

МИНОБНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ЮГО ЗАПАДНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра вычислительной техники

МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ЛАНДШАФТОВ

Методические указания к лабораторной работе №1
по дисциплине "Устройства человеко-машинного интерфейса"

Курск 2004

Составил Д.Б. Борзов

УДК 681.3:519.68

Работа с трехмерной графикой. Генерация трехмерных ландшафтов: Методические указания к лабораторной работе №1 по дисциплине «Устройства человеко-машинного интерфейса» / Курск. гос. техн. ун-т; Сост.: Д.Б. Борзов. Курск, 2004. __ с.

Изложены методические основы построения программ, обеспечивающих создание эффектов трехмерной графики, одним из которых является эффект трехмерного ландшафта. Представлены, контрольные вопросы и список рекомендуемой литературы. Представленные алгоритмы ориентированы на программирование в интегрированных средах Turbo-Pascal, Borland-Pascal, Object Pascal, Turbo-C, Borland-C++, C++ Builder.

Предназначены для студентов специальности 220100 «ЭВМ, комплексы, системы и сети».

Рецензент: Титенко Е.А.

Библиогр.: 6 назв.

Текст печатается в авторской редакции

ЛР№020280 от ПЛД №50-25 от

Подписано в печать . Формат 60×84 1/16. Печать офсетная.

Усл. печ. л. . Уч.-изд. л. . Тираж 40 экз. Заказ .

Бесплатно.

Курский государственный технический университет.

Подразделение оперативной полиграфии Курского государственного технического университета.

Адрес университета и подразделения оперативной полиграфии:
305040 г. Курск, ул. 50 лет Октября, 94.

1. Цель работы

Целью данной работы является изучение принципов создания эффектов трехмерной графики и приобретение навыков в написании программ, обеспечивающих создание эффекта трехмерных ландшафтов.

2. Основные теоретические положения

В настоящее время существует несколько основных принципов представления данных для хранения информации о ландшафтах:

Первый - использование регулярной сетки высот (или другое название Карта Высот - HeightMap).

Второе - использование иррегулярной сетки вершин и связей, их соединяющих (т.е. хранение простой триангулизированной карты).

Третий - хранение карты ландшафта, но в данном случае хранятся не конкретные высоты, а информация об использованном блоке. В этом случае создается некоторое количество заранее построенных сегментов, а на карте указываются только индексы этих сегментов.

2.1. Построение ландшафтов с помощью регулярной сетки высот

В данном способе данные представлены в виде двумерного массива. Уже заданы две координаты: x – по высоте и y по ширине массива, а третья координата, заданная значением в конкретной ячейке – это высота точки будущего ландшафта. Пример карты высот представлен на рис. 1а, а соответствующий, а соответствующий ландшафт, построенный на ее основе – на рис. 1б.

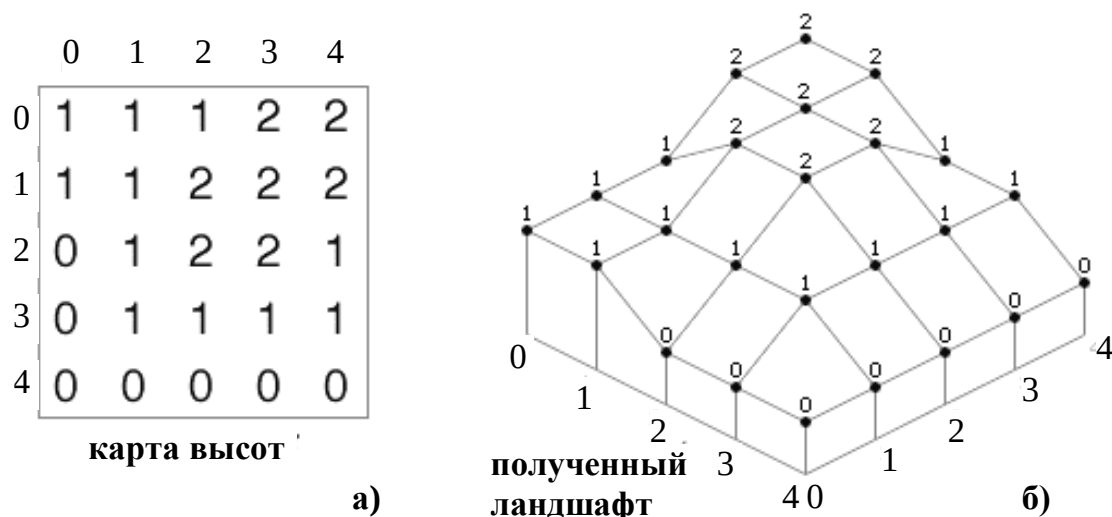


Рис. 1. Пример карты высот и соответствующего трехмерного ландшафта для метода регулярной сетки высот

