Union and Difference of Difference based Fractals using IFS

B Dinesh Rao
Associate Professor
SOIS, Manipal
Manipal University

H Harishchandra Hebbar
Professor
SOIS, Manipal
Manipal University

Sudhakara Upadya P
Assistant Professor (Sen. grade)
SOIS, Manipal
Manipal University

ABSTRACT

Fractals are self similar images. There are many techniques for generating fractals. IFS[1] are one of them. IFS use a set of linear transformations for generation of fractals. IFS have been modified to difference based IFS[2] to use differences in distance between the points to figure out the new point. Two different fractals have been plotted and the union and difference between the fractals have been explored. Two fractals have been combined in various ways in this paper. Non-linear fractals have been developed by Frederic Raynal, Evelyne, Lutton and Pierre Collet [1].

Keywords

.IFS, Fractals, Union, Difference

1. INTRODUCTION

The use of a system of affine transforms to define a fractal object has been described by Barnsley [3]. A system of transforms W_i can be written as

$$W_i\text{:}\ Z \rightarrow T_i Z + V_i$$

Where

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad V = \begin{bmatrix} e \\ f \end{bmatrix}$$

and

$$Z = \begin{bmatrix} x \\ y \end{bmatrix}$$

That is if $Z = \begin{bmatrix} x \\ y \end{bmatrix}$

then $x = ax + by + e$

$y = cx + dy + f$

The set of transforms need to be contractive and there exists a unique attractor set containing infinitely many points Z. For graphical purposes we say that there is a fixed set of pixels which approximate this attractor [4][5].

Our proposed technique is intended to add two transforms with different varying factors. We wish to demonstrate that by adding two transforms or fractals to generate a new fractal. We will demonstrate that the usage of the above equations also generate fractals.

2. MATERIALS AND METHODS

2.1 Iterated Function Systems

We will first look at a method for generating IFS. Consider the following three transformations.

1. $x = 0.5\ x + 0$
   $y = 0.5y + 0$
2. $x = 0.5x + 0$
   $y = 0.5y + 150$
3. $x = 0.5x + 150$
   $Y = 0.5y + 0$ (Equation 1)

If we select an arbitrary point x, y and select one of the three affine transformations in random, apply it to the point x, y. we get a new point x', y'. If the above procedure is repeated on the point x', y', we get a new point x", y". This procedure is repeated for fixed number of times. The resultant fractal is depicted in figure 1. It is called the Sierpensky's triangle.

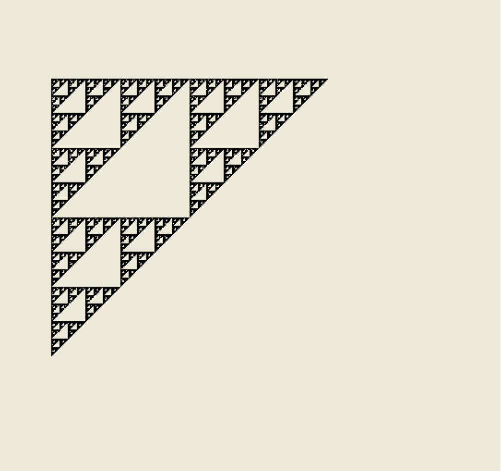

Figure 1: Sierpensky's triangle

2.2 Difference based IFS variant

We modify IFS methodology slightly to get meaningful images in a more intuitive way. We will call this difference based IFS. In IFS, each function is scaled by a constant between 0 and 1. It is generally difficult to predict the result of application of this scaling factor. In difference based IFS, we are going closer to a fixed point, by a measure; this measure is either half the distance or $3/4^{th}$ of the distance, etc which gives us some idea as to what is happening in the image. In difference based IFS, we are always going closer to a fixed point by a measure. If we move away from the fixed point, the image will not converge and a random set of points are generated. The points also move outside the screen.

To convert an IFS to difference based IFS, we obtain an invariable point for every transformation of IFS i.e., next point x' = x and y' = y. Consider the above three transformations in equation 2. We notice that the invariant points are (300,300), (0,300) and (300, 0) for the three transforms respectively. If we apply transform to these points, they do not change. Hence, they are called invariants or fixed points.

