Государственное бюджетное общеобразовательное учреждение

города Москвы

«Гимназия № 1505 «Московская городская педагогическая гимназия-лаборатория»»

**ДИПЛОМНОЕ ИССЛЕДОВАНИЕ**

на тему:

**Фрактальная размерность изображений**

Авсатов Александр Эльдарович 10В

Руководитель

Ветюков Дмитрий Алексеевич

Москва

2016/2017 уч.г.

**Содержание**

1. Введение
2. Теоретическая часть
   1. Понятие «фрактал»
      1. Классификации фракталов
      2. Применение фракталов
   2. Понятие «Фрактальная размерность»
      1. Метод определения фрактальной размерности
      2. Применение фрактальной размерности
3. Практическая часть
   1. Структура программы
   2. Примеры результатов
4. Заключение
5. Список литературы

**Введение**

Что такое фрактал и фрактальная размерность? Какое применение им можно найти в жизни. Многие из нас знают из математики начальной школы, что фрактал — это постоянное повторение какой-то геометрической фигуры. Фрактал имеет свое применение в высшей математике. Но на этом, объем наук, в которых есть понятие фракталов не заканчивается: фракталы существуют в информатике, радиотехнике и естественных науках. Однако фракталы встречаются не только в науке. Часто мы можем встретится с ними в реальной жизни, например, в природе есть много примеров объектов фрактальной формы. Например, капуста Романеско имеет форму, очень близкую к фрактальной. Поэтому такое понятие как «фрактал» будет актуально для всех времен, начиная с его первого определения в 1975 году Бенуа Мандельбротом, французским математиком, которой посвятил всю свою жизнь изучению фрактальной геометрии. Вместе с понятием фрактала, было введено и понятие «фрактальной размерности». Несколько упрощая фрактальная размерность – это коэффициент фрактала, который описывает его сложность (можно сказать сколько раз повторяется большая фигура). Строгое определение может быть записано в виде формулы, которая будет приведена ниже.

Целью моего диплома является написание программы, которая будет определять фрактальную размерность космических фотоснимков. С помощью фрактальной проекции фотографии, мы сможем узнать, например, чем два одинаковых леса отличаются друг от друга, и понять, болен ли один из них на примере остальных лесов. Написанию соответствующей программы на основе С# посвящена вторая глава моего диплома. Так же целью моего диплома является объяснение таких терминов как фрактальная размерность и фрактала простым и понятным языком.

Ну что ж, приступим.

1. **Теоретическая часть**
   1. **Понятие «фрактал»**

«Фрактал» был придуман американским математиком Бенуа Мандельбротом, по праву считающимся основоположником фрактальной геометрии, в 1975 году. Это слово обозначало повторяющееся самоподобие структуры объекта изучения. Но начало всего этого «переполоха» с фракталами и самой фрактальной геометрии была книга Бенуа Мандельброта «Фрактальная геометрия природы», вышедшая в свет в 1977 году. В книге он подробно и понятно объясняет, что такое «фрактал», и где мы можем с ним встретится в повседневной жизни.

Сейчас фрактал часто используется во многих частях прикладной науки, начиная от биологии и заканчивая экономикой.

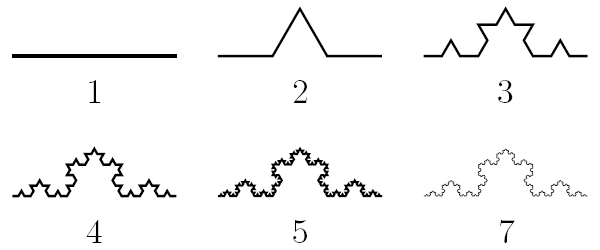
Основным свойством фрактала является частичное и полное самоподобие. Получив малую долю фрактала, мы сможем с легкостью сказать, как будет выглядеть целый.

* 1. **Классификация фракталов**

Есть много видов фракталов и методов их классификации, но в основном их делят на 3 класса.

