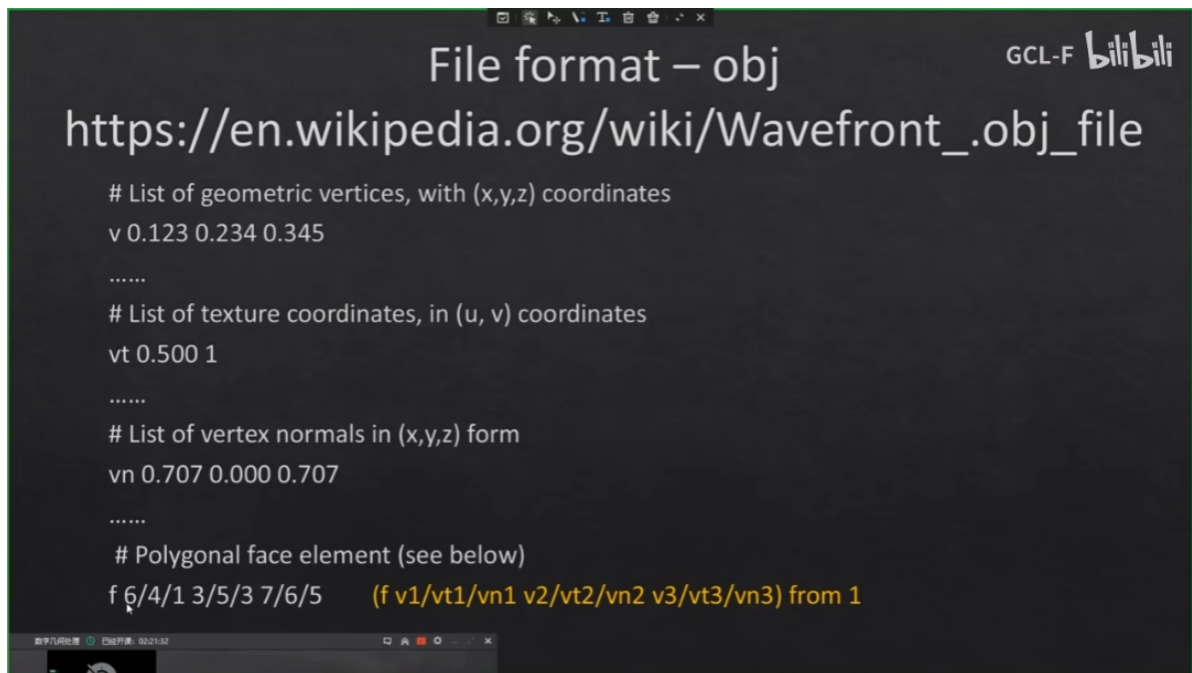


数字几何处理笔记

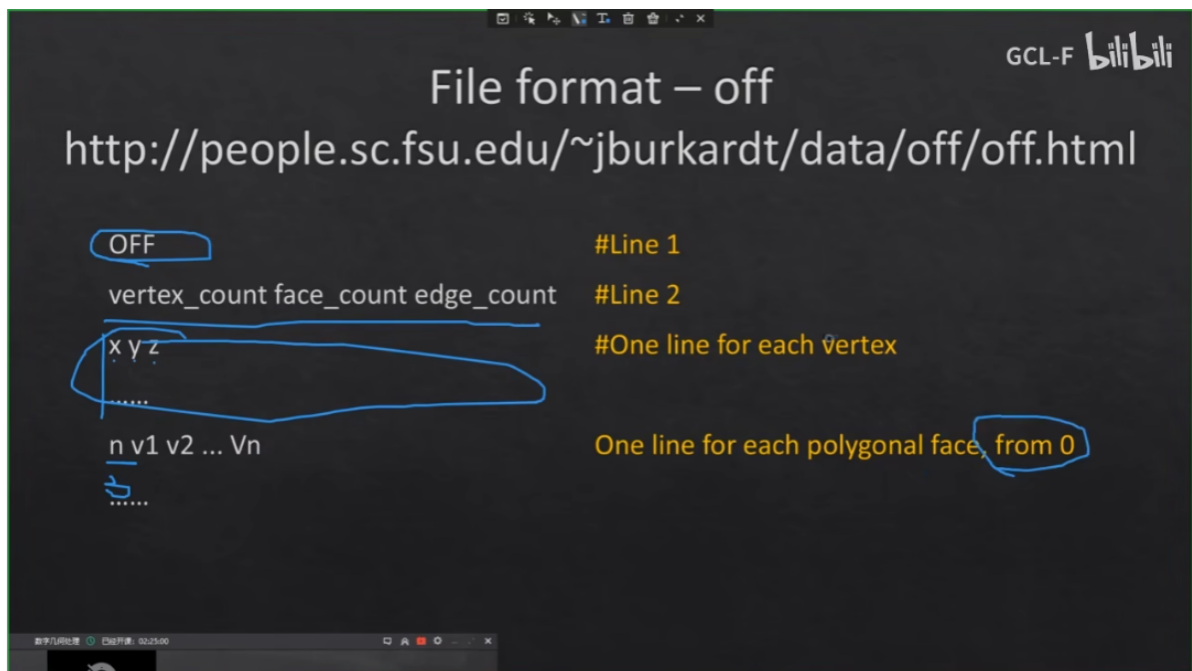
模型数据格式

- obj数据格式



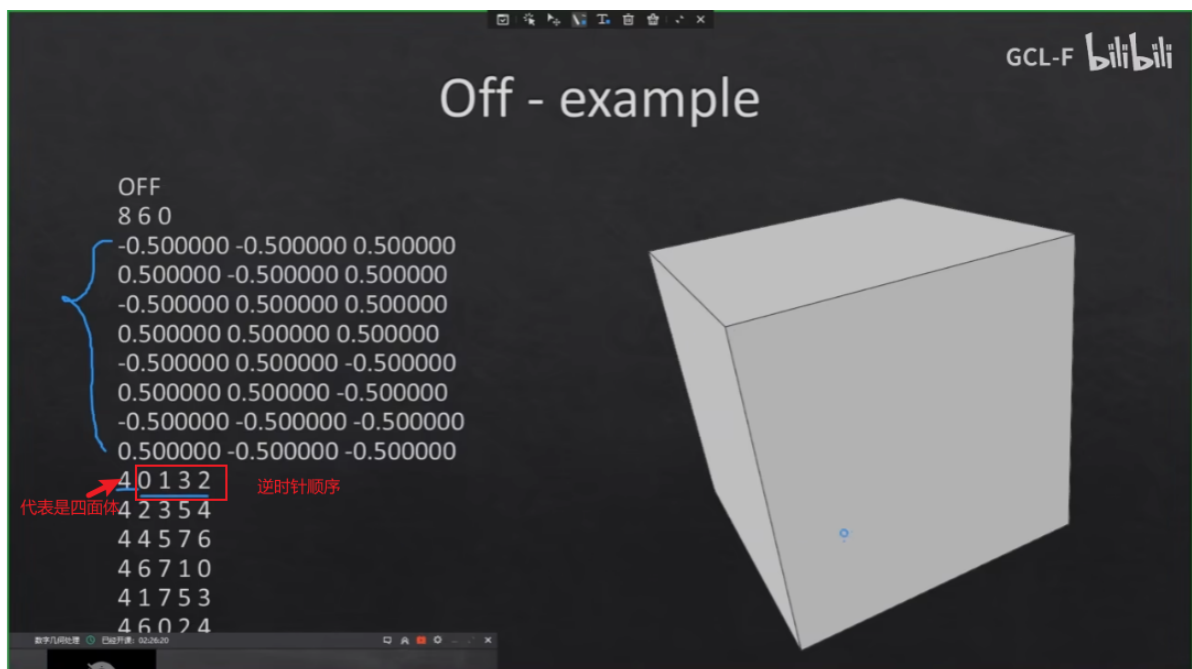
索引下标是从1开始的

- off数据格式



索引下标是从0开始的

例子：



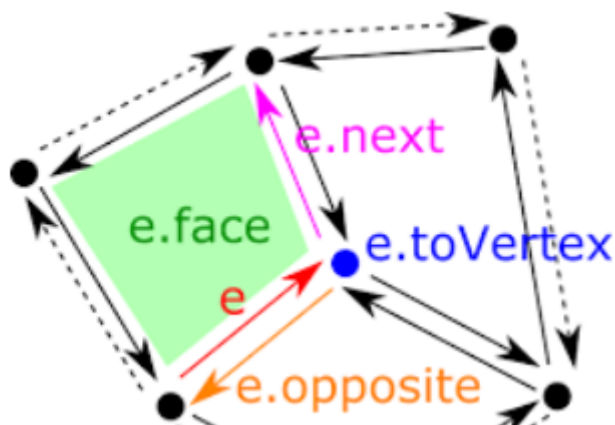
半边数据结构

半边数据结构只适用于流形网格

计算机图形学上，表达表面网格的数据结构有三种，分别是面列表、邻接矩阵、半边结构。

面列表的典型代表是.obj格式文件和Unity。其优点是，简单而且能够表达非流形网格，但网格查询和处理能力差。

当然还有一些不是利用索引，而是直接储存整个顶点的信息，典型代表是OpenGL可编程管线和.stl格式文件。



一个半边包含以上五种变量

- 半边数据结构的基本元素是固定的吗？

重心坐标

