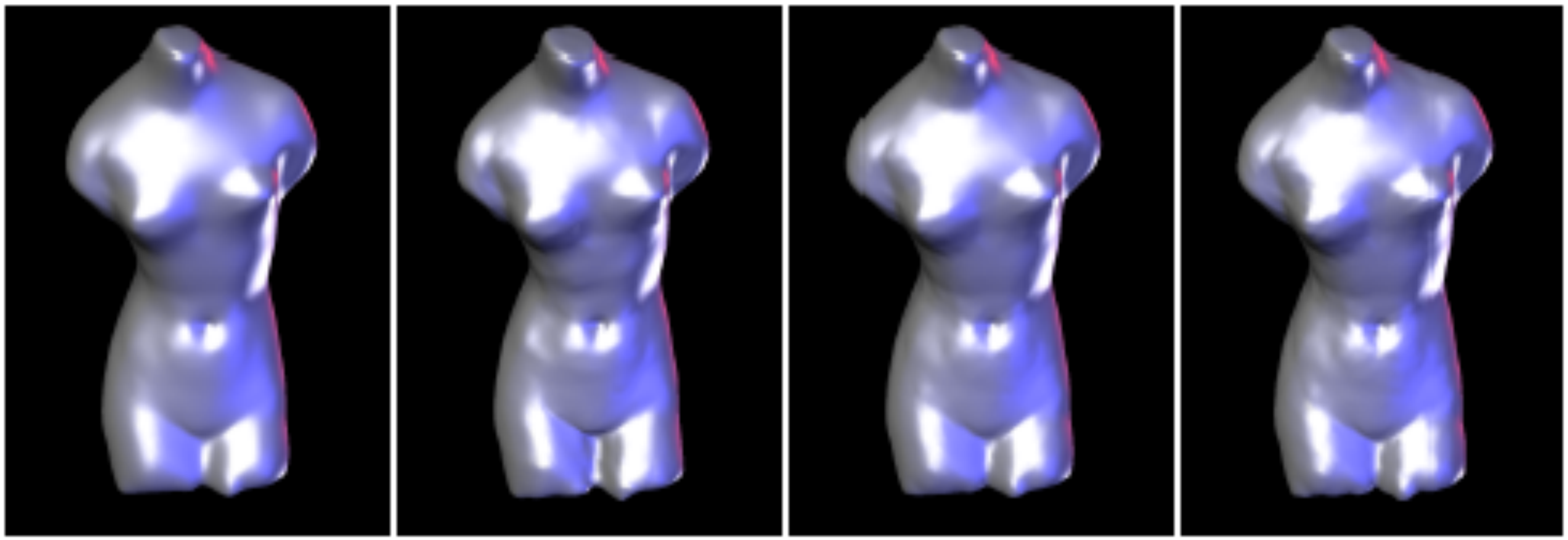
Figure 2: Doo-Sabin subdivision. On left a mesh (solid dots and solid lines) that has been refined (open dots and dashed lines). At right the weights used to generated one of the refined vertices.



