## 选题

本项目实现了二次曲面可视化，采用光线投影法（ray casting）。

## 编程环境

|  |  |
| --- | --- |
| 操作系统 | Linux Mint 18 Cinnamon 64-bit |
| Linux 内核版本 | 4.4.0-21-generic |
| 处理器 | Interl© CoreTM i5-3230M [CPU @ 2.60GHz](mailto:CPU@2.6GHz)×1 |
| 内存 | 2.0 GiB |
| 显卡 | 本项目不使用GPU |
| 编程语言 | C++ |
| 编译器 | gcc 5.4.0 |
| 依赖库 | OpenGL版本 3.0 Mesa 17.0.7  Freeglut 3  AntTweakBar：<http://anttweakbar.sourceforge.net/doc/> 我们使用AntTweakBar来制作简单的交互系统 |
| 项目构建工具 | CMake 3.5.1 |

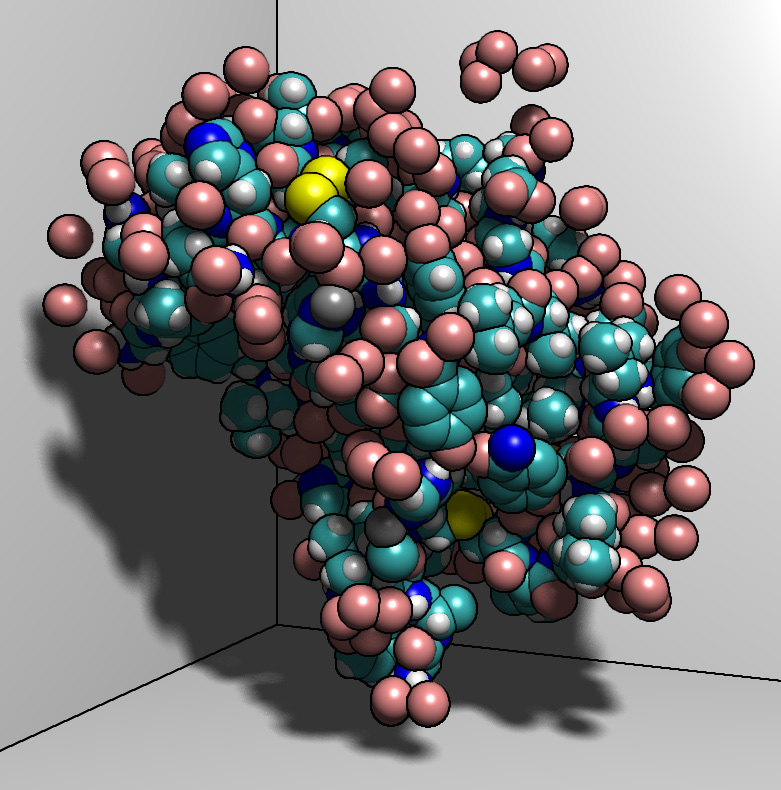
## 实现目标

本项目目标是实现一个使用正交投影绘制任意形如以下数学形式的三维二次曲面的系统。

并且能交互性地查看它。

## 算法原理

本项目参考了论文[1]。该论文采用了GPU来实时绘制多个二次曲面。如下图所示。



而本项目只需绘制一个二次曲面即可，所以只采用CPU即可。

首先，方程可以写成二次型, 其中是一个对称阵。

在屏幕空间，我们固定x,y, 对应屏幕上的某个像素点。则可以将上述方程转化为一个关于z的二次方程。如下所示：

的情况需要我们特殊考虑。如果该二次方程没有实数解。那么在该像素点上则不绘制。否则，解出z以后，我们可以计算法向量。

然后我们使用Phong shading来绘制该二次曲面 。

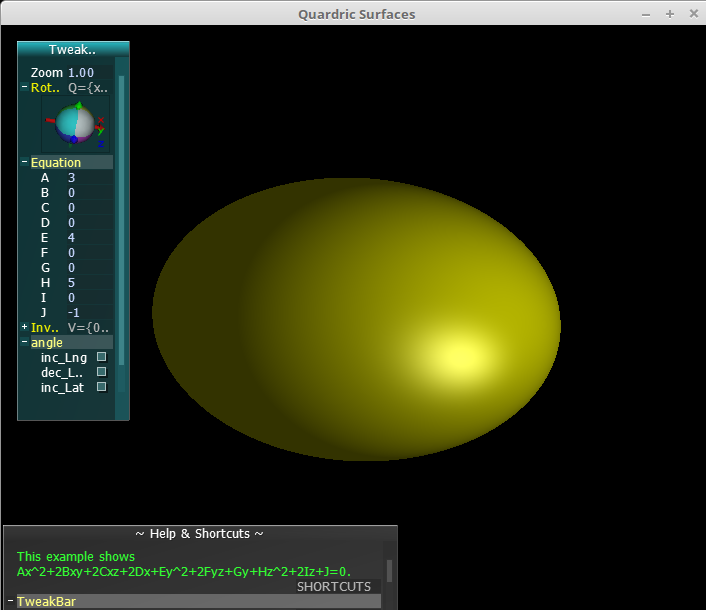
上述方法绘制得到的是视线为z轴，即正视xy平面时的视图。要有交互效果，我们需要在**X**前乘上一个放缩矩阵**Z**和旋转矩阵**R**，和**。**这样求解的方程就变成了。

