Butterfly Effect Fractal

Yin-Wei Chang and Fay Huang

Institute of Computer Science and Information Engineering
National Ilan University, Taiwan
fay@niu.edu.tw

Abstract. In this paper, a new concept of integration of fractal and the butterfly effect is proposed and implemented. A new fractal program was designed and developed to perform such integration. Among many existing fractal and chaos software programs, none of them allow us to achieve the resulting patterns demonstrated in this paper. Moreover, it is the first time that a fractal program provides functional concepts of overlapping results in 3D space and sequential transformations, which allow us to generate a wider variety of patterns. Therefore, potentially an artist can use this program to create 2D digital artworks.

Keywords: Butterfly effect, Lorenz attractor, fractal and chaos, mathematical art.

1 Introduction

In recent years there has been an increasing interest in the fields of digital art and mathematical art while computing power increases rapidly. Computers offer a wide range of creative possibilities for digital artists to explore new ideas and generate or deliver their artworks. Combining fractal and chaos theory is a well-known approach which enables to create amazing artistic patterns [3,7]. In this paper, we elaborate particularly on the potential of *The Butterfly Effect* in chaos theory for creating artistic patterns and animations.

There are software programs that support various fractal formulas and a wide variety of color filters for creating fractal art images [4]. Eye-catching fractal images are easily accessible through web-based fractal art galleries. Fractal techniques have became a powerful tool not only for digital art design but also for architectural design [1,6]. Moreover, fractal formulas have been applied to the creation of simulated natural landscapes and plants in computer graphics [8].

Among all the studies of fractal and chaos theories, to the best of our knowledge, no one has reported so far a mixed model of *The Butterfly Effect* fractal. The integration of The Butterfly Effect formulas into fractal art is a newly proposed concept, which provides a novel variety of possibilities in terms of forms and shapes in fractal art creation. Figure 1 illustrates the potential of our developed program, in which a typical fractal and a typical butterfly effect pattern are shown on the left and middle, respectively, as well as a resulting artistic pattern generated by our program on the right.
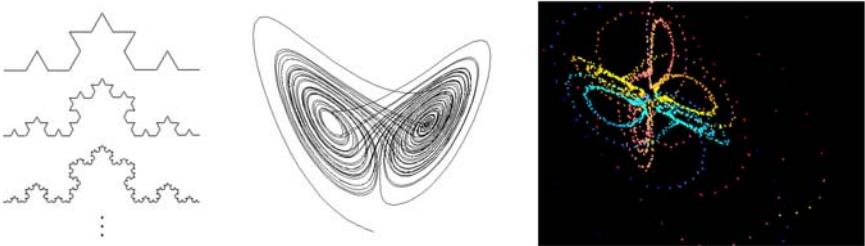
Fig. 1. Left: a typical fractal structure. Middle: a typical Lorenz attractor. Right: a 2D artwork generated by our program.

2 The Butterfly Effect

The Butterfly Effect is not only a fascinating mathematical theory, it also happens to be the name of a well-known movie series. Perhaps more people know about the stories of those movies than about the history and essence of this theory. We provide some useful mathematical background and a brief history of The Butterfly Effect (partially recalled from [2]) in this section.

2.1 Lorenz Equations and Lorenz Attractor

Edward Lorenz was a Mathematician and Meteorologist at the Massachusetts Institute of Technology who was interested in weather prediction. He constructed a mathematical model of the weather, defined by a set of twelve differential equations that represented changes in temperature, pressure, wind velocity, and so forth. On a particular winter day in 1961, Lorenz wanted to re-examine a sequence of data coming from this weather model. He discovered that his model exhibits the phenomenon known as "sensitive dependence on initial conditions". This is sometimes referred to as the *Butterfly Effect* (i.e., a butterfly flapping its wings in South America may affect the weather in Central Park, London). The data obtained by two different runs diverged dramatically due to rounding-off errors; results differed in more than three decimal places. This lead to the question, why does a set of completely deterministic equations exhibit this behavior? It is due to the nature of the equations themselves which were nonlinear equations. Nonlinear systems are central to chaos theory and often exhibit fantastically complex and chaotic behavior.

