

# Geomagic 专利阅读笔记

杨成林

在谷歌专利搜索中以 Geomagic 为专利权人的专利共有 5 项，本文主要关注其中三项：6377865(Methods of generating three-dimensional digital models of objects by wrapping point cloud data points, 以下简称专利 1)、6996505(Methods, apparatus and computer program products for automatically generating NURBS models of triangulated surfaces using homeomorphisms, 以下简称专利 2)、7023432(Methods, apparatus and computer program products that reconstruct surfaces from data point sets, 以下简称专利 3)。此外还有两项专利分别关于几何变形和带有颜色的模型处理。

## 一、邻域定义及搜索结构

在点云处理中需要点云的邻域信息。在 Geomagic 中，点  $p_i$  的邻域定义为处在以  $p_i$  为中心、 $2r_0$  为边长的正方体中的点(包括  $p_i$  本身)。在图 1，实心点即为点  $p_i$  的邻域点。使用正方体而不是球来定义邻域，可能是基于计算速度的考虑。 $r_0$  的选取以使邻域点的数量接近某个给定值为目标。专利 3 上给出的三个典型的邻域点数量为 25、50、100(参考[1], P114)。在实现中，首先以从点云中做若干次随机采样以确定合适的  $r_0$ 。

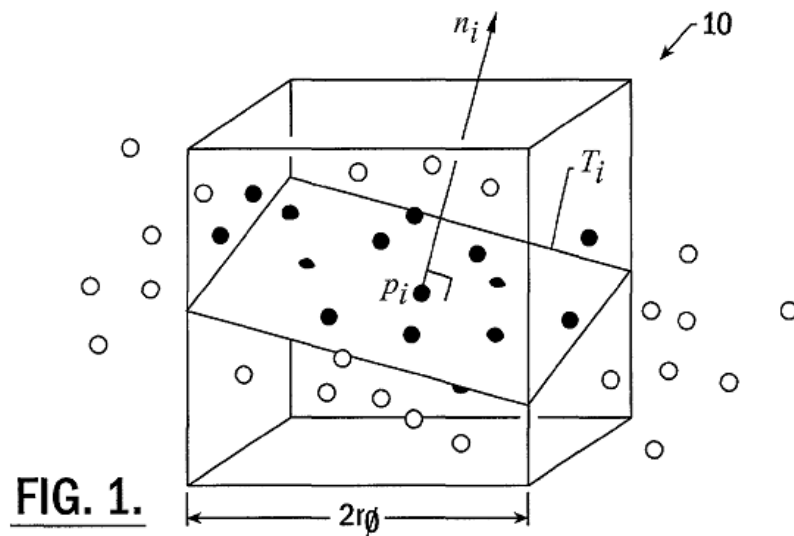


图 1: 邻域定义(专利 3)

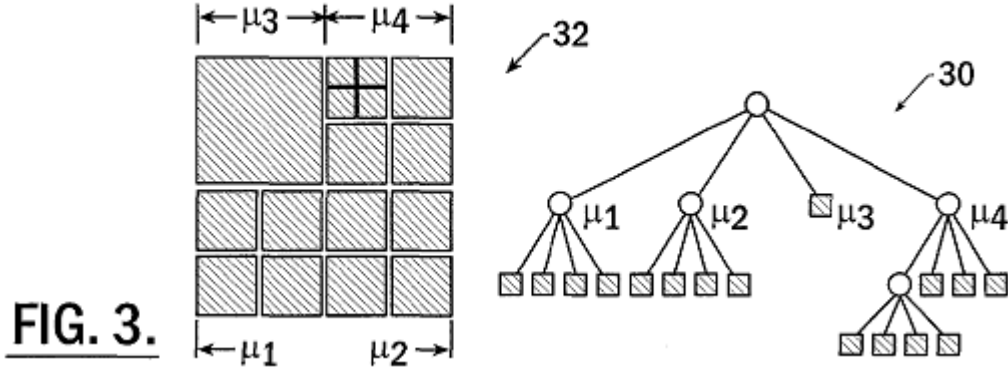


图 2: 八叉树(专利 3)

为了加快邻域搜索，通常建立一定的搜索结构。在 Geomagic 中采用的是八叉树，如图 2(在二维以四叉树示意)。在专利 3 中有八叉树的建立、遍历等较详细的介绍。且八叉树用于空间搜索是很早出现、非常成熟的技术，这里就不再赘述。本人推测，为了提高效率，八叉树的建立甚至邻域点搜索是在点云导入时完成。

