

High Velocity Tail Flame Simulation Algorithm Based on GPU Acceleration

Song Tang, Zejun Lv, Tiangang Dong

College of Computer, Sichuan University, Chengdu Sichuan

Email: tangsong1994@126.com, lvzj186@163.com, dongtiangang@scu.edu.cn

Received: Aug. 6th, 2019; accepted: Aug. 20th, 2019; published: Aug. 27th, 2019

Abstract

In some special application scenarios such as the Air Force training system, it is necessary to use a computer group composed of multiple computers to simulate dozens of aircraft in real time, especially to simulate the tail flame effects of fighters in different weather conditions. The tail flame of the aircraft is different from the ordinary flame, and has the characteristics of high flow velocity and special flame color. The traditional simulation method will tear when simulating high-flow flame, and requires a large number of particles. When simulating large-scale fighter groups and other particle effects, the total number of particles in the scene will be too high, which requires a high graphics card performance. In order to achieve better simulation results and reduce the hardware requirements, this paper proposes a high-speed jet tail flame simulation algorithm based on GPU acceleration for large-scale fighters. It uses multiple rendering pipelines to accelerate the calculation of flame particle motion and disturbs the flame by white Gaussian noise. Finally, the heat value decay function is designed according to the characteristics of the tail flame of the fighter to further improve the simulation effect of the flame. The computer simulation results verify the feasibility and effectiveness of the proposed algorithm, and reduce the hardware performance requirements while satisfying the realism.

Keywords

GPU, Parallel Computing, Flame Simulation, Particle System, Gaussian Noise

基于GPU加速的高流速尾焰仿真算法

唐松, 吕泽均, 董天罡

四川大学, 计算机学院, 四川 成都

Email: tangsong1994@126.com, lvzj186@163.com, dongtiangang@scu.edu.cn

收稿日期: 2019年8月6日; 录用日期: 2019年8月20日; 发布日期: 2019年8月27日

摘要

在空军训练系统等一些特殊应用场景，需要使用由多个计算机组成的计算机组对数十架飞机进行实时虚拟仿真，模拟不同天气环境下战斗机的尾焰特效。战斗机尾焰和普通火焰不同，具有高流速、焰色特殊等特点。传统仿真方法在模拟高流速火焰时会出现撕裂现象，同时仿真需要的粒子数偏高，在模拟大规模战斗机群和其它粒子特效时会造成场景中总粒子数过高，需要较高的显卡性能。为了达到较好的仿真效果，同时降低对硬件的需求，本文提出了一种基于GPU加速的大规模战斗机高速喷射尾焰仿真算法，使用多个绘制管线加速火焰粒子运动计算，采取高斯白噪声来扰动尾焰，最后根据战斗机尾焰燃烧特性设计了热值衰变函数来进一步提高火焰的仿真效果。计算机仿真实验结果验证了本文提出算法的可行性和有效性，在满足真实感的前提下减小了对硬件性能的要求。

关键词

GPU，并行计算，火焰仿真，粒子系统，高斯噪声

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

三维场景中的火焰特效仿真一直是虚拟仿真领域的一个重要的问题，目前一些火焰渲染算法已经广泛应用于游戏、动画特效、虚拟现实或其他仿真系统中。在一些特殊的应用场景，比如空军训练虚拟仿真系统中，需要实时模拟出大规模的发动机尾焰。同时在各种气候条件下满足真实感。

目前的仿真算法可以分为两类，一种是基于计算流体力学来体现火焰粒子的流体性质，另一种则是使用特殊的粒子运动控制函数来表现出火焰粒子随机运动的特性，两者都是基于粒子系统[1]。第一种方法具有真实感好的特点，可以模拟出火焰在复杂湍流中的流动特性，但缺点是算法复杂度很高且数值稳定性很差[2]，既不适用于大规模仿真，也难以模拟出高流速的尾焰。第二种方法特点是简单易用，可以模拟任意流速的粒子，缺点是真实感较差，且在模拟高流速火焰时会出现撕裂现象，严重不符合实际观感。

