A gentle introduction to Fourier analysis

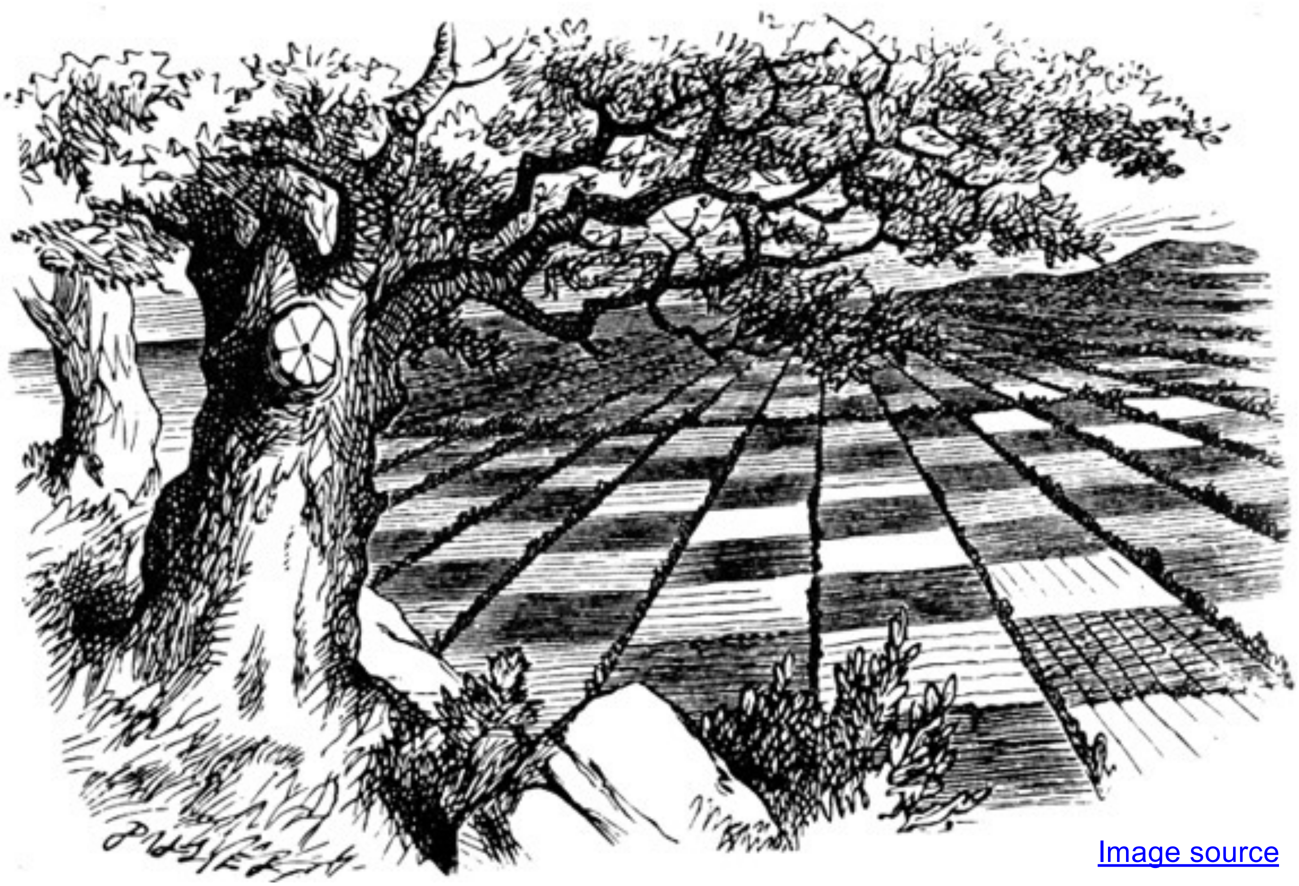


Image source

Many slides borrowed from S. Seitz, A. Efros, D. Hoiem, B. Freeman, A. Zisserman

Formal models of sampling

Passing from a continuous function—like the irradiance at the back of a camera system—to a collection of values on a discrete grid —like the pixel values reported by a camera—is referred to as *sampling*. For sampling in one dimension, the most important case involves sampling on a uniform discrete grid. Assume that the samples are defined at integer points, yielding a process that takes some function and returns a vector of values:

$$\texttt{sample}_{1D}(f(x)) = \mathbf{f}.$$

Here the ith component of $\mathbf{f}$ is $f(x_i)$, and $\mathbf{f}$ is an infinite vector to avoid having to write indices, etc. (Figure 43.2).

Sampling in 2D is very like sampling in 1D. Although sampling can occur on nonregular grids (the best example being the human retina), the most important case has samples on a uniform grid of integer coordinates. This gives

$$\texttt{sample}_{2D}(F(x, y)) = \mathcal{F},$$

where the i, jth element of the array $\mathcal{F}$ is $F(x_i, y_j) = F(i, j)$. The grid is infinite in each dimension to avoid having to write ranges, etc. (Figure 43.4). Notice that

Problem: no FT

For these sampled functions, any integral will be zero

   This isn't good (eg no Fourier Transform)

Q: sensible model of a sampled function that has an FT?

A model of Sampling

We want to model sampling in a way that allows us to take Fourier Transforms.

Challenges:

Should be able to compute a meaningful integral of the sampled data

In particular, we would like

$$\int_W \texttt{sample}_{1D}(f(t))g(t)dt$$

(where W is some interval) to be as similar as possible to

$$\int_W f(t)g(t)dt.$$

A trick – the delta function

Define the delta function in 1D by

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ \text{uncomfortable} & x = 0 \end{cases}$$

$$\int f(x)\delta(x)dx = f(0)$$

This isn't a function in any familiar sense, but it is useful and crops up in all sorts of places

Desirable property

Understanding aliasing will require a continuous model of a sampled signal. Write $C(\mathcal{I})$ for the operation that maps a sampled image $\mathcal{I}$ to this continuous model. This model should respect convolution and sampling in a sensible way. Choose some continuous convolution kernel $g(x, y)$ A desirable property of this model is that if you convolve $C(\mathcal{I})$ with $g(x, y)$, then sample the result, you get what you would have gotten if you convolve $\mathcal{I}$ with $\texttt{sample}_{2D}(g)$. To write this out, it is helpful to distinguish discrete convolution (I will write $*_d$) and continuous convolution (I will write $*_c$). The property is:

$$\texttt{sample}_{2D}(C(\mathcal{I}) *_c g) = \mathcal{I} *_d \texttt{sample}_{2D}(g).$$

Bed of nails functions

Now $C(\mathcal{I}$ cannot just be a function that takes the value of the signal at integer points and is zero everywhere else, because this model has a zero integral so the left hand side will be zero. Instead, use

$$C(\mathcal{I})(x,y) = \sum_{i,j} \mathcal{I}_{ij}\delta(x-i, y-j)$$

and find

$$C(\mathcal{I}) *_c g = \sum_{i,j} \mathcal{I}_{ij} g(x - x_i, y - y_j)$$

so that the u, v'th component of

$$\texttt{sample}_{2D}(C(\mathcal{I}) *_c g) \text{ is } \sum_{i,j} \mathcal{I}_{ij} g(x_u - x_i, y_v - y_j)$$

and the property holds.

