ENPM 673
PERCEPTION FOR AUTONOMOUS ROBOTS

PROJECT 2

KIRAN AJITH
119112397

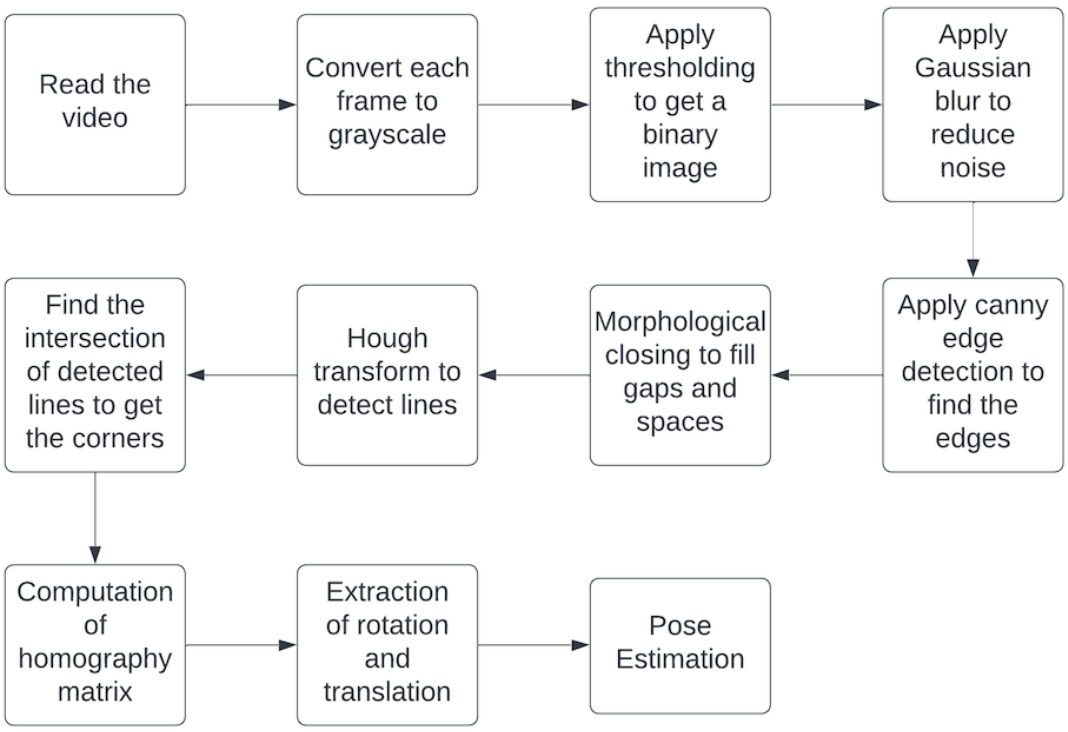
Problem 1

Pipeline

The task is to perform pose estimation using homography given a video containing a sheet of paper on a surface. The goal is to compute the rotation and translation between the camera and a frame of reference on the corner of the sheet of paper. The various processes involved in achieving this is illustrated in the pipeline given below.

Flowchart

Kiran Ajith | March 8, 2023



The steps involved are briefly discussed in the following sections.

Noise Reduction

After reading the video and converting each frame to grayscale, a Gaussian filter is applied to it to minimise the effect of noise in the signal smoothening out the image. Gaussian blur is done by convolving the image with a Gaussian kernel. By performing this convolution, each value of the pixel is replaced by a weighted average of the pixel values in its neighbourhood.

Gaussian function:

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Edge Detection

An edge is a point in an image where the intensity changes rapidly. Canny edge detection is a technique used to detect the presence of an edge in an image. The gradient operator is used to identify the changes in intensity in an image.The magnitude of the gradient operator gives the rate of change of intensity and the gradient orientation gives the direction of change.
Non-maximum suppression is used to thin out edges and keep only the local maxima. Strong and weak edges are identified by double suppression. The weak edges are then linked to strong edges using the technique of hysteresis. This gives a continuous contour.
Hysteresis threshold:

$$\begin{aligned}\|\nabla f(x,y)\| &\geq \mathrm{t_h} \\ \mathrm{t_l} \geq &\|\nabla f(x,y)\| < \mathrm{t_h} \\ \|\nabla f(x,y)\| &< \mathrm{t_l}\end{aligned}$$

After this step, we perform the morphological closing of spaces and gaps. This is done by dilation and erosion.

