Iterative Closest Point（ICP）匹配算法

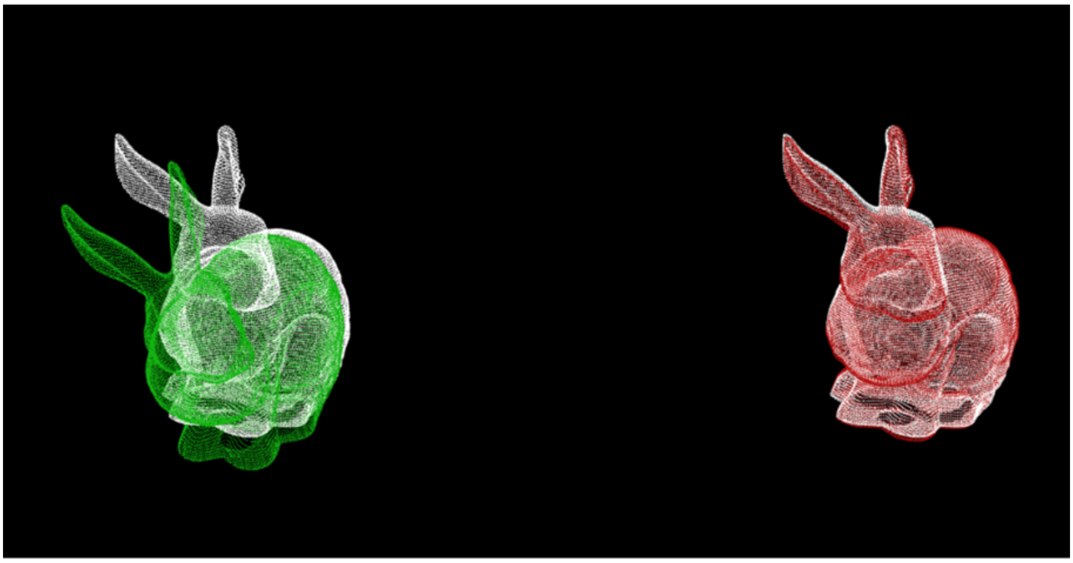
算法功能简介

- ICP（Iterative Closest Point，迭代最近点）算法是一种点集-点集的匹配方法，基于最小二乘法，计算一个匹配变换以使得点集与点集尽量重叠。
- ICP算法的输入为两帧待匹配的点云，数据格式为[x, y, z]，即点云的三维坐标。
- 重要的接口函数：
    - 【输入】`void pcl::IterativeClosestPoint::setInputCloud(const PointCloudSourceConstPtr& cloud)`

      设置源点云（被变换以进行匹配的点云），在匹配过程中不断迭代变换位置。需要的数据格式为[x, y, z]，即点云的三维坐标。
    - 【输入】`void pcl::IterativeClosestPoint::setInputTarget(const PointCloudTargetConstPtr& cloud)`

      设置目标点云（通常是地图点云），在匹配过程中保持不变。需要的数据格式为[x, y, z]，即点云的三维坐标。
    - 【输出】`void pcl::IterativeClosestPoint::align(PointCloudSource& output)`

      进行ICP匹配，参数返回匹配后的源点云。产生的数据格式为[x, y, z]，即点云的三维坐标。

      
    - 【可选输入】`void setMaximumIterations(int nr_iterations)`

      设置最大迭代次数。
    - 【可选输入】`void setMaxCorrespondenceDistance(double distance_threshold)`