重心坐标 (Barycentric Coordinates) 是一种用来描述三角形内的点位置的坐标系统。

重心坐标通常指的是三角形内部任意点相对于三个顶点的位置关系的一种表示方法，而不是指三角形的重心。

在三角形中，任意一点都可以表示为三个顶点的加权和，这三个顶点构成了三角形的顶点。重心坐标就是用来描述这三个顶点的权重，从而表示三角形内的点。

对于给定的三角形ABC和其中的一点P，其重心坐标可以表示为 (w_A, w_B, w_C) ，其中：

- w_A 是点P相对于顶点A的权重，

- w_B 是点P相对于顶点B的权重,
- w_C 是点P相对于顶点C的权重。

重心坐标有一个重要的性质, 即三个权重的总和等于1, 即 $w_A + w_B + w_C = 1$ 。

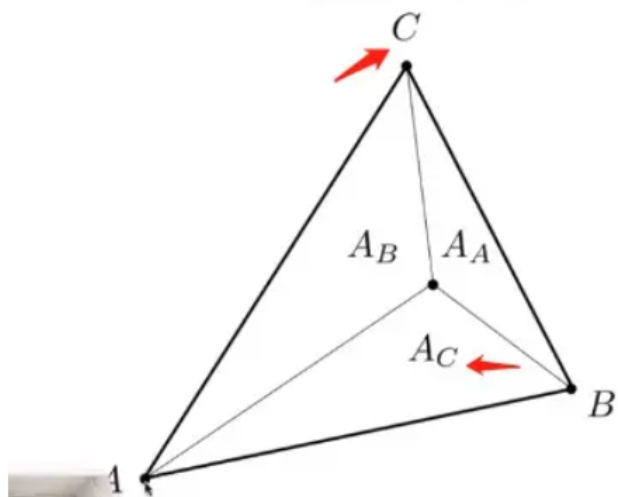
重心坐标常常用于在三角形内插值某些属性, 比如颜色、纹理坐标、法向量等。

利用重心坐标插值三角形顶点的任何属性, 注意: 如果三角形经过投影后, 重心坐标可能会改变。所以如果要插值属性, 一定要在三维空间中利用重心坐标插值, 不要投影后才插值。

