第一章 绪论

1.1 研究背景

1.2 等几何拓扑优化简介

1.3 三维可视化图形工具OpenGL概述

1.4 本文的主要研究内容

1.5 本章小结

第二章 基于变密度法的等几何拓扑优化方法

2.1 引言

2.2 B样条与NURBS理论

2.3 变密度法的等几何拓扑优化理论

2.4 本章小结

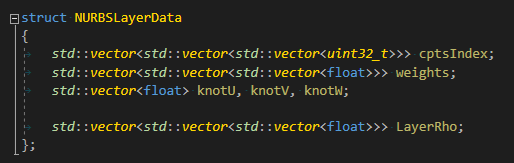
第二章主要说明拓扑优化的理论，这一部分为后续可视化做了一个数据上的支撑。重点特别在于第二小节中的B样条与NURBS理论部分，在后续第三章计算NURBS面片的时候需要这一方面的理论知识作为支撑。即，第二小节主要为后续计算几何轮廓提供理论支撑，第三小节主要是为后续显示单元密度值而给出计算密度值的方法。

第三章 三维情形下的等几何拓扑优化可视化研究

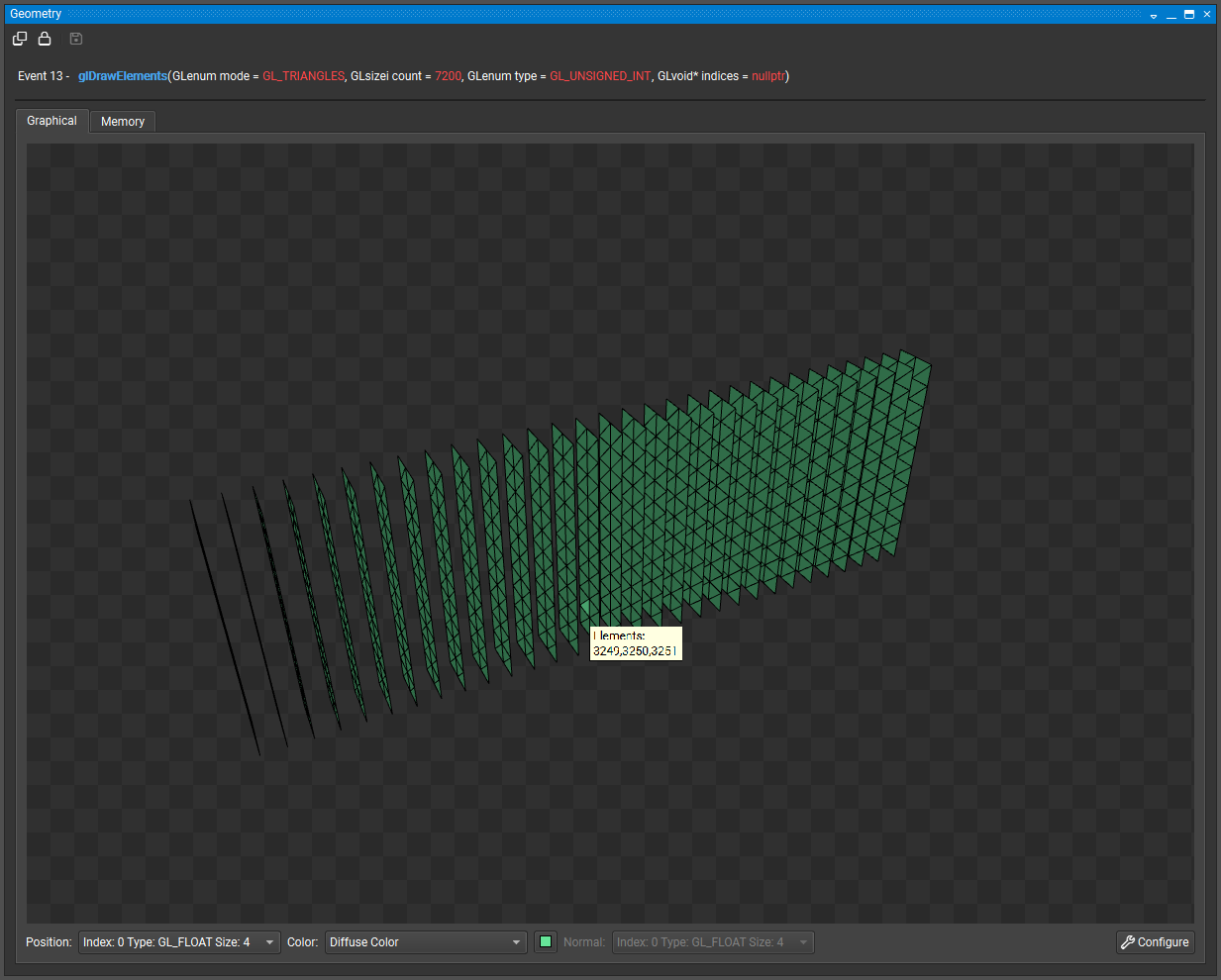
3.1 引言

3.2 图形学基础（这一部分还是否需要存疑）

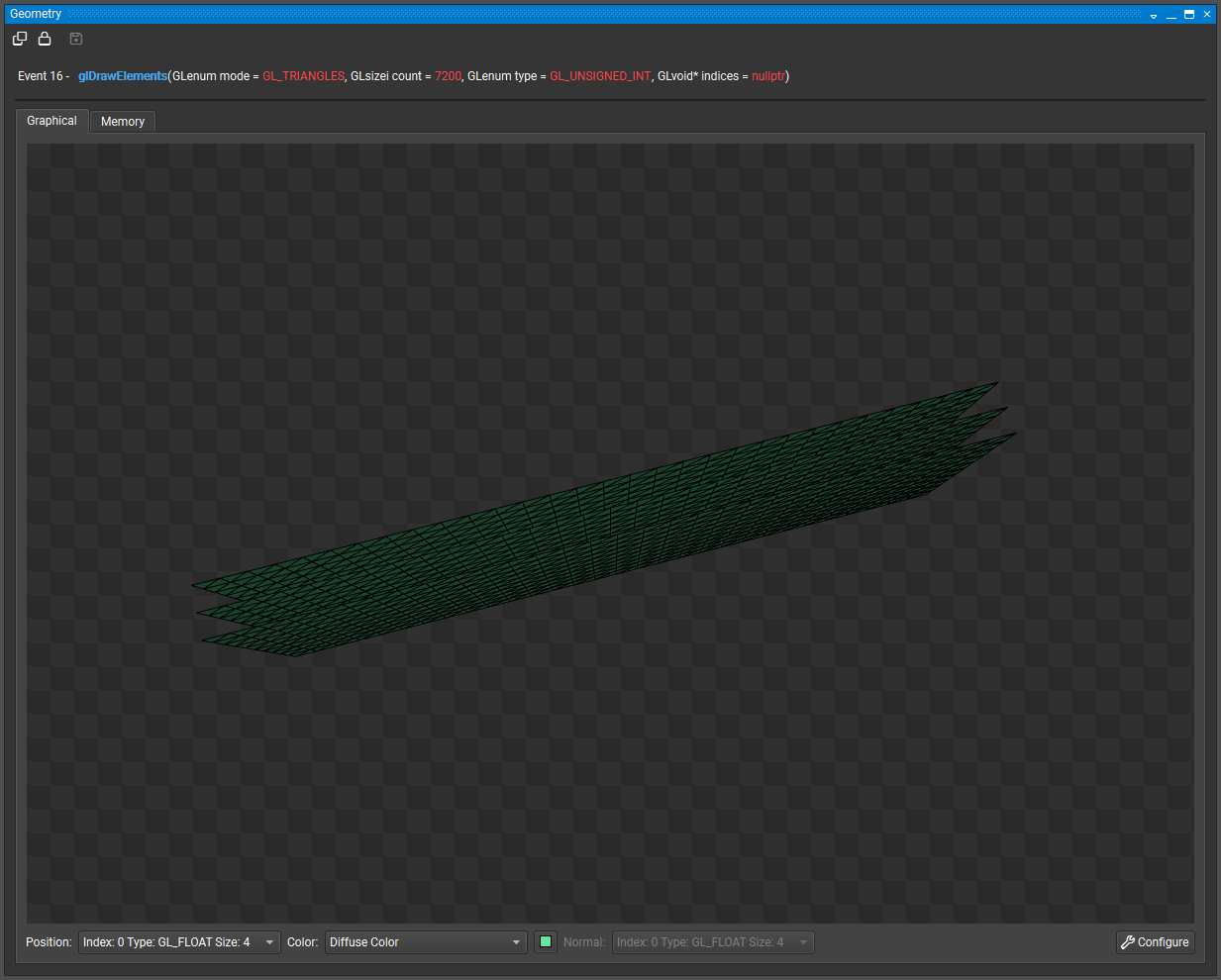
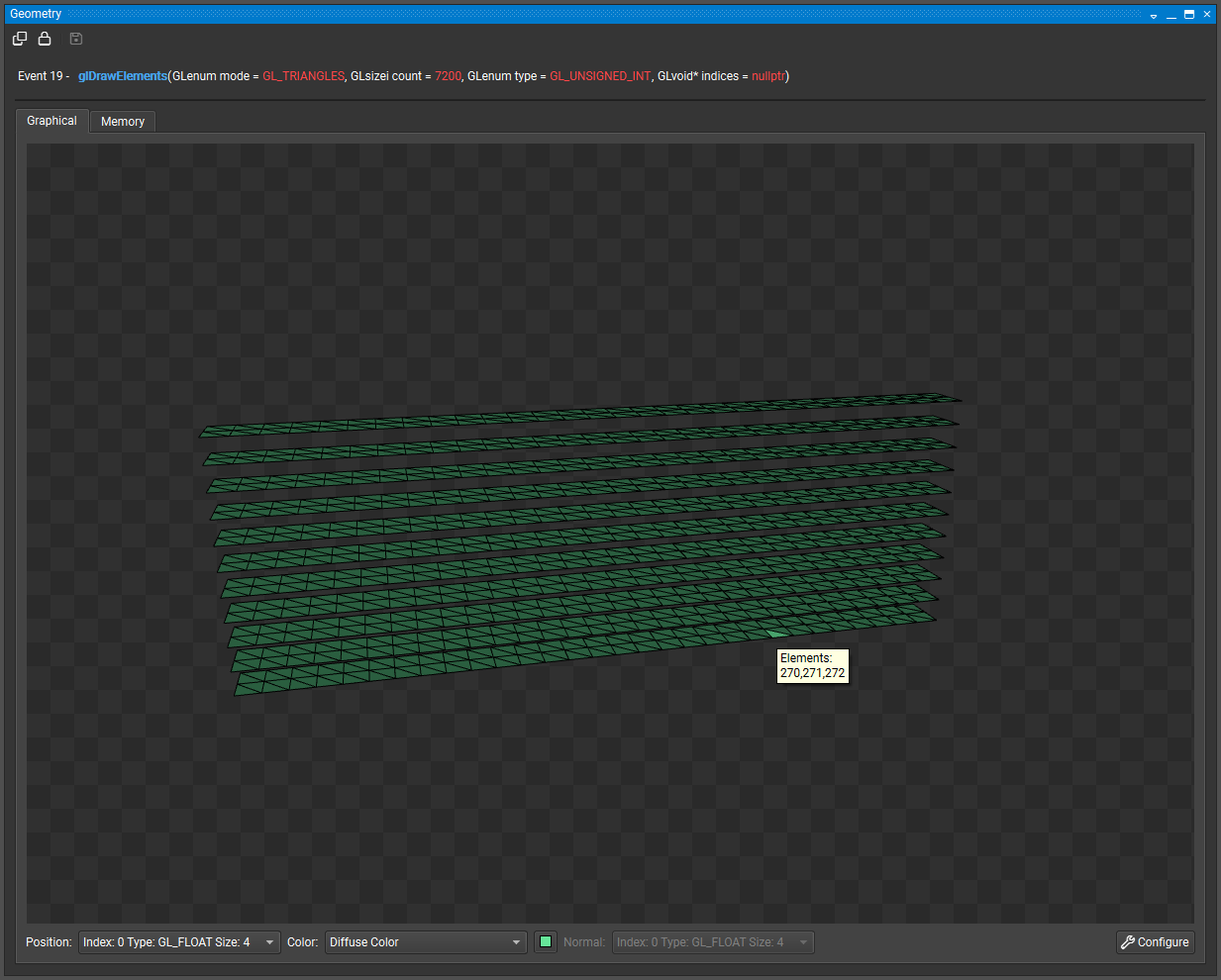
3.3 等几何模型表达（模型的控制点与变形方法）



