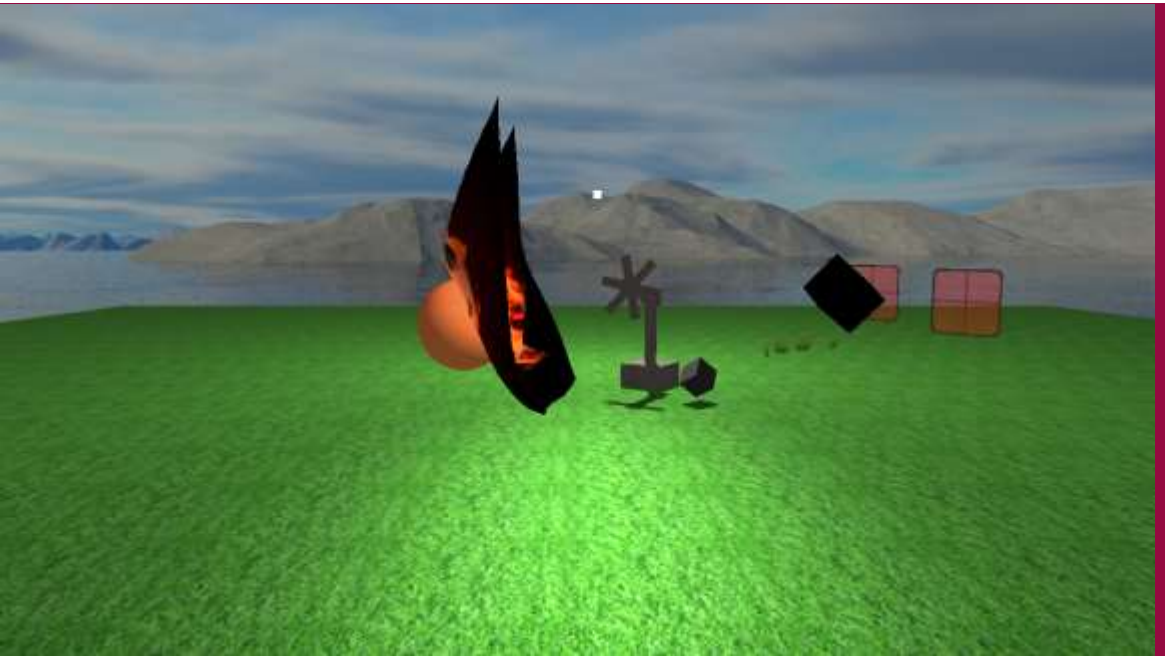
图形学期末大作业答辩

风车吹布料

2021192010 王曦

深圳大学 SHENZHEN UNIVERSITY

深圳大学
SHENZHEN UNIVERSITY



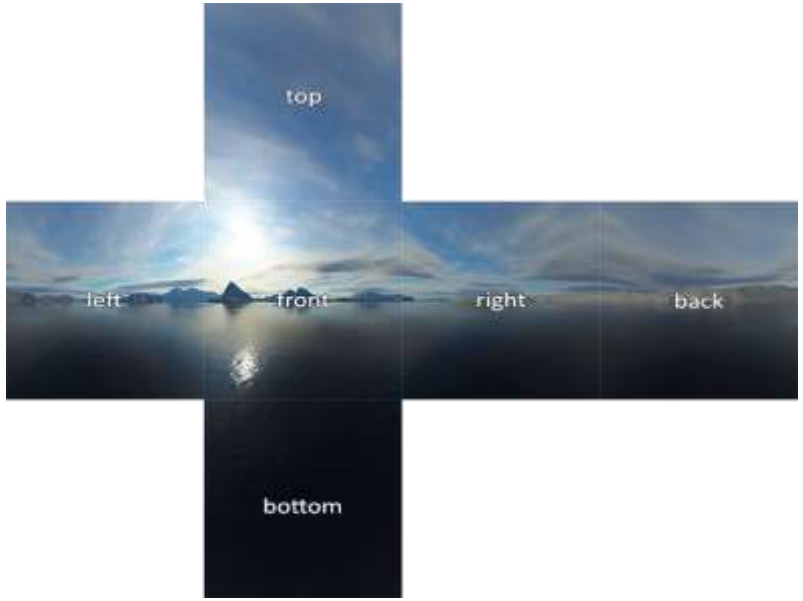
Part 1
场景介绍、透明材质

深圳大学
SHENZHEN UNIVERSITY

天空盒

正方体贴图

3D 纹理，可为正方体的每个面设置不同的纹理。



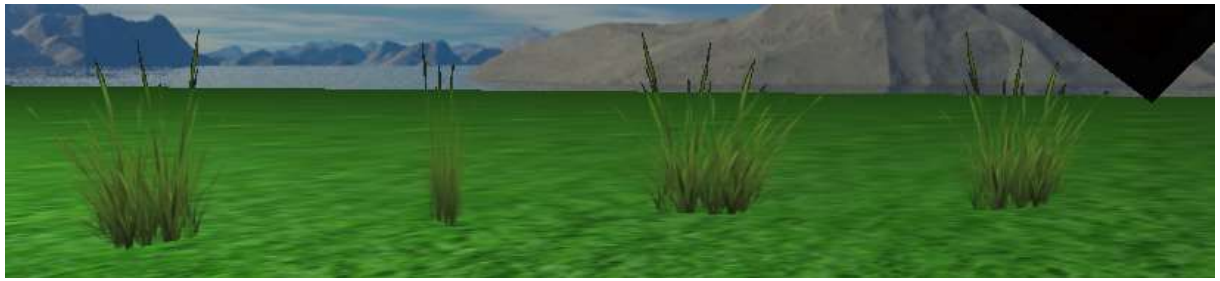
全透明材质

全透明材质

在 grass.fshader 中丢弃 alpha < 0.1 的片段。

防止边缘出现半透明边框

有 alpha 通道时将环绕模式设为 GL_CLAMP_TO_EDGE 。



环绕模式 GL_REPEAT

采样纹理边缘时，

OpenGL 会对边缘的值和纹理下一个重复的值进行插值

深圳大学 SHENZHEN UNIVERSITY

半透明材质

OpenGL 混合

Full transparent window

Partially transparent window

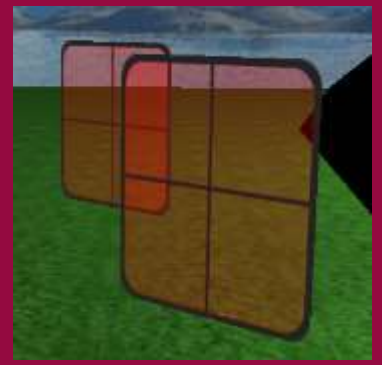
$$\vec{C}_{\text{result}} = \vec{C}_{\text{source}} \cdot F_{\text{source}} + \vec{C}_{\text{destination}} \cdot F_{\text{destination}}$$

