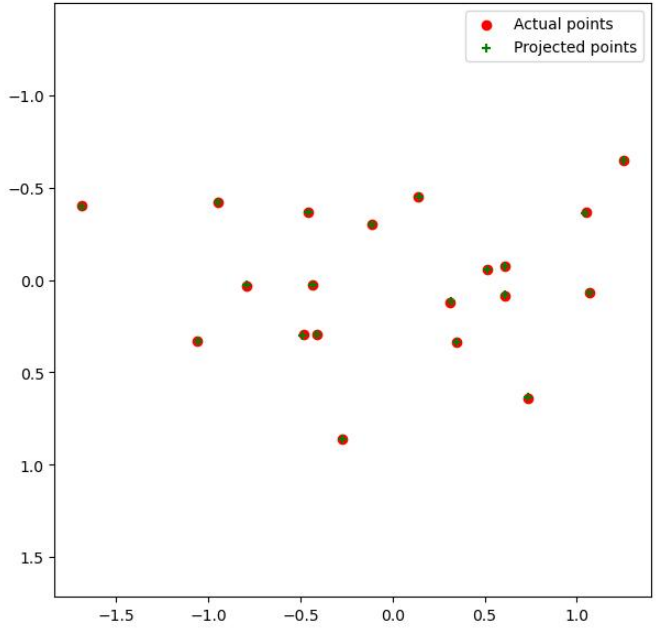
CS 6476 Project 3

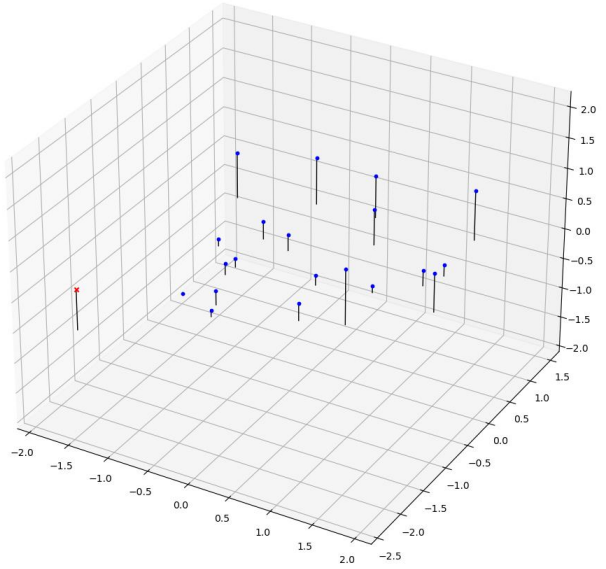
Mengying Lin
mlin365@gatech.edu
mlin365
904016037

Part 1: Projection matrix

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

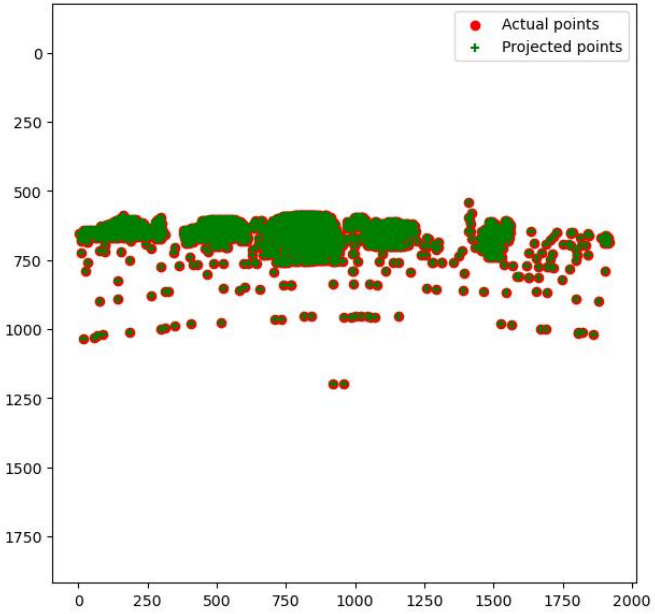


[insert visualization of camera center for the CCB image here]

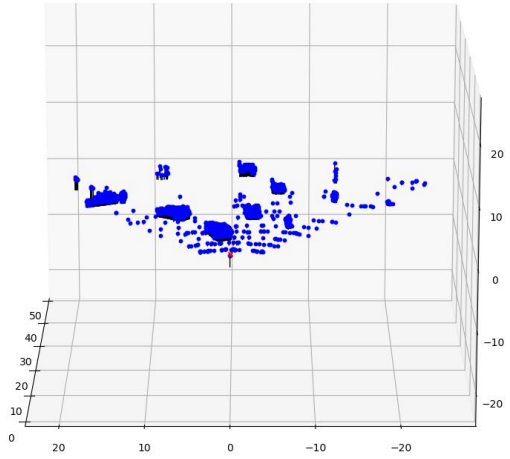


Part 1: Projection matrix

[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?]

