HAL
open science

A Shape-Aware Model for Discrete Texture Synthesis

Pierre-Edouard Landes, Bruno Galerne, Thomas Hurtut

To cite this version:

Pierre-Edouard Landes, Bruno Galerne, Thomas Hurtut. A Shape-Aware Model for Discrete Texture Synthesis. Computer Graphics Forum, 2013, 32 (4), pp.13. hal-00825735

HAL Id: hal-00825735

https://hal.science/hal-00825735

Submitted on 24 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Eurographics Symposium on Rendering 2013
Nicolas Holzschuch and Szymon Rusinkiewicz
(Guest Editors)

Volume 32 (2013), Number 4

A Shape-Aware Model for Discrete Texture Synthesis

Pierre-Edouard Landes[1,3], Bruno Galerne[2,3] and Thomas Hurtut[1,3]

[1]LIPADE [2]Laboratoire MAP5 (UMR CNRS 8145) [3]Université Paris Descartes, Sorbonne Paris Cité

Figure 1: ***Our Model-Based Discrete Texture Synthesis.*** *Given a 2D or 3D exemplar texture composed of discrete vector elements, our model captures the pairwise element interactions that govern the texture's spatial organization, and accounts for their complex shapes. New output textures can then be generated to fill specified domain.*

Abstract

We present a novel shape-aware method for synthesizing 2D and 3D discrete element textures consisting of collections of distinct vector graphics objects. Extending the long-proven point process framework, we propose a shape process*, a novel stochastic model based on spatial measurements that fully take into account the geometry of the elements. We demonstrate that our approach is well-suited for discrete texture synthesis by example. Our model enables for both robust statistical parameter estimation and reliable output generation by Monte Carlo sampling. Our numerous experiments show that contrary to current state-of-the-art techniques, our algorithm manages to capture anisotropic element distributions and systematically prevents undesirable collisions between objects.*

1. Introduction

The recently coined term of *discrete element textures* refers to 2D or 3D texturing data organized as the collections of individual, visually distinguishable objects. By definition, these textures naturally use an object-based representation, with each element becoming a separate, manipulable entity (in our context, 2D cubic Bézier curves or 3D triangular meshes). This contrasts with their raster counterparts, whose features, although visible, lose geometric information when sampled onto regular grids.

The past few years have seen an increased interest in the development of techniques for handling and creating discrete textures. Discrete textures indeed present many desirable properties for computer assisted creation. Their resolution independence and unmatched fineness make them appealing for artists struggling against aliasing artifacts. Their compact representation, suitable for object instancing, considerably reduces memory storage requirements. In addition, their object-based representation paves the way for novel design and editing procedures, greatly exceeding the possibilities offered by sampled images.

But despite their advantages, discrete textures still suffer from a severe lack of dedicated creation tools, a domain in which raster textures clearly keep the upper hand. Part of the explanation comes from the lack of obvious spatial organization and adjacency relationships between their entities, which impedes even the simplest layout analysis.

Here, we address the challenging issue of by-example discrete texture synthesis: Once an artist has manually created a discrete element texture over a small domain, an automatic procedure for filling wider regions with visually similar textures immediately becomes desirable in order to spare him

from tediously filling regions (see Figure 1). Such a tool provides flexibility to the creation process since it enables one to quickly create artwork for different domains and/or textures.

Contributions Unlike raster textures, the visual signature of a discrete texture is not conveyed by specific co-occurrences of sampled colors, but by the visual aspect and spatial arrangement of their elements. Thus, discrete texture synthesis by example consists of identifying, capturing, and reproducing what makes the organization of the exemplar elements specific. In order to reach this goal, we propose three main contributions:

- We introduce relevant *spatial relationship measurements* between elements that improve element interaction modeling by accounting for their actual geometry.
- We propose a novel stochastic texture model consisting of a *shape process* that extends the long-proven point process theory, and we provide a robust statistical parameter estimation for it.
- We demonstrate the applicability of our shape process by *synthesizing 2D and 3D discrete textures* that benefit from the advantageous properties of our simulation procedure (*i.e.* shape preservation, no unexpected interpenetrations, no initialization dependence).

2. Related Work

Our method belongs to the domain of discrete element placement. Depending on the shape model chosen to guide the analysis and synthesis of the element distribution, existing work falls into two main categories. The first category encompasses all point-based approaches which resort to one or several spatial point positions to locate each element. The second category studies the placement of more elaborate 2D shapes, such as the bounding boxes of the elements. Before providing detail on element placement methods, we briefly review the use of point processes in Graphics, since our shape process model aims at extending such models.

Point Processes in Graphics There has recently been a renewed interest in the generation of random point patterns in Computer Graphics. Initially, literature focused mostly on point processes with the blue noise property (see *e.g.* [LD08, Wei08, Fat11] and the references therein). More recently, several contributions focused on simulating point processes with more varied spectra or correlation functions [LWSF10, ZHWW12, OG12]. Such approaches show that pairwise point distances are the key characteristics of point processes. In this paper, we extend this framework and propose a novel *shape process* which can be thought of as random points with shapes attached to them. These models are called particle processes or germ-grain processes in the stochastic geometry literature [SKM95, SW08]. As for point processes, the main characteristics we focus on are the distances between pair of shapes, but we also consider the relative direction between shapes in order to deal with discrete textures whose elements are anisotropically distributed.

Point-Based Element Placement Unlike raster texture synthesis where an appropriate color must be assigned to each fixed pixel location, the biggest challenge in discrete textures synthesis is to handle the arbitrary positioning of long-range element shapes. Pioneering works often reduce elements to shapes attached to anchor points. The spatial distribution of the anchor points is then considered as a good approximation for the distribution of the elements during analysis and synthesis. A common choice for such points are the centroids of their bounding boxes fitted by principal component analysis (PCA) [BBT*06, BBMT06, IMIM08, HLT*09, AdPWS10, LZZ11]. As highlighted in [HLT*09, LZZ11], this centroidal model implicitly assumes that the elements exhibit an isotropic spatial extent. These approaches thus perform poorly on discrete textures with elongated or complex shapes. They are also unable to capture possible correlations between the different element shapes and their spatial distribution. Li *et al.* [LWSF10] alleviates this problem by performing isotropic point sampling in a distorted, shape-guided space. It is nevertheless limited to the specific type of blue noise-like distributions.