窗户的混合函数：

GL_ONE_MINUS_SRC_ALPHA

透过半透明的窗户看物体（含全透明的物体、天空盒）

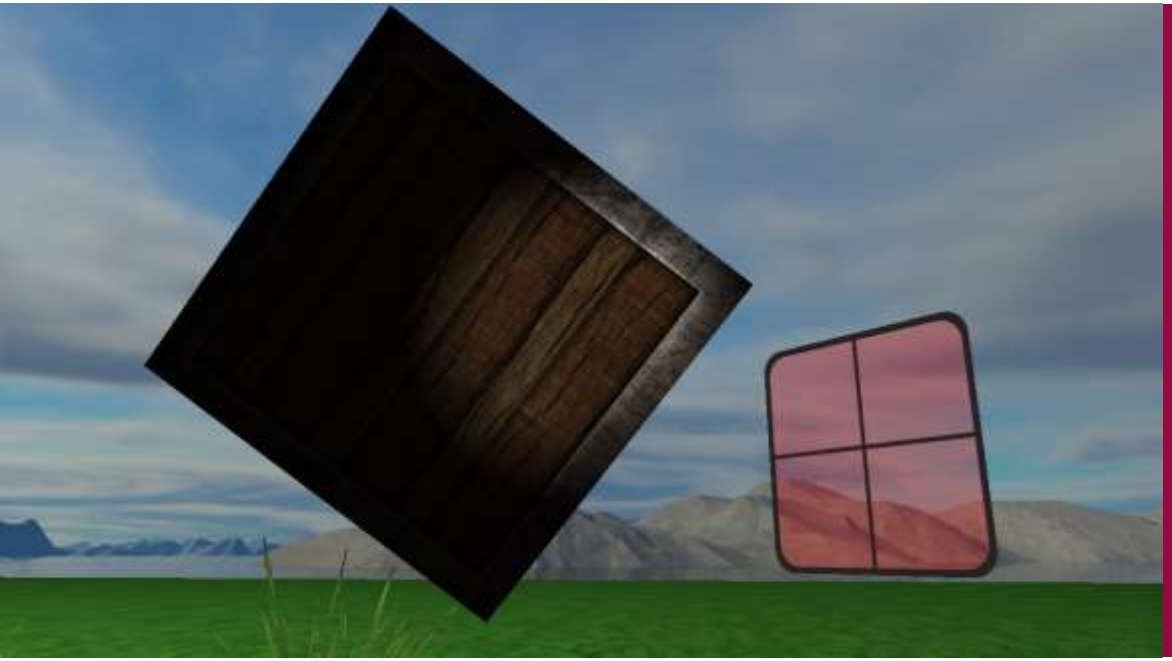
天空盒在窗户之前绘制。



透过半透明的窗户看半透明的窗户

按距离顺序绘制。

深圳大学
SHENZHEN UNIVERSITY

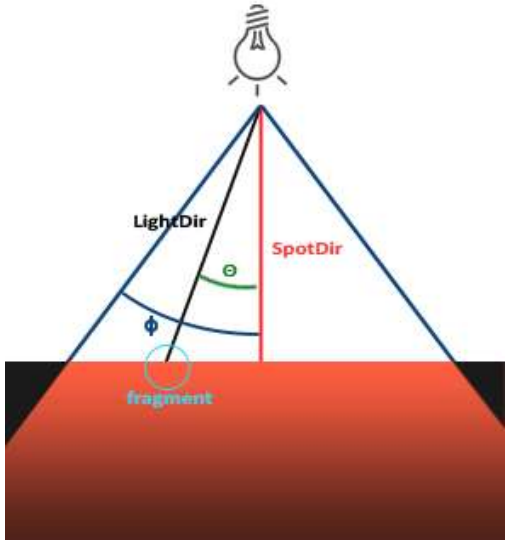


Part 2
聚光、镜面光贴图

深圳大学
SHENZHEN UNIVERSITY

聚光

聚光



软化边缘

内外圆锥

- 内圆锥中：直接计算
- 内外圆锥间：渐弱
- 外圆锥外：黑暗

关闭聚光

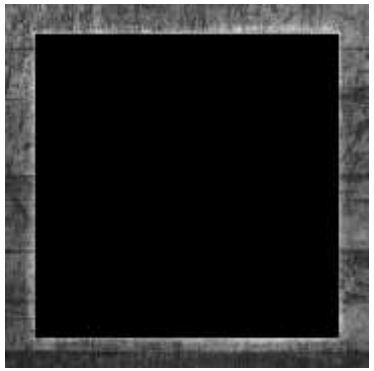
开启聚光

深圳大学
SHENZHEN UNIVERSITY

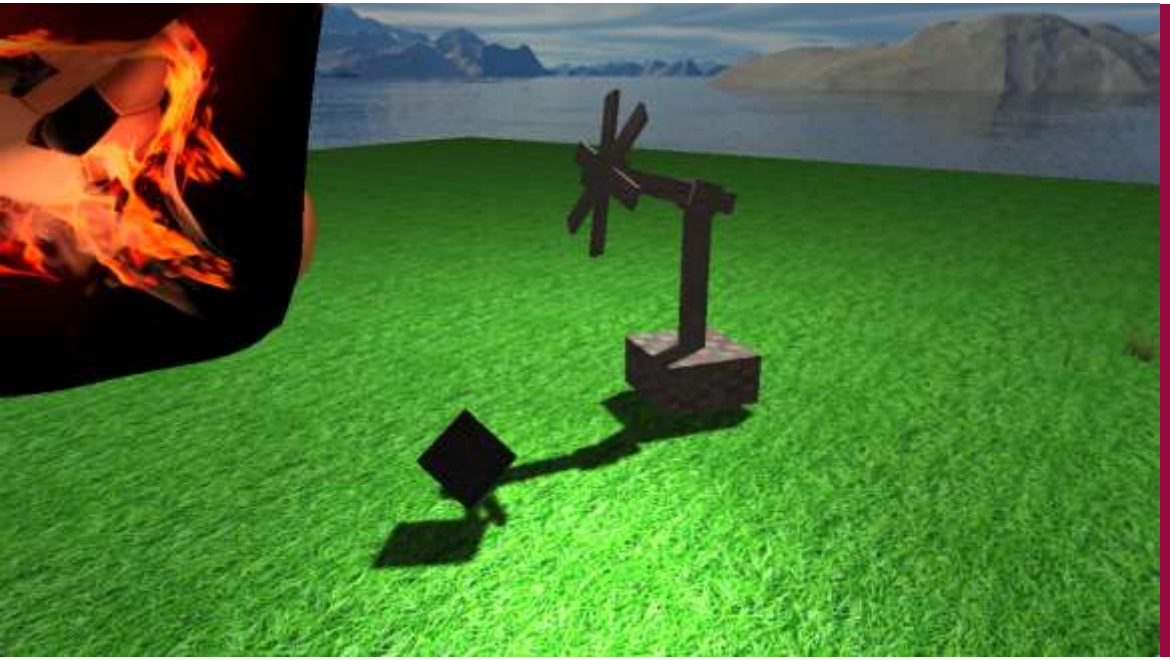
光照贴图

漫反射贴图

镜面光贴图



深圳大学
SHENZHEN UNIVERSITY



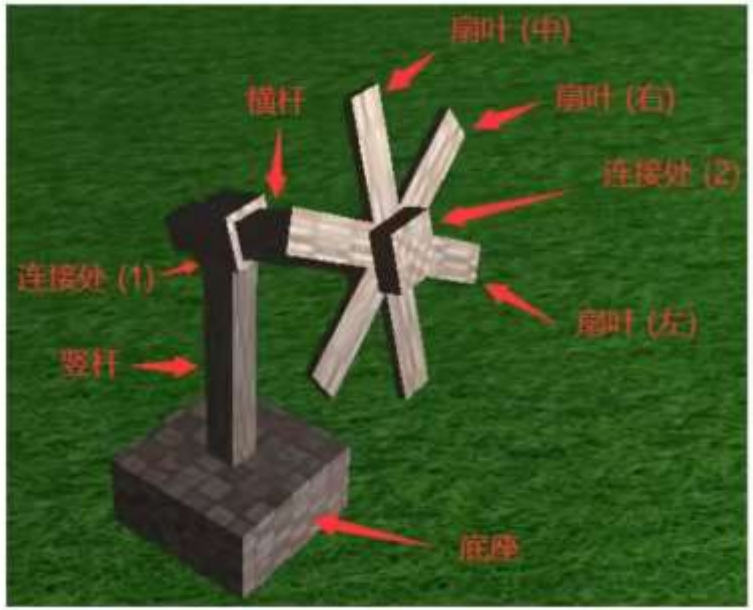
Part 3、4、5
层级建模、阴影映射、风车旋转、布料仿真 (1)

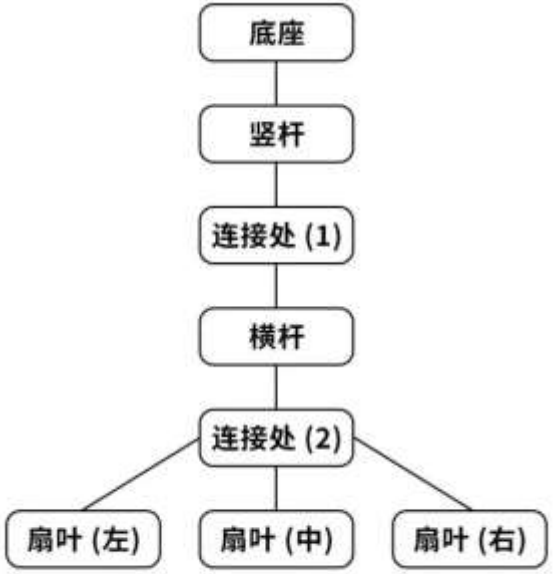
深圳大学 SHENZHEN UNIVERSITY

层级建模

层级建模

6 层





风车位置移动、视角变换

仿照摄像机类

风车旋转

两种实现：

- 施加一个旋转矩阵
- 改变绘制时扇叶的初始角度

风车吹风

* 给布料施加一个对应方向的力，

  力的大小由风车转速决定

* 风车朝向判定：

  求直线与平面的交点，判断交点在三角面片中，

  判断向量同向（特判除以 0 ）

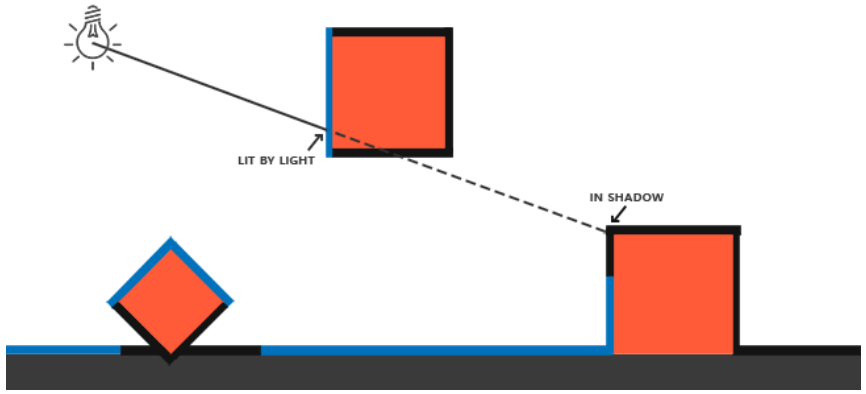
布料仿真

稍后介绍 ...