Sampling in 2D

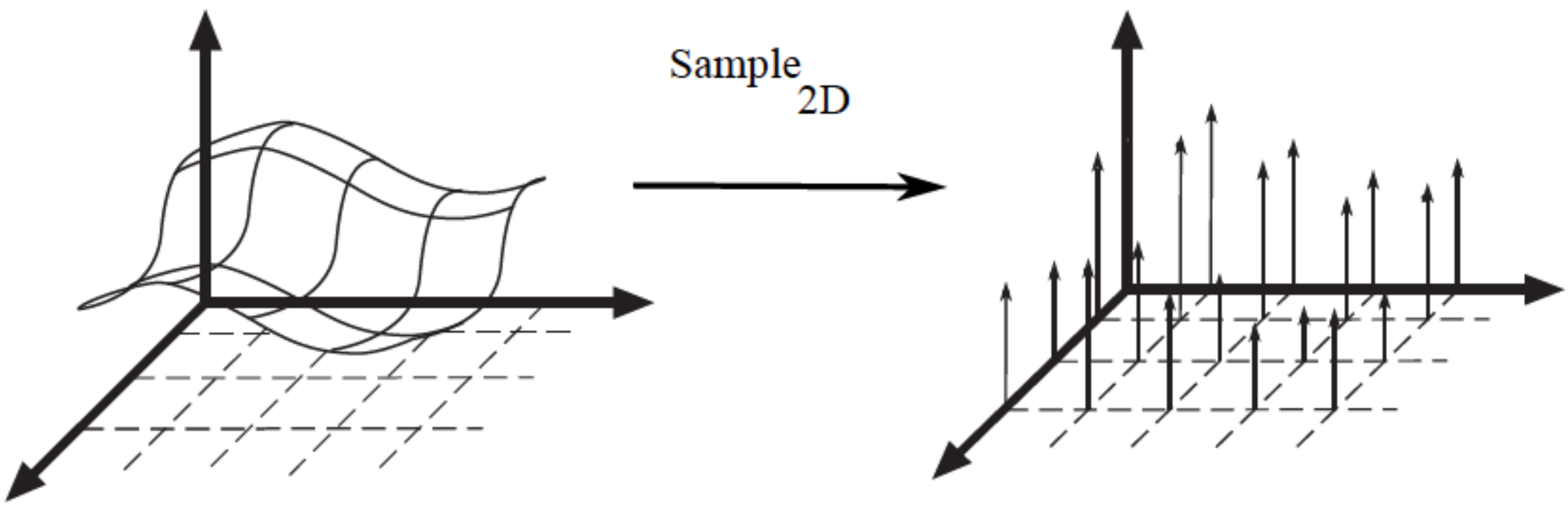


FIGURE 7.4: *Sampling in 2D takes a function and returns an array; again, we allow the array to be infinite dimensional and to have negative as well as positive indices.*

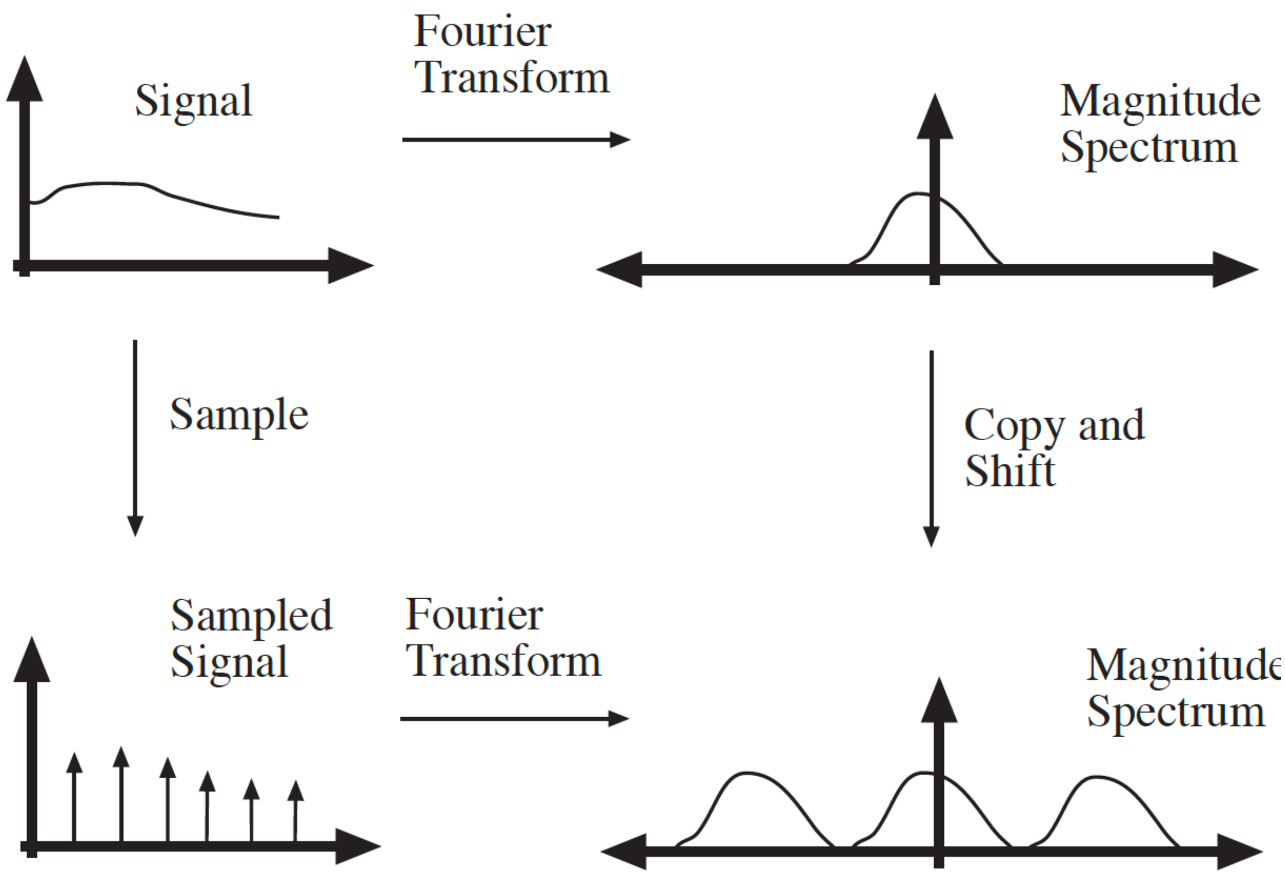
The FT of a sampled signal

Now convolving a function with a shifted δ-function merely shifts the function (see exercises). This means that the Fourier transform of the sampled signal is the sum of a collection of shifted versions of the Fourier transforms of the signal, that is,

$$\begin{aligned}\mathcal{F}(\mathtt{sample}_{2D}(f(x,y))) &= \mathcal{F}\left(f(x,y)\left\{\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right\}\right)\\ &= \mathcal{F}(f(x,y)) **\mathcal{F}\left(\left\{\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right\}\right)\\ &= \sum_{i=-\infty}^{\infty} F(u-i,v-j),\end{aligned}$$

where we have written the Fourier transform of $f(x,y)$ as $F(u,v)$.

