基于 GPU 和区间分析的隐式曲面绘制和网格化

秦 阳，蔺宏伟*，冼楚华，高曙明
(浙江大学 CAD & CG 国家重点实验室 杭州 310058)
(hwlin@zjucadcg.cn)

摘 要：为了通过并行化技术提高隐式曲面绘制和网格化的速度，提出一种基于 GPU 并行计算架构的区间分析方法来网格化和绘制隐式曲面.首先按照给定的绘制分辨率将绘制空间离散成体素表示，充分利用 GPU 的并行计算能力，采取区间分析方法并行计算隐函数在所有体素上的取值区间，从而确定出包含隐函数零等值面的特征体素；进一步，抽取特征体素的外表面对其进行拓扑校正，确保得到的网格是二维流形；然后使用 Laplace 操作对这个网格进行光滑处理，得到隐式曲面的网格表示.大量实验结果表明，隐式曲面的网格化和绘制时间一般小于 0.1 s，达到了实时化的水平.

关键词：隐式曲面；绘制；网格化；拓扑校正；几何造型
中图法分类号：TP391

Rendering and Polygonization of Implicit Surface by Interval Analysis Based on GPU

Qin Yang, Lin Hongwei*, Xian Chuhua, and Gao Shuming
(*State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou* 310058)

Abstract: Implicit surface has a simple expression, and meets the smoothness requirements naturally. However, rendering implicit surface is computationally complex and inefficient. More importantly, it is difficult to control and adjust the shape of the implicit surface. Therefore, the modeling technology of implicit surface is far from being practical. To overcome these deficiencies, this paper presents an interval analysis method for rendering and polygonizing implicit surfaces based on GPU parallel computing architecture, which greatly improves the computational efficiency and rendering speed. The method first divides the rendering-space into voxels with a given resolution. Then, taking full advantage of GPU's parallel computing ability, the algorithm uses an interval analysis method to compute the range of implicit function in every voxel in parallel to find the feature-voxels which contain the zero-isosurface of the implicit function. Next, we extract the outer surface of feature-voxels, and handle it with topology-correction operations to ensure that the mesh is two-dimensional manifold; at last, using Laplace operation to smooth the mesh, we get the mesh representation of the implicit surface. A large number of experiments have been done, and the results show that the time of the implicit surface rendering and mesh smoothing are usually in millisecond magnitude which achieves the real-time level.

Key words: implicit surface; rendering; polygonization; topology correction; geometric modeling

收稿日期：2010-07-06；修回日期：2010-08-24.基金项目：国家自然科学基金(60970150，60933008，60736019)；浙江省自然科学基金(Y1090416).秦 阳(1986—)，男，硕士研究生，主要研究方向为几何设计；蔺宏伟(1973—)男，博士，副研究员，硕士生导师，论文通讯作者，CCF 会员，主要研究方向为几何设计；冼楚华(1982—)，男，博士研究生，主要研究方向为 CAD；高曙明(1964—)，男，博士，研究员，博士生导师，主要研究方向为 CAD.

隐式曲面,特别是多项式形式表示的代数曲面,其表达形式简单,可以方便地判定点与隐式曲面的相对位置,而且曲面的光滑性自然满足;不仅如此,隐式曲面还便于表示复杂拓扑曲面,并且易于改变曲面的拓扑结构.但是,隐式曲面的绘制比较复杂,难以直观地控制和调整曲面的形状.此外,与NURBS曲面类似,隐式曲面之间求交同样存在数值不稳定的问题.因此,除一些特殊形式的隐式曲面在计算机动画领域得以应用,如Metaball(元球)方法,一般的隐式曲面造型方法还远未进入实用领域.

由于隐式曲面的绘制较复杂,使得高质量的绘制难以实时化.目前,对隐式曲面的直接绘制方法主要采用基于光线投射算法,从屏幕的每一个像素发射一条光线到隐式曲面上,通过计算并显示这条光线与隐式曲面的最近交点,在屏幕上绘制出这张隐式曲面的可见部分.对于一些特殊类型的隐式曲面,Kalra等[1]设计了一种算法,保证能找到投射光线和隐式曲面的最近交点.Hart发明了Spheretracing技术对隐式曲面进行高效绘制[2].最近,Gamito等[3]提出一种渐进式算法来绘制满足Lipschitz连续条件的隐式曲面.有关基于光线投射算法绘制隐式曲面的方法可参考文献[3].

为计算光线与隐式曲面的交点,Mitchell引入了区间算术方法[4].对于一个给定的区间,区间算术可以计算出函数在这个区间取值的范围.区间算术最初用于在数值计算中控制误差传播[5],由于其计算可靠性,被广泛应用于计算机图形学[6].但是,经典的区间算术对于区间界的估计过于保守.仿射算术是区间算术的一种改进,它的计算效率更高,计算更精确[7].进一步,仿射算术被表示成矩阵形式,称为矩阵形式的仿射算术,提高了计算精度[8-9].不同于光线投射方法,利用区间和仿射算术可以确定出所有与隐式曲面相交的体素,通过绘制这些体素可以达到绘制隐式曲面的目的.Shou等[10-11]利用矩阵形式(张量形式)的仿射算术和四(八)叉树搜索算法确定出与代数曲线曲面相交的特征体素,从而达到绘制隐式曲线曲面的目的.

