Felipe Moreno
Collaborators: None
6.869 Advances in Computer Vision
May 8, 2020

Robotic Stereo Depth with Heterogeneous Cameras

1 Abstract

Currently, Jibo, the original social robot, lacks plenty of functionalities after its company Jibo, Inc abruptly shut down in 2018. Jibo is currently used by researchers in the Personal Robots Group in the MIT Media Lab, to conduct user studies in the field of social robotics. In this project, we develop 3D reconstruction functionality and object detection on Jibo. The motivation is that currently Jibo lacks the functionality to interact with the environment and to have a semantic understanding of its surroundings. In this project we the solve depth estimation problem using the stereo reconstruction from Jibo's two frontal cameras.

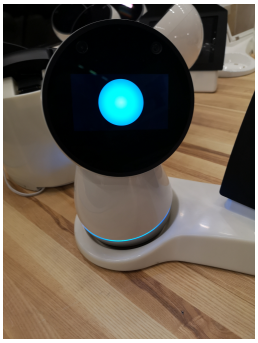


Figure 1: Jibo robot

2 Introduction

The Jibo robot was created with the goal to become the first social robot for the home. By the time the company Jibo, Inc. shutoff in 2018, due to a series of mistakes and the ruthless competition with smart speakers such as Alexa and Google home with costs under 50$ as compared to Jibo which sold for a price of $899 at the time, only an SDK with limited functionality was released to developers to produce third-party apps. Since then Jibo has been kept alive by researchers and, in particular, the MIT Media Lab Personal Robots Group which to this day uses Jibo to deploy user studies on Social Robotics.

Jibo features a peculiar design. It is a statically fixed to the ground i.e it does not have legs to move, yet its torso has 360deg of freedom to rotate around and it's head the ability to look upwards and downwards. For that reason, it is great interest for Jibo to have an understanding of its environment. Future studies plan to have Jibo deployed on living spaces where it can extract meaningful information from its surroundings to better engage with the user. Jibo has two frontal cameras as shown in Figure(2).

In this work, the focus is the integration of depth perception for environment understanding.

(a) Jibo frontal cameras. Left is a narrow lens camera and right is a wide lens.

(b) Jibo camera board inside view

(c) Jibo camera board side view

Figure 2: Jibo camera board setup

3 Related Work

In the literature review, we explored multiple approaches to having Jibo reconstruct the depth maps of its surrounding environment in real time. We reviewed many possible methods using both cameras or a single camera. We looked at to recover intrinsic from a single image from [Tappen et al.(2003)Tappen, Freeman, and Adelson]. We also explored the idea of using video to extract depth maps as done in and [Koch et al.(1998)Koch, Pollefeys, and Van Gool]. We looked into the deep learning literature to understand data driven ways to approach the problem as done in [Gur and Wolf(2019)].

However these approaches fail to be ideal for our problem - real-time depth estimation. This problem has been extensively studied in robotics with stereography. We choose this approach because of its ease of implementation and the ability to make use of the robot's default hardware. This is particularly important to reduce the cost deploying user studies with jibo and to reduce the complexity of integrating additional components into the platform.

Lastly we looked at some algorithms for computing the disparity maps between stereo images as described in which uses a Graph Cuts algorithm to achieve high performance [Tappen and Freeman(2003)]. But we decide to go with using SURF [Bay et al.(2006)Bay, Tuytelaars, and since it is a fast and high performance algorithm for keypoint matching which is very appropriate to use in our application. Since the robot, is expected to estimate depths in real time performance is one of our main concerns. Its worth noting that SURF is a patented algorithm, but it can be freely used for research purposes which is our academic goal.

We also explored the possibility of replacing the cameras for two cameras of the same type. As it turns out, the platform that the robot runs expects the cameras to be the given ones and replacing the cameras would be a task that needs to also be handled at robot's architecture level.

4 Approach

As expected by a company that abruptly shutoff, the Jibo documentation was very obscure and hard to dig into. Our first approaches to the problem were optimistic about being able to find a solution without completely knowing the actual hardware of the robot nor the software running on it. We were able to get images from the camera using API calls yet were uninformed about any processing done on the images extracted through the API calls. The resulting images looked as show in Figure(3).

(a) Capture made with the narrow lens camera.

(b) Capture made with the wide lens camera.

Figure 3: Raw camera captures

Digging through the documentation, we found the part numbers for both of the robot's cameras. We discovered that their field of views were 82° and 131°. The documentation referred to the wide lens camera as a 'fisheye camera' and as a 'wide lens camera' interchangeably. For this reason, we tried different camera calibration approaches on this camera until we found good empirical results.

For camera calibration we used the chessboard detect corners approach to compute the intrinsic of each camera. We then undistorted the images and cropped the region of interest. With a pinhole camera model calibration method we achieve good results for the narrow lens camera.

Calibrating the wide lens camera was much more difficult. First attempt was to calibrate using the fisheye camera model or the pinhole camera model which gave terrible results as shown in Figure (4). The rational camera model is described, in the OpenCV documentation as computing 8 distortion coefficients instead of the standard 5, gives much better results although we still find some distortion towards the sides of the image. The solution we found was to increase the number of calibration images, and add more diversity to the calibration set by placing the chessboard at different depths.

We then proceeded with the next stage of the problem, image rectification. We decided to use an asymmetrical chess board to avoid the epipolar lines to be detected in two different orientations. With this step, we achieve the results of a rectified image as shown in Figure (6).

