Project : 3D Scanner

Software Engineering Progress Report : 6

Group Number : 2

Submitted to : Yohan Fougerolle

Dated : 13/11/2016

Centre Universitaire Condorcet - Master Computer Vision

1 Objective

The goal of this report is to briefly explain the development,difficulties and future aspects of the project 3D scanner with a small summary of overall progress.

2 Tasks Assigned Last Week

- Planar Removal
- Performing ICP
- 3D-Reconstruction
- Streaming through GUI Qt
- Code Optimization
- Individual Skills Improvement
- Management of the project

2.1 Planar Removal

- **Algorithm and results**

  First algorithm: Background subtraction in the streaming image.
  The previous algorithm is working on the stream image. Meaning the background in each images shall be removed before integration in the 3D format (for instance *.ply). So we can consider that algorithm as a preprocessing step to the main 3D rendering process. Fig.1 shows an overall view of the background subtraction using raw image stream algorithm:

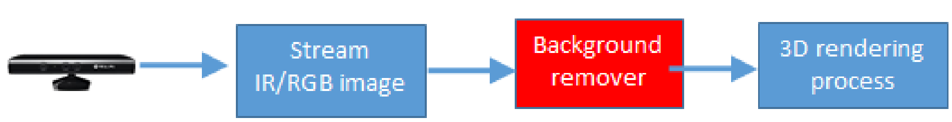

Figure 1: Steps for Background Removal

Second algorithm: Background subtraction after 3D rendering using planar surface adjustment.

This algorithm uses two adjustable planes to select the region of interest. Unlike first algorithm, this algorithm works after the formation of the 3D model which means we use this algorithm as a post processing algorithm. The overall view of the algorithm is as shown in Fig.2.

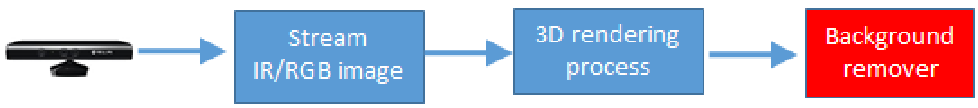

Figure 2: Steps for Background Removal

The second algorithm will be detailed in another report. However, only the result of the algorithm are presented here from Fig.3 to Fig.5. Fig.3 represents the original image and Fig.4 and Fig.5 show the target object and background. Here, background is defined as table which the object is placed over.

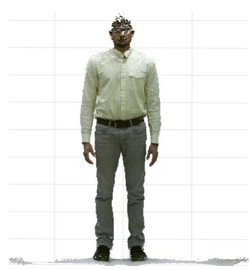

Figure 3: Original .ply 3D model

Figure 4: Target object

Figure 5: Background removed from original 3D mode

Note:

- A new report will be sent in the upcoming week to update previous report and provide very detail information about new background remover algorithm (second algorithm).
- A C++ code is uploaded for integration in the 3D scanner project.
- Github link : https://github.com/WajahatAkhtar/Project-S.E/tree/master/Source-code-3D-Scanner/Individual-Operation-Source-Code/Planar-Removal

2.2 Performing ICP

We are currently working on feature extraction of a point cloud, Once we are finished with that we will concatenate point cloud data and then apply Icp , we have successfully applied concatenation on 2 point cloud and trying to apply on incrementally increasing the the number of point clouds we are trying to stitch together, We have stitched images but still we are struggling to get the desired results. We tried to increase the the number of point clouds we took from kinect v2, We also took full body scan in order to get better results with mores scan of hands but still we were unable to get the desired results, we are using some sample code to perform icp and then making changes and understanding it once we get the desired results and understand the problem we will write the code in our own way. the links for the code is attached below.

Github link : https://github.com/WajahatAkhtar/Project-S.E/tree/master/Source-code-3D-Scanner/Individual-Operation-Source-Code/icp

The results we have got till now are given below.

