

文章编号:1001-9081(2010)12-3281-03

基于三角形不规则网模型的快速体素化方法

陈学工,邱华,付金华,马金金

(中南大学信息科学与工程学院,长沙410083)

(csucxg@csu.edu.cn)

摘要:为了改善在大数据量时体素化效率不高的缺点,针对三角形不规则网(TIN)模型的三角网特性,提出了一种快速简单的体素化算法。首先通过细划三角面片,将面体素化转换为简单的点体素化生成体表面模型,然后利用深度缓存原理快速寻找初始种子体素进行体内填充。实验结果表明,对于精细复杂的大规模TIN模型,算法能确实有效地生成逼近原模型的26-连通的体素模型,且具有高效的时间效率。

关键词:表面体素化;实体体素化;三角形不规则网模型;种子填充;三角形细划

中图分类号:TP391.41 **文献标志码:**A

Fast voxelization based on triangulated irregular network model

CHEN Xue-gong, QIU Hua, FU Jin-hua, MA Jin-jin

(School of Information Science and Engineering, Central South University, Changsha Hunan 410083, China)

Abstract: In order to improve the voxelization efficiency of large data, a fast and simple voxelization algorithm was proposed for the Triangulated Irregular Network (TIN) model. The surface voxelization was realized by dividing triangular facets, which could convert surface voxelization to points voxelization. Then, the solid voxelization was realized by using the theory of depth buffer to quickly search initial seed. The experimental results show that for the complex and precise TIN model, a voxel model of 26-connected which is approached to original model can be efficiently generated, and it has a high time efficiency.

Key words: surface voxelization; solid voxelization; Triangulated Irregular Network (TIN) model; seed filling; triangular division

0 引言

三维空间数据模型的研究大体上可以分为三个方向^[1]:面模型、体模型和混合模型。面模型具有数据存储量小,建模速度快的优点。目前基于面模型的建模方法有不规则三角网建模(Triangulated Irregular Network, TIN)、边界表示法和断面建模等。体模型相对于面模型有其特有的优势,如对物体表面模型复杂度的不敏感性、适合于描述采样数据、内部信息及便于块操作。随着硬件的不断发展,体对象的体模型表示已被广泛应用于各个领域。因此,如何快速地将体对象的面模型表示转换为体模型表示是一个值得研究的问题。

从面模型到体模型的转换过程称之为体素化,其任务是把物体表面的几何表示形式离散成最接近该物体的体素表示形式,从而产生体数据集。体素可以理解为二维像素在三维空间的推广,它们是一组分布在正交网格中心的立方体单元,类似于将二维图像叠加起来形成的三维光栅^[2]。目前,体素化的方法有很多,如用三维带符号的欧氏距离变换方法生成体素模型^[3];利用尺度空间理论获得点、线、三角面的反走样体素模型^[4];利用欧氏距离测度理论得到体素模型^[5];利用GPU特性的硬件体素化方法^[6]。这些方法各有其优缺点,适应的场合也各不相同。文献[7]提出了一种基于八叉树的体素化方法,具有较高的空间效率,但其在体素化过程中存在大量的射线求交运算,影响了该算法的时间效率,使其不能适应于精细复杂的体模型。为了解决该问题,本文根据TIN模型

的特有性质,提出了一种简单高效的体素化算法,该算法提高了体素化的执行效率,降低了时间复杂度。

1 基本概念与模型简述

1.1 TIN 模型

TIN也称不规则三角形网或不规则三角形模型,是一种基于三角形的空间镶嵌模型^[8]。在工程计算中,TIN是一种强有力的模型,广泛应用于3D物体的表面建模。

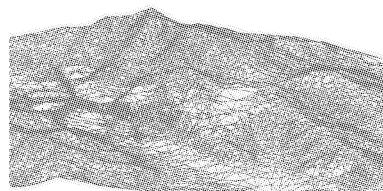


图1 TIN 地表模型

1.2 三维离散空间

设 \mathbf{Z}^3 是三维欧几里得空间 \mathbf{R}^3 的一个子集,包含 \mathbf{R}^3 中所有坐标为整数的点,这个子集被称为网格^[9]。体素则是以网格为中心点的单位立方体,一个网格点关联一个体素。每个体素都可以映射到集合 $(0,1)$,目标体素赋值为“1”,非目标体素赋值为“0”。

