Computer Vision (CSL7360)

Programming Assignment-1

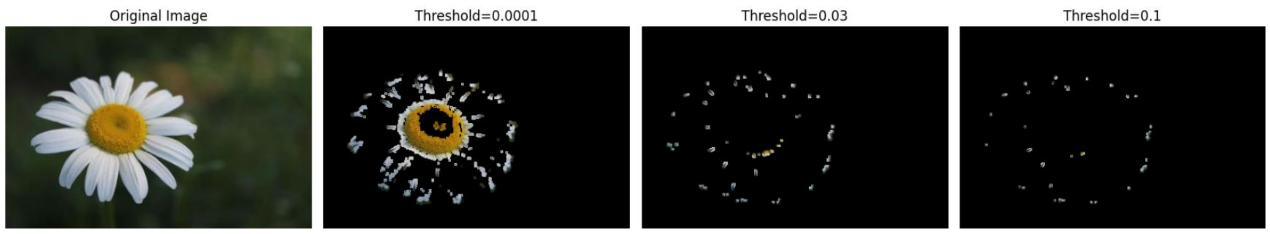
Anish Kumar Maurya(M22EE002)

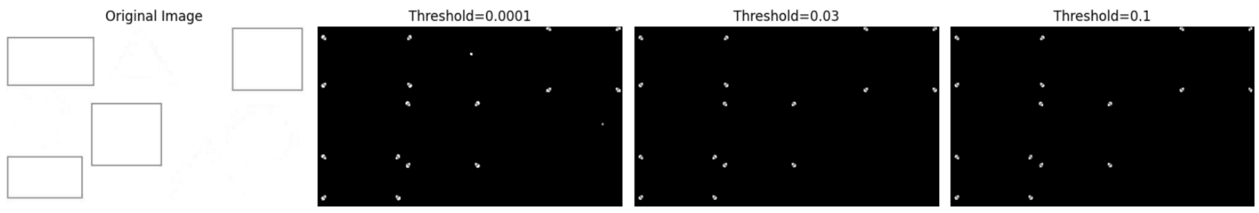
Q-1) Harris Corner detection:

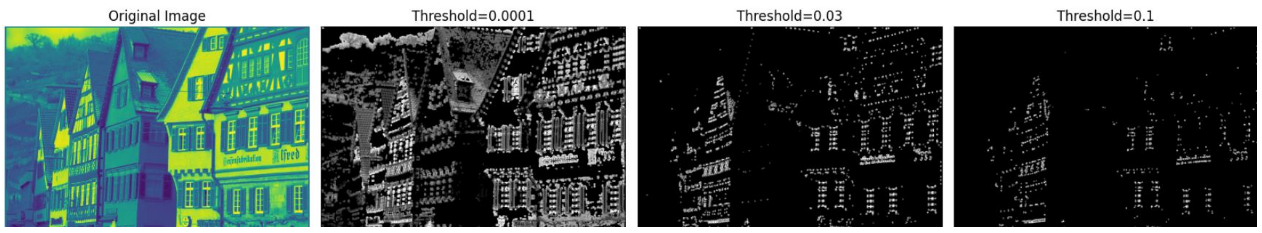
Harris corner detection finds unique elements in a picture that are referred to be corners or interest spots. In this assignment perform the several task to match result with library result.

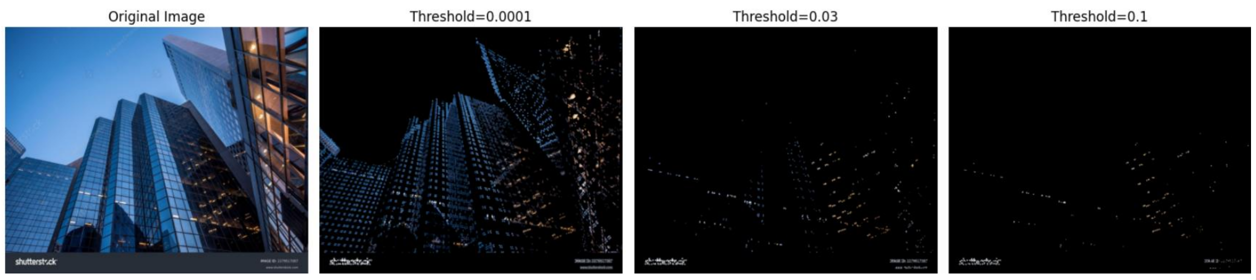
First Iteration:

Window size: 3 & threshold value: [0.0001, 0.03, 0.1]



