Computer Vision 2024

(CSE344/ CSE544/ ECE344/ ECE544)

Assignment-3

Max Marks (UG/PG): 45/ 50 Due Date: 14/04/2024, 11:59 PM Instructions

MOHAMMAD SHARIQ
2020220

—---

• Keep collaborations at high-level discussions. Copying/plagiarism will be dealt with strictly.

• Your submission should be a single zip file Roll Number HW[n].zip. Include only the relevant files arranged with proper names. A single .pdf report explaining your codes with relevant graphs, visualization and solution to theory questions.

• Remember to turn in after uploading on Google Classroom. No justifications would be taken regarding this after the deadline.

• Start the assignment early. Resolve all your doubts from TAs during their office hours two days before the deadline.

• Kindly document your code. Don't forget to include all the necessary plots in your report.

• All [PG] questions, if any, are optional for UG students but are mandatory for PG students. UG students will get BONUS marks for solving that question.

• All [BONUS] questions, if any, are optional for all the students. As the name suggests, BONUS marks will be awarded to all the students who solve these questions.

• Your submission must include a single python (*.py*) file for each question. You can submit *.ipynb* along with the *.py* files. Failing to follow the naming convention or not submitting the python files will incur a penalty.

1. (10 points) Theory Questions on Epipolar Geometry

1. (3 points) The Essential matrix E = $[t_\times]R$, where t and R are the translation vector and the rotation matrix. The epipoles are the points at the intersection of the baseline (the line connecting the two camera centres) and the image planes. The epipoles also happen to be the null-space and the left null-space of E. Find the epipoles in terms of t and R. Show your work.

2. (2 points) Assume that a stereo camera setup has no relative rotation (i.e., R = I) and the translation is purely horizontal (i.e., t = $[t_x, 0, 0]^\top$. Construct the Essential matrix and show that the *corresponding* points (i.e., image points in the two cameras that are projections of the same 3D point) will always have the same *y*-coordinate.

3. [PG] (5 points) Stereo rectification is the process of transforming the two images obtained from an arbitrary stereo camera setup such that their epipolar lines are parallel *and* horizontal (equivalent to the two cameras having no relative rotation and translation only along the horizontal direction). This amounts to applying a rotation to the image points such that the epipoles get mapped to points at infinity along the x axis. Refer to this slide deck[1] and derive the expression for the rotation matrix (R_{rect}) required for the stereo rectification.

Q1-SOLUTIONS::

(1):

To find the epipoles in terms of the translation vector t and the rotation matrix R, let's first define the Essential matrix E as:

$$E = [t]_\times R$$

where $[t]_\times$ is the skew-symmetric matrix representation of the translation vector t.

Given that the epipoles are the null-space and left null-space of E, let's denote the epipoles as e_l and e_r, respectively.

Finding the Left Epipole ::

The left epipole is the null-space of E. Therefore, for any vector x in the null-space of E, we have:

$$Ex=0$$

Expanding this equation we get-

$$[t]_{\times}Rx=0$$

Multiplying both sides by R^{-1} (assuming R is invertible):

$$[t]_{\times}x = 0$$

Since $[t]_{\times}$ is a skew-symmetric matrix, its null-space consists of vectors parallel to t. Therefore, the left epipole e_l lies along the direction of the translation vector t.

$$e_l = \lambda_l t$$

where λ_l is a scalar. This equation indicates that the left epipole e_l lies along the direction of the translation vector t. Therefore, it is parallel to t.

Right Epipole e_r can be obtained by :

The right epipole is the left null-space of E. Therefore, for any vector y in the left null-space of E, we have:

$$y^T E = 0$$

Expanding this equation:

$$y^T [t]_\times R = 0$$

Taking into account the properties of skew-symmetric matrices, we can rewrite $[t]_\times$ as $-[t]_\times^T$:

$$y^T [-t]_\times^T R = 0$$

$$(R^T [-t]_\times y)^T = 0$$

Since this equation must hold for any vector y in the left null-space of E, the expression $R^T [-t]_\times y$ must be equal to zero. This implies that:

$$R^T [-t]_\times = 0$$

$$R^T t = 0$$

Therefore, the right epipole e_r is given by:

$$e_r = R^T t$$

(2):

