Exploring the space of abstract textures by principles and random sampling

Luis Alvarez Yann Gousseau Jean-Michel Morel Agustín Salgado

L. Alvarez and A. Salgado
CTIM. Departamento de Informática y Sistemas
Campus de Tafira.
Universidad de Las Palmas de Gran Canaria, 35017, Spain.
Email: {lalvarez,asalgado}@ctim.es

Y. Gousseau
Telecom Paristech - LTCI CNRS
46, rue Barrault, 75013 Paris France
Email: yann.gousseau@telecom-paristech.fr

J-M. Morel
Centre de Mathématiques et de Leurs Applications (CMLA)
Ecole Normale Supérieure de Cachan
61 Av du Pdt Wilson, 94235 Cachan, France.
Email: morel@cmla.ens-cachan.fr

The final publication is available at Springer via http://link.springer.com/article/10.1007/s10851-015-0582-z

Abstract

Exemplar based texture synthesis methods try to emulate textures observed in our visual world. Yet the field of all possible textures (natural or not) has been little explored. Indeed, existing abstract synthesis methods focus on a single generation rule and generate a rather limited set of textures. This limitation can be overcome by combining randomly various generation principles and rule parameters. Doing so gives access to a vast and still unexplored set of possible images. In this paper we introduce an image sampling method combining the main painting techniques of abstract art. This sampler synthesizes what we call multi-layered textures (ML-textures). The underlying image model extends three abstract image synthesis models: the dead leaves model, the spot noise and fractal generators. By respecting minimal self-similarity rules keeping Gestalt theory grouping principles at each texture layer, the abstract textures remain understandable to human perception. The complexity of the generated textures derives from the systematic and randomized use of shape interaction principles taken from abstract art such as occlusion, transparency, exclusion, inclusion and tessellation.

keywords: Texture synthesis, abstract painting, graphic design, dead leaves model, Gestalt theory.

1 Introduction

To meet the growing demand of the entertainment industry, computer graphics has been naturally focusing on the graphical modeling of natural objects and environments, and their physical rendering in realistic scenes. For example in the field of exemplar based texture synthesis, spectacular progress has been made, both in terms of speed and visual realism [46].

However, research on image structure can go far beyond the simple reproduction of natural textures. One feels the need to also explore all means of creating new shapes and textures, regardless of their plausibility in the real world. There is, however, a limitation to the creation of new pictures: these pictures must be understandable to the human visual system. We show experimentally here that there are many more perceptually understandable images than those actually created by current mathematical synthesis methods. The exploration of the subject by mathematicians and computer scientists has been, indeed, rather limited. Many algorithms have been proposed for Non Photorealistic Rendering (NPR), but they are mostly concerned with the artistic stylization of scene acquisitions (photographs, video, 3D objects), and not with abstract synthesis as discussed above. To the best of our knowledge, the mathemati-

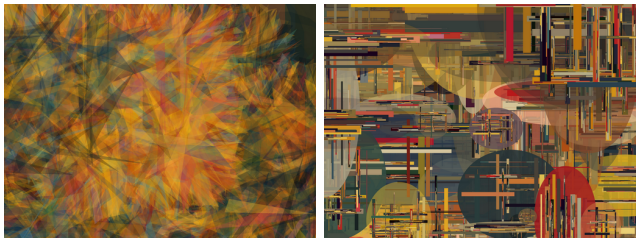

Figure 1: Sample images generated using the proposed technique to create new sorts of textures.

cal attempts at creating new abstract images have each been based on a single generation principle. Classical examples of such models are iterated function systems and more generally fractal models, Fourier spectrum modeling and random noises, classical models from stochastic geometry (such as the dead leaves, hard sphere or random tessellation models). We shall review these models in the next section.

In a different context, the question of generating new sorts of images was investigated almost simultaneously and in strikingly similar terms by abstract painters and Gestalt psychologists at the beginning of the past century. The first technical treatises of abstract art sketch general laws for image formation. They explain the technical rules for creating abstract paintings, starting from simple non-figurative shapes and colors [24], [20]. In parallel, Gestaltists have determined which features in an image are perceptually significant to humans and which are not [47]. Following these studies, a general question arises: how can we create more general classes of images than those existing in nature, that would still be visually understandable? A trivial answer to the question of finding the most general class of images would be to sample randomly, independently and uniformly all pixels of a given image. But, as shown by the American Gestaltist Attneave in a founding work [2], white noise images look completely uniform and are meaningless to humans. Thus, even if the set of white noise image realizations does contain all possible images, the perceptually significant ones are lost in the crowd, being extremely unlikely. So the question remains: how to generate general classes of perceptually understandable computer generated images? Figure 1 shows two images obtained by the image sampling method introduced in the present paper. They cannot be termed "natural images". They obey more complex mathematical generation principles than former generators. Yet, their structure is perceptually straightforward.

Thus, the main goal of this paper is to use the techniques of abstract painting and gestaltism to propose a more general definition of (perceptually understandable) textures than those existing. Let us here emphasize that we do not aim at mimicking a given artist or a given artistic movement. We observed that most abstract paintings are grounded on a rich but short list of interaction principles: occlusion, transparency, exclusion, inclusion. Some of these principles have been individually investigated through mathematical models such as the spot noise or the dead leaves model, where a single principle is used to combine random shapes spread on the plane. These principles are also routinely used in computer graphics to create synthetic images from 3D scenes. Nevertheless, the systematic exploration of the visual possibilities offered by their combination has never been attempted, to the best of our knowledge. Moreover, the use of computers offers possibilities that are beyond the reach of classical painting techniques, especially when combined with a multi-scale or multi-layered structure, as explained in the next section.

Classical models generating abstract images combine structures using few principles and transformations. For example, iterated function systems simply iterate a fixed list of affine transforms. Procedural noises often rely on a single scaling rule. Spot noise models are obtained by adding shapes that are uniformly spread on the plane. The natural generalization of these structures is to allow for a multi-scale (multi-layered) structure with free intra– and inter-scale shape combination rules. On the other hand randomizing all interactions may lead back to chaotic images, close again to white noise. To avoid this, a main feature of the proposed image sampler is the fact that objects are still *coordinated* in each layer. By forcing objects of a given layer to share some properties, a minimal unification of the layer is obtained. In the terms of the Gestalt theory such a unification is called a grouping. It follows that the image remains visually understandable because our perception sees groups of similar objects. Furthermore, the systematic random positioning of all shape elements at each layer implies that *most images generated in this way can be considered to be textures in the sense of Julesz [19].* In accordance with Gestalt theory, these images have enough redundancy to be understood by human perception. Nevertheless, as will be observed, these textures, with some exceptions (figure 2) look rather unnatural (figure 3). This means that the chance of synthesizing natural looking textures like sand or bark exist with this sampler, but are limited.

The technical core of the present paper is the specification of a texture synthesis method which we call multi-layered texture sampler (MLTS). The term "layer" is more general than the term "scale" used in fractal generative models. Objects in successive layers interact with a mix of occlusion, inclusion or transparency that add incremental complexity to the image. There are two kinds of random parameters in MLTS. The layer and inter-layer structure is first fixed by random hyperparameters. Then the shapes, their characteristics and variability are also controlled by random numerical parameters.

The goal in computer graphics, perception theory and mathematical image modeling that ensues from the

Figure 2: ML-textures generated randomly by MLTS, that bear some similarity to textures observable in our visual environment.