不同细分曲面的比较



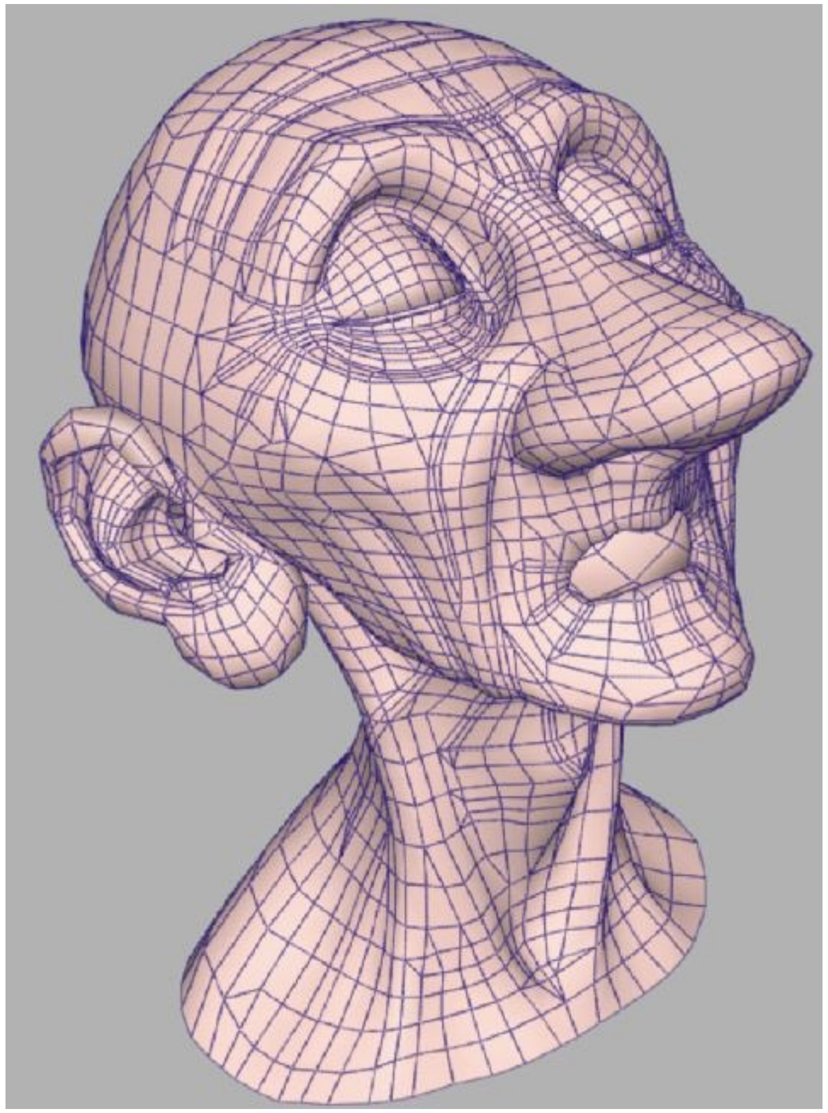
不同细分曲面的比较



Loop *Butterfly* *Catmull-Clark* *Doo-Sabin*

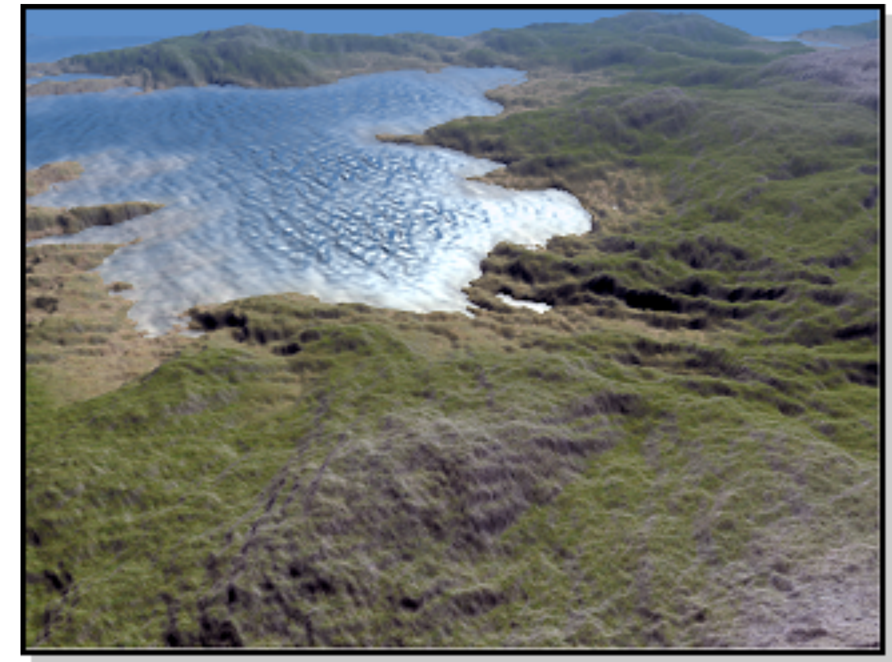
参考资料

- [普林斯顿大学的课件]
- [半静态回插细分]