- 计算重心坐标的方法

1. 面积法: 利用向量叉积求面积, 通过面积计算

Geometric viewpoint — proportional areas



$$\alpha = \frac{A_A}{A_A + A_B + A_C}$$

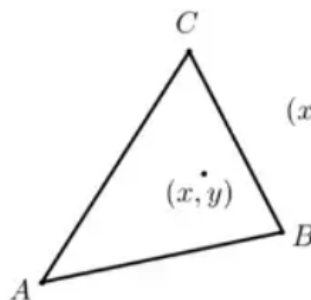
$$\beta = \frac{A_B}{A_A + A_B + A_C}$$

$$\gamma = \frac{A_C}{A_A + A_B + A_C}$$

知乎 @aganztracy

2. 公式法: 用已经推出的公式计算

Barycentric Coordinates: Formulas



$$(x, y) = \alpha A + \beta B + \gamma C$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha = \frac{-(x - x_B)(y_C - y_B) + (y - y_B)(x_C - x_B)}{-(x_A - x_B)(y_C - y_B) + (y_A - y_B)(x_C - x_B)}$$

$$\beta = \frac{-(x - x_C)(y_A - y_C) + (y - y_C)(x_A - x_C)}{-(x_B - x_C)(y_A - y_C) + (y_B - y_C)(x_A - x_C)}$$

$$\gamma = 1 - \alpha - \beta$$

知乎 @aganztracy

拉普拉斯算子

- 笛卡尔坐标系

笛卡尔坐标系是用来描述二维和三维空间中点的位置的一种坐标系统。它由数学家笛卡尔于17世纪提出, 是平面几何和立体几何研究的基础。

在二维笛卡尔坐标系中, 平面被划分为水平和垂直两个轴, 通常分别称为 x 轴和 y 轴。点的位置可以用 (x, y) 表示, 其中 x 表示点在 x 轴上的水平位置, y 表示点在 y 轴上的垂直位置。

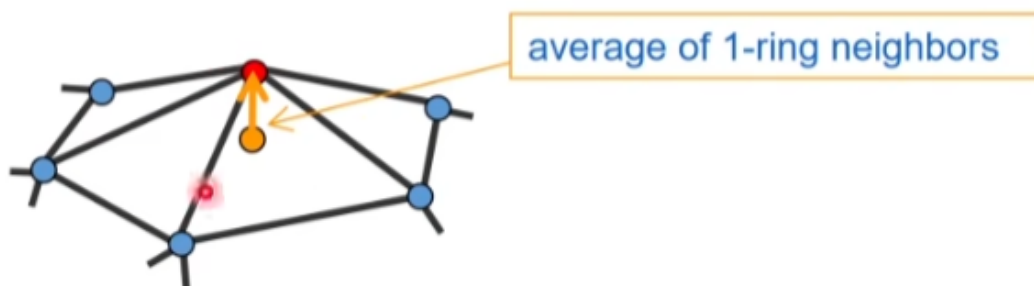
在三维笛卡尔坐标系中，除了水平和垂直两个轴外，还增加了一个垂直于平面的轴，通常称为 z 轴。点的位置可以用 (x, y, z) 表示，其中 x 表示点在 x 轴上的水平位置， y 表示点在 y 轴上的垂直位置， z 表示点在 z 轴上的垂直位置。

除了笛卡尔坐标系外，还有许多其他常用的坐标系，其中一些包括：

1. **极坐标系**：极坐标系使用极径和极角来描述点的位置。极径表示点到原点的距离，而极角表示点与某个参考方向的夹角。极坐标系在描述圆形和对称性问题时很有用。
2. **球坐标系**：球坐标系用来描述三维空间中的点，它由点到原点的距离、点与 z 轴的夹角以及点在 xy 平面上的投影的极角组成。球坐标系通常用于描述球体、球面上的点或者在球面上进行积分等问题。
3. **柱坐标系**：柱坐标系通常用于描述三维空间中的点，它由点到 z 轴的距离、点在 xy 平面上的投影的极角以及点在 xy 平面上的投影的坐标组成。柱坐标系常用于描述柱面、柱体上的点或者在柱面、柱体上进行积分等问题。
4. **贝塞尔坐标系**：贝塞尔坐标系是用来描述曲线和曲面的坐标系，它使用贝塞尔曲线或者贝塞尔曲面的参数方程来表示点的位置。贝塞尔坐标系在计算机图形学和计算机辅助设计等领域中被广泛应用。

- 拉普拉斯介绍

拉普拉斯方程^[3]是二阶偏微分方程，对于 n 维空间的标量函数 $f(x_1, x_2, \dots, x_n)$ 有 $\nabla^2 f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \dots + \frac{\partial^2 f}{\partial x_n^2} = 0$ 。为了方便，引入符号 $\Delta = \nabla \cdot \nabla = \nabla^2$ 。这里 Δ 是**拉普拉斯算子**，定义为**梯度** (gradient) 的**散度** (divergence)。 Δ 也用作微小增量符号、一元二次方程的判别式符号，不要搞混了。 $\nabla \cdot$ 是散度记号，字母符号是 div 。 ∇ 是梯度记号，字母符号记作 grad ^[4]。于是，拉普拉斯方程也可以写成 $\text{div}(\text{grad}(f)) = 0$ 。



