

Blue Noise Sampling using an SPH-based Method

Min Jiang¹ Yahan Zhou² Rui Wang² Richard Southern¹ Jian Jun Zhang¹

¹Bournemouth University ²University of Massachusetts Amherst

Density Diff.	2	4	8	25	100	500
Iterations	373	344	300	288	217	228
Time (s)	1.108	0.936	0.842	0.780	0.593	0.640
Samples						
Fourier power spectrum						
Radial means						
Anisotropy						

Figure 1: This table shows samples generated using our algorithm, by varying the density difference parameter—the difference between the actual density and the rest density in SPH. From left to right the spectrum transitions from low effective Nyquist Frequency and oscillation to high effective Nyquist Frequency and oscillation. Our sampling method generates various blue noise profiles at 10000 points per second.

Abstract

We propose a novel algorithm for blue noise sampling inspired by the Smoothed Particle Hydrodynamics (SPH) method. SPH is a well-known method in fluid simulation – it computes particle distributions to minimize the internal pressure variance. We found that this results in sample points (i.e., particles) with a high quality blue-noise spectrum. Inspired by this, we tailor the SPH method for blue noise sampling. Our method achieves fast sampling in general dimensions for both surfaces and volumes. By varying a single parameter our method can generate a variety of blue noise samples with different distribution properties, ranging from Lloyd’s relaxation to Capacity Constrained Voronoi Tessellations (CCVT). Our method is fast and supports adaptive sampling and multi-class sampling. We have also performed experimental studies of the SPH kernel and its influence on the distribution properties of samples. We demonstrate with examples that our method can generate a variety of controllable blue noise sample patterns, suitable for applications such as image stippling and re-meshing.

CR Categories: I.4.1 [Computer Graphics]: Digitization and Image Capture—Sampling;

Keywords: blue noise sampling, smoothed particle hydrodynamics, surface sampling, Fourier spectrum.

1 Introduction

Blue noise sampling is a well-known technique useful in many graphics applications, such as image synthesis, physically-based simulation, non-photorealistic rendering, and geometry processing. It generates sample points that are stochastic yet evenly dispersed in the spatial domain. Fourier spectrum analysis shows that the spectral energy of such sample points is largely absent in the low-frequency region, while evenly spread in the high-frequency region. This ‘blue noise’ property turns out to greatly benefit anti-aliasing, and the sample patterns are also visually pleasing, leading to its popularity in the aforementioned applications.

Beginning with the first dart throwing algorithm [Cook 1986; Mitchell 1987], various methods have been proposed to generate samples with blue noise property, such as Lloyd relaxation [Lloyd 1982], Poisson disk sampling [McCool and Fiume 1992], half-toning [Ulrichney 1987], Capacity Constrained Voronoi Tessellations (CCVT, [Balzer et al. 2009]), the kernel-density model [Fattal 2011] and tiling [Wachtel et al. 2014]. Others have proposed methods to analyse and characterize the distribution properties of samples in various domains [Dippé and Wold 1985; Wei and Wang 2011] and their anti-aliasing properties [Heck et al. 2013; Subr and Kautz 2013].

One limitation of classic blue noise methods is that they are difficult to steer or even vary the distribution properties of the generated samples. For example, samples generated using dart throwing have a characteristic blue noise profile, but it is unclear how to modify or adapt the method to generate samples with a different profile. The ability to control the spectral profile is important since different applications may perform better with alternative sample patterns.

Recently, several techniques have been proposed to generate samples with user-specific spectral profiles [Zhou et al. 2012; Heck et al. 2013; Wachtel et al. 2014]. These techniques are flexible and can generate samples with a range of different distribution proper-

ACM Reference Format

Jiang, M., Zhou, Y., Wang, R., Southern, R., Zhang, J. 2015. Blue Noise Sampling using an SPH-based Method. *ACM Trans. Graph.* 34, 6, Article 211 (November 2015), 11 pages.

DOI = 10.1145/2816795.2818102 <http://doi.acm.org/10.1145/2816795.2818102>.

Copyright Notice

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright held by the Owner/Author.
SIGGRAPH Asia ’15 Technical Paper, November 02 – 05, 2015, Kobe, Japan.

ACM 978-1-4503-3931-5/15/11.

DOI: <http://doi.acm.org/10.1145/2816795.2818102>

ties. However, they are typically slow to compute and difficult to extend to general sampling domains such as surfaces and volumes.

In this paper we propose a new sampling method using a well-known fluid simulation method – Smoothed Particle Hydrodynamics (SPH). It computes particle distributions to minimize the internal pressure variance. Experiments show that this results in sample points with a high-quality blue noise spectrum. Our contributions are as follows:

1. We present a novel blue noise sampling method inspired by SPH. Our method is fast, easy to parallelize, and can produce samples with a variety of different blue noise profiles, ranging from Lloyd’s relaxation to CCVT samples.
2. We extend the method to support general domain sampling (including surfaces and volumes), as well as adaptive sampling and multi-class sampling.
3. Using our framework enables a continuous trade-off between the Nyquist frequency and oscillation. Through experimentations we also evaluate the SPH kernel and its influence on the distribution properties of samples.

Finally, we demonstrate the use of our method applied to popular problems such as image stippling and re-meshing.

2 Related work

Blue Noise Sampling. One of the earliest blue noise sampling algorithms is the ‘dart throwing’ method, which generates random samples with minimum distance constraints [Cook 1986; Mitchell 1987]. The original algorithm is computationally expensive, and difficult to precisely control the number of samples. Since then, many alternatives have been proposed to improve the performance of dart throwing [Wei 2008; Guo et al. 2015].

Relaxation-based methods introduce additional topological information into the point pattern, making it easier to precisely control the number of samples and improve the spectrum properties. A well-known approach in this class is the Lloyd’s method [1982], also known as Voronoi relaxation, firstly introduced to computer graphics by McCool and Fiume [1992]. Despite its popularity, the produced sampling suffers from too much regularity which causes aliasing problem for intended applications.

Balzer and coworkers [2009] presented a variant of Lloyd’s method with capacity constraints (CCVT), which solved the regularity issue. However, CCVT is considerably more expensive to compute, as it needs to preserve the capacity constraints at every iteration step. Fast CCVT [Li et al. 2010] increased the speed while still maintaining the basic discrete sampling technique. More recently, Xu et al. [2011] proposed Capacity-Constrained Delaunay Triangulation (CCDT), replacing the Voronoi cells of CCVT with Delaunay triangles on the same sites to avoid the dual problem. De Goes et al. [2012] modelled CCVT as a constrained transport problem. Similarly Capacity-Constrained Surface Triangulation (CCST) [Xu et al. 2012] extended CCDT to allow for surface sampling with surface area constraints. Compared with CCVT-like methods which focus on one kind of sampling pattern, our method provides controllable sampling pattern and can provide better trade-off between effective Nyquist Frequency and oscillation with a much faster speed.

Using a kernel density model, Fattal [2011] proposed a new sampling algorithm with excellent blue noise characteristic and linear time complexity. His method introduced a ‘randomness’ parameter to avoid hexagonal patterns. However, it is not capable of varying the samples’ distribution properties. Our method, in contrast, allows users to choose the desired blue noise patterns with the

change of one parameter. Although we both use kernel functions to define the density, the density in SPH implicitly considered the volume, and the use of pressure term ensures the volume preservation. This is analogous to the capacity constraint in CCVT, and thus our method naturally avoids hexagonal patterns without using the ‘randomness’ parameter. In addition, Fattal’s method changes the number of samples dynamically to match the points density with the target density, resulting in difficulties of controlling the exact number of samples. Our method, on the other hand, allows user-specified number of samples.