另一种间接绘制隐式曲面的方法是首先将隐式曲面网格化,然后显示这个网格,以实现对隐式曲面的绘制.不仅如此,更为重要的是,由于隐式曲面的几何形状难以控制,隐式曲面网格化是获取隐式曲面的几何形状,并对其进行控制和操作的有效手段.隐式曲面网格化方法一般可以分为曲面追踪方法、铺嵌方法、基于Morse理论的方法和Delaunay细化方法4类.曲面追踪方法从一个种子点开始,依次计算隐式曲面上相邻的点,将它们连成网格[12-14],该方法得到的隐式曲面网格的完整性取决于种子点的选取,如果隐式曲面有多个分支,这种方法难以获得曲面的全部分支.基于铺嵌的方法首先将隐式曲面所在空间体素化,然后计算与隐式曲面相交的体素单元,最后从这些单元中抽取等值面片来近似表示隐式曲面.著名的Marching Cube方法就属于这一类[15].虽然这种方法效率高并易于实现,但是它生成的网格模型存在很多缺点,比如拓扑歧义性、难于表示尖锐边等.近几十年来,研究人员做了很多努力,使铺嵌方法生成的网格不仅可以表示尖锐的边[16],还消除其拓扑歧义性[17-18].此外,网格的质量也可以用后处理方法来提高,比如面片抽取[19]、重新网格化[20-21]等.第三类方法根据Morse理论,通过计算隐式方程的level sets,并在关键点处改变拓扑结构.Stander等提出的方法可以使生成的网格同胚于原始隐式曲面,但是没有严格证明[22].在假定所有的拓扑关键点已知的前提下,文献[23]方法可以保证生成的网格是ISOtropic各向同性的.在一定的假设条件下,Delaunay细化方法可以生成拓扑正确的网格[24-25].

本文算法首先利用张量形式的仿射算术和区间泰勒模型[26]确定与隐式曲面相交的体素,称为特征体素.传统的方法[11]是利用八叉树搜索算法对与隐式曲面相交的子空间进行递归分割来确定包含隐式曲面的体素,计算效率低.本文算法利用GPU的并行计算能力首先将隐式曲面所在空间体素化,然后利用张量形式的仿射算术和区间泰勒模型,在GPU上并行计算隐式曲面在这些体素的取值范围,确定出特征体素,极大提高了计算效率,计算时间从几秒、几百秒缩短到零点几秒,提高了几十到几百倍.特征体素的外表面是对隐式曲面的近似,对其进行Laplace光滑操作后可以得到光滑的网格曲面.本文设计的特征体素外表面抽取算法保证得到的网格曲面是二维流形,不存在拓扑歧义性.另外,区间分析的性质使生成的网格曲面不会丢失隐式曲面的任何分支.

1 区间分析及其GPU实现

本节首先简要介绍张量形式的仿射算术和一阶区间泰勒模型,然后详细叙述它们的GPU实现算法.关于张量形式仿射算法以及区间泰勒模型的详

细内容,请参阅文献[9,11,26].

1.1　张量形式的仿射算术

张量形式的仿射算术是对代数曲面进行区间分析的有效方法.对于一个三元多项式

$$f(x,y,z)=\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l}\boldsymbol{A}x^iy^jz^k,$$

$$(x,y,z)\in\Omega=[\underline{x},\bar{x}]\times[\underline{y},\bar{y}]\times[\underline{z},\bar{z}];$$

其中,$\boldsymbol{A}=[a_{ijk}]$是该方程的系数矩阵,Ω 是自变量取值区间.为将其转化成张量形式的仿射算术表示,首先定义中间变量

$$x_0=(\bar{x}+\underline{x})/2,\ y_0=(\bar{y}+\underline{y})/2,\ z_0=(\bar{z}+\underline{z})/2,$$

$$x_1=(\bar{x}-\underline{x})/2,\ y_1=(\bar{y}-\underline{y})/2,\ z_1=(\bar{z}-\underline{z})/2;$$

和矩阵 $\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$,它们的元素 b_{ij},c_{ij} 和 d_{ij} 分别为

$$b_{ij}=\begin{cases}\binom{j}{i}x_0^{j-i}x_1^i, i\leqslant j,\\ 0, i>j,\end{cases}\quad i=0,1,\cdots,n,j=0,1,\cdots,n,$$

$$c_{ij}=\begin{cases}\binom{i}{j}y_0^{i-j}y_1^j, i\geqslant j,\\ 0, i<j,\end{cases}\quad i=0,1,\cdots,m,j=0,1,\cdots,m,$$

$$d_{ij}=\begin{cases}\binom{j}{i}z_0^{j-i}z_1^i, i\leqslant j,\\ 0, i>j,\end{cases}\quad i=0,1,\cdots,l,j=0,1,\cdots,l.$$

张量 $\boldsymbol{G}=[G_{ijk}]$由 $\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$ 3 个矩阵和系数矩阵 $\boldsymbol{A}$ 计算得到,即 $\boldsymbol{G}=\boldsymbol{B}\otimes_x(\boldsymbol{D}\otimes_z\boldsymbol{A})\otimes_y\boldsymbol{C}$.

