

A0:IFS (迭代函数系统)

一、概述

1.目标: 熟悉 C++和两个简单的库。同时了解奇异分形。

2.IFS 介绍: 分形几何用来描述不规则的几何形态,例如海岸线、山脉、树叶等。IFS 就是一种构建分形的系统。IFS 具有自相似性,即物体的局部和整体是相似的。最经典的例子是 Barnsley's fern, 每一片蕨叶都和整体的蕨叶相同。

3.算法介绍: IFS 被一系列的仿射变换所描述。IFS 可以在任意维度上定义,但本次实验只在二维上定义。通常,1 个 IFS 被 n 个仿射变换定义,每个变换 f_i 必须是收缩的,这意味着点之间的距离必须减小。IFS 的一个吸引子 A 是一个固定点。

具体来说,渲染一个 IFS 分形图形,通过在一个单位正方形中采样随机点,并对这些点进行 N 次仿射变换。选取变换矩阵的每个概率都是被提前指定的(在不同的区间上),而不是每个矩阵的选取概率都相同。算法伪代码描述如下:

```
for "lots" of random points (x0, y0)
    for k=0 to num_iters //迭代次数
        pick a random transform fi //随机选取变换矩阵
        (xk+1, yk+1) = fi(xk, yk) //下一个点通过对上一个点的变换得到
        display a dot at (xk, yk) //对变换得到的点着色
```

二、实现细节

1.接收输入:

输入格式:

```
-input sierpinski_triangle.txt -points 10000 -iters 0 -size 200 -output sierpinski_triangle_0.tga
```

因为作业提示给的解析方式需要用 cmd 打开程序,比较繁琐,我改成了直接在控制台程序中输入命令。

```
string tempStr;
cin >> tempStr >> input_file;
cin >> tempStr >> num_points;
cin >> tempStr >> num_iters;
cin >> tempStr >> m_size;
cin >> tempStr >> output_file;
```

2. IFS 类:

```
class IFS{
public:
    IFS ();
    ~IFS ();
    void Input (const char *file);
    void Render (Image *image, const int &num_points, const int &num_iters);
    void CleanUp ();
private:
    int num_transforms; //the number of transformations
    Matrix *matrixArray; //变换矩阵数组
```

```
float *matrixRandom; //选择矩阵的概率数组  
};
```

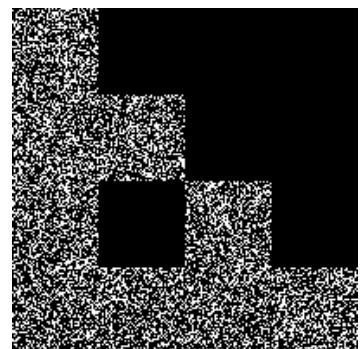
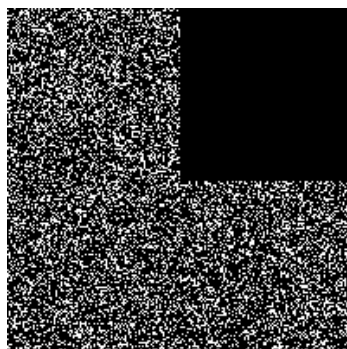
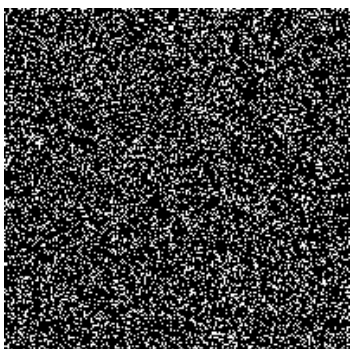
3.渲染过程:

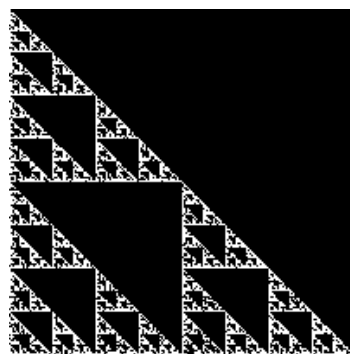
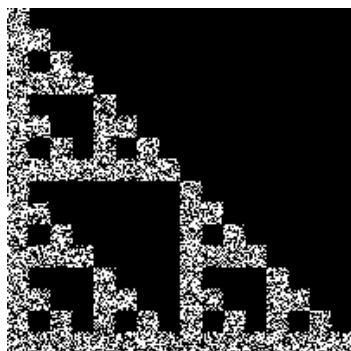
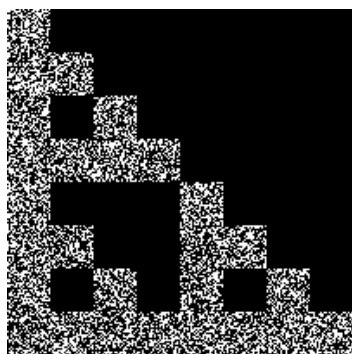
```
void IFS::Render (Image *image, const int &num_points, const int &num_iters){  
    int width = image->Width ();  
    int height = image->Height ();  
    for (int i = 1; i <= num_points; i++) { //绘制点总数  
        Vec2f V0 = Vec2f (rand () / (RAND_MAX + 0.0), rand () / (RAND_MAX + 0.0));  
        for (int k = 0; k < num_iters; k++) { //迭代次数  
            float p = rand () / (RAND_MAX + 0.0); //0~1之间的随机数  
            float a = 0.0; int j=0;  
            for (j = 0; j < num_transforms; j++) { //遍历概率数组, 找到p对应的区间  
                float b = a+matrixRandom[j];  
                if (p >= a && p < b) break;  
                a = b;  
            }  
            matrixArray[j].Transform (V0);  
        }  
        if (V0.x () >= 0 && V0.x () <= 1 && V0.y () >= 0 && V0.y () <= 1)  
            image->SetPixel (V0.x ()*width, V0.y ()*height, Vec3f (1, 1, 1));  
    }  
}
```

- 1) 要在一张 tga 图片上绘制分形图形, 是通过对其上的某些像素点着色实现的, 这个通过 Image 类中的 SetPixel 函数完成。
- 2) 而具体在图片上怎么选取这些像素点, 则需要把输入的一系列随机点的 x,y 坐标都映射到 0 到 1 之间, 然后再根据 Image 的长和宽确定绘制位置。
- 3) 对于将要绘制的每个点, 先生成一个随机固定点 V0, 然后对这个点进行 k 次迭代, 每次迭代时, 随机选取一个变换矩阵 (如果迭代 0 次, 点就会近似均匀地分布在整个正方形内)。
- 4) 然后对每个经过变换 k 次后的点进行判断, 如果 x,y 坐标都还在 0 到 1 的区间内, 就进行着色。

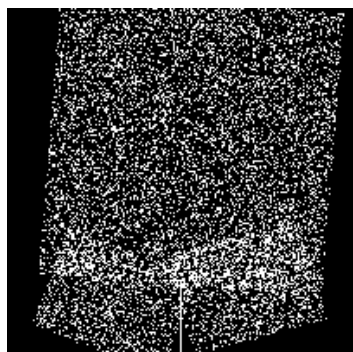
三、结果展示

1.谢尔宾斯基三角形:

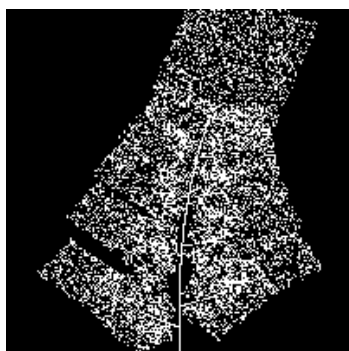




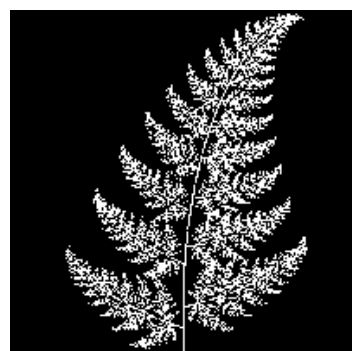
2. 巴恩斯利蕨



k=1



k=5



k=30

四、心得体会

前期审题是相当重要的，最好先查好背景资料，把思路捋清再动手。IFS 分形算法对于每个随机点都进行 k 次仿射变换，观察谢尔宾斯基三角形的 IFS 码可以很直观地看出做了平移和缩放。但是对于比较复杂的图形比如巴恩斯利蕨，IFS 码就比较复杂，看了一下资料发现需要进行比较多的计算和调试。虽然如此，越复杂的图案也就越显示出了分形的神奇！