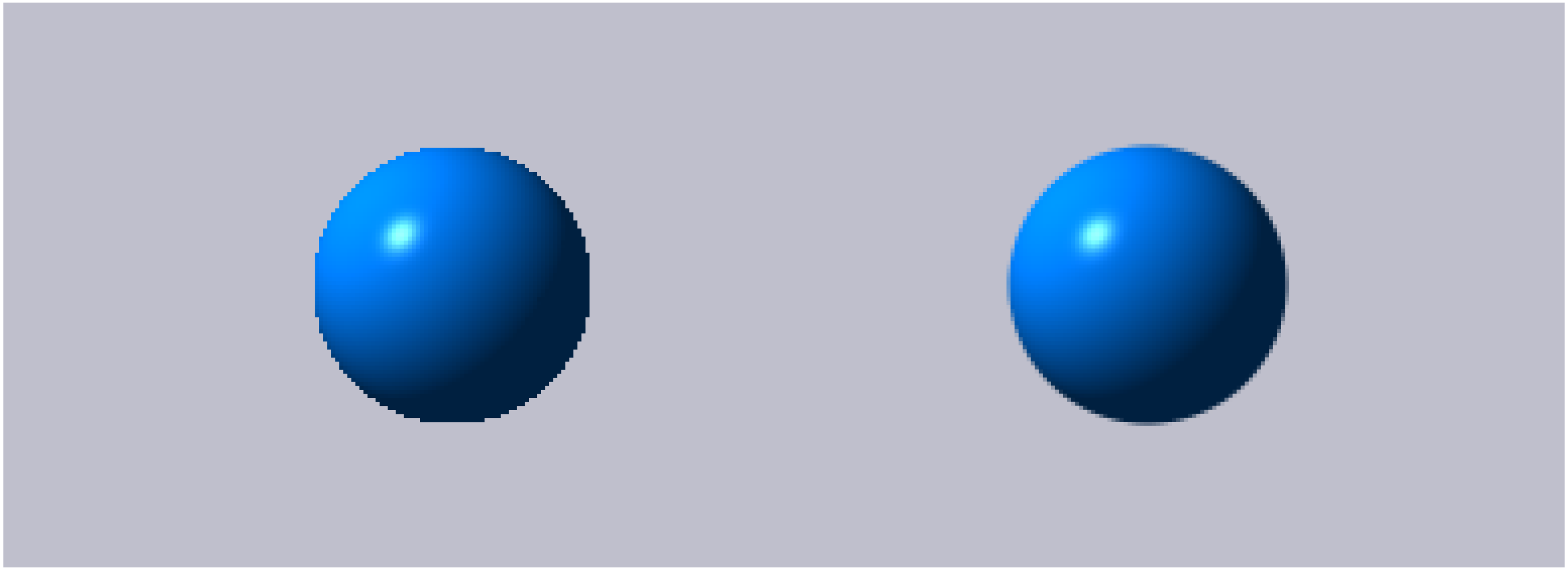
# Lecture 8:

# Sampling & Reconstruction

## Contents

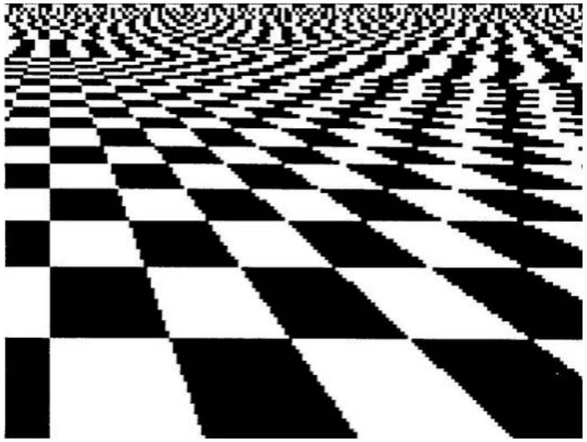
1. Motivation
2. Sampling Theory
3. Fourier Transformation
4. Aliasing
5. Stratified Sampling
6. Image Reconstruction

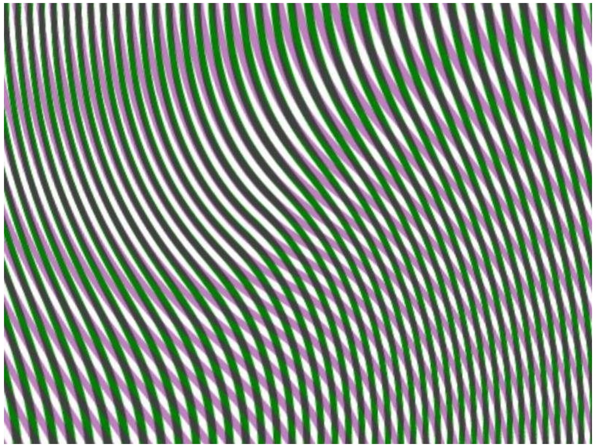


Motivation

# Aliasing

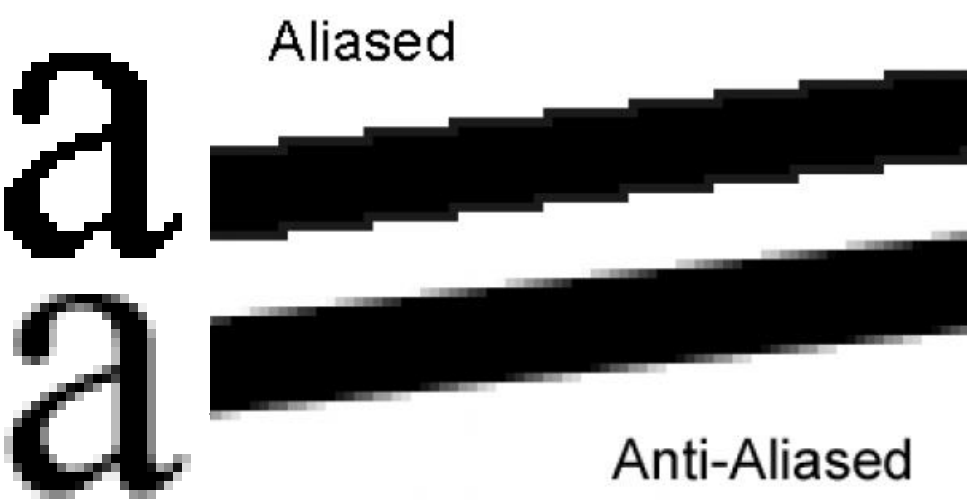
- **Aliasing** is an error arising because the ***sampling*** and ***reconstruction*** process involves *approximation*
- These errors occur because the sampling process is not able to capture all of the information from the continuously defined image function
- Aliasing can manifest itself in many ways, including jagged edges, flickering in animations, moiré patterns, *etc.*

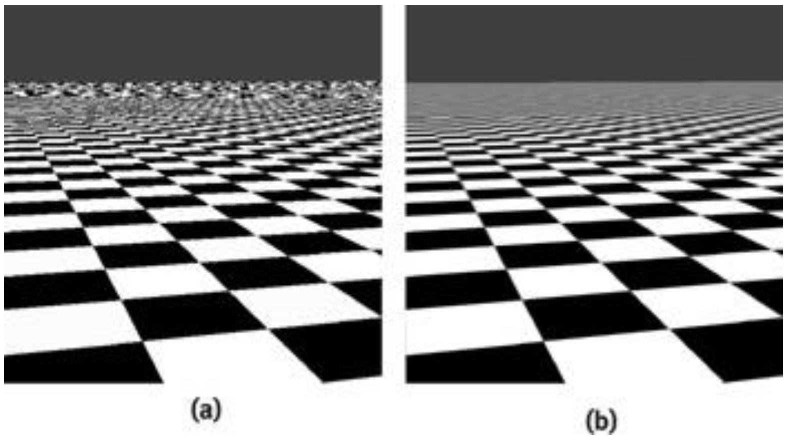




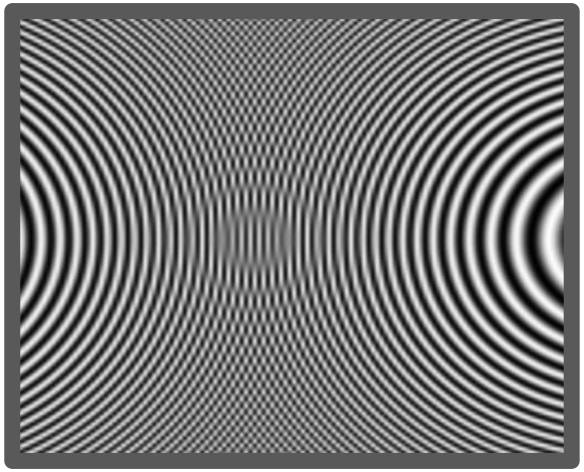
## Spatial Aliasing

- Stair cases, Moiré patterns, *etc*.



## Moiré pattern



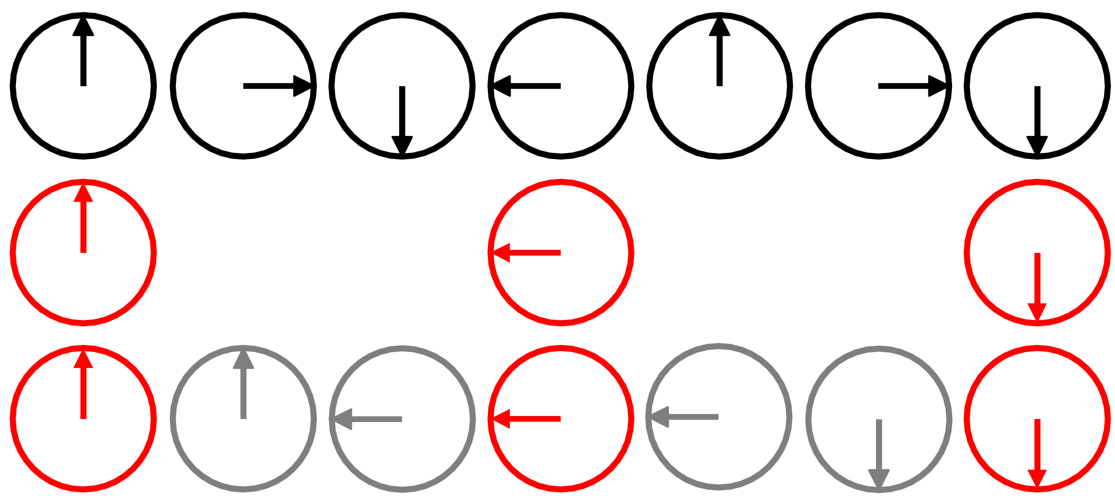
# Temporal Aliasing

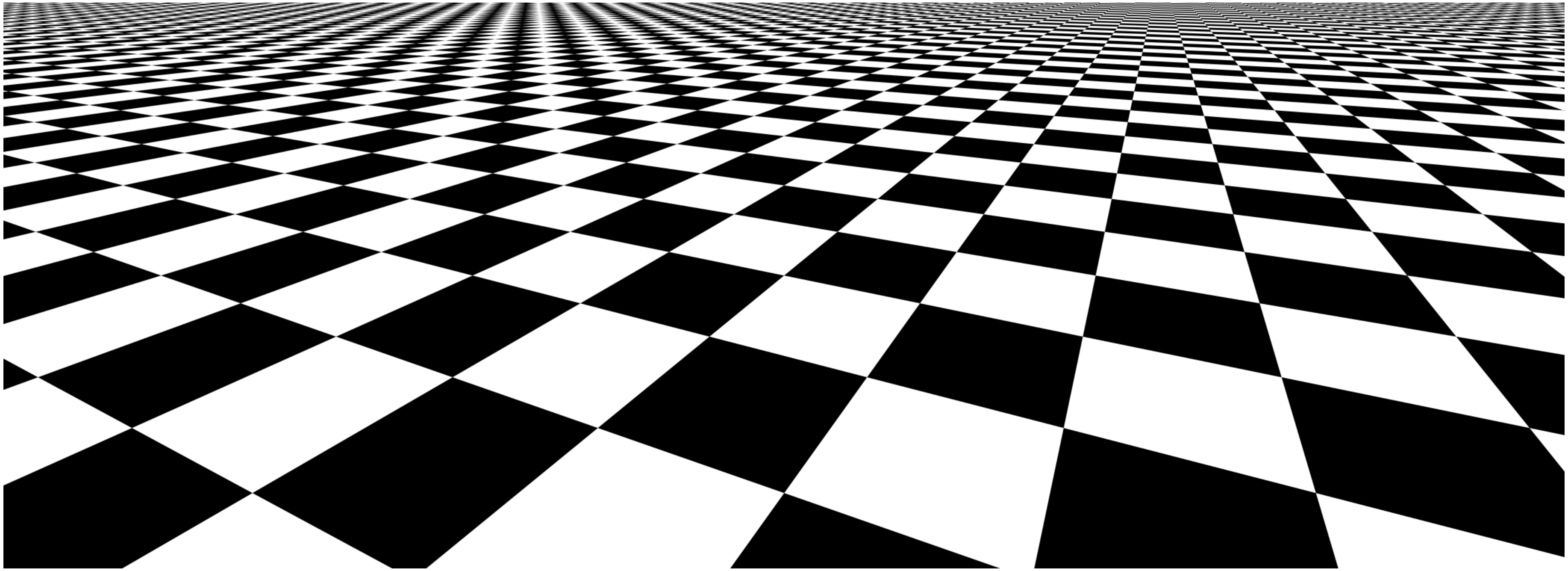

Optical Illusion: The wagon wheel effect (aliasing)

**Real World**: clockwise rotation

**Sampling**: captured data

**Reconstruction**: counter-clockwise rotation

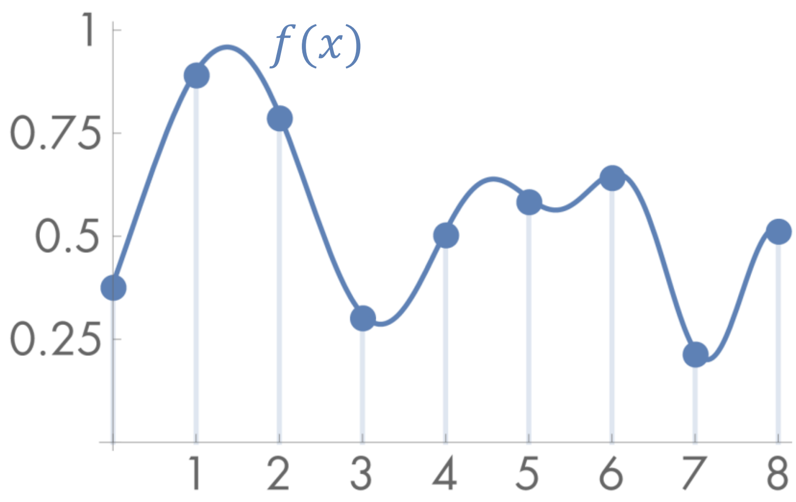




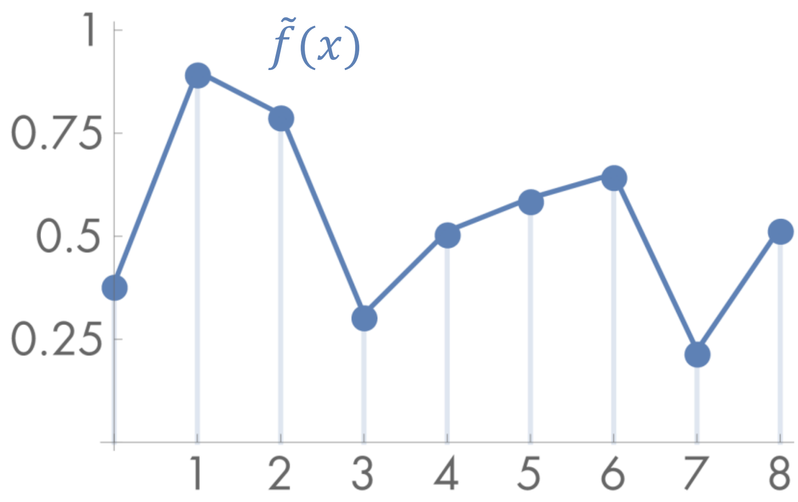
# Sampling Theory

## Sampling and Reconstruction

- Taking a set of point samples of $f(x)$ (indicated by dots), we determine the value of the function at those positions
- These sample values can be used to reconstruct a function $\tilde{f}(x)$ that is an approximation to $f(x)$
- The sampling theorem, makes a precise statement about the conditions on $f(x)$, the number of samples taken, and the reconstruction technique used under which $\tilde{f}(x)$ is exactly the same as $f(x)$.
  - The fact that the original function can sometimes be reconstructed exactly from point samples alone is remarkable.




