多边形的三角形剖分结构及其应用

罗云千，2019012343，luoyq19@mails.tsinghua.edu.cn

摘要

三角剖分问题在理论研究和实际问题中都有大量应用，因此也产生了众多相关的研究。本文介绍了其中的一些重要的问题和结果。

本文主要选取了三方面的内容。第一方面是三角形剖分的应用，包括在理论上的应用，例如组合几何中的美术馆问题，以及更加实际的应用，例如 3D 图形建模和渲染，揭示了它们和三角形剖分问题的重要联系。第二方面是关于三角形剖分的算法，简要介绍了平面三角形剖分算法从 $O(n^2)$ 算法到 $O(n)$ 算法的发展过程，以及一些优化曲面三角形剖分的一些方法。第三方面是关于三角形剖分在高维情形，特别是三维情形的推广，说明了多面体的四面体剖分的不可行性和判定方法的困难性。

1 引言

在平面几何和组合几何领域，三角形这一图形由于其简单性而在得到了广泛的研究。很多时候，三角形可以用来解决多边形中的问题，这种方法的理论基础是一个广为人知且符合直觉的性质：

性质. *每一个简单（边不交）的多边形区域都可以剖分成若干个三角形区域，使得其中每个三角形的顶点都是原多边形的顶点 [1]。*

证明. 对多边形的边数 n 归纳，当 $n=3$ 时结论显然。若所有边数小于 n $(n>3)$ 的多边形都存在三角剖分，考虑任意一个 n 边形。如果能够证明它有一条对角线位于多边形的内部，那么由于这个对角线把多边形分成了两个边数小于 n 的多边形，由归纳假设，这两个多边形都可以三角剖分，从而原来的 n 边形有三角剖分。进而完成了证明。

为了证明这个多边形有位于内部的对角线，考虑它的所有内角。由于 n 边形的内角和为 $(n-2)\pi < n\pi$，故存在一个小于 π 的内角。设这个内角为 $\angle APC$，其中 A, P, C 都是多边形的顶点。若 $\triangle APC$ 内部没有多边形的顶点，那么 AC 即为多边形的一条内部对角线。若 $\triangle APC$ 内部有其它多边形的顶点，在所有这样的顶点中，选取一个顶点 X，使得 $\angle PAX$ 最小。如果这样的点 X 有多个，那么选取与 P 距离最近的一个。下证 PX 是多边形的内部对角线，若不然，若 PX 有一部分位于多边形外部，那么它一定和多边形的某条边有交点，设这条边为 UV。延长 AX 交边 PC 与点 Q，由于 UV 与 PQ 有交点，但是 UV 与 AP, PQ 没有交点（因为多边形的边不交），故 U, V 两点中必然有一个点在 $\triangle APQ$ 的内部，从而 $\angle APX$ 的最小性矛盾。从而 PX 是一条内部对角线。 □

可以看到，虽然三角剖分的剖分直觉上是显然的，但是其证明却并不平凡。这暗示了三角形剖分并不是一个简单的问题，事实上也确实如此。关于这个问题，人们已经做了大量的研究，产生了许多有意义的结果，下文将介绍其中的一些方面。

2 三角形剖分的应用

三角形剖分作为一个基本的组合几何结构，它的重要之处在于，利用三角剖分，可以把一个比较复杂的图形分解为若干个相对独立的子图形，对每个子图形分别进行处理，然后利用恰当的方法把这些子图形上的结论组合起来，从而解决原来的比较复杂的问题。三角形剖分的这个特性使得它在许多地方得到了应用。

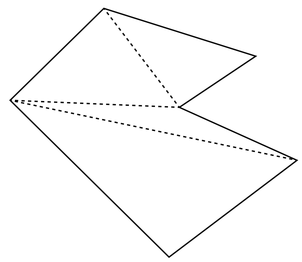

图 1. 一个将多边形剖分成三角形的简单示例。

2.1 美术馆问题

美术馆问题就是三角剖分在理论数学中的一个典型应用。这个问题的原始形式如下：

问题. *若 S 是一个 n 边形，求最小的 m，使得存在一个包含 n 个点的点集 T，对于 S 内部的任何一个点 p，都存在 $q \in T$，并且线段 pq 完全位于 S 内部（含边界，下同）*[2]。

