

# 隐函数曲面造型和动画技术

金小刚

Email: [jin@cad.zju.edu.cn](mailto:jin@cad.zju.edu.cn)

浙江大学CAD&CG国家重点实验室

紫金港校区蒙民伟楼512

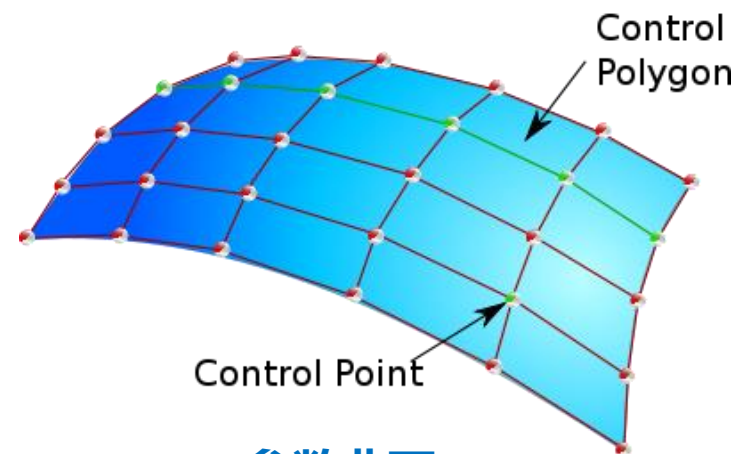
# 如何制作Terminator 中的液态特效?



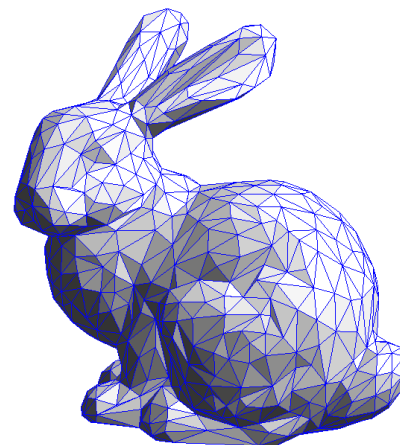
# 自由曲面的表示

- **参数曲面(Parametric Surface)**
  - 如Bezier曲面、B样条曲面、NURBS曲面等。参数曲面在造型和动画设计中取得了非常大的成功，很多造型和动画软件系统包含基于NURBS曲面的造型功能。
- **多边形网格(Polygon Mesh)**
- **细分曲面(Subdivision Surface)**
- **隐式曲面(Implicit surface, level set)**

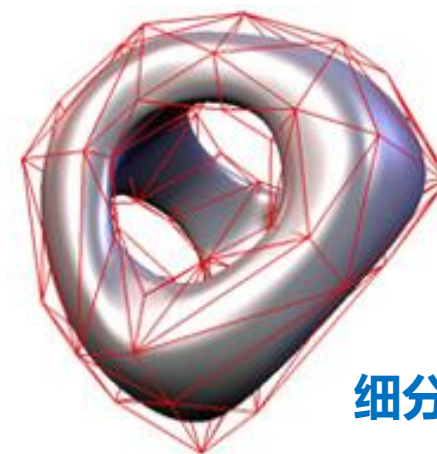
B-Rep(边界表示)



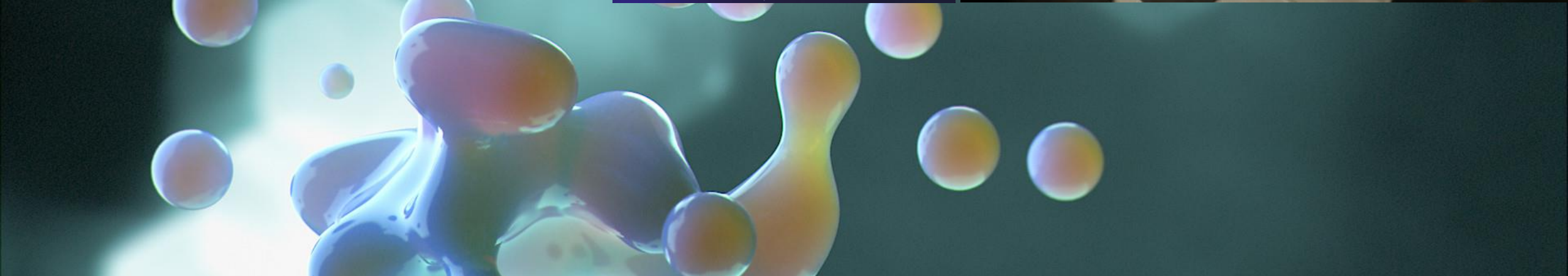
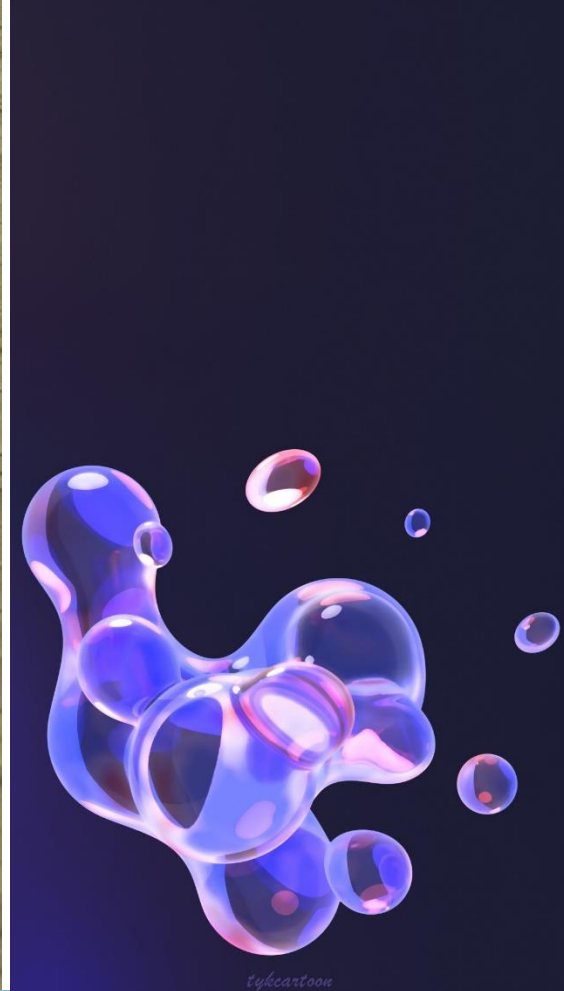
参数曲面

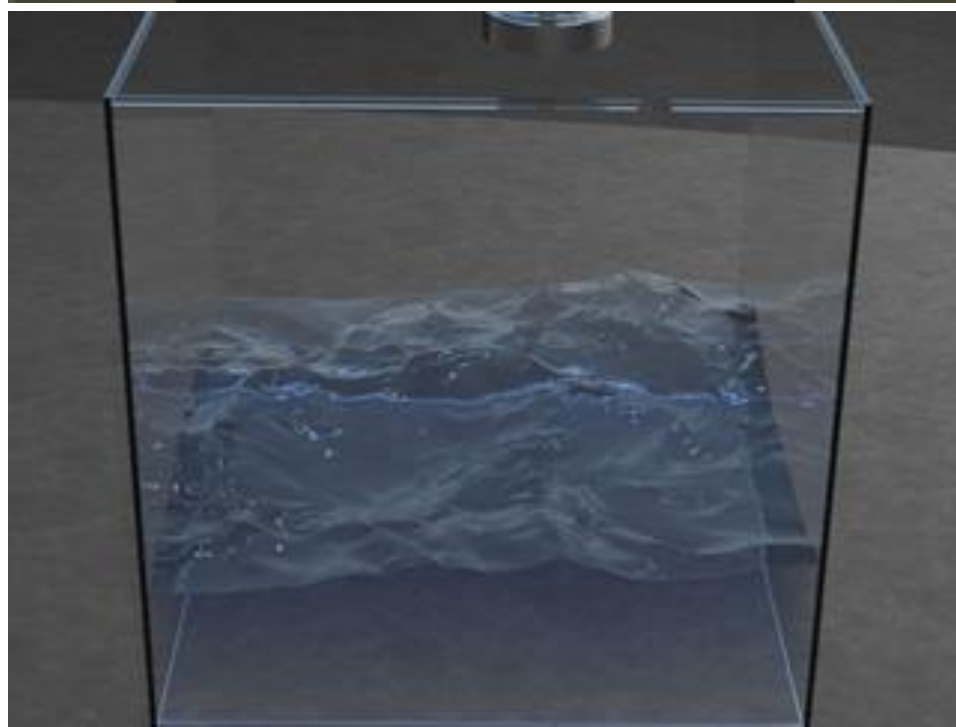
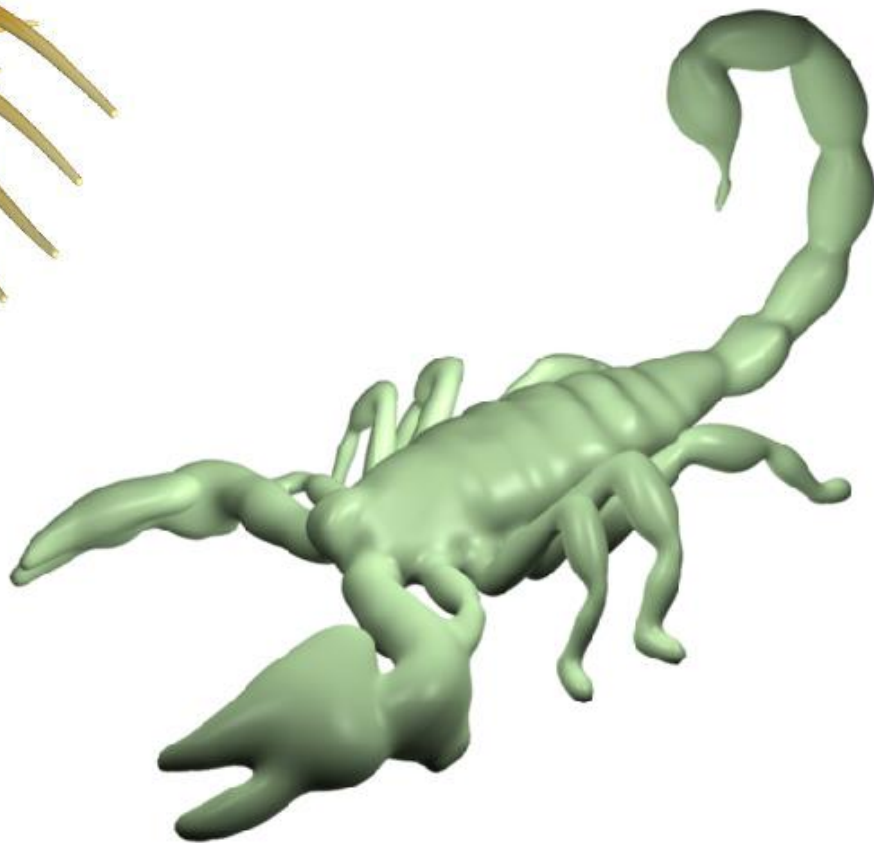


多边形网格



细分曲面



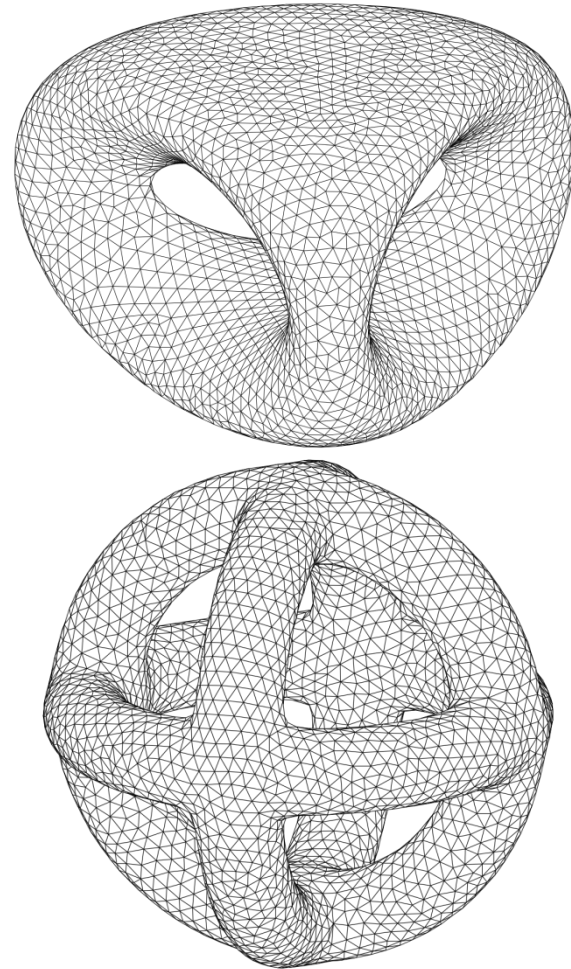
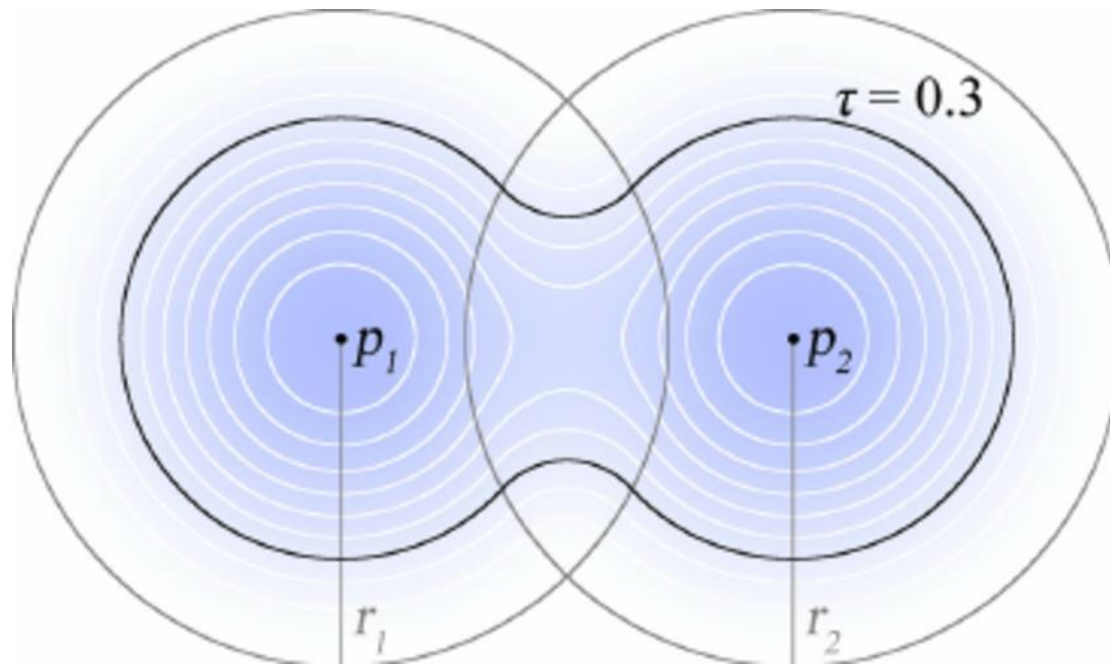






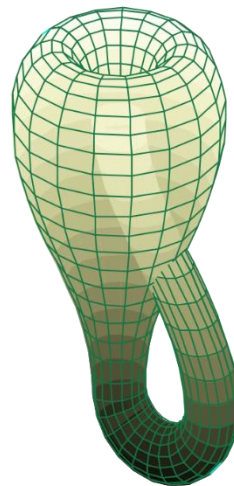
# 隐式曲面

$$f(x, y, z) = T$$



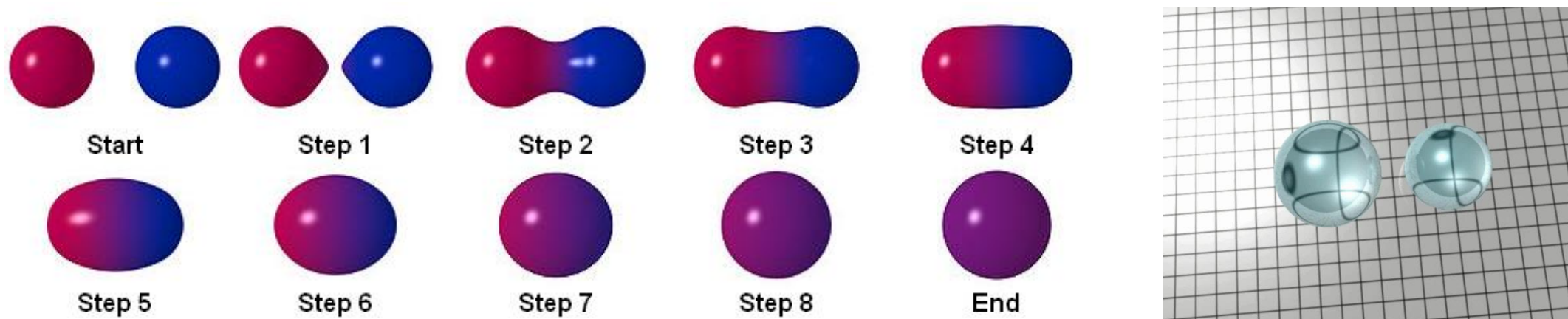
# 隐式曲面

- 边界表示(B-rep)曲面的缺点：
  - 改变拓扑很困难；
  - 边界曲面不一定表示一个体模型，如Klein瓶；



- 从八十年代初期开始，出现了一种全新的造型技术，称为元球造型技术 (Metaball)。
- 元球造型是一种隐式曲面造型技术，该技术采用具有等势场值的点集来定义曲面。因此，元球生成的面实际上是一张等势面。

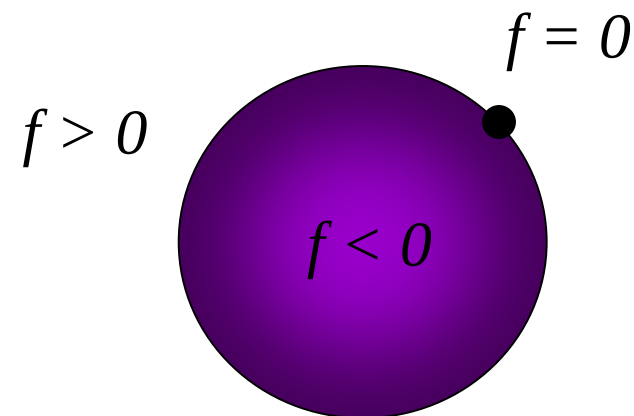
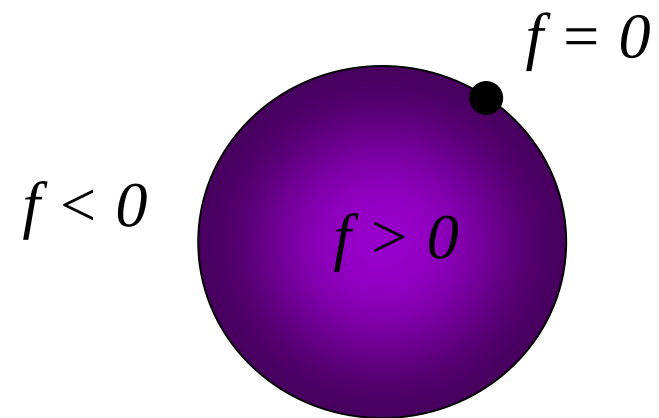
# 隐式曲面