3D world coordinates and 2D pixel coordinates

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

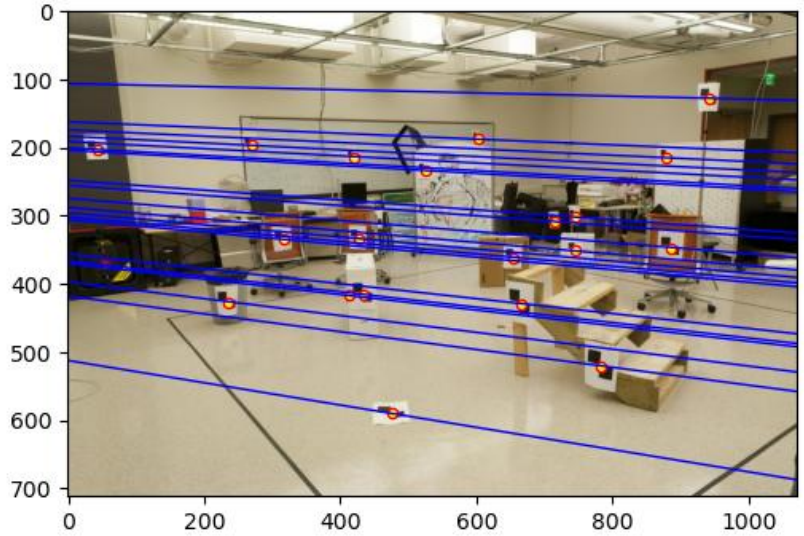
Intrinsic parameters (focal length, optical center);
Extrinsic parameters:camera's position and orientation

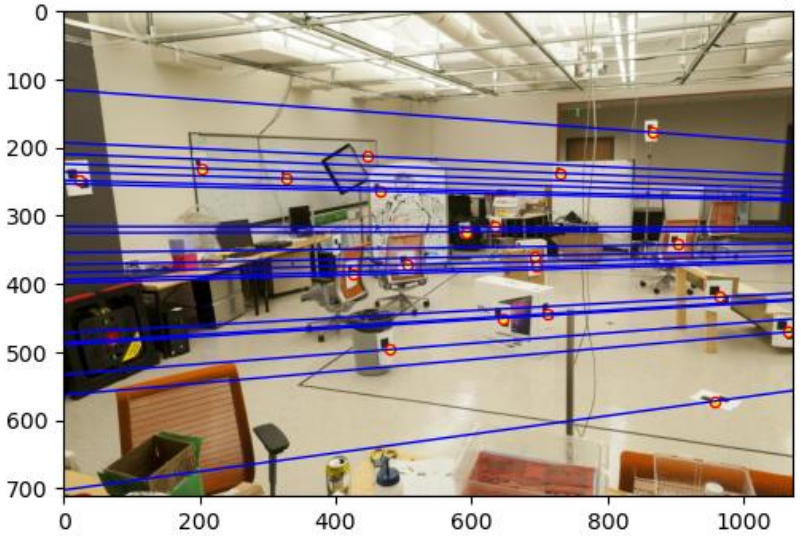
[List any 3 factors that affect the camera projection matrix.]

Focal length, camera’s position and orientation.

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?]

Because fundamental matrix encodes the geometric relationship between two camera views. The 3D points, the two camera centers, and their projections form an epipolar plane. This plane intersects both image planes along epipolar lines, meaning any point in one image must correspond to some point along the epipolar line in 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?]

The epipolar lines will form a fan-like pattern. When the camera centers are within the images, the epipoles—the points where the line connecting the two camera centers intersects the image planes—will also be inside the images. In this case, the epipolar lines radiate out from the epipoles, forming a fan-like pattern across the image.

