

基于改进 Delaunay 三角剖分的水下地形三维重建算法



陈士杰 张森林 刘妹琴 郑荣濠

浙江大学电气工程学院 杭州 310000

(21710105@zju.edu.cn)

摘要 在对水下地形进行三维表面重建时,常用的方法是将点云数据投影到二维平面,用 Delaunay 三角剖分算法生成三角网格,然后结合水深高程值还原到三维空间中。但是此方法效率较低,同时在投影时舍去了水深高程值信息,在三维空间内易生成狭长三角形,不利于地形地貌的三维展示效果。因此在采用逐点插入法的基础上,对其中的插入点定位和局部优化过程分别进行了改进,提出了一种融合定位算法,计算三角形矢量面积后,找到搜索前进方向并进行定位,保证了定位路径的唯一性且提高了效率;同时在局部优化过程中引入了水深高程值,计算三维空间内三角形的角度标准差,并将其作为与正三角形相似程度的衡量标准,替换空外接圆准则,使得三维空间内的网格更加均匀化。实验结果表明,该方法在水下地形三维重建的模型质量和构建效率上均优于传统的 Delaunay 三角剖分算法。

关键词: 水下地形;Delaunay 三角剖分;最小标准差;逐点插入法;局部优化

中图法分类号 P229;TP391

Underwater Terrain Three-dimensional Reconstruction Algorithm Based on Improved Delaunay Triangulation

CHEN Shi-jie, ZHANG Sen-lin, LIU Mei-qin and ZHENG Rong-hao

College of Electrical Engineering, Zhejiang University, Hangzhou 310000, China

Abstract In surface reconstruction of underwater terrain, the commonly used method is to project point cloud data to two-dimensional plane, and use Delaunay triangulation algorithm to generate triangular grid, then combine the depth elevation value in order to restore to three-dimensional space. However, the efficiency of this method is low, and the depth elevation value is abandoned during projection, so it's easy to generate long and narrow triangles in three-dimensional space, which is not conducive to the display effect. Based on the point by point insertion method, the location of the insertion point and the local optimization procedure are improved respectively. The specific performance is as follows: a fusion location algorithm is proposed to find the search direction and locate after calculating the triangle vector area, so as to ensure the uniqueness of the location path and improve the efficiency. At the same time, the depth elevation value is introduced to the local optimization procedure. The standard deviation of triangle angle in three-dimensional space is calculated and used as the standard to measure the similarity to normal triangle. The criterion of empty circumscribed circle is replaced to make the grid more uniform in three-dimensional space. The experimental results show that this method is superior to the traditional Delaunay triangulation algorithm in terms of model quality and construction efficiency.

Keywords Underwater terrain, Delaunay triangulation, minimum standard deviation, point by point insertion, Local optimization

1 引言

近年来,声学技术的迅速发展为获取大面积、高精度的海洋数据信息奠定了基础,当前大多数海底地形仍处于未勘测状态^[1],其三维可视化对于海洋测绘、资源开发等具有重要的研究价值和意义^[2]。数字高程模型(Digital Elevation Model, DEM)是对地形表面的数字描述,带有空间位置和高程信

息^[3],主要有规则网格(Regular Square Grid, RSG)和不规则三角形网格(Triangulated Irregular Network, TIN)两种表现形式^[4]。与RSG相比,虽然TIN的数据结构复杂且存储空间大,但是它由互不交叉重叠的一系列三角形连接而成,能够更加准确地表现山脊、山谷等地形陡峭区域的细节特征,因此在地形重建和三维可视化领域得到了广泛的运用。

由于地形数据点集的非封闭性,在构建TIN时通常倾向

到稿日期:2019-11-07 返修日期:2020-04-01 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(U1709203, U1809212);浙江省科技计划项目(2018C03030)

This work was supported by the National Natural Science Foundation of China(U1709203, U1809212) and Science and Technology Program of Zhejiang Province, China (2018C03030).

