Generating Stratified Random Lines in a Square

Peter Shirley
NVIDIA

Chris Wyman
NVIDIA

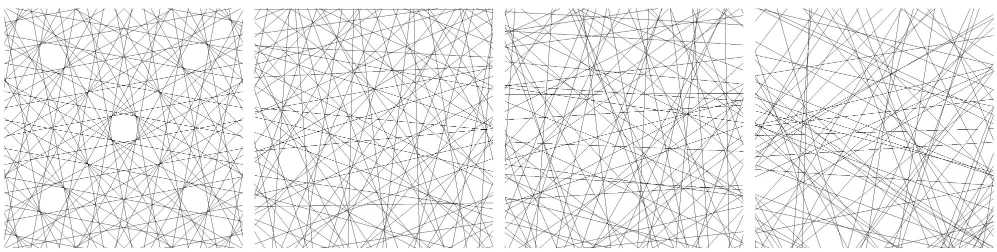

Figure 1. One hundred uniform lines in the square. The lines are generated with uniform seed points $(\xi_1, \xi_2) \in [0,1]^2$ (left to right): regular lattice, Hammersley, jittered (stratified), uniformly random.

Abstract

When generating a set of uniformly distributed lines through a square, some care is needed to avoid bias in line orientation and position. We present a compact algorithm to generate unbiased uniformly distributed lines from a uniform point set over the unit square.

1. Lines in the Square

It is often useful to generate sets of uniform lines through a square (e.g., our own use case to test antialiasing algorithms on a sampling of all possible edges, or to antialias using line samples rather than point samples [Jones and Perry 2000; Singh et al. 2017]). Our lines are restricted to 2D as opposed to the 3D lines we see in many ray-tracing and light-field applications, or the line segments in 2D with an underlying 3D parameterization [Sun et al. 2013]. Our application is typical in benefitting from stratification, which often changes the order of convergence of sampling-based algorithms [Mitchell 1996]. This paper is a reaction to our experience translating what is available in the literature into code; the code ended up being somewhat involved and difficult to debug until we used a polar parameterization that allowed negative radii which results in very compact code. This does not produce better stratified lines than what is already in the literature, but it does both spell out the details needed for implementation as well as result in a small fast implementation.

 ISSN 2331-7418

We use the positive unit square $(x, y) \in [0, 1]^2$. Mitchell [1992] presented a way to generate uniform lines from random triples (ξ_1, ξ_2, ξ_3) in the unit cube using the slope intercept (m, b) dual space (where the line equation is $y = mx + b$):

$$m = \frac{\xi_1 - 0.5}{\xi_2 - 0.5}, \qquad b = \begin{cases} (1-m)\xi_3 & \text{if } m < 0 \\ (-1-m)\xi_3 + 1, & \text{otherwise.} \end{cases}$$

This method is fast and effective but does not easily enable 2D stratification over the dual space, since it relies on 3D random input. Literature in computational geometry discusses line-space measures that provide a methodology to sample the 2D dual spaces [Santaló 2004]. Marschner et al. [2015] sketch some of the details of two of the popular computational geometry approaches to uniformly sample these spaces, one in slope-intercept space ((m, b) space) and one in polar space ((r, θ) space), but leave details and boundary cases unresolved. This paper grew from trying to fill in those details, but we found the resulting code fairly complicated because of the number of cases that arose from the multiple sides of the square. We got the idea for the improved method from two things: the recurring $\sin\theta$ term that arises and the great simplification of square-disk mappings resulting from using negative radii as proposed by David Cline [Shirley 2011]. In this paper, we sample in $(r, \sin\theta)$ space and provide compact and efficient code that covers all cases.

