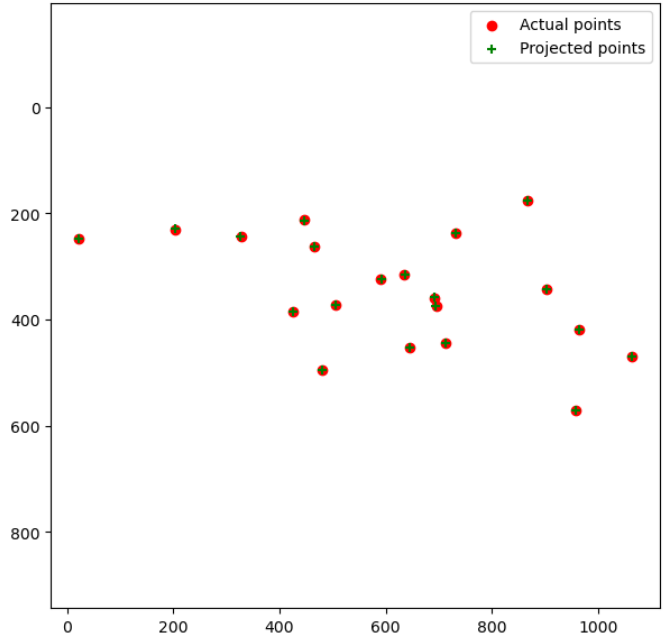
CS 4476/6476 Project 3

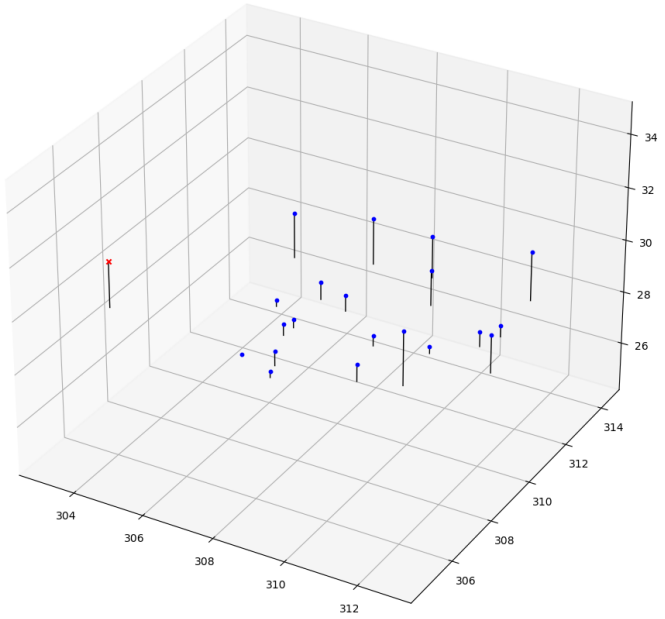
Shuvo Newaz
shuvo.newaz@gatech.edu
snewaz3
903614132

Part 1: Projection matrix

[insert visualization of projected 3D points and actual 2D points for the CCB image we provided here]

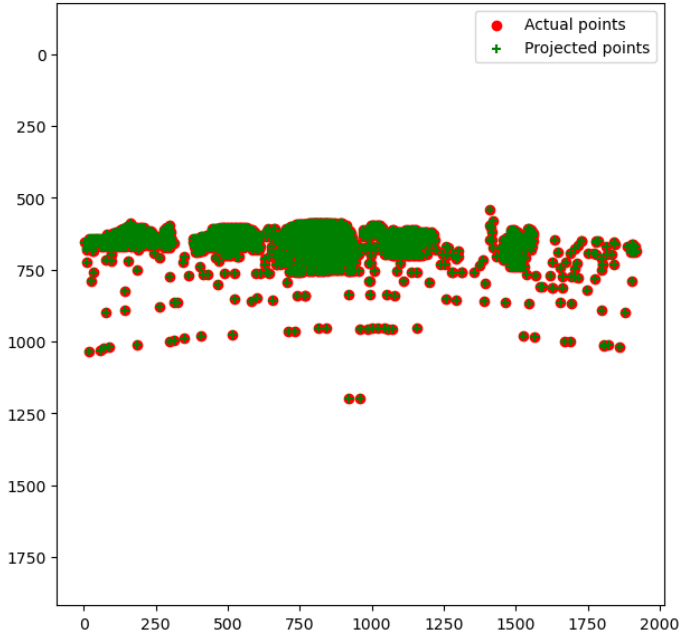


[insert visualization of camera center for the CCB image here]

