实验三：全景图

0. 关键信息

本实验 2 人 1 组完成（报告中要注明两人工作量的比例）。

代码与报告提交信箱：visionexp@126.com

截止日期：2019 年 12 月 3 日 23 点 59 分

1. 概述

在本实验中，你会将一系列水平重叠的照片合成一幅全景图。你将首先使用 ORB 特征检测器和描述符检测图像中的特征，并在其他图像中找到最佳匹配特征；然后，你将使用 RANSAC 对齐照片（确定它们的重叠和相对位置），最后将生成的图像混合到一个无缝的全景图中。我们为你提供了一个图形界面，可让你查看流程中各个中间步骤的结果。我们还为你提供了一些测试图像和框架代码。

下面列出了创建全景图的所有步骤。你将实现两种拼接全景图的方法：使用平移变换（你需要首先将图片转为球面坐标）和单应映射。方括号中的步骤仅用于球面投影方法：

1) 在三脚架（或手持设备）上拍照
2) [变形到球面坐标]
3) 提取特征
4) 匹配特征
5) 使用 RANSAC 对齐相邻图像对
6) 写出相邻图像对之间的变换矩阵
7) 纠正漂移
8) 将所有图像融合
9) 裁剪结果

2. 实施细节

注意：不要修改 TODO 块之外的代码

2.1 将每张图像变形为球面坐标

（文件：warp.py，函数：computeSphericalWarpMappings）

[**TODO 1**] 填写 *computeSphericalWarpMappings* 函数中的代码，计算从普通图像到球面坐标变换的**逆映射**以卷绕图像：

1) 使用讲义（第 11 讲：全景图）中的坐标变换方程将给定的球面图像坐标转换为相应的平面图像坐标；
2) 使用讲义（第 10 讲：相机）中的公式校正径向畸变。

（注 1：测试此函数时必须使用给定图像的焦距 f 估计值。如果使用不同的图像尺寸，请记住根据图像尺寸缩放 f。）

（注 2：使用单应映射对齐相邻图像时不使用本函数）

2.2 对齐相邻图像对

（文件：alignment.py，函数：alignPair，getInliers，computeHomography 和 leastSquaresFit）

[**TODO 2,3**] computeHomography 从图像 1、图像 2 中的两个特征集 f1、f2 及特征间的匹配关系中估计从图像 1 到图像 2 的单应映射。你将使用奇异值分解 SVD 来计算最佳拟合单应映射。参考讲义（第 8 讲 : 图像配准）：对于方程式 Ah = 0，最佳解 h 是 A^TA 具有最小特征值的特征向量，其中 A^T 是 A 的转置。回想一下，SVD 通过 $A=USV^T$ 分解矩阵，其中 U 和 V 是正交矩阵，它们的列向量是左和右奇异向量，S 是奇异值的对角矩阵，通常从最大到最小排序。此外，奇异向量和特征向量之间存在非常强的关系。考虑：$A^TA=(VSU^T)(USV^T)=V(S^2)V^T$，也就是说，A 的右奇异向量是 A^TA 的特征向量，A^TA 的特征值是 A 的奇异值的平方。

[**TODO 4**] AlignPair 的参数是两个特征集 f1 和 f2，以及两幅图特征之间的匹配对列表 matches，以及 m(MotionModel)；它估计并返回图像间的最佳变换矩阵 M。枚举值 m(MotionModel)有两个可能的值：eTranslate 和 eHomography,分别表示用平移变换对齐，或者用单应映射对齐。AlignPair 使用 RANSAC 估计 M：首先，它随机抽出一组最小的特征匹配对（对于平移变换只需要一个匹配对，对于单应映射需要四个匹配对），估计相应的变换矩阵（对齐）（在单应映射时调用 computeHomography），然后调用 getInliers 获取与当前运动估计一致的特征匹配索引；在重复试验之后，选择具有最大内点数量的运动估计，并使用 LeastSquaresFit 来对这一最大内点集合计算两幅图间的变换 M。

[**TODO 5**] getInliers 在给定来自图像 1 和图像 2 的特征集 f1、f2 以及特征匹配对列表 matches 和从图像 1 到图像 2 的变换矩阵 M 的情况下，返回用 M 变换后欧几里德距离低于 RANSACthresh 的所有特征匹配对的索引。