## 二、 点云预处理

扫描得到的点云通常存在多种缺陷，需要一定的预处理以满足后续步骤需要。在 Geomagic 的专利中没有找到关于点云对齐方法的描述，不过本人推测应为 ICP 方法的某种变体，而且其中用到了点云的法矢信息——这点可以由 unshaded 点云在对齐后变为 shaded 看出。关于点云对齐，本人推荐 Szymon Rusinkiewicz 的主页\*。下面介绍点的去噪以及异常点(outlier)的判定。

### 1. 点云局部曲面拟合

利用第一节所述的方法可以得到点的邻域。由邻域点集可以计算拟合平面(图 1 中  $T_i$  为  $p_i$  邻域点集的拟合平面),进而得到点云的法矢(图 1 中  $n_i$  为  $p_i$  的法矢)。具体是采用协方差分析方法。点  $p_i$  的邻域点集记为  $S_i$ ，其质心为

$$\bar{p}_i = \frac{1}{|S_i|} \sum_{j \in S_i} p_j$$

拟合平面应通过质心。矢量表示为列矢量，定义协方差矩阵

$$C = \begin{bmatrix} p_{j_1} - \bar{p}_i & \cdots & p_{j_n} - \bar{p}_i \end{bmatrix} \begin{bmatrix} p_{j_1} - \bar{p}_i & \cdots & p_{j_n} - \bar{p}_i \end{bmatrix}^T, j_k \in S_i$$

在专利 3 中称此矩阵为质量分布矩阵(Mass Distribution Matrix, MDM)。此矩阵为对称、半正定矩阵，其对应于最小特征值的特征向量即为拟合平面的法向量。

在得到点云的法矢后，可以进一步构造邻域点集的法矢方差矩阵

$$C' = \sum_{j \in S_i} n_j \cdot n_j^T$$

在专利 3 中称此矩阵为法矢分布矩阵(Normal Distribution Matrix, NDM)。此矩阵同样是对称、半正定矩阵。其三个非负特征值记为  $\nu_1 \geq \nu_2 \geq \nu_3$ ，归一化为  $t_k = \nu_k / (\nu_1 + \nu_2 + \nu_3)$ ，对  $k=1,2,3$ 。两个较小的特征值对应两个主曲率，相应的特征向量则对应主曲率方向。

\*<http://www.cs.princeton.edu/~smr/>

根据  $t_2, t_3$  的大小，将邻域分为三种类型：平面(plane-like)、边(edge-like)、角(corner-like)。判定关系如下：

平面：  $\epsilon_0 \geq t_2 \geq t_3$

边：  $\epsilon_0 \geq t_3, \epsilon_1 \leq t_2$

角：  $\epsilon_1 \leq t_3 \leq t_2$

$\epsilon_0, \epsilon_1$  是设定的门限值，专利 3 给出的值分别为 0.0075 和 0.015。

把邻域绝对地分为三种类型，这并不太符合实际的情况。为此，可以采用弱分类的方法，定义一种权重系数来表征其与某种类型的符合程度。如，一个邻域可以是 0.7 的平面类型又是 0.3 的边类型。专利 3 中对此有详细的说明。

下一步，依据邻域类型，将邻域拟合成三种不同的曲面：平面邻域拟合为平面，边邻域拟合为以圆锥曲线为截面的柱面，角邻域则拟合为一般二次曲面。平面拟合在前面已经得到。下面介绍另外两种曲面的拟合方法。