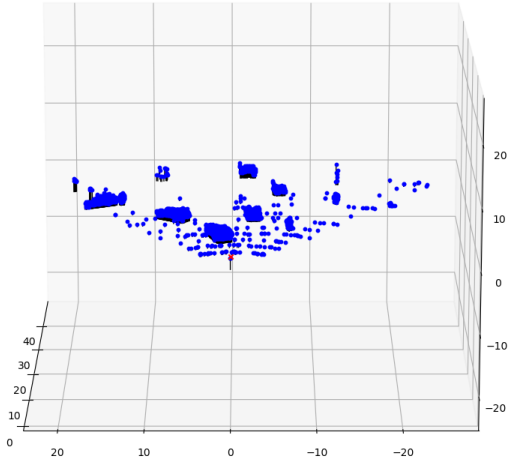


Part 1: Projection matrix

[insert visualization of projected 3D points and actual 2D points for the Argoverse image we provided here]



[insert visualization of camera center for the Argoverse image here]



Part 1: Projection matrix

[What two quantities does the camera matrix relate?]

The 3d world coordinates and the 2d image coordinates.

[What quantities can the camera matrix be decomposed into?]

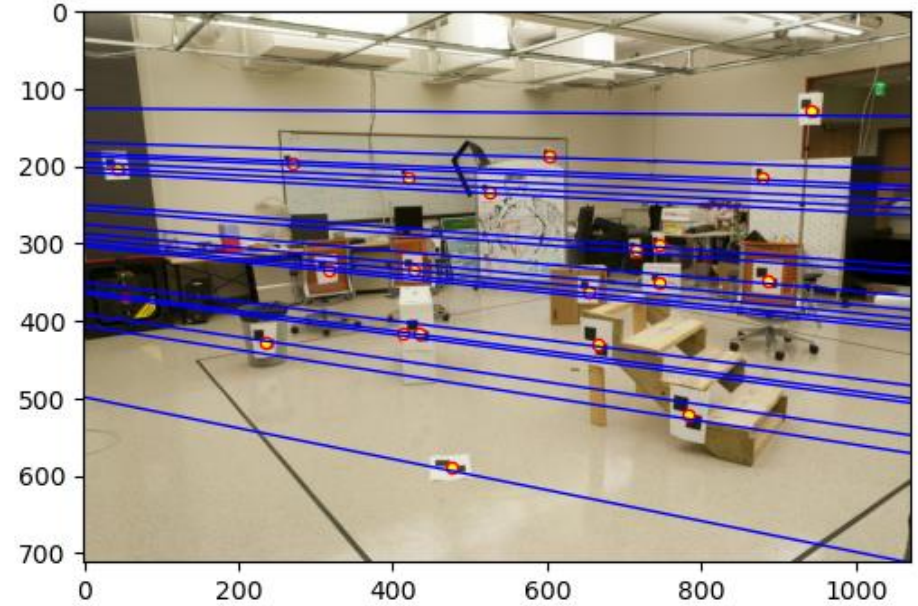
It can be decomposed into the Intrinsic matrix and the Extrinsic (Rotation and Translation) matrix.