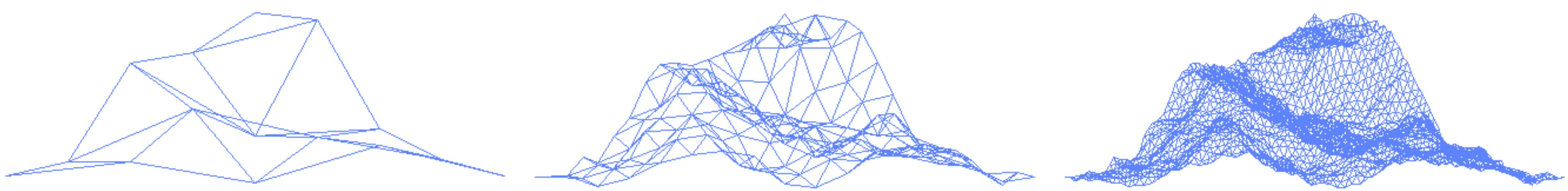


From Pixar, Geri’s Game

Geri's Game (youku link)



subdivision + random

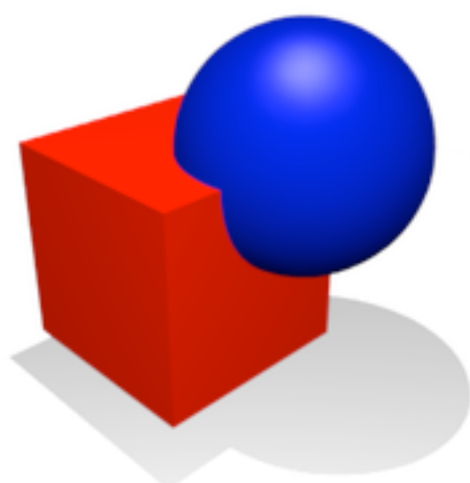


terrain generation

