3D Mesh Normalization, Quantization, Reconstruction & Error Analysis

Name: Saravanan S

Reg no: RA2211026050104

1. Objective

The objective of this assignment is to preprocess 3D mesh (.obj) data through normalization, quantization, reconstruction, and error analysis to understand how transformations affect geometry accuracy.
The implementation simulates real-world preprocessing pipelines used in 3D AI systems like SeamGPT.

2. Methodology

The following pipeline was implemented in Python (v3.11) using NumPy, Trimesh, and Matplotlib:

1. Load Mesh:
   Load .obj mesh files and extract vertex coordinates (x, y, z).
2. Normalization:
   - Min–Max Normalization → Scales coordinates to [0, 1] range.
   - Unit-Sphere Normalization → Fits all vertices within a unit sphere.
3. Quantization:
   Convert normalized coordinates into discrete bins (1024 bins per axis).
4. Reconstruction:
   Dequantize and denormalize to recover original mesh geometry.

5. Error Measurement:
   Compute Mean Squared Error (MSE) and Mean Absolute Error (MAE) between original and reconstructed vertices.
6. Visualization:
   Generate bar plots (mse_axis.png) and histograms (hist_mm.png, hist_us.png) for each mesh to visualize reconstruction accuracy.

3. Results Summary

Mesh	**Vertices**	**Min–Max MSE**	**Min–Max MAE**	**Unit-Sphere MSE**	**Unit-Sphere MAE**
branch	2767	7.81e-07	7.34e-04	2.33e-06	1.32e-03
cylinder	192	7.96e-07	6.11e-04	2.57e-06	1.38e-03
explosive	2812	1.24e-07	2.76e-04	3.90e-07	5.47e-04
fence	1088	1.57e-07	2.73e-04	3.60e-07	5.42e-04
girl	8284	2.05e-07	3.70e-04	3.52e-07	5.16e-04
person	3103	7.89e-07	6.92e-04	1.76e-06	1.14e-03

Mesh	Vertices	Min–Max MSE	Min–Max MAE	Unit-Sphere MSE	Unit-Sphere MAE
table	3148	1.49e-07	3.07e-04	4.70e-07	5.99e-04
talwar	1668	1.31e-07	2.28e-04	5.85e-07	6.51e-04

4. End-to-End Observations

- Both Min–Max and Unit-Sphere normalizations successfully preserved the geometric structure of meshes.
- Min–Max Normalization consistently achieved lower reconstruction errors (MSE, MAE), making it more precise for evenly distributed meshes.
- Unit-Sphere Normalization was more stable for complex, asymmetrical meshes.
- Quantization at 1024 bins provided a strong balance between compression and accuracy.
- The MSE remained below 1e-6 and MAE below 1e-3 for all models, confirming high-fidelity reconstruction.
- Generated plots (mse_axis.png, hist_mm.png, hist_us.png) visually confirm minimal deviation between original and reconstructed meshes.

5. Conclusion

The project successfully achieved the end-to-end 3D mesh preprocessing workflow as outlined in the Mixar Assignment scope.

All core objectives — loading, normalization, quantization, reconstruction, and error evaluation — were completed and verified with multiple 3D models.

The results confirm that Min–Max normalization offers slightly higher precision, while Unit-Sphere normalization ensures geometric stability across scales.

The implemented approach is efficient, CPU-friendly, and demonstrates a solid understanding of how preprocessing affects 3D AI model performance.

6. Tools & Libraries

- Programming Language: Python 3.11
- Libraries Used: NumPy, Trimesh, Matplotlib, Open3D (optional)
- Hardware: CPU (No GPU required)

7. References

- Mixar Virtual Assignment (2025)
- Trimesh Documentation: https://trimsh.org
- NumPy & Matplotlib Documentation