Sampling




Reconstruction

# Sampling Theory

**Nature**

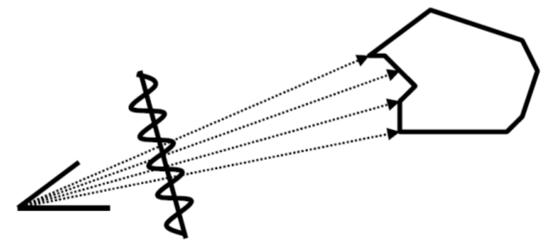
## Continuous signal (2D / 3D / 4D with time)

- Defined at all points

**Acquisition**

## Sampling

- Rays, pixel / texel, spectral values, frames, *etc*.



**Representation**

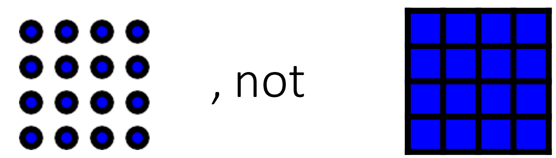
## Discrete data

- Discrete points, discretized values

**Reconstruction**

## Interpolation

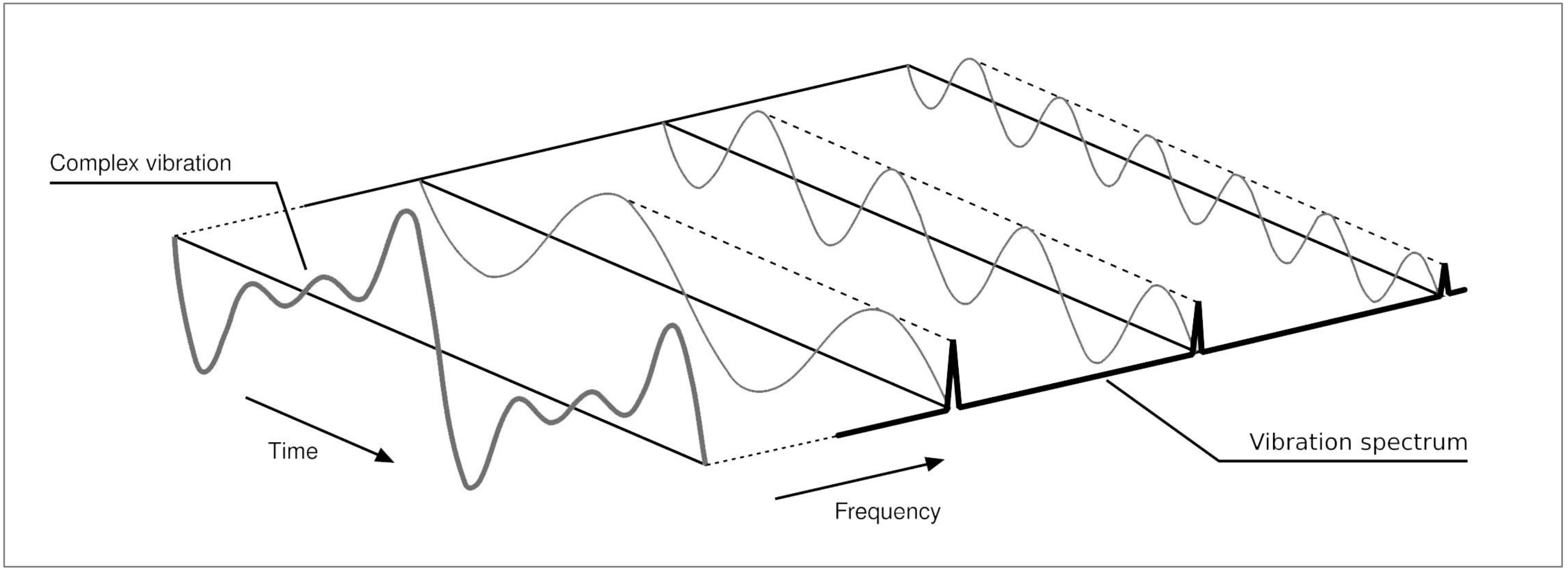
- Mimic continuous signal




**Impression**

## Natural

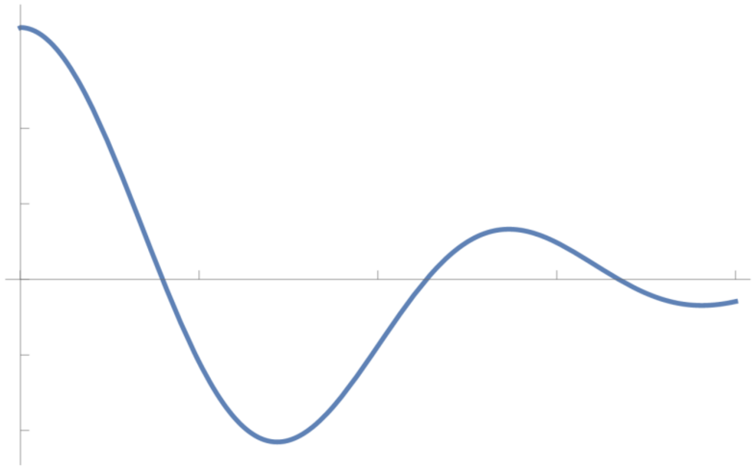
- Hopefully similar to the original signal, no artifacts



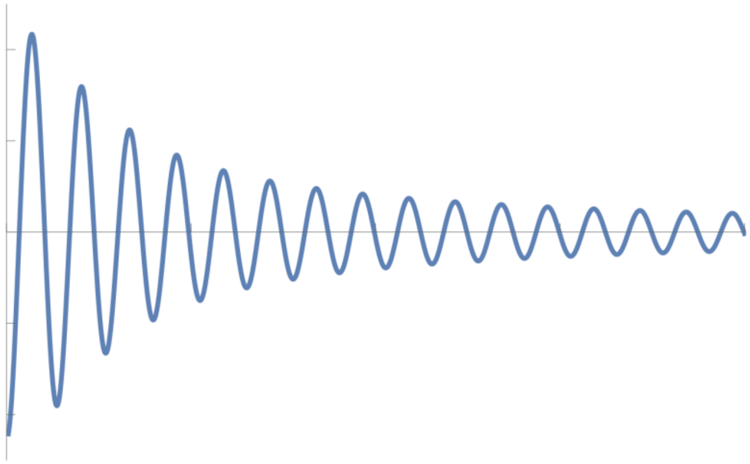

Fourier Transformation

# Fourier analysis

- Characterize a signal (function) by its frequency
  - The higher frequency a function is, the more quickly it varies over a given region
- Fourier transform represents a function in the *frequency domain*



Low-frequency function



High-frequency function

# Fourier analysis

- Any periodic, continuous function can be represented as the sum of sine or cosine functions

$$f(x) = \sum_k a_k \cos\frac{2\pi}{\tau}kx + b_k \sin\frac{2\pi}{\tau}kx$$

Euler's formula:

$$e^{ix} = \cos x + i \sin x$$

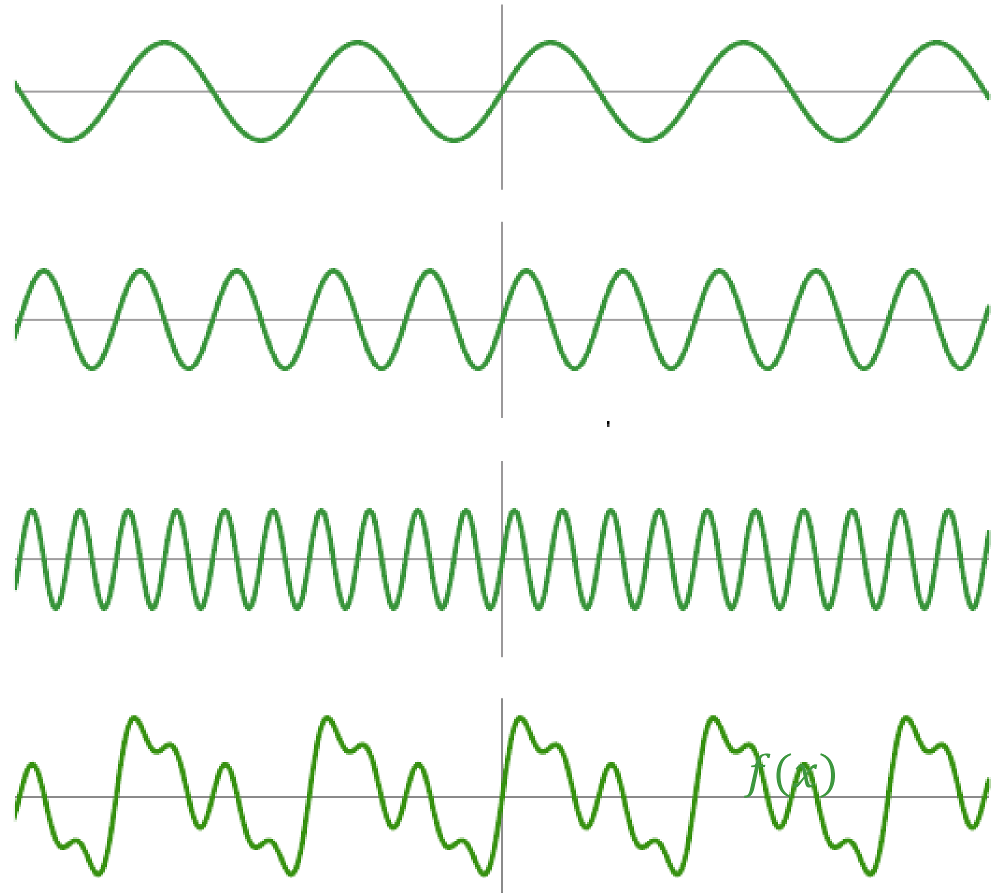
here $e$ is the base of the natural logarithm and $i = \sqrt{-1}$ is the imaginary unit. Cosine is thus on the real axis and sine on the imaginary one.

$f_1(x) = \sin(\pi x)$

$f_2(x) = \sin(2\pi x)$

$f_4(x) = \sin(4\pi x)$

$f(x) = f_1(x) + 0.75\, f_2(x) + 0.5\, f_4(x)$

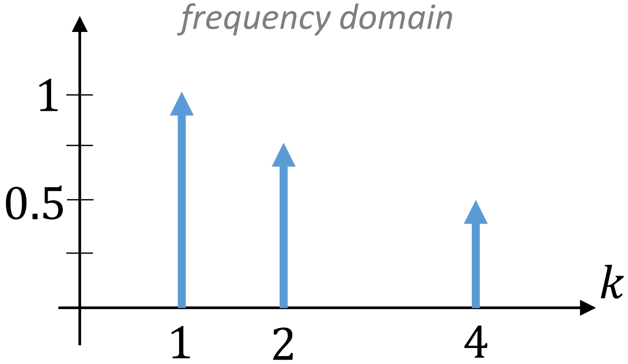


# Fourier analysis

- Any periodic, continuous function can be represented as the sum of sine or cosine functions

$$f(x) = \sum_k F(k) e^{i\frac{2\pi}{\tau}kx}$$

*frequency domain*




- Splitting a function into frequency components
- Works well even for non-ideal signals
- Discrete signals (*e.g.* images) are inherently limited in bandwidth

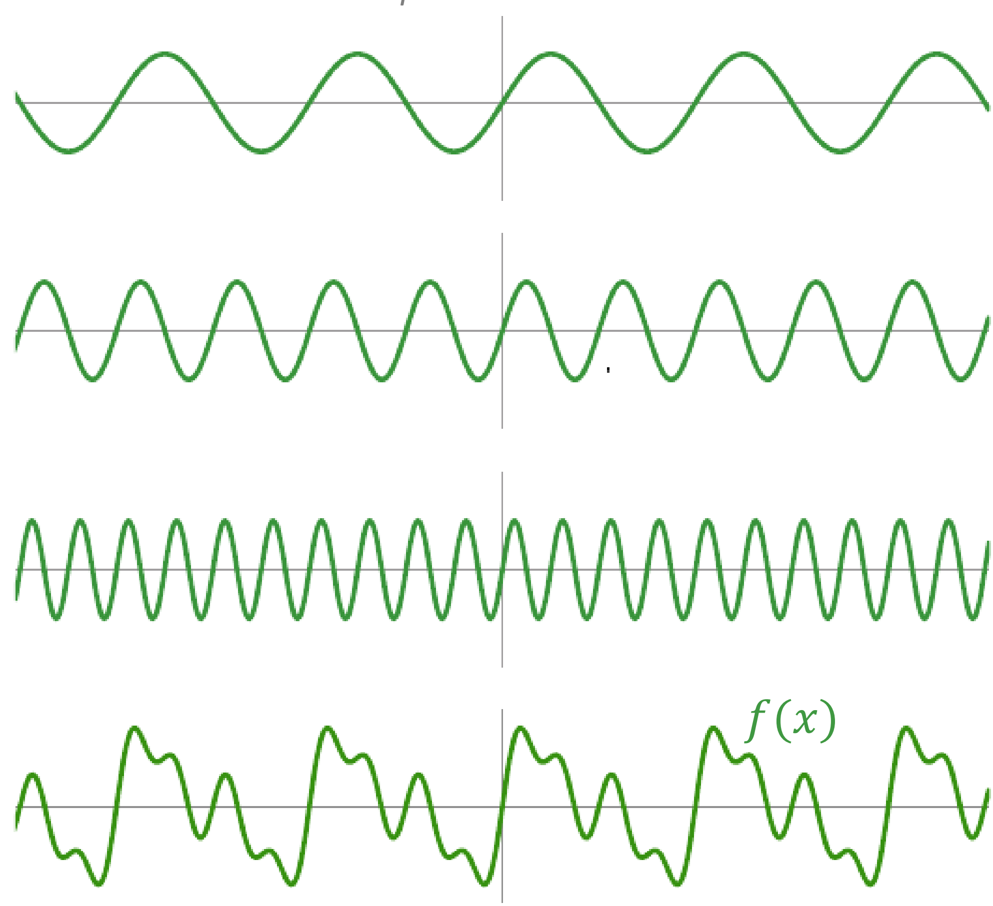
*spatial domain*

$f_1(x) = \sin(\pi x)$

$f_2(x) = \sin(2\pi x)$

$f_4(x) = \sin(4\pi x)$

$f(x) = f_1(x) + 0.75\, f_2(x) + 0.5\, f_4(x)$



# Fourier analysis

- Any periodic, continuous function can be represented as the sum of sine or cosine functions

$$f(x) = \sum_k a_k \cos\frac{2\pi}{\tau}kx + b_k \sin\frac{2\pi}{\tau}kx = \sum_k F(k)e^{i\frac{2\pi}{\tau}kx}$$

