Triangle Mesh Subdivision

Release 0.00

Wanlin Zhu[1]

July 24, 2011

[1]NPI Institute, School of Psychiatry,
University Of New South Wales, Sydney

Abstract

This document describes a contribution to the Insight Toolkit intended to support the process of subdivision of triangle mesh. Four approaches, linear, Loop, modified butterfly and $\sqrt{3}$ subdivision schemes were introduced.

Contents

1 Introduction

The subdivision of surface modeling has been widely used in computer graphics and computer assisted geometric design. The multi-resolution representation of surface model is required for many applications. It is a sequence of successive refinements for the input mesh. In this report, we only concerned with subdivision of triangular meshes. In [2], the authors classified most of the subdivision schemes based on the following four criteria:

- Face split or vertex split

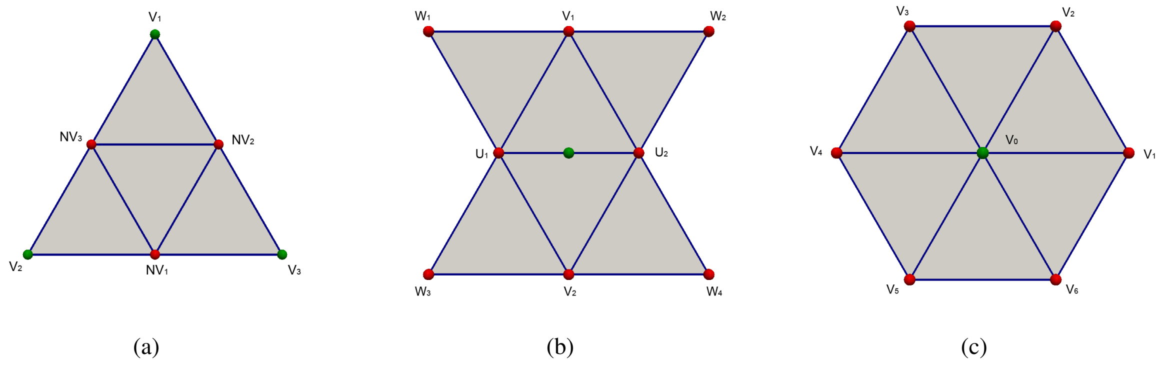

Figure 1: 1(a) Regular subdivision of a planar triangle, Green are initial vertices and red are new vertices. 1(b) The butterfly neighborhood of a vertex (green). 1(c) First ring neighborhood vertices (red) of a given vertex (green).

- Triangular or quadrilateral mesh.
- Approximating or interpolating.
- smoothness(C^1,C^2 etc.)

Here, we contribute new classes for performing triangular mesh subdivision algorithms, as follows:

- Linear subdivision
- Butterfly subdivision
- Loop subdivision
- $\sqrt{3}$ subdivision

Two are interpolating subdivision schemes and one approximating scheme, besides, the $\sqrt{3}$ subdivision is implemented which is out of the aforementioned classification. The Linear, butterfly and Loop subdivision add new vertices at each edges of a triangle, then each triangle was subdivided into four triangles 1(a). The differences among them is how to calculate the location of new points(approximation or interpolated). Whereas, the $\sqrt{3}$ subdivision inserted new vertex for each triangle face, then the triangle face was subdivided into three triangles. The difference was illustrated in Figure 2.

1.1 Linear subdivision scheme

It is the simplest interpolating subdivision scheme, the new vertices are defined as the middle point of each edge in a triangle.

$$NV_k = \frac{1}{2}(V_k^1 + V_k^2) \tag{1}$$

where NV_k is the new created point at edge k and V_k^1 and V_k^2 are two vertices of edge k. For a triangle $[V_1, V_2, V_3]$ and three new vertices $[NV_1, NV_2, NV_3]$ as 1(a). The subdivided triangles are $[V_1, NV_3, NV_2]$ and $[V_2, NV_1, NV_3]$ and $[V_3, NV_2, NV_1]$ and $[NV_1, NV_2, NV_3]$.

