GENERAL ARTICLE

Interactive Design of Random Aesthetic Abstract Textures by Composition Principles

LUIS ALVAREZ, NELSON MONZÓN AND **JEAN-MICHEL MOREL**

ABSTRACT

The automatic synthesis of abstract textures is, to some extent, feasible. As evidenced by abstract art theoreticians, one can think of an abstract picture as a tree of elementary shapes interacting according to a short list of compositional laws such as occlusion, exclusion and bordering, and by rendering rules such as transparency, tessellation and color selection. Randomizing the shape generator and the composition and rendering laws yields an algorithm generating random abstract textures. We have designed a user-friendly online tool that implements this algorithm.

In 1917, Jean Arp discussed the incorporation of chance into the principles of artistic creation in the following terms: "We want to produce directly and without meditation. . . . I wanted to create new appearances, to extract new forms from man" [1].

He wrote further:

> The forms arrive pleasant, or strange, hostile, inexplicable, mute, or drowsy. They are born from themselves. It seems to me as if all I do is move my hands. . . . The proportion of these planes and their colors seemed to depend only upon chance, and I declared that these works were ordered "according to the law of chance." [2]

Arp explored these ideas in creating abstract collages. Figure 1 shows how he casually displayed basic shapes on a canvas.

Our goal is to describe a random image creation program using a formal approach and an interactive online computing tool.

Luis Alvarez (mathematician, educator), CTIM, Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, 35017, Spain. Email: lalvarez@ulpgc.es. Web: www.sites.google.com/site/luisalvarezsite. ORCID: 0000-0002-6953-9587.

Nelson Monzón (computer engineer, educator), CTIM, Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, 35017, Spain. Email: nelson.monzon@ulpgc.es. Web: www.linkedin.com/in/nelsonmonzonlopez. ORCID: 0000-0003-0571-9068.

Jean-Michel Morel (mathematician, educator), Centre Borelli, 4 avenue des Sciences, Université Paris-Saclay, 91190 Gif-sur-Yvette, France. Email: moreljeanmichel@gmail.com. Web: www.sites.google.com/site/jeanmichelmorelcmlaenscachan. ORCID: 0000-0002-6108-897X.

See https://direct.mit.edu/leon/issue/54/2 for supplemental files associated with this issue.

The objective of our design facility is to explore randomly all ways to combine shapes on a surface. The result of each random trial is what we call an aesthetic abstract texture (AAT). Our online interface allows an easy way for artists and designers to create/modify their own AAT styles. The interface's main challenge was to define an adequate organization of the parameters governing an AAT. Texture parameterization should be powerful enough to generate a large

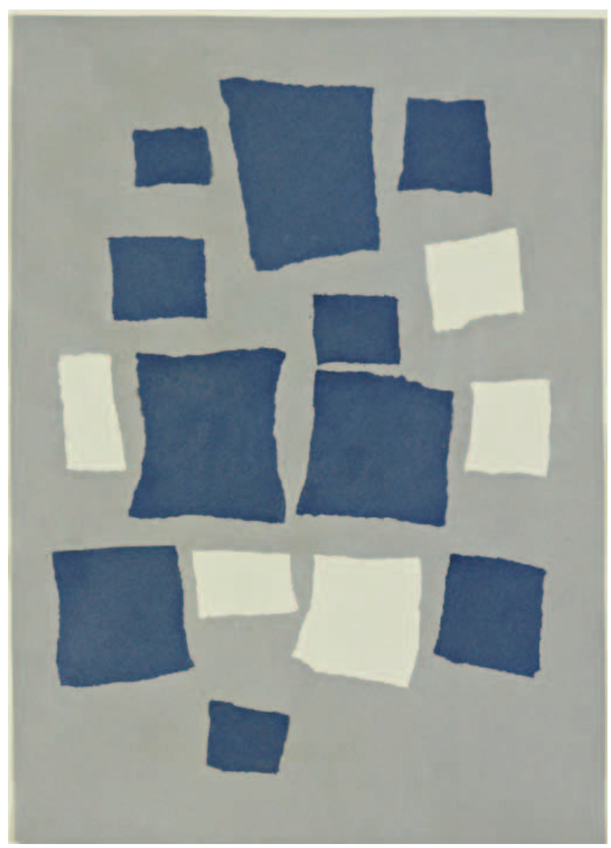

Fig. 1. Jean Arp, *Collage with Squares Arranged According to the Laws of Chance*, torn-and-pasted paper and colored paper, 19⅛ × 13⅝ in, 1917. (Public domain)