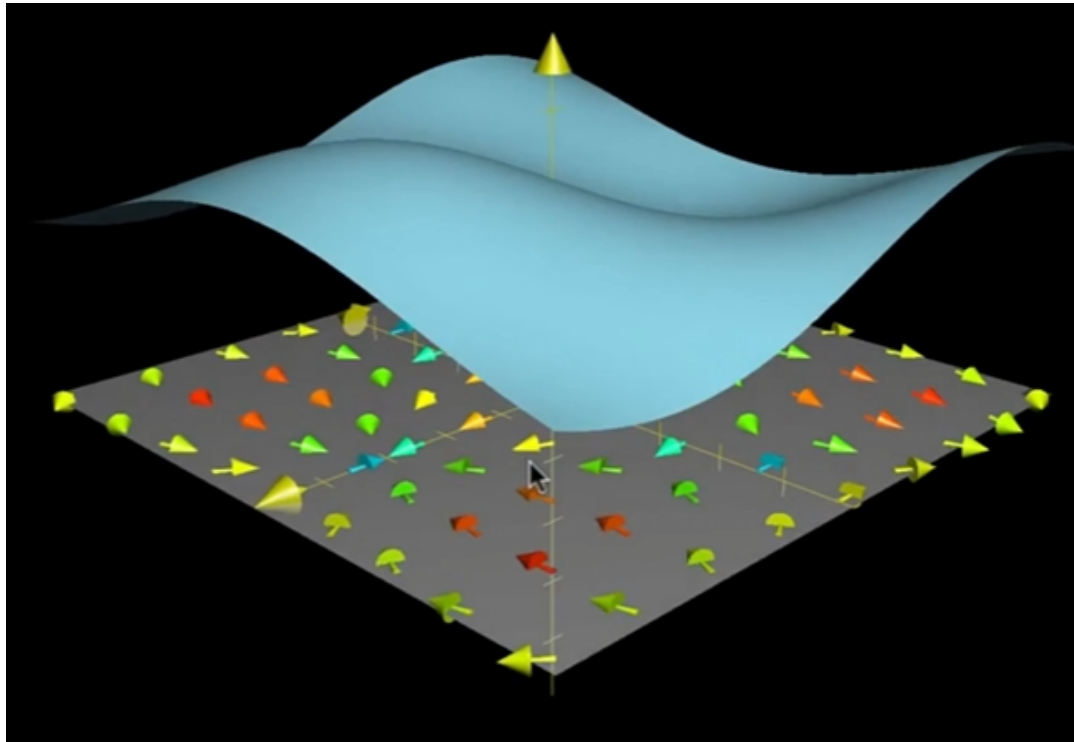
拉普拉斯直观化

- 梯度和散度

1. **梯度**：

- 在标量场中，梯度表示标量场在某一点上变化最快的方向和速率。如果有一个标量函数 $f(x, y, z)$ ，其在三维空间中的梯度记作 ∇f ，其中 ∇ 称为“nabla”或“del”操作符，表示求偏导数。梯度的方向即函数在该点上变化最快的方向，而梯度的大小表示该变化的速率。
 - 在二维情况下，梯度表示函数在平面上变化最快的方向和速率。梯度是一个二维向量，其两个分量分别对应函数在 x 和 y 方向上的偏导数。

- 例如一个二元函数 $f(x, y)$, 则 $\text{grad}f(x, y)$ 就是各点处指向函数上升最快的方向



2. 散度:

- 在向量场中, 散度描述了向量场在某一点上的流出或流入程度。如果有一个向量函数 $F(x, y, z) = (P(x, y, z), Q(x, y, z), R(x, y, z))$, 其在三维空间中的散度记作 $\nabla \cdot F$, 表示对该向量场的每个分量分别求偏导数后的和。如果散度为正, 表示向量场在该点上存在流出的趋势; 如果散度为负, 表示向量场在该点上存在流入的趋势; 如果散度为零, 表示向量场在该点上没有流动, 是一个“源”或“汇”
- 有点像二维坐标系下的二阶导, 在函数的局部极大值处小于0, 在函数的局部极小值处大于0

面的法向, 顶点的法向

- 面的法向

可以通过计算两个边向量的叉积来获得该平面的法线向量。

```

1  import numpy as np
2
3  # 三角形的三个顶点坐标
4  P1 = np.array([x1, y1, z1]) # 第一个顶点坐标
5  P2 = np.array([x2, y2, z2]) # 第二个顶点坐标
6  P3 = np.array([x3, y3, z3]) # 第三个顶点坐标
7
8  # 计算边向量
9  v1 = P2 - P1
10 v2 = P3 - P1
11
12 # 计算法线向量
13 normal = np.cross(v1, v2)
14
15 # 归一化法线向量
16 normal /= np.linalg.norm(normal)
17
18 # 打印法线向量
19 print("法线向量: ", normal)
20

```


- 顶点的法向

顶点的法向没有唯一的定义，需要借助顶点附近邻域的面积来计算。根据 α_T 取值的不同而不同

- Vertex normal: spatial averages of normal vectors in a local one-ring neighborhood.

$$\mathbf{n}(v) = \frac{\sum_{T \in \Omega(v)} \alpha_T \mathbf{n}(T)}{\|\sum_{T \in \Omega(v)} \alpha_T \mathbf{n}(T)\|_2}$$

1. constant weights: $\alpha_T = 1$
2. triangle area: $\alpha_T = \text{area}(T)$
3. incident triangle angles: $\alpha_T = \theta(T)$

- 点积、叉积的区别

点积（也称为内积）和叉积（也称为外积）是两种常见的向量运算，它们具有不同的性质和用途。

1. 点积（内积）：

- 定义：对于两个向量 $\mathbf{a} = [a_1, a_2, a_3]$ 和 $\mathbf{b} = [b_1, b_2, b_3]$ ，它们的点积定义为 $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$ 。
- 性质：点积的结果是一个标量（数量），表示了两个向量之间的夹角关系。具体地，如果两个向量正交（垂直），它们的点积为0；如果两个向量平行且同向，它们的点积为正；如果两个向量平行但反向，它们的点积为负。
- 应用：点积常用于计算向量的投影、判断向量的正交性、计算向量的夹角等。