Assuming a stereo camera setup with no relative rotation ($R = I$) and purely horizontal translation ($t = [t_x, 0, 0]^T$), the Essential matrix simplifies to:

$$\boldsymbol{E} = [\boldsymbol{t}]_x$$

The skew-symmetric matrix for t is:

$$[t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

Given $t = [t_x, 0, 0]^T$, this becomes:

$$[t]_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

Since $R = I$, the Essential matrix is just $[t]_x$:

$$E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

For corresponding points p and p′ in the two images, they satisfy:

$$p'^T E p = 0$$

If $p = [x, y, 1]^T$ and $p' = [x', y', 1]^T$, then:

$$[x', y', 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

Which can be simplified to:

$$y'(-t_x) + t_x y = 0$$

$$-t_x(y' - y) = 0$$

Since $t_x \neq 0$, it follows that $y' = y$, showing that the y-coordinates of corresponding points in the two images are equal.

(3):

The rotation matrix R_{rect} required for stereo rectification depends on the camera's intrinsic parameters and the extrinsic parameters describing the relative orientation and position of the two cameras.

2. (40 points) Panorama Generation

Download the dataset from link. The dataset contains set of images that will be used for panorama generation. For the steps 1 to 5 you are required to use only the first two images from the set.

1. (5 points) Keypoint detection: Extract the keypoints and descriptors from the first two images using the SIFT algorithm. SIFT (Scale-Invariant Feature Transform) is a computer vision algorithm used for feature detection and description. After extracting the keypoints and descriptors, draw them overlaid on the original images to visualize and verify their correctness.

2. (5 + 5 points) Feature matching: Match the extracted features using two different algorithms: BruteForce and FlannBased . BruteForce is a simple algorithm that matches features by comparing all the descriptors of one image with all the de scriptors of the other image. FlannBased (Fast Library for Approximate Nearest Neighbors) is a more efficient algorithm that uses a hierarchical structure to speed up the matching process. After performing the matching, display the matched fea tures by drawing lines between them.

3. (5 points) Homography estimation: Compute Homography matrix using RANSAC.

[RANSAC](#) (Random Sample Consensus) is an iterative algorithm used for robust estimation of parameters in a mathematical model. The homography matrix is used to align the two images so that they can be stitched together to form a panorama.

4. (5 points)Perspective warping: Perspective warping is a process that transforms the perspective of an image so that it appears as if it was taken from a different viewpoint. Warp any pair of images (with overlapping field of view) using their respective homography matrices and display the left and right images side-by-side. These *warped* images will be part of your panorama. Display the images without cropping the images or stitching them (as asked in the next part).

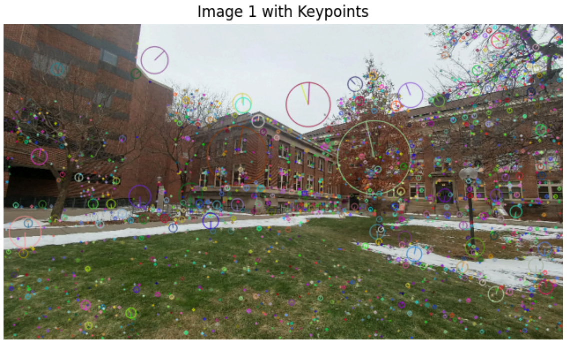
5. (5 points) Stitching: Two images need to be stitched together to form a panorama. Display the final panorama without any cropping or blending, along with the panorama obtained after cropping and blending.

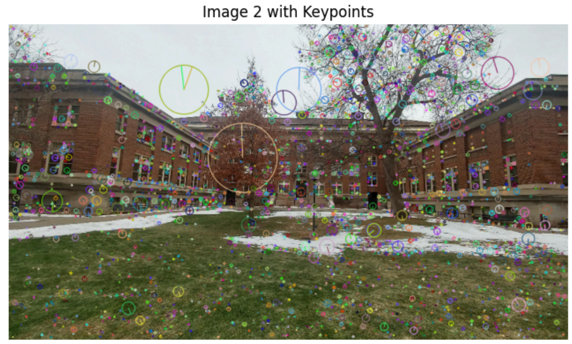
6. (10 points) Multi-Stitching: Perform multi stitching for all the images in the folder and display the final result. (Hint: Use the function implemented for Stitching). \