由此可得到张量形式的仿射算术

$$f(\hat{x},\hat{y},\hat{z})=\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l}\boldsymbol{G}_{ijk}\varepsilon_x^i\varepsilon_y^j\varepsilon_z^k;$$

其中 $\varepsilon_x,\varepsilon_y,\varepsilon_z$ 是取值范围在$[-1,1]$的随机量,并且

$$\begin{cases}\hat{x}=x_0+x_1\varepsilon_x\\ \hat{y}=y_0+y_1\varepsilon_y\\ \hat{z}=z_0+z_1\varepsilon_z\end{cases}.$$

如果 i,j,k 是偶数,则 $\varepsilon_x,\varepsilon_y,\varepsilon_z$ 的取值范围为$[0,1]$;否则,它们的取值范围是$[-1,1]$.由此可以计算出多项式 $f(x,y,z)$在区间$[\underline{x},\bar{x}]\times[\underline{y},\bar{y}]\times[\underline{z},\bar{z}]$的取值范围$[\underline{F},\bar{F}]$,即

$$\begin{aligned}\bar{F}=&\boldsymbol{G}_{000}+\sum_{k=1}^{l}\begin{cases}\max(0,\boldsymbol{G}_{00k}), k\text{ 是偶数}\\ |\boldsymbol{G}_{00k}|, k\text{ 是奇数}\end{cases}+\\ &\sum_{j=1}^{m}\sum_{k=0}^{l}\begin{cases}\max(0,\boldsymbol{G}_{0jk}), j,k\text{ 是偶数}\\ |\boldsymbol{G}_{0jk}|, j,k\text{ 不全是偶数}\end{cases}+\\ &\sum_{i=1}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l}\begin{cases}\max(0,\boldsymbol{G}_{ijk}), i,j,k\text{ 是偶数}\\ |\boldsymbol{G}_{ijk}|, i,j,k\text{ 不全是偶数}\end{cases}\quad(1)\\ \underline{F}=&\boldsymbol{G}_{000}+\sum_{k=1}^{l}\begin{cases}\min(0,\boldsymbol{G}_{00k}), k\text{ 是偶数}\\ -|\boldsymbol{G}_{00k}|, k\text{ 是奇数}\end{cases}+\\ &\sum_{j=1}^{m}\sum_{k=0}^{l}\begin{cases}\min(0,\boldsymbol{G}_{0jk}), j,k\text{ 是偶数}\\ -|\boldsymbol{G}_{0jk}|, j,k\text{ 不全是偶数}\end{cases}+\\ &\sum_{i=1}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l}\begin{cases}\min(0,\boldsymbol{G}_{ijk}), i,j,k\text{ 是偶数}\\ -|\boldsymbol{G}_{ijk}|, i,j,k\text{ 不全是偶数}\end{cases}\quad(2)\end{aligned}$$

1.2　一阶区间泰勒模型

对于一般的隐式曲面,比如超越方程定义的曲面,可以采用区间泰勒模型进行区间分析.假设解析函数 $f(x,y,z)$在区间 $\Omega=[\underline{x},\bar{x}]\times[\underline{y},\bar{y}]\times[\underline{z},\bar{z}]$ 具有一阶连续偏导数,将 $f(x,y,z)$在区间 Ω 的中心点(x_0,y_0,z_0)处零阶泰勒展开得

$$f(x,y,z)=f(x_0,y_0,z_0)+(x-x_0)f_x(\varepsilon_x,\varepsilon_y,\varepsilon_z)+$$
$$(y-y_0)f_y(\varepsilon_x,\varepsilon_y,\varepsilon_z)+(z-z_0)f_z(\varepsilon_x,\varepsilon_y,\varepsilon_z);$$

其中,$x_0=\dfrac{\bar{x}+\underline{x}}{2}$,$y_0=\dfrac{\bar{y}+\underline{y}}{2}$,$z_0=\dfrac{\bar{z}+\underline{z}}{2}$,$\varepsilon_x\in[\underline{x},\bar{x}]$,$\varepsilon_y\in[\underline{y},\bar{y}]$,$\varepsilon_z\in[\underline{z},\bar{z}]$.

估计出函数 $f(x,y,z)$的 3 个一阶偏导数 $f_x(x,y,z)$,$f_y(x,y,z)$和 $f_z(x,y,z)$在区间 Ω 上的取值范围 B_x,B_y 和 B_z 后,函数 $f(x,y,z)$在区间 Ω 的取值范围可以表示为

$$[\underline{F},\bar{F}]=f(x_0,y_0,z_0)+hB_x+kB_y+lB_z\quad(3)$$

其中,

$$h=[\underline{x},\bar{x}]-x_0=\frac{\bar{x}-\underline{x}}{2}[-1,1],$$

$$k=[\underline{y},\bar{y}]-y_0=\frac{\bar{y}-\underline{y}}{2}[-1,1],$$

$$l=[\underline{z},\bar{z}]-z_0=\frac{\bar{z}-\underline{z}}{2}[-1,1].$$

式(3)称为一阶区间泰勒模型.

