

多媒体搜索技术实验说明

一、实验课程要求

实验成绩由**实验课当堂验收成绩**和**实验报告**两部分组成。

其中，**实验课当堂任务**要求如下：

1. **两个人一组，自行组队**，在最后一次实验课下课之前完成实验任务，并找助教验收；
2. 实验要求现场运行源码，并根据源码改进和实验结果进行验收（做好截图和数据记录以便进行实验报告的撰写）。

实验报告的具体要求如下：

1. 封皮为华中科技大学，写清楚姓名，学号，班级，课程等必要信息
2. 报告中要说明实验的分组分工情况，并把个人负责的工作写清楚；
3. 报告内容**至少**包括实验任务，实验原理，实验结果，总结与分析等部分；
4. 实验报告每人一份，须在**11月8日之前**发到指定邮箱（邮件主题请注明：多媒体搜索技术实验报告-姓名-班级-学号，实验报告 文件名同邮件主题一致并放到附件之中，指定邮箱地址 wqj@hust.edu.cn）。

二、实验源码与数据集说明

1. 源码与数据集介绍

本次实验课程的源码和链接已经上传到百度云盘（链接如下所示），其中数据集很大需要提前下载。

源码： 链接：<https://pan.baidu.com/s/1xF9a5rIhBS2Cfv5P80DSyg>
提取码：kopi

数据集： 链接：https://pan.baidu.com/s/1HL8l2XJFevG_38Dw0e8lyw
提取码：uvf6

其中，源码有两个版本，基于 python 的 bag-of-words-python-dev-version；和基于 C++ 的 bag-of-words-dev-version。数据集 dataset_oxbuild，包括了 1000 多张图片。

2. 实验环境配置与 python 源码 debug

本次实验要求在 **python 版本** 上进行操作，以下我只介绍源码在 windows 系统的配置过程（linux 系统类似，可以自行探索）。

1. 在 windows 平台配置 python2.7 的运行环境，具体过程可以自行百度，提供一个链接供大家参考：<https://www.cnblogs.com/parryyang/p/5328368.html>
2. 打开 windows 终端测试 python 环境配置情况，配置 pip 链接如上，如果 pip 安装三方库下载速度缓慢，请更换**源镜像**方法参

照如下链接：<https://blog.csdn.net/chenghuikai/article/details/55258957>

3. 采用 pip 安装 python 源码需要的三方库（eg: `pip install numpy`, `pip install scipy` 等），**opencv 版本请安装 `pip install --user opencv-contrib-python==3.3.0.10`**,

如果源码在运行的时候 opencv 版本报错，请自行百度或参考如下链接解决问题 <https://www.cnblogs.com/Allen-rg/p/9667926.html>

4. 实验源码运行，在终端下进入到 python 源码根目录，并将数据集复制到相应文件夹。

构建码本命令：`python findFeatures.py -t dataset/training/`

图片检索命令：`python search.py -i dataset/testing/all_souls_000000.jpg`

5. 如果代码再生成码本过程中显示内存错误，如下所示：

```
File "C:\Python27\lib\site-packages\scipy\cluster\vq.py", line 211, in vq
    return _vq.vq(c_obs, c_code_book)
File "_vq.pyx", line 235, in scipy.cluster._vq.vq
File "_vq.pyx", line 101, in scipy.cluster._vq._vq
MemoryError
```

解决方法如下：

(1) `findFeatures.py` 源码之中对读入的图片做 `resize` 处理，改变图像的大小以减小特征点数量（不推荐）：

```
for i, image_path in enumerate(image_paths):
    im = cv2.imread(image_path)
    im_size = im.shape
    im = cv2.resize(im, (im_size[1]/4, im_size[0]/4))

    print "Extract SIFT of %s image, %d of %d images" %(training_names[i], i, len(image_paths))
```

(2) 在生成码本时对图像的描述子进行降采样（不推荐）：

```
downsampling = 2
descriptors = des_list[0][1][:downsampling,:]
for image_path, descriptor in des_list[1:]:
    # print np.size(descriptor)
    # print image_path
    # print descriptor
    descriptors = np.vstack((descriptors, descriptor[:downsampling,:]))
```