本文的应用场景是大规模战斗机群尾焰仿真，因此需要同时渲染多个尾焰，需要解决大规模火焰仿真时系统效率问题和高流速尾焰撕裂现象。因此本文提出了一种改进方法。首先使用渲染管线构建了基于GPU加速的粒子系统，保证了大规模场景下，即使总体粒子数较高，也不会影响程序性能。根据发动机尾焰的特点，摒弃了尾焰的湍流特性，转而模拟其层流特性，使用效率很高的线性运动控制代替传统的运动控制方法来体现层流的特点，使得尾焰在高流速状态下保持火焰外形的稳定，克服了撕裂现象。使用了高斯噪声来扰动粒子运动，让尾焰具有更好的真实感。最后本文根据发动机尾焰燃烧特性，设计了热值衰变函数，使得尾焰的焰色实时衰变，与实际现象保持一致。

2. 火焰模拟算法分析

2.1. 基于 N-S 方程的火焰仿真算法

这种方法的主要思想是将火焰的不规则运动性质看作是流体中的湍流。而对于湍流的仿真方法目前

主要是通过解 N-S 方程来控制流体微元的运动状态[3]。N-S 方程的一般形式如下：

$$\rho \left(\frac{\partial V}{\partial t} + V \cdot \nabla V \right) = \nabla P + \mu \nabla^2 V + \rho F \quad (1)$$

然而求解 N-S 方程时间复杂度较高，目前最主流的方法是用 SPH 法，使用粒子来表示问题域，每个粒子都承载了一些物理属性，比如速度、加速度、质量、压力、密度、位置等。下面简述 SPH 算法在每个时间步里的主要步骤：对每个粒子查找所有粒子中的临近粒子，然后计算粒子的密度，再次查找粒子的临近粒子，再计算出压力、粘滞力和外力，然后更新粒子的加速度和速度，最终计算出粒子的位置[4]。

由此可以得知这种算法为了计算出一帧中粒子的位置，需要多次查询临近粒子，所以当系统中的粒子数目增大时，算法的性能会急剧下降，如果使用这种算法来模拟火焰外形，单个火焰需要的粒子数目在 4000 到 6000 个时能取得较好的仿真效果[5]。在本文的应用场景中，一个大规模战斗机群需要模拟的尾焰数量比较大，因此仿真效率极低。且计算流体力学在处理大变形流体时，一向具有不稳定的特点。过大的粒子流速需要极大的压强和粒子密度，很容易造成粒子的错误运动，出现无规则扩散和爆炸现象。所以这种基于计算流体力学的火焰仿真方法只适用于常规的火焰，粒子在较低的速度范围内运动，才能取得较好的仿真结果。

2.2. 基于动态纹理和粒子系统的火焰仿真算法

这种算法的主要思想是构建一个粒子系统，每个粒子具有诸如速度、位置、生命周期等性质。使用随机运动控制方程来让火焰粒子近似地模拟出火焰的随机运动状态。并根据火焰粒子的状态来动态地使用不同的纹理贴图来渲染火焰粒子[6]。

这种算法效率很高，但只适合模拟正常状态下燃烧的火焰，如果粒子运动速度加快，为了保证粒子的随机运动特点，火焰粒子会出现较大的速度差，火焰会呈现出明显的撕裂现象。本文需要模拟出发动机尾焰高速喷射的效果，这种算法难以模拟出真实度很高的尾焰。且火焰的焰色需要根据真实发动机燃烧特性来决定，故而这种简单的动态纹理绑定也不能满足要求。

3. 基于 GPU 加速的改进型高流速尾焰仿真算法

针对传统方法的不足之处，本文提出了一种基于 GPU 加速的改进型算法。主要从几个方面进行改进：首先是构建了有利于渲染结果做进一步图像处理的粒子系统，这样方便对火焰在多气象条件下进行融合仿真。其次是简化了火焰粒子的控制方程，使其在高速喷射状态下不会呈现撕裂现象，并使用高斯噪声扰动火焰，使其真实感进一步提高。最后是改进了粒子的渲染算法，根据发动机的燃烧特性设计了热值衰变函数，使得尾焰的焰色贴近于真实发动机尾焰的焰色。接下来将对这三个方面进行详细阐述。