深圳大学 SHENZHEN UNIVERSITY

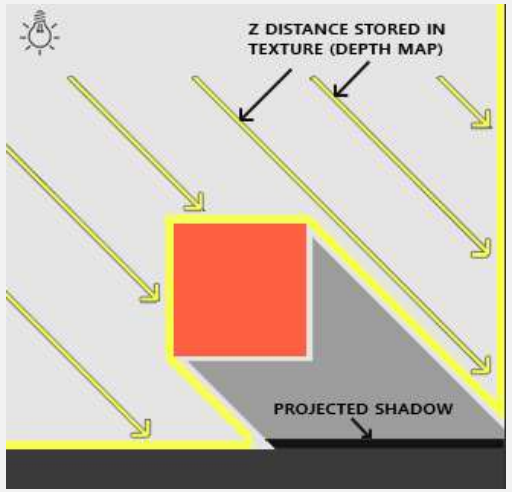
阴影映射

阴影映射

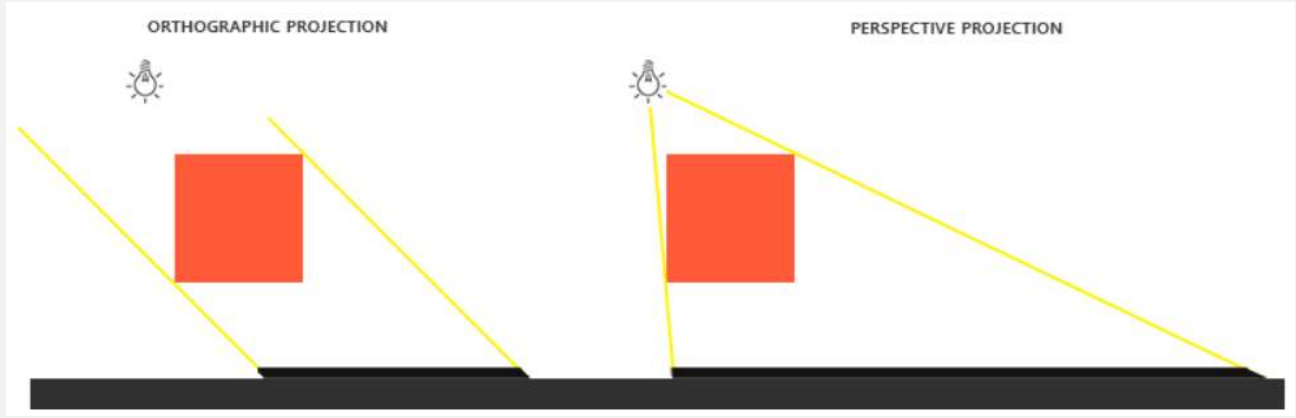


阴影映射的绘制步骤

1. 从光的视角渲染深度贴图
   - 渲染深度贴图：depth.vshader/fshader
   - 可视化深度贴图：debug_depth.vshader/fshader
2. 渲染场景，从深度贴图中采样，判断片段是否在阴影中，绘制阴影
   - 渲染场景：colors.vshader/fshader



深度贴图示意图



正交投影与透视投影产生的阴影的区别

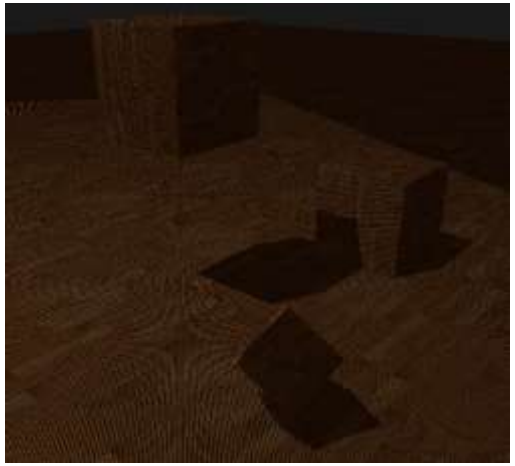
正交投影：定向光、平行光

透视投影：点光源、聚光灯

深圳大学
SHENZHEN UNIVERSITY

阴影偏移

阴影失真

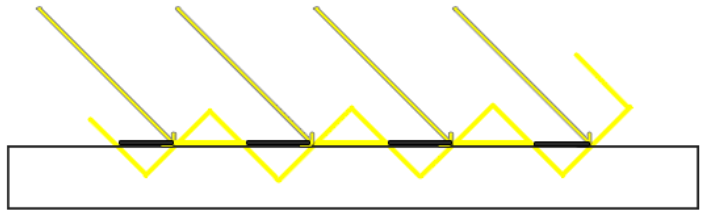


原因：

阴影贴图受分辨率限制，

在距光源较远的情况下，

多个片段从深度贴图的同一值采样。

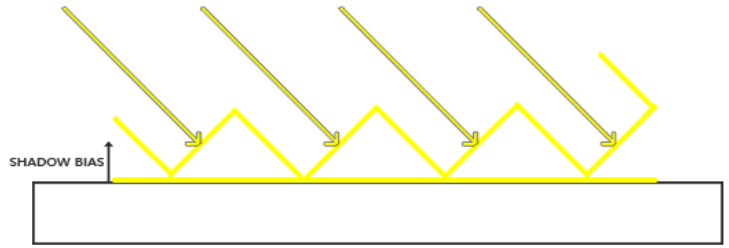


阴影偏移

解决：

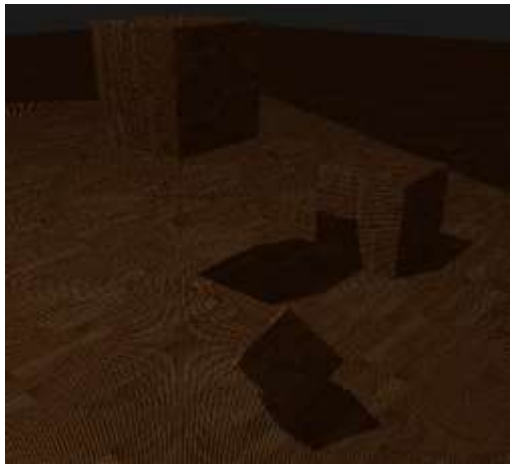
对表面深度（或深度贴图）

加一个偏移量。



深圳大学
SHENZHEN UNIVERSITY

采样过多



现象：

左上方的地面被错误地添加阴影。

原因：

阴影映射认为光的视锥不可见的区域都在阴影中，

此时若超出光的视锥的投影坐标大于 0.1，

则采样的深度纹理超出 [0, 1]的范围。

上述被误认为在阴影中的区域实际上是深度贴图的大小，

深度贴图影响地板是因为其环绕方式为 GL_REPEAT 。

解决：

让所有超出阴影贴图的坐标的深度为 1.0，

这样超出的部分被认为不在阴影中，

再将深度贴图的环绕方式设置为

GL_CLAMP_TO_BORDER，

并设置超出范围后的返回 1.0 的深度值。

现象：

右上方的地面被错误地添加阴影。

原因：

该区域中的坐标超出光的正交视锥的远平面，

此时其投影坐标的 z > 0.1， 环绕方式

GL_CLAMP_TO_BORDER 不起作用。

解决：

在 colors.fshader 中将 z > 0.1 的投影向量

对应的 shadow 值设为 0.0。

深圳大学
SHENZHEN UNIVERSITY

PCF

锯齿、硬边

原因：

阴影贴图受分辨率限制，

多个片段对应同一纹理像素时会

从深度贴图的同一深度采样。

PCF
(Percentage-closer Filtering)