另外，因为二次曲面中的抛物面、圆锥面等形状是无限延伸的，为了观察的效果比较好，我们需要对进行裁剪，让它的xyz分量都小于1。

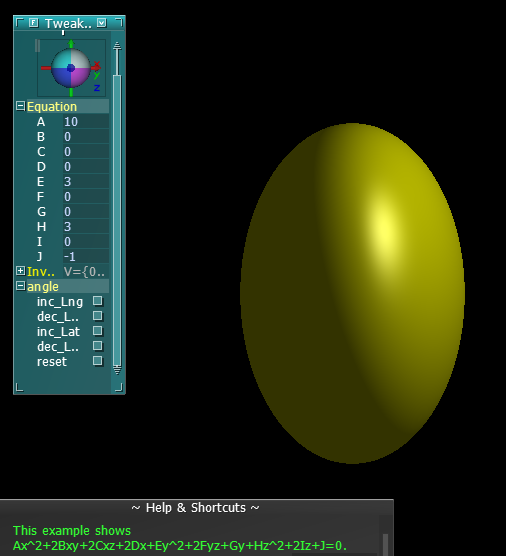
## 效果演示

下面演示课件上展示的各种实二次曲面：

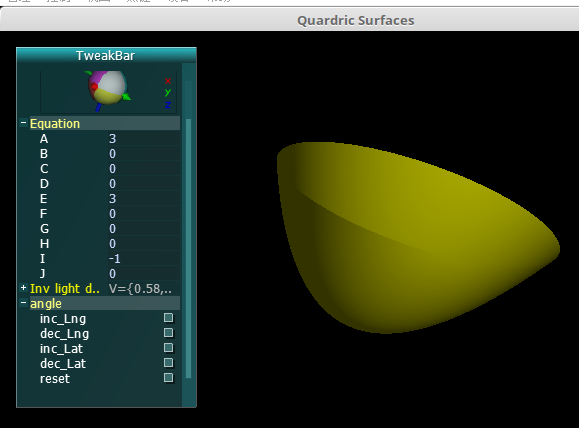
1. Real ellipsoid



