Performance Analysis of 3D Mesh Simplification Algorithms

Ahnaf An Nafee (G00154707)

1. Problem Statement

High-polygon 3D models create performance bottlenecks for real-time graphics applications. Mesh simplification algorithms are essential for reducing model complexity, but their performance is typically benchmarked on idealized data, which may not reflect real-world usage. This project addresses this gap by investigating the research question: *Do the performance and geometric fidelity of common mesh simplification algorithms differ significantly when applied to clean, well-structured CAD models versus noisy, user-generated models?* The goal is to provide a rigorous, quantitative answer through formal experimental design and statistical analysis.

2. Methodology

A two-factor, between-subjects experimental design will be employed. The methodology consists of the experimental setup, data collection procedure, and the statistical analysis plan.

Experimental Design

- **Independent Variables (Factors)**:
  - *Factor 1: Simplification Algorithm* (2 Levels): Quadric Edge Collapse Decimation (QEM) and Clustering Decimation.
  - *Factor 2: Mesh Type* (2 Levels): "Clean CAD" models from the **ModelNet40** dataset and "Organic/Scanned" models from the **Thingi10K** dataset.
- **Dependent Variables (Metrics)**:
  - *Geometric Fidelity*: Quantified by the **Two-Sided Hausdorff Distance** ($max(d(A, B), d(B, A))$).
  - *Computational Performance*: Measured by the **Wall-Clock Time** in seconds (averaged over 5 repetitions with warm-up).

Procedure and Tools

A preliminary data pre-processing step will ensure all models are correctly formatted. The main experiment will then be automated using a Python script that utilizes **MeshLab**'s Python library (`pymeshlab`) to apply each algorithm to a curated set of 30 models (15 from each dataset) and record the dependent variables. This ensures a reproducible and error-free data collection process.

Statistical Analysis Plan

The collected data will be analyzed using a formal hypothesis-testing framework with a significance level of $\alpha = 0.05$.

- **Hypotheses**: Two separate **Two-Way ANOVAs** will test the null hypotheses for the main effects of algorithm and mesh type, and for their interaction effect. The interaction effect ($H_{0,int}$) is of primary interest, as rejecting it would provide evidence that the optimal algorithm choice depends on the model's quality.
- **Analysis Pipeline**:
  1. **Descriptive Statistics**: Calculate mean, standard deviation, standard error, and **95% confidence intervals** for each of the 4 experimental groups.
  2. **Assumption Checks**: Validate the use of ANOVA by testing for normality (**Shapiro-Wilk test**) and homogeneity of variances (**Levene's test**).
  3. **ANOVA Execution**: Perform the Two-Way ANOVA using Python's `statsmodels` library to obtain F-statistics and p-values.
  4. **Post-Hoc Analysis**: If significant interaction effects are found, independent T-tests (Welch's) will be conducted to analyze simple main effects.