1.3　基于 GPU 的并行绘制算法

基于 CPU 实现的区间分析方法绘制隐式曲面是通过八叉树递归细分算法进行的[11].首先计算函数在整个定义域上的取值区间,如果这个区间包含零值,将其细分为 8 个子体素,计算函数在这些子体素上的取值区间,并确定出包含零值的体素,称为特征体素.不断细分特征体素,直到达到一定的精度.由于这种算法递归进行,效率很低,绘制一张隐式曲面至少需要几分钟,甚至几个小时.

本文算法利用 GPU 的并行计算能力来提高区间分析方法绘制隐式曲面的速度,其基本思想是:对于隐式曲面所在的绘制空间,按照给定分辨率将其细分成体素表示;若曲面方程是多项式方程,将其转化成张量形式的仿射算术,根据式(1)(2)计算出多项式在每个体素上的取值区间;若曲面方程是超越

方程，则采用一阶区间泰勒模型，根据式(3)计算超越函数在每个体素的取值区间. 值得指出的是，计算函数在所有体素的取值区间是在 GPU 计算架构下并行实现的，该架构提供多层次的众多线程用于完成并行计算任务，为每个体素的计算任务分配一组线程，每组线程并行工作. 由此可以同时计算函数在每个体素的取值区间，从而极大地提高计算效率，缩短计算时间. 若函数在某个体素上的取值区间包含零值，则这个体素为特征体素，所有的特征体素构成了对隐式曲面的一个逼近表示，如图 1 所示.

a 特征体素集合　　b 光滑后的特征体素外表面

图 1 表示双曲面的特征体素集合及其外表面的抽取和光滑

算法 1. 基于 GPU 实现的区间分析方法绘制隐式曲面

输入. 函数 f，绘制空间 Ω，绘制分辨率 $T=N_x\times N_y\times N_z$.

输出. 逼近隐式曲面的特征体素集合.

Step1. 将绘制空间 Ω 按照分辨率 $N_x\times N_y\times N_z$ 细分成体素表示. 若曲面方程是多项式方程，执行 Step2；若曲面方程是超越方程，转 Step3.

Step2. 根据式(1)(2)并行计算出多项式在所有体素中的取值范围. 给定一个阈值 K，设多项式的系数矩阵 $\boldsymbol{A}$ 的阶数为 $n\times m\times l$，若 n,m,l 都不大于 K，则为每个体素的矩阵计算任务分配一个线程；否则为每个体素的矩阵计算任务分配一组线程，用于优化矩阵乘法. 如图 2 所示，假设矩阵 $\boldsymbol{A}$ 与矩阵 $\boldsymbol{B}$ 相乘，结果矩阵为 $\boldsymbol{C}$. 本文将计算任务划分成若干个 $K\times K$ 子矩阵的乘法，子矩阵 $\boldsymbol{C}_{\mathrm{SUB}}$ 由矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 中的橙色部分相乘得到，这一步由一个计算线程执行，每个子矩阵计算线程都是相互独立、并行执行的.

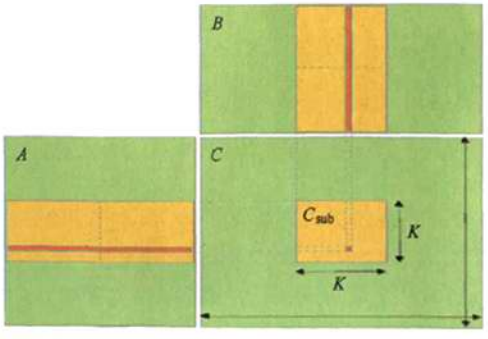


图 2 矩阵乘法分解

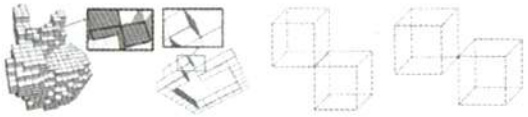
Step3. 首先估计输入函数 $f(x,y,z)$ 的 3 个一阶偏导数 $f_x(x,y,z)$，$f_y(x,y,z)$ 和 $f_z(x,y,z)$ 在指定区间上的取值区间 B_x，B_y 和 B_z；然后，根据式(3)计算函数在所有体素上的取值区间. 为每个体素的计算任务分配一个线程，所有线程并行执行.

Step4. 若函数 f 在某个体素中的取值区间包含零值，则此体素为特征体素，标记为 0；同样，将取值区间只包含正数或负数的体素分别标记为 1 和 −1. 这样可得到逼近隐式曲面 $f(x,y,z)=0$ 的特征体素集合及其内外体素集合.

2 隐式曲面的网格化

由算法 1 得到的特征体素集合构成了隐式曲面的一个逼近，同时也确定了隐式曲面的内部体素集合和外部体素集合. 为了得到隐式曲面的网格表示，本文抽取特征体素集合的外表面，也就是外体素集合和特征体素集合的交界面作为隐式曲面的网格表示.