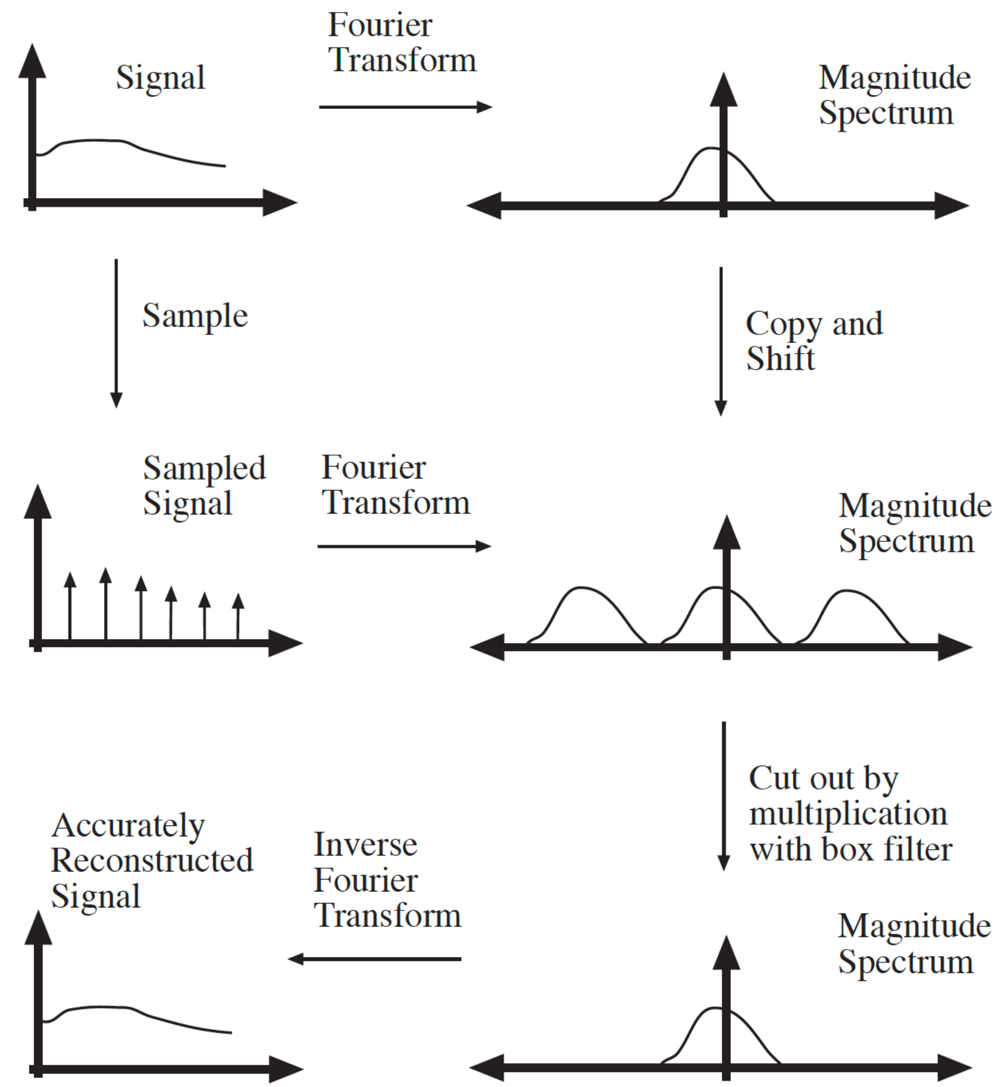
The FT of a sampled signal



Sampling

If the magnitude blobs don't overlap, you can reconstruct from samples

Just cut out the blob in FT space with a box filter, inverse FT

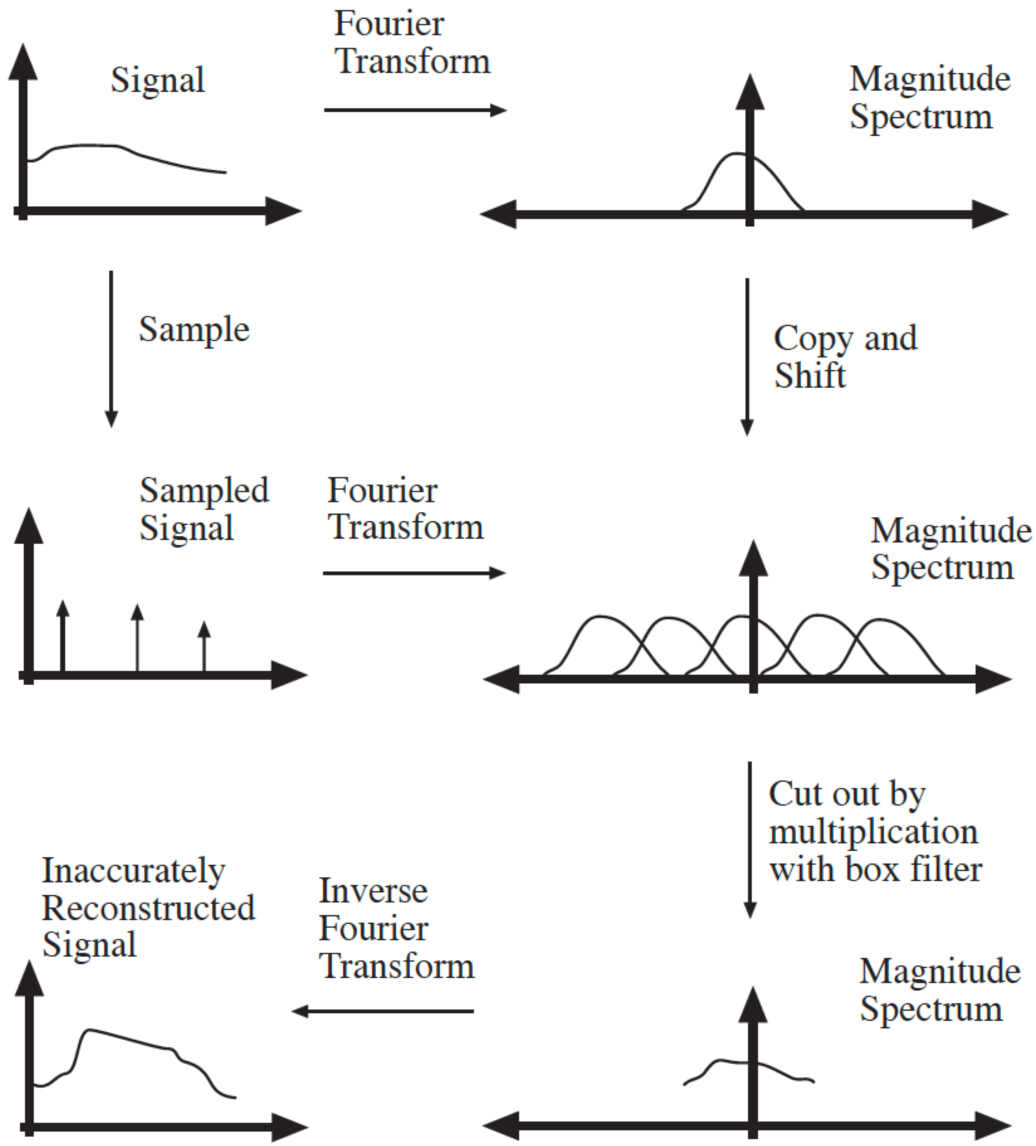


Sampling

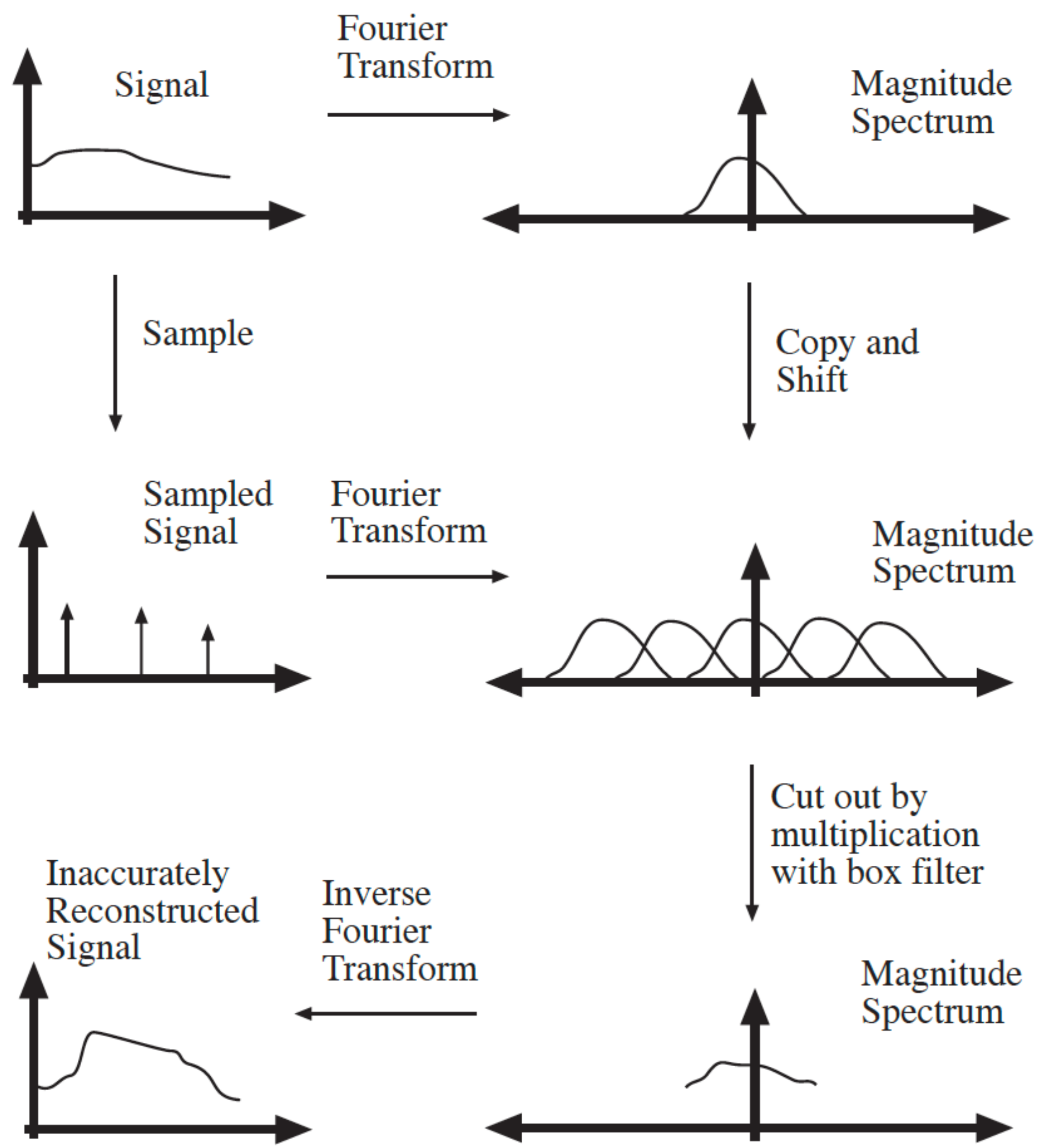
If the magnitude blobs overlap, you can't reconstruct from samples

