

# 基于 RBF 的点模型布尔运算<sup>①</sup>

胡国军 秦绪佳 (浙江工业大学 计算机科学与技术学院 浙江 杭州 310032)

**摘要:** 由已有模型构造新的模型是数字娱乐领域的重要造型方法,提出了一种基于 RBF(径向基函数)的点模型布尔运算方法。该方法首先将输入点模型进行空间八叉树划分,然后采用 RBF 将点模型隐式化生成隐函数表示的符号距离场,最后通过隐式曲面的布尔运算方法剔除多余的点,得到需要的新点模型。对点模型进行空间八叉树划分加速了点模型的 RBF 隐式化过程。该方法可由多个简单的点模型快速构造出复杂的点模型。

**关键词:** 点模型; 隐函数; 径向基函数(RBF); 布尔运算; 八叉树

## Boolean Operations for Point Models Based on RBF

HU Guo-Jun, QIN Xu-Jia

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310032, China)

**Abstract:** To construct a new model from the existing models is an important modeling method in digital entertainment. Boolean operation method for point models based on RBF (Radial Basis Function) is proposed in this paper. Firstly, it divides the input point models by octree. Then it transforms the models to implicit function represented signed distance field. Finally through the implicit surface Boolean operation method to exclude the extra point, a wanted new point model is obtained. The process of RBF implicitization for point model is accelerated by space octree division. A complex model can be constructed from several simple point models with this method.

**Keywords:** point model; implicit function; radial basis function(RBF); Boolean operation; octree

## 1 引言

随着计算机图形学的发展,研究和应用的不断深入,传统的基于面片技术,绘制简单,算法也比较成熟,但也存在着许多瓶颈,如显示复杂的具体细节时,需要大量的几何面片。为了解决这些瓶颈,人们开始研究基于点的绘制和造型技术。Szeliski 等<sup>[1]</sup>在 1992 年提出了有向粒子系统,并用于模型的表面造型及编辑。1994 年 Witkin 等<sup>[2]</sup>将有向粒子用于隐式曲面采样及交互编辑。Pauly 等<sup>[3]</sup>结合无约束点云(点云是指一组离散点集)及移动最小二乘法近似得到隐式曲面,构造出混合表示的实体模型。Ohtake 等<sup>[4]</sup>提出一种新的隐式外形表示方式:利用整体多层分割来构造大规

模点云的表面模型,通过八叉树将点模型分割成若干区域块,并用分段二次函数来描述局部形状,这种方法通过隐函数表示方式拟合点模型表面从而实现了点模型的造型及布尔运算。Adams 等<sup>[5]</sup>实现了基于 surfel 表示实体的交互式布尔运算。杨振羽等<sup>[6]</sup>对 Adams 的算法做了改进,实现了自适应细分加密采样,使求交结果更精确。

随着对点模型曲面重建研究的深入,基于 RBF 的隐式建模已经成为曲面重建的重要方向之一。许多学者在此都作出了重要的贡献: Carr 等<sup>[7]</sup>用 RBFs 重构光滑的多样的表面。Turkand、o' Brien 等<sup>[8]</sup>使用 globally supported RBFs 来重建隐式表面。同时 Morse 等<sup>[9]</sup>使用 CSRBFs 从离散点集插值隐式表面。

① 基金项目:国家自然科学基金(60673063, 60873033);国家高技术发展计划(863)(2007AA12Z141);浙江省自然科学基金(Y1080436)

收稿时间:2009-06-04

本文第 2 节介绍了用径向基函数来建立点模型隐函数曲面方法,第 3 节提出和实现了用八叉树来加速 RBF 曲面拟合,第 4 节描述了隐函数布尔运算的原理,第 5 节提出和实现了基于 RBF 的点模型的布尔运算方法,最后出了基于本文思想的一些实验结果,并对全文进行总结。

## 2 基于径向基函数(RBF)的点模型的隐式化

隐函数拟合就是用 0 值函数来表示曲面的曲面方程方法。设  $f(X)$  定义曲面  $S$ , 若满足  $f(X)=0$ , 则  $X$  在曲面  $S$  上。

$$\begin{cases} \{x | f(x) < 0, x \in R^3\} \text{ 集合 } X \text{ 在曲面外} \\ \{x | f(x) = 0, x \in R^3\} \text{ 集合 } X \text{ 在曲面上} \\ \{x | f(x) > 0, x \in R^3\} \text{ 集合 } X \text{ 在曲面内} \end{cases} \quad (1)$$

