

字典生成工具测试文档

|  |  |
| --- | --- |
| 文档版本 | V 1.0.0 |
| 发布日期 | 2021-03-16 |

**版权所有©舵敏智能2021，保留一切权力。**

非经本公司书面许可，任何单位或个人不得擅自摘抄、复制部分或全部本文档，不得以任何形式传播。

**商标声明**

、舵敏、舵敏智能和其他舵敏商标均为舵敏智能科技的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人所有。

**上海舵敏智能科技有限公司**

**舵敏智能科技（苏州）有限公司**

|  |  |
| --- | --- |
| 网址: | https://www.dominant-tech.com/ |
| 客户服务邮箱: | support@dominant-tech.com |

**目 录**

# 简介

## 项目背景

DBoW2\_tools是一个特征字典生成工具，其基于DBoW2进行开发，DBoW2的主要功能为提取图像数据集中的视觉特征并将其聚类为一个视觉词汇表，该视觉词汇表使用深度为L的K叉树进行表示，树中的每个节点均为使用K-means进行逐层聚类得到，树中的每个叶子节点表示词汇表中的一个单词。

DBoW2\_tools在DBoW2的基础生增加了一些功能，以方便对视频数据的训练和测试，其主要包括：

1. 从单个或多个视频中构建词典
2. 将词表存储为占用空间更小的二进制格式
3. 使用gFlags配置文件进行输入参数的设置
4. 将提取的特征保存到yml文件

## 测试目的

由于现有的DBoW2词袋模型训练的词袋，占用的空间较大，不能满足项目需求，因此需要对DBoW2的字典大小进行调整，以在保证字典性能的条件下降低辞典的大小，减少内存占用。

在DBoW2中对字典大小有直接影响的参数有两个：字典树中节点的支数K和字典树的深度L，其共同决定了最终字典的大小以及最终生成的单词数量（KL）。理论上最终生成的单词数量越多，字典的表征能力越强，字典也越大，这意味着对字典大小的优化，必然会导致字典性能的下降。

因此本次的测试目的是**在保证词表征性能可以满足应用的条件下，寻找能使得词典最小的最优K、L组合。**

## 定义、首字母缩写词和缩略语

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 意义 | 配置文件中的对应变量 | 代码中的变量 |
| K | 词表树的分支数 | K |  |
| L | 此表数的 | L |  |
| 判定窗口/WindowSize | 视频中查询帧前后多帧图像，其作为查询图像的真值，当查询结果中有匹配到该范围内的图像时，认为查询成功 | testWindowSize |  |
| TopM | 每次查询时返回相似度的升序结果，TopM则表示选取前多少个结果进行判断。 | testTop\_M |  |
| 准确率 | TopM中落在判定窗口内的图像数量/TopM | \ | ACCindence |
| 相似度 | 两张图像对应的词袋向量之间的L1\_Norm，用于判断图像之间是否相似 | \ | score |
| 词袋向量 | 图像查询字典得到的包含的单词序列 | \ | \ |
| 采样间隔 | 读取视频时，每隔多少帧进行处理 | trainVideoSampleRate、testVideoSampleRate |  |
| 索引距离 | TopM中落在评定窗口内的图像与查询图像之间有多少中间帧 |  |  |
| 成功率 | 查询成功的图像与总帧数的比值 |  |  |