[**TODO 6,7**] LeastSquaresFit 使用先前估计为内点的所有特征匹配对来计算平移或者单应映射的最小二乘估计，返回平移或单应映射矩阵。

2.3 融合并裁剪生成的对齐图像

（文件：blend.py，函数：imageBoundingBox，blendImages，accumulateBlend，normalizeBlend）

[TODO 8]给定图像和变换矩阵，求出应用变换后图像的框。（imageBoundingBox）

[TODO 9]根据每幅图像的边框（使用 imageBoundingBox），计算出最终拼接图像的大小以及它们在全景图中的绝对位移。（getAccSize）

[TODO 10]然后，将每个图像重新映射到其最终位置（此处需要使用反向卷绕），并将其与相邻图像融合。尝试使用简单的羽化功能作为加权函数，请参考讲义（第 11 讲：全景图）“羽化”部分。（accumulateBlend）

提示：

1）使用齐次坐标时，不要忘记在将它们转换回笛卡尔坐标时进行规范化(即除以 z)。

2）反向卷绕时，注意源图像中的黑色像素。不应该把黑色像素加到累积值中。

3）进行反向卷绕时，对源图像像素使用线性插值。

4）首先尝试通过循环每个像素来计算累积值。之后可以使用数组指令和 numpy 技巧（numpy.meshgrid，cv2.remap）来优化代码。

[TODO 11]使用累计的权重通道规范化图像。注意不要除以零。最后记住生成的全景图不应该有权重通道。（normalizeBlend）

[TODO 12]如果是 360 度全景照片，要使左右边缘具有完美的接缝效果。第一幅图像将出现在拼接序列的最左端和最右端（在该图像的中间绘制“切割”线）。对图像使用线性变换以消除第一个和最后一个图像之间的任何垂直“漂移”。这种形为 y’=y+ax 的线性扭曲变换将使得第一图像在左端和右端具有相同的 y 坐标。计算执行此变换所需的“a”值。（blendImages）

3. 使用 GUI

使用图形界面程序 gui.py，可以执行以下操作：

1) **可视化 Homography：**UI 中的第一个选项卡提供了一种加载图像并将任意单应映射应用于图像的方法。例如，你可以可视化手动和编程生成的转换矩阵的结果。
2) **可视化球形扭曲变换：**UI 上的第二个选项卡可以给定的焦距对图像进行球面扭曲。
3) **对齐图像：**第三个选项卡允许你选择具有重叠的两个图像，并使用 RANSAC 计算将右图像映射到左图像的单应映射或平移变换。
4) **生成全景图：**UI 中的最后一个选项卡可用于生成全景图。为了能够创建全景图，你需要有一个文件夹，其中的图像在按字母顺序排序时，为从左到右（或从右到左）的次序，这确保了可以计算所有相邻对之间的映射。我们当前的代码假设全景图中的所有图像都具有相同的宽度。

4. 调试指南

test.py 提供了 TODO 1-9 非常基本的测试用例。你需要自己设计测试用例来测试 TODO 10-12。test.py 提供的测试非常简单，能通过提供的测试用例并不意味着一定能通过最终评分的测试用例。

你也还可以使用 gui.py 可视化来检查程序是否正常运行。

（1）在 campus 测试集中，用于这些示例的相机参数是

f = 595
k1 = -0.15
k2 = 0.00

在 yosemite 测试集中，这些示例的相机参数是

f = 678
k1 = -0.21
k2 = 0.26

（2）请注意，campus 测试集仅适用于平移变换模型！yosemite 图像适用于两种变换模型（平移和单应）。

（3）yosemite 和 campus 测试集中都包含了示例全景图，可将生成的全景图与这些图像进行比较。请注意，使用指定的 f，k1，k2 参数来获取相同的图像非常重要。

（4）campus 测试集可进行 360 度全景图实验。

5. 所提供的文件

warp.py：需要实现的 TODO 1 在此文件中。
alignment.py：需要实现的 TODO 2-7 在此文件中。
blend.py：需要实现的 TODO 8-12 在此文件中。
test.py：包含了对 TODO 1-9 的测试例。
gui.py：可视化界面。
Resources 目录：包含了 yosemite、campus 和 melbourne 三个数据集。

