基于 RGB-D 视觉的 SLAM 运动位姿估计

王广福[1,2]，盛选禹[1,3*]，刘沛宇[1,3]

[1]季华实验室　广东佛山
[2]中国煤炭科学研究院矿山人工智能研究院　北京
[3]清华大学机械工程系　北京

【摘要】针对三维重建系统使用 ORB 进行特征的提取并应用快速最近邻进行特征匹配，以及应用 Shi-Tomasi 角点检测的算法进行特征选择，从而实现位姿估计的数据关联，在此基础上应用 ICP 算法实现了系统的位姿初步计算。当匹配数对较少，系统基于点到面误差最小化的原则进行位姿计算，并在跟踪失败时进行重定位。这样做的优点是，系统不但可以应对特征丰富的场合，同时也可以处理特征数量较少的场景，提高了系统的鲁棒性。在位姿估计中还应用了单指令多数据流及多线程的支持，提高了运行的实时性能。位姿的初步计算完成后利用捆束优化的方法进行位姿的优化，并通过回环检测降低长时间运行的累积误差，提高了位姿估计的精度。

【关键词】RGB-D；SLAM；特征提取；地图构建；位姿估计
【基金项目】季华实验室科研项目（X220011TN220）
【收稿日期】2025 年 8 月 15 日　　【出刊日期】2025 年 9 月 18 日　【DOI】10.12208/j.aics.20250055

SLAM motion pose estimation based on RGB-D vision

Guangfu Wang[1,2], Xuanyu Sheng[1,3], Peiyu Liu[1,3]*

[1]Ji Hua Laboratory, Foshan, Guangdong
[2]Research Institute of Mine Artiffcial Intelligence, Chinese Institute of Coal Science, Beijing
[3]Department of Mechanical Engineering, Tsinghua University, Beijing

【**Abstract**】For the 3D reconstruction system, ORB is used for feature extraction, fast nearest neighbor is used for feature matching, and Shi-Tomasi corner detection algorithm is used for feature selection, so as to realize the data association of pose estimation. On this basis, the ICP algorithm is used to achieve Preliminary calculation of the pose of the system. When the matching logarithm is small, the system calculates the pose based on the principle of minimizing the point-to-surface error, and relocates when the tracking fails. The advantage of this is that the system can not only deal with situations with rich features, but also can handle scenes with a small number of features, which improves the robustness of the system. In the pose estimation, the support of single instruction, multiple data streams and multiple threads is also applied, which improves the real-time performance of the operation. After the preliminary calculation of the pose is completed, the bundle optimization method is used to optimize the pose, and the accumulated error of long-time running is reduced through loop detection, and the accuracy of the pose estimation is improved.

【**Keywords**】RGB-D; SLAM; Feature extraction; Map construction; Pose estimation

前言

RGB-D 是可同时获取彩色图与深度图的相机，也称深度相机。RGBD = RGB +Depth Map。SLAM 技术全称为 Simultaneous localization and mapping，通常翻译为同步定位与建图[1]。无人驾驶汽车及智能机器人的快速发展使越来越多的研究人员关注 SLAM 技术，并对其进行了深入的研究。随着研究的深入，SLAM 在机器人平台、自动驾驶汽车、无人机上的应用均得到了快速发展，这也推动了各种复杂功能如自动避障、地图构建等在这些平台上的实现。

由于应用视觉相机比应用激光雷达在价格上要具有明显的优势，因此目前研究的重点集中于利用相机

*通讯作者：盛选禹（1969-）男，博士，副研究员，研究方向：反应堆设计与仿真

数据来实现 SLAM 的同步定位与建图功能。自 SLAM 被提出三十多年以来[2]，定位与建图方面已经有丰富的开源算法可供调用。随着 Google 开源了 2D-SLAM 算法 Cartographer，通常认为 2D 的 SLAM 已经成熟，并且基于 2D-SLAM 的应用方面已经有不少商业化产品（如扫地机器人）。但是，应用视觉的 3D-SLAM 仍有很多方面需要深入研究。

本文基于优化方法进行位姿估计。应用图优化求解时，速度与精度均较优秀，鲁棒性也较好，适合大场景及较长时间的定位建图。就 RGB-D 视觉的位姿初步估计、长时间运行的位姿优化、回环时的检测与处理以及初步估计失败时的重定位算法进行展开，并重点说明这些算法的原理与流程。在图优化的理论与应用中位姿的初步估计通常被称为前端，长时间的位姿估计优化通常被称为后端，前端与后端运行在不同的线程上。整个位姿估计的结构如图 1 所示。

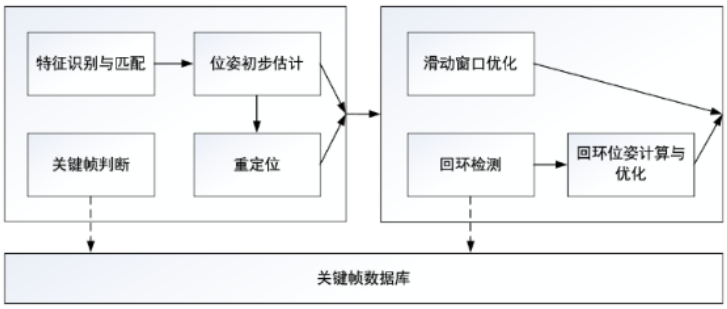


图 1　位姿估计的结构

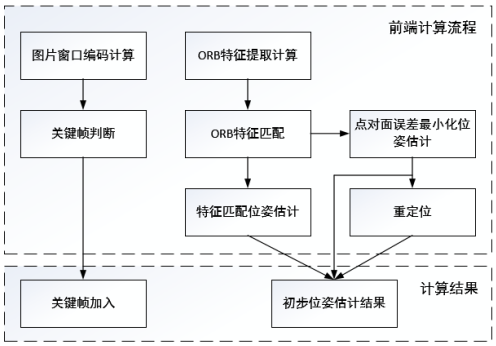


图 2　前端工作流程

1 前端运动位姿估计

前端首先应用 ORB 特征[3]进行特征提取与识别，然后将识别到的 ORB 特征进行筛选并匹配，计算位姿的预估计，获得预估计之后，利用单帧的数据点以及预估计的李代数位姿进行局部的 BA 优化获得位姿的初步估计值，这也是前端部分向后端的输出，如图 2 所示。

ORB 特征综合了 FAST 角点[4]与 BRIEF 特征描述，包括两部分：1）关键点，2）描述子。其中，关键点指的是角点在图片当中的位置，也即像素点坐标，描述子通常是一个向量，表示该角点的方向与大小等信息，通常用来比较两个特征是否相同或相近。ORB 中应用了 FAST 算法，其原理为检测某一个像素点 p 半径为 3 的圆周上分布的 16 个像素点，并比对圆周上像素点的灰度与中心点灰度，灰度变化超过阈值时（通常为 12 个点均大于中心点的 1.2 倍或均小于 0.8 倍）判定为角点，其中 $N=12$ 时算法称为 FAST-12 算法。其中 1.2 倍或者 0.8 倍的数值为阈值，实际应用时也可以设置不同的阈值。ORB 对该角点检测算法进行了优化，策略为先检测圆上编号距离为 4 的 4 个像素点，仅在符合阈值条件的点不少于 3 个时进一步检测，否则不再检测。

