Visualizing Differences Between Two VTK Meshes

Release 0.00

David Doria and Adam Gerlach

July 25, 2011

Rensselaer Polytechnic Institute

Abstract

This document presents two methods of computing and visualizing the differences between to vtkPolyData mesh objects. One method, vtkPolyDataDelta, relies on reasonable normal estimates, while the other, vtkMeshPointDifference, does not.

The code is available here:

`https://github.com/agerlach/vtkPolyDataDelta`

Contents

1 Introduction

When two meshes represent the same physical object, it is informative to be able to compare them in a qualitative way. We do this by visualizing the difference between them by coloring the surface based on

the ”difference” between the meshes. We present two methods of computing and visualizing the differences between to vtkPolyData mesh objects. One method (vtkPolyDataDelta) relies on reasonable normal estimates, while the other does not. When the normals are known, the distance is computed by ”shooting” the normal of every point of mesh A and finding its intersection with mesh B. When the normals are not known (vtkMeshPointDifference), the closest point on mesh A is found from every point on mesh B. In both cases, these distances are mapped to colors and these colors are applied to the mesh for visualization of the computations.

2 Included Files

- vtkPolyDataDeltaDemo programmatically creates two meshes with known normals and compares them.
- vtkPolyDataDeltaExample reads two meshes with known normals from vtp files and compares them.
- vtkMeshPointDifferenceDemo programmatically creates two meshes with unknown normals and compares them.
- vtkMeshPointDifferenceExample reads two meshes with unknown normals from vtp files and compares them.

3 Difference Computation with Normals

The distance to mesh A is found by shooting the normal from every point of mesh B and determining its intersection with mesh A. A Modified BSP tree is used to speed up this ray/triangle intersection computation.

Figure 1 explains this distance computation graphically.

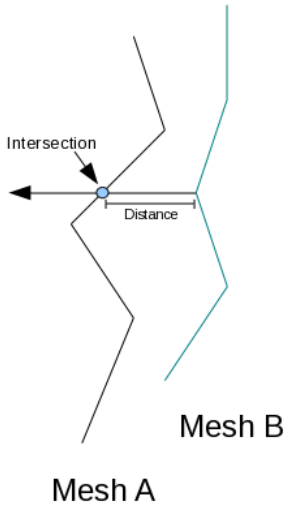

Figure 1: A graphical explanation of the normal intersection distance computation.

4 Difference Computation without Normals

The closest point in mesh A is found from every point of mesh B. The distance between these points is taken as the distance between the two meshes at that point. A KD-Tree is used to speed up this nearest neighbor lookup.

Figure 2 explains this distance computation graphically.

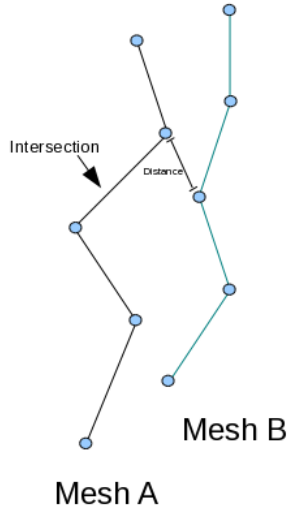

Figure 2: A graphical explanation of the nearest neighbor distance computation.

