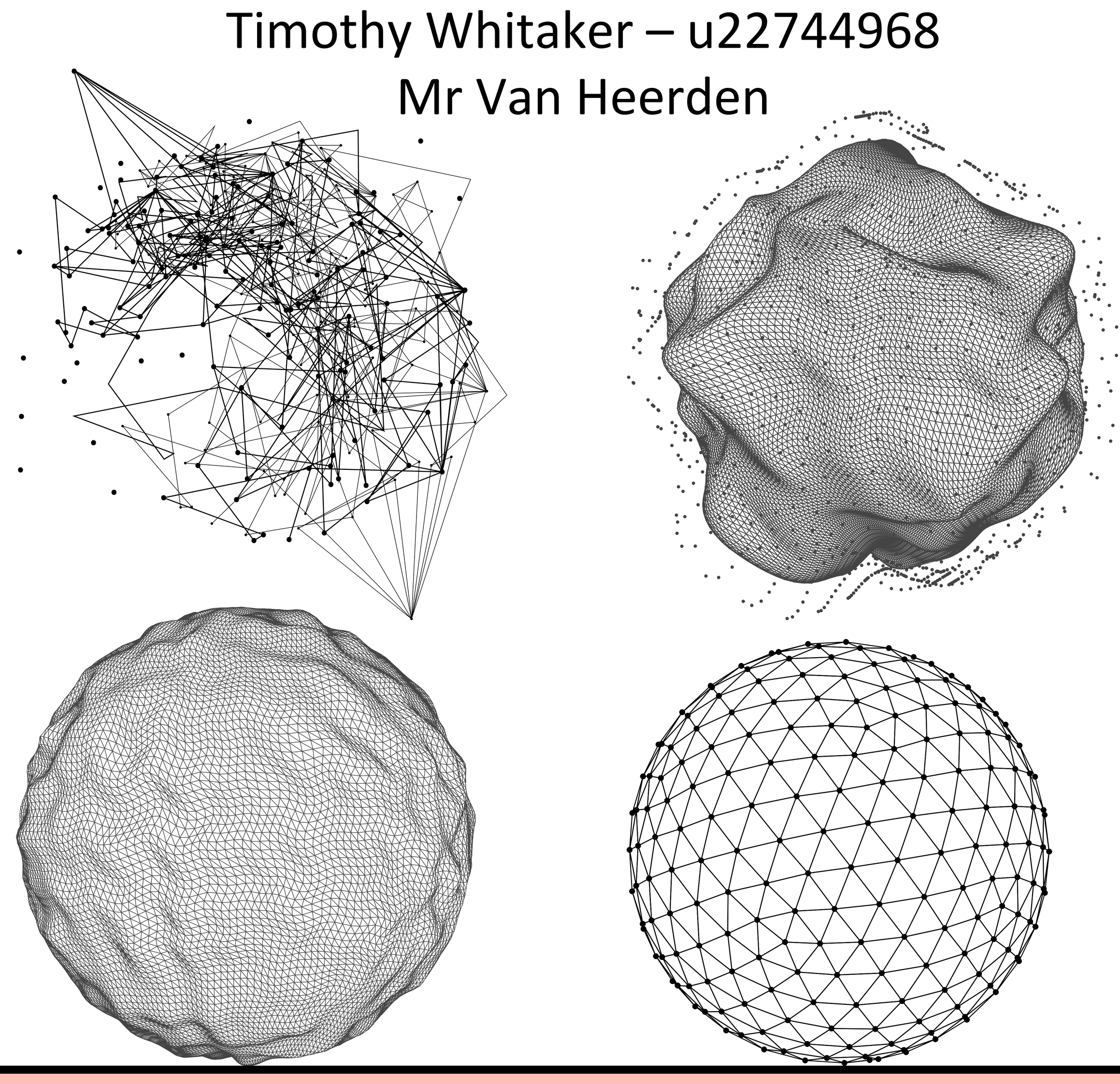
Multi-Objective Particle Swarm Optimization for Polygon Reduction in 3D Models

Timothy Whitaker – u22744968
Mr Van Heerden



Abstract

Reducing polygon count while preserving shape fidelity is critical for real-time 3D rendering and data transmission. Traditional mesh decimation (e.g., QEM) is fast but often removes thin structures and degrades curvature at aggressive budgets.

