

Journal of Image  
and Graphics

# 中国图象图形学报



ISSN1006-8961  
CN11-3758/TB

2012 **10**  
Vol.17 No.

中国科学院遥感应用研究所  
中国图象图形学学会主办  
北京应用物理与计算数学研究所

# 中国图象图形学报

Zhongguo Tuxiang Tuxing Xuebao

2012 年 10 月 第 17 卷 第 10 期(总第 198 期)

## 目 次

### 综述

张量值图像插值方法综述 ..... 邵宇, 刘莹, 孙富春(1197)

### 图像处理和编码

利用模运算及其周期性特点的安全隐写算法 ..... 廖琪男(1206)

结合快速融合与颜色传递的双波段夜视图像染色 ..... 李郁峰, 冯晓云, 徐铭蔚, 黄文丽(1213)

Weber 定律下尺度空间的自适应构建 ..... 刘立, 张瑞军, 万亚平, 黄欣阳, 彭复员(1222)

改进权值函数的非局部均值去噪算法 ..... 单建华(1227)

### 图像分析和识别

非相似度保持投影 ..... 陈才扣, 侯钰(1232)

非正交二值子空间模板表示的并行生成 ..... 杨培, 武港山, 杨扬, 任桐炜(1237)

Adaboost 和随机图划分的无监督图像分类 ..... 李巍, 杨素锦, 段晓华(1245)

aiNet 背景抑制的单帧红外弱小目标检测 ..... 陈炳文, 王文伟, 秦前清(1252)

韦伯-中心环绕结构的图像显著性检测模型 ..... 林丽莉, 周文晖(1261)

气象传真图信息提取 ..... 李存东, 肖传毅, 潘海朗, 陈日清, 杨劲松(1268)

### 图像理解和计算机视觉

多要素空间场景相似性匹配模型及应用 ..... 宋腾义, 汪闽(1274)

### 计算机图形学

Delaunay 三角网通用合并算子及分治算法的简化 ..... 刘永和, 冯锦明, 郭维栋, 田根, 金毅(1283)

基于 Laplace 谱嵌入和 Mean Shift 的三角网格一致性分割 ..... 马亚奇, 李忠科, 赵静(1292)

断裂面匹配的破碎刚体复原 ..... 李群辉, 周明全, 耿国华(1298)

虚拟现实与增强现实

核爆炸外观景象实时模拟 ..... 郑涛, 徐晓刚, 邵承永(1305)

医学图像处理

全局脑白质纤维群智能跟踪算法 ..... 冯远静, 王哲进, 张贵军, 俞立(1312)

遥感图像处理

对偶四元数线阵遥感影像几何定位 ..... 盛庆红, 姬亭, 刘微微, 王惠南(1319)

修正安装矩阵提高 FY-3B/MERSI 的地理定位精度 ..... 吴荣华, 杨忠东, 关敏, 李翔翔(1327)

中国图象图形学报

刊名题字: 宋 健      月刊(1996 年创刊)      第 17 卷 第 10 期      2012 年 10 月 16 日出版

|         |                                                                                                                         |                    |                                                                                                                                                      |
|---------|-------------------------------------------------------------------------------------------------------------------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 主管单位    | 中国科学院                                                                                                                   | Superintended by   | Chinese Academy of Sciences                                                                                                                          |
| 主 办     | 中国科学院遥感应用研究所<br>中国图象图形学学会<br>北京应用物理与计算数学研究所                                                                             | Sponsored by       | Institute of Remote Sensing Application,<br>CAS China Society of Image and Graphics<br>Institute of Applied Physics and Computational<br>Mathematics |
| 主 编     | 李小文                                                                                                                     | Chief editor       | LI Xiaowen                                                                                                                                           |
| 编辑出版    | 《中国图象图形学报》编辑出版委员会<br>北京 9718 信箱 邮编 100101<br>电子信箱:jig@irsa. ac. cn<br>电话:010-64807995 010-82614429<br>网 址:www. cjig. cn | Editor , Publisher | Editorial and Publishing Board<br>of Journal of Image and Graphics<br>( P. O. Box 9718 ,Beijing 100101 ,China)<br>E-mail:jig@ irsa. ac. cn           |
| 印刷装订    | 北京北林印刷厂                                                                                                                 | Distributed by     | Beijing Bureau for Distribution of Newspapers<br>and Journals                                                                                        |
| 广告经营许可证 | 京朝工商广字第 0346 号                                                                                                          | Domestic           | All Local Post Offices in China                                                                                                                      |
| 总 发 行   | 北京报刊发行局                                                                                                                 | Foreign            | China International Book Trading Corporation<br>( P. O. Box 399 ,Beijing 100044 ,China)                                                              |
| 订 购     | 全国各地邮局                                                                                                                  | Printed by         | Beijing Beilin Printing House                                                                                                                        |
| 国外发行    | 中国国际图书贸易总公司<br>( 中国国际书店 )<br>( 北京 399 信箱 邮编 100044 )                                                                    |                    |                                                                                                                                                      |

ISSN 1006-8961    CN11-3758/TB    CODE ZTTFXZ    国内邮发代号: 82-831    国外发行代号: M1406    国内定价: 45.00 元

# Journal of Image and Graphics

(Monthly, Started in 1996)

Vol. 17 No. 10 October 2012

## Contents

### Review

Overview of tensor valued images interpolation technology ..... Shao Yu, Liu Ying, Sun Fuchun (1197)

### Image Processing and Coding

Secure steganography based on modulo and its cyclical characteristic ..... Liao Qinan (1206)

Night vision dual-band images coloration using fast fusion and color transfer  
..... Li Yufeng, Feng Xiaoyun, Xu Mingwei, Huang Wenli (1213)

Adaptive algorithm of scale-space construction method based on Weber's law  
..... Liu Li, Zhang Ruijun, Wan Yaping, Huang Xinyang, Peng Fuyuan (1222)

Non-local means denoising algorithm with enhanced weight function ..... Shan Jianhua (1227)

### Image Analysis and Recognition

Dissimilarity preserving projection ..... Chen Caikou, Hou Yu (1232)

Parallel algorithm for generating template representation based on non-orthogonal binary subspace  
..... Yang Pei, Wu Gangshan, Yang Yang, Ren Tongwei (1237)

Unsupervised image categorization based on Adaboost and stochastic graph partition ..... Li Wei, Yang Sujin, Duan Xiaohua (1245)

