Structure-Aware Geometry Decimation

CMPT 764 Course Project Presentation

Motivation

- **Problem** : Structure aware simplification of man made objects e.g tables, chair, buildings etc.
  - Existence of inherited symmetry between different parts of mesh
  - Need to maintain orthogonality
  - helps in shape preservation while decimating
- What are the problems with general decimation approach ?
  - It is based on local information at vertex/edge to collapse
  - Ignore global information and corners
  - In general not satisfactory for extreme simplification as the successive erroneous approximations may accumulate
- Our approach consider local as well as global mesh information to calculate quadric error

Pipeline

Segmentation

Manually segmenting symmetric parts	Coloring symmetric parts

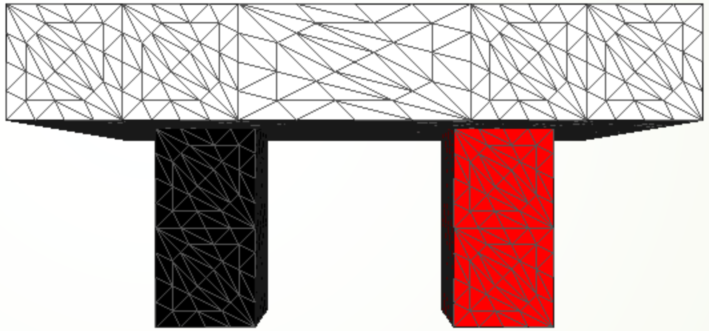
Vertex-to-vertex correspondence

Establishing vertex-to-vertex correspondence	Using Iterative closest pair algorithm (ICP)

Decimation

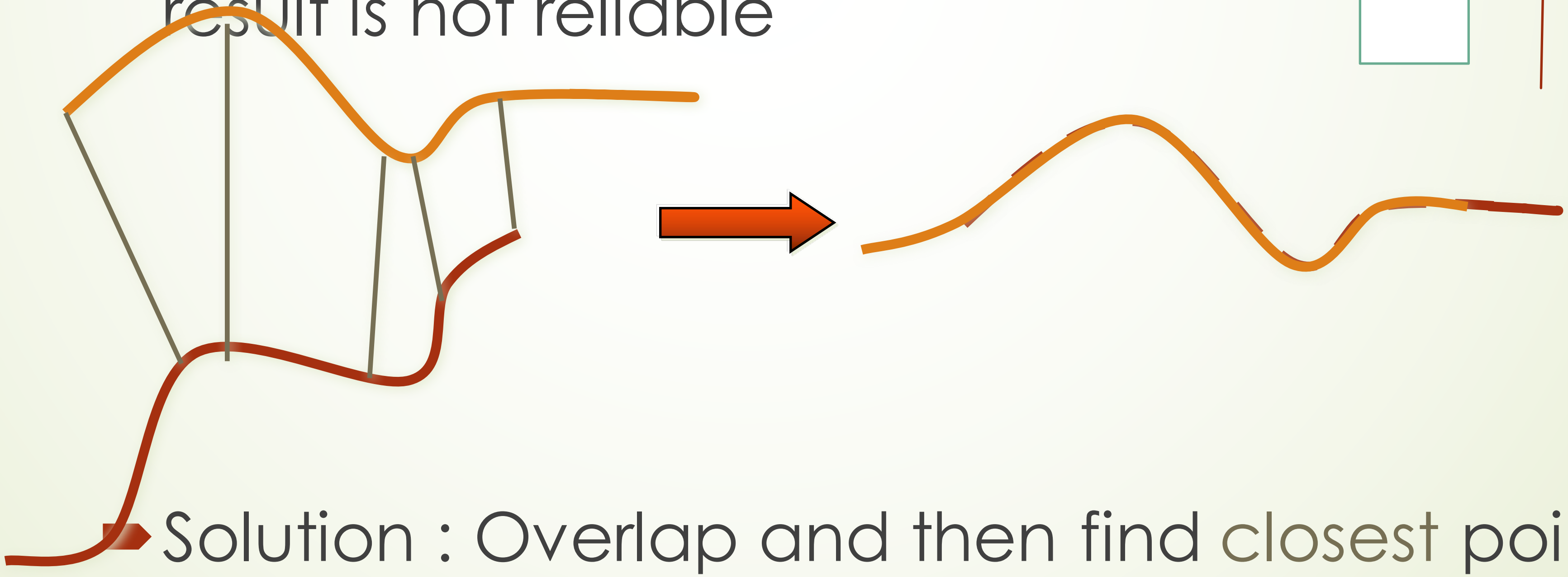
Collapse symmetric edges	Preserving shape by using modified quadric error estimation using proxy planes

