Delaunay 三角网生成的改进算法

青文星 陈 伟
（西南石油大学计算机科学学院 成都 610500）

摘 要 在石油领域，经过多年的研究和发展，一些网格相关的基本算法如 Delaunay 三角网生成算法等已逐渐趋于成熟。然而伴随技术的发展，行业对相关算法和软件的要求也不断提高，现有方法已经不能满足实际需要。文中分析了目前常规三角网生成算法的特点和缺点，提出了一种将逐点和分治相结合的快速生成 Delaunay 三角网格的方法，使得布点数目对构网效率不会产生较大影响。通过大量测试验证了该算法在正确性、稳定性和效率上较传统算法具有较大优势。

关键词 Delaunay 三角网，逐点法，分治法

中图法分类号 TP311 **文献标识码** A

Delaunay Triangular Mesh Optimization Algorithm

QING Wen-xing CHEN Wei
(School of Computer Science，Southwest Petroleum University，Chengdu 610500，China)

Abstract After years of research and development in the petroleum field，some grid-related basic algorithms such as Delaunay triangulation algorithm have gradually matured. However，with the development of technology，the requirements of relevant algorithms and software in the industry are constantly improving，so the existing methods can no longer meet the actual needs. This paper studied and analyzed the characteristics and shortcomings of the conventional triangulation algorithm. It proposed a method by using the idea of point-by-point and divide-and-conquer to quickly generate Delaunay triangle mesh，so that the number of layout points will not have a great impact on the efficiency of network construction. A large number of tests verify that the algorithm has greater advantage compared with traditional algorithms in terms of correctness，stability and efficiency.

Keywords Delaunay triangulation，Point by point method，Divide-and-conquer method

Delaunay 三角剖分算法是图形处理研究的重点之一，与结构化网格相比，非结构化网格可以更好地适应不规则计算区域并快速实现网格的自动化生成，因此非结构化网格得到了人们的重视。目前，voronoi 网格作为典型的非结构网格，因具有较强的灵活性和局部正交的特点，同时拥有固定的数据结构，被灵活运用在图像处理的方方面面。基于 voronoi 网格和 Delaunay 三角网的对偶关系，我们可以间接优化和改进 Delaunay 三角网的生成算法，从而提高 voronoi 网格的质量，进而得到满足需求的图像。

本文通过分析一般三角网生成算法的优缺点，取长补短，得到将逐点法和分治法相结合的一种快速生成三角网格的算法。

1 传统三角网算法

到目前为止，国内外大量文献表明三角网格剖分技术已经十分成熟，并已有了许多公开有效的算法及实现。其中，Delaunay 准则方法是最为有效的基础算法。在一般情况下，平面域网格剖分算法中，Delaunay 准则具有“三角剖分最小内角最大”的性质，最大限度地减少了畸形三角形（某个内角过小），为后期模型表征及计算机迭代求解提供了可靠的网格质量保证[1]。

Delaunay 三角网有两条重要的性质，也是其优化准则[2]，即外接圆性质和最大最小角度性质。这两条特性，决定了 Delaunay 三角网具有极大的应用价值。Miles 证明了 Delaunay 三角网是最为均匀的三角网；Lingas 也从不同角度证明了“Delaunay 准则下三角网最优”的结论。以上定义及性质是建立 Delaunay 三角网的算法依据。

Tsai 根据实现过程，把生成 Delaunay 网的各种算法分为 3 类：逐点插入法、分治法及三角网生长法；并对 3 种算法的性能进行了比较分析，对比了各种算法的时间复杂度[3]。

1.1 分治算法

分治算法的基本步骤如下：

1）以横坐标为主、纵坐标为辅对点集进行升序排序；

2）把点集分为近似相等的两个子集；

3）在两个子集中生成三角网；

4）用 Lawson 提出的局部优化算法 LOP 优化所生成的三角网，使之成为 Delaunay 三角网；

5）找出连接两个子集中两个凸壳的底线和顶线；

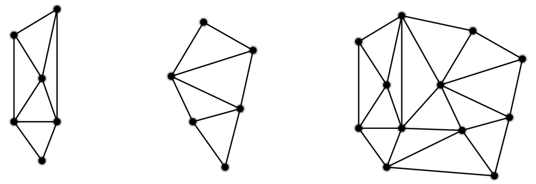
6）由底线至顶线合并两个子集的三角网；

7）递归执行步骤 2）—步骤 6），直到点集中的所有点都参

青文星（1993－），男，硕士生，主要研究方向为 GIS 与计算几何；陈 伟（1965－），男，博士，主要研究方向为科学计算可视化与石油工程计算，E-mail：ncchenwei@163.com（通信作者）。