Line Detection

Lines in the image are obtained using hough transformation. This is done by representing lines in the image space by points in the parameter space(slope and intercept). A two-dimensional array of accumulator cells are produced, which represents the possible value of line parameters.

Gradient operator:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The magnitude of the gradient r:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Gradient direction:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Line equation and d

$$y = mx + c$$

$$r = x\cos\theta + y\sin\theta$$

$$H(r,\theta) = \sum_i \delta(r - x_i\cos(\theta) - y_i\sin(\theta))$$

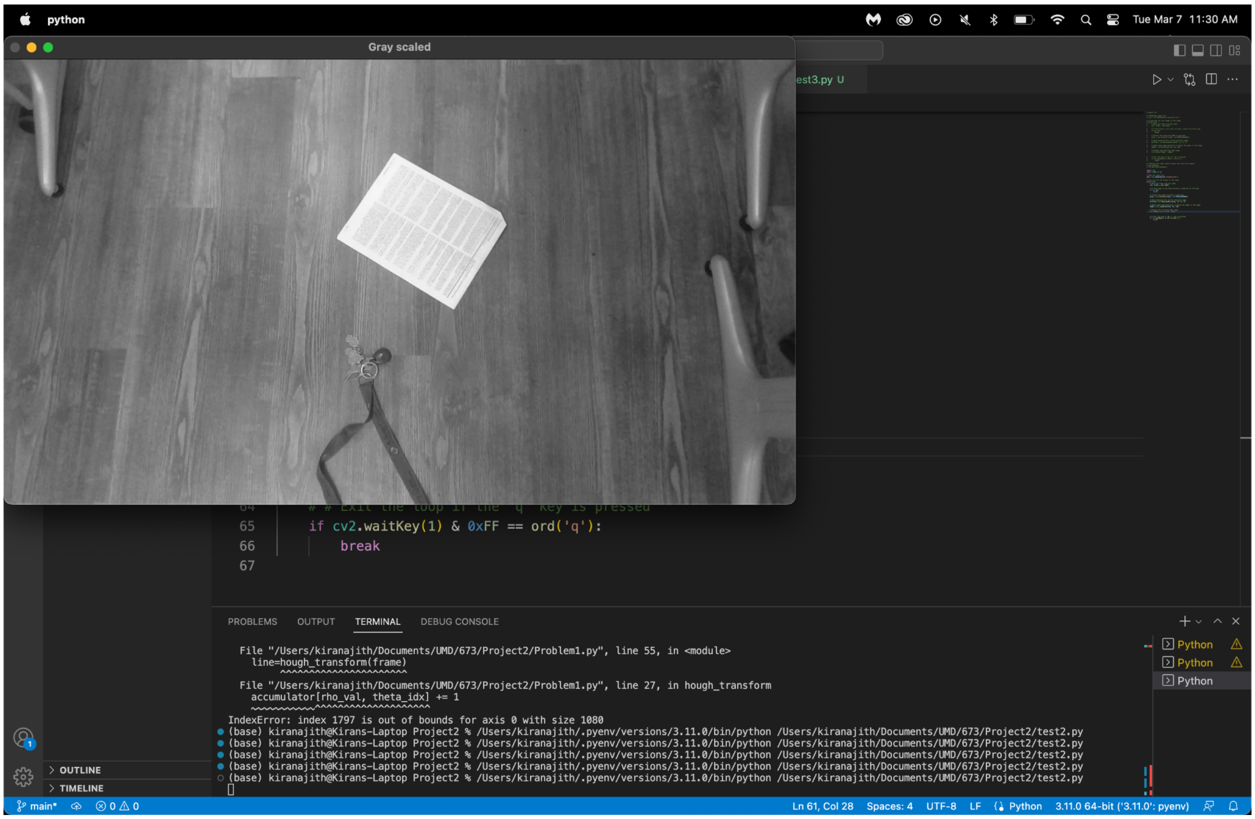
Homography

Homography is the technique used to connect the coordinates of points in one image to another image in the form of a matrix called homography matrix Once line detection using hough transform is done, we find the corresponding pairs of line in the two images.