(a) Undistorted image using the fisheye camera model.

(b) Undistorted image using the pinhole camer model.

(c) Undistorted image using the rational camera model.

Figure 4: Difficulties in calibrating the wide lens camera.

(a) Wide lens camera rectification result.

(b) Narrow lens camera rectification result.

Figure 5: Rectification results

Now that we have rectified images, we proceeded to find matching keypoints between the two rectified images. We used SURF algorithm to detect the matching keypoints, we then used Lowes distance ration method to filter out bad matches. These results look like Figure(6).

From the disparity maps, we can triangulate the depth of every keypoint using the focal length of the cameras and the baseline distance between them.

5 Experimental Results

For system validation, we measured 10 objects in the environment at various distances from the robot. We then predict the depths for all of the objects in question by averaging all the keypoints in each object. Predicted depths are reported in Figure(7).

For each object, we calculated the error as the absolute difference between the ground truth and the prediction. We clustered the objects into 3 range distances from the robot (0.5m, 1.5m), [2.5m, 3.5m), [3.5m, 4.5m). We report the mean errors for each range in Figure(8). We observe high accuracy for nearby objects, but increasing errors the farther the object is

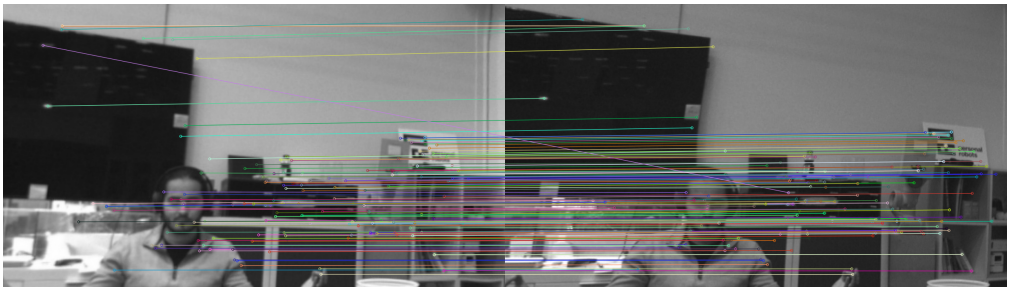
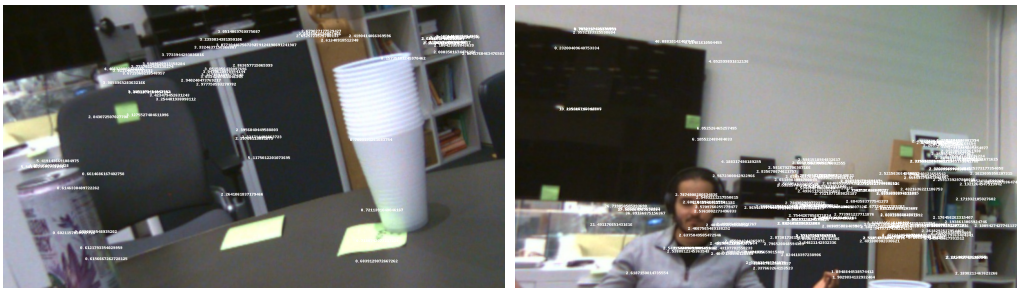


Figure 6: Keypoint matching pairs



(a) (b)

Figure 7: Keypoint depth results

from the robot.

6 Conclusion

The proposed method is successful at measuring with high accuracy objects nearby to the robot. Nevertheless, far away object depths, can't be reliable determined. For future work, we propose using belief propagation to further refine predictions based on past observations as done by [Xie et al.(2017)Xie, Chen, and Orchard]. Also future work will hope to incorporate the robot's semantic understanding of the environment with depth perception and all of this functionality to run natively on the robot. Currently, these results are obtained on fetched images from the robot. There is still work to be done at the architecture level to allow installation of libraries and packages in the robot that will allow for this project to run on the robot.

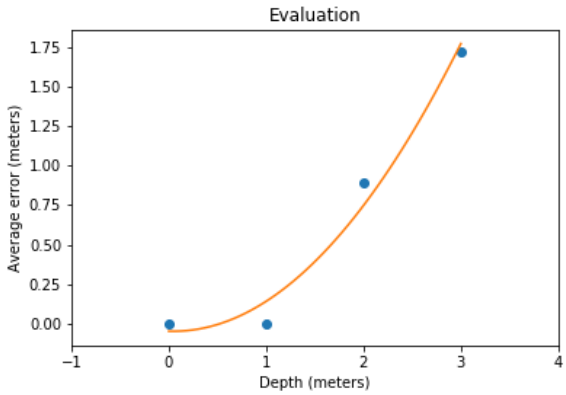


Figure 8: Measured error

7 References

References

[Bay et al.(2006)Bay, Tuytelaars, and Van Gool] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[Gur and Wolf(2019)] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7683–7692, 2019.

[Koch et al.(1998)Koch, Pollefeys, and Van Gool] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *European conference on computer vision*, pages 55–71. Springer, 1998.

[Tappen and Freeman(2003)] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *null*, page 900. IEEE, 2003.

[Tappen et al.(2003)Tappen, Freeman, and Adelson] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. In *Advances in neural information processing systems*, pages 1367–1374, 2003.

[Xie et al.(2017)Xie, Chen, and Orchard] Zhen Xie, Shengyong Chen, and Garrick Orchard. Event-based stereo depth estimation using belief propagation. *Frontiers in neuroscience*, 11:535, 2017.