Segmentation



Vertex-to-vertex correspondence

- How to find correspondences: User input?
- Tried approaches like edge swap and copying vertices
- take reflection across plane -> Manual work (need to know plane of reflection) as well as result is not reliable



- Solution : Overlap and then find closest points

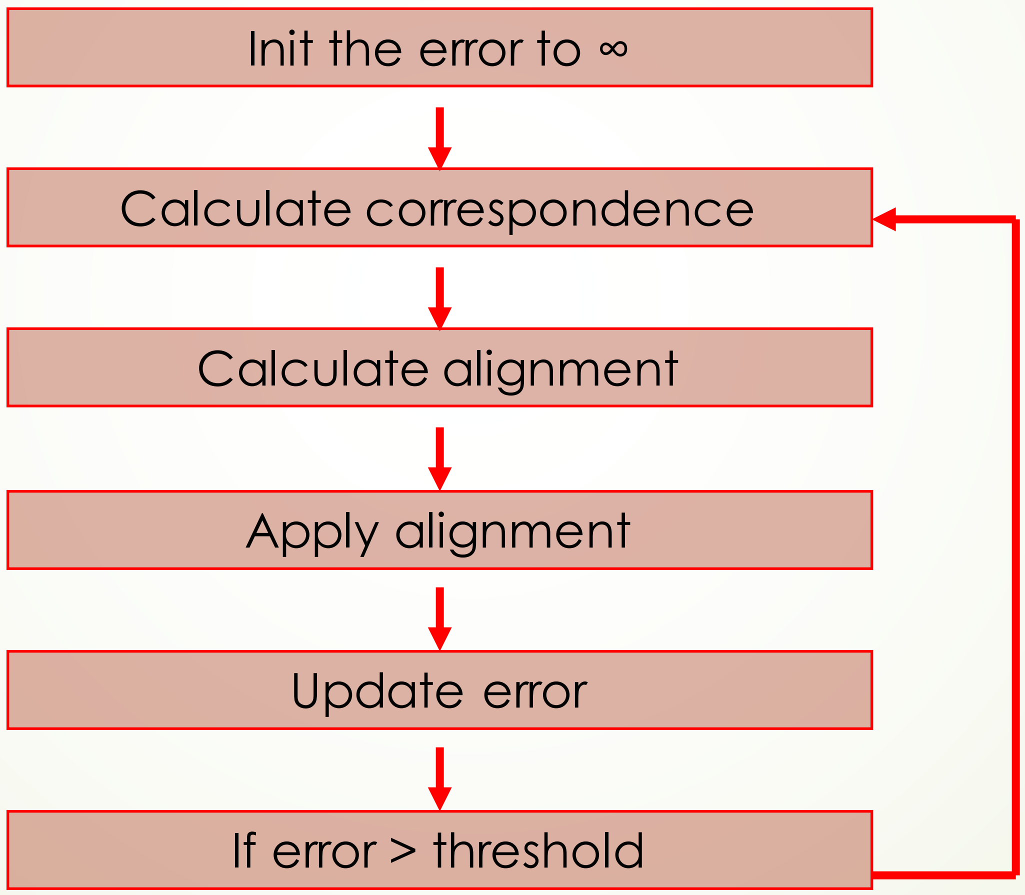
Vertex-to-vertex correspondence

- To decimate edges at all symmetric part , require correspondence
- Works for partially deformed shape as well, where number of vertices in source and target is different
- Using Iterative closest pair algorithm
  - Let S be a Source mesh point set.
  - Let T be a target mesh point set.