如果我们把 S 看成是一个美术馆，T 看成是美术馆的保安的集合的话，那么这个问题描述的就是：就为了能够让保安看到美术馆的每一个角落，我们需要保安的最小数目。

文 [3] 对此给出了一个漂亮的结论：我们最多需要 $\lfloor \frac{n}{3} \rfloor$ [1] 个点（保安）就可以保证能够看到美术馆的每一个角落了。文 [3] 中的证明用的是对 n 的归纳法，并且对于多边形的形状进行了不少的分类讨论。随后 Fisk 在文 [2] 中给出了只有几行的简短证明。Fisk 的证明主要思想是将多边形剖分成一些三角形，通过对点的数目归纳证明，将多边形的顶点染成三种颜色，使得没有两个相邻（通过多边形或三角形的边相连）的顶点具有相同的颜色（从而每个三角形的三个顶点颜色各不相同）。由于一共有 n 个顶点，所以必然有一种颜色的顶点至多有 $\lfloor \frac{n}{3} \rfloor$ 个，将这些顶点上放上保安。由于每个三角形三个顶点具有不同的颜色，因此必然有一个顶点上有保安，从而这个三角形内的每个点都被保安观察到了（见图2）。进而整个美术馆都在保安们的视野当中。[2]

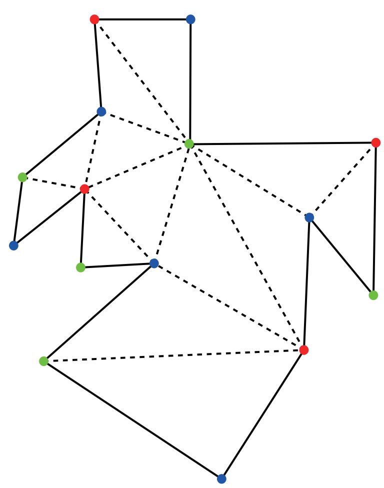

图 2. 将美术馆三角剖分后，把它的顶点染成三种颜色，最后把所有红色的顶点放上保安。

在这个证明中，最重要的想法在于，监视整个美术馆转可以转化为监视每个剖分出的三角形，所以只需要在每个三角形中选取一个顶点即可，考虑到这一点之后，那么后面的证明就是水到渠成的。研究美术馆问题在一定程度上就转化为了对于三角形剖分的研究。美术馆问题有诸多推广，例如让保安监视所以美术馆外

[1] $\lceil x \rceil$ 表示不小于 x 的最小整数。例如 $\lceil 2.4 \rceil = 3$。

[2] 这个结论的漂亮之处还在于它给出了最好的上界，事实上，$\lfloor \frac{n}{3} \rfloor$ 是最优的结果，当我们把美术馆取为一个类似无柄梳子的形状（每个梳齿都是一个三角形）时，少于 $\lfloor \frac{n}{3} \rfloor$ 个保安是无法看到美术馆的每一个角落的 [4, p3]。

