A Mesh Front Iterator for VTK

Release 0.00

David Doria

April 9, 2010

Rensselaer Polytechnic Institute, Troy NY

Abstract

Region growing is a technique that can be used to propagate information over a mesh. We provide an iterator that can be used with vtkPointSet subclasses to traverse a mesh in a reasonable fashion. A “front” is emanated from a selected starting vertex, and the iterator provides the ID of the next vertex on this front.

Contents

1 Introduction

Region growing on a mesh is a technique that can be used to perform segmentation, or propagate information over a mesh. We provide an iterator that can be used with vtkPointSet subclasses to traverse a mesh in a reasonable fashion. A “front” is emanated from a selected starting vertex, and the iterator provides the ID of the next vertex on this front.

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 connected to NextId 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 Algorithm Walkthrough

Consider the grid of points shown in Figure 1.

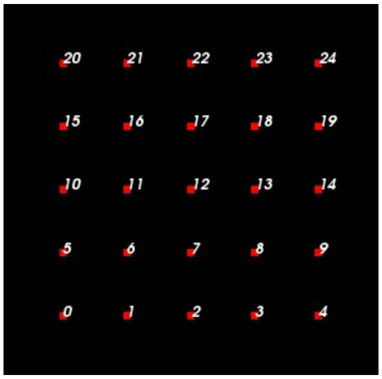

Figure 1: A grid of points

The first level of iterations is shown below:

- Set point 12 as the seed.
- Step to point 12. Add points 7, 13, 17, and 11 to the queue, remove point 12, and mark point 12 as visited.
- Step to point 7. Add points 6, 8, and 2 to the queue, remove point 7, and mark point 7 as visited.
- Step to point 13. Add points 14, and 18 to the queue, remove point 13, and mark point 13 as visited.
- Step to point 17. Add points 22, and 16 to the queue, remove point 17, and mark point 17 as visited.
- Step to point 11. Add points 16 and 10 to the queue, remove point 11, and mark point 11 as visited.

4 Demonstration

Figure 2 shows the iterations of the propagation on a grid, starting in the center. Red indicates not visited, while blue indicates visited.

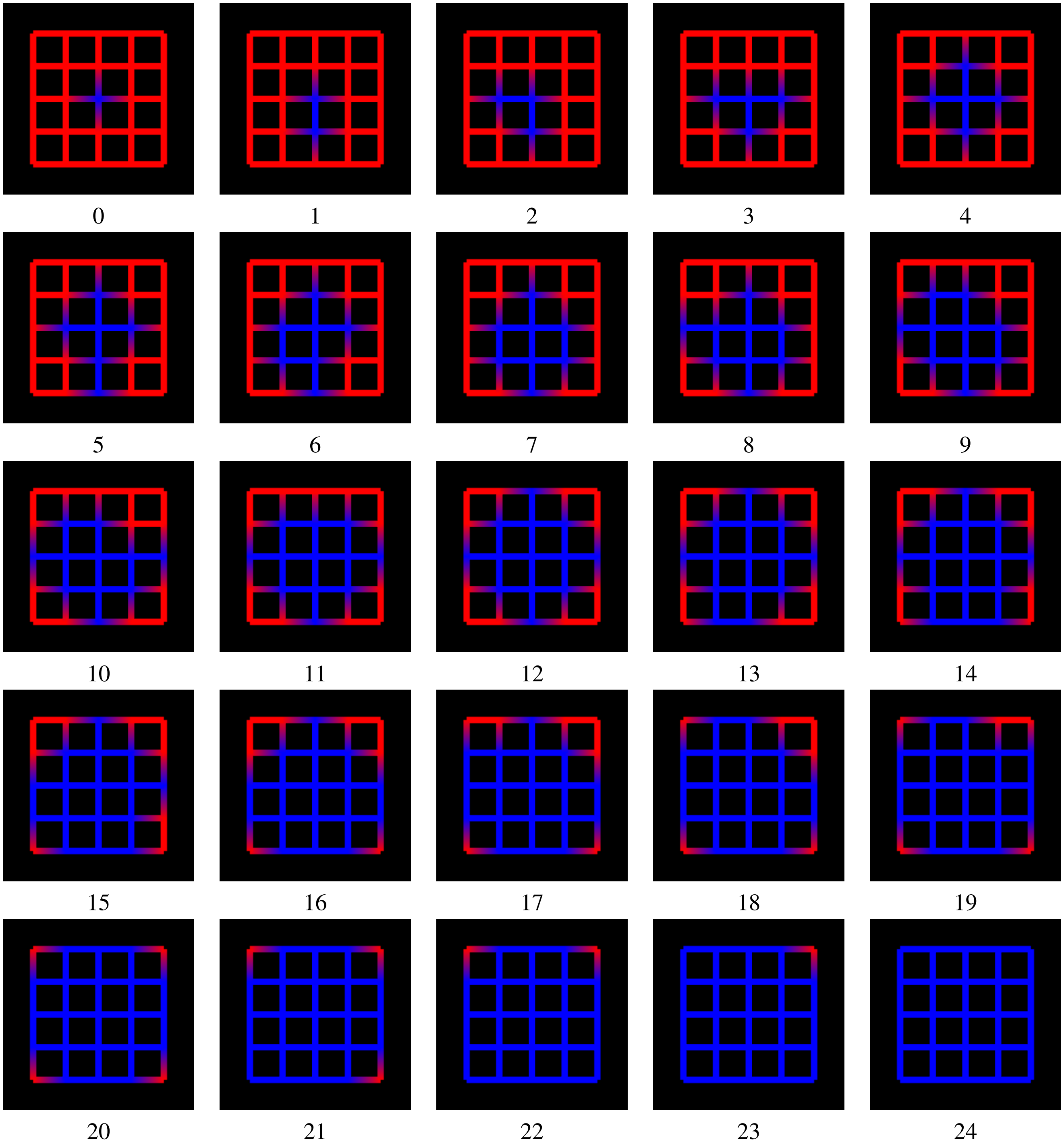

Figure 2: Demonstration of front iteration on a grid

5 Code Snippet

```
vtkSmartPointer<vtkSphereSource> sphereSource =
vtkSmartPointer<vtkSphereSource>::New();
sphereSource->Update();

vtkSmartPointer<vtkExtractEdges> extractEdges =
vtkSmartPointer<vtkExtractEdges>::New();
extractEdges->SetInputConnection(sphereSource->GetOutputPort());
extractEdges->Update();

cout << "There are " << extractEdges->GetOutput()->GetNumberOfPoints() << " points." << endl;

vtkSmartPointer<vtkMeshFrontIterator> meshFrontIterator =
vtkSmartPointer<vtkMeshFrontIterator>::New();
meshFrontIterator->SetMesh(extractEdges->GetOutput());
meshFrontIterator->SetStartVertex(0);
meshFrontIterator->Initialize();

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