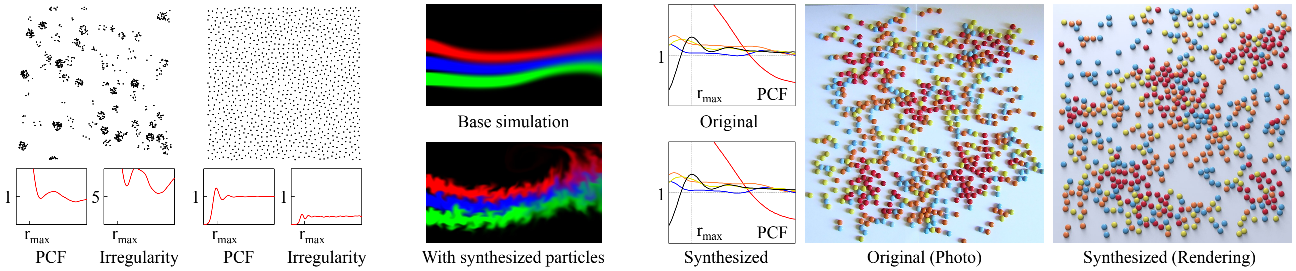
Analysis and Synthesis of Point Distributions based on Pair Correlation

A. Cengiz Öztireli*
ETH Zürich

Markus Gross†
ETH Zürich



Base simulation

With synthesized particles

Original

Synthesized

Original (Photo)

Synthesized (Rendering)

Figure 1: *Left: Our analysis and synthesis methods offer a unified treatment of point distributions, middle: to understand and recreate complex structures such as turbulence, right: by generating distributions of possibly multiple classes of objects with pair correlation functions (PCF) precisely matching those of provided input examples, such as the distributions of these candies of four colors.*

Abstract

Analyzing and synthesizing point distributions are of central importance for a wide range of problems in computer graphics. Existing synthesis algorithms can only generate white or blue-noise distributions with characteristics dictated by the underlying processes used, and analysis tools have not been focused on exploring relations among distributions. We propose a unified analysis and general synthesis algorithms for point distributions. We employ the pair correlation function as the basis of our methods and design synthesis algorithms that can generate distributions with given target characteristics, possibly extracted from an example point set, and introduce a unified characterization of distributions by mapping them to a space implied by pair correlations. The algorithms accept example and output point sets of different sizes and dimensions, are applicable to multi-class distributions and non-Euclidean domains, simple to implement and run in $O(n)$ time. We illustrate applications of our method to real world distributions.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—;

Keywords: Sampling, distributions, point processes, multi-class, blue-noise, Poisson-disk sampling, anti-aliasing

Links: DL PDF WEB

1 Introduction

Observing and recreating complex structures and phenomena of the real world is one of the main goals of computer graphics. The structures often exhibit repetitive patterns and symmetries and can be explained well in a distributional sense. Analysis of a specific phenomena is followed by fitting appropriate models and developing specialized algorithms for synthesis. Modeling complex object distributions [Deussen et al. 1998; Lagae and Dutré 2006; Wei 2010; Ma et al. 2011], turbulence synthesis for fluids [Selle et al. 2005a; Pfaff et al. 2009], crowd simulations [Narain et al. 2009] and many more applications can be seen as instances of this process. Hence, a very important open problem is learning and synthesizing general distributions.

Although point distributions arise in many contexts, analysis tools and synthesis algorithms in graphics have been mostly focused on blue-noise distributions, where points are distributed randomly with a minimum distance between each pair. It is well-known that using such a distribution for sampling leads to high quality anti-aliasing. They are also used in many contexts such as image sampling [Cook 1986], geometry processing and synthesis [Alliez et al. 2002; Ma et al. 2011], object placement [Deussen et al. 1998; Lagae and Dutré 2006; Wei 2010], or procedural noise generation [Lewis 1989]. This has led to various synthesis algorithms that generate distributions with different regularity, density, and randomness. However, the characteristics of the generated distributions are solely determined by the underlying point processes used and limited to blue-noise sampling. In contrast, nature is full of distributions with complex characteristics and many applications in graphics require analyzing and synthesizing such datasets.

Point distributions have also been extensively studied in physics and spatial statistics [Torquato 2002; Illian et al. 2008]. The emphasis in these fields is put on analyzing general distributions and fitting models to understand natural processes. Statistics that depend on correlations of locations and marks of points are used to analyze a diversity of distributions and have been proven to be powerful and discriminative.

In this paper, we introduce methods for analysis and synthesis of general multi-class point distributions based on the statistical measure *pair correlation function* (PCF). To explore the nature of this measure, we introduce an analysis based on the interpretation of it as a mean in a high dimensional vector space that we call the *pair*

*e-mail: cengizo@inf.ethz.ch
†e-mail: grossm@inf.ethz.ch

correlation space (PCS). The vector for a given point in the PCS simply measures the distribution of its distance to all other points. We show that in this space, the degrees of freedom to characterize point distributions is low and directly linked to regularity. This analysis allows us to explain distributions and existing synthesis algorithms in a unified way, propose an irregularity measure, and show that the PCF provides a compact representation for the characteristics.

Following this analysis, we propose two general synthesis algorithms. The first one is a generalization of dart throwing for arbitrary PCFs and the other is a gradient descent based fitting of the PCFs. The output of the first algorithm is used as the input for the second to facilitate convergence. These algorithms can generate point distributions with desired characteristics extracted from example distributions or synthesized. The example point sets and generated point sets can be of different dimensions and sizes, contain multiple classes, and reside on non-Euclidean domains. The algorithms are simple to implement and run in $O(n)$ time.

2 Related Work

Point Distributions in Computer Graphics

Spectral measures based on periodograms, along with simple scalar measures such as the packing density, have been the prevalent tools for analysis of point distributions in graphics [Ulichney 1987; Lagae and Dutré 2008]. Recently, diagrams with similar analysis power have been proposed for distributions in general non-Euclidean domains by considering the distribution of difference vectors between points [Wei and Wang 2011]. The constructed measures are then typically used for qualitative evaluation of the properties of distributions. We propose a new analysis that can quantitatively explain and relate distributions by mapping them into a space constructed by using distances between points, where the distance metric can belong to an arbitrary metric space.

Synthesis algorithms in graphics are mostly focused on distributions with blue-noise characteristics. The earliest thread of methods to generate blue-noise distributions is based on randomly generating points in space and accepting or rejecting based on a criterion. This algorithm is known as *simple sequential inhibition* in statistics, *random sequential adsorption* in physics [Illian et al. 2008], and dart throwing in graphics. The initial dart throwing algorithm [Cook 1986] has been accelerated using a hierarchy of allowable radiuses [McCool and Fiume 1992], specialized data structures [Dunbar and Humphreys 2006], parallelization [Wei 2008], adaptive trees [Jones 2006; White et al. 2007; Gamito and Maddock 2009], or explicit void region representations [Ebeida et al. 2011]. It has also been extended to multiple classes of objects [Wei 2010] such that distributions of points in different classes as well as that of all points have blue-noise characteristics. We extend the standard and multi-class dart throwing algorithms such that general distributions with arbitrary characteristics can be handled.

In order to increase density and regularity, relaxation methods where points are iteratively moved so as to optimize an energy function and satisfy certain constraints are used. Lloyd's algorithm [Lloyd 1982] minimizes the quantization error and thus can be utilized to obtain optimum placement of points that capture the whole space well [McCool and Fiume 1992]. To avoid the optimum involving regular structures, an equal area constraint for the Voronoi regions [Balzer et al. 2009], or injection of randomness into the relaxation algorithms [Schmaltz et al. 2010; Fattal 2011] can be used. Each of these methods generate distributions with certain blue-noise characteristics dictated by the construction of the algorithms. In contrast, our gradient descent based relaxation algorithm can be used to generate distributions with general characteristics directly controllable via specifying example distributions or statistics.

