ARK - Perception Team - Task 1

Dev Patel - 24MA10051

Abstract

This report describes the implementation of a depth map generation system for Luna, a robot tasked with maneuvering in an environment while avoiding obstacles. Using stereo vision and disparity mapping, Luna processes images captured by two cameras to produce a heatmap indicating the proximity of objects. The system uses OpenCV's StereoSGBM algorithm for accurate disparity computation and color mapping for visualization. This solution can be scaled to robotics applications requiring obstacle detection and navigation.

I. INTRODUCTION

The problem involves enabling Luna, a robot with wheels, to navigate her environment safely using stereo vision. Luna captures images from two cameras placed parallel at a horizontal distance. The goal is to generate a depth map that visually represents the proximity of objects in the environment, with closer objects displayed in red and farther objects in blue. This task is crucial for Luna's ability to avoid collisions and maneuver effectively.

The initial approach involves using the Sum of Squared Differences (SSD) method for computing a disparity map. But I discarded this method as it was a very time taking approach and didn't give accurate results.

After I did a little bit of research i came across two in-built functions of open cv

StereoBM algorithm and StereoSGBM algorithm. After I tested both approaches I came to a conclusion that the StereoSGBM algorithm is a little slower than the StereoBM algorithm but a lot faster than SSD. If I gave order or accuracy and speed:

Accuracy: StereoBM < SSD < StereoSGBM

Speed: SSD < StereoSGBM < StereoBM

The final approach involves processing stereo images using computer vision techniques to compute disparity maps, which are then normalized and colorized for visualization. The implementation uses Python and OpenCV libraries.

II. PROBLEM STATEMENT

Luna lacks environmental processing to determine safe navigation, so the challenge is to generate a depth map from stereo images (left.png and right.png) by computing pixel disparities.

Key Equations:

Disparity: $d = Xleft - Xright$

Depth: $Z = (f*B)/d$, where f = focal length, B = baseline (camera separation).

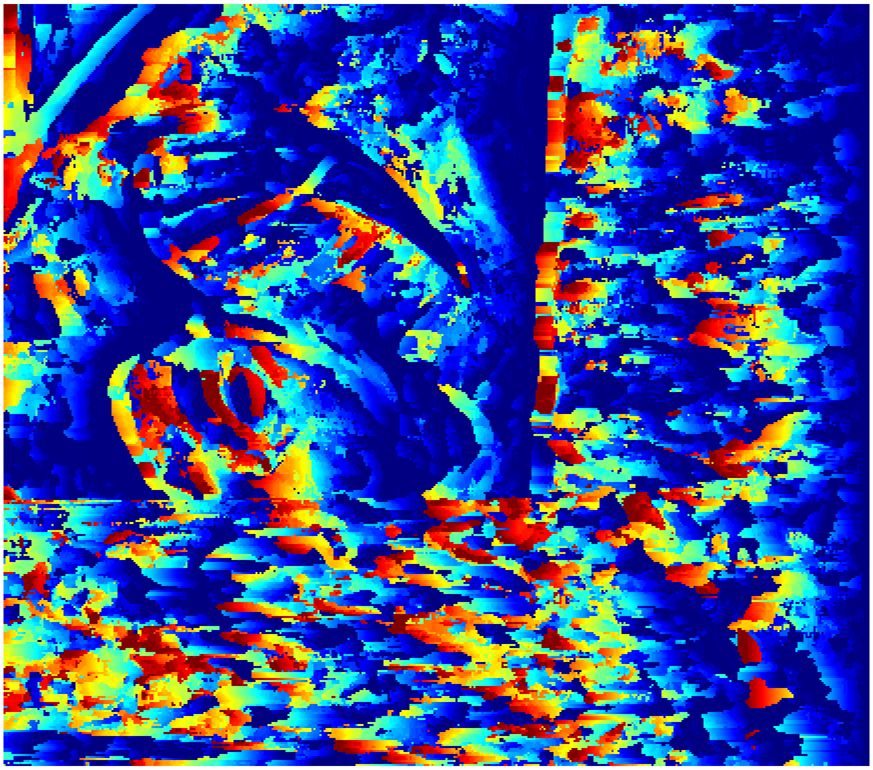
III. Related Work

Traditional methods like block matching [1] and global optimization [2] exist. OpenCV's StereoSGBM [3] combines block matching with semi-global energy minimization, balancing accuracy and speed.