加了 Delaunay 三角网的构造。

以上步骤显示，分治算法的基本思想是问题简化，把点集划分到足够小，使其易于生成三角网。然后把子集中的三角网合并生成最终的三角网，用 LOP 算法保证其成为 Delaunay 三角网。不同的实现方法可有不同的点集划分法、子三角网生成方法及合并方法，如图 1 所示。



（a）子网1　　（b）子网2　　（c）合并

图 1　三角网合并

1.2　逐点插入算法

1977 年，Lawson 提出了新的 Delaunay 三角网的算法——逐点插入法[4-5]，其可以在逐点增加的情形下动态地求出点集的 Delaunay 三角剖分。之后，Lee 和 Schachter，Sloan，Bowyer，Macedonio，Floriani 和 Pareschi，Watson 和 Puppo，Tsai 先后对其进行了发展和完善。其中，Waston 提出的算法效果不错，其主要思想是不断地向原三角网加入新的点，根据点的位置不断更新三角网，如图 2 所示。

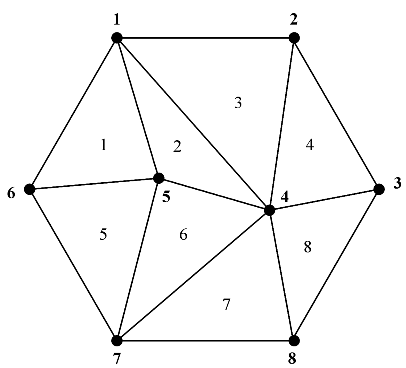


图 2　逐点更新三角网

1.3　三角网生长算法

三角网生长法的示意图如图 3 所示。

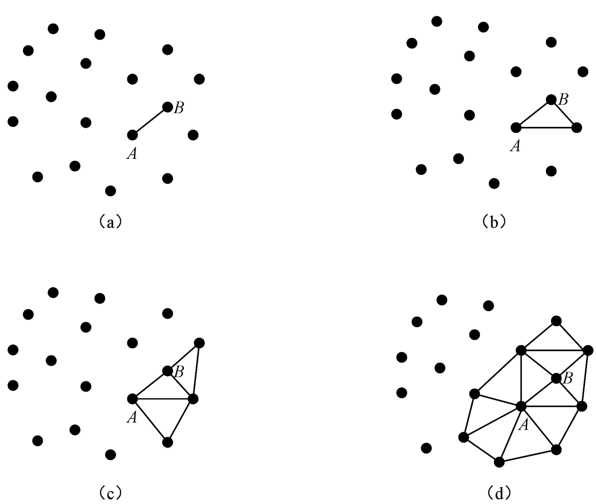


图 3　生长算法过程

1）以任一点为起始点（一般位于数据点几何中心附近）；

2）以与起始点最近的数据点相互连接形成 D-三角形的一条边作为基线，按 Delaunay 三角网的判别法则（即它的两条基本性质），找出与基线构成 D-三角形的第三点；

3）基线的两个端点与第三点相连，成为新的基线；

4）迭代以上两步，直至所有基线都被处理。

2　改进三角网算法

上述 3 类算法中，三角网生长算法和分治法是静态算法[6-8]，而逐点插入法是动态算法[9]。

逐点插入法虽然实现比较简单，占用内存较小，但其效率低。分治算法是通过递归把复杂问题简单化，但它需要较大的内存空间。把逐点插入法和分治算法相结合，可以取长补短，从而达到很好的效果，即本文提出的改进算法。

首先，凸壳的生成算法有很多，本文主要采用一种对包裹算法进行改进的算法进行凸包的构建。凸壳生成中主要涉及离散点，凸壳由有序点组成。

1）对输入的点集进行分类

算法 1　改进的凸壳算法

输入：含有 n(n≥3)个平面离散点的集合 P

输出：凸壳点集

1）如果 n<3，则返回。

2）取出点集中 x 坐标最大和最小的点，以及 y 坐标最大和最小的点（若存在多个最值点则一起取出），这些点必为凸壳边上的点。

3）连接这些点构成一个凸多边形，然后选取剩下的点依次和凸边形的边连线进行矢量判断。对于向量我们定义逆时针为正方向，那么如果有点在向量的右侧，则表示该点与当前判断向量起始点所构成的向量乘积小于 0，则将该点加入凸壳链表中，并删除当前比较向量所表示的边，将新加入的点和凸壳链表当前最后一点重新连接构成新的凸边。