Point Processes

Exploring patterns in point distributions is very important in a diverse selection of disciplines such as physics, chemistry, sociology, geology, and astronomy. Hence, developing models for point patterns to explain variations in nature has been attracting attention of statisticians and physicists for a long time. In these fields, the generating processes of point patterns are called *point processes* [Illian et al. 2008]. A particular point set is regarded as a realization of a point process. Here, we provide a brief overview of the field that is in the scope of this paper.

On the analysis side, the point process statistics are mainly concerned with correlations of point locations and marks. The measures range from intensity of points to pairwise or higher order correlations and topological characteristics such as statistics related to Voronoi regions. In particular, we will base our analysis on the *pair correlation function* which measures the probability of having a pair of points at certain locations in space. This statistical measure was introduced in the physics literature at the beginning of the 20th century and is also known as the *radial distribution function* [Torquato 2002]. We show that the PCF is sufficient to precisely describe characteristics of distributions with diverse properties, and it can be utilized to define new analysis tools.

On the synthesis or simulation side, a typical approach is to generate distributions from fitted models. The models can be generally categorized as *hard-core* (with a minimum distance between each pair of points) or *clustering* processes, or combinations of the two at different scales, depending on the interaction between the points. Examples of some models are Gibbs, Cox, Gaussian [Kerscher 2001], Matérn hard-core and clustering, and Neyman-Scott processes, random sequential adsorption (dart throwing), and force based algorithms [Jodrey and Tory 1985]. Once the model parameters are estimated, variants of Markov Chain Monte Carlo methods can be used to generate point sets following the model. However, as with all model-based approaches, this method is limited by the models chosen in the first place. Instead of relying on estimating parameters of a model, it is possible to randomly explore the configuration space of all point locations and marks by adding or removing a point at a time so as to satisfy a given condition [Torquato 2002]. However, this direct approach is not feasible and scalable for many cases. Our algorithms do not rely on a model and still produce precise reconstructions of characteristics for general metric spaces and multiple classes of objects efficiently.

3 Analysis of Point Distributions

Statistical measures are extensively used to analyze distributions of locations of objects ranging from atoms to galaxies. The main aim of the measures is to extract correlations in the point cloud data, which makes them deviate from classical statistical measures. The field of point processes deals with modeling and analyzing stochastic distributions of points based on the extracted correlations. For an excellent review of the theory and applications of point processes, we refer the reader to Illian et al.'s book [2008].

In the rest of the section, we will consider stationary (i.e. translation invariant), isotropic (i.e. rotation invariant) and ergodic (i.e. analysis on a finite window is sufficient) point processes. Hence, the analysis of such a point process does not depend on the point, direction, or window chosen. We will call such point processes simply as isotropic. If an adaptive or anisotropic distribution is re-

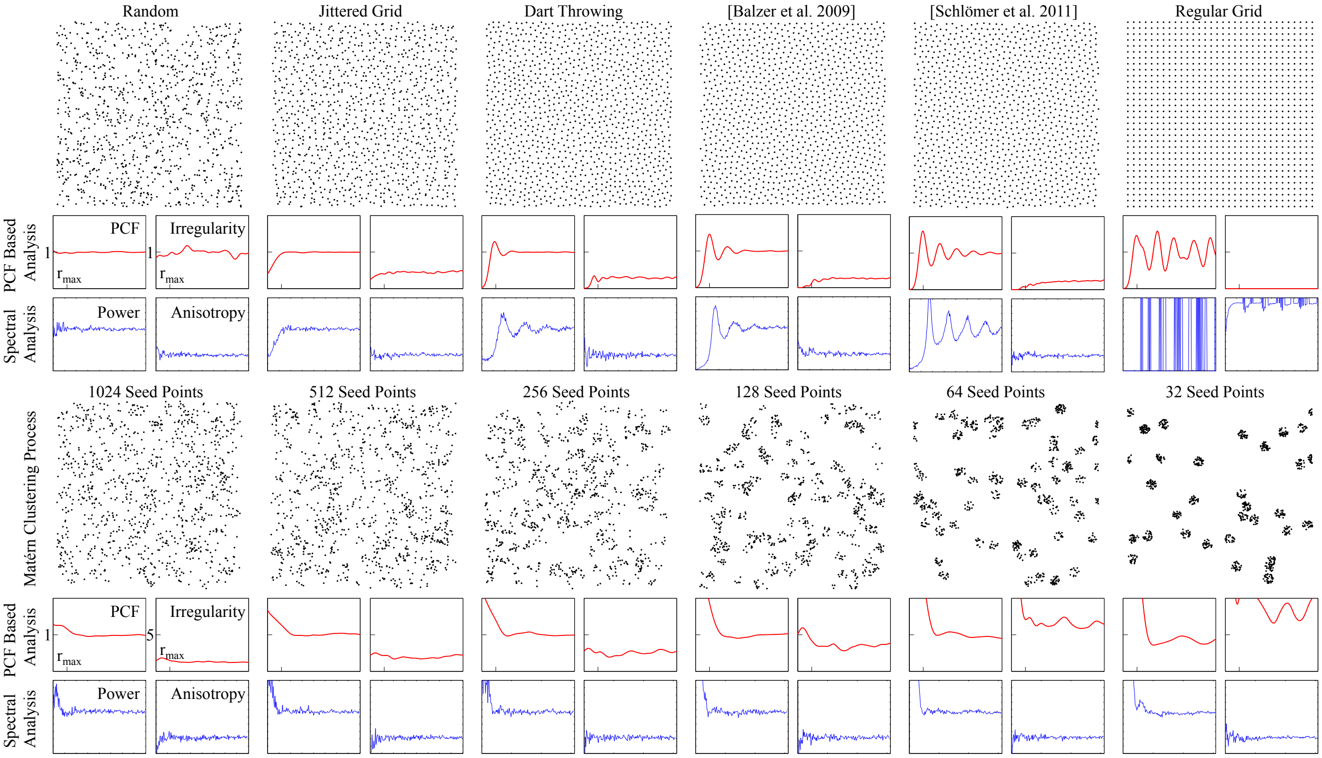


Figure 2: *PCF, irregularity, and radial spectral measures for distributions generated by various algorithms. Matérn clustering process is run with clustering radius* $0.8r_{max}$*. All point sets have approximately* 1024 *points. For the PCF and irregularity graphs,* $r \in [\sigma, 5]\, r_{max}$*.*

quired, a warped domain can be constructed via a modified distance metric [Li et al. 2010].

3.1 The Pair Correlation Function

Among the various statistics used to describe and analyze patterns, the *pair correlation function* is widely accepted as the most informative. Intuitively, this measure $g(\mathbf{x}, \mathbf{y})$ describes the joint probability of having points at locations $\mathbf{x}$ and $\mathbf{y}$ at the same time.