Stochastic Sampling with Controllable Spectrum. A limitation of classic blue noise sampling methods is that they are difficult to steer or even vary the distribution properties of the resulting samples. The ability to change the distribution property is important as different applications may call for different sample patterns. In [Zhou et al. 2012; Öztireli and Gross 2012], gradient-based methods were presented to generate samples with general, user-specified spectral profiles. Wachtel et al. [2014] proposed a tile-based method to achieve the same capability. To improve stochastic integration, other research has proposed blue noise with controlled bias, variance [Subr and Kautz 2013] and aliasing [Heck et al. 2013]. These methods are flexible, but many of them are expensive to compute, and it’s not clear how to extend such methods to general sampling domains such as surfaces and volumes. Our method is focused on generating a variety of blue noise samples, controllable with one parameter. We analyse the trade-off between effective Nyquist frequency and oscillation in the Fourier power spectrum for our method. In addition, our method can be easily extended to surface sampling and volume sampling.

Smoothed Particle Hydrodynamics (SPH). SPH was originally introduced for the simulation of astrophysical problems [Gingold and Monaghan 1977], with its first application to free surface flows by Monaghan [1994]. In the last two decades SPH has enjoyed popularity as one of the key methods for fluid simulation [Akinci et al. 2012; Adami et al. 2013; Ihmsen et al. 2014b]. The localised nature of the simulation naturally lend themselves to parallel implementations, which greatly improves performance of intensive fluid computations [Harada et al. 2007; Hérault et al. 2010] and resulting in more accurate simulations. There are few fluid papers related with sampling idea. Adams et al. [2007] proposed adaptive SPH. De Goes et al. [2015] related SPH model to a power diagram based CCVT energy solver. Both methods applied to fluid simulation without considering the sampling properties of the point distribution.

We used the standard SPH model to demonstrate the fundamental idea, and simplify the evaluation and exposition of our method, as it is the first application of fluid simulation to this problem domain. Alternative formulations of particle-based fluid [Solenthaler and Pajarola 2009; Ihmsen et al. 2014a; Macklin and Müller 2013] allow the simulation to run with larger time steps through greater stability but at a higher cost per step. While these methods have the potential to improve computational performance, they could also lead to slow convergence as the number of points increases since they all used Jacobi-style iterative methods [Macklin and Müller 2013]. We would like to investigate the usage of these schemes as future work.

3 The Core Idea

SPH is an interpolation method for particle systems. The concept of SPH in fluid simulation is to interpolate fluid quantities at arbitrary positions and to approximate the spatial derivatives with a finite number of sample positions [Ihmsen et al. 2014b]. SPH itself is a process which forces the density to approach the rest density by

minimizing the pressure difference between points [Müller et al. 2003]. This explains why in the regions with a smooth flow field SPH particles are uniformly distributed [Adami et al. 2013] – which provides an intuition for its use in generating blue noise samples.

3.1 The Basic Algorithm

In SPH [Monaghan 2005], a scalar quantity A_i at location \mathbf{x}_i is approximated with a set of neighboring points j at \mathbf{x}_j using radially symmetrical smoothing kernels W_{ij} :

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij} \quad (1)$$

$$W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (2)$$

where m_j is the mass of point and ρ_j the density. The kernel satisfies $\int W(\mathbf{x}, h) d\mathbf{x} = 1$ and $W(\mathbf{x}, h) = 0$ when $\|\mathbf{x}\| > h$. Here h as a smooth radius defines the neighbourhood size.

Following Equation 1, the point density can be represented by:

$$\rho_i = \sum_j m_j W_{ij} \quad (3)$$

We know that the relationship between the mass and the density can also be written as:

$$\rho_i = \frac{m_i}{V_i} \quad (4)$$

where V_i is the volume. Notice that the volume of points will be the same if the density of points is the same (assuming the mass is the same for each point).

The pressure force is computed using spatial derivatives by applying SPH rules to the Navier-Stokes equation [Girault and Raviart 1986]:

$$\mathbf{F}_i^{\text{pres}} = -m_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} \quad (5)$$

where p is the pressure calculated directly from density:

$$p = k(\rho - \rho_0), \quad (6)$$

where ρ_0 is the **rest density**, k is a gas constant. The pressure force is created whenever there is a pressure difference and it acts in the direction from high pressure to low pressure along the negative pressure gradient.

Equation 5 introduces a spring-like behavior to the pressure calculation: while a density higher than the rest density produces a positive pressure that pushes the points away, a lower density will result in a negative pressure and brings the points together. As such, the pressure term aims to equalize the density differences throughout the fluid. For blue noise sampling, to avoid the attracting pressure forces we adopt the same concept with fluid simulation and clamp the negative pressure to 0 in each iteration.

The pressure forces move the points towards the density center. The density center is actually the mass center according to the definition of the density. This idea is consistent with Lloyd's algorithm which moves the points towards the center of the cell and converges to *equal area*, while SPH converges to *equal density*.

If a point is inserted at a density center where there is no density difference, the pressure force returns to 0. This implies in an equilibrium the pressure and density will be identical everywhere, thus the same volume (capacity) is achieved according to Equation 4.

This implicitly matches the capacity-constraint in CCVT. However, we don't need to guarantee the same capacity (volume) in every iteration in SPH — instead the SPH algorithm automatically reaches the same capacity (density) in convergence which guarantees the quality of SPH sampling.

When using SPH for fluid simulations a viscosity force is included to synchronize the velocity of all particles. In the sampling context the smooth dynamic flow of a simulation is irrelevant: instead we want the samples to settle with 0 velocity as quickly as possible. Since SPH is a force driven solver, without viscosity points easily overshoot. In our method, a damping coefficient δ is applied to the velocity, achieving an equivalent behavior to viscosity but with considerably better performance. The damping coefficient plays a similar role to the viscosity coefficient: it affects how quickly the particle reacts to the pressure. We've found $\delta = 0.9$ works well in our experiments.

3.2 Convergence

We consider the sampling to have converged when the point displacement $\|\mathbf{x}' - \mathbf{x}\|$ is less than ϵ , where \mathbf{x}' is the updated point position. We set $\epsilon = 0.01 d$, where d is the average distance between adjacent samples which can be approximated by $d = \sqrt[n]{V/N}$ for n -dimensional sampling.

Upon convergence, the points have the following properties:

1. All points have approximately the same density, i.e. $\forall i, j, \rho_i \approx \rho_j$. Notice that this does *not* imply point density reaches the rest density.
2. The pressure force of all points reaches zero, i.e. $\forall i, \mathbf{F}_i^{\text{pres}} = 0$.

The first property means there is no density difference among points and the volume of the points are the same (surface area in 2D). Combining this with Equations 5 and 6, under the setting of constant point mass, the second property can be written as:

$$\mathbf{F}_i^{\text{pres}} = 0 \approx -\frac{2m^2 k (\rho_i - \rho_0)}{\rho_i^2} \sum_j \nabla W_{ij} \quad (7)$$

The above equation implies that upon convergence, for any point i , one of the two following conditions must stand:

1. $\rho_i = \rho_0$;
2. $\sum_j \nabla W_{ij} = 0$.

Notice that in practice those two conditions are not strictly satisfied since in implementation the algorithm stops when point movement is small enough. There is a trade off between those two conditions, and its effect will be discussed in Section 5.1. When the first condition is satisfied for all points with a smooth kernel, we can assume that the density is approximately the same in the whole domain, or:

$$\forall \mathbf{x}, m \sum_j W(\mathbf{x} - \mathbf{x}_j, h) \approx \rho_0 \quad (8)$$

The condition of $\sum_j \nabla W_{ij} = 0$ actually asks for all the forces acting on a sample to be symmetrical, the points in this case will favor a regular distribution — hexagonal pattern. The results are very similar with the fully-converged Lloyd's relaxation, shown in Fig. 2. To avoid the regularity artifacts, we meet the condition of $\rho_i = \rho_0$, rather than $\sum_j \nabla W_{ij} = 0$.

