《智能机器人设计》

智能机器人SLAM

智能机器人SLAM

随着近几十年来机器人技术和产业化的快速发展，越来越多的机器人出现在人们生活的方方面面。

机械臂

安保机器人

医疗机器人

清洁机器人

智能机器人的工作环境大多是复杂的、危险的，这就意味着对机器人的智能化要求非常高，智能机器人必须具有足够强大的自主性。不难发现，在工业机器人和服务机器人领域，都需要机器人具备自主移动的能力。自主移动性能要求机器人具有在未知的环境中感知周围的环境特征、实时进行路径决策与规划及执行任务等多项任务综合的能力。

智能机器人SLAM

机器人在复杂的环境中执行任务时，机器人需要解决3个问题：我在哪，我周围的环境是什么样的，我要去哪，分别对应机器人定位，建图，自主导航与路径规划。首先需要解决建图问题，为了获得精确的地图，需要借助测量精度较高的外部传感器。SLAM技术常用于解决机器人建图与定位同步进行的问题。随着近年来理论研究与实际应用的不断发展，已经形成**基于激光的SLAM**和**基于视觉的SLAM**两大分支。

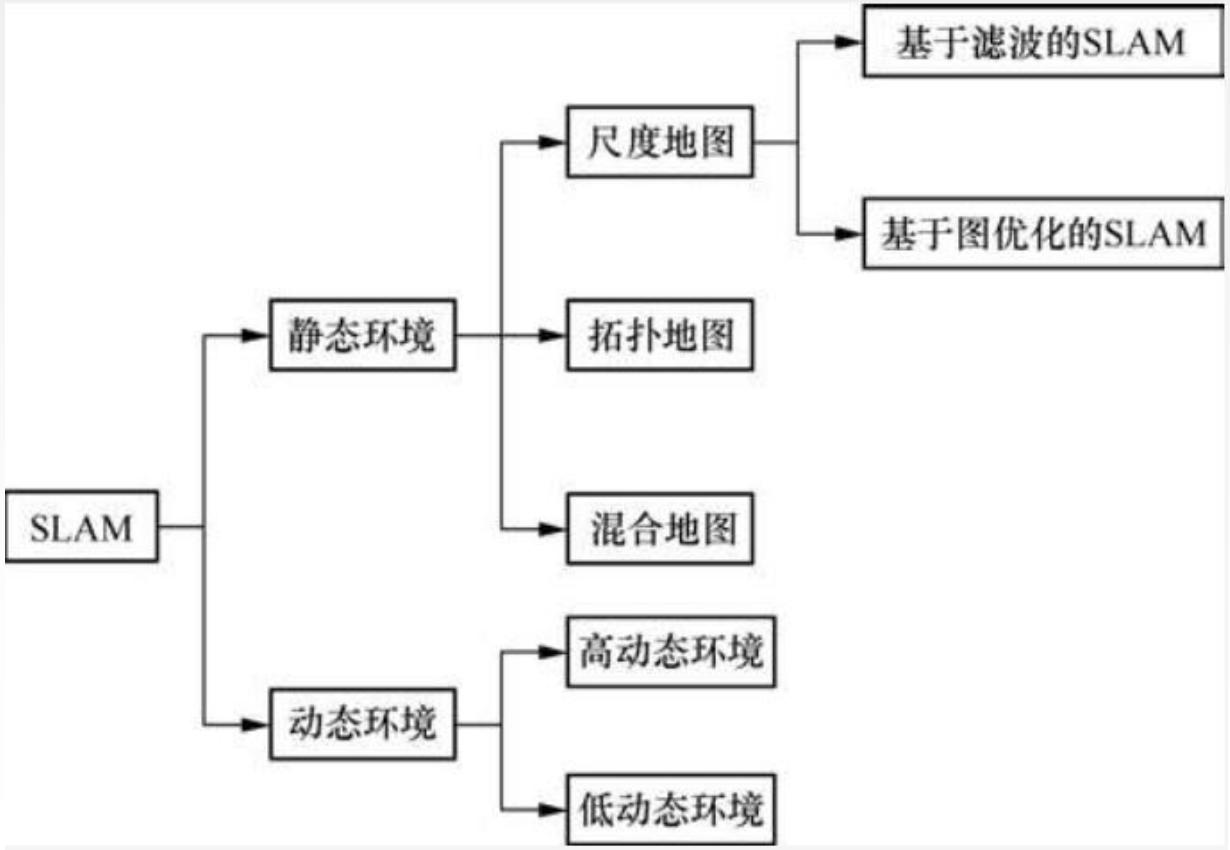
智能机器人SLAM

★ SLAM的定义

当机器人运行在一个未知环境中，而且不能根据外界的设备提供环境模型和实时位姿信息时，就需要该机器人具备对环境的感知能力，并且根据环境信息确定自身位置来规划行进路线。**SLAM是指机器人根据自身携带的传感器感知周围环境来创建环境模型，根据已建立的地图进行自身定位，并且不断地通过状态估计构建并更新地图的同时更新自身位置的过程。**因此，SLAM算法是机器人具有自主工作能力的基础。

★ SLAM的分类

SLAM的分类方法有很多种，较流行的分类方法是根据传感器的不同进行SLAM算法分类，可以分为基于激光雷达的SLAM和基于视觉的SLAM；也可以根据SLAM的应用环境进行分类，如图所示，分为静态环境SLAM和动态环境SLAM。静态环境又可以根据地图表达形式的不同分为尺度地图、拓扑地图、混合地图。



★ SLAM数学描述

SLAM问题的本质是状态估计问题：如何通过带有噪声的测量数量估计机器人的位姿状态。同时，定位与建图过程可以总结为运动模型和测量模型两部分。

运动模型描述的是机器人受控制系统而产生的运动；

测量模型描述的是机器人根据自身携带的传感器获取周围的环境信息。

为了更加科学和系统地求解这个过程，可以建立一个SLAM数学模型。

机器人的位姿需要用6个变量描述，包括机器人相对于外部坐标系的三维空间坐标(x、y、z)及3个欧拉角（横滚角、俯仰角、偏航角)。通常，机器人的运动被限制在某一平面上，此时机器人的位姿只需要用3个变量描述即可，包括机器人相对于外部平面坐标系的坐标及其方位角，机器人的位姿用向量$[x\ y\ \theta]^T$描述。

★ SLAM数学描述

运动模型

首先需要了解在运动过程中，机器人的位姿是如何变化的。假设一段时间内机器人在室内环境中移动，在运动过程中，每一时刻都有对应的机器人运动状态，可以根据机器人的内部传感器（如里程计）估算出机器人的状态，建立机器人的运动学模型。为了便于计算和理解，可以把时间用离散时刻表示，记为$t=1,2,3,\cdots k$，与时间对应的机器人的状态记为$x_1, x_2, x_3, \cdots\ x_k$，其用来描述机器人的运动轨迹。在机器人由状态$X_k$运动至状态$X_{k+1}$时，位姿状态由机器人运动控制系统与噪声共同决定，构成运动方程：

$$X_{k+1}=f(x_k, u_{k+1})+w_{k+1}$$

其中，是x_k时刻机器人的位姿信息，u_{k+1}是机器人在k+1时刻的运动控制，w_{k+1}是运动噪声。非线性函数f()是系统的状态转移矩阵。

★ SLAM数学描述

测量模型

机器人通过观测模型对周围环境进行物理建模。目前，机器人使用较多的传感器主要有视觉传感器和激光测距仪两类，针对不同的场景和不同的精度需求，可以选择合适的传感器。

当机器人在x_k位置处通过第j个传感器观测到环境信息y_j，产生观测数据z_k时，可以用观测方程表示为：

$$z_{k,j} = h\left(x_k, y_j\right) + v_{k,j}$$

其中，x_k是机器人的位姿状态信息，$v_{k,j}$是传感器的观测噪声，h(x)是抽象的测量函数，根据不同的传感器转换为不同的形式。

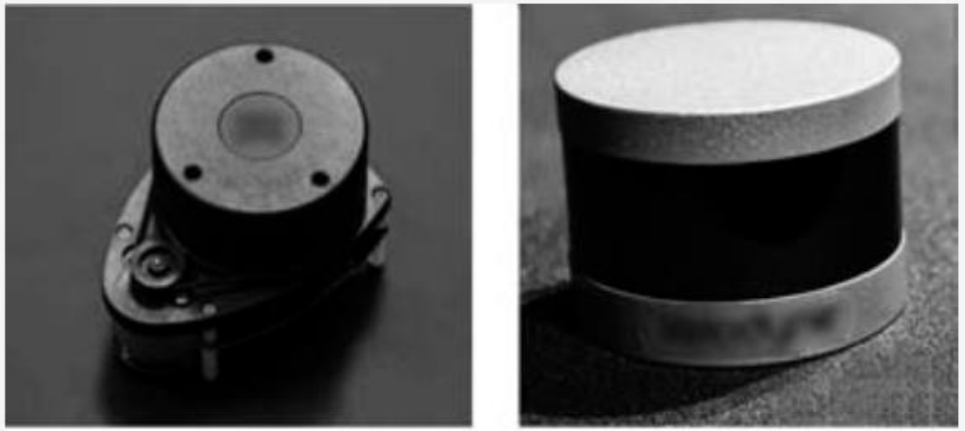
★ SLAM数学描述

SLAM本质上是对运动状态和观测状态进行估计的过程，理论上可以分为两类：基于滤波的SLAM（Filter-based SLAM）算法和基于图优化的SLAM（Graph-based SLAM）算法。基于滤波的SLAM算法根据对后验概率表示的不同，可以分为基于扩展卡尔曼滤波器的SLAM（extended Kalman filter based SLAM, EKF-SLAM）、基于粒子滤波器的SLAM（particle filter based SLAM,PF-SLAM）等。这类方法根据观测信息和运动信息结合贝叶斯原理与马尔可夫假设，对当前时刻的位姿和地图进行预测和更新。基于图优化的方法将SLAM算法分为前端和后端，前端通过顺序数据关联和回环检测构建位姿图，后端使用优化算法对位姿图进行全局优化。

机器人激光雷达SLAM

★ 激光雷达

机器人根据实现不同的功能需求搭载不同的传感器，相比于视觉传感器（如深度相机），激光雷达SLAM具有测量精确、测量距离远、可融合多传感器、受光线环境因素影响较小、能够生成便于导航的环境地图等优点。激光雷达分为2D激光雷达和3D激光雷达。2D激光雷达为单线激光雷达，如图(a)所示，造价低廉，适用于室内环境。3D激光雷达为多线激光雷达，如图(b)所示，具有扫描频率高，采样点密集的特点，常用于室外环境。在激光SLAM中，激光雷达作为观测模型的传感器感知周围的环境信息，惯性测量单元(inertial measurement unit, IMU）或者里程计（odometry）作为运动模型的传感器，配合完成SLAM过程。



（a）2D激光雷达　　（b）3D激光雷达

★ 基于扩展卡尔曼滤波的SLAM

在机器人定位与导航章节，我们已经利用卡尔曼滤波方式介绍了定位与建图，而SLAM即同步定位与建图，将机器人的定位与建图用卡尔曼滤波方式进行进一步融合，即可完成基于扩展卡尔曼滤波的SLAM。

新的状态向量由机器人当前位姿和已被探测到的地标状态组成：

$$\hat{q}=\left[\left(q_k\right)^{\mathrm{T}} \quad \left(q_m^1\right)^{\mathrm{T}} \quad \left(q_m^2\right)^{\mathrm{T}} \quad \cdots \quad \left(q_m^i\right)^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbf{R}^{3+2\cdot i}$$

对应的协方差矩阵$\hat{P}$变为：

$$\hat{P}=\begin{bmatrix} P_q & P_{qM} \\ \left(P_{qM}\right)^{\mathrm{T}} & P_M \end{bmatrix}$$

其中P_q是机器人位姿的协方差矩阵，P_M是地标状态的协方差矩阵，P_{qM}代表机器人与地标之间的状态关联矩阵。

★ 基于扩展卡尔曼滤波的SLAM

当检测到一个新的地标时，用来扩展协方差矩阵的Y有所变化：

$$\nabla Y_{\hat{q}} = \begin{bmatrix} I^{n\times n} & 0^{n\times 2} \\ \begin{bmatrix} G_q & 0^{2\times(n-3)} \end{bmatrix} & \dfrac{\partial g}{\partial z} \end{bmatrix}$$

