Summary of Professional Accomplishments

dr Krzysztof Gdawiec

Department of Modelling and Computer Graphics
Institute of Computer Science
Faculty of Computer Science and Materials Science
University of Silesia

1. Personal data

Krzysztof Gdawiec

Institute of Computer Science, University of Silesia
ul. Będzińska 39
41-200 Sosnowiec

e-mail: krzysztof.gdawiec@us.edu.pl

www: http://kgdawiec.bplaced.net/
ORCID: http://orcid.org/0000-0001-9434-9307
ResearchID: http://www.researcherid.com/rid/O-3587-2015
Scopus: https://www.scopus.com/authid/detail.uri?authorId=35174554200
Google Scholar: https://scholar.google.pl/citations?user=Kc4rf0kAAAAJ

2. Diplomas

- 2010 – Ph.D. in computer science, Institute of Computer Science, Faculty of Computer Science and Materials Science, University of Silesia

  Thesis title: *Local fractal analysis in recognition of two-dimensional shapes* (in Polish)

  Supervisor: prof. dr hab. inż. Wiesław Kotarski

- 2005 – M.Sc. in mathematics, Institute of Mathematics, Faculty of Mathematics, Physics and Chemistry, University of Silesia

  Thesis title: *Strong version of the Master Theorem* (in Polish)

  Supervisor: dr Andrzej Biela

3. Information on employment in scientific institutions

- 10.2010–now – assistant professor, Institute of Computer Science, Faculty of Computer Science and Materials Science, University of Silesia

- 10.2009–09.2010 – assistant, Institute of Computer Science, Faculty of Computer Science and Materials Science, University of Silesia

4. Scientific achievement

4.1. Title of scientific achievement

ITERATIVE METHODS OF AESTHETIC PATTERN GENERATION

4.2. Publications that are part of scientific achievement

Basic

[A1] GDAWIEC, K. Fractal patterns from the dynamics of combined polynomial root finding methods. *Nonlinear Dynamics*. (in press) DOI: 10.1007/s11071-017-3813-6
IF: 3.464, MNiSW: 45 pts.

[A2] GDAWIEC, K. Procedural generation of aesthetic patterns from dynamics and iteration processes. *International Journal of Applied Mathematics and Computer Science*. (in press)
IF: 1.42, MNiSW: 25 pts.

[A3] GDAWIEC, K. Aesthetic patterns from the perturbed orbits of discrete dynamical systems. In *Computer Information Systems and Industrial Management*, K. Saeed, R. Chaki, A. Cortesi, and S. Wierzchoń, Eds., vol. 8104 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2013, pp. 358–366. DOI: 10.1007/978-3-642-40925-7_33
MNiSW: 15 pts.

[A4] GDAWIEC, K. Polynomiography and various convergence tests. In *WSCG 2013 Communication Proceedings* (Pilsen, Czech Republic, 2013), pp. 15–20.
MNiSW: 15 pts.

[A5] GDAWIEC, K. Mandelbrot- and Julia-like rendering of polynomiographs. In *Computer Vision and Graphics*, L. J. Chmielewski, R. Kozera, B.-S. Shin, and K. Wojciechowski, Eds., vol. 8671 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 25–32. DOI: 10.1007/978-3-319-11331-9_4
MNiSW: 15 pts.

[A6] GDAWIEC, K. Star-shaped set inversion fractals. *Fractals 22*, 4 (2014), 1450009. DOI: 10.1142/S0218348X14500091
IF: 1.22, MNiSW: 25 pts.

[A7] GDAWIEC, K. Perturbation mappings in polynomiography. In *Man–Machine Interactions 4*, A. Gruca, A. Brachman, S. Kozielski, and T. Czachórski, Eds., vol. 391 of *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2015, pp. 499–506. DOI: 10.1007/978-3-319-23437-3_42
MNiSW: 15 pts.

[A8] GDAWIEC, K. Pseudoinversion fractals. In *Computer Vision and Graphics*, L. J. Chmielewski, A. Datta, R. Kozera, and K. Wojciechowski, Eds., vol. 9972 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2016, pp. 29–36. DOI: 10.1007/978-3-319-46418-3_3
MNiSW: 15 pts.

[A9] GDAWIEC, K. Inversion fractals and iteration processes in the generation of aesthetic patterns. *Computer Graphics Forum 36*, 1 (2017), 35–45. DOI: 10.1111/cgf.12783
IF: 1.611, MNiSW: 35 pts.

[A10] GDAWIEC, K. Switching processes in polynomiography. *Nonlinear Dynamics 87*, 4 (2017), 2235–2249. DOI:10.1007/s11071-016-3186-2
IF: 3.464, MNiSW: 45 pts.

[A11] GDAWIEC, K., AND KOTARSKI, W. Polynomiography for the polynomial infinity norm via Kalantari's formula and nonstandard iterations. *Applied Mathematics and Computation 307* (2017), 17–30. DOI: 10.1016/j.amc.2017.02.038
IF: 1.738, MNiSW: 40 pts.

[A12] GDAWIEC, K., KOTARSKI, W., AND LISOWSKA, A. Automatic generation of aesthetic patterns with the use of dynamical systems. In *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds., vol. 6939 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2011, pp. 691–700. DOI: 10.1007/978-3-642-24031-7_69
MNiSW: 5 pts.

[A13] GDAWIEC, K., KOTARSKI, W., AND LISOWSKA, A. Polynomiography based on the non-standard Newton-like root finding methods. *Abstract and Applied Analysis 2015* (2015), Article ID 797594, 19 pages. DOI: 10.1155/2015/797594
MNiSW: 40 pts.

[A14] GDAWIEC, K., KOTARSKI, W., AND LISOWSKA, A. Biomorphs via modified iterations.

Journal of Nonlinear Science and Applications 9, 5 (2016), 2305–2315.
IF: 1.34, MNiSW: 35 pts.

[A15] Gdawiec, K., Kotarski, W., and Lisowska, A. Polynomiography for square systems of equations with Mann and Ishikawa iterations. In *WSCG 2016 Short Papers Proceedings* (Pilsen, Czech Republic, 2016), pp. 1–5.
MNiSW: 5 pts.

Additional

[B1] Gdawiec, K., Kotarski, W., and Lisowska, A. Automatyczne generowanie estetycznych wzorów za pomocą transformacji Gumowskiego-Miry. In *Systemy wspomagania decyzji*, A. Wakulicz-Deja, Ed. Instytut Informatyki Uniwersytetu Śląskiego, Katowice, 2011, pp. 219–226.
MNiSW: 4 pts.

[B2] Gdawiec, K., Kotarski, W., and Lisowska, A. Wielomianografia wyższych rzędów z iteracjami Manna i Ishikawy. In *Systemy wspomagania decyzji*, A. Wakulicz-Deja, Ed. Instytut Informatyki Uniwersytetu Śląskiego, Katowice, 2013, pp. 171–181.
MNiSW: 5 pts.

[B3] Gdawiec, K., Kotarski, W., and Lisowska, A. Polynomiography with non-standard iterations. In *WSCG 2014 Poster Papers Proceedings* (Pilsen, Czech Republic, 2014), pp. 21–26.
MNiSW: 5 pts.

