二维空间中基于虚拟坐标系和重心坐标的线性 SLAM 后端优化算法

First A. Author*. Second B. Author, Jr.**
Third C. Author***

**National Institute of Standards and Technology, Boulder, CO 80305 USA (Tel: 303-555-5555; e-mail: author@ boulder.nist.gov).*
***Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar. colostate.edu)*
**** Electrical Engineering Department, Seoul National University, Seoul, Korea, (e-mail: author@snu.ac.kr)}*

Abstract: xx.

Keywords: Registration, Scan-matching, Lidar, Transformation.

1. INTRODUCTION

results.

2. 二维空间中的重心坐标

重心坐标是用于描述一个点相对其他位置的几何概念；设三个已知点j,k,l在 2D 空间的欧式坐标为$p_j,p_k,p_l\in R^2$；设点 i 的坐标满足如下方程；

$$p_i=a_{ij}p_j+a_{ik}p_k+a_{il}p_l$$

$$a_{ij}+a_{ik}+a_{il}=1$$

那么$\{a_{ij},a_{ik},a_{il}\}$是节点 i 相对节点i,j,k的重心坐标；在二维空间中，点 i 的的重心坐标可以使用三角形的有向面积计算；具体来说，对于图 1 中的例子，点 i 的重心坐标$\{a_{il},a_{ij},a_{ik}\}$计算如下；

$$a_{ij}=\frac{S_{ikl}}{S_{jkl}},a_{ik}=\frac{S_{jil}}{S_{jkl}},a_{il}=\frac{S_{jki}}{S_{jkl}}$$

其中$S_{ikl},S_{jil},S_{jki},S_{jkl}$分别是三角形 ikl, jil, jki, jkl 的有向面积；如果点j,k,l满足右手法则，那么有向面积S_{jkl}的符号是正的，否则有向面积S_{jkl}的符号是负的；

假设我们已知二维空间中 4 个点i,j,k,l的坐标，$p_i,p_j,p_k,p_l\in R^2$；那么三角形$S_{ikl},S_{jil},S_{jki},S_{jkl}$的面积可以使用行列式计算；

$$S_{ikl}=\frac{1}{2}\begin{vmatrix}1&1&1\\p_i&p_k&p_l\end{vmatrix}$$

$$S_{jil}=\frac{1}{2}\begin{vmatrix}1&1&1\\p_j&p_i&p_l\end{vmatrix}$$

$$S_{jki}=\frac{1}{2}\begin{vmatrix}1&1&1\\p_j&p_k&p_i\end{vmatrix}$$

$$S_{jkl}=\frac{1}{2}\begin{vmatrix}1&1&1\\p_j&p_k&p_l\end{vmatrix}$$

此处要求三角形 jkl 的面积不能为 0，否则计算重心坐标a_{ij},a_{ik},a_{il}时会除零；

3. 使用重心坐标表示位姿变换

3.1 使用节点 i,ix,iy 表示节点 j,jx,jy

SLAM 算法中可以使用多种方式获得两个关键帧之间的位姿变换（二维情况下，旋转平移矩阵$T\in R^{3\times3}$）；下面表格列出使用相机传感器得到两个关键帧之间的不同类型特征点，可以使用对应的算法计算位姿变换；

表 1，相机的位姿求解算法

特征点类型	位姿变换算法
2d-2d	对极几何
2d-2d（平面）	
2d-3d	DLT算法、EPnP算法
3d-3d	icp算法

同样的，如果使用激光雷达传感器获得两帧点云，那么可以使用多种点云配准算法得到位姿变换；此外，我们也可以通过 IMU 传感器、GPS 传感器、轮速计和回环检测等方法得到两帧之间的位姿变换；

不管使用什么类型传感器，我们得到的初始信息都是两个关键帧（节点）之间的位姿变换；后端优化的目的是使用这些位姿变换信息估计出精确的全局地图；下面我们使用节点$i\in V$来指代关键帧，每个节点 i 都有自己的局部坐标系Σ_i；节点 i 的局部坐标系Σ_i与全局坐标系Σ_g之间存在未知旋转矩阵R_i；设节点 i 在二维空间中的坐标为$p_i=[x_i,y_i]^T$；设所有位姿变换T_{ij}的集合为$E=\{T_{ij},i,j\in V\}$；我们可以定义有向图 G=[V,E]，其中 V 表示所有节点（关键帧）的集合，E 表示所有位姿变换的集合；

下面我们将任意两个节点（节点 i、节点 j）之间的位姿变换$T_{ij} \in R^{3\times3}$转换为线性方程约束；已知节点 i 的坐标是$p_i \in R^2$；我们设虚拟节点i_x位于局部坐标系Σ_i的 x 轴，坐标为p_{ix}，并且满足$||p_{ix} - p_i|| = 1$；设虚拟节点i_y位于局部坐标系Σ_i的 y 轴，坐标为p_{iy}，并且满足$||p_{iy} - p_i|| = 1$；那么在局部坐标系Σ_i中，节点 i 是局部坐标系的原点，i_x, i_y是局部坐标系上两个轴的单位向量的节点；

同样节点 j 的坐标是$p_j \in R^2$；我们设虚拟节点p_{jx}位于局部坐标系Σ_j的 x 轴，并且满足$||p_{jx} - p_j|| = 1$；设虚拟节点p_{jy}位于局部坐标系Σ_j的 y 轴，并且满足$||p_{jy} - p_j|| = 1$；那么局部坐标系Σ_j中，节点 j 是局部坐标系的原点，p_{jx}, p_{jy}是局部坐标系上两个轴的单位坐标向量；

设节点 i 到节点 j 的位姿变换为$T_{ij} \in R^{3\times3}$；

$$T_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{bmatrix} \in R^{3\times3}, R_{ij} \in R^{2\times2}$$

$$t_{ij} = \left[x_{ij}, y_{ij}\right]^T \in R^2$$

那么节点 j 和虚拟节点j_x, j_y在局部坐标系Σ_i中的位置如下；

$$p_j^i = t_{ij}$$

$$p_{jx}^i = R_{ij}[1,0]^T + t_{ij}$$

$$p_{jy}^i = R_{ij}[0,1]^T + t_{ij}$$

节点 i 和虚拟节点i_x, i_y在局部坐标系Σ_i中的坐标如下；

$$p_i^i = [0,0]^T$$

$$p_{ix}^i = [1,0]^T$$

$$p_{iy}^i = [0,1]^T$$

如果已知局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}），我们需要使用重心坐标表示局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}）；通过这种方式，我们可以从第一个节点开始，递推所有节点的局部坐标系；并且因为节点 i 和虚拟节点i_x, i_y是坐标系的不同轴上的点，三角形 i-ix-iy 的面积不等于 0，满足重心坐标计算的条件；

因为重心坐标在节点集合$V_{ij} = \{i, i_x, i_y, j, j_x, j_y\}$经过任意的旋转平移后仍然保持不变，那么我们可以使用节点在局部坐标系Σ_i中的相对位置（$p_i^i, p_{ix}^i, p_{iy}^i, p_j^i, p_{jx}^i, p_{jy}^i$）计算重心坐标；这样计算得到的重心坐标对于节点在全局坐标系中的位置（$p_i, p_i x, p_{iy}, p_j, p_{jx}, p_{jy}$）仍然成立；

