

# Quadric Error Metrics

计 73 郑林楷 2017011474

2019 年 6 月

## 目录

|     |                                    |   |
|-----|------------------------------------|---|
| 1   | 基本功能                               | 2 |
| 1.1 | 实验要求 . . . . .                     | 2 |
| 1.2 | 完成效果 . . . . .                     | 2 |
| 2   | 扩展                                 | 3 |
| 2.1 | Half-Edge Data Structure . . . . . | 3 |
| 2.2 | 边界简化 . . . . .                     | 3 |

# 1 基本功能

## 1.1 实验要求

阅读 SIG'97 论文 [1], 实现基于边收缩的二次误差网格简化方法。程序能指定输入输出的 obj 格式文件, 以及面数的简化比 (输出面数占输入面数的百分比), 例如命令行程序可以支持如下参数:

```
./mesh_simp <input obj file> <output obj file> <simp ratio>
```

## 1.2 完成效果

图 1 展示了在不同简化比下网格简化的结果, 可以看到简化效果还是十分明显的。

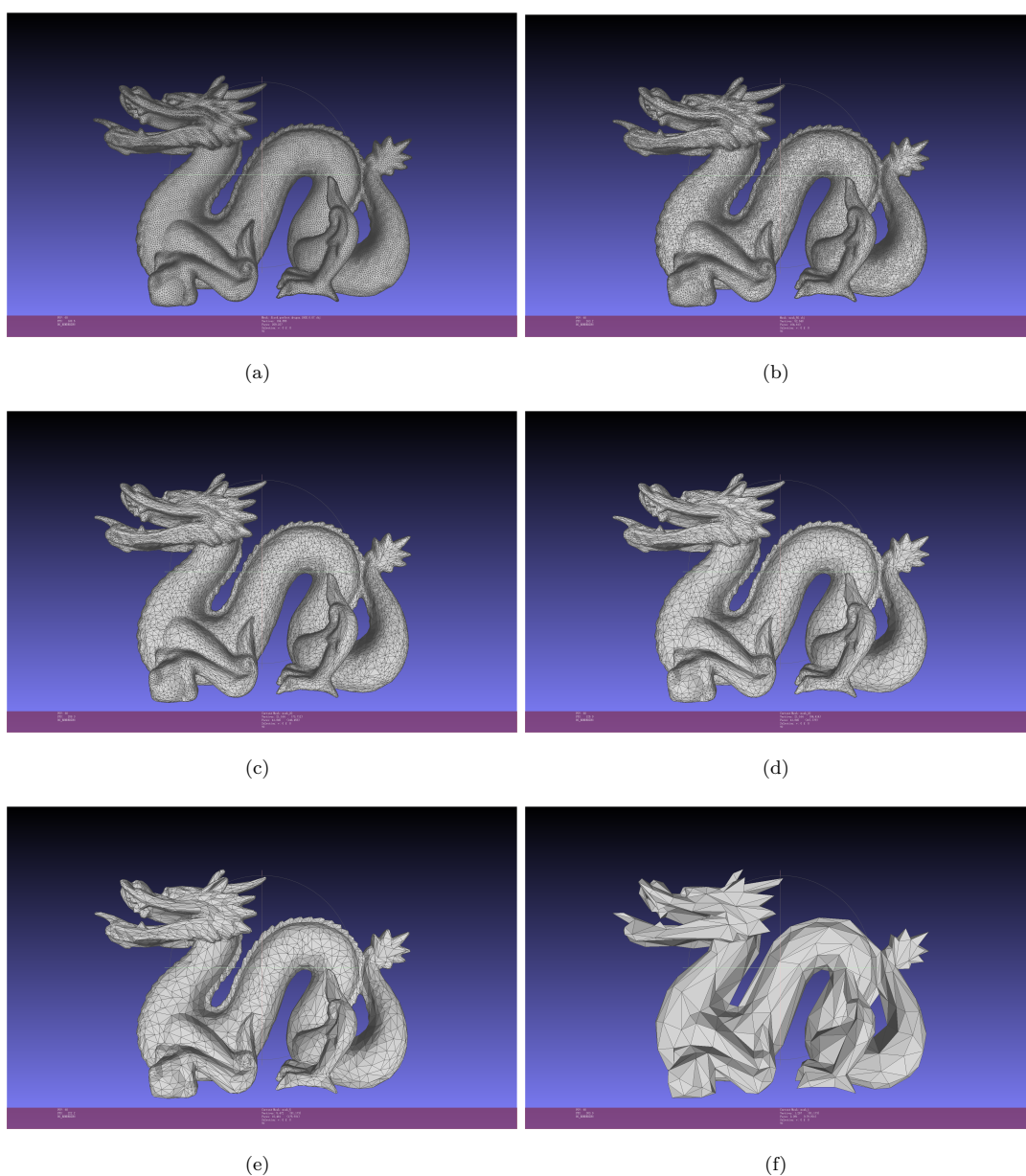
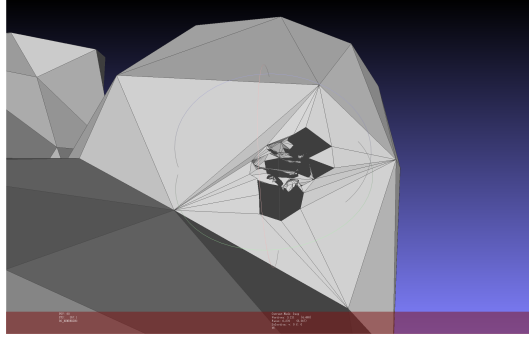


