极小曲面生成和网格参数化与纹理映射

PB17000123 张湛

2020 年 3 月 14 日

摘要

极小曲面在数学上指的是平均曲率处处为零的曲面。我们对于极小曲面使用的一个不可避免的问题就是需要有能力去建立任意给定边界下的极小曲面。而现在我们的给定边界以及待求曲面所采用的形式均为三角网格数据。原始数据为三角网格形式的带有一条边界曲面模型，我们将通过局部法和全局法两种方法来实现内部点的位置重建，并在保持固定边界的前提之下使之化为一个极小曲面。

三维曲面离散化为三维网格后，为了进行纹理贴图，有时还要展开到二维的 uv 坐标平面上。虽然大部分空间曲面都不是可展的，但是在曲面的局部区域观察是可以表示成 $f(u,v)$ 的形式的，这个过程在数字图形处理中的术语叫参数化（$(Parameterization)$。从三角网格的角度，每个三角形和它的 uv 平面上的对应都是仿射变换的关系，根据不同的应用，通常要给这种局部映射加约束，比如要求角度或者面积的变形最小等，在曲面不撕裂的情况下，这种假设只能近似满足。

纹理映射是真实感图像制作的一个重要部分，运用它可以方便的制作出极具真实感的图形而不必花过多时间来考虑物体的表面细节。然而纹理加载的过程可能会影响程序运行速度，当纹理图像非常大时，这种情况尤为明显。如何妥善的管理纹理，减少不必要的开销，是系统优化时必须考虑的一个问题。还好，OpenGL 提供了纹理对象对象管理技术来解决上述问题。与显示列表一样，纹理对象通过一个单独的数字来标识。这允许 OpenGL 硬件能够在内存中保存多个纹理，而不是每次使用的时候再加载它们，从而减少了运算量，提高了速度。

1　三角网格与半边数据结构

最简单的情形，多边形网格不过是一个多边形列表；三角网格就是全部由三角形组成的多边形网格。多边形和三角网格在图形学和建模中广泛使

用，用来模拟复杂物体的表面，如建筑、车辆、人体，当然还有茶壶等。当然，任意多边形网格都能转换成三角网格，三角网格以其简单性而吸引人，相对于一般多边形网格，许多操作对三角网格更容易。关于半边数据结构，

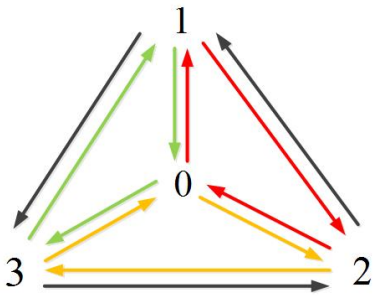

图 1: 半边数据结构

其最大特点当然是半边，每个边分为两个半边，每个半边都是一个有向边，方向相反。如果一个边被两个面片共用（正则边），则每个面片都能各自拥有一个半边。如果一个边仅被一个面片占有（边界边），则这个面片仅拥有该边的其中一个半边，另一个半边为闲置状态。

2 极小曲面生成的局部方法与全局方法

2.1 极小曲面

在数学中，极小曲面是指平均曲率为零的曲面。举例来说，满足某些约束条件的面积最小的曲面。物理学中，由最小化面积而得到的极小曲面的实例可以是沾了肥皂液后吹出的肥皂泡。肥皂泡的极薄的表面薄膜称为皂液膜，这是满足周边空气条件和肥皂泡吹制器形状的表面积最小的表面。这种曲面的研究始于有关满足一定的约束条件（比如边界固定或容纳体积满足一定条件）下表面积最小的曲面，因此被称为“极小曲面”。实际上极小曲面所囊括的内涵比此类最小面积曲面更广泛。极小曲面的定义还可以扩展到恒定平均曲率曲面，即曲面上由平均曲率等于某个常数的点组成的子曲面。当这个常数等于零的时候，恒定平均曲率曲面就是极小曲面。