When you cut the blob out in FT space with a box filter, you'll get it wrong

Nyquist's theorem

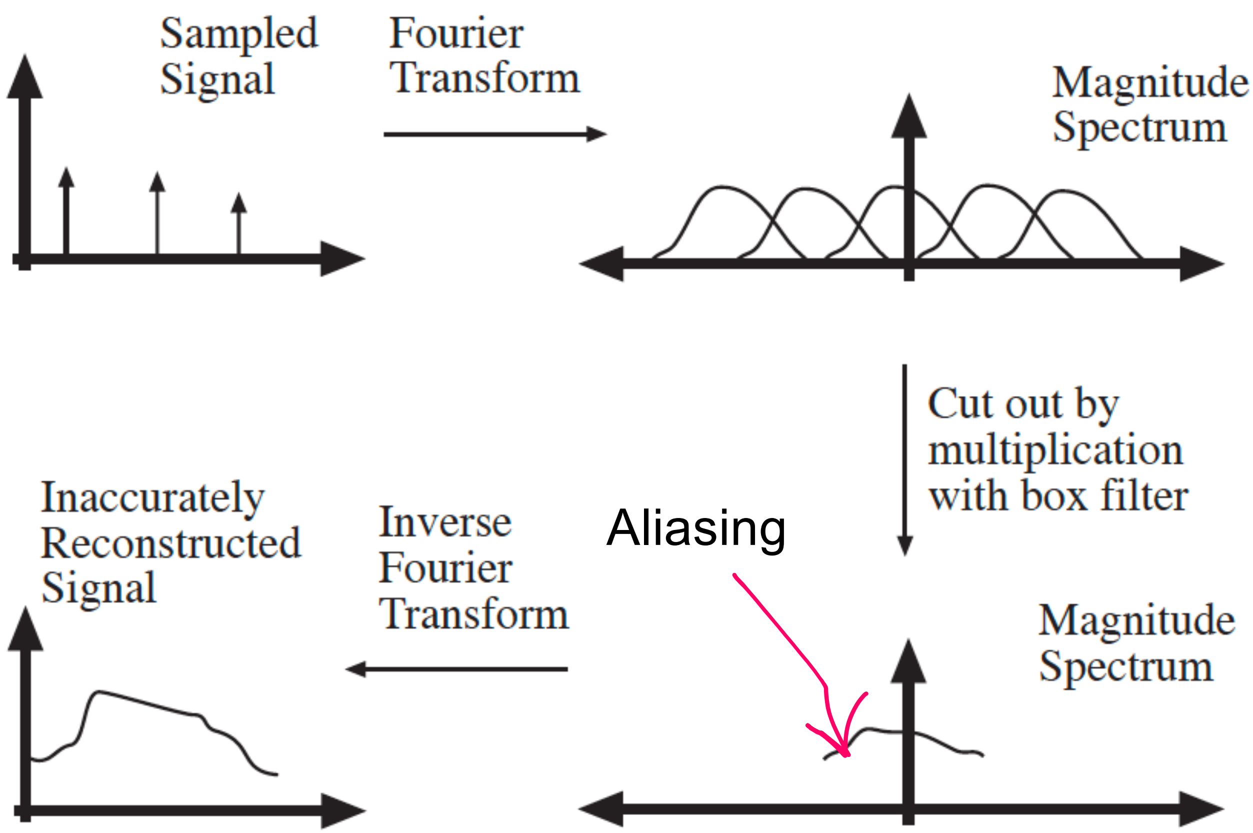


Sampling



Signal
Fourier Transform
Magnitude Spectrum
Sample
Copy and Shift
Sampled Signal
Fourier Transform
Magnitude Spectrum
Cut out by multiplication with box filter
Inaccurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum

Sampling



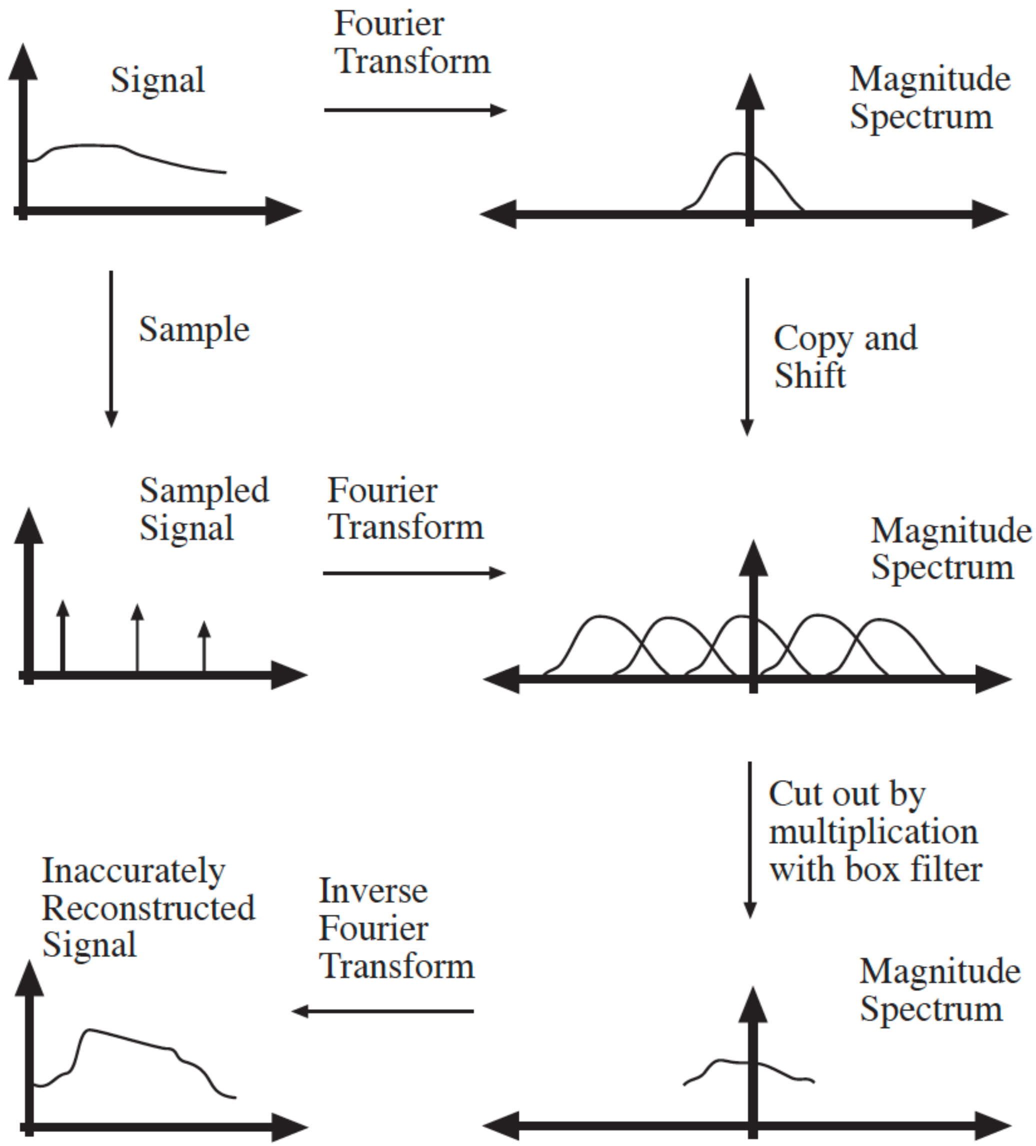
Sampled
Signal
Fourier
Transform
Magnitude
Spectrum
Cut out by
multiplication
with box filter
Inaccurately
Reconstructed
Signal
Inverse
Fourier
Transform
Aliasing
Magnitude
Spectrum

Sampling

If the magnitude blobs overlap, you can't reconstruct from samples

When you cut the blob out in FT space with a box filter, you'll get it wrong

Nyquist's theorem



Consequences

Nyquist limits aren't really viable

Apply the convolution theorem

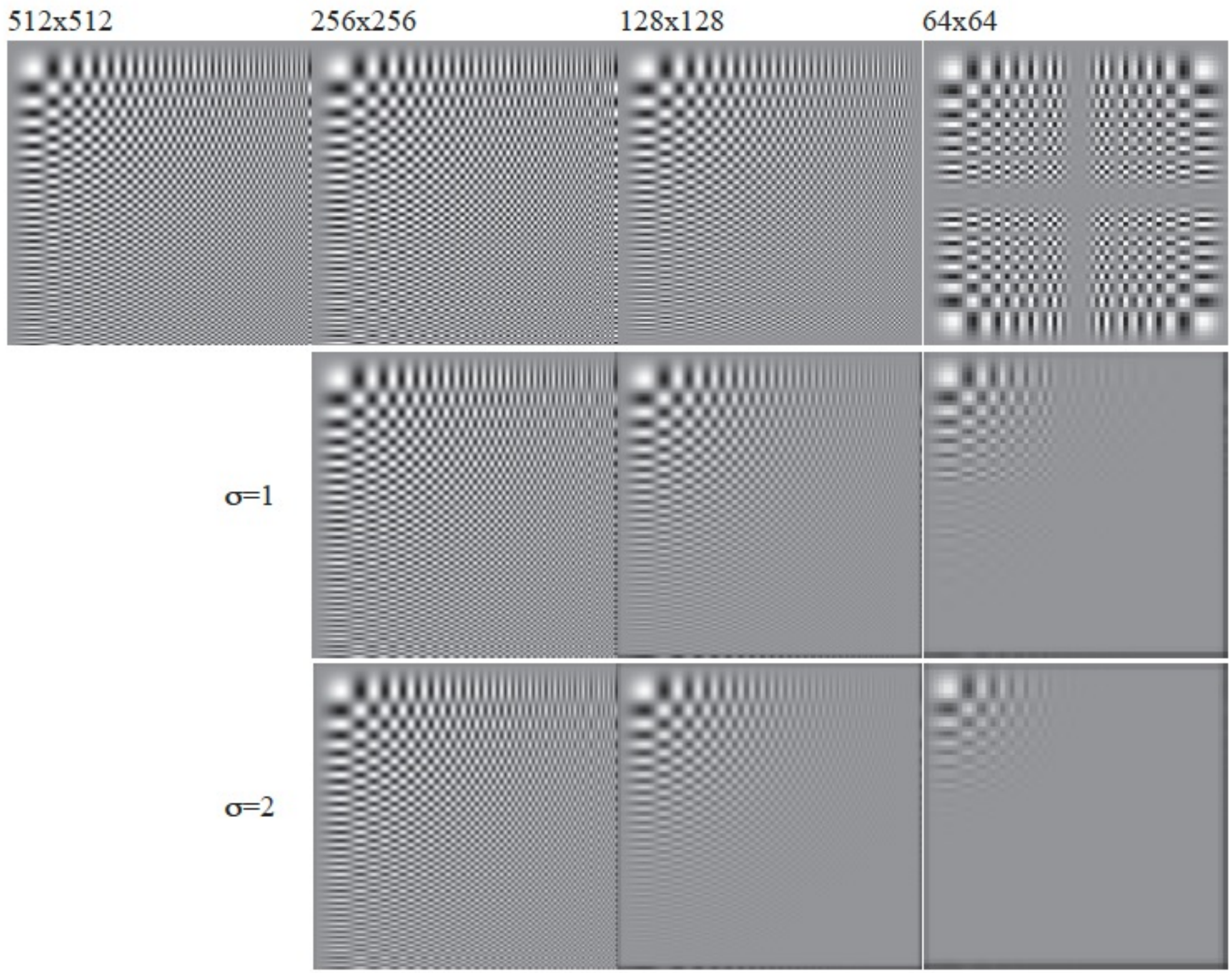
A box in FT magnitude space is a filter with infinite support

