The Effect of Multiple Rotations on a Unified System of Affine Transformations with related Trigonometric Coefficients

T. Gangopadhyay XLRI C.H.Area(E), Jamshedpur, India

ABSTRACT

In IFS fractals generated by affine transformations with arbitrary coefficients often there is a lot of chaotic noise. In the present paper the author studies the effect of multiple rotations on affine transformations with related trigonometric coefficients in terms of the IFS fractals generated by them. In the process a unified set of equations for generating both the Highway Dragon and the C curve have been developed. The effect of multiple rotations is to lend additional depth to the generated fractal as well as create new fractal designs..

General Terms

Fractal, Algorithm, Turbo C++, Program. related trigonometric coefficients

Keywords

affine, IFS, rotation, trigonometric.

1. INTRODUCTION

In earlier papers the author has studied the effect of the average of two special transformations on standard escapetime fractals (Gangopadhyay[3]) as well as the effect of related trigonometric coefficients on affine transformations in IFS by terms of the fractals generated them(Gangopadhyay[4]). In the present paper, the effect of multiple rotations on affine transformations with trigonometric coefficients that generate IFS fractals has been studied. As mentioned in the earlier paper (Gangopadhyay[4]).the related trigonometric coefficients in affine transformations considerably reduce noise in the generated IFS fractal. The effect of multiple rotations is to lend additional depth to the generated fractal as well as create new fractal designs. In a sense, the method used is a generalization of the process used for Highway Dragon curves (Bulaevsky[2], Gardner[5]) and the C curve(Levy[6]). First a set of affine equations are produced that unify both the Dragon curve and the C curve equations. Then these are further generalized through related generalized trigonometric coefficients. Next multiple rotations are applied to produce more well-rounded and new fractal designs. These are primarily the distinctive features of this paper.

2. THE ALGORITHM

In iterated function systems Michael Barnsley [1] used affine transformations repeatedly on a starting point to produce fractal shapes. Bulaevsky[2] used this system to generate the Highway Dragon curve. He used two affine transformations:

1. $x = (1/\sqrt{2}) \cos(\pi/4) x - (1/\sqrt{2}) \sin(\pi/4) y$,

 $y=(1/\sqrt{2})*\sin(\pi/4)*x+(1/\sqrt{2})*\cos(\pi/4)*y.$

2. $x = (1/\sqrt{2})*\cos(3*\pi/4)*x - (1/\sqrt{2})*\sin(3*\pi/4)*y + 1,$ $y = (1/\sqrt{2})*\sin(3*\pi/4)*x + (1/\sqrt{2})*\cos(3*\pi/4)*y$.

Similarly Levy[] used the following affine equations to generate the C curve :

- 1. $x = (1/\sqrt{2}) \cos(\pi/4) x (1/\sqrt{2}) \sin(\pi/4) y,$ $y = (1/\sqrt{2}) \sin(\pi/4) x + (1/\sqrt{2}) \cos(\pi/4) y.$
- 2. $x = (1/\sqrt{2}) \cos(3\pi/4) + (1/\sqrt{2}) \sin(3\pi/4) + 1,$ $y = (1/\sqrt{2}) \sin(3\pi/4) + (1/\sqrt{2}) \cos(3\pi/4) + y.$

The author first unifies these two sets of equations through the use of a variable 'sig' which takes the value +1 and -1 respectively to generate the Dragon and the C curves. The unifying set of equations are given below:

- 1. $x = (1/\sqrt{2}) \cos(\pi/4) x (1/\sqrt{2}) \sin(\pi/4) y$, $y = (1/\sqrt{2}) \sin(\pi/4) x + (1/\sqrt{2}) \cos(\pi/4) y$.
- 2. $x = (1/\sqrt{2}) \cos(3\pi/4) x (1/\sqrt{2}) \sin(3\pi/4) y + 1,$ $y = (1/\sqrt{2}) \sin(3\pi/4) x + (1/\sqrt{2}) \sin(3\pi/4) y.$

Next, the author uses angles other than $\pi/4$ and $3*\pi/4$ and use multipliers other than $1/\sqrt{2}$. Then the number of transformations are increased by rotating the affine transformations through multiple angles thereby producing interesting and original effects..