每个体素都有26个相邻的体素,且体素间具有三种邻接关系:即6-邻接、18-邻接、26-邻接^[9]。如果 $W \subseteq \mathbf{Z}^3$ 空间中的连续体素均为相同属性,且其中任意相邻的体素对都是N-邻接的,这些

收稿日期:2010-05-09;修回日期:2010-07-28。基金项目:国家863计划项目(2006AA06Z114)。

作者简介:陈学工(1965-),男,湖南新化人,副教授,博士,主要研究方向:地理信息系统; 邱华(1986-),女,贵州贵阳人,硕士研究生,主要研究方向:地理信息系统、计算机图形学; 付金华(1984-),男,河南信阳人,硕士研究生,主要研究方向:地理信息系统、计算机图形学; 马金金(1985-),男,江苏南通人,硕士研究生,主要研究方向:地理信息系统、计算机图形学。

体素则组成了 N -路径。如果体素集合中任意对体素都存在一条 N -路径，则称这个体素集合为 N -连通。前缀“ N ”定义了邻接关系， $N \in \{6, 18, 26\}$ （二维情况下 $N \in \{4, 8\}$ ）。

2 表面体素化

2.1 点的体素化

在三维欧几里得空间中，任意一点都存在唯一的一个体素与之对应。因此，通过坐标变换找到点的对应体素即可完成点的体素化。式(1)^[7]为点 $p(x, y, z)$ 的体素化公式，其中 (vox_i, vox_j, vox_k) 为体素坐标， w 为体素宽度。

$$\begin{cases} vox_i = \lfloor x/w \rfloor \\ vox_j = \lfloor y/w \rfloor \\ vox_k = \lfloor z/w \rfloor \end{cases} \quad (1)$$

2.2 线段体素化

对于线段 L 的体素化，根据式(1)可得到线段两个端点的体素 Vox_1 和 Vox_2 。如果线段 L 足够短，其长度 l 小于一个阈值 ε ，使得 Vox_1 和 Vox_2 正好是26-邻接，那么 Vox_1 和 Vox_2 即是线段 L 的体素化结果。如图2所示，在二维情况下，对于长度 $l < \varepsilon$ 的线段，两端点体素的位置有三种：重合、边邻接和点邻接。在三维情况下，两端点体素位置有四种：重合、面邻接、边邻接和点邻接。

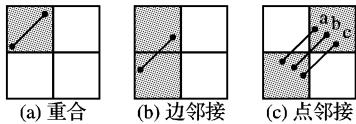


图2 小于体素长度的线段体素化示意图

若要两个端点体素不是26-邻接，则两个体素间至少还应再间隔一个体素。如果端点体素与一个间隔体素之间以6-连通的方式邻接，则可以保证两个端点体素之间距离最小，其值为体素宽度 w 。因此，只要两个端点体素之间的距离小于 w ，就能保证两端点体素为26-邻接，则阈值 ε 的取值应为 w 。

因此，对于任意长度的线段，首先将其划分为若干个长度相等的子线段，且这些子线段的长度 $l' < \varepsilon$ ，即 $l' < w$ ，然后将各子线段端点体素化，即可实现任意长度线段的体素化。通过上述方法，保证了长线段的端点体素化结果正好是26-连通的，如图3所示。

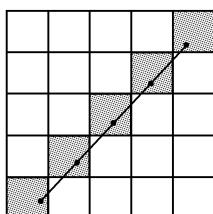


图3 任意长度线段的体素化示意图

2.3 三角面片的体素化

由2.2节提到的线段体素化方法可知，在进行三角面片体素化时，如果三角面片足够小，其三个顶点所在的体素为26-连通的，那么这三个顶点所在体素即为该三角面片的体素化结果。

图4为一个足够小的三角面片的体素化示意图。由图可知，三角面片顶点体素化结果只可能有三种情况：1) 三个顶点都占据一个相同体素；2) 两个顶点占据一个相同体素；3) 三个顶点各占据一个体素。1)和2)可看做是3)的特例，只要保证在3)情况下的三个体素为26-连通的，就可以得到一个26-连通的三角面片体素化结果。由线段体素化原理可知，

若两点之间线段长度小于体素宽度，则可保证两点所在体素之间的邻接性。因此对于三角面片，只要其三边均小于体素宽度，即可保证三个顶点所在体素之间两两邻接，从而保证了三个顶点所在体素的连通性。