Infrared dim target detection in single image based on background suppression by aiNet  
..... Chen Bingwen, Wang Wenwei, Qin Qianqing (1252)

Image saliency detection model with Weber's law based on center-surround structure ..... Lin Lili, Zhou Wenhui (1261)

Information extraction from meteorological facsimile maps  
..... Li Cundong, Xiao Chuanyi, Pan Hailang, Chen Riqing, Yang Jinsong (1268)

### Image Understanding and Computer Vision

Multi-feature based spatial scene matching model and its application ..... Song Tengyi, Wang Min (1274)

### Computer Graphics

Merging planar Delaunay triangulations based on universal operators and the implementation of a divide-conquer algorithm  
..... Liu Yonghe, Feng Jinming, Guo Weidong, Tian Gen, Jin Yi (1283)

Consistence segmentation of triangle mesh using Laplace spectral embedding and Mean Shift  
..... Ma Yaqi, Li Zhongke, Zhao Jing (1292)

Reassembly of broken 3D solids based on fractured surfaces matching ..... Li Qunhui, Zhou Mingquan, Geng Guohua (1298)

### Virtual Reality and Augmented Reality

Real-time simulation of nuclear explosion scene ..... Zheng Tao, Xu Xiaogang, Shao Chengyong (1305)

### Medical Image Processing

Global white matter tractography using swarm optimization ..... Feng Yuanjing, Wang Zhejin, Zhang Guijun, Yu Li (1312)

### Remote Sensing Image Processing

Geo-positioning line-array CCD images with dual quaternion ..... Sheng Qinghong, Ji Ting, Liu Weiwei, Wang Huinan (1319)

Improved FY-3B/MERSI geolocation accuracy using installation matrix  
..... Wu Ronghua, Yang Zhongdong, Guan Min, Li Xiangxiang (1327)

中图法分类号: P283 文献标识码: A 文章编号: 1006-8961(2012)10-1283-09

论文引用格式: 刘永和, 冯锦明, 郭维栋, 田根, 金毅. Delaunay 三角网通用合并算子及分治算法的简化[J]. 中国图象图形学报, 2012, 17(10): 1283-1291.

## Delaunay 三角网通用合并算子及分治算法的简化

刘永和<sup>1</sup>, 冯锦明<sup>2</sup>, 郭维栋<sup>3</sup>, 田根<sup>1</sup>, 金毅<sup>1</sup>

1. 河南理工大学资源环境学院, 焦作 454000;

2. 中国科学院东亚区域气候-环境重点实验室, 全球变化东亚区域研究中心, 中国科学院大气物理研究所, 北京 100029;

3. 南京大学气候与全球变化研究院/大气科学学院, 南京 210093

**摘 要:** Delaunay 三角网在未来地学数值模拟中将发挥重要作用。分治算法是一种著名的经典构网算法, 但其子网合并过程十分复杂, 限制了其应用。提出使用通用算子的概念, 并用从以往算法中独立出来的算子和 3 个新算子来简化分治算法的子网合并。扩展三角形算子用于构造每个新三角形并维护三角网的拓扑关系和边界链表。凹边界填充算子对边界链表用递归来自动完成凹边界的智能三角形填充。子网合并算子先用一个新三角形连接两个子三角网, 再合并边界链表, 调用凹边界填充算子填充子网间的缝隙区域。所有算子都基于有向边的数据结构和用链表管理的三角网外边界, 借助链表操作, 使算法的构建简洁而又高效。除分治法外, 这些算子还被成功用于构建其他算法。由随机点集以及 LiDAR 点云的测试表明, 所有算法的构网均准确无误且分治算法的执行效率较高。

**关键词:** Delaunay 三角网; 通用算子; 子网合并; 凹边界填充; 分治法

## Merging planar Delaunay triangulations based on universal operators and the implementation of a divide-conquer algorithm

Liu Yonghe<sup>1</sup>, Feng Jinming<sup>2</sup>, Guo Weidong<sup>3</sup>, Tian Gen<sup>1</sup>, Jin Yi<sup>1</sup>

1. Institute of Resources and Environment, Henan Polytechnic University, Jiaozuo 454000, China;

2. Key laboratory of Regional Climate-Environment Research for Temperate East Asia, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China;

3. ICGCR, School of Atmospheric Sciences, Nanjing University, Nanjing 210093, China

**Abstract:** Delaunay triangulation will play an important role in numerical simulation of the geophysical process in the future. The divide-conquer algorithm is one of the well-known traditional fast Delaunay algorithms, but its complicated merging procedure has limited the applicability of this algorithm. In this study, we propose the concept of universal operators, which are used to simplify the construction of the Delaunay triangulation algorithm. Several operators are extracted from some previous Delaunay algorithms, and three new operators are used in the merging process of the divide-conquer algorithm. The operator of expanding a triangle is developed for constructing each new triangle and for managing the topology information and the linked list which represents the border edges of the triangulation. The operator of filling basins is developed for automatically filling the basins by creating new triangles using recursions. The triangulation-merging operator is developed for linking two sub-triangulations using one new triangle, transforming the two original linked list of border edges

收稿日期: 2011-09-26; 修回日期: 2012-04-11

**基金项目:** 国家自然科学基金项目(41105074, 40975048, 40975049); 中国科学院数字地球重点实验室开放研究基金项目(2011LDE010); 河南理工大学博士基金项目(B2011-038); 淮河流域气象开放研究基金项目(HRM200904)

**第一作者简介:** 刘永和(1976—), 男, 副教授, 2010 年于中国科学院大气物理研究所获气象学专业理学博士学位, 主要从事气候环境资料分析、分布式水文模型以及地球信息科学研究。E-mail: sucksis@163.com

into one new linked lists with a new sequential order, and filling the two basins by calling the operator of filling basins. All these operators are based on the same set of data structures and the linked list represented triangulation hull. Using operations on the linked list, all search operations in the triangulation hull are avoided, which make the final algorithm very concise and fast. The operators are successfully used for construct other Delaunay algorithms besides the divide-conquer algorithm. Large point datasets generated stochastically as well as LiDAR point clouds are used for testing the operator-based algorithms, and the result shows that the algorithms can correctly generate triangulations. Meanwhile, it is shown that the divide-conquer algorithm based on the operators are almost as fast as the horizontal expanding algorithm.