4）重复步骤 3），直到所有点都进行了遍历操作，结束。

2）判断点在向量两侧的算法

设判断点为 C，向量两边的点为 A，B。我们只需求得向量 $\boldsymbol{CA} * \boldsymbol{AB}$ 的正负关系，其值大于零，表示点在右侧，将其加入凸壳链表，反之则进行下一个点的判断，如图 4 所示。

（a）$\boldsymbol{p_1p_0} * \boldsymbol{p_2p_0} > 0$，则 p_2 在 $\boldsymbol{p_1p_0}$ 右侧　　（b）$\boldsymbol{p_1p_0} * \boldsymbol{p_2p_0} < 0$，则 p_2 在 $\boldsymbol{p_1p_0}$ 左侧

图 4　利用向量积判定点位置

其中，向量积的坐标运算公式如下：

向量 $\boldsymbol{CA} = (X_1, Y_1)$，向量 $\boldsymbol{AB} = (X_2, Y_2)$

$$|\ \boldsymbol{CA} * \boldsymbol{AB}\ | = X_1 * Y_2 - X_2 * Y_1$$

对于已经生成好的凸包，我们需要进行三角剖分。

算法 2　改进的三角网算法

输入：凸壳点集，内部离散点

输出：三角网

1）读入凸壳点，以 y 坐标为最小的点 A 按逆时针方向存入。把 A 与随后的凸壳点连接形成初始三角形，这样整个凸包就被三角化了，如图 5 所示。

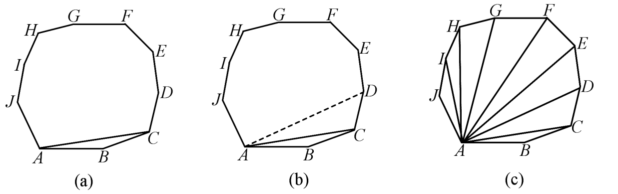


图 5 凸壳三角网初始化

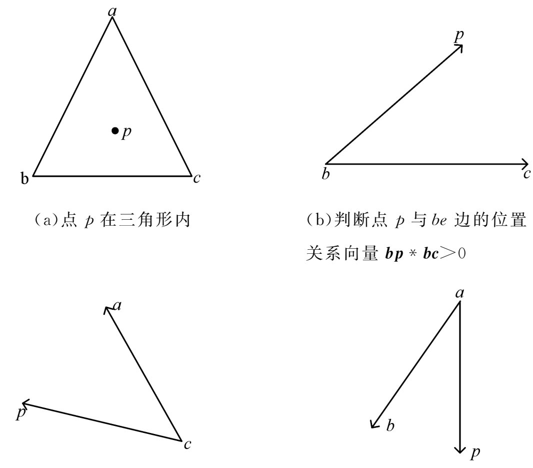
2)初步的三角网形成之后，我们就按照逐点的思想将凸包内部的点依次加入到初始三角网中。主要流程是判断点在当前三角网中哪个三角形的内部，如果在某个三角形内部，则该点连接此三角形的 3 个顶点，组成新的三角形，将原本的三角形编号删除，加入新的三角形编号。

3)再进行 Lop 优化，得到 Delaunay 三角形。

4)依次递归，直到所有内部点都进行了三角化。

判断点在具体某个三角形内部的方法，与前面确定凸壳边的思想一致，即根据插入点与三角形顶点所构成的向量，与三角形三边的向量做向量积的运算，根据数值的正负即可得到点与三角形的位置拓扑关系。一般情况下，此方法需要使插入点遍历整个凸壳中的三角形，当点足够多时，效率可能得不到保障。因此，对算法做出一些改进，尤其是当点不在当前三角形内部时，无需再逐次遍历，只需根据边结构体中当前边对应的另一个三角形即可，以此类推，直到向量积都大于零时，停止查找，如图 6 所示。

为了保证生成的三角网为 Delaunay 三角网，我们需要对生成的三角形进行 LOP 优化。运用 Delaunay 三角网的空外接圆性质对由插入点生成的三角形和相邻三角形组成的四边形进行判断。主要通过插入点到相邻三角形外心的距离和外接圆半径的大小来决定是否进行对角交换：如果前者大于后者，则表示插入点在外接圆外，不需要交换四边形对角线；如果前者小于后者，则插入点在外接圆内，需要交换。这个过程中主要要求得外接圆的圆心和外接圆半径以及插入点到圆心的距离。