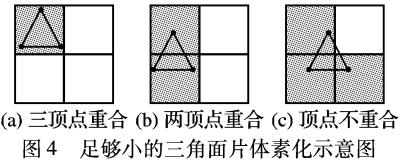


图4 足够小的三角面片体素化示意图

根据以上思想，对一个任意大小的三角面片体素化，可将其细划为若干个子三角面片，并使这些子三角面片三条边的长度均小于体素宽度，从而将三角面片的面体素化转化为了点体素化，简化了三角面片的体素化方法。

对于三角形细划的方法，本文提出了一种简单的非递归算法，其主要思想为：首先将三角面片的三边 n 等分；然后通过各等分点作其余两边的平行线，各平行线的相交结果即为三角面片的划分结果。

以图5为例对 $\triangle abc$ 进行划分，首先计算 $\triangle abc$ 的三条边的长度，得到最长的边为 ab ，用 ab 边的长度 l_{ab} 除以体素宽度 w 并向上取整得到划分系数6，将 $\triangle abc$ 的 ab, ac 边分别划分为6等分得到等分点 b_i 及 c_i ($i = 0, 1, \dots, 6$)，依次连接对应的 b_i 及 c_i 得到线段 $b_i c_i$ ，并将 $b_i c_i$ 划分为 i 等分，各等分点即为子三角形的顶点，也即体素化对象。然后只需将各顶点体素化即可得到三角面片的体素化结果。

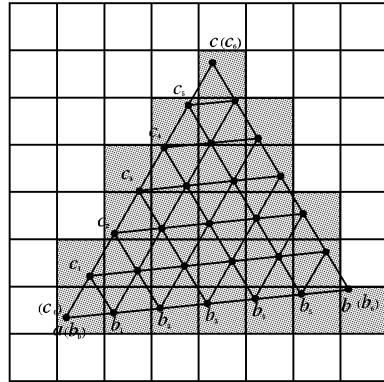


图5 三角面片体素化示意图

在三角形细划算法中： w 表示体素宽度； a, b, c 表示三角面片的顶点； l_{max} 存储三边中最长边的长度； seg 存储三角面片的划分系数；数组 $vexArrayAB$ 和数组 $vexArrayAC$ 分别存储 ab 边和 ac 边的分割顶点，数组 $vexArray$ 存储最终得到的划分点，数组下标均从零开始； i 为计数器，初值为0。

算法1

- 1) 输入 w 及三角面片的三条边，计算三边长度，并将其最大值存入 l_{max} ；
- 2) 用 l_{max} 除以 w ，并向上取整，所得结果存入 seg ；
- 3) 将 ab 边和 ac 边等分为 seg 份分别得到 $seg + 1$ 个点，并从 a 点开始依次将 ab 边上各点存入数组 $vexArrayAB$ ， ac 边上各点存入数组 $vexArrayAC$ ；
- 4) 判断 $i > seg$ 是否成立，成立转入7)；否则转入5)；
- 5) 将点 $vexArrayAB[i]$ 和点 $vexArrayAC[i]$ 连接得到一线段，并将此线段等分为 i 份，得到 $i + 1$ 个点，将这些点依次存入数组 $vexArray$ ；
- 6) 将 i 加1后转入4)；
- 7) 结束。

3 实体的体素化

在第 2 章中实现了体的表面体素化,但就实体而言还需要对其内部进行填充,通常采用种子填充法完成内部体素化。目前,对于种子的选取有两种方法。一种是选取非目标体素作为种子填充非目标体素区域^[10]。该方法可快速得到种子进行填充,但对于内部有空洞的物体则会将空洞误判为实体,造成错误。另一种做法与第一种相反^[11],选取目标体素作为种子体素填充目标体素区域。该方法可避免第一种方法所造成的错误,但种子的选取却极为繁琐,往往需要大量的求交运算,影响了算法的效率。

根据深度缓存的原理^[12],对于体素 $vox(i, j, k)$,其相对应的六个表面体素可表示为: $vox_{-x}(i_{-x}, j, k)$ 、 $vox_{+x}(i_{+x}, j, k)$ 、 $vox_{-y}(i, j_{-y}, k)$ 、 $vox_{+y}(i, j_{+y}, k)$ 、 $vox_{-z}(i, j, k_{-z})$ 、 $vox_{+z}(i, j, k_{+z})$ 。若体素 vox 与六个表面体素同时满足 $i_{-x} < i < i_{+x}$ 、 $j_{-y} < j < j_{+y}$ 和 $k_{-z} < k < k_{+z}$,则可判定该体素位于实体内部,为目标体素。由此,本文提出了一种快速选取种子体素的算法。

