Blue-Noise Point Sampling using Kernel Density Model

Raanan Fattal*
Hebrew University of Jerusalem, Israel

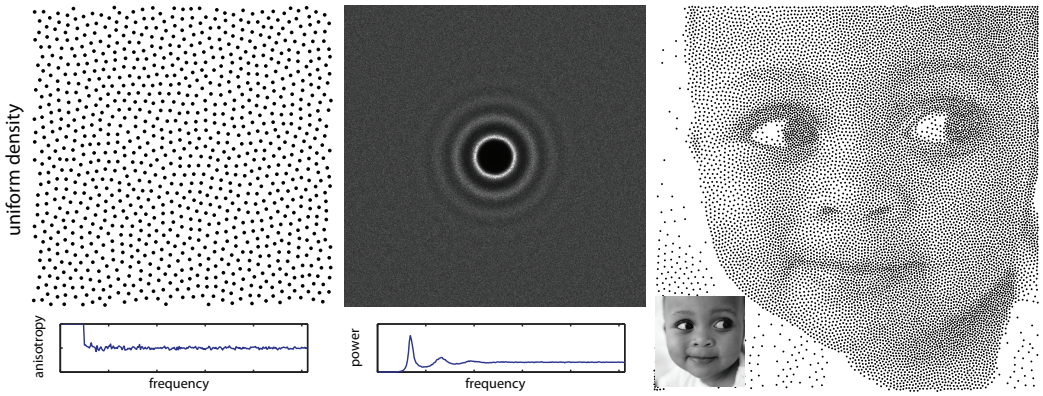

Figure 1: *Our result on uniform stochastic point distribution with isotropic spectrum and spatially-varying point density (13,000 points).*

Abstract

Stochastic point distributions with blue-noise spectrum are used extensively in computer graphics for various applications such as avoiding aliasing artifacts in ray tracing, halftoning, stippling, etc. In this paper we present a new approach for generating point sets with high-quality blue noise properties that formulates the problem using a statistical mechanics interacting particle model. Points distributions are generated by sampling this model. This new formulation of the problem unifies randomness with the requirement for equidistant point spacing, responsible for the enhanced blue noise spectral properties. We derive a highly efficient multi-scale sampling scheme for drawing random point distributions from this model. The new scheme avoids the *critical slowing down* phenomena that plagues this type of models. This derivation is accompanied by a model-specific analysis.

Altogether, our approach generates high-quality point distributions, supports spatially-varying spatial point density, and runs in time that is linear in the number of points generated.

Keywords: Poisson disk distribution, stochastic sampling, importance sampling, blue noise, image synthesis, and anti-aliasing.

Links: DL PDF WEB

*e-mail: raananf@cs.huji.ac.il

1 Introduction

Stochastic point arrangements, or point distributions, are used in various computer graphics applications. Originally these distributions were used to overcome the visually disturbing aliasing artifacts, such as Moiré patterns, that arise in regular sampling when the grid spacing fails to meet the signal's Nyquist rate. Dippé et al. [1985] and Cook [1986] analyze the spectral properties of different stochastic point sampling procedures and demonstrate their ability to produce perceptually superior images in which the spurious aliasing patterns of regular sampling are converted into featureless noise. Among these stochastic point distributions, the Poisson disk distribution (a.k.a. minimal-distance Poisson) stands out for the *blue noise* characteristics its spectrum possess. Such distributions accurately capture the visually important lower-end frequency content of a signal and scatter the higher frequencies into broadband random noise. Interestingly, Yellott [1983] found that the arrangement of photoreceptors in the extra-foveal part of the human retina possesses blue noise characteristics. Since then, stochastic blue-noise point distributions were used for various other applications such as populating plants in virtual ecosystems [Deussen et al. 1998], and basis functions in procedural textures [Cohen et al. 2003], halftoning and stippling [Deussen et al. 2000; Secord 2002], illumination quadrature [Kollig and Keller 2003], and geometry processing [Surazhsky et al. 2003]. Recently, the generation of stochastic point distributions was extended to arbitrary manifold surfaces [Bowers et al. 2010], multiple classes of samples [Wei 2010] and anisotropic samples [Li et al. 2010].

The use of the quantization optimization method by Lloyd [1982], first proposed by McCool et al. [1992], became a popular mean of enhancing the blue noise properties of a given distribution. Lloyd's method is a deterministic iterative procedure that spreads the points more evenly in space. It is commonly used as a post-processing step. Being an optimization procedure, Lloyd's method is known to converge to compact piecewise hexagonal patterns and reintroduces periodicity to the sampling pattern. Therefore, only a small number of iterations is used in practice. However, as pointed out by Balzer et al. [2009], there is no known satisfactory termination criterion for Lloyd's method—a problem that is most acute when patterns of spatially-varying density are sought for. In their work, Balzer et al.

present a new variant of Lloyd's method in which the Voronoi cells, associated with each point, are constrained to have an equal capacity. This process, which runs until convergence, achieves superb spectral enhancement and closely matches the target density. This performance comes at a computational cost; each iteration runs in $O(n^2)$ time where n is the number of points generated.

In this paper we propose a new approach for generating stochastic blue-noise point distributions that formulates the problem as sampling a statistical mechanics interacting particle model. In this this model a radially-symmetric kernel function is placed around every point to produce an approximate density function. The difference between this approximation and the given target point density function assigns an energy value to the points configuration. Rather than minimizing this energy, we use it to define a Boltzmann-Gibbs statistical model that introduces randomness. Thus, this framework unifies randomness with the requirement for uniform point spacing that achieves the enhanced blue noise spectral properties.

Drawing samples from such high-dimensional statistical models is a non-trivial task and standard samplers suffer from a problem known as *critical slowing down* when applied to this type of models. In this problem, the number of stochastic relaxation sweeps needed to achieve a single independent sample, grows as a function of the number of variables in the system and leads to a quadratic computational complexity. We analyze this problem, in the specific context of our model, and develop a novel multi-scale sampling scheme that constructs the samples by adding points progressively. This strategy eliminates the critical slowing down altogether and generates high-quality stochastic point distributions in a time that is linear in the number of points generated.

Our approach does not involve computing Voronoi diagrams, Delaunay triangulations, nor any type of nearest-point searche, and produces more than 15,000 spatially-varying points per second on a single CPU core. We validate the quality of the resulting point distributions against ones produced by state-of-the-art methods.

2 Background

The generation of stochastic sampling patterns became an active topic of research in computer graphics since the early eighties when several works [Whitted 1979; Cook et al. 1984; Dippé and Wold 1985; Cook 1986; Mitchell 1987] demonstrated the advantages in using randomized sample locations over regular grids in the context of ray tracing. In particular, Dippé et al. [1985] and Cook [1986] study three different random point distributions: *Poisson* sampling, where points are uniformly distributed in space, *Jittered* sampling, where points are placed on a regular grid (usually rectangular) with random offsets, and the *Poisson disk* sampling, where points are also uniformly distributed in space but do not allow an overlap between disks, of some predefined radius, that surround them. Both papers point out the Poisson disk model as the superior over the other two as it has a large low-energy annulus around the origin in Fourier space and is approximately uninform at higher frequencies. Spectrum with this shape is known as *blue-noise* and sampling patterns that possess it accurately capture the lower end of a signal's spectrum and restrict the scattering to the higher frequency band.

Since then, blue-noise stochastic point distributions were used for various other applications that we mentioned earlier. Given this demand, the generation of such point distributions received much attention over the last two decades and evolved in several directions; very efficient sampling techniques were developed for sampling the true Poisson disk distribution [White et al. 2007], relaxed versions of it were introduced [McCool and Fiume 1992; Wei 2008], and various tiling-based accelerations were proposed [Shade et al. 2000; Lagae and Dutré 2008]. Here we review some of this literature.

Cook [1986] describes a simple yet expensive rejection-based procedure for drawing samples from the Poisson disk distribution, where a new point is proposed from a uniform (Poisson) distribution and is accepted only if its surrounding disk does not overlap any disk around the points laid so far. This procedure is known as *dart throwing* and terminates once no point can be further added or, in practice, when the rejection count becomes large. To avoid the large number of rejections Jones [2006] encodes the area not covered by any disk via a tree, constructed using a Voronoi diagram, and generate n points in $O(n \log n)$ running time. Dunbar et al. [2006] achieve a similar asymptotic running time using a unique data structure called *scalloped sectors* to encode the vacant space. By drawing proposal points from a model that keeps track of the vacant space using a quadtree, White et al. [2007] generate samples from the true Poisson disk model in $O(n)$ time. Recently, Gamito et al. [2009] described an accurate multi-dimensional Poisson-disk sampling method that has $O(n \log n)$ time and space complexity.