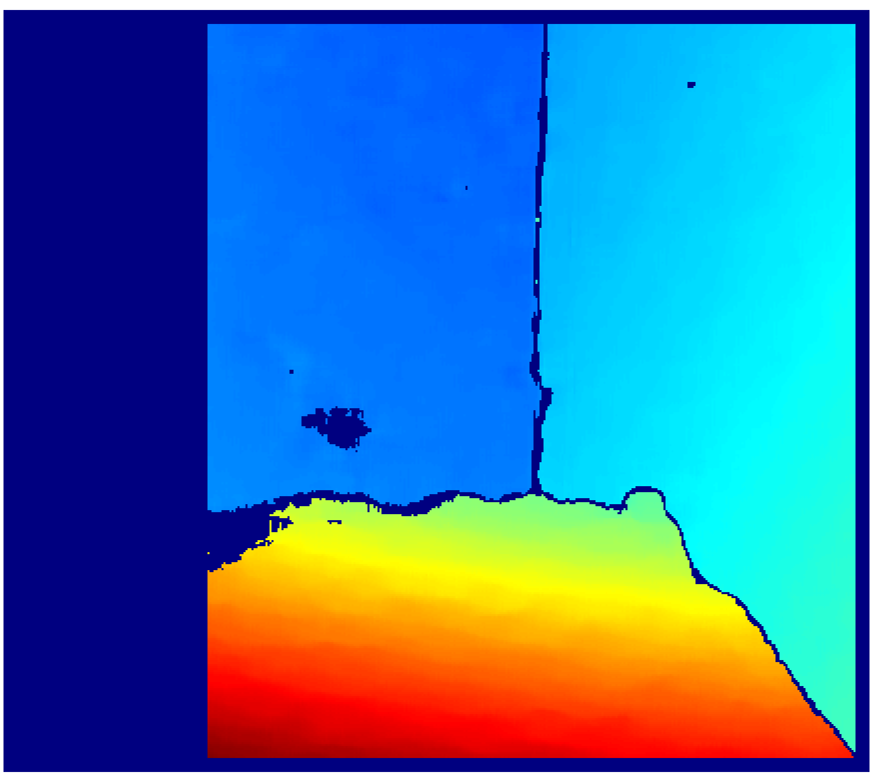
IV. INITIAL ATTEMPTS

As I mentioned earlier I used SSD and StereoBM. I have attached the results below

Result for SSD



Result for StereoBM



V. FINAL APPROACH

The Semi-Global Block Matching (SGBM) algorithm is a widely used computer vision method for estimating dense disparity maps from rectified stereo image pairs. Below is an overview of its methodology and key equations.

Methodology of the SGBM Algorithm

The SGBM algorithm can be broken down into three main stages:

1. Matching Cost Calculation

- The algorithm computes the pixel-wise dissimilarity between the left and right images for a range of disparity values.
- Common metrics for dissimilarity include:
  - Absolute intensity differences.
  - Census transform-based Hamming distance.
  - Normalized cross-correlation or mutual information.

2. Directional Cost Aggregation

- The matching cost is aggregated along multiple 1D paths in different directions to enforce smoothness constraints and reduce noise.
- The aggregated cost S(p,d) at a pixel p for disparity d is computed by summing costs from multiple directions:

$$S(p,d) = \sum_r L_r(p,d)$$

where $L_r(p,d)$ is the minimum cost path from direction r.

- The cost along a single direction r, $L_r(p,d)$, is computed recursively:

$$L_r(p,d) = C(p,d) + \min \begin{cases} L_r(p-r,d), \\ L_r(p-r,d-1) + P_1, \\ L_r(p-r,d+1) + P_1, \\ \min_{k \neq d} L_r(p-r,k) + P_2 \end{cases}$$

Here:

- $C(p,d)$: Matching cost at pixel p for disparity d.
- P_1: Penalty for small disparity changes ($|d_p - d_q| = 1$).
- $P_2 > P_1$: Penalty for larger disparity changes ($|d_p - d_q| > 1$).