算法 2

- 1) 在体中间位置任意选取一体素 $vox(i, j, k)$;
- 2) 沿 i 方向扫描体素,首先由负方向向正方向扫描体素,记录得到的第一个表面体素坐标 $vox_{-x}(i_{-x}, j_x, k_x)$,接着由正方向向负方向扫描体素,记录得到的第一个表面体素坐标 $vox_{+x}(i_{+x}, j_x, k_x)$;
- 3) 以 $vox_{-x}(i_{-x}, j_x, k_x)$ 为起点,沿 i_{-x} 向 i_{+x} 方向扫描得到第一个非目标体素记为 $vox(i, j, k)$;
- 4) 判断 $i_{-x} < i < i_{+x}$ 是否成立,成立则转 5),否则转入 3);
- 5) 沿 j 方向扫描体素,首先由负方向向正方向扫描体素,记录得到的第一个表面体素坐标 $vox_{-y}(i_y, j_{-y}, k_y)$,接着由正方向向负方向扫描体素,记录得到的第一个表面体素坐标 $vox_{+y}(i_y, j_{+y}, k_y)$;
- 6) 判断 $j_{-y} < j < j_{+y}$ 是否成立,成立则转入 8);否则转入 7);

表 1 模型表面体素化时间

模型	三角面片数	分辨率及表面体素化时间										
		128 × 128 × 128				256 × 256 × 256				512 × 512 × 512		
		三角面片平 均划分系数	表面体素 化时间/ms	表面体素个 数/(×10 ³)		三角面片平 均划分系数	表面体素 化时间/ms	表面体素个 数/(×10 ³)		三角面片平 均划分系数	表面体素 化时间/ms	表面体素个 数/(×10 ³)
Robot ₁	564	41	165	565.6		68	637	1516.3		136	2004	5908.6
Robot ₂	712	10	206	54.1		19	261	186.1		39	472	687.7
Surface ₁	5511	4	79	100.4		7	156	307.3		14	625	1056.8
Surface ₂	24141	2	187	147.8		3	284	360.6		7	859	1055.0

表 2 寻找初始种子的遍历次数

模型	i, j, k 三方向遍历次数		
	i	j	k
Robot ₁	1	1	1
Robot ₂	2	2	1
Machine	2	2	2
Cow	1	1	1

从表 1 中可以看出, Robot₂ 的三角面片数比 Robot₁ 多,但由于其平均三角面片较小,划分系数较低,所耗时间反而减少。对于 Surface₁ 和 Surface₂,虽然都含有大量的三角面片,但由于其划分系数较小,所以也具有较高的执行效率。因此可以看出,对于精细复杂的 TIN 表面模型,本文算法能很好地提高体素化的效率。表 2 给出了利用算法 2 寻找初始种子时

7) 以 $vox_{-y}(i_y, j_{-y}, k_y)$ 为起点,沿 j_{-y} 向 j_{+y} 方向扫描得到第一个非目标体素存为 $vox(i, j, k)$,转入 2);

8) 沿 k 方向扫描体素,首先由负方向向正方向扫描体素,记录得到的第一个表面体素坐标 $vox_{-z}(i_z, j_z, k_{-z})$,接着由正方向向负方向扫描体素,记录得到的第一个表面体素坐标 $vox_{+z}(i_z, j_z, k_{+z})$;

9) 判断 $k_{-z} < k < k_{+z}$ 是否成立,成立转入 11),否则转入 10);

10) 以 $vox_{-z}(i_z, j_z, k_{-z})$ 为起点,沿 k_{-z} 向 k_{+z} 方向扫描得到第一个非目标体素存为 $vox(i, j, k)$,转入 2);

11) 找到种子 $vox(i, j, k)$,算法结束。

4 算法分析及实验结果

在表面体素化算法中,假设三角面片数为 n ,三角面片的平均划分系数为 k ,则三角面片的平均划分次数为 $k!$,因此算法 1 的时间复杂度为 $O(k^2 \times n)$ 。表面体素化过程中,当系数 k 较低时,由于其三角面片平均划分次数较小,即 $k^2 \ll n$,算法 1 的时间复杂度可以近似为线性时间复杂度 $O(n)$ 。随着 k 的提升,三角面片平均划分次数变多,算法 1 的时间复杂度退化为 $O(k^2)$ 。对于算法 2,因表面体素具有 26- 连通特性,所以在相邻的局部区域内,表面体素不会存在极端复杂的情况。算法 2 能在常数时间复杂度内寻找到种子体素。体内填充的时间算法复杂度可参见文献[8]。

