

**BBM 418 - Computer Vision Laboratory
Programming Assignment-2 Report**

Spring 2020
Assoc. Prof. Dr. Nazlı İkizler Cinbis

**Arda Hüseyinoğlu
21627323**

Department of Computer Engineering, Hacettepe University
April 24, 2020

1 Explanation

First of all, all directory names which are under the directory of HW2 Dataset are obtained. Then, first and second image which are under the first panorama directory is read and they are merged. If there are not enough matches to create homography matrix (since BF matcher algorithm using keypoints, which is found by ORB, to find matchings did not find more than 4 good matches in Pano5, I need to implement this error handling) since more than 4 good matches are needed, then `merge` function is returned with an error message and next panorama images(directory) are taken. If there is no problem, the first two images are merged. Then merged image is started to be merged with the next images ($img12 + img3 = img123\dots$). Same possible error is also checked here. If there are more than 4 good matches, images are merged, and next image are taken.

After the first two image are merged -since it should be guaranteed that there will be no such distortion in the first merging-, an input is taken from user. If user does not want to proceed to create panorama for current one due to distortions, he/she presses 'n' ,which is standing for 'next', and Enter, and it will be started to create next panorama and he/she presses any key and Enter to proceed to create current panorama. (The number of images that I can merge is really depend upon the image set belonging to panorama directory. Some panoramas that I get includes 3 sub images and some of them includes 6. So, If the user feels that the panorama which is created so far have some distortions and it will be worse, then he/she may want to make it stop and proceed with the next panorama directory.)

In the `merge` function, left and right images are firstly converted to the gray scale. Then, ORB is used for feature extraction. After getting key points and their descriptors, keypoints on each image are plotted. After that, brute-force matcher (BF matcher) is used for feature matching. It takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned. All matches found is sorted and last 50 matches is stored as good matches. After that, good matching points are plotted.

After getting good matches, their corresponding coordinates for left and right images are obtained: Two empty numpy array is created with 2 columns(x,y) and the rows with the number of good matches. Then, each matching is iterated. By using `queryIdx` and `trainIdx`, descriptor index of each matching is obtained for each image. Then, coordinates of good matching points are obtained by using keypoint that has same index with descriptor index.

After getting good matching coordinates in left and right images, `findHomographyMatrix` is called to find homography matrix. Bu using RANSAC algorithm, 4 random integer value is determined via `random.sample`. Then, 4 coordinates for right and left image is obtained by using these random integer values as index. Then, matrix 'A' is created manually to find homography matrix from the equation of $A.h = 0$. From the fact that is "the eigen-decomposition of $A^T.A$, we see that h should equal the eigenvector of $A^T.A$ that has an eigenvalue of zero (or, in the presence

of noise the eigenvalue closest to zero), which means the minimum one.”.

After this calculation, homography matrix is obtained in vector form. So, it is reshaped as 3x3. Then, error between coordinates of good matched points on right image and coordinates of good matched points on left image which are multiplied by homography matrix is calculated: Numpy matrix of good matched points on left image is transposed ($..x2- > 2x..$) and ‘ones’ row is added so that we can do matrix multiplication. After getting transformed coordinates, all coordinates are divided by its third element so that we can fix the third element as 1 ($[w * x, w * y, w] - > [x, y, 1]$). Then, ‘ones’ column is added to the numpy matrix of good matched points on right image to have same shape with the transformed coordinates matrix that we obtained before.

After that, distance between each vector(corresponding coordinates on image left and image right) is calculated by using these matrices (Euclidian distance is used when calculating.). After getting all error values for each coordinates, these error values are summed up to represent an error value for corresponding homography matrix. If total error sum is smaller than the previous sums, then it is assigned as minimum error sum that is going to be used for next comparisons and homography matrix is upgraded to current one. This process is repeated N times to find best fitting homography matrix (in my implementation, this N is 50.)

After getting the best fitting homography matrix, `warpAndMerge` function is called to implement to homography matrix to right image to get proper perspective, and to merge left and right images. Firstly a free image space is created. Its size is height of left image x (width of left image + width of right image). Then, left image is placed to the leftmost position of that free image space. After that, each possible coordinate values for right image is iterated in nested loops. Each coordinate is multiplied by the inverse of the homography matrix (if we multiplied by non-inverse homography matrix, then the right image will be transformed to the left: out of the free space coordinates and it will overlap with left image). After getting transformed coordinate, the coordinate is divided by its third element so that we can fix the third element as 1 ($[w * x, w * y, w] - > [x, y, 1]$). Then the transformed coordinate is checked if it exceeds the borders of the free image space. If not, then color value for corresponding coordinate is assigned to that transformed coordinate on free image space. Lastly, black pixels taking part in the right part of the merged image is removed.

