Project - 3: Stereo Vision

Perception for Autonomous Robots

April 19, 2022

Instructors:
Samer Charifa

Semester:
Spring 2022

Course code:
ENPM 673

Student:
Joseph Pranadeer Reddy Katakam
UID: 117517958

Contents

List of Figures

1 Introduction and Project Constraints

- **Stereo Vision**: A camera can be seen as the function that maps 3-D world to a 2-D image. Computer stereo vision is the extraction of 3D information from digital images, such as those obtained by a CCD camera.
- **Project Description**: In this project, we are going to implement the concept of Stereo Vision. We will be given 3 different data sets, each of them contains 2 images of the same scenario but taken from two different camera angles. By comparing the information about a scene from 2 vantage points, we can obtain the 3D information by examining the relative positions of objects.
- **Project Resource**: MiddleBury Stereo Dataset
- `Disallowed Functions`:
  1. Any inbuilt function that computes the fundamental and essential matrices directly.
  2. Any other inbuilt function that directly computes the disparity or stereo correspondences.
  3. Not allowed to use in-built functions unless instructed.
- `Pipeline`:

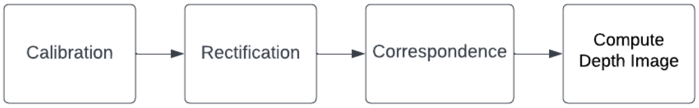


Figure 1: Pipeline for Stereo Vision

2 Calibration

2.1 Feature Matching

- First, we need to obtain matching features of both the images.
- Here, SIFT feature detector was used. SIFT helps locate the local features in an image, commonly known as the key points' of the image. These key points are scale rotation invariant that can be helpful.
- Next these feature points are matched from both the images using Brute-force Matcher.
- The lower distance between the descriptors is chosen for greater efficiency. Thus the correspondences are obtained.

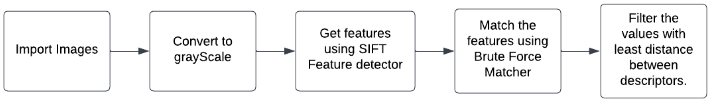

Import Images
Convert to grayScale
Get features using SIFT Feature detector
Match the features using Brute Force Matcher
Filter the values with least distance between descriptors.