本算法在实现过程中,硬件开发环境为:CPU 主频为 2.0 GHz、内存为 384 MB 的 PC。软件开发环境为:MS Windows XP 操作系统、Visual C++ 6.0、显示处理库 OpenCL。图 6 是分辨率分别为 $128 \times 128 \times 128$ 、 $256 \times 256 \times 256$ 和 $512 \times 512 \times 512$ 时的体素化效果图。表 1 给出了本文表面体素化算法的实验模型参数及计算时间,其中 Robot₁ 和 Robot₂ 是两个形状相似的机器人模型, Surface₁ 和 Surface₂ 则是两个形状相似的地表模型。

在 i, j, k 方向的遍历次数,从表中可以看出在通常情况下本算法能快速寻找到种子。表 3 给出了相同环境下文献[7]中四种模型的表面体素化时间。

表 3 文献[7]模型表面体素化时间

模型	三角面片数	分辨率及计算时间					
		128 × 128 × 128		256 × 256 × 256		512 × 512 × 512	
		表面体素 化时间/s	表面体素个 数/(×10 ³)	表面体素 化时间/s	表面体素个 数/(×10 ³)	表面体素 化时间/s	表面体素个 数/(×10 ³)
Gear	1196	3	43.2	12	182.0		
Dacubic	1648	5	134.2	20	559.4		
Daktoolcor	3802	11	84.2	45	349.4		
Gear_MM	4748	14	77.5	57	321.2		

(下转第 3287 页)

为了达到这一目的,本文实时捕获了仿真的视频流并进行播放显示,对视频质量进行了主观评价,并结合客观数据包传输率的定量分析,评估了无线 MAC 协议在不同跳数下网络性能。主观评价视频序列中同一帧图像在不同跳数下的截图性能,情况如图 6 所示。

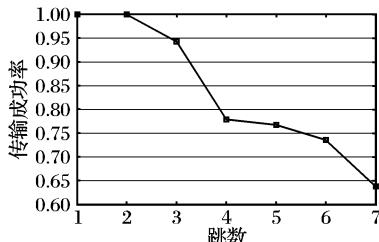


图 5 HIL-NS 实时流传输成功率

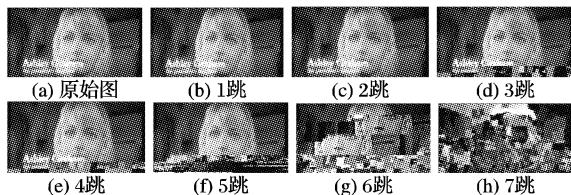


图 6 HIL-NS 实时流性能帧对比

容易看出,在前两跳下视频能正常接收显示。从 3 跳开始,视频质量逐渐下降。到了 7 跳以后,已经难以恢复原来视频样貌。由图 5 的数据包传输率随跳数变化的关系可以发现:802.11b 协议下,前两跳都能无丢包地传输,所以视频在前两跳下能正常显示;而从 3 跳开始,数据传输率逐渐下降,到 6 跳时,数据传输率已经不足 75%。结合主观和客观的评价可以发现,当数据包传输率保持在 75% 以上时,视频质量下降缓慢;一旦不足 75%,视频质量急剧下降,效果如图 6 (g)、(h)所示。

(上接第 3283 页)

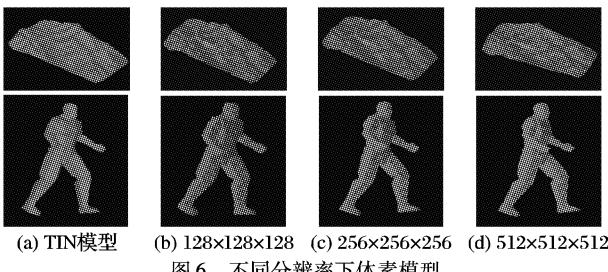


图 6 不同分辨率下体素模型

5 结语

本文提出了一种简单高效的体素化算法,整个算法避免了求交运算,提高了体素化的效率,且该算法能够快速处理复杂精细的表面模型,特别是大规模 TIN 模型。本算法的不足之处在于,在处理三角面片大的 TIN 模型时,在相同分辨率的情况下由于三角面片划分系数变大,导致算法时间效率下降,所以笔者将进一步对该类问题进行后续研究。