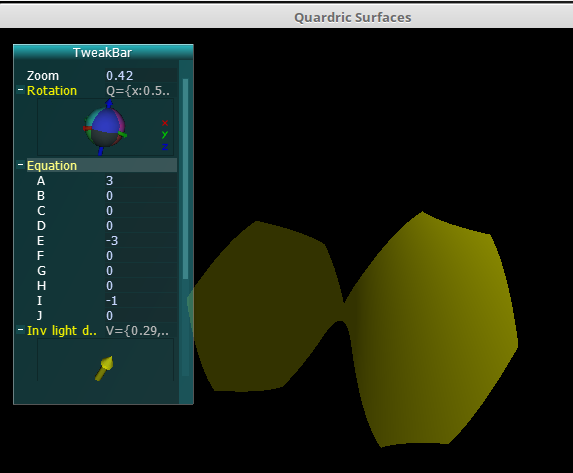
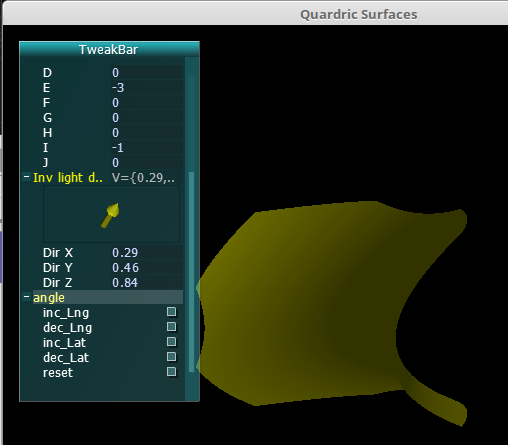
2. Spheroid



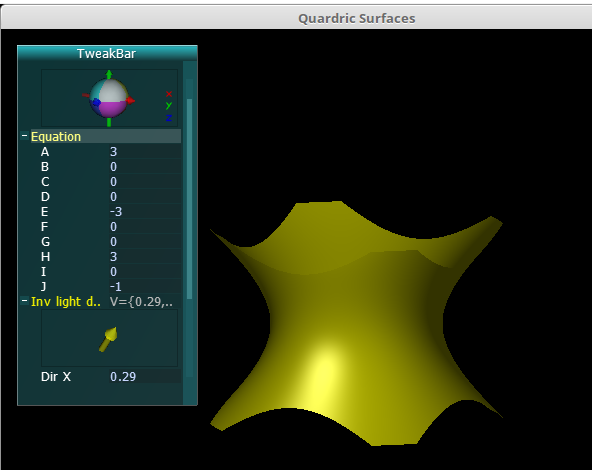
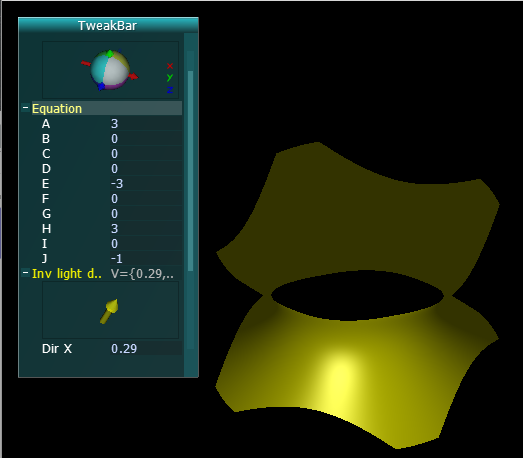
3. Elliptic paraboloid