2 Result

Figure 1: Pano1 - Feature Points of Img0 and Img1

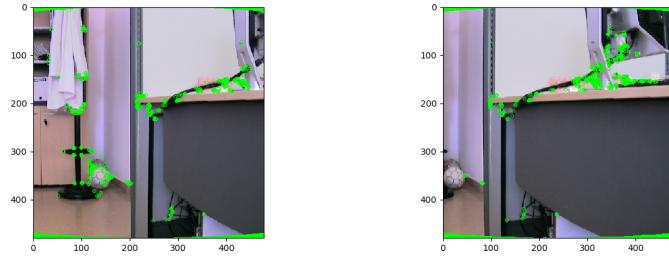


Figure 2: Pano1 - Feature Point Matching Lines of Img0 and Img1

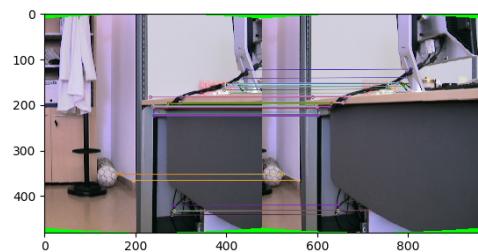


Figure 3: Pano1 - Subpanorama of Img0 and Img1

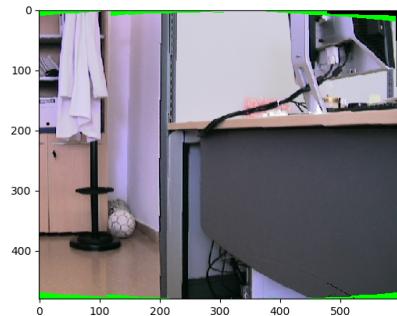


Figure 4: Pano1 - Feature Points of Img01 and Img2

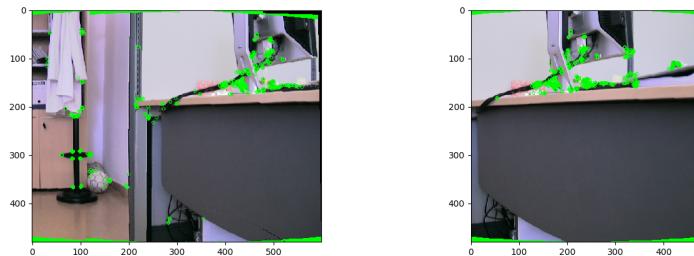


Figure 5: Pano1 - Feature Point Matching Lines of Img01 and Img2

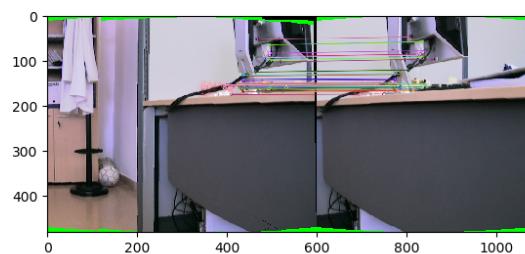


Figure 6: Pano1 - Subpanorama of Img01 and Img2

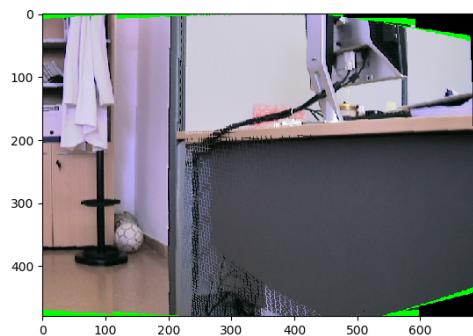


Figure 7: Pano1 - Feature Points of Img012 and Img3

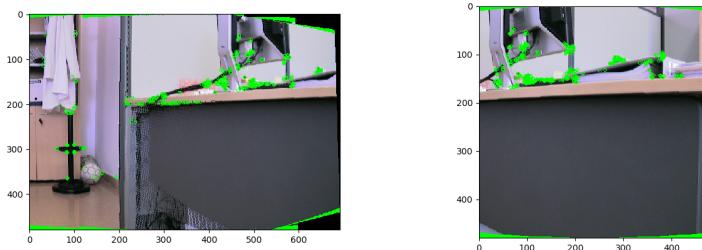


Figure 8: Pano1 - Feature Point Matching Lines of Img012 and Img3

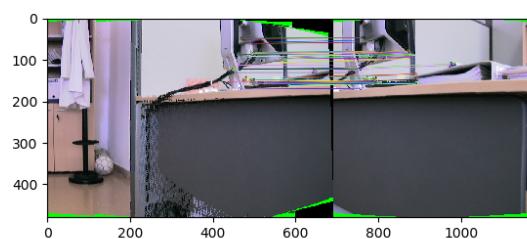


Figure 9: Pano1 - Subpanorama of Img012 and Img3

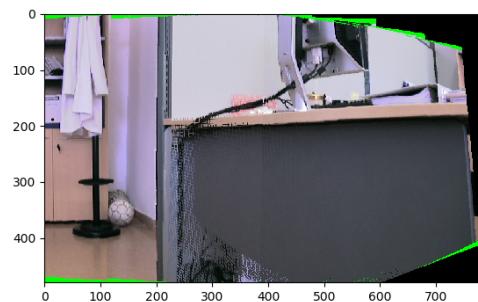


Figure 10: Pano1 - Feature Points of Img0123 and Img4

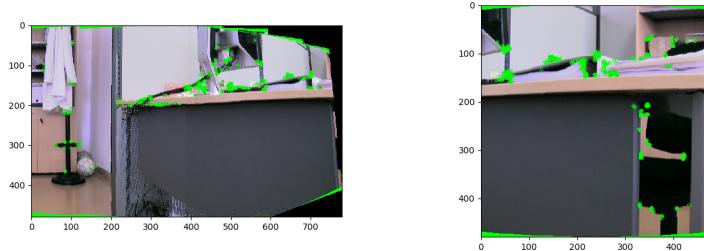


Figure 11: Pano1 - Feature Point Matching Lines of Img0123 and Img4

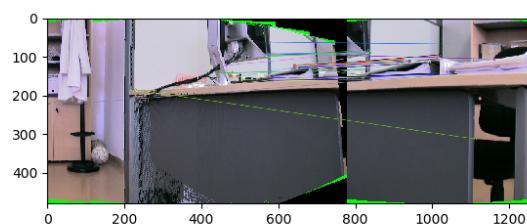


Figure 12: Pano1 - Subpanorama of Img0123 and Img4

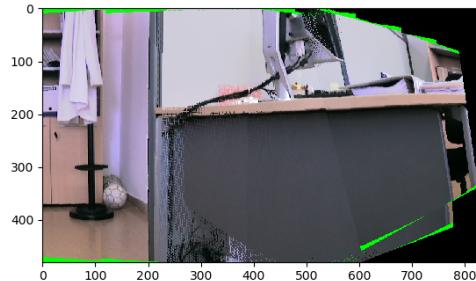


Figure 13: Pano1 - Ground Truth Panorama