[1]https://www.cs.cmu.edu/ 16385/s17/Slides/13.1 Stereo Rectification.pdf

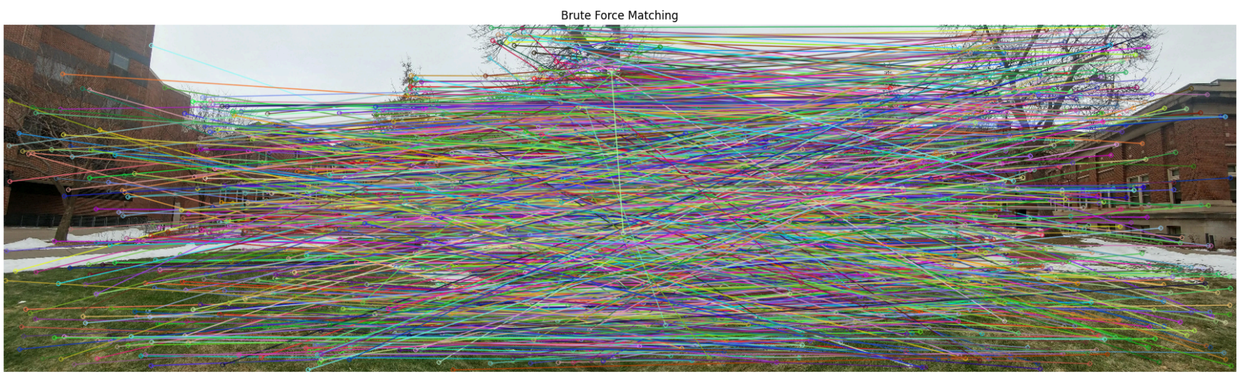
Q2 -SOLUTIONS::

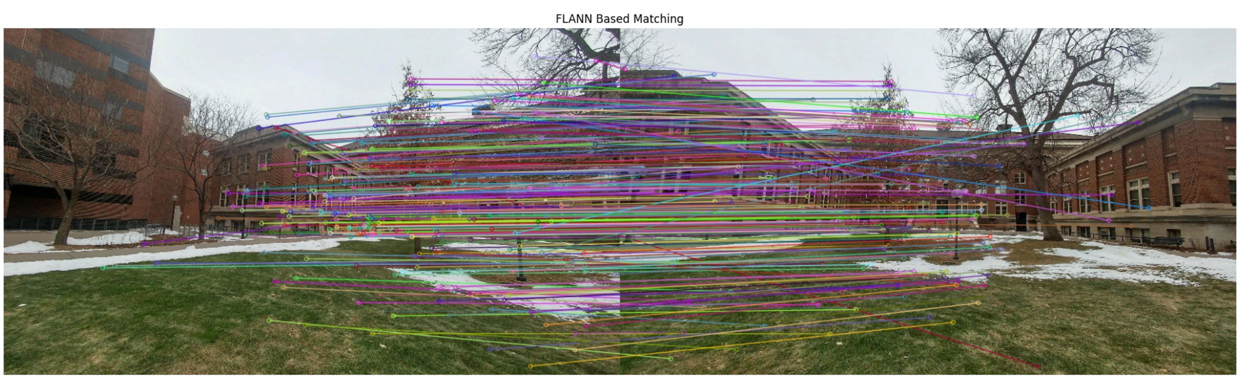
(1):





(2):





