数据可视化课程论文

FFD 三维形变平台

姓　　名：程文宣、蔡睿杰、杨嘉溢

学　　院：大数据学院

专　　业：国际商务（大数据商务与管理）

任课老师：庄吓海

2018 年 12 月 30 日

1. 问题和数据

1.1 问题描述

根据本次项目要求，我们需要基于 VTK 开发一个具有 GUI 功能的工具，用于实现非线性变换 FFD 的可视化和交互，开发的项目需要实现以下功能：

- 可视化形变场
- 可读入 FFD 文件设置位移并输出
- 可设置调节控制点的位移

1.2 数据描述

本次输入文件和输出文件都是 FFD 格式文件，FFD 文件是一种自定义文件格式，用来存储进行 FFD 变换后每个控制点的位移数据。

FFD 文件中的各条信息以#开始，两个##号之间的文字表示信息名称，例如 dimension, one to one 等。具体内容位于下一行。Control grid size 告诉我们整个 FFD 网格维度是 17(u)*17(v)*17(w)，即每个维度上都有 17 个控制点。FFD 文件中最关键的信息是 offsets of the control points,即控制点的偏移量。例如第一行中-0.001843 0.000363 -0.004586 告诉第一个控制点在三维空间中该偏移多少单位。

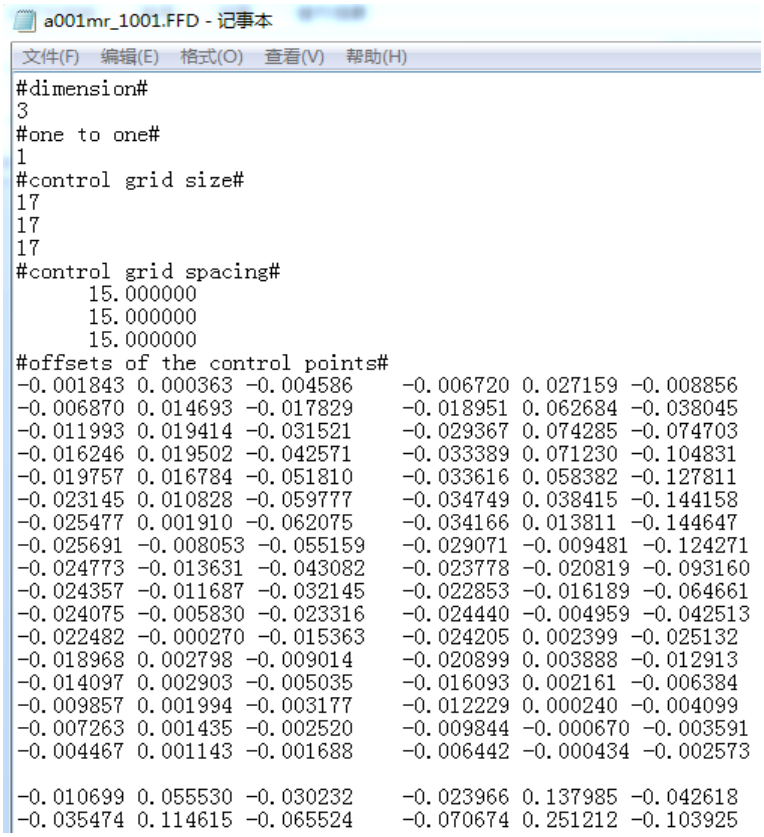

图 1 FFD 文件

上图是使用微软系统自带的记事本打开 FFD 文件，每换一行，u 增加 1。每打印一个制表符，v 增加 1，每打印一个空行，w 增加 1。

2. 数据处理

2.1 算法描述

1. 3D Free-Form Deformation

FFD 的基本思想是通过操纵包含物体的一个空间平行点阵来完成形变。被操纵的空间点阵决定了该物体的形变函数，该函数指定了物体每个点的新位置。形变原理是若控制点发生变化，则通过形变函数影响对象点阵，由新的对象点阵生成新的 3D 图像。