Figure 14: Pano2 - Feature Points of Img0 and Img1

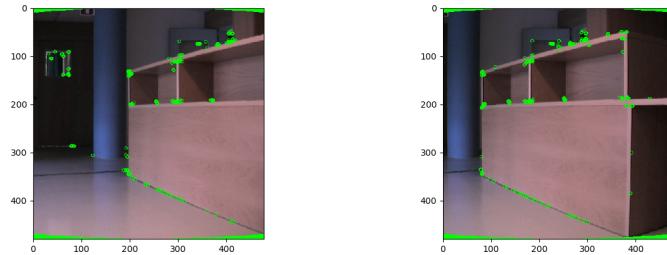


Figure 15: Pano2 - Feature Point Matching Lines of Img0 and Img1

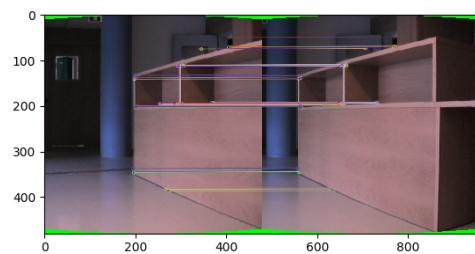


Figure 16: Pano2 - Subpanorama of Img0 and Img1

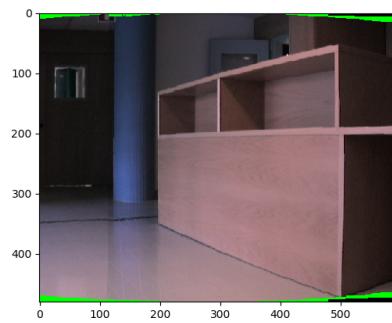


Figure 17: Pano2 - Feature Points of Img01 and Img2

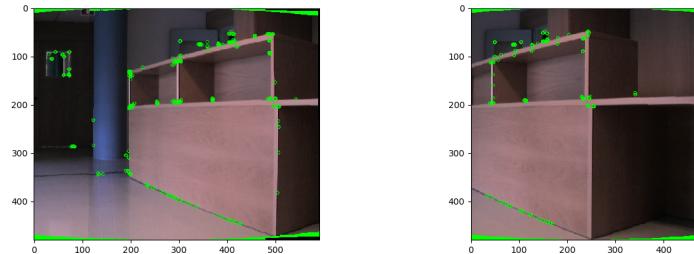


Figure 18: Pano2 - Feature Point Matching Lines of Img01 and Img2

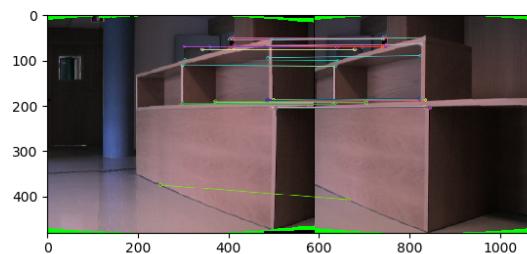


Figure 19: Pano2 - Subpanorama of Img01 and Img2

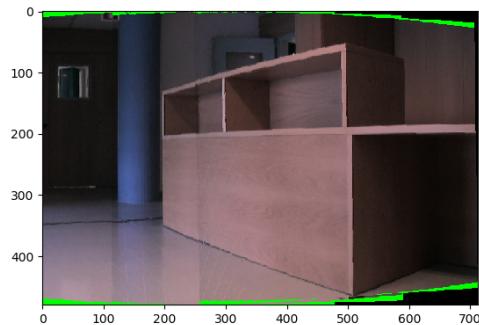


Figure 20: Pano2 - Ground Truth Panorama



Figure 21: Pano3 - Feature Points of Img0 and Img1

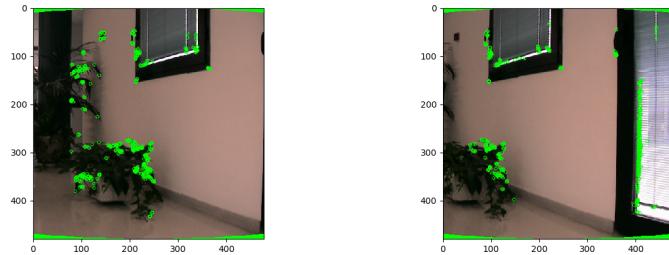


Figure 22: Pano3 - Feature Point Matching Lines of Img0 and Img1

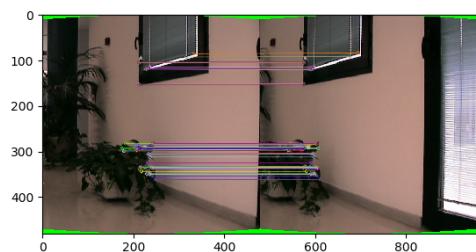


Figure 23: Pano3 - Subpanorama of Img0 and Img1

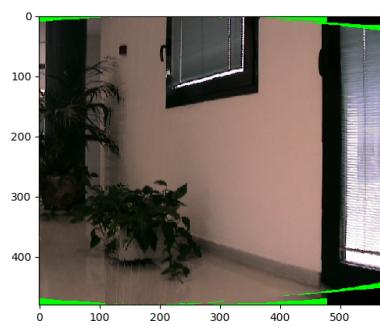


Figure 24: Pano3 - Feature Points of Img01 and Img2

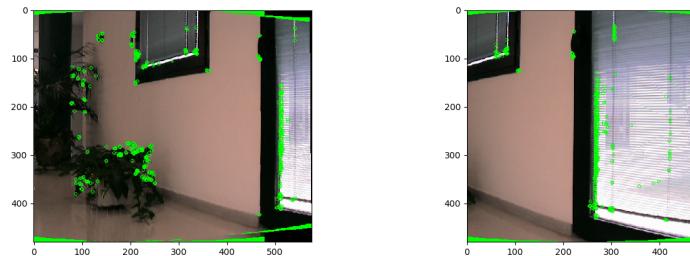


Figure 25: Pano3 - Feature Point Matching Lines of Img01 and Img2

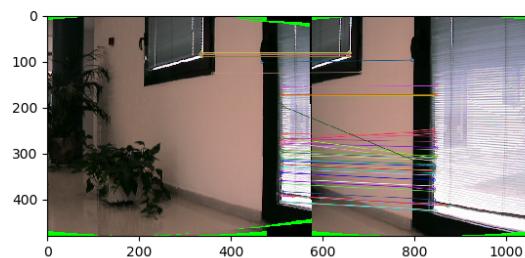


Figure 26: Pano3 - Subpanorama of Img01 and Img2

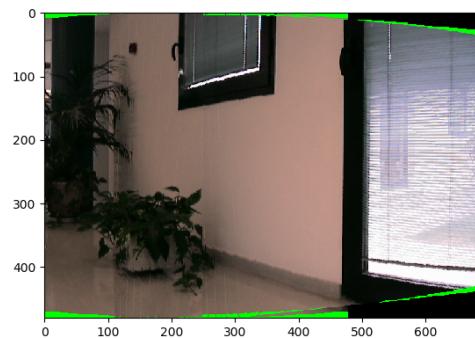