[B4] Gdawiec, K., Kotarski, W., and Lisowska, A. Wielomianografia z niestandardową rodziną iteracji Eulera-Schrödera. In *Systemy inteligencji obliczeniowej*, U. Boryczka, M. Boryczka, and M. Przybyła-Kasperek, Eds. Instytut Informatyki Uniwersytetu Śląskiego, Katowice, 2014, pp. 75–85.
MNiSW: 5 pts.

[B5] Kotarski, W., Gdawiec, K., and Lisowska, A. On Gumowski-Mira aesthetic superfractal forms. In *Proceedings of The 2010 IRAST International Congress on Computer Applications and Computational Science* (Singapore, 2010), S. Chellappan, A. C. Cheng, M. Min, V. N. Quang, P. Ramalingam, and H.-C. Yang, Eds., International Research Alliance for Science and Technology, pp. 562–565.
MNiSW: 7 pts.

[B6] Kotarski, W., Gdawiec, K., and Lisowska, A. Metody generowania estetycznych wzorów. In *Systemy wspomagania decyzji*, A. Wakulicz-Deja, Ed. Instytut Informatyki Uniwersytetu Śląskiego, Katowice, 2012, pp. 331–339.
MNiSW: 4 pts.

[B7] Kotarski, W., Gdawiec, K., and Lisowska, A. Polynomiography via Ishikawa and Mann iterations. In *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, Eds., vol. 7431 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2012, pp. 305–313. DOI: 10.1007/978-3-642-33179-4_30
MNiSW: 10 pts.

4.3. Review of scientific achievement

One of the most elusive goals in computer aided design is artistic design and pattern generation. Pattern generation involves diverse aspects: analysis, creativity and development [39]. A designer has to deal with all of these aspects in order to obtain an interesting pattern, which later could be used in jewellery design, carpet design, as a texture etc. Usually the most work during the design stage is carried out by a designer manually, especially in the cases in which the designed pattern should contain some unique, unrepeatable artistic features. Therefore, it is highly useful to develop methods (e.g. automatic, semi-automatic) that will assist pattern generation, and will make the whole process easier.

Aesthetics in the world of art and photography is connected with the principles of the nature and the perception of beauty. Assessing the beauty and other aesthetic features of patterns, paintings, photographs is a highly subjective task, so there is no standard method of measuring aesthetic values. Development of such methods is a challenge in the discipline named computational aesthetics. However, in most of the works about pattern generation the aesthetic is assessed by the subjective feeling of the authors and not by the aesthetic measures developed in computational aesthetics.

In the literature we can find many pattern generation methods that found various applications: jewellery design [27, 36, 38], textile [6, 10, 20] or package decoration [40] patterns design, in games (textures, assets) [34], in architecture [11, 41] or purely artistic applications [21, 22]. In all the methods for pattern generation various approaches are used. The most basic approach is the use of various mathematical formulas and notions. Among these methods the mostly used methods are fractals, e.g., iterated function systems (IFS) [5, 27, 38], complex fractals [12, 35, 37], and methods that use dynamical systems, e.g., orbits of dynamical systems [23, 25] or its dynamics [3, 19, 26]. Besides the methods that use various mathematical equations we can find in the literature many other approaches to aesthetic pattern generation. The most popular approach is the use of different types of grammars. Shape grammars are the most popular type of grammars used among these methods. For instance, they were used to generate ethnic Zhuang embroidery designs [10] and Islamic geometric patterns [31]. Other types of grammars used to generate patterns are collage grammars [15] and L-systems [2]. Further popular approaches used in the field include the use of: graph methods [41], cellular automata [8], neural networks [33] and Petri nets [16]. Moreover, we can also find programmable methods [18], methods that are based on examples [29] and in which a user-driven planning strategy is used [1].

In my research work on aesthetic pattern generation methods I have chosen the approach based on the mathematical formulas. Because the fractal methods and the ones based on dynamical systems give many possibilities of obtaining patterns with shapes, that are non-trivial and rich in details my attention has focused on this type of methods. In the research I developed various aesthetic pattern generation methods, that can be divided into the following groups:

- methods that generate patterns with the help of complex polynomial root finding algorithms and methods for solving systems of non-linear equations,
- biomorphs,
- inversion and pseudoinversion fractals,
- methods that generate patterns with the use of orbits of discrete dynamical systems,
- methods that generate patterns with the use of dynamics of discrete dynamical systems.

In the next points the individual groups of the developed methods will be reviewed.

4.3.1. Patterns generated with the help of complex polynomial root finding algorithms and methods for solving systems of non-linear equations

Methods, which use complex polynomial root finding algorithms are, besides the Julia and the Mandelbrot sets, one of the most used fractal methods for generating aesthetic patterns. Around 2000 these methods obtained its own name – polynomiography. One image generated with the help of these methods is called polynomiograph. B. Kalantari was the creator of these notions, and he defined polynomiography as: the art and science of visualisation in approximation of the zeros of complex polynomials, via fractal and non-fractal images created using the mathematical convergence properties of iteration functions [12]. In 2005 Kalantari obtained an U.S. patent on the use of polynomiography in the generation of aesthetic patterns [13].

The main components in polynomiography, that have impact on the generated pattern's shape, are the polynomial and the root finding method. Most commonly used algorithm for root finding of complex polynomial is the classical Newton's method. In the literature we can find many other methods, e.g., the Halley method, the Whittaker method, the Traub-Ostrowski method, and each of these methods can be successfully used in the aesthetic pattern generation. Besides the polynomial and the root finding method in the basic pattern generation algorithm we need the area in which we generate the pattern, the maximum number of iterations, the accuracy of computations and colour map that will be used to colour the pattern. Algorithm 1 presents the basic version of polynomiography pattern generation method.

Algorithm 1: Polynomiograph generation

Input: $p \in \mathbb{C}[Z]$ – polynomial of degree at least 2, $R : \mathbb{C} \to \mathbb{C}$ – the root finding method, $A \subset \mathbb{C}$ – area, M – the maximum number of iterations, ε – accuray, $colours[0..C]$ – colour map.

Output: polynomiograph for the area A

for $z_0 \in A$ **do**
    $n = 0$
    while $n \leq M$ **do**
        $z_{n+1} = R(z_n)$
        if $|z_{n+1} - z_n| < \varepsilon$ **then**
            break
        $n = n + 1$
    determine the colour for z_0 using the colour map *colours*

Paper [B6] presents a review of some aesthetic pattern generation methods among which one can find polynomiography. Besides the history of polynomiogrphy it also presents applications of polynomiography in the visualization of perturbation matrices, i.e., matrices in which in each row and column we have only one element equal to 1 and the remaining elements are equal to 0. The article was an introduction to further work on polynomiography and other methods of aesthetic pattern generation.