- 元球相互靠近到一定距离产生变形，再进一步靠近时则融合成光滑表面。
- 以两个元球为例，元球靠近时的变形过程如上图所示。最初是两个独立的球，彼此接近时，相对的面开始隆起变形，接近到一定程度就会象水滴（水银）一样融合成一个面。然后变成花生形状、胶囊形状、最后变成一个球。
- 上述过程实际上提供了一种模拟两滴水（水银）融合的动画过程，而用参数曲面和多边形网格造型方法来模拟此类动画过程是很困难的。
- 隐式曲面很适合表示可变形和可变拓扑的物体，因而对动画非常有用(如morphing)。

# 隐式曲面的表示

- 把曲面表示为定义于空间的实函数  $f(x, y, z)$
- 在空间对点进行分类
- 曲面通过隐式定义
  - 在内部  $f > 0$
  - 在外部  $f < 0$
  - 在曲面上  $f = 0$
- 等价地
  - 在内部  $f < 0$
  - 在外部  $f > 0$

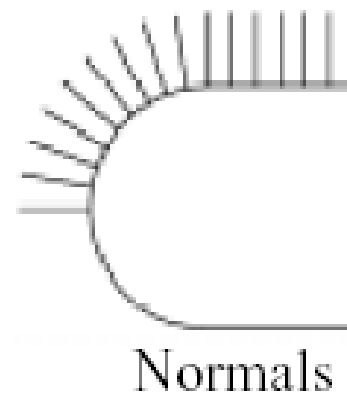


# 隐式曲面的法向

- 用函数的梯度来计算曲面的法向

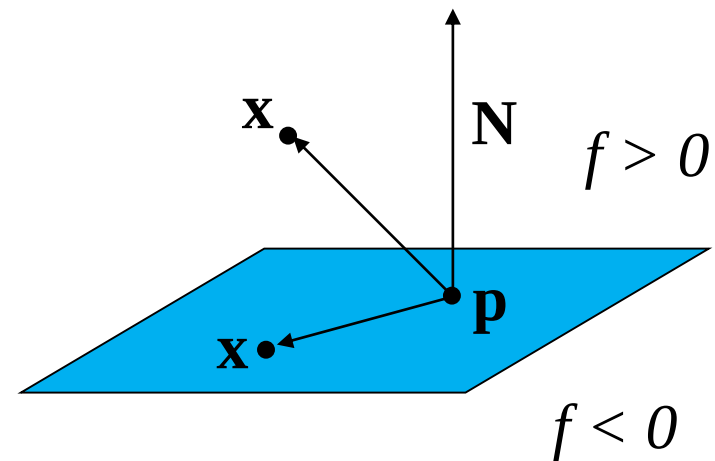
$$\nabla f(x,y,z) = (df/dx, df/dy, df/dz)$$

- 梯度没有必要一定是单位长度
- 梯度指向函数 $f$ 递增最快的方向
  - 当  $f < 0$  表示内部时, 梯度指向**外**
  - 当  $f > 0$  表示内部时, 梯度指向**内**



# 平面的隐式表示

- 平面界定半空间
- 平面用点 $\mathbf{p}$  和法向  $\mathbf{N}$ 来指定
- 平面内的任意点  $\mathbf{x}$  与 $\mathbf{p}$ 相连的直线与法向 $\mathbf{N}$ 垂直
- 如果 $\|\mathbf{N}\| = 1$ ，则 $f$ 为点 $\mathbf{x}$ 到平面的距离

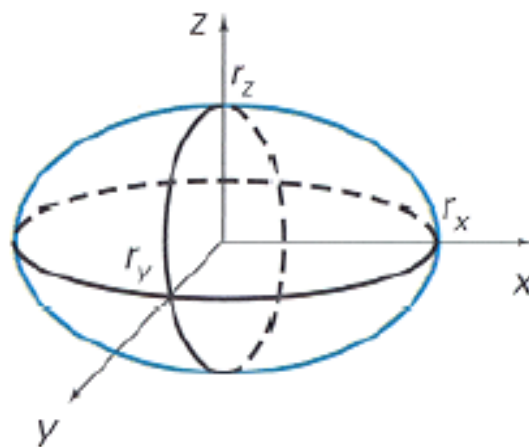


$$f(\mathbf{x}) = (\mathbf{x} - \mathbf{p}) \cdot \mathbf{N}$$

# 隐式曲面的性质(1)

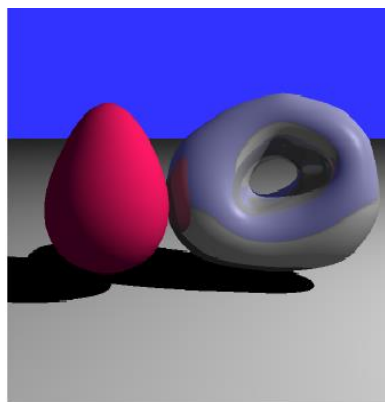
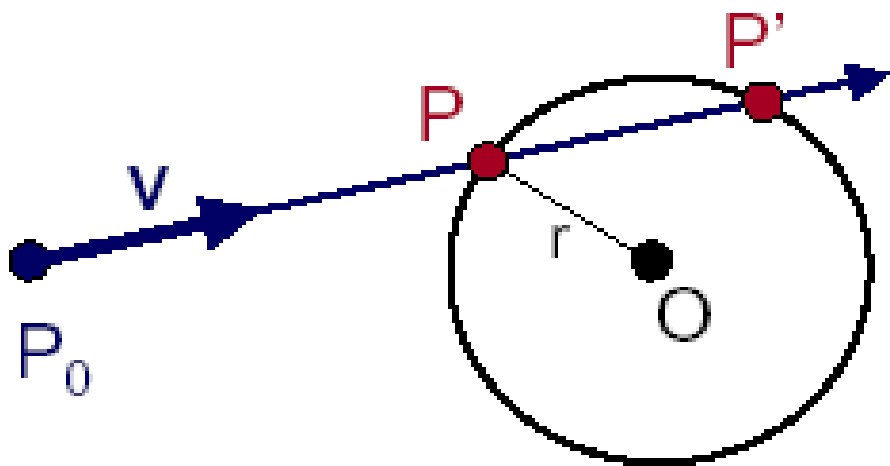
- 可有效判断一个点是否位于曲面的内部
  - 计算  $f(x,y,z)$  , 判断点 $(x,y,z)$ 是 **inside/outside/on**

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1 = 0$$



# 隐式曲面的性质(2)

- 有效的曲面求交计算
  - 把光线方程  $\mathbf{P} = \mathbf{P}_0 + t*\mathbf{V}$  代入隐式曲面方程



$$\left(x^2 + \frac{9}{4}y^2 + z^2 - 1\right)^3 - x^2 z^3 - \frac{9}{80}y^2 z^3 = 0$$

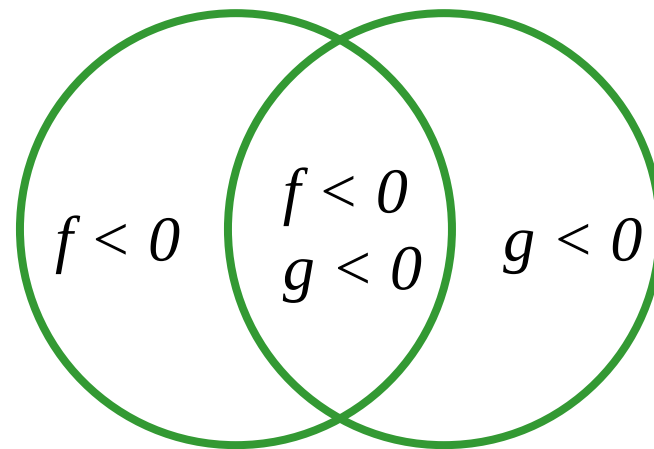
# 隐式曲面的性质(3)

- 有效的布尔操作(CSG)

- 假设  $f < 0$  表示内部

- 布尔操作可简单地用min/max操作来实现

- 并操作(Union):  $\min(f, g)$
- 交操作(Intersection):  $\max(f, g)$
- 补操作(Complement):  $-f$
- 减操作(Subtraction):  $\max(f, -g)$



# 隐式曲面的表示

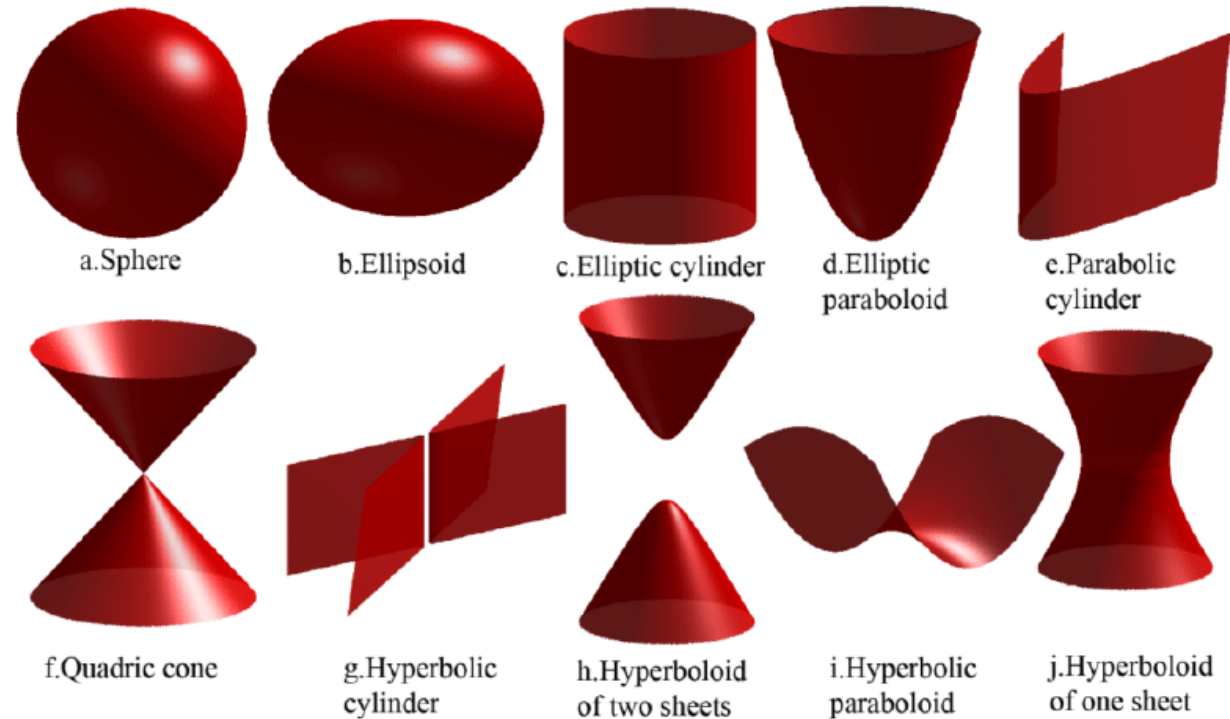
- 如何定义隐式曲面?
  - 代数曲面(Algebraic)
  - 距离曲面(Distance surface)
  - 卷积曲面(Convolution surface)
  - 通过样本定义(Samples)
  - RBF曲面(Radial Basis Function)
  - 水滴模型(Bloppy models, Metaball)
  - ...

# 二次代数曲面(Quadrics)

$$f(x,y,z) = Ax^2 + 2Bxy + 2Cxz + 2Dx + Ey^2 + 2Fyz + 2Gy + Hz^2 + 2Iz + J$$

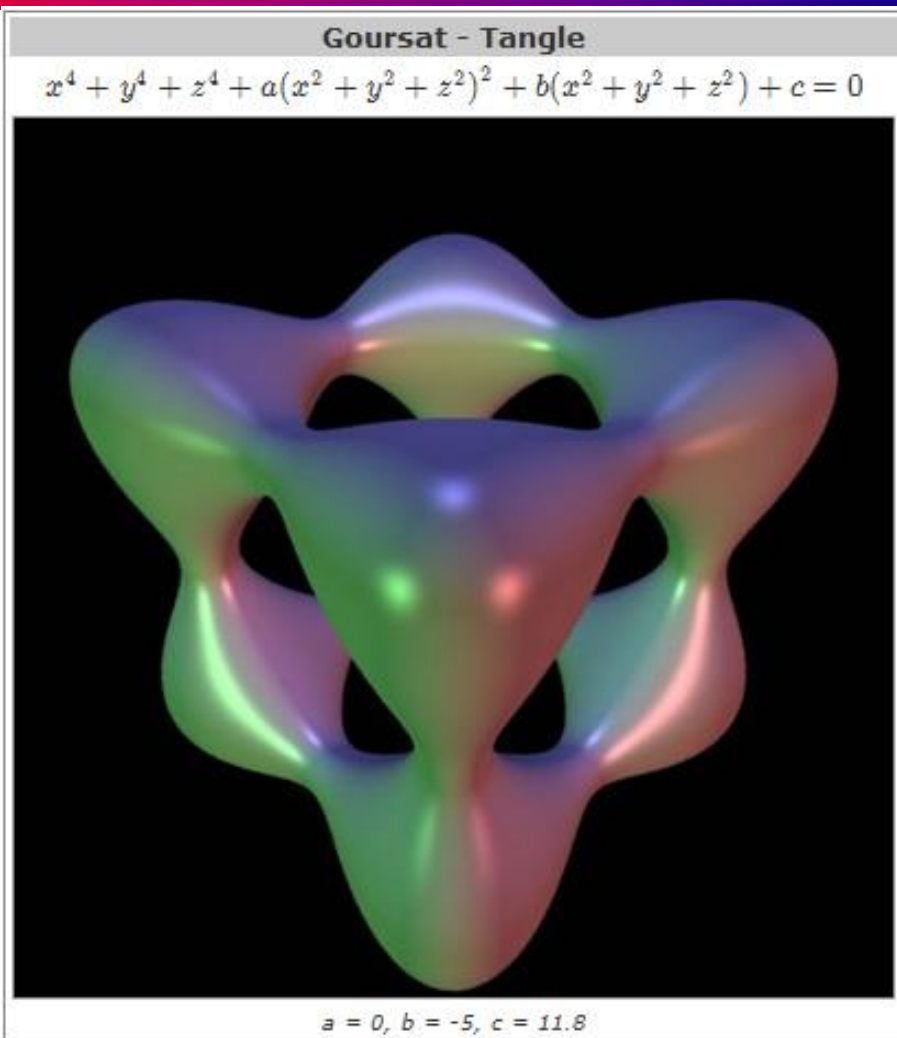
- 可以表示:

- 球(Sphere)
- 椭球(Ellipsoid)
- 圆柱(Cylinder)
- 双曲面(Hyperboloid)
- 抛物面(Paraboloid)
- .....

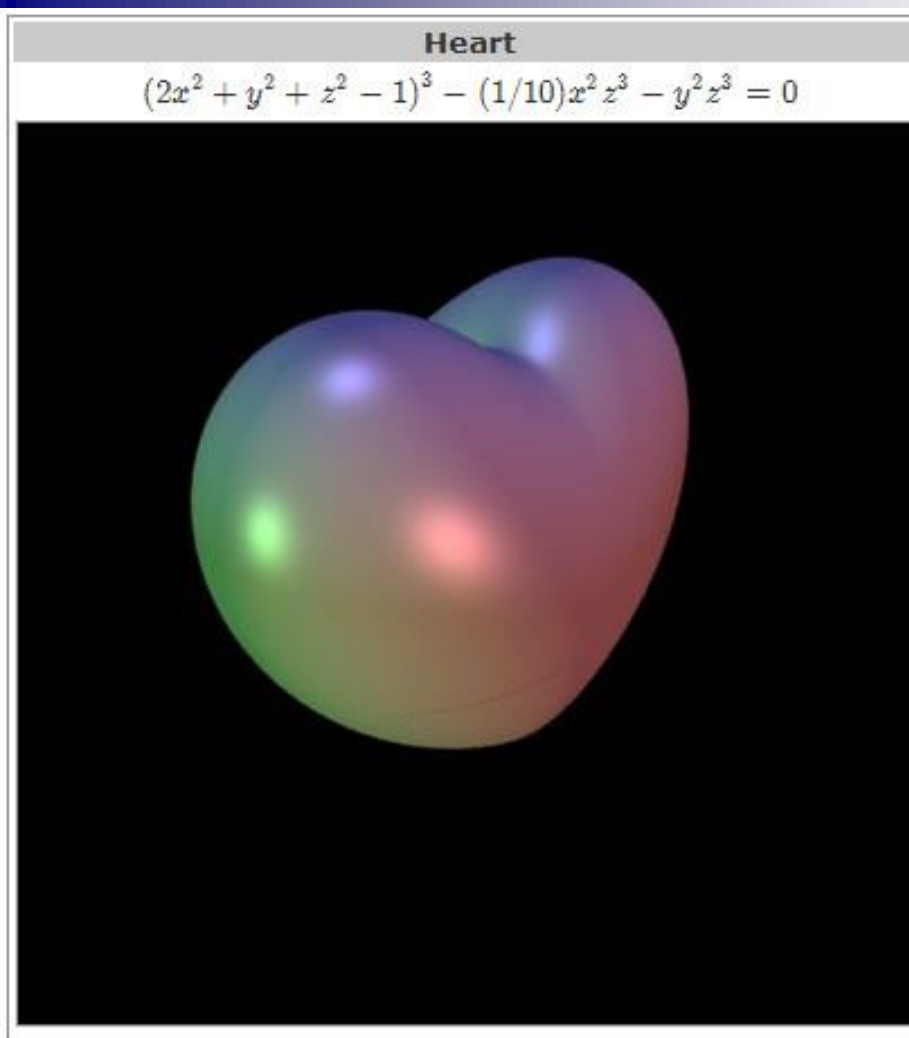


# 高次代数曲面

(Higher degree algebraic surfaces)



四次代数曲面



六次代数曲面

# 高次代数曲面

(Higher degree algebraic surfaces)