ORB 特征针对上述算法进行了改进，包括：1）对角点计算 Harris 响应，取前 M 个响应值最大的角点，2）在所有层次的图像上检测角点，3）记录特征方向。

实际使用时，由于 Harris 响应函数[5]式中有关参数需要手动设置，为了避免由于手动设置参数的不同引起造成结果的不稳定，应用 Harris 角点响应的改进版 Shi-Tomasi 响应[6]选取前 M 个特征点。本文在实现时也应用了 SIMD 支持来加速计算角点与响应。

在金字塔的每一层上计算 FAST 角点，即在所有分辨率的图像上提取角点，如图 3。本文应用 4 层金字塔结构，图像金字塔的构建采用区域插值算法进行计算，缩放因子为 1.2。

应用图像金字塔的目的在于获得尺度不变性。尺度不变性有两点应用上的优势：（1）初步计算时在远处获取的图片中较为模糊，可能不认为是角点，但由近处获取时角点特征明显的区域，经尺度不变性的处理后能够识别这些区域，（2）由于图像的非凸性，优化时容易陷入局部极小导致无法进一步优化，应用尺度不变性可以有效缓解这一情况。

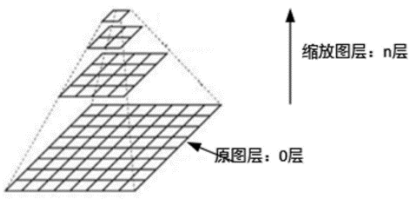


图 3　图像金字塔

计算角点的灰度方向来表示角点的方向，可表示每一个角点的旋转情况。公式为：

$$\begin{aligned} m_{00} &= \sum_{x,y\in B} I(x,y) \\ m_{10} &= \sum_{x,y\in B} xI(x,y) \\ m_{01} &= \sum_{x,y\in B} yI(x,y) \end{aligned} \tag{1}$$

$$center = (x_c, y_c) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right)$$

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right)$$

得到特征点后，需要将特征点进行矢量化描述，以便于计算图片之间特征关联。ORB 特征应用 BRIEF 描述子[7]，也即 Binary Robust Independent Elementary Feature。BRIEF 使用 256 位矢量来表示特征，并且根据计算的特征的方向进行匹配。提取图像的 ORB 特征后，根据特征之间的汉明距离来比较相似性，利用快速最近邻匹配 FLANN[8]对相邻帧的图像中特征进行匹配，建立特征匹配点对。

匹配关系建立后，由于本文中应用的视觉设备为深度相机，故而像素对应的 3D 点可以直接根据深度图数据来确定，也即求解位姿变换的问题可以等效为 3D-3D 问题。可以采用 ICP 算法或者对 3D 问题进行退化处理，解决相应的 3D-2D 问题，而 3D-2D 的问题可以使用 PnP 或者 BA 优化的方式来解决，例如 P3P、EPnP、UPnP 等算法。

由图像中的点计算三维坐标点需要相机的内参以及外参。深度相机的内参由相机本身决定，具体数据可通过相机标定来确定；而相机的外参指的就是运动位姿，需要通过前后端来估计确定。由图片像素点转换为相机坐标系的公式为：

$$\begin{cases} u = f_x \times \dfrac{x_q}{z_q} + c_x \\ v = f_y \times \dfrac{y_q}{z_q} + c_y \end{cases} \tag{2}$$

即：

$$z_q \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_q \\ y_q \\ z_q \end{pmatrix} \Leftrightarrow z_q \hat{\boldsymbol{u}} = \boldsymbol{K}\boldsymbol{q} \tag{3}$$

式中 $\boldsymbol{K}$ 表示相机内参矩阵，并定义 $\hat{\boldsymbol{u}}$ 为 $\boldsymbol{u}$ 的齐次表示，$\boldsymbol{q}$ 表示局部坐标的点。本文约定使用 $\hat{\boldsymbol{u}}$ 表示 $\boldsymbol{u}$ 矢量的齐次形式；约定使用 $\tilde{\boldsymbol{u}}$ 表示 $\boldsymbol{u}$ 的估计量。并定义函数 $\mathcal{H}(\boldsymbol{u}) = \hat{\boldsymbol{u}}$，实现将一个矢量变换为齐次形式的矢量，将齐次形式的矢量转换为非齐次定义为函数 $\mathcal{H}^{-1}(\hat{\boldsymbol{u}}) = \boldsymbol{u}$。后续其他变量也将采用这一约定。

本文首先判断匹配点对的数量及匹配的点对在深度图像中对应的深度值是否合理（由于深度相机本身的误差以及瑕疵，某些像素点可能没有对应值或者有对应值但其明显不符合深度相机拍摄深度），若匹配的有效 3D 点对数超过 50 采用 ICP 算法进行位姿估计；否则基于点对面的误差最小化进行位姿估计；如果基于点对面估计达到最大迭代轮数时未收敛则认为跟踪失败，需要重定位算法进行重新定位。

选用这样的策略的原因在于：（1）当 3D-3D 点对匹配数量较多时，使用 ICP 算法可以快速地求解，并且精度较好，（2）当匹配点对较少时，这种情况多是环境的纹理较少，导致匹配的特征点数目过少，因此直接使用 ICP 进行位姿估计结果不可靠，此时采用基于点对面的误差最小化原则进行位姿估计更加鲁棒。下述将首先介绍基于匹配点对 ICP 的位姿估计算法，然后介绍匹配点对较少时的点到面误差最小化估计方法。

匹配点对 ICP 求解 3D-3D 问题的典型场景为已知匹配的点对：

$\{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n\}$，$\{\boldsymbol{p}'_1, \boldsymbol{p}'_2, ..., \boldsymbol{p}'_n\}$，求解欧式变换使得：$\forall i \in \{1, 2, ..., n\}, \boldsymbol{p}_i = \boldsymbol{R}\boldsymbol{p}'_i + \boldsymbol{t}$，其中 $\boldsymbol{R}$，$\boldsymbol{t}$ 为待求的变量，定义误差为：

$$\boldsymbol{e}_i = \boldsymbol{p}_i - \left(\tilde{\boldsymbol{R}}\boldsymbol{p}'_i + \tilde{\boldsymbol{t}}\right) \tag{4}$$