下面我们首先使用局部坐标系Σ_i中四个节点i, i_x, i_y，构建节点 j 的重心坐标；

$$p_j = a_{j,i}p_i + a_{j,ix}p_{ix} + a_{j,iy}p_{iy} \tag{1}$$

$$a_{j,i} = \frac{S_{j,ix,iy}}{S_{i,ix,iy}} \tag{1.1}$$

$$a_{j,ix} = \frac{V_{i,j,iy}}{V_{i,ix,iy}} \tag{1.2}$$

$$a_{j,iy} = \frac{V_{i,ix,j}}{V_{i,ix,iy}} \tag{1.3}$$

$$S_{i,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{1.4}$$

$$S_{j,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{1.5}$$

$$S_{i,j,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_j^i & p_{iy}^i \end{vmatrix} \tag{1.6}$$

$$S_{i,ix,j} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_j^i \end{vmatrix} \tag{1.7}$$

然后我们使用局部坐标系Σ_i中四个节点i, i_x, i_y，构建节点j_x的重心坐标；

$$p_{jx} = a_{jx,i}p_i + a_{jx,ix}p_{ix} + a_{jx,iy}p_{iy} \tag{2}$$

$$a_{jx,i} = \frac{S_{jx,ix,iy}}{S_{i,ix,iy}} \tag{2.1}$$

$$a_{jx,ix} = \frac{S_{i,jx,iy}}{S_{i,ix,iy}} \tag{2.2}$$

$$a_{jx,iy} = \frac{S_{i,ix,jx}}{S_{i,ix,iy}} \tag{2.3}$$

$$S_{i,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{2.4}$$

$$S_{jx,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_{jx}^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{2.5}$$

$$S_{i,jx,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{jx}^i & p_{iy}^i \end{vmatrix} \tag{2.6}$$

$$S_{i,ix,jx} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_{jx}^i \end{vmatrix} \tag{2.7}$$

然后我们使用局部坐标系Σ_i中三个节点i, i_x, i_y，构建节点j_y的重心坐标；

$$p_{jy} = a_{jy,i}p_i + a_{jy,ix}p_{ix} + a_{jy,iy}p_{iy} \tag{3}$$

$$a_{jy,i} = \frac{S_{jy,ix,iy}}{S_{i,ix,iy}} \tag{3.1}$$

$$a_{jy,ix} = \frac{S_{i,jy,iy}}{S_{i,ix,iy}} \tag{3.2}$$

$$a_{jy,iy} = \frac{S_{i,ix,jy}}{S_{i,ix,iy}} \tag{3.3}$$

$$S_{i,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{3.4}$$

$$S_{jy,ix,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_{jy}^i & p_{ix}^i & p_{iy}^i \end{vmatrix} \tag{3.5}$$

$$S_{i,jy,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{jy}^i & p_{iy}^i \end{vmatrix} \tag{3.6}$$

$$S_{i,ix,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^i & p_{ix}^i & p_{jy}^i \end{vmatrix} \tag{3.7}$$

如果已知局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}），我们可以使用方程(1),(2),(3)计算局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}）；但是问题反过来，如果已知局部坐标系Σ_j中所有点的全局坐标

（p_j, p_{jx}, p_{jy}），使用上述方程不一定可以计算局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}）；

因此在下一节中，我们可以构建对称的线性方程；我们可以使用局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}），构建重心坐标表示局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}）；

2.2 使用节点 j,jx,jy 表示节点 i,ix,iy

在本节中，我们构建对称的线性方程；我们可以使用局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}），构建重心坐标表示局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}）；

已知节点 i 到节点 j 的位姿变换为$T_{ij} \in R^{3\times3}$，那么节点 j 到节点 i 的位姿变换为$T_{ji} = \left(T_{ij}\right)^{-1}$；

$$T_{ji} = \begin{bmatrix} R_{ji} & t_{ji} \\ 0 & 1 \end{bmatrix} \in R^{3\times3}, R_{ji} \in R^{2\times2}$$

$$t_{ji} = \left[x_{ji}, y_{ji}\right]^T \in R^2$$

那么节点 i 和虚拟节点i_x, i_y在局部坐标系Σ_j中的位置如下；

$$p_i^j = t_{ji}$$

$$p_{ix}^j = R_{ji}[1,0]^T + t_{ji}$$

$$p_{iy}^j = R_{ji}[0,1]^T + t_{ji}$$

节点 j 和虚拟节点j_x, j_y在局部坐标系Σ_j中的坐标如下；

$$p_j^j = [0,0]^T$$

$$p_{jx}^j = [1,0]^T$$

$$p_{jy}^j = [0,1]^T$$

已知局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}），我们需要使用重心坐标表示局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}）；并且因为节点 j 和虚拟节点j_x, j_y是坐标系的不同轴上的点，三角形 j-jx-jy 的面积不等于 0，满足重心坐标计算的条件；

因为重心坐标在节点集合$V_{ij} = \{i, i_x, i_y, j, j_x, j_y\}$经过任意的旋转平移后仍然保持不变，那么我们可以使用节点在局部坐标系Σ_j中的相对位置（$p_i^j, p_{ix}^j, p_{iy}^j, p_j^j, p_{jx}^j, p_{jy}^j$）计算重心坐标；这样计算得到的重心坐标对于节点在全局坐标系中的位置（$p_i, p_{ix}, p_{iy}, p_j, p_{jx}, p_{jy}$）仍然成立；

下面我们首先使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点 i 的重心坐标；

$$p_i = a_{i,j}p_j + a_{i,jx}p_{jx} + a_{i,jy}p_{jy} \tag{4}$$

$$a_{i,j} = \frac{S_{i,jx,jy}}{S_{j,jx,jy}} \tag{4.1}$$

$$a_{i,jx} = \frac{S_{j,i,jy}}{S_{j,jx,jy}} \tag{4.2}$$

$$a_{i,jy} = \frac{S_{j,jx,i}}{S_{j,jx,jy}} \tag{4.3}$$

$$S_{j,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{4.4}$$

$$S_{i,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_i^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{4.5}$$

$$S_{j,i,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_i^j & p_{jy}^j \end{vmatrix} \tag{4.6}$$

$$S_{j,jx,i} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_i^j \end{vmatrix} \tag{4.7}$$

然后我们使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点i_x的重心坐标；

$$p_{ix} = a_{ix,j}p_j + a_{ix,jx}p_{jx} + a_{ix,jy}p_{jy} \tag{5}$$

$$a_{ix,j} = \frac{S_{ix,jx,jy}}{S_{j,jx,jy}} \tag{5.1}$$

$$a_{ix,jx} = \frac{S_{j,ix,jy}}{S_{j,jx,jy}} \tag{5.2}$$

$$a_{ix,jy} = \frac{S_{j,jx,ix}}{S_{j,jx,jy}} \tag{5.3}$$

$$S_{j,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{5.4}$$

$$S_{ix,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_{ix}^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{5.5}$$

$$S_{j,ix,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{ix}^j & p_{jy}^j \end{vmatrix} \tag{5.6}$$

$$S_{j,jx,ix} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_{ix}^j \end{vmatrix} \tag{5.7}$$

