



毕业设计(论文)开题报告

题目： 基于 OpenGL 的真实感水面渲染系统

院（系） 计算机科学与工程学院

专 业 计算机科学与技术

班 级 16060101

姓 名 余硕

学 号 15060101120

导 师 傅妍芳

年 月 日

开题报告填写要求

1. 开题报告作为毕业设计（论文）答辩委员会对学生答辩资格审查的依据材料之一。
此报告应在指导教师指导下，由学生在毕业设计（论文）工作前期内完成。
2. 开题报告内容必须按教务处统一设计的电子文档标准格式（可从教务处网页上下载）
填写并打印（禁止打印在其它纸上后剪贴），完成后应及时交给指导教师审阅。
3. 开题报告字数应在 1500 字以上，参考文献应不少于 15 篇（不包括辞典、手册，其中外文文献至少 3 篇），文中引用参考文献处应标出文献序号，“参考文献”应按附件中《参考文献“注释格式”》的要求书写。
4. 年、月、日的日期一律用阿拉伯数字书写，例：“2005 年 11 月 26 日”。

1. 毕业设计（论文）综述（题目背景、研究意义及国内外相关研究情况）

1.1 题目背景及研究意义：

由于人们对于动画真实感的需求越来越强烈，在大多数的特效电影和视频游戏中，场景中水体的真实感效果会极大程度影响人们的视觉感受和游戏体验，而且涉及相关技术的计算机图形学是计算机学科下数字媒体技术的核心，基于真实场景的水面动画模拟正逐渐成为计算机图形学研究领域的一个重点方向。《GPU 精粹》简介中描述到：水面的动画与光照在计算机图形学中属于最难的工作。

无论是游戏开发者还是电影创作人，渲染不同种类型的水体是渲染三维场景不可缺少的部分，比如池塘、溪流、湖面、海洋等，水体的表面不但要有接近真实场景的色彩值、光照效果和高低起伏等诸多特征，还要能够随着时间的推移而不断变化和波动。无论是波函数的合成，还是利用 GPU 通过傅里叶变换加速计算出水面高度，都是实时渲染水面的有效方法，能够以保证计算快速且实时的同时追求更加逼真的水面渲染效果。

高度还原现实中场景是计算机图形学进行模拟水面最基本和最重要的要求，比如水面的模拟要有随时间的变化而波动的效果。随着多媒体编程接口和 GPU 硬件的发展，模拟水面的过程能变得更加高效、快速，模拟出的水面也更加真实。

1.2 国内外相关研究情况：

近几年，实时渲染技术渐渐从离线渲染的世界脱离出来。正如 2001 年 Tessendorf 所述的那样，快速傅里叶变换（FFT）技术在大尺寸取样栅格中产生了难以置信的逼真效果；在消费级的 PC 上，对中等大小的取样栅格也可以进行实时处理。还能够基于体素（Voxel）对简化的 Navier-Stokes 方程求解（Yuan 2003）。虽然尚未达到 Enright 等人 2002 年创建的离线流体模拟那样的尖端水平，但是差距正在缩小。如今，傅里叶变换（FFT）库将能适用于顶点 shader 和像素 shader，并且正在朝着抛开 CPU，在 GPU 上独立执行而前进。

与此同时，在 GPU 上运行的水面模拟模型也得到了改进。Isidoro 等人 2002 年提出在一个顶点着色器中加和 4 个正弦波以计算水面的高度和方位。Laeuchli 在 2002 年发表了一个 shader，使用三个 Gerstner 波计算水面高度。

在近年的电子游戏中，GPU 模拟的水面渲染系统已经大量适用，如 Uru: Age Beyond Myst（2011）、The Elder Scrolls V: Skyrim（2011-2016）等，证明了把基本网格的几何波动与动态法线贴图的水面模拟方法，适用于各种类型下模

拟复杂场景的实时交互游戏及应用。

2. 本课题研究的主要内容和拟采用的研究方案、研究方法或措施

2.1 研究内容：

常用的水体模拟的方法可分为以下三类：

（1）基于物理粒子的水体模拟

本方法将连续的介质模拟成离散的系统，并基于流体动力学方程来进行空间插值模拟流体动画，多适用于需要水体的流体物理特性的实际应用中，由于其整体离散的特性，不易对整个系统中的单独一部分进行追踪。