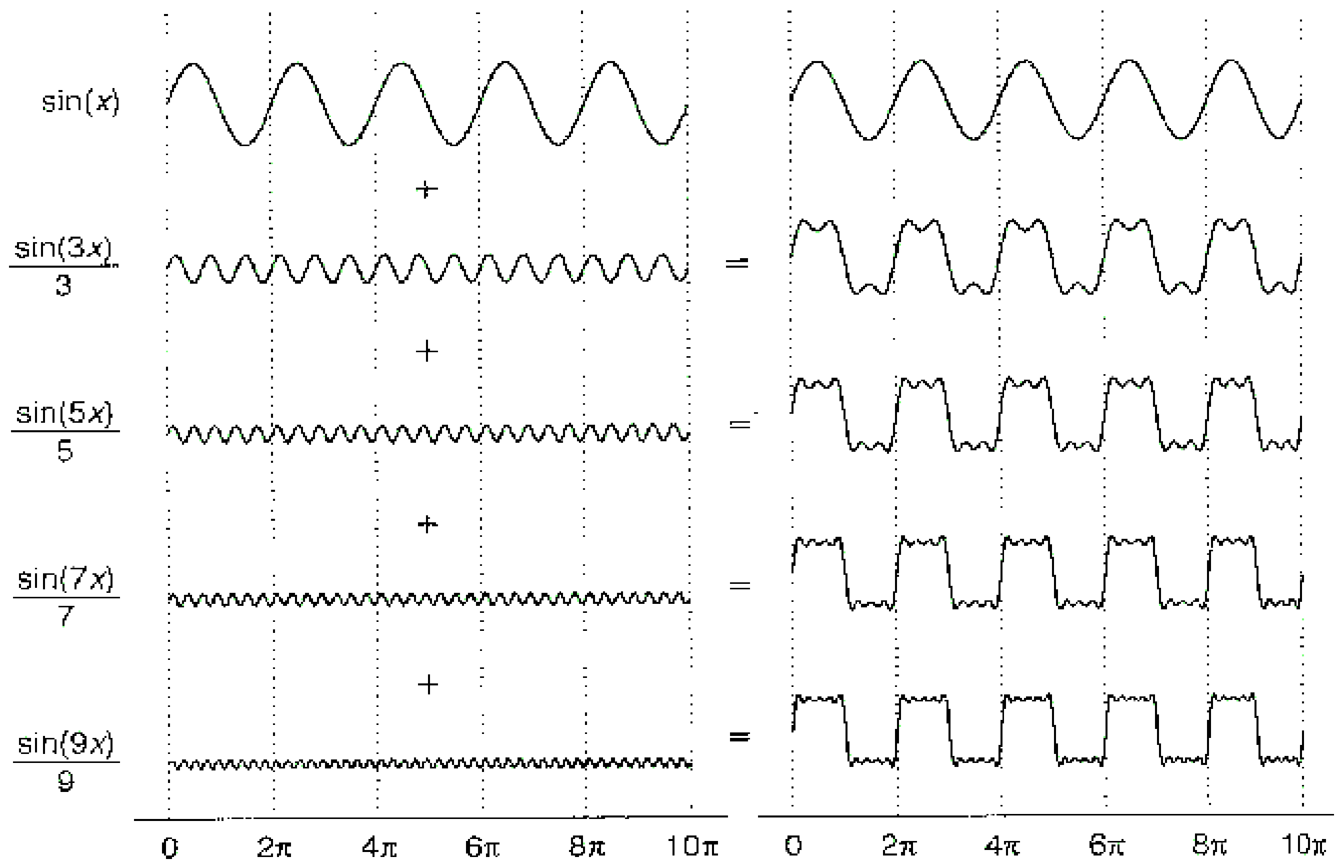
- $k$: frequency band
  - $k = 0$ mean value
  - $k = 1$ function period, lowest possible frequency
  - $k = 1{,}5?$ not possible
  - $k_{max}?$ band limit, no higher frequency present in signal
- $a_k, b_k$: Fourier coefficients
- Odd function $f(-x) = -f(x)$
  - $a_k = 0$
  - $f(x) = \sum_k b_k \sin\frac{2\pi}{\tau}kx$
- Even function $f(-x) = f(x)$:
  - $b_k = 0$
  - $f(x) = \sum_k a_k \cos\frac{2\pi}{\tau}kx$

# Example:

- Square wave: a perioding odd function

$$f(x) = \begin{cases} 0.5 & \forall \quad 0 < (x \; mod \; 2\pi) < \pi \\ -0.5 & \forall \quad \pi < (x \; mod \; 2\pi) < 2\pi \end{cases}$$

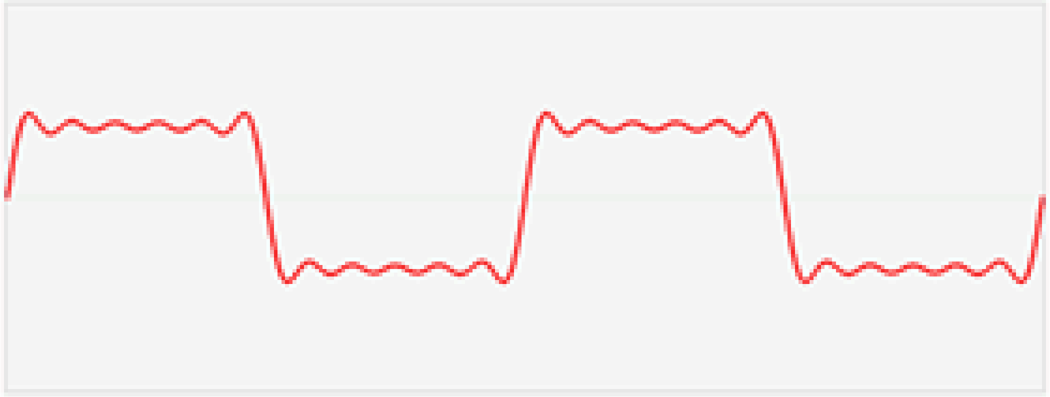
- $b_0 = 0$
- $b_1 = 1$
- $b_2 = 0$
- $b_3 = {}^1/_3$
- $b_4 = 0$
- $b_5 = {}^1/_5$
- $b_6 = 0$
- $b_7 = {}^1/_7$
- $b_8 = 0$
- $b_9 = {}^1/_9$
- …



## Example:

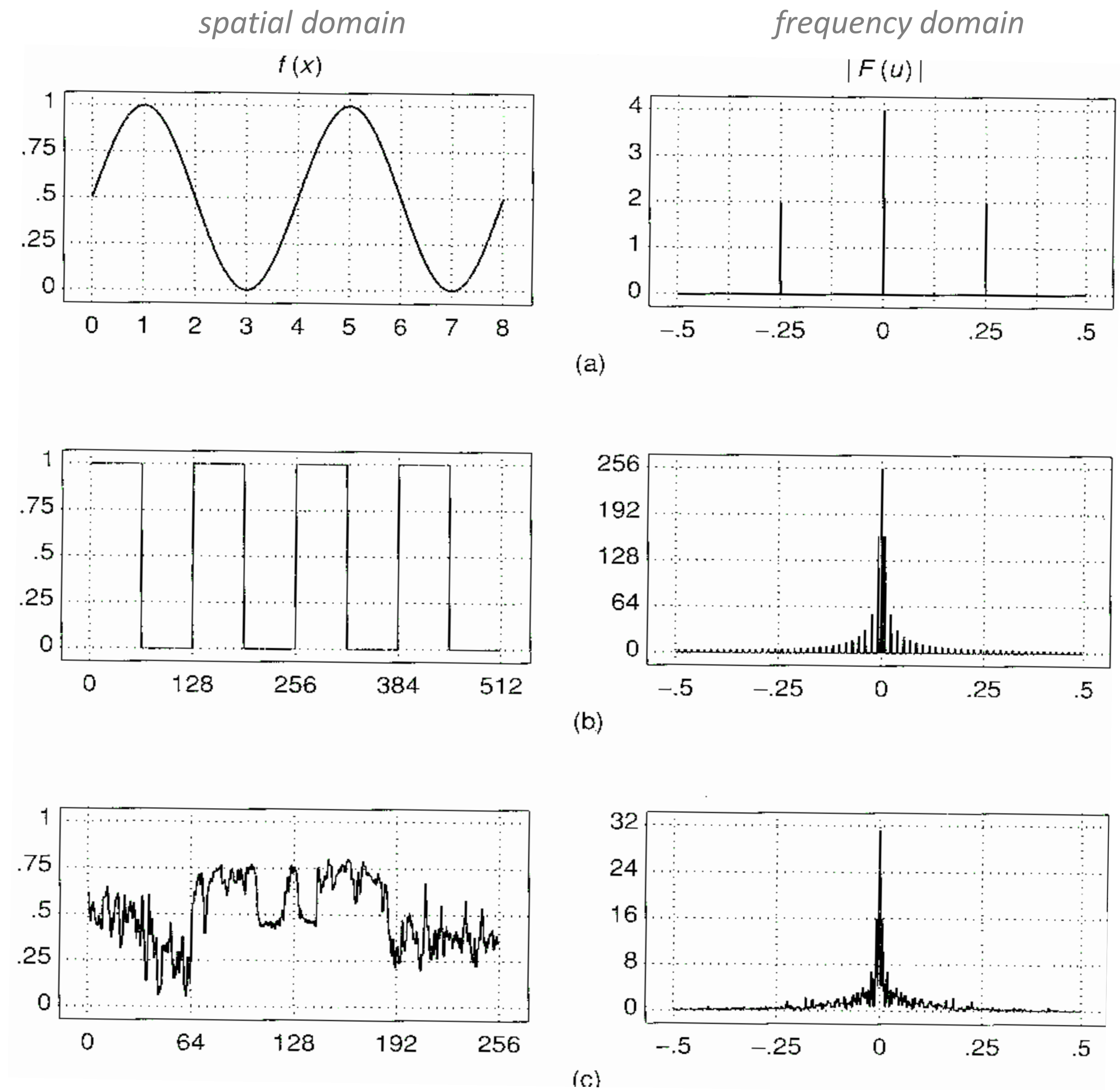
- Square wave: a perioding odd function

$$f(x) = \begin{cases} 0.5 & \forall \quad 0 < (x\ mod\ 2\pi) < \pi \\ -0.5 & \forall \quad \pi < (x\ mod\ 2\pi) < 2\pi \end{cases}$$



# More Examples

- Sine wave with positive offset
- Square wave
- Scanline of an image



# Important basis functions

- Box ⟺ sinc (*sinus cardinalis*)