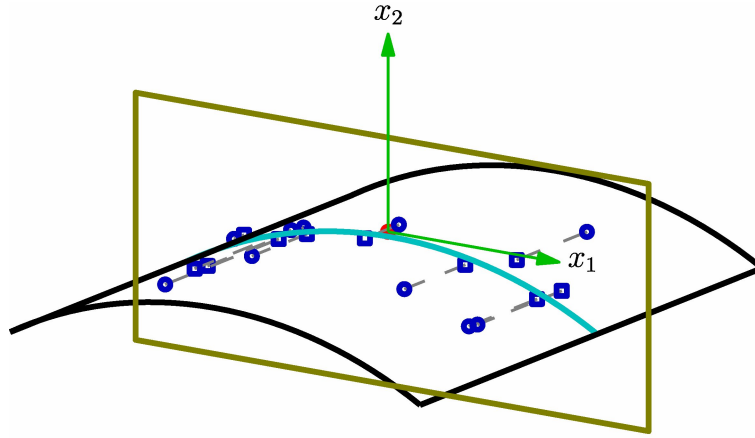


图 3: 柱面拟合

对于边类型的邻域，其较小主曲率方向可以作为柱面的母线方向。这也就是法矢方差矩阵的最小特征值对应的特征方向。将邻域点集投影到法矢方差矩阵的另两个特征方向定义的平面上，再将投影得到的平面点集拟合为一条圆锥曲线。为了后续处理的方便，可以把点  $p_i$  的投影点作为原点。平面点的坐标记为  $x_1, x_2$ ，一般的圆锥曲线方程为

$$a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6 = 0$$

易看出，各项系数同乘以某非 0 常数并不改变方程。记  $\mathbf{x} = [x_1^2, x_1 x_2, x_2^2, x_1, x_2, 1]^T$ ， $\mathbf{a} = [a_1, a_2, \dots, a_6]^T$ ，则拟合目标为

$$\min_{\mathbf{a}} \sum_{j \in S_i} (\mathbf{a}^T \mathbf{x}_j)^2$$

上式中求和的部分可化为  $\mathbf{a}^T (\mathbf{x}_j \mathbf{x}_j^T) \mathbf{a}$ 。记  $\mathbf{X} = \sum_{j \in S_i} \mathbf{x}_j \mathbf{x}_j^T$ ，这是对称、半正定矩阵，则拟合目标的解即  $\mathbf{X}$  的最小特征值对应的特征向量。

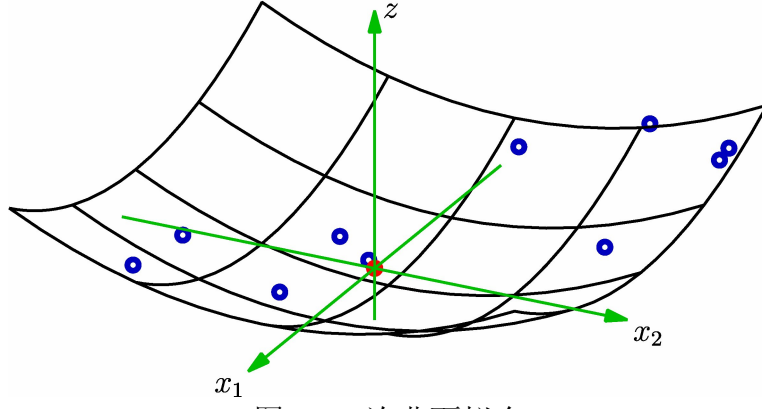


图 4: 二次曲面拟合

相较前面两种情况，角类型的邻域是最一般的情形。将邻域点集转换到质量分布矩阵的三个特征向量组成的坐标系中。同样的，这里把点  $p_i$  作为原点。最大特征值对应的坐标记为  $z$ ，其余两个记为  $x_1, x_2$ ，则曲面方程为

$$z = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6$$

依然采用前面的记法，拟合目标可写为

$$\min_{\mathbf{a}} \sum_{j \in S_i} (\mathbf{a}^T \mathbf{x}_j - z_j)^2$$

也即求线性方程  $\begin{bmatrix} \mathbf{x}_{j_1}^T \\ \dots \\ \mathbf{x}_{j_n}^T \end{bmatrix} \cdot \mathbf{a} = \begin{bmatrix} z_{j_1} \\ \dots \\ z_{j_n} \end{bmatrix}$ ， $j_k \in S_i$  的最小二乘解。这一问题用基本的矩阵方法就可

解决。

## 2. 点云去噪

前面将邻域分为三种类型并拟合为三种不同的曲面是为了后续的点云处理。在 Geomagic 中，点云去噪是通过将点投影到拟合曲面上实现的。在平面的情形，计算较简单：

$$\mathbf{p}_i' = \mathbf{p}_i - ((\mathbf{p}_i - \bar{\mathbf{p}})^T \mathbf{n}_i) \mathbf{n}_i$$