构建最小二乘问题：

$$\min_{\boldsymbol{R}^*, \boldsymbol{t}^*} \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{e}_i\|_2^2 = \min_{\boldsymbol{R}^*, \boldsymbol{t}^*} \frac{1}{2} \sum_{i=1}^{n} \left\| \boldsymbol{p}_i - \left(\tilde{\boldsymbol{R}}\boldsymbol{p}'_i + \tilde{\boldsymbol{t}}\right) \right\|_2^2 \tag{5}$$

为求解这一问题，首先定义两点对的中心为：

$$\boldsymbol{p}_c = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{p}_i,\ \boldsymbol{p}'_c = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{p}'_i \tag{6}$$

原最小二乘问题可以变化为：

$$\min \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{e}_i\|_2^2 = \min \frac{1}{2} \sum_{i=1}^{n} \left\| \left(\boldsymbol{p}_i - \boldsymbol{p}_c - \tilde{\boldsymbol{R}}(\boldsymbol{p}'_i - \boldsymbol{p}'_c)\right) + \left(\boldsymbol{p}_c - \tilde{\boldsymbol{R}}\boldsymbol{p}'_c - \tilde{\boldsymbol{t}}\right) \right\|_2^2 \tag{7}$$

由式右侧可知，只需求解出第一项即可得到 R，再令第二项为 0 可得到 t，也即有：

$$\boldsymbol{R}^{*}=arg\min_{\boldsymbol{R}}\frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{p}_i-\boldsymbol{p}_c-\tilde{\boldsymbol{R}}(\boldsymbol{p}_i'-\boldsymbol{p}_c')\right\|_2^2 \quad (8)$$

$$\boldsymbol{t}^{*}=\boldsymbol{p}_c-\boldsymbol{R}^{*}\boldsymbol{p}_c'$$

实际上求解这一问题可首先求解$\boldsymbol{R}$。对上式进行变换，有：

$$\boldsymbol{q}_i=\boldsymbol{p}_i-\boldsymbol{p}_c,\ \boldsymbol{q}_i'=\boldsymbol{p}_i'-\boldsymbol{p}_c'$$

$$\frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{q}_i-\tilde{\boldsymbol{R}}\boldsymbol{q}_i'\right\|_2^2=\frac{1}{2}\sum_{i=1}^{n}\boldsymbol{q}_i^{T}\boldsymbol{q}_i+\frac{1}{2}\sum_{i=1}^{n}\boldsymbol{q}_i'^{T}\tilde{\boldsymbol{R}}^{T}\tilde{\boldsymbol{R}}\boldsymbol{q}_i'-\sum_{i=1}^{n}\boldsymbol{q}_i^{T}\tilde{\boldsymbol{R}}\boldsymbol{q}_i' \quad (9)$$

旋转运动矩阵为单位正交矩阵，即：$\tilde{\boldsymbol{R}}^{-1}=\tilde{\boldsymbol{R}}^{\boldsymbol{T}}$，通常称为特殊正交群，即：$\tilde{\boldsymbol{R}}\in SO(3)$，进而有$\tilde{\boldsymbol{R}}^{\boldsymbol{T}}\tilde{\boldsymbol{R}}=\boldsymbol{I}$。则式（9）中前两项中均与$\tilde{\boldsymbol{R}}$无关，因此求解最小化等同于最大化最后一项，由文献[8]可知，最优解$\tilde{\boldsymbol{R}}$可通过 SVD 分解来计算，即：

$$\begin{aligned}\boldsymbol{S}&=\sum_{i=1}^{n}\boldsymbol{q}_i\boldsymbol{q}_i'^{T}\\ \boldsymbol{S}&=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T}\\ \tilde{\boldsymbol{R}}&=\boldsymbol{U}\boldsymbol{V}^{T}\end{aligned} \quad (10)$$

注意此时若$\det\left(\tilde{\boldsymbol{R}}\right)<0$，取$\tilde{\boldsymbol{R}}=-\tilde{\boldsymbol{R}}$并相应地计算$\boldsymbol{t}$。

如前所述，如果匹配点对小于 50 对，则认为匹配点对较少，此时不再应用基于匹配点对的 ICP 算法，而是应用基于点对面的误差最小化的求解算法。基于点对面的误差最小化求解算法不再对点对进行匹配，而是计算每一像素对应的局部坐标系三维点，每次迭代时均重新计算一点对应的帧中的像素点。

首先根据匹配点对计算包围盒的大小，即匹配点对在前一帧图像中的像素坐标最靠左上与最靠右下的坐标，将包围盒包围的区域称为共视区域；进而计算共视区域的像素点对应的三维坐标以及法矢量，构造误差项并最小化总误差求解位姿变换，为降低图片非凸性带来的影响，应用图片金子塔结构进行计算，这里重复使用 FAST 特征计算阶段时的图片金字塔，并仅选取 2 层金字塔结构，包括原始图片与上 1 层位置图片。

局部坐标系中的点及法向矢量变换到全局坐标系的公式为：

$$\boldsymbol{v}_g=\boldsymbol{T}_{g,k}\hat{\boldsymbol{q}}$$

$$\boldsymbol{N}_k=(\boldsymbol{q}(u+1,v)-\boldsymbol{q}(u,v))\times(\boldsymbol{q}(u,v+1)-\boldsymbol{q}(u,v))$$

$$\boldsymbol{N}=\frac{1}{\|\boldsymbol{N}_k\|}\boldsymbol{R}_{g,k}\boldsymbol{N}_k \quad (11)$$

式中$\boldsymbol{v}_g$表示在全局坐标中的三维坐标，$\boldsymbol{T}_{g,k}$为从第 k 帧向全局坐标进行变换的位姿矩阵的李群，$\hat{\boldsymbol{q}}$为齐次的局部坐标点，$\boldsymbol{N}_k$为第 k 帧图片数据点对应像素位置的法向矢量，$\boldsymbol{N}_g$则为全局坐标系下的法向矢量，$\boldsymbol{R}_{g,k}$为位姿变换的旋转矩阵，为$\boldsymbol{T}_{g,k}$的前三行与前三列。

定义误差项为：

$$e(\boldsymbol{u})=\left(\boldsymbol{T}_{g,k}\hat{\boldsymbol{v}}_k(\boldsymbol{u})-\hat{\boldsymbol{v}}_{k-1}^{g}(\tilde{\boldsymbol{u}})\right)^{T}\cdot\hat{\boldsymbol{N}}_{k-1}^{g}(\tilde{\boldsymbol{u}})$$

此误差代表的实际意义如图 4 所示。求解最小化误差项可以表示为：

$$\min_{\boldsymbol{u}\in\Omega_k}\|e(\boldsymbol{u})\|_2^2=\min_{\boldsymbol{u}\in\Omega_k}\left\|\left(\boldsymbol{T}_{g,k}\hat{\boldsymbol{v}}_k(\boldsymbol{u})-\hat{\boldsymbol{v}}_{k-1}^{g}(\tilde{\boldsymbol{u}})\right)^{T}\cdot\hat{\boldsymbol{N}}_{k-1}^{g}(\tilde{\boldsymbol{u}})\right\|_2^2 \quad (12)$$

