

Texture Synthesis on Surface by Lapped Texture

Xi Wang, 014912

Computer Science Department, Tsinghua University

wangx@ncc.cs.tsinghua.edu.cn

1. Question Background

In Computer Graphics, there are many kind of natural and manmade texture, mapping them to surface of geometry, will enhance the effect of rendering. The purpose of Texture Synthesis is to generate a "similar" texture from a sample texture. The synthesis result texture should keep the pattern and distribution, which should looked similar in human vision. Figure 1.1 illustrate a sample of texture synthesis on 2D surface.

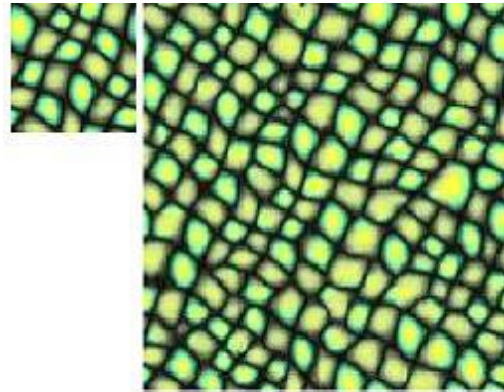


Figure 1.1 2D Texture Synthesis

The goal of Texture Synthesis on surface is synthesis texture directly to the surface. The pattern and element of source texture should also be kept on the result surface. Figure 1.2 is another example of texture synthesis on surface.

The difficulty of texture synthesis on surface is surface parametrization. It seems impossible to create a continuous parameter distribution on



Figure 1.2 Texture Synthesis on Surface

surface. Turk and Li-Yi Wei, have produced two similar algorithms[1][2], based on Pixel Based Texture Synthesis on 2D. In their algorithm, every vertex on mesh has been looked as a point on 2D plane, which was assigned a color independently, and the color of each triangle is using the color blur of its three vertices. The rule of assign vertex color is based on the

neighborhood of vertices of the target vertex. So those algorithms, should be applied on very density mesh. For each vertex, you should do the patch flatten operation. And the result of the synthesis is sensitive to the size of pattern you selected. Which is depends on the style of pattern of source texture and color distribution etc. uneasy to control.

In this report, I will introduce implementation of another texture synthesis algorithm—Lapped Texture[3], which is produced by Hugus Hoppe. The Lapped Texture is a patch based algorithm, which have high synthesis speed, simple algorithm and acceptable result.

2. Algorithm

The algorithm of Lapped Texture is very simple and clear, which can be described in a short way.

- a. Cut texture patches from input texture
- b. Specify direction and scale fields over mesh
- c. **Repeat**
 1. Select random texture patch T
 2. Select random uncovered location L for paste
 3. Grow surface patch S around L to size of T
 4. Flatten S over T
 5. Record paste operation
 6. Update face coverages