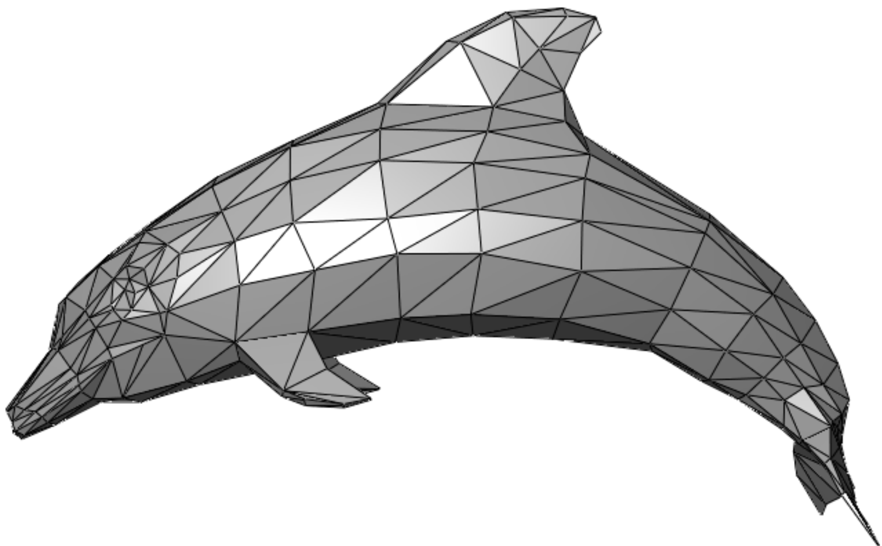

图 3. 将一个海豚的模型表面进行三角剖分并加上光照
https://en.wikipedia.org/wiki/File:Dolphin_triangle_mesh.png

部的区域，考虑墙壁只朝着两个垂直的方向的美术馆（称为正交美术馆）等等。解决这些问题的主要手段都是对于某个平面区域的剖分情况进行研究 [4, p31, p145]。

2.2 图形渲染问题

计算机图形设备和 3D 图形引擎是 3D 动画，游戏，电影特效的核心之一，它使得我们可以将一个三维物体的模型渲染成包含光影，材质，透视关系的图像。对于这些硬件和软件，渲染三角形常常是最底层的接口之一。因此，在很多情况下，三角形是用来对实际物体的表面建模的主要工具 [5]。

尽管很多图形引擎也提供了对于曲面，例如二次曲面，样条曲面的支持，但是三角形还是有很大的优越性。其中最主要的一点是它能保证它的三个点在同一个平面上。而平面和曲面比起来具有很好的性质。首先，计算平面上的光照比起曲面要容易得多：平面每个位置的方向都相同，这意味着对于平行光来说，这个平面上的光照强度是恒定的，即使对于非平行光，当三角形面相对较小时，它的各个位置的光照强度基本上是不变的。但对于曲面来说，即使这个曲面很小，但是由于照射角度问题，它的各个位置的光照强度仍可能会发生很大的变化。此外，由于平面所表示的方程是线性的，所以计算两个三角形是否相交，或者计算一条光线是否穿过这个三角形都是很容易的，但对于曲面来说，一般都只能近似地求解，而且计算要比平面复杂得多。

虽然三角形渲染在计算上具有如此的优越性，但是由于在现实世界中的大部分物体的表面并不是三角形，而是一个曲面。所以为了能够使用渲染三角形的方法渲染这样的曲面，我们需要把这些表面用一些三角形来近似，这个近似的过程被称为曲面的三角剖分 (triangulation)，剖分出的三角形组成的表面称为三角形网格 (triangle mesh)。图3是一个典型的三角剖分的例子。因此研究曲面上的三角剖分是计算机图形学中的重要课题。

3 三角形剖分的算法

三角剖分的广泛应用促使人们寻找更加高效的算法来进行三角剖分，从而衍生出了大量相关的研究。

3.1 平面上的三角剖分算法

对于平面上的三角剖分，主要的研究问题是：

问题. *给定若干个平面上顶点的坐标，考虑由这些点顺次连接所形成的简单多边形，求一些线段，使得这些线段把多边形剖分成若干个三角形。*

显然将一个凸 n 边形剖分成三角形只需要 $O(n)$ 次操作 [1]，因为我们只需要把一个顶点与其它的所有顶点相连即可。对于非凸的多边形，事情就要复杂得多。

对于这样的问题，一个 $O(n^2)$ 的算法是很容易构造出来的，但是令人惊讶的是这样一个宽松的时间复杂度长时间内都没有得到改进，直到 1978 年文 [6] 中才终于给出了一个 $O(n \ln n)$ 的算法。由于它比较简单，因此比较实用。