Most recently, Ma *et al.* [MWT11] use multiple point samples per element, and their synthesis method proves to be a valuable designing tool for artists [KIZD12]. Indeed, the tracking of several samples on top of the element surfaces better captures their shape, and enables consistent element deformations. Still, their disconnected element representation may lead to interpenetration issues which are corrected via physics simulation during synthesis. Although physical simulation enables the enforcement of additional physical properties to their outputs, we believe it contradicts our strictly by-example objective, and boils down to adding prior knowledge to their method. In addition, their Expectation-Maximization (EM) approach inspired by [KEBK05] is sensitive to the quality of initialization.

Shape-Aware Element Placement Rather than points, some patch-based texture techniques explicitly handle axis-aligned [DMLG02], or PCA-fitted [GZW03] element boxes. Statistics of the pairwise interactions based on these boxes are assembled in co-occurrence lists or histograms, and enforced on the output textures via iterative procedures. Yet, some measurements, such as the relative orientations between elements, remain limited to the simplistic centroidal assumption. Besides, though preventing interpenetrations, the bounding box model is hardly representative for complex element shapes. To the best of our knowledge, only two approaches directly consider element shapes to guide synthesis [HHD03, LLW12]. However, as in [LWSF10], they mainly focus on blue-noise distributions, where each element repels one another by a roughly constant distance.

In this paper, we propose a new discrete texture model that

fully considers element shapes, disregards the centroidal assumption, and prevents irrelevant interpenetrations. It solely relies on meaningful pairwise interactions, and its parameters are robustly estimated from the exemplar configuration.

3. Overview

As previously motivated, the goal of our synthesis is to capture and reproduce the pairwise spatial interaction statistics observed between the exemplar elements. Since we want to explicitly deal with arbitrary shapes and not points, we first define in Section 4 measurements that accurately assess the spatial interaction between element pairs, and account for both their spatial extents and relative orientation. With such measurements at hand, we conceive a stochastic shape-aware texture model in Section 5 that organizes and probabilistically sums up the observed spatial interactions. Derived from the extension of point processes to our shape-driven objective, the texture model constitutes a *shape process* for which we provide an automatic parameter estimation and synthesis procedure in Section 6.

Two distinct critical challenges are faced by our shape process texture model. Indeed, contrary to existing techniques, we want it to natively support:

- *shape anisotropy*: to account for the extent and privileged direction of the elements during the analysis of their relative spatial organization;
- *distribution anisotropy*: to handle layouts where spatial statistics strongly differ in some directions.

As demonstrated by the results presented in Section 7, those two complementary objectives are fulfilled thanks to our shape-aware spatial interaction measurements and orientation-dependent parametric model.

Notations We consider 2D or 3D vector textures composed of a finite set of discrete elements $\mathbf{x} = \{x_1, \ldots, x_N\}$. Each element $x_n = (u_n, s_n)$ is summed up as the combination of a spatial position u_n in a bounded 2D or 3D domain $\mathcal{P}$, and a shape $s_n \in \mathcal{S}$. In our context, the shape space $\mathcal{S}$ is the discrete union of all the shapes observed within the input exemplar denoted by $\mathbf{z}$. The u_n positions are made coincident with the centroids of the elements' fitted bounding boxes. But contrary to existing work, these positions do not take part in the evaluation of the pairwise element interactions, and are simply used in the context of numerical integration and random sampling.

4. Pairwise Shape Interactions

In the context of point processes, a pair of points is fully described by the distance between the points and their relative orientation. For discrete textures, the interaction between two elements cannot be as easily described due to the shapes of the involved elements. We adapt the aformentioned spatial measurements to our case in this section.

Distance between Element Shapes An intuitive way of evaluating the interaction strength between an element pair is to consider the *length of the shortest path* between their shapes, here denoted $d_{\text{shp}}(x_n, x_m)$. But because of the computational cost of such computations, the tractability of this approach lies in the assessment of an appropriate shape approximation. Contrary to Ma *et al.*'s [MWT11] argument in favor of point-based sampling, we believe that the loss of connectivity would be detrimental to the estimation of accurate element interactions. Hence, instead of samples, we rely on coarser versions of the elements' actual geometries. Consisting in either 2D polylines or 3D low-resolution meshes, our proxy geometries are respectively obtained through the progressive simplification of the elements' original geometries using the Douglas-Peucker algorithm in 2D [Ram72], or successive error-bound edge collapses in 3D [GH97] (see Figure 2). Evaluating $d_{\text{shp}}(x_n, x_m)$ then comes down to computing the minimal distance between pairs of 2D segments, or 3D triangles. While closed-form solutions exist for segments, shortest distances between triangles are computed via an iterative projected gradient descent.

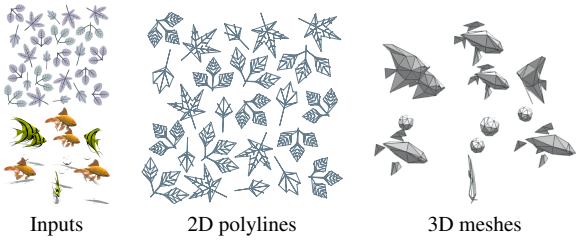

Figure 2: ***Proxy Geometries.*** *In order that our element interactions account for shape connectivity, we resort to simplified proxy geometries to compute shortest path distances in tractable computation times.*

Relative Orientation between Elements As previously stated, part of our textures' spatial layout information is conveyed by the relative orientation between their elements. Straightforwardly considering the line passing through the pair of centroids overlooks their spatial extent, and does not yield reliable directions when the elements lie close to one another, a commonplace phenomenon in most textures. Known as a challenging issue [Blo05], a sensible description of the relative position of arbitrary shapes is mandatory for our technique to capture anisotropic distributions. Inspired by [MW99], we find in the *gravitation force direction* between two element shapes, a robust indicator of the relative orientation between them. Theoretically defined in the continuous case as for a pair of elements (x_n, x_m), the gravitation force vector between two elements is approximated along the different principal axes of the two elements using the cubature integration method [BEG91].