In the basic algorithm for polynomiograph's generation (Algorithm 1) for the iteration of a point an iteration of the following form:

$$z_{n+1} = R(z_n). \tag{1}$$

is used. This type of iteration is often used in many numerical algorithms and in other fields, e.g., in the fixed point theory (e.g., Banach fixed point theorem). In the fixed point theory iteration (1) is called the Picard iteration. Besides this type of iteration in fixed point theory the researchers study other types of iteration, which are used in the appoximation of the fixed points of different kinds of functions, not necessarily contractive ones. Examples of such iterations are:

- Mann iteration [24]

$$z_{n+1} = (1 - \alpha_n)z_n + \alpha_n T(z_n), \tag{2}$$

  where $\alpha_n \in (0, 1]$ and T is the function for which we search for the fixed points,

- Ishikawa iteration [9]

$$\begin{aligned} z_{n+1} &= (1 - \alpha_n)z_n + \alpha_n T(v_n), \\ v_n &= (1 - \beta_n)z_n + \beta_n T(z_n), \end{aligned} \tag{3}$$

  where $\alpha_n \in (0, 1]$, $\beta_n \in [0, 1]$ and T is the function for which we search for the fixed points.

Papers [B2, B7] present the use, in the polynomiography generation algorithm, instead of the Picard iteration the Mann and the Ishikawa iterations, where in these iterations the role of T plays the root finding method R. As the root finding method in [B7] the Newton method (the first order method) and in [B2] the elements of the so-called basic family (the higher order methods) were used. In case of both papers in the examples the polynomials given in explicit form and obtained from the permutation matrices were used. The examples showed that the change of the Mann and Ishikawa iteration parameters' values has a significant effect on the shape of the obtained pattern and that with the use of other types of iterations we are able to obtain patterns, that we had not been able to obtain with the help of the Picard iteration. Moreover, in [B2] the impact of other parameters that are used in the polynomiograph generation algorithm on the pattern's shape was studied. The studies covered the change of the basic family elements and the maximum number of iterations. In case of these parameters the change of the pattern was noticeable, but not as significant as in the case of the use of the Mann and Ishikawa iteration.

In line 5 of Algorithm 1 we check if the modulus of the difference between two consecutive elements of the generated sequence is less than the given accuracy. This is the so-called convergence test of the algorithm. In Algorithm 1 we use the standard test, which is used in many iterative algorithms. In [A4] it was noticed that the standard convergence test is equivalent with the computation of the distance between two complex plane elements with the use of the modulus metric. Basing on this observation, firstly, the use of other standard metrics and the metrics generated by various theorems on metric spaces was proposed. Next, some modifications of the convergence tests were proposed. The first modification was the omission of some of the assumptions needed for the function generated by the theorems to be a metric. The second modification was the addition of weights into the functions that are metrics. The next group of convergence tests was created by the use of various metrics, weights and joining the conditions with logical conjunctions. The last group of the proposed convergence tests were the tests based on the idea taken from the escape time algorithm used for the generation of Julia and Mandelbrot sets. The examples showed that with the help of a simple alternation of the metric in the standard convergence test the shape of the generated pattern changes in a small way. More interesting results, from the point of generating patterns, were obtained using the other proposed convergence tests.

Basing on the idea introduced in [B2, B7] in articles [A13] [B3, B4] the use of further iterations from fixed point theory was proposed. In [B3, B4] the authors used seven different iterations (Mann, Ishikawa, Noor, Khan, SP, Suantai, Karakaya) and in [A13] ten iterations – the same seven as in [B3, B4] and the S, CR and Picard-S iterations. When we look at the Mann (2) and the Ishikawa (3) iterations we can notice that for particular values of the α_n and β_n parameters these iterations can reduce to other iterations, e.g., for $\alpha_n = 1$ and $\beta_n = 0$ the Ishikawa iteration reduces to the Picard iteration, and for $\beta_n = 0$ to the Mann iteration. In each of the papers [A13] [B3, B4] the authors studied the dependencies between the iterations. Moreover, in [A13] the use in the iterations of complex parameters instead of the real ones was proposed. In the examples presenting the capabilities of using the different iterations the following root finding methods and polynomials were used: the Newton method for $z^3 - 1$ in [B3], the elements of the Euler-Schröder family for $z^4 + 4$ and $z^3 - 3z + 3$ in [B4] and the Newton method for $z^7 + z^2 - 1$ and the E_3 method (element of the Euler-Schröder family) for $z^4 + 4$ in [A13]. In case of the [A13] paper the authors also presented examples showing the impact of the complex parameters and different convergence tests (from [A4]) on the shape of the obtained patterns. In [A13] [B3] the impact of changing the colour map on the graphics feeling of the generated patterns was also studied. For this purpose the same pattern was generated using various colour maps. The patterns were later given to the visual analysis.

Polynomiography is primarily used as a method of visualization of the root finding process of complex polynomials. For this purpose one can use standard rendering methods: basins of attraction or method based on the iteration number (Algorithm 1). In [A5] two polynomiograph rendering algorithms were proposed. Their purpose was the generation of artistic patterns and not the visualization of the root finding process. Both algorithms are based on the ideas taken from the standard methods of rendering of Julia and Mandelbrot sets. Similarly to the case of the Julia and the Mandelbrot sets when we iterate the root finding method R we use a constant $c \in \mathbb{C}$. In the first proposed algorithm, for each point $z_0 \in A$ the constant c is taken as $f(z_0)$, where $f : \mathbb{C} \to \mathbb{C}$ is some function. Next, the standard iteration process which uses the Picard iteration is replaced by the following process:

$$z_{n+1} = R(z_n) - c. \tag{4}$$

After performing the iteration the constant c is transformed with the use of additional function $g : \mathbb{C} \to \mathbb{C}$. The idea of this algorithm reminds the idea of rendering of the Mandelbrot set. In the second proposed algorithm the constant c is an input parameter of the algorithm, similarly to the case of generating Julia sets. Next, the iteration process is replaced by the following process:

$$z_{n+1} = R(z_n) + c. \tag{5}$$

Similarly to the first algorithm after performing the iteration the constant c is transformed by an additional function $f : \mathbb{C} \to \mathbb{C}$. In [A5] one can find examples presenting different patterns generated with the use of various constants c, functions f and g, and various convergence tests. The presented examples showed a big potential of the proposed rendering algorithms in the generation of intriguing and aesthetic patterns.

In Algorithm 1, for each $z_0 \in A$ we can look at the sequence $z_1, z_2, \ldots$ as on an orbit of discrete dynamical system. Using this observation in [A7] a modification of the iterative process with the orbit's perturbation was proposed. The perturbation of the orbit was made by a perturbation function $\rho : \mathbb{C} \times \mathbb{N} \to \mathbb{C}$ in the following way:

$$z_{n+1} = R(\rho(z_n, n + 1)). \tag{6}$$

Thus, the point generated in the previous iteration is subject to perturbation. Moreover, the convergence test was modified, so it also includes the perturbation function:

$$|z_{n+1} - \rho(z_n, n+1)| < \varepsilon. \tag{7}$$

The iteration process and the convergence test are more general than the ones from Algorithm 1, because for $\rho(z, n) = z$ we obtain the standard iteration process and the convergence test. In the same article another modification was proposed. The iteration process was replaced by a combination of the standard iteration and the iteration with the perturbation. This process is defined in the following way:

$$z_{n+1} = \alpha R(z_n) + (1 - \alpha) R(\rho(z_n, n+1)), \tag{8}$$

where $\alpha \in \mathbb{C}$. The presented examples of patterns generated with the perturbed iteration process and the combination of iteration processes showed that using these methods we are able to obtain many interesting patterns, which we had not been able to generate using the standard methods known in polynomiography.