A precise definition of the PCF can be given in terms of the intensity λ and product density ϱ of a point process. The intensity $\lambda(\mathbf{x})$ of a point process is the average number of points in an infinitesimal volume around $\mathbf{x}$. Hence, intuitively it measures the average density of the points. For isotropic point processes, this is a constant value λ. To define the product density, let $\mathbf{x}_i$ denote the points, B_i infinitesimal spheres around the points, and $\mathrm{d}V_i$ the volume measures of B_i. Then $p(\mathbf{x}_1, \cdots, \mathbf{x}_n) = \varrho(\mathbf{x}_1, \cdots, \mathbf{x}_n)\mathrm{d}V_1 \cdots \mathrm{d}V_n$ is defined as the probability of having $\mathbf{x}_i$ in the infinitesimal spheres B_i. For a pair of points, a second order version of this probability is $p(\mathbf{x}, \mathbf{y}) = \varrho(\mathbf{x}, \mathbf{y})\mathrm{d}V_x\mathrm{d}V_y$. In the isotropic case, ϱ only depends on the distance between the points, hence one can write $\varrho(\mathbf{x}, \mathbf{y}) = \varrho(\|\mathbf{x} - \mathbf{y}\|) = \varrho(r)$ and $p(r) = \varrho(r)\mathrm{d}x\mathrm{d}y$. The PCF is then defined as

$$g(r) = \frac{\varrho(r)}{\lambda^2}. \tag{1}$$

For Poisson processes, there are no correlations between the point locations and thus $p(r) = \lambda\mathrm{d}x\lambda\mathrm{d}y$, which implies that $g(r) = 1$. Generally, the shape of the PCF depends on the clustering and repulsion among the points. It can be shown that as $r \to \infty$, $g(r) \to 1$. For many point sets, there is a finite r_c such that $g(r) = 1$ for $r > r_c$ and hence, most information about the point set is contained in $g(r)$ for the lower values of r.

3.2 Estimation of the Pair Correlation Function

In order to estimate the PCF, the intensity and product density should be estimated. The trivial way to estimate the intensity is dividing the number of points by the volume of the observation region, that is $\hat{\lambda} = n/|V|$, which provides an unbiased estimator. Estimating the product density is more involved and window edge effects should be taken into account. In practice, edge effects are less important when hard-core processes are considered.

In this paper, we adapt an estimator designed for isotropic distributions [Ohser and Mücklich 2000; Illian et al. 2008]. Disregarding the window edge effects, the estimator can be given by

$$\hat{g}(r) = \frac{|V|}{|\partial V_d| r^{d-1} n^2} \sum_{i \neq j} k_\sigma\left(r - d(\mathbf{x}_i, \mathbf{x}_j)\right). \tag{2}$$

Here $|\partial V_d|$ denotes the volume of the boundary of a unit sphere in a d dimensional domain, and $d(\mathbf{x}_i, \mathbf{x}_j)$ is its distance measure. We use the Gaussian kernel $k_\sigma(x) = \frac{1}{\sqrt{\pi}\sigma} e^{-x^2/\sigma^2}$ in our estimators.

This estimator makes it clear that the PCF boils down to a density estimation of the distribution of the distances. The inverse weighting by the term $|\partial V_d| r^{d-1}$ normalizes the distribution by the volume of an infinitesimal spherical shell of radius r since there will be naturally more distances at larger radiuses.

In Figure 2, PCFs of some point distributions are plotted. Unlike the spectral measures, the smoothing level we set makes the PCF estimates smooth and indistinguishable for different instances of the same distribution. This property is important if only a single distribution is all we have to extract the properties of the underlying point process.

Relation to Other Analysis Methods The power spectrum and the differential domain analysis of Wei and Wang [2011] are two proposed tools for the analysis of distributions in computer graphics. It can be shown [Wei and Wang 2011] that the power spectrum is the cosine transform of the function used for differential domain analysis. Hence, both contain the same information. The power spectrum and the differential domain function are computed using the difference vectors $\mathbf{x}_i - \mathbf{x}_j$ (assuming $\mathbb{R}^d$ for simplicity). The differential domain function can be defined in terms of the probability density function of these difference vectors, which implies that power spectrum also depends on this density. For analysis, radial averages around the origin are computed to generate $\mathbb{R} \rightarrow \mathbb{R}$ functions via a histogram with a bin for each concentric shell. The variation of the function values within each bin is defined as anisotropy, which measures the deviation of the distribution from being isotropic. Thus, the main statistics used for discriminating isotropic distributions are the radial mean plots.

The estimator $\hat{g}$ approximates the probability density function of the magnitude of the difference vectors, $||\mathbf{x}_i - \mathbf{x}_j||$. This means it contains the same information as the radial averages for the mentioned analysis methods, and one can be obtained from the other. For isotropic distributions, the magnitudes of the difference vectors are the only quantities that need to be used in a 2nd order statistic, since they are rigid motion invariant.

Parameters The most important parameter of the estimation is σ. In point process statistics, there is no general consensus on how to choose this parameter [Illian et al. 2008]. Small values will cause fluctuations in the density estimation and make the estimator change from one instance of a point distribution to another. Although this is desired for the analysis of a particular instance, obtaining a general characteristic for a particular type of distributions requires to choose a certain degree of smoothness. Another parameter is the range $[r_a, r_b]$ of the r values. This range should ideally capture enough of the characteristics to distinguish different processes without redundancy.

In order to define these parameters in relative terms, we first normalize the distances by the distance r_{max} defined as the minimum distance between pairs of points for the maximum packing of points in a given volume [Lagae and Dutré 2008; Gamito and Maddock 2009]. This normalization ensures that the number of points and the volume considered do not affect the PCFs. We then assign $\sigma = 0.25$ and $r_b = 2.5$ or 5, for all results obtained in this paper (several PCF plots in the figures have $r_b = 5$ for illustration purposes). In our experiments, values $\sigma \in [0.1, 0.5]$ and $r_b > 2.0$ provided good results. The lower limit r_a cannot be set to 0 due to the numerical problems. Although there exist solutions such as the reflection method to accurately handle the unstable range $[0, \sigma]$ [Illian et al. 2008], we refrained from using them for simplicity of the algorithms and expressions. In practice, we used values as low as $r_a = 0.01\sigma$ in our algorithms without problems. All plots in the figures also use this value for r_a, unless stated otherwise. Finally, a sampling of r should be specified to reliably capture the shape of the estimator for the given smoothness level. We use a simple regular sampling with a spacing of 0.05 between the samples.

3.3 The Pair Correlation Space

Equation 2 can be interpreted as the average of distance distributions for each point

$$\hat{g}(r) = \frac{1}{n} \sum_i \hat{g}_i(r), \tag{3}$$

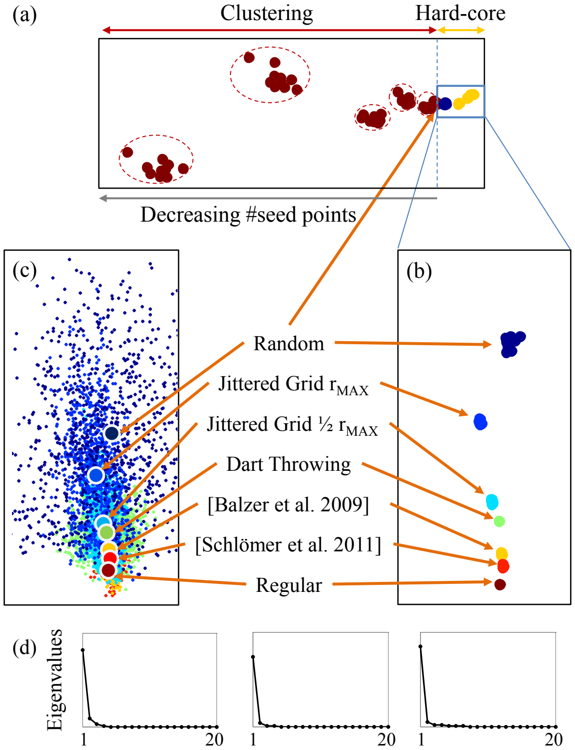


Figure 3: *(a) 2D embedding of the vectors $\bar{\phi}$ in the PCS for both clustering and hard-core point processes with 10 different distributions for each different point process. (b) Embedding of the hard-core part. (c) Embedding of the vectors ϕ_i for the hard-core distributions. (d) Left to right: the eigenvalues of the covariance matrices for the three embeddings in (a), (b), and (c), respectively.*

where $\hat{g}_i(r) = \frac{|V|}{n|\partial V_d|r^{d-1}} \sum_{j \neq i} k_\sigma\left(r - d(\mathbf{x}_i, \mathbf{x}_j)\right)$. This implies that we can consider all $\hat{g}_i(r)$'s to describe the distributional characteristics instead of just using the mean given by the PCF. Although one can work in the functional space, for the convenience of the exposition and its connection to the algorithms we use in practice, we consider a discretized PCF such that the functions are evaluated at discrete radiuses $r_1 \cdots r_{n_r}$. With this discretization, one can define the vectors $\phi_i = [\hat{g}_i(r_1) \cdots \hat{g}_i(r_{n_r})]^T$ and their mean $\bar{\phi} = \frac{1}{n}\sum_i \phi_i$ to describe the distribution of the points. We call the space where ϕ_i's live as the *pair correlation space* (PCS).