## 参考资料

<https://github.com/dorian3d/DBow>

<https://github.com/dorian3d/DBoW2>

基于有序视觉词袋模型的图像相似性衡量

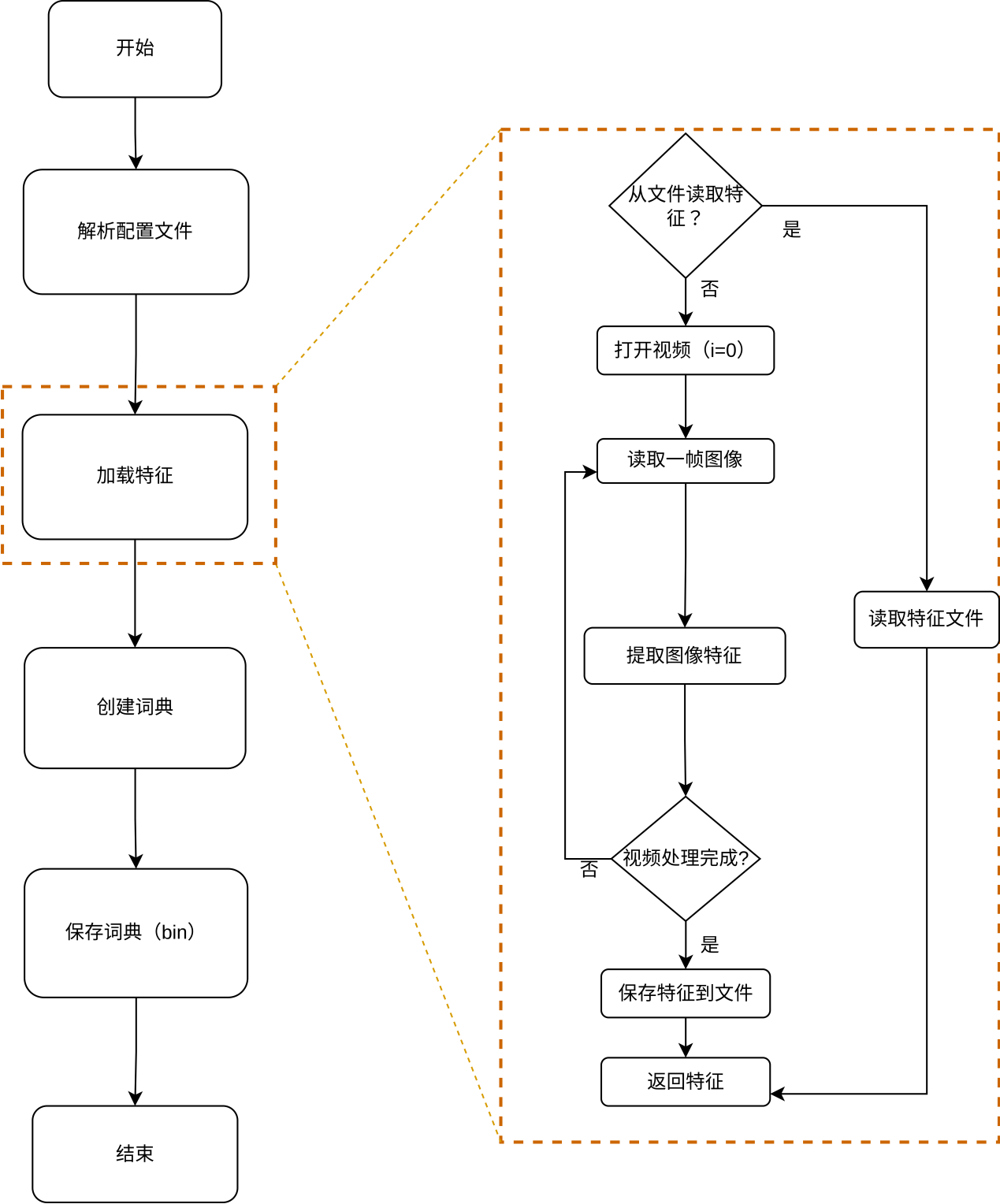
# 测试概要

## 测试方法

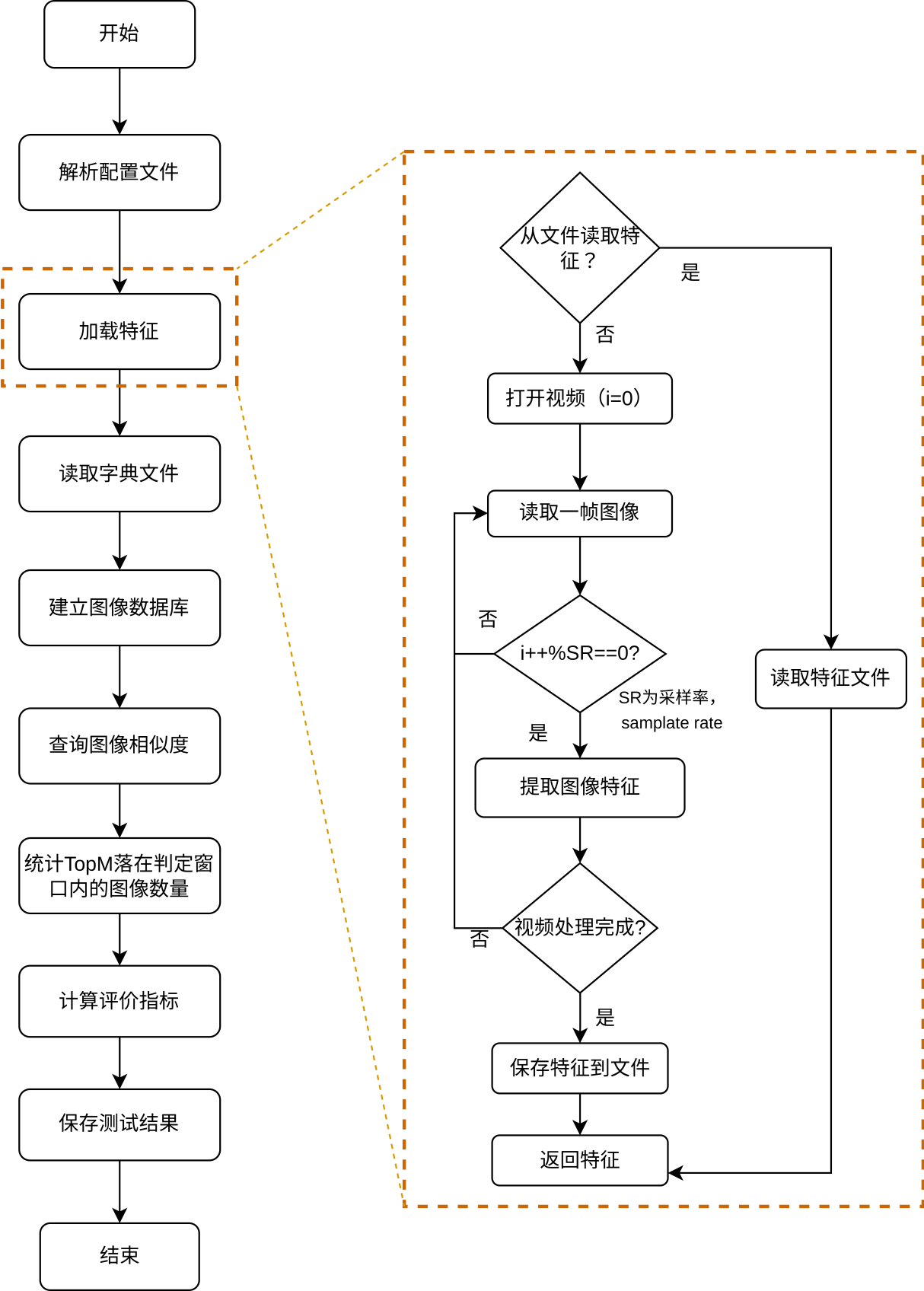
测试过程整体分为字典训练和字典性能测试两个部分。

1. 字典训练的整体流程如2.2所示，每次测试都需要根据不同的条件和参数训练待测试字典。
2. 字典性能测试的整体流程如2.3所示，主要是使用训练的字典测试

## 字典训练流程



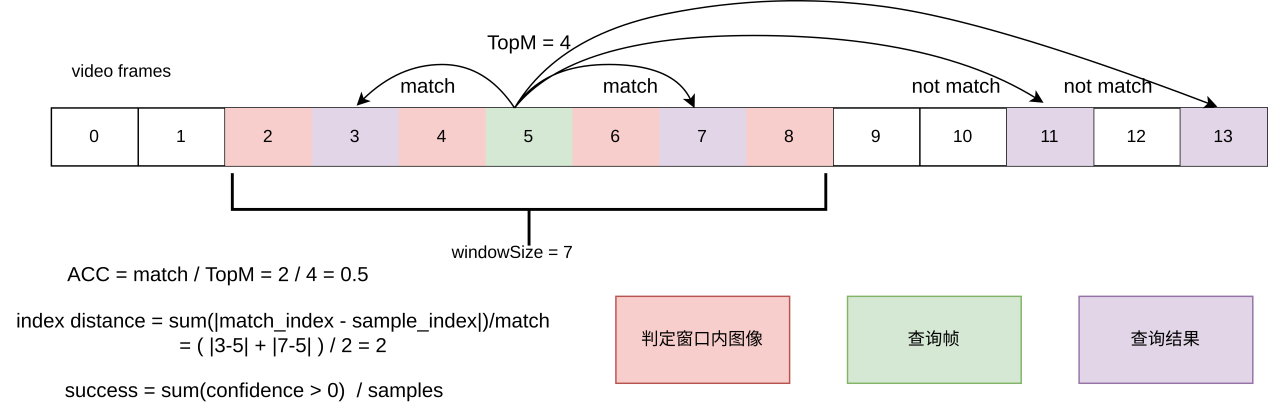
## 字典性能测试流程



## 评价指标

本次测试使用视频作为测试数据，没有参考图像作为真值进行对照，因此本次测试利用视频的连续性，选取查询图像的临近帧作为其真值，其思想是在视频序列中”**与查询帧最相似的帧是它的相邻帧”**。具体的计算方法如下图所示，假定输入测试数据为包含13帧的视频序列，选取其中一帧作为查询帧（图中为5号帧），查询过程中会计算查询帧与其它帧之间的相似度，并选取相似度最高的TopM个图像作为查询结果，当TopM个查询结果中至少有一帧落在查询窗口內时认为查询成功，否则认为查询失败。因此本次测试主要使用了个评价指标：

1. 准确率（ACC），其为单帧图像的TopM个查询结果中落在判定窗口内的图像数量与TopM的比值。
2. 成功率（Success），其为所有视频帧序列中查询成功的图像数量与总的查询帧数的比值。
3. 平均索引距离（avg index distance），索引距离为两张图像在视频序列中相隔的图像帧数量，其表现了两张图像之间在时间序列上的关系。平均索引距离则是TopM个查询结果中落在判定窗口内的图像与查询图像的索引距离之和，再除以TopM个查询结果中落在判定窗口内的图像数量。
4. 平均相似度（avg score），相似度得分score是两张图像对应的词向量之的距离，其根据不同距离度量方法，有不同的计算方式，本次测试均使用L1 Normal作为向量之间的距离度量方式。平均相似度则是TopM个查询结果中落在判定窗口内的图像与查询图像的相似度之和，再除以TopM个查询结果中落在判定窗口内的图像数量。



## K、L取值对应的节点数与字典大小

本节给出了字典大小与K、L之间的对应关系，并本根据字典的叶子节点数进行排序，以方便后续的查早。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | L | load size(Mb) | dict size(Mb) | K |
| 4 | 2 | 0.0057 | 0.0008 | 4 |
| 5 | 2 | 0.0085 | 0.0012 | 5 |
| 6 | 2 | 0.0118 | 0.0017 | 6 |