Figure 3 compares the behavior of our distance and relative orientation measurements with their centroid-based

equivalents. Those examples show that the gravitational force direction is generally less slanted and perceptually more satisfying in most cases.

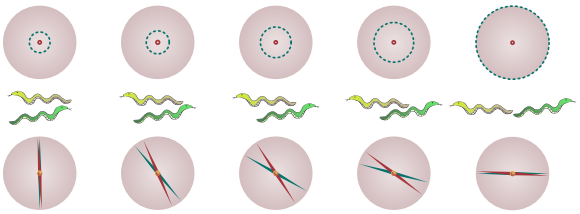

Figure 3: *Shape-Aware* vs. *Centroidal Interactions*. *Our element interactions account for two complementary spatial measurements: the interspace between the element pair computed on their proxy geometries (constant in this figure and symbolized by the small red circles in the top row) and their relative orientation given by the gravitational force integrated over their principal axes (red bottom compasses). The centroidal counterparts are represented in blue: dotted line circles on top, and blue compasses at the bottom. The two interaction models differ on both measurements and our proposed shape-aware interactions are perceptually more satisfying.*

5. Our Shape Process Model

With our pairwise element interactions formalized, we build up our analysis to the scale of the texture. Our model draws inspiration from pairwise interaction point processes [BM89] that we extend to handle complex shapes. These point processes, that are commonly used for the description of spatial data, are defined by their density f

$$f(\mathbf{x}) \propto \prod_n \beta(x_n) \prod_{n,m} \Gamma(x_n, x_m) \tag{1}$$

where β is the first-order *intensity function* related to the object frequencies, and Γ is the second-order *interaction function*. The spatial stationarity of our exemplars allows us to assume that β does not vary with the element position, and to consider it as a constant for each examplar.

The tailoring of the Γ function to our problem is one of our key contributions. In accordance with our interaction measurements introduced in Section 4, $\Gamma(x_n, x_m)$ is chosen so that it only depends on the shape distance and relative orientation of the element pair (x_n, x_m). Intuitively, Γ must express that two objects cannot get too close or interact if they are too distant, and that these distance thresholds may depend on their relative orientation. Nonetheless, Γ should remain as simple as possible to allow for statistical parameter estimation. We thus propose a piecewise constant interaction function in order to supply our model with statistically significant populations of measurements. To do so, we partition the space of possible relative directions into a finite set of *canonical directions* $\mathbf{e}_i$, along which the observed pairwise element distances are then distributed into K intervals. Given two elements (x_n, x_m) whose relative orientation leans towards $\mathbf{e}_i$, their interaction is modeled as:

$$\Gamma(x_n, x_m) = \begin{cases} \gamma_{i,k} & \text{if } \delta_{i,k} \leq d_{\text{shp}}(x_n, x_m) < \delta_{i,k+1} \\ 1 & \text{if } \delta_{i,K} \leq d_{\text{shp}}(x_n, x_m) \end{cases} \tag{2}$$

where $\delta_{i,k}$ are the distance thresholds associated with the direction $\mathbf{e}_i$. A visual representation of such an interaction function is provided on the right part of Figure 4. Particularly important for simulations, the bottom case of Equation (2) specifies that the organization between elements whose interspace exceeds the last distance threshold along their relative orientation is considered as random.

With this general form for Γ, our model density f in Equation (1) can be rewritten as

$$f(\mathbf{x}) \propto \beta^N \prod_{\mathbf{e}_i} \left(\prod_{[\delta_{i,k}, \delta_{i,k+1})} \gamma_{i,k}^{n_{i,k}} \right) \tag{3}$$

where $n_{i,k}$ is the number of $\mathbf{e}_i$-oriented element pairs whose distance falls into the $[\delta_{i,k}, \delta_{i,k+1})$ interval. Experts may notice that our *shape process model* constitutes a generalization of the classic Strauss interaction model for random points [BM89], when objects are points, $K = 1$, and without any orientation dependence.

6. Model Fitting and Sampling

Our shape process model described in Section 5 is entirely controlled by its intensity β and pair interaction function Γ. We now explain how a specific model can be learned from any input discrete texture $\mathbf{z}$, and how we synthesize new output textures associated with this model.

As summarized by Figure 4, the fitting of Γ to $\mathbf{z}$ starts with the subdivision of its support according to a set of canonical directions $\mathbf{e}_i$, and by a set of distance thresholds $\delta_{i,k}$ for each direction. We then statistically estimate the optimal interaction values $\gamma_{i,k}$ of each subdivision cell, as well as the intensity β.

6.1. Tailoring of the Interaction Function Subdivision

Choosing Canonical Directions In order to provide a balanced partition of the possible relative element orientations, the set of unit directions $\mathbf{e}_i$ must exhibit a symmetry with respect to the origin. A constant, even subdivision of the $[0, 2\pi)$ angular interval is sufficient in 2D (8 directions in all our examples), while we resort to the HealPix sphere parameterization in 3D (4 subdivisions for the azimuthal angle, 3 for the polar angle) [GHB*05].

