Title: Inversion fractals and iteration processes in the generation of aesthetic patterns

Author: Krzysztof Gdawiec

Citation style: Gdawiec Krzysztof. (2017). Inversion fractals and iteration processes in the generation of aesthetic patterns. "Computer Graphics Forum" (Vol. 36, iss. 1 (2017), s. 35-45).

Version: post-print

Volume xx (200y), Number z, pp. 1–12

Inversion Fractals and Iteration Processes in the Generation of Aesthetic Patterns

K. Gdawiec

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

Abstract

In this paper, we generalize the idea of star-shaped set inversion fractals using iterations known from fixed point theory. We also extend the iterations from real parameters to so-called q-system numbers and proposed the use of switching processes. All the proposed generalizations allowed us to obtain new and diverse fractal patterns that can be used, e.g., as textile and ceramics patterns. Moreover, we show that in the chaos game for iterated function systems – which is similar to the inversion fractals generation algorithm – the proposed generalizations do not give interesting results.

Keywords: inversion, fractal, iteration, aesthetic pattern, generative art

Categories and Subject Descriptors (according to ACM CCS): G.1.2 [Mathematics and Computing]: Approximation—Wavelets and fractals, I.3.5 [Computer Graphics]: Computational geometry and object modeling—Geometric algorithms, languages and systems, I.3.m [Computer Graphics]: Miscellaneous—

1. Introduction

Today, aesthetic patterns are widely used – for instance, they are used in jewellery design, carpet design, as textures and as patterns on wallpaper, etc. To obtain an interesting pattern, the designer must deal with different aspects, i.e., analysis, creativity and development [WUB04]. Usually, most of the work during the designing stage is carried out by the designer manually, and if the pattern is to have more unrepeatable artistic features, then the amount of work needed is larger. Therefore, it is useful to develop methods of automatic or semi-automatic pattern generation that will make the design process much easier.

In the literature, we can find many different methods of aesthetic pattern generation which are based on different approaches. For instance, in [WBA08] the authors use fractals based on iterated function systems (IFSs) and genetic algorithms. Another fractal algorithm based on the filling of space was proposed in [SB13]. An algorithm for the generation of patterns with wallpaper symmetry based on dynamics was presented in [LYZ07], and in [OCCZ12] a method based on the invariant mapping method was proposed. Field and Golubitsky [FG09] used chaotic dynamics for the generation of patterns with different types of symmetries. Recently, the attention of scientists has focused on the use of different iteration schemes from fixed point theory in the generation of patterns. The iteration schemes have been mainly used in the generation of fractal patterns defined in the complex plane, such as the well-known Mandelbrot and Julia sets [AA12, ARC14, RA10], and in polynomiography [GKL15] (a method that is based on the root-finding methods of complex polynomials).

In this paper, we present extensions of the star-shaped set inversion fractals [Gda14] which are based on different iteration schemes taken from fixed point theory and q-system numbers. The proposed extensions allow us to obtain new and diverse fractal patterns, which we had not been able to obtain previously.

The paper is organized as follows. In Sec. 2, we briefly introduce the star-shaped set inversion fractals and the algorithm to generate them. Next, in Sec. 3, some information about different types of iterations used in fixed point theory is presented. Sec. 4 is devoted to the generalization of the star-shaped set inversion fractals. We show how to use the iterations from Sec. 3 in inversion fractals. Moreover, we show the way in which we can extend the iterations to use the q-system numbers and propose a switching process. Next,

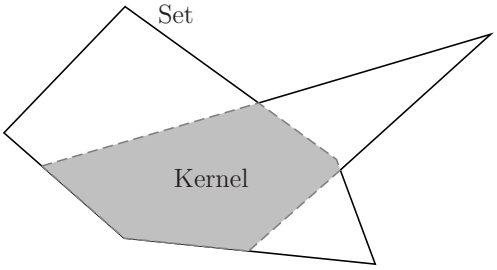


Figure 1: *A star-shaped set (a concave polygon) and its kernel.*

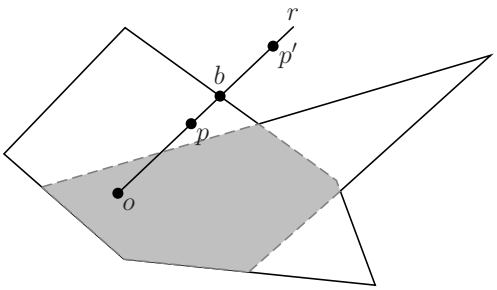


Figure 2: *Inversion of p with respect to a given star-shaped set.*

in Sec. 5, we describe a colouring method used to colour fractal patterns obtained with the help of the proposed algorithm. Some examples of fractal patterns obtained with the proposed generalizations are presented in Sec. 6. Finally, in Sec. 7, we give some concluding remarks.

2. Star-shaped Set Inversion Fractals

The first fractals based on the inversion transformation, namely circle inversion, were presented in [CF95, FC00]. In 2005, Leys in [Ley05] extended the idea of circle inversion to the sphere and presented sphere inversion fractals. Later, in 2014, Gdawiec in [Gda14] showed that we can generalize circle inversion to any star-shaped set. Moreover, he presented some examples of fractals obtained with star-shaped set inversion. In this section, we briefly introduce the star-shaped fractals presented in [Gda14].

Let us start with some basic definitions.

Definition 2.1 A set S in a metric space $(\mathbb{R}^2, d)$, where d is the Euclidean distance, is *star-shaped* if there exists a point $z \in \operatorname{int} S$ ($\operatorname{int} S$ means the interior of S) such that for all points $p \in S$ the line segment $\overline{zp}$ lies entirely within S. The locus of the points z having the above property is the *kernel* of S and is denoted by $\ker S$.

An example of a star-shaped set (a concave polygon) and its kernel (in grey) is presented in Fig. 1. Notice that every convex set is star-shaped and its kernel is equal to the interior of the set.

To define the star-shaped set inversion, let us assume that we have a star-shaped set S and some point $o \in \ker S$. Moreover, let us assume that we have a point p other than o for which we want to calculate the inversion. Now, we shoot a ray r from o in the direction $p-o$, i.e.,

$$r(t) = o + t(p-o), \tag{1}$$

where $t \in [0,\infty)$. Next, we find the intersection point of r and the boundary of S and denote it by b. The considered situation is schematically shown in Fig. 2.

Definition 2.2 Point p' is said to be the *inverse* of p with respect to S if it satisfies the following equation:

$$d(o,p) \cdot d(o,p') = [d(o,b)]^2. \tag{2}$$

Point o is called the *centre of inversion*. The transformation that takes p and transforms it into p' is called the *star-shaped set inversion transformation* and it is denoted by I_S.

At the beginning, we assumed that $p \neq o$. We can also define I_S for $p = o$. We do this in the following way: $I_S(o) = \infty$ and $I_S(\infty) = o$. In this way, I_S is defined not on $\mathbb{R}^2$ but on $\hat{\mathbb{R}}^2 = \mathbb{R}^2 \cup \{\infty\}$.

To calculate I_S for a given point equation (2) is complicated, so we can use the algebraic formula

$$\begin{aligned} p' &= I_S(p) \\ &= (x_o, y_o) + \frac{[d(o,b)]^2}{(x_p - x_o)^2 + (y_p - y_o)^2} \cdot (x_p - x_o, y_p - y_o), \end{aligned} \tag{3}$$

where $o = (x_o, y_o)$, $p = (x_p, y_p)$.