When SPH reaches the stable state in a free-surface fluid simulation, the pressure tends towards 0. In sampling context, the user

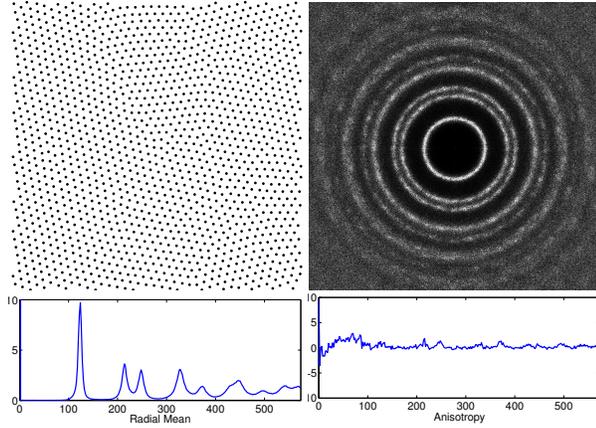


Figure 2: Using our SPH sampling method to simulate the Lloyd's relaxation profile. This regular hexagonal distribution results from the convergence condition of $\sum_j \nabla W_{ij} = 0$.

controls the number of the points, so we cannot guarantee the density equals to the initial rest density. Therefore when the distribution converges, the difference between the actual density and the rest density will tend towards a constant. Note that this constant may not be 0, in which case the points are under high pressure. If we assign the rest density close to the actual density of all points, most points will quickly converge with $\rho_i = \rho_0$ rather than $\sum_j \nabla W_{ij} = 0$. In this case the resulting point distribution will be more likely to form tetragons, heptagons or pentagons — with similar results to CCVT — as demonstrated in Fig. 3.

Discussion. The choice of kernel function affects the accuracy of the summation [Ihmsen et al. 2014b]. For the sampling example in Section 4 and Section 6, we use the standard kernels of Müller et al. [2003]. The use of alternative kernels is discussed in Section 5.2.

4 SPH Sampling and Implementation

We initialize the points as white noise within a given boundary. Each sample is modeled as an SPH particle, which carries associated properties, e.g. position, velocity, density and pressure. We run SPH over all the points, with the acceleration of the points obeying Newton's third law $\mathbf{a}_i = \mathbf{F}_i^{\text{pres}}/m_i$. Fig. 4 shows the SPH sampling within a 2D bunny shape. Note that in the sampling context, gravity is not needed.

4.1 Surface and Volume Sampling

In SPH, points are relocated to achieve the same density within the sampling region. However, due to the insufficient sampling along the object boundary the points on the boundary become highly disordered, as shown in Fig. 5(a).

4.1.1 Correction Force

The net forces push the points along the boundary in the direction of the boundary normal. We therefore need a correction force which allows the boundary points to relax onto boundary without being pushed outwards. To allow points to move freely along the boundary, we can counteract the force in the normal direction, only keeping the force along the tangential direction:

$$\mathbf{F}_i^{\text{surf}} = -\mathbf{n}(\mathbf{n} \cdot \mathbf{F}_i^{\text{pres}}) \quad (9)$$

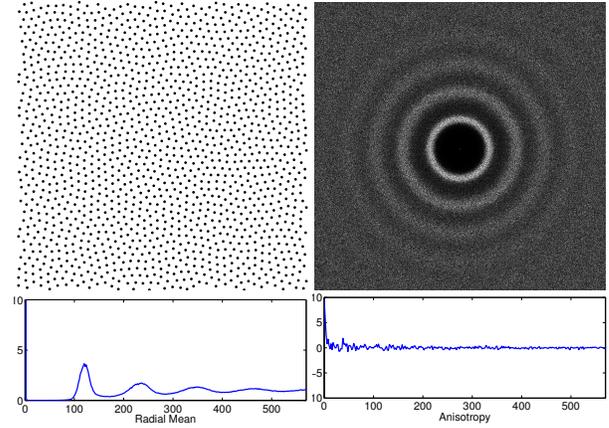


Figure 3: Using our SPH sampling method to simulate the CCVT profile. This distribution results from the convergence condition of $\rho_i = \rho_0$.

This correction force ensures that points near the boundary remain on the boundary of the object, leading to a better quality surface sampling. This principle is also applied to the velocity of points on the boundary, counteracting the perpendicular component of the velocity.

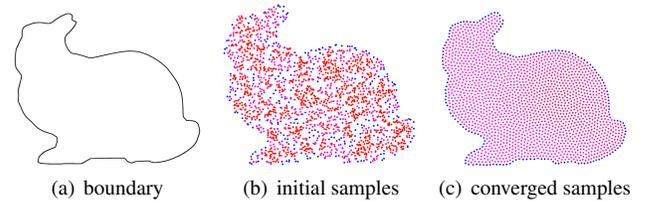


Figure 4: Sampling the interior and boundary of a 2D bunny shape using our SPH-based method. The color visualizes the density value at each point, from high (red) to low (blue).

Normal Calculation. Due to the insufficient sampling around the boundary, density changes dramatically for points on the boundary. Therefore we can calculate the normal for the surface points from the gradient of the density [Müller et al. 2003]:

$$\mathbf{n} = -\sum_j \frac{m_j}{\rho_j} \nabla W_{ij} \quad (10)$$

The resulting normal points from the high density to the low density, out the volume.

4.1.2 Cohesion Force

As we do not adjust density of boundary points, the boundary will attract more points to compensate the density loss, leading to a more dense sampling on the surface (shown in Fig. 5(b)). With only the correction force, once points are on the surface they cannot leave — potentially causing discontinuities between surface points and interior points. For this reason we introduce an additional force which adjusts the density difference between surface points and interior points.

We use the surface tension model of Akinci et al. [2013] which contains cohesion and surface area minimization terms. As our

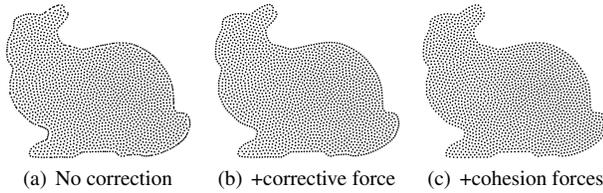


Figure 5: In (a) the boundary points are irregularly spaced after sampling. In (b), the points are more relaxed after adding the correction force, but the sampling rate is much denser than the interior. In (c) both correction and cohesion forces are applied, and the resulting boundary samples are coherent with the interior samples.

sampling model defines the surface area beforehand (unlike free surface fluid), the surface area minimization term does not affect our results. We only apply the cohesion force:

$$\mathbf{F}_i^{\text{cohe}} = -\gamma m_i m_j K_{ij} W_{\text{cohe}}(|\mathbf{x}_i - \mathbf{x}_j|) \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|} \quad (11)$$

where γ is surface tension coefficient, W_{cohe} is the spline function defined by Akinci et al. [2013], and K_{ij} is a symmetric correction factor defined as $K_{ij} = \frac{2\rho_0}{\rho_i + \rho_j}$.

This cohesion force is applied to both interior and surface points. For interior points the cohesion force is almost the same everywhere as it converges. Surface points with smaller density have larger K_{ij} due to the lack of neighboring samples, which amplifies the cohesion forces of boundary points. The effect of this force is demonstrated in Fig. 5(c). Note also that the corrected surface sampling refines the points near the boundary which have boundary points as neighbors.

