

第八章 几何表示

在几何部分，如何表示空间中的几何体是贯穿其中的主题，研究者创造出多面体网格、点云等方式来描述几何体。本章我们将介绍一种表示几何的经典数据格式——网格模型(mesh)，并分析如何通过网格模型得到几何体的局部性质。

8.1 几何表示：描述三维物体的形状

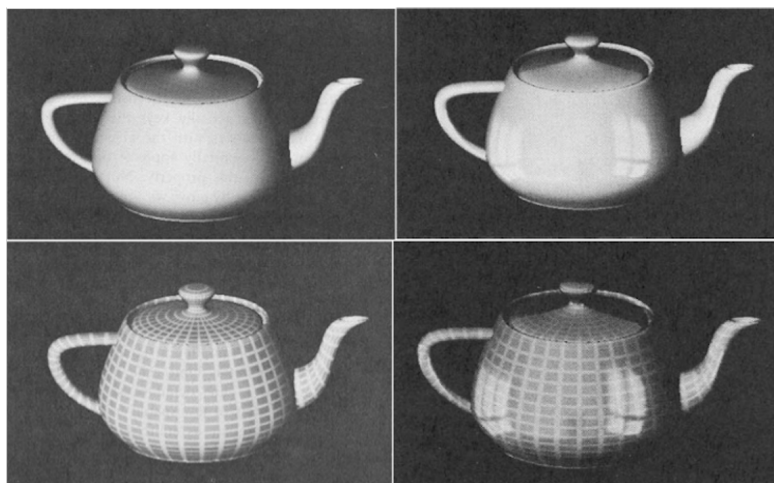


图 8.1: 用贝塞尔曲面渲染成的犹他茶壶.



(a) Stanford Bunny 真实图片

(b) 通过激光扫描得到的网格模型

图 8.2: Stanford Bunny.

几何学 (Geometry) 一词来源于希腊语, 原意是土地测量的学问, 后被我国明朝的徐光启翻译成“几何学”。在可视计算领域中, 我们主要研究对物体的几何形状、位置等信息进行计算机建模、处理和分析。“犹他茶壶” (Utah Teapot) 是最早被用于使用数学曲线而非多边形建模的 3D 模型之一, 最早的版本由 Martin Newell 在 1975 年用贝塞尔曲面渲染而成, 如图8.1所示。另一个著名的模型是由 Greg Turk 和 Marc Levoy 在 1994 年开发的“斯坦福兔子” (Stanford Bunny), 如图8.2所示, 该模型被经常用于测试各种图形学算法中, 之后我们还会经常见到这只兔子。

那么我们可以用什么方法来在计算机中表示几何体呢? 常见的方法有点云 (point cloud)、多边形网格 (polygon mesh)、细分表面 (subdivision surface) 等显式表示方法, 同时也有水平集 (level set)、代数曲面 (algebraic surface) 等隐式表示方法。每种方法都有其特点, 对于不同的任务和不同类型的几何体, 选择合理的表示方式会极大程度地简化问题。

