Software Engineering Project
Report

3D Scanning and Reconstruction

Week (5) Report 25/12 - 31/12

Project
Sculptura

Submitted by

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

Under the guidance of
Professor Yohan Fougerolle
Tutor: Dr.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

Group Meetings Summary

Meeting Discussions and Work:
1) Meeting for GUI testing and RGBd information grabbing.

Objectives

For this week our goal was to capture RGBd data, continue developing GUI (in particular, adding QVTK Widget). Also, we wanted to test different filters to "clean" point clouds and implement ICP refining alignment.

Work Done

Summary

GUI development

In our application we would like to show preview from camera in a dedicated window before starting scanning. While everything worked well when camera was opening in a separate window, it started to crash when we tried to show preview in a label. Because there were too many operations in a main thread. To solve this issue, we have created a new thread called camerapreview. In this thread we take frames from camera, once the frame is ready, it sends a signal to render the image in a window.

```
preview = new CameraPreview();
QObject::connect(preview, SIGNAL(frameReady(QImage)), this,
                         SLOT(renderFrame(QImage)));
```

All format conversions work properly, as a result, we may see an RGB preview without any delays or errors.

This week we also implemented visualizer of point clouds. Since program crashed, when we tried to upload files in main thread, the new class read_point_clouds was designed in order to read files separately. Once the point cloud has been read, it is stored into vector of all point clouds and by default the first point cloud is shown in VTK visualizer.

```
readPointClouds = new ReadPointClouds();
QObject::connect(readPointClouds, SIGNAL(pointCloudsReady(std::
    vector<PointCloudT::Ptr>)), this, SLOT(savePointClouds(std::
    vector<PointCloudT::Ptr>)));
```

User can delete any point cloud from a list. The GUI has been upgraded so now it is resizable and more convenient for usage.

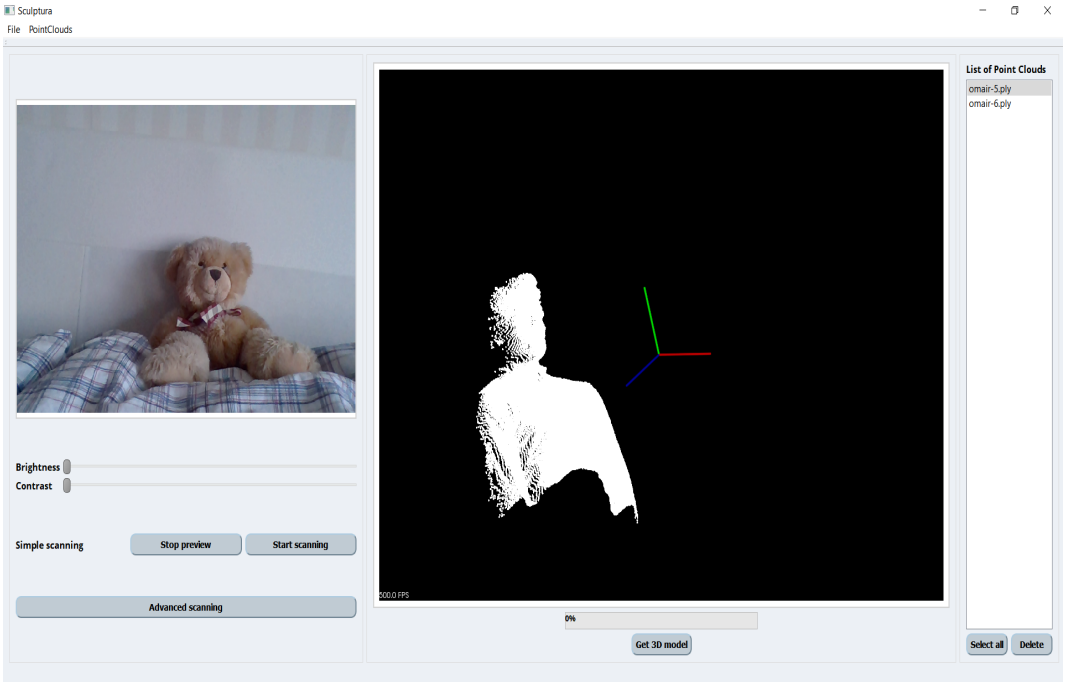

Figure 2.1: Main window updated

Results

Results for this week:

- We succeded with RGBd data acquisition using Kinect. We updated GUI, and were working on ICP implementation and filters.
- Daria worked with threads in order to make a camera preview (successfully tested with webcamera) and reading point clouds. VTK visualizer has been implemented. GUI has been upgraded, now it is resizable;
- Elizaveta was working on filtering and testing GUI;
- Mohamed was working on the ICP class writing;
- Karim managed to grab RGBd data and was testing GUI.

To do list

Plans for the next week:

- We will polish the code and link the parts of code that we already have together;
- We will test final implementation;
- We'll write final report.