А. В. Ельфимов, А. Г. Ельфимова, Т. Н. Кульман

РИСОВАНИЕ ФРАКТАЛОВ

Филиал «Протвино» университете «Дубна»
Кафедра информационных технологий

Целью данной работы являлось изучение фракталов и освоение рекурсивного программирования для рисования не только двумерных, но и трёхмерных фракталов.

Математика, если на нее
правильно посмотреть,
отражает не только истину,
но и несравненную красоту.
Бертранд Рассел

Фрактал — геометрическая фигура, обладающая свойством самоподобия, то есть составленная из нескольких частей, каждая из которых подобна всей фигуре в целом. Понятие «фрактал» появилось в конце 70-х годов XX столетия. Первой книгой по фракталам была книга Б. Мандельброта «Фрактальная геометрия природы». Многие объекты природы обладают фрактальными свойствами, с их помощью могут быть описаны и изображены (такие как облака, деревья, горы и т. д.).Фракталы можно разделить на несколько групп [1, 2]:

- геометрические — они получаются с помощью некоторой ломаной (или поверхности — в трехмерном случае). К каждой части ломаной применяют набор правил, порождающих новые фрагменты этой линии и т. д. Примеры геометрических фракталов : треугольник Серпинского (рис.1), кривая Коха (рис.2), Драконова ломаная и др.
- алгебраические — строятся на основе нелинейных функций. Возникают такие фракталы при исследовании динамических систем. Примерами могут служить: множество Мандельброта (рис.3), множество Жюлиа, фрактал Ньютона и др. Раскрашивание точек плоскости в зависимости от поведения систем приводит к удивительным по красоте графическим образам.

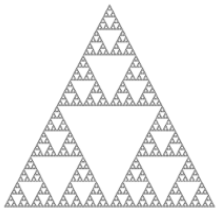

Рис. 1. Треугольник Серпинского

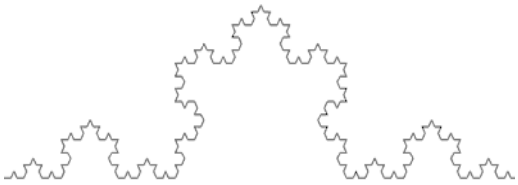

Рис. 2. Кривая Коха

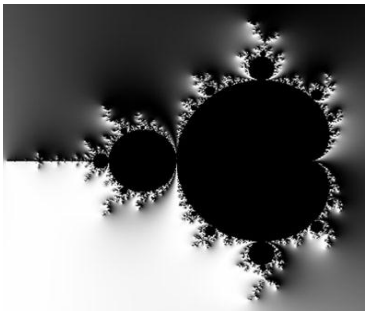

Рис. 3. Множество Мандельброта

Стохастические — получаемые при изменении каких-либо параметров итерационного процесса случайным образом. Наиболее часто упоминаемым примером является траектория броуновского движения на плоскости и в пространстве. Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.

Изучение фракталов представляет большой интерес для математиков и программистов, поскольку имеет большое прикладное значение. Перечислим лишь несколько сфер применения фракталов:

- в физике фракталы естественным образом возникают при моделировании нелинейных процессов (турбулентность, горение);
- в биологии применяются для моделирования популяций и для описания систем внутренних органов (бронхиальное дерево, сеть кровеносных сосудов);
- в радиотехнике — для проектирования антенных устройств;
- в информатике — для сжатия изображений, для сжатия информации (для компактного сохранения информации об узлах сети);
- в компьютерной графике — для изображения природных объектов [3];

- в экономике — для анализа биржевых котировок.

Использование фракталов для сжатия может дать очень хорошие результаты. В лучших примерах сжатие происходило до 1000 раз. Фрактальное сжатие можно применить даже для тех изображений, в которых не заметны очевидные элементы самоподобия. К примеру, компания Microsoft, выпускавшая *MicrosoftEncarta* с 1993 по 2009 гг, записала на диск 7 часов звука, 100 анимационных роликов, около 800 масштабируемых карт, а также 7.000 качественных фотографий — и всё это на обычный компакт-диск в 650 мб [6].