6. 实验环境配置

先在 Linux 或 Windows 上安装 Python3，之后安装 Numpy、Scipy、OpenCV for python。如果要运行 gui.py，还要安装 PIL（Pillow）。

7. 提交文件

warp.py：所实现的 TODO 1 在此文件中。
alignment.py：所实现的 TODO 2-7 在此文件中。
blend.py：所实现的 TODO 8-12 在此文件中。
yosemite_homography.jpg：应用单应映射对 yosemite 数据集生成的全景图。
yosemite_translation.jpg：应用平移变换对 yosemite 数据集生成的全景图。
campus__translation.jpg:应用平移变换对 campus 数据集生成的 360 度全景图。
（可在 gui.py 运行界面中，Panorama 标签页中，点击 ScreenShot 按钮保存所

生成的全景图）

report(.docx,.pdf)：写明以上图片生成时的参数。可选地，你可以对你在项目中做的有趣或别具一格的事情写写感想。报告中应注明两人工作量的比例。

将以上文件打包压缩，压缩后的文件命名为“学号 1+姓名 1+学号 2+姓名 2+expX”;如“16030140095 李乐天+16030140012 王学斌 exp3.rar”、“16030140095 李乐天+16030140012 王学斌 exp3.zip”。Email 的标题类似命名。

8. 额外加分

三种方式可以获得额外加分：

（1）有时，图像之间存在曝光差异，导致最终全景图中的亮度波动。尝试通过在混合图像之前过滤图像来消除此伪影。

（2）尝试拍摄一些物体移动的图像序列。你如何来消除全景图中移动对象的“虚影”？可参考[1]。

（3）实现更好的融合技术，例如金字塔混合[2]，泊松成像混合[3]和图形切割[4]。

如果你希望有额外加分，除提交代码外，在报告中必须详细描述你的改进算法及性能提升结果。

在尝试额外加分之前，你首先应该完成基本任务。基本任务比额外加分更容易得分。

参考文献

[1] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pages 509--516, Kauai, Hawaii, December 2001.

[2] PETER J. BURT and EDWARD H. ADELSON, A Multiresolution Spline With Application to Image Mosaics,ACM Transactions on Graphics, Vol. 2. No. 4, October 1983, Pages 217-236. , http://www.cs.princeton.edu/courses/archive/fall05/cos429/papers/burt_adelson.pdf

[3] Patrick Perez,Michel Gangnet,Andrew Blake,Poisson Image Editing, ACM Transactions on Graphics, 2003 , 22 (3) :313-318 , http://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf

[4] Vivek Kwatra , Arno Schödl , Irfan Essa , Greg Turk and Aaron Bobick,Graphcut Textures: Image and Video Synthesis Using Graph Cuts, ACM Transactions on Graphics (TOG) , 2003 , 22 (3) :277-286. https://www.cc.gatech.edu/cpl/projects/graphcuttextures/

9. 调试注意事项

•生成全景图的时候会非常慢，而且比较吃内存，配置不太好的电脑请把各项数值控制在比较低的水平，否则程序可能会处于长时间的未响应状态。
•测试的时候发现内存跳出了 2G，所以根据经验猜测，32 位的 python 可能无法运行生成全景图的实验。
•由于对同学的提交代码进行测试时将运行在 Linux 系统中，因此不允许使用\t（Tab 键）替代四个空格。嫌敲空格麻烦？最简单的解决方法就是请装一个 PyCharm，Alt+L。
•理论上本次实验中全为浮点数运算，因此 python 2 中的/运算符等同于 python 3 中的/运算符，对代码进行从 python 2 到 3 的转化时没有、也不能进行强制修改。潜藏的 BUG 将取决于你的代码编写质量，如果真出现了什么玄学问题，请从这个方面进行考虑解决方案，比如最简单的方法就是先乘上一个 1.0 再进行运算等。
•raise Exception(....) 什么的请完成 TODO 后注释掉吧，单元测试疯狂 Warning 看着心烦...
•OpenCV 的问题参照上一个实验的改动说明。