其中：

$$G_q = \frac{\partial g}{\partial q_k} = \begin{bmatrix} 1 & 0 & -z_r \sin(\theta_k + z_\alpha) \\ 0 & 1 & z_r \cos(\theta_k + z_\alpha) \end{bmatrix}$$

这是因为新地标的状态估计式与机器人状态有关，而新的状态向量式中包含q_k。

因为根据新的状态向量，传感器观察到的信息和机器人位姿状态及地标的位置状态都有变化，

因此等到雅可比举证如下：

$$H_{\hat{q}} = \begin{bmatrix} \dfrac{\partial h}{\partial q_k} & \cdots 0 \cdots & \dfrac{\partial h}{\partial q_m^j} & \cdots 0 \cdots \end{bmatrix}$$

★ 基于扩展卡尔曼滤波的SLAM

基于扩展卡尔曼滤波的SLAM方法是SLAM问题的经典解决方案，本节重点阐述其中的数据关联问题。

SLAM问题中，数据关联是将当前机器人观测到的传感器数据与地图中的物体一一对应的过程。只有有了数据关联的结果，SLAM算法才能够利用观测数据与预测地图之间的新息（innovation）对系统状态进行更新。它是所有基于EKF的SLAM算法都必须要解决的一个问题。在SLAM问题中，错误的数据关联将会导致状态更新时地图中的物体得到不正确的更新，使地图出现扭曲，甚至会直接导致整个SLAM的结果发散到使算法无法继续下去的地步。因此，数据关联在SLAM问题中至关重要，只有有了正确的数据关联，SLAM算法才能给出满意的结果。

传统上解决数据关联的方法都是基于特征地图，常用的方法有**最近邻算法**、**联合相容分支界定算法**、**基于极大似然法的方法**、**多假设跟踪算法**等。

★基于扩展卡尔曼滤波的SLAM_最近邻算法

最近邻（NN）算法采用马氏距离度量当前测量值中的某个特征与地图中某个特征的预测值之间的距离，并用一个阈值判断两者是否为同一特征。当两者之间的马氏距离小于该阈值时，则认为两者关联；否则，认为它们不关联。

NN算法整体流程简洁，计算速度快，是SLAM领域里最常用的一种数据关联算法。当观测到的物体不多并且机器人具有较高精度的运动学模型时，NN算法能够快速给出正确的结果。但是，当观测到很多物体，并且这些物体之间的距离比较接近，或者机器人运动学模型的精度比较低的时候，NN算法通常很难得到正确的关联结果。

★基于扩展卡尔曼滤波的SLAM_联合相容分支界定算法

联合相容分支界定算法（JCBB)采用联合相容检验准则判断所有观测值和局部地图中的特征之间的关联。相较于NN算法只使用独立相容性，也就是单个观测值和预测值之间的距离，该算法利用了预测状态之间的协相关性，综合考虑所有可能关联的联合相容性挑选最佳数据关联，从而在机器人位姿估计出现较大偏差时能够避免错误的数据关联。

JCBB算法将地图中各个特征之间的协相关性引入数据关联，可以从整体上判断多个数据关联的总体效果，因而可以排除许多因几何距离比较近而引入的错误的关联假设，相比于NN算法，它的鲁棒性更强。

★基于扩展卡尔曼滤波的SLAM_基于极大似然法的方法

极大似然法（ML）从概率最大化的角度提出了一个关于当前数据关联方案的概率函数，而最佳的数据关联是能够使该概率函数取极大值的那个数据关联。极大似然法是一种常用的统计学方法，用于估计未知参数。它的基本思想是，给定一组观测数据，我们可以通过最大化似然函数来估计未知参数的值。似然函数是指在给定参数下，观测数据出现的概率。因此，极大似然法的目标是找到一个参数值，使得观测数据出现的概率最大。

极大似然法的优点是简单易用，而且在很多情况下都能得到比较准确的估计结果。但是，它也有一些缺点。首先，它需要假设数据的分布形式，如果这个假设不正确，那么估计结果可能会出现偏差。其次，它对于极端值比较敏感，如果数据中存在一些极端值，那么估计结果可能会受到影响。

★基于扩展卡尔曼滤波的SLAM_多假设跟踪算法

前面提到的NN、JCBB及基于ML的算法都是用当前时刻的预测值与当前时刻的测量值进行关联，在路标比较密集的情况下，仅依靠当前时刻及之前的信息很难获得正确的数据关联。在这种情况下，可以同时保存各种可能的数据关联，通过之后一段时间采集到的数据对各种数据关联方案进行评价，最终获得正确的数据关联，这就是多假设跟踪（MHT）算法的基本原理。

基于MHT的数据关联算法通常都通过一个有限长度的时间窗口建立多个候选关联假设，假设的产生可以使用前面介绍的NN、JCBB等算法。然后，基于各个候选的数据关联独立进行SLAM过程，并用在窗口时间段内的观测值评价各个候选假设的优异。随着时间的推移，错误的数据关联的评价会越来越低，直至被剔除，最终只有正确的数据关联才能被保存下来。

与其他数据关联算法相比，基于MHT的数据关联方法在某一时刻的数据关联不仅使用了该时刻及之前的信息，而且还延迟使用了该时刻之后的传感器信息，因而具有极高的准确性。但是，由于其同时维持了多种可能的数据关联方案，需要并行运行多个SLAM算法，由此带来存储量和计算量的迅速增加。所以，在实际应用中，该方法受到较大的限制。

★ 基于粒子滤波的SLAM

FastSLAM是粒子滤波在SLAM中的应用。其将SLAM问题分解成两个部分：机器人运动轨迹的估计和基于轨迹的环境模型描述，使用粒子滤波对轨迹进行估计，每个粒子表示一种可能的轨迹；使用扩展卡尔曼滤波进行环境建模，根据每个粒子的轨迹估计及观测信息对环境进行描述，并根据观测信息对该粒子进行权重计算，评价粒子的好坏，则有：

$$p(\boldsymbol{X}_{1:k}, \boldsymbol{X}_l \mid \boldsymbol{Z}_{1:k}, \boldsymbol{u}_{1:k}) = p(\boldsymbol{X}_{1:k} \mid \boldsymbol{Z}_{1:k}, \boldsymbol{u}_{1:k}) p(\boldsymbol{X}_l \mid \boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k})$$

其中，$p(X_{1:k} \mid Z_{1:k}, u_{1:k})$表示根据观测信息和运动控制量对机器人位姿进行估计，$p(X_l \mid X_{1:k}, Z_{1:k})$表示根据机器人位姿及观测信息对环境特征的位置进行估计。

FastSLAM可以分为3步：根据运动模型及轨迹的先验信息采样新位姿；用EKF对观测数据进行处理，对环境模型进行描述；根据观测信息计算粒子的权重，根据权重进行重采样。

★ 基于粒子滤波的SLAM_采样新位姿

运动模型如上页所示，其中，对于FastSLAM，X_k只表示机器人的位姿。根据贝叶斯原理和马尔可夫假设，将运动模型表示成递归形式，如下所示。

$$
\begin{aligned}
p(X_{1:k} \mid Z_{1:k}, u_{1:k}) &= \eta p(Z_k \mid X_{1:k}, Z_{1:k-1}, u_{1:k}) p(X_{1:k} \mid Z_{1:k-1}, u_{1:k}) \\
&= \eta p(Z_k \mid X_k) p(X_{1:k} \mid Z_{1:k-1}, u_{1:k}) \\
&= \eta p(Z_k \mid X_k) p(X_k \mid X_{1:k-1}, Z_{1:k-1}, u_{1:k}) p(X_{1:k-1} \mid Z_{1:k-1}, u_{1:k}) \\
&= \eta p(Z_k \mid X_k) p(X_k \mid X_{1:k-1}, u_k) p(X_{1:k-1} \mid Z_{1:k-1}, u_{1:k-1})
\end{aligned}
$$

其中，其中，$p(X_{1:k-1} \mid Z_{1:k-1}, u_{1:k-1})$表示k-1时刻粒子群的分布。通过上式及上页的运动模型，即可根据上一时刻的粒子群分布和运动控制量估计下一时刻的粒子群分布。

★ 基于粒子滤波的SLAM_环境模型的建立

环境模型的建立使用EKF算法，并且在已知机器人位姿的情况下对观测信息进行预测和更新。不同于EKF-SLAM,FastSLAM在使用EKF时的系统状态维度为二维，即特征在环境中的位置。而EKF-SLAM中系统状态的维度为2N+3,其中N表示环境中的特征个数。因此，随着观测到的环境特征越来越多，EKF-SLAM的维数将变得很大，由于涉及矩阵求逆，算法的整体计算量也会急剧增大，因此，FastSLAM在对环境进行建模时的计算复杂度比EKF-SLAM小得多。

★ 基于粒子滤波的SLAM_粒子权重的计算及重采样

在FastSLAM中，重采样是保证算法准确性和效率的关键。通过对每个粒子计算权重，权重较大表示真实分布与建议分布相似；权重较小表示该粒子代表的轨迹可信度较低。权重的计算公式如式所示。

$$
\begin{aligned}
w_k &= \frac{\text{真实分布}}{\text{建议分布}} \\
&= \frac{\eta p(\boldsymbol{Z}_k \mid \boldsymbol{X}_k)\, p(\boldsymbol{X}_k \mid \boldsymbol{X}_{1:k-1}, \boldsymbol{u}_k)\, p(\boldsymbol{X}_{1:k-1} \mid \boldsymbol{Z}_{1:k-1}, \boldsymbol{u}_{1:k-1})}{p(\boldsymbol{X}_k \mid \boldsymbol{X}_{1:k-1}, \boldsymbol{u}_k)\, p(\boldsymbol{X}_{1:k-1} \mid \boldsymbol{Z}_{1:k-1}, \boldsymbol{u}_{1:k-1})} \\
&\models \eta p(\boldsymbol{Z}_k \quad \boldsymbol{X}_k)
\end{aligned}
$$

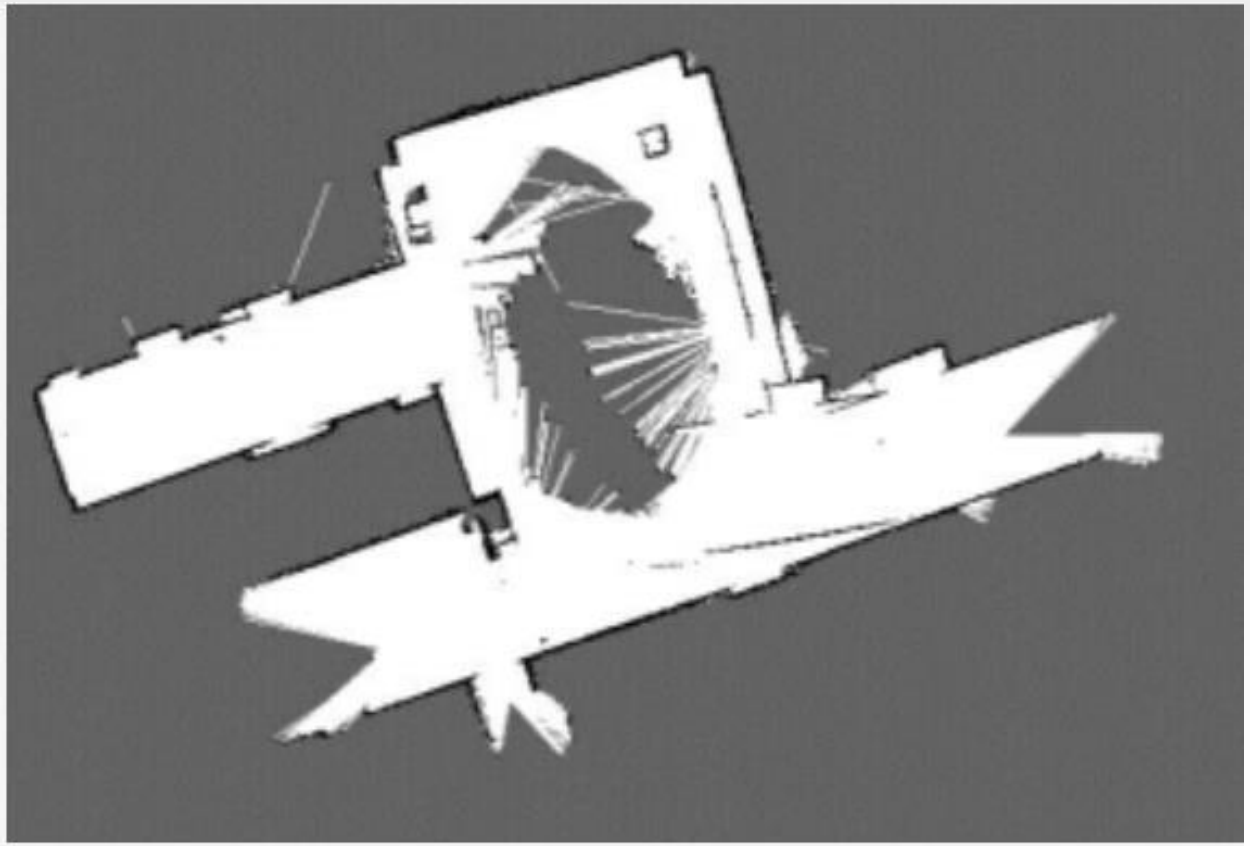
FastSLAM算法使用权重较大的粒子表示机器人的运动轨迹，在进行重采样时，权重较小的粒子被剔除。但是，每一步都进行重采样会造成粒子多样性减少，并且增加计算量。FastSLAM算法使用有效粒子数决定是否需要进行重采样：