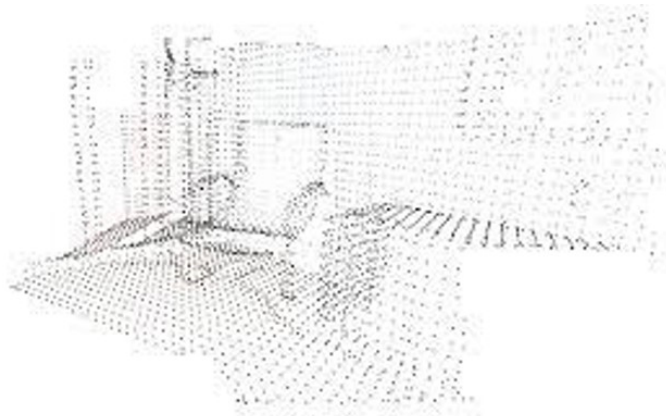


图 8.3: 点云是几何体表面上的点组成的集合, 通常可以用激光雷达或者其他扫描器获取。

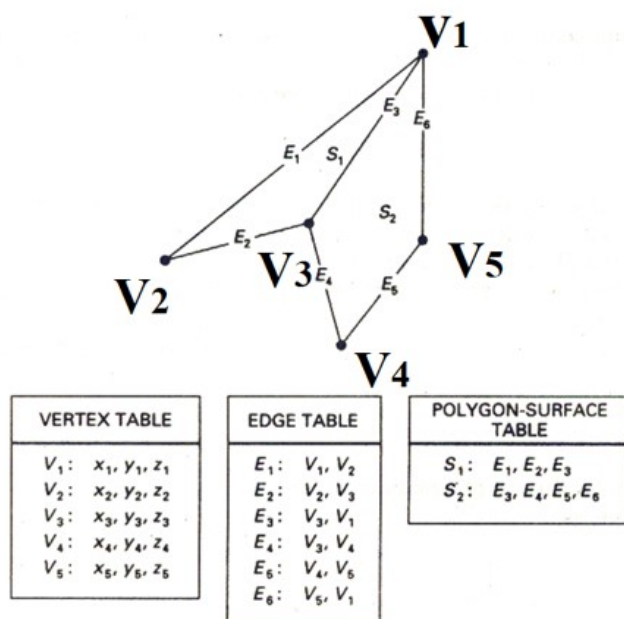


图 8.4: 通过记录顶点表、边表和多边形表面表来表示一个多边形网格。

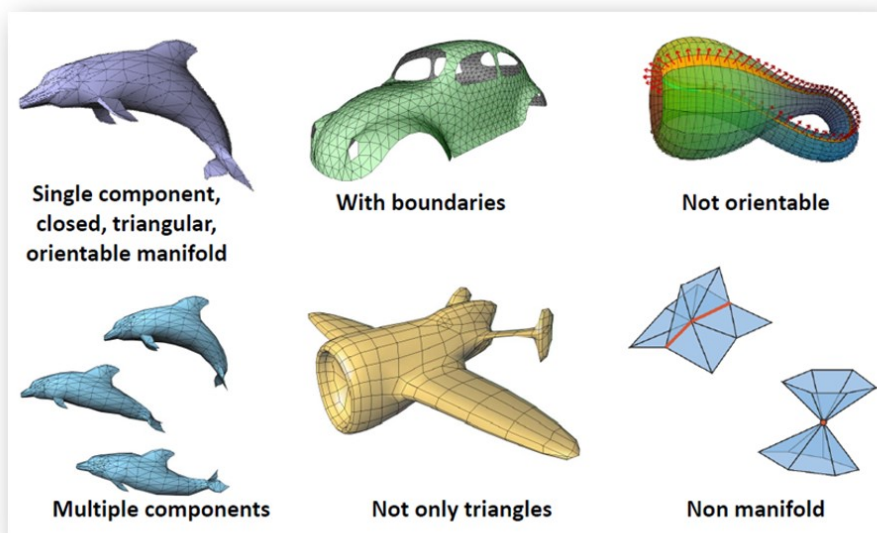


图 8.5: 用多边形网格表示几何体的例子。

8.2 多边形网格模型

多边形网格模型是计算机图形学领域中最常用的几何表达方法，通过记录顶点和多边形来表示几何体的表面。多边形网格模型是由一组顶点、边和面组成，这些定义了物体的形状。顶点是 3D 空间中的单一点，边是连接两个顶点的线，面是边的封闭环。面通常是三角形或四边形，并可用于表示曲面或其他复杂形状。这种表示方法有利于对几何体进行表面处理，但与点云相比需要更为复杂的数据结构。一个典型的多边形网络数据结构如图 8.4 所示，由顶点列表、边列表和面列表构成。在某些应用中，人们还会设计一些额外的数据结构来方便邻居查找等常用操作，从而提高效率。值得一提的是，多面体的顶点数 v 、边数 e 、面数 f 满足 $v - e + f = 2$ ，这个规律被称为欧拉定理。

多边形网格相较于其他几何表达有很多优点。它相对容易理解和使用，也能十分简便地进行渲染和几何形状处理。多边形网格也非常通用，它可以用来创建各种形状，从简单的立方体到复杂的角色。它还可以用于创建 3D 打印模型，以及逼真的动画和模拟。多边形网格具有很高的可扩展性。它可以用来创建任何大小的 3D 模型，从微小的物体到大型结构。图 8.5 展示了用网格模型表示各种形状的例子。

