二维平面上基于虚拟坐标系的线性 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. 使用线性方程表示三角形

我们使用线性方程表示平面上任意三角形的形状；设平面上三角形 ijk 的三个点的坐标是$p_i, p_j, p_k \in R^2$；已知边(i,j)的长度为ρ_{ij}，边(i,k)的长度为ρ_{ik}，角$\angle jik$为θ_{jik}；使用上述三个条件我们可以得到全等三角形的约束；为了使用线性方程表示节点之间的位姿约束，我们将全等三角形约束进行放松，变成相似三角形约束$\rho_{jik} = \rho_{ik}/\rho_{ij}, \theta_{jik}$；这样我们只能保证三角形 ijk 的形状不变，但是三角形整体可以进行缩放；此时可以使用如下线性方程组表示这组相似三角形约束；

$$p_k - p_i = \rho_{jik} R_{jik}(p_j - p_i)$$

$$R_{jik} = \begin{bmatrix} \cos(\theta_{jik}) & -\sin(\theta_{jik}) \\ \sin(\theta_{jik}) & \cos(\theta_{jik}) \end{bmatrix}$$

其中，p_i, p_j和p_k分别是节点 i、节点 j 和节点 k 的坐标，属于需要约束的变量；$R(p_j - p_i)$表示对向量 ij 旋转θ_{jik}角度，得到与向量 ik 同方向的向量；然后$R(p_j - p_i)$与向量 ik 之间存在$\rho_{jik} = \rho_{ik}/\rho_{ij}$的比例；

在论文[1]中使用复数的线性方程给出了上述方程的等效形式；设节点 i 的复数坐标为$p_i^c = x_i + iy_i$，其中x_i是节点 i 坐标的 x 轴分量，y_i是节点 i 坐标的 y 轴分量；那么上述线性方程组可以使用如下的复数线性方程表示；

$$p_k^c - p_i^c = \omega_{jik}(p_j^c - p_i^c)$$

$$\omega_{jik} = \rho_{jik} e^{i\theta_{jik}}$$

我们设p_{ki}^c是向量 ik 的复数坐标形式，θ_{ki}是向量 ik 的角度；设p_{ji}^c是向量 ij 的复数坐标形式，θ_{ji}是向量 ij 的角度；

$$p_{ki}^c = p_k^c - p_i^c = \rho_{ki} e\hat{\ }(i\theta_{ki})$$

$$p_{ji}^c = p_j^c - p_i^c = \rho_{ji} e^{i\theta_{ji}}$$

我们将p_{ki}^c和p_{ji}^c替换上述方程中的p_k^c, p_i^c和p_j^c，得到如下方程；

$$\rho_{ki} e^{i\theta_{ki}} = \omega_{jik} \rho_{ji} e^{i\theta_{ji}}$$

$$\rho_{ki} e\hat{\ }(i\theta_{ki}) = \rho_{jik} \rho_{ji} e^{i(\theta_{ji} + \theta_{jik})}$$

上述方程可以转化为模长部分的约束和角度部分的约束如下；

$$\rho_{jik} = \rho_{ik}/\rho_{ij}$$

$$\theta_{ji} + \theta_{jik} = \theta_{ki}$$

这些约束刚好与节点 i、节点 j 和节点 k 的相似三角形约束相同；因此上述复数的线性方程同样可以等效的表示三角形 jik 的形状约束；无论使用线性方程还是复数的线性方程，都可以等效的表示三角形 jik 的约束；那么在后续文本中，我们使用复数的线性方程来表示三个节点之间的位姿约束；因为复数的线性方程相对来说稍微简洁，只有一个方程；那么任意节点 i 的坐标使用复数$p_i^c = x_i + iy_i$表示；

如果三角形 ijk 中任意两个节点重合，那么方程将退化；设节点 i 和节点 k 重合，那么$\rho_{ik} = 0, \rho_{jik} = \rho_{ik}/\rho_{ij} = 0$；参数$\omega_{jik} = \rho_{jik} e^{i\theta_{jik}} = 0$；那么复数的线性方程如下；

$$p_k^c - p_i^c = 0 \times (p_j^c - p_i^c)$$

节点 i 和节点 k 重合时，方程退化为$p_i^c = p_k^c$；同理如果节点 i 和节点 j 重合，方程退化为$p_i^c = p_j^c$；如果节点 j 和节点 k 重合，方程变为$p_j^c = p_k^c$；如果节点i, j, k重合，那么方程变成$p_i^c = p_j^c = p_k^c$；因此我们增加如下的判断步骤；

$$p_i^c = p_j^c, \text{if } \rho_{ij} < \epsilon_{thres}$$

$$p_i^c = p_k^c, \text{if } \rho_{ik} < \epsilon_{thres}$$

$$p_j^c = p_k^c, \text{if } \rho_{jk} < \epsilon_{thres}$$

$$p_i^c = p_j^c = p_k^c, \text{if } \rho_{ij}, \rho_{ik}, \rho_{jk}\text{中任意两个小于}\epsilon_{thres}$$

其中ϵ_{thres}是用于判断节点是否重合的阈值；

然后我们给出一个特定的可定位性问题的条件；在三角形 ijk 中，如果已知节点 i 和节点 k 的坐标，需要使用线性方程计算节点 j 的坐标，那么节点 j 的坐标可定位的条件如下；如果节点 i 和节点 k 不是重合节点，那么节点 j 是可定位的；如果节点 i 和节点 k 是重合节点，那么线性方程退化为$p_i^c = p_j^c$；这个线性方程与节点 j 的坐标没有关系，那么节点 j 的坐标不能使用线性方程计算；从几何上我们也可以得出相同的结论，如果节点 i 和节点 k 是重合节点，并且已知ρ_{ij}和ρ_{kj}，并不能确定节点 j 的坐标；因此此时节点 j 是位于一个圆上；

3. 使用线性方程表示位姿变换

3.1 构建线性方程

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

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$p_i = [x_i, y_i]^T$，节点 i 在二维平面上的复数坐标为$p_i^c = x_i + iy_i$；节点的复数坐标中实部表示节点坐标的 x 轴分量，虚部表示节点坐标的 y 轴分量；后续论文中我们使用节点的向量坐标（如p_i）加上标 c 为该节点的复数坐标（如p_i^c），并且 x 轴分量和 y 轴分量依次对应；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$p_i \in R^2$，复数坐标为$p_i^c \in C$；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_x的复数坐标为$p_{ix}^c \in C$；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$||p_{iy} - p_i|| = 1$；设虚拟节点i_y的复数坐标为$p_{iy}^c \in C$；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$p_j \in R^2$，复数坐标为$p_j^c \in C$；我们设虚拟节点j_x位于局部坐标系Σ_j的 x 轴，坐标为p_{jx}，并且满足$||p_{jx} - p_j|| = 1$；设虚拟节点j_x的复数坐标为$p_{jx}^c \in C$；设虚拟节点j_y位于局部坐标系Σ_j的 y 轴，坐标为p_j^c，并且满足$||p_{jy} - p_j|| = 1$；设虚拟节点j_y的复数坐标为$p_{jy}^c \in C$；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；

如果已知局部坐标系Σ_i中所有点的坐标（$p_i^c, p_{ix}^c, p_{iy}^c$），我们需要构建线性方程来表示局部坐标系$\Sigma_j$中所有点的坐标（$p_j^c, p_{jx}^c, p_{jy}^c$）；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；