This work looks at optimising meshes, meaning optimising a point cloud using a multi-objective particle swarm optimisation (MOPSO) that can then be reconstructed at a lower polygon count. Particles encode candidate point sets on the target surface. Objectives balance multiple objectives, including Chamfer distance (How close are two surfaces to each other). Optimised points are reconstructed mainly with Ball Pivoting (Imagine rolling a ball along the point cloud and forming a triangle where the ball touches three points). An adaptive pipeline inserts points in high-error regions between rounds. Evaluation compares results with the baselining and the decimated version.

Introduction

- Problem: Aggressive polygon budgets collapse features with standard decimation.
- Idea: optimise where to place a fixed number of points before meshing.
- Approach: MOPSO + reconstruction feedback. Ball Pivoting as the primary mesher.
- Ball Pivoting is primarily used because it produces triangle counts that are comparable to the point counts.
- Goals: maximise fidelity under triangle budgets and remain robust at low samples.
- Contributions:
  - Pre-reconstruction, multi-objective MOPSO for point placement.
  - Ball point reconstruction optimisation.
  - Adaptive rounds add points only in high-error regions.
  - Fair baselines: decimation, same number of triangles, origin model with dense points, original model reconstructed with the same number of points

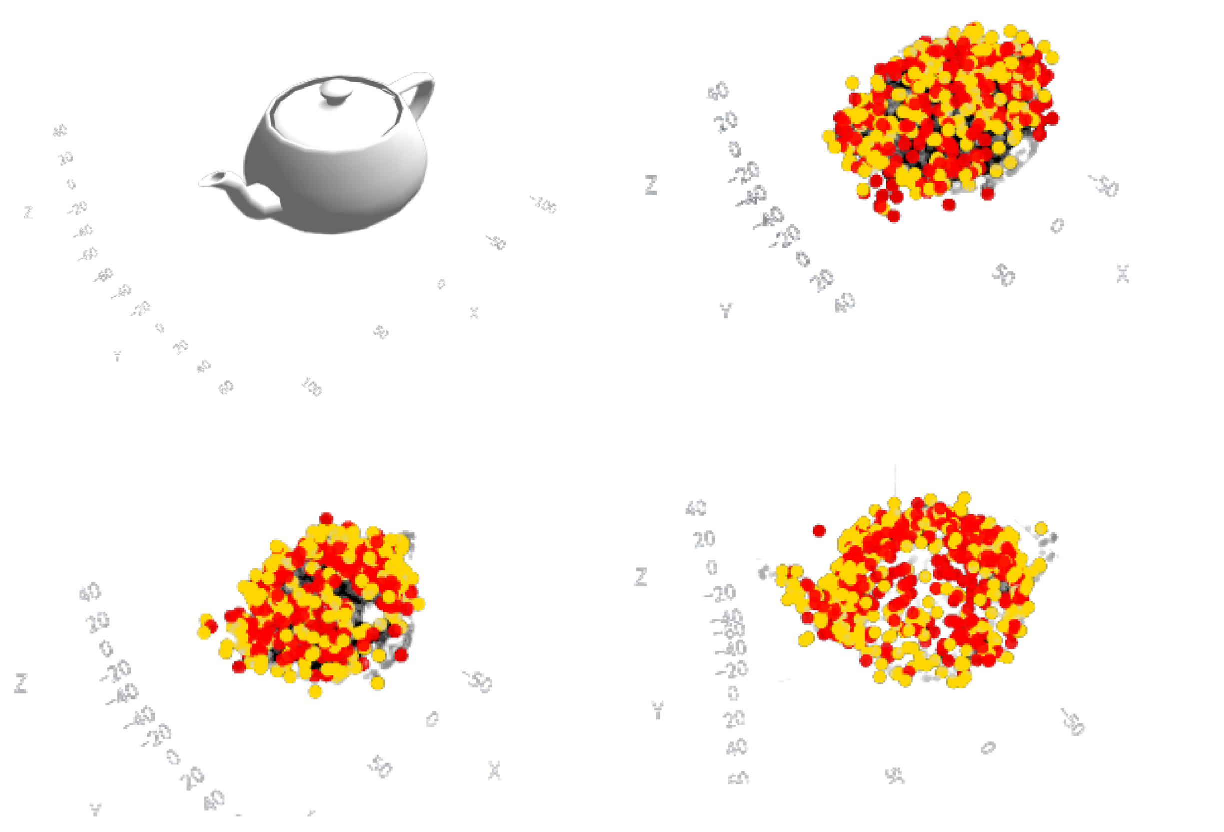


Fig 1. Showing Teapot Random Bounding Box inialization over 100 iterations

Methods and Materials

- Data and prep: Benchmark meshes
- Initialisation: Seed particles with qem, Curvature Biased, Surface Face, Vertex, Mixed (default), Random Bounding Box.
- Optimisation (MOPSO): Each particle = candidate point set. Multi-objective fitness balances Chamfer (bidirectional and after reconstruction), curvature coverage, blue-noise regularity, normal alignment, plus reconstruction penalties (over-budget triangles, holes).
- Adaptive rounds: Add points only in high-error regions (KD-Tree farthest samples). Seed multiple Pareto solutions (best Chamfer, knee, best curvature/normals) to preserve diversity.
- Reconstruction: Ball Pivoting (ball radius auto scaling). Optional Poisson/Point2Mesh for comparison.
- Baselines: Decimation via QEM to target triangle count, reconstruction from uniformly sampled points (same count as MOPSO).
- Analysis: Consistent surface resampling (N points) for all meshes. Bootstrap replicates with Holm–Bonferroni correction.