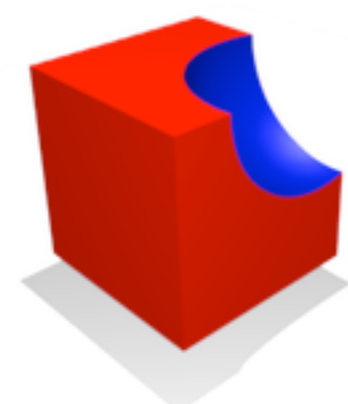
terrain generation

More complex 3D

Boolean operation



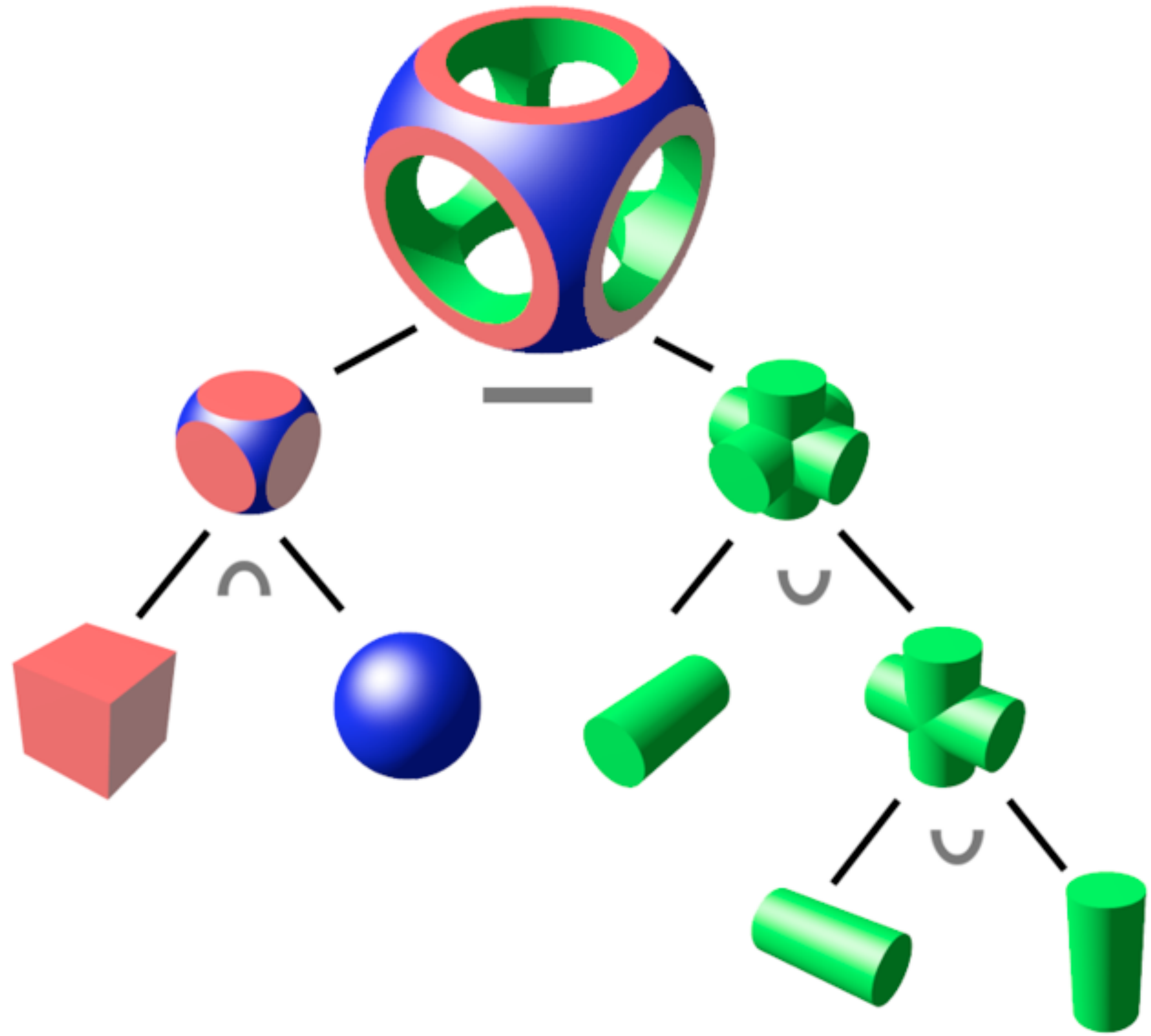
Boolean
union



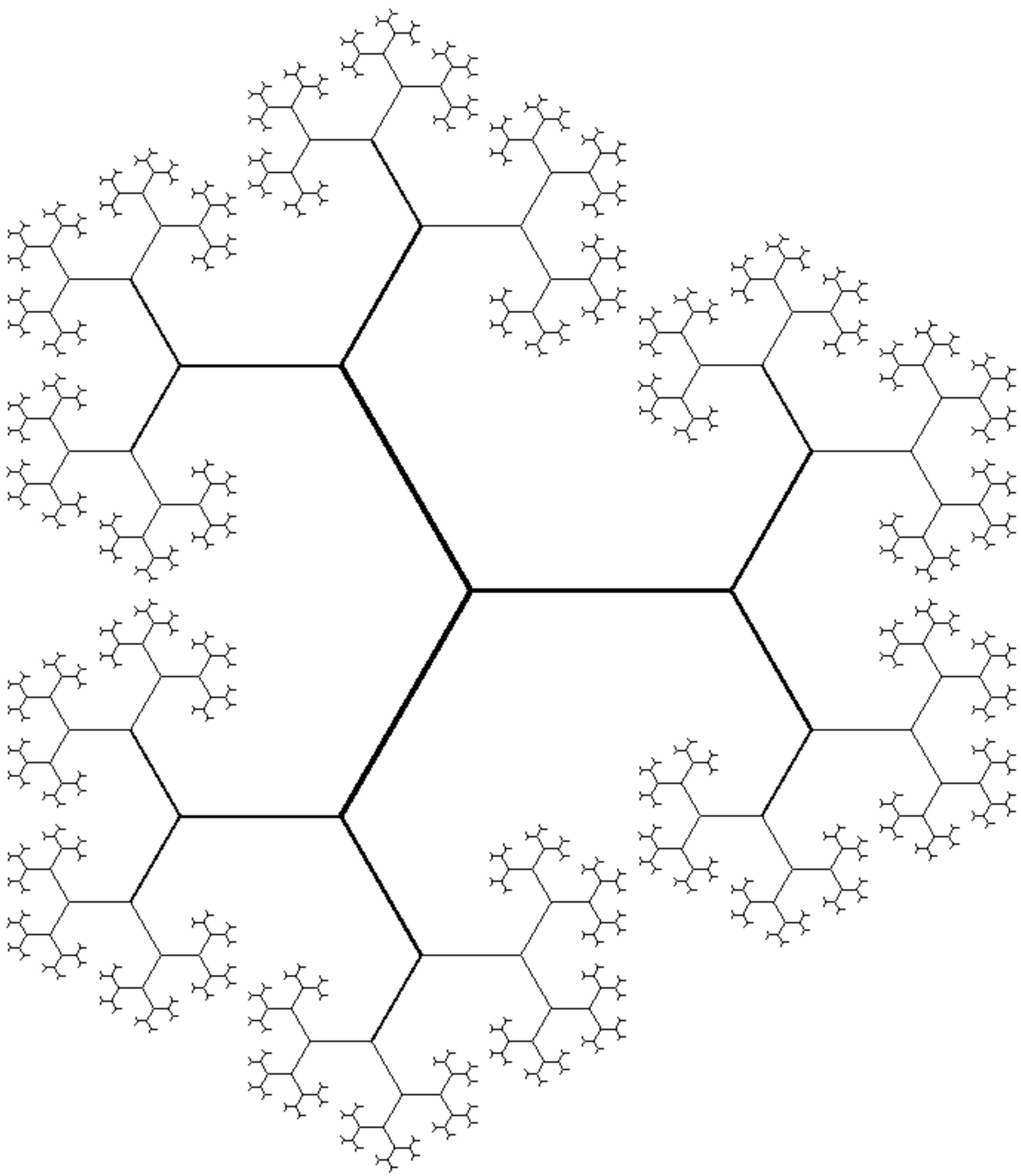
Boolean
difference

