

Shi XingYue (16337208)

Dr. Zhuo Su

Course title: Computer Graphics

5 June 2018

## 作业 4 报告

# 1. 用 OpenGL 实现交互式三次 Bezier 曲线的构建

## 1.1 绘制 Bezier 曲线

实现思路如下：

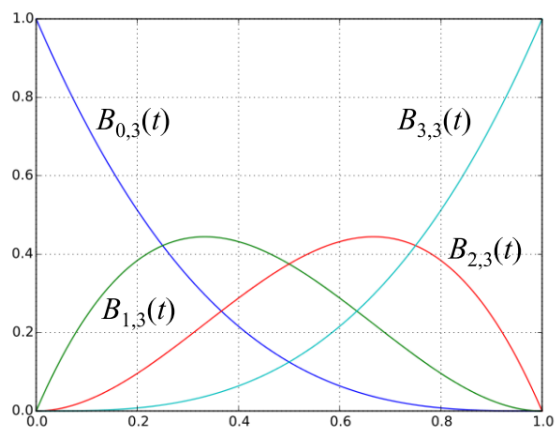


其中三次 Bezier 曲线的 4 个调和函数如图 1 所示：

## Bernstein Basis Functions

- 根据 Bernstein 多项式构成了三次 Bézier 曲线的一组基，或称为三次 Bézier 曲线的调和函数，即：

$$\begin{cases} B_{0,3}(t) = (1-t)^3 \\ B_{1,3}(t) = 3t(1-t)^2 \\ B_{2,3}(t) = 3t^2(1-t) \\ B_{3,3}(t) = t^3 \end{cases}$$

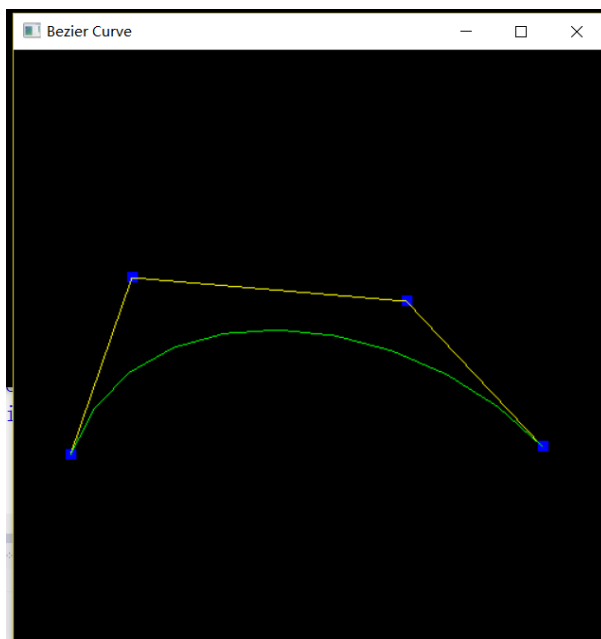


The basis functions of cubic Bézier curve  
on the range  $t$  in  $[0,1]$

(图 1 三次 Bezier 曲线的 4 个调和函数)

## 1.2 结果

运行程序后，首先用鼠标左键给出 4 个点（图 2 中 4 个蓝色的点），之后程序会输出 4 个点之间的连线（图 2 中 3 条黄色的线段）和由这 4 个点构造的 Bezier 曲线（图 2 中绿色的曲线）。



(图 2 构造 Bezier 曲线程序运行结果)

## 2. 用 OpenGL 实现简单的多边形网格数据读取和操作

### 2.1 过程

#### 2.1.1 安装配置 openmesh 库

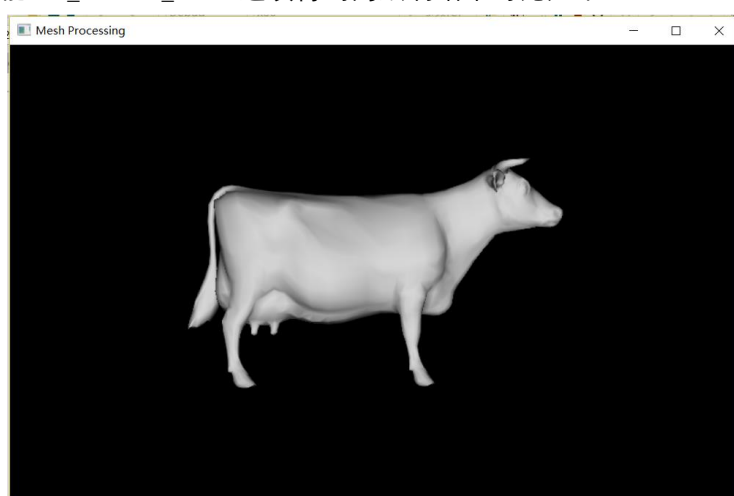
安装配置 openmesh 库需要注意以下几点：

1. 建议安装 static 版本，而不是 dll 版本。首先网上相关的教程中都是针对 static 版本的。其次经过尝试安装 dll 版本并配置后，一直提示有连接错误，无法使用 openmesh 库中的函数。
2. 建议安装 precompiled 版本。虽然源版本中有 cmake 文件，用 cmake 软件也可以自动 make，但总的来说还是 precompiled 版本安装起来更简单。
3. 项目的预处理器定义中需要加上“\_USE\_MATH\_DEFINES”和“\_NOMINMAX”两项。