**Key words:** Delaunay triangulation; universal operators; merging of triangulations; basin-filling; divide-conquer algorithm

## 0 引言

平面 Delaunay 三角网在计算机辅助制造 (CAD)、物体表面 3 维重建、力学有限元分析和地形地貌 3 维可视化等领域具有广泛的用途,并在未来的地貌演化、水文及生态过程等数值模拟领域将有广泛的应用前景。

构建 Delaunay 三角网需要一套复杂的算法。一些商业 GIS 软件集成了这些算法和部分空间分析功能。然而这些工具仅能满足一些可视化、表面积计算和路径决策等初级应用,而对于复杂的数值模拟等研究,使用起来有诸多不便。一些软件开发者或科研人员有必要掌握一些 Delaunay 三角网算法并积累一些代码。直至目前已有多种 Delaunay 三角网算法,这些方法中有一些共同的处理过程,可以将其独立为通用算子。使用通用算子可大大简化构网算法的编写、扩展和改进。

经典的 Delaunay 三角网生成方法诞生于 20 世纪 70—90 年代,通用算子源于这些方法。所有其他三角网算法都可以看作这些经典方法的变体。按照 Zalik 等人<sup>[1]</sup>的介绍,这些方法有:

- 1) 三角网扩张法<sup>[2-3]</sup>;
- 2) 增点法<sup>[1,4-9]</sup>;
- 3) 分治法由 Lee 等人早在 1980 年提出,后来由 Dwyer 于 1987 年进行了改进<sup>[10-11]</sup>;
- 4) 扫描线法如 Fortune 法<sup>[12]</sup>和 Zalik<sup>[1]</sup>法。刘永和等人<sup>[13]</sup>提出的横向扩张法也属于扫描线算法,其执行效率与分治法相当。

目前大部分算法都可以归结为以上 4 类。由以往的实测显示,这些算法中运行效率最高的是 Zalik 法,其次为分治法。分治法虽然诞生较早,执行效率很高,也较为著名,但实际应用却不多,主要源于其三角网合并过程难于理解和编程实现。本文提出了借助算子来解决分治算法的这个问题。

关于 Delaunay 三角网构网技术,以往人们关注的基本单位是算法。尽管也意识到了如三角网局部优化这样的重要步骤,但除此之外人们很少去关注能在多种算法通用的其他步骤。这导致了重要算法步骤在不同算法中难以重复利用的问题,这是因为各种算法步骤在编程时可能会混在一起,独立性较差,而且对于同一算法而言,不同的人编写的程序会有很大差别,即使同一个人在不同时间也会写出差别较大的程序。这里所述的算子(operator)是指可不经修改直接在多种算法中使用的功能函数组合,它是比算法更小的范畴。算子是一个逻辑概念,不能完全等同于子程序。算子有 3 个性质:一是算子应具有一定的独立性,算子之间要具有很低的耦合度;二是算子应具有通用性,采用通用接口,可用于不同的算法和应用之中;三是算子应在算法中能发挥很关键的作用且具有一定的设计难度,它是一个重要的子算法。由此来看,不是任何算法步骤或者子程序都能称之为是一个算子。每个算子在具体的计算机程序中可由一至多个子程序构成。多个算子的组合使用可以构成一种特定的构网算法。算子之间较低的耦合度能保证大多数情况下只需要一些简单的调用就能实现完整的算法。通用算子的使用可以使算法逻辑变得十分清晰,可提高算法设计的可靠性和容易度,同时不影响算法的执行效率。

在已往的算法步骤中独立出用创建独立(或初始)三角形和实现局部优化的算子,并设计了 3 种用于在三角网中增加每个三角形、填充已有三角网的凹边界、合并子三角网等算子,用于简化分治算法中的子网合并问题。

## 1 适于通用算子的数据结构

采用一种基于有向边的结构:有向边包含始点和终点、左侧邻接三角形以及反向边。三角形的定义设有存放 3 条邻边的数组,其 3 条邻边在数组中

的存放为逆时针顺序。三角形的顶点由其 3 条边隐含表示,即第 1 条边的起始点为三角形的第 1 个顶点,以此类推。显然,两个相邻三角形之间的边为共享两个顶点的两条方向相反的有向边。用 3 个动态数组结构存放点、边和三角形的对象记录。另需专门为三角网边界设置一个存放边记录的链表结构。以类 C++ 语法给出的定义如下:

```
struct DPoint //顶点的定义
{
    Double x;
    Double y;
};

vector< DPoint > Points; //存放点的数组容器对象
struct DEdge //有向边的定义
{
    int beginPoint; //边的始点在 Points 数组中的索引下标
    int endPoint; //边的终点在 Points 中的下标
    int ltri; //本边的左邻三角形下标
    int ContraryEdge; //本边的反向边下标
    int Index; //本边的存放位置下标
};

vector< DEdge > edgeList; //存放边的数组容器
struct DTriangle //三角形的定义
{
    int Edges[3]; //三角边的位置下标
    int index; //本三角形的存放位置下标
};

vector< DTriangle > triangleList; //存放三角形的数组容器
list< DEdge > hull; //存放边界上边的链表
```

根据该数据结构,有以下拓扑推论存在:1)三角形边的左侧为三角形内部,右侧为外部;2)三角形外边界为顺序相接的逆时针闭合环,无论是最终的凸包边界,还是更新过程中的临时非凸包边界。

## 2 Delaunay 三角网构建的通用算子

为了实现 Delaunay 三角网的合并,从已有算法中抽取独立三角形创建算子和局部优化及三角形对角边交换算子,又设计了扩展三角形算子、凹边界填充算子及子网合并算子。这些算子是基于相同的有向边数据结构,且构网时对三角网外边界链表上的操作依赖很大。算子间有简单的相互调用和依存关系,是构成算法的统一整体,但又能通用于其他算法的实现。

### 2.1 独立三角形创建算子

独立三角形是指不依托任何已存在的边而直接

生成的三角形,用于构建初始三角形。

用给定的任意 3 点  $p_0$ 、 $p_1$ 、 $p_2$  构建独立三角形,首先判断  $p_2$  是否位于  $p_0p_1$  构成有向边的左侧,若是(符合逆时针排列规则)则直接生成首三角形  $\Delta p_0p_1p_2$  及有向边  $p_0p_1$ 、 $p_1p_2$ 、 $p_2p_0$ ,否则逆转前两个点的有向边为  $p_1p_0$ ,生成首三角形  $\Delta p_1p_0p_2$  及有向边  $p_1p_0$ 、 $p_0p_2$ 、 $p_2p_1$ 。在创建了三角形及相应的 3 条边记录之后,需要设置 3 条边的反向边(ContraryEdge)为空,左三角形(ltri)设为当前新三角形。

