Title: Automatic generation of aesthetic patterns with the use of dynamical systems

Author: Krzysztof Gdawiec, Wiesław Kotarski, Agnieszka Lisowska

Citation style: Gdawiec Krzysztof, Kotarski Wiesław, Lisowska Agnieszka. (2011). Automatic generation of aesthetic patterns with the use of dynamical systems. "Lecture Notes in Computer Science" (vol. 6939 (2011), pp. 691-700), doi 10.1007/978-3-642-24031-7_69

[postprint]

Automatic Generation of Aesthetic Patterns with the Use of Dynamical Systems

Krzysztof Gdawiec, Wiesław Kotarski and Agnieszka Lisowska

Institute of Computer Science, University of Silesia, Poland
{kgdawiec, kotarski, alisow}@ux2.math.us.edu.pl

Abstract. The aim of this paper is to present some modifications of the orbits generation algorithm of dynamical systems. The well-known Picard iteration is replaced by the more general one – Krasnosielskij iteration. Instead of one dynamical system, a set of them may be used. The orbits produced during the iteration process can be modified with the help of a probabilistic factor. By the use of aesthetic orbits generation of dynamical systems one can obtain unrepeatable collections of nicely looking patterns. Their geometry can be enriched by the use of the three colouring methods. The results of the paper can inspire graphic designers who may be interested in subtle aesthetic patterns created automatically.

1 Introduction

There are many domains in which aesthetic value plays an important role, i.e. architecture, jewellery design, fashion design, etc. Judging aesthetics is a highly subjective task. Different people have different beauty appreciation principles. In formal considerations the following features of pattern geometry aesthetics are taken into account [12]: golden ratio, symmetries (rotational, logarithmic, spiral, mirror), complexity, compactness, connectivity and fractal dimension. Colours give additional contribution to beauty of patterns. In the paper we do not formally evaluate aesthetics of the generated patterns. We are basing on the opinions of our colleges and students who have seen our patterns. Majority of them said that the patterns were looking nicely and they might stimulate creativity of designers and reduce the number of physical prototypes of patterns.

Usually the most of work during a design stage is carried out by a designer manually. Especially, in the cases in which a graphic design should contain some unique unrepeatable artistic features. Therefore, it is highly useful to develop an automatic method for aesthetic patterns generation. In recent years several approaches to create nicely looking patterns have been described in literature. For example in [8], [12] the methods based on Iterated Function Systems (IFS) and Genetic Algorithms (GA) for jewellery design were proposed. The approach presented in [12] is limited to 2D patterns. Iterated Function Systems and Gumowski-Mira transform were used for fashion design in [7]. An interesting method based on root-finding of polynomials, called polynomiography, was presented in [4]. Polynomiography, patented in the USA in 2005, produces nicely looking highly predictable art patterns.

In the paper we present algorithms for aesthetic patterns generation using one dynamical system or a set of them. The dynamical systems used in our research generate nicely looking aesthetic complex geometric shapes, different from those created with the help of IFS, GA or polynomiography based methods. Additionally, we propose three colouring algorithms to enrich the aesthetic value of the generated shapes.

In Section 2 the basic information about dynamical systems with their examples producing nicely looking orbits are presented. Section 3 describes two algorithms for patterns generation and three further ones for colouring of the obtained patterns. The sample results are presented in Section 4. Finally, in Section 5 some concluding remarks and plans for the future work are given.

2 Dynamical System

Let us start with the definition of a dynamical system [1].

Definition 1. *A* dynamical system *is a transformation $f : X \to X$ on a metric space (X, d).*

Next, define the orbit of a dynamical system [1].

Definition 2. *Let $f : X \to X$ be a dynamical system on a metric space (X, d). The* orbit *of a point $x \in X$ is the sequence $\{x_n\}_{n=0}^{\infty}$, where*

$$x_n = \underbrace{f \circ \ldots \circ f}_{n \ times}(x) = f^{\circ n}(x). \tag{1}$$

For $n > 0$ equation (1) can be written in the following form:

$$x_n = f^{\circ n}(x) = f(f^{\circ(n-1)}(x)) = f(x_{n-1}). \tag{2}$$

Iteration (2) is called Picard iteration and it is used usually to generate the orbit of a given point for any dynamical system.

In the rest of the paper the space $\mathbb{R}^2$ with the Euclidean metric as the metric space (X, d) is used.

