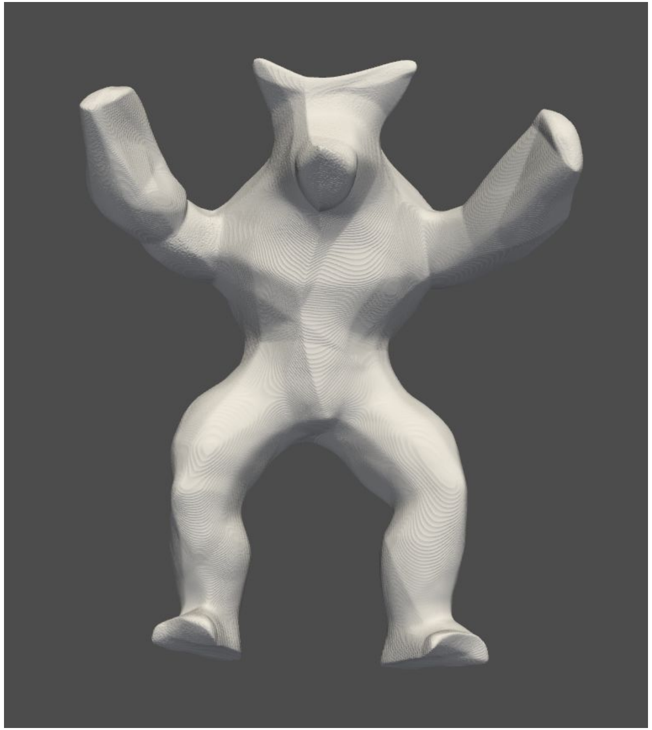
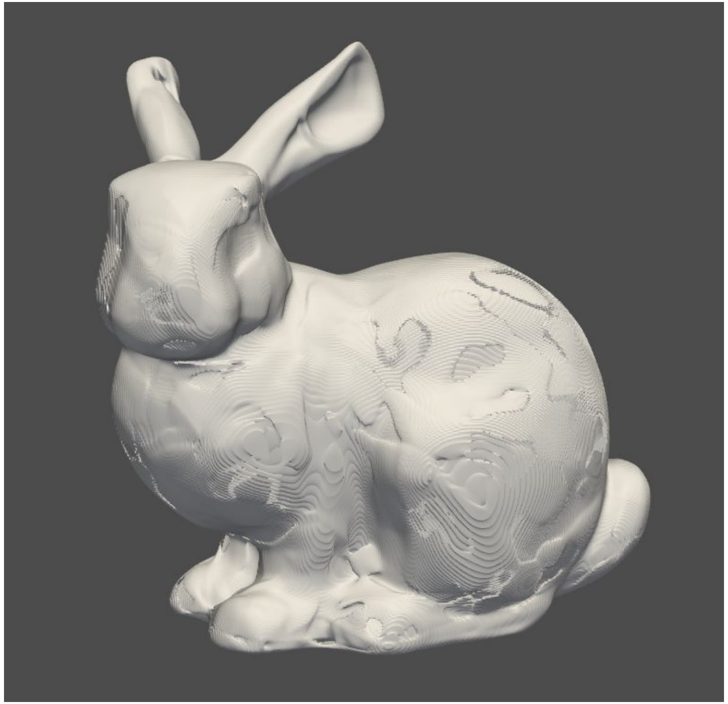
# PHASE: Implementation

