Software Engineering Project
Report

3D Scanning and Reconstruction

Week (1) Report 20/11 - 26/11

Project
Sculptura

Submitted by

Program	Names of Students
MAIA	Elizaveta Genke
MAIA	Daria Zatova
MSCV	Mohamed Ali
MSCV	Karim Botros

Under the guidance of
Professor Yohan Fougerolle
Tutor: Cansen Jiang
Tutor: David Strubel

UNIVERSITÉ DE BOURGOGNE
720 Avenue de l'Europe, 71200 Le Creusot, Tél : 03 85 77 00 70

Contents

List of Figures

Introduction

Recent Activity And Planning Phase

This week our team was reading the reports of the previous projects and their codes as well, then we have done small research on PCL and SVD.

Group Meetings Summary

Three meetings during the week:
1) Installation of the libraries and the previous year projects running;
2) Trials of data acquisition;
3) Project Planning.

Objectives

We wanted to know how registering and data acquisitions works, and how meshing is performed on the registered point clouds. We tried to acquire data with different codes and we were successful in that. We also wanted to figure out how the mechanical setup was working regarding the turning table and the elevating stand for the Kinect sensor. And we were able to configure and run the Kinect sensor as well as the mechanical parts to acquire the data.

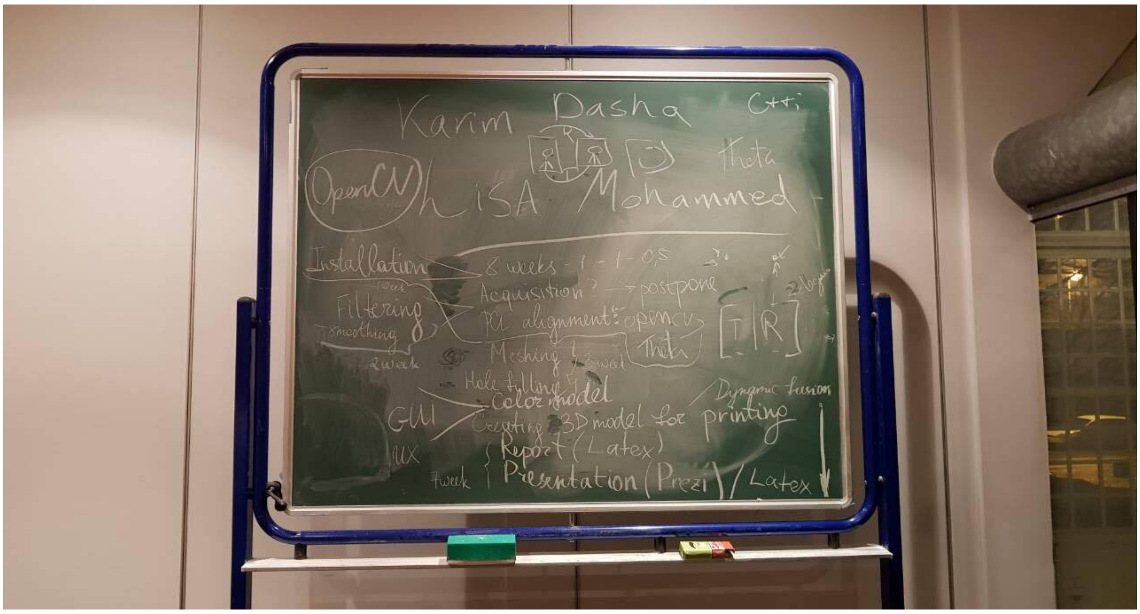


Figure 1.1: General plan figure

Figure 1.2: Data Acquisition

Work Done

Reviews

PCL Review

Review about PCL:
The Point Cloud Library (PCL) is a fully templated C++ library for 3D point cloud processing. PointCloud represents the basic class in PCL for storing collections of 3D points.
It contains the following mandatory data fields:
width, height, points, is.dense (specifies if all the data in points is finite (true), or whether it might contain Inf/NaN values (false)).
For example, the following point cloud will contain points with X, Y and Z data:

$$PCL :: PointCloud < PCL :: PointXYZ > cloud$$

Point clouds can be stored to disk as files, into the PCD format:
Point Cloud Data (PCD) file format v .5
FIELDS x y z rgba
SIZE 4 4 4 4
TYPE F F F U
WIDTH 307200
HEIGHT 1
POINTS 307200
DATA binary
The PCL framework contains numerous algorithms including filtering, feature estimation, surface reconstruction, registration, segmentation, visualization. Filters, features, segmentation all use the same basic usage interface:

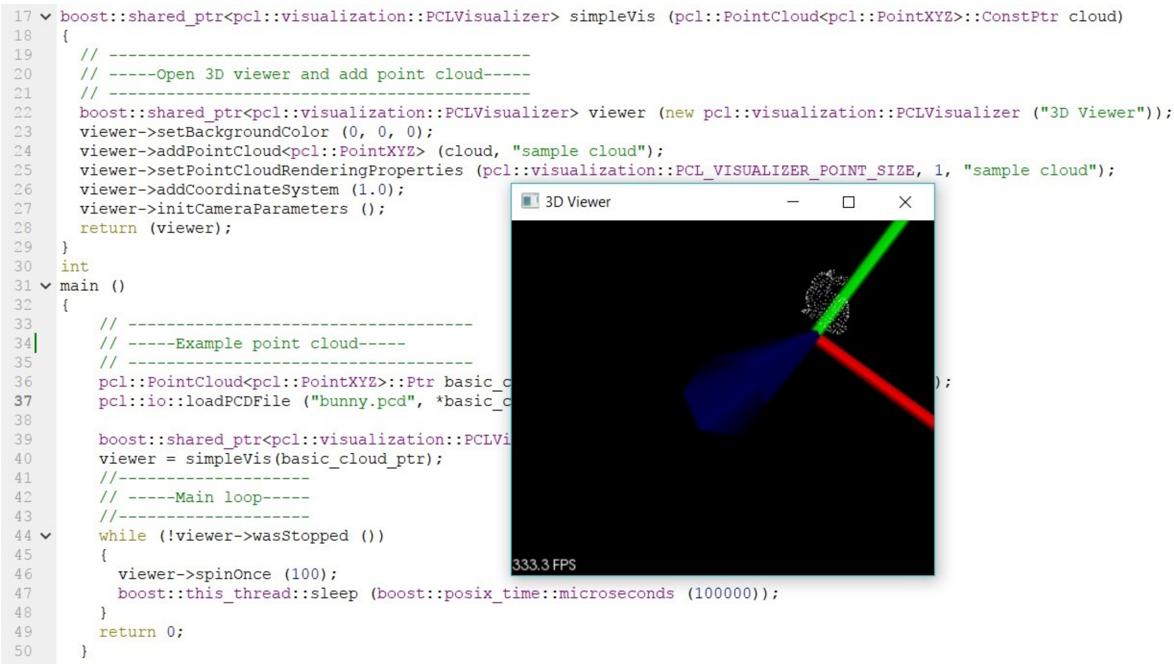
1. Use setInputCloud to give the input

2. Set some parameters

3. Call compute to get the output

Example of filtering:
$pcl :: PassThrough < pcl :: PointXYZ > pass$ //Create the filtering object
$pass.setInputCloud(cloud);$
$pass.setFilterFieldName("z");$
$pass.setFilterLimits(0.0, 1.0);$
$pass.filter(*cloudfiltered);$ //calls the filtering method and returns the filtered dataset in output

PCL provides many tutorials and sample codes [1]. In order to get familiar with main functionality we tried to implement some code in Qt creator. As conclusion, we can say that PCL has sufficient number of functions which will help us to achieve good results.



Basic Filtering code

SVD Method

A review on SVD method:

SVD for Point Cloud Matching Singular value decomposition (SVD) is method that can be applied to any matrix: $M = U\Sigma V^T$, where U - left singular vectors matrix, V - right singular vector matrix, $\Sigma -$singular values matrix. It has many applications for linear algebra problems: matrix rank, matrix inverse, solving linear equation systems etc. We can use it for shape matching problem.

Suppose, we have two point clouds of the same rigid object P and Q, but from different perspectives (points of view). We can say then, that q is the same p cloud (almost) but somehow rotated and translated. So we want to find rotation and translation matrices (actually, translation matrix is simply a vector). These two problems can be solved separately, as we can centralize both p and q clouds (get rid of translation component). So, first we find rotation matrix R and from this we can find translation vector t.

As described in the article [2], to find rotation matrix we should calculate covariance matrix S $=XWY^T$ (X and Y are set of points centralized, W is weight matrix). Then we do singular value decomposition S $=UWV^T$. Due to some analysis we consider that the case when M $= V^TRU = I$ is the best for minimizing the error, from this R$= VU^T$. (regarding this source [3], we also need to transpose it, and this source [4] tells that if determinant of a matrix R is equal to -1 its a special reflected case and in such case we need to multiply the third column by -1). After we find translation vector t $=$ $\mathbf{p}$ - R$\mathbf{q}$ ($\mathbf{p}$ and $\mathbf{q}$ are centroids of these clouds).

But SVD is just one of the point cloud registration. Good results also gives ICP especially when its prealigned using SVD [5].

Results

Results for the week:

- We have 1 and 2 projects of the previous running on all of our laptops (installed PCL 1.8.0, VTK, IntelR SDK, Kinect SDK libraries)
- We had a discussion and now we have list of things we would like to implement and improve (ordered by preferences and importance)
- Elizaveta has made a review on SVD method
- Daria has written a review on PCL library and proposed Gantt diagram for the project management
- Karim acquired data from Kinect v.2 and have read some parts of previous projects codes and Reports

- Mohamed has done research on some topics (OpenCV, shape matching), shared many useful resources, created a GitHub repository.

To do list

Plans for the next week:

- We will have a session about SVD/PCA/ICP (theoretical approaches for point cloud registration)
- We will discuss with David/Cansen are we obliged to use OpenCV
- We'll install OpenCV library on our laptops and try to implement OpenCV functions for point clouds alignment
- We'll meet for a session on reading code of previous years projects

References

[1] PCL tutorials http://pointclouds.org/documentation/tutorials/

[2] *Least squares rigid motion using SVD* Olga Sorkine-Hornung and Michael Rabinovich https://igl.ethz.ch/projects/ARAP/svd_rot.pdf

[3] ICP with SVD Hands on http://www.markuszancolo.at/2015/12/icp-with-svd-hands-on/

[4] Finding optimal rotation and translation between corresponding 3D points http://nghiaho.com/?page_id=671

[5] *A Survey of Rigid 3D Pointcloud Registration Algorithms* Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn https://repository.uantwerpen.be/docman/irua/1ab789/4a6a3c9a.pdf.