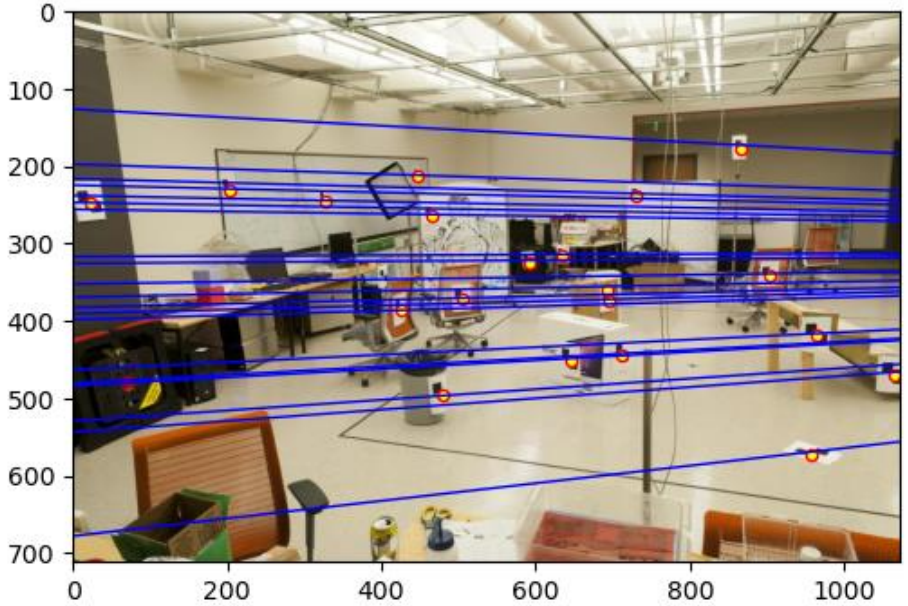
[List any 3 factors that affect the camera projection matrix.]

1. Focal length of camera.
2. Rotation
3. Shape of camera lens, i.e., plane/convex.

Part 2: Fundamental matrix

[insert visualization of epipolar lines on the CCB image pair]





Part 2: Fundamental matrix

[Why is it that points in one image are projected by the fundamental matrix onto epipolar lines in the other image?]

$Fx' = 0$ is the epipolar line associated with x'. Also, by definition $x^T F x' = 0$ for every corresponding pixels. If we take the epipolar line associated with x', the above equation will be satisfied regardless of the coordinates of x^T, i.e., every point in that image can be projected onto a corresponding point on the epipolar line of the other image.

[What happens to the epipoles and epipolar lines when you take two images where the camera centers are within the images? Why?]

If the cameras are looking at each other, the epipolar lines should diverge everywhere from the center of the image. This is because the epipoles will also be somewhere near the center, and the matches will be detected at the edges where the cameras see similar things.

Part 2: Fundamental matrix

[What does it mean when your epipolar lines are all horizontal across the two images?]

It means the difference between the images is only a translation (no rotation). The images are parallel to each other.

[Why is the fundamental matrix defined up to a scale?