Other point distributions, beside jittered grids, were proposed as alternatives to the Poisson disk model. These distributions also possess blue noise spectral characteristics, they are easier to generate and control, and some of the methods can adapt their point density to match arbitrary spatially-varying density function. Mitchell [1991] describes a variation of dart throwing in which several points are thrown at each step and only the one, farthest from the existing points, is added. McCool and Fiume [1992] also relax the dart throwing method and reduce the disk radius while constructing the pattern. These algorithms avoid the large number of rejections in dart throwing and allow a precise control over the number of points generated. Without using any accelerations for the nearest-point search, these methods run in $O(n^2)$ time. By restricting the addition of new points to the immediate boundary of the existing ones, Dunbar et al. [2006] construct distributions that obey a strict minimal-distance in linear-time $O(n)$. A parallel algorithm based on domain subdivision is proposed by Wei [2008]. Both this method and the one by McCool and Fiume can adapt their point density given density functions.

The methods discussed thus far either constrain a minimal distance between every pair of points or follow a greedy strategy that maximizes this distance. A full optimization of the inter-point distances is a coupled n-body problem which was first considered by McCool and Fiume [1992]. McCool and Fiume perform this optimization indirectly, through an optimal metric space quantization method by Lloyd [1982]. Given any initial points configuration, this process repeats the following iteration: a Voronoi diagram is computed for every point and then the point is shifted to the centroid of its Voronoi cell. Some connections between Lloyd's method, centroidal Voronoi tessellations, and quantization error minimization are given in [Du et al. 1999].

In this process points get spread more evenly in space by equating the inter-point distances of neighboring points. This reduces the maximal inter-point distance and allows to capture higher frequencies. Moreover, Lloyd's method provides a control over the point density by specifying a target density function. For these reasons Lloyd's method became a tool-of-choice for generating high-quality point distributions and is used in many applications. Despite its popularity, Lloyd's method suffers from two main shortcomings. The first, which was already recognized by McCool and Fiume [1992], is related to its termination. As the iterations progress, the points converge to a regular deterministic anisotropic hexagonal patterns (or piecewise hexagonal in large point sets) which correspond to optima (or local optima) in the sense of spatial quantiza-

tion error. Balzer et al. [2009] demonstrate the lack a reliable termination criteria and its dependence on the number of points, initial distribution, and target density. The second drawback, as indicated by Balzer et al., is the impreciseness at which Lloyd's method approximates target spatially-varying density functions, where an apparent blur is observed.

In their work Balzer et al. propose a variant of Lloyd's method that solves these issues by imposing a capacity constraint over the Voronoi tessellation; each Voronoi cell must contain an equal amount of integrated density. This approach achieves an improved blue noise characteristics, precise adaptation to density functions, and shows no regularity artifacts. These high-quality results are obtained at considerable computational costs; each iteration runs in $O(n^2 + nm \log \frac{m}{n})$ time where m is the number of discrete spatial coordinates needed for computing the Voronoi cells (m is typically few hundreds times n). Li et al. [2009] accelerate this method by reducing its complexity to $O(nm)$ and using various parallelization and low-level optimizations.

A recent method by Schmaltz et al. [2010] generates halftoning using an interacting particle model inspired by electrostatics. This involves global Coulomb interactions between every two points in the system and hence the running time of this algorithm is $O(n^2)$. This approach was shown to achieve high-quality blue-noise properties as well as an closely match the target density. In contrast, our formulation is based on localized kernels which result in a system with a few and *local* interactions that allows updating the points in linear time. We show however that locally-coupled systems, such as ours, suffer from a slowing down that requires $O(n)$ stochastic updates in order to produce a valid point set. We avoid this slowing down using a novel multi-scale sampling scheme and describe an algorithm whose overall running time is linear.

Error diffusion methods [Floyd and Steinberg 1976] are very efficient low-level half-toning algorithms designed also for hardware implementation in printers. Similarly to our approach, these methods approximate the target density using filters that model the presence of laid points. However, unlike our method and the rest of the methods mentioned here, these algorithms place the points at discrete integer grid locations and are typically deterministic. Hence they are not ideal for many of the applications mentioned above.

In the last category, we mention a more recent trend to accelerate the construction of point distributions by tiling precomputed patterns. This was initially proposed by Shade et al. [2000] who construct these building blocks such that they obeys the minimal-distance requirement with the points of all possible neighboring tiles. This ideas was later refined in several respects and this approach can even produce spatially adapted distributions, see [Ostromoukhov et al. 2004] and [Kopf et al. 2006]. For a more elaborate survey and evaluation of these methods, we refer the readers to the paper by Lagae et al. [2008]. In our context, many of these methods precompute the tile patches using one of the basic point generation methods mentioned above. Our work focuses on the preliminary stage of computing initial point generations.

3 New Approach

We model the target density ρ, whether it is constant or not, as a sum of nonnegative radially-symmetric kernels that represent the points. Thus, the presence of each point is represented by the decaying profile of the kernel surrounding it. In the context of signal sampling, these profiles represent the regions covered by the point samples (when assuming local correlation), and in the context of density function approximation, the kernels represent the spread of matter around the points (e.g., black ink in stippling).

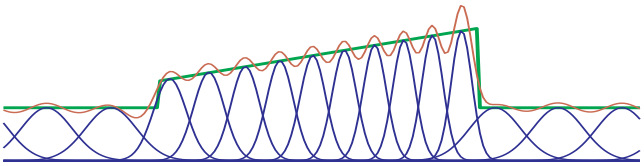

Figure 2: *1D example showing the kernels (blue), the resulting approximation function (red), and the target density function (green).*

We postulate that each point is equally important and assign an equal amount of matter, i.e., integrated density, to every kernel. We define the j-th kernel centered around the point $\mathbf{x}_j$ by

$$\frac{1}{\sigma_j^d}\Phi\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma_j}\right), \tag{1}$$

where σ_j defines the scale, and $\mathbf{x}, \mathbf{x}_j$ are points in $\Omega \subset \mathbb{R}^d$ where Ω is the domain of interest. We denote by $\|\cdot\|$ the Euclidian norm in this d dimensional real space. The function Φ is positive, finitely-integrable, and monotonically decaying away from zero. The normalization factors σ_j^{-d} ensure that the kernels' integrals are fixed and independent of their scale. This is easily verified by a variable transformation $\mathbf{y} = \sigma_j \mathbf{x}$ in $\mathbb{R}^d$ when integrating (1).

In d-dimensional space the point density, defined as the number of points per unit volume, is inversely proportional to the d-th power of the inter-point distance. Therefore, to get an average point density proportional to $\rho(\mathbf{x})$, we define the scale function $\sigma(\mathbf{x}) = \sqrt[-d]{\rho(\mathbf{x})}$, and set the j-th kernel scale to be $\sigma_j = \sigma(\mathbf{x}_j)$. Under these definitions, the kernels defined in (1) become $\rho(\mathbf{x}_j)\Phi(\|\mathbf{x}-\mathbf{x}_j\|/\sigma(\mathbf{x}_j))$, i.e., functions with an amplitude proportional to the density function at their centers $\mathbf{x}_j$. This property makes the kernels appropriate building blocks for the approximation of $\rho(x)$ and, in fact, a similar type of kernels is used for scattered-data interpolation in the method of Shepard [1968].

Given any assignment $\{\mathbf{x}_j\}_{j=1}^n$ of n points, we can define a corresponding *approximation function* $A(\mathbf{x})$ that approximates $\rho(\mathbf{x})$ by summing the kernels together, i.e.,

$$A(\mathbf{x}) = \sum_{j=1}^{n} \frac{1}{\sigma(\mathbf{x}_j)^d}\Phi\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma(\mathbf{x}_j)}\right). \tag{2}$$

We define the error of this approximation by

$$E\big(\{\mathbf{x}_j\}_{j=1}^n\big) = \int_\Omega |A(\mathbf{x}) - \rho(\mathbf{x})|^\gamma \, d\mathbf{x}, \tag{3}$$

where, for reasons we explain below, we use $\gamma = 1$ in practice and use $\gamma = 2$ for derivation and theoretical analysis at the Supplemental Text. This construction is illustrated in Figure 2.

Once the kernels scale is determined (ultimately by ρ) the only remaining degrees of freedom in this model are the kernels centers, i.e., the point locations $\mathbf{x}_j$ themselves. Minimizing E, with respect to the kernel centers, in case of a constant ρ in two-dimensional space, achieves a configuration in which the kernels are arranged in an hexagonal pattern. This optimum is equivalent to the one obtained by converged Lloyd's iterations and is undesirable for its periodicity. Nonetheless, this connection implies that, besides accurately matching the target density function, low error values achieve equidistant point arrangement and therefore, similarly to Lloyd's, our method has the ability to achieve spectral enhancement.