对于边类型的邻域，点  $p_i$  的投影垂直于柱面的母线方向，只需计算另两个方向上的移动。将一个点投影到圆锥曲线上，精确的计算较为困难，而且也不必要。将曲面方程中二函数记为  $f(x_1, x_2)$ ，则  $\nabla f = \begin{bmatrix} 2a_1 x_1 + a_2 x_2 + a_4 \\ a_2 x_1 + 2a_3 x_2 + a_5 \end{bmatrix}$ 。为简单起见，将该函数在  $p_i$  的对应点梯度方向作为投影方向，由于在前面的计算中该点被设为原点，投影点的坐标可以写为  $\hat{\mathbf{p}}_i = t[a_4, a_5]^T$ ，又投影点应在圆锥曲线上，得到方程：

$$(a_1 a_4^2 + a_2 a_4 a_5 + a_3 a_5^2) t^2 + (a_4^2 + a_5^2) t + a_6 = 0$$

求解方程，得到两个根，对应平面上的两个点，取和原点较近的那个解。再将这个点转换到三维空间中，便得到最终结果。

角类型的邻域对应的拟合曲面方程更为复杂。为此，将投影方向做简化，取拟合曲面时定义的  $z$  坐标方向做为投影方向，余下的计算就很简单了。

在弱分类的情况，邻域被拟合为若干种曲面类型并分别计算投影点，再将多个投影点按照前面定义的权重系数混合得到最终的结果。专利 3 中还提到某些情况下应使用三次曲面来

进行拟合，而有时使用一些特殊的曲面，如圆柱、圆锥，可以带来较好的结果。

专利中对曲面去噪这一部分的叙述是比较详细的，不过在 Geomagic 中曲面去噪有三个选项：Free-form shapes、Prismatic shapes(conservative)、Prismatic shapes(aggressive)，其对应的算法上的变化还是不清楚。

### 3. 异常点的判定

异常点(outlier)是指远离曲面的点，通常是由于错误或扫描设备的不足而产生。固然可以采用前面去噪中点到投影点的距离作为判定标准，不过 Geomagic 使用另一种思路。如果只考虑点到拟合平面的距离，会使邻域类型为边或角的点更容易被认定为异常点，但是这些点的邻域点到其拟合平面的距离也会比较大。定义平均平方距离：

$$\bar{D}_i = \sum_{j \in S_i} ((\mathbf{p}_j - \bar{\mathbf{p}}_j)^T \mathbf{n}_j)^2$$

异常点的判定不等式为：

$$((\mathbf{p}_i - \bar{\mathbf{p}}_i)^T \mathbf{n}_i)^2 > C \cdot \bar{D}_i$$

$C$  为给定的参数，可以通过调整  $C$  以获得更好的结果。(本人对专利的这一部分理解还不是很清楚，可能有误。)

除了前述的两项，Geomagic 还提供了 Disconnected 的判定。本人推测这一功能大概只是根据点的邻域做了简单的判断。而点云采样部分在专利中没有描述，Pauly 博士论文[1]中一章专门讨论这一问题，不过本人对 Geomagic 中的这部分功能无法做出任何推测。

## 三、 曲面重构

曲面重构是 Geomagic 中的关键技术，Edelsbrunner 正是由这方面的技术开办了 Raindrop 公司。Geomagic 中的曲面重构方法被称作 Wrap，分为 Surface 和 Volume 两种。这两种方法的想法都源自 Boissonnat[2]早期的论文。

### 1. Volume Wrap

Volume Wrap 是较早的方法，Edelsbrunner[3-4]于 1995 年提出了这一方法，但是直到[4]才公布算法。曲面重构的专著[5]有一章介绍此类算法，专利 1 是以此算法为核心，相关的资料还是比较丰富。

对点集做 Delaunay 三角化，可以得到一个单复形(所有数学术语这里不加解释，要了解可以参看前列文献或拓扑学教材)，称作 Delaunay 复形。而要重构的曲面可以表示为此单复形的一个子复形。这一思路是许多基于 Delaunay-Voronoi 曲面重构方法的基础(PowerCrust[6]不是这样)。关键在于怎样得到这个子复形，Volume Wrap 方法是使用距离场的梯度流。