Since the PCS only depends on the distribution of the distances, it is rigid motion invariant. As long as the same discretization for r is used, one can map different point sets into the same space. Hence, each point set is described by a distribution of vectors in this space and the empirical mean of this distribution is given by $\bar{\phi}$. Precise matching of the distributional characteristics requires matching of the probability distributions of ϕ_i's. However, as we will see in the next section, the distribution of ϕ_i's, the mean $\bar{\phi}$, and *regularity* are highly correlated.

3.4 Characterization of Distributions in the PCS

In general, matching probability distributions requires parameter estimates that can be costly. In the high dimensional PCS, this will manifest itself as computational and algorithmic complexity when one tries to match point distributions. On the other hand, we expect that the mean $\bar{\phi}$ might properly capture all characteristics of point distributions since it is widely used in many fields. To pro-

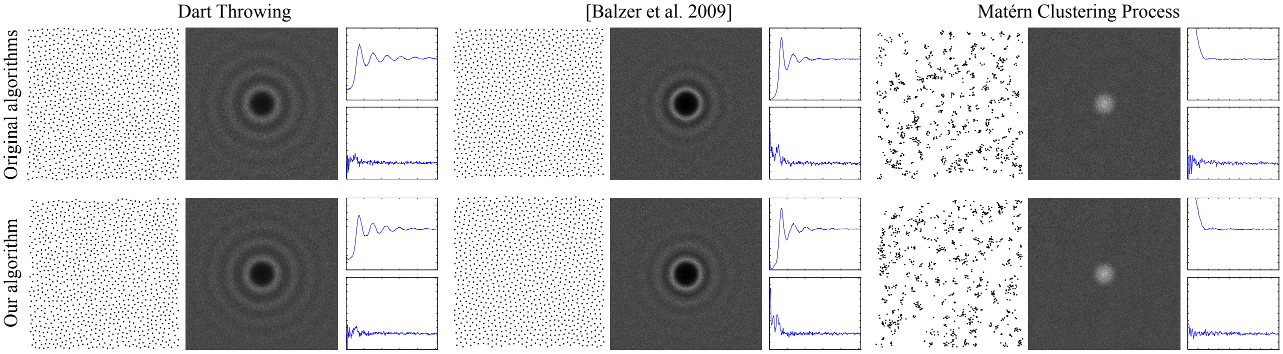


Figure 4: *Given a single example point set generated by different algorithms shown on the top, our algorithm can generate new point sets with matching spectral characteristics. Average spectra of* 10 *distributions are shown.*

vide evidence for this conjecture, we perform an analysis of the distributions of the vectors in the PCS.

In Figure 3, PCS vectors for different distributions are plotted. We embedded the vectors into a 2D space using principle component analysis computed on all ϕ_i's (c) and means $\bar{\phi}$ (a and b) of 10 instances of distributions with approximately 1000 points. To generate the clustered distributions, we used the Matérn clustering process with different seed points and radiuses of clusters. This process is simulated by seeding a number of points in space following a Poisson distribution and generating clusters of points around those seeds within a given clustering radius uniformly. The number of points in the clusters follows a Poisson distribution. The mean of this distribution is set to the number of points desired (1000) divided by the number of seed points.

The magnitude of the eigenvalues (d) clearly show that the data can be well explained using only 2 dimensions. Figure 3 (a) illustrates that there is a clear separation between clustering and hard-core processes, with the random distribution on the border. For clustering processes, decreasing number of seed points for clusters results in scattering of $\bar{\phi}$'s. For hard-core processes, going from random to more regular point distributions, the variance of the ϕ_i (c) and scattering of $\bar{\phi}$ vectors (b) diminish and in particular, for the regular grid, the ϕ_i distribution becomes a single spike. Hence, regularity and uniformity in point distributions result in less variance of the vectors in the PCS.

Another interesting property of the PCS is that the $\bar{\phi}$'s effectively lie on a line, as can be observed from the magnitude difference between the first eigenvalue of the covariance matrices and the others in Figure 3 (d), and also from the embeddings in Figure 3 (a), (b), and (c). For the hard-core distributions, this line extends from the $\bar{\phi}$ for random distributions to that of the regular grid. The parameter of the means on this line correctly determines the order of regularity in the point distributions. In particular, Schlömer et al.'s algorithm [2011] is found to be the most regular, followed by Balzer et al.'s algorithm [2009] and then dart throwing and different levels of jittering. This ordering matches the regularity and packing density observed for these algorithms in practice [Schlömer et al. 2011].

This analysis gives us a tool to characterize the distributions generated by different processes or algorithms. The means $\bar{\phi}$ for a distribution are clustered with variance getting smaller as the means for different distributions start to approach each other. Hence, the $\bar{\phi}$'s are sufficient to discriminate different types of distributions. Furthermore, the means approximately lie on a 1D line in the high dimensional PCS, which provides an easy quantitative measure of the *closeness* of distributions generated by different algorithms. We quantify such a measure in the next section.

3.5 A Measure of Irregularity

Regularity in a point distribution can be intuitively described as the indistinguishability of the neighborhoods of points. As the distribution becomes more random, or exhibit clearly distinctive structures at different locations in space, the neighborhoods deviate more and more from each other. Following this intuition, irregularity in a distribution can be described by the vectors ϕ_i, which effectively describe the neighborhoods of the points in terms of the distance distributions.

We illustrate that the variance in ϕ_i's correctly captures irregularity in Figure 3 (c). When points are regularly distributed, the components of different ϕ_i's match since the distance distribution around each point $\mathbf{x}_i$ is the same. As more randomness is added, this regularity degrades and in the case of complete randomness, the vectors no longer correlate with each other. Clustering processes further increase irregularity by introducing different structures at different points.

This observation can be quantified as an irregularity measure as follows:

$$p_k = \frac{1}{n} \sum_i \left(\phi_{ik} - \bar{\phi}_k\right)^2. \qquad (4)$$

This measure describes the observed irregularity in the point set at different radiuses r_k. In practice, we normalize this measure by the irregularity of the empirical random distribution obtained by averaging 10 p's. The p statistics (irregularity) for different distributions are plotted in Figure 2. The level of irregularity in hard-core and clustering processes considered exactly match the order and scattering of the means $\bar{\phi}$ in Figure 3 (a) and (b), which shows that p is also an accurate measure of closeness in the PCS.

4 Synthesis of Point Distributions

All statistics for point processes are aimed at providing a good summary of the distributional characteristics. Unlike other measures such as periodograms, the PCF has a simple form and interpretation directly linked to the distribution of the distances between pairs of points. The PCS analysis we presented also shows that it characterizes point distributions well. In this section, we build on these properties and propose two simple synthesis algorithms that use the PCF as a global statistic. The first one relies on simple random sampling and can provide an initial distribution for the second relaxation based method. We assume that a target PCF $g_0(r)$ is computed using one or more examples, or given by the user. The goal of the algorithms is to transform an arbitrary input point set such that its PCF matches the target PCF.

