

浙江大学

本科实验报告

实验课题: 基于粒子系统的黑洞爆炸模拟

学院: 计算机学院

专业: 数字媒体技术

姓名: 王维维

学号: 3110105133

指导教师: 金小刚, 冯结青

一、实验目标和要求

(1)目标

在学习和掌握计算机动画学中的粒子系统的基本原理的前提下,运用这些原理在 opengl 中创建一个较为完整的粒子系统（通过模拟实现万有引力来控制宇宙中星体粒子的运动）,模拟出黑洞爆炸的效果。

(2)要求

- a. 创建出一个完整的粒子系统,赋予每个粒子独立的属性,使粒子系统中的粒子能够有规律地在黑洞的引力下进行运动。
- b. 粒子在引力的作用下运动的轨迹要平滑,不能杂乱无章。此外,粒子爆炸时要充分模拟出黑洞爆炸效果,要有一定的视觉冲击力。
- c. 提供一定的交互方式,使的程序能够根据不同的输入参数绘制出不一样的动画效果。

二、开发环境与技术说明

(1)开发环境

IDE: QT Creator, MINGW

第三方库: opengl

语言: C++

(2)粒子系统简介:

粒子系统是最实用的过程动画技术之一,是影视特技中生成视觉特效的一种主要方法。它采用了一套完全不同于以往的造型、绘制方法来构造、绘制景物,造型和动画巧妙地连成一体。景物被定义为由成千上万个不规则的、随机分布的粒子所组成,而每个粒子均有一

定的生命周期, 速度, 合外力等属性, 使得它们不断改变形状、不断运动。粒子系统常用来模拟表现三维计算机图形学中一些特定的模糊现象, 如: 火、爆炸、烟、水流、火花、落叶、云、雾、雪、尘、流星尾迹或者象发光轨迹等等这样的抽象视觉效果。

三、粒子系统模拟黑洞爆炸的算法实现过程

(一)准备工作

(1)配置好 qt creator 以及 opengl 等开发环境

(2)学习粒子系统实现的基本原理, 星体运动及黑洞爆炸等基本知识

(3)本次实验最终的动画有两个动画场景, 一是许多发光的星体粒子因为黑洞的强大引力而被黑洞吸收的过程; 二是黑洞因内部能量失衡而发生爆炸的场景。

这两个场景中, 受黑洞引力作用而运动的星体可以用同一种粒子实现, 而黑洞这个星体则单独处理。

星体粒子类的定义如下:

```
class Particle
{
    //.....//
    //粒子的主要属性
    float mPos[3]; //粒子的三维空间位置
    float mColor[4]; //粒子的颜色属性(RGBA)
    int mTTL; //粒子的生命周期 (单位: 帧数)
    float mSpeed[4]; //运动速度 mSpeed[0]为速率, 下标 1, 2, 3 的分别对应 x, y, z 方向的矢量速度
    float mForce[4]; //力累加器 mF[0]为合力, 下标 1, 2, 3 的分别对应 x, y, z 方向的矢量力
    float mM; //粒子的质量
    float mSize; //粒子尺寸大小
    bool mVisible; //粒子的可见性
    long random_seed; //生成随机属性的种子
};
```

(二)算法步骤

(1)指定一个三维坐标, 在这个坐标点生成一个黑洞, 要求质量较大, 足够产生强

大的吸引力

(2) 在黑洞周围的三维空间中, 随机生成一系列大小, 质量, 颜色等属性各异的星体粒子, 赋予一定范围内的随机初速度, 粒子便开始在黑洞的引力下运动

(3) 粒子在运动的过程中的每一帧都要更新粒子所受的引力大小和方向, 从而计算出粒子的速度和空间位置。(计算公式即万有引力公式 (其中的参数 G , M , m , r 的取值做了适当的调整以方便在计算机中的运算和显示))。

(4) 粒子在运动过程中, 若接近了黑洞的吸收范围, 粒子会变为黑洞的一部分从而消失。此外, 粒子还具有一定的生命周期, 每次运动生命都会减少。若生命为 0 时, 则不管粒子是否被黑洞吸收, 也会消失。

