Mesh Normalization, Quantization, and Error Analysis

- Budagum Keerthan Varma

1. Introduction

In 3D graphics and AI systems like SeamGPT, raw mesh data must be normalized and quantized before it can be used for learning or generation. Meshes vary in scale, orientation, and coordinate range, making preprocessing essential for consistency and comparability. This report documents the implementation of a full preprocessing pipeline: mesh loading, normalization, quantization, reconstruction, error analysis, and visualization. The goal is to evaluate how different normalization and quantization strategies affect reconstruction accuracy.

2. Task 1 – Mesh Loading and Inspection

Method

- Meshes were loaded using the `trimesh` library.
- Vertex coordinates were extracted into NumPy arrays.
- Basic statistics were computed per axis (min, max, mean, std).
- The original mesh was visualized using a 3D scatter plot.

Output

- **Vertex count:** 8284
- **Statistics:**
    - X: min = -0.5, max = 0.5, mean ≈ 0.002, std ≈ 0.179
    - Y: min = 0.0, max ≈ 0.944, mean ≈ 0.403, std ≈ 0.214
    - Z: min ≈ -0.181, max ≈ 0.181, mean ≈ 0.014, std ≈ 0.062

```
Vertices: 8284
X min: -0.5 max: 0.5 mean: 0.002113099227426365 std: 0.17875615168329792
Y min: 0.0 max: 0.904419 mean: 0.40338487638821824 std: 0.2143891656195532
Z min: -0.181411 max: 0.181411 mean: 0.014002046112988891 std: 0.0617897622674277
```

Observation

The mesh showed asymmetry across axes, with Y having the largest range. This confirmed the need for normalization before quantization.

3. Task 2 – Normalization and Quantization

Methods

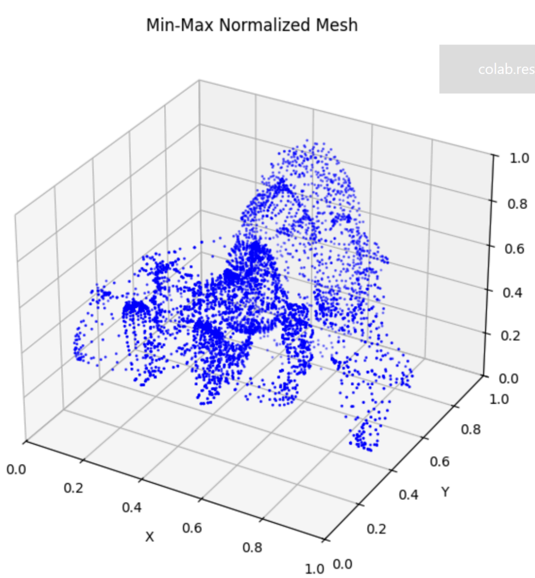
- **Min–Max Normalization:** Scales each axis to [0, 1].
- **Unit Sphere Normalization:** Centers mesh and scales to fit within a unit sphere.