1. **Геометрические фракталы**

Эти фракталы более понятные и являются главным наглядным примером. Они могут быть как двумерными, так и трехмерными, и в любом измерении они задаются ломанной, который называется генератором. Для создания геометрического фрактала, нужно повторение замены генератора на «образующий элемент», при соблюдении масштаба. Образующий элемент – тот элемент в фрактале, который постоянно повторяется.

  
*Рисунок 1. Построение кривой Коха.*

На примере кривой Коха (рисунок 1) мы видим, что изначально нам дан отрезок. Далее мы делим данный отрезок на три равные части, и центральную заменяем на «образующий элемент» (два отрезка под углом). На втором поколении у нас получается кривая, состоящая из 4 отрезков. К каждому из этих отрезков применяем операцию, которую мы применяли ранее. Кривая получившаяся на n-ом поколении называется предфракталом. При n, стремящемся к бесконечности, кривую Коха можно назвать фракталом.

На примере кривой Коха (рисунок1) мы видим, что создается прямая и на ней берется образующий элемент (в данном случае – 2 отрезка под углом), как на d=1. Далее данный «образующий элемент» повторяется на оставшихся участках прямой с учетом масштаба. Каждое появление «образующего элемента» в масштабе называется поколением. Кривая, получившаяся при n-ном поколении, называется предфракталом. Лишь при n, стремящейся к бесконечности, кривую Коха можно назвать фракталом.

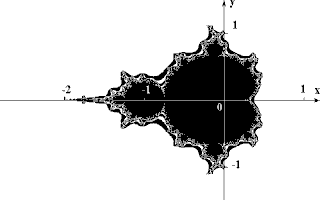


*Рисунок 2. Кривая Дракон*

Кривая «Дракон», как мы видим на рисунке, строится более ассиметричным путем. Образующим элементом данной прямой являются 2 равных отрезка перпендикулярных друг другу. Далее образующий элемент будет появляться по разные стороны от генератора (как на рисунке: один появляется внутри угла 90 градусов, а другой внутри 270 градусов). На рисунке показано построение «дракона». При n-ном поколении, где n стремится к бесконечности, данная прямая называется фракталом Хартера-Хейтуэя.

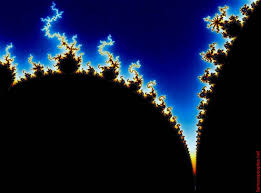
1. **Алгебраические фракталы**

Это наиболее распространенная группа фракталов. Их получают с помощью алгебраических формул, в честь которых их и назвали. Наиболее популярны двухмерные формулы, которые можно построить на графике. Алгебраические фракталы стали популярны благодаря тому, что с помощью тривиальных формул, можно составить сложно сконструированный график.

*****Рисунок 3. Множество Мандельброта.*

В качестве примера возьмем множество Мандельброта, которые мы видим на рисунках 3 и 4. Такой замысловатый рисунок получился из простой функции

где и могут принимать любое значение. Впервые эту функция была описана Пьером Фату, французским математиком. Он заметил, что эта функция имеет бесконечное количество преобразований (для каждого свое c). Он пытался построить эту функцию, но нужное количество вычислений было бы невозможно произвести вручную, а то время не было компьютеров. Мандельброт стал первым, кто воспользовался компьютером, для вычислений значений данной функции. Она получается конечна, мы видим кардиоиду (само черное пятно). Но эта кардиоида окружена мелкими фигурами, которые в свою очередь состоят из меньших фигур, и так до бесконечности (подробнее по ссылке: <http://imgur.com/MSRy5k8>).

*****Рисунок 4. Часть множества Мандельброта в масштабе.*

На множестве черным цветом отмечены те точки, которые при бесконечном количестве итераций (повторов функции) не уходят в бесконечность. Те же точки, которые уходят в бесконечность, отмечены белым цветом.