Given an invariant point, we develop IFS like equations as follows:

$x = x + (\text{invariant}\ (x) - x)*a + b$

$y = y + (\text{invariant}\ (y) - y)*c + d$ (Equation 2)

Here, we are computing the new point as a linear function of distance between the current point and the invariant point. A constant value can be added resulting in translation.

The new set of transforms for equation 2 will be

1. $x = x + (300 - x) * 0.5 + 0$

   $y = y + (300 - x) * 0.5 + 0$
2. $x = x + (0 - x) * 0.5 + 0$

   $y = y + (300 - x) * 0.5 + 0$
3. $x = x + (300 - x) * 0.5 + 0$

   $y = y + (0 - x) * 0.5 + 0$

We use the same method as in IFS to generate the set of points in the image. We select an arbitrary point (x, y) and select one of the three affine transformations in random, apply it to the point (x, y) we get a new point (x', y'). The above procedure is repeated on the point (x', y') and, we get a new point (x", y") and so on. These points are plotted for some fixed number of iterations, say 100000. We get figure 1 as a result of the equations shown above. The two techniques i.e., IFS and difference based IFS are equivalent and are inter-convertible.

2.3 IFS variant with union

The following image is the average result of addition between two fractals. We get non-linear fractals with combination of fractals. We can combine 2 or 3 independent fractals to get combination fractals. The fractals have to move in a similar way, that is, each fractal should follow the same sequence of transforms. This implies that the random numbers have to be generated in the same sequence for the two fractals. A midpoint of the two fractal points can be plotted or any point in between or on the extended line between the two points can be plotted to get a fractal.

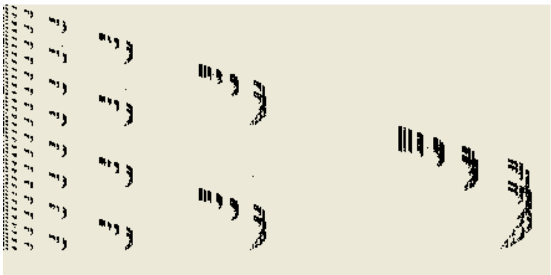

Figure 2: Image Generated by Union Operation

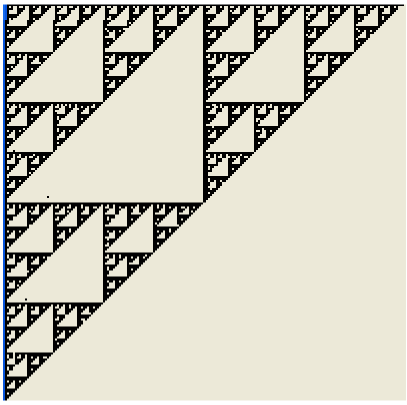

Figure 3. :First fractal

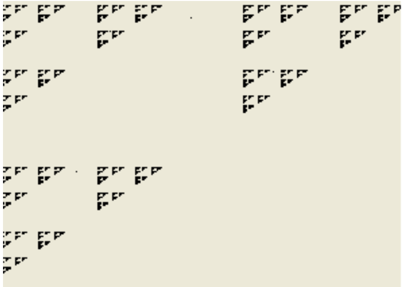

Figure 4:second fractal

Example pseudo code is illustrated below.

```
1.  Array points[6][2] ={{0,0}, {0,200}, {200,0},
    {0,300},{0,500},{300,300}};
2.  x,y,x1,x2=0
3.  Loop 100000 times the following code till line 14.
4.  Choose a random integer number between 0 and 2
    into variable ran.
5.  Variable p is assigned difference of points [ran][0]
    and x. All three transforms are similar.
6.  x = x+p*0.5
7.  x1 = x1+ (points [ran+3] [0] -x1)*0.6 // second
    fractal. All three transforms are similar.
8.  Varible q is assigned difference of points [ran][1]
    and y. All three transforms are similar.
9.  y = y+ q*0.5;
10. y1 = y1+ (points [ran+3] [1] -y1)*0.6// second
    fractal. All three transforms are similar.
```

```
11.  if random number ran is 2 run upto line 13
12. x = (x -x1)/2.0; //  if the random number ran is 2 add
    a function of the second fractal
13. y = (y - y1)/2.0;// running from points 3 to 5
14. Plot the point (x,y)
```