To generate fractals using the star-shaped set inversion transformation, we can use the random inversion algorithm (Algorithm 1). When we look at the algorithm, we see that it is similar to the chaos game algorithm used in the generation of fractals based on IFSs [Bar88]. The first difference between these two algorithms is that we use different types of transformations. For IFS, typically affine transformations are used while star-shaped set inversion transformation is used for the random inversion algorithm. The second difference is that in the random inversion algorithm we use an additional while loop. The function $inSet(S_l, p)$ used in this loop returns true if point p is in S_l and false otherwise. The loop is needed to ensure that the transformation used to transform the point is a contraction and that in two successive iterations we do not transform a point with the same transformation. We need these conditions because I_S is contractive only on $\hat{\mathbb{R}}^2 \setminus S$ and it is an involution (i.e., $I_S(I_S(p)) = p$ for all $p \in \hat{\mathbb{R}}^2$).

Examples of fractals generated with the use of the random inversion algorithm are presented in Fig. 3. The left side of the figure presents star-shaped sets with centres of inversion

Algorithm 1: The random inversion algorithm

Input: $S_1,\dots,S_k$ – star-shaped sets with chosen centres of inversion, p_0 – starting point external to $S_1,\dots,S_k$, $n > 20$ – number of iterations

Output: Approximation of a restricted limit set (a star-shaped set inversion fractal)

```
j = random number from {1,...,k}
p = I_{S_j}(p_0)
for i = 2 to n do
    l = random number from {1,...,k}
    while j = l or inSet(S_l, p) do
        l = random number from {1,...,k}
    j = l
    p = I_{S_j}(p)
    if i > 20 then
        Plot p
```

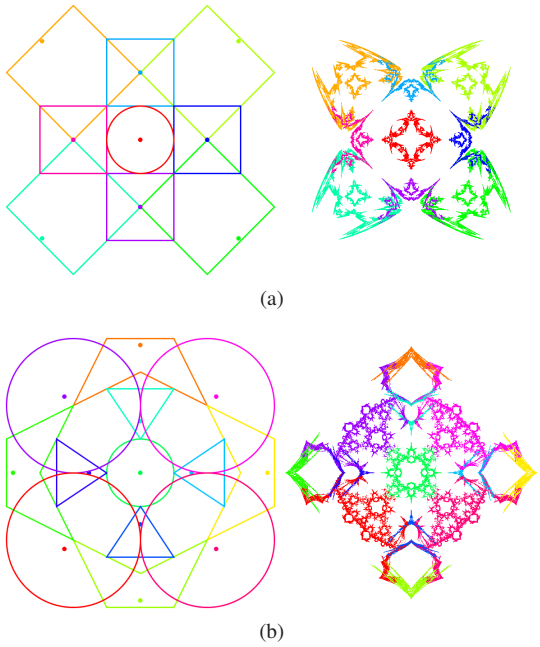


Figure 3: *Examples of the star-shaped sets defining the transformations (left) and corresponding inversion fractals (right).*

used to generate the fractals presented on the right side. Each point of the fractal was coloured according to the colour of the set used to obtain the point.

3. Iterations

In fixed point theory there exist many theorems and methods that allow one to find fixed points of a given mapping. One of the areas in the theory is an iterative approximation of the fixed points. In this area, we use different kinds of iteration processes. Let us recall some of them.

Let (X,d) be a metric space, $T : X \to X$ be a mapping and $p_0 \in X$ be a starting point.

1. The standard Picard iteration [Pic90] introduced in 1890 is defined as
$$p_{n+1} = T(p_n), \quad n \in \mathbb{N}. \tag{4}$$
2. The Suantai iteration [Sua05] was defined in 2005 as a three-step iteration process with five parameters,
$$\begin{cases} p_{n+1} = (1-\alpha_n-\beta_n)p_n + \alpha_n T(u_n) + \beta_n T(v_n), \\ u_n = (1-a_n-b_n)p_n + a_n T(v_n) + b_n T(p_n), \\ v_n = (1-\gamma_n)p_n + \gamma_n T(p_n), \quad n \in \mathbb{N}, \end{cases} \tag{5}$$
where $\alpha_n,\beta_n,\gamma_n,a_n,b_n \in [0,1]$, $\alpha_n+\beta_n \in [0,1]$, $a_n+b_n \in [0,1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty}(\alpha_n+\beta_n) = \infty$.
3. In 2007, Agarwal et al. in [AOS07] introduced the S-iteration,
$$\begin{cases} p_{n+1} = (1-\alpha_n)T(p_n) + \alpha_n T(u_n), \\ u_n = (1-\beta_n)p_n + \beta_n T(p_n), \quad n \in \mathbb{N}, \end{cases} \tag{6}$$
where $\alpha_n \in (0,1]$ and $\beta_n \in [0,1]$ for all $n \in \mathbb{N}$.
4. In 2012, Chugh et al. in [CKK12] introduced the CR iteration,
$$\begin{cases} p_{n+1} = (1-\alpha_n)u_n + \alpha_n T(u_n), \\ u_n = (1-\beta_n)T(p_n) + \beta_n T(v_n), \\ v_n = (1-\gamma_n)p_n + \gamma_n T(p_n), \quad n \in \mathbb{N}, \end{cases} \tag{7}$$
where $\alpha_n,\beta_n,\gamma_n \in [0,1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty}\alpha_n = \infty$.
5. In 2013, Karakaya et al. in [KDGE13] defined a very general three-step iteration process with five parameters,
$$\begin{cases} p_{n+1} = (1-\alpha_n-\beta_n)u_n + \alpha_n T(u_n) + \beta_n T(v_n), \\ u_n = (1-a_n-b_n)v_n + a_n T(v_n) + b_n T(p_n), \\ v_n = (1-\gamma_n)p_n + \gamma_n T(p_n), \quad n \in \mathbb{N}, \end{cases} \tag{8}$$
where $\alpha_n,\beta_n,\gamma_n,a_n,b_n \in [0,1]$, $\alpha_n+\beta_n \in [0,1]$, $a_n+b_n \in [0,1]$ for all $n \in \mathbb{N}$ and $\sum_{n=0}^{\infty}(\alpha_n+\beta_n) = \infty$.

Other iteration methods that can be found in the literature are: Mann [Man53], Ishikawa [Ish74], Noor [Noo00], SP [PS11], Khan [Kha13], Picard-S [GK14]. The presented iterations for particular values of the parameters can reduce to other iterations. For instance, if we take in the Suantai iteration $\beta_n = b_n = 0$ and $\alpha_n \neq 0$ for all $n \in \mathbb{N}$, we obtain a Noor iteration. The dependencies between all the mentioned iterations are shown in Fig. 4.

4. Iterations in Inversion Fractals

The convergence of the random inversion algorithm – similarly to the chaos game for IFS – is guaranteed by the Banach

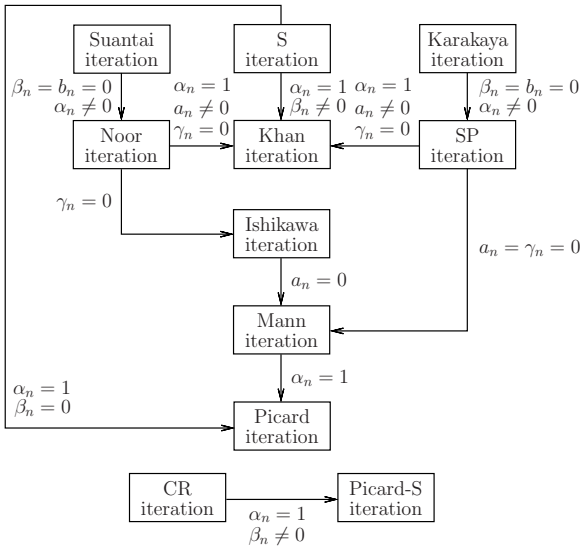


Figure 4: *Diagram of the iterations' dependencies.*

fixed point theorem. In the theorem, we use Picard iteration to find a fixed point in the Banach space of a contraction mapping. The iterations from Sec. 3 allow us to find fixed points for different types of mappings, e.g., non-expansive or contractive-like mappings, and not only for contractions. Accordingly, a straightforward idea would be to use in the random inversion algorithm instead of the standard Picard iteration the different iterations presented in Sec. 3. For instance, when we want to use the Mann iteration, the line in the algorithm with the Picard iteration (Line 8) is replaced by

$$p = (1-\alpha_i)p + \alpha_i I_{S_j}(p). \tag{9}$$

The parameters used in all the iterations are real numbers that belong to $[0,1]$ or $(0,1]$. In fixed point theory, these assumptions guarantee that the iteration process converges (weak, strong) to a fixed point of the mapping. In this paper, we are interested in obtaining interesting patterns rather than guaranteeing the convergence to a fixed point, so we can omit the assumptions about the boundedness of the parameters and take values outside the intervals $[0,1]$, $(0,1]$.