Adjusting the Distance Thresholds Instead of proposing an automatic detection scheme for $\delta_{i,k}$, we take a more pragmatic approach. We choose the same fixed set of quantile indices q_k for all directions $\mathbf{e}_i$ (always $\{0, 0.1, 0.2\}$ here), and compute the corresponding quantiles from the Cumulative

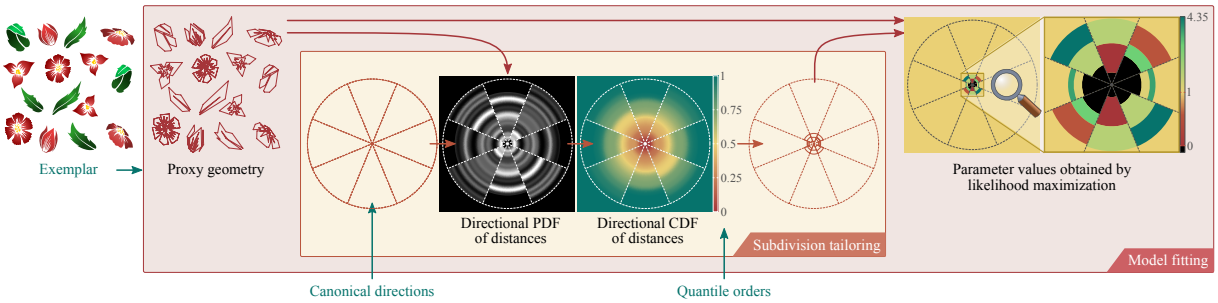

Figure 4: ***Computation of the Interaction Function.*** *While the distance thresholds* $(\delta_{i,k})$ *are chosen as quantiles of the observed distance distributions (see Section 6.1), the interaction strength parameters* $(\gamma_{i,k})$ *are computed by maximizing our model's log pseudo-likelihood (see Section 6.2). Note that each canonical direction* $\mathbf{e}_i$ *creates its own set of parameters.*

Density Function (CDF) of the shape distance distribution along each canonical orientation, as illustrated by the central part of Figure 4. Quantiles, known for their robustness in histogram specification, also prove to be solid data-driven distance thresholds for our model.

6.2. Statistical Estimation of the Interaction Strengths

Once the support of Γ has been subdivided, we must determine the optimal values for the intensity β and the various interaction strengths $\gamma_{i,k}$ with respect to the discrete texture exemplar $\mathbf{z}$.

Following classical spatial statistics methods for point patterns [BT00], the linear parameters β and $\gamma_{i,k}$ of the density f are adjusted to any exemplar $\mathbf{z}$ by maximizing the model's log pseudo-likelihood (log PL) of the configuration $\mathbf{z}$. Since we want to explicitly handle entities (u_n, s_n) made of the union of a position $u_n \in \mathcal{P}$ and a shape $s_n \in \mathcal{S}$, here the usual point-based log PL is written:

$$\log \mathrm{PL} = \sum_{(u_n,s_n)\in\mathbf{z}} \log \frac{f(\mathbf{z})}{f(\mathbf{z}\setminus(u_n,s_n))} - \iint_{\mathcal{P}\times\mathcal{S}} \frac{f(\mathbf{z}\cup(u,s))}{f(\mathbf{z})}\,\mathrm{d}u\,\mathrm{d}s \tag{4}$$

The involved f-ratios, or *conditional intensities*, represent the conditional probabilities of observing a given element, at a specific location, given $\mathbf{z}$. These conditional intensities are explicitly related to the parameters β and $\gamma_{i,k}$ (see supplemental material for explicit formulas). Intuitively, the discrete summation over the elements of $\mathbf{z}$ in Eq. (4) tries to maximize the explanation by the model of the *presence* of the observed elements (u_n, s_n). Conversely, the continuous integral over $\mathcal{P}\times\mathcal{S}$ acts to justify the *absence* of an element at all other locations u, with any possible shape s.

The existence of a unique set of parameters maximizing log PL has been recently demonstrated [BCD08]. In practice, these optimal parameters are estimated by approximating the integral of (4) using an adapted quadrature rule over $\mathcal{P}\times\mathcal{S}$, and revealing the maximization of log PL as a generalized linear model regression (Poisson regression) [BT00] (theoretical and implementation details regarding the log PL computation and maximization are given in supplemental material). For illustration, an example of interaction function Γ with automatically fitted $\gamma_{i,k}$ values is displayed on the right part of Figure 4.

6.3. Texture Synthesis as Model Sampling

A key feature of our model-based approach is that once the model parameters have been adjusted to the exemplar $\mathbf{z}$, synthesizing new discrete texture outputs is straightforward using Monte-Carlo Markov Chain (MCMC) simulations. More specifically, we use here a simple Metropolis-Hastings algorithm with equiprobable element birth and death events [GM94] that is first initialized with an empty configuration. Based on iterative birth or death perturbations, the synthesis algorithm ensures the convergence of the configuration sequence it generates to the targeted probability density. By using the density function f of our model trained on $\mathbf{z}$, our output discrete textures are guaranteed to *globally* respect the same oriented shape-distance statistics as the ones observed throughout $\mathbf{z}$.

Boundary Handling The stationarity assumption of the MCMC sampling forces cautious handling of the boundaries of the synthesis window. In our situation, synthesis is carried out with toroidal border conditions. Since pairwise interactions make elements repel one another, without toroidal conditions, the elements aggregate on the borders of the rectangular domain where they have less neighbors. Additionally, synthesis can be effortlessly achieved for arbitrary domain shapes by spatially varying the model intensity β near the domain border, as illustrated in Figure 1 and Figure 6.

7. Results and Comparisons

7.1. Results

The first column of Figure 5 presents 2D discrete element textures synthesized using our method. 3D results are dis-

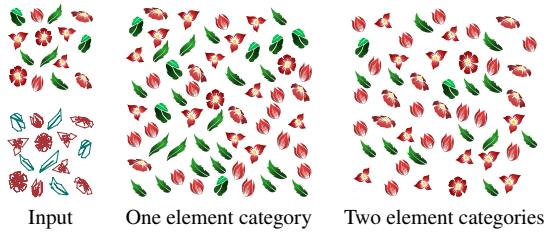

Figure 7: ***Element Categories.*** *When subsets of input elements exhibit strong visual differences (*e.g. *size disparity, specific orientations), it is possible to have our texture model account for this phenomenon via element category. Here, splitting the elements based on color substantially improves the alternation of red and green objects in the output.*

played in Figure 6 and Figure 1 (right). Observe that in all these results element collisions are avoided, even when the element shapes are elongated, and that the distribution anisotropy is fairly well reproduced. As in previous work, for few of these results, we have recourse to element categories as explained below.

