Signal Processing and Sampling

CS 348B, Lecture 7

Now with color demos!

"the aliasing side" by FabriceNeyret2
https://www.shadertoy.com/view/4slXR4

Announcements

HW2: A Ray Marching-Based Distance Estimator

- Due Thursday

Participation Credit:

- Two comments per lecture

Office hour changes this week:

- Doug: Thurs cancelled
- Kayvon: Wed 11:30-1pm & Thur 5-6pm (Gates 366)

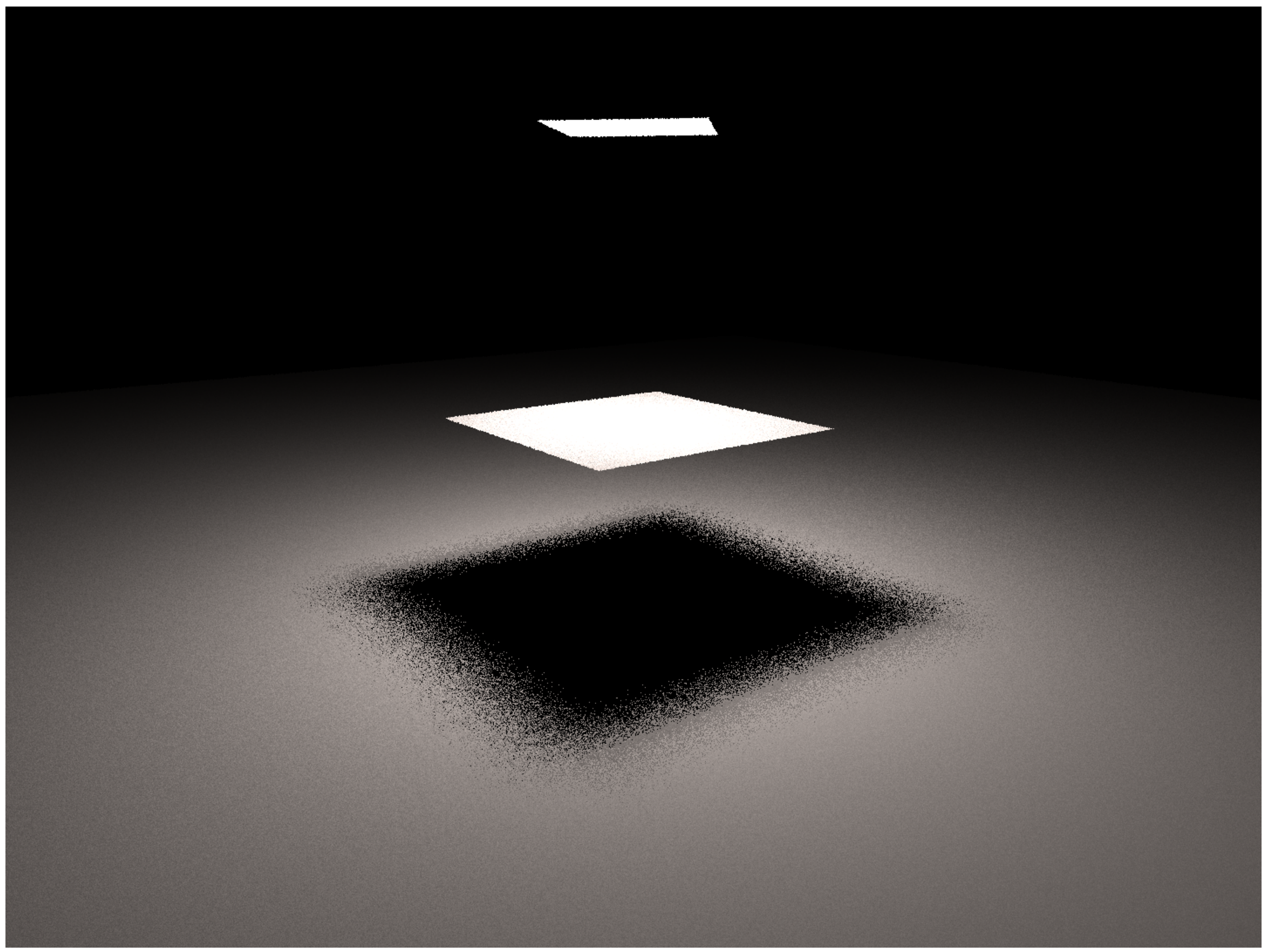
Mathematica demos for today:

https://www.dropbox.com/s/nzyzq92e9z89fpk/sampling.nb

https://www.dropbox.com/s/w7yf21v4f9r08vf/sampling.nb.pdf

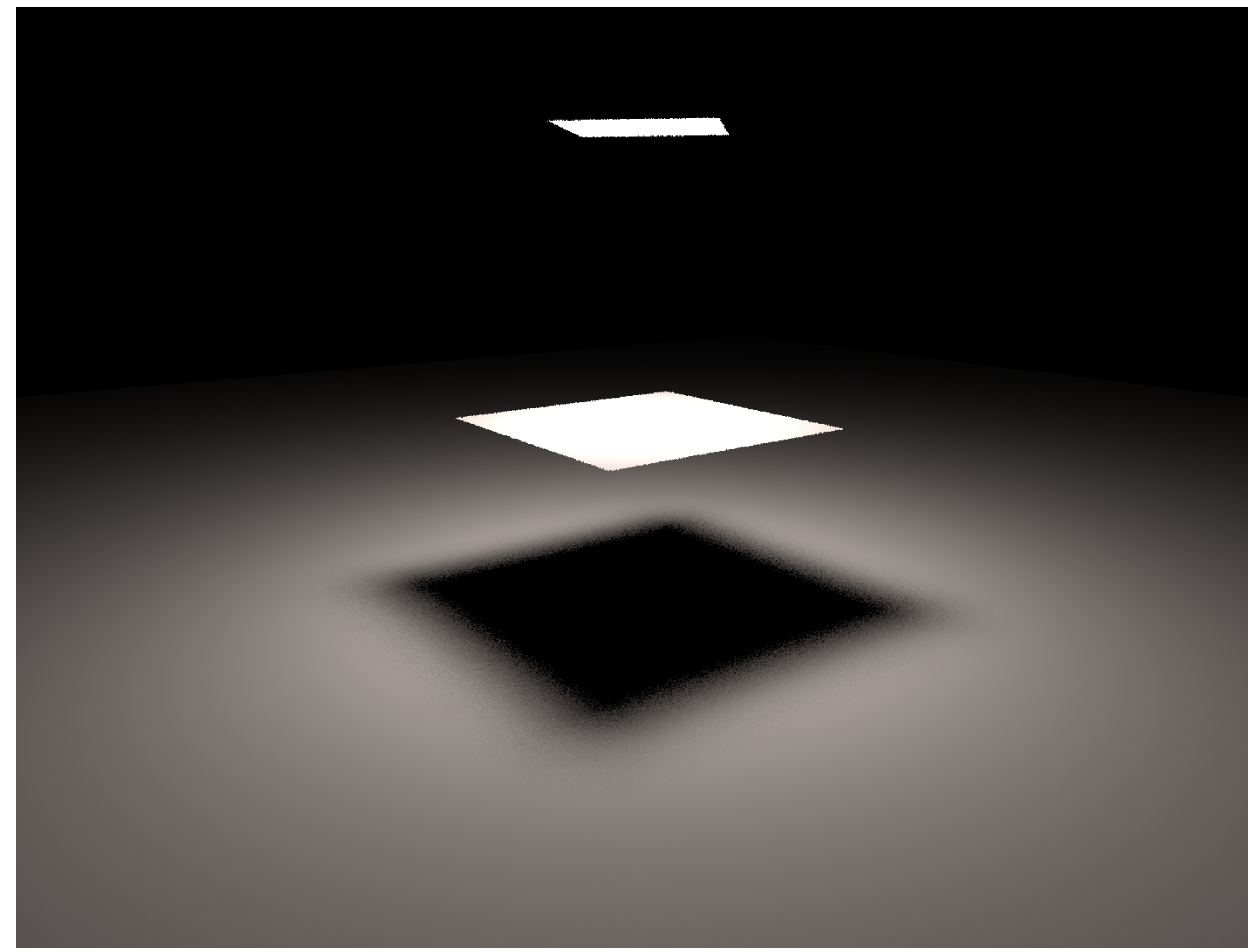
Recap

Quality Improves with More Rays



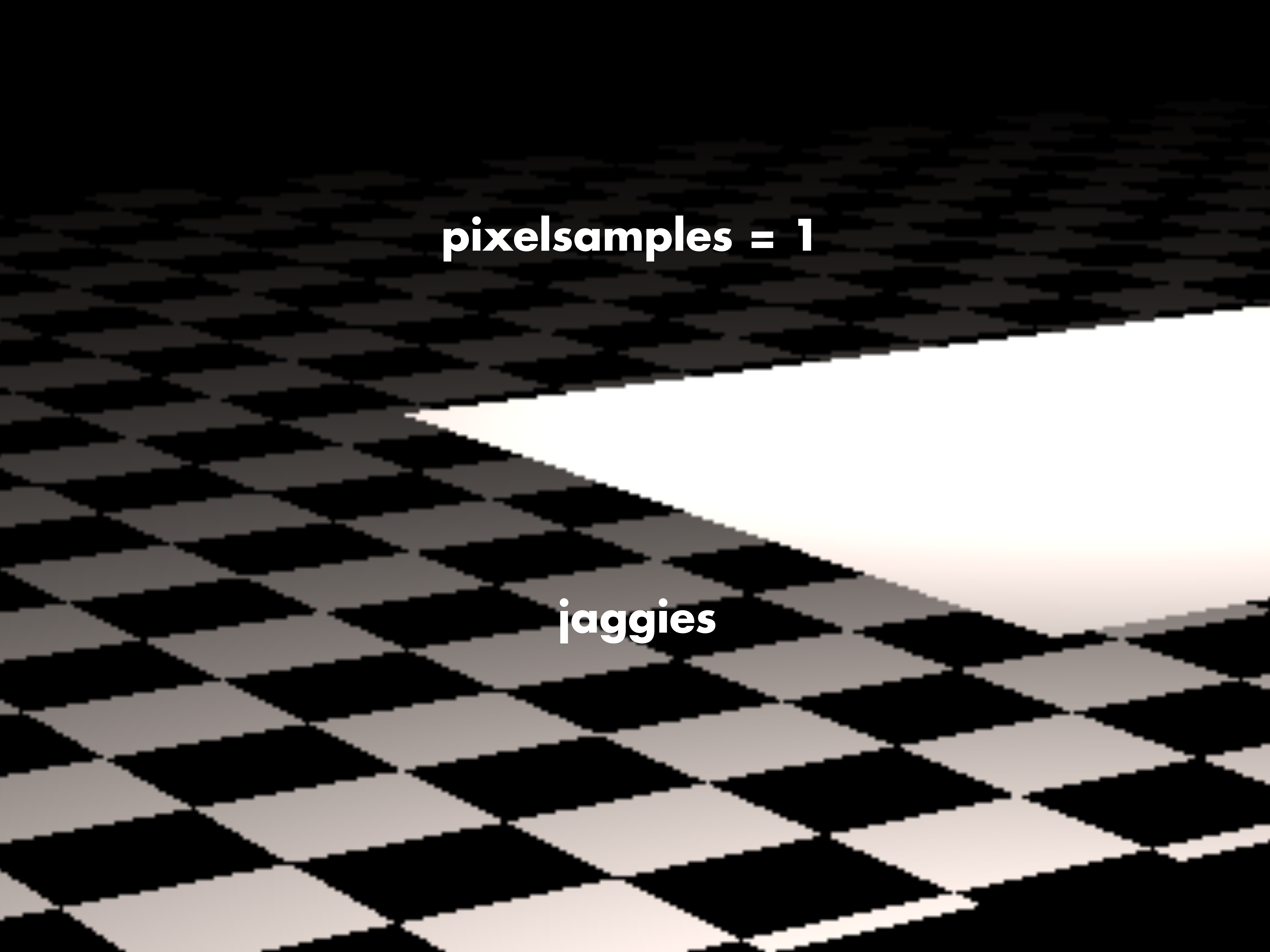
Area

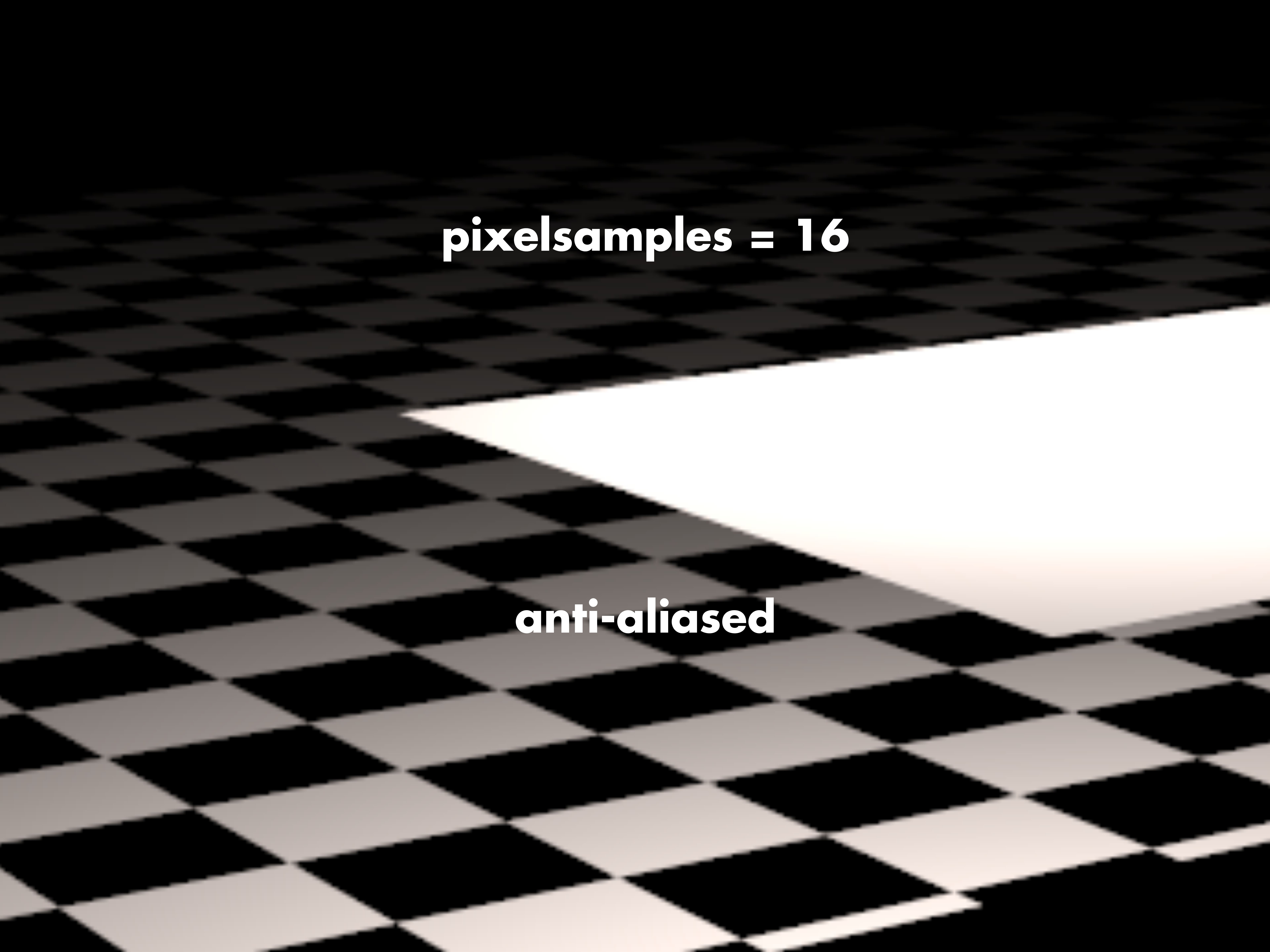
1 shadow ray



Area

16 shadow rays


pixelsamples = 1
jaggies


pixelsamples = 16
anti-aliased

Sampling and Reconstruction

Basic signal processing

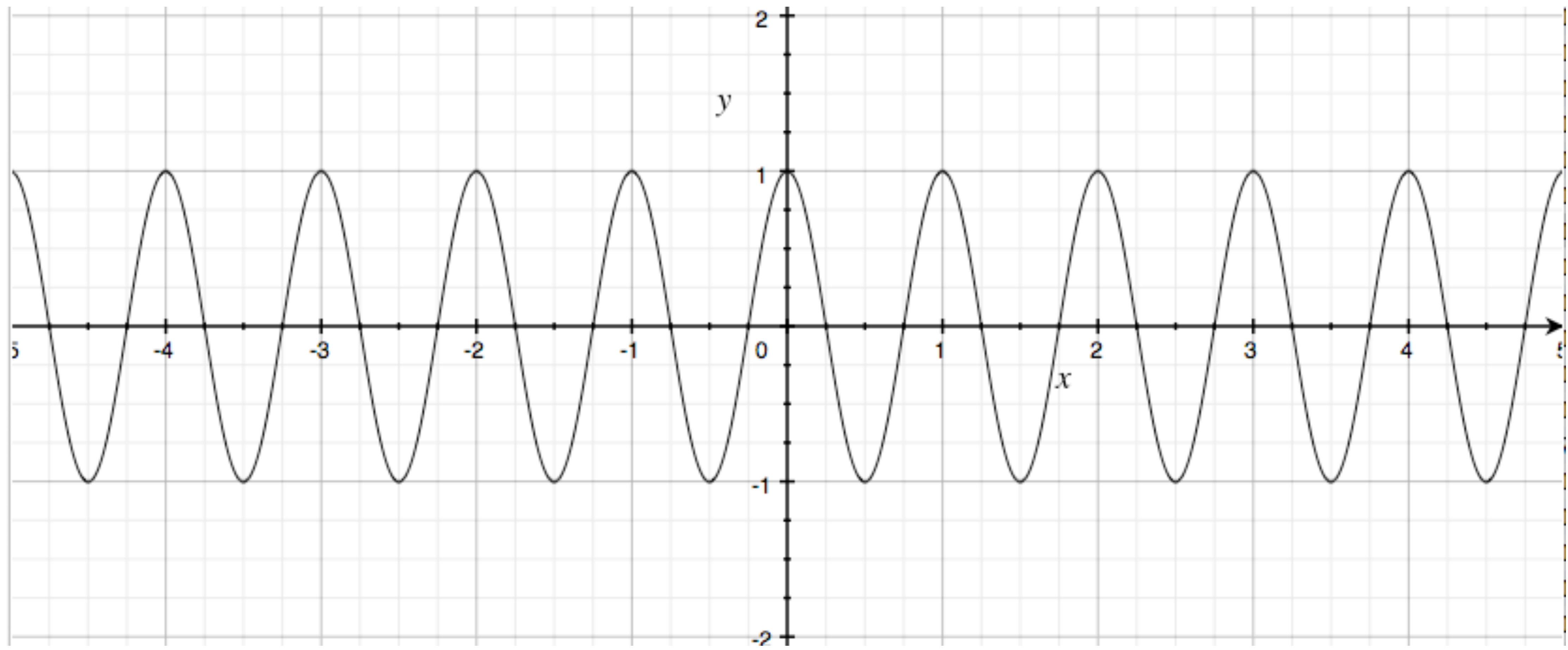
- **Fourier transforms**
- **The convolution theorem**
- **The sampling theorem**

Aliasing and antialiasing

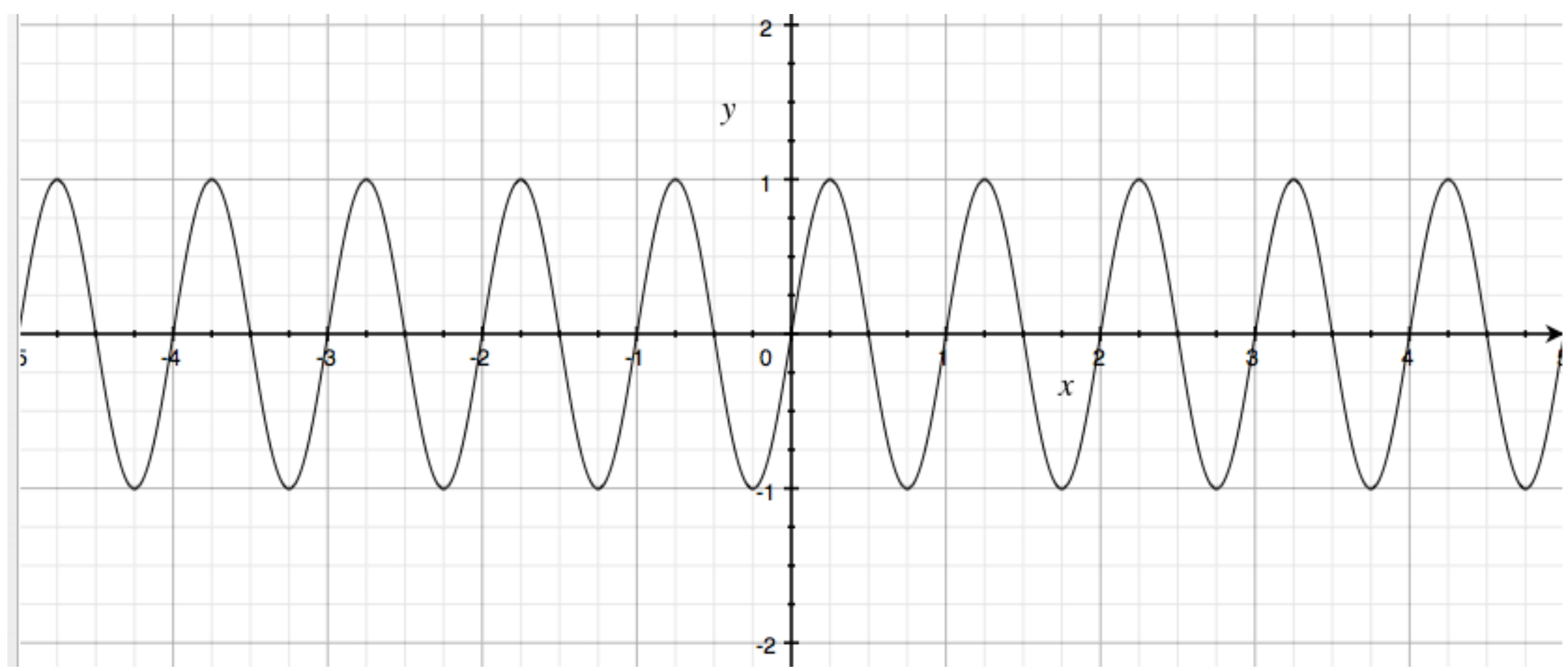
- **Uniform supersampling**
- ***Stochastic sampling***

Review: Basic Signal Processing

Sines and Cosines



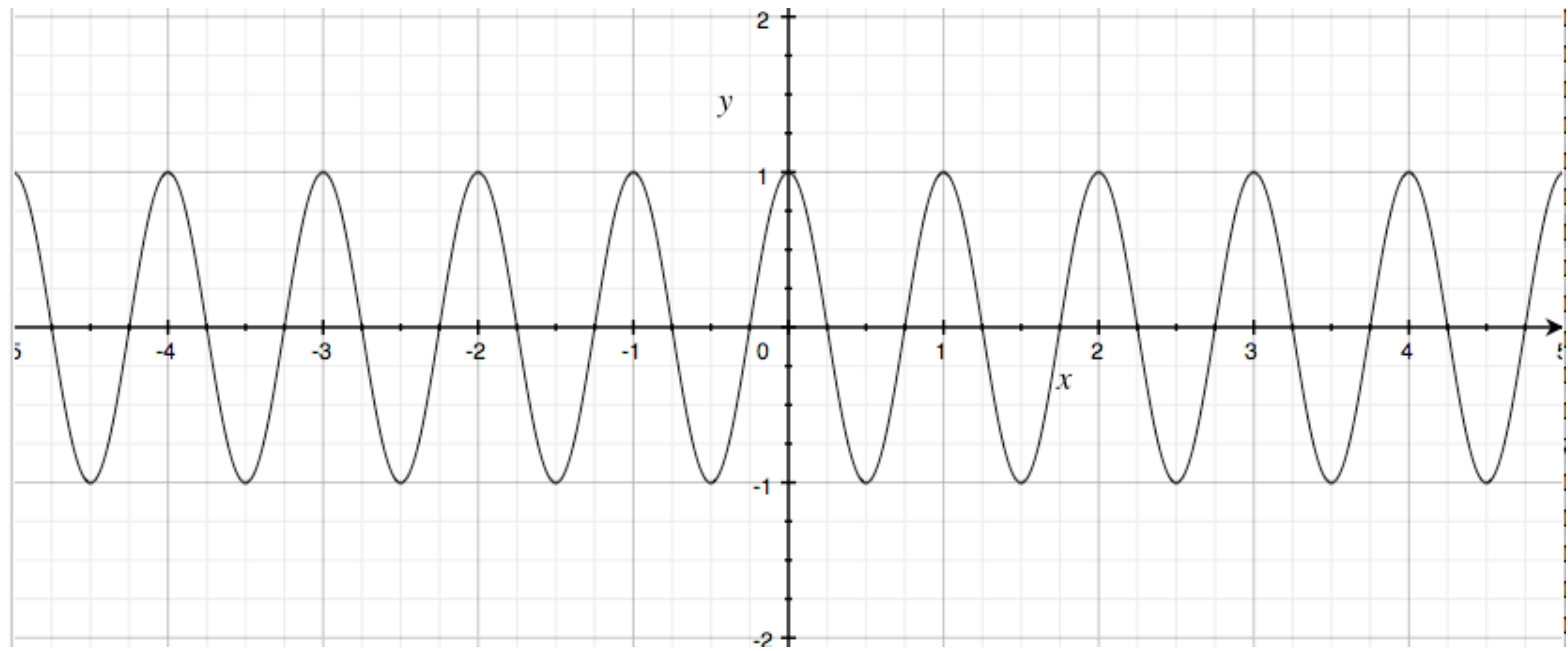
$$\cos 2\pi x$$



$$\sin 2\pi x$$

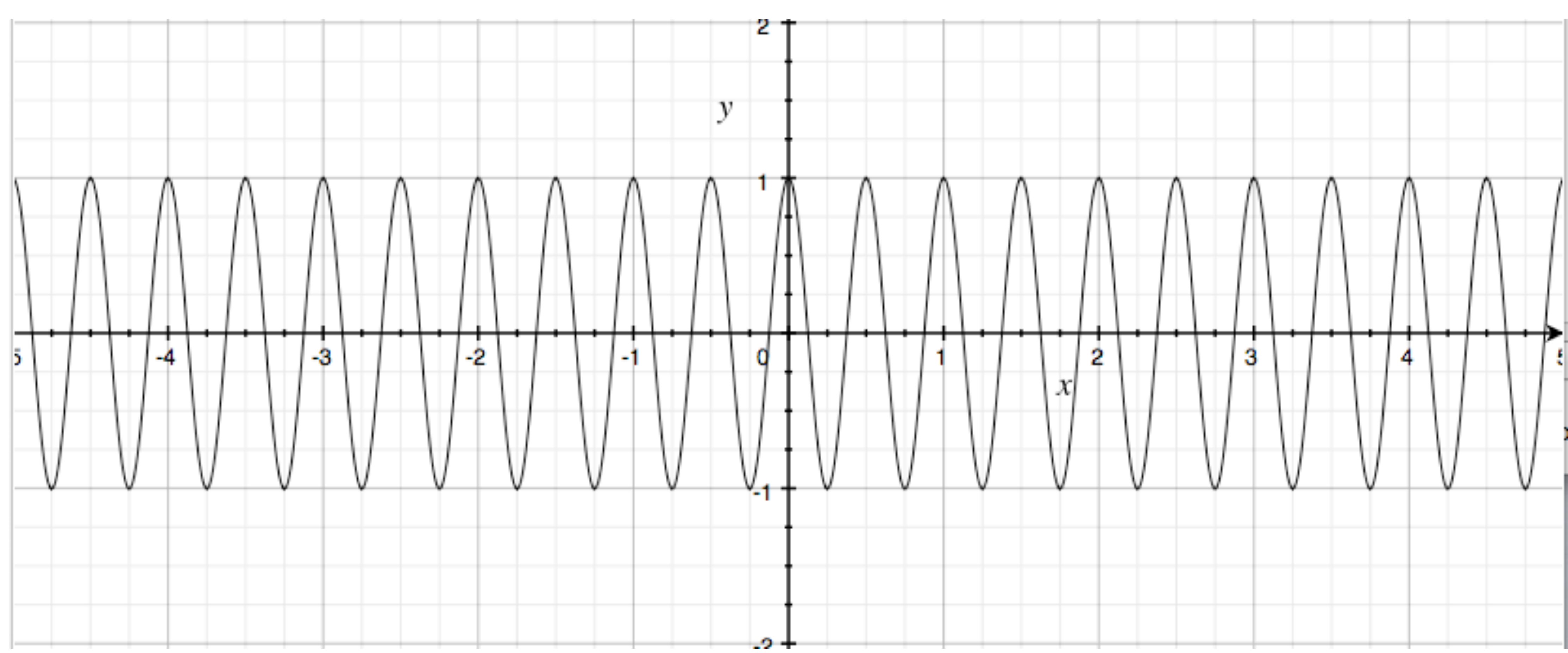
Frequencies $\cos 2\pi f x$

$$f = \frac{1}{T}$$



$$f = 1$$

$$\cos 2\pi x$$

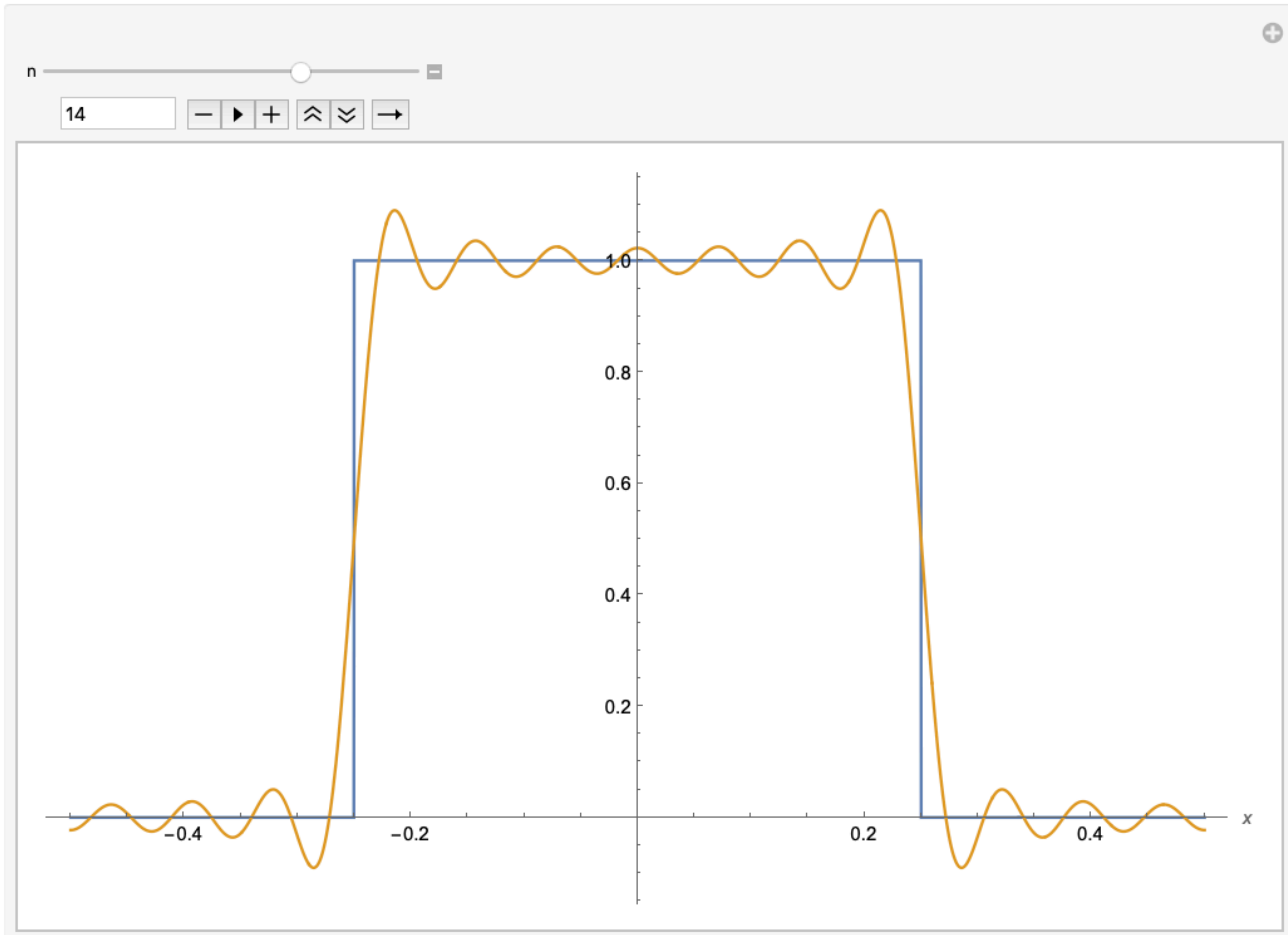


$$f = 2$$

$$\cos 4\pi x$$

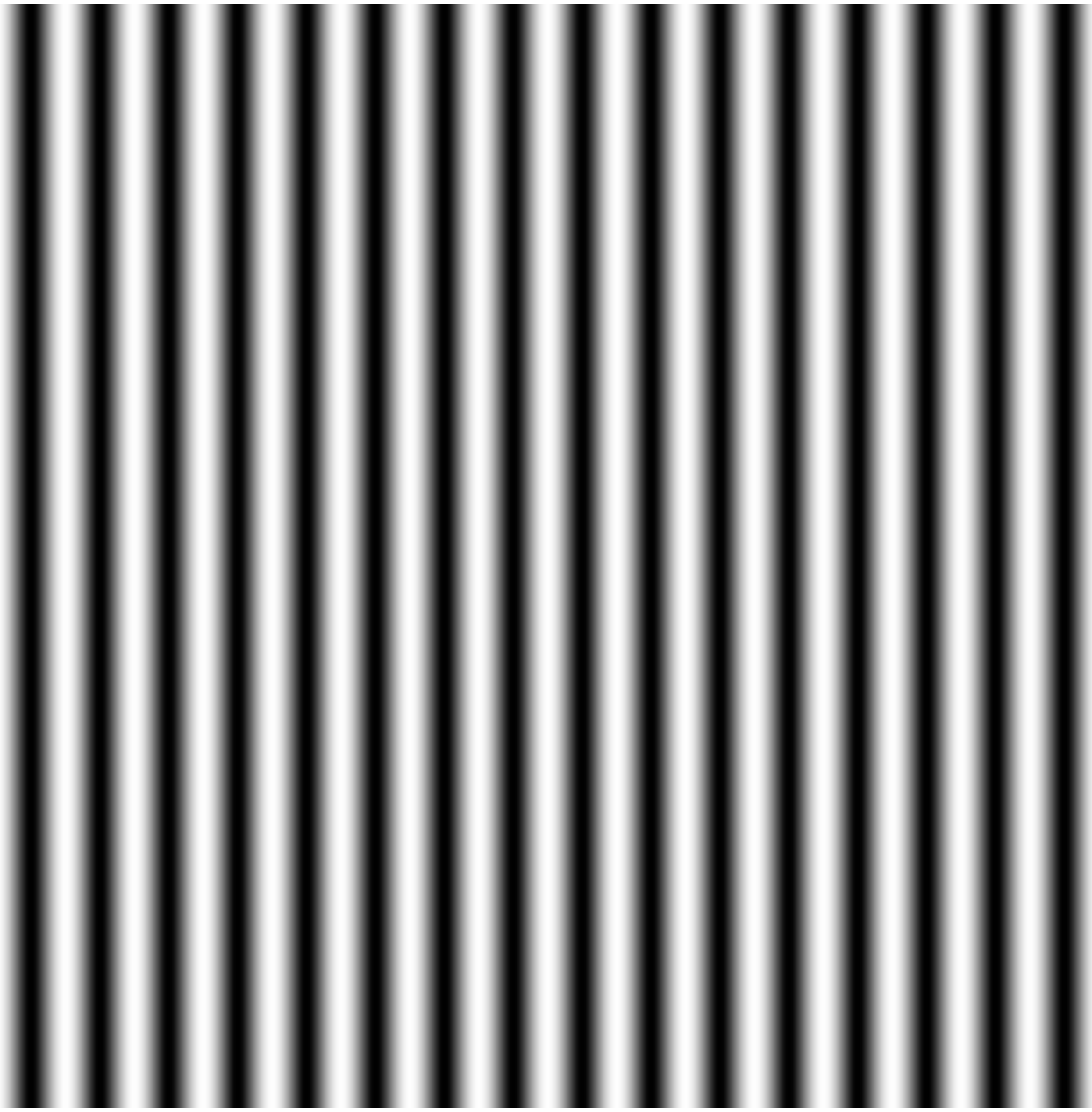
Mathematica Demo
Fourier Series Approximations

The n^{th}-order Fourier series of $f(t)$ is by default defined to be $\sum_{k=-n}^{n} c_k \, e^{i k t}$ with $c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\, e^{-i k t}\, dt$.