Discussion. Our method for surface and volume sampling ensures that the surface points are exactly on the surface and also ensures the consistency between surface and volume sampling. Most boundary correction methods in SPH do not guarantee the particles to converge exactly on a fixed boundary, e.g., [Akinci et al. 2012; Akinci et al. 2013], therefore we introduce the correction force. Schechter et al. [2012] sampled both the interior and the surface, using an initial surface sampling as seeds to sample the volume. Our method solves both surface and volume sampling at the same time, reducing algorithmic complexity and computation overhead. Unlike the density of the ghost particles in Schechter and Bridson [2012], the density of the surface samples in our method does not need extra treatment, since they only receive forces tangential to the surface. The full method of volume and surface sampling is provided in Algorithm 1.

4.2 Surface Sampling

Our method is also suitable for surface sampling without the interior points. We achieve that by only using the correction force and replace the distance metric with geodesic distance.

Samples are initialized on the surface. We then run SPH for all surface samples, calculating \mathbf{F}^{surf} for each sample. Instead of restricting the points within the object, we constrain the points to only move on the surface. Therefore within every time step of the SPH process we need to map the samples back onto the surface. To calculate the geodesic distance, we apply the light-weight algorithm of Bowers et al. [2010], which approximates the geodesic distance using normal information. The normal is calculated according to the distance field. The full algorithm is shown in Algorithm 2.

Input: Initial Random Sample Set P_0 , Boundary Level set L , Sampling number N , count $i = 0$

Output: Blue noise samples P

```

while not converged do
  for  $i = 0$  to  $N$  do
    calculate  $\rho$ ;
  end
  for  $i = 0$  to  $N$  do
     $\mathbf{x}, \mathbf{v} \leftarrow$  the position and velocity of current point
    if  $\mathbf{x}$  is on the surface then
       $\mathbf{n} \leftarrow$  calculate normal
       $\mathbf{a} \leftarrow \mathbf{F}^{\text{pres}} + \mathbf{F}^{\text{cohe}} + \mathbf{F}^{\text{surf}}$ 
    else
       $\mathbf{a} \leftarrow \mathbf{F}^{\text{pres}} + \mathbf{F}^{\text{cohe}}$ 
    end
    Integrate  $\mathbf{a}$  to get velocity  $\mathbf{v}'$  and position  $\mathbf{x}'$ 
    Apply velocity damping  $\mathbf{v}' \leftarrow \delta \mathbf{v}'$ 
    if  $\|\mathbf{x}' - \mathbf{x}\| > \epsilon$  then
      if  $\mathbf{x}'$  is within  $L$  then
         $\mathbf{x} \leftarrow \mathbf{x}'$ 
         $\mathbf{v} \leftarrow \mathbf{v}'$ 
      else
         $\mathbf{v} \leftarrow \mathbf{v}' - \mathbf{n}(\mathbf{n} \cdot \mathbf{v}')$ 
      end
    end
  end
end

```

Algorithm 1: The SPH sampling algorithm for interior and boundary points.

As we discussed in Section 5.1, our algorithm can also generate surface samples with a variety of blue noise properties, including distributions similar to CCVT. In Fig. 6 we compare our surface sampling results with Poisson disk sampling of [Bowers et al. 2010] on a sphere. Our samples exhibit uniform distributions similar to the CCVT profile. In Fig. 7 we demonstrate a surface sampling on a bunny model and compare the Differential Domain Function (DDF) [Wei and Wang 2011] with that of 2D CCVT [Balzer et al. 2009]. The results look qualitatively similar – the difference is mainly due to the geodesic distance approximation when evaluating surface DDF.

Another interesting property of our algorithm is that samples implicitly distribute along ridges and valleys on the mesh. This is because in high curvature regions, points will be automatically pushed outwards along the normal, attracting samples to regions of high curvature. This effect is actually beneficial for many sampling

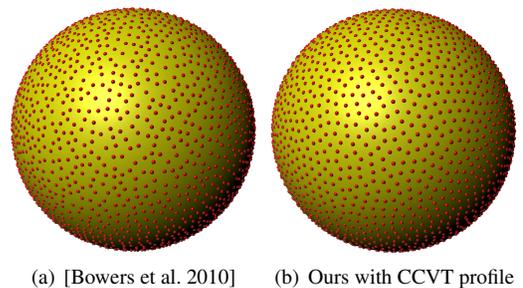


Figure 6: Comparison of Poisson disk surface sampling ([Bowers et al. 2010]) and our method.

Input: Initial Random Sample Set P_0 , Boundary Level set L ,
Sampling number N , count $i = 0$

Output: Blue noise samples P

```

while not converged do
  for  $i = 0$  to  $N$  do
    calculate  $\rho$ ;
  end
  for  $i = 0$  to  $N$  do
     $\mathbf{x}, \mathbf{v} \leftarrow$  the position and velocity of current point
     $\mathbf{n} \leftarrow$  calculate normal
     $\mathbf{a} \leftarrow \mathbf{F}^{\text{pres}} + \mathbf{F}^{\text{surf}}$ 
    Integrate  $\mathbf{a}$  to get position  $\mathbf{x}'$  and velocity  $\mathbf{v}'$ 
    Apply velocity damping  $\mathbf{v}' \leftarrow \delta \mathbf{v}'$ 
    Map  $\mathbf{x}'$  to the surface position  $\mathbf{x}'_{\text{surf}}$ 
    if  $\|\mathbf{x}'_{\text{surf}} - \mathbf{x}\| > \epsilon$  then
       $\mathbf{x} \leftarrow \mathbf{x}'_{\text{surf}}$ 
       $\mathbf{v} \leftarrow \mathbf{v}'$ 
    end
  end
end
end

```

Algorithm 2: The SPH sampling algorithm for surfaces.

applications such as rendering and remeshing, capturing high frequency features on the mesh. Its application will be discussed in Section 6.3.

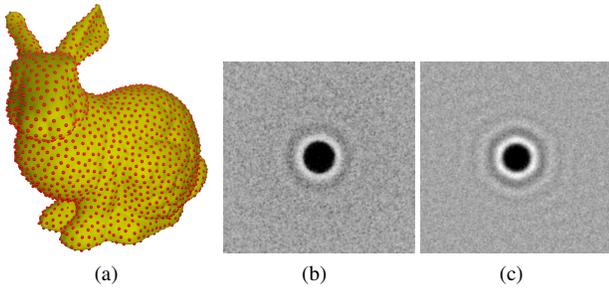


Figure 7: Surface sampling of a bunny model. Fig. 7(a) visualizes the samples. Fig. 7(b) shows the Differential Domain Function (DDF) of the bunny surface samples (averaged over 10 runs) and Fig. 7(c) shows the DDF of the CCVT method [Balzer et al. 2009], both calculated using the method of Wei and Wang [2011].

4.3 Adaptive Sampling

Adaptive sampling is achieved by applying a distance field scale $s(\mathbf{x})$ to the sample properties. As in previous work [Wei and Wang 2011; Zhou et al. 2012], the distance field in 2D can be calculated from the intensity image $I(\mathbf{x})$ as $s(\mathbf{x}) = 1/\sqrt{I(\mathbf{x})}$, or in 3D it can be defined as the inverse of the size function. The unwrapped distance between two points now becomes:

$$\tilde{s}(\mathbf{x}_i, \mathbf{x}_j) = \frac{2(\mathbf{x}_i - \mathbf{x}_j)}{s(\mathbf{x}_i) + s(\mathbf{x}_j)} \quad (12)$$

Which is then used in the kernel functions in Equation 2:

$$W_{ij} = W(\tilde{s}(\mathbf{x}_i, \mathbf{x}_j), h) \quad (13)$$

As we increase the scale, the kernels will cover larger regions. In equilibrium, to keep a similar density ρ as defined in Equation 3, the total number of points (and their distribution pattern) in the