Unlike the standard practice of Lloyd's method where a limited number of deterministic minimization iterations are performed over

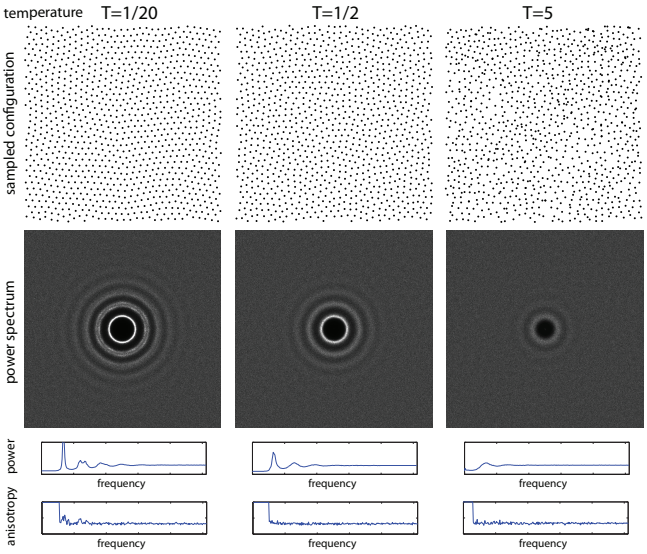

Figure 3: *Comparison between samples obtained from different temperature values. As the temperature decreases the more ordered the points become.*

an initial point set, constructed by an unrelated random step, we unify error minimization and randomness by defining a statistical mechanics particle model using E. This is done by interpreting the error E as an *energy function* and assigning each configuration $\{\mathbf{x}_j\}_{j=1}^n$ a probability density according to the following Boltzmann-Gibbs distribution

$$P\left(\{\mathbf{x}_j\}_{j=1}^n\right) \propto e^{-E\left(\{\mathbf{x}_j\}_{j=1}^n\right)/T}, \tag{4}$$

where T is a temperature parameter that defines the disorder or randomness in the system. Low temperatures concentrate the measure around the low-energy point configurations and at the limit of zero temperature, P converges to a delta function around the lowest energy configuration, known as the ground state—the hexagonal pattern mentioned above in case of two-dimensional constant target density. Increasing the temperature 'flattens' the distribution P and allows states that are less energetically-favorable to realize and thus randomness increases. As $T \to \infty$, all configurations are made equally probable, ρ ceases to affect, and a (uniform) Poisson distribution is obtained. This effect of the temperature parameter is demonstrated in Figure 3 and 13. The optimum of a similar kernels-based model is used for uniformly sampling surfaces in [Öztireli et al. 2010].

The choice L_1 norm ($\gamma = 1$ in (3)) ensures the following scale invariance: if ρ is mapped to $c\rho$ with $c > 0$, then $\sigma \mapsto \sigma / \sqrt[d]{c}$ and hence the kernels normalization factors $\sigma(\mathbf{x}_j)^{-d}$ are also multiplied by c, meaning that the pointwise error in 3 is multiplied by c. On the other hand, the kernels scale divides by $\sqrt[d]{c}$ along each axis. Thus, if we consider the same number of kernels, the integrated volume divides by c and therefore E and hence P remain unchanged. This property guarantees that the same relative disorder will be produced for every density value ρ, e.g., if we stretch a point set produced by a constant ρ, we obtain a set whose scale *and* disorder correspond to a different values of target ρ.

In practice we use a rectangular domain Ω and approximate the integral in (3) by discretizing it over a grid of nodes we call *computational grid*, defined by $\mathcal{G} = \{\Delta x, 2\Delta x, .., m\Delta x\}^d \subset \Omega$, where Δx is the grid spacing and m is the number of grid points along each axis. To simplify notations we assume an equal spacing and number of grid points along axis. Note that the kernel centers, i.e., the points $\mathbf{x}_j$ we are interested in, have continuous coordinates and are not limited by this discretization. The grid spacing is determined by the minimal kernel scale $\sigma_{\min}$, and set such that $\Delta x < 2/3\sigma_{\min}$ in order to properly capture the Gaussian kernels we use. Without loss of generality, we can assume that $0 < \rho \leq 1$, which means that $\sigma_{\min} \geq \sqrt[-d]{\rho} \geq 1$ allowing us to use $\Delta x = 1$ with kernels defined by $\Phi(x) = e^{-(2x/3)^2}$. The total number of points can be controlled, indirectly, through m, the resolution at which we represent ρ. As we noted earlier, the number of points in a finite domain is $n \propto \sigma^{-1/d}$ meaning that $m \propto n^{1/d}$ and therefore the total number of grid points we use, m^d, is $O(n)$. Since we need to discretize the entire computational domain our method grows exponentially as function of the dimension d. Note that in many practical scenarios the expected number of points n also grows at this rate and hence our algorithm is not necessarily asymptotically inefficient. In the next sections we describe the difficulties in drawing samples from P and derive a new algorithm which we summarize in Section 3.3.

3.1 Drawing Samples

In order to fully define the model given by (4), the number of kernels or points n must be specified. We will explain how this is done later, in Section 3.3, and for now assume it is known. Given this probability model, generating stochastic point patterns amounts to *sampling* it, i.e., generating point configurations $\{\mathbf{x}_j\}_{j=1}^n$ at probability $P(\{\mathbf{x}_j\}_{j=1}^n)$. Sampling high-dimensional probability distributions is a non-trivial task that lies behind the Markov-chain Monte Carlo (MCMC) method for approximating high-dimensional integrals [Robert and Casella 2005].

The idea behind this method is to construct a random Markov chain (MC) of configurations by applying simple, typically low-dimensional, random change to the configuration in each step. These small transitions in the configuration space are designed such that, given any initial configuration, later configurations will distribute according to P. The density P in this case is known as the *equilibrium* distribution of the chain. The following four conditions are sufficient for producing a chain with P as its equilibrium distribution: (i) the chain is a Markov process; the distribution of the next configuration given the current is independent of any past configuration, (ii) the chain is irreducible; it can switch between every two configurations within a finite number of steps, (iii) the chain is aperiodic; it can return to a configuration at irregular times, and (iv) the chain obeys the *detailed balance* condition with respect to P; the chance that the chain moves from configuration $\mathbf{X}$ to $\mathbf{Y}$ times $P(\mathbf{X})$ must be equal to the chance for the reverse transition times $P(\mathbf{Y})$. In practice, a proposal distribution that obeys conditions (i)-(iii) is first constructed. Then, its transitions are accepted or rejected based on a Metropolis-Hastings (MH) test that 'corrects' the chain such that condition (iv) is met. More detail on the theory of MC and the related sampling methods can be found in [Robert and Casella 2005; Landau and Binder 2005].

One possible MC construction is the *random walk* that generates the proposal configuration by adding a random offset vector to the current configuration. In our case, the manifold of configurations that are likely according to P is rather 'slender' as the points must be kept almost uniformly apart. Large random offsets violate this relation and lead to high rejection rates, whereas small offsets require long running times. The *Gibbs* sampler is another common choice in which the conditional distribution along every coordinate is sampled sequentially. However, the conditional distributions of (4) do not correspond to simple analytical models that we can directly draw samples from. Instead, we use the *Langevin* method to approximate these conditional distributions and use a Metropolis-Hastings correction [Besag 1994; Robert and Casella 2005] to sample them *sequentially*.

Algorithm 1: Single-level MH-corrected Langevin relaxation.

for *every point* $j = 1...n$ **do**
  compute $\mathbf{x}_j^{k+1}$ according to Eq. (5);
  if *MH test in Eq.* (6) *fails* **then**
    set $\mathbf{x}_j^{k+1} = \mathbf{x}_j^k$;
  end
end

We denote the point configuration at the k-th step of the chain by $\mathbf{X}^k = \{\mathbf{x}_j^k\}_{j=1}^n$, and use the Langevin method [Grenander and Miller 1994] to produce a new configuration by altering *only* a single point, the j-th point, of the current state by

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k - \frac{\Delta t_j}{2T}\nabla_{\mathbf{x}_j} E(\mathbf{X}^k) + \sqrt{\Delta t_j}\boldsymbol{\xi}_j^k, \tag{5}$$

and set $\mathbf{x}_{j'}^{k+1} = \mathbf{x}_{j'}^k$ for $j' \neq j$, where $\boldsymbol{\xi}_j^k \in \mathbb{R}^d$ are vectors of random variables drawn independently from the standard Normal distribution in $\mathbb{R}^d$, and $\nabla_{\mathbf{x}_j}$ denotes the partial derivative vector with respect to the coordinates of the j-th point. We set the step size $\Delta t_j = T\sigma(\mathbf{x}_j)\Delta t$ so that the same expected change in probability will occur regardless of the kernel scale and temperature. The overall step size factor is set by Δt. At every step k, a different point is chosen sequentially, i.e., $j = k \bmod n$.