Element Categories A refinement of our synthesis can be achieved by introducing the concept of *element categories* as in [IMIM08, HLT*09]. Element categories improve results for special exemplars for which subsets of the texture elements do need to be treated separately (see Figure 7). For instance, in Figure 5 (f), the bimodal distribution of the elements' absolute orientations is best preserved with the resort to two distinct categories. Our model lends itself particularly well to this refinement by simply duplicating the model parameters: one β_c parameter per category c, and one $\Gamma_{(c_n,c_m)}(x_n,x_m)$ function per category pair (c_n,c_m). Results of Figure 5 (f) and Figure 7 use two categories while all the other results use only one. The diversity of these results demonstrates that with only one category our shape aware algorithm performs well for most discrete textures.

7.2. Comparison with Existing Techniques

Figure 5 presents a thorough comparison of our results with four previous works. The exemplar distributions range from regular (Fig.5 (a,e,f)) to fairly random (Fig.5 (b,c,d)); from isotropic (Fig.5 (a)) to anisotropic and structured (Fig.5 (b,e)). Input elements can either be rather simple shapes (Fig.5 (c)) or highly complex (Fig.5 (e)). In contrast with our results, no previous method systematically prevents element collisions and captures anisotropic distributions in all cases.

Comparison with Barla et al. [2006] In Barla *et al.*'s method, pairwise element interactions are only based on the distance between element centroids, and as such performs well only for isotropic elements shapes (Fig.5 (a)). As for the spatial distribution, this technique uses a centroidal Voronoï distribution jittered with small local perturbations learned from the inputs. The results show that this approach is not able to capture complex anisotropic distributions (Fig.5 (b,e,f)).

Comparison with Ijiri et al. [2008] Based on the monitored, greedy growing of its elements' centroid triangulation, Ijiri *et al.*'s technique provides satisfactory results for densely packed textures (Fig.5 (a)). Nonetheless, it still suffers from the same lack of shape awareness as its predecessors and may fail to prevent undesirable overlap between element shapes. It also performs poorly on sparser distributions (Fig.5 (b,d)) where the skewed triangulation faces hinder neighborhood matching and subsequent relaxation, leading to holes or accumulations in the output. Lastly, since the boundary vertices of the input triangulation are excluded from the neighborhood matching, Ijiri *et al.*'s method is overall not suited to small inputs (Fig.5 (b,e)) where identical elements are pasted over and over, revealing distracting lattice structures.

Comparison with Hurtut et al. [2009] Similarly to ours, Hurtut *et al.*'s approach is based on a statistical spatial process, yet its differs in two essential aspects. First, whereas we propose a shape process, they use a point process with a centroid-based approach, and therefore all their results present undesirable element collisions. Second, their distribution model assumes strictly isotropic textures, and thus fails to capture anisotropic distributions (Fig.5 (b,e)). It still performs slightly better than Barla *et al.* thanks to its global pairwise interaction analysis.

Comparison with Ma et al. [2011] Ma *et al.*'s state of the art method relies on a multiple sample based element representation and cast texture synthesis into an Expectation-Maximization (EM) initialized by random patch copy. Since the rectangular patches refer to objects that can go beyond their boundaries, elements collisions may appear in the initialized random output. We observed that these unwanted collisions are not always resolved by the EM algorithm, as illustrated in Figure 5. It must be noted that this issue is not strictly related to the EM algorithm itself. Our statistical model would suffer from the same problem if we also adopted a point based element representation, which pleads in favor of our more accurate proxy geometries for representing objects. Ma *et al.* manage to prevent such collisions in 3D using a physics solver. However, this solution is not straightforwardly applicable in 2D illustration since no physics law may *a priori* blend with the layout of most examplars. A physics solver also tends to produce outputs that locally differ from the user's exemplars. Remaining collisions usually undergo severe deformations in order to respect the local spatial configuration of the exemplar sample neighborhoods, which raises the issue of the reconstruction of each output element based on its associated set of de-