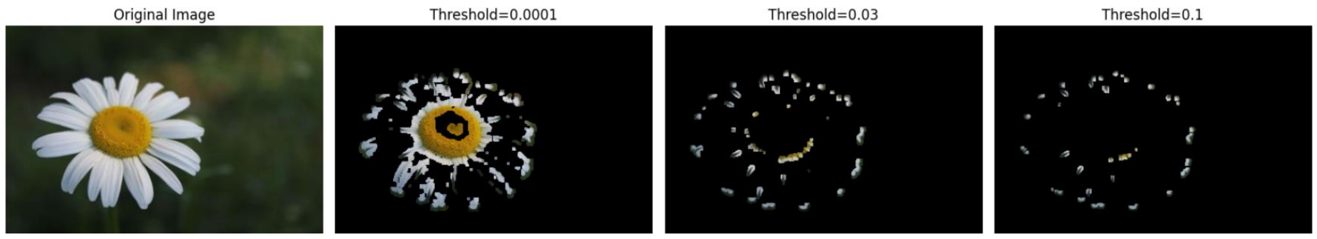


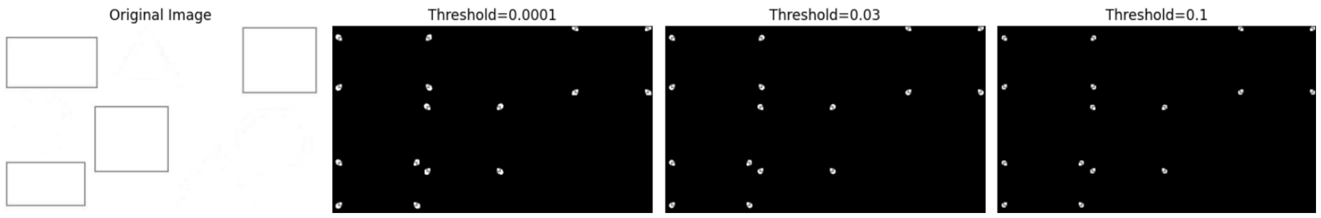


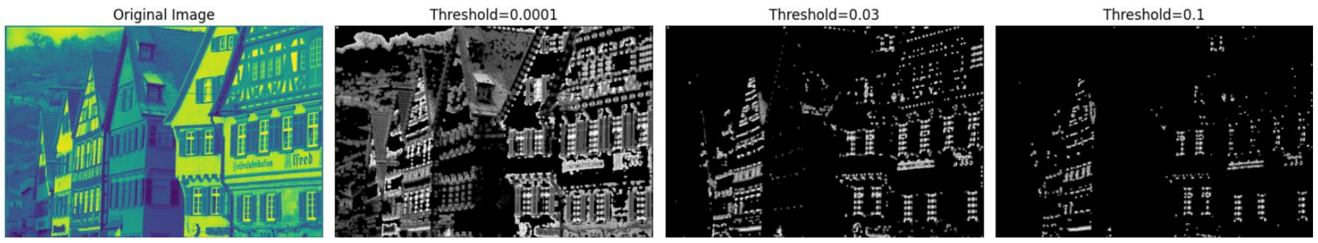


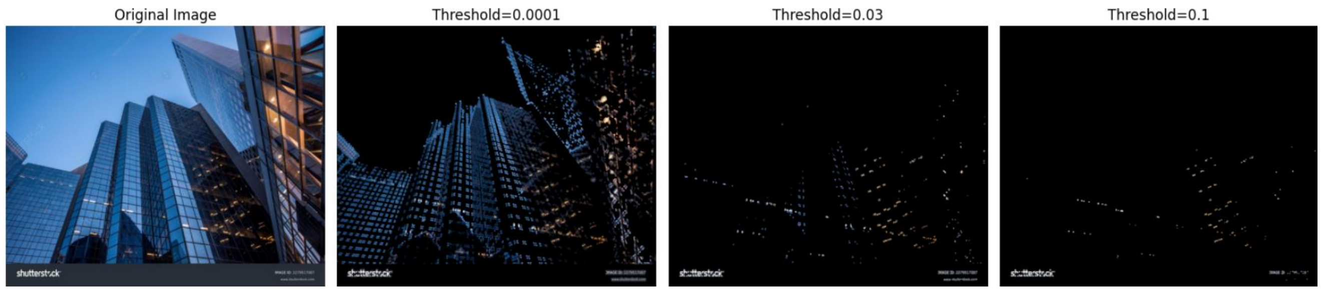
Second Iteration:

Window size: 5 & threshold value: [0.0001, 0.03, 0.1]