Grenander and Miller [1994] use this update rule to produce the final chain, based on the fact that as $\Delta t \to 0$ (and hence $\Delta t_j \to 0$), this process converges to the *Langevin equation*. This is a continuous-time stochastic differential equation that describes a Brownian motion that distributes according to $P(\mathbf{X})$, see [Binney et al. 1986]. In computer simulation, only a finite time step size $\Delta t > 0$ can be used and, in fact, the larger Δt is, the farther we move and the faster we explore the configuration space and reach equilibrium. However, any finite Δt introduces some errors that cause the chain to deviate from P. Accurate sampling of P can still be obtained by treating the configuration $\mathbf{X}^{k+1}$ defined above as a *proposal* configuration and using an MH test [Besag 1994; Robert and Casella 2005] to determine the next configuration of the chain. In this test $\mathbf{X}^{k+1}$ is accepted with probability

$$\frac{P(\mathbf{X}^{k+1})}{P(\mathbf{X}^k)} \frac{e^{-\left\| \frac{\sqrt{\Delta t_j}}{2T}\left(\nabla_{\mathbf{x}_j} E(\mathbf{X}^k) + \nabla_{\mathbf{x}_j} E(\mathbf{X}^{k+1})\right) + \boldsymbol{\xi}_j^k \right\|^2/2}}{e^{-\|\boldsymbol{\xi}_j^k\|^2/2}}, \tag{6}$$

and otherwise rejected, by setting $\mathbf{X}^{k+1} = \mathbf{X}^k$. This step ensures that the chain obeys the detailed balance condition with respect to P, see [Robert and Casella 2005]. This test is implemented by drawing a uniform variable $u \sim U[0,1]$ and comparing it to (6).

This original MH-corrected Langevin sampler, described in [Besag 1994; Robert and Casella 2005], updates *all* the points $\mathbf{x}_j^k$, where $j = 1..n$, according to (5) in each time step before performing the MH test. In this case, as the number of points n grows, the magnitude of the random vector $\{\boldsymbol{\xi}_j^k\}_{j=1}^n$ added to the current state $\mathbf{X}^k$ in (5) grows as well. Roberts and Rosenthal [1998] show that for a large class of models time steps of order $\Delta t = O(n^{-1/3})$ must be used in order to maintain a constant acceptance rate. In contrast, the MH-corrected Langevin method we described operates sequentially, by updating a single point $\mathbf{x}_j$ before every MH test. Thus in our approach, much like in the Gibbs sampler it approximates, the magnitude of the stochastic fluctuation $\boldsymbol{\xi}_j$ is independent of n. This strategy does not involve performing much more calculations than the traditional MH-corrected Langevin method, yet since Δt is independent of n larger and thus fewer chain steps are needed for reaching the equilibrium. Unfortunately, as we show in the next section, the sampling complexity of this scheme is still unsatisfactory. However, we use this stochastic relaxation as a component in our final sampling scheme and therefore summarize it in Algorithm 1.

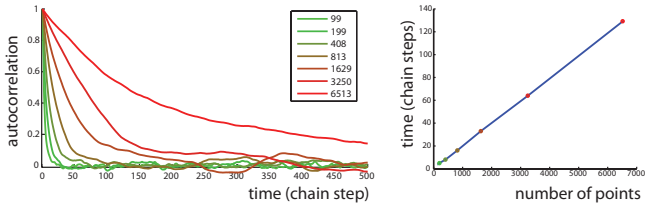

Figure 4: *Left plots show the autocorrelation of the lowest non-trivial Fourier mode (real part), computed from models containing different number of points n (matching a constant density in 2D space). Right plot shows how the number of chain steps k, required to achieve the same correlation value ($R^k = 0.5$), grows linearly with n. These tests were performed with $\gamma = 2$. Qualitatively equivalent behavior is observed for $\gamma = 1$.*

We conclude with two important implementation remarks. We speedup computations by ignoring the change in $\sigma(\mathbf{x}_j)$ computing the gradient term $\nabla_{\mathbf{x}_j} E$ in (5) and (6) and approximate it by

$$\nabla_{\mathbf{x}_j} E \approx \frac{\gamma}{\sigma(\mathbf{x}_j)^d} \int \frac{\mathbf{x}_j - \mathbf{x}}{\|\mathbf{x} - \mathbf{x}_j\|} \Phi'\left(\frac{\|\mathbf{x} - \mathbf{x}_j\|}{\sigma(\mathbf{x}_j)}\right) \big(A(\mathbf{x}) - \rho(\mathbf{x})\big)^{\gamma - 1} d\mathbf{x}, \tag{7}$$

where for $\gamma = 1$ we define the power $(\cdot)^0 = \text{sign}(\cdot)$. This change in the proposal configuration does not effect the equilibrium distribution P nor does it introduce any noticeable decrease in the acceptance rate. Secondly, when we evaluate the integral (7) above we do not need to access all the nodes of the computational grid. The only relevant nodes are at distance $\propto \sigma(\mathbf{x}_j)$ from $\mathbf{x}_j$, where Φ is above some epsilon.

Critical Slowing Down. The distribution of the initial configuration $\mathbf{X}^0$ is different from P and, in fact, most MCMC methods start with a predetermined (deterministic) initial configuration. Therefore, the main concern when developing an MCMC sampling method is how fast the distribution of the k-th step of the chain $\mathcal{P}(\mathbf{X}^k)$ converges to P. This rate is typically estimated empirically from the correlation of various measures at different time steps. We estimate the following time autocorrelation function

$$R^k(\boldsymbol{\omega}) = \frac{\text{cov}\big(|\hat{A}^0(\boldsymbol{\omega})|, |\hat{A}^k(\boldsymbol{\omega})|\big)}{\sqrt{\text{var}\big(|\hat{A}^0(\boldsymbol{\omega})|\big)\ \text{var}\big(|\hat{A}^k(\boldsymbol{\omega})|\big)}}, \tag{8}$$

where A^k is the approximation function that corresponds to the k-th configuration, $\mathbf{X}^k$, and $\hat{A}^k(\boldsymbol{\omega})$ is its Fourier transform at frequency coordinates $\boldsymbol{\omega}$.

The *correlation time* measures the number of chain steps separating two statistically uncorrelated states. It is typically defined as the number of steps beyond which the autocorrelation is small, i.e.,

$$\tau(\boldsymbol{\omega}) = \text{argmin}_k\{R^k(\boldsymbol{\omega}) < \varepsilon\}, \tag{9}$$

where ε is a small number below which we consider the configurations to be uncorrelated. In practice, this correlation time is used to estimates the *burn-in time*; the number of chain steps that must be computed before the chain loses its correlation with the initial configuration $\mathbf{X}^0$ and reaches its equilibrium distribution P. Therefore, in the MCMC method the first $k < k_0 = \max_{\boldsymbol{\omega}} \tau(\boldsymbol{\omega})$

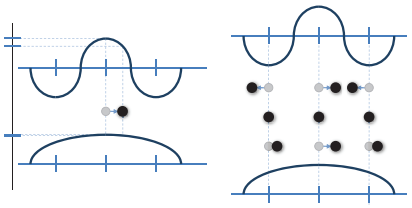

Figure 5: *(left) Show the effect a displacement of a single point has on low and high frequency modes. The high-frequency mode shows a greater change in response than the low-frequency mode. (right) The effect of low- and high-frequency displacements, of an equal magnitude, have on the inter-particle distance. Positive values correspond to right offsets and negative to left ones. The high-frequency offset has a greater effect over the inter-point distances and violates more the local point density.*

steps are usually discarded, and in our application, where we need only a single configuration of points, we stop at k_0 and pick $\mathbf{X}^{k_0}$ as the stochastic point set we output. The definitions and measurements we used here are standard tools, commonly used to study and estimate MCMC convergence, see [Binney et al. 1986; Goodman and Sokal 1989] for more information.

Figure 4 shows plots of R^k that correspond to the lowest non-trivial frequency $\boldsymbol{\omega}$, computed from configurations containing different number of points n. The graphs show that the correlation time τ, and hence the number of MC steps needed, grow linearly with n, or equivalently, $\tau \propto m^2$ in this two-dimensional test. This long burn-in time implies that the current solution is suboptimal and runs in $O(n^2)$ time (since each relaxation sweep, in this test, updates *all* the n points). Indeed, this phenomenon plagues many lattice and off-lattice statistical physics models and it is known as *critical slowing down* [Goodman and Sokal 1989].

In the Supplemental Text we formally study the convergence rate of the Langevin sampling scheme, defined in (5), applied to our probability model P. This analysis shows that we expect relaxation time $\tau \propto m^2$ in every space dimension d and confirms the empirical finding we saw here. We show that every iteration has an effect of $O(\|\boldsymbol{\omega}\|^2)$ over $\hat{A}(\boldsymbol{\omega})$, i.e., the low frequencies are weakly affected by the stochastic update and that this slowing factor results from two reasons. Small movements in point locations have lower effect on low-frequency modes, of order $O(\|\boldsymbol{\omega}\|)$, since these functions change slowly in space (e.g., it has no effect on $\hat{A}(\mathbf{0})$). The second $O(\|\boldsymbol{\omega}\|)$ results from the fact that smooth low-frequency changes in the point locations do not change the inter-point distances by much and therefore the local point density and the energy that measures it, in (3), are weakly effected by these modes. Thus, the energy gradient, $\nabla_{\mathbf{x}_j} E$ in (5), which points to the direction of maximal change, barely contains these low-frequency modes. These effects are illustrated in Figure 5. The critical slowing down is observed since as the number of points n grows, lower frequency components are introduced to the system, $\min \|\omega\| = O(2\pi/m)$, and hence more chain steps are needed to relax and decorrelate these modes. This cannot be solved by increasing the time step Δt, in (5), since this will also increase the change along the high-frequency components which will, in turn, lead to large energy fluctuations and hence a higher rejection rate.

