Project - COMP 6341 Computer Vision

Smeet Pareshkumar Patel **40093202**

FEATURE DETECTION AND MATCHING [Steps 1 and 2]:

STEP 1:

Input files:

1. 'image_sets/project_images/Boxes.png'
2. 'image_sets/project_images/Rainier1.png'
3. 'image_sets/project_images/Rainier2.png'

Output files:

1. 'Results/project_images/1a.png' - for Boxes.png
2. 'Results/project_images/1b.png' - for Rainier1.png
3. 'Results/project_images/1c.png' - for Rainier2.png
4. 'Results/project_images/2.png' - matches between key points of ' Rainier1.png' and 'Rainier2.png'.

Run 'feature_detection_and_matching.py' to generate these results.

Steps for Harris Corner detection:

1. Convert image to grayscale image.
2. Compute x and y gradients of the image.
3. Compute I_x^2, I_y^2, and I_xI_y based on earlier step.
4. Apply Gaussian filter to these matrices.
5. Calculate corner response for each pixel.
6. Scale corner responses between 0 and 255.
7. Threshold the corner responses.
8. Perform non-maximum suppression.
9. Compute SIFT descriptors as following:
    a. For a 16 x 16 window around the keypoint, divide the window into windows of 4 x 4.
    b. Calculate dominant orientation in each sub-window and subtract all orientations with the dominant orientation to make the patch rotation invariant.
    c. Create histogram for feature descriptor and threshold normalize it to make it contrast invariant by clipping values to 0.2.
    d. Again normalize it.
    e. Get 128 feature descriptor for keypoint.

STEP 2:
Run ‘feature_detection_and_matching.py’ to generate these results.
As mentioned above, 2.png is generated by running this script.

Steps for getting matches:

1. Get two argument images between which matches are to be found.
2. For all key points of image 1:
    a. Calculate the sum of squared distance of its descriptor with all the descriptors of all key points of image 2.
    b. Record matches smaller than the threshold value.
    c. Perform ratio test between the best two matches, i.e. matches with minimum distance.
    d. Record the match that passes the ratio test.

STEP 3:
Input files:

1. ‘image_sets/project_images/Rainier1.png’
2. ‘image_sets/project_images/Rainier2.png’

Output files:

1. ‘Results/project_images/2.png’ - matches between key points of ‘ Rainier1.png’ and ‘Rainier2.png’.
2. ‘Results/project_images/3.png’ - only inlier/correct matches between key points of ‘ Rainier1.png’ and ‘Rainier2.png’.

Run ‘ransac_check.py’ to generate these results.

Steps for RANSAC:
Preliminary step: Calculate Harris corners and find matches between images as discussed above.

1. For said number of iterations do the following steps:
    a. Select 4 random matches
    b. Calculate homography for these matches.
    c. Calculate inlier count for this homography as follows:
        i. Project every point as follows:
            1. x2 = H[0][0]*x1 + H[0][1]*y1 + H[0][2]
            2. y2 = H[1][0]*x1 + H[1][1]*y1 + H[1][2]
            3. w = H[2][0]*x1 + H[2][1]*y1 + H[2][2]
            4. x2 /= w
            5. y2 /= w

        6. Return x2 and y2.
    ii. Calculate distance between the projected point as per the homography and actual coordinates of those points.
    iii. If the distance is lesser than the threshold distance, increase the inlier count.
  d. For the homography with maximum inliers, calculate new homography with all the points rather than just 4 points.
  e. Calculate inverse homography.

<u>STEP 4:</u>

Input files:

1. 'image_sets/project_images/Rainier1.png'
2. 'image_sets/project_images/Rainier2.png'

Output files:

1. 'Results/project_images/4.png' - Stitched image of 'Rainier1.png' and 'Rainier2.png'.

Run <u>stitch_check.py'</u> to generate these results.

Steps for stitching:

Preliminary step: Calculate Harris corners, find matches between images, and apply RANSAC as discussed above.

1. Project all four corners of image 2 using inverse homography.
2. Calculate the size of stitched images based on the projections calculated in step 1 with zeros as default value.
3. Copy Image 1 at the right place of stitched image. If the projection of original coordinates of image 1 lies within the boundary of image 2 using calculated homography, fill the stitched image with the value of image 2 at the projection. Else, fill the stitched image with the values of image 1.
4. For each pixel of stitched image, find the projected pixel in image 2 using the calculated homography. If it falls in the boundary of image 2, fill the stitched image pixel with the value of the projected point in image 2. Else, leave it be.

<u>STEP 5:</u>

Input files:

1. 'image_sets/project_images/Rainier1.png'
2. 'image_sets/project_images/Rainier2.png'
3. 'image_sets/project_images/Rainier3.png'
4. 'image_sets/project_images/Rainier4.png'
5. 'image_sets/project_images/Rainier5.png'
6. 'image_sets/project_images/Rainier6.png'

Output files:

1. ‘Results/project_images/All_Rainier_stitched.png’ - Stitched image of all the Rainier images.

Run ‘<u>rainier_stitch.py’</u> to generate these results.

Steps for stitching multiple images:

Preliminary step: Calculate Harris corners, find matches between images, and apply RANSAC as discussed above.

1. Select the first image from the given set of images.
2. For the remaining images do the following:
    a. Find matches, apply RANSAC, and calculate the inliers between the first image or stitched image till this point and the remaining images.
    b. Stitch the image with which maximum inlier matches are found.
    c. Calculate interest points in the new stitched image.
    d. Repeat Step 2 with the newly stitched image and the remaining images to be stitched.
3. Save the final stitched image.

<u>STEP 6:</u>

Stitching of images captured by me.

Input files:

1. ‘image_sets/misc/personal_1.png’
2. ‘image_sets/misc/personal_2.png’
3. ‘image_sets/misc/personal_3.png’

Output files:

1. ‘Results/project_images/personal_images_stitched.png’ - Stitched image of all the personal images.

Run ‘<u>personal_image_stitch.py’</u> to generate these results.

Same steps are followed for stitching these images as discussed above in **<u>STEP 5.</u>**