（1） 控制点阵的生成

对于 obj 文件，获取每个点的 x,y,z 值。每个维度上用最大值减去最小值，得到点阵的边长，确定 8 个顶点的位置，这样控制点阵就能包裹整个物体。在读取 ffp 文件后，应根据文件中的 grid size 调整每个维度的控制点个数。

（2） 对象点阵的生成

调用 Getpoints()接口获取读入的 obj 文件中 Polydata 的顶点信息。

（3） 形变函数

对于图像上任意一个对象点，其在控制点变化后的 Q(x',y',z')可由形变函数计算得到：

$$Q(x',y',z')=\sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}B_{i,3}(u)\,B_{j,3}(v)B_{k,3}(w)P(i,j,k)$$

$P(i,j,k)$表示索引为 i,j,k 的控制点的 x,y,z 值。u,v,w 参数位于 0-1 之间。

对于不同的基函数对应不同的形变函数，我们选择了三次贝塞尔曲面的基函数和三次 B 样条曲面进行分析。

三次贝塞尔曲面的基函数：

$$B_{m,3}(t)=\frac{3!}{m!(3-m)!}t^{m}(1-t)^{3-m}$$

三次 B 样条曲面：

$$B_{m,3}(t)=\frac{1}{3!}\sum_{j=0}^{3-i}\frac{3!}{m!(3-m)!}(t+3-i-j)^{3}$$

导入同一文件，对上述两种曲面分别进行形变，如下图所示：

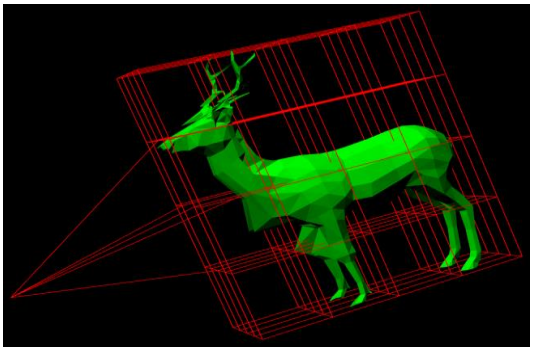

图 2 三次贝塞尔曲面形变

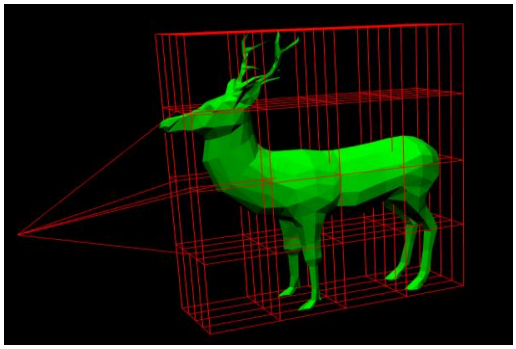

图 3 三次 B 样条曲面形变

由上图，可以发现贝塞尔曲面存在以下缺点：

（1）其不能表示局部变化，当一个控制点发生变化时，整个贝塞尔曲面都发生变化。

（2）贝塞尔曲面的拼接比较复杂。图 2 展示了使用贝塞尔曲面时，拖动脖子的控制点，脖子附近没有太大的形变，但是头部，膝盖都发生了形变。

同时，由于三次 B 样条曲面具有适用于局部形变的优点，我们在具体的项目开发中侧重于三次 B 样条曲面的形变。

2. VTK

VTK（visualization toolkit）是一个开源的免费软件系统，主要用于三维计算机图形学、图像处理和可视化。VTK 是在面向对象原理的基础上设计和实现的，它的内核是用 C++构建的，还包含有几个转换界面，因此也可以自由的通过 Java 和 Python 各种语言使用 VTK。

VTK 的开发流程如下展示：

- Source：读取数据;
- Filter：将数据对象变成图形数据;
- Mapper：指定了渲染数据和图形库中基本图元;之间的关系。
- Actor：在场景中表现一个可视化实体;
- Renderer:舞台背景,接受 actor 的信息;
- RendererWindow：窗体,可以存放多个舞台;
- RendererWindowInteractor:用于场景交互;