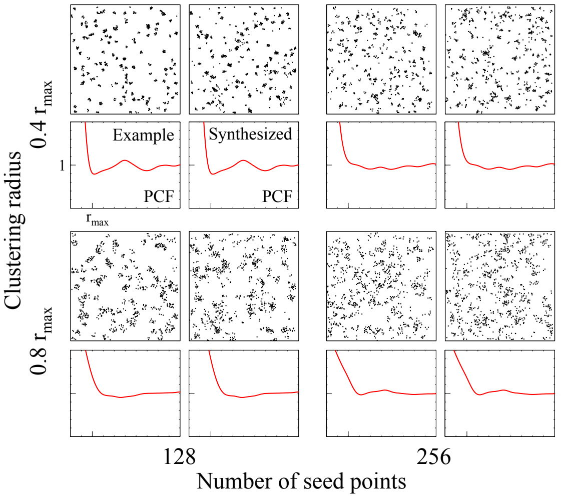


Figure 5: *Reconstruction of clustering patterns generated by the Matérn clustering process with different number of seeds and clustering radiuses.*

4.1 Generalized Dart Throwing

Our first algorithm generalizes the well-known dart throwing algorithm. In the original algorithm, at each step a random point is generated. If the distance of this point to the closest point in the set of already accepted points is smaller than desired, it is rejected and otherwise it is accepted. From this definition of the algorithm, it is clear that only the lower end of the PCF is used in the decision step. In order to extend this algorithm for a given target PCF g_0, one can simply impose the condition that at any given iteration, $g(r_k) \leq g_0(r_k)\ \forall r_k$. Here, $g(r)$ is normalized by the target number of points.

The pseudo code of the resulting algorithm is provided in Algorithm 1. The algorithm reduces to standard dart throwing if the range upper limit r_b is set to the hard-core radius between the points. In practice, we use a parameter ϵ and relax the condition as $\max_k(g(r_k) - g_0(r_k)) \leq \epsilon$. The ϵ is changed at each iteration with a user defined function. We used a simple function $f_\epsilon(\text{iteration}) = c$ iteration, for some constant c. Using this relaxed version avoids the expected high rejection rates and deadlocks due to infeasible configurations but also distorts the desired characteristics. Nevertheless, this algorithm provides a very good initial distribution for our gradient descent based fitting algorithm.

Algorithm 1 Generalized Dart Throwing

```
  Input: Target $g_0(r_k)$, number of points $n_T$, dimension $d$, volume $V$
  Output: Point Set $P$
  $P = \emptyset$, $g(r_k) = 0\ \forall k$, iter $= 0$
  while ($|P| < n_T$)
    $\epsilon \leftarrow f_\epsilon$(iter)
    generate a random point $\mathbf{x}$ in V
    update $g(r_k)\ \forall k$ for $\mathbf{x}$ using equation 6
      with $n = n_T$ in the normalization
    if ($\max_k(g(r_k) - g_0(r_k)) \leq \epsilon$)
      add $\mathbf{x}$ to $P$
    else
      restore previous $g(r_k)\ \forall k$
    iter $\leftarrow$ iter $+ 1$
```

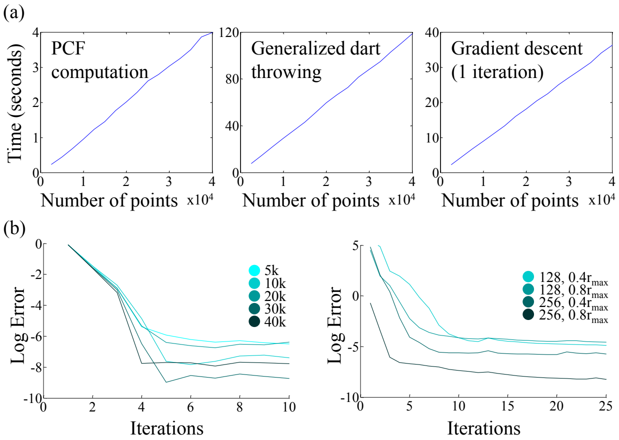


Figure 6: *(a) Running times of the algorithms as a function of the number of points. (b) Convergence of the gradient descent algorithm for (left) hard-core distributions with different number of output points, and (right) for the distributions in Figure 5.*

4.2 PCF Fitting by Gradient Descent

This fitting algorithm takes a random point set, or the output of Algorithm 1 as the input, and tries to minimize the least squares fitting error $E(\mathbf{x}_1, \cdots, \mathbf{x}_n) = \int_0^\infty (g(r) - g_0(r))^2\, dr$. This corresponds to a least squares fitting of the means $||\bar{\phi} - \bar{\phi}_0||^2$ in the PCS. Since we discretize r to r_k's, the integral turns into a sum and the normalized gradient with respect to a point $\mathbf{x}_m$ can be computed as follows:

$$\Delta_m = -\frac{\sum_{i \neq m} \mathbf{u}_{mi} w_{mi}}{|\sum_{i \neq m} w_{mi}|}, \tag{5}$$

$$w_{mi} = \sum_k \frac{g(r_k) - g_0(r_k)}{r_k^{d-1}} (d_{mi} - r_k)\, k_\sigma(d_{mi} - r_k).$$

Here, $\mathbf{u}_{mi} = \nabla_{\mathbf{x}_m} d_{mi}$ (the unit vector from $\mathbf{x}_i$ to $\mathbf{x}_m$ for Euclidean spaces) and $d_{mi} = d(\mathbf{x}_m, \mathbf{x}_i)$, which is assumed to be symmetric for brevity of the expressions.

At each iteration, each point $\mathbf{x}_m$ is moved by a gradient descent $\mathbf{x}_m^{k+1} = \mathbf{x}_m^k - \lambda\Delta_m$. To determine the step size λ, the algorithm performs a simple search by taking 5 different λ values 10^{-i} $i = 1, \cdots, 5$ and accepts the one that causes the most reduction in the error. When a random point set is used as the input, reaching convergence takes longer, but the characteristics of the output point sets are not affected significantly. However, computing the initial point set with Algorithm 1 improves convergence such that the number of iterations are not affected by the number of points [Schmaltz et al. 2010].

4.3 Analysis

Updating the PCF In both algorithms, the PCFs should be updated after each operation of point insertion or movement. The effect of a point $\mathbf{x}_m$ on the PCF can be computed using the following formula

$$\delta_k(\mathbf{x}_m) = \frac{|V|}{|\partial V_d| r_k^{d-1} n^2} \sum_{i \neq m} k_\sigma(r_k - d_{im}) + k_\sigma(r_k - d_{mi}). \tag{6}$$

As an example, when moving a point, $\delta_k(\mathbf{x}_{new}) - \delta_k(\mathbf{x}_{old})$ should be added to the PCF.