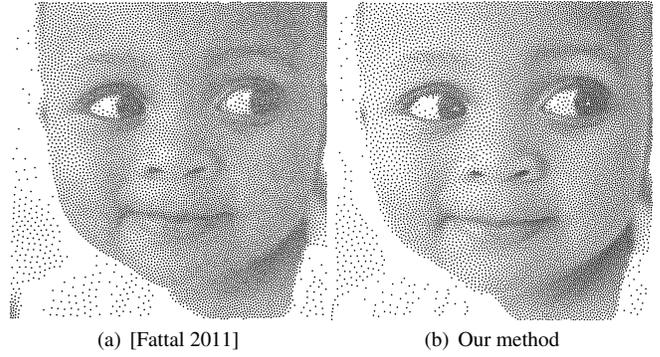


Figure 8: In (a) and (b) we show the result of Fattal [2011] and our method respectively.

large kernel region shall remain similar to those with smaller scale. As a result, points will become more sparse in high scale regions and denser in low scale regions. The update of the sample position therefore needs to correspond to the distance field scale as well:

$$(\mathbf{x} - \mathbf{x}') \leftarrow \tilde{s}(\mathbf{x})(\mathbf{x} - \mathbf{x}'). \quad (14)$$

Notice that when computing the convergence criterion, one still needs to use the unscaled distance instead of the new one.

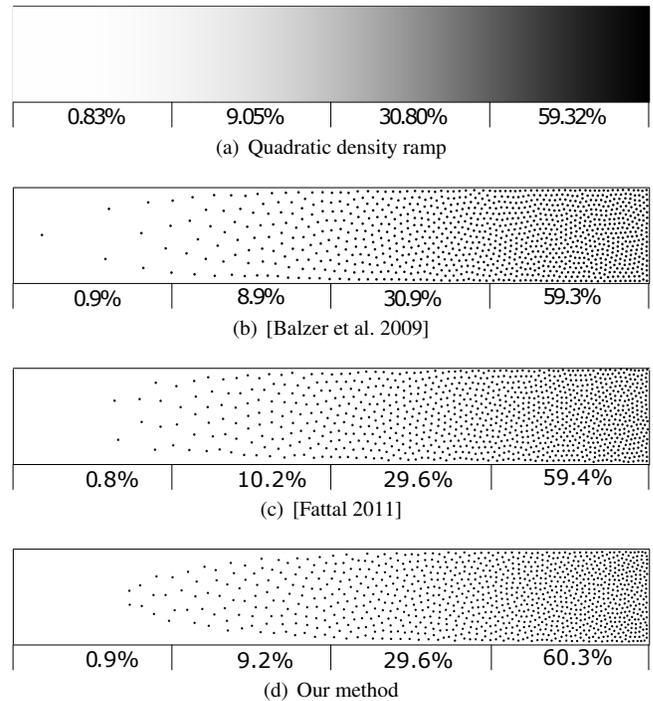


Figure 9: Comparison of adaptive sampling of our method vs. [Balzer et al. 2009] and [Fattal 2011]. Here the spatial density function is defined using the quadratic ramp as show on the top. Every example contains 1000 points, the percentage underneath indicate the density ratios for each quarto of the ramp.

Image stippling is a common problem in Computer Graphics. In Fig. 8 we compare our method with the work of Fattal [2011]. We additionally compare the quality of adaptive sampling in Fig. 9 with

Balzer et al. [2009] and Fattal [2011] using a quadratic ramp as the scale function. In the coarse region, our method does not produce as good results as in the dense region. It is because the choice of h limits the influence range. If h is too large, it will cause the points to blur the given density function.

5 Varying Kernel Functions

The choice of SPH parameters and kernels directly affect the sampling patterns and distribution properties. In this section we evaluate these choices experimentally.

5.1 Trade-Off Between Noise and Aliasing

Heck and coworkers [2013] characterized the blue noise sampling property by its effective Nyquist frequency ν_{eff} and oscillation Ω . ν_{eff} measures the size of the empty low-frequency region in its Fourier power spectrum. A blue-noise sampling with high ν_{eff} can effectively reduce the low-frequency noise. Ω measures the variation in the non-zero region of a blue noise spectrum. High oscillation would result in structured aliasing [Heck et al. 2013].

Ideally the perfect blue noise shall have high ν_{eff} with low Ω . But unfortunately high ν_{eff} always comes with high Ω , thus users have to make the trade-off between those two properties, or in other words, the trade-off between noise and aliasing. While most of the existing blue noise sampling methods are implicitly making such trade-offs, there are few discussions about how to have direct control over it. In our SPH sampling method, we can control the trade-off between ν_{eff} – Ω by varying the density difference parameter.

The *density difference* is the difference between the rest density ρ_0 and the actual density $\bar{\rho}$ which is calculated as the average density of all points. To have direct control of the density difference, we can calculate $\bar{\rho}$ and adaptively set ρ_0 so that it is lower than $\bar{\rho}$ by the predefined density difference. For accuracy, only points not near the surface count towards the average density since the density of surface points are deficient.

The trade-off between ν_{eff} – Ω has to do with the convergence conditions of our algorithm. As we mentioned before, there is a trade-off between the two convergence criteria of the algorithm. When we set ρ_0 closer to $\bar{\rho}$ upon convergence, the first convergence condition will be more strictly satisfied. In this case, as is shown by experiments (Fig. 3) samples will give lower Ω while compromising ν_{eff} .

When the density difference is large, the algorithm will adhere more strictly to the second convergence condition and produce results with high ν_{eff} , at the cost of high Ω as in Fig. 2. Notice that those samples with hexagon patterns actually give the theoretical upper bound of effective Nyquist frequency [Dippé and Wold 1985].

One can easily control the trade-off between ν_{eff} and Ω by tuning the density difference, exposing a single user controlled parameter for this purpose. A lower difference refers to both low ν_{eff} and Ω , and a high difference gives samples with high ν_{eff} and Ω . The experimental results are shown in Fig. 1. By changing the density difference, we can see a smooth transition on the Fourier power spectrum.

To directly see how the density difference affects ν_{eff} and Ω , we also plot their relation in Fig. 10: providing a guide to achieve preferred ν_{eff} and Ω . Fig. 11 plots the achievable ν_{eff} against the corresponding Ω alongside other sampling methods for comparison. This curve describes what are the ν_{eff} – Ω pairs that our sampling method can achieve. The curve of SPH sampling passes through Lloyd’s relaxation and is also close to the CCVT method, and can

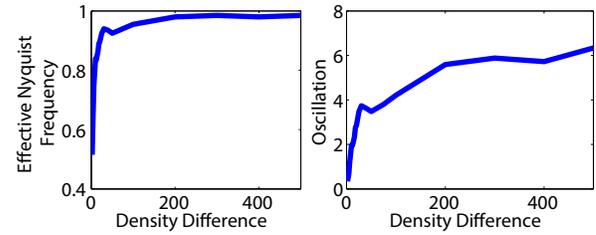


Figure 10: Effective Nyquist frequency and Oscillation changes as we increase the ‘density difference’ parameter.

also give sampling patterns in a continuous range between these two methods. Methods that are optimized for low oscillation with high effective Nyquist frequency, such as the step function and single-peak function [Heck et al. 2013] give a better trade-off than our method, although these methods are not directly tunable.

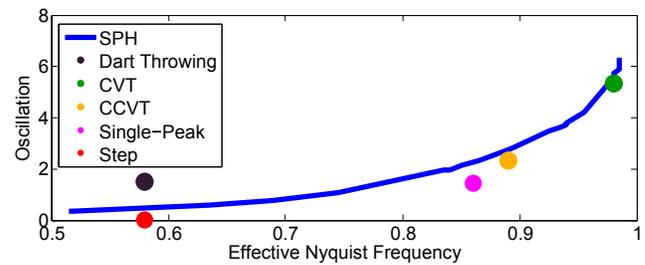


Figure 11: Trade-off between Effective Nyquist Frequency and oscillation. The blue line shows the range that our SPH sampling method can achieve, by varying the ‘density difference’ parameter. We compare with other methods by plotting their trade-off in the same figure. Note that most of them can only generate one distribution profile, therefore are represented by single dots. The data source of these methods is from Heck et al. [2013].