  We assume :

  1. $N_S = N_T$.
  2. Each point S correspond to T .

ICP Algorithm



It will converge after fixed number of iterations or when error > fixed threshold

Mapping of tabe legs

Decimation Framework

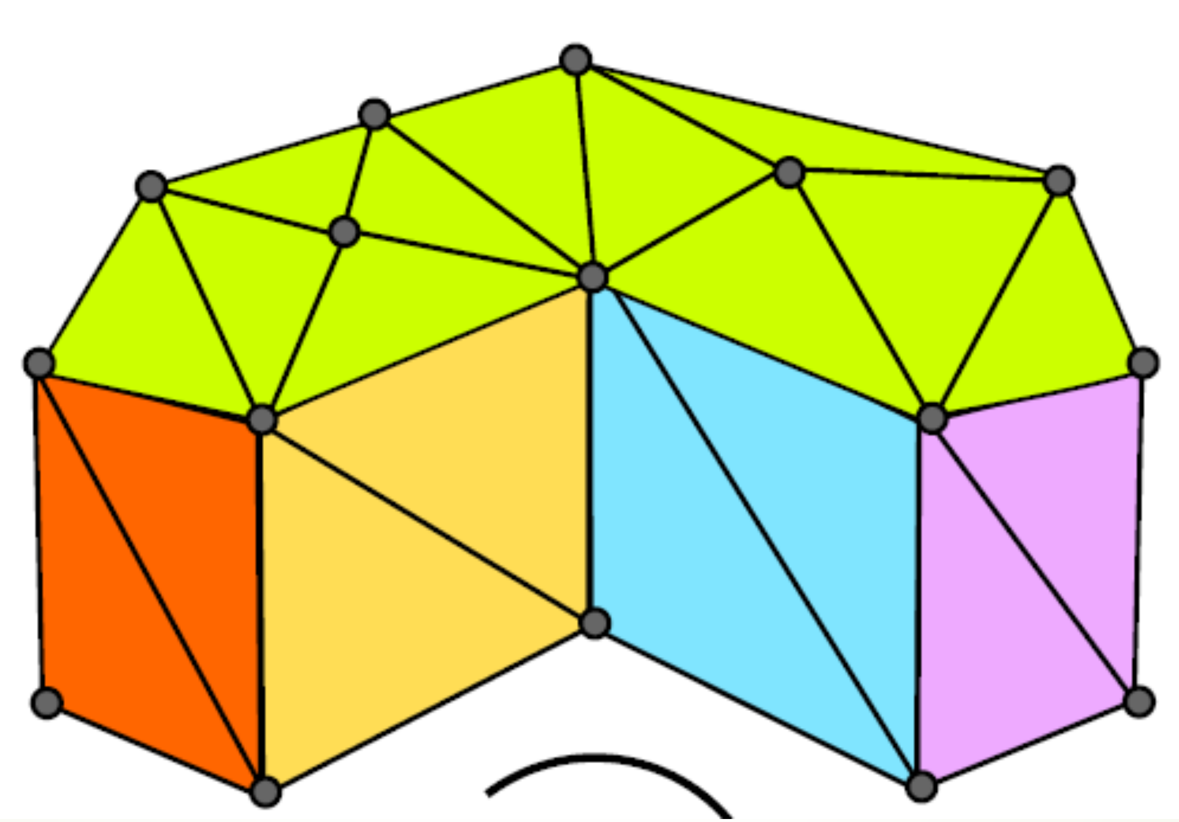
- Input :
  - Triangulated mesh M with identified symmetric parts
  - Mapping file with vertex-to-vertex correspondence information
- Output :
  - Simplified Triangulated mesh with shape preservation
- How Mapping file helps ?
  - Suppose edge E (V1->V2) is decimation candidate and reside in identified symmetric part, from mapping file fetch correspondence of V1 and V2 in other parts
  - Let it V1' and V2', while decimating V1->V2 from part1 also decimate V1'->V2'
- What global mesh information used to preserve shape ?
  - Set of proxy planes
  - Boundary/ corner preservation rules