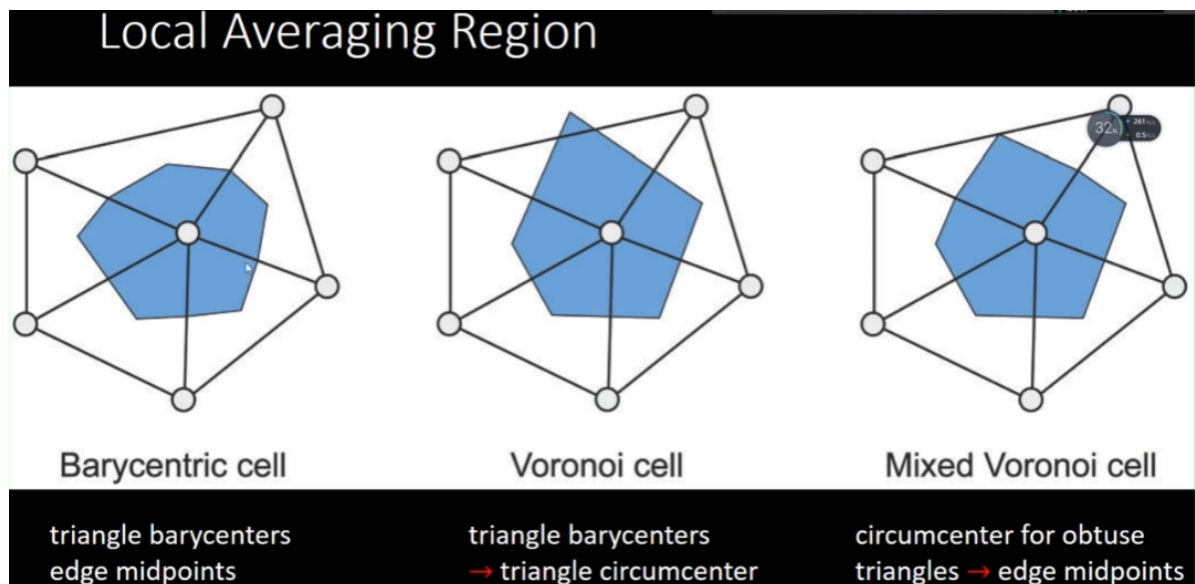
2. 叉积（外积）：

- 定义：对于两个三维向量 $\mathbf{a} = [a_1, a_2, a_3]$ 和 $\mathbf{b} = [b_1, b_2, b_3]$ ，它们的叉积定义为 $\mathbf{a} \times \mathbf{b} = [a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1]$ 。
- 性质：叉积的结果是一个向量，它垂直于参与叉积的两个向量所在的平面，并且符合右手法则。叉积的长度等于两个向量构成的平行四边形的面积，方向由右手法则确定。
- 应用：叉积常用于计算平面的法线向量、计算向量的方向、计算力矩等。

总的来说，点积主要用于衡量向量之间的夹角关系和投影，而叉积主要用于计算向量之间的垂直关系和计算与向量相关的几何量。

Local Averaging Region

计算某个顶点上的属性，用的是该顶点的邻域，求这个区域的积分后再除以这个区域的面积



- 作业（只考虑封闭曲面）【如果去噪是把高频信号去掉，那是不是相当于将要去噪的物体模糊化，就是没有细节】
- 1. 输入需要自己造：用一个干净的网格，然后随机在顶点添加扰动
 2. normal 双边滤波：确定两个高斯函数的标准差和迭代次数（或用normal变化量大小作为终止条件）
 3. 顶点位置恢复（移动）：迭代次数
 4. 在3的过程中保体积

傅里叶分析

[video](#)

• 1D Fourier Transform:

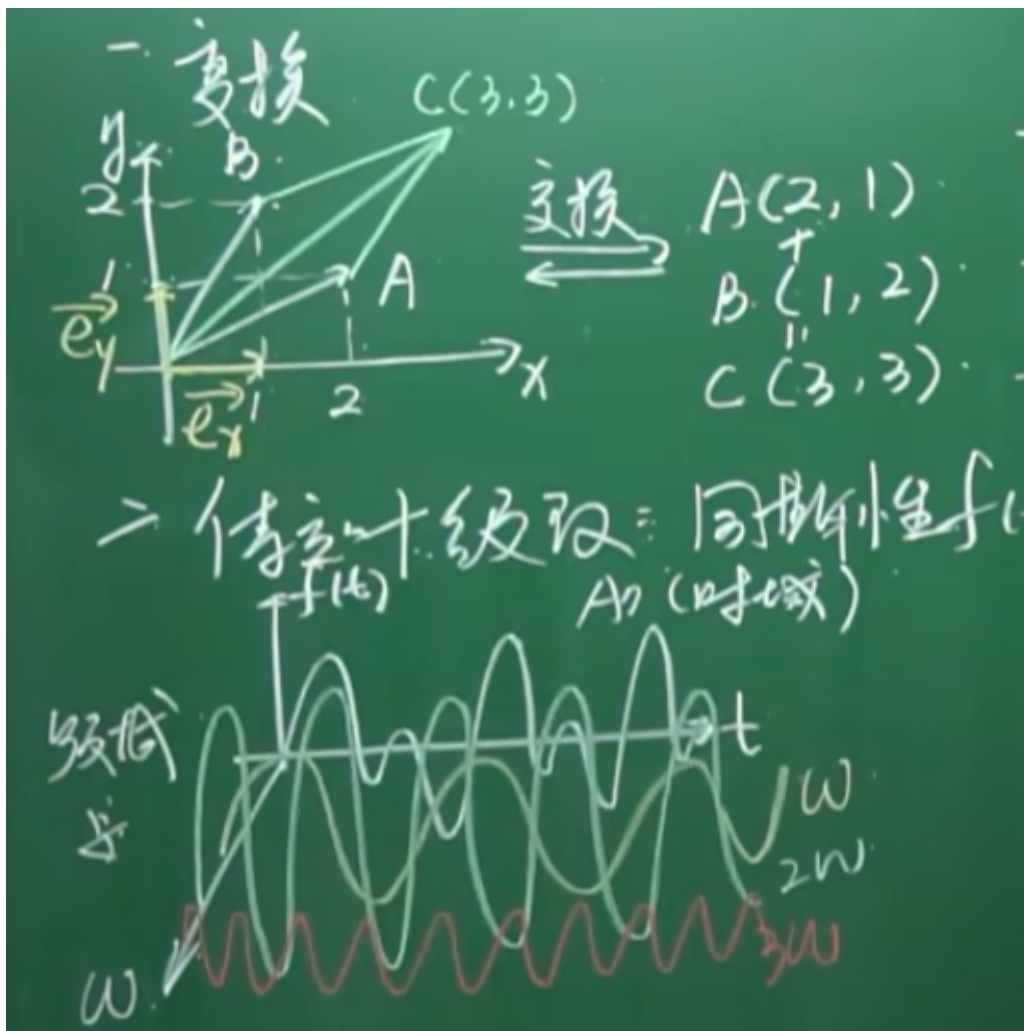
$$F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i \omega x} dx$$