3.2 Multi-Scale Sampling

The critical slowing down, we observed here, appears in a board range of statistical mechanics particle models. There are a few related works that address it by applying *global* modifications, that target low-frequency modes, when constructing proposal states. In the context of molecular dynamics of fluids, Dress and Krauth [1995] construct the proposal configuration by rotating all the points about an arbitrary pivot point and replace particles between the rotated and the original configurations. Liu et al. [2004] use the pivot to reflect points around it until an acceptable state is found. Both these methods are designed to operate on stationary spatially-invariant models and cannot be used in our application. Goodman and Sokal [1989] perform stochastic relaxation at multiple scales using the Multigrid machinery, originally designed for solving elliptic linear systems, to treat the slowing down in lattice models. Applying this strategy to non-structured models, like our kernels model, will be costly; it will require us to perform repeated linearizations, particle connectivity searches, and construct the multiple matrices needed in the Multigrid method. Based on our understanding of the slowing down, we derive a very simple and efficient coarse-to-fine sampling procedure that obtains stochastic relaxation at progressively shorter wavelengths of $\hat{A}(\boldsymbol{\omega})$.

The idea is to adapt the Multigrid concepts [Trottenberg et al. 2001] into our context by deriving a hierarchy of probability models that contain fewer points which are discretized on coarser computational grids. These models are constructed such that the low frequencies of $\hat{A}(\boldsymbol{\omega})$ distribute in these models the same way they did in the original model P. Once there are only few points in the coarsest model, the Langevin relaxations are able to generate an *independent* sample with all its frequencies in equilibrium. Then, we refine this point configuration to a finer level in the hierarchy, by adding new points *while* preserving its low-frequency response. In the finer level, the new high frequencies are relaxed efficiently using the Langevin sampler. This process continues recursively and higher and higher frequency bands reach their equilibrium distribution, and terminates once the finest level is reached. In this section we describe the two components needed for this sampling strategy: the coarser models and the configuration refinement scheme.

Model Coarsening. The derivation of a coarser model is done in Fourier space for $\gamma = 2$, where we can express the energy in terms of $\hat{A}$ and $\hat{\rho}$ using Parseval identity,

$$E = \int_{\Omega} \big(A(\mathbf{x}) - \rho(\mathbf{x})\big)^2 d\mathbf{x} = \frac{1}{2\pi} \int_{[-\pi,\pi]^d} \big|\hat{A}(\boldsymbol{\omega}) - \hat{\rho}(\boldsymbol{\omega})\big|^2 d\boldsymbol{\omega}. \qquad (10)$$

The integration in Fourier space can be divided into the lower $\mathcal{L} = [-\pi/2, \pi/2]^d$ and the remaining higher $\mathcal{H} = ([-\pi, -\pi/2] \cup [\pi/2, \pi])^d$ frequency bands. The energy E can then be expressed as the sum of $E(\hat{A}|_{\mathcal{L}}, \hat{\rho}|_{\mathcal{L}})$ and $E(\hat{A}|_{\mathcal{H}}, \hat{\rho}|_{\mathcal{H}})$, where $\hat{A}|_{\mathcal{L}}$ and $\hat{\rho}|_{\mathcal{L}}$ are $\hat{A}$ and $\hat{\rho}$ restricted to $\mathcal{L}$ respectively, and set to be zero outside these sets. $\hat{A}|_{\mathcal{H}}$ and $\hat{\rho}|_{\mathcal{H}}$ are the analog restriction to $\mathcal{H}$. This decomposition of the energy leads to the following factorization of P

$$P(\hat{A}, \hat{\rho}) = e^{-E(\hat{A},\hat{\rho})/T} = e^{-E(\hat{A}|_{\mathcal{L}},\hat{\rho}|_{\mathcal{L}})/T} e^{-E(\hat{A}|_{\mathcal{H}},\hat{\rho}|_{\mathcal{H}})/T}, \qquad (11)$$

i.e., $\hat{A}|_{\mathcal{L}}$ and $\hat{A}|_{\mathcal{H}}$ are statistically independent sets of coordinates.

The key point is that according to the sampling theorem the energy term $E(\hat{A}|_{\mathcal{L}}, \hat{\rho}|_{\mathcal{L}})$, that contains the highly-correlated low-frequencies of $\hat{A}|_{\mathcal{L}}$, can be expressed at half the spatial resolution; over $m/2$ grid points along each axis. In other words, the low-frequency factor of the model gives rise to a *coarse model*, defined at a coarser spatial resolution by subsampling $\rho|_{\mathcal{L}}(\mathbf{x})$, the ideally low-passed ρ. We denote this density by $\rho_{\mathcal{C}}(\mathbf{y})$ where $\mathbf{y} \in [0, m/2]^d$ and define, similarly to (4), the following coarse grid model

$$P_{\mathcal{C}}\big(\{\mathbf{y}_j\}_{j=1}^{n_{\mathcal{C}}}\big) = e^{-2^d E(A_{\mathcal{C}},\rho_{\mathcal{C}})/T}, \qquad (12)$$

where $A_{\mathcal{C}}$ is the approximation function defined by (2) using the kernels centered at $\{\mathbf{y}_j\}_{j=1}^{n_{\mathcal{C}}} \subset [0, m/2]^d$. As a matter of convention, we assume the coarse grid has the same spacing as the fine

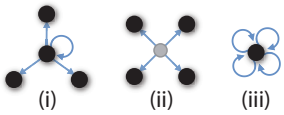

Figure 6: *Three refinement rules where a point belonging to a coarse configuration divides into four points (black circles) in two-dimensional space. In all cases the center of mass of the new points falls at the original point.*

grid ($\Delta x = 1$) and therefore the kernels have roughly the same dimensions as in the finer model (since $\rho_{\mathcal{C}}$ has roughly the same values as ρ). As we shall see later, this model is expected to contain fewer points, roughly quarter in $d = 2$. The 2^d in (12) compensates for having fewer nodes at the coarse grid that the fine (2^{-d} times less). By its derivation, this model is consistent with P and assigns the same (relative) probabilities for $\boldsymbol{\omega} \in \mathcal{L}$. The fact that $\{\mathbf{y}_j\}_{j=1}^{n_{\mathcal{C}}}$ span, through $\hat{A}_{\mathcal{C}}$, the equivalent potion of $\mathcal{L}$ that $\{\mathbf{x}_j\}_{j=1}^{n}$ span in $\mathcal{L} \cup \mathcal{H}$ (through $\hat{A}$) is explained by a similarity argument; as we shall see, the number of points in the coarse grid $n_{\mathcal{C}}$ compared to n is roughly the same proportion as the number of computational grid nodes in $\rho_{\mathcal{C}}$ compared to ρ, i.e., the two models have the same ratio between their degrees of freedom and target objectives. Finally, given $P_{\mathcal{C}}$, even coarser models are obtained by computing coarser and coarser target density functions.

Refinement Scheme. It remains to show how a sample $\{\mathbf{y}_j\}_{j=1}^{n_{\mathcal{C}}}$ drawn from a coarser model, $P_{\mathcal{C}}$, is mapped to a valid sample $\{\mathbf{x}_j\}_{j=1}^{n}$ at the finer configuration space while keeping its low frequencies sufficiently unchanged. In case of functions defined on grids, such as $A_{\mathcal{C}}$, this can be achieved via standard interpolation. However, the values of $A_{\mathcal{C}}$ are not free parameters but determined via $\{\mathbf{y}_j\}_{j=1}^{n_{\mathcal{C}}}$. Implementing this by doubling the kernels coordinates and scale will lead to an inconsistency with the kernel scales dictated by the target density at the finer grid. This is because the target density functions in the two grids have approximately the same values (e.g., exactly the same in case of constant ρ). The difference must therefore be in the number of kernels; every point $\mathbf{y}_j$ must be translated to 2^d points in the finer model. In the Supplemental Text we show that a spectrally-correct refinement is achieved by mapping every $\mathbf{y}_j$ as follows

$$\mathbf{y}_j \mapsto 2\mathbf{y}_j + \sigma(\mathbf{y}_j)\mathbf{z}^l, \tag{13}$$

where $\mathbf{z}^l$ for $1 \le l \le 2^d$ are small offset vectors in $\mathbb{R}^d$ such that $\sum_l \mathbf{z}^l = \mathbf{0}$, i.e., the center of mass of the new points falls at $2\mathbf{y}_j$. Figure 6 shows several refinement rules in two-dimensional space that preserve the center of mass. A Fourier analysis, which we detail at the Supplemental Text, shows that the accuracy of the approximation function resulting from this refinement is

$$\hat{A}(\boldsymbol{\omega}) = 2^d \hat{A}_{\mathcal{C}}(2\boldsymbol{\omega}) + O\big(\|\boldsymbol{\omega}\|^2\big). \tag{14}$$

This applies for the low frequencies $\boldsymbol{\omega}$ we are most concerned about and for kernels that admit the admissibility condition $\hat{\Phi}'(0) = 0$, which is obeyed by any symmetric function that is differentiable at zero, including the Gaussian kernels we use. Experiments show that setting $\|\mathbf{z}^l\| = 1$ in the first two options shown in Figure 6 requires the smallest number of Langevin iterations (we use 15) to relax at every scale. The third option requires considerable more iterations (about 50) in order to smooth the high-frequency interpolation aliasing it produces. To produce the results shown in the paper, we used the the second (square) refinement rule in Figure 6.