然后我们使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点i_y的重心坐标；

$$p_{iy} = a_{iy,j}p_j + a_{iy,jx}p_{jx} + a_{iy,jy}p_{jy} \tag{6}$$

$$a_{iy,j} = \frac{S_{iy,jx,jy}}{S_{j,jx,jy}} \tag{6.1}$$

$$a_{iy,jx} = \frac{S_{j,iy,jy}}{S_{j,jx,jy}} \tag{6.2}$$

$$a_{iy,jy} = \frac{S_{j,jx,iy}}{S_{j,jx,jy}} \tag{6.3}$$

$$S_{j,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{6.4}$$

$$S_{iy,jx,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_{iy}^j & p_{jx}^j & p_{jy}^j \end{vmatrix} \tag{6.5}$$

$$S_{j,iy,jy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{iy}^j & p_{jy}^j \end{vmatrix} \tag{6.6}$$

$$S_{j,jx,iy} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ p_j^j & p_{jx}^j & p_{iy}^j \end{vmatrix} \tag{6.7}$$

如果已知局部坐标系Σ_j中所有点的全局坐标（p_j, p_{jx}, p_{jy}），我们可以使用方程(4),(5),(6)计算局部坐标系Σ_i中所有点的全局坐标（p_i, p_{ix}, p_{iy}）；

同时使用方程(1)-(6)，我们既可以使用节点i, i_x, i_y的坐标计算节点j, j_x, j_y的坐标，也可以使用节点j, j_x, j_y的坐标计算节点i, i_x, i_y的坐标；因此对于任意位姿变换$T_{ij} \in E$，可以使用上述 6 个线性方程进行表示；然后这 6 个线性方程在节点集合$V_{ij} = \{i, i_x, i_y, j, j_x, j_y\}$的所有节点构成的图形进行缩放时仍然成立，因此这些线性方程大致可以看作构建了一种相似变换的约束；

设虚拟节点集合为$V_v = \{1_x, 1_y, \dots, n_x, n_y\}$；设所有节点$i \in V$的坐标向量为$p = [p_1^T, p_2^T, \dots, p_n^T]^T \in R^{2n}$；设所有节点$i \in V$的虚拟节点$i_x, i_y$的坐标向量为$p_v = \left[p_{1x}^T, p_{1y}^T, \dots, p_{nx}^T, p_{ny}^T\right]^T \in R^{4n}$；将坐标向量 p 和$p_v$合并得到坐标向量$p_{acc} = [p^T, p_v^T]^T \in R^{6n}$；每个位姿变换$T_{ij} \in V$可以转换为 6 个线性方程；将这些线性方程组合可以得到如下线性方程组；

$$Ap_{acc} = 0$$

该算法的伪代码如下；

过程 1，线性方程表示位姿变换

输入，图G=[V,E]，V表示n个节点集合，E表示所有位姿变换集合

输出，矩阵A

设所有节点向量$p = [p_1^T, p_2^T, \dots, p_n^T]^T \in R^{2n}$；

设所有节点的虚拟节点向量$p_v = \left[p_{1x}^T, p_{1y}^T, \dots, p_{nx}^T, p_{ny}^T\right]^T \in R^{4n}$；

将坐标向量p和p_v合并得到$p_{acc} = [p^T, p_v^T]^T \in R^{6n}$；

For $T_{ij} \in V$:

//计算节点i, i_x, i_y, j, j_x, j_y在局部坐标系Σ_i中的坐标

已知节点i到节点j的位姿变换为$T_{ij} \in R^{3\times3}$，$T_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{bmatrix} \in R^{3\times3}, R_{ij} \in R^{2\times2}, t_{ij} = \left[x_{ij}, y_{ij}\right]^T \in R^2$；

节点j和虚拟节点j_x, j_y在局部坐标系Σ_i中的位置如下；

$p_j^i = t_{ij}$

$p_{jx}^i = R_{ij}[1,0]^T + t_{ij}$

$p_{jy}^i = R_{ij}[0,1]^T + t_{ij}$

节点i和虚拟节点i_x, i_y在局部坐标系Σ_i中的坐标，$p_i^i = [0,0]^T, p_{ix}^i = [1,0]^T, p_{iy}^i = [0,1]^T$；

//使用节点i, i_x, i_y表示节点j, j_x, j_y

使用局部坐标系Σ_i中三个节点i, i_x, i_y，构建节点j的重心坐标，$p_j = a_{j,i}p_i + a_{j,ix}p_{ix} + a_{j,iy}p_{iy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{j,i}, a_{j,ix}, a_{j,iy}$使用相对坐标$p_i^i, p_{ix}^i, p_{iy}^i, p_j^i, p_{jx}^i, p_{jy}^i$计算，见公式(1.1)-(1.7)；

使用局部坐标系Σ_i中三个节点i, i_x, i_y，构建节点j_x的重心坐标，$p_{jx} = a_{jx,i}p_i + a_{jx,ix}p_{ix} + a_{jx,iy}p_{iy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{jx,i}, a_{jx,ix}, a_{jx,iy}$使用相对坐标$p_i^i, p_{ix}^i, p_{iy}^i, p_j^i, p_{jx}^i, p_{jy}^i$计算，见公式(2.1)-(2.7)；

使用局部坐标系Σ_i中三个节点i, i_x, i_y，构建节点j_y的重心坐标，$p_{jy} = a_{jy,i}p_i + a_{jy,ix}p_{ix} + a_{jy,iy}p_{iy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{jy,i}, a_{jy,ix}, a_{jy,iy}$使用相对坐标$p_i^i, p_{ix}^i, p_{iy}^i, p_j^i, p_{jx}^i, p_{jy}^i$计算，见公式(3.1)-(3.7)；

//计算节点i, i_x, i_y, j, j_x, j_y在局部坐标系Σ_j中的坐标

已知节点j到节点i的位姿变换为$T_{ji} \in R^{3\times3}$，$T_{ji} = \begin{bmatrix} R_{ji} & t_{ji} \\ 0 & 1 \end{bmatrix} \in R^{3\times3}, R_{ji} \in R^{2\times2}, t_{ji} = \left[x_{ji}, y_{ji}\right]^T \in R^2$；

节点i和虚拟节点i_x, i_y在局部坐标系Σ_j中的位置如下；

$p_i^j = t_{ji}$

$p_{ix}^j = R_{ji}[1,0]^T + t_{ji}$

$p_{iy}^j = R_{ji}[0,1]^T + t_{ji}$

节点j和虚拟节点j_x, j_y在局部坐标系Σ_j中的坐标，$p_j^j = [0,0]^T, p_{jx}^j = [1,0]^T, p_{jy}^j = [0,1]^T$；

//使用节点j, j_x, j_y表示节点i, i_x, i_y

使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点i的重心坐标，$p_i = a_{i,j}p_j + a_{i,jx}p_{jx} + a_{i,jy}p_{jy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{i,j}, a_{i,jx}, a_{i,jy}$使用相对坐标$p_i^j, p_{ix}^j, p_{iy}^j, p_j^j, p_{jx}^j, p_{jy}^j$计算，见公式(4.1)-(4.7)；