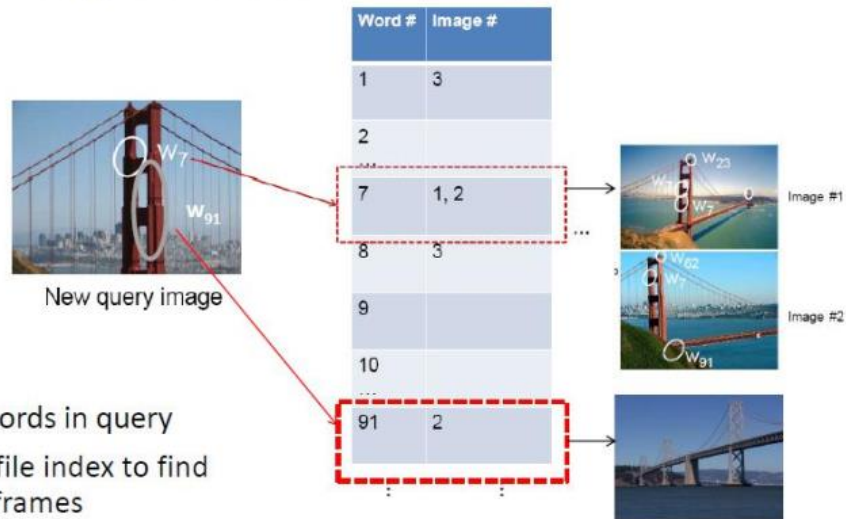
(3) 采用倒词表（inverted file index）操作，可以有效减少码本的内存，避免暴力 k-means 聚类之后码本矩阵过大造成内存溢出的情况（推荐）。

三、实验任务

实验要求以 python 版本的图像检索源码为基础，实验任务如下所示，其中数据集采用 `dataset_oxbuild`，实验提供的版本已经是对于原始数据集进行了降采样之后的结果。本次实验提供的 C++ 源码已经基本实现了以下功能，可以参考 C++ 源码与课程 PPT 来完成实验任务。

1. 倒词表 (Inverted file index)

Inverted file index and bags of words similarity



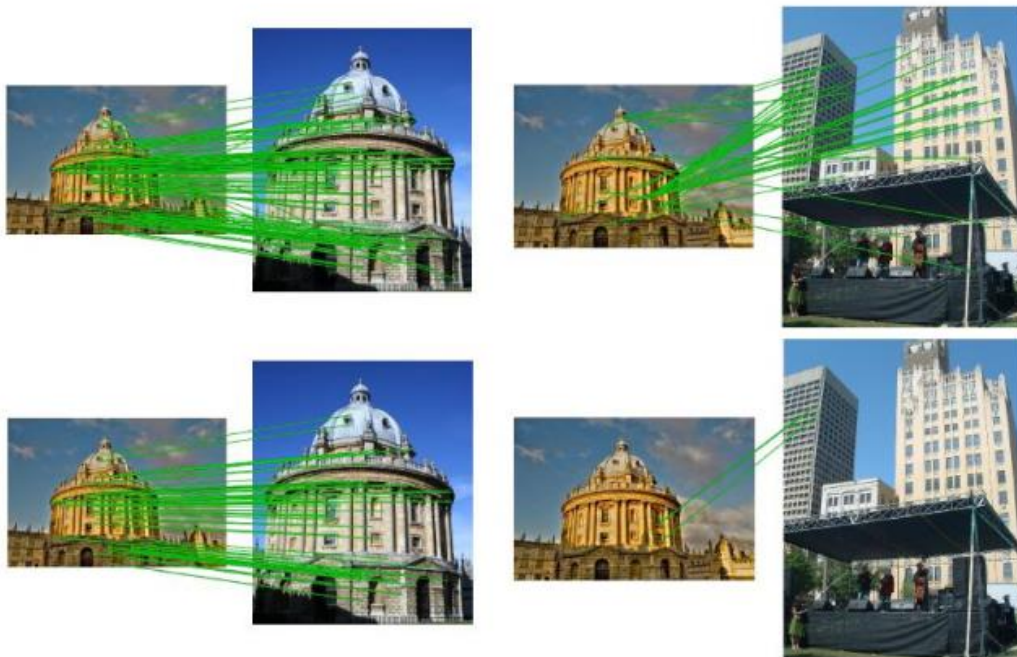
1. Extract words in query
2. Inverted file index to find relevant frames
3. Compare word counts

2. 加入空间校验 (Spatial Verification)

Spatial Verification: two basic strategies

- RANSAC
 - Typically sort by BoW similarity as initial filter
 - Verify by checking support (inliers) for possible transformations
 - e.g., "success" if find a transformation with $> N$ inlier correspondences
- Generalized Hough Transform
 - Let each matched feature cast a vote on location, scale, orientation of the model object
 - Verify parameters with enough votes