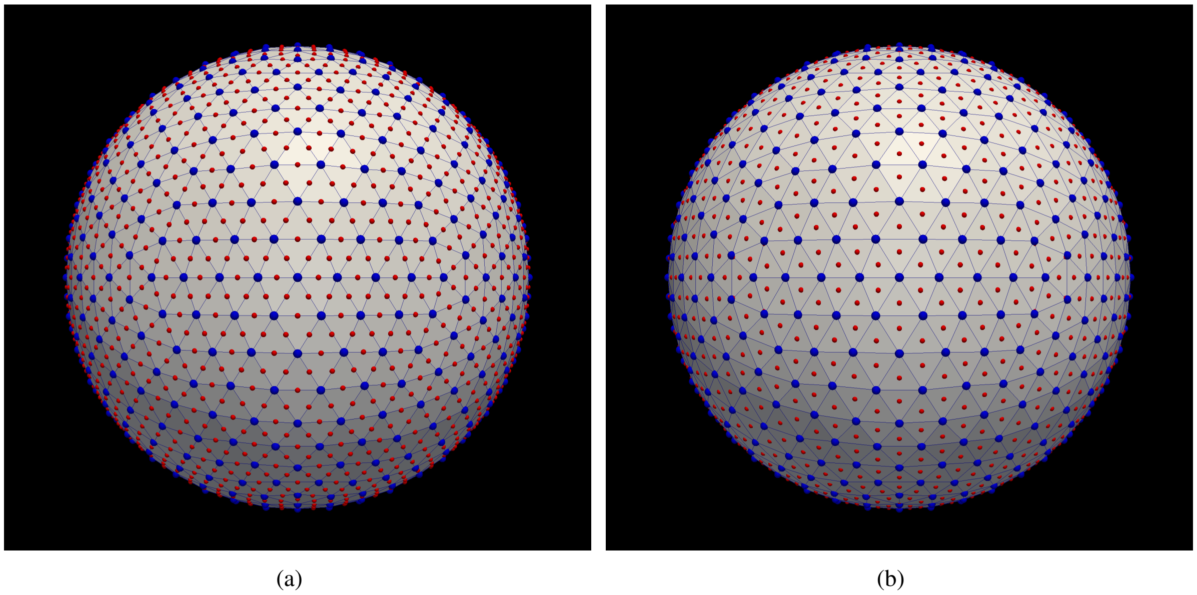

Figure 2: 2(a) Insertion of points for Linear, Butterfly and Loop subdivision schemes. 2(b) Insertion of points for $\sqrt{3}$ subdivision schemes.

1.2 Butterfly subdivision scheme

Similar to linear subdivision, except that the new vertices created using the butterfly neighborhood as 1(b)

$$NV_k = \frac{1}{2}\sum_{i=1}^{2} U_k^i + \frac{1}{8}\sum_{i=1}^{2} V_k^i - \frac{1}{16}\sum_{i=1}^{4} W_k^i \tag{2}$$

1.3 Loop subdivision scheme

The Loop subdivision scheme is a simple approximating face-split scheme for triangular meshes. The new points defined as

$$NV_k = \frac{3}{8}\sum_{i=1}^{2} U_k^i + \frac{1}{8}\sum_{i=1}^{2} V_k^i \tag{3}$$

where NV_k is the new inserted points, U_k are the two vertices of edge k, V_K are two neighborhood vertices, both defined and shown in figure 1(b). In addition, the original vertices are smoothed, for each vertex in the original mesh

$$OV_k = (1 - N * \beta) * OV_k + \beta * \sum_{i=1}^{N} V_i \tag{4}$$

Where N denotes the number of vertices of first ring neighborhood points as indicated in 1(c). OV_k is the original vertex. V_i are first ring neighborhood points.The weighting β is defined as

$$\beta = \frac{1}{N}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} * cos(\frac{2\pi}{N})^2\right)\right) \tag{5}$$

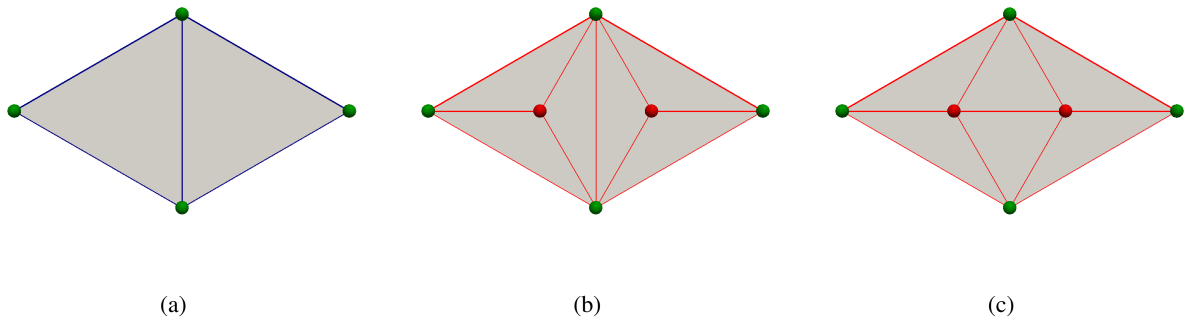

Figure 3: $\sqrt{3}$ subdivision of two planar triangles, green are initial vertices and red are new vertices. 3(a) The input two triangles. 3(b) Subdivision of two triangles with newly inserted points 3(c) Flipping the original edges between the two triangles.

1.4 $\sqrt{3}$ subdivision scheme

The $\sqrt{3}$ subdivision scheme inserts a new vertex into each triangular face of the given mesh, then flipping the original edges to produce the refinement of the original mesh. Figure 3 shows how points are inserted and how the edges flipped. In general, the new inserted point is obtained by

$$NV_k = \frac{1}{3}\left(\sum_{i=1}^{3} V_i\right) \tag{6}$$

where V_k are points belong to a triangle. Detail explanation of the algorithm could refer to [1]

2 Implementation

We provide implementations of these triangle cell subdivision algorithms as below.