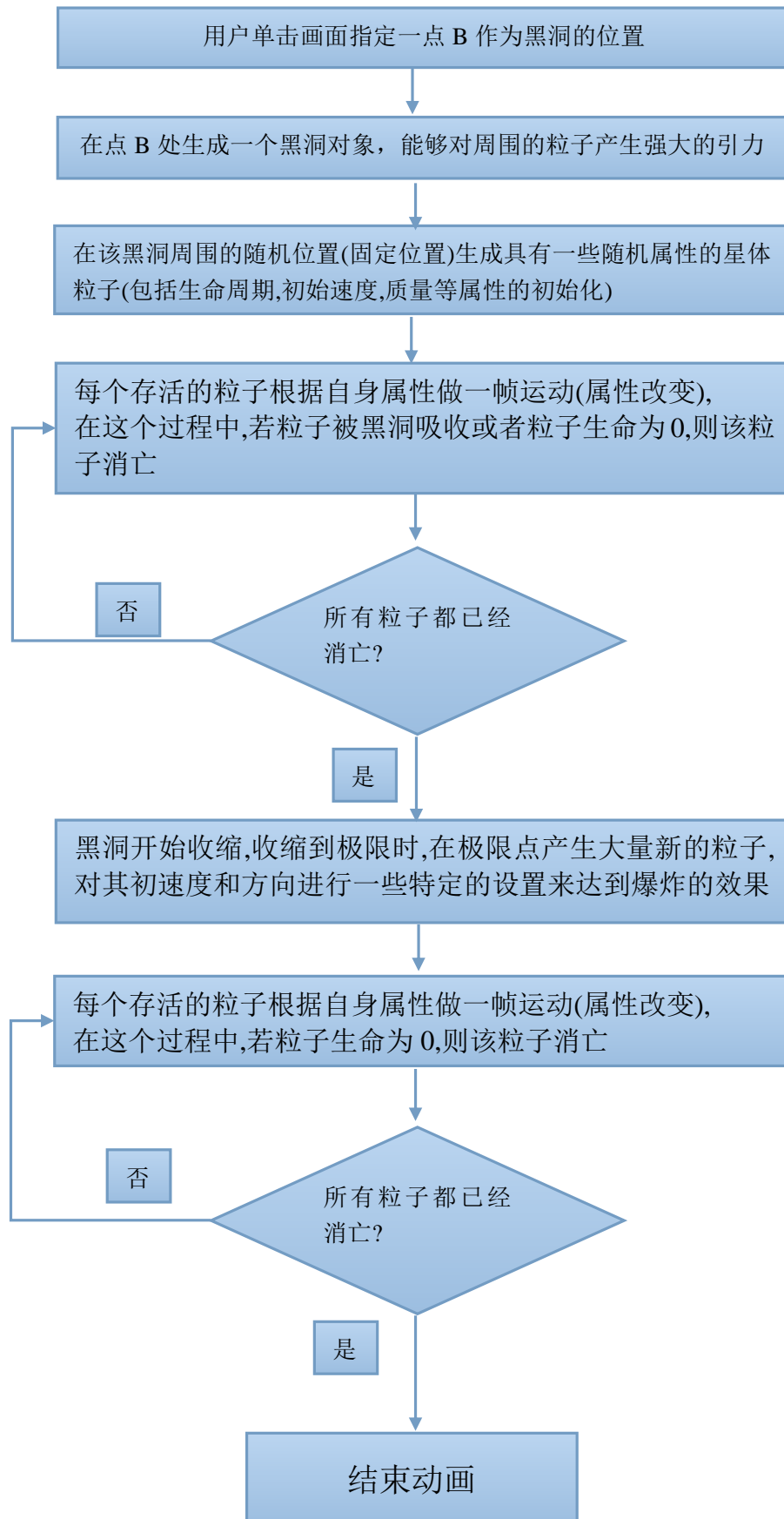
(5) 当所有粒子消失后, 黑洞便已经吸收完了所有的粒子, 动画便进入下一个场景, 黑洞的爆炸阶段。

