计算机视觉课程第一次作业：兴趣点/关键点检测及匹配

作业概要：

1. 用哈里斯(Harris)角点检测算法或拉普拉斯高斯滤波器(LoG)，对拍摄场景相同但视角不同的两幅图像进行关键点检测。（见 dlut_cv22_hw1/data/Notre Dame 文件夹）注：两种方法任选其一。
2. 对步骤1中的关键点提取SIFT描述子，为减小工作量，SIFT编码不要求实现算法的所有步骤，只需要实现指定环节，完成一种“简化的SIFT”即可。如果能按照David Lowe的论文实现完整的SIFT描述子, 则酌情加分，但最后总分不超过100分。**此步骤本科生不做硬性要求，详见“具体要求”一节。**
3. 利用步骤2中的关键点描述子进行关键点匹配；如果能够进一步正确估计出两幅图像之间变换关系(Affine transformation or Homography), 则酌情加分，但最后总分不超过100分。**此步骤本科生不做硬性要求，详见“具体要求”一节。**

语言与环境：

编程语言：python3
测试环境(评分环境)：linux, Intel i5，1.6GHz, RAM 8G
注: python程序在windows & linux平台上的兼容性很好，程序不需要额外适应平台。

具体要求：

附件中已经提供了相应功能的空文件**需严格按照下述要求，在相应文件中实现对应功能**，否则评分时代码可能会报错。

1. **研究生 & 本科生：**角点检测程序写在提供的 dlut_cv22_hw1/code/student_corner_detector.py 文件中
2. **研究生：**简化的SIFT特征描述子程序写在提供的 dlut_cv22_hw1/code/student_sift.py 文件中
3. **研究生：**特征匹配程序写在 dlut_cv22_hw1/code/student_feature_matching.py 文件中

• **研究生 & 本科生：**哈里斯角点检测（ student_corner_detector.py）
在 student_corner_detector.py文件中提供了一些具体的提示及要求。注意在实现该功能的过程中，无需进行尺度不变与方向估计。此外，由于进行哈里斯角点检测后，对比度强烈的区域可能会出现非常密集的关键点，所以还需要实现非极大值抑制(non-maximal suppression)的功能。为了减小后续的特征编码及匹配的计算代价，需要合理设置非极大值抑制的参数，保证完整的程序可以在测试环境下运行10min以内可以运行完，否则酌情扣分。注意：哈里斯角点检测与非极大值抑制均为采分点。

• **研究生 & 本科生：**拉普拉斯高斯角点检测（ student_corner_detector.py）
1. 拉普拉斯高斯算子需要通过对拉普拉斯算子与高斯算子卷积得出，不可直接赋值。其中，卷积过程需要手动编码实现，不可直接调用包实现。 2. 为提升代码效率，利用拉普拉斯高斯算子对图像进行滤波时，可调用集成工具包中的滤波函数进行。3. 过零点的阈值自行设置。 4. 由于进行拉普拉斯高斯滤波后，对比度强烈的区域可能会出现非常密集的关键点，所以需要对候选关键点采用非极大值抑制(non-maximal suppression)，减少冗余。在 student_corner_detector.py文件中提供了一些具体的提示及要求。注意：拉普拉斯角点检测与非极大值抑制均为采分点。

• **研究生：**简化的SIFT特征描述子 (student_sift.py)
为了减小工作量，SIFT特征部分不需要实现完整的SIFT算法。关键点直接用哈里斯角点检测的结果，以关键点为中心，划分出一个16x16像素大小的区域作为该关键点的特征提取区域，将该区域按

照4x4的大小划分cell。对每个cell计算梯度直方图，按照相位(即梯度方向)将直方图设置为8个区间，统计每个区间内梯度的幅值，直接将其作为该cell的SIFT特征。因此一个关键点最终得到的SIFT特征维度将是4x4x8=128维。最终对每个关键点得到的SIFT特征向量，规范化到单位长度。更多详细信息参见 student_sift.py 文件。

- **研究生：**特征匹配 (student_feature_matching.py)

特征匹配的方法请采用最近邻距离比(Nearest Neighbor Distance Ratio)