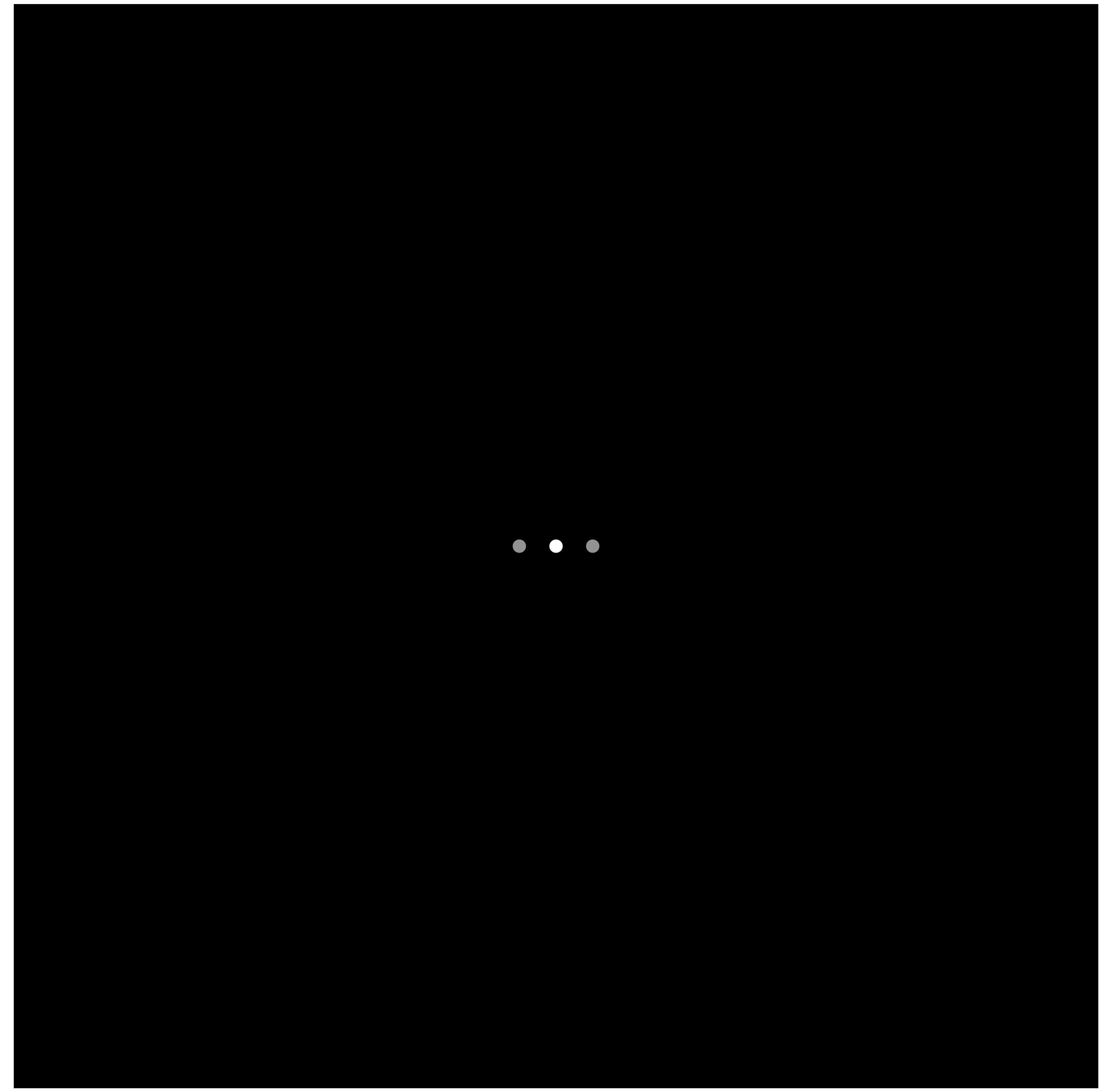


NOTE: Mathematica is freely available to Stanford students.
Go to https://www.wolfram.com/siteinfo/

$\sin(2\pi/32)x$ **32 pixels per cycle**

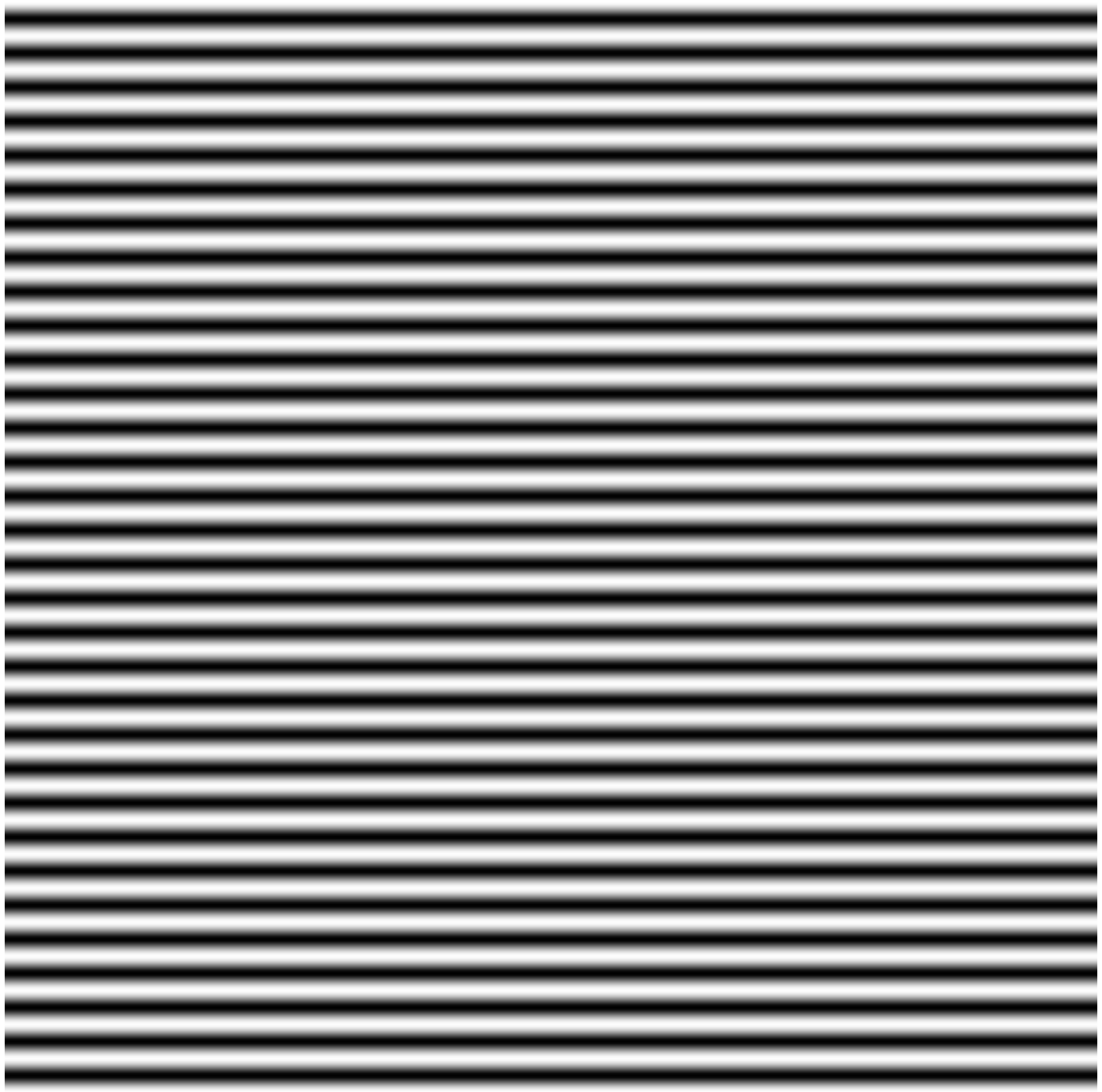


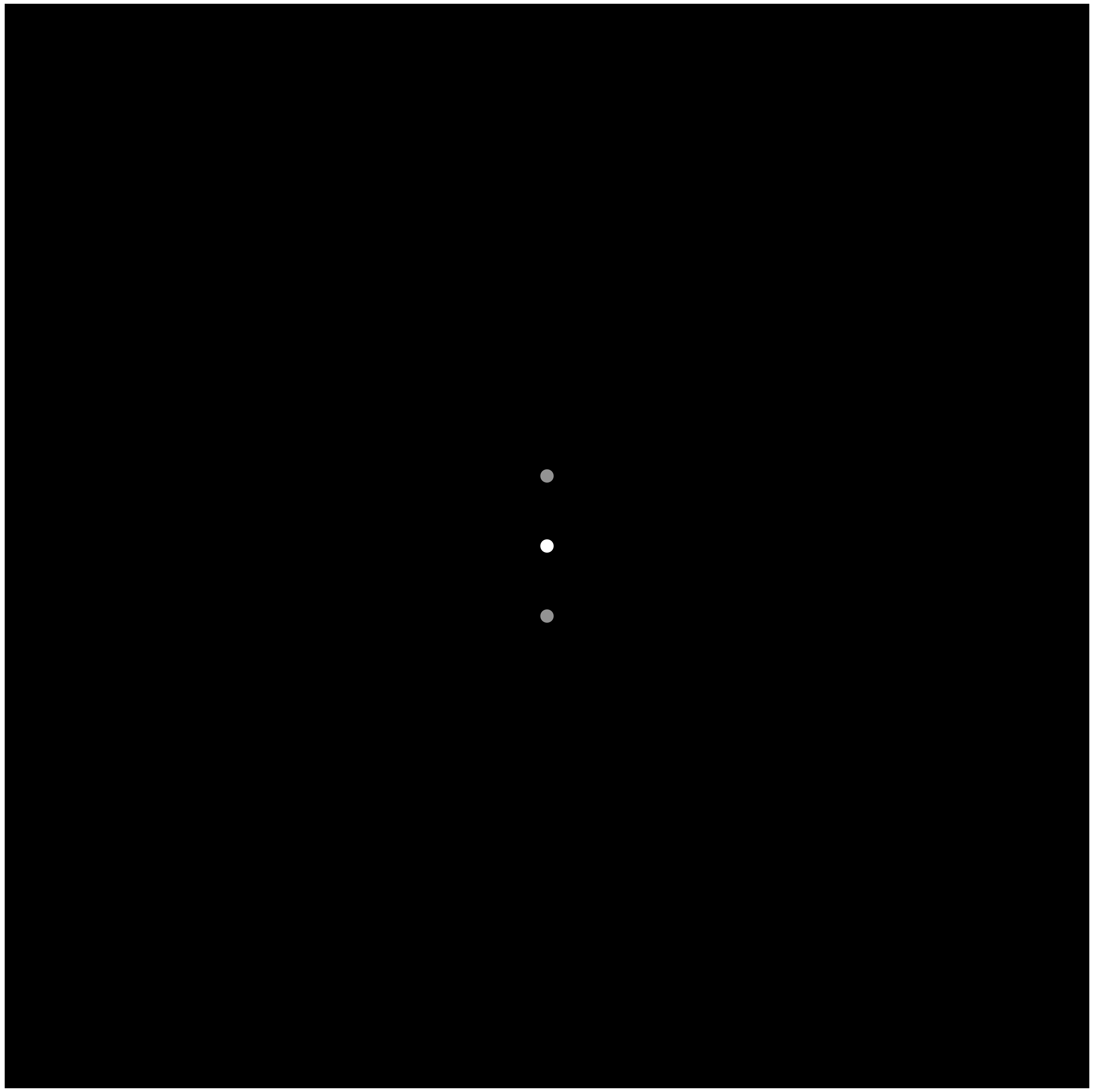
Spatial Domain



Frequency Domain

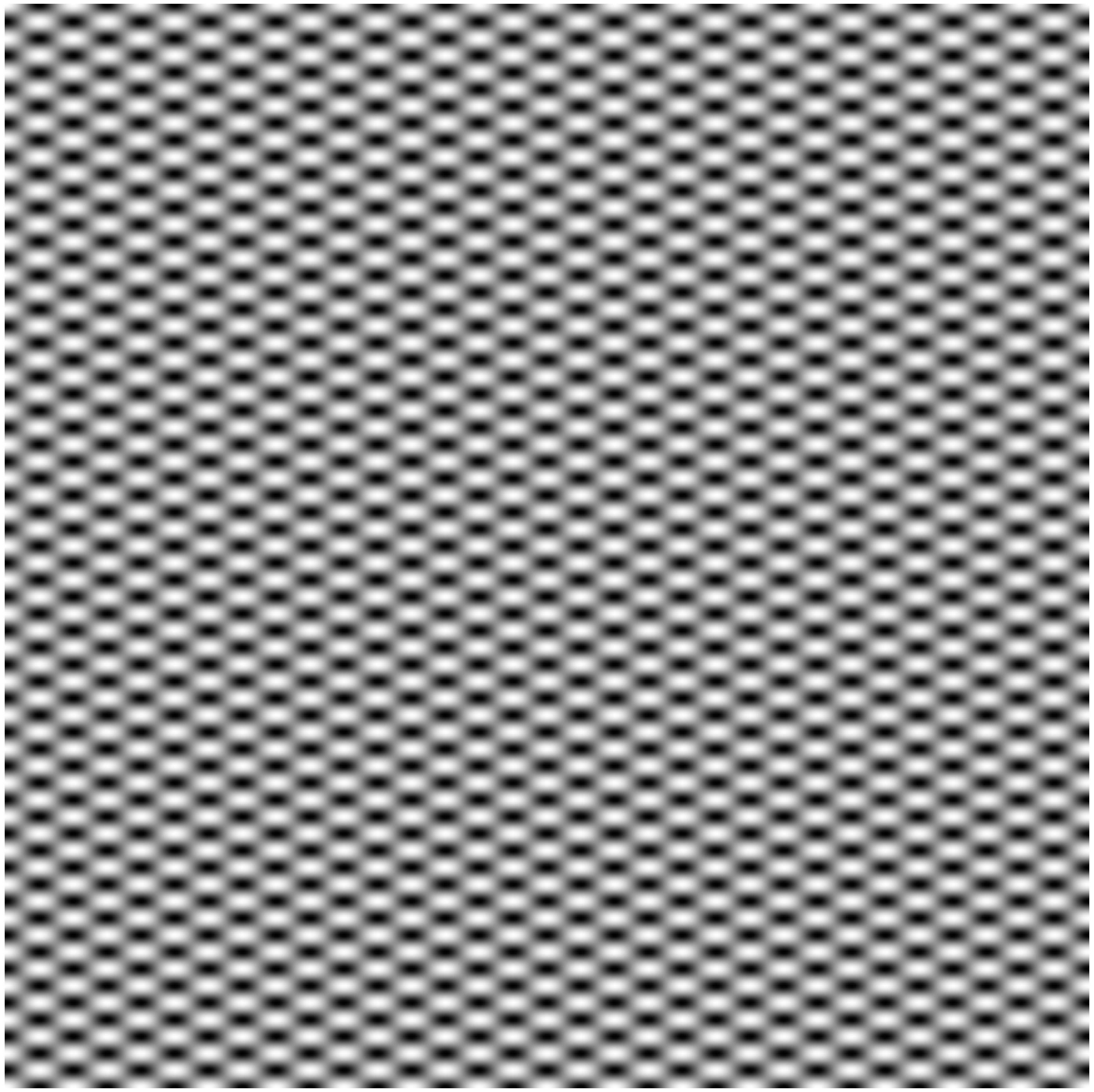
$\sin(2\pi/16)y$ **16 pixels per cycle**