通信作者:张森林(slzhang@zju.edu.cn)

于曲面而非整体形态的重建,经典的方法有平面投影法和直接剖分法^[5]。其中平面投影法先将三维地形数据投影到二维平面后进行三角剖分,再结合第三维信息还原到三维空间中,相比直接剖分法,其计算简单、效率更高。在此过程中,TIN对三角形几何形状的要求是三边之和最小,每个三角形由最近的相邻点构成且尽量接近于正三角形^[6]。

TIN中的每个三角形描绘了局部的地形变化状态,当数据点个数越多时,TIN中的三角形越密集,对地形特征和细节的展示效果越好。但是,水下地形测深数据主要通过安装在测量船上的声呐进行地毯式勘测得到,相比陆地地形采用激光扫描仪获取海量离散点数据,其密度较为稀疏^[7]。同时海洋勘测的成本高、周期长,无法通过重复测量的方式增加测深点。在此约束下,若三角剖分算法不当,会导致生成狭长三角形,或者非最近的相邻点构成三角形,导致TIN无法反映实际的水下地形特征。

针对上述问题,本文提出了一种改进的水下地形三维重建算法,对逐点插入法中点在三角形格网中的定位速度和局部优化过程分别进行了改进,在减少三维空间内狭长三角形的同时提高了三角形格网的构建效率,并通过实验加以证明。

2 Delaunay 三角剖分和逐点插入法

Delaunay 三角形格网是 TIN 的经典表现形式之一,其二维平面内的离散数据点连接成互不交叉的三角形格网。它的两条重要准则是空外接圆准则和最大最小角准则^[8],特点是在三角形格网中任意一个三角形的外接圆不包含点集中的任意点,以及两个相邻三角形构成的凸四边形在交换对角线后其最小内角不再增大^[4]。因此,生成的单个三角形和正三角形的相似程度更高,最大限度地避免了狭长三角形的出现。用 Lawson^[9]提出的逐点插入法构建 Delaunay 三角形格网,完成对水下地形的三维重建,具体步骤为:

(1)将水下地形点云数据(三维坐标)投影到 xoy 平面,找到二维平面点集中 x 轴和 y 轴正负 4 个方向的极值点 x_{\max} , x_{\min} , y_{\min} 和 y_{\max} ,并在此基础上添加正偏移值 $offset$ ($offset > 0$),构建最小矩形包围盒,横纵坐标的极值分别为:

$$X_{\max} = x_{\max} + offset, X_{\min} = x_{\min} - offset \quad (1)$$

$$Y_{\max} = y_{\max} + offset, Y_{\min} = y_{\min} - offset$$

即矩形包围盒 4 个顶点的坐标分别为 (X_{\max}, Y_{\max}) , (X_{\max}, Y_{\min}) , (X_{\min}, Y_{\max}) 和 (X_{\min}, Y_{\min}) ,之后连接任意一条对角线,构建初始的三角形格网。

(2)将点集中的所有点依次插入到当前的三角形格网中,每次插入时需要遍历所有三角形,确定该插入点位于哪个三角形中。

(3)在成功定位插入点后,寻找其所在三角形附近所有外接圆包含该插入点的三角形并构成影响域凸包,删除影响域内所有三角形之间的公共边,最后连接插入点和影响域构成的多边形的所有顶点^[8],完成该插入点的局部优化过程。

迭代执行步骤(2)和步骤(3),在每次插入点后完成局部优化并更新三角形格网,直到所有点插入完毕,生成最终的三角形格网。最后删除包含任意一个矩形包围盒顶点的三角形,并在此基础上添加水深高程值^[6],如图 1 所示,完成水下

地形三维表面的重建。

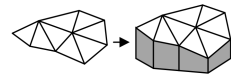


图 1 平面投影法的三维还原

Fig. 1 Three-dimensional reduction of plane projection method

3 算法改进

虽然在平面投影后逐点插入的方法相比直接的三维重建效率更高,但是存在两个问题:1)在定位插入点时需要遍历所有三角形,严重制约了三角网的生成速率;2)在局部优化过程中未考虑水深高程值,在地形陡峭处可能生成三维空间内的狭长三角形^[7]。下面分别从这两个方面进行改进。

3.1 融合定位算法

判断二维平面内的点是否在三角形内的常见做法是连接点和三角形的 3 个顶点,形成 3 个新的小三角形,检测其面积之和是否等于原三角形。但此过程涉及较多的三角函数、开方、平方等运算。三角形矢量面积法是一种较为有效的判断点和三角形位置的方法^[10]。图 2 中 $\triangle ABC$ 3 个顶点的坐标分别为 $A(x_A, y_A)$, $B(x_B, y_B)$, $C(x_C, y_C)$,插入点 P 的坐标为 $P(x_P, y_P)$ 。顶点和边均以逆时针的方向存储。