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

$$\mathrm{sinc}(0) = 1$$

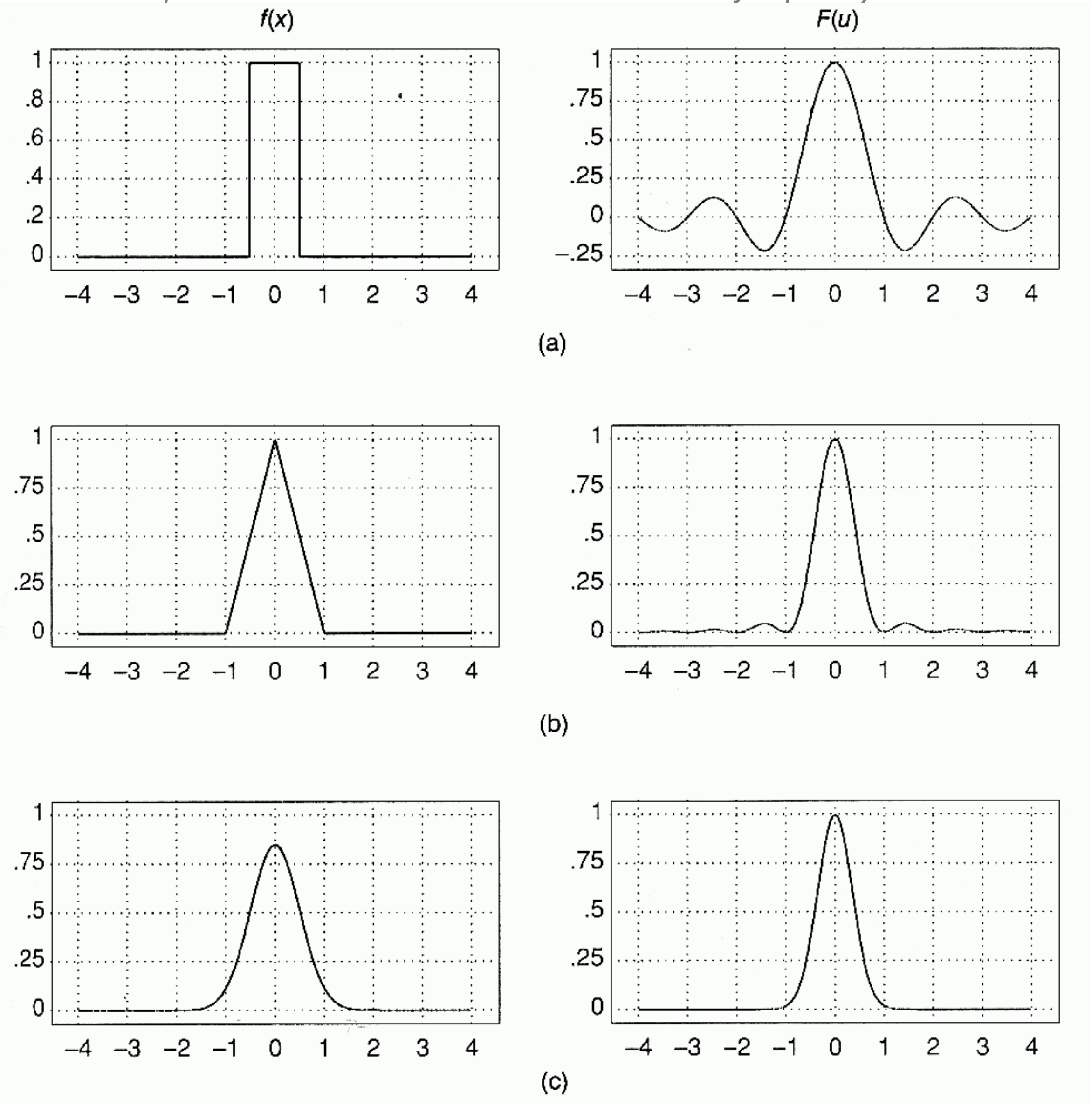
$$\int \mathrm{sinc}(x)dx = 1$$

  - Wide box ⇒ small sinc
  - Negative values
  - Infinite support

- Triangle ⟺ $\mathrm{sinc}^2$

- Gauss ⟺ Gauss

*spatial domain* *frequency domain*



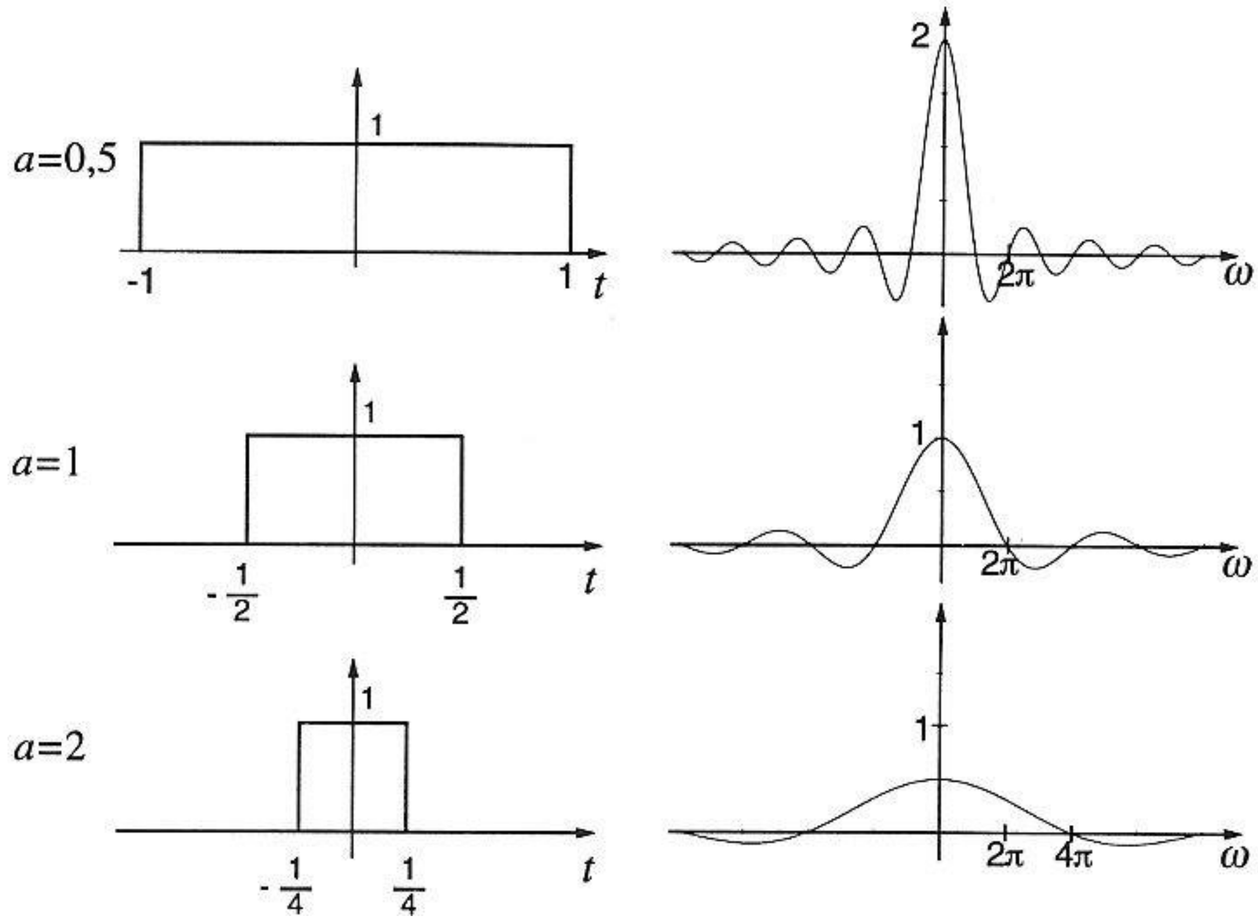
## Important basis functions

- Example box function transform behavior

- Fourier transform: sinc

- Wide box: narrow sinc

- Narrow box: wide sinc



## Transformation from the **frequency domain** to the **spatial domain**

- This formula is also knows as "*Fourier synthesis* equation" or *inverse Fourier transform*

$$f(x) = \sum_k F(k) e^{i\frac{2\pi}{\tau}kx}$$

## Transformation from the **spatial domain** to the **frequency domain**

- The reverse operation is known as the "*Fourier analysis* equation" or *Fourier Transform*:

$$F(k) = \sum_x f(x) e^{-i\frac{2\pi}{\tau}kx}$$

## Transformation from the **frequency domain** to the **spatial domain**

- This formula is also knows as "*Fourier synthesis* equation" or ***inverse Fourier transform***:

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega x} d\omega$$

## Transformation from the **spatial domain** to the **frequency domain**

- The reverse operation is known as the "*Fourier analysis* equation" or ***Fourier Transform***:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx$$

# Images have discrete values and are not continuous functions

- The Fast Fourier Transform (FFT) is a divide and conquer algorithm that enables us to compute the discrete Fourier transform and its inverse in $O(n \log n)$ time where $n$ is the number of samples
- Compute the 2D Fourier transform by applying 1D discrete Fourier transforms (*i.e.* FFTs) along each dimension

# Compute the discrete Fourier transform (via FFT)

- Transform the discrete image values into another array of discrete values

# Perform operations

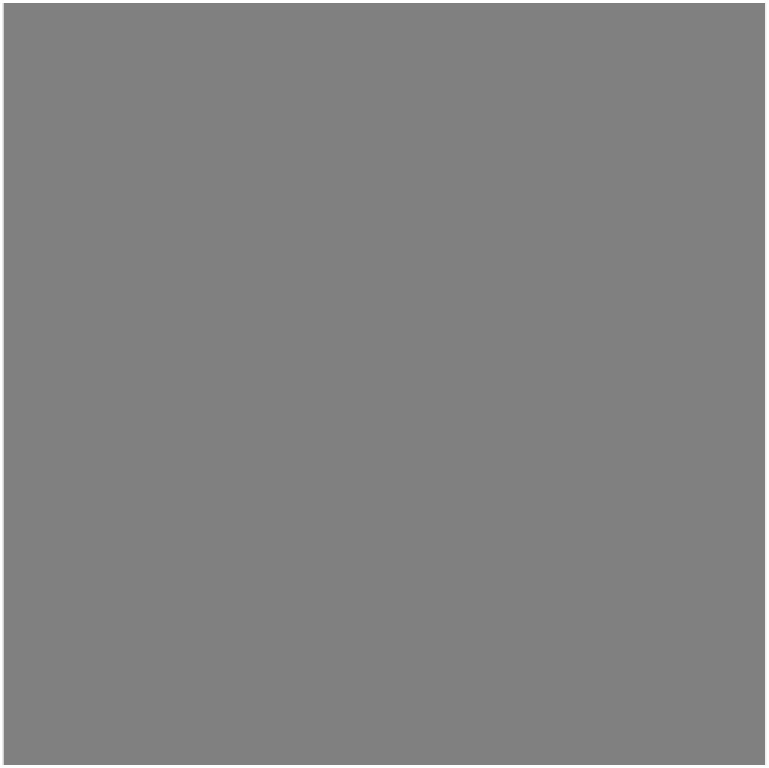
- Efficient filtering in frequency domain

# Compute the inverse discrete Fourier transform (via FFT)

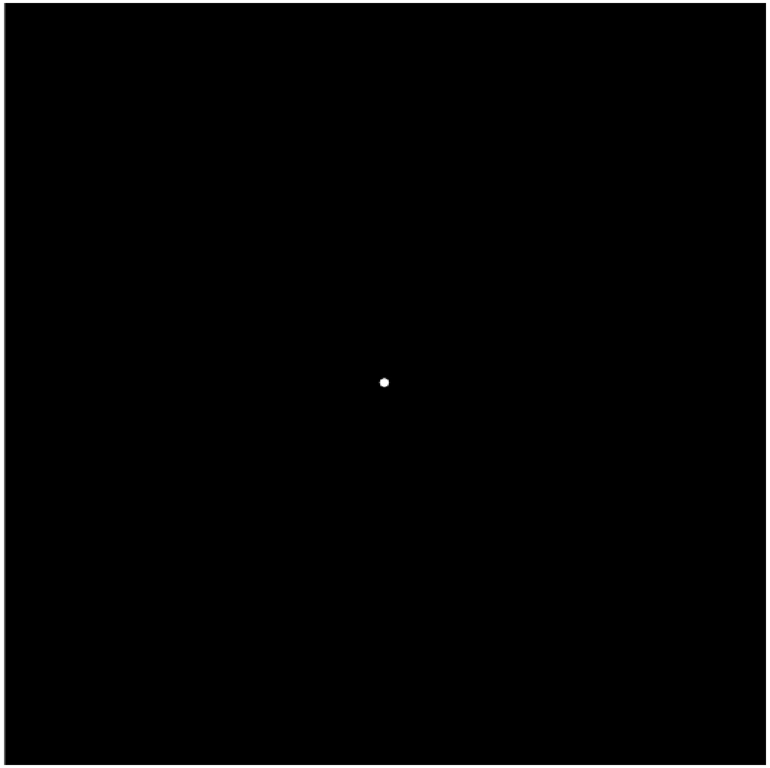
There is an opinion that FFT is the most influential algorithm!
Check out: *3 Applications of the (Fast) Fourier Transform ft. Mikhail Kapralov (Михаил Капралов)*
https://www.youtube.com/watch?v=aqa6vyGSdos

# Fourier Analysis

- On a simple input, *e.g.* grayscale image (= constant function), the output of a Fourier analysis is also simple



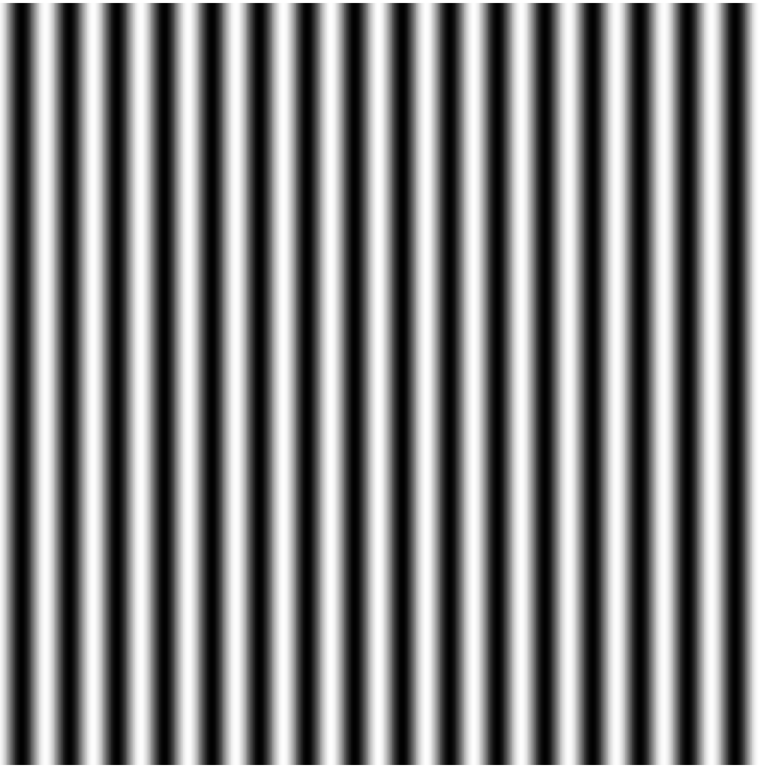
Spatial Domain



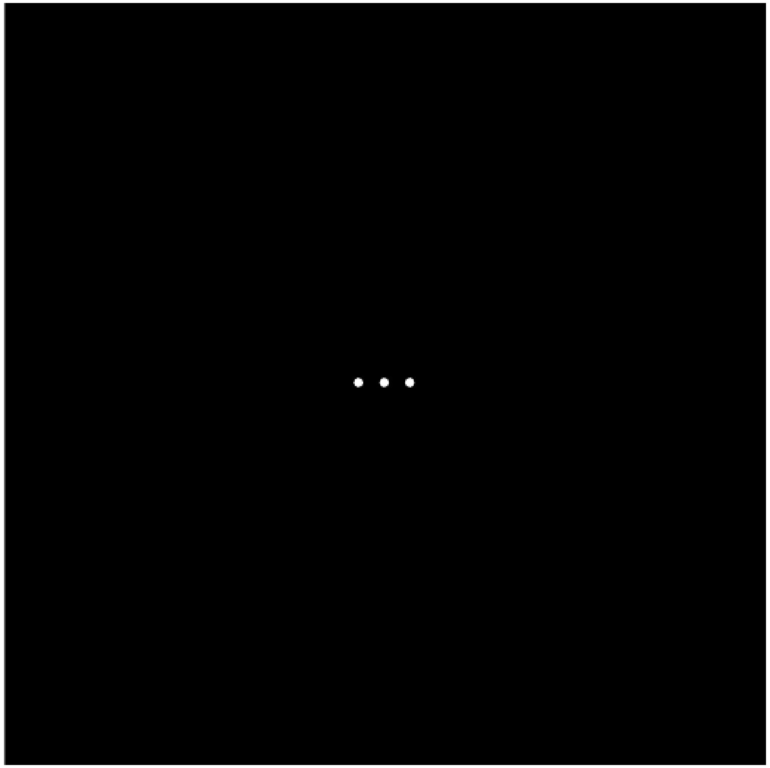
Frequency Domain

## Fourier Analysis

- Let's imagine that our image is created using a sinusoid function in the horizontal ($x$) direction: $\sin \frac{2\pi}{32} x$