Figure 27: Pano3 - Feature Points of Img012 and Img3

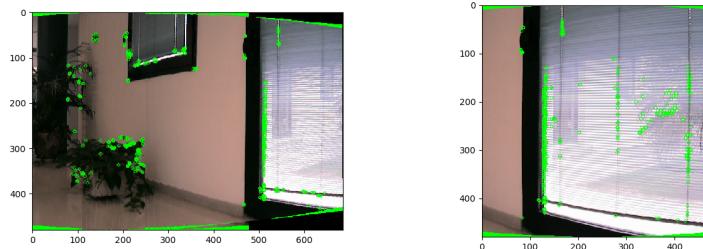


Figure 28: Pano3 - Feature Point Matching Lines of Img012 and Img3

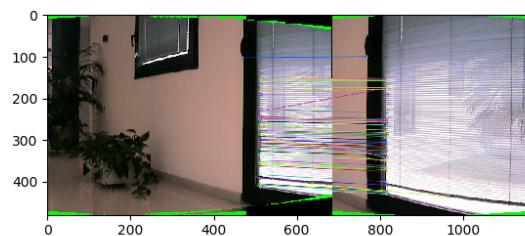


Figure 29: Pano3 - Subpanorama of Img012 and Img3



Figure 30: Pano3 - Ground Truth Panorama



Figure 31: Pano4 - Feature Points of Img0 and Img1

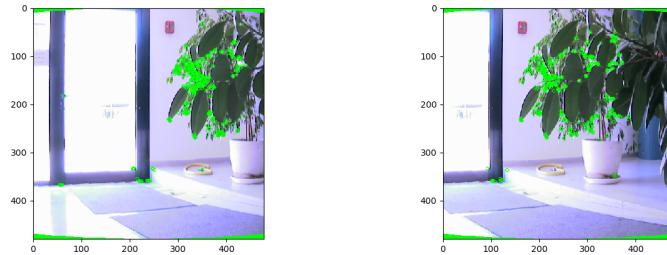


Figure 32: Pano4 - Feature Point Matching Lines of Img0 and Img1

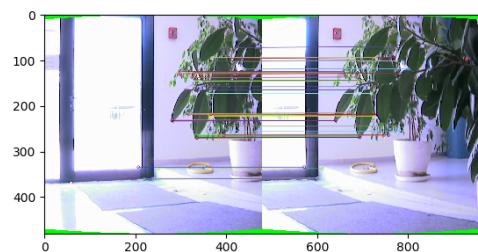


Figure 33: Pano4 - Subpanorama of Img0 and Img1

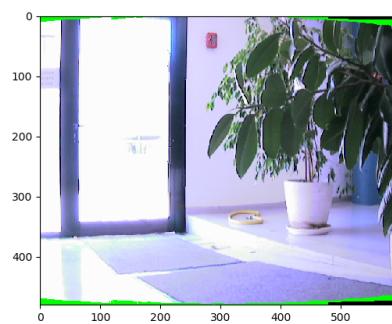


Figure 34: Pano4 - Feature Points of Img01 and Img2

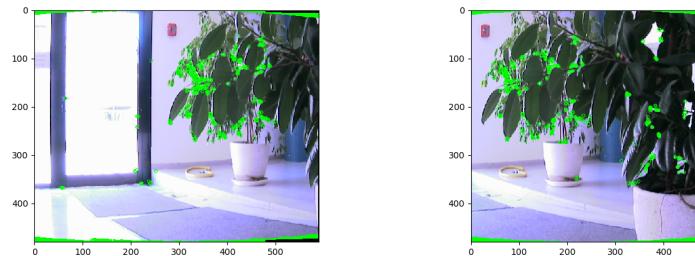


Figure 35: Pano4 - Feature Point Matching Lines of Img01 and Img2

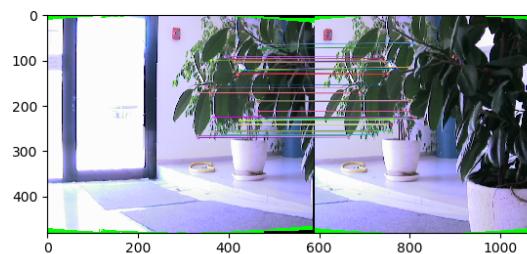


Figure 36: Pano4 - Subpanorama of Img01 and Img2

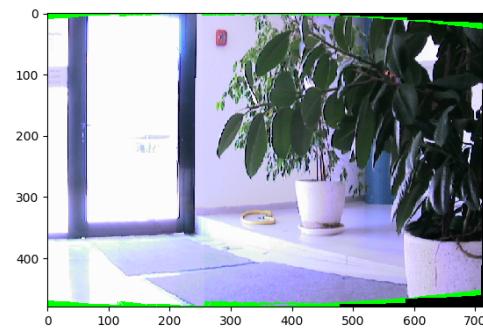


Figure 37: Pano4 - Feature Points of Img012 and Img3

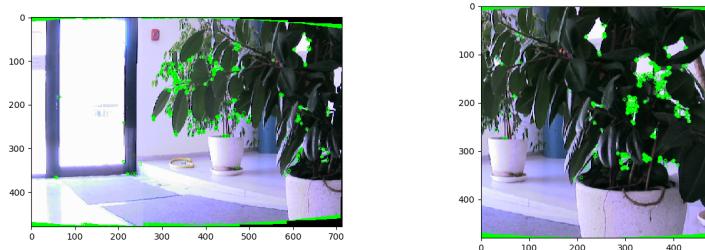


Figure 38: Pano4 - Feature Point Matching Lines of Img012 and Img3

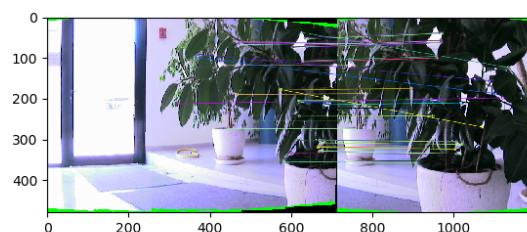


Figure 39: Pano4 - Subpanorama of Img012 and Img3



Figure 40: Pano4 - Feature Points of Img0123 and Img4

