

实验报告

在本实验中，实现了一个函数 `getHomography`，用于计算两幅图像之间的 Homography 变换。该函数接收两个图像的关键点列表 `KeyPointsA` 和 `KeyPointsB`，以及匹配对 `matches` 和重投影误差阈值 `reprojThresh` 作为输入。首先，将关键点列表转换为 `float32` 类型的 numpy 数组，以便后续计算使用。接着，函数检查匹配点的数量是否超过 4 对，这是计算 Homography 矩阵所需的最低匹配点数。如果匹配点数量足够，函数从匹配对象中提取对应的关键点坐标，并使用 OpenCV 的 `cv2.findHomography` 函数通过 RANSAC（随机抽样一致性）算法计算 Homography 矩阵 `H`。其中，`reprojThresh` 是 RANSAC 算法的重投影误差阈值，用于决定哪些点是内点。最后，函数返回匹配点 `matches`、计算得到的 Homography 矩阵 `H` 和每个匹配点的状态 `status`，状态 `status` 是一个掩码数组，表示哪些点是内点（在计算 Homography 矩阵时被 RANSAC 算法认为是可靠的匹配点）。若匹配点数量不足，函数返回 `None`。通过这种方式，图像关键点和匹配点中计算 Homography 变换，从而实现图像之间的几何变换匹配。

实验效果

将图像之间匹配的特征点画出得到下图，作为单应变换的输入。通过单应变换图像进行拼接，可以得到全景图