The technique by Chen et al. [2012] also provides trade-offs of the cut-off frequency and undulation, allowing intermediate results between CVT and CCVT. However their range of trade-off is limited by the CCVT pattern, which corresponds to the (cut-off frequency) region of 0.89–0.98 in Fig. 11. Our method, on the other hand, can go way beyond the CCVT pattern and provides a much wider range of sampling patterns, which allows for bigger trade-offs area 0.52–0.98.

As discussed above, the density difference significantly effects the sampling quality. If one wants a specific trade-off between ν_{eff} and Ω , we can easily achieve it by looking up the corresponding density difference value. For convenience, we plot the density difference parameter against effective Nyquist frequency of the 2D domain, and fit it with a curve using the regression function $f(x) = ae^{bx} + ce^{dx}$, with parameters $a = 15.28$, $b = 1.154$, $c = 0.2565$, $d = 7.729$, as is shown in Fig. 12. To achieve a specific effective Nyquist frequency ν_{eff} , one just needs to set the corresponding density difference $\Delta\rho = f(\nu_{\text{eff}})$.

5.2 Sampling with different kernels

In Table 1 we show how the sampling patterns are affected by the choice of the kernel. The kernel gradient of the pressure force (which controls the mapping from the density difference to the force) is replaced by a box kernel, tent kernel, quadratic kernel and an off-center double-peak kernel respectively. The results show that the sampling patterns are significantly influenced by the kernel

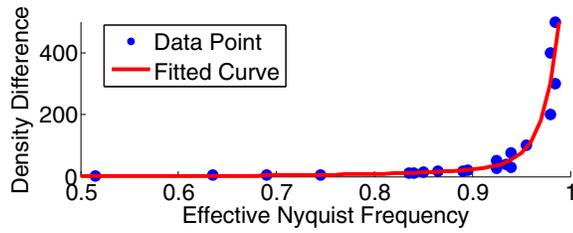


Figure 12: Plot of density difference against effective Nyquist frequency and the fitted curve in 2D domain.

choice. The tent kernel and quadratic kernel give results with blue-noise type distributions, but the other kernels generate distributions that are more like red or green noise. This set of examples provide experimental studies, and further experiments and theoretical analysis would be interesting to understand the exact relationship between the kernel and the resulting distribution.

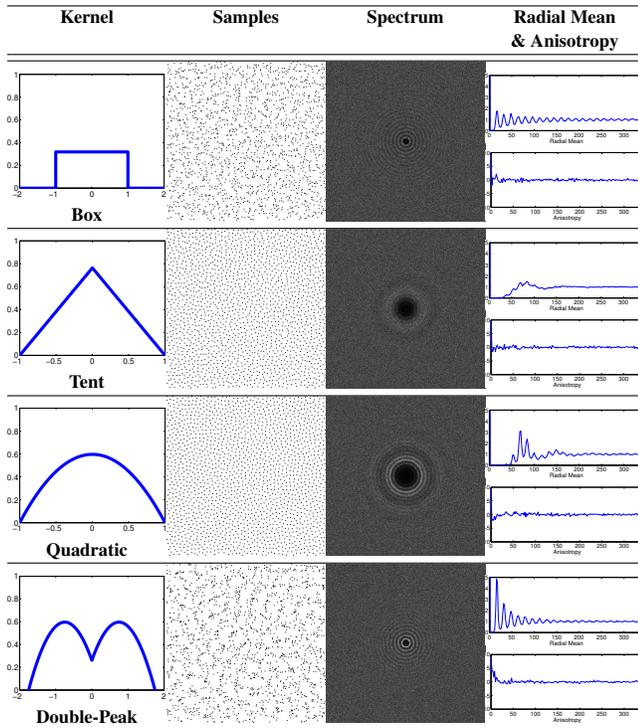


Table 1: Experimental studies of how using different kernels affects the distribution of the resulting samples. Here we show four kernels with different shapes, the corresponding samples, and Fourier spectrum analysis results.

5.3 Kernel Size

According to the Nyquist-Shannon theorem, the sufficient sampling rate is twice the bandwidth of a band-limited function, so the relationship between smooth radius h and average adjacent distance d can be written as $h \geq 2d$. Therefore, h should not be too small.

However if the object has features which are small enough, our sampling method may fail if h is chosen too large. In Fig. 13, we demonstrate the case of an object with an internal boundary. The smoothing radius must be chosen to be smaller than the minimum feature size, otherwise the sampling on the boundary will fail.

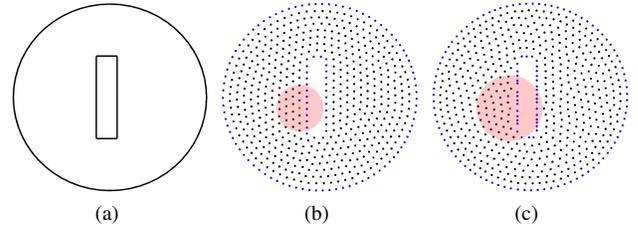


Figure 13: If the smooth radius is smaller than the feature size (b), the interior boundary will be correctly sampled. Otherwise the sampling on the interior boundary will be incorrect (c).

6 Results and Applications

6.1 Performance

Our method can be easily parallelized on the GPU. All experiments below were performed on an Intel Xeon W3680 (8M Cache, 3.33GHz) with 8GB RAM. The SPH fluid simulation is implemented in CUDA and executed on a GeForce GTX 560Ti with 1GB onboard RAM. Our GPU implementation is based on the open-source code of [Hoetzlein 2014]. We used a simple uniform grid for the neighbour searching – further acceleration can be achieved by more advanced data structures. Currently our method performs slower than [Wachtel et al. 2014], but their method is difficult to extend to general sampling domains.

The computational cost of our algorithm depends on the number of samples. Specifically, the time complexity of our method is $O(N \log N)$ for each iteration (where N is the number of samples). This is the same complexity as Lloyd’s relaxation method [Du and Emelianenko 2006]. In the table below we show the computation times of our method for generating 16384 samples with Lloyd and CCVT profiles respectively, and compare them with previous work.

	<i>Lloyd’s profile</i>	<i>CCVT profile</i>
Fast CCVT [Li et al. 2010]	-	35.75s
Lloyd’s relaxation	37.38s	-
Our method	0.687s	0.952s

Other supported features of our method, such as surface and adaptive sampling, incur additional computational costs. In the table below we show the computation times of different examples presented in this paper.

<i>Example</i>	<i>Samples</i>	<i>Iterations</i>	<i>Time(s)</i>
Fig. 4	2000	49	2.04
Fig. 6	3000	94	2.85
Fig. 7	2500	90	4.79
Fig. 8	13000	159	2.93
Fig. 15	10000	312	4.77
Fig. 16	5000	223	6.71
Fig. 17	10000	445	9.10

6.2 Multi-class sampling

The natural property of mixing fluid using SPH gives rise to the application of multi-class noise sampling similar to Wei [2010]. We defined two methods for multi-class sampling – soft disk sampling and hard disk sampling. Hard disk sampling cannot control the number of points for each class, and while soft disk sampling avoids

this problem, it is much more expensive to ensure the sample uniformity. In contrast, our method can accurately control the number for each class without adding computational cost to our SPH single sampling method.