Until the mesh is covered

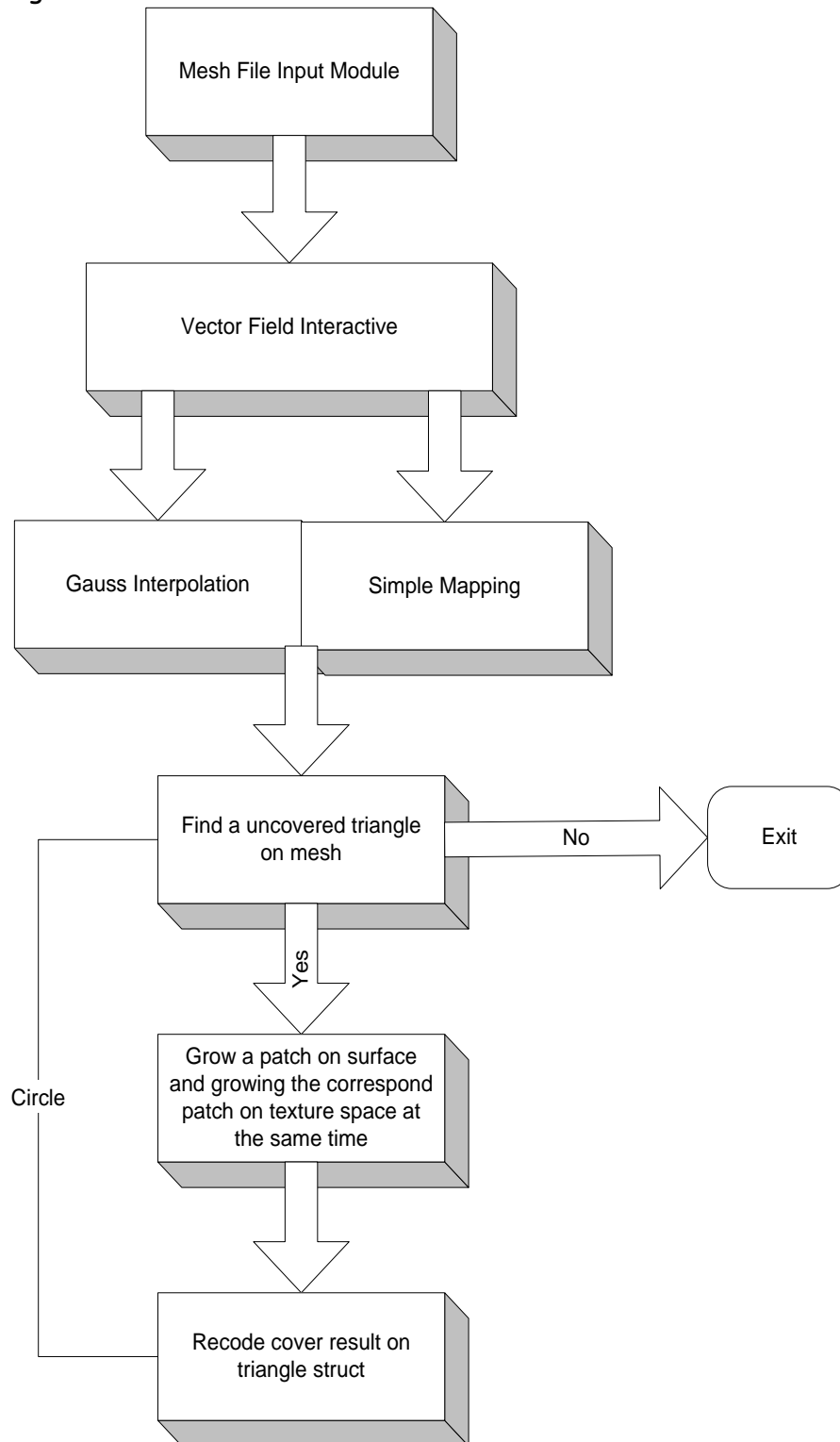
In step a. you just need to create a mask picture which can identify a region of texture you want to paste (texture patch),

In step b, the direction and scale fields are created by several specification vector which assigned by use on the vertex of mesh, in our program, I use the guass interpolation between those specification vector to get a direction to all vertex, the weight of interpolation is depends on Dijkstra's distance on surface. Another way to get the vector is to map one vector to surface, which is convenient and worked well.

In step c, in our program, at first, I find a uncoverd triangle and then grow the patch. When each triangle in geometry space added to the patch, it will correspondently a triangle in texture space is added to patch. When the patch in texture space have reach the boundary of sample texture patch which is determined in step a, the growth stop.. so now a patch of the same shape of sample texture patch are paste on the surface. The new patch will covered the previous patch on the surface. If every triangle have covered, the calculation stopped.