(a)点 p 在三角形内

(b)判断点 p 与 be 边的位置关系向量 $\boldsymbol{bp}*\boldsymbol{bc}>0$

(c)判断点 p 与 ac 边的位置关系向量 $\boldsymbol{cp}*\boldsymbol{ca}>0$

(d)判断点 p 与 ab 边的位置关系向量 $\boldsymbol{ab}*\boldsymbol{ap}>0$

图 6 点与三角形的位置关系

设外接圆的三角形三边顶点为 $A(x_1,y_1)$，$B(x_2,y_2)$，$C(x_3,y_3)$，插入点为 $D(x_4,y_4)$，其外心坐标 P 为 (x,y)，则有：

$$x=\frac{(y_2-y_1)(y_3^2-y_1^2+x_3^2-x_1^2)-(y_3-y_1)(y_2^2-y_1^2+x_2^2-x_1^2)}{2[(x_3-x_1)(y_2-y_1)-(x_2-x_1)(y_3-y_1)]}$$

$$y=\frac{(x_2-x_1)(x_3^2-x_1^2+y_3^2-y_1^2)-(x_3-x_1)(x_2^2-x_1^2+y_2^2-y_1^2)}{2[(y_3-y_1)(x_2-x_1)-(y_2-y_1)(x_3-x_1)]}$$

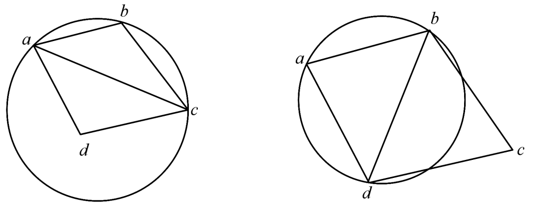
外接圆半径 r 以及插入点到圆心的距离 d 的公式为：

$$r=\sqrt{(x_1-x)^2+(y_1-y)^2}$$

$$d=\sqrt{(x_4-x)^2+(y_4-y)^2}$$

由于开方运算可能会对精度和效率产生影响，并且我们只需要半径与第四点的圆心距的大小关系，因此只需比较两者的平方大小即可，无需开方运算。

对角线交换前后的效果如图 7 所示。



(a)交换前 (b)交换后

图 7 对角线交换前后的对比

在进行 LOP 优化时，如果执行了对角线交换，改变了三角形的形状，则可能使得之前相邻的三角形不满足空外接圆准则，因此需要重新判断。这里我们可以用栈这种数据结构来对需要判断的三角形进行存储。需要 LOP 操作的则压入栈中，执行完成则出栈，同时把出栈三角形相邻的三角形压入栈中，直到空栈，一次 LOP 优化完成。

此外，当点集数目比较大时，单纯的逐点插入在效率上大打折扣，我们可以将点集划分成几个小的点集，将分治和逐点二者结合可以很大程度地提高三角网生成效率。

首先我们制定一个标准，以横坐标为主、纵坐标为辅，将所有个点纳入一个点集中，并将点集分成两个子点集，分别按照上述方法对这两个子集生成凸包和三角网，并执行 LOP 优化，得到两个 Delaunay 三角网。然后对这两个三角网进行缝合，形成最终的三角网。

这里最关键的就是对凸包进行缝合，使缝合区域能正确生成 Delaunay 三角网。

首先要确定凸壳间的顶线和底线，这两条线其实就是两者的公切线。由于计算公式繁琐，且存在鲁棒性问题，我们依旧可以利用向量积的方式来确定顶线和底线，如图 8 所示。

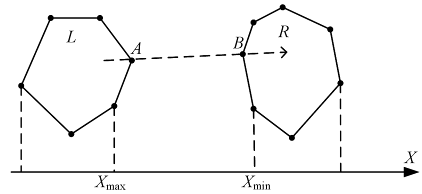


图 8 确定初始边

算法 3 三角网合并算法

输入：两个子三角网点集

输出：合并三角网点集

1)首先确定两个子集中横坐标最大最小值的点。把左子集横坐标最

大值的点和右子集横坐标最小的点连接起来。

2）寻找底线。首先，右子集中凸壳边上的点从点 B 出发，按照逆时针的顺序判断点是否在向量 **AB** 的右侧，如果是则更新 B 点为当前点，然后遍历下一个点，直到右侧没有点为止。同理，左子集按照同样的方式和方法进行判断，不过凸壳边上点的遍历顺序改为顺时针，直到右侧没有点时，当前线段 AB 就为底边。

3）寻找顶线和底线的方法一样，但判断的是点在 AB 左侧的情况，并且左右凸壳边界点的遍历顺序正好与底边相反。

4）在找完上下公切线后，对凸包进行缝合。以底边为界，左凸包从凸壳链表中按逆时针顺序，右凸包按照顺时针顺序，以底边两点为起始点，分别寻找下一个点，判断两个点的纵坐标，数值小的点和底边构成新的三角形，并且链表指向下一个点，然后对该三角形进行 LOP 优化，最后以新构成的边为底边。