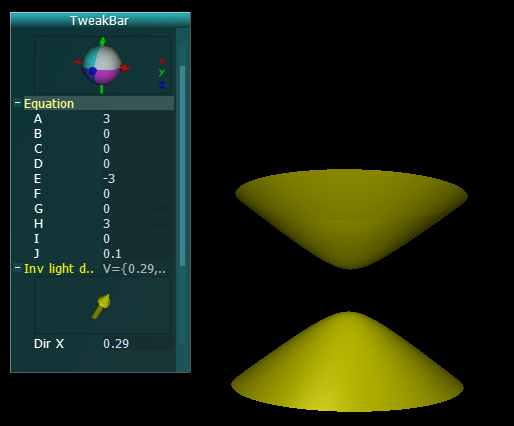
4. Hyperbolic paraboloid（这个曲面的光照不太好）

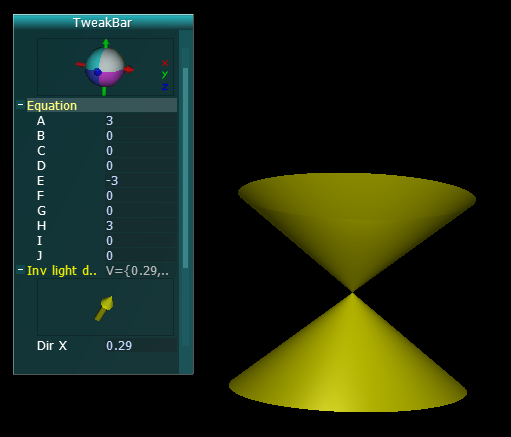
5. Hyperboloid of one sheet

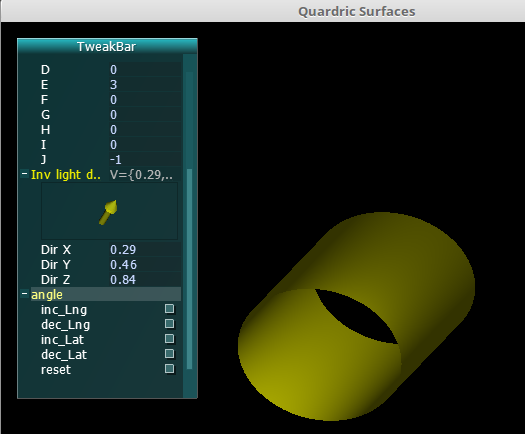
6. Hyperboloid of two sheets



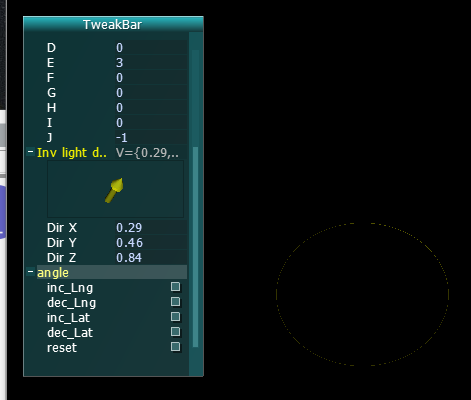
7. Elliptic cone



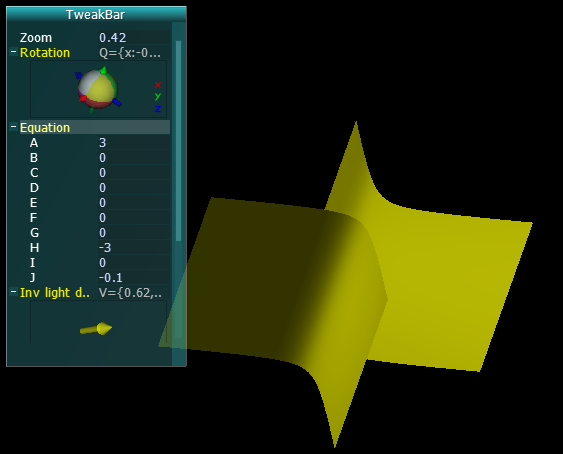
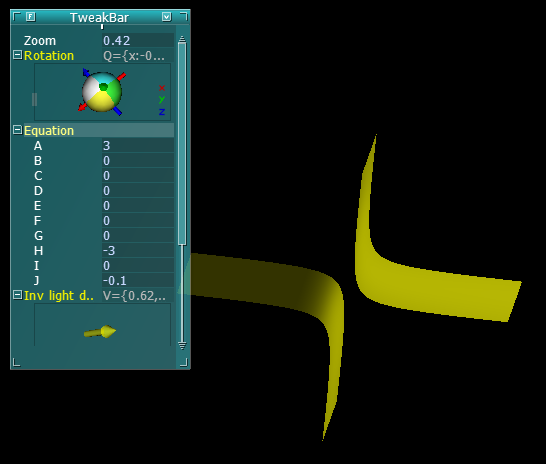
8. Elliptic cylinder