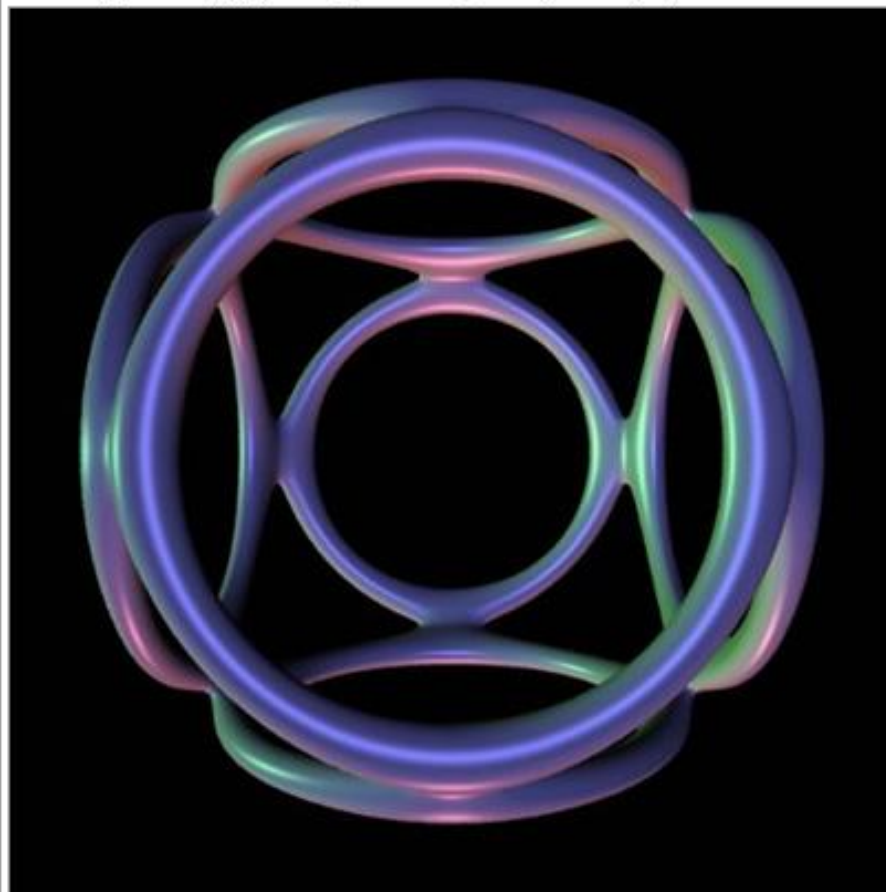
Bretzel (4 holes)

$$0.1(x^2(1.21 - x^2)^2(3.8 - x^2)^3 - 10y^2)^2 + 60z^2 - 2 = 0$$



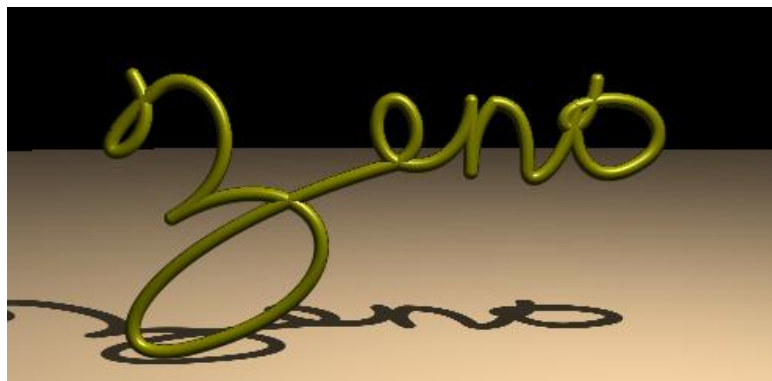
Decocube

$$((x^2 + y^2 - 0.8^2)^2 + (z^2 - 1)^2)((x^2 + z^2 - 0.8^2)^2 + (y^2 - 1)^2)((x^2 + y^2 - 0.8^2)^2 + (z^2 - 1)^2) - 0.01 = 0$$



# 距离曲面(Distance Surfaces)

- 球:  $d(\mathbf{x}, \text{point}) - r$
- 圆柱:  $d(\mathbf{x}, \text{line}) - r$
- 圆环:  $d(\mathbf{x}, \text{circle}) - r$
- 广义圆柱:  $d(\mathbf{x}, \text{curve}) - r$
- 距离场:  $d(\mathbf{x}, \text{object}) - r$

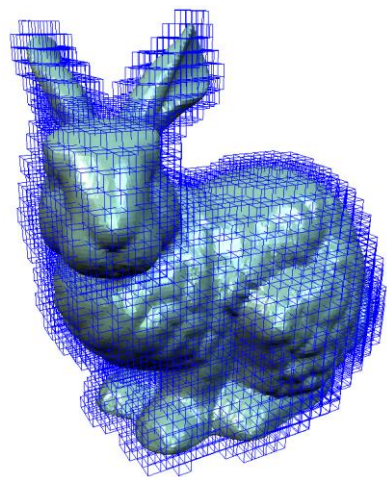


|     |     |      |      |      |      |      |      |      |      |      |      |      |     |     |     |
|-----|-----|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|
| 4.9 | 4.4 | 4.0  | 3.7  | 3.5  | 3.5  | 3.7  | 4.0  | 4.1  | 4.0  | 4.0  | 4.1  | 4.4  | 4.8 | 5.2 | 5.8 |
| 4.1 | 3.5 | 3.0  | 2.7  | 2.5  | 2.5  | 2.7  | 3.0  | 3.2  | 3.0  | 3.0  | 3.2  | 3.5  | 3.9 | 4.4 | 5.0 |
| 3.4 | 2.7 | 2.1  | 1.7  | 1.5  | 1.5  | 1.7  | 2.1  | 2.2  | 2.0  | 2.0  | 2.2  | 2.5  | 3.0 | 3.6 | 4.3 |
| 2.7 | 1.9 | 1.3  | 0.8  | 0.5  | 0.5  | 0.8  | 1.3  | 1.2  | 1.0  | 1.0  | 1.2  | 1.6  | 2.2 | 2.9 | 3.6 |
| 2.1 | 1.3 | 0.5  | -0.1 | -0.5 | -0.5 | -0.1 | 0.5  | 0.3  | 0.0  | 0.0  | 0.3  | 0.8  | 1.4 | 2.2 | 3.0 |
| 1.7 | 0.8 | -0.1 | -0.9 | -1.4 | -1.4 | -0.9 | -0.1 | -0.6 | -1.0 | -1.0 | -0.6 | 0.0  | 0.8 | 1.6 | 2.5 |
| 1.5 | 0.5 | -0.5 | -1.4 | -2.3 | -2.3 | -1.4 | -0.6 | -1.4 | -1.9 | -1.9 | -1.4 | -0.6 | 0.3 | 1.2 | 2.2 |
| 1.5 | 0.5 | -0.5 | -1.4 | -2.3 | -2.3 | -1.4 | -1.0 | -1.9 | -2.8 | -2.8 | -1.9 | -1.0 | 0.0 | 1.0 | 2.0 |
| 1.7 | 0.8 | -0.1 | -0.9 | -1.4 | -1.4 | -0.9 | -1.0 | -1.9 | -2.8 | -2.8 | -1.9 | -1.0 | 0.0 | 1.0 | 2.0 |
| 2.1 | 1.3 | 0.5  | -0.1 | -0.5 | -0.5 | -0.1 | -0.6 | -1.4 | -1.9 | -1.9 | -1.4 | -0.6 | 0.3 | 1.2 | 2.2 |
| 2.7 | 1.9 | 1.3  | 0.8  | 0.5  | 0.5  | 0.8  | 0.0  | -0.6 | -1.0 | -1.0 | -0.6 | 0.0  | 0.8 | 1.6 | 2.5 |
| 3.4 | 2.7 | 2.1  | 1.7  | 1.5  | 1.5  | 1.4  | 0.8  | 0.3  | 0.0  | 0.0  | 0.3  | 0.8  | 1.4 | 2.2 | 3.0 |
| 4.1 | 3.5 | 3.0  | 2.7  | 2.5  | 2.5  | 2.2  | 1.6  | 1.2  | 1.0  | 1.0  | 1.2  | 1.6  | 2.2 | 2.9 | 3.6 |
| 4.9 | 4.4 | 4.0  | 3.7  | 3.5  | 3.5  | 3.0  | 2.5  | 2.2  | 2.0  | 2.0  | 2.2  | 2.5  | 3.0 | 3.6 | 4.3 |
| 5.7 | 5.3 | 4.9  | 4.6  | 4.5  | 4.4  | 3.9  | 3.5  | 3.2  | 3.0  | 3.0  | 3.2  | 3.5  | 3.9 | 4.4 | 5.0 |
| 6.6 | 6.2 | 5.9  | 5.6  | 5.5  | 5.2  | 4.8  | 4.4  | 4.1  | 4.0  | 4.0  | 4.1  | 4.4  | 4.8 | 5.2 | 5.8 |

其中:

$$d(\mathbf{x}, A) = \min \{ \|\mathbf{x} - \mathbf{y}\| : \mathbf{y} \in A \}$$

# 距离场(Distance Field)应用很广



距离为零的面

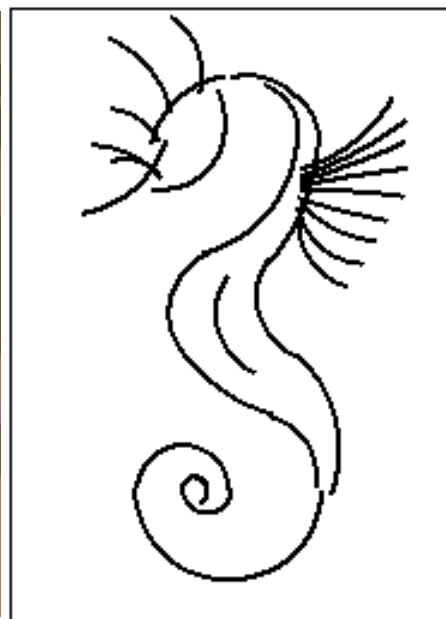
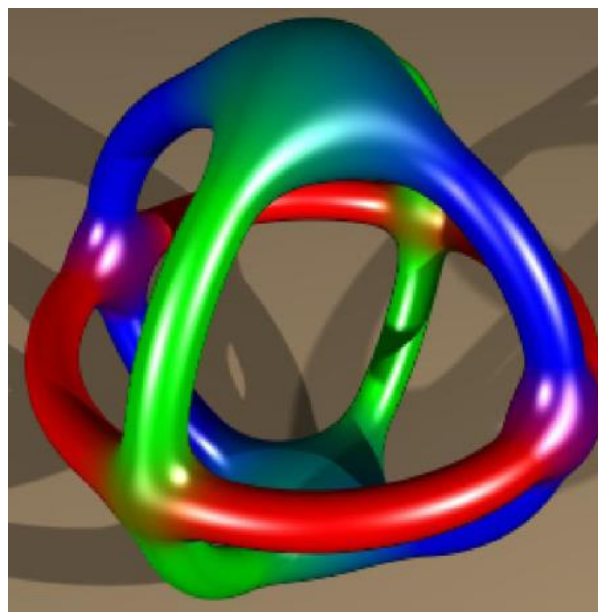
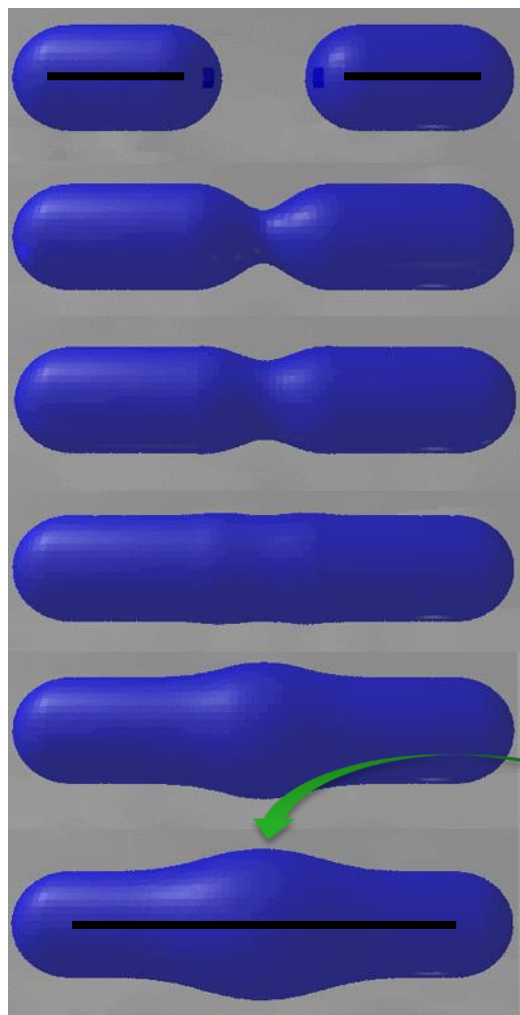


Kinect Fusion



Poisson Reconstruction

# 卷积曲面(Convolution surfaces)



为了解决距离曲面中出现的  
**鼓包(bulge)**问题而提出。

# 卷积曲面

- 通过骨架和核函数的卷积来生成标量场

- 标量场

$$f(\mathbf{P}) = \int_V g(\mathbf{Q})h(\mathbf{P} - \mathbf{Q})dV = (f \otimes h)(\mathbf{P})$$

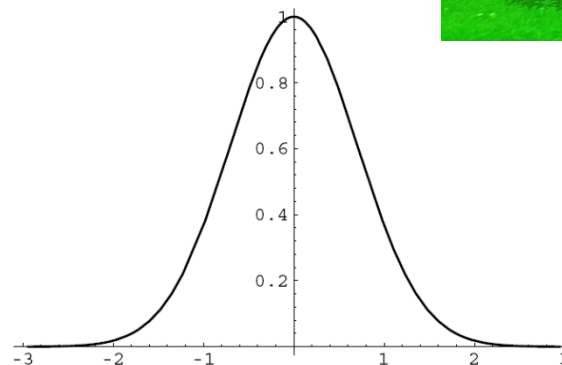
- 骨架

$$g(\mathbf{P}) = \begin{cases} 1, & \mathbf{P} \in \text{skeleton } V; \\ 0, & \text{otherwise.} \end{cases}$$

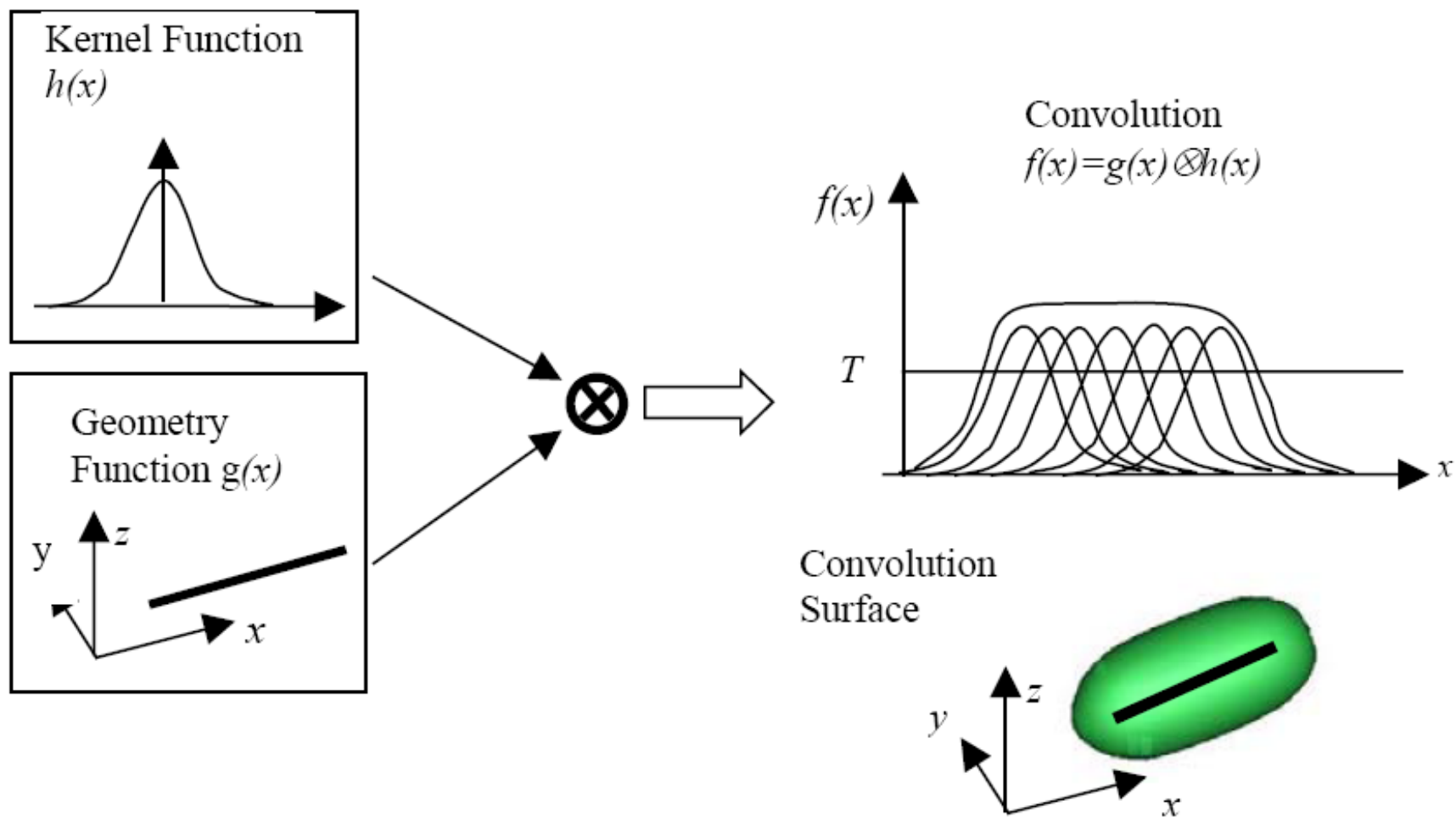
- 核函数(低通滤波函数)

- 例如: 高斯函数

$$h(r) = h(\mathbf{P} - \mathbf{Q}) = \exp(-a^2 r^2)$$



# 卷积曲面



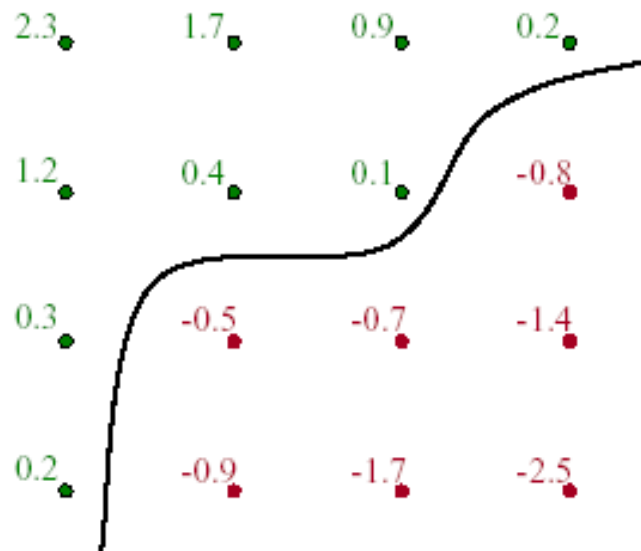
# 卷积曲面(Convolution surfaces)

**Analytical Solutions for  
Sketch-Based Convolution Surface  
Modeling on the GPU**

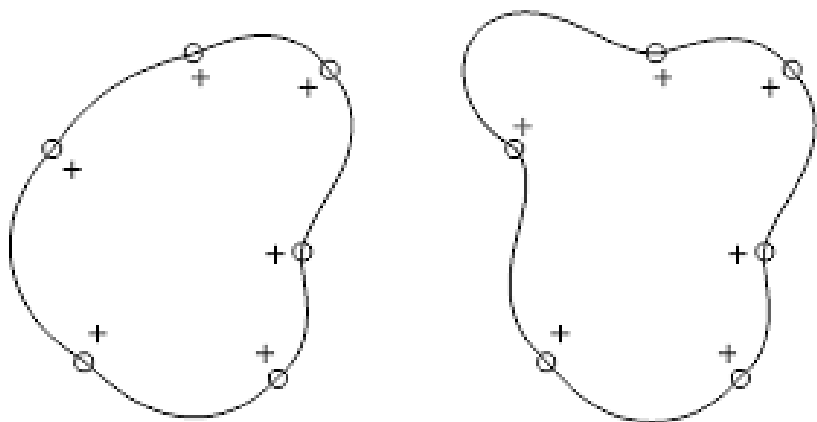
Xiaoqiang Zhu, Xiaogang Jin, Shengjun Liu, and Hanli Zhao, “Analytical Solutions for Sketch-Based Convolution Surface Modeling on the GPU ”, The Visual Computer, 2012, 28(11): 1115-1125.