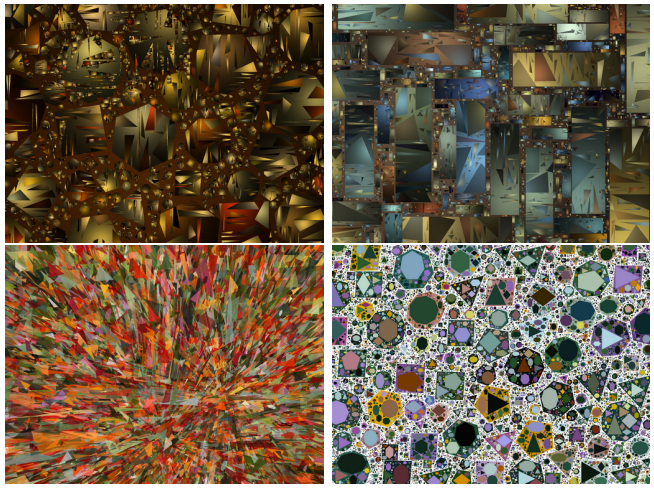

Figure 3: ML-textures generated randomly by MLTS that are definitely abstract but are nevertheless quite understandable to the human perception.

above considerations is to create more general image classes than those currently observable. This program is not new. Ever since the Neolithic, abstract figures and textures have been drawn on potteries. Oriental tapestries display a constant search for new abstract designs. The search for new decorative patterns is present today in all decorative arts and in the textile industry. Providing artists and designers with new principled algorithms to cover surfaces is therefore a valid goal for mathematicians.

2 Antecedents

Abstract painting schools: These schools have published technical treatises for art students where painting is described as a combination of elementary shapes and colors obeying regularity and equilibrium laws [24], [20]. Their shape formation and painting techniques originate partly in Cubism and are best illustrated by their paintings themselves. See Figure 8 for a quick presentation of these painting techniques, relying on a short list of interaction rules between objects, through several abstract paintings mostly from the early 20th century.

Digital art: Digital art opens the way to computer aided design and painting and to the use of simulated randomness [39]. In the sixties, Georg Nees, a pioneer of computer art, used random number generators to generate drawings automatically by controlling a primitive plotter. The numerical works by John Maeda use a random or pseudo-random dead-leaves model based on a repetitive shape. The transparency technique is obviously dominant in some of his works (Figure 4). Michael Noll's 1964 "Computer Composition With Lines" closely mimics the painting "Composition With Lines" by Piet Mondrian. When reproductions of both works were shown to 100 people, the majority preferred the computer version and believed it was done by Mondrian [40].

Gestalt theory was founded by Wertheimer [47, 48]. According to the initial theory our perception proceeds by grouping the local percepts in an image into more global entities, the Gestalts. The main Wertheimer grouping laws are proximity, symmetry, similarity, same color, same shape, same orientation, good continuation, periodicity. The exploration of this theory by the Gestalt schools culminated in the publication of two ground breaking books by Kanizsa in 1979 [21] and Metzger in 1975 [37]. (We refer to the very complete third edition which became the Gestalt Bible. The first edition was published in 1936.)

Geometric marked point processes are mathematical models that yield images that are obtained by the combination of random shapes that are spread on the plane. The simplest way to combine shapes is by addition, resulting in the spot noise model, used for abstract [45] or realistic [12], [13] texture synthesis. A more involved combination principle is occlusion, by which objects occlude each other in natural scenes. The dead leaves model [36] is the mathematical formalization of this principle. In [1], [27], a multi-scale version of this model was introduced for the modeling of natural images (figure 5). In the first of these works, the model is used for several abstract texture synthesis attempts. A transparent version of the dead leaves model is introduced in [11], constituting a non-linear generalization of the spot noise model. Hard core processes [4] study the distribution of object centers when these objects exclude each others. These combination principles (occlusion, transparency, exclusion) will be implemented in our texture generator. See Figure 6 for some instances where transparency is used by our texture generator. In

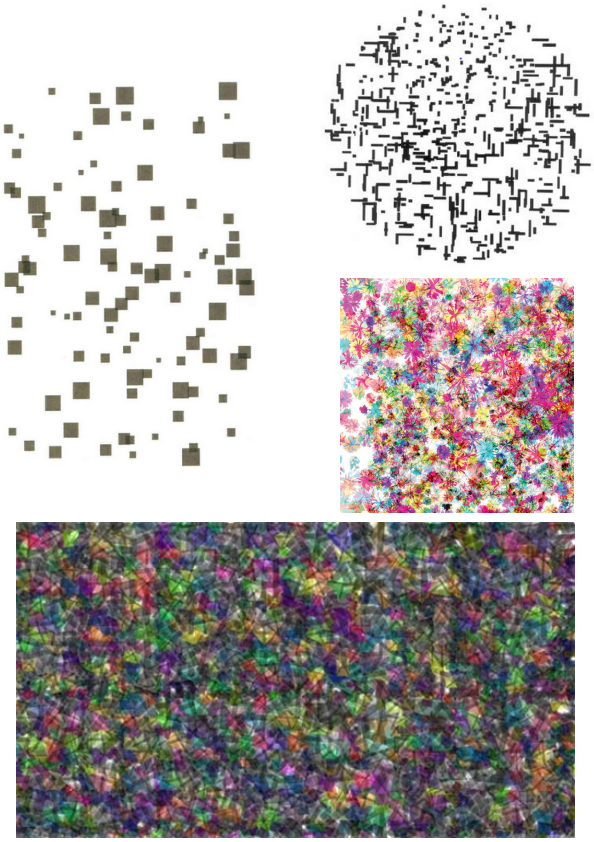

Figure 4: Some founding examples in digital art in the sixties involving the same shape combination techniques as those used in the texture synthesis method. From left to right and top to bottom: Georg Nees painting from the 1960's; Michael Noll's 1964 "Computer composition with lines", a pastiche of Mondrian's 1917 "Composition with lines" (Fig. 8); John Maeda's "Florada, a computational study" [34] and a study contained in "Linear way" [33]. These last two paintings clearly use a dead leaves models with occlusion and transparency techniques.

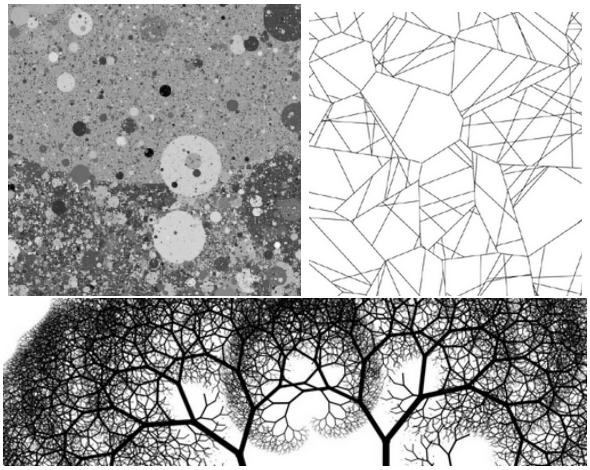

Figure 5: Basic mathematical processes generating abstract textures: a scaling dead leaves model [1], [27], a random tessellation [4], a fractal (iterated function system).

[28] authors present a study of natural image statistics accounts for local geometries. Random tessellations are another way to simulate planar textures by simple subdivision processes [4]. See Figure 5, which also shows a fractal generated by iterated function system[1].