Many examples of dynamical systems are known [6], but we are mainly interested in those which produce geometric patterns that can be recognized as aesthetic ones. Now, we present the examples of such dynamical systems:

- Gumowski-Mira transformation (CERN, 1980) [3]

$$\begin{aligned} x_n &= y_{n-1} + \alpha(1 - 0.05y_{n-1}^2)y_{n-1} + g(x_{n-1}), \\ y_n &= -x_{n-1} + g(x_n), \end{aligned} \tag{3}$$

where $g : \mathbb{R} \to \mathbb{R}$ is defined as follows:

$$g(x) = \mu x + \frac{2(1-\mu)x^2}{1+x^2} \tag{4}$$

and $\alpha, \mu \in \mathbb{R}$,

– Martin or Hopalong transformation [5]

$$\begin{aligned} x_n &= y_{n-1} - sgn(x)\sqrt{|bx_n - c|}, \\ y_n &= a - x_{n-1}, \end{aligned} \tag{5}$$

where $a, b, c \in \mathbb{R}$ and $sgn : \mathbb{R} \to \mathbb{R}$ is defined as follows:

$$sgn(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0, \end{cases} \tag{6}$$

– Zaslavsky transformation [6]

$$\begin{aligned} x_n &= (x_{n-1} + K \sin y_{n-1}) \cos\alpha + y_{n-1} \sin\alpha, \\ y_n &= -(x_{n-1} + K \sin y_{n-1}) \sin\alpha + y_{n-1} \cos\alpha, \end{aligned} \tag{7}$$

where $K \in \mathbb{R}$, $\alpha = \frac{2\pi}{q}$, $q \in \mathbb{N}$, $q \geq 3$,

– Chip transformation created by Peters for the HOP program [9]

$$\begin{aligned} x_n &= y_{n-1} - sgn(x_{n-1}) \cos(\ln|bx_{n-1} - c|)^2 \cdot \arctan(\ln|cx_{n-1} - b|)^2, \\ y_n &= a - x_{n-1}, \end{aligned} \tag{8}$$

where $a, b, c \in \mathbb{R}$,

– Quadrup Two transformation also created by Peters for the HOP program [9]

$$\begin{aligned} x_n &= y_{n-1} - sgn(x_{n-1}) \sin(\ln|bx_{n-1} - c|) \arctan(cx_{n-1} - b)^2, \\ y_n &= a - x_{n-1}, \end{aligned} \tag{9}$$

where $a, b, c \in \mathbb{R}$.

The other examples of dynamical systems, producing interesting orbits, are: Three Ply [9], Quadrup Two [9] and Cockatoo [5]. In Fig.1 the examples of orbits for transformations (3), (5), (7), (8), (9) are presented.

3 Automatic Generation of Patterns

In this section we present the algorithms for patterns generation using one dynamical system or a set of them. The obtained geometry of shapes is further coloured with the help of the three colouring algorithms. In both algorithms more general iteration process than Picard iteration is used.

Definition 3 ([2]). *Let $T : E \to E$ be a selfmap on a real normed space $(E, \|\cdot\|)$, $x_0 \in E$ and $\lambda \in [0, 1]$. The sequence $\{x_n\}_{n=0}^{\infty}$ given by*

$$x_{n+1} = (1 - \lambda)x_n + \lambda T(x_n) \tag{10}$$

is called Krasnoselskij iteration procedure *or simply* Krasnoselskij iteration.

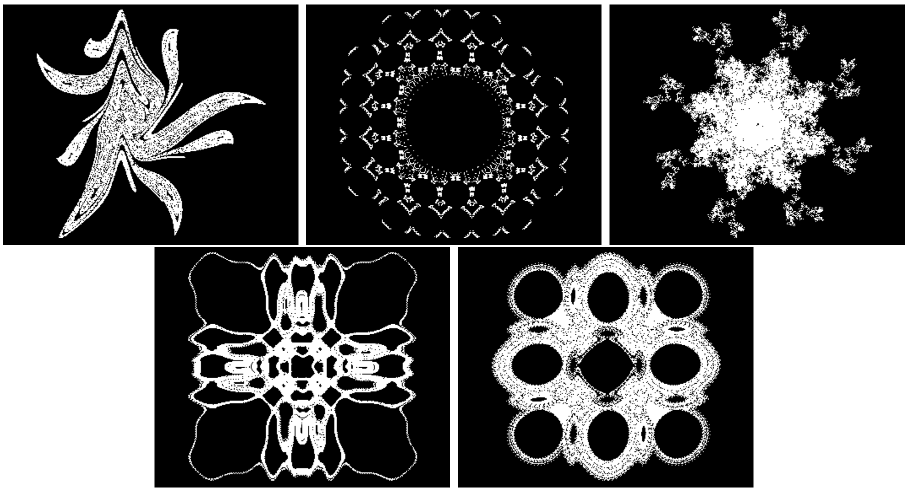

Fig. 1. The examples of orbits ($n = 70000$), the top row (from the left): Gumowski-Mira, Hopalong, Zaslavsky, the bottom row (from the left): Chip, Quadrup Two.

