Abhijit Vinod Mahalle 117472288
ENPM673 Project 3 Report

Part 1 (Calibration):

1. The task here was to find the Fundamental Matrix, Essential Matrix, and rotation and translation matrices.
2. Best features were detected in the given two images using SIFT.
3. Best fundamental matrix was calculated using 8 random feature points and RANSAC such that the best fundamental matrix satisfies the epipolar constraint for maximum number of points. This reduces the effect of noise in the two images.
4. Essential matrix was calculated using calibration matrices of two cameras and fundamental matrix.
5. Essential matrix was decomposed to yield rotation and translation matrices for four camera configurations.
6. Four projection matrices were calculated using four configurations of rotation and translation.
7. 2D features were transformed into 3D world points for each projection matrix.
8. Each of the 3D points were then checked for positive depth using Cheirality condition.
9. The rotation and translation matrices that yield highest number of points with positive depth using the above condition was selected as the best configuration.

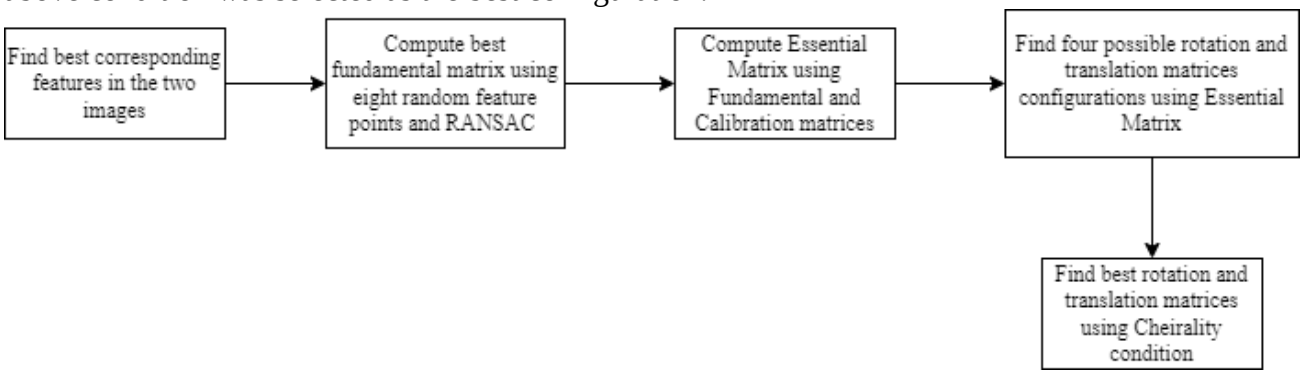


Fig. 1: Block diagram to find rotation and translation

Four configurations of Rotation and Translation matrices:

Curule dataset:

1.

```
([array([[ 0.99997721, -0.00160019, -0.00655899],
         [ 0.00161467,  0.99999627,  0.00220239],
         [ 0.00655544, -0.00221293,  0.99997606]]),

  array([ 0.99806468, -0.00225318, -0.06214357])
```

2.

```
 array([[ 0.99997721, -0.00160019, -0.00655899],
        [ 0.00161467,  0.99999627,  0.00220239],
        [ 0.00655544, -0.00221293,  0.99997606]])

 array([-0.99806468,  0.00225318,  0.06214357])
```

3.

```
array([[ 0.99142314, -0.00581093, -0.1305618 ],
       [-0.00611035, -0.99997954, -0.00189283],
       [-0.13054813,  0.00267437, -0.99143837]])

 array([ 0.99806468, -0.00225318, -0.06214357])
```

4.

```
array([[ 0.99142314, -0.00581093, -0.1305618 ],
       [-0.00611035, -0.99997954, -0.00189283],
       [-0.13054813,  0.00267437, -0.99143837]])

 array([-0.99806468,  0.00225318,  0.06214357])
```

Octagon dataset:

1.

```
array([[ 9.99999319e-01, -1.23184200e-04, -1.16095388e-03],
       [ 1.22771645e-04,  9.99999929e-01, -3.55422381e-04],
       [ 1.16099759e-03,  3.55279607e-04,  9.99999263e-01]])

       array([-0.9452613 , -0.00109926,  0.32631252])
```