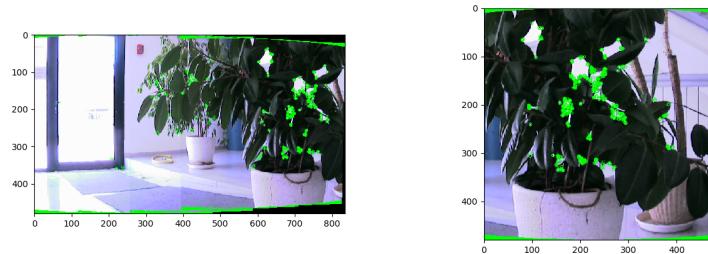


Figure 41: Pano4 - Feature Point Matching Lines of Img0123 and Img4

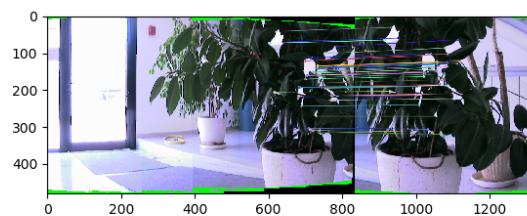


Figure 42: Pano4 - Subpanorama of Img0123 and Img4

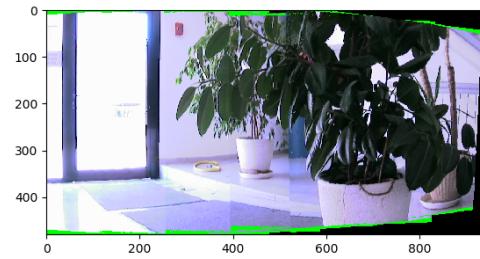


Figure 43: Pano4 - Feature Points of Img01234 and Img5

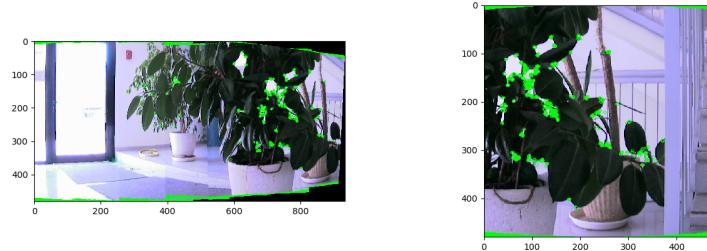


Figure 44: Pano4 - Feature Point Matching Lines of Img01234 and Img5

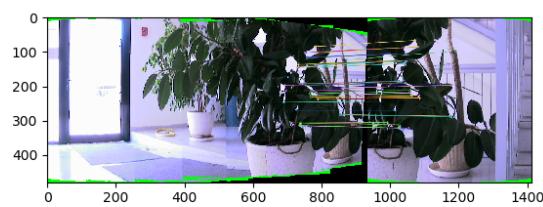


Figure 45: Pano4 - Subpanorama of Img01234 and Img5

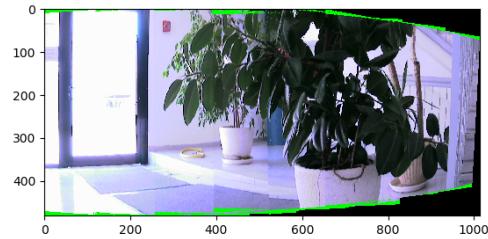


Figure 46: Pano4 - Ground Truth Panorama



Figure 47: Pano5 - Feature Points of Img0 and Img1

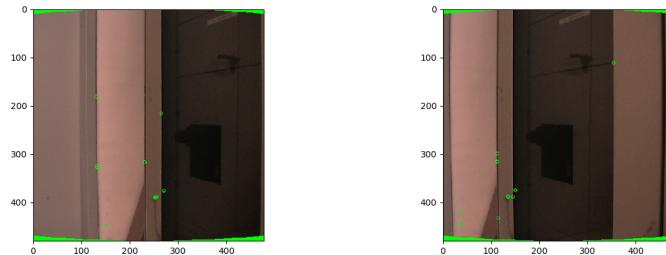


Figure 48: Pano5 - Feature Point Matching Lines of Img0 and Img1

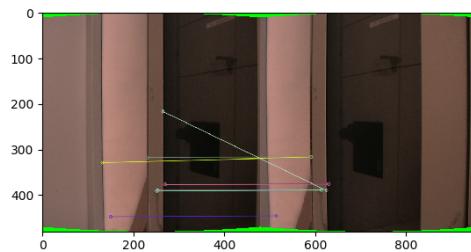


Figure 49: Pano5 - Subpanorama of Img0 and Img1

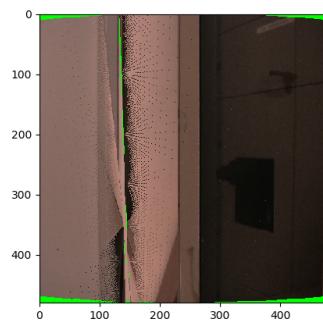


Figure 50: Pano5 - Ground Truth Panorama



Figure 51: Pano6 - Feature Points of Img0 and Img1

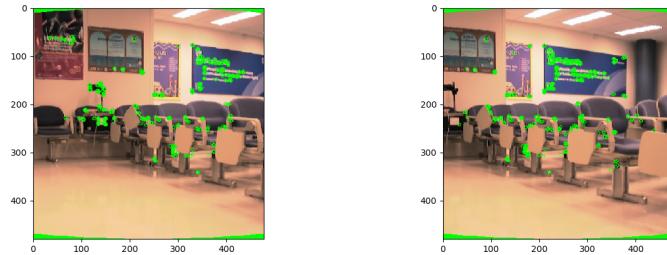


Figure 52: Pano6 - Feature Point Matching Lines of Img0 and Img1

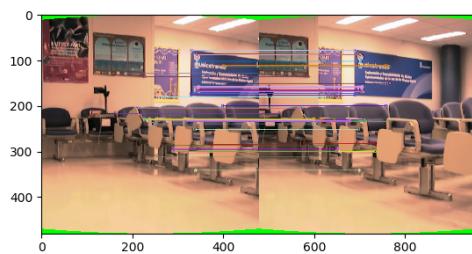


Figure 53: Pano6 - Subpanorama of Img0 and Img1



Figure 54: Pano6 - Feature Points of Img01 and Img2