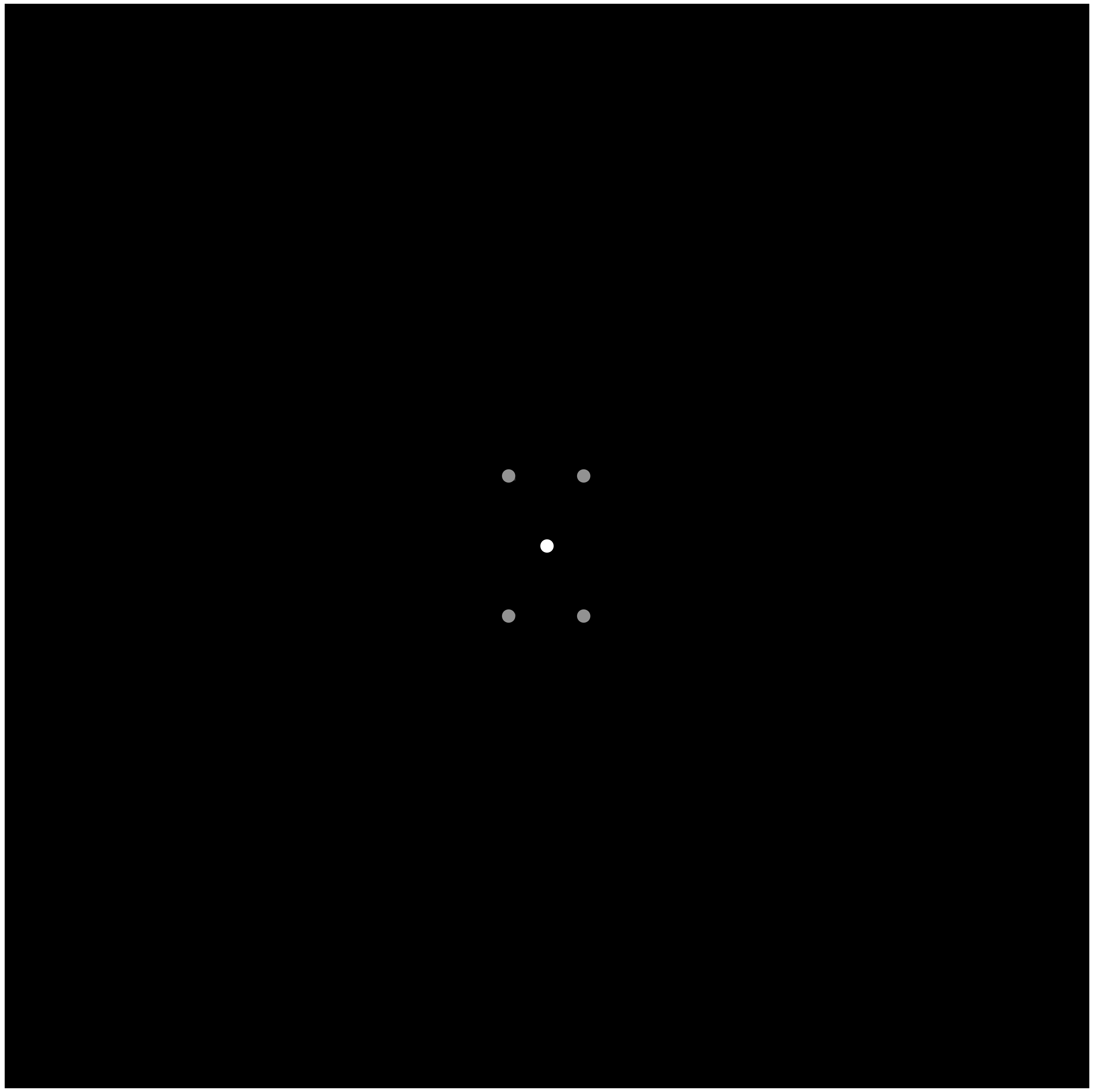


Spatial Domain **Frequency Domain**

$\sin(2\pi/32)x \times \sin(2\pi/16)y$

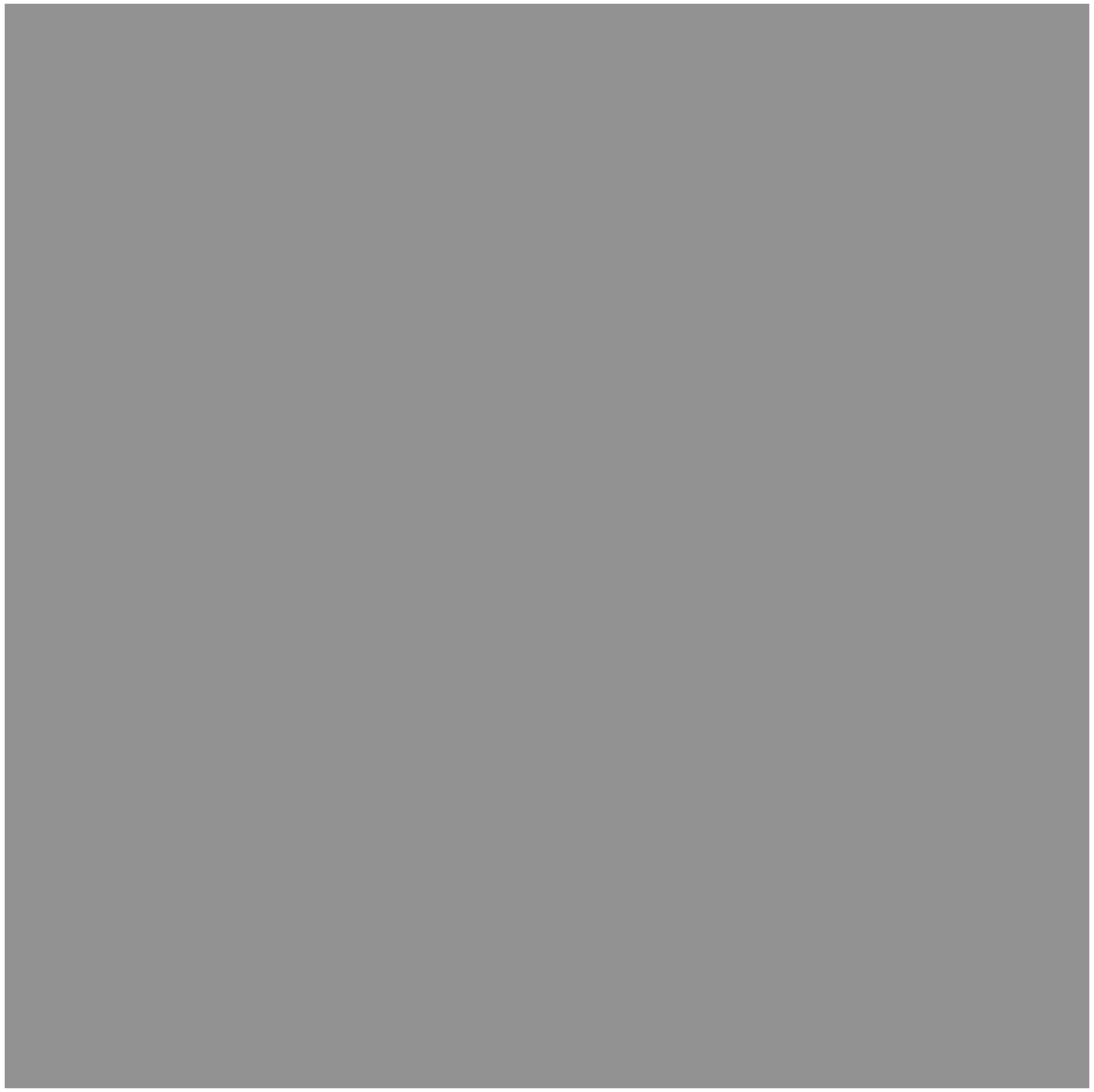


Spatial Domain

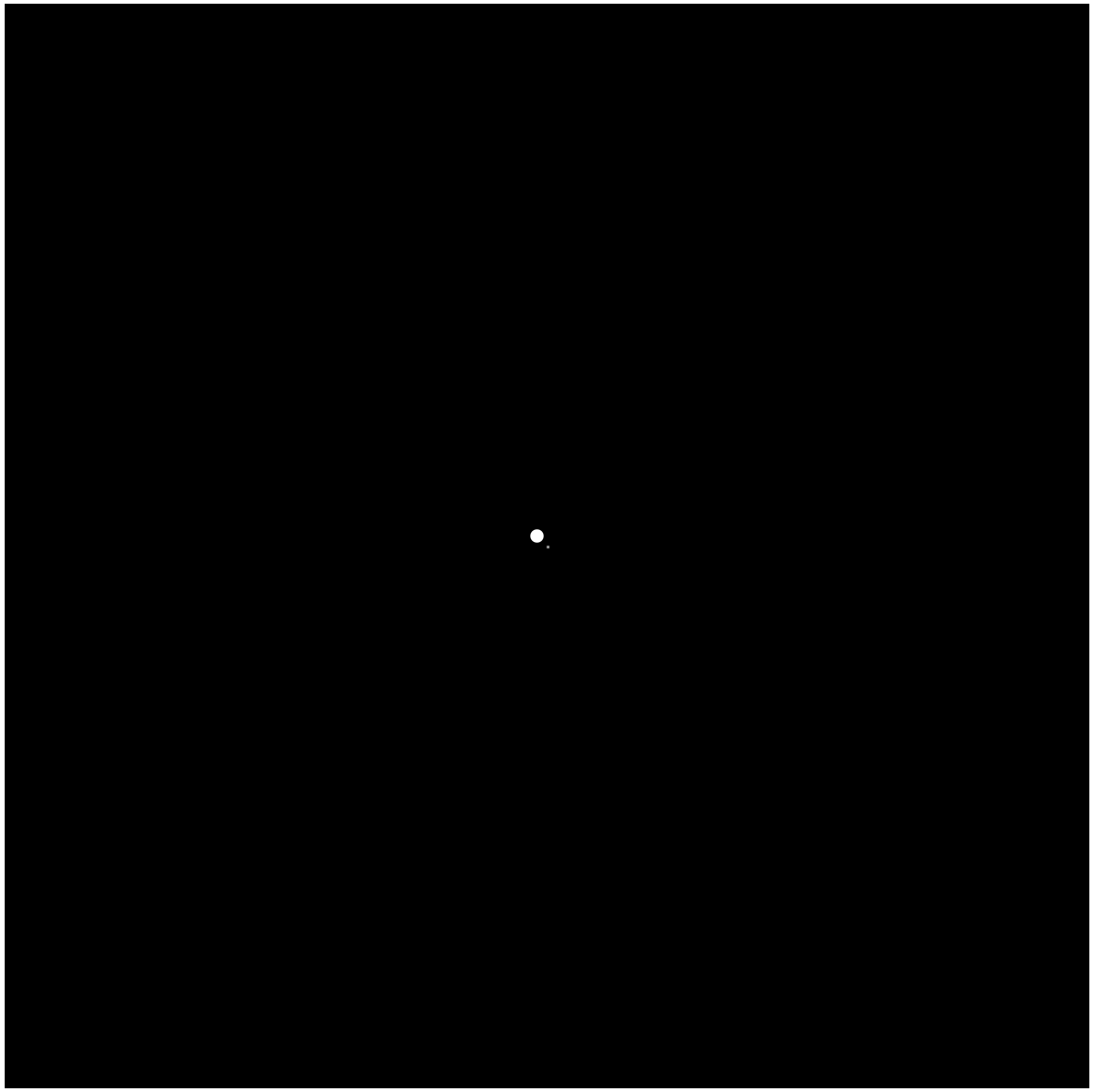


Frequency Domain

Constant

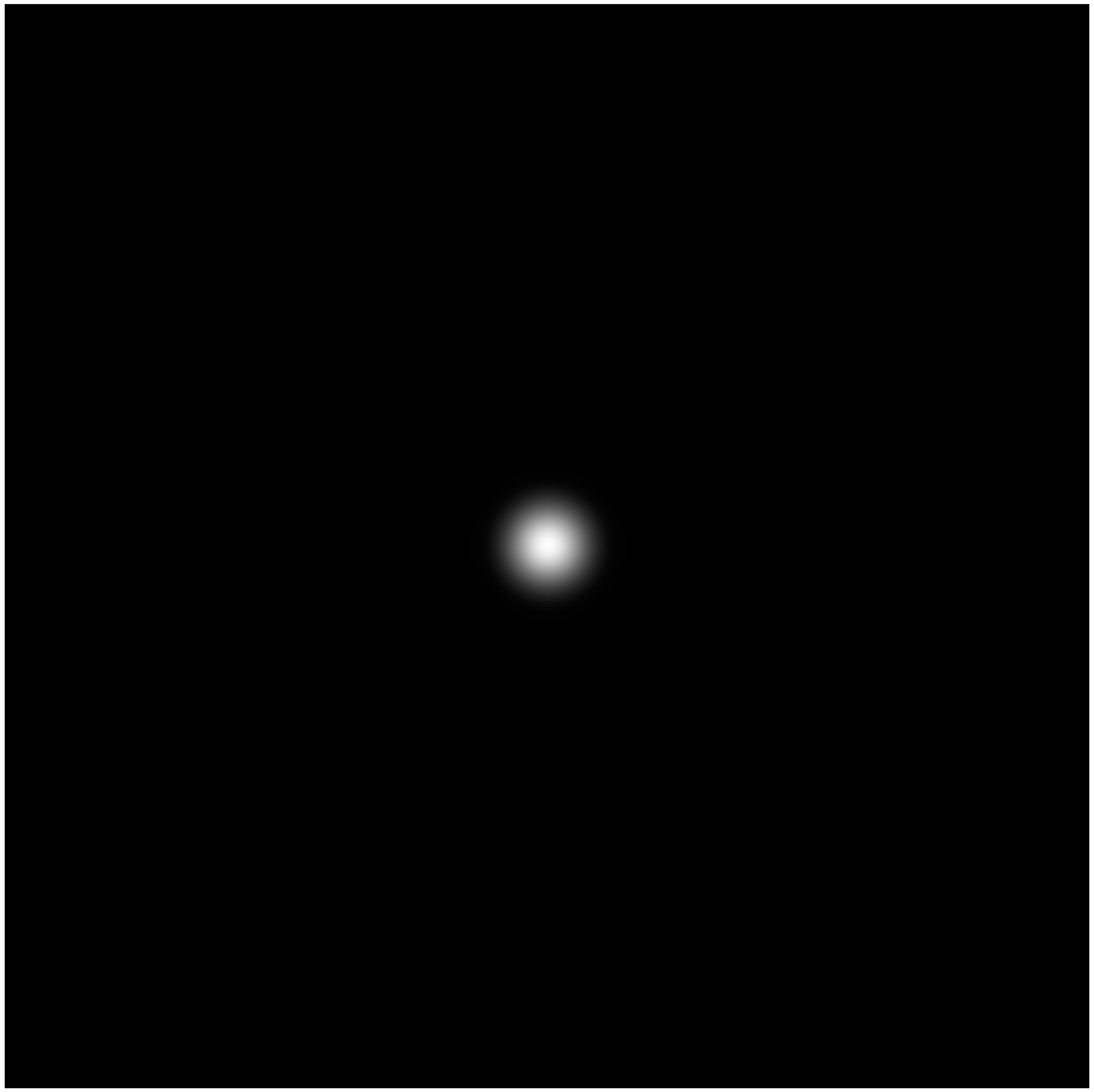


Spatial Domain

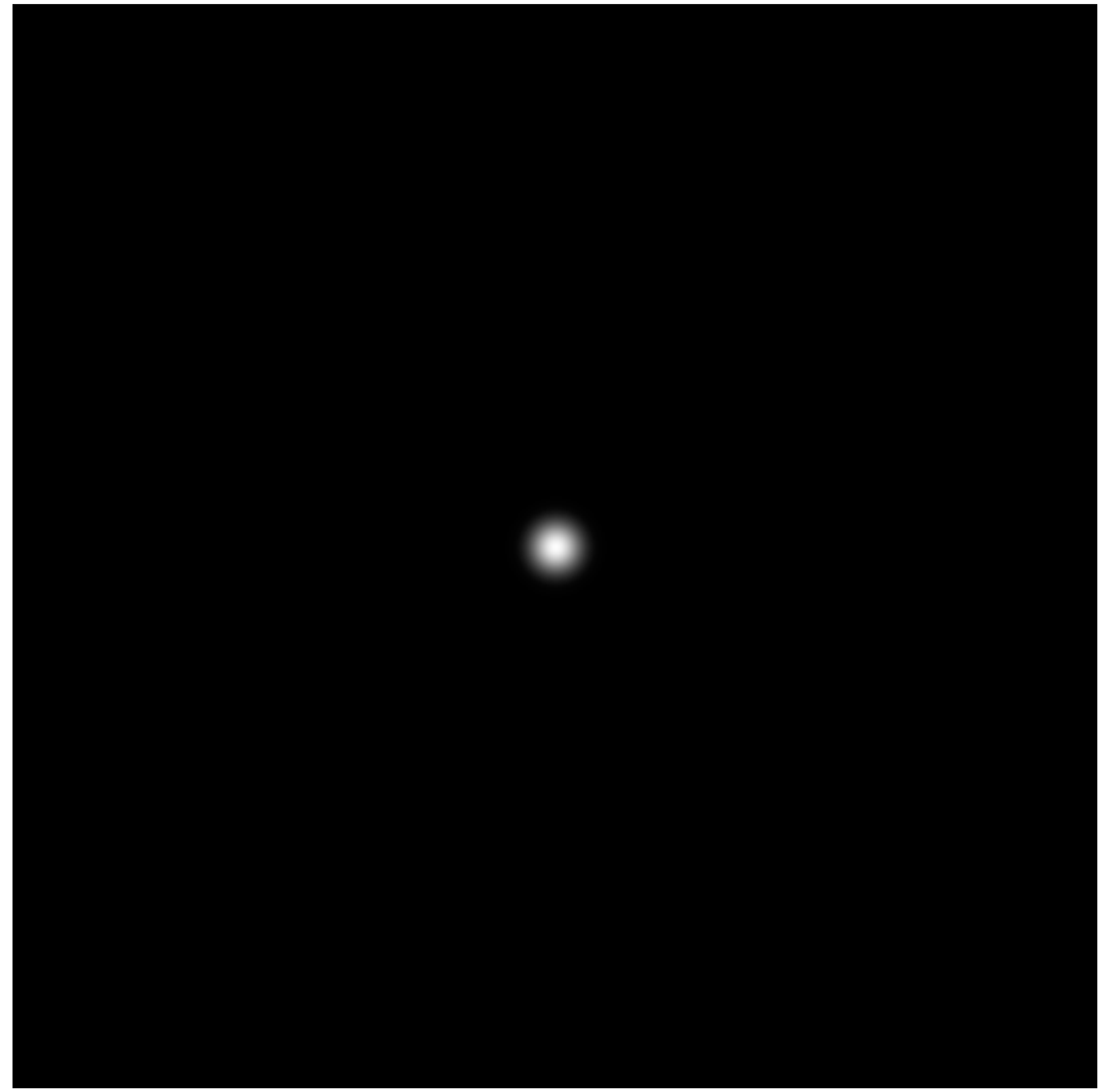


Frequency Domain

$$e^{-r^2/16^2}$$

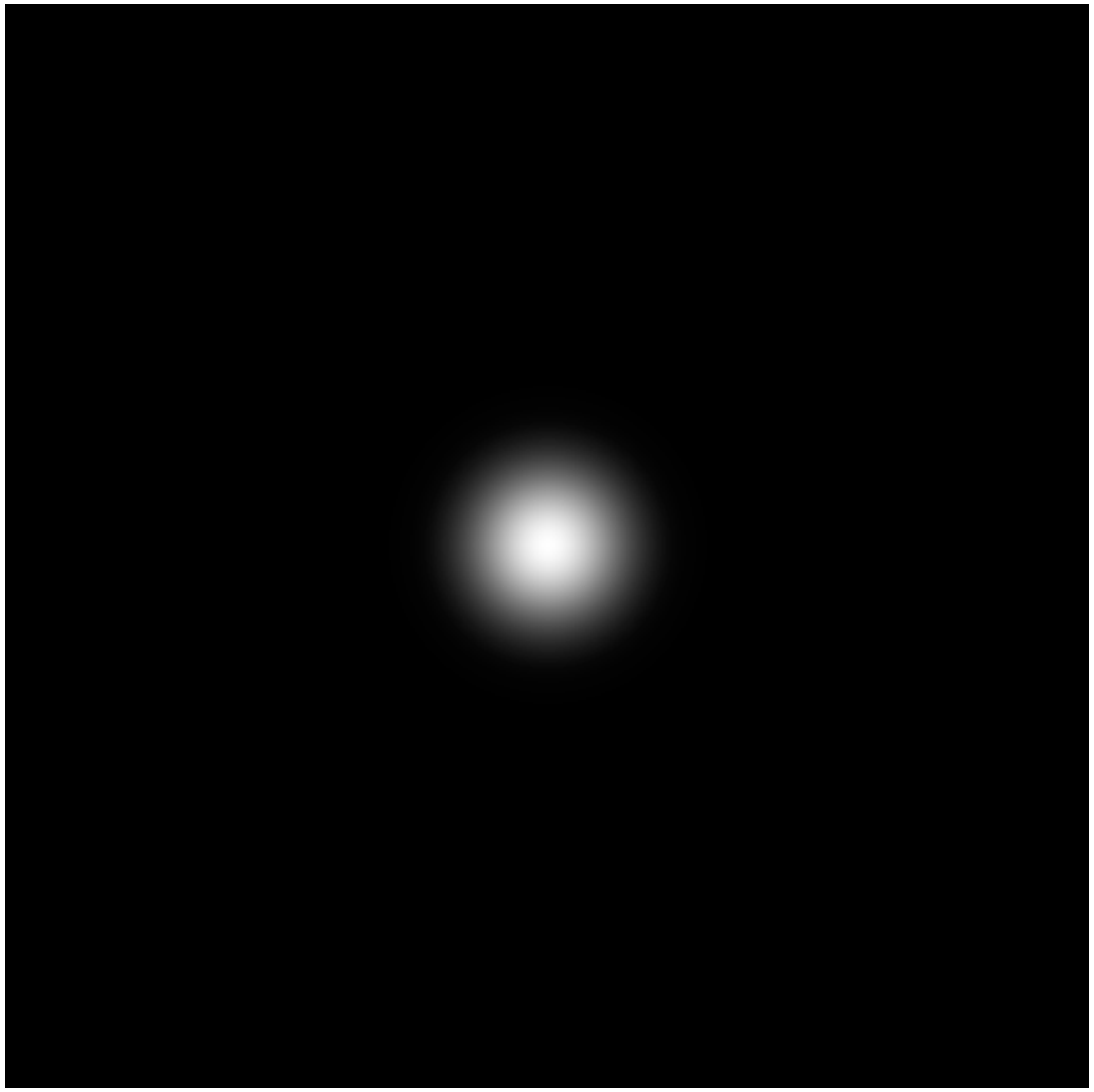


Spatial Domain

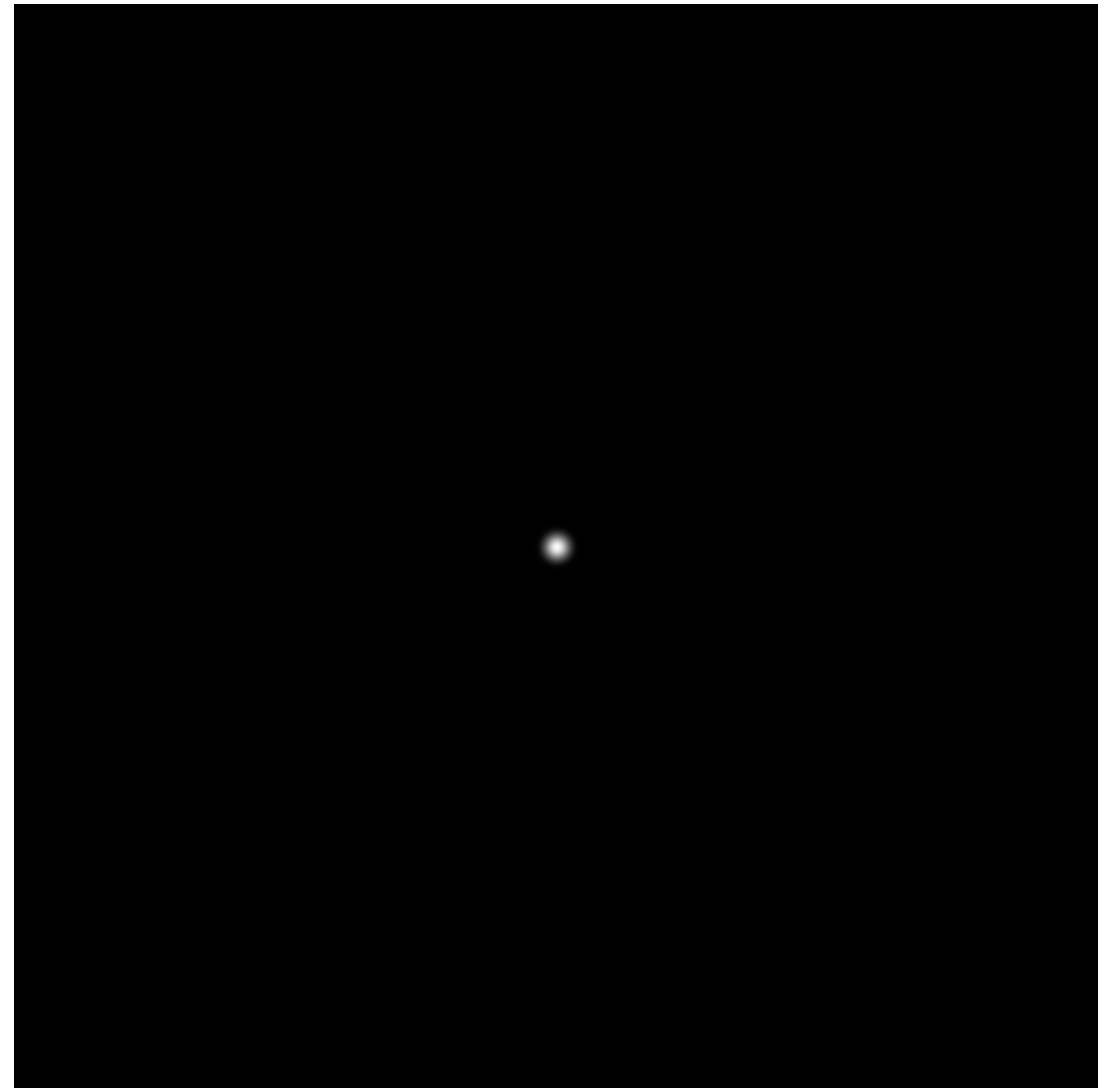


Frequency Domain

$e^{-r^2/32^2}$

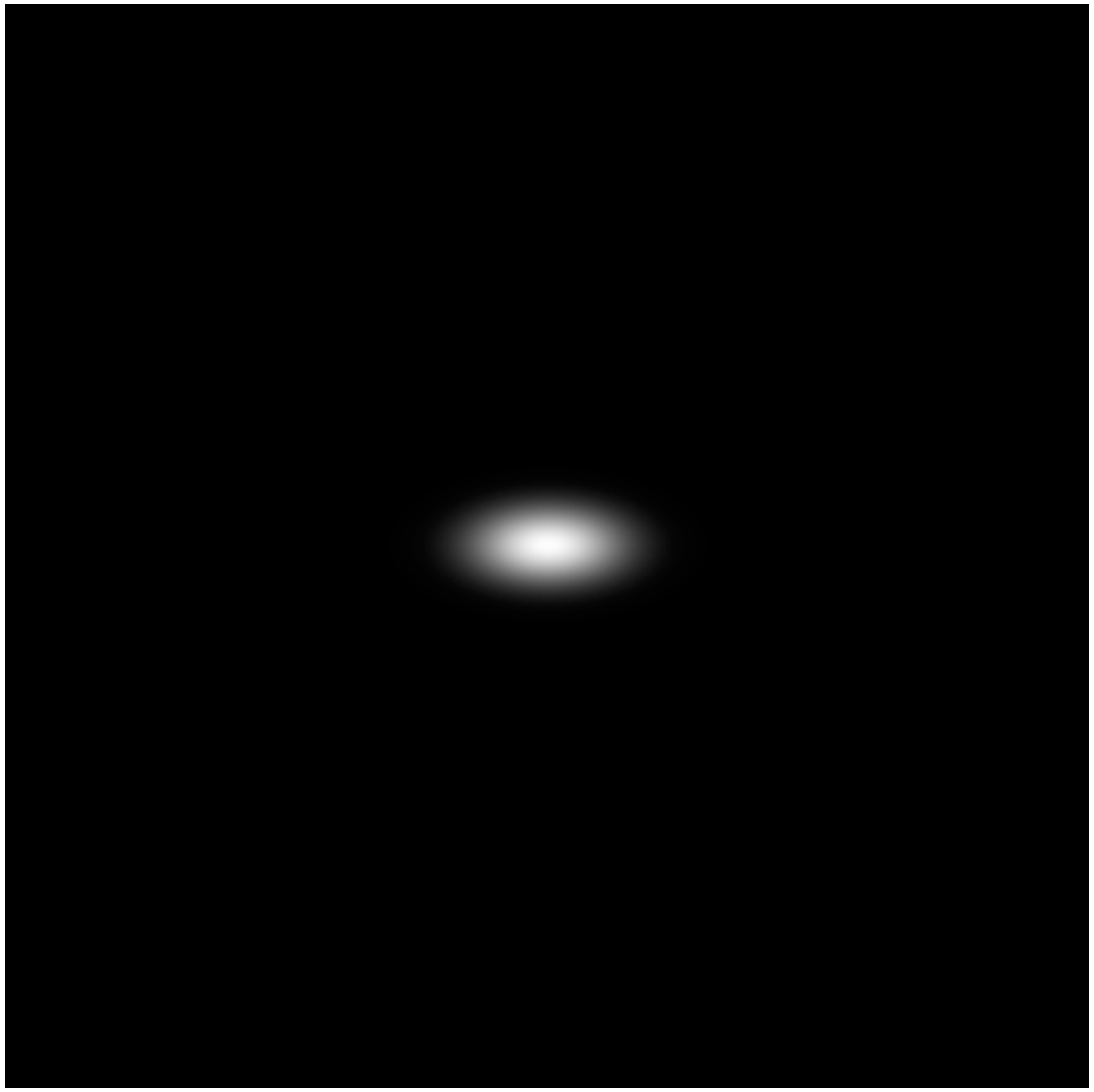


Spatial Domain

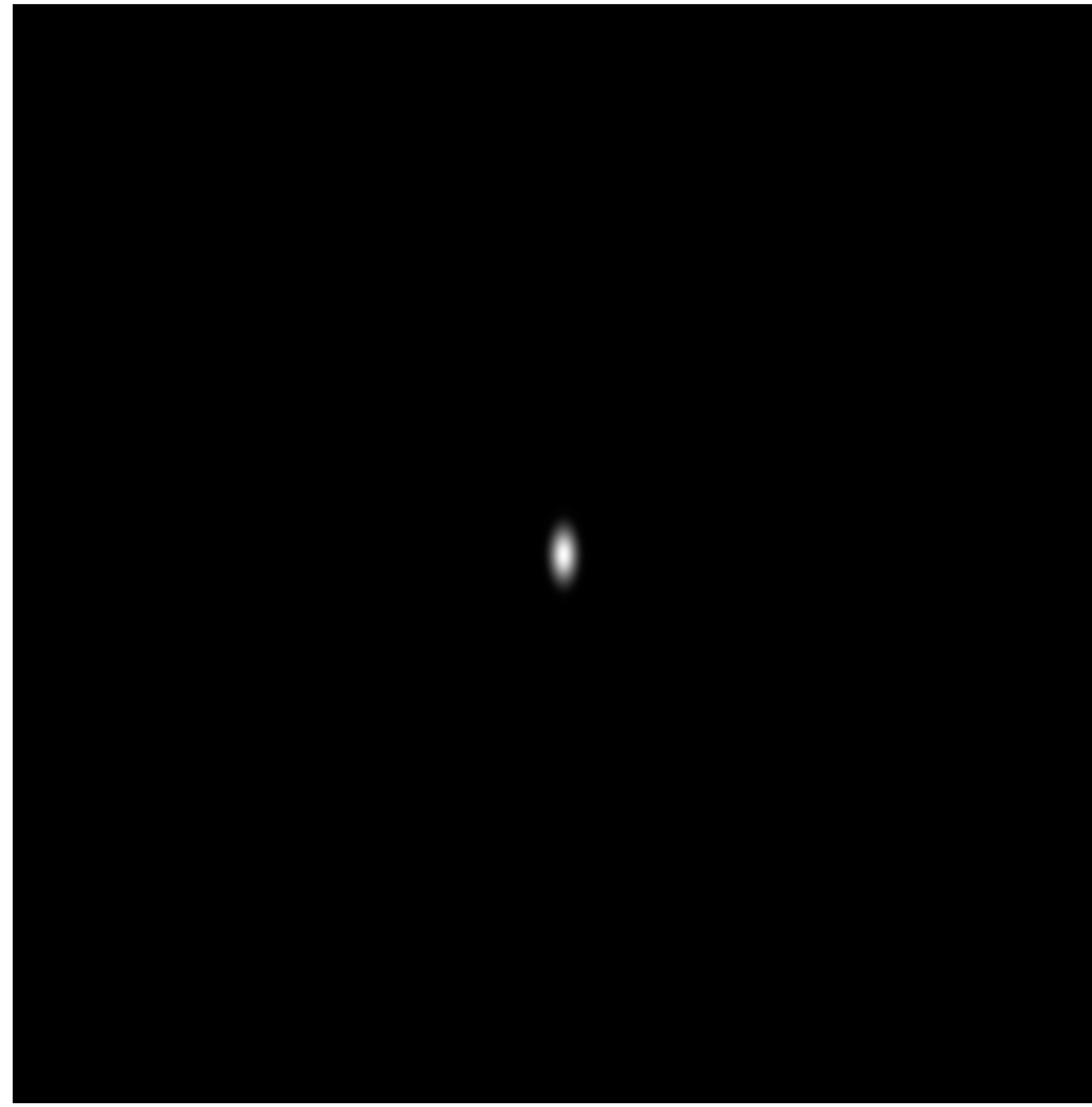


Frequency Domain

$$e^{-x^2/32^2} \times e^{-y^2/16^2}$$

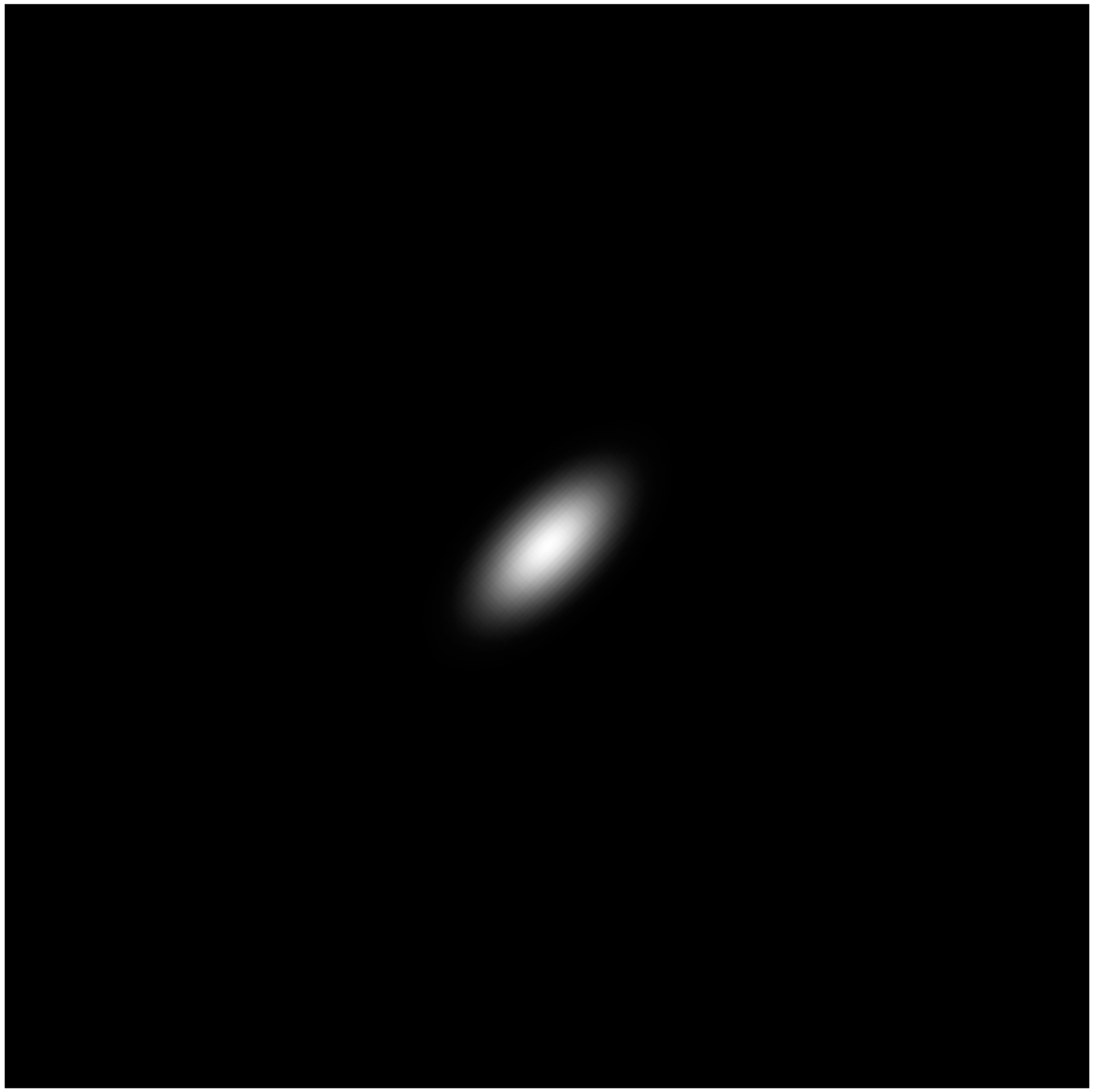


Spatial Domain

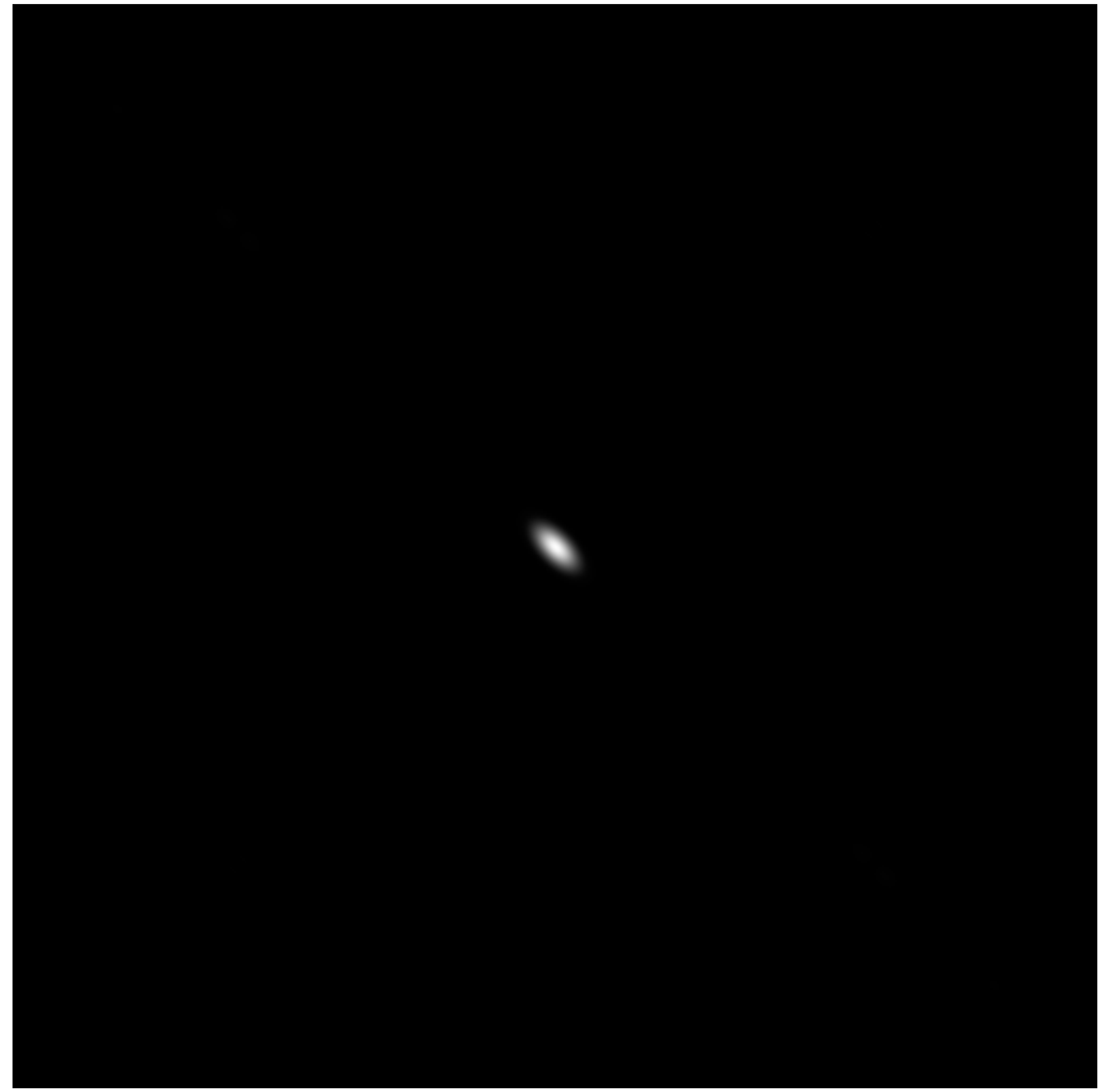


Frequency Domain

Rotate 45 $e^{-x^2/32^2} \times e^{-y^2/16^2}$



Spatial Domain



Frequency Domain

Fourier Transforms

The Fourier transform converts between the spatial and frequency domain

Spatial Domain

$f(x)$

$$\Longrightarrow F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx \Longrightarrow$$

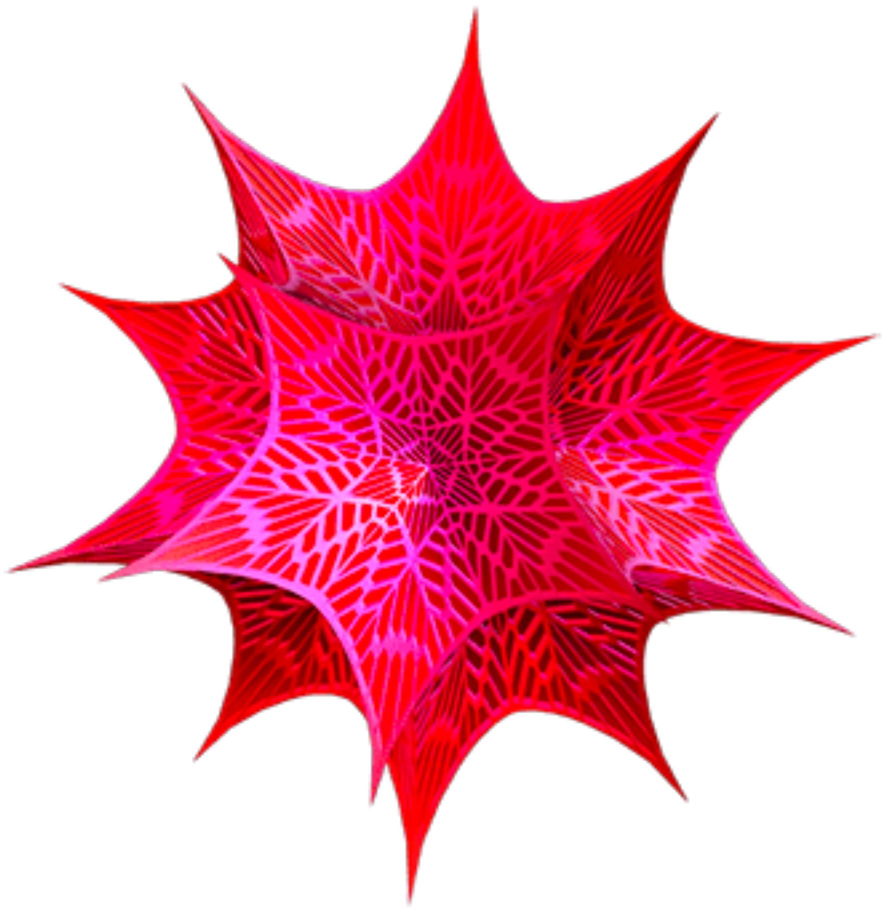
Frequency Domain

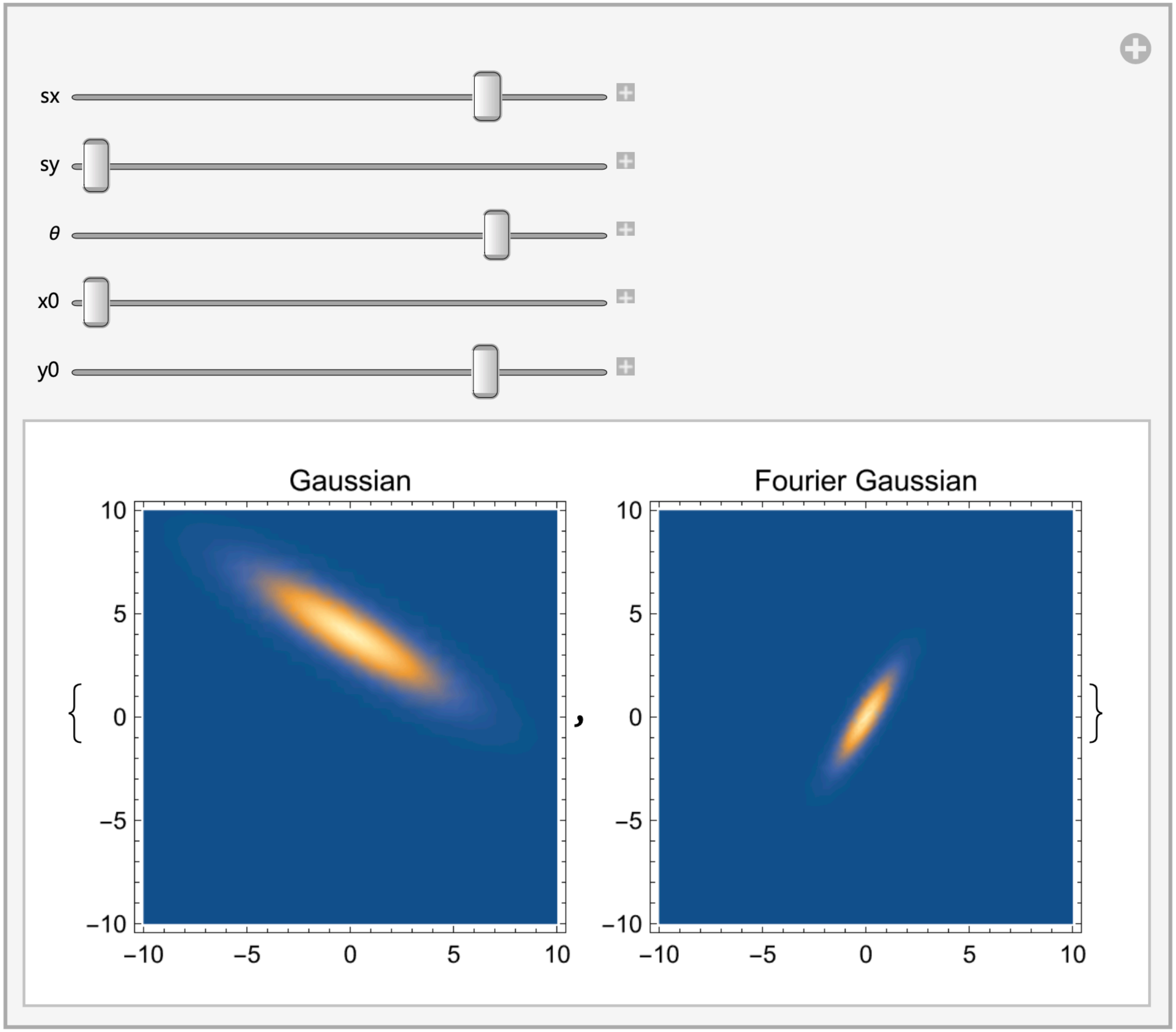
$F(\omega)$

$$\Longleftarrow f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x}\, d\omega \Longleftarrow$$

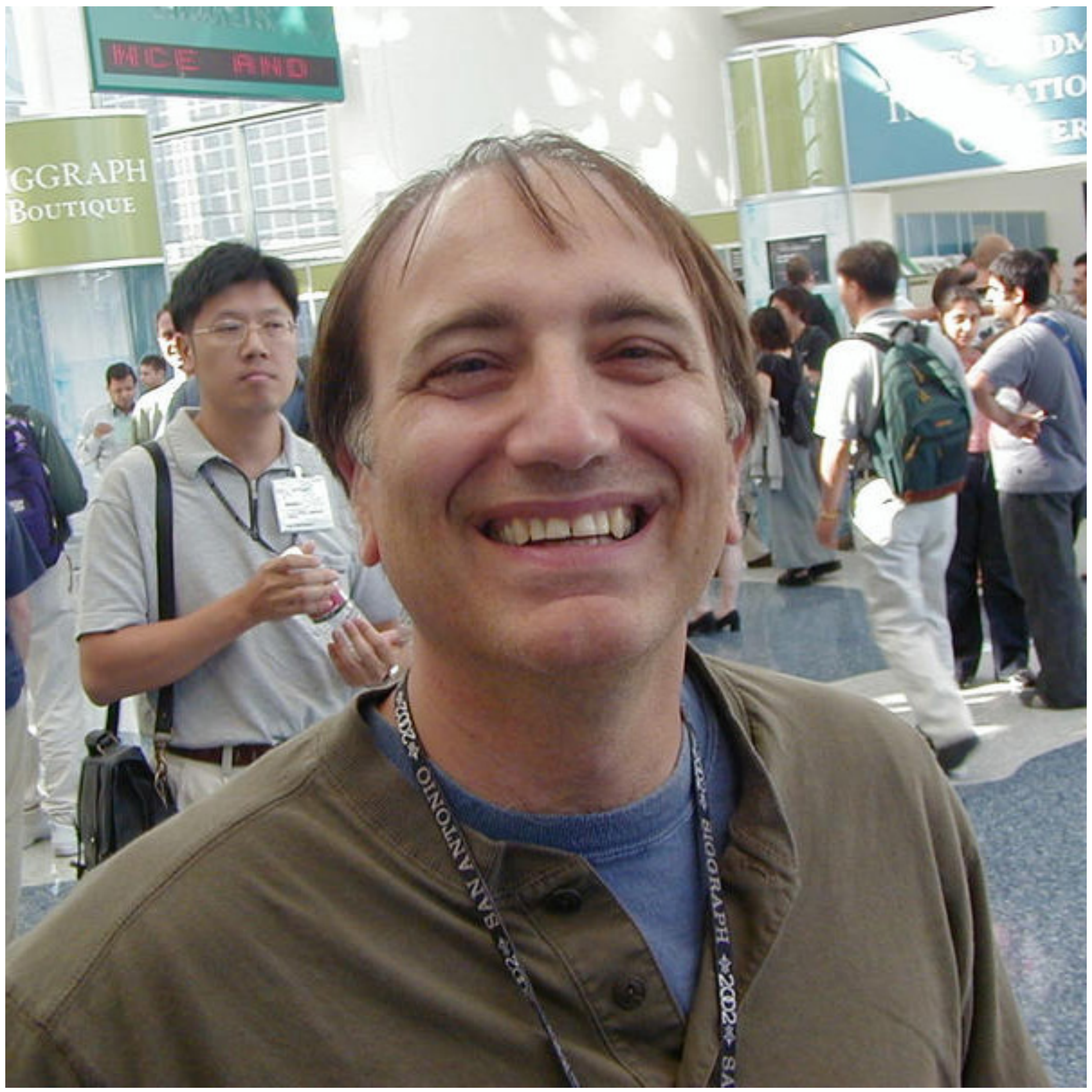
Figures generated using `fft2d.py`

Mathematica Demos (Waves, Gaussians)

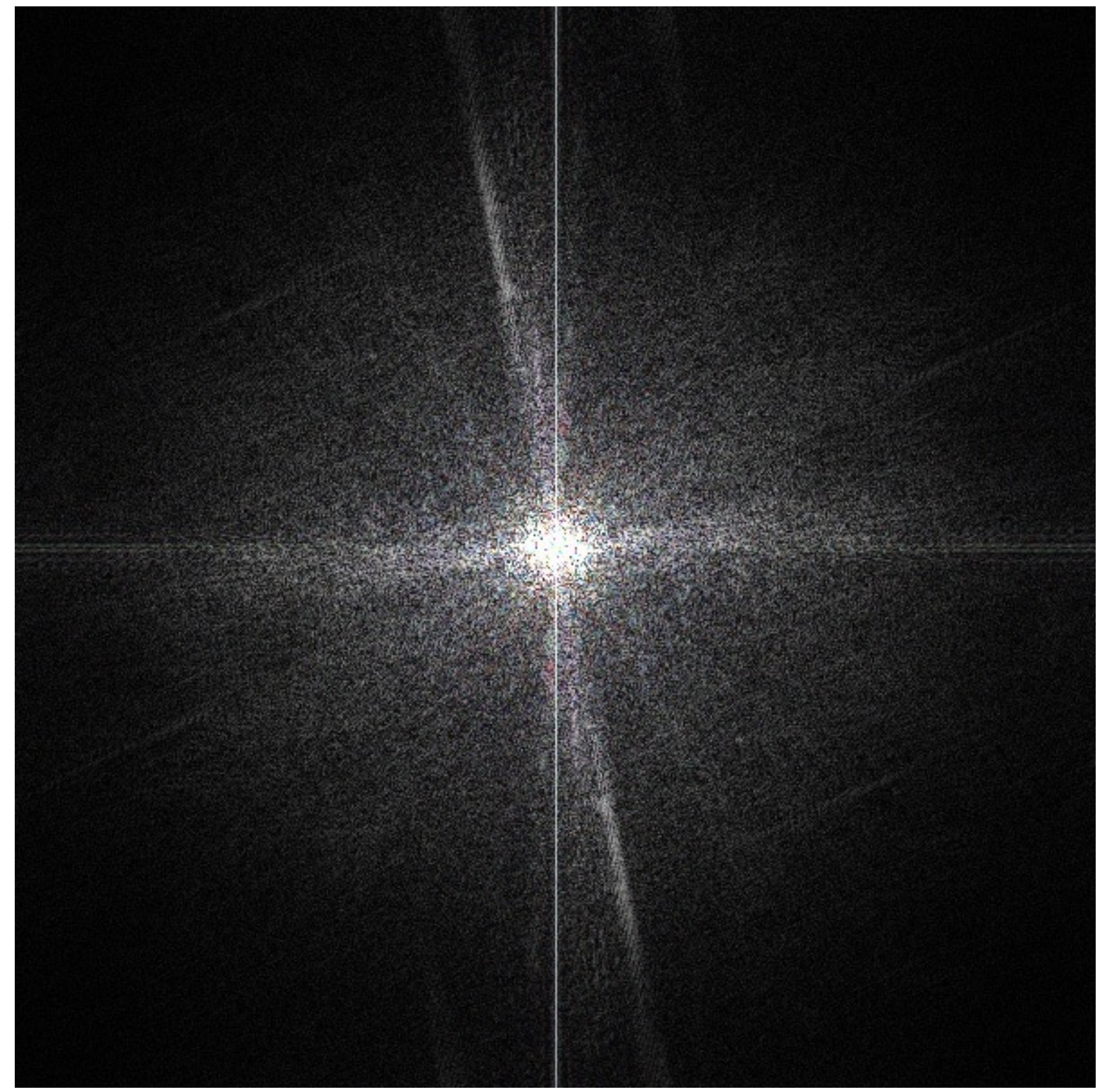




Pat's Frequencies



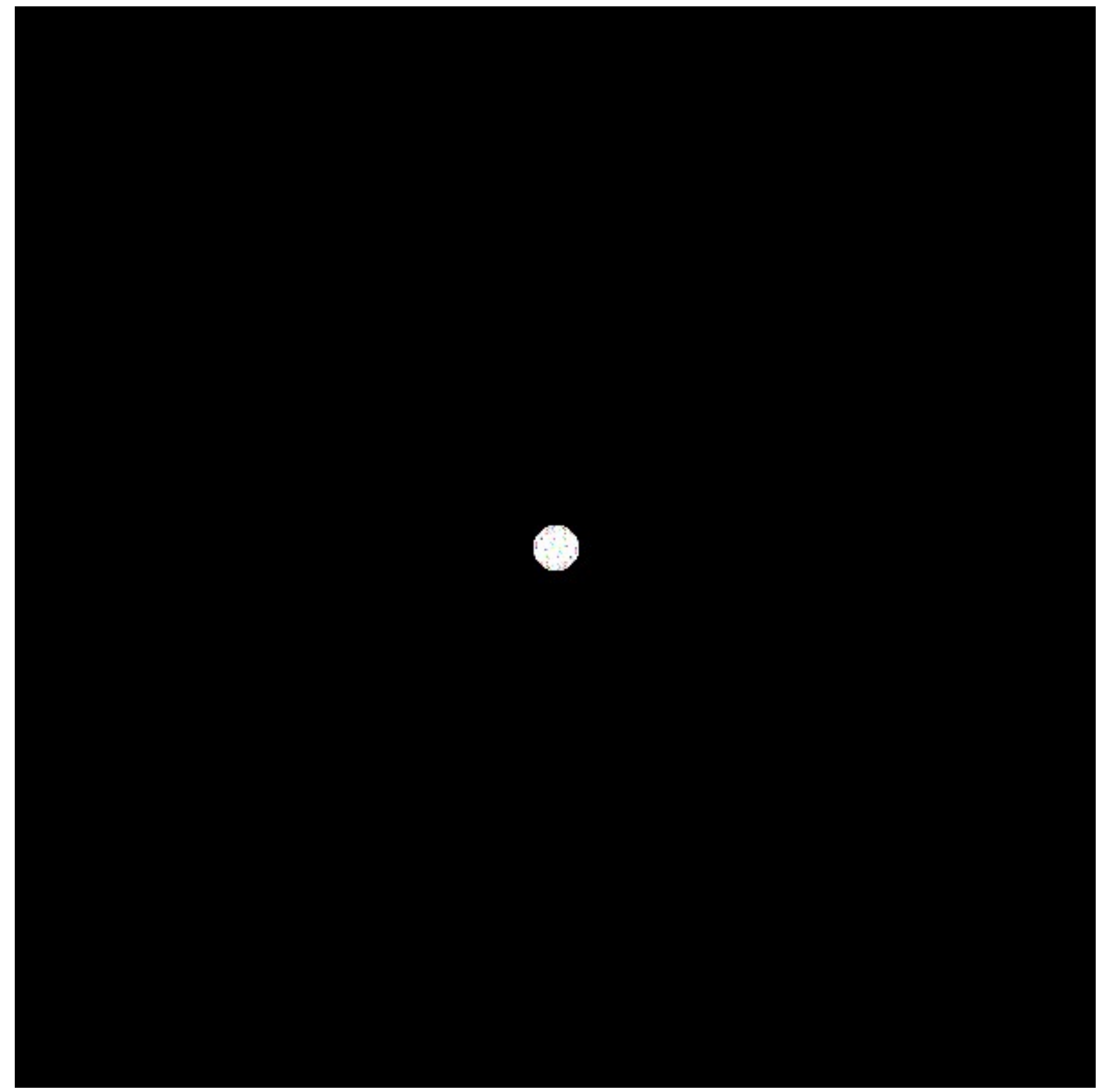
Spatial Domain



Frequency Domain

Filtering: Low Pass Filter

?