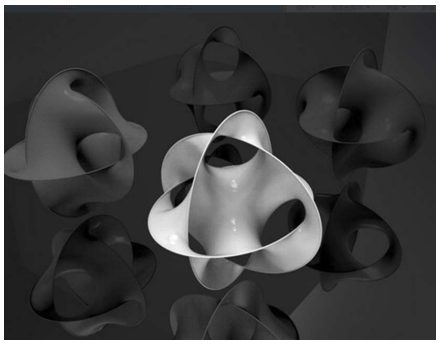

图 2: 极小曲面

2.2 局部方法

局部法的思想是用新的点不断代替旧的点，每次都向其当下所有临点的重心的方向移动一定比例的距离，执行到一定次数为止。我在网上看到为了避免改变过猛引起的极端情况，我们采取少量多次的方法。将学习比例取 0.3 而不是 1 是，并迭代 3000 次，所以我们的点不会一下子就移到重心处。

下面是给出的效果：

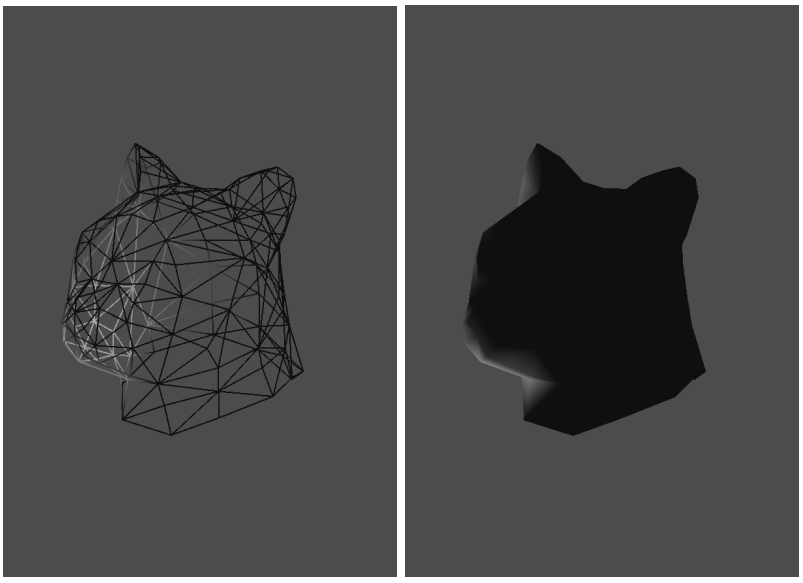

图 3: cat_head

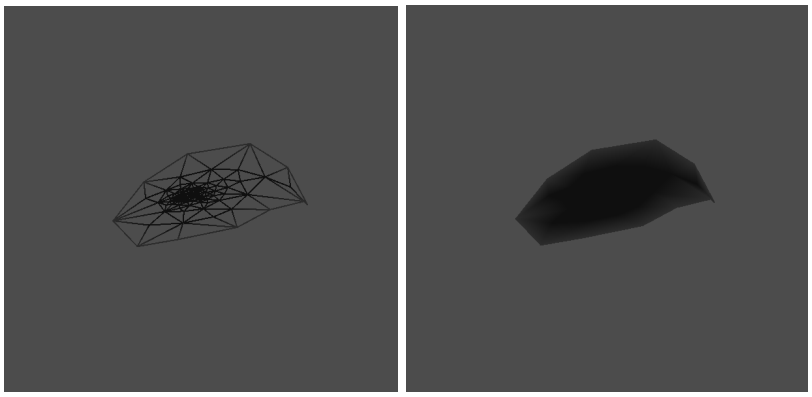

图 4: cat_head_minimalsurface

2.3 全局方法

全局法的思想就是解方程。每个非边界点的位置设都是未知量，每个边界点的位置都是已知量，由极小曲面的限制条件来构成方程。

$$p_i = \frac{1}{d_i} \sum_{p \in N(p_i)} p \text{ ，若 } p_i \text{ 不在边界上}$$

我们通过 Eigen 库解稀数矩阵方程 $Ax = b$ 即可。

下面是给出的效果：

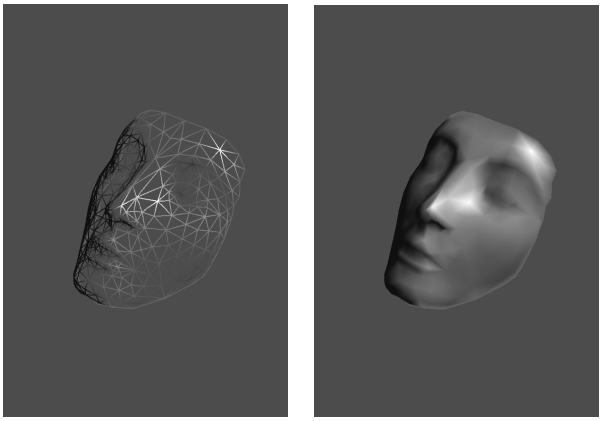

图 5: Nefertiti_face

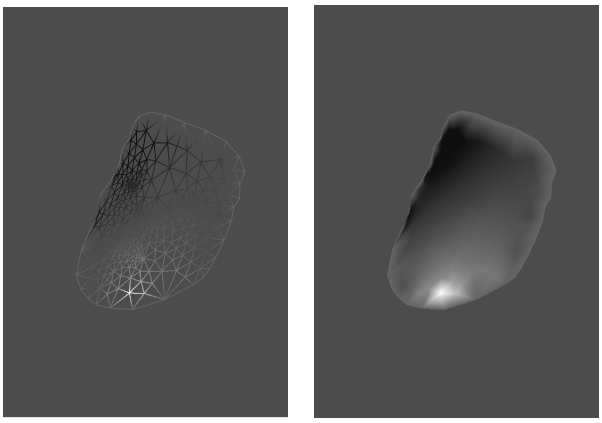