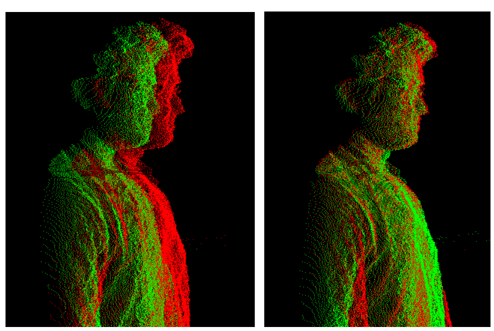

Figure 6: Left without ICP : Right with ICP

Results with Full body images are given below

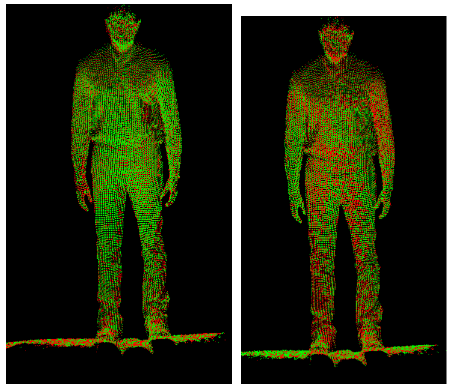

Figure 7: Left without ICP : Right with ICP

Problems: We are facing small problem in getting better result from ICP and trying to get the desired results if we start lacking on this task we will add more human resources for next week.

2.3 3D-Reconstruction

We implemented two 3D Reconstruction methods on test point clouds. The details are as follows.

Greedy Triangulation

We used Greedy triangulation (GD) for 3D Reconstruction in order to test out if it would be suitable or not. In that, we tried tuning its different parameters, like Variance (mu), Maximum Nearest Neighbours (D) and Search Radius (R) to optimize the output. We found out that the optimum parameters, for the test point cloud image (horse.ply), are: D = 50, R = 0.025 and mu = 5. The mesh we got at this point was neither smooth, nor water tight. We could also observe choppiness in the surface of the mesh.

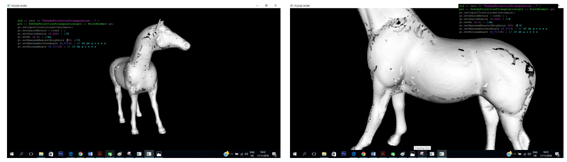

Figure 8: R = 0.025, mu = 2.5, D = 50

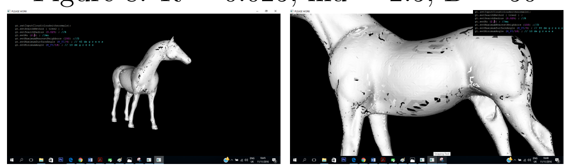

Figure 9: R = 0.025, mu = 2.5, D = 150

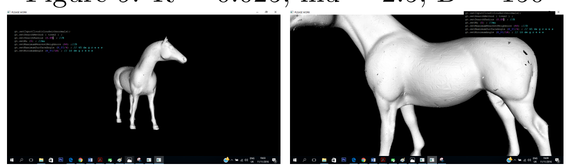

Figure 10: R = 0.05, mu = 5, D = 50

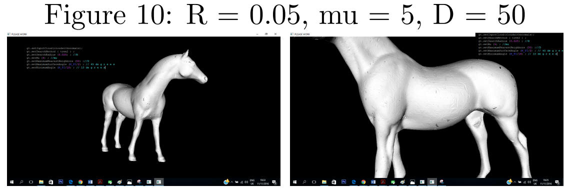

Figure 11: R = 0.0125, mu = 5, D = 50

Exploring Effects of Moving Least Squares and Laplacian Smoothing

In order to further increase the performance of the algorithm, we decided to apply the Moving Least Square (MLS) method to the input point cloud before doing 3D reconstruction using GD. Next, we applied the post-3D reconstruction smoothing filter i.e. Laplacian Filter (LF) to the output of the GD, in order to see if it could improve the result. We observed that both MLS and LF did not help much to improve the result of GD. Hence, we will need a better meshing algorithm to improve the final results.

