УДК 514.75.

# СПЛАЙНОВЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ РЕЛЬЕФА

***В. Б. Ким, Е. В. Прокопенко***

*В работе предложен новый подход к обработке геоинформационных данных. Он основан на использовании B-сплайновых поверхностей для моделирования рельефа. Описана структура программного комплекса, реализующего предложенный алгоритм.*

*In this article a new approach to processing of geoinformatic data is suggested. It is based on using of B-spline surfaces for relief modeling. A program of realization of this algorithm is described.*

***Ключевые слова:*** геоинформационные данные, B-сплайновая поверхность, математическая модель.

В настоящее время развитие геометрических исследований стимулируется не только внутренними проблемами и задачами геометрии и математики в целом, но и развитием информатики и информационных технологий. Компьютерная геометрия и графика, вычислительная геометрия, геоинформатика – вот некоторые разделы информатики, которые, с одной стороны, широко используют результаты геометрических исследований, а с другой – подталкивают геометров к решению, казалось бы, абстрактных задач, находящих тем не менее неожиданные практические применения.

Так, например, одной из актуальных задач в геоинформатике является построение математической модели рельефа по данным космической или аэрофотосъемки, или геодезических измерений.

Удачным подходом к решению этой задачи можно считать сплайновый подход, при котором математическая модель рельефа рассматривается как составная поверхность. Одним из главных его достоинств является то, что сплайновые поверхности и кривые однозначно определяются массивом точек (опорным массивом), причем расположение самих точек в пространстве и их нумерация, как правило, не связаны. Поэтому с геометрической точки зрения, исследуемые массивы данных можно во многих случаях трактовать как опорные массивы составных кривых или поверхностей.

Для решения указанной задачи нужно расширить класс допустимых поверхностей, включив в него и поверхности, которые нельзя однозначно спроектировать ни на одну из координатных плоскостей, а также поверхности с самопересечениями. Такие поверхности удобно описывать с помощью параметрических уравнений вида:

$$\{x(u,v),\ y(u,v),\ z(u,v)\},\quad (u,v)\in[\alpha,\beta]\times[\gamma,\delta]. \tag{1}$$

При этом на функции $x(u,v),\ y(u,v),\ z(u,v)$ накладываются дополнительные условия достаточной гладкости и граничные условия.

Укажем некоторые вполне естественные требования, которые нужно наложить на внедряемые классы поверхностей и выполнение которых необходимо для успешного решения задачи приближения или сглаживания (разумеется, есть и другие):

- выбираемый класс должен описываться достаточно просто;

- поверхности, влияющие на выделенный класс, не должны иметь особенностей, то есть должны быть достаточно гладкими – нигде не рваться, иметь непрерывно меняющуюся касательную или непрерывную кривизну;

- поиск нужной поверхности в выделенном классе должен быть сравнительно легким, что предполагает наличие эффективного алгоритма её построения;

- для достаточно больших массивов точек найденные поверхности должны вести себя вполне предсказуемо.

На практике многие поверхности имеют довольно сложную форму, не допускающую универсального аналитического задания в целом при помощи элементарных функций. Поэтому их собирают из сравнительно простых фрагментов, каждый из которых может быть вполне удовлетворительно представлен в виде элементарной функции или двух переменных.

Общую задачу можно сформулировать так: по заданному массиву вершин построить гладкую поверхность, которая, плавно изменяясь, проходила бы вблизи этих вершин. Тем самым решение задачи создания составной поверхности разбивается на несколько этапов, описанных Е. В. Шикиным [1]:

1) задание подходящего набора опорных точек;

2) выборка универсального и сравнительного несложного способа построения элементарных фрагментов поверхности;

3) отыскание эффективного механизма плавной состыковки элементарных фрагментов;

4) анализ полученных результатов.

Ясно, что решение такой задачи далеко не однозначно, так как приблизить массив точек плавноизменяющейся или поверхностью можно очень многими способами.

Рассмотрим поэтапное решение задачи построения поверхности, моделирующей рельеф.