Decimation approach

- We use a hybrid approach that combines:
  - **Concept of planar proxies:** Can be used in the calculation of local error metric. Adds larger scales of planar parts to local calculations.
  - vertices automatically migrate toward the proxies during decimation.
  - reduce the accumulation of local erroneous approximations during decimation.
  - **Structure preserving rules:** Structure of proxies is preserved during decimation by not performing edge collapse operators that violate a set of structure-preserving rules.

Planar Proxies

- Many input meshes exhibits near-planar parts that can be detected by common shape detection approaches.
- These near-planar parts are represented by **planar proxies**.
- More specifically, a planar proxy consists of a set of vertices and a plane $\boldsymbol{ax+by+cz+d} = 0$ represented as a vector = [a b c d], where n = [a b c] is the unit normal vector to the plane.
- We have used region growing approach to detect planar proxies.
- We grow iteratively the regions, accepting an adjacent triangle when its normal deviates less than the specified normal tolerance



Every triangle is associated with a proxy. Every vertex is associated with proxies of faces in which it belongs.

Error Quadrics

- We associate each triangle T with a quadric Q calculated in following manner:

$$Q = (1 - \mu)\, Q' + \mu\, Q''$$

where Q' is the quadric associated with the supporting plane of T

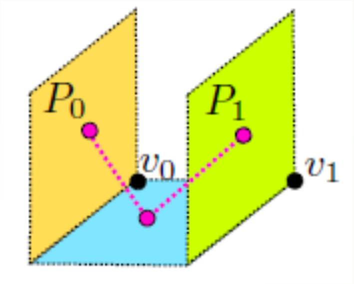
and Q'' is the quadric associated with the plane of proxy to which T belongs.

$$\begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix}$$

Structure preserving rules

- Structure-unaware decimation algorithms often destroy many features of the objects which are key to defining its geometry.
- Especially man made objects, buildings etc. have symmetry, orthogonality and defined boundaries which sometimes are lost in decimation process.
- We use following checks:
  - Proxy merge avoidance
  - Proxy preservation
  - Corner preservations

- **Proxy merge avoidance:** Figure shows that collapsing would introduce a new edge between P0 & P1 into the proxy graph.

