Software Engineering Project
Report

3D Scanning and Reconstruction

Week (2) Report 27/11 - 03/12

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: 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 Cansen and David tutors to know more about OpenCV library
2) Creating of workflow process diagram
3) First draft of GUI creating.

Objectives

For this week our goal was to understand which steps we need to take in order to get 3D model and which libraries we use for these steps. We wanted to know how we transfer data, which data types we use on which step. For this reason we needed to define which functionality gives us OpenCV library. Another goal was to design structure of the program and draft for graphical user interface.

Work Done

Reveiws Summary and Critics

Workflow of data processing

First, we acquire RGB-d data with Kinect v.2 or Intel R200 sensors. Then we extract features from frames and do feature matching operation for further alignment of point clouds. For feature operations, we are going to use the OpenCV libraries functions. Then we calculate translation and rotation matrices with SVD method and refine this alignment with ICP (there is a possibility that we add RANSAC). After this, we calculate loop closure to exclude overlapping. Then we work with filtering/smoothing for denoising and better quality 3D model. After this we make 3D meshing. Finally, we do texture mapping to add color to our model. This algorithm is shown below in the figure 2.1.

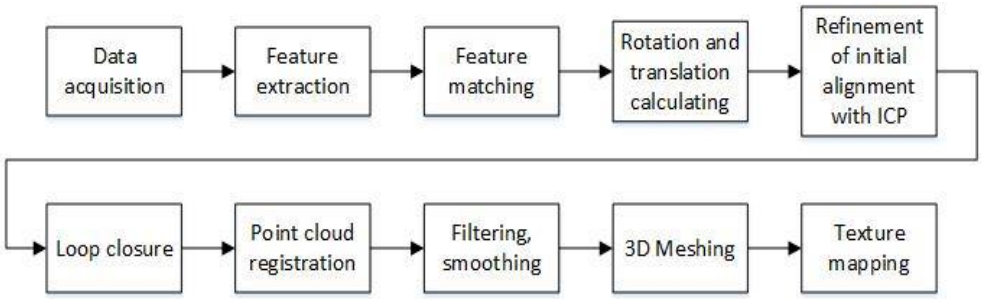

Figure 2.1: Data Processing Workflow

There are many decisions to take regarding the implementation (choice of libraries, choice of methods) and this will be more fully described in the next weekly reports as we reach these points. We have started designing classes for our program (fig.2.2).

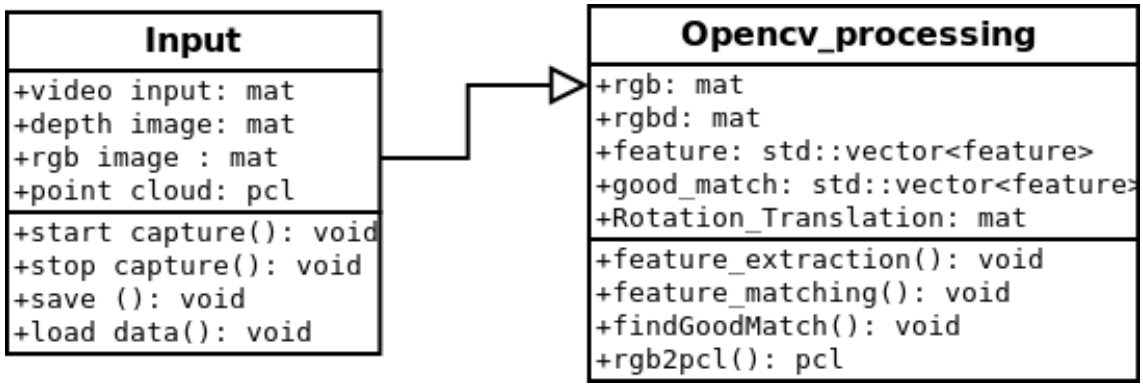

Figure 2.2: Initial class diagram

This picture briefly describes first two classes which we would like to implement. The first class Input is a class dedicated to work with Kinect or imported point clouds. The second class is a class for work with OpenCV (feature extraction and feature matching, rotation and translation calculating and finding best matches using RANSAC).

Iterative Closest Point (ICP)

ICP (Iterative Closest Point is one of the methods used for rigid point cloud registration. This method requires initial alignment (with SVD), then iteratively removes outliers, and redefines point correspondences. There are 2 most widely used variants of ICP: ICP Point-to-Point and ICP Point-to-Surface [1. The goal is to minimize Euclidean distance between corresponding points in two point clouds for Point-to-Point ICP, or scalar projection on planar surface for the ICP Point-to-Surface.

Graphical User Interface GUI

Since we have started implementing new method of point clouds alignment using OpenCV library, we face the need to create our own user interface. There are two ways of writing GUI:
1) We can do it manually (as it was for the project 3D-Korn)
2) Or we can use Qt-Designer.

