Software Engineering Project
Report

3D Scanning and Reconstruction

Week (4) Report 11/12 - 17/12

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 with David for project discussion and information about final report, presentation and defense
2) Meeting for GUI designing and code implementation

Objectives

For this week our goal was to solve the problem with data acquisition from Kinect using OpenNI. Also, we wanted to do theoretical research on RANSAC method, continue the design of GUI and extend code for feature operations.

Work Done

Summary

RANSAC Overview

RANSAC (RAndom SAmple Consensus) is the method designed to cope with outlier problem (due to noise presenting in the images and change in the point of view there always will be outliers).

Algorithm of RANSAC [1]:

1. Select randomly the minimum number of points required to determine the model parameters.
2. Solve for the parameters of the model.
3. Determine how many points from the set of all points fit with a predefined tolerance ε.
4. If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
5. Otherwise, repeat steps from 1 to 4 (maximum of N times).

For choosing number of iterations we should solve the following equation for N[2]:

$$1 - (1 - (1 - u)^s)^N = p \tag{2.1}$$

where u - probability that point is an outlier, s - number of points in sample, N - number of iterations ensuring that we get at least one good sample, p - probability that we get a good sample.

From (2.1) we can deduce

$$N = \frac{log(1 - p)}{log(1 - (1 - u)^s)} \tag{2.2}$$

RANSAC divides points into inliers and outliers and yields estimate computed from minimal set of inliers. Then we improve this initial estimate using Least Squares minimization method.

GUI Template

For this week we managed to create template GUI in Qt Designer. To test it, we have used Signal/Slot approach to create certain events when button is clicked. At the figures 2.2 and 2.3 you may see an example of opening a web-camera.