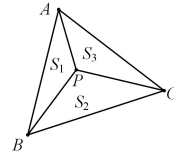


图 2 矢量面积

Fig. 2 Vector area

点 P 与三条有向边构成的矢量面积定义为:

$$\begin{cases} S_1 = \overrightarrow{AP} \times \overrightarrow{AB} \\ S_2 = \overrightarrow{BP} \times \overrightarrow{BC} \\ S_3 = \overrightarrow{CP} \times \overrightarrow{CA} \end{cases} \quad (2)$$

通过计算叉积可判断点与有向边的关系。以式(2)中插入点 P 与有向边 AB 所构成的 $\triangle ABP$ 的矢量面积 S_1 为例,若 S_1 为负,则点 P 在 \overrightarrow{AB} 的左边;若 S_1 为正,则点 P 在 \overrightarrow{AB} 的右边;若 S_1 为 0,则点 P 在 \overrightarrow{AB} 所在的直线或延长线上^[10]。

因此,可通过计算矢量面积的正负个数得到点与三角形的几种位置关系:1) S_1, S_2 和 S_3 均小于 0,则点 P 在 $\triangle ABC$ 的内部;2) S_1, S_2 和 S_3 中有一个大于 0、一个小于 0 且一个等于 0,则点 P 在 $\triangle ABC$ 的某一边的延长线上;3) S_1, S_2 和 S_3 中有两个小于 0 且另一个等于 0,则点 P 在 $\triangle ABC$ 的某一边上;4) S_1, S_2 和 S_3 中有两个等于 0,则点 P 与 $\triangle ABC$ 的某个顶点重合;5) S_1, S_2 和 S_3 中有两个小于 0 且另一个大于 0,或者有两个大于 0 且另一个小于 0,则点 P 在 $\triangle ABC$ 的外部。

为简化后续说明,进行如下定义:若下一搜索三角形为当前三角形某条边右侧的三角形(以逆时针方向存储),则定义该边为当前搜索边。结合图 3 对上述 5 种点和三角形的位置关系进行区域的划分:1)插入点位于三角形内部,即区域 R_1 ;2)插入点位于三角形某一条边的外部或某一条边的延长线上,即区域 R_2 ;3)插入点位于某两条边延长线的交叉区域中,

即区域 R_3 ;4)插入点位于三角形的边上或与某一顶点重合,即区域 R_1 。

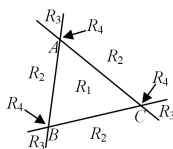


图3 区域分类

Fig. 3 Region classification

虽然矢量面积法简化了判断点与三角形位置关系的运算步骤,但是穷举遍历的效率依然较低。对于加快点的定位速度,很多学者提出了不同方法:Lawson^[9]判断三角形顶点和插入点的位置关系,若每个顶点和插入点均位于该顶点对边的同侧,则定位成功,否则定义该边为当前搜索边,继续搜索,算法的运行效率较低;Chen等^[10]通过类似二叉树的树形结构建立三角形之间的拓扑关系来加快定位速度,但是需要额外的存储空间;Liu等^[11]提出了两种方法,一是利用三角形之间的拓扑关系和矢量面积对插入点进行定位,步骤简单,效率较高,但当矢量面积中有两个为正时,会出现两条搜索路径,因此定位路径不唯一;二是在计算矢量面积的基础上,若插入点不在三角形内,连接插入点和当前三角形的重心,将连线和三角形的相交边作为当前搜索边,但在定位成功前需要计算每一个经过的三角形重心,计算效率较低,同时未考虑到重心、顶点和插入点3点共线的特殊情况,此时无法确定下一搜索三角形;Zou等^[12]对Liu的方法加以改进,只有当矢量面积结果中的两个大于0时才求解重心和插入点的连线与三角形的相交边,减少了重心的计算次数,也保证了定位路径的唯一性,但是未考虑插入点位于三角形边上或边的延长线上等特殊状况,算法可能陷入死循环。针对上述不足,本文提出了一种融合定位算法,具体如算法1所示。