Обычно карту высот хранят в файлах картинок. Это позволяет легко вносить изменения и более-менее наглядно просматривать данные. Тогда двумя координатами будет положение конкретного пикселя на картинке, а третья координата будет представлена цветом (чем выше значение, прямая зависимость от яркости пикселя - тем больше значение высоты для этой точки). Обычно такие картинки содержатся в монохромном варианте, но можно использовать и все цвета радуги. Второй вариант дает нам больше градаций высоты, чем предполагаемые 256 градаций в случае монохромного представления.

Примерный вид такого ландшафта выглядит следующим образом (Рис. 2):

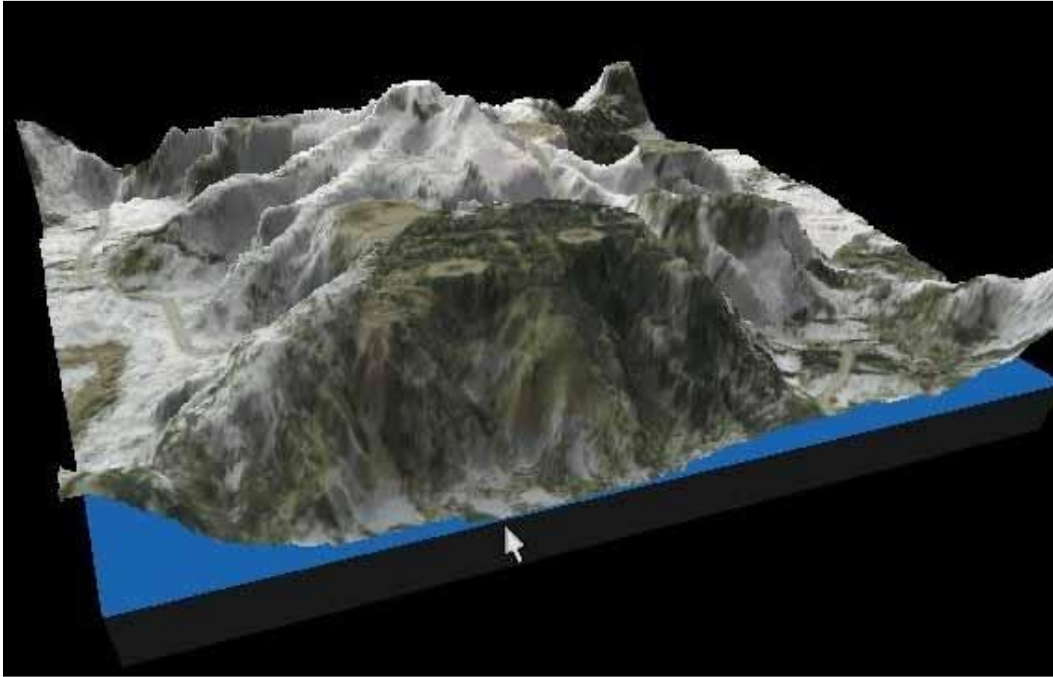


Рис. 2. Пример трехмерного ландшафта

Один из алгоритмов для построения ландшафтов состоит из следующих основных этапов:

Создаем двумерный массив размером $n \times n$, где величина n зависит от размера будущего ландшафта и определяется в конечном счете количеством точек по ширине и высоте экране. Например, массив может быть размером 100×100 или 200×200 ;

Генерируем в этот массив, во все его ячейки случайные значения;

Проходимся простым сглаживающим фильтром:

1. Берем точку (все по порядку);
2. Вокруг этой точки берем значения всех восьми точек + значение выбранной точки;
3. Суммируем все эти девять значений;
4. Делаем арифметическое усреднение с помощью деления полученного на шаге 3 значений на 9;
5. Полученный результат записываем в исходную точку;
6. Пробегам весь массив;
7. Заполняем случайными значениями еще несколько точек в исходном массиве, конкретная точка выбирается случайным образом;
8. Заново проходимся сглаживающим фильтром.