$$f(x) = \int_{-\infty}^{+\infty} F(\omega) e^{2\pi i \omega x} d\omega$$

spatial domain $f(x) \Leftrightarrow$ frequency domain $F(\omega)$

空域和频域之间相互转换

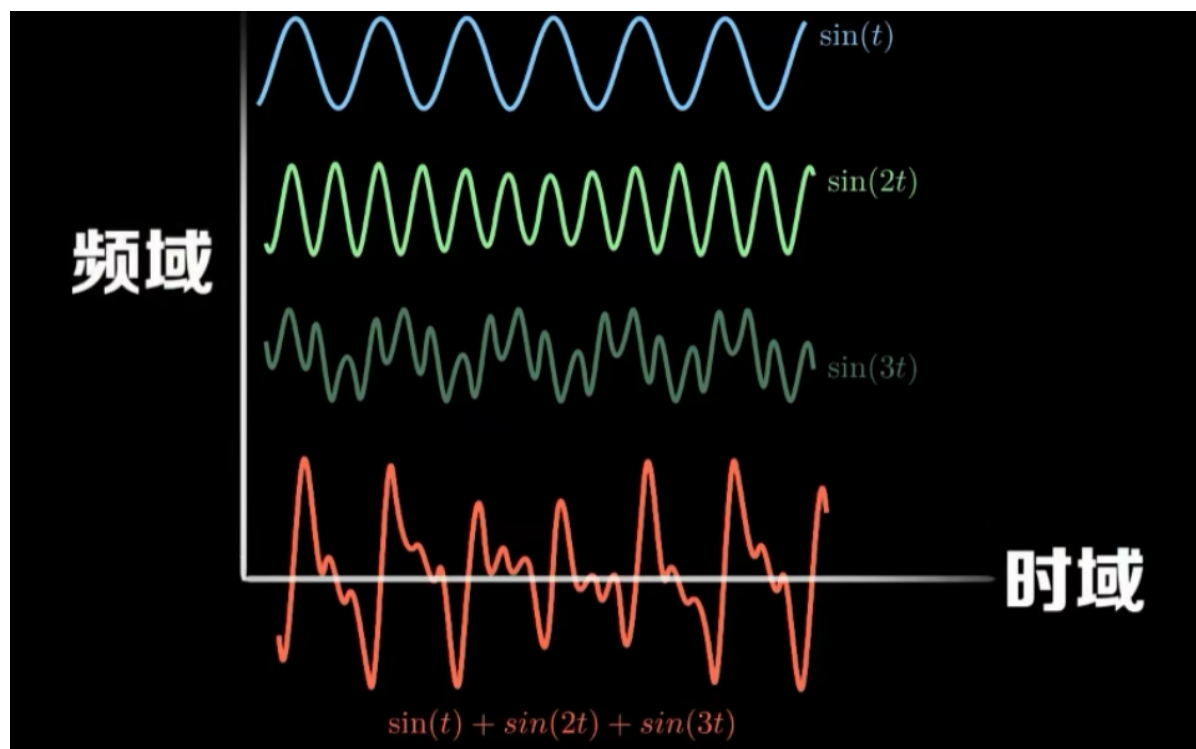
- 傅里叶级数
所有周期性的函数 $f(X)$ 等于 一大堆正余弦函数的和

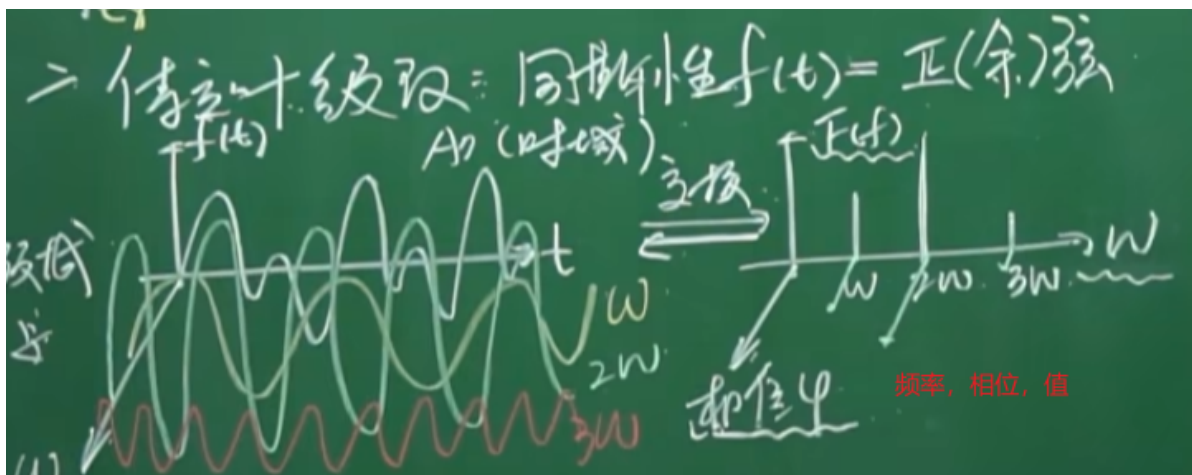
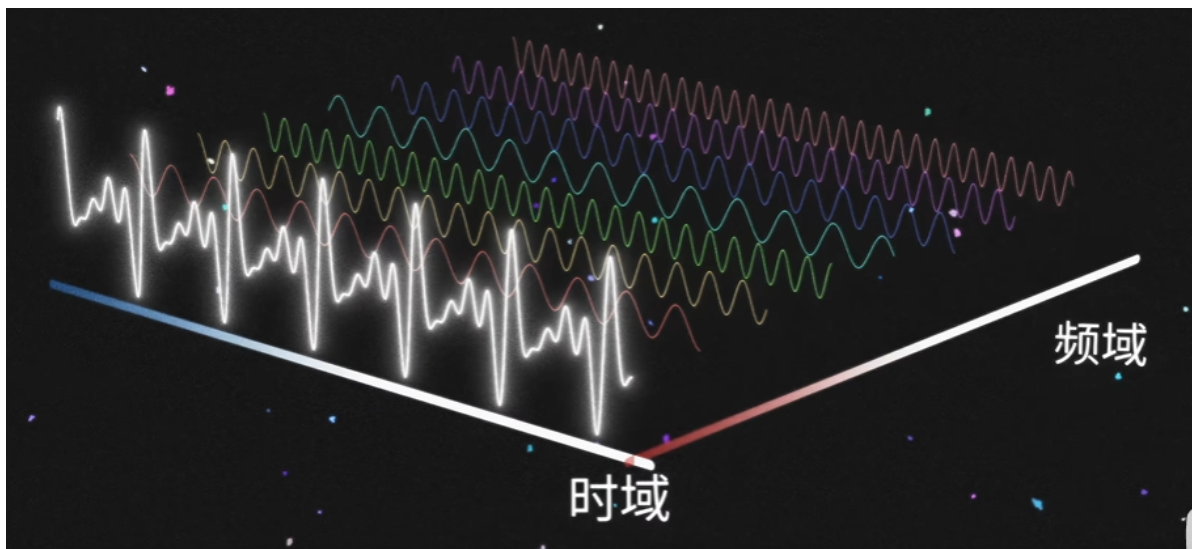