图 6: Nefertiti_face_minimalsurface

2.4 对比

总体上，两者都实现了极小曲面的生成，且效果不错。若原始模型点分布不是非常均匀，且有明显密集尖锐部分，局部法迭代的结果将不是太好。与此同时，从算法的速度上来看：若局部方法迭代次数不多的，则会相对速度快些；若网格点数较少，则全局方法快些。

下面是给出的效果：

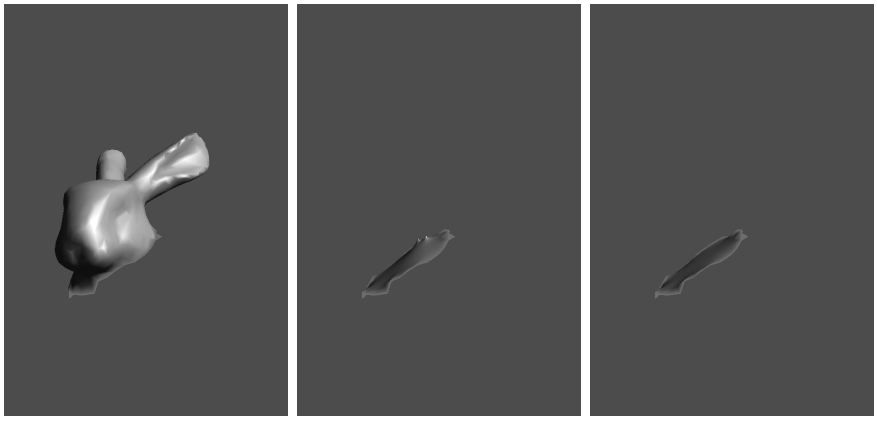

图 7: Bunny_head orignal, local, global

3 网格参数化

三角网格参数化可归结为这样一个问题：给定一个由空间点集组成的三角网格和一个二维参数域。通常为平面或者球面。求一个参数域上的点 x_i 到网格上的点 u_i 的一一映射。使得参数域上的网格与原网格拓扑结构同构，并保证参数与上的三角形步重叠的同时谋求某种与原始网格之间几何变量的变形的最小化。

$$u_i = \sum_{j=1}^{N} \lambda_{i,j} u_j \text{ , 当 } u_i \text{ 非边界点时}$$

其中，我们假设 $\{u_1, ..., u_n\}$ 是非边界点，$\{u_{n+1}, ..., u_N\}$ 是边界点。

$$\sum_{j=1}^{N} \lambda_{i,j} = 1, \lambda_{i,j} = 0 \text{ 当} (i,j) \notin E, \lambda_{i,j} > 0 \text{ 当} (i,j) \in E$$

