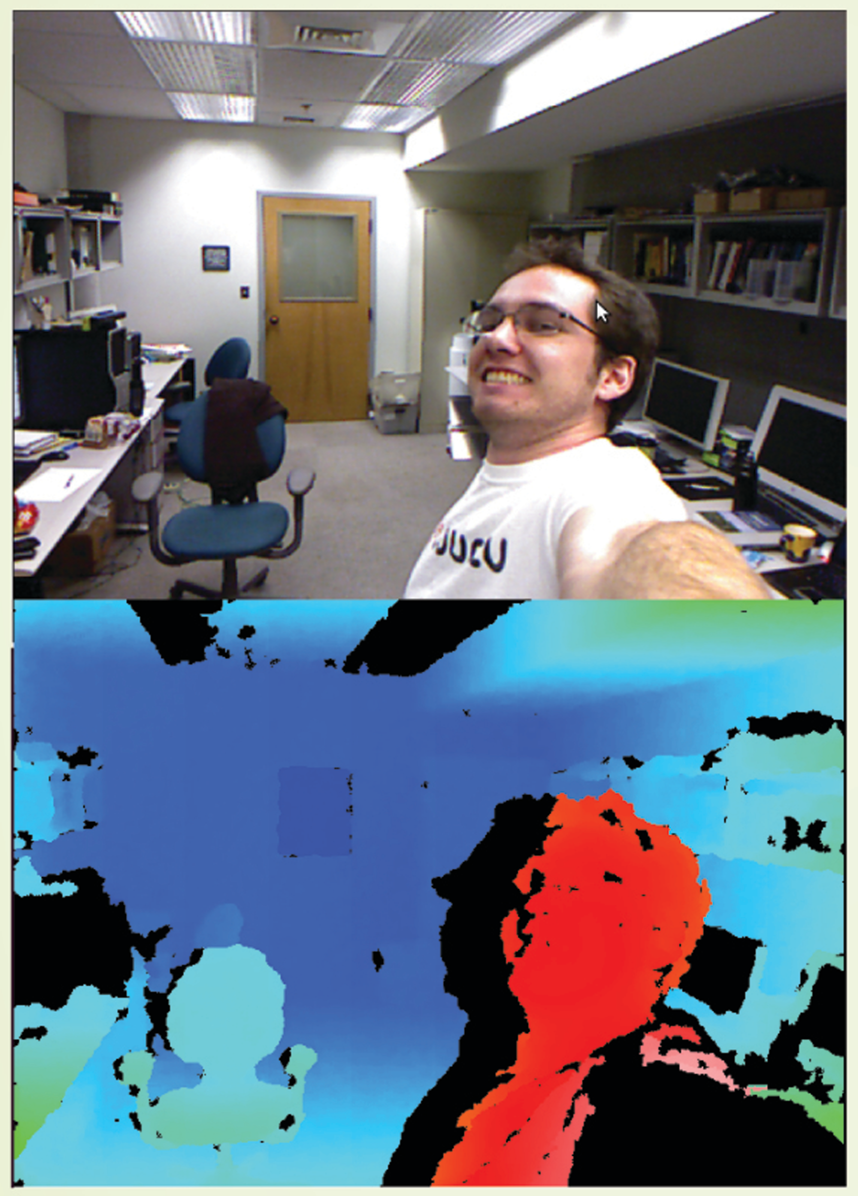
Department of Mathematics
Florida State University

3D Point Cloud Registration Using Multiple Kinects

Nathan Crock
Washington Mio PhD.
Xiuwen Liu PhD.
Gordon Erlebacher PhD.
Mark Susman PhD.

Department of Scientific Computing
Florida State University



Above is the color (RGB) and color mapped depth image extraced directly from the Kinect. The black shadow in the depth image is due to occlusion. In my reseach I hope to minimize occlusion by viewing the scene from multiple angles.

INTRODUCTION

Before any registration can be done, I need some data to work on. Accessing the information from the Kinect is the first step, using the libfreenect driver I acquire an RGB matrix and a depth matrix both 640x480 in dimension. The next step in the process of registration requires having correspondences. I begin finding the correspondences by locating SIFT features in each of the RGB images. A quick comparison between finger prints of all of the SIFT features gives me a set of potential correspondences. To narrow down the large collection of correspondences I use a filtering process which I'll elaborate on below. Once the correspondences have been located I perform an implementation of Procrustes Analysis to align the correspondences and ultimately the point clouds.

FINDING CORRESPONDENCES

To begin finding the correspondences, I first locate SIFT features in the RGB images from each camera. Brute force matching is done between the finger prints for each SIFT feature which creates a set of potential correspondences. Using an initializer from the user, my program filters out the bad correspondences according to a set of criteria. The criteria are based on information found by superimposing the images side by side.

The Correspondence is Dismissed if Certain Criteria Are Met

1) If the magnitude of the slope of the line between two correspondences is a certain threshold above the average slope of all approved correspondences.
2) If either SIFT feature in the match currently belongs to an approved correspondence.
3) If the line segment connecting the macth is a certain threshold longer than the average

The top image shows the remaining correspondences after my program filters out the bad matches. The bottom image is the full collection of correspondences found.

PROCRUSTES ANALYSIS

Once the correspondences are found and stored I need to find the transformation matrix which best aligns the correspondences. Let p_i be a point in one image that belongs to a correspondence.