| 7 | 2 | 0.0157 | 0.0022 | 7 |
| 8 | 2 | 0.0202 | 0.0028 | 8 |
| 4 | 3 | 0.0219 | 0.0033 | 4 |
| 9 | 2 | 0.0253 | 0.0035 | 9 |
| 10 | 2 | 0.0309 | 0.0043 | 10 |
| 11 | 2 | 0.0372 | 0.0052 | 11 |
| 5 | 3 | 0.0411 | 0.0061 | 5 |
| 6 | 3 | 0.0692 | 0.0101 | 6 |
| **4** | **4** | **0.0869** | **0.0133** | **4** |
| 7 | 3 | 0.1080 | 0.0156 | 7 |
| 8 | 3 | 0.1594 | 0.0229 | 8 |
| 5 | 4 | 0.2041 | 0.0305 | 5 |
| 9 | 3 | 0.2249 | 0.0320 | 9 |
| 10 | 3 | 0.3063 | 0.0434 | 10 |
| 4 | 5 | 0.3466 | 0.0534 | 4 |
| 11 | 3 | 0.4055 | 0.0572 | 11 |
| 6 | 4 | 0.4136 | 0.0608 | 6 |
| 7 | 4 | 0.7544 | 0.1095 | 7 |
| 5 | 5 | 1.0195 | 0.1527 | 5 |
| 8 | 4 | 1.2726 | 0.1830 | 8 |
| 4 | 6 | 1.3857 | 0.2135 | 4 |
| 9 | 4 | 2.0214 | 0.2886 | 9 |
| 6 | 5 | 2.4801 | 0.3648 | 6 |
| 10 | 4 | 3.0606 | 0.4344 | 10 |
| 11 | 4 | 4.4572 | 0.6297 | 11 |
| 5 | 6 | 5.0965 | 0.7637 | 5 |
| 7 | 5 | 5.2790 | 0.7667 | 7 |
| 8 | 5 | 10.1789 | 1.4643 | 8 |
| **6** | **6** | **14.8793** | 2.1891 | **6** |
| 9 | 5 | 18.1896 | 2.5974 | 9 |
| 10 | 5 | 30.6027 | 4.3445 | 10 |
| 7 | 6 | 36.9497 | 5.3666 | 7 |
| 11 | 5 | 49.0263 | 6.9269 | 11 |
| **8** | **6** | **81.3584** | 11.7046 | **8** |
| 9 | 6 | 162.9640 | 23.2758 | 9 |
| 10 | 6 | 300.1630 | 42.6449 | 10 |
| 11 | 6 | 509.5410 | 72.1413 | 11 |