当然，如果要使用网格模型表达光滑的曲面，则需要大量的面片才能让模型变得精细，图 8.6 展示的就是用网格模型表达光滑曲面的例子。

8.2.1 三角网格

三角网格是最常用的几何表示方式，通过记录组成每个三角面片的顶点坐标，可以唯一表达一个几何体。一种常见的表示三角网格的数据结构是分别记录所有顶点的三维坐标，以及每个面片的三个顶点的下标，例如一个四面体就可以按照图 8.7 的方式进行记录。这种数据结构通过共享顶点位置，减少了存储占用，同时保证了网格的整体性，即改变一个顶点在 3 维空间中的位置可以让与该顶点有关的所有三角面片发生移动。

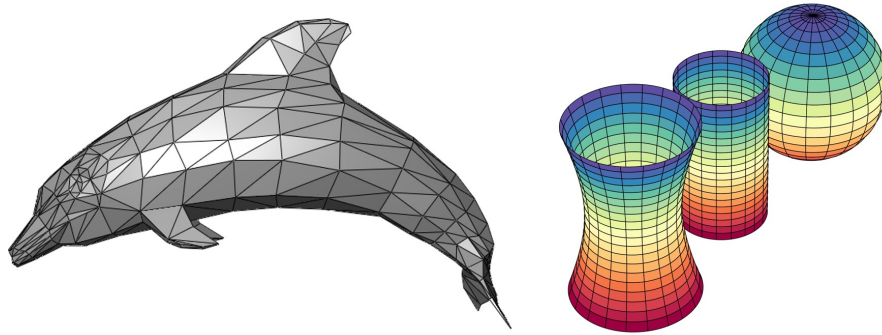


图 8.6: 用网格模型表示曲面. 可以看到当使用的面片数量较少时, 得到的曲面就不够光滑.

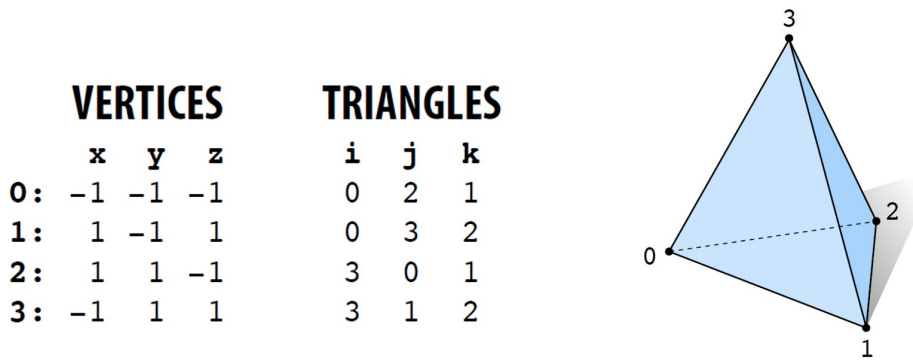


图 8.7: 用三角面片表示一个四面体.

8.2.2 四边形网格

四边形网格也是较为常用的一种网格表示, 所有多边形面片全部是四边形构成, 与三角网格一样, 结构规整, 易于存储. 另外四边形网格的一大优点在于, 其四边形特性在纹理映射贴图时的计算非常方便.

8.3 几何表面细分

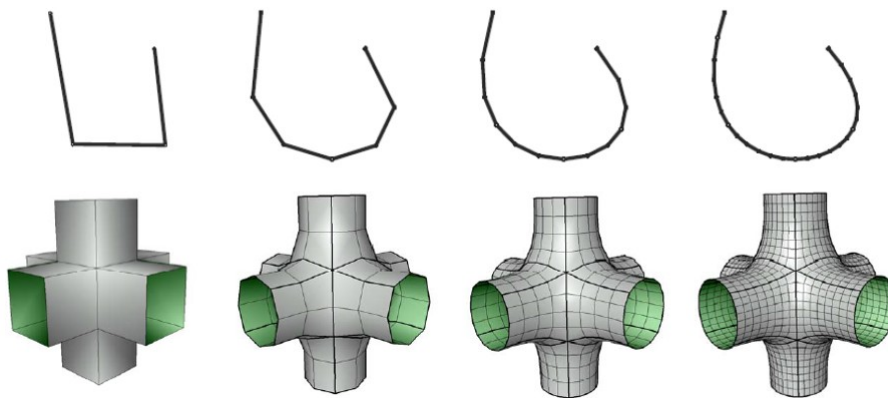


图 8.8: 通过对粗糙表面的不断细分, 可以得到平滑的曲线或曲面.