其中$\tilde{\boldsymbol{u}}$为上次迭代中与$\boldsymbol{u}$对应的像素点，即：

$$\begin{aligned}\tilde{\boldsymbol{u}}&=\pi\left(\boldsymbol{K}\left(\boldsymbol{T}_{k-1,k}^{z-1}\boldsymbol{v}_k(\boldsymbol{u})\right)\right)\\ \pi\left(\boldsymbol{q}(x,y,x)^{T}\right)&=\left(\frac{x}{z},\frac{y}{z}\right)^{T}\end{aligned} \quad (13)$$

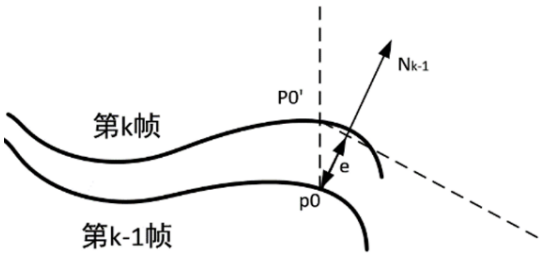


图 4　误差项实际意义

位姿矩阵共有 16 项内容，但独立的自由度共计有 6 个，包含三个平移分量与三个欧拉角，因此可以用矢量$\boldsymbol{x}_T=(\alpha,\beta,\gamma,t_x,t_y,t_z)^{T}$表示位姿增量矩阵，并考虑每次迭代时位移分量与欧拉角分量均比较小，初始位姿估计使用上一帧位姿估计的结果，记为$\boldsymbol{T}_{g,k}^{0}$，则每次迭代更新可以表示为：

$$\boldsymbol{T}_{inc}^{z}=\begin{pmatrix}1&\gamma&-\beta&t_x\\-\gamma&1&\alpha&t_y\\\beta&-\alpha&1&t_z\\0&0&0&1\end{pmatrix}=\begin{pmatrix}0&\gamma&-\beta&t_x\\-\gamma&0&\alpha&t_y\\\beta&-\alpha&0&t_z\\0&0&0&0\end{pmatrix}+\mathbf{I}$$

$$\boldsymbol{T}_{g,k}^{z}=\boldsymbol{T}_{inc}^{z}\boldsymbol{T}_{g,k}^{z-1}\triangleq\Delta\boldsymbol{T}^{z}\boxplus\boldsymbol{T}_{g,k}^{z-1} \quad (14)$$

位姿估计尚未完成时，实际上$\boldsymbol{T}_{g,k}$是未知变量，定义$z-1$轮迭代后的位姿估计为$\tilde{\boldsymbol{T}}_{g,k}^{z-1}$，并定义该轮迭代后的点：$\tilde{\boldsymbol{v}}_k^{g}(\boldsymbol{u})=\tilde{\boldsymbol{T}}_{g,k}^{z-1}\hat{\boldsymbol{v}}_k(\boldsymbol{u})$，将$\boldsymbol{T}_{g,k}\hat{\boldsymbol{v}}_k(\boldsymbol{u})$变换为：

$$\boldsymbol{T}_{g,k}\hat{\boldsymbol{v}}_{\boldsymbol{k}}(\boldsymbol{u})=\boldsymbol{T}_{inc}^{z}\tilde{\boldsymbol{v}}_{k}^{g}(\boldsymbol{u})=\boldsymbol{G}\boldsymbol{x}_{T}+\tilde{\boldsymbol{v}}_{k}^{g}(\boldsymbol{u}) \quad (15)$$

且当$\tilde{\boldsymbol{v}}_{k}^{g}(\boldsymbol{u})=(\tilde{x}_u,\tilde{y}_u,\tilde{z}_u)$时，$\boldsymbol{G}$的形式为：

$$\boldsymbol{G}=\begin{pmatrix}0 & -\tilde{z}_u & \tilde{y}_u & 1 & 0 & 0\\ \tilde{z}_u & 0 & -\tilde{x}_u & 0 & 1 & 0\\ -\tilde{y}_u & \tilde{x}_u & 0 & 0 & 0 & 1\end{pmatrix} \quad (16)$$

则误差项可以表示为：

$$\begin{aligned}E^2&=(\boldsymbol{N}_{k-1}^{g\,T}\boldsymbol{G}\boldsymbol{x}_T+m)^T(\boldsymbol{N}_{k-1}^{g\,T}\boldsymbol{G}\boldsymbol{x}_T+m)\\&=\boldsymbol{x}_T^T\boldsymbol{G}^T\boldsymbol{N}_{k-1}^{g}\boldsymbol{N}_{k-1}^{g\,T}\boldsymbol{G}\boldsymbol{x}_T+2m\boldsymbol{N}_{k-1}^{g\,T}\boldsymbol{G}\boldsymbol{x}_T+m^2\\&=\boldsymbol{x}_T^T\boldsymbol{h}\boldsymbol{h}^T\boldsymbol{x}_T+2m\boldsymbol{h}^T\boldsymbol{x}_T+m^2\\m&=\boldsymbol{N}_{k-1}^{g\,T}(\tilde{\boldsymbol{v}}_{k}^{g}(\boldsymbol{u})-\hat{\boldsymbol{v}}_{k-1}^{g}(\tilde{\boldsymbol{u}}))\\\boldsymbol{h}^T&=\boldsymbol{N}_{k-1}^{g\,T}\boldsymbol{G}\end{aligned} \quad (17)$$

误差项关于$\boldsymbol{x}_T$的偏导数为：

$$\frac{\partial E^2}{\partial \boldsymbol{x}_T}=(\boldsymbol{h}^T\boldsymbol{h}+(\boldsymbol{h}^T\boldsymbol{h})^T)\boldsymbol{x}_T+2m\boldsymbol{h} \quad (18)$$

令该导数为 0 并整合所有像素点有：

$$\left(\sum_{\boldsymbol{u}\in\Omega_k}\boldsymbol{h}\boldsymbol{h}^T\right)\boldsymbol{x}_T=-\sum_{\boldsymbol{u}\in\Omega_k}m\boldsymbol{h} \quad (19)$$

应用 Cholesky 分解求解此线性方程组可得到位姿递增矢量$\boldsymbol{x}_T$，进而可估计位姿变换。值得注意的是，在迭代时首先在低分辨率的图片上进行迭代，迭代一轮后换第二层迭代然后在原始图像上进行迭代，即由粗到细的方式进行计算。由于$\boldsymbol{T}_{inc}^{z}$的表示实际上假设了小的转动，即应用了θ接近 0 时$\sin\theta\approx\theta,\cos\theta\approx1$这一计算得来，对于迭代中的转角过大时不再成立。迭代次数总数为 6，且每一层均有 3 次迭代。每一层迭代的顺序为从低分辨率图像到高分辨率图像。

