

SLAM14讲—12回环检测

12.1回环检测概述

12.1.1回环检测的意义

回环检测模块，能够给出除了相邻帧之外的，一些时隔更加久远的约束。我们察觉到相机经过了同一个地方，采集到了相似的数据。而回环检测的关键，就是如何有效地检测出相机经过同一个地方这件事。如果能够成功地检测这件事，既可以为后端的Pose Graph提供更多的有效数据，使之得到更好的估计，特别是得到一个全局一致的估计。

- 回环检测的意义
 - 关系到我们估计的轨迹和地图在长时间下的正确性
 - 在跟踪算法丢失之后，我们还可以利用回环检测进行重定位

12.1.2方法

- 基于里程计的几何关系
当我们发现当前相机运动到了之前的某个位置附近时，检测它们有没有回环关系。但是由于累计误差的存在，我们往往没法正确地发现“运动到了之前的某个位置附近”这件事实。故这种方法逻辑上存在搁蹄。
- 基于外观
仅根据两张图像的相似性确定回环检测关系。核心问题在于如何计算图像间的相似性。

12.1.3准确率和召回率（怎样的函数s(A,B)可以反映图像间的相似性）

程序可能检测出的四种情况：

表 12-1 回环检测的结果分类

算法 \ 事实	是回环	不是回环
是回环	真阳性（True Positive）	假阳性（False Positive）
不是回环	假阴性（False Negative）	真阴性（True Negative）



图 12-2 假阳性与假阴性的例子。左侧为假阳性，两个图像看起来很像，但并非同一个走廊；右侧为假阴性，由于光照变化，同一个地方不同时刻的照片看起来很不一样。

- 准确率和召回
准确率描述的是，算法提取的所有回环中，确实是真实回环的概率。

召回率描述的是，在所有真实回环中，被正确检测出来的概率。

$$\text{Precision} = TP / (TP + FP), \quad \text{Recall} = TP / (TP + FN). \quad (12.2)$$

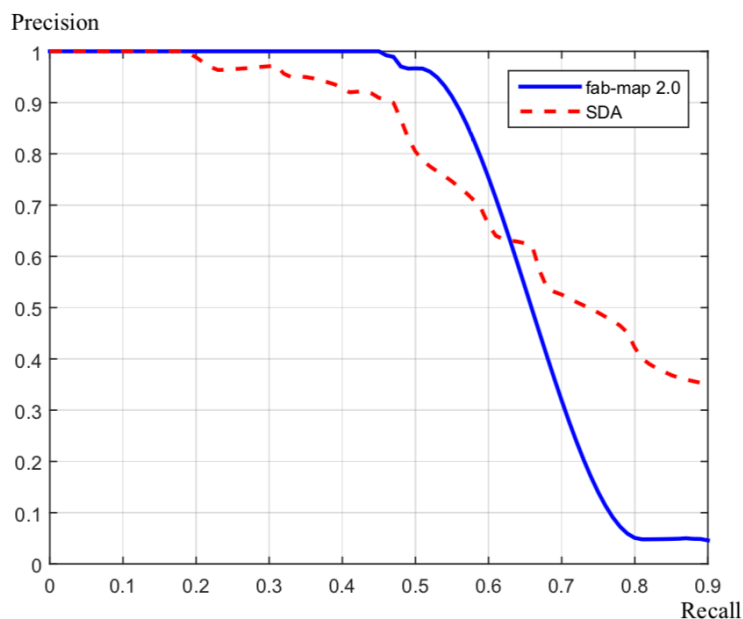


图 12-3 准确率-召回率曲线的例子 [91]。随着召回率的上升，检测条件变得宽松，准确率随之下降。好的算法在较高召回率情况下仍能保证较好的准确率。

为了评价算法的好坏，我们会测试它在各种配置下的 P 和 R 值，然后做出一条 Precision-Recall 曲线。当用召回率为横轴，用准确率为纵轴时，我们会关心整条曲线偏向右上方的程度、100% 准确率下的召回率，或者 50% 召回率时候的准确率，作为评价算法的指标。不过请注意，除去一些“天壤之别”的算法，我们通常不能一概而论算法 A 就是优于算法 B 的。我们可能说 A 在准确率较高时还有很好的召回率，而 B 在 70% 召回率的情况下还能保证较好的准确率，诸如此类的评价。