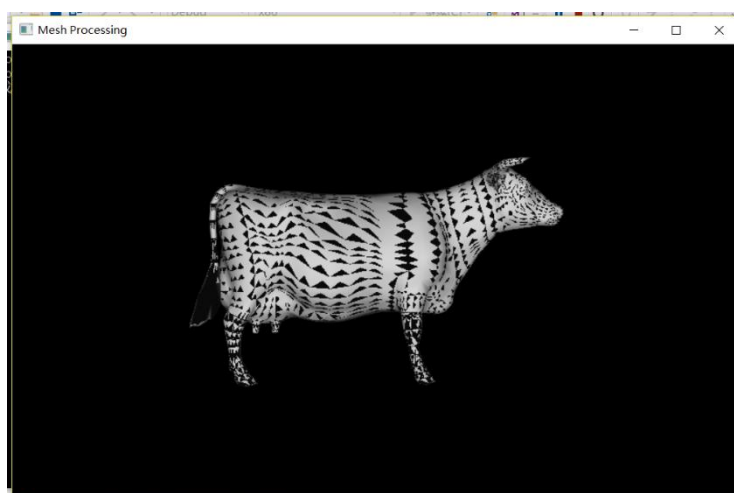
#### 2.1.2 GL\_DEPTH\_TEST 选项

启用这个选项，OpenGL 在绘制的时候就会检查当前像素前面是否有别的像素；如果别的像素挡道了它，则该像素不会被绘制，即 OpenGL 就只绘制最前面的一层。

使能/未使能 GL\_DEPTH\_TEST 选项得到的绘制结果对比如下：



（使能 GL\_DEPTH\_TEST 选项得到的绘制结果）



(未使能 GL\_DEPTH\_TEST 选项得到的绘制结果)

### 2.1.3 GLUT\_Double 选项

glutInitDisplayMode()中可以指定是 GLUT\_Double 选项或者 GLUT\_Single 选项，前者即双缓冲模式，后者即单缓冲模式。本次作业的绘制 Bezier 曲线的程序中使用的是单缓冲模式，但如果在这个涉及动态的程序中也使用单缓冲模式的话，得到的窗口会是全白的，看不到结果，要使用双缓冲模式。

经查阅资料单缓冲与双缓冲的特点与对比如下表：

单缓冲	实际上就是将所有的绘图指令在窗口上执行，就是直接在窗口上绘图。这样的绘图效率是比较慢的，如果使用单缓冲，而电脑比较慢，会看到屏幕的闪烁。（🐼我的情况则是全白没有显示）
双缓冲	实际上的绘图指令是在一个缓冲区完成，这里的绘图非常的快，在绘图指令完成之后，再通过交换指令把完成的图形立即显示在屏幕上，这就避免了出现绘图的不完整，同时效率很高。
对比	一般用 OpenGL 绘图都是用双缓冲，单缓冲一般只用于显示单独的一副非动态的图像。

(表 1 单缓冲模式与双缓冲模式)

## 2.1.4 和显示列表相关的几个函数

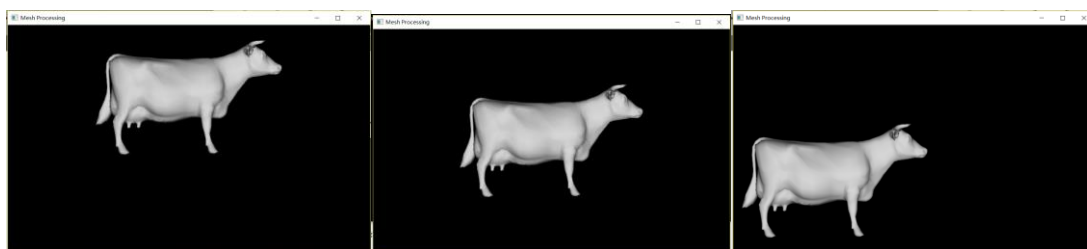
以 showFaceList 这一显示列表为例，分析几个相关函数，代码如下：

```
1 //用到的几个和顶点列表 LIST 相关的函数
2
3 GLuint showFaceList, showWireList;//declaration
4 showFaceList = glGenLists(1);
5 glNewList(showFaceList, GL_COMPILE);//glNewList()
6     ...
7 glEndList();//glEndList()
8 glCallList(showFaceList);//glCallList()
```