# 采样函数(Sampled Functions)

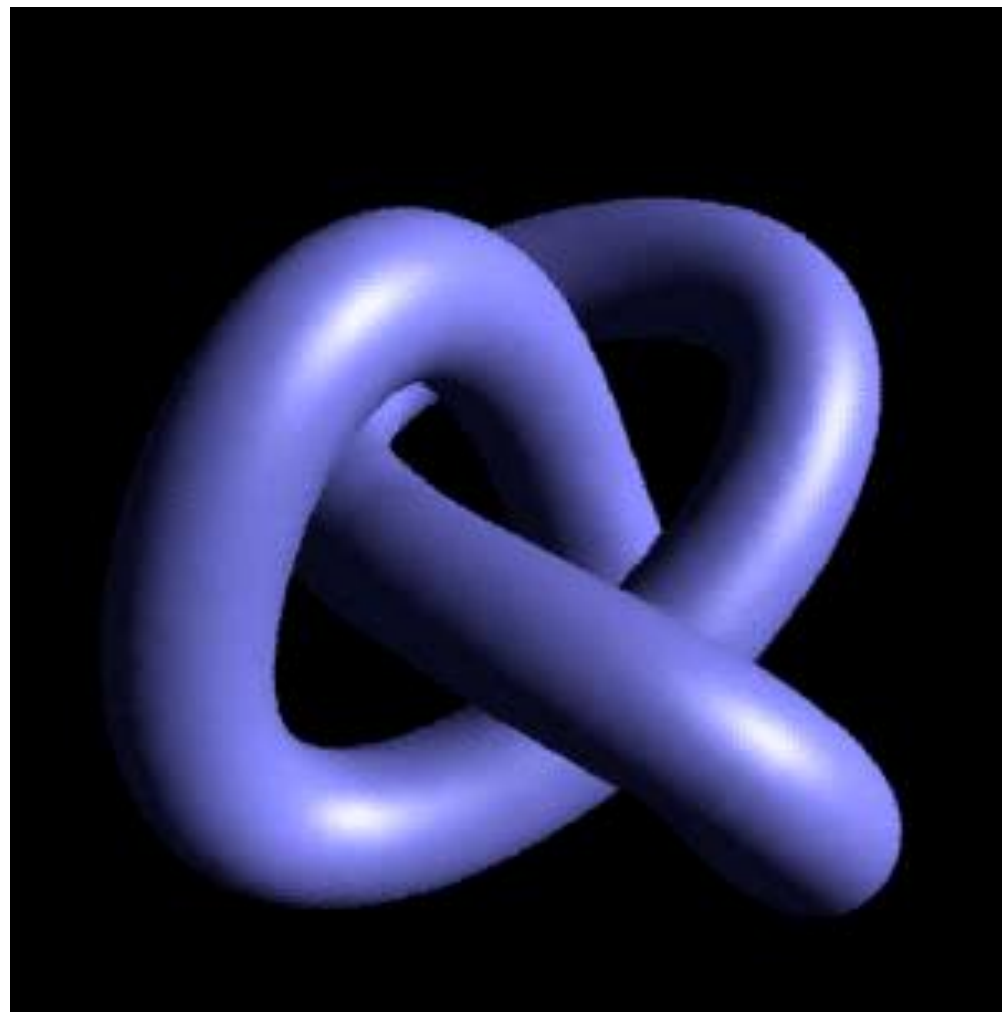
- 最常见的样本: 体素(voxels)
  - 对存储在规整网格(Regular Grid)上的采样点进行插值
  - 等值面(Isosurface)  $f(x, y, z) = 0$  定义了曲面



# RBF曲面(Radial Basis Function)

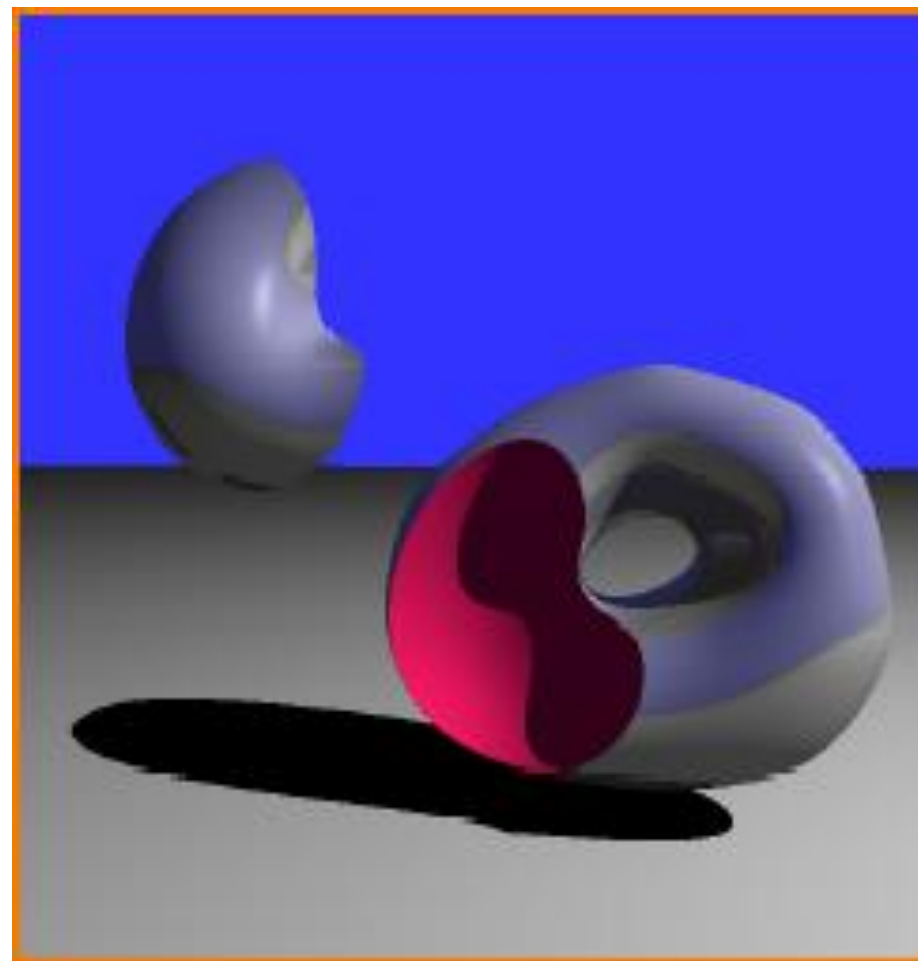


$$f(\mathbf{x}) = \sum_{j=1}^n d_j \phi(\mathbf{x} - \mathbf{c}_j) + P(\mathbf{x});$$

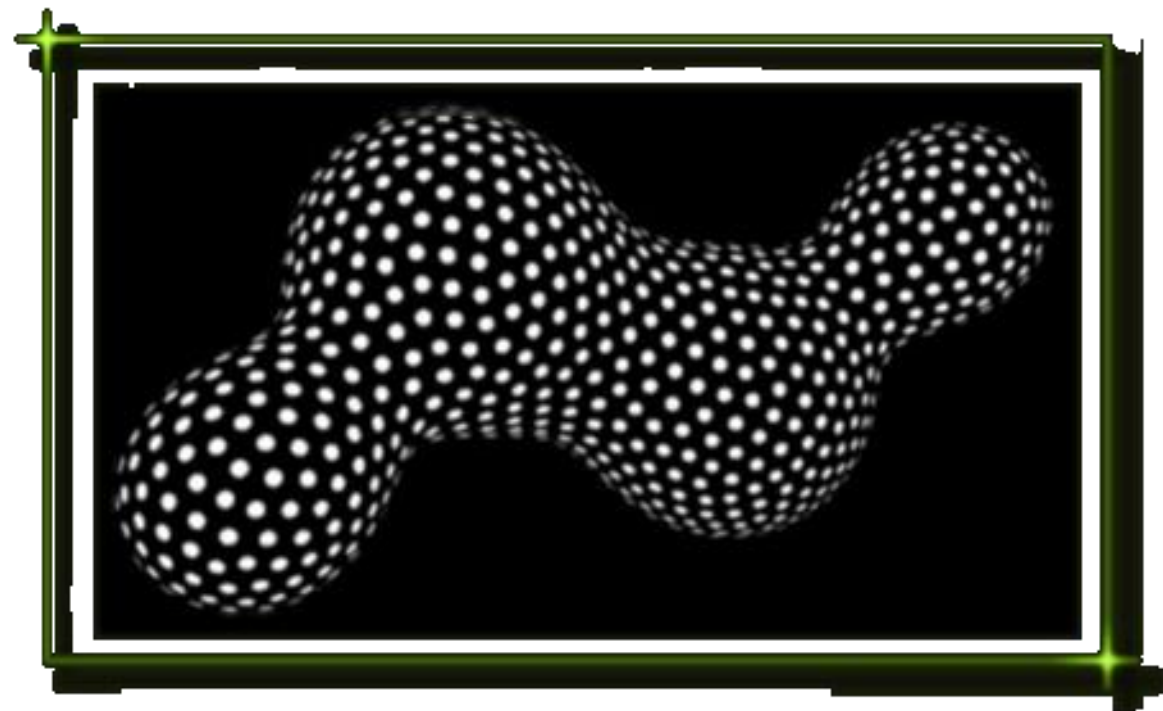
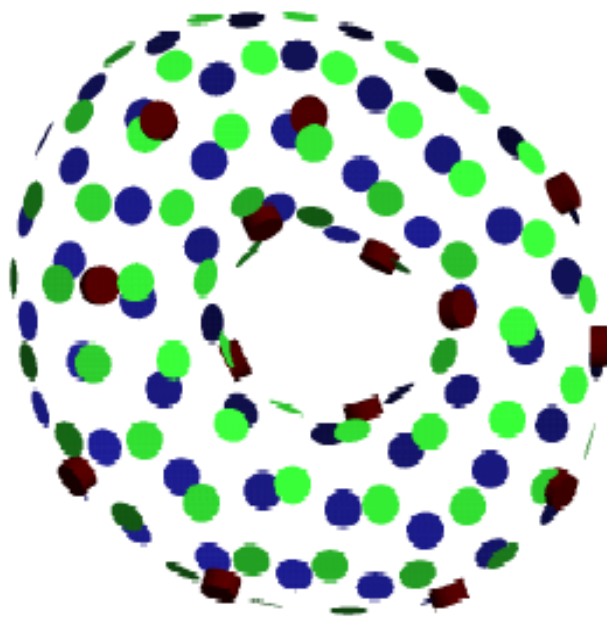
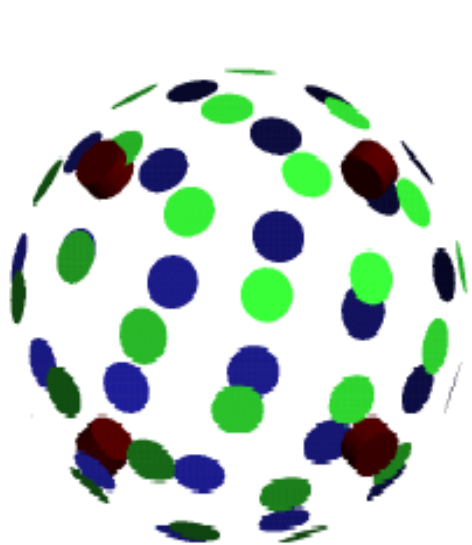


# 隐式曲面的绘制

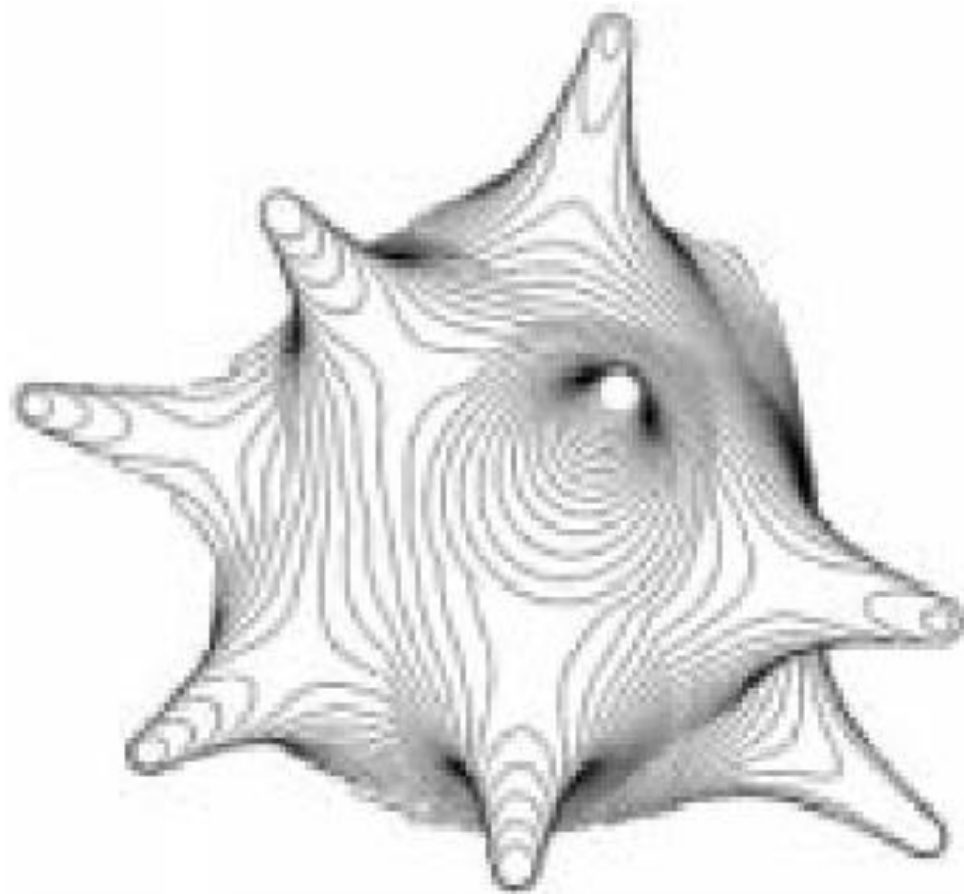
- 浮动粒子法  
(Floating particles)
- 轮廓法  
(Contour)
- 光线跟踪  
(Ray tracing)
- 多边形化(Polygonization)



# 浮动粒子法



# 轮廓法



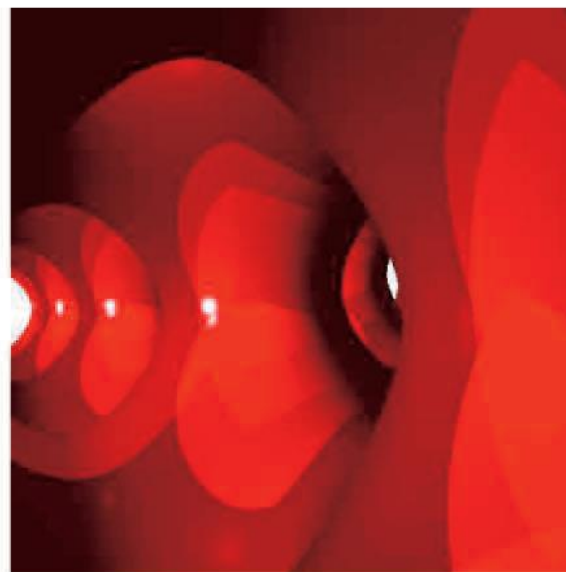
# 光线跟踪法



(a)



(b)



(c)

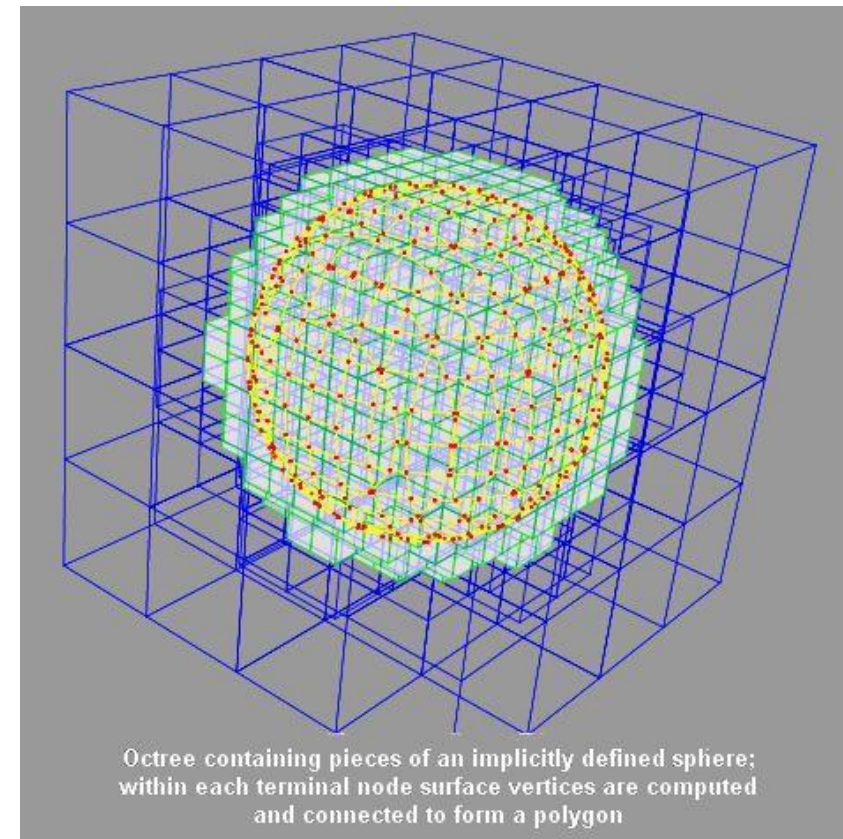


(d)

Singh, J.M. and Narayanan, P.J., 2009. Real-time ray tracing of implicit surfaces on the GPU. *IEEE transactions on visualization and computer graphics*, 16(2), pp.261-272.

