

基于 Unity Shader 石油泄漏现象模拟的研究

刘贤梅, 葛昊天, 赵 娅

(东北石油大学 计算机与信息技术学院, 黑龙江 大庆 163318)

摘 要:为模拟油田生产过程中的意外情况,应急仿真系统应运而生,而在系统的场景中特效模拟的真实感是应急仿真系统中非常重要的一环。随着硬件的不断发展,GPU(graphics processing unit,图形处理单元)性能的大幅度提高及其可编程能力的快速发展,开发人员可以通过可编程着色器(Shader),开发出更加具有高质量、高真实感的效果。文中针对漏油应急培训中石油泄漏现象进行模拟,设计并制作了三维模型来模拟石油泄漏部分的主体。在 Unity 平台上,使用 Unity 中对于可编程着色器(Shader)进行封装的 Unity Shader,首先采用纹理切换方法模拟石油泄漏的流动效果,然后针对纹理切换无法模拟出反射折射效果的问题,设计了基于噪声方法,既模拟出石油的流动效果,又模拟出流动过程中的反射折射效果,大大提高了真实感。

关键词:石油泄漏;真实感模拟;Unity Shader;渲染流水线;GPU

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2020)07-0145-05

doi:10.3969/j.issn.1673-629X.2020.07.031

Simulation of Phenomenon of Petroleum Spill Based on Unity Shader

LIU Xian-mei, GE Hao-tian, ZHAO Ya

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: In order to simulate the unexpected situation in the process of petroleum production, the emergency simulation system comes into being. The reality of simulation in the scene of the system is quite important in the emergency simulation system. With the continuous development of hardware, significant improvement in performance and rapid development in programmable ability of GPU (graphics processing unit), developers can use the programmable Shader to develop more high-quality, realistic effects. Aiming at petroleum leakage phenomenon to simulate in petroleum spill emergency training, we design and construct a three-dimensional model to simulate the main body of petroleum spill. In the Unity platform, the Unity Shader encapsulating Unity for programmable Shader is applied. Firstly, the texture switching method is used to simulate the flow effect of petroleum leakage. Then, aiming at the problem that the texture switching method cannot simulate the reflection and refraction effect, a noise-based method is designed to simulate not only the flow effect of petroleum, but also the reflection and refraction effect in the flow process, which greatly improves the sense of reality.

Key words: petroleum spills; realistic simulation; Unity Shader; rendering pipeline; GPU

0 引 言

近年来不断发展成熟的虚拟现实技术(virtual reality),利用计算机图形学、三维建模、多媒体等多种技术手段^[1],可以在计算机上实现逼真的三维场景的渲染,为模拟石油生产过程中的意外情况,提高工人的突发情况应急处理能力,应急仿真系统应运而生。在模拟石油泄漏事故的应急仿真系统中,石油泄漏现象的流动效果的模拟具有重大意义。传统的模拟方式以粒子系统为主,但是使用粒子系统时,随着粒子数目的增加,需要的内存增加,运行速率也变得很慢。

随着硬件设备的不断发展,GPU(graphics

processing unit,图形处理单元)性能的大幅度提高,由于GPU的可编程着色器的存在,开发人员可以通过可编程着色器(Shader)^[2],对顶点及像素进行灵活的处理,模拟出更多逼真的效果。2006年上海师范大学苏蕴等人通过渲染流水线的可编程着色器的控制,完成对于室内光的各种形式(点光源、平行光源等)的真实感模拟^[3]。2015年,海军航空工程学院的王彦等人通过可编程着色器完成大面积动态海洋红外视景仿真,提高了水流的真实感^[4]。因此,针对石油泄漏的流动效果,可以采用可编程着色器(Shader)进行模拟,不仅能够模拟出石油流动的效果,还不需要消耗很多内存,

