

1 几何重建

1.1 几何重建概述

现实世界中的三维物体和场景无法直接以数学模型或几何形式被计算机存储或处理. 大多数时候, 我们只能通过图像, 视频或点云数据间接地获取三维信息, 这些数据本身离散而稀疏, 带有噪声, 也没有拓扑结构. 为了用计算机处理形状, 空间结构和几何关系, 必须将这些数据重建为连续的几何模型.

定义 1.1 几何重建 几何重建 (Geometry Reconstruction) 从离散的三维数据 (如点云, 深度图像或多视图图像) 中, 重建出物体或场景的连续几何形状与结构的过程.

1.2 几何变换基础

定理 1.2 平移 平移对应的数学操作为向量加法. 设物体上任意一点 P 的坐标为 \mathbf{p} , 平移向量为 \mathbf{t} , 则平移后的点 P' 的坐标 $\mathbf{p}' = \mathbf{p} + \mathbf{t}$. 例如, 对于三维空间而言有

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \end{bmatrix}$$

定理 1.3 旋转 旋转对应的数学操作为矩阵乘法. 设物体上任意一点 P 的坐标为 \mathbf{p} , 旋转矩阵为 \mathbf{R} , 则旋转后的点 P' 的坐标 $\mathbf{p}' = \mathbf{R}\mathbf{p}$. 在二维平面中, 绕原点逆时针旋转 θ 角度的旋转矩阵为

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

在三维空间中, 绕 x, y, z 轴分别旋转 α, β, γ 角度的旋转矩阵分别为

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad \mathbf{R}_y = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad \mathbf{R}_z = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

定理 1.4 绕任意轴的旋转 在三维空间中, 绕单位向量 $\mathbf{a} = (a_x, a_y, a_z)$ 定义的轴逆时针旋转 θ 角度的旋转矩阵为

1.3 点云的配准/注册

在几何重建中, 我们往往会从不同角度和位置对同一场景或物体分别采集点云数据. 将不同视角下的点云对齐到同一坐标系下, 才能得到完整的重建数据.

定义 1.5 点云配准/注册 点云配准/注册 (Point Cloud Registration) 是将来自不同视角或传感器的多个点云数据集对齐到同一坐标系下的过程.

现在我们来介绍一种经典的点云配准算法, 即 ICP(Iterative Closest Point) 算法.

1.3.1 点云配准的 ICP 算法

我们在不同角度测得的点云总是保长度的, 这意味着描述同一物体的点云之间总是可以通过平移和旋转变换对齐.

给定待变换点云 S 和目标点云 T , ICP 算法的目标是找到一组最佳的旋转和平移变换 \mathbf{R}, \mathbf{t} 使得 S 经变换后最大程度上地与 T 对齐. ICP 算法的具体流程如下:

1. 通过主成分分析 (PCA, Principal Component Analysis) 初始化一组变换 \mathbf{R}, \mathbf{t} 作为估计变换的初值.
2. 将当前求得的变换 \mathbf{R}, \mathbf{t} 应用于 S , 对于 S 中的各点 \mathbf{p}_i 变换后得到 $\mathbf{p}'_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}$. 然后找到 T 中与 \mathbf{p}'_i 中最接近的点 \mathbf{q}_i , 如此构成 S' 与 T 的一一对应. 同时, 舍去距离过远的点对.
3. 在余下的 N 组点对 $\{(\mathbf{p}'_i, \mathbf{q}_i) : i = 1, \dots, N\}$ 中, 构造累计误差函数

$$E = \sum_{i=1}^N \|\mathbf{p}'_i - \mathbf{q}_i\|^2 = \sum_{i=1}^N \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2$$

注意这里的 \mathbf{R}, \mathbf{t} 是需要求解的变量, 与 2. 中前一步迭代求得的 \mathbf{R}, \mathbf{t} 是不同的. 前一步求得的 \mathbf{R}, \mathbf{t} 只用于对应关系的确定, 不直接参与后续的最小化计算.

然后, 通过奇异值分解 (SVD, Singular value decomposition) 求解最小化问题:

$$(\mathbf{R}, \mathbf{t}) = \arg \min_{\mathbf{R}, \mathbf{t}} E$$

4. 重复步骤 2 和 3, 直到累计误差 E 小于预设阈值或达到最大迭代次数.

现在我们具体地介绍上述算法中重要的两步, 即 PCA 和 SVD.

1.3.2 主成分分析

定义 1.6 主成分分析 主成分分析 (Principal Component Analysis, PCA) 是一种统计方法, 用于通过线性变换将数据从高维空间映射到低维空间, 以保留数据的主要特征和结构.