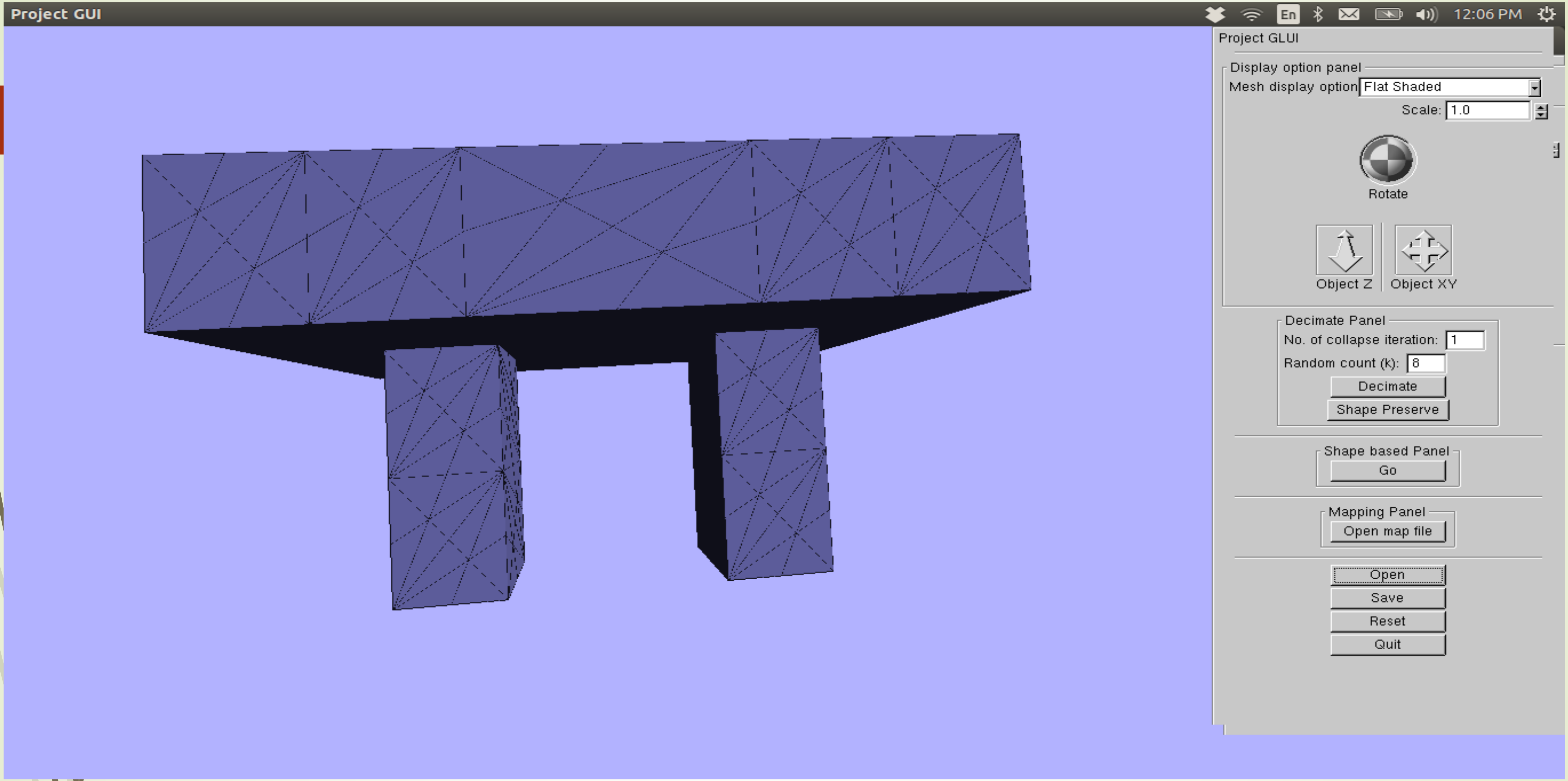


- **Proxy preservation:** When performing a collapse operation the resulting vertex receives the union of proxies of the two edge vertices. However, a proxy may degenerate into a single vertex or edge during decimation.

