An algorithm for generating uniform points on a cylindrical surface

Jiayi Zhu and Hongjun Li¹

College of Science, Beijing Forestry University, Beijing 100083

¹Email: lihongjun69@bjfu.edu.cn

Abstract. The existing Poisson-disk sampling methods work well in traditional space, whatever two dimensional or three dimensional space. However, there are few methods on generating random points in non-Euclid space. For generating uniform points on the cylinder surface, we propose an algorithm which combines Poisson-Disk sampling algorithm with cylindrical distance constraints in this paper. Experiment results show that the quality of the point set generated by the proposed algorithm is better than that generated by the pseudo-random number method in both uniformity and stability.

1. Introduction

Random data generation is an essential issue in data science research and computer simulation. Researchers often use known or a priori data to extract statistics or numerical characteristics, then generate random data with those characteristics [1]. This kind of virtual data is widely used in image complementing [2], matrix interpolation, statistics, training database construction [3, 4].

In theory, any random distribution can be generated by uniform distribution, so generating data with uniform distribution is a fundamental task in statistics. People in different fields put forward a variety of methods to generate uniform distributed data, the representative methods are pseudo-random number method and Poisson-disk sampling method. Although the pseudo-random number method is convenient, the samples generated by this method are of low quality [5].

Poisson-disk sampling method is often applied in fields like digital image processing [2] and computer graphics sampling [1, 6, 7]. The spectrum of points generated by Poisson-disk sampling method has excellent properties, which is both random and uniform, and has blue noise characteristics. Dart throwing method [3] is the classic way to implement Poisson-disk sampling. The core idea of dart throwing is specifying the sampling radius. It will detect each generated sampling point’s distance, and discard the sampling point if the distance between the new sampling point and the existing sampling points is less than the specified sampling radius. Although dart throwing method could implement Poisson-disk sampling, but its efficiency is very low, in worst conditions its time complexity will be \(O(n^2)\).

In order to accelerate the Poisson-disk sampling, many optimal algorithms have been proposed. The sampling algorithms with hierarchical strategy [1, 8] could ensure the sampling procedure to terminate. Sampling within grids methods [9] will significantly accelerate the sampling efficiency but with proximate Poisson-disk distribution. A very popular algorithm called Fast Poisson-disk sampling [10, 11], which generalize maximum Poisson-disk sampling beyond two dimensions, could permits generation of Poisson-disk samples in \(O(n)\) time, and easily implement in arbitrary dimension.
There are also several analysis tools used to evaluate the quality of the sampling algorithms. Radial power and radial anisotropy are proposed in [12], which will measure the power at each spatial frequency and measure the variability of this measure respectively. Further more, tools like power spectral analysis and pair correlation function [1, 13] occurs, and these two analysis methods are wildly applied in existing sampling works.

All these existing methods work well in traditional space, whatever two dimensional (2D) or three dimensional (3D) space, for example sampling in a tetrahedron [14] or on the surface of a sphere [15, 16, 17]. However, there are few methods on generating random points in non-Euclidean space, for example on cylindrical surface. In this paper, we present a method to generate points uniformly distributed on a cylindrical surface and address a question that it is not obey the Poisson-disk distribution among the stitching region if samples on a cylinder are just a folding from a rectangle by using cylindrical transformations.

2. Cylindrical distance calculation

It is obvious that a cylindrical surface is a 2D imbedding in a 3D space. Thus, the cylinder could be considered as a constraint when generating random points in a 3D space. The cylindrical distance should be taken into account in advance.

2.1. Arc-Length calculation formula between any two points on cylinder

In 3D space, an analytic formula of cylinder \( S \) could be defined as:

\[
\begin{align*}
x &= r \cos \theta, & 0 \leq \theta \leq 2\pi, \\
y &= r \sin \theta, & -\infty \leq u \leq \infty, \\
z &= u,
\end{align*}
\]

(1)

where \( r \) denotes the radius of the cylinder, \( \theta \) and \( u \) are two parameters.