# K、L选择与字典性能测试

该部分测试的主要目的是，对不同大小的字典进行性能测试，得到K、L选取的推荐值。

## 单视频字典的性能测试

该部分测试的主要目的是，对不同大小的字典进行性能测试，得到K、L选取的推荐值。

### 测试数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | Size | FPS | Frames | Time |
| 训练 | | | | |
| 2022.07.06@20.55.03/AVMFront.avi | 1280\*718 | 15 | 4420 | 4:54 |
| 测试 | | | | |
| 2022.07.06@21.03.07/AVMFront.av | 1280\*718 | 15 | 2379 | 2:38 |

### 参数设置

红色部分为测试时的变化参数，K、L的不同取值，会产生不同大小的字典。

|  |  |  |
| --- | --- | --- |
| 参数 | 推荐值 | 搜索范围 |
| **K** | **8** | **4~12** |
| **L** | **6** | **2~9** |
| windowSize | 15 | \ |
| TopM | 15 | \ |
| trainVideoSampleRate | 1 | \ |
| testVideoSampleRate | 10 | \ |
|  |  |  |

### 测试结果

### 结果分析

1. 当dict size>0.032Mb，即K=9、L=3时，index开始稳定
2. 在dict size> 0.1095(Mb)，即K=7、L=4时，成功率、准确率、索引距离开始稳定。
3. 随着字典大小的增大，score逐步下降。
4. K=11，L=3时，准确率开始稳定

### 结论

当K=7、L=4时，成功率和准确率都能得到较好的结果，此时字典大小为 0.1095(Mb)。

## 测试视频使用不同采样间隔的测试

BoW主要的功能是相似图像检索，由于其高效的搜索速度、较高的准确率和场景适应性而被使用。在ORB\_SLAM2中BoW的主要应用于重定位和回环检测两个地方。

在重定位中ORB\_SLAM2使用BoW计算当前帧与数据库中关键帧的相似度，将符合条件的关键帧作为当前帧的参考帧，用于计算相机位姿的初步估计，然后再在初步估计的位姿上进行优化，实现最终的相机位姿跟踪。

在回环检测中ORB\_SLAM2使用BoW在关键帧数据库中查找与当前帧不相邻且相似度较高的关键帧作为回环检测的候选帧。

在重定位和回环检测过程中ORB\_SLAM2都是在关键帧数据库中进行查找，但是ORB\_SLAM2的关键帧选取不是连续的，其主要有三个条件：

1. 距离上次插入很远(MaxFrame=fps)
2. 距离上次插入有了一定时间（MinFrames=0）&& LocalMaper空闲
3. 跟踪效果很差（当前帧看到的地图点数量<0.25\*参考帧看到的地图点数量 && 当前帧至少看到了15个地图点）

从ORB\_SLAM2的参考帧插入条件可以看出，ORB\_SLAM2的关键帧是稀疏的，他是选取视频中信息丰富和有一定间隔的帧作为关键帧。因此本次测试主要是通过对测试视频使用不同的采样间隔进行测试，以模拟ORB\_SLAM2中关键帧的稀疏插入，从而判断字典在不同的采样间隔的测试视频中，其性能是否会受到影响。

### 测试数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | Size | FPS | Frames | Time |
| 训练 | | | | |
| 2022.07.06@20.55.03/AVMFront.avi | 1280\*718 | 15 | 4420 | 4:54 |
| 测试 | | | | |
| 2022.07.06@21.03.07/AVMFront.av | 1280\*718 | 15 | 2379 | 2:38 |

### 参数设置

红色部分为变化参数，黑色部分为固定的参数

|  |  |  |
| --- | --- | --- |
| 参数 | 推荐值 | 搜索范围 |
| **K** | **8** | **4~12** |
| **L** | **6** | **2~9** |
| windowSize | 15 | \ |
| TopM | 15 | \ |
| trainVideoSampleRate | 1 | \ |
| **testVideoSampleRate** | **10** | **[1,3,5,8]** |
|  |  |  |

### 测试结果

### 结果分析

1. 通过成功率的图可以看出，当视频采样间隔为1，3，5时，最终的成功率相差不大，但是当视频的采样间隔为8时，最终测试的成功率有一定的下降。
2. 通过准确率的图可以看出，视频采样间隔越高，测试的准确率越低。即查询结果的TopM中有更少的图像落在判定窗口内。
3. 通过对score图可以看出，视频采样间隔越高，其score的分数越低。
4. 通过index的图可以看出，视频采样间隔对index没有确定的影响，但是当采样间隔1时，其index最大；当采样间隔为5时，其index最小。（index越小，其查询结果离查询图像越近）

### 结论

采样间隔的不同会对字典评测结果产生影响，但是这种影响主要体现在准曲率上，对score的影响不大，但是当采样间隔大于等于8时，会导致测试成功率的整体下降。

同时不同采样间隔也会对字典最佳性能的有一定的影响，主要体现在当测试视频的采样间隔较低时，能在字典较小时就能达到最好的成功率，根据成功率得出：

SR=1时，K、L的搜索区间为[7^3,8^6]

SR=3时，K、L的搜索区间为[8^3,8^6]

SR=5时，K、L的搜索区间为[9^3,8^6]

SR=10时，K、L的搜索区间为[7^4,8^6]

根据准确率得出：

K、L的搜索区间为[11^3,8^6]

### 结论

1. **使用本地数据扩充训练数据集的效果优于使用其他数据集进行扩充。**

## 测试总结

通过对比各个测试的结果，我们根据成功率曲线大致得到了两个推荐取值K=4、L=4和K=7、L=4。K=4、L=4对应的是在保证成功率的条件下字典大小能取的最小值，对应了K、L的取值下限，但它是不稳定的，当测试条件改变时，其成功率的波动比较严重；K=7，L=4是在考虑测试视频采样间隔和字典的训练数据量的条件下成功率比较稳定的值，其是相对保守或稳定的推荐值。在具体的应用场景中可以根据实际的情况和参数在[4^4,7^4]范围内做更精确的选择。