已知局部坐标系Σ_i中两个轴的坐标p_{ix}^c, p_{iy}^c，我们首先构建局部坐标系Σ_j中虚拟节点坐标p_{jx}^c的线性方程；我们可以使用三角形的线性方程表示方法，构建三角形$j_x - i_x - i_y$的线性方程；

$$p_{iy}^c - p_{jx}^c = \omega_{ix,jx,iy}(p_{ix}^c - p_{jx}^c)$$

其中$p_{ix}^c, p_{iy}^c, p_{jx}^c \in C$分别是节点$i_x, i_y, j_x$的坐标；参数$\omega_{ix,jx,iy}$将在后续具体给出；

然后我们使用Σ_j中的虚拟节点坐标p_{jy}^c与Σ_i中的两个点p_{ix}^c, p_{iy}^c构建线性方程；使用三角形的表示方法，构建三角形$j_y - i_x - i_y$的线性方程；

$$p_{iy}^c - p_{jy}^c = \omega_{ix,jy,iy}(p_{ix}^c - p_{jy}^c)$$

其中$p_{ix}^c, p_{iy}^c, p_{jy}^c \in C$分别是节点$i_x, i_y, j_y$的坐标；参数$\omega_{ix,jy,iy}$将在后续具体给出；

然后我们使用Σ_i中的节点坐标p_i^c与Σ_i中的两个点p_{ix}^c, p_{iy}^c构建线性方程；使用三角形的表示方法，构建三角形$i - i_x - i_y$的线性方程；

$$p_{ix}^c - p_i^c = \omega_{ix,i,iy}(p_{ix}^c - p_i^c)$$

其中$p_{ix}^c, p_{iy}^c, p_i^c \in C$分别是节点$i_x, i_y, i$的坐标；参数$\omega_{ix,i,iy}$将在后续具体给出；

那么我们已经使用局部坐标系Σ_i中的两个虚拟节点i_x, i_y表示出局部坐标系Σ_j中的节点 j 和两个虚拟节点j_x, j_y；因为虚拟节点i_x, i_y分别是局部坐标系Σ_i两个轴上的单位向量，所以节点i_x和节点i_y不是重合节点（满足三角形的可定位条件）；如果我们已知局部坐标系Σ_i中两个虚拟节点的坐标p_{ix}^c, p_{iy}^c，可以计算局部坐标系Σ_j中节点 j 的坐标p_j^c和两个虚拟节点坐标p_{jx}^c, p_{jy}^c；以此类推，SLAM 问题中所有节点（关键帧）可以按照顺序连成一条线，如果我们已知第一个节点的位姿（两个虚拟节点坐标p_{1x}^c, p_{1y}^c），就可以计算任意节点 i 的位姿（两个虚拟节点坐标p_{ix}^c, p_{iy}^c）；

已知局部坐标系Σ_i中两个虚拟节点i_x, i_y的坐标，使用线性方程可以计算出局部坐标系Σ_j中两个虚拟节点j_x, j_y的坐标；那么我们讨论下面的反向表示问题；如果已知局部坐标系Σ_j中的两个虚拟节点j_x, j_y的坐标，是否可以使用上述线性方程计算局部坐标系Σ_i中两个虚拟节点i_x, i_y的坐标；我们可以从几何角度讨论上述线性方程的含义；三角形$i_x - i_y - j_x$对应的 1 个线性方程，这个线性方程在几何上的含义是节点i_x, i_y, j_x的形状（相似三角形），但是尺度是可以缩放的；同理三角形$i_x - i_y - j_y$的线性方程约束了节点i_x, i_y, j_y的形状，但是尺度可以缩放；因此 4 个节点（i_x, i_y, i_z, j_x）构成的几何图形的形状是确定的，尺度可以缩放；那么已知虚拟节点j_x, j_y的坐标，同样可以使用该线性方程组计算节点i_x, i_y的坐标；因此两个三角形（三角形$i_x - i_y - j_x$和三角形$i_x - i_y - j_y$）对应的两个线性方程，可以等效的表示位姿变换T_{ij}；

上述线性方程组中，我们使用局部坐标系Σ_i中的两个虚拟节点i_x, i_y表示局部坐标系Σ_j中的两个虚拟节点j_x, j_y；如果为了保证对称性，我们同样可以使用局部坐标系Σ_j中的两个虚拟节点j_x, j_y表示局部坐标系Σ_i中的两个虚拟节点i_x, i_y，并列出对应的 2 个线性方程；但是上面已经证明，使用节点i_x, i_y表示节点j_x, j_y的 2 个线性方程，已经可以等效的表示位姿变换T_{ij}；所以再次使用节点j_x, j_y表示节点i_x, i_y是冗余的；

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；因为每个节点$i \in V$的局部坐标系Σ_i中定义了两个虚拟节点i_x, i_y；因此我们定义虚拟节点集合如下，$V_v = \{i_x, i_y, i \in V\}$；这些虚拟节点的坐标是$p_{ix}, p_{iy} \in R^2, i_x, i_y \in V_v$；定义虚拟节点坐标向量为$p_v = [p_{1x}, p_{1y}, \dots, p_{nx}, p_{ny}] \in R^{2n}$；这些虚拟节点的复数坐标是$p_{ix}^c, p_{iy}^c \in C, i_x, i_y \in V_v$；定义虚拟节点的复数坐标向量为$p_v^c = [p_{1x}^c, p_{1y}^c, \dots, p_{nx}^c, p_{ny}^c] \in C^n$；

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 2 个线性方程；那么我们设所有位姿变换$T_{ij} \in E$对应的线性方程组如下；

$$Ap_v^c = b$$

此外我们还使用局部坐标系Σ_i中的两个虚拟节点i_x, i_y的坐标表示节点 i 的坐标；那么当计算得到所有虚拟节点的坐标p_v^c后，我们可以使用p_v^c计算图$G = [V, E]$中每个节点$i \in V$的坐标；定义所有节点$i \in V$的坐标向量为$p = [p_1, p_2, \dots, p_n] \in R^{2n}$；定义所有节点$i \in V$的复数坐标向量为$p^c = [p_1^c, p_2^c, \dots, p_n^c] \in C^n$；那么使用虚拟节点坐标$p_v^c$计算所有节点$i \in V$的坐标$p^c$的方程如下；

$$Cp^c = Dp_v^c$$

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, b, C, D

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

设所有节点的复数向量$p^c = [p_1^c, p_2^c, \dots, p_n^c]^T \in C^n$；

设所有虚拟节点向量$p_v = [p_{1x}, p_{1y}, \dots, p_{nx}, p_{ny}] \in R^{2\times 2n}$；

设所有虚拟节点的复数向量$p_v^c = [p_{1x}^c, p_{1y}^c, \dots, p_{nx}^c, p_{ny}^c]^T \in C^{2n}$；

For $T_{ij} \in V$:

  构建三角形$i_x - i_y - j_x$的线性方程
  $p_{iy}^c - p_{jx}^c = \omega_{ix,jx,iy}(p_{ix}^c - p_{jx}^c)$，加入线性方程组
  $Ap_v^c = b$；

  构建三角形$i_x - i_y - j_y$的线性方程
  $p_{iy}^c - p_{jy}^c = \omega_{ix,jy,iy}(p_{ix}^c - p_{jy}^c)$，加入线性方程组
  $Ap_v^c = b$；

For $i \in V$:

  构建三角形$i_x - i_y - i$的线性方程
  $p_{iy}^c - p_i^c = \omega_{ix,i,iy}(p_{ix}^c - p_i^c)$，加入线性方程组
  $Cp^c = Dp_v^c$；