The next idea how to modify the iteration process that occur in polynomiography was the use of switching processes. The idea was proposed in [A10], where four group of switching processes were introduced: (1) switching of the root finding methods, (2) switching of the iterations, (3) switching of the polynomials, (4) switching of the convergence tests. In each of the groups three processes were proposed: (1) modulo, (2) using the $|z|$, (3) using the $|p(z)|$, where p is a polynomial. In case of switching of the iterations and the polynomials we cannot use any iteration and polynomial. For iterations, we select such iterations that do not reduce to each other. If we had let the use of iterations that reduce to each other, then the following situation could happen: we do not have a switching process, but a single iteration process. For polynomials, firstly we select a set of roots S of the first polynomial, then from S we select subset of the roots for the second polynomial. Such choice of the roots is due the fact that if the polynomials could have arbitrary roots, especially disjointed, then in the switching process of the polynomials the root finding method might not converge to any of the roots. In [A10] examples presenting the use of each of the proposed switching processes were presented. Those examples showed artistic potential of the switching processes.

When creating patterns often we use combination of different techniques to obtain new patterns. Pattern generation methods presented among other things in [A7, A13] used only one method of root finding, iteration or polynomial, so if we want to obtain patterns from their combination, then we need to use some graphics software like GIMP or Adobe Photoshop. To generate combination of patterns we must introduce new methods. We can acknowledge the switching processes from [A10] as a first attempt to create such methods. In [A1] three different methods of combining the methods from polynomiography were introduced. In the first method, a new iteration process was introduced that is based on the use of the affine and the s-convex combinations of root finding methods. In fixed point theory besides iterations, which search for fixed points of only one function we can find iterations that are used for an approximation of common fixed points of several functions. The Das-Debata iteration [4] is an example of such iterations, and it has the following form:

$$\begin{aligned} z_{n+1} &= (1 - \alpha_n) z_n + \alpha_n T_2(v_n), \\ v_n &= (1 - \beta_n) z_n + \beta_n T_1(z_n), \end{aligned} \tag{9}$$

where $\alpha_n \in (0,1]$, $\beta_n \in [0,1]$, and T_1, T_2 are the functions for which we search for the common fixed points. The second proposed method in [A1] was the use instead of iterations for a single root finding method the iterations that search for common fixed points. The last proposed method was the so-called multistep polynomiography. This method consists of the selected number of steps. In each step we give: the polynomial, the iteration, the maximum number of iterations, the convergence test and the so-called area transformation function. In each step of the multistep polynomiography we make an iteration process as in the standard polynomiography (lines 2–7 in Algorithm 1) and at the end of the step we transform the difference between the last generated point and the starting point using the area transformation function. The transformed point is a starting point for the next step. In [A1] the author gives also some remarks on how to implement the proposed methods using the OpenGL Shading Language (GLSL). The presented examples showed that the use of combination of the root finding methods and the iterations gives new capabilities in the generation of aesthetic patterns. In particular, the multistep polynomiography gives many capabilities in obtaining patterns that are rich in different details.

In [14] Kalantari introduced methods of finding the maximum of $|p(z)|$ in $D = \{z \in \mathbb{C} : |z| \leq 1\}$. The first proposed method was based on the finding of fixed points of properly defined function. In the second method, Kalantari studied a method of solving an equation given by a pseudo-polynomial given by the input polynomial p. For solving the equation he used a pseudo-Newton method for sequence G_n of functions. Moreover, Kalantari introduced polynomiographs presenting the behaviour of both methods. In [A11] it was proved, that G_n is C^∞ class and the form of the k-th derivative of this functions. These two facts allowed to extend the idea of the pseudo-Newton method on other root finding methods, giving in this way many new methods, e.g., the pseudo-Halley method, the pseudo-basic family. Moreover, similarly like in [A13] the authors proposed the use of different iterations from fixed point theory. In comparison to [A13] the list of iterations was expanded with the iterations that appeared in the recent two years. The dependencies between these iterations were also studied. Finally, the list consists of 17 iterations. In the paper different examples of polynomiographs were presented. The examples included: the use of various iteration methods in the pseudo-Newton method, the change of the parameter in the Mann iteration, the use of various pseudo-methods and examples of purely artistic value.

Polynomiography, except the visualization of the root finding process, can be also used in visualization of other processes. In [A15] the authors proposed the use of the polynomiography techniques in the visualization of finding the solution of a system of non-linear equations – the focus was on the system of two equations with two variables. To solve the system the vector version of the Newton method was used. Moreover, the use of the Mann and Ishikawa iterations instead of the Picard one was proposed. The presented examples of the basins of attraction and polynomiographs coloured based on the iteration count showed very intriguing patterns that could be used, for instance, as patterns on wallpapers.

4.3.2. Biomorphs

In 1986 Pickover [28] introduced biomorphs (biological morphologies). He discovered them accidentally while writing a program for drawing approximation of the Julia sets. In his program he made a mistake which caused that the generated images were completely different from the expected Julia sets. The obtained shapes remained single cellular organisms with internal structures – organelle.

In the generation of a biomorph for each point z_0 from the considered area of the complex plane we use like in the case of Julia sets a similar iterative process:

$$z_{n+1} = f(z_n) + c, \tag{10}$$

where $f : \mathbb{C} \to \mathbb{C}$ is the function which defines the biomorph and $c \in \mathbb{C}$. Function f can be any function, e.g., Pickover in his examples used: $f(z) = z^3$, $f(z) = z^2 + \sin z$, $f(z) = 1/z^5$. Constant c plays a similar role like in the case of the Julia sets.

Two modifications of the biomorphs' generation algorithm were proposed in [A14]. Because the iteration process (10) has the form of the Picard iteration, where $T(z) = f(z) + c$, so the first modification replaces the Picard iteration with the Mann and Ishikawa iterations. The second modification is based on the use of two constants $c_1, c_2 \in \mathbb{C}\setminus\{0\}$. These two constants are used to define a recurrent sequence:

$$\begin{cases} d_0 = c_1, \\ d_{2n-1} = \frac{1}{c_1^{2n-1}} - d_{2n-2}, \quad n \geq 1, \\ d_{2n} = \frac{1}{c_2^{2n}} - d_{2n-1}, \quad n \geq 1, \end{cases} \tag{11}$$

which is then used in the original biomorphs' generation algorithm instead of the constants c. In the article several different examples were presented: the use of various constants c_1, c_2, the change of the parameters in the Mann and Ishikawa iterations and the use of both the constants c_1, c_2 and the Mann and Ishikawa iterations. Each of the examples showed that with the help of the proposed modifications one can obtain patterns with shapes different than those obtained with the original Pickover's algorithm. Moreover, the examples with the Mann and Ishikawa iterations showed that the shape of the biomorph changes in a continuous way with the change of values of the parameters used in these iterations. Using this observation one can create animation of changing organisms.