На первом этапе нам дан набор точек с координатами *(X,Y,Z)*. Обозначим исходный массив как: $P=\left(x_i y_i z_i \quad i=1,...N\right)$. Заметим, что величины $(x,y,z)$, которые получаются в результате геодезических измерений, содержат случайную погрешность. Поэтому при построении по этим точкам поверхности нецелесообразно использовать интерполяцию, так как интерполирующая функция будет воспроизводить осцилляции, вызванные случайной компонентой в координатном массиве $\{x_i, y_i, z_i\}$. Более разумным нам видится подход, основанный

на процедуре сглаживания, призванной как-то уменьшить элемент случайности в результатах измерений.

Мы будем рассматривать данный массив точек как двумерный граф

$\{P_{ij}, i = 0,1,2,...,m;\ j = 0,1,2,...,n\}$, топологически эквивалентный правильной прямоугольной сетке. В этом массиве вершины:

$$P_{ij}, i = 1,...,m-1;\ j = 1,...,n-1 \qquad (2)$$

называются внутренними, а вершины:

$$\begin{aligned} &P_{0j} && j = 0,1,..,n-1,\\ &P_{mj} && j = 1,2,..,n,\\ &P_{in} && i = 1,2,...m-1,\\ &P_{i0} && i = 1,2,..,m \end{aligned} \qquad (3)$$

– граничными.

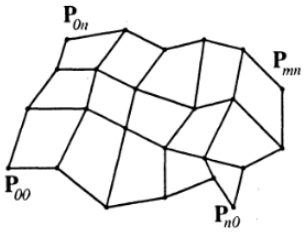


*Рис. 1.* ***Вершины опорного массива***

Для отыскания подходящих параметрических уравнений нужной поверхности удобно воспользоваться параметрическими уравнениями составной бикубической B-сплайновой поверхности [1]:

$$R^{(i,j)}(u,v) = \sum_{k=0}^{3}\sum_{l=0}^{3} n_k(u) n_l(v) P_{k-1+i,l-1+j} \qquad (4)$$

$$0 \le u,v \le 1, \quad 1 \le i \le m-2, \quad 1 \le j \le n-2,$$

или в матричной форме:

$$R^{(i,j)}(u,v) = $$

$$= U^T M^T \begin{pmatrix} P_{i-1,j-1} & P_{i,j-1} & P_{i+1,j-1} & P_{i+2,j-1} \\ P_{i-1,j} & P_{i,j} & P_{i+1,j} & P_{i+2,j} \\ P_{i-1,j+1} & P_{i,j+1} & P_{i+1,j+1} & P_{i+2,j+1} \\ P_{i-1,j+2} & P_{i,j+2} & P_{i+1,j+2} & P_{i+2,j+2} \end{pmatrix} MV, \qquad (5)$$

$$0 \le u,v \le 1, \quad 1 \le i \le m-2, \quad 1 \le j \le n-2,$$

где матрица M и коэффициенты имеют вид:

$$M = \frac{1}{6}\begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{aligned} &n_0(u) = \frac{1}{6}(1-u)^3, \quad n_1(u) = \frac{1}{6}(3u^3 - 6u^2 + 4),\\ &n_2(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1),\\ &n_3(u) = \frac{u^3}{6}, \quad n_0(v) = \frac{1}{6}(1-v)^3,\\ &n_1(v) = \frac{1}{6}(3v^3 - 6v^2 + 4),\\ &n_2(v) = \frac{1}{6}(-3v^3 + 3v^2 + 3v + 1), \quad n_3(u) = \frac{v^3}{6}. \end{aligned} \qquad (6)$$

Выбор B-сплайновой поверхности обусловлен несколькими причинами.

Во-первых, основное преимущество поверхности, построенной с помощью B-сплайнов, заключается в том, что изменение одной из вершин влечет за собой изменение только 16 фрагментов поверхности, в определении которых участвует эта точка. Поэтому местная корректировка может вестись без перестройки всей поверхности. Указанное обстоятельство оказывается существенным для применения метода «кратных» точек.

Во-вторых, элементарные фрагменты B-сплайновой поверхности соединяются между собой по B-сплайновым кривым, вдоль которых касательный вектор и вектор кривизны меняются непрерывно. Тем самым решается и проблема третьего этапа построения.

