

# Loop Subdivison 算法

## 开发环境

### 系统信息

系统	信息
操作系统	Linux Mint 19 Cinnamon
处理器	Intel® Core™ i7-4790 CPU @ 3.60GHz × 4
内存	32GB
显卡	GeForce GTX 750
编译器	gcc 7.3
编译工具	Cmake 3.10.2

### 项目依赖库

矩阵库：Eigen3

并行库：Openmp(verison:4.5)

常用库:Boost(version:1.65.1)

具体查看根目录下的CMakeLists.txt

## 编译运行

在满足依赖的情况下，进入根目录，开启终端运行：

```
$ mkdir build & cd build & cmake .. & make -k
```

### 输入模型

将模型（.obj格式）放入根目录下的models/中，并修改根目录下的input.json文件。

*input.json*

```
{
    "indir": "../models/",
    "outdir": "../output/",
    "surf": "bunny",
    "times": 5
}
```

将surf中的bunny改为模型名称。可以直接使用bunny作为demo测试。times是细分次数。

## 运行

开启终端进入build/bin/目录，并运行./z\_buffer

```
$ cd build/bin/
$ ./z_buffer
```

## 数据结构 edge\_core类

很明显该算法是一个可以并行的算法。

本来考虑使用半边结构来做，后来觉得使用半边结构在处理并行的时候比较复杂比较烦，就简单的建立一个边表，类似于翼边结构。

## 单边结构

```
struct one_edge{
    int f1,f2,v1,v2,v3,v4;
}
};
```