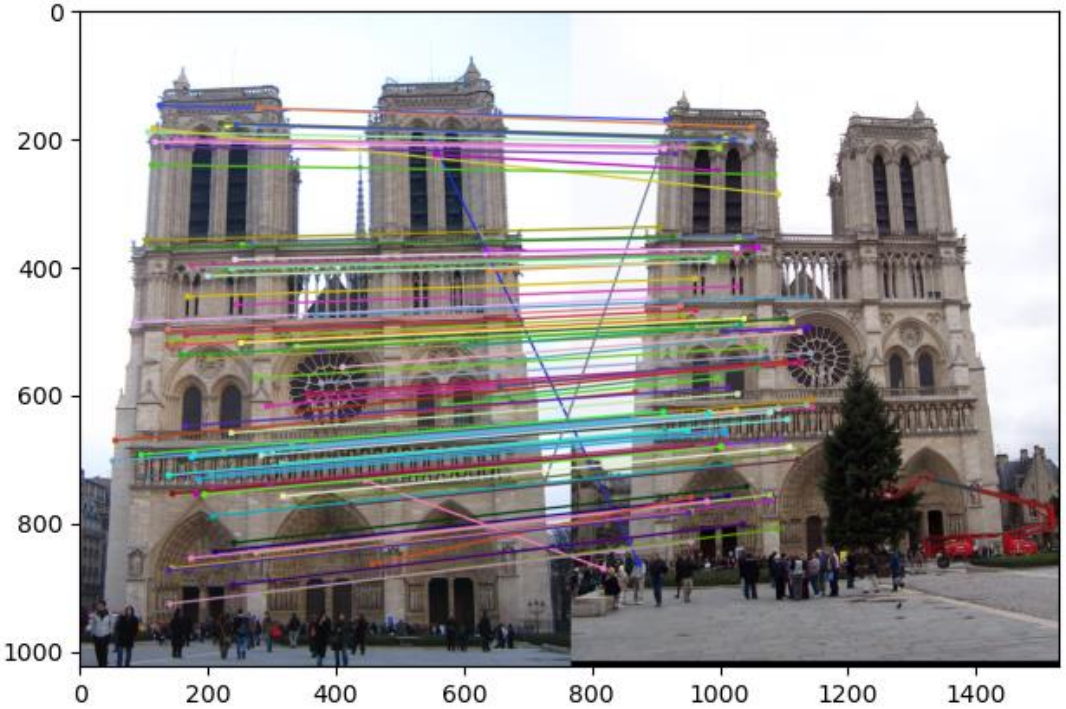
If we know the solution to ${x'}^{\mathrm{T}} F x = 0$, we can get the solution to ${x'}^{\mathrm{T}} (aF) x = 0$ for an arbitrary a.

[Why is the fundamental matrix rank 2?]

The Fundamental matrix is derived from the Essential matrix as $F = K^{-T} E K'^{-1}$. $E = [t]_{\times} R$ where $[t]_{\times}$ is a rank 2 matrix. Any multiplication with this matrix will result in a rank-2 matrix.

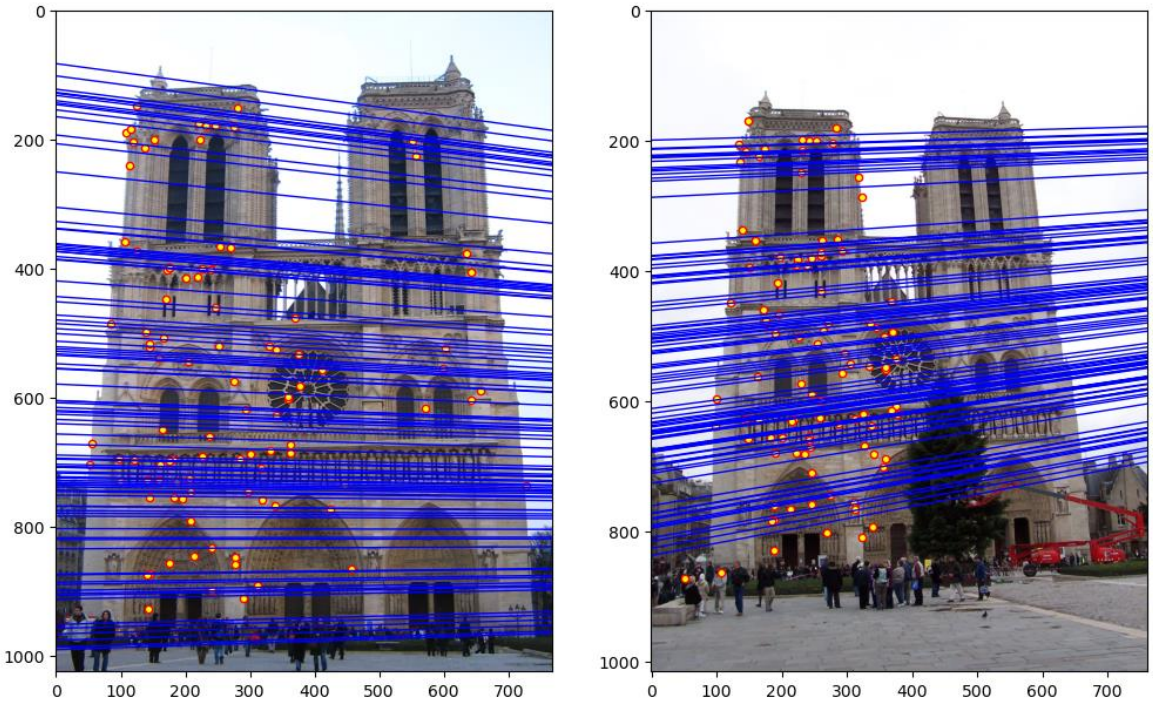
Part 3: RANSAC

[insert visualization of correspondences on Notre Dame after RANSAC]