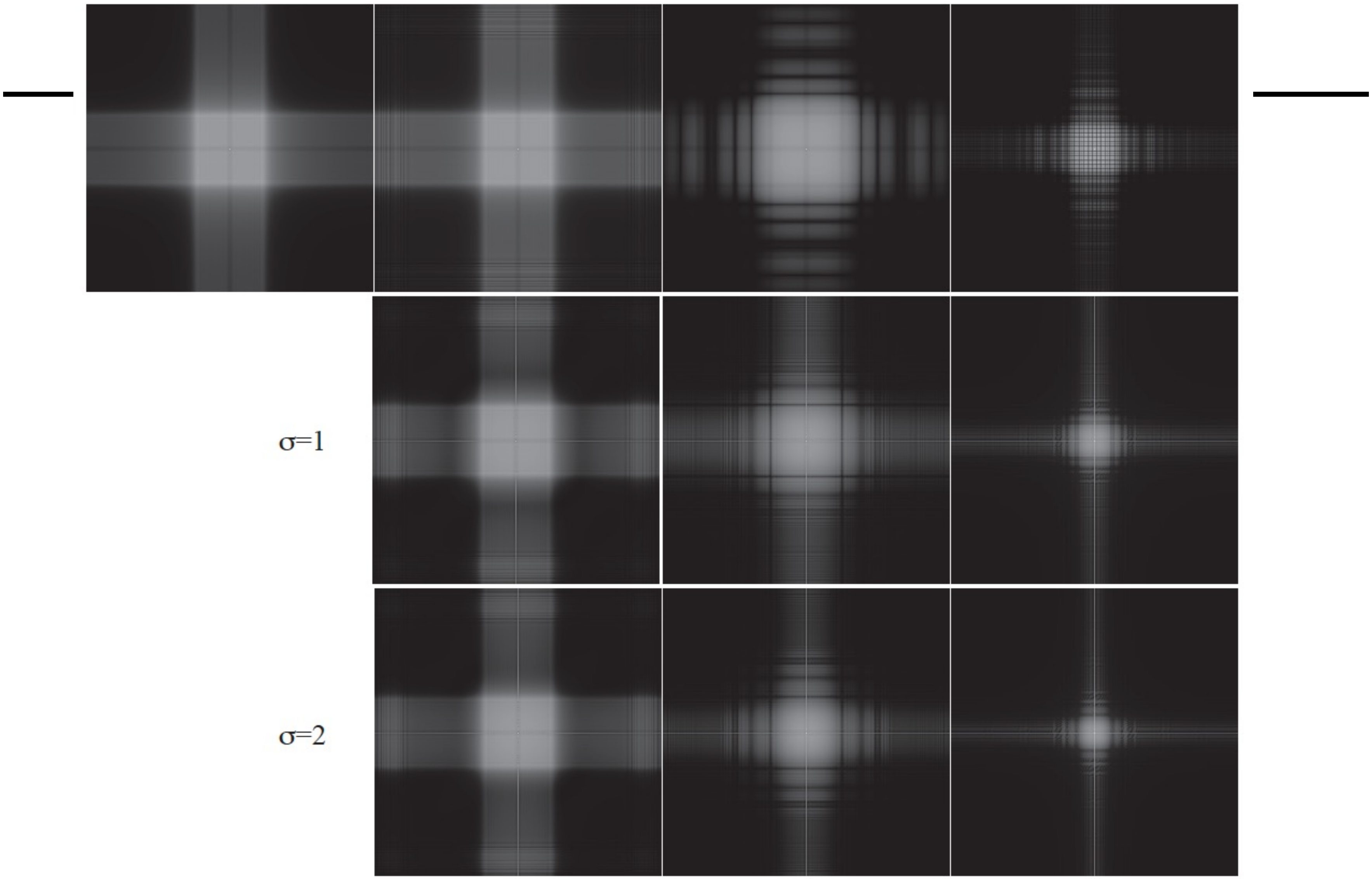
(and you can't make one of those)

