实验三：全景图

组员：梁耀宗　李阔

1.实验环境

Python3.8

2.实验内容

1) 将图像变形到球面坐标

2) 提取特征并进行特征匹配

3) 使用 RANSAC 对齐相邻图像对

4) 写出相邻图像对之间的变换矩阵

5) 纠正漂移

6) 将所有图像融合

7) 裁剪结果

3.实现细节

1) 将每张图像变形为球面坐标

按照球面坐标变换公式，将图像映射到球面上，并根据矫正畸变公式矫正径向畸变

```
# TODO-BLOCK-BEGIN

xs = np.sin(xf) * np.cos(yf)
ys = np.sin(yf)
zs = np.cos(xf) * np.cos(yf)

xd = xs / zs
yd = ys / zs

r2 = xd ** 2 + yd ** 2

xt = xd * (1 + k1 * r2 + k2 * r2 ** 2)
yt = yd * (1 + k1 * r2 + k2 * r2 ** 2)

# TODO-BLOCK-END
```

2) 对齐相邻图像对

通过 SVD 奇异值分解计算最佳拟合的单应映射

首先按照求解单应映射矩阵形式填充 A 矩阵，SVD 分解用将合适的元素填充至单应映射矩阵 H 中解得 H。

```
#BEGIN TODO 2
#Fill in the matrix A in this loop.
#Access elements using square brackets. e.g. A[0,0]
#TODO-BLOCK-BEGIN

#将每对匹配点按照求解单应映射矩阵A的形式填入
A[2*i]  = np.array([a_x,a_y,1,0,0,0,-b_x*a_x,-b_x*a_y,-b_x])
A[2*i+1]= np.array([0,0,0,a_x,a_y,1,-b_y*a_x,-b_y*a_y,-b_y])

#TODO-BLOCK-END
#END TODO
```

```
#TODO-BLOCK-BEGIN

h = Vt[Vt.shape[0]-1]
H = np.dot(H,np.array([h[0:3],h[3:6],h[6:9]]))

#TODO-BLOCK-END
```

之后根据 RANSAC 估计两幅图之间的最佳匹配，对于平移变换和单应映射分情况求解。其中，getInliers()获得欧式距离在阈值范围内的匹配点的个数，RANSAC 选择匹配点数最多的匹配，leastSquareFit()根据上诉匹配点计算出图像 1 到图像 2 的变换矩阵。

```
maxinn = []
maxidx = -1
i = 0
if m == eTranslate:

    while i < nRANSAC:
        index_0 = np.random.randint(0, len(matches), 1)
        m = matches[index_0[0]]
        (a_x, a_y) = f1[m.queryIdx].pt
        (b_x, b_y) = f2[m.trainIdx].pt
        H = np.array([[1, 0, b_x - a_x], [0, 1, b_y - a_y], [0, 0, 1]])

        inn = getInliers(f1, f2, matches, H, RANSACthresh)
        if maxidx < inn.__len__():
            maxinn.append(inn)
            maxidx = inn.__len__()
        i += 1

    M = leastSquaresFit(f1, f2, matches, eTranslate, maxinn)
```

```
elif m == eHomography:

    while i < nRANSAC:
        #rand set
        rs = np.random.randint(0, matches.__len__(), 4)

        _matchs = []
        for m_index in rs:
            _matchs.append(matches[m_index])

        H = computeHomography(f1,f2, _matchs)

        inn = getInliers(f1, f2, matches, H, RANSACthresh)
        maxinn = inn if maxidx < inn.__len__() else maxinn
        maxidx = max(maxidx, inn.__len__())

        i += 1
    M = leastSquaresFit(f1, f2, matches, eHomography, maxinn)
```

3) 融合并裁剪生成的对齐图像

根据给定的图像和变换矩阵，求出应用变换后的图像的框，根据边框计算出最终拼接图的大小及他们在全景图中的绝对位移，然后将每个图象重新映射到其最终位置，并将其与相邻图像融合，其中，使用羽化功能作为加权函数。

```
acc_rows, acc_cols, _ = acc.shape
rows, cols = img.shape[:2]
img = cv2.copyMakeBorder(img, 0, acc_rows - rows, 0, acc_cols - cols, cv2.BORDER_CONSTANT, value=0)

row, col, _ = img.shape
x_range = np.arange(col)
y_range = np.arange(row)
(x_grid, y_grid) = np.meshgrid(x_range, y_range)
ones = np.ones((row, col))
coordinates = np.dstack((x_grid, y_grid, ones))
coordinates = coordinates.reshape((col * row, 3))
coordinates = coordinates.T

location = np.linalg.inv(M).dot(coordinates)
location = location / location[2]

map_x = location[0].reshape((row, col)).astype(np.float32)
map_y = location[1].reshape((row, col)).astype(np.float32)
img_warped = cv2.remap(img, map_x, map_y, cv2.INTER_LINEAR)

(minX, minY, maxX, maxY) = imageBoundingBox(img, M)
```