在这样一个巨大的改进产生之后，许多更好的算法开始出现。10 年之后，一个比较复杂的 $O(n \ln\ln n)$ 的算法在文 [7] 中被提出。不久之后，文 [8] 又表明存在一个复杂度仅有 $O(n \ln^* n)$ 的算法 [2]，并且这个算法相比起上一个的算法要简单得多。

几年之后，Chazelle 发现了一个极其复杂的 $O(n)$ 算法 [9]。虽然这个算法的复杂度已经是理论上的最优，但是由于它的步骤过于复杂，所以除非对于极大的数据，否则一般它并不比前文的算法快，所以这一算法基本上仅有理论意义。

3.2 三角剖分的优化

对于计算机图形学来说，比起平面上的结构，更关心的是在它在逼近曲面上的应用。在这样的情况下，我们的主要目的是利用三角形对曲面进行恰当的逼近。显然，如果我们剖分出更多的三角形，那么就能更好地逼近曲面，渲染出的图形也会更加平滑真实；如果减少三角形的数目，那么我们就可以得到更快的渲染速度，而速度在一些对于实时性要求比较高的场景（例如游戏）中非常重要。如何在质量和速度这两个因素间进行权衡，以及如何在一个因素不变的前提下改进另一个因素得到了充分的研究。

文 [10] 对此提出了一个算法，这个算法被称为 decimation algorithm，主要思路是按照某种判据不断选取比较“平”的顶点，将这个顶点和它所在的所有三角形去掉。去掉这些三角形之后原来的网格会出现一个洞，用恰当的方式把这个洞剖分成一些三角形，然后更新判据，并继续重复上述过程，直到三角形数目达到想要的数目为止。这个算法可以把一个复杂的三角形网格转化成一个三角形数量大大减少的网格，并且使得新的网格和原来的网格大体上保持一致，从而能够很大程度上提高渲染速度。

在三角形剖分的过程中，我们还要关心的是这些三角形的“质量”，也就是这些三角形在大小和形状方面的均匀程度。如果三角形网格不够均匀（例如有一些三角形的面积远大于其它三角形），或者三角形的形状比较极端（例如有很接近于 0 度的内角），都会导致渲染出来的图形不够平滑。为了解决这样的问题，文 [11] 提出了 Delaunay 剖分 (Delaunay triangulation) 的概念：

定义. *一个三角形网格称为是 Delaunay 剖分，如果每个小三角形的外接圆内都不含有其它三角形的顶点。*（见图4）

Delaunay 剖分有很多良好的性质，例如它最大化了这些三角形的最小内角，这一点确保了这种剖分下三角形的形状能够尽可能地均匀。文 [12] 给出了两个算法，当给定顶点时，可以快速地给出包含这些顶点的 Delaunay 剖分。这些算法的平均时间复杂度都是 $O(n \ln n)$，因此是十分有效的。

4 将三角形推广到四面体的困难性

既然二维上的三角剖分有如此之多的漂亮结果和应用，自然会让人思考这个问题在三维（甚至更高维）空间中会是怎么样的情况。但是研究表明，三维的剖分要比二维的情况要复杂得多。

[1] $O(n)$ 指的是算法的时间复杂度，它表示当 n 趋向于无穷时，算法所需要花费的时间不大于 n 的某个常数倍。类似地可以定义 $O(n^2)$，$O(n \ln n)$ 等符号。

[2] $\ln^*$ 表示迭代对数函数，定义如下：

$$\ln^* x = \begin{cases} 1, & x \le 0 \\ 1 + \ln^*(\ln x), & x > 0 \end{cases}$$

