

# 第1讲：特征点检测与匹配 作业

## 1.1 安装并编译代码

从课程网站上下载本课程的参考代码（ImageBasedModellibgEdu.tar）并编译。本课程的代码采用 Cmake 管理，具有跨平台的特性，理论上在 Windows, MacOS 以及 Linux 系统上都可与顺利编译并运行，但是本课程目前仅提供 MacOS 和 Linux 系统的调试。建议采用 Linux 系统，因为 ubuntu 系统在工程上比较便利，体验过 windows 和 Linux 下的依赖库的编译和安装便可深有体会。建议采用 clion IDE，便于阅读和修改代码，这是一款跨平台采用 CMake 管理的开发环境，大家可以到其网站上申请免费版本。

## 1.2 推导 SIFT 点亚像素定位的公式，并完成代码

SIFT 特征点检测过程中寻找 DoG 极值点是在离散的三维空间上进行的，因此当图像分辨率尤其是尺度的分辨率较低的时候，找到的极值点位置可能不准确。为了解决这个问题，比较好的做法是在将 DoG 值函数  $f(\mathbf{x})$  假设为关于三维空间  $\mathbf{x}=[x, y, \sigma]^T$  中的连续函数，将其在极值点处

$\mathbf{x}_0=[x_0, y_0, \sigma_0]^T$  进行泰勒二阶近似，得到如下表达式

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^T \nabla^2 f(\mathbf{x}_0) (\mathbf{x} - \mathbf{x}_0)$$

(1) 求  $f(\mathbf{x})$  极值点的位置  $\mathbf{x}^*$  以及  $f(\mathbf{x}^*)$  的表达式。

(2) 阅读 Sift 实现代码，并填充极值点求解部分的代码。与 Sift 特征点相关的代码位于 features/sift.h 和 features/sift.cc。其中与极值点相关的函数是 Sift::keypoint\_localization()。

(3) 完成后测试 sift 生成代码，位于 examples/task1/test\_features.cc，结果保存在 tmp 文件夹中。

## 1.3 阅读 Sift 特征配代码

(1) 与特征配相关的代码包括：

features/matching.h

features/matching.cc

features/matching\_base.h (基类提供接口，有两个子类分别表示两种匹配策略)

features/exhaustive\_matching.h (最近邻暴力搜索)

features/exhaustive\_matching.cc

features/cascade\_hashing(利用哈希表进行加速)

features/cascade\_hashing.cc

(2) 填充 matching.h 中函数 oneway\_match() 函数中关于 lowe-ratio 利用最近邻与次近邻比进行筛选的代码部分。

(3) 完成后测试 examples/task1/test\_matching.cc 函数，结果保存在 tmp 文件夹中,比较添加 lowe-ratio 前后的匹配效果。