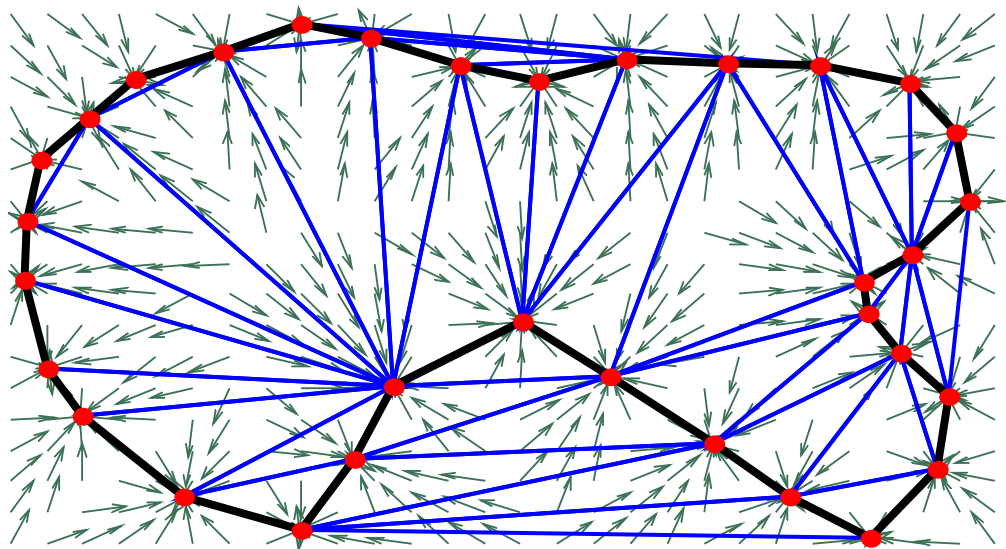


图 5: Volume Wrap 原理示意

这里只做大致地介绍，可能有不精确的地方。对于  $d$  维点集，定义一个标量场：该标量场在空间中任意一点的值为此点到点集中最近点的距离的相反数(之所以取负值是为了使流的方向符合习惯)。由标量场的梯度得到一个向量场，如图 5。根据此向量场，可以将单纯形分为以下三类(图 6)，从左至右依次为 centered, confident, equivocal。

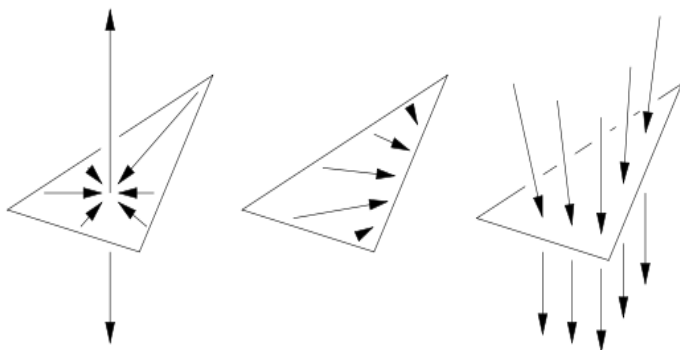


图 6: 单纯形分类

边界上的  $d-1$  维单纯形如果是 equivocal，那么它是 collapsible, 通过 collapse 操作，删去它和包含它的单纯形。反复这一过程，直到不存在 collapsible 面。图 5 中黑色粗线是重构结果。Volume Wrap 还提供了对重构后的结果进行编辑的功能，这部分可参看论文并对照 Geomagic 中的实现。