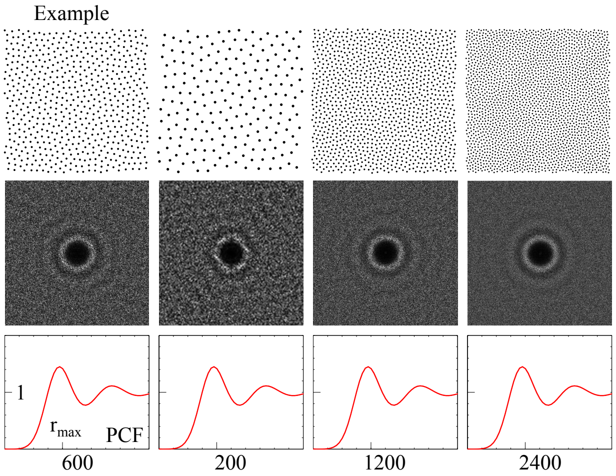


Figure 7: *Given an example point set with 600 points, our algorithm can generate point sets with any number of points.*

Time Complexity The time complexity of the algorithms depends on the number of iterations to convergence, number n_s of r_k samples, and the neighborhoods used when computing the gradient of the PCF and the PCF itself. The maximum neighborhood size depends on two factors, σ and r_b, and can be precisely given as $r_{max}(r_b + \varsigma\sigma)$ where ς is the cutoff factor used for the Gaussian kernels. For isotropic point sets, $r_{max} = c_d\lambda^{-1/d}$ for a constant c_d that depends only on the dimension. The number of points in a hypersphere of radius r in d dimensions is given by $r^d|V_d|\lambda$ with $|V_d|$ the volume of a unit sphere. Substituting the expression of the maximum neighborhood size, we get that number of points in the neighborhoods is $\alpha_d^d|V_d|$ with $\alpha_d = c_d(r_b + \varsigma\sigma)$, which only depends on the dimension d. Hence, updating the PCF or computing the gradient of the PCF for a single point takes constant time with respect to the number of points, resulting in $O(n)$ complexity for all algorithms. We verify the linear complexity of our algorithms and illustrate the convergence of the gradient descent algorithm in Figure 6 (a) and (b), respectively. Convergence of the gradient descent algorithm is independent of the number of points and 5 iterations for hard-core and 10 for clustering distributions are sufficient to get accurate characteristics.

4.4 Extensions

Multi-class Sampling In point process statistics, marked point processes are used to describe point sets where the points have associated properties in addition to locations [Illian et al. 2008]. Each point can have a qualitative, discrete, or continuous mark. When discrete marks are used to indicate different classes of points, the pair correlation functions are extended to include correlations between classes such that one has the interclass pair correlations $g_{ij}(r)$ for all classes i and j. The exact form of $g_{ij}(r)$ depends on the mark correlation functions chosen. Since interclass correlations are coupled, minimization of an energy that involves all $g_{ij}(r)$'s will be unnecessarily complex. Instead of using all interclass correlations, we include only the intraclass pair correlations in the energy function to be minimized such that $E_{total} = E + \sum_i E_i$, where E is defined in Section 4.2 and E_i is computed using only the points in class i. In our experiments, this produced comparably accurate reconstructions and also reduced the time complexity and number of iterations of the algorithms.

Adaptive Sampling By adjusting the distance measure used, one can easily extend the algorithm to generate adaptive isotropic or anisotropic samplings. Here, we concentrate on adaptive isotropic samplings where the distance is given by $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||/s(\mathbf{x})$ for a scaling function s. Ignoring the change in the scaling function [Fattal 2011], the expressions for the PCF and its gradient change trivially. Given an arbitrary importance function $f(\mathbf{x})$, the scaling function can be set as $s(\mathbf{x}) \propto f(\mathbf{x})^{-1/d}$.

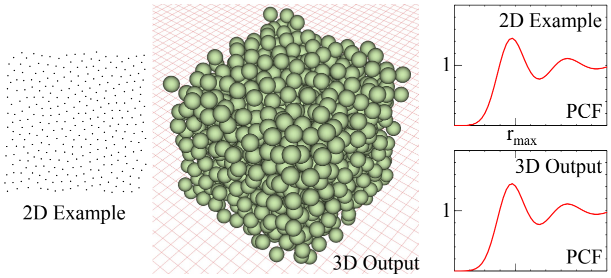


Figure 8: *A 3D point set generated using a given 2D example and their PCFs.*

5 Results

As the first experiment, we tested if our algorithms can reproduce the characteristics of distributions generated by existing algorithms. In Figure 4, top, we show averaged spectra of 10 distributions generated by dart throwing, Balzer et al.'s algorithm [2009], and Matérn clustering process. For each algorithm, we used a single distribution generated as the example (shown in the figure), and ran our algorithm 10 times using the same example. The spectra of the generated distributions (Figure 4, bottom) almost exactly match those of the distributions by the original algorithms. As a further test, we generated points from the Matérn clustering process with different number of seed points and clustering radiuses in Figure 5. For all cases, our algorithm could accurately reproduce the same characteristics. We used $r_b = 5$ for the clustering processes to get more accurate results.

Multi-class distributions can also be accurately reconstructed by our method. This is illustrated in Figures 1 (right), 11, and 12 (a, b). Intraclass as well as overall PCFs are well-preserved. The distributions of points in different classes can have very different characteristics. In Figure 1 (right), although a small number of points from each class are used as examples, accurate results are obtained for all four classes.

As illustrated in Figure 7, our algorithm does not need the example sample size to be the same as the output size. An example point set with 600 points is sufficient to generate outputs of various sizes with identical characteristics. Yet, since the PCF depends on a density estimation of distances, for extremely small point sets, there can be unwanted fluctuations. We experimentally found out that for $r_b = 2.5$, example sets of around a hundred points are sufficient to produce accurate PCFs for 2 dimensions. Dimensionality of the example and output point sets can also be set arbitrarily since the PCF is defined for any dimensions. We show an example in 3 dimensions in Figure 8.

By interpolating the PCFs, a family of distributions with novel characteristics can be obtained. Since the PCFs effectively lie on a line as shown in Figure 3, simple linear interpolation can generate valid PCFs from which distributions can be synthesized. An interesting application of this interpolation is to combine hard-core and clustering distributions. Results of such an experiment are shown in Figure 9. The PCF $\bar{\phi}_0$ of a hard-core (Balzer et al.'s algorithm [2009]) and that $\bar{\phi}_1$ of a clustering (Matérn clustering process) distribution

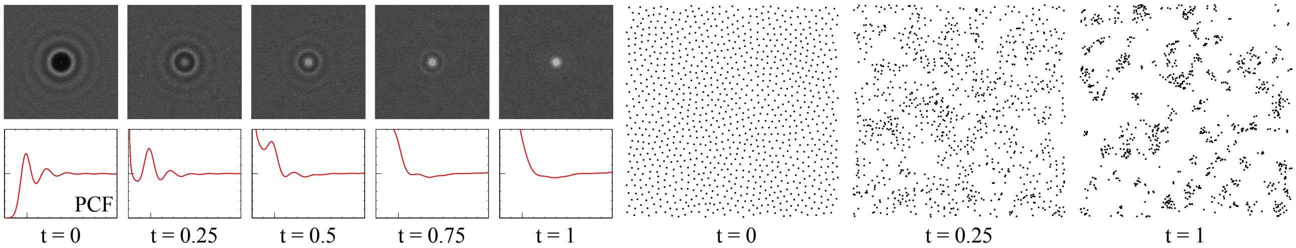


Figure 9: *By interpolating PCFs of point sets generated by Balzer et al.'s algorithm [2009] and Matérn clustering process, distributions with novel characteristics can be obtained. Average spectra of* 10 *distributions are shown.*

are interpolated as $\bar{\phi}_0(1-t)+\bar{\phi}_1 t$ and new distributions having the interpolated PCFs are synthesized (with $r_b = 5$). As the parameter t is changed, the evolution of the spectra and point distributions can be clearly seen.

Finally, density adaptation of our algorithm is illustrated in Figure 10 for different number of points. Given an example sample distribution and an input density, our algorithm can generate adaptive samplings with characteristics matching the example, and density matching the input density.

6 Applications

6.1 Synthesizing Real World Structures

Many entities in the real world have distributions with particular characteristics. Often, these distributions have various scales of clustering and repulsion, and interclass as well as intraclass interactions, leading to interesting patterns. We applied our algorithms to learn PCFs of real world distributions and generate new distributions with the same characteristics.