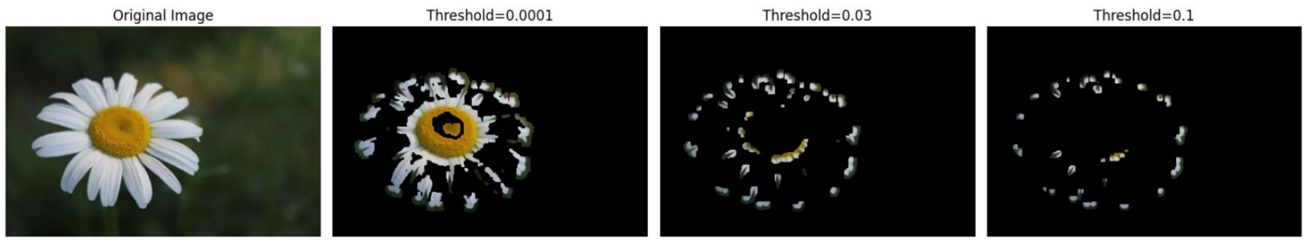


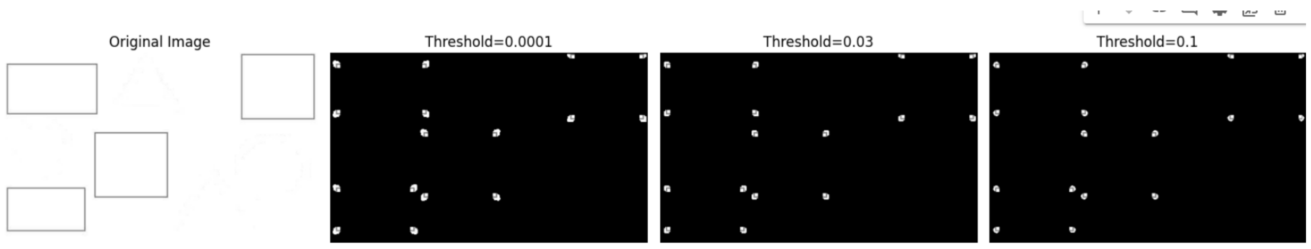


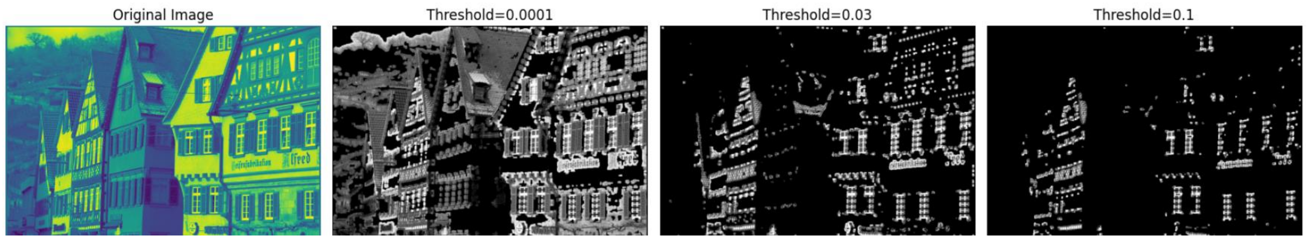


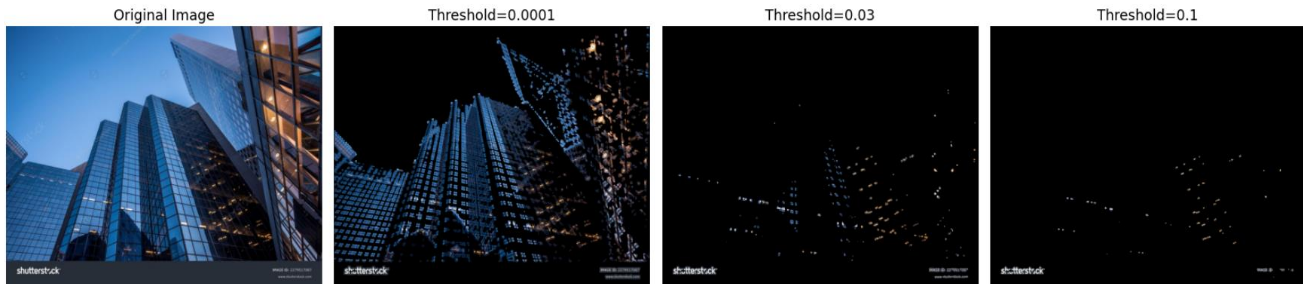
Third Iteration:

Window size: 7 & threshold value: [0.0001, 0.03, 0.1]





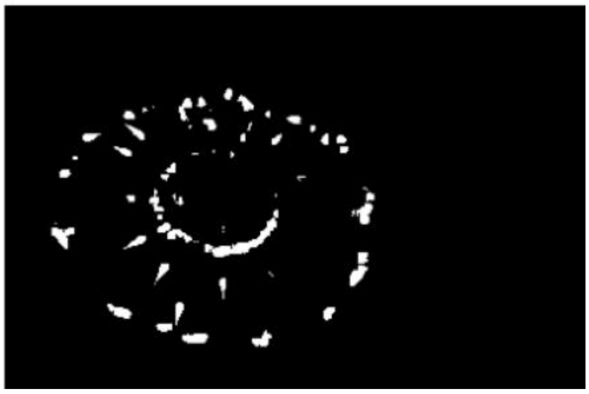




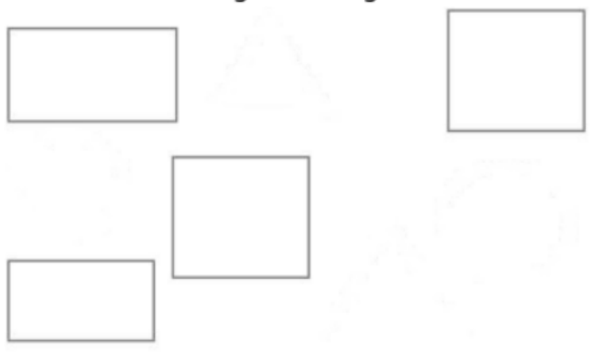
Library Result:

Original Image

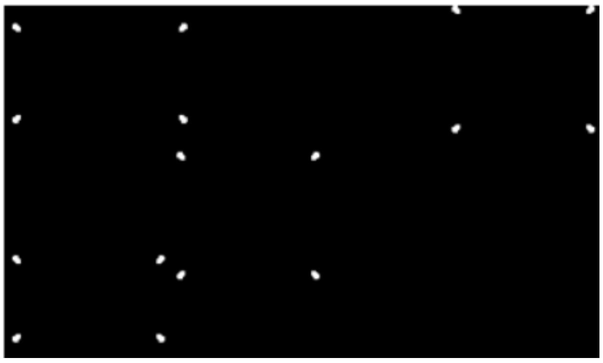
Detected Corners



Original Image

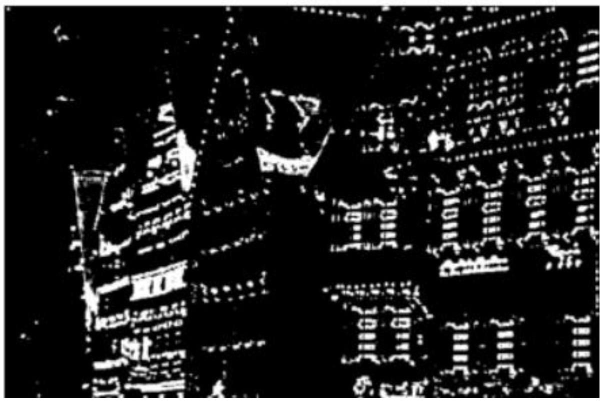


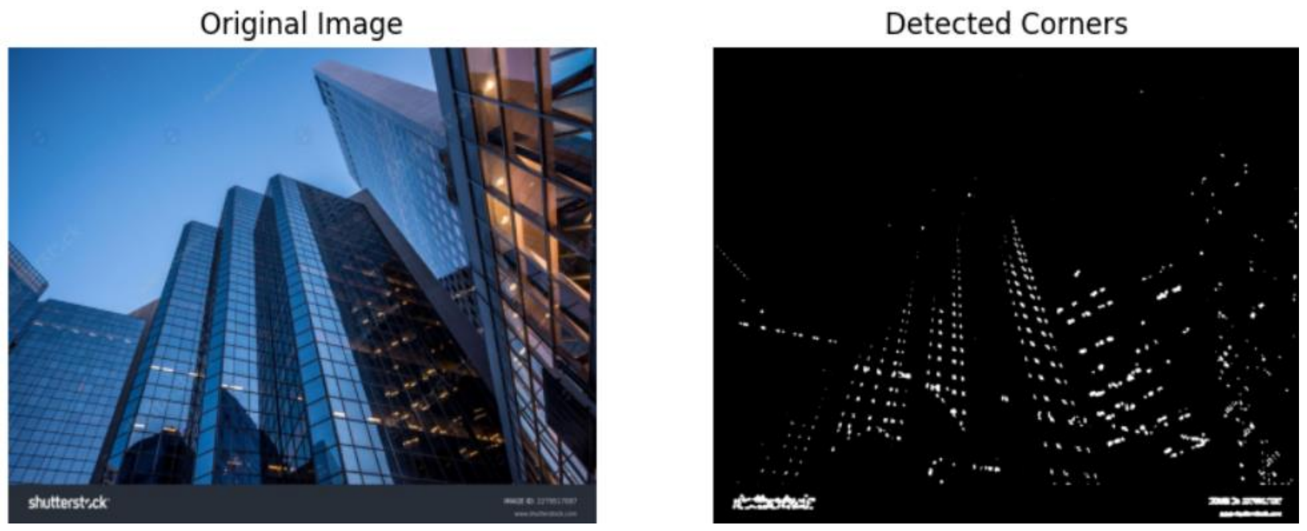
Detected Corners



Original Image

Detected Corners





Conclusion:

Threshold:

If I change the threshold value from lower threshold value detects more corners, including weaker ones and noise, while a higher threshold value only detects stronger corners. On simulation on different threshold value if threshold value is .0001 then it will detect very fine corner when I change the threshold .03 and .1 then it will only bigger corner.

Window size:

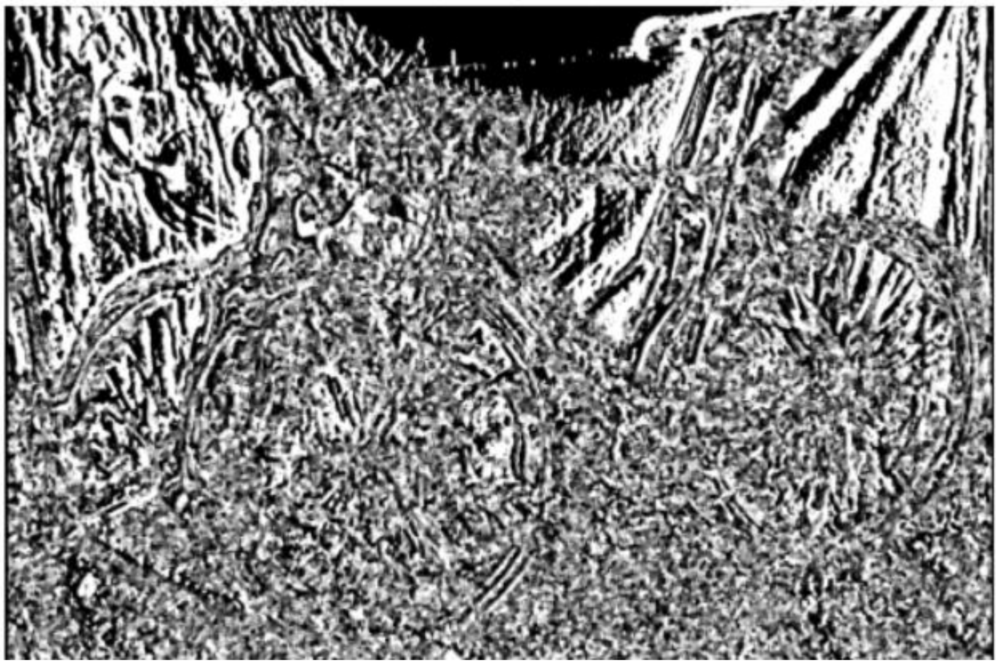
Changing the window size influences the scale at which corners are detected. A larger window size considers a broader local area, potentially capturing larger-scale corners, whereas a smaller window size focuses on finer details and smaller corners.