3.1. 使用 GPU 加速的粒子系统

在 CPU 中实现的粒子系统的一般思想是，首先设计单个粒子的数据结构，然后用额外的数据结构来组织管理这些粒子。每个粒子具有自己的属性诸如位置、速度、加速度、生命周期等等。在 CPU 中实现粒子系统有着鲜明的优缺点。优点是可以方便地实现常用的数据结构，如链表、树、散列表等等。因为在 CPU 中往往是串行执行运算程序，在更新粒子的时候一般也不用考虑并行带来的数据同步问题，缺点则是串行执行带来的效率过低问题。对于粒子系统而言，每个粒子在不考虑与其他粒子互相作用的前提下是具有高度并行特性的，因此为 SIMD (单指令流多数据流)架构而设计的 GPU 是非常适合于实现粒子系统的。

CUDA 是 Nvidia 公司于 2006 年推出的通用并行计算架构。目前已经广泛地应用于科学计算，粒子

特效, 流体模拟, 通用计算加速等等领域[7]。CUDA 可以方便地处理常见并行计算问题。CUDA 的推出是因为使用传统的 GPU 加速计算的方法限制太大, 必须将问题伪装成图形学问题以加入图形绘制管线进行并行计算, 且输入输出的形式为纹理缓存, 这大大限制了可用于 GPU 加速的问题域。但是本文所涉及的算法并没有以上限制, 所以可以方便地使用传统图形绘制管线进行并行计算, 且这样带来的好处是可以不用将硬件局限于 Nvidia 的显卡, 从而减小了对硬件的要求。

实现本文中粒子系统的原理是将图形学中的“顶点”这一概念表示一个粒子。顶点上可以设置包括坐标、多重纹理坐标、颜色等属性, 这些属性可以被用于保存粒子所需要的值。通过粒子运动计算管线得到一张纹理缓存用于保存粒子的位置信息。在渲染绘制管线中取到每个粒子的位置信息进行坐标转换, 再对每个像素点进行渲染计算。

本文将粒子系统中的粒子转化为图形绘制管线中的“顶点”, 所以预先定义好所需数目的顶点。然后初始化粒子的各个属性。把顶点的顶点坐标设置为对应纹理缓存的二维纹理坐标。这样做的目的是使得 RTT (渲染到纹理)过程中, 让每个顶点最后能输出到纹理缓存的对应位置。本文将顶点 - 几何 - 片元着色器这一绘制流程称为一个绘制流。在粒子运动计算这个过程交替执行两个相同算法的绘制流, 这是因为用于保存粒子属性的纹理缓存既作为输入又作为输出, 这在一些显卡中可能会导致不可预知的错误, 因此使用两个绘制流轮流输出粒子属性纹理缓存。整个粒子系统的处理流程图如图 1 所示。