3.2 计算三角形 ix-iy-jx 的线性方程

在之前章节中，我们使用三角形$i_x - i_y - j_x$构建的线性方程如下；

$$p_{iy}^c - p_{jx}^c = \omega_{ix,jx,iy}(p_{ix}^c - p_{jx}^c)$$

其中$p_{ix}^c, p_{iy}^c, p_{jx}^c \in C$分别是节点$i_x, i_y, j_x$的坐标；参数$\omega_{ix,jx,iy}$在之前章节中没有具体描述计算方法；下面我们给出这些参数的计算方法；在局部坐标系Σ_i中，虚拟节点i_x, i_y, i_z的坐标如下；

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

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

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

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

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

那么虚拟节点j_x在局部坐标系Σ_i中的位置如下；

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

下面我们给出局部坐标系Σ_i中向量$(j_x, i_x), (j_x, i_y)$的相对位置；

$$p_{jx,ix}^i = p_{ix}^i - p_{jx}^i = [x_{jx,ix}^i, y_{jx,ix}^i]^T$$

$$p_{jx,iy}^i = p_{iy}^i - p_{jx}^i = [x_{jx,iy}^i, y_{jx,iy}^i]^T$$

此处坐标的上下标较多，我们重新描述下各上下标的含义；坐标$p^i_{jx,ix}$中，下标j_x, i_x表示这是向量(j_x, i_x)的坐标；上标 i 表示该坐标位于局部坐标系Σ_i中；下面我们计算参数$\omega_{ix,jx,iy}$；其中坐标$p^i_{jx,ix}, p^i_{jx,iy}$之间的长度如下；

$$\rho_{jx,ix} = ||p^i_{jx,ix}||$$

$$\rho_{jx,iy} = ||p^i_{jx,iy}||$$

模长的比例如下；

$$\rho_{ix,jx,iy} = \frac{\rho_{jx,iy}}{\rho_{jx,ix}}$$

角度计算如下；

$$\theta_{ix,jx,iy} = \arccos\left(\frac{p^i_{jx,ix} p^i_{jx,iy}}{\left|\left|p^i_{jx,ix}\right|\right|\left|\left|p^i_{jx,iy}\right|\right|}\right)$$

参数$\omega_{ix,jx,iy}$如下；

$$\omega_{ix,jx,iy} = \rho_{ix,jx,iy} e^{i\theta_{ix,jx,iy}}$$

节点i_x和节点j_x重合时，方程退化为$p^c_{ix} = p^c_{jx}$；同理如果节点i_y和节点j_x重合，方程退化为$p^c_{iy} = p^c_{jx}$；节点i_x和节点i_y是局部坐标系Σ_i的两个轴的单位向量，不会重合；我们增加如下的判断步骤；

$$p^c_{ix} = p^c_{jx}, \text{if } \rho_{jx,ix} < \epsilon_{thres}$$

$$p^c_{iy} = p^c_{jx}, \text{if } \rho_{jx,iy} < \epsilon_{thres}$$

其中ϵ_{thres}是用于判断节点是否重合的阈值；使用节点 i 到节点 j 的位姿变换T_{ij}可以计算三角形$i_x - i_y - j_x$的线性方程；具体的伪代码如下；

过程 2，计算三角形$i_x - i_y - j_x$的线性方程

输入，位姿变换T_{ij}，虚拟节点j_x在局部坐标系Σ_j中的坐标$p^j_{jx} = [1,0]^T$，距离阈值ϵ_{thres}

输出，节点i_x, i_y, j_x的线性方程

虚拟节点i_x, i_y在局部坐标系Σ_i中的位置，$p^i_{ix} = [1,0]^T, p^i_{iy} = [0,1]^T$；

虚拟节点j_x在局部坐标系Σ_i中的位置，$p^i_{jx} = R_{ij} p^j_{jx} + t_{ij}$；

向量$(j_x, i_x), (j_x, i_y)$在局部坐标系Σ_i中的相对位置；

$p^i_{jx,ix} = p^i_{ix} - p^i_{jx} = \left[x^i_{jx,ix}, y^i_{jx,ix}\right]^T$

$p^i_{jx,iy} = p^i_{iy} - p^i_{jx} = \left[x^i_{jx,iy}, y^i_{jx,iy}\right]^T$

向量坐标$p^i_{jx,ix}, p^i_{jx,iy}$的模长；

$\rho_{jx,ix} = \left|\left|p^i_{jx,ix}\right|\right|, \rho_{jx,iy} = \left|\left|p^i_{jx,iy}\right|\right|$

If$(\rho_{jx,ix} < \epsilon_{thres})$:

  线性方程退化为，$p^c_{ix} = p^c_{jx}$；

Elseif$(\rho_{jx,iy} < \epsilon_{thres})$:

  线性方程退化为，$p^c_{iy} = p^c_{jx}$；

Else:

  模长的比例为，$\rho_{ix,jx,iy} = \rho_{jx,iy}/\rho_{jx,ix}$；

  计算角度，$\theta_{ix,jx,iy} = \arccos\left(\frac{p^i_{jx,ix} p^i_{jx,iy}}{\left|\left|p^i_{jx,ix}\right|\right|\left|\left|p^i_{jx,iy}\right|\right|}\right)$

  计算参数，$\omega_{ix,jx,iy} = \rho_{ix,jx,iy} e^{i\theta_{ix,jx,iy}}$；

  线性方程，$p^c_{iy} - p^c_{jx} = \omega_{ix,jx,iy}(p^c_{ix} - p^c_{jx})$；

3.3 计算三角形 ix-iy-jy 和三角形 ix-iy-i 的线性方程

在之前章节中，我们使用三角形$i_x - i_y - j_y$构建的线性方程如下；

$$p^c_{iy} - p^c_{jy} = \omega_{ix,jy,iy}(p^c_{ix} - p^c_{jy})$$

其中$p^c_{ix}, p^c_{iy}, p^c_{jy} \in C$分别是节点$i_x, i_y, j_y$的坐标；参数$\omega_{ix,jy,iy}$在之前章节中没有具体描述计算方法；下面我们给出伪代码；

过程 3，计算三角形$i_x - i_y - j_y$的线性方程

输入，位姿变换T_{ij}，虚拟节点j_y在局部坐标系Σ_j中的坐标$p^j_{jy} = [0,1]^T$，距离阈值ϵ_{thres}

输出，节点i_x, i_y, j_y的线性方程

虚拟节点i_x, i_y在局部坐标系Σ_i中的位置，$p^i_{ix} = [1,0]^T, p^i_{iy} = [0,1]^T$；

虚拟节点j_y在局部坐标系Σ_i中的位置，$p^i_{jy} = R_{ij} p^j_{jy} + t_{ij}$；

向量$(j_y, i_x), (j_y, i_y)$在局部坐标系Σ_i中的相对位置；

$p^i_{jy,ix} = p^i_{ix} - p^i_{jy} = \left[x^i_{jy,ix}, y^i_{jy,ix}\right]^T$

$p^i_{jy,iy} = p^i_{iy} - p^i_{jy} = \left[x^i_{jy,iy}, y^i_{jy,iy}\right]^T$

向量坐标$p^i_{jy,ix}, p^i_{jy,iy}$的模长；

$\rho_{jy,ix} = \left|\left|p^i_{jy,ix}\right|\right|, \rho_{jy,iy} = \left|\left|p^i_{jy,iy}\right|\right|$

If$(\rho_{jy,ix} < \epsilon_{thres})$:

  线性方程退化为，$p^c_{ix} = p^c_{jy}$；

Elseif$(\rho_{jy,iy} < \epsilon_{thres})$:

  线性方程退化为，$p^c_{iy} = p^c_{jy}$；

Else:

  模长的比例为，$\rho_{ix,jy,iy} = \rho_{jy,iy}/\rho_{jy,ix}$；

  计算角度，$\theta_{ix,jy,iy} = \arccos\left(\frac{p^i_{jy,ix} p^i_{jy,iy}}{\left|\left|p^i_{jy,ix}\right|\right|\left|\left|p^i_{jy,iy}\right|\right|}\right)$

  计算参数，$\omega_{ix,jy,iy} = \rho_{ix,jy,iy} e^{i\theta_{ix,jy,iy}}$；

线性方程，$p_{iy}^c - p_{jy}^c = \omega_{ix,jx,iy}(p_{ix}^c - p_{jy}^c)$；

在之前章节中，我们使用三角形$i_x - i_y - i$构建的线性方程如下；

$$p_{iy}^c - p_i^c = \omega_{ix,i,iy}(p_{ix}^c - p_i^c)$$

其中$p_{ix}^c, p_{iy}^c, p_i^c \in C$分别是节点$i_x, i_y, i$的坐标；参数$\omega_{ix,i,iy}$在之前章节中没有具体描述计算方法；下面我们给出伪代码；

过程 4，计算三角形$i_x - i_y - i$的线性方程

输入，位姿变换T_{ij}，节点i在局部坐标系Σ_i中的坐标$p_i^i = [0,0]^T$

输出，节点i_x, i_y, j_y的线性方程

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

向量$(i, i_x), (i, i_y)$在局部坐标系Σ_i中的相对位置；

$p_{i,ix}^i = p_{ix}^i - p_i^i = \left[x_{i,ix}^i, y_{i,ix}^i\right]^T$

$p_{i,iy}^i = p_{iy}^i - p_i^i = \left[x_{i,iy}^i, y_{i,iy}^i\right]^T$

向量坐标$p_{i,ix}^i, p_{i,iy}^i$的模长；

$\rho_{i,ix} = \left\|p_{i,ix}^i\right\| = 1, \rho_{i,iy} = \left\|p_{i,iy}^i\right\| = 1$

模长的比例为，$\rho_{ix,i,iy} = \rho_{i,iy}/\rho_{i,ix}$；

计算角度，$\theta_{ix,i,iy} = \arccos\left(\frac{p_{i,ix}^i p_{i,iy}^i}{\left\|p_{i,ix}^i\right\|\left\|p_{i,iy}^i\right\|}\right) = \frac{\pi}{2}$；

计算参数，$\omega_{ix,i,iy} = \rho_{ix,i,iy} e^{i\theta_{ix,i,iy}} = e^{\frac{i\pi}{2}}$；

线性方程，$p_{iy}^c - p_i^c = \omega_{ix,i,iy}(p_{ix}^c - p_i^c)$；

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}，可以转换为两组虚拟节点（虚拟节点i_x, i_y和虚拟节点j_x, j_y）之间的线性方程；

4.2 GPS 传感器

GPS 传感器数据的处理方式在论文[1]中已经给出，为了 SLAM 方案的完整性，我们仍然具体给出 GPS 数据的处理方法；

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，$T_{i,i+n} = T_{i,i+1}T_{i+1,i+2}\ldots T_{i+n-1,i+n}$；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；

在图$G = [V, E]$中共有 n 个节点（关键帧）；GPS 数据的更新频率通常远低于相机传感器或者 IMU 传感器；所以不是每个节点都有对应的 GPS 数据，我们设拥有 GPS 数据的节点序号集合为$G_{index} = \{g_i, i \in [1, m]\}$；序号集合中每个序号值$g_i \in [1, n]$，然后该集合中共有 m 个元素，对应 m 个 GPS 经纬度数据；设 m 个 GPS 经纬度数据的集合为$G = \{pos_{g(i)} = \left[lon_{g(i)}, lat_{g(i)}\right]^T, g(i) \in [1, n], i \in [1, m]\}$；

设第 i 个 GPS 经纬度数据为$pos_{g(i)} = \left[lon_{g(i)}, lat_{g(i)}\right]^T$，第 j 个 GPS 经纬度数据为$pos_{g(j)} = \left[lon_{g(j)}, lat_{g(j)}\right]^T$；我们可以通过球面距离公式或者 UTM 平面坐标，将节点g_i和节点g_j的经纬度数据转化为相对平移向量；

$$t_{g(i),g(j)} = f(pos_{g(i)}, pos_{g(j)})$$

其中，函数 f 将两个节点的经纬度$pos_{g(i)}, pos_{g(j)}$转化为平移向量$t_{g(i),g(j)}$；

获得 GPS 传感器的数据后，我们会构建两种类型的线性方程，分别是相邻 GPS 数据的线性方程和跨节点 GPS 数据的线性方程；下面首先介绍相邻 GPS 数据的线性方程；

已知第 i 个 GPS 数据是$pos_{g(i)}$，第$i+1$个 GPS 数据是$pos_{g(i+1)}$，第$i+2$个 GPS 数据是$pos_{g(i+2)}$；使用节点g_{i+1}作为中间节点，那么节点g_{i+1}到节点g_i的相对位置是$t_{g(i+1),g(i)} = f(pos_{g(i+1)}, pos_{g(i)})$；节点$g_{i+1}$到节点$g_{i+2}$的相对位置是$t_{g(i+1),g(i+2)} = f(pos_{g(i+1)}, pos_{g(i+2)})$；节点$g_{i+1}$到节点$g_{i+2}$的相对位置是$t_{g(i+1),g(i+2)} = f(pos_{g(i+2)}, pos_{g(i+1)})$；首先需要保证这三个节点没有重合的节点，判断方式如下；

$$\rho_{g(i+1),g(i)} = \left\|t_{g(i+1),g(i)}\right\| > \epsilon_{thres}$$

$$\rho_{g(i+1),g(i+2)} = \left\|t_{g(i+1),g(i+2)}\right\| > \epsilon_{thres}$$

$$\rho_{g(i),g(i+2)} = \left\|t_{g(i),g(i+2)}\right\| > \epsilon_{thres}$$

在保证没有重合节点后，我们可以构建如下线性方程；

$$p^c_{g(i+2)} - p^c_{g(i+1)} = \omega_{i+1}(p^c_{g(i)} - p^c_{g(i+1)})$$

$$\omega_{i+1} = \rho_{i+1} e^{i\theta_{i+1}}$$

其中参数ρ_{i+1}和θ_{i+1}可以如下计算；

$$\rho_{i+1} = \frac{\rho_{g(i+1),g(i+2)}}{\rho_{g(i+1),g(i)}}$$

$$\cos(\theta_{i+1}) = \frac{t_{g(i+1),g(i)} t_{g(i+1),g(i+2)}}{\left\|t_{g(i+1),g(i)}\right\| \left\|t_{g(i+1),g(i+2)}\right\|}$$

此外我们还可以构建跨节点 GPS 数据的线性方程；任意两个相邻的 GPS 数据结合计算平移向量$t_{g(i),g(i+1)}$并不能完整的使用 GPS 提供的信息；因为 GPS 数据的优势在于相距很远的两个节点之间的相对位置测量仍然能保持很好的精度；所以除了构建上述线性方程，我们还需要对距离很远的节点使用 GPS 数据构建线性方程；