Figure 2: Feature Matching Pipeline

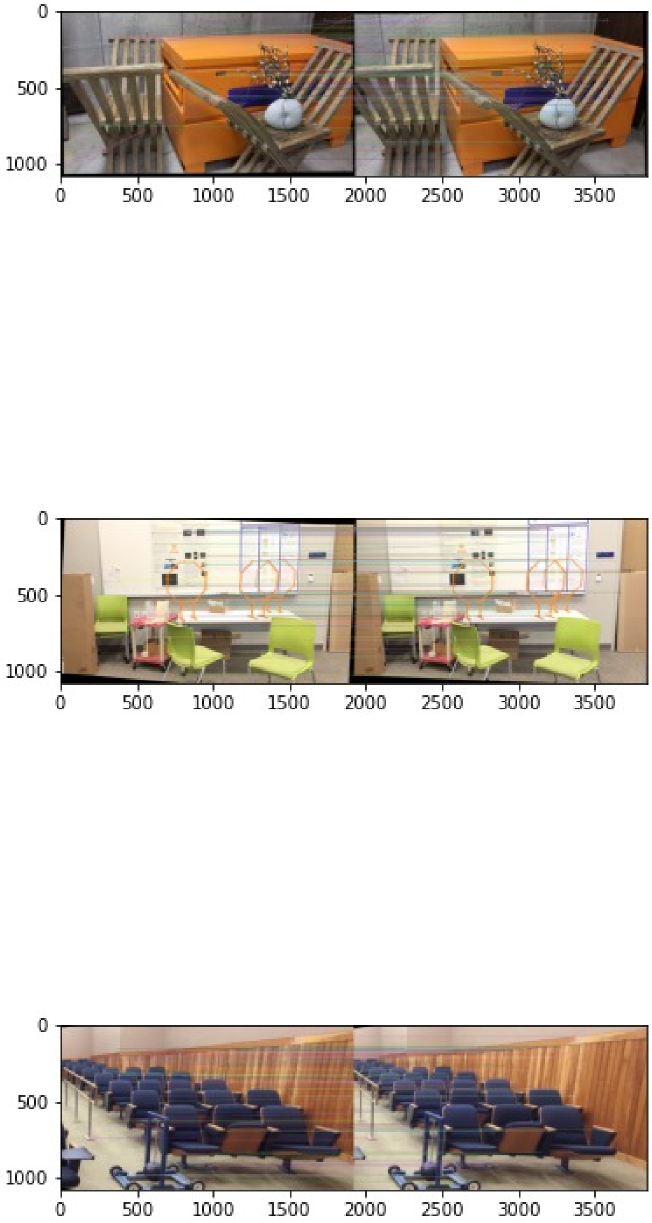


Figure 3: SIFT features matched using Brute-Force matcher for the given Data sets

2.2 Estimating Fundamental Matrix

- **Fundamental Matrix**: The fundamental matrix gives the relationship between any two images of the same scene that constrains where the projection of points from the scene can occur in both images i.e. it gives the relationship between the feature correspondences of these stereo images.

- The F matrix is only an algebraic representation of epipolar geometry.

- To obtain the fundamental matrix, 8-point Hartley algorithm is used. The eight-point algorithm is an algorithm used in computer vision to estimate the essential matrix or the fundamental matrix related to a stereo camera pair from a set of corresponding image points.

$$x_i'^T * F * x_i = 0 \tag{1}$$

  where i=1,2,....,m. This is known as epipolar constraint or correspondence condition. Since, F is a 3×3 matrix, we can set up a homogeneous linear system with 9 unknowns.

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_11 & f_12 & f_13 \\ f_21 & f_22 & f_23 \\ f_31 & f_32 & f_33 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0 \tag{2}$$

$$x_i{*}x_i{*}f_11+x_i{*}y_i{*}f_21+x_i{*}f_31+y_i{*}xi{*}f_12+y_i{*}y_i{*}f_22+y_i{*}f_32+x_i{*}f_13+y_i{*}f_23+f_33 = 0 \tag{3}$$

- The solution for the fundamental matrix is computed by solving the homogeneous system of equations given by

$$A * F = 0 \tag{4}$$

  where f is the 9 × 1 vector which is later reshaped to a 3 × 3 fundamental matrix F'.

- F can be obtained using singular value decomposition. The rank of the F matrix must be 2, However, due to noise in the correspondences, the estimated F matrix can be of rank 3.

- So, to enforce the rank 2 constraint, the last singular value of the estimated F must be set to zero. If F has a full rank then it will have an empty null-space i.e. it won't have any point that is on the entire set of lines. Thus, there wouldn't be any epipoles.

- If the F matrix estimation is good, the terms Homogeneous equation should be close to 0, where x1 and x2 are features from image1 and image2.

- Since the point correspondences are computed using SIFT or some other feature descriptors, the data is bound to be noisy and (in general) contains several outliers. Thus, to remove these outliers, we use the RAN SAC algorithm (Yes! The same as used in Panorama stitching!) to obtain a better estimate of the fundamental matrix. So, out of all possibilities, the F matrix with maximum number of inliers is chosen.

2.3 Estimating Essential Matrix

- **Essential Matrix**: The Essential Matrix is a 3 x 3 matrix that encodes epipolar geometry. Given a point in one image, multiplying by the essential matrix will tell us the epipolar line in the second view.

- We already know the intrinsic parameters of the cameras(K1 and K2). The Essential matrix can be estimated as:

  $$E = K^T * F * K, \tag{5}$$

  where K is the calibration matrix or camera intrinsic Matrix.

- Relative camera poses between two views can be computed using the E matrix.