The refinement error in (14) is proportional to the Langevin relaxation correction size $\|\boldsymbol{\omega}\|^2$ we saw. Hence, a fixed number

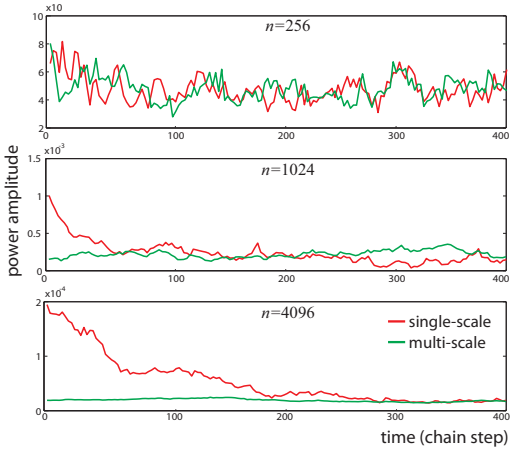

Figure 7: *The lowest non-trivial Fourier mode (real part) of the approximation function is shown at every chain step. Red plots show relaxations at the finest scale of configurations created at that scale (using our initialization step) and green plots show the relaxation at the same scale of configurations that were relaxed at coarser scales and refined to the finest scale.*

of stochastic iterations are needed in order to reach equilibrium across the entire spectrum—implying the critical slowing down is eliminated. Indeed, this is the main principle behind the Multigrid MCMC [Goodman and Sokal 1989] as well as its deterministic case [Trottenberg et al. 2001]. For this reason it is not needed to compute $\rho|_{\mathcal{C}}$ as an ideal low-passed version of ρ and any standard compact filtering that transfers the low-frequency content to the coarse grid with an error proportional to $\|\boldsymbol{\omega}\|^2$ is sufficient. The high-frequencies introduced at the finer scale are not assumed to be close to equilibrium but are relaxed efficiently by the Langevin iterations at that grid.

Validation. Given these two ingredients we can now test the following multi-scale sampling scheme. Given an initial set of points at the coarsest grid we repeat the following two steps: we perform several Langevin iterations using (5) and then refine the resulting configuration to the next (finer) grid by mapping every point to 2^d points according to (13). This process repeats itself until the finest grid is reached. In order to evaluate the effectiveness of this scheme, we compare it to applying the Langevin iterations only at the finest scale. Figure 7 shows the evolution of the lowest non-trivial Fourier coefficient of $\hat{A}$ at the finest grid computed by both strategies. It is clear from the plots that as the number of points n grows, it takes the single-scale approach more iterations in order to reach equilibrium values while in the multi-scale strategy the low-frequency component is *already at equilibrium* when the iterations of the finest level start. This independence of the number of points n indicates that the critical slowing down is eliminated using the multi-scale sampling. This empirical behavior is shared by both models $\gamma = 1, 2$.

3.3 Proposed Algorithm

Initial Configuration. In order to fully define the model given by (3) and (4), the number of points n must be determined. This is done by scanning the computational grid in a random order and testing whether a kernel should be place at every node. The point is accepted if it leads to a reduction in the energy (3) and discarded otherwise. Note that, as we discussed in Section 3, the change in energy can be computed by considering only the nodes where Φ is non-negligible. Moreover, a vast majority of the nodes can be efficiently tested and rejected if the *pointwise* error does not decrease when adding the (center) kernel value at that node. Thus when accepting a kernel, nodes in a neighborhood of size proportional to the

Algorithm 2: Multi-scale sampling scheme.

compute ρ_q, for $q = 1...p$ by smoothing and sampling $2^{dq}\rho$;
for *every node* $\mathbf{y}$ *(scanned randomly)* **do**
  set $q = \lceil(-\log_2 \rho_p(\mathbf{y}))/d\rceil$;
  compute kernel with $\sigma = \sqrt[-d]{\rho_q(\mathbf{y})/w}$ and $w = 2^{d(p-q)}$;
  if *kernel with* σ *and* w *reduces energy* **then**
    add point $\mathbf{y}$;
  end
end
relax points $\mathbf{Y}$ using Algorithm 1 ;
for $q = p - 1...1$ **do**
  for *every point* $\mathbf{y}$ **do**
    if $w > 1$ **then**
      create points $\mathbf{x}^l = 2\mathbf{y} + \sqrt[-d]{\rho_{q-1}(\mathbf{y})/w}\,\mathbf{z}^l$;
      with $w^l = w/2^d$, where $l = 1...2^d$;
    else
      create point $\mathbf{x} = 2\mathbf{y}$ with $w = 1$;
    end
    set new kernels scale $\sigma = \sqrt[-d]{\rho_q(\mathbf{x})/w}$;
  end
  swap $\mathbf{Y} = \mathbf{X}$;
  relax points $\mathbf{Y}$ with $\sigma < \sigma^{\max}$ using Algorithm 1 ;
end

kernel scale around its center are eliminated from the list of candidate point locations. While we use this initialization at the coarsest level (containing a fixed number of grid nodes), the running time of this initialization procedure is linear even if computed at the finest grid. The construction of a random permutation over the nodes indices is also done in linear time; the list of indices is scanned once and every index is swaped with another index chosen randomly and uniformly from the list. Finally, while the kernels centers are initialized at integer grid coordinates, this ceases to be the case after the first Langevin iteration took place.

While this initialization produces a sufficient coverage of the domain and can be used as the starting point for the coarse-to-fine scheme we described, we would like to introduce one modification that will make the entire algorithm more efficient. The factor 2^d multiplying the energy in the coarse model (12) can be inserted inside the energy term. In the case of $\gamma = 1$, the integrand will become $|2^d A_{\mathcal{C}} - 2^d \rho_{\mathcal{C}}|$. This can be interpreted as if the coarser density $\rho_{\mathcal{C}}$ is 2^d times denser than ρ, and similarly, $2^d A_{\mathcal{C}}$ can be viewed as if every kernel is assigned with 2^d times more mass. Now, assume ρ is smaller than 2^{-d} in some region, then $2^d\rho_{\mathcal{C}} < 1$ in the corresponding region of the coarse grid. Recall that, as we discussed in Section 3, densities below one define kernel scales that are properly captured by the computational grid resolution. The scale resulting from this density is $\sigma = \sqrt[-d]{2^d\rho_{\mathcal{C}}} = \sqrt[-d]{\rho_{\mathcal{C}}}/2$, i.e., half the scale produced by $\rho_{\mathcal{C}}$. Thus, when constructing the initial coarse-grid point configuration we can place 2^d kernels of half the size, each weighing a unit, instead of placing one large kernel that weighes 2^d. Clearly, these kernels must not be split in the refinement step and therefore we must keep track of the kernels weights. Note that, when computing the initial configuration, we do not need to explicitly create the 2^d kernels. By placing smaller kernels of less mass, more such kernels will be created automatically.

We shall now describe our modified scheme. Suppose we are about to operate on p grid levels. In our implementation we use $m = 32$ grid points at the coarsest level and set $p = \log_2 m - 5$. We denote by ρ_q the target density function reduced to the q-th scale (via filtering and subsampling) *including* the multiplication by the factor $2^{d(q-1)}$. When proposing to place a new kernel at some node $\mathbf{x}$ in the coarsest grid, we compute the maximal $q \le p$ such that $2^{dq}\rho_p(\mathbf{x}) \le 1$, which is given by $q = \lceil(-\log_2 \rho_p)/d\rceil$. We set the weight w of this point to $w = 2^{d(p-q)}$ and its scale to $\sigma = \sqrt[-d]{\rho_p/w}$. The number q specifies the number of refinement steps in which this point must not be split as it is represented by a kernel small enough.

