

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Компьютерная графика»

Студент: П. А. Мохляков
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19
Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №3

Основы построения фотореалистичных изображений.

Задача: Используя результаты Л.Р.№2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света. Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант 14: Усеченный прямой эллиптический цилиндр.

1 Исходный код

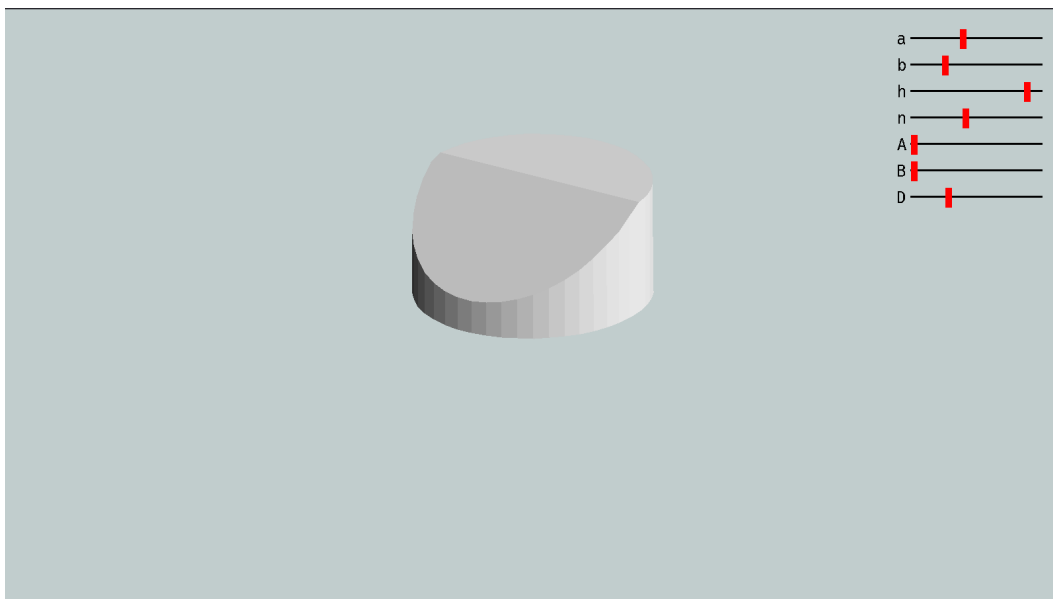
```
1 def Gen_py(a,b,h,n,A,B,D):
2     points = list()
3     t = 0
4     m = 0
5     top = list()
6     cut = list()
7     for i in range(n):
8         x = a*b*np.cos(t)/(np.sqrt(b**2*np.cos(t)**2 + a**2*np.sin(t)**2))
9         y = a*b*np.sin(t)/(np.sqrt(b**2*np.cos(t)**2 + a**2*np.sin(t)**2))
10        points.append([x,y,0])
11        if(A*x + B*y > m):
12            m = A*x + B*y
13            t += 2*np.pi / n
14    D += m
15    for i in range(n):
16        x = a*b*np.cos(t)/(np.sqrt(b**2*np.cos(t)**2 + a**2*np.sin(t)**2))
17        y = a*b*np.sin(t)/(np.sqrt(b**2*np.cos(t)**2 + a**2*np.sin(t)**2))
18        if(h <= D - A*x - B*y):
19            points.append([x,y,h])
20            top.append(i + n)
21        else:
22            points.append([x,y,D - A*x - B*y])
23            cut.append(i + n)
24        t += 2*np.pi / n
25
26
27    faces = list()
28    faces.append(range(n)[::-1])
29    for i in range(n):
30        faces.append([i,(i+1)%n,(i+1)%n + n,i + n])
```

```

31     if(len(top) >= 3):
32         faces.append(top)
33     if(len(cut) >= 3):
34         for i in range(len(cut)):
35             j = (i + 1) % len(cut)
36             if(abs(cut[j] - cut[i]) > 1 and abs(cut[j] - cut[i]) < n):
37                 a = (cut[i] + 1) % n + n
38                 b = (cut[j] - 1) % n + n
39                 cut.insert(i+1,a)
40                 cut.insert(i+2,b)
41                 cut = cut[-5:]+ cut[:-5]
42         faces.append(cut)
43     return objct.Object(points,faces)

```

2 Скриншоты программы



3 Выводы

Выполнив данную лабораторную работу я изучил навыки по аппроксимации такого рода фигур и их отсечений. Так же я узнал простой алгоритм освещения, основанный на угле между нормалью полигона и вектором падения света.