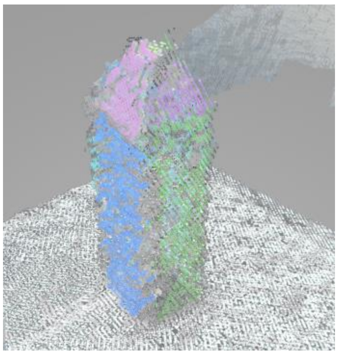
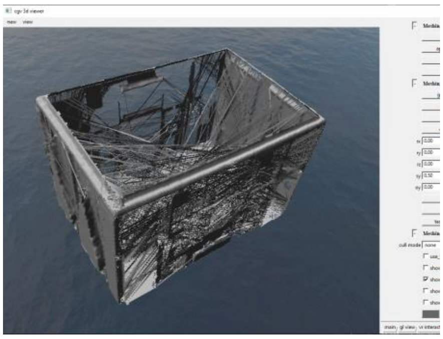
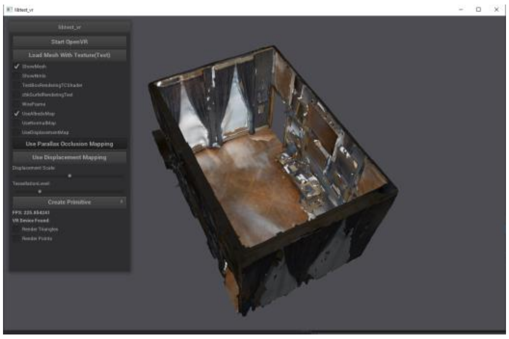
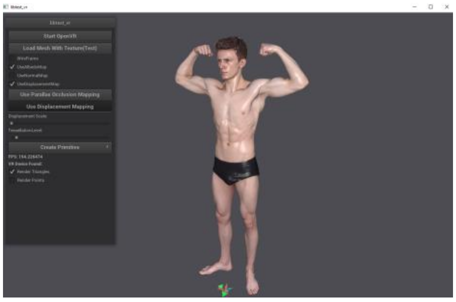
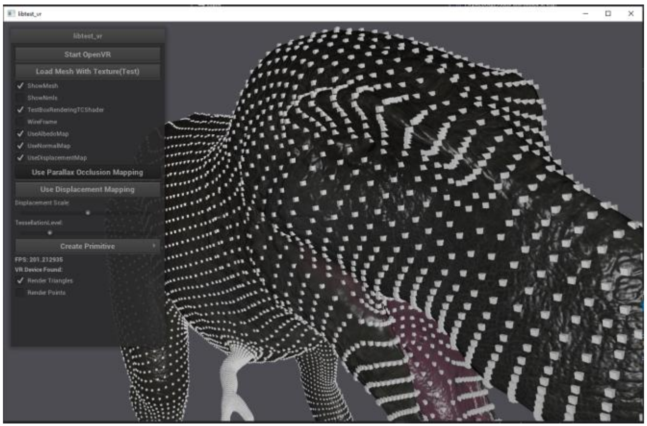
report: 07/03/2021

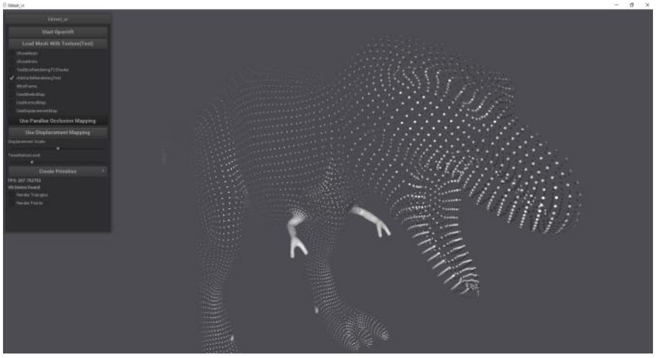
- Fit to low poly model
    - Last week I clicked feature points manually, which is found to be inaccurate. To solve this, we have to select faces first, then, do region growing to determine boundaries.
    - I've made the manual selection of the points possible. Give start seeds for the region growing algorithm at the same time.
    - 
- Features of a point cloud
    - Together with Tianfang, we filtered the normal of the point clouds as preparation for algorithms to be implemented.
- Surface Reconstruction:
    - In this week, I compiled the code https://github.com/NSchertler/OnlineSurfaceReconstruction for our APB model reconstruction. It is an improved version for the mesh cleaning method used in this paper: "https://github.com/facebookresearch/Replica-Dataset".
    - But I think It's better to clean point clouds first, then feed the cleaned point cloud to this reconstruction algorithm.
    - Implemented a better surface renderer with a custom framework: with opensource projects like nanogui/ nse/ tinyobjloader...
    - The load and render of obj files is relatively better than the fw I used before.
    -  

    - The simple_mesh loader in the cgv kept the simplicity of the code but is not robust enough for our future research, so I tried to improve this.
    - The picture on the left side shows that the model corrupts while It can be corrected loaded with my new framework.
    - The loading of the mesh geometry was done with tinyobjloader https://github.com/tinyobjloader/tinyobjloader
    - When compared to Unity, we have a full controlment over the whole modeling and rendering process. Which gives us maximum flexibility for our research.
    - Here is a brief introduction:
    - Model loading Capability:

        - Capable of robust loading/ rendering obj models
        - Capable of loading of hdr textures, thanks to https://github.com/NSchertler/CG1
        - 

          □ WireFrame
          ☑ UseAlbedoMap
          □ UseNormalMap
          ☑ UseDisplacementMap
          Displacement Scale
          TessellationLevel
          FPS: 194.226474
          VR Device Found:
          ☑ Render Triangles
          □ Render Points
        - That means, we are able to render high quality textures later on.
    - Rendering Capability:
        - In this week, I also implemented renderers for rendering boxes and surfels which will be used for point cloud rendering in this framework.
        - As for the renderers, I've taken a look at the cgv framework as reference.
        - 

          ☑ ShowMesh
          □ ShowNmls
          ☑ TestBoxRenderingTCShader
          □ WireFrame
          ☑ UseAlbedoMap
          ☑ UseNormalMap
          ☑ UseDisplacementMap
          Displacement Scale
          TessellationLevel
          FPS: 201.212935
          VR Device Found:
          ☑ Render Triangles
          □ Render Points

          □ ShowMesh
          □ ShowNmls
          □ TestBoxRenderingTCShader
          ☑ [illegible]
          □ WireFrame
          □ UseAlbedoMap
          □ UseNormalMap
          □ UseDisplacementMap
          Displacement Scale
          TessellationLevel
          FPS: 207.752753
          VR Device Found:
          □ Render Triangles
          □ Render Points
        - VR rendering is enabled with OpenVR and OpenXR. Currently, OpenVR integration is tested on my device and ready to use.
- report: 14/03/2021