Weighted feature graph via hierarchical clustering

M. Ladeuil[1,2], M. Trabucato[2], A. Vaisse[2] and N. Faraj[1]

[1] LIRMM, Univ Montpellier, CNRS, Montpellier, France
[2]Virtuos, Montpellier, France

Abstract

In computer graphics, mesh clustering is a key component of various applications such as shape matching or skinning weight computation, especially when using hierarchical clustering. Garland et al. [GWH01] proposed to build a hierarchy of clusters by simplifying the dual graph of the mesh. We extend their method to provide control over cluster shapes through a combination of error metrics. Additionally, we alleviate the challenging task of finding an optimal threshold (stopping criterion) by considering a weighted feature graph that incorporates persistent cluster information throughout the hierarchy.

1. Introduction

Mesh segmentation is a significant research topic in computer graphics, as it is used for purposes such as reconstruction, mesh analysis, and skinning weights computation. Typically, mesh clustering aims to achieve either a semantic goal, by identifying meaningful parts, or a geometric goal, by finding geometrically similar parts. Different approaches have been proposed for geometric clustering such as k-means [SPDF14], mode seeking [LKjK15], variational [CSAD04] and also hierarchical clustering [GWH01]. K-means approaches require to know in advance the target number of clusters, and mode seeking approaches cannot let user choose clusters' shape whereas hierarchical clustering require finding an appropriate threshold. We propose an algorithm that leverages the strengths of hierarchical clustering and offers alternative metrics.

2. Method

Our method builds upon a hierarchical clustering algorithm proposed by Garland et al. [GWH01]. First, every face is a cluster then we compute a hierarchical representation by merging clusters together until a single cluster remains using edge collapse on the dual graph of the input mesh (each face is a node in the dual and adjacent faces are connected by an edge). An edge collapse on this dual graph leads to cluster merging. We consider the boundary of the face cluster as clear feature lines. The challenge is to compute a distance metric between nodes that preserves important features during graph simplification and enhances them via associated mesh clustering. To do so, we propose to use a weighted combination of a normal deviation and adjacency based metrics respectively noted E_{dir} and $E_{adjacency}$ to achieve our hierarchical clustering:

$$E = w_{dir} * E_{dir} + w_{adjacency} * E_{adjacency} \tag{1}$$

The **normal deviation metric** is computed through a quadric like Garland et al. did [GWH01] measuring normal deviation between two clusters:

$$E_{dir,a,b} = \frac{1}{w} \sum_i w_i \left(1 - \mathbf{n}_a^\top \mathbf{n}_{i,b}\right)^2 \tag{2}$$

where $\mathbf{n}$ is the normal of optimal plane in cluster a and $\mathbf{n}_i$ are face normals in b.

The **adjacency metric** prevents merging of clusters separated by a strong feature (a crease or a riff), we compute for all faces in a cluster adjacent to another cluster the sum of their normal deviation. This gives an adjacency metric that depends on the boundary size between clusters

$$E_{adjacency,a,b} = \sum_i 1. - 0.5 * (1 + \mathbf{n}_{i,a} \mathbf{n}_{j,b}{}^\top) \tag{3}$$

where $\mathbf{n}_{i,a}$ are boundary face normals of cluster a and $\mathbf{n}_{j,b}$ are their associated boundary face normals neighbors in cluster b.

From a given clustering, we can identify cluster boundaries edges as feature edges and intersection of 3 or more clusters as a

Figure 1: *Effect of our adjacency metric : Hierarchical clustering [GWH01] (Left) and our (Right), we can notice feature preservation on the upper part of the leg.*

Figure 2: *Construction of a hierarchical clustering, as edge collapse of the dual graph merge clusters.*

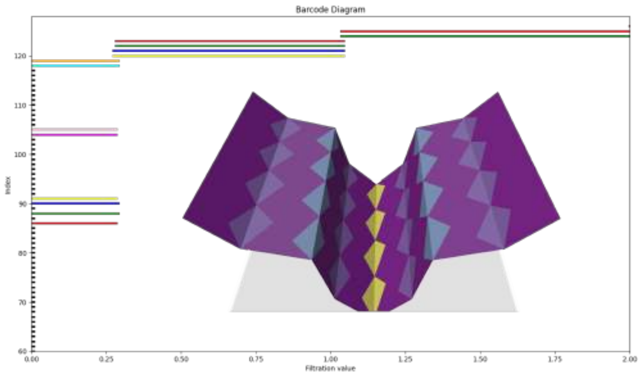

Figure 3: *Barcode diagram encodes evolution of topology, we extract lifespan of the topological component to weight our feature graph.*

feature point. This construction results in a feature graph that does not need post processing to connect edges or feature point.

So far, our method generates a feature graph given a user defined merging threshold, Figure 2. Choosing an appropriate threshold is a tedious task for a more automated algorithm thus, we propose to use a weighted feature graph. We draw inspiration from Persistent Homology to identify important features and remove topological noise. Persistent Homology provides a way to analyze and track the evolution of topological features over varying scales. In this work, nodes in the hierarchy are the topological components that have a trivial topology initially, we use the simplification metric as a distance function between nodes. Considering a complete simplification sequence in our hierarchical clustering algorithm we end up with a single node, which is the root of a tree containing every clusters; the leaves are all the initial face clusters. We can record this tree and for every pair of edges, infer when associated clusters have been merged. We have information of topological change through merging of nodes, thus we can record their birth and death and draw a persistence barcode diagram Figure3 where the length of the bars correspond to the lifespan of the components. Longer bars correspond to more robust features whereas shorter bars are more likely to be topological noise. The weight of an edge in the feature graph corresponds to the highest persistence value among all nodes whose boundary contains that edge.

3. Applications

Our motivation for this work is to extend k-way partitioning algorithms for meshes by introducing a feature aware metric in the weights of the graph. K-way partitioning algorithms seek to find an as balanced as possible partition of nodes while minimizing edge cut Directly, we demonstrate an example when using our resulting weights as input for k-way partitioning algorithm such as METIS [KK99], Figure 4 shows feature aware k-way partitioning. A more complete work will include exploration of the different metrics for the hierarchical clustering and a regularization step as features are not always recovered after the k-way partitioning.

3.1. Conclusions

In conclusion, we showed that we can use hierarchical clustering for feature detection and moreover multi resolution feature detection. In future work, we would like to improve our heuristic for feature graph weights as it does not favor lasting features but features that have the last topological "death". Also, adjacency and shape metrics are difficult to manipulate as they are not normalized, future work shall find a better formulation to ease parameter manipulation.

References

[CSAD04] Cohen-Steiner D., Alliez P., Desbrun M.: Variational shape approximation. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 905–914. 1

[GWH01] Garland M., Willmott A., Heckbert P.: Hierarchical face clustering on polygonal meshes. pp. 49–58. 1

[KK99] Karypis, Kumar: “a fast and highly quality multilevel scheme for partitioning irregular graphs”. 2

[LKjK15] Lee J., Kim S., Jeong Kim S.: Mesh segmentation based on curvatures using the gpu. *Multimedia Tools and Applications 74* (2015), 3401–3412. 1

[SPDF14] Simari P., Picciau G., De Floriani L.: Fast and scalable mesh superfacets. *Computer Graphics Forum 33*, 7 (2014), 181–190. 1

Figure 4: *Resulting partitions using METIS [KK99] from weighted edges (Left) of 8, 16 and 32 clusters of similar size, minimizing graph cut and cluster boundaries on feature.*