**HuweiMediaLab 3D**

There are two basic post processes to convert the point cloud into a form readable by the application:

1. Generate mesh from the point cloud. We use meshlab

2. Move the origin of the object to its center of mass. We use Blender.

The reason why step 1 is important is obvious(we want to render meshes).

For step 2, it happens that the generatd mesh does not have its origin at the center of mass. As a result, any rotation would be relative to the local object oigin which gives unintended results. Basically we would be rotating about some point set by matlab as origin.

**Post process 1**

1. Launch MeshLab (already installed on development system). To install, use google search as online location may change

2. Drag and drop the generated PLY file into MeshLab

3. Follow the steps below(if confused, refer to video in link below). It is highly recommended you follow the video ( <https://www.youtube.com/watch?v=6wP_e37t7PI>).

- Compute normals ( Filters -> Normals, curvatures and orientation -> Compute normals for point sets)

- Generate mesh (Filters -> Remeshing, Simplification and Reconstruction -> Surface Reconstruction )

-obtain texture (Filters -> Texture -> Parametrization: Trivial Per-Triangle)

-Save Project(use any names you like, you will be saving the ply aswell as a new mlp project)

-Transfer color to texture(Refer to video)

-Export Mesh as obj. This will generate 2 files(with obj and mtl extensions)

Ref: <https://www.youtube.com/watch?v=6wP_e37t7PI>

**Post process 2**

1. Launch Blender

2., Right click on the cube and hit delete, a pop should appear for confirmation. Hit the Enter Key

3. Import newly generated mesh(file->import->wavefront(.obj) , select the mesh and finaly hit the "import obj" buton on the top right corner )



3. Right click on the object to ensure its selected(Notice orange outline along its edges)

4. In blender , hit "ctrl + alt +shift + c ", on the keyboard select move origin to center of mass

5. Now, with the object origin at the center of mass, move the object to the center.

-Hit G, this activates object translation.

-To lock the translation on one axis, hit the letter for that axis, e.g X, Y or Z

For example, to move along the Y, I hit G, then when tranlation is active, I hit Y. Now

the object follows the cursor.

-You can also rotate if you wish. This time hit 'R'

6. Finally export mesh as obj again. This is the mesh to be transferred to the phone.

**Importing into the mobile for viewing**

1. Copy these files to C:\Users\samuel\projects\mediaLab3D\app\src\main\res\raw

i.e The raw folder of the 3D project.

2. Rename this file by changing "." to "\_" . E.g face.obj should be changed to face\_obj and face.mtl should be changed to face\_mtl.

3. Copy the generated texture in process 1 above to drawable-nodpi folder

4. Open The Renderer class in the project

5. Look for the code below:

**objParser** = **new** LoaderOBJ(**mContext**.getResources(),  
 **mTextureManager**, R.raw.***facesmall\_obj***);

Change **facesmall\_obj** to the new obj file name

6. Look for getCurrentCamera().setZ(5.0);

Edit this value until the object is visible. Sometimes the camera may be too far, you need to reduce the value. Sometimes, it may be too close, you need to increase the value.

If the object seems upside down, or layed out in the wrong direction, look for

getCurrentCamera().setUpAxis(-1.0f, 0.0f, 0.0f);

Edit these values unti desired result is obtained. (-1.0f, 0.0f, 0.0f) implies that the camera's up axis is pointed upside down. This final value depends on the orientation of the object model.

emial: gadesoye@ucsc.edu