Q-2) Consider two stereo images I1 (“bikeL.png”) and I2 (“bikeR.png”) of a static scene captured from a stereo camera with the given intrinsic matrices of both the cameras in the file (“bike.txt”). Implement the stereo 3D reconstruction algorithm to find the disparity map, depth map (depth map at each pixel), and 3D point cloud representation of the underlying scene.

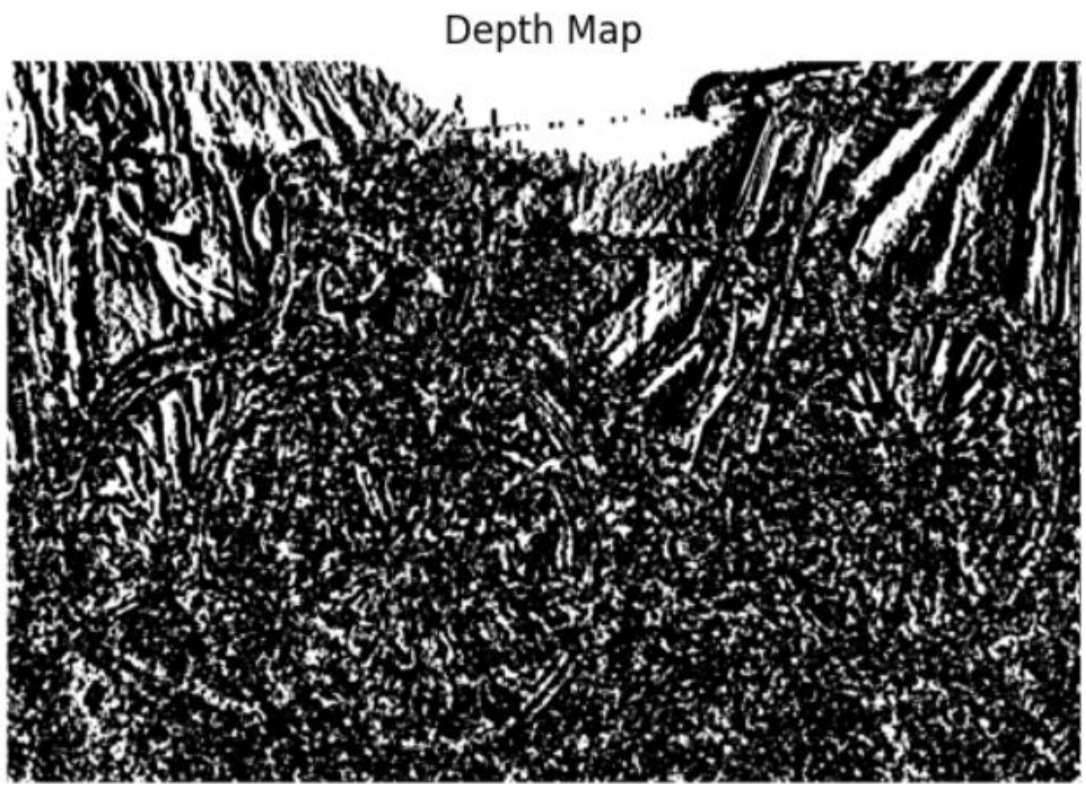
Ans:

- ❖ The disparity map represents the horizontal shift or displacement between corresponding points in a pair of stereo images.
- ❖ Disparity (d) is computed as the difference in horizontal pixel coordinates between corresponding points in the left and right images: *d=Xr−Xl*.
- ❖ The disparity map is a grayscale image where brighter pixels represent larger disparities (closer objects) and darker pixels represent smaller disparities (farther objects).
- ❖ It quantifies the differences in pixel positions between these points, providing valuable information about the depth or distance of objects in a scene

Disparity Map



depth map:

- Often called the depth map or depth image, it provides information on each object's distance from the camera in a scene. Each pixel in the picture has a depth value assigned to it that represents the distance between the camera and the corresponding point in the scene. The depth map is created using the disparity map, which calculates the horizontal shift or displacement between corresponding points in a sequence of stereo images. If we know the baseline distance between the stereo camera pair and the focal length of the camera, we can use the following formula to determine the depth (Z) from the disparity (d) at each pixel:
- Depth (Z) is computed from the disparity (d) using the baseline distance, focal length, and a small constant to prevent division by zero.
- The depth map is a grayscale image where brighter pixels correspond to objects closer to the camera, and darker pixels correspond to objects farther away.
- 