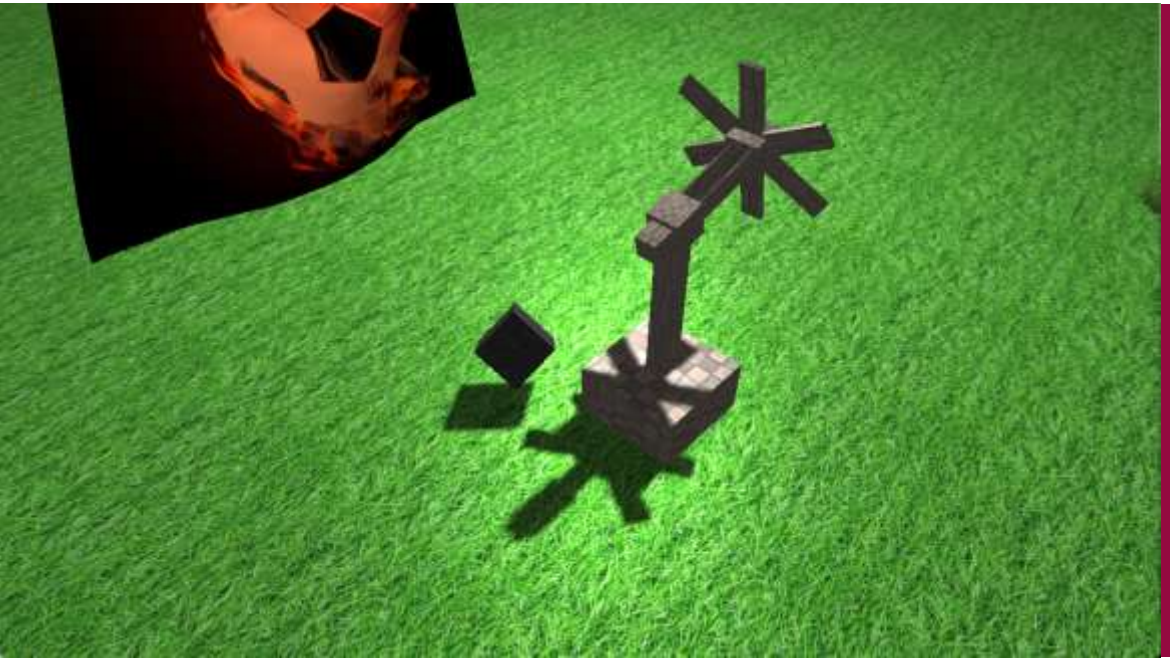
解决：

从深度贴图中多次采样，

每次采样的纹理坐标稍有不同，

结合多次采样的结果取平均。

深圳大学
SHENZHEN UNIVERSITY



Part 6 、7
Blinn-Phong 光照、Gamma 校正

高光断层

现象：

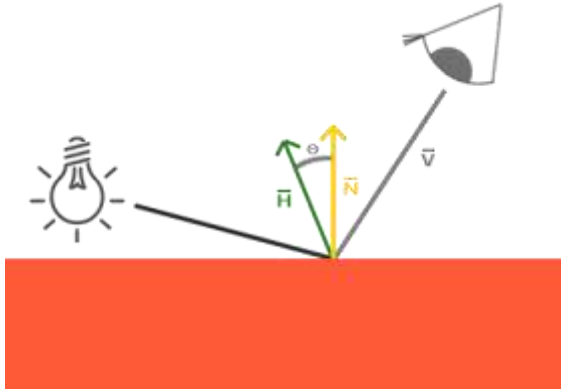
物体反光度较低时，出现高光断层。

原因：

Phong 光照依赖反射向量，观察向量和反射向量间的夹角 > 90° 时点积为负，镜面光分量变为 0.0 ，在镜面反射中会出问题。

解决：

Blinn-Phong 光照引入半程向量 $\vec{H} = \frac{\vec{L}+\vec{V}}{\left\|\vec{L}+\vec{V}\right\|}$ 。



视线与反射向量对齐时，半程向量与法线重合。

观察者视线越接近于原本反射光线的方向时，镜面高光增强。

无论观察者向哪个方向看，半程向量与表面法线间的夹角都不会超过90度（除非光源在表面以下）。

Blinn-Phong 光照更自然：

深圳大学 SHENZHEN UNIVERSITY

Gamma 校正

现象：

片段计算出的颜色、显示器显示的颜色、人眼看到的颜色有差别。

解决：

校正阴极射线管显示器（CRT）的Gamma值，一般取2.2。

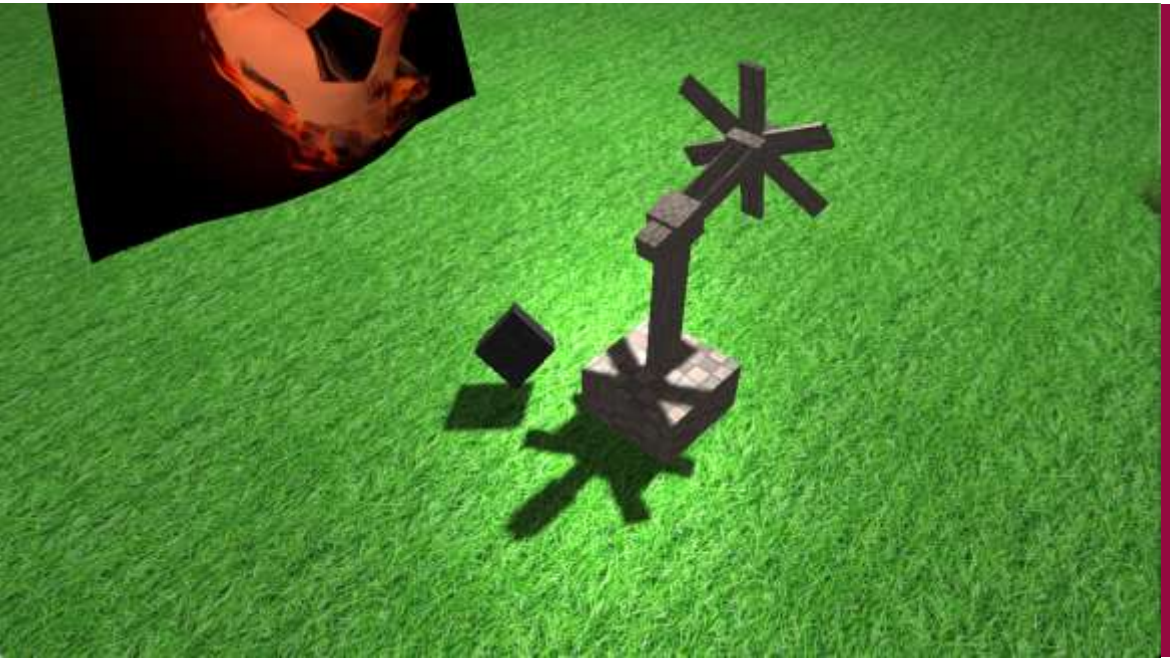
本次大作业取gamma = 1.2以获得更自然的效果。

```
88      FragColor = vec4(lighting, 1.0);
89
90      // Gamma 校正
91      if (gamma) {
92          float Gamma = 1.2f;
93          FragColor.rgb = pow(FragColor.rgb, vec3(1.0 / Gamma));
94      }
95  }
```

关闭 Gamma 校正

开启 Gamma 校正

深圳大学
SHENZHEN UNIVERSITY



Part 8
布料仿真

深圳大学 SHENZHEN UNIVERSITY

Spring-Mass System 模拟布料

网格状规则布料：

1. 在网格原本的边上加弹簧，抵抗布料产生横向和纵向的褶皱。
2. 在每个网格的两条对角线上加弹簧，抵抗布料在对角方向上的褶皱。
3. 跨过一个节点连弹簧，抵抗布料沿长边产生的翻折（Bending）。