Unfortunately, the dual space $(r, \sin\theta)$ does not have a bijection with the full circle (since $\sin(\theta) = \sin(\pi - \theta)$). We instead use $\theta \in [-\frac{\pi}{2} \ldots \frac{\pi}{2})$ and allow negative r, as shown in Figure 2. This improves stratification quality by ensuring nearby $(r, \sin\theta)$ correspond to nearby lines (and vice versa). It also avoids using inverse trigonometric operations. We choose random lines by first selecting $\sin\theta$ and then choosing lines uniformly in the corresponding range of valid r. The observation that the number of

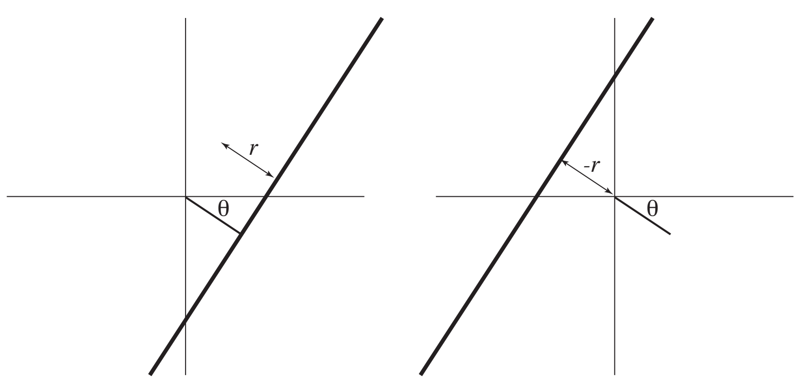

Figure 2. Left: A line specified by angle θ and the perpendicular distance r from the origin. Right: Instead of varying θ in $[0 \ldots 2\pi)$ with positive radii, we use θ in $\left[-\frac{\pi}{2} \ldots \frac{\pi}{2}\right)$ and allow negative radii. For the two lines shown, θ is the same but their radii vary in sign. This ensures two nearby lines (e.g., $\pm r$ near zero) are also nearby in the dual $(r, \sin\theta)$ space.

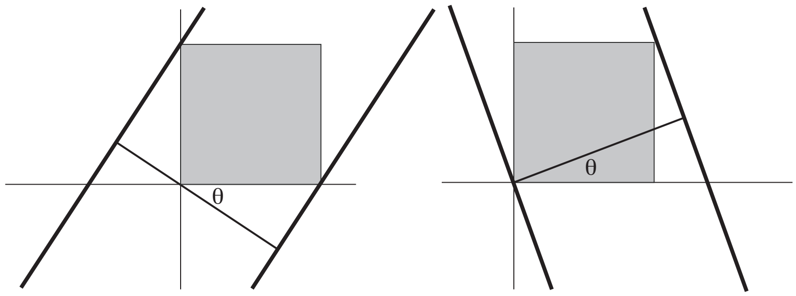

Figure 3. The most range of positive and negative radii that hit the square for a given θ. The "measure" of the set of lines that share that theta is proportional to the length of the line in the figure and thus maximum at $\theta = \pm\pi/4$; the generated lines at a given θ are perpendicular to that line, and the longer that line is, the more possible lines that are perpendicular.

lines at a given orientation is proportional to the cross section of the square at that angle (Figure 3) yields

$$p(\theta) \propto \begin{cases} \cos\left(\theta + \frac{\pi}{4}\right) & \text{if } \theta < 0 \\ \cos\left(\theta - \frac{\pi}{4}\right), & \text{otherwise.} \end{cases}$$

We treat these two cases independently, as each has the same number of lines. Using the standard method of integrating p to get a cumulative distribution function and inverting, we get two functions for positive and negative θ. This gives two cumulative probability distribution functions each varying from 0 to 1:

$$P(\theta) = \begin{cases} \frac{\sqrt{2}}{2} \int_{-\frac{\pi}{2}}^{\theta} \cos\left(\theta + \frac{\pi}{4}\right) d\theta & \text{if } \theta < 0 \\ \frac{\sqrt{2}}{2} \int_{0}^{\theta} \cos\left(\theta - \frac{\pi}{4}\right) d\theta, & \text{otherwise.} \end{cases}$$