另外K=7、L=4准确率曲线和成功率曲线都能取得很高的结果；当K=4、L=4时，成功率较高，但是准确率往往较低。综合考成功率和准确率时，8^3或5^4是一个较好的选择。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| index | K | L | K^L | dictSize(Mb) |
| 11 | **4** | **4** | **256** | **0.0133** |
| 12 | 7 | 3 | 343 | 0.0156 |
| 13 | 8 | 3 | 512 | 0.0229 |
| 14 | 5 | 4 | 625 | 0.0305 |
| 15 | 9 | 3 | 729 | 0.032 |
| 16 | 10 | 3 | 1000 | 0.0434 |
| 17 | 4 | 5 | 1024 | 0.0534 |
| 18 | 6 | 4 | 1296 | 0.0608 |
| 19 | 11 | 3 | 1331 | 0.0572 |
| 20 | **7** | **4** | **2401** | **0.1095** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 条件因素 | | 推荐取值 | | 节点数 | 字典大小  （Mb） | 成功率 | 准确率 |
| 训练数据量  （帧） | 采样间隔  （SR） | K | L |
| 4420 | 1 | 7 | 3 | 343 | 0.0156 | 0.9947 | 0.2953 |
| 3 | 8 | 3 | 512 | 0.0229 | 1.0 | 0.2247 |
| 5 | 9 | 3 | 729 | 0.032 | 0.9758 | 0.1919 |
| 8 | 6 | 4 | 4096 | 0.183 | 0.96 | 0.16 |
| 10 | **7** | **4** | **2401** | **0.1095** | **0.95** | **0.16** |
| 4420\*5 | 1 | **4** | **4** | **256** | **0.0133** | **0.993** | **0.298** |

## 字典的训练数据量和数据多样性测试

本次测试的主要目的是测试训练数据量对字典性能的影响。

### 评价指标

由于原来成功率和准确率的评价指存在一些问题：

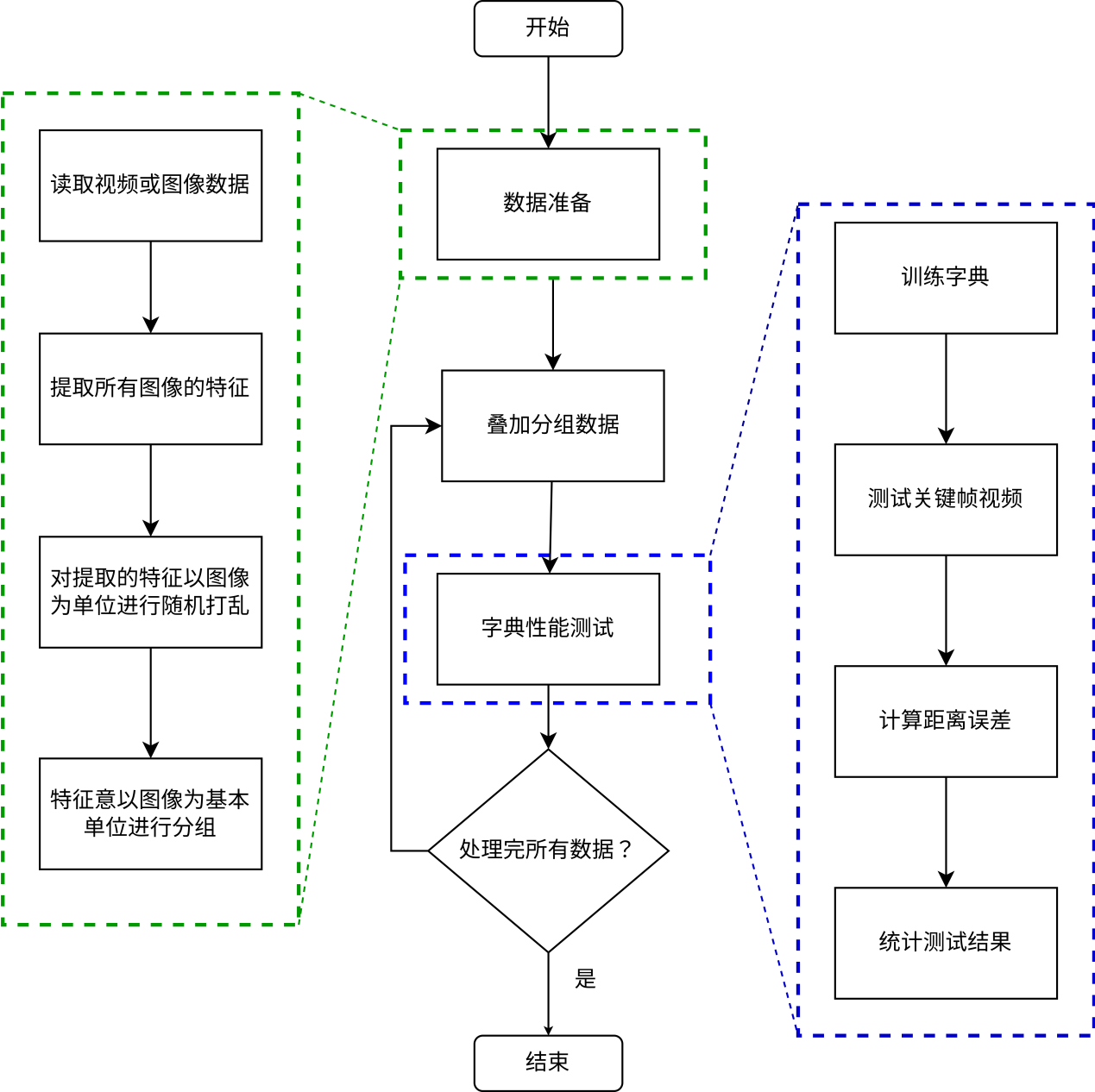
1. 在字典大于4^4的时候，成功率基本都为1，无法明确的区分字典的性能差异。
2. 基于视频的时间窗口进行判断，没有考虑视频的回环，其找的局部时间内的帧作为参考帧，对于回环帧当作错误查询，这是不合理的。