Part 3: RANSAC

[insert visualization of epipolar lines on the Notre Dame image pair]



Part 3: RANSAC

[How many RANSAC iterations would we need to find the fundamental matrix with 99.9% certainty from your Mt. Rushmore and Notre Dame SIFT results assuming that they had a 90% point correspondence accuracy if there are 9 points?]

9 per formula. In reality, 794 iterations to get good correspondence for Notre Dame. 2593 for Mount Rushmore.

[One might imagine that if we had more than 9 point correspondences, it would be better to use more of them to solve for the fundamental matrix. Investigate this by finding the # of RANSAC iterations you would need to run with 18 points.]

28 iterations per formula. 569 for Notre Dame. The number is smaller that when we used 9 points because of the randomness in selecting the sample points for fundamental matrix estimation.

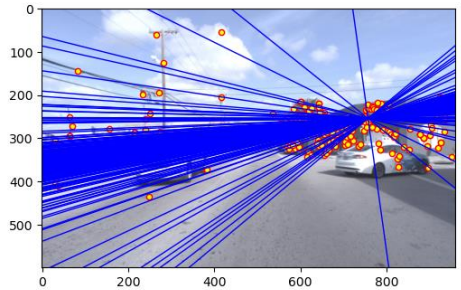
[If our dataset had a lower point correspondence accuracy, say 70%, what is the minimum # of iterations needed to find the fundamental matrix with 99.9% certainty?]

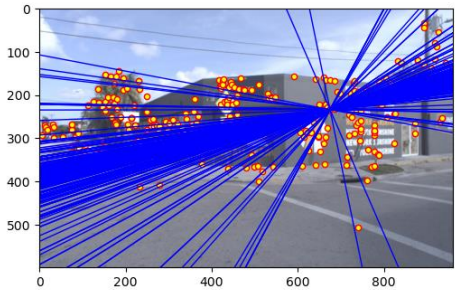
With 9 points, we would need 111 iterations according to the formula.

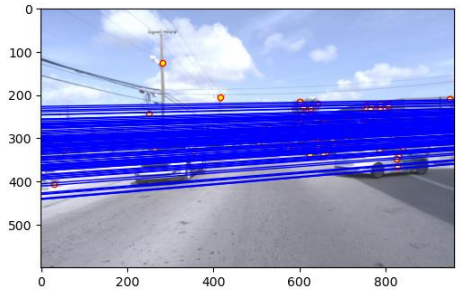
Part 4: Performance comparison

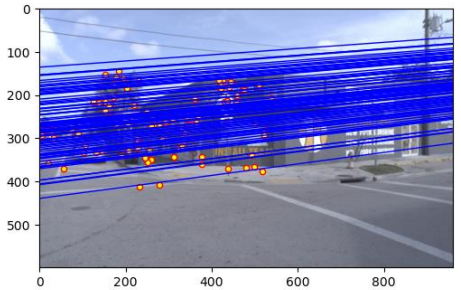
[insert visualization of epipolar lines on the Argoverse image pair using the linear method]

[insert visualization of epipolar lines on the Argoverse image pair using RANSAC]









Part 4: Performance comparison

[Describe the different performance of the two methods.]

Without RANSAC, the epipoles were in the middle of the images, even though the plane of the images appeared to be parallel. This was not the case when RANSAC was employed.

[Why do these differences appear?]

A lot of spurious matches have been allowed because of outliers. These appear as a false image plane. When RANSAC is used, contribution from outliers are reduced, and the image plane is percieved as almost correct.

