Aerial Robotics Kharagpur

Taranisen Naik ,24ME10035

SAVE LUNA !

In this project, we have to compute the depth map from a stereo image pair captured by robot Luna's left and right cameras. The process involves converting images to grayscale, then extracting a patch from the left image and finding the corresponding patch in the right image using the Sum of Squared Difference. A disparity map is generated and normalized to create a final heatmap representation of depth.

Applications:

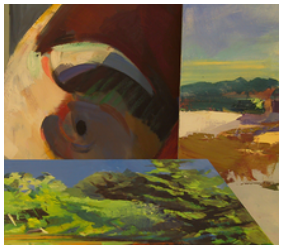
This is a very useful technology in self driving cars, humanoid robots, and 3d reconstruction from 2d images.

INTRODUCTION

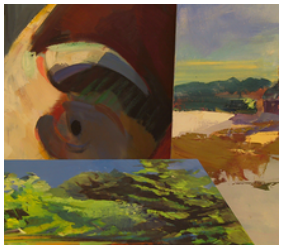
My first solution was faulty because of individual pixel checks and loop mistakes. I refined it with the application of block-based comparison, edge cases, and optimization with NumPy. The resulting disparity heatmap shows near objects in red and far-off ones in blue, allowing ARK to navigate around obstacles effectively, enhancing autonomous navigation.

PROBLEM STATEMENT

Luna is an autonomous robot that has two cameras mounted side by side at a constant horizontal distance. Even with the capability to record its surroundings, Luna does not have the ability to process visual data and move around safely. We have to create an algorithm that will allow Luna to sense depth and avoid collision.

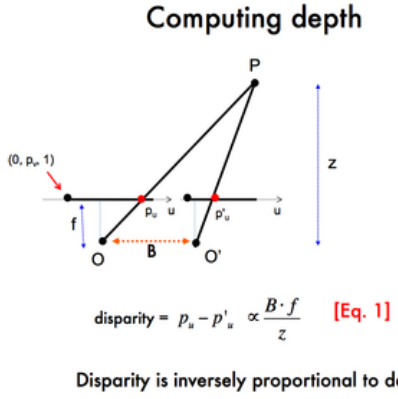


left.png



right.png

Stereo Vision and Depth Perception :
The issue is based on stereo vision, which simulates human 2 eyes vision. The two cameras take slightly different views of the same scene. By calculating the disparity between corresponding points in the left and right images, we can estimate depth.



INITIAL ATTEMPTS

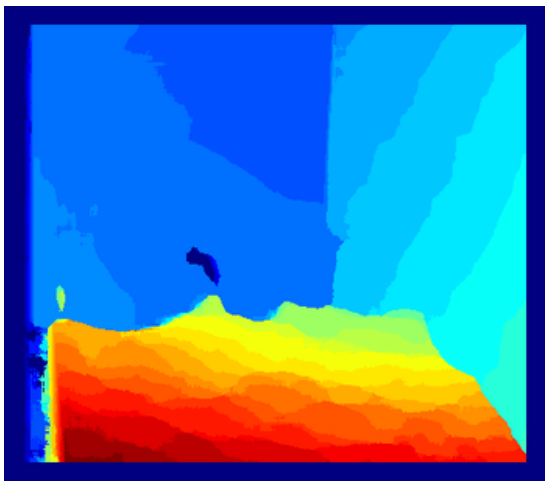
Initially, I started iterating from the top row of the left image, but that was wrong because pixels of the same intensity could be found in different locations. In my previous attempt. Then I used a block-based comparison for each pixel, considering not only the pixel but also the surrounding pixels. This improved the algorithm to a greater extent. But in my second run, I initialized the loop from the initial pixel of the image, and that is why the mistake was made.

FINAL APPROACH

I minimized my for loop by efficiently processing edge cases and selecting an appropriate block size. I also made use of NumPy operations to calculate more efficiently. For each block, I calculated the total disparities and placed the corresponding pixel with the smallest disparity value. Next, I normalized the disparity map by rescaling it in the range of 0 to 255. Lastly, I converted this grayscale image to a heatmap to arrive at the final disparity heatmap.

RESULTS AND OBSERVATION

In the final result, objects with higher disparity, which are at a closer distance colored in red, while objects at a farther distance appear in blue. This allows the robot to move safely, as it will prioritize avoiding collisions with nearer objects (red regions) first, resulting in obstacle-free movement.



Final heat map

CONCLUSION

My first solution was faulty because of individual pixel checks and loop mistakes. I refined it with the application of block-based comparison, edge cases, and optimization with NumPy. The resulting disparity heatmap shows near objects in red and far-off ones in blue, allowing Robot luna to navigate around obstacles effectively, enhancing autonomous navigation.

RESOURCES USED

Apar Garg, "Stereo Vision: Depth Estimation between object and camera"
Analytics Vidhya, Nov 17, 2021

https://youtu.be/hUVyDabn1Mg?si=ab1MtH5c5sijDL2w