4.3.3. Inversion and pseudoinversion fractals

Circle inversion was introduced in antiquity by Apollonius of Perga. Since then it found many applications in geometry. In circle inversion we have a circle C with the centre (x_o, y_o) and the radius R, which defines the following transformation:

$$I_C(p) = (x_o, y_o) + \frac{R^2}{(x_p - x_o)^2 + (y_p - y_o)^2}(x_p - x_o, y_p - y_o), \tag{12}$$

where $p = (x_p, y_p)$. The circle centre is called the centre of inversion.

In 2000 Frame and Cogevina in [7] used circle inversion to define new fractals – the so-called circle inversion fractals. The circle inversion fractal is given by a finite set of circles which creates transformations similarly like in the case of fractals given by the iterated function system. In their work Frame and Cogevina also introduced two algorithms for generating the circle inversion fractals. The algorithms are analogous to the deterministic and the random method used to generate the IFS fractals.

In [A6] it was observed that the radius R in the circle inversion is the distance from the centre of the circle to the intersection point of the boundary of the circle and a ray from the centre and passing through the point for which we calculate the inverse transformation. Using this observation the inversion transformation was extended to a star-shaped set, i.e., set having the following property: inside this set one can find point z such that for each point p of this set the segment between z and p lies entirely within this set. In the proposed extension, for a star-shaped set we fix one point which satisfies the property of the star-shaped set and we take it as the centre of inversion. Then, the star-shaped set inversion transformation takes the following form:

$$I_S(p) = (x_o, y_o) + \frac{[d(o,b)]^2}{(x_p - x_o)^2 + (y_p - y_o)^2}(x_p - x_o, y_p - y_o), \tag{13}$$

where d – the Euclidean metric, $o = (x_o, y_o)$ – the centre of inversion, $p = (x_p, y_p)$ – the point for which we calculate the inversion, b – the intersection point of the boundary of a star-shaped set S and a ray from o and passing through p. Of course, circle is a special case of a star-shaped set in which each interior point satisfies the property of the star-shaped set. Thus, for the circle as the centre of inversion we can take any interior point and not only its centre as in the case of circle inversion. Using the star-shaped set inversion instead of the circle inversion the possibilities of obtaining fractals patterns generated with the random algorithm for circle inversion given in [7] were expanded. In [A6] the author also presented various examples of fractal patterns obtained with the help of the star-shaped set inversion. The first example presented the circle inversion with changing centres of inversion. This example showed that we are able to deform the circle inversion fractals in an easy way. In the next example the impact of the change of the shape of the sets defining the inversion transformations on the shape of the resulting fractal was presented. The last example presented various fractals obtained with the help of the star-shaped set inversion. In all the examples to colour the patterns the same algorithm was used. For each transformation we fix its colour and the point is coloured with the colour of the transformation that was used to obtain this point.

The convergence of the random algorithm used in the generation of the star-shaped set inversion fractals is guaranteed – similarly to the case of IFS fractals – by the Banach fixed point theorem in which we use the Picard iteration. Because in fixed point theory we have many different iterations that are used for finding fixed points, so in [A9] instead of the Picard iteration the use of various iterations was proposed. Moreover, two modifications of these iterations were proposed. In the iterations the parameters belong to $[0,1]$ or $(0,1]$. Thus, in the first modification the use of parameters outside of these intervals was proposed. The second modification relied on the replacement of the real parameters by the q-system numbers, i.e., numbers that are a generalization of the complex numbers and which were introduced by Levin in [17]. Next, it was shown that in fixed point theory there exist iterations that for inversion do reduce to other iterations irrespective of the parameters used. This is a consequence of the fact that inversion is an involution, i.e., $I_S(I_S(p)) = p$ for each p. An example of this type of iteration is the Schu iteration [32]

$$p_{n+1} = (1 - \alpha_n)p_n + \alpha_n T^{(n+1)}(p_n), \tag{14}$$

which for involution reduces to the Mann iteration. Except the use of various iterations from fixed point theory in [A9] the use of two switching processes was proposed. In the first process we switch the iterations and in the second the q-systems. To colour the fractals other method than the one from [A6] was proposed. In the proposed method we use a colour transformation, the histogram with the information on how many times each pixel was hit and a logarithmic scaling with gamma correction. The presented examples showed that the proposed extensions enrich geometry of the obtained fractals in new, more intriguing details, and the colouring algorithm enrich the aesthetic value. Moreover, in the article it was shown that the use of different iterations in the classical IFS fractals given by the affine mappings does not enrich their geometry, as in the case of inversion fractals, but the fractals become less interesting.

The fraction in the equation of the star-shaped set inversion (13) can be written in the following form:

$$\frac{[d(o,b)]^2}{(x_p - x_o)^2 + (y_p - y_o)^2} = \frac{[d(o,b)]^2}{[d(o,p)]^2} = \left[\frac{d(o,b)}{d(o,p)}\right]^2. \tag{15}$$

In [30] Ramírez et al. studied the properties of the inversion transformation after the change of the Euclidean metric in (15) with the metrics from the family:

$$d_q(a,b) = (|x_a - x_b|^q + |y_a - y_b|^q)^{1/q}, \tag{16}$$

where $a = (x_a, y_a)$, $b = (x_b, y_b)$ and $q \in [1, \infty)$.

Using the idea of the change of the Euclidean metric with the metrics from the d_q family in [A8] it was shown that with the fixed: S – star-shaped set, o – centre of inversion, p – point and b – intersection point, for each $q_1, q_2 \in [1, \infty)$ the following equality:

$$\frac{d_{q_1}(o,b)}{d_{q_1}(o,p)} = \frac{d_{q_2}(o,b)}{d_{q_2}(o,p)}. \tag{17}$$

is true. From this fact it follows that with the fixed parameters of the inversion transformation and the change of the metric the value of the transformation does not change. Thus, the change of only the metric does not change the shape of the inversion fractal. Because the family d_q is a monotonic one with respect to q, so in [A8] it was proposed to use two metrics instead of one. In this way the value of the fraction is smaller or greater, and this changes the value of the inversion transformation. This modification of the inversion transformation causes that the transformation lose some properties and it is no longer inversion. Because of this the modified transformation was called the pseudoinversion transformation, and the fractals generated with it the pseudoinversion fractals. Of course, each of the pseudoinversion transformations that define the fractal can be given by different pair of metrics. In [A8] the use of a switching process of the metrics was also proposed. The presented examples showed that using the pseudoinversion we are able to obtain fractals with different shapes than the ones obtained with the inversion transformations given by the same star-shaped sets. Moreover, the examples showed that the use of different pairs of metrics for different pseudoinversion transformations locally changes the shape of the resulting fractal.