共同点：

考察布料的三种变形：拉伸（stretch）、剪切（shear）、弯曲（bend）。

不同点：

1. Spring-Mass System 模拟布料从力的角度出发。
2. Large steps in cloth simulation 从能量的角度出发。

类似于：

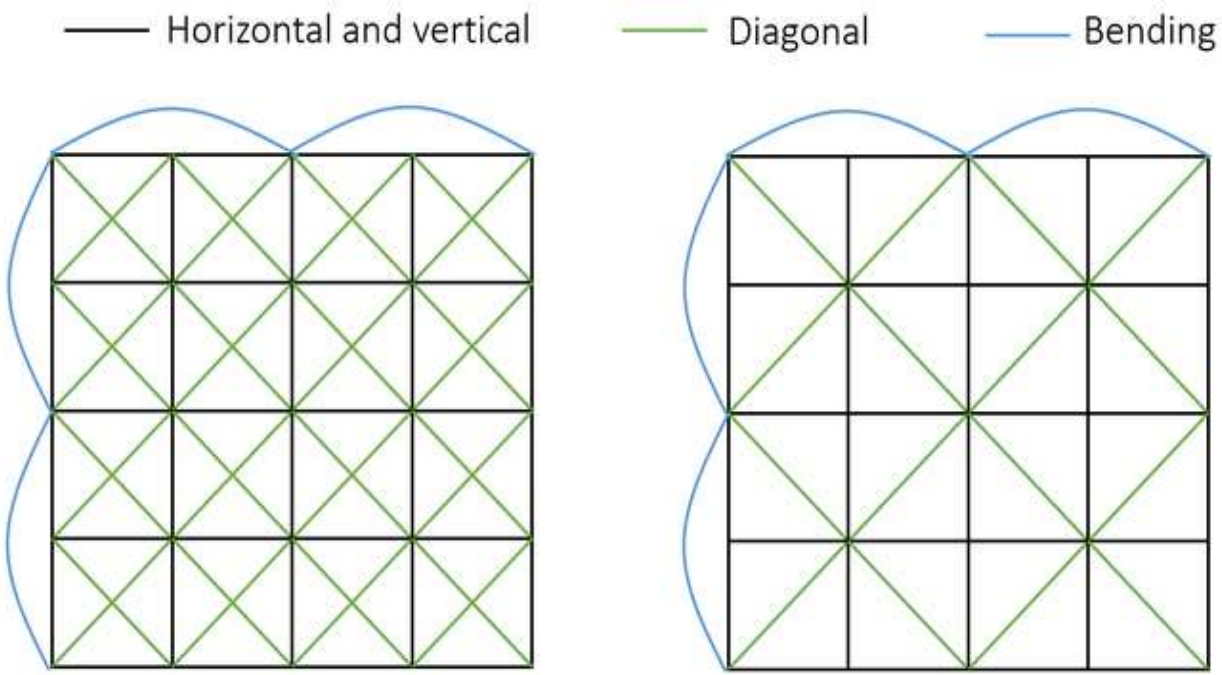
SIGGRAPH 98, Orlando, July 19–24 COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998

Large Steps in Cloth Simulation

David Baraff Andrew Witkin

Robotics Institute
Carnegie Mellon University

Structured Spring Networks



调参：

1. 布料的结构系数：1000.0
2. 布料的剪切系数：50.0
3. 布料的弯曲系数：400.0

深圳大学 SHENZHEN UNIVERSITY

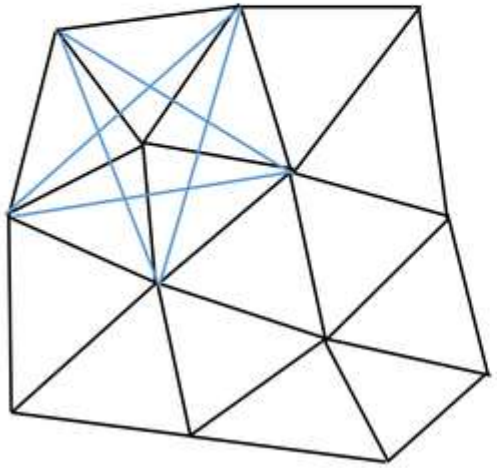
Spring-Mass System 模拟布料

不规则布料：

1. 三角剖分。
2. 在三角网格的每条边上加弹簧。
3. 跨过内部边加弹簧。

Unstructured Spring Networks

We can also turn an unstructured triangle mesh into a spring network for simulation.

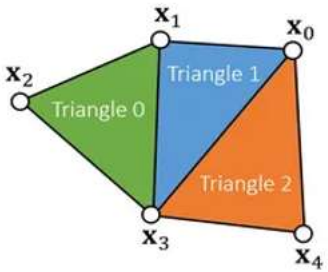


Edges

Bending (every neighboring triangle pair)

Triangle Mesh Representation

The basic representation of a triangle mesh uses vertex and triangle lists.



Vertex list: $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ (3D vectors)

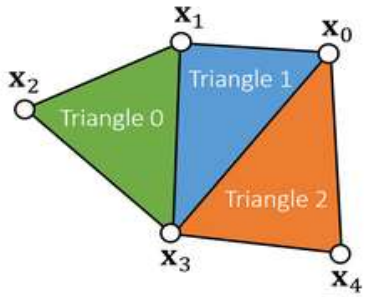
Triangle list: {1, 2, 3, 0, 1, 3, 0, 3, 4} (index triples)

Each triangle has three edges. But there are repeated ones.

Topological Construction

The key to topological construction is to sort triangle edge triples.

Each triple contains: edge vertex index 0, edge vertex index 1 and triangle index (index 0<index).



Triple list: {{1, 2, 0}, {2, 3, 0}, {1, 3, 0},
{0, 1, 1}, {1, 3, 1}, {0, 3, 1},
{0, 3, 2}, {3, 4, 2}, {0, 4, 2}}

Sorting

Sorted triple list: {{0, 1, 1}, {0, 3, 1}, {0, 3, 2}, {0, 4, 2},
{1, 2, 0}, {1, 3, 0}, {1, 3, 1}, {2, 3, 0}, {3, 4, 2}}

Removal

Edge list: {{0, 1}, {0, 3}, {0, 4}, {1, 2}, {1, 3}, {2, 3}, {3, 4}}

Neighboring triangle list: {{1, 2}, {0, 1}} (for bending)

深圳大学 SHENZHEN UNIVERSITY

显式 Euler 法

计算顺序：

合外力→加速度→ 速度→位移

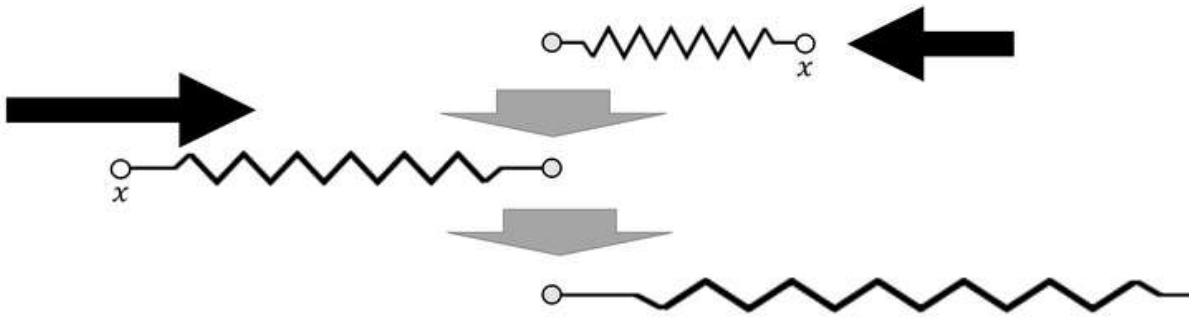
问题：

可能出现 overshooting，需要调参。

Explicit Integration of A Mass-Spring System

Explicit integration suffers from numerical instability caused by overshooting, when the stiffness k and/or the time step Δt is too large.

A naive solution is to use a small Δt . But that slows down the simulation.



Explicit Integration of A Mass-Spring System

Compute Spring Forces

For every edge e

$i \leftarrow E[e][0]$

$j \leftarrow E[e][1]$

$L_e \leftarrow L[e]$

$\mathbf{f} \leftarrow -k(\|\mathbf{x}_i - \mathbf{x}_j\| - L_e)\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}$

$\mathbf{f}_i \leftarrow \mathbf{f}_i + \mathbf{f}$

$\mathbf{f}_j \leftarrow \mathbf{f}_j - \mathbf{f}$

A Particle System

For every vertex

$\mathbf{f}_i \leftarrow \text{Force}(\mathbf{x}_i, \mathbf{v}_i)$

$\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t m_i^{-1}\mathbf{f}_i$

$\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\, \mathbf{v}_i$

Done

E: Edge list　　L: Edge length list　　m_i: Mass of vertex i

调参：

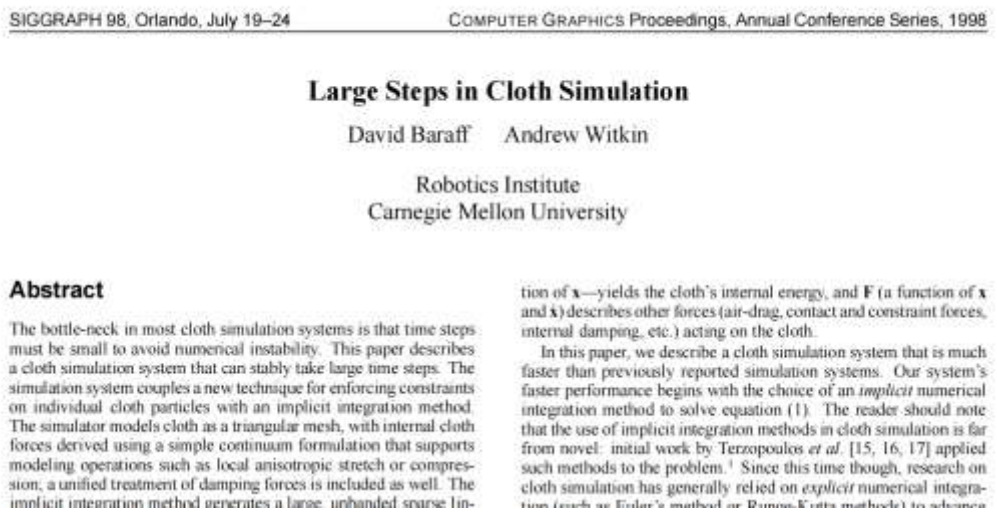
1. 弹簧的劲度系数：400.0
2. 弹簧的阻尼系数：5.0
3. 每次渲染循环计算 25 次

