Project 3 Report

PRATIK SUNIL BHUJBAL

UID:117555295

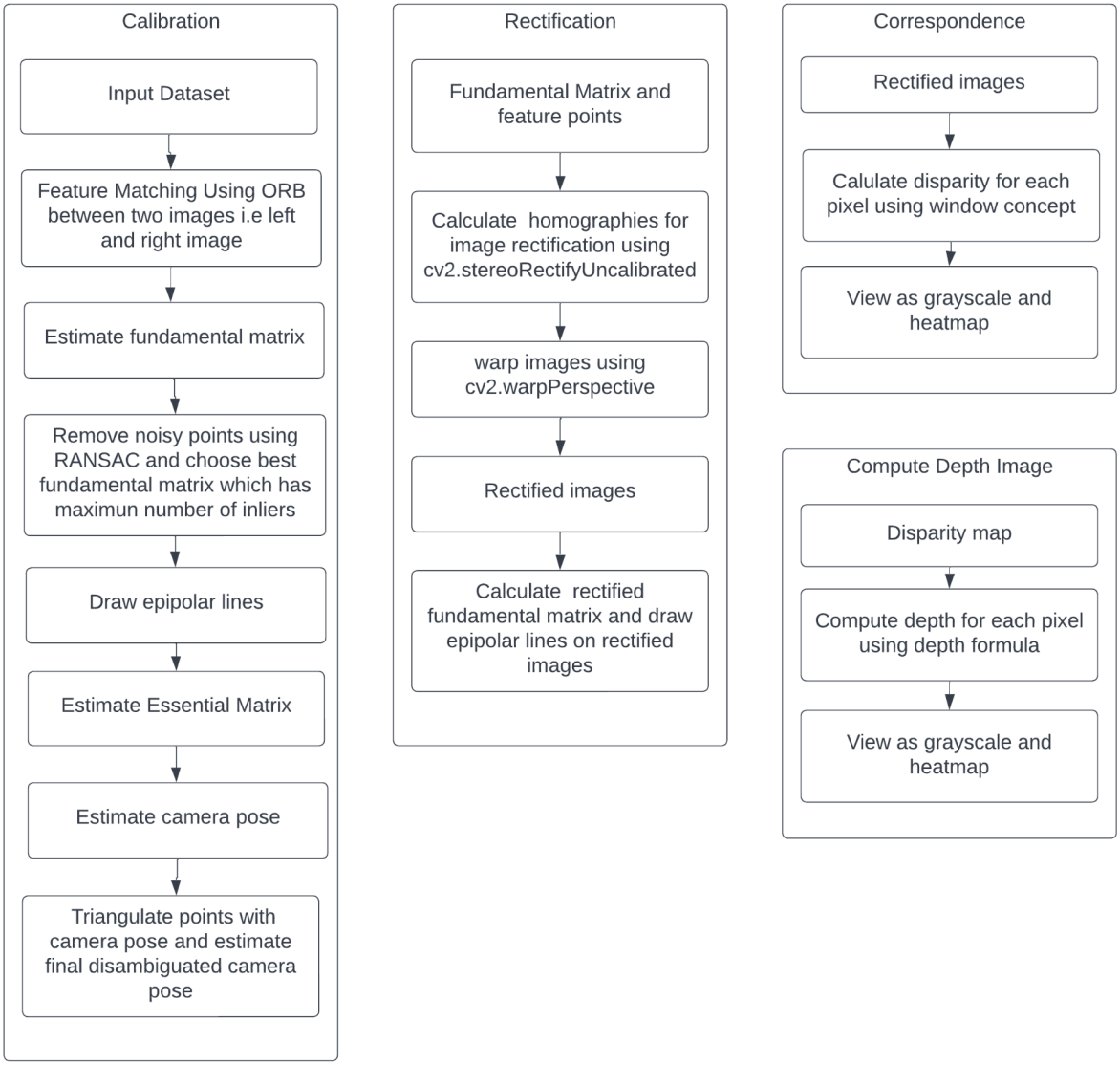

Fig1. Flowchart

CALIBRATION:

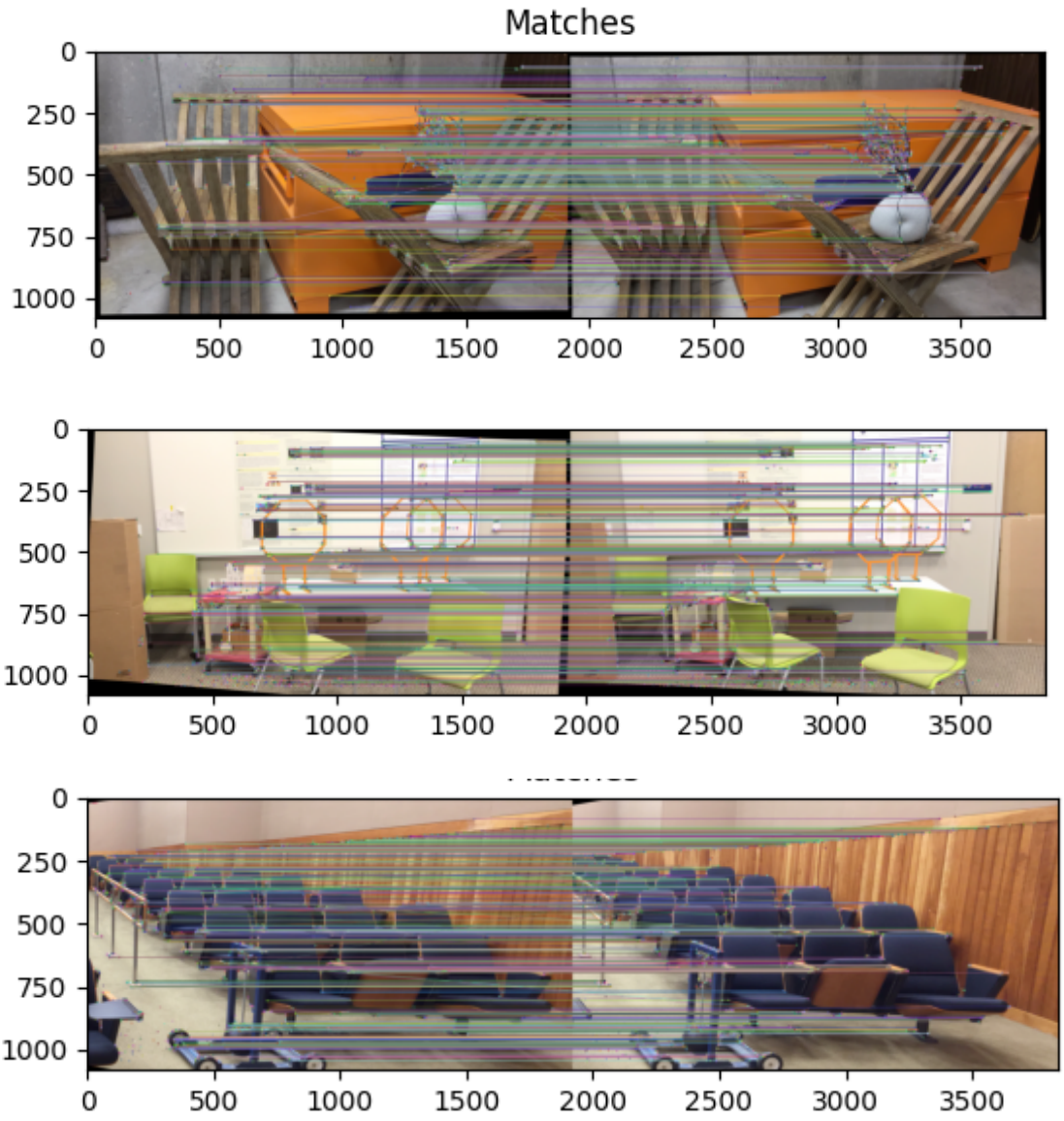

Fig2. Matches for datasets 1, 2, and 3

Fig 2 shows the first 1000 matches.

First, convert the images to a grayscale for feature matching using the ORB feature detector. Then estimate the fundamental matrix using the 8-point algorithm and remove outliers using RANSAC. Draw epipolar lines using those points. Estimate the essential matrix using: $\mathbf{E=K^{T}FK}$ where K is the camera calibration matrix and F is the Fundamental matrix.

Estimate four different possible camera poses from the essential matrix and use these to triangulate 3d points and find the final disambiguated camera pose using cheieality condition

$R3(X - C) > 0$ where R3 is 3rd column of a rotation matrix, X is the point in homogeneous coordinated and C is the camera center.