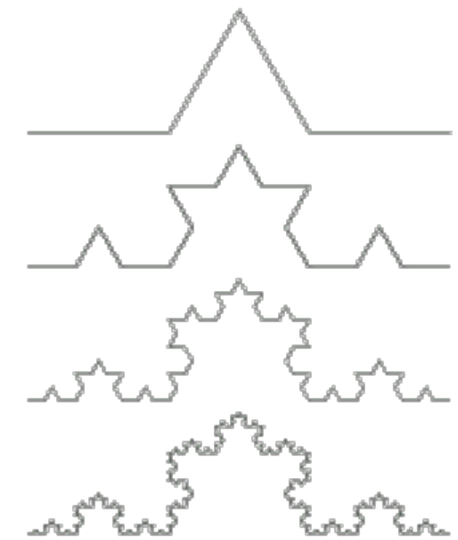
Boolean
intersection

Constructive **S**olid **G**eometry



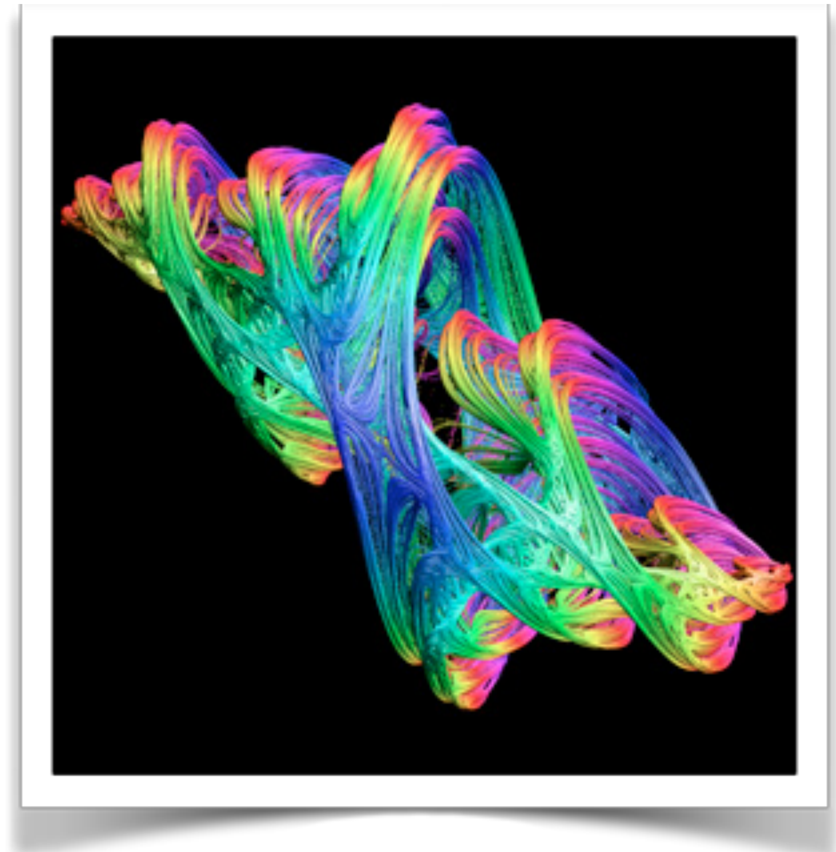
Constructive **S**olid **G**eometry