前端误差通常可达分米量级，如图 5 所示，且随着时间增加误差有增加的趋势。前端位姿估计的误差将会影响后端优化的迭代速度。前端的主要作用是为后端优化提供迭代初值，因而前端位姿估计的结果将直接影响到优化的速度，若前端误差过大，将有可能导致后端也难以对位姿进行有效的优化计算，从而使得建图出现扭曲、割裂等现象。

前端除了计算相邻两帧图片数据的相对位姿变换并转换为全局位姿变换之外，还计算每帧图片的二进制表示。本文应用随机窗口编码法[9]计算每一帧的编码，并根据编码值来判断是否把当前帧加入关键帧序列。

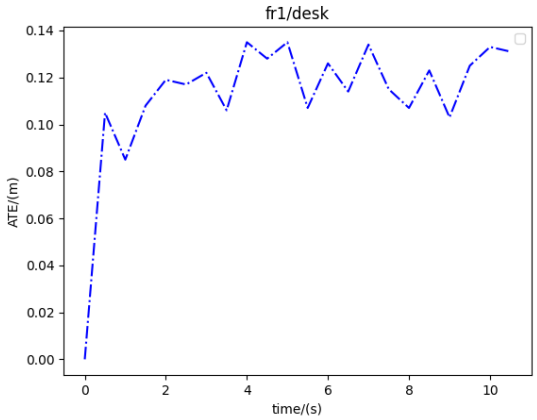


图 5　前端估计的绝对误差（fr1/desk 数据集）

关键帧包含的主要数据有：（1）时间戳，（2）该关键帧位姿与位姿的逆，（3）关键帧的 ORB 特征。时间戳为系统获得该帧的时间；该关键帧位姿及其逆为求得的优化位姿；ORB 特征序列主要是在回环检测阶段使用，当检测到可能有回环时，使用 ORB 特征进行特征的匹配并计算相对的位姿变换，进而可以由此构建位姿图进行回环的位姿优化。

窗口编码的算法流程为：随机选择窗口内一个像素点并计算窗口的等效灰度值，该等效灰度值与阈值的大小对比，若等效灰度值较小则二进制值设为 0，反之则反。一个二进制值表示一个窗口，即一个 bit，然后计算在不同图像金字塔层中的所有窗口的二进制表示，每一个窗口实际为包含有2×2个像素点的方形窗口。这样构成一个二进制序列，进而比较当前帧数据的二进制序列与上一关键帧的二进制序列的相似度，相似度可用汉明距离衡量，汉明距离越大则相似度越小。相似度低于规定的阈值时，则认为此帧为关键帧。邻近帧序列主要用于重定位算法及优化环节，如图 6 所示。

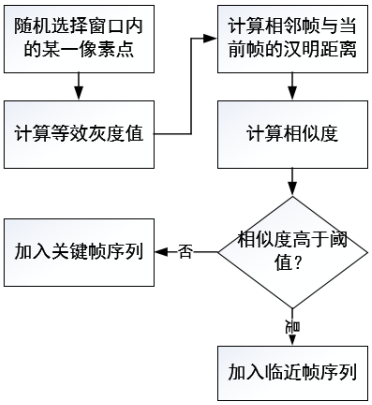


图 6　窗口编码流程

对于像素点$\boldsymbol{u}\in\Omega\in\mathbb{R}^2$，其等效灰度值表示为$I_e(\boldsymbol{u})\in\mathbb{R}$，其对应像素点的通道$c\in\{R,G,B,D\}\in\mathbb{R}^4$，对于每一个像素点而言，其相应的函数为：

$$f(I_e,\tau)=\begin{cases}1 & I_e\geq\tau\\ 0 & I_e<\tau\end{cases} \tag{20}$$

等效灰度值的计算公式为：

$$I_e=\frac{R+G+B}{3} \tag{21}$$

对于一个窗口而言，一个窗口可以表示为：$F=random(f_i),0\leq i<4$，每帧图像的编码可以表示为：$C=\{F_i\}_{i=1}^{n}$。对于第一层的图像金字塔而言，其尺寸为533×400，对于这一层的图像而言，共有$n=floor(533/2)\times400/2=5.32\times10^4$个窗口。因此该层图像数据的编码长度为$5.32\times10^4$ bit，约 6.5KB。

对于不同的帧之间相似度的计算，定义I帧的图片与J帧的图片相似函数$\xi(I,J)$（符号$\oplus$表示异或操作）：

$$\begin{aligned}BlockHD_{i,j}&=\sum_{k=1}\left(b_{F_k}^{i}\oplus b_{F_k}^{j}\right)\\ \xi_{i,j}&=1-BlockHD_{i,j}/n\end{aligned} \tag{22}$$

系统运行时，将当前帧I与关键帧序列中最近关键帧及临近帧序列进行比对，计算相似度函数，定义总体相似度函数ξ_I：

$$\xi_i=\min_{\forall j}(\xi_{i,j}) \tag{23}$$

总体相似度小于 0.7 时，判定该帧为关键帧并将其插入关键帧队列中。

2 后端位姿优化

后端优化中常用的方法是 BA，利用多帧数据以及前端对这些数据估计的位姿，进行整体的优化。基于 BA 的后端优化的思想是将所有位姿与观测点均当成待优化的变量，并最小化总误差。如果认为位置观测到的数据点发射出光线，那么该数据点会在相应的几个相机的成像平面上变成相应的像素点，对于特征明显的环境数据点，则会形成成像平面上的特征点。当优化各个相机的位姿以及环境中数据点的实际位置时，使得这些光束可以收聚到相机的光心处，此时认为对于各个相机的位姿估计是正确的，这一算法实现过程称为 BA[10]。实际就是将待优化的变量捆束起来（Bundle），然后每次迭代优化时进行调整（Adjustment）。

后端优化中首先需要确定优化的帧。本文结合前端对于关键帧的判断来决定优化时的帧，如果当前帧是关键帧并且上轮优化已完成，则优化临近队列到当前帧的所有数据；若当前帧为关键帧但上轮优化过程未完成则等待；若当前帧不是关键帧则等待。值得注意的是，本文在优化前判断上一关键帧到当前帧的等效位移来判断系统是否处于准静止的状态，避免在机器人处于静止状态时系统进行无谓的优化。等效位移的计算公式为：

$$s_e=\sum_i\|t_i\|_2^2+w_e\sum_i(\alpha_i^2+\beta_i^2+\gamma_i^2) \tag{24}$$

其中t_i为上一关键帧到当前帧的所有帧中连续的两帧之间的相对平移矢量，α，β，γ为相对转动的欧拉角，w_e为权重系数，并取$w_e=15$。设阈值为 0.0001，即当$s_e\geq0.0001$时，进行 BA 优化，否则认为此时机器人已经静止不动，不再进行进一步优化。优化的帧如图 7 所示，BA 优化的流程如图 8 所示。