Considering the cylinder constraint occurs, the Euclidean distance is not suitable for measuring the distance between two points which is attaching on the cylindrical surface. We need to introduce Arc-Length. In order to illustrate the Arc-Length measurement, some assumptions are listed as follows.

- \( A(\theta_1, u_1) \), \( B(\theta_2, u_2) \) is point located on cylindrical surface;
- \( \rho_0(x, y, z) \) is line density;
- \( dc(AB) \) is the Arc-Length between \( A \) and \( B \).

According to formula (1), we can write the Arc(\( AB \))'s analytic formula:

\[
\Gamma(x, y, z) = \begin{cases} 
  x = r \cos(\theta_1 + t\Delta\theta) \\
  y = r \sin(\theta_1 + t\Delta\theta) \\
  z = t\Delta u
\end{cases}
\]

(2)

where \( \Delta \theta = \theta_2 - \theta_1 \) denotes the angle difference between \( A \) and \( B \), \( \Delta u = u_2 - u_1 \) denotes the height difference between \( A \) and \( B \).

If Arc(\( AB \))'s line density \( \rho_0(x, y, z) = 1 \), using the line integral theory, the Arc-Length is:

\[
dc(AB) = \int_0^1 \sqrt{\left[\frac{dx}{dt}\right]^2 + \left[\frac{dy}{dt}\right]^2 + \left[\frac{dz}{dt}\right]^2} \, dt
\]

(3)

Then

\[
dc(AB) = \int_0^1 \sqrt{(\Delta u)^2 + (r\Delta \theta)^2} \, dt
\]

(4)

i.e.

\[
dc(AB) = \sqrt{(\Delta u)^2 + (r\Delta \theta)^2}
\]

(5)

In fact, on cylinder surface, \( |\Delta \theta| \leq \pi \), so, the angle difference transformation is
\[ |\Delta \theta| = \begin{cases} |\theta_2 - \theta_1|, & |\theta_2 - \theta_1| \leq \pi \\ 2\pi - |\theta_2 - \theta_1|, & |\theta_2 - \theta_1| > \pi \end{cases} \]  

This subsection function is corresponding to the different cases (Figure 1) of angle difference.

![Figure 1](image)

**Figure 1.** The different cases of angle difference.

With formula (5) and formula (6), we obtain

\[
dc(AB) = \begin{cases} \sqrt{(u_2 - u_1)^2 + r^2(\theta_2 - \theta_1)^2}, & |\theta_2 - \theta_1| \leq \pi \\ \sqrt{(u_2 - u_1)^2 + r^2(2\pi - |\theta_2 - \theta_1|)^2}, & |\theta_2 - \theta_1| > \pi \end{cases} \tag{7} \]

This formula defines the distance on cylinder and recovers that the Arc-Length between any two points on cylinder only depends on 3 factors: angle difference \( \Delta \theta \), height difference \( \Delta u \) and radius \( r \).

2.2. Arc-Length calculation formula between any two points on ellipse

Let an elliptic cylinder be

\[
\begin{align*}
x &= a \cos \theta, & 0 \leq \theta \leq 2\pi, \\
y &= b \sin \theta, & -\infty < u < \infty, \\
z &= u,
\end{align*} \tag{8}
\]

where \( a \) is the major axis, and \( b \) is the minor axis of the ellipse. Any two points \( P(\theta_1, u_1), Q(\theta_2, u_2) \), are located on elliptic cylinder surface. Then \( Arc(PQ) \) ‘s analytic formula is

\[
\begin{align*}
x &= a \cos(\theta_1 + t\Delta \theta), \\
y &= b \sin(\theta_1 + t\Delta \theta), & 0 \leq t \leq 1, \\
z &= t\Delta u,
\end{align*} \tag{9}
\]

The Arc-Length can be calculated by

\[
dc(PQ) = \int_{\theta_1}^{\theta_2} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} \, dt \tag{10}
\]

With formula (9) and formula (10), we obtain

\[
dc(PQ) = \int_{\theta_1}^{\theta_2} \sqrt{(\Delta u)^2 + (\Delta \theta)^2 a^2 \sin^2(\theta_1 + t\Delta \theta) + (\Delta \theta)^2 b^2 \cos^2(\theta_1 + t\Delta \theta)} \, dt \tag{11}
\]