9. При необходимости получения более сглаженного ландшафта повторяем пункты 3-4 еще несколько раз.

В приведенном алгоритме необходимо учитывать, что выполнение пунктов 2–4 для первой и последней строки и столбца не будет иметь смысла, в следствие того, что вокруг каждой точки будет не восемь значений, а только пять. В этом случае следует либо игнорировать усреднение в данных точках массива и заранее установить их в ноль, либо для первой строки и первого столбца суммировать с n -м, а для n -х – соответственно с 1-й.

2.2. Построение ландшафтов с помощью иррегулярной сетки

Другой способ построения трехмерных ландшафтов – использование иррегулярной сетки вершин и связей их соединяющих. Зачастую такие решения применяются в специализированных пакетах для игр или специальных пакетах для работы с трехмерной графикой, типа 3Dmax. Данные хранятся в виде трехмерных моделей.

Трехмерный ландшафт, сгенерированный по данному способу выглядит следующим образом (Рис. 3):

Алгоритм построения такого ландшафта крайне прост:

1. Случайным образом на предполагаемой карте ландшафта располагаем вершины будущего ландшафта. Количество вершин также как и в предыдущем способе зависит от размера рабочей области экрана и размерности необходимого ландшафта;

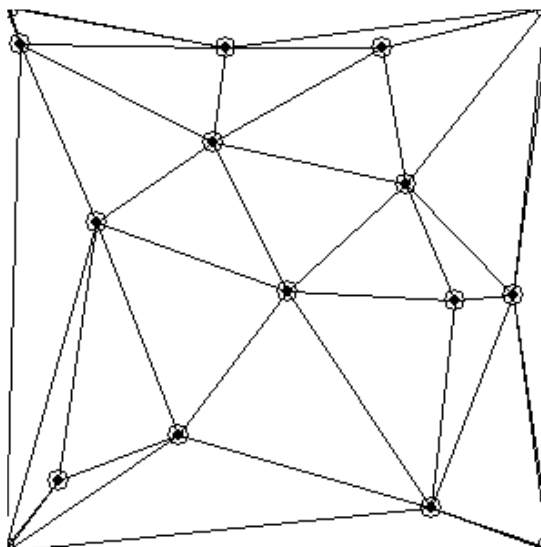


Рис. 3. Ландшафт на основе иррегулярной сетки вершин и связей

2. Соединяем все вершины связями таким образом, чтобы каждая вершина была соединена связью с каждой из соседних вершин;

3. При необходимости окрашиваем готовый ландшафт каким-либо цветом.

2.3. Сферические ландшафты

Другой тип ландшафтов представляют сферические ландшафты, которые применяются в области космической науки и техники, например, для создания неискаженных континентов на планетах.

Этот метод крайне прост, но труден в реализации и состоит из следующих шагов:

1. Взять сферу.
2. Разделить ее пополам следующим образом (Рис. 4)

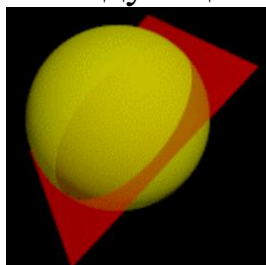


Рис. 4. Пример разделения сферы

3. Сделать одну половину немного больше
4. Сделать другую половину немного меньше.

5. Повторить этот процесс много раз, выбирая при этом каждый раз случайный путь обхода половин сферы.

В результате после одной итерации вид сферы будет следующим (Рис. 5):

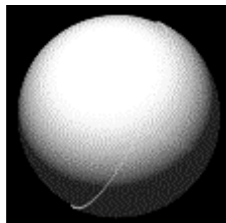


Рис. 5. Ландшафт после одной итерации
После двух итераций следующим (Рис. 6):

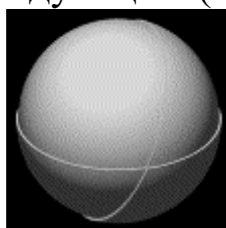


Рис. 6. Сфера после двух итераций
После трех итераций вид сферы будет таким (Рис. 7):

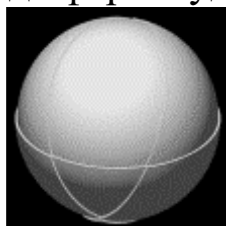


Рис. 7. Ландшафт после трех итераций

После большого количества итераций, небольшие горные хребты на сфере начинают приобретать форму континентов. В конечном счете, небольшие горные хребты исчезают, постепенно превращаясь в горные цепи. Если задать большое количество итераций, то образец земли через некоторое время приобретет вид планеты.