Lorenz first reported his weather model and the discovery in *Deterministic Nonperiodic Flow* [5], which was a journal paper published in 1963 and had great influence on many subsequent related studies. His work is important even today because he had an insight into the essence of chaos, and his work settled in today's chaos theory. Later, Lorenz decided to look for complex behavior in an even simpler set of equations, and was led to the phenomenon of rolling fluid convection. He attempted to simplify a few fluid dynamic equations (called the

Navier-Stokes equations) and ended up with a set of three nonlinear equations known as the *Lorenz equations*:

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

where σ is the *Prandtl number* representing the ratio of the fluid viscosity to its thermal conductivity, ρ is the *Rayleigh number* representing the difference in temperature between top and bottom of the system, and β is the ratio of width to height of the box used to hold the system. All σ, ρ, $\beta > 0$, but usually $\sigma = 10$, $\beta = 8/3$ and ρ is varied. The system exhibits chaotic behavior for $\rho = 28$ but displays knotted periodic orbits for other values of ρ.

Although the equations look quite simple, they are nonlinear depending on the products of the variables xz and xy, and an analytic solution is impossible in general. We employed the iterative method that allows solving the system of nonlinear equations numerically [9]. To compute a numerical approximation for the solution, we employ the following iterative equations:

$$x_{n+1} = x_n + 10(y_n - x_n)h$$
$$y_{n+1} = y_n + (-x_n z_n + 28x_n - y_n)h \tag{1}$$
$$z_{n+1} = z_n + (x_n y_n - \frac{8}{3}z_n)h$$

We obtained the 3D plot as shown in Figure 1(middle), which is known as Lorenz attractor, generated by an iterative process as in Equations (1) with initial values $x_0 = 0.0001$, $y_0 = 0.0001$, and $z_0 = 0.000001$. This illustrates the typical butterfly-like pattern when projecting data points onto the xz-plane.

2.2   Integration of Fractal and Lorenz Equations

We launched the concept of fractals for Equations (1). The Lorenz attractor becomes a basic element of a fractal structure. As a result of a two-leveled fractal structure, the initial base Lorenz attractor will grow into smaller version of Lorenz attractors along its path as shown in Figure 2. The equations used to generate such fractal Butterfly Effect pattern are as follows:

$$A_{n+1} = A_n + 10(B_n - A_n)$$
$$B_{n+1} = B_n + (-A_n C_n + 28A_n - B_n)$$
$$C_{n+1} = C_n + (A_n B_n - \frac{8}{3}C_n)$$
$$a_{m+1} = a_m + 10(b_m - a_m)$$
$$b_{m+1} = b_m + (-a_m c_m + 28a_m - b_m) \tag{2}$$
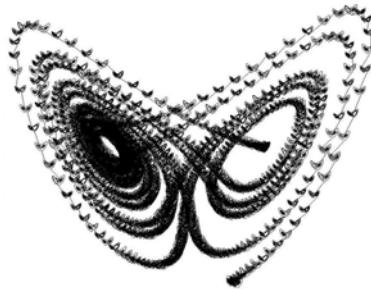
Fig. 2. The fractal *Butterfly Effect*

$$c_{m+1} = c_m + (a_m b_m - \frac{8}{3} c_m)$$
$$x_{nm} = 20A_n + a_m$$
$$y_{nm} = 20B_n + b_m$$
$$z_{nm} = 20C_n + c_m$$

We show only the equations for a two-leveled fractal structure here. Equations or n-leveled fractal structures, with $n > 3$, would involve huge amounts of computations. Since we are interested to achieve real-time calculation, and accelerating computation is not our major objective in this work, our program offers at most three-leveled fractal.