图 1: 简化效果: (a) 原始网格; (b) 简化率 50%; (c) 简化率 70%; (d) 简化率 90%; (e) 简化率 95%; (f) 简化率 99%。

**效率** 实现代码中使用了大量的 C++ STL，提高了算法的运行效率。对 `dragon.obj` 进行简化比为 0.01（即简化率 99%）的简化，总耗时仅为 11.109s。

**边界保持** 网格简化时，边界上的点往往误差矩阵较小，导致其易被删去，而很多时候我们会不希望边界点被过多地删去，因此算法对于网格边界采取了保留的处理方式。效果见下图 2。



(a)

图 2: 边界保持

## 2 扩展

### 2.1 Half-Edge Data Structure

将网格采用半边结构 `Half-Edge Data Structure`[2] 进行储存，并对算法内对于点线面的添加、删除和遍历等部分进行了修改。代码实现于 `halfedge.cpp`。

图 3 展示了使用半边结构的网格简化算法在不同简化比下网格简化的结果。

同时算法的运行效率也得到了提升。对 `bunny.obj` 采用相同的简化比 0.01，原算法耗时为 3.436s，采用半边结构储存的新算法耗时为 2.320s。

### 2.2 边界简化

虽然原算法保留了网格边界，但当边界过于复杂时，我们又想要简化边界。例如图 2，我们想要的是将中间杂乱的面都给删除，并将空洞简化，同时尽量保持空洞的大小。

可以注意到，若点对内有边界点且其边为边界，则消除后会造成空洞缩小；若点对内有边界点但其边不为边界，则消除后会造成空洞增大。于是我采取了一种点对分类赋权的方法，根据各点对的情况进行分类并采取不同的赋权，使得空洞在被简化时仍能保持原来的大小。简化效果见图 4。