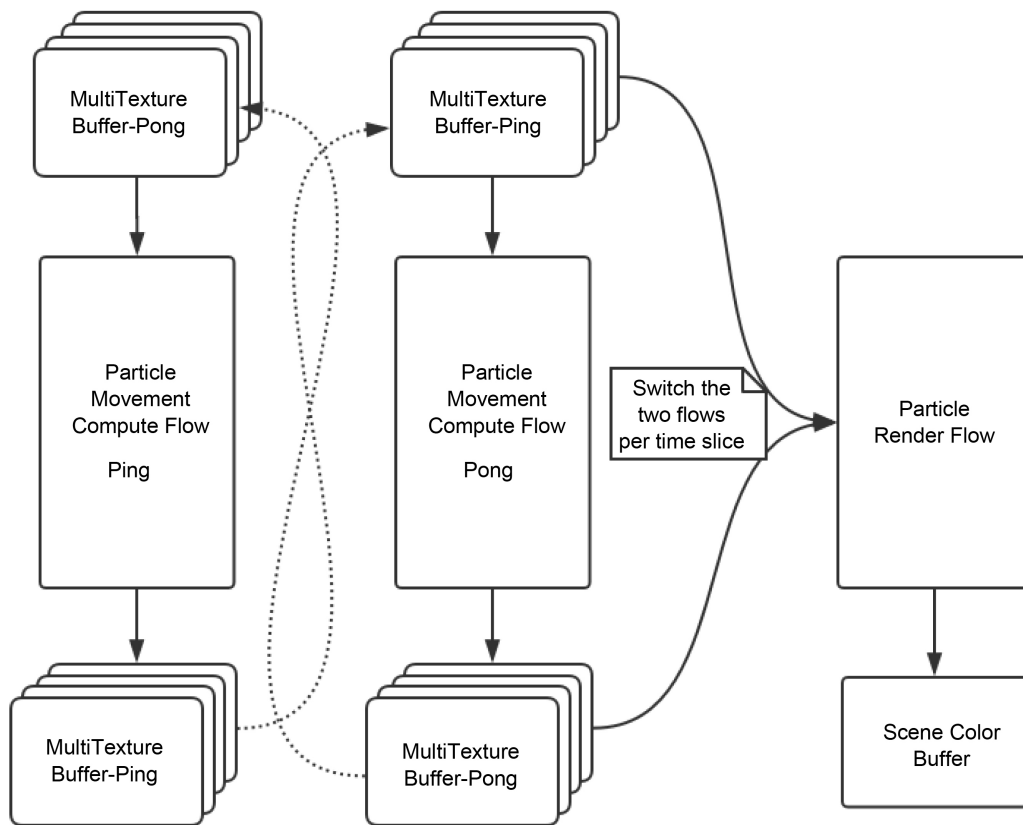


Figure 1. Parallel computing process of particle system
图 1. 粒子系统的 GPU 并行计算过程

在 CPU 程序中会每一帧切换绘制流的输入纹理, 即第 n 帧采用 Texture-Ping 作为输入, 第 $n + 1$ 帧采用 Texture-Pong 作为输入。并将输出的颜色值保存到名为 Scene Color Buffer 的纹理对象中。因此这个

尾焰的渲染结果将和场景渲染结果一起生成整个场景的图像，可以将尾焰的渲染结果和场景的渲染结果进行统一的图像处理，这样就保证了在使用一些基于图像处理的天气模拟算法(如雾、沙尘暴、雪、雨等)之后，尾焰的渲染效果能和场景保持一致。

本文将顶点坐标设置为纹理空间中的纹理坐标，这样做的原因是，为了保证粒子经过一个绘制流之后能将当前粒子的输出数据准确无误地保存到对应的纹理缓冲区中。下一个绘制流在对这个纹理进行纹理取样的时候就能取到对应粒子的数据。与此同时在多重纹理坐标中保存粒子的某些属性值，例如粒子的初始位置、生命周期等信息。

3.2. 尾焰粒子的喷射运动控制

除发动机尾部气体喷射的初始状态可以被近似地认为是层流而不是湍流。在尾焰的尾部一段距离后，由于周边气流的扰动，发动机尾部气流会快速形成湍流[8]，但是这部分湍流一般是不可见的，因此只需要模拟出可见部分的层流即可。

模拟层流意味着不需要复杂的粒子运动控制方程，粒子从发动机口线性运动到尾焰末端。为了模拟出发动机高速喷射的效果，本文使用高斯白噪声来扰动尾焰。因为高斯白噪声是一种理想的加性噪声，发动机尾焰的长度(FlameLength)总体上是在一个稳定的值附近无规则的上下波动，故可以使用高斯白噪声来扰动预设好的尾焰长度，让它每帧都产生随机变化。最后观测到的火焰喷射的效果实际上是由FlameLength每一帧的变化导致的，所有粒子之间的相对位置并没有变化，粒子的绝对位置会随着当前帧的FlameLength来进行调整。粒子的空间排列如图2所示。

