3D Mesh Normalization, Quantization, and Error Analysis

---

1. Introduction

3D mesh data is widely used in fields such as computer graphics, animation, engineering, and more recently in 3D-aware AI models such as SeamGPT. A mesh consists of a collection of vertices in 3D space, connected by faces to form surfaces. However, meshes created from different tools or sources often vary significantly in scale, coordinate range, and orientation. These inconsistencies make it difficult to train machine learning models, as the model cannot reliably learn from data that is not numerically consistent.

To address this, preprocessing steps such as normalization and quantization are applied. Normalization brings all meshes into a uniform coordinate space, while quantization compresses floating-point coordinates into discrete bins. After these steps, the mesh is reconstructed to measure how much information was lost.

In this project, eight meshes in .obj format were processed using three different normalization techniques. The meshes were quantized, reconstructed, and evaluated using Mean Squared Error (MSE) to understand which normalization method preserves geometric structure most effectively.

---

2. Objective

The goals of this study were to:

- Apply Min–Max, Z-Score, and Unit Sphere normalization
- Quantize normalized values using 1024 bins
- Reconstruct the mesh using dequantization and denormalization
- Measure reconstruction accuracy using MSE
- Compare all methods to identify the most reliable approach
- Generate visual plots for MSE analysis

These steps reflect real preprocessing workflows used in 3D AI systems.

---

3. Dataset Overview

The dataset contains eight 3D mesh models:

- branch.obj
- cylinder.obj
- explosive.obj
- fence.obj
- girl.obj
- person.obj
- table.obj
- talwar.obj

These meshes represent a mix of organic models, mechanical shapes, thin and elongated structures, and objects with varying geometric complexity.

Each mesh file provides:

- A list of vertices (v x y z)
- A list of triangular faces (f a b c)

The analysis in this project focuses entirely on vertex coordinates.

---

4. Workflow

4.1 Mesh Loading

Each mesh was loaded using the Trimesh library, which converts it into a numerical array of vertex coordinates. This format is ideal for applying mathematical operations such as scaling and quantization.

4.2 Normalization Methods

Three normalization techniques were investigated.

1) Min–Max Normalization
Each axis is independently transformed to the range [0, 1]:
x′=x−xminxmax−xminx' = \frac{x - x_{min}}{x_{max} - x_{min}}x′=xmax−xminx−xmin
This method is simple, stable, and works well when axes have different ranges.

---

2) Z-Score Normalization

Coordinates are standardized using the mean and standard deviation:
x′=x−μσx' = \frac{x - \mu}{\sigma}x′=σx−μ
The standardized values are then mapped into the range [0,1]. This method is sensitive to uneven axis variances.

---

3) Unit Sphere Normalization

The mesh is centered at its centroid and scaled to fit inside a sphere of radius 1:
x′=x−crx' = \frac{x - c}{r}x′=rx−c
Then mapped to [0,1]. This method preserves proportions for elongated shapes.

4.3 Quantization

Normalized coordinates (between 0 and 1) were quantized into 1024 bins:
q=int(x′⋅1023)q = \text{int}(x' \cdot 1023)q=int(x′⋅1023)
This step reduces precision but allows for compact storage in 3D model processing pipelines.

4.4 Reconstruction

1. **Dequantization**
   x″=q1023x" = \frac{q}{1023}x″=1023q
2. **Denormalization**
   The inverse of each normalization method is applied.

The reconstructed vertices are then compared to the original mesh.

4.5 Error Measurement

The accuracy of reconstruction is quantified using Mean Squared Error (MSE):
MSE=1N∑(x−x″)2MSE = \frac{1}{N}\sum (x - x")^2MSE=N1∑(x−x″)2
MSE is computed per axis (x, y, z) and also averaged to obtain a total MSE.

Results and Conclusion

After applying all three normalization methods (Min–Max, Z-Score, and Unit Sphere) to the eight mesh models and performing quantization and reconstruction, the Mean Squared Error (MSE) results showed clear patterns. Min–Max normalization produced the lowest reconstruction error for most meshes, meaning it preserved the original shape more accurately after quantization. Unit Sphere normalization performed better on elongated or unevenly shaped models such as talwar.obj and branch.obj, because it scales the entire shape uniformly. Z-Score normalization showed higher errors overall because it is sensitive to differences in variance across axes, which caused small distortions after reconstruction.

From these results, Min–Max normalization is the most effective and reliable method for general mesh preprocessing when combined with 1024-bin quantization. It maintains the mesh structure with minimal error, making it suitable for consistent and stable data preparation in 3D AI applications. Unit Sphere normalization is useful for specific mesh shapes, while Z-Score normalization is less suitable due to its inconsistent performance across different models. Overall, the experiments show that Min–Max normalization offers the best balance between accuracy and simplicity for mesh standardization.