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

Fig. 2. Art at Cueva de las Manos, Argentina, ca. 9,000 years ago. (Photo © Marianocecowski. Source: Wikimedia Commons.)

variety of new "appearances." At the same time, it should be simple enough to be easily understandable and modifiable by a user without programming skills. We have jointly designed the grammar organizing the texture and the online interface, which we invite the reader to test at www.ipol.im/aat.

The art at Cueva de las Manos (Fig. 2) is one of the first known human creations displaying random shape distributions. It illustrates well the problem of abstract random painting: The hands were disposed randomly but with the intention to cover the whole space. The orientation of the hands is randomly distributed, and they are combined through occlusion or transparency. These basic composition rules are shared by all abstract artists, and we use them to generate our AATs, among other objectives.

Ideally, the generator should authorize all possible shapes and shape combination rules or, at the very least, those observed in abstract painting and decorative arts. This may seem an impossible task. Fortunately, abstract art has formalized the painting generation problem, in an effort comparable with parallel formalizations of mathematical logic or Gestalt theory.

Abstract painting theoreticians [3,4] were indeed confronted with three new problems that were only implicitly resolved in figurative painting. One of these problems was to invent shapes. Another was to combine them on a flat surface, and a third was to attribute to them a color. Noticeably, abstract painting did not abandon the idea that basic elements are shapes. There are exceptions, such as Jackson Pollock's endeavor to entrust shape formation to physical processes like spilling.

The basic abstract shapes often were the simplest geometric shapes, such as polygons, circles, smooth curves or thick strokes. This simplification was not considered a serious drawback by the likes of Wassily Kandinsky, Paul Klee, František Kupka or Kazimir Malevich. Indeed, using the simplest basic shapes leaves many degrees of freedom to the painter, who can still choose their size, color, orientation and relative disposition. Nevertheless, abstract painting faces the same physical constraints as figurative painting. The addition of a new shape with its color on the canvas can be either subtractive or additive. In the subtractive case, the new shape hides its background; in the additive case, the color of the new shape is combined with the background's colors. This creates the classic transparency effect, caused in the physical world by light shafts, shadows or glasses.

Our facility combines basic abstract shapes by randomly applying classic shape composition laws. In that way, the styles of many abstract painters can be casually retrieved, because their parameters are straightforward shape combination techniques as in Figs 1 and 2.

Our algorithm extends numerous algorithms reproducing specific styles by mimicking the shapes and composition rules used by abstract painters. For example, Kirsch and Kirsch [5] proposed algorithms to emulate works by Richard Diebenkorn and Joan Miró. Tao, Liu and Zhang [6] reproduce Malevich's style, Zhang and Yu [7] Kandinsky's style, and Taylor, Micolich and Jonas [8] Pollock's style.

Barla et al. [9] proposed to create new images through random spatial arrangements of specific patterns, a task that requires modeling shape interactions. Similarly, we do not aim to reproduce any particular painter's style. Our goal is to enable artists and designers to test new styles, based on random combinations of shapes, composition and rendering rules. We presented the mathematical description of a texture generator based on an organization in successive layers [10]. As we argue below, the tree organization that we propose in this article is far more general. We show a number of collections of textures made with the AAT generator [11].

Shapes: Beyond very basic abstract shapes like circles, rectangles, triangles, etc., we have added more sophisticated ones to the generator, for instance, rosettes used from antiquity as decorative ornaments, or active lines as Klee described: "An active line on a walk, moving freely, without goal. A walk for a walk's sake" [12].