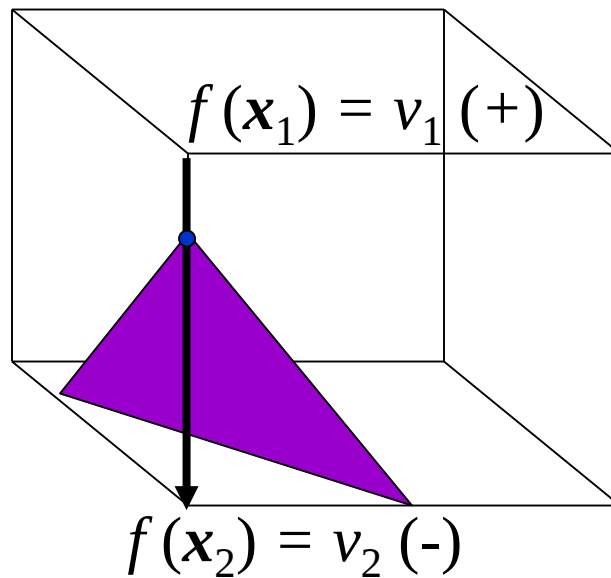
# 多边形化法(Polygonization)

- 把隐式曲面转化为多边形网格
- 两步
  - 把空间分割成单元
  - 在每个单元中用多边形去拟合曲面



Lorensen, William E., and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." ACM siggraph computer graphics. Vol. 21. No. 4. ACM, 1987. (到2021年1月12日, 引用次数**16536**)

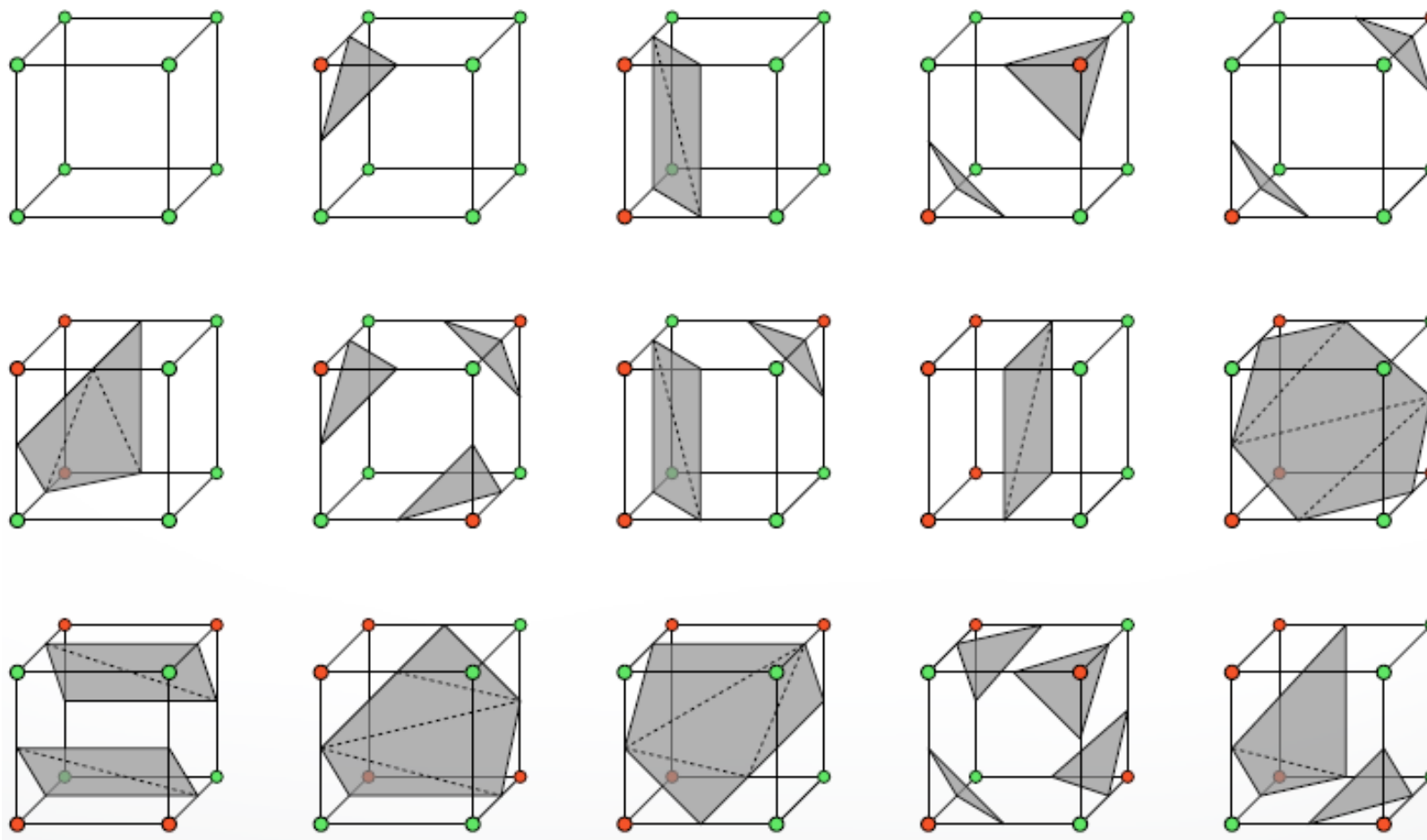
# 表面顶点的计算



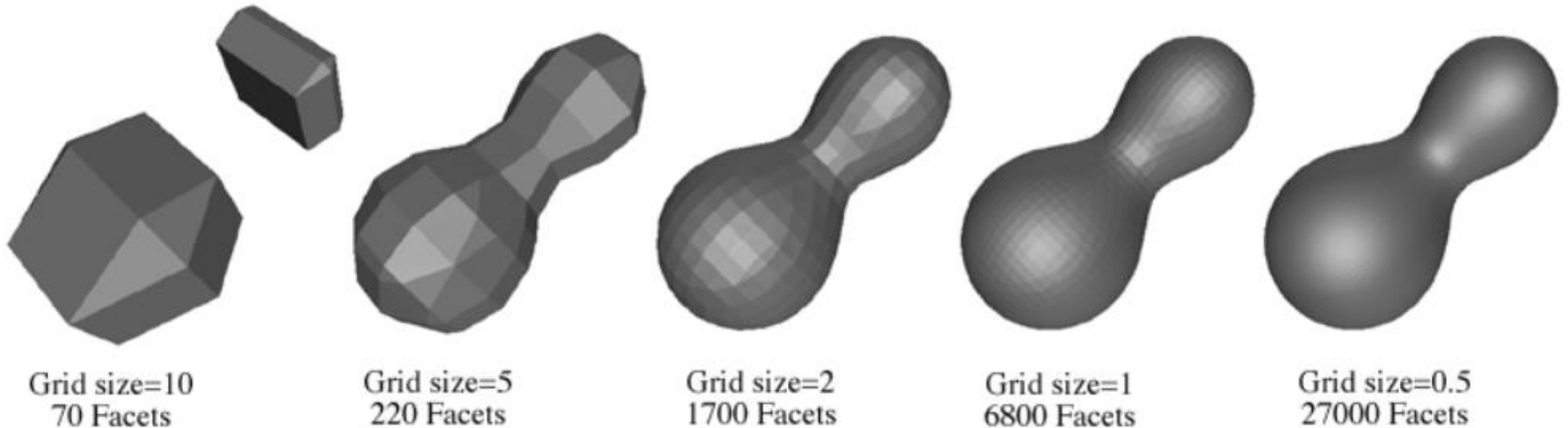
$$\mathbf{X} = \frac{v_1}{v_1 + v_2} \mathbf{X}_1 + \frac{v_2}{v_1 + v_2} \mathbf{X}_2$$

# Marching Cubes

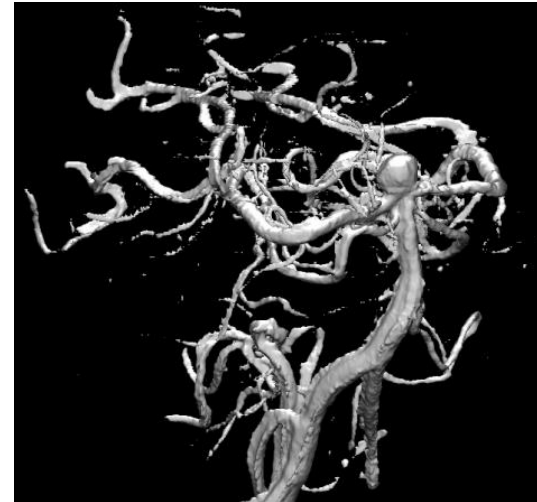
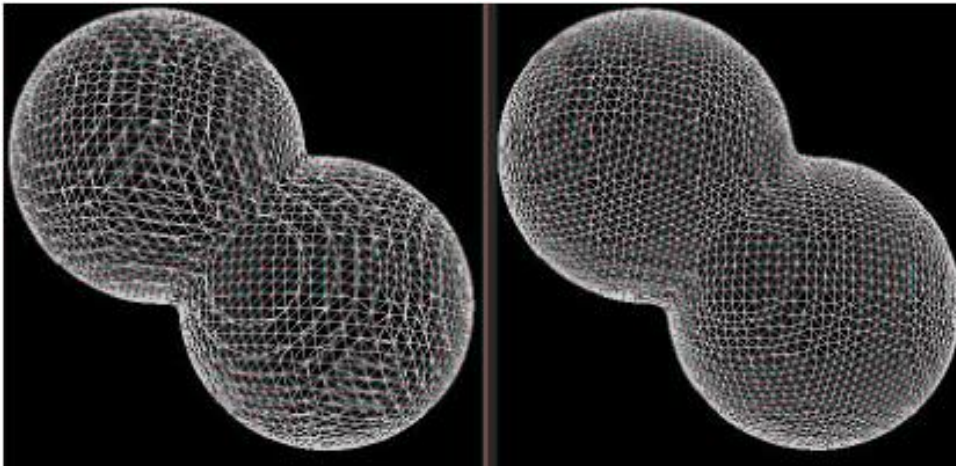
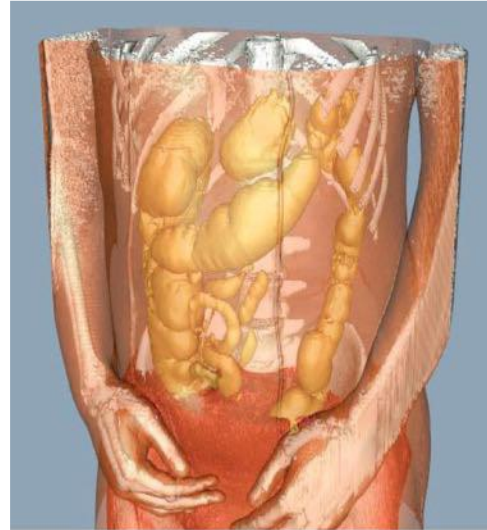
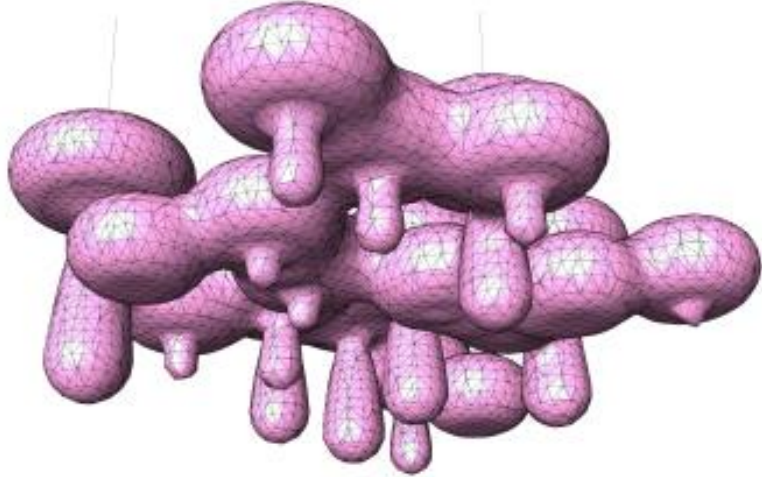
- 考虑到对称性，256 ( $2^8$ ) 种三维的单元情况可简化为15种不同的情形



# Marching Cubes——Effect of grid size

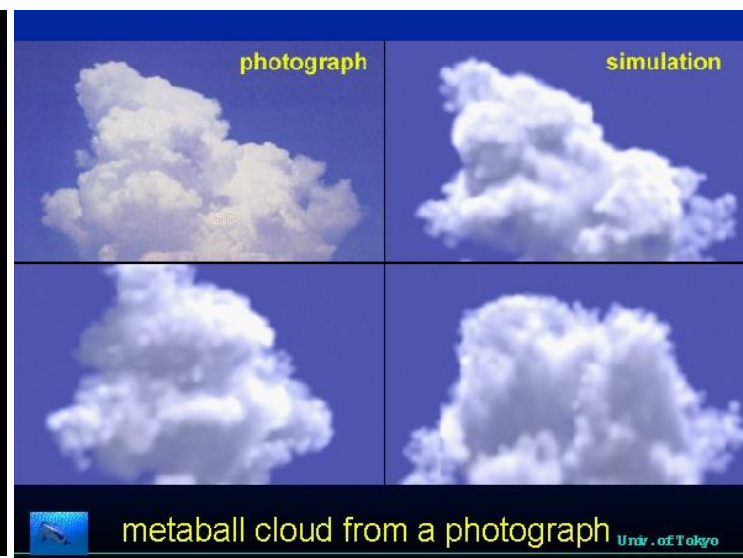


# Marching Cubes

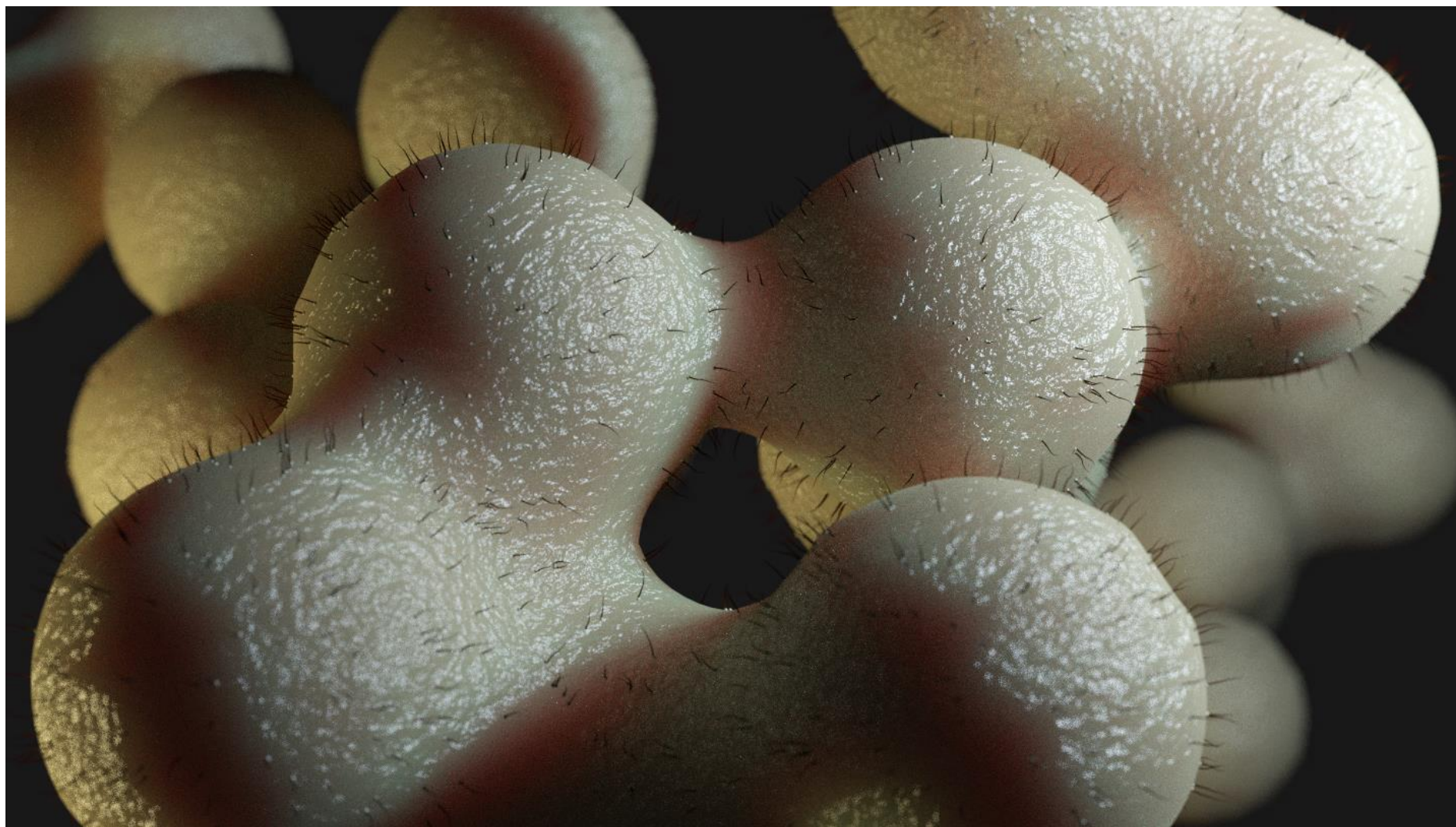


# 元球(Metaball)的造型

- 两个元球所能生成的形状相对较简单。但是，用更多元球却能够生成非常复杂的形状，如人的肌肉、胃、肠等。
- 下图显示了通过元球造型生成的一些场景，图中的手掌、人体、云彩等都是通过元球造型生成的。