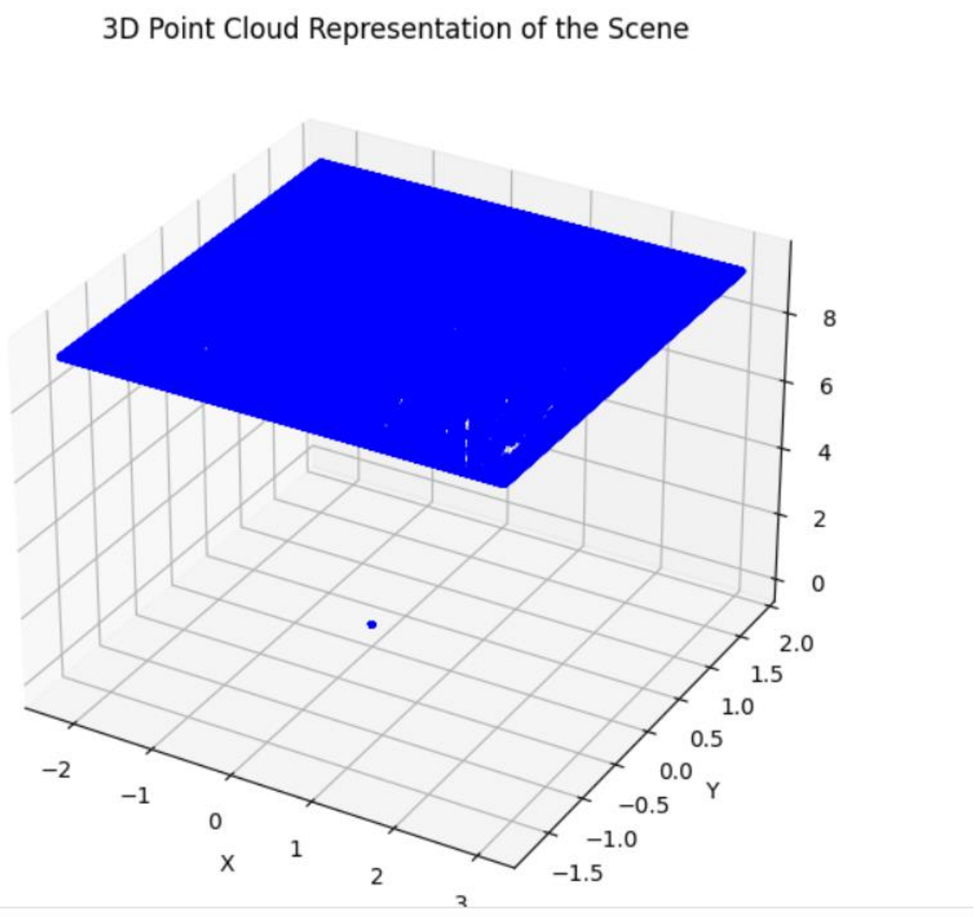
3D point cloud :

A 3D point cloud is useful for many applications, including 3D reconstruction, since it offers a precise and detailed depiction of the surfaces and structures inside a scene.

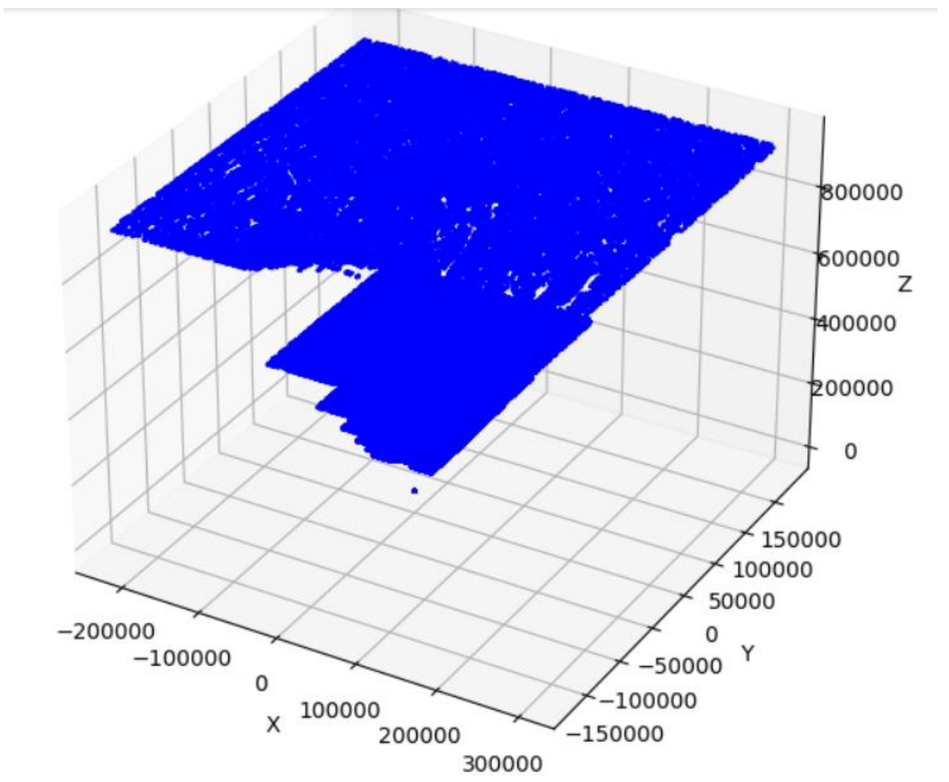
Creating 3D Points: A 3D point in the scene is represented by each pixel in the depth map. We may create a set of 3D points that represent the surfaces and objects in the scene by converting the depth values to 3D coordinates using the intrinsic properties of the camera (such as focus length and main point).

Making a Point Cloud: A point cloud is created by gathering 3D points, each of which represents a surface point or other feature in the scene. The three-dimensional scene's underlying geometry is represented by all of these points combined.