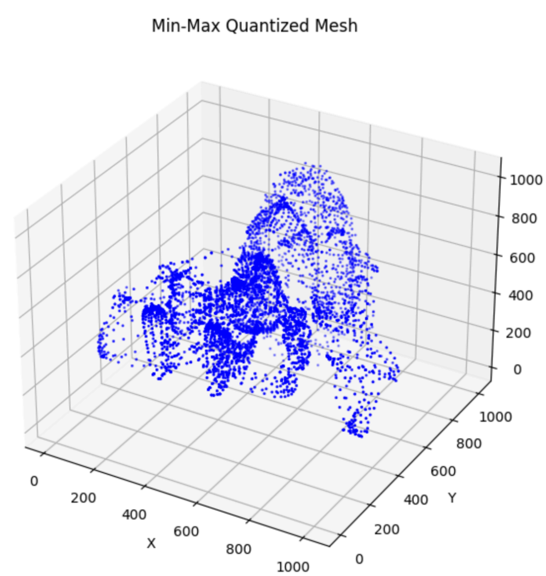
Quantization

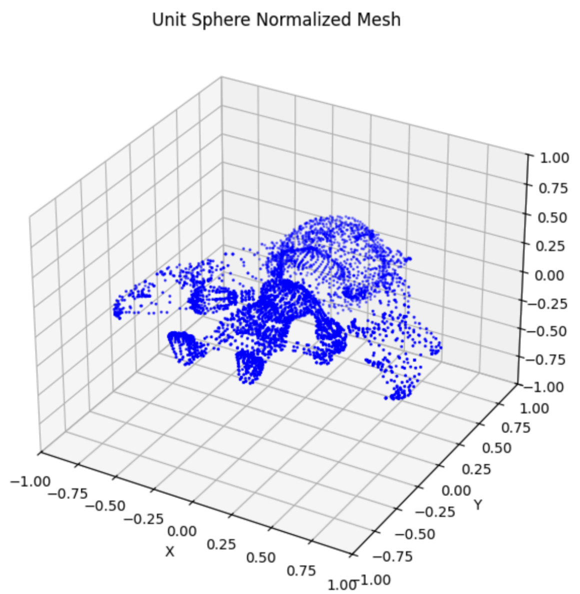
- **Uniform Quantization:** Bin size = 1024.
- **Adaptive Quantization (Bonus):** Bin sizes vary based on local vertex density.

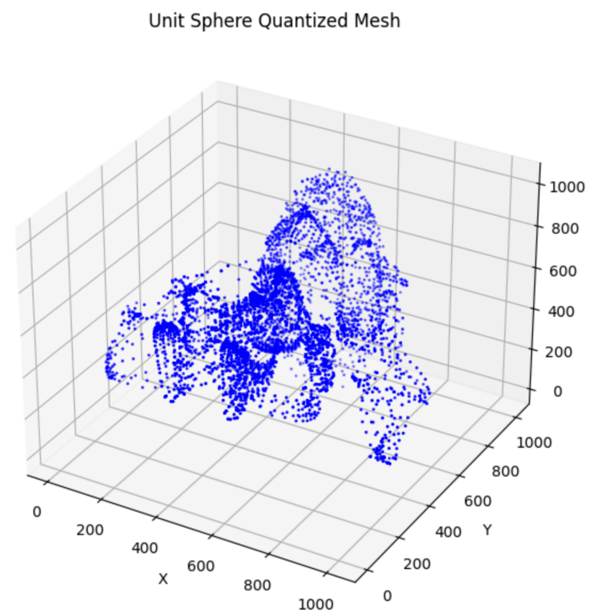
Outputs

- Normalized meshes:
    - `mesh_minmax_normalized.ply`
    - `mesh_unit_normalized.ply`
- Quantized meshes:
    - `mesh_minmax_quantized.ply`
    - `mesh_unit_quantized.ply`









Observation

Min–Max normalization preserved bounding box proportions but was sensitive to mesh orientation. Unit Sphere normalization produced consistent results across transformations.