收稿日期:2019-07-24

修回日期:2019-11-25

基金项目:国家自然科学基金(61502094);黑龙江省自然科学基金(F2016002);东北石油大学研究生创新科研项目(YJSCX2015-033NEPU)

作者简介:刘贤梅(1968-),女,教授,硕导,研究方向为虚拟现实;葛昊天(1995-),男,硕士研究生,研究方向为虚拟现实。

能够一直保持一个较快的运行速率。

Unity 是一个专业的游戏引擎,而 Unity 的着色器 (Unity Shader) 是对着色器上层的一个抽象,它负责将输入的 Mesh(网格)以指定的方式与输入的贴图或者颜色等组合作用,输出纹理^[5]。文中首先探讨了着色器和着色器语言,并对 Unity Shader 的文件结构与分类进行探讨,然后通过 Unity Shader 控制底层 GPU 中的可编程着色器,采用纹理切换方法与基于噪声的方法对输入的贴图进行处理,模拟出石油流动的效果。

1 着色器

1.1 着色器与着色器语言

Shader,中文翻译为“着色器”,其本质是作用于渲染流水线上的一段程序,能够控制渲染流水线的特定阶段进行计算,以达到目标效果。现代可编程渲染流水线的最大特点在于 GPU 内部硬件设计具有可编程性,可以通过程序控制 GPU 内部渲染流水线。GPU 中的可编程结构,包括顶点着色器和片元着色器,即可以针对顶点和像素进行编程^[6]。因此,可以在顶点和像素着色器中编写具有不同功能的程序来处理输入信息,使整个处理过程更具灵活性和可控性。

Shader 的主流编程语言有 HLSL、GLSL、CG。HLSL (high level Shader language) 是微软基于 DirectX 的语言,只能运行在 Windows 平台上^[7]。GLSL (OpenGL Shading language),是用在 OpenGL 中着色编程的语言,也是一门跨平台的着色器语言。Cg 语言 (C for graphics) 是为 GPU 编程设计的高级着色器语言,由 NVIDIA 公司开发^[8],可作为片段程序镶嵌在 Unity Shader 程序中。

1.2 Unity Shader

1. ShaderLab。

为了避免直接控制底层着色器,Unity 提供一层抽象——Unity Shader,此外还提供了一种专门为 Unity Shader 服务的语言——ShaderLab。它是一种说明性的语言,使用了一些嵌套在花括号内部的语义,这些语义包含了渲染所需的数据,包括着色器所需的各种属性^[9]。而 Unity 会自动将 ShaderLab 文件编译成控制 GPU 中各个着色器的代码。

2. ShaderLab 文件的结构。

一个 ShaderLab 的基础文件结构如下所示:

```
Shader "ShaderName"
{
    Properties {
        //属性
    }
    SubShader {
```

```
//显卡 A 使用的着色器
```

```
}
```

```
SubShader {
```

```
//显卡 B 使用的着色器
```

```
}
```

```
.....
```

```
Fallback "VertexList"
```

```
}
```

对各个部分的释义如下:

(1) Properties。

Properties 语义块中包含了一系列属性,并且这些属性会出现在材质面板中,定义方式通常如下:

```
_Name ( " Display Name" , type ) = defaultValue
[ { options } ]
```

这些属性的存在是为了能够更好地调整材质的属性,方便在着色器中访问它们。“Display Name”是出现在材质面板上的名字,type 是属性的类型,常用的属性类型如表 1 所示,defaultValue 是指定的默认值。

表 1 Properties 语义块支持的属性类型

属性类型	默认值的定义语法
Int(整形)	number
Float(浮点型)	number
Color(颜色值)	(number , number , number , number)
Vector(向量)	(number , number , number ,)
2D	defaultValue { }

(2) SubShader。

一个 Shader 有多个 SubShader。一个 SubShader 可理解为一个 Shader 的渲染方案。即 SubShader 是为了针对不同的渲染情况而编写的。每个 Shader 至少 1 个 SubShader。Unity 会扫描所有的 Unity Shader,然后选择第一个能够在目标平台上运行的 SubShader。SubShader 语义块中包含的定义如下:

```
SubShader
{
    Pass { }
```