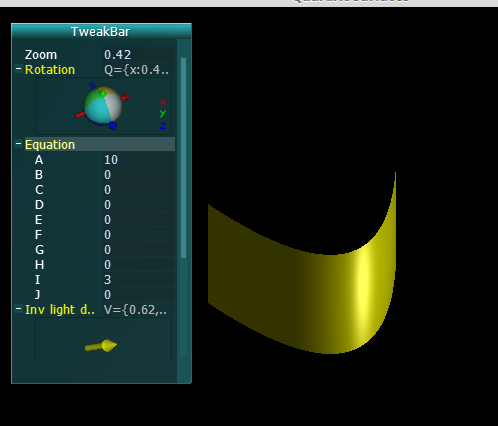
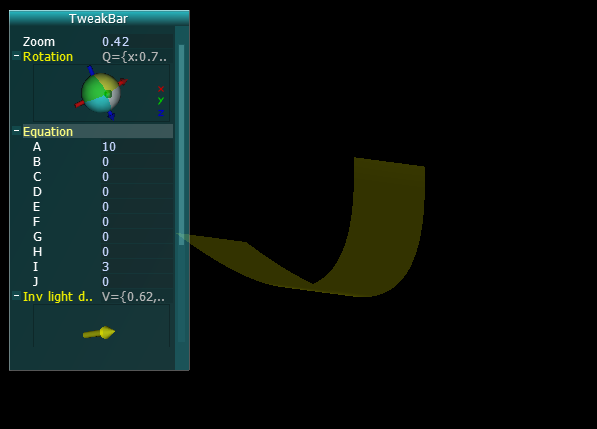
二次方程的二次系数等于0的情况，呈现为虚线，存在离散采样精度的问题



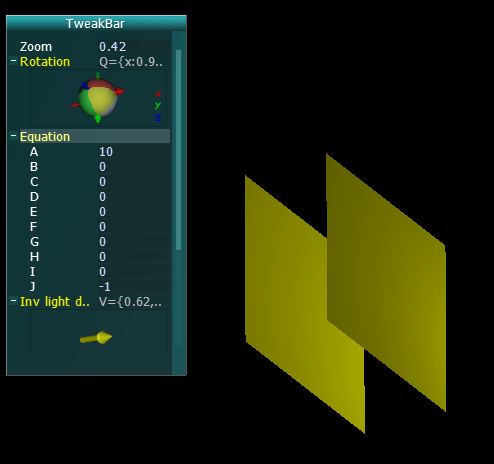
9. Hyperbolic cylinder

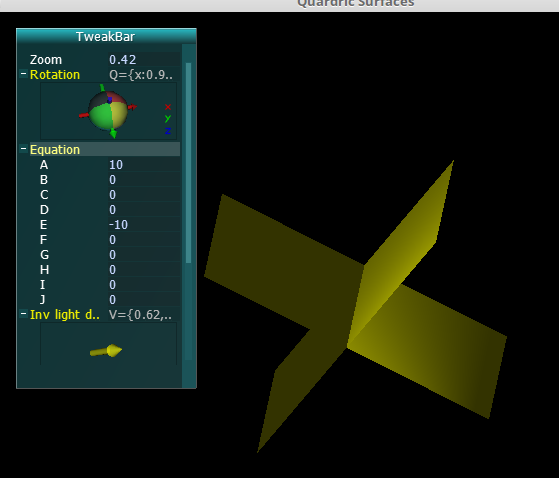
10. Parabolic cylinder

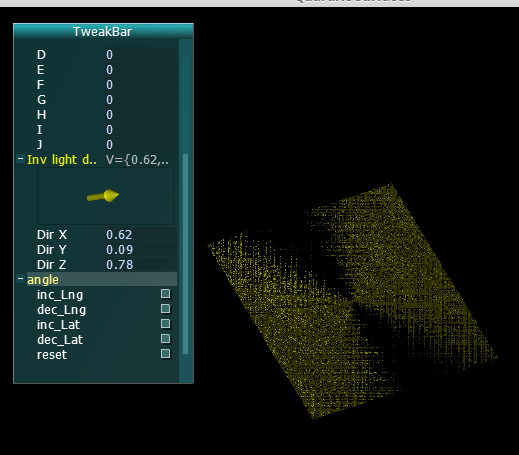
11. Parallel planes



12. cross planes



13. Coincident planes, 同样存在着数值不稳定的问题。



## 存在的问题与改进

正如上面展示的那样，我们的程序还存在三个问题.一是对于特殊情况，数值不稳定。二是光照模型还有不太完善，计算出来的亮度不准确，视觉效果不够好。三是裁剪效果不好。需要未来进一步改进。

## 参考文献

1. Sigg, Christian, Tim Weyrich, Mario Botsch, and Markus H. Gross. "GPU-based ray-casting of quadratic surfaces." In *SPBG*, pp. 59-65. 2006.