3   Program Design

In this section, we explain the design issues and functions of control buttons provided in our fractal program. Target users of this program are people who do not have mathematical background on chaos theory but are interested in creating fractal art. Thus, there is no formula appearing in the user interface. An artistic pattern can be created purely by pressing buttons, selecting data-points, shape or color, and changing the view point by simple mouse drag.

Functionality provided by our program can be classified into five categories, namely initialization, transformation, viewing, point style, and coloring filters. We explain the purpose of each of these categories in the following:

Initialization. There are three initial values defining the x, y, and z coordinates of the Butterfly Effect's starting position in 3D space. A different starting position leads to different butterfly pattern, which is the core meaning of "Butterfly Effect". Moreover, there are two initial values indicating the complexity of the butterfly pattern, namely *Big loop* and *Small loop*. The Big (Small) loop corresponds to the first (second) level of a fractal structure. The larger the value the more complex the butterfly pattern and thus more computation time.

Transformation. There are three menus provided in this category. The first menu is called *Old Shape*, in which a user may decide whether to keep the current

Spade

$$x_s = x_{nm}$$
$$y_s = y_{nm}^5$$
$$z_s = z_{nm}^3$$

Eye

$$x_h = x_{nm}$$
$$y_h = \frac{\sin(\frac{z_{nm}}{2})\sin^2(6z_{nm})}{3}$$
$$z_h = \sin(\frac{z_{nm}}{2})\cos(6z_{nm})$$

Shell

$$x_s = x_{nm}^2$$
$$y_s = y_{nm}^2$$
$$z_s = z_{nm}^3$$

Hourglass

$$x_h = x_{nm}$$
$$y_h = \sin(\frac{z_{nm}}{2})\sin^2(6z_{nm})$$
$$z_h = \sin(\frac{z_{nm}}{2})\cos^4(6z_{nm})$$

Mask
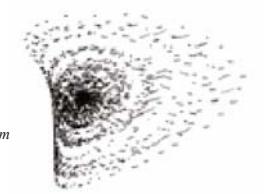
$$x_m = x_{nm}$$
$$y_m = x_{nm}y_{nm}$$
$$z_m = z_{nm}$$

Fly

$$x_f = \frac{x_{nm}}{3}$$
$$y_f = \log(x_{nm}^8 + y_{nm}^9)$$
$$z_f = \frac{z_{nm}}{3}$$

Heart
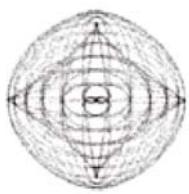
$$x_h = x_{nm}^5$$
$$y_h = y_{nm}$$
$$z_h = z_{nm}^3$$

Sun

$$x_s = x_{nm}$$
$$y_s = \sin(\frac{z_{nm}}{2})\sin(6z_{nm})$$
$$z_s = \sin(\frac{z_{nm}}{2})\cos(6z_{nm})$$

Meteor
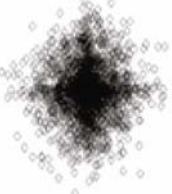
$$x_m = \log(x_{nm})$$
$$y_m = \log(y_{nm}^4)$$
$$z_m = \log(z_{nm})$$

Diamond

$$x_d = x_{nm}$$
$$y_d = x_{nm}\sin(\frac{y_{nm}}{2})\sin(6z_{nm})$$
$$z_d = \frac{6x_{nm}\sin(\frac{z_{nm}}{2})\cos(6z_{nm})}{5}$$

Fig. 3. Predefined transformations and their equations

data or to transform them. If we choose *Keep*, then the current pattern(s) will be kept in the displaying window. In this case, the next generated pattern will overlap the previous one(s) in the window. Figure 5 (middle) is an example of overlapping butterfly patterns generated using different colors and by different looping values. If we choose *Release*, then the last generated pattern will be transformed according to the next transformation command. The performed