使用简单的羽化功能作为加权函数

```
feather_right = np.clip(np.linspace(-k * minX, k * (acc_cols - 1 - minX), acc_cols), 0, 1).reshape((1, acc_cols,
feather_left = np.ones((1, acc_cols, 1)) - feather_right

img_feathered = feather_right * dst_img
acc *= feather_left

grayimg = cv2.cvtColor(img_warped, cv2.COLOR_BGR2GRAY)
grayacc = cv2.cvtColor(acc[:, :, :3].astype(np.uint8), cv2.COLOR_BGR2GRAY)
maskimg = (grayimg != 0).reshape((acc_rows, acc_cols, 1))
maskacc = (grayacc != 0).reshape((acc_rows, acc_cols, 1))

img_masked = maskimg * img_feathered
acc *= maskacc
acc += img_masked
```

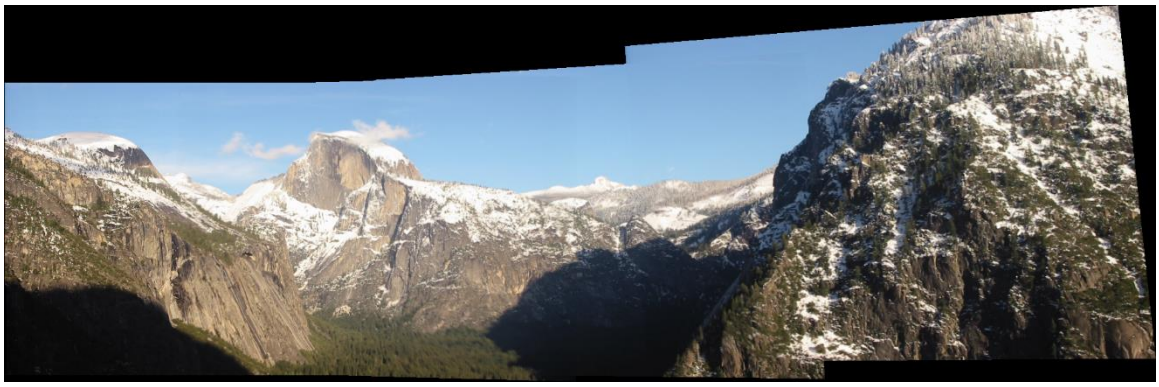
4) 消除漂移

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凭借序列的最左端和最右端。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”

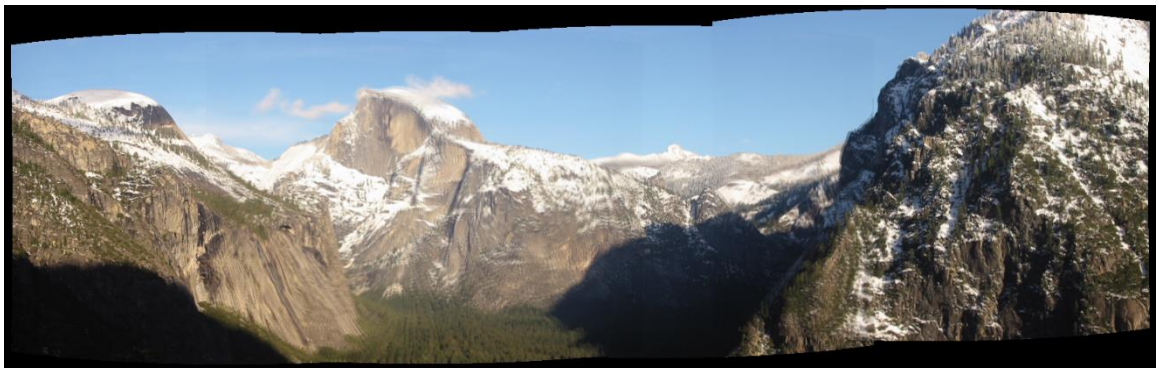
```
#TODO-BLOCK-BEGIN

if is360:
    A = computeDrift(x_init, y_init, x_final, y_final, width)

#TODO-BLOCK-END
# END TODO
```

4.实验结果

应用单应映射对 yosemite 数据集生成的全景图



应用平移变换对 yosemite 数据集生成的全景图



应用平移变换对 campus 数据集生成的 360 度全景图

5.总结与收获

通过本次实验，加深了对 numpy 等常用库的理解与掌握，深刻理解了全景图生成与制作的整个过程，对于过程中使用的矩阵运算和 RANSAC 等算法有了详细的了解，并且加强了自主学习能力与阅读英文文献的能力，此次实验让我受益匪浅。

工作量比例：梁耀宗　18030100139　50%

李阔　18030100224　50%