This formula can be represented as

\[
dc(PQ) = \sqrt{(\Delta u)^2 + (\Delta \theta)^2 a^2} \int_{\theta_1}^{\theta_2} \sqrt{1 - e^2 \cos^2(\theta_1 + t\Delta \theta)} \, dt \tag{12}
\]

where

\[
e = \frac{\sqrt{(\Delta u)^2 + (\Delta \theta)^2 a^2} - (\Delta u)^2 + (\Delta \theta)^2 b^2}{\sqrt{(\Delta u)^2 + (\Delta \theta)^2 a^2}} = \frac{\sqrt{(\Delta \theta)^2 a^2} - (\Delta \theta)^2 b^2}{\sqrt{(\Delta u)^2 + (\Delta \theta)^2 a^2}} \tag{13}
\]

Unfortunately, formula (12) is an elliptic integral which can only be approximately calculated [18].
3. Generating points on cylindrical surface

3.1. Mapping plane to cylinder surface
For the uniform distribution on plane, the probability density function of random variable \((X, Y)\) is

\[
p(x, y) = \begin{cases} 
1/L \times H, & (x, y) \in D, \\
0, & \text{otherwise}.
\end{cases}
\]

(14)

where \(D\) is a bounded rectangle area in two-dimensional plane, with length of \(L\) and width of \(H\).

For the case of cylindrical surface, we assume
(1). \(S\) is a cylinder in three-dimensional space, with radius of \(r\) and height of \(Z\);
(2). \(\Theta\) is a random variable which obey uniform distribution on the interval \([0, 2\pi]\), \(U\) is a random variable which obey uniform distribution on the interval \([0, H]\);
(3). \(\Theta\) and \(U\) are independent and \((\Theta, U)\) obey the two-dimensional uniform distribution.

Then, the probability density function of \((\Theta, U)\) is

\[
p(\theta, u) = \begin{cases} 
1/2\pi \times Z, & (\theta, u) \in S, \\
0, & \text{otherwise}.
\end{cases}
\]

(15)

The coordinate transform function between the rectangle and the cylindrical surface is

\[
\begin{align*}
& r = \frac{L}{2\pi} \\
& \theta = 2\pi \frac{x}{L} \quad x \in [0, L], \quad y \in [0, H] \\
u &= y
\end{align*}
\]

(16)

It is easy to prove that the points that obey uniform distribution on the plane also obey uniform distribution on the cylinder with the map formula (16), which means that points uniform on the cylinder can be generated by the way of mapping.

3.2. Points on elliptic cylinder surface
It is not a good approach to generate points on an elliptic cylinder surface by the same mapping as on the cylinder surface (Subsection 3.1). This can be illustrated by an example as follows.

From Figure 2, we found that the random points generated on the elliptic cylinder are not “uniform”. There are significantly more points around the major axis of the ellipse than around the minor axis (Figure 2(b)). Therefore, this is what should be paid attention to in algorithm construction.

![Figure 2](image)

**Figure 2.** Generating random points on an elliptic cylinder by the method as in Subsection 3.1.

3.3. Mapping formula from elliptic cylinder to plane
It is obviously to know that a rectangular paper can be fold into one unique cylinder. However, it can be rolled up into various elliptic cylinders with different major axis and minor axis. So we only give the unidirectional transformation formula of elliptic cylinder to plane.

With formula (11), assuming that point \(M(\theta, u)\) and \(N(\theta, u)\) are on an elliptic, then

\[
dc(MN) = \int_{0}^{1} \sqrt{\left(\frac{a}{b}\right)^2 \sin^2(t\theta) + \left(\frac{b}{a}\right)^2 \cos^2(t\theta)} dt
\]

(17)
Let \((\theta, u)\) be the coordinate of a point on the elliptic cylinder surface, and \((x, y)\) be the coordinate that transformed into rectangle. With formula (17), we obtain the mapping formula (18) from elliptic cylinder to plane

\[
\begin{align*}
    x &= \int_{0}^{1} \sqrt{\left(\theta^2 a^2\right) \sin^2(\theta t) + \left(\theta^2 b^2\right) \cos^2(\theta t)} dt \\
    y &= u
\end{align*}
\]

(18)

We use the adaptive Simpson integral method to calculate the elliptic integral formula (18), and find a pair of circle and ellipse with approximate equal circumference. It is easy to prove that on elliptic cylinder, when \(\theta\) obeys uniform distribution on interval \([0, 2\pi]\), \(x\) is non-uniformly distributed.