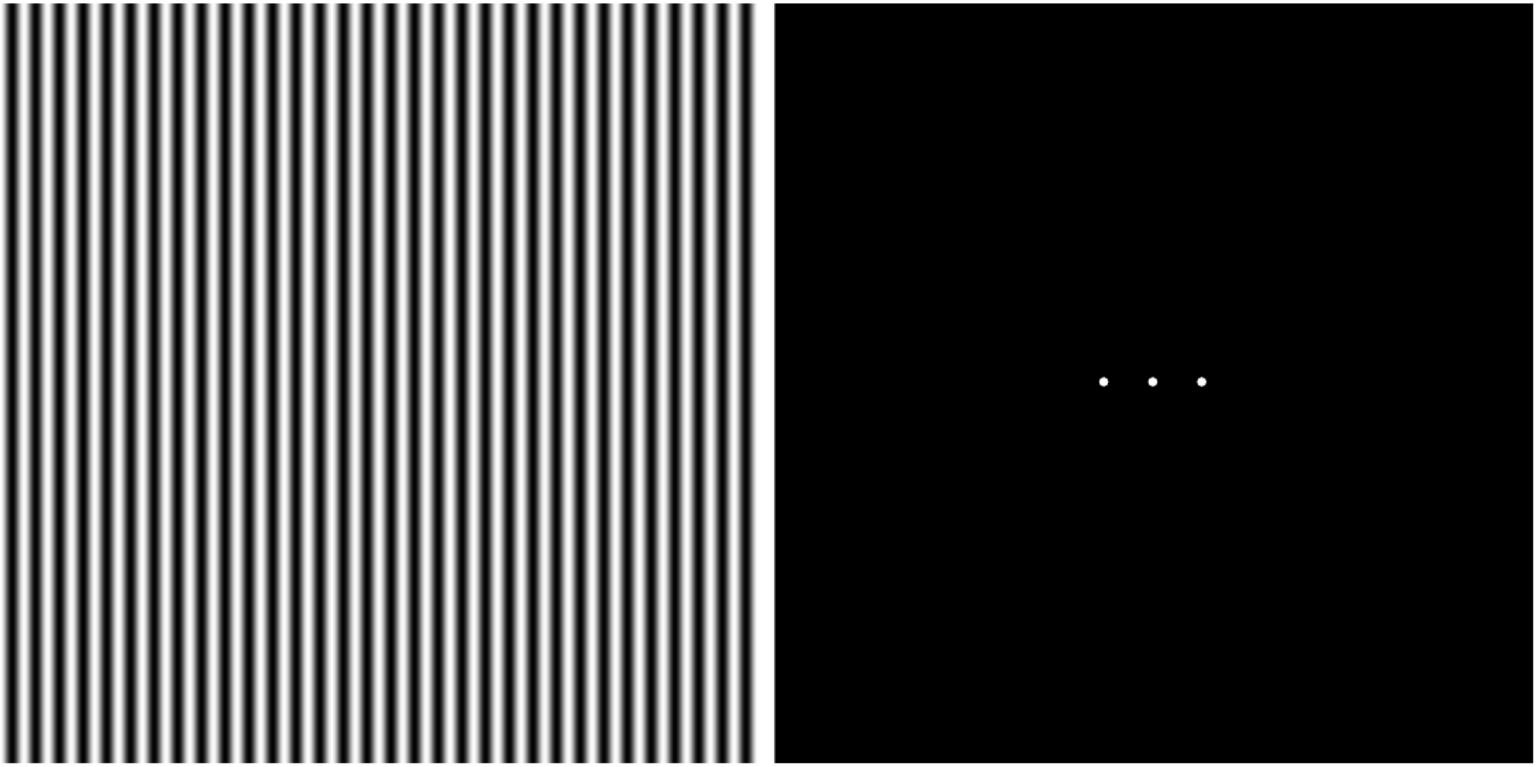
Spatial Domain



Frequency Domain

# Fourier Analysis

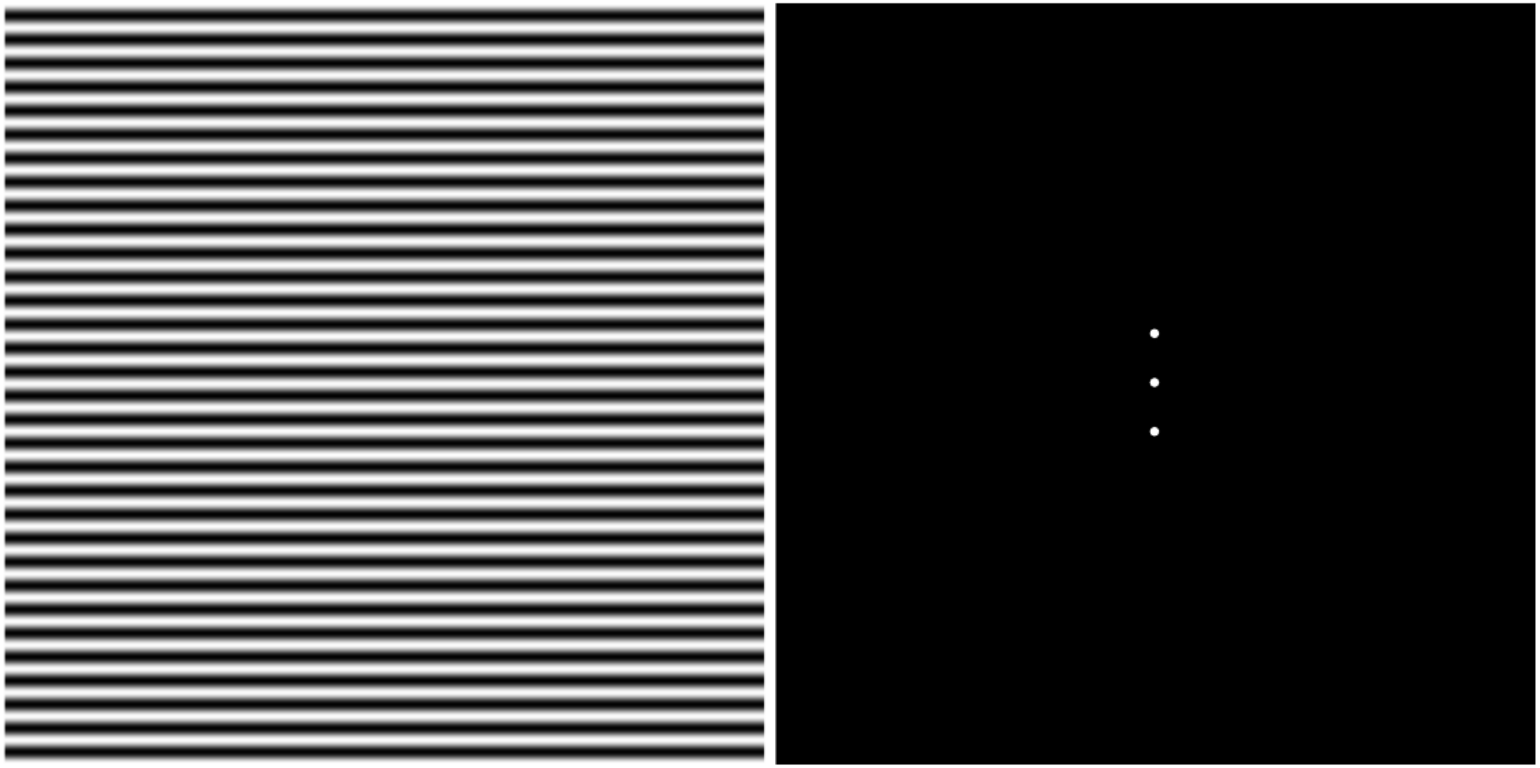
- How about a higher frequency: $\sin\frac{2\pi}{16}x$



Spatial Domain Frequency Domain

# Fourier Analysis

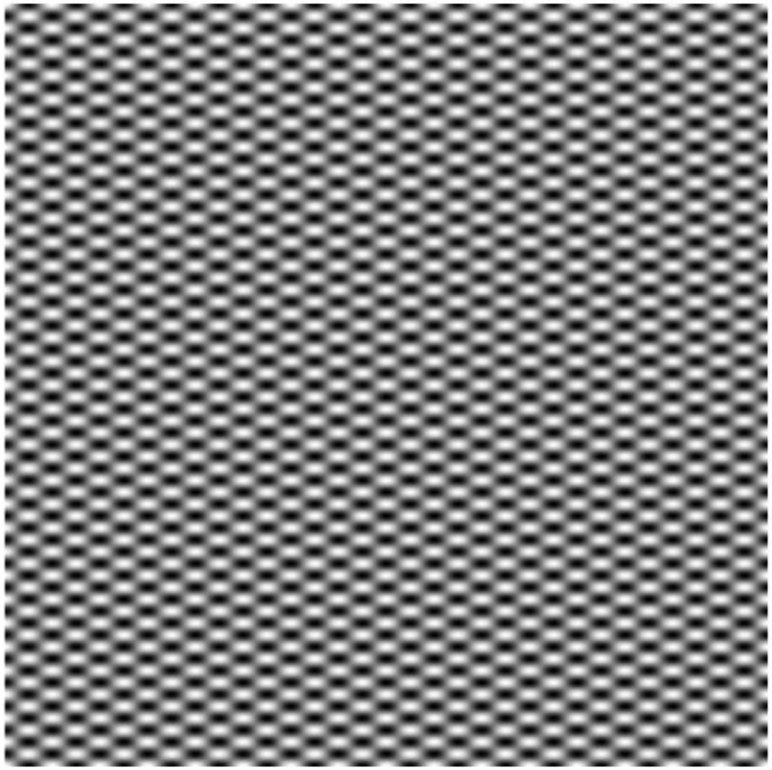
- How about applying the same pattern vertically: $\sin\frac{2\pi}{16}y$



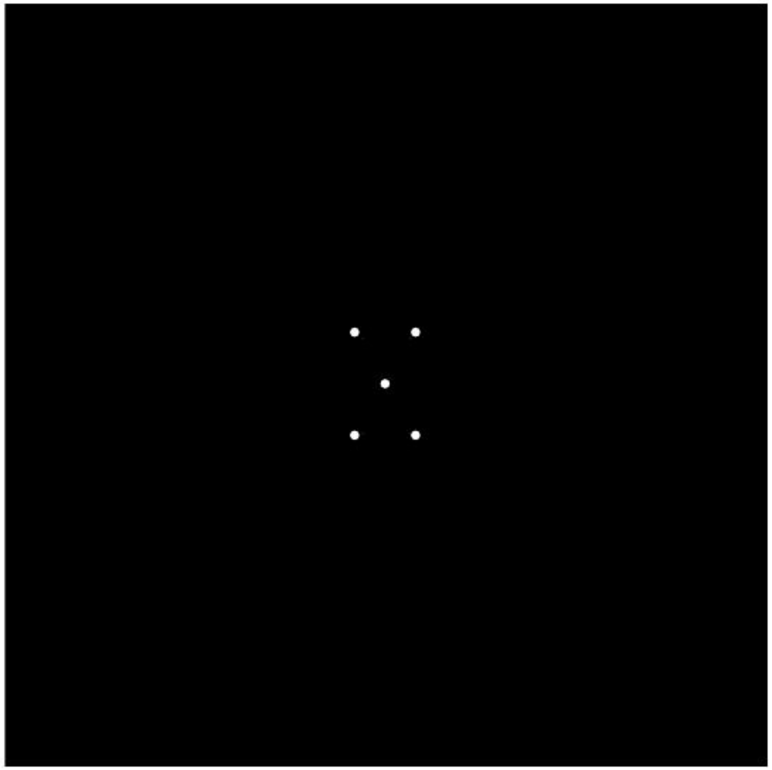
Spatial Domain    Frequency Domain

# Fourier Analysis

- Different sinusoidal components applied in two directions: $\sin\frac{2\pi}{32}x \times \sin\frac{2\pi}{16}y$