Noise models: Some classical noise models used in computer graphics also rely on the linear combination of simple structures. The most well known of those is the Perlin noise [41] similar to a multi-scale spot noise, that in turn has triggered many studies on procedural noises [26]. These models can also be used for realistic texture synthesis [14].

Non-photorealistic rendering (NPR) aims at producing artistic stylizations from photographs, 3D models or videos, see the recent monograph [43]. While primarily focused on mimicking drawing or painting styles, some of these works produce abstract images from simple shapes. An early such example can be found in [16] and a more systematic investigation of such image abstraction is presented in [44]. In [50], a generative process to produce variations on Kandinsky's paintings is proposed, using occlusion and transparency. While producing interesting images from a small number of patterns, this work does not explore the wide range of possibilities offered by a systematic use of combination principles.

Color palettes will be needed for the rendering of our multi-layered textures. The method we use in this paper relies on color sampling from exemplar images to build a color palette. Building such palettes, usually by unsupervised clustering, is actually very common in data visualization and graphic design. More elaborated

[1] `http://matthewjamestaylor.com/blog/create-fractals-with-recursive-drawing`

approaches to color palette should consider the relationships between colors, building on concepts such as color harmony [38]. A recent approach to coloring problems, relying on a probabilistic modeling of color interactions, may be found in [29].

The authors of [49] present a wavelet based painting analysis method permitting to discriminate painting styles.

Texture perception modelling: Attneave [2] created the first white noise image and postulated that human perception saw nothing but the mean and variance in such images. In a series of founding papers on perceptual texture perception culminating in [19] Julesz followed this lead and conjectured that human texture perception was characterized by second order statistical moments. This theory being notoriously insufficient, he extended it and proposed to characterize texture perception by their local densities of features called *textons*. In [17] there is an attempt to characterize texture perception by wavelet coefficient histograms. This leads to one of the first psychophysically inspired texture synthesis algorithms that works on (some) textures. The study in [42] is probably the most complete attempt to realize Julesz's program: it shows that a wide range of examples of natural textures are efficiently characterized (and can be approximately synthesized) by about 700 statistical wavelet moments and wavelet coefficient correlations.

(Exemplar based) texture synthesis: Its goal is to synthesize a new texture image from a texture sample. One of the earliest suggestion on this subject was to model the textures as parametric Markov random fields [5]. This path was later followed by works on texture synthesis using wavelet decompositions, as mentioned above. In 1999 a founding paper, [8], goes back to Shannon's ideas to simulate artificial text. Textures are synthesized by a non-parametric Markov random field learnt directly from image neighborhoods in the texture sample. This successful technique has been expanded by [7] to apply a recursive copy-paste of such texture neighborhoods, or "texels". In this paper, artificial images are also formed by combining real images and texels from different objects. Related synthesis techniques are presented in the next paragraph.

Random arrangement of patterns A number of works propose to create new images by random spatial arrangements of specific patterns or of image extracts, a task which in particular implies the modeling of interactions between these basic elements. The elements can be elementary patterns [3, 18], text excerpts [35], image extracts to produce mosaics [22], texture extracts to perform texture synthesis [25] or realistic primitives for 3D texture synthesis [32].

Fractal based methods are also well known techniques to generate (not necessarily realistic) synthetic textures (see [6], [23], [31], [15]). In [10] the authors combine shape-from-shading and texture synthesis techniques to create new texturing objects in photographs.

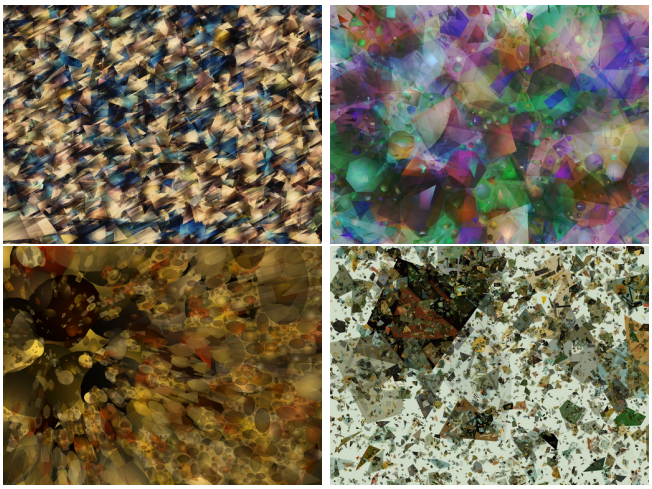

Figure 6: ML-textures built applying the transparency principle between layers.

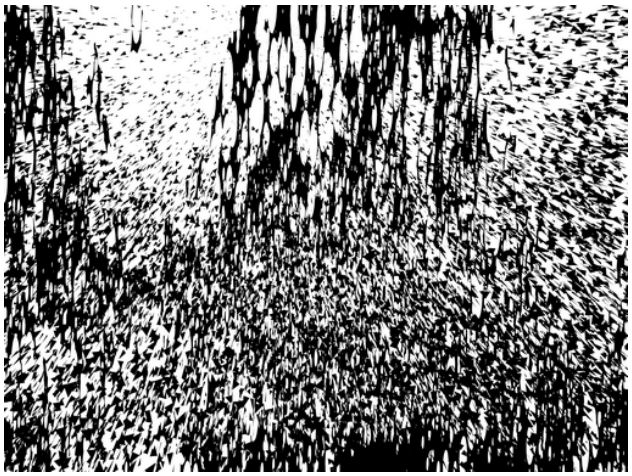

Figure 7: Black and white ML-texture with no transparency. Having objects with arbitrarily large scales, a multi-scale causality and non-local spatial constraints, such textures are not Markovian.

Figure 8: Abstract painting technique. Several famous abstract paintings illustrate how abstract painting has introduced and systematically applied the techniques proposed here for texture synthesis, namely: repetition of simple shapes, occlusion, exclusion, transparency, (random) positioning inside a parent shape, grouping of shapes by some common feature such as orientation, color, vanishing point. R. Delaunay's "Joie de vivre" is based almost exclusively on the reuse of one shape, the disk, with inclusion principle, occlusion principle, and a transparency principle applied on them with a white polygon. Kandinsky's 1930 "Thirteen rectangles" uses rectangles as basic shapes, with a transparency principle and a pseudo random organization which sketches a human silhouette. Hans Arp's 1916 "Rectangles ordered following a random law" also applies the exclusion principle in a pseudo-random disposition of rectangles. A few non random corner coincidences are enforced. Malevitch's 1915 famous "Black rectangle, blue triangle" is based on two simple shapes and the occlusion principle. Van Doesburg's 1930 "Arithmetic Composition" involves a single square black shape on white background and a perspective effect created by decreasing the shape size along a vanishing line. Mondrian's 1917 "Composition with lines" is a pseudo random distribution of rectangles with occlusion and their inclusion in a father shape (a disk). Pollock's 1952 "Convergence" is based on random shape generation and occlusion, by spreading plain color paint pseudo randomly in several successive layers occluding each other. Notice that in these paintings, all shapes in a common "layer" share properties (orientation, color, shape, vanishing point, parallelism, proximity,...) in agreement with Wertheimer's grouping laws. In our abstract texture design, the very same laws are