4. Simulation
This section takes several experiments of random points generation on a cylinder. The sampling area is defined in a rectangle area with length of \(L = 10\) and width of \(H = 10\), and sampling number \(n = 1000\). Our algorithms is written in Matlab 2018b and tested on laptops with 2.5ghz CPU and 8G memory.

4.1. Poisson-Disk sampling algorithm with cylinder distance constraint
In practical applications, Poisson-disk Sampling method can often obtain a more uniform sampling point set. In this paper, we will use dart-throwing method to generate Poisson-disk sample. The core idea of the method is to define a sampling radius \(r\) and conduct point-distance detection for each newly generated sampling point. If the distance \(R\) between the new sampling point and any existing sampling point is less than the specified sampling radius \(r\), the sampling point will be abandoned and re-sample. It should be noted that cylindrical distance (formula (7)) is adopted in this algorithm and the minimum distance \(R\) is obtained by naive comparing of all distance values. Our algorithm is called Poisson-Disk Sampling algorithm with Cylindrical Distance Constraints (CDC-PDS) as below.

**Algorithm: Poisson-Disk sampling algorithm with cylindrical distance constraint**

**Input:** cylinder radius \(r\), cylinder height \(H\), sampling number \(N\) and sampling radius \(r_0\)

**Output:** sampling point set \(X\)

1: Initialization: randomly generate one point in \(\Omega\), and count it in the points set \(X\);
2: while true do
3: randomly generate one point in \(\Omega\), and detect the distance \(R\) from all points in \(X\);
4: if \(R > r\) then
5: count this point to \(X\);
6: else
7: abandon this point and go back to step 3;
8: end if
9: end while
10: return \(X\)

The main features of our CDC-PDS algorithm are as follows:

- The time complex of the algorithm is \(O(n^2)\) according to dart-throwing method [3].
- Adopting the strategy of transforming points in the plane rectangular area to cylinder random points. By converting points in the plane rectangular area to points on the cylinder surface, calculation and randomness evaluation could be more convenient.
- Adopting the strategy of dart-throwing method based on cylinder distance. As described in section 2, the Euclidean distance originally used to measure the distance between two points in three-dimensional space is not suitable for measuring the distance between two points "attached" on the cylinder surface.

In order to evaluate the effectiveness of our CDC-PDS algorithm, we compare it with the following 2 algorithms:

**Algorithm A:** generate points set in the rectangle domain \(\Omega\) by pseudo-random method, then transform the points set to cylinder surface.
Algorithm B: generate points set in the rectangle domain $\Omega$ by Euclidean distance based Poisson-Disk sampling method, then transform the points set to cylinder surface.

What should be mentioned is that we use the “default” radius $r_0$ as sampling radius when we employ the CDC-PDS algorithm, which will be illustrated in Section 4.4.

From Figure 3, we can found that some points generated by the pseudo-random number method (Figure 3(a) and 3(d)) are "stuck" together, making the points set less "uniform" than those points generated by Algorithm B (Figure 3(b) and 3(e)) and CDC-PDS algorithm (Figure 3(c) and 3(f)).

![Figure 3](image-url)

**Figure 3.** Comparison of 3 algorithms: Algorithm A (The first column), Algorithm B (The second column) and CDC-PDS (The last column).

4.2. Randomness evaluation

In this paper, non-parametric statistical tests on the coordinate values of the sampling points sets will be applied directly to determine whether the points set obeys uniform distribution and evaluate the quality of the sampling points set under different generation methods.