Для получения фракталов используется рекурсивная функция, где задаётся переменная *n* — количество итераций или условие окончания рекурсивного процесса. Указав конечное количество итераций, можно получить тот или иной фрактал. При создании фрактала вызывается функция, которая создаёт минимальную часть фрактала, в ней идёт вызов этой же функции, и так до максимального количества итераций. При сжатии изображений происходит поиск самоподобных частей различными методами. Такой вид сжатия довольно долгий, так как программа перебирает множество различных вариаций.

В данной работе выполнялось программирование следующих фракталов : снежинки Коха, треугольника Серпинского, множества Мандельброта, некоторых других простых фракталов.

Исследовались возможности специальной *программы L*-системы [2, 5], позволяющей генерировать фракталы. Идея *L*-системы заключается в том, что каждый шаг рисующего пера»управляется несколькими параметрами, например, движение вперёд или назад, поворот направо или налево, запоминание позиции и др. С математической точки зрения, *L*-система является грамматикой. В оригинале [2] возникает фрактал в виде ветки растения. Были разработаны алгоритмы прорисовки фракталов, которые содержат изменения параметров рисования (углы, длина шага, последователь-ность различных видов шагов и пр.). В результате фрактал преобразовывался в ветки хвойных растений, в деревья и т.д. (рис. 4). В оригинале фрактал был написан на языке программирования *VisualBasic.NET* и был переписан на язык программирования *VisualC#.*

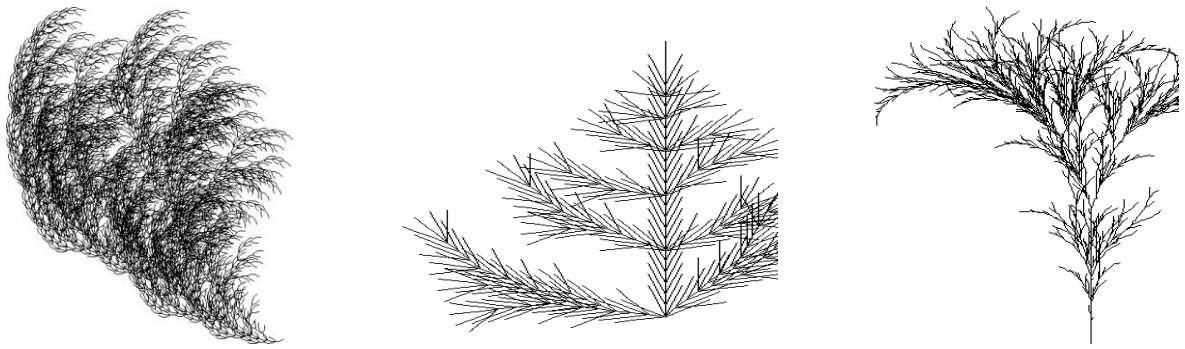

Рис. 4. Использование *L*-системы

Далее проводились преобразования двумерных фракталов в трёхмерные с использованием языка *VisualC#* и библиотеки *DirectX* [4] (рис.5, 6).

MicrosoftDirectX — это ряд технологий, благодаря которым компьютеры на основе *Windows* становятся идеальной средой для запуска и отображения приложений, богатых элементами мультимедиа, такими как цветная графика, видео, трехмерная анимация и стереозвук. *DirectX* включает обновления, повышающие безопасность и производительность, а также новые функции, относящиеся к различным технологиям, к которым приложение может обращаться с помощью *DirectX API.*