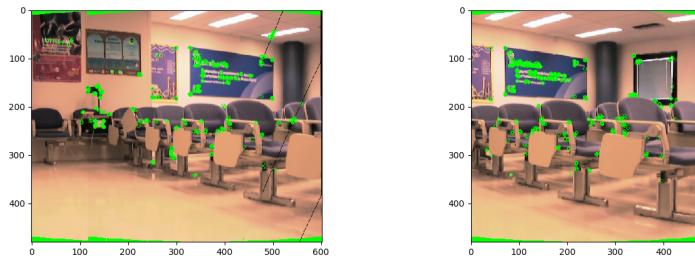


Figure 55: Pano6 - Feature Point Matching Lines of Img01 and Img2

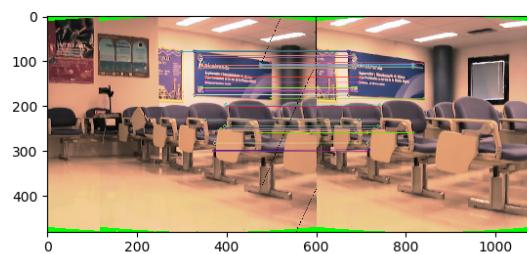


Figure 56: Pano6 - Subpanorama of Img01 and Img2



Figure 57: Pano6 - Feature Points of Img012 and Img3

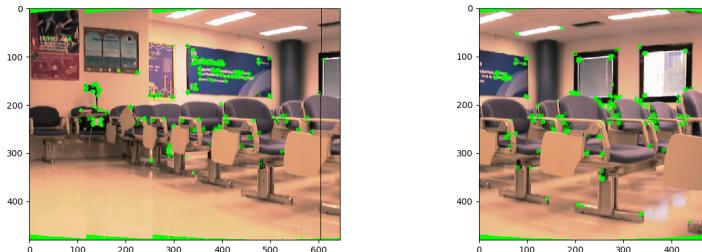


Figure 58: Pano6 - Feature Point Matching Lines of Img012 and Img3

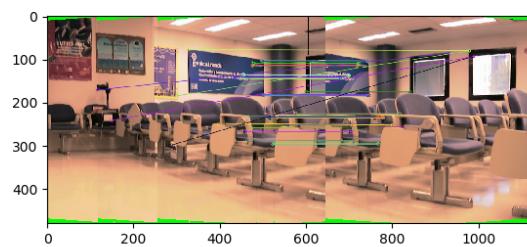


Figure 59: Pano6 - Subpanorama of Img012 and Img3

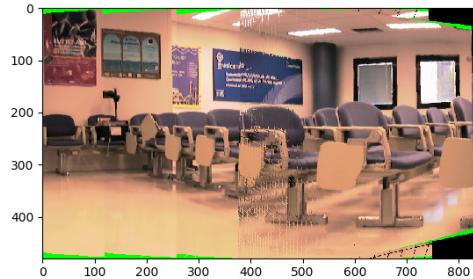


Figure 60: Pano6 - Feature Points of Img0123 and Img4

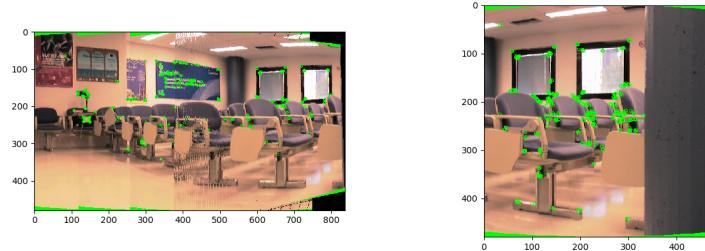


Figure 61: Pano6 - Feature Point Matching Lines of Img0123 and Img4

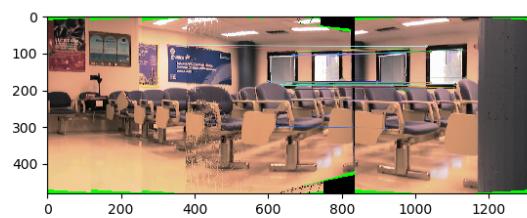


Figure 62: Pano6 - Subpanorama of Img0123 and Img4

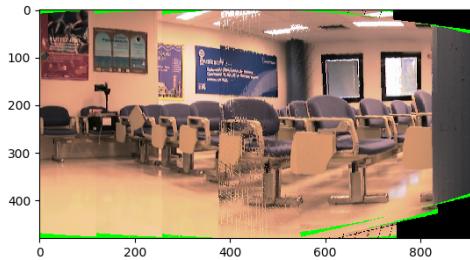


Figure 63: Pano6 - Ground Truth Panorama



Figure 64: Pano7 - Feature Points of Img0 and Img1

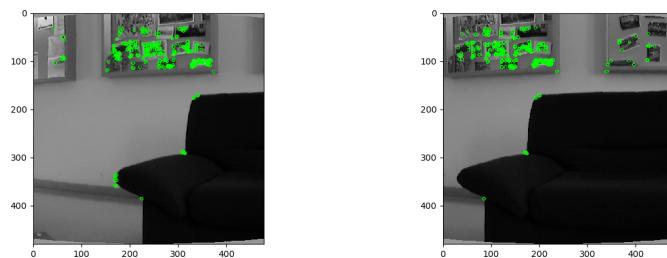


Figure 65: Pano7 - Feature Point Matching Lines of Img0 and Img1

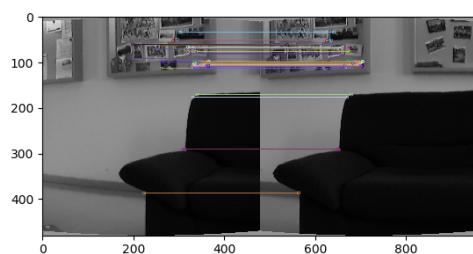


Figure 66: Pano7 - Subpanorama of Img0 and Img1

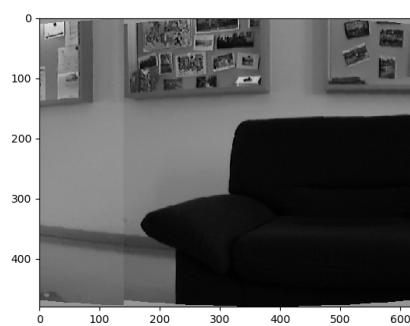


Figure 67: Pano7 - Feature Points of Img01 and Img2