Eye->Mask = Spider   Eye->Sun = Explode   Eye->Fly = Candle

Eye->Sun->Shell = Peacock   Sun->Fly = Jellyfish   Meteor->Heart = Fireworks
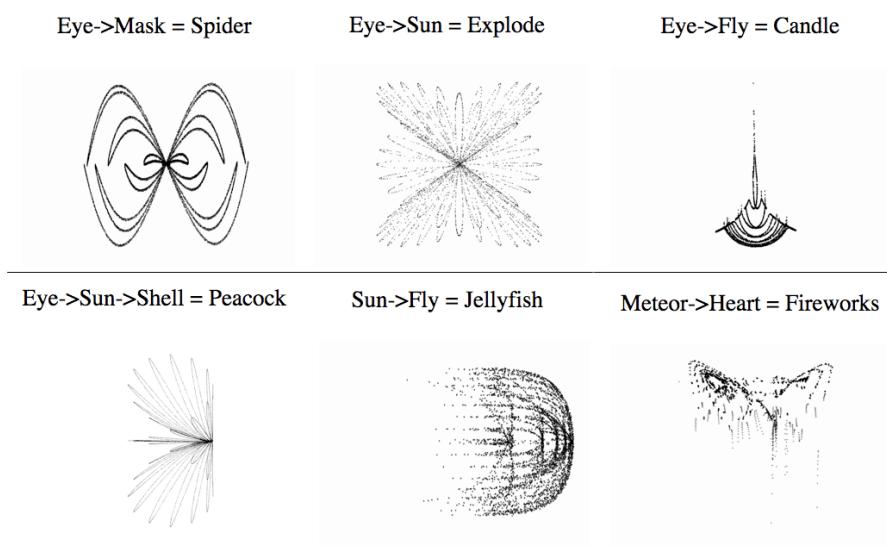
Fig. 4. Patterns generated by different combinations of predefined transformations

transformations are displayed in a text region. The second menu is called *Evolve*, in which there are many predefined transformations. The key to explore a new pattern is to try different combinations of transformations and in different order. The third menu is called *Mirror*, which performs mirror transformation to the current data.

Viewing. This changes user's viewing position and direction. The buttons and menus provided here are quite trivial for computer graphics users. For instance, you can choose the predefined view by menu selection, rotate the data by mouse drag, or continuously rotate by pressing a button. The program is able to perform real-time rendering which changing viewing position and angle, and hence the sequence of renderred images can be recorded and converted to an animation format. This function enables users to design their own "viewing paths" within the 3D data, and offers the potential of creating artistic (abstract) animations.

Point Style. This changes the style (shape, size, color) of the data point.