### 2.2 扩展三角形算子

扩展三角形是指以三角网外边界上的一条边为基边向三角网外侧一点  $P$  扩展而创建新三角形。生成时,需要在 triangleList 数组中注册新三角形,并生成和注册新三角形的 3 条邻边,更新边界链表。3 条邻边分别是基边的反向边、新三角形的右侧边和左侧边。由于基边的始点与  $P$  构成右侧边,与基边的前驱相接,而其终点与  $P$  构成左侧边,与基边的后继相接。最终基边必须从链表 hull 中删去记录。

只有那些没有反向边的新边才能加入 hull 作为边界边,因此还需要检查 hull 中是否有其反向边。而右侧边的反向边若已存在,则一定是基边的前驱,左侧边的反向边若已存在,则一定是基边的后继。因此不需要任何遍历,直接将右侧边与基边的前驱比较,左侧边与基边的后继比较即可完成检查,并更新边界链表。

当未找到右侧边或左侧边的反向边时,在边界链表 hull 中,将右侧边作为基边的前驱插入,将左侧边作为基边的后继插入,最后在 hull 中删除基边的记录。如图 1(a)中,当  $ab$  为基边时,则需要生成其反向边  $ba$ ,以及右侧边  $ae$ 、左侧边  $eb$ ;在 hull 中将  $ae$  和  $eb$  分别作为基边  $ab$  的前驱和后继插入,并删除基边  $ab$ ,得到图 1(b)的效果。

当新生成的左侧边或右侧边在 hull 中已有反向边时,则它与反向边均不再是外部边界,因此在互设各自的 ContraryEdge 后,需要将原 hull 中反向边的记录删去。如图 1(b)中,当  $bc$  作为基边时,右侧边  $be$  与 hull 中原有的边  $eb$  互为反向边,从 hull 中删除  $eb$  的记录,而左侧边  $ec$  由于没有反向边,因此它作为  $bc$  的后继被加入 hull,最后  $bc$  的记录被从 hull 中删除,得到了图 1(c)的效果。

### 2.3 递归局部优化和三角形对角边交换算子

当生成一个新三角形后,需要执行 LOP 优化(即 Lawson 提出的三角网局部优化方法)<sup>[9]</sup>,即当一个三角形的外接圆包含了其某邻接三角形的对角

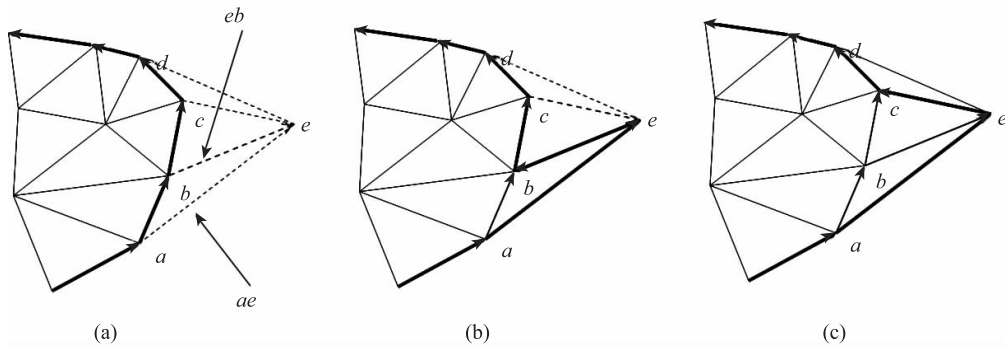


图1 扩展三角形的过程

Fig. 1 The process of expanding triangles

顶点时,需要交换对角线。该优化采用递归执行方式,即将其发生交换后的三角形再执行同样的判断和优化,还要负责三角形交换操作以及与之相关的一系列繁琐的拓扑关系维护操作。如果每次新扩展一个三角形时就立即执行 LOP 优化,则可以扩展三角形算子的末尾加上对递归 LOP 优化算子的调用。该算子简称为 LOP 优化算子。

#### 2.4 凹边界填充算子

在 Zalik 扫描线算法中,由于会生成非凸包边界,这时凹边界处需要补充三角形。为了该填充自动化,采用递归执行方式。填充方式是找到凹区内的一条边,以此边为基边开始向外连接新三角形。新三角形只可能是基边与其前驱或后继相连而成,且前驱和后继必须位于基边的右侧。因此需要分别判断前驱和后继在基边的哪一侧,有 3 种情况:

- 1) 前驱和后继都位于基边的左侧,则表明位于凸区,不需要填充;
- 2) 前驱和后继只有一个位于基边的右侧,则位于右侧的这条边与基边用扩展三角形算子生成新三角形;
- 3) 前驱和后继都位于基边的右侧,则比较它们与基边之间的夹角,基边与夹角大的连成新三角形(使用扩展三角形算子)。

当生成新三角形时,必然会有一条新边生成,将该新边作为新的基边进入下一递归循环中继续上面的过程,直至遇到第 1) 种情况结束,此时该凹区已被填充完毕。

在图 2(a) 中,当前基边为  $bc$ ,则其后继  $cd$  和前驱  $ab$  都位于其右侧,因此根据夹角可确定前驱  $ab$  与基边相连成新三角形。之后,新生成的  $ac$  成为新的基边(见图 2(b)),其前驱  $a'a$  和后继  $cd$  都位于其右侧,根据夹角对比后确定后继  $cd$  可与基边相连成新三角形。