算法1 融合定位算法

输入:待插入点P和当前三角形网格DT(由三角形链表构成)

输出:点P所在的三角形T

1. 初始三角形 T_{start} 定义为 $DT[0]$ 。
2. 计算插入点P和 T_{start} 的3条有向边所构成的矢量面积并统计正负个数,得到点P所在区域。
3. 若点P位于区域 R_1 中,则定位成功并返回当前三角形 T_{start} ,结束本次定位过程。
4. 若点P位于区域 R_2 中,定义矢量面积结果为正的边为当前搜索边,并将其右侧的相邻三角形 T_{next} 作为下一搜索三角形并替代 T_{start} ,返回步骤2。
5. 若点P位于区域 R_3 中,找到矢量面积结果为正的两条边,连接该两边的交点(即三角形顶点)和点P作为搜索前进方向,找出包含该顶点且在搜索前进方向内的三角形 T_{next} ,若其与搜索前进方向的线段不相交则定位成功,返回结果 T_{next} ,结束本次定位过程;若存在相交边,则将其定义为当前搜索边,将其右侧的三角形作为下一搜索三角形并替换 T_{start} ,返回步骤2。
6. 若点P位于区域 R_4 中,则不需要插入该点,结束本次定位过程,返回Null。
7. 循环执行步骤2—步骤6,直到定位成功。

以图4所示的三角形网格中的定位点 P_{12} 为例,初始三角形为 T_1 。采用文献[11]中的第一种方法,需遍历经过三角

形 $T_1, T_2, T_3, T_4, T_5, T_6$ 和 T_7 ,或者三角形 $T_1, T_7, T_8, T_9, T_{10}, T_5, T_6$ 和 T_7 ,搜索路径较长且不唯一;采用文献[11]中的第二种方法,连接三角形的重心和插入点确定搜索方向,遍历经过三角形 $T_1, T_2, T_3, T_4, T_5, T_6$ 和 T_7 ,虽然搜索路径唯一,但是重心的计算次数太多;采用文献[12]中的方法,对于三角形 T_1 和 T_4 ,两条边的矢量面积大于0,计算其重心 G_1 和 G_4 ,并与插入点连接成有向线段 G_1P_{12} 和 G_4P_{12} ,找到与当前三角形相交的有向边 $P_{10}P_1$ 和 $P_{11}P_{13}$,将其右侧的三角形 T_2 和 T_5 作为下一搜索三角形。而对于三角形 T_2, T_3, T_5, T_6 ,只有一条边的矢量面积大于0,直接将该边作为当前的搜索边。最终成功定位 P_{12} 位于 T_7 内,相比前一种方法大大减少了重心的计算次数。

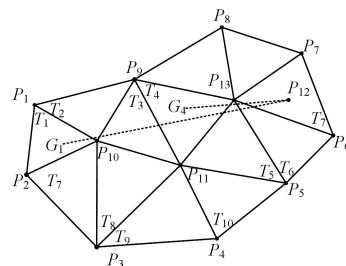


图4 点定位算法比较

Fig. 4 Comparison of point positioning algorithms

而采用融合定位算法,三角形 T_1 两条边的矢量面积大于0,插入点位于区域 R_3 中,连接顶点 P_{10} 和插入点 P_{12} ,找出包含顶点 P_{10} 且在搜索前进方向线段 $P_{10}P_{12}$ 内的三角形 T_3 ,将其与有向线段 $P_{10}P_{12}$ 的相交边 $P_{11}P_9$ 作为当前搜索边,则三角形 T_4 为下一搜索三角形。 T_4 中两条边的矢量面积大于0,找出包含顶点 P_{13} 且在搜索前进方向线段 $P_{13}P_{12}$ 内的三角形 T_7 ,两者没有相交边,因此插入点 P_{12} 在三角形 T_7 内。可以看出,融合定位算法在计算三角形矢量面积的基础上,最大限度地避免了多余的计算步骤,同时考虑到了定位过程中可能出现的所有特殊情况,兼顾了定位路径的唯一性。