Composition laws: In addition to basic shape combination effects using *occlusion* and *transparency*, our algorithm authorizes *tessellation*, *light spot* and shape *contouring*. *Tessellation*, used by artists like Klee, Robert Delaunay, Joaquín Torres-García or Bruce Gray, is based on the assignment of independent colors to each connected component generated by the boundaries of intersecting shapes. *Light spot* consists of an attenuation of the color's intensity according to the distance to a point named the light spot center. *Contouring* shape boundaries in black is a classic boundary enhancement technique used in cartoons and by artists like Mondrian or Picasso. It produces a stained-glass effect. To illustrate the bridge between abstract and decorative art, our algorithm also authorizes a setup of the image similar to a scarf or a Persian carpet where several different AATs build successive borders surrounding a central AAT.

In general, digital abstract image synthesis requires programming skills and tools. This is the case for the Processing programming language [13] developed for artists and designers. The potential of creative coding can also be improved using code bending [14]. Design using our online facility does

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

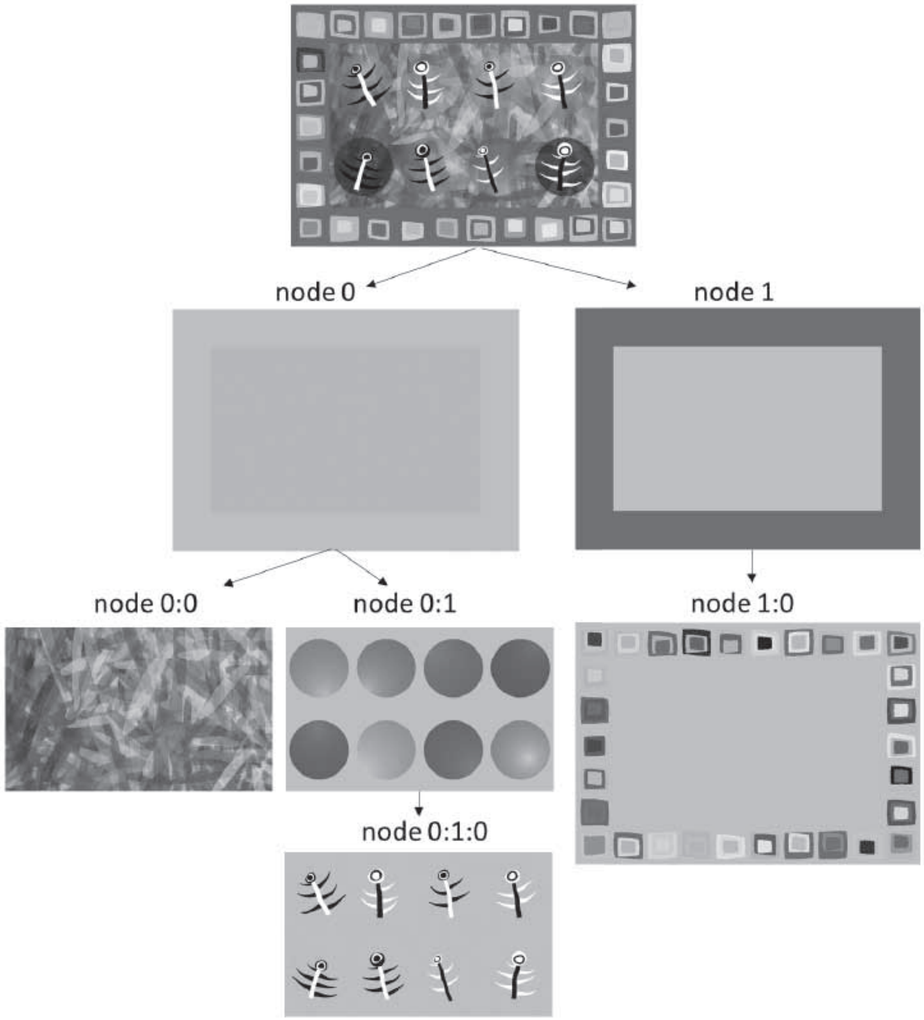

Fig. 3. Illustration of the tree data structure of an aesthetic abstract texture (AAT). In each node, composition rules, shapes and rendering properties are fixed randomly and independently. (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel)

not require programming skills. We propose an experiential learning model. In seconds, users can create new AATs by editing one of a predefined collection of AATs or by creating one from scratch using the automatic random creation mode. A full understanding of the AAT parameters can be achieved through experiments, in an hour's training given moderate computer skills (this estimate is based on the observation of the training of two scientists and three designers).