值得一提的是，在 SLAM 中，我们对准确率要求更高，而对召回率则相对宽容一些。由于假阳性的（检测结果是而实际不是的）回环将在后端的 Pose Graph 中添加根本错误的边，有些时候会导致优化算法给出完全错误的结果。想象一下，如果 SLAM 程序错误地将所有的办公桌当成了同一张，那建出来的图会怎么样呢？——你可能会看到走廊不直了，墙壁被交错在一起了，最后整个地图都失效了。而相比之下，召回率低一些，则顶多有部分的回环没有被检测到，地图可能受一些累积误差的影响——然而仅需一两次回环就可以完全消除它们了。所以说在选择回环检测算法时，我们更倾向于把参数设置地更严格一些，或者在检测之后再加上回环验证的步骤。

12.2 词袋模型

词袋，Bag-of-Words，目的是用“图像上有哪几种特征”来描述一个图像。具体步骤如下：

1. 确定概念——对应于BoW中的“单词”，许多单词放在一起，组成了“字典”
2. 确定一张图像中，出现了哪些在字典中定义的概念——我们用单词出现额情况（或直方图）描述整张图像。
把一个图像转换成了一个向量的描述。
3. 比较上一步中的描述的相似程度。

以上面举的例子来说，首先我们通过某种方式，得到了一本“字典”。字典上记录了许多单词，每个单词都有一定意义，例如“人”、“车”、“狗”都是记录在字典中的单词，我们不妨记为 w_1, w_2, w_3 。然后，对于任意图像 A ，根据它们含有的单词，可记为：

$$A = 1 \cdot w_1 + 1 \cdot w_2 + 0 \cdot w_3. \quad (12.3)$$

字典是固定的，所以只要用 $[1, 1, 0]^T$ 这个向量就可以表达 A 的意义。通过字典和单词，只需一个向量就可以描述整张图像了。该向量描述的是“图像是否含有某类特征”的信息，比单纯的灰度值更加稳定。又因为描述向量说的是“是否出现”，而不管它们“在哪儿出现”，所以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，我们就仍然保证描述向量不发生变化。^① 基于这种特性，我们称它为 Bag-of-Words 而不是什么 List-of-Words，强调的是 Words 的有无，而无关其顺序。因此，可以说字典类似于单词的一个集合。

回到上面的例子，同理，用 $[2, 0, 1]^T$ 可以描述图像 B 。如果只考虑“是否出现”而不考虑数量的话，也可以是 $[1, 0, 1]^T$ ，这时候这个向量就是二值的。于是，根据这两个向量，设计一定的计算方式，就能确定图像间的相似性了。当然如果对两个向量求差仍然有一些不同的做法，比如说对于 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^W$ ，可以计算：

$$s(\mathbf{a}, \mathbf{b}) = 1 - \frac{1}{W} \|\mathbf{a} - \mathbf{b}\|_1. \quad (12.4)$$

其中范数取 L_1 范数，即各元素绝对值之和。请注意在两个向量完全一样时，我们将得到 1；完全相反时（ \mathbf{a} 为 0 的地方 \mathbf{b} 为 1）得到 0。这样就定义了两个描述向量的相似性，也就定义了图像之间的相似程度。

12.3 字典

12.3.1 字典的结构

字典生成问题类似于一个聚类的问题，我们可以用 K-means 算法解决。

1. 随机选取 k 个中心点： c_1, \dots, c_k ；
2. 对每一个样本，计算与每个中心点之间的距离，取最小的作为它的归类；
3. 重新计算每个类的中心点。
4. 如果每个中心点都变化很小，则算法收敛，退出；否则返回 1。