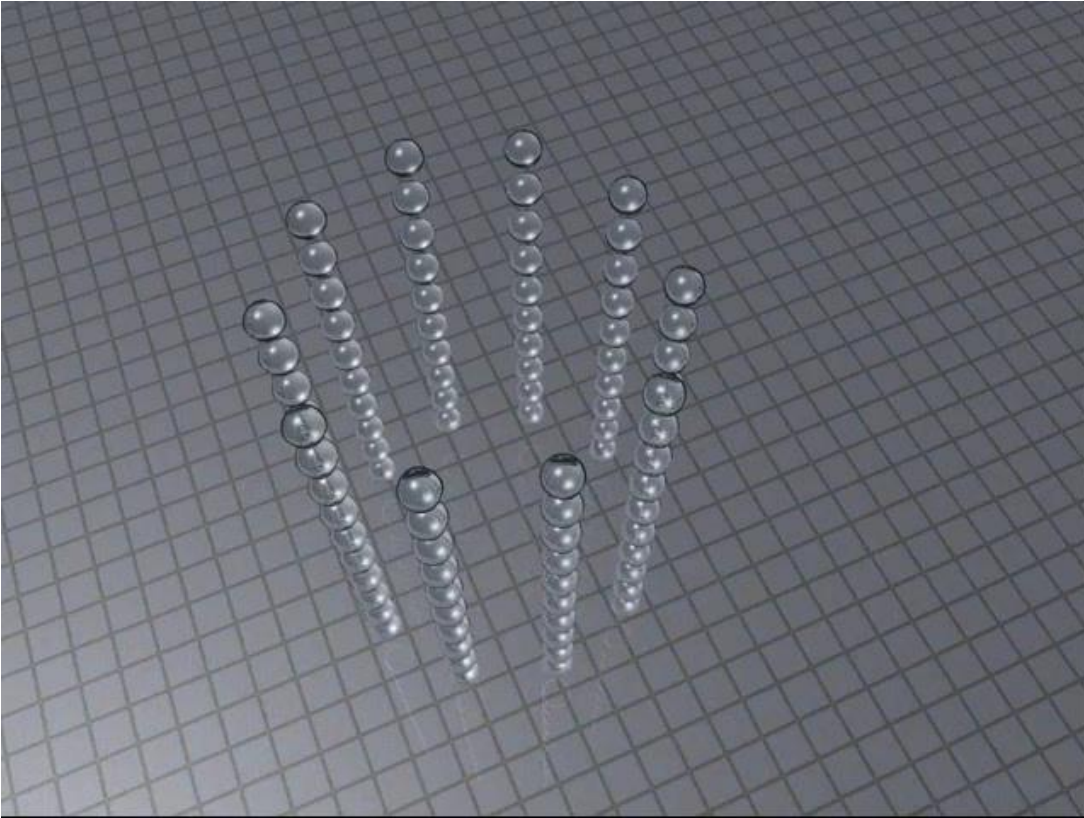
# 用Blend 3D建的Metaball模型



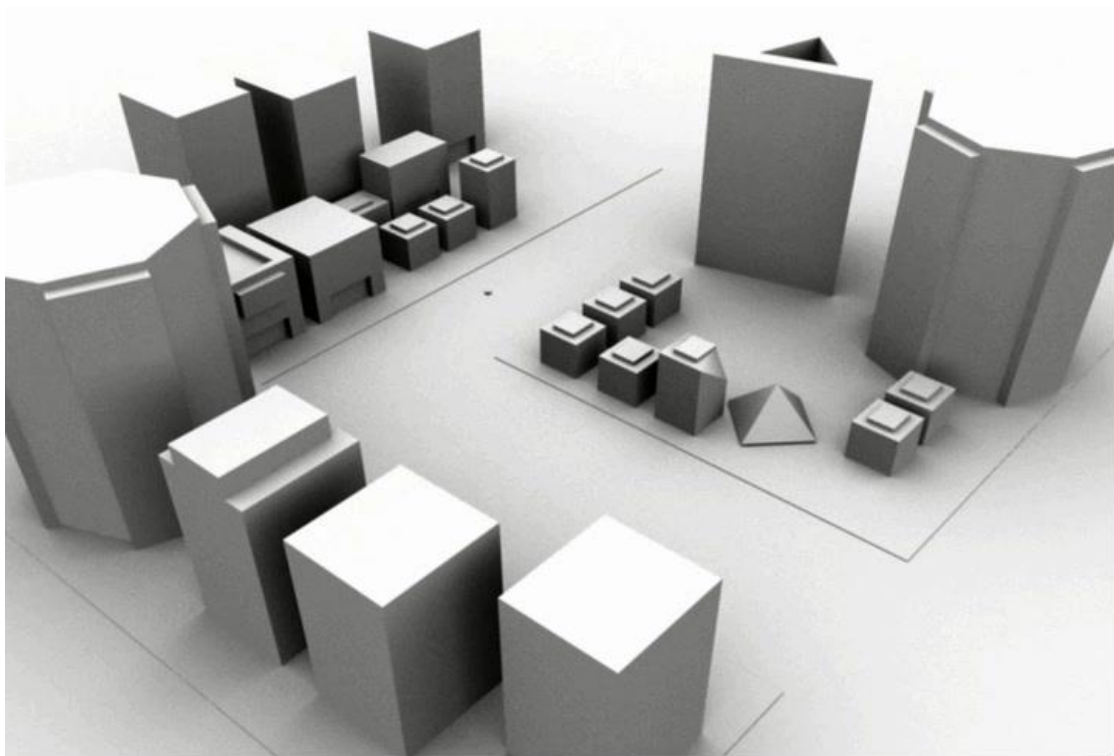
# 元球造型的优点

- 所需的数据量通常要比用多边形造型少2~3个数量级。例如，用500个元球就可以较好地表现一个人的造型。
- 很适合于采用CSG模型来描述。
- 很适合表示可变形的物体，因而对柔性物体的动画非常有用。
- 很适合于人体、动物器官和液体的造型。
- 生成的曲面永远是光滑的。

# Metaballs System for Maya (水滴)

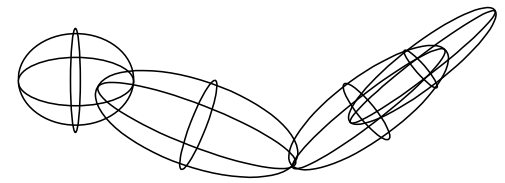


# Metaballs System for Maya



# 元球(Metaball)

- 1982年, Blinn最早提出了元球造型的雏形, 他称之为blobby模型。
- 1985年, Nishimura在元球造型上做出了许多开拓性的工作, 使该技术得到了迅速的发展, 他首先提出了metaball这一单词。
- 在元球造型技术中, 一张自由曲面定义为一张等势面, 用户需指定的参数为包括元球的中心、元球中心的密度、势函数、颜色。
- 元球的造型过程与CSG造型非常类似, 根据所需造型物体的形状, 用户放置元球, 调整元球的大小、方向, 便可很容易得到一张光滑的曲面。在右图中, 只需放置图所示的4个元球, 即可生成一只手臂。



# 元球的势函数

- 在一个元球系统中，每个元球可以有不同的势函数，我们用 $f_i$ 表示第 $i$ 个元球 $P_i$ 的势函数。
- 假设第 $i$ 个元球的中心为 $(x_i, y_i, z_i)$ ，空间某一点 $(x, y, z)$ 与 $P_i$ 的距离用来 $r$ 表示，

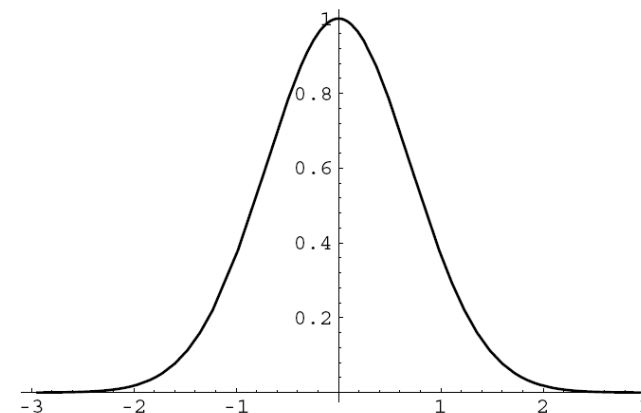
$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$$

- 假设 $R_i$ 是第 $i$ 个元球的有效半径。

# 元球的势函数

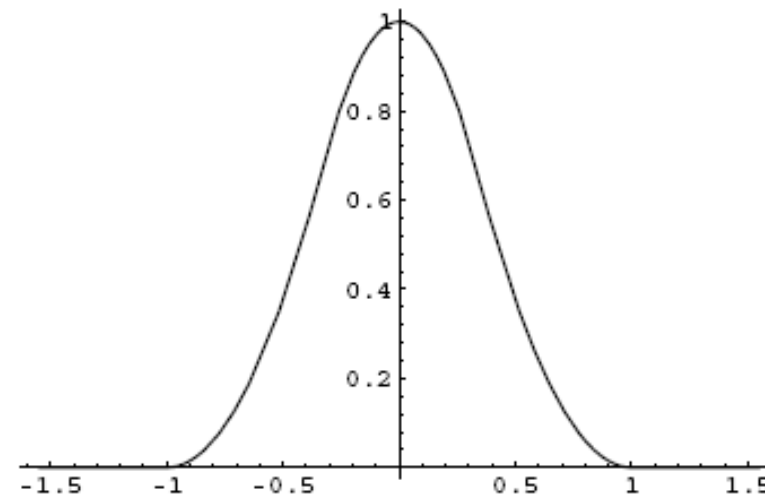
- Blinn的幂函数

$$f_i(r) = b_i \exp(-a_i r^2)$$



- Nishimura的分段二次多项式

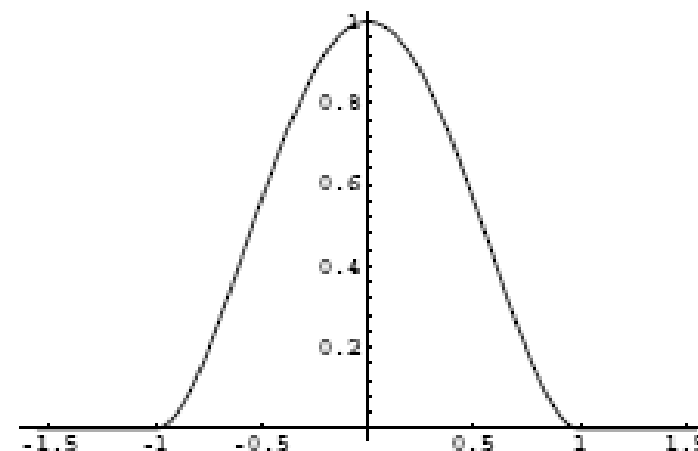
$$f_i(r) = \begin{cases} 1 - 3\left(\frac{r}{R_i}\right)^2, & 0 \leq r \leq \frac{R_i}{3} \\ \frac{3}{2}\left(1 - \left(\frac{r}{R_i}\right)\right)^2, & \frac{R_i}{3} \leq r \leq R_i \\ 0, & r \geq R_i \end{cases}$$



# 元球的势函数

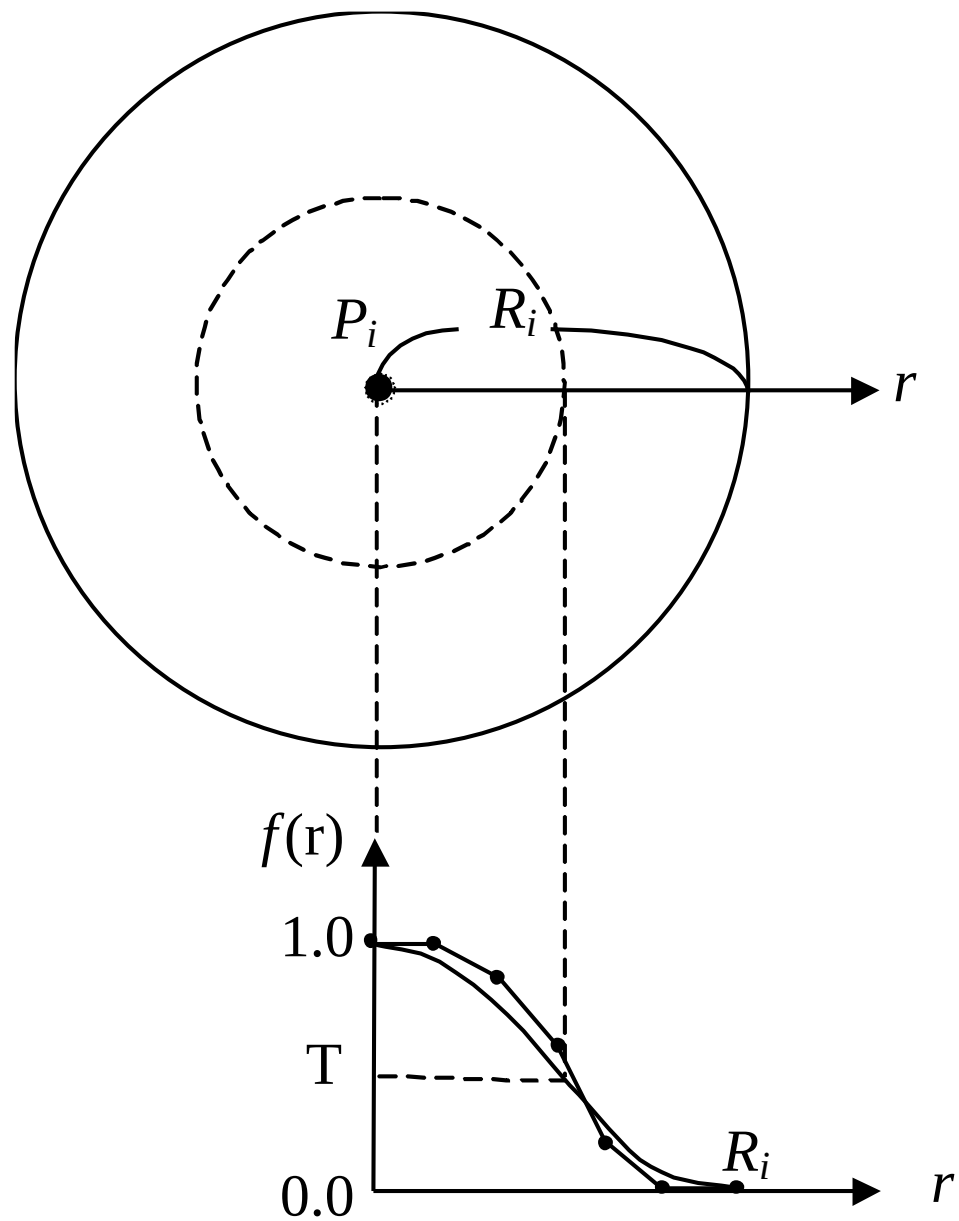
- Murakami的四次多项式

$$f_i(r) = \begin{cases} (1 - (\frac{r}{R_i})^2)^2, & 0 \leq r \leq R_i \\ 0, & r \geq R_i \end{cases}$$



- Wyvill的六次多项式

$$f_i(r) = \begin{cases} -\frac{4}{9}(\frac{r}{R_i})^6 + \frac{17}{9}(\frac{r}{R_i})^4 - \frac{22}{9}(\frac{r}{R_i})^2 + 1, & 0 \leq r \leq R_i \\ 0, & r \geq R_i \end{cases}$$



六次多项式的势函数， $T$ 为阈值，7个点为Bezier曲线的控制顶点

# 元球系统的法向计算

- 对于一个由 $n$ 个元球所构成的元球系统，所对应的等势面为满足以下方程的点集：

$$f(x, y, z) = \sum_{i=1}^n q_i f_i - T = 0$$

其中 $q_i$ 为第 $i$ 个元球的密度值(允许为负值)， $T$ 为阈值。该曲面的法向为：

$$\mathbf{N}(x, y, z) = -\nabla f(x, y, z) = \left( -\frac{\partial f(x, y, z)}{\partial x}, -\frac{\partial f(x, y, z)}{\partial y}, -\frac{\partial f(x, y, z)}{\partial z} \right)$$