Part 2: Fundamental matrix

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

The two camera views are horizontally aligned (i.e., they share the same plane). In this case the intersection between the line connecting the two camera centers and the image planes will be a line.

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

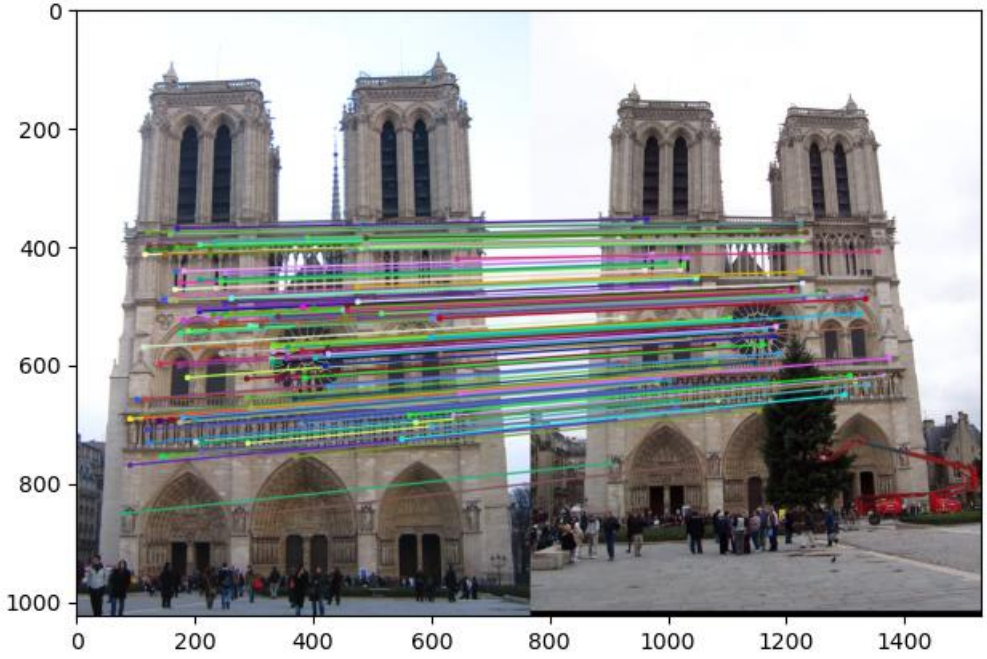
The fundamental matrix F operates in homogeneous coordinates, where multiplying all elements of F by a non-zero scalar does not change the underlying geometric relationships. In the epipolar constraint $x'^T F x = 0$ (x' and x are in homogeneous coordinates), scaling F by a factor λ still satisfies the equation because λ factors out, so it it is typically normalized for convenience.

[Why is the fundamental matrix rank 2?]

The fundamental matrix encodes a mapping from points in one image to epipolar lines in the other, and all epipolar lines must pass through the epipole (the projection of the other camera center). This constraint means that F must be singular, meaning it cannot have full rank (rank 3).