Camera Pose for dataset 1:

R = [[-1.39924219e-01 -9.90162216e-01 9.55989411e-14]
[-1.38547674e-01 1.95787869e-02 -9.90162216e-01]
[9.80421213e-01 -1.38547674e-01 -1.39924219e-01]]
C = [-1.00000000e+00 1.39888101e-14 -9.74220704e-14]

Camera Pose for dataset 2:

R = [[5.79188830e-01 -1.07492312e-02 -8.15122539e-01]
[-3.93398647e-04 -9.99916629e-01 1.29066275e-02]
[-8.15193317e-01 -7.15470636e-03 -5.79144771e-01]]
C = [0.88951481 0.00686536 -0.45685475]

Camera Pose for dataset 3:

R = [[9.99999865e-01 4.35501140e-04 2.84799834e-04]
[-4.33686320e-04 9.99979798e-01 -6.34157894e-03]
[-2.87555845e-04 6.34145456e-03 9.99979851e-01]]
C = [-0.99902953 0.00375803 -0.0438848]

RECTIFICATION:

Calculate homographies for image rectification using cv2.stereoRectifyUncalibrated. Transform the feature points in homogeneous coordinates using H matrices. And warp both images to make the epilines horizontal. Refer to figure 3 and figure 4 for comparison.

Homography Matrices for dataset 1:

H1 = [[2.28281222e-03 -1.29385148e-04 -1.95398323e-01]
[4.15104480e-14 2.30471216e-03 7.01660952e-14]
[7.69295612e-17 7.67584088e-17 2.30471216e-03]]
H2 = [[1.00000000e+00 1.42739532e-14 -7.73070497e-12]
[-1.42739532e-14 1.00000000e+00 1.36424205e-11]
[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Homography Matrices for dataset 2:

H1 = [[-1.59063451e-03 1.85840362e-04 1.53030027e-01]
[-2.65136872e-04 -1.36479727e-03 2.55188049e-01]
[-2.21900112e-07 3.03468088e-08 -1.18277453e-03]]
H2 = [[1.14577003e+00 -1.20860697e-01 -7.46744519e+01]
[1.90001463e-01 9.85505911e-01 -1.74574597e+02]
[1.57591137e-04 -1.66233836e-05 8.57689135e-01]]

Homography Matrices for dataset 3:

H1 = [[1.90860814e-03 -2.96371950e-04 -3.46541686e-01]
[3.35592358e-04 1.50385917e-03 -3.28880908e-01]
[4.28296228e-07 -6.01902278e-08 1.13383539e-03]]

H2 = [[1.27256334e+00 -8.99436628e-02 -2.13091225e+02]
[2.25219873e-01 9.86576314e-01 -2.08962288e+02]
[2.86512278e-04 -2.02504370e-05 7.35883450e-01]]