После всего 100 итераций уже видны очертания континентов (Рис. 8).

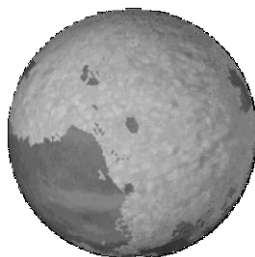


Рис. 8. Ландшафт после 100 итераций

После 1000 итераций начинают появляться очертания хребтов (Рис. 9).

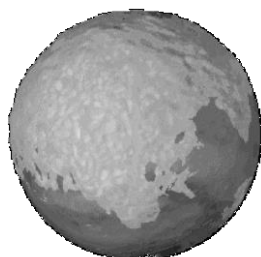


Рис. 9. Ландшафт после 1000 итераций

10000 итераций дает вид достаточно больших гор (Рис. 10).



Рис. 10. Ландшафт после 10000 итераций

Для реализации этого метода возможно два способа, выбор которого зависит от того, что необходимо получить в конечном итоге. Один метод заключается в генерации карты текстуры, которая может быть «обернута» вокруг сферы без искажений. Другой метод основан на создании 3D модели планеты, используя полигоны.

2.4. Метод 3D полигонов

Этот метод дает более качественные результаты, по сравнению с предыдущим методом, но требует больших затрат памяти. Планета в этом случае содержит более 10000 полигонов.

В первую очередь необходимо выбрать вектор нормали к точке, из которой ведется наблюдение. Процедура для этого может выглядеть, например, следующим образом.

```
procedure Normals //случайным образом выбирает проекцию
                  //нормали
```

```
VECTOR: n
```

```
loop
```

```
  n = VECTOR( random(-1,1), random(-1,1), random(-1,1) )
```

```
  magnitude = mag(n)
```

until magnitude(n) < 1

Сначала необходимо задать сферу. Чем больше количество вершин в ней будет, тем лучше. Также, лучше иметь сферу с распределением вершин поперек ее поверхности.

После того как задана сфера, выбран вектор нормали, метод может быть реализован следующим образом.

```

loop
  n = random vector    ;см. Пример выше
  loop                ;сквозь каждую вершину сферы
    d = dotproduct( n , this vertex) ;если вершина впереди или
                                      ;позади точки наблюдения

    if d > 0 then          ; если впереди, то
      move this vertex out a little bit
    else                  ; если позади, то
      move this vertex in a little bit
    end if
  end loop
end loop
end loop

```

Для обеспечения скорости и точности работы алгоритма можно сохранять дополнительный номер к каждой вершине. Когда происходит перемещение «в» или «из» точки, достаточно добавить или вычесть 1 из номера вместо перемещения положения вершины.

Небольшой модификацией к методу, описанному в п.3 является применение нормали, выбранной в данном методе, как расстояние от точки наблюдения до центра сферы.

Центр сферы определен вектором n. Это является и нормальным вектором к точке наблюдения и одновременно расстоянием от центра сферы до точки наблюдения (Рис. 11).

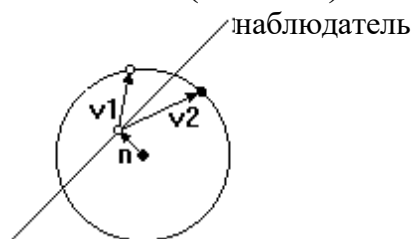


Рис. 11. Выбор нормали к точке наблюдения

На диаграмме отмечены две точки: одна из них впереди точки наблюдения, а другая позади этой нее. Из диаграммы видно, что вектор v_1 , начинающийся из конца вектора n и заканчивающийся в точке, отмеченной на диаграмме белым кружочком, больше единицы, а вектор v_2 и вектор n меньше единицы. Тогда соответствующая процедура может выглядеть следующим образом:

```

loop
    n = random vector          ;выбрать случайный вектор по
                               методу, описанному выше
    m = randomly either -1 or 1

loop                            ;по всем вершинам сферы

    p = координаты данной вершины
    v = p - n
    d = dotproduct(n , v)      ; вершина впереди
                               ; или позади точки наблюдения

    if d > 0 then                ; если впереди, то
        move this vertex by m
    else                          ; если позади, то
        move this vertex by -m
    end if
end loop

end loop