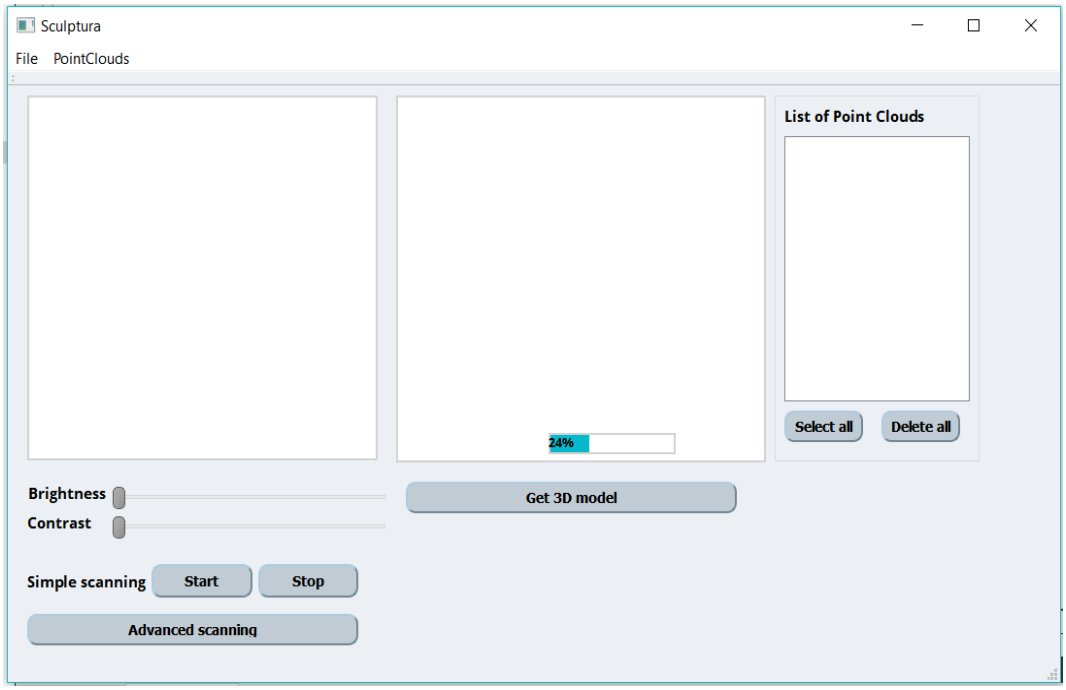

Figure 2.1: Main window

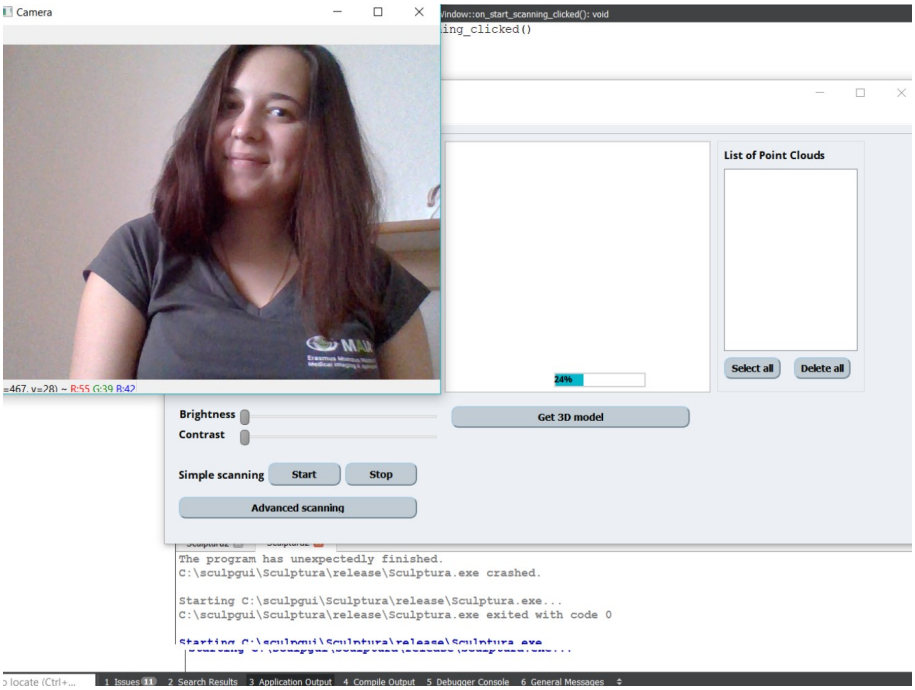

Figure 2.2: Start of Acquisition window

```
void MainWindow::on_start_scanning_clicked()
{
    VideoCapture camera(0);

    if( !camera.isOpened() )
    {
        cout << "Can not open a laptop camera." << endl;

    }
    for(;;){
        Mat depth;

        if( !camera.grab() ){
            cout << "Sensor1 can not grab images." << endl;

        }else if( camera.retrieve( depth, CV_CAP_OPENNI_DEPTH_MAP ) )

             imshow("depth",depth);


        if( waitKey( 30 ) == 27 )   break;// 27 is a number of ESC button

   }
}
```

Figure 2.3: Code to open a camera

Results

Results for this week:

- We continued our trials of data acquisition using Kinect. We managed to grab depth map in Qt from Kinect, but still struggling with acquiring both RGB and depth images.
- Trials of feature detection and feature matching provided nice results;
- Daria has designed template GUI and connected it using Signal/Slot approach with parts of code that we already have written;
- Elizaveta has written theoretical review on RANSAC method;
- Mohamed was working on the feature tracing and extraction of the rotation and translation from the images;
- Karim was working on Kinect connection and data grabbing and made progress in this problem.

To do list

Plans for the next week:

- We will finish work Kinect data acquisition and scan a rigid object and test feature extraction;
- We will continue our work on GUI;
- We'll finish designing class for 2D image processing (for feature extraction, matching etc.)

References

[1] *Overview of the RANSAC Algorithm* Konstantinos G. Derpanis `http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf`

[2] *Computer Vision I: Introduction to Computer Vision* Lecture Notes, CSE Department, Penn State University `http://www.cse.psu.edu/~rtc12/CSE486/`