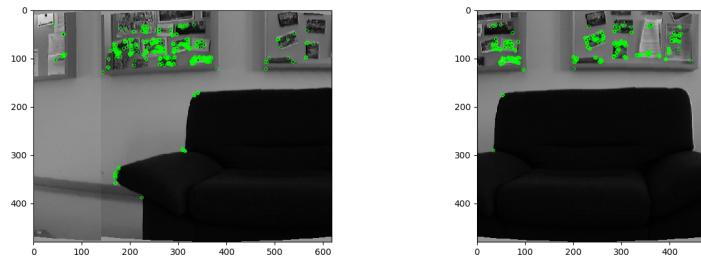


Figure 68: Pano7 - Feature Point Matching Lines of Img01 and Img2

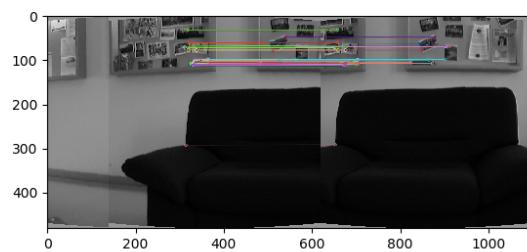


Figure 69: Pano7 - Subpanorama of Img01 and Img2

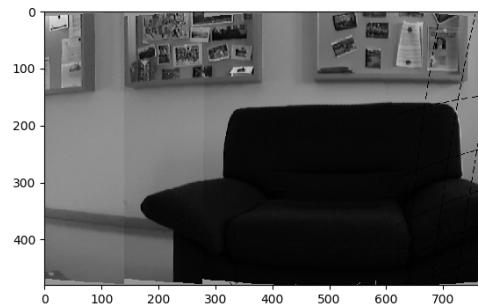


Figure 70: Pano7 - Feature Points of Img012 and Img3

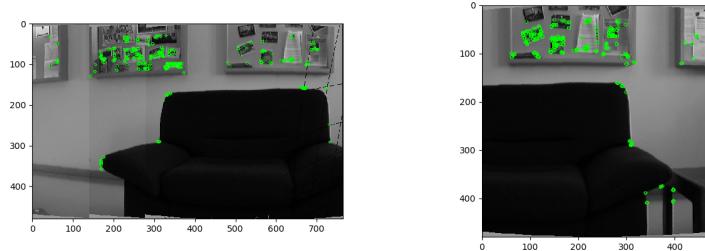


Figure 71: Pano7 - Feature Point Matching Lines of Img012 and Img3

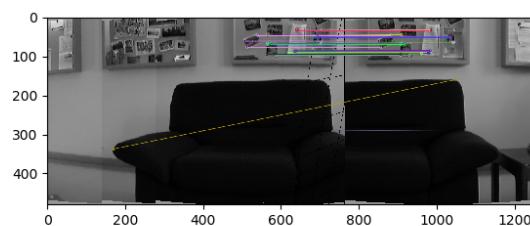


Figure 72: Pano7 - Subpanorama of Img012 and Img3

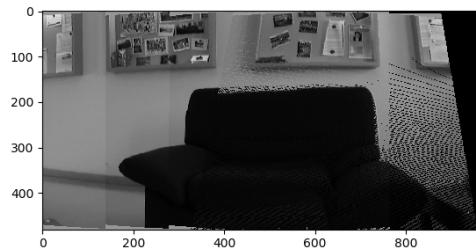


Figure 73: Pano7 - Ground Truth Panorama



Figure 74: Pano8 - Feature Points of Img0 and Img1

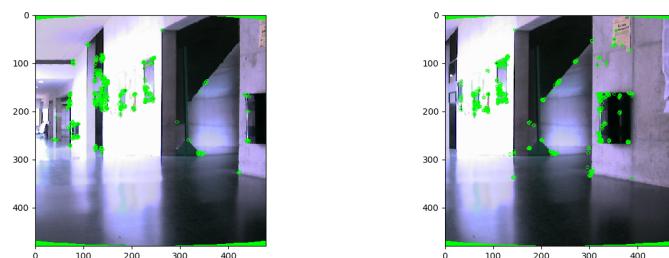


Figure 75: Pano8 - Feature Point Matching Lines of Img0 and Img1

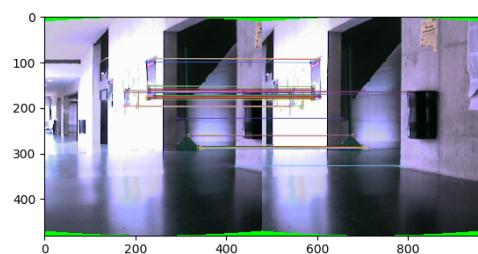


Figure 76: Pano8 - Subpanorama of Img0 and Img1

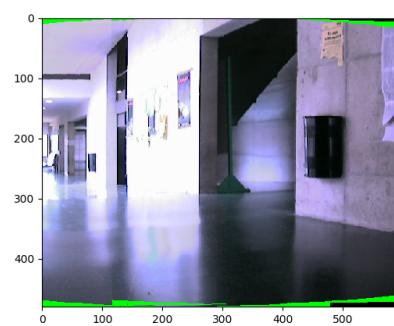


Figure 77: Pano8 - Feature Points of Img01 and Img2

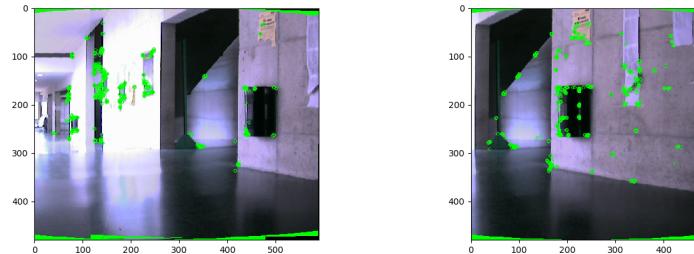


Figure 78: Pano8 - Feature Point Matching Lines of Img01 and Img2

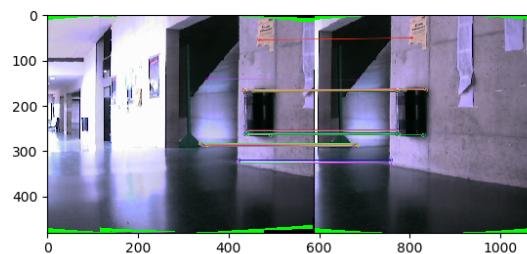


Figure 79: Pano8 - Subpanorama of Img01 and Img2

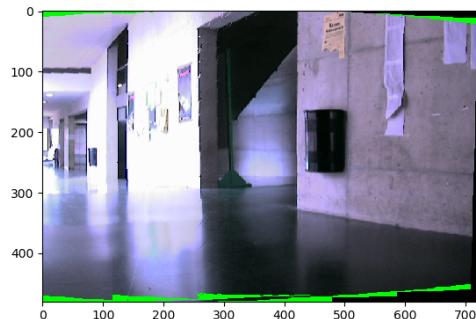


Figure 80: Pano8 - Ground Truth Panorama