В третьих, поскольку B-сплайновые поверхности можно рассматривать как «сшитые» из B-сплайновых кривых, то при нахождении точек выброса можно использовать метод канонических моделей, предложенный в [2].

Одна из проблем построения B-сплайновой поверхности заключается в том, что построенная по данному массиву поверхность не проектируется на плоский массив (X, Y). Чтобы исправить данную ситуацию, используем метод кратных точек, суть которого будет описана ниже, а именно – добавим так называемые «кратные точки» на границах поверхности по следующему алгоритму:

Для массива $P = \{P_{ij}, i = 0,..,m,\ j = 0,...,n\}$ задаем $2m + 2n + 8$ новых вершин, положив:

$$
\begin{aligned}
&P_{-1,j} = P_{0,j}, \qquad j = 0,1,..,n,\\
&P_{m+1,j} = P_{m,j}, \qquad j = 0,1,..,n,\\
&P_{i,-1} = P_{i,0}, \qquad i = 0,1,..,m,\\
&P_{i,n+1} = P_{i,n}, \qquad i = 0,1,..,m,\\
&P_{-1,-1} = P_{0,0}, \qquad P_{-1,n+1} = P_{0,n},\\
&P_{m+1,-1} = P_{m,0}, \quad P_{m+1,n+1} = P_{m,n}.
\end{aligned} \tag{7}
$$

Эти вершины как бы «окаймляют» заданный массив $P$ и вместе с ним образуют новый массив $P^* = \{P_{i,j},\ i = -1,...,m+1,\ j = -1,..,n+1\}$ из $(m+3)(n+3)$ вершин.

Заметим, что как множества точек массивы $P$ и $P^*$ неразличимы, так как все добавляемые вершины отличаются от заданных только номерами – геометрически новых вершин не появилось

Это позволяет рассматривать B-сплайновую поверхность как поверхность, построенную над массивом (X,Y).

Данный метод апробирован на геоинформационных данных по Змеиногорскому району Алтайского края, вычислена погрешность, проведена визуализация.

Опишем алгоритм работы с данными, реализованный при помощи языка программирования Visual FoxPro.

Массив состоит из трех групп чисел: первая – это широта, вторая 2 – долгота, третья – высота. Обозначим эти значения соответственно: X, Y, Z.

Далее проводим сортировку следующим образом: в первую очередь, сортируем массив по значению X по возрастанию, соответственно сортируются и значения Y и Z. Далее сортируем значения Y для одинаковых блоков значений X, значения Z оставляем в том порядке как они получились при сортировке по значениям X и Y.

Завершающий этап – это переформировка массива, а именно: разбиение его из одного текстового документа «Sort.txt» в три разных документа с набором значений X, Y, Z соответственно в каждом: «SortX.txt», «SortY.txt», «SortZ.txt».

Каждый текстовый документ содержит наборы числовых значений, разделенных табулятором в столбцах и по строкам. Определяем равные значения X и помещаем их в отдельные столбцы, соответственно формируется массив Y и Z.

На выходе получаем 3 текстовых документа, содержащие координаты точек. Добавляем в них кратные точки, и уже с этими документами у нас в дальнейшем будет идти работа.