该结构体存储的是每一个方向上（后面有图详细描述）的所有控制点、权重、节点向量和单元密度值。这里说的“每一个方向”，如下图所示



这里显示的所有面都垂直于x轴，相当于在x轴方向上每隔一段距离计算一个NURBS面。余下两个方向同理，分别垂直于y轴和z轴。由这三组NURBS面构成了整个网格。



垂直于y轴的面和垂直于z轴的面

当时这么考虑的原因是之前试验过如果每次调用计算着色器只得到一个单元的话，将会导致非常多的调用次数（因为NURBS单元的数量可能非常多），而每次计算着色器的调用将对应一次draw call，会极度影响效率。这里设计程序的时候让程序只调用了三次draw call，所以相较于数万次、数十万次的draw call会更加高效。

但是这样实现导致了一个问题，就是如果要让被单元密度阈值排除掉的面片不显示，可能会比较困难（4/17晚上回来试了一下，还没有跑出想要的结果），这一方面需要多做考虑。

3.4 有限元与等几何拓扑优化可视化差异

这一小节的话不如就把怎么具体实现NURBS面片算法的具体流程讲一下（这一部分和后面计算着色器的部分重合了，所以直接移到这里来讲比较好）。标题名字要改掉，有限元与等几何拓扑优化可视化差异也要讲到，但是最好在这一章开头一笔带过。

3.5 本章小结

第四章 基于OpenGL等几何拓扑优化可视化系统搭建

4.1 引言

4.2 Win32与OpenGL程序框架

（这一节干脆不要子标题了，就“4.2 OpenGL程序框架”然后在这下面把所有的OpenGL的框架和Win32的框架写清楚,而且就讲C++的部分）

4.2.1 OpenGL程序框架

4.2.2 win32程序下的OpenGL绘图环境

4.2.3 计算着色器（考虑移到3.4节）

（这几个标题都不要了，计算着色器移到3.4）

4.3 基于OpenGL的可视化系统开发

4.3.1 界面设计

4.3.2 数据的产生与读取

4.3.3 可视化功能的实现（这一节就把可以动态地显示拓扑优化迭代过程的那个功能写一下，不然直接去掉这一节，没有多少内容可以写，如果写变换那些东西太基础，写可视化实现的话又会和前面重复。）

4.4 本章小结

第五章 结论与展望