1. **Стохастические фракталы**

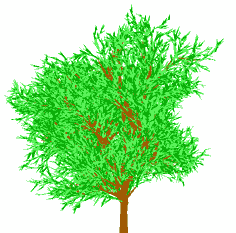
Эти фракталы образуются, если при применении алгоритма построения фрактала, менять значения *c*. Тогда, изображения получаются более живыми, в них появляется, что-то природное, появляется антисимметрия. Благодаря их близости к природе, именно эти фракталы чаще всего используются в компьютерной графике. Они используются для построения волн, глади моря, гор, деревьев и других рельефов местности.

|  |  |
| --- | --- |
|  |  |

*Рисунок 5. Примеры стохастических фракталов*

* 1. **Применение фракталов**

Основное свое применение фракталы находят в компьютерной графике. Благодаря своей схожести с природными объектами фракталы используются построения ландшафтов, таких как горы, деревья, морская гладь и другие. Так же в экономике они используются для анализа кривой взлёта и падений акций, а в географии для подсчёта береговых линий или длины всех дорог страны.



*Рисунок 6. Фрактальное дерево*

Очень важна роль фракталов в машинной графике. С их помощью можно описать или создать поверхности сложной структуры и формы, всего лишь изменяя коэффициенты в уравнении. И благодаря изменению параметров можно получить бесконечное количество вариантов конечного изображения. Без фрактальной геометрии не обходится ни одно облако в фильме ни одна гора, потому что это самое легкое решение для получения объектов неевклидовой геометрии (фрактальная геометрия используется также для фигур трехмерного пространства, не только двумерного). Эта графика используется при создании мультфильмов и фильмов, в которых присутствует хромакей (зеленый фон, на который накладывается графический, нарисованный фон) (пример: <http://imgur.com/LezFZwB>). Так же антенны и микрочипы получили предфрактальное строение, так как они получали тот же коэффициент усиления, что и у евклидового строения, только при меньших размерах, что пригодилось в телефонах, компьютерах и других устройствах «мобильных» устройствах.

Фракталы так же нашли свое применение в медицине, ведь многое в человеке имеет фракталоподобные структуры, например, строение нервной системы человека, строение кровеносных сосудов, структура дыхательных путей и другие.

Сердцебиение человека, также действует по фрактальным законам, и при изменении фрактального строения сердечных колебаний, можно заранее выявить болен ли человек.

Так же фракталы важны в экономике. Из-за непостоянства фондового рынка, сложно сказать будет ли цена расти или падать. Но благодаря фрактальной модели фондового рынка, которая высоко востребована специалистами, можно предсказать поведение рынка.

**2. Фрактальная размерность**

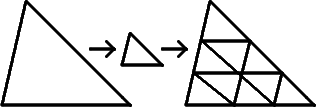
**2.1 Понятие «фрактальная размерность»**

«*Фрактальная размерность (*[*англ.*](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)*fractal dimension) — один из способов определения размерности множества в*[*метрическом пространстве*](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE) *(метрическое пространство это множество и функция в этом множестве). Фрактальную размерность n-мерного множества можно определить с помощью формулы:*

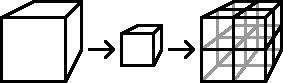
*Где -минимальное число n-мерных «шаров» радиуса ɛ, необходимых для покрытия множества.*

[*Фрактальная*](http://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB)*размерность  - коэффициент, описывающий* [*фрактальные*](http://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB) *структуры или*[*множества*](http://encyclopaedia.bid/%D0%B2%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE)*на основе количественной оценки их , как коэффициент изменения в детали с изменением масштаба.*» ©Википедия.

Получается, что размерность треугольника равна 2, так как если мы возьмём треугольник, который в 3 раза меньше исходного, то в исходном треугольнике он поместится 9 раз.

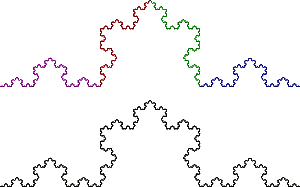


У куба же размерность получается равна 3, так как если куб уменьшить в 2 раза, то он поместится в исходном 8 раз.



Именно так и получается трехмерное и двумерное пространство. Но однако встречаются и дробные размерности.

Для примера рассмотрим вышеупомянутую кривую Коха.