(6) 黑洞首先进行收缩, 当黑洞收缩到极限时 (本实验设定黑洞收缩到半径为 0 时), 便会发生爆炸。在爆炸前对爆炸产生的粒子进行初始化。因为黑洞已经消失, 因此, 在忽略宇宙中其他星体的影响下, 爆炸产生的粒子应该是向各个方向做初速度很大的匀速运动。因此, 初始化时只需指定粒子各方向的速度即可。

(7) 在初始化时, 也赋予粒子新的生命周期, 在粒子运动的过程中, 生命也在不断减少, 当生命为 0 时, 粒子消失。

(8) 这样我们便完成了粒子模拟黑洞爆炸的算法设计, 接下来的工作便是通过代码实现这些功能

(二)算法框图



(三)关键代码说明

(1)黑洞类的定义

实验是用 opengl 进行图形的绘制,黑洞是用一个球体表示。因为黑洞与其他星体粒子均为宇宙中的星体,并且都受万有引力定律的制约,因此黑洞类与粒子类的属性大致相同,但也有所不同。因为在模拟黑洞爆炸动画汇总,黑洞的位置是不会变化的也忽略外界粒子对它的引力作用,因此没有速度和力累加器的属性。其代码如下:(CneterObj 即黑洞类)

```
class CenterObj
{
    float mPos[3]; //黑洞的空间位置
    float mColor[4]; //黑洞的颜色,数组中四个变量分别对应 RGBA
    float mM; //黑洞的质量
    float mSize; //黑洞的尺寸
    bool mVisible; //可见性
    float mExpendSpeed; //黑洞膨胀或收缩的速度
};
```

(2)tick()函数

包括粒子和黑洞类中都会有 tick()函数的定义,此函数为自定义函数。主函数每过一帧都会调用每个类的 tick()函数,而所有类(包括粒子和黑洞)的每一帧的动作都在自身的 tick()函数中完成。

下面是粒子类的 tick 函数

```
void Particle::tick(bool speed_limit)
{
    if(mTTL == 0 || (mTTL > 0 && --mTTL==0))
        return;
    for(int i = 0; i<4; ++i)
        mSpeed[i] += (mLastForce[i]/mM + mForce[i]/mM) / 2 * SPF; //计算出当前时刻的速度
    //每一帧粒子根据自身参数自己运动
    float adjust_avg_speed[4] = {0};
    for(int i = 1; i<4; ++i){
        adjust_avg_speed[i] = mSpeed[i] + (mForce[i] / mM)/2 * SPF;
        if(speed_limit){
```

```

        if(adjust_avg_speed[i]>MAX_AVG_V){
            adjust_avg_speed[i] = MAX_AVG_V;
        }
        else if(adjust_avg_speed[i]<-MAX_AVG_V){
            adjust_avg_speed[i] = -MAX_AVG_V;
        }
    }
}
mPos[0] += adjust_avg_speed[1]*SPF;
mPos[1] += adjust_avg_speed[2]*SPF;
mPos[2] += adjust_avg_speed[3]*SPF;
}

```

(3)paint()函数

和 tick () 函数一样，每个类都会有自己的 paint () 函数的定义。主函数每过一帧就会更新一次画面，在更新画面时，会调用每个类的 paint () 函数。而所有类（包括粒子和黑洞）的绘制方法都在自身的 paint () 函数中完成。

下面是粒子类的 paint 函数：

```

void Particle::paint()
{
    if(mVisible){
        glColor4f(mColor[0],mColor[1],mColor[2],mColor[3]);
        glPointSize(mSize);
        glBegin(GL_POINTS);
        glVertex3f(mPos[0],mPos[1],mPos[2]);
        glEnd();
    }
}

```

有了 paint () 和 tick () 函数，主程序便可以很方便地控制每个粒子和黑洞的运动状态和以及方便进行不同对象的绘制。若要修改粒子的形状，不必大费周章地去主程序中修改，只需在粒子类中的 paint () 修改相应的绘制语句即可。