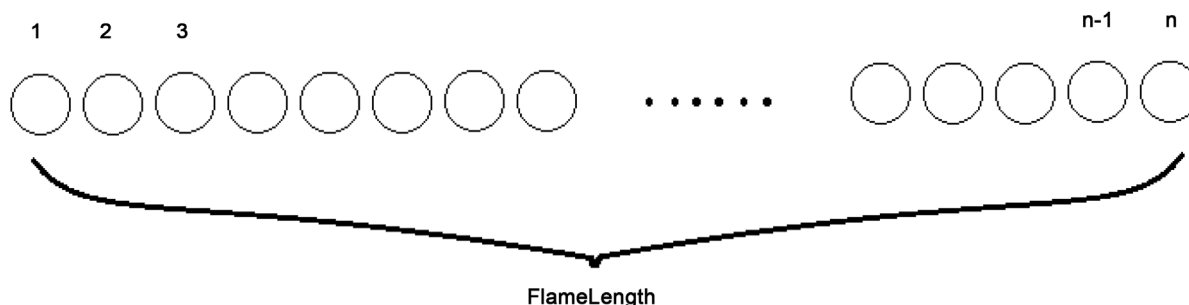


Figure 2. Particle spatial distribution

图2. 粒子空间分布

一种快速生成高斯随机噪声的方法为：先在一定范围随机生成一个随机数，然后查找高斯分布映射表，以生成一组高斯噪声随机数[9]。本文中 will 使用一种更为便捷快速的方法。高斯白噪声的本质是一组均值为0且噪声幅值的概率满足正态分布(高斯分布)，因此只要生成一组随机数，使得这组随机数值的概率满足正态分布。在C++11中引入了正态分布模板类，它的本质是在计算一个随机变量 x ，在均值为 μ ，方差为 σ 情况下的概率密度函数。公式如下：

$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

在程序中只需要给定预设好的方差，并把均值设为0，配合使用C++标准库中的随机数生成器就可以方便的生成这样一组数据。对这组数据的采样过程不需要放到GPU中进行，因为这个值对于所有粒子来说是统一的，所以只在CPU中每帧对这组数据逐次取样，最后将取样得到的值以uniform值的形式传给GPU。

3.3. 火焰粒子的渲染

本文渲染火焰粒子的方法为：在顶点着色器中对 Ping-Pong Flow 中输出的纹理进行采样，粒子坐标纹理采样值经过相机坐标系坐标变换和投影变换即得到用于输出的坐标。在几何着色器中将图元展开成一个四边形，以便于在片元着色器对这个张开的粒子进行贴图。纹理贴图如图 3 所示。

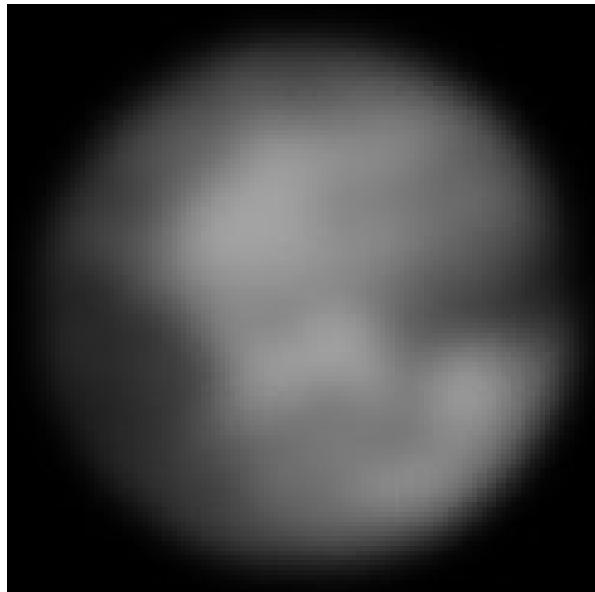


Figure 3. Particle texture map
图 3. 粒子纹理图

对这个纹理采样得到一个四维向量，只需要使用该向量的第四个分量，即颜色的透明度 α 值。在当前的片元着色器中，需要最终输出该像素的颜色值。尾焰从喷口到消失这一过程颜色变化主要体现在颜色的纯度变化，因此使用颜色值的透明度 α 的变化来体现尾焰温度的变化。可以近似地认为，透明度与火焰热度正相关。即温度越高，透明度 α 值越高，也即焰色纯度越高。

本文根据热流热度与距离函数图像的数值关系以及在实验中通过实验结果反复调整[10]，最终确定了 α 的计算函数， α 的计算公式如下：

$$\alpha = k \cdot ParticalTex \cdot (1 - Dis/FlameLength)^n \quad (3)$$

Dis 是粒子和发动机喷口间的距离， $FlameLength$ 为尾焰长度， $ParticleTex$ 为粒子纹理图采样值。 k 和 n 为调节参数，该公式计算出来的 α 值总体上符合热值距离函数。由于粒子采样图是一个类似高斯函数的图像，这就意味着当前粒子最终 α 的空间函数图像虽然形状和其他粒子类似，但形状会越来越尖锐，体现在实验结果中的现象是火焰的尾部越来越尖，这也与热流密度关于平面位置的函数图像相契合。

