

计算机动画 碰撞检测作业 中期文档

软件 03 陈启乾 2020012385

作业要求

1. 需要实现一种基于 GPU 的快速大规模碰撞检测算法。
2. 测试分析算法的性能。
3. 将算法应用在如下的应用中：一个固定场景中有大量小球或者物体。各个小球有不同的半径、质量、初速度和弹性系数，利用所实现的最近邻查找算法对小球或物体的运动和碰撞进行仿真，制作一段动画。（动画制作可以自己完成，也可以使用现有的软件进行渲染。）

说明碰撞检测的加速算法设计

朴素的碰撞检测算法中，我们会对所有物体两两之间进行碰撞检测，这样的话需要进行 $O(n^2)$ 次碰撞检测。在 n 很大的时候，这样的算法效率会比较低。通过一个两阶段的碰撞检测算法

1. 粗检测阶段：在这一阶段中，我们希望能够通过较为简单、易于维护的数据结构，尽可能减少需要进行检测的物体对的数量
2. 细检测阶段：在这一阶段中，我们对粗检测阶段中判断对进行精确的碰撞检测，并用 GPU 进行并行化加速

粗检测阶段

因为题目中要求的是大量小球和物体，可以假设物体大小相差不会很悬殊，而且大多数物体都是比较规则的几何形状。

因此，我们打算采用基于均匀网格技术的粗检测方法。假设所有的物体中，AABB 包围盒最大的大小为 d ，则我们在整个空间构造宽度为 $2d$ 的网格，每个网格的大小为 d 。这样的话，我们可以将所有的物体放入网格中，每个网格中的物体数量不会太多。

因为物体的大小不会超过网格大小，所以一个物体最多只会和其所在的网格为中心的 $3 \times 3 \times 3$ 个网格中的物体发生碰撞，在这些之外的物体就不会在碰撞检测的考虑范围内。

假设物体均匀分布，那么我们就只需要进行 $O(n)$ 次碰撞检测。

细检测阶段

在细检测阶段中，我们对粗检测阶段中判断对进行精确的碰撞检测，并用 GPU 对不同物体之间的碰撞，进行并行化加速。

我们假设在场景中，只会出现两类物体：球体和正方体。我们对球体考虑特殊处理，对于除了球体之外的三维物体，我们用三角面片来近似表示。

因此，我们只需要考虑如下几种情况：

1. 球体和球体之间的碰撞
2. 球体和三角面片之间的碰撞
3. 三角面片和三角面片之间的碰撞

对这些碰撞，都有非常成熟的基于距离的碰撞检测算法，可以在 $O(1)$ 的时间内完成。

说明 GPU 实现的思路设计

本项目采用 WebGPU 技术，使用 Rust 语言的 wgpu 库实现跨平台的 GPU 程序。基于以上的快速碰撞检测算法，我们设计一个并行、对 GPU 架构友好的实现。

在粗检测阶段, 我们希望维护一个索引数组, 在索引数组中, 我们把所有物体的索引按照其所在的网格进行排序; 除此之外, 我们还希望维护每一个网格在索引数组中的起始位置和终止位置。

具体来说, 假设共有 N 个物体, m 个网格。

我们需要维护长度为 N 的索引数组 $\text{index}[i]$ 。我们希望 $\text{index}[i] \neq \text{index}[j], \forall i \neq j$ 而且 $\forall i, 0 \leq \text{index}[i] < N$ 。

我们还需要维护两个数组 $\text{start}[i]$ 和 $\text{end}[i]$, 其中 $0 \leq i < m$, 表示第 i 个网格中的物体在物体数组中的起始位置和终止位置, 并且应该满足 $\text{end}[i - 1] = \text{start}[i]$ 且 $\text{end}[m - 1] = N$, 这个时候满足 $\forall i, \forall \text{start}[i] \leq j < \text{end}[i]$, 第 j 个物体在第 i 个网格中。

在每次更新的时候, 实现步骤如下:

1. 以 Thread Group 为计算的单元, 对网格并行处理。对每个网格, 我们会并行计算每个网格中的物体, 经过位置的更新后所在的网格的编号。
2. 接下来我们需要计算每个网格中的物体的个数, 构建 start 和 end 数组。我们维护全局的 size , 然后利用 atomic 算子去计算全局的 size 。
3. 接下来我们把对应的索引填入到索引数组的对应的位置。这个我们通过分块计算前缀和来计算得到每个位置所在。

在细检测阶段, 我们仍然以 Thread Group 为计算的单元, 对网格并行处理。对每个网格中的所有元素, 我们遍历其周围网格的所有元素, 在 GPU 上进行具体的碰撞检测算法。

参考文献

1. R. Weller, “A Brief Overview of Collision Detection,” in New Geometric Data Structures for Collision Detection and Haptics, R. Weller, Ed., in Springer Series on Touch and Haptic Systems. , Heidelberg: Springer International Publishing, 2013, pp. 9–46. doi: 10.1007/978-3-319-01020-5_2.
2. 用 39 行 Taichi 代码加速 GPU 粒子碰撞检测: <https://zhuanlan.zhihu.com/p/563182093>
3. 空间划分算法优化碰撞检测研究: <https://blog.csdn.net/yhn19951008/article/details/119899092>
4. wgpu - Rust: <https://docs.rs/wgpu/latest/wgpu/>