传统的用多张独立的代数曲面拼接一张曲面的方法,比较容易构造复杂曲面,但需要同时处理面片间光滑拼接的问题,而 RBF 插值曲面正好可以解决这些问题,并且可以任意指定曲面要通过的点,不要求数据点分布在规则网格上。还有 RBF 比较受欢迎的原因是即使插值数据点排列非常无序都能保证相应的线性方程组系数矩阵可逆。RBF 的插值曲面是一些径向基函数的加权和,可表示为:

$$f(X) = p(X) + \sum_{i=0} \lambda_i \varphi_i (\|X - X_i\|)$$

$p(x)$  为线性项,  $\|\bullet\|$  为欧式范数,  $\lambda_i$  为实值系数。当  $X_i \in R^3$ , 这里  $X_i$  可看作是在三维空间的点。

上式可化为:

$$\begin{bmatrix} \varphi_{01} & \varphi_{02} & \dots & \varphi_{0n} & x_1 & y_1 & z_1 & 1 \\ \varphi_{11} & \varphi_{12} & \dots & \varphi_{1n} & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{n1} & \varphi_{n2} & \dots & \varphi_{nn} & x_n & y_n & z_n & 1 \\ x_1 & x_2 & \dots & x_n & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_n & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \dots & z_n & 0 & 0 & 0 & 0 \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

$$h_i = f(x_i), \quad \varphi_{ij} = \varphi(\|X_i - X_j\|), \quad i, j = 1, 2, \dots, n, \quad p(X) = c_0x + c_1y + c_2z + c_3。$$

简化(2)式为:

$$\begin{bmatrix} A & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} h \\ 0 \end{bmatrix}, \quad A_{ij} = \varphi_{ij}, \quad X = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix}$$

由于  $\begin{bmatrix} A & X \\ X^T & 0 \end{bmatrix}$  为对称阵,对于给定的  $X$  和  $h$ , 只

要适当的选取基函数  $\varphi$ , 就能解得  $\lambda$  和  $c$ , 从而求得  $f(X)$ 。

这里我们选取的是现在普遍采用的的紧支基函数  $\varphi(r) = (1-r)^4(4r+1)$ , 由于紧支基函数使  $\begin{bmatrix} A & X \\ X^T & 0 \end{bmatrix}$  更为稀疏, 比起薄板样条基函数, 大大节省了计算时间和数据存储的花费。用紧支基函数构造系统方程组花费  $o(n \log(n))$ , 解方程组花费  $o(n^{1.5})$ , 存储花费  $o(n)$ , 估值的计算花费  $o(\log(n))$  [10]。

## 3 八叉树拓扑结构建立

在点模型中, 由于许多离散点之间没有显式的拓扑关系, 所以任何点的搜索必须在点的全局范围内进行, 这就造成了曲面拟合速度变慢。所以建立点之间的拓扑关系, 是我们加速曲面拟合的关键。我们用八叉树对点云数据进行递归分割, 建立八叉树拓扑关系, 即加快了点的搜索速度, 从而加快了径向基函数的求解过程。