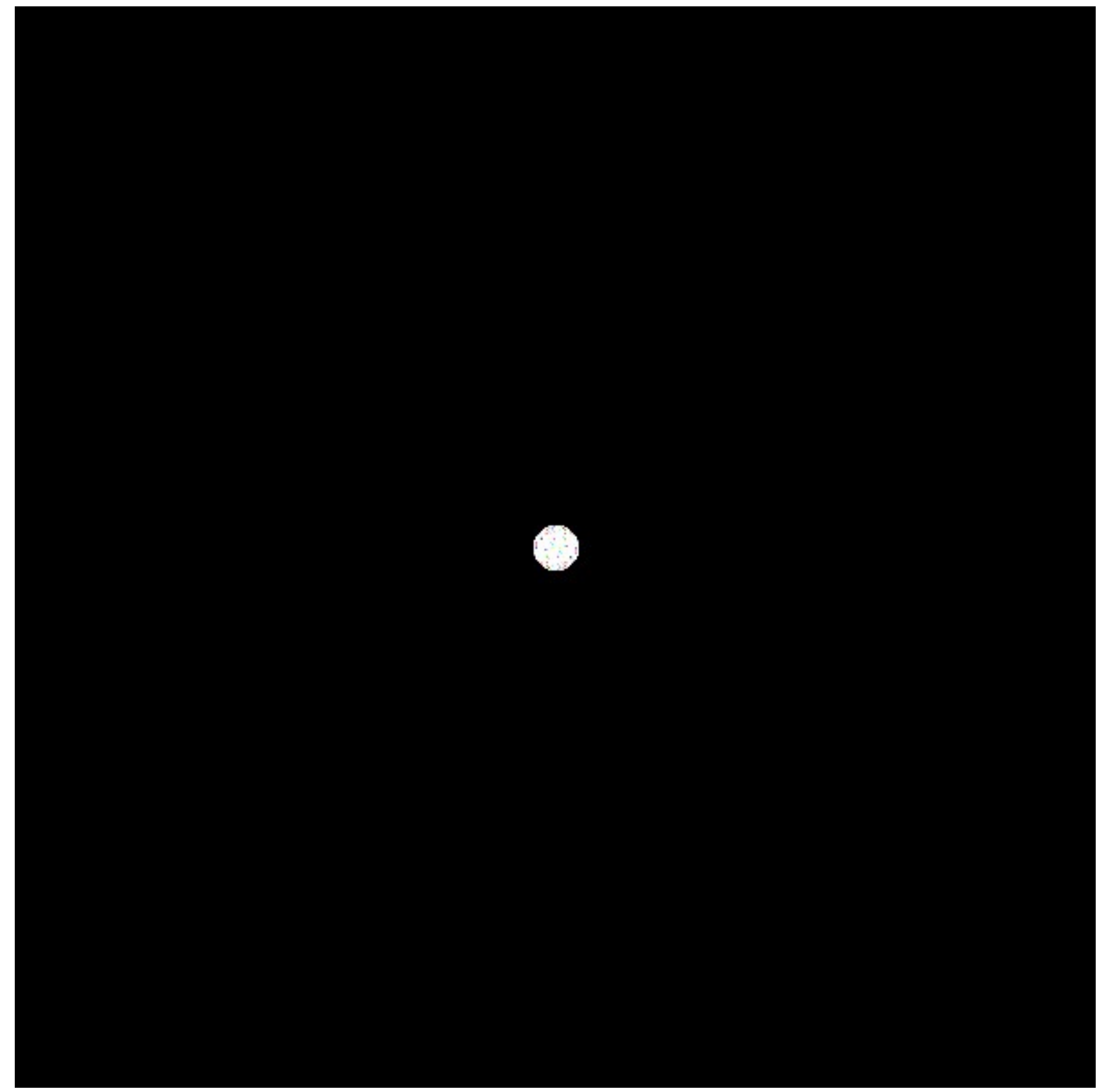
Spatial Domain

Frequency Domain

Filtering: Low Pass Filter

Keep low frequencies

Spatial Domain

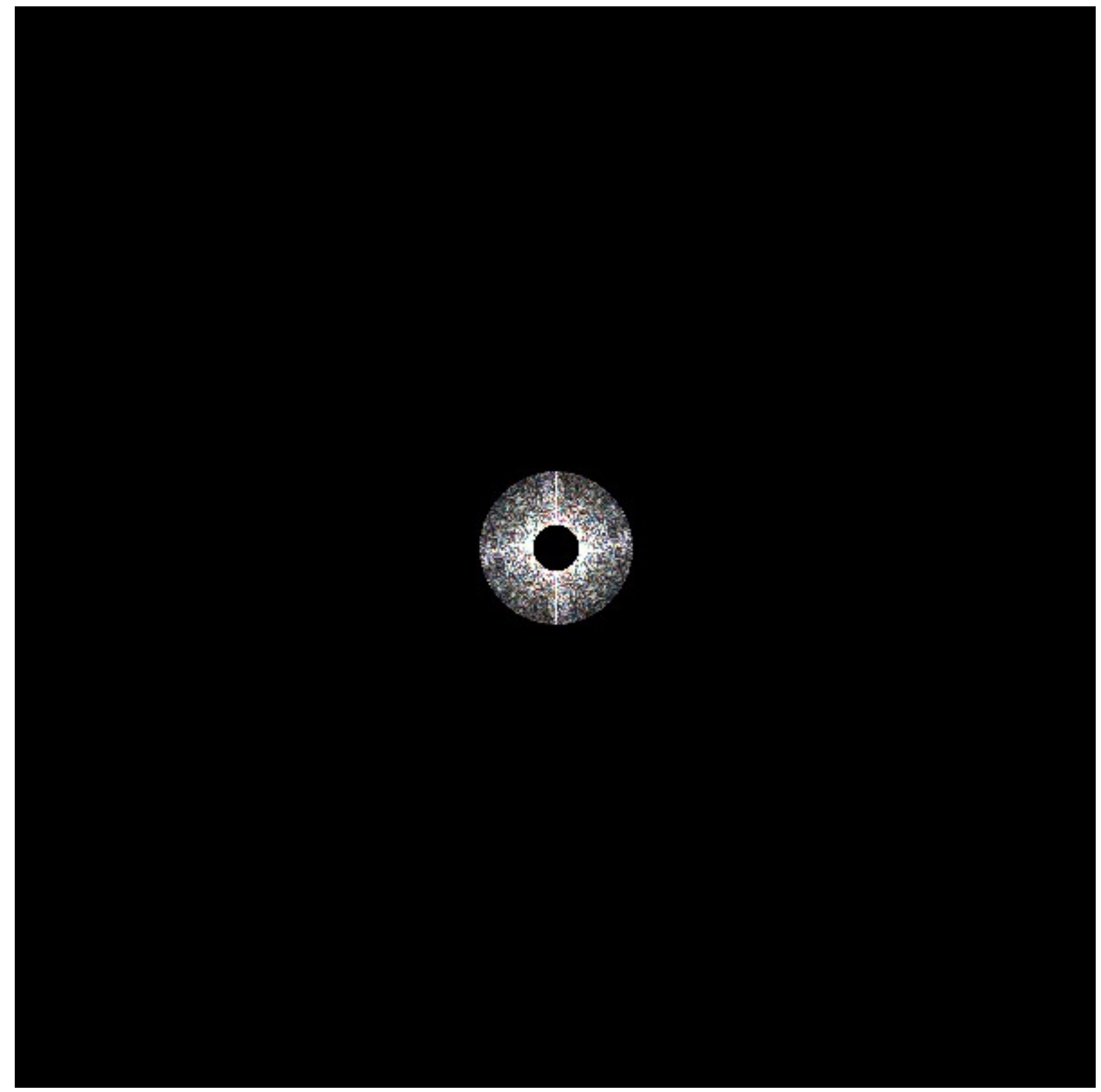


Frequency Domain

Filtering: Band Pass Filter

Keep band of frequencies

Spatial Domain

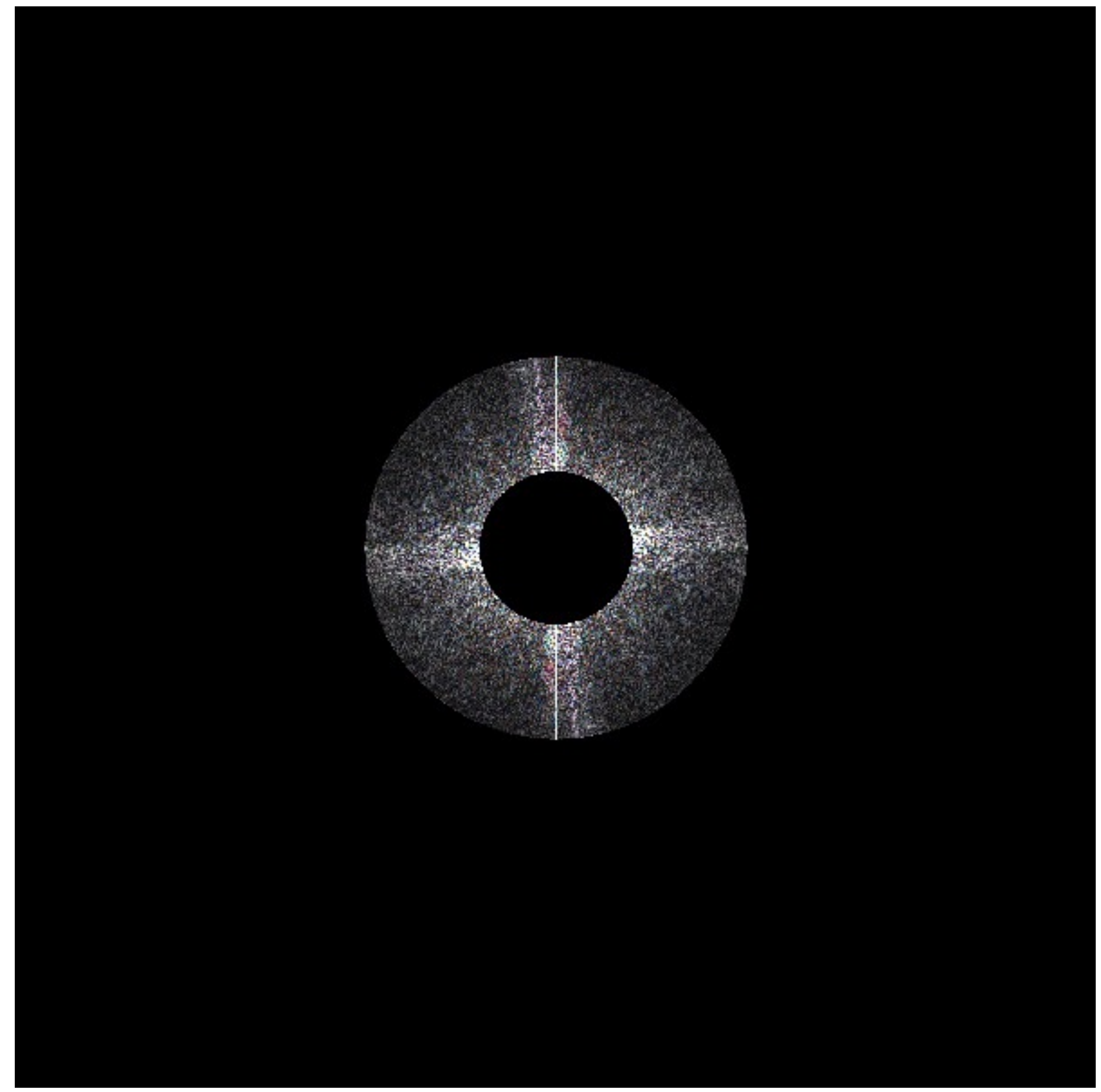


Frequency Domain

Filtering: Band Pass Filter

Keep band of frequencies

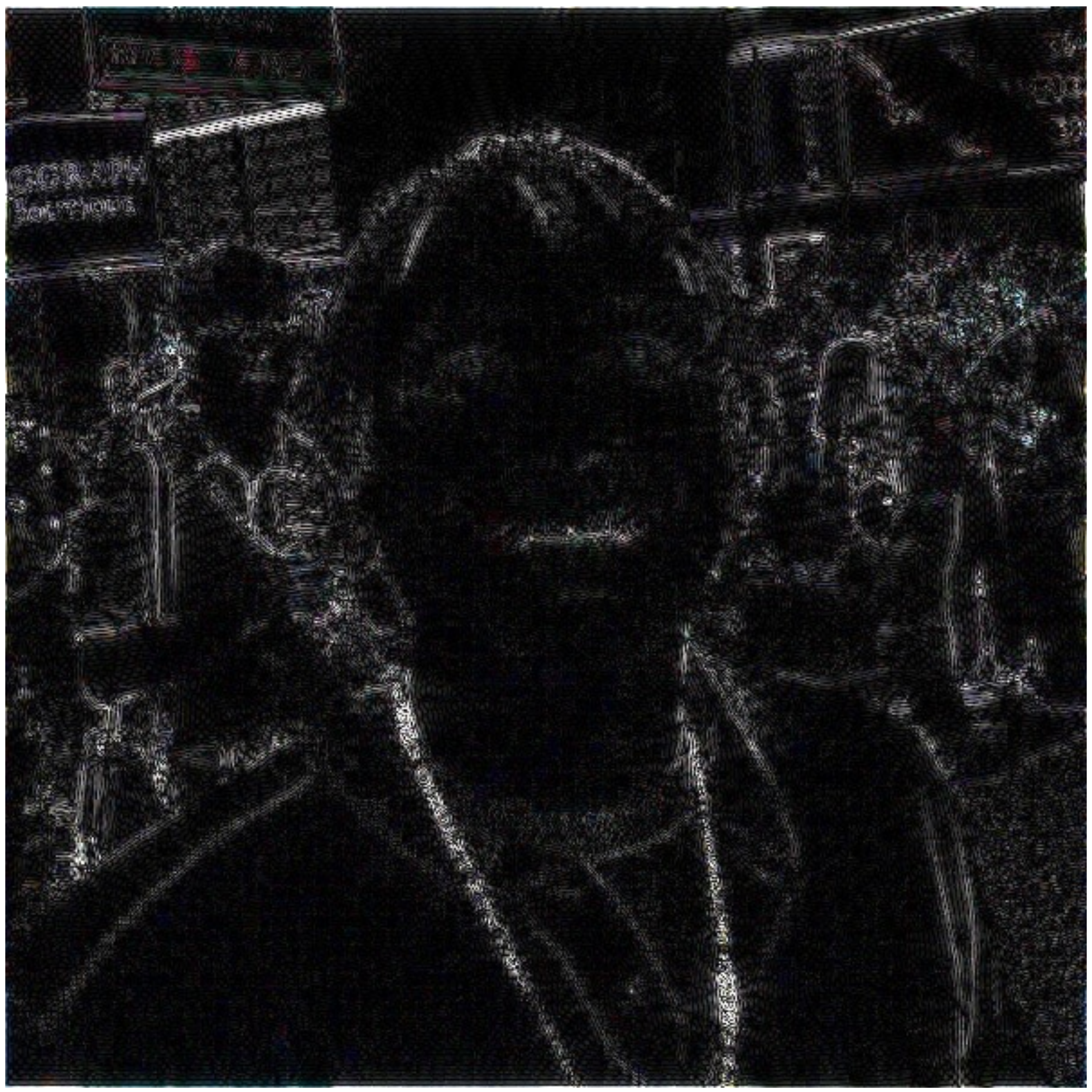
Spatial Domain



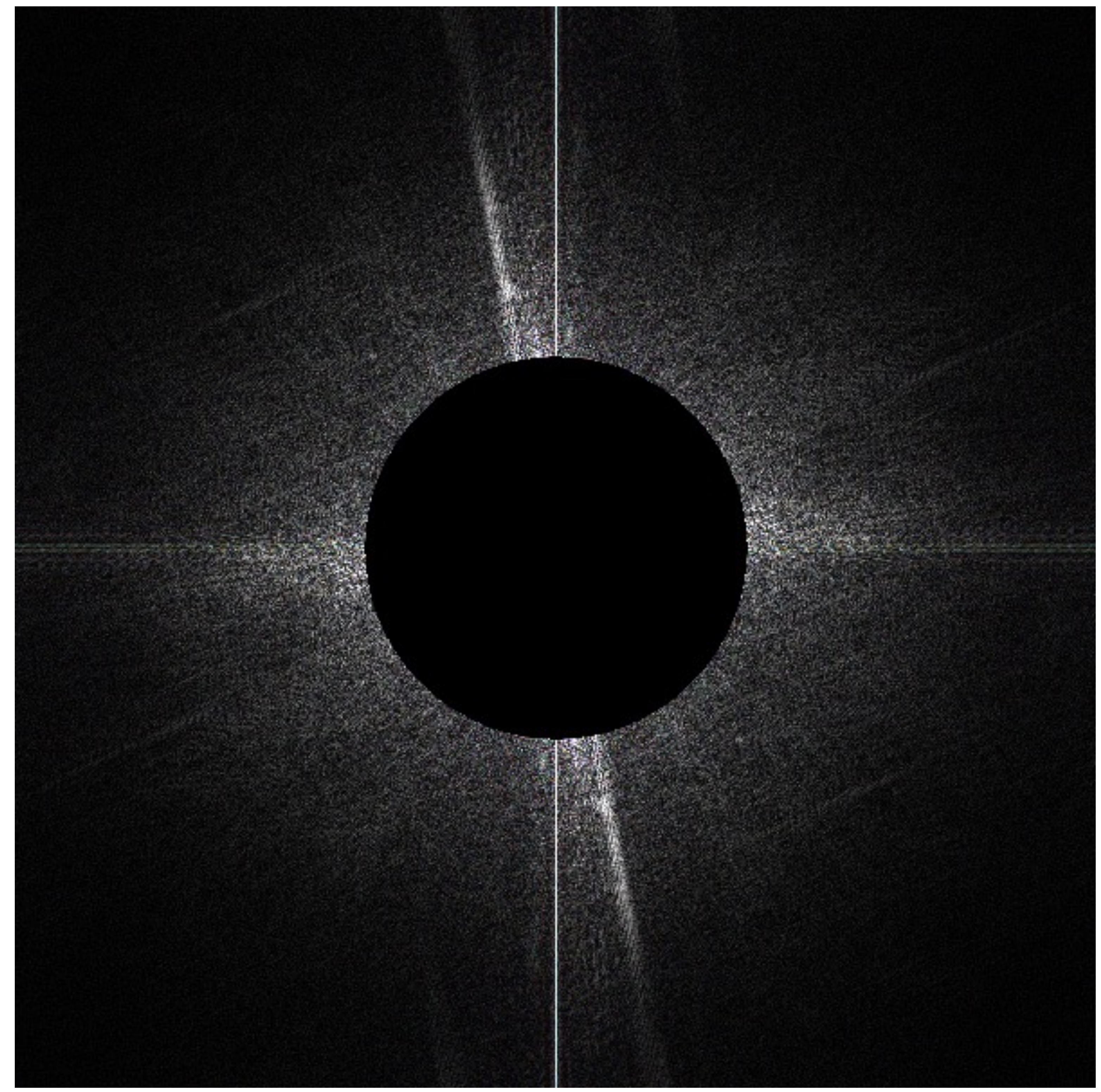
Frequency Domain

Filtering: High Pass Filter

Keep high frequencies

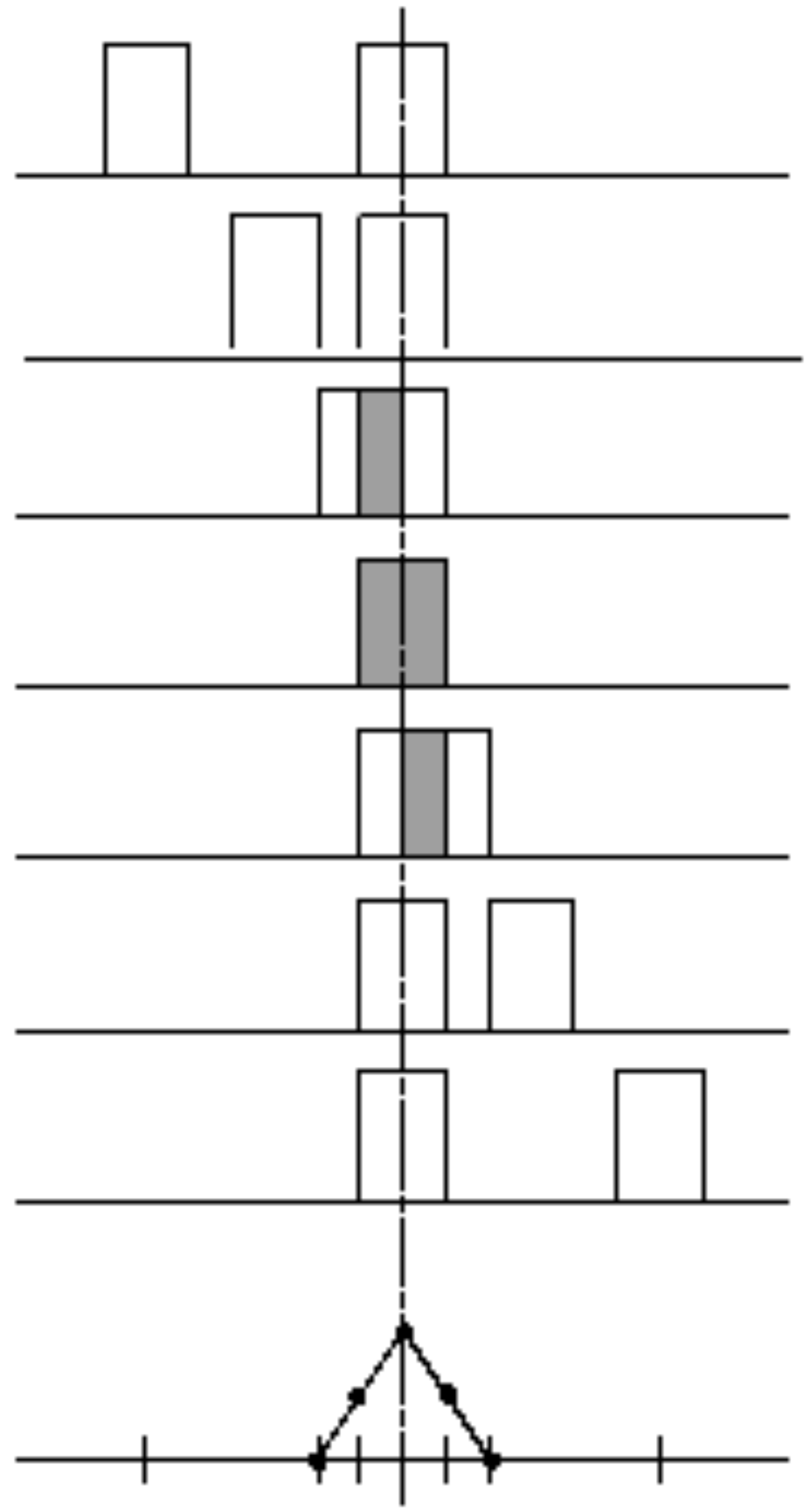


Spatial Domain



Frequency Domain

Filtering by Convolution



$$h(x) = f \otimes g = \int f(x')g(x - x')\,dx'$$

https://mathworld.wolfram.com/Convolution.html

Convolution Theorem

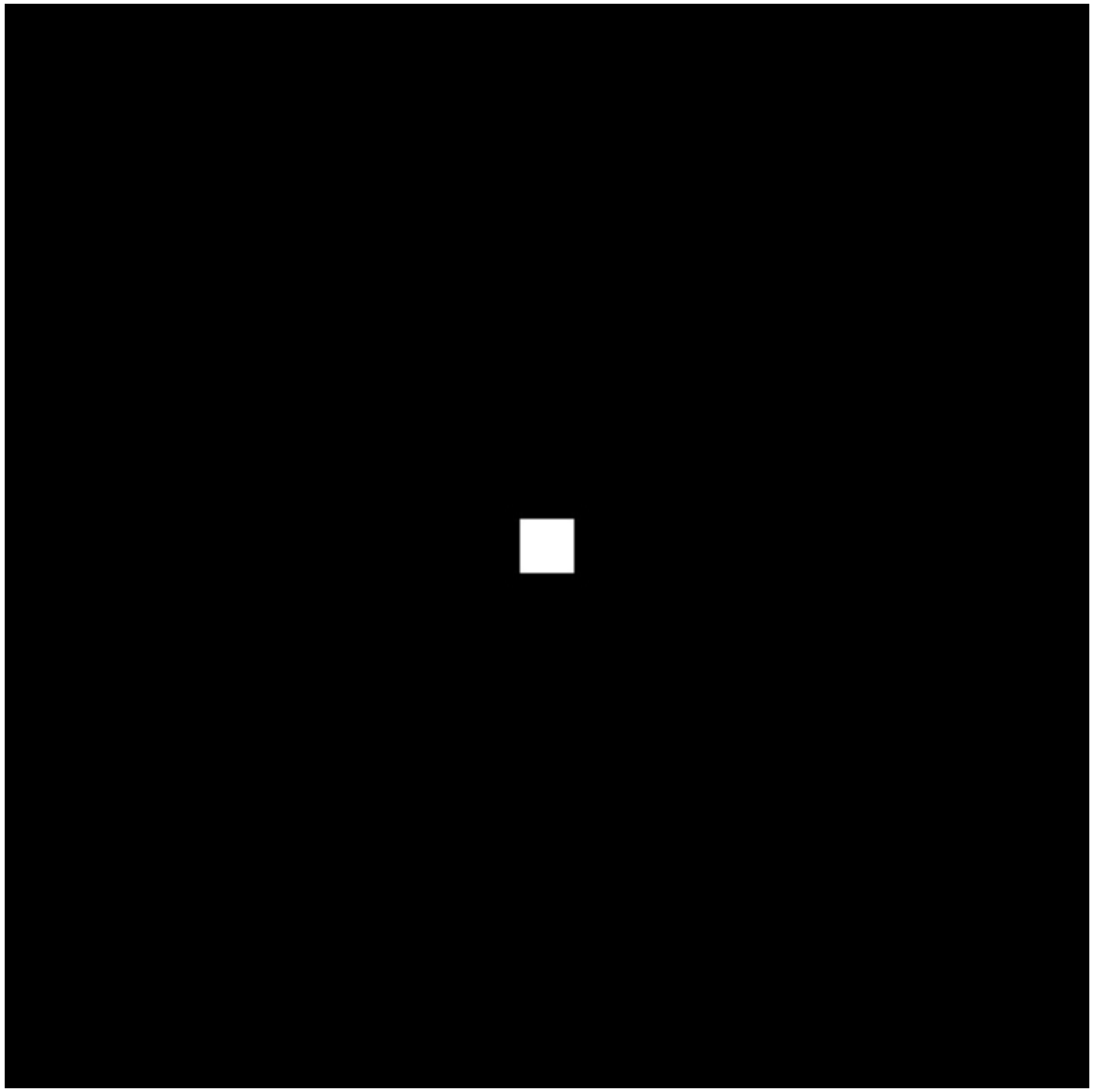
Convolution Theorem: Multiplication in the frequency domain is equivalent to convolution in the space domain.

$$f \otimes g \leftrightarrow F \times G$$

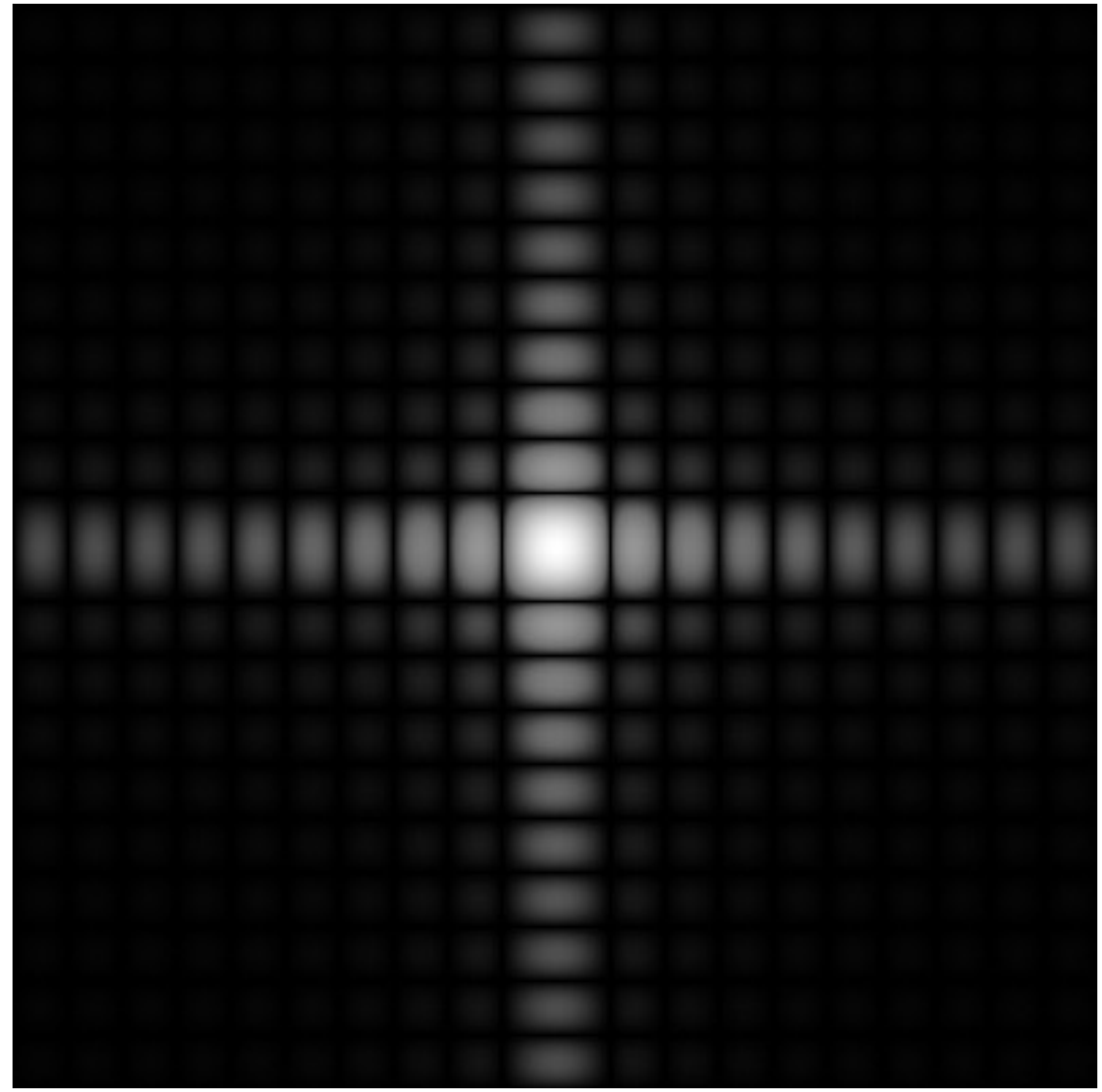
Symmetric Theorem: Multiplication in the space domain is equivalent to convolution in the frequency domain.

$$f \times g \leftrightarrow F \otimes G$$

Spatial and Frequency Domain

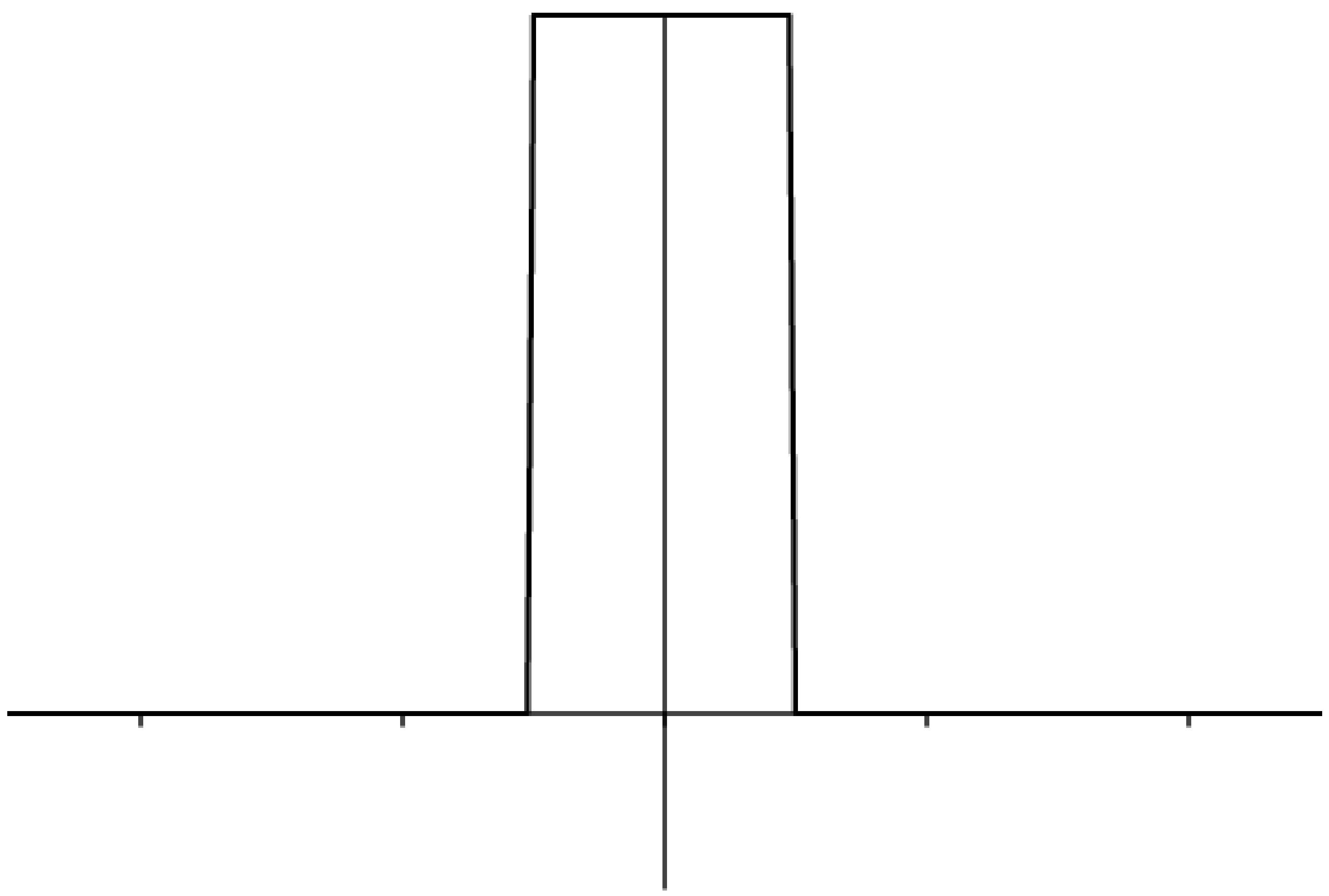


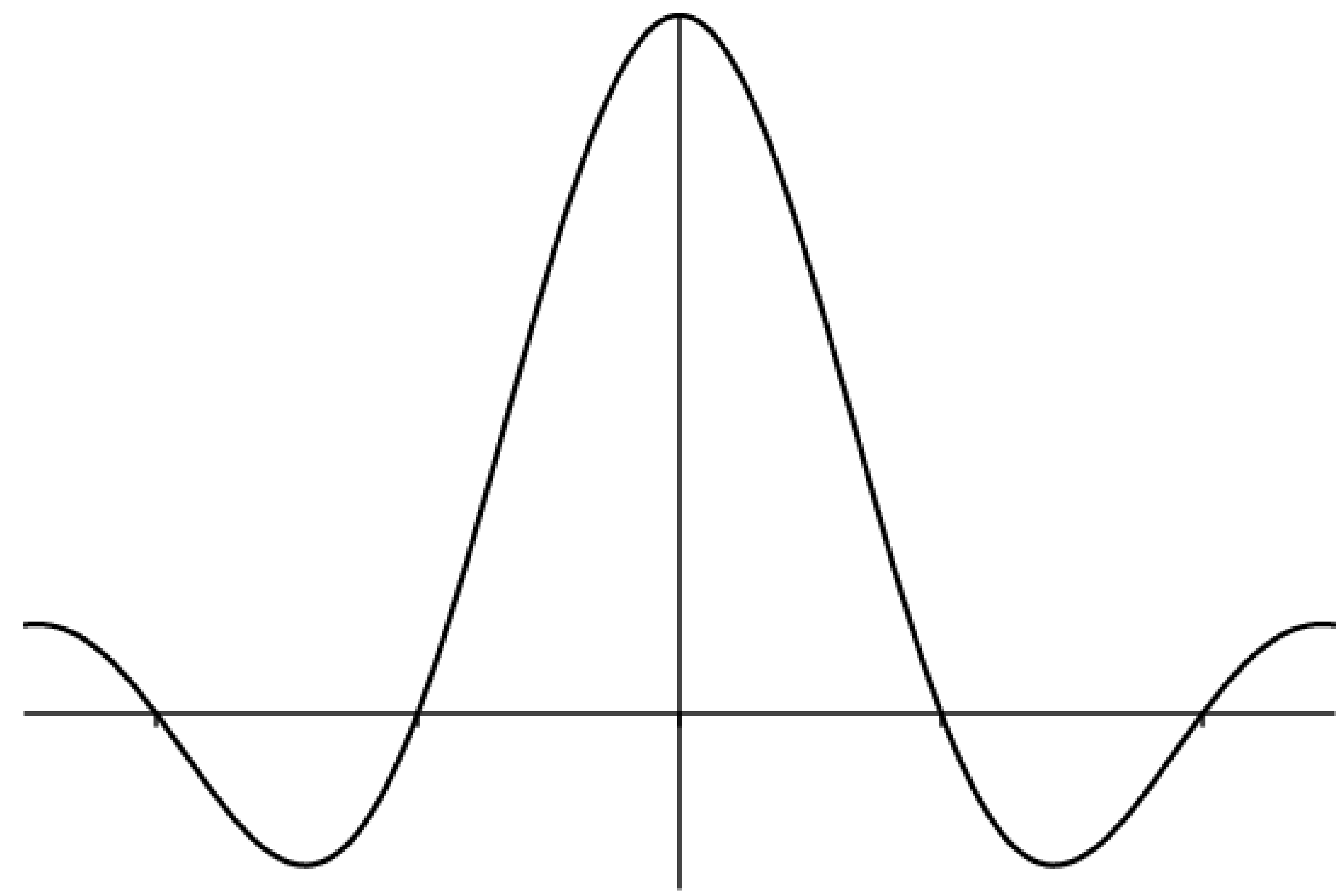
Spatial Domain



Frequency Domain

Math: Box and Sinc Functions



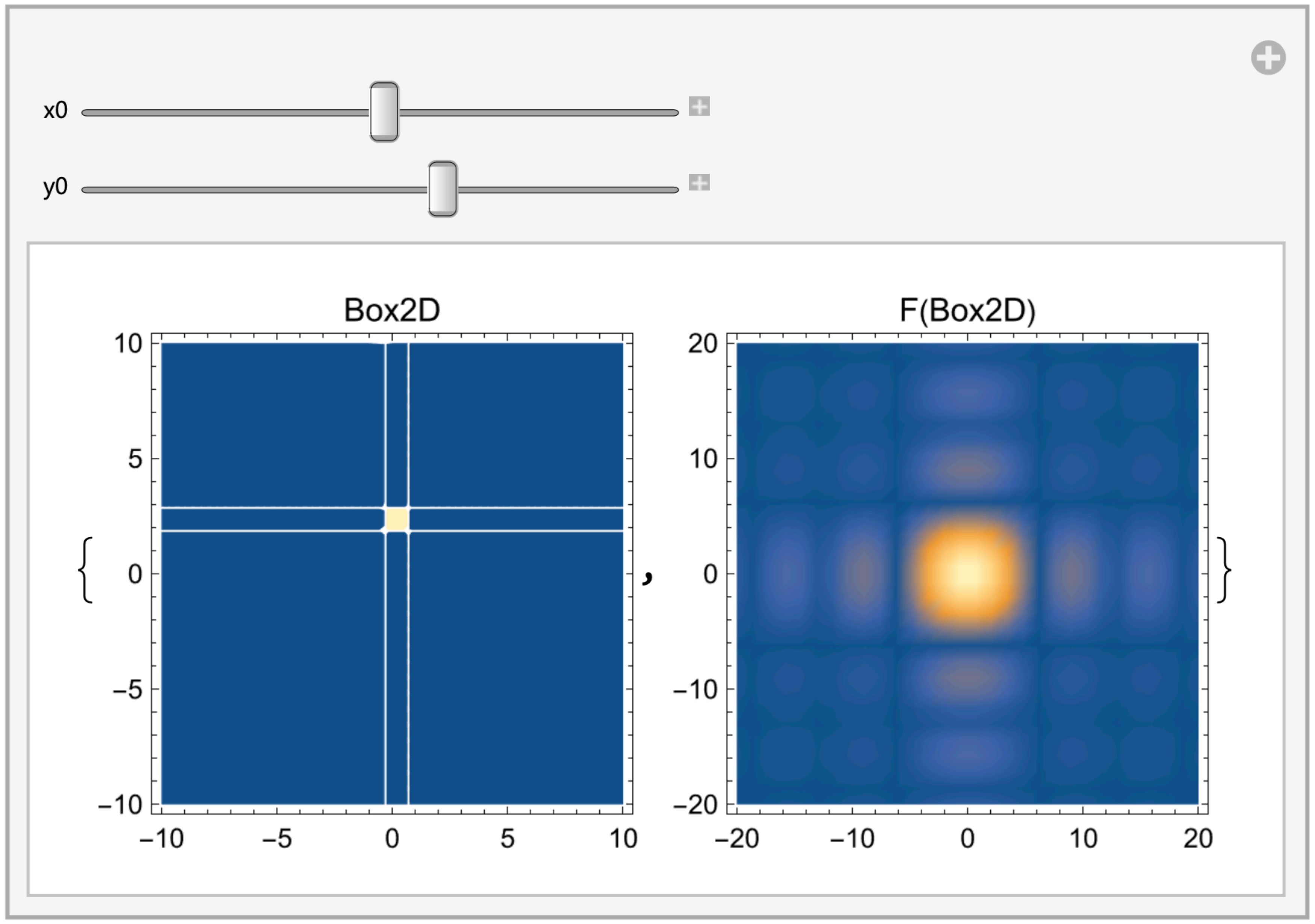


$$\prod_T(x) = \begin{cases} 1 & |x| \le \dfrac{T}{2} \\ 0 & |x| > \dfrac{T}{2} \end{cases}$$

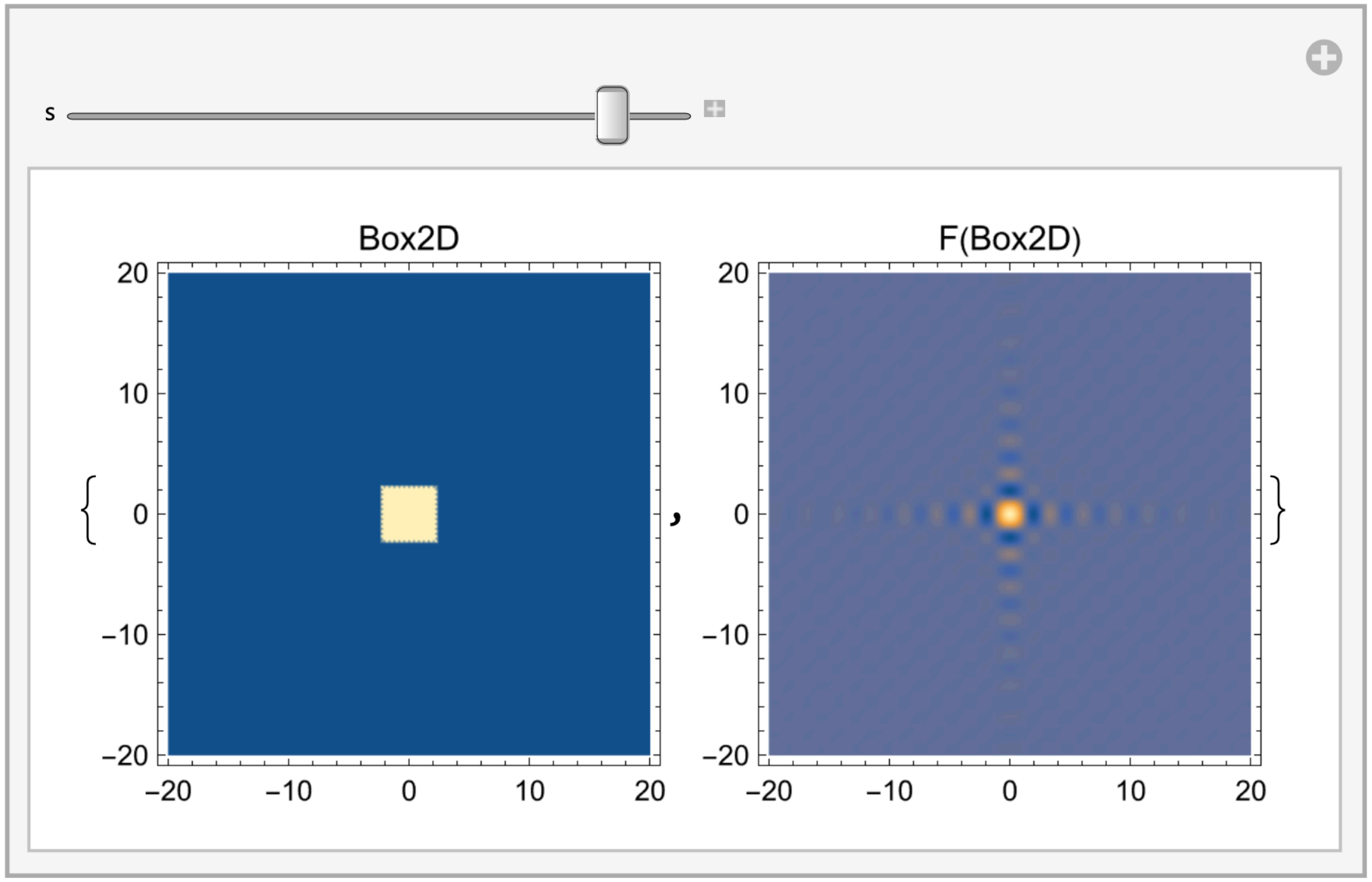
$$\mathrm{sinc}(\mathrm{x}) \equiv \frac{\sin \pi \mathrm{x}}{\pi \mathrm{x}}$$

$$\mathcal{F}(\Pi_T) \propto \; T\,\mathrm{sinc}(\pi T f)$$

Mathematica Demo (Box/Sinc)



