

周总结七

卢婧宇

13020022023

一、本周进展

- 1.安装并学习 latex，上次安装过程中出现了由于默认路径安装后占满内存的情况，这次学习了如何更改路径。
- 2.对 vlfeat 官网中针对 sift 的说明和调用进行学习，大致整理了思路，了解了每一步骤需要调用的函数。总的来说，大体思路就是，提取关键点，对关键点附加详细的信息（局部特征）也就是所谓的描述器，关键点匹配，消除错配点。

● 使用 `vl_sift_new()` 初始化 SIFT 过滤器对象。滤波器可以重用于相同大小的多个图像（例如，对于整个视频序列）。

`vl_sift_new` 的调用：

`VLsiftFilt* vl_sift_new (int width,int height,int noctaves,int nlevels,int o_min)`

`width` 图像的宽度

`height` 图像的高度

`noctaves` octave 数

`nlevels` octave 的层数

`o_min` 第一个 octave 的指数

该功能为指定的图像和比例尺空间几何分配并返回一个新的 SIFT 过滤器

●对于尺度空间中的 octave：

- ◆使用 `vl_sift_process_first_octave()` 或 `vl_sift_process_next_octave()` 计算 DOG 标度空间的下一个 octave。（如果返回 `VL_ERR_EOF`，则停止处理）。

`vl_sift_process_first_octave()`：

`int vl_sift_process_first_octave (VLsiftFilt * f, vl_sift_pix const * im)`

`f` sift 过滤器

`im` 图像数据

该函数通过计算低八度的高斯尺度空间开始处理新图像。它还清空内部关键点缓冲区。错误代码。如果没有更多的 octave 要处理，该函数返回 `VL_ERR_EOF`（文件结尾或序列发生错误）。

- ◆使用 `vl_sift_detect()` 运行 SIFT 检测器以获取关键点。

`void vl_sift_detect (VLsiftFilt * f)`

功能检测当前倍频程中的关键点，填充内部关键点缓冲区。

关键点可以通过 `vl_sift_get_keypoints()` 检索。

调用语句 `VLsiftKeypoint const * vl_sift_get_keypoints (VLsiftFilt const *f)` 返回一个指向关键点列表的指针。

- ◆对于每个关键点

a.使用 `vl_sift_calc_keypoint_orientations()` 获取关键点方向。

调用：`int vl_sift_calc_keypoint_orientations (VLsiftFilt * f,double angles[4],VLsiftKeypoint const * k)`

`f` sift 过滤器

`angles` 方向输出

`keypoint` 关键点

该函数计算关键点 `k` 的方向。该函数返回找到的方向数（最多四个）。方向被写入的是向量的角度。

b. 对于每个方向

使用 `vl_sift_calc_keypoint_descriptor()` 获取关键点描述符。

调用 `void vl_sift_calc_keypoint_descriptor (VLSiftFilt * f, vl_sift_pix * descr, VLSiftKeypoint const * k, double angle0)`

`descr` SIFT 描述符 (输出)

`k` 关键点

`angle0` 关键点方向

该函数计算取向角度为 0 的关键点 `k` 的 SIFT 描述符。该函数填充必须足够大以容纳描述符的缓冲区。

- ♦ 通过 `vl_sift_delete()` 删除 SIFT 过滤器。

`void vl_sift_detect (VLSiftFilt * f)`

功能检测当前倍频程中的关键点，填充内部关键点缓冲区。关键点可以通过

`vl_sift_get_keypoints()` 实现。

3. 对编程进行学习，对计算机导论与 c 语言基础这门课程进行学习，了解了部分计算机的历史，回顾 c 语言的基本知识。适当进行编程训练。

二、下周安排

1. 继续对 latex 进行学习。
2. 继续将遇到的问题、做的笔记以及视频中的作业记录在 github。
3. 参考网上的一些调用方式进行学习，开始编程，争取下周可以将 sift 部分完成，并且实验。