由上面的分析我们知道, 通过增加组成几何表面的网格面片数量, 减小每个面片的面

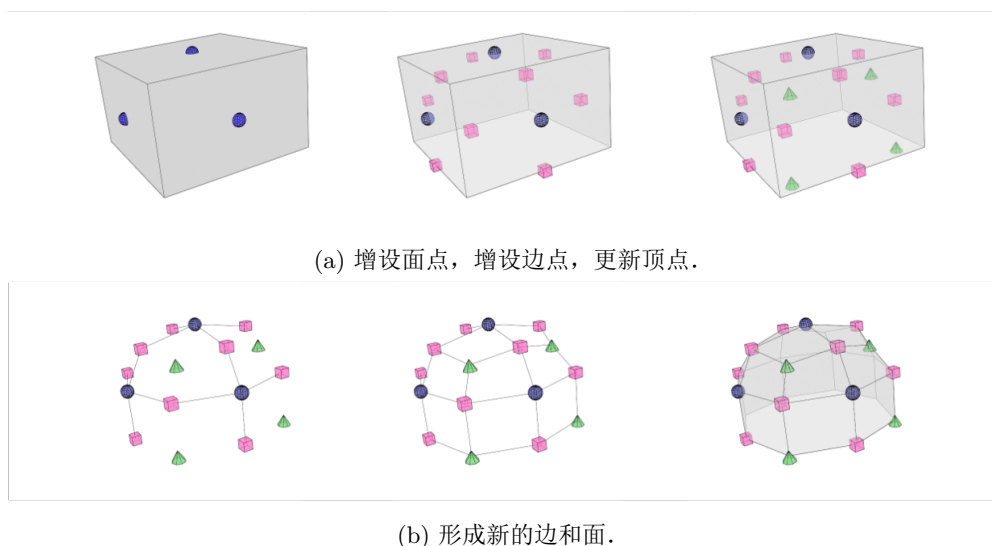


图 8.9: 单轮 Catmull-Clark 细分 (图源: 维基百科).

积, 可以使几何表面看起来更加光滑. 网格细分 (mesh subdivision), 也称为网格的上采样, 是指通过反复细分初始的多边形网格, 不断得到更精细的网格的过程. 因此, 如图9.8所示, 我们可以通过对几何表面进行不断细分, 用这一系列细分的极限来定义出一个平滑的曲线或曲面.

8.3.1 Catmull-Clark 细分

Catmull-Clark 细分是最常用的几何表面细分方法之一, 通过将表面的多边形细分为更小的多边形, 用相邻的顶点重新定位先前的顶点, 对三维多边形网格表面起到平滑效果. 这种方法采用网格中包含的每一个原始多边形, 并将多边形细分为四边形, 基于平均值构建新的顶点, 并根据周围环境调整原始多边形的先前顶点.

Catmull-Clark 细分在 1978 年由 Edwin Catmull 和 Jim Clark 提出, 之后在各种渲染场景中都可以发挥作用, 从学术界到游戏再到动画电影都有其身影, 也因此获得了 2006 年奥斯卡技术成就奖.

Catmull-Clark 算法每一次细分由增设面点 (face point)、增设边点 (edge point)、更新顶点 (vertex point)、形成新的边和面四个步骤组成, 如图8.9所示, 下面我们依次介绍.

1. 增设面点: 对多面体的每个面片计算一个面点, 这个面点是这个多边形面上所有顶点坐标的平均值.
2. 增设边点: 对多面体的每条边计算一个边点, 找出该条边的两个端点和共享该条边的两个面的面点, 对这四个点的坐标取平均.
3. 更新顶点: 对于多边形原有的每个顶点 v , 使用 (1) 所有包含顶点 v 的边的中点 (注意不是上述步骤中的边点) 的平均值 R , (2) 所有包含顶点 v 的多边形面片的面点的平均值 F , (3) 以及顶点 v 的原有值的加权平均值来调整其三维坐标. 加权平均遵循以下公式:

$$\frac{F + 2R + (n - 3)v}{n}$$

其中 n 是面点的数量.

