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

Monday 15th May, 2017

Students are supposed to work on this assignment for three weeks in **groups of two**. Some additions and changes might be done during this period. Students will be informed for these changes via blackboard. The **analysis** and the **conclusion** must be included in a report. A final report and source code should be zipped and uploaded to Dropbox `https://www.dropbox.com/request/OzyCe5344CypwlJBGQ3f` not later than 06-06-2017, 23:59:59 (Amsterdam Time). The submitted zip file should be named with authors' surname.

Please keep a report short and concise (about 8 pages max), preferably in the research paper format. This assignment requires some knowledge in C++. For this assignment, you can make use of the existing libraries (e.g. `http://pointclouds.org/`), you **are not expected** to implement these functions 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 use login "user" and password "computer-vision2" for authorization. To build the starter project do the following in terminal:

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

For people with a little experience in C++ a VirtualBox (`https://www.virtualbox.org/wiki/Downloads`) Linux environment with all configured dependencies is provided (`https://drive.google.com/uc?id=0B_f8lHMYsLosVnlmR1QwdUxIMms&export=download`).

Pre-installed editors (QtCreator, vim) can be used for writing code. If vim has been accidentally opened, type ":q". Root permission is provided using the same password if it's required to install additional packages.

1 Depth-based and Texture-based 3D Reconstruction Comparison (5 pt)

In the first three weeks of the computer vision you have implemented a depth based 3D reconstruction method (i.e. ICP). The following three 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 (Please comment on this question)?

2 3D Meshing and Watertighting (20 pt)

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 ".3ds" 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 will have holes on the part where there is no camera view. You are expected to fill the holes by applying watertight.

In the report please include 3D models from different view points before and after watertighting. Please indicate what the advantages and disadvantages of mesh generation methods are and how do they reflect on 3D mesh results.

3 Coloring 3D Model (15pt)

The 3D model that you have generated in section 2 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.

Algorithm 2 Texturing

```
procedure TEXTURE($mesh, 3DFrames$)
    polygons ← mesh.polygons
    point_cloud ← mesh.point_cloud
    for frame in 3DFrames do
        depth_image ← frame.depth_image
        focal_length ← frame.focal_length
        camera_pose ← frame.camera_pose
        transformed_point_cloud ← transformPointCloud(point_cloud, camera_pose.inverse())
        for polygon in polygons do
            if polygon visible to this camera then
                uv_coordinates ← getUVCoordinates(polygon, transformed_point_cloud)
                assign uv_coordinates of this camera to the polygon
            end if
        end for
    end for
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`.