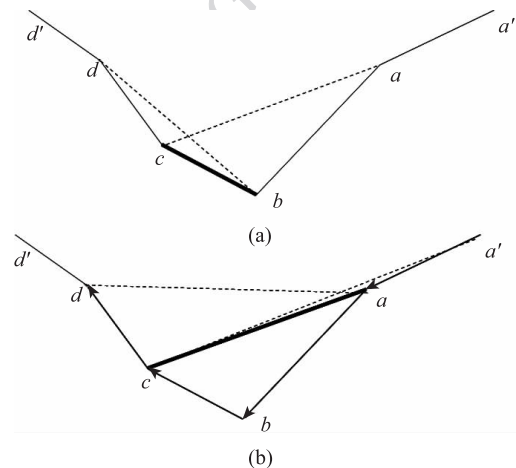


图2 凹边界填充示例

Fig. 2 The basin-filling process of the border hull

#### 2.5 子网合并算子

传统的分治算法中,子网合并时是通过找出相邻子网的上下支撑线,由支撑线开始向内部连接三角形。支撑线的查找是首先要找出各子网的上下极值坐标点。由于存在如图 3(a) 中下支撑线与子网相交的问题,支撑线不能直接由极值点连接来完成。因需要考虑应用中可能遇到的各种异常问题,需要一个繁琐的判断和迭代。此外,传统分治算法没有将 LOP 优化过程独立出来,而是与合并子网的过程混合在一起,显得代码十分臃肿繁杂,大大增加了程序调试的难度。

采用一种利用子网凸包边界链表结合填充凹边界算子来巧妙构建一种子网合并方法。其基本思路是先连接后填充,即找出左侧三角网的极右顶点和右侧三角网的极左顶点,然后以左侧极右顶点两条邻边中选出一条合适的边向极左顶点连成一个新三角形,再把两个相邻子网的边界链表合并成一个新的边界链表,至此便完成了两个子网的连接(见图 3(c)),剩余的工作是由凹边界填充算子来补充其他

的三角形。具体步骤如下:

1) 找出以左侧子网的最右极点  $R_{\text{far}}$  (如图 3(a) 中的  $g$  点) 为起点的边 (如图 3(a) 中的  $ge$ , 要求以链表结点表示的边) 和右侧子网的最左极点  $L_{\text{far}}$  (如图 3(a) 中的  $f$  点)。令右侧子网极左顶点起始的边为  $E_R$ 。

2) 判断左侧子网找到的边节点, 其前驱边和后继边中找出任一个符合以右侧面向极左点  $L_{\text{far}}$  的边。

该步是为了避免生成与已有子网交叉的新三角形。此时找到的符合条件的边仍为  $ge$ 。

3) 以步骤 2) 找出的边与左子网中的极左点相连生成新三角形 (直接使用扩展三角形算子)。令原有基边的后继为  $E_{\text{next}}$ , 前驱为  $E_{\text{prev}}$ , 新生成的两个外侧边分别在  $E_{\text{next}}$  之前和  $E_{\text{prev}}$  之后插入。两个子网虽然共享了同一顶点  $L_{\text{far}}$ , 但各自的边界仍是独立的。

在图中, 此时  $gf$  与  $fe$  在原来  $ge$  的位置上接入左侧子网边界, 左侧子网边界闭合环变为  $gfebia$  顶点序列, 右侧子网边界闭合环变为  $dhfck$  顶点序列。

4) 将右侧子网边界按序接入左侧子网边界链表: 令  $E_{\text{next}}$  的前驱为  $E_L$ ,  $E_L$  即为左侧子网中以  $L_{\text{far}}$  为起点的新边; 将右侧子网中从  $E_R$  开始向后继方向的边逐条作为  $E_L$  的前驱插入。此时得到了图 3(c) 的边界链表效果。

5) 从新生成的两条边开始, 用填充凹边界算子分别向上和向下递归填充合并后三角网中的凹边界区。

以上步骤仅需要较少的条件判断, 因而需要考虑的复杂情形较少, 一些创建三角形对象、三角形 LOP 优化以及凹边界填充等任务交由通用算子来完成, 使逻辑上十分明了。

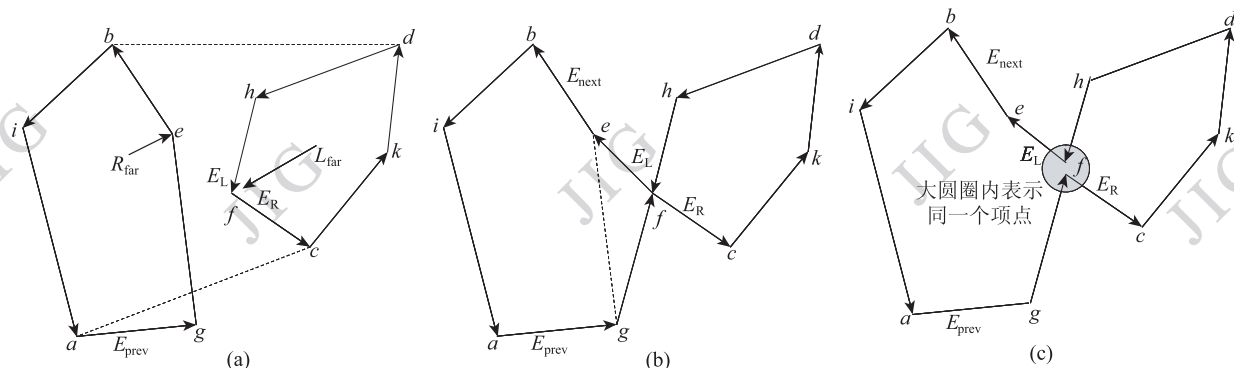


图 3 子网合并示例

Fig. 3 The merging process of two sub-triangulations

### 3 分治算法的实现

参考 Lee&Schachter 分治算法的实现, 将点集采用以横(纵)坐标排序, 采用递归分割点集来实现, 借助独立三角形算子和子网合并算子, 而子网合并算子中又使用了扩展三角形算子、LOP 优化算子和凹边界填充算子。下面是(C#语言)采用通用算子实现的分治算法核心代码:

```
public Linklist DivConquer(int left, int right) {
    int n = right - left + 1; // 点数
    if (n < 6) { // 构建一个三角形
        // 将前三个点构建三角形
        Linklist hull = SingleTriangle(left, left + 1, left + 2);
        // 后两个点扩张进去
        if (n == 4) {
            expandPoint(left + 3, hull);
        }
    }
}
```

```
if (n == 5) {
    expandPoint(left + 3, hull);
    expandPoint(left + 4, hull);
}
return hull;
}
else {
    int divider = n / 2;
    Linklist hull1 = DivConquer(left, left + divider - 1);
    Linklist hull2 = DivConquer(left + divider, right);
    Linklist hull = MergeHull(hull1, hull2);
    return hull;
}
```

借用通用算子实现的分治法在理论上应该与 Lee 等人的原始分治法<sup>[10]</sup>具有相近的执行效率。在 Lee 等人的实现中, 子网合并过程也涉及了三角形边的检验和交换, 它与 LOP 优化交换本质上是