参考文献:

- [1] 管伟光,解林,马颂德. 体图形学[J]. 中国图象图形学报,1999,4(6): 486-490.
- [2] KAUFMAN A, COHEN D, YAGEL R. Volume graphics [J]. IEEE Computer, 1993, 26(7): 51-64.
- [3] 蒋宏伟,王国瑾. 三维带符号的欧氏距离变换及其应用[J]. 计算机学报, 2003, 26(12): 1645-1651.
- [4] OOMES S, SNOEREN P, DIJKSTRA T. 3D shape representation:

4 结语

本文探讨了半实物仿真平台设计中的关键问题和解决方案,基于 NS-2 网络仿真器设计并实现了一种半实物仿真平台。该平台能够把实际业务流加入仿真网络进行仿真。通过车载无线网络下 802.11b MAC 协议的仿真实例分析,验证了所提平台的可行性。相比传统仿真软件基于文件形式的结果分析,该平台支持的可视化结果更具直观性。此外,从实验结果可以看出,实时视频流的正确传输和播放对数据成功接收率要求很高。为满足这种业务需求,这也对研究者在研究和设计协议时提出了更高的要求。

参考文献:

- [1] 张中伟. 半实物仿真技术在通信系统中的应用研究[C]//第四届中国软件工程大会. 杭州:[s. n.], 2007: 86-88.
- [2] 段威,任华,柳海峰. 基于 OPNET 的通信网络半实物仿真方法研究[J]. 计算机仿真, 2007, 24(11): 138-143.
- [3] 张筱,林孝康. 一种基于 OPNET 的网络半实物仿真模型[J]. 微计算机信息, 2007, 23(1): 257-258.
- [4] 赵玉亭,戴冠中,徐浩,等. 多跳 Ad Hoc 网络视频实时传输性能评估[J]. 计算机工程与应用, 2007, 43(24): 149-151.
- [5] 赖安琪. 基于 QualNet 的战术数据链半实物仿真[J]. 电讯技术, 2009, 49(5): 74-77.
- [6] 于斌,孙斌,温暖,等. NS2 与网络模拟[M]. 北京: 人民邮电出版社, 2007.
- [7] 吴堃,戴萌,陆桑璐,等. 基于 NS2 的 P2P 网络模拟平台研究[J]. 系统仿真学报, 2006, 18(10): 2152-2157.
- [8] 陈正宇. 基于 NS2 的无线网络视频 QoS 平台的实现[J]. 电子设计工程, 2009, 17(10): 72-76.
- [9] 管天,卢泽新,白建军. 基于半实物网络仿真的包截获关键技术研究[J]. 计算机工程与科学, 2006, 28(12): 99-102.
- [10] 刘文涛. 网络安全开发包详解[M]. 北京: 电子工业出版社, 2005.

Transforming polygons into voxels[C]// Proceedings of the First International Conference on Scale-Space Theory in Computer Vision. Berlin: Springer Verlag, 1997: 349-352.

- [5] 吴晓军,刘伟军,王天然,等. 改进的基于欧氏距离测度网格模型体素化算法[J]. 计算机辅助设计与图形学学报, 2004, 16(4): 1645-1651.
- [6] HSIEH H H, LAI Y Y, TAI W K, et al. A flexible 3D slicer for voxelization using graphics hardware[C]// Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. New Zealand: ACM Press, 2005: 285-288.
- [7] 吴晓军,刘伟军,王天然. 基于八叉树的三维网格模型体素化方法[J]. 工程图学学报, 2005, 26(4): 1-7.
- [8] 聂庆华. 地理信息系统及其在环境科学中的应用[M]. 北京: 高等教育出版社, 2006: 186.
- [9] HUANG J, YAGEL R, FILLIPOV V, et al. An accurate method to voxelize polygonal meshes [C]// Proceedings of the 1998 IEEE symposium on Volume visualization. New York: ACM, 1998: 119-126.
- [10] 柳稼航,方涛,杨建峰. 适用于任意复杂区域的全自动填充方法[J]. 计算机工程, 2008, 34(4): 238-240.
- [11] 薛斌党,薛文芳,姜志国. 三维种子填充算法的改进[J]. 计算机辅助设计与图形学学报, 2006, 18(10): 1553-1556.
- [12] 温佩芝,吴晓军,史泽林,等. 一种三维实体模型的离散表示方法[J]. 计算机工程与应用, 2003, 39(15): 14-15.