In 1994, Levin in [Lev94] introduced a generalization of complex numbers called q-system numbers and used them in complex fractals (Mandelbrot and Julia sets, biomorphs, Newton root-finding fractals). We can use the q-system numbers to extend the iterations, but before we present the extension we briefly introduce the q-system numbers.

Complex numbers are numbers of the form $a+ib$, where $i^2 = -1$ is the imaginary unit and $a,b \in \mathbb{R}$. The q-system numbers have a similar form,

$$a+qb, \tag{10}$$

where q satisfies following condition,

$$q^2 = A + q(2B) \tag{11}$$

for arbitrary $A,B \in \mathbb{R}$.

Now, let us define basic operations on q-system numbers:

1. addition
$$(a+qb)+(c+qd) = (a+c)+q(b+d), \tag{12}$$
2. subtraction
$$(a+qb)-(c+qd) = (a-c)+q(b-d), \tag{13}$$
3. multiplication
$$(a+qb)\cdot(c+qd) = (ac+bdA)+q(bc+ad+2Bbd), \tag{14}$$
4. division
$$\frac{a+qb}{c+qd} = \frac{ac+2Bad-Abd}{c^2+2Bcd-Ad^2} + q\frac{cb-ad}{c^2+2Bcd-Ad^2}. \tag{15}$$

Following elementary calculations, we can show that addition and multiplication are commutative and associative, and that multiplication is distributive over addition. Moreover, notice that for $A=-1$ and $B=0$ the q-system numbers with the basic operations reduce to complex numbers.

Let us denote the q-system numbers for fixed $A,B \in \mathbb{R}$ by $\mathbb{Q}_{A,B}$. Moreover, let us define two mappings $\Re, \Im : \mathbb{Q}_{A,B} \to \mathbb{R}$ in the following way,

$$\Re(a+qb) = a, \tag{16}$$
$$\Im(a+qb) = b \tag{17}$$

for all $a+qb \in \mathbb{Q}_{A,B}$. We see that the mappings $\Re$ and $\Im$ are analogous to the mappings used for complex numbers to obtain the real and imaginary parts of a given number.

In inversion fractals, we deal with two-dimensional real space $\mathbb{R}^2$. To use q-system numbers in inversion fractals, we need to be able to transform $\mathbb{Q}_{A,B}$ and $\mathbb{R}^2$ into each other. The transformation $R : \mathbb{Q}_{A,B} \to \mathbb{R}^2$ that transforms a q-system number into a point in two-dimensional real space is given by the formula

$$R(p) = (\Re(p), \Im(p)) \tag{18}$$

for all $p \in \mathbb{Q}_{A,B}$, and the transformation $Q : \mathbb{R}^2 \to \mathbb{Q}_{A,B}$ that transforms points from $\mathbb{R}^2$ into q-system numbers is given by the formula

$$Q(p) = x_p + q y_p \tag{19}$$

for all $p = (x_p, y_p) \in \mathbb{R}^2$.

Now, we are ready to extend the iterations with the q-system numbers. The first thing we need to do is the replacement of the real parameters by the q-system numbers, so $\alpha_n, \beta_n, \gamma_n, a_n, b_n \in \mathbb{Q}_{A,B}$. Next, we transform the points used in the desired iteration to an appropriate space using the mappings R and Q so that the calculations for the iterations are made in $\mathbb{Q}_{A,B}$ and the final result of the calculations

is in $\mathbb{R}^2$. We want it so that the final result is in $\mathbb{R}^2$ because the points of the inversion fractal are drawn in this space. For instance, let us take the CR iteration (7) with $X = \mathbb{R}^2$. According to our assumptions $\alpha_n, \beta_n, \gamma_n \in \mathbb{Q}_{A,B}$ and

$$\begin{cases} p_{n+1} = R((1-\alpha_n)u_n + \alpha_n(Q \circ T \circ R)(u_n)), \\ u_n = (1-\beta_n)(Q \circ T)(p_n) + \beta_n(Q \circ T \circ R)(v_n), \\ v_n = (1-\gamma_n)Q(p_n) + \gamma_n(Q \circ T)(p_n), \end{cases} \tag{20}$$

for all $n \in \mathbb{N}$.

In Sec. 3, we showed some iterations from fixed point theory that can be used with star-shaped set inversion. There also exist iterations that, when used with inversion transformation, reduce to other iterations irrespective of the parameters used while at the same time they are computationally expensive. For instance, let us consider following iteration introduced by Schu in [Sch91],

$$p_{n+1} = (1-\alpha_n)p_n + \alpha_n T^{(n+1)}(p_n), \tag{21}$$

where $\alpha_n \in [0,1]$. When we look at this iteration, we see that it is very similar to the Mann iteration. The only difference between these two iterations is that, in the $n+1$ iteration of the Schu iteration, we use the $n+1$-time composition of the mapping T, whereas in the Mann iteration we use only T.

Now, let us assume that $X = \mathbb{R}^2$ and that the mapping T is an involution, i.e., $T^2(p) = p$. We can easily prove by induction that

$$T^n(p) = \begin{cases} T(p), & \text{if } n \text{ is odd}, \\ p, & \text{if } n \text{ is even}. \end{cases} \tag{22}$$

The first elements of the Schu iteration for an involution appear as follows:

$$\begin{aligned} p_1 &= (1-\alpha_0)p_0 + \alpha_0 T(p_0), \\ p_2 &= (1-\alpha_1)p_1 + \alpha_1 T^2(p_1) = (1-\alpha_1)p_1 + \alpha_1 p_1 = p_1, \\ p_3 &= (1-\alpha_2)p_2 + \alpha_2 T^3(p_2) = (1-\alpha_2)p_2 + \alpha_2 T(p_2), \\ p_4 &= (1-\alpha_3)p_3 + \alpha_3 T^4(p_3) = (1-\alpha_3)p_3 + \alpha_3 p_3 = p_3, \\ p_5 &= (1-\alpha_4)p_4 + \alpha_4 T^5(p_4) = (1-\alpha_4)p_4 + \alpha_4 T(p_4). \end{aligned}$$

It is easy to show that:

$$\begin{aligned} p_{n+1} &= \begin{cases} (1-\alpha_n)p_n + \alpha_n T(p_n), & \text{if } n \text{ is odd}, \\ p_n, & \text{if } n \text{ is even}, \end{cases} \\ &= \begin{cases} (1-\alpha_n)p_n + \alpha_n T(p_n), & \text{if } n \text{ is odd}, \\ (1-0)p_n + 0 \cdot T(p_n), & \text{if } n \text{ is even}. \end{cases} \end{aligned} \tag{23}$$

Defining the sequence β_n as

$$\beta_n = \begin{cases} \alpha_n, & \text{if } n \text{ is odd}, \\ 0, & \text{if } n \text{ is even} \end{cases} \tag{24}$$

we obtain:

$$p_{n+1} = (1-\beta_n)p_n + \beta_n T(p_n). \tag{25}$$

We can see that this is a Mann iteration. Therefore, using the Schu iteration with α_n for the involution we obtain the Mann iteration with parameters given by (24). From the computational point of view, it is better to use the Mann iteration rather than the Schu iteration because we do not need to calculate the compositions of the mapping.