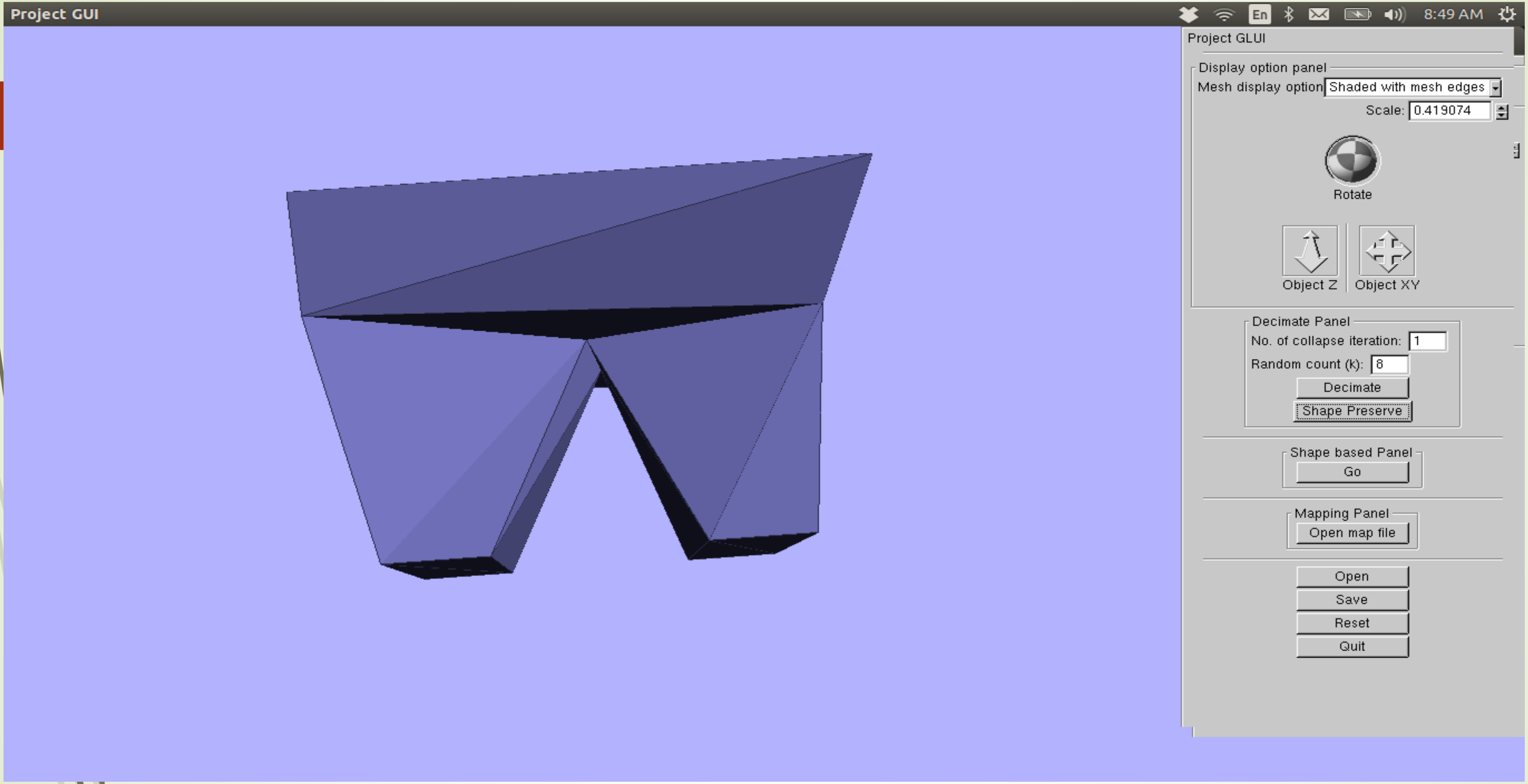
- **Corner Preservation:** Corners are very noticeable on objects made up of large planar parts. To preserve them we try to maintain the vertices which are junction of 3 or more proxy planes.