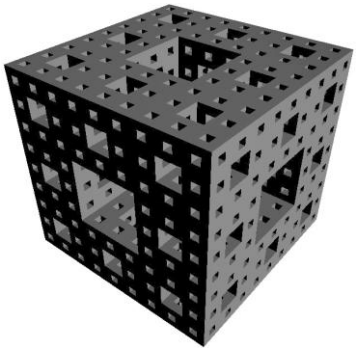

Рис. 5. Губка Менгера

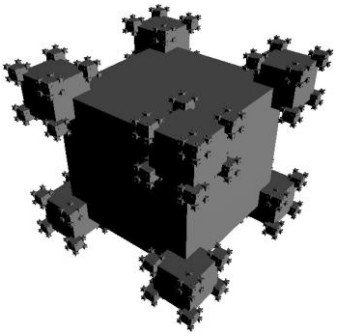

Рис. 6. Простой фрактал

Пример фрагмента кода губки Менгерав *Visual C#* с использованием *DirectX:*

```
voidDrawSQube(inti, float S, Vector3 Pos)//i-итерация, S-размер, Pos-позиция
{i--;//Уменьшение шага итераций
if (i == 0) // Рисование куба
{   // Присваивание новой позиции для куба
device.Transform.World = Matrix.Translation(Pos);
mesh = Mesh.Box(device, S, S, S); // Присваивание размеров куба
mesh.DrawSubset(0); // Прорисовка куба
   }
Else // Рекурсия
   {
S / = 3; // Уменьшение размера
        //Рекурсия с новыми параметрами
DrawSQube(i, S, new Vector3(Pos.X - S, Pos.Y + S, Pos.Z + S));
…
}}
protected override void OnPaint(PaintEventArgs e)
{//Закрашиваниеэкрана
device.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.White, 1.0f, 0);
device.BeginScene(); // Созданиесцены
DrawSQube(4, 6.9f, new Vector3(0, 0, 0)); // Запуск
device.EndScene(); // Завершение создания сцены
device.Present(); // Представление сцены пользователю
}
```

Функция *OnPaint* создаёт сцену, где вызывает функцию *DrawSQube*, в которую передаёт количество итераций, размер и позицию нового объекта. Функция *DrawSQube* при ненулевой итерации создаёт рекурсию, куда передаётся сокращённое количество итераций, уменьшенный размер и новая позиция. Когда переменная итерации равна нулю, то есть рекурсия достигла финальной точки, рисуется один объект : куб. Таким образом, при большем количестве итераций происходит рисование множества маленьких кубов, из которых и состоит фрактал.

Прорисовка фракталов требует довольно больших затрат системных ресурсов. В зависимости от того, в каком виде будут прорисовываться фракталы (2*d* или 3*d)* и от сложности их структуры, будут отличаться требуемые системные характеристики для их реализации. Для примера возьмём нашу программу «Губка Менгера». Каждая итерация увеличивает количество создаваемых кубов в 20 раз, в итоге их количество может достигать 1/20/400/8000/160000 и т. д. При четырёх итерациях программа требует всеголишь 184 мб оперативной памяти, но при пяти итерацияхпрограмма требует3.300 мб, что невозможно для большинства стандартных ПК. Время создания фрактала при четырёх итерациях всего 1.1 сек, тогда как при пяти итерациях — 43.1 сек.

Библиографический список

1. *Вирт Н.* Алгоритмы и структуры данных : Пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПБ. : Невский Диалект, 2001, — 352 с.
2. *Климов А. П.* Занимательное программирование на *Visual Basic.NET*. / А. П. Климов. — СПБ : БХВ-Петербург, 2005. — 528 с.

3. *Блинова Т. А*. Компьютерная графика / Т. А. Блинова, В. Н. Порев // Под ред. В. Н. Порева. — К. : Издательство Юниор, Киев, 2005. — 520 с.
4. *Миллер Т. ManagedDirectX* 9 с управляемым кодом. Программирование игр и графика. / Т. Миллер — М. : Издательство КомБук, Москва, 2005. — 400 с.
5. *http://www.codenet.ru/progr/fract/intro6.php — L-Systems.*
6. *http://compression.ru/arctest/descript/fract-comp.htm* — о сжатии информации.