Опишем процедуру работы с массивом, произведенной в Visual FoxPro:

```
CREATE CURSOR [CursorDest] (X N(20), Y N(20), Z N(20)) && Результирующий курсор отсортированных координат (X,Y,Z) без порядковых номеров
CREATE CURSOR [CursorSource] (ID N(20), X N(20), Y N(20), Z N(20)) && Курсор с начальными координатами, включающий столбец с порядковым номером
SELECT [CursorSource]
APPEND FROM C:\Draw\Rasn.txt TYPE DELIMITED WITH CHARACTER « « && Загрузка начальных координат во временный курсор
&& Сортировка по возрастанию всех координат X
SELECT DISTINCT a.x FROM [CursorSource] a ORDER BY a.x INTO CURSOR Cursor_x
SELECT Cursor_x
m.TypeSort = 1 && У первой координаты X, координаты Y по возрастанию
SCAN ALL && Сканируем курсор с отсортированными координатами X сверху вниз
SCATTER FIELDS x MEMVAR && Получаем координату X
IF m.TypeSort = 1 && Если 1, то сортируем координаты Y по возрастанию
&& Добавляем список координат (Y и Z), принадлежащих координате X и отсортированных по возрастанию Y к результирующему курсору
INSERT INTO [CursorDest] (X,Y,Z) SELECT a.X,a.Y,a.Z FROM [CursorSource] a WHERE a.X = m.X order by a.Y Asc
m.TypeSort = 2 && Для следующей координаты X координаты Y по убыванию
ELSE && Иначе по убыванию
&& Добавляем список координат (Y и Z), принадлежащих координате X, отсортированных по убыванию Y к результирующему курсору
INSERT INTO [CursorDest] (X,Y,Z) SELECT a.X,a.Y,a.Z FROM [CursorSource] a WHERE a.X = m.X order by a.Y Desc
m.TypeSort = 1 && Для следующей координаты X координаты Y по возрастанию
ENDIF
ENDSCAN && Конец цикла
&& Далее разбиваем текстовый документ на 3, в каждом из которых будут находиться координаты X, Y, Z, соответственно:
SELECT [CursorDest]
COPY TO C:\Draw\SortNew.txt TYPE DELIMITED WITH CHARACTER « « && Сохранение результирующего курсора в текстовый файл с разделителями
CREATE CURSOR Curs (X N(20), Y N(20), Z N(20))
SELECT curs
APPEND FROM c:\draw\SortNew.txt TYPE DELIMITED WITH CHARACTER " "
SELECT a.x, count(*) as KolvoX FROM curs a GROUP BY a.x INTO CURSOR ListX ORDER BY a.x
SELECT MIN(a.KolvoX) as MinX FROM ListX a INTO CURSOR ListMinX
SELECT ListMinX
GO TOP IN ListMinX
SCATTER FIELDS MinX MEMVAR
m.StringX = ""
m.StringY = ""
m.StringZ = ""
```

```
DIMENSION ArrayX(32,32), ArrayY(32,32), ArrayZ(32,32)
STORE "" TO ArrayX, ArrayY, ArrayZ
SELECT ListX
GO TOP IN ListX
SCAN NEXT 32
SCATTER FIELDS X MEMVAR
FOR I=1 TO 32
ArrayX(I, RECNO("ListX")) = ALLTRIM(STR(m.X))
ENDFOR
SELECT a.Y FROM Curs a WHERE a.X = m.X INTO CURSOR ListY ORDER BY a.Y
SELECT ListY
SCAN NEXT 32
SCATTER FIELDS Y MEMVAR
ArrayY(RECNO("ListY"), RECNO("ListX")) = ALLTRIM(STR(m.Y))
ENDSCAN
SELECT a.Z FROM Curs a WHERE a.X = m.X INTO CURSOR ListZ
SELECT ListZ
SCAN NEXT 32
SCATTER FIELDS Z MEMVAR
ArrayZ(RECNO("ListZ"), RECNO("ListX")) = ALLTRIM(STR(m.Z))
ENDSCAN
SELECT ListX
ENDSCAN
FOR K=1 TO 32
FOR L=1 TO 32
m.StringX = m.StringX + ArrayX(K,L) + SPACE(1)
    m.StringY = m.StringY + ArrayY(K,L) + SPACE(1)
m.StringZ = m.StringZ + ArrayZ(K,L) + SPACE(1)
ENDFOR
m.StringX = LEFT(m.StringX, LEN(m.StringX)-1) + CHR(13) + CHR(10)
m.StringY = LEFT(m.StringY, LEN(m.StringY)-1) + CHR(13) + CHR(10)
m.StringZ = LEFT(m.StringZ, LEN(m.StringZ)-1) + CHR(13) + CHR(10)
ENDFOR
STRTOFILE(m.StringX, "c:\draw\SortX32_Column.txt", 0)
STRTOFILE(m.StringY, "c:\draw\SortY32_Column.txt", 0)
STRTOFILE(m.StringZ, "c:\draw\SortZ32_Column.txt", 0)
```

После формирования документов с данными, в удобной для нас форме, строим саму поверхность при помощи системы Maple V.