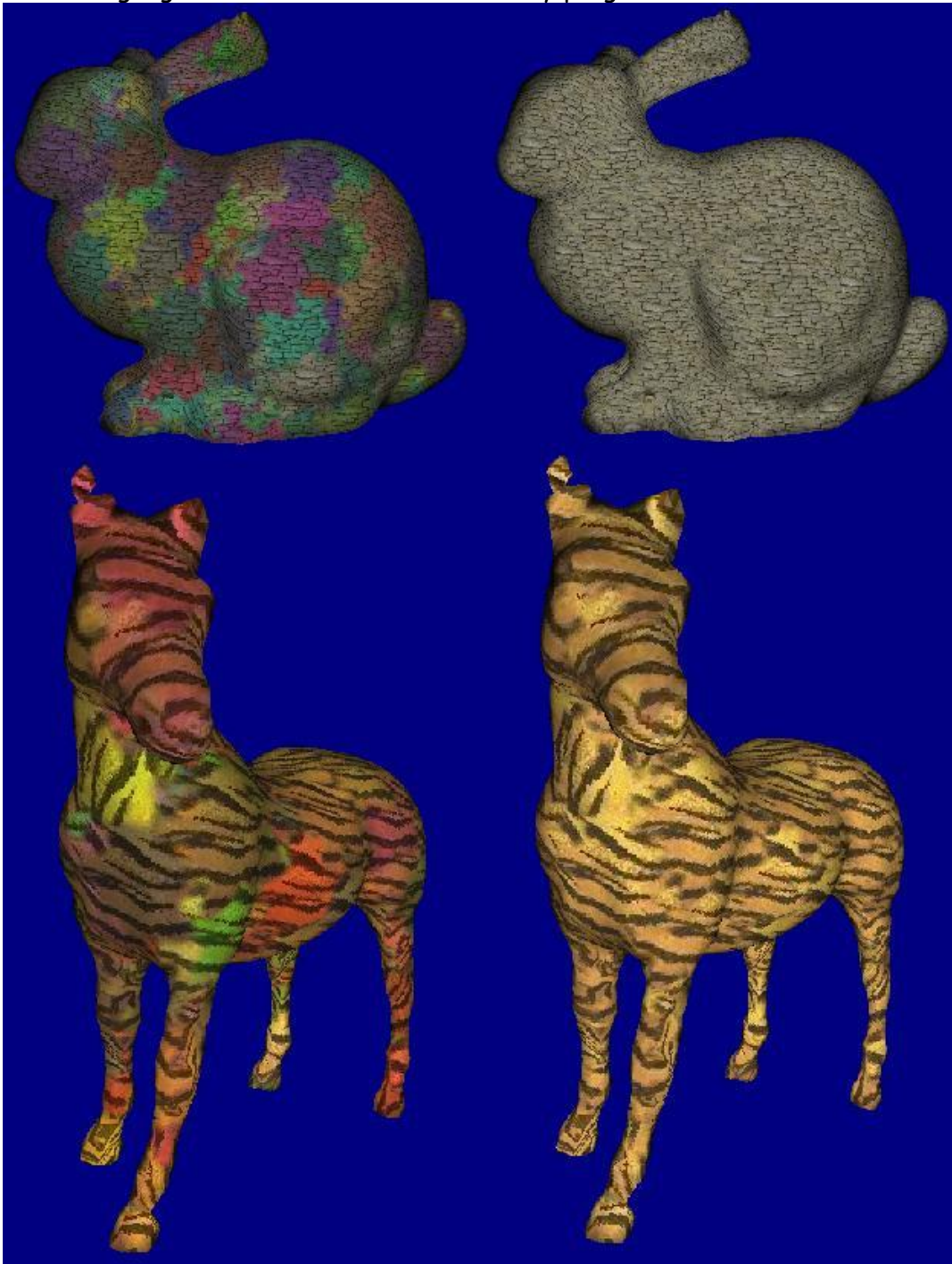
3. System Design

The program is written in Visual C++ platform, using MFC. The following figure will illustrate the pipeline of our system which have a close relation with the algorithm.