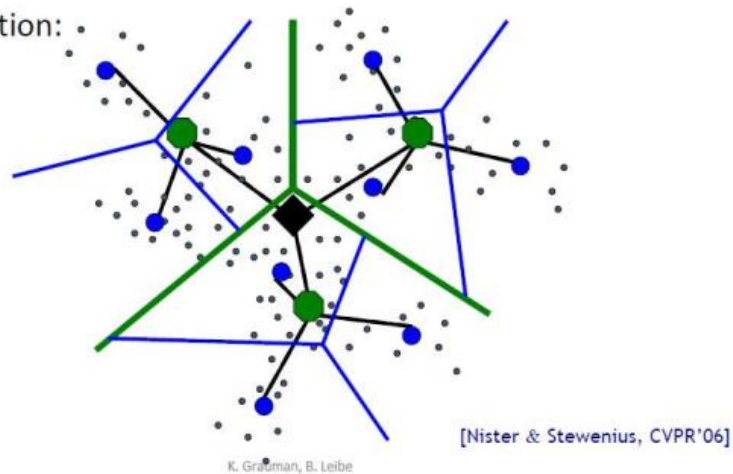
RANSAC verification



3. 采用 Vocabulary Tree (hierarchical clustering) 的方式进行聚类

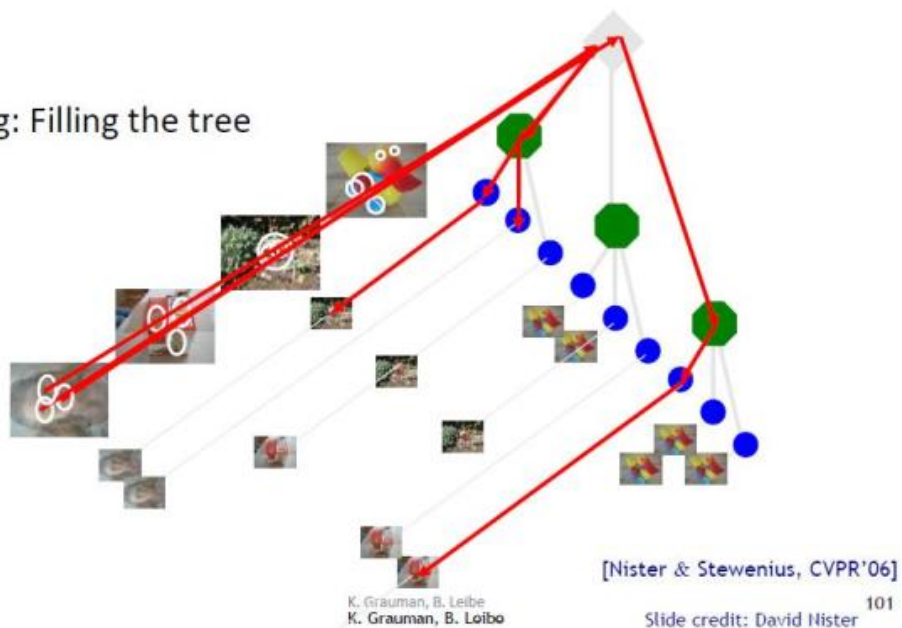
Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:



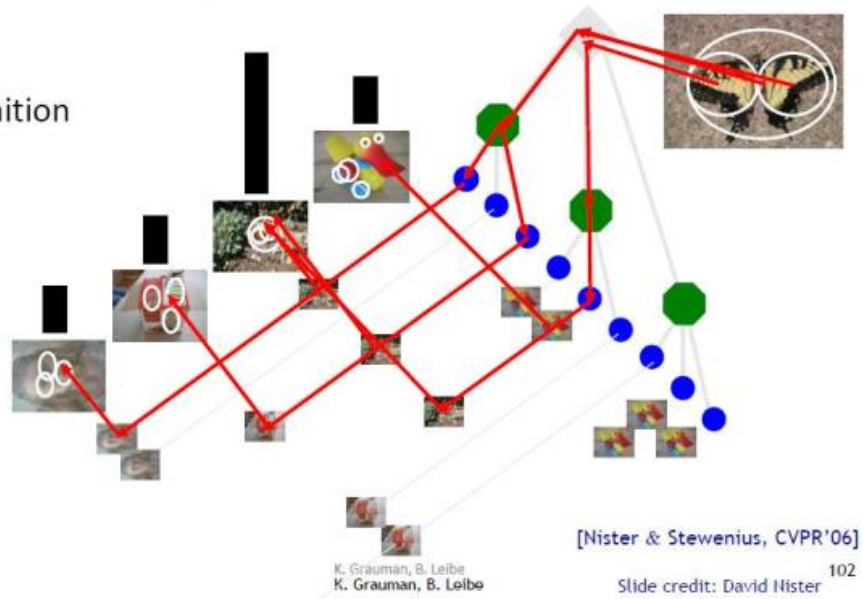
Vocabulary Tree

- Training: Filling the tree



Vocabulary Tree

- Recognition



4. 增加 Relevance feedback 功能