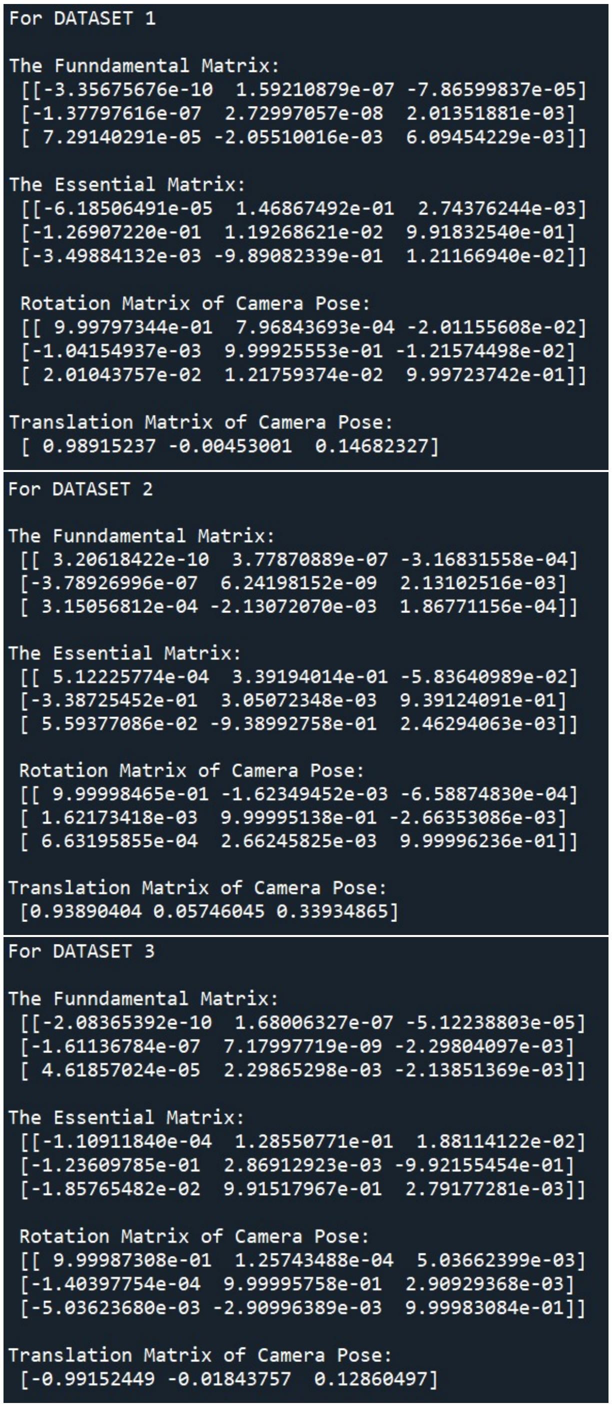
2.4 Estimation of Camera Pose

- The camera pose consists of 6 degrees-of-freedom (DOF) Rotation (Roll, Pitch, Yaw) and Translation (X, Y, Z) of the camera with respect to the world.

- The essential matrix E can be decomposed to 4 mathematically possible rotation and translation matrices.

- Given their triangulated points, finding unique came posed can be accomplished by checking the cheirality condition i.e. the reconstructed points must be in front of the cameras. To check the cheirality condition, triangulate the 3D points (given two camera poses) using linear least squares to check the sign of the depth Z in the camera coordinate system w.r.t. camera center.

- The best camera configuration, (C,R,X) is the one that produces the maximum number of points satisfying the cheirality condition.