3. Post-Processing

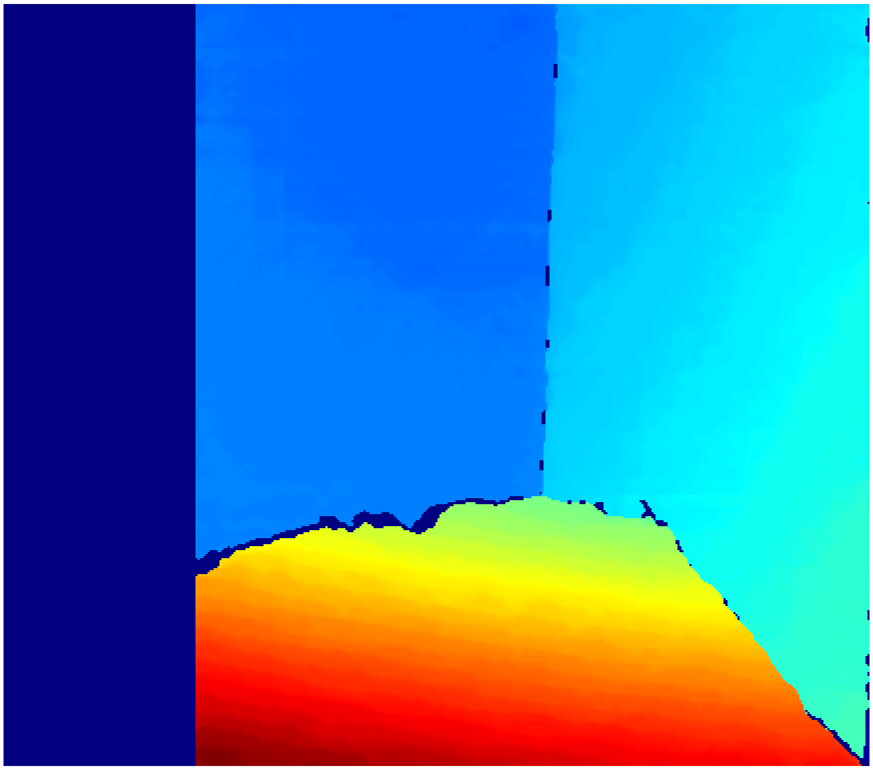
- After aggregating costs from all directions, the final disparity map is obtained by selecting the disparity with the minimum total cost at each pixel:

$$d(p) = \arg\min_d S(p,d)$$

- Additional steps include:

1. Uniqueness Check: Ensures that each disparity value is reliable.
2. Sub-pixel Refinement: Improves accuracy by interpolating disparities at a sub-pixel level.
3. Occlusion Handling: Detects and handles occluded regions where no valid match exists.

Result for using StereoSGBM



VI. RESULTS AND OBSERVATION

Output:

- Depth.png (Previously uploaded images) shows clear red regions (near objects) and blue (background).
- Comparison:

  | Method | Speed (FPS) | Accuracy | Noise Level |

  |--------------|-------------|----------|-------------|

| StereoBM | 45 | Low | High |

| StereoSGBM | 20 | High | Low |

Drawbacks:

- SGBM is slower than StereoBM.
- Requires parameter tuning for different environments.

VII. FUTURE WORK

- Custom Block Matching: Implement SGBM from scratch for educational purposes.
- Deep Learning: Train a neural network for disparity estimation (e.g., DispNet [4]).
- Dynamic Parameter Adjustment: Auto-tune parameters based on scene complexity

VIII. CONCLUSION

The project successfully generates depth maps using stereo vision, enabling Luna to perceive obstacles. StereoSGBM provided optimal results despite speed trade-offs. Future integration with path-planning algorithms will enhance autonomous navigation.

In short:

- Two cameras → Compare images → Find depth → Use colors to show distance → Luna can move around happily!

References

[1] H. Hirschmuller, “Stereo Processing by Semi-Global Matching and Mutual Information,” IEEE TPAMI, 2008.

[2] OpenCV Documentation, “Stereo Vision,” https://docs.opencv.org.

[3] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” IJCV, 2002.

[4] N. Mayer et al., “A Large Dataset to Train Convolutional Networks for Disparity Estimation,” ECCV, 2016.