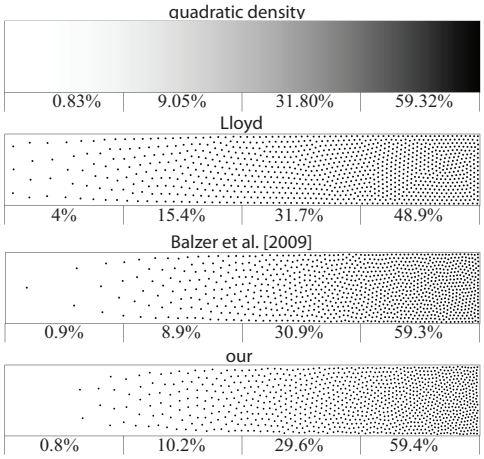

Figure 8: *A quadratic density profile approximated using 1000 points using Lloyd's method, Balzer et al. [2009], and our method.*

Configuration Refinement. When refining a point configuration, every point with weight w greater than one must be split into 2^d points that weigh $w/2^d$. The kernels scales are defined, in every level q, by $\sigma = \sqrt[-d]{\rho_q/w}$. When a weight reaches one, the point ceases to split when transferred to a finer grid. In this case, the kernel scale start growing exponentially (due to the factor $2^{d(q-1)}$ in ρ^q). Therefore, after a few non-splitting refinement steps, when the kernel gets large enough $\sigma \ge \sigma^{\max}$, its contribution to $\hat{A}$ is confined to the low-frequency band which, by now, reached equilibrium. Hence, we can *stop* relaxing it using the Langevin update and thus avoid processing large kernels. This properly makes our scheme much more efficient and in practice when updating a point we do not access more than 16-by-16 grids nodes.

This makes our method well-suited for handling target densities of a high dynamic range where very large kernels are rendered on grid resolution that is fine enough to capture the smaller kernels. Besides the computational cost, also the memory requirements can be reduced using an adaptive computational grid where the nodes resolution depends on the target density. We discuss this option as future work. Despite these detailed considerations, our scheme is based on very simple steps that we summarize in Algorithm 2.

4 Results

We implemented our algorithm in C++ and ran it on a Pentium 2.83GHz CPU. Our single-core implementation produces more than 15,000 spatially-varying points per second. We use 15 Langevin iterations at each scale (each sweep scans all the points at that scale). The coarsest grid resolution we use is 32-by-32 nodes. Unless specified otherwise, the results are computed with the temperature set to $T = 1/2$.

In Figure 11 we compare the results we obtain for a constant target density with the results of Lloyd's method and the state-of-the-art method of Balzer et al. [2009]. This test shows that our result computed with $T = 1/2$ meets the blue-noise qualities of Balzer et al.,

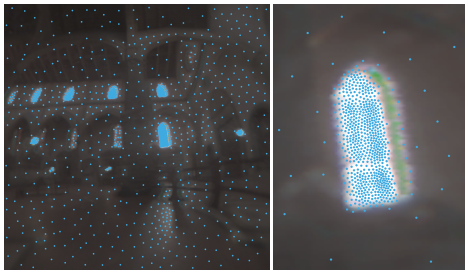

Figure 9: *Galileo's Tomb, Santa Croce, Florence; (left) high-dynamic range luminance map used as target density for generating an importance sampling map. (right) A blowup of the main window. Blue dots indicate samples. Map contains 1784 points.*

both in terms of isotropy and annulus dimension. Another important test is shown in Figure 8 where a quadratic density function is set as the target density. Our method optimizes explicitly, through the definition of the energy (3), the approximation of this density, and therefore achieves close numbers. Further comparisons with the methods of Secord [2002] and Balzer et al. are shown in Figure 12, where spatially-varying densities are matched. A comparison with the method of Schmaltz et al. [2010] is shown at Figure 13 where we also demonstrate our method at different temperatures. In Figure 9 we place samples according to high-dynamic range target ρ with density ratio of $1:5000$. These 1784 points were generated in $2/3$ of a second. This is about fives times slower than stippling (spatially-varying) images with intensity ratio of $1:200$ due to the fine computational grid resolution needed. Finally in Figure 10 we verify the linear running-time of our method and compare it to the running time of Balzer et al., which is three orders of magnitude higher on point sets of moderate size. We also compare to the running time of Schmaltz et al. on a GPU that crosses our linear performance also at moderate sizes, lower then 10^3 points.

5 Conclusions

We presented a new formalism for generating multi-dimensional blue-noise stochastic point distributions. In this approach randomness is unified with an equidistance requirement by defining a statistical interacting particle model. Point sets are then produced by drawing samples from this model. We described an efficient multi-scale sampling procedure that generates proper samples from this model in running times that are linear in the number of points generated. To achieve this, we analyzed the critical slowing down occurring in our model and used this understanding to develop a very simple and effective multi-scale coarse-to-fine sampling scheme that eliminates this deficiency.

While the number of points n produced can be controlled by the computational grid resolution m, it cannot be set explicitly. The proportionate relation between the two numbers allows approximating n in one or two extra trial-and-error iterations. As we noted earlier, an adaptive computational grid in the form of quadtree or naive (spatially-varying) random nodes arrangement can be used for reducing memory costs when dealing with high-dynamic range target densities. We leave these investigations as well as the option of a parallelizing our algorithm for future work.

We believe that defining random models explicitly offers a general framework that allows specifying detailed requirements. While sampling such models may require some thought, very efficient solutions can be found. Hopefully, this paradigm be found useful for other computer graphics applications.

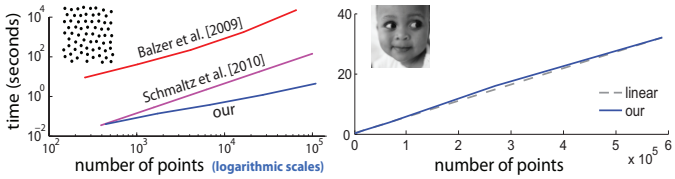

Figure 10: *Running time dependency on the number of points. The slopes of the left plots show the quadratic dependency of the method's of Balzer et al. [2009] and Schmaltz et al. [2010] compared to the linear performance of our algorithm. On the right we verify the linear running time of our algorithm on spatially-dependent target density.*

Acknowledgements

I would like to thank Yehuda Arav, Guy Kindler and Raz Kupferman for their useful feedback. I would like to expressed my deepest gratitude to Alexandre J. Chorin who taught me most that I know on Markov-chain sampling methods. Finally, I would like to thank the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities as well as Microsoft New Faculty Fellowship Program for supporting this research.

References

BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: a variant of lloyd's method. In *ACM SIGGRAPH 2009 papers*, ACM, New York, NY, USA, 86:1–86:8.

BESAG, J. 1994. Discussion: Markov chains for exploring posterior distributions. *Annals of Statistics 2*, 4, 1734–1741.

BINNEY, J., DOWRICK, N., FISHER, A., AND NEWMAN, M. 1986. *The Theory of Critical Phenomena*. Clarendon Press, Oxford.

BOWERS, J., WANG, R., WEI, L.-Y., AND MALETZ, D. 2010. Parallel poisson disk sampling with spectrum analysis on surfaces. In *ACM SIGGRAPH Asia 2010 papers*, ACM, New York, NY, USA, 166:1–166:10.

COHEN, M. F., SHADE, J., HILLER, S., AND DEUSSEN, O. 2003. Wang tiles for image and texture generation. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, 287–294.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. In *ACM SIGGRAPH 1984 papers*, ACM, New York, NY, USA, 137–145.

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

DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., MĚCH, R., PHARR, M., AND PRUSINKIEWICZ, P. 1998. Realistic modeling and rendering of plant ecosystems. In *ACM SIGGRAPH 1998 papers*, ACM, New York, NY, USA, 275–286.

DEUSSEN, O., HILLER, S., VAN OVERVELD, C., AND STROTHOTTE, T. 2000. Floating points: A method for computing stipple drawings. *Computer Graphics Forum 19*, 40–51.

DIPPÉ, M. A. Z., AND WOLD, E. H. 1985. Antialiasing through stochastic sampling. In *ACM SIGGRAPH 1985 Papers*, ACM, New York, NY, USA, 69–78.

DRESS, C., AND KRAUTH, W. 1995. Cluster algorithm for hard spheres and related systems. *Journal of Physics A: Mathematical and General 28*, 23.