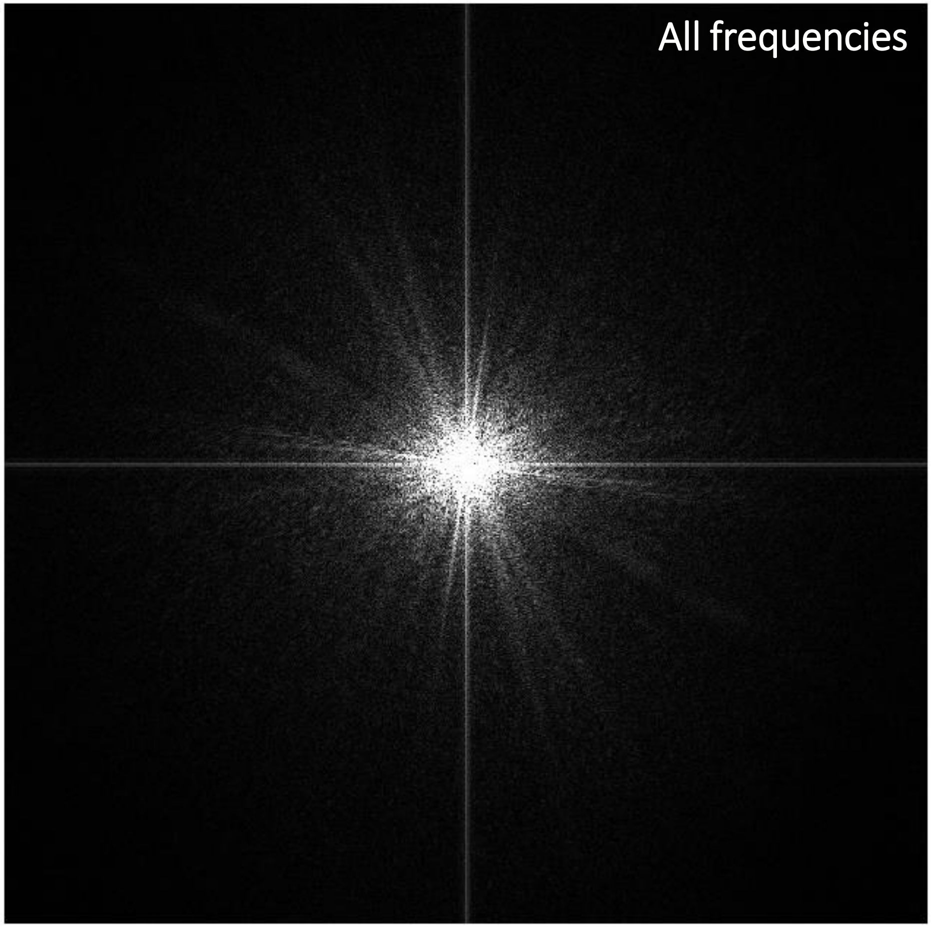
Spatial Domain



Frequency Domain

# Frequencies in Image

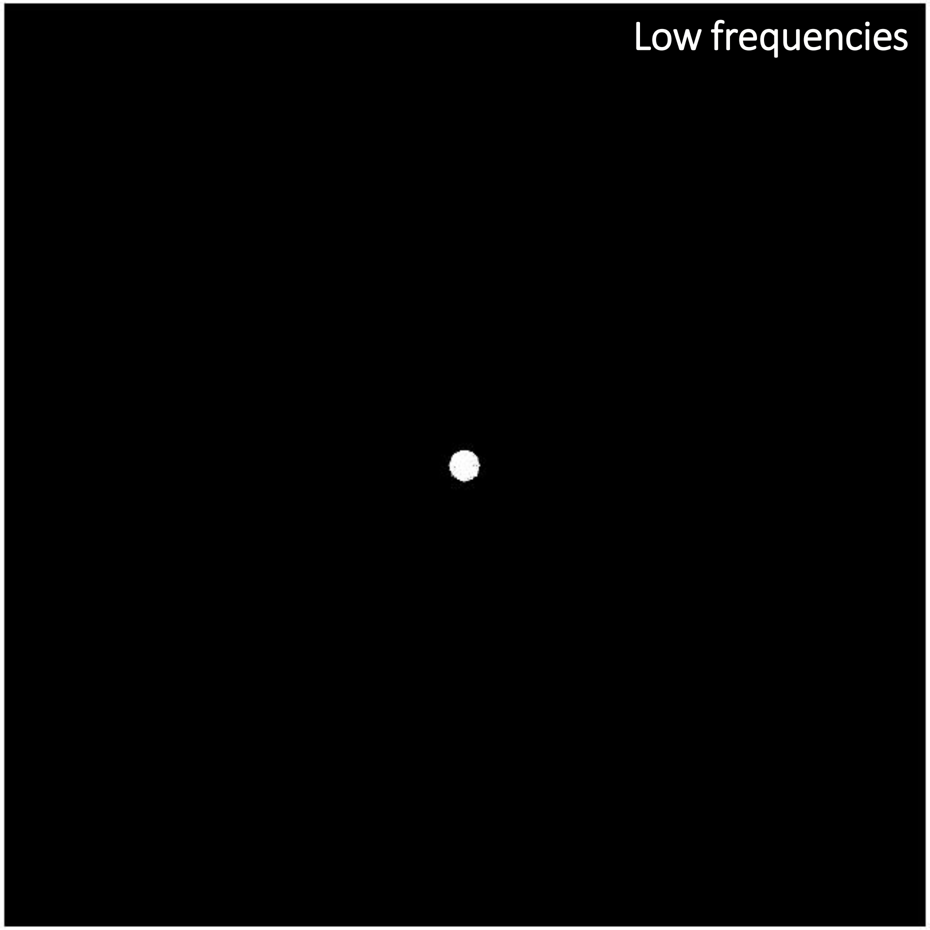
Spatial domain result




Spectrum

# Frequencies in Image

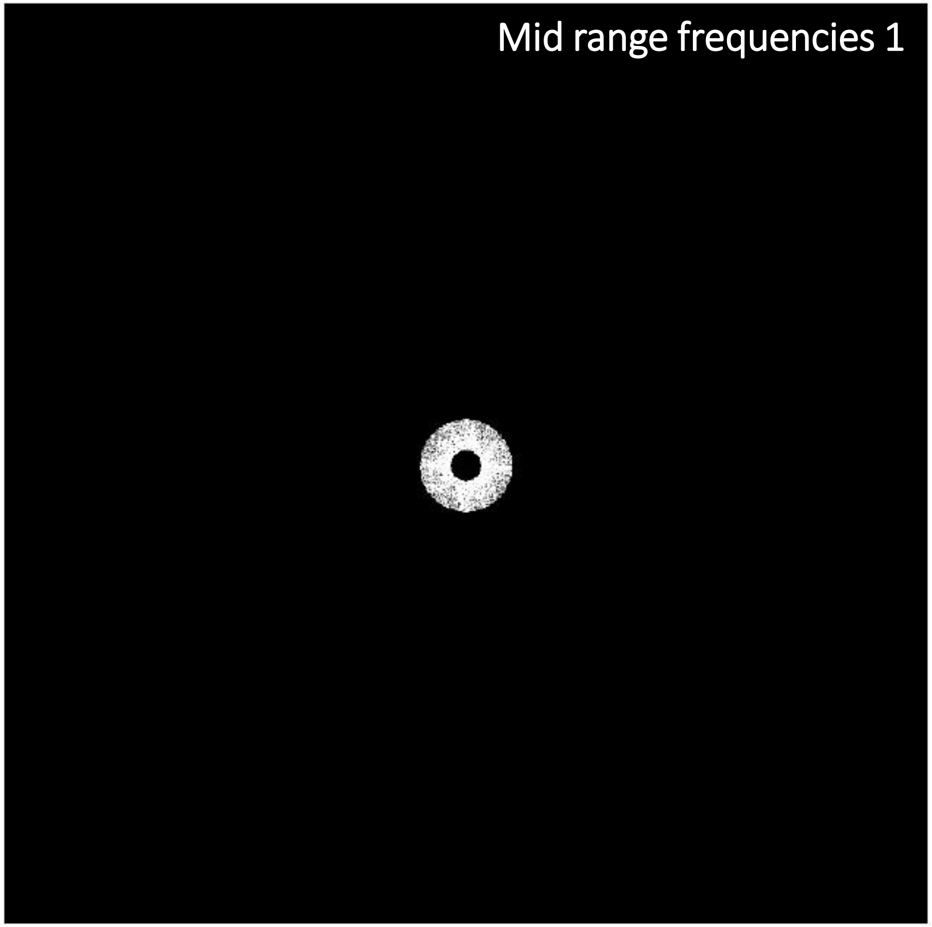
Spatial domain result




Spectrum (after low-pass filter)
All frequencies above cutoff have 0 magnitude

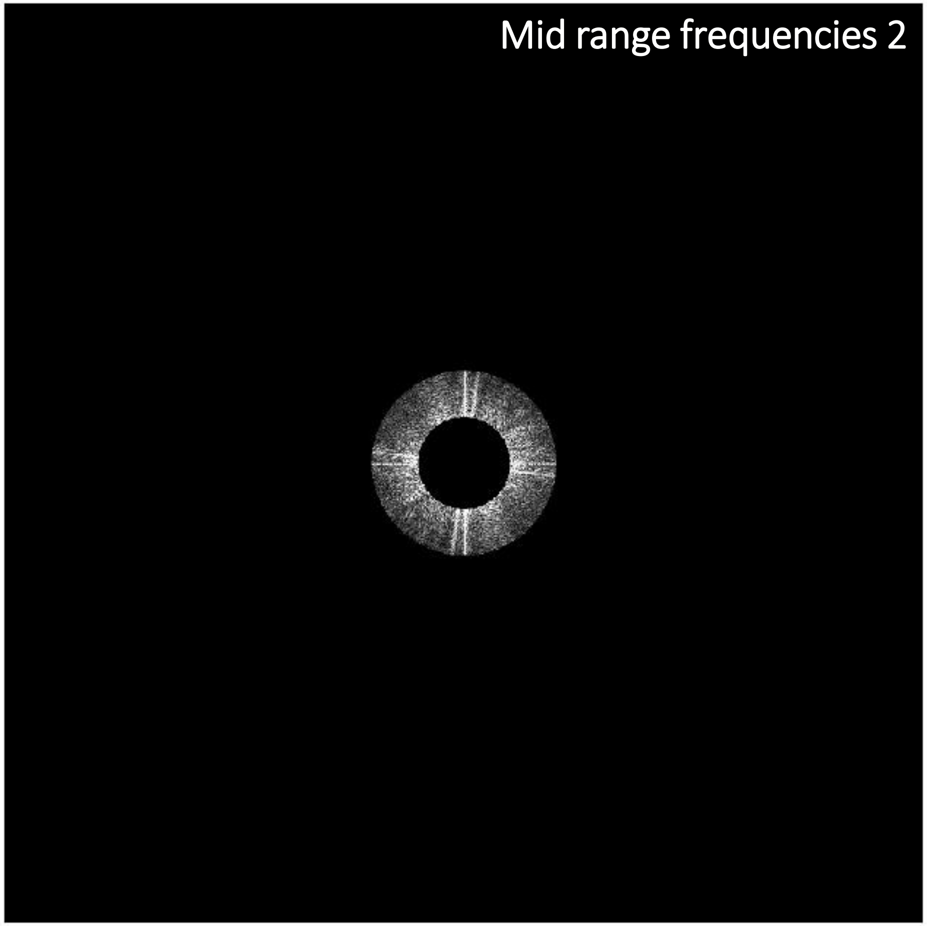
Let’s eliminate some frequencies present in these images!

Spatial domain result



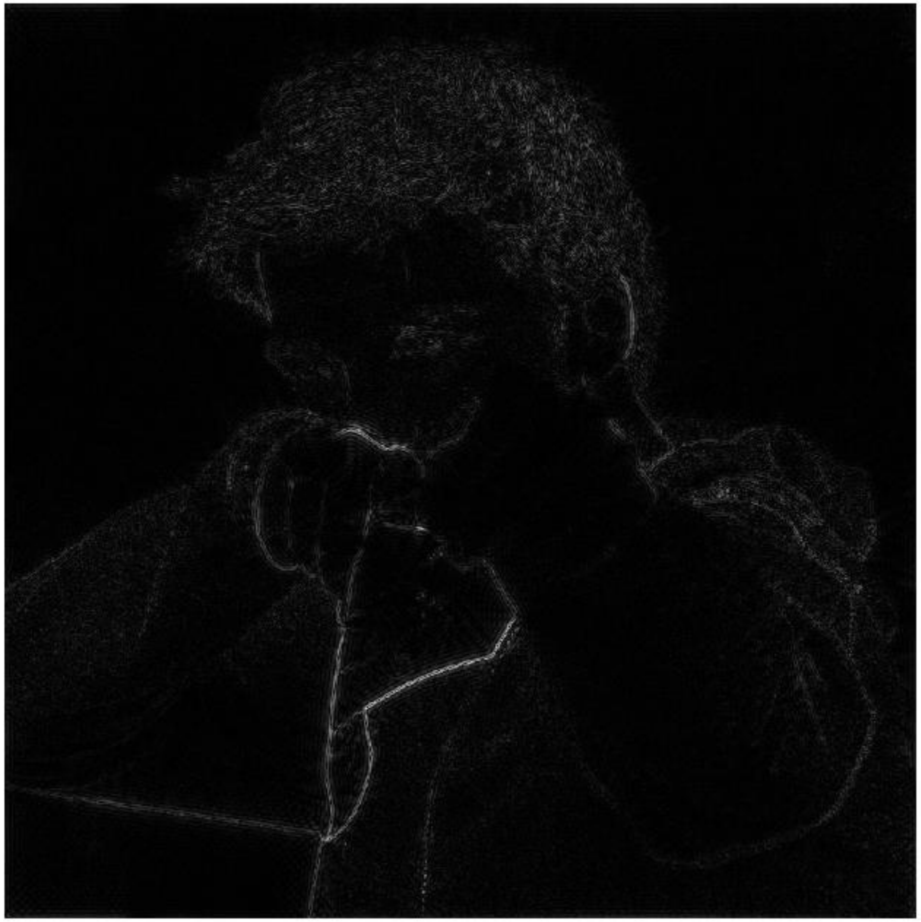

Spectrum (after band-pass filter)

Spatial domain result

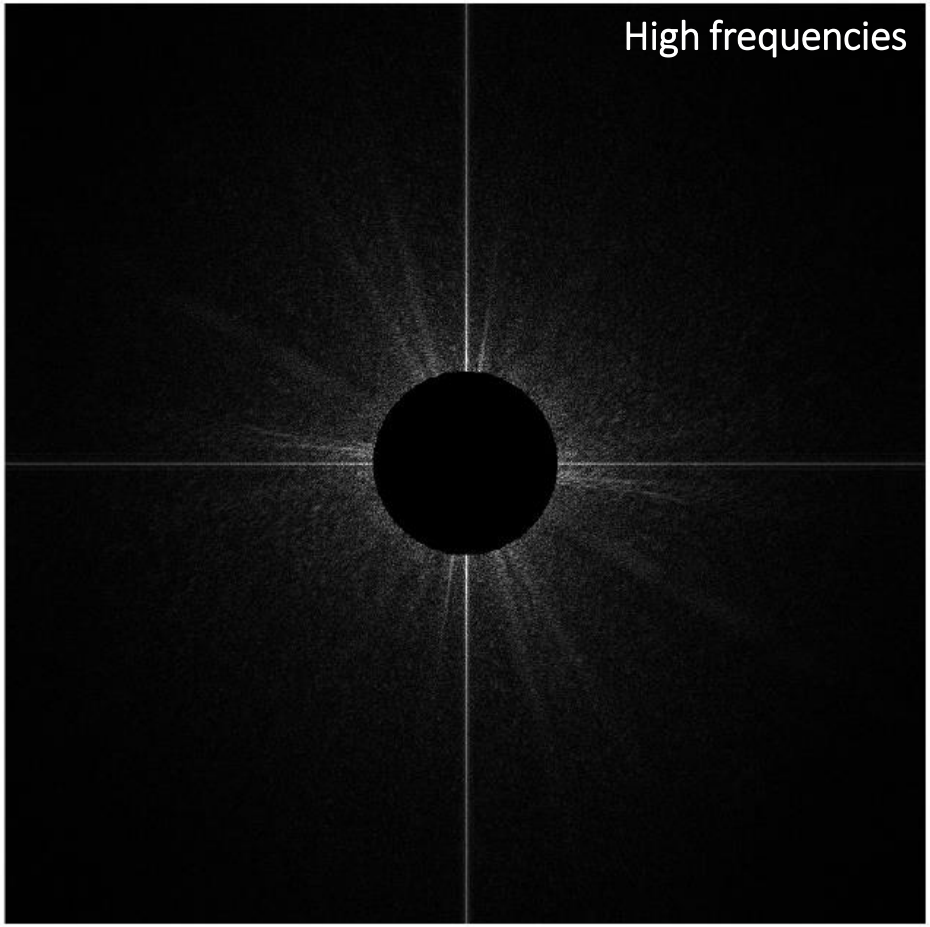



Spectrum (after band-pass filter)

# Frequencies in Image

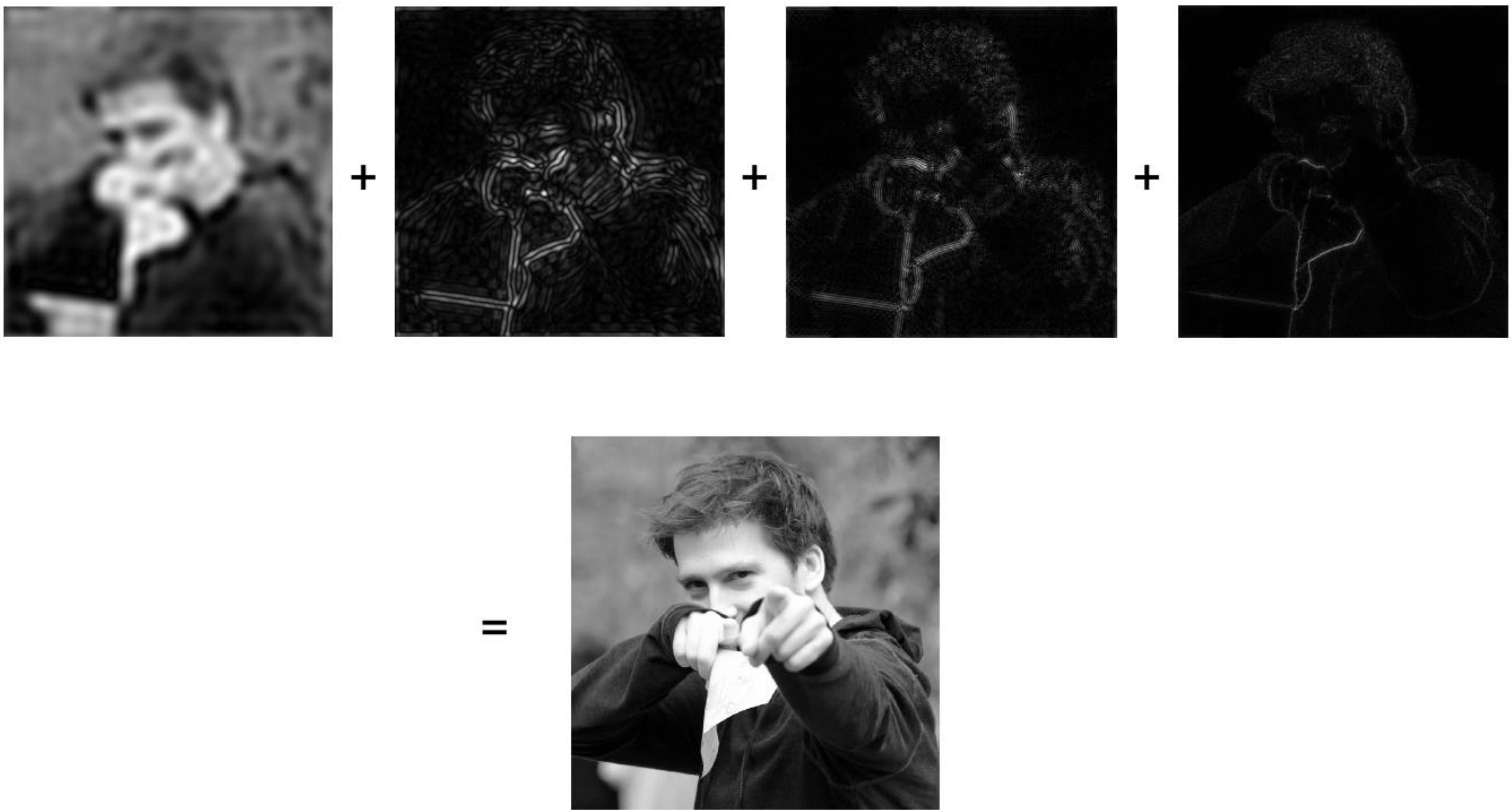


Spatial domain result
(strongest edges)




Spectrum (after high-pass filter)
All frequencies below threshold
have 0 magnitude

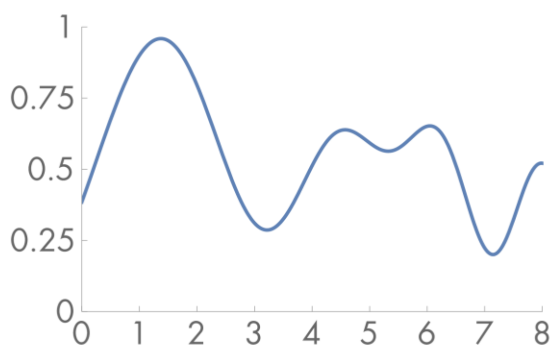
## The sum is the image itself!