It is easy to see that Krasnoselskij iteration for $\lambda = 1$ reduces to Picard iteration. In [11] Krasnoselskij iteration has been used to obtain a new class of superfractals and in [10] to obtain a new class of Julia sets.

In the first algorithm we use only one dynamical system, a starting point $[x_0, y_0]^T$, the number of iterations n and a value of parameter $\lambda \in [0, 1]$. The orbit of the dynamical system is generated by the use of randomly chosen Picard or Krasnosielskij iteration at every iteration step. The algorithm is presented in Algorithm 1.

Algorithm 1: Pattern generation with the use of one dynamical system

Input: $[x_0, y_0]^T$ – starting point, $n \in \mathbb{N}$ – number of iterations, $f : \mathbb{R}^2 \to \mathbb{R}^2$ – dynamical system, $\lambda \in [0, 1]$

Output: sequence of points $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$ forming the pattern

for $i = 1$ **to** n **do**
  draw a number $r \in [0, 1]$;
  if $r < 0.5$ **then**
    $[x_i, y_i]^T = f([x_{i-1}, y_{i-1}]^T)$;
  else
    $[x_i, y_i]^T = \lambda \cdot f([x_{i-1}, y_{i-1}]^T) + (1 - \lambda)[x_{i-1}, y_{i-1}]^T$;

In the second algorithm we use: a set of dynamical systems $\{f_1, \ldots, f_k\}$, probabilities $p_1, \ldots, p_k$ such that $\sum_{i=1}^{k} p_i = 1$, $p_i > 0$, a starting point $[x_0, y_0]^T$, the number of iterations n and a value of parameter $\lambda \in [0,1]$. In each iteration step one dynamical system is drawn (e.g. f_j) from the set of them according to the given probability distribution. The point from the previous iteration is then transformed by f_j with the use of Krasnoselskij iteration. The procedure described above is presented in Algorithm 2.

Algorithm 2: Pattern generation with the use of dynamical systems set

Input: $[x_0, y_0]^T$ – starting point, $n \in \mathbb{N}$ – number of iterations, $\{f_1, \ldots, f_k\}$ – dynamical systems, $f_i : \mathbb{R}^2 \to \mathbb{R}^2$, $p_1, \ldots, p_k$ – probabilities $\sum_{i=1}^{k} p_i = 1$, $\lambda \in [0,1]$

Output: sequence of points $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$ forming the pattern

```
for i = 1 to n do
    draw a number j ∈ {1, . . . k} according to the probability distribution
    {p_1, . . . , p_k};
    [x_i, y_i]^T = λ · f_j([x_{i−1}, y_{i−1}]^T) + (1 − λ)[x_{i−1}, y_{i−1}]^T;
```

By using Algorithm 1 or 2 we obtain a sequence of points $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$ forming a pattern. That pattern can be easily modified by joining all consecutive points with lines. We can also skip some points (with a predefined step) in the joining process. These modifications often create nicely looking geometrical shapes.

A severe geometry of shapes can be enriched by using colours because the colour plays an important role in pattern perception. If we use wrong palette of colours or colours distribution over the pattern is wrong then the pattern might be considered as unattractive or not aesthetic. So, we use three different algorithms to colour points of patterns: distance colouring (Algorithm 3), iteration step colouring (Algorithm 4), mixed colouring (Algorithm 5).

The colouring according to the distance is presented in Algorithm 3. In this algorithm we use points for which we want to compute the colour, a colour map (the table of K colours) and arbitrary metric on $\mathbb{R}^2$. First, the bounding box of the points is determined together with its centre and the half length of its diagonal D. Next, for each point the distance between this point and the centre of the bounding box is computed. Having this distance, it is divided by D obtaining a number in $[0,1]$. Finally, the index of the colour in the given colour map is computed by transforming the number from $[0,1]$ to the number belonging to set $\{0, 1, \ldots, K-1\}$.