In the random inversion algorithm, we use k star-shaped set inversion transformations $I_{S_1}, I_{S_2}, \ldots, I_{S_k}$ that are involutions. Using what we have just shown for a transformation that is an involution, we obtain that, in the case of using the Schu iteration in Algorithm 1, we get the same result as if we had used the Mann iteration with the parameters given by (24). Moreover, using the Mann iteration instead of the Schu iteration is computationally more efficient.

Having such a variety of iterations that can be used in the generation of inversion fractals, we can use them in the so-called switching process. Let us assume that $P_0, P_1, \ldots, P_{m-1}$ are iterations ($P_i : \mathbb{R}^2 \to \mathbb{R}^2$ for $i = 0, 1, \ldots, m-1$) that do not reduce to each other for any possible combination of the parameters. In the diagram presented in Fig. 4, we can easily find such iterations. Two iterations in the diagram reduce to each other if there is a path in the diagram from one iteration to the other. As such, two iterations do not reduce if there is no path in the diagram between them. For instance, Khan and Mann iterations do not reduce to each other. Now, the switching process is defined in a following way:

$$p_{n+1} = \begin{cases} P_0(p_n), & \text{if } n \mod m = 0, \\ P_1(p_n), & \text{if } n \mod m = 1, \\ \ldots & \\ P_{m-1}(p_n), & \text{if } n \mod m = m-1. \end{cases} \tag{26}$$

In a very similar way, we can define a switching process for the q-system numbers. Let us denote by $P^{A,B} : \mathbb{R}^2 \to \mathbb{R}^2$ the iteration in which we use the $\mathbb{Q}_{A,B}$ space for the calculations. Assume that, for $i = 0, 1, \ldots, m-1$, A_i, B_i represent $\mathbb{Q}_{A_i,B_i}$. The switching process between the q-system numbers is defined in the following way:

$$p_{n+1} = \begin{cases} P^{A_0,B_0}(p_n), & \text{if } n \mod m = 0, \\ P^{A_1,B_1}(p_n), & \text{if } n \mod m = 1, \\ \ldots & \\ P^{A_{m-1},B_{m-1}}(p_n), & \text{if } n \mod m = m-1. \end{cases} \tag{27}$$

In Sec. 2 we mentioned that the random inversion algorithm is similar to the chaos game for IFSs. Therefore, we can use the change of the standard Picard iteration with other iterations, but this does not give interesting results from an aesthetic point of view. In Sec. 6, we present an example of an IFS fractal and its shape alternation with the help of Ishikawa iteration.

5. Colouring Method

In the original star-shaped set inversion fractals, each transformation has a distinct colour. During the iteration process, the point receives the colour of the transformation that was used to obtain the point. From the examples presented in Fig. 3, which were obtained using this type of colouring, we see that the distribution of the colours in the fractal is not continuous and we can distinguish, based on the colours, different parts of the fractal.

To better colour the fractal, we use another colouring method. The method is a simplified version of the colouring used in the fractal flame algorithm [DR03]. Let us assume that we want to render the fractal image of dimensions $W \times H$ and that each transformation I_{S_j} receives a distinct colour c_j. The point generation process in Algorithm 1 is expanded in the following way. At the beginning, we take a random colour c. Next, for each generated point we find the coordinates of the nearest pixel and then we do two things: (1) we increment a counter for the pixel, and (2) we take the colour c_j of the transformation used to obtain the considered point, calculate the new colour $c = (c + c_j)/2$, and set the colour of the pixel to c. In this way, we obtain a histogram $\mathcal{H}$ that tells us how many times each pixel was hit as well as an image I with a coloured fractal pattern.

Now, we calculate a maximum value in $\mathcal{H}$. Let us denote this value by $m_{\mathcal{H}}$. Next, for each pixel with the coordinates $(x,y) \in \{0,1,\ldots,W-1\} \times \{0,1,\ldots,H-1\}$, we calculate the final colour of the pixel using the gamma correction in the following way:

$$I(x,y) = \left(\frac{\log_2(1+\mathcal{H}(x,y))}{\log_2(1+m_{\mathcal{H}})} \right)^{1/\gamma} I(x,y), \qquad (28)$$

where $\gamma \in \mathbb{R}_+$.

The described extensions of Algorithm 1 and the colouring method are summarized in Algorithm 2. In the algorithm, we use the notation P_v for any iteration process described in the paper (with real and q-system parameters, with switching), where v is the vector of the parameters used in the process. The dimension of v depends on the iteration process. In the successive steps of the generation process, we use different inversion transformations, and so we extend the number of arguments of the iteration P_v from one to two, where the first argument is the inversion transformation that should be used and the second argument is the point for which we make the calculations.

Algorithm 2: Extended random inversion algorithm with colouring

Input: $S_1,\ldots,S_k$ – star-shaped sets with chosen centres of inversion, $c_1,\ldots,c_k$ – colours of the transformations, p_0 – starting point external to $S_1,\ldots,S_k$, $n > 20$ – number of iterations, P_v – iteration with parameters v, W, H – image dimensions, $\gamma \in \mathbb{R}_+$

Output: Image I with an approximation of a star-shaped set inversion fractal

for $(x,y) \in \{0,1,\ldots,W-1\} \times \{0,1,\ldots,H-1\}$ **do**
  $I(x,y) =$ black
  $\mathcal{H}(x,y) = 0$
$c =$ random colour
$j =$ random number from $\{1,\ldots,k\}$
$p = P_v(I_{S_j}, p_0)$
for $i = 2$ **to** n **do**
  $l =$ random number from $\{1,\ldots,k\}$
  while $j = l$ **or** $inSet(S_l, p)$ **do**
    $l =$ random number from $\{1,\ldots,k\}$
  $j = l$
  $p = P_v(I_{S_j}, p)$
  if $i > 20$ **then**
    $x = \lfloor x_p \rfloor$
    $y = \lfloor y_p \rfloor$
    $\mathcal{H}(x,y) = \mathcal{H}(x,y) + 1$
    $c = \frac{c+c_j}{2}$
    $I(x,y) = c$
$m_{\mathcal{H}} = \max_{(x,y)} \mathcal{H}(x,y)$
for $(x,y) \in \{0,1,\ldots,W-1\} \times \{0,1,\ldots,H-1\}$ **do**
  if $\mathcal{H}(x,y) > 0$ **then**
    $I(x,y) = \left(\frac{\log_2(1+\mathcal{H}(x,y))}{\log_2(1+m_{\mathcal{H}})} \right)^{1/\gamma} I(x,y)$

6. Examples

In Sec. 4, we mentioned that we can use different iterations in the chaos game for IFSs, but the obtained patterns did not look interesting. We start our examples with such an example. In Fig. 5(a), we see a twig obtained using the Picard iteration for IFSs consisting of four affine transformations. Each point is coloured with the colour of the transformation that was used to obtain the point. In Figs. 5(b)-(e), the use of the Ishikawa iteration is presented. The parameters used to obtain the images were as follows: (b) $\alpha_n = 0.6$, $a_n = 0.0$, (c) $\alpha_n = 0.8$, $a_n = 0.0$, (d) $\alpha_n = 0.8$, $a_n = 0.3$, (e) $\alpha_n = 0.8$, $a_n = 0.6$. From the examples, we see that the parts of the attractor increasingly overlap or move away from each other, losing connectivity. Similar behaviour is also noticeable in the case of other attractors and the use of other iterations.