2.

```
array([[ 9.99999319e-01, -1.23184200e-04, -1.16095388e-03],
       [ 1.22771645e-04,  9.99999929e-01, -3.55422381e-04],
       [ 1.16099759e-03,  3.55279607e-04,  9.99999263e-01]])

       array([ 0.9452613 ,  0.00109926, -0.32631252])
```

3.

```
array([[ 7.86321362e-01,  1.76205174e-03, -6.17815192e-01],
       [ 1.95456926e-03, -9.99998023e-01, -3.64394885e-04],
       [-6.17814613e-01, -9.21031102e-04, -7.86323251e-01]])

       array([-0.9452613 , -0.00109926,  0.32631252])
```

4.

```
array([[ 7.86321362e-01,  1.76205174e-03, -6.17815192e-01],
       [ 1.95456926e-03, -9.99998023e-01, -3.64394885e-04],
       [-6.17814613e-01, -9.21031102e-04, -7.86323251e-01]])

       array([ 0.9452613 ,  0.00109926, -0.32631252])
```

Pendulum dataset:

1.

```
array([[ 9.99997866e-01, -1.15503589e-03, -1.71309684e-03],
       [ 1.15395161e-03,  9.99999133e-01, -6.33786845e-04],
       [ 1.71382740e-03,  6.31808661e-04,  9.99998332e-01]])

       array([ 0.99792297,  0.04821734, -0.042718  ])
```

2.

```
array([[ 9.99997866e-01, -1.15503589e-03, -1.71309684e-03],
       [ 1.15395161e-03,  9.99999133e-01, -6.33786845e-04],
       [ 1.71382740e-03,  6.31808661e-04,  9.99998332e-01]])

       array([-0.99792297, -0.04821734,  0.042718  ])
```

3.

```
array([[ 0.99166334,  0.09503497, -0.08701827],
       [ 0.09507852, -0.99546307, -0.00365351],
       [-0.08697069, -0.00465052, -0.99620002]])


       array([ 0.99792297,  0.04821734, -0.042718  ])
```

4.

```
array([[ 0.99166334,  0.09503497, -0.08701827],
       [ 0.09507852, -0.99546307, -0.00365351],
       [-0.08697069, -0.00465052, -0.99620002]])


       array([-0.99792297, -0.04821734,  0.042718  ])
```

Rotation and translation matrices satisfying Cheirality condition:

Curule dataset:

```
array([[ 0.99997721, -0.00160019, -0.00655899],
       [ 0.00161467,  0.99999627,  0.00220239],
       [ 0.00655544, -0.00221293,  0.99997606]])


       array([ 0.99806468, -0.00225318, -0.06214357])
```

Octagon dataset:

```
array([[ 9.99999319e-01, -1.23184200e-04, -1.16095388e-03],
       [ 1.22771645e-04,  9.99999929e-01, -3.55422381e-04],
       [ 1.16099759e-03,  3.55279607e-04,  9.99999263e-01]])

       array([ 0.9452613 ,  0.00109926, -0.32631252])
```

Pendulum dataset:

```
array([[ 9.99997866e-01, -1.15503589e-03, -1.71309684e-03],
       [ 1.15395161e-03,  9.99999133e-01, -6.33786845e-04],
       [ 1.71382740e-03,  6.31808661e-04,  9.99998332e-01]])

       array([ 0.99792297,  0.04821734, -0.042718  ])
```

Challenges:

1. Not enough resources were available to decompose the Essential Matrix into rotation and translation matrices.

Part 2 (Rectification)

1. Equation of epipolar lines were found out using the inbuilt OpenCV function that takes feature points in two images and Fundamental matrix as input.
2. The start and end co-ordinates of these epipolar lines were calculated using their equations.
3. The epipolar lines were drawn on the original images and they pass through their corresponding feature points.
4. Homography between two images was calculated using inbuilt OpenCV function that takes feature points as input
5. Perspective transformation was applied to the two images using Homography such that feature points of the two images lie on the same line.
6. Transform co-ordinates of the epipolar lines in the warped image was computed using Homography.
7. Epipolar lines were drawn on the rectified image such that the corresponding lines were horizontal and in-line.