$$
N_{\text{eff}} = \frac{1}{\sum \left(w^i\right)^2}
$$

当N_{eff}较大时，说明有效粒子较多，粒子之间的差异性较小，不需要进行重采样；当N_{eff}较小时，说明有效粒子较少，粒子之间的差异性较大，此时需要对粒子进行重采样。

★ 基于粒子滤波的SLAM

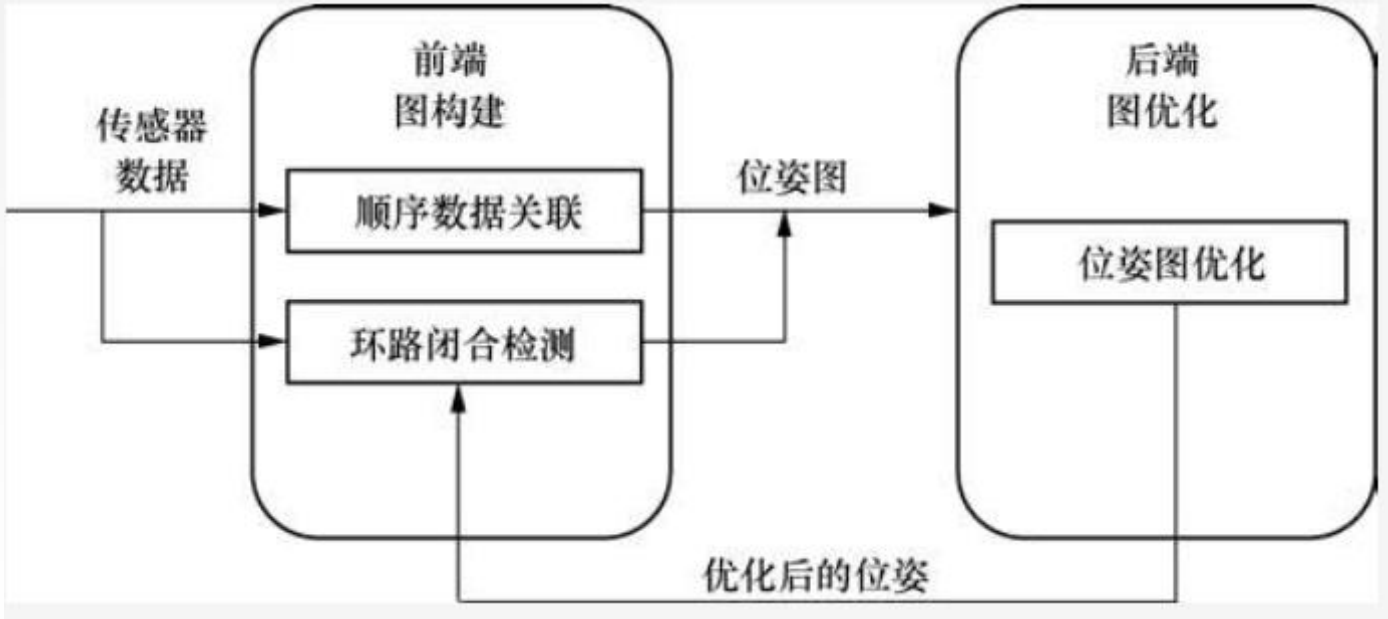
下图为基于粒子滤波的SLAM算法创建的栅格地图，可以看出地图的结构，但是在回环检测方面性能较弱，导致地图不够精确。



机器人激光雷达SLAM

★ 基于图优化的SLAM

基于图优化的SLAM方法将SLAM问题描述成图，图由节点和边组成，节点表示机器人的位姿和观测信息，边表示节点之间的约束关系。基于图优化的SLAM方法可以分为前端和后端两部分，如图所示。前端主要用来处理传感器的数据，构建位姿图(pose graph)；后端对位姿图进行优化，得到最优的机器人轨迹及全局地图。



顺序数据关联对连续两帧的传感器数据进行融合，并且计算机器人的位姿变换。环路闭合检测是将当前传感器的观测值与机器人已经获取的所有传感器数据进行对比，判断机器人是否出现在曾经到过的地方，如果是，则在这两个节点之间添加一条回环约束。前端用来处理激光雷达、相机、里程计等传感器数据，后端直接对前端构建的位姿图进行优化。

★ 基于图优化的SLAM_顺序数据关联

在激光SLAM中，由于前后两帧激光数据创建的栅格子地图比较相似，因此一般采用扫描匹配实现数据关联。扫描匹配是指找到一个最优的变换矩阵，将当前时刻机器人创建的栅格子地图与前一帧子地图融合在一起，也就是将当前帧的栅格子地图变换到前一帧子地图的坐标系下：

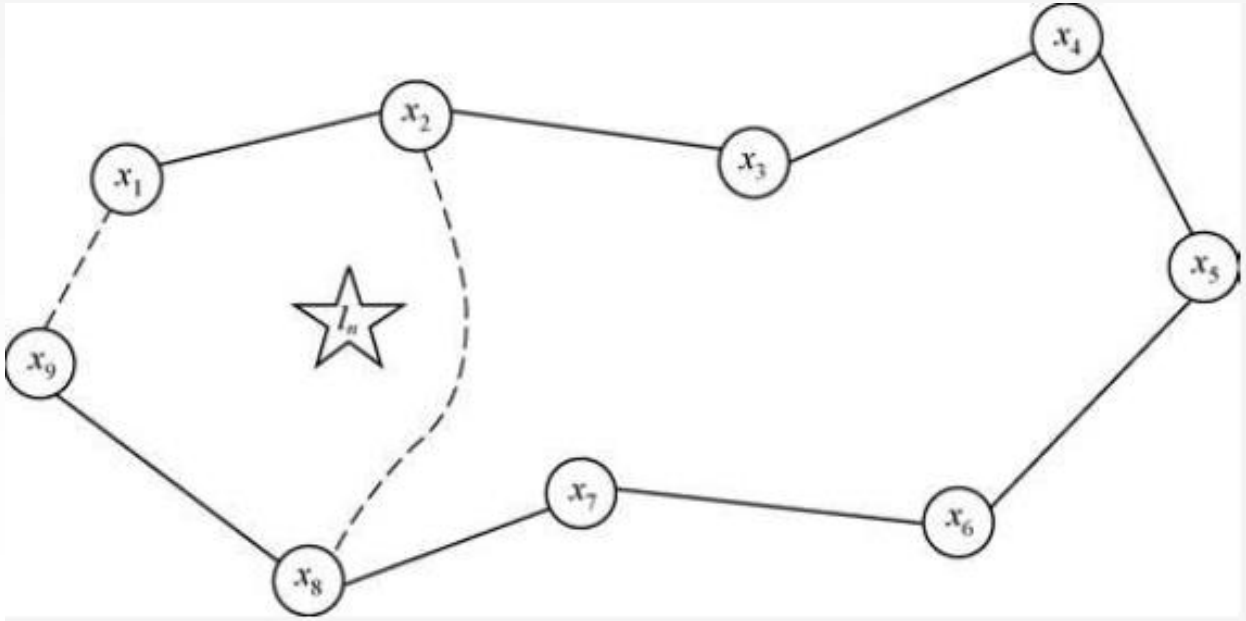
$$G_{k,k+1} = T_k^{k+1} G_{k+1}$$

其中，$G_{k,k+1}$表示k时刻创建的栅格子地图，T_k^{k+1}表示将k+1时刻创建的栅格子地图变换到k时刻创建的栅格子地图的坐标系下。 T_k^{k+1}由旋转和平移组成。

★ 基于图优化的SLAM_环路闭合检测

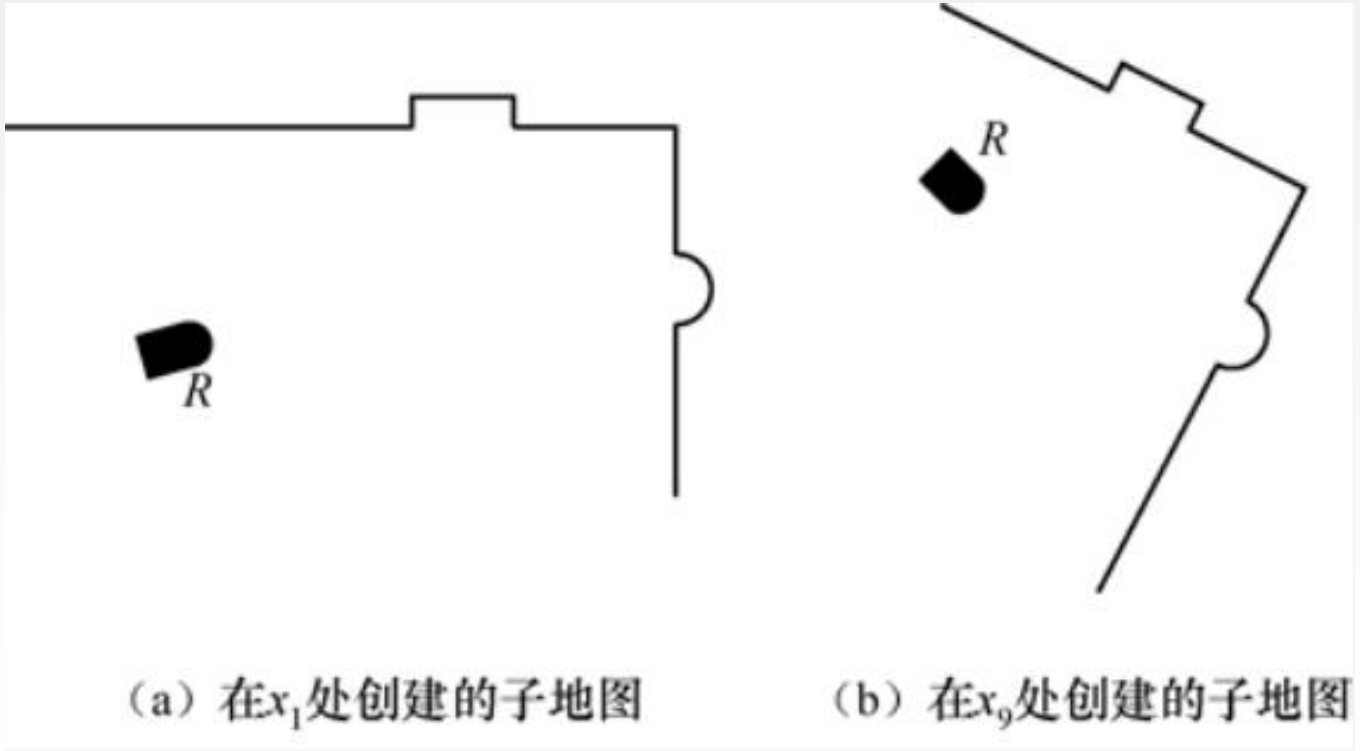
环路闭合检测是判断机器人当前所处的位置是否曾经到达过。正确的回环可以减少累积误差，保证所创建地图的一致性；错误的回环带入后端优化中，优化结果会出现严重的误差，导致整个SLAM算法失败。因此，环路闭合检测对于SLAM算法是很重要的一环。