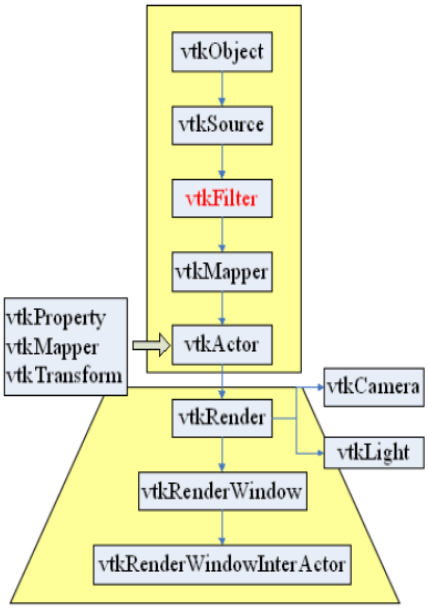

图 4　VTK 开发流程

综合考虑 VTK 的功能和优点，我们最终选择使用 Python 语言调用 VTK 来开发本项目。

2.2 实施步骤

根据上述算法描述，在项目开发中我们按照如下的实施步骤构建模型：

1. 建立局部坐标系

（1）确定控制点位置和数量：首先需要根据读入文件的坐标位置，确定控制点的位置，我们在目标物体的周围生成 N*N 的立方体，立方体的长宽高由物体大小确定，使得该立方体恰好包住目标物体。

（2）计算邻居结点：对于每一个控制点，计算以它为中心的前后上下左右的六个邻居结点的位置。

（3）对邻居结点连线：将每个控制点与他们所对应的邻居结点用连接起来。

2. 重设控制点

（1）控制点变化：人工拖动控制点，使其位置变化。

（2）查询控制点位置：对每个控制点的位置进行查询。

（3）重新连线：如果控制点位置发生更新，则去掉该控制点与邻居结点连接的“旧线”，并重新生成该控制点与邻居结点的连线

（4）返回新位置：把该控制点的新位置返回给 FFD 算法进行计算形变。

3. 形变函数计算

获取 FFD 算法计算好后的数据，去除之前显示的物体并用显示更新形变后的物体。

4. 设计可视化界面

调用 pyqt5 进行 GUI 的开发与设计。

2.3 功能展示

1. GUI 功能

我们导入一个 iPhone6 的 obj 文件，作为 GUI 的封面，在 GUI 内菜单栏和工具栏分别能实现的功能如下展示。

菜单栏：

- Change_obj_background 选择导入 obj 文件变形物体
- Add_FFD_file 导入 FFD 偏移量文件
- To_FFD_file 导出生成的 FFD 文件
- To_Bezier 进行贝塞尔变换
- Turn_on_light 加入灯光照射

工具栏:

- 一键复原
- 一键退出

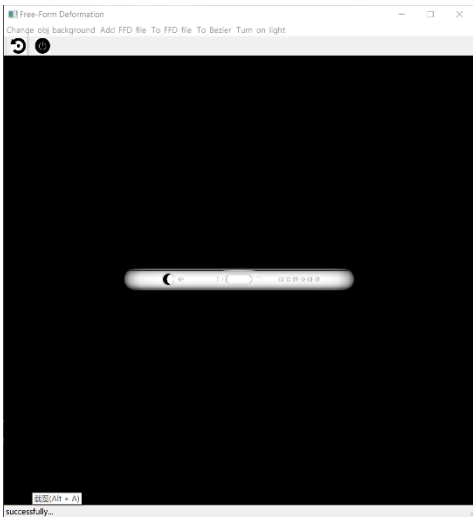

图 5　GUI 封面

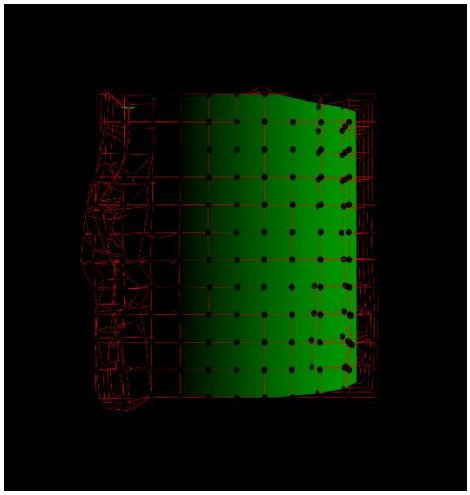

图 6　聚光灯效果

同时，　我们采用 vtk 的移动摄像头的交互方式，使用户点击鼠标左键并拖动时从另外角度观察物体。

在具体的开发中，我们发现在导入文件时，由于文件较大，如果误选会造成长时间不必要的卡顿，因此添加了一个判别文件类型的功能。如果导入文件类型与选择的类型不符，会弹出提示框，提醒用户重新选择。

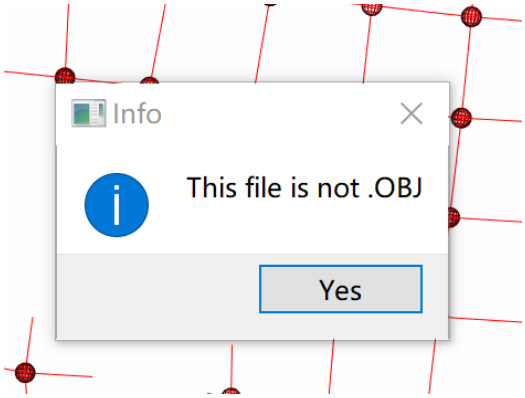

图 7 误选文件弹出提示框

2. 形变效果

由于不同的基函数对应不同的形变函数，我们分别进行了贝塞尔函数和 B 样条函数的形变，效果如下展示。

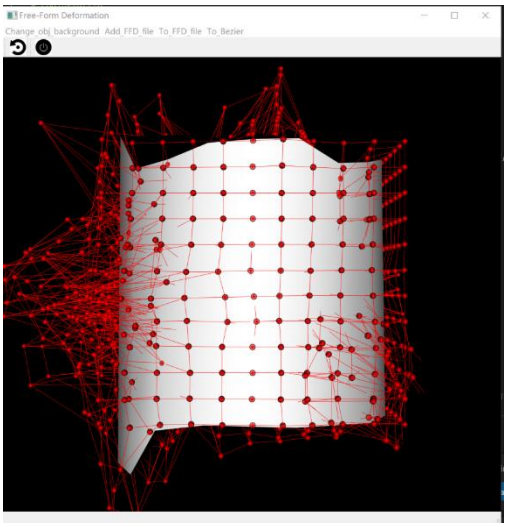

图 8 贝塞尔函数形变

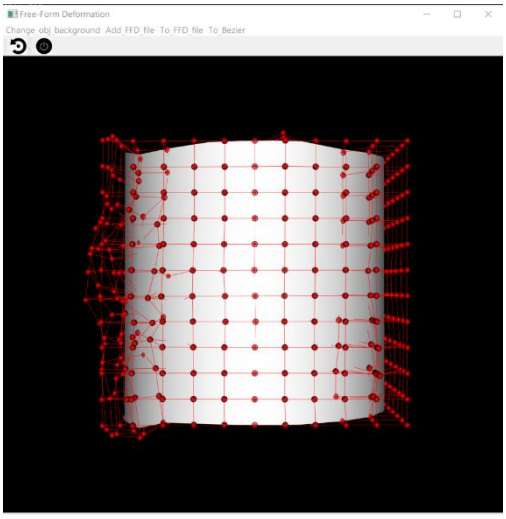

图 9 B 样条函数形变

从两种函数形变的正面展示就可以开到控制点排列上的显著差异，B 样条函数形变后的控制点更为规则，而贝塞尔函数形变导致控制点更为杂乱。这一点在俯视图中有更明显的展示。