4. 形成新的边和面: 将每个面点连接到所有构成了它所在的原始面的边的边点, 将每个新顶点连接到所有连接着它原始点的边的边点. 新的面就由这些边包围而成.

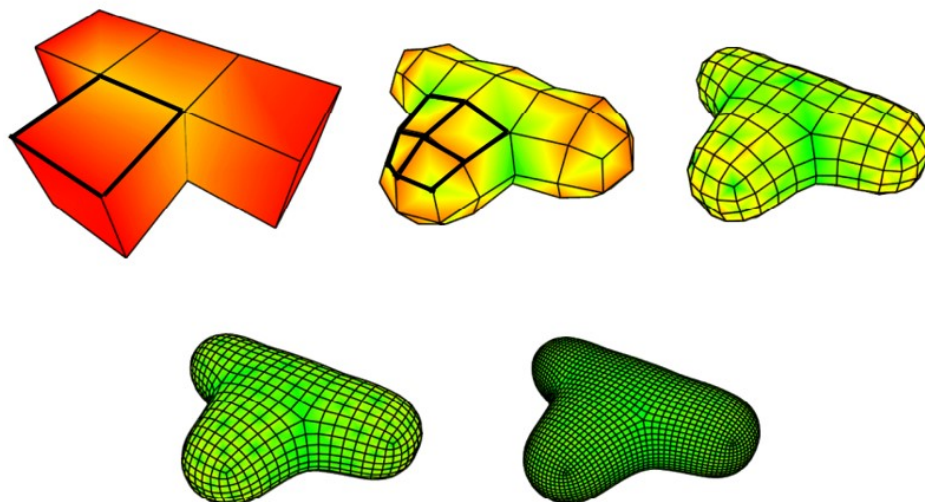


图 8.10: 通过不断地进行 Catmull-Clark 细分, 原本有棱有角的几何表面被细分成平滑的曲面.

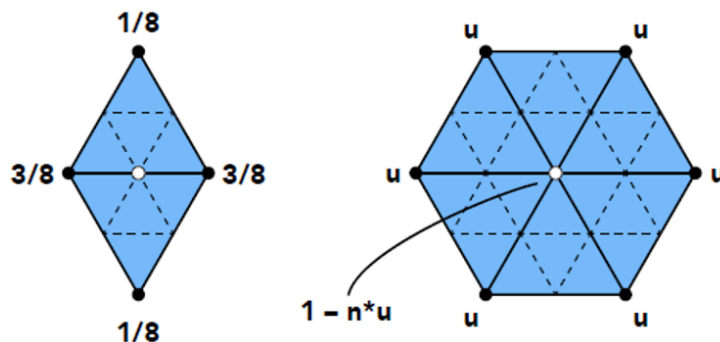


图 8.11: Loop 细分过程.

可以证明, 经过一轮细分后, 不论原多边形网格是何结构, 新得到的多边形网格将只由四边形构成. 如图8.10所示, 在不断细分的过程中, 表面会不断趋向平滑和圆润.

8.3.2 Loop 细分

Loop 细分是常见的针对三角网格模型的细分方法, 由 Charles Loop 在 1987 年提出. 每一次细分将三角面片细分为四个更小的面片, 迭代下去使几何表面变得平滑.

如图8.11, Loop 细分的过程由计算新顶点和更新原有顶点两步组成.

1. 计算新顶点:

- 对于每一条边, 如果这条边被两个三角形面所包含, 如8.11左图所示, 则由这条边的两个端点 v_0, v_2 和“跨过”这条边的两个顶点 v_1, v_3 加权平均计算出新顶点 ep , 具体的计算公式为

$$ep = \frac{3}{8}(v_0 + v_2) + \frac{1}{8}(v_1 + v_3)$$

- 如果这条边只被一个三角形面所包含 (即为边界上的边), 则直接取这条边的中点作为新顶点.