3.2 最小标准差准则

使用空外接圆准则对插入点的影响域进行局部优化,当空间三角形和二维平面的夹角较大时,还原到三维空间后可能会出现狭长三角形。

图5中 P_1, P_2, P_3 和 P_4 是三维空间 xyz 内的测深点,使用传统的 Delaunay 三角剖分算法需要将其投影到二维平面 xoy 内,对应的投影点分别为 P_5, P_6, P_7 和 P_8 。假设 $\triangle P_5P_6P_8$ 的外接圆包含点 $P_7, \triangle P_6P_7P_8$ 的外接圆也包含点 P_5 ,根据空外接圆准则需要交换对角线。但三维空间内的 $\triangle P_1P_2P_4$ 和 $\triangle P_2P_3P_4$ 均接近于正三角形,即不需要交换对角线。

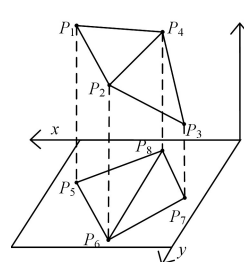


图5 测深点投影

Fig. 5 Projection of sounding points

这是由于在三角剖分过程中未考虑水深高程值,因此用三角形角度的最小标准差准则替换空外接圆准则。假设三维空间内 $\triangle ABC$ 的顶点坐标分别为 $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$ 和 $C(x_C, y_C, z_C)$,求出边长 BC, AC 和 AB ,再由余弦定理分别计算出 $\angle A, \angle B$ 和 $\angle C$ 的大小,则该三角形的角度标准差为:

$$\sigma = \sqrt{\frac{(\angle A - 60^\circ)^2 + (\angle B - 60^\circ)^2 + (\angle C - 60^\circ)^2}{3}} \quad (3)$$

由式(3)可知,角度标准差越小,该三维空间内的三角形与正三角形越相似。最小标准差准则的定义为:在二维平面内构成凸四边形的两个相邻三角形,其角度标准差之和为 q_1 ,交换凸四边形的对角线后,新的两个相邻三角形的角度标准差之和为 q_2 。若 $q_1 > q_2$,则需要交换对角线,反之不需要交换。具体步骤如算法2所示。

算法2 局部优化过程

输入:插入点 P 及其所在三角形 T_a ,以及当前三角形格网 DT (由三角形链表构成)

输出:构成影响域凸包的三角形链表 L

1. 将 T_a 添加到影响域链表 L 中。 T_a 的相邻三角形分别为 T_{n_1}, T_{n_2} 和 T_{n_3} ,将它们添加到相邻三角形链表 N 中,连接 P 和 T_a 的3个顶点,形成 T_{a_1}, T_{a_2} 和 T_{a_3} ,并将它们添加到新三角形链表 A 中。
2. 当前 L 的长度为 l_1 ,设 T_{n_i} 和 T_{a_j} 分别是 N 和 A 中的三角形,循环执行以下操作。
3. 判断 T_{n_i} 和 T_{a_j} 能否在二维平面上构成凸四边形。若不能,则进入下一循环,反之进入步骤4。
4. 若 T_{n_i} 和 T_{a_j} 构成的凸四边形不需要交换对角线,则从 N 中删除 T_{n_i} 并进入下一循环;若需要交换对角线,则将新生成的三角形 T_{ax} 和 T_{ay} 添加到 A 中,并从 A 中删除 T_{a_j} ,将 T_{n_i} 添加到 L 中,同时将 T_{n_i} 除了 T_a 之外的两个相邻三角形添加到 N 中。
5. 对 N 和 A 中的每一个三角形均执行步骤2—步骤4。每次循环结束后,计算此时 L 的长度 l_2 。若 $l_1 = l_2$,则不再增加三角形,结束循环并返回影响域链表 L ;反之回到步骤2,继续遍历 N 和 A 。

下文以图6所示的三角形格网中的插入点 P_{11} 为例,说明运用最小标准差准则的局部优化过程。首先定位到其所在的 $\triangle P_5 P_{13} P_{10}$,分别连接3个顶点,形成3个新的三角形并找到能够在二维平面上与其构成凸四边形的相邻三角形,即 $\triangle P_{10} P_{11} P_{13}$ 和 $\triangle P_9 P_{10} P_{13}$, $\triangle P_{10} P_5 P_{11}$ 和 $\triangle P_{10} P_4 P_5$, $\triangle P_{11} P_5 P_{13}$ 和 $\triangle P_5 P_6 P_{13}$ 这3对,并计算它们的角度标准差。

