AERIAL ROBOTICS KHARAGPUR - TASK 1

ABSTRACT :-

Implemented the Sobel edge detection Algorithm for detecting the edges and Hough transform to detect the lines in edges obtained. By analyzing the image obtained from left and Right camera, created the Depth map.
Edge detection had wide applications such as Preventing the robot to fall off, used in space Programs to detect craters and big holes on the Surface. Depth maps helps the robot to prevent collision with the nearby objects and thus safely navigating it.

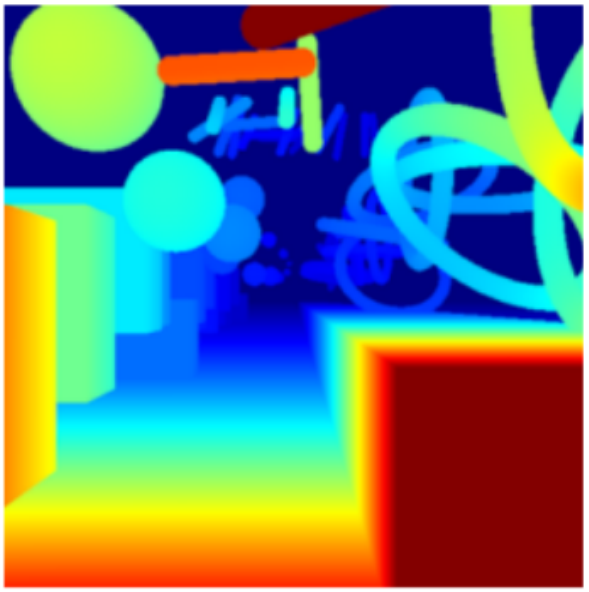
I. INTRODUCTION

Luna is a robot. It is kept on a table with no obstacles in front of it. We have been given a table image to detect the edges and lines. Initially used Laplace method but observed faint edges. Then used Sobel method and Hough Transform method to detect edges & lines respectively. Heatmap is generated by combining the elements obtained from the left and right cameras of Luna. The depth map is used prevent collision of Luna with nearby objects.

II. PROBLEM STATEMENT

To come up with a program that detect the edges and lines in the Table image captured by Luna to prevent it from fall

Also to prevent Luna from Colliding with the nearby objects, implement a program which gives the heatmap as output signifying the distance of Luna from the objects in terms of color



III. RELATED WORK

The Canny edge detection provides much cleaner edges (without noise) as it includes non-maximum suppression & eliminates the false edges. Task b involves handling the stereo images and analyzing them using the function StereoBM under the module cv2.

IV. INITIAL ATTEMPTS

In the beginning I was unfamiliar with Python. So decided to learn it for Finishing the tasks and also looked up at OpenCV Lectures. Initially I started by Understanding the concepts (math) behind it. For eg Sobel, Laplacian Edge detection Method, Hough transform and got to know that there exist function corresponding to those operators like Sobel, Laplacian etc. For creating depth map, couldn't realise The empty spaces, the by changing numDisparities got resolved.

V. FINAL APPROACH

(*Task-A*)

First converted the image to a grayscale image. Since there was time lag while executing the function cv2.imshow(). As a result resized the window and found out the Sobel gradients in x and y direction and combined the results to single Sobel Gradient.

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2} \qquad \Theta = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Magnitude **Direction**

For Hough Transform we transform the image space to hough space. The line in Image space is converted to a point in hough space (rho,theta). When two lines in hough space intersect at a point (a,b). This will represent the line detected in Image space.

(*Task - B*)

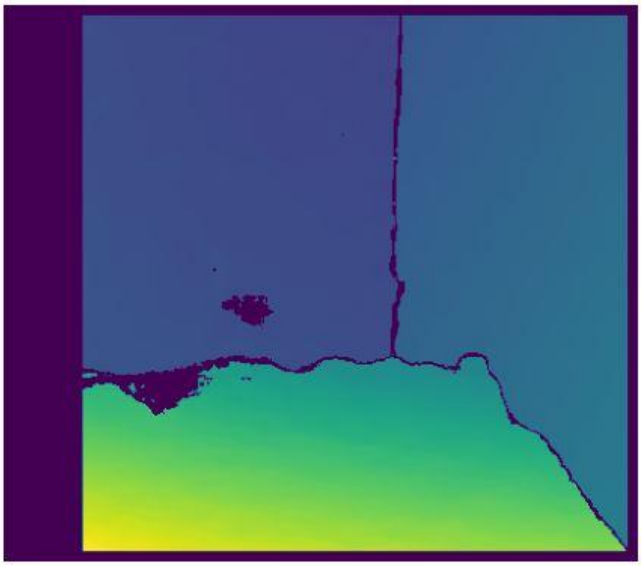
This is based on stereo vision where we analyze two images. Disparity is inversely proportional to depth

$$Z = \frac{f}{d} \times \frac{T}{D}$$

- T = Baseline distance between center of left and right cameras
- f = Focal length of the camera
- d = Physical size of a pixel in camera sensor CMOS/CCD
- Z = Distance between point P and camera center

Disparity refers to the The disparity is the apparent motion of objects between a pair of stereo images.

VI. RESULTS & OBSERVATION



VII. FUTURE WORK

Expecting this algorithm could

function for all images. Instead of using Sobel we can use Canny as it provides accurate edge detection by suppressing noise and false edges.

VIII. CONCLUSION

Enjoyed the tasks a lot and got Hands on experience with project

involving concepts related to Robotics. Initially tasks were felt

As Mammoth but once got into the process

Of learning could feel the joy.

REFERENCES

- Stereo Vision: Depth Estimation between object and camera | by Apar Garg | Analytics Vidhya | Medium
- Edge Detection Using OpenCV | LearnOpenCV #
- OpenCV Python Tutorials - YouTube