解决：

隐式 Euler 法。

深圳大学
SHENZHEN UNIVERSITY

隐式 Euler 法



SIGGRAPH 98, Orlando, July 19–24 COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998

Large Steps in Cloth Simulation

David Baraff Andrew Witkin

Robotics Institute
Carnegie Mellon University

Abstract

The bottle-neck in most cloth simulation systems is that time steps must be small to avoid numerical instability. This paper describes a cloth simulation system that can stably take large time steps. The simulation system couples a new technique for enforcing constraints on individual cloth particles with an implicit integration method. The simulator models cloth as a triangular mesh, with internal cloth forces derived using a simple continuum formulation that supports modeling operations such as local anisotropic stretch or compression; a unified treatment of damping forces is included as well. The implicit integration method generates a large, unbanded sparse lin-

tion of $\mathbf{x}$—yields the cloth's internal energy, and $\mathbf{F}$ (a function of $\mathbf{x}$ and $\dot{\mathbf{x}}$) describes other forces (air-drag, contact and constraint forces, internal damping, etc.) acting on the cloth.

In this paper, we describe a cloth simulation system that is much faster than previously reported simulation systems. Our system's faster performance begins with the choice of an *implicit* numerical integration method to solve equation (1). The reader should note that the use of implicit integration methods in cloth simulation is far from novel: initial work by Terzopoulos *et al.* [15, 16, 17] applied such methods to the problem.[1] Since this time though, research on cloth simulation has generally relied on *explicit* numerical integra-

工作:

为增加稳定性，需计算更多节点，这导致效率降低。

采用隐式 Euler 法，将解方程组转化为优化问题。

这样可采用更大的时间步长进行仿真，即 Large Steps 。

Implicit Integration

Implicit integration is a better solution to numerical instability. The idea is to integrate both $\mathbf{x}$ and $\mathbf{v}$ implicitly.

$$\begin{cases}\mathbf{v}^{[1]} = \mathbf{v}^{[0]} + \Delta t\mathbf{M}^{-1}\mathbf{f}^{[1]} \\ \mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t\mathbf{v}^{[1]}\end{cases} \quad \text{or} \quad \begin{cases}\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t\mathbf{v}^{[0]} + \Delta t^2\mathbf{M}^{-1}\mathbf{f}^{[1]} \\ \mathbf{v}^{[1]} = (\mathbf{x}^{[1]} - \mathbf{x}^{[0]})/\Delta t\end{cases}$$

Assuming that $\mathbf{f}$ is *holonomic*, i.e., depending on $\mathbf{x}$ only, our question is how to solve:

$$\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t\mathbf{v}^{[0]} + \Delta t^2\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}^{[1]})$$

Implicit Integration

These two are equivalent:

$$\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t\mathbf{v}^{[0]} + \Delta t^2\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}^{[1]})$$

$$\mathbf{x}^{[1]} = \operatorname{argmin} F(\mathbf{x}) \quad \text{for} \quad F(\mathbf{x}) = \frac{1}{2\Delta t^2}\left\|\mathbf{x} - \mathbf{x}^{[0]} - \Delta t\mathbf{v}^{[0]}\right\|_{\mathbf{M}}^2 + E(\mathbf{x})$$

$\|\mathbf{x}\|_{\mathbf{M}}^2 = \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}$

This is because:

$$\nabla F(\mathbf{x}^{[1]}) = \frac{1}{\Delta t^2}\mathbf{M}(\mathbf{x}^{[1]} - \mathbf{x}^{[0]} - \Delta t\mathbf{v}^{[0]}) - \mathbf{f}(\mathbf{x}^{[1]}) = \mathbf{0}$$

$$\mathbf{x}^{[1]} - \mathbf{x}^{[0]} - \Delta t\mathbf{v}^{[0]} - \Delta t^2\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}^{[1]}) = \mathbf{0}$$

Note that this is applicable to every system, not just a mass-spring system.

深圳大学
SHENZHEN UNIVERSITY

彩蛋

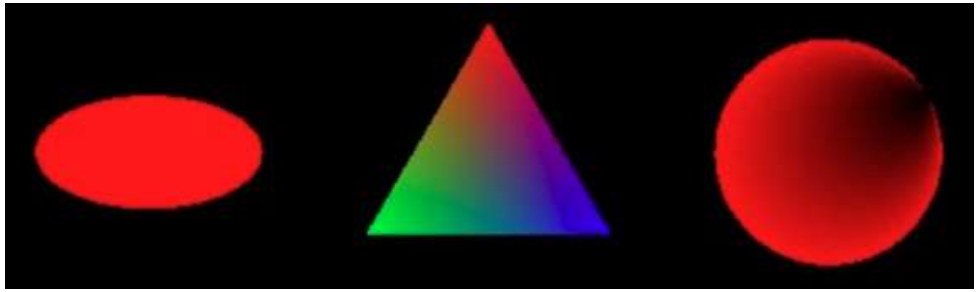
现象:

蛋黄（球面）很光滑。

原因:

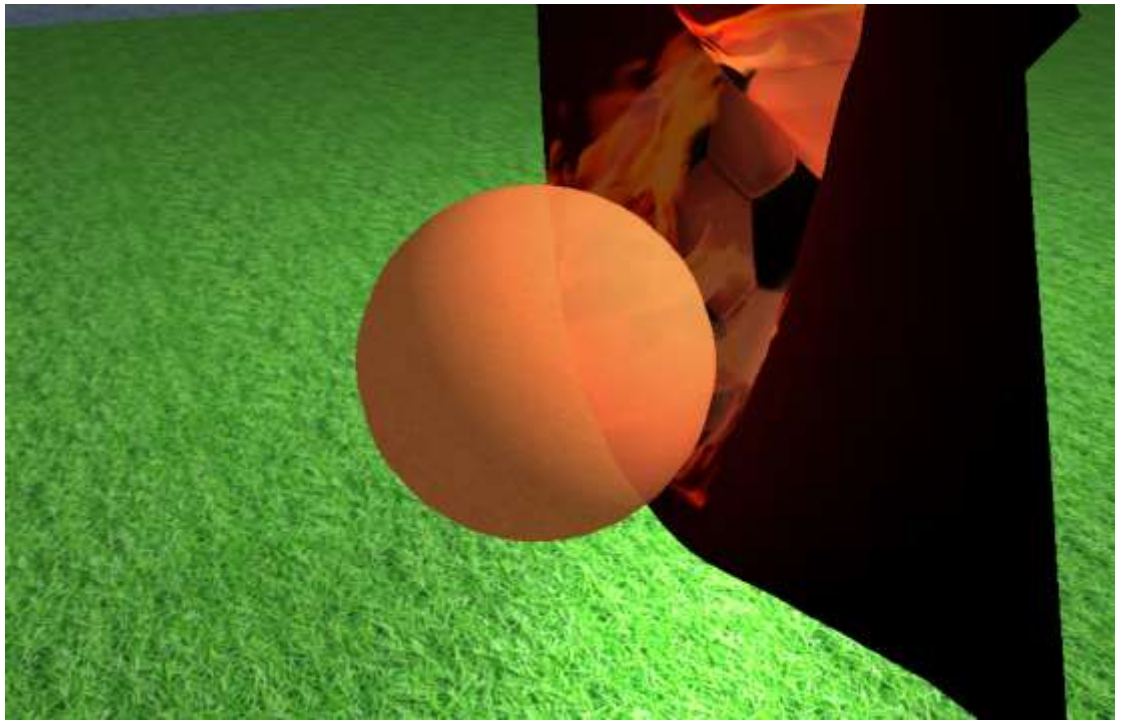
不是通过模型文件绘制，而是通过采样绘制。

类似于 **实验一** 绘制的圆和椭圆。



类似于:

对球的三重积分，先取一个截面，在其上积分后，再对截面高度积分。



解 ∵被积函数仅为 z 的函数，

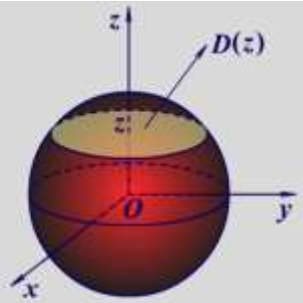
截面 $D(z)$ 为圆域 $x^2+y^2\le 1-z^2$，

故采用“先二后一”法.