4.3.4. Patterns generated with the help of the orbits of discrete dynamical systems

In mathematical modelling many models use dynamical systems whether in a discrete or continuous version. Except modelling of various phenomena dynamical system can be used to generate aesthetic patterns. Since orbits of discrete dynamical systems are able to create very interesting patterns, so one of the ways to generate aesthetic patterns is the use of such orbits. An example of a transformation that generates a discrete dynamical system with an interesting orbits is the Gumowski-Mira transformation. It was used in a direct way among other things for generating textile patterns [25].

The use of Gumowski-Mira transformation in a way similar to the superfractal algorithm introduced by M. Barnsley was proposed in [B5]. The input for this algorithm are two Gumowski-Mira transformations (with different values of the parameters), probabilities of drawing each of the transformations and a starting point. In each iteration one of the transformations is drawn, and then it is used to transform the orbit's point from the previous iteration (Picard iteration). Such approach allows to obtain geometric patterns which shapes are completely different than the ones obtained with the direct use of one Gumowski-Mira transformation, which for instance was used in [25].

The extension of the algorithm presented in [B5] from two to k Gumowski-Mira transformations was proposed in [B1]. Except the extension to k transformations in [B1] the junction with the help of segments of the consecutive points was proposed. Moreover, to enrich the pure geometry which occurs from the use of the Gumowski-Mira transformation three colouring algorithms were proposed. In the first colouring algorithm the colour was assigned according to the distance of the coloured point from the centre of pattern's bounding rectangle. In the second algorithm the colour was assigned based on the number of iteration in which the coloured point arose. The last algorithm joins features of the first two algorithms.

In [A12] to generate geometrically interesting orbits of discrete dynamical systems the authors – except the Gumowski-Mira transformation – used four further functions. All the functions were used in two algorithms. In the first algorithm it was proposed to use a single dynamical system in a switching process between the Picard iteration and the Krasnoselskij iteration (a special case of the Mann iteration, in which $\alpha_n = const$). The switching is made in a random way – the probability of drawing each of the iterations is equal to $1/2$. In the second algorithm a set of k dynamical systems and a probability distribution of drawing the dynamical system were used. In the algorithm's iteration a dynamical system is drawn and then the point from the previous iteration is transformed using the drawn dynamical system and the Krasnoselskij iteration. To colour the patterns the algorithms from [B1] were used. The presented examples showed that the proposed algorithms have potential in generating aesthetic patterns. Moreover, it was shown that the patterns are very sensitive on the parameter in the Krasnoselskij iteration. The best results were obtained for the values of this parameter in the $[0.99, 1]$ interval.

In [A3] based on the observation that for different starting points in the generation of an orbit we obtain different patterns it was proposed to change the orbits during the iteration process. For this purpose the Picard iteration was modified using the so-called perturbation function. After the modification the iteration process took the following form:

$$x_{n+1} = (f \circ p)(x_n), \tag{18}$$

where $f : \mathbb{R}^2 \to \mathbb{R}^2$ is the function that defines a dynamical system, and $p : \mathbb{R}^2 \to \mathbb{R}^2$ is the perturbation function. In the article several different perturbation functions were proposed. Moreover, it was proposed to use a combination of the standard Picard iteration and the iteration with the perturbation:

$$x_{n+1} = \alpha x'_{n+1} + (1 - \alpha) x''_{n+1}, \tag{19}$$

where $x'_{n+1} = f(x_n)$ (the Picard iteration) and $x''_{n+1} = f(p(x_n))$ (the iteration with the perturbation). The presented examples showed that using the modified iteration processes we are able to obtain patterns different than the ones generated with the help of the Picard iteration. Moreover, it was shown that the introduction of the perturbation function gives many possibilities in the generation of new patterns.

4.3.5. Pattern generated with the help of the dynamics of discrete dynamical systems

Using discrete dynamical systems the aesthetic patterns can be generated not only by the use of their orbits. Other approach that can be used is the use of dynamics of a dynamical system, which is presented as a phase portrait. In the literature we can find many studies on the generation of symmetrical patterns using the dynamics of discrete dynamical systems. In [3] Chung and Chan studied the following dynamical system:

$$\begin{cases} x_{n+1} = x_n - f(x_n, y_n), \\ y_{n+1} = y_n - g(x_n, y_n), \end{cases} \tag{20}$$

where $(x_0, y_0) \in \mathbb{R}^2$ is the staring point, and $f, g : \mathbb{R}^2 \to \mathbb{R}$ are the functions that define the system. In their work Chung and Chan presented conditions under which the phase portrait of this dynamical system will have different types of symmetries: translation, rotational, glide, wallpaper etc.

In [A2] the results from [3] were expanded. It was observed that the system (20) uses the Picard iteration, i.e., $p_{n+1} = T(p_n)$, where T is given by the following formula:

$$T(p) = \begin{pmatrix} x_p - f(p) \\ y_p - g(p) \end{pmatrix}, \tag{21}$$

where $p = (x_p, y_p)$. Similar to polynomiography or inversion fractals in [A2] it was proposed to use other iterations from the fixed point theory. Moreover, it was proposed to expand the range of the parameters outside the $(0, 1]$, $[0, 1]$ intervals and to introduce vector parameters instead of the scalar ones that were used in the iterations. Next, example conditions that the functions f and g must satisfy in order to obtain a translation symmetry along the x axis of the phase portraits using the Mann and the S-iteration were introduced. The next proposed modification of the iteration process was the use of an affine combination of iterations with different functions f and g. In order to enrich the patterns some convergence tests were also proposed. In [A2] the possibility of implementing the proposed modifications using the shaders written in GLSL was also discussed. The presented examples showed that with the help of the proposed modifications we are able to generate symmetrical patterns which we have not been able to generate using the method from [3]. Moreover, a comparison of the generation times for two algorithm's implementation was made. The first implementation was made in the Processing language (the CPU implementation) and the second in the GLSL (the GPU implementation). From the comparison it results that using the shaders we can obtain a speed-up from 1000 to 2500 times compared to the CPU implementation.

5. Review of other scientific achievements

Other publications

[C1] GDAWIEC, K. Pseudofractal 2D shape recognition. In *Rough Set and Knowledge Tech-*

nology, J. Yu, S. Greco, P. Lingras, G. Wang, and A. Skowron, Eds., vol. 6401 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin Heidelberg, 2010, pp. 403–410. DOI: 10.1007/978-3-642-16248-0_57
MNiSW: 13 pts.

[C2] GDAWIEC, K., AND DOMAŃSKA, D. Partitioned iterated function systems with division and a fractal dependence graph in recognition of 2D shapes. *International Journal of Applied Mathematics and Computer Science 21*, 4 (2011), 757–767. DOI: 10.2478/v10006-011-0060-8
IF: 0.487, MNiSW: 20 pts.

[C3] GDAWIEC, K., AND DOMAŃSKA, D. Recognition of two-dimensional shapes based on dependence vectors. In *Artificial Intelligence and Soft Computing*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds., vol. 7267 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin Heidelberg, 2012, pp. 501–508. DOI: 10.1007/978-3-642-29347-4_58
MNiSW: 10 pts.