一般而言, 点云的分布在空间中是各向异性的, 即在某些方向上点云的变化更显著. 通过 PCA 可以找到点云数据的主要方向 (即主轴), 从而为 ICP 算法提供一个合理的初始变换估计. 现在给出 PCA 的具体步骤.

首先, 考虑给定的一组点¹ $\mathbf{p}_1, \dots, \mathbf{p}_N$ 以及中心坐标

$$\bar{\mathbf{p}} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i$$

首先, 为了防止平移对 PCA 结果的影响, 我们需要将点云数据进行中心化处理, 即将每个点减去中心坐标:

$$\mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}}, \quad i = 1, \dots, N$$

构造矩阵

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \vdots \\ \mathbf{p}'_N \end{bmatrix}$$

然后计算 \mathbf{P} 的协方差矩阵

$$\Sigma_P = \mathbf{P}\mathbf{P}^t$$

二维情况下的协方差矩阵的两个特征向量 $\mathbf{v}_1, \mathbf{v}_2$ 表示数据的两条轴线. 对应于更大的特征值的特征向量 \mathbf{v}_1 表示数据变化更显著的方向, 即主成分方向, 而 \mathbf{v}_2 则表示数据变化最小的方向². 对于三维情况也是同理.

对协方差矩阵 Σ_P 进行特征值分解, 可得:

$$\Sigma_P = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t$$

其中矩阵 \mathbf{V} 的列向量即为 Σ_P 的特征向量, 它们也就确定了点云的主轴方向.

现在, 对给定的两个点云 S 和 T , 我们分别对它们进行 PCA, 得到各自的中心 $\bar{\mathbf{p}}, \bar{\mathbf{q}}$ 以及主轴 $\mathbf{v}_{s1}, \mathbf{v}_{s2}, \mathbf{v}_{s3}$ 和 $\mathbf{v}_{t1}, \mathbf{v}_{t2}, \mathbf{v}_{t3}$. 既然变换仅由平移和旋转构成, 我们可以认为 S 和 T 的主轴应当近似地重合. 于是, 先通过平移使得两个点云的中心重合, 即

$$\mathbf{t} = \bar{\mathbf{q}} - \bar{\mathbf{p}}$$

然后, 通过旋转使得两个点云的主轴方向对齐. 设旋转矩阵为 \mathbf{R} , 则有

$$\mathbf{R}\mathbf{s}_i = \mathbf{t}_i, \quad i = 1, 2, 3$$

这样求得的 \mathbf{R}, \mathbf{t} 即可作为 ICP 算法的初始变换估计.

1.3.3 奇异值分解

对于前述的最小化问题, 事实上是具有唯一解的. 我们可以用奇异值分解来求解.

注意到 E 中含有两个变量 \mathbf{R}, \mathbf{t} . 我们先求解使得 E 最小的 \mathbf{t} , 然后再求解 \mathbf{R} . 这是为了对齐质心后才能进

¹每个点都是一个三维向量, 表示该点在三维空间中的位置.

²例如, 如果数据点沿一条直线附近随机分布, 那么 \mathbf{v}_1 就是该直线的法向量, \mathbf{v}_2 是该直线的方向向量

行旋转矩阵的求解.

设 S 和 T 的中心分别为 $\bar{\mathbf{p}}, \bar{\mathbf{q}}$. 令

$$\tilde{\mathbf{p}}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \quad \tilde{\mathbf{q}}_i = \mathbf{q}_i - \bar{\mathbf{q}}$$

于是

$$E = \sum_{i=1}^N \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2 = \sum_{i=1}^N \|\mathbf{R}\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i + (\mathbf{R}\bar{\mathbf{p}} + \mathbf{t} - \bar{\mathbf{q}})\|^2$$

关于 $\mathbf{R}\bar{\mathbf{p}} + \mathbf{t} - \bar{\mathbf{q}}$ 展开后对 \mathbf{t} 求导并令其为 0, 可得 (实际上直接观察也容易得到)

$$\mathbf{R}\bar{\mathbf{p}} + \mathbf{t} - \bar{\mathbf{q}} = \mathbf{0}$$

于是

$$\mathbf{t} = \bar{\mathbf{q}} - \mathbf{R}\bar{\mathbf{p}}$$

注意这里的 \mathbf{t} 是关于 \mathbf{R} 的函数, 因此求解 \mathbf{R} 完毕后需要重新进行代入. 现在把 \mathbf{t} 代入 E , 可得

$$E = \sum_{i=1}^N \|\mathbf{R}\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i\|^2$$

