3D mesh description using "Subdivided Shape-Curvature-Graphs"

Florian Beguet[1,2,3] Jacques De Guise[2,3] Thierry Cresson[2,3] Matthieu Schmittbuhl[3,4] Jean-Luc Mari[1]

[1]Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
[2]Ecole de Technologie Supérieure, Montréal QC, Canada
[3]Laboratoire de recherche en imagerie et orthopédie, CRCHUM, Montréal QC, Canada
[4]Faculté de médecine dentaire, Université de Montréal, Canada

Abstract

This paper presents a shape descriptor for 3D meshes using a graph to represent a polyhedral mesh which is then used to extract patterns from the shape. The use of Subdivided Shape-Curvature-Graphs *makes it possible to not only recognize the similarities of mesh details but also determine the self-similarity of local portions of the object by adding topological information to the graph. The proposed method divides the mesh into 8 categories of patches using the discrete curvatures. These patches are cleaned; afterwards, to add topological information, a new "segmentation" patch is added. Finally, an approach is developed to extract and compare the subgraphs and thus be able to obtain the self-similarity of local parts of the mesh.*

CCS Concepts

• ***Computing methodologies*** → *Shape descriptors; Discrete curvatures; 3D mesh; Graphs;*

1. Introduction

As extraction and comparison of shapes on meshes are important steps in applications requiring shape recognition, there are many studies dealing with extraction of shapes or features of meshes using discrete curvatures, e.g. via a skeleton using an average curvature [Kea13], via a mesh divided into patches using a method of growth by region surrounded by strong variations of curvature [Lea05], and using the Gaussian curvature to extract the salient lines in a multi-scale framework [Yea12]. These algorithms break down the meshes into patches of homogeneous curvature or patches of salient features but it is not possible to extract and to pair similar local shapes of the mesh according to a percentage of similarity. Polette et al. [Pea17] propose a curvature-based analysis technique to construct a graph representative of the shape characteristics of a surface mesh. This makes it possible to find all the occurrences of a particular subgraph resulting from a known shape and according to a threshold of similarity. In the study [Pea17], the mesh is divided into eight categories of patches according to local curvature: vertices can be a peak, ridge, saddle ridge, minimum, saddle valley, valley, pit or flat spot (Figure 1.A). The ridge, minimal and valley patches are considered as transition boundaries between patches and must be added to ensure consistent continuity. This leads to a more consistent and robust descriptor graph that allows the characterization of a local form (Figure 1.B). However, this method has two flaws. The first is that it does not automatically extract topologically consistent subgraphs from the created graph, the other one is related of complex and closed meshes for which a particular node will take a central place. The graph shown in Figure 2.A is an example of a graph obtained for this kind of mesh. The central node, here circled in red prevents the proper extraction of subgraphs that can correctly describe the shape of the mesh. Our goal is to extend the existing method to categorize a 3D mesh with a *Subdivided Shape-Curvature-Graph*, then to subdivide the graph into several subgraphs with a finer structural meaning. According to a similarity criterion, these different subgraphs can then be compared with each other to determine self-similarity but also with known subgraphs which would allow the detection of deformations in cases where the similarity is lower than expected.

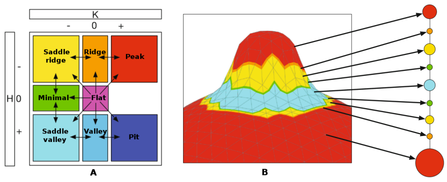

Figure 1: ***A.*** *Adjacency rules between the patches.* ***B.*** Shape-Curvature-Graph *after enrichment. (Image from [Pea17])*

2. Construction of the Subdivided Shape-Curvature-Graph

The proposed method consists of four stages:

- In the first step, an initial graph is computed based on the adjacency rules in Figure 1. This graph is then cleaned to remove all patches with a number of vertices less than a given number to obtain noise-resistant patches. When necessary, junctions between patches are added to maintain continuity [Pea17].