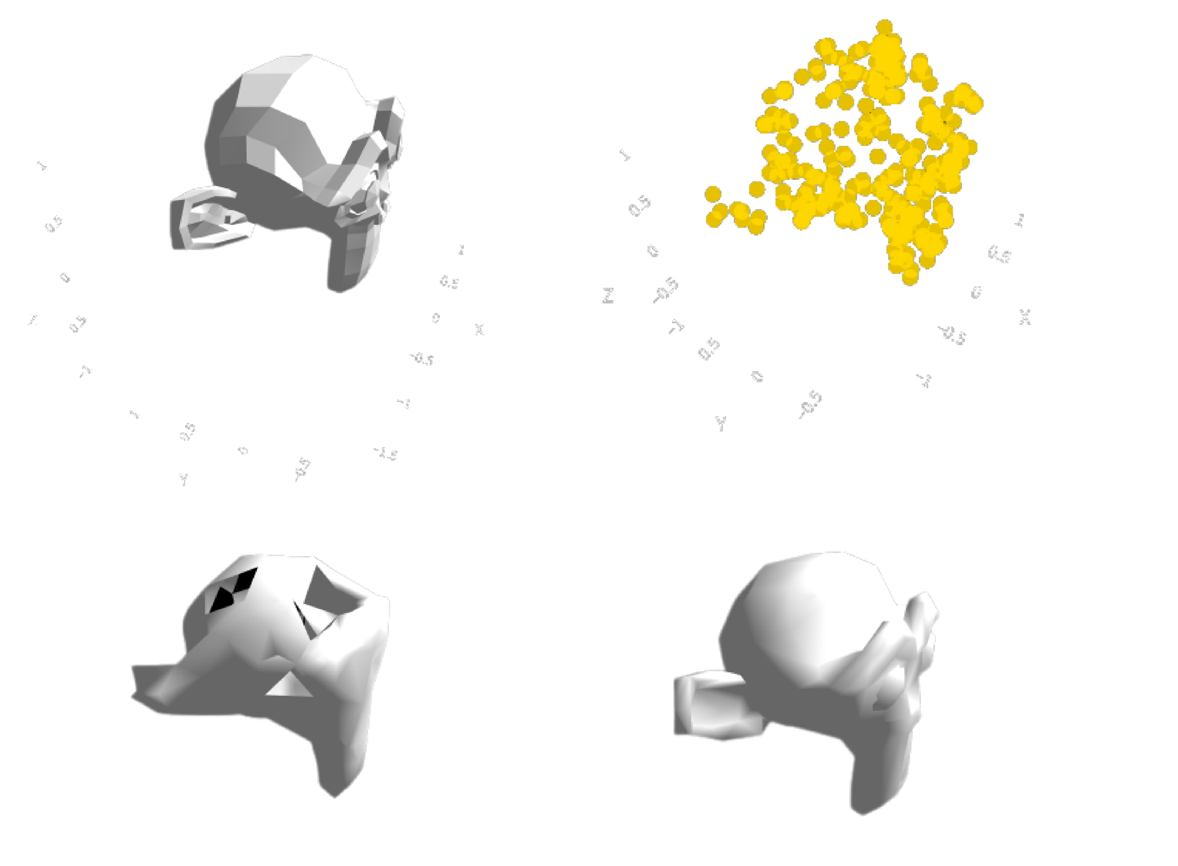


Fig 2. Showing Monkey Mixed inialization over 50 iterations with one 50 point adaptive round (from 968 to 431 Triangles).
(TL: Original. TR: MOPSO Point Cloud. BL: MOPSO Reconstructed. BR: Decimated)

Results

- Visual: side-by-side meshes (Original | Decimated | MOPSO Recon, Base Reconstructed)
- Trend chart (overlay, default view):
  - X: number of points (final)
  - Y: metric value (lower is better)
- Typical observations:
  - Decimation almost always outperforms MOPSO reconstruction; however, this gap shrinks as the number of points increases.
  - Baseline Recon (random sampling) underperforms PSO at the same point count every time.
  - On meshes with a lot of vertices unnecessarily “clumped” together, MOPSO can outperform Decimation, as Decimation uses most of its triangle budget trying to preserve those features.
  - Baseline reconstruction can perform worse at higher point sizes due to the randomness of the point selection.