Получается наш рисунок разобьется на 4 равные части (отмечены цветом), но каждая из них по длине равна трети исходной фигуры. Тогда по формуле получается:

D = ln (4)/ln (3) ≈ 1.26185950714291487419

**2.2 Математические методы определения фрактальной размерности**

Фрактальная размерность является одним из главных параметров фракталов в целом. Фрактальная размерность описывает сложность фрактала и сложность его поверхности. Она используется часто для создания или же описания трехмерных структур (рельефа местности и поверхности моря). Так же используется фрактальная размерность в технике, где с помощью нее можно найти максимальный коэффициент полезного действия того или иного устройства (пример фрактальные антенны).

Есть разные методы оценивания фрактальной размерности. Возьмём только самый основной из них.

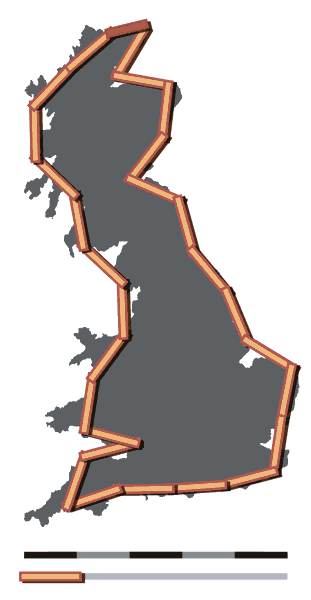
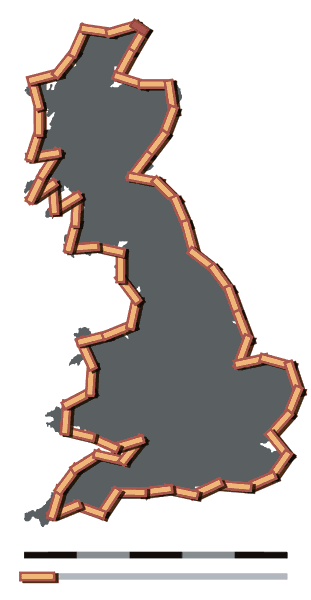
**2.2.1 Метод отрезков**

Для использования метода отрезков (yardstick method) нам нужно измерить фрактальную кривую длиной *S* отрезками размером *r*. Число отрезков n, необходимых для измерения *S* растет с уменьшением *r*. Получается,

image013

где D-искомая фрактальная размерность.

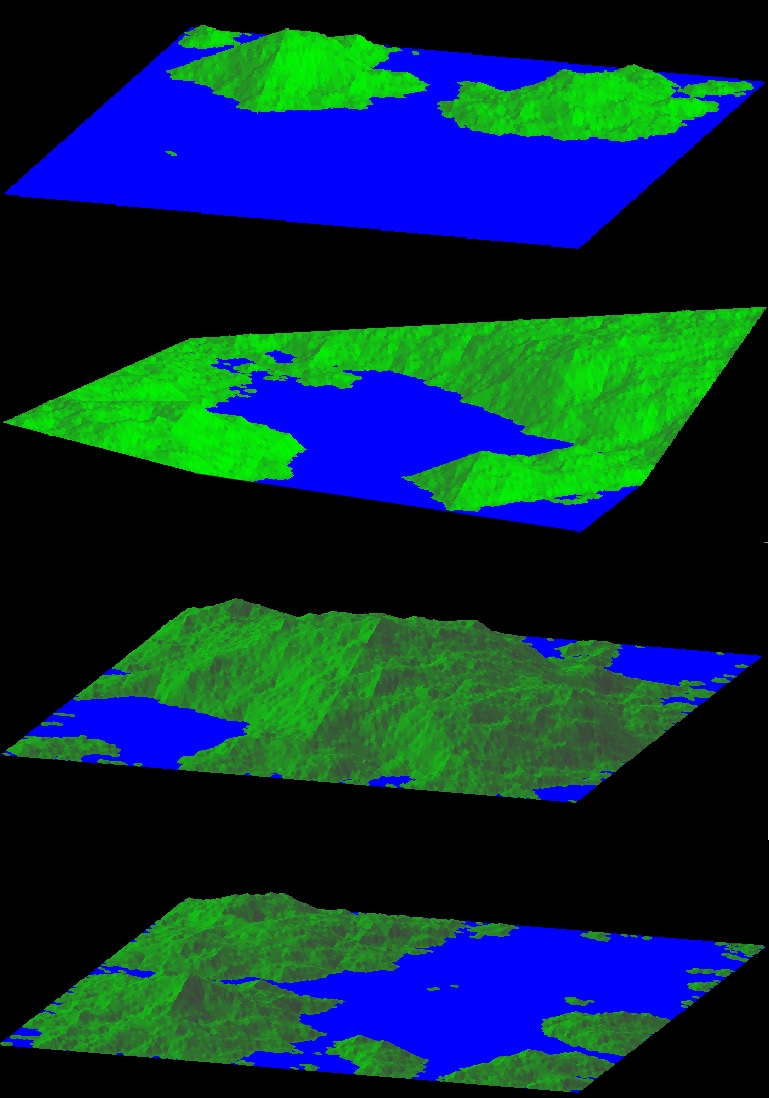
Именно на этом и построен «парадокс береговой линии». Это явление впервые было замечено Льюисом Фрай Ричардсоном. Он пытался измерить длину береговой линии Великобритании. И он заметил, что чем меньше мы возьмём «линейку», которой мы будем все измерять, тем больше получится длина береговой линии.