[Which one should be more robust in real applications? Why?]

Using RANSAC should be more robust. When RANSAC is not used, the outliers also contribute to the estimation of the fundamental matrix. Depending on the how bad the outlier is, the fundamental matrix can be very unreliable. RANSAC removes the outliers to generate a more reliable fundamental matrix.

Part 5: Visual odometry

[How can we use our code from part 2 and part 3 to determine the “ego-motion” of a camera attached to a robot (i.e., motion of the robot)?]

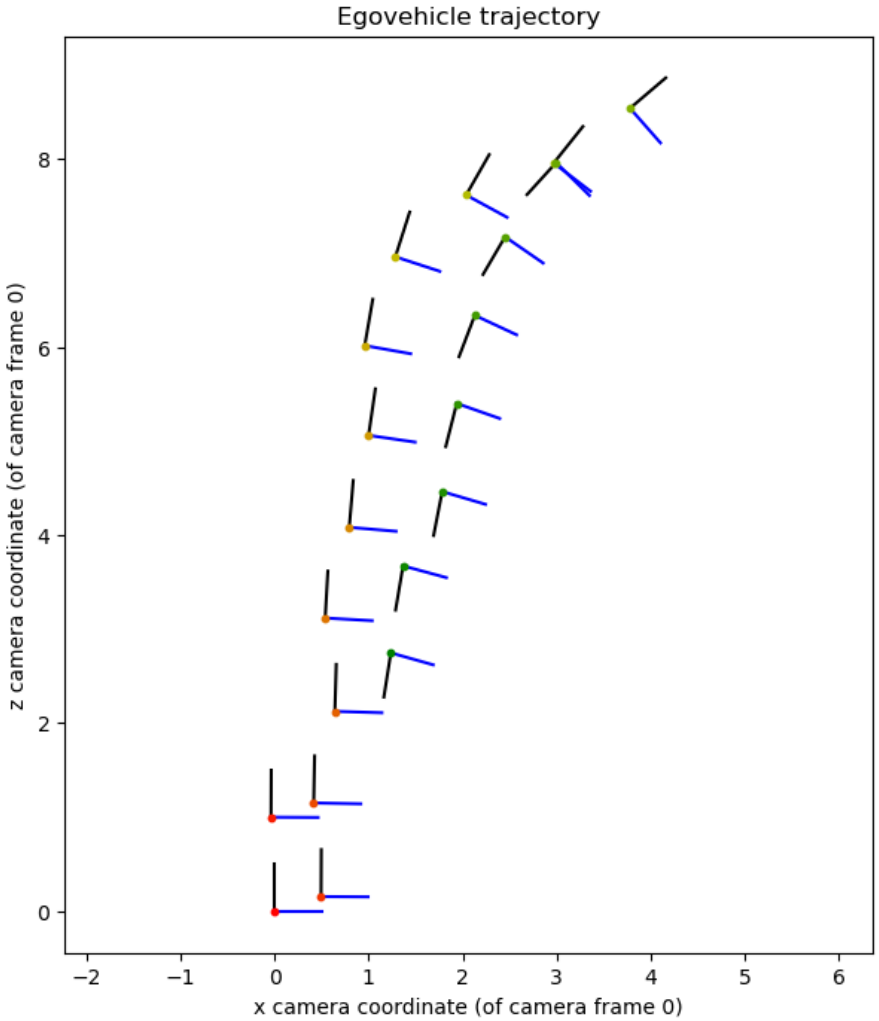
Use RANSAC and estimate the fundamental matrix that relates 2 consecutive frames. Use the fundamental matrix to extract the rotation and translation information of the frames. Use these information to determine the ego-motion.

[In addition to the fundamental matrix, what additional camera information is required to recover the ego-motion?]

The intrinsic camera matrix. Consequently, the focal length of the camera.

