

# 回环检测

---

## 12.1 回环检测概述

---

### 12.1.1 回环检测的意义

SLAM长期运行会产生**累计误差**，回环检测就是用来消除累计误差的方法。**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情**。若检测到，则将这时候的位姿拉到上一次经过这个地方的位姿（消除误差）。

### 12.1.2 方法

### 12.1.3 准确率核召回率

## 12.2 词袋模型

---

词袋(Bag-of-Words , BoW)，目的是用“图像上有哪几种特征”来描述一个图像。我们要做的如下，举例：

1. 假设一幅图中有“人，车，狗”等概念（特征点）——对应于BoW中的单词(Word)，许多单词放在一起，组成了**字典**
2. 确定一张图像中，出现了哪些字典中定义的概念——我们用单词出现的情况(或者直方图)描述整张图像。这样就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。

第三步，我们可以构造**评价函数**来比较相似程度，判断向量即可，若a、b向量都是W维的，那么有：

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

## 12.3 字典

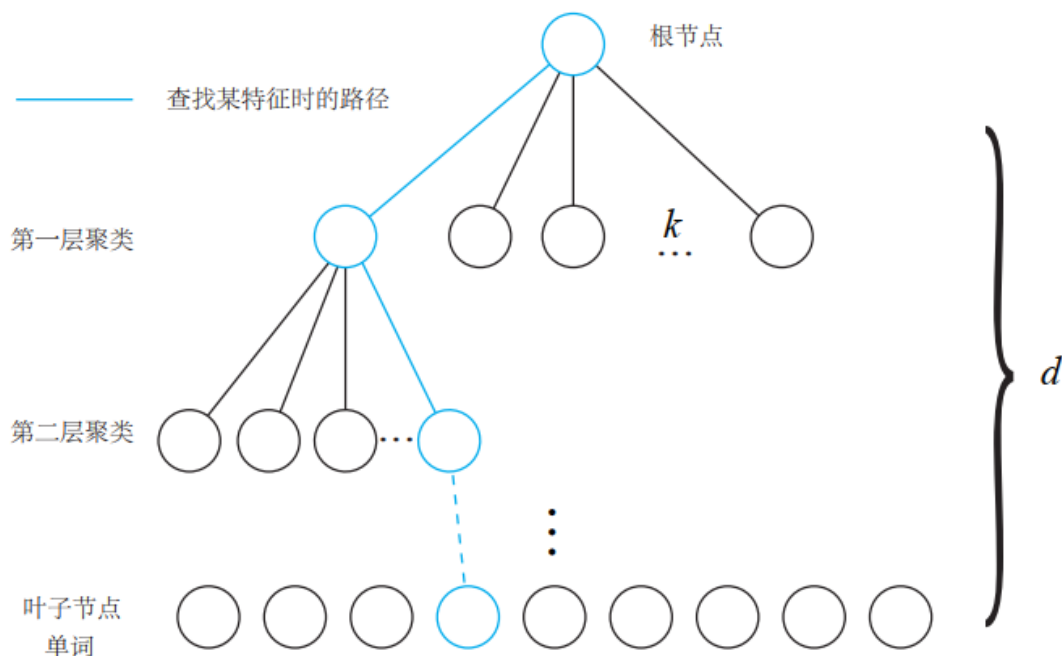
---

### 12.3.1 字典的结构

字典由很多单词组成，每一个单词代表了一个概念。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，**因此，字典生成问题类似于一个聚类(Clustering)问题**。

K-means是一个非常简单有效的方法，在无监督学习中广为使用，介绍在[机器学习(自)]中有。

我们再解决查字典的问题，类似二分查找，“K叉树”字典如图所示：



构建K叉树字典的步骤比较简单：

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。

最终，我们在叶子层构建了单词，而树结构中的中间节点仅供快速查找时使用。一个  $k$  分支，深度为  $d$  的树，可以容纳  $k^d$  个单词。此外，查找某个给定特征对应的单词时，只需将它与每个中间结点的聚类中心比较 ( $d$  次) 即可，对数级别的查找效率。

## 12.4 相似度计算

### 12.4.1 理论部分

我们使用 TF-IDF (Term Frequency-Inverse Document Frequency) 方法。其中 TF 部分的意思是：某单词在一个图像中经常出现，则其区分度就高；IDF 的意思是：某单词在字典中出现的频率越低，则分类图像时区分度越高。

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：

$$IDF_i = \log\left(\frac{n}{n_i}\right)$$

另一方面，假设图像  $A$  中，单词  $w_i$  出现了  $n_i$  次，而一共出现的单词数为  $n$ ，那么 TF 为：

$$TF_i = \frac{n_i}{n}$$

于是， $w_i$  的权重等于 TF 与 IDF 的乘积

$$\eta_i = TF_i \times IDF_i$$

考虑权重以后，对于某个图像  $A$ ，它的特征点可以对应到许多个单词，则组成它的 BoW 为：

$$A = \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} = \overrightarrow{v_A}$$

我们用向量 $v_A$ 描述了一个图像A，这个向量 $v_A$ 是一个稀疏向量。

最后，给定 $v_A$ 和 $v_B$ ，如何计算它们的差异？存在很多解决方法，只提一种：

$$s(v_A - v_B) = 2 \sum_{i=1}^N |v_{Ai}| + |v_{Bi}| - |v_{Ai} - v_{Bi}|$$

## 12.5 实验分析与评述

---