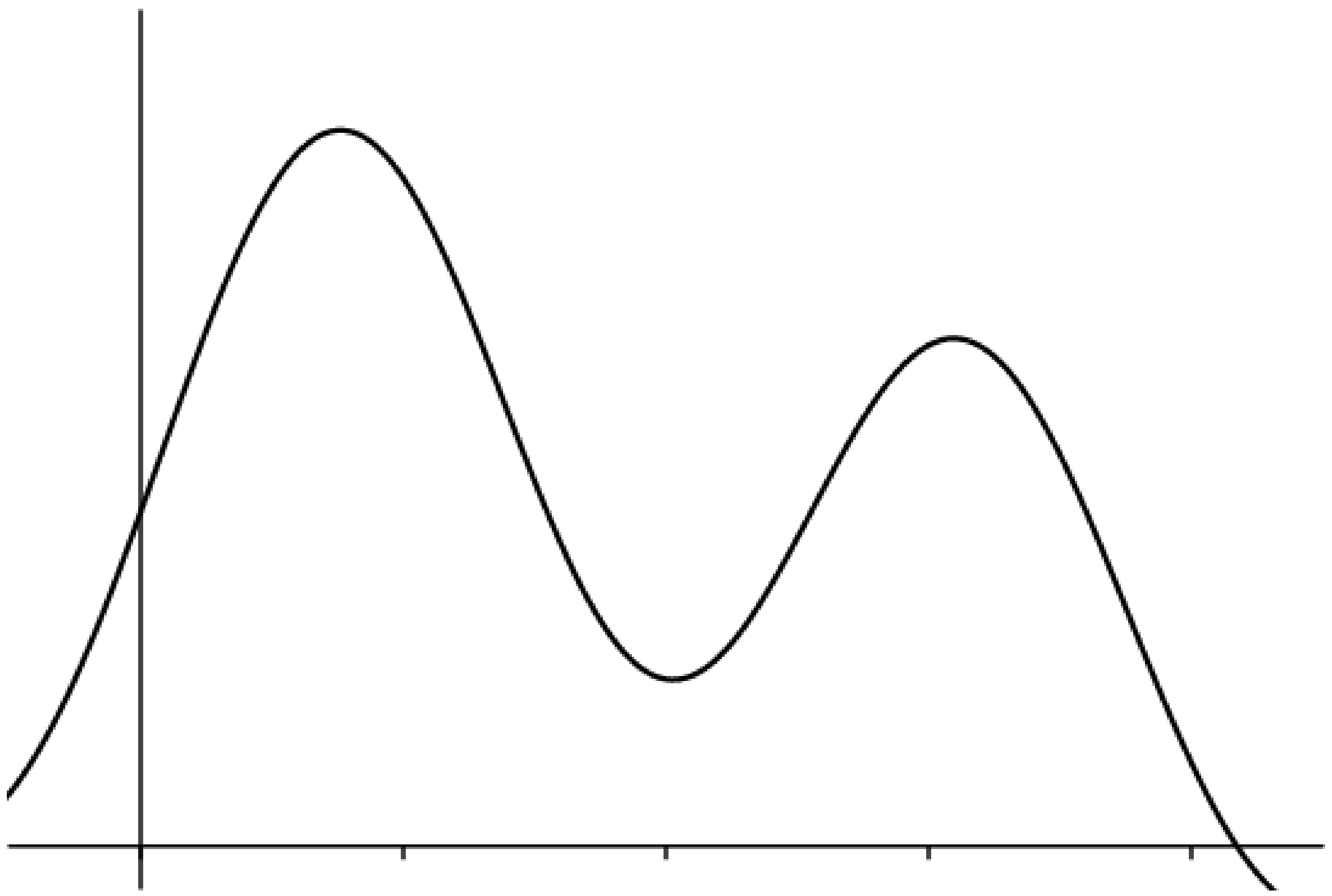
Translation



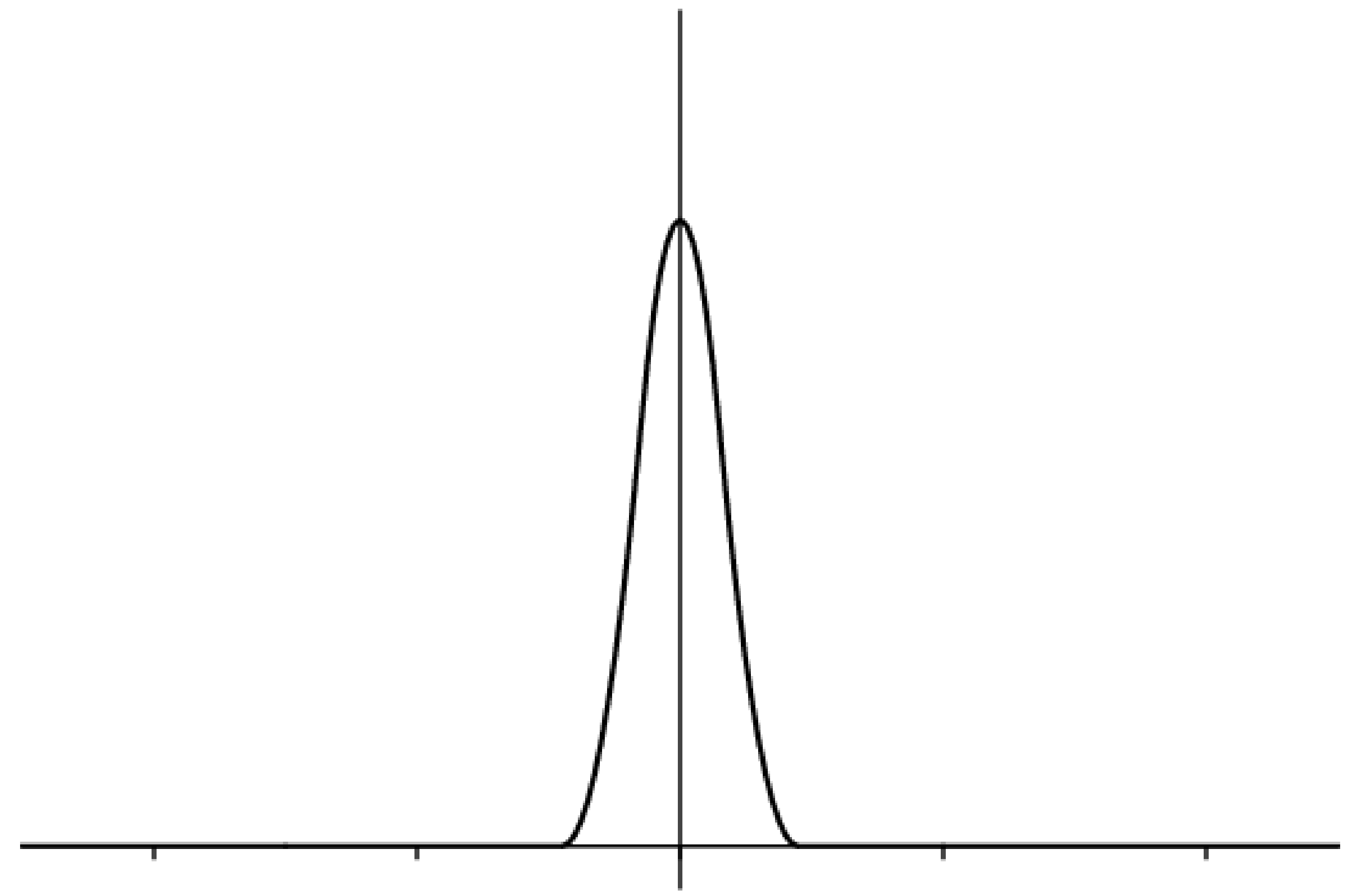
Scaling

The Sampling Theorem

Simple Function

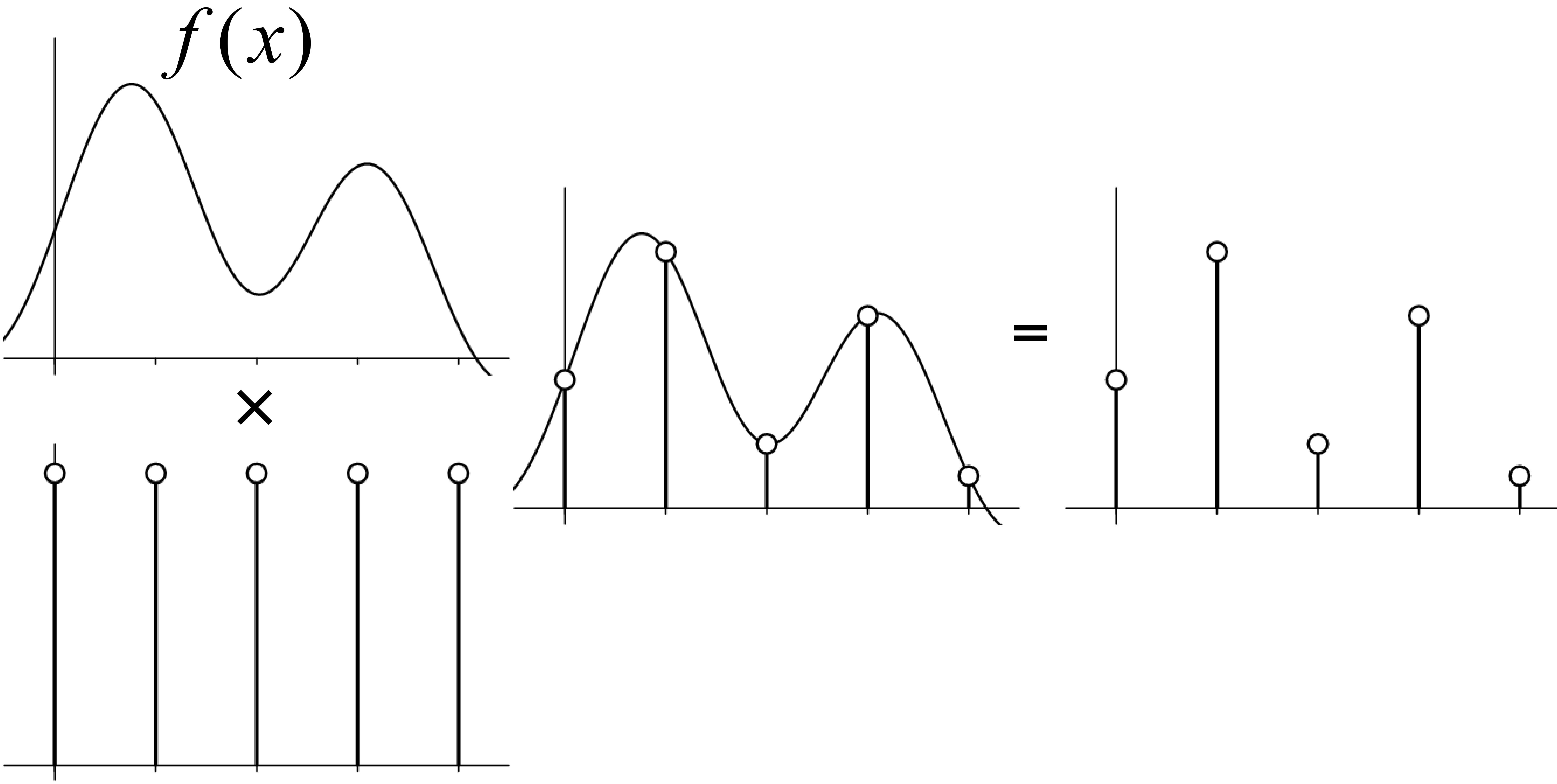


$f(x)$



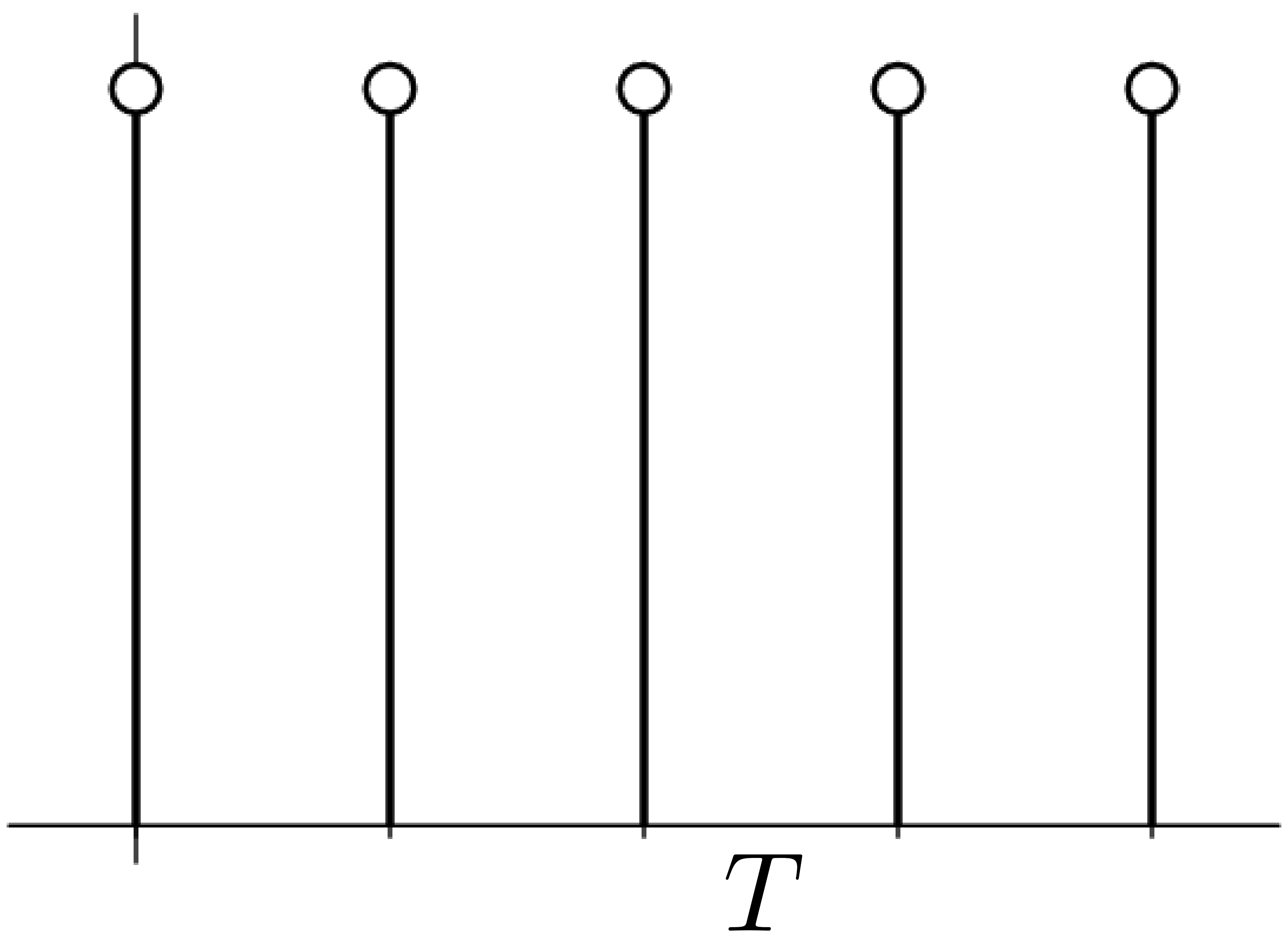
$F(\omega)$

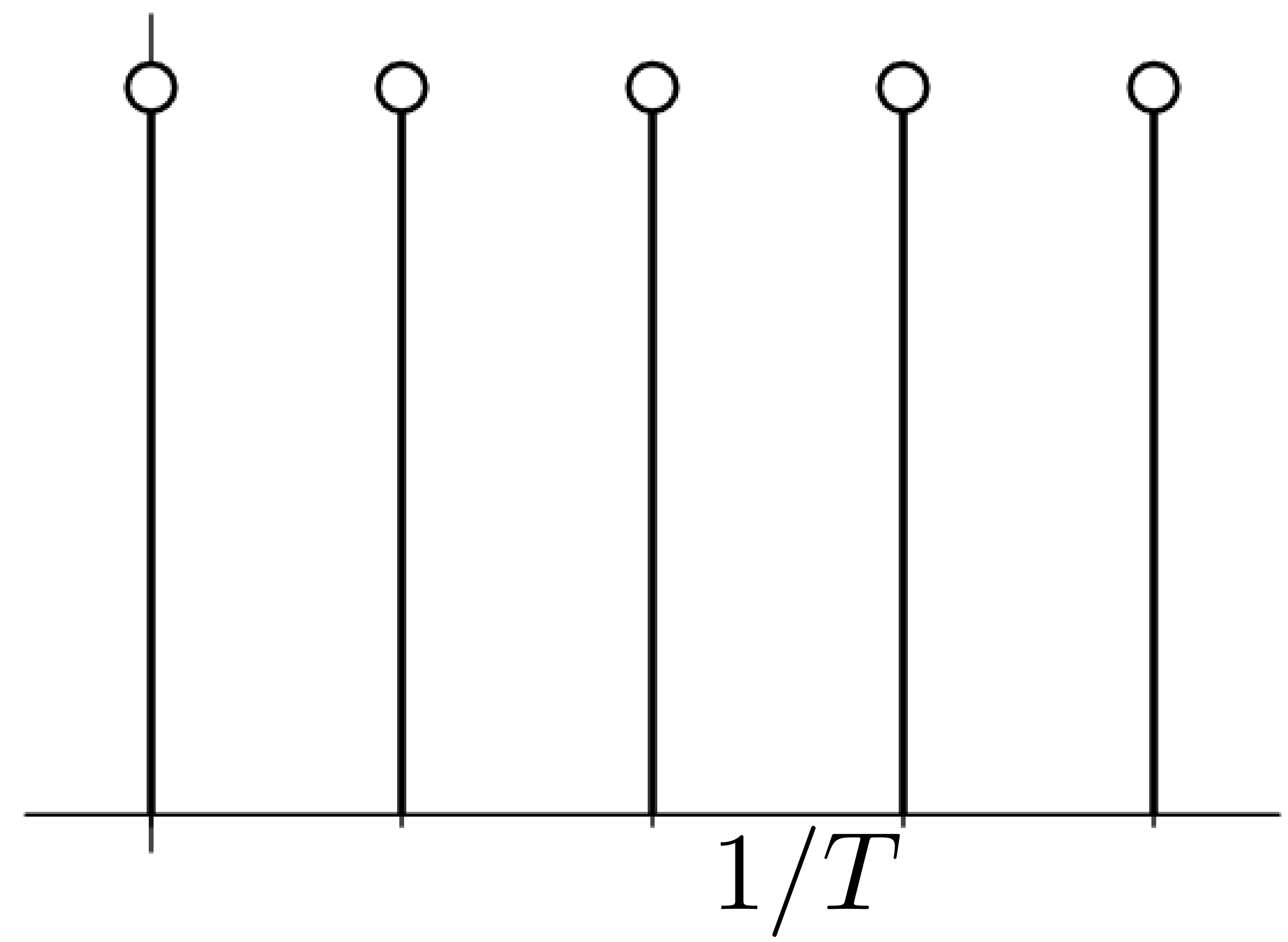
Sampling: Multiply in Spatial Domain



Some Magic

The Fourier transform of a sequence of spikes is a sequence of spikes

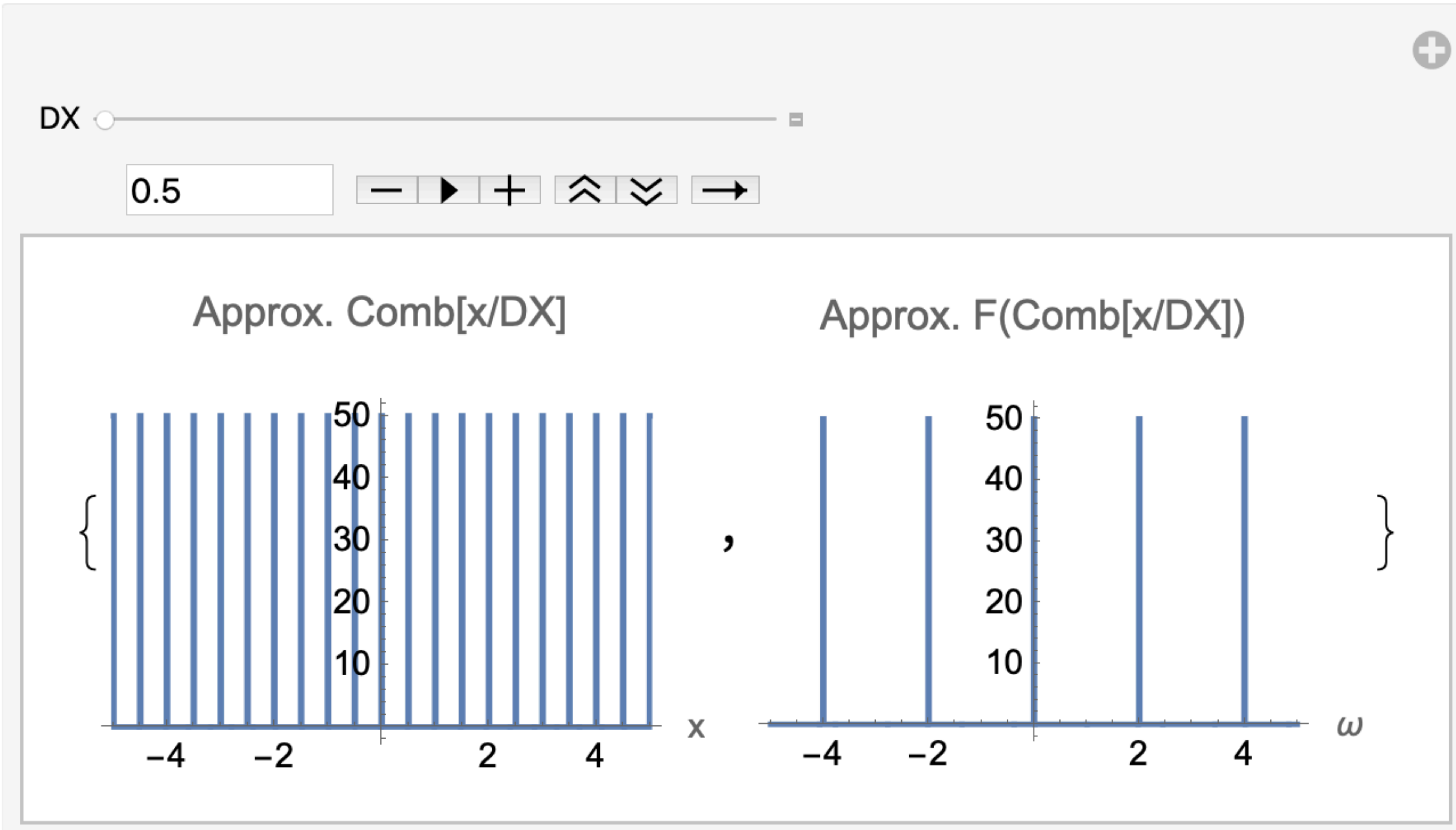




$$\mathrm{III_T(t)} = \sum_{\mathrm{n}=-\infty}^{\mathrm{n}=\infty} \delta(\mathrm{t} - \mathrm{nT}) \qquad \mathrm{III_{1/T}}(\omega) = \sum_{\mathrm{n}=-\infty}^{\mathrm{n}=\infty} \delta(\omega - \mathrm{n/T})$$

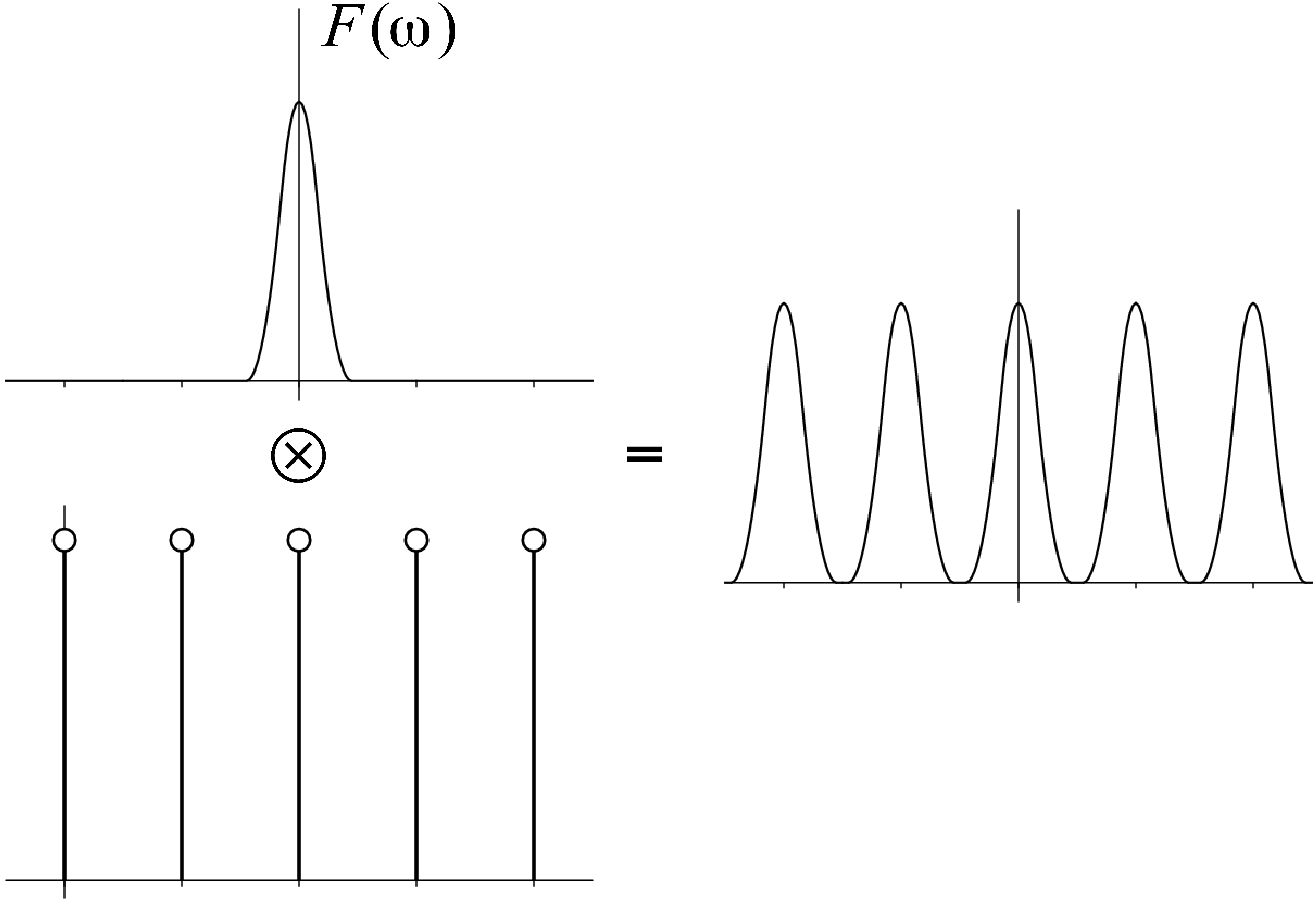
Comb or Shah function

Mathematica Demo (Combs)

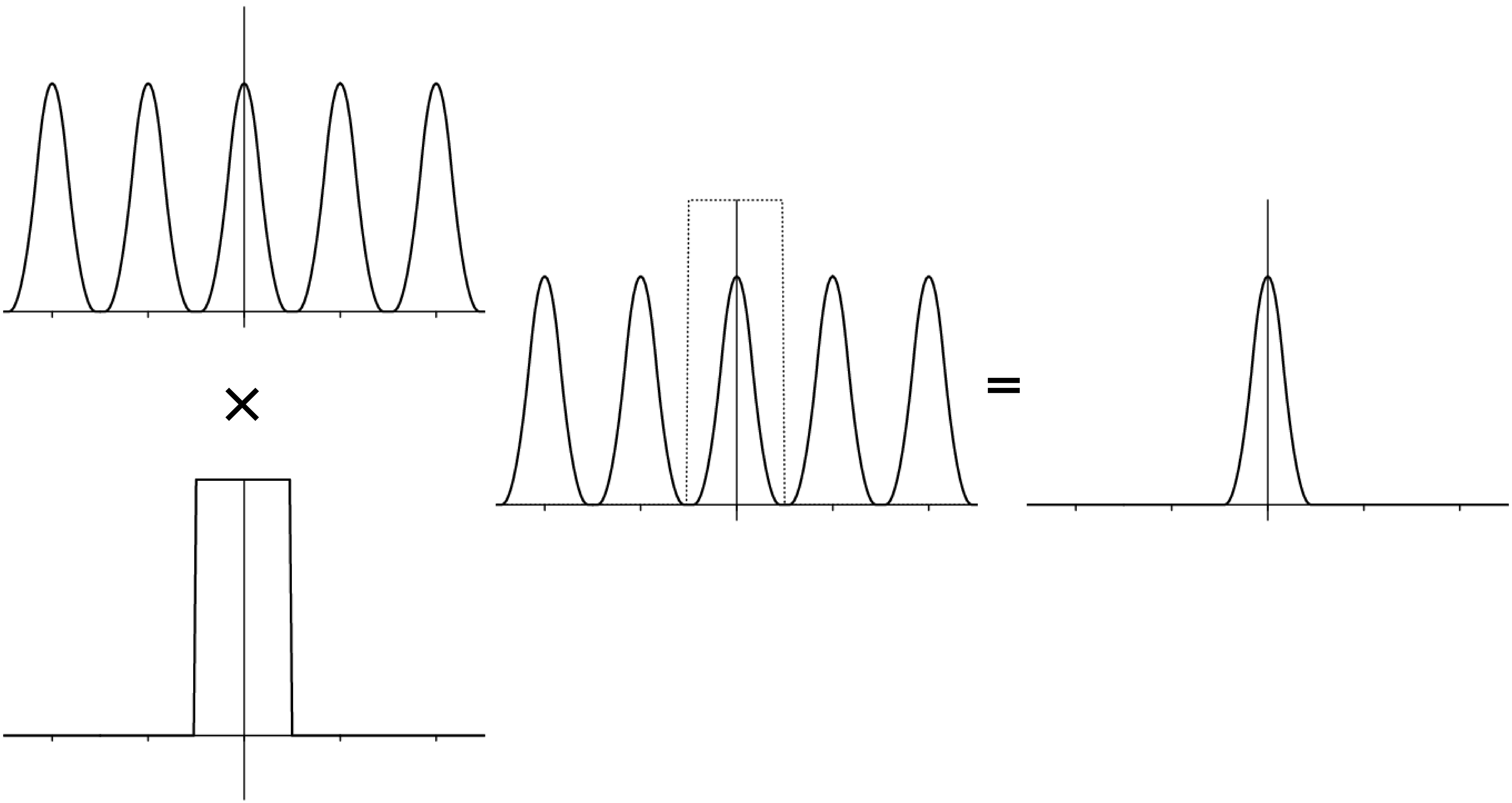


Sampling with Cheap Combs

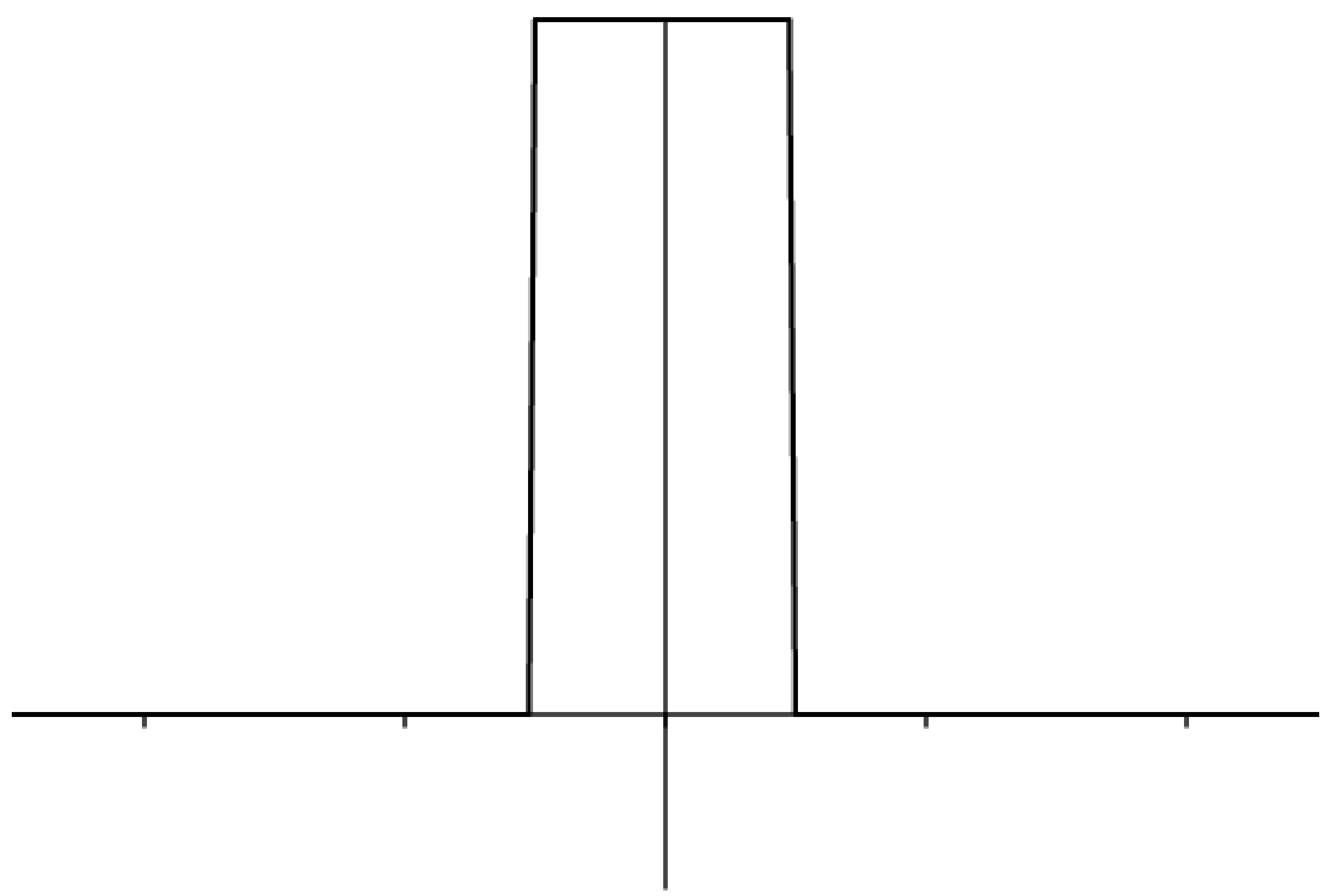
Sampling: Convolve in Freq Domain

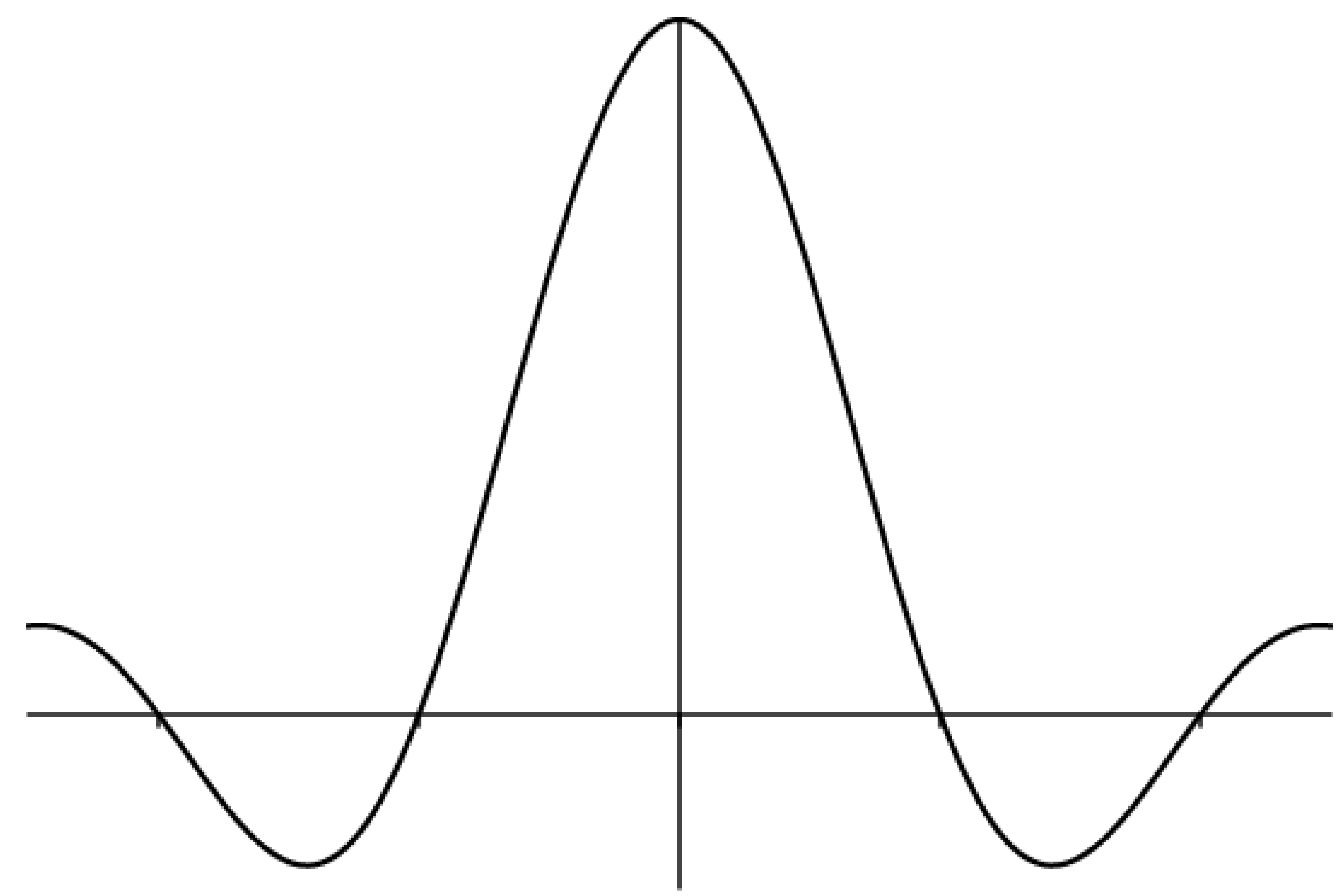


Reconstruction: Frequency Domain



Recall

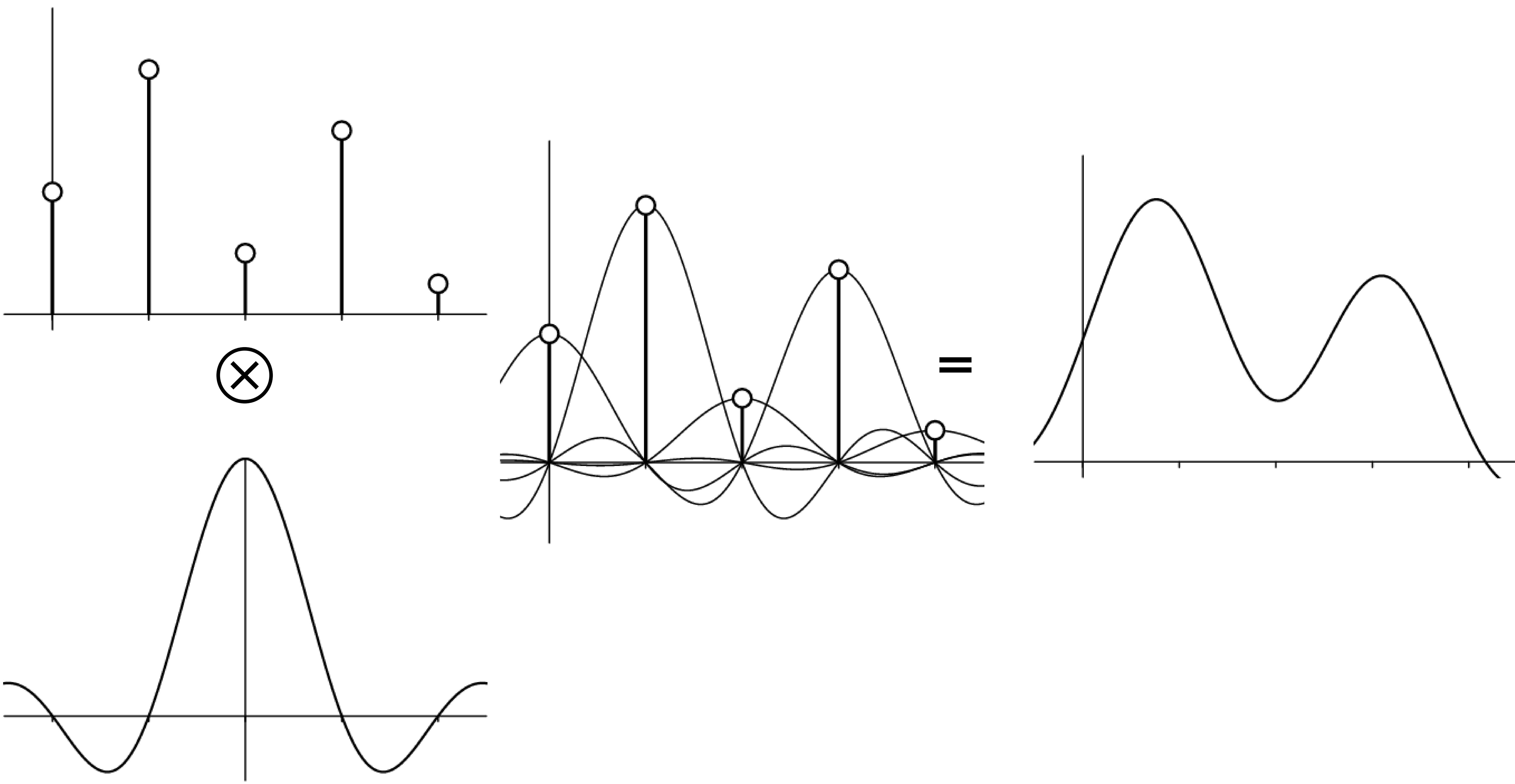




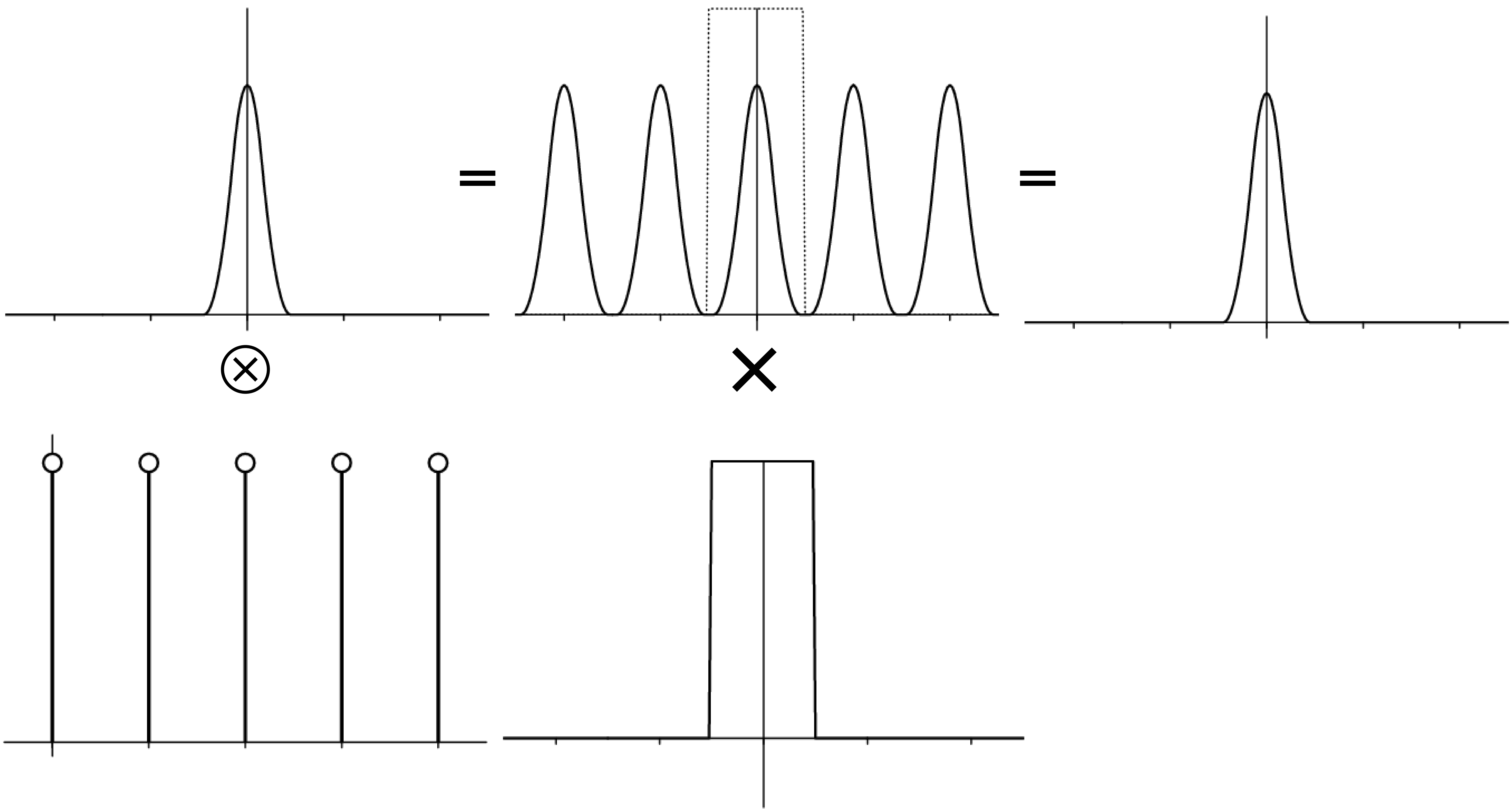
$$\prod_T(x) = \begin{cases} 1 & |x| \le \frac{T}{2} \\ 0 & |x| > \frac{T}{2} \end{cases}$$

$$\mathrm{sinc}(\mathrm{x}) \equiv \frac{\sin \pi \mathrm{x}}{\pi \mathrm{x}}$$

Reconstruction: Spatial Domain



Recovering a Sampled Signal



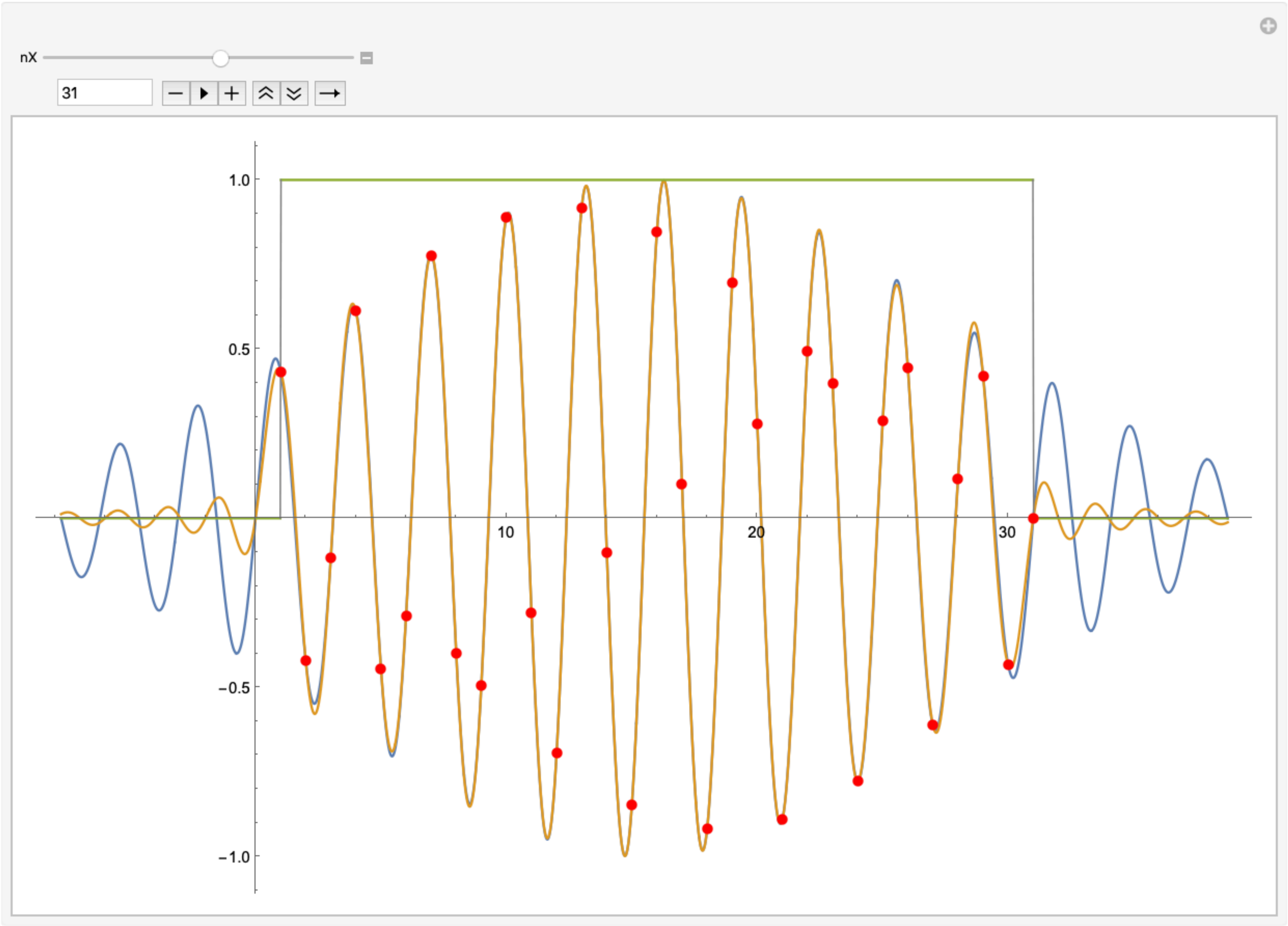
Sampling Theorem

This result is known as the Sampling Theorem, and Claude Shannon is credited with discovering it in 1949.

A SIGNAL CAN BE RECONSTRUCTED FROM ITS SAMPLES WITHOUT LOSS OF INFORMATION, IF THE ORIGINAL SIGNAL HAS NO FREQUENCIES ABOVE 1/2 THE SAMPLING FREQUENCY.

For a given band-limited function, the rate it must be sampled is called the Nyquist Frequency.

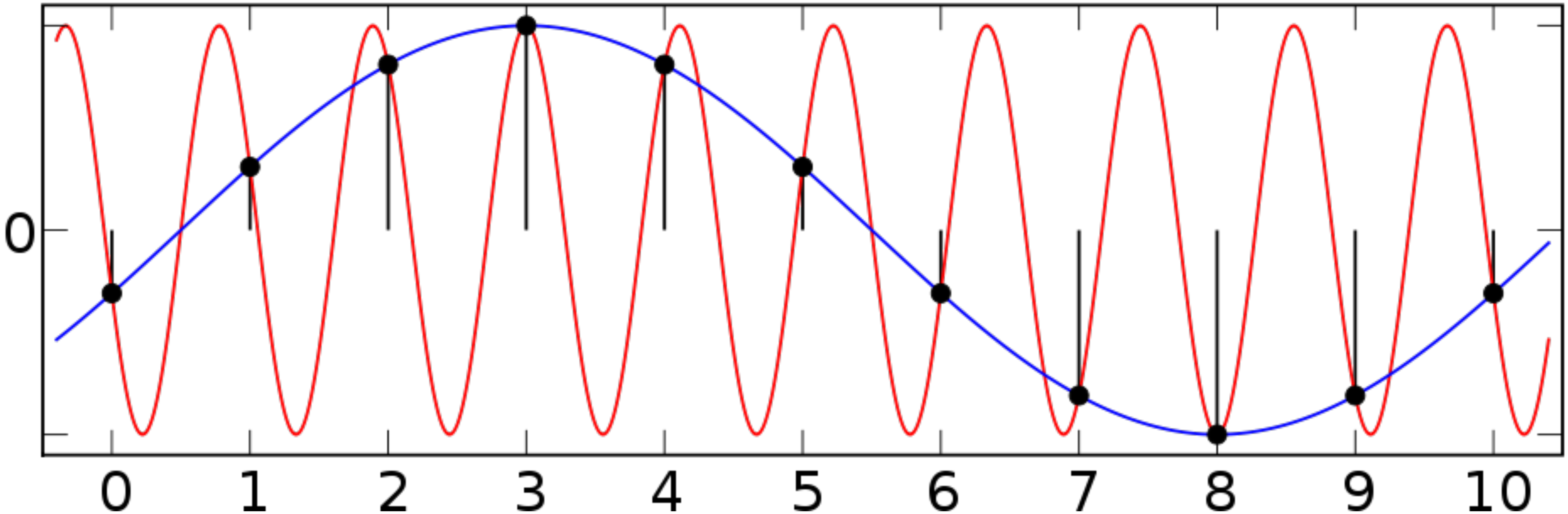
Mathematica Demo (Reconstruction)