下图是一个位姿图的示意图，图中{x1, x2, …, x9)表示机器人在不同时刻的位姿，也就是位姿图中的节点，连接两个连续节点之间的边是通过顺序数据关联得到的约束，l_n表示环境中的路标。假设机器人在x_1和x_9处观测到环境中的同一个墙角。



★ 基于图优化的SLAM_环路闭合检测

下图为机器人在不同时间创建的子地图，图(a)表示机器人在x_1处创建的子地图，图(b)表示机器人在x_9处创建的子地图，通过匹配算法可以确定机器人在x_1和x_9位置创建的子地图对应环境中的相同位置，即达到了回环要求，则在位姿图中的这两个节点之间添加回环约束，如上页图中的虚线所示。



（a）在x_1处创建的子地图　　（b）在x_9处创建的子地图

★ 基于图优化的SLAM_位姿图优化

后端优化通过构建全局误差的最小二乘形式实现：

$$e_{ij} = \hat{z}_{ij} \ominus z_{ij}$$

$$E_{ij} = e_{ij}^{\mathrm{T}} \Omega_{ij} e_{ij}$$

$$F(x) = \sum E_{ij} = \sum e_{ij}^{\mathrm{T}} \Omega_{ij} e_{ij}$$

$$x^{*} = \arg\min_{x} F(x)$$

$\hat{z}_{ij}$ 表示根据机器人的运动信息计算出的相对位姿的变换，

z_{ij} 表示根据传感器观测值计算出来的变换。

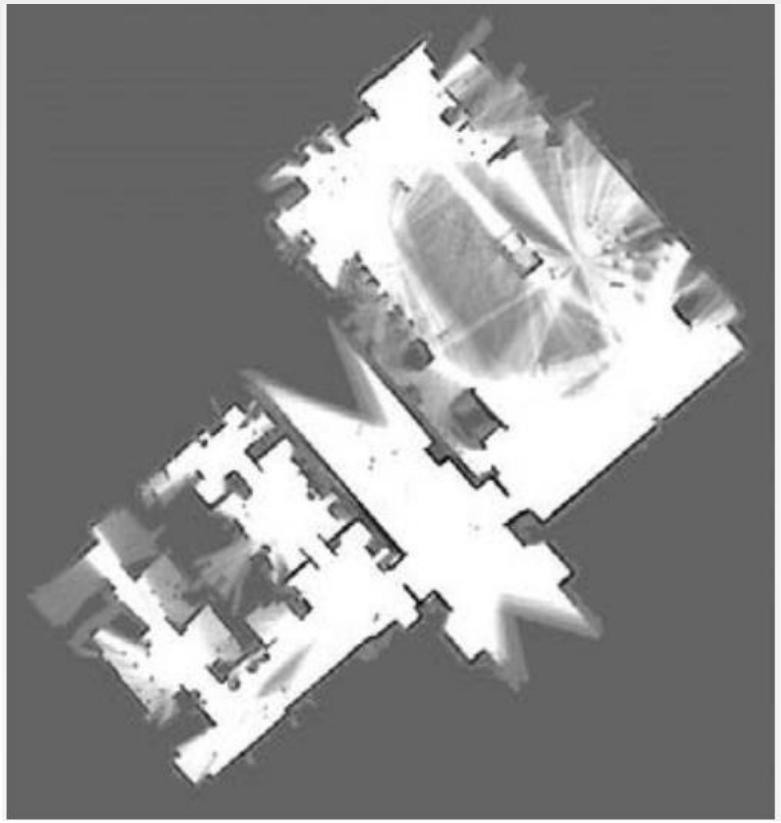
Ω_{ij} 为信息矩阵，是协方差矩阵的逆，用来表示数据的可信度。

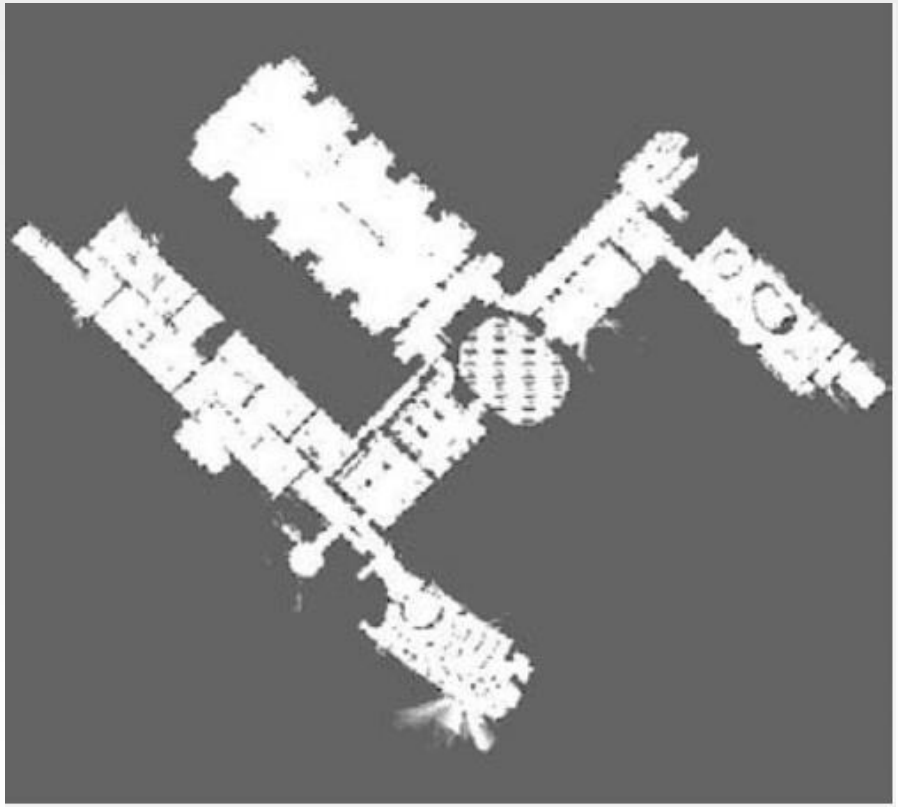
后端优化通过求解全局误差的最小值计算机器人的最优轨迹，再根据该轨迹创建全局地图。

机器人激光雷达SLAM

★ 基于图优化的SLAM

图为基于图优化SLAM算法创建的栅格地图，从中可以看出基于图优化的方法能够实现局部回环和全局回环检测，这样可以修正由于机器人传感器测量噪声带来的误差，可以创建更加准确的环境地图。





VSLAM基础

★ VSLAM的概念

相机具有成本低、便于携带、获取的环境信息丰富等优点。VSLAM是通过相机获取环境信息并实现机器人定位与环境建模的，如图所示，根据相机类型的不同，可以将VSLAM分为三大类：基于单目相机的单目SLAM、基于双目相机的双目SLAM和基于深度相机的RGB-D SLAM。

(a) 单目相机

(b) 双目相机

(c) 深度相机

VSLAM基础

★ VSLAM的概念

单目相机只有一个摄像头，结构简单，成本非常低，但单目相机拍出的图像中的像素没有尺度信息，需要辅助方法确定像素的深度。MonoSLAM是第一个实时的单目VSLAM系统。当摄像头在空间运动时，MonoSLAM可以利用特征匹配的方法，估计摄像头的位置和特征点的位置。

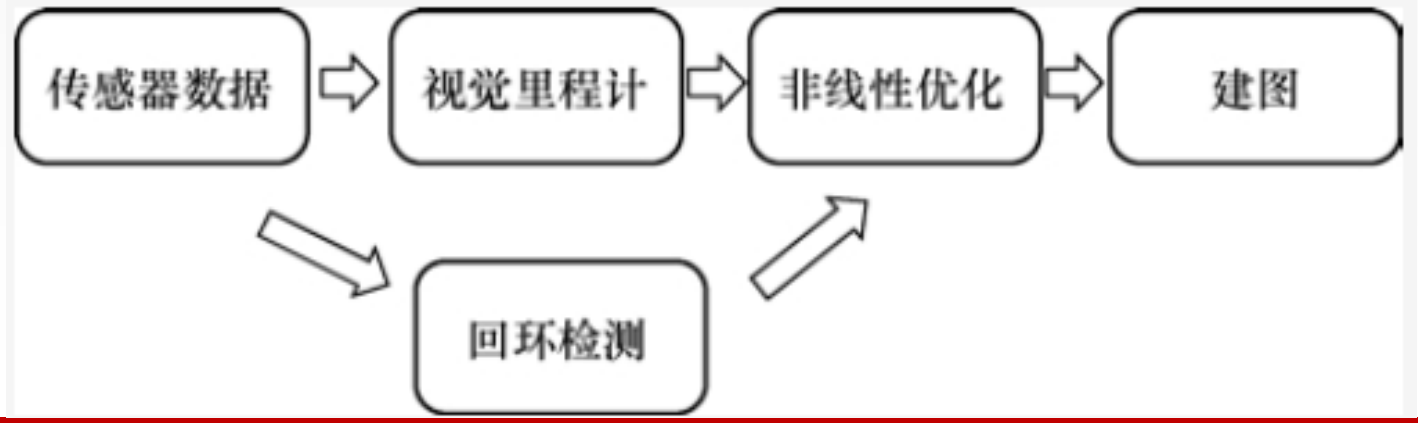
双目相机，通过左右图像中像素的差异可以计算出像素点的深度，由于深度信息是通过计算得到的，因此双目相机SLAM需要大量的计算，而且计算得到的像素深度的误差较大，影响最终地图的精度。基于双目相机的常用的VSLAM算法有LSD算法等。

深度相机不再是通过计算获得，而是通过物理测量手段获取，可以节省大量的计算量。但是，深度相机获取深度的精度易受环境影响，如日光导致得到的结果中噪声很大，一般只在室内应用。常见的深度相机有Kinect、Astra Pro、 Xtion Pro Live和Realsense等。优秀的RGB-D SLAM算法有KinectFusion、RGB-D SLAM2、RTAB-Map等。

VSLAM基础

★ VSLAM的概念

VSLAM技术已经形成较为稳定的理论框架，如图所示。VSLAM框架可以分为**传感器数据**、**视觉里程计**、**非线性优化**、**回环检测**和**建图**5个模块。首先，通过相机传感器获取周围环境信息，可以获得环境彩色图像信息，如果相机是深度相机，还会获得环境的深度图像信息。其次，由视觉里程计根据相机得到一系列的图像信息估计相机的位姿变换，通过计算相邻两帧图像的相对位姿变换，形成视觉里程计。然后，通过后端优化计算相机位姿。后端优化是基于状态估计理论或者是图优化理论实现的，视觉里程计中的任何一帧都会被传入前端进行定位，而后端优化中只有关键帧才会被保存下来进行优化。随着环境地图的增长，累积误差逐渐增大，最明显的表现是当相机回到之前经过的位置，地图没有形成闭环。因此，需要检测相机有没有回到曾经到达的地方，这个过程叫作回环检测。



★ 特征提取

在环境建模的过程中，相机处于移动状态，期望通过某种方法估计相机位姿变化。估计相机运动的方法有：**特征点法**和**直接法**。

特征点是图像中灰度值变化剧烈或者在图像边缘上曲率较大的点。通过对特征点建立独特的描述子，可以将不同图片上的相同位置特征点关联，得到特征点对，然后估计两帧之间相机的运动。常用的特征检测方法有Harris、 SIFT(scale invariant feature transform)、SURF (speed up robust feature)、 ORB (oriented FAST and rotatedBRIEF) 等

直接法，根据相机的亮度信息估计相机的运动，可以不需要计算关键点和描述子，优化的是光度误差，根据使用像素数量可分为稀疏、半稠密、稠密三种。常见开源方案有SVO，LSD-SLAM等。