Опишем процедуру (написанную в среде Maple V) построения рельефа местности и определения погрешностей (абсолютной и относительной) построения. Вывод необходимых нам уравнений или значений производит при необходимости команда *print()*. Так как для построения поверхности используем параметрический вид поверхности, то данную процедуру разбиваем на несколько.

Подключаем необходимые пакеты для произведения вычислений и построения поверхности:

- with(linalg): with(plots):

Производим считывание данных командой readdata(`SortY.txt`, float, 1000) из текстовых документов «SortX.txt», «SortY.txt», «SortZ.txt», подготовленных нами ранее и расположенных в той же папке, что и исходный рабочий файл. Перед обращением к данной команде ее необходимо подключить командой readlib(readdata). Далее формируем матрицы с числами (координатами поверхности) с плавающей точкой; ставим максимальное число считываемых элементов, для того чтобы можно было работать с большими массивами данных.

- readlib(readdata):
- dataX:=matrix(readdata(`SortX.txt`,float,1000)):
- dataY:=matrix(readdata(`SortY.txt`,float,1000)):
- dataZ:=matrix(readdata(`SortZ.txt`,float,1000)):

Определяем размерность массивов данных для возможности дальнейшего правильного образования циклов:

- Mx:=rowdim(dataX): Nx:=coldim(dataX):
- My:=rowdim(dataY): Ny:=coldim(dataY):
- Mz:=rowdim(dataZ): Nz:=coldim(dataZ):

Задаем функциональные коэффициенты, определяющие B-сплайновую поверхность:

- n0(u):=(1-u)^3/6:
  n1(u):=(3*u^3-6*u^2+4)/6:
- n2(u):=(-3*u^3+3*u^2+3*u+1)/6:
  n3(u):=u^3/6:
- n0(v):=(1-v)^3/6: n1(v):=(3*v^3-6*v^2+4)/6:

n2(v):=(-3*v^3+3*v^2+3*v+1)/6:
n3(v):=v^3/6:

Определяем параметрические уравнения функций X(u,v) с помощью цикла:

- for ix from 1 to Mx-3 do
- for jx from 1 to Nx-3 do
- Rx[ix,jx]:=normal((n0(u)*n0(v)*dataX[ix,jx]+
+n0(u)*n1(v)*dataX[ix,jx+1]+n0(u)*n2(v)*dataX[ix,jx+2]
+n0(u)*n3(v)*dataX[ix,jx+3])+
+(n1(u)*n0(v)*dataX[ix+1,jx]+n1(u)*n1(v)*dataX[ix+
+1,jx+1]+n1(u)*n2(v)*dataX[ix+1,jx+2]+n1(u)*n3(v)*
dataX[ix+1,jx+3])+
+(n2(u)*n0(v)*dataX[ix+2,jx]+n2(u)*n1(v)*dataX[ix+
+2,jx+1]+n2(u)*n2(v)*dataX[ix+2,jx+2]+n2(u)*n3(v)*
dataX[ix+2,jx+3])+
+(n3(u)*n0(v)*dataX[ix+3,jx]+n3(u)*n1(v)*dataX[ix+
+3,jx+1]+n3(u)*n2(v)*dataX[ix+3,jx+2]+n3(u)*n3(v)*
dataX[ix+3,jx+3])):
- od od:

Определяем параметрические уравнения функций Y(u,v), с помощью цикла:

- for iy from 1 to My-3 do
- for jy from 1 to Ny-3 do
- Ry[iy,jy]:=normal((n0(u)*n0(v)*dataY[iy,jy]+
+n0(u)*n1(v)*dataY[iy,jy+1]+n0(u)*n2(v)*dataY[iy,jy+
+2]+n0(u)*n3(v)*dataY[iy,jy+3])+
+(n1(u)*n0(v)*dataY[iy+1,jy]+n1(u)*n1(v)*dataY[iy+
+1,jy+1]+n1(u)*n2(v)*dataY[iy+1,jy+2]+n1(u)*n3(v)*
dataY[iy+1,jy+3])+