Now, let us consider examples of the star-shaped set inversion fractals with the proposed extensions. Figure 6 presents an example of the use of different iterations with parameters that have real values. In Fig. 6(a), on the left, we see star-shaped sets with centres of inversion that define the inversion transformations, and on the right a fractal pattern obtained using the standard Picard iteration. The fractal patterns from Figs. 6(b)-(e) were obtained using the same inversion trans-

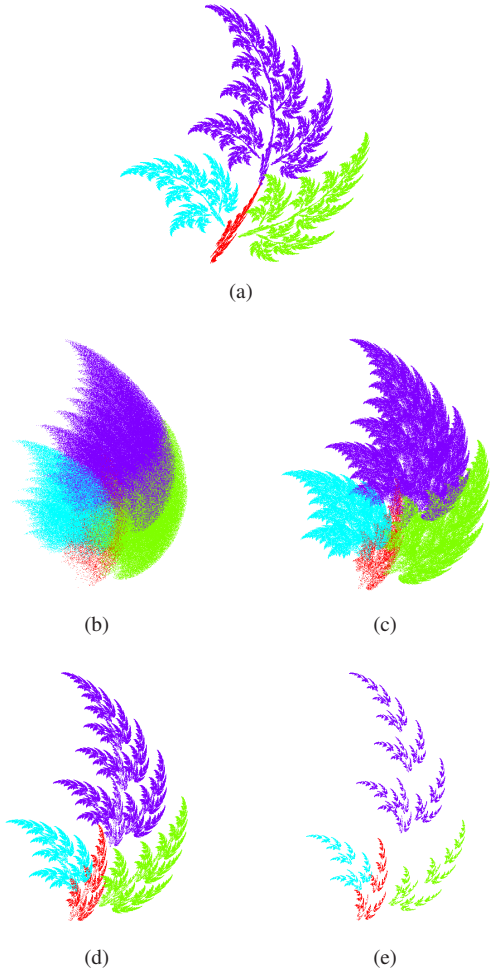


Figure 5: *Example of the (a) Picard and (b)-(e) Ishikawa iterations for an IFS fractal.*

formations but using different iterations. The iterations and their parameters were as follows:

(b) Mann iteration with $\alpha_n = 0.7$,
(c) Karakaya iteration with $\alpha_n = 0.7$, $\beta_n = 0.7$, $\gamma_n = 0.1$, $a_n = 0.1$, $b_n = 0.4$,
(d) Mann iteration with

$$\alpha_n = \begin{cases} 0.7, & \text{if } n \text{ is even,} \\ 0.9, & \text{if } n \text{ is odd,} \end{cases} \tag{29}$$

(e) Picard-S iteration with $\beta_n = 1.5$, $\gamma_n = 0.8$.

From the examples, we see that with the use of different iterations with real parameters we are able to obtain, from the basic pattern, many very diverse patterns.

In the next example we used different iterations, but this time the real parameters were replaced by the q-system numbers. Figure 7 presents the obtained fractal patterns. Similarly to the previous example in Fig. 7(a), we see a pattern obtained with the Picard iteration and the star-shaped sets defining the inversion transformations. The patterns from Figs. 7(b)-(e) were obtained using the following parameters:

(b) SP iteration with $\alpha_n = 0.5 + 0.5q$, $a_n = 0.9 + 0.5q$, $\gamma_n = 0.75 - 0.9q$ and q-system with $A = -1.0$, $B = 0.0$,
(c) SP iteration with $\alpha_n = 0.5 + 0.5q$, $a_n = 0.9 + 0.5q$, $\gamma_n = 0.75 - 0.9q$ and q-system with $A = -1.5$, $B = 0.2$,
(d) Khan iteration with $a_n = 0.3 + 0.5q$ and q-system with $A = -1.0$, $B = 0.0$,
(e) Noor iteration with $\alpha_n = 1.1 - 0.2q$, $a_n = 0.1 - 0.2q$, $\gamma_n = 0.1 - 0.2q$ and q-system with $A = -1.0$, $B = 0.0$.

The use of the q-system numbers, generally, adds some swirls and twists to the obtained pattern. Moreover, in this case, the patterns significantly differ from the original pattern obtained with the Picard iteration. Moreover, from the examples, we can observe that using the same values of the parameters but different q-systems in the iteration we obtain different fractal patterns (Fig. 7(b) and 7(c)).

The next examples show the use of the switching processes. We start with switching between iterations. In this example, as the base star-shaped set inversion fractal, we used the fractal presented in Fig. 8.

In Figs. 9 and 10, examples of patterns obtained with two different iterations and a switching process between them are presented. The parameters used to generate the images in Fig. 9 were as follows:

(a) S iteration with $\alpha_n = 0.9 + 0.9q$, $\beta_n = 0.2 + 0.2q$,
(b) Mann iteration with $\alpha_n = 0.8 + 0.1q$,
(c) switching of iterations from (a) and (b).

In each case, the same q-system with the parameters $A = -1.0$, $B = 0.0$ was used.

The parameters used to generate the images in Fig. 10 were as follows:

(a) Suantai iteration with $\alpha_n = 0.7 + 0.1q$, $\beta_n = 0.5$, $\gamma_n = 0.4 + 0.5q$, $a_n = 0.3$, $b_n = 0.1 - 0.2q$,
(b) Picard-S iteration with $\beta_n = 0.6$, $\gamma_n = 0.8$,
(c) switching of iterations from (a) and (b).

and the same q-system as in the case of Fig. 9.

By comparing the pattern in Fig. 8 with the patterns in Figs. 9(a) and (b) and 10(a) and (b) we see further examples of how the use of different iterations and q-systems affects the original star-shaped set inversion fractal. Meanwhile, by comparing images in Figs. 9(a) and (b) and 10(a) and (b) with images in Fig. 9(c) and 10(c), respectively, we see that the pattern obtained with the switching process differs from the patterns that were used to generate it, and that in this way we are able to generate new and diverse patterns.

The last example presents the use of the switching process between different q-systems. In Fig. 11, similarly to