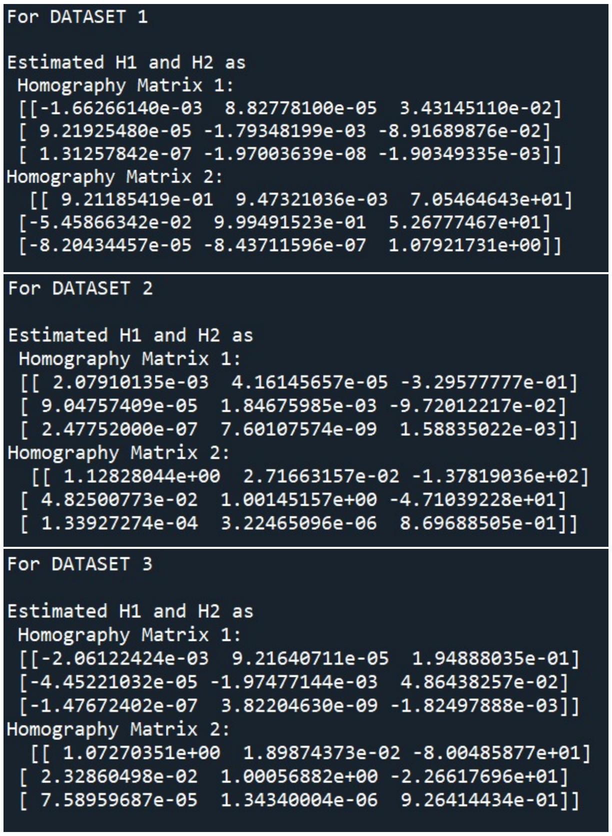


```
For DATASET 1

The Funndamental Matrix:
 [[-3.35675676e-10  1.59210879e-07 -7.86599837e-05]
 [-1.37797616e-07  2.72997057e-08  2.01351881e-03]
 [ 7.29140291e-05 -2.05510016e-03  6.09454229e-03]]

The Essential Matrix:
 [[-6.18506491e-05  1.46867492e-01  2.74376244e-03]
 [-1.26907220e-01  1.19268621e-02  9.91832540e-01]
 [-3.49884132e-03 -9.89082339e-01  1.21166940e-02]]

 Rotation Matrix of Camera Pose:
 [[ 9.99797344e-01  7.96843693e-04 -2.01155608e-02]
 [-1.04154937e-03  9.99925553e-01 -1.21574498e-02]
 [ 2.01043757e-02  1.21759374e-02  9.99723742e-01]]

Translation Matrix of Camera Pose:
 [ 0.98915237 -0.00453001  0.14682327]
```

```
For DATASET 2

The Funndamental Matrix:
 [[ 3.20618422e-10  3.77870889e-07 -3.16831558e-04]
 [-3.78926996e-07  6.24198152e-09  2.13102516e-03]
 [ 3.15056812e-04 -2.13072070e-03  1.86771156e-04]]

The Essential Matrix:
 [[ 5.12225774e-04  3.39194014e-01 -5.83640989e-02]
 [-3.38725452e-01  3.05072348e-03  9.39124091e-01]
 [ 5.59377086e-02 -9.38992758e-01  2.46294063e-03]]

 Rotation Matrix of Camera Pose:
 [[ 9.99998465e-01 -1.62349452e-03 -6.58874830e-04]
 [ 1.62173418e-03  9.99995138e-01 -2.66353086e-03]
 [ 6.63195855e-04  2.66245825e-03  9.99996236e-01]]

Translation Matrix of Camera Pose:
 [0.93890404 0.05746045 0.33934865]
```

```
For DATASET 3

The Funndamental Matrix:
 [[-2.08365392e-10  1.68006327e-07 -5.12238803e-05]
 [-1.61136784e-07  7.17997719e-09 -2.29804097e-03]
 [ 4.61857024e-05  2.29865298e-03 -2.13851369e-03]]

The Essential Matrix:
 [[-1.10911840e-04  1.28550771e-01  1.88114122e-02]
 [-1.23609785e-01  2.86912923e-03 -9.92155454e-01]
 [-1.85765482e-02  9.91517967e-01  2.79177281e-03]]

 Rotation Matrix of Camera Pose:
 [[ 9.99987308e-01  1.25743488e-04  5.03662399e-03]
 [-1.40397754e-04  9.99995758e-01  2.90929368e-03]
 [-5.03623680e-03 -2.90996389e-03  9.99983084e-01]]

Translation Matrix of Camera Pose:
 [-0.99152449 -0.01843757  0.12860497]
```