这是一个增长极其缓慢的函数。当 x 不大于约 $2.331504384 \times 10^{1656520}$ 时，$\ln^* x < 6$。

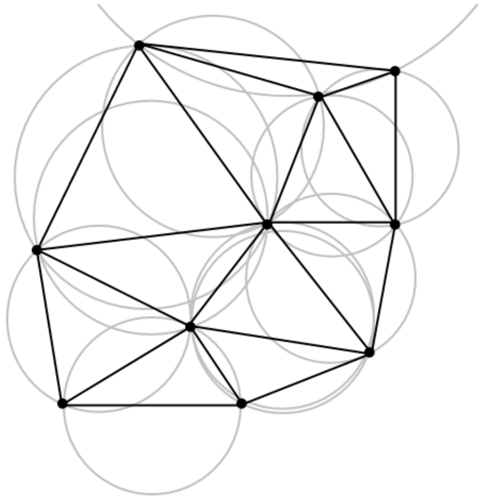

图 4. Delaunay 剖分的示例
https://commons.wikimedia.org/wiki/File:Delaunay_circumcircles.png

一个简单的区别是，二维情况下一个 n 边形能够被剖分成的三角形数目是固定的数 $n-2$，但是即使是同一个多面体，它被剖分成的四面体数目并不是唯一的，例如将两个全等的正四面体以面重合的方式拼成一个六面体，这个六面体既能够被分成两个四面体，也能被分成三个四面体 [13]。由于这样的复杂性，在考虑三维空间剖分的算法和应用之前，首先需要解决的问题是：三维空间中是否任意多面体都可以被剖分成若干个四面体?

对于凸的多面体，答案是肯定的，对顶点数使用归纳法不难给出证明。但是对于凹的多边形来说，这个命题并不成立，也就是说，存在不能被剖分成若干个四面体的多面体。由 6 个顶点，12 条棱，8 个面组成的 Schönhardt 多面体 (Schönhardt polyhedron) 是其中最简单的一个（见图5）[14]。在 Schönhardt 多面体中，每个四个顶点组成的四面体都至少有一条棱完全位于多面体的外部（包括边界，下同），因此根本无法从它当中剖分出一个四面体出来，自然更无法将它剖分成若干个四面体。

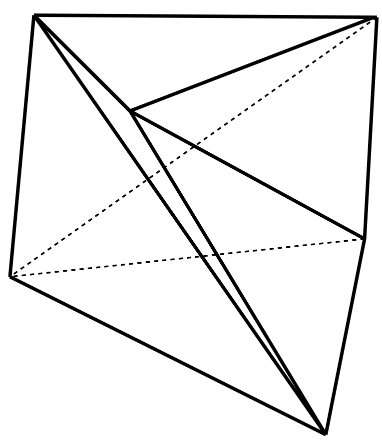

图 5. Schönhardt 多面体

关于 Schönhardt 多面体的对角线还有一个更加完整的结论：对于它的每一条对角线，如果这条对角线不是多面体的棱的话，那么这条对角线必然完全位于多面体的外部。这样的性质并不专属于 Schönhardt 多面体，事实上，[14] 证明了对于任何一个不小于 6 的正整数 n，都存在一个包含 n 个顶点，且满足上述性质的多面体。

现在问题转化为如何判定一个多面体能否被剖分成四面体。我们期望能够找到有效的算法来解决这样的

问题，但是 [13] 指出，这一点是非常困难的，因为这个问题实际上是 NP-完备 (NP-Complete) [1] 的。甚至即使假定这个多面体是星型的（即存在一个内部的点，它与所有顶点的连线都在多面体内部），这个问题依然是 NP-完备的。

我们回到最开始的美术馆问题，考虑一个把美术馆从一个二维的多边形转化为一个三维的多面体进行。从上面的分析中我们可以预料到这个时候，美术馆问题应该也会复杂得多。实际上也确实如此。[4, p255] 给出了一个精巧的反例，反例的大致构造方法是在正方体的前，上，右三个表面交错地钻出一些正方形的洞，使得这些洞几乎但不贯穿整个正方体（见图6），在这种情况下，n 个顶点的美术馆需要至少 $O(n^{3/2})$ 个保安。