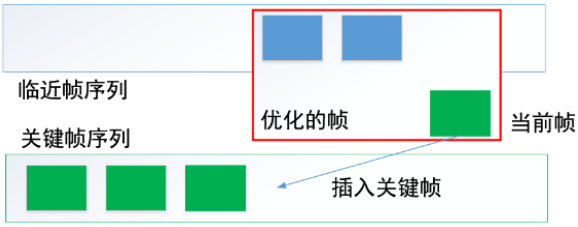


图 7　BA 优化时选择的帧

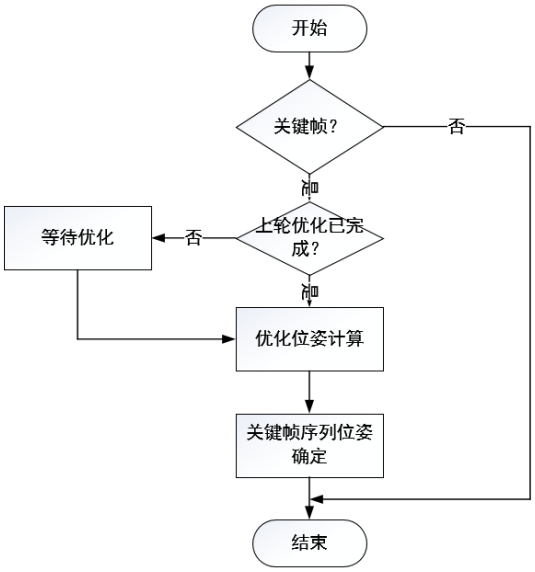


图 8　BA 流程

进行 BA 优化时，计算优化帧中的所有像素点在不同帧中的对应点，记函数$h(\boldsymbol{T}_i,\boldsymbol{p}_j)$表示某三维点$\boldsymbol{p}_j$经相对位姿变换$\boldsymbol{T}_i$变换之后的在第$i$帧中的像素点（本文使用已经校准的深度相机，认为相机的内参矩阵已知）。记$\boldsymbol{z}_{ij}$为$\boldsymbol{p}_j$三维点实际应当对应的在第i帧中的像素点，并记误差项：$\boldsymbol{e}=\boldsymbol{z}_{ij}-h(\boldsymbol{T}_i,\boldsymbol{p}_j)$，表示由于观测以及位姿估计的不准确导致的误差。误差项定义中的变量既包含相对位姿变换$\boldsymbol{T}_i$，也包含实际的三维坐标点。观测方程$h(\boldsymbol{T}_i,\boldsymbol{p}_j)$的具体形式为：

$$\begin{aligned} h(\boldsymbol{T}_i,\boldsymbol{p}_j)&=\mathcal{H}^{-1}\left(\frac{1}{z_q}\boldsymbol{K}\boldsymbol{q}_{ij}\right)\\ \boldsymbol{q}_{ij}&=\mathcal{H}^{-1}\left(\boldsymbol{T}_i^{-1}\hat{\boldsymbol{p}}_j\right)\end{aligned} \tag{25}$$

将所有的误差项汇总起来，得到最小化误差的目标方程：

$$\min_{i,j}\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{e}\|_2^2=\min_{i,j}\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{z}_{ij}-h(\boldsymbol{T}_i,\boldsymbol{p}_j)\|_2^2 \tag{26}$$

这样的最小二乘问题的求解通常需要确定误差项关于待优化变量的雅可比矩阵。由式 25 可知，该方程关于位姿矩阵以及三维坐标点均为非线性，因此可以采用基于非线性优化的策略来优化求解。求解时首先设置变量的初值，对于位姿矩阵而言，使用前端估计的结果作为初值并利用深度相机数据以及相机内参数计算三维坐标点初值，进而计算误差项关于变量的雅可比矩阵，并迭代最小化误差以求解更新量。

定义位姿与三维点的变量为：

$$\begin{aligned}\boldsymbol{x}&=(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2,...,\boldsymbol{\zeta}_m,\boldsymbol{p}_1,\boldsymbol{p}_2,...,\boldsymbol{p}_n)^T\in\mathbb{R}^{6m+3n}\\ \boldsymbol{\zeta}_i&=(\alpha,\beta,\gamma,t_x,t_y,t_z)^T\in\mathbb{R}^6\end{aligned} \tag{27}$$

并且误差项关于优化变量的函数定义为：$f(\boldsymbol{x})$，则对误差函数进行一阶 Taylor 展开可以写为：

$$\begin{aligned} f(\boldsymbol{x}+\Delta\boldsymbol{x})&\approx f(\boldsymbol{x})+\boldsymbol{P}_{ij}\Delta\boldsymbol{\zeta}_i+\boldsymbol{C}_{ij}\Delta\boldsymbol{p}_j\\ \boldsymbol{P}_{ij}&=\frac{\partial f}{\partial\boldsymbol{\zeta}_i}\\ \boldsymbol{C}_{ij}&=\frac{\partial f}{\partial\boldsymbol{p}_j}\end{aligned} \tag{28}$$

定义位姿估计的变量为：$\boldsymbol{x}_\zeta=(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2,\cdots,\boldsymbol{\zeta}_m)^T$，三维路标点的变量为：$\boldsymbol{x}_p=(\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_n)^T$，则有：

$$f(\boldsymbol{x}+\Delta\boldsymbol{x})\approx f(\boldsymbol{x})+\boldsymbol{P}\Delta\boldsymbol{x}_\zeta+\boldsymbol{C}\Delta\boldsymbol{x}_p \tag{29}$$

对应的雅可比矩阵为：$\boldsymbol{J}=(\boldsymbol{P}\ \ \boldsymbol{C})$，采用高斯牛顿法或 LM 方法求解该非线性优化问题时，可以用雅可比矩阵构建$\boldsymbol{H}$矩阵，高斯牛顿法中$\boldsymbol{H}=\boldsymbol{J}^T\boldsymbol{J}$，而 LM 方法中$\boldsymbol{H}=\boldsymbol{J}^T\boldsymbol{J}+\lambda\boldsymbol{I}$，待求解的方程为$\boldsymbol{H}\Delta\boldsymbol{x}=\boldsymbol{b}$，为线性方程组求解。

由于 BA 优化问题中 H 矩阵的稀疏性，上述问题可以应用消元的方法进行快速求解。H 矩阵的稀疏性在于，雅可比矩阵分量$\boldsymbol{J}_{ij}$仅仅描述了在位姿$\boldsymbol{T}_i$处可以观测到$\boldsymbol{p}_j$路标点，而其他项均为 0，也就是：

$$\boldsymbol{J}_{ij}=\left(0_{2\times6},0_{2\times6},...,\frac{\partial e_{ij}}{\partial\boldsymbol{\zeta}_i},0_{2\times6},...,0_{2\times3},0_{2\times3},...,\frac{\partial e_{ij}}{\partial\boldsymbol{p}_j},0_{2\times3},...\right) \tag{30}$$