### 3.1 八叉树的创建

我们首先求出点云数据的最小包围合立方体  $\{x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}\}$ 。然后  $Size_x = x_{\max} - x_{\min}$ ,  $Size_y = y_{\max} - y_{\min}$ ,  $Size_z = z_{\max} - z_{\min}$ 。我们定义八叉树的起始原点为  $Or_x = (x_{\min} + x_{\max}) / 2 - Size_x / 2$ ,  $Or_y = (y_{\min} + y_{\max}) / 2 - Size_y / 2$ ,  $Or_z = (z_{\min} + z_{\max}) / 2 - Size_z / 2$ , 定义的节点最多能放的点数为 8 个, 若大于 8 个就进一步划分子树。Level 为当前节点层次, 令根节点 level 为 0, 下一层就加 1, 依次类推。  $Sx = \frac{1}{2} \times Size_x \times 2^{-level}$ ,  $Sy = \frac{1}{2} \times Size_y \times 2^{-level}$ ,  $Sz = \frac{1}{2} \times Size_z \times 2^{-level}$ , 我们可以把每个子节点看作是棵子树, 八子节点分别创建为  $ChildNode0(level + 1, x, y, z)$ ,  $ChildNode1(level + 1, x + Sx, y, z)$ ,  $ChildNode2(level + 1, x, y + Sy, z)$ ,  $ChildNode3(level + 1, x + Sx, y + Sy, z)$ ,  $ChildNode4(level + 1, x, y, z + Sz)$ ,  $ChildNode5(level + 1, x + Sx, y, z + Sz)$ ,  $ChildNode6(level + 1, x, y + Sy, z + Sz)$ ,  $ChildNode7(level + 1, x + Sx, y + Sy, z + Sz)$ 。

### 3.2 八叉树构建算法描述

我们把八叉树的空间划分看作是将立方体划分成八个相等的子立方体的递归划分。八叉树的空间分割简单算法描述如下:

Step1: 建立点数据的初始列表;

Step2: 求出最小包围合, 然后创建根节点;

**Step3:** 对点数据进行空间划分, 从点数据的初始列表中取出每一点依次加入八叉树中。

**Step4:** 若为叶节点, 且加入的点数少于 8 个, 则放入此叶节点中, 若加入点的数量等于 8, 就进一步划分成子树, 然后转 **Step5**;

**Step5:** 若放入的节点不是叶节点, 则继续向下收索, 直到碰到叶节点。然后转 **Step4**;

**Step6:** 递归完成后, 删除空节点;

完成了八叉树对空间的划分后, 我们就可以对点模型采用第 2 节的方法进行基于 RBF 的曲面拟合。

## 4 基于RBF隐函数的点模型布尔运算

### 4.1 一些早期的隐式化布尔运算思想

设  $f_1 = 0$  和  $f_2 = 0$  为两个隐式曲面, 通过  $f_1$  和  $f_2$  等并、交、差等布尔运算生成  $f_3$ 。Wyvill 等<sup>[1]</sup>给出可以融合函数的极限形式定义隐式函数曲面的布尔运算形式:  $f_3 = (f_1^n + f_2^n)^{\frac{1}{n}}$ 。Pasko 等<sup>[2]</sup>给出了一个包含布尔运算融合函数的一般形式:

$f_3 = (f_1 + f_2 + \alpha \sqrt{f_1^2 + f_2^2}) \sqrt{f_1^2 + f_2^2}$ ,  $\alpha \in [-1, 1]$ 。当  $\alpha = -1$  时, 此式等价于交运算, 当  $\alpha = 1$  时, 此式等价于并运算。Adams 等<sup>[5]</sup>给出了一种内外测试算法来做布尔运算: 设  $A$  和  $B$  为基于 surfels 的表示的基于点模型的集合,  $S_A$  和  $S_B$  为  $A$  和  $B$  所覆盖点模型的表面,  $V_A$  和  $V_B$  为  $S_A$  和  $S_B$  所包围的实体,  $s_1, s_2$  为 surfel,  $s_1 \in S_A, s_2 \in S_B$ 。

$$\begin{cases} A \cup B = \{s_1 + s_2 \mid s_1 \text{ 在 } V_B \text{ 的外部且 } s_2 \text{ 在 } V_A \text{ 的外部}\} \\ A \cap B = \{s_1 + s_2 \mid s_1 \text{ 在 } V_B \text{ 的内部且 } s_2 \text{ 在 } V_A \text{ 的内部}\} \\ A - B = \{s_1 + s_2 \mid s_1 \text{ 在 } V_B \text{ 的外部且 } s_2 \text{ 在 } V_A \text{ 的内部}\} \end{cases} \quad (3)$$