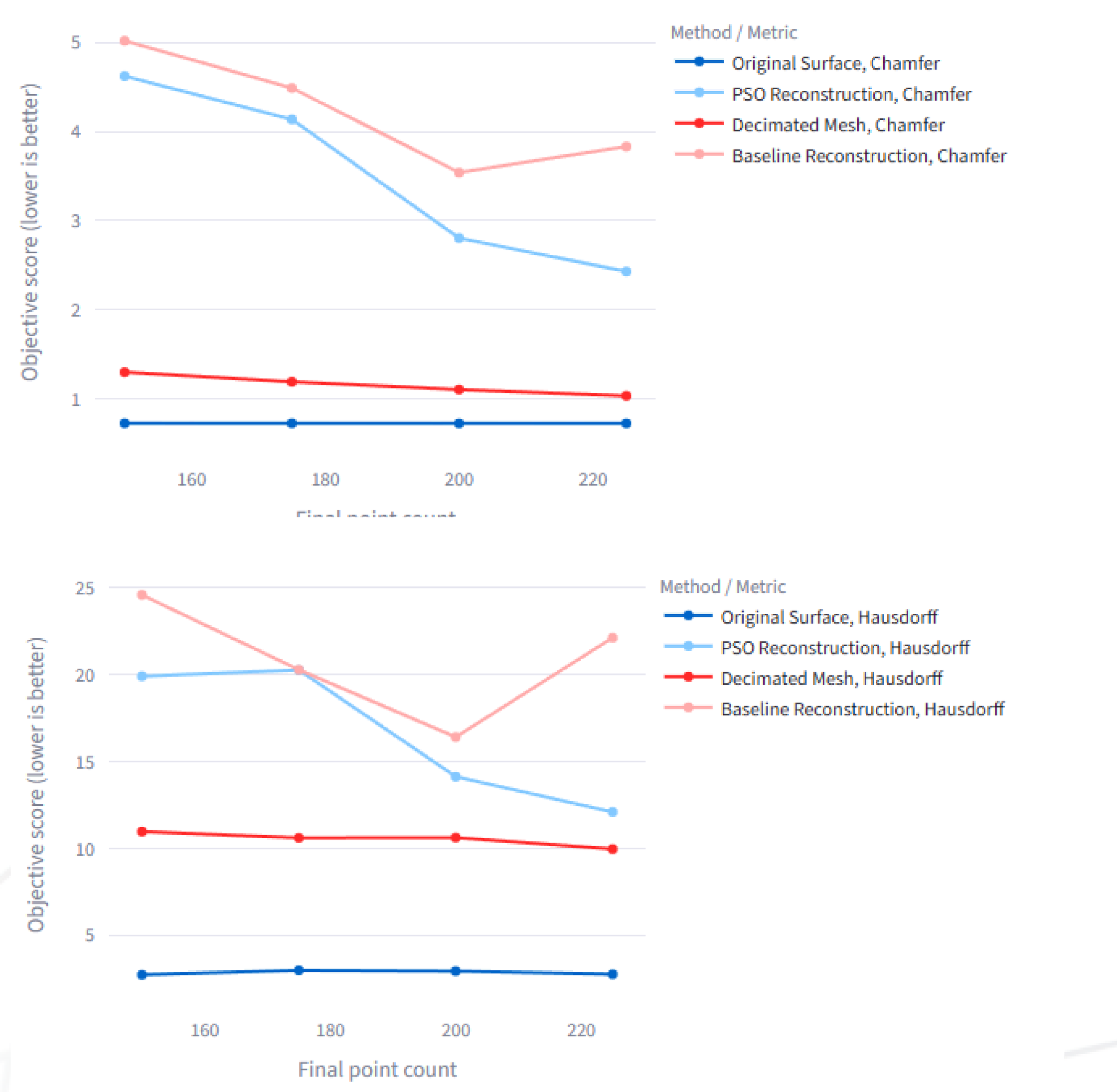


Fig 3. Line graphs comparing the results of the teapot inialized with four different point cloud sizes

Discussion

This MOPSO implementation performs relatively well at creating optimised point clouds. However, it still can not outperform decimation in the majority of cases. A problem as well is that it can be the case that initialised particles sometimes dominate the search space, especially those initialised using QEM. The MOPSO also performs best at point clouds with fewer points, as they are less likely to be dominated by initialised solutions.

It is also noted that point clouds tend to collapse in with MOPSO due to the dimensionality of the particle swarm, and to counteract this, as well as “help” the particles out, a customizable force is added to points that nudges them towards their closest surface.

Conclusions

MOPSO-based pre-reconstruction point optimisation consistently improves over naïve/random initialisation but, on average, does not yet match QEM decimation across all tested models and metrics. Current limitations are runtime cost, sensitivity to initialisation/ball-pivot radius, and failure modes on extremely thin features below sampling resolution.
It still needs to be tested the practicality of this in very large point clouds like those given by object scanning, and the refinement of other reconstruction methods using these optimised point clouds is something to be looked into further.

References

- Garland, M., and Heckbert, P. S., “Surface Simplification Using Quadric Error Metrics,” SIGGRAPH, 1997.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G., “The Ball-Pivoting Algorithm for Surface Reconstruction,” IEEE TVCG, 1999.
- Coello Coello, C. A., Pulido, G. T., and Lechuga, M. S., “Handling Multiple Objectives with Particle Swarm Optimization,” IEEE TEC, 2004.
- Fan, H., Su, H., and Guibas, L., “A Point Set Generation Network for 3D Object Reconstruction,” CVPR, 2017. (Chamfer distance usage in 3D)
- Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J., “Comparing Images Using the Hausdorff Distance,” IEEE TPAMI, 1993.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

GitHub

Contact

Timothy Whitaker
Department of Computer Science
University of Pretoria
Email: u22744968@tuks.co.za
Website: https://github.com/Timothy-333
Phone: 0723761431