Each pixel is then colored by the number of times it is visited. The final output is contrasted against a dark blue background generated by very weak Perlin noise[7].

In the next section the author submit a programming code in Turbo C++ that captures the algorithm and generate some sample output.

3. THE CODE

The code uses a function tgatt which is declared first. The function has seven arguments – scale, atang which gives the value of the generalized angle in the original affine equations, angg which gives the value of the rotating angle, sig which takes values +1 or -1, t which gives the number of final rotations, a which decides the number of rotations for each transformation and m which is a multiplier for the x and y terms only. Thus if t =2 and a =1, then each transformation is allotted only a single rotation, the angle of rotation being given by the value of the parameter angg. Both the tgatt function and the code are given below:.

void tgatt(int scale,int atang,int angg,int sig,int t,int a,float m)

International Journal of Computer Applications (0975 – 8887) Volume 139 – No.2, April 2016

{randomize();

```
int j,k; double x,y,px; cmplx z=cmplx(.1,.5);
```

float ang,ang1=atang*3.14/180.,

ang2=(180-atang)*3.14/180.;

cmplx o=cmplx(1,0),ii=cmplx(0,1);

for(long i=0;i<2700000;i++)

{ px=x;

j=random(999)%1000;

k=j%t;

if (k<a) // ...(1)

ang=(angg+k*360/a)*3.14/180.;

```
x=real(z),y=imag(z);px=x;
```

x=x*cos(ang1)-y*sin(ang1);

```
y=px*sin(ang1)+y*cos(ang1);
```

z=cmplx(x,y);;

z=z*cmplx(sin(ang1)*m,0);

x=real(z), y=imag(z); px=x;

x=x*cos(ang)-y*sin(ang);

y=px*sin(ang)+y*cos(ang);

z=cmplx(x,y); }

else

```
ang=(angg+(k-a)*360/(t-a))*3.14/180.;
```

x=real(z),y=imag(z);px=x;

x=x*cos(ang2)-y*sin(ang2)*sig;

y=px*sin(ang2)+y*cos(ang2)*sig;

z=cmplx(x,y);

```
z=z*cmplx(cos(ang1)*m,0); z=z+o;
```

```
x=real(z), y=imag(z); px=x;
```

x=x*cos(ang)-y*sin(ang);

y=px*sin(ang)+y*cos(ang);

z=cmplx(x,y); }

int x1=scale*x+512,y1=scale*y+400;

if(x1>0&&x1<1024&&y1>0&&y1<800)

putpixel(x1,y1,getpixel(x1,y1)+1);

}}

void main()

{

tgatt(200,45,0,1,2,1,1);

getch();

closegraph();

}

The output of the sample code is illustrated in Figure 1. Here scale =200, atang = 45, angg=0, sig = 1, t=2 and a=1.. The resulting output is the Highway dragon illustrated in Figure 1.



Fig 1 : Output of the sample code

By changing the value of sig to -1 and other parameters unchanged in the function tgatt one can generate he Levy C curve as illustrated in Figure 2.



Fig. 2 : Output of sample code with sig = -1

Further, by changing the value of atang to 30 one obtains figures 3 and 4 respectively for sig = 1 and -1.



Fig. 3 : Output of sample code with atang = 30 and sig = 1



Fig. 4 : Output of sample code with at ang = 30 and sig = -1.

Finally, by changing the value of atang to 135 and setting sig to -1, one obtains figure 5.



Fig. 5 : Output of sample code with atang = 135 and sig = -1.

4. EFFECT OF A SINGLE ROTATION

To show the effect of a single rotation one changes the parameter angg, the rotation angle which was earlier set to zero. Thus, by changing angg to 10 in the Highway Dragon(Fig. 1.) one obtains a more connected dragon where the the depth of colour also happens to be visible. The result is shown in Figure 6.