如果从指向 ω 的方向看，只能看到那条白色的线随时间的变化，所有叫做时域

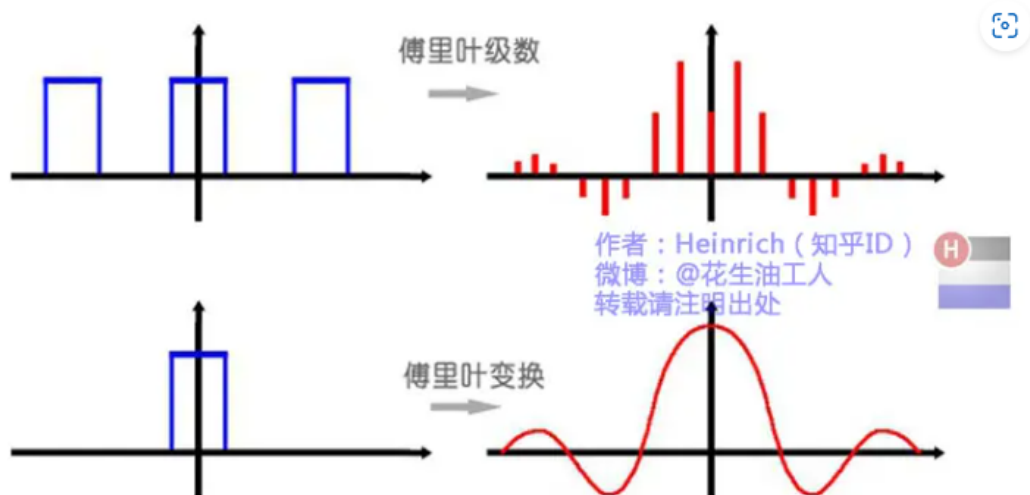
如果从指向 t 的方向看，可以看到所有的频率，所以叫做频域

信号随时间的变化称为时域，随频率的变换称为频域





- 傅里叶变换

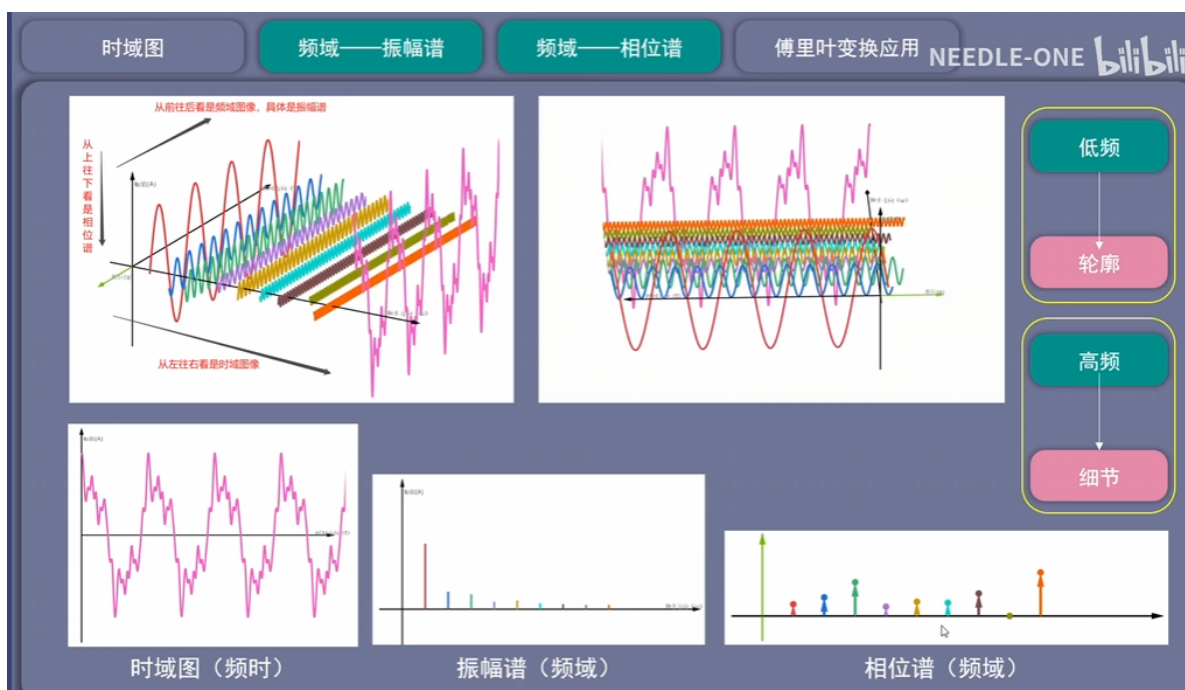
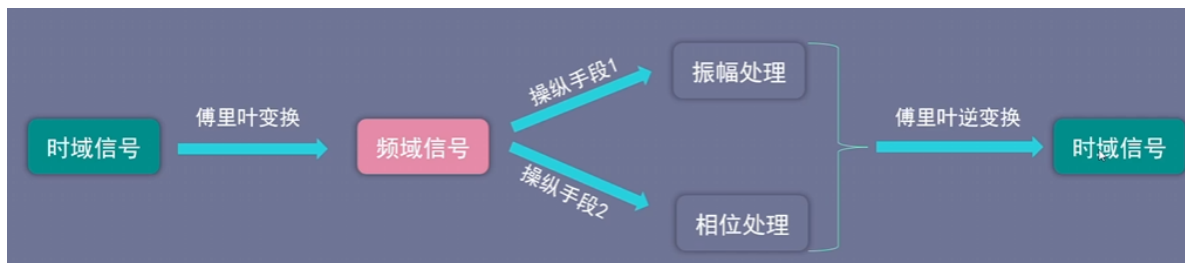


原来离散谱的叠加，变成了连续谱的累积。所以在计算上也从求和符号变成了积分符号。

将混合后的最终信号分解为单个频率

将时域信号转换为频域信号

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$



频率越低，波形越大

各向异性网格,各向同性网格

• 各向异性网格:

- 各向异性指的是在不同方向上具有不同特性或性质。在网格中，各向异性网格意味着网格单元在不同方向上的尺寸、形状或密度不同。这种网格通常用于需要更高精度或更大灵活性的场景，例如在物体的形状或特征非均匀分布的情况下，或者在模拟某些具有各向异性特性的材料行为时。
- 各向异性网格的生成通常需要特殊的算法或技术，以确保在不同方向上的特性得到合适的控制。

• 各向同性网格:

- 各向同性指的是在各个方向上具有相似的特性或性质。在网格中，各向同性网格意味着网格单元在各个方向上的尺寸、形状或密度基本相同。这种网格通常用于简化建模、计算效率高的场景，例如在规则形状的物体或需要均匀分布特性的情况下。
- 各向同性网格通常易于生成，可以使用一般性的网格生成算法来生成，而无需额外的处理。

gauss 分布

高斯分布最简单的形式是一维标准高斯分布，可以由概率密度函数(PDF)表示为