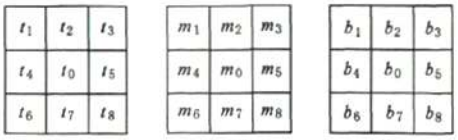
然而，特征体素的外表面并不一定是二维流形，为了确保隐式曲面网格化的拓扑正确性，在抽取网格时必须消除特征体素之间的非流形连接情况. 如图 3 所示，特征体素之间的非流形连接分为 2 种情况，如图 3 b 所示的边连接和如图 3 c 所示的点连接 2 种情况.



a 非流形连接实例　　b 边连接　　c 点连接

图 3 特征体素的非流形连接

下面分情况详细介绍如何判定特征体素之间的非流形连接. 一个任意特征体素 m_0 最多有 6 个面邻接的特征体素，有 12 个边邻接特征体素，有 8 个点邻接的特征体素；它们组成一个以 m_0 为中心，边长为 3 个单位长度的立方体. 对这 27 个特征体素分层编号，按照 Y 轴正方向分层，依次为 bot 层，mid 层和 top 层，每层的编号如图 4 所示.



a 顶层体素编号　　b 中间体素编号　　c 底层体素编号

图 4 特征体素 m_0 及其相邻体素的分层编号

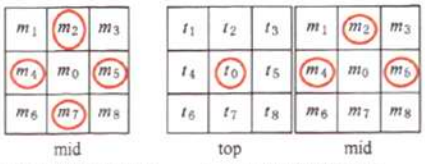
首先计算与特征体素 m_0 面邻接的特征体素的数量，根据这个数量分为 6 种情况，每种情况下非流

形连接都具有特定的空间分布特征. 所有的相邻特征体素的空间分布情况，在旋转变换下等价于下述的 6 种情况.

1) 若 m_0 有 6 个面邻接的特征体素，则 m_0 的 6 个面都不会成为曲面的外表面，因此不会造成非流形连接.

2) 如图 4 所示，若 m_0 有 5 个面邻接的特征体素，则 m_0 有一个面(记为面 C)是外表面或者内表面. 假设面 C 是外表面，并设这里缺少的面邻接特征体素是 t_0，若 t_3 位置存在特征体素，可能会造成与 m_0 的点连接，以及与 m_2，m_5 的特征体素边连接 2 种非流形连接. 本文将这种点连接和边连接同时存在的情形称为关联存在. 在关联存在下，消除了边连接就一定可消除点连接；反之不然. 因此本文提出判定非流形连接优先级的概念，认为边连接的优先级高于点连接，即当点连接和边连接关联存在时忽略点连接造成非流形的情况，考虑边连接造成非流形连接的情况. 本例中，暂时忽略 t_3 和 m_0 造成非流形连接的可能性. 因为当以 m_2 或 m_5 为中心体素进行非流形连接判断时会得到正确的判断. 同理，top 层和 bot 层的 1，6，8 位置的体素造成非流形连接的可能性也被暂时忽略，它们的连接情况会在后续操作中判断. 在这种情况下，算法暂时认为它们不会造成非流形连接.

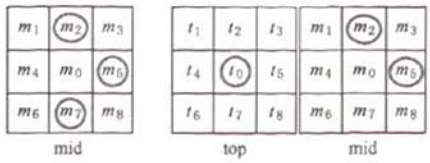
3) m_0 有 4 个面邻接的特征体素时有 2 种分布情况. 一种是有 2 对面相邻的特征体素，另一种是只有 1 对面相邻特征体素. 第一种分布情况如图 5 a 所示，与 m_0 面相邻的特征体素是 m_2，m_4，m_5，m_7(圆圈所示)，根据判定优先级的原则暂时忽略 top 层和 bot 层的 1，3，6，8 位置造成非流形连接的可能性. 显然，其他体素不会造成非流形连接. 另一种分布情况分布如图 5 b 所示，与 m_0 面相邻的是 m_2，m_4，m_5，t_0 位置的特征体素，而 b_7 位置的特征体素会造成非流形连接.



a 2对面相邻特征体素　　b 1对面相邻特征体素

图 5　特征体素 m_0 有 4 个面邻接特征体素时的 2 种连接情况

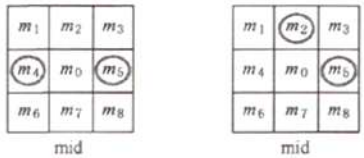
4) m_0 有 3 个面邻接的特征体素时有 2 种空间分布情况. 如图 6 a 所示，第一种情况只有 1 对与 m_0 面相邻的特征体素，设与 m_0 面相邻的特征体素为 m_2，m_5，m_7，特征体素 t_4，b_6 会造成非流形连接. 如图 6 b 所示，第二种情况没有成对的面邻接特征体素，设与 m_0 面相邻的特征体素是 m_2，m_5，t_0，则特征体素 b_7，b_4，b_6，m_6 会造成非流形连接.



a 1对与m_0面相邻特征体素　　b 无成对面相邻特征体素

图 6　特征体素 m_0 有 3 个面邻接特征体素时的 2 种连接情况

5) 当 m_0 有 2 个面相邻的特征体素，有 2 种空间分布情况. 第一种情况是存在 1 对与 m_0 面相邻的特征体素，如图 7 a 所示，与 m_0 面相邻的特征体素是 m_4，m_5，此时，特征体素 b_7，b_2，t_2，t_7 会造成非流形连接；另一种情况是面相邻的特征体素不成对，如图 7 b 所示，与 m_0 面相邻的特征体素是 m_2，m_5，特征体素 b_7，b_4，b_6，t_7，t_4，t_6，m_6 会造成非流形连接.