FORMALIZATION OF THE AAT DESIGN

To formalize the design of an AAT, we argue that the painting's or image's natural organization is a tree where the root is the whole image and the general construction rule is to combine new shapes inside—or around—a parent shape. Thus any painting can be formalized as a tree of shapes. What differentiates the result is the choice of shapes, the choice of colors and the shape composition rules at each tree node. Hence, an AAT is specified when a tree data structure is given and, at each node shape, color and composition rules are specified.

In the AAT example presented in Fig. 3, the first level of the tree is composed of the image subdivision given the central rectangle and one border strip band. The next levels of the tree specify, for each node, the way it is filled with child shapes respecting a number of composition rules. The user never manually specifies the choice and location of a shape in the AAT. The shapes are automatically sorted in the parent shape following composition rules and random sampling.

In this tree organization an intuitive interface can easily modify the texture parameters. Roughly speaking, at each tree node the generator chooses by random sorting (and the user can manually modify each choice):

1. The kind of basic shape used in the node.
2. The spatial relation of the shapes with respect to the parent node. Shapes can be distributed in several random distribution rules or following a choice of regular distributions inside or around the parent shape.
3. Geometric transformations: rotation, scaling, etc., applied to the basic shapes to obtain the shapes to include in the texture.
4. Shape interaction: The shapes can be allowed or forbidden to intersect, to exceed the parent shape, to meet the boundary of the parent shape, etc.
5. Coloring: Each shape is attributed a random color taken from a palette, possibly a contour and a light spot.
6. The shape color's rendering with respect to former shapes: transparency, tessellation, occlusion.

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

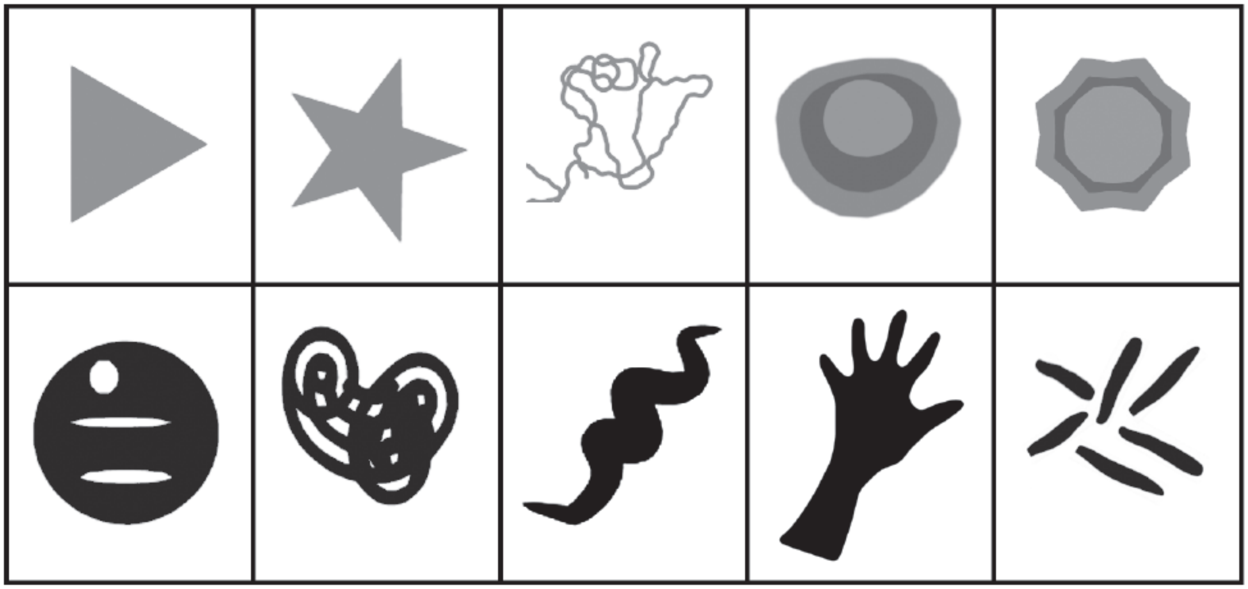

Fig. 4. Illustration of some shapes generated automatically (top row) and some obtained from silhouettes (bottom row). (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel)