注：对于“简化的SIFT特征描述子”与“特征匹配”两个环节，本科生不做要求。对于额外实现这两个环节的本科生，酌情加分，加分上限为20分，且总分不超过100分。

启动代码

1. dlut_cv22_hw1/code/main.py为整个project的主函数，包括上述三个功能的依次调用、匹配结果可视化及匹配结果打分。可视化只是辅助进行代码调试及结果展示的工具，不作为评分依据。为方便算法调试，工程中提供了图像匹配的真值：dlut_cv22_hw1/data/Notre Dame/Notre_Dame_match_ground_truth.pkl。evaluate_correspondence()会根据真值自动评估程序输出的匹配结果的准确率(详见“评价标准”一节)。虽然工程提供了真值，但这并不意味着可以直接提交真值作为结果，因为最终只需要提交程序，不提交匹配结果，得分将由提交程序的性能表现决定。
2. main.py及“具体要求”中提到的三个文件，各个空函数的输出和输入变量均为示例，具体传入传出哪些变量，可根据实际情况灵活处理。

建议的代码实现流程：

- 仅实现角点检测一个要求：

直接在 get_interest_points()函数中完成一种角点检测结果即可，程序会自动对检测结果进行评估。

- 完整实现所有作业要求：

本作业的三个问题环环相扣，角点检测结果将对第二、三个问题的表现产生直接的影响。如果按部就班依次完整实现这三个功能，一旦匹配结果较差，无法定位是哪一个步骤出的问题。所以建议在第一个步骤中先引入哈里斯角点检测或拉普拉斯滤波器角点检测的真值作为一个伪检测结果，先实现后续的功能，等后面的功能基本可以确定没有问题了，最后实现哈里斯角点检测的功能。**强烈建议同学按照下述的步骤进行编程及调试。**

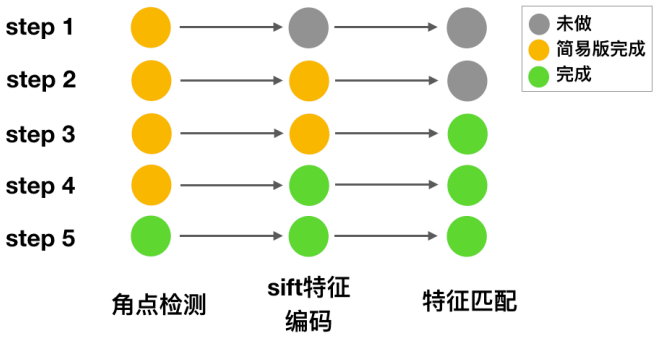

图1. 各步骤的代码进度示意图

1. 首先，用 cheat_interest_points() 代替 get_interest_points()。(注释dlut_cv22_hw1/code/main.py中44、45行，取消47、48行的注释) 其目的是加载关键点检测的真值坐标，以便代码调试。该函数并不允许用于最终的作业提交。cheat_interest_points()返回的真值坐标是浮点数。
2. 用 get_feature()实现提取一个简单的图形块作为'模板'的特征。例如，可以以每个关键点为中心，划分出16x16大小的区域，将该区域的像素灰度值直接作为提取的特征。这个特征并不理想，因为它并不能对光照变化、对比度变化、空间偏移等变化具有不变性，但这个'模版特征'非常易于实现，因此可以作为一个简单的基准(baseline)，便于后续步骤的调试对比。
3. 实现 match_features()。main.py中默认match_feature中提交的匹配点对的位置坐标按照置信度降序排列，默认取前100个置信度最高的匹配点对进行评估。如果采用步骤2中的简易特征(16x16=256维)，在Notre Dame 图像对上大致能够达到40%+的准确率。
4. 按照题目要求在get_features()函数中完成简化的SIFT描述子。此时Notre Dame 图像匹配的准确率约为70%。
5. 编程实现基于哈里斯角点检测的 get_interest_points()，停止使用cheat_interest_points()。哈里斯角点检测的结果一般无法做到与真值完全吻合，所以准确率会有所下降。可以通过增加提交的关键点的个数来覆盖住尽可能多的真值，提交置信度最高的100个特征。在Notre Dame 图像对上，只要所提交的置信度最高的100个匹配结果能够达到80%以上的准确率，就能够得到本次作业匹配性能评估环节的满分。

注意：上述各个步骤中的准确率仅为参考值，如果实际运行过程中有少量的性能偏差，继续下一个步骤即可。

函数限制与提示：

哈里斯角点检测、拉普拉斯高斯角点检测、SIFT特征提取及最近邻距离比的程序必须自己完成，**禁止直接调用python中现有的函数，否则该作业直接记零分**。

禁止使用的python函数：任何检测关键点的函数、计算直方图的函数、计算梯度的函数、计算最近邻比率的函数。包括但不限于：
cv2.Laplacian(), cv2.SIFT(), cv2.SURF(), cv2.BFMatcher(), cv2.BFMatcher.match(), np.gradient(), cv2.FlannBasedMatcher().knnMatch(), cv2.BFMatcher().knnMatch(), cv2.HOGDescriptor(), cv2.cornerHarris(), cv2.FastFeatureDetector(), cv2.ORB(), skimage.feature(), skimage.feature.hog(), skimage.feature.daisy(), skimage.feature.corner_harris(), skimage.feature.corner_shi_tomasi(), skimage.feature.match_descriptors(), skimage.feature.ORB().