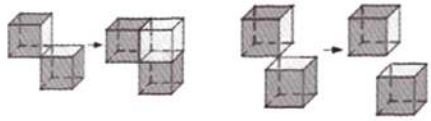


a 1对与m_0面相邻特征体素　　b 面相邻特征体素不成对

图 7　特征体素 m_0 有 2 个面邻接特征体素时的 2 种连接情况

6) 当 m_0 只有 1 个面相邻的特征体素时，假设是 m_5，这时有 12 个位置的特征体素会造成非流形连接，分别是 b_2，b_7，b_4，b_1，b_6，t_2，t_7，t_4，t_1，t_6，m_1 和 m_6.

在发现特征体素的非流形连接后，分别采取 2 种方法进行拓扑校正. 若存在特征体素边连接情况，需要在 2 个特征体素之间补充一个特征体素，使之成为面连接，如图 8 a 所示；若存在特征体素点连接情况，在拓扑结构上采取点分离措施，使点连接的 2 个特征体素在拓扑结构上分离开来，如图 8 b 所示.



a 补充特征体素　　b 拓扑分离

图 8　非流形连接的拓扑校正

经过上述处理后，就能确保所抽取特征体素集合的外表面是二维流形. 然后对其进行三角剖分，采用 Laplace 平滑算法对其进行平滑处理[27-28]后，就生成了逼近隐式曲面的三角网格曲面. 如图 1 所示，图 1 a 是特征体素集合，图 1 b 就是抽取特征体素集合外表面，并光滑后的三角网格曲面.

3 实验结果与讨论

本文算法已经在 VC 开发环境下实现，我们进行了大量的实验. 程序的运行环境为 Intel core2 Quad CPU Q9400，2.66 GHz CPU；4 GB 内存；NVIDIA GeForce GTX 260 显卡.

下面展示 9 个隐式曲面的近似三角网格曲面表示，其中前 6 个为代数曲面，后 3 个为超越方程定义的隐式曲面. 对所有隐式曲面的绘制分辨率为128×128×128.

为了说明 GPU 实现的区间分析方法绘制隐式曲面的效率，我们在表 1 中比较了区间分析方法绘制隐式曲面算法分别基于 GPU 和 CPU 实现的运行时间，以及光滑三角网格所需时间. 可以看出，CPU 绘制时间从几秒钟到几分钟不等，而 GPU 绘制时间都在零点几秒这个数量级. 时间差对比最大的第 2 个例子，GPU 用时 0.199 s，CPU 用时长达 108.093 s，时间缩短了 543 倍.

表 1 隐式曲面 CPU 和 GPU 绘制时间以及网格光滑时间比较

例子	面片数	绘制时间/s		光滑时间/s
		CPU	GPU	
1	81060	10.530	0.156	0.132
2	156040	108.093	0.199	0.276
3	86462	17.597	0.182	0.135
4	66784	7.316	0.168	0.128
5	212758	214.439	0.608	0.426
6	62464	8.003	0.208	0.126
7	196328	32.065	0.160	0.343
8	158720	63.758	0.134	0.314
9	124576	21.088	0.129	0.249

为了绘制出更精确的曲面模型，一种直接有效的方法是选取较高的绘制分辨率，但是同时会带来效率降低的问题. 实际上，在较高的分辨率下，绘制空间中的特征体素数量只占全部体素数量的较小比例，因此可以对隐函数在每个体素空间中的取值进行预估，排除明显不包含曲面的体素(非特征体素)，然后采取本文算法确定特征体素. 同时，针对不同的曲面方程动态调整计算资源的分配方式，比如高次幂的曲面方程在每个体素空间内的计算量较大(需要处理的矩阵规模相应增大)，可以适当降低全局的并行性，增大每个体素内计算任务的并行性，为每个体素的计算任务分配更多的线程. 而对低次幂的曲面方程则相反，增加全局的并行性，减少每个体素的计算线程，并行完成更多的体素计算任务.

为了比较网格光滑前后三角网格对隐式曲面的逼近程度，本文引进了 2 个误差值：一个是平均平方误差，定义为

$$e_s=\frac{\sum_{i=1}^{n}f^2(x_i,y_i,z_i)}{n};$$

另一个为最大值误差，定义为

$$e_m=\max_{i=1}^{n}\{|f(x_i,y_i,z_i)|\}.$$

其中 n 为三角网格的顶点数. 表 2 所示为光滑前后的误差数据，可以看出，网格光滑前后这 2 个误差值的变化不大.

表 2 网格光滑前后的平均平方误差和最大值误差的比较

例子	平均平方误差		最大值误差	
	光滑前	光滑后	光滑前	光滑后
1	0.163032	0.105540	0.530391	0.530391
2	0.106492	0.101183	0.332597	0.299397
3	1.092042	0.880559	4.312500	4.312500
4	0.525343	0.524275	1.381317	1.378546
5	0.418889	0.417485	0.771313	0.767028
6	0.100045	0.100182	0.245651	0.245651
7	0.853527	0.853242	1.644250	1.644028
8	0.784486	0.784366	1.396299	1.396299
9	2.562403	2.562396	2.649020	2.641328

下面给出隐式曲面的方程的例子.