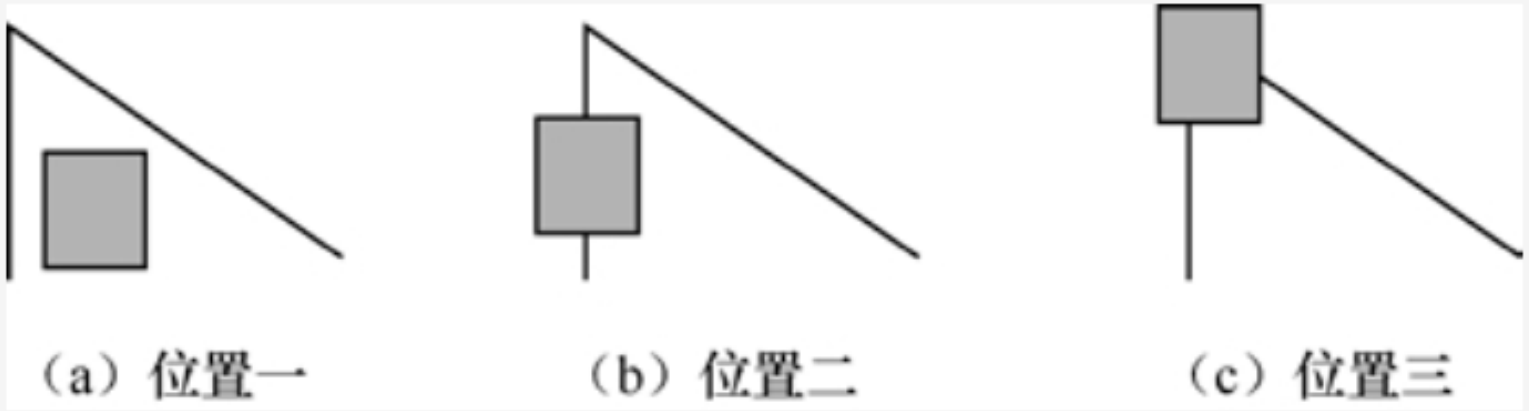
★ 特征提取-Harris算法

Harris算法是一种经典的特征点提取的方法，特征点被定义为在各个方向上灰度值都有变化的点。因此，特征点是两条边缘线的相交点，如图所示。Harris算法得到的特征点很容易在图像中定位，并且大量存在于环境中，如桌椅、门窗等。

不同类型的特征点

★ 特征提取-Harris算法

Harris算法的主要思想是在图像中设定一个小窗口，如图所示。图(a)中窗口内图像的灰度值没有发生变化，窗口内就不存在特征点；图(b)中窗口在某一方向移动时，窗口内图像的灰度值发生了较大的变化，而在另一些方向上图像的灰度值没有发生变化，则窗口内的图像可能是一条直线的线段；图(c)中窗口内若图像的灰度值在任意方向都发生了较大的变化，则在窗口内遇到了特征点。



★ 特征提取-ORB方法

ORB方法特征提取部分使用FAST（features from accelerated segment test）算法，描述子为BRIEF （binary robust independent elementary features）。

★ 特征提取-ORB方法

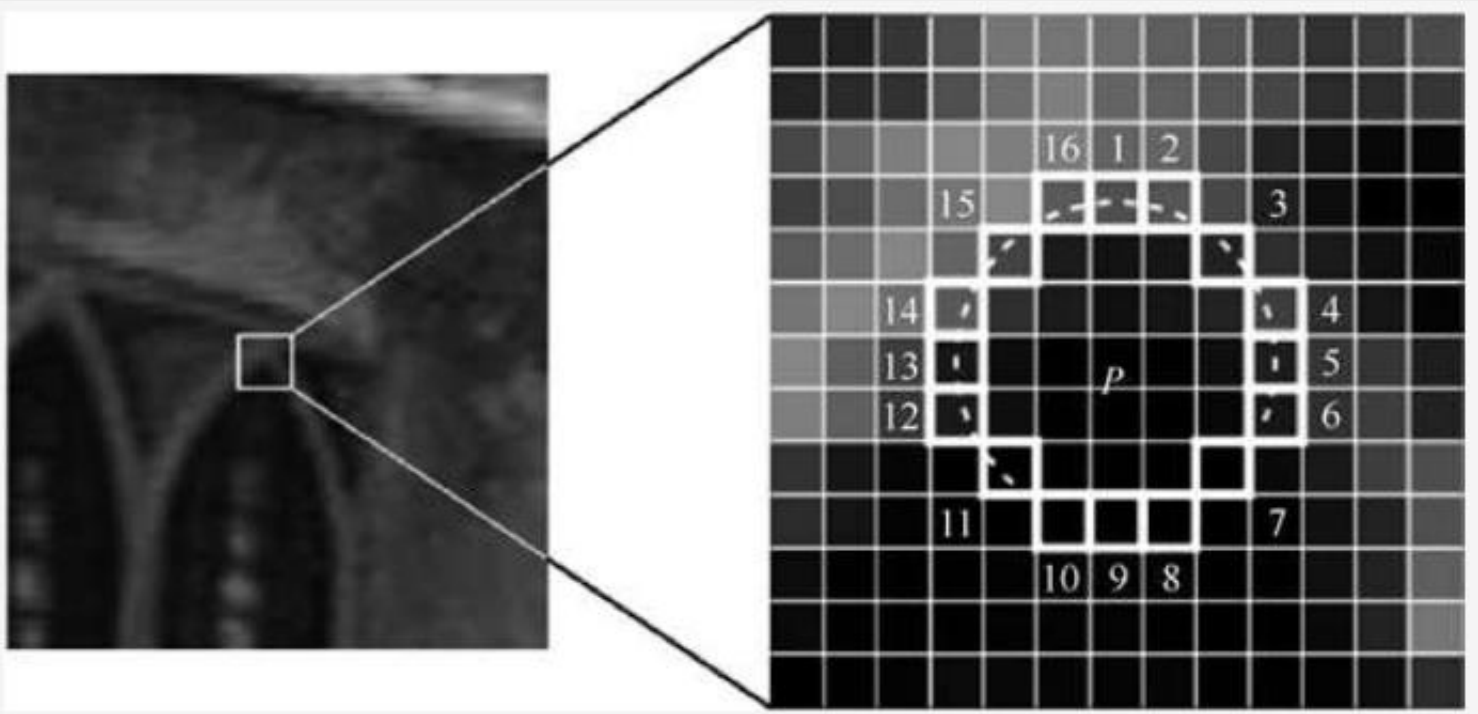
FAST算法只需要比较像素之间灰度的不同，特征点的提取过程简单、快速，准确率较高。图为FAST特征点的提取示意图。FAST特征点的提取过程如下。

(1) 取图像中的像素点P，点P的灰度值表示为I_p。

(2) 以像素点P为圆心3为半径，得到圆上16个像素点，如图所示。

(3) 在16个像素点中，如果出现连续M个像素点，它们的灰度值都比I_p +T大或者都比I_p -T小，则点P为FAST特征点。T为设定的灰度差阈值，M的值可取9或者12。

(4) 遍历所有像素，重复上述过程，完成整张图片FAST特征点的提取。



FAST特征点的提取示意图

★ 特征提取-ORB方法

为了提高特征点的提取速度，减少检测的像素点数量，将本来需要检测的16个像素点变为只检测4个。首先，在上、下、左、右位置各取一个像素，对应图中1、5、9和13号位置的4个像素点。当1、5、9和13号位置中至少有3个位置的像素灰度值同时都比I_p +T大或者都比I_p -T小，则这个点就可能是特征点，否则，就排除了该点。初步筛选后，得到的待检测像素点数量大量减少，再通过上述FAST特征点提取方法对待检测像素点检测。

FAST提取的特征点常出现点密集的现象。在一个区域内，如果特征点比较密集，则需要对其进行非极大值抑制算法处理，去除局部较密集的特征点。通过建立一个模型，抑制响应值非极大的特征点。

★ 特征提取-ORB方法

此外，FAST特征点没有尺度和方向，通过对图像的不同层次降采样，获得不同分辨率的图像，建立图像金字塔，在构建的金字塔的不同层次上检测特征点，解决FAST尺度问题。FAST特征点的方向问题利用灰度质心法解决，其具体过程如下。

在图像中将某小块图像的矩阵m_{pg}用下式表述，可将其具体描述为：

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), \quad p,q = \{0,1\}$$

用C表示图像块的质心，则质心表述为：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

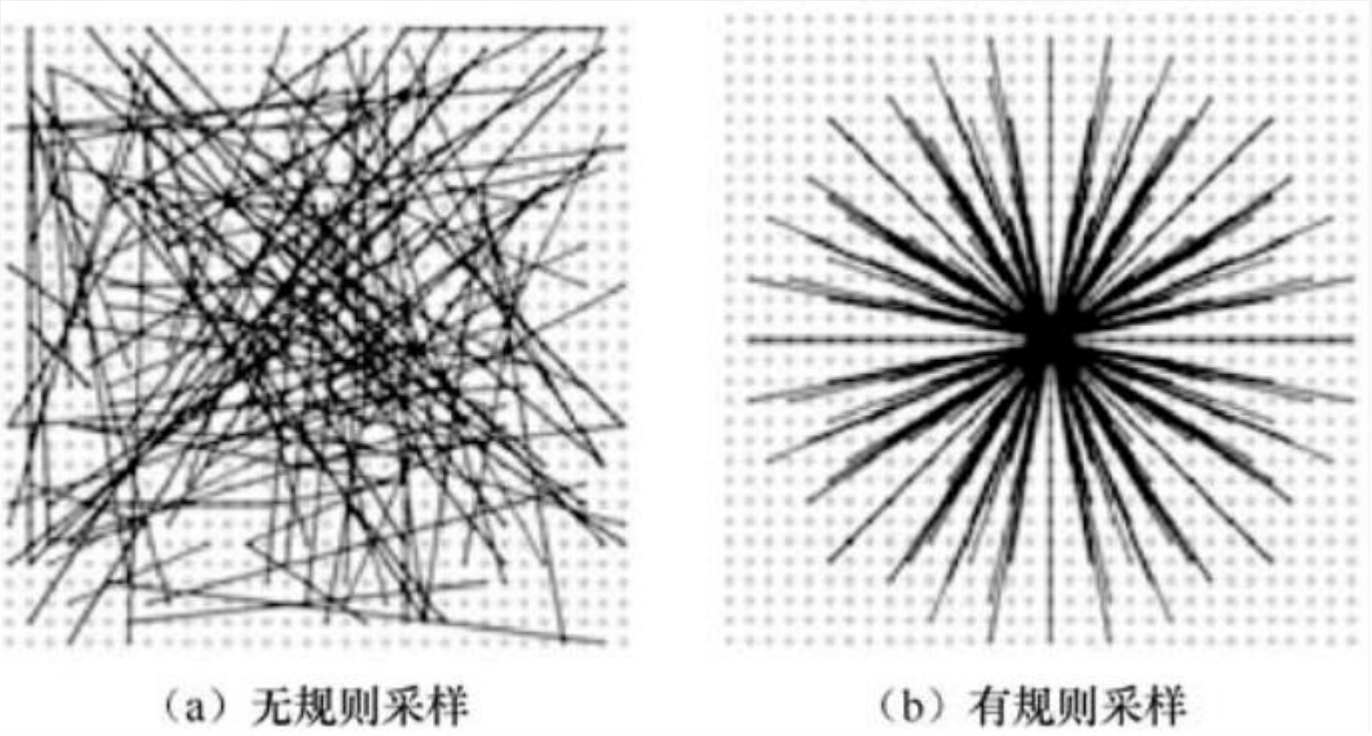
设图像的几何中心位置为O，通过式C式得到质心位置C，此时可以得到向量OC，也就是特征点的方向，用 θ 表示特征点方向，可以表述为：

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

★ 特征提取-ORB方法

BRIEF是基于二进制编码的特征描述子，利用汉明距离描述特征点的相似性，汉明距离越小，表示这两个点的匹配程度越大。采用二进制表示描述子，不仅可以节约存储，更重要的是，可以通过简单的异或操作与位值计算实现特征匹配，提高特征点匹配的效率。