Psuedocode 1: union of two fractals

2.4 Difference between two fractals using IFS variant

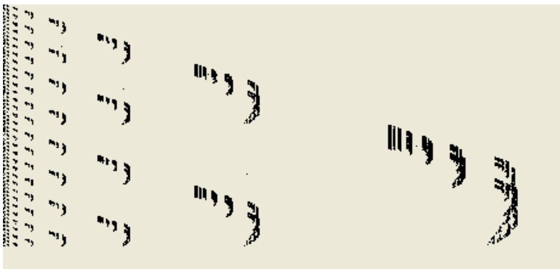

Figure 5: Difference between two fractals

```
1.  Array points[6][2] ={{0,0}, {0,200}, {200,0},
    {0,300},{0,500},{300,300}};
2.  x,y,x1,y1 =0
3.  Loop 100000 times the following code till line 14.
4.  Choose a random integer number between 0 and 2
    into variable ran.
5.  Varible p is assigned difference of points [ran][0]
    and x. All three transforms are similar.
6.  x = x+p*0.5
7.  x1 = x1+ (points [ran+3][0] -x1)*0.6
8.  Varible q is assigned difference of points [ran][1]
    and y. All three transforms are similar.
9.  y = y+ q*0.5;
10. y1 = y1+ (points [ran+3][1] -y1)*0.6;
11.  if random number ran is 2 run upto line 13
12. x = (x –x1) + 400 //  if random number ran is 2
    subtract a function of the second fractal
13. y = (y –y1) + 400 // running from points 3 to 5
14. Plot the point (x,y)
```

Pseudo code 2: difference between two fractals

The image for difference between two fractals is depicted in Figure 5. There is not much variation with respect to union of two fractals.

3. APPLICATIONS

We need not add the generated points for all the fixed points. For example, if fractal A has 4 fixed points and fractal B also has 4 fixed points, the first fractal can be plotted as it is with only the third point being combined to get a combinational fractal.

We can selectively combine two fractals. Figure 6 represents first fractal with four transforms and figure 7 represents the second fractal with four transforms. The second transform of the second fractal is merged with the second transform of the first fractal resulting in figure 8. We see that lower left quadrant is altered. We have to be move the random numbers in the same sequence in both the fractals. Otherwise the image will not converge.

We can also selectively add the contents of the second fractal by inserting a condition that the addition would take place only if the previous transform visited is 1. The result is depicted in figure 8. Similarly a union of figure 6 and figure 7 is depicted in figure 9 whrere the second point of figure 7 is combined only if the previous point is 1.

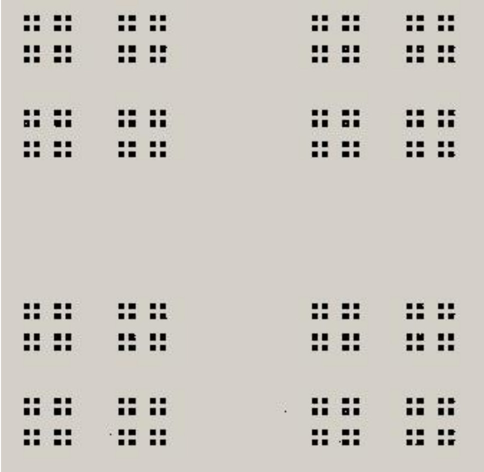

Figure 6: Fractal 1

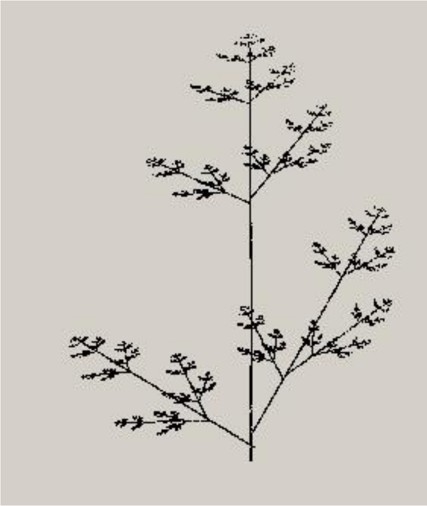

Figure 7: Fractal 2

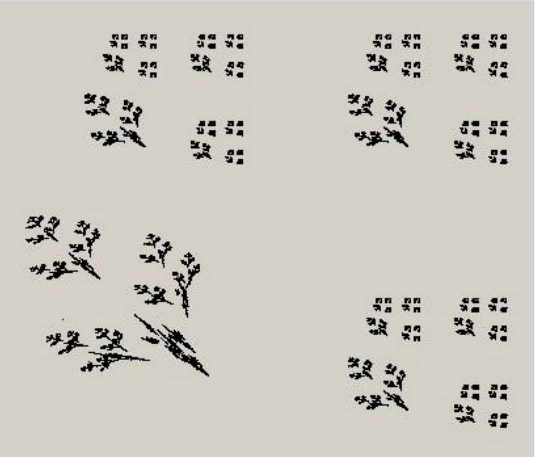

Figure 8:Union with one transform combined

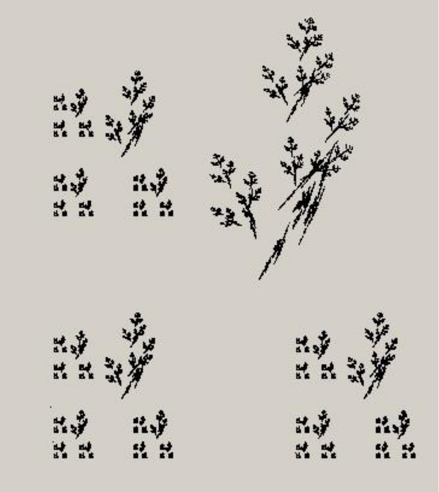

Figure 9: Union with transform 2 if previous point is 1

4. RESULTS AND DISCUSSION

We see that good looking fractals can be generated with union or difference of two already generated fractals. When we move towards any one of the fractal we get a fractal similar to the closer fractal in the range. The line between two original fractals will result in union or difference of two fractals. If we draw the fractal at the midpoint between the two fractals it will result in addition of two fractals. We find interesting figures with respect to the two original fractals when union or difference operations are done on them. One can selectively add one fractal to the other by combining selective transforms. Fractals can also be combined based on the previous sequence of random numbers generated thus fine tuning the addition. The union and subtraction of two fractals preserve the fractal nature of images and the resulting nature of the image is similar to the original fractals. The number of transforms need not be the same in the two fractals.

5. CONCLUSIONS

Good looking fractals can be generated with union or difference of two already generated fractals. Questions were raised as to what is the meaning of union of two different fractals etc. We can combine two independent fractals and induce parts of one generating fractal to another. A function of one of the transforms of a fractal can be added to another generating fractal. Both the fractals should move in the same way with respect to the random numbers generated. If one fractal moves in one sequence and the other fractal moves in another sequence, a fractal is not generated. The number of fixed points (transforms) need not be the same in the two fractals. Similarly, if we plot a point in between the two generating fractals, we get fractals. A point intermediate to the two generating fractals can be plotted and this leads to a fractal. The points can also be plotted beyond the boundaries of the two generating fractals and the resultant is a fractal. When we move towards one of the fractals we get a fractal similar to the closer fractal in the range. If we draw the fractal at the midpoint between the two fractals it will result in combination of two fractals. Components of only certain transforms can be changed by adding a part of the second transform only when a certian random number is generated. This results in selective finetuning of an existing fractal.

6. REFERENCES

[1] Frederic Raynal, Evelyne, Lutton, Pierre Collet, Manipulation of Non-Linear IFS Attractors Using Genetic Programming (1999), Proceedings of the Congress on Evolutionary Computation

[2] Dinesh B Rao, Deepak Rao B and U C Niranjan. Article: Difference based Non-Linear Fractals using IFS. *International Journal of Computer Applications* 80(13):38-42, October 2013. Published by Foundation of Computer Science, New York, USA

[3] Barnsley, Fractals everywhere, Academic Press 1988.

[4] Stephen Demko, Laurie Hodges and Bruce Naylor , Construction of Fractal Objects with Iterated Function Systems, ACM SIGGRAPH Computer Graphics, 1985.

[5] Lu Ning, Fractal Imaging, academic Press 1997.