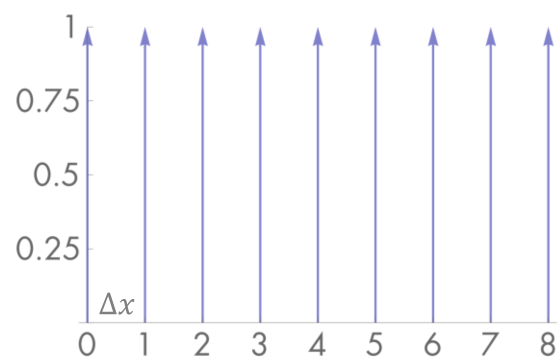
# Formal definition of sampling

- Sampling process requires us to choose a set of equally spaced sample positions and compute the function's value at those positions
- Formally, this corresponds to multiplying the function by a "*shah*" or "*impulse train*" function (an infinite sum of equally spaced delta Dirac functions)

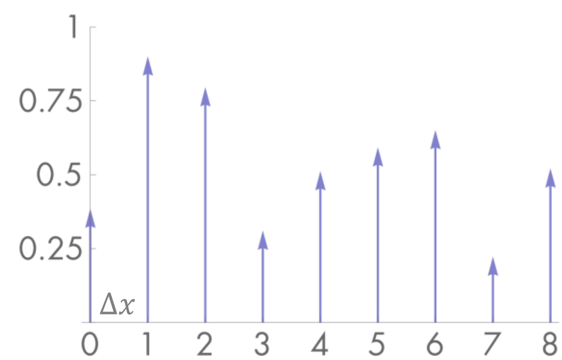
$$Ш_{\Delta x}(x) = \Delta x \sum_{k=-\infty}^{\infty} \delta(x - k\Delta x)$$




$f(x)$




$Ш_{\Delta x}(x)$




$Ш_{\Delta x}(x) f(x)$

$$Ш_{\Delta x}(x) f(x) = \Delta x \sum_{k=-\infty}^{\infty} \delta(x - k\Delta x) f(k\Delta x)$$

# Formal definition of reconstruction

- The sample values are used to calculate a reconstructed function $\tilde{f}(x)$ by choosing a reconstruction filter function $r(x)$ and computing the *convolution*

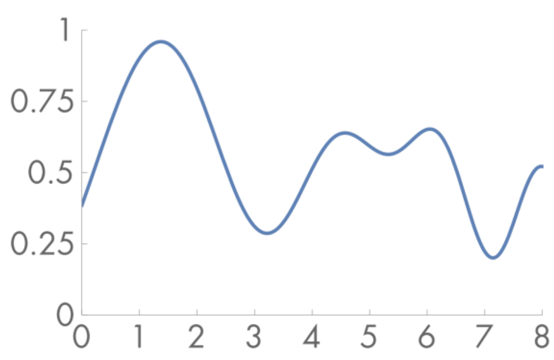
$$\tilde{f}(x) = \left(\text{Ш}_{\Delta x}(x) f(x)\right) * r(x)$$

- The convolution is defined as

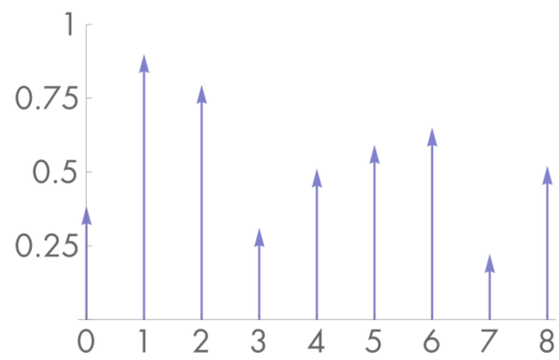
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(t) g(x-t) dt$$

- For reconstruction, convolution gives a weighted sum of scaled instances of the reconstruction filter centered at the sample points:

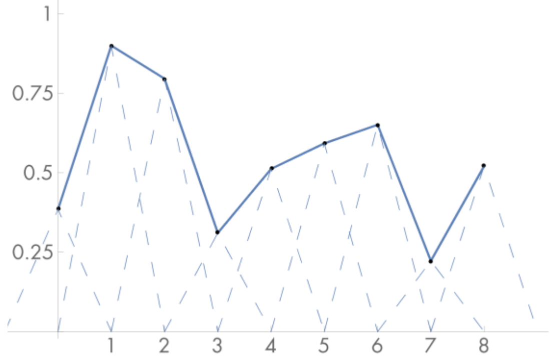
$$\tilde{f}(x) = \Delta x \sum_{k=-\infty}^{\infty} f(k\Delta x) r(x - k\Delta x)$$




$f(x)$




$\text{Ш}_{\Delta x}(x) f(x)$

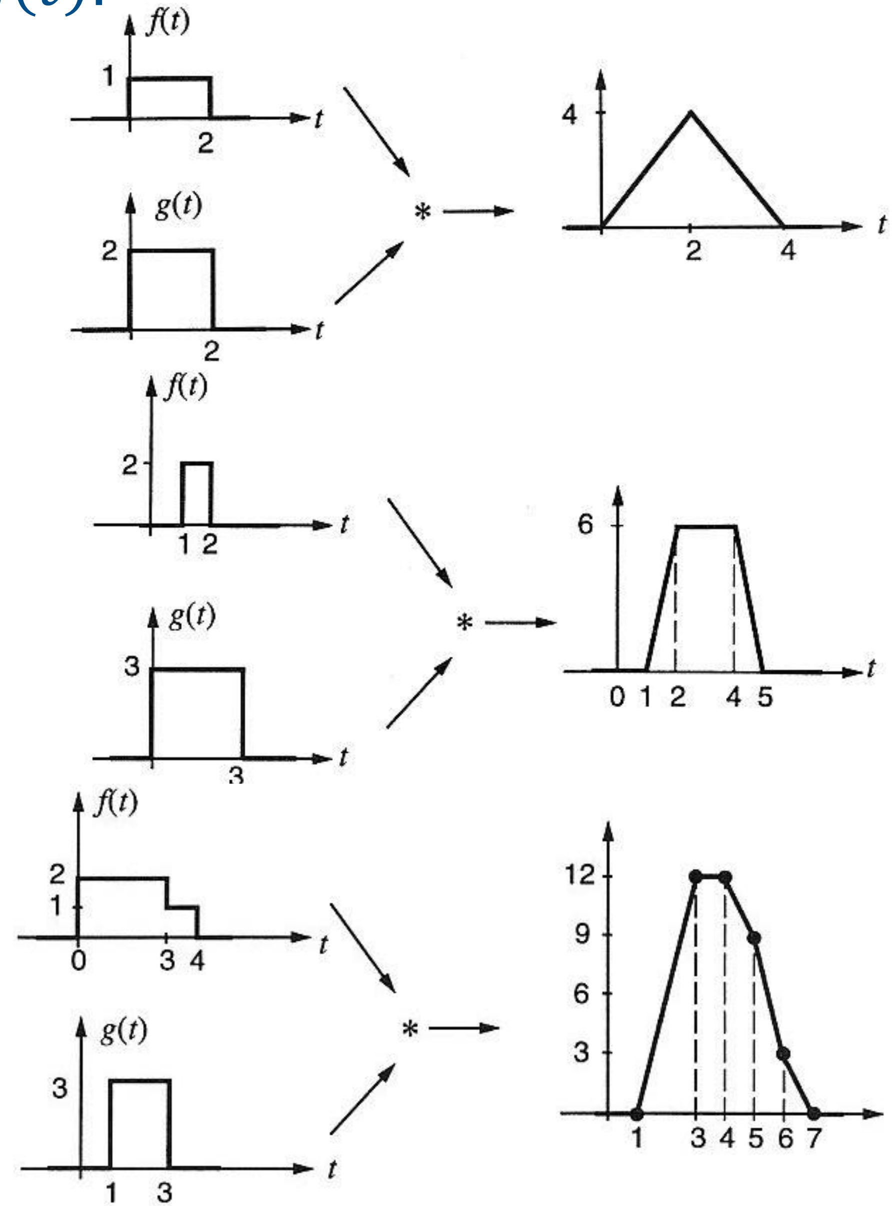



$\tilde{f}(x)$

# Convolution of two functions $f(t)$ and $g(t)$:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dx$$

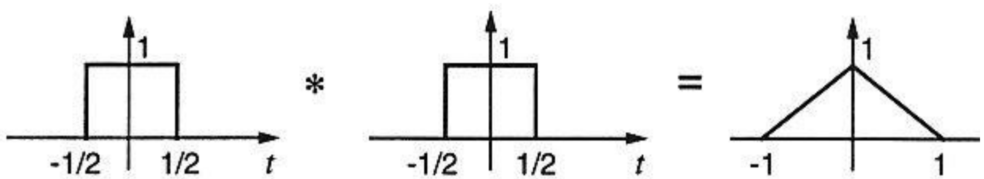
- Shift one function against  the other by $x$
- Multiply function values
- Integrate overlapping region
- Numerical convolution: Expensive operation
  - For each $x$: integrate over non-zero domain



## Convolution in image domain: multiplication in Fourier domain

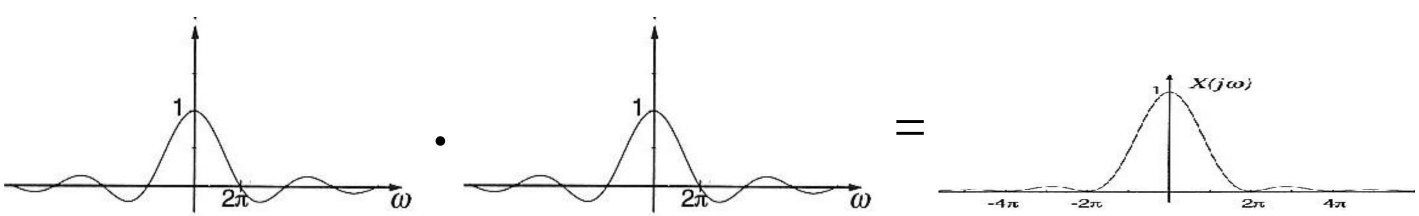
## Convolution in Fourier domain: multiplication in image domain

- Multiplication much cheaper than convolution!

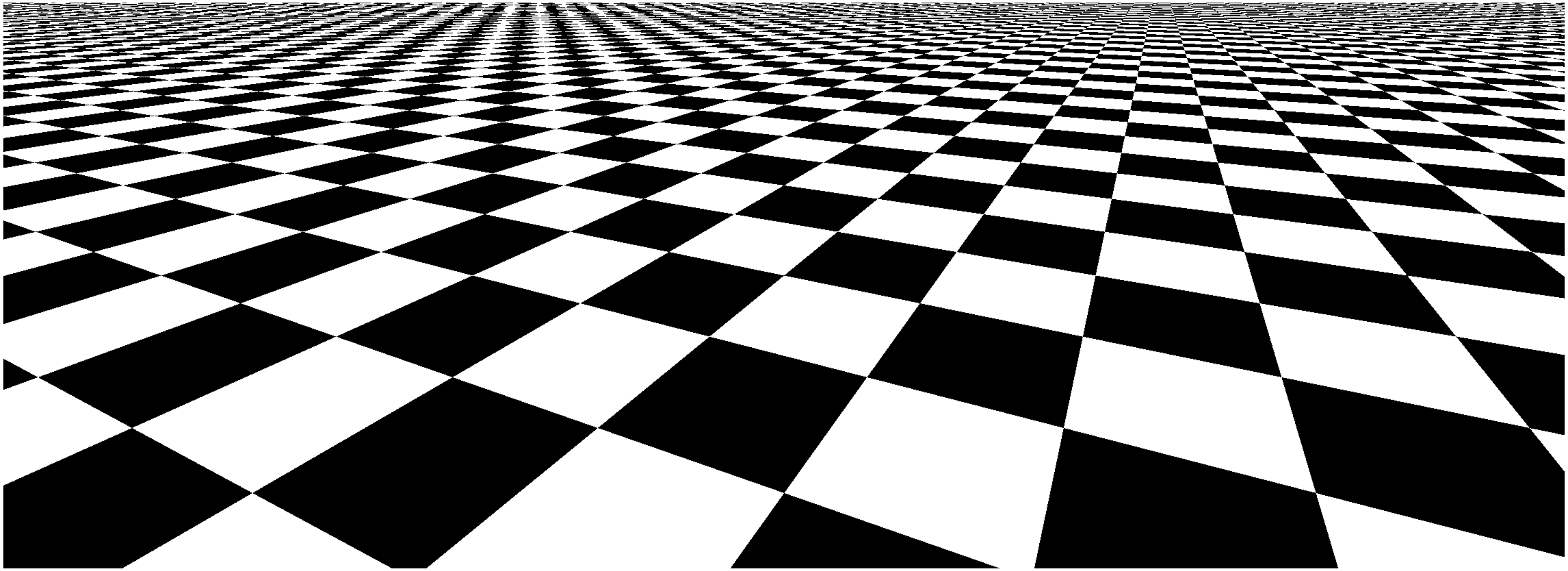



$$x(t) = \begin{cases} t+1 & \text{für } -1 \leq t < 0 \\ -t+1 & \text{für } 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases}$$

