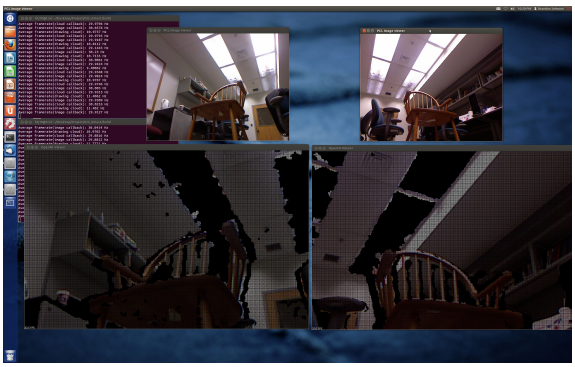
3D Point Cloud Recording Using Multiple Kinects

Brandon Johnson, Dr. Gordon Erlebacher, Nathan Crock
Department of Scientific Computing, Florida State University

INTRODUCTION

Since its introduction, the Microsoft Kinect has shown itself to be quite versatile in its ability to record both imagery and depth information in real time, leading to new paradigms of computer-human interaction. The goal of this project is to perform three-dimensional recordings of time-dependent movement with two Kinects, and replay this animation in real time with the ability to explore the animation as it plays. Challenges that are addressed are the registration of multiple point clouds into a single coordinate system, the merging of point clouds from multiple Kinects that are not synced with each other, and cleaning up the data in real time.



Point Clouds and Images from two Kinects

BACKGROUND KNOWLEDGE

To begin the project, a basic level of C/C++ is required to interact with the individual Point Clouds and vectors of point clouds such as using pointers and callbacks to functions. To obtain the point clouds, we need an unique combination of hardware and software. The hardware required were two Microsoft Kinects, as the camera from the Kinect can store RGB and depth data from its surroundings, creating a point cloud. The software needed for this project was OpenNI, which is a software interface created for interaction with the Microsoft Kinect, and the Point Cloud Library, which is a C++ library created to produce and interact with point clouds. Tutorials for the Point Cloud Library can be found at http://pointclouds.org.

(Rasu & Cousins, 2011)

METHOD: MANUAL REGISTRATION

To get a point cloud recording using one kinect requires no extra work and can be achieved entirely through computer programming. However, to get a 3D point cloud recording from multiple kinects, a process called registration is necessary. The goal of registration is to find correspondences in a given input dataset, or the point clouds, and rotate and translate the correspondences onto one another to create one consistent image. The algorithm used for this project was the Iterative Closest Point algorithm, which can be described as the following:

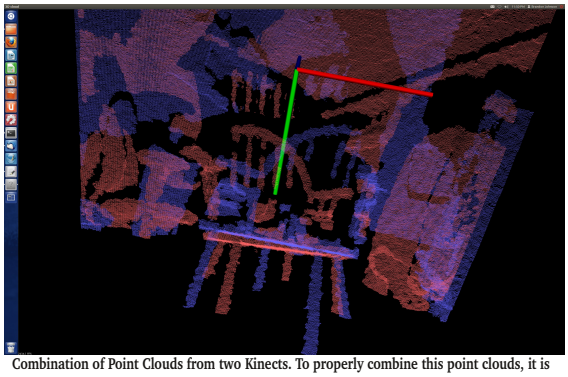
Step 1. Search for correspondences.

Step 2. Reject bad correspondences.

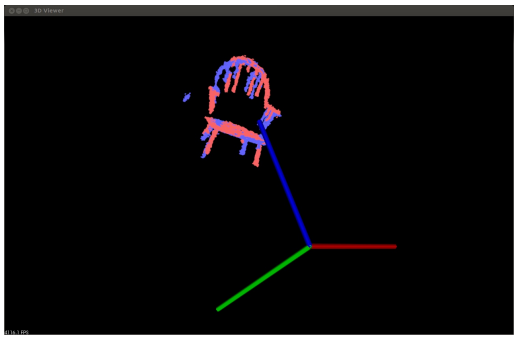
Step 3. Estimate a transformation using the good correspondences.

Step 4. Iterate.

(Holz, Rusu & Sprickerhof, 2012)



Combination of Point Clouds from two Kinects. To properly combine this point clouds, it is necessary to perform manual registration and the Iterative Closest Point Algorithm.



SUMMARY & FUTURE WORK

Point clouds can be taken from two Kinects and the frames from each Kinect are “synced” by using a multi-thread technique. Using this technique, corresponding point clouds from the individual kinects are save to their respective directories with the same file names. The registration process takes the first point clouds from each directory and manually combines the point clouds based on the features that are similar to both clouds. This process produces a transformation matrix that is then saved to a file. The transformation matrix is then applied to all files saved from both kinects and the corresponding clouds are combined, creating frames that can be viewed as a 3D recording.

One main concern within the project was the accuracy of the ICP algorithm on the point clouds. If the kinects do not store the point clouds at a certain angle, then the ICP algorithm fails. Future work includes finding a registration algorithm that allows a kinect to be further away from another kinect and being able to register the point clouds from both kinects onto each other.

REFERENCES

Holz, D., Rusu, R., & Sprickerhof, J. (2012). The PCL Registration API. The Point Cloud Library. http://pointclouds.org/documentation/tutorials/registration_api.php

Ben-Chen, M., & Vaxman, A. (2011). Iterative Closest Point. Digital Geometry Processing – Registration. http://webcourse.cs.technion.ac.il/236329/Spring2011/ho/WCFiles/10 – Registration.pdf

Rasu, R., & Cousins, S. (2011). 3d is here: Point Cloud Library (PCL). The Point Cloud Library. http://www.gorerle.com/pcl/3d is here pcl.pdf

ACKNOWLEDGEMENTS

I would like to thank Dr. Gordon Erlebacher and Nathan Crock for their huge contributions to this projects, especially during the ICP and Manual Registration coding and implementation.