Save Luna !

M B Thejesshwar

***Abstract*— Luna is a robot with wheels and two cameras, but she does not know how to use them to navigate. This project focuses on developing a depth estimation method using stereo vision so that Luna can detect obstacles. The approach involves calculating disparity between the left and right images and generating a heatmap where nearby objects appear red and distant ones appear blue. The implementation is done from scratch using OpenCV.**

I. INTRODUCTION

Stereo vision is widely used in robotics to estimate depth, which helps in navigation and obstacle avoidance. Luna has two cameras placed parallel to each other, capturing left and right images. The challenge is to process these images to generate depth information, which can then be used for movement decisions. The core idea is to compute the displacement between corresponding points in both images and use this to estimate depth.

II. PROBLEM STATEMENT

Luna needs to move in an environment while avoiding obstacles. She takes images using two cameras, and our task is to generate a depth map from these images. The depth map should visually represent objects, with red showing closer obstacles and blue showing farther objects. The implementation must be done from scratch while using basic OpenCV functions.

III. RELATED WORK

There are many methods for depth estimation, including Semi-Global Matching (SGM) and Block Matching. However, these are more advanced and not required for this project. Instead, a simpler Sum of Absolute Differences (SAD)-based block matching technique is used to compute disparity.

IV. INITIAL ATTEMPTS

Initially, OpenCV's built-in stereo matching functions were considered, but since the task was to implement disparity calculation from scratch, we opted for a simpler approach. The first attempt used a fixed block size for matching, but it resulted in noisy outputs. To improve results, normalization was applied to disparity values.

V. FINAL APPROACH

1) lgorithm for Depth Estimation: The final implementation follows these steps:

1) **Convert Images to Grayscale**: Since color is not needed for depth estimation, the images are converted to grayscale.
2) **Sliding Window for Block Matching**:
   - A fixed-size window is placed around each pixel in the left image.
   - The same-sized window is searched along the horizontal axis in the right image.
   - The Sum of Absolute Differences (SAD) is computed for each shift.
   - The shift (disparity) with the lowest SAD is selected.
3) **Compute Disparity Map**: For each pixel, store the best-matching disparity value.
4) **Normalize Disparity Values**:
   - The computed disparity values are normalized to fit within 0-255.
   - This ensures that the depth map is visually interpretable.
5) **Apply Heatmap for Visualization**:
   - OpenCV's `applyColorMap` function is used to color-code the disparity map.
   - Red represents closer objects, and blue represents farther objects.

VI. RESULTS AND OBSERVATION

The generated depth heatmap successfully highlights objects based on their distance. The method works well in structured environments but struggles in textureless regions where matching is difficult. Compared to OpenCV's StereoBM algorithm, this method is simpler but produces noisier outputs.

VII. FUTURE WORK

This approach can be improved by using more advanced matching techniques like Semi-Global Matching or deep learning-based methods. Another limitation is that the current implementation does not handle occlusions well, which could be improved in future versions

CONCLUSION

This project implements a basic depth estimation algorithm to help Luna navigate without colliding with obstacles. The approach involves calculating disparity using a simple block-matching method and visualizing depth using a heatmap. While it works effectively, there is scope for further improvements in accuracy and robustness.

REFERENCES

[1] OpenCV Documentation - https://docs.opencv.org/