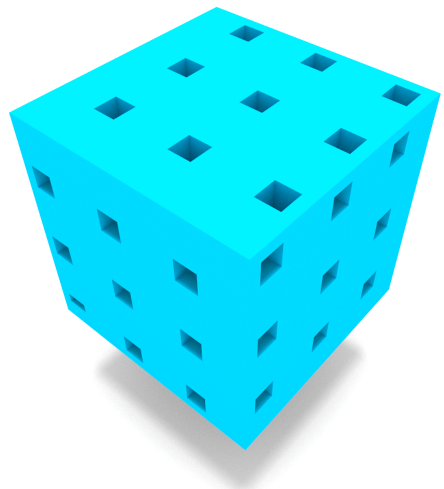

图 6. [4, p255] 所构造出的三维美术馆，其中每个洞都几乎贯穿整个立方体

这些事实表明将三角剖分这个结构推广到高维存在本质性的困难，这种困难来自高维情况下图形会具有更加复杂的结构和组合，从而表明了三角剖分是二维情况下特有的结构，这也是三角剖分这一结构如此独特的原因之一。

5 结论

本文从多个角度介绍了三角形剖分这一结构的相关研究和它在几何学和图形学上的众多应用，展示了三角形剖分这样一个简洁的结构中蕴含的复杂性，也看到了其广泛的应用前景。另一方面，我们也认识到三角形剖分难以直接推广到高维上，因此如何在高维上对图形进行适当的分解，仍是一个值得进一步探究的课题。

参考文献

[1] Peter Giblin. “Computational geometry: algorithms and applications (2nd edn.)” In: *The Mathematical Gazette* 85.502 (2001), pp. 175–176.

[2] Steve Fisk. “A short proof of Chvátal’s Watchman Theorem”. en. In: *Journal of Combinatorial Theory, Series B* 24.3 (June 1978), p. 374.

[3] V Chvátal. “A combinatorial theorem in plane geometry”. en. In: *Journal of Combinatorial Theory, Series B* 18.1 (Feb. 1975), pp. 39–41.

[4] Joseph O’Rourke. *Art gallery theorems and algorithms*. en. The International series of monographs on computer science 3. New York: Oxford University Press, 1987.

[1] 粗略地说，NP-完备问题是一种人们目前还没有找到多项式级别时间复杂度算法的问题。这些问题一般在计算上都是非常困难地。关于它的详细介绍可见文 [15, p127]。

[5] Donald Hearn, M Pauline Baker, et al. *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

[6] Michael R. Garey et al. "Triangulating a simple polygon". en. In: *Information Processing Letters* 7.4 (June 1978), pp. 175–179.

[7] Robert E. Tarjan and Christopher J. Van Wyk. "An $O(n \log \log n)$-Time Algorithm for Triangulating a Simple Polygon". en. In: *SIAM J. Comput.* 17.1 (Feb. 1988), pp. 143–178.

[8] Kenneth L. Clarkson, Robert E. Tarjan, and Christopher J. Van Wyk. "A fast las vegas algorithm for triangulating a simple polygon". en. In: *Discrete Comput Geom* 4.5 (Oct. 1989), pp. 423–432.

[9] Bernard Chazelle. "Triangulating a simple polygon in linear time". en. In: *Discrete Comput Geom* 6.3 (Sept. 1991), pp. 485–524.

[10] William J Schroeder, Jonathan A Zarge, William E Lorensen, et al. "Decimation of triangle meshes." In: *Siggraph*. Vol. 92. 1992, pp. 65–70.

[11] Boris Delaunay et al. "Sur la sphere vide". In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800 (1934), pp. 1–2.

[12] Der-Tsai Lee and Bruce J Schachter. "Two algorithms for constructing a Delaunay triangulation". In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242.

[13] Jim Ruppert and Raimund Seidel. "On the difficulty of triangulating three-dimensional Nonconvex Polyhedra". en. In: *Discrete Comput Geom* 7.3 (Mar. 1992), pp. 227–253.

[14] F. Bagemihl. "On Indecomposable Polyhedra". en. In: *The American Mathematical Monthly* 55.7 (Aug. 1948), pp. 411–413.

[15] Cristopher Moore and Stephan Mertens. *The nature of computation*. OCLC: ocn180753706. Oxford [England] ; New York: Oxford University Press, 2011.