

1 几何处理

1.1 几何操作的数学基础

1.2 离散微分几何

我们总是用离散的网格¹来描述连续的曲面, 但网格定义出的几何形状只能给出 C^0 连续性. 如何将连续曲面的各种微分量应用在离散的情况下就属于**离散微分几何**研究的内容. 常见的各种微分量有法向量, 曲率, 拉普拉斯算子等.

1.2.1 局部平均区域

我们可以用下面的方法刻画三角网格顶点 \mathbf{x}_i 处的微分量 f :

$$f(\mathbf{x}_i) = \left(\iint_{\Omega(\mathbf{x}_i)} f dS \right) / \left(\iint_{\Omega(\mathbf{x}_i)} dS \right)$$

即在 \mathbf{x} 的某个邻域 $\Omega(\mathbf{x})$ 的邻域上该微分量的平均值. 邻域的选择直接影响离散微分量的结果和准确度. 邻域越大, 结果越平滑, 但也会丢失更多的细节. 邻域越小, 结果越接近真实值, 但也对噪声更敏感. 常见的邻域选择有以下几种:

1. **重心单元 (barycentric cell)**: 把与顶点 \mathbf{x} 相邻的所有三角形的两边的中点和重心连起来形成的多边形区域.
2. **泰森多边形单元 (Voronoi cell)**: 把与顶点 \mathbf{x} 相邻的所有三角形的外心连起来形成的多边形区域.
3. **混合泰森多边形单元 (mixed Voronoi cell)**: Voronoi cell 在钝角三角形中取的点可能超出三角形外, 因此在钝角三角形中用边的中点代替外心, 其余情形不变.

1.2.2 法向量

三角形内部点的法向量可以唯一地定义为三角形所在平面的法向量. 对于顶点 \mathbf{x} 处的法向量 $\mathbf{n}(\mathbf{x})$, 可以通过对顶点周围三角形的法向量加权平均得到:

$$\mathbf{n}(\mathbf{x}) = \frac{\sum_{T \in \Omega(\mathbf{x})} \alpha_T \mathbf{n}(T)}{\sum_{T \in \Omega(\mathbf{x})} \alpha_T \|\mathbf{n}(T)\|}$$

其中 $\Omega(\mathbf{x})$ 是与顶点 \mathbf{x} 邻接的三角形构成的几何, α_T 和 $\mathbf{n}(T)$ 是三角形 T 的权重和法向量. 常见的权重有以下几种:

1. **常数权重**: $\alpha_T = 1$. 这在计算上最简单, 但对于一些不规则网格效果不好.
2. **面积权重**: $\alpha_T = A_T$. 以三角形面积 A_T 加权, 比常数权重更合理, 但对过于不规则的网格效果也不好.
3. **角度权重**: $\alpha_T = \theta_T$, 其中 θ_T 是顶点 \mathbf{x} 在三角形 T 中的对角. 这样加权效果最好, 但计算上也最复杂.

¹在本节, 我们主要以三角网格作为研究对象.

1.2.3 梯度

梯度是非常重要的微分量, 在计算拉普拉斯算子, 网格参数化等方面有重要应用.

对于一个在三角网格上定义的标量函数 f , 我们可以通过 f 在三角形顶点上的值通过重心插值的方式得到三角形内部的值. 不妨设三角形的顶点为 $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c$, 则其内部任意一点 \mathbf{x} 的值为:

$$f(\mathbf{x}) = \alpha(\mathbf{x})f(\mathbf{x}_a) + \beta(\mathbf{x})f(\mathbf{x}_b) + \gamma(\mathbf{x})f(\mathbf{x}_c)$$

其中 α, β, γ 是点 \mathbf{x} 在三角形中的重心坐标, 在 **Lecture 3 绘图** 一节中已经讲过其定义方式. 我们在那时已经证明重心坐标是线性的, 因此 f 的梯度 ∇f 也是定值. 现在来证明之.

证明. 不妨假定 $\mathbf{x}_a, \mathbf{x}_b$ 和 \mathbf{x}_c 在三角形 T 中对应的顶点分别为 A, B 和 C , \mathbf{x} 对应 T 内的一点 P . 以 $\alpha(\mathbf{x})$ 为例, 根据重心坐标与三角形面积的关系, 有

$$\alpha(\mathbf{x}) = \frac{A_{\triangle PBC}}{A_{\triangle ABC}} = \frac{\overrightarrow{BP} \cdot \overrightarrow{BC} \cdot \sin \angle PBC}{2A_T} = \frac{\|\overrightarrow{BP} \times \overrightarrow{BC}\|}{2A_T} = \frac{(\mathbf{x} - \mathbf{x}_b) \cdot (\mathbf{x}_c - \mathbf{x}_b)^\perp}{2A_T}$$