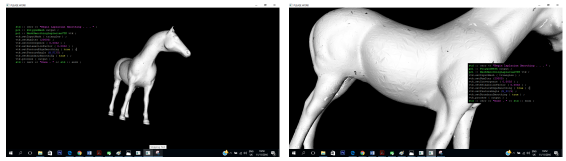

Figure 12: Result of Laplacian Smoothing

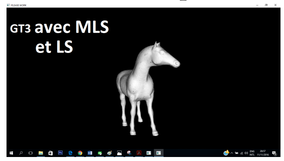

Figure 13: Result of Moving Least Squares and Laplacian Smoothing

Poisson Surface Reconstruction

This method closes all the borders with no openings of the points, which allow to have a reconstruction without holes, however, one of the biggest problem of this method is its highly sensitivity to the noise because it could produce over mesh to the reconstruction. Another thing to underline of Poisson method is that it has a dynamic variable calls D, which defines how detailed the mesh is going to be, this algorithm only works on the D range of 1-14, over this range the program doesn't respond.

In order to test its quality, it was provided some .ply images on the algorithm (REFERENCE OF THE ALGORITHM: Mesh Reconstruction Using The Point Cloud Library by Andrea Keiser, Ann-Marie H. B. Bech, Frederik V. Brentsen). The figures show the difference when we set the values of D of 4, 10 and 14 respectively applied to Omair images. It can be seen that the result of reconstruction is without holes, but it is not very clear so most of the details are not well represented

Description of the images below.

Figure 14. Frontal view.- a. D=4, time of processing: 02.1seg. b. D=10, time of processing: 18.4seg. c. D=14, time of processing: 9.12 min.

Figure 15. Lateral View.- a. D=4, time of processing: 02.1seg. b. D=10, time of processing: 18.4seg. c. D=14, time of processing: 9.12 min.

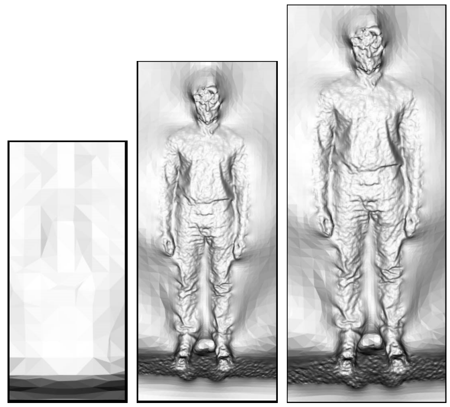

Figure 14: 3D Reconstruction of unstiched point cloud of omair front view

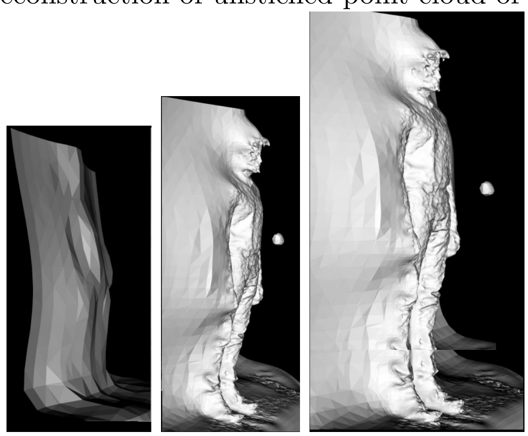

Figure 15: 3D Reconstruction of unstiched point cloud of omair Lateral view

Conclusions and Future Work

The GD algorithm does not produce a water tight mesh, even when the input point cloud is smoothed. According to our research so far, Poisson method for 3D Reconstruction will be desirable in order to create a water-tight mesh. We shall study more algorithms in-depth to better understand how to use them more effectively.

2.4 Streaming through GUI Qt

GUI Advancement VTK and Kinect Integration

During the last two weeks, we've been trying to integrate the VTK libraries inside the general GUI by means of the QVTK plugin provided by the VTK developers. This would give us the capability to show and manipulate from simple pointclouds to complete 3D models inside the program's interface.

However, such integration turned out to be difficult as the usage of those libraries required to compile the source code from scratch by using a full Visual Studio 2015 setup, the search of several external windows SDK dependencies, among other configuration issues.

