Computer Vision (CO462) Assignment 2
Photo Stitching

Vignesh K (14CO252)

Chosen Images

Assignment_2/Input_Images/build_left.jpg
Assignment_2/Input_Images/build_right.jpg

The approximate area of overlap of the two images is around 33% (one-third).

Methodology

The following steps have been adopted to stitch the left and right perspectives of the chosen buildings:

- The right side image is chosen as the destination image (the image whose perspective is preserved) and the left side image (the other perspective) is transformed to exist in this destination coordinate system. In other words, the left side image is the source image.

- The destination image is embedded in a bigger image of zeros (black image) so as to accommodate the transform of the source image after stitching both of them together.

- Both the images are then **temporarily** blurred with a **5 x 5 Gaussian Blur** to enhance the performance of the Harris Corner Detector.

- Harris corner points are detected from both the images with the following parameters: **threshold = 0.01 x max(R-score), neighborhood = 3 x 3 , k (free-parameter) = 0.04, Gaussian weights**
  We then retain **the original non-blurred images** for further use.

- **26 Harris corner points** common to both the images are chosen manually for computing the homography matrix of the perspective transformation.
- Following this the homography matrix is computed using two approaches:

  - Using the OpenCV built-in function for finding the homography matrix given the pairs of corresponding corner points in both the coordinate systems.

  - Using a user-defined function which constructs the matrix ***A*** formed by the pairs of corresponding corner points in both the coordinate systems and then computes the following:

    SVD (*A*) = *UL*(*V*^T)

    The last column of ***V*** is reshaped into a **3 x 3** matrix to get the homography matrix ***H***.

  Both the above approaches yield approximately equal matrices for ***H.***

- To compute the transform of the source image, two approaches have been used as described below:

  - The homography matrix ***H*** is multiplied with each pixel coordinate of the source image to get its transformed coordinates in the destination coordinate system. Only those pixels are retained which have valid coordinates in the new coordinate system i.e. **x belongs to [0, M - 1]** and **y belongs to [0, N - 1]** where the resolution of the destination image embedded in the bigger image of zeros is **M x N**. A relatively large value of M and N is chosen so that we can accommodate most of the pixels of the transformed source image in the destination coordinate system and only few will be discarded.

  - The second approach is using the built-in function of OpenCV which computes the transform of the source image given the homography matrix ***H*** and the dimensions of the destination coordinate system.

- After computing the perspective transformation of the source image, we now have two images which are in the same coordinate system, i.e. the destination coordinate system.

- These two images (the transform of the source image and the destination image) are then superimposed to get the stitched image in the destination coordinate system. We then mark the common corner points on the stitched image.

- The stitched image is cropped to get a rectangular shape.

Results

The common corner points have been marked with red circles on the original images, perspective changed image and stitched image as directed below:

Original images with corners:
Assignment_2/Output_Images/build_left_harris_26_corners.jpg
Assignment_2/Output_Images/build_right_harris_26_corners.jpg

Perspective changed image with corners:
With OpenCV: Assignment_2/Output_Images/warp_corners_with_opencv.jpg
Without OpenCV: Assignment_2/Output_Images/warp_corners_without_opencv.jpg

Stitched image with corners:
With OpenCV:
Assignment_2/Output_Images/build_stitched_with_opencv.jpg
Assignment_2/Output_Images/build_stitched_with_opencv_cropped.jpg (CROPPED)
Without OpenCV:
Assignment_2/Output_Images/build_stitched_without_opencv.jpg
Assignment_2/Output_Images/build_stitched_without_opencv_cropped.jpg (CROPPED)

Analysis

While the **computed homography matrix is the same in both the approaches** (with OpenCV and without OpenCV) as is evident from the transformed perspectives of the source images in both the approaches, **the continuity of the resultant transform is NOT**. The approach with OpenCV gives a better continuity in the transformed source image as compared to the approach without OpenCV. This is on account of the following. The approach without OpenCV relies on computing the transform by simple multiplication of each coordinate of the source image with the Homography matrix $\boldsymbol{H}$. But there is **NO guarantee that the transformed coordinates will be adjacent to each other** as in the original source image.

This is especially the case when the **transform stretches the original source image** thus covering more pixels in its area than before. If the area of the transformed image is more, it is quite easy to see that **a one-to-one mapping from the pixels of the original source image to the pixels of the transformed image is not enough to cover the entire area of the transformed image simply because we do not have so many pixels in the original source image.** To put it concretely, **there are pixels in the area of the transformed image which do not have a pre-image in the original source image. The transformation is NOT onto with respect to the area of the transformed image.** Therefore we end up seeing black gaps in the perspective transformed image. Even a simple **5 x 5 nearest neighbour interpolation** or a **blur of the transformed image** does not seem to improve the result of this approach. The best possible result using known methods is given by a **morphological closing operation** applied on the image with a **3 x 3 kernel**. Bigger kernels affect the overlapping region significantly thus disturbing the stitched image.

Result of Morphological Closing with a 3 x 3 kernel:
Assignment_2/Output_Images/warp_without_opencv_closed.jpg

Hence a non-trivial interpolation method is needed to fill up these black gaps. Another approach to reduce this artefact would be to increase the area of overlap between the two images so that the computed perspective transform is more or less in a similar plane (separated by a small angle) as the destination image plane.

The approach using OpenCV to compute the transformed source image given the homography matrix performs better due to a **robust interpolation algorithm**. The interpolation algorithm used by this function yields a satisfactory blur of the perspective transformed source image and is the **best attained result of this assignment.**