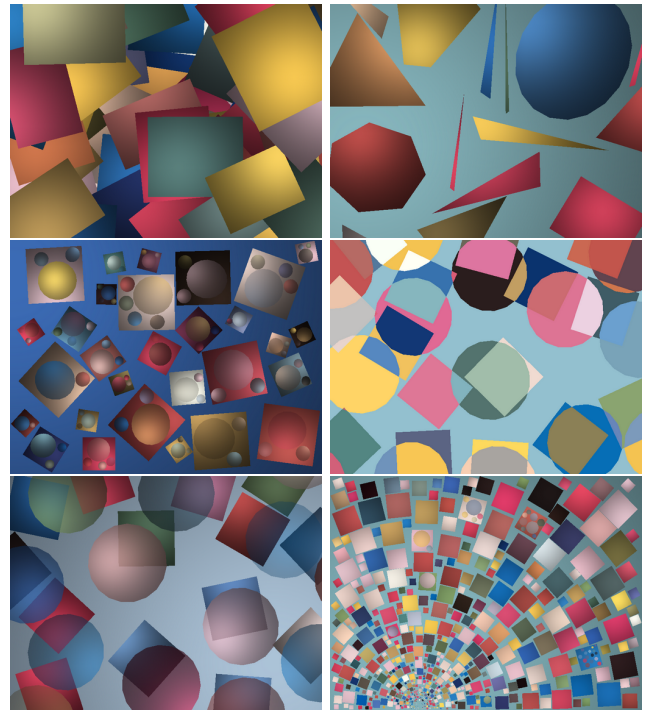

Figure 9: Basic painting principles combining elementary shapes or "texels": Occlusion, exclusion, inclusion, tessellation, transparency (those last three principles operate between two successive layers) and perspective (orienting the texels). These principles are obvious in all abstract paintings where the lack of figuration gives preeminence to shape combination principles, as illustrated in figure 8.

3 Multi-layered textures (ML-textures)

This section details the painting principles and the structure of what we shall call a multi-layered texture (ML-texture). Figure 9 summarizes the painting principles as they arise spontaneously in any painting activity, and were actually formalized in abstract art. Their application on simple basic shapes in abstract paintings is easy to identify, as illustrated in the examples of figure 8.

As in abstract painting, complex textures and images are created by applying interaction laws between texels. Texels are very basic shapes like triangles, rectangles, ellipses and convex polygons. The following list summarizes all considered interactions laws. It is illustrated in figures 8 and 9.

- *occlusion* (corresponds to the physical notion of "being in front" of);
- *exclusion* (corresponds to the physical notion that solid objects cannot interpenetrate, and therefore exclude each other spatially). Figure 13 shows an image mainly generated by this principle;

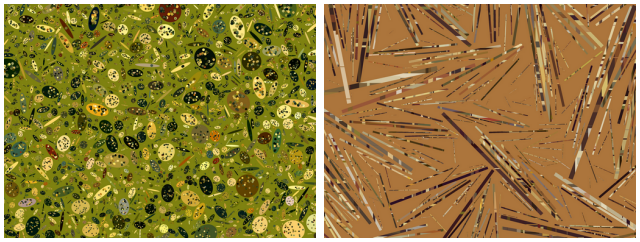

Figure 10: Illustration of the combination of the inclusion and exclusion principles and high shape aspect ratio using elongated rectangles in ML-textures.

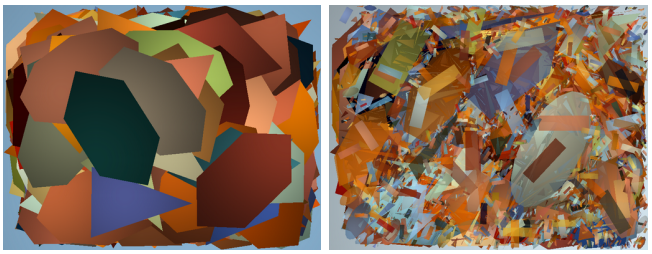

Figure 11: Evolution of the image complexity with the number of layers: one and four layers.

- *transparency* is a variant of "being in front of" in which the objects behind can still be seen by transparency, like in X-ray images.
- *tessellation* is a variant of transparency which is recurrent in cubist and abstract paintings. The shape added in front of other shapes causes the shapes behind to change color in their respective intersections. This principle is illustrated in Figure 15.
- *inclusion principle with the parent* defines the relationship between a container and an object included in the container. This spatial inclusion may be obtained by forcing the child-object not to touch the boundary of its parent. Another mode is that only parts of the sub-object included in the parent are visible, the rest being occluded. Figure 14 shows two artificial multi-layered textures (ML-textures) obtained by a reiteration of the inclusion principle for each layer.
- *similarity* of objects in the same layer. This is the main interaction, equivalent to the gestaltic grouping laws. Objects in the same layer behave in the same specified way with respect to their parents of the preceding layer and can share (some) properties such as: type, size, shape, orientation toward a vanishing point (*perspective effect*), color, interaction of objects in the same layer and the objects of the previous layer.

Next, we introduce the formal definition of the texture model we use in this paper which allows us to manage the above interaction principles in a layered structure.

Definition 1 (ML-texture) *A multi-layered texture (ML-texture) is given by a pair* $(\mathbf{O}, F)$ *where* $\mathbf{O}$ *is a collection of 2D objects :*

$$\mathbf{O} = \{O_{m,k} \subset R^2\}_{m=1,\ldots,N_L}^{k=1,\ldots,N_O(L_m)}, \tag{1}$$

organized in different layers $\{L_m\}_{m=1,\ldots,N_L}$*, in such a way that for any object* $O_{m,k}$*, with* $m > 1$ *there exist an associated father object* $O_{m-1,k'}$ *in the previous layer.* $F : \mathbf{O} \rightarrow R^3$ *is a function which assigns RGB colors to the different objects of* $\mathbf{O}$ *in order to render the ML-texture image.*

The random model of the objects in $\mathbf{O}$, their interaction rules, and the construction of F are specified in continuation. The number of layers is the main ingredient of the texture complexity (figure 11). Each object $O_{m,k}$ is obtained by an affine transformation of a texel (the basic objects representing the texture element) according to the multi-layer rules to be stated next.

- object type : the texels used to generate objects in the layer using affine transformations. See figure 12 to see how this shape influences (mildly) the resulting texture.
- exclusion rule (object interaction with other objects of the same layer) : in each layer it is (randomly) decided if objects can intersect or not (see figure 13).
- inclusion rule (object interaction with objects of the previous layer (the so-called "father objects")) : in each layer it is (randomly) decided if objects have to be included in the associated father object or not.
- object spatial organization : The object spatial organization introduces restrictions in the affine transformations used to generate objects from texels. These restrictions concern :
  - object orientation : to each layer are associated two random points in projective coordinates (to allow points at infinity) such that each layer object is oriented towards one of these two points. For instance, if the points are $(1, 0, 0), (0, 1, 0)$, the objects are oriented in vertical or horizontal directions.
  - object size : For each layer, the object size is fixed by the layer configuration but this size can be reduced when it is not possible to include new objects in the scene satisfying all layer rules.
  - object location : object location is usually chosen randomly according to a Poisson point process, as in the spot noise or dead leaves models.