其中 $(\mathbf{x}_c - \mathbf{x}_b)^\perp$ 表示该向量在平面上逆时针旋转 90° 后的向量. 于是

$$\nabla \alpha(\mathbf{x}) = \frac{(\mathbf{x}_c - \mathbf{x}_b)^\perp}{2A_T}$$

于是

$$\nabla f(\mathbf{x}) = \frac{(\mathbf{x}_c - \mathbf{x}_b)^\perp}{2A_T} f(\mathbf{x}_a) + \frac{(\mathbf{x}_a - \mathbf{x}_c)^\perp}{2A_T} f(\mathbf{x}_b) + \frac{(\mathbf{x}_b - \mathbf{x}_a)^\perp}{2A_T} f(\mathbf{x}_c)$$

这表明 $\nabla f(\mathbf{x})$ 与 \mathbf{x} 无关, 即在三角形内部为常值. □

1.2.4 离散拉普拉斯算子

定义 1.1 拉普拉斯算子 拉普拉斯算子 (Laplace Operator, Laplacian) 是一个二阶微分算子, 记号为 Δ 或 ∇^2 , 定义为函数梯度的散度:

$$\Delta f = \operatorname{div}(\operatorname{grad} f)$$

在 n 维欧氏空间中, 对于一个标量函数 $f(x_1, \dots, x_n)$, 拉普拉斯算子可以表示为:

$$\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$$

我们知道, 对于一维函数, 其二阶导数表示该函数的凹凸程度; 而 Laplacian 是多维函数在各个正交的方向上的二阶导数之和, 也就刻画了该函数在某一点整体的凹凸程度.

我们也可以用梯度和散度的几何意义理解 Laplacian. 梯度作为向量场衡量了函数在各点的变化方向, 再对其求散度就可以得到函数在该点附近梯度的情况. 如果梯度场流入某一点更多, 此时该点的 Laplacian 为正, 表明此处函数内凹 (类似于山谷); 反之, 如果梯度场流出某一点更多, 此时该点的 Laplacian 为负, 表明此处函数外凸 (类似于山峰). 当梯度场流入和流出的量近似时, 该点的 Laplacian 接近零, 表明此处函数比较平坦²(类

²这里的平坦不是指函数值变化不大, 而是指函数值的变化速率变化不大, 类似于一次函数.

似于山坡).

Laplacian 对于刻画曲面性质有重要作用. 例如, 我们可以用 Laplacian 计算由参数方程 $S(u, v)$ 在三维空间中定义的曲面 S 的平均曲率³:

$$H = -\frac{\Delta S}{2n}$$

在离散情况下, 我们可以得到如下结论.

定义 1.2 离散拉普拉斯算子 在三角形网格中, 函数 f 在顶点 \mathbf{x}_i 处的离散拉普拉斯算子 (Discrete Laplace Operator) 定义为:

$$\Delta f(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \Omega(\mathbf{x}_i)} w_{ij} (f(\mathbf{x}_j) - f(\mathbf{x}_i))$$

其中 $\Omega(\mathbf{x}_i)$ 是与顶点 \mathbf{x}_i 邻接的顶点集合, w_{ij} 是权重.

我们现在来证明这一定义的合理性.

证明. 对于 \mathbf{x}_i 的一个邻域 $\Omega(\mathbf{x}_i)$, 根据散度定理有

$$\iint_{\Omega(\mathbf{x}_i)} \Delta f dS = \int_{\partial\Omega(\mathbf{x}_i)} \nabla f \cdot \mathbf{n} ds$$

其中 \mathbf{n} 表示给定点处的法向量. 在梯度一节中我们已经证明 ∇f 在给定三角形上是定值, 因此我们对 \mathbf{x}_i 邻接的各个三角形 T_j 单独考虑, 设 T_j 的顶点为 $\mathbf{x}_i, \mathbf{x}_{j1}, \mathbf{x}_{j2}$. □

1.3 网格平滑

随着三维扫描和曲面重建技术的发展, 得到实体表面的多边形网格表示已经不是难事, 但所得到的表面往往包含噪声. 为了将其转化为更光滑的网格以便后续处理, 我们需要用到网格平滑技术.

定义 1.3 网格平滑 网格平滑 (Mesh Smoothing)(又称为网格降噪 (Mesh Denoising) 通过) 对多边形网格进行处理, 以减少噪声, 从而得到更光滑, 性质更优良的网格表示.