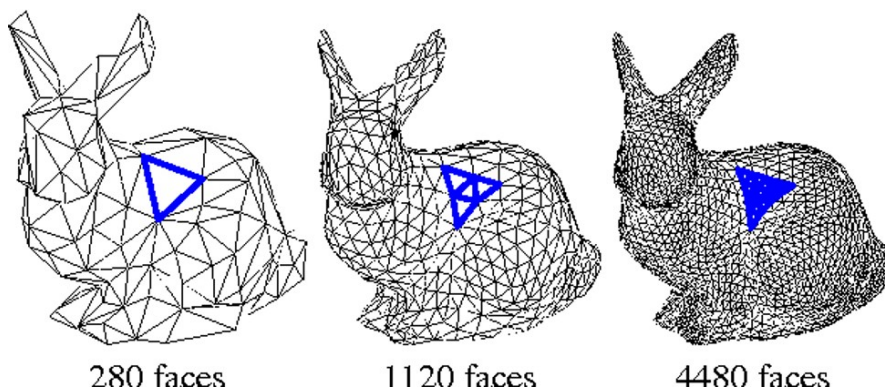


图 8.12: 不断 Loop 细分平滑兔子模型.

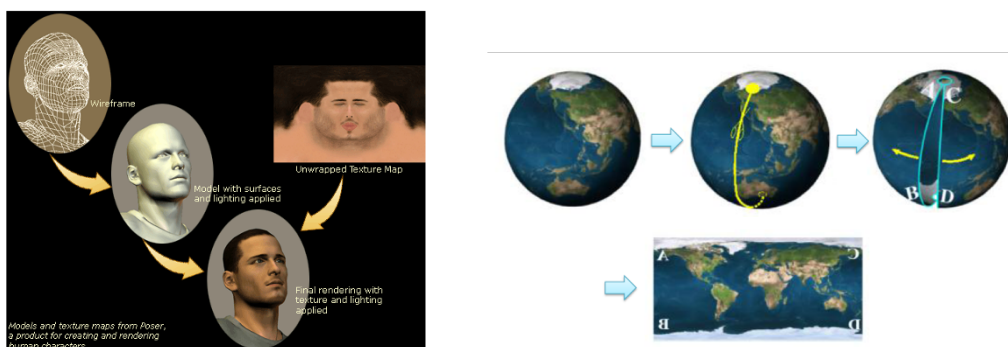
2. 更新原有顶点: 对于每个原有顶点 \mathbf{v} , 根据以下公式更新后的位置 \mathbf{v}'

$$\mathbf{v}' = (1 - n * u)\mathbf{v} + \sum_{i=1}^n u\mathbf{v}_i$$

其中 n 为顶点 v 的度数, v_i 为与 v 有边相连的顶点, 当 $n = 3$ 时 $u = 3/16$, 否则 $u = \frac{3}{8n}$. 通过不断地生成新顶点和更新原有顶点, 每一次细分后连接新顶点与原有顶点, 三角面片变得更加精细, 得到的几何表面更加平滑, 如图8.12所示.

8.4 网格参数化

在计算机图形学中有一个非常独特的领域: 网格参数化 (mesh parameterization), 它是将三维和二维联系起来的桥梁. 网格参数化是数字几何处理中非常重要的问题. 其最早起源于纹理映射的需要. 1974 年, Catmull 提出纹理映射技术, 通过二维纹理空间和三维曲面之间的对应关系, 计算三维曲面上每一点颜色值, 如图8.13a. 这个过程需要建立三维曲面和二维纹理平面之间的一一映射, 也就是参数化的过程, 即计算三维空间曲面的每个网格顶点对应的两个纹理空间中坐标值 (u, v) .



(a) 纹理映射.

(b) 世界地图绘制.

图 8.13: 网格参数化的应用.

参数化的另一个具体应用是世界地图的绘制. 通常可以采用球面坐标系来表示地球表面的每一个点, 给定半径后, 每一个点对应着唯一一组方位角和仰角. 但在实际应用中, 是将世界地图画在平面上的, 如图8.13b所示. 世界地图的绘制技术由来已久, 出现了不同的