(4)主程序中的控制动画的主要的功能函数

```

void initPartices();//初始化粒子
void initExploding();//初始化爆炸前的粒子
void initShrinking();//初始化黑洞吸收前的粒子
void updatePartices();//每一帧更新粒子状态

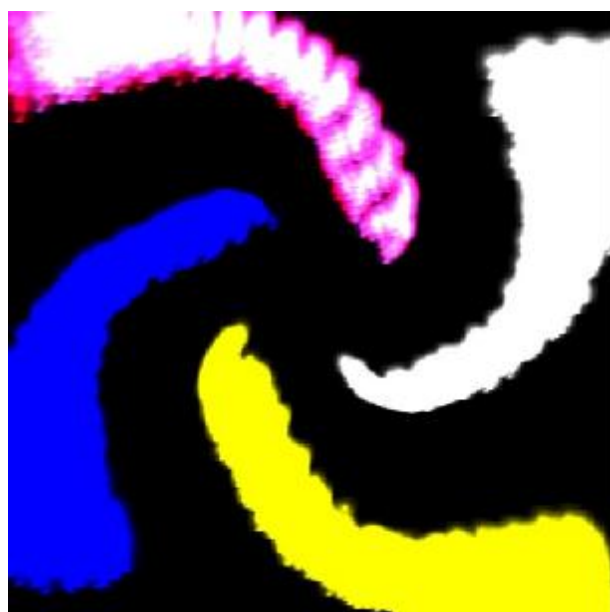
```

四、实验结果与分析

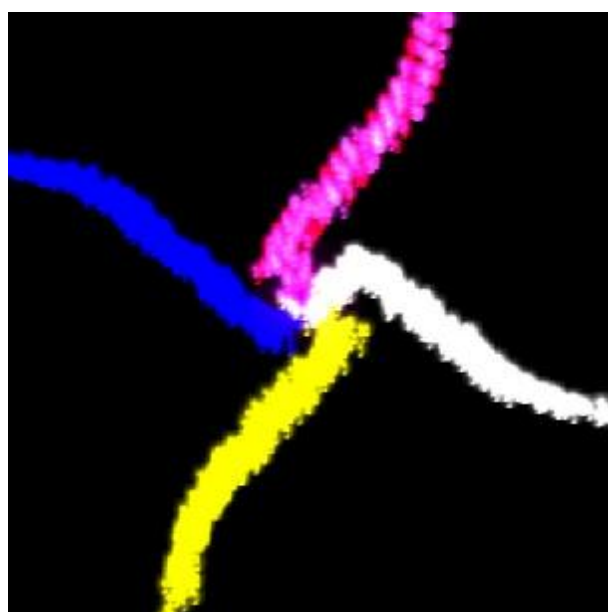
(一)实验结果： 本实验成功完成了基于粒子系统对黑洞爆炸的模拟效果。下面是此次实验的结果视频的截图，若想看完整的动画，请执行附带的可执行文件或者打开附带的视频文件进行观看。

黑洞吸收粒子时的动画截图：

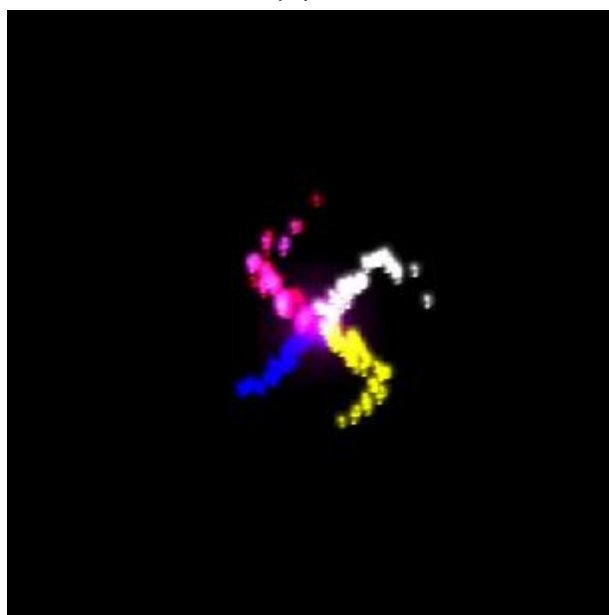
(1)