[C4] KOTARSKI, W., GDAWIEC, K., AND LISOWSKA, A. Nieliniowe podziały i fraktale. In *Systemy wspomagania decyzji*, A. Wakulicz-Deja, Ed. Instytut Informatyki Uniwersytetu Śląskiego, Katowice, 2010, pp. 363–371.
MNiSW: 3 pts.

5.1. Impact of iterations on the complex polynomial roots finding methods

Besides the research on the aesthetic pattern generation methods in polynomiography some other studies on the numerical properties of the proposed methods were made.

In [A13] a visual analysis of the impact of the iteration parameters values on the convergence of the root finding method was performed, and in [A11] a similar analysis was made for the pseudo-methods. The analyses showed that using different iterations instead of the Picard one we are able to improve algorithm's convergence, but we are not able to explicitly give the values that are good in every case. Depending on the root finding method or the pseudo-method and the polynomial the best iteration and the values of the parameters are different.

In [A1, A10, A13] an analysis of the impact of the parameters that are used in the iteration processes on the generation time of the polynomiograph was made. In [A13] the analysis was made for the parameters of different iterations for a single root finding method, in [A10] for the parameters of the different switching processes, and in [A1] for the parameters of an affine combination of the root finding methods and different iterations that use several root finding methods. The analyses showed that the dependency of the generation time on the parameters is mostly non-trivial, strongly non-linear and non-monotonic.

In [A1] except the studies on the generation times of different polynomiographs an analysis of the impact of the parameters used in the affine combination of root finding methods and in different iterations that use several root finding methods on different numerical measures was made. The analysis was made for the following numerical measures: the convergence area index, the average number of iterations, the fractal dimension of the boundaries of basins of attraction and the Wada measure of the basins of attraction. Similar to the case of the generation times, the dependency in the case of every measure is non-trivial, strongly non-linear and non-monotonic.

5.2. Fractal methods of 2D shapes recognition

In [C2] to introduce a new method of fractal recognition of 2D shapes the following observation was used: when we divide the image into sub-images and use a fractal compression on the independent sub-images the resulting fractal description which consists of the set of partitioned iterated function systems (PIFS) is less sensitive on the changes of the shape. The set of the PIFSs obtained in this way is called PIFS with division. In the article it was proved that the space of PIFSs with division with appropriate metric is a metric space and that the input image is an attractor of a PIFS with division. Using PIFS with division and a dependence graph computed from this PIFS with division a method of fractal dependence graph was proposed. The tests of the effectiveness of the method were conducted on two bases with 2D shapes images. The first base was created by the authors and consists from three test sets: (1) shapes changed by elementary transformations, i.e., rotation, translation and scaling, (2) shapes altered by locally small changes together with the transformation with the elementary transformations, (3) shapes altered by locally large changes together with the transformation with the elementary transformations. The second base was the standard MPEG7 CE-Shape-1 Part B base. The tests showed that the fractal dependence graph method obtains higher accuracy than the other fractal methods known in the literature.

In [C1] another drawback of the fractal image compression used as the base for the fractal features representing a shape was observed. It was observed that when we create the dictionary for the fractal compression even the change of a single element (a shape change in this area) of this dictionary can significantly influence the resulting PIFS. In the pessimistic case the whole system can change. To eliminate this drawback it was proposed to use – in the fractal image compression – a pseudofractal approach. This approach is based on a dictionary that is not created from the original image that is compressed, but from a fixed image that is selected specially for this aim. In this way the change of object's shape does not have impact on the resulting PIFS. The proposed pseudofractal approach was used in the fractal dependence graph method. The tests showing the effectiveness of the proposed approach were made on the same bases as in [C2]. The tests showed that the use of the pseudofractal approach reduces the recognition error in comparison to the methods that are based on PIFS and PIFS with division.

The pseudofractal approach was also used in [C3], where another method of fractal recognition of 2D shapes was introduced. In this method as the fractal features the set of dependence vectors was used. Dependence vectors for a range block R are created by vectors between R and the range blocks that intersect the domain block D that corresponds to R. To compute the similarity between objects the Haussdorf metric on the sets of dependence vectors was used. In the tests aside from the bases used in [C1, C2] the Kimia-99 and Kimia-216 bases were used. The tests showed that the use of the pseudofractal approach and the dependence vectors gives the smallest recognition error among the 2D object fractal recognition methods.

5.3. Non-linear subdivision and fractals

In [C4] generalizations of the subdivision technique were presented. Application of these generalizations gives possibilities of generating of smooth graphical objects such as curves, patches or fractal objects that are given by a set of starting points. The generalizations go in two directions. The first one introduces a complex parameter into a linear subdivision and the second non-linearity with the use of averages other than the arithmetic average. The discussed subdivision generalizations expanded in a significant way the class of graphical objects which can be generated with the help of linear subdivision. Subdivision remain in a close connection with fractal methods, because with their help and the given set of control points one can define IFSs used in the fractal rendering of graphical objects. In the article the connection between subdivision and fractals was also shown. Moreover, some applications of the subdivision method were presented.

References

[1] ANDERSON, D., AND WOOD, Z. User driven two-dimensional computer-generated ornamentation. In *Advances in Visual Computing: 4th International Symposium, ISVC 2008, Las Vegas, NV, USA, December 1-3, 2008. Proceedings, Part I*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. Chun, T.-M. Rhyne, and L. Monroe, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 604–613. DOI: 10.1007/978-3-540-89639-5_58.

[2] CHEN, Y.-S., SHIE, J., AND CHEN, L.-H. A NPR system for generating floral patterns based on l-system. *Bulletin of Networking, Computing, Systems, and Software 1*, 1 (2012), 38–41.

[3] CHUNG, K., AND CHAN, H. Symmetrical patterns from dynamics. *Computer Graphics Forum 12*, 1 (1993), 33–40. DOI: 10.1111/1467-8659.1210033.

[4] DAS, G., AND DEBATA, J. Fixed points of quasinonexpansive mappings. *Indian Journal of Pure and Applied Mathematics 17*, 11 (1986), 1263–1269.

[5] DRAVES, S. The electric sheep and their dreams in high fidelity. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2006), NPAR '06, ACM, pp. 7–9. DOI: 10.1145/1124728.1124730.

[6] FEIJS, L., AND TOETERS, M. A novel line fractal pied de poule (houndstooth). In *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture* (Phoenix, Arizona, 2015), K. Delp, C. Kaplan, D. McKenna, and R. Sarhangi, Eds., Tessellations Publishing, pp. 223–230.

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

[8] GREENFIELD, G. Turing-like patterns from cellular automata. In *Proceedings of Bridges 2016: Mathematics, Music, Art, Architecture, Education, Culture* (Phoenix, Arizona, 2016), E. Torrence, B. Torrence, C. Séquin, D. McKenna, K. Fenyvesi, and R. Sarhangi, Eds., Tessellations Publishing, pp. 151–158.

[9] ISHIKAWA, S. Fixed points by a new iteration method. *Proceedings of the American Mathematical Society 44*, 1 (1974), 147–150. DOI: 10.1090/S0002-9939-1974-0336469-5.