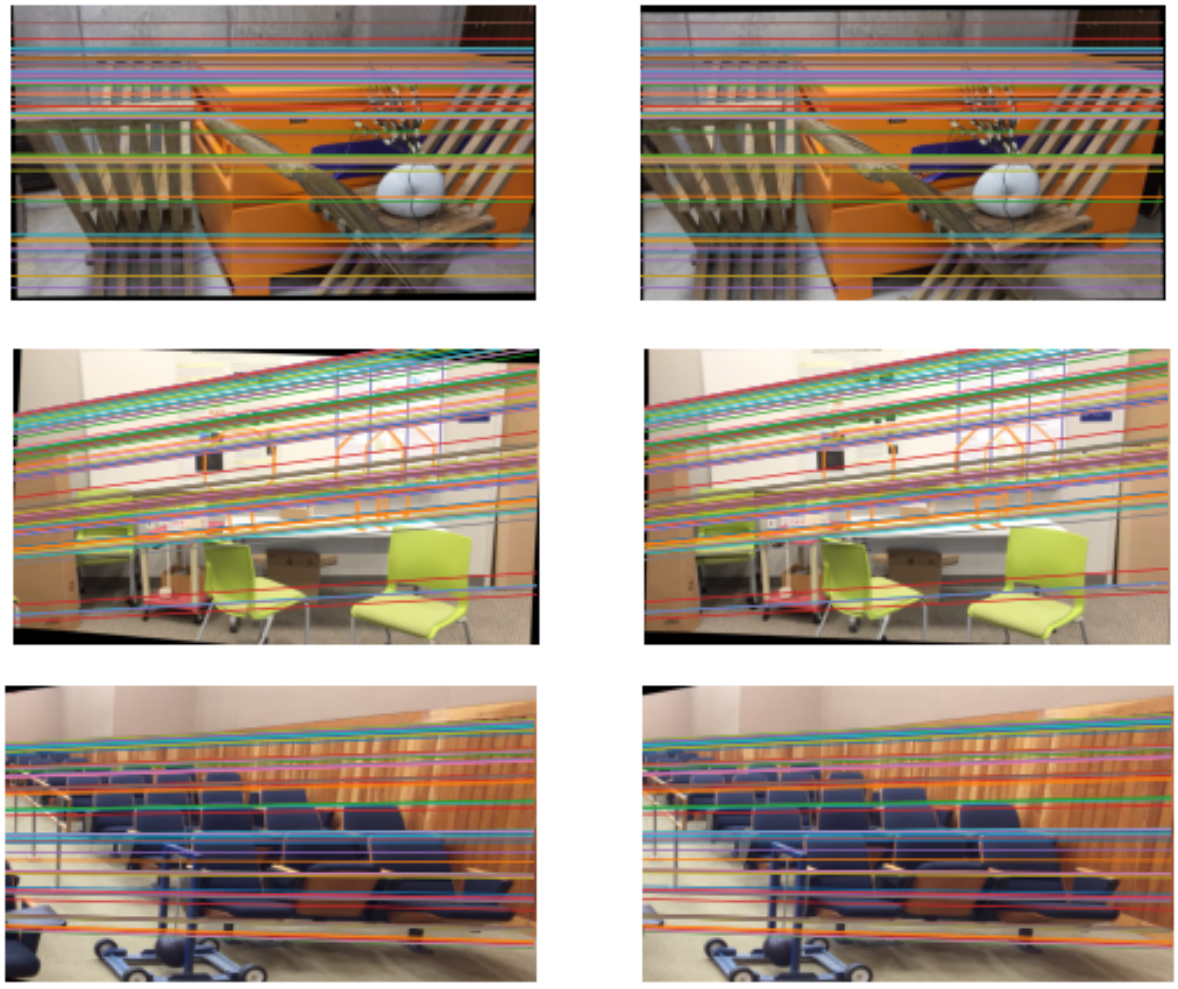

Fig3. Epipolar lines for unrectified images

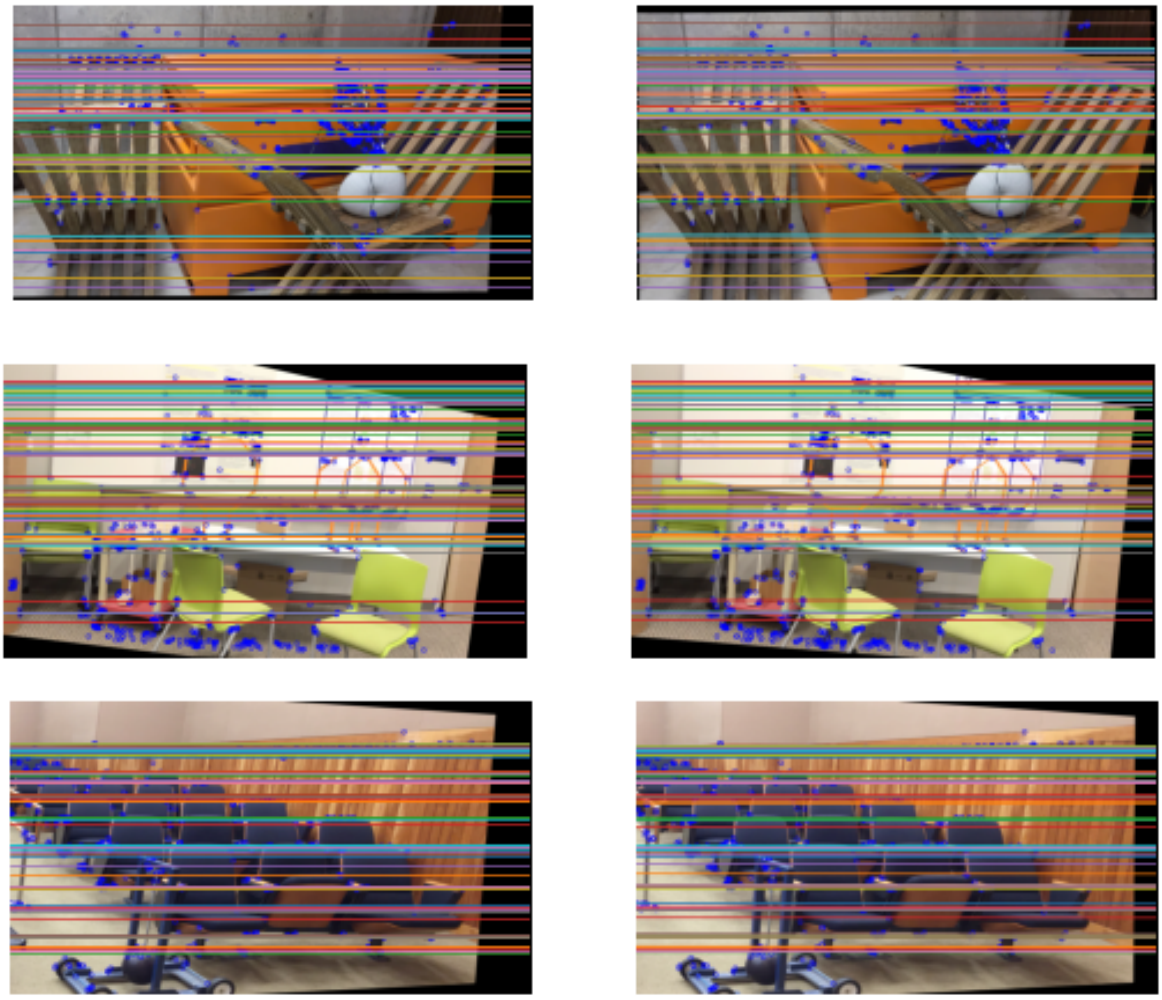

Fig4. Epipolar lines for rectified images

CORRESPONDENCE:

For each pixel in the image calculate the disparity using a sliding window and rescale the disparity from 0-255 and save the grayscale and heatmap images. First search for the best matching pixel in the right image for a particular pixel in the left image and then calculate the SSD of blocks and find the index with the minimum difference in the block intensity values

Results:

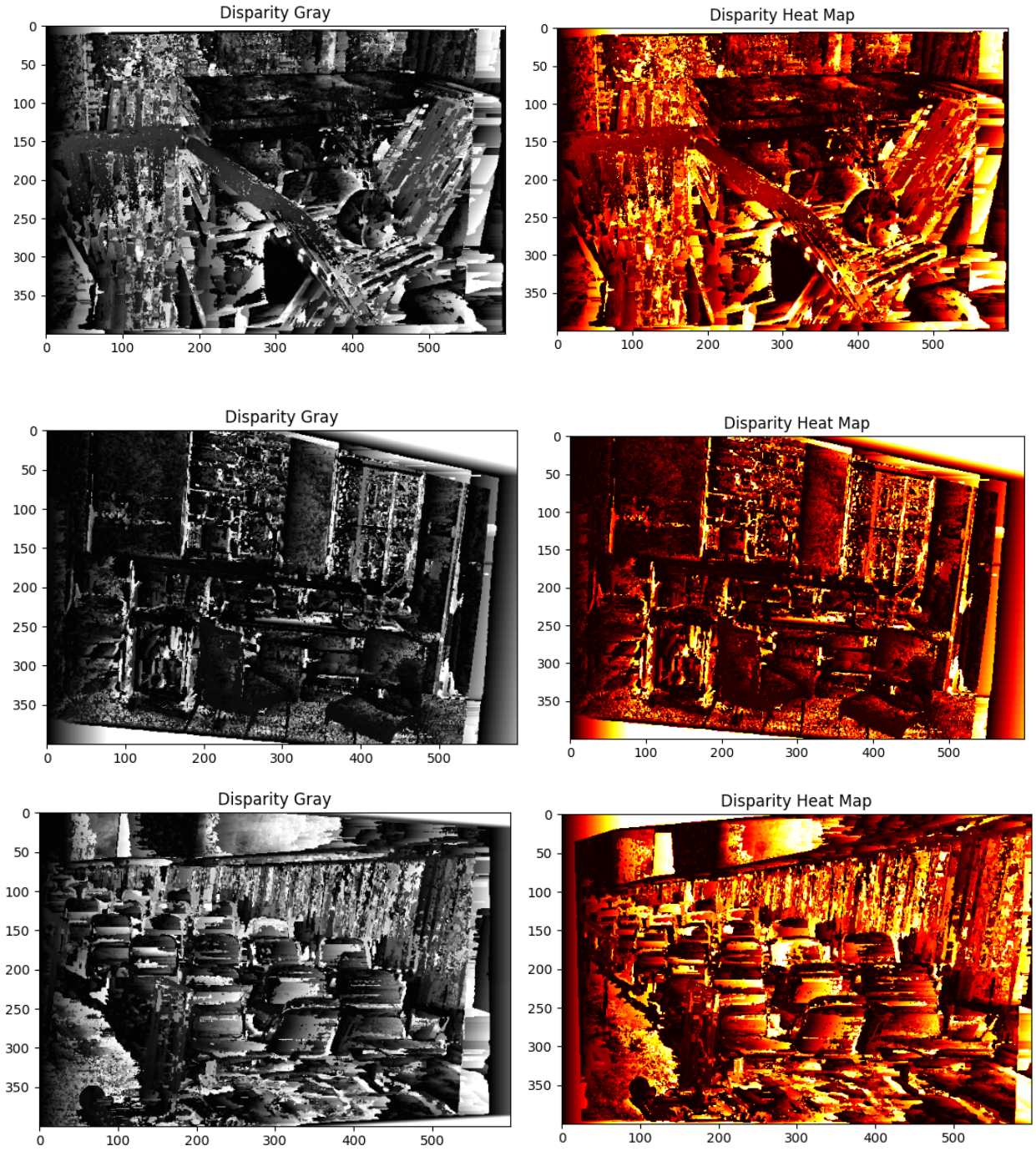

Fig5. Disparity Gray and Heat Map

COMPUTE DEPTH IMAGE:

Using the disparity map compute the depth using the below formula:

$$\text{Depth} = \frac{\text{baseline} * f}{\text{Disparity}}$$

Results:

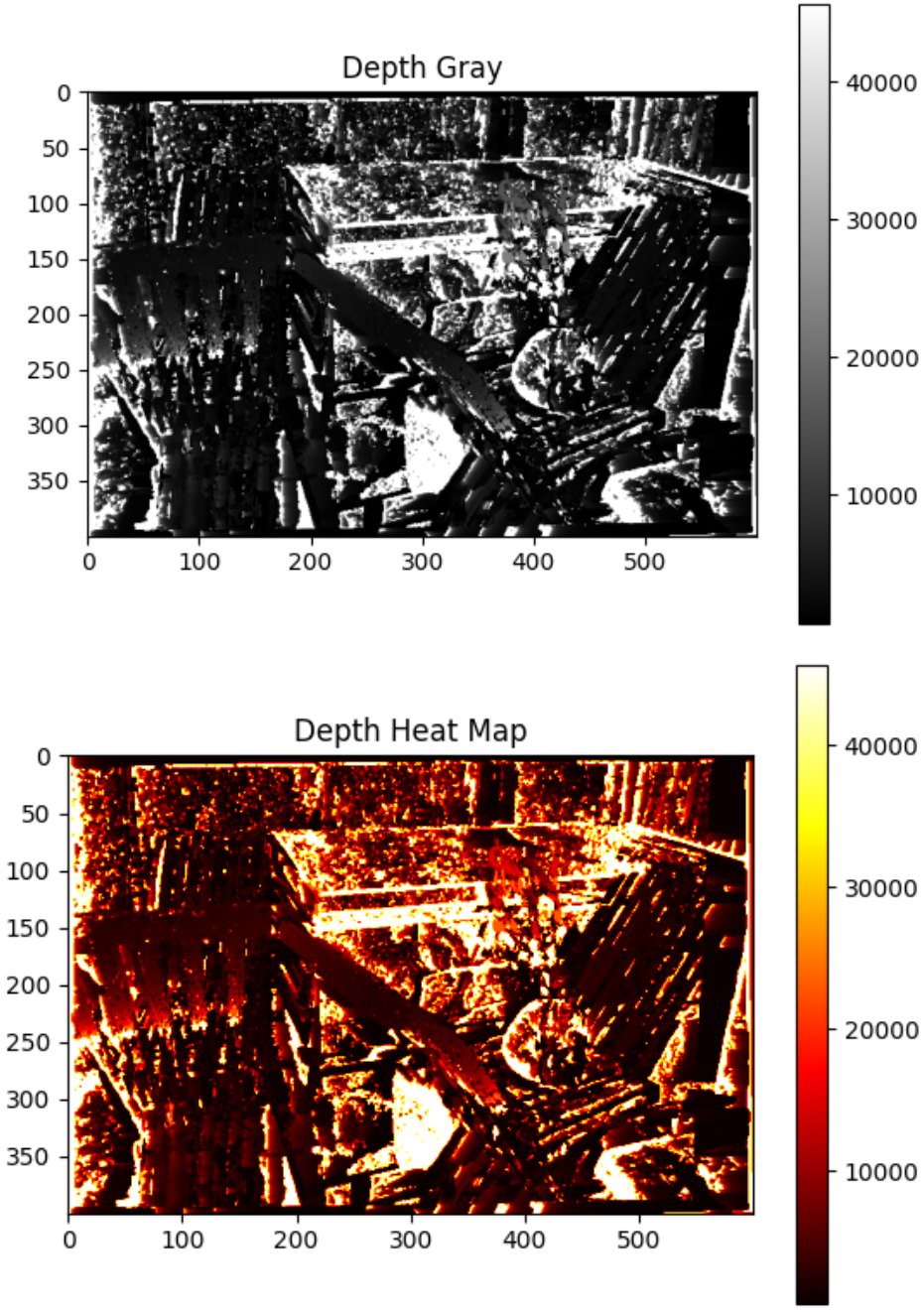

Fig6. Depth gray and Heat map for dataset 2

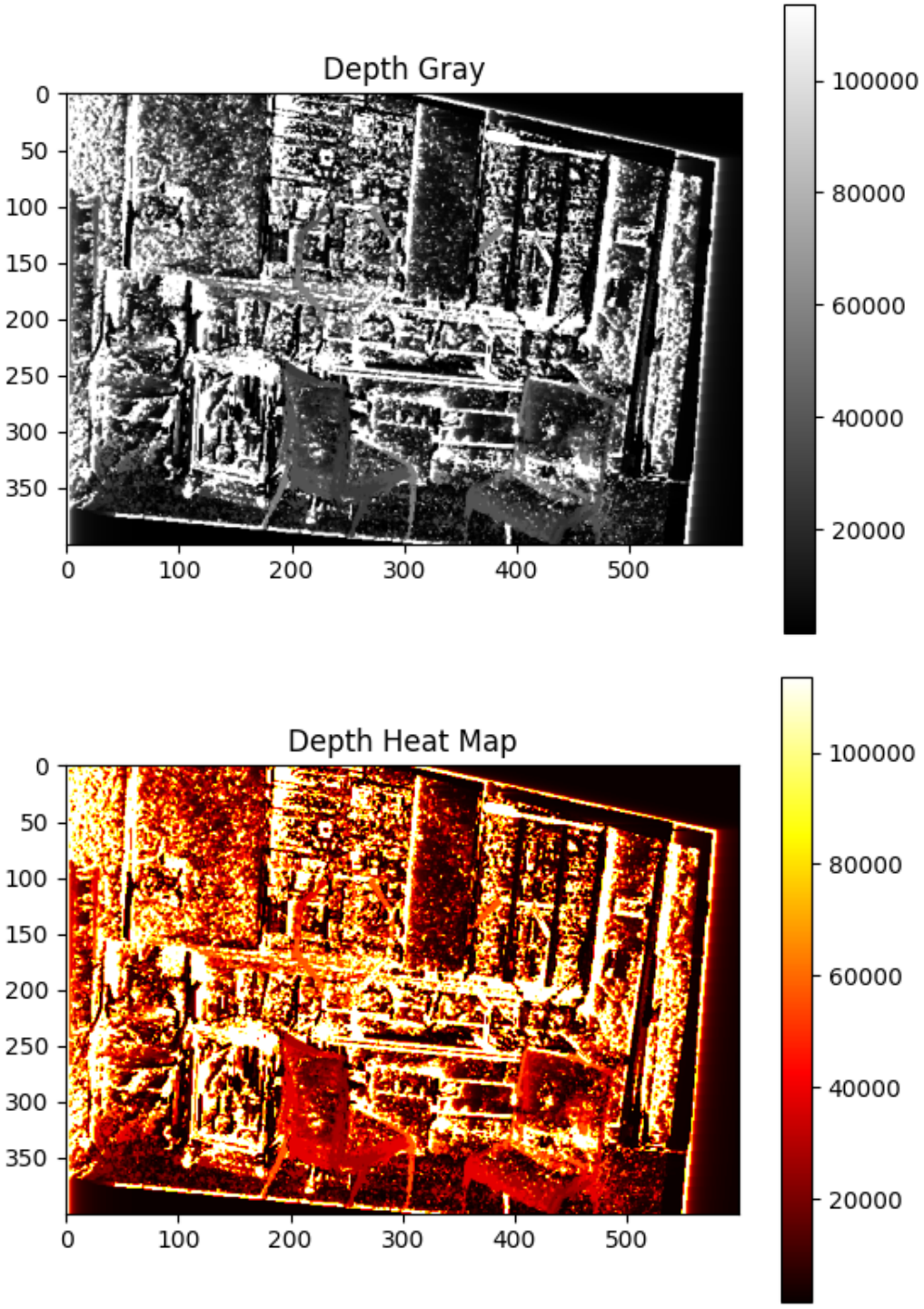

Fig7. Depth gray and Heat map for dataset 2

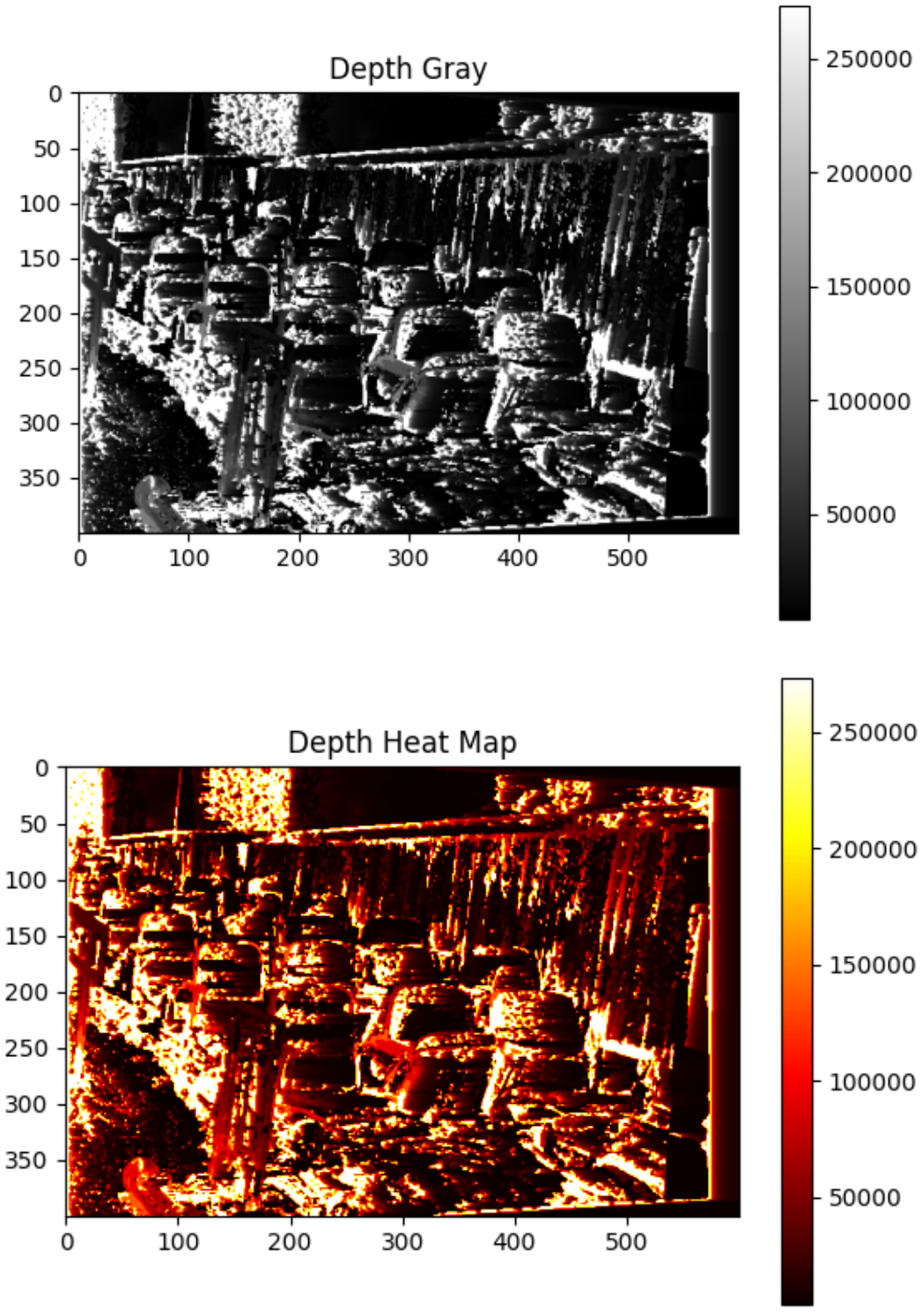

Fig8. Depth gray and Heat map for dataset 2

Results-link:

Link for all results - https://drive.google.com/drive/folders/1dojDawr0XPYM93OYG142GtI9eQYBduca?usp=sharing

References:

[1] https://cmsc733.github.io/2022/proj/p3/
[2] https://matplotlib.org/3.3.3/tutorials/colors/colormaps.html