STEREO IMAGE BASED COLLISION AVOIDANCE

Amrit Lal Singh

***Abstract*— Here is a successful attempt to stereo image based depth detection.**

This was possible by generating a disparity map and then generating a depth map. Then finding the location of object, then finding the distance of it from the reference frame fixed to the camera. This method has immense use cases from upgrading the data we feed into machine learning models to enabling bots to know the distance of an object without usage of other complex technology some of them detects the time it took light to come back or other point cloud generating methods instead using the commonly available camera technology after some calibration to generate depth maps and thus offers immense possibilities for robotics, aerial and other. It will prove useful to detect distance from objects, ground landing points, walls, trees and others. It is in the current day proving useful by applying this to the modern day phone camera to generate the depth map and blur the background and focus on the subject. this depth perception is very important to our task of perceiving the surrounding.Accurate depth perception in co-located teleoperation has the potential to improve task performance in manipulation and grasping tasks. As now we are aware of the position of the object along with the distance of our grippers. Other distance detection/mapping methods like LIDAR commonly involve a rotating sensor/projector this may not be suitable for aerial vehicles as the rotational mechanism will add some weight and its starting and stopping may lead to unwanted mechanical effects on the flight of the drone. Also the effect of high frequency vibration like as on drones have to be taken care of which feels could be better dealt on image based algorithm as we could go through image stabilization before this computation.

I. INTRODUCTION

Describe the problem statement and your approach using which you have approached the problem. Describe all the things and methods you have tried which might have failed in brief.

We here have to estimate the distance of the camera from the given obstacle which is a bike given to us in the form of an image. This allows us to use template matching to find the location of the image on the 2D image but that is not our goal, the main goal is to find the location of the object and in this case the distance data is important.

Stereo vision is a technique used to estimate the depth of a point object from the camera using two cameras. The foundation of stereo vision is similar to 3D perception in human vision and is based on the triangulation of rays from multiple viewpoints.

II. PROBLEM STATEMENT

This should cover one full column of page.

*The internet and to it's seeming infinite store of knowledge. The gods of stack overflow

In order to find distances to obstacles we need depth information, however sometimes we don't have depth cameras so we end up using two cameras in specific orientation relative to each other to generate slightly different images called stereo images. In this task you are provided with a pair of stereo images. You need to generate a depth map, find the obstacle and calculate the distance. The obstacle is a blue bike which you have to avoid crashing into.

Fig. 1. ArUco detection

Fig. 2. ArUco detection

III. RELATED WORK

Work has been done on generating disparity map and is available. Also work on generating depth from camera data and stereo image has been done, but existing work based on the data given to us is not present. Also most of the work done includes real life images and work done on simulated images are rare, but this may work to our advantage as the simulated version may have less noise. And hence may be more supportive to test the proper functioning of the algorithm.

IV. INITIAL ATTEMPTS

We can not attempt the problem with the goal of finding the distance to the object without the camera matrix. My initial attempt was to find the disparity map of the images with respect to the left image, but this at best gives us a number proportional to the relative depth of a point with respect to other point and does not gives its absolute distance that we need. So we somehow have to reference the unlabelled camera data.

Write all the equations and results with pictures if applicable.

V. Final Approach

This should cover both columns (full page) excluding images

Write about the final algorithm used or developed which was able to finally solve the problem statement. Write all the assumptions and equations properly with images and result tables / comparison.

If report is about implementation, write all the steps of implementing the report with step-wise syntax and details about the errors faced and their solution.

At first we import numpy, opencv-contrib-python and matplotlib.

Then we store the given to us in the code in variables. Then we use the decompose projection matrix function to decompose the left and right projection matrices into left and right camera matrix, rotation matrix and the translation vector. THen we conver both the right and left image in to greyscale inorder to reduce the computational load and define the matcher type and run it with the recommended parameters by the opencv doccumentation. the matcher takes a small pixel window in the reference image and matches it to the its counterpart in the other image. This is one of the most important parts in the process as it locates where the point visible in the reference image is to be found in the second image. we set the minimum disparity to 0 as it would be the best case scenario, max disparity, blocksize and importantly we sect the matching method the cv2.STEREOSGBM MODE HH4 being one of the most resource intensive but as we have images and not video feed we would go with this. Then using the matcher we compute the disparity map And we store the camera intrinsics into variables. As the focal length of both the cameras are same we take the focal length from one of the camera matrix we calculated. and the variable represents the distance between the both cameras which we obtained by subtracting the translation vector's axis element.

Then each element of depth map was generated by dividing the f*b by the disparity map. We now have the depth of all the points of the frame. Now we run Template matching over the frame with the bike as the template, and then seclude the hotspot by using cv2.minmaxLoc() then we find the distance of the hotspot plus image hight/2 and width/2 as the hotspot would be the point where the top left corner of the rectangle lies and then we print the distance.

It is not possible to estimate the distance (depth) of a point object 'P' from the camera using a single camera 'O'. This is because however close or far 'P' is on the projective line, it will map to the same point 'p' in the image.

Stereo vision is a technique used to estimate the depth of a point object 'P' from the camera using two cameras. The foundation of stereo vision is similar to 3D perception in human vision and is based on the triangulation of rays from multiple viewpoints.

Depth is inversely proportional to disparity. The more the disparity is, the closer the object is to the baseline of the camera. The less the disparity is, the farther the object is to the baseline.