*Рисунок 7. Метод отрезков*

**2.3 Применение фрактальной размерность**

Чаще всего она используется в компьютерной графике, для визуализаций изображений. Так же в медицине, телекоммуникациях, во многих естественных науках (география, анатомия, биология и т.д.) и также для сжатия изображений. Также есть зависимость частоты лесных пожаров, от фрактальной размерности леса (подробнее в [файле](http://eprints.tversu.ru/3233/1/92-97%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D1%8B_%D0%B8%D0%B7_%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0_%D0%B8_%D1%83%D0%BF%D1%80%D1%8E_2010%2C_%D0%B2%D1%8B%D0%BF._8-6.pdf)).



*Рисунок 8. Рельеф, созданный с помощью программы построения геоландшафта разной фрактальной размерности.*

## Глава 2. Исследовательская часть. Программа, считающая фрактальную размерность изображения.

**2.1 Структура программы**

Наша задача написать программу считывающую фрактальную размерность изображения. В данное время, фрактальная размерность изображений лесов используется в изучении лесных пожаров. Найдена зависимость между фрактальной размерностью леса и скоростью распространения пожара в данном лесу. Также эта программа выдает нам коэффициент, с помощью которого можно узнать сложность структуры изображения. Итак, перейдем к программному коду.

Цель диплома - написать программу, которая будет делить изображение на n-ое количество квадратов и будет говорить нам, есть ли в этом квадрате черный пиксель. Далее строим зависимость логарифма количества квадратов с черным пикселем от логарифма размера стороны квадрата. Угловой коэффициент этого графика и будет наша фрактальная размерность. Начнем с самого начала.

for (int y = 0; y==img.Height; y++)

{

for (int x = 0; x==img.Width; x++)

{

cr = cr + 1;

}

Этот цикл считывает все пиксели изображения. Теперь нам нужно считать только черные пиксели. Для этого зададим ограничение по цвету, только после которого программа будет его считать. Итак, для того чтобы это сделать, мы должны определить цвет пикселя. Для этого на понадобится команда file.GetPixel(i, y), где I и Y месторасположение пикселя. Но так как нам нужен его цвет, то нам нужна ячейка color, где будут цвета выбранного пикселя.

Color c = file.GetPixel(i, y);

int r = c.R; //красный цвет

int b = c.B; //синий цвет

int g = c.G; //зеленый цвет

if (g == 0 && r == 0 && b == 0) {n = n + 1;}//черный цвет в параметрах rgb-(0,0,0)

Дальше нам нужен цикл, который будет делить изображение на n-ое количество квадратов, пиксели которых мы будем считывать. Каждый раз количество квадратов, на которые мы делим наше изображение, должно увеличиваться. После того как мы считали наш первый квадрат, мы должны перейти к следующему.

for (int x = 8; x <= 60; x=x+2)