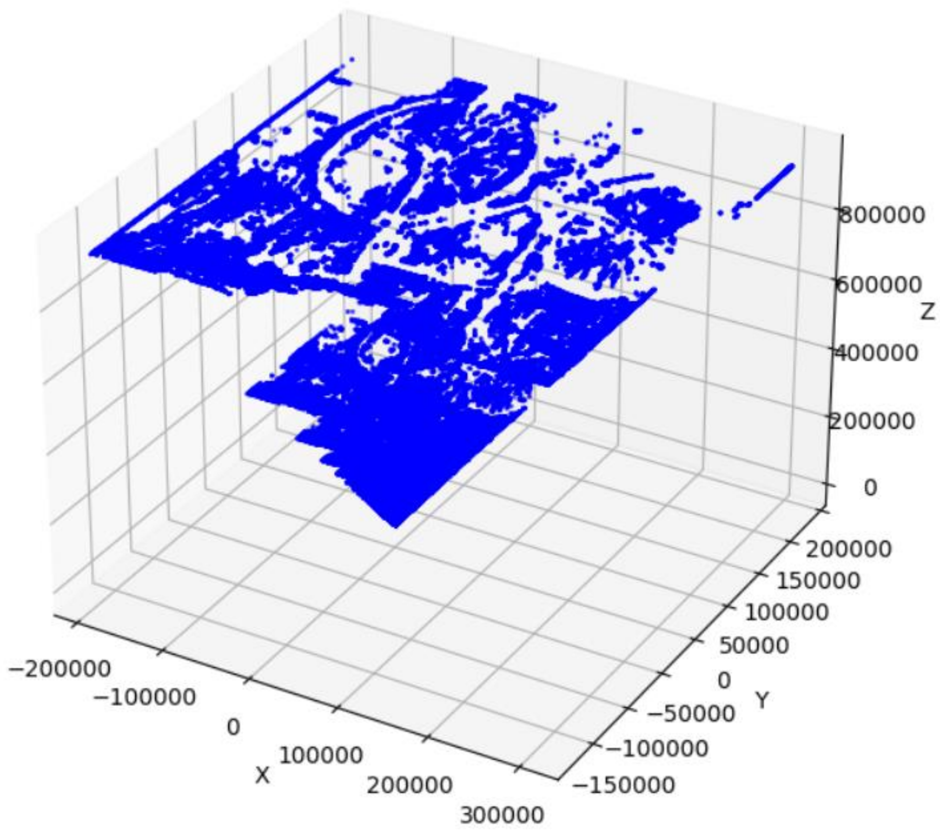
This my 3D point cloud : without fine tune



If I more fine tune for to match result with exact library :



Library result of 3D POINT cloud:



Conclusion:

Error Handling: Before doing any calculations, error handling was included to verify that the disparity map is genuine and that the photos have loaded correctly. This guarantees that potential failures in the picture loading and disparity map computation processes are handled gently by the programme.

Coding was divided into discrete functions (stereo_block_matching, compute_depth_map, compute_3d_point_cloud, plot_3d_point_cloud, load_images, main) in order to modularize the code. This makes the code easier to read, maintain, and reuse.

Q-3) Consider two images I1 (“000000.png”) and I2 (“000023.png”) of a static scene captured from a single camera with the given Fundamental matrix F (“FM.txt”). Write down a code to find the epipolar lines and draw the epipolar lines in both the images. Now, consider uniformly spaced 10 pixels on the epipolar line in the first image and find the corresponding pixels on the second epipolar line. Now, consider uniformly spaced 10 pixels on the epipolar line in the second image and find the corresponding pixels on the epipolar line in the first image. You should search only on the epipolar lines.

Ans:

Epipolar Line: When I have two cameras observing the same scene from different positions, a point in one image corresponds to a line in the other image. This line is called the epipolar line. The fundamental matrix describes the relationship between these points and lines. Each point in one image corresponds to a unique epipolar line in the other image, and vice versa.

Epipolar Triangle: consider three points: two points in one image and their corresponding points in the other image. These points and their corresponding epipolar lines form an epipolar triangle. The sides of this triangle are the epipolar lines, and the vertices are the corresponding points. This triangle is useful for geometric reasoning and for tasks like estimating the depth of a point in the scene.

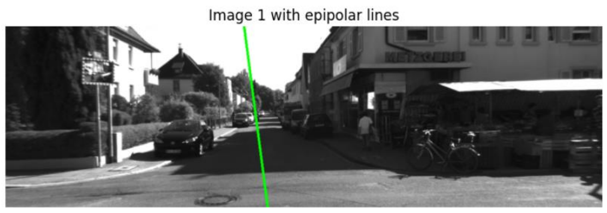
First I have find image size :

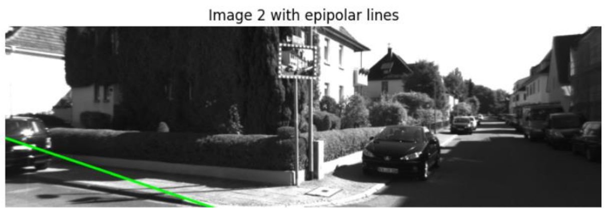
Image 1 size: 1226x370

Image 2 size: 1226x370

Now epipoler line should be within the image size and also point lies within the image shape:

Part A:





Part B:

Sample points on the epipolar lines in the first image and find corresponding points on the second epipolar line

[[(490, 1), (500, 77), (510, 153), (520, 230), (530, 306)]]

Corresponding points on epipolar lines in the second image:

[[(0, 228), (10, 231), (20, 234), (30, 238), (40, 241), (50, 244), (60, 248), (70, 251), (80, 254), (90, 258), (100, 261), (110, 264), (120, 268), (130, 271), (140, 274), (150, 278), (160, 281), (170, 284), (180, 288), (190, 291), (200, 294), (210, 298), (220, 301), (230, 304), (240, 308), (250, 311), (260, 314), (270, 318), (280, 321), (290, 324), (300, 328), (310, 331), (320, 334), (330, 338), (340, 341), (350, 344), (360, 348), (370, 351), (380, 354), (390, 358), (400, 361), (410, 364), (420, 368)]]

All these point I plot on the image :

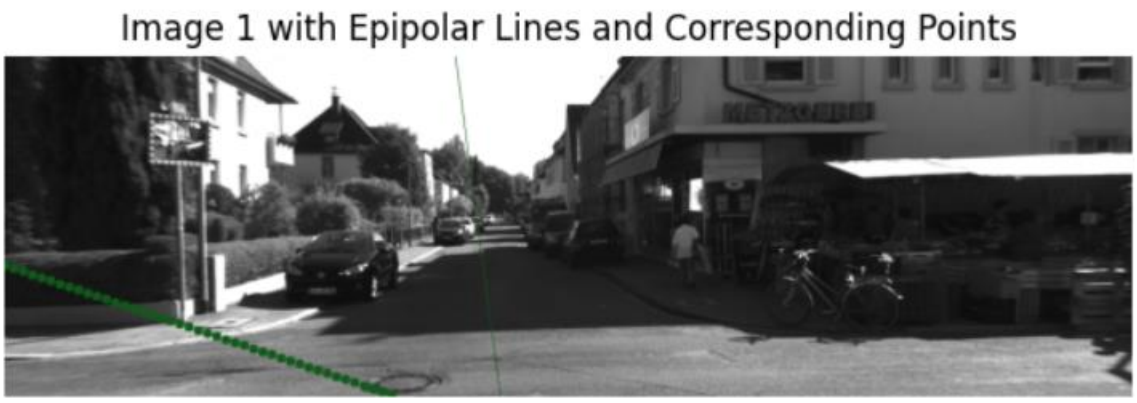
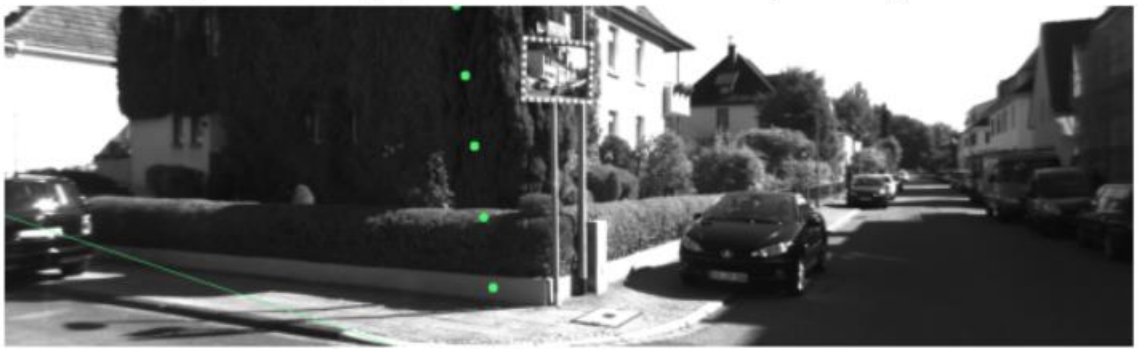
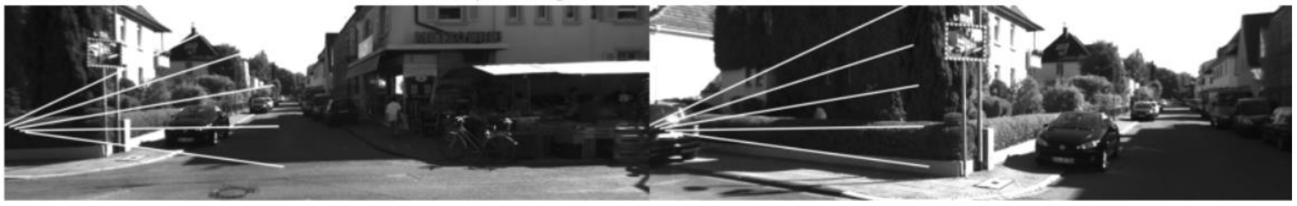


Image 2 with Epipolar Lines and Corresponding Points



Now I am connect this point in image :

Corresponding Points Connected with Lines



Corresponding Points Connected with Lines

Google collab File Link:

Q-1) https://colab.research.google.com/drive/1oiLb0PJVnrsq63DLGA2xjayrw4tDf4VW?usp=sharing

Q-2) https://colab.research.google.com/drive/1HlWC-FFb5944i4f5clu0VzbDuLj2fT7Y?usp=sharing

Q-3) https://colab.research.google.com/drive/11lBt2pd16VY-3D-ZRYqp7M-_o3L4ioQ3?usp=sharing