Computational topology - group project

Surface simplification

November 25, 2023

The goal of this project is to implement an algorithm for simplyfing surface triangulations by deleting superfluous edges. A detailed description of the algorithm is provided in the Computational topology book (by Edelsbrunner and Harer) pages 52 – 57. Here is a short summary.

The *edge contraction algorithm* removes an edge $\langle a, b\rangle$ in the triangulation and replaces it with a new vertex c (and modifies the triangulation around the vertex c). This operation is only allowed if the topological type of the surface is not altered by the modification. This can be checked by comparing the link of the the edge $\langle a, b\rangle$ and the links of the vertices a and b (the Link Condition Lemma, page 54).
Every edge contraction deforms the shape and is assigned a number indicating how much the shape is deformed by contracting the edge. The contractions are performed in order according to the size of the shape modification resulting from the edge contraction.
Your task is to implement the following procedure and test it on given surface triangulations:

- Sort the edges in the triangulation according to the deformation to the shape caused by contracting it (the Error($\langle a, b\rangle$), page 56).
- For each edge, check the link condition and decide if it can be removed,
  - if yes, determine the position of the new vertex c and modify the triangulation,
  - if no, go to the next edge.
- For evaluating the Error measure throughout the procedure, use the Maintenance of the error measure scheme (page 56).

Use your algoritm to simplify at least one of the triangulations (compute at least 3 simplifications for each model at various detail level) from the The Stanford 3D Scanning Repository (for example Stanford Bunny). Compute the homology of the original and the simplified version. Are they the same? Should they be?