对此我们需要在 GPS 数据集合 G 中寻找这样的三个节点（节点g_i、节点g_j和节点g_k）；这三个节点中没有重合节点；

$$\rho_{g(i),g(j)} > \epsilon_{thres}$$

$$\rho_{g(i),g(k)} > \epsilon_{thres}$$

$$\rho_{g(j),g(k)} > \epsilon_{thres}$$

并且这三个节点互相之间的距离近似，或者我们使用如下公式描述；

$$\frac{\rho_{g(i),g(j)}}{\rho_{g(i),g(k)}} < p_{thres}, \text{if } \rho_{g(i),g(j)} \geq \rho_{g(i),g(k)}$$

$$\frac{\rho_{g(i),g(k)}}{\rho_{g(i),g(j)}} < p_{thres}, \text{if } \rho_{g(i),g(j)} < \rho_{g(i),g(k)}$$

$$\frac{\rho_{g(j),g(i)}}{\rho_{g(j),g(k)}} < p_{thres}, \text{if } \rho_{g(j),g(i)} \geq \rho_{g(j),g(k)}$$

$$\frac{\rho_{g(j),g(k)}}{\rho_{g(j),g(i)}} < p_{thres}, \text{if } \rho_{g(j),g(i)} < \rho_{g(j),g(k)}$$

$$\frac{\rho_{g(k),g(i)}}{\rho_{g(k),g(j)}} < p_{thres}, \text{if } \rho_{g(k),g(i)} \geq \rho_{g(k),g(j)}$$

$$\frac{\rho_{g(k),g(j)}}{\rho_{g(k),g(i)}} < p_{thres}, \text{if } \rho_{g(k),g(i)} < \rho_{g(k),g(j)}$$

其中p_{thres}表示两边的比例；我们需要保证三角形$g_i g_j g_k$中任意两边的比例小于p_{thres}（长边比短边）；

因为节点g_i、节点g_j和节点g_k在整个地图上随机寻找的，通常三个点之间的距离都较大；那么使用这三个点构成的相似三角形构建线性方程，会对整个地图的形状进行很好的约束；因为为了尽量满足这三个点构成的相似三角形形状，那么需要对节点g_i、节点g_j和节点g_k的位置进行均匀且较大的移动；

对于如何在 GPS 数据集合 G 中寻找尽量多的这样的三角形组合；我们采用如下方案；在 GPS 数据集合中随机寻找三个点，判断是否满足上述边的比例条件，如果不满足则重新寻找；我们可以设定寻找的最大次数f_{max}和所需的三角形个数Δ_{req}；得到满足要求的三个节点g_i, g_j, g_k后，我们可以构建如下线性方程；

$$p^c_{g(k)} - p^c_{g(i)} = \omega_{g(j),g(i),g(k)}(p^c_{g(j)} - p^c_{g(i)})$$

$$\omega_{g(j),g(i),g(k)} = h(t_{g(i),g(j)}, t_{g(i),g(k)})$$

我们将上述线性方程构成方程组，记为如下；

$$Ep^c = 0$$

其中$p^c \in C^n$是所有节点$i \in V$的平面坐标向量；对于 GPS 数据集合 G，我们会构建两种类型的线性方程，分别是相邻 GPS 数据的线性方程和跨节点 GPS 数据的线性方程；下面首先介绍相邻 GPS 数据的线性方程；下面分别使用伪代码展示具体方法；

过程 5，相邻 GPS 数据构建线性方程

输入，GPS数据集合$G = \{pos_{g(i)} = [lon_{g(i)}, lat_{g(i)}]^T, g_i \in [1, n], i \in [1, m]\}$，重合节点距离$\epsilon_{thres}$

输出，节点位置相邻$p^c = [p^c_1, p^c_2, \ldots, p^c_n]^T$，矩阵E

For i in $[1, m-2]$:

  节点g_i, g_{i+1}的相对平移$t_{g(i),g(i+1)} = f(pos_{g(i)}, pos_{g(i+1)})$，距离为$\rho_{g(i),g(i+1)}$；

  节点g_{i+1}, g_{i+2}的相对平移$t_{g(i+1),g(i+2)} = f(pos_{g(i+1)}, pos_{g(i+2)})$，距离为$\rho_{g(i+1),g(i+2)}$；

  节点g_i, g_{i+2}的相对平移$t_{g(i),g(i+2)} = f(pos_{g(i)}, pos_{g(i+2)})$，距离为$\rho_{g(i),g(i+2)}$；

  If($\rho_{g(i+1),g(i)} > \epsilon_{thres}$):

    break

If($\rho_{g(i+1),g(i+2)} > \epsilon_{thres}$):

break

If($\rho_{g(i),g(i+2)} > \epsilon_{thres}$):

break

构建线性方程，$p^c_{g(i+2)} - p^c_{g(i+1)} = \omega_{i+1}(p^c_{g(i)} - p^c_{g(i+1)})$，加入矩阵E；

过程 6，跨节点 GPS 数据构建线性方程

输入，GPS数据集合$G = \{pos_{g(i)} = [lon_{g(i)}, lat_{g(i)}]^T, g_i \in [1,n], i \in [1,m]\}$，边比例$k_{thres}$，最大寻找次数$f_{max}$，所需三角形个数$\Delta_{req}$

输出，节点位置相邻$p^c = [p^c_1, p^c_2, \ldots, p^c_n]^T$，矩阵E

迭代次数$iter_{num} = 0$，找到三角形个数$\Delta_{num} = 0$

For $iter_{num} < f_{max}$ and $\Delta_{num} < \Delta_{req}$

$iter_{num} = iter_{num} + 1$；

GPS数据集合G中，随机选择三个点g_i, g_j, g_k；

节点g_i, g_j的相对平移$t_{g(i),g(j)} = f(pos_{g(i)}, pos_{g(j)})$，距离为$\rho_{g(i),g(j)}$；

节点g_i, g_k的相对平移$t_{g(i),g(k)} = f(pos_{g(i)}, pos_{g(k)})$，距离为$\rho_{g(i),g(k)}$；

节点g_j, g_k的相对平移$t_{g(j),g(k)} = f(pos_{g(j)}, pos_{g(k)})$，距离为$\rho_{g(j),g(k)}$；

If($\rho_{g(i),g(j)} < \epsilon_{thres}$):

break

If($\rho_{g(i),g(k)} < \epsilon_{thres}$):

break

If($\rho_{g(j),g(k)} < \epsilon_{thres}$):

break

If$\left(\frac{\rho_{g(i),g(j)}}{\rho_{g(i),g(k)}} > p_{thres} \text{ and } \rho_{g(i),g(j)} \geq \rho_{g(i),g(k)}\right)$:

break

If$\left(\frac{\rho_{g(i),g(k)}}{\rho_{g(i),g(j)}} > p_{thres} \text{ and } \rho_{g(i),g(j)} < \rho_{g(i),g(k)}\right)$:

break

If$\left(\frac{\rho_{g(j),g(i)}}{\rho_{g(j),g(k)}} > p_{thres} \text{ and } \rho_{g(j),g(i)} \geq \rho_{g(j),g(k)}\right)$:

break

If$\left(\frac{\rho_{g(j),g(k)}}{\rho_{g(j),g(i)}} > p_{thres} \text{ and } \rho_{g(j),g(i)} < \rho_{g(j),g(k)}\right)$:

break

If$\left(\frac{\rho_{g(k),g(i)}}{\rho_{g(k),g(j)}} < p_{thres} \text{ and } \rho_{g(k),g(i)} \geq \rho_{g(k),g(j)}\right)$:

break

If$\left(\frac{\rho_{g(k),g(j)}}{\rho_{g(k),g(i)}} < p_{thres} \text{ and } \rho_{g(k),g(i)} < \rho_{g(k),g(j)}\right)$:

break

使用节点g_i、节点g_j和节点g_k构建线性方程，$p^c_{g(k)} - p^c_{g(i)} = \omega_{g(j),g(i),g(k)}(p^c_{g(j)} - p^c_{g(i)})$，该方程加入矩阵E；