An example synthesis result is shown in Figure 1 (right), where a photo of candies with four different colors is used to generate the example PCFs (the black curve in each PCF graph is the PCF of all points). As shown in the PCF graphs, our algorithms are able to precisely match the characteristics of all the distributions. This is also apparent in the rendering, where red candies are in clusters, and all candies as well as candies of the same colors are distributed with a hard-core distance between them. A further example is shown in Figure 11, where data gathered from a real distribution of oak and beech trees [Pommerening 2002] is used as the input example. Our synthesis algorithm can precisely reproduce the characteristics, which allows us to construct and render a bigger forest following the real world distributions.

6.2 Turbulent Fluid Simulations

In fluid simulations, methods for detail enhancement have become popular in recent years. These methods augment a low-resolution base simulation with synthetic sub-grid detail, by e.g. applying a curl noise turbulence texture [Kim et al. 2008]. While the detail structure is known for the special case of homogeneous, fully-developed turbulence in which Kolmogorov's law holds, turbulent details in real flows tend to be more complex, due to anisotropic effects and transition processes. It would therefore be interesting to extract the fingerprint of a set of given reference turbulent flow simulations, and use this as a basis for detail synthesis on top of other arbitrary flows.

Our method allows us to achieve this transfer directly in a Lagrangian manner, using a very compact fingerprint. A small vorticity field from a real high-resolution simulation is taken as the input (Figure 12 (a), top). This field is positive (red) in some regions and negative (blue) in others. The sign determines the direction of rotation (clockwise or counterclockwise), and the absolute value determines how strong the vorticity is, i.e. how much the fluid is rotating in that region. The field is separated into a negative and a positive field, and for each, a set of vortex particles is randomly placed according to the absolute value of the density field (Figure 12 (a), bottom). Next, we compute the PCFs for this two-class distribution of negative and positive particles, and use our multi-class synthesis algorithm to generate a large distribution of arbitrary size and resolution (Figure 12 (b)). Finally, these synthesized vortex particles act as our turbulence representation [Selle et al. 2005b], which means they induce small-scale rotations to the flow. Figure 12 (c) illustrates the base simulations and the synthesized flows using the generated vortex particles. Using fingerprints recorded from different reference simulations, we can also obtain different turbulence strength and behavior as illustrated in Figure 12 (d).

7 Conclusions and Future Directions

We introduced novel analysis and synthesis techniques for point distributions with general characteristics. We also presented several experiments and example applications where our techniques can be useful. We believe our methods can be extended in several ways and be utilized in many other interesting applications.

Limitations A fundamental limitation of utilizing the PCF is that it is only a second-order statistic depending on pairs of points and hence cannot uniquely characterize a given point pattern. However, as explained in the literature [Illian et al. 2008], statisticians and physicists regard 2nd order statistics and in particular the PCF as the most informative and for most distributions sufficient for unique determination (this is called the "second order dogma" [Illian et al. 2008]). This is supported by a theorem by Boutin et al. [Boutin and Kemper 2004; Boutin and Kemper 2007]. The theorem states that the set of point distributions that are not uniquely determined by their distance distributions has Lebesgue measure zero in the nd dimensional space of point configurations, when $n \geq \max(3, d+2)$ for n points in d dimensions. Thus, the distribution of pairwise distances, and hence the PCF, uniquely determines most of the point distributions.

In practice, we observed that point distributions with highly regular structures are harder to synthesize, as illustrated for the hexagonal and regular grids in Figure 13. Furthermore, the Poisson disk radiuses [Lagae and Dutré 2008] of the generated blue-noise distributions are slightly lower than expected. For example, if dart throwing or Balzer et al.'s algorithm [2009] is used to produce an example distribution, the Poisson disk radiuses are in the range $[0.67, 0.75]$ and $[0.73, 0.74]$ with an average of 0.7031 and 0.7352, respectively. In comparison, the example distributions we used

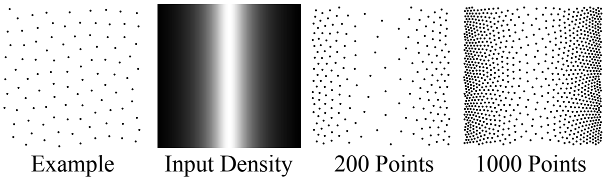


Figure 10: *An example distribution and input density can be combined for adaptive sampling.*

have 0.7137 and 0.7765 as Poisson disk radiuses. Due to the global nature of the PCF, our algorithms cannot precisely see the minimum of the distances, in other words, summing many small distances reduces the effect of a single distance. In addition, setting a lower limit r_a due to numerical instability causes the algorithms to not see some of the low distances. To solve this problem, a minimum distance constraint can be imposed, and estimation methods for lower r values [Illian et al. 2008] can be utilized.

Future Directions Although we presented results based on discrete marks, i.e. classes, the synthesis algorithms can also be extended to reconstruct distributions of points with continuous marks. This will allow to reconstruct various quantitative properties of objects such as length, size or age along with their locations. Another important direction is using space-time processes and associated statistics [Illian et al. 2008] to learn, categorize, and synthesize realistic movements of objects such as humans or animals along with marks for further properties such as gazing direction. Space-time processes can be further used to generate point samplings for dynamic data such as videos or simulations. By approximating the distance metric of a manifold [Wei and Wang 2011; Li et al. 2010], our methods can also be extended to curved spaces. Running times of our algorithms can be significantly improved by integrating parallelization [Wei 2008; Schmaltz et al. 2010] or tiling [Ostromoukhov et al. 2004; Kopf et al. 2006] approaches. Finally, we believe that many ideas from the point process statistics literature can be extended and adapted for various applications in computer graphics to help us better understand and computationally mimic the nature.

Acknowledgments

We would like to thank the reviewers for their insightful comments and suggestions, and Dr. Tobias Pfaff for his help with the fluid simulations and renderings. A. Cengiz Öztireli is supported by the Swiss National Science Foundation (grant No. 200021-132438 / 1).

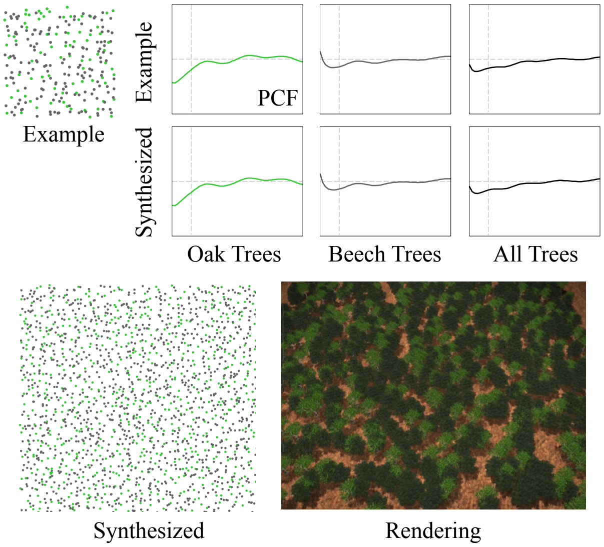


Figure 11: *Given a small example distribution of two types of trees, our algorithm can generate a bigger forest with the same characteristics. For the PCF graphs, $r \in [\sigma, 5]\, r_{max}$.*

References

ALLIEZ, P., MEYER, M., AND DESBRUN, M. 2002. Interactive geometry remeshing. *ACM Trans. Graph. 21*, 3, 347–354.

BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graph. 28*, 3, 86:1–8.

BOUTIN, M., AND KEMPER, G. 2004. On reconstructing n-point configurations from the distribution of distances or areas. *Adv. Appl. Math. 32*, 709–735.

BOUTIN, M., AND KEMPER, G. 2007. Which point configurations are determined by the distribution of their pairwise distances? *International Journal of Computational Geometry & Applications 17*, 31–43.

COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1, 51–72.

DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., MĚCH, R., PHARR, M., AND PRUSINKIEWICZ, P. 1998. Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH 98, 275–286.

DUNBAR, D., AND HUMPHREYS, G. 2006. A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graph. 25* (July), 503–508.

EBEIDA, M. S., PATNEY, A., MITCHELL, S. A., DAVIDSON, A., KNUPP, P. M., AND OWENS, J. D. 2011. Efficient maximal poisson-disk sampling. *ACM Trans. Graph. 30*, 4.

FATTAL, R. 2011. Blue-noise point sampling using kernel density model. *ACM Trans. Graph. 30*, 4 (July), 48:1–48:12.

GAMITO, M. N., AND MADDOCK, S. C. 2009. Accurate multidimensional poisson-disk sampling. *ACM Trans. Graph. 29* (December), 8:1–8:19.

ILLIAN, J., PENTTINEN, A., STOYAN, H., AND STOYAN, D., Eds. 2008. *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, Ltd.

JODREY, W. S., AND TORY, E. M. 1985. Computer simulation of close random packing of equal spheres. *Phys. Rev. A 32* (Oct), 2347–2351.

JONES, T. R. 2006. Efficient generation of poisson-disk sampling patterns. *journal of graphics, gpu, and game tools 11*, 2, 27–36.

KERSCHER, M. 2001. Constructing, characterizing, and simulating gaussian and higher-order point distributions. *Phys. Rev. E 64* (Oct), 056109.

KIM, T., THÜREY, N., JAMES, D., AND GROSS, M. 2008. Wavelet turbulence for fluid simulation. *ACM Trans. Graph. 27*, 3 (Aug.), 50:1–50:6.

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive wang tiles for real-time blue noise. *ACM Trans. Graph. 25* (July), 509–518.

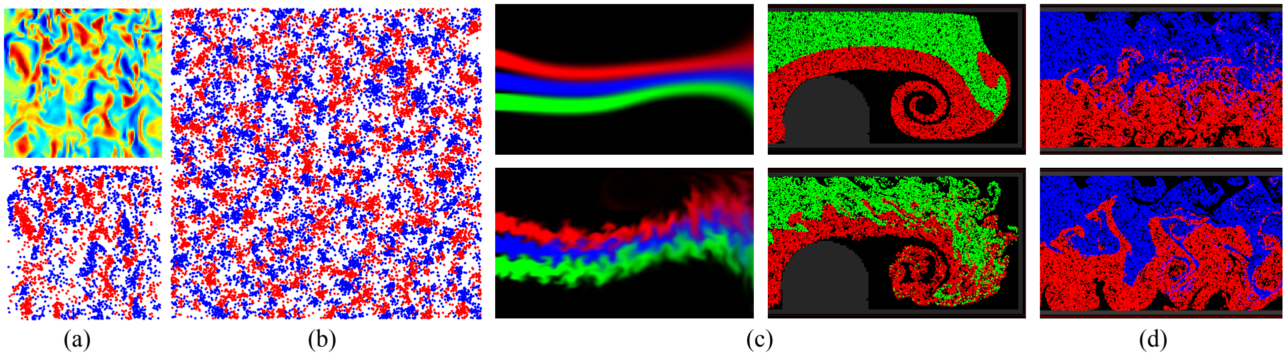


Figure 12: *(a) Top: a vorticity field extracted from a reference high-resolution simulation, and bottom: an example distribution with negative (blue) and positive (red) particles generated according to the field. (b) Synthesized particle distribution. (c) Top: base simulations, and bottom: base simulations with the synthesized vorticity. (d) Using different example fields leads to different behavior.*

LAGAE, A., AND DUTRÉ, P. 2006. Poisson sphere distributions. In *Vision, Modeling, and Visualization 2006*, Akademische Verlagsgesellschaft Aka GmbH, Berlin, L. Kobbelt, T. Kuhlen, T. Aach, and R. Westermann, Eds., 373–379.

LAGAE, A., AND DUTRÉ, P. 2008. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (March), 114–129.

LEWIS, J. P. 1989. Algorithms for solid noise synthesis. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH 89, 263–270.

LI, H., WEI, L.-Y., SANDER, P. V., AND FU, C.-W. 2010. Anisotropic blue noise sampling. *ACM Trans. Graph. 29*, 6 (Dec.), 167:1–167:12.

LLOYD, S. 1982. Least squares quantization in pcm. *Information Theory, IEEE Transactions on 28*, 2 (mar), 129 – 137.

MA, C., WEI, L.-Y., AND TONG, X. 2011. Discrete element textures. *ACM Trans. Graph. 30*, 4 (Aug.), 62:1–62:10.

MCCOOL, M., AND FIUME, E. 1992. Hierarchical poisson disk sampling distributions. In *Proceedings of the conference on Graphics interface '92*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 94–105.

NARAIN, R., GOLAS, A., CURTIS, S., AND LIN, M. C. 2009. Aggregate dynamics for dense crowd simulation. *ACM Trans. Graph. 28*, 5 (Dec.), 122:1–122:8.

OHSER, J., AND MÜCKLICH, F., Eds. 2000. *Statistical Analysis of Microstructures in Materials Science*. John Wiley and Sons, Ltd.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph. 23* (Aug.), 488–495.

PFAFF, T., THUEREY, N., SELLE, A., AND GROSS, M. 2009. Synthetic turbulence using artificial boundary layers. *ACM Trans. Graph. 28*, 5 (Dec.), 121:1–121:10.

POMMERENING, A. 2002. Approaches to quantifying forest structures. *Forestry 75*, 305–324.

SCHLÖMER, T., HECK, D., AND DEUSSEN, O. 2011. Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, ACM, New York, NY, USA, HPG '11, 135–142.

SCHMALTZ, C., GWOSDEK, P., BRUHN, A., AND WEICKERT, J. 2010. Electrostatic halftoning. *Computer Graphics Forum 29*, 8, 2313–2327.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (July), 910–914.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (July), 910–914.

TORQUATO, S., Ed. 2002. *Random Heterogenous Materials. Microstructure and Macroscopic Properties.* Springer-Verlag, New York.

ULICHNEY, R. 1987. *Digital halftoning*. MIT Press, Cambridge, MA, USA.

WEI, L.-Y., AND WANG, R. 2011. Differential domain analysis for non-uniform sampling. *ACM Trans. Graph. 30*, 4 (July), 50:1–50:10.

WEI, L.-Y. 2008. Parallel poisson disk sampling. *ACM Trans. Graph. 27* (August), 20:1–20:9.

WEI, L.-Y. 2010. Multi-class blue noise sampling. *ACM Trans. Graph. 29* (July), 79:1–79:8.

WHITE, K., CLINE, D., AND EGBERT, P. 2007. Poisson disk point sets by hierarchical dart throwing. In *Interactive Ray Tracing, 2007. RT '07. IEEE Symposium on*, 129 –132.

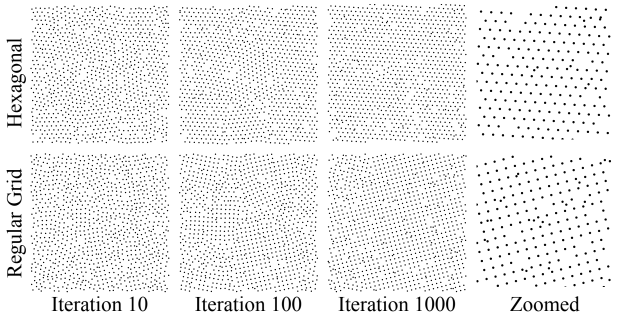


Figure 13: *Distributions with highly regular structures are harder to synthesize.*