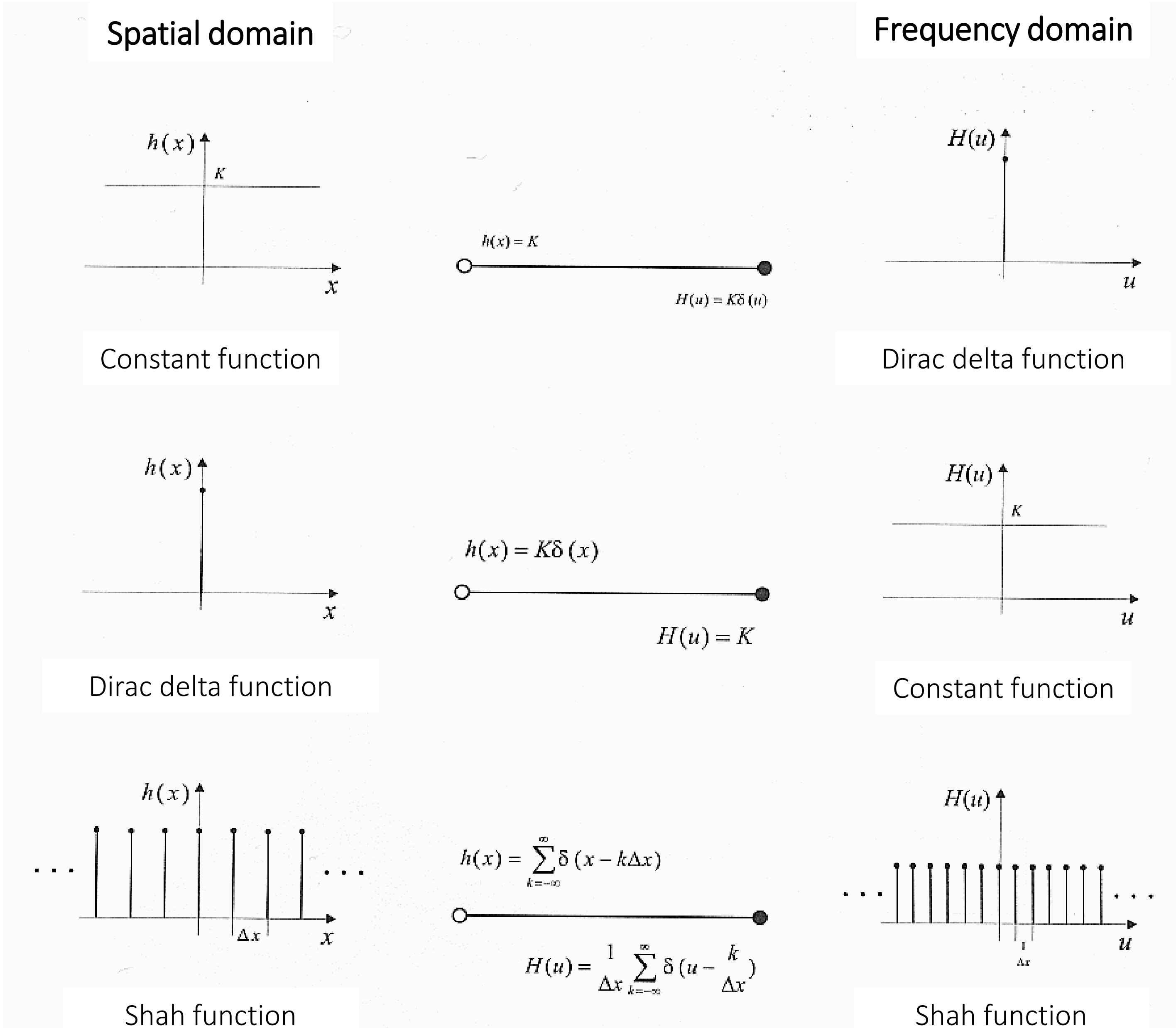
$$\begin{array}{ccccc} \mathrm{rect}(t) & * & \mathrm{rect}(t) & = & x(t) \\ \circ\!\!-\!\!\bullet & & \circ\!\!-\!\!\bullet & & \circ\!\!-\!\!\bullet \\ \mathrm{si}\left(\frac{\omega}{2}\right) & \cdot & \mathrm{si}\left(\frac{\omega}{2}\right) & = & X(j\omega)\,. \end{array}$$




$$X(j\omega) = \mathrm{si}^2\left(\frac{\omega}{2}\right).$$



Aliasing



Spatial domain
Frequency domain
h(x)
K
x
h(x) = K
H(u) = Kδ(u)
H(u)
u
Constant function
Dirac delta function
h(x) = Kδ(x)
H(u) = K
Dirac delta function
Constant function
Δx
h(x) = Σ_{k=-∞}^{∞} δ(x − kΔx)
H(u) = 1/Δx Σ_{k=-∞}^{∞} δ(u − k/Δx)
1/Δx
Shah function
Shah function

## Constant & Dirac-delta function $\delta()$

- Duality

$$f(x) = K$$

$$F(\omega) = K\delta(\omega)$$

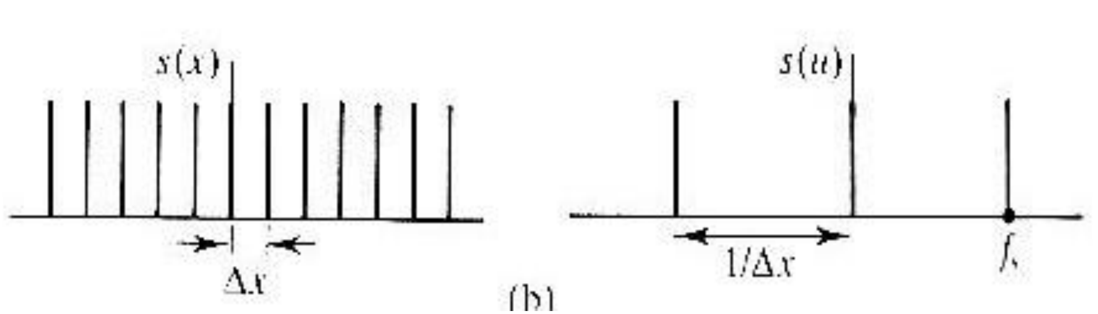
- And vice versa

## Shah function

- Duality: The dual of a Shah function is again a Shah function
    - Inverse wave length, amplitude scales with inverse wave length

$$f(x) = \sum_{k=-\infty}^{\infty} \delta(x - k\Delta x)$$

$$F(\omega) = \frac{1}{\Delta x} \sum_{k=-\infty}^{\infty} \delta\left(\omega - k\frac{1}{\Delta x}\right)$$

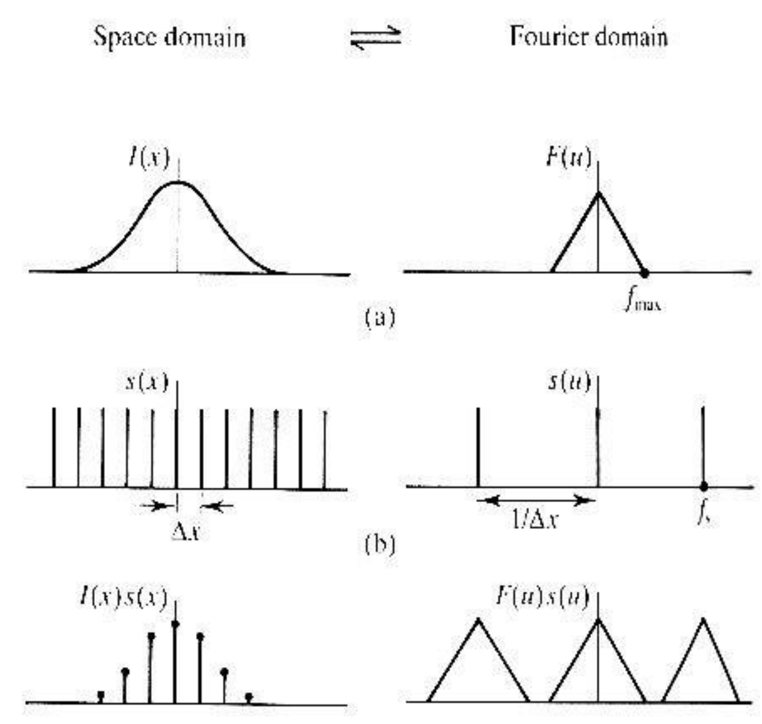


# Continuous function

- Band-limited Fourier transform

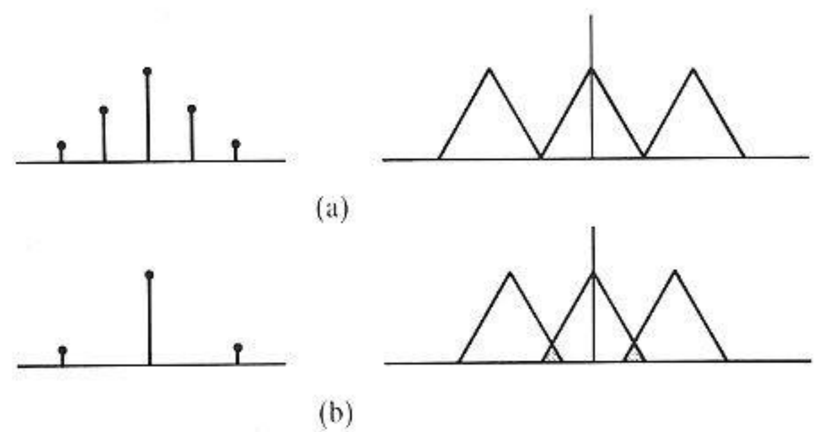
# Sampled at discrete points

- Multiplication with Shah function in spatial domain corresponds to convolution in Fourier domain



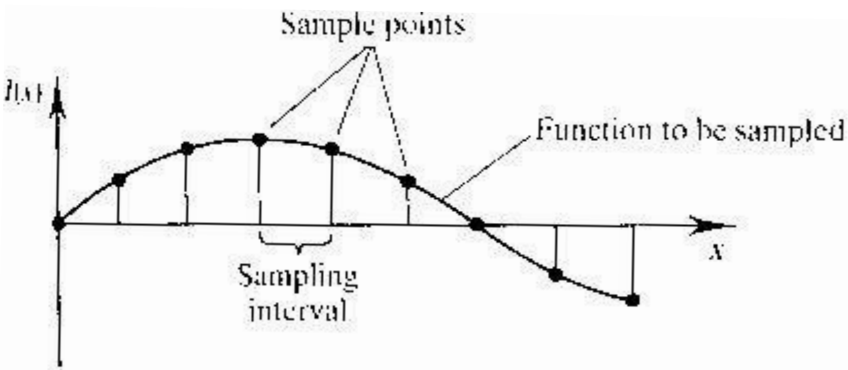

# Frequency bands overlap?

- No: good
- Yes: bad, **aliasing**

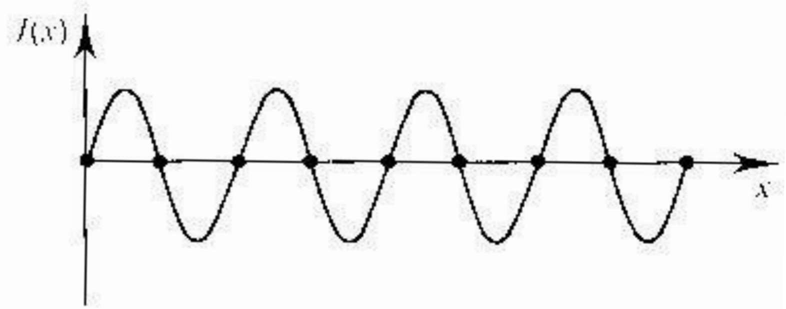


# Definition:

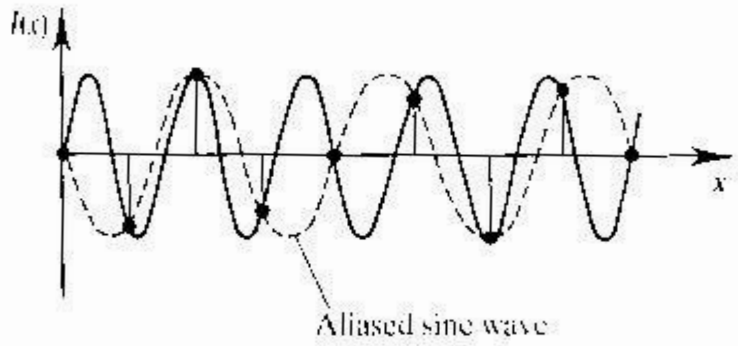
- **Nyquist Frequency** is the highest (spatial) frequency that can be represented
- Determined by image resolution (pixel size):



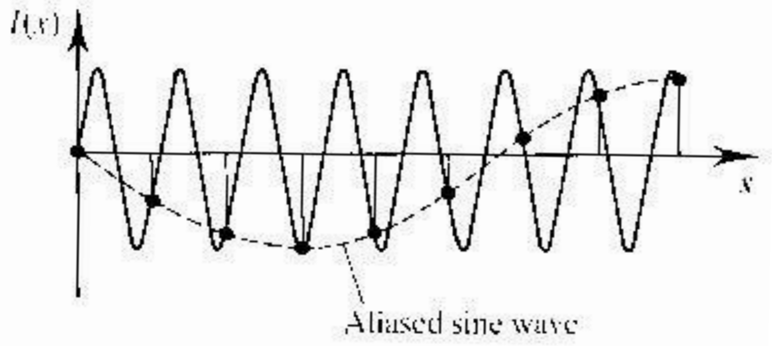

Spatial frequency < Nyquist




Spatial frequency = Nyquist
2 samples / period




Spatial frequency > Nyquist




Spatial frequency >> Nyquist

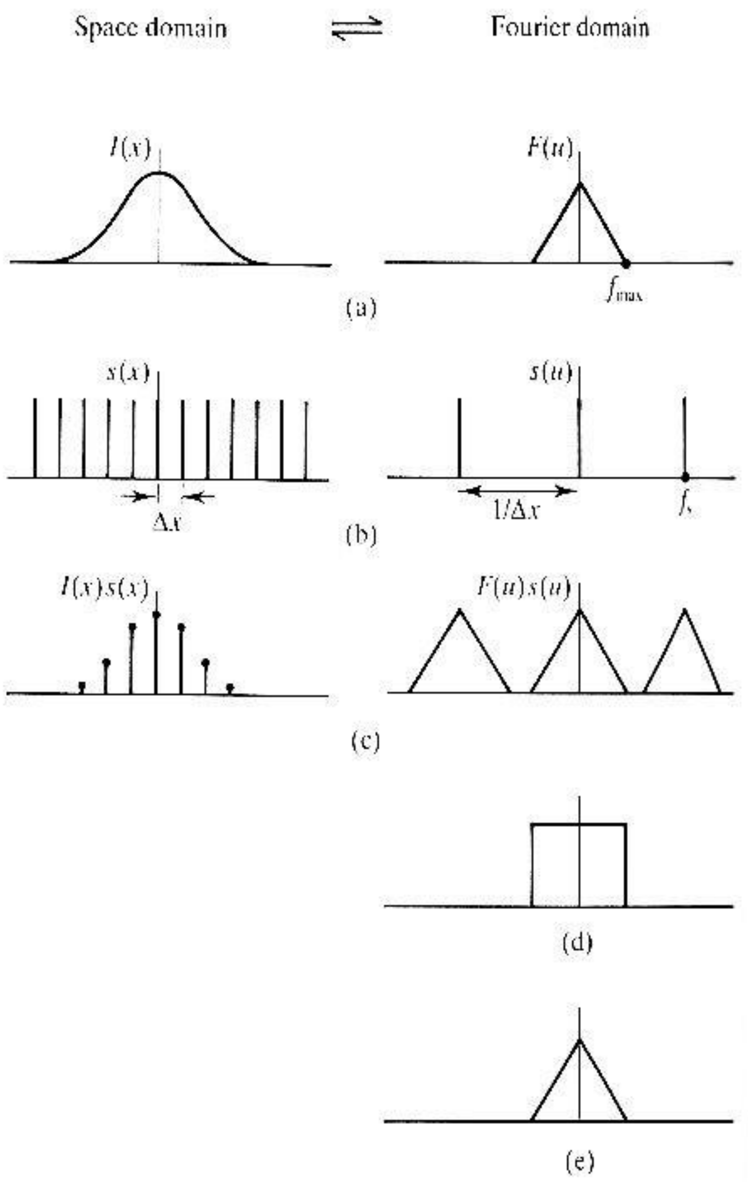
# Only original frequency band desired

## Filtering