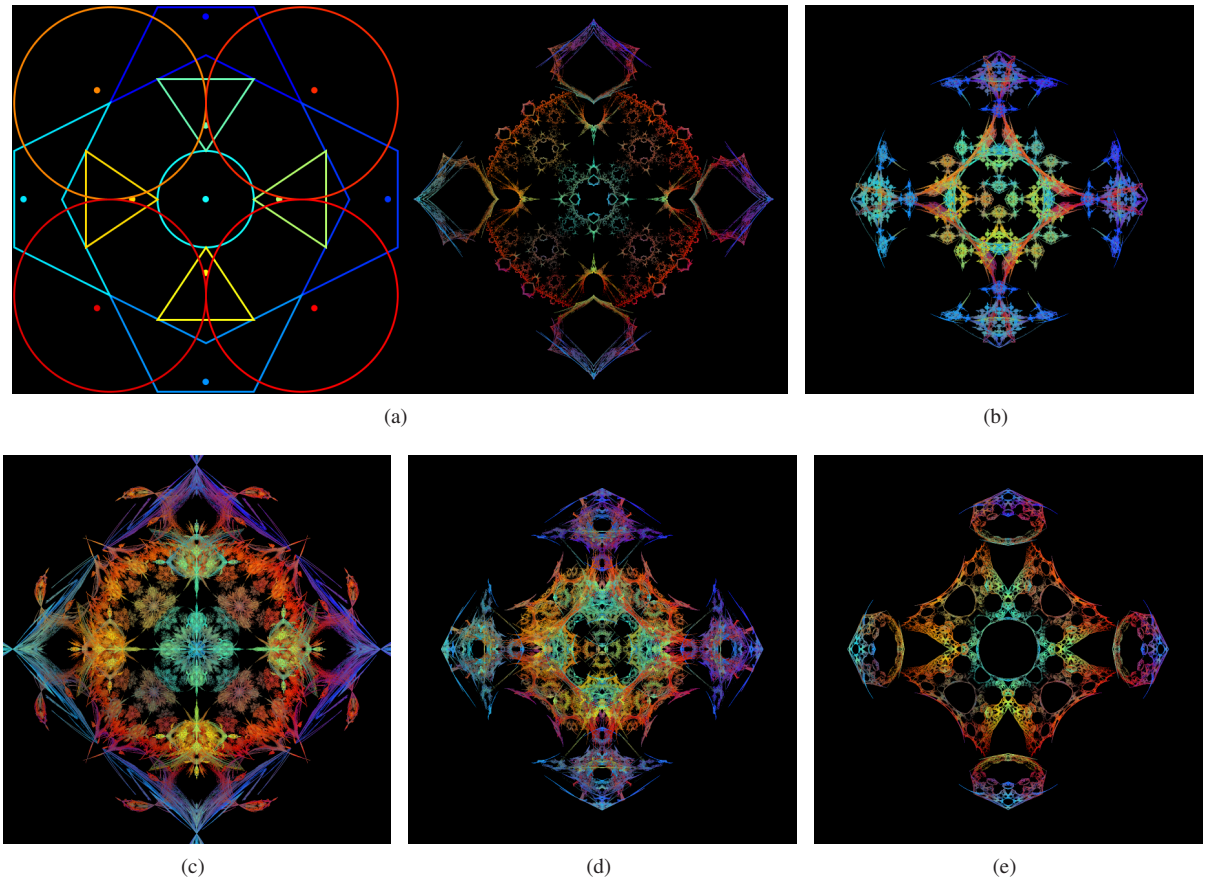


Figure 6: *(a) Sets defining the star-shaped set inversion transformations and their fractal pattern. (b)-(e) Examples of patterns obtained with the use of iterations with real parameters.*

the example with the iterations' switching, the original fractal pattern (obtained with the Picard iteration) is presented. Meanwhile, Figs. 12 and 13 present the fractal patterns obtained with two different q-systems and a switching process between them.

In Figs. 12 and 13 we used the same iteration, namely the Ishikawa iteration with $\alpha_n = 1.0 + 0.1q$, $a_n = 0.1q$. The parameters used to generate the images in Fig. 12 were as follows:

(a) q-system with $A = -3.0$, $B = 0.0$,
(b) q-system with $A = 3.0$, $B = 0.0$,
(c) switching of q-systems from (a) and (b).

The parameters used to generate the images in Fig. 13 were as follows:

(a) q-system with $A = -1.0$, $B = 0.0$,
(b) q-system with $A = -4.0$, $B = 2.0$,
(c) switching of q-systems from (a) and (b).

From the example images we also see that the switching between different q-systems gives new fractal patterns. The patterns look less interesting in comparison to the other examples, but still they are useful as, e.g., desktop wallpaper patterns.

7. Conclusions

In this paper, we presented the concept of the generalization of star-shaped set inversion fractals using different iteration processes. The use of iterations from fixed point theory gives us more possibilities for obtaining new and very interesting fractal patterns. Moreover, the use of q-systems numbers instead of real ones in the iterations gives us further diversification of the patterns.

References

[AA12] AGARWAL R., AGARWAL V.: Dynamic noise perturbed generalized superior Mandelbrot sets. *Nonlinear Dynamics 67*, 3 (2012), 1883–1891. doi:10.1007/s11071-011-0115-2. 1

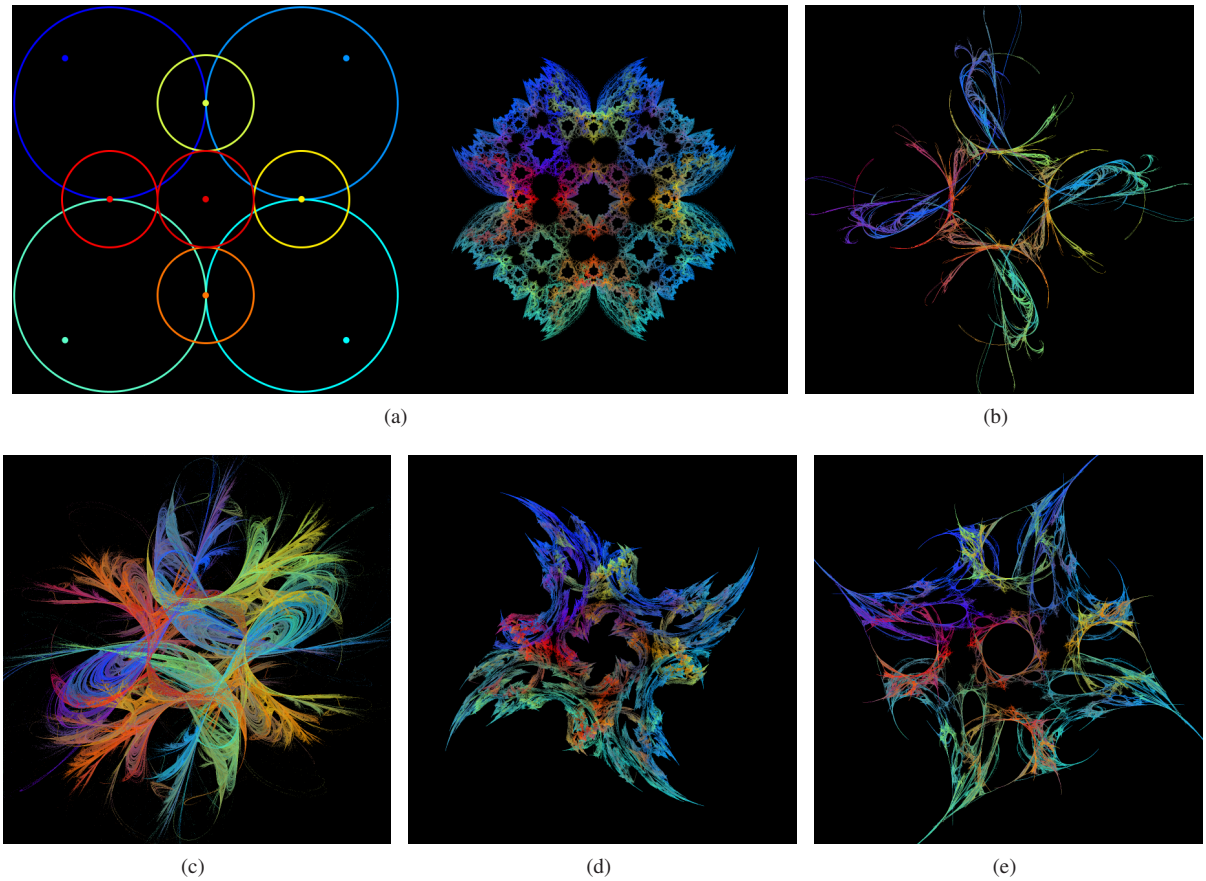


Figure 7: *(a) Sets defining the star-shaped set inversion transformations and their fractal pattern. (b)-(e) Examples of patterns obtained with the use of iterations with q-system number parameters.*