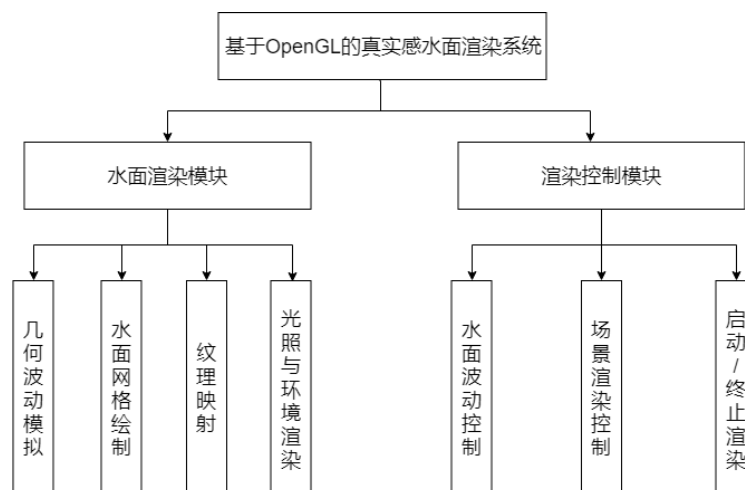
（2）基于替换纹理的水体模拟

本方法先对渲染对象表面纹理进行不断扰动，然后模拟表面及其表面下的各类反射，即可得到较好的表现效果。不过由于并没有实际的顶点高度变化，构成其本体的网格、几何体并不能模拟出实际的流体波动效果。因为不需大量的计算就可得到满足视觉、性能需求的效果，视频游戏中一般采用了此种方法。

（3）基于几何波叠加的水体模拟

本方法先生成一个基础平面网格，再与生成平面的垂直方向叠加与真实水体波动效果波形类似的几何波，最后叠加而成的网格中所有不同高度值的顶点就表示出了整个水体表面。基于几何波叠加的水体模拟方法不消耗过多性能的同时生成了具有一定实际中水体物理几何特性的逼真水面效果。

本系统综合以上三种方法的优点，并划分为以下两大模块：



模块的具体说明如下：

水面渲染模块：通过 OpenGL 拓展库完成渲染系统的搭建，包括：计算并得到模拟真实水面波动效果的数据；利用 GPU 绘制出基础水面网格；在可编程的着色器中完成纹理映射，模拟出接近真实色彩值的水面效果；添加光照与环境场景

效果，美化水面渲染场景。

渲染控制模块：通过 QT 平台实现对系统渲染功能的控制，包括：使用滑动条控制水面从平缓到剧烈的波动；使用复选框开启/关闭场景中的各类渲染效果（如漫反射光、环境光等）；使用 QT 平台的 QProcess 类与多线程技术，实现两个模块间命令行的传递，并完成对系统渲染功能的开启与控制。

2.2 研究方法与措施：

本系统综合基于物理粒子、替换纹理、几何波叠加三种方法，研究得到下列研究措施：

- (1) 选择 OpenGL3.3 版本、GLEW 库(对各类操作系统的支持)、GLFW 库（窗口的创建与操作）、以及 Qtcreator、VS2015 等 IDE，完成开发环境的配置。
- (2) 使用 OpenGL 中的 API 完成渲染系统的底层接口，包括 TGA 格式图片导入接口、shader 读取接口、shader program 创建接口、Windows 平台窗口创建接口、性能计算接口等。
- (3) 使用（2）中完成的一系列接口，并编写 vertex shader, 按照 GPU 显存的格式读入 x-y 平面顶点数据, 并从对象坐标系转换到屏幕空间坐标系下，完成基础网格的绘制，得到一个平整的 x-y 平面网格。
- (4) 使用几何波叠加方法中的算法，叠加多个二维平面上的模板几何波（具有波形代表性的简单波形），得到水面几何波动的原始数据，再依照基于物理粒子方法中的空间插值算法，将小范围内的几何波顶点数据扩大数倍，读入到渲染系统主程序中。
- (5) 设计算法将（4）中得到的二维波转换到三维平面的 Z-XY 平面，并编写新接口将三维波叠加到平面网格，得到具有真实几何特性的水面模拟网格。
- (6) 依据替换纹理中的算法把每帧经过时间 (Δt) 作为自变量，纹理坐标的坐标值作为因变量，编写 shader, 得到随时间变换的纹理映射坐标。
- (7) 编写纹理映射接口，使用 TGA 格式图片导入接口将绘制好的漫反射贴图、法线贴图转换到对应的纹理坐标系（法线与切线）下，绑定贴图，完成纹理映射，得到具有真实色彩以及更加立体、细致的水面效果。
- (8) 依据冯氏标准光照模型，编写环境光、漫反射光、高光等光照模型参数，并依据兰博特定律、贝利定律编写 fragment shader, 完成水面渲

染光照的添加。

- (9) 使用绘制的立方体纹理,编写天空盒对应的 vertex、geometry shader,完成对渲染场景的天空盒包裹;使用预先制备的 OBJ 文件,编写 OBJ 导入接口,为场景添加模型文件,美化渲染效果。
- (10) 使用 Qtcreator 设计渲染控制界面,使用 QWidget、QSlider 等类绑定控制渲染主程序渲染效果的命令行到对应 GUI 元素上,并使用 QProcess 类下的外部程序控制接口,实现对渲染模块的控制。
- (11) 不断调整控制模块传递命令行参数的阈值,直到得到效果明显的渲染控制效果;不断调整 shader 以及光照经验模型,直到得到满意的水面渲染效果。