4.2.1. K-S test. The K-S test is to calculate the maximum distance between the empirical distribution function and the theoretical distribution function to determine whether the sample is from the hypothesis distribution.

We first check the $x$-axis coordinate. Since the experiments in this paper sampled 1000 points in a $10 \times 10$ rectangular domain, the $x$-axis coordinate of the sampling points are from the uniform distribution $X \sim U(0, 10)$. Assuming that the empirical distribution function of $X$ is $F(x)$, then the hypotheses of K-S test for $x$-axis data are

$$H_0 : F(x) = F_0(x), \text{ vs. } H_1 : F(x) \neq F_0(x)$$
We repeat the experiment which generates points by the pseudo-random number method (Algorithm A) 10 times, and carry out K-S test on the results of each experiment. The experimental results are listed in Table 1. In this table, \( D_X \), \( p \)-value and \( Result_X \) are derivation, \( p \)-value and test result of \( x \)-axis data respectively. The last 3 columns are those indexes for \( y \)-axis data. The results show that among the 20 time K-S test results, there is one result that accepts \( H_1 \). This means that the \( y \)-axis coordinate values did not obey uniform distribution in this time. According to the two-dimensional uniform distribution theory, if \( Y \) does not obey the uniform distribution, then \((X, Y)\) does not obey the uniform distribution.

Similarly, we examine the sampling points set generated by our CDC-PDS method. The K-S test results (\( \alpha = 0.05 \)) are listed in Table 2. In this table, \( x \)-axis coordinate value and \( y \)-axis coordinate value are all pass in K-S test.

From the perspective of \( p \)-value, although most of the sample test results of the pseudo-random number method accept the null hypothesis, the \( p \)-value of the test is unstable, as shown in Table 1. Among the experimental results of the pseudo-random number method, the smallest \( p \)-value that passes the K-S test is 0.1577. However, the experimental results of CDC-PDS have very high \( p \)-value. The smallest \( p \)-value is 0.5705, while the highest \( p \)-value is close to 1, which indicates that the empirical distribution function of samples is very close to the theoretical distribution function, as shown in Table 2. The results show that the quality of the points set generated by CDC-PDS method is better than that generated by the pseudo-random number method in both uniformity and stability.

**Table 1.** K-S test results for points generated by Algorithm A. (\( \alpha = 0.05 \)).

| \( D_X \) | \( p \)-value | \( Result_X \) | \( D_Y \) | \( p \)-value | \( Result_Y \) |
|---|---|---|---|---|---|
| 1  | 0.019289 | 0.8508 | \( H_0 \) | 0.024761 | 0.5720 | \( H_0 \) |
| 2  | 0.020604 | 0.7896 | \( H_0 \) | 0.045074 | 0.03438 | \( H_1 \) |
| 3  | 0.027891 | 0.4181 | \( H_0 \) | 0.023335 | 0.6475 | \( H_0 \) |
| 4  | 0.019159 | 0.8564 | \( H_0 \) | 0.019933 | 0.8218 | \( H_0 \) |
| 5  | 0.029809 | 0.3366 | \( H_0 \) | 0.03495 | 0.1737 | \( H_0 \) |
| 6  | 0.024593 | 0.5808 | \( H_0 \) | 0.025297 | 0.5442 | \( H_0 \) |
| 7  | 0.032023 | 0.2567 | \( H_0 \) | 0.020503 | 0.7945 | \( H_0 \) |
| 8  | 0.029072 | 0.3666 | \( H_0 \) | 0.035636 | 0.1577 | \( H_0 \) |
| 9  | 0.030835 | 0.2977 | \( H_0 \) | 0.023633 | 0.6317 | \( H_0 \) |
| 10 | 0.031469 | 0.2753 | \( H_0 \) | 0.031941 | 0.2594 | \( H_0 \) |