The second method has more advantages as it helps to compose and customize main windows, widgets and forms created with Qt Designer integrate

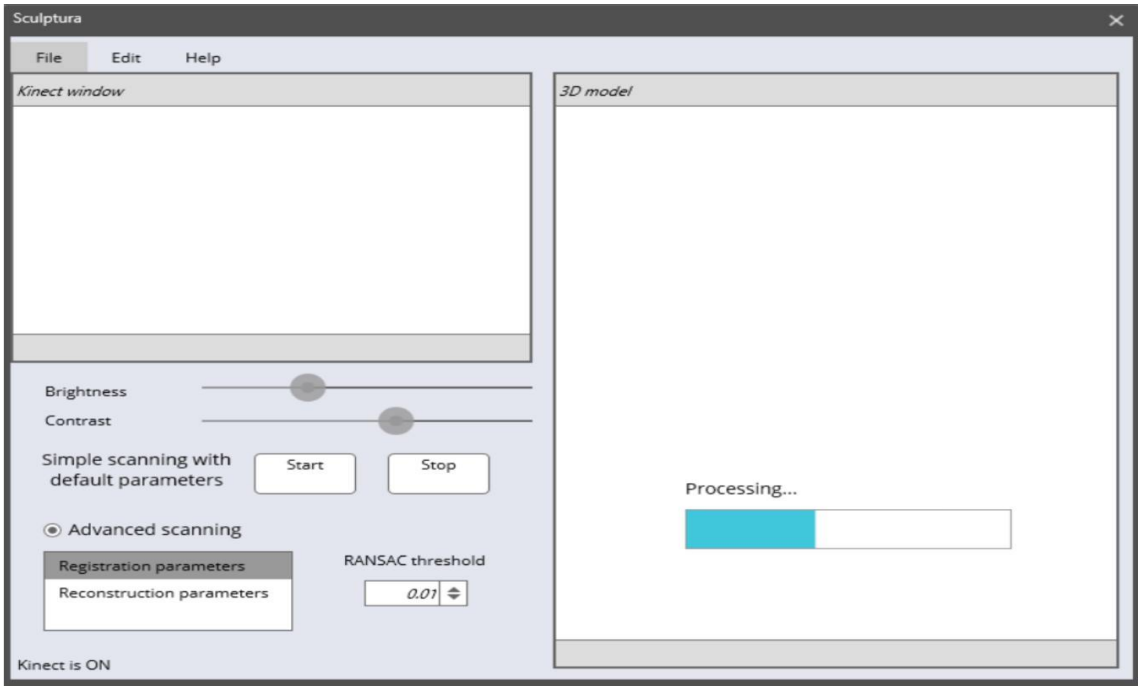

Figure 2.3: Initial Draft of GUI

seamlessly with programmed code, using signals and slots mechanism, so that we can assign behavior to graphical elements.

When we edit widgets, objects can be dragged from the main window's widget box and we can edit them, resize, set properties, so changes are seen immediately. The editing interface is intuitive for simple operations. It will also help us at the stage of testing new functions, since we can set different parameters through the main window without need to go directly into the code.

In order to get familiar with principals of creating GUI in Qt, a set of video tutorials was covered during this week as well as the documentation at the official website: url:`http://doc.qt.io/qt-5/qtdesigner-manual.html`. During the meeting we have described the functionality of our application that we would like to have and agreed on its style and structure.

Critics

We have the code for the previous year projects and we might use it for our project development. For this, we decided that its important to study their reports and code and find out what can be improved and what should we work on.

The first thing that we will work on is Data Acquisition. Students of the previous year were using Kinect not efficiently; they were scanning the whole body, so point clouds they got are noisy and poorly detailed. We would like rather move Kinect and scan from closer distance to get better quality RGB-d data.

The other (probably the main) issue is that they were using signal from encoder to calculate rotation. This decision has some disadvantages: firstly, it neglects translation component in point cloud transformation. Also, the signal from encoder can be corrupted by environment parameters and rotating table is not the common thing. We would rather suggest calculating translation and rotation using just images from camera using feature extraction and feature matching with OpenCV library.

From not satisfactory alignment of point clouds derives the problem of 3D meshing. Quality of 3D models is not that high as desired. Besides, we noticed that mostly 3D models created with Poisson method were not really good in sense of detalisation (faces, for example, were poorly reconstructed). Therefore, most probably, we will work with greedy triangulation algorithm, as it was giving the best results out of all. 4th group was working with hole filling, but it still can be improved. None of the group built colored 3D model that is also what we can implement.

Other than that, we would like to mention that code they were writing contains small amount of comments and that makes difficult to understand it. We would try to be as clear in our code as we can. We also are going to make our own graphical user interface to make it more user- and developer-friendly.

Results

Results for this week:

- Installation of OpenCV libraries and running both of codes examples provided by Cansen Jiang tutor on our laptops, the old and the updated codes;
- We have created the first draft for Class Design and GUI;
- Daria has started watching GUI in Qt-Designer course and read documentation and created a First Draft Visualization of the GUI;
- Elizaveta has written critics for previous year project and described workflow for data processing;
- Mohamed has designed code for feature extraction and proposed some classes to be implemented (class diagram);
- Karim has installed Meshlab for testing point cloud registration at the beginning of project development.

To do list

Plans for the next week:

- We will scan a rigid object and test feature extraction.
- We will continue our work on GUI
- We'll start to design class for 2D image processing (for feature extraction, matching etc.)

References

[1] *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`.

[2] Qt Designer Manual `http://doc.qt.io/qt-5/qtdesigner-manual.html`