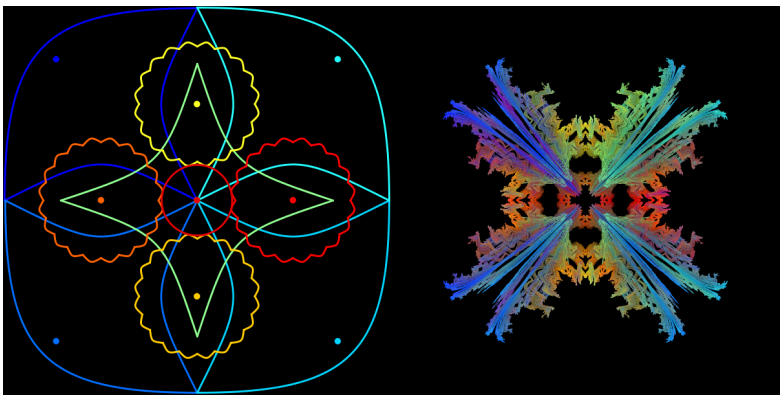
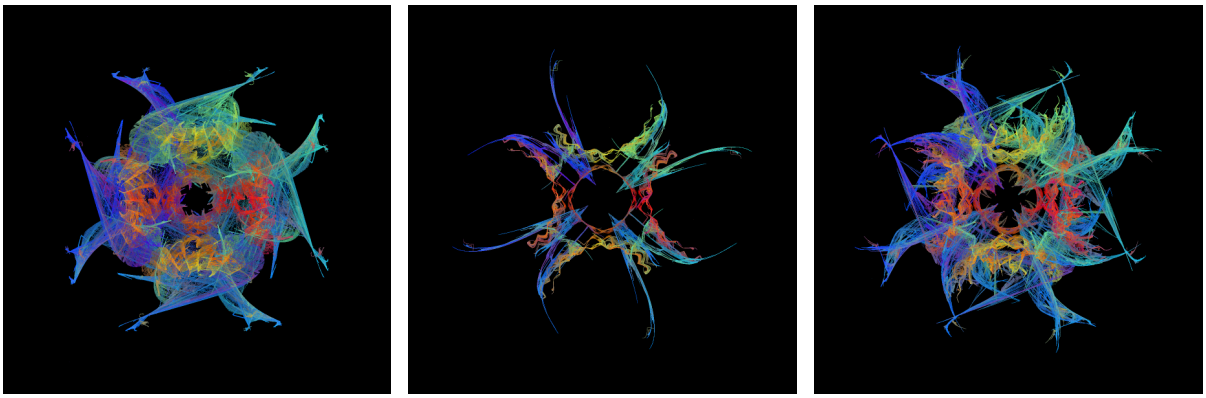
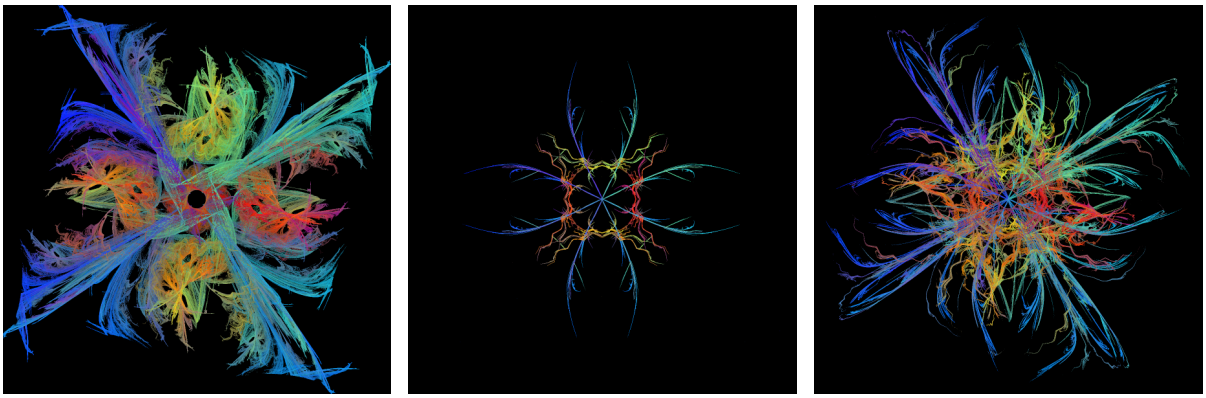


Figure 8: *Sets defining the star-shaped set inversion transformations and their fractal pattern.*



(a) (b) (c)

Figure 9: *Fractal pattern obtained with: (a) the iteration used in the even steps, (b) the iteration used in the odd steps, (c) the switching process of two iterations from (a) and (b).*



(a) (b) (c)

Figure 10: *Fractal pattern obtained with: (a) the iteration used in the even steps, (b) the iteration used in the odd steps, (c) the switching process of two iterations from (a) and (b).*

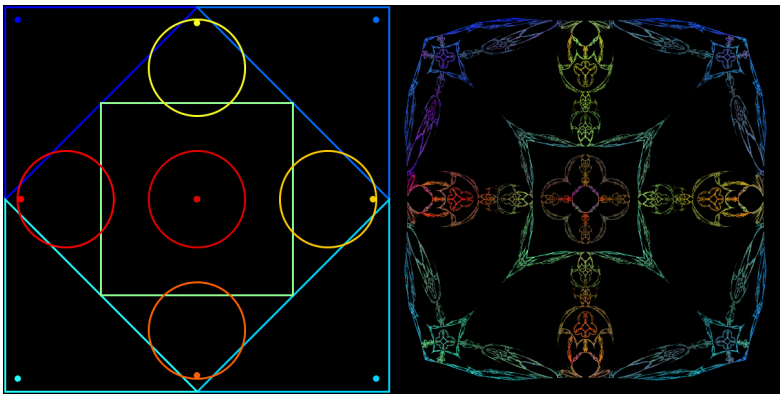
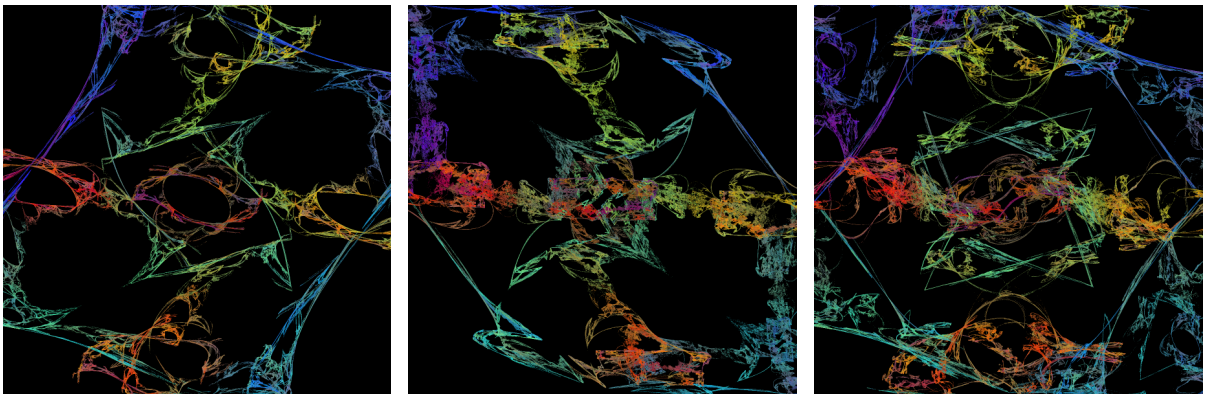
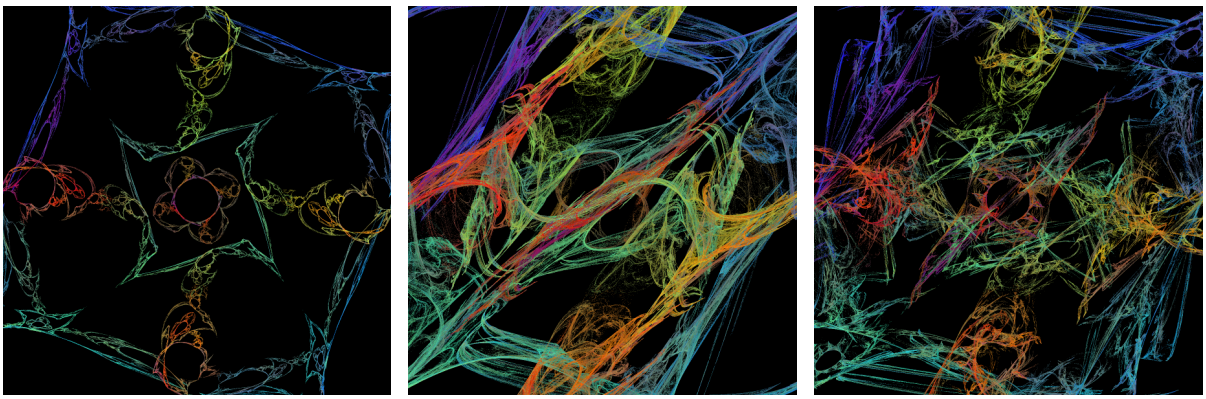