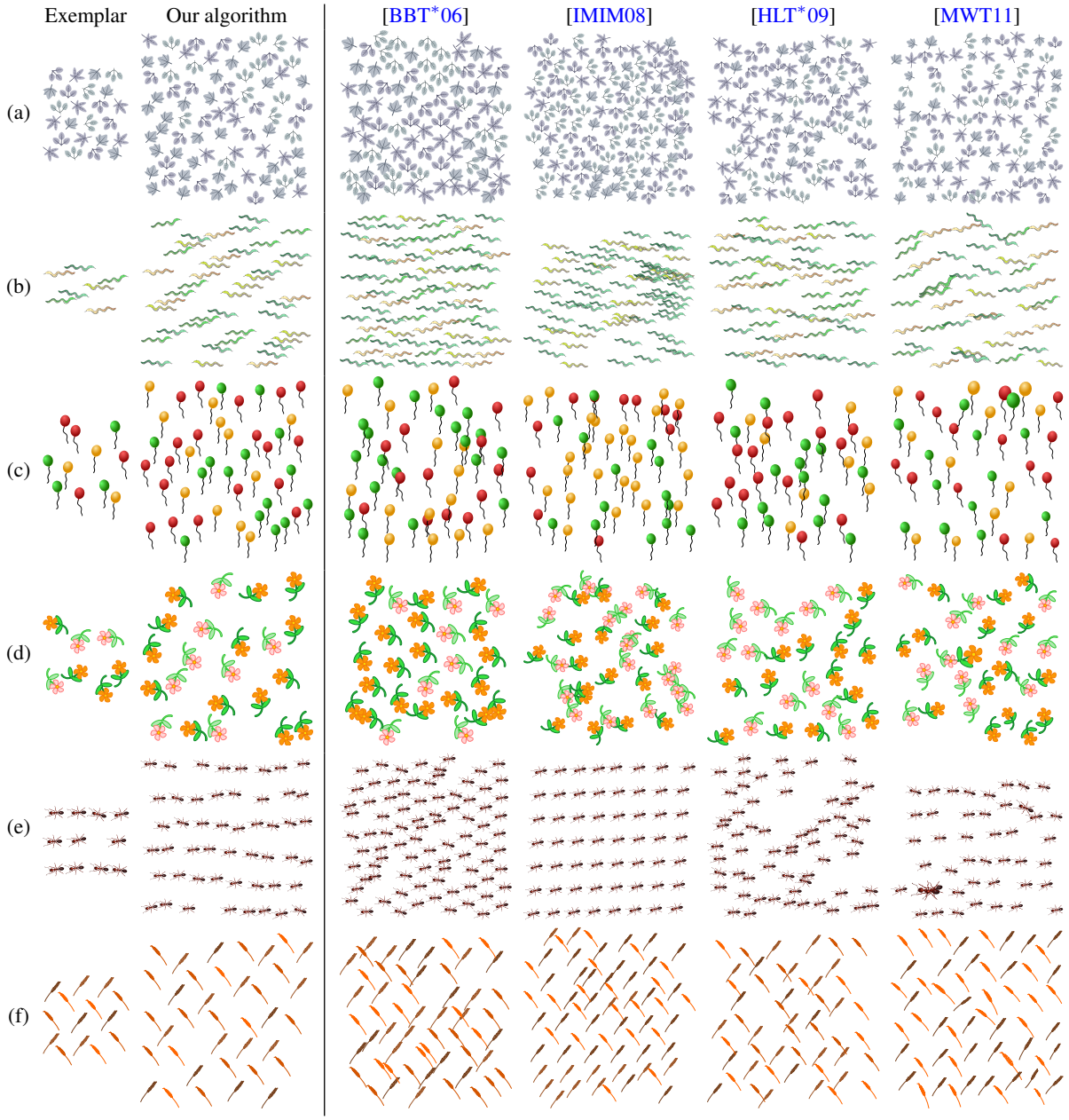

Figure 5: ***2D Generated Textures.*** *We show here the 2D discrete textures produced by our approach compared to previous works for various input. All our results are obtained with the same default parameters, except in* (f) *for which we segment the input in two distinct categories (*cf. *Section 7.1). We observe the preservation of the global distribution of the mutual element interspaces which accurately characterize the input distributions. Thanks to its piecewise-constant oriented interaction functions, our model indeed captures the pairwise element interactions at all scales and directions. Note that, unlike most previous work, highly anisotropic shapes are correctly handled* (b,c,e)*. All previous methods also fail to prevent unwanted element collisions.*

Figure 6: ***3D Generated Textures.*** *Although these two 3D exemplars contain a fairly limited set of complex elements, our model synthesizes consistent results in arbitrary 3D volumes.*

formed samples. Several options can be considered depending on the trade-off between element deformations and local faithfulness to the exemplar spatial organization. We tested three different strategies: a rigid transform fitting (rotation, translation and scaling), an interpolating Thin-Plate Spline (TPS) mapping which exactly respects the sample spatial organization returned by the EM algorithm, and a relaxed TPS mapping. Best rigid transforms generally produce the most visually pleasing results, and have therefore been preferred for illustrating Ma *et al.*'s results in Figure 5. We show the three reconstruction strategies on a collision area in Figure 8.

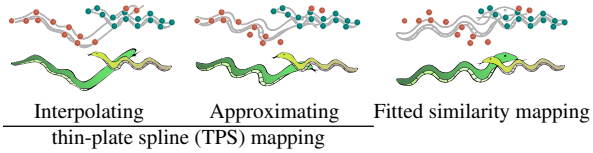

Figure 8: ***Element Reconstruction Issue in [MWT11].*** *The sample set of each object may endure spatial deformations during the EM iterations depending on the initialization and the constraint to preserve the exemplar organization. Reconstructing the objects based on their samples depends on the desired faithfulness to the sample locations dictated by the EM algorithm. We here show reconstructions based on an interpolating and approximating thin-plate spline mapping (left and middle respectively), and a rigid transform (right).*

7.3. Editing

Our model-based synthesis algorithm relies on few parameters. Varying the fitted parameters paves the way to interesting editing operations that we describe below.

Intensity Modulation Although the β intensity function is considered as constant in the exemplar (according to the stationarity assumption of textures), we can locally adjust its value over the synthesis window in order to obtain spatially-varying textures. This variation can intuitively be given by the user using a grayscale mask image, as the beard gradient image in Figure 9.

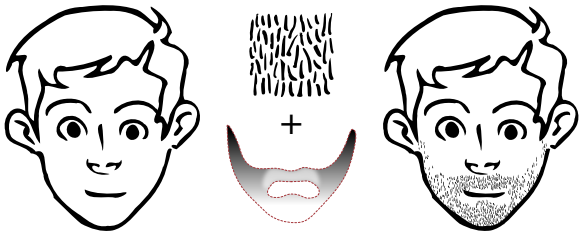

Figure 9: ***Local Intensity Modulation.*** *Spatially modulating the intensity β with control maps (middle) results in local density variation, and thus creates shading-like effects (right).*

Interaction Variation More elaborate editing operations can be achieved by changing the values of the parameters obtained by our input analysis. Indeed, changing the parameters of the fitted interaction function Γ boils down to defining a new shape process model that can be synthesized by our MCMC algorithm. Examples of editing operations are illustrated in Figure 10. These kinds of procedural operations enable one to impose global geometrical organizations such as clustering or alignment.