      设置搜寻最近点时的搜寻范围半径。
    - 【可选输入】`void setTransformationEpsilon(double epsilon)`

      设置判断收敛的迭代步长阈值。
    - 【可选输入】`void setEuclideanFitnessEpsilon(double epsilon)`

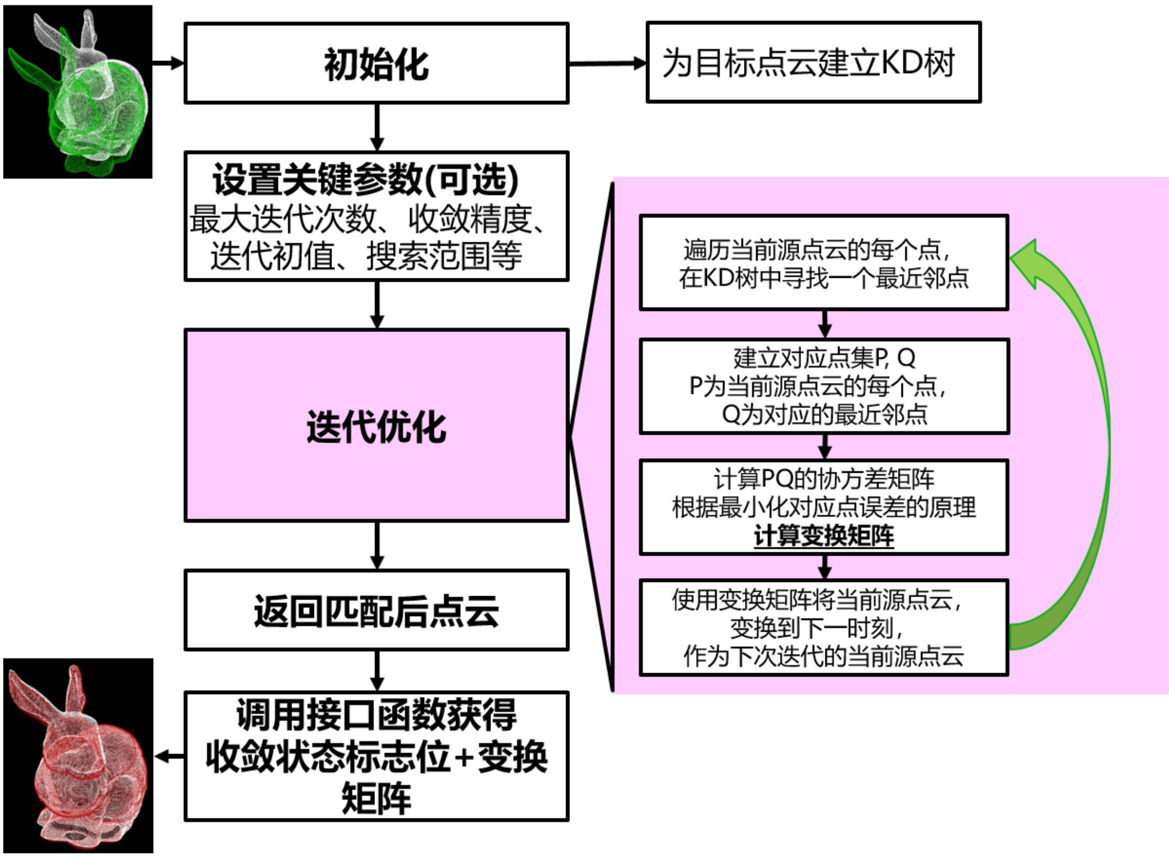
      设置判断收敛的误差精度阈值。
    - 【可选输出】`double getFitnessScore(double max_range = std::numeric_limits<double>::max());`

函数返回两帧点云匹配的均方误差值。

- 【可选输出】`Matrix4 getFinalTransformation()`

  函数返回两帧点云匹配使用的4*4变换矩阵。

算法计算流程



todo：KD树的建立算法流程。

其中，计算变换矩阵的具体数学推导过程如下。

> 关于矩阵求解数学推导：https://zhuanlan.zhihu.com/p/107218828

两组对应的点集:

$$
\begin{aligned}
P &= \{p_1, p_2, \ldots, p_n\} \\
Q &= \{q_1, q_2, \ldots, q_n\}
\end{aligned}
$$

求欧式变换 R, t 使得:

$$\forall i, \quad q_i = Rp_i + t$$

ICP 算法基于最小二乘法进行迭代计算，使得误差平方和达到极小值:

$$\underset{R,t}{\operatorname{argmin}} \sum_{i=1}^{n} \|(Rp_i + t) - q_i\|^2$$

（1）求解 translation

$$F(\mathbf{t}) = \sum_{i=1}^{n} \|(R\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2$$