Coloring Filter. Sometimes only changing the point style is not sufficient to create impressive results. Here, we provide some coloring methods in the *Color Map* menu to create different coloring effects. This category is called *Sphere Controller* because those coloring filters are only applicable to facets, and thus it is necessary to represent the data by some 3D geometric shapes before applying coloring filters. In this first version of the program, we only implemented spheres and six coloring methods. Other coloring filters (or methods) can be easily included in the program.
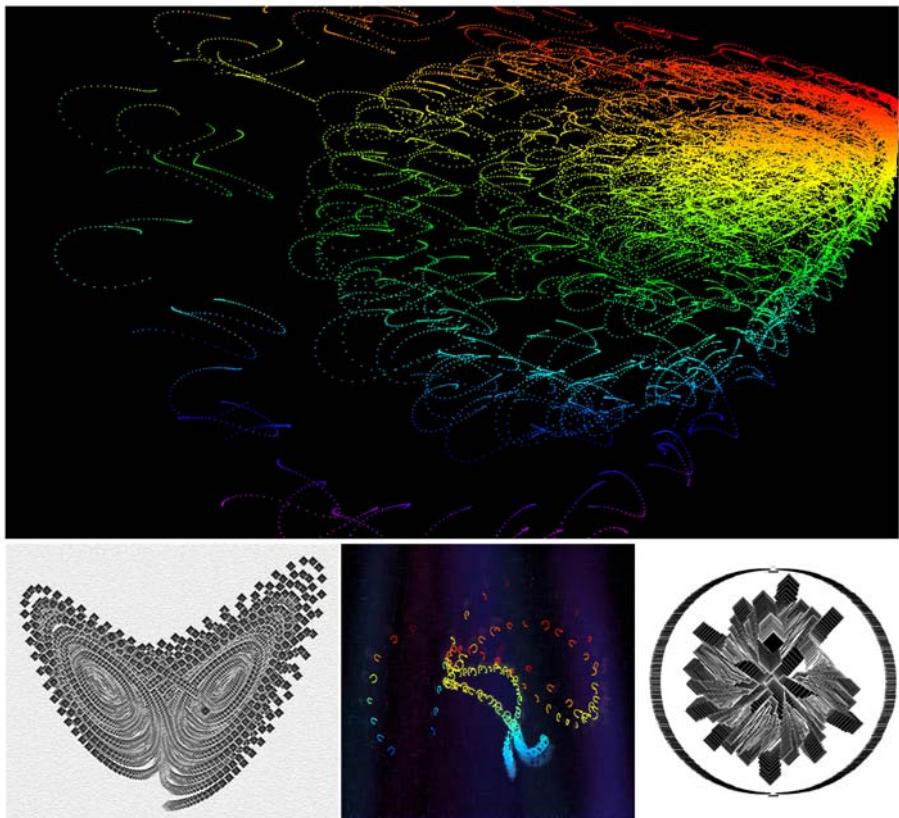
Fig. 5. Generated artworks utilizing the fractal butterfly effect

3.1   Results

Typical patterns and corresponding equations of ten predefined transformations are illustrated in Figure 3. Those are meant to show the structure of the pattern, and are thus displayed in black only. Applying different transformations in different sequential order will result in very different patterns. Some example combinations of transformations are illustrated in Figure 4. Figure 5 illustrates variety of 2D digital artworks generated by our program. In particular, the structures of the images on the bottom row of Figure 5 are generated by our program, with subsequent minor graphical fine-tuning in Adobe's Photoshop.

4   Conclusions

A new fractal art program was developed. The friendly user interface enables users who have no mathematical background to create artistic fractal patterns.

The major contribution is the proposal and implementation of a fractal Butterfly Effect concept; we believe that this is the first program that allows us to generate such an integration. Other contributions include the specification of equations of the predefined transformations and concepts for applying overlapping results and sequential transformations in the fractal program. There are various existing programs for generating various fractals or Lorenz attractors, but none of them provides overlapping and sequential transformation functions.

References

1. Bovill, C.: Fractal Geometry in Architecture and Design. Birkhäuser, Boston (1996)
2. Bradley, L.: Chaos & Fractals (2009),
   `http://www.pha.jhu.edu/~ldb/seminar/index.html`
3. Gleick, J.: Chaos: Making a New Science. Penguin Books (1987)
4. Lee, P.N.: Fractal Links (2009),
   `http://home.att.net/~Paul.N.Lee/Fractal_Software.html`
5. Lorenz, E.N.: Deterministic Nonperiodic Flow. J. Atmospheric Science 20, 130–141 (1963)
6. Ostwald, M.J.: Fractal Architecture: Late Twentieth Century Connections Between Architecture and Fractal Geometry. Nexus Network J. 3(1), 73–84 (2001)
7. Parker, B.: Chaos in the Cosmos: The Stunning Complexity of the Universe. Plenum Press, New York (1996)
8. Peitgen, H., Saupe, D. (eds.): The Science of Fractal Images. Springer, New York (1988)
9. Tucker, W.A.: Rigorous ODE Solver and Smale's 14th Problem. Found. Comp. Math. 2, 53–117 (2002)