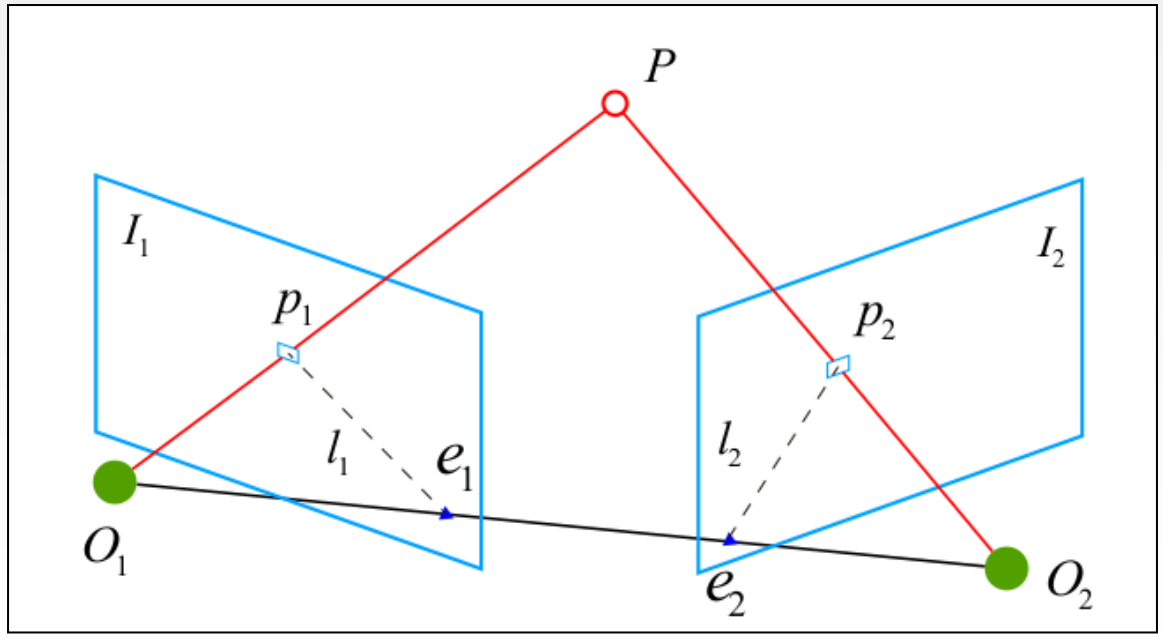
BRIEF在特征点的周围采样像素点数一种是无规则的采样方式，如图(a)所示，另一种是有规则的采样方式，如图(b)所示。采样像素点的个数可以为128、256或512。若像素点的灰度值大于或等于阈值，则该像素点对应的二进制值为1；否则，该像素点对应的二进制值为0。



（a）无规则采样　　（b）有规则采样

★ 视觉里程计

通过特征提取的方法得到相邻帧之间的特征点点对，然后根据得到的特征点点对计算相邻帧相机的运动，一系列相机的运动形成视觉里程计。根据使用的相机不同，当传感器是单目时，已知2D的像素坐标，属于2D-2D模型。当相机为双目、RGB-D时，已知像素点的深度，属于3D-2D、3D-3D模型。



首先是2D-2D模型，已知两个相邻帧和图中的匹配点对，要求求出这两个相邻帧之间相机的运动。如图所示，点P为在两个视角下看到的同一个点，连接O_1P和O_2P分别交成像平面I_1于点p_1，交成像平面I_2于点p_2，连接O_1O_2交成像平面于点e_1、e_2，称直线p_1e_1为成像平面I_1上点p_1对应的极线。成像平面I_1上所有点的极线都会经过e_1，把点e_1称为成像平面I_1的极点。同理，点e_2称为成像平面I_2的极点，这样的约束称为**对极几何约束**。

★ 视觉里程计

设第一帧的坐标系下P的空间位置为：$P = [X, Y, Z]^T$

两个像素点p_1，p_2的像素坐标（单位像素）：$s_1 p_1 = KP, \quad s_2 p_2 = K(RP + t)$

使用齐次坐标：$p_1 = KP, \quad p_2 = K(RP + t)$　　取：$x_1 = K^{-1} p_1, \quad x_2 = K^{-1} p_2$

x_1, x_2是两个像素点的归一化的坐标，代入上式得：$x_2 = Rx_1 + t$

两边同时左乘t^：$t\hat{}x_2 = t\hat{}Rx_1$　　再左乘一个x_2^T 有：$x_2^T t\hat{}x_2 = x_2^T t\hat{}Rx_1$

由式可得：$x_2^T t\hat{}Rx_1 = 0$　　带入p_1, p_2　$p_2^T K^{-T} t\hat{}Rk^{-1} p_1$

本质矩阵E, 基础矩阵F, 简化对极约束：

$$E = t\hat{}R, F = K^{-T} E K^{-1}, x_2^T t\hat{}Ex_1 = p_2^T F p_1 = 0$$

对极约束简化给出两个点的空间位置关系，相机位姿估计问题变成一下两部：① 根据匹配点的位置，求出E或者F。②根据E或者F，求出R, t。

★ 视觉里程计—齐次坐标介绍

齐次坐标在电脑图形内无处不在，因为该坐标允许平移、旋转、缩放及透视投影等可表示为矩阵与向量相乘的一般向量运算。依据链式法则，任何此类运算的序列均可相乘为单一个矩阵，从而实现简单且有效之处理。与此相反，若使用笛卡儿坐标，平移及透视投影不能表示成矩阵相乘，虽然其他的运算可以。

优点：许多图形应用涉及到几何变换，主要包括平移、旋转、缩放。以矩阵表达式来计算这些变换时，平移是矩阵相加，旋转和缩放则是矩阵相乘，综合起来可以表示为p' = m1*p+ m2（m1旋转缩放矩阵，m2为平移矩阵， p为原向量 ，p'为变换后的向量）。引入齐次坐标的目的主要是合并矩阵运算中的乘法和加法，表示为p' = p*M的形式。即它提供了用矩阵运算把二维、三维甚至高维空间中的一个点集从一个坐标系变换到另一个坐标系的有效方法。其次，它可以表示无穷远的点。

★ 视觉里程计—齐次坐标介绍

齐次坐标就是用N+1维来代表N维坐标

我们可以在一个2D笛卡尔坐标末尾加上一个额外的变量w来形成2D齐次坐标，因此，一个点(X,Y)在齐次坐标里面变成了（x,y,w），并且有X = x/w，Y = y/w

例如，笛卡尔坐标系下（1，2）的齐次坐标可以表示为（1，2，1），如果点（1，2）移动到无限远处，在笛卡尔坐标下它变为(∞,∞)，然后它的齐次坐标表示为（1，2，0）。

★ 视觉里程计—齐次坐标介绍

我们把齐次坐标转化为笛卡尔坐标的方法是前面n-1个坐标分量分别除以最后一个分量即可。

$$\underset{\text{Homogeneous}}{(x,y,w)} \quad \Leftrightarrow \quad \underset{\text{Cartesian}}{\left(\frac{x}{w},\frac{y}{w}\right)}$$

转化齐次坐标到笛卡尔坐标的过程中，我们有一个发现，例如

$$\begin{aligned}
(1,2,3) &\Rightarrow \left(\frac{1}{3},\frac{2}{3}\right) \\
(2,4,6) &\Rightarrow \left(\frac{2}{6},\frac{4}{6}\right) &&= \left(\frac{1}{3},\frac{2}{3}\right) \\
(4,8,12) &\Rightarrow \left(\frac{4}{12},\frac{8}{12}\right) &&= \left(\frac{1}{3},\frac{2}{3}\right) \\
&\vdots &&\vdots \\
(1a,2a,3a) &\Rightarrow \left(\frac{1a}{3a},\frac{2a}{3a}\right) &&= \left(\frac{1}{3},\frac{2}{3}\right)
\end{aligned}$$

你会发现(1，2，3)，(2，4，6) 和(4，8，12)对应同一个Euclidean point (1/3，2/3)，任何标量的乘积，例如(1a，2a，3a) 对应 笛卡尔空间里面的(1/3，2/3) 。因此，这些点是“齐次的”，因为他们代表了笛卡尔坐标系里面的同一个点。换句话说，齐次坐标有规模不变性。

★ 视觉里程计—齐次坐标介绍

证明两条直线可以相交：考虑如下方程组：

$$\begin{cases} Ax + By + C = 0 \\ Ax + By + D = 0 \end{cases}$$

我们知道在笛卡尔坐标系里面，该方程组无解，因为C ≠ D,如果C=D,两条直线就相同了。

让我们在透视空间里面，用齐次坐标x/w，y/w代替x ,y,

$$\begin{cases} A\frac{x}{w} + B\frac{y}{w} + C = 0 \\ A\frac{x}{w} + B\frac{y}{w} + D = 0 \end{cases} \Rightarrow \begin{cases} Ax + By + Cw = 0 \\ Ax + By + Dw = 0 \end{cases}$$

现在我们有一个解(x，y，0)，两条直线相交于(x，y，0)，这个点在无穷远处。

★ 视觉里程计

3D-2D模型：

PnP(Perspective-n-Point)是求解 3D 到 2D 点对运动的方法。最少只需三个点对(需要至少一个额外点验证结果)就可以估计相机运动。PnP 问题有很多种求解方法,例如用三对点估计位姿的P3P,直接线性变换(DLT),EPnP(Efficient PnP),UPnP等等)。此外,还能用非线性优化的方式,构建最小二乘问题并迭代求解,Bundle Adjustment。

3D-3D模型：

迭代最近点ICP（ Iterative Closest Point） ，指代匹配好的两组点云间运动估计问题。求解3D坐标到3D坐标的转换矩阵

★ 后端优化与建图

视觉里程计求解的相机变换仅为粗略估计，随着时间的推移，误差会逐渐累积。因此，仅由视觉里程计得到的地图误差较大，需采取优化方法优化地图，保证长时间内地图的准确性。后端优化是SLAM中非常重要的一部分，可以减少建图过程中产生的累积误差。VSLAM的优化方法主要分为两种：**基于滤波的方法**和**基于图优化**的方法。

★ 后端优化与建图：

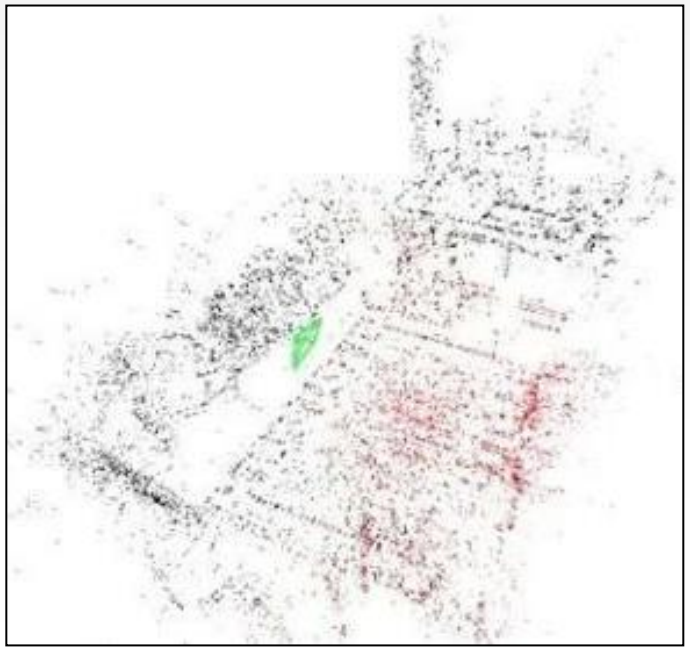
(1)EKF:核心思想是以概率表示状态不确定性。卡尔曼滤波是一种高效率的递归滤波器，它能够从一系列的不完全包含噪声的测量中估计动态系统的状态。EKF为扩展卡尔曼滤波器，用于解决非线性问题。该方法是通过前一个状态估计当前状态。由于只更新当前状态，在大场景下长时间运行时，累积误差逐渐增大，因此地图的精度降低。