因此本次测试在原来的评价指标的基础上增加了新的评价指标，即**平均最小距离**，使用图像之间的位置距离作为评价指标，其思想**是查询图像的相似图像的拍摄位置是相近的**。它的计算方法与前面的评价指标的计算方法类似，具体如下：

1. 选取输入视频中的一帧，使用BoW查询其与视频中其它帧的相似性，每两张图像得到一个相似度得分。
2. 选取相似度得分中TopM个数据，作为BoW的查询结果。
3. 计算TopM中的图像与当前帧之间的位置距离，选取最小的距离作为查询的距离误差。
4. 统计所有查询图像距离误差的平均值，得到平均最小距离，作为查询结果的误差。

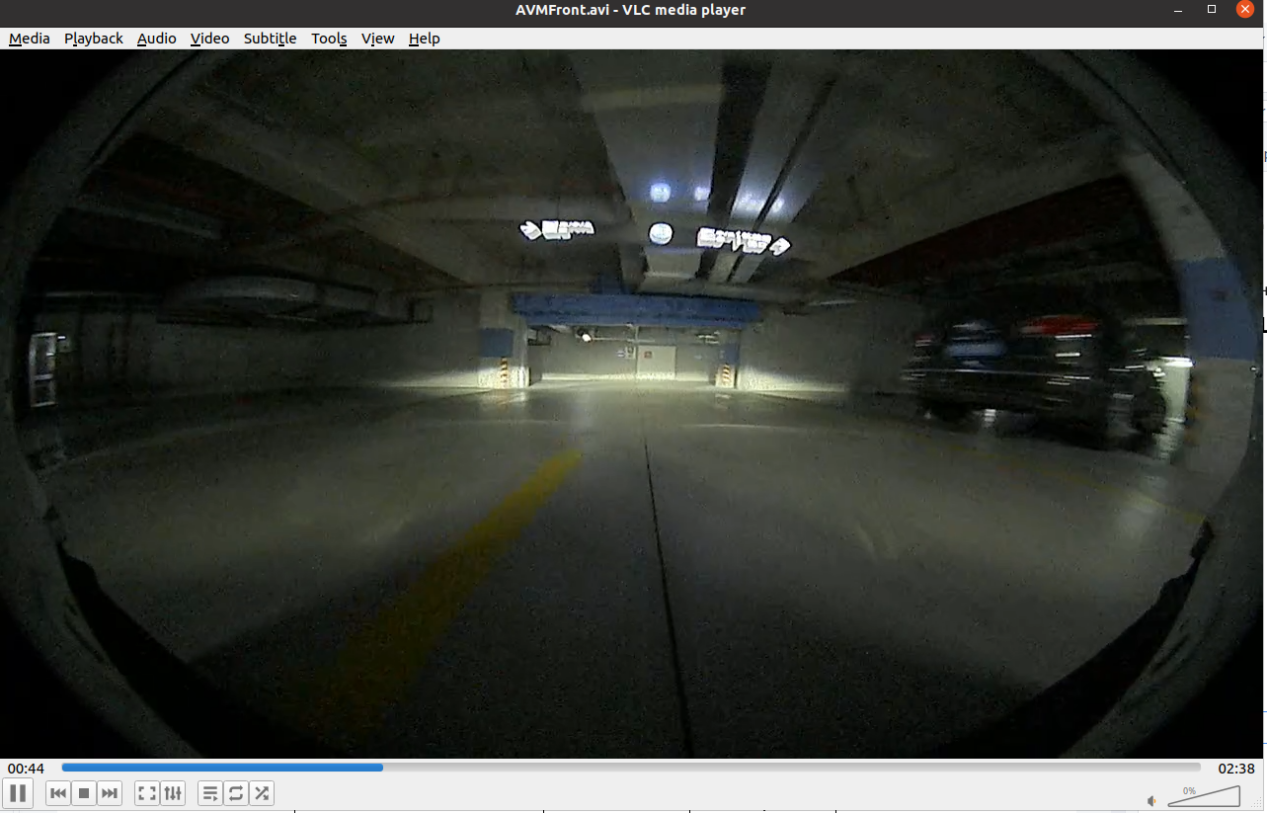
但是目前的计算方法**只考虑了位置**，没有考虑相机的拍摄角度，主要是由于我们的测试视频是车载的前视摄像头获取的，我们**假定其在回环时，相机的角度变化不大**。

### 测试流程



### 测试数据

本次测试共选取了5个地下停车场视频和1个公路视频作为训练数据，共约26000张图像，我们首先提取所有视频中的特征点，然后以图像为单位进行乱序处理，最后以200为一组逐步添加训练数据，共得到训练数据量为200～4000的20个不同数据量训练的字典。地下停车场和公路视频的示例图如下：



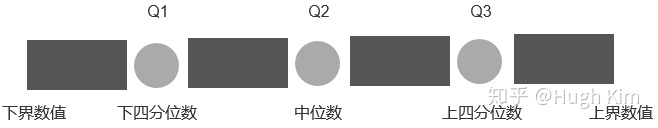
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | | Size | FPS | Frames | Time |
| 训练 | | | | | |
| garage | 2022.07.06@20.55.03/AVMFront.avi | 1280\*718 | 15 | 2379 | 2:38 |
| 2022.07.06@20.55.03/AVMLeft.avi | 1280\*718 | 15 | 2379 | 2:38 |
| 2022.07.06@20.55.03/AVMRight.avi | 1280\*718 | 15 | 2379 | 2:38 |
| 2022.07.06@20.55.03/AVMRear.avi | 1280\*718 | 15 | 2379 | 2:38 |
| 2022.07.06@20.55.03/AVMLeftCamera.avi | 1280\*718 | 15 | 2379 | 2:38 |
| road | \ | 640\*360 | 15 | 14792 | 16:26 |
| 测试 | | | | | |
| OfflineData2022.08.19@14.10.24\_loc/AVMFront\_KeyFrame.avi | | 1280\*718 | 15 | 6986 | 2:38 |