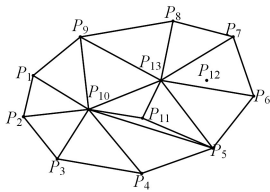


图6 点 P_{11} 的定位

Fig. 6 Positioning of point P_{11}

图7中用最小标准差准则进行检测,假设检测结果为凸四边形 $P_{10} P_{11} P_{13} P_9$ 和 $P_4 P_5 P_{11} P_{10}$ 不需要交换对角线,而凸四边形 $P_5 P_6 P_{13} P_{11}$ 需要交换对角线,因此将 $\triangle P_5 P_6 P_{13}$ 添加到影响域链表 L 中。同时 $\triangle P_5 P_6 P_{13}$ 的其余相邻三角形和新生成的 $\triangle P_{11} P_6 P_{13}$ 和 $\triangle P_5 P_6 P_{11}$ 均满足最小标准差准则,即

不需要交换对角线,因此 L 不再增加,至此插入点 P_{11} 的影响域确定完毕,删除影响域内所有三角形之间的公共边,连接插入点和影响域构成的凸多边形的所有顶点,完成点 P_{11} 的局部优化。

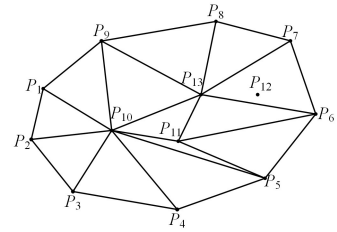


图7 局部优化过程

Fig. 7 Local optimization procedure

4 实验与结果分析

实验所用数据为某水下样区的船载多波束测深数据,共有4796个数据点。进行预处理后将数据格式统一为ASCII XYZ并进行三维显示。如图8所示,数据分布较为均匀,但部分区域地形起伏较大。

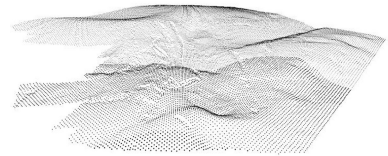


图8 水下样区地形数据

Fig. 8 Topographic data of underwater sample area

4.1 评价因子

为增强论文的严谨性,构建评价因子来定量地评价三维重建效果。用每插入一个点平均需要搜索的三角形个数 n 和总网格时间 t 来衡量算法的效率,用网格质量系数 C 衡量三角形格网的均匀程度。

对于三维空间内的三角形, R 和 r 分别为该三角形外接圆和内切圆的半径,定义圆径比 c 为:

$$c = \frac{R}{2r} \quad (4)$$

从式(4)可以看出,正三角形为最优情况,其圆径比为1。即对于单个三角形而言,圆径比越接近1,该三角形就越接近正三角形。

假设三角形格网中共有 N 个三角形,可进一步衍生出网格质量系数为:

$$C = \frac{1}{N} \sum_{i=1}^N c_i \quad (5)$$

其中, c_i 表示第 i 个三角形的圆径比。对于三角形格网而言,网格质量系数越接近于1,网格的质量和均匀性就更好。

4.2 结果对比

将改进算法和传统逐点插入法以及文献[12]中的算法进行了对比。从表1中可以看出,相比另外两种算法,改进算法的网格质量系数、圆径比最小值及其标准差大大减小,即三维空间内狭长三角形的个数减少,在地形陡峭区域的重建效果更好。同时平均每插入一个点需要搜索的三角形个数减少,

改进后的算法具有更高的点定位和构网效率。

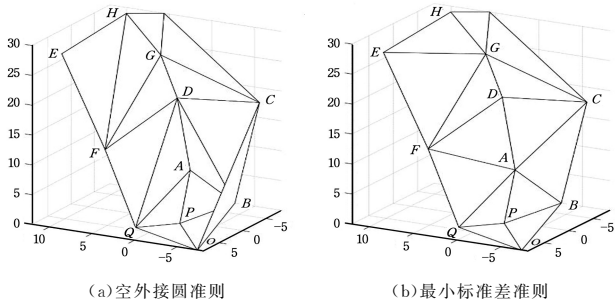


图 9 重建效果对比

Fig. 9 Reconstruction comparison

表 1 质量对比表

Table 1 Quality comparison

Factor	Lawson	Ref 12	Our
C	3.775	3.775	1.09
c_{min}	1.00021	1.00021	1.00013
σ_c	84.5471	84.5471	1.02070
n	4788.54	89.91	51.87
t	489.49	13.57	6.24