可能用到的python的函数：（滤波允许调用函数实现）
np.arctan2(), np.sort(), np.reshape(), np.newaxis, np.argsort(), np.hypot(), np.fliplr(), np.flipud(), cv2.Sobel(), cv2.filter2D(), cv2.getGaussianKernel(), scipy.signal.convolve()

代码的其他注意事项：

1. 请注意，附件中给出了整个工程涉及到的所有文件，但最后提交的代码中只包括"具体要求"一节中提到的三个文件，以及main.py，**其他文件并不提交。**对于工程中的其他文件将会在评分时直接使用最初工程中的代码版本，所以**切勿在调试时修改其他文件**，以免造成调试与评分测试的程序不一致，如有特殊情况可以与助教高子淋联系。

2. main.py中需要在13、14行手动输入自己的学号与姓名，其余代码在作业完成过程中几乎不用更改，但是为了最大程度地保证各位同学程序的灵活性，允许修改main.py，但要注意main.py中的下述变量，不要更改：feature_width, scale_factor.
3. 程序中需要的参数在调试后需直接设置为默认值，保证程序可以独立地完整运行，不要设置中途从键盘获取反馈。
4. 程序中的坐标x为图像x轴方向的索引，即列号；相应的y为图像的行号。在这一约定下，numpy中默认的图像维度返回顺序是[y,x]，在程序中注意各函数返回的坐标不要写反。

作业提交：

1. 实现相应功能的三个程序文件: student_corner_detector.py student_sift.py student_feature_matching.py，以及主程序 main.py.
2. 结合程序对涉及到的算法进行简要介绍，要求为pdf格式，页数不限。
3. 请将上述五个文件打包发送到 dlut_cv@163.com 邮箱中，压缩包命名方式：学号_姓名.zip，如：22109001_李晓明.zip. 注意只提交这五个文件即可，其他程序或结果不要提交。
4. 作业提交的截止时间：4月5日23:59，逾期视作未提交。

ps: 本次作业建议至少提前一周开始准备。

性能评分的过程及标准：

最后得分一部分由代码及报告决定，另一部分由代码的性能决定。工程可以自动对提交的代码进行性能评判，在代码提交后，将由以下过程进行评分：

1. 评分时，将同学提交的四个程序拷贝到工程的相应位置，对性能的评分将参照程序自动评分的结果，所用工程为附件中的完整工程。各位同学提交之前可自行进行测试。评分运行命令：python main.py
2. 关键点判定为正确的标准：（详见 dlut_cv22_hw1/code/utils.py中的evaluate_keypoints())
   找到与预测的关键点空间距离最近的真值，距离小于80个像素记为正确，每个真值只能对应一个关键点。
3. 匹配点判定为正确的标准：（详见 dlut_cv22_hw1/code/utils.py中的evaluate_correspondence())
   假定对于A,B两幅图，已知图A所有关键点的真值坐标，及图A中每个关键点在图B上的最佳匹配点。现有一对预测的匹配点：图A的关键点a，及其在图B中的对应匹配点b。那么对于a与b是否为一对准确的匹配点，判定方法如下：在图A的所有关键点真值中，找到与a空间位置最接近的真值像素点g_a，并得到g_a在图B中的匹配点g_b，如果a,b,g_a,g_b之间的关系能够同时满足以下两个条件，即判定为预测准确：
   (1) g_a与a的空间距离小于150个像素，
   (2)计算a与b的空间距离d_{ab}，计算g_a与g_b之间的空间距离$d_{g_{ab}}$。d_{ab}与$d_{g_{ab}}$之间的差值小于25个像素。

评分标准：按满分100分计

- 程序得分：满分50分
  相应算法实现正确无误，代码运行速度快。

- 书面报告及实验分析：满分50分

  所使用的理论、方法和算法描述简洁准确，条理清楚，逻辑性强；实验充分，分析深入，匹配正确率高。

-5分 x N：不遵循“代码的其他注意事项”及“作业提交”两节中的要求，每发生一处扣5分。