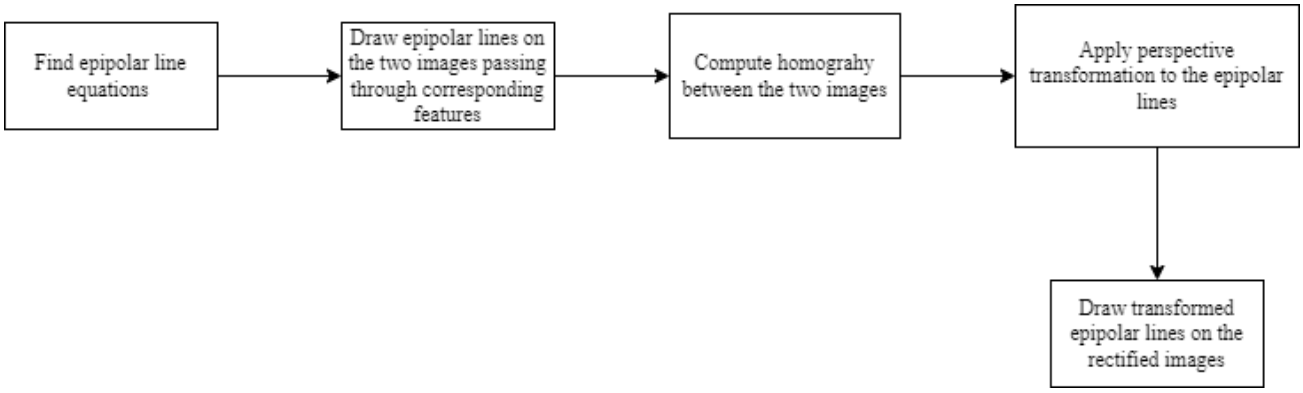


Fig. 2: Block diagram to rectify images and draw horizontal epipolar line

Results:

Fig. 3: Epipolar lines on unrectified image

Images after rectification

Fig. 4: Rectified images

Fig. 5: Rectified Epipolar lines

Challenges:

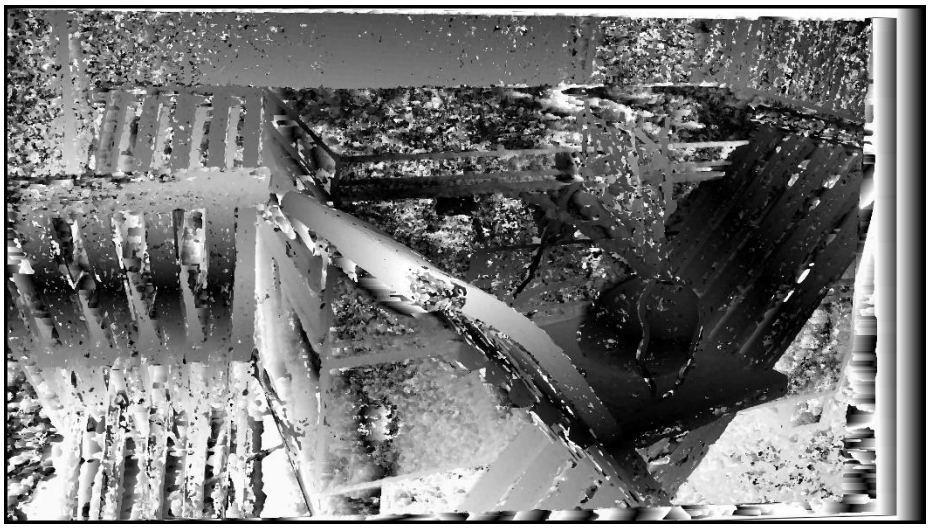
1. Epipolar lines on unrectified images were converging sometime.
2. It was realized that the epipolar lines are very sensitive to the values of the Fundamental Matrix
3. Elements with absolute value lower than 10^{-4} were clipped and the resulting output after that was satisfying.