After several days of making efforts to successfully compile the VTK libraries with Qt support and of gathering all the numerous dependenciees, we proceed to integrate the QVTK plugin into the GUI, along with some controls to manipulate the loaded pointclouds :

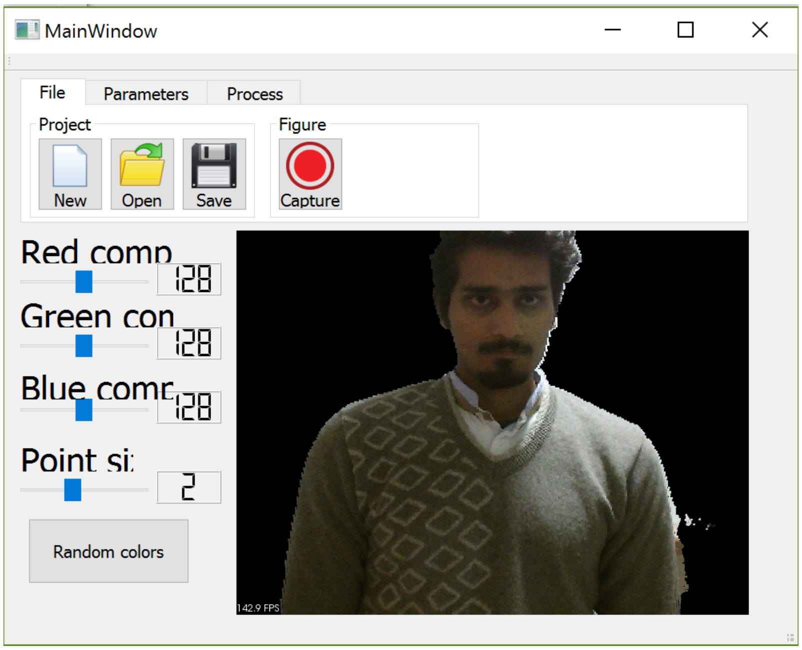

Figure 16: GUI

As seen in the previous figure, after clicking in the Open button and selecting a .PCD or a .PLY file, the pointcloud data is loaded and shown through the QVTK plugin. By clicking on the figure and dragging it, the user is able to move, rotate or zoom the model.

As a next step, we try to integrate the live Kinect streaming into the GUI. We make use of the available Kinect libraries as we intend to connect them also to the QVTK plugin in order to show the streamed data. The filtering-by-depth feature is also used, by setting a range in the Z edge from 50cm to 4.5m. After adding and configuring the corresponding libraries, we compile the entire project successfully.

However, by the time of writing this report, the program crashes after being executed. We suspect there is a buffering problem when communicating with the Kinect, and we are in process of finding a solution for this. The final goal of this stage is, after clicking the Capture button, to show the live information captured by the Kinect sensors, while registering the data into several .PCD files for its posterior processing.

The main codes, dependenccies and graphical ressources of the project (which includes, as mentioned before, pointcloud visualization and kinect streaming) can be found at :

Github link : https://github.com/WajahatAkhtar/Project-S.E/tree/master/Source-code-3D-Scanner/Un-Optimized-Full-source-Code/With-GUI

2.5 Code Optimization

As we have one team working on optimization and combining the code of all individual operation perform by different teams so the work they have done on can be seen on the following link below. We are looking to add ICP and 3D reconstruction soon in our optimized code and it will be the task for next week.

Github link Optimized Code (same as above) : https://github.com/WajahatAkhtar/Project-S.E/tree/master/Source-code-3D-Scanner/Un-Optimized-Full-source-Code/With-GUI

2.6 Filtering by depth

one of our team also filtered the result once again in order to have better results for future operations previously the files we saved from kinect v2 were filtered only in z direction but now the team filtered the results by x, y and z direction to have better results then before.

2.7 Individual Skills Improvement

As we have two developming teams, So We every thursday we have a meeting where we discuss what is the progress of both team and with sharing of knowledge between the teams with training sessions by different group members in order to put everyone on same page with knowledge of what is our progress as a group and understanding of the code.

2.8 Management of the Project

Project is managed by three Managers which includes Wajahat as main manager monitoring the progress of the Project with handling of github and Reports, Marc and leo as sub-managers monitoring and head of main developing team, As we have divided our team into developing teams , each team having six members.Things are going good we will try to illustrate our results in much better way next week report.

3 Future Work

- Appplying Meshes and Finishing ICP
- Enhanced GUI with Streaming
- Optimizing and Combining Code
- Streaming through GUI Qt with advancement
- Studying more algorithm for 3D reconstruction

Github link : https://github.com/WajahatAkhtar/Project-S.E.