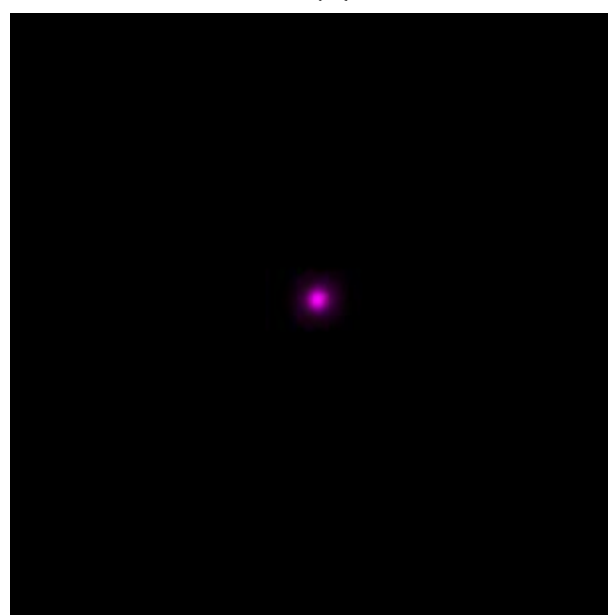
(2)



(3)

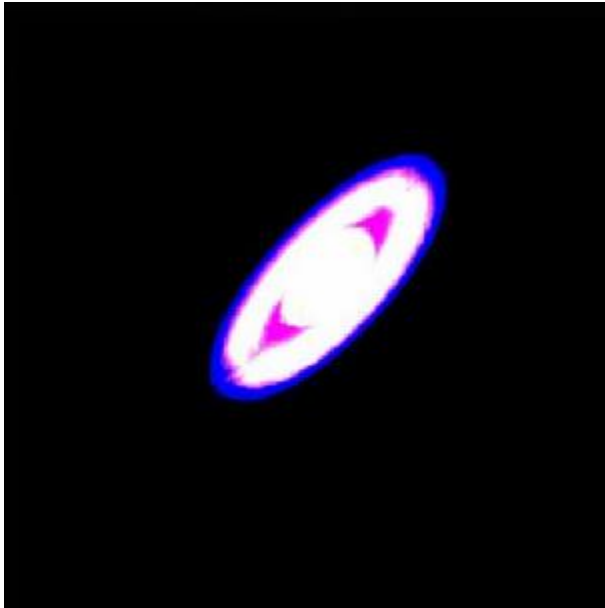


(4)