相同的。

## 4 算子的通用性

构建每种算法可能使用的通用算子见表 1。独立三角形创建算子一般用于创建三角网中的首三角形,它几乎在各种算法中都会用到,但大多只会被执行一次,只有在分块算法中它会在生成每一子块时各执行一次。LOP 优化算子的应用非常普遍,而且在增点法、分治法和横向扩张法中其调用十分频繁,几乎每增加一个三角形都要递归调用多次。在一些分块

算法(适合于并行处理)中,子块合并后会在块与块之间有很多窄细三角形需要使用该算子进行优化。扩展三角形算子是除增点法外另一种在其他算法中均可广泛应用的算子,需要说明的是该算子与凹边界填充算子和子网合并算子可在同一算法中结合使用。凹边界填充算法是 Zalik 扫描线法中用于完善最后的凹边界而提出的,而子网合并算子的提出使它在两个子网连接后为补充剩余缝隙可发挥重要作用。因而这两个算子的结合可以通过十分简洁的方式来解决各种三角网合并问题,也可能成为并行和分布式算法的关键技术。

表 1 三角形构网算法中可用的通用算子

Table 1 Universal operators used in many Delaunay triangulation algorithms

| 算 法                       | 独立三角形创建算子 | 扩展三角形算子 | LOP 优化算子 | 凹边界填充算子 | 子网合并算子 |
|---------------------------|-----------|---------|----------|---------|--------|
| 增点法 <sup>[1,4-9]</sup>    | 1         |         | $n$      |         |        |
| 扩张法                       | 1         | $n$     |          |         |        |
| 逐块生长法 <sup>[15]</sup>     | 1         | $n$     | $n$      |         |        |
| Fortune 法                 | 1         | $n$     |          |         |        |
| 分治法                       | $n$       | $n$     | $n$      | $n$     | $n$    |
| Zalik 法 <sup>[1]</sup>    | 1         | $n$     | $n$      | $n$     |        |
| 横向扩张法 <sup>[13]</sup>     | 1         | $n$     | $n$      |         |        |
| 其他分块算法 <sup>[16-18]</sup> | 1         | $n$     | $n$      | $n$     | $n$    |

注:表中 1 和  $n$  分别表示算子的执行次数为 1 次和多次

## 5 算法测试

### 5.1 运行效果检验

在同一个 C# 程序中借助这些通用算子实现了横向扩张法、Zalik 扫描线法、逐块生成法、分治法以及多种其他算法。这些算法全部能够在大数据量三角网生成的应用中全部正确构网,并具有完善的拓扑关系,可支持基于行走法的三角形定位(图 4(a)为三角网(6 万个点)局部,深(蓝)色条带为从一三角形起始用行走法定位一个目的三角形(红色)时的沿途三角形组成的路径;3 个绿色三角形为目的三角形的 3 个邻居)。通过使用随机生成的点数据集的测试,表明这些算法的任意一种都能获得图 4(a)的显示效果。

将分治法得到的 Delaunay 三角网用于表达 LiDAR 点云数据(含有高程),计算其中每个三角面片的法向,并假定一个虚拟的光源法向,利用 Lambert 漫射光照模型生成反射光强,得到了图 4(b)的渲染效果。该图为 40 万个点的 LiDAR 点云数据三角网

中部分区域的光照渲染,右上角的一系列长条三角形并非构网错误,而是点云截取时造成个别点的分布太远的构网效果,图 4(b)清晰地反映了地面景观,由于算法执行过程中,递归式 LOP 优化算子使用空圆准则来优化三角网,保证了最终的三角网满足 Delaunay 三角网的要求。因此所有算法都能够正确构网。

### 5.2 运行效率测试

下面测试多种算法的运行效率,以验证算法对比是否与理论上及前人测试有一致的结论。尽管很多文献都声称所介绍的算法平均复杂度为  $O(n \log n)$ ,但实际执行效率却有很大差异。有关各种算法空间复杂度的理论分析比较可参见文献[14]。这里只通过对实际数据集的应用测试来比较。Zalik<sup>[1]</sup>曾指出,算法的实际运行效率受很多因素影响,包括计算机硬件、运行平台、编程语言、编译器及其选项设置、编程者的编程经验以及具体的程序实现细节等。因此单靠少数实例获得的比较结果也许并不完全可靠。但当大多数条件都很接近时,基于实测的比较能在很大程度上反映算法本身的运行效率。

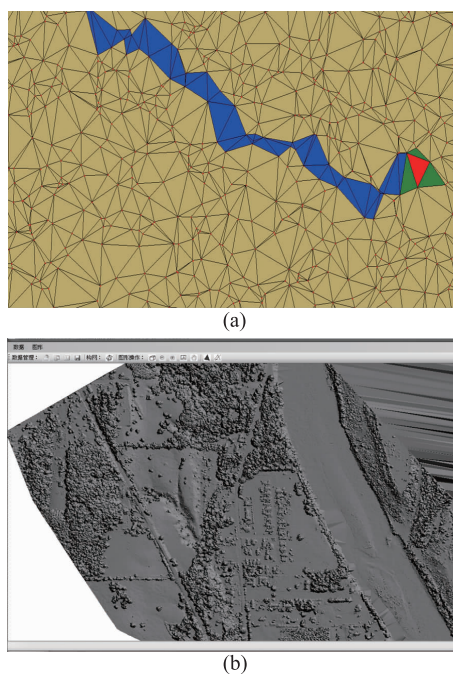


图 4 Delaunay 三角网的生成效果

Fig. 4 Two applications of Delaunay triangulation constructed using the universal operators

参与比较的算法如下:

1) 增点法<sup>[1,49]</sup>,用行走法辅助来加快三角形查找的效率,但没有采用任何分块技术,其执行效率比较低;

2) 逐块生长法<sup>[15]</sup>,基于分块方式和借助三角网扩张法原理实现的算法;

3) Fortune 扫描线法,是由 Matt Brubeck 实现(<http://www.cs.hmc.edu/~mbrubeck/voronoi.html>),经少量修改后由笔者以模块的形式加入 C# 所编程序,与其他的算法实现具有相同的数据结构基础;

4) 分治法<sup>[10-11]</sup>,利用高效的通用算子实现;