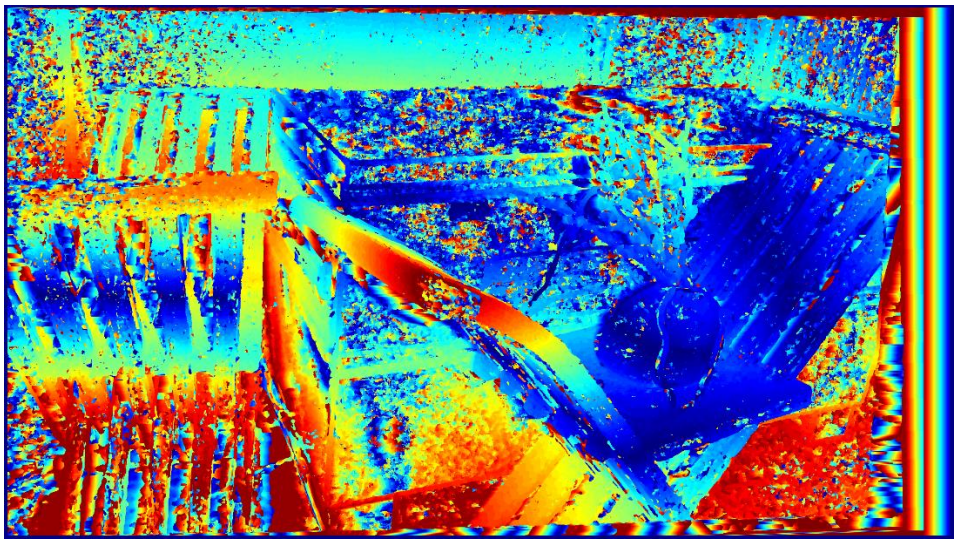
Part 3 (Correspondence)

1. Window of suitable size was selected at the top left corner of the left image. This window was then compared with every window of the corresponding row on the right image using sum of squared differences. The index of the window on the right image with lowest value of ssd was stored. The minimum ssd value was then assigned to the corresponding window position in the blank image.
2. The window on the top left corner was then moved by one window size to the right and the above process was repeated for all the windows of the left image.
3. The resulting disparity map was then scaled so that all the values were between 0 and 255.
4. The resulting image gave the Disparity between the two images.

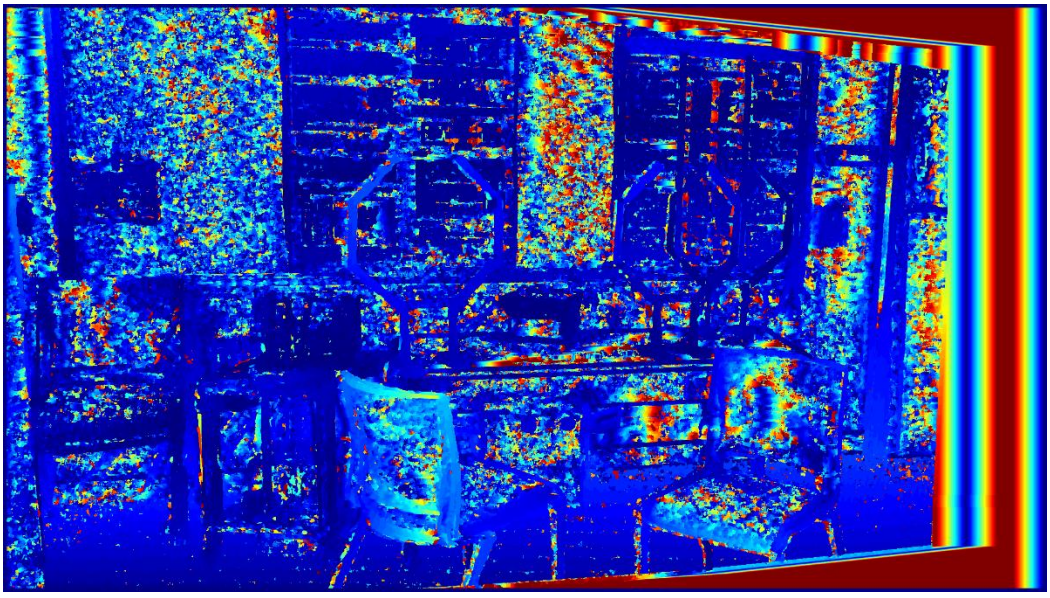
Results:

Curule

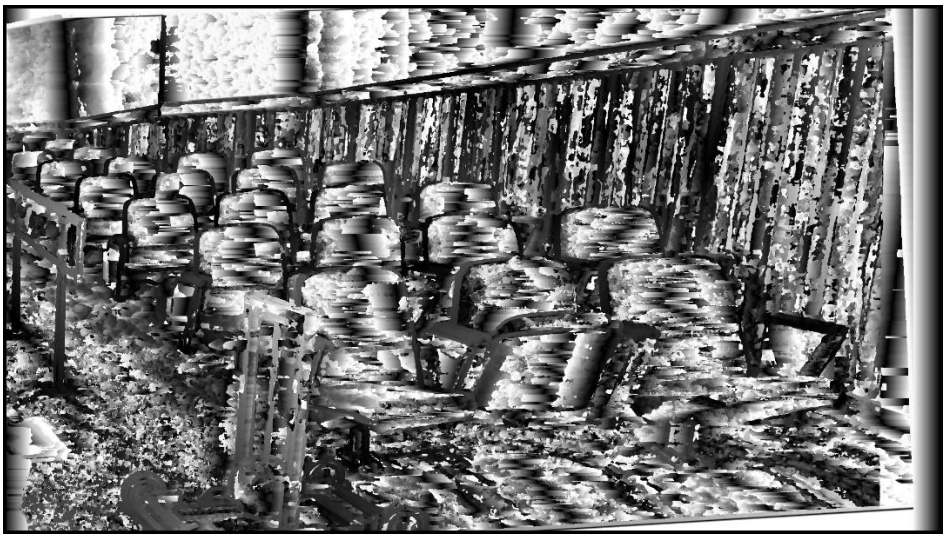


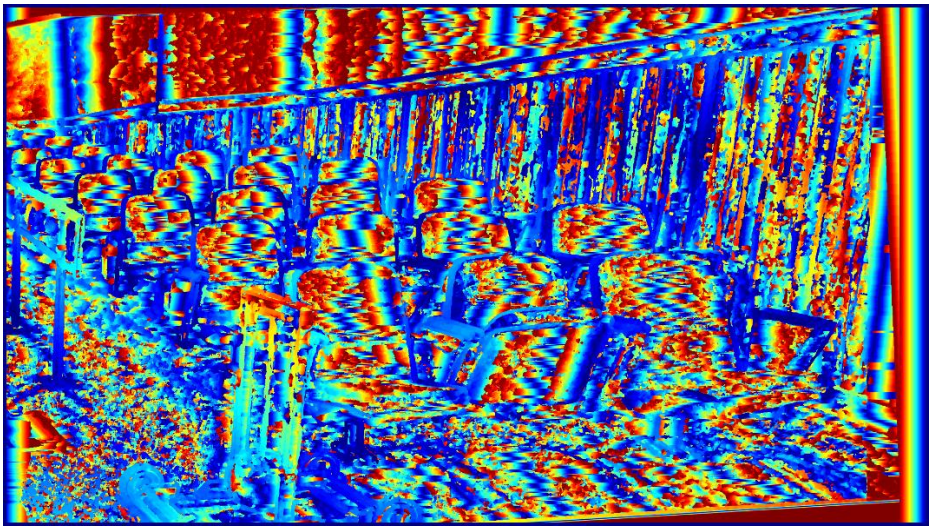


Octagon



Pendulum





Challenges:

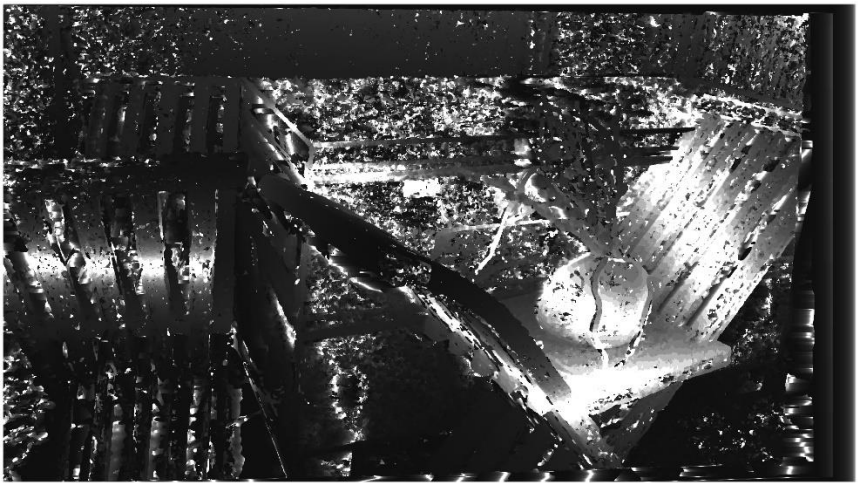
1. The output of disparity map depends on the window size and the stride.
2. The output also depends on the image. Images with repeating background and repeating patterns don’t give a good disparity map.

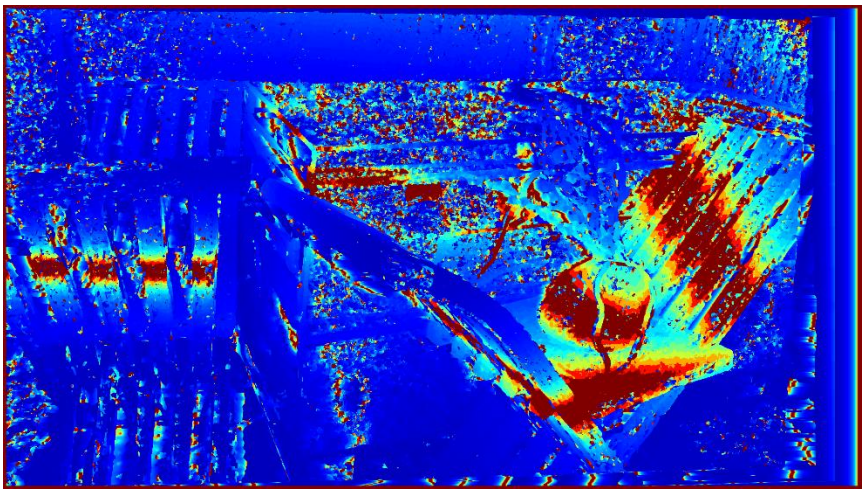
Part 4 (Compute Depth)

1. Depth was computed using focal length, baseline, and disparity map.
2. The resulting depth matrix was clipped to reduce the effect of huge values on scaling.
3. The depth matrix was then scaled so that all the values were between 0 and 255.

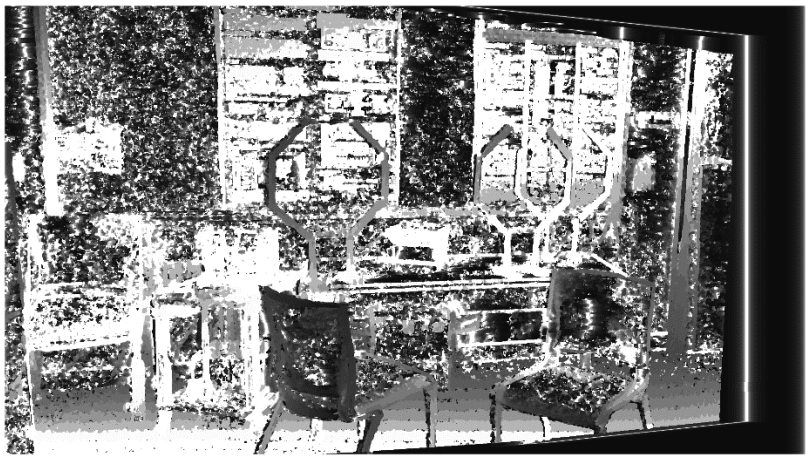
Results:

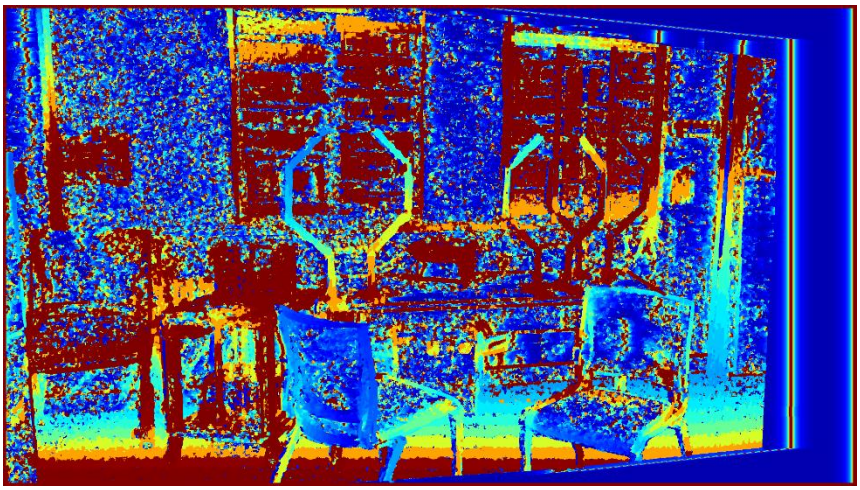
Curule



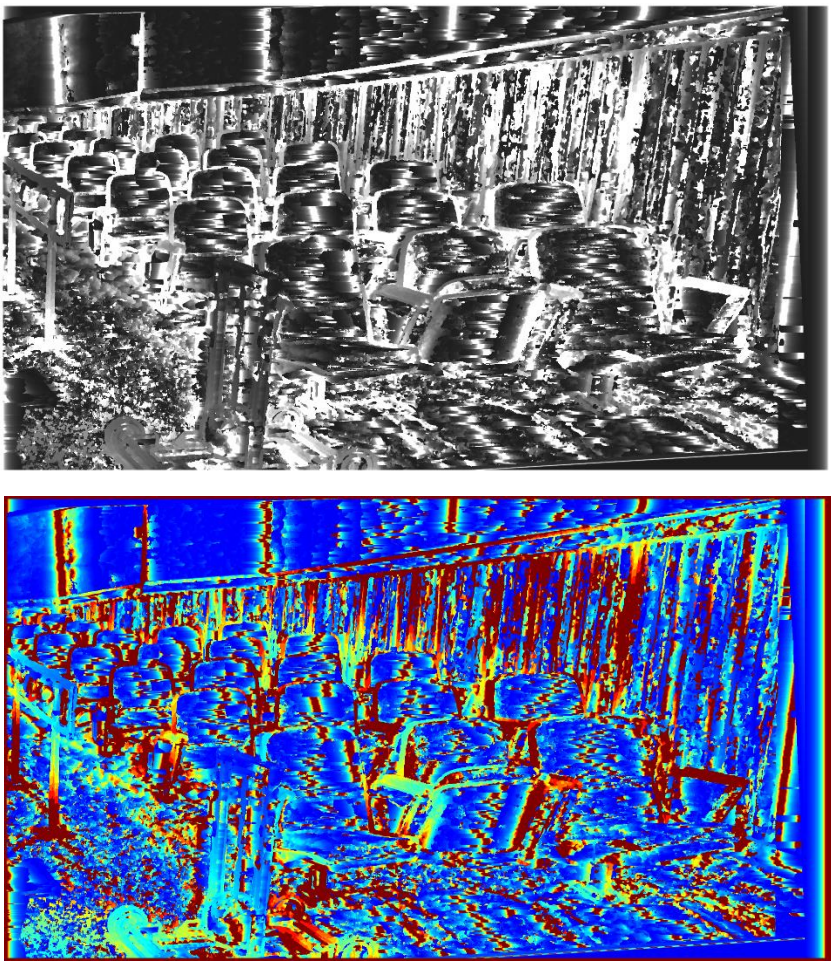


Octagon





Pendulum



Challenges:

1. The output of depth map depends on the window size and the stride.
2. The output also depends on the image. Images with repeating background and repeating patterns don’t give a good depth map

Results Link:

https://drive.google.com/drive/folders/1ucvk7Xy3_Mlf8s792yx0IFTmrNTcbvmx?usp=sharing