$\Delta_{num} = \Delta_{num} + 1$；

4.3 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^2_{ij}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$i_x, i_y, i_z \in V_v$之间的线性方程如下；

$$Ap^c_v = b + v_1$$

$$Cp^c = Dp^c_v + v_2$$

$$Ep^c = 0 + v_3$$

其中误差向量$v_1 \sim N(0, \Sigma_1), v_2 \sim N(0, \Sigma_2), v_3 \sim N(0, \Sigma_3)$；我们忽略各个线性方程之间的联系，那么协方差矩阵$\Sigma_i$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^2_{ij}作为该线性方程的方差；对于误差向量v_2的方差Σ_2，可以设为 0；因为这个线性方程组表示使用节点$i \in V$中的虚拟坐标i_x, i_y表示节点 i 的坐标，并没有引入观测数据；对于误差向量v_3，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$\sigma^2 = \sigma^2_{ij} + \sigma^2_{ik}$；

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；每个节点$i \in V$的局部坐标系Σ_i中定义了两个虚拟节点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$p_v = [p_{1x}, p_{1y}, \ldots, p_{nx}, p_{ny}] \in R^{2\times n}$；虚拟节点的复数向量是$p^c_v = [p^c_{1x}, p^c_{1y}, \ldots, p^c_{nx}, p^c_{ny}] \in C^n$；

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 2 个线性方程；那么我们设所有位姿变换$T_{ij}\in E$对应的线性方程组如下；

$$Ap_v^c = b$$

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；设复数坐标向量$p_v^c\in C^{2n}$中，虚拟节点1_x的复数坐标是$p_{1x}^c = 1 + i\times 0$，虚拟节点1_y的复数坐标是$p_{1y}^z = 0 + i\times 1$；根据节点可定位性条件，只要确定一个节点的所有坐标（如节点 1、节点1_x、节点1_y的坐标），我们可以计算得到所有其他节点的坐标；那么将p_{1x}^c, p_{1y}^c代入坐标向量p_v^c，然后使用线性方程$Ap_v^c = b$计算得到向量p_v^c中其他节点的坐标；

平面方向上的线性方程$Ap_v^c = b$在几何上的含义是所有节点$i\in V$构成的图形的形状是确定的，但是尺度是可以缩放的；因为每个线性方程等效于一个相似三角形的约束；如果使用p_{1x}^c, p_{1y}^c两个平面坐标计算p_v^c中其他节点的坐标，那么相当于使用p_{1x}^c, p_{1y}^c两个平面坐标确定了整个平面图形的尺度；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中两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$\Sigma_n$中两个轴的向量（节点 n、节点$n_x$和节点$n_y$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 的复数坐标为$p_1^c = 0 + i\times 0$，虚拟节点1_x的复数坐标是$p_{1x}^c = \rho + i\times 0$，虚拟节点$1_y$的复数坐标是$p_{1y}^c = 0 + i\times\rho$；设虚拟节点$i_x, i_y\in V_v/(1_x, 1_y)$的复数坐标向量是$p_{sub}^c = \left[p_{2x}^c, p_{2y}^c, \ldots, p_{nx}^c, p_{ny}^c\right]\in C^{2n-2}$；将坐标$p_{1x}^c, p_{1y}^c$的复数坐标代入方程，线性方程组$Ap_v^c = b + v_1$变成如下；

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

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；

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

复数坐标向量p_{sub}^c表示从节点 2 到节点 n 的$2(n-1)$个虚拟节点的复数坐标，并且每个复数坐标（如p_{ix}^c, p_{iy}^c）中的实部和虚部都是未知量ρ的一次函数；

$$p_{ix}^c = (a_{ix}\rho + b_{ix}) + i(c_{ix}\rho + d_{ix})$$
$$p_{iy}^c = (a_{iy}\rho + b_{iy}) + i(c_{iy}\rho + d_{iy})$$

同时节点 1 的两个虚拟节点$1_x, 1_y$的复数坐标中，复数坐标的实部和虚部都是未知量ρ的一次函数；因此所有虚拟节点$i_x, i_y\in V_v$的复数坐标中，复数坐标的实部和虚部都是未知量ρ的一次函数；

计算得到所有虚拟节点的坐标p_v后，我们可以使用p_v计算图$G=[V, E]$中每个节点$i\in V$的坐标；所有节点$i\in V$的坐标向量为$p=[p_1, p_2, \ldots, p_n]\in R^{2\times n}$；所有节点$i\in V$的复数坐标向量为$p^c = [p_1^c, p_2^c, \ldots, p_n^c]\in C^n$；那么使用虚拟节点坐标$p_v^c$计算所有节点$i\in V$的坐标$p^c$的方程如下；

$$Cp^c = Dp_v^c$$

上述方程的最小二乘解如下；

$$p^c = (C^TC)^{-1}C^TDp_v^c$$

其中坐标向量$p^c\in C^n$中每个节点$i\in V$的复数坐标中，每个复数坐标的实部和虚部都是未知量ρ的一次函数；

$$p_i^c = (a_i\rho + b_i) + i(c_i\rho + d_i)$$

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 到节点i_y的模长应该为 1；

$$\begin{aligned}J_1 &= \sum\nolimits_{i\in V}\left(\left(\left\|p_i - p_{ix}\right\|^2 - 1\right)^2 + \left(\left\|p_i - p_{iy}\right\|^2 - 1\right)^2\right)\\ &= \sum\nolimits_{i\in V}\Big(((a_i\rho - a_{ix}\rho)^2 + (b_i - b_{ix})^2 - 1)^2\\ &+ \left(\left(a_i\rho - a_{iy}\rho\right)^2 + \left(b_i - b_{iy}\right)^2 - 1\right)^2\\ &+ ((a_i\rho - a_{iz}\rho)^2 + (b_i - b_{iz})^2 - 1)^2\Big)\\ &= A_1\rho^4 + B_1\rho^3 + C_1\rho^2 + D_1\rho + E_1\end{aligned}$$

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$T_{ij}\in R^{3\times 3}$，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ρ的一次函数）；我们使用两个模长的差异定义误差函数；

$$\begin{aligned}J_2 &= \sum\nolimits_{T_{ij}\in E}\left(\left\|p_i - p_j\right\|^2 - \rho_{ij}^2\right)^2\\ &= \sum\nolimits_{T_{ij}\in E}\left(\left(a_i\rho - a_j\rho\right)^2 + \left(b_i - b_j\right)^2 - \rho_{ij}^2\right)^2\\ &= A_2\rho^4 + B_2\rho^3 + C_2\rho^2 + D_2\rho + E_2\end{aligned}$$

将误差函数J_1和误差函数J_2相加，得到误差函数 J；

$$\begin{aligned}J &= J_1 + J_2\\ &= A\rho^4 + B\rho^3 + C\rho^2 + D\rho + E\end{aligned}$$

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$\rho_i, i \in [1,3]$的二次导数为正，那么根ρ_i是函数 J 的最小值或局部最小值（使用函数值判断）；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$p_v^c \in C^{2n}, p^c \in C^n$；计算地图尺度和节点坐标的伪代码如下；