5) Zalik 法<sup>[1]</sup>,其运行效率慢于 Zalik 的实现 C 语言版本(因不同的数据结构以及语言执行性能所导致);

6) 横向扩张法,是在文献[13]基础上通过扩展三角网算子改进后的版本,减少了对边界链表的搜索,其算法效率比改进前有所提高。

以上参与比较的 6 个算法是基于相同的数据结构。其中横向扩张法、Zalik 和分治法是充分利用本文介绍的通用算子构建而成。逐块生长法不需要合并过程,用到了 3 种通用算子,但还需要其他的算子支持。而 Fortune 算法并非本文作者所实现,因而也未用到任何通用算子。

在 Intel i5 CPU(2.53 GHZ), Windows XP 和 Microsoft .net 3.5 的笔记本电脑上对用以上算法进行对比测试。测试所用数据为由随机生成的点,分别使用高斯分布的点集和均等分布的点集(见图 5)。

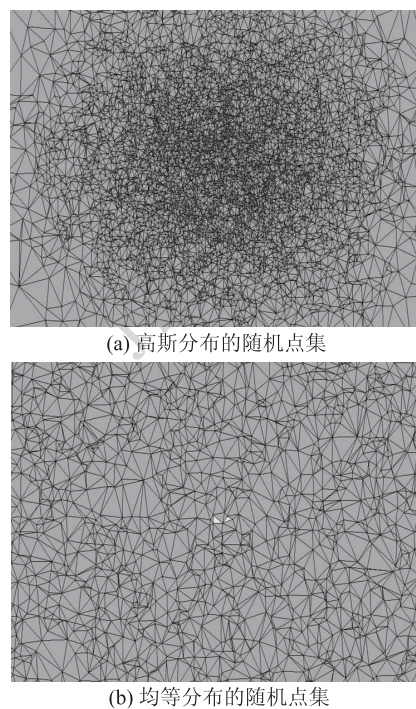


图 5 随机点集生成的三角网

Fig. 5 Delaunay triangulation based on artificial point

通过表 2 和表 3 的测试结果及对比可见:

1) 基于行走法三角形搜索的增点法效率很低,当有 5 万个均等分布点时就需要花费 1 min 左右的时间,而高斯分布花费的时间则接近 2 min;当 10 万个点时,其耗时已无法承受;逐块生长法运行效率仅高于增点法,但当有 10 万个点时花费时间为 14 s,效率远低于其他算法。

2) Fortune 扫描线法比增点法快得多,但比其他 3 种算法慢得多,当点数为 100 万个时,它需要运行 260 s 以上。对如此的大数据量,它能在 3~4 min 内完成也表明其执行效率并不算慢,仍具有较好的适用性。

3) Zalik 扫描线法是这些算法中运行效率最快速的,它比分治法和横向扩张法要快一倍,这归功于 Zalik 所提出的启发式限制,大大减少了狭长三角形的生成,从而避免了深度的递归式 LOP 优化。但是它有一个问题是程序实现存在的异常情况难于调试发现,可能与算法实现不够严密有关,会偶尔产生一些未知的计算错误,导致构网失败。这表明 Zalik 法至少存在难以调试的问题。

4) 分治法在 100 万个以下点数时要略慢于横向扩张法,而在 100 万个点时效率与横向扩张法相同;这与横向扩张法的边界搜索缺少优化有关,即当点数

增大时,边界上的边数也就增大,使搜索效率降低了。但横向扩张法对 200 万个点仍能成功构网,而分治法此时运行崩溃,这是由于它更加占用内存所造成的。

表 2 均等分布下不同点数的三角网构网耗时

Table 2 The CPU time required for triangulations of different numbers of artificial points in a uniform distribution

| 点数        | 增点法    | 逐块生长法  | Fortune 扫描线法 | 分治法    | 横向扩张法 | Zalik 扫描线法 |
|-----------|--------|--------|--------------|--------|-------|------------|
| 10 000    | 2.75   | 0.203  | 0.453        | 0.125  | 0.125 | 0.046      |
| 20 000    | 11.843 | 0.734  | 1.125        | 0.281  | 0.250 | 0.093      |
| 30 000    | 26.296 | 1.421  | 1.937        | 0.484  | 0.375 | 0.171      |
| 40 000    | 47.89  | 2.406  | 3.031        | 0.656  | 0.593 | 0.281      |
| 50 000    | 79.062 | 3.671  | 4.125        | 0.875  | 0.671 | 0.343      |
| 100 000   | —      | 14.156 | 10.890       | 1.812  | 1.468 | 0.937      |
| 500 000   | —      | —      | 102          | 10.890 | 11    | 5.093      |
| 1 000 000 | —      | —      | 261          | 24     | 24    | 11.421     |
| 2 000 000 | —      | —      | —            | X      | 57    | 28         |

注:符号“—”代表运行时间太长,而没有成功实现测试;“X”表示由于内存不足造成的测试失败。

表 3 高斯分布下不同点数的三角网构网耗时

Table 3 The CPU time required for triangulations of different numbers of artificial points in a Gaussian distribution

| 点 数       | 增点法    | Fortune 扫描线法 | 分治法   | 横向扩张法 | Zalik 描线法 |
|-----------|--------|--------------|-------|-------|-----------|
| 10 000    | 3.031  | 0.390        | 0.125 | 0.093 | 0.031     |
| 20 000    | 12.062 | 1.046        | 0.296 | 0.250 | 0.093     |
| 30 000    | 27.843 | 1.796        | 0.468 | 0.390 | 0.140     |
| 40 000    | 58.812 | 2.781        | 0.625 | 0.515 | 0.265     |
| 50 000    | 111    | 3.750        | 0.828 | 0.703 | 0.328     |
| 100 000   | —      | 10           | 1.718 | 1.484 | 0.703     |
| 500 000   | —      | 93           | 11    | 10    | 5.343     |
| 1 000 000 | —      | 265          | 24    | 24    | 11.781    |
| 2 000 000 | —      | —            | X     | 58    | 25.7      |

注:符号“—”和“X”的意义同表 2。逐块生长法未进行针对高斯分布数据点的测试。

可见,基于通用算子的分治算法,其执行效率是比较高的,仅次于将 LOP 优化减少到极致的 Zalik 算法,而与横向扩张法相当。分治算法表现得如此高效,这与国际上一些学者<sup>[14]</sup>的测试得到的结论一致。需要说明的是,通用算子的使用并不会影响构网效率,即构网效率只与具体的算法思路有关。这是由于通用算子只是表现为在不同算法中均可使用的独立的子程序组,其执行效率并不会因为具有通用性而降低。相反,正是由于它们之间的低耦合度