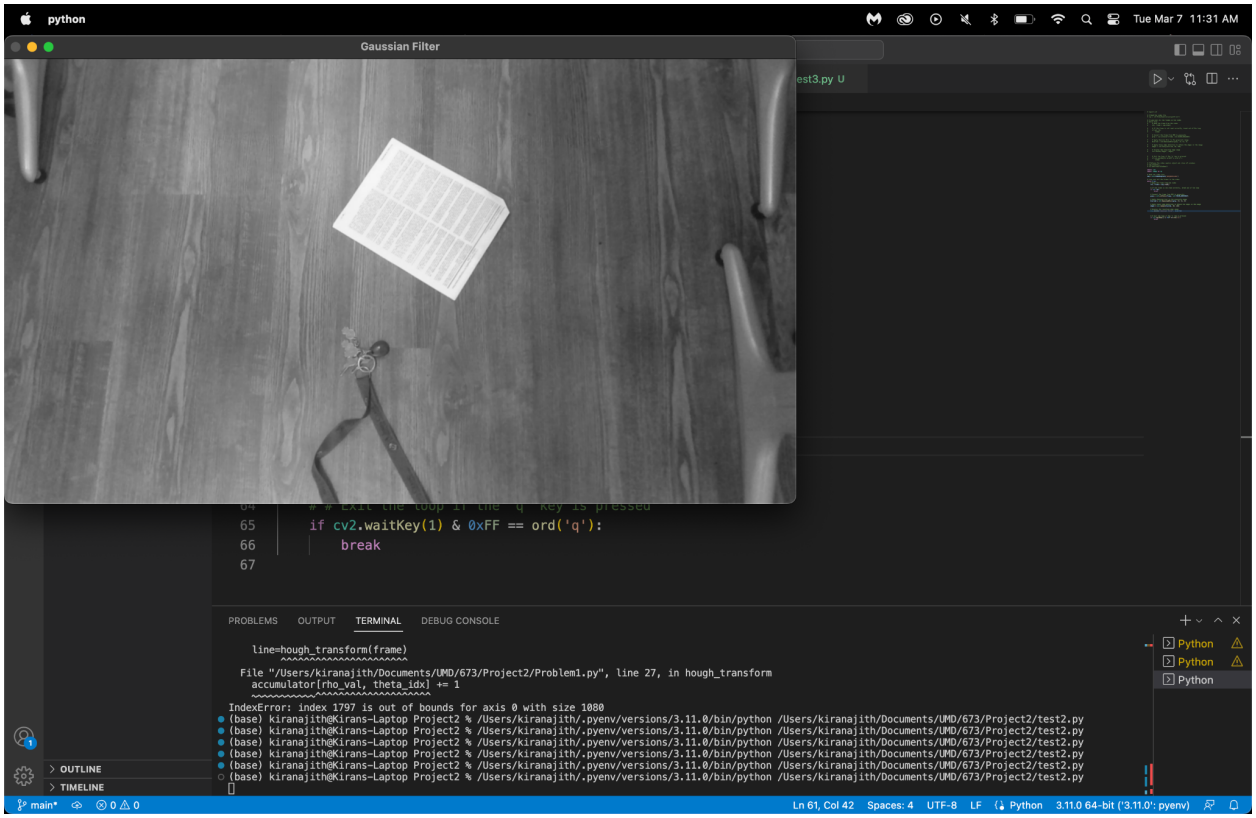
Homography matrix is a 3x3 matrix calculated by solving a system of linear equations.