AAT Generation Configuration

There are two ways to select the basic shapes:

1. Automatic shape generation. The algorithm automatically generates basic shapes such as triangles, circles and rectangles or more complex ones such as rosettes, Pollock-like active lines, Kandinsky eyes, etc.
2. The user can increase at will the list of basic shapes: Shapes can be obtained from a silhouette drawn in an image (Fig. 4). In the online interface, users can pick a single shape or a collection of shapes.

We also need to define the way the shapes are distributed with respect to the shapes in the parent node. In the current version of the generator, shapes can be distributed as follows:

1. Random distribution: The locations of the shapes are distributed randomly within the shapes of the parent nodes.
2. Regular distribution: The shapes are distributed inside the parent shapes following a regular lattice.
3. Contouring: The shapes are distributed along the boundary of the parent shapes.
4. Mondrian type: The parent shape is filled using Mondrian-type rectangle distribution. (We chose this more peculiar distribution to illustrate the versatility of the designing structure.)

Alternatively, Loi et al. [15] have proposed shape distributions based on partitioning, mapping and merging operators.

Once the way the shapes are distributed is fixed, we define a default shape diameter that determines the default shape size and the number of shapes. The number of shapes is fixed to cover the parent node area using shapes with the default shape size. The lower the default shape diameter, the larger the number of shapes to include.

The generator applies a random geometric affine map to the selected basic shape database to generate new shapes. Affine maps are parameterized by the following 2×2 matrix that composes four intuitive operations: a rotation, a horizontal squeeze, a second rotation of the squeezed shape and a zoom:

$$\mathbf{A} = \mathbf{s} \begin{pmatrix} \cos \boldsymbol{\alpha}_2 & \sin \boldsymbol{\alpha}_2 \\ -\sin \boldsymbol{\alpha}_2 & \cos \boldsymbol{\alpha}_2 \end{pmatrix} \begin{pmatrix} \mathbf{s}_x & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \boldsymbol{\alpha}_1 & \sin \boldsymbol{\alpha}_1 \\ -\sin \boldsymbol{\alpha}_1 & \cos \boldsymbol{\alpha}_1 \end{pmatrix}$$

For each shape the algorithm randomly selects the parameters of the affine transformation in a reasonably wide range of values (which, again, the user can modify).

We also use the following shape interaction rules when including new shapes in the tree:

- Inclusion with respect to parent shape: A new shape is added only if it is included in the parent shape.
- Exclusion with respect to brothers: New shapes are added only if they do not intercept brother shapes associated to the same parent shape.
- Exclusion with respect to other shapes in the node: A new shape is added only if it does not intercept other shapes in the same node level but is associated to different shapes.

Once the shape tree structure is created, a crucial issue is the way it is rendered to create the AAT result. The following rendering configurations provide a variety of rendering options (which are again chosen randomly and can be modified manually with the online interface).

Texture Rendering Configuration

- Color palette: Associates with each shape a random color in a palette given by an image.
- Transparency: Assigns each shape a transparency factor.
- Light spot: When activated, causes the intensity of the colors in the points of the shape to attenuate according to the distance to a point sorted inside the shape.
- Tessellation: Uses image tessellation generated by the object boundaries to render the texture "behind" the

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

new drawn shape. Colors are associated independently to each new connected component of the image.
- Shape border delineation: Optionally delineates a black stripe border along the shape contour.
- Drawing child shape outside its parent: Decides whether a shape extends outside its parent shape.

INTERACTIVE ONLINE AAT GENERATOR

The AAT generator is fully automatic. This means that clicking "run" causes a random AAT to be generated. An AAT is both a texture and a texture style. Indeed, the same AAT can be "rerun" by changing any of its parameters but also only its seed random parameters. Then a "similar" but distinct image will be generated.