例 1. $6x^2-0.6x-6xy-2.4y+12yz+0.18=0$，绘制空间为$[-1,1]\times[-1,1]\times[-1,1]$，是描述两张双曲面的 2 次多项式方程，如图 1 所示.

例 2. $x^2y^2+y^2z^2+x^2z^2+xyz=0$，绘制空间为$[-0.5,0.5]\times[-0.5,0.5]\times[-0.5,0.5]$，是描述 Steiner's Roman 曲面的 4 次多项式方程，如图 9 所示.

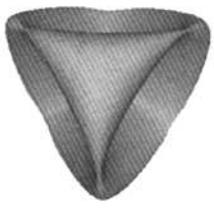


图 9 Steiner Roman 曲面

例 3. $60x^4-30x^2-120x^2y+24yz+36x-12xy=0$，绘制空间为$[-1,1]\times[-1,1]\times[-1,1]$，是一个 4 次多项式方程，如图 10 所示.

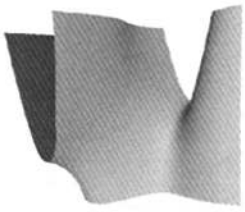


图 10　一个 4 次多项式方程定义的代数曲面

例 4. $z^3+xz+y=0$,绘制空间为$[-5,5]\times[-5,5]\times[-5,5]$,是描述立方尖点突变的 3 次多项式方程,如图 11 所示.

图 11　描述立方体尖点突变的代数曲面

例 5. $x^4+y^4+z^4+1-(x^2+y^2+z^2+x^2y^2+z^2y^2+x^2z^2)=0$,绘制空间为$[-2,2]\times[-2,2]\times[-2,2]$,是描述 Quadruple Kummer 曲面的 4 次多项式方程,如图 12 所示.

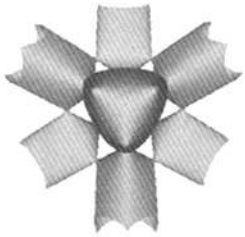


图 12　Quadruple Kummer 曲面

例 6. $\frac{55}{256}-x+2x^2-2x^3+x^4-\frac{55}{64}y+2xy-2x^2y+\frac{119}{64}y^2-2xy^2+2x^2y^2-2y^3+y^4=0$,绘制空间为$[-2,2]\times[-2,2]\times[-2,2]$,描述一对切圆柱的 4 次多项式方程,如图 13 所示.

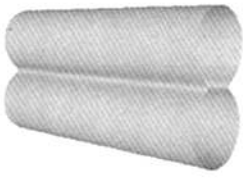


图 13　一对切圆柱

例 7. $\sin x\sin y\sin z+\sin x\cos y\cos z+\cos x\sin y\cos z=0$,绘制空间为$[-\pi,\pi]\times[-\pi,\pi]\times[-\pi,\pi]$,是描述 Diamond 曲面的超越方程,如图 14 所示.

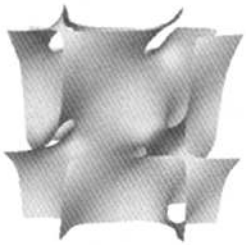


图 14　Diamond 曲面

例 8. $\cos x\sin y+\cos y\sin z+\cos z\sin x=0$,绘制空间为$[-\pi,\pi]\times[-\pi,\pi]\times[-\pi,\pi]$,是描述 Gyroid 曲面的超越方程,如图 15 所示.

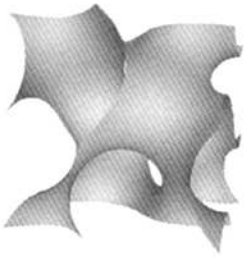


图 15　Gyroid 曲面

例 9. $\cos x+\cos y+\cos z=0$,绘制空间为$[-\pi,\pi]\times[-\pi,\pi]\times[-\pi,\pi]$,是描述 Schwarz P 曲面的超越方程,如图 16 所示.

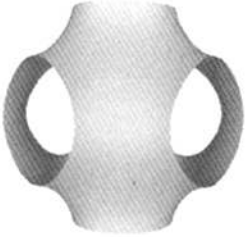


图 16　Schwarz P 曲面

4　结　　论

本文提出了一种基于 GPU 实现的利用区间分析方法绘制隐式曲面并对其网格化的算法.根据给定的绘制分辨率,首先将绘制空间离散成体素表示,然后利用 GPU 并行计算能力同时计算隐函数在各个体素的取值范围,从而确定出包含隐函数零等值面的特征体素.GPU 的并行计算能力极大地提高了绘制效率,使绘制时间缩短了几十倍到几百倍.进一步,

本文设计的网格抽取算法在抽取特征体素集合外表面的同时,保证得到的网格曲面是二维流形,对这个网格曲面光滑处理后就生成了隐式曲面的近似网格表示.大量实验表明,基于本文算法的隐式曲面绘制和网格光滑的运行时间都在零点几秒数量级,达到了实时化.这为快速绘制隐式曲面和生成它的网格表示提供了可能,有利于促进隐式曲面造型技术早日进入实用领域.