4. Result

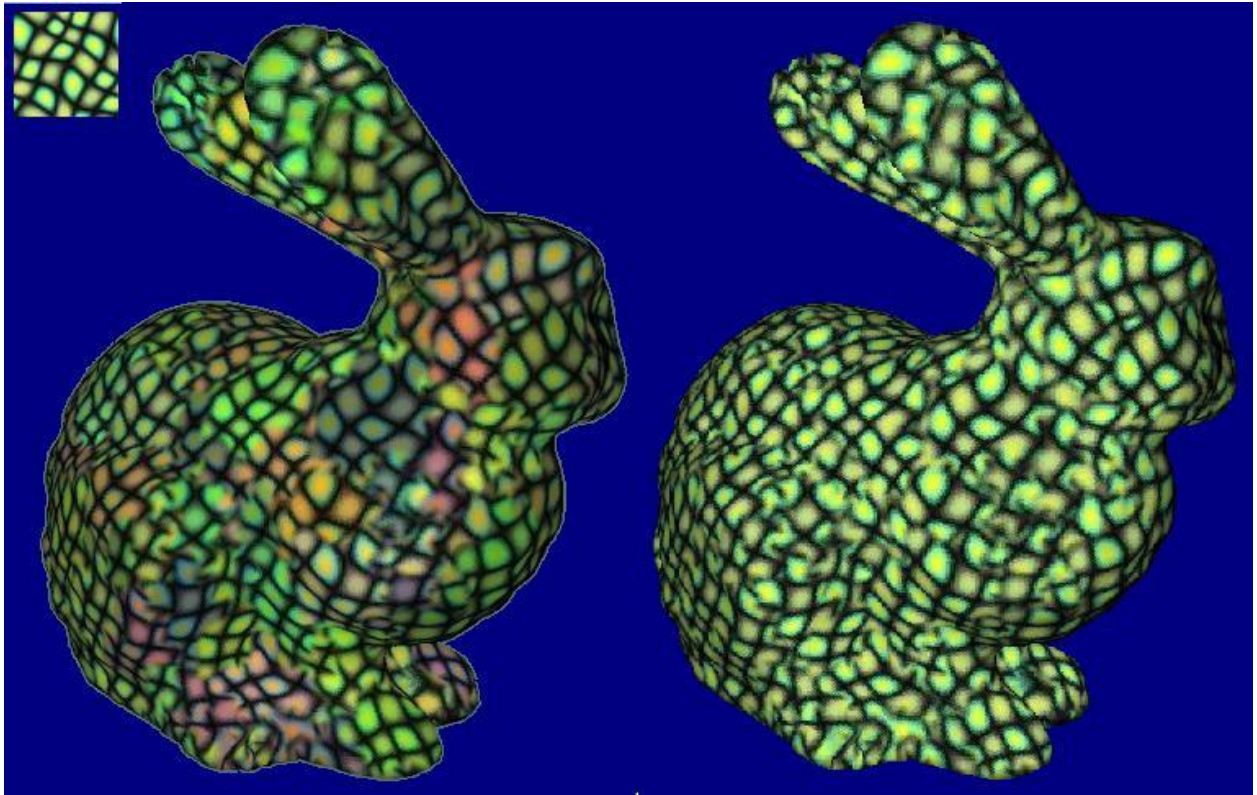
The lapped texture is very simple but actually it worked in many case. In following figure will show some result of my program.



5. Future Work and Acknowledgment

Actually, recent lapped texture algorithm just blend the boundary of each patch, in most case , the result is acceptable ,but in some case , the shortcoming of brute blend is unavoidable. The following figure will show one example. In the

figure, The pattern was break by at the boundary blending. So in my future research work, I will fixed the patch in geometry space , and displace the patch on texture space and find a most reasonable place to get the texture paste on the surface.



6. Reference

- [1] Texture Synthesis on Surface, Greg Turk, Siggraph Conference, 2001
- [2] Texture Synthesis over Arbitrary Manifold Surface, Li-Yi Wei, Siggraph conference, 2001
- [3] Lapped Texture, Hugues Hoppe, Siggraph conference, 2000