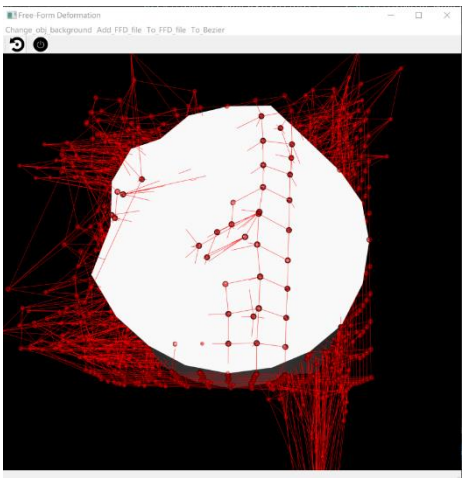

图 10 贝塞尔函数形变

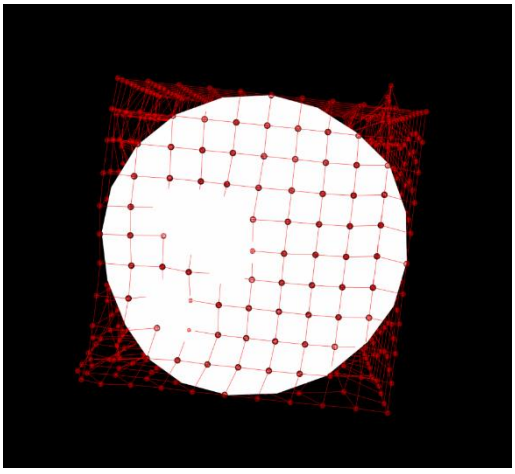

图 11 B 样条函数形变

3. 代码结构

GUI 结构:

Class GUI()

def menus()定义界面的菜单栏和工具栏

def add_obj()用户可导入 obj 文件作为背景

def add_ffd()用户可导入 ffd 文件条件控制点偏移量

def load_control_size()针对导入的 ffd 文件读取三维度上的控制点个数

def to_ffd()导出控制点改变后的 ffd 文件

def initial()复原为初始化界面

def show_obj()显示导入的 obj 文件

def show_vtk()生成用户可移动控制点的界面

调用 ffd_B 与 ffd_Bezier 进行 B 样条函数或贝塞尔曲线形变

def neighbor()找到每个控制点邻近的六个控制点

def sphereCallback()对移动后的控制点位移进行保存，重新进行 ffd 偏移量算法计算

def show_control_points()显示控制点与控制点间的连线

FFD 结构:

Class ffd()

def read_ffd()读取 ffd 文件，因为 vtk 不自带 ffd 读取语句，因此按行读入进行处理

def cover_obj()根据导入的 obj 物体大小生成能将其正好包围的立方体控制

点网格，为了便于显示偏移变形，根据控制点间距*ffd 定义的偏移量来计算偏移度

def B_func()实现 B 样条函数算法

4. 开发环境

系统环境: Windows 10

开发语言: Python>=3.6

安装包: numpy>=1.15.2， tensorFlow>= 1.12.0， vtk=8.1.2，pyqt5>= 5.11.3

5. 可执行文件使用手册

操作环境： Windows 10（也可以适应 Windows 7 版本）

操作文件： FFD.exe

操作简介:

1. 打开 FFD.exe 文件。

2. 在 GUI 中点击选择导入 FFD 文件，选择合适的 FFD 文件导入。

3. 点击鼠标左键并拖动，可以从另外角度观察物体。

4. 人工拖动控制点变化，观察变化。

5. 在 GUI 中点击导出 FFD 文件，以 FFD 格式导出变化后的文件。

6. 小组分工

姓　名	工作内容
程文宣	PyQt5，　VTK，PPT 制作，撰写报告
蔡睿杰	VTK 代码编写，PPT 制作，撰写报告
杨嘉溢	FFD 算法、PPT 制作，撰写报告

7. 参考文献

【1】VTK User's Guide 来源：VTK 官网

【2】3D-FFD-IN-VTK　来源：GitHub　作者：Anthony

【3】BezierMaker　来源：GitHub　作者：Venshine