[10] JIA, C., AND MING-XI, T. Integrating shape grammars into a generative system for Zhuang ethnic embroidery design exploration. *Computer-Aided Design 45*, 3 (2013), 591–604. DOI: 10.1016/j.cad.2012.08.002.

[11] JIANG, C., TANG, C., VAXMAN, A., WONKA, P., AND POTTMANN, H. Polyhedral patterns. *ACM Transactions on Graphics 34*, 6 (2015), Article No. 172, 12 pages. DOI: 10.1145/2816795.2818077.

[12] KALANTARI, B. Polynomiography and applications in art, education and science. *Computers & Graphics 28*, 3 (2004), 417–430. DOI: 10.1016/j.cag.2004.03.009.

[13] KALANTARI, B. Method of creating graphical works based on polynomials, 2005. US patent 6894705 B1.

[14] Kalantari, B. A necessary and sufficient condition for local maxima of polynomial modulus over unit disc, 2016.

[15] Klempien-Hinrichs, R., and von Totth, C. Generation of celtic key patterns with tree-based collage grammars. *Electronic Communications of the EASST 26* (2010), 16 pages. DOI: 10.14279/tuj.eceasst.26.356.

[16] Lalitha, D., and Rangarajan, K. Petri nets generating Kolam patterns. *Indian Journal of Computer Science and Engineering 3*, 1 (2012), 68–74.

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

[18] Loi, H., Hurtut, T., Vergne, R., and Thollot, J. Programmable 2D arrangements for element texture design. *ACM Transactions on Graphics 36*, 3 (2017), Article No. 27, 17 pages. DOI: 10.1145/2983617.

[19] Lu, J., Ye, Z., and Zou, Y. Automatic generation of colorful patterns with wallpaper symmetries from dynamics. *The Visual Computer 23*, 6 (2007), 445–449. DOI: 10.1007/s00371-007-0116-9.

[20] Lu, S., Mok, P., and Jin, X. From design methodology to evolutionary design: An interactive creation of marble-like textile patterns. *Engineering Applications of Artificial Intelligence 32* (2014), 124–135. DOI: 10.1016/j.engappai.2014.02.015.

[21] Lutton, E. Evolution of fractal shapes for artists and designers. *International Journal on Artificial Intelligence Tools 15*, 4 (2006), 651–672. DOI: 10.1142/S0218213006002850.

[22] Machado, P., Romero, J., Santos, A., Cardoso, A., and Pazos, A. On the development of evolutionary artificial artists. *Computers & Graphics 31*, 6 (2007), 818–826. DOI: 10.1016/j.cag.2007.08.010.

[23] Machnik, G., Chodacki, M., and Kotarski, W. Using genetic algorithm to aesthetic patterns design. In *Computational Collective Intelligence: 8th International Conference, ICCCI 2016, Halkidiki, Greece, September 28-30, 2016. Proceedings, Part II*, N. Nguyen, L. Iliadis, Y. Manolopoulos, and B. Trawiński, Eds. Springer International Publishing, Cham, 2016, pp. 123–132. DOI: 10.1007/978-3-319-45246-3_12.

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

[25] Naud, M., Richard, P., Chapeau-Blondeau, F., and Ferrier, J.-L. Automatic generation of aesthetic images for computer-assisted virtual fashion design. In *10th Generative Art Conference* (Milan, Italy, 2007), Domus Argenia Publisher, pp. 191–200.

[26] Ouyang, P., Zhao, W., and Huang, X. Beautiful math, part 5: Colorful Archimedean tilings from dynamical systems. *IEEE Computer Graphics and Applications 35*, 6 (2015), 90–96. DOI: 10.1109/MCG.2015.135.

[27] Pang, W., and Hui, K. Interactive evolutionary 3d fractal modeling. *The Visual Computer 26*, 12 (2010), 1467–1483. DOI: 10.1007/s00371-010-0500-8.

[28] Pickover, C. Biomorphs: Computer displays of biological forms generated from mathematical feedback loops. *Computer Graphics Forum 5*, 4 (1986), 313–316. DOI: 10.1111/j.1467-8659.1986.tb00317.x.

[29] Qi, W., and Li, X. Example-based floral pattern generation. In *Proceedings of the Fifth International Conference on Image and Graphics* (Los Alamitos, 2009), Y.-J. Zhang, Ed., IEEE Computer Society, pp. 553–558. DOI: 10.1109/ICIG.2009.176.

[30] Ramírez, J., Rubiano, G., and Zlobec, B. Generating fractal patterns by using p-circle inversion. *Fractals 23*, 4 (2015), 1550047. DOI: 10.1142/S0218348X15500474.

[31] Sayed, Z., Ugail, H., Palmer, I., Purdy, J., and Reeve, C. Auto-parameterized shape grammar for constructing Islamic geometric motif-based structures. In *Transactions on Computational Science XXVIII: Special Issue on Cyberworlds and Cybersecurity*, M. Gavrilova, C. K. Tan, and A. Sourin, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 146–162. DOI: 10.1007/978-3-662-53090-0_8.

[32] 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.

[33] Setti, R. Generative dreams from deep belief networks. In *Generative Art 2015: Proceeding of XVIII Generative Art Conference* (Milan, 2015), C. Soddu and E. Colabella, Eds., Domus Argenia Publisher, pp. 260–273.

[34] Shier, J., and Bourke, P. An algorithm for random fractal filling of space. *Computer Graphics Forum 32*, 8 (2013), 89–97. DOI: 10.1111/cgf.12163.

[35] Sisson, P. Fractal art using variations on escape time algorithm in the complex plane. *Journal of Mathematics and the Arts 1*, 1 (2007), 41–45. DOI: 10.1080/17513470701210485.

[36] Soo, S., Yu, K., and Chiu, W. Modeling and fabrication of artistic products based on IFS fractal representation. *Computer-Aided Design 38*, 7 (2006), 755–769. DOI: 10.1016/j.cad.2006.04.003.

[37] Szyszkowicz, M. Computer art from numerical methods. *Computer Graphics Forum 10*, 3 (1991), 255–259. DOI: 10.1111/1467-8659.1030255.

[38] Wannarumon, S., Bohez, E., and 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.

[39] Wannarumon, S., Unnanon, K., and Bohez, E. Intelligent computer system for jewelry design support. *Computer-Aided Design and Applications 1*, 1–4 (2004), 551–558. DOI: 10.1080/16864360.2004.10738298.

[40] Xu, S., Yang, J., Wang, Y., Liu, H., and Gao, J. Application of fractal art for the package decoration design. In *2009 IEEE 10th International Conference on Computer-Aided Industrial Design Conceptual Design* (2009), pp. 705–709. DOI: 10.1109/CAIDCD.2009.5375057.

[41] Yeh, Y.-T., Breeden, K., Yang, L., Fisher, M., and Hanrahan, P. Synthesis of tiled patterns using factor graphs. *ACM Transactions on Graphics 32*, 1 (2013), Article No. 3, 13 pages. DOI: 10.1145/2421636.2421639.

Krzysztof Gdawiec