8. Discussion

Performance Run times for all the discrete texture outputs are provided in Table 1. They range from less than 30 seconds and up to 7 minutes, and are comparable with the ones from previous work such as [MWT11] . These non-negligible run times are explained by the algorithmic complexity of our method's core components (model fitting, and sampling). As shown in Table 1, their execution hinders direct user interaction, especially the intrinsically sequential MCMC iterative synthesis procedure. Yet, some subroutines could be multi-threaded, such as part of the numerical integration of the $\log \mathrm{PL}$ maximization, or the batch process of the MCMC perturbations.

Element Shape Distance Figure 11 compares different pairwise element interaction strategies. The centroidal representation combined with the Euclidean distance is used in most previous work (see Section 2). As pointed out

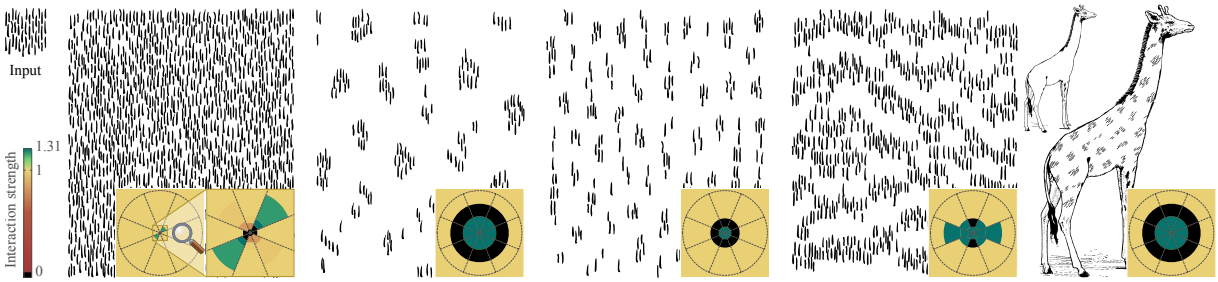

Figure 10: ***Model-Based Texture Editing.*** *From left to right: original texture, output texture without editing, three different textures obtained through parameter editing, a fourth example of edited texture applied on a giraffe (after rotation). The editing operations consist in changing interaction distance thresholds* $\delta_{i,k}$ *and enforcing to zero some interaction values* $\gamma_{i,k}$*. They enable to impose clustering and/or alignment over the whole output texture.*

Table 1: ***Computation Times.*** *The* proxy number*, which greatly impacts the overall times of our technique, designates the average number of proxy sub-parts (2D segments or 3D triangles) per texture element. This information is crucial as discrete texture elements can exhibit arbitrary complexities. The MCMC algorithm is run during* 50,000 *iterations and* 10,000 *iterations, for 2D outputs and 3D outputs respectively.*

Fig	input complexity		output synthesis		
	#elements	proxy number	proxy creation	fitting	MCMC
1*(left)*	12	106.1	1.06s	19.45s	71.58s
5*(a)*	24	55.4	3.9s	8.5s	89.3s
5*(b)*	7	98.8	1.1s	4.4s	195.9s
5*(c)*	11	8.3	0.3s	1.6s	16.8s
5*(d)*	17	23.4	0.7s	1.4s	28.1s
5*(e)*	11	48	2.7s	1.5s	31.1s
5*(f)*	13	25.8	0.3s	1.1s	23.2s
9	51	5	0.8s	4.8s	178.2s
1*(right)*	12	54.4	3.9s	17.8s	108.9s
6*(left)*	12	152.3	1.3s	113.2s	350.4s
6*(right)*	7	140.7	0.9s	62s	451.1s

in [HLT*09, LZZ11], this simple representation performs poorly on textures whose elements show a pronounced anisotropic spatial extent. The Hausdorff metric has been suggested as a possible alternative [HLT*09]. In [BBT*06, BBMT06], it is also used to trigger the clustering of too sufficiently close elements. Initially asymmetric, this metric can be made a distance by taking the maximum values between the two one-sided Hausdorff distances. Still, since it is based on the farthest point-shape distance, the Hausdorff distance does not represent an intuitively exploitable measurement for element interaction.

Interlocking Elements A limitation of our model is its lack of control over occlusions and interpenetrations between elements. While our shape model can be used with texture elements that interlock (see Figure 1 (left)), it does not encode

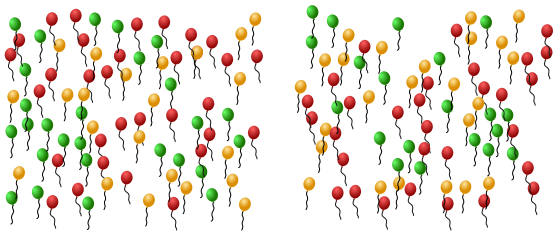

Figure 11: ***Influence of Shape Distance.*** *The Euclidean distance between centroids is only suitable for objects with strictly isotropic spatial extents. When used as the interaction distance between elongated elements, it generally produces undesirable overlaps (left). The Hausdorff distance is a costly alternative that is also unable to avoid overlap. Observe that both distances underestimate the mean nearest-neighbor distance, leading to denser output textures. Conversely, we propose to quantify interactions using the* shortest path length *between element pairs to overcome such shortcomings (Figure 5).*

relevant features such as the location or the amount of overlap between element pairs, as illustrated by Figure 12. Additional measurements could be embedded to our model in order to mitigate this issue, but we must be careful to keep its parameter number reasonable.

9. Conclusion

We presented a new shape-aware method for by-example synthesis of discrete element textures. Our approach relies on a shape process model for which we proposed an automatic procedure for parameter estimation and an MCMC simulation algorithm that ensures the correct sampling of the targeted distribution. Since our model is based on spatial measurements that fully take into account the geometries of the elements, our algorithm is able to reproduce anisotropic element distributions and systematically prevents undesirable object collisions, contrary to previous techniques. As

Figure 12: ***Inaccurate Handling of Interlocking Elements.*** *If element penetrations are observed within the exemplar (left), our model can only account for this phenomenon to a limited extent. Indeed, since ignored during the fitting of our model's parameters, visual features such as the exact amount, or location of overlap cannot be faithfully preserved (right).*

for future work, we would investigate the extension of our approach to time-varying discrete textures, especially for 2D animation where physical simulation is better avoided.