技术，典型的有立体投影 (或球极平面投影)、墨卡托投影和朗伯投影等方法，如图8.14所示。

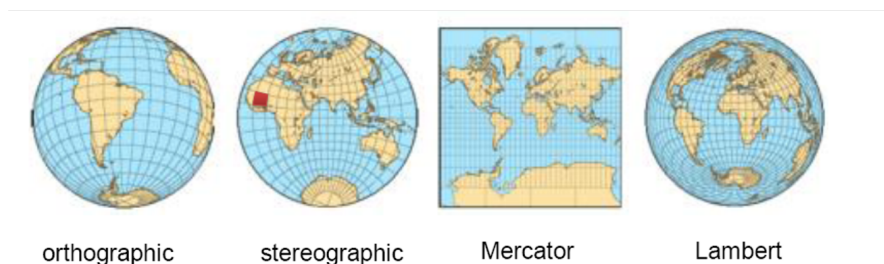


图 8.14: 不同的世界地图绘制方法. 图中依次展现了对地球表面进行正交投影、球极平面投影、墨卡托投影、朗伯投影得到的二维平面图.

1. 球极平面投影: 将一个圆球面通过指定极点射影至一个平面的映射. 能够保持角度不变, 但是面积会发生改变, 尤其在极点附近.
2. 墨卡托投影: 投影后经线是一组竖直的等距离平行直线, 纬线是垂直于经线的一组平行直线. 各相邻纬线间隔由赤道向两极增大. 一点上任何方向的长度比均相等, 即没有角度变形, 而面积变形显著, 随远离标准纬线而增大. 该投影具有等角航线被表示成直线的特性, 故广泛用于编制航海图和航空图等.
3. 朗伯投影: 一种等面积的平面投影, 它不仅可以精确记录面积, 也能不改变方向, 一般用于绘制精度要求较高的地质图或导航图. 这种投影一般不会把整个世界划在同一张图上, 因为地图外围的区域形变会比较大 (外围会在远离基准点的方向压缩, 导致轮廓变得很扁). 它一般只包括某个半球、某个大陆甚至某个区域.

按照参数域的不同, 网格参数化可以分为: (1) 平面 (planar) 参数化, 即将网格映射到平面; (2) 球面 (spherical) 参数化, 即将网格映射到规则的球面; (3) 基域 (simplicial) 参数化, 采用一个同构的简化模型作为参数域; 以及基于其它参数域的参数化.

按照参数化过程中保持哪种几何量, 可以将参数化分为保长度 (Isometric) 的参数化、保角度 (Conformal) 的参数化、保面积 (Equiareal) 的参数化等等, 即: 将网格曲面映射到平面上时, 分别保证长度、角度、面积不发生扭曲. 事实上, 保长度的映射等价于既保角度又保面积的映射. 从三维网格表面映射到平面或其他参数域, 理想的参数化是保形参数化, 即保持局部形状不发生改变. 这里的形状描述包括长度、角度、面积等几何度量. 但理论上只有可展曲面 (developable surface) 存在保形参数化, 如圆柱侧面, 而一般的曲面在进行映射时都会发生不同程度的扭曲.

寻找网格表面顶点和平面上点之间的一一对应, 又可以分为两种类型的平面参数化: (1) 开网格参数化, 这种情形下网格是具有边界的开网格, 又分为固定边界映射和自由边界映射; (2) 闭网格参数化, 这种情形下网格构成一个封闭曲面, 通常人为指定一条边界将封闭网格切开, 然后转化为开网格进行参数化.

开网格的平面参数化中, 最基础的模型是弹簧模型. 如图8.15所示, 网格顶点作为图节点, 网格边作为连接节点的边, 形成多个弹簧连接的一个整体. 建立从网格到纹理空间的参数化映射, 也就是将每个网格顶点 $\mathbf{s}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ 映射成二维参数空间中的点 $\mathbf{t}_i = (u_i, v_i) \in \mathbb{R}^2$, 我们可以采用弹簧系统的能量函数来描述参数化后的系统状态. 弹簧的拉伸最后会达到平衡状态, 使得整个系统能量最低, 这个状态也就是参数化的最终结果. 我