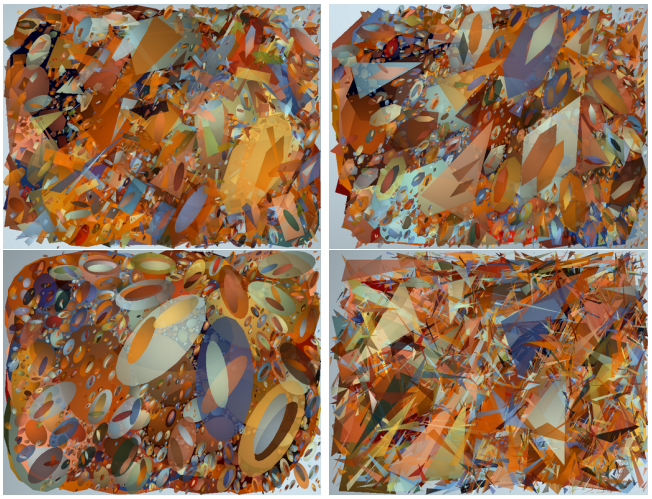

Figure 12: Influence of the basic shape on the texture. From top to bottom and left to right: random shapes, all regular polygons, all ellipses, and all triangles. All other parameters of the synthesis are equal.

Nevertheless, the associated object should satisfy all layer rules, otherwise the new object is rejected.

– vanishing point effect : the points used to define object orientation are also used to introduce a vanishing point effect in the case the orientation points are not at infinity. The size of objects are reduced according to the distance to such orientation points.

Once the multi-layered object structure $\mathbf{O}$ is created, we have to state the function $F : \mathbf{O} \rightarrow R^3$ (see definition 1) which determines the way the ML-texture is rendered. To do that, we use the following rendering rules :

- Color palette : A single color is given to each object by sampling a color from a fixed image, the "palette".
- Object transparency : In the image rendering step a transparency factor is associated to each object.
- Spot light distance : Represents the 3D distance from the objects to the spot light source, see section 5 for more details.
- Tessellation : The image tessellation generated by the object boundaries can also be used to render the image. In this case, colors are associated in an independent way to each connected component of the image tessellation.

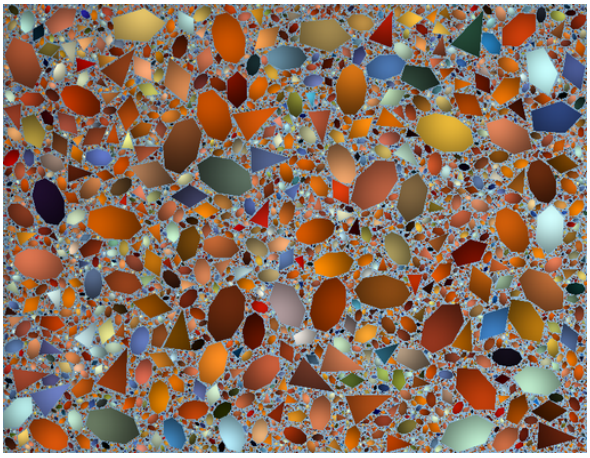

Figure 13: The exclusion principle on a single layer ML-texture. All objects are iteratively added by MLTS with the constraint of not intersecting any of the former ones. Object insertion trials are alternated with shape size reduction.

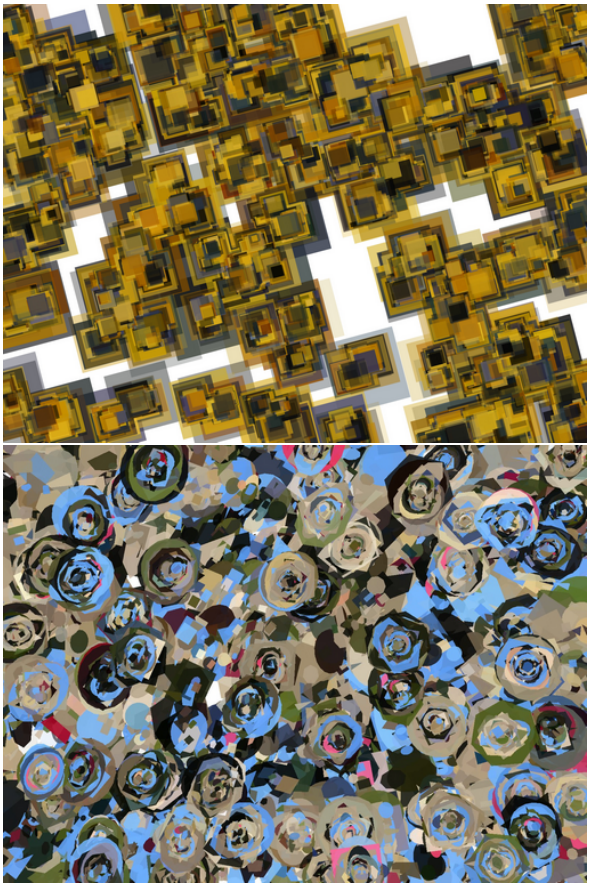

Figure 14: The inclusion principle. All objects in a successive layer are forced to be included in an object of the preceding layer. Object insertion trials are alternated with shape size reduction.

4 The multi-layer texture sampler (MLTS)

This section explains the structure of the proposed multi-layer texture sampler (MLTS). The hierarchical multi-layered structure $L = \{L_m\}_{m=1,\ldots,N_L}$ defines for each layer L_m the object parameter configuration. In accordance with the general description presented above, the layer structure is determined by the following collection of parameters :

Layer configuration parameters (see section 3 for further explanations):

- N_L : the number of layers;
- $\mathbf{T}(L_m)$: texels used in layer L_m to generate objects;
- $Max_O(L_m)$: maximum number of objects in layer L_m for each father object;
- $AspRat_O(L_m)$: object aspect ratio in layer L_m;
- $Max_D(L_m)$: maximum diameter of an object in layer L_m;
- $Max_T(L_m)$: maximum number of trials to include a new object in layer L_m;
- $s_R(L_m)$: size reduction factor of the object diameter applied to try to include new objects in layer L_m;
- $Max_DR(L_m)$: maximum number of diameter reductions to include a new object in layer L_m;
- $Incl(L_m)$: inclusion rule for each layer. If inclusion rule is activated in a layer each object has to be included in his father object;
- $Excl(L_m)$: exclusion rule for each layer. If exclusion rule is activated in a layer objects of the same layer cannot intersect;
- $V_0(L_m), V_1(L_m)$: object orientation in layer L_m (given by two points in projective coordinates);
- $rF(L_m)$: reduction factor of object size when approaching a vanishing point;
- $Seed(L)$: random seed to initiate random object generation.

Implementation The above collection of parameters fully specifies the multi-layered object structure $\mathbf{O}$ defined in (1). We now describe how a multi-layered texture sampler can generate it. There are two possible ways to operate the synthesis from there. The first is fully automatic, while the second permits the user to specify or modify the parameters. Both of these modes are implemented in an online demo [2].

[2]The synthesis framework presented in this paper can be tested online at `www.ctim.es/AbstractTextures`

In the fully automatic mode, ranges are fixed for each parameter, which are then uniformly sampled in this range. The only parameter which is left to the user is the input color image yielding the color palette. Each execution also produces a text file or script containing the image parameters. The same script can be employed again with variations left to the user. In the experiments presented here, unless specified otherwise, we used the purely random procedure. This fairly illustrates the large variety of textures directly obtained by MLTS.