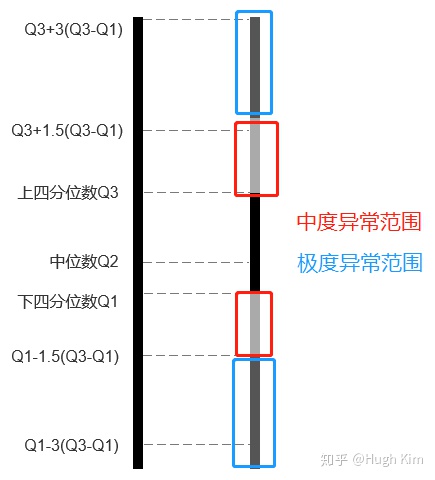
### 参数设置

|  |  |  |
| --- | --- | --- |
| 参数 | 推荐值 | 搜索范围 |
| K | 6 | \ |
| L | 6 | \ |
| windowSize | \ | \ |
| TopM | 4 | \ |
| trainVideoSampleRate | 1 | \ |
| testVideoSampleRate | 1 | \ |
| TrainDataNum | \ | 200~4000 |

### 测试结果

数据量测试的平均距离误差波动较大，考虑使用4分位数对测试结果进行分析，去除一些对平均误差影响较大的错误匹配。





### 结果分析

1. 使用Bicocca训练的字典，其平均距离在1～1.4之间波动，使用本地数据训练的字典，其平均距离在0.6～0.8之间波动。
2. 使用Bicocca训练的字典，其方差在6～10之间波动，使用本地数据训练的字典，方差在2～5之间波动。
3. 去掉serious数据后，

### 结论

1. **使用本地数据训练的字典，平均距离误差更小，且性能更稳定。**

## 字典的特征点匹配性能测试

### 测试目的

在ORB\_SLAM2的系统中，对DBoW的使用方式有两种：

1. 利用BoW的词向量在关键帧数据库中进行快速的相似图像检索。
2. 利用BoW对图像特征进行初步的分类，加速图像之间特征点的匹配。特征点的匹配在回环检测和参考帧跟踪模式中都有使用。其具体的使用方式为：
3. 将图像转为词向量时，在BoW的某一层中（通过levelsup指定）将落同一节点对应子树下的特征点分为一类，记录在featureVec中。
4. 在进行特征点匹配时，对落在同一节点子树下的特征点进行两两匹配，取特征描述符之间距离最小的特征点为一对匹配点。
5. 根据特征点描述符的距离阈值和角度（图像发生旋转时，特征点的旋转角度应该是一致的）过滤误匹配的特征点。
6. 根据匹配到的特征点数量判断两张图像是否具有足够的相似性。

其次在回环检测的局部地图跟踪过程中也使用了BoW加速特征点的匹配，其使用方法与上诉类似，只是得到的匹配点用于三角化得到地图点。

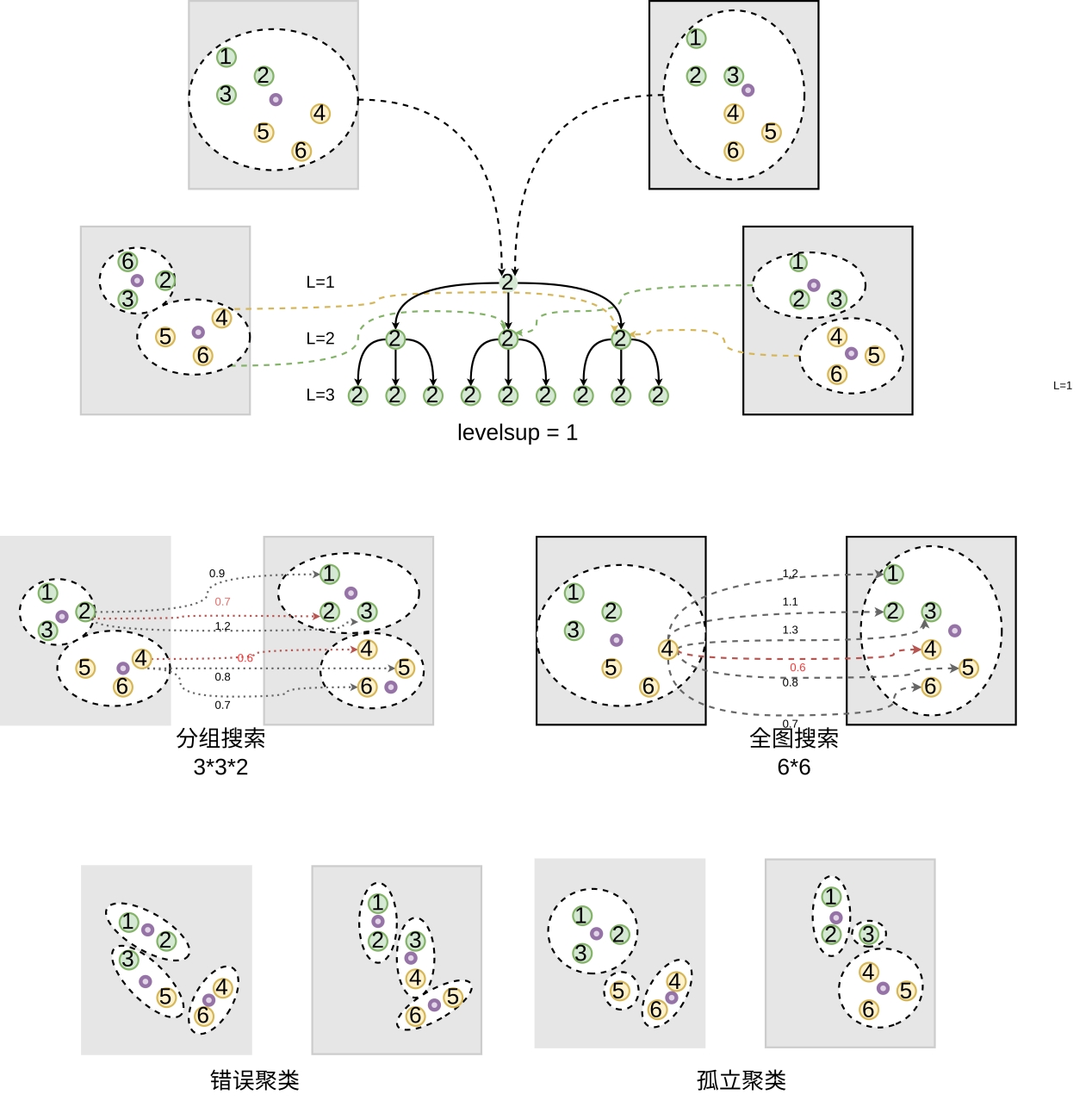
使用BoW对特征点进行初步分类，虽然加速了特征点的匹配过程（理论上的匹配效率提升了K(L-levelsup)倍），但是如果BoW的前L-levelsup层中的分类不准确，将原本相近的特征点分类到了不同的字节点中。这就会产生两种情况：