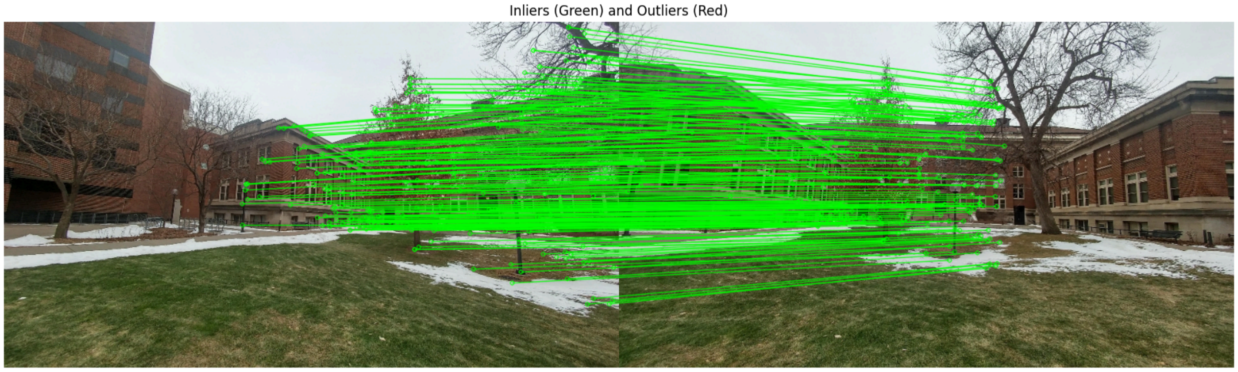
(3): BRUTE FORCE FEATURE MAPPING

```
[50] ✓ 0.0s

Brute Force Homography Matrix:
 [[-6.80990623e+01  1.45362419e+00  2.45788405e+04]
 [-2.05919598e+01 -4.22170887e+01  1.28122833e+04]
 [-7.04686127e-02 -2.02201087e-03  1.00000000e+00]]

Brute Force Inliers Mask:
 [1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 1 0 0 1 1 0 0
 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0 0 0 1 1 0 0 0 1 0
 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1
 0 1 0 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1
 1 0 0 1 1 0 1 1 1 1 0 0 1 0 1 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 0 0 1 1 0 1 0
 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0
 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 1 0 0 1 0 0 1 0 0
 1 0 0 0 1 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 0 0 1 1 1 1 1 1
 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0
 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1
 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1
 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
...
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Inliers (Green) and Outliers (Red)

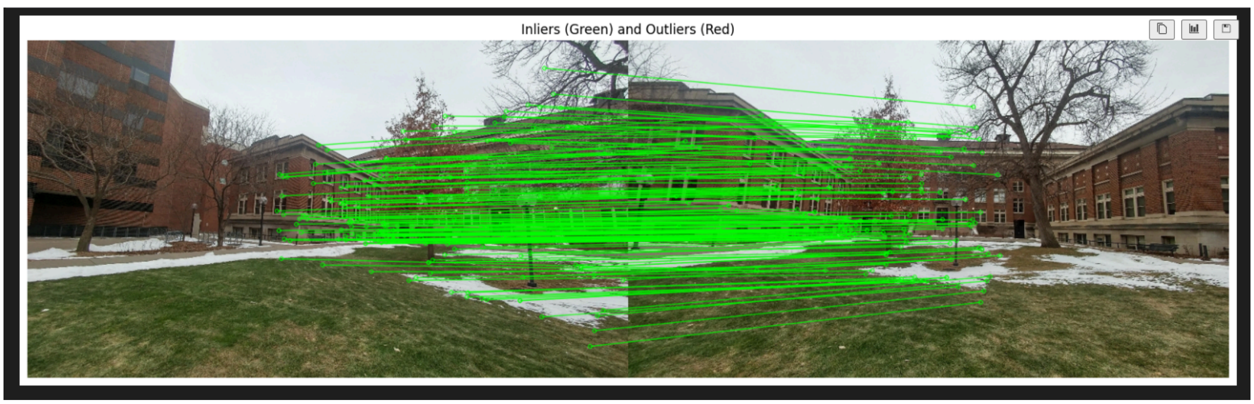


FLANN BASED FEATURE MAPPING

```
[48]  ✓ 0.0s

...  Homography Matrix:
      [[-6.27174239e+01  2.67045257e+00  2.23793754e+04]
      [-1.91022444e+01 -3.82036943e+01  1.17735441e+04]
      [-6.55340490e-02  2.05497829e-04  1.00000000e+00]]

     Inliers Mask:
      [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 0
      0 1 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1
      1 0 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 1 1
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 0 1 0 0 1 1 1 1
      0 1 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1
      1 0 0 1 1 1 0 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1
      0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1]
```



(4):

Typically, the image to be warped (source) is the one that will be adjusted to align with the reference image. In this case, let’s choose img2 as the source to warp to the perspective of img1.

BRUTE BASED HOMOGRAPHY WARPED

FLANN BASED HOMOGRAPHY WARPED

(5):

(6):

Final Panorama