Manoj S - Intern Task

# Results

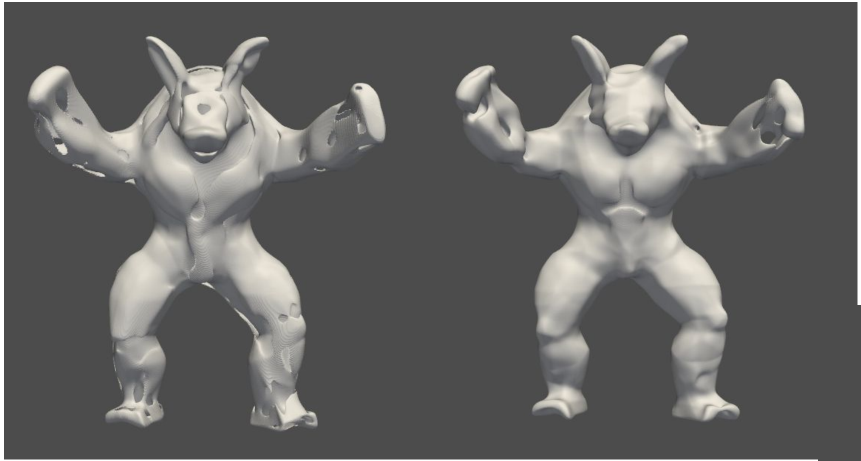


*Mesh generated from armadillo_normals_50000*



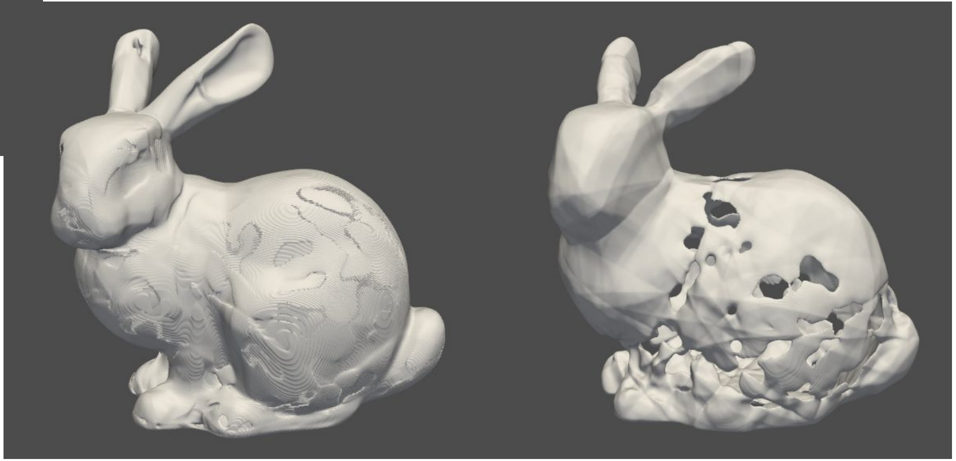
*Mesh generated from bunny_10000*

# Results: Variation with NumPoints

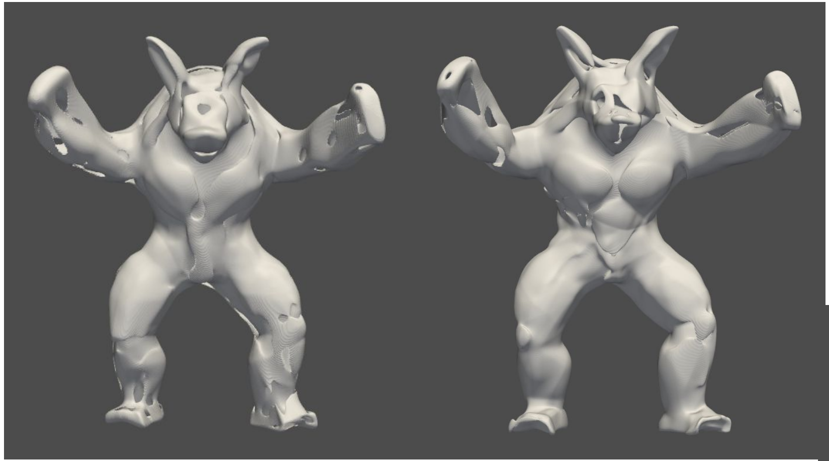


*Armadillo: 10000 vs 50000 Numpoints (Left and Right Respectively)*

*Bunny: 10000 vs 50000 Numpoints (Left and Right Respectively)*

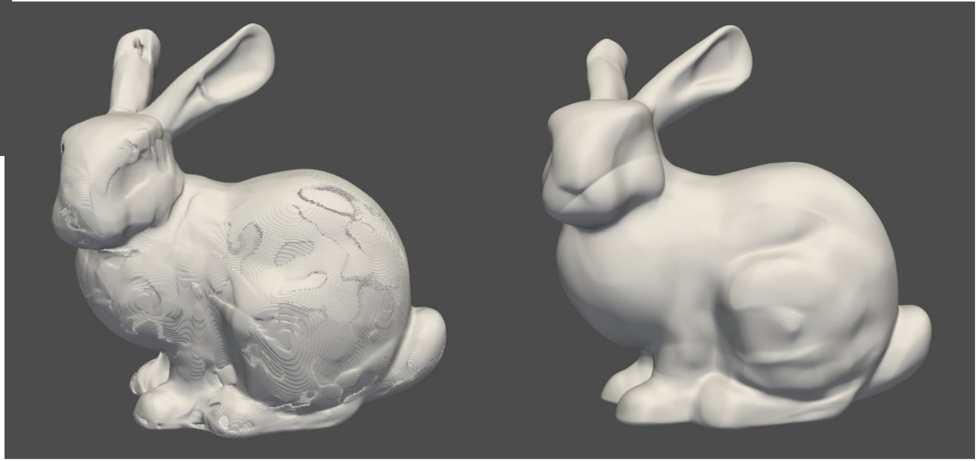