Fig 6 : Output of the sample code with angg = 10.

Again, by changing angg to 75 in the Highway Dragon(Fig. 1.) one obtains a Julia fractalesque shape sown in Figure 7.



Fig 7 : Output of the sample code with angg = 75.

Again, similarly,by changing angg to 80,110,280 respectively in figure 2, one obtains the output shown in Figures 8, 9 and 10. Figure 10 depicts a more well-rounded C-curve than the one shown in figure 2.

International Journal of Computer Applications (0975 – 8887) Volume 139 – No.2, April 2016



Fig 8 : Figure 2 with angg = 80.



Fig 9 : Figure 2 with angg = 110.



Fig 10. Figure 2 with angg = 280.

5. EFFECT OF MULTIPLE ROTATIONS

To show the effect of multiple rotations, one changes parameters t, m and a in the function tgatt/ For instance, if t is changed to 3 and m is set to .85, then the output of sample code is as shown in Figure 11.



Fig 11. Output of the sample code with t=3 and m=.85.

Similarly, if t is changed to 3, m is set to .83 and a is set to 2, then the output of sample code is as shown in Figure 12.



Fig 12. Output of the sample code with t=3, a=2 and m=.83.

Again, if t is changed to 4, m is set to .8 and a is set to 2, then the output of sample code is as shown in Figure 13, a more well-rounded figure compared to Figure 11.



Fig 13. Output of the sample code with t=4, a=2 and m=.8.

Finally, Figure 14 shows the output of the sample code with t=4, a=1 and m=.61.



Fig 14. Output of the sample code with t=4, a=1 and m=.61.

The change in the output of various figures for different values of the parameters is depicted in the next five figures.



Fig 15. Changes in Figure 2 with t=3 a=2 and m=.8.



Fig 16. Changes in Figure 2 with t=4, a=1 and m=.61.



Fig 17. Changes in Figure 5 with t=4 a=1 and m=.7.



Fig 18. Changes in Figure 1 with t=5 a=2 and m=.63



Fig 19. Changes in Figure 1 with t=6, a=1 and m=.58.

Finally, in figure 20, one shows the output of multiple rotation for a large t(=6) when angg is not 0.



Fig 20. Output with with atang=35, angg=10, sig=-1, t=6, a=1 and m=.56.

6. CONCLUSION

In this paper, the second transformation uses 180 degrees as a pivotal angle. This could also be altered, creating a new parameter. Similarly rotations could also be used successfully on Mobius fractals. These and other modifications would be explored in future work.

7. ACKNOWLEDGMENTS

The author wishes to acknowledge his debt to the referee(s) for their constructive suggestions and encouragement

8. REFERENCES

- [1] Barnsley, M. 1983 Fractals Everywhere, Academic Press.
- [2] Bulaevsky, J. "The Dragon Curve or Jurassic Park fractal,

http://ejad.best.vwh.net/java/fractals/jurasic.shtml.

- [3] Gangopadhyay, T. 2012 On generating skyscapes through escape-time fractals, International journal of Computer Applications 43(2012)17-19.
- [4] Gangopadhyay, T. 2012 IFS Fractals generated by affine transformation with trigonometric coefficients and their transformations, International journal of Computer Applications 53(2012)29-32,.
- [5] Gardner, M. Mathematical Magic Show: More Puzzles, Games, Diversions, Illusions and Other Mathematical Sleight-of-Mind from Scientific American. New York: Vintage, pp. 207-209 and 215-220, 1978.
- [6] Paul Lévy, Plane or Space Curves and Surfaces Consisting of Parts Similar to the Whole (1938), reprinted in Classics on Fractals Gerald A. Edgar ed. (1993) Addison-Wesley Publishing ISBN 0-201-58701-7.
- [7] Perlin, K. 'Coherent noise function', original source code, http://mrl.nyu.edu/~perlin/doc/oscar.html#noise.