- `itk::LinearTriangleCellSubdivisionQuadEdgeMeshFilter`
- `itk::LoopTriangleCellSubdivisionQuadEdgeMeshFilter`
- `itk::ModifiedButterflyTriangleCellSubdivisionQuadEdgeMeshFilter`
- `itk::SquareThreeTriangleCellSubdivisionQuadEdgeMeshFilter`

All these classes take `itk::QuadEdgeMesh` as input and output. Below is an example usage of linear subdivision.

```
#include "itkLinearTriangleCellSubdivisionQuadEdgeMeshFilter.h"
#include "itkQuadEdgeMesh.h"

  typedef float       MeshPixelType;
  const unsigned int Dimension = 3;

```

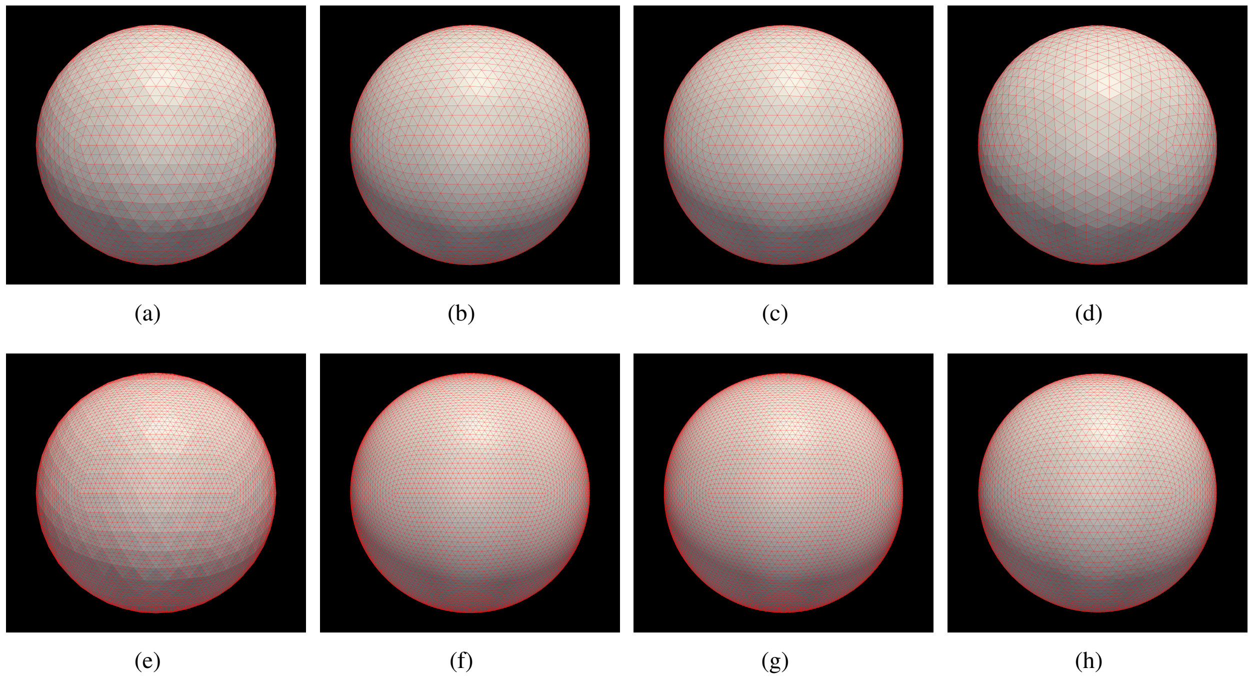

Figure 4: The first row is subdivision of resolution 1 and the second row with subdivision resolution equal 2. 4(a) and 4(e)Liner Subdivision. 4(b) and 4(f)Butterfly Subdivision. 4(c) and 4(g)Loop Subdivision. 4(d) and 4(h) $\sqrt{3}$ Subdivision.

```
  typedef itk::QuadEdgeMesh< MeshPixelType, Dimension >   MeshType;
  typedef itk::LinearTriangleCellSubdivisionQuadEdgeMeshFilter<MeshType>
FilterType;

  FilterType::Pointer filter = FilterType::New()
  filter->SetInput(reader->GetOutput());
  filter->Update();
```

In the `Testing` directory, you will find an example called

`SubdivisionQuadEdgeMeshFilterTest.cxx`

As the name indicates, the example tests four different subdivision algorithms. You can use it to conduct triangle mesh subdivision as well, the usage could be

```
SubdivisionQuadEdgeMeshFilterTest  inputMesh.vtk outputMesh.vtk 1 1
```

Where the first two parameters are the input and output file name, with the example, only the ASCII vtk file is accepted. The third parameter is subdivision type , with 0 for modified butterfly, 1 for Linear, 2 for Loop and 3 for $\sqrt{3}$ subdivision scheme.

3 Results

We tested all the four algorithms with resolution equivalent to 1 and 2. All results were shown in Figure 4. Note `itk::SmoothingQuadEdgeMeshFilter` was utilized to smooth the subdivision results.

4 Software Requirements

Following software packages are required:

- CMake 2.4 or above
- Insight Toolkit 3.20 or above

References

[1] Leif Kobbelt. sqrt (3) subdivision. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 103–112, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 1.4

[2] Denis Zorin and Peter Schröder. *Subdivision for Modeling and Animation*. 2000. 1