5）重复步骤 4），寻找纵坐标低的点，直到两边的凸壳链表都遍历到了各自的顶线端点，合并结束，如图 9 所示。

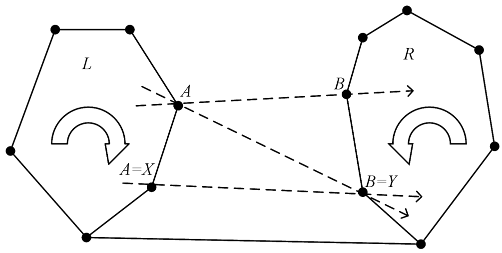


图 9　确定子凸壳间的下公切线

3　实例

笔者在 Visual Studio 2013 编译平台上进行了算法验证（CPU：AMD Phenom II X4 945 四核，内存 6 GB）。实例的初始状态如图 10 所示（布点数量为 300，并且根据点的位置关系生成了凸包）。

找到了凸包之后，我们将其内部的点逐一三角化，最终得到关于此凸包的三角网，如图 11 所示。

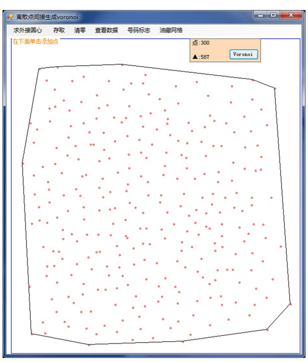


图 10　初始化布点及凸包的生成

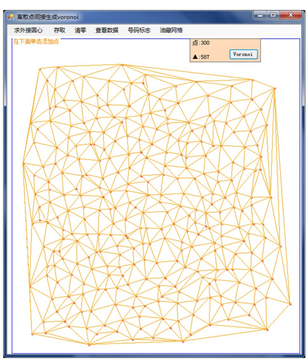


图 11　Delaunay 三角化

本文算法与文献[2]中给出的分治法三角网生成算法的时间比较如表 1 所列。

表 1　测试时间的比较

点数目/个	参考算法/ms	本文算法/ms
200	150	21
500	140	105
1000	137	137
1000（逐点）		430
2000	148	147

从表 1 可以看出，布点数较少时逐点法的布网效率远远高于分治法；而在点数目较大时，分治法的并行思想大大缩短了构网时间。因此，将逐点和分治相结合来构建三角网，能够最大限度地提高三角网的生成效率。

结束语　Delaunay 三角网在图像处理领域有着巨大的利用价值，良好的构网算法能够大大提高图形处理精度和效率。本文提出的逐点和分治相结合的构网算法，既包含了逐点法能够快速三角化的特点，又包含了分治法在布点数较多时构网效率快的优点，具有极高的应用价值。未来，我们可以在三维 Delaunay 三角网中利用这种思路进行构网，探索三维 Delaunay 三角剖分方法。

参 考 文 献

[1]　陈树强，陈学工，王丽青．判定检测点是否在多边形内的新方法[J]．微电子学与计算机，2006，23(8)：1-4．

[2]　武晓波，王世新，肖春生．Delaunay 三角网的生成算法研究[J]．测绘学报，1999，28(1)：28-35．

[3]　TSAI V J D．Delaunay Triangulations in TIN Creation：an Overview and a Linear-time Algorithm[J]．International Journal of GIS，1993，7(6)：501-524．

[4]　BOWYER A．Computing Dirichlet tessellations[J]．The Computer Journal，1981，24(2)：162-166．

[5]　LAWSON C L．Generation of a Triangular Grid with Applitation to Contour Plotting，Technical Memo[R]．Jet Propultion Laboratory，Pasadina，Califormia，1972．

[6]　WU W Z，RUI Y K，SU F Z，et al．Novel parallel algorithm for constructing Delaunay triangulation based on a twofold-divide-and-conquer scheme[J]．GIScience and Remote Sensing，2014，51(5)：537-554．

[7]　YANG S W，CHOI Y，JUNG C K．A divide-and-conquer Delaunay triangulation algorithm with a vertex array and flip operations in two-dimensional space[J]．International Journal of Precision Engineering and Manufacturing，2011，12(3)：435-442．

[8]　ANDRÉ L M，CAMACHO，GUIMARÃES，S C，et al．A functional language to implement the divide-and-conquer Delaunay triangulation algorithm[J]．Applied Mathematics and Computation，2005，168(1)：178-191．

[9]　黄清华．一种基于逐点插入 Delaunay 三角剖分生成 Voronoi 图的算法[J]．微型电脑应用，2014，30(6)：43-45．