一个 SubShader 是由 Pass 块来执行的。每个 Pass 定义了一次完整的渲染流程。在 Pass 内,定义了输入输出结构体,在结构体中进行光照,法线与空间变换等的计算。

(3) FallBack。

在各个 SubShader 语义块之后,会存在一个 FallBack,它会规定最低级的 Shader 是哪一个。由于一段 Shader 有可能在不同等级的显卡上运行,而显卡的能力也是参差不齐的。所以,FallBack 相当于留了一条“后路”,以保证若之前所有的 SubShader 都无法

在当前的显卡上运行,则运行 FallBack 所指定的 Shader^[10]。

1.3 Unity Shader 的形式

(1) 表面着色器。

表面着色器(surface Shader)是 Unity 创造的一种着色器代码类型。它在本质上就是对顶点/片元着色器的封装,省去了一些重复代码编写的工作量。也就是 Unity 在提供一个表面着色器之后,在背后仍然将它变为顶点片元着色器。但是相比于顶点/片元着色器,表面着色器更加简洁,效率更高^[11]。

(2) 顶点/片元着色器。

相比于表面着色器,顶点/片元着色器(vertex/fragment Shader)更加复杂,但同时灵活性更高。可以根据自己当前的不同需求,编写控制顶点着色器和片元着色器的代码,对 GPU 渲染流水线中的可编程部分直接进行灵活的控制。

(3) 固定函数着色器。

上述的两种 Shader 都使用了可编程管线,而对于一些比较老旧的,不支持可编程管线着色器的设备,则需要使用固定函数着色器(fixed function Shader)来完成渲染^[12],这些着色器往往只能完成一些比较简单的效果,目前很少使用。

1.4 Unity Shader 的使用方法

首先根据需求,选择一种 Unity Shader 形式,编写 Shader 着色程序。然后在工程中新建材质球,将上述

写好的着色器程序赋给材质球,并给材质球设置贴图,作为着色器程序的输入,最后将材质球附在物体模型上,就得到了目标效果^[13]。

2 石油流动效果的实现

针对石油的泄漏现象,为模拟出石油流动时石油的主体部分,在 3dsMax 中制作出一个边缘光滑,形状不规则的物体,作为模拟石油流动的主体模型。石油主体模型在 3dsMax 中的截图如图 1 所示。

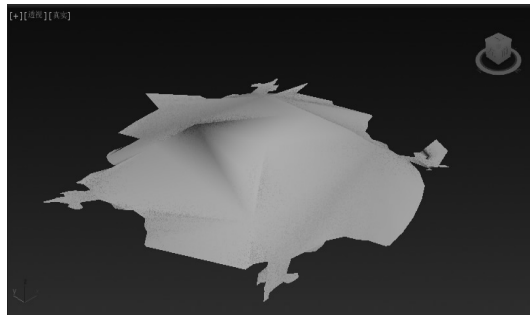


图1 模型截图

2.1 基于纹理切换的石油流动效果设计与实现

该方法实现石油流动效果采用的是表面着色器(surface Shader),规定了两个偏移速度 x, y , 速度乘以随时间变化的时间变量得到偏移量。对图片进行偏移计算之后的点进行采样,获得偏移点的颜色值作为返回值输出,以达到石油流动的效果。流程图如图 2 所示。