```

В данной процедуре введен новый вектор m . Все точки сферы позади точки наблюдения «чуть-чуть» смещены за сферу, а точки впереди сферы – соответственно «чуть-чуть» перемещены в сферу. Всякий раз, когда выбирается новая точка наблюдения, центр сферы находится позади точки наблюдения. В данном случае делается исключение для случая, когда точка наблюдения проходит через середину. Это означает, что большинство точек находятся позади точки наблюдения. В результате планета состоит полностью из земли.

Для того, чтобы это исправить в процедуре, описанной выше случайным образом проверяется, являются ли точки сферы впереди или позади точки наблюдения. В случае, если являются, делается замена положения точки сферы. Переменная *m* является индикатором: '-1' означает «немного внутри», а '+1' – «немного вне».

3. Задание на лабораторную работу

Написать программу для моделирования трехмерного ландшафта с помощью любого из описанных в теоретической части методов. Программа может быть написана на языке программирования Turbo-Pascal, Borland-Pascal, Object Pascal, Turbo-C, Borland-C++, C++ Builder. Программа обязательно должна содержать следующие элементы:

- В случае использования метода карт высот должен задаваться коэффициент сглаживания поверхности, который должен быть задан пользователем;
- Необходимо предусмотреть возможность повторной прорисовки ландшафта путем добавления в форму проекта программы специальной кнопки, которой должна быть назначена эта функция;
- Необходимо предусмотреть возможность явного задания размеров ландшафта путем добавления в форму проекта программы специальных полей, в которую пользователь может задавать размер предполагаемого ландшафта;
- Ландшафт должен быть изображен в цвете.

4. Контрольные вопросы

1. Какие существуют основные методы для построения трехмерных ландшафтов?
2. В чем состоит основной принцип сглаживания ландшафтов?
3. Охарактеризуйте метод регулярной карты высот. В чем состоят его достоинства и недостатки?
4. Охарактеризуйте метод иррегулярной сетки вершин и связей. Опишите его достоинства и недостатки.

5. Сравните достоинства и недостатки метода регулярной карты высот и иррегулярной сетки вершин и связей.
6. На чем основан метод сферических ландшафтов?
7. В чем отличие метода 3D полигонов от метода сферических ландшафтов?
8. Для чего в методе 3D полигонов применяется вектор нормали?
9. В чем состоит модификация метода 3D полигонов?
10. Для чего в модификации метода 3D полигонов введена дополнительная переменная m ?

5. Содержание отчета

Результаты выполнения лабораторной работы оформляются в виде отчета, который должен содержать следующие разделы:

1. тема лабораторной работы;
2. цель работы;
3. общая часть и индивидуальный вариант задания;
4. описание алгоритма (граф-схема алгоритма);
5. текст программы;
6. результаты моделирования;
7. анализ результатов и выводы.

Библиографический список

1. Ю. Тихомиров. Программирование трехмерной графики, ВНУ, СПб.– 1999 г.
2. http://freespace.virgin.net/hugo.elias/models/m_landsp.htm
3. www2.unil.ch/biomapper/opengl/Landscapes.html
4. Е.В. Шикин, А.В. Боресков. Компьютерная графика. Динамика, реалистические изображения.– М.: «Диалоги–Мифи», 1995 г.
5. Н. Томпсон. Секреты программирования трехмерной графики для Windows 95. СПб: Питер, 1997. – 352 с.
6. Д. Роджерс. Алгоритмические основы машинной графики. Пер. с англ. – М.: Мир, 1989. (2-ое издание – 2001 год).

Е.А. Никулин. Компьютерная геометрия и алгоритмы машинной графики.: БХВ-Петербург, 2003.– 560 с.

М.В. Михайлюк. Основы компьютерной графики.: ИТЦ «МАТИ», 2001.– 194 с.

Романычева Э.Т., Соколова Т.Ю., Шандурина Г.Ф. Инженерная и компьютерная графика.: ДМК Пресс, 2001.– 592 с.

Френсис Хилл. OpenGL. Программирование компьютерной графики. Для профессионалов.: Питер, 2002.– 1088 с.