Computer Vision 2 - Assignment 3
3D Mesh Generation and Texturing

Monday 7th May, 2018

Deadline: 01-06-2018, 23:59:59 (Amsterdam time)

General guidelines

Students should work on the assignments in **groups of 2-3**. Students are supposed to work on this assignment for three weeks. Some minor additions and changes might be done during these three weeks. Students will be informed for these changes via Piazza and Blackboard.
Any questions regarding to the assignment content can be discussed on Piazza `https://piazza.com/class/jf0xj72ryz96k4` (Access code: 52042cov6y).
For this assignment, students are expected to do in C++, however you can make use of the existing libraries (e.g. `http://pointclouds.org/`). You **are not expected** to implement algorithms on your own. However, you are expected to verify your choices of algorithms. For instance, you should explain why do you select Poisson algorithm over Marching Cube. Please provide requirements.txt with all external dependencies listed and README.txt file with description about how to reproduce results.
Source code and report must be handed in together in a zip file (ID1_lastname1_ID2_lastname2_ID3_lastname3.zip) before the deadline by sending to Dropbox `https://www.dropbox.com/request/RhlKFWOKBldezKCTRQ7F`. For full credit, make sure your report follows these guidelines:

- Include an introduction and a conclusion to your report.
- The maximum number of pages is 10 (single-column, including tables and figures). Please express your thoughts concisely. The number of words does not necessarily correlate with how well you understand the concepts.
- Answer all given questions. Briefly describe what you implemented.
- Try to understand the problem as much as you can. When answering a question, give evidences (qualitative and/or quantitative results, references to papers, etc.) to support your arguments.
- Analyze your results and discuss them, e.g. why algorithm A works better than algorithm B in a certain problem.
- Tables and figures must be accompanied by a brief description. Do not forget to add a number, a title, and if applicable name and unit of variables in a table, name and unit of axes and legends in a figure.

Late submissions are not allowed. Assignments that are submitted after the strict deadline will not be graded. In case of submission conflicts, TAs' system clock is taken as reference. We strongly recommend submitting well in advance, to avoid last minute system failure issues.
Plagiarism note: Keep in mind that plagiarism (submitted materials which are not your work) is a serious crime and any misconduct shall be punished with the university regulations.
Award: The team with highest assignment grades will be 3D scanned and printed. The winner will be announced at the end of this course.

1 Preparation

1.1 Data

Data is provided in the 3dframes directory. It contains RGB, depth images together with estimated camera pose. It's accompanied by sample code. Please check *.cpp files for more details. main.cpp contains TODO sections which are required to filled in.

1.2 Configuring

Following instruction we provide for GNU/Linux operating system. To build the sample code following dependencies are required (sorted in order):

- CMake
- Eigen 3
- OpenCV $\geq$ 3.0
- VTK = 6.3 (`https://www.vtk.org/files/release/6.3/VTK-6.3.0.zip`)
- PCL = 1.7.2 (`https://github.com/PointCloudLibrary/pcl/releases/tag/pcl-1.7.2`)

Make sure OpenCV, VTK, PCL are all linked with Qt4.

1.3 Building

```
$ cd source_directory
$ mkdir build && cd build
$ cmake ../
$ make -j2
```

After running

```
$ ./final ../3dframes 0 t
```

you should expect a mesh visualizer to appear:

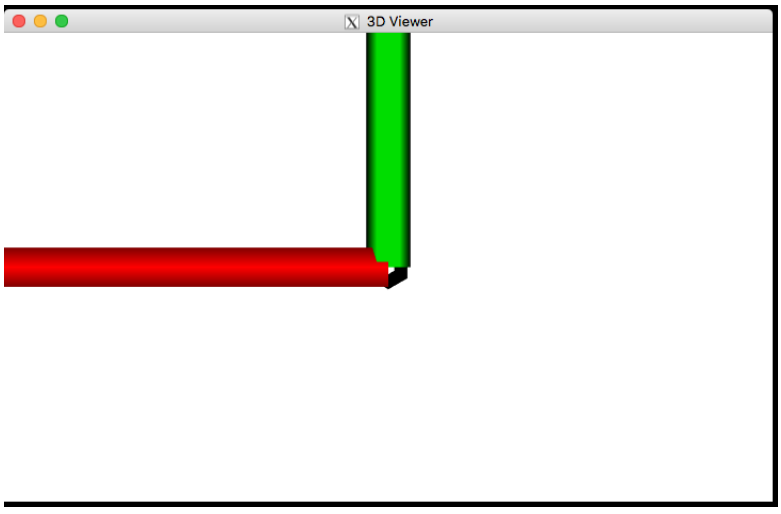

Figure 1: 3D mesh visualizer

1.4

In the case of reproduction with non success, TAs have one single machine with environment configured which can be accessed after **14 May**. Please contact Minh Ngo `mailto:l.m.ngo@uva.nl` for user credentials.

2 Depth-based and Texture-based 3D Reconstruction Comparison (5 points)

In the first three weeks of the Computer Vision 2 course you have implemented a depth based 3D reconstruction method (i.e. ICP). The following 2 weeks you have implemented a texture based 3D reconstruction method (i.e. SFM).

- Please briefly explain what the **advantages** and **disadvantages** of these methods are.
- Do you think these two methods could be used together to get better 3D reconstruction?

3 3D Meshing and Watertighting (20 points)

In this assignment, you are expected to generate 3D mesh from registered point clouds. You will work with the provided depth and color images, and camera poses. The sample data is in ".3df" format, you will use the provided C++ source code to read this data. To merge point clouds into a single one, you should follow the steps as described by algorithm 1:

Algorithm 1 Merging

```
procedure MergingPointclouds(3DFrames)
    model_point_cloud ← emptyPointCloud()
    for frame in 3DFrames do
        depth_image ← frame.depth_image
        focal_length ← frame.focal_length
        camera_pose ← frame.camera_pose
        point_cloud ← depthToPointCloud(depth_image, focal_length)
        point_cloud_with_normals ← computeNormals(point_cloud)
        point_cloud_with_normals ← transformPointCloud(point_cloud_with_normals, camera_pose)
        model_point_cloud ← concatPointClouds(model_point_cloud, point_cloud_with_normals)
    end for
    return model_point_cloud                          ▷ Contains all points XYZ and normals
end procedure
```

To get the final 3D mesh, you need to pass *model_point_cloud* to a mesh generation method (i.e. **Poisson Surface Reconstruction** [Kazhdan et al., 2006], **marching cube** [Lorensen and Cline, 1987]). You will get a 3D mesh. This 3D model may have holes on the part where there is no camera view. You are expected to fill the holes.

In the report please include 3D models from different view points. Please indicate what the advantages and disadvantages of mesh generation methods are and how do they reflect on 3D mesh results. Play with algorithm hyper parameters and and analyse and influence of them on final results.

4 Coloring 3D Model (15 points)

The 3D model that you have generated in section 3 is not colored yet. To this end, you should follow the steps as described by algorithm 2. In the report please include an obtained colored 3D model from different view points.

5 Additional references

- `http://www.pointclouds.org/documentation/`

Algorithm 2 Texturing

```
1: procedure TEXTURE(mesh, 3DFrames)
2:     polygons ← mesh.polygons
3:     point_cloud ← mesh.point_cloud
4:     for frame in 3DFrames do
5:         depth_image ← frame.depth_image
6:         focal_length ← frame.focal_length
7:         camera_pose ← frame.camera_pose
8:         transformed_point_cloud ← transformPointCloud(point_cloud, camera_pose.inverse())
9:         for polygon in polygons do
10:            if polygon visible to this camera then
11:                uv_coordinates ← getUVCoordinates(polygon, transformed_point_cloud)
12:                assign uv_coordinates of this camera to the polygon
13:            end if
14:        end for
15:    end for
16: end procedure
```

References

M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 3-905673-36-3. URL `http://hhoppe.com/poissonrecon.pdf`.

W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM. ISBN 0-89791-227-6. doi: 10.1145/37401.37422. URL `http://academy.cba.mit.edu/classes/scanning_printing/MarchingCubes.pdf`.