### 4.2 基于 RBF 点模型的布尔运算

本文在内外判别也是采用 Adams 等(3)式的思想, 但实现的算法主要采用(1)式的思想, 设  $X_A$  和  $X_B$  为模型  $A$  和模型  $B$  的点集,  $f_A(x) = 0$  和  $f_B(x) = 0$  定义了模型  $A$  和模型  $B$  的两个隐式曲面,  $x_A \in X_A, x_B \in X_B$ :

$$\begin{cases} A \cup B = \{x_A + x_B \mid f_A(x_B) < 0, f_A(x_A) < 0\} \\ A \cap B = \{x_A + x_B \mid f_A(x_B) > 0, f_A(x_A) > 0\} \\ A - B = \{x_A + x_B \mid f_A(x_B) > 0, f_A(x_A) < 0\} \end{cases} \quad (4)$$

这里我们采用恐龙模型和球模型来做布尔运算, 采用(4)式的思想, 我们分别对这两模型做了并、交、差运算。

我们在 P4 3.0GHZ, 内存为 512M 的电脑上对球模型和恐龙模型的基于 RBF 的曲面拟合如图 1, 点数如表 1, 分别花费 3.3 秒和 18.5 秒, 进行并交差运算都为 2.1 秒。

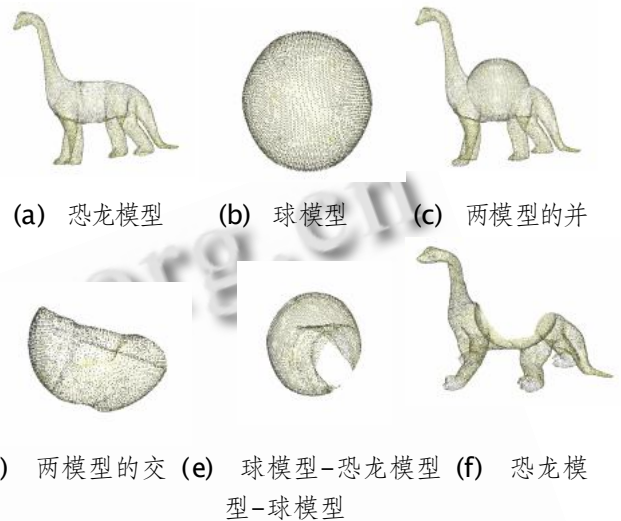


图 1 恐龙模型和球模型的并、交、差布尔运算

表 1 各模型布尔运算前后的点数量

模型	恐龙模型	球模型	龙∪球	龙∩球	球-龙	龙-球
模型点数	23931	40962	50411	14483	31981	32912

### 4.3 点模型布尔运算过程

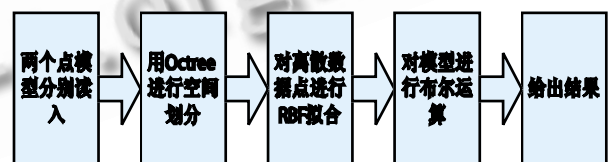


图 2 两点模型的布尔运算流程图

彩用点模型布尔运算生成新的点模型, 其处理过程如图 2 所示。我们将离散数据点进行空间八叉树划分, 建立了点数据的空间拓扑结构, 以加速后续的 RBF 点模型隐式化; 然后将两点模型进行 RBF 隐式化拟合后, 产生了(1)式效果, 就可以用(4)式进行并交差的布尔运算。

## 5 实验结果分析

我们在 P4 3.0GHz, 内存 512MB 的 PC 机上测试, 使用 VC++6.0 和 OpenGL 开发了一个基于 RBF 隐式曲面的点模型布尔运算的快速建模原型系统。图 3 为两

个简单点模型间的布尔运算,图3(a)为 horse 模型,图3(b)为由 feline 模型经交互切割得到的翅膀局部模型 wing,图3(c)为 horse 模型与 wing 模型的交互对准,为布尔运算作准备,图3(d)为布尔运算后的点模型。

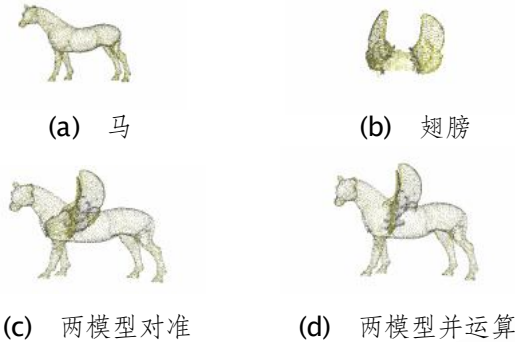


图3 简单布尔运算

在几何建模中,特别是数字娱乐领域,复杂模型经常由多个点模型经过布尔运算来获得。图4说明用三个点模型由我们设计的布尔运算方法构造新模型的过程。我们首先给出一个鸟模型图4(a)、球模型图4(b)和人头模型图4(c),图4(d)为鸟模型与球模型作布尔运算前的交互对准。然后用鸟模型减去球模型得到如图4(e),最后将运算结果与人头模型作并运算,就得到了最终结果图4(f)。

