A Conditional Mesh Front Iterator for VTK

Release 0.00

David Doria

April 19, 2010

Rensselaer Polytechnic Institute, Troy NY

Abstract

Region growing is a technique that can be used to propagate information over a mesh. In a previous submission, "A Mesh Front Iterator for VTK", we introduced an iterator that can be used with vtkPointSet subclasses to traverse a mesh. It is sometimes useful to visit only vertices that are "similar" to their neighbors by some definition. This concept can be used to select many types of common regions, including planar regions or regions with similar color, to name a few. In this paper, we propose an iterator which propagates on a mesh using a condition test which can be easily modified.

Contents

1 Introduction

Region growing is a technique that can be used to propagate information over a mesh. In a previous submission, "A Mesh Front Iterator for VTK", we introduced an iterator that can be used with vtkPointSet

subclasses to traverse a mesh. It is sometimes useful to visit only vertices that are "similar" to their neighbors by some definition. This concept can be used to select many types of common regions, including planar regions or regions with similar color, to name a few. In this paper, we propose an iterator which propagates on a mesh using a condition test which can be easily modified.

2 Algorithm

To propagate the front, we perform the following procedure.

Initialization:

- Add the seed vertex to the queue.

Iteration:

- Get the vertex in the front of the queue. Set NextId to this value.
- Add all of the vertices that are connected to NextId and pass the condition test to the back of the queue, unless they have already been visited or are already in the queue.
- Mark NextId as visited.
- Return NextId.

3 Sample Application: Planar Regions (Normal Similarity)

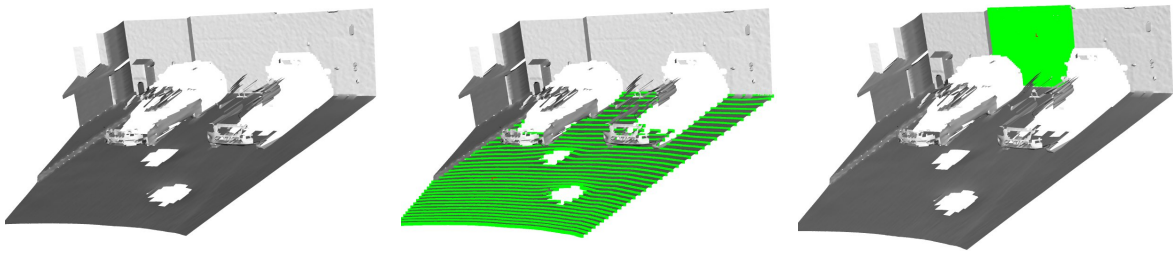
This process can be used to select regions of a mesh that are locally planar. To do this, one must simply subclass vtkMeshConditionalFrontIterator and implent the Condition function. Below is the condition function which propagates the front only to where the mesh is quite planar.

```
bool vtkMeshNormalConditionFrontIterator::Condition(const int a, const int b)
{
  //get normals
  vtkSmartPointer<vtkFloatArray> normals =
    //  vtkDoubleArray::SafeDownCast(this->Mesh->GetPointData()->GetArray("Normals"));
    vtkFloatArray::SafeDownCast(this->Mesh->GetPointData()->GetNormals());

  double n1[3];
  normals->GetTuple(a, n1);
  vtkMath::Normalize(n1);

  double n2[3];
  normals->GetTuple(b, n2);
  vtkMath::Normalize(n2);

  double angle = acos(vtkMath::Dot(n1,n2));
```

```
  if(angle > vtkMath::RadiansFromDegrees(5.0))
    {
    return false;
    }
  else
    {
    return true;
    }
}
```

4 Demonstration

Figure 1 shows two selections on a typical LiDAR data set. Figure 1(a) shows the input mesh. The green points in Figure 1(b) shows how the ground can be selected by selecting a single seed point (the red point). Figure 1(c) shows how unlike many ground detection algorithms, this region growing approach is much more general and can also be used to select a wall. These examples all use the vtkMeshNormalCondition-FrontIterator described above. Please again note that the condition can be modified to check not only normal similarity, but any region growth function you wish.



(a) Input mesh (b) Selected ground (c) Selected wall

Figure 1: Demonstration of growing regions conditioned on normal similarity

5 Code Snippet

First, you must subclass vtkMeshConditionalFrontIterator and implement the Condition function. Then the usage is straight forward:

```
vtkSmartPointer<vtkMeshNormalConditionFrontIterator> iterator =
  vtkSmartPointer<vtkMeshNormalConditionFrontIterator>::New();
iterator->SetMesh(polydata);
iterator->SetStartVertex(0);
```

```
iterator->Initialize();

while(iterator->HasNext())
  {
  vtkIdType nextVertex = iterator->Next();
  cout << "Next vertex: " << nextVertex << endl;
  }
```