参考文献(References):

[1] Kalra D, Barr A H. Guaranteed ray intersections with implicit surfaces [J]. Computer Graphics, 1989, 23(3): 297-306

[2] Hart J C. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces [J]. The Visual Computer, 1996, 12(9): 527-545

[3] Gamito M N, Maddock S C. Progressive refinement rendering of implicit surfaces [J]. Computers & Graphics, 2007, 31(5): 698-709

[4] Mitchell D P. Robust ray intersection with interval arithmetic [C] //Proceedings of Graphics Interface. Toronto: Canadian Information Processing Society Press, 1990: 68-74

[5] Moore R E, Yang C T. Interval analysis [M]. New Jersey: Prentice-Hall, 1966

[6] Gamito M N, Maddock S C. Ray casting implicit fractal surfaces with reduced affine arithmetic [J]. The Visual Computer, 2007, 23(3): 155-165

[7] Comba J L D, Stolfi J. Affine arithmetic and its applications to computer graphics [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1993: 9-18

[8] Martin R, Shou H, Voiculescu I, *et al*. Comparison of interval methods for plotting algebraic curves [J]. Computer Aided Geometric Design, 2002, 19(7): 553-587

[9] Shou H H, Martin R, Voiculescu I, *et al*. Affine arithmetic in matrix form for polynomial evaluation and algebraic curve drawing [J]. Progress in Natural Science, 2002, 12(1): 77-80

[10] Shou H H, Lin H W, Martin R, *et al*. Modified affine arithmetic is more accurate than centered interval arithmetic or affine arithmetic [M]. //Lecture Notes in Computer Science. Heidelberg: Springer, 2003, 2768: 355-365

[11] Shou H H, Lin H W, Martin R, *et al*. Modified affine arithmetic in tensor form for trivariate polynomial evaluation and algebraic surface plotting [J]. Journal of Computational and Applied Mathematics, 2006, 195(1): 155-171

[12] Hilton A, Stoddart A J, Lllingworth J, *et al*. Marching triangles: range image fusion for complex object modeling [C] //Proceedings of International Conference on Image Processing. Piscataway: IEEE Press, 1996, 2: 381-384

[13] Akkouche S, Galin E. Adaptive implicit surface polygonization using marching triangles [J]. Computer Graphics Forum, 2001, 20(2): 67-80

[14] Schreiner J, Scheidegger C E, Silva C T. High-quality extraction of isosurfaces from regular and irregular grids [J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(5): 1205-1212

[15] Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1987: 163-169

[16] Kobbelt L P, Botsch M, Schwanecke U, *et al*. Feature sensitive surface extraction from volume data [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 2001: 57-66

[17] Chernyaev E V. Marching cubes 33: construction of topologically correct isosurfaces [R]. Genevese: Technical Report CN 95-17, CERN, 1995

[18] Lewiner T, Lopes H, Vieira A W, *et al*. Efficient implementation of Marching Cubes cases with topological guarantees [J]. Journal of Graphics Tools, 2003, 8(2): 1-15

[19] Garland M, Heckbert P S. Surface simplification using quadric error metrics [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1997: 209-216

[20] Surazhsky V, Alliez P, Gotsman C. Isotropic remeshing of surfaces: a local parameterization approach [C] //Proceedings of the 12th International Meshing Roundtable. Heidelberg: Springer, 2003: 215-224

[21] Alliez P, de Verdiere E C, Devillers O, *et al*. Isotropic surface remeshing [C] //Proceedings of the Shape Modeling International. Los Alamitos: IEEE Computer Society Press, 2003: 49-58

[22] Stander B T, Hart J C. Guaranteeing the topology of an implicit surface polygonization for interactive modeling [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1997: 279-286

[23] Boissonnat J D, Cohen-Steiner D, Vegter G. Isotopic implicit surface meshing [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 2004: 301-309

[24] Boissonnat J D, Oudot S. Provably good sampling and meshing of surfaces [J]. Graphical Models, 2005, 67(5): 405-451

[25] Cheng S W, Dey T K, Ramos E A, *et al*. Sampling and meshing a surface with guaranteed topology and geometry [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 2004: 280-289

[26] Berz M, Hoff stätter G. Computation and application of Taylor polynomials with interval remainder bounds [J]. Reliable Computing, 1998, 4(1): 83-97

[27] Desbrun M, Meyer M, Schröder P, *et al*. Implicit fairing of irregular meshes using diffusion and curvature flow [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1999: 317-324

[28] Taubin G. A signal processing approach to fair surface design [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1995: 351-358

基于GPU和区间分析的隐式曲面绘制和网格化

作者： 秦阳， 蔺宏伟， 冼楚华， 高曙明， Qin Yang， Lin Hongwei， Xian Chuhua， Gao Shuming

作者单位： 浙江大学CAD&CG国家重点实验室,杭州,310058

刊名： 计算机辅助设计与图形学学报 ISTIC EI PKU

英文刊名： JOURNAL OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS

年，卷(期)： 2011,23(5)