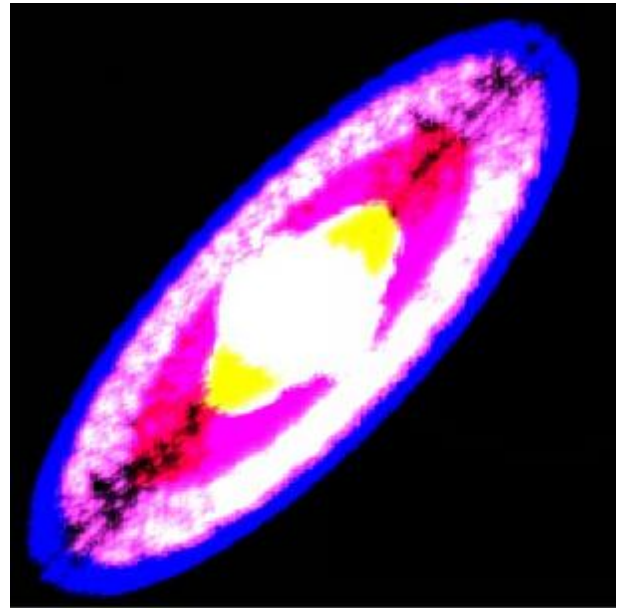


黑洞爆炸时的动画截图：

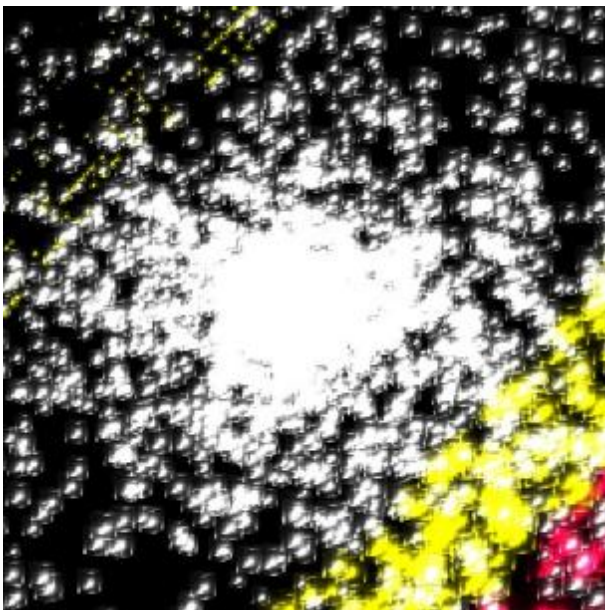
(1)



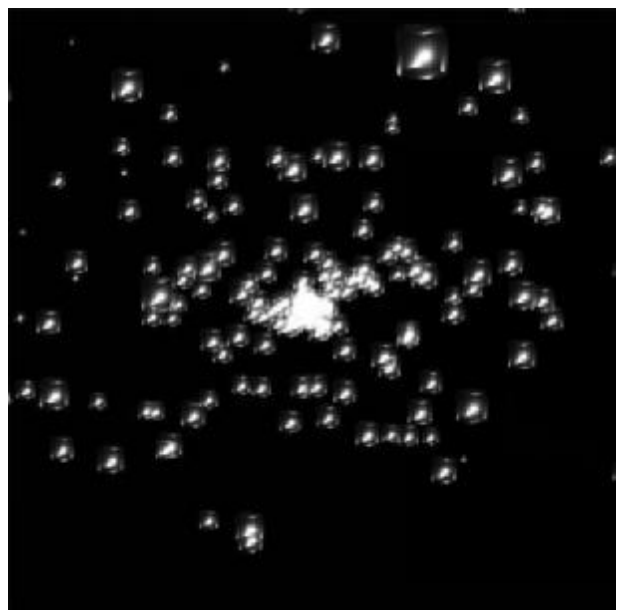
(2)



(3)



(4)



(二)问题与分析:

经过反复地调试和修改代码，成功地完成了此次粒子动画的制作和显示。因为是粒子动画，因此最终的成果并不是单一的一个视频。用户

能够通过单击鼠标指定黑洞的初始位置,这样便能形成在不同位置的爆炸。此外,程序中还有许多参数可以调节从而控制动画中的粒子的运动速度从而影响到运动的轨迹以及爆炸设定等相关参数。

此次实验中也遇到了不少问题,其中最主要有两个问题。

(1)粒子的运动方程的求解

因为星体粒子受到的是黑洞的万有引力,而此力是随粒子运动的位置变化而不断变化的(包括大小和方向)。动画是逐帧显示的,已经将时间离散化了。因此需要一个合适的求解运动过程中的位置变化的方法。在实验的前期,由于没有注意到此类问题,因此采用的是普通的欧拉积分法($X(t+h) = X(t) + V(t) * h$),这样导致的问题便是相邻帧的间隔时间越大,计算误差越大,粒子的运动不平滑。后来注意到了此问题,便采用 Verlet 积分方法进行位置的计算。在 Verlet 积分中,