+(n2(u)*n0(v)*dataY[iy+2,jy]+n2(u)*n1(v)*dataY[iy+
+2,jy+1]+n2(u)*n2(v)*dataY[iy+2,jy+2]+n2(u)*n3(v)*
dataY[iy+2,jy+3])+
+(n3(u)*n0(v)*dataY[iy+3,jy]+n3(u)*n1(v)*dataY[iy+
+3,jy+1]+n3(u)*n2(v)*dataY[iy+3,jy+2]+n3(u)*n3(v)*
dataY[iy+3,jy+3])):

- od od:

Определяем параметрические уравнения функций Z(u, v) с помощью цикла:

- for iz from 1 to Mz-3 do
- for jz from 1 to Nz-3 do
- Rz[iz,jz]:=normal((n0(u)*n0(v)*dataZ[iz,jz]+
+n0(u)*n1(v)*dataZ[iz,jz+1]+n0(u)*n2(v)*dataZ[iz,jz+
+2]+n0(u)*n3(v)*dataZ[iz,jz+3])+
+(n1(u)*n0(v)*dataZ[iz+1,jz]+n1(u)*n1(v)*dataZ[iz+1,
jz+1]+n1(u)*n2(v)*dataZ[iz+1,jz+2]+n1(u)*n3(v)*data
Z[iz+1,jz+3])+
+(n2(u)*n0(v)*dataZ[iz+2,jz]+n2(u)*n1(v)*dataZ[iz+2,
jz+1]+n2(u)*n2(v)*dataZ[iz+2,jz+2]+n2(u)*n3(v)*data
Z[iz+2,jz+3])+
+(n3(u)*n0(v)*dataZ[iz+3,jz]+n3(u)*n1(v)*dataZ[iz+3,
jz+1]+n3(u)*n2(v)*dataZ[iz+3,jz+2]+n3(u)*n3(v)*data
Z[iz+3,jz+3])):
- od od:

После определения параметрических уравнений частей составной поверхности строим саму составную поверхность:

- Vx:=convert(dataX,vector):Vy:=convert(dataY,
vector):Vz:=convert(dataZ,vector):[Vx[1],Vy[1],Vz[1]]:
- macro(palegreen=COLOR(RGB, 0, 0, 255)):
- J1:=seq(pointplot3d([Vx[i],Vy[i],Vz[i]],symbo
l=diamond,symbolsize=2,axes=BOXED),i=1..324):
- J2:=seq(seq(plot3d([Rx[i,j],Ry[i,j],Rz[i,j]],u=
=0..1,v=0..1),i=1..50),j=1..50):
- display({J1,J2},color=palegreen, labels=[`Ось
X`,` Ось Y`,` Ось Z`]).

Определяем погрешность построения, для этого для начала выбираем точку, в которой будем вычислять, далее находим значения параметров u и v, и находим соответствующее для них значение Z(u,v). Из полученных данных вычисляем абсолютную и относительную погрешности:

- x:=dataX[3,5];y:=dataY[3,5];
- CC:=fsolve({Rx[2,4]=x,Ry[2,4]=y},{u,v},
{u=0..1,v=0..1});assign(CC);
- Rz[2,4]: dataZ[3,5]:
- pogr_abs:=abs(Rz[2,4]-dataZ[3,5]);
- pogr_otn:=abs(Rz[2,4]-
-dataZ[3,5])/abs(dataZ[3,5]).

Результаты анализа построения модели рельефа представлены в таблице 1.

Таблица 1

**Значения погрешностей**

| ***Средняя относительная погрешность*** | ***Средняя абсолютная погрешность, м*** | ***Среднее квадратичное отклонение (взвешенная дисперсия)*** | ***Максимальная относительная погрешность*** | ***Максимальная абсолютная погрешность, м*** |
|---|---|---|---|---|
| 0,008296 | 3,22476 | 11,82 | 0,14 | 58,0456 |

**Литература**

1. Шикин, Е. В. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей [Текст] / Е. В. Шикин, А. И. Плис. – М.: Диалог-МИФИ, 1996. – 240 с.

2. Прокопенко, Е. В. Канонические модели кубически параметризованных кривых / Е. В. Прокопенко // Исследовано в России. – Режим доступа: http://zhurnal.ape.relarn.ru/articles/2008/029.pdf. – С. 329 – 337.

*Рецензент – В. А. Петин, ГОУ ВПО «Кемеровский государственный университет».*