1. 特征点在另一张图像中找不到与匹配的特征点（没有特征点与其落在相同的子树下）
2. 特征点的匹配质量变差（最佳匹配点已被BoW分离），无法通过后续的距离阈值和角度检测。

这两种情况都会导致**匹配的特征点变少，**从而产生以下问题：

1. 回环检测检测失败，因为需要根据特征点的匹配数量选择候选帧。
2. 参考帧跟踪失败，因为需要根据特征点的匹配数量判断上一帧是否可以作为当前帧的参考帧。
3. 局部地图跟踪效果变差，因为三角化出的地图点变少，导致BA优化的效果变差。
4. 位姿优化的效果变差，导致图像帧之间的共视关系变弱（变弱可能不太准确），从而导致生成的地图点变少（因为生成的地图点必须能被反投影到关键帧上(深度域值) - Tracking::CreateNewKeyFrame），进而导致图像之间的特征点匹配变少（因为匹配的特征点必须要有对应的地图点 - ORBmatcher::searchbybow）

从以上分析可知，ORB\_slam2使用了BoW对特征点进行初步分类以加速图像之间特征点的匹配过程，其加速效率约为K(L-levelsup)，但是在使用BoW加速匹配时，也可能会导致性能的下降，这取决于BoW的前L-levelsup层聚类效果以及levelsup的选取。本节测试主要关注的是levelsup的选取，通过对同一字典使用不用的levelsup进行特征点的匹配，通过判断特征点的匹配情况，以**探讨levelsup的选取对特征点匹配效果的影响**。



### 评价指标

1. 本节的评价与以往不同，由于当levelsup=L时，两张图像的特征点之间会进行两两暴力匹配，从而得到一组全局距离最小的匹配结果，这是最慢的匹配，但也是理论上最准确的。因此本节以levelsup=L时的匹配结果作为基准匹配，计算levelsup=[0～L)时的匹配结果与基准匹配的差异（也即不一致的匹配）作为评价标准。差异越大，那么特征匹配的效果就越差。
2. 对于两幅图像A和B，P和Q分别为图像A和图像B中的提取的特征描述符，Bm,n为BoW中第m层的第n个节点，图像A和图像B中落在BoW的第m层的第n个节点子树下的特征点分别表示为Pm,n，Qm,n。则特征描述符的匹配过程表示为：

### 测试流程



### 测试结果

### 结果分析

1. levelsup相同的字典越大，没有匹配到的点越多
2. 同一个字典，levesup越大，通过ORB\_SLAM2的检验的匹配点数量越多，3. 在leveslup相同时，分支数越多，通过ORB\_SLAM2的检验的匹配点数量越少。
3. 当levelsup>4时，不同分支数得到的匹配点数量开始相差不大。也就是说不同分支数的字典，BoW字典的前两层聚类效果相差不大。

### 结论

1. 字典越小、levelsup越大，得到的特征点匹配数量越多。
2. BoW字典的前二层聚类效果相差不大（levels=4）。
3. Bicocca训练的8\_6字典与本地训练的8\_6字典在特征点匹配的性能上基本相同

## 训练数据随机性测试

### 测试方法

与数据量测试相同，只是这里固定了数据量进行多次测试。

### 测试结果

### 结论

1. **多次测试的7000和16500的误差均值相差较小。**

## ORB\_SLAM2回环测试

本次测试的主要目的是检测BoW字典对ORB\_SLAM2性能的影响。

### 结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **task model** | **vslam mileage (km)** | **time avg (s/km)** | **KF num (/1km)** | **map num (/km)** | **map size (MB/km)** | **BreakNum** |
| 6\_6 | VSLAM | 76.2128 | 641.2505 | 1593.9065 | 64388.4759 | 44.829 | 12 |
| loop close | 76.2128 | 797.0209 | 1405.7359 | 53078.479 | 39.6491 | 12 |
|  |  |  |  |  |  |  |  |
| 8\_6 | VSLAM | 76.2069 | 621.4295 | 1655.8728 | 64341.5156 | 46.0475 | 10 |
| loop close | 76.2069 | 786.18 | 1419.0046 | 51306.4429 | 39.6812 | 11 |
|  |  |  |  |  |  |  |  |
| 6\_6对比8\_6 | | 大 | 大 | 小 | 大 | 小 |  |
|  |  |  |  |  |  |  |  |
| 6-6 break num 对某个视频很大 | | | | | | |  |
|  |
| 8-6 break num 分布比较均匀 | | | | | | |  |
|  |