Integrating yields

$$P(\theta) = \begin{cases} \frac{1}{2} + \frac{\sqrt{2}}{2} \sin\left(\theta + \frac{\pi}{4}\right) & \text{if } \theta < 0 \\ \frac{1}{2} + \frac{\sqrt{2}}{2} \sin\left(\theta - \frac{\pi}{4}\right), & \text{otherwise.} \end{cases}$$

To maintain stratification, we can use the first random (or stratified) number ξ_1 sample both functions by scaling $[0, 0.5)$ to $[0, 1)$ when $\xi_1 < 0.5$, and scaling $[0.5, 1)$ to $[0, 1)$ otherwise. These yield two expressions to be inverted, the first when $\xi_1 < 0.5$:

$$2\xi_1 = \frac{1}{2} + \frac{\sqrt{2}}{2} \sin\left(\theta + \frac{\pi}{4}\right), \tag{1}$$

and then when $\xi_1 > 0.5$:

$$2(\xi_1 - 0.5) = \frac{1}{2} + \frac{\sqrt{2}}{2} \sin\left(\theta - \frac{\pi}{4}\right). \tag{2}$$

These equations can be solved for $\sin(\theta + k)$, but we would really like to solve for $\sin\theta$. Both of these equations are of the form

$$\sin\left(\theta \pm \frac{\pi}{4}\right) - C = 0.$$

Solving for θ:

$$\theta = \arcsin C \mp \frac{\pi}{4}.$$

So solving for $\sin\theta$ yields

$$\sin\theta = \sin\left(\arcsin C \mp \frac{\pi}{4}\right).$$

From the identity $\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \sin\beta\cos\alpha$, we have

$$\sin\theta = \sin\left(\arcsin C\right)\cos\frac{\pi}{4} \mp \sin\frac{\pi}{4}\cos\left(\arcsin C\right),$$

which reduces to

$$\sin\theta = \frac{\sqrt{2}}{2}\left(C \mp \sqrt{1 - C^2}\right).$$

In the case of Equation (1),

$$C = \frac{4\xi_1 - 1}{\sqrt{2}},$$

and for Equation (2),

$$C = \frac{4\xi_1 - 3}{\sqrt{2}}.$$

This gives us the two corresponding equations,

$$\sin\theta = \frac{1}{2}\left((4\xi_1 - 1) - \sqrt{2 - (4\xi_1 - 1)^2}\right)$$

for $\xi_1 < 1/2$, and

$$\sin\theta = \frac{1}{2}\left((4\xi_1 - 3) + \sqrt{2 - (4\xi_1 - 3)^2}\right)$$

for $\xi_1 \geq 1/2$. This yields an equation with two cases:

$$\sin\theta = \begin{cases} \frac{1}{2}\left((4\xi_1 - 1) - \sqrt{2 - (4\xi_1 - 1)^2}\right) & \text{if } \xi_1 < 0.5 \\ \frac{1}{2}\left((4\xi_1 - 3) + \sqrt{2 - (4\xi_1 - 3)^2}\right), & \text{otherwise.} \end{cases}$$

```
// all variables double
// x1, x2 are uniform random numbers on [0,1)
v = 1 - fabs(4*x1 - 2);
sinT = copysign( 0.5 * (v - sqrt(2-v*v)), x1-0.5 );
r = fmin(sinT,0) + x2 * (sqrt(1-sinT*sinT) + fabs(sinT));
```

Listing 1. A mapping from uniform samples to uniform lines. The `copysign` function allows the two "if" cases to be collapsed into one code statement.

We can use this equation as is, but the two cases are similar enough that perhaps we can get them to share terms in code form. If we define a variable

$$u = 4\xi_1 - 2,$$

then we have something similar to the two C above, but also a variable that is negative when $\xi > 0.5$ and non-negative otherwise, so its sign can serve to distinguish cases. So we can also transform u to one equation for all values of ξ:

$$v = 1 - |u| = \begin{cases} 4\xi_1 - 1 & \text{if } \xi_1 < 0.5 \\ -4\xi_1 + 3, & \text{otherwise.} \end{cases}$$

We will be able to use this along with the `copysign` function to avoid an explicit branch.

