Multicolor Symmetrical Fractal Pattern Generator

C E Widodo
Department of Physics, Faculty of Science and Mathematics, Diponegoro University
Jl. Prof. Soedharto, SH, Tembalang, Semarang 50275, Indonesia
E-mail: catur.ediwidodo@gmail.com

Abstract. A fractal is a geometric figure that combines the several characteristics among others: its parts have the same form as the whole, fragmented, and formation by iteration. The concept of fractals has been spread over all fields of sciences, technology, and art. The properties of fractal sets are necessary to be investigated and favorable for the practical applications. This paper aims to provide an algorithm for creating multicolor symmetrical fractal pattern generator. Generator algorithm consists of base, iteration, coloration, and duplication. To help the reader better understand the algorithm, we will present some script using Matlab. We describe a mathematically based algorithm that can fill a spatial region with a sequence of randomly placed which may be transformed copies of one motif or several motifs. This flexible algorithm can be used to produce a variety of aesthetically pleasing fractal patterns, of which we show some examples. We hope that the existence of fractals can show that mathematics is not a dry and flat subject, but it is a beautiful subject and can produce works that have a high degree of art and intellectual value.

1. Introduction
Various types of fractals were originally studied as mathematical objects. Fractal geometry is a branch of mathematics that studies fractal properties and behavior. Fractals can help explain many situations that are difficult to describe using classical geometry and have been applied to science, technology, and the art of computer work. In the past fractal conceptual ideas emerged when traditional definitions of Euclidean geometry and calculus failed to analyze these difficult curve objects

The term Fractal was introduced in the field of mathematics by Mandelbrot, which defines what is called Fractal Geometry [1, 2, 3]. This new geometry can describe the irregular but beautiful shapes found in nature. Fractal Geometry is based on the use of the repetition principle of geometric shapes so that an object can reproduce itself when experiencing magnification. Mathematically, fractals have regions that are defined by boundary curves that are more than 1 dimensional, so that the direction of the line can change infinitely many times. With Fractal Geometry, art can find new creativity. Even with fractals, we can find new patterns that were previously unimaginable. With fractals, we don't need to create shapes, we just have to do the shape selection, because fractals will give an infinite variety of shapes.

After computer visualization was applied to fractal geometry, powerful visual arguments can be presented to show that fractal geometry connects many fields of mathematics and science, larger and wider than the previous ones. The fields connected by fractal geometry are mainly image processing [4], medical imaging [5, 6, 7, 8, 9, 10] and ‘batik’ art [11, 12, 13], fractal geometry has also been used for data compression and model complex geological and organic systems, such as tree growth and development of river valleys.
Fractals can be generated only by repeating the function again and again [3]. For example, the famous Mandelbrot set can be made by the following function:

\[ f(z) = z^2 + c. \]

Suppose that the values \( c = 4 \) and \( z_0 = 4 \). By placing the \( z_0 \) value into the function as the input and the iteration function, again and again, we get a larger and larger value as given below:

\[
\begin{align*}
    z_1 &= 42 + 4 = 20 \\
    z_2 &= 202 + 4 = 404 \\
    z_3 &= 4042 + 4 = 163220 \\
    z_4 &= 1632202 + 4 = 26640768404
\end{align*}
\]

Thus, different points can have very different behaviors under iterations by the same function. If we have a two-dimensional plane with a continuous gradation value, then we will get the value of the function that varies continuously. If we plot the value of the function as a color in the appropriate coordinates, we will get beautiful colorful patterns.

2. Algorithm

The computer generation of Fractal is mainly based upon a filling along the screen rows and columns. For each screen point \( P \), one defines an ordered and unique pair of coordinates \((x, y)\), associated with the complex number \( p = x + iy \). Let \( f \) be a given map, so the goal is to set a color, related to the value of the \( n \)-fold iterate \( f^{(n)}(p) \) at \( P \). P is a fractal formula, which in this study was given by:

\[
P = P^P + c \tag{1}
\]

Where \( p \) is complex number and \( c \) is constant. This is the Multicolor Symmetrical Fractal Pattern Generator algorithm which is done in three phases:

1. Make a base
2. Running the iteration
3. Fill in the color
4. Duplicate

This is just the algorithm that must be carried out. For more details, let's follow the steps directly using the Matlab code.

The first phase is to make a base. The base is a two-dimensional matrix with continuous values, that can be linear, sine, quadratic and others. To get a symmetrical fractal, the value is also symmetrical. This is a symmetrical base example in the Matlab code:

\[
x=\text{linspace}(-1,1,200);
y=\text{linspace}(-1,1,200);
\]

\[
[X,Y]=\text{meshgrid}(x,y);
\]

The second phase is to run the iteration. On this occasion, we will give the function \( f^{(0)} = z^2 + c \) because this function has got good results since the first iteration. If the number of iterations and \( c \) is given a random value, then each execution will produce a different image. This is an example of iteration in the Matlab code.

\[
\text{iter}=4;
\text{for } u=1:\text{iter};
\text{    } Z=Z.^Z+c;
\text{end};
\]

The third phase is filling colors. We set the number of iterations for a given point as the red, green and blue components of its color. In Matlab, we can make a color map matrix as we wish. We can specify red, green and blue components randomly. This is an example of a color map with 16 random colors.
for i=1:16;
    kolor(i,1)=rand;
    kolor(i,2)=rand;
    kolor(i,3)=rand;
end;

If you want to get a color map with the base color, you must give value 0 or 1 in each color component. The algorithm is:

for i=1:16;
    kolor(i,1)=round (rand);
    kolor(i,2)=round (rand);
    kolor(i,3)=round (rand);
end;

The last phase is duplication. Duplication can be done inside the program or outside the program. To get more degrees of symmetry, duplication can be done with both rotation and reflection. Duplication outside the program can be done using an image editor.