Figure 11: *Sets defining the star-shaped set inversion transformations and their fractal pattern.*



(a) (b) (c)

Figure 12: *Fractal pattern obtained with: (a) the q-system used in the even steps, (b) the q-system used in the odd steps, (c) the switching process of two q-systems from (a) and (b).*



(a) (b) (c)

Figure 13: *Fractal pattern obtained with: (a) the q-system used in the even steps, (b) the q-system used in the odd steps, (c) the switching process of two q-systems from (a) and (b).*

[AOS07] AGARWAL R. P., O'REGAN D., SAHU D. R.: Iterative construction of fixed points of nearly asymptotically nonexpansive mappings. *Journal of Nonlinear and Convex Analysis 8*, 1 (2007), 61–79. 3

[ARC14] ASHISH, RANI M., CHUGH R.: Julia sets and Mandelbrot sets in Noor orbit. *Applied Mathematics and Computation 228* (2014), 615–631. doi:10.1016/j.amc.2013.11.077. 1

[Bar88] BARNSLEY M.: *Fractals Everywhere*. Academic Press, Boston, 1988. 2

[CF95] CLANCY C., FRAME M.: Fractal geometry of restricted sets of circle inversions. *Fractals 3*, 4 (1995), 689–699. doi:10.1142/S0218348X95000618. 2

[CKK12] CHUGH R., KUMAR V., KUMAR S.: Strong convergence of a new three step iterative scheme in Banach spaces. *American Journal of Computational Mathematics 2*, 4 (2012), 345–357. doi:10.4236/ajcm.2012.24048. 3

[DR03] DRAVES S., RECKASE E.: The fractal flame algorithm, 2003. URL: http://flam3.com/flame_draves.pdf. 6

[FC00] FRAME M., COGEVINA T.: An infinite circle inversion limit set fractal. *Computers & Graphics 24*, 5 (2000), 797–804. doi:10.1016/S0097-8493(00)00080-7. 2

[FG09] FIELD M., GOLUBITSKY M.: *Symmetry in Chaos, 2nd Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009. 1

[Gda14] GDAWIEC K.: Star-shaped set inversion fractals. *Fractals 22*, 4 (2014), 1450009. doi:10.1142/S0218348X14500091. 1, 2

[GK14] GÜRSOY F., KARAKAYA V.: A Picard-S hybrid type iteration method for solving a differential equation with retarded argument, 2014. URL: http://arxiv.org/abs/1403.2546. 3

[GKL15] GDAWIEC K., KOTARSKI W., LISOWSKA A.: Polynomiography based on the non-standard Newton-like root finding methods. *Abstract and Applied Analysis 2015* (2015), 797594. doi:10.1155/2015/797594. 1

[Ish74] ISHIKAWA S.: Fixed points by a new iteration method. *Proceedings of the Ameri-*

can Mathematical Society 44, 1 (1974), 147–150. doi:10.1090/S0002-9939-1974-0336469-5. 3

[KDGE13] KARAKAYA V., DOĞAN K., GÜRSOY F., ERTÜRK M.: Fixed point of a new three-step iteration algorithm under contractive-like operators over normed spaces. *Abstract and Applied Analysis 2013* (2013), 560258. doi:10.1155/2013/560258. 3

[Kha13] KHAN S. H.: A Picard-Mann hybrid iterative process. *Fixed Point Theory and Applications 2013* (2013), 69. doi:10.1186/1687-1812-2013-69. 3

[Lev94] LEVIN M.: Discontinuous and alternate q-system fractals. *Computers & Graphics 18*, 6 (1994), 873–884. doi:10.1016/0097-8493(94)90014-0. 4

[Ley05] LEYS J.: Sphere inversion fractals. *Computers & Graphics 29*, 3 (2005), 463–466. doi:10.1016/j.cag.2005.03.011. 2

[LYZ07] LU J., YE Z., ZOU Y.: Automatic generation of colourful patterns with wallpaper symmetries from dynamics. *The Visual Computer 23*, 6 (2007), 445–449. doi:10.1007/s00371-007-0116-9. 1

[Man53] MANN W. R.: Mean value methods in iteration. *Proceedings of the American Mathematical Society 4*, 3 (1953), 506–510. doi:10.1090/S0002-9939-1953-0054846-3. 3

[Noo00] NOOR M. A.: New approximation schemes for general variational inequalities. *Journal of Mathematical Analysis and Applications 251*, 1 (2000), 217–229. doi:10.1006/jmaa.2000.7042. 3

[OCCZ12] OUYANG P., CHENG D., CAO Y., ZHAN X.: The visualization of hyperbolic patterns from invariant mapping method. *Computers & Graphics 36*, 2 (2012), 92–100. doi:10.1016/j.cag.2011.12.005. 1

[Pic90] PICARD E.: Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives. *Journal de Mathématiques Pures et Appliquées 6*, 4 (1890), 145–210. 3

[PS11] PHUENGRATTANA W., SUANTAI S.: On the rate of convergence of Mann, Ishikawa, Noor and SP iterations for countinuous functions on an arbitrary interval. *Journal of Computational and Applied Mathematics 235*, 9 (2011), 3006–3014. doi:10.1016/j.cam.2010.12.022. 3

[RA10] RANI M., AGARWAL R.: Effect of stochastic noise on superior Julia sets. *Journal of Mathematical Imaging and Vision 36*, 1 (2010), 63–68. doi:10.1007/s10851-009-0171-0. 1

[SB13] SHIER J., BOURKE P.: An algorithm for random fractal filling of space. *Computer Graphics Forum 32*, 8 (2013), 89–97. doi:10.1111/cgf.12163. 1

[Sch91] SCHU J.: Weak and strong convergence to fixed points of asymptotically nonexpansive mappings. *Bulletin of Australian Mathematical Society 43*, 1 (1991), 153–159. doi:10.1017/S0004972700028884. 5

[Sua05] SUANTAI S.: Weak and strong convergence criteria of Noor iteration for asymptotically non-expansive mappings. *Journal of Mathematical Analysis and Applications 311*, 2 (2005), 506–517. doi:10.1016/j.jmaa.2005.03.002. 3

[WBA08] WANNARUMON S., BOHEZ E. L. J., ANNANON K.: Aesthetic evolutionary algorithm for fractal-based user-centered jewelry design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 22*, 1 (2008), 19–39. doi:10.1017/S0890060408000024. 1

[WUB04] WANNARUMON S., UNNANON K., BOHEZ E. L. J.: Intelligent computer system for jewelry design support. *Computer-Aided Design and Applications 1*, 1-4 (2004), 551–558. doi:10.1080/16864360.2004.10738298. 1