

**一. 概念**

k近邻 (k-Nearest Neighbors， KNN) 算法是一种**分类算法**， 应用场景有字符识别、 文本分类、 图像识别等领域。该算法的思想是： 一个样本与数据集中的k个样本最相似， 如果这k个样本中的大多数属于某一个类别， 则该样本也属于这个类别。在图像处理的描述子匹配也面临着同一个问题: **如何快速的找到当前样本点周围最近的K个样本点?** 如果是匹配的话K=1，如果用暴力匹配的话,算法的效率是N2, 当特征点的数量较多时,效率难以接受, 这一问题可以通过kd树来解决。

是一种空间划分数据结构，用于组织k维空间中的点，其本质上也就等同于二叉搜索树。不同的是，在kd树中每个节点存储的并不是一个值，而是一个k维的样本点。

**二. 构造kd树**

**1.计算每个维度的方差**;

**2.以方差最大的维度进行排序**; 因为这些数据在该维度上分散的比较开，我们就更容易在这个维度上将它们划分开，其次,该维度间距较大, 更容易达到维度距离大于记录的最小距离的条件, 实现剪枝;

**3.取排序后的中位数做父节点,划分左右区间递归构造左右节点;**

**三. kd树搜索**

1. 设定一个当前全局最佳点和全局最短距离, 分别用来保存当前离搜索点最近的样本点和最短距离;

2. 根据当前划分维度与二叉搜索树的性质(小的在左,大的在右)搜索样本点, 并记录走过的路径;

3. 根据空间局部性原理,离样本点最近的区域得到最近点的概率最大,从路径的尾向头遍历,

4. 如果样本点到当前点的维度距离大于全局最短距离, 则进行剪枝,否则更新全局最佳点与全局最短距离, 并递归遍历其另一子树;

**四. 比较**

还有一种优先搜索k-means树算法, 但是该算法质量非常依赖与参数K的选择, 且大多数情况下kd树都很有效。

**五. FLANN**

FLANN是快速最近逼近邻搜索库（Fast\_Library\_for\_Approximate\_Nearest\_Neighbors）的简称。它是一个对大数据集和高维特征进行最近邻搜索的算法的集合，而且这些算法都已经被优化过了, 可以根据数据本身选择最合适的算法来处理数据集。在面对大数据集是它的效果要好于BFMatcher。值得注意的是，FLANN匹配器只能使用SURF和SIFT算法来检测角点。

**六. 匹配点筛选**[->](https://blog.csdn.net/sinat_41686583/article/details/115186277)

**1**. 第一种方法：最简单的利用经验值筛选大于最小距离4倍的matches或者小于最大距离的0.6倍；

**2**. 第二种方法: 如果最近邻距离与次近邻距离比率ratio少于某个阈值T, 则接受这一对匹配点。因为对于错误匹配，由于特征空间的高维性，相似的距离可能有大量其他的错误匹配，从而它的ratio值比较高。显然降低这个比例阈值T，SIFT匹配点数目会减少，但更加稳定，反之亦然。建议ratio的取值原则如下:

ratio=0. 4：对于准确度要求高的匹配；

ratio=0. 6：对于匹配点数目要求比较多的匹配；

ratio=0. 5：一般情况下。

**3**. 第三种方法:

通过RANSAC方法计算透视变换矩阵来筛选符合相同透视的特征点，这样做可以去除很多错误的匹配; 该方法需要选择的参数较多;

4. GMS:见另一文档GMS。