$$V(t+h) = V(t) + 1/2 * (a(t) + a(t+h));$$

$$X(t+h) = X(t) + (V(t) + 1/2 * a(t)*h)*h$$

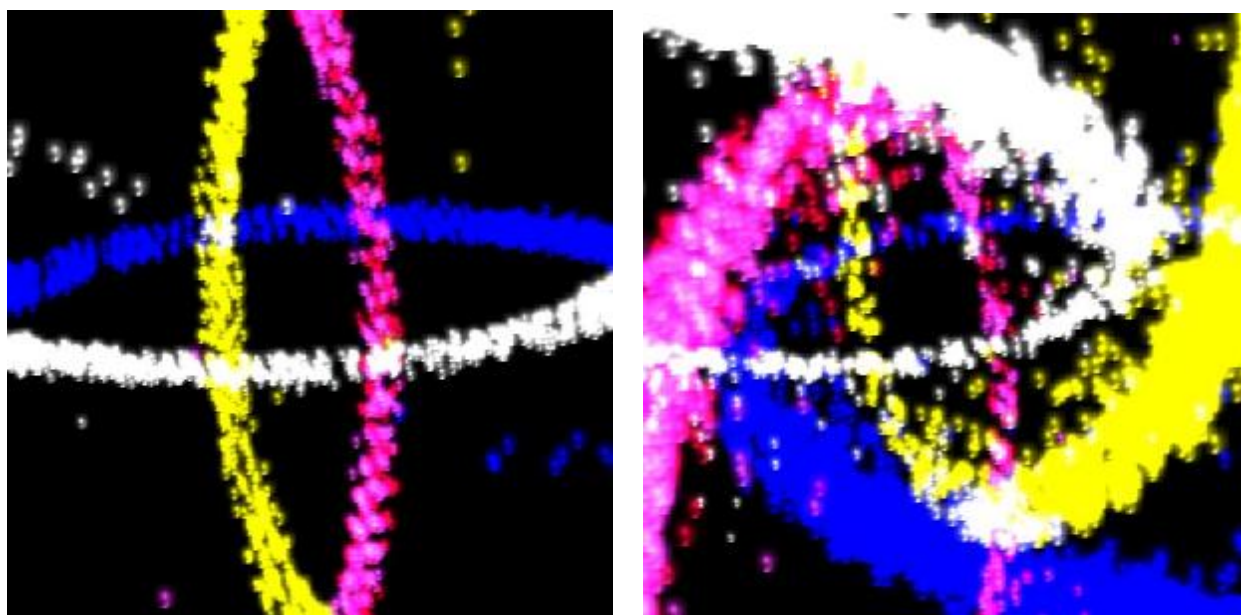
这样,粒子的运动便能够较为流畅。

(2)粒子的参数设定

实验中主要的力是由万有引力定律产生的引力,随着粒子到黑洞的距离缩小,这种力也在不算变大,但是同时粒子的速度也在不断增大。

如果粒子的速度过大,那么基于离散化时间来产生的计算机动画可能会将粒子直接移至远方,从而有可能脱离黑洞的控制,这显然是与现实不符的。因此,实验中设定了一个最大的速度上限,这个上限能够保证物体一帧运动的长度不会直接超过黑洞的控制范围。这样虽然违

背了一些物理规律，却解决了另外一种较为明显的问题，是可以接受的。此外，速度达到一定程度，这种变化是很难被观看者所察觉的。下面两张图是不设置最大速度限定的结果(粒子无法收拢或收拢时间较长)



五、总结体会

此次实验完成制作了基于粒子系统模拟黑洞爆炸场景的动画。在报告中阐述了粒子系统理论的主要思想和方法。此外，也详细介绍了粒子系统的创建步骤和黑洞爆炸模拟的算法实现步骤。

从算法的实现上来讲，此次实验已经达到了预期的目的，成功地用粒子系统模拟出了黑洞吸收星体以及黑洞的爆炸场景。这样的画面效果差强人意，却并没有科幻大片中的爆炸场景那么真实和震撼。除了表现手法上的不足之外，还有一个重要的原因便是粒子数量的稀少。本实验中一共使用了 10000 个粒子，这样每次更新画面便需要至少绘制 10000 次。因为需要实时显示形成动画，所以要提高绘制的效率。

此实验中采用 opengl 中的立方体(结合 alpha blend 进行贴图)来代替粒子在个人电脑上运行尚且有些吃力,如果用更为复杂的形状如球体代替点,则画面不能够流畅地显示,也就形成不了动画。如果运行的机器足够好,那么可以适当增加粒子的数量以及修改粒子的形状(如球体),相信效果会上一个层次。