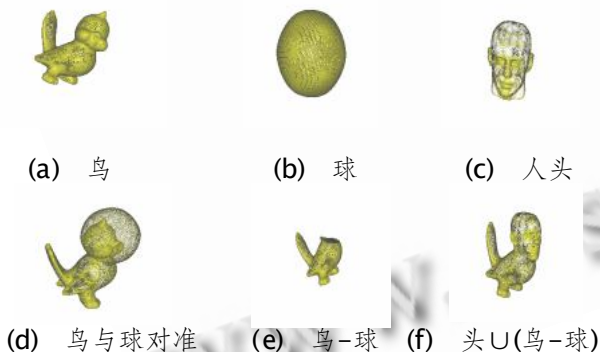


图4 多个点模型的布尔运算

应用该算法,我们可以由已有样例模型库,构建出具有多个模型局部特征的新模型。

## 6 结语

随着三维数据获取设备(如三维扫描仪)的普及和应用,获取的物体表面数据常以点云数据的形式表示,因此点模型在逆向工程、数字娱乐造型领域有着重要应用。本文通过由已有点模型来快速构建新的模型,提出和实现了一种基于 RBF 的点模型布尔运算方法。

基于 RBF 的点模型的布尔运算实现思想简单,但随着点模型数据量的增大,基于 RBF 的隐式化时间也会随之增加,本文采用八叉树结构来加快计算时间。

本文采用的点模型为简单点模型,只包括点的位置和方向,所以本文方法也可以应用于网格模型的布尔运算中,即首先保存网格模型的拓扑点结构,然后采用本文的方法对点模型布尔运算,最后网格化点模型,得到处理后的网格模型。

## 参考文献

- 1 Szeliski R, Tonnesen D. Surface modeling with oriented particle systems. *Computer Graphics*, 1992, 26(2): 185 - 194.
- 2 Witkin PA, Heckbert SP. Using particles to sample and control implicit surfaces. *Computer Graphics*, 1994, 28(7): 269 - 278.
- 3 Pauly M, Keiser R, Kobbelt P L, et al. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics*, 2003, 22(3): 641 - 650.
- 4 Ohtake Y, Belyaev A, Alexa M, et al. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 2003, 22(3): 463 - 470.
- 5 Adams B, Dutré P. Interactive Boolean Operations on Surface-Bounded Solids. *ACM Transactions on Graphics*, 2003, 22(3): 651 - 656.
- 6 杨振羽, 文庭, 群生. 一般点模型的交互式布尔运算. *计算机辅助设计与图形学学报*, 2005, 17(5): 954 - 961.
- 7 Carr JC, Beatson RK, Cherie JB, et al. Reconstruction and representation of 3D objects with radial basis functions. Fiume E. Ed. *Proc. of ACM SIGGRAPH 2001, Los Angeles, California, USA, 001.7 - 76*.
- 8 Turk G, O'Brien JF. Modeling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 2002, 1(4): 855 - 873.
- 9 Morse BS, Yoo TS, Rheingans P, et al. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. *Shape Modeling International 2001, Genova*, 2001. 89 - 98.
- 10 Morse BS, Yoo TS, Rheingans P, Chen D T, Subramanian KR. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Shape Modeling International 2001, Genova, Italy, May 2001. 89-98*.
- 11 Wyvill B, Gallin E, Guy A. Extending the CSG tree: warping, blending and Boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, 1999, 18(2): 149 - 158.
- 12 Pasko A, Adzhiev V, Sourin A, et al. Function Representation in Geometric Modeling: Concepts, Implementation and Applications. *The Visual Computer*, 1995, 11(8): 429 - 446.