In the supervised mode, the user can guide the choice of parameters. Available parameters have been described in the previous sections and are made of layer configuration parameters as well as rendering parameters (lighting, transparency, tessellation, color palette, "keep father inclusion"). Most parameters have an intuitive meaning (number or size of the objects, number of layers, inclusion parameters, etc.). However, the process may benefit from some user training. Indeed, the combined effects of parameters is not always easy to predict (as happens for example with the interaction between inclusion and exclusion at several layers). In order to facilitate the interaction, the proposed demo allows the user to vary the initial parameters.

An obvious question which arises is the specification of the range of the parameters. All parameters have been given the largest range compatible with image size, the sampling rate, and our perception accuracy. For example, there is not need to create shapes larger than the image itself, as they would cover the whole surface. Conversely, shapes thinner than one pixel cannot be rendered without violating anti-aliasing constraints. Similarly, vanishing points need not be at an infinite distance. Indeed, our eye cannot discriminate far enough vanishing points from vanishing points at infinity.

Many steps of MLTS require a random number. To simplify, we always use the pseudo-random sequence provided by the C/C++ rand() function. Pseudo-random means that the sequence is completely determined by the initial seed which allows us to reproduce the same sequence of random numbers and in particular to generate images only differeing by shape or color, see Figures 12 and 18.

After fixing the layer configuration parameters MLTS builds the multi-layered object structure: first it initiates the random sequence using the random seed given by the parameter $Seed(L)$. The first layer is composed by a single object, namely the rectangle defining the image. For the next layers each object is generated by a random affine transformation of a texel If a generated object does not satisfy the inclusion or exclusion layer rules (in the case any of them is activated in the layer) then it is rejected, and a new trial is made with a new affine transform.

The computational complexity of MLTS mainly depends on the number of layers, the total amount of objects, and the activation in the layers of the inclusion /

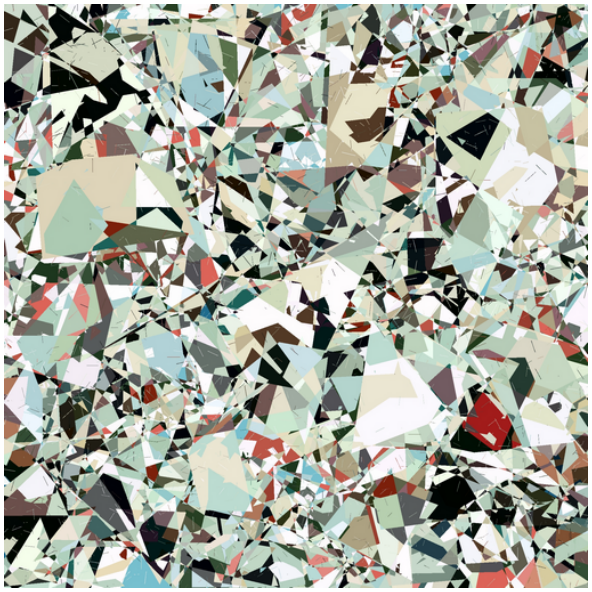

Figure 15: Complex ML-texture where colors are associated in an independent way to connected components of image tessellation generated by the texture object boundaries.

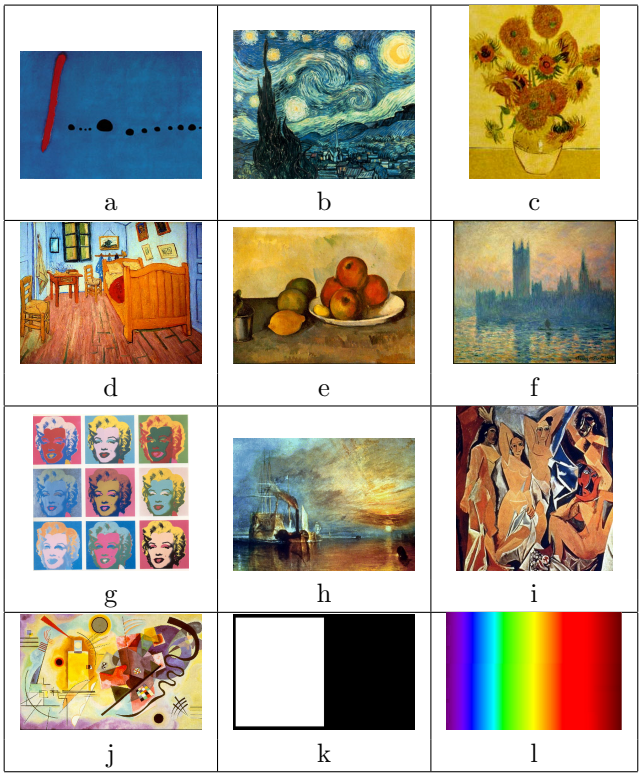

Figure 16: Some color palettes used in this paper.

exclusion rules. Generating a texture using about 3.000 basic objects (it is the case in most textures presented here) takes less than five seconds on a laptop. This rapidity is important as the texture creation process must allow the user to quickly select the textures of interest and discard the others.

5 MLTS rendering of the geometric structure

In the image rendering stage MLTS the function $F : \mathbf{O} \rightarrow R^3$ of the ML-texture is fixed using the following rules :

Color palette. For aesthetic reasons, in our implementation the image colors are sampled from another image serving as a color palette. Digitalized classic paintings or real worlds photographs can provide an effortless color harmony in the final rendered image. The color of each object $O_{m,k}$ is sampled uniformly from the palette. In the tessellation case, a different color is sampled independently for each connected component of the image tessellation generated by the boundaries of $O_{m,k}$.

Transparency effect. Each object receives a random transparency value (between 0 and 1). A value of 0 means that the object is opaque and a value of 1 means that the object is completely transparent.

Light spot effect. Each object is lit using a very simple single light spot source model. A point is randomly chosen in the object. The light spot source is fixed as a point in the 3D space located in the line passing by the selected object point and orthogonal to the image plane. At the selected point the color is the value obtained from the palette image. For the other points the color intensity is attenuated according to the distance to the light spot source (the intensity of the illumination is proportional to the inverse square of the distance to the light source).

Rendering parameters: the rendering step in MLTS involves the following parameters and hyper-parameters:

- image size;
- the color palette image;
- transparency value for each object;
- "keep father inclusion property": when this property is activated objects are not drawn outside their father object;
- light spot distance. 3D distance from each object to his spot light source.
- seed to initiate random assignment of palette colors to objects.

The algorithm to render a texture using the above parameters is straightforward: the pseudo-random sequence is initiated using the random seed. Then a color sampled from the color palette is associated randomly to each object (or to each connected component of the

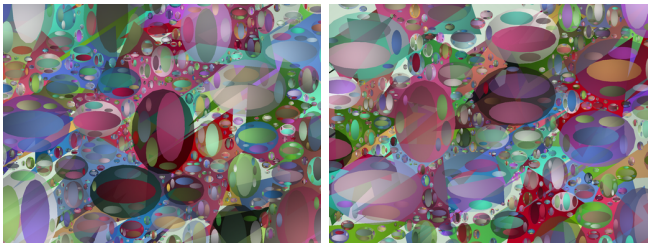

Figure 17: Two realizations of the same ML-texture using different random seeds

tessellation) before the object is drawn in the image by the rendering rules.

The script of the randomly chosen layer and rendering parameters is the other output of MLTS. It fully determines the image texture output. Once a texture is generated, it is therefore possible to generate other textures with more or less similar structure by modifying the script and launching MLTS again. In figure 17 we illustrate the results obtained when changing the seed of the random object generator. In figure 18 we changed the color palette and maintained the same geometry. In figure 19 we present instead some ML-textures obtained using independent parameter configurations.