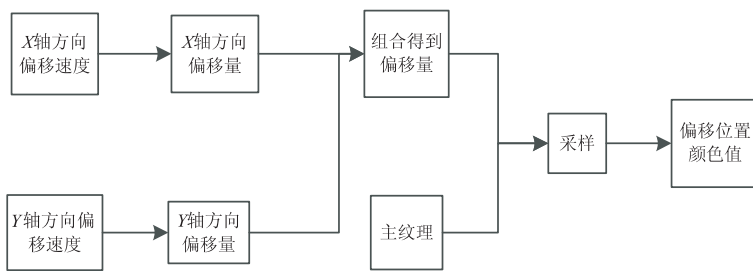


图2 纹理切换流程

在 Properties 块中,需要声明贴图与贴图的 x, y 轴偏移速度。输入的主纹理贴图的格式是 2D,而偏移速度为范围在 $(0, 100)$ 之间的数值。

```
_WaterTex("水的纹理图:", 2D) = "white" {}
_XSpeed("X 轴方向的纹理滚动速度", :
Range(-10, 10)) = 1
_YSpeed("Y 轴方向的纹理滚动速度", :
Range(-10, 10)) = 1
```

Properties 块结束后,需要在 SubShader 块(渲染方案)中,嵌套 CG 代码块。首先,通过结构体 Input,将模型顶点的 uv 值作为 surf 函数的输入。在表面函数 surf 中,对贴图进行计算。通过内置的时间变量 $_Time$ 乘以 x 轴, y 轴方向的偏移速度,得到偏移量,通过 x 轴

偏移量与 y 轴偏移量的组合,得到偏移量 scrolledUV, 并与顶点的 uv 值相加得到偏移量,通过 tex2D 函数,对贴图在偏移处进行采样,得到偏移处的颜色值。由于偏移量随时间改变而改变,因此,采样得出的颜色值也会随时间改变,这样就类似于贴图在进行运动,以实现流动的效果。

```
fixed xValue = _XSpeed * _Time;
fixed yValue = _YSpeed * _Time;
UV += fixed2(xValue, yValue);
half4 c = tex2D(_WaterTex, UV);
o.Albedo = c.rgb;
```

2.2 基于纹理切换的石油流动效果展示

图 3 为使用纹理切换方法在 RenderDoc 中截取的

八帧图像。

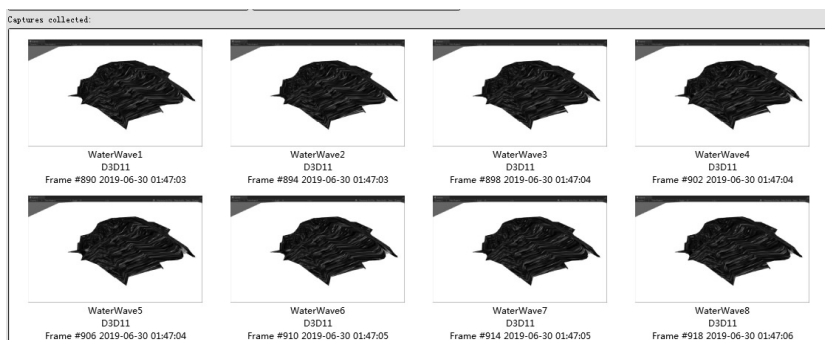


图 3 RenderDoc 中的截图(1)

使用上述方法模拟在场景中发生漏油的效果如图 4 所示。

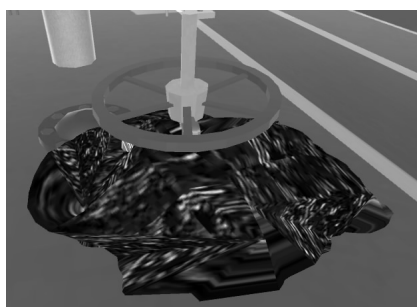


图 4 纹理切换方法模拟效果图

2.3 基于噪声的石油流动效果的设计与实现

使用上述方法能够有效模拟出石油流动的效果,但是无法模拟出石油在光照下的反射与折射效果。因此,引入噪声图片,对噪声进行采样得到的随机值,来实现贴图的动态扭曲效果,以实现贴图的反射折射效