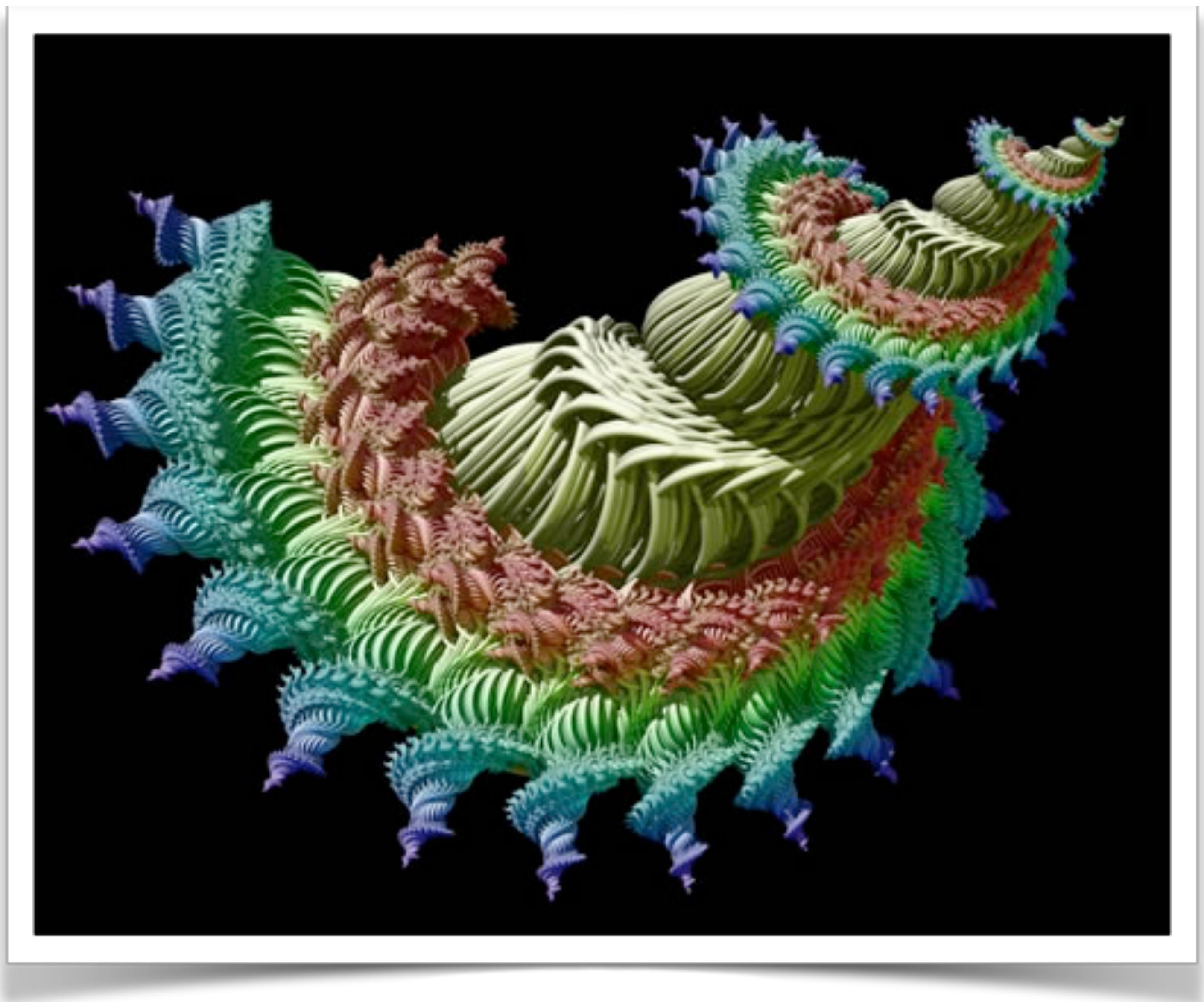


Self-similar
Self-affine
Invariant fractal set

Fractal



Julia Set



Fractal

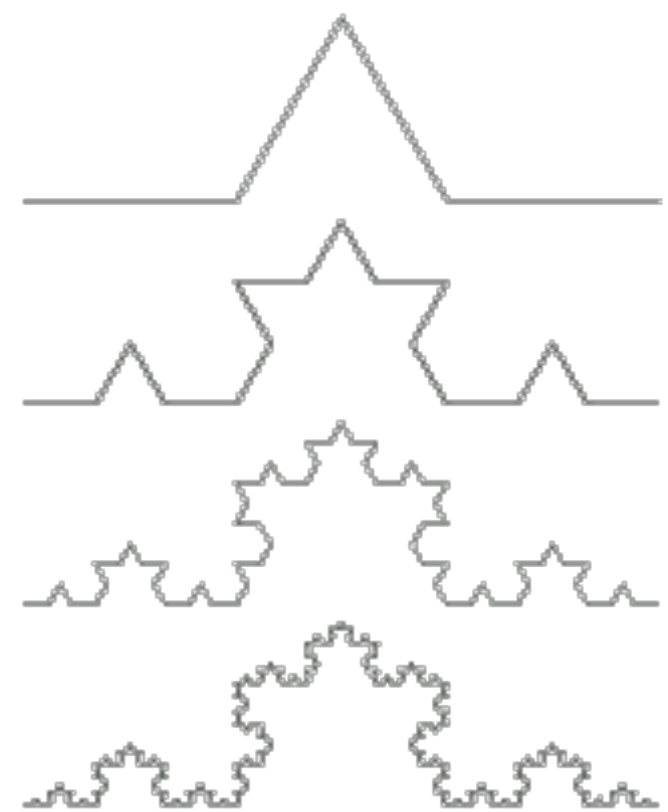
字符 : F

常数 : +, −

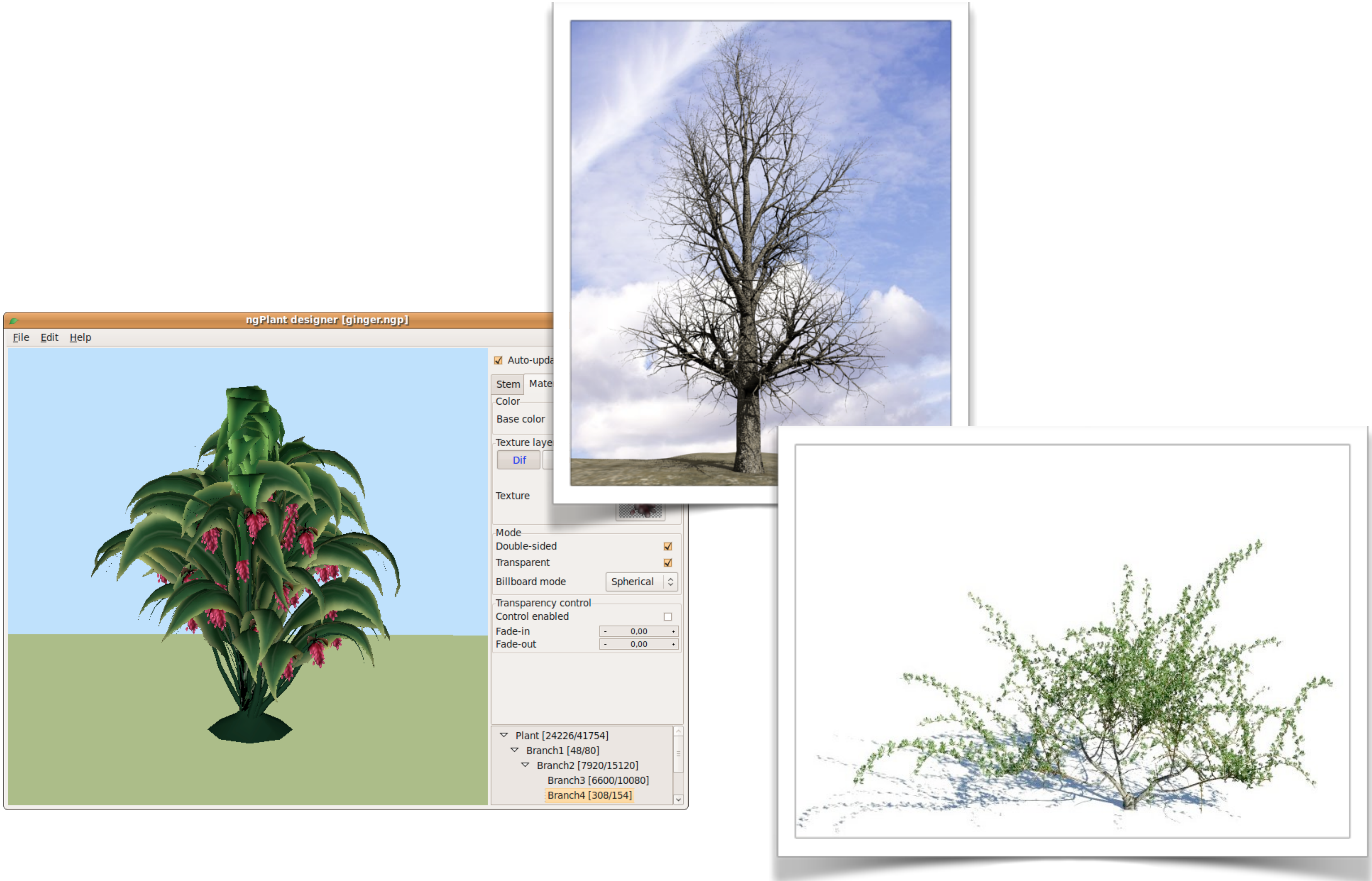
公理 : F++F++F

规则: F → F−F++F−F

- F : 向前
- - : 左转60°
- + : 右转60°



L-system



L-system for tree

V-Plant: A Variational Approach for
Interactive Tree Modeling

Online Submisson ID: 355

Sketch Tree Demo

procedure (script) modeling

- A lot of research
  - Natural scene modeling
  - City modeling (shape grammar)

- Amazing software
  - nodebox v.s. processing

Particle systems