在计算完 α 值之后，还需要计算颜色的 RGB 值，本文设计了一种色值衰变函数，在发动机喷口的火焰与焰尾的区别不只是透明度的差别，颜色也在改变。在喷口出火焰温度最高，颜色接近于白色，然后渐变成火焰的常态焰色。这个衰变过程与 α 的衰变过程略微不同。

$$\mathbf{a} = \mathbf{c} - (\mathbf{c} - \mathbf{b}) \cdot (1 - Dis/FlameLength)^n \quad (4)$$

发动机尾焰的颜色主要由燃料成分、发动机燃烧效率和喷口材质等因素有关。向量 \mathbf{b} 为发动机预设的尾焰末端焰色值， \mathbf{c} 为最高温度颜色值，向量 \mathbf{a} 为计算出来的当前像素点最终焰色。

4. 计算机仿真实验及分析

使用基于计算流体力学(CFD)的算法进行仿真,求解算法为 SPH 法,用 OpenGL 实现的 2D 效果的仿真结果及分析如下图 4。

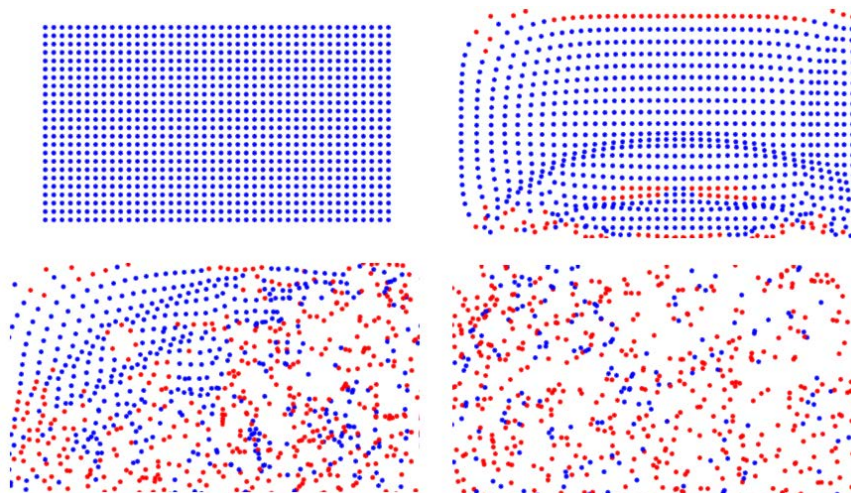


Figure 4. SPH particle state in four time steps

图 4. SPH 粒子在四个时间步的状态

实验中粒子密度为 2 倍标准值,粒子在短时间内就发生了剧烈的不稳定。若要用此方法模拟发动机尾焰的高流速粒子,需要较高的粒子密度来产生较大的压强,远远高于实验中设置的密度和压强。在这种情况下极易发生数值不稳定,造成错误的计算结果,粒子会出现无规律飞溅或是爆炸的现象。

使用非 CFD 传统粒子运动控制方法的仿真结果如下图 5 [11]。



Figure 5. Simulation results of non-CFD traditional method

图 5. 非 CFD 传统方法仿真结果

非 CFD 传统在粒子运动速度较快时，各个粒子容易出现较大的速度差，火焰会出现撕裂现象，且焰色变化较为单一，不能准确模拟出发动机尾焰焰色特点。

本文中改进算法的仿真结果及分析如下：

计算机仿真实验环境为：

CPU：酷睿四核、2.5 GHz；

GPU：gtx780，显存 3 G；

内存：4 G。

渲染软件：基于 OSG 开发的渲染引擎。

仿真结果如下图 6~11 所示。

从仿真结果可以看出，尾焰在高速喷流状态下保持了尾焰的形状特性，没有撕裂现象。在雾天下能融入背景环境。在大规模战斗机集群同时开启尾焰特效的状态下，开启喷火特效和关闭喷火特效帧率没有明显的变化，帧率稳定在 60 帧左右。单个火焰粒子数在 500 个左右就能达到较好的真实感。