使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点i_x的重心坐标，$p_{ix} = a_{ix,j}p_j + a_{ix,jx}p_{jx} + a_{ix,jy}p_{jy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{ix,j}, a_{ix,jx}, a_{ix,jy}$使用相对坐标$p_i^j, p_{ix}^j, p_{iy}^j, p_j^j, p_{jx}^j, p_{jy}^j$计算，见公式(5.1)-(5.7)；

使用局部坐标系Σ_j中三个节点j, j_x, j_y，构建节点i_y的重心坐标，$p_{iy} = a_{iy,j}p_j + a_{iy,jx}p_{jx} + a_{iy,jy}p_{jy}$，加入线性方程组$Ap_{acc} = 0$；参数$a_{iy,j}, a_{iy,jx}, a_{iy,jy}$使用相对坐标$p_i^j, p_{ix}^j, p_{iy}^j, p_j^j, p_{jx}^j, p_{jy}^j$计算，见公式(6.1)-(6.7)；

4. 不同传感器的位姿变换

本节中我们讨论不同类型的传感器获得的数据，怎样转化为不同节点之间的位姿变换；我们可以通过相机、激光雷达、IMU 传感器、GPS 传感器、轮速计和回环检测得到两帧之间的位姿变换；下面分别讨论各种传感器获得的位姿变换；

4.1 非 GPS 传感器

我们首先讨论相机和激光雷达传感器的情况；这两种传感器可以获得确定的两个节点（如节点 i 和节点 j）之间

的位姿变换T_{ij}；通常节点 i 与节点 j 是相邻的两个关键帧，或者距离非常接近的两个关键帧；

然后对于 IMU 传感器和轮速计，这两种传感器通常获得的是相邻的两个关键帧之间位姿变换$T_{i-1,i}$；

回环检测部分同样可以获得两个节点（节点 i 和节点 j）之间的位姿变换T_{ij}；并且通常节点 i 与节点 j 是机器人绕了一圈后回到原点找到的两个相邻节点；节点 i 与节点 j 在机器人运动时间上相隔较远，但是在距离上较近；

通过非 GPS 传感器获得两个节点（节点 i 和节点 j）之间的位姿变换T_{ij}，可以转换为两组节点（节点i, i_x, i_y和节点j, j_x, j_y）之间的线性方程；

3.2 GPS 传感器

GPS 传感器的情况较为特殊，GPS 传感器获得的数据是经度和纬度；我们可以使用公式将经纬度转换为具体的距离值；那么 GPS 传感器每次检测的位置都是在固定的经纬度坐标系的，或者说 GPS 传感器具有一个固定的全局坐标系；其他类型的传感器（如相机和激光雷达）只能获得具体的两个节点（节点 i、节点 j）之间的位姿变换T_{ij}；如果需要获得距离很远的两个节点（节点 i、节点$i+n$）之间的位姿变换，只能通过不断累加得到，$T_{i,i+n} = T_{i,i+1}T_{i+1,i+2}\dots T_{i+n-1,i+n}$；这样获得累加位姿变换$T_{i,i+n}$的误差会逐渐累加；对于 GPS 传感器，因为具有固定的经纬度坐标系，对于距离很远的两个节点（节点 i、节点$i+n$）之间的平移量$t_{i,i+n}$的仍然具有非常高的精度；因为 GPS 传感器的数据对于大型地图在保持整体地图形状的准确性上具有很大的作用；

GPS 传感器的另一个特殊情况是 GPS 数据只有位移，没有旋转信息；假设我们获得了节点 i 的 GPS 数据（经纬度坐标系中的位置），然后获得了节点 j 的 GPS 数据；那么我们只能得到在经纬度坐标系中，节点 i 到节点 j 的平移量$t_{ij} \in R^2$，没有旋转信息；因此 GPS 传感器获得的平移向量$t_{ij} \in R^2$不能和其他传感器获得位姿变换$T_{ij} \in R^{3\times3}$组合，构建线性方程约束；虽然我们可以构建非线性的位姿图约束，但是这个和我们希望构建线性方程的优化算法不同；

GPS 传感器获得的数据通常只有经纬度，没有高度数据；并且即使 GPS 传感器获得了高度数据，高度数据的精度远低于经纬度数据；如果 GPS 传感器获得的数据只有经纬度，那么可以得到两个节点（节点 i、节点 j）之间的平面平移$t_{ij} \in R^2$；我们可以选择三个节点（节点 i,j,k）构建线性方程（使用平面三角形）；

对于 GPS 传感器的处理我们采用论文[1]中的方法，使用相似三角形约束构建线性方程；为了避免论文的描述过于冗长，我们直接给出使用 GPS 数据构建的线性方程组如下；具体的线性方程构建过程在论文[1]描述；

$$Cp_{acc} = 0$$

其中包括使用相邻 GPS 数据构建线性方程和使用跨节点 GPS 数据构建线性方程；

3.3 位姿变换的方差

我们可以通过相机、激光雷达、IMU 传感器、GPS 传感器、轮速计和回环检测得到两帧之间的位姿变换；通常不同传感器获得位姿变换具有不同的方差，我们可以根据传感器自身的特点预设设定每种传感器的方差；或者根据 SLAM 算法的前端模块中，计算位姿变换的过程中得到位姿T_{ij}的方差；位姿变换矩阵T_{ij}是3×3矩阵，为了方便起见，我们使用单个方差σ_{ij}^2当作整个位姿变换矩阵的方差；

已知各节点$i \in V$和虚拟节点$i_x, i_y \in V_v$之间的线性方程如下；

$$Ap_{acc} = 0 + v_1$$

$$Cp_{acc} = 0 + v_2$$

其中误差向量$v_1 \sim N(0, \Sigma_1), v_2 \sim N(0, \Sigma_2)$；我们忽略各个线性方程之间的联系，那么协方差矩阵$\Sigma_1, \Sigma_2$是对角矩阵；

对于误差向量v_1的方差Σ_1，我们使用线性方程对应的位姿变换T_{ij}的方差σ_{ij}^2作为该线性方程的方差；对于误差向量v_2，每个线性方程使用两个位姿变换（如T_{ij}, T_{ik}）构建，为了方便起见，我们设对应线性方程的方差为$\sigma^2 = \sigma_{ij}^2 + \sigma_{ik}^2$；

5. 地图尺度计算

5.1 不使用 GPS 传感器情况

在图 G=[V,E]中，V 是所有节点（关键帧）的集合；设集合 V 中存在 n 个节点；所有节点的坐标向量是$p = [p_1^T, \dots, p_n^T]^T \in R^{2n}$；每个节点$i \in V$的局部坐标系$\Sigma_i$中定义了 2 个虚拟节点$i_x, i_y$；虚拟节点集合是$V_v = \{i_x, i_y, i \in V\}$；虚拟节点坐标向量为$p_v = \left[p_{1x}^T, p_{1y}^T, \dots, p_{nx}^T, p_{ny}^T\right] \in R^{4n}$；坐标向量 p 和$p_v$合并得到$p_{acc} = [p^T, p_v^T]^T \in R^{6n}$；