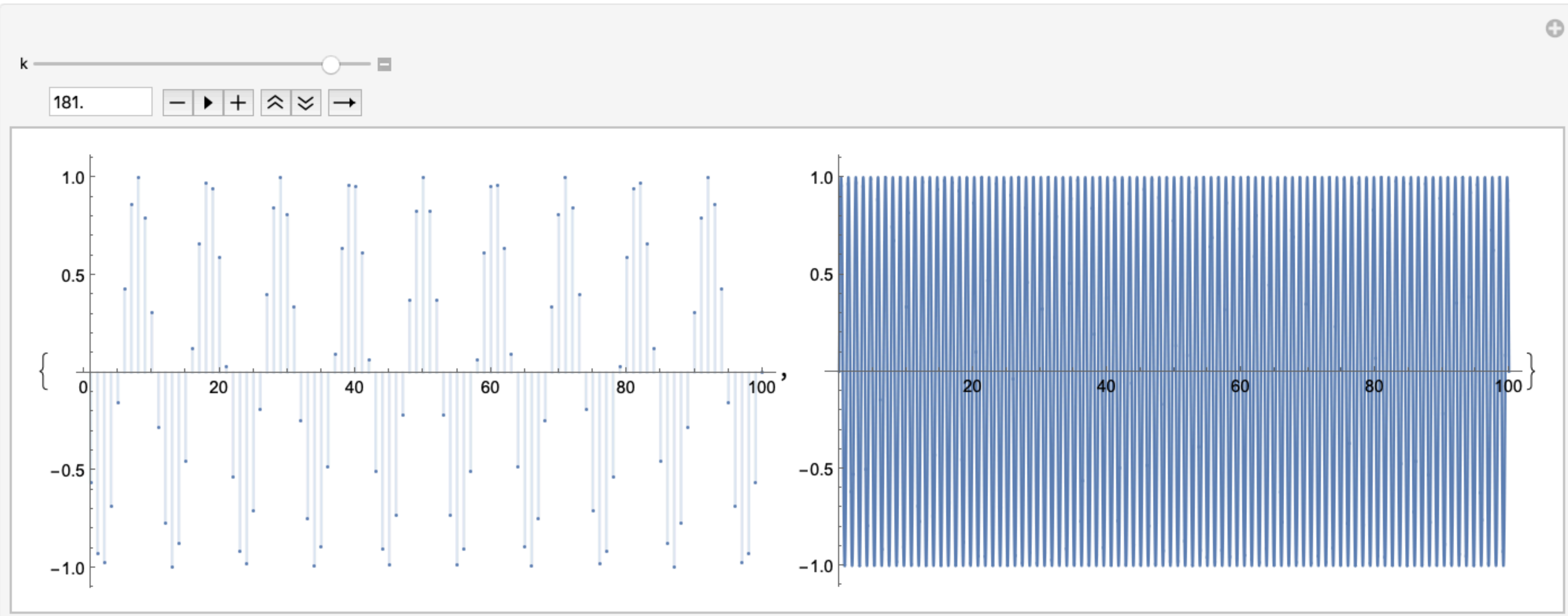
Spatial Reconstruction

Aliasing

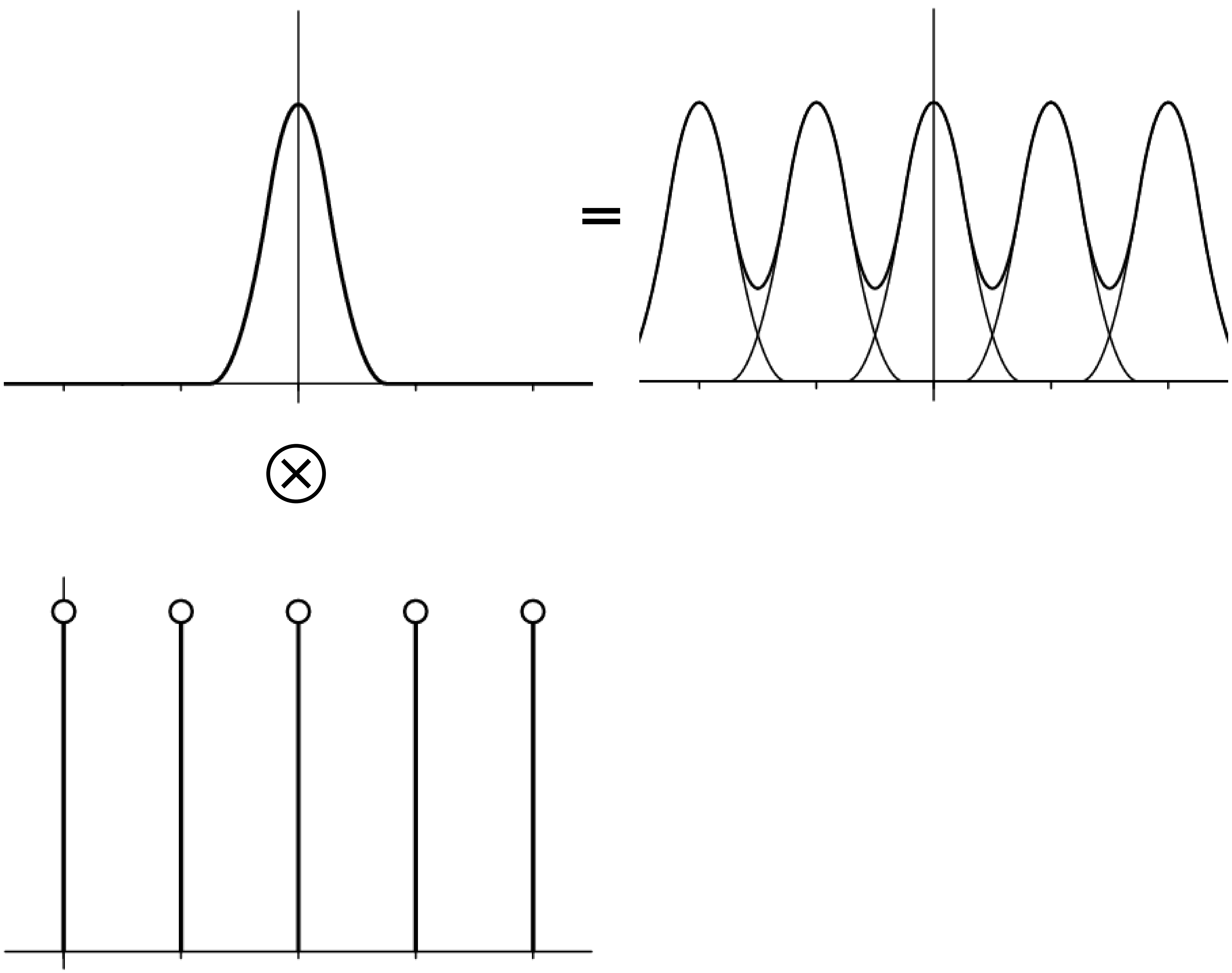
Aliasing



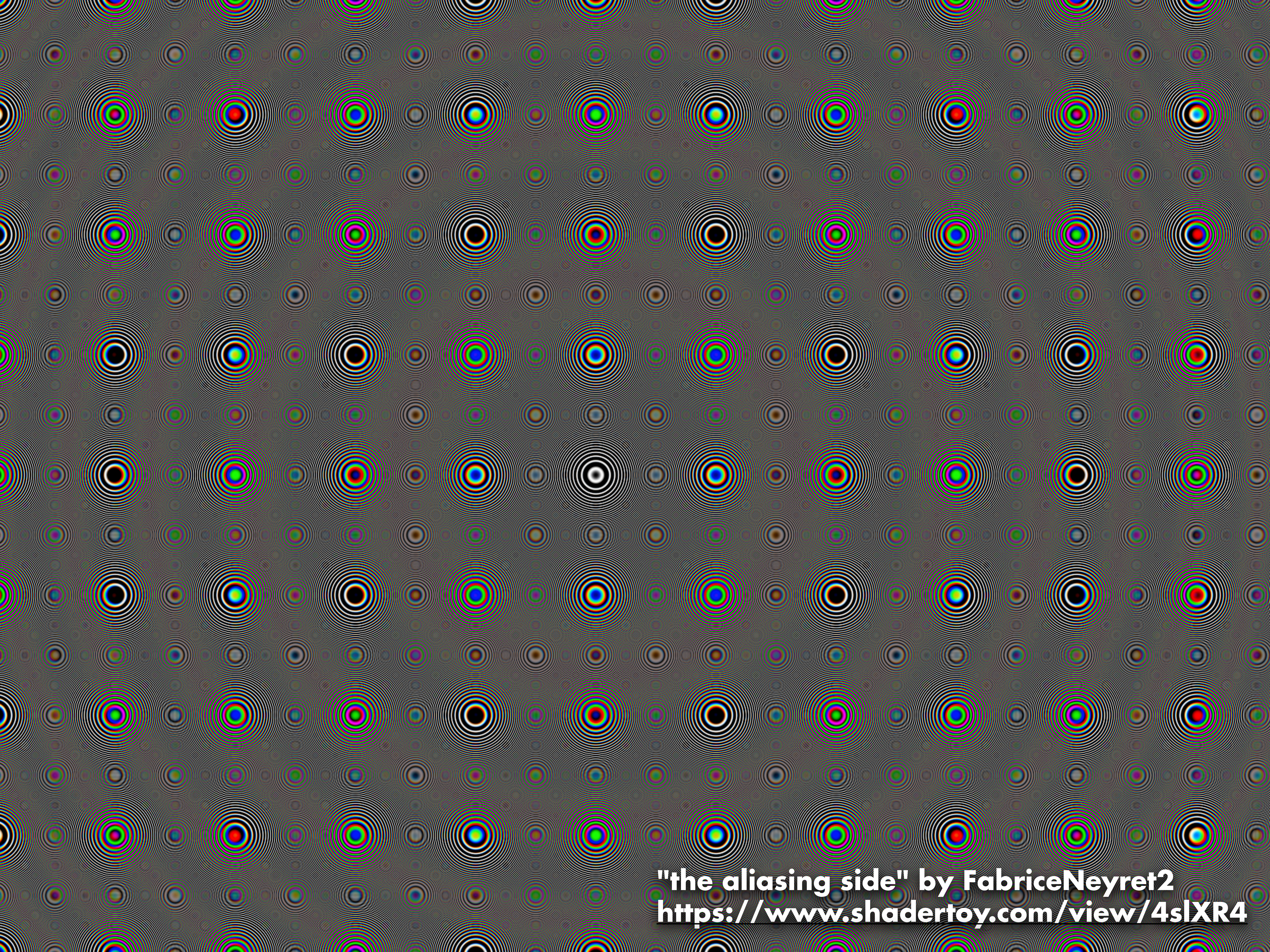
Mathematica Demos (Aliasing)



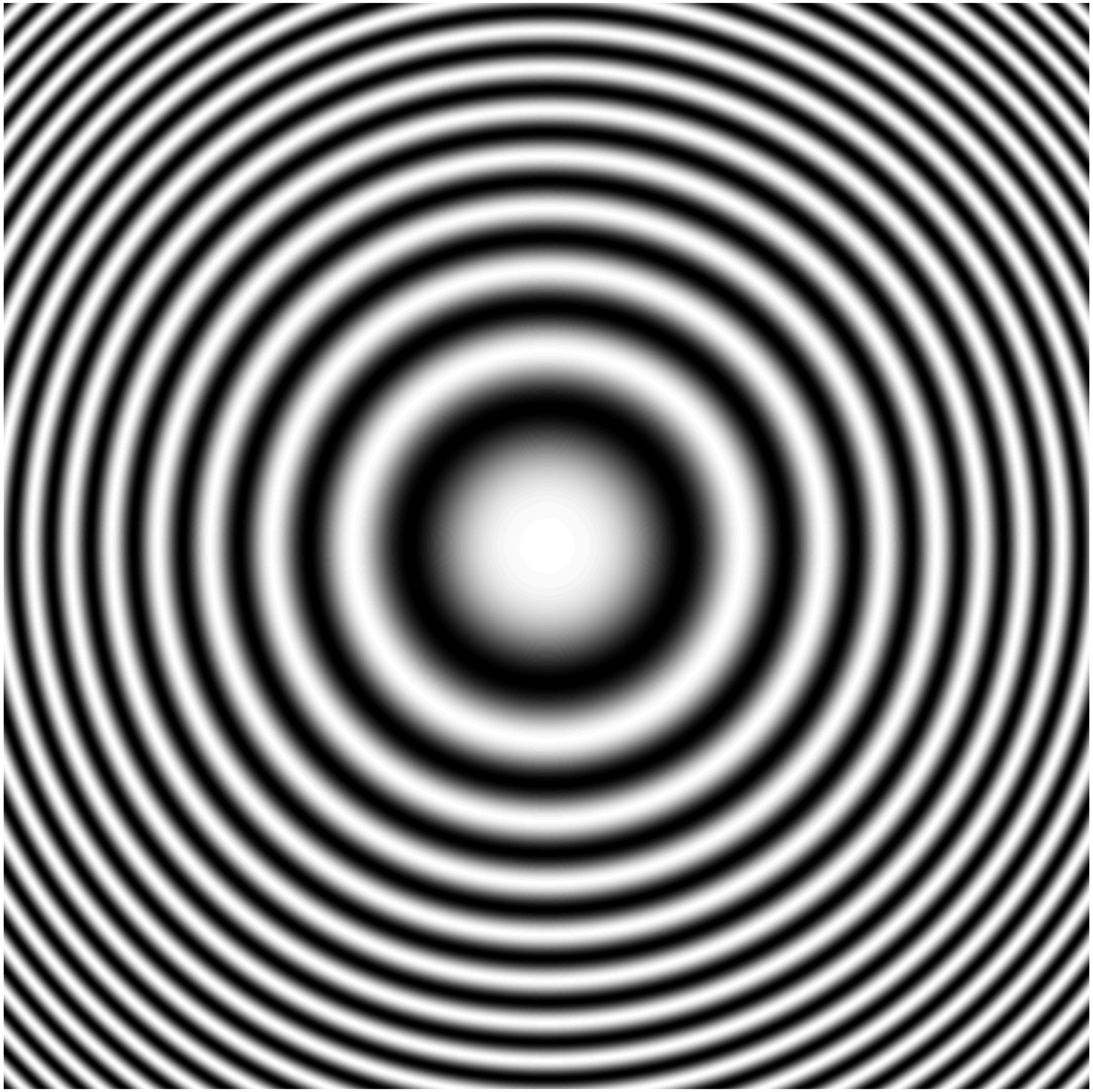
Undersampling: Overlaps/Aliases



Recall Shannon's Sampling Theorem


"the aliasing side" by FabriceNeyret2
https://www.shadertoy.com/view/4slXR4

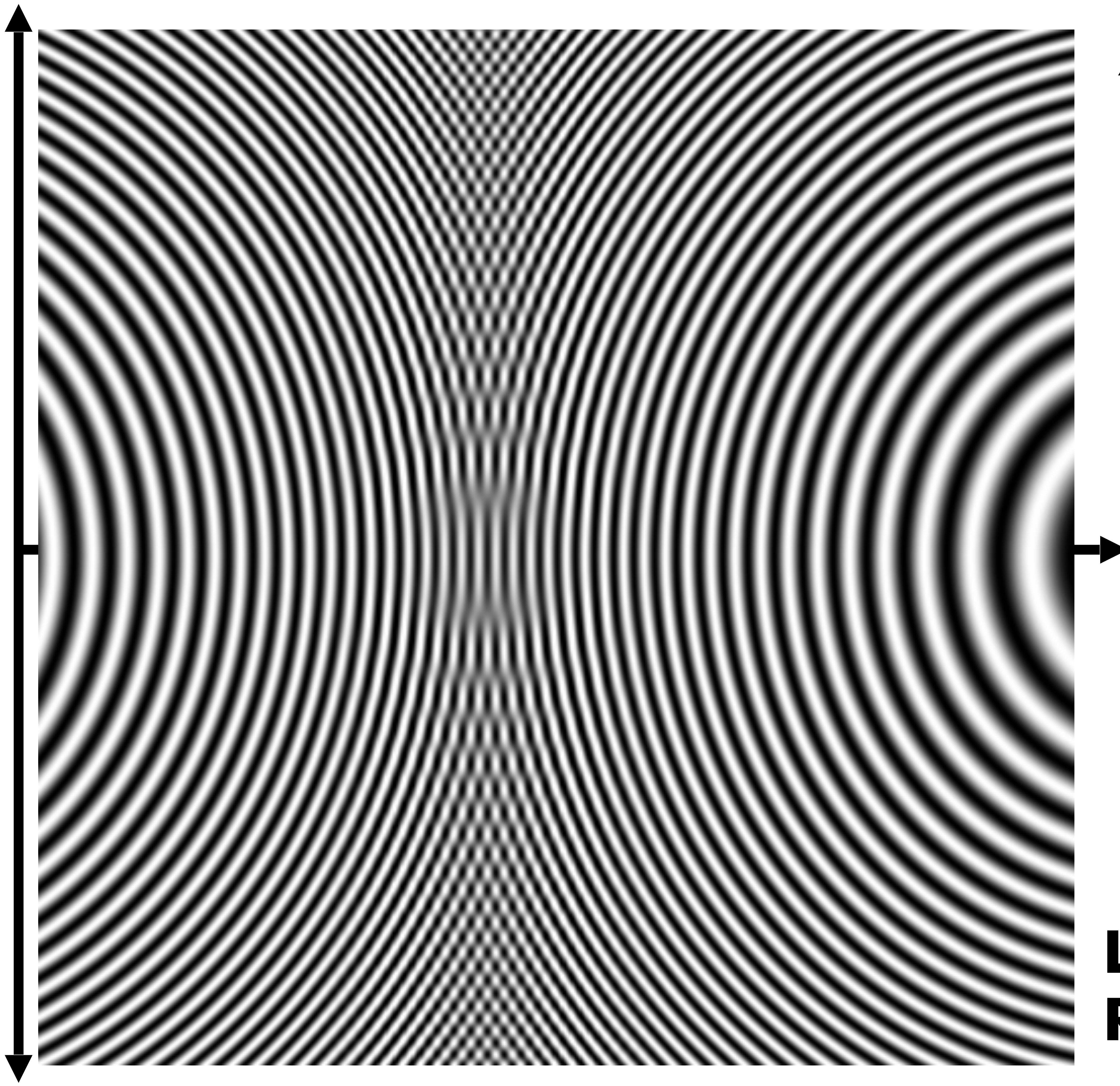
Sampling a "Zone Plate"



"A **zone plate** is a device used to focus light or other things exhibiting wave character. Unlike lenses or curved mirrors, zone plates use **diffraction** instead of refraction or reflection."

https://en.wikipedia.org/wiki/Zone_plate

Sampling a "Zone Plate"

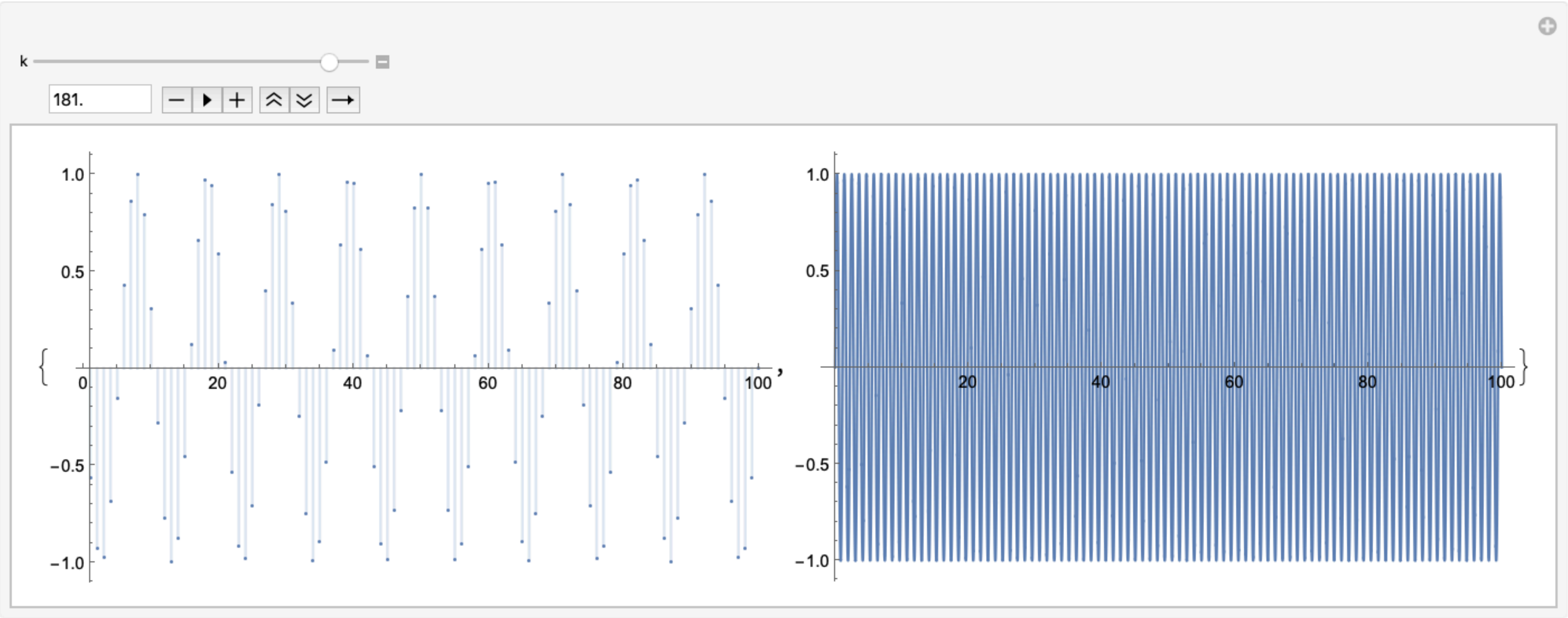


Zone plate:

$$\sin(x^2 + y^2)$$

Left rings: signal
Right rings: aliasing

Mathematica Demos (Aliasing)



Antialiasing

Preventing Aliasing

(pre-filtering and post-filtering)

Aliasing and Filtering Concepts

ALIASING can occur in multiple places:

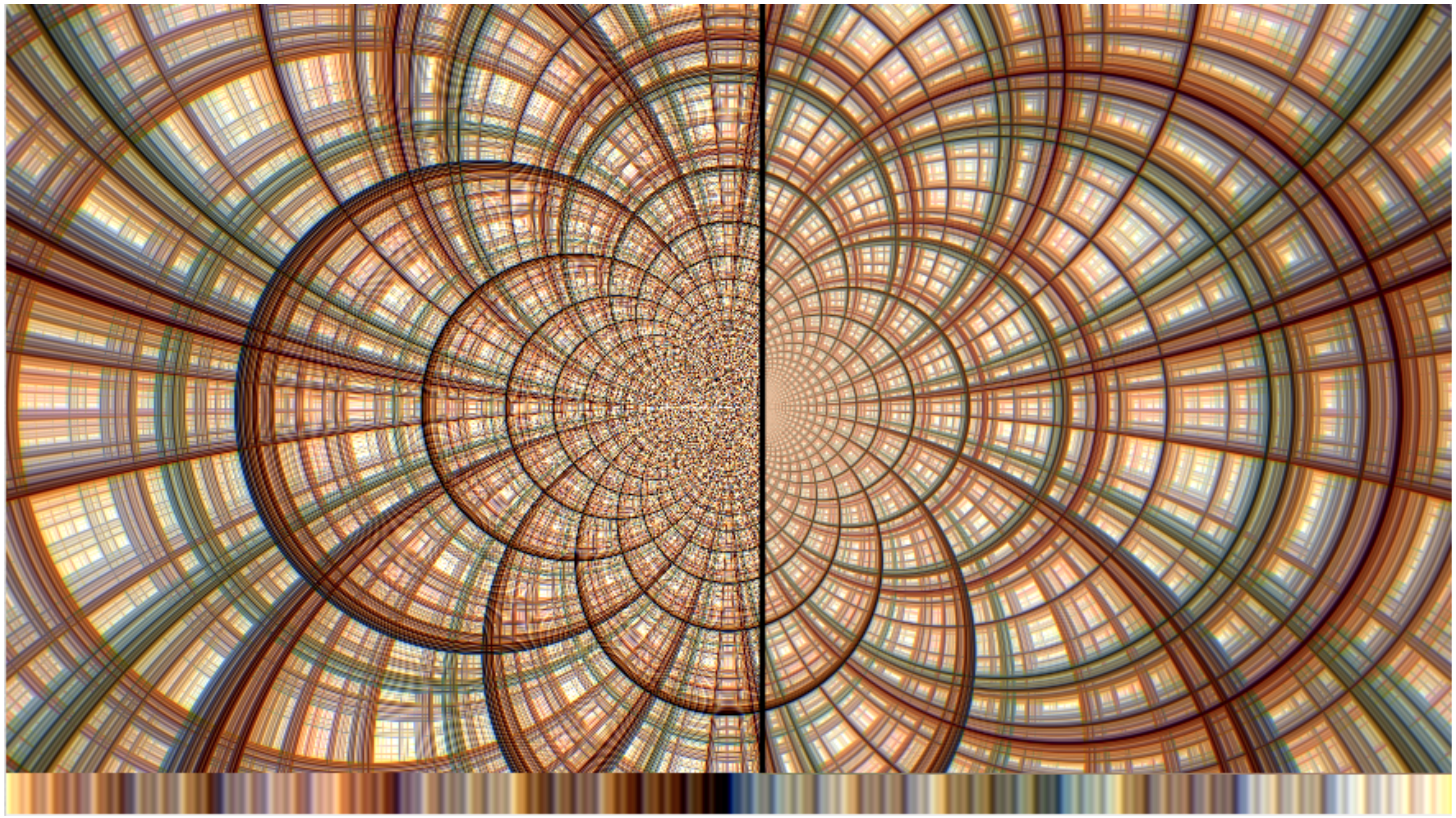
- **pre-aliasing: during sampling**
- **post-aliasing: during reconstruction**

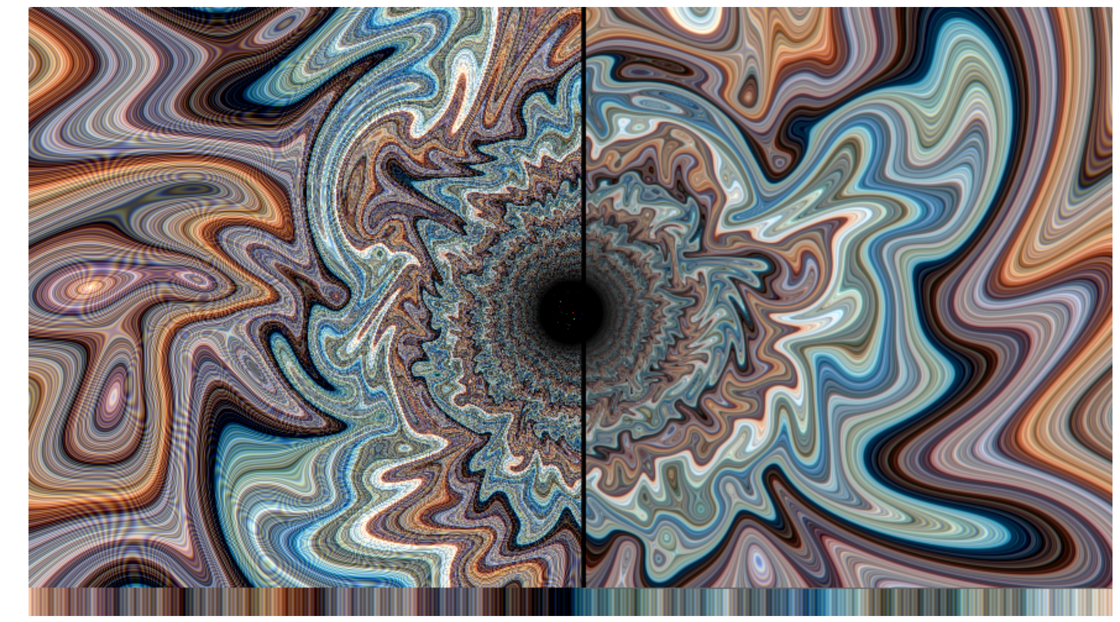
ANTI-ALIASING FILTERS (try to) reduce artifacts:

- **pre-filtering: exploit representation to reduce aliasing in samples *a priori***
  - **E.g., Exact area sampling, analytical filtering, texture filtering, etc.**
- **post-filtering: use samples to reduce aliasing *a posteriori***
  - **E.g., Supersampling**

Useful analytical **pre-filtering** trick to anti-alias sin(x) and cos(x)

Band-limited Synthesis (Inigo Quilez)





https://www.shadertoy.com/view/WtScDt

https://iquilezles.org/articles/bandlimiting/

Trick:

Average cos(x) over [x-h,x+h]

Simply replace

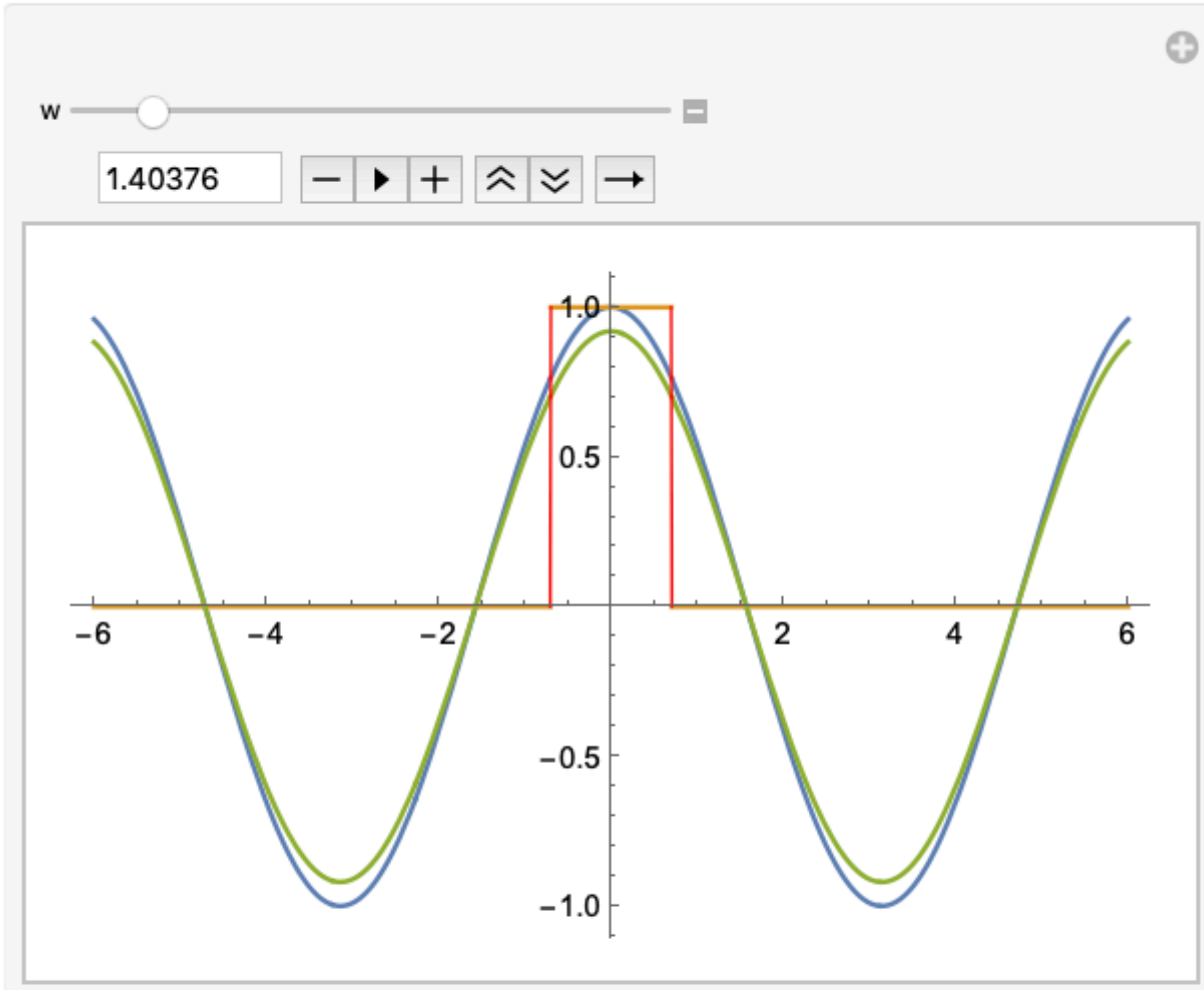
cos(x) → cos(x) sinc(h)

where h is the bin halfwidth.

Note: sinc(h) → 1 as h→0.

$$\frac{1}{w}\int_{x-\frac{w}{2}}^{x+\frac{w}{2}} \cos(t)dt = \frac{1}{w}\left[\sin\left(x+\frac{w}{2}\right) - \sin\left(x-\frac{w}{2}\right)\right] = \cos(x)\frac{\sin\left(\frac{w}{2}\right)}{\frac{w}{2}} = \cos(x)\,\mathrm{sinc}\left(\frac{w}{2\pi}\right)$$

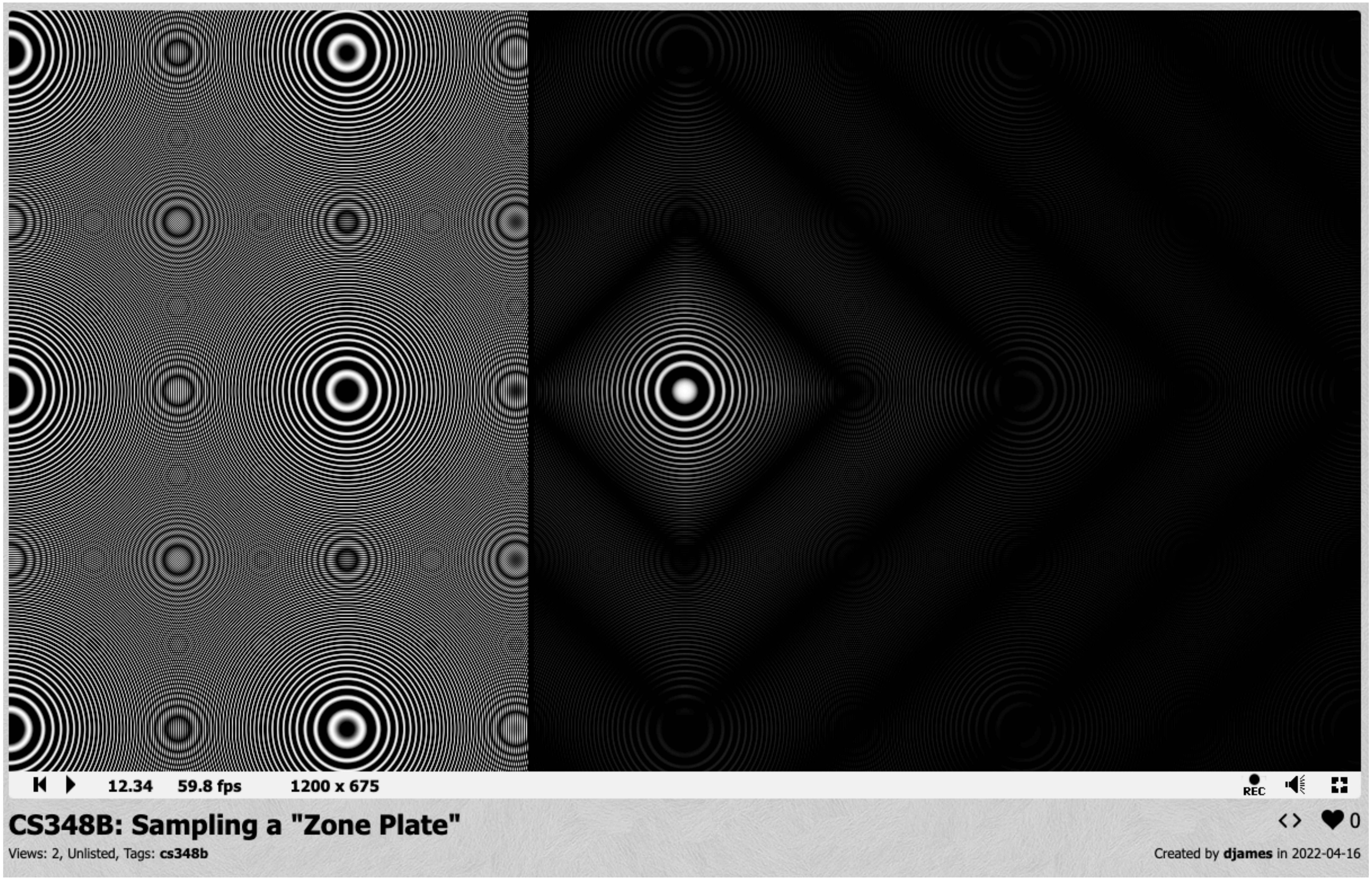
Mathematica Demo
Band-limited Synthesis



$$\frac{1}{w}\int_{x-\frac{w}{2}}^{x+\frac{w}{2}}\cos(t)dt = \frac{1}{w}\left[\sin\left(x+\frac{w}{2}\right)-\sin\left(x-\frac{w}{2}\right)\right] = \cos(x)\frac{\sin\left(\frac{w}{2}\right)}{\frac{w}{2}} = \cos(x)\,\mathrm{sinc}\left(\frac{w}{2\pi}\right)$$

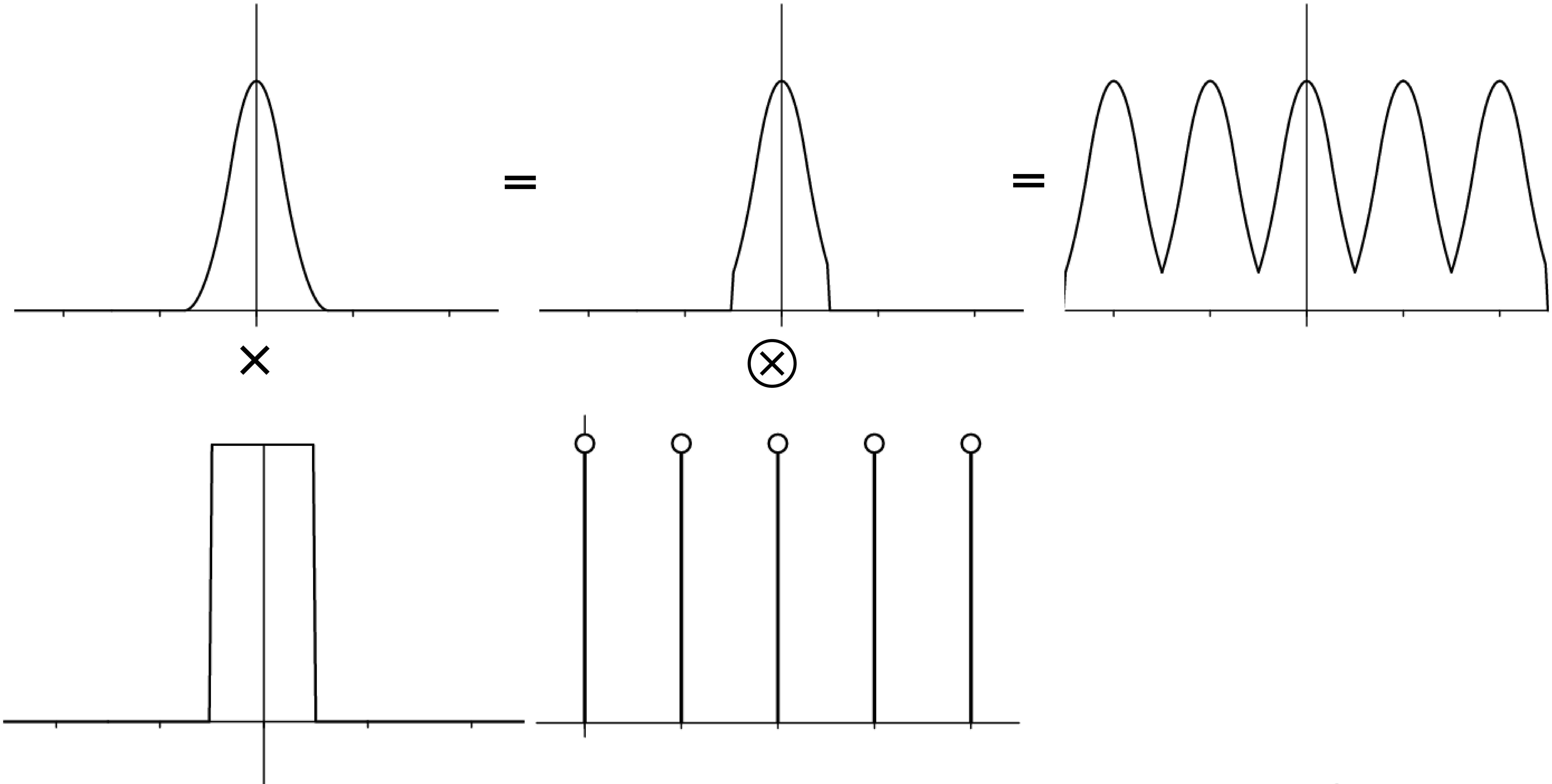
ShaderToy Pre-filtering Demo

Revisited: "Sampling a Zone Plate"



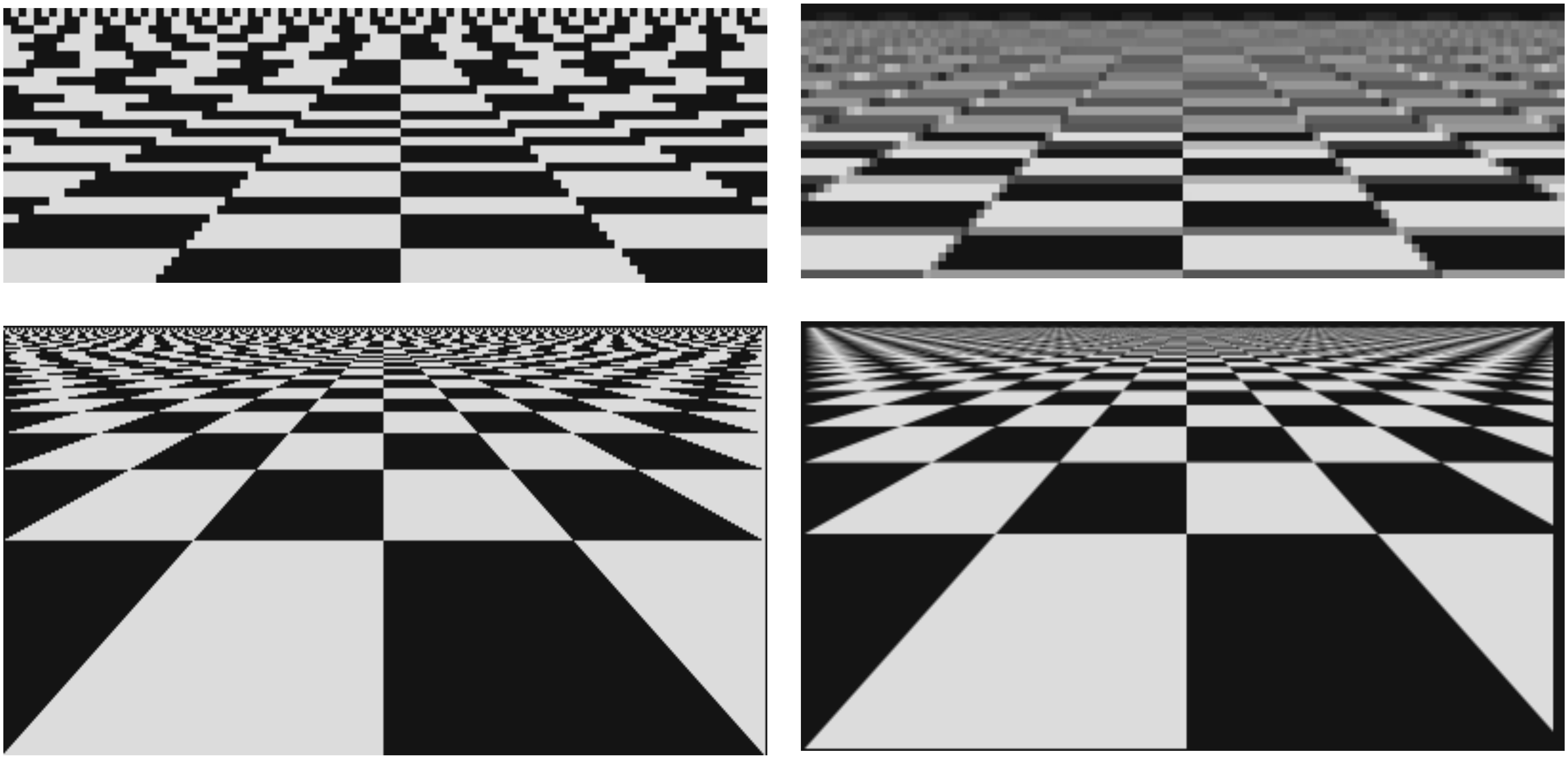
https://www.shadertoy.com/view/7ljczt

Antialiasing by Pre-filtering



Frequency Space

Point vs. Area Sampling



Point **Exact Area**

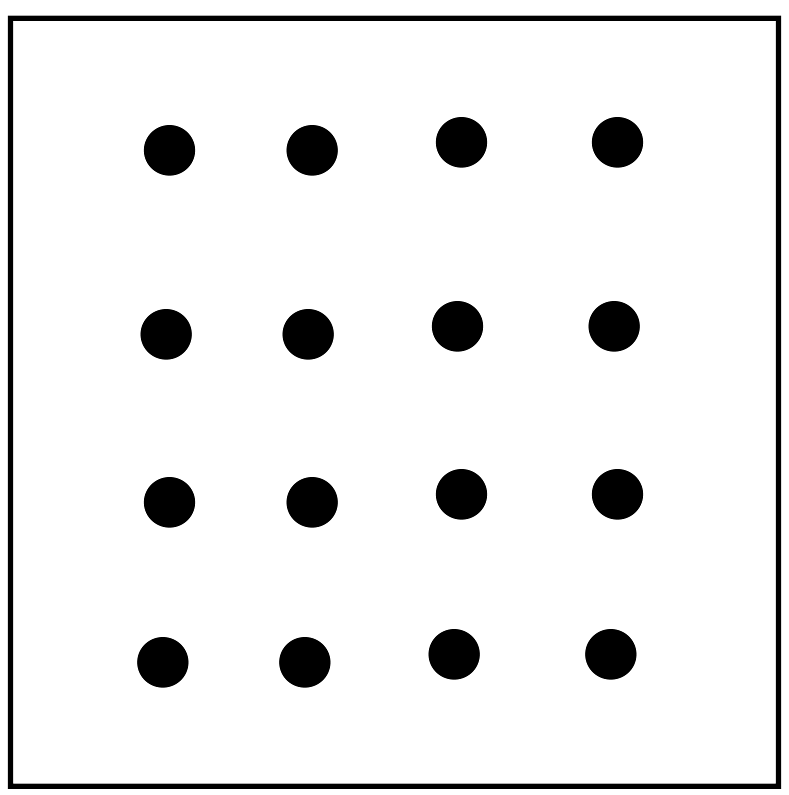
Checkerboard sequence by Tom Duff

Simple post-filtering

Uniform Supersampling

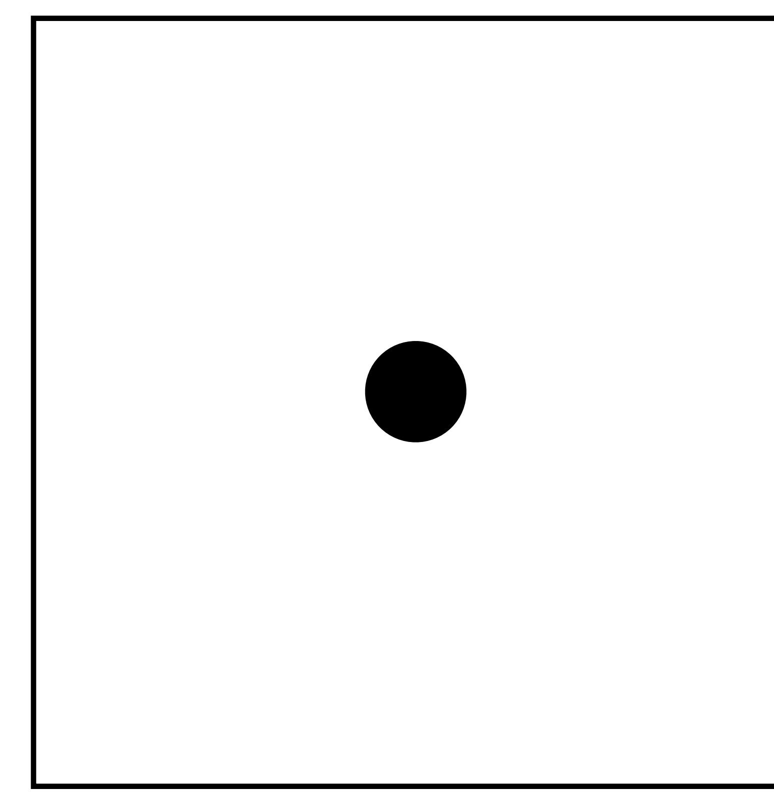
Take *n* samples

Average them together (filter = weighted average)