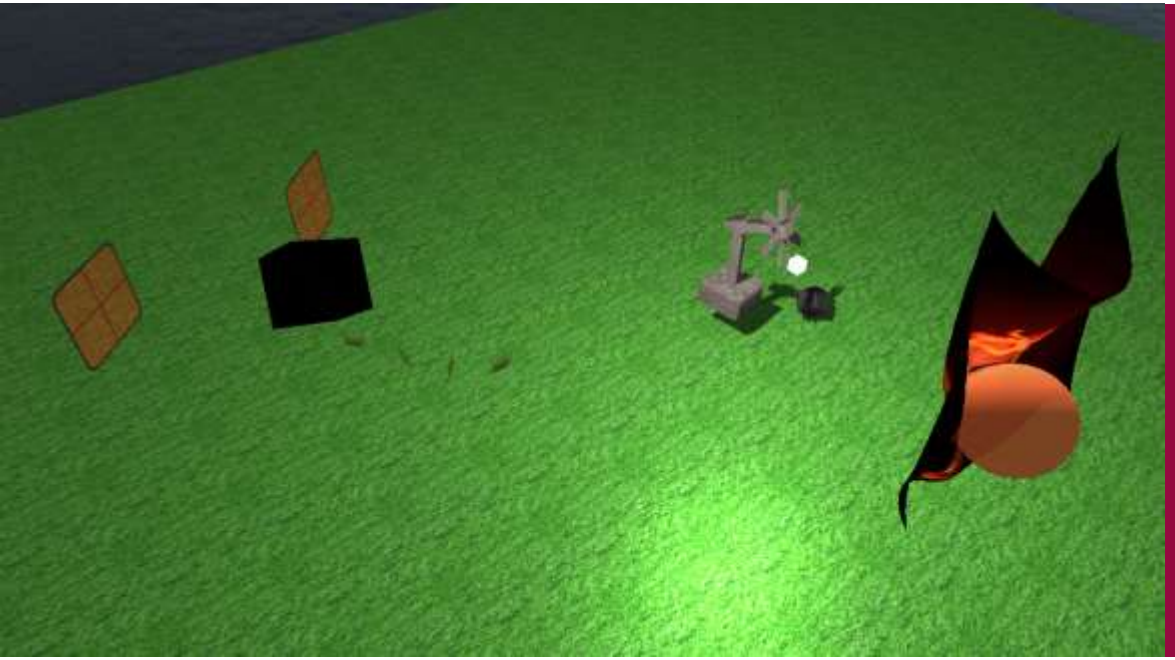
$$\iiint_{\Omega} e^{|z|}dv = 2\iiint_{\Omega_{\perp}} e^{z}dv$$