已知所有位姿变换T_{ij}的集合为$E = \{T_{ij}, i, j \in V\}$；任意两个节点（节点 i、节点 j）之间的位姿变换T_{ij}都可以转换为 6 个线性方程；我们设所有位姿变换$T_{ij} \in E$对应的线性方程组如下；

$$Ap_{acc} = 0$$

我们设第一个节点的局部坐标系Σ_1为全局坐标系Σ_g，计算其他节点的坐标；如果我们使用其他坐标系Σ_i作为全局坐标系Σ_g，那么使用对应的位姿变换T_{1i}就可以将所有坐标变换到坐标系Σ_i上；设局部坐标系Σ_1为全局坐标系Σ_g，那么节点 1 的坐标为$p_1 = [0,0]^T$，虚拟节点1_x的坐标为$p_{1x} = [1,0]^T$，虚拟节点1_y的坐标为$p_{1y} = [0,1]^T$；根据线性方程组$Ap_{acc} = 0$的构建方法，只要确定任意一个节点的局部坐标系的坐标（如p_1, p_{1x}, p_{1y}），就能计算所有节点的坐标p_{acc}；线性方程组$Ap_{acc} = 0$约束了不同节点的局部坐标系之间的关系，但是尺度是可以缩放的；

如果使用p_1, p_{1x}, p_{1y}坐标计算p_{acc}中其他节点的坐标，那么相当于使用p_1, p_{1x}, p_{1y}坐标确定了图形的尺度；那么这个尺度信息会从节点 1 向外传播到其他节点时，误差会

逐渐的放大；比如我们设定局部坐标系Σ_1中三个轴的单位向量模长是 1（$\rho_{i,ix}=1,\rho_{i,iy}=1$），但是计算得到节点 n 的坐标后，局部坐标系$\Sigma_n$中三个轴的向量（节点 n、节点$n_x$、节点$n_y$组成的向量）的模长的模长不一定保持为 1（$\rho_{n,nx},\rho_{n,ny}$）；对此我们可以采用如下方法确定更为精确的地图尺度；

设节点 1 的坐标为$p_1=[0,0]^T$，虚拟节点1_x的坐标是$p_{1x}=[\rho,0]^T$，虚拟节点1_y的坐标是$p_{1y}=[0,\rho]^T$；设坐标向量p_{acc}中去除p_1,p_{1x},p_{1y}后的坐标向量为$p_{sub}\in R^{6n-6}$；将坐标p_1,p_{1x},p_{1y}的坐标代入方程，线性方程组$Ap_{acc}=0+v_1$变成如下；

$$A_{sub}p_{sub}=b_{sub}+v_{sub1}$$

其中$v_{sub1}\sim N(0,\Sigma_{sub1})$表示各个线性方程对应的噪声；每个线性方程的方差使用对应的位姿变换T_{ij}的方差σ_{ij}^2表示；线性方程组中线性方程的个数没有变化，并且每个线性方程对应的方差没有变换；那么误差向量$v_{sub1}=v_1$，方差矩阵$\Sigma_{sub1}=\Sigma_1$；该线性方程组的最小二乘解如下；

$$p_{sub}=\left(A_{sub}^T\Sigma_1^{-1}A_{sub}\right)^{-1}A_{sub}^T\Sigma_1^{-1}b_{sub}$$

坐标向量p_{sub}表示从节点 2 到节点 n 坐标，并且每个坐标（如p_i,p_{ix},p_{iy}）中的每个分量都是未知量ρ的一次函数；

$$p_i=[a_i\rho+b_i,c_i\rho+d_i]^T$$

$$p_{ix}=[a_{ix}\rho+b_{ix},c_{ix}\rho+d_{ix}]^T$$

$$p_{iy}=\left[a_{iy}\rho+b_{iy},c_{iy}\rho+d_{iy}\right]^T$$

$$p_{iz}=[a_{iz}\rho+b_{iz},c_{iz}\rho+d_{iz}]^T$$

同时节点 1 和虚拟节点$1_x,1_y$的坐标中，因此坐标向量p_{acc}中每个分量都是未知量ρ的一次函数；然后我们定义如下误差函数J_1；该误差函数表示每个节点 i 的坐标系Σ_i中，节点 i 到虚拟节点i_x的模长应该为 1，节点 i 到虚拟节点i_y的模长应该为 1；

$$J_1=\sum\nolimits_{i\in V}\left(\left(\left||p_i-p_{ix}|\right|^2-1\right)^2+\left(\left||p_i-p_{iy}|\right|^2-1\right)^2\right)$$

$$=\sum\nolimits_{i\in V}+\left(\left((a_i-a_{iz})\rho+(b_i-b_{iz})\right)^2\right.$$

$$+\left((c_i-c_{iz})\rho+(d_i-d_{iz})\right)^2+\left((e_i-e_{iz})\rho+(f_i-f_{iz})\right)^2$$

$$\left.-1\right)^2+$$

$$\left(\left((a_i-a_{ix})\rho+(b_i-b_{ix})\right)^2+\left((c_i-c_{ix})\rho+(d_i-d_{ix})\right)^2\right.$$

$$\left.+\left((e_i-e_{ix})\rho+(f_i-f_{ix})\right)^2-1\right)^2$$

$$=A_1\rho^4+B_1\rho^3+C_1\rho^2+D_1\rho+E_1$$

定义误差函数J_2如下；如果节点 i 和节点 j 之间存在位姿变换$T_{ij}\in R^{3\times3}$，其中平移变换$t_{ij}\in R^2$，那么节点 i 和节点 j 之间的长度是$\rho_{ij}=\left||t_{ij}|\right|$；同时我们可以使用求解得到的节点 i 的坐标$p_i$与节点 j 的坐标$p_j$计算模长（未知量$\rho$的一次函数）；我们使用两个模长的差异定义误差函数；

$$J_2=\sum\nolimits_{T_{ij}\in E}\left(\left||p_i-p_j|\right|^2-\rho_{ij}^2\right)^2$$

$$=\sum\nolimits_{T_{ij}\in E}\left(\left(\left(a_i-a_j\right)\rho+\left(b_i-b_j\right)\right)^2\right.$$

$$\left.+\left(\left(c_i-c_j\right)\rho+\left(d_i-d_j\right)\right)^2-\rho_{ij}^2\right)^2$$

$$=A_2\rho^4+B_2\rho^3+C_2\rho^2+D_2\rho+E_2$$

将误差函数J_1和误差函数J_2相加，得到误差函数 J；

$$J=J_1+J_2$$

$$=A\rho^4+B\rho^3+C\rho^2+D\rho+E$$

为了使该误差函数最小，我们对 J 计算导数；

$$\frac{dJ}{d\rho}=4A\rho^3+3B\rho^2+2C\rho+D=0$$

然后使用三次方程的公式计算三个根ρ_1,ρ_2,ρ_3；舍弃三个根中的复数根之后，对于剩余的根使用 J 的二次导数判断；

$$\frac{dJ^2}{d\rho^2}=12A\rho^2+6B\rho+2C$$

如果根ρ_i的二次导数为正，那么根ρ_i是函数 J 的最小值；如果根ρ_i的二次导数为负，那么根ρ_i是函数 J 的局部最大值；得到优化的地图尺度ρ之后，我们可以代入坐标$p_{acc}\in R^{6n}$；