Figure 6. Real engine fire

图 6. 真实发动机喷火



Figure 7. Tail flame simulation

图 7. 尾焰仿真效果



Figure 8. The small formation opens the tail flame
图 8. 小型编队开启尾焰



Figure 9. Simulation effect of tail flame under foggy conditions
图 9. 雾天条件下尾焰仿真效果



Figure 10. Before the large formation opens the tail flame
图 10. 大型编队开启尾焰前



Figure 11. After the large formation opens the tail flame
图 11. 大型编队开启尾焰后

本文改进算法使用线性运动控制函数克服了非 CFD 传统仿真方法的容易出现撕裂现象的缺点，即使在高流速状态下，尾焰仍保持着完整的外形，使用噪声扰动尾焰也使得尾焰更符合发动机高速喷射气流的无规则变化现象。同时由于利用发动机尾焰层流的特性，避免了基于 CFD 的仿真方法的数值不稳定性 and 低效率的缺点。整个尾焰能长时间稳定高效的进行渲染。即使当场景中同时渲染多架战机时，总粒子数仍然较低，系统性能不会发生较大的波动。本文改进算法设计的基于 GPU 加速的粒子系统，在高效进行并行计算的同时，将所有的渲染结果输出一张纹理中，极大地方便进一步的图像处理和多渲染结果的融合。例如雾天或是雨天等天气情况。

5. 结论

本文通过使用为 GPU 而设计的粒子系统达到了加速粒子运动计算的目的，通过分析发动机燃烧特性，设计了合适的色值衰变函数模拟出真实尾焰的焰色变化。同时保持了尾焰在高速喷流状态下的形态稳定。最终将渲染结果输出到图像空间，方便进一步的图像处理，如雾效等。粒子数目被控制在较低的范围，也保证了大规模渲染时系统性能的稳定。

参考文献

- [1] 王双双, 许化强, 白成杰. 基于数学物理方法和纹理作用的火焰模拟[J]. 计算机与数字工程, 2016(2): 197-199, 206.
- [2] 徐蕊芸, 蒋涛. 浅谈三维广义牛顿流体复杂流动过程修正 SPH 方法模拟[J]. 科技创新导报, 2017, 14(20): 255-256.
- [3] Wang, F., Liu, R., Dou, L., Liu, D.H. and Jin, J. (2019) A Dual Timescale Model for Micro-Mixing and Its Application in LES-TPDF Simulations of Turbulent Non-Premixed Flames. *Chinese Journal of Aeronautics*, **32**, 875-887. <https://doi.org/10.1016/j.cja.2019.01.005>
- [4] Duan, X.F., Ren, H.X., Qiu, S.Y. and Shen, H.L. (2019) Semi-Implicit Incompressible SPH Fluid Simulation in a Marine Simulator. *International Conference on Graphic and Image Processing*, Volume 11069, Article ID: 1106903. <https://doi.org/10.1117/12.2524398>
- [5] 鲍宁. 一种基于粒子系统的交互式导弹尾焰模拟方法[D]: [硕士学位论文]. 长春: 东北师范大学, 2016: 29.
- [6] 胡彭辉, 胡爱兰, 刘吉娟, 等. 基于 Direct3D 和粒子系统的导弹尾焰模拟[J]. 微型机与应用, 2016, 35(14): 83-85, 88.
- [7] 吴辉, 罗清海, 彭文武. GPU 并行计算的 CUDA 架构浅析[J]. 教育教学论坛, 2019(6): 277-278.

-
- [8] Aspden, A.J., Day, M.S. and Bell, J.B. (2019) Towards the Distributed Burning Regime in Turbulent Premixed Flames. *Journal of Fluid Mechanics*, **871**, 1-21. <https://doi.org/10.1017/jfm.2019.316>
- [9] 吴国望, 屈晓旭, 徐丹. 基于 FPGA 的高斯白噪声信号源实现[J]. 舰船电子工程, 2016, 36(6): 119-121.
- [10] 白杰, 杨毅成, 王伟. 航空发动机防火试验火焰特征数值仿真分析[J]. 燃气涡轮试验与研究, 2017, 30(5): 1-7.
- [11] 魏碧云, 孔样红, 李晖. 基于粒子系统的 3D 动态火焰模拟[J]. 武汉工程大学学报, 2018, 40(1): 87-92.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-2286, 即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: sea@hanspub.org