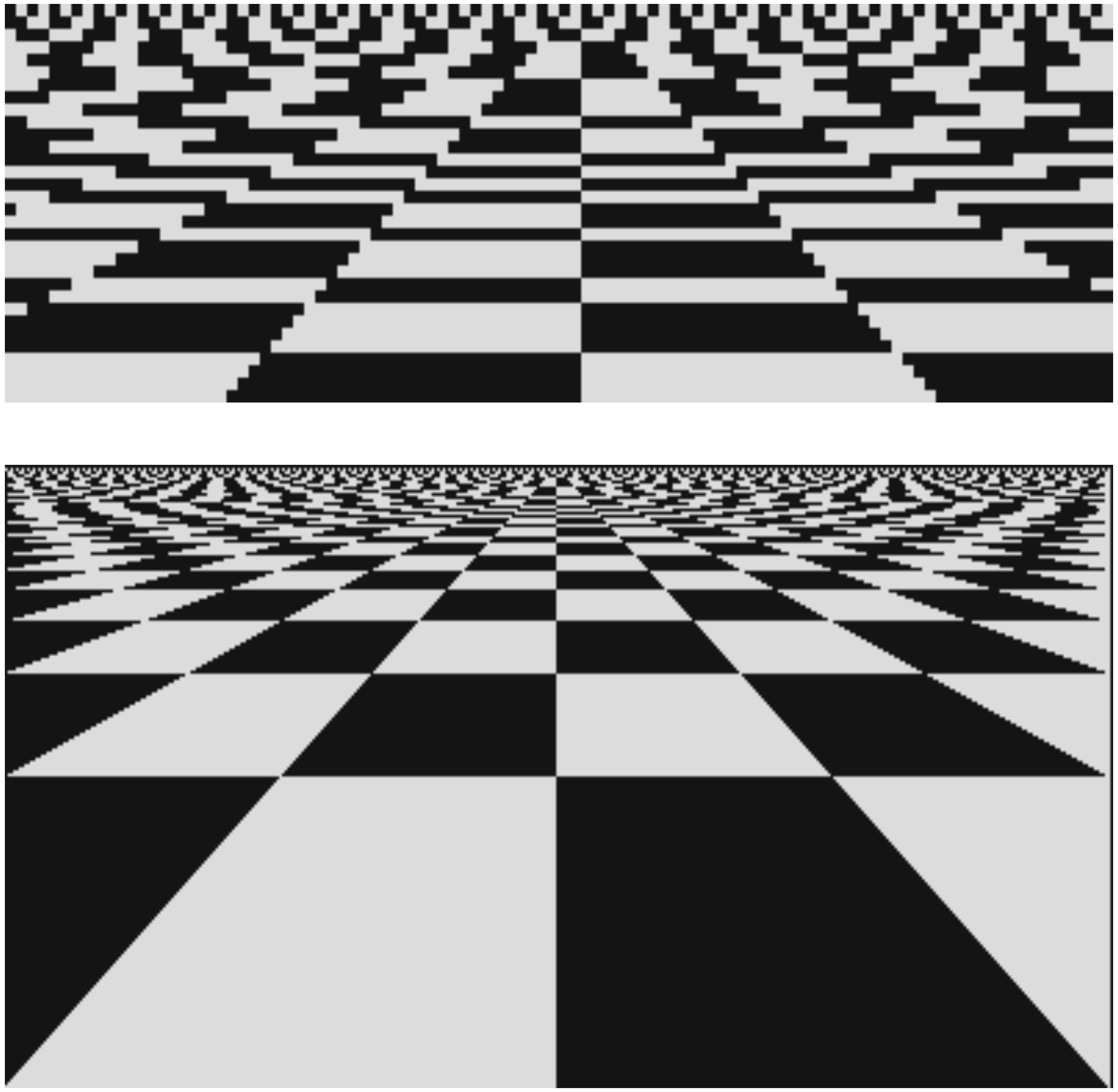
Samples

$$Pixel = \sum_{s} w_s \cdot Sample_s$$

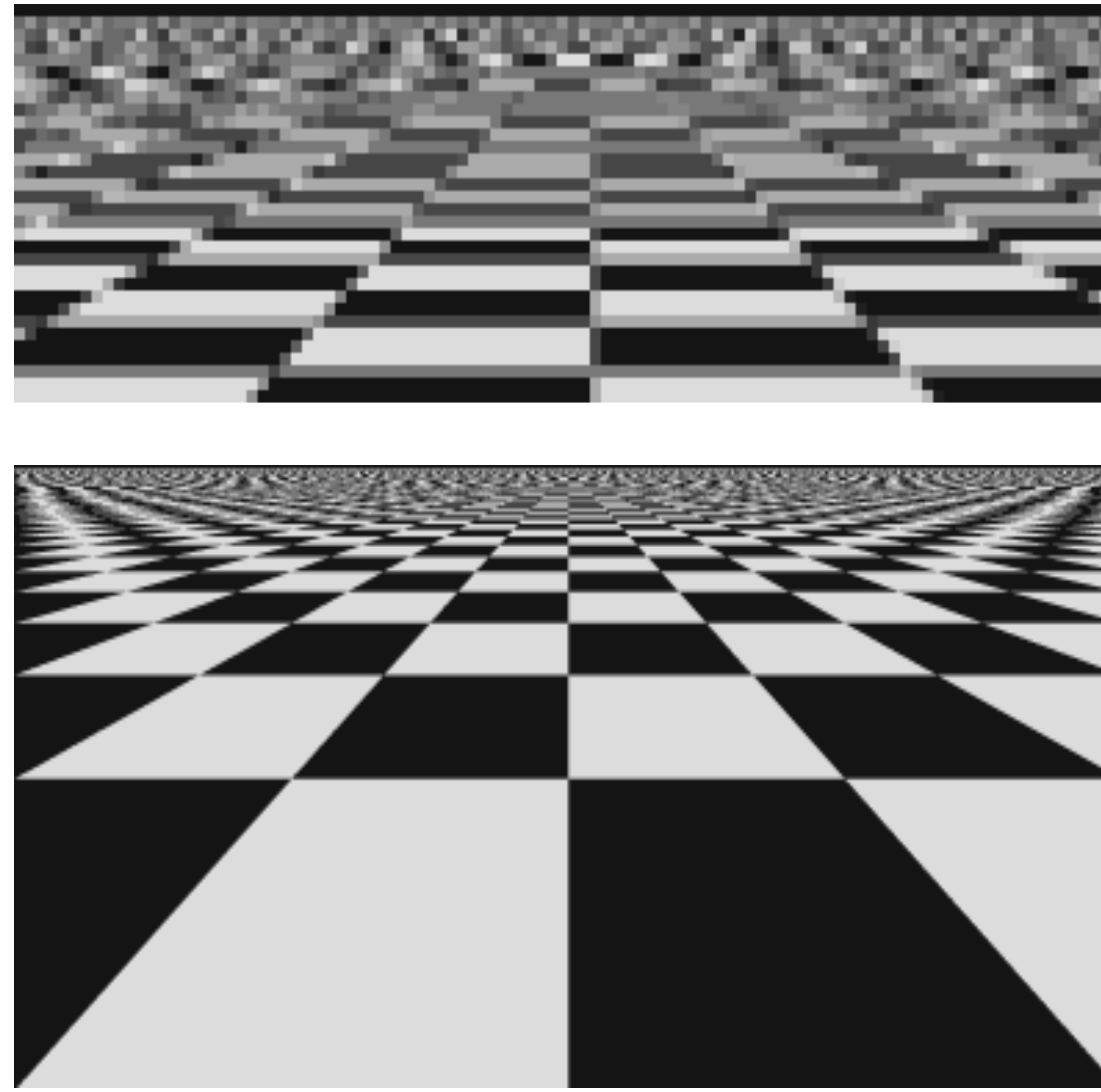


Pixel

Point vs. Supersampled



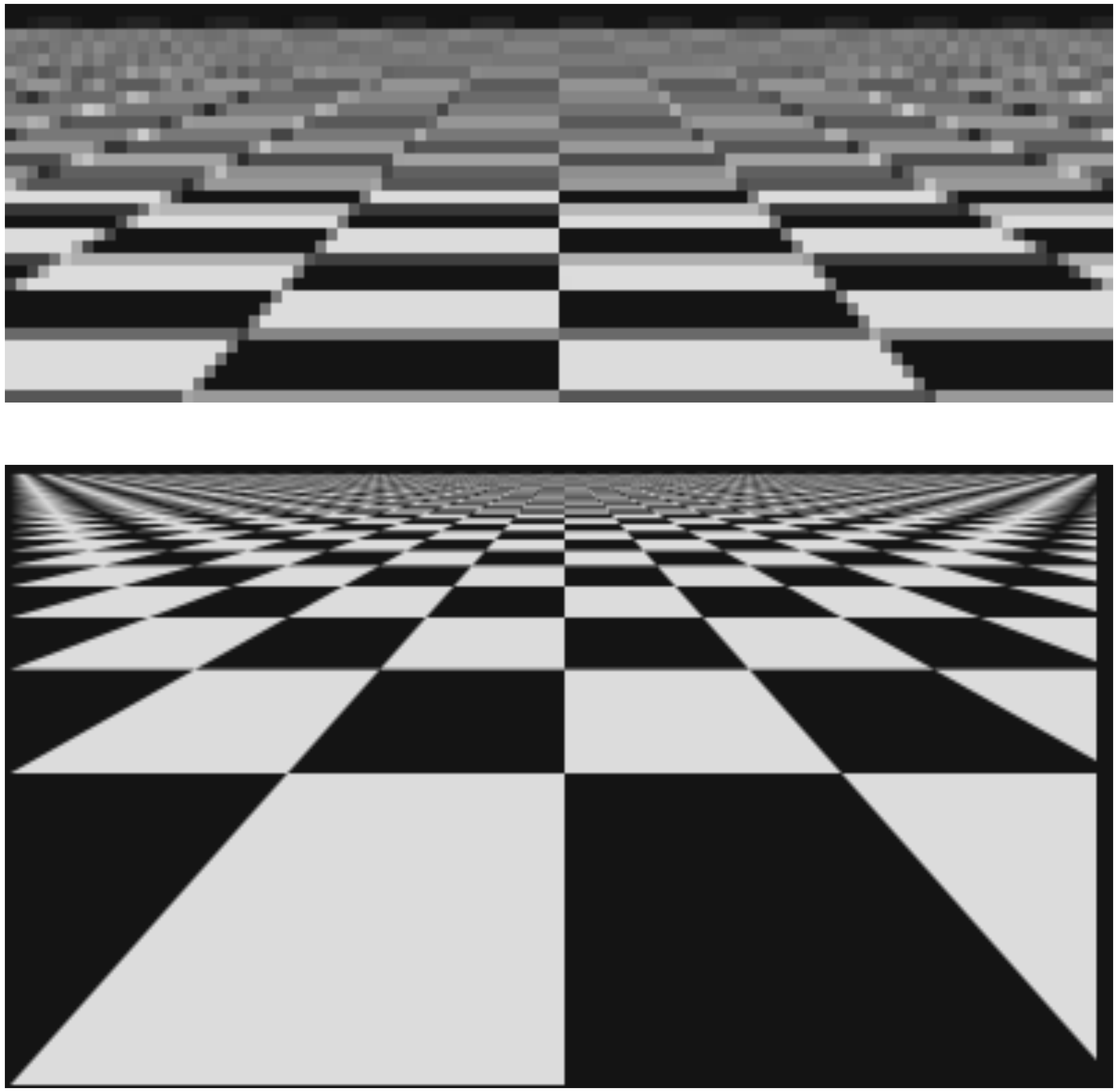
Point



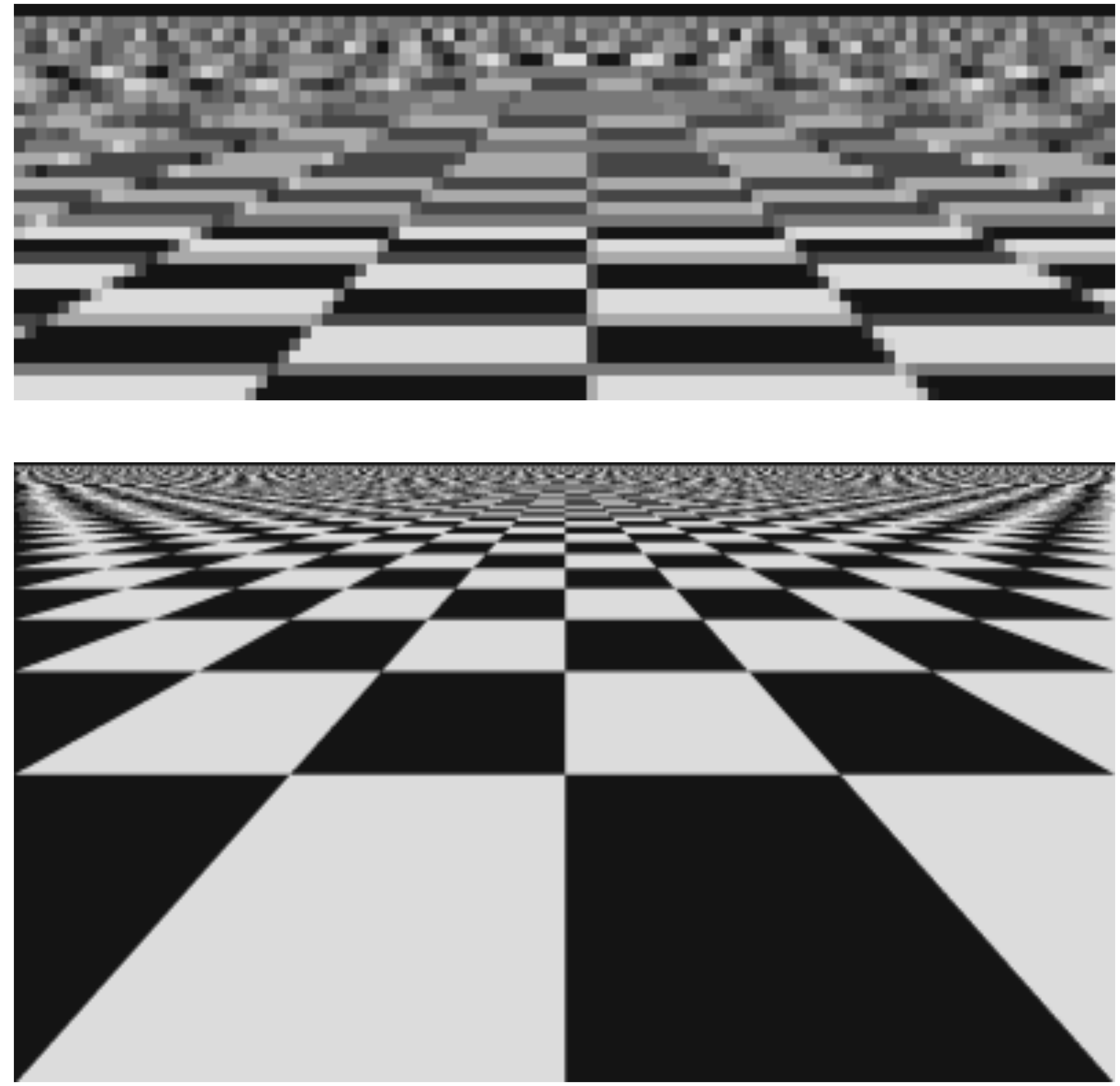
4x4 Uniform

Fewer aliases, but still present

Area vs. Supersampled



Exact Area



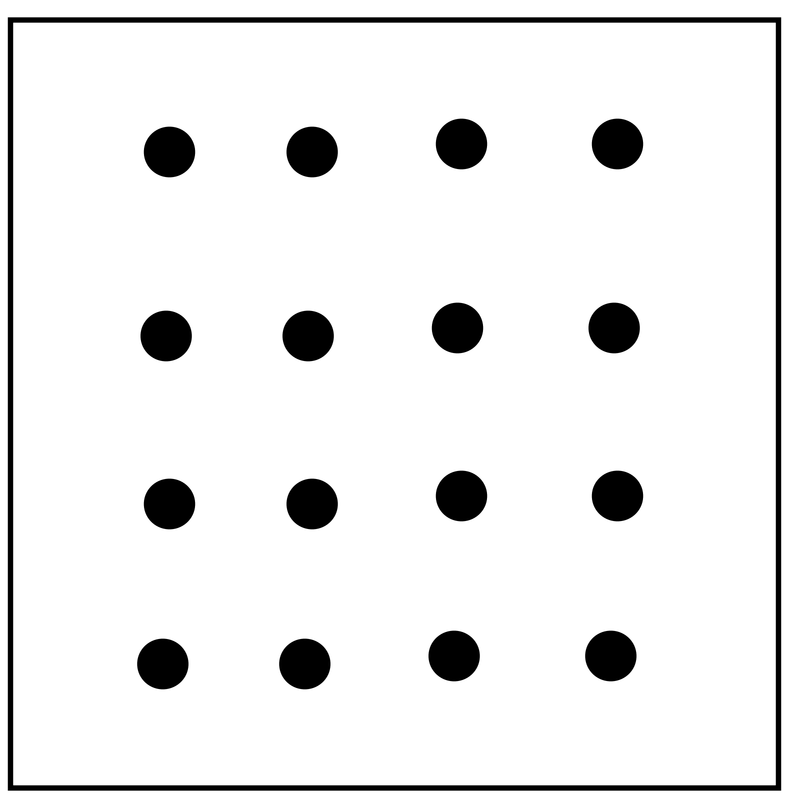
4x4 Uniform

Fewer aliases, but still present

Uniform Supersampling

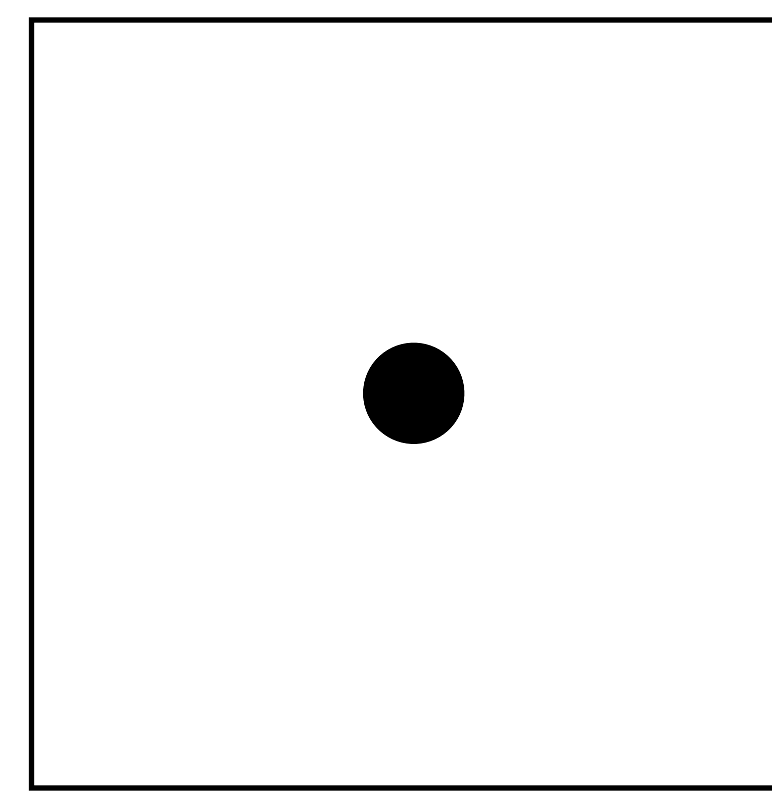
Increasing the number of samples moves each copy of the spectra further apart, thus there is less overlap

This reduces, but does not eliminate, aliasing



Samples

$$Pixel = \sum_s w_s \cdot Sample_s$$



Pixel

Blue Noise - Motivation

- **Yellott observed that monkey retina exhibits *blue noise* pattern**

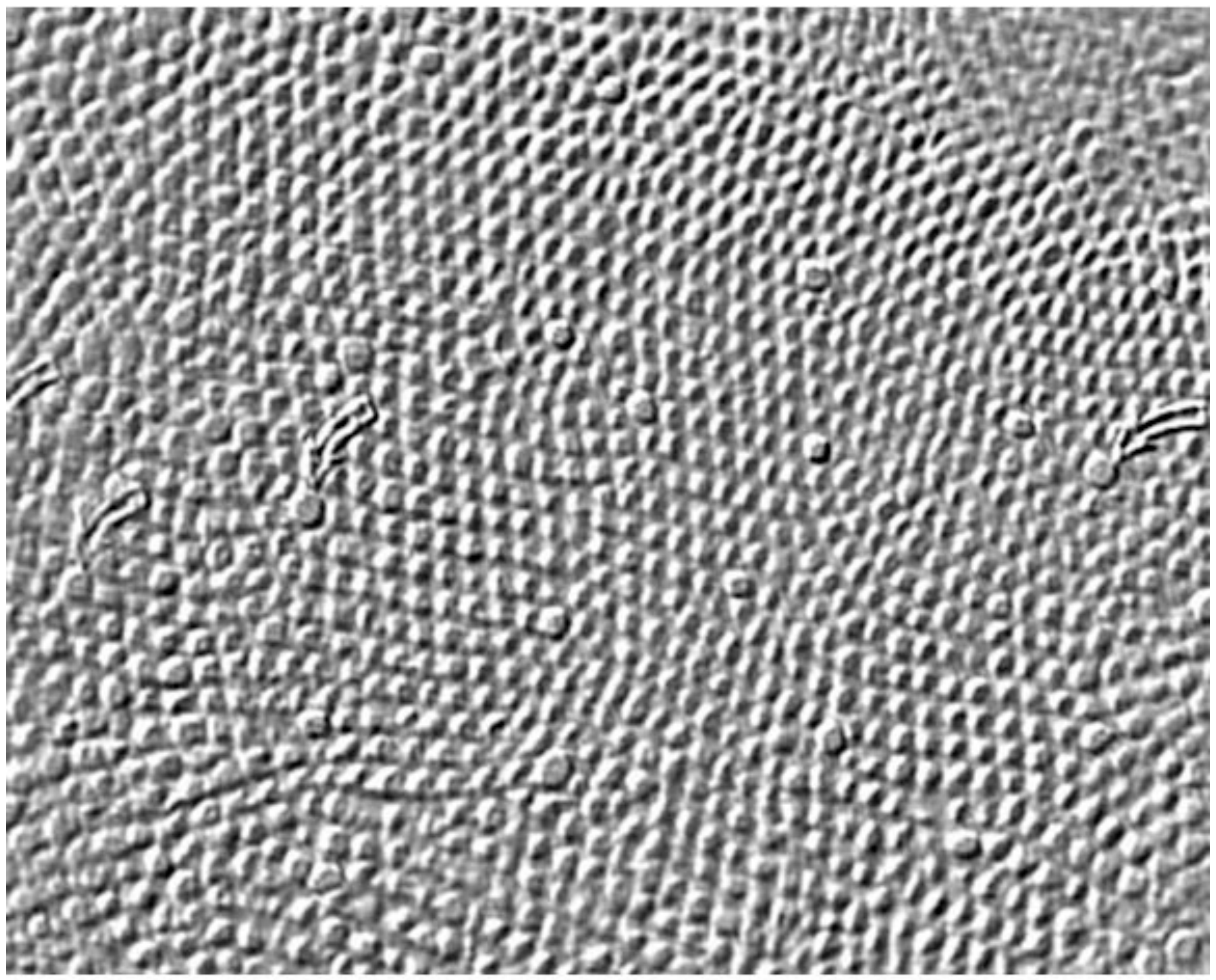
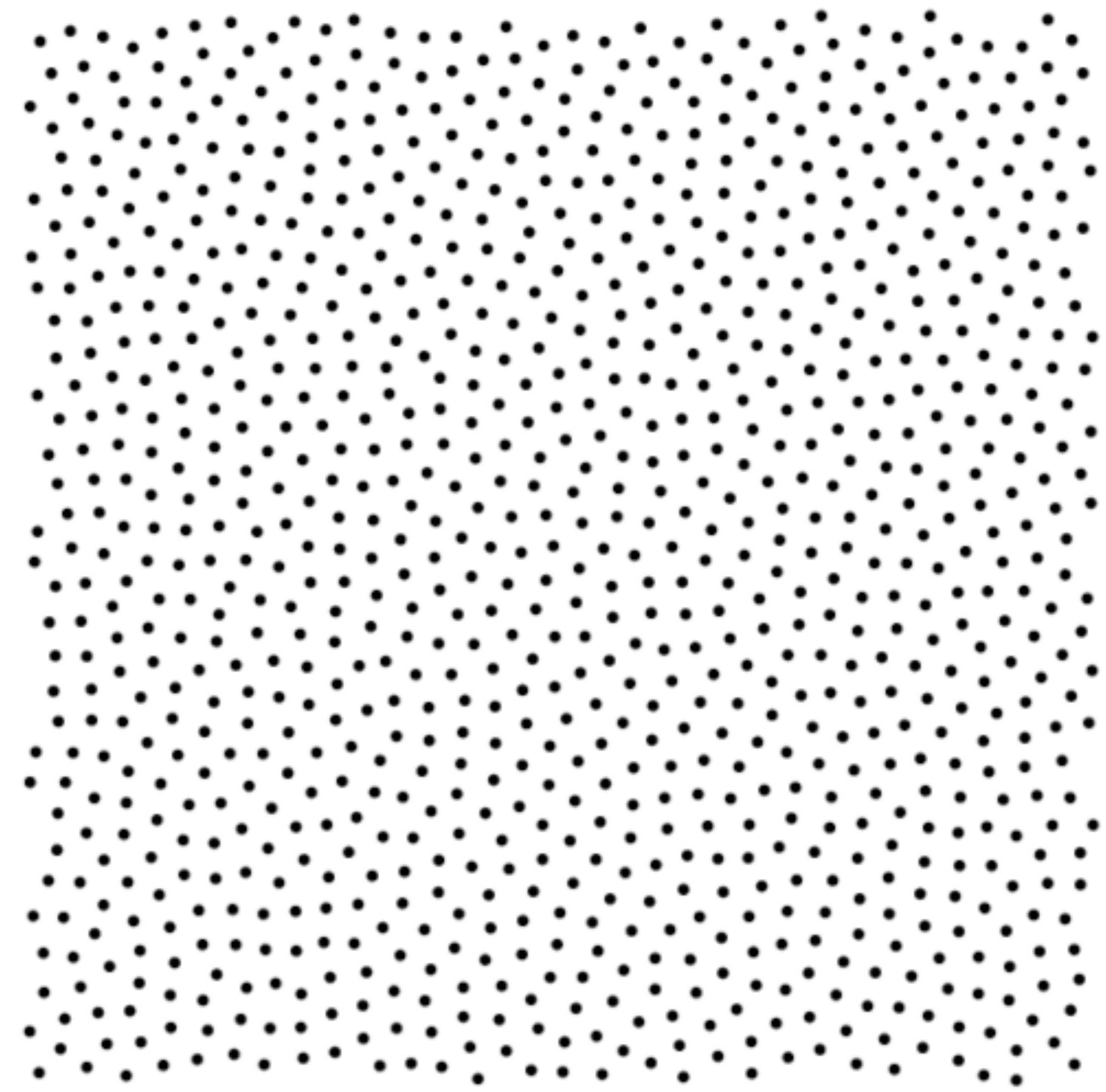


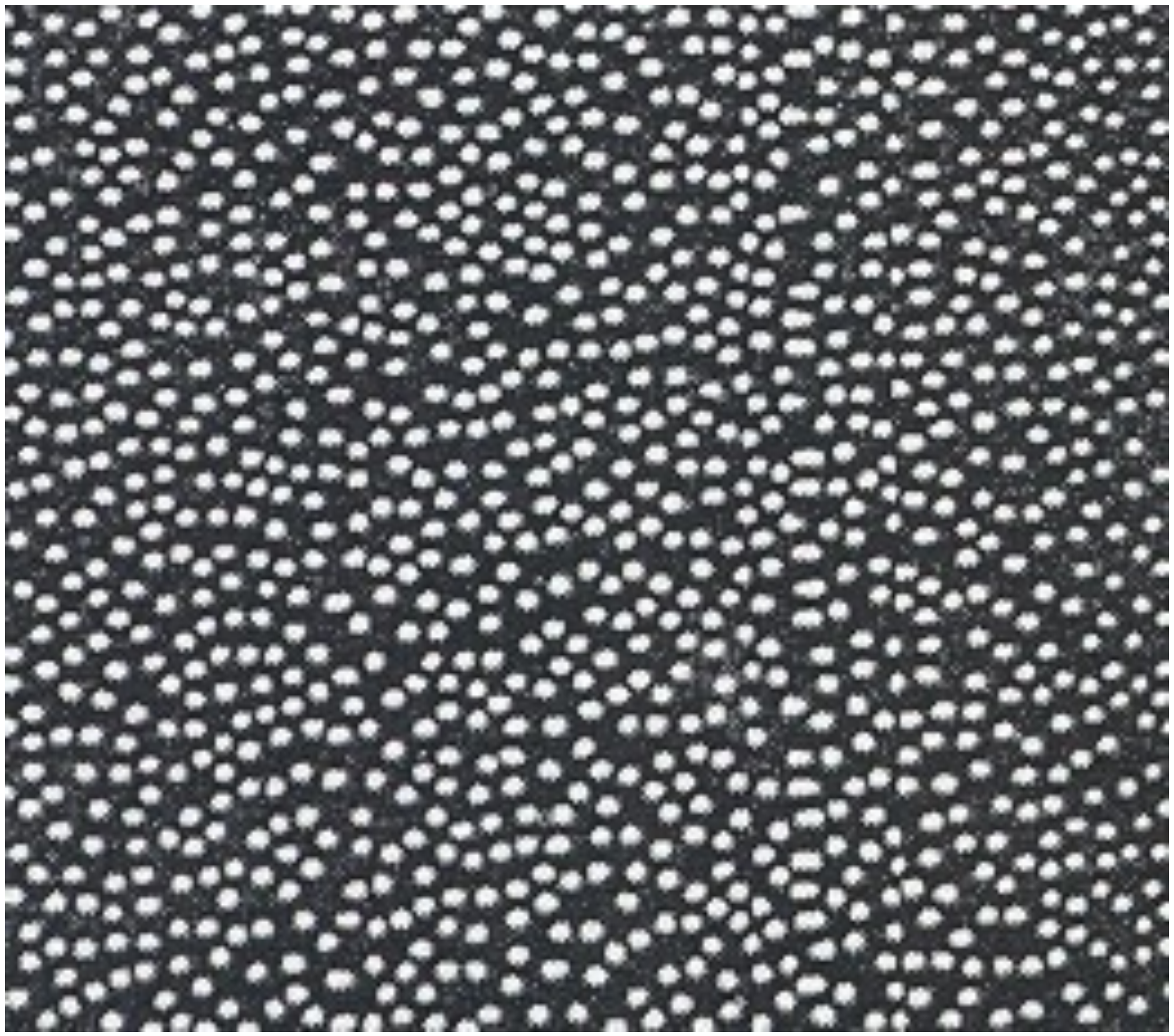
Fig. 13. Tangential section through the human fovea. Larger cones (arrows) are blue cones. From Ahnelt et al. 1987.



“blue noise”

- **No obvious preferred directions (anisotropic)**
- **What about frequencies?**

Distribution of Extrafoveal Cones



Monkey eye cone distribution

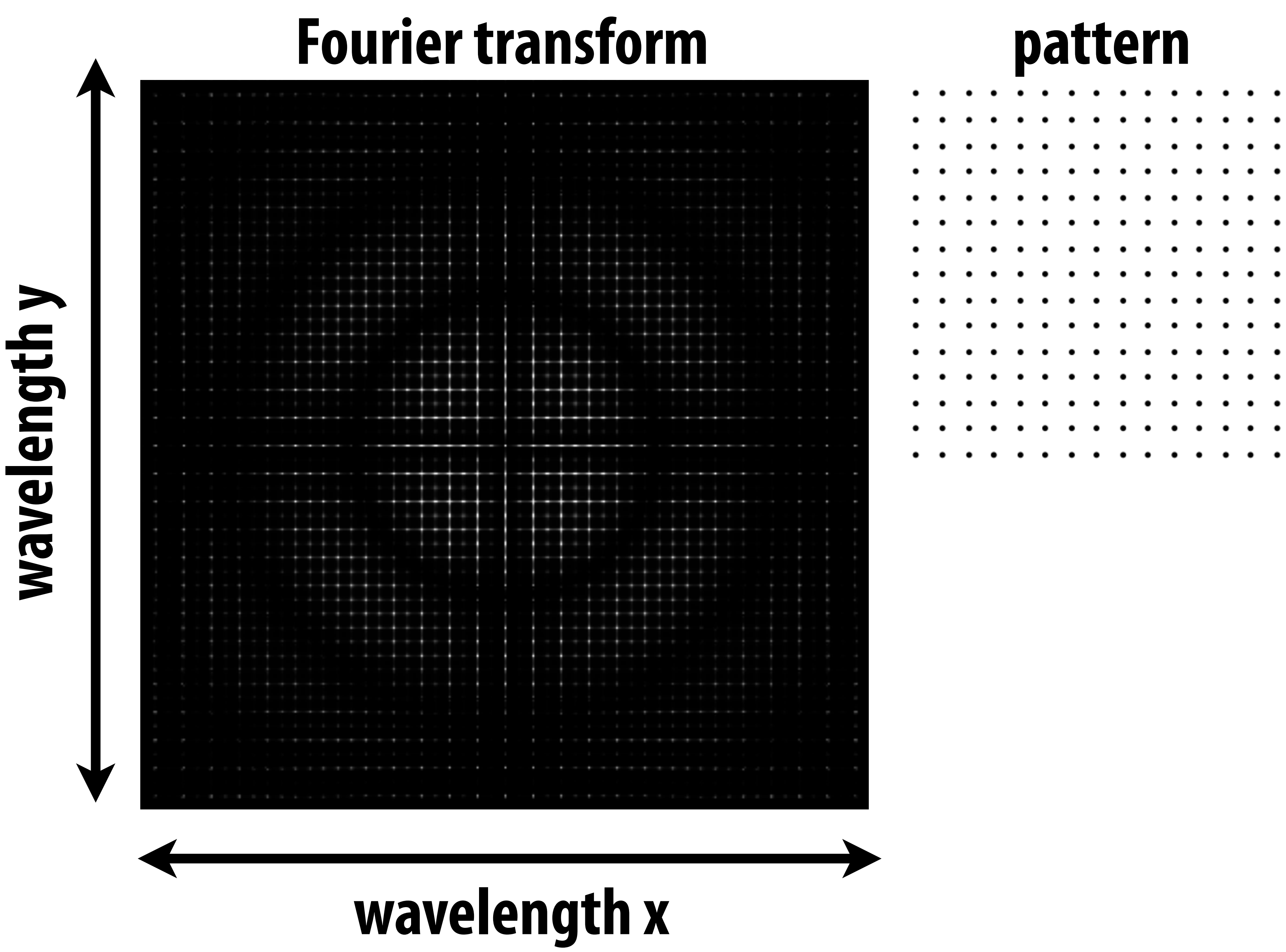
Fourier transform

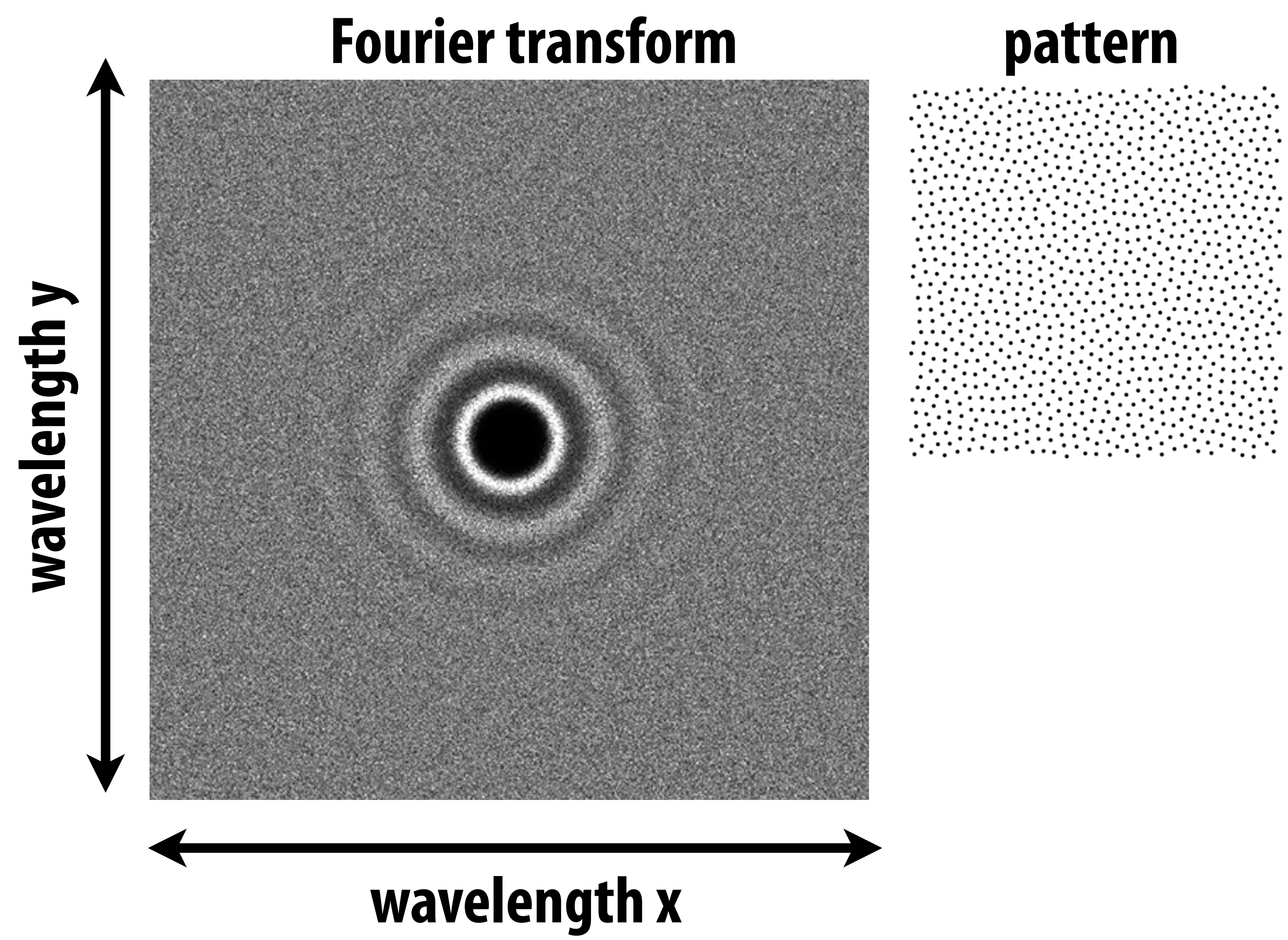
Yellott theory (1983)

- Aliases replaced by high-freq noise
- Visual system less sensitive to high-freq noise

Blue Noise - Fourier Transform

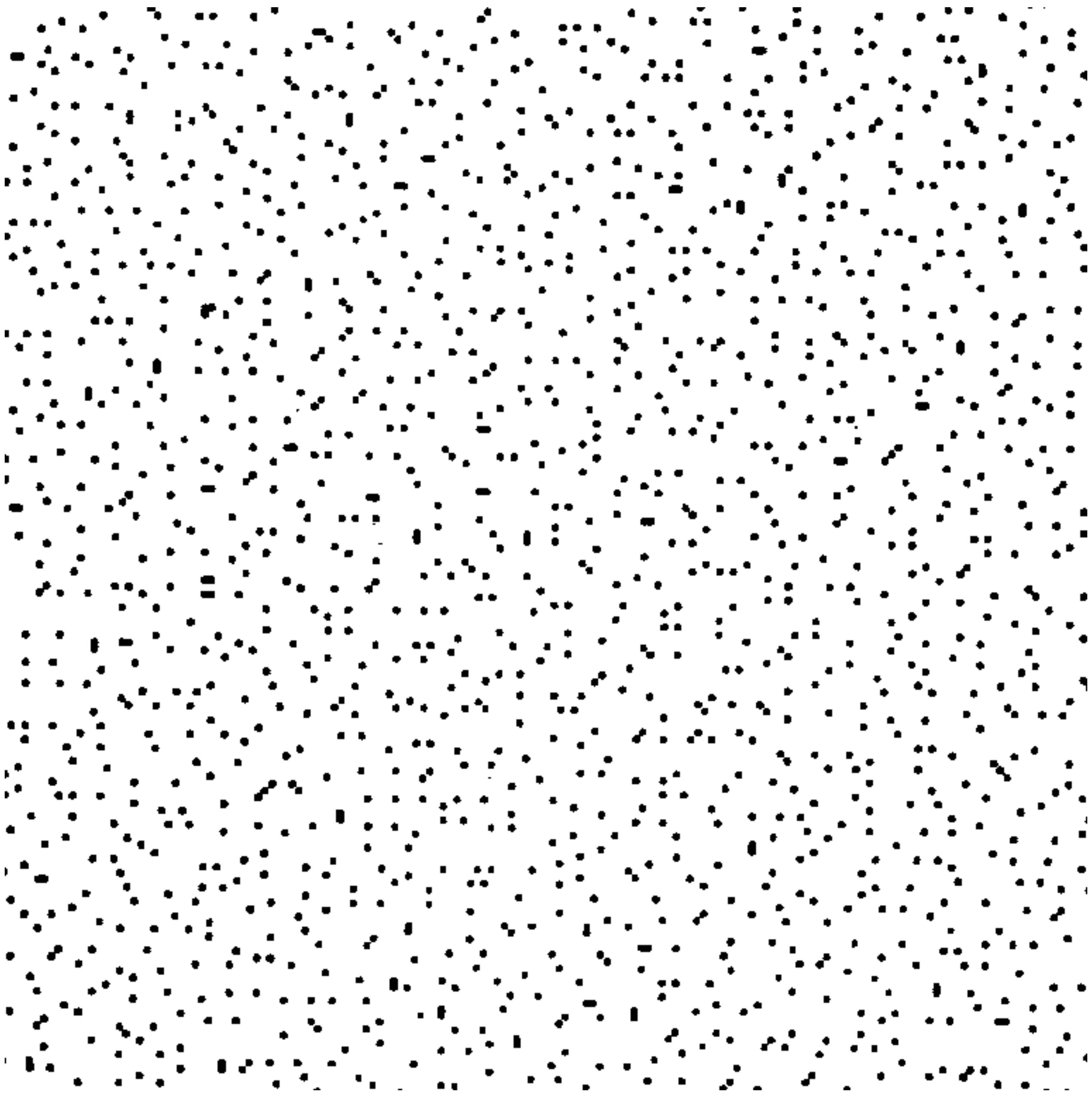
- **Can analyze quality of a sample pattern in *Fourier domain***