(2)图优化：图优化本质上是一个优化问题。图是由顶点和边组成的结构，这里顶点可以是相机的位姿和地图中的特征点，边表示的是顶点之间的关系，在VSLAM中是相机对特征点的观测，以及相机位姿的估计。将所有顶点和边组合在一起可以构建需要优化的目标函数，优化变量为相机的位姿。图优化考虑了整个地图的顶点和边，计算量较大，但构建的地图精度较高。同时，出现一些简化计算的方法，使得图优化成为VSLAM中主流的优化方法

★ 后端优化与建图

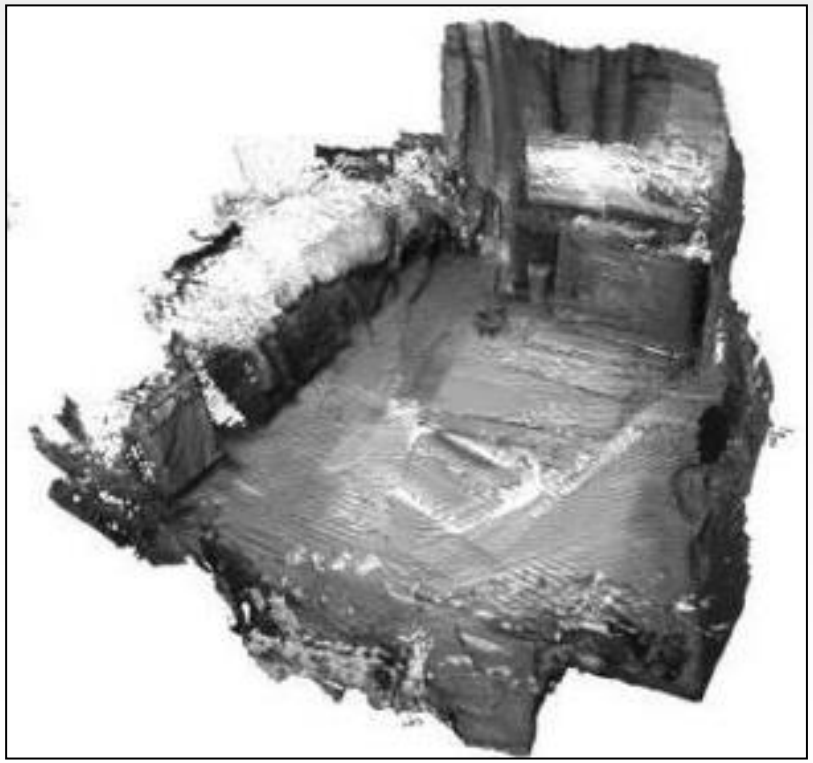
地图是SLAM系统的一个重要的输出，它是机器人利用传感器采集到的数据对周围位置环境的建模。VSLAM大致可以得到以下3种地图形式。

(1)稀疏地图：稀疏地图只建模感兴趣的部分，也就是地图中的特征点。稀疏地图的运行速度很快，节约内存。但是，稀疏地图只能用于机器人的自我定位，无法完成更高层次的任务。稀疏地图如图所示。



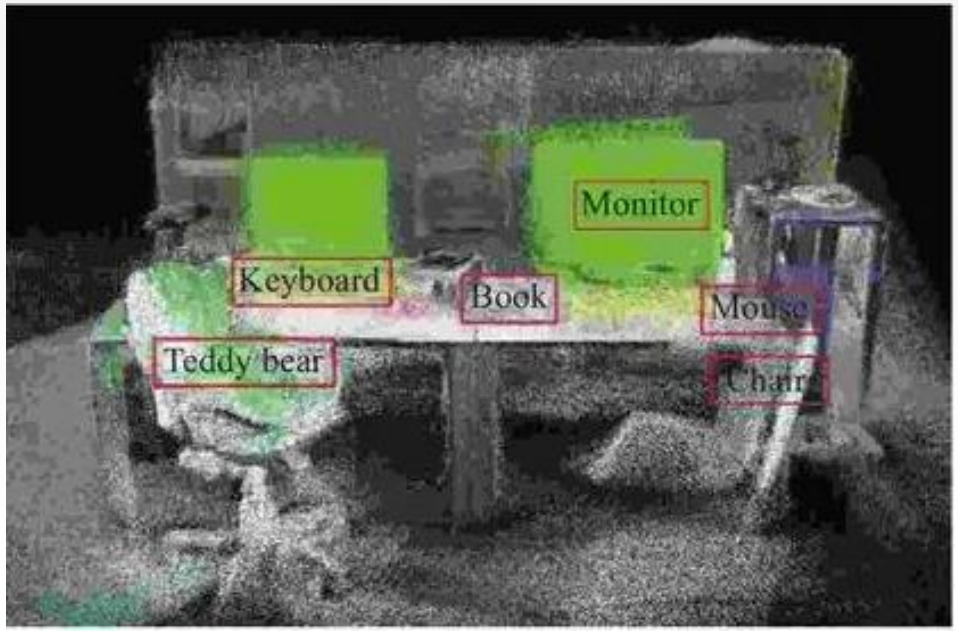
★ 后端优化与建图

(2)稠密地图：稠密地图是指建模所有经过的场景。对于同一个物体,稀疏地图可能只建模物体的角点,而稠密地图会建模整个物体。稠密地图可以用于导航、避障和三维重建等任务。稠密地图如图所示。



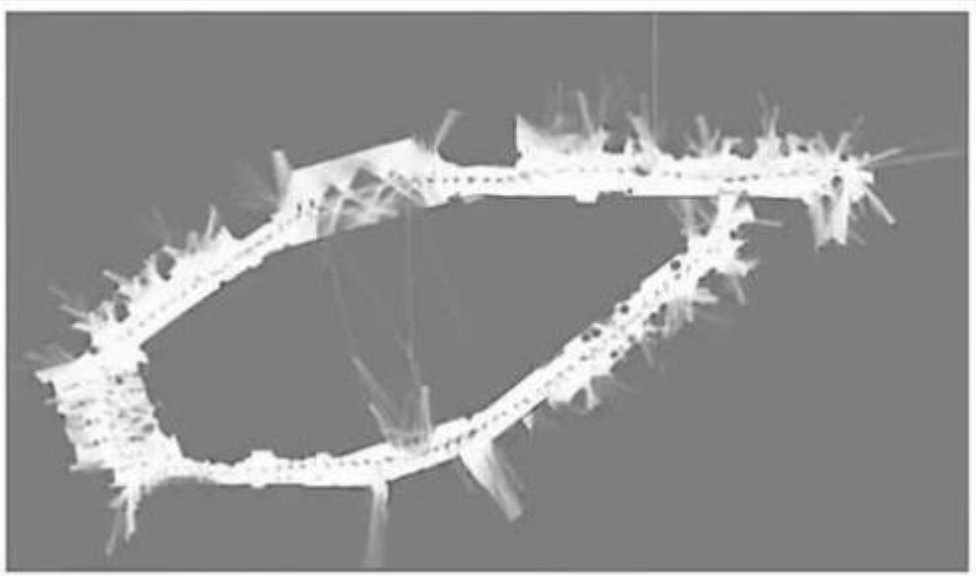
★ 后端优化与建图

(3)语义地图：现在，VSLAM的一个重要的研究方向就是与深度学习技术相结合，建立周围环境的带物体标签的语义地图。语义地图的建立不仅能够获取周围环境的几何结构信息，还可以识别环境中的各个物体，获取其类别和功能属性等语义信息。语义地图的建立能够让机器人真正感知这个世界，有更高层次的认知。语义地图如图所示。

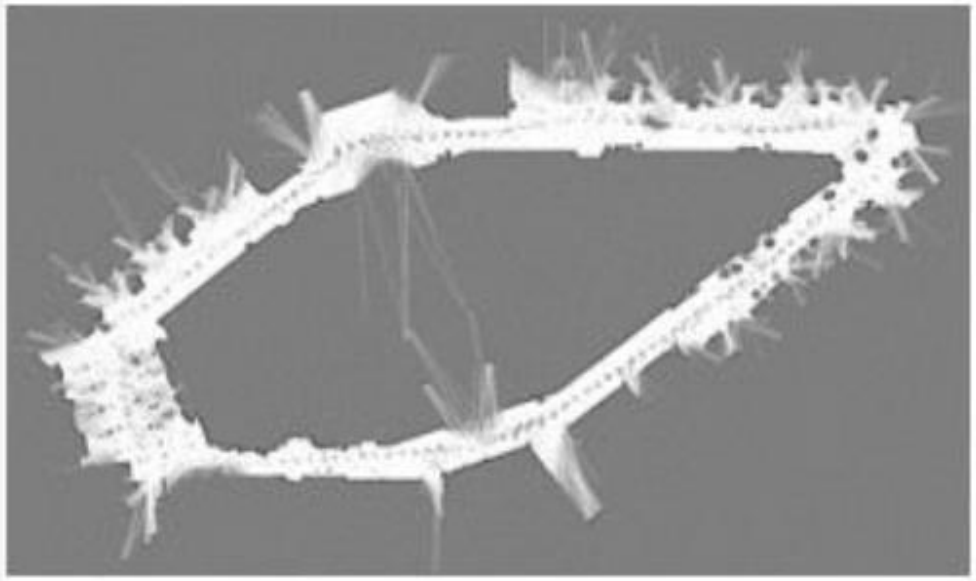


★ 回环检测与词袋模型

回环检测指机器人能够识别曾经到过的地方，使地图形成回环。回环检测的作用是增加与以前机器人位姿之间的约束，提高定位和建图的精度。没有加入回环检测和加入回环检测后的建图效果对比如图所示。可以看出，加入回环检测后，机器人识别出曾经到达的场景，地图精度得到保障。



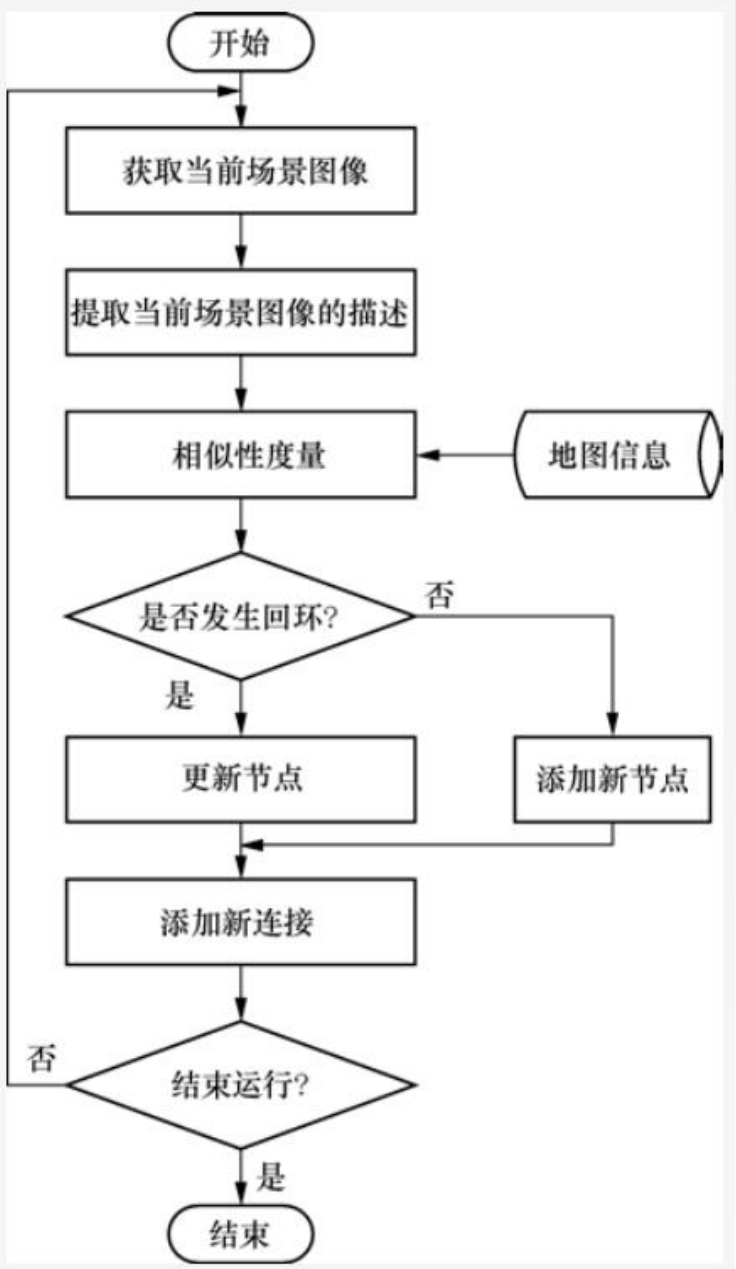
(a) 没有回环检测



(b) 加入回环检测

★ 回环检测与词袋模型

回环检测是VSLAM的重要组成部分。如前所述，后端会对视觉里程计计算出的相机位姿进行优化。然而，由于传感器和算法精度的问题，还是不可避免地会产生累积误差。如果没有回环检测，VSLAM在长时间和大范围的情况下会出现严重的偏差，从而无法产生全局一致的轨迹和地图。回环检测可以为后端提供长时间段内的约束，以此消除或减小累积误差，对机器人实时更新地图和避免引入错误节点起着关键作用。回环检测算法的流程图如图所示。