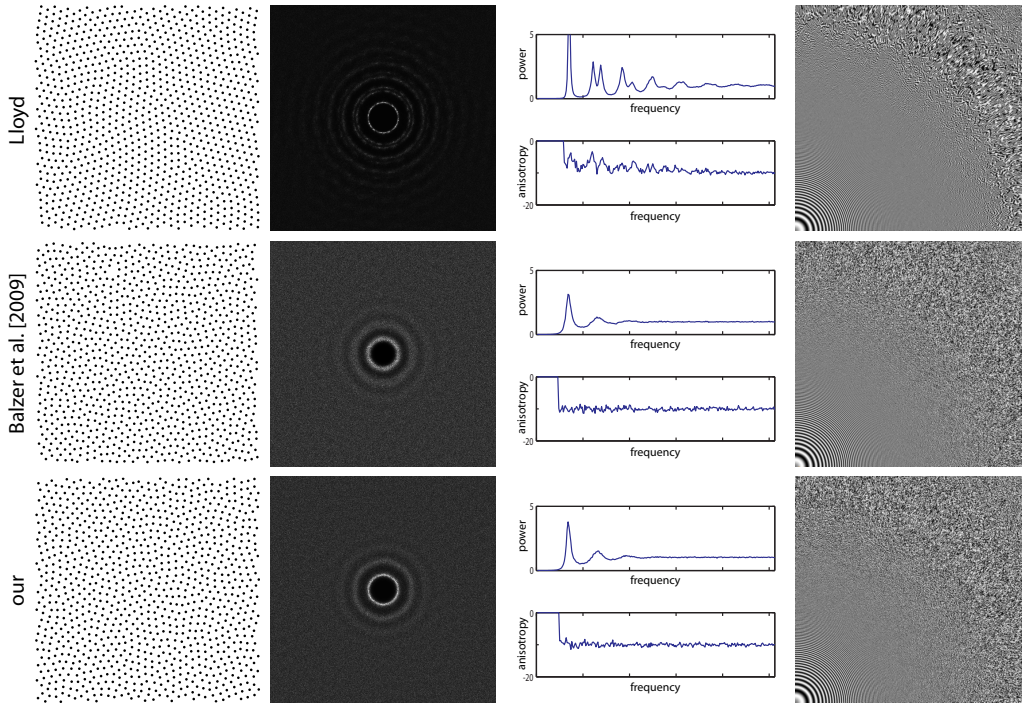

Figure 11: *A comparison between Lloyd, Balzer et al. [2009], and our methods. Left to right are: uniform sample pattern of 1024 points, average power spectrum, the radial average and variances, and the zone plate test function sampled by* 512^2 *points. Statistics gathered from 10 samples.*

Figure 12: *A comparison between Secord [2002], Balzer et al. [2009], and our method on non-uniform target density (20,000 points used).*

DU, Q., FABER, V., AND GUNZBURGER, M. 1999. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review 41*, 4, 637–676.

DUNBAR, D., AND HUMPHREYS, G. 2006. A spatial data structure for fast poisson-disk sample generation. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH '06, 503–508.

FLOYD, R. W., AND STEINBERG, L. 1976. An Adaptive Algorithm for Spatial Greyscale. *Proceedings of the Society for Information Display 17*, 2, 75–77.

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

GOODMAN, J., AND SOKAL, A. D. 1989. Multigrid monte carlo method. conceptual foundations. *Phys. Rev. D 40*, 6 (Sep), 2035–2071.

GRENANDER, U., AND MILLER, M. I. 1994. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B (Methodological) 56*, 4, 549–603.

HILLER, S., DEUSSEN, O., AND KELLER, A. 2001. Tiled blue noise samples. In *Proc. of Vision Modeling and Visualization*, Aka GmbH, 265–272.

JONES, T. R. 2006. Efficient generation of poisson-disk sampling patterns. *Graphics, Fpu, and Game Tools 11*, 2, 27–36.

KOLLIG, T., AND KELLER, A. 2003. Efficient illumination by high dynamic range images. In *Proc. of Eurographics workshop on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, 45–50.

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive wang tiles for real-time blue noise. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 509–518.

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

LANDAU, D., AND BINDER, K. 2005. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, New York, NY, USA.

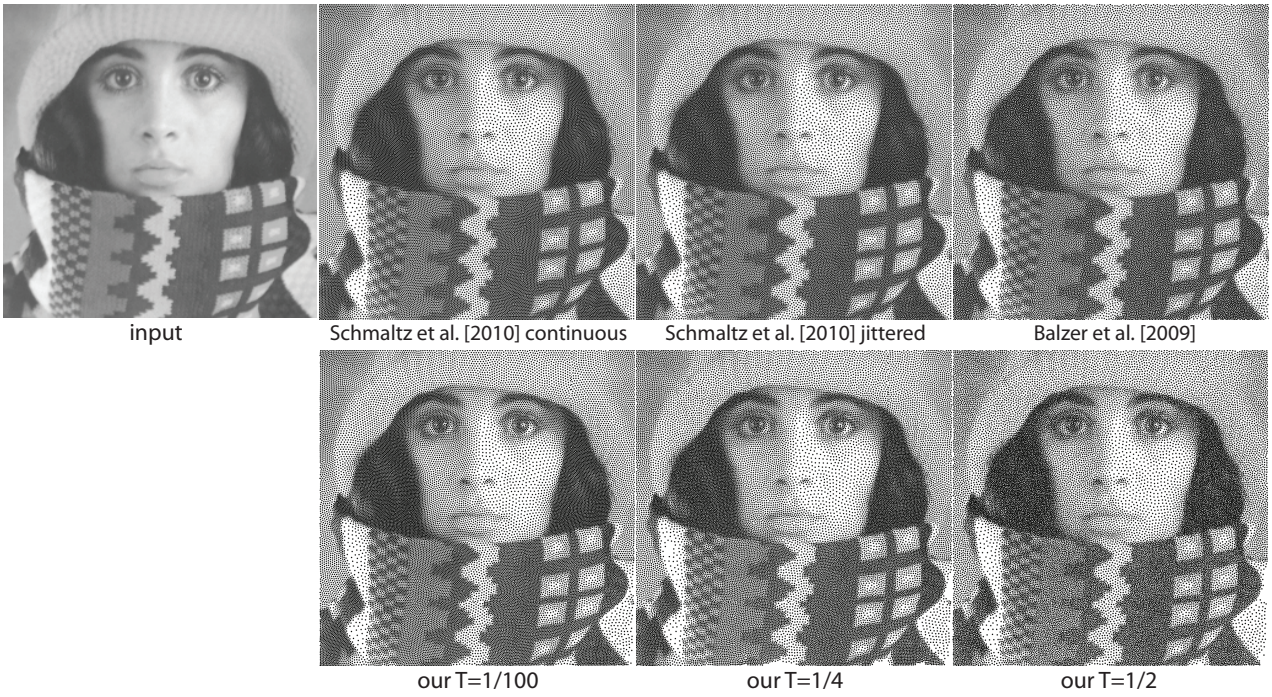

Figure 13: *A comparison between Schmaltz [2010], Balzer et al. [2009], and our method at three temperature values (30,000 points used).*

LI, H., NEHAB, D., WEI, L.-Y., SANDER, P., AND FU, C.-W. 2009. Fast capacity constrained voronoi tessellation. Microsoft Research, no. MSR-TR-2009-174.

LI, H., WEI, L.-Y., SANDER, P. V., AND FU, C.-W. 2010. Anisotropic blue noise sampling. In *ACM SIGGRAPH Asia 2010 papers*, ACM, New York, NY, USA, 167:1–167:12.

LIU, J., AND LUIJTEN, E. 2004. Rejection-free geometric cluster algorithm for complex fluids. *Phys. Rev. Lett. 92*, 3 (Jan), 035504.

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

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

MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. In *ACM SIGGRAPH 1987 Papers*, ACM, New York, NY, USA, 65–72.

MITCHELL, D. P. 1991. Spectrally optimal sampling for distribution ray tracing. 157–164.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast hierarchical importance sampling with blue noise properties. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, 488–495.

ÖZTIRELI, A. C., ALEXA, M., AND GROSS, M. 2010. Spectral sampling of manifolds. In *ACM SIGGRAPH Asia 2010 papers*, ACM, New York, NY, USA, SIGGRAPH ASIA '10, 168:1–168:8.

ROBERT, C. P., AND CASELLA, G. 2005. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

ROBERTS, G. O., AND ROSENTHAL, J. S. 1998. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 60*, 1, 255–268.

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

SECORD, A. 2002. Weighted voronoi stippling. In *Proc. int. symposium on Non-photorealistic animation and rendering*, ACM, New York, NY, USA, 37–43.

SHADE, J., COHEN, M. F., AND MITCHELL, D. P. 2000. Tiling layered depth images. 231–242.

SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. of ACM national conference*, ACM, New York, NY, USA, 517–524.

SURAZHSKY, V., ALLIEZ, P., AND GOTSMAN, C. 2003. Isotropic remeshing of surfaces: a local parameterization approach. In *Proc. of 12th Int. Meshing Roundtable*, 215–224.

TROTTENBERG, U., OOSTERLEE, C. W., AND SCHULLER, A. 2001. *Multigrid*. Accademic Press, London, UK.

WEI, L.-Y. 2008. Parallel poisson disk sampling. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, 20:1–20:9.

WEI, L.-Y. 2010. Multi-class blue noise sampling. In *ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, 79:1–79:8.

WHITE, K. B., CLINE, D., AND EGBERT, P. K. 2007. Poisson disk point sets by hierarchical dart throwing. *Symposium on Interactive Ray Tracing 0*, 129–132.

WHITTED, T. 1979. An improved illumination model for shaded display. ACM, New York, NY, USA, ACM SIGGRAPH 1979 papers, 14–.

YELLOT, J. I. 1983. Spectral consequences of photoreceptor sampling in the rhesus retina. 382–385.