则对于所有的$\boldsymbol{J}_{ij}^T\boldsymbol{J}_{ij}$，上述的雅可比结构使得 H 矩阵中存在大量零元素，因而 H 矩阵是稀疏的。H 矩阵分块结构为：

$$\boldsymbol{H}=\begin{pmatrix}\boldsymbol{B}&\boldsymbol{E}\\ \boldsymbol{E}^T&\boldsymbol{C}\end{pmatrix} \tag{31}$$

其中 B，C 为对角块矩阵，B 矩阵仅和位姿有关，C 矩阵仅和路标点有关。从而通过 Schur 消元，即有：

$$\begin{aligned}\begin{pmatrix}\boldsymbol{B}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{E}^T&0\\ \boldsymbol{E}^T&\boldsymbol{C}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{x}_\zeta\\ \Delta\boldsymbol{x}_p\end{pmatrix}&=\begin{pmatrix}\boldsymbol{v}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{w}\\ \boldsymbol{w}\end{pmatrix}\\ \begin{pmatrix}\boldsymbol{B}&\boldsymbol{E}\\ \boldsymbol{E}^T&\boldsymbol{C}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{x}_\zeta\\ \Delta\boldsymbol{x}_p\end{pmatrix}&=\begin{pmatrix}\boldsymbol{v}\\ \boldsymbol{w}\end{pmatrix}\end{aligned} \tag{32}$$

通过 Schur 消元求解的步骤为：（1）求解方程$(\boldsymbol{B}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{E}^T)\Delta\boldsymbol{x}_\zeta=\boldsymbol{v}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{w}$，（2）通过方程求解$\Delta\boldsymbol{x}_p=\boldsymbol{C}^{-1}(\boldsymbol{w}-\boldsymbol{E}^T\Delta\boldsymbol{x}_\zeta)$。

实际求解时，由于路标点三维坐标由深度相机测得，不是多视图几何关系估算而来，因而迭代两轮之后认为路标点不再发生改变，也即 Schur 消元步骤 2 不再进行，仅仅计算步骤 1 中的位姿增量。

为了避免个别异常值对于整体优化解的影响，本文中使用鲁棒核函数进行求解。鲁棒核函数由很多种类型，来自谷歌研究院的 Barron 提出可使用一个公式表示通用的核函数[11]，本文中使用 Huber 核函数，即为通用核函数式的特例，可以表示为：

$$Kernel(e)=\begin{cases}\frac{1}{2}e^2 & |e|\leqslant\delta\\ \delta\left(|e|-\frac{1}{2}\delta\right) & |e|>\delta\end{cases} \tag{33}$$

Huber 核函数本身为光滑的，可以方便地求导进行计算。

计算得到位姿增量之后，由于旋转矩阵不在欧式空间，因此不能简单地将增量加在迭代初值之上，也即不能使用式$\alpha^{z+1}=\alpha^{z}+\Delta\alpha$进行计算从第$z$轮迭代到第$z+1$轮迭代的结果。参考德国 Bremen 大学的 Christoph Hertzberg 的关于流形的介绍以及关于旋转矩阵所处流形的证明[12]，应用式（14）中定义的符号⊞进行更新计算，从而对于旋转分量而言迭代式为：

$$\boldsymbol{R}^{z+1}=\Delta\boldsymbol{R}\boxplus\boldsymbol{R}^{z}=\Delta\boldsymbol{R}\cdot\boldsymbol{R}^{z}\in SO(3)$$

$$\Delta\boldsymbol{R}=\begin{pmatrix}1 & \Delta\gamma & -\Delta\beta\\ -\Delta\gamma & 1 & \Delta\alpha\\ \Delta\beta & -\Delta\alpha & 1\end{pmatrix} \tag{34}$$

位姿更新公式也可定义为：

$$\boldsymbol{T}_i^{z+1}=\Delta\boldsymbol{T}_i\boxplus\boldsymbol{T}_i^{z} \tag{35}$$

迭代优化时使用此更新公式每次更新对于位姿的估计，对于三维点而言，由于其处于欧式空间，因此⊞与＋等同，可直接计算。每次更新迭代后使用新值作为迭代的初值。

3 回环检测

当系统长时间运行时，随着位姿估计误差的累积，不可避免地使得定位越来越不准确，而对于本文以构建环境三维模型为中心任务的系统来说，位姿估计的不准确会使模型呈现出扭曲、割裂的现象。例如在重复扫描时，本应该是一个平面的物体，由于位姿估计的不准确，平面在两次扫描中不能很好地形成一个平面，而可能出现两张相交平面的情况。

本文使用基于词袋[13]进行回环检测。首先根据词袋模型计算当前帧与关键帧队列帧的相似度，若判断达到阈值则进行当前帧与该疑似回环帧临近关键帧的比对判断，当疑似回环帧左右紧邻的关键帧中有不少于 5 帧均与当前关键帧相似度较高时进行特征匹配，并且计算匹配的特征数量，对于少于 50 个特征匹配对时，认为并未检测到回环，若匹配特征对数大于 50 对，则计算相对位姿变换，并根据相对位姿变换计算位姿误差，如果位姿误差小于一定阈值κ，则判定当前帧与该疑似回环帧为回环，否则不为回环。回环检测的流程如图 9 所示。

在当前帧与疑似关键帧匹配特征对数大于 50 时仍计算相对位姿变换，是考虑对于不同环境布局相似的情况，这时回环检测的匹配特征数目较多，但是并非遇到了回环的情况。如图 10 所示，计算位姿误差可排除由于本身物体特征非常相近导致的误“回环”。

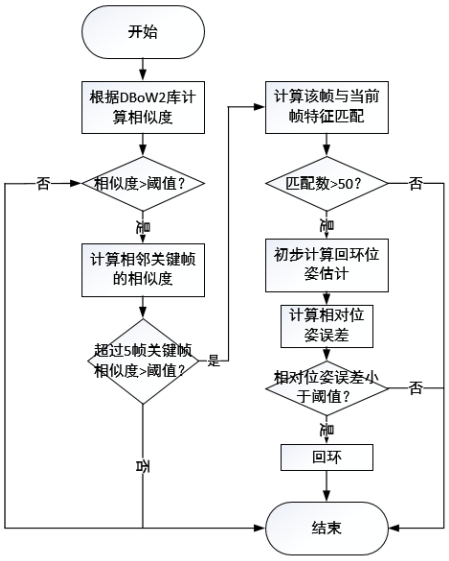
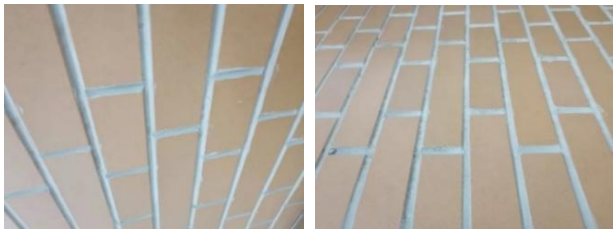