Part 5: Visual odometry

[Attach a plot of the camera's trajectory through time]



Part 6: Panorama Stitching

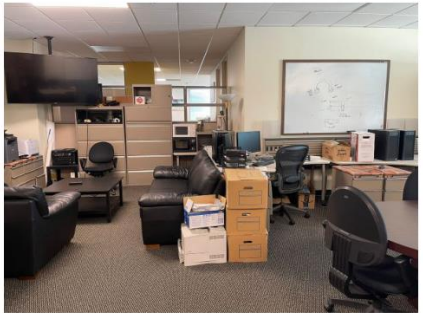
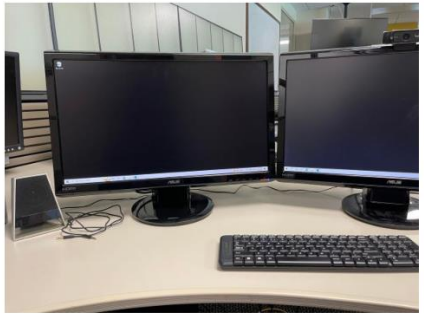
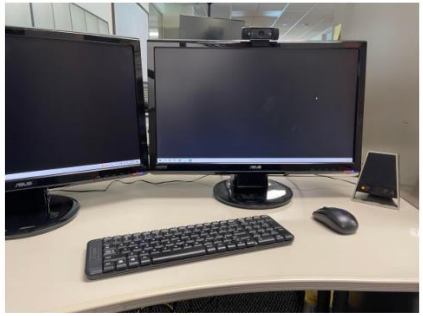
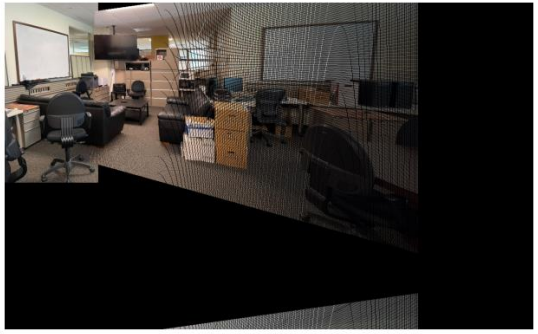
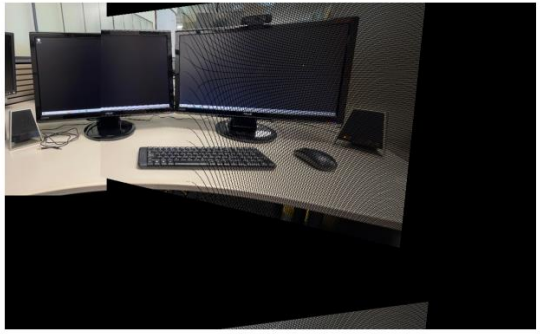
[Please add a README style documentation here for your implementation of panorama stitching with: description of what you implemented, instructions on how to replicate the results in clear steps that can be followed by course staff. Failure to replicate results by following this documentation will result in point penalties on this question of the assignment.]

Convert both images to grascale. Detect keypoints and SIFT descriptor for the image pair using cv2's *detectAndCompute()*. Get matches using these descriptors. Use the matches to determine the homography matrix using cv2's *findHomography()*. Project image B coordinate onto image A plane using the homography matrix. Create an empty canvas of zero values. Put the first image at the top left corner (horizontal panorama). Fill the projected image B coordinates with image B pixels values.

An argument *use_cv2* can be used (=True/False) to choose to compute the homography matrix using least-square as done in part 1 instead of using OpenCV's *findHomography()* function. The computation procedure is exactly the same as in part 1, except points_3d is now also a 2d matrix (destination points), points_2d are the source points, and the left-hand side matrix is $N \times 8$ instead of $N \times 11$. This method, however, does well with only the first image on the next slide.

Part 6: Panorama Stitching

[Insert visualizations of your stitched panorama here along with the 2 images you used to stitch this panorama (**there should be 3 images in this slide**)].

Part 6: Panorama Stitching

[Insert visualizations of your stitched panorama here along with the 2 images you used to stitch this panorama (**there should be 3 images in this slide**)].