We wanted users to create and modify AATs easily and intuitively. To develop this online interface we used the Image Processing On Line (IPOL) facilities. IPOL is a journal enabling the creation of sophisticated online interfaces well adapted to many image processing tasks. All information required to reproduce an AAT is stored in a parameter configuration file that can be edited online using the interface. The usual starting point for a user is to select a particular AAT model among those proposed as examples or to automatically generate a new AAT by clicking "run." Once the AAT model is created, users can modify it in two ways: They can modify directly the text parameter file (expert mode) or they can modify the most significant texture parameters using buttons and sliders in the interface. These parameters are the following:

- The color palette used to color the texture.
- The basic shapes used to generate the AAT (users can upload their own collection of shapes given by silhouettes).
- Resolution: By changing the resolution, users change the texture scale (the lower the resolution, the smaller the shapes).
- Reshuffle shapes: This option generates a similar AAT changing the random shape positions.
- Reassign colors: Without changing the palette, the assignment of colors to shapes is randomly modified.
- Change drawing order: Change the order in which the shapes are drawn.

We show at right a variety of AATs obtained with the online interface. The examples of Figs 5–7 were not modified by a user; they result from a random parameter choice. Color Plate B is the result of modifications using the online facility; in these examples we can easily recognize elements from the styles of Mondrian, Pollock, Miró and Kandinsky. These textures can include elements of known styles but more notably the combination of shapes and composition rules in a tree organization allows the creation of new styles.

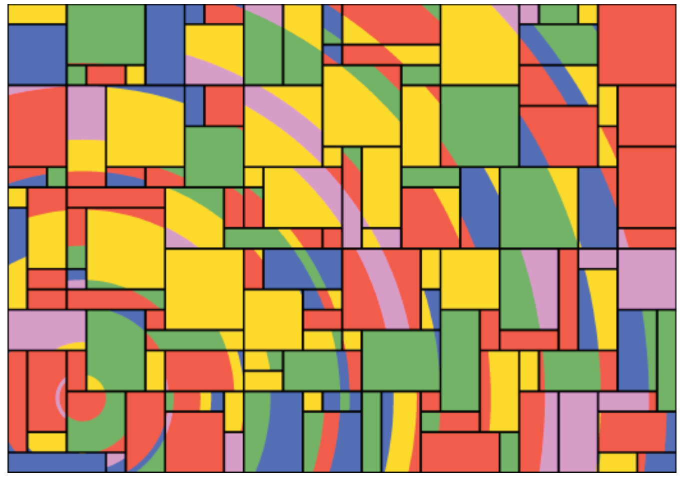

Fig. 5. AAT with recognizable elements of Mondrian's style. (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel)

Fig. 6. AAT with recognizable elements of Pollock's style. (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel

Fig. 7. AAT with recognizable elements of Miró's style. (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel)

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

CONCLUSION

In this article we presented the elements of a formal structure describing aesthetic abstract textures (AAT). Each texture is organized as a collection of composition rules, shapes and rendering properties stored using a data tree structure. The advantage of this approach is that the information is simple to manage and modify but at the same time powerful enough to create a large variety of textures implicitly containing many different styles. We have devoted significant attention to the design of a user-friendly online interface, www.ipol.im/aat, to create/modify textures using this formal organization. Artists and designers without programming skills can use this online facility to create/modify their own AATs. This online tool does not pretend to emulate artistic creation or to mimic existing paintings. It simply helps users explore the technical constraints and degrees of freedom of any combination of shapes and colors. When some superficial similarity occurs between an AAT and an existing abstract painting, this is attributable to the abstract technique having been well emulated. This does not imply that an AAT is an artwork, of course.

We conclude this description with a number of limitations on our approach. We are aware that a much more complete shape generator should be built, based on abstract principles. Complex shapes could be constructed by combining elementary shapes with concatenation rules such as bordering, periodization, symmetry, recursive tree structures, etc., as developed, for example, in Klee's lecture notes [16].

We have not yet implemented all reasonable image-tiling schemes. For example, recursive subdivision schemes or random tessellations with straight lines are popular ways to subdivide a shape. We only included a Mondrian-like subdivision scheme. A last caveat is that the textual tree structure does not completely describe the AAT: A drawing order must be specified, and by default it is the order in which the tree's nodes are read.