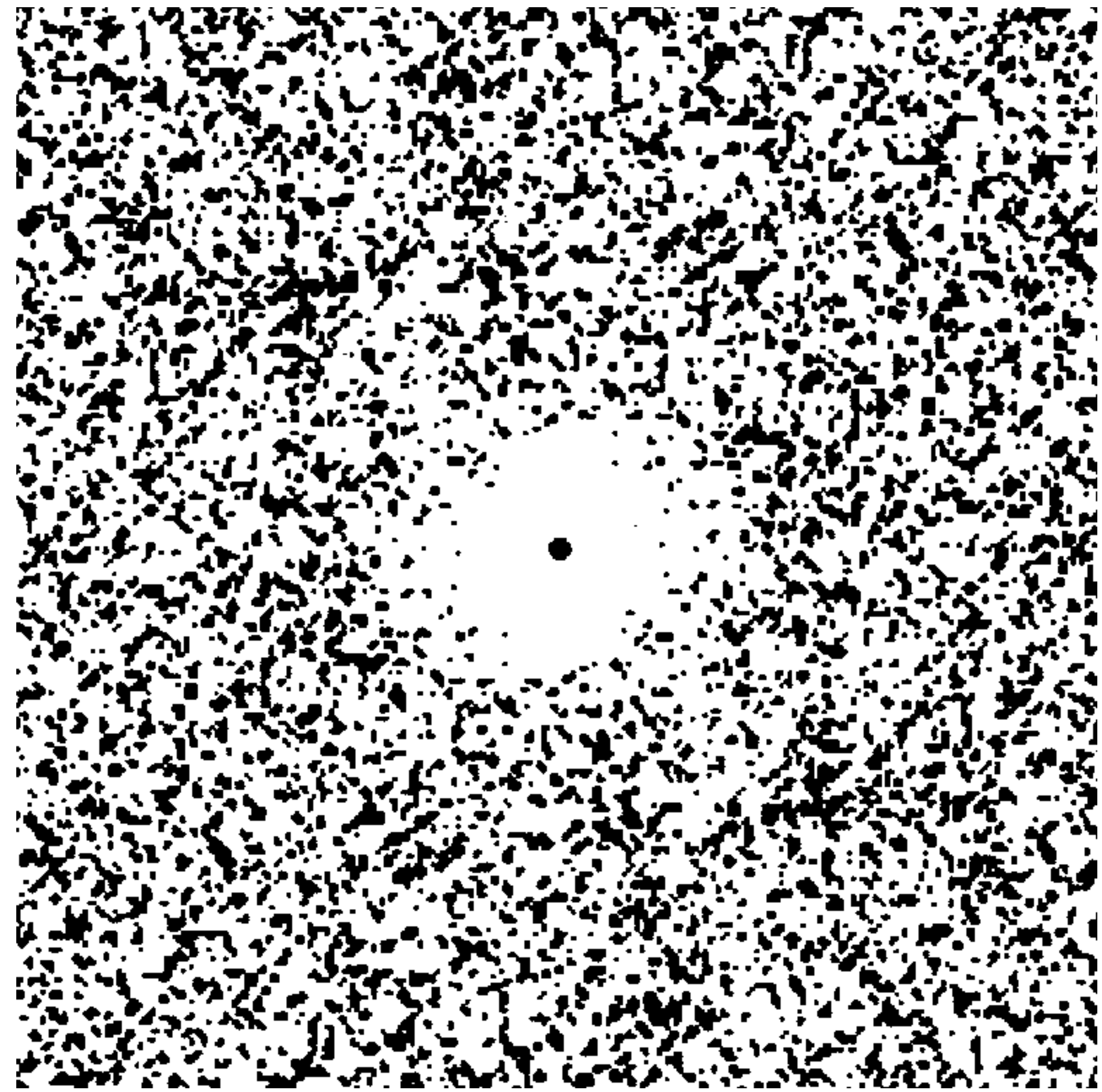




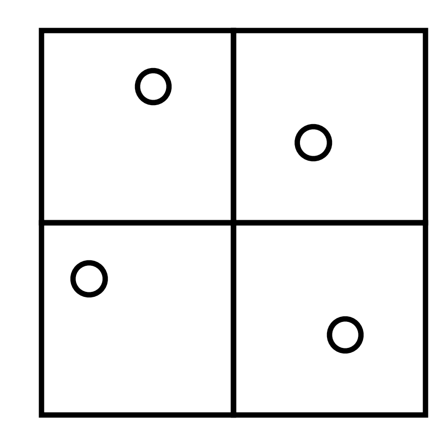
- **Regular pattern has "spikes" at regular intervals**
- **Blue noise is spread evenly over higher frequencies in all directions, with less at low frequencies**
- **bright center "ring" corresponds to sample spacing**

Jittered Sampling

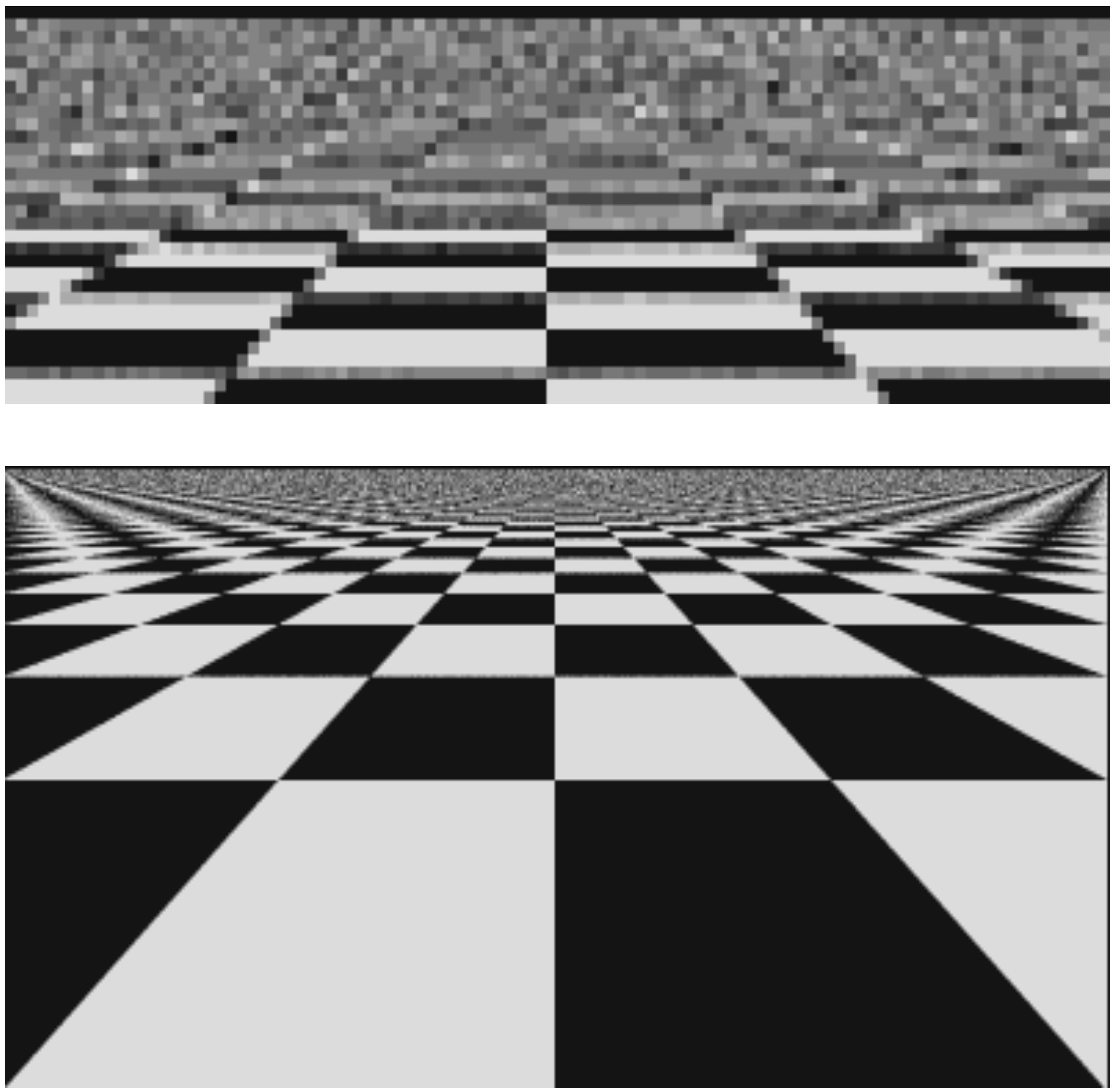




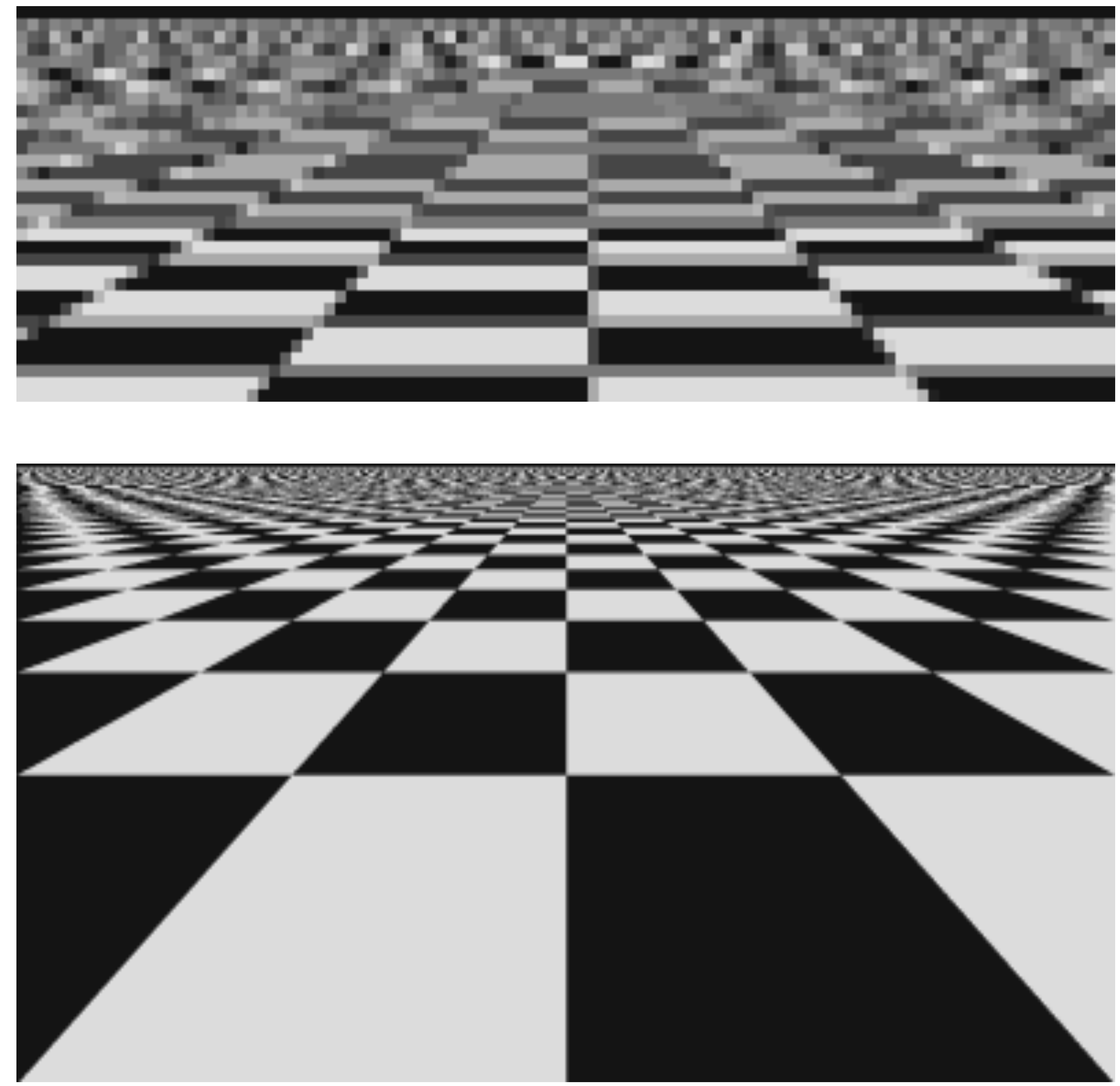
Add uniform random jitter to each sample



Jittered vs. Uniform Supersampling



4x4 Jittered Sampling



4x4 Uniform

Theory: Analysis of Jitter

Non-uniform sampling

$$s(x) = \sum_{n=-\infty}^{n=\infty} \delta\,(x - x_n)$$

$$x_n = nT + j_n$$

Jittered sampling

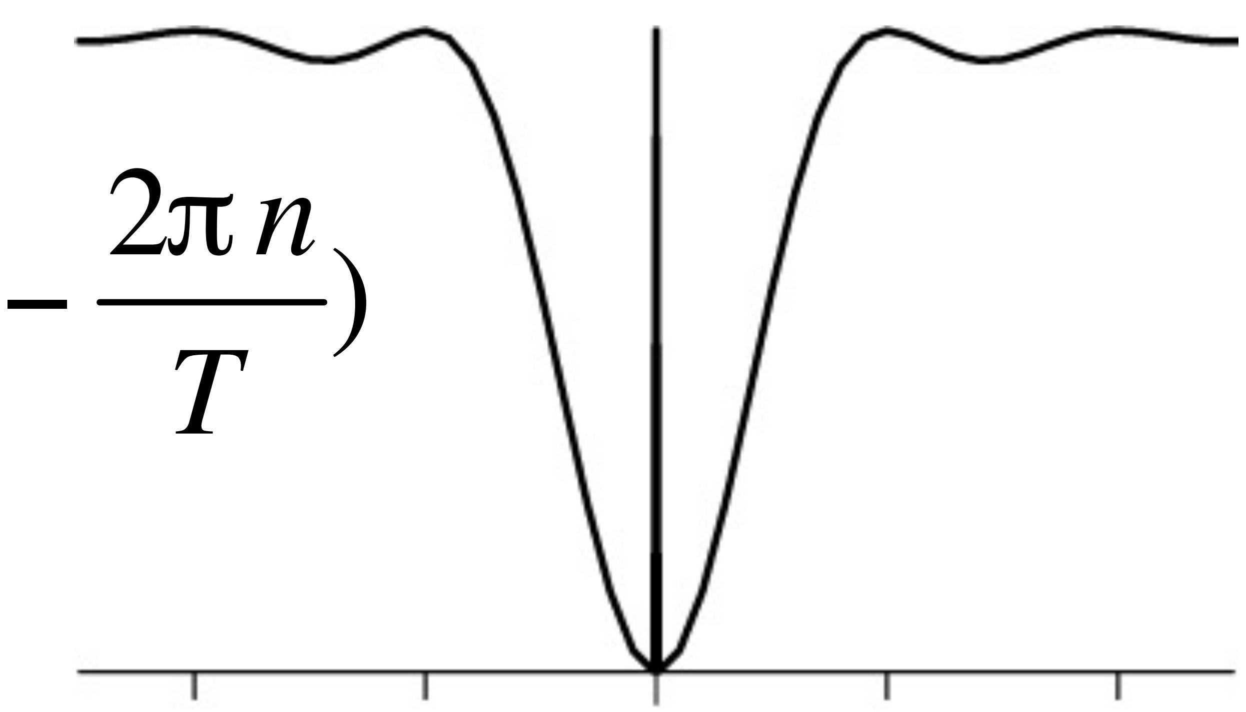
$$j_n \sim j(x)$$

$$j(x) = \begin{cases} 1 & |x| \le 1/2 \\ 0 & |x| > 1/2 \end{cases}$$

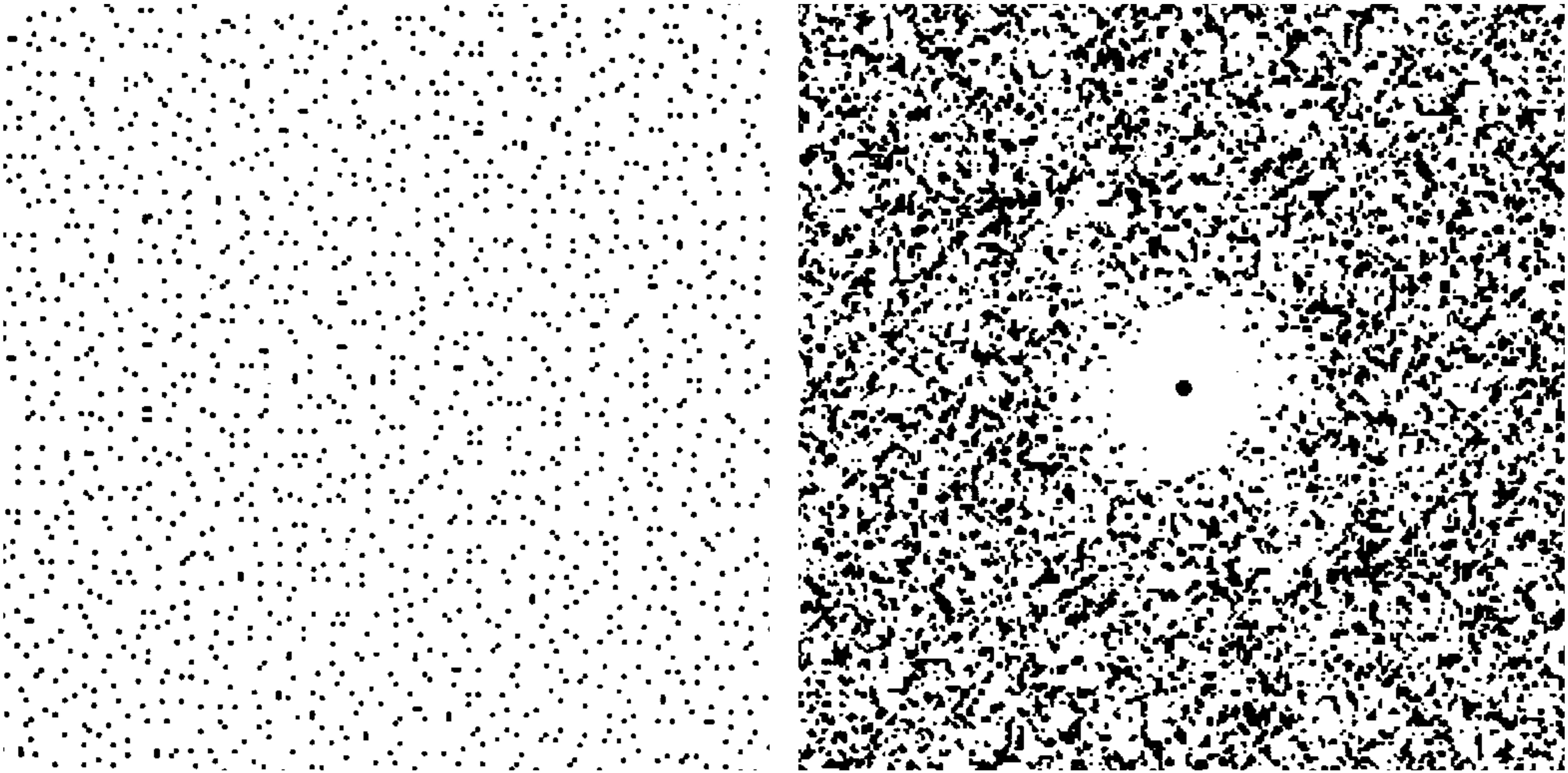
$$J(\omega) = \operatorname{sinc}\omega$$

$$S(\omega) = \frac{1}{T}\left[1 - |J(\omega)|^2\right] + \frac{2\pi}{T^2}|J(\omega)|^2 \sum_{n=-\infty}^{n=-\infty} \delta\,(\omega - \frac{2\pi n}{T})$$

$$= \frac{1}{T}\left[1 - \operatorname{sinc}^2 \omega\right] + \delta\,(\omega)$$



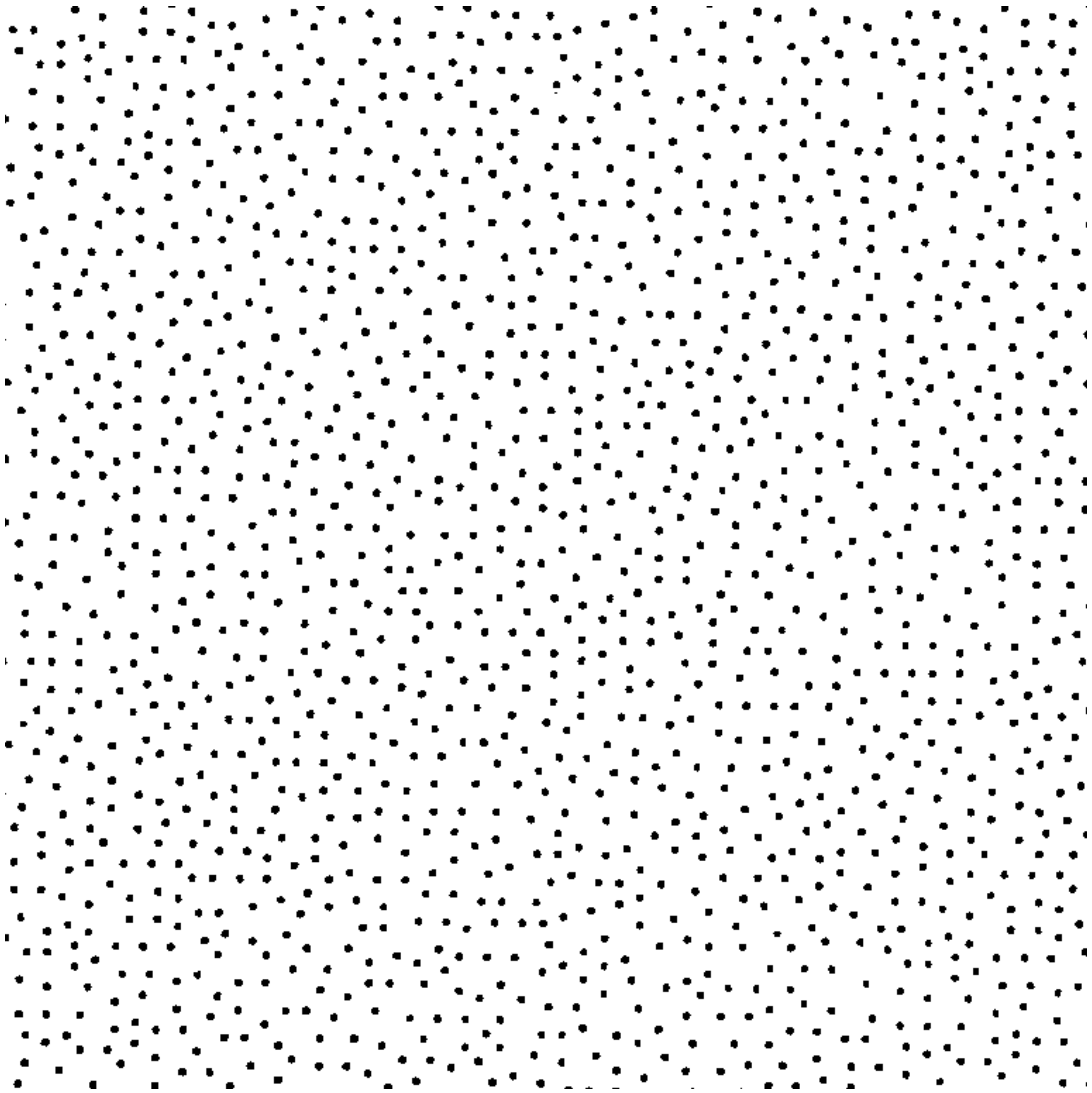
Jittered Sampling

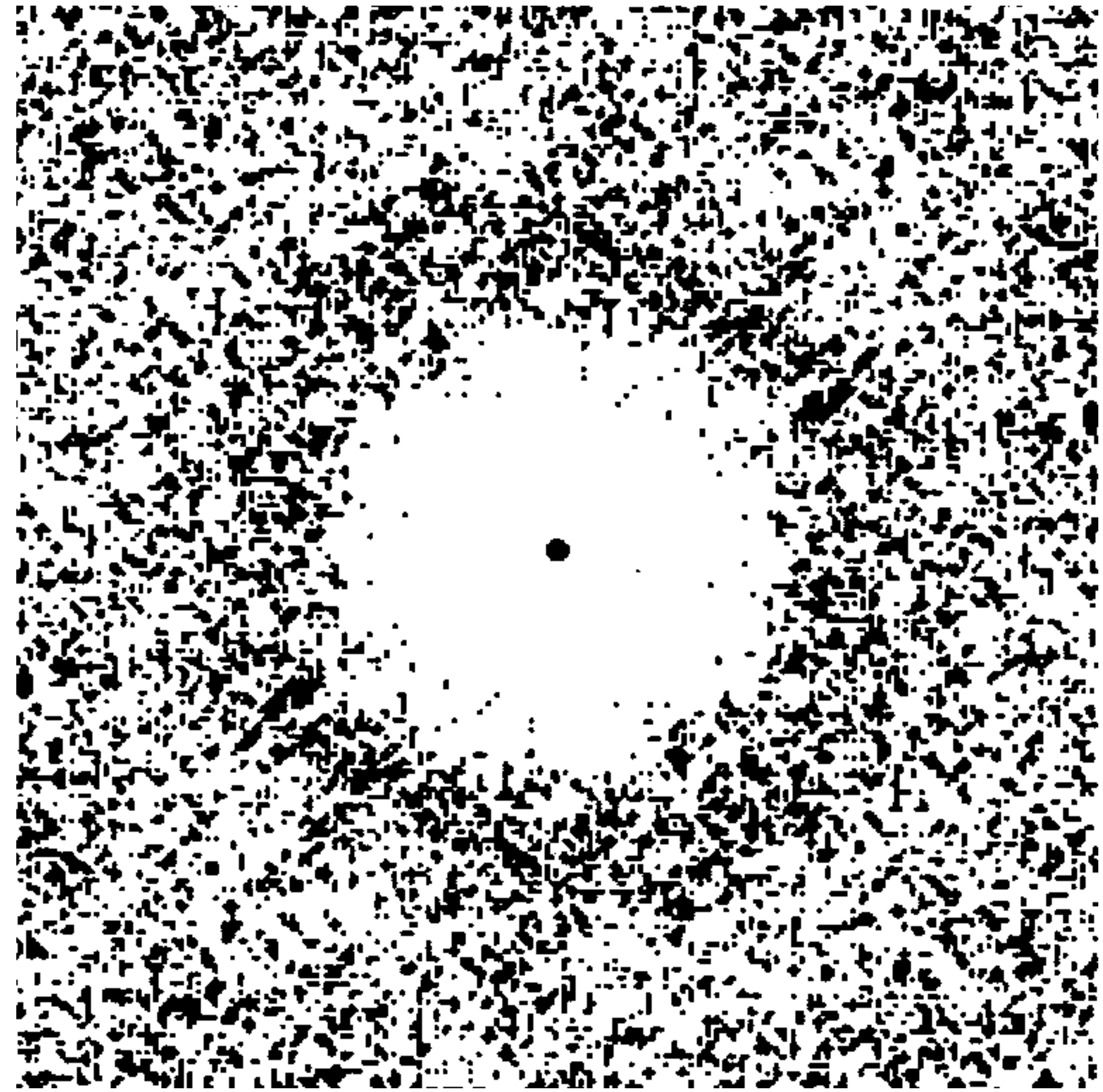


From [Mitchell 1987]

Don P. Mitchell. 1987. Generating antialiased images at low sampling densities. SIGGRAPH Comput. Graph. 21, 4 (July 1987), 65–72. DOI:https://doi.org/10.1145/37402.37410

Poisson Disk Sampling



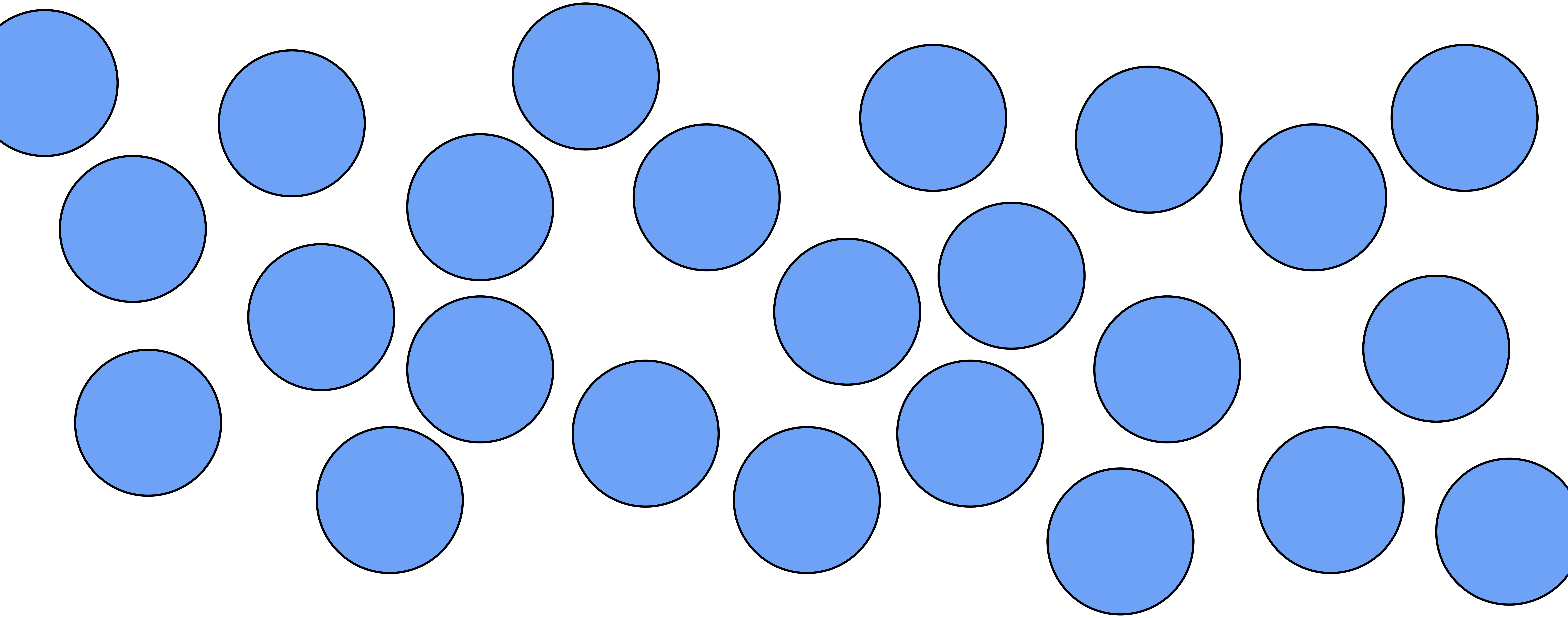


From [Mitchell 1987]

Don P. Mitchell. 1987. Generating antialiased images at low sampling densities. SIGGRAPH Comput. Graph. 21, 4 (July 1987), 65–72. DOI:https://doi.org/10.1145/37402.37410

Poisson Disk Sampling

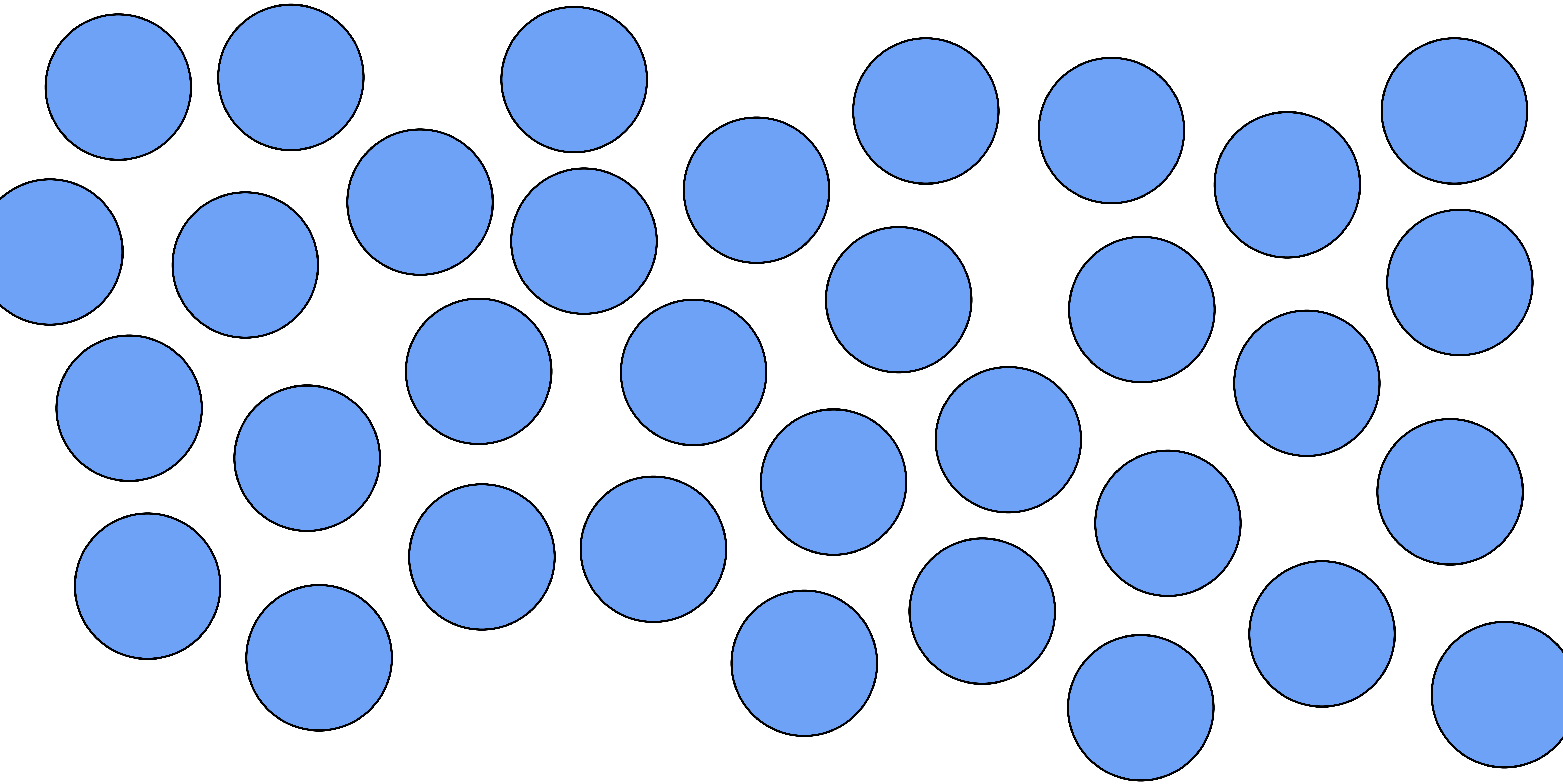
- How do you generate a "nice" sample?
- One of the earliest algorithms: *Poisson disk sampling*
- Iteratively add random non-overlapping disks (until no space left)



Decent spectral quality, but we can do better.

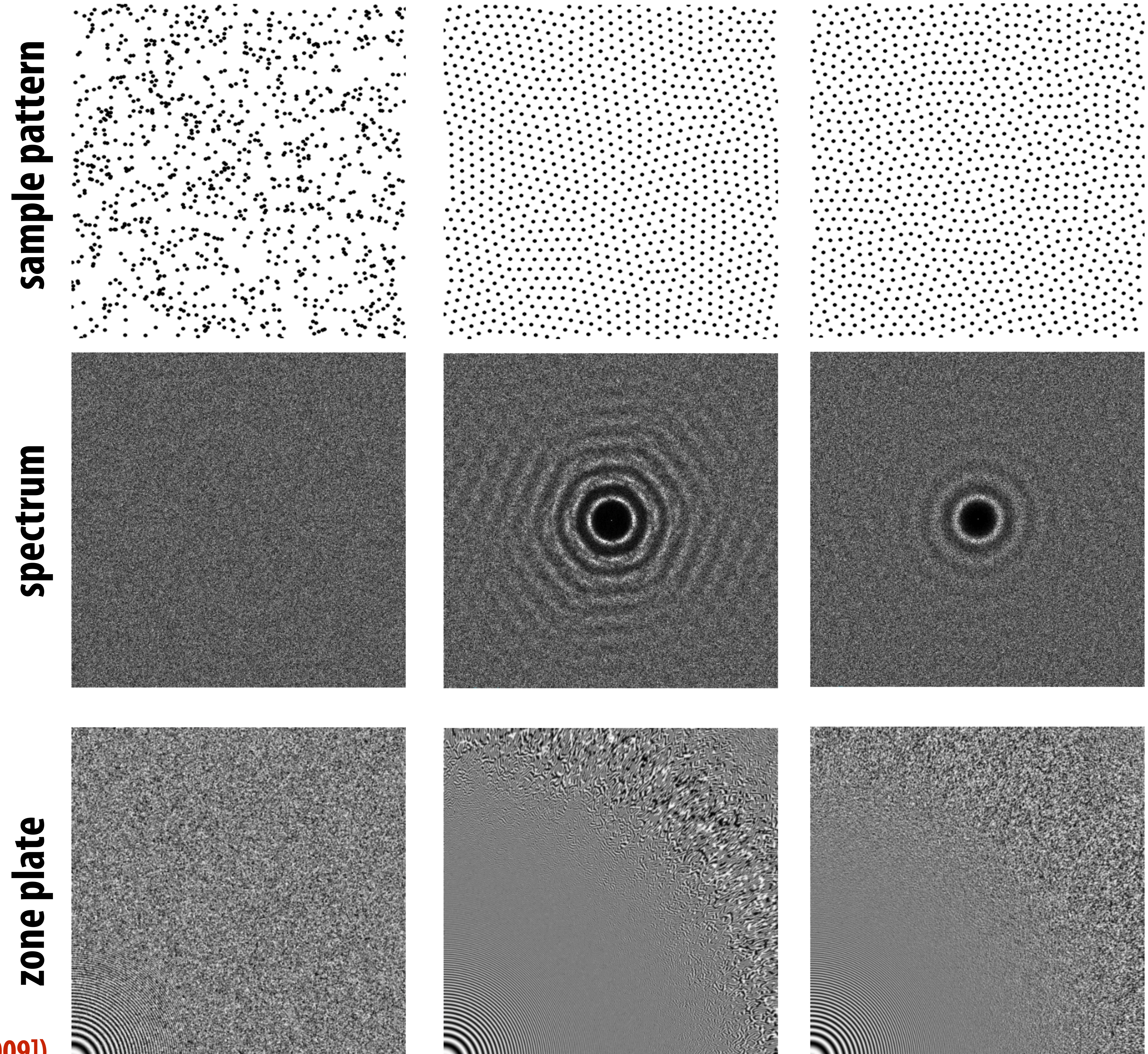
Lloyd Relaxation

- Iteratively move each disk to the center of its neighbors



Better spectral quality, slow to converge. Can do better yet...

Spectrum affects reconstruction quality



(from [Balzer et al. 2009])

Integration Error when Sampling

Integral

$$I(f) = \int f(x)\,dx = F(0)$$

Sample Pattern

$$s(x) = \frac{1}{N}\sum \delta(x - x_i)$$

Sampled integral

$$I_s(f) = \int f(x)s(x)\,dx = \frac{1}{N}\sum f(x_i)$$

$$f(x)s(x) \Longleftrightarrow F(\omega) \otimes S(\omega)$$

$$I_s(f) = \int f(x)s(x)\,dx = F(\omega) \otimes S(\omega)|_{\omega=0}$$

Error

$$\Delta = F(0) - F(\omega) \otimes S(\omega)|_{\omega=0}$$

Non-uniform Sampling

UNIFORM SAMPLING

- The spectrum of uniformly spaced samples is also a set of uniformly spaced spikes
- Multiplying the signal by the sampling pattern corresponds to placing a copy of the spectrum at each spike (in freq. space)
- Aliases are coherent, and very noticeable

NON-UNIFORM SAMPLING

- Samples at non-uniform locations have a different spectrum; a single spike plus (blue) noise
- Sampling a signal in this way converts aliases into higher-frequency noise
- Noise is incoherent, and much less objectionable visually
- May cause error in the integral

Further Reading

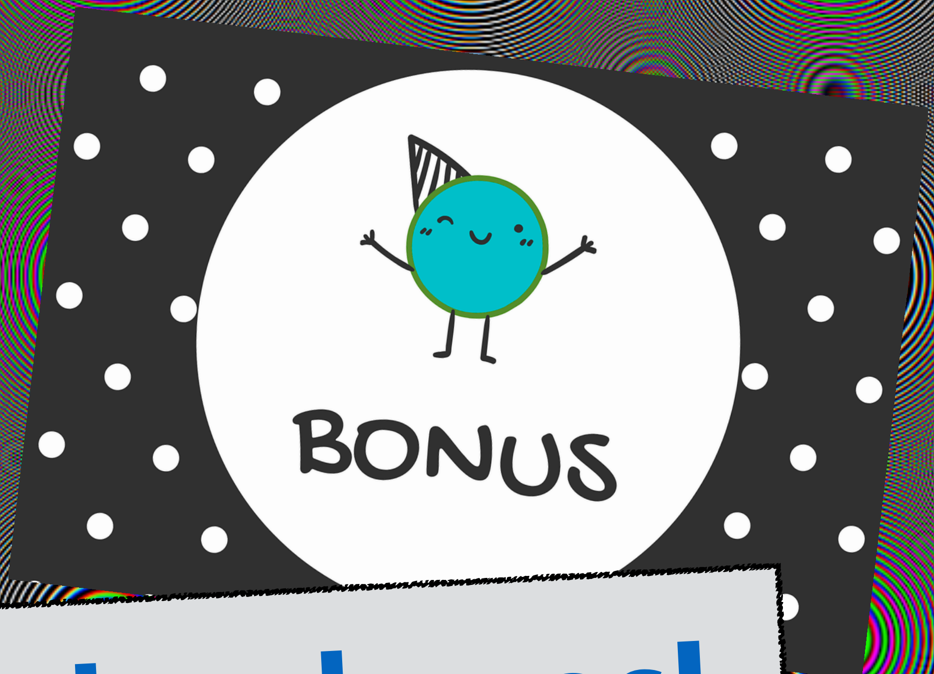
Chapter 7

SAMPLING AND RECONSTRUCTION

- **Mostly section 7.1 for now**

Signal Processing and Sampling

CS 348B, Lecture 7



Now with color demos!

"the aliasing side" by FabriceNeyret2
https://www.shadertoy.com/view/4slXR4