**Table 2.** K-S test results for points generated by CDC-PDS algorithm. (\( \alpha = 0.05 \)).

| \( D_X \) | \( p \)-value | \( Result_X \) | \( D_Y \) | \( p \)-value | \( Result_Y \) |
|---|---|---|---|---|---|
| 1  | 0.020391 | 0.8 | \( H_0 \) | 0.017884 | 0.9064 | \( H_0 \) |
| 2  | 0.007771 | 1 | \( H_0 \) | 0.018262 | 0.8926 | \( H_0 \) |
| 3  | 0.0146 | 0.9834 | \( H_0 \) | 0.01729 | 0.926 | \( H_0 \) |
| 4  | 0.016462 | 0.9492 | \( H_0 \) | 0.01723 | 0.9279 | \( H_0 \) |
| 5  | 0.024789 | 0.5705 | \( H_0 \) | 0.012672 | 0.9971 | \( H_0 \) |
| 6  | 0.024593 | 0.5808 | \( H_0 \) | 0.011243 | 0.9996 | \( H_0 \) |
| 7  | 0.008778 | 1 | \( H_0 \) | 0.020669 | 0.7864 | \( H_0 \) |
| 8  | 0.009829 | 1 | \( H_0 \) | 0.021689 | 0.7346 | \( H_0 \) |
| 9  | 0.011435 | 0.9994 | \( H_0 \) | 0.012086 | 0.9986 | \( H_0 \) |
| 10 | 0.015344 | 0.9726 | \( H_0 \) | 0.01819 | 0.8953 | \( H_0 \) |

### 4.2.2. Chi-square test

We use Chi-square test to evaluate the quality of points generated by our CDC-PDS algorithm. Based on the distribution of \( x \)-axis data (formula (19)), we divided the interval \([0, 10]\) into several blocks with length 1 and 0.5 respectively. When the block length is 1, the interval \([0, 10]\) is divided into 10 blocks. Theoretically, the number of sample points falling in each interval should be 100. Assume the actual observed value of the sample size in interval \( i \)th is \( n_i \), so the hypotheses are:
We collect the 10 times results of pseudo-random number (Alg.A) and CDC-PDS sampling experiments, and conduct Chi-square test in 10 and 20 blocks for the results of each experiment respectively. The experimental results are listed in Table 3. In the table, \( p_{\text{min}}^{(1)} \) and \( p_{\text{max}}^{(1)} \) are the minimum and the maximum \( p\)-value respectively in the 10 blocks experiment, while \( p_{\text{min}}^{(2)} \) and \( p_{\text{max}}^{(2)} \) are the minimum and the maximum \( p\)-value in the 20 blocks experiment. \( \eta \) is the pass rate. From the table, it can also be safely concluded that the point sets generated by CDC-PDS method are superior to the pseudo-random number method in both the test’s pass rate and the stability of \( p\)-value.

### Table 3. Chi-square test results for points generated by Alg.A and CDC-PDS.

| Data           | \( p_{\text{min}}^{(1)} \) | \( p_{\text{max}}^{(1)} \) | \( \eta_1 \) | \( p_{\text{min}}^{(2)} \) | \( p_{\text{max}}^{(2)} \) | \( \eta_1 \) |
|----------------|-----------------|-----------------|----------|-----------------|-----------------|----------|
| Alg.A: x-axis  | 0.1608          | 0.9814          | 100%     | 0.0212          | 0.9814          | 90%      |
| Alg.A: y-axis  | 0.1738          | 0.9798          | 100%     | 0.0811          | 0.8940          | 100%     |
| Ours: x-axis   | 0.8111          | 0.9983          | 100%     | 0.8487          | 0.9997          | 100%     |
| Ours: y-axis   | 0.7238          | 0.9984          | 100%     | 0.9170          | 0.9999          | 100%     |

### 4.3. Analysis of sampling radius

We tested the dart-throwing method’s sampling efficiency under different sampling radius \( r_0 \). For sampling number \( n = 1000 \), the average sampling time corresponding to different sampling radius \( r_0 \) is calculated. It was found that when \( r_0 = 0.26 \), the sampling time reaches the maximum. If we continue increase the sampling radius, the algorithm will fall into an endless loop. Experiment results show that if \( r_0 \) is too small, the generated sample point set is not very different from the pseudo-random number method, and if \( r_0 \) is too big, dart-throwing method will crash. Different from progressive refinement strategy [8], we present a default sampling radius formula according to our experiments:

\[
\eta_0 = \sqrt{\frac{L \times H}{n \pi}}
\]

(19)

Noted that the default sampling radius \( r_0 \) is not necessarily the optimized sampling radius.