据本人分析，至少有两种情况 Volume Wrap 不能很好地处理：一是尖锐的凹处，二是中空的结构。对于这些，其他的作者提出了一些修正的方法[5][7]。

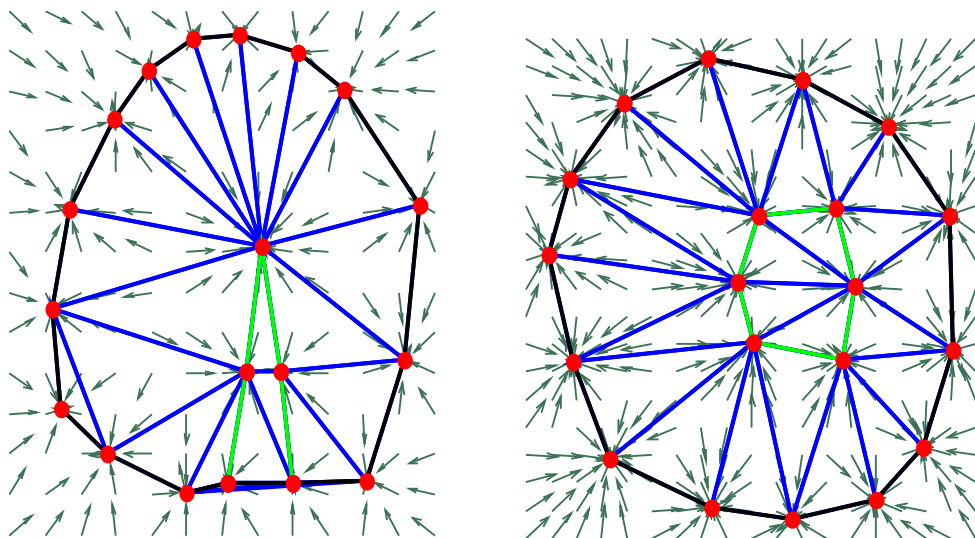


图 7: Volume Wrap 难以处理的情形

## 2.Surface Wrap

Surface Wrap 则是从局部三角化出发，构造全局三角化。构造局部三角化的方法是将邻域点集投影到拟合平面上，将三维问题转为二维，再计算平面正则三角化。正则三角化 (regular triangulation)，又称 weighted Delaunay triangulation，是 Delaunay 三角化的一般化，是对带权值点集的三角化。带权值点，是一般点加上权值，两个权值点的平方距离定义为 (以二维为例)：

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 - w_1 - w_2$$

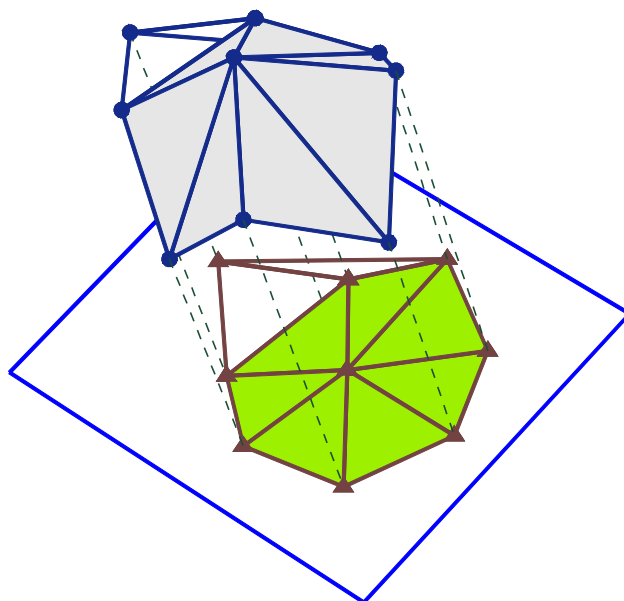


图 8: 局部三角化

所谓 Delaunay 三角化，即其中任意一个三角形的外接圆中均不包含点集中的其它点。将“外接圆”、“包含”等概念用上面的距离定义修改，便得到正则三角化的定义。在专利 3 中，权值取为点集到平面的平方距离的相反数，这一取值方法由 Edelsbrunner 论文[8]中提出的 restricted Delaunay triangulation 得来。在点的邻域点集的局部三角化构造中，实际是要



构造包含该点的“一圈”三角形，称作 star。这对 Delaunay 三角化是不成问题的。而在正则三角化中，并不是所有点集中的点都会成为三角形的顶点，一些点会消失(submerge)。所以，以点  $p_i$  的邻域构造的三角化却可能不包含  $p_i$  的投影点。所以专利 3 中实际使用的是约束正则三角化(constrained regular triangulation)。约束正则三角化的严格定义本人没看懂，有兴趣可以看 Jonathan 的论文[9]，不过粗略地讲，就是强制一些边存在于三角化中。专利 3 约束的是由点  $p_i$  和  $p_i$  最近点的对应点构成的边。在  $p_i$  的投影点是正则三角化的顶点时，这条边也一定是正则三角化的一条边(证明?)。

考虑到实际上只需要一部分三角形，不同于一般的三角化方法，专利 3 中提出了一套专门的算法。问题转化为：已知平面点集，构造其中一点  $q_0$  的约束正则三角化，这里的约束为  $q_0$  和其最近点的连线。把  $q_0$  从平面点集中去除，将剩余点以关于  $q_0$  的逆时针顺序排序。 $q_0$  在 star 中的位置有两种可能：在边界上或不在边界上(图 9)，如果点集在通过  $q_0$  的某条直线的一侧，则为前者，否则为后者。在前面的排序中，如果  $q_0$  在边界上，则最接近划分直线的点为起始，如果不在边界上，则以  $q_0$  的最近点为起始，参看图 9 中的  $q_1$ 。而如果不在边界上，再将  $q_1$  点重复地加到序列的末尾。