$$= 2\int_0^1 \left[\iint_{D(z)} dxdy\right] e^{z}dz$$

$$= 2\int_0^1 \pi(1-z^2)e^{z}dz = 2\pi.$$

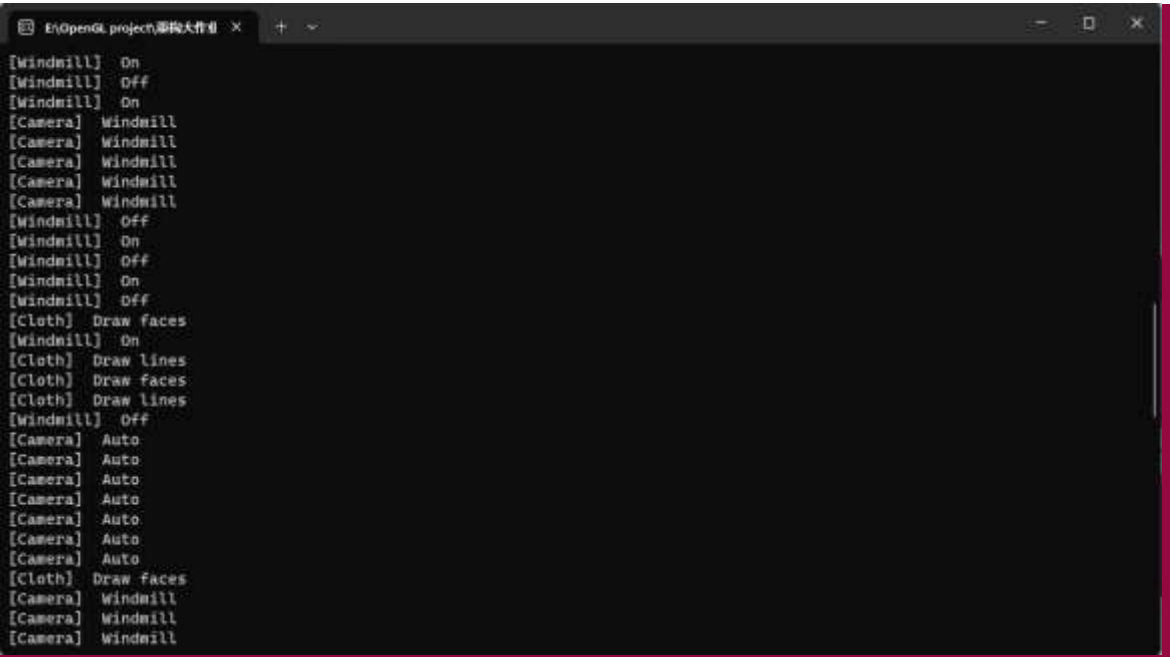


深圳大学
SHENZHEN UNIVERSITY



Part 9
视角切换

深圳大学
SHENZHEN UNIVERSITY



Part 10

命令行输出提示信息

深圳大学
SHENZHEN UNIVERSITY

Part 11
心得体会

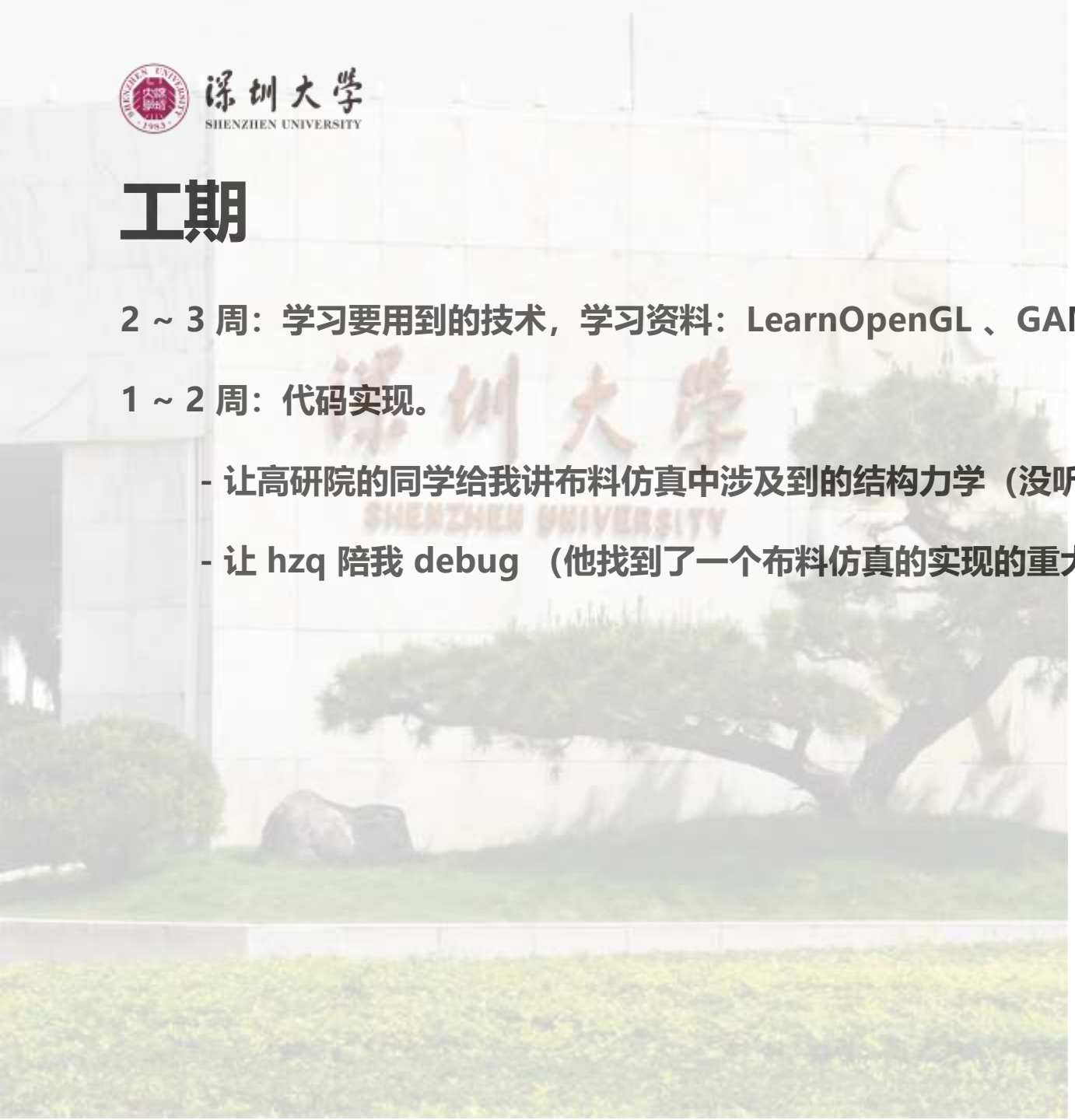
深圳大学
SHENZHEN UNIVERSITY

工期

2 ~ 3 周：学习要用到的技术，学习资料：LearnOpenGL 、GAMES 101 、GAMES 103 、相关论文、知乎、b 站。

1 ~ 2 周：代码实现。

- 让高研院的同学给我讲布料仿真中涉及到的结构力学（没听懂）。
- 让 hzq 陪我 debug （他找到了一个布料仿真的实现的重大 bug， 太伟大了）。



深圳大学
SHENZHEN UNIVERSITY

OpenGL 调试方法

1. 为 shader 的读取、编译、链接设置异常处理, 输出对应的日志信息. 若程序能运行, 但有物体未显示出来, 可能是其对应的 shader 读取或编译失败.

2. 若物体不能正常显示, 一般是 vshader 的问题, 可检查设置的 uniform 变量的名称是否对应.

3. 若物体位置能正确显示, 但纹理或光照等不正确, 一般是 fshader 的问题, 可检查变量名是否写错.

4. 反射光只能在一定范围内看见, 若光源和材质的参数设置正确但未见物体的反射, 可能是摄像机太高了(高于光源).

5. 注意函数的参数向量的方向, 如光源指向片段还是片段指向光源.

6. 将物体加入场景中、对物体做变换时, 要考虑是在何坐标系下进行.

7. 若将顶点坐标、法向量、颜色、纹理坐标等集合在一个 float[] 中, 则修改后要更新 glVertexAttribPointer() 中的步长、偏移量等.

   若程序能过编译但运行崩溃, 考虑为此原因.

8. 若一个物体涉及到多个顶点, 如一个弹簧有两个质点, 计算空间时要乘上顶点个数.

9. 若有多个 shader , 则设置每个 shader 的 uniform 变量时要注意使用该 shader 的着色器程序对象.

10. 注意 model 矩阵中平移、旋转、缩放顺序.

深圳大学
SHENZHEN UNIVERSITY

图形学学习心得

1. 学图形学就像学 ACM，理论简单实现难，轻理论重实践.

2. 从学习图形学的角度，最好能看懂所有理论和实现，如深度缓冲等.

   从编程的角度，无需看懂所有理论和实现，只需将代码贴到正确的位置和修改参数.

3. GAMES 101 讲得很好，但是课堂上纯理论，几乎不讲实现，不适合新手. LearnOpenGL 理论和实现结合，顺序合理，适合新手入门图形学.

   SZU 的图形学理论课顺序安排合理，实验课注重实现.

4. 多文件，OOP，多封装繁琐的操作，如 shader 类、camera 类、绘制正方体、绘制球、绘制刚体、绘制布料等.

5. 手推几何关系时不能将图画得太特殊.

6. 要画出好看的场景，除了需要过硬的理论和实现，还需要良好的审美. 认知决定上限.

7. 仿真写起来太恶心了，物理又看不懂，代码又调不对.

8. LearnOpenGL 的中文翻译有挺多错的，如果发现自己有地方读不懂，可以怀疑是翻译错了. 建议对照英文版阅读.

9. 不迷信权威，如 LearnOpenGL 中关于有透明物体的场景的渲染顺序的介绍有点问题.

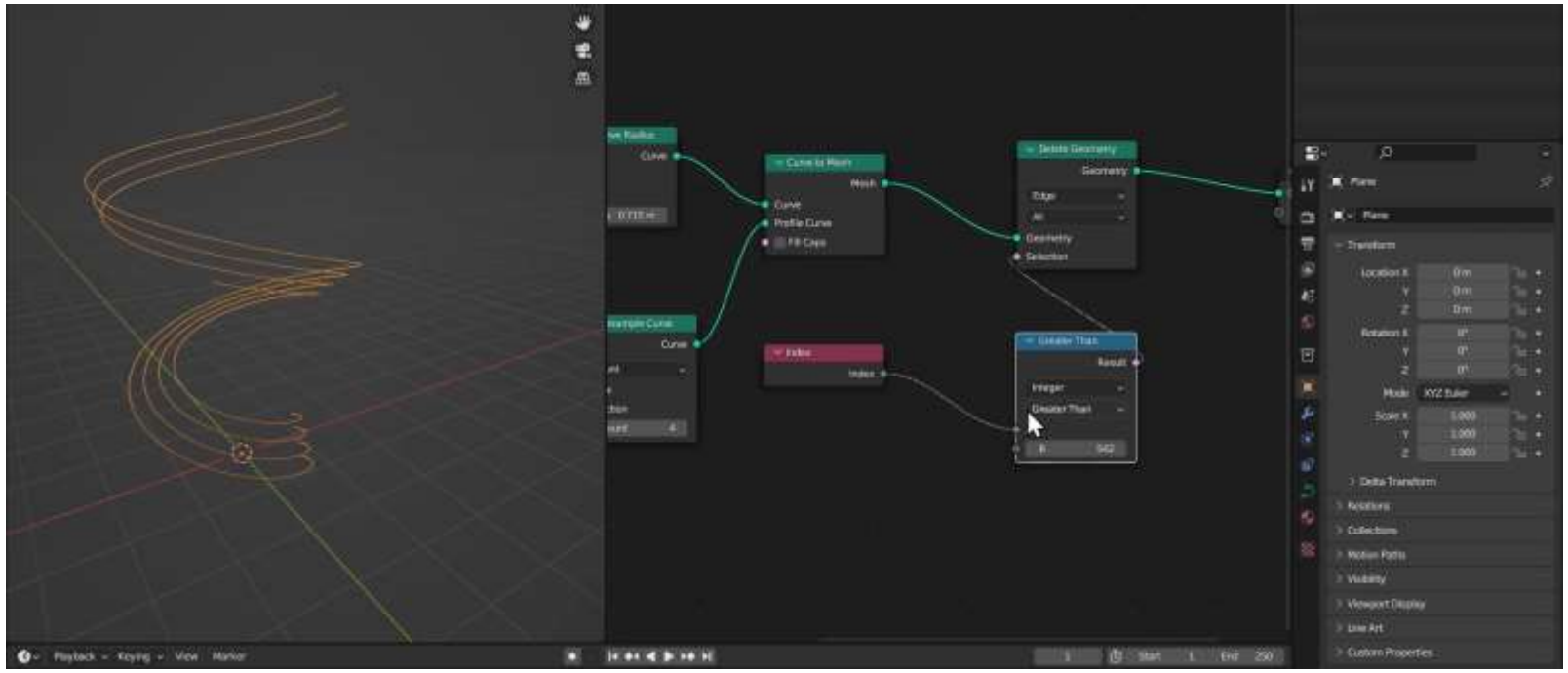
   我认为的顺序：① 不透明物体、全透明物体；② 天空盒；③ 按顺序的半透明物体.

深圳大学
SHENZHEN UNIVERSITY

图形学学习心得

10. 有人说 OpenGL 很难用, 哪里难了, 这么多年一直是这个难度.

当然, 还是 blender 好用.



blender


哔哩哔哩晚会 bilibili
淘宝
bilibili
浮光掠影
重山彩云间
淘宝 2023
最美的夜
bilibili·晚会

THANKS