$$p(x) = \phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

其中， $\frac{1}{\sqrt{2\pi}}$ 用于保证概率密度函数的积分为 1，这个分布的中心为 $x = 0$ 且衰减率或者说分布的“宽度”为 1。

更加一般地，我们可以通过平移和伸缩得到任意中心 $\mu \in \mathbb{R}$ 和宽度 $\sigma > 0$ 的高斯分布，它的pdf为

$$p(x) = \frac{1}{\sigma} \phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

通常 μ 称为均值或者密度的peak、mode， σ^2 称为方差。如果一个随机变量 X 服从这一分布，则记作

$$X \sim \mathcal{N}(\mu, \sigma^2).$$

更进一步地，一个多元随机变量，假设为 d 维列向量 x ，其均值向量 μ ，协方差均值 Σ ，概率分布可以写为

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right),$$

- 叉乘计算三角形面积

首先， e_{12} 表示从第一个顶点到第二个顶点的向量， e_{13} 表示从第一个顶点到第三个顶点的向量。

然后，利用向量叉乘的性质，可以得到一个新的向量，其大小等于两个原向量的叉乘结果的模，方向垂直于这两个向量所在的平面。这个新向量的大小实际上就是三角形的面积的两倍。因此，我们将这个向量的模除以 2，即得到了三角形的面积。

这个计算方法是基于向量的几何性质，可以很好地用来计算三角形的面积，同时也可以推广到计算其他几何形状的面积。

```
1 | e12 = points[1] - points[0]
```