The colouring according to iteration step is presented in Algorithm 4. In this algorithm we use only the points for which the colours are computed and a colour map (the table of K colours). For each point the quotient of its number and the total number of points is computed obtaining a number in $[0,1]$. Next, by transforming this number to the number belonging to set $\{0, 1, \ldots, K-1\}$ an index of the colour from the given colour map is obtained.

Algorithm 3: Distance colouring

Input: $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$, $rgb[0..K-1]$ – colour map, K – number of colours, $d : \mathbb{R}^2 \times \mathbb{R}^2 \to [0, +\infty)$ – metric

Output: colours $c_0, \ldots, c_n$

Find the bounding box of the given points. Let $[x_c, y_c]^T$ be the centre of the bounding box and D be the half length of its diagonal;
for $i = 0$ **to** n **do**
    $j = \left\lfloor (K-1) \frac{d([x_c, y_c]^T, [x_i, y_i]^T)}{D} \right\rfloor$;
    $c_i = rgb[j]$;

Algorithm 4: Iteration step colouring

Input: $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$, $rgb[0..K-1]$ – colour map, K – number of colours

Output: colours $c_0, \ldots, c_n$

for $i = 0$ **to** n **do**
    $j = \left\lfloor (K-1) \frac{i}{n} \right\rfloor$;
    $c_i = rgb[j]$;

The last colouring algorithm (mixed colouring) is presented in Algorithm 5. In this algorithm, similarly to the two algorithms presented earlier, we use points, colour map and a distance on $\mathbb{R}^2$. The colour index is determined as the mean value of the indices obtained in Algorithms 3, 4.

Algorithm 5: Mixed colouring

Input: $[x_0, y_0]^T, \ldots, [x_n, y_n]^T$, $rgb[0..K-1]$ – colour map, K – number of colours, $d : \mathbb{R}^2 \times \mathbb{R}^2 \to [0, +\infty)$ – metric

Output: colours $c_0, \ldots, c_n$

Find the bounding box of the given points. Let $[x_c, y_c]^T$ be the centre of the bounding box and D be the half length of its diagonal;
for $i = 0$ **to** n **do**
    $j = \left\lfloor \frac{1}{2}(K-1) \left(\frac{d([x_c, y_c]^T, [x_i, y_i]^T)}{D} + \frac{i}{n} \right) \right\rfloor$;
    $c_i = rgb[j]$;

4 Examples

The examples of the proposed algorithms results are presented in Figs. 2-5. In Fig. 2 we can see the example of the pattern colouring. In the upper part there is the colour map used to colour the pattern and in the bottom part there is the same pattern coloured by the use of the proposed three colouring algorithms, from the left: the distance colouring, the iteration step colouring and the mixed colouring.

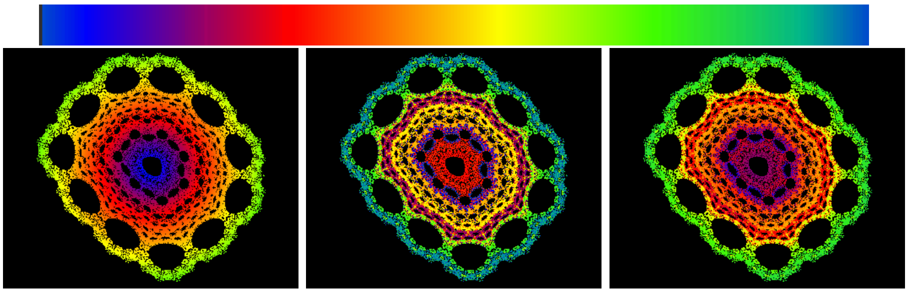

Fig. 2. The example of pattern colouring. The colour map (top), the coloured pattern (bottom from the left: the distance, the iteration step, the mixed colouring).

An example of pattern obtained from Algorithm 1 is presented in Fig. 3. In the upper part we can see the original orbit obtained by the use of Picard iteration and in the bottom part we can see the three examples of patterns obtained with the first algorithm and coloured with the use of the proposed methods. As one can see the obtained patterns have very diverse forms and differ from the original one.