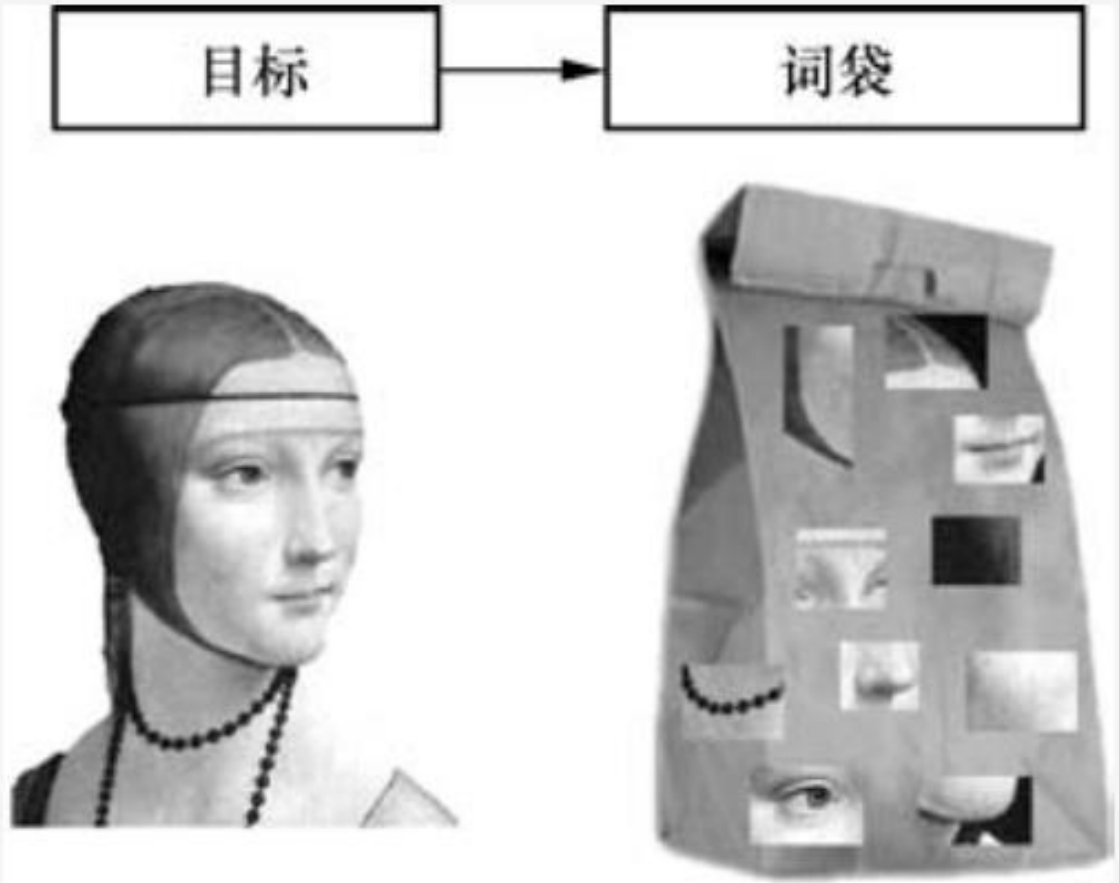
★ 回环检测与词袋模型

最简单的回环检测的方法是特征匹配的方法，通过比较两幅图像中相同特征的个数，判断两幅图像之间是否形成回环。然而，特征匹配本身就非常耗时，回环检测需要与过去所有的关键帧进行特征比较，需要消耗大量的时间。所以，特征匹配的方法的运算量大，回环检测时间长。为了减少回环检测的时间，随机抽取历史关键帧与当前帧进行比较。虽然随机抽取关键帧的方法减少了计算量，但是随机抽取的方法没有针对性，随着关键帧的数量越来越多，能够随机抽取到回环的概率会越来越小，导致检测效率低，回环检测失败。因此，研究者提出词袋（bag of words）模型，加速特征匹配，提高回环检测的效率。

★ 回环检测与词袋模型

词袋的目的是用“图像上有哪几种特征”描述一幅图像。根据这样的描述，可以度量这两幅图像的相似性。在目前流行的VSLAM中，词袋模型是回环检测的主流做法。使用词袋模型分为以下几个步骤：确定单词的概念，许多单词放在一起组成字典，确定一幅图像中出现哪些在字典中定义的概念，用单词出现的情况描述整幅图像。这就把一幅图像转换成了一个向量的描述，该向量描述的是“图像是否含有某类特征”的信息，比单纯的灰度值更稳定。又因为描述向量代表的是“是否出现”，所以不管它们在哪里出现，都与物体的空间位置和排列顺序无关，因此，在相机发生少量运动时，只要物体仍在视野中出现，就认为描述向量不发生变化。最后，比较上一步中的向量描述的相似程度。

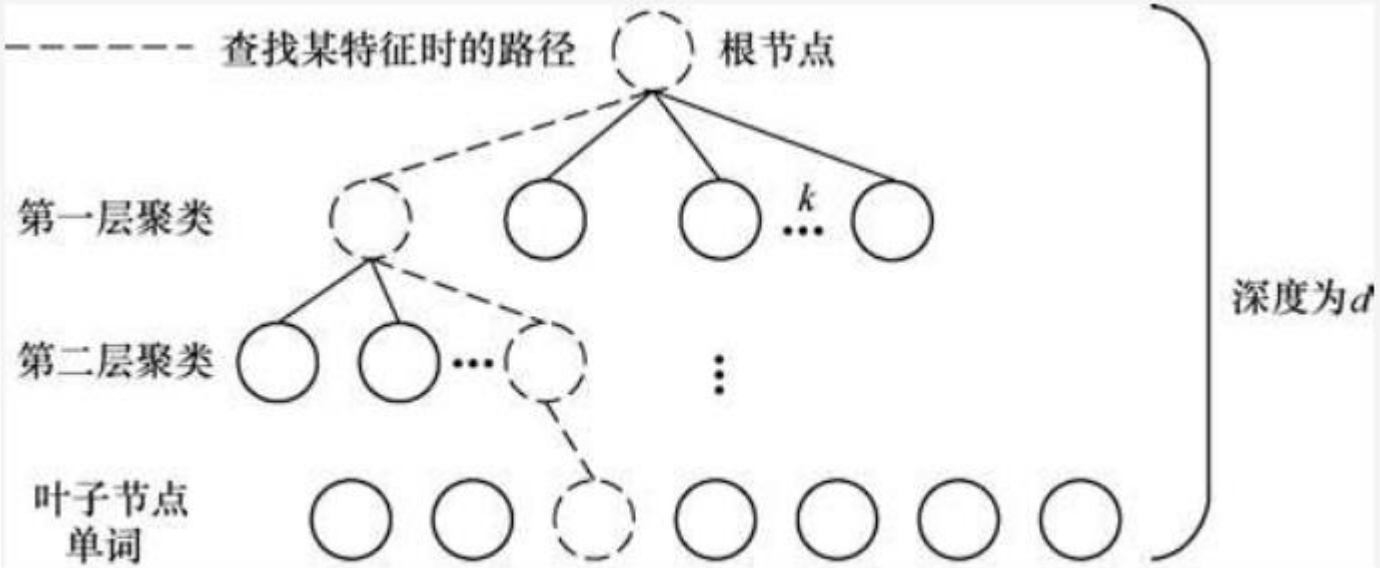
★ 回环检测与词袋模型



词袋模型示意图

★ 回环检测与词袋模型

词袋模型中的字典由很多单词组成，而每个单词代表一个概念。一个单词与一个单独的特征点不同，它不是从单个图像上提取出来的，而是某一类特征的组合。所以，字典生成问题类似于一个聚类问题。聚类问题是无监督机器学习中一个特别常见的问题，用于让机器自行寻找数据中的规律。词袋模型的字典生成问题也属于其中之一。对大量的图像提取特征点之后，采用一种k叉树表达字典，如图所示



k叉树字典示意图

★ 回环检测与词袋模型

整体思路，类似于层次聚类，是对K-means方法的深化。假定有N个特征点，希望构建一个深度为d,每次分叉为k的树，训练字典时，逐层使用K-means方法聚类。根据已知特征查找单词时，也可逐层比对，找到对应的单词。其步骤如下：

(1)在根节点，用K-means方法把所有样本聚成k类，这样便得到第一层。

(2)对第一层的每个节点，把属于该节点的样本再用K-means方法聚成k类，得到下一层。

(3)依次往下推，最后得到叶子节点。叶子层即所谓的Words。

实际上，最终在叶子层构造了Word,在快速查找时使用，用到树的中间节点。这样一个k分支，深度为d的树，可以容纳k个单词。另外，在查找某个给定特征对应的单词时，用此方法只需将它与每层聚类中心比较，总共只需要比较d次，就能够快速找到最后的Word,这样查找效率就能大大提升。

★ 回环检测与词袋模型

有了字典之后，给定任意特征f,只要在字典树中逐层查找，最后就能找到与之对应的单词w。假设从一幅图像中提取了N个特征，找到这N个特征对应的单词之后，相当于拥有了该图像在单词列表中的分布。考虑到不同的单词在区分性上的重要性并不相同，因此希望对单词的区分性或重要性加以评估，给它们不同的权值，以获得更好的效果。在文本检索中，常用的一种做法称为TF-IDF（term frequency-inverse document frequency）译词频-逆文档词频。TF指的是图像中出现频率越高的部分区分度越高，IDF指的是在词典中出现的频率越低区分度越高。

★ 回环检测与词袋模型

当建立词典时考虑IDF，该Word的IDF为：

$$\mathrm{IDF}_i = \log \frac{n}{n_i}$$

n为所有特征数量，n_i为叶子节点中特征点的数量。

TF是指某个特征在单幅图像中出现的频率。TF为：

$$\mathrm{TF}_i = \frac{n_i}{n}$$

n为单幅图像中单词出现的总次数，n_i为单词w出现的次数。

权重等于TF乘IDF之积：

$$\eta_i = \mathrm{TF}_i \times \mathrm{IDF}_i$$

考虑权重以后，对于图像A，它的特征点可对应到许多单词，组成它的词袋：

$$\mathrm{A} = \{(w_1, \eta_1), (w_2, \eta_2), \cdots, (w_N, \eta_N)\} = \boldsymbol{\nu}_\mathrm{A}$$

★ 回环检测与词袋模型

通过词袋，可以用单个向量描述一幅图像A。这个向量以是一个稀疏向量，它的非零部分表示图像A中含有哪些单词，这些部分的值为TF-IDF的值。给定两幅图片A和B,可以得到描述向量A和B,然后通过L范数计算这两幅图像之间的相似性，如式所示。

$$s(\boldsymbol{v}_{\mathrm{A}}-\boldsymbol{v}_{\mathrm{B}})=2\sum_{i=1}^{N}\left|\boldsymbol{v}_{\mathrm{A}i}\right|+\left|\boldsymbol{v}_{\mathrm{B}i}\right|-\left|\boldsymbol{v}_{\mathrm{A}i}-\boldsymbol{v}_{\mathrm{B}i}\right|$$

单词与特征点不同，单词不是只从一幅图像上提取出来的，而是对某一类特征的总结，属于更高一级的特征。利用大量数据训练字典，可以提高字典的准确性，从而提高回环检测的准确率。