6 Limitations and possible extensions

The list of object and layer interaction rules in the proposed random texture model could be extended to obtain a still more general model. The following list sketches what extensions can be envisioned.

- several choices in the above synthesis algorithm are fixed arbitrarily, and should be randomized. This is the case for the number of vanishing points (limited to two), or for the fact that the interaction between objects in a same layer does not allow transparency or tessellation, this interaction being allowed only between successive layers.

- color palettes used in this paper are built directly from the pixels of an exemplar image. Another possibility would be to build the palette by a clustering procedure, as it is often done for graphic design applications. A more involved approach would consider simultaneously the choice of color for nearby objects, e.g. by respecting some principles of color harmony [38]. Furthermore, the color density function each natural image or painting is demonstrated experimentally in [30] to be supported by a two-dimensional sub-manifold of the color cube. Thus the design of a harmonic color space remains widely open.

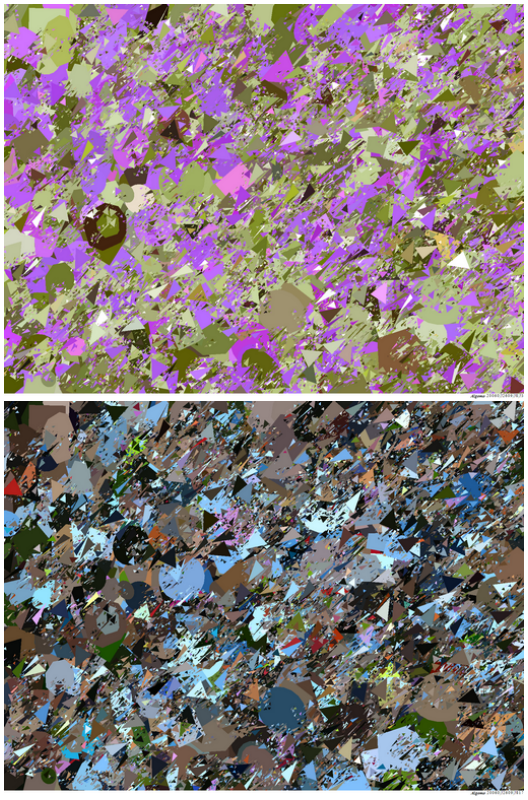

Figure 18: The synthesis method permits to reiterate exactly the same structure with two different palettes. The visual result can be very different. (It is not a one-to-one color transform between both images.) These two particular images are interesting because in spite of their uniform generation principle, they appear unstructured to the eye.

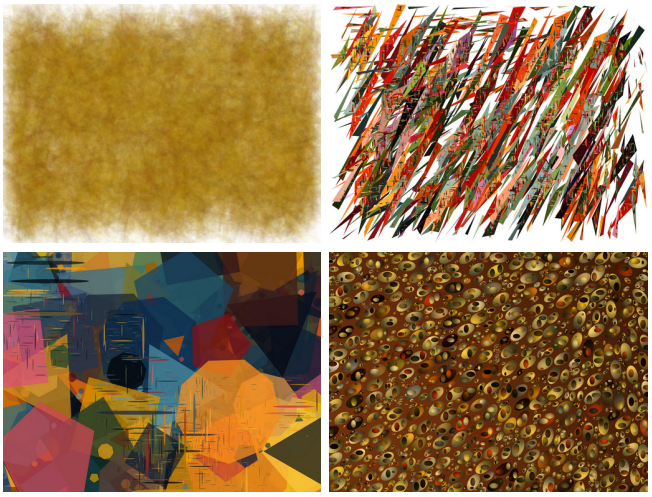

Figure 19: ML-textures using basic texels. The first is close to natural, the second shows the influence of the choice of the vanishing points. The third one is closer to an abstract painting than to a texture. The fourth is a complex and rather unnatural combination of inclusion and exclusion principles.

- passing from textures to scenes: this can be done by enriching the structure so that the children of a given parent are no more coordinated. In that case the human perception of a unique texture will be broken into a perception of several objects.

- it can be objected that a 2D texture model should be derived from a 3D texture model, a rendering method specifying the ensuing 2D scene. In that way shading laws, perspective laws, etc. would be naturally respected (see the paintings of Tanguy for such abstract 3D scenes). Nevertheless abstract art, decorative arts, and many texture synthesis processes prefer to explore 2D layered techniques. We followed this lead.

- The synthesis method is currently organized in an object tree where only successive layers are internally coordinated. This prevents more complex interactions between tree branches or tree layers.

- The multi-layer texture sampler (MLTS) does not yet consider all Gestalt organization laws: objects can be coordinated spatially in more ways than just exclusion or inclusion. The Gestalt grouping laws also include symmetry, spatial periodicity, alignment, good continuation on a curve [9]. Furthermore the sub-objects of an object could have a much better organized relation to the parent than just inclusion. For example sub-objects may form periodic series along the boundary of the parent like in a carpet; they may inherit the symmetries of the parent, etc.

- relations between an object and its parent might be of another sort than just inclusion: in particular we can think of articulation (by which an object touches the parent object), and even coordinated articulation (like the hairs of an object for example). See [20] and [24] for the synthesis of such organized objects, articulated in sub-objects.

- Are the ML-textures generated by the MLTS algorithm textures in the usual sense? Since the term texture has never been formalized, the question will necessarily remain open. The pro arguments are a) the synthesized images retain a self-similarity of shapes and shape behavior at each layer; b) the color palette is uniform in the image, and c) some of the rules impose a Poisson spatial distribution of shape centers. The cons are: a) the inclusion and exclusion rules are sequential. They enforce a causal, non commutative, texel interaction; b) some of the texels are so large and apparent that we obtain an abstract image rather than a texture.

7 Conclusion

We have introduced an abstract texture model, the ML-texture, and its corresponding synthesis algorithm, the MTLS, based on a much more complete list of organization principles than those of the current models. Experimental evidence shows that this algorithm produces a broad variety of texture styles. On the other hand the number of involved parameters in the current algorithm is approximately $14 \times m + 5$ where m is the number of layers (in our experiments not exceeding 3). It can be objected that two of these parameters have a high dimension themselves: one is the choice of the texels used as basic shapes. Yet all things being equal, we observed that the form of the texels has no major influence on the texture visual aspect. The other parameter with a large dimension is the color palette. Here again, the influence of the palette is minor on the geometric style of the texture. All in all, we can therefore surmise that a very large set of texture styles might be obtained with a rather small number of parameters (about 50 for three layers, 100 for six layers). This leads us to the question of the real number of dimensions involved in the creation of all perceptually different texture styles. This question was implicitly posed by Julesz [19] who wondered about the number of parameters characterizing all discriminable textures. The only consistent answer to this question so far has been given by Portilla and Simoncelli [42] who showed that a large number of natural textures could be characterized by a parametric sampling algorithm with about 700 numerical parameters. Yet these authors think that these parameters are significantly redundant (personal communication). Our contribution brings perhaps to this discussion a wider set of perceptually discriminable textures.

We found thanks to our experimental facility that a good proportion of the ML-textures obtained by MLTS look "interesting" to humans and that digitalized classic paints or photographs of natural scenes are excellent color palettes.