{

for (int z = 1; z <= x; z++)//задаем переменную, которая будет считать номер квадрата для x

for (int m = 1; m <= x; m++)// задаем переменную, которая будет считать номер квадрата для y

for (int i = file.Width \* (z - 1) / x; i < (file.Width) \* z / x; i++)//

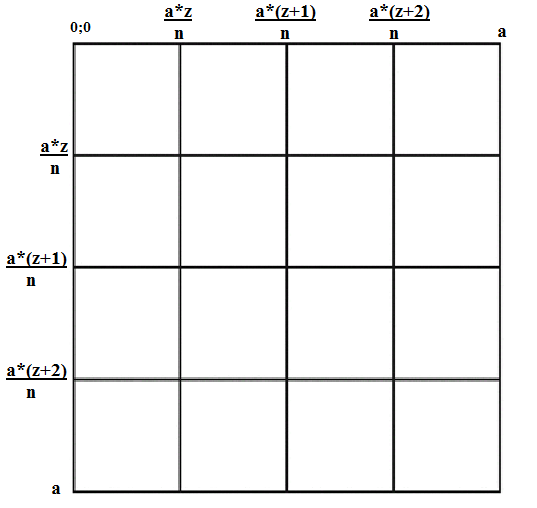
цикл, который считывает x от левого верхнего пикселя квадрата, на который мы делим, до начала следующего квадрата по x.

{

for (int y = file.Width \* (m - 1) / x; y < (file.Width) \* m / x; y++)//

цикл, который считывает y от левого верхнего пикселя квадрата, на который мы делим, до начала следующего квадрата по y.

Тогда, стороны квадратов, на которые мы делим изображение, равна ширине изображения поделённое на количество квадратов (рисунок 9 сторона квадрата равна a/n).



*Рисунок 9. Пояснение к программе*

Получается сначала мы проверяем координату (х, у), и если это не черный пиксель, то мы продолжаем, проверяем координату (х+1, у). Делаем это до того момента, пока пиксель, который мы проверяем не будет черным, или же пока х не достигнет конца квадрата (на рисунке 9, мы начинаем с координаты (0;0), смотрим является ли этот пиксель черным, если нет, то мы продолжаем с координаты (1;0) и так далее; если же пиксель черный, то мы добавляем в количество квадратов с черным пикселем один и начинаем со следующего квадрата, x координата которого равна (а\*z/n); если мы не находим черного пикселя, то мы увеличиваем х, пока он не станет началом следующего квадрата (a\*z/n); когда он достигнет начала следующего квадрата, мы х снова делаем равным 0, а у увеличиваем на 1, и повторяем цикл). Делаем это до того момента, пока мы не найдем черный пиксель, или же, не просмотрим все пиксели в квадрате. В том или ином случаем, мы переходим к следующему квадрату и выполняем цикл уже с ним.

for (int x = 8; x <= 60; x=x+2)

{

for (int z = 1; z <= x; z++)

for (int m = 1; m <= x; m++)

for (int i = file.Width \* (z - 1) / x; i < (file.Width) \* z / x; i++)

{

for (int y = file.Width \* (m - 1) / x; y < (file.Width) \* m / x; y++)

{

s = 0;

Color c = file.GetPixel(i, y);

int r = c.R;

int b = c.B;

int g = c.G;

if (g == 0 && r == 0 && b == 0) { n = n + 1; s = s + 1; break; }

if (s == 1) { break; }

}

if (s == 1) { break; }

}

После проверки всех квадратов изображения, мы записываем в первый массив их ширину, а в другой массив количество квадратов с черным пикселем. Получается в первом массиве хранится ширина квадрата, а во втором количество квадратов с черным пикселем при данной ширине квадрата.