过程 2，计算地图尺度和节点坐标（不使用 GPS 数据）

输入，线性方程组$Ap_{acc}=0$，方差矩阵Σ_1

输出，坐标向量$p_{acc}\in R^{6n}$，地图尺度ρ

设节点$i\in V$的坐标向量为$p=[p_1^T,p_2^T,\ldots,p_n^T]^T\in R^{2n}$；

设虚拟节点坐标向量为$p_v=\left[p_{1x}^T,p_{1y}^T,\ldots,p_{nx}^T,p_{ny}^T\right]\in R^{4n}$；

设坐标向量p和p_v合并得到$p_{acc}=[p^T,p_v^T]^T\in R^{6n}$；

设节点1的坐标为$p_1=[0,0]^T$，虚拟节点1_x的坐标是$p_{1x}=[\rho,0]^T$，虚拟节点1_y的坐标是$p_{1y}=[0,\rho]^T$；

设坐标向量p_{acc}去除p_1,p_{1x},p_{1y}的坐标向量是$p_{sub}\in R^{6n-6}$；

将坐标p_1,p_{1x},p_{1y}的坐标代入方程，线性方程组$Ap_{acc}=0$变为$A_{sub}p_{sub}=b_{sub}$；

使用线性方程组$A_{sub}p_{sub}=b_{sub}+v_{sub1}$计算虚拟节点的平面分量$p_{sub}$；该线性方程组的最小二乘解，$p_{sub}=\left(A_{sub}^T\Sigma_1^{-1}A_{sub}\right)^{-1}A_{sub}^T\Sigma_1^{-1}b_{sub}$（$\rho$的一次函数）；

定义误差函数J_1，$J1=\sum_{i\in V}\left(\left(\left||p_1-p_{1x}|\right|^2-1\right)^2+\left(\left||p_1-p_{1y}|\right|^2-1\right)^2\right)=A_1\rho^4+B_1\rho^3+C_1\rho^2+D_1\rho+E_1$；

定义误差函数J_2，$J2=\sum_{T_{ij}\in E}\left(\left\|p_i-p_j\right\|^2-\rho_{ij}\right)^2=A_2\rho^4+B_2\rho^3+C_2\rho^2+D_2\rho+E_2$；

相加得到误差函数J，$J=J_1+J_2=A\rho^4+B\rho^3+C\rho^2+D\rho+E$；

计算使得该误差函数最小的参数ρ，对J计算导数，$\frac{dJ}{d\rho}=4A\rho^3+3B\rho^2+2C\rho+D=0$；

使用三次方程的公式计算三个根ρ_1,ρ_2,ρ_3；舍弃三个根中的复数根，并使用函数J的二次导数选择最小值对应的根ρ_i；

参数ρ代入坐标向量p_{acc}；

4.2 使用 GPS 传感器情况

不使用 GPS 传感器时，各节点$i\in V$和虚拟节点$i_x,i_y\in V_v$之间的线性方程如下；

$$Ap_{acc}=0+v_1 \tag{7}$$

使用 GPS 传感器时，各节点$i\in V$和虚拟节点$i_x,i_y\in V_v$之间的线性方程还需要增加如下；

$$Cp_{acc}=0+v_2 \tag{8}$$

然后将方程(7),(8)合并为如下；

$$Ep_{acc}=0+v_3$$

其中误差向量$v_3=[v_1^T,v_2^T]^T,v_3\sim N(0,\Sigma_3)$；我们忽略各个线性方程之间的联系，那么协方差矩阵$\Sigma_3$是对角矩阵；并且满足如下关系，$\Sigma_3=diag\{\Sigma_1,\Sigma_2\}$；

我们设定如下坐标$p_1=[0,0]^T,p_{1x}=[\rho,0]^T,p_{1y}=[0,\rho]^T$,；将这些坐标代入方程$Ep_{acc}=0+v_3$，该方程变成如下形式；

$$E_{sub}p_{sub}=F_{sub}+v_{sub3}$$

其中坐标向量$p_{sub}\in R^{6n-6}$是坐标向量p_{acc}删除p_1,p_{1x},p_{1y}后的向量；误差向量$v_{sub3}\sim N(0,\Sigma_{sub3})$表示各个线性方程对应的噪声；线性方程组中线性方程的个数没有变化，并且每个线性方程对应的方差没有变化；那么误差向量$v_{sub3}=v_3$，方差矩阵$\Sigma_{sub3}=\Sigma_3$；该线性方程组的最小二乘解如下；

$$p_{sub}=\left(E_{sub}^T\Sigma_3^{-1}E_{sub}\right)^{-1}E_{sub}^T\Sigma_3^{-1}F_{sub}$$

计算地图尺度和节点坐标的伪代码如下；

过程 3，计算地图尺度和节点坐标（使用 GPS 数据）

输入，线性方程组$Ap_{acc}=0,Cp_{acc}=0$，方差矩阵Σ_1,Σ_2

输出，坐标向量$p_{acc}\in R^{6n}$，地图尺度ρ

设节点$i\in V$的坐标向量为$p=[p_1^T,p_2^T,\ldots,p_n^T]^T\in R^{2n}$；

设虚拟节点坐标向量为$p_v=\left[p_{1x}^T,p_{1y}^T,\ldots,p_{nx}^T,p_{ny}^T\right]\in R^{4n}$；

设坐标向量p和p_v合并得到$p_{acc}=[p^T,p_v^T]^T\in R^{6n}$；

设节点1的坐标为$p_1=[0,0]^T$，虚拟节点1_x的坐标是$p_{1x}=[\rho,0]^T$，虚拟节点1_y的坐标是$p_{1y}=[0,\rho]^T$；

设坐标向量p_{acc}去除p_1,p_{1x},p_{1y}的坐标向量是$p_{sub}\in R^{6n-6}$；

线性方程$Ap_{acc}=0$和$Cp_{acc}=0$合并得到线性方程$Ep_{acc}=0$；

将坐标p_1,p_{1x},p_{1y}的坐标代入方程，线性方程组$Ep_{acc}=0$变为$E_{sub}p_{sub}=F_{sub}$；

使用线性方程组$E_{sub}p_{sub}=F_{sub}+v_{sub3}$计算坐标向量$p_{sub}$；该线性方程组的最小二乘解，$p_{sub}=\left(E_{sub}^T\Sigma_3^{-1}E_{sub}\right)^{-1}E_{sub}^T\Sigma_3^{-1}F_{sub}$（ρ的一次函数）；

定义误差函数J_1，$J_1=\sum_{i\in V}\left(\left(\left\|p_1-p_{1x}\right\|^2-1\right)^2+\left(\left\|p_1-p_{1y}\right\|^2-1\right)^2\right)=A_1\rho^4+B_1\rho^3+C_1\rho^2+D_1\rho+E_1$；