$$H[x_1, y_1, 1]^T = [x_2, y_2, 1]^T$$

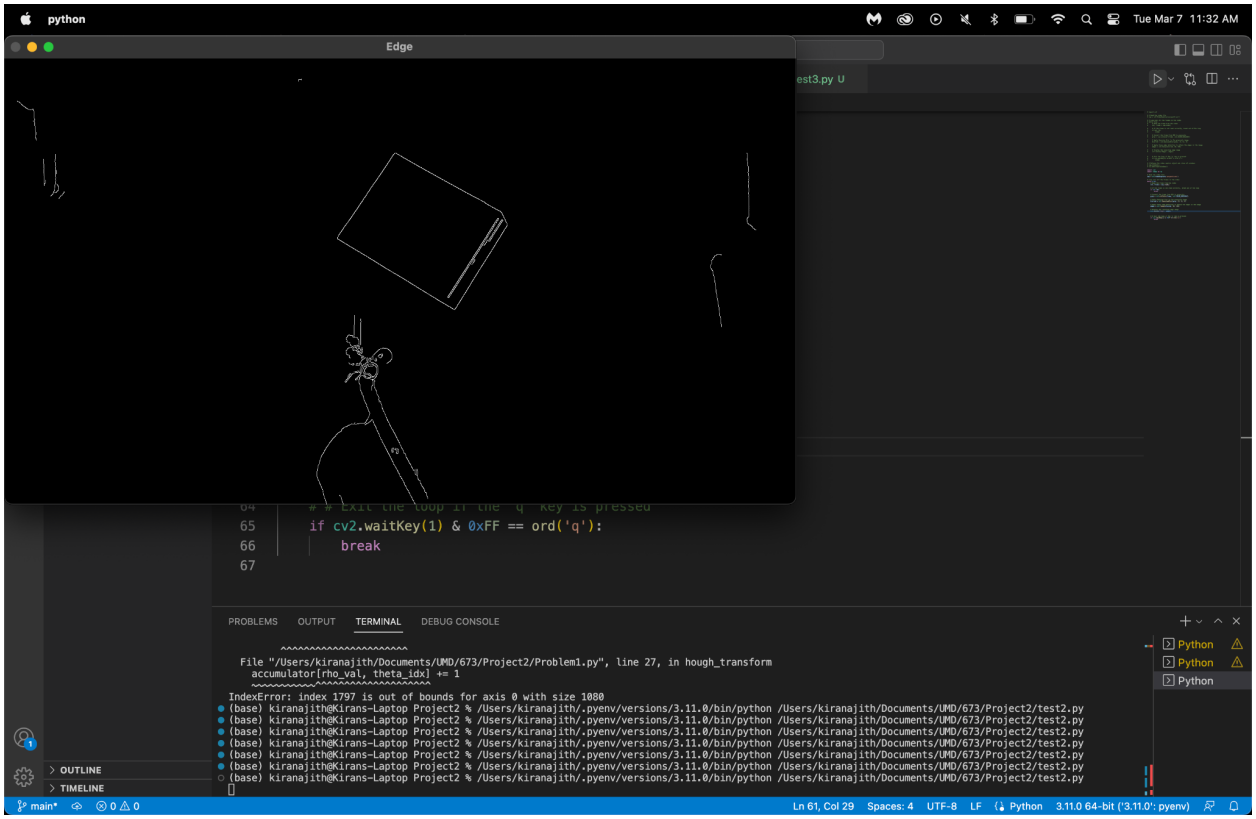
Results



Gray scaled


python
Tue Mar 7 11:31 AM
Gaussian Filter
65 if cv2.waitKey(1) & 0xFF == ord('q'):
66 break
67
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
line=hough_transform(frame)
File "/Users/kiranajith/Documents/UMD/673/Project2/Problem1.py", line 27, in hough_transform
accumulator[rho_val, theta_idx] += 1
IndexError: index 1797 is out of bounds for axis 0 with size 1080
(base) kiranajith@Kirans-Laptop Project2 % /Users/kiranajith/.pyenv/versions/3.11.0/bin/python /Users/kiranajith/Documents/UMD/673/Project2/test2.py
OUTLINE
TIMELINE
main*
Ln 61, Col 42 Spaces: 4 UTF-8 LF Python 3.11.0 64-bit ('3.11.0': pyenv)

Smoothed


python
Tue Mar 7 11:32 AM
Edge
65 if cv2.waitKey(1) & 0xFF == ord('q'):
66 break
67
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
File "/Users/kiranajith/Documents/UMD/673/Project2/Problem1.py", line 27, in hough_transform
accumulator[rho_val, theta_idx] += 1
IndexError: index 1797 is out of bounds for axis 0 with size 1080
(base) kiranajith@Kirans-Laptop Project2 % /Users/kiranajith/.pyenv/versions/3.11.0/bin/python /Users/kiranajith/Documents/UMD/673/Project2/test2.py
OUTLINE
TIMELINE
main*
Ln 61, Col 29 Spaces: 4 UTF-8 LF Python 3.11.0 64-bit ('3.11.0': pyenv)