{

int s = 0;

double[] size = new double[30];//массив в котором хранятся размеры квадратов

int[] count = new int[30];// массив в котором хранится количество квадратов с черными пикселями

OpenFileDialog f = new OpenFileDialog();// берем с вашего компьютера файл, размерность которого вы хотите посчитать

f.Filter = "JPG(\*.JPG)|\*.jpg";

if (f.ShowDialog() == DialogResult.OK)

{

Bitmap file = new Bitmap(f.FileName);//вставляем вашу картинку в окно программы

pictureBox1.Image = file;

int n = 0;//количество квадратов с черным пикселем

int p = -1;//на какой позиции будут стоять значения размера и количества квадратов

for (int x = 8; x <= 60; x=x+2)

{

n = 0;

p = p + 1;

size[p] = file.Width / x;//записываем в ячейку массивы размер квадрата

for (int z = 1; z <= x; z++)

for (int m = 1; m <= x; m++)

for (int i = file.Width \* (z - 1) / x; i < (file.Width) \* z / x; i++)

{

for (int y = file.Width \* (m - 1) / x; y < (file.Width) \* m / x; y++)

{

s = 0;

Color c = file.GetPixel(i, y);

int r = c.R;

int b = c.B;

int g = c.G;

if (g == 0 && r == 0 && b == 0) { n = n + 1; s = s + 1; break; }

if (s == 1) { break; }//выходим из цикла x

}

if (s == 1) { break; }//выходим из цикла у

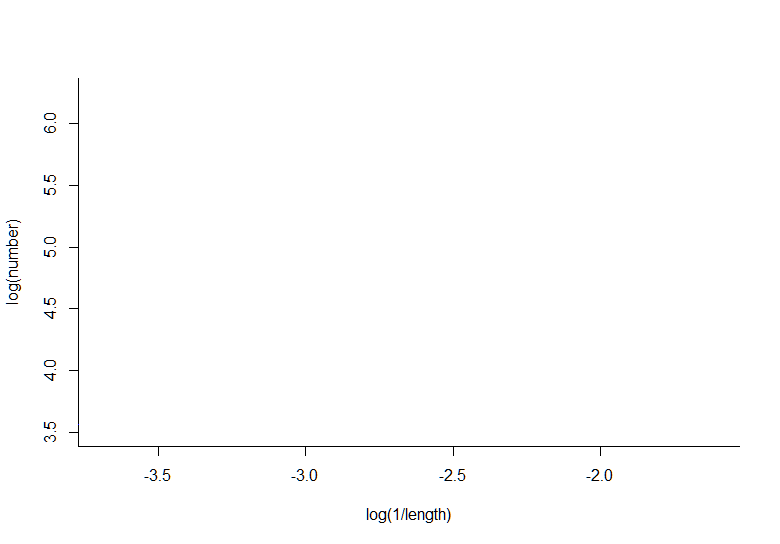
}

count[p] = n;//записываем в ячейку количество квадратов

}

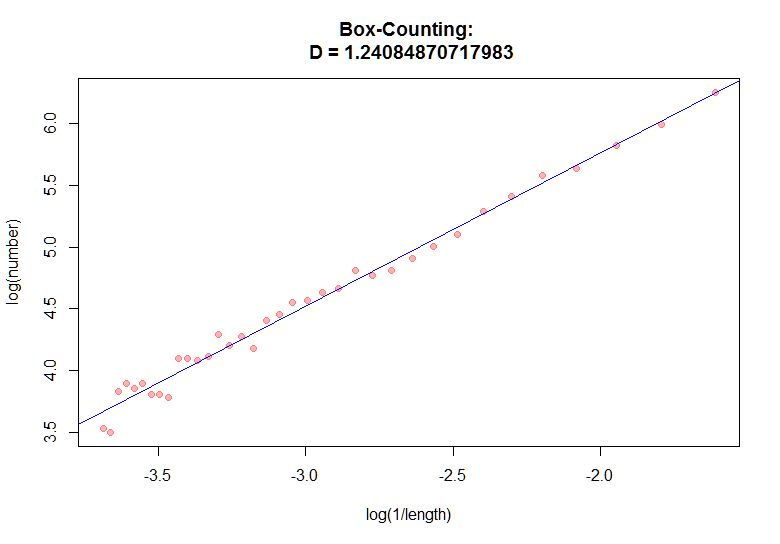
}

Далее, для того, чтобы найти фрактальную размерность нужно найти угловой коэффициент, по графику:



*Рисунок 10. График построения*

На этот график мы выносим значения наших массивов. Получается что-то вроде этого:



*Рисунок 11. Пример графика*

Для того чтобы найти угловой коэффициент, воспользуемся формулой k=(y2-y1)/(x2-x1).

double[] raz = new double[30];// массив для углового коэффициента

double o;// ячейка для суммы угловых коэффициентов

o = 0;

for(int i=0; i<21;i++)

{

raz[i]= ((Math.Log(count[i+4]) - Math.Log(count[i])) / (Math.Log((1 / size[i+4])) - Math.Log(1 / size[i]))) ;//находим угловой коэффициент

o =o+raz[i];//получаем сумму

}

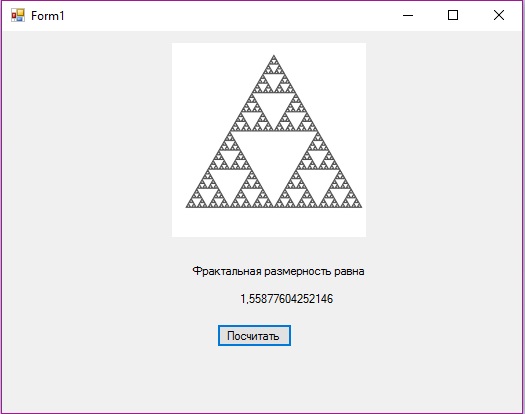
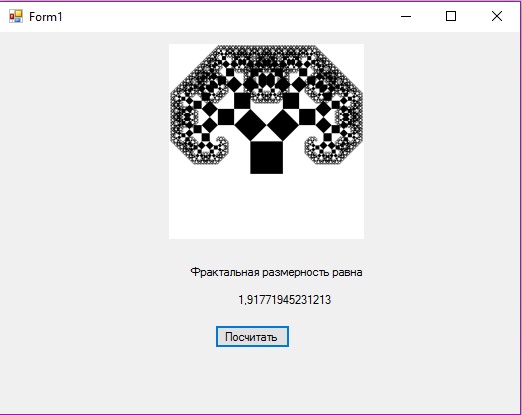
label1.Text = (o/20).ToString();// в ячейку с ответом получаем среднее значение.

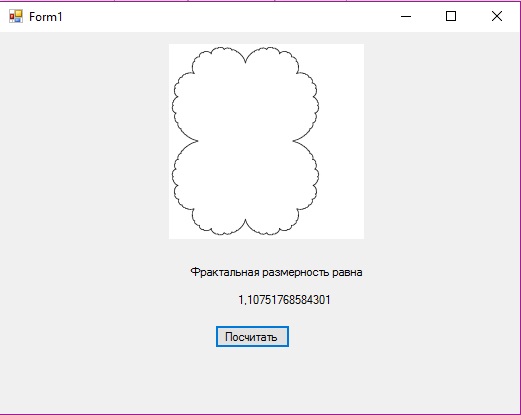
Минус программы в том, что она может посчитать размерность только для квадратных изображение, из-за чего размерность получается неточной, так как мы должны прямоугольный рисунок сделать квадратным, что добавляет ему дополнительный размер. Из-за это значение погрешности страдает (+-0,2).

**2.2 Примеры результатов**









**Заключение**

Итак, мы с вами убедились, что фракталы достаточно молоды, и в перспективе их объем в науке будет только расширятся. Даже сейчас спектр наук фракталов достаточно велик. Так же мы узнали самый главный коэффициент фракталов – фрактальную размерность и научились ее измерять. Так же я написал программу, которая за вас будет находить фрактальную размерность изображения.

**Список литературы**

1. Крупнейший в Европе ресурс для IT-специалистов

2. Ресурс для программистов и интересующихся <http://www.michurin.net/fractals/dim.html>

3. Википедия — свободная энциклопедия <https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB>

4. Википедия — свободная энциклопедия <https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C>

5. Алгоритмы и методы <http://algolist.manual.ru/graphics/fracart.php>

6. Все о фракталах <https://m-rush.ru/theory/item/173-fraktaly-v-ekonomike.html>