上面的式子中已经没有 \mathbf{t} , 即已经完成对齐质心的工作. 我们只需要最小化 E 关于 \mathbf{R} 的部分. 将平方展开, 并且注意到 $\|\mathbf{R}\tilde{\mathbf{p}}_i\| = \|\tilde{\mathbf{p}}_i\|$ (旋转显然不改变向量的模长), 于是可得

$$E = \sum_{i=1}^N \left(\|\tilde{\mathbf{p}}_i\|^2 + \|\tilde{\mathbf{q}}_i\|^2 \right) - 2 \sum_{i=1}^N \tilde{\mathbf{q}}_i^t \mathbf{R} \tilde{\mathbf{p}}_i$$

注意到上式中只有最后一项与 \mathbf{R} 有关, 因此等价于最大化

$$\mathbf{R}^* = \arg \max_{\mathbf{R}} \sum_{i=1}^N \tilde{\mathbf{q}}_i^t \mathbf{R} \tilde{\mathbf{p}}_i$$

自然地, 我们会想到标量与矩阵的迹之间的关联. 由于每个求和项都是向量的内积, 可以视作 1×1 的矩阵, 于是自然可以将它写做迹的形式:

$$\tilde{\mathbf{q}}_i^t \mathbf{R} \tilde{\mathbf{p}}_i = \text{trace}(\tilde{\mathbf{q}}_i^t \mathbf{R} \tilde{\mathbf{p}}_i) = \text{trace}(\tilde{\mathbf{p}}_i \tilde{\mathbf{q}}_i^t \mathbf{R})$$

于是求和后就有

$$\sum_{i=1}^N \tilde{\mathbf{q}}_i^t \mathbf{R} \tilde{\mathbf{p}}_i = \text{trace} \left(\sum_{i=1}^N \tilde{\mathbf{p}}_i \tilde{\mathbf{q}}_i^t \mathbf{R} \right) = \text{trace} \left(\mathbf{R} \sum_{i=1}^N \tilde{\mathbf{p}}_i \tilde{\mathbf{q}}_i^t \right)$$

为了写成矩阵形式, 定义

$$\mathbf{M} = \sum_{i=1}^N \tilde{\mathbf{p}}_i \tilde{\mathbf{q}}_i^t$$

这是一个 3×3 的矩阵, 对其进行奇异值分解, 可得

$$\mathbf{M} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^t$$

其中 \mathbf{U}, \mathbf{V} 是正交矩阵, $\mathbf{\Sigma}$ 是对角矩阵, 其对角线元素为非负的奇异值. 于是

$$\text{trace} \left(\mathbf{R} \sum_{i=1}^N \tilde{\mathbf{p}}_i \tilde{\mathbf{q}}_i^t \right) = \text{trace}(\mathbf{R}\mathbf{M}) = \text{trace}(\mathbf{R}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^t) = \text{trace}(\mathbf{\Sigma}\mathbf{V}^t\mathbf{R}\mathbf{U})$$

由于 \mathbf{R} 是旋转矩阵, 因此它自然也是正交矩阵. 于是设 $\mathbf{X} = \mathbf{V}^t\mathbf{R}\mathbf{U}$, 则 \mathbf{X} 也是正交矩阵. 于是

$$\text{trace}(\mathbf{\Sigma}\mathbf{V}^t\mathbf{R}\mathbf{U}) = \text{trace}(\mathbf{\Sigma}\mathbf{X}) = \sigma_1 x_{11} + \sigma_2 x_{22} + \sigma_3 x_{33}$$

其中 σ_i 为 \mathbf{M} 的奇异值 (即 $\mathbf{\Sigma}$ 的对角线元素), x_{ii} 为 \mathbf{X} 的对角线元素. 由于 \mathbf{X} 是正交矩阵, 因此其元素的绝对值均不大于 1, 即 $|x_{ii}| \leq 1$. 为了最大化上式, 显然需要 $x_{ii} = 1$. 这就要求 $\mathbf{X} = \mathbf{I}$, 即

$$\mathbf{V}^t\mathbf{R}\mathbf{U} = \mathbf{I} \Rightarrow \mathbf{R} = \mathbf{V}\mathbf{U}^t$$

最后, 还需确保 \mathbf{R} 是合法的旋转矩阵, 即要求 $\det \mathbf{R} = 1$. 如果 $\det \mathbf{R} = -1$, 这样的 \mathbf{R} 表示了一个反射变换. 这里直接给出修正后的结果 (这涉及到一些线性代数知识):

$$\mathbf{R}^* = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det \mathbf{V}\mathbf{U}^t \end{bmatrix} = \mathbf{U}^t$$

当然, 在课程中并没有讲到这种情况, 因此有

$$\mathbf{R}^* = \mathbf{V}\mathbf{U}^t$$

回代求出 \mathbf{t}^* , 有

$$\mathbf{t}^* = \bar{\mathbf{q}} - \mathbf{R}^* \bar{\mathbf{p}}$$

这样得到的 $\mathbf{R}^*, \mathbf{t}^*$ 即可作为下一轮迭代用于确定对应关系的变换.