As promised earlier, the following is an example of a Multicolor Symmetrical Fractal Pattern Generator in the Matlab code. This program each execution will produce a different pattern image. With this program, we can produce infinite fractal patterns.

clear;
m=700;
n=500;
k=rand+0.5;
x=linspace(-k,k,m);
y=linspace(-k,k,n);
[X,Y]=meshgrid(x,y);
Z=X+i*Y;
c=rand;
iter=rand*9+1;
for u=1:iter;
    Z=Z.^Z+c;
end;
W=exp(-abs(Z));
W=imrotate(W,90);
makskolor=256;
for i=1:makskolor;
    kolor(i,1)=rand;
    kolor(i,2)=rand;
    kolor(i,3)=rand;
end;
colormap (kolor);
pcolor(W);
shading flat;
axis('square','equal','off');

3. Experimental Results
In this work, programming has been carried out with different base and iteration formulas. By using random values for constant values, counting iterations and color map values, each execution produces a different fractal pattern. In this section, some examples of fractal patterns will be given and will be
discussed about one of the interesting things we found namely entropy. Some examples of fractal patterns that we find interesting can be seen in Figures 1, 2, 3 and 4.

Figure 1. Reflecting symmetry pattern.

Figure 2. Rotational symmetry pattern.
Let's discuss these four images. Figure 1 has reflection symmetry. Mirroring is obtained from the base:

\[ x = \text{linspace}(-1,1,200) \]

Which is the basis of symmetry. The colors used are RGB and CYM colors obtained from the basic color map. Figure 2 has rotational symmetry. The pattern formed can be duplicated with a 90-degree rotation, 180 degrees, and 270 degrees. This pattern is very good as a veil pattern. If the pattern formed is duplicated to form a matrix, then skewed, it will get the pattern of 'batik rejeng' as shown in Figure 4. Batik is one ethnic clothing in Java, Indonesia. With certain random values, complex patterns are sometimes obtained as shown in Figure 4. With a light yellow, dark yellow, brown, pink, dark red green and dark gray pattern

In physics, entropy is a measure of irregularity. In image processing, entropy can be determined based on the number of gray levels [14]. Visually, high entropy images have complicated patterns, so there is no focus of interest. In this work, it can be seen that iteration can reduce entropy. The more iterations, the more entropy goes down. The pattern becomes more organized, so that beautiful objects appear. But if the iteration continues, all items will disappear so that entropy becomes zero. There is nothing left there except space. Examples of fractal patterns with one to four iterations can be seen in Figure 5.
Figure 5. Pattern changes as a result of iteration

In (a) we only see curved lines of different colors. In (b) and (c) the arches begin to move towards a certain destination. Finally, in (d) several beautiful objects are formed. From the first to the fourth iteration, entropy drops from 7.78 then 7.06 then 6.12 and then 4.91.

4. Conclusions
Fractal Geometry discovery has made it possible to find complex but pleasant geometric shapes. Fractals allow you to better understand geometric figures with regular structures — fractal Geometry, playing with unlimited and very complicated simulations through digital operations, which gradually includes images involving all forms of science and art.

References
[1] Mandelbrot, B.B. 1977. *The fractal geometry of nature*, W.H., Freeman and Company, New York.
[2] Garg, A., Agrawal, A., and Negi, A. 2014. *International Journal of Computer Applications*. 85 15
[3] Garg, A., Agrawal, A., and Negi, A. 2014. *International Journal of Computer Applications*. 86 4.
[4] Chaudhuri, B.B. and Sarkar, N. 1995. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17 1
[5] Losa, G.A. 2009. *Biology Forum*. 102 29-60
[6] Li, X., Polygiannakis, J., Kapiris P., Peratzakis, A., Eftaxias, K. and Yao, X, 2005. *Journal of Neural Engineering*. 2 11–16
[7] Kisan, S., Mishra, S. and Rout, S.B. 2017. *International Research Journal of Engineering and Technology*. 4 5
[8] González a, J.J. and Peredab, E. 2004. *Current Vascular Pharmacology*. 2 149-162
[9] El-zbieta, G., Rybaczuk , M. and Kezdzia, A. 2005. *Chaos, Solitons and Fractals*. 24 707–715
[10] Moore, B. and Dasi, L.P. 2013. *Solitons and Fractals*. 57 19–23
[11] Oparinde, S.S. 2012. *International Refereed Research Journal*. 3
[12] Hengky, S. H. 2014. *Business Management and Strategy*. 5 2
[13] Riani, A.L., Sarungu, J.J. and Margana, 2016. *Strategic Management Quarterly*. 4 4
[14] Dougherty, G. 2009. Digital image processing for medical applications, Cambridge University Press, Cambridge, New York.