3. 本课题研究的重点及难点,前期已开展工作

3.1 重点:

- (1)基于 OpenGL 相关 API 的使用。
- (2)利用 OpenGL 的 API 与真实感渲染理论实现渲染系统下所需的各个底层接口(图片、模型读入,shader 读取、绑定等)。
- (3)依据几何特性(相位、周期)以及物理特性(波形)挑选适合的几何波并叠加到平面网格(如正弦波、G 波等)。
- (4)掌握图形学标准光照模型的相关理论算法,并对应系统中纹理坐标、顶点坐标系的选择,使用 3D 数学下坐标转换的算法,编写 shader。
- (5)熟悉 Windows 平台的命令行,熟悉 Qtcreator 的使用,使用相关类完成可对水面起伏程度、场景效果调整的功能,并绑定到图形界面的元素上,完成渲染控制模块。

3.2 难点:

掌握计算机图形学 Rendering Pipeline 相关理论,了解 GPU 的多数据流、多核心的特殊硬件结构,以完成渲染系统下相关底层接口的实现;依据 3D 数学知识设计几何波转换维度的算法;几何波叠加中对模板波的选择、参数的控制,以得到可模拟真实水面波动效果的波形;凹凸纹理贴图(法线贴图)中相关颜色理论的学习以及其绘制,以得到可模拟水面细微起伏的纹理;光照经验模型的不断调整与对比,以得到真实的水面渲染效果。

3.3 前期已展开的工作:

- (1)前期相关知识理论的学习调研。

(2) 软件平台的安装与熟悉。

(3) 在网上、图书馆查阅相关资料，进行项目分析。

4. 完成本课题的工作方案及进度计划（按周次填写）

第一周——第三周：准备毕业设计材料，进行毕设选题、理解题目、查阅资料，编写开题报告。

第四周——第六周：结合题目查阅相关资料，学习 OpenGL，熟悉开发环境，进行技术选型。

第七周——第十周：编码测试阶段，分步完成各个模块，画出流程图和系统结构图，并进行单元测试。

第十一周——第十三周：模块的整合、对模块完善，对界面优化。

第十四周——第十六周：对系统集成测试，由导师验收。

第十七周——第十八周：写毕业论文并准备毕业答辩。

注：1、正文：宋体小四号字，行距 22 磅。

2、开题报告装订入毕业设计（论文）附件册。

参考文献

- [1] 邹清. 基于物理模型的真实感水波模拟[D]. 中南大学, 2007
- [2] Randima Fernando. GPU 精粹[M]. 北京:人民邮电出版社, 2006, 8-12
- [3] 孟晓宁, 王宝华. 图形处理中 GPU 固定渲染管线研究[J]. 集成电路应用, 2018, 35(02): 90-94
- [4] 约翰 克赛尼希. OpenGL 编程指南[M]. 北京:机械工业出版社, 2010, 147-168
- [5] 彭群生. 计算机真实感图形的算法基础[M]. 北京:科学出版社, 2009, 31-34
- [6] 周飞. 法线贴图技术在次世代游戏图像中的原理及应用[J]. 湖北经济学报, 2012, 9(06): 40-41
- [7] 美贝利, 美坎宁安. 图形着色器[M]. 北京:清华大学出版社, 2013, 103-117
- [8] 张春艳. 基于 QT 的嵌入式图形用户研究与实现[D]. 大连:大连海事大学, 2008
- [9] 黄宇东, 胡跃明. 基于 QT 的多线程技术应用与研究[J]. 软件导刊, 2009, 40-42
- [10] Steve Marschner, Peter Shirley. Fundamentals of Computer Graphics[M]. 4. Boca Raton: CRC Press. 2016. 14-51
- [11] Wolfgang Engel. GPU Zen Advanced Rendering Techniques[M]. Encinitas: Black Cat Publishing Inc. 2017. 33-43
- [12] 冯乐乐. Unity Shader 入门精要[M]. 北京: 人民邮电出版社, 2016, 20-40
- [13] 朱晓飞, 万哲. 基于 OpenGL 的三维场景的模拟[J]. 电子技术与软件工程, 2014(03): 103
- [14] 张肇同. 基于 OpenGL 的三维模型读取与动态观察[J]. 科技视界, 2017(34): 4-5
- [15] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, Angelo Pesce, Michal Iwanicki, Sébastien Hillaire. Real-Time Rendering Fourth Edition[M]. Boca Raton: CRC Press. 2018. 12-27