The ML-textures showed in this paper represent on average the result of picking one in ten assays (by subjective and aesthetic choices that are hard to make explicit). Each synthesized ML-texture can be reobtained in different (but perceptually equivalent) instances by changing the random generator seed (figure 17). We conjecture that all of the "equivalent" textures generated in that way are pre-attentively indiscriminable in the Julezs sense.

We have created, as a complementary material to this work, a demo web page where users can create their own ML-textures in an interactive way using their own color palette images. They can also vary the parameters in an interactive way or manually in a text file and therefore steer the texture synthesis. This demo web page is located at the project url : `www.ctim.es/AbstractTextures`.

Acknowledgments

Work partially supported by ERC advanced Grant Twelve Labours, and ONR Grant N00014-97-1-0839 (J.-M.M.).

References

[1] Luis Alvarez, Yann Gousseau, and Jean-Michel Morel. The size of objects in natural and artificial images. *Advances in Imaging and Electron Physics*, 111:167–242, 1999.

[2] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[3] Pascal Barla, Simon Breslav, Joëlle Thollot, François Sillion, and Lee Markosian. Stroke pattern analysis and synthesis. In *Computer Graphics Forum*, volume 25, pages 663–671. Wiley Online Library, 2006.

[4] Sung Nok Chiu, Dietrich Stoyan, Wilfrid S Kendall, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[5] George R Cross and Anil K Jain. Markov random field texture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):25–39, 1983.

[6] Stephen Demko, Laurie Hodges, and Bruce Naylor. Construction of fractal objects with iterated function systems. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 271–278, New York, NY, USA, 1985. ACM.

[7] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[8] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[9] Willis Davis Ellis. *A source book of Gestalt psychology*, volume 2. Psychology Press, 1967.

[10] Hui Fang and John C. Hart. Textureshop: Texture synthesis as a photograph editing tool. *ACM Trans. Graph.*, 23(3):354–359, August 2004.

[11] Bruno Galerne and Yann Gousseau. The transparent dead leaves model. *Adv. Appl. Probab.*, 44(1):1–20, 2012.

[12] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Micro-Texture Synthesis by Phase Randomization. *Image Processing On Line*, 2011, 2011. `http://dx.doi.org/10.5201/ipol.2011.ggm_rpn`.

[13] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random phase textures: Theory and synthesis. *Image Processing, IEEE Transactions on*, 20(1):257–267, 2011.

[14] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *ACM Transactions on Graphics (TOG)*, 31(4):73, 2012.

[15] Felix Haase, Maximilian Klein, Andreas Tarnowsky, and Franz-Erich Wolter. Interactive fractal compositions. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '12, pages 181–188, New York, NY, USA, 2012. ACM.

[16] Paul Haeberli. Paint by numbers: Abstract image representations. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 207–214. ACM, 1990.

[17] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 229–238, New York, NY, USA, 1995. ACM.

[18] Thomas Hurtut, Pierre-Edouard Landes, Joëlle Thollot, Yann Gousseau, Remy Drouillhet, and Jean-Franois Coeurjolly. Appearance-guided synthesis of element arrangements by example. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 51–60. ACM, 2009.

[19] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 1981.

[20] Wladimir Kandinsky. *Concerning the spiritual in art*. Courier Dover Publications, 1910.

[21] Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*. Praeger New York, 1979.

[22] Junhwan Kim and Fabio Pellacini. Jigsaw image mosaics. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 657–664, New York, NY, USA, 2002. ACM.

[23] Theodore Kim, Jason Sewall, Avneesh Sud, and Ming C. Lin. A fast fractal growth algorithm. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[24] Paul Klee. *Pedagogical sketchbook*. Praeger, 1960.

[25] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 277–286, New York, NY, USA, 2003. ACM.

[26] Ares Lagae, Sylvain Lefebvre, Rob Cook, Tony DeRose, George Drettakis, David S Ebert, JP Lewis, Ken Perlin, and Matthias Zwicker. A survey of procedural noise functions. In *Computer Graphics Forum*, volume 29, pages 2579–2600. Wiley Online Library, 2010.

[27] Ann B. Lee, David Mumford, and Jinggang Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *Int. J. Comput. Vision*, 41(1-2):35–59, January 2001.

[28] Ann B. Lee, Kim S. Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *Int. J. Comput. Vision*, 54(1-3):83–103, August 2003.

[29] Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Trans. Graph.*, 32(4):37:1–37:12, July 2013.

[30] Jose-Luis Lisani, Antoni Buades, and Jean-Michel Morel. Image Color Cube Dimensional Filtering and Visualization. *Image Processing On Line*, 1, 2011. `http://dx.doi.org/10.5201/ipol.2011.blm-cdf`.

[31] Ruimin Lv, Shouqian Sun, and Kejun Zhang. Fractal strokes. In *SIGGRAPH Asia 2011 Sketches*, SA '11, pages 33:1–33:2, New York, NY, USA, 2011. ACM.

[32] Chongyang Ma, Li-Yi Wei, and Xin Tong. Discrete element textures. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 62:1–62:10, New York, NY, USA, 2011. ACM.

[33] John Maeda. *Linear way*. Fondation Cartier pour lart contemporain, Paris, 2005.

[34] John Maeda. *The laws of simplicity*. Mit Press, 2006.

[35] Ron Maharik, Mikhail Bessmeltsev, Alla Sheffer, Ariel Shamir, and Nathan Carr. Digital micrography. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 100:1–100:12, New York, NY, USA, 2011. ACM.

[36] Georges Matheron. Modèle séquentiel de partition aléatoire. *rapport technique*, page 102, 1968.

[37] Wolfgang Metzger. *Gesetze des Sehens (die Lehre vom Sehen der Formen und Dinge des Raumes und der Bewegung)*. Frankfurt/M.: Kramer, 1975.

[38] Parry Moon and Domina Eberle Spencer. Geometric formulation of classical color harmony. *JOSA*, 34(1):46–50, 1944.

[39] A Michael Noll. Computers and the visual arts. *Design Quarterly*, pages 64–71, 1966.

[40] A Michael Noll. Human or machine: A subjective comparison of Piet Mondrian's" Composition With Lines"(1917) and a computer-generated picture. *The psychological record*, 1966.

[41] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, 1985.

[42] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

[43] Paul L. Rosin and John Collomosse. *Image and Video-based Artistic Stylisation*, volume 42. Springer, 2013.

[44] Yi-Zhe Song, Paul L. Rosin, Peter M Hall, and John Collomosse. Arty shapes. In *Proceedings of the Fourth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 65–72. Eurographics Association, 2008.

[45] Jarke J Van Wijk. Spot noise texture synthesis for data visualization. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 309–318. ACM, 1991.

[46] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, Greg Turk, et al. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117, 2009.

[47] Max Wertheimer. Untersuchungen zur Lehre von der Gestalt. II. *Psychological Research*, 4(1):301–350, 1923.

[48] Max Wertheimer. *Gestalt theory*. Hayes Barton Press, 1938.

[49] Tong Wu, Gungor Polatkan, David Steel, William Brown, Ingrid Daubechies, and A. Robert Calderbank. Painting analysis using wavelets and probabilistic topic models. In *ICIP*, pages 3264–3268. IEEE, 2013.

[50] Kang Zhang and Jinhui Yu. Generating abstract paintings in kandinsky style. In *SIGGRAPH Asia 2013 Art Gallery*, page 18. ACM, 2013.