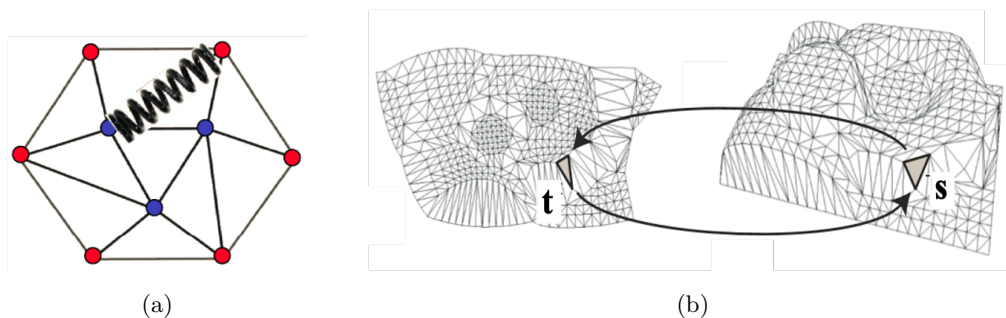


图 8.15: 开网格平面参数化的弹簧模型. (a) 将网格建模成节点-弹簧模型. (b) 将三维空间中的网格点 $\mathbf{s} = (x, y, z)$ 与二维参数空间中的点 $\mathbf{t} = (u, v)$ 建立一一映射关系.

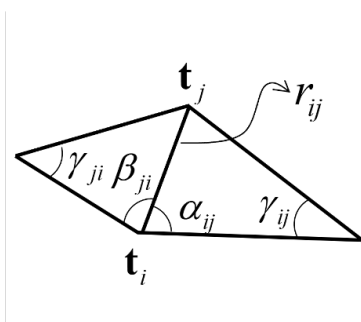


图 8.16: 用边两侧的三角形中的量计算组合系数.

们将总能量定义为加权的弹簧能量:

$$E = \sum_{i=1}^n \sum_{j \in \Omega(i)} \frac{1}{2} D_{ij} \|\mathbf{t}_i - \mathbf{t}_j\|^2 \quad (8.1)$$

其中 n 是节点总数, D_{ij} 是连接第 i 和第 j 个顶点的弹簧的劲度系数, 也是其所对应边在系统中的权重.

这里的自变量即参数化后顶点 $\mathbf{t}_i, i = 1, \dots, n$ 对应的二维坐标. 那么平衡状态对应于关于变量的导数为零:

$$\frac{\partial E}{\partial \mathbf{t}_i} = \sum_{j \in \Omega(i)} D_{ij} (\mathbf{t}_i - \mathbf{t}_j) \quad (8.2)$$

记系数 $\lambda_{ij} = D_{ij} / \sum_{k \in \Omega(i)} D_{ik}$, 可将上述式子整理为:

$$\mathbf{t}_i - \sum_{j \in \Omega(i)} \lambda_{ij} \mathbf{t}_j = \mathbf{0} \quad (8.3)$$

即, 网格每个顶点在参数化后可以表示为相邻顶点的线性组合, 组合系数 λ_{ij} 是连接该顶点的弹簧系数的加权平均, 也被称为仿射组合系数. 在指定好组合系数之后, 该方程变成关于顶点参数化坐标的线性方程组. 对所有顶点的约束构成一个齐次方程组, 包含 $2n$ 个未知数和 $2n$ 个线性方程. 如果不增加约束条件, 那么零解是其平凡解. 可以通过对系统施加额外约束从而得到非平凡解.

一种约束条件采用固定边界映射, 也称为重心坐标映射方法. 将开网格边界上的顶点首先映射到指定的凸多边形边界上, 例如正方形, 那么这些顶点对应的参数化坐标在方程组中变为已知, 从而可以求解内部顶点的参数化后的坐标.

另外，在弹簧系统中，可以通过设置合适的仿射组合系数来引导平面网格的参数化。特别的，这组仿射组合系数需要满足一些条件，例如凸组合，即保证凸多边形的加权平均仍然在凸多边形内部；线性重构，即如果网格本身是平面上的网格，组合后的点保持不变。常用的仿射组合系数有三种：

1. 平均系数： $\lambda_{ij} = \frac{1}{n_i}$ ，即设置为相邻顶点个数的倒数；
2. 均值坐标系数： $\lambda_{ij} = (\tan \frac{\beta_{ji}}{2} + \tan \frac{\alpha_{ij}}{2})/r_{ij}$ ，式中符号含义见图8.16
3. 调和坐标系数： $\lambda_{ij} = (\cot \gamma_{ij} + \cot \gamma_{ji})/2$ ，只和角度有关，近似于保角映射。