$F(\mathbf{t})$ 对 t 求偏导，可得：

$$\frac{\partial F}{\partial \mathbf{t}}=\sum_{i=1}^{n} 2\left(R \mathbf{p}_{i}+\mathbf{t}-\mathbf{q}_{i}\right)=2 R \sum_{i=1}^{n} \mathbf{p}_{i}+2 \mathbf{t} \cdot n-2 \sum_{i=1}^{n} \mathbf{q}_{i}$$

令 $\frac{\partial F}{\partial \mathbf{t}}=0$，求得：

$$\mathbf{t}=\frac{1}{n} \sum_{i=1}^{n} \mathbf{q}_{i}-R \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_{i}$$

定义质心：

$$\begin{aligned}
\bar{p} &=\frac{1}{n} \sum_{i=1}^{n} p_{i} \\
\bar{q} &=\frac{1}{n} \sum_{i=1}^{n} q_{i}
\end{aligned}$$

因而，

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

将 t 代入 $F(\mathbf{t})$ 可得：

$$\begin{aligned}
\sum_{i=1}^{n}\left\|\left(R \mathbf{p}_{i}+\mathbf{t}\right)-\mathbf{q}_{i}\right\|^{2} &=\sum_{i=1}^{n}\left\|R \mathbf{p}_{i}+\overline{\mathbf{q}}-R \overline{\mathbf{p}}-\mathbf{q}_{i}\right\|^{2} \\
&=\sum_{i=1}^{n}\left\|R\left(\mathbf{p}_{i}-\overline{\mathbf{p}}\right)-\left(\mathbf{q}_{i}-\overline{\mathbf{q}}\right)\right\|^{2}
\end{aligned}$$

定义去质心坐标：

$$\begin{aligned}
x_{i} &=\mathbf{p}_{i}-\overline{\mathbf{p}} \\
y_{i} &=\mathbf{q}_{i}-\overline{\mathbf{q}}
\end{aligned}$$

因此，目标函数变为：

$$\underset{R}{\operatorname{argmin}} \sum_{i=1}^{n}\left\|R \mathbf{x}_{i}-\mathbf{y}_{i}\right\|^{2}$$

（2）求解 rotation

$$\begin{aligned}
\left\|R \mathbf{x}_{i}-\mathbf{y}_{i}\right\|^{2} &=\left(R \mathbf{x}_{i}-\mathbf{y}_{i}\right)^{\top}\left(R \mathbf{x}_{i}-\mathbf{y}_{i}\right) \\
&=\left(\mathbf{x}_{i}^{\top} R^{\top}-\mathbf{y}_{i}^{\top}\right)\left(R \mathbf{x}_{i}-\mathbf{y}_{i}\right) \\
&=\mathbf{x}_{i}^{\top} R^{\top} R \mathbf{x}_{i}-\mathbf{y}_{i}^{\top} R \mathbf{x}_{i}-\mathbf{x}_{i}^{\top} R^{\top} \mathbf{y}_{i}+\mathbf{y}_{i}^{\top} \mathbf{y}_{i} \\
&=\mathbf{x}_{i}^{\top} \mathbf{x}_{i}-\mathbf{y}_{i}^{\top} R \mathbf{x}_{i}-\mathbf{x}_{i}^{\top} R^{\top} \mathbf{y}_{i}+\mathbf{y}_{i}^{\top} \mathbf{y}_{i}
\end{aligned}$$

$\mathbf{x}_{i}^{\top} R^{\top} \mathbf{y}_{i}$ 是标量，对于任意标量 a，都满足 $a=a^{T}$，因此，

$$\mathbf{x}_i^\top R^\top \mathbf{y}_i = \left(\mathbf{x}_i^\top R^\top \mathbf{y}_i\right)^\top = \mathbf{y}_i^\top R \mathbf{x}_i$$

$$\|R\mathbf{x}_i - \mathbf{y}_i\|^2 = \mathbf{x}_i^\top \mathbf{x}_i - 2\mathbf{y}_i^\top R\mathbf{x}_i + \mathbf{y}_i^\top \mathbf{y}_i$$