过程 7，计算地图尺度和节点坐标（不使用 GPS 数据）

输入，线性方程组$Ap_v^c = b, Cp^c = Dp_v^c$，方差矩阵$\Sigma_1, \Sigma_3$

输出，虚拟节点坐标向量$p_v^c \in C^{2n}$，节点坐标向量$p \in C^n$，地图尺度ρ

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

设节点$i \in V$的复数坐标向量为$p^c = [p_1^c, p_2^c, \dots, p_n^c] \in C^n$；

设虚拟节点坐标向量为$p_v = \left[p_{1x}, p_{1y}, \dots, p_{nx}, p_{ny}\right] \in R^{2\times 2n}$；

设虚拟节点的复数坐标向量为$p_v^c = \left[p_{1x}^c, p_{1y}^c, \dots, p_{nx}^c, p_{ny}^c\right] \in C^{2n}$；

设设节点1的复数坐标为$p_1^c = 0 + i \times 0$，虚拟节点1_x的复数坐标是$p_{1x}^c = \rho + i \times 0$，虚拟节点$1_y$的复数坐标是$p_{1y}^c = 0 + i \times \rho$；

设虚拟节点$i_x, i_y \in V_v/(1_x, 1_y)$的复数坐标向量是$p_{sub}^c = \left[p_{2x}^c, p_{2y}^c, \dots, p_{nx}^c, p_{ny}^c\right] \in C^{2n-2}$；

将坐标p_{1x}^c, p_{1y}^c 的复数坐标代入方程，线性方程组$Ap_v^c =$变为$A_{sub}p_{sub}^c = b_{sub}$；

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

设虚拟节点$i_x, i_y, i_z \in V_v/(1_x, 1_y, 1_z)$的垂直坐标向量是$p_{sub}^z = \left[p_{2x}^z, p_{2y}^z, p_{2z}^z, \dots, p_{nx}^z, p_{ny}^z, p_{nz}^z\right]^T \in R^{3n-3}$；

将坐标$p_{1x}^z, p_{1y}^z, p_{1z}^z$的垂直坐标代入方程，线性方程组$A^zp_v^z = b^z$变为$A_{sub}^zp_{sub}^z = b_{sub}^z$；

使用线性方程组$A_{sub}^zp_{sub}^z = b_{sub}^z$计算虚拟节点坐标的垂直分量$p_{sub}^z$；该线性方程的最小二乘解，$p_{sub}^z = \left({A_{sub}^z}^T\Sigma_2^{-1}A_{sub}^z\right)^{-1}{A_{sub}^z}^T\Sigma_2^{-1}b_{sub}^z$（固定数值）；

使用线性方程组$C^{xy}p^{xy} = D^{xy}p_v^{xy}$计算节点$i \in V$的平面分量$p^{xy}$；该线性方程组的最小二乘解，$p^{xy} = \left({C^{xy}}^TC^{xy}\right)^{-1}{C^{xy}}^TD^{xy}p_v^{xy}$（$\rho$的一次函数）；

使用线性方程组$C^zp^z = D^zp_v^z$计算节点$i \in V$的垂直分量p^z；该线性方程组的最小二乘解，$p^z = \left({C^z}^TC^z\right)^{-1}D^zp_v^z$（固定数值）；

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

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

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p_v^{xy}, p^{xy}，计算坐标p^v, p；

5.2 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$i_x, i_y, i_z \in V_v$之间的线性方程如下；

$$A^{xy}p_v^{xy} = b^{xy} + v_1$$

$$A^zp_v^z = b^z + v_2$$

$$C^{xy}p^{xy} = D^{xy}p_v^{xy} + v_3$$

$$C^zp^z = D^zp_v^z + v_4$$

我们设定如下坐标$p_1^{xy} = 0 + i \times 0, p_{1x}^{xy} = \rho + i \times 0, p_{1y}^{xy} = 0 + i \times \rho, p_1^z = 0, p_{1z}^{xy} = 0 + i \times 0, p_1^z = 0, p_{1x}^z = 0, p_{1y}^z = 0, p_{1z}^z = 1$；使用方程(1)和(2)计算坐标向量$p_v^{xy}$和$p_v^z$；然后将坐标向量$p_v^{xy}, p_v^z$代入方程(3)和(4)，得到坐标向量$p^{xy}$和$p^z$；

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

$$E^{xy}p^{xy} = 0 + v_5$$

此时在设定坐标$p_1^{xy}, p_{1x}^{xy}, p_{1y}^{xy}, p_{1z}^{xy}, p_1^z, p_{1x}^z, p_{1y}^z, p_{1z}^z$后，需要联立方程(1),(3),(5)计算节点向量p_v^{xy}, p^{xy}；我们设坐标向量$p_{acc}^{xy} = \left[{p_v^{xy}}^T, {p^{xy}}^T\right]^T \in C^{4\times n}$作为坐标向量$p_v^{xy}$和$p_{xy}$的合并向量；然后将方程(1),(3),(5)合并为如下；

$$F^{xy}p_{acc}^{xy} = G^{xy} + v_6$$

其中误差向量$v_6 = [v_1^T, v_3^T, v_5^T]^T, v_6 \sim N(0, \Sigma_6)$；我们忽略各个线性方程之间的联系，那么协方差矩阵$\Sigma_6$是对角矩阵；并且满足如下关系，$\Sigma_6 = \mathrm{diag}\{\Sigma_1, \Sigma_3, \text{\Simga \_5 }\}, \Sigma_3 = \text{0\_(|v3|} \times \text{|v3|)}$；

将平面坐标$p_1^{xy}, p_{1x}^{xy}, p_{1y}^{xy}, p_{1z}^{xy}$代入上述方程，该方程变成如下形式；

$$F_{sub}^{xy}p_{sub,acc}^{xy}=G_{sub}^{xy}+v_{sub6}$$

其中坐标向量$p_{sub,acc}^{xy}\in C^{4n-4}$是坐标向量$p_{acc}^{xy}$删除$p_1^{xy},p_{1x}^{xy},p_{1y}^{xy},p_{1z}^{xy}$后的向量；误差向量$v_{sub6}\sim N(0,\Sigma_{sub6})$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$v_{sub6}=v_6$，方差矩阵$\Sigma_{sub6}=\Sigma_6$；该线性方程组的最小二乘解如下；

$$p_{sub,acc}^{xy}=\left({F_{sub}^{xy}}^T\Sigma_6^{-1}F_{sub}^{xy}\right)^{-1}{F_{sub}^{xy}}^T\Sigma_6^{-1}G_{sub}^{xy}$$

计算地图尺度和节点坐标的伪代码如下；

过程 6，计算地图尺度和节点坐标（使用 GPS 数据）

输入，线性方程组$A^{xy}p_v^{xy}=b^{xy},A^zp_v^z=b^z,C^{xy}p^{xy}=D^{xy}p_v^{xy},C^zp^z=D^zp_v^z$，方差矩阵$\Sigma_1,\Sigma_2$

输出，虚拟节点坐标向量$p_v\in R^{3\times3n}$，节点坐标向量$p\in R^{3\times n}$，地图尺度ρ

设节点$i\in V$的坐标向量为$p=[p_1,p_2,\dots,p_n]\in R^{3\times n}$；

设节点$i\in V$的平面坐标向量为$p^{xy}=\left[p_1^{xy},p_2^{xy},\dots,p_n^{xy}\right]\in C^n$；

设节点$i\in V$的垂直坐标向量为$p^z=[p_1^z,p_2^z,\dots,p_n^z]^T\in R^n$；

设虚拟节点坐标向量为$p_v=\left[p_{1x},p_{1y},p_{1z},\dots,p_{nx},p_{ny},p_{nz}\right]\in R^{3\times n}$；

设虚拟节点的平面坐标向量为$p_v^{xy}=\left[p_{1x}^{xy},p_{1y}^{xy},p_{1z}^{xy},\dots,p_{nx}^{xy},p_{ny}^{xy},p_{nz}^{xy}\right]\in C^n$；