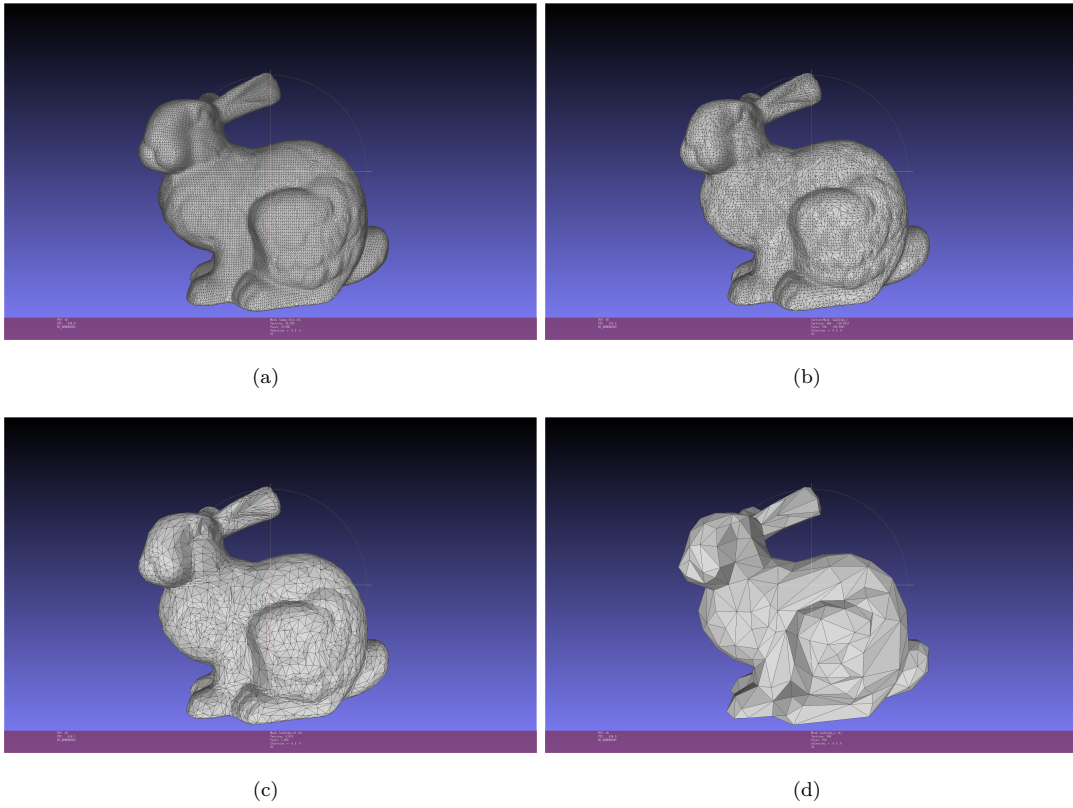


图 3: 简化效果: (a) 原始网格; (b) 简化率 50%; (c) 简化率 90%; (d) 简化率 99%。

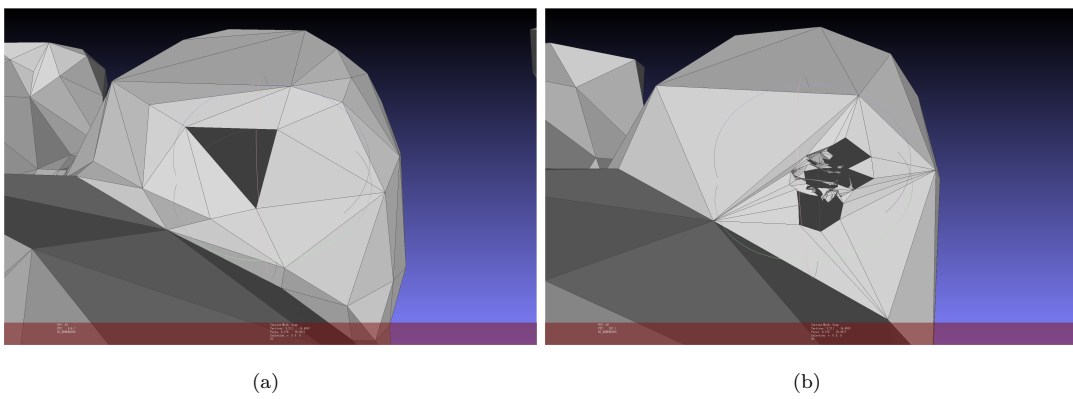


图 4: 边界简化: (a) 原算法; (b) 边界简化后。

## 参考文献

- [1] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209--216. ACM Press/Addison-Wesley Publishing Co., 1997.
- [2] The half-edge data structure. [http://www.flipcode.com/archives/The\\_Half-Edge\\_Data\\_Structure.shtml](http://www.flipcode.com/archives/The_Half-Edge_Data_Structure.shtml). Accessed: 2019-06-29.