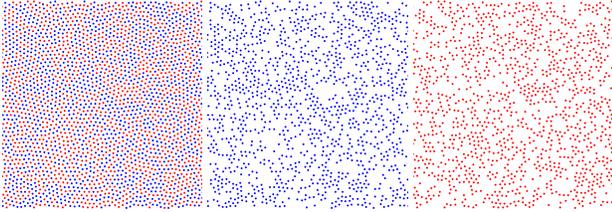


Figure 14: Multi-class sampling without different class handling fails to reproduce a blue-noise profile in each class.

We achieve multi-class blue noise sampling by giving different points a different class ID $\{c_i\}_{i=0:C-1}$ for C classes of points. We initially specify a number of samples for each class. If we run SPH for all samples without special handling, the total set will be uniform blue noise, but not for individual classes (as shown in Fig. 14).

Take two classes sampling as an example, if we want samples within the same class to be uniformly sampled, the distance between samples in the same class r_1 and distance of samples in different classes r_2 in 2D should satisfy $r_1 = \sqrt{3}r_2$ in an ideal hexagon pattern, shown on the right. This implies that we can achieve multi-class sampling by simply modifying the concept of adaptive sampling. We set the scale for samples from the same classes as s_1 , the scale for samples from different classes s_2 has to fulfill $3s_1 = s_2$ in order to achieve $r_1 = \sqrt{3}r_2$. The results are shown in Fig. 15

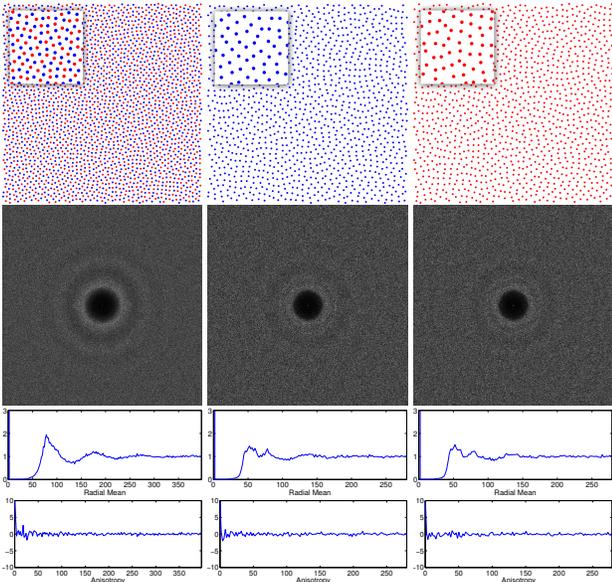
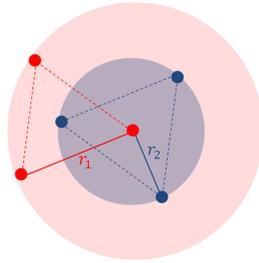


Figure 15: SPH sampling produces samples which exhibits blue noise distribution for each class as well as the total set.

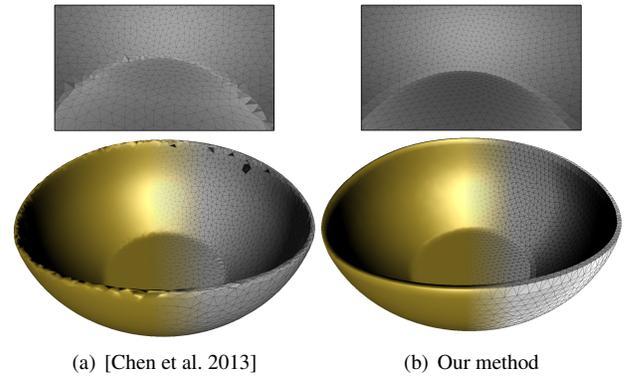
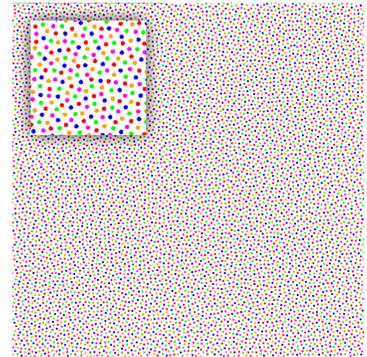


Figure 16: The surface of a bowl model remeshed using 5000 samples. With our method (b), points are implicitly attracted to regions of high curvature. In contrast, artifacts appear in (a) using [Chen et al. 2013], due to uneven sampling across the edges when sampling rate is low.

This works for multi-classes without the need of building a matrix for multi-class as proposed in [Wei 2010]. A five-class blue noise sampling is shown on the right. It should be noted that while Wei [2010] produces results that are Poisson-disk sampled, our method gives results closer to that of CCVT.



6.3 Remeshing

Our surface sampling algorithm enables us to achieve excellent surface sampling with accurate control of the sampling number, which can be used for remeshing. A good remeshing algorithm should keep sharp features of the mesh while maintaining samplers uniformly distributed. Our algorithm can easily generate uniform samples due to blue noise property. As discussed in Section 4.2 our algorithm will naturally place points in regions with sharp features.

In Fig. 16 we sample a bowl surface and mesh the result using the Ball Pivoting algorithm [Bernardini et al. 1999]. Our algorithm preserves the features of the bowl without the need for any additional techniques such as sub-sampling in regions of high curvature [Chen et al. 2013], thanks to the properties of SPH sampling described in Section 4.2. We compared the surface remeshing results of Chen et al. [2013] with ours. Our method turns out better on sharp feature preservation. Further improvement of this result by adaptive sampling based on surface curvature is a promising research direction.

6.4 Adaptive Volume Sampling

Adaptive volume sampling according to different features can be very useful for a variety of applications, such as variational tetrahedralization and volume rendering. SPH sampling allows us to choose a different size function based on the particular application.

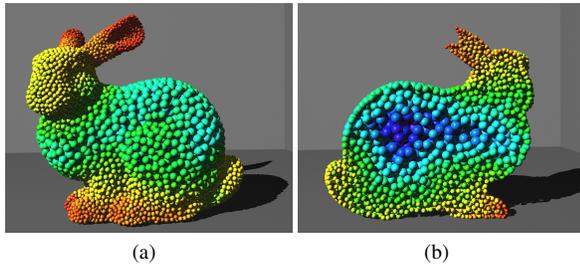


Figure 17: The bunny model is adaptively sampled based on a size function deduced using the medial axis [Adams et al. 2007]. This results in dense sampling around the ears, tail and feet, and sparse sampling in the body. The cutaway demonstrates that both the surface and internal volume are adaptively sampled well. The radius and color of each rendered particle are set according to the size function.

In Fig. 17 we sample the interior and surface of the bunny adaptively by determining the feature size using the medial axis of the manifold, similar to Adams et al. [2007]. The scale function of the adaptive sampling is set according to the computed size function. Other size functions, such as that of Alliez et al. [2005] are also applicable.

7 Conclusion and Future work

We have presented a new sampling method based on Smoothed Particle Hydrodynamics that is fast and provides controllable blue noise spectral profiles. By adjusting the rest density we are able to generate a variety of blue noise sample patterns with distribution properties that range from Lloyd’s relaxation to CCVT. The flexibility of SPH allows spatially-varying point density, leading to adaptive sampling. Our method supports multi-class blue noise sampling and surface/volume sampling in general dimensions. Moreover, SPH is easy to implement in parallel which ensures the efficiency of computation. We have also performed experimental studies of the choice of SPH kernel and its influence on the resulting samples.

There are several directions for future research. The relationship between the kernel and its resulting sample spectrum warrants further investigation. An interesting problem is to study how to automatically compute, given a target spectral distribution function, a kernel function that would lead to that target distribution. This would make it possible to use our SPH-based method to generate samples with a noise with an arbitrarily designed spectral by the user. Another possible direction is to study how to improve the trade-off between Nyquist Frequency and oscillation. The trade-off of our algorithm is currently not competitive with those of ideal blue noise or step blue noise and the single-peak blue noise discussed in Heck et al. [2013]. Finally, we would like to try other fluid simulation frameworks for the blue noise sampling and to study their trade-off tendency and potential performance improvement.