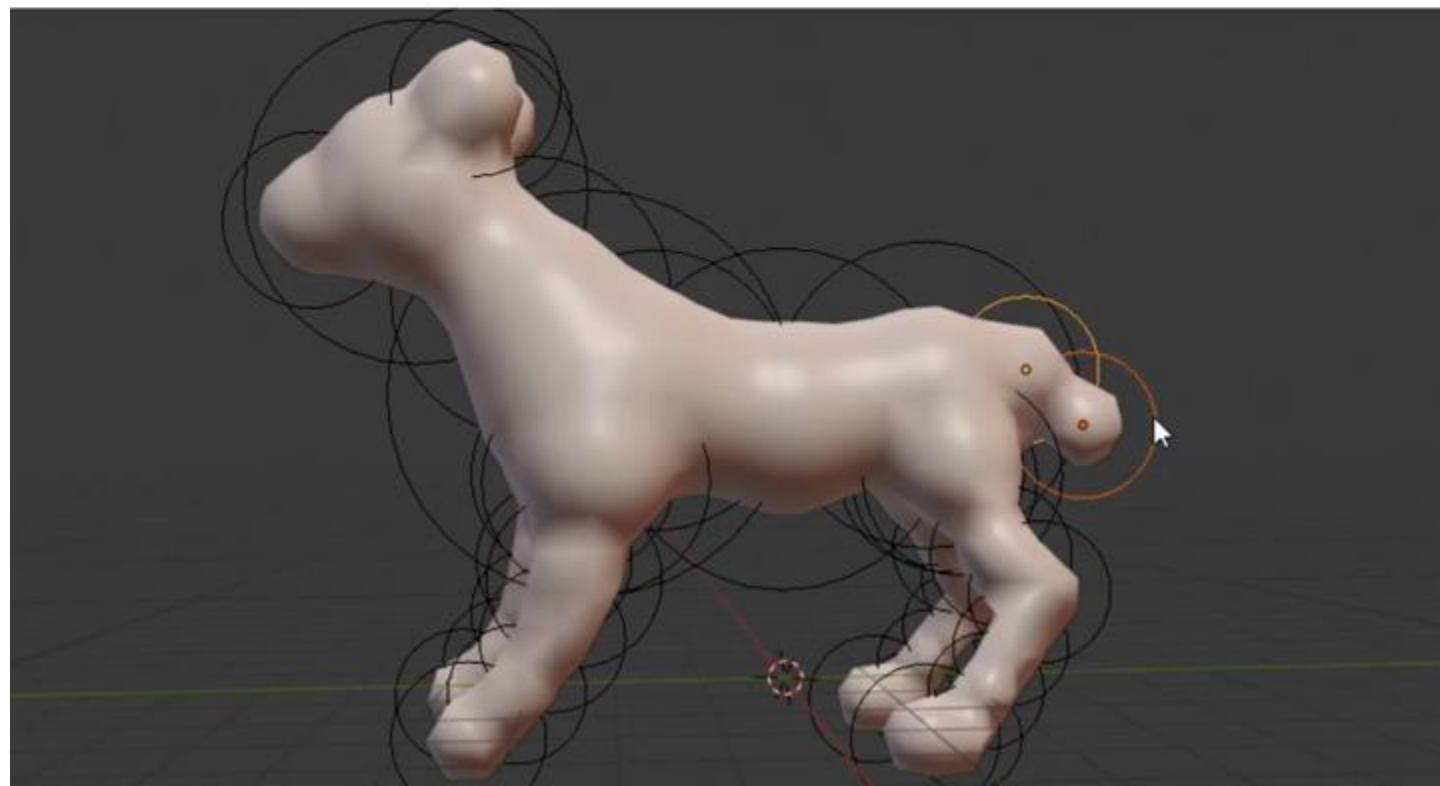
# DAILY BLENDER TIP

## 218

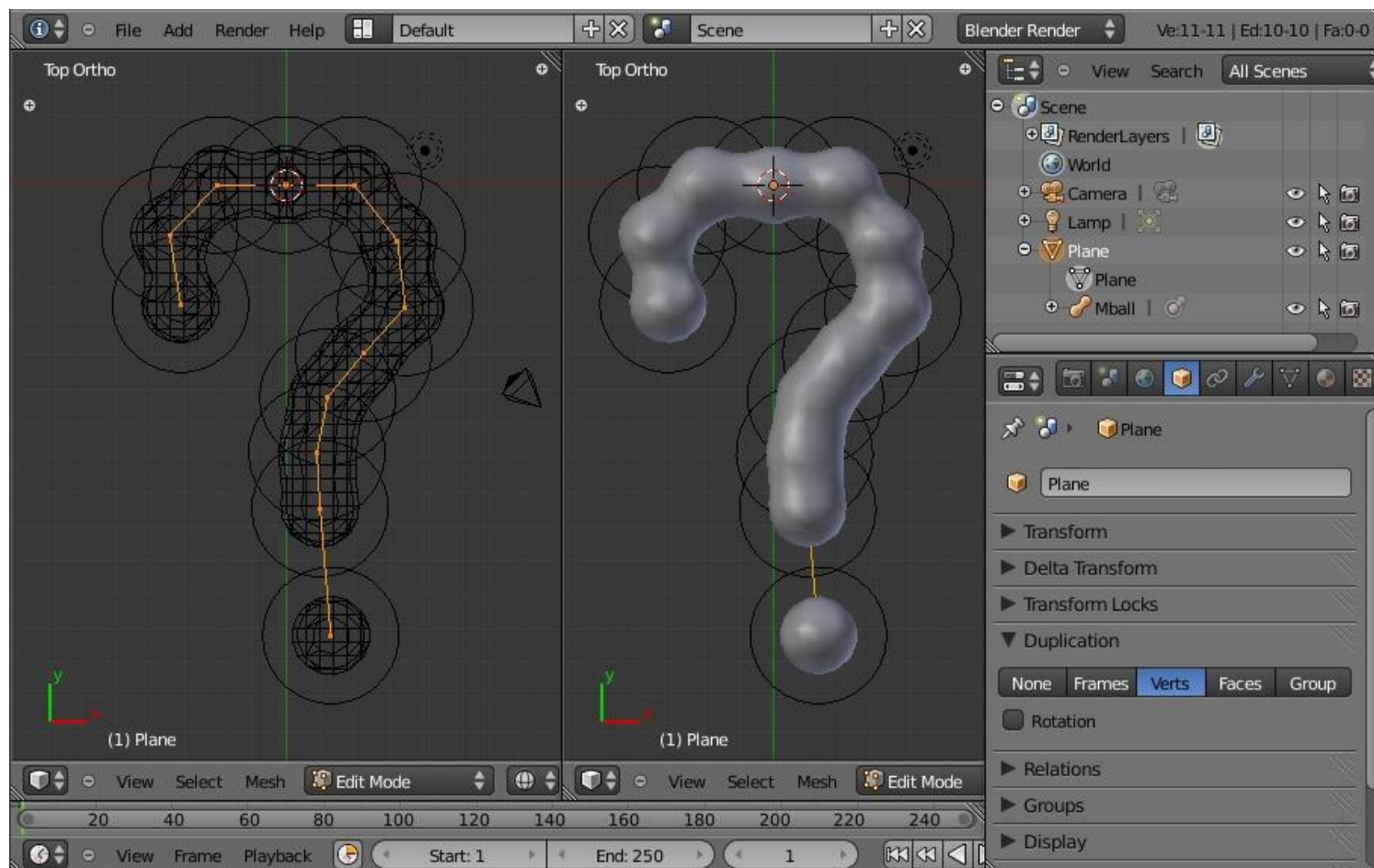
### METABALLS

### QUICK START

**Blender**  
**3D中基于**  
**Metaball**  
**的建模演示**



# 给定几何模型的元球自动拟合法



问题：如何用一些元球去逼近一个物体的形状？

# 给定几何模型的元球自动拟合法

- 在基于元球的动画中，首先需用元球对物体进行造型。
- 但是，给定一个物体的几何模型，要用交互的方法放置元球去描述它并非易事，这促使人们寻找一种用元球去自动拟合给定物体的方法。
- 1991年，Muraki最早提出了一种自动拟合的方法。

\* Muraki S. Volumetric shape description of range data using blobby model. Computer Graphics, 1991, 25(4):227~235

# 位置能量函数

- 他采用的势函数为Blinn的幂函数，此时等势面表示为：

$$f(x, y, z) = \sum_{i=1}^n b_i \exp(-a_i r^2) = T$$

- 给定待拟合物体，我们首先对它进行点采样，得到 $M$ 个采样点  $\Omega = \{x_j, y_j, z_j\}_{j=1}^M$ 。
- 采样点的数目 $M$ 应足够多，使得采样点能基本表示物体的形状。则描述的元球模型应在能量函数  $E_{value} = \sum_{j=1}^M \{f(x_j, y_j, z_j) - T\}^2$  最小的意义上逼近 $\Omega$ 。

# 法向能量函数

- 元球模型有内表面、外表面之分，为了避免元球用内表面去拟合 $\Omega$ ，我们同时考虑物体在采样点的法向。
- 根据物体的几何模型，可求得采样点 $(x_j, y_j, z_j)$ 处的法向 $\mathbf{n}_j$ 。
- 因此，我们应把求解模型作某种修正，不仅使位置能量函数最小，同时也使法

向能量函数：

$$E_{normal} = \sum_{j=1}^M \left\| \mathbf{n}_j - \frac{\mathbf{N}(x_j, y_j, z_j)}{\|\mathbf{N}(x_j, y_j, z_j)\|} \right\|$$

最小，其中 $\mathbf{N}(x_j, y_j, z_j)$ 为元球模型在点 $(x_j, y_j, z_j)$ 的法向。

# 收缩能量函数

- 能量函数 $E_{\text{value}}$ 和 $E_{\text{normal}}$ 只定义于采样点，对没有采样点的空间却没有任何限制。
- 这可能导致这种情况：元球模型很好地逼近了采样点，但在采样点的附近却远离采样点。
- 为了避免该问题，我们将定义一个使每个基元球的影响最小的能量函数 $E_{\text{shrink}}$ 。
- **Blinn**幂函数在整个三维空间的势函数积分为：

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_i(r) dx dy dz = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} b_i \exp(-a_i r^2) dx dy dz = \left( \frac{\pi}{a_i} \right)^{\frac{3}{2}} b_i$$

- 考虑到 $b_i$ 可能为负值，这个新的约束定义为：

$$E_{\text{shrink}} = \left( \sum_{i=1}^n a_i^{-\frac{3}{2}} |b_i| \right)^2$$

该约束的目的是使基元球尽量收缩。

# 复合能量函数

- 因此，所求的元球模型应使能量函数 $E_{value}$ ， $E_{normal}$ 和 $E_{shrink}$ 的和极小：

$$E = \frac{1}{M}(E_{value} + \alpha E_{normal}) + \beta E_{shrink}$$

其中  $\alpha$ 和 $\beta$ 分别为控制表面法向约束和收缩约束相对大小的加权参数。

- 通过改变它们的大小，我们可以得到合适的拟合效果。

# 拟合的求解过程

- 我们必须求得 $n$ 个基元球，使能量函数 $E$ 最小。由于每个基元球有5个参数 $(x_j, y_j, z_j, a_i, b_i)$ ，所以总共有 $5n$ 个未知参数。
- 能量函数的最小化问题是一个非线性问题，无法得到解析解，通常采用如牛顿迭代之类的数值求解方法。但当 $n$ 是一个比较大的数时，很难同时求得 $5n$ 个未知参数。

# 拟合的求解过程

- Muraki提出了这样的求解策略：首先用一个元球去拟合采样点集，然后把该元球分成两个，以增加拟合的精度。对所有元球重复上述分割过程，直至元球的描述满足拟合要求。
- 元球 $P_i$ 的参数可以用下面的五维向量来描述：

$$P_i = (x_i, y_i, z_i, a_i, b_i)$$

# 拟合的求解过程

- 对于初始的元球 $P_1$ , 其位置 $(x_1, y_1, z_1)$ 的初始值取为采样点集 $\Omega$ 的中心,  $a_1$ 的初始值取为 $\Omega$ 方差的倒数,  $b_1$ 的初始值取为 $eT(e \approx 2.718)$ , 通过对能量函数 $E$ 求极小值, 得到的 $P_1$ 五个参数。建立一张元球表, 并把 $P_1$ 加入该表。
- 按某种策略从元球表中选择一个元球 $P_i$ , 把它分成两个新的元球 $P'_i, P''_i$ , 把 $P_i$ 从元球表中删除, 并在表中加入两个新的元球 $P'_i, P''_i$ 。

# 拟合的求解过程

- $P'_i, P''_i$ 的初始参数取为:

$$P'_i = P''_i = (x_i, y_i, z_i, a_i, b_i/2)$$

- 保持元球表中其余元球的参数不变, 通过对能量函数 $E$ 取极小, 求得 $P'_i$ 和 $P''_i$ 的10个参数。重复上述过程, 直至能量 $E$ 达到足够小。

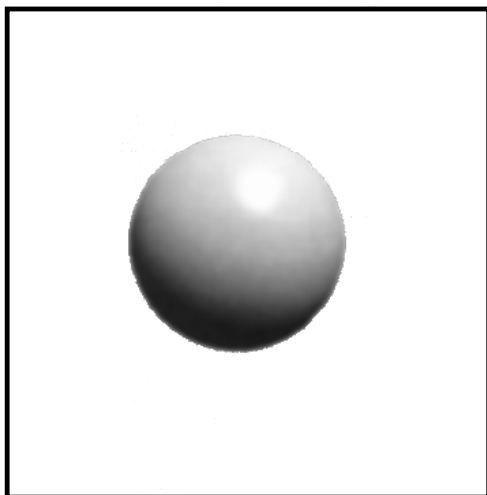
# 算法的伪语言

根据采样点集决定 $P_1$ 的初始参数值;  
通过求解能量最小问题, 用 $P_1$ 拟合采样点集;  
把 $P_1$ 加入元球表;  
while (能量未达到足够小)  
{  
    从元球表中选择元球 $P_i$ ;  
    把 $P_i$ 分成 $P_i'$ ,  $P_i''$ ;  
    从 $P_i$ 元球表中删除, 并把 $P_i'$ 和 $P_i''$ 加入表中;  
    通过求解能量最小问题, 用 $P_i'$ 和 $P_i''$ 拟合采样点集;  
}

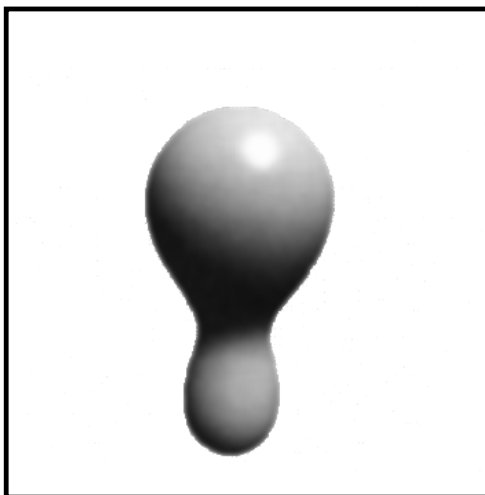
# 基元球的选取次序

- 由于一个元球系统中的势函数互相混合，基元球选取次序对结果的影响很大。
- 如果某种选择次序能以最少数目的元球去很好逼近采样点集，则我们认为该选择方案是最佳的。
- 为了选取最佳的基元球，我们搜索元球表中的所有元球并判断某个元球的分割对系统能量的减少情况，**从中选取使系统能量减小最大的元球为候选元球**。这种选择方案的缺点是当 $n$ 的值很大时计算速度很慢。