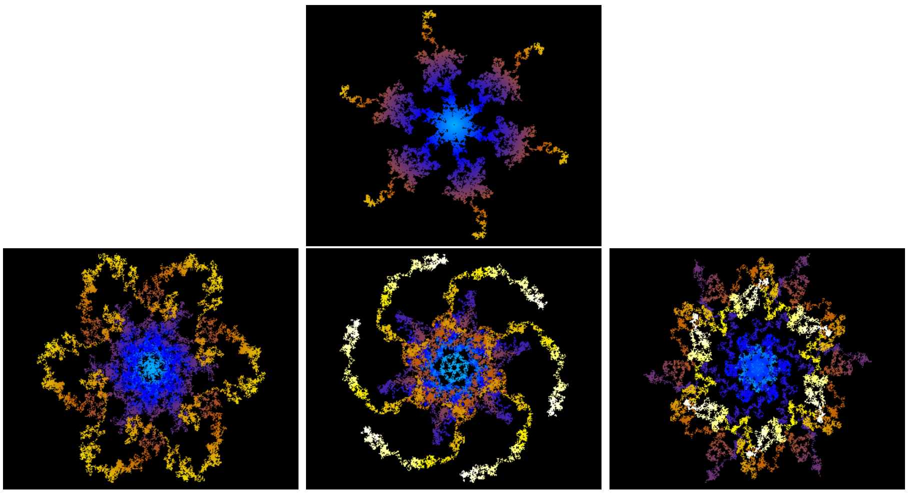

Fig. 3. The sample patterns obtained from Algorithm 1. The original orbit (top), the obtained patterns (bottom).

Another example is presented in Fig. 4. In this case Algorithm 2 was used to generate the patterns. The set of dynamical systems consists of two systems presented in the upper part of the figure. In the lower part of the figure the obtained patterns are presented. We can see that some of the patterns have inherited parts of the original patterns geometry. We can also note that we have obtained quite new geometrical forms.

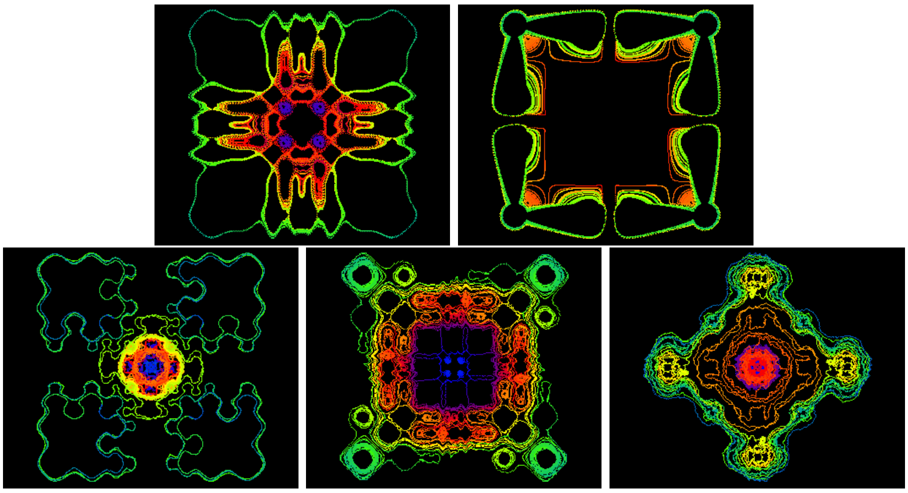

Fig. 4. The sample patterns obtained from Algorithm 2. The original orbits (top), the obtained patterns (bottom).

In the last example, presented in Fig. 5, the patterns obtained from Algorithm 2 are presented. In the upper part the patterns that form the set of dynamical systems are presented. The lower part presents the obtained patterns, but this time instead of drawing points we have joined them by lines with different omitting step. The colour of each line was determined according to the colour of the line starting point which was computed using one of the proposed colouring algorithms. The obtained examples, as one can see, form very interesting patterns.

5 Conclusions

In the paper two new algorithms for patterns generation with the use of dynamical systems are presented. The first algorithm applies only one dynamical system and the random iteration procedure – Krasnoselskij or Picard type iteration which should be drawn. In the second one a set of dynamical systems is applied and the dynamical system, which should be used to transform the point by Krasnoselskij iteration, is chosen randomly. Using these two algorithms one