降低了算法的复杂性,有利于算法代码的规范化,使冗余代码的执行尽可能降低到最少,从而保证了算法的效率。

## 6 结 论

提出了以算子作为构建 Delaunay 三角网生成算法的基础,重点设计了扩展三角形算子、凹边界填充算子和子网合并算子,并用这些算子解决了分治

算法中子网合并例程偏复杂的问题。这些算子具有很强的通用性,可以在特定数据结构基础上直接应用于其他 Delaunay 三角网算法。利用这些算子成功构建了分治算法、横向扩张法(一种扫描线法)、Zalik 法、增点法等。

使用低耦合度的算子可以简化不同三角网算法的程序设计。通用算子能够保证算法执行具有很好的执行效率,这是因为本文设计的通用算子充分利用了一种基于有向边的数据结构,以及用链表表示的三角网外边界,利用拓扑关系和链表操作高效执行,避免了以往在搜索时需要遍历整个外边界集合的耗时执行。

文中所述的算子是在设计多种 Delaunay 三角网算法中抽象出来的通用且独立执行的子程序组,通过优化设计和合适的数据结构相配合而成,采用通过简单调用就可实现算法的无缝执行。但算子和相应的数据结构是一个统一的整体,不能随意移植到其他具有不同应用特点的 Delaunay 三角网构网中。

独立三角形创建算子和 LOP 优化算子实际上是在增点法、分块法等许多算法共用的计算步骤。扩展三角形算子、凹边界填充算子以及子网合并算子是本研究的重点,它们都巧妙利用了边界边存放链表的功能。其中三角网合并算法是除分治法外其他很多新算法(如<sup>[16-18]</sup>)尤其是并行执行的构网算法所需要必要步骤,但一直以来是一大难点,而子网合并算子的提出对于解决这类问题提供了一种简便的方法,也为今后解决各种并行和分布式三角网构网算法提供了很好的技术参考。

## 参考文献(References)

- [1] Zalik B. An efficient sweep-line Delaunay triangulation algorithm [J]. Computer-Aided Design, 2005, 37(10): 1027-1038.
- [2] Mavriplis D J. An advancing front Delaunay triangulation algorithm designed for robustness [J]. Journal of Computational Physics, 1995, 117(1): 90-101.
- [3] Du C J. An algorithm for automatic Delaunay triangulation of arbitrary planar domains [J]. Advances In Engineering Software, 1996, 27(1-2): 21-26.
- [4] Guibas L J, Knuth D E, Sharir M. Randomized incremental construction of Delaunay and Voronoi diagrams [J]. Algorithmica, 1992, 7(4): 381-413.
- [5] Kolingerova I, Zalik B. Improvements to randomized incremental Delaunay insertion [J]. Computers & Graphics, 2002, 26(3): 477-490.
- [6] Sloan S W. A fast algorithm for constructing Delaunay triangulations in the plane [J]. Advances in Engineering Software and Workstations, 1987, 9(1): 34-55.
- [7] Zalik B, Kolingerova I. An incremental construction algorithm for Delaunay triangulation using the nearest-point paradigm [J]. International Journal of Geographical Information Science, 2003, 17(2): 119-138.
- [8] Zhou S, Jones C B. HCPO: an efficient insertion order for incremental Delaunay triangulation [J]. Information Processing Letters, 2005, 93(1): 37-42.
- [9] Lawson C L. Software for C1 Surface Interpolation. Mathematical Software III [M]. New York: Academic Press, 1977.
- [10] Lee D T, Schachter B J. 2 algorithms for constructing a Delaunay triangulation [J]. International Journal of Computer & Information Sciences, 1980, 9(3): 219-242.
- [11] Dwyer R. A faster divide-and-conquer algorithm for constructing delaunay triangulations [J]. Algorithmica, 1987, 2(1): 137-151.
- [12] Fortune S. A sweep line algorithm for Voronoi diagrams [J]. Algorithmica, 1987, 2(1): 153-174.
- [13] Liu Y H, Wang Y P, Qi Y A. Horizontal expanding method—a quick algorithm for generating Delaunay triangulation from points in the plane [J]. Geo-Information Science, 2008, 10(1): 20-25. [刘永和, 王燕平, 齐永安. 一种快速生成平面 Delaunay 三角网的横向扩张法 [J]. 地球信息科学, 2008, 10(1): 20-25.]
- [14] Yu J, Lv P, Zheng C W. A comparative research on methods of Delaunay triangulation [J]. Journal of Image and Graphics, 2010, 15(8): 1158-1167. [余杰, 吕品, 郑昌文. Delaunay 三角网构建方法比较研究 [J]. 中国图象图形学报, 2010, 15(8): 1158-1167.]
- [15] Liu Y H, Wang Y P, Qi Y A. A simple and quick block-by-block generating method for generating Delaunay triangulation from points in the plane [J]. Science of Surveying and Mapping, 2008, 33(6): 133-135. [刘永和, 王燕平, 齐永安. 一种简单快速的 Delaunay 三角网逐块生成算法 [J]. 测绘科学, 2008, 33(6): 133-135.]
- [16] Zeng M S, Tian D L, Guo J M. An efficient Delaunay triangulation algorithm basing on grid partitioning approach [J]. Control & Automation, 2006: 127-130. [曾闽山, 田冬玲, 郭吉民. 一种基于网格划分的高效 Delaunay 三角网格化算法 [J]. 微计算机信息, 2006: 127-130.]
- [17] Huang Z G, Chen J J, Zheng Y. Triangulated irregular network based chunk gridding algorithm for terrain rendering [J]. Journal of Zhejiang University: Engineering Science, 2009, 43(10): 1939-1944. [黄争舸, 陈建军, 郑耀. 基于不规则三角网的分块地形网格生成算法 [J]. 浙江大学学报: 工学版, 2009, 43(10): 1939-1944.]
- [18] Shi S, Zhu Q F, Tang L Y. An algorithm for building Delaunay triangulation of high efficiency [J]. Computer Engineering, 2005, 31(18): 87-89. [石松, 朱泉峰, 唐丽玉. 四叉树高效 Delaunay 三角网生成算法 [J]. 计算机工程, 2005, 31(18): 87-89.]