Once a θ is chosen, we then choose r uniformly for the legal values for that θ. As can be derived from the range of r hitting the square in Figure 3, this gives

$$r = \begin{cases} \sin\theta + \xi_2(\cos\theta - \sin\theta) & \text{if } \theta < 0 \\ \xi_2(\cos\theta + \sin\theta), & \text{otherwise.} \end{cases}$$

Solving for ξ_1 and picking an r yields equations that can be simplified by grouping common subexpressions using a sign function; they can be made surprisingly compact as shown in Listing 1.

The random input can be from any uniform points on $[0, 1)^2$ as shown in Figure 1. The results for a jittering for a variety of numbers of lines are shown in Figure 4.

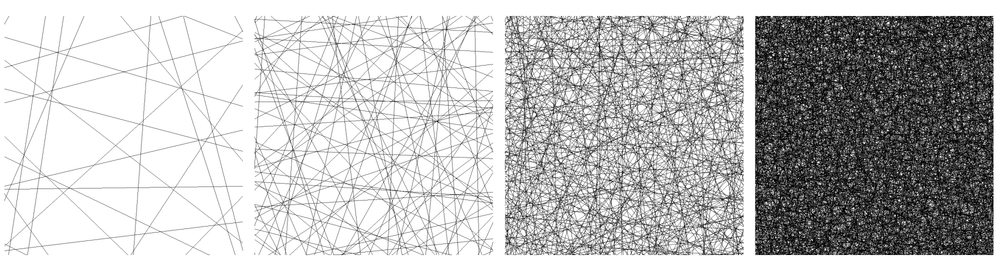

Figure 4. Jittered input: 25, 100, 400, 1600 lines.

References

JONES, T. R., AND PERRY, R. N. 2000. Antialiasing with line samples. In *Proceedings of the Eurographics Workshop on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, 197–206. 48

MARSCHNER, S., SHIRLEY, P., GLEICHER, M., HOFFMAN, N., JOHNSON, G., MUNZNER, T., REINHARD, E., THOMPSON, W. B., , AND WYVILL, B. 2015. *Fundamentals of Computer Graphics*, fourth ed. CRC Press, Boca Raton, FL. 49

MITCHELL, D. 1992. Ray tracing and irregularities of distribution. In *Proceedings of the Eurographics Workshop on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, 61–69. 49

MITCHELL, D. 1996. Consequences of stratified sampling in graphics. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, SIGGRAPH '96, 277–280. 48

SANTALÓ, L. A. 2004. *Integral Geometry and Geometric Probability*. Cambridge University Press, Cambridge, UK. 49

SHIRLEY, P., 2011. Blog post: Improved code for concentric map. `http://psgraphics.blogspot.com/2011/01/improved-code-for-concentric-map.html`, January. Accessed: 2017-03-11. 49

SINGH, G., MILLER, B., AND JAROSZ, W. 2017. Variance and convergence analysis of monte carlo line and segment sampling. *Computer Graphics Forum (Proceedings of EGSR) 36*, 4. 48

SUN, X., ZHOU, K., GUO, J., XIE, G., PAN, J., WANG, W., AND GUO, B. 2013. Line segment sampling with blue-noise properties. *ACM Trans. Graph. 32*, 4 (July). 48

Author Contact Information

Peter Shirley
NVIDIA
2150 S 1300 East
Suite 500
Salt Lake City, UT 84106
pshirley@nvidia.com
https://research.nvidia.com/person/peter-shirley

Chris Wyman
NVIDIA
11431 Willows Road NE
Suite 200
Redmond, WA 98052
cwyman@nvidia.com
https://research.nvidia.com/person/chris-wyman

Peter Shirley, Chris Wyman, Generating Stratified Random Lines in a Square, *Journal of Computer Graphics Techniques (JCGT)*, vol. 6, no. 2, 48–54, 2017
http://jcgt.org/published/0006/02/03/

Received:	2016-08-22		
Recommended:	2016-11-11	Corresponding Editor:	Marc Stamminger
Published:	2017-06-30	Editor-in-Chief:	Marc Olano