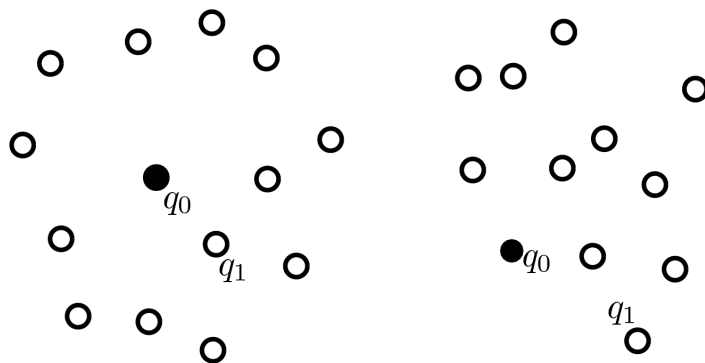


图 9: 平面点集关于  $q_0$  的位置

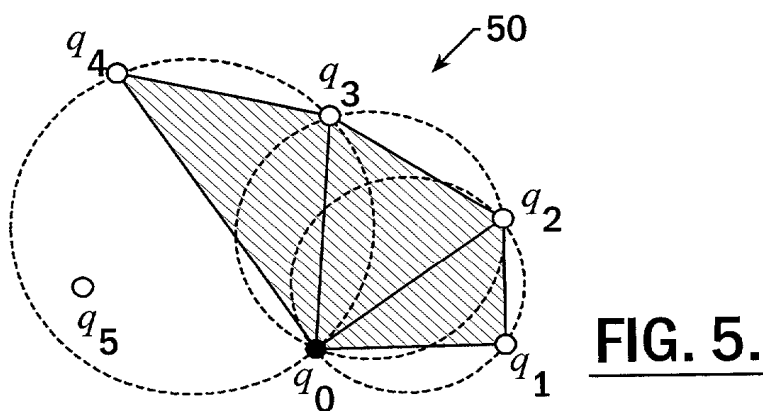


图 10: 三角化算法(专利 3)

得到点集序列后，就按顺序插入点，但在插入点时，要把之前的形成的三角形中外接圆包含当前点的删除。如图 10(这里权值都取为 0)，在插入  $q_5$  时应先删去三角形  $q_0q_3q_4$ 。



具体的方法如下：建立一个堆栈，表示  $\text{star}$  中除  $q_0$  以外的点。先将点集序列中的前两个点压入堆栈。然后遍历点集中其余点，对于每个点：判断此点是否在  $q_0$  与堆栈栈顶两个点组成的三角形外接圆内(根据一个行列式的正负，具体可看文献)，如果是就弹出栈顶元素，并重复这一过程，否则(包括堆栈中元素数量小于 2 的情况)就将此点压入堆栈。最终的堆栈中的点按顺序和点  $q_0$  构成一系列三角形，就是所需的三角化(证明?)。

在得到局部三角化后，下面的工作就是将这些局部三角化整合到一起。这一过程非常复杂，本人未能完全理解，以后希望能补充这一部分。

#### 四、 NURBS 曲面自动生成

待补充。

## 参考文献

- [1] Mark Pauly. Point Primitives for Interactive Modeling and Processing of 3D Geometry. 2003.
- [2] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. ACM Transactions on Graphics (TOG). 1984, 3(4), 266-286.
- [3] H. Edelsbrunner, M. A. Facello, P. Fu, et al. Wrapping 3D scanning data. Proceedings of SPIE. 1998, 3313, 148-158.
- [4] H. Edelsbrunner. Surface reconstruction by wrapping finite sets in space. Discrete and Computational Geometry—The Goodman-Pollack Festschrift. , 379–404.
- [5] T. K. Dey. Curve and Surface Reconstruction: Algorithms with Mathematical Analysis. 2006,
- [6] N. Amenta, S. Choi, T. K. Dey, et al. A simple algorithm for homeomorphic surface reconstruction. Proceedings of the sixteenth annual symposium on Computational geometry. 2000, 213-222.
- [7] E. A. Ramos B. Sadri. Geometric and topological guarantees for the WRAP reconstruction algorithm. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. 2007, 1086-1095.
- [8] H. Edelsbrunner N. R. Shah. Triangulating topological spaces. Proceedings of the tenth annual symposium on Computational geometry. 1994, 285-292.
- [9] J. R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. Proceedings of the nineteenth annual symposium on Computational geometry. 2003, 181-190.