Experiment results

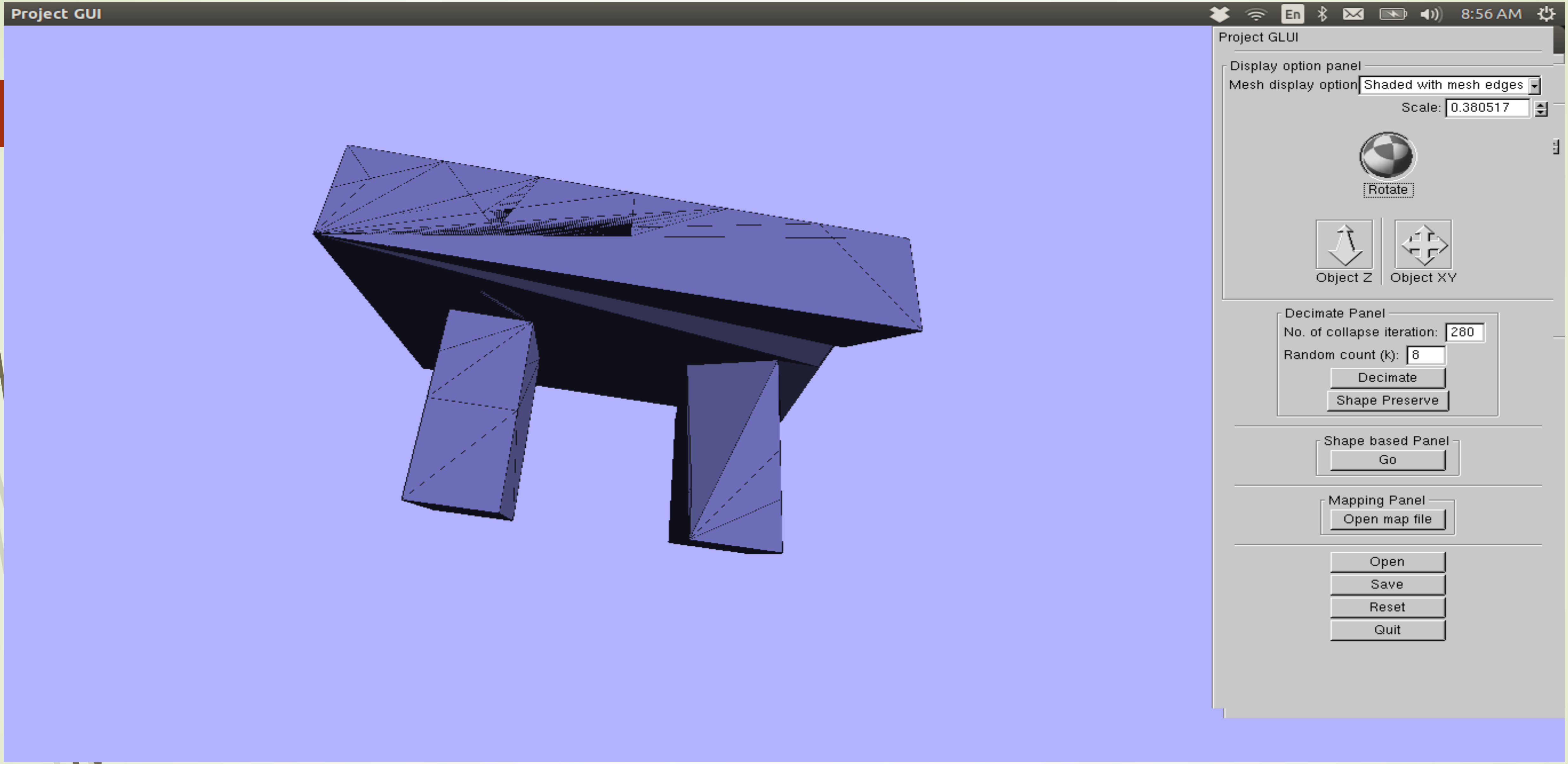
- We tried our algorithm first on a table mesh with 64 v and 124 f, then with 374 v and 744 f, and then with1490 v and 2976f
- We set μ parameter as 0.5
- We found our algorithm to preserve shape (symmetry and boundaries) even with less number of faces and vertices.
- Mesh structure of symmetrical parts (which were provided as input) were found to be similar even for high number of decimation iterations.