Edge detection

```
(base) kiranajith@MacBook-Pro Project2 % /Users/kiranajith/.pyenv/versions/3.11.0/bin/python /Users/kiranajith/Documents/UMD/673/Project2/Problem1.py
Pose Estimation
Rotation
Roll: 137.15, Pitch: 54.01, Yaw: 104.91
Translation
 x: -0.00, y: -0.00, z: 5.43
Pose Estimation
Rotation
Roll: -146.15, Pitch: 54.22, Yaw: -118.63
Translation
 x: -0.01, y: -0.00, z: 13.95
Pose Estimation
```

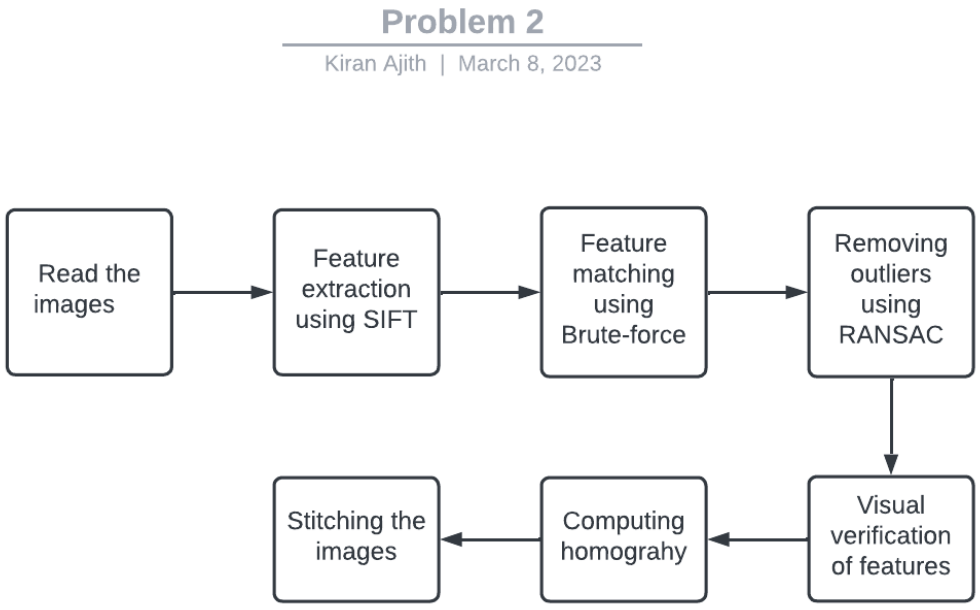
Output values

Run the problem1.py in this project package to get the output seen in the above images.
Note that the program runs very slowly using the custom homography function. Replacing the custom function with the built-in cv2.FindHoughlinesP() makes the output faster.

Problem 2

In this problem, we are given four images.The task is to stitch the images together to create a panoramic image.The four images are taken from the same camera position with only rotation and no translation.

The pipeline given below illustrates the various processes involved in achieving this



SIFT Detection

Feature detection is performed using the technique of SIFT detection, or Scale-Invariant Feature Transform. It is done by detecting key points in an image that are based on scale, rotation and illumination. A gaussian filter is applied to each frame, and the difference of

gaussian is calculated for each frame. The key points are represented by the local maxima and local minima. Once the keypoints are obtained using the SIFT detector, feature extraction for each keypoint is calculated by measuring the orientation of the gradient.

Feature Matching

It is a technique used to find similar features between two images. Brute force algorithm is one such algorithm that is used to find the best match between features in two images. Each feature is compared using a distance metric such as Euclidean distance or Hamming distance. This algorithm selects the best match that has the smallest distance to the corresponding feature in the other image.

Results

The output is not the ideal panoramic image. It could be due to the inaccuracy of the homography estimation. The RANSAC algorithm might have failed to accurately estimate the homography matrix due to the presence of outliers in the matching process

I was able to obtain a perfect panoramic image using the cv2.Stictcher() method, but as the function is disallowed for this project, I have not used it for submission.