果,提高模拟的真实感。

在图形学中使用噪声是为了把一些随机变量引入到程序中,如火焰、地形、云朵的模拟等等都要使用随机变量。在图形学中称这种噪声为“白噪声”(功率谱密度在整个频域内均匀分布的噪声)^[14]。噪声的基础来自于随机数,若屏幕上的每个像素点给一个 0~1 之间的随机数来表示像素点的亮度^[15],就能得到白噪声纹理。因此可以通过对白噪声图片进行采样以得到随机数,完成对于石油反射折射效果的模拟。

该方法采用的是顶点/片元着色器。在该模拟中,对噪声图进行采样,其中对采样点设置时间变量,随着时间不断发生偏移,它提供的 R 值就会不停的变化,将 R 值再与顶点数据相加作为采样点,对主贴图进行采样,将得到的颜色作为输出,那么图片就会出现动态的扭曲效果,就实现了想要的石油流动效果。

流程图如图 5 所示。

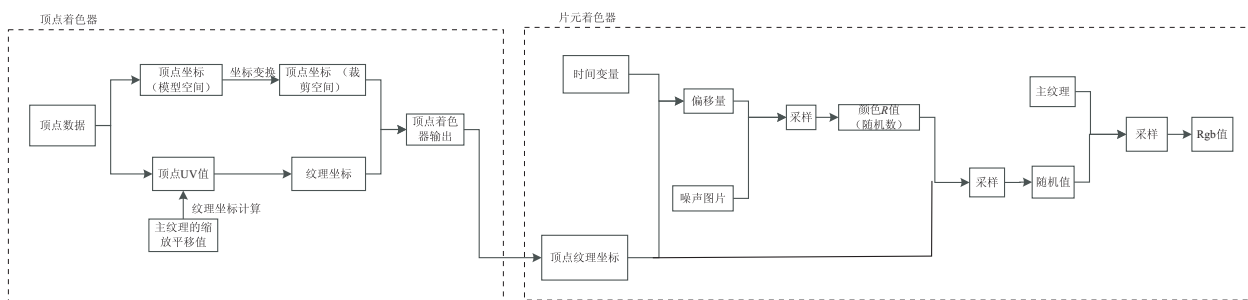


图 5 噪声模拟流程

在 Pass 通道嵌套的 CG 代码段中,首先对顶点着色器进行控制,在顶点阶段主要完成两个工作,包括把顶点坐标从模型空间转换到裁剪空间,再使用纹理的属性(`_MainTex_ST.xy`, `_MainTex_ST.zw`)对顶点坐标进行计算,得到该阶段计算之后的纹理坐标。主要计算过程如下:

```
v2f vert (appdata v)
{
    o.vertex = UnityObjectToClipPos( v.vertex );
    o.uv = TRANSFORM_TEX( v.uv, _MainTex );
}
```

在片元着色器中,实现的是噪声模拟方法的主体。

首先设置时间变量乘以速度,得到偏移量,偏移量与顶点数据相加作为采样点,对噪声图片进行采样,得到 rgb 值,此时,将 R 值提出,作为随机数,再与顶点数据相加,作为采样点对主贴图进行采样,得到对应点的 rgb 值,作为返回值返回。图片就会出现动态的扭曲效果,就实现了想要的石油流动效果。

```
fixed4 frag( v2f i ):SV_Target{
    fixed4 noise_col = tex2D( _NoiseTex, i.uv + fixed2( _Time.y * _XSpeed, 0 ));
    fixed uOffset = noise_col.r;
    fixed vOffset = noise_col.r;
```



```
fixed4 col = tex2D (_MainTex, i. uv + _Intensity * fixed2
(uOffset, vOffset));
UNITY_APPLY_FOG(i. fogCoord, col);}
```



图6 RenderDoc 中的截图(2)