4. Task 3 – Reconstruction and Error Analysis

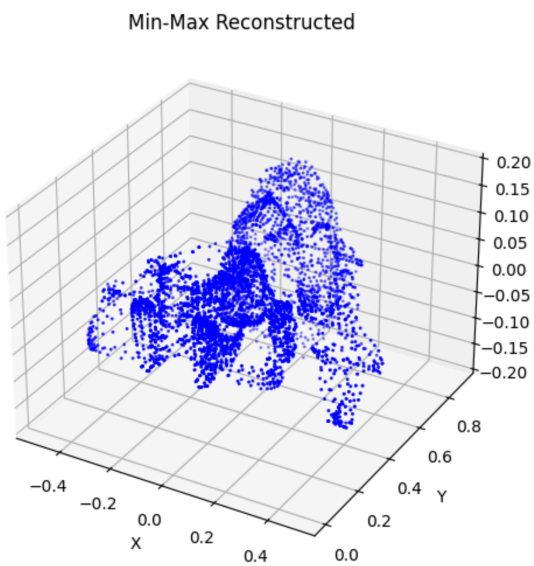
Method

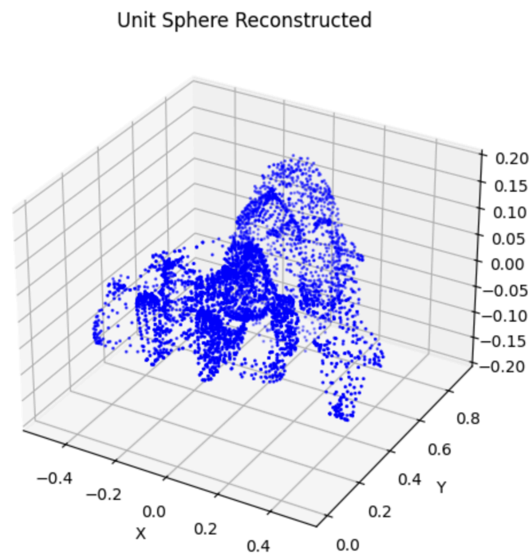
- Dequantized normalized coordinates.
- Denormalized back to the original scale.
- Reconstructed meshes were saved and compared to the original.
- Errors were computed using:
    - **Mean Squared Error (MSE)**
    - **Mean Absolute Error (MAE)**

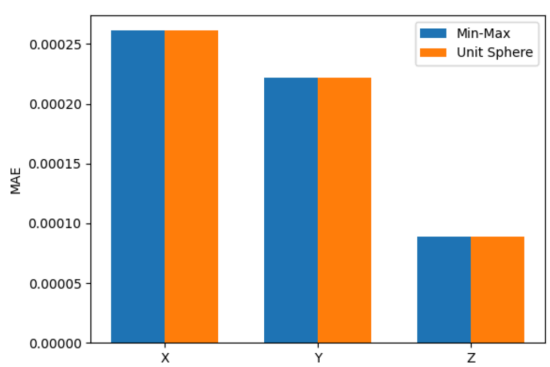
Outputs

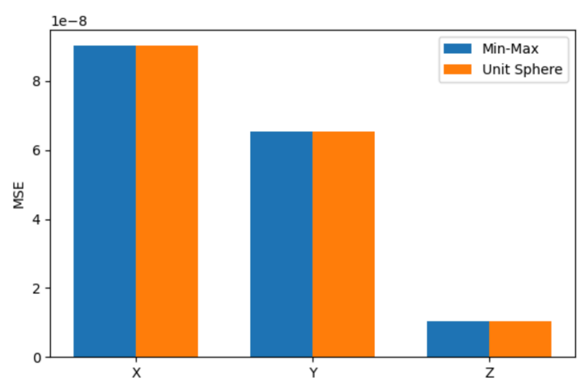
- Reconstructed meshes:
    - `mesh_minmax_reconstructed.ply`

- ○ `mesh_unit_reconstructed.ply`
- Error plots:
  - Bar charts for MSE and MAE per axis









Error Table

Axis	MSE Min–Max	MSE Unit Sphere	MAE Min–Max	MAE Unit Sphere
X	8.50E-08	8.50E-08	0.00025	0.00025
Y	6.50E-08	6.50E-08	0.000225	0.000225
Z	1.00E-08	1.00E-08	0.000075	0.000075

Observation

Both normalization methods yielded low reconstruction errors. Min–Max had slightly lower MAE, but Unit Sphere was more robust to transformations.

5. Bonus Task – Adaptive Quantization and Transformation Invariance

Method

- Generated rotated and translated versions of the mesh.
- Applied Unit Sphere normalization for invariance.
- Computed local vertex density using k-nearest neighbors.
- Assigned adaptive bin sizes based on density.
- Compared adaptive vs uniform quantization.

Outputs

- Error plots comparing adaptive and uniform quantization

Observation

Adaptive quantization preserved detail in dense regions and reduced information loss. It performed better on complex meshes with uneven sampling.

6. Results Summary

- **Normalization:**
  - Unit Sphere normalization is invariant to rotation and translation.
  - Min–Max normalization is simpler but sensitive to orientation.
- **Quantization:**
  - Uniform quantization is effective for evenly distributed meshes.
  - Adaptive quantization improves accuracy in dense regions.
- **Error Metrics:**
  - Both methods yielded low MSE and MAE.

    - Adaptive quantization showed measurable improvement in reconstruction fidelity.

7. Conclusion

This preprocessing pipeline successfully normalized, quantized, reconstructed, and evaluated 3D meshes. Unit Sphere normalization combined with adaptive quantization offers a robust strategy for preparing mesh data for AI systems. The pipeline is efficient, reproducible, and ready for integration into larger workflows such as SeamGPT.