Building an Autonomous Indoor Visual SLAM System using an iRobot and Kinect

Chase Lewis, Kang Zhang, Xin Li
Center for Computation and Technology, Louisiana State University

INTRODUCTION

The SLAM system is a key process in robotic navigation that is becoming increasingly prevalent in society, with applications such as self-driving vehicles and exploratory rovers. Effective SLAM will allow for more accurate environment mapping and safer operations. For this reason, we have been working on developing an implementation of a low-bandwidth image-based SLAM system that does not rely on expensive radar technologies. The previous work on the project involved attaching a Kinect and Raspberry Pi to an iRobot. The Pi is used both to control the robot and to collect the image data from the Kinect and stream it to a remote computer, where the main processing is carried out. To continue the project, this summer we focused on stitching the sequential images streamed from the Kinect into a single, three dimensional point cloud representation.

Figure 1: The Google Self-Driving Car, which implements advanced SLAM techniques

Figure 2: The iRobot mounted with the Kinect, which is used to take RGB and depth images.

METHODS / DEFINITIONS

- Simultaneous Localization and Mapping (SLAM) : The process of creating a map of an area while tracking the sensor unit in the space of said map.
- Speeded Up Robust Features (SURF) [1]: An algorithm used to detect and describe key points in an image.
- Random Sample Consensus (RANSAC) [1]: An algorithm used to find corresponding key points based upon the descriptors returned by SURF.
- Depth image: A standard greyscale image where the color value represents the distance from the camera.
- Point cloud: A 3D representation of a scene, represented using a set of spatial points with X, Y, and Z coordinates.

PROCESS

The constructed system begins by streaming multiple images from the Kinect, containing both RGB and grayscale depth format for each image. It then follows the process outline below to take the images, covert them into a three dimensional representation, and merge them together into a single map.

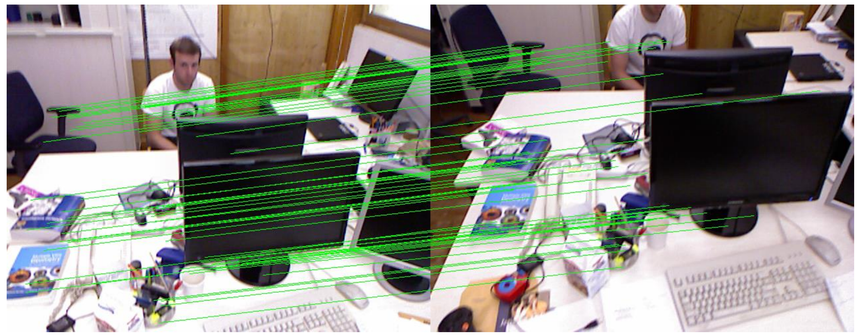

Figure 3 [2]***:*** The RGB images, with lines connecting the matching key points.

Step 1: Apply the SURF and RANSAC Algorithms to detect the corresponding key points in the two images.
Step 2: Convert the detected key points into an XYZ-coordinate format using the depth information from the grayscale depth image.

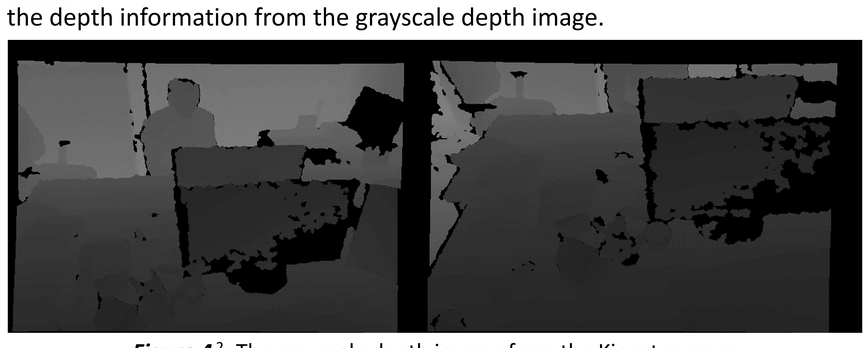

Figure 4 [2]***:*** The grayscale depth images from the Kinect camera

Step 3: Using the key points from step 2, create a transformation matrix describing the relationship between the images.
Step 4: Create a point cloud representation of the images by converting the pixels to XYZ-coordinates using the depth from the grayscale images and the color from the RGB images.

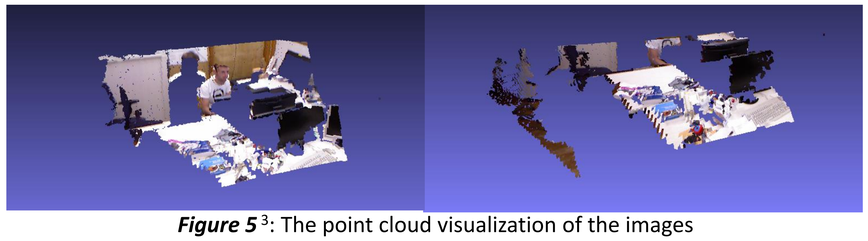

Figure 5 [3]: The point cloud visualization of the images

Step 5: Apply the transformation matrix from step 3 to the first point cloud, which adjusts the coordinate points to align with those in the second cloud.

RESULTS

The image processing aspect of the project succeeded, however attempts to incorporate the code with the previously created parts of the project were unsuccessful. There remains a lot of potential for the implementation of additional features and improved functionality.

Figure 6: The stitched point cloud of the images shown in the "process" section.

FUTURE WORK

Potential focuses for this project in the future include:

- Integration with the Kinect and iRobot, allowing for a functioning implementation of SLAM
- Completion optimization to decrease runtime and increase support for higher framerates
- Replacement of the RANSAC algorithm with graph matching for improved performance and greater accuracy
- Implementation of support for more complex operations, such as loop closure and moving object detection

REFERENCES

OpenCV [1] : http://opencv.org/
Kinect Image Library [2]: http://vision.in.tum.de/
MeshLab [3]: http://meshlab.sourceforge.net/
Feature Detection Support: http://pyimagesearch.com/
Google Self-Driving Car:
https://google.com/selfdrivingcar/

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under award IIS-1320959, OCI-1560410 with additional support from the Center for Computation & Technology at Louisiana State University.