Acknowledgments: This work has been sponsored by the ANR #11-JCJC-008-01 and ANR #12-CORD-0025.

References

[AdPWS10] ALVES DOS PASSOS V., WALTER M., SOUSA M. C.: Sample-based synthesis of illustrative patterns. In *Pacific Graphics* (2010), pp. 109–116. 2

[BBMT06] BARLA P., BRESLAV S., MARKOSIAN L., THOLLOT J.: *Interactive hatching and stippling by example*. Tech. rep., INRIA, 2006. 2, 9

[BBT*06] BARLA P., BRESLAV S., THOLLOT J., SILLION F., MARKOSIAN L.: Stroke pattern analysis and synthesis. In *Eurographics* (2006), vol. 25, pp. 663–671. 2, 7, 9

[BCD08] BILLIOT J.-M., COEURJOLLY J.-F., DROUILHET R.: Maximum pseudolikelihood estimator for exponential family models of marked gibbs point processes. *Electronic Journal of Statistics 2* (2008), 234–264. 5

[BEG91] BERNTSEN J., ESPELID T. O., GENZ A.: An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Trans. Math. Software 17*, 4 (1991), 437–451. 3

[Blo05] BLOCH I.: Fuzzy spatial relationships for image processing and interpretation : a review. *Image and Vision Computing 23*, 2 (2005), 89–110. 3

[BM89] BADDELEY A., MØLLER J.: Nearest-neighbour markov point processes and random sets. *International Statistical Review 57* (1989), 89–121. 4

[BT00] BADDELEY A., TURNER R.: Practical maximum pseudolikelihood for spatial point patterns. *Australian & New Zealand Journal of Statistics 42*, 3 (2000), 283–322. 5

[DMLG02] DISCHLER J.-M., MARITAUD K., LÉVY B., GHAZANFARPOUR D.: Texture particles. In *Eurographics* (2002), vol. 21, pp. 401–410. 2

[Fat11] FATTAL R.: Blue-noise point sampling using kernel density model. In *SIGGRAPH* (2011), vol. 28, pp. 1–10. 2

[GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *SIGGRAPH* (1997), pp. 209–216. 3

[GHB*05] GORSKI K., HIVON E., BANDAY A., WANDELT B., HANSEN F., REINECKE M., BARTELMANN M.: HEALPix: a framework for high-resolution discretization and fast analysis of data distributed on the sphere. *Journal of Astrophysics 622* (2005), 759–771. 4

[GM94] GEYER C., MØLLER J.: Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics* (1994), 359–373. 5

[GZW03] GUO C., ZHU S., WU Y.: Modeling visual patterns by integrating descriptive and generative methods. *IJCV 53*, 1 (2003), 5–29. 2

[HHD03] HILLER S., HELLWIG H., DEUSSEN O.: Beyond stippling - methods for distributing objects on the plane. In *Eurographics* (2003), vol. 22, pp. 515–522. 2

[HLT*09] HURTUT T., LANDES P.-E., THOLLOT J., GOUSSEAU Y., DROUILHET R., COEURJOLLY J.-F.: Appearance-guided synthesis of element arrangements by example. In *NPAR* (2009). 2, 6, 7, 9

[IMIM08] IJIRI T., MECH R., IGARASHI T., MILLER G.: An Example-based Procedural System for Element Arrangement. In *Eurographics* (2008), vol. 27, pp. 429–436. 2, 6, 7

[KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. In *SIGGRAPH* (2005), vol. 24, pp. 795–802. 2

[KIZD12] KAZI R. H., IGARASHI T., ZHAO S., DAVIS R.: Vignette: Interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *CHI* (2012). 2

[LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (2008), 114–129. 2

[LLW12] LU L., LÉVY B., WANG W.: Centroidal voronoi tessellations for line segments and graphs. In *Eurographics* (2012). 2

[LWSF10] LI H., WEI L., SANDER P., FU C.: Anisotropic blue noise sampling. In *SIGGRAPH Asia* (2010), vol. 29, p. 167. 2

[LZZ11] LIU D., ZHANG J., ZHOU C.: Perceptually-based stroke pattern synthesis. In *Pacific Graphics (Short Papers)* (2011), pp. 13–17. 2, 9

[MW99] MATSAKIS P., WENDLING L.: A new way to represent the relative position between areal objects. *Pattern Analysis and Machine Intelligence 21*, 7 (1999), 634–643. 3

[MWT11] MA C., WEI L.-Y., TONG X.: Discrete element textures. In *SIGGRAPH* (2011), vol. 30, pp. 1–10. 2, 3, 7, 8

[OG12] ÖZTIRELI A. C., GROSS M.: Analysis and synthesis of point distributions based on pair correlation. In *SIGGRAPH Asia* (2012), vol. 31, pp. 1–10. 2

[Ram72] RAMER U.: An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing 1*, 3 (1972), 244 – 256. 3

[SKM95] STOYAN D., KENDALL W. S., MECKE J.: *Stochastic geometry and its applications*, second ed. Wiley series in probability and mathematical statistics. John Wiley & Sons, 1995. 2

[SW08] SCHNEIDER R., WEIL W.: *Stochastic and Integral Geometry*. Probability and Its Applications. Springer, 2008. 2

[Wei08] WEI L.-Y.: Parallel poisson disk sampling. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 20:1–20:9. 2

[ZHWW12] ZHOU Y., HUANG H., WEI L.-Y., WANG R.: Point sampling with general noise spectrum. *ACM Trans. Graph. 31*, 4 (July 2012), 76:1–76:11. 2