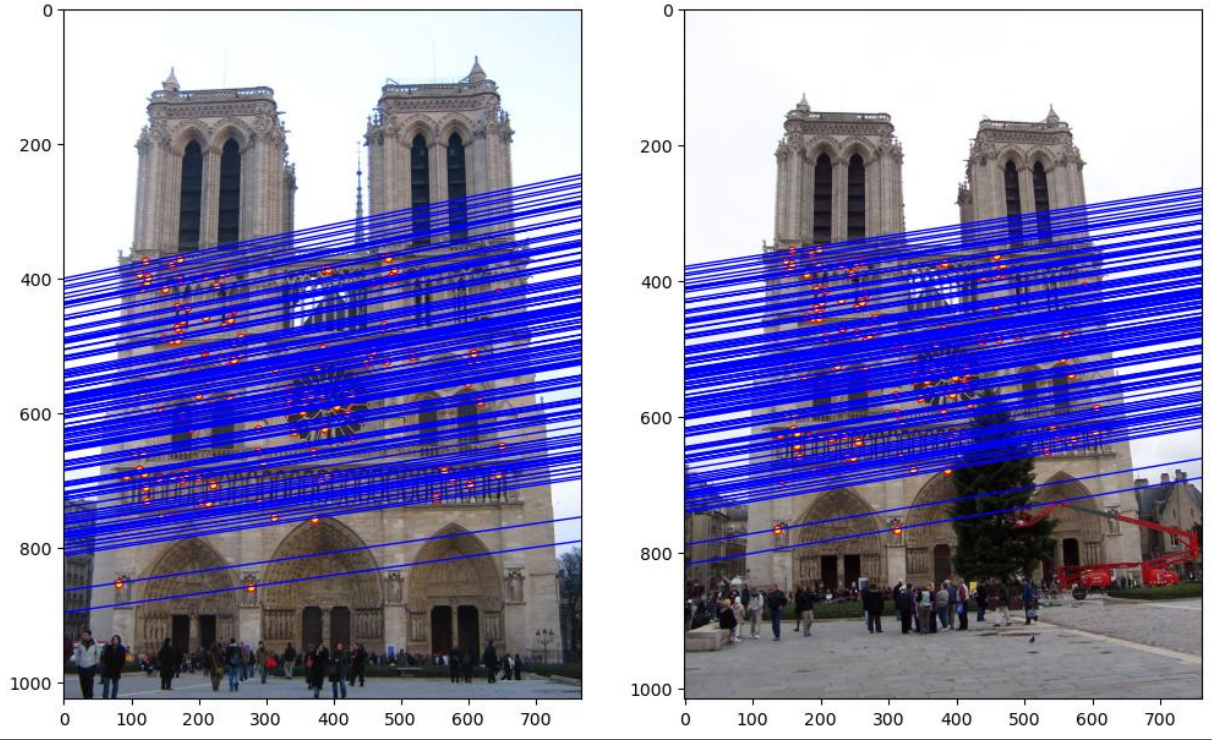
Part 3: RANSAC

[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?]

14

[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.]

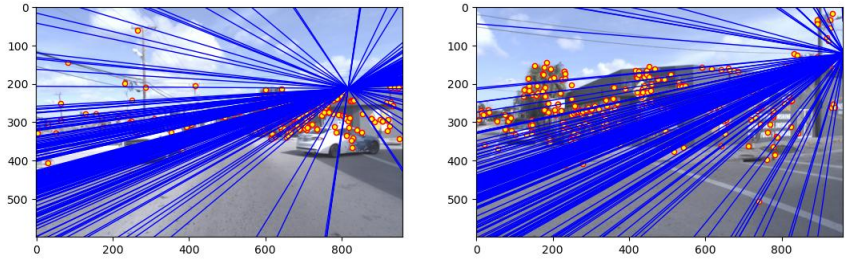
With 99.9% certainty and 90% point correspondence accuracy, we would need 42 iterations.

[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?]

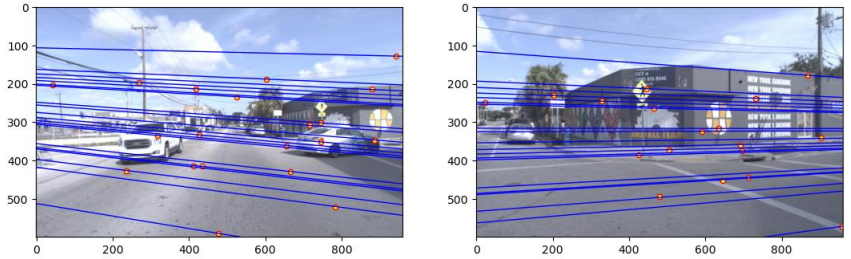
167 iterations.

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.]

For linear method, many lines appear to radiate from a central point whilr the other has epipolar lines that are more consistently horizontal and better aligned with the geometry of the scene.

[Why do these differences appear?]

Linear method is sensitive to noise and outliers in point correspondences, leading to poor alignment of epipolar lines and overfitting the data, while RANSAC is designed to handle outliers and can estimate the fundamental matrix better.

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

RANSAC. It is specifically designed to handle noise and outliers in the data. Real-world data often contain errors or mismatches due to sensor inaccuracies or other environmental factors.