一般而言, 噪声所在的位置都是曲率较大, 形状比较尖锐的地方. 因此, 网格平滑的目标是减少这些高频噪声, 同时尽可能地保留网格的整体形状和细节.

我们在本节主要介绍拉普拉斯平滑 (Laplacian Smoothing).

我们用扩散流的数学模型来描述网格平滑的过程. 考虑热量传导的物理模型, 设 $f(\mathbf{x}, t)$ 表示时间 t 时刻位置 \mathbf{x} 处的温度, 则扩散方程为:

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \lambda \Delta f(\mathbf{x}, t)$$

³对于一个曲面 S 上某一点 P , 过 P 的法向量做平面可以与 S 交出一条曲线 C . 改变平面的方向所得的 C 中在 P 处的曲率有极大值 k_1 和极小值 k_2 , 称作曲面 S 在点 P 处的**主曲率** (Principal Curvature). 曲面 S 在 P 处的**平均曲率** (Mean Curvature) 定义为 $H = (k_1 + k_2) / 2$. 以后也许还会用到**高斯曲率** (Gaussian Curvature), 定义为 $K = k_1 k_2$.

在网格中, 高频的噪声就对应于杂乱分布的高温点. 随着扩散的进行, 热量分布趋于均匀, 对应顶点构成的表面也趋于平滑.

现在, 我们需要将扩散流公式在时间和空间上进行离散化, 以便在计算机中处理. 在空间上的 Laplacian 离散化已经在前面一节中介绍, 即

$$\frac{\partial f(\mathbf{x}_i, t)}{\partial t} = \lambda \Delta f(\mathbf{x}_i, t) = \lambda \sum_{\mathbf{x}_j \in \Omega(\mathbf{x}_i)} w_{ij} (f(\mathbf{x}_j) - f(\mathbf{x}_i))$$

而在时间上的离散化, 可以将 t 划分为步长为 h 的若干个间隔, 用有限差分的思想对微分方程进行近似:

$$\frac{\partial f(\mathbf{x}_i, t)}{\partial t} = \frac{f(\mathbf{x}_i, t+h) - f(\mathbf{x}_i, t)}{h}$$

于是

$$f(\mathbf{x}_i, t+h) = f(\mathbf{x}_i, t) + h \frac{\partial f(\mathbf{x}_i, t)}{\partial t} = f(\mathbf{x}_i, t) + h\lambda \sum_{\mathbf{x}_j \in \Omega(\mathbf{x}_i)} w_{ij} (f(\mathbf{x}_j) - f(\mathbf{x}_i))$$

在网格简化中, 我们考虑的“温度”就是顶点的空间坐标. 于是, 在每次时间间隔为 h 的第 k 次迭代中, 每个顶点 \mathbf{x}_i 的位置按照如下的方式更新:

$$\mathbf{x}_i^{(k+1)} \leftarrow \mathbf{x}_i^{(k)} + \lambda \Delta \mathbf{x}_i^{(k)}$$

其中 λ 为平滑参数即为前面推导中的 $h\lambda$.

在两种 Laplacian 下, 上述迭代方法可以得到不同的平滑效果. 均匀 Laplacian 下, 每个顶点都向其邻居的均值移动, 这种方法简单且计算效率高, 但容易导致网格收缩和细节丢失. 余切 Laplacian 下, 每个顶点向平均曲率的方向移动, 也被称作平均曲率流 (Mean Curvature Flow). 这种方法能够更好地保留网格的形状和细节, 但是计算复杂度较高.

1.4 网格融合

1.5 网格简化

在图形学中, 渲染效率与渲染质量需要兼顾. 同样的物体在近处时需要使用精度较高, 面数较多的模型以保证真实; 而在远处时由于物体在屏幕上占据的像素较少, 使用精度较低, 面数较少的模型也能保证视觉效果. 因此, 我们需要对高精度的网格进行简化以提高渲染效率. 这时就需要用到**网格简化**算法来生成不同**细节层次 (Level of Detail, LoD)** 的模型用于不同情况下的渲染.

定义 1.4 网格简化 (Mesh Simplification) 通过对多边形网格进行处理, 以减少其面数, 从而得到更简洁, 渲染效率更高的网格表示.

网格简化一般可以通过移除顶点或坍塌边实现, 实际应用中后者更容易实现, 也更常用. 边坍塌算法的核心是设计方法找出哪些边的坍塌对网格形状的影响最小, 以及确定坍塌后新顶点的位置, 从而在简化时尽量保

留网格的整体形状.

我们在本节主要介绍二次误差度量 (Quadric Error Metrics, QEM) 方法⁴.

⁴Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. 1997, p. 209-16.