3.1 Uniform Parameterization

取 $\lambda_{i,j} = \frac{1}{d_i}$，$\forall j = 1, ..., N$，这里 d_i 是 u_i 的度。

下图，是对猫头进行的参数化：

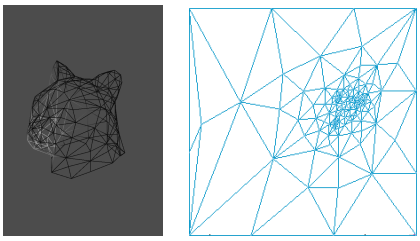

图 8: uniform

3.2 Weighted Least Squares Parameterization

令 $F(u_1, u_2, ..., u_n) = \sum_{(i,j)\in E} w_{i,j} \|u_i - u_j\|^2$，我们将这个能量函数最小化即可。取 $w_{i,j} = 1/\|x_i - x_j\|^q$，计算出最小化 F 的 $\{u_i\}$ 即可。该方法，等价于取 $\lambda_{i,j} = w_{i,j} / \sum_{j:(i,j)\in E} w_{i,j}$，进行运算。

下图，是对猫头进行的参数化：

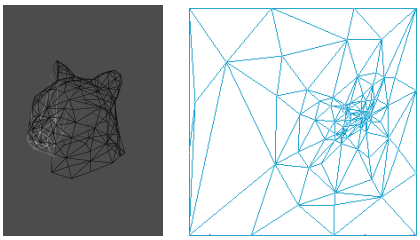

图 9: weighted least squares

3.3 Shape-preserving Parameterization

对于保形参数化内部点的求解方法，这里比较复杂一点。

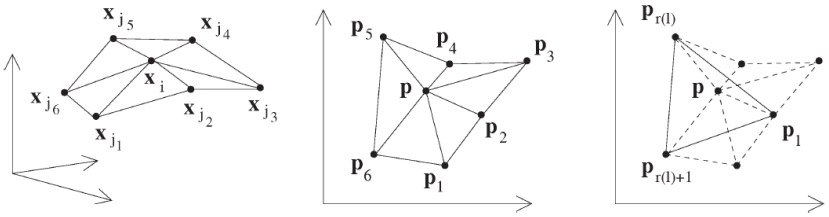

图 10: The subtriangulation and local parametrization

在局部参数化中，有

$$\|p_k - p\| = \|x_{j_k} - x_i\|,\ ang(p_k, p_i, p_{k+1}) = 2\pi\ ang(x_{j_k}, x_i, x_{j_{k+1}})/\theta_i$$

这里 $\theta_i = \sum_{k=1}^{d_i} ang(x_{j_k}, p, x_{j_{k+1}})$ 也就是领域内各个角度的和。其实这两个公式的意思就是保证长度一样和角度的比一样了。

然后就可以求得每个点的权重了。对于每个 p_l，我们找与 $p_l p$ 相交的对边的端点 $p_{r(l)}, p_{r(l)+1}$（若是交于一点，则后面面积比变为线段比即可）。有

$$\mu_{l,l} = \frac{area(p, p_{r(l)}, p_{r(l)+1})}{area(p_l, p_{r(l)}, p_{r(l)+1})}, \mu_{r(l),l} = \frac{area(p, p_{r(l)+1}, p_l)}{area(p_l, p_{r(l)}, p_{r(l)+1})}, \mu_{r(l)+1,l} = \frac{area(p, p_l, p_{r(l)})}{area(p_l, p_{r(l)}, p_{r(l)+1})}$$