定义误差函数J_2，$J_2=\sum_{T_{ij}\in E}\left(\left\|p_i-p_j\right\|^2-\rho_{ij}\right)^2=A_2\rho^4+B_2\rho^3+C_2\rho^2+D_2\rho+E_2$；

相加得到误差函数J，$J=J_1+J_2=A\rho^4+B\rho^3+C\rho\ \ 2+D\rho+E$；

计算使得该误差函数最小的参数ρ，对J计算导数，$\frac{dJ}{d\rho}=4A\rho^3+3B\rho^2+2C\rho+D=0$；

使用三次方程的公式计算三个根ρ_1,ρ_2,ρ_3；舍弃三个根中的复数根，并使用函数J的二次导数选择最小值对应的根ρ_i；

参数ρ代入坐标向量p_{acc}；

6. 离群点去除

该论文中线性方程通过节点之间的位姿变换T_{ij}构建；如果图$G=[V,E]$中的位姿变换集合$E=\{T_{ij},i,j\in V\}$中存在离群点，对应的线性方程的系数也会出现离群点；线性方程组在使用最小二乘法计算时，每个线性方程都对应一个误差项；离群点会使得方程的解出现很大的偏移；

在非线性优化的 SLAM 方案中，每个位姿变换T_{ij}都会使用观测函数构建误差函数$h(T_{ij})$；然后由这些误差函数组成的误差项$e(T_{ij})=h\left(T_{ij}\right)^2$，通过将这些误差项相加得到误差函数 J；但是通常我们会在误差项外侧套一个鲁棒函数如下；

$$\rho(x)=\begin{cases}x^2, if\ -1<x<1\\ x, if\ x\geq 1\\ -x, if\ x\leq -1\end{cases}$$

鲁棒函数ρ在 0 附近的函数值上升很快；在超过某个范围后，鲁棒函数的函数值上升很慢；增加鲁棒函数后的误差函数 J 如下；

$$J = \sum_{T_{ij} \in E} \rho\left(h\left(T_{ij}\right)^2\right)$$

因为鲁棒函数的存在，即使某个位姿变换T_{ij}属于离群点，对应的误差项的数值大小也不会很大；因此离群点对整个误差函数的最小值的影响不大；并且原始的误差项$e(T_{ij}) = h\left(T_{ij}\right)^2$是非线性函数，增加鲁棒函数后的误差项$\rho\left(h\left(T_{ij}\right)^2\right)$仍然是非线性函数；我们仍然使用数值计算方法求解误差函数的最小值；

然而该论文中的线性方程，我们不能采用鲁棒函数的方案；如果在每个线性方程构成的误差项外侧增加鲁棒函数，那么就变成非线性函数；我们将位姿变换构建成线性方程的目的是为了减少计算量，如果使用非线性方程，就没法达到所需的目的；

论文[1]给出了一种在位姿变换中去除离群点的方法；该方法的具体步骤较多，因此我们简要介绍大致的方法；该算法输入图$G = [V, E]$中的所有位姿变换集合$E = \{T_{ij}, i, j \in V\}$，输出去除离群点的位姿变换集合$E' = \{T_{ij}, i, j \in V\}$；我们使用集合 E'中的所有位姿变换构建线性方程，就能避免离群点对方程的影响；

该算法需要对输入的位姿变换集合$E = \{T_{ij}, i, j \in V\}$中的每个位姿变换$T_{ij}$预设一个属于离群点的概率$p_{ij}$；这个概率并不是需要绝对精确的数值，我们可以使用位姿变换T_{ij}对应的方差σ_{ij}^2大致取值；比如可以采用如下的方案；

$$p_{ij} = \begin{cases} 0.1, \text{if } \sigma_{ij}^2 < \sigma_{\text{thres1}}^2 \\ 0.2, \text{if } \sigma_{\text{thres1}}^2 \le \sigma_{ij}^2 < \sigma_{\text{thres2}}^2 \\ 0.3, \text{if } \sigma_{\text{thres2}}^2 \le \sigma_{ij}^2 \end{cases}$$

概率p_{ij}在离群点去除算法中用于寻找不同节点（节点 i、节点 j）之间的路线；位姿变换T_{ij}表示路径中的边；那么算法会尽量寻找一条离群点少的边组成的路径实现从节点 i 到达节点 j；

该离群点去除算法的大致方法如下；设$G = [V, E]$中存在两个节点$p_1, p_2 \in V$；然后节点p_1, p_2之间存在三个旋转平移矩阵$T_{12}^1, T_{12}^2, T_{12}^3$；

同时节点p_1, p_2经过另外一个节点（如p_3, p_4, p_5）也能构成连线；那么p_1到p_2的位姿变换可以如下计算；

$$T_{12}^4 = T_{13}T_{32}$$

$$T_{12}^5 = T_{14}T_{42}$$

$$T_{12}^6 = T_{15}T_{52}$$

然后节点p_1, p_2经过另外两个节点（如p_6, p_7或p_8, p_9或p_{10}, p_{11}）可以构成连线；那么p_1到p_2的位姿变换计算如下；

$$T_{12}^7 = T_{16}T_{67}T_{72}$$

$$T_{12}^8 = T_{18}T_{89}T_{92}$$

$$T_{12}^9 = T_{1,10}T_{10,11}T_{11,2}$$

那么我们共得到 9 组节点 1 到节点 2 之间的位姿变换，$T_{12}^i, i \in [1,9]$；如果使用的所有位姿变换（集合 E）没有离群点，那么这些位姿变换T_{12}^i应该是非常近似的；如果使用的某个位姿变换是离群点，那么对应的位姿变换T_{12}^i和其他位姿变换相比，差别较大；我们可以将位姿变换T_{12}^i转化为平移向量和旋转向量，然后对每个分量都使用基于统计的 IQR 算法去除离群点；

7. 局部坐标系求解

局部坐标系的计算方法在论文[1]中已经给出，但是为了 SLAM 方案的完整性，此处我们仍然具体给出局部坐标系的计算方法；我们已经计算得到图$G = [V, E]$中每个节点在全局坐标系Σ_g中的坐标$p = [p_1^T, p_2^T, \dots, p_n^T] \in R^{2n}$；设节点 i 的全局坐标是$p_i = [x_i, y_i]^T$；设节点 i 的邻居节点为$j \in N_i$，其中节点 j 在全局坐标系$\Sigma_g$中坐标是$p_j = \left[x_j, y_j\right]^T$；节点 j 相对节点 i 的坐标是$p_j^r = \left[x_j - x_i, y_j - y_i\right]^T$；这些邻居节点在全局坐标系中的相对坐标向量为$P_i^n = \left[p_{j1}^r, \dots, p_{jn}^r\right] \in R^{2\times|N_i|}, j_1, \dots, j_n \in N_i$；设节点 i 到邻居节点$j \in N_i$的位姿变换是$T_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{bmatrix} \in R^{3\times3}, j \in N_i$，其中平移向量是$t_{ij} = \left[x_{ij}, y_{ij}\right]^T, j \in N_i$；那么节点 i 的邻居节点$j \in N_i$在局部坐标系$\Sigma_i$中的相对位置向量是$Q_i^n = \left[t_{ij1}, \dots, t_{ijn}\right] = \left[q_{ij1}, \dots, q_{ijn}\right] \in R^{2\times|N_i|}, j_1, \dots, j_n \in N_i$；