Figure 4: Fundamental, Essential and Extrinsic's for the given Data sets



```
For DATASET 1

Estimated H1 and H2 as
 Homography Matrix 1:
 [[-1.66266140e-03  8.82778100e-05  3.43145110e-02]
 [ 9.21925480e-05 -1.79348199e-03 -8.91689876e-02]
 [ 1.31257842e-07 -1.97003639e-08 -1.90349335e-03]]
Homography Matrix 2:
  [[ 9.21185419e-01  9.47321036e-03  7.05464643e+01]
 [-5.45866342e-02  9.99491523e-01  5.26777467e+01]
 [-8.20434457e-05 -8.43711596e-07  1.07921731e+00]]
```

```
For DATASET 2

Estimated H1 and H2 as
 Homography Matrix 1:
 [[ 2.07910135e-03  4.16145657e-05 -3.29577777e-01]
 [ 9.04757409e-05  1.84675985e-03 -9.72012217e-02]
 [ 2.47752000e-07  7.60107574e-09  1.58835022e-03]]
Homography Matrix 2:
  [[ 1.12828044e+00  2.71663157e-02 -1.37819036e+02]
 [ 4.82500773e-02  1.00145157e+00 -4.71039228e+01]
 [ 1.33927274e-04  3.22465096e-06  8.69688505e-01]]
```

```
For DATASET 3

Estimated H1 and H2 as
 Homography Matrix 1:
 [[-2.06122424e-03  9.21640711e-05  1.94888035e-01]
 [-4.45221032e-05 -1.97477144e-03  4.86438257e-02]
 [-1.47672402e-07  3.82204630e-09 -1.82497888e-03]]
Homography Matrix 2:
  [[ 1.07270351e+00  1.89874373e-02 -8.00485877e+01]
 [ 2.32860498e-02  1.00056882e+00 -2.26617696e+01]
 [ 7.58959687e-05  1.34340004e-06  9.26414434e-01]]
```

Figure 5: Homography Matrices of the given Data sets

3 Rectification

- Rectification is a transformation which makes pairs of conjugate epipolar lines become collinear and parallel to the horizontal axis (i.e., baseline).
- It warps the input images (perspective transformation) so that epipolar lines are horizontal.
- Searching for corresponding points becomes much simpler for the case of rectified images.
- Using the fundamental matrix and the feature points, we can obtain the epipolar lines for both the images. The epipolar lines need to be parallel for further computations to obtain depth. This can be done by re-projecting image planes onto a common plane parallel to the line between camera centers.
- Opencv offers a rectifying function that applies homographic transformation to project the image plane of each camera onto a perfectly aligned virtual plane. This transformation is computed from a set of matched points and a fundamental matrix.
- Hence, the homography matrices H1 and H2 are calculated for the left and right camera images using the function cv2.stereoRectifyUncalib() which intakes only the estimated fundamental matrix and point correspondences.

