Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Компьютерная графика»

Студент: Ф.М. Шавандрин
Преподаватель: Г.С. Филиппов
Группа: М8О-308Б-19
Дата: 21.12.2021
Оценка:
Подпись:

Москва, 2021

Курсовая работа

Каркасная визуализация порции поверхности.

Задача: Составить и отладить программу, обеспечивающую каркасную визуализацию порции поверхности заданного типа. Исходные данные готовятся самостоятельно и вводятся из файла или в панели ввода данных. Должна быть обеспечена возможность тестирования программы на различных наборах исходных данных. Программа должна обеспечивать выполнение аффинных преобразований для заданной порции поверхности, а также возможность управлять количеством изображаемых параметрических линий. Для визуализации параметрических линий поверхности разрешается использовать только функции отрисовки отрезков в экранных координатах.
Вариант 7: Линейчатая поверхность Кунса (границы – Cardinal Spline 3D).

Описание

Линейная поверхность Кунса - это тип поверхности или параметризации многообразия, используемый в компьютерной графике для плавного соединения других поверхностей вместе, а также в приложениях вычислительной механики, особенно в методе конечных элементов и методе граничных элементов, для объединения проблемных областей в элементы. Уравнение данной поверхности имеет вид:

$$r(u,v)=r^{(1)}(u,v)+r^{(2)}-r^{(3)}(u,v) \quad \text{, где}$$

$$r^{(1)}(u,v)=(1-u)r_3(v)+ur_4(v);$$

$$r^{(2)}(u,v)=(1-v)r_1(u)+vr_2(u);$$

$$r^{(3)}(u,v)=(1-u)(1-v)r(0,0)+u(1-v)r(1,0)+(1-u)vr(0,1)+uvr(1,1) \quad \text{, где}$$

r_1, r_2, r_3, r_4 - граничные кривые (в моём случае — кривые Cardinal Spline 3D). Данные кривые задаются через четыре опорные точки следующим уравнением:

$$P(t)=\frac{(-t(1-t)^2P_0+(2-5t^2+3t^3)P_1+t(1+4t-3t^2)P_2-t^2(1-t)P_3)}{2}$$

Исходный код

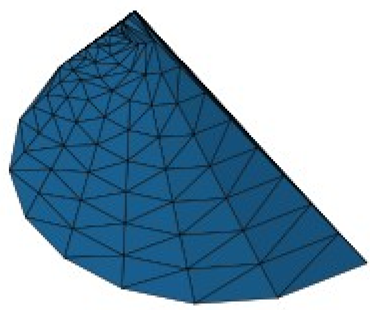
```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits import mplot3d
from mpl_toolkits.mplot3d.art3d import Poly3DCollection


def interpolate(P0, P1, P2, P3, steps):
    spl = []
    for i in range(steps):
        t = i / steps
```

```
        k0 = -1 * t * (1 - t) ** 2 / 2
        k1 = 2 - 5 * t ** 2 + 3 * t ** 3 / 2
        k2 = t * (1 + 4 * t - 3 * t ** 3) / 2
        k3 = -1 *(1 - t) * t ** 2 / 2
        spl.append(k0 * P0 + k1 * P1 + k2 * P2 + k3 * P3)
    return spl

P01 = np.array([10, 10, -10])
P11 = np.array([-40, 20, 30])
P21 = np.array([40, 20, -30])
P31 = np.array([-100, -60, -70])

P02 = np.array([-100, -60, -70])
P12 = np.array([70, 40, 30])
P22 = np.array([80, 100, -100])
P32 = np.array([30, 60, 70])

P03 = np.array([30, 60, 70])
P13 = np.array([-70, -50, -50])
P23 = np.array([40, 10, 20])
P33 = np.array([100, 70, -80])

P04 = np.array([100, 70, -80])
P14 = np.array([10, 10, 10])
P24 = np.array([-20, -40, -70])
P34 = np.array([10, 10, -10])

print("Введите параметр аппроксимации: ")
approximation = max(4, int(input()))

curve1 = interpolate(P01, P11, P21, P31, approximation)
curve2 = interpolate(P02, P12, P22, P32, approximation)
curve3 = interpolate(P03, P13, P23, P33, approximation)
curve4 = interpolate(P04, P14, P24, P34, approximation)

u = np.linspace(0, 1, approximation)
v = np.linspace(0, 100, approximation)

x_array = []
y_array = []
z_array = []

points = []

for i in range(approximation):
    for j in range(approximation):
        point1 = (1 - u[i]) * curve3[j] + u[i] * curve4[j]
        point2 = (1 - v[j]) * curve1[i] + v[j] * curve2[i]
          point3 = (1 - u[i]) * (1 - v[j]) * curve1[0] + u[i] * (1 - v[j]) *
curve4[0]  + (1 - u[i]) * v[j] * curve2[0] + u[i] * v[j] * curve4[-1]
        points.append(point1 + point2 - point3)

points = np.array(points)
figure = plt.figure("Курсовой проект - Шавандрин Фёдор")
axes = figure.add_subplot(111, projection='3d')
axes.grid(True)
axes.plot_trisurf(points[:,  0],  points[:,  1],  points[:,  2],  linewidth=0.3,
edgecolors='k')
plt.xlabel('x')
plt.ylabel('y')
plt.axis('off')
axes.set_xlim([-7000, 7000])
axes.set_ylim([-7000, 7000])
axes.set_zlim([-7000, 7000])
```

```
plt.show()
```

Результат работы

```
Введите параметр аппроксимации:
10
```



Выводы

Выполнив курсовую работу по курсу «Компьютерная графика», я познакомился с кривой Cardinal Spline 3D и линейной поверхностью Кунса. Для закрепления знания я построил данную поверхность, граничные кривые которой — Cardinal Spline 3D. Реализовал ввод пользователем параметра аппроксимации для построения заданной поверхности.