然后我们已经计算得到每个节点$i \in V$的虚拟节点$i_x, i_y \in V_v$在全局坐标系Σ_g中坐标，$p_{ix} = [x_{ix}, y_{ix}]^T, p_{iy} = \left[x_{iy}, y_{iy}\right]^T$；节点 i 的全局坐标是$p_i = [x_i, y_i]^T$，那么虚拟坐标$i_x, i_y$相对节点 i 的坐标如下；

$$p_{ix}^r = [x_{ix} - x_i, y_{ix} - y_i]^T$$

$$p_{iy}^r = \left[x_{iy} - x_i, y_{iy} - y_i\right]^T$$

虚拟节点i_x, i_y在全局坐标系中的相对坐标向量为$P_i^v = [p_{ix}^r, p_{iy}^r]$；虚拟节点$i_x, i_y$在局部坐标系$\Sigma_i$中的坐标为$p_{ix}^i = [1,0]^T, p_{iy}^i = [0,1]^T$；设虚拟节点$i_x, i_y$在局部坐标系$\Sigma_i$中的坐标向量为$Q_i^v = [p_{ix}^i, p_{iy}^i]$；将坐标向量$P_i^n$和$P_i^v$合并得到$P_i = [P_i^n, P_i^v] \in R^{2|N_i|+2}$；将坐标向量$Q_i^n$和$Q_i^v$合并得到$Q_i = [Q_i^n, Q_i^v] \in R^{2|N_i|+2}$；然后我们可以使用点集配准的方法计算节点 i 的局部坐标系的旋转矩阵R_i；

此处我们使用了节点 i 的邻居节点$j \in N_i$和虚拟节点i_x, i_y来构建坐标向量P_i和Q_i；虽然我们还可以将邻居节点$j \in N_i$的虚拟坐标$j_x, j_y, j \in N_i$也加入坐标向量中，不过此处我们并没有加入，原因如下；首先只要有三个节点（比如虚拟节点i_x, i_y）就可以得到旋转矩阵R_i；然后邻居节点$j \in N_i$自身的信息已经大致可以代表节点 j 的位置，再增加节点 j 的虚拟节点j_x, j_y的位置可能增加过多计算复杂性；

已知一组点在全局坐标系Σ_g中的位置向量$P_i \in R^{2|N_i|+2}$，并且已知这组点在局部坐标系Σ_i中的位置向量$Q_i \in R^{2|N_i|+2}$；我们需要寻找一个旋转矩阵R_i，构建如下误差函数；

$$E(R) = \sum_{j \in N_i} \left|\left|R_i p_{ij}^r - q_{ij}\right|\right|^2$$

我们需要使误差函数$E(R_i)$最小的旋转矩阵$R_i \in R^{2\times2}$；该问题属于点云配准问题，我们可以使用 ICP 算法的奇异值分解计算矩阵R_i；矩阵R_i表示将点集从全局坐标系Σ_g中变换到局部坐标系Σ_i的旋转矩阵；坐标系本身的旋转和点集的坐标旋转相反，那么局部坐标系Σ_i相对于全局坐标系Σ_g的旋转矩阵是$(R_i)^{-1}$；根据 ICP 算法中的奇异值分解方法，使得误差函数 E(R)取得最小值的旋转矩阵如下；

$$H = P_i Q_i^T$$

$$= U\Sigma V^T$$

$$R_i = VU^T$$

对矩阵$H \in R^{2\times2}$计算奇异值分解，得到正交矩阵U, V；旋转矩阵$R_i = VR^T$，然而如果点集P_i或者Q_i退化，那么计算得到的矩阵R_i可能是反射矩阵（$\det(R_i) = -1$）；我们可以通过如下步骤将反射矩阵R_i调整为旋转矩阵；

$$R_i = V\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} U^T$$

那么完整的旋转矩阵计算公式如下；

$$R_i = V\begin{bmatrix} 1 & 0 \\ 0 & |VU^T| \end{bmatrix} U^T$$

过程 4，局部坐标系求解

输入，节点i的坐标$p_i \in R^2$，节点i的邻居节点$j \in N_i$，邻居节点$j \in N_i$的坐标$p_j \in R^2$，邻居节点$j \in N_i$的位姿变换T_{ij}，虚拟节点i_x, i_y的坐标$p_{ix}, p_{iy} \in R^2$

输出，局部坐标系Σ_i相对全局坐标系Σ_g的旋转矩阵$(R_i)^{-1}$

邻居节点$j \in N_i$相对节点i的坐标为$p_j^r = \left[x_j - x_i, y_j - y_i\right]^T$；

邻居节点$j \in N_i$在全局坐标系Σ_g中的相对坐标是，$P_i^n = \left[p_{j1}^r, \ldots, p_{jn}^r\right] \in R^{2\times|N_i|}, j_1, \ldots, j_n \in N_i$；

邻居节点$j \in N_i$在局部坐标系Σ_i中的坐标是，$Q_i^n = \left[t_{ij1}, \ldots, t_{ijn}\right] = \left[q_{ij1}, \ldots, q_{ijn}\right] \in R^{2\times|N_i|}, j_1, \ldots, j_n \in N_i$；

虚拟节点i_x, i_y相对节点i的坐标为$p_{ix}^r = [x_{ix} - x_i, y_{ix} - y_i]^T, p_{iy} = \left[x_{iy} - x_i, y_{iy} - y_i\right]^T$；组合得到向量，$P_i^v = \left[p_{ix}^r, p_{iy}^r\right] \in R^{2\times2}$；

虚拟节点i_x, i_y在局部坐标系Σ_i中的坐标为$p_{ix}^i = [1,0]^T, p_{iy}^i = [0,1]^T$；组成向量为，$Q_i^v = \left[p_{ix}^i, p_{iy}^i\right] \in R^{2\times2}$；

坐标向量P_i^n和P_i^v合并为$P_i = [P_i^n, P_i^v] \in R^{2\times(2n+2)}$，坐标向量$Q_i^n$和$P_i^v$合并为$Q_i = [Q_i^n, Q_i^v] \in R^{2\times(2n+2)}$；

使用点集配准计算旋转矩阵R_i，矩阵$H = P_i Q_i^T$；

对矩阵H奇异值分解，$H = U\Sigma V^T$；

旋转矩阵为$R_i = V \times \mathrm{diag}([1, |VU^T|]) \times U^T$；

局部坐标系Σ_i相对全局坐标系Σ_g的旋转矩阵$(R_i)^{-1}$；

REFERENCES

Brown, F., Harris, M.G., and Other, A.N. (1998). Name of paper. In Name(s) of editor(s) (ed.), *Name of book in italics*, page numbers. Publisher, Place of publication.

Smith, S.E. (2004). *Name of book in italics*, page or chapter numbers if relevant. Publisher, Place of publication.

Smith, S.E. and Jones, L.Q. (2008). Name of paper. *Name of journal in italics*, volume (number), page numbers.

【待补充，调整格式后加入】