Could our random image generator create realistic images? From a probabilistic viewpoint, realistic painting is improbable. Indeed, the natural forms created by geological or biological processes and by humans are too specific. Thus, realistic compositions are very unlikely with our generator, while familiar abstract setups do arise spontaneously. Certain natural textures are, however, prone to random mathematical modeling [17].

References and Notes

1 Jean Arp, *Jours effeuillés: Poèmes, essais, souvenirs 1920–1965* (Gallimard, 1966), p. 183.

2 Arp [1] p. 307.

3 Paul Klee, *Pedagogical Sketchbook* (New York: Praeger Publishers, 1972).

4 Wassily Kandinsky, *Punkt und Linie zu Fläche, Beitrag zur Analyse der malerischen Elemente* (Munich: Verlag Albert Langen, 1926).

5 Joan L. Kirsch and Russell A. Kirsch, "The Anatomy of Painting Style: Description with Computer Rules," *Leonardo* **21**, No. 4, 437–444 (1988).

6 Wenyuan Tao, Yaxuan Liu and Kang Zhang, "Automatically Generating Abstract Paintings in Malevich Style," *Proceedings of the 2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)* (Taiyuan, Chinam 2014) pp. 201–205.

7 Kang Zhang and Jinhui Yu, "Generation of Kandinsky Art," *Leonardo* **49**, No. 1, 48–54 (2016).

8 Richard P. Taylor, Adam P. Micolich and David Jonas, "The Construction of Jackson Pollock's Fractal Drip Paintings," *Leonardo* **35**, No. 2, 203–207 (2002).

9 Pascal Barla et al., "Stroke Pattern Analysis and Synthesis," *Computer Graphics Forum* **25**, No. 3, 663–671 (2006).

10 Luis Alvarez et al., "Exploring the Space of Abstract Textures by Principles and Random Sampling," *Journal of Mathematical Imaging and Vision* **53**, No. 3, 332–345 (2015).

11 Luis Alvarez and Jean-Michel Morel, "Image Design by Principles and Random Sampling": www.ctim.es/ImageSynthesis.

12 Klee [3] p. 17.

13 Casey Reas and Ben Fry, *Processing: A Programming Handbook for Visual Designers and Artists* (Cambridge, MA: MIT Press, 2007).

14 Ilias Bergstrom and R. Beau Lotto, "Code Bending: A New Creative Coding Practice," *Leonardo* **48**, No. 1, 25–31 (2015).

15 Hugo Loi et al., "Programmable 2D Arrangements for Element Texture Design," *ACM Transactions on Graphics* **36**, No. 3, 27 (2017).

16 Klee [3].

17 Carola-Bibiane Schönlieb and Franz Schubert, "Random Simulations for Generative art Construction—Some Examples," *Journal of Mathematics and the Arts* **7**, No. 1, 29–39 (2013).

Manuscript received 2 July 2018.

LUIS ALVAREZ *is a professor at the Universidad de Las Palmas de Gran Canaria, Spain. His areas of research are computer vision and mathematics.*

NELSON MONZÓN *is a postdoctoral researcher and software engineer at the Universidad de Las Palmas de Gran Canaria, Spain. His main scientific interests are computer vision and image/video processing.*

JEAN-MICHEL MOREL *is a professor of mathematics at École Normale Supérieure Paris-Saclay, France. His areas of research are image processing and computer vision.*

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021

COLOR PLATE B: **INTERACTIVE DESIGN OF RANDOM AESTHETIC ABSTRACT TEXTURES BY COMPOSITION PRINCIPLES**

Aesthetic Abstract Texture (AAT) with recognizable elements of Kandinsky's style (*Color Study: Squares with Concentric Circles*, 1913). This one was modified manually in the online interface. (© Luis Alvarez, Nelson Monzón and Jean-Michel Morel) (See the article in this issue by Luis Alvarez, Nelson Monzón and Jean-Michel Morel.)

Downloaded from http://direct.mit.edu/leon/article-pdf/54/2/179/1910879/leon_a_01768.pdf by guest on 11 August 2021