使用上述方法模拟在场景中发生漏油的效果如图7所示。

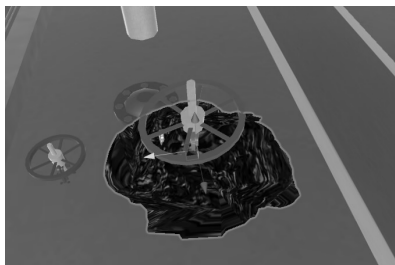


图7 噪声方法模拟效果图

3 结束语

利用 Unity Shader 技术,模拟出石油的流动效果。首先对着色器进行简单介绍,然后针对 Unity Shader,探讨 ShaderLab 的结构及其分类。通过 ShaderLab 的两种形式—表面着色器和顶点/片元着色器,完成两种石油流动效果的模拟。一种使用纹理切换方式,引入时间变量对贴图进行采样,实现贴图运动效果;另一种对噪声进行采样,得到随机数,通过该随机值与顶点进行组合进行采样,贴图就会出现动态的扭曲效果,完成对石油流动效果的模拟。

参考文献:

- [1] FOSTER N, METAXAS D. Modeling the motion of a hot, turbulent gas[C]//Proceedings of the 4th annual conference on computer graphics and interactive techniques. Chicago: ACM, 1997:181-188.
- [2] AMENTA N, KILY J. Defining point-set surfaces[J]. ACM Transactions on Graphics, 2004, 23(3):264-270.
- [3] 苏 蕴, 陈操宇, 郭善良. 基于顶点 Shader 实现的光照计算技术[J]. 计算机技术与发展, 2006, 16(12):58-60.

2.4 基于噪声的石油流动效果展示

图6为使用上述方法模拟时在 RenderDoc 中截取的八帧图像。

- [4] 王 彦, 谢晓方, 孙海文, 等. 中波段动态海洋红外视景仿真[J]. 红外与激光工程, 2015, 44(12):3604-3609.
- [5] 王长波, 张卓鹏, 张 强, 等. 基于 LBM 的自由表面流体真实感绘制[J]. 计算机辅助设计与图形学学报, 2011, 23(1):104-110.
- [6] 胡贵菊, 吴志红, 王丹霞. 基于 GPU 的实时雾效模拟[J]. 计算机应用, 2010, 30(3):585-588.
- [7] 朱红斌, 刘学惠, 柳有权, 等. 基于 Lattice Boltzmann 模型的液-液混合流模拟[J]. 计算机学报, 2006, 29(12):2071-2079.
- [8] 吴恩华. 图形处理器用于通用计算的技术、现状及其挑战[J]. 软件学报, 2004, 15(10):1493-1504.
- [9] 李建明, 吴云龙, 何荣盛, 等. 基于粒子系统和 GPU 加速的喷泉实时仿真[J]. 系统仿真学报, 2009, 21(10):3139-3141.
- [10] 周翰玮, 杨士普, 司江涛. 飞机机头附近液态水含量分布规律的数值研究[J]. 航空工程进展, 2015, 6(4):465-472.
- [11] 张绪树, 郭 媛, 陈维毅. 基于任意拉格朗日-欧拉流固耦合方法的标枪飞行仿真[J]. 医用生物力学, 2008, 23(2):124-126.
- [12] JOHN E, JOHN A, HENRY F, et al. Pixel-plane 4: a summary, advances in computer graphics hardware II[C]//Eurographic seminars tutorials and perspectives in computer graphics. New York: Springer-Verlag, 1988:183-208.
- [13] IHMSEN M, AKINCI N, BECKER M, et al. A parallel SPH implementation on multi-core CPUs[J]. Computer Graphics Forum, 2011, 30(1):99-112.
- [14] FOSTER N, FEDKIW R. Practical animation of liquids [C]//Computer graphics conference series. Los Angeles: [s. n.], 2001.
- [15] 李 纲, 李 辉. GPU 上的实时三维云仿真[J]. 系统仿真学报, 2009, 21(23):7511-7514.