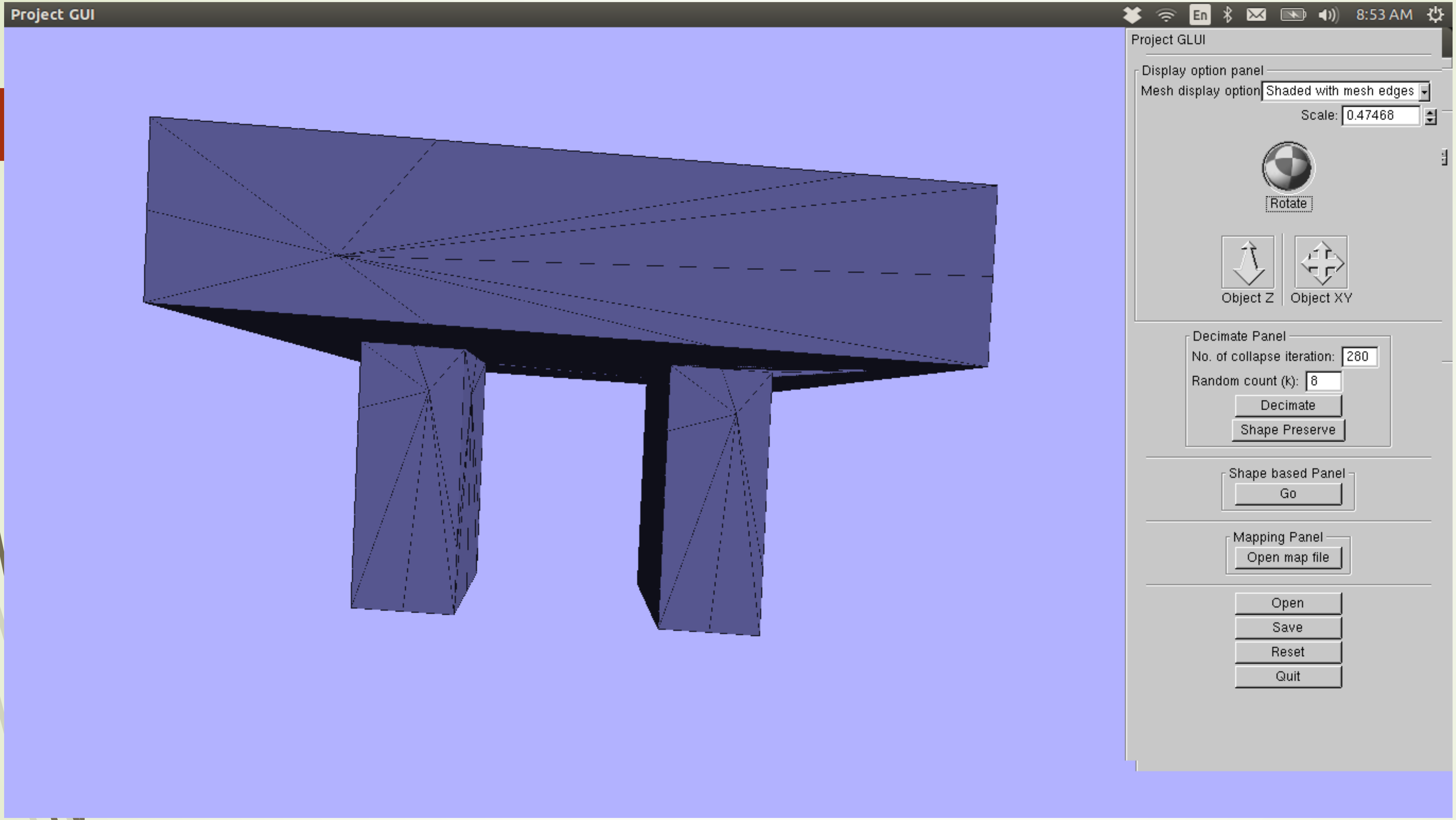
Original mesh with 744 faces.



Original mesh had744 faces. After decimation mesh structure with 34 faces.



With multiple choice algorithm having structure unaware decimation. Shape distorted when no. of iteration was 280.



With modified algorithm with structure aware decimation. Shape maintained when no. of iteration was 280.

Improvements possible

- Handling of mesh when number of vertex and faces are less. Symmetry not maintained when face count very low.
- Coming up with better error metric calculation approach.
- Coming up with a general cost function for edge selection which incorporate costs for planar information and works for all type of meshes.
- This requires meshes to have planar parts and thus might not work with non planar proxies.

References

- [1] Aleksey Golovinskiy, Joshua Podolak, and Thomas Funkhouser, *Symmetry-Aware Mesh Processing*
- [2] https://en.wikipedia.org/wiki/Iterative_closest_point
- [3] David Salinas, Florent Lafarge, Pierre Alliez, *Structure-Aware Mesh Decimation*