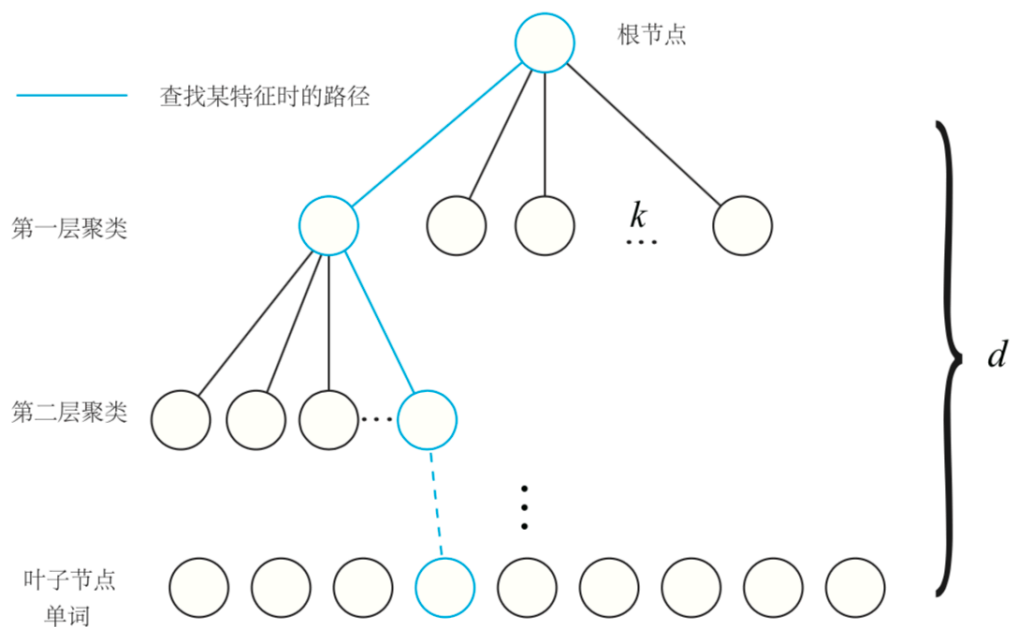


图 12-4 K 叉树字典示意图。训练字典时，逐层使用 K-means 聚类。根据已知特征查找单词时，亦可逐层比对，找到对应的单词。

在 [98] 中，使用一种 k 叉树来表达字典。它的思路很简单，类似于层次聚类，是 k-means 的直接扩展。假定我们有 N 个特征点，希望构建一个深度为 d ，每次分叉为 k 的树，那么做法如下（见图 12.3.1）^①：

1. 在根节点，用 k-means 把所有样本聚成 k 类（实际中为保证聚类均匀性会使用 k-means++）。这样得到了第一层。
2. 对第一层的每个节点，把属于该节点的样本再聚成 k 类，得到下一层。
3. 依此类推，最后得到叶子层。叶子层即为所谓的 Words。

12.3.2 实践：创建字典（未实践）

12.4 相似度计算

12.4.1 理论部分

我们对单词的区分性或重要性进行评估，给它们不同的权值以起到更好地效果。

- TF-IDF 方法

TF 部分的思想：某单词在一个图像中经常出现，它的区分度就越高。

IDF 部分的思想：某单词在字典中出现的频率月底，则分类图像时区分度越高。

我们在词袋模型中，建立字典时可以考虑 IDF 部分。我们统计某个叶子节点 w_i 中的特征数量相对于所有特征数量的比例，作为 IDF 部分。假设所有特征数量为 n ， w_i 数量为 n_i ，那么该单词的 IDF 为：

$$\text{IDF}_i = \log \frac{n}{n_i}. \quad (12.5)$$

另一方面，TF 部分则是指某个特征在单个图像中出现的频率。假设图像 A 中，单词 w_i 出现了 n_i 次，而一共出现的单词次数为 n ，那么 TF 为：

$$\text{TF}_i = \frac{n_i}{n}. \quad (12.6)$$

于是 w_i 的权重等于 TF 乘 IDF 之积：

$$\eta_i = \text{TF}_i \times \text{IDF}_i. \quad (12.7)$$

考虑权重以后，对于某个图像 A ，它的特征点可对应到许多个单词，组成它的 Bag-of-Words：

$$A = \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} \triangleq \mathbf{v}_A. \quad (12.8)$$

由于相似的特征可能落到同一个类中，因此实际的 \mathbf{v}_A 中会存在大量的零。无论如何，通过词袋，我们用单个向量 \mathbf{v}_A 描述了一个图像 A 。这个向量 \mathbf{v}_A 是一个稀疏的向量，它的非零部分指示出图像 A 中含有哪些单词，而这些部分的值为 TF-IDF 的值。

- 若给定 \mathbf{v}_A 和 \mathbf{v}_B ，如何计它们的差异？

$$s(\mathbf{v}_A - \mathbf{v}_B) = 2 \sum_{i=1}^N |\mathbf{v}_{Ai}| + |\mathbf{v}_{Bi}| - |\mathbf{v}_{Ai} - \mathbf{v}_{Bi}|. \quad (12.9)$$

12.4.2 实践：相似度的计算（未实践）

12.5 实验分析与评述