设虚拟节点的垂直坐标向量为$p_v^z=\left[p_{1x}^z,p_{1y}^z,p_{1z}^z,\dots,p_{nx}^z,p_{ny}^z,p_{nz}^z\right]^T\in R^n$；

设设节点1的平面坐标为$p_1^{xy}=0+i\times0$，虚拟节点1_x的平面坐标是$p_{1x}^{xy}=\rho+i\times0$，虚拟节点$1_y$的平面坐标是$p_{1y}^{xy}=0+i\times\rho$，虚拟节点$1_z$的平面坐标是$p_{1z}^{xy}=0+i\times0$；

设虚拟节点$i_x,i_y,i_z\in V_v/(1_x,1_y,1_z)$的平面坐标向量是$p_{sub}^{xy}=\left[p_{2x}^{xy},p_{2y}^{xy},p_{2z}^{xy},\dots,p_{nx}^{xy},p_{ny}^{xy},p_{nz}^{xy}\right]\in C^{3n-3}$；

将坐标$p_{1x}^{xy},p_{1y}^{xy},p_{1z}^{xy}$ 的平面坐标代入方程，线性方程组$A^{xy}p_v^{xy}=b^{xy}$变为$A_{sub}^{xy}p_{sub}^{xy}=b_{sub}^{xy}$；

设坐标向量$p_{acc}^{xy}=\left[{p_v^{xy}}^T,{p^{xy}}^T\right]^T\in C^{4\times n}$；将方程(1),(3),(5)合并为$F^{xy}p_{acc}^{xy}=G^{xy}$；

设坐标向量$p_{sub,acc}^{xy}\in C^{4n-4}$是坐标向量$p_{acc}^{xy}$删除$p_1^{xy},p_{1x}^{xy},p_{1y}^{xy},p_{1z}^{xy}$后的向量；方程变为$F_{sub}^{xy}p_{sub,acc}^{xy}=G_{sub}^{xy}$；

使用线性方程组$F_{sub}^{xy}p_{sub,acc}^{xy}=G_{sub}^{xy}$计算节点的平面分量$p_{sub,acc}^{xy}$；该线性方程组的最小二乘解，$p_{sub,acc}^{xy}=\left({F_{sub}^{xy}}^T\Sigma_6^{-1}F_{sub}^{xy}\right)^{-1}{F_{sub}^{xy}}^T\Sigma_6^{-1}G_{sub}^{xy}$（$\rho$的一次函数）；

设虚拟节点$i_x,i_y,i_z\in V_v/(1_x,1_y,1_z)$的垂直坐标向量是$p_{sub}^z=\left[p_{2x}^z,p_{2y}^z,p_{2z}^z,\dots,p_{nx}^z,p_{ny}^z,p_{nz}^z\right]^T\in R^{3n-3}$；

将坐标$p_{1x}^z,p_{1y}^z,p_{1z}^z$的垂直坐标代入方程，线性方程组$A^zp_v^z=b^z$变为$A_{sub}^zp_{sub}^z=b_{sub}^z$；

使用线性方程组$A_{sub}^zp_{sub}^z=b_{sub}^z$计算虚拟节点坐标的垂直分量$p_{sub}^z$；该线性方程组的最小二乘解，$p_{sub}^z=\left({A_{sub}^z}^T\Sigma_2^{-1}A_{sub}^z\right)^{-1}{A_{sub}^z}^T\Sigma_2^{-1}b_{sub}^z$（固定数值）；

使用线性方程组$C^{xy}p^{xy}=D^{xy}p_v^{xy}$计算节点$i\in V$的平面分量$p^{xy}$；该线性方程组的最小二乘解，$p^{xy}=\left({C^{xy}}^TC^{xy}\right)^{-1}{C^{xy}}^TD^{xy}p_v^{xy}$（$\rho$的一次函数）；

使用线性方程组$C^zp^z=D^zp_v^z$计算节点$i\in V$的垂直分量p^z；该线性方程组的最小二乘解，$p^z=\left({C^z}^TC^z\right)^{-1}D^zp_v^z$（固定数值）；

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

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

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p_v^{xy},p^{xy}，计算坐标p^v,p；

---

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$e(T_{ij})=h(T_{ij})^2$，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；

$\rho(x)=x^2,\text{if }-1<x<1$

$x,\text{if }x\geq1$

$-x,\text{if }x\leq-1$

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

$$J = \sum\nolimits_{T_{ij} \in E_T} \rho\left(h(T_{ij})^2\right)$$

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$e(T_{ij}) = h(T_{ij})^2$是非线性函数，增加鲁棒函数后的误差项$\rho\left(h(T_{ij})^2\right)$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；

$p_{ij} = 0.1, \text{if } \sigma_{ij}^2 < \sigma_{thres1}^2$

$0.2, \text{if } \sigma_{thres1}^2 \le \sigma_{ij}^2 < \sigma_{thres2}^2$

$0.3, \text{if } \sigma_{thres2}^2 \le \sigma_{ij}^2$

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

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$p = [p_1, p_2, \dots, p_n] \in R^{2\times n}$；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$p_j = [x_j, y_j]^T$；节点 j 相对节点 i 的坐标是$p_j^r = [x_j - x_i, y_j - y_i]^T$；这些邻居节点在全局坐标系中的相对坐标向量为$P_i^n = [p_{j1}^r, \dots, p_{jn}^r] \in R^{2\times|N_i|}, j_1, \dots, j_n \in N_i$；设节点 i 到邻居节点$j \in N_i$的位姿变换是$T_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ 1 & 0 \end{bmatrix} \in R^{3\times3}, j \in N_i$，其中平移向量是$t_{ij} = [x_{ij}, y_{ij}]^T, j \in N_i$；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$Q_i^n = [t_{ij1}, \dots, t_{ijn}] = [q_{ij1}, \dots, q_{ijn}] \in R^{2\times|N_i|}, j_1, \dots, j_n \in N_i$；

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，$p_{ix} = [x_{ix}, y_{ix}]^T, p_{iy} = [x_{iy}, y_{iy}]^T$；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；

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

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

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$P_i = [P_i^n, P_i^v] \in R^{2\times|N_i|+2}$；将坐标向量$Q_i^n$和$Q_i^v$合并得到$Q_i = [Q_i^n, Q_i^v] \in R^{2\times|N_i|+2}$；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$P_i \in R^{2\times(|N_i|+2)}$，并且已知这组点在局部坐标系$\Sigma_i$中的位置向量$Q_i \in$

$R^{2\times(|N_i|+2)}$；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$R_i\in R^{2\times 2}$；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$E(R_i)$取得最小值的旋转矩阵如下；

$$H=P_iQ_i^T$$
$$=U\Sigma V^T$$
$$R_i=VU^T$$

对矩阵$H\in R^{2\times 2}$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 7，求解局部坐标系

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}

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

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中的相对坐标向量是，$P_i^n=\left[p_{j1}^r,\dots,p_{jn}^r\right]\in R^{2\times|N_i|},j_1,\dots,j_n\in N_i$；

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

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

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，$Q_i^v=\left[p_{ix}^i,p_{iy}^i\right]\in R^{2\times 2}$；

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

使用点集配准计算旋转矩阵R_i，矩阵$H=P_iQ_i^T$；

对矩阵H奇异值分解，$H=U\Sigma V^T$；

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

局部坐标系Σ_i相对全局坐标系Σ_g的旋转矩阵是$(R_i)^{-1}$；

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】