Acknowledgements

We would like to thank Nadir Akinci for initial discussions, and Xiaosong Yang, Mathieu Sanchez, Kun Qian, Guangwei Wang, Zhao Wang for their valuable suggestions. We also wish to thank all the anonymous reviewers for their useful feedbacks. This project was partially funded by Chinese Scholarship Council PhD studentships, EU FP7 612627, EU FP7 611383, and the NSF grant IIS-1423082.

References

- ADAMI, S., HU, X. Y., AND ADAMS, N. A. 2013. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics* 241 (May), 292–307.
- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph.* 26, 3 (July), 48:1–48:7.
- AKINCI, N., IHMSEN, M., AKINCI, G., SOLENTHALER, B., AND TESCHNER, M. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph.* 31, 4 (July), 62:1–62:8.
- AKINCI, N., AKINCI, G., AND TESCHNER, M. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Trans. Graph.* 32, 6 (Nov.), 182:1–182:8.
- ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., AND DESBRUN, M. 2005. Variational tetrahedral meshing. *ACM Trans. Graph.* 24, 3 (July), 617–625.
- BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: A variant of Lloyd’s method. *ACM Trans. Graph.* 28, 3 (July), 86:1–86:8.
- BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, 4 (Oct.), 349–359.
- BOWERS, J., WANG, R., WEI, L.-Y., AND MALETZ, D. 2010. Parallel poisson disk sampling with spectrum analysis on surfaces. *ACM Trans. Graph.* 29, 6 (Dec.), 166:1–166:10.
- CHEN, Z., YUAN, Z., CHOI, Y.-K., LIU, L., AND WANG, W. 2012. Variational blue noise sampling. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (Oct.), 1784–1796.
- CHEN, J., GE, X., WEI, L.-Y., WANG, B., WANG, Y., WANG, H., FEI, Y., QIAN, K.-L., YONG, J.-H., AND WANG, W. 2013. Bilateral blue noise sampling. *ACM Trans. Graph.* 32, 6 (Nov.), 216:1–216:11.
- COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph.* 5, 1 (Jan.), 51–72.
- DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. 2012. Blue noise through optimal transport. *ACM Trans. Graph.* 31, 6 (Nov.), 171:1–171:11.
- DE GOES, F., WALLEZ, C., HUANG, J., PAVLOV, D., AND DESBRUN, M. 2015. Power particles: An incompressible fluid solver based on power diagrams. *ACM Trans. Graph.* 34, 4 (July), 50:1–50:11.
- DIPPÉ, M. A. Z., AND WOLD, E. H. 1985. Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.* 19, 3 (July), 69–78.
- DU, Q., AND EMELIANENKO, M. 2006. Acceleration schemes for computing centroidal Voronoi tessellations. *Numerical Linear Algebra with Applications* 13, 2-3, 173–192.
- FATTAL, R. 2011. Blue-noise point sampling using kernel density model. *ACM Trans. Graph.* 30, 4 (July), 48:1–48:12.
- GINGOLD, R. A., AND MONAGHAN, J. J. 1977. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 375–389.

- GIRAULT, V., AND RAVIART, P. 1986. *Finite element methods for Navier-Stokes equations: theory and algorithms*. Springer series in computational mathematics. Springer-Verlag.
- GUO, J., YAN, D.-M., JIA, X., AND ZHANG, X. 2015. Efficient maximal Poisson-disk sampling and remeshing on surfaces. *Computers and Graphics* 46, 72 – 79. Shape Modeling International 2014.
- HARADA, T., TANAKA, M., KOSHIZUKA, S., AND KAWAGUCHI, Y. 2007. Real-time coupling of fluids and rigid bodies. *APCOM in conjunction with EPMESC XI*, 3-6.
- HECK, D., SCHLÖMER, T., AND DEUSSEN, O. 2013. Blue noise sampling with controlled aliasing. *ACM Trans. Graph.* 32, 3 (July), 25:1–25:12.
- HÉRAULT, A., BILOTTA, G., AND DALRYMPLE, R. 2010. SPH on GPU with CUDA. *Journal of Hydraulic Research* 48, 1, 74–79.
- HOETZLEIN, R. C. 2014. Fast fixed-radius nearest neighbors: Interactive million-particle fluids. In *GPU Technology Conference*.
- IHMSEN, M., CORNELIS, J., SOLENTHALER, B., HORVATH, C., AND TESCHNER, M. 2014. Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (Mar.), 426–435.
- IHMSEN, M., ORTHMANN, J., SOLENTHALER, B., KOLB, A., AND TESCHNER, M. 2014. SPH fluids in computer graphics. In *Eurographics 2014 - State of the Art Reports*.
- LI, H., NEHAB, D., WEI, L.-Y., SANDER, P. V., AND FU, C.-W. 2010. Fast capacity constrained voronoi tessellation. *I3D '10*, 13:1–13:7.
- LLOYD, S. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.* 28, 2 (Sept.), 129–137.
- MACKLIN, M., AND MÜLLER, M. 2013. Position based fluids. *ACM Trans. Graph.* 32, 4 (July), 104:1–104:12.
- MCCOOL, M., AND FIUME, E. 1992. Hierarchical Poisson disk sampling distributions. In *Proceedings of the Conference on Graphics Interface '92*, 94–105.
- MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. *SIGGRAPH Comput. Graph.* 21, 4 (Aug.), 65–72.
- MONAGHAN, J. J. 1994. Simulating free surface flows with SPH. *J. Comput. Phys.* 110, 2 (Feb.), 399–406.
- MONAGHAN, J. J. 2005. Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68, 8, 1703.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. *SCA '03*, 154–159.
- ÖZTIRELI, A. C., AND GROSS, M. 2012. Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph.* 31, 6 (Nov.), 170:1–170:10.
- SCHECHTER, H., AND BRIDSON, R. 2012. Ghost SPH for animating water. *ACM Trans. Graph.* 31, 4 (July), 61:1–61:8.
- SOLENTHALER, B., AND PAJAROLA, R. 2009. Predictive-corrective incompressible SPH. *ACM Trans. Graph.* 28, 3 (July), 40:1–40:6.
- SUBR, K., AND KAUTZ, J. 2013. Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Trans. Graph.* 32, 4 (July), 128:1–128:12.
- ULICHNEY, R. 1987. *Digital Halftoning*. MIT Press.
- WACHTEL, F., PILLEBOUE, A., COEURJOLLY, D., BREEDEN, K., SINGH, G., CATHELIN, G., DE GOES, F., DESBRUN, M., AND OSTROMOUKHOV, V. 2014. Fast tile-based adaptive sampling with user-specified Fourier spectra. *ACM Trans. Graph.* 33, 4 (July), 56:1–56:11.
- 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, 3 (Aug.), 20:1–20:9.
- WEI, L.-Y. 2010. Multi-class blue noise sampling. *ACM Trans. Graph.* 29, 4 (July), 79:1–79:8.
- XU, Y., LIU, L., GOTSMAN, C., AND GORTLER, S. J. 2011. Capacity-constrained delaunay triangulation for point distributions. *Computers and Graphics* 35, 3, 510 – 516.
- XU, Y., HU, R., GOTSMAN, C., AND LIU, L. 2012. Blue noise sampling of surfaces. *Computers and Graphics* 36, 4, 232 – 240.
- ZHOU, Y., HUANG, H., WEI, L.-Y., AND WANG, R. 2012. Point sampling with general noise spectrum. *ACM Trans. Graph.* 31, 4 (July), 76:1–76:11.