Figure 81: Pano9 - Feature Points of Img0 and Img1

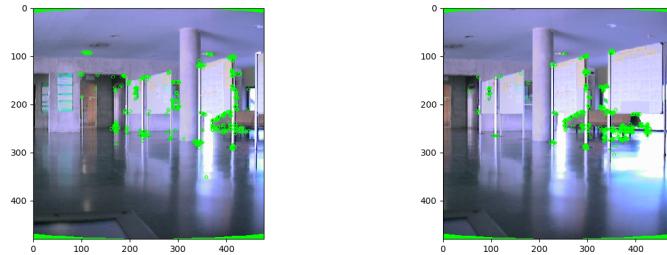


Figure 82: Pano9 - Feature Point Matching Lines of Img0 and Img1

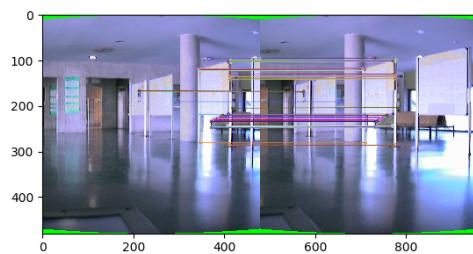


Figure 83: Pano9 - Subpanorama of Img0 and Img1

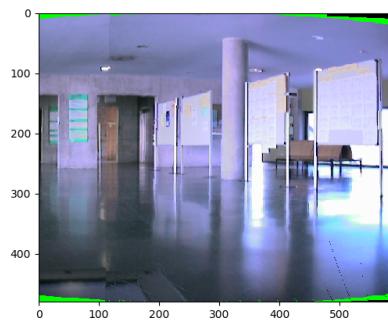


Figure 84: Pano9 - Feature Points of Img01 and Img2

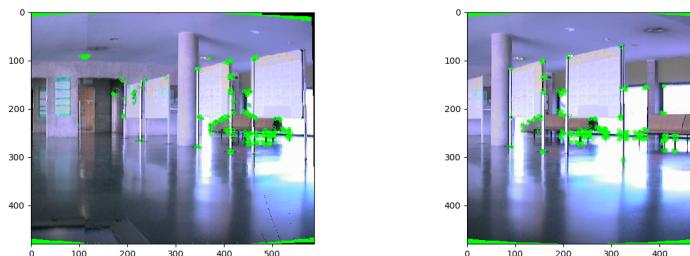


Figure 85: Pano9 - Feature Point Matching Lines of Img01 and Img2

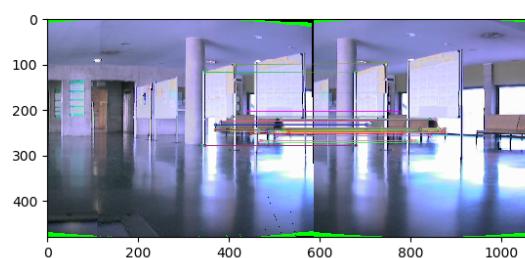


Figure 86: Pano9 - Subpanorama of Img01 and Img2

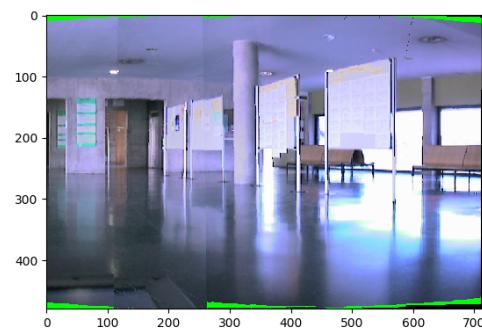


Figure 87: Pano9 - Feature Points of Img012 and Img3

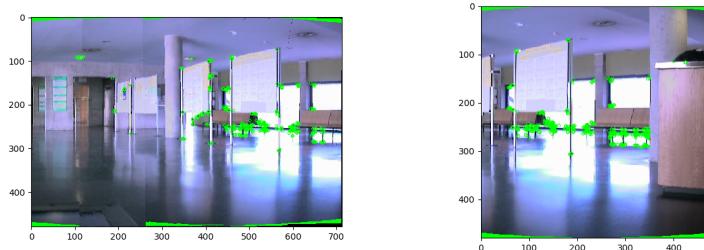


Figure 88: Pano9 - Feature Point Matching Lines of Img012 and Img3

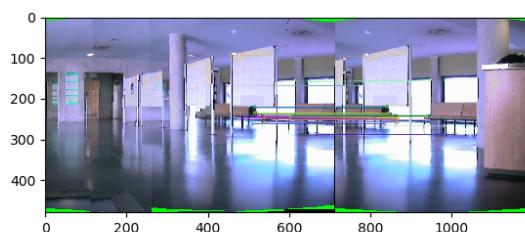


Figure 89: Pano9 - Subpanorama of Img012 and Img3

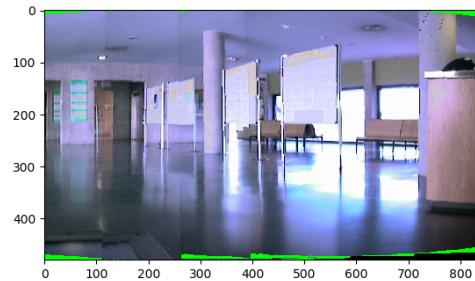


Figure 90: Pano9 - Feature Points of Img0123 and Img4

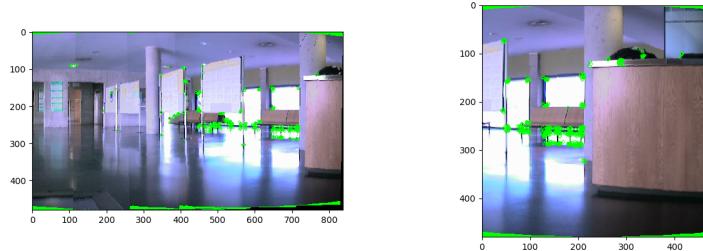


Figure 91: Pano9 - Feature Point Matching Lines of Img0123 and Img4

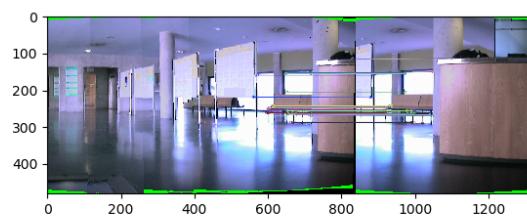


Figure 92: Pano9 - Subpanorama of Img0123 and Img4

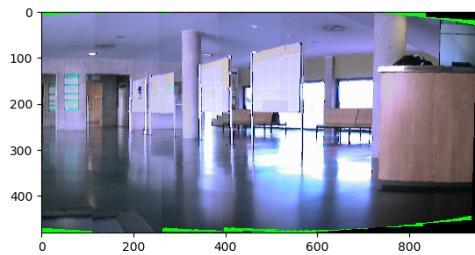


Figure 93: Pano9 - Ground Truth Panorama