将其代入目标函数，可得：

$$\begin{aligned}
\underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} \|R\mathbf{x}_i - \mathbf{y}_i\|^2 &= \underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{x}_i^\top \mathbf{x}_i - 2\mathbf{y}_i^\top R\mathbf{x}_i + \mathbf{y}_i^\top \mathbf{y}_i\right) \\
&= \underset{R}{\operatorname{argmin}} \left(\sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{x}_i - 2\sum_{i=1}^{n} \mathbf{y}_i^\top R\mathbf{x}_i + \sum_{i=1}^{n} \mathbf{y}_i^\top \mathbf{y}_i\right) \\
&= \underset{R}{\operatorname{argmin}} \left(-2\sum_{i=1}^{n} \mathbf{y}_i^\top R\mathbf{x}_i\right)
\end{aligned}$$

因而，

$$\underset{R}{\operatorname{argmin}} \left(-2\sum_{i=1}^{n} \mathbf{y}_i^\top R\mathbf{x}_i\right) = \underset{R}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbf{y}_i^\top R\mathbf{x}_i$$

$$\begin{bmatrix} — & \mathrm{y}_1^\top & — \\ — & \mathrm{y}_2^\top & — \\ & \vdots & \\ — & \mathrm{y}_n^\top & — \end{bmatrix} \begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathrm{x}_1 & \mathrm{x}_2 & \ldots & \mathrm{x}_n \\ | & | & & | \end{bmatrix} =$$

$$= \begin{bmatrix} — & \mathrm{y}_1^\top & — \\ — & \mathrm{y}_2^\top & — \\ & \vdots & \\ — & \mathrm{y}_n^\top & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ R\mathrm{x}_1 & R\mathrm{x}_2 & \ldots & R\mathrm{x}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \mathrm{y}_1^\top R\mathrm{x}_1 & & & * \\ & \mathrm{y}_2^\top R\mathrm{x}_2 & & \\ & & \ddots & \\ * & & & \mathrm{y}_n^\top R\mathrm{x}_n \end{bmatrix}$$

由上图可知，

$$\sum_{i=1}^{n} \mathbf{y}_i^\top R\mathbf{x}_i = \operatorname{tr}\left(Y^\top RX\right) = \operatorname{tr}\left(RXY^\top\right)$$

其中，X 和 Y 为 $3 \times n$ 维矩阵。

定义协方差矩阵 $S = XY^\top$，对 S 做 **SVD 分解**：

$$S = U\Sigma V^\top$$

因而，求解问题变为使下式最大化：

$$tr\left(RXY^\top\right) = tr(RS) = tr\left(RU\Sigma V^\top\right) = tr\left(\Sigma V^\top RU\right)$$

令 $M = V^\top RU$，由于满足$MM^\top = E$，故M是正交阵，其列向量 m_j 是 orthonormal vectors，即 $m_j^\top m_j = 1$。因此，对M 的所有元素都有 $m_{ij} \leq 1$。

$$
\mathrm{tr}(\Sigma M)=\begin{pmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_d\end{pmatrix}\begin{pmatrix}m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ m_{d1} & m_{d2} & \cdots & m_{dd}\end{pmatrix}=\sum_{i=1}^{d}\sigma_i m_{ii}\le\sum_{i=1}^{d}\sigma_i
$$

当 $m_{ii}=1$ 时，$\mathrm{tr}(\Sigma M)$ 最大；M 又是正交阵，因此 M 必为单位阵。

$$
I=M=V^\top RU\Rightarrow R=VU^\top
$$

（3）SVD计算方法

$$
S=U\Sigma V^\top
$$

1. 计算 SS^T 和 S^TS；
2. 分别计算 SS^T 和 S^TS 的特征向量及其特征值；
3. SS^T 的特征向量组成 U；而 S^TS 的特征向量组成 V；
4. 对 SS^T 和 S^TS 的非零特征值求平方根，对应上述特征向量的位置，填入 Σ 的对角元。

样例效果展示

白色代表目标点云（通常为地图点云）。

红色代表本次迭代前的源点云，绿线代表源点云及其在目标点云中找到的对应点。

蓝色代表本次迭代后的源点云，作为下次迭代前的输入。