**Лабораторная работа 5.**

**«Кривые и поверхности»**

1. Задание

Карта высот задана лоскутами некоторого вида и данными о накладываемых текстурах.

Разработать формат хранения такой карты. Написать программу, загружаюшую сцену из файла, выводящуюю её и позволяющую редактировать. Перемешивать при выводе текстуры соседних лоскутов. Лоскуты должны стыковаться "без швов".

Вариант лоскута:

Линейчатая поверхность, на основе двух кривых.

Вариант кривой:

B-spline

1. Теория

Линейчатые поверхности образуются путем перемещения прямой линии по определенной траектории. По определению, поверхность называется линейчатой, если через каждую ее точку проходит хотя бы одна прямая, целиком лежащая на этой поверхности.

Параметрическая форма линейчатых поверхностей имеет вид:



Функции Р1 и Р2 определяют кривые, расположенные в трехмерном пространстве.

Линейчатый лоскут формируется посредством ограничения значений u и v.

Линейчатая поверхность задается двумя кривыми P1 и P2. Вместо этих кривых возьмем В-сплайн кривую. Тогда поверхность будет вести себя следующим образом:

u- контуры являются прямыми, соединяющие точки 2х В-сплайнов.

v-контуры -- это сами В-сплайн кривые, построенные на контрольных точках. Контрольные точки взяты из файла, в котором хранится информация о 2х кривых.

Уравнение В-сплайн кривых в общем случае может быть описано следующей формулой:



где:



-- стыковочная функция

Частными случаями В-сплайнов являются:

1) линейные

2) квадратичные

3) кубические

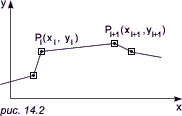
1. Алгоритм

B-сплайн -- это гладкая кривая или, точнее, кривая с непрерывными старшими производными до n-ой, где n-ый порядок сплайна. Линия, составленная из В-сплайнов, не будет проходить точно через заданные точки.

Подобную кривую составляют из дуг полиномов третьей степени, так как такой полином обеспечивает необходимую непрерывность. Построение линии происходит с помощью итерационной процедуры.

Будем рассматривать построение кубического сплайна:

Пусть нам даны две соседние точки, через которые проведем кубический полином. У полинома 4 коэффициента, следовательно нужно еще два дополнительных условия или точки. Для этого возьмем еще две соседние точки. Чем более плавной мы хотим видеть линию, тем сложнее пройти точно через точки.



Пусть t -- параметр, по которому пробегаем от точки Pi к точке Pi+1. При t = 0 мы находимся в точке Pi, при t = 1 -- в точке Pi+1. Если 0 < t < 1, то мы находимся между Pi и Pi+1.

Эта линия в каждой точке имеет систему:

x(t) = ((a3t + a2)t + a1)t + a0, для 0 <= t <= 1

y(t) = ((b3t + b2)t + b1)t + b0, для 0 <= t <= 1

a3 = (-xi-1 + 3xi - 3xi+1 + xi+2)/6

a2 = (xi-1 - 2xi + xi+1)/2

a1 = (-xi-1 + xi+1)/2

a0 = (xi-1 + 4xi+ xi+1)/6

Точки b3 - b0 расписываются так же, но вместо x подставляют у. Между Pi и Pi+1 точки а и b не меняются.

В итоге получаем функцию:

def Bspline(coords):

N = 30

bspline = []

for i in range(0,len(coords)-3,1):

xA, yA = coords[i][0], coords[i][1]

xB, yB = coords[i+1][0], coords[i+1][1]

xC, yC = coords[i+2][0], coords[i+2][1]

xD, yD = coords[i+3][0], coords[i+3][1]

a3 = (-xA + 3 \* (xB - xC) + xD) / 6.0

a2 = (xA - 2 \* xB + xC) / 2.0

a1 = (xC - xA) / 2.0

a0 = (xA + 4 \* xB + xC) / 6.0

b3 = (-yA + 3 \* (yB - yC) + yD) / 6.0

b2 = (yA - 2 \* yB + yC) / 2.0

b1 = (yC - yA) / 2.0

b0 = (yA + 4 \* yB + yC) / 6.0

for i in range(N):

t = i\*1.0/N

X = (((a3 \* t + a2) \* t + a1) \* t + a0)

Y = (((b3 \* t + b2) \* t + b1) \* t + b0)

Z = sin(X)

bspline.append((X,Y,Z))

return bspline

Функция принимает контрольные точки и возвращает точки, для построения В-сплайна.

Линейчатая поверхность изменяется в соответствии с траекторией синуса.

Первый В-сплайн строиться по контрольным точкам, координаты которого формируются с помощью формул:

*10\*sin(t\*5) и 10\*cos(t\*10)*

Второй В-сплайн строится по аналогичной формуле, но со смещением на константу:

*10\*sin(t\*5)+10.0, 10\*cos(t\*10)+5.0*

def AutogenFile(filename):

"""Генерация опорных точек в файл"""

inits = [(10\*sin(t\*5),10\*cos(t\*10)) for t in xrange(0,200)]

f = open(filename,'w+')

#~ write first coords

for x,y in inits:

f.write(repr(x))

f.write(' ')

f.write(repr(y))

f.write('\n')

f.write('---\n')

#~ write second coords

for x,y in inits:

f.write(repr(x+10.0))

f.write(' ')

f.write(repr(y+5.0))

f.write('\n')

1. Выводы

В этой лабораторной работе научились строить карту высот, разобрались с разнообразными видами кривых и поверхностей.

Линейчатая поверхность содержит большое разнообразие полезных и интересных форм. К самым известным семействам относят конус, цилиндр, билинейный лоскут. При помощи линейчатых поверхностей и их модификаций, можно создавать такие формы, как ленты Мебиуса, двойные спирали, купола и т.д.

В качестве кривых, ограничивающих поверхность, были выбраны В-сплайн кривые.

Преимущество В-сплайна перед кривой Безье заключается в том, что теперь мы обеспечили локальный контроль над формой кривой. При изменении одной из опорных точек, не нужно будет перерисовывать всю кривую -- изменения коснутся лишь ближайших соседних точек.

Квадратичный В-слайн является одним из наиболее используемых. При увеличении порядка В-сплайна, уменьшается локальный контроль над кривой и уменьшает ее точность. Однако, чем больше n, тем более гладкой становится кривая. Кубический сплайн позволяет достигнуть компромисса в выборе между гладкостью и точностью.

Гладкость диктуется физическими задачами, и здесь часто приходится искать компромисс между гладкостью и точностью. Например, гидродинамика работает с поверхностями, которые описываются уравнениями четвертой степени. Такой высокий порядок необходим, чтобы повысить гладкость различного рода физических устройств, рассчитанных с помощью этих уравнений, и таким образом избежать завихрений.

Один из существенных недостатков В-сплайна -- проблема при аппроксимации прямой, имеющей разрывы вторых производных.

1. Литература
2. Ф.Хилл "OpenGL. Программирование компьютерной графики"
3. http://www.codenet.ru

**Screen:**