最后求出的权重求一个平均值就好了:

$$\lambda_{i,j} = \frac{1}{d_i} \sum_{l=1}^{d_i} \mu_{k,l},\ k = 1, ..., d_i$$

这样就得到了整个权重了。最后在得到了权重以后再根据上述凸组合的公式求解线性方程求出参数化结果。

下图，是对猫头进行的参数化：

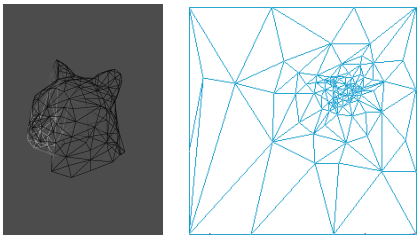

图 11: shape-preserving

4 纹理映射

完成参数化后，纹理映射就简单许多了。我们把下面这张图映到球上：

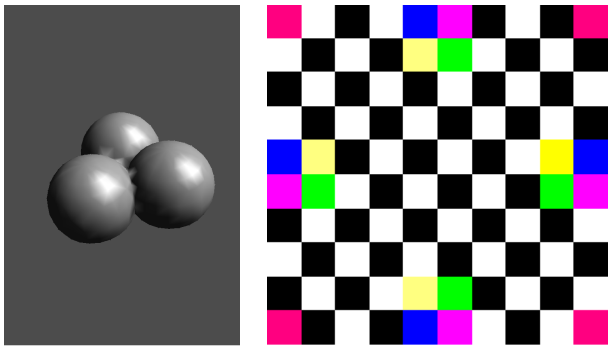

图 12: texture

这里，是三种不同参数化方法纹理映射后的结果：

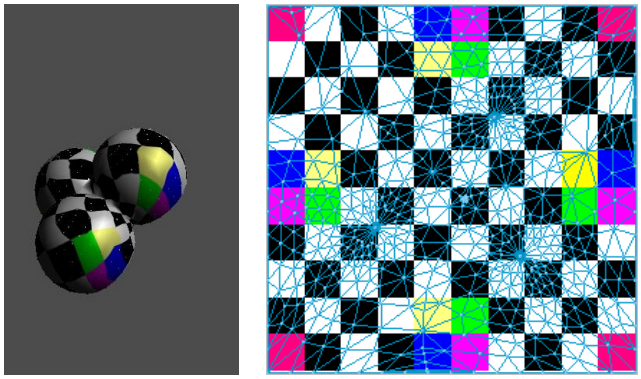

图 13: uniform parameterization

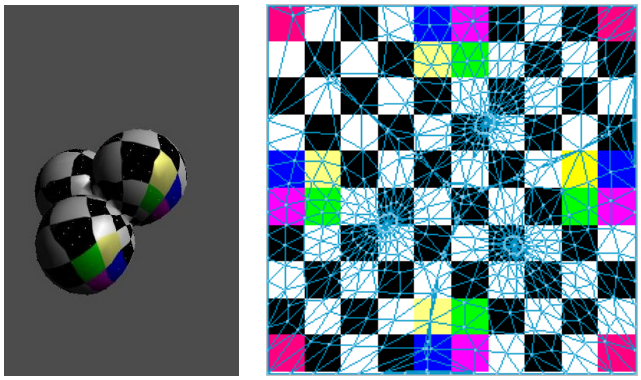

图 14: weighted least squares parameterization

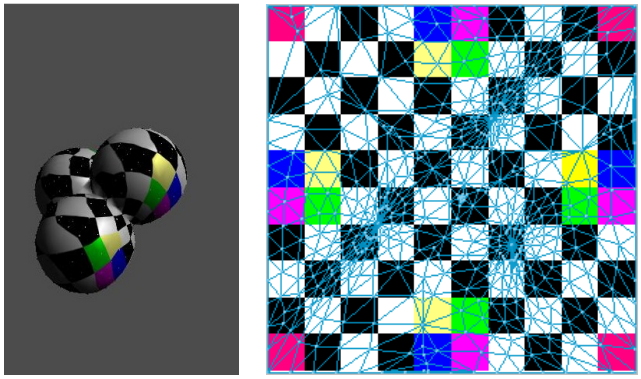

图 15: shape-preserving parameterization