Disparity is proportional to baseline. This is easy to visualize. If we have a small baseline distance between the two cameras, then the difference/disparity between the two images is going to be small. As we increase the baseline, the disparity is going to scale up. These two are very important considerations. When we are designing a stereo system, we want to measure disparity very precisely because that's what gives up depth. For this, we would have to use a stereo configuration where the baseline is large enough because larger the baseline more precisely we can make disparity measurements. For point PL, how did we get the corresponding PR?

TEMPLATE MATCHING

Template matching refers to the image processing where we find similar templates in a source image by giving a base template to compared on.

The process of template matching is done by comparing each of the pixel values of the source image one at a time to the template image. The output would be an array of similarity values when compared to the template image.

To be able to look at the similar templates found on the image, we can find the peaks in the resulting array of the template matching.

We first take a window/block of pixels in the left image with PL as the center of that window.

Now we need to use this window as a template to find a matching window of the same size in the right image.

But here's a catch! We don't need to scan the whole right image to find the matching window. The corresponding window must lie in the same horizontal line (called scanline) in the right image. A parallel stereo camera system doesn't give any disparity along the vertical direction. In other words, yl=yr. After the calibration we can use the stereoRectify() OpenCV function. Where R1 is the 3x3 Rectification Transformation (rotation matrix) for the first Camera and R2 for the second camera. p1 is Projection matrix 3x4 in the new and rectified coordinate system of the first camera and p2 for the second camera. Q is the disparity matrix by depth 4x4. The initUndistortRectifyMap() is called for images from both camera, so two times, here are computed the undistortion and the rectification transforms. Next we can use the remap() function to do the geometrical transformation in the images.

Advantages of multiple cameras • In the multicamera case, we use correspondences between what is seen from each of the separate cameras to draw conclusions about where the object is (i.e., by triangulation). • The advantage of such a technique is that it will work with even unknown objects or entire unknown scenes. • The disadvantage is that it requires multiple cameras.

VI. Results and Observation

Compare your results with all the available algorithms which you may have used to tackle the PS. If possible,

present your and their results in tabular / graphical format.

Explain the trend of results in details. Mention the drawbacks (if any) of your algo compared to other algo and the reason of picking up the approach over the other if you have implemented any algo over the other.

Although the result of the experiment was not known the disparity map looks clean and according to a human perception of the world is rather true.

The resulting distance was found out to be 9.71537 meters. and the location around (547,479)

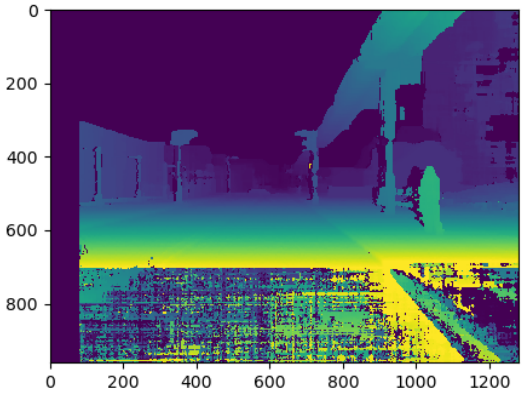

Fig. 3. Disparity map

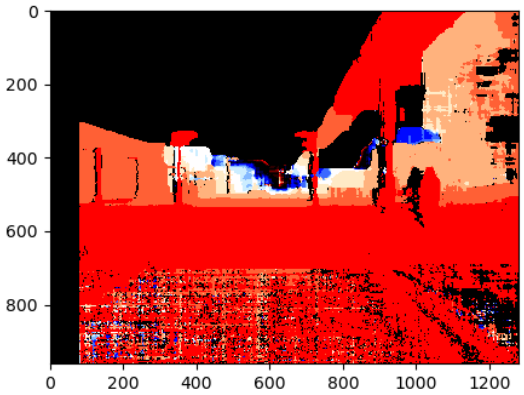

Fig. 4. Depth map

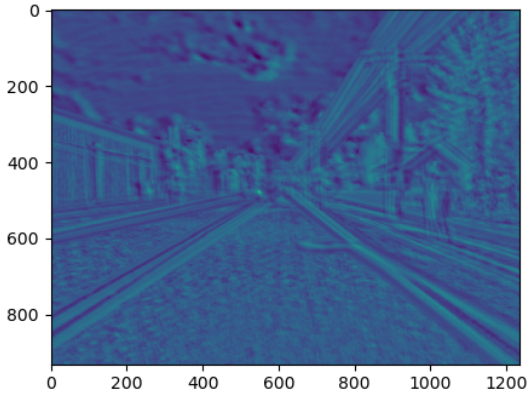

Fig. 5. Cross correlation map

VII. Future Work

After the template matching, we add a constant to the pixel where the templates corner lied so as to bring the point of which we calculate the depth to center. This would have to be readjusted if the size of the template changes one woy was to add half the length and breadth of the size of themplat. In cases of much larger objects; imagine an oblique house. We would calculate the distance of all the points and then have more of an idea of where to go. This code could be modified so that it takes the bike location on the 2d and the depth and actualy makes a bot move so that it avoids the obstacle.

CONCLUSION

Write overall about what the problem was, how you solved it, difficulties faced and the net output with it's usefulness to ARK or in general. We here have to estimate the distance of the camera from the given obstacle which is a bike given to us in the form of an image. This could be used in Aerial robotics to determine the distance from ground/walls or objects. This could be used to map a terrain to find a suitable point to land in a new region. Also could be used to detect any object better as we now have the ability to know what's the subject of the image. Could be used to locate an object to be carried.

References

[1] https://docs.opencv.org/4.x/

[2] http://tnt.etf.bg.ac.rs/ mv/MV1718$_v$4.*pdf*