24-681 Computer-Aided DEsign Interim Project Report

**Team Name: Mumbai Indians**

**Members: Ninad Kamat, Akash Sambrekar, Rahul Patil**

# Motivation

Very often data received from STL files contains several details and features which requires a high number of elements for define them accurately. However, in practice, we may not always require such detailed features. It becomes essential to define ways in which a polygonal mesh can be approximated into mesh that can be represented using lesser elements while retaining several details and features of the mesh.

# Aim of Project

This project aims at implementing one such technique. Here, given k partitions of a polygonal mesh, we wish to obtain an approximate mesh that accurately represents the mesh with much lesser polygons as compared to the one we started with.

# Error Metric

# Algoritm Used

The basic high-level algorithm for achieving Geometric Approximation of a Polygonal Surface Mesh is as follows:

1. Clustering Triangles in the mesh based on the error metric
2. Once an accurate clustering is achieved, we extract anchor vertices from the clusters that define the clustering.
3. Given the Anchor vertices, obtain a new polygon mesh from these vertices.

We shall now see each of these steps in detail.

## Modified Lloyd Clustering

In this method, we follow the following steps to obtain clustering:

1. We start with k random triangles and assign its barycenter and normal as initial proxy.
2. Using the error metric, we calculate neighbor of these triangle that is most similar to the triangle and add it to a global priority queue of triangles sorted according to the error metric after assigning the label of the proxy.
3. In this way we pass through the all the triangles using a growing algorithm such that any triangle appears in the priority queue thrice (from three of its neighbors).
4. We then traverse through the queue. If a triangle is appearing for the first time, we assigned the proxy indicated by its label and proceed. If it already is assigned a proxy, we do nothing.
5. We then calculate the new proxy for the given set of triangles associated with a given label as their barycenter and normal.
6. We repeat step 2 to 5 until we reach a point where the proxy does not change within a specified tolerance.

## Edge Extraction

## Remeshing

# Data Strctures Required

## Shell for Geometric Associativity

The shell data structure stores the connectivity information for a given polygonal mesh. It employs Hash Tables to store the associativity of polygons with edges and vertices. This data structure is useful to obtain neighboring polygons required for clustering and also to obtain polygons containing a particular vertex that is necessary in identifying Anchor vertices for edge extraction.

The data structure only stores the Vertex positions in form of an array of objects of the Vec3 class. Further, it also stores an array of Polygon information such as containing vertices and normal. The data structure is very well protected and cannot be modified by anyone from outside the class. This is essential as this represents the data that we begin with. Hence, all the tables are kept protected and can be only accessed through member functions. The class is designed in a way that it can provide answers to queries that any outside program may have through its object. Since, the class encapsulates a lot of operations. It is made such that there exists only one such object in the project, and data transfer is achieved by passing pointer to this object. Similarly, vertex and polygon information is passed through pointer to avoid excess memory usage.

## Lloyd CLuster Information

The Lloyd cluster stores the information regarding associativity of a proxy to any polygon and vice versa. It allows queries for array of polygons with a given proxy and the proxy of a given polygon. Similar to Shell data structure, we protect the data within the class and only allow access through member functions designed to answer specific queries. Associativity is stored in the form of Hash Tables.

# Work Distribution

1. Data Structures and Clustering : Ninad Kamat
2. Anchor Vertex Extraction : Akash Sambrekar
3. Remeshing and Constrained Delaunay Triangulation: Rahul Patil

# Work Accomplished So Far

* Shell Data structure functional for clustering
* Lloyd Cluster Data handling programmed
* Anchor vertex extraction program complete

# Future WOrk

* Vertex to Polygon Associativity in Shell.
* Clustering in Lloyd Cluster.
* Anchor Vertex addition for interpolating boundary.
* Constrained Delaunay Triangulation of Anchor vertices.