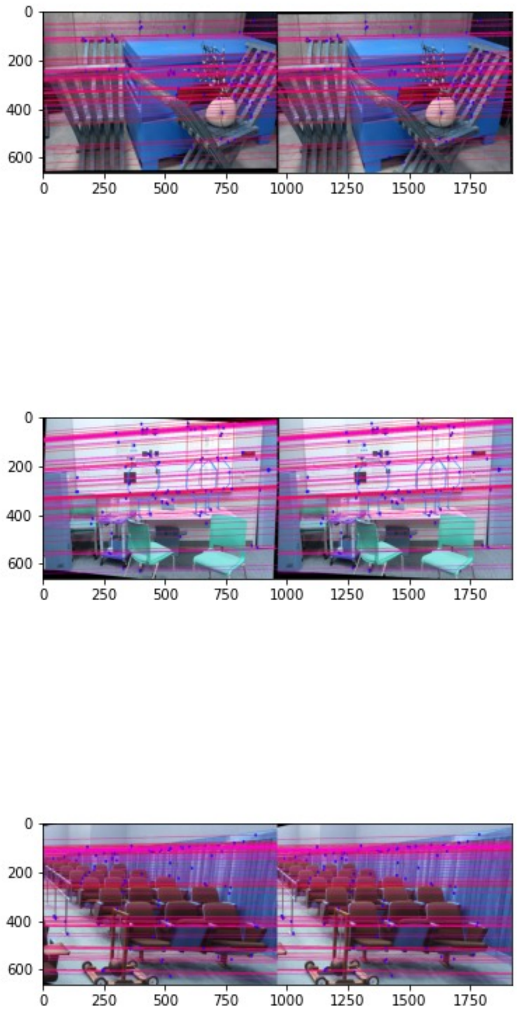


Figure 6: Epipolar lines for unrectified images of the given Data sets

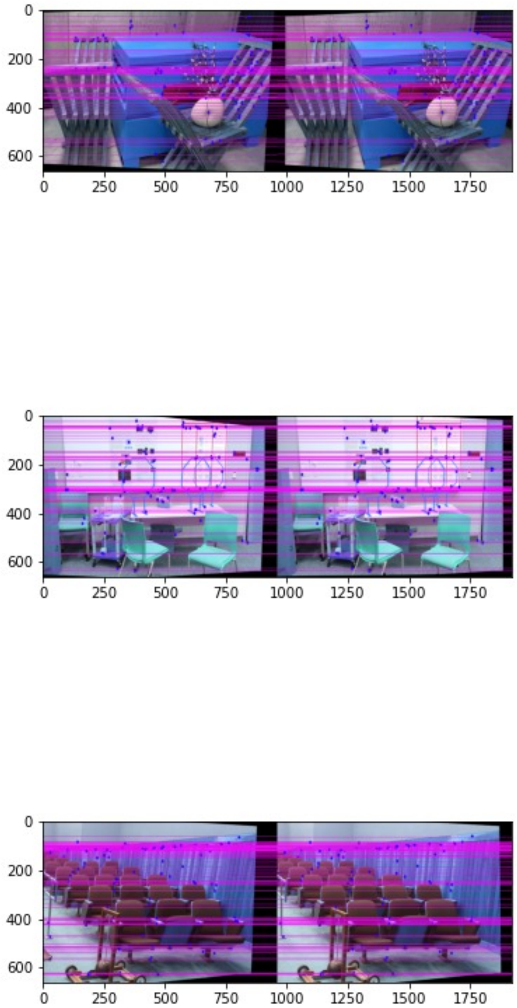


Figure 7: Epipolar lines for rectified images of the given Data sets

4 Correspondence

- When trying to find a corresponding matching for pixel in image 1 to image 2 along epipolar line, Epipolar geometry reduces the search to a line.
- A Sliding window operation is performed to find the nearest corresponding match from image 1 to image 2.
- To perform the above operation, Sum of Absolute difference is used.
- After a lot of trial and error, I have fixed on a window size and threshold value for each dataset to obtain accurate results.

5 Compute Depth Image

- **Disparity Map**: Disparity is the difference between the optical center of left image and the location of point in the world
- An exact match would yield a result zero.
- After we get the matching pixel location, the disparity can be found but take the absolute of the difference between the source and matched pixel location.
- The depth of image can be found by the length of line joining the camera centers which contains the epipoles, the focal length and the disparity.
- **Depth Map Formula**:If we know the focal length(f) and baseline(b), the depth can be calculated as:

$$depth = \frac{focallength.baseline}{disparity} \tag{6}$$

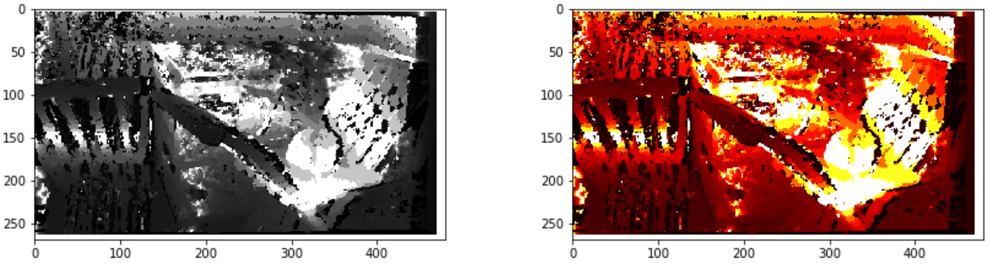


Figure 8: Gray and Heat map of Depth Image for Data-set 1

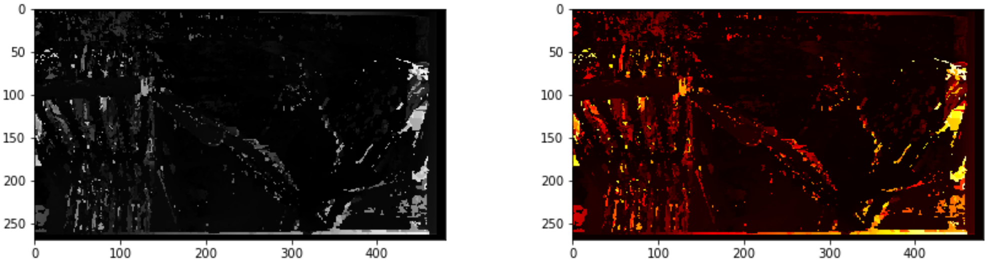


Figure 9: Gray and Heat map of Disparity Image for Data-set 1

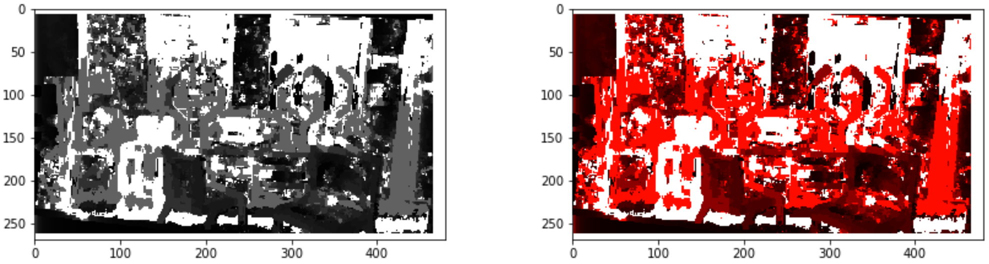


Figure 10: Gray and Heat map of Depth Image for Data-set 2

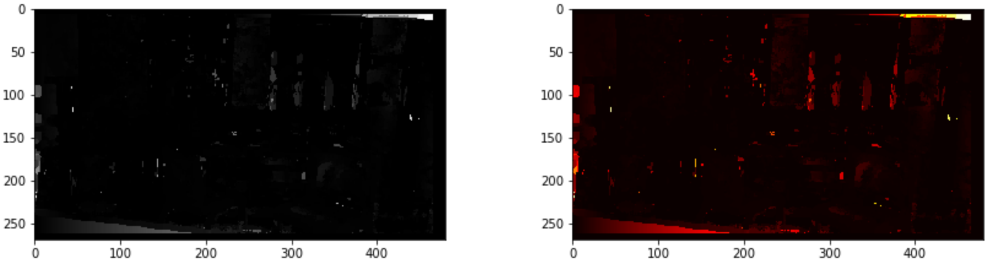


Figure 11: Gray and Heat map of Disparity Image for Data-set 2

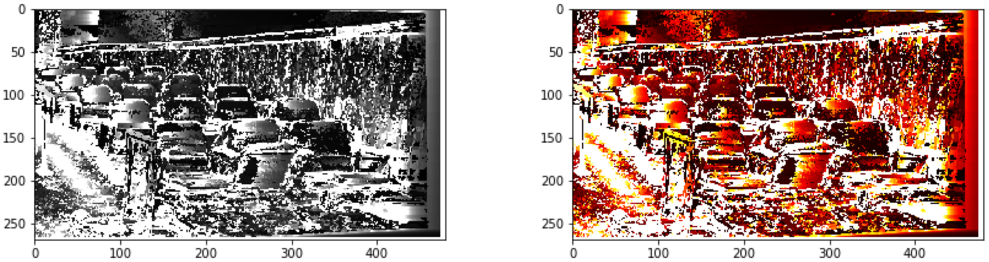


Figure 12: Gray and Heat map of Depth Image for Data-set 3

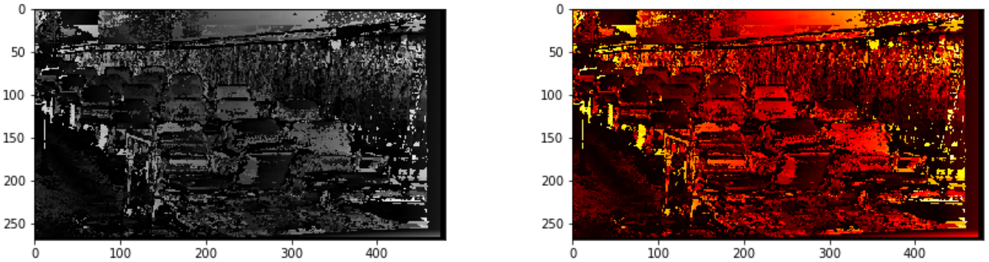


Figure 13: Gray and Heat map of Disparity Image for Data-set 3

6 Resources

- Structure from Motion
- Essential Matrix
- 3D Reconstruction of Vision
- Epipolar Geometry

7 Output Images Link

- Project Output Images (Click Here)