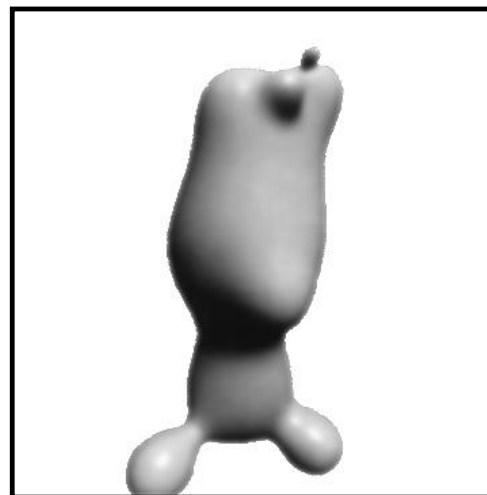
# 例子1



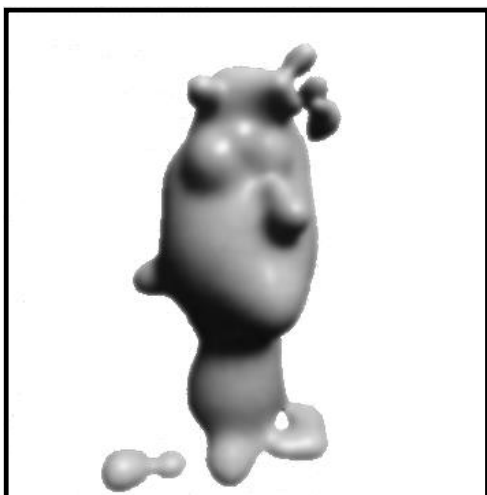
$n=1$



$n=2$



$n=10$



$n=35$

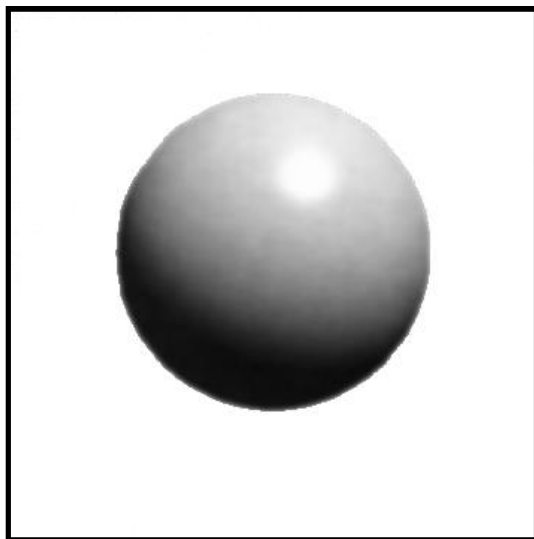


$n=70$

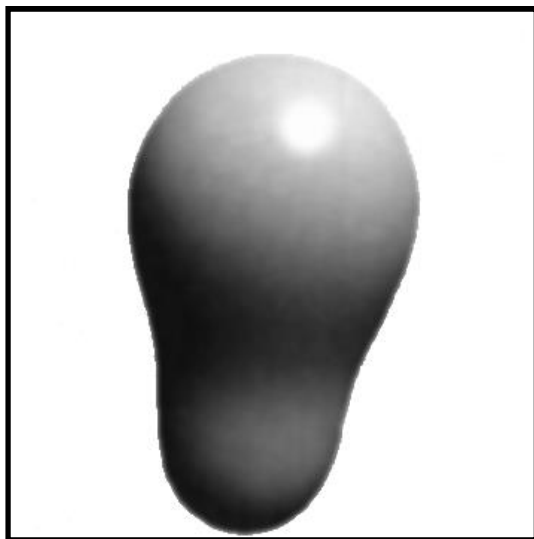


$n=243$

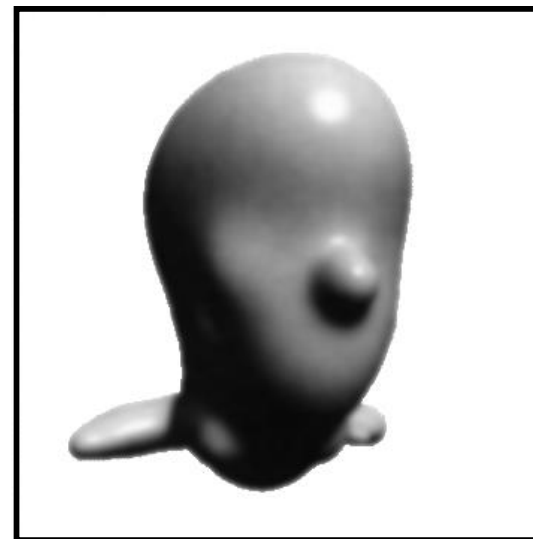
## 例子2



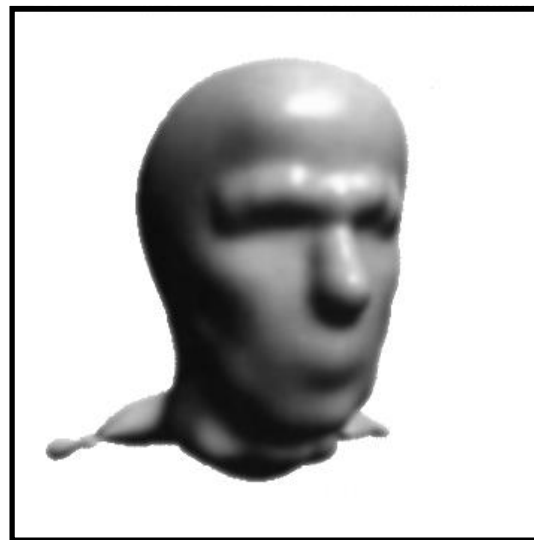
$n=1$



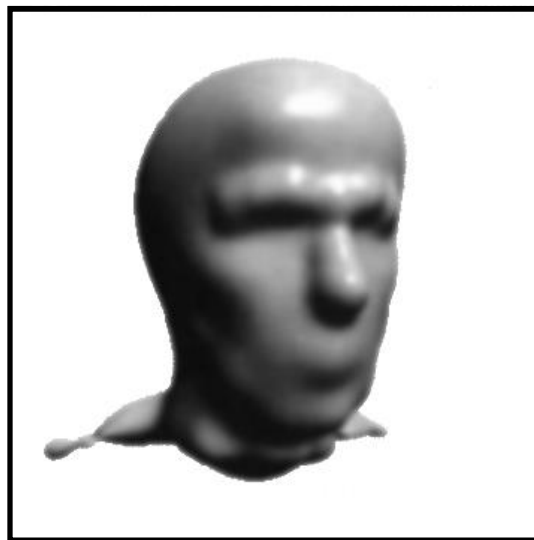
$n=2$



$n=20$



$n=60$



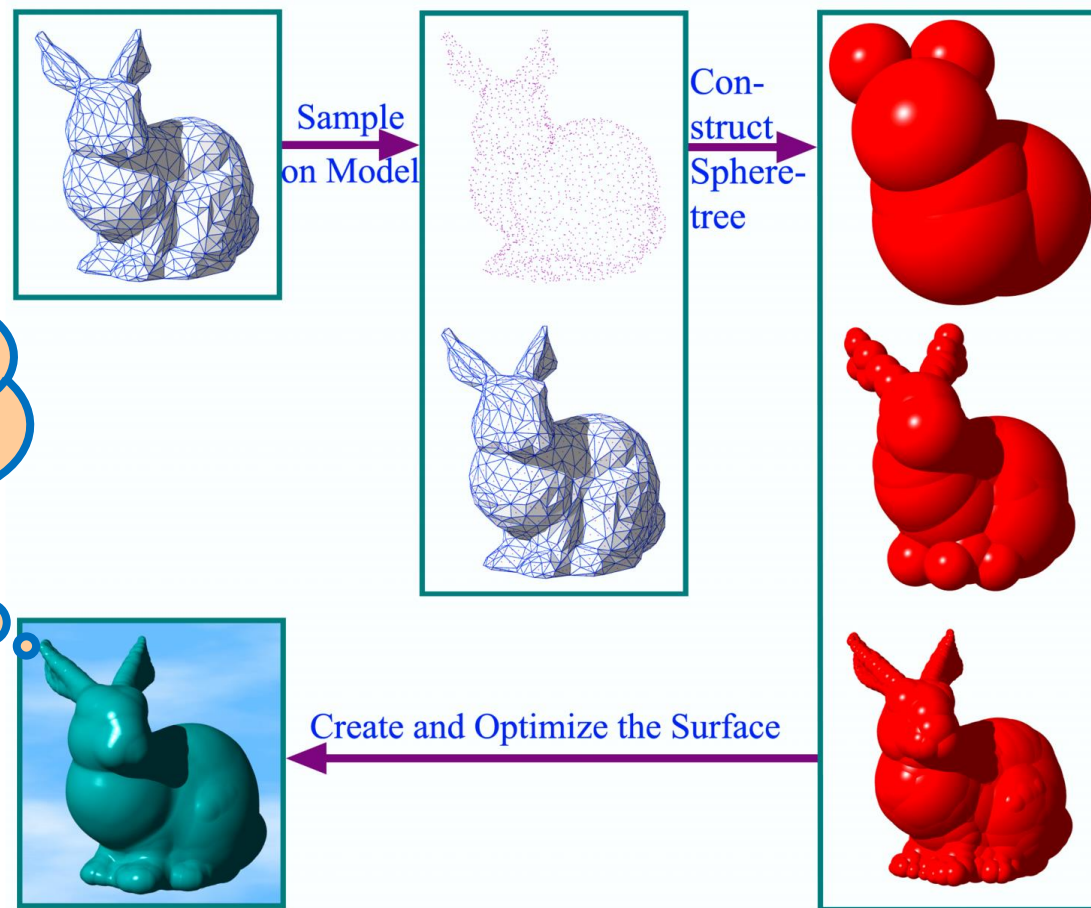
$n=120$



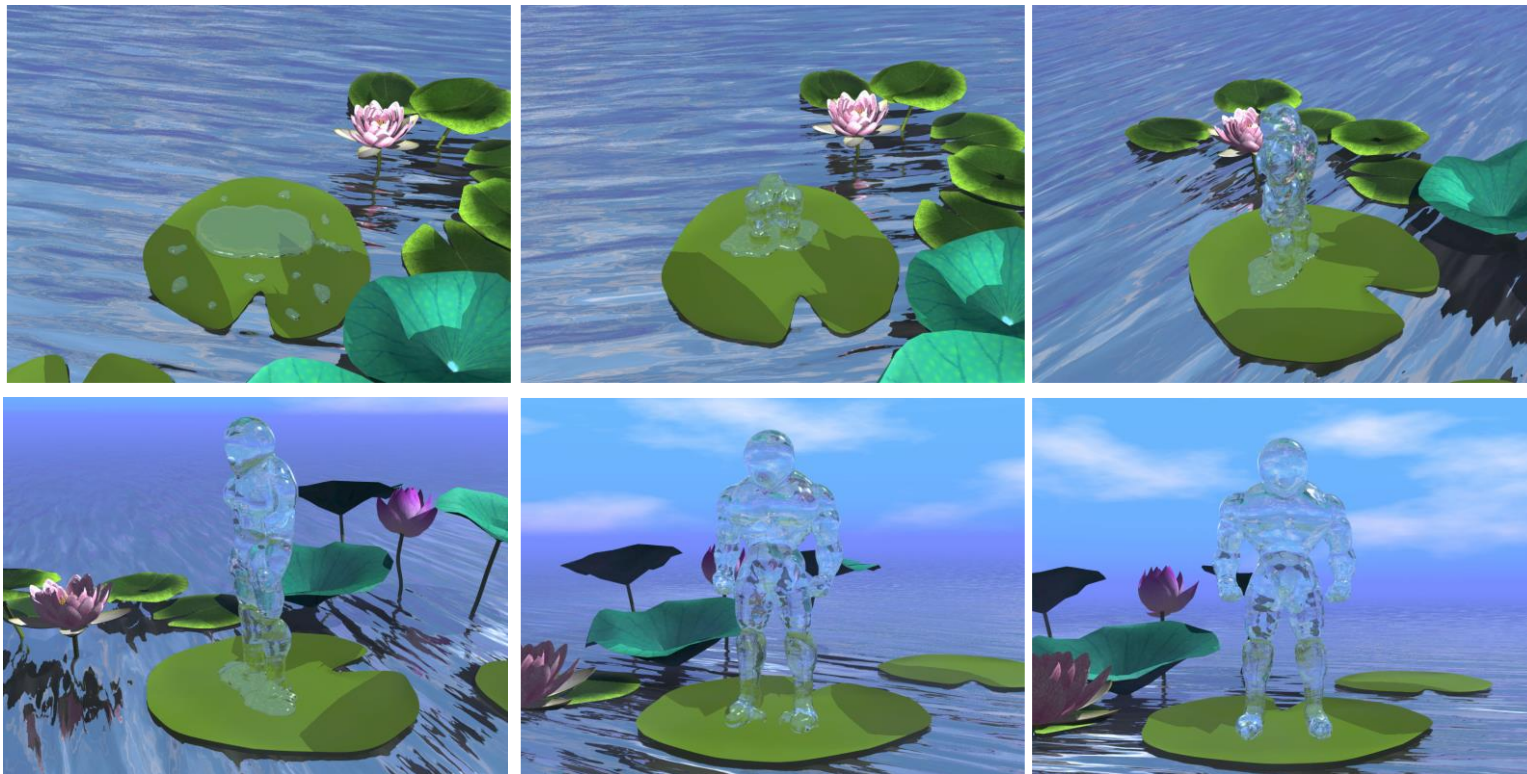
$n=451$

# 球型元球模型曲面重构

球型元球的各向同性，在表示尖锐或扁平的特征时会产生较大变形。



# 结果



Xiaogang Jin, Shengjun Liu, Charlie CL Wang, Jieqing Feng, Hanqiu Sun, "Blob-based Liquid Morphing", Computer Animation and Virtual Worlds, 2005, 16(3-4):391-403

# DEMO



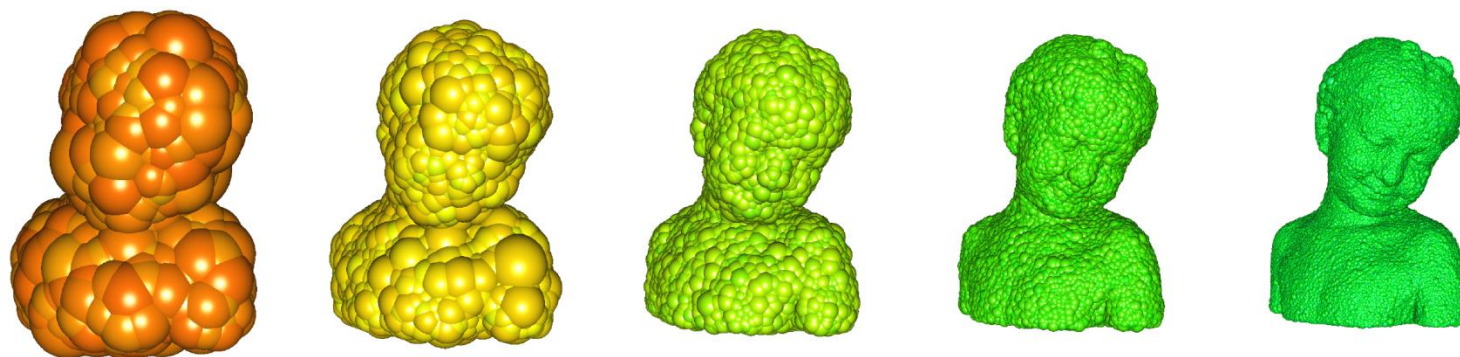
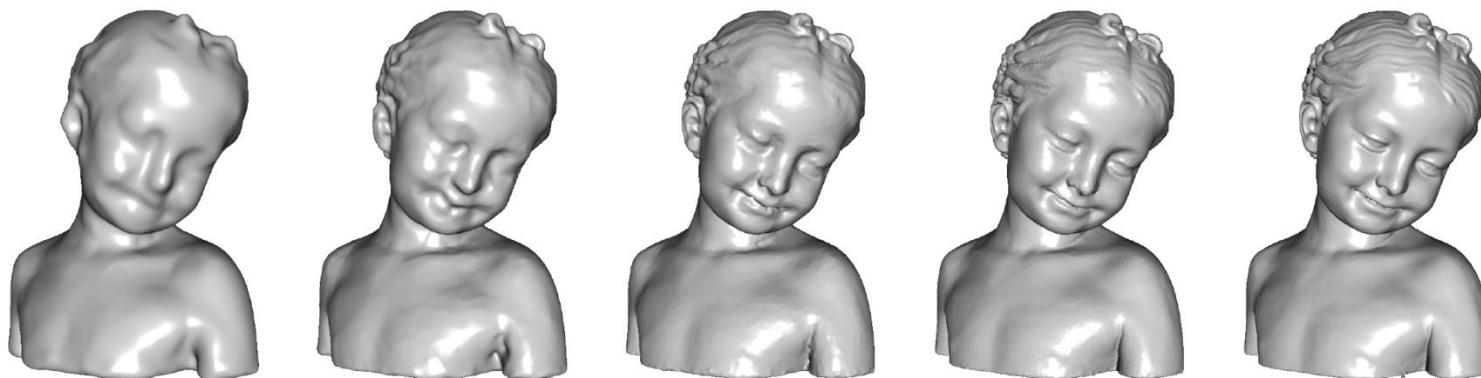
# 椭球型元球模型曲面重构



Shengjun Liu, Xiaogang Jin, Charlie CL Wang, K.C. Hui, "Ellipsoidal-Blob Approximation of 3D Models and Its Applications", Computers & Graphics, 2007, 31(2): 243-251

# 元球模型曲面重构

*Approximation of Implicit Surfaces from 1M polygons*



$$E < 1.0 \times 10^{-1}$$

$$E < 1.0 \times 10^{-2}$$

$$E < 1.0 \times 10^{-3}$$

$$E < 1.0 \times 10^{-4}$$

$$E < 1.0 \times 10^{-5}$$

#nodes: 721

2,258

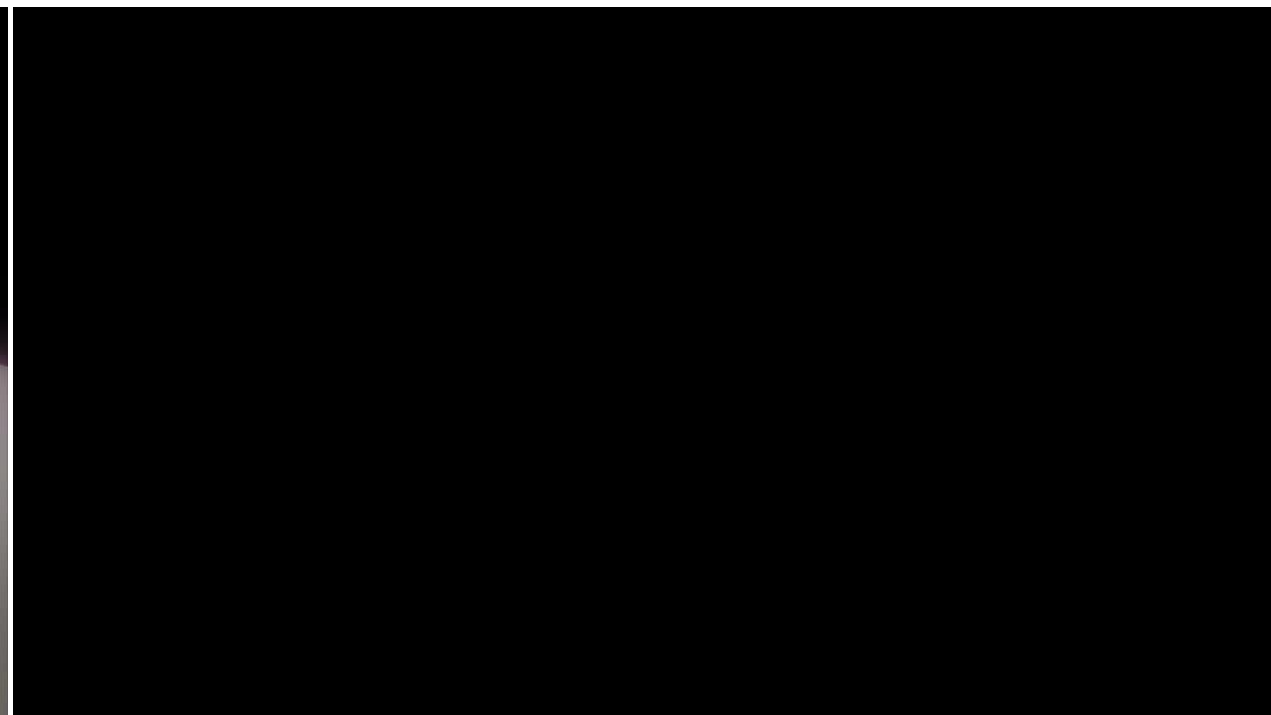
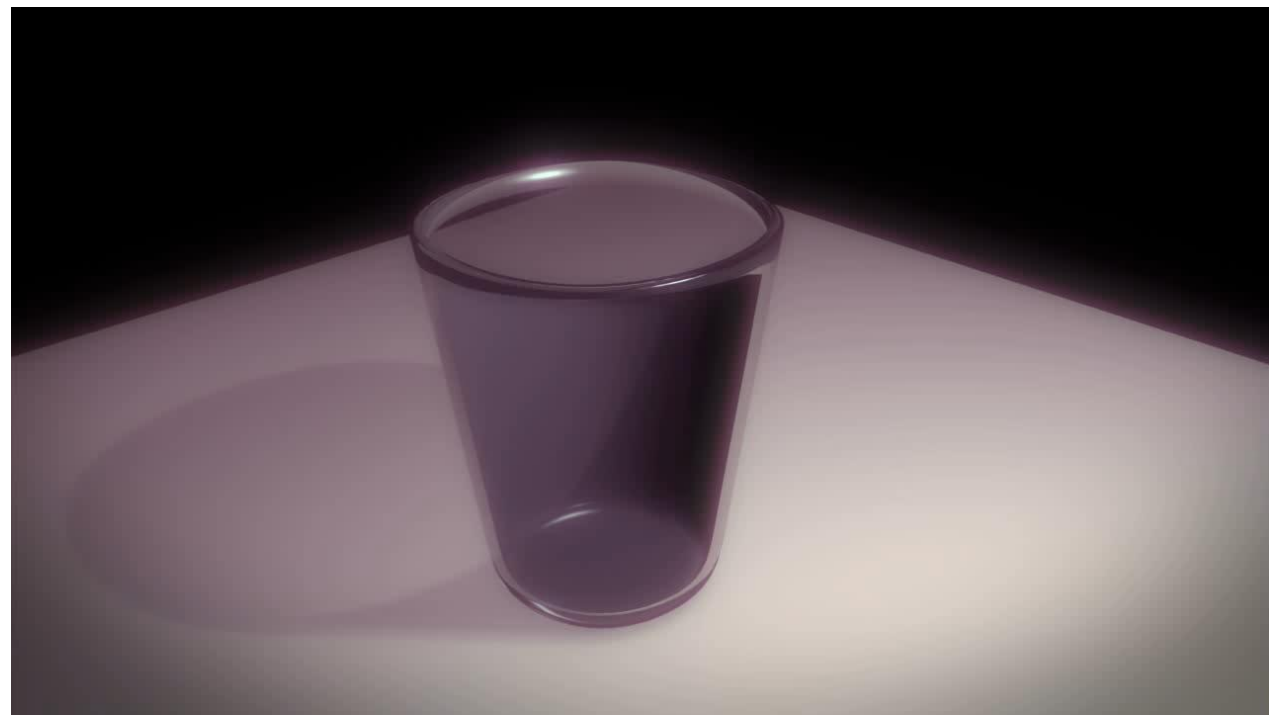
7,076

22,317

70,273

Takashi Kanai, Yutaka Ohtake, Kiwamu Kase: "Hierarchical Error-Driven Approximation of Implicit Surfaces from Polygonal Meshes" Proc. 4th Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp.21-30, Switzerland, 2006

# Metaball动画应用(Blend 3D)



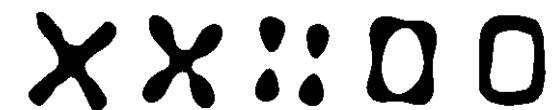
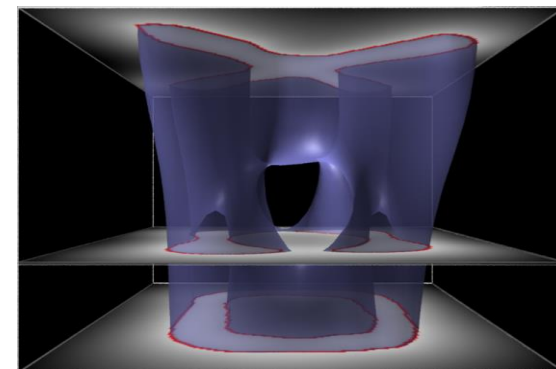
# 隐式曲面总结

- 优点

- 很容易进行点的内外测试(in/out test) (用于碰撞检测等)
- 很容易计算曲面的交/并/差 (用于复杂物体的建模)
- 由于其Blending性质, 很容易处理拓扑变化 (用于动画)

- 缺点

- 曲面通过间接指定
- 很难描述尖锐特征
- 很难对表面上的点进行枚举
- 绘制慢



**参考书:** Bloomenthal, J., Bajaj, C., Blinn, J., Wyvill, B., Cani, M.P., Rockwood, A. and Wyvill, G. eds., 1997. *Introduction to implicit surfaces*. Morgan Kaufmann.

**The End**