You're forced to choose a filter that is low pass, but isn't perfect

the choice has consequences

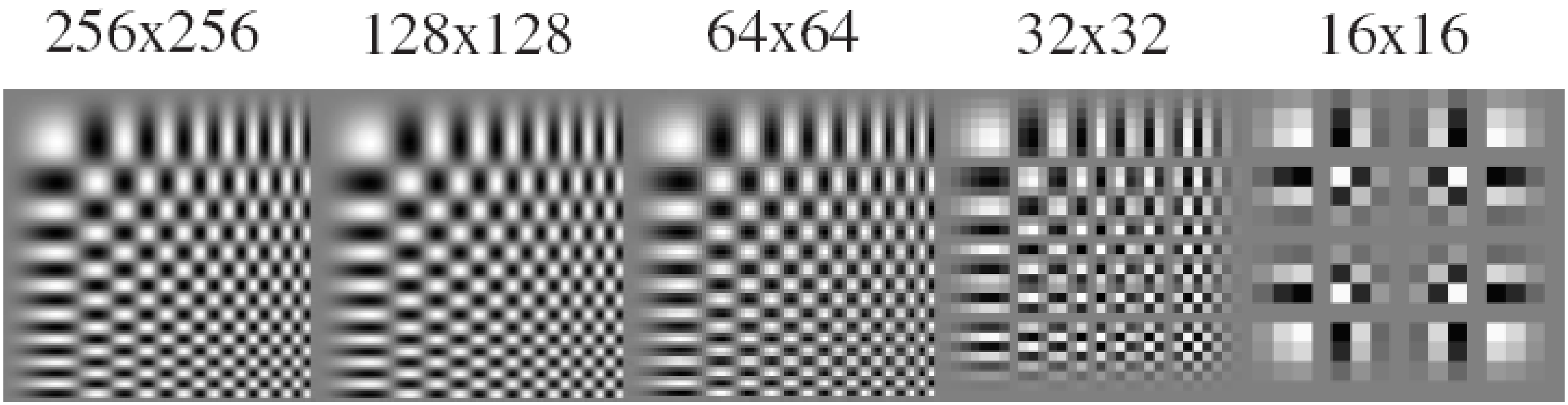
Gaussian is such a filter


512x512
256x256
128x128
64x64
σ=1
σ=2


σ=1
σ=2

Mystery 2 SOLVED

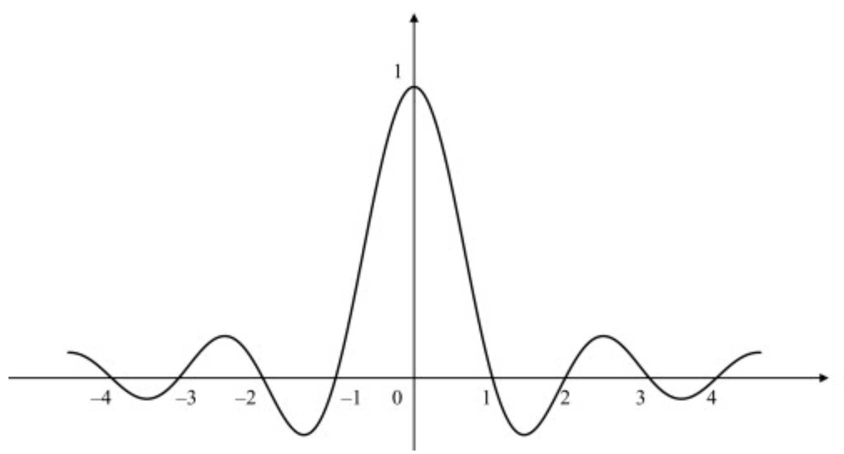
- Why can downsampling sometimes lead to aliasing?



The downsampling mangles the Fourier Transform magnitude spectrum UNLESS

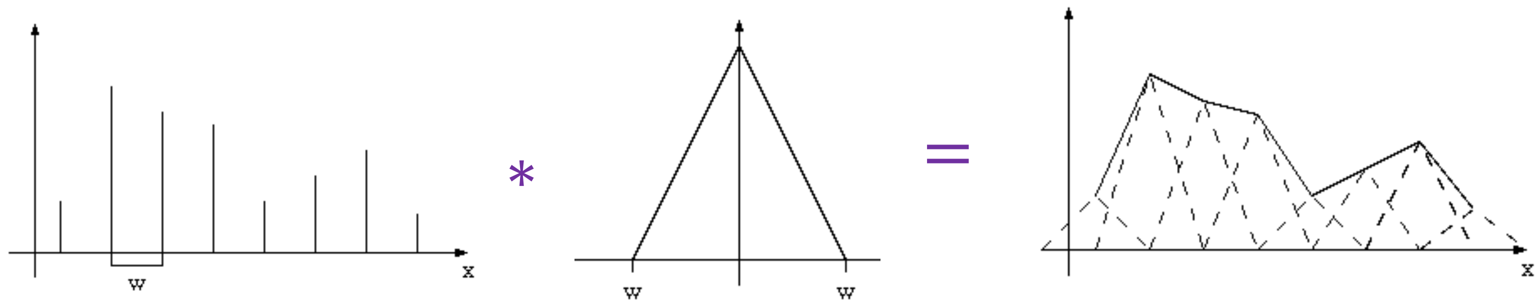
Aside: Analyzing interpolation methods

- Perfect reconstruction of the subsampled signal requires convolution with a sinc filter in the spatial domain, which is bad because sinc has infinite support
- Instead, simpler reconstruction (interpolation) methods are typically used



Aside: Analyzing different interpolation methods

- Linear reconstruction can be done by convolving the sampled signal with a *triangle filter*:



- However, the Fourier transform of the triangle filter is the sinc^2 function, so multiplying the signal's spectrum by it introduces high-frequency artifacts