同时选取部分区域进行重建效果的对比,如图 9 所示。图 9(a) 基于空外接圆准则, $\triangle ABD$, $\triangle BCD$, $\triangle EFH$, $\triangle FGH$, $\triangle FQD$ 和 $\triangle QAD$ 虽然在二维平面中的三角剖分结果接近于正三角形,但在三维空间内却是狭长三角形,同时 $\triangle OBC$ 也不符合地形变化的特点。图 9(b) 基于最小标准差准则,由于在二维平面进行三角剖分时考虑了水深高程值,因此上述各三角形组成的凸四边形交换了对角线,在三维空间内接近于正三角形。此外,由于 $\triangle OBC$ 和其相邻的 $\triangle OBP$, $\triangle BCD$ 在二维平面内无法构成凸四边形,因此 $\triangle OBC$ 不作为水下地形三维表面的一部分。综上所述,改进算法的三维重建效果能够更好地反映地形地貌的真实形态。

结束语 针对如何更好地对水下地形测深数据点集进行三角剖分从而完成地形表面的三维重建,本文基于 Delaunay 三角剖分算法和逐点插入法,改进了点定位和局部优化过程,在提高插入点定位速度和构网效率的同时,三维空间内不规则三角形格网的几何形状更接近于正三角形。通过实验证明了本文方法的有效性,增强了水下地形三维模型的展示效果。后续可对水下地形的纹理和底质映射展开进一步的研究。

参考文献

- [1] FAN M, SUN Y, XING Z, et al. Seafloor high precision terrain reconstruction based on multi-source water depth data fusion [J]. Journal of Oceanography, 2017, 39(1): 130-137.
- [2] ZHAI J S, ZHANG C, LI Z X, et al. The representation and calculation of the complexity of submarine topography [J]. Journal

of China Ocean University (Natural Science Edition), 2019, 49(S1): 143-147.

- [3] HU J X, PAN N, MA Z T, et al. Research on the algorithm of constructing Delaunay triangulation digital terrain model efficiently [J]. Journal of Peking University (Natural Science Edition), 2003, 39(5): 736-741.
- [4] XU J Z, MA L Z. An improved algorithm for incremental insertion of Delaunay triangulation [J]. Computer Engineering, 2008 (17): 254-256.
- [5] KONG D W. Triangulation and 3D reconstruction of point cloud data [J]. Journal of Southwest Normal University (Natural Science Edition), 2019, 44(7): 87-92.
- [6] JIA J H, HUANG M, LIU X L. Surface reconstruction algorithm based on three-dimensional Delaney triangulation [J]. Journal of Surveying and Mapping, 2018, 47(2): 281-290.
- [7] YUAN Q L, WU X Q. 3D surface reconstruction algorithm combining Delaunay triangulation method and search ball strategy [J]. Journal of Graphics, 2018, 39(2): 278-286.
- [8] YU J, LV P, ZHENG C W, et al. Comparative study on the construction methods of Delaunay triangulation network [J]. Chinese Journal of Image Graphics, 2010, 15(8): 1158-1167.
- [9] LAWSON C L. Generation of a triangular grid with application to contour plotting [M]. California: Jet Propulsion Laboratory, Pasadena, 1972.
- [10] CHEN M J, FANG Y M, LI G Z, et al. An improved generation algorithm of Delaunay triangulation network [J]. Journal of Kunming University of Science and Technology (Natural Science Edition), 2016(5): 33-38.
- [11] LIU S H, WU D S, LUO X L, et al. Research on fast positioning algorithm of point target in Delaunay triangle network [J]. Surveying and Mapping Science, 2007, 32(2): 69-70.
- [12] ZOU Y G, ZHANG T. Improved Delaunay triangulation algorithm in plane domain [J]. Computer Engineering and Application, 2013, 49(20): 171-174.



CHEN Shi-jie, born in 1995, postgraduate. His main research interests include underwater information processing and 3D visualization.



ZHANG Sen-lin, born in 1964, postgraduate, professor, Ph.D supervisor. His main research interests include ocean information technology, artificial intelligence, image processing and pattern recognition.