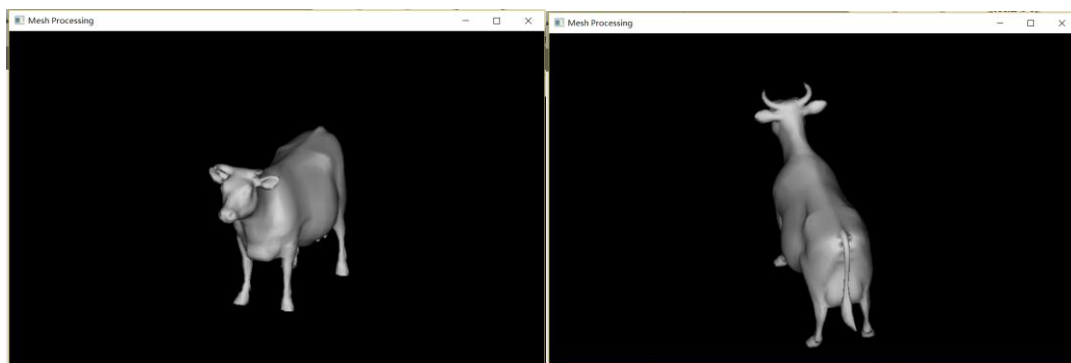
1. `glGenLists`(GLuint range)：用于生成一系列空的显示列表，参数 range 为生成的空显示列表的个数。
2. `glNewList`(GLuint list, GLenum mode)：用于创建一个显示列表，参数 list 为列表名称，mode 为编译模式。
3. `glEndList`()：用于替换参数列表。
4. `glCallList`()：用于执行多个显示列表，将所有收集的要显示的点传给 GPU 显示。

## 2.2 结果

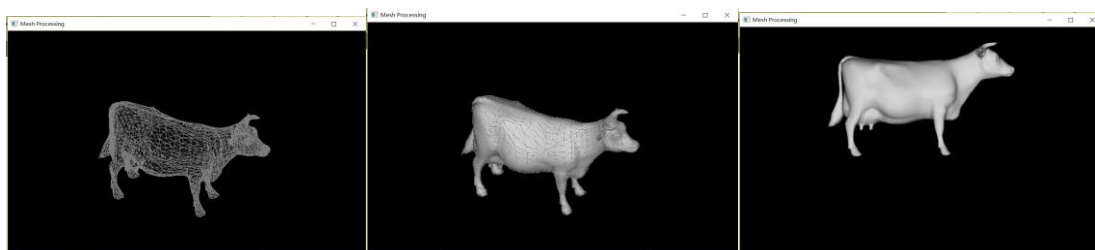
### 2.2.1 平移



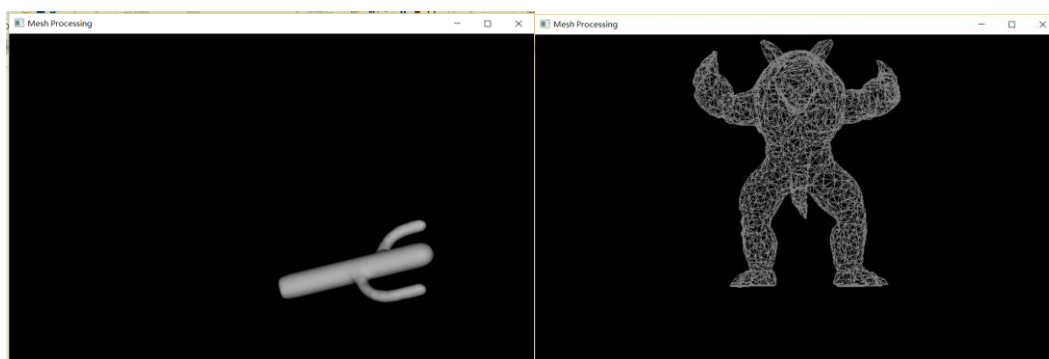
## 2.2.2 旋转



## 2.2.3 三种显示模式



## 2.2.4 读取不同格式的文件



# 3. 文件说明

作业提交中的各个文件（夹）内容见下表：

code 文件夹	2 个程序分别的代码，分别名为 bezier 和 mesh
----------	-------------------------------

demodata 文件夹	mesh 程序中用到的 3 种类型的输入文件 (.off, .ply, .obj 文件)
demovideo 文件夹	2 个程序的运行录屏
exe 文件夹	2 个程序的可执行文件 (由于配置问题可能无法运行, 可以看运行录屏)
glutdlls37beta 文件夹	使用的 OpenGL 的 dll
reportHW4 文件	本报告