# Results: Variation with Normal

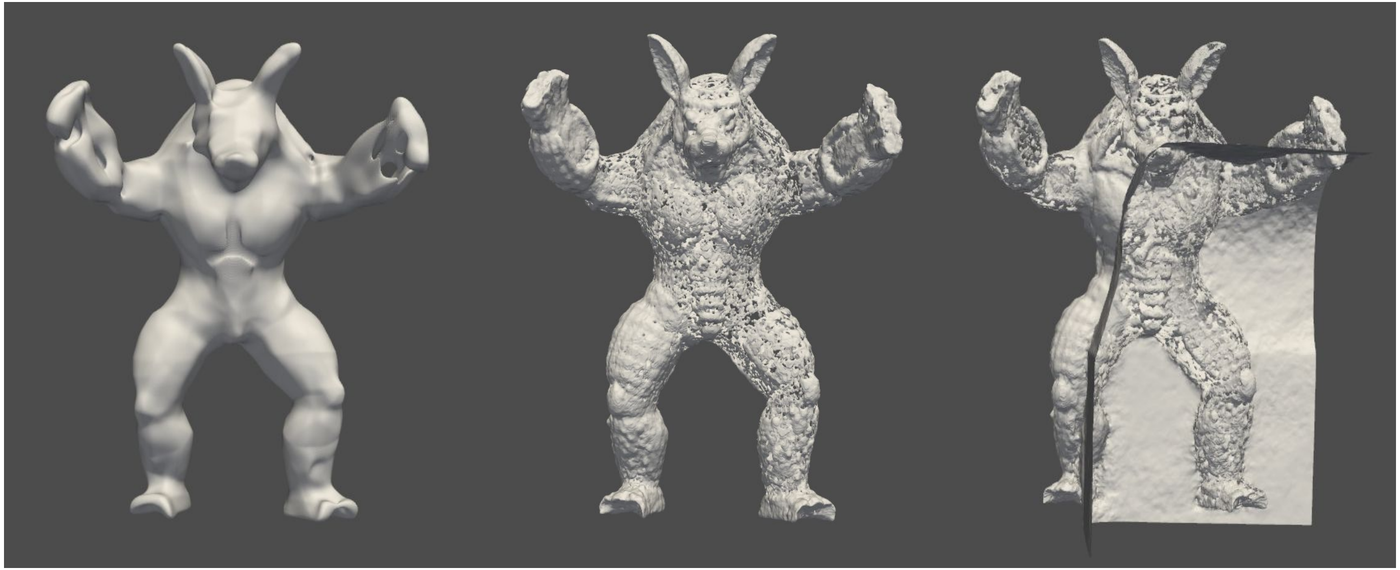


*Armadillo: 10000 Numpoints (Left without Normal and Right with Normal)*

*Bunny: 10000 Numpoints (Left without Normal and Right with Normal)*



# Results: Variation with Fourier

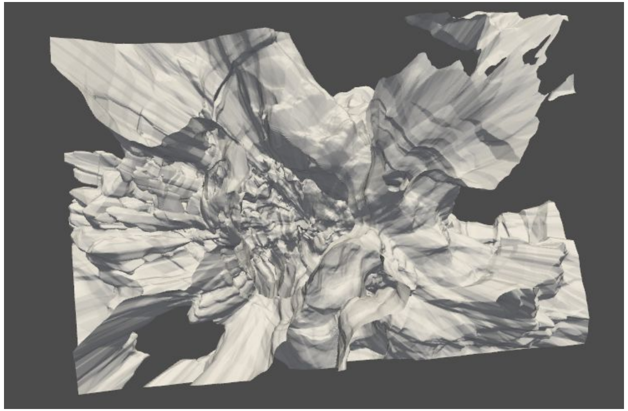


*Armadillo: 50000 Numpoints (Left without Fourier, Middle with Fourier and Right with Fourier and Normal)*
*Details like eyes, nose, inner ear cavity, etc are captured in Fourier (middle)*