Part 5: Visual odometry

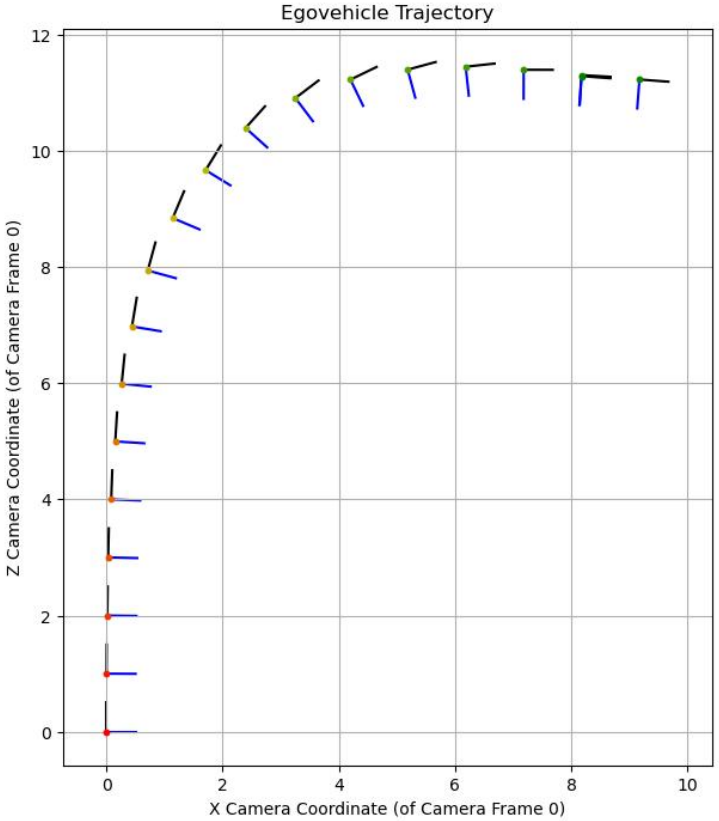
[How can we use our code from part 2 and part 3 to determine the “ego-motion” of a camera attached to a robot (i.e., motion of the robot)?]
Capture consecutive image frames as the robot moves, then detect feature correspondences between the frames. Use RANSAC to compute the fundamental matrix, ensuring robustness to outliers. With the camera's intrinsic parameters, convert the fundamental matrix into the essential matrix, which encodes the motion between the frames. Finally, decompose the essential matrix into the camera's rotation and translation, which reflect the robot motion.

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

Camera intrinsics.

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.]

Overview

The panorama_stitch function implements a panorama stitching technique using feature detection, matching, homography computation, and manual image warping.

Part 6: Panorama Stitching

Key Steps:

- Detects interest points using either SIFT / ORB feature detectors.
- Matches keypoints using Brute-Force (BF) / FLANN-based matchers.
- Computes the homography matrix using RANSAC to ensure robust matching.
- Manually warps one image into the coordinate space of the other.
- Outputs a stitched panorama with the two images aligned and blended together.

Part 6: Panorama Stitching

Result replication

Please use the following code snippet:

```
fromvision.part6_panorama_stitchingimportpanorama_stitch
imageA=cv2.imread("path to image A")
imageB=cv2.imread("path to image B")
downscale_factor=1
imageA=cv2.resize(imageA,None,fx=downscale_factor,fy=downscale_factor)
imageB=cv2.resize(imageB,None,fx=downscale_factor,fy=downscale_factor)
panorama=panorama_stitch(imageA,imageB)
cv2.imwrite("panorama.jpg",panorama)
plt.figure()
plt.imshow(cv2.cvtColor(panorama,cv2.COLOR_BGR2RGB))
plt.axis("off")
```

Part 6: Panorama Stitching

Result replication

You might also want to change the variables in the function:

```
panorama=panorama_stitch(imageA, imageB, detector_mode='SIFT',
matcher_mode ='BF')
# detector_mode: choosing from {'SIFT', 'ORB'}; matcher_mode: choosing
from {'BF', 'FLANN'}
```

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

[Discuss the results of your panorama stitching. What feature detectors and matchers did you use? How did they influence the quality of the final panorama?].

	SIFT	ORB
BF		
FLANN		

Part 6: Panorama Stitching

[Discuss the results of your panorama stitching. What feature detectors and matchers did you use? How did they influence the quality of the final panorama?].

I use 4 configurations: SIFT+BF; SIFT+FLANN; ORB+BF; ORB+FLANN.

I would say it is hard to evaluate the quality based on the obtained images. They all warp second image and nicely align it with the first one, and all contain some artifact patterns in the warped images. While the visual differences were subtle, it is notable that FLANN-based matching was consistently faster than BF during the tests.