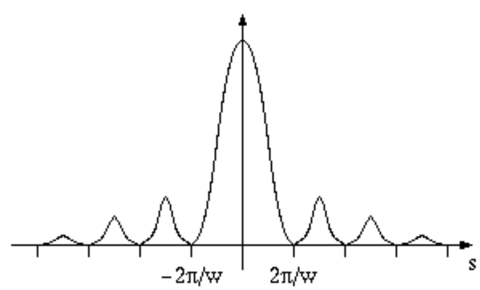


Image source

Bilinear interpolation closeup

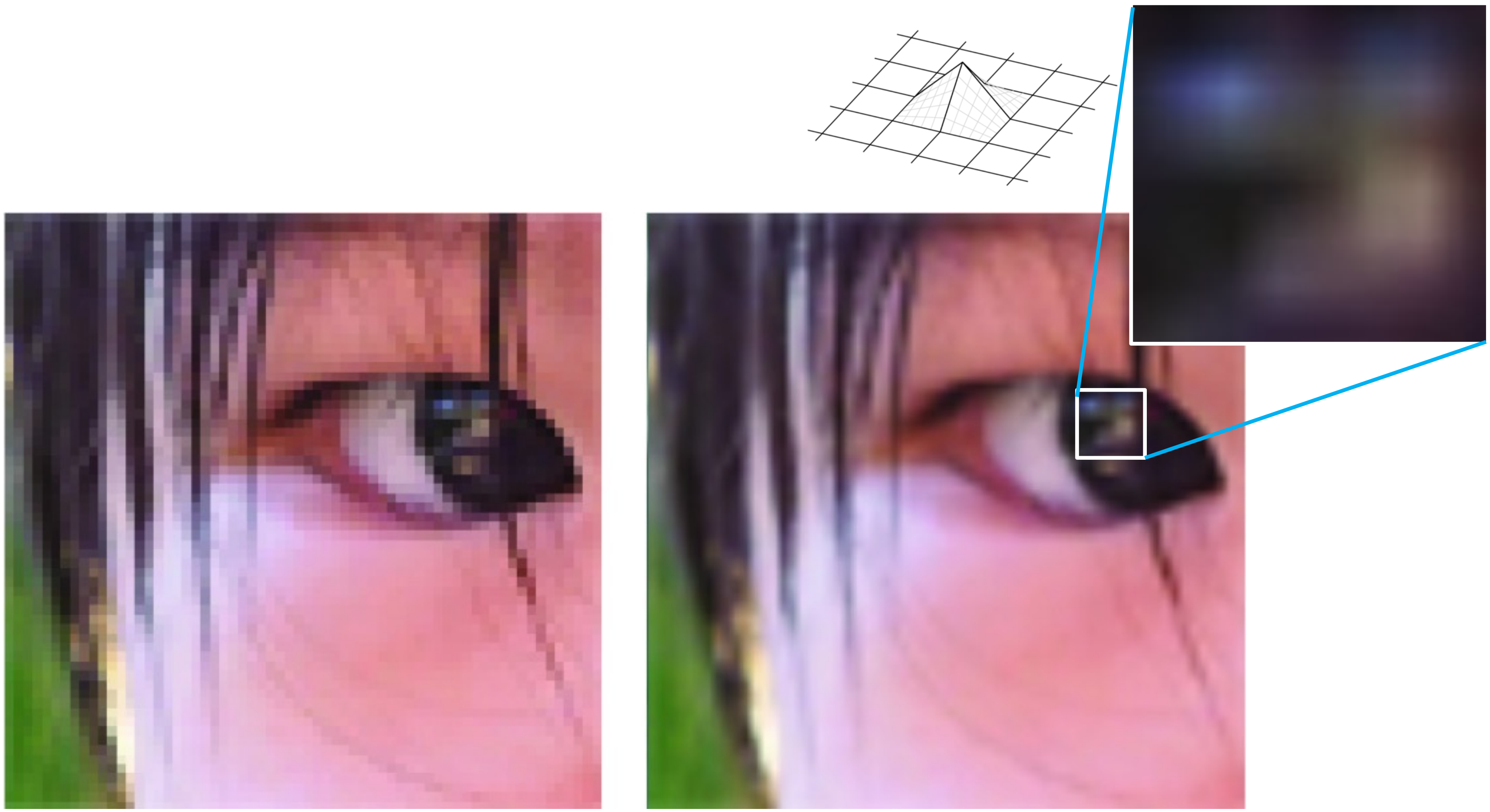


Image source

Why else should you care about Fourier analysis?

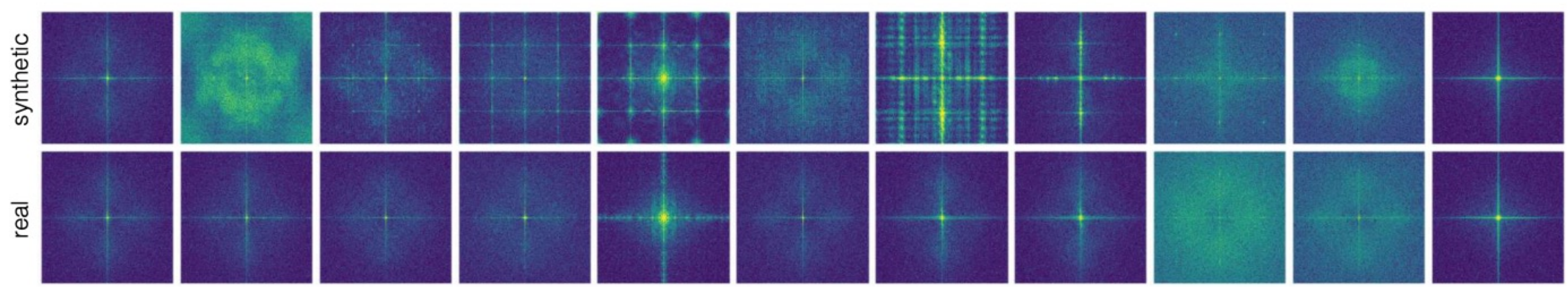
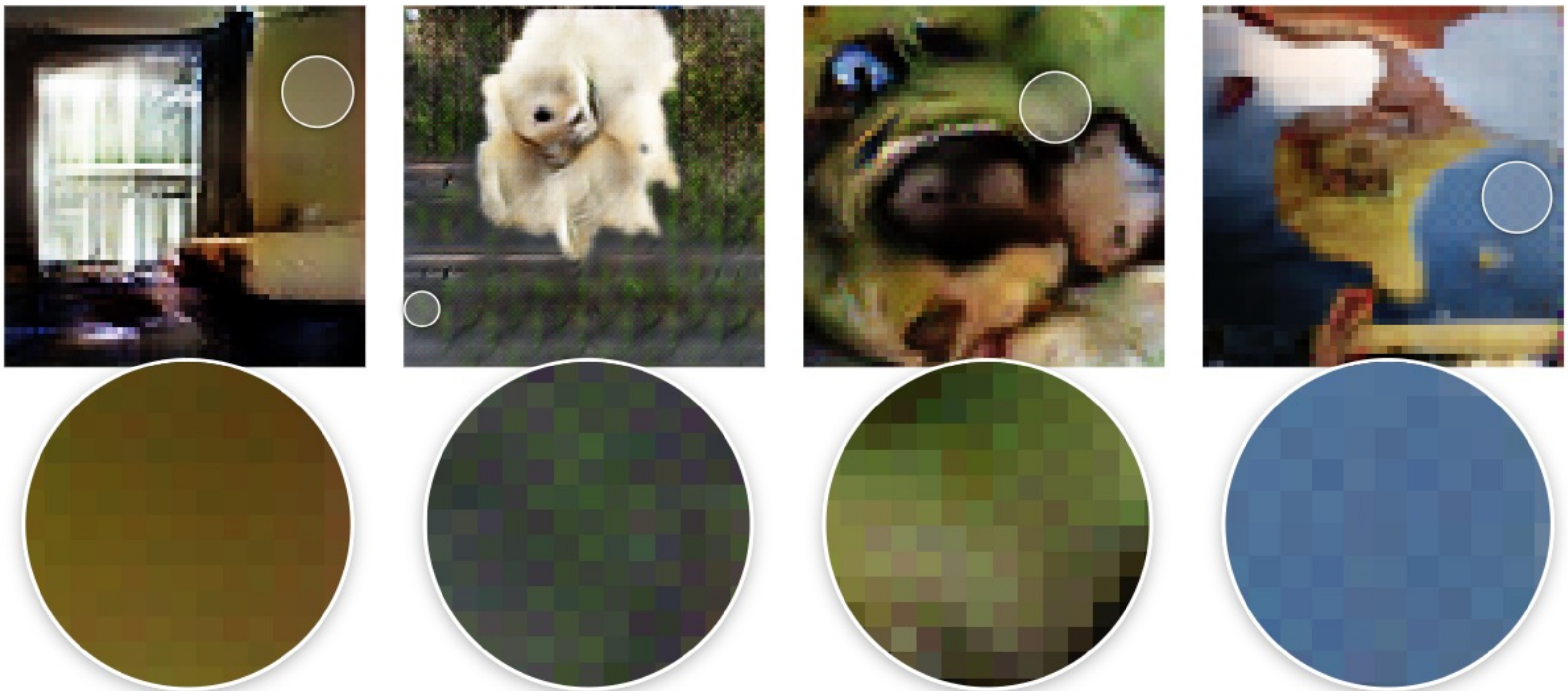


Figure 7: **Frequency analysis on each dataset.** We show the average spectra of each high-pass filtered image, for both the real and fake images, similar to Zhang *et al*. [50]. We observe periodic patterns (dots or lines) in most of the synthetic images, while BigGAN and ProGAN contains relatively few such artifacts.

S.-Y. Wang et al. CNN-generated images are surprisingly easy to spot... for now. CVPR 2020

Why else should you care about Fourier analysis?

Checkerboard and repetition artifacts in GAN-generated images



Radford, et al., 2015 [1] Salimans et al., 2016 [2] Donahue, et al., 2016 [3] Dumoulin, et al., 2016 [4]

https://distill.pub/2016/deconv-checkerboard/