图 9　回环检测流程



（a）二楼墙壁一角　　（b）三楼墙壁一角

图 10　误匹配的回环，a 与 b 分别为不同的楼层里的墙壁图像，但却易由特征匹配认定为回环

之所以选用 50 对匹配特征主要是考虑到一旦回环检测错误地匹配，将导致后续的位姿估计出现严重错误，因此要求特征匹配的数目足够，主要是为了尽可能避免误匹配带来的后续巨大定位误差。对于相对位姿误差的阈值，本文中κ取值为 1，意味着等效最大 1m 的运动误差。这一取值的主要考虑为，根据其他一些 SLAM 系统的测试结果以及本系统的运行情况，SLAM 系统的定位估计误差的量级通常在分米级，因此选用米级的门限值来排除由于特征非常接近的但是实际为截然不同的物体的回环误匹配。

检测到回环后，相对位姿用当前图片数据的特征与回环特征进行计算。考虑到建图的时间开销，回环计算后的位姿不再进行位姿图优化，直接应用于建图中。

4 重定位运动位姿

当特征匹配对数较少且基于点对面的迭代中已到

达迭代最大轮数仍未收敛时，用重定位来确定机器人的运动位姿估计。当机器人运动速度过高，或突然的运动时，特征匹配算法往往会失败，无法匹配到足够的特征；且由于运动量较大，基于点对面的误差最小化也无法满足小运动的假设，进而需要重定位算法来进行处理。

本文重定位算法基于速度匀变速变化这一假设，也就是说，在重定位算法运行的前后一段时间内，机器人的运动速度呈线性关系，当时间间隔较小时可以粗略地认为速度线性变化。基于这一假设，仅需要两个点即可确定重定位前后一段时间内的速度曲线。根据当前帧是否为关键帧，确定这两个速度的取值。

特征匹配的算法定位失败后，如果当前帧为关键帧，则计算当前帧之前 2 个临近帧平均运动速度，包含角速度与平移运动速度；如果当前帧不是关键帧，则选择当前帧之前的两关键帧的平均运动速度。根据两个运动速度以及时间戳信息，构建速度曲线并计算当前帧的速度，进而计算当前帧的位姿变换。如图 11 所示。当获得当前帧的速度估计时，当前帧的位姿估计应用线性插值的方式计算。

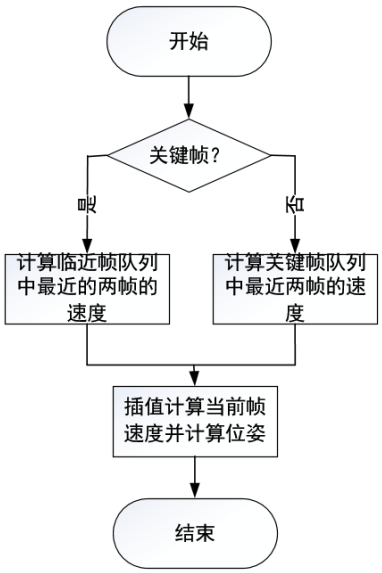


图 11　重定位流程

5 结论

本文介绍了应用特征点法的位姿估计过程，包括前端的初步位姿估计、后端的优化位姿、长时间运行时的回环检测以及位姿估计失败时的重定位算法。前端的初步位姿估计中，使用 ORB 特征计算图像特征点并用最近邻算法进行匹配。当匹配点的数目较多时用 ICP 算法进行位姿估计，纹理较少时基于点对面的误差最小化原则进行位姿估计，而跟踪失败时使用基于速度线性变化的假设进行重定位，此外前端还应用随机窗口编码算法判断当前帧是否为关键帧。前端的算法均在一个线程中进行。基于 BA 的优化方法进行位姿的优化，并且基于窗口编码来判断需要优化的帧。对于回环检测，首先根据图像的“单词”计算汉明距离粗略地判断回环，通过粗检测的帧再使用特征匹配算法来进行回环的检测，对于匹配点对数目较多的帧计算相对位姿，并据此计算位移误差，位移误差小于一定的阈值时认为回环产生，其他情况均认为未检测到回环。回环确定之后，利用特征匹配的算法计算相对位姿变换。

位姿计算的精度将直接影响后续建图环节的效果。由于建图模块依赖位姿估计模块提供较为准确的运动信息，运动位姿估计的不准确将导致建图中出现表面堆叠、撕裂扭曲等不光滑的现象，因此对后续模型的影响较大。

运动位姿的估计是进行全局一致地图的构建的前提条件，对于本文提出的以构建环境的三维模型为中心任务的系统而言，更是非常重要的一个过程，因此位姿估计中考虑了各种情况下的处理。例如在前端中，特征较少的情况中应用优化的思路进行位姿估计，较多时利用特征匹配进行估计，跟踪失败时利用短时间内速度线性变化的假设进行估计；在回环中考虑了误检测，因而使用等效位移的方法来排除由于环境中外观相似但是本身不是同一个物体的错误回环判断。

参考文献

[1] Liu H M, Zhang G F, Bao H J. A survey of monocular simultaneous localization and mapping[J].Journal of Computer-Aided Design & Computer Graphics, 2016, 6: 855-868.

[2] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5: 56-68.

[3] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// 2011 International conference on computer vision. IEEE, 2011: 2564-2571.

[4] Viswanathan D G. Features from accelerated segment test (fast)[C]// Proceedings of the 10th workshop on Image Analysis for Multimedia Interactive Services, London, UK. 2009: 6-8.

[5] Harris C G, Stephens M. A combined corner and edge detector[C]// Alvey vision conference. 1988, 15: 10-5244.

[6] Shi J, Tomasi C. Good Features to Track[J]. Proceedings CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2002, 600.

[7] Calonder M, Lepetit V, Strecha C, et al. Brief: Binary robust independent elementary features[C]// European conference on computer vision. Springer, Berlin, Heidelberg, 2010: 778-792.

[8] Pomerleau F, Colas F, Siegwart R . A Review of Point Cloud Registration Algorithms for Mobile Robotics[J]. Foundations & Trends in Robotics, 2015, 4: 1-104.

[9] Alonso I, Riazuelo L, Murillo A C. Enhancing v-slam keyframe selection with an efficient ConvNet for semantic analysis[C]// 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 4717-4723.

[10] 高翔, 张涛. 视觉 SLAM 十四讲：从理论到实践[M]. 北京: 电子工业出版社, 2019.

[11] Barron J T. A general and adaptive robust loss function[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 4331-4339.

[12] Hertzberg C. A framework for sparse, non-linear least squares problems on manifolds[C]// Universität Bremen, 2008.

[13] Galvez-Lpez D , Tardos J D . Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28: 1188-1197.