### 5. Conclusions

Aiming at the problem of generating uniform random points on cylinder surface, an algorithm, Poisson-Disk Sampling algorithm with Cylindrical Distance Constraints (CDC-PDS), is proposed. The highlights of CDC-PDS algorithm include three aspects.

The new algorithm addresses a question that it is not obey the Poisson-disk distribution among the stitching region if samples on cylinder are just folding from a rectangle by using cylindrical transformations;

The cylinder distance is calculated using a subsection function and the Poisson-disk sampling method is extended and used for generating uniform points set on the cylinder surface;

Experimental results show that the new method in this paper is effective. According to the results of K-S test and Chi-square test, the quality of the points set generated by CDC-PDS method is better than that generated by the pseudo-random number method in both uniformity and stability.

### Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (No. 2019SG05).

### References

[1] Öztireli C and Gross M 2012 Analysis and synthesis of point distributions based on pair correlation. *ACM Transactions on Graphics* (TOG) 31

[2] Yang X, Li G, Sun J et al. 2019 High-resolution and wide-swath sar imaging via Poisson-disk sampling and iterative shrinkage thresholding. *IEEE Transactions on Geoscience and Remote
Sensing 57(7) 1-13

[3] Cook R 1986 Stochastic sampling in computer graphics. ACM Trans. Graph. 5 51-72

[4] Houkjær K, Torp K and Wind R 2006 Simple and realistic data generation. Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea pp. 1243-1246

[5] Carpintero D D 1990 About the problem of generating three-dimensional pseudo-random points. Boletin De La Asociacion Argentina De Astronomia La Plata Argentina 36 166-174

[6] Hermosilla P, Ritschel T, Vázquez et al. 2018 Monte carlo convolution for learning on non-uniformly sampled point clouds. ACM Trans. Graph. 37(6) 235:1-235:12

[7] Wu H, Lyu X and Wen Z 2018 Automatic texture exemplar extraction based on global and local textureness measures. Computational Visual Media 4(2) 1-12

[8] White K, Cline D and Egbert P 2007 Poisson disk point sets by hierarchical dart throwing. Interactive Ray Tracing pp. 129-132

[9] Ebeida M, Patney A, Mitchell S et al. 2011 Efficient maximal Poisson-disk sampling. ACM Transactions on Graphics, 30(4) 49.1-49.12

[10] Bridson R 2007 Fast Poisson disk sampling in arbitrary dimensions. SIGGRAPH '07: ACM SIGGRAPH 2007 sketches. 22

[11] Ebeida M S, Mitchell S A, Patney A et al. 2012 A Simple Algorithm for Maximal Poisson-Disk Sampling in High Dimensions[J]. Computer Graphics Forum 31(2pt4) 785-794

[12] Mccool M 1992 Hierarchical Poisson Disk Sampling Distributions. Proc of Graphics Interface 94-105

[13] Heck Dl, Schlömer T and Deussen O 2013 Blue Noise Sampling with Controlled Aliasing[J]. Acm Transactions on Graphics 32(3) 25.1-25.12

[14] Rocchini C and Cignoni P 2000 Generating random points in a tetrahedron. Journal of Graphics Tools 5(4) 9-12

[15] Tashiro Y 1977 On methods for generating uniform random points on the surface of a sphere. Annals of the Institute of Statistical Mathematics 29(1) 295-300

[16] Liang J and Ng K W 2008 A method for generating uniformly scattered points on the lp-norm unit sphere and its applications. Metrika 68(1) 83-98

[17] Zhou R, Chen C, Sun L et al. 2018 Online uniformly inserting points on the sphere. Algorithms 11(10) 156.1-156.13

[18] Good H R 2001 Elliptic integrals, the forgotten functions. European Journal of Physics 22(2) 119-126