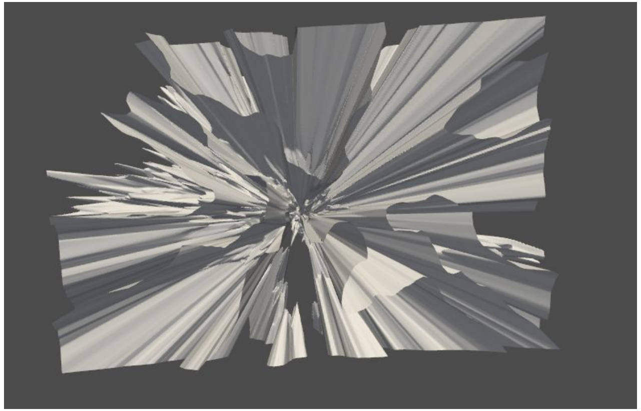
# Reconstruction Quality (Chamfer Distance)

| Object | Num Points | Aspects | Chamfer Distance |
|---|---|---|---|
| Armadillo | 10000 | - | 358.787 |
| | 50000 | - | 182.179 |
| Bunny | 10000 | - | 178.644 |
| | 50000 | - | 3407.222 |
| Armadillo | 10000 | With Normals | 327.193 |
| | 50000 | With Normals | 196.260 |
| Armadillo | 50000 | With Fourier | 510.932 |
| Armadillo | 50000 | With Normals, Fourier | 21230.71 |
| Bunny | 10000 | With Normals | 65.358 |
| Bunny | 10000 | With Fourier | 33010.18 |

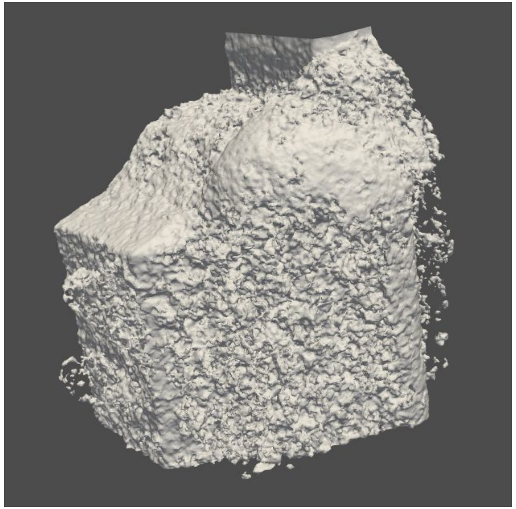
# Failed Generations: Dragon



*Dragon 20000 Num Points*



*Dragon 100000 Num Points*



*Bunny 10000 with Fourier*

# Observations and Challenges

- The 3D mesh generation is **not working good for bigger point clouds** like that of the dragon. The training process seems to be unstable for larger point cloud images.
- Adding a Fourier encoding is able to **capture more details** when compared to that of no encoding (details like eyes, nose, ear cavities can be observed in Armadillo). But the training of fourier model seems is **unstable** (In the case of bunny and armadillo with normals).
- The point cloud which has higher coordinate data seems to be giving better 3D mesh (observed in armadillo Numpoints 10000 vs 50000).
- Point cloud data with normals give the model additional information, thus giving a better rendering. (observed in the case of bunny).

# Further Steps

- Adding Normalization to Input Data:
  - The training process can be noticed to be unstable, especially for bigger point cloud images.
  - The coordinate values (xyz) belong to a very large range for bigger point clouds. The bigger variance among the coordinate values might be the cause of the unstable training process.
  - The coordinate values could be normalized to bring it to a smaller range (like 0 to 1). This reduces the variance while training making it stable.
- Increasing the points per batch:
  - Due to limit compute capability, the points supplied per batch had to be reduced. This reduces the capability of the model to generalize properly over all points in the point cloud.
  - Thus increasing the points per batch would definitely increase the performance of the model
  - PS: The points per batch supposed to be used is 15000, while that we used is 3000.