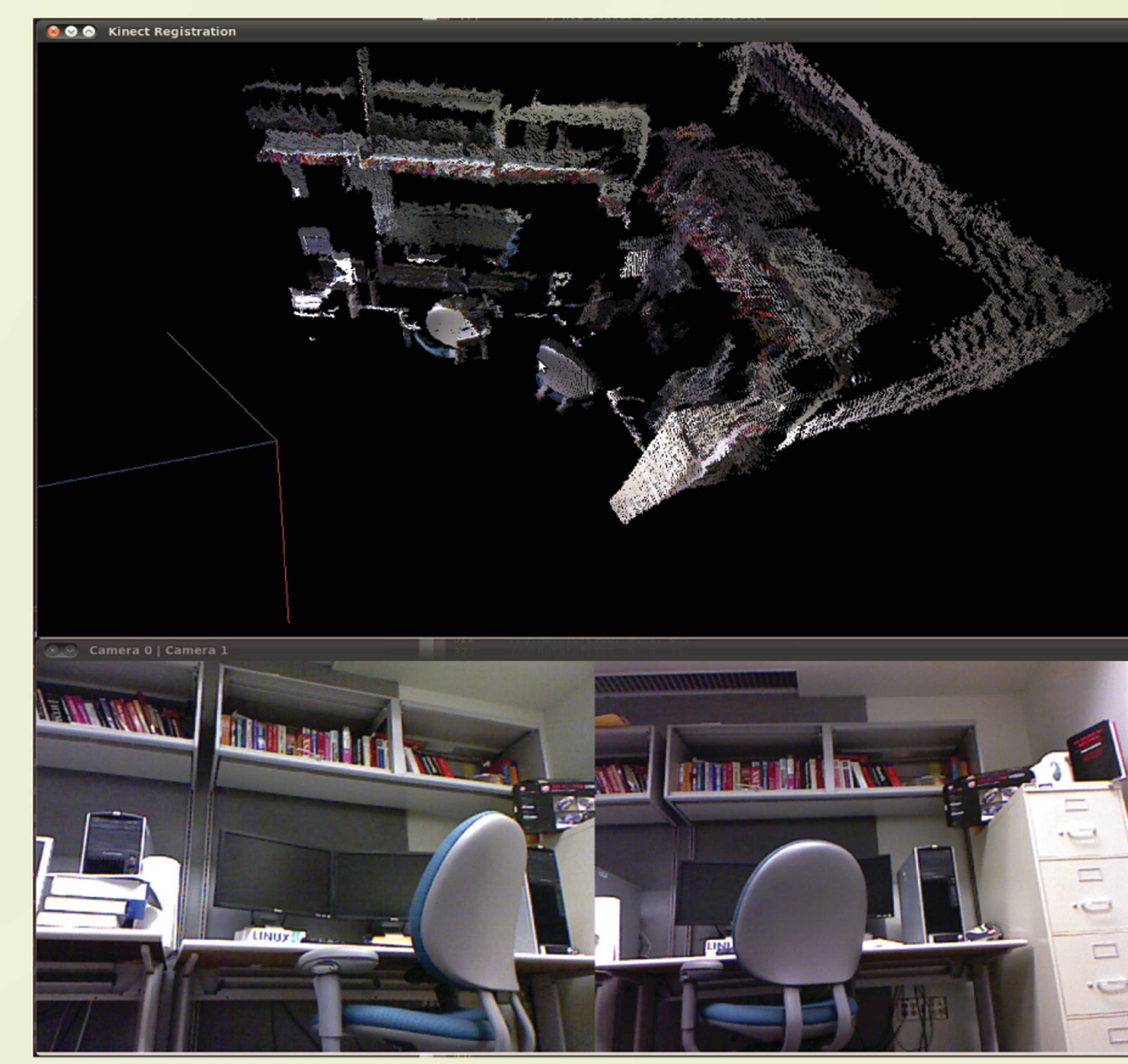
$$p_i = (x_i, y_i, z_i) \qquad P = [p_1, p_2, \ldots, p_{n-1}, p_n]$$

If we then calculate the centroid of the correspondences and move it to the origin, the point cloud will be translation invariant.

$$\bar{P} = \frac{1}{n}\sum_{i=1}^{n} p_i \qquad P_{new} = [p_1 - \bar{p}, \ldots, p_n - \bar{p}]$$

If we follow the same prodcedure with the corresponding points, let's call Q, then both P and Q will be centered at the origin, so the translation is done. All that remains is the rotation. If we post multiply P by Q transpose and take it's singular value decomposition, the product of U and V transpose yields the rotation matrix which minimizes the Euclidean distance between correspondences.

$$PQ^T = U\Sigma V^T \qquad R = UV^T$$



A projection of the scene observed from two Kinects

ACKNOWLEDGMENTS

I would like to provide scpecial acknowledgements to a particular few without whose help I would not have been able to complete this task. I'd like to thank Evan Bollig, Andrew Young, and Ian Johnson for sharing their OpenGL expertise with me. I'd also like to thank Dr. Mio and Dr. Liu for their input and advice to my procedure, and a gratious thank you to Dr. Erlebacher for his unrelenting efforts to motivate me and his passion and interest in my project.