

数字图像分析第二次作业

2018年5月1日

姓名：姜贵平

学号：SA17011142

1 task1：基于kmeans 的sift特征匹配的加速优化

实验有三个部分组成:task1.m, siftmatch.m, kmeansAcc.m。由task1.m调用siftmatch.m进行两幅图像相似特征的匹配，并设置标志位accFlag来决定是否使用kmeans加速优化。实验步骤如作业说明所介绍的，这里不再赘述。

我们来看看实验结果：

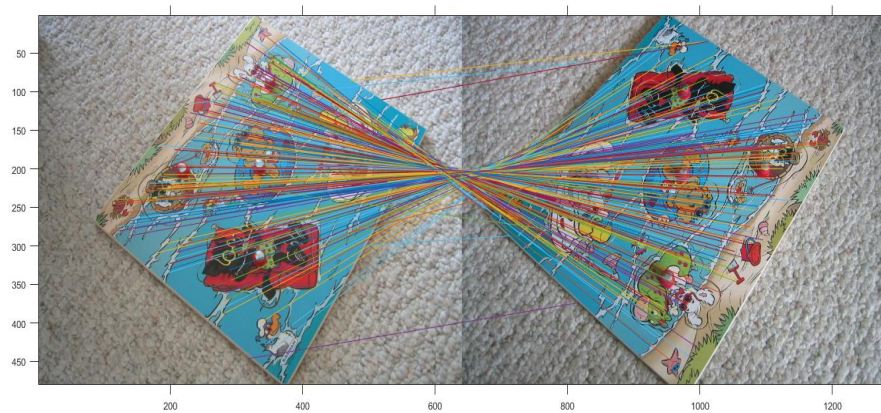


图 1: 未加速的匹配,耗时21.76s

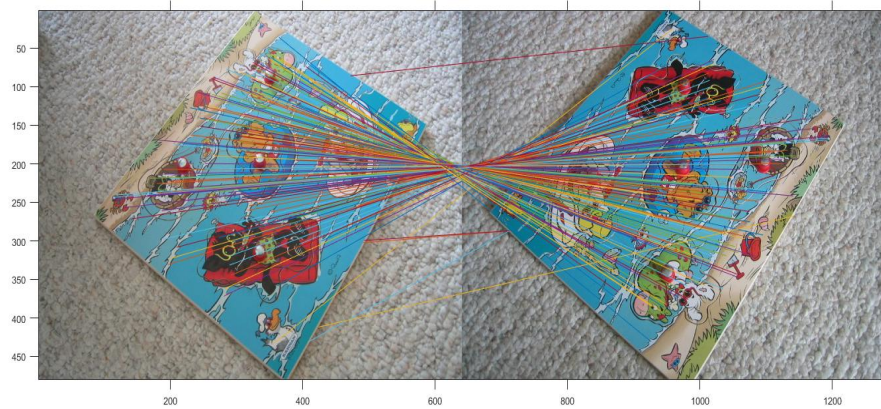


图 2: 使用kmeans加速的匹配,耗时11.7s

由实验结果看, 使用kmeans加速优化和原始匹配还是很相近的, 但耗时少一半, 可见kmeans确实起到加速优化的效果。

实验分析:

1 时间复杂度

给定两幅图像A: $m \times N_1$ 和B: $m \times N_2$ (sift特征维数, 本实验是128) 其时间复杂度为 $O(m \times N_1 \times N_2)$ 对于图像A每一个sift特征都与B中所有的sift的特征进行空间距离计算。利用kmeans进行加速, 我们对128维的sift特征分成16分, 对每一份的所有的特征进行kmeans聚类 ($k: 64$ 类,), 其中kmeans的时间复杂度为 $O(\text{Iter} \times N_1 \times k \times d)$ iter是迭代次数 (matlab默认100), N 是数据个数, k 是聚类个数, d 是每个数据的维度。我们需要对数据A和B分别聚类 $m/16$ 次, 本实验是8次。所以总共的时间复杂度为: $(O_A(\text{kmeans}) + O_B(\text{kmeans})) \times 8$, 计算16个中心distmatrix时间复杂的为 $16 \times k \times k$, 相对较小。所以采用kmeans聚类 N 很大的时候占有优势。粗略将图像sift算子大小带入, 估计一下, 大概加速了一倍, 与实验结果相吻合。

2 空间复杂度

原始方式, 最大存储是两个图像sift特征, 使用kmens加速优化之后需要额外建立一个 $16 \times k \times k$ 的中心距离的distmatrix, 以及每个数组每一段sift特征所属类别, 空间复杂度基本不变。

2 task2: 基于kmeans的图像匹配

本实验有4个部分组成，分别是利用kmeans计算codebook的codebook.m和利用vlad方式将每幅图像合成 $128 \times k$ 的长向量的imvlad.m，还有图像匹配imsearch.m。实验步骤如说明所示，这里不再赘述。
实验结果:

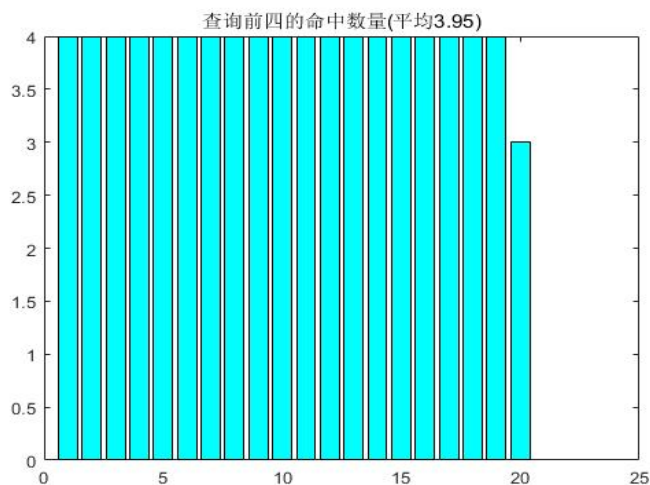


图 3: 查询前4命中

由图像可知，命中率很高。

实验分析

1 时间复杂度

本实验一共20幅图像，利用kmeans对所有的图像进行聚类，这个耗时最多，复杂度为 $O(\text{Iter} \times \sum N \times k \times d)$ ，其中iter是迭代次数（matlab默认100），N是所有图像sift特征和，k是聚类个数（本文16），d是每个数据的维度（本实验128），再分析一下imvlad对图像特征聚合，对于每一个我们需要计算 $N \times k \times d$ 次，进行聚合，相比kmeans相对较小。

2 空间复杂度

首先我们需要将20幅图像sift特征读取，需要 $20 \times N \times d$ 空间，这是本实验存储所需要的主要存储空间，然后将每幅图像聚合，需要一个额外

变量newFeat（大小 $128*k$ ）存储相差和，剩下的匹配内存消耗也比较小。
所需空间大约在 $O(20*N*d)$ 。