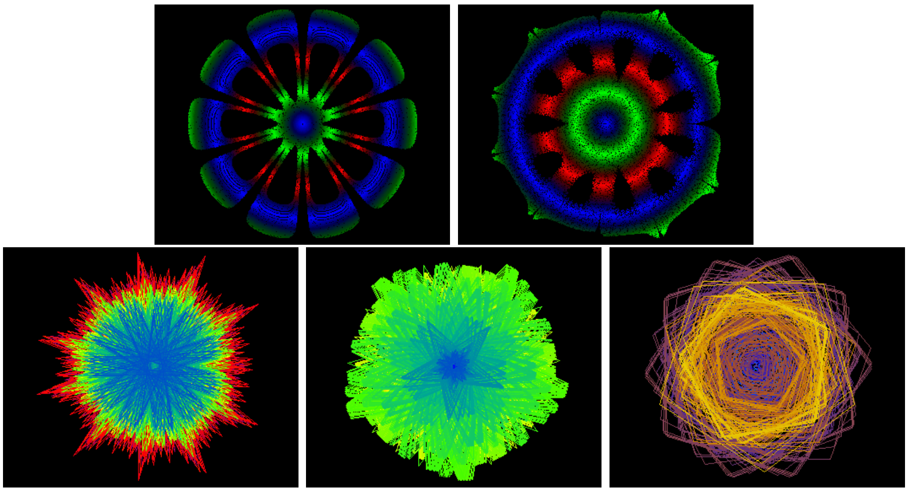

Fig. 5. The sample patterns obtained from Algorithm 2. The original orbits (top), the obtained patterns with points joined with lines (bottom).

can obtain new and interesting patterns that are very sensitive with respect to λ parameter from Krasnoselskij iteration. The best results can be obtained for $\lambda \in [0.99, 1]$, that means λ should be close to 1.

Our algorithms generate new and unrepeatable patterns which differ from the original ones, whereas the method from [7], which also is based on dynamical systems, generates patterns which are determined by the form of the used dynamical system and cannot create new patterns.

An important role in obtaining aesthetic patterns plays the colour, so we proposed three colouring algorithms. The first one is based on the distance from the centre of the points bounding box. The second one is based on the iteration number and the third one is the average of the two previous methods.

The obtained patterns have an aesthetic value, so, they can be used as any usable patterns, e.g. textile patterns, ceramics patterns, etc., or can be used in jewellery and decoration design.

In our further research we will try to replace Krasnoselskij iteration by any other types of iterations, e.g. Mann (Krasnoselskij iteration is a special case of this iteration type), Ishikawa and ergodic iterations [2]. Additionally, we will try to use texture synthesis techniques to create not only coloured but also textured patterns [13]. We also would like to concentrate on finding an automatic evaluation procedure which tells the user whether the obtained pattern satisfies some formal predefined criteria of patterns aesthetics evaluation. Such automatic evaluation procedures for fractal patterns can be found e.g. in [8], [12].

References

1. Barnsley, M.: Fractals Everywhere. Academic Press, Boston (1988)

2. Berinde, V.: Iterative Approximation of Fixed Points, 2nd edn. Springer, Heidelberg (2007)
3. Gumowski, I., Mira, C.: Recurrences and Discrete Dynamic Systems. Springer, New York (1980)
4. Kalantari, B.: Polynomial Root-Finding and Polyniomiography. World Scientific, Singapore (2009)
5. Martin, B.: Graphic Potential of Recursive Functions. In: Landsdwon, J., Earnshaw, R.A. (eds.) Computers in Art, Design and Animation, pp. 109-129. Springer, Heidelberg (1989)
6. Morozov, A.D., Dragunov, T.N., Boykova, S.A., Malysheva, O.V.: Invariant Sets for Windows. World Scientific, Singapore (1999)
7. Naud, M., Richard, P., Chapeau-Blondeau, F., Ferrier, J.L.: Automatic Generation of Aesthetic Images for Computer-assisted Virtual Fashion Design. In: Proceedings 10th Generative Art Conference, Milan, Italy (2007)
8. Pang, W., Hui, K.C.: Interactive Evolutionary 3D Fractal Modeling. Visual Computer 26(12), 1467-1483 (2010)
9. Peters, M.: HOP – Fractals in Motion, http://www.mpeters.de/mpeweb/hop/
10. Rani, M., Agarwal, R.: Effect of Stochastic Noise on Superior Julia Sets. Journal of Mathematical Imaging and Vision 36(1), 63-68 (2010)
11. Singh, S.L., Jain, S., Mishra, S.N.: A New Approach to Superfractals. Chaos, Solitons and Fractals 42(5), 3110-3120 (2009)
12. Wannarumon, S., Bohez, E.L.J.: A New Aesthetic Evolutionary Approach for Jewelry Design. Computer-Aided Design & Applications 3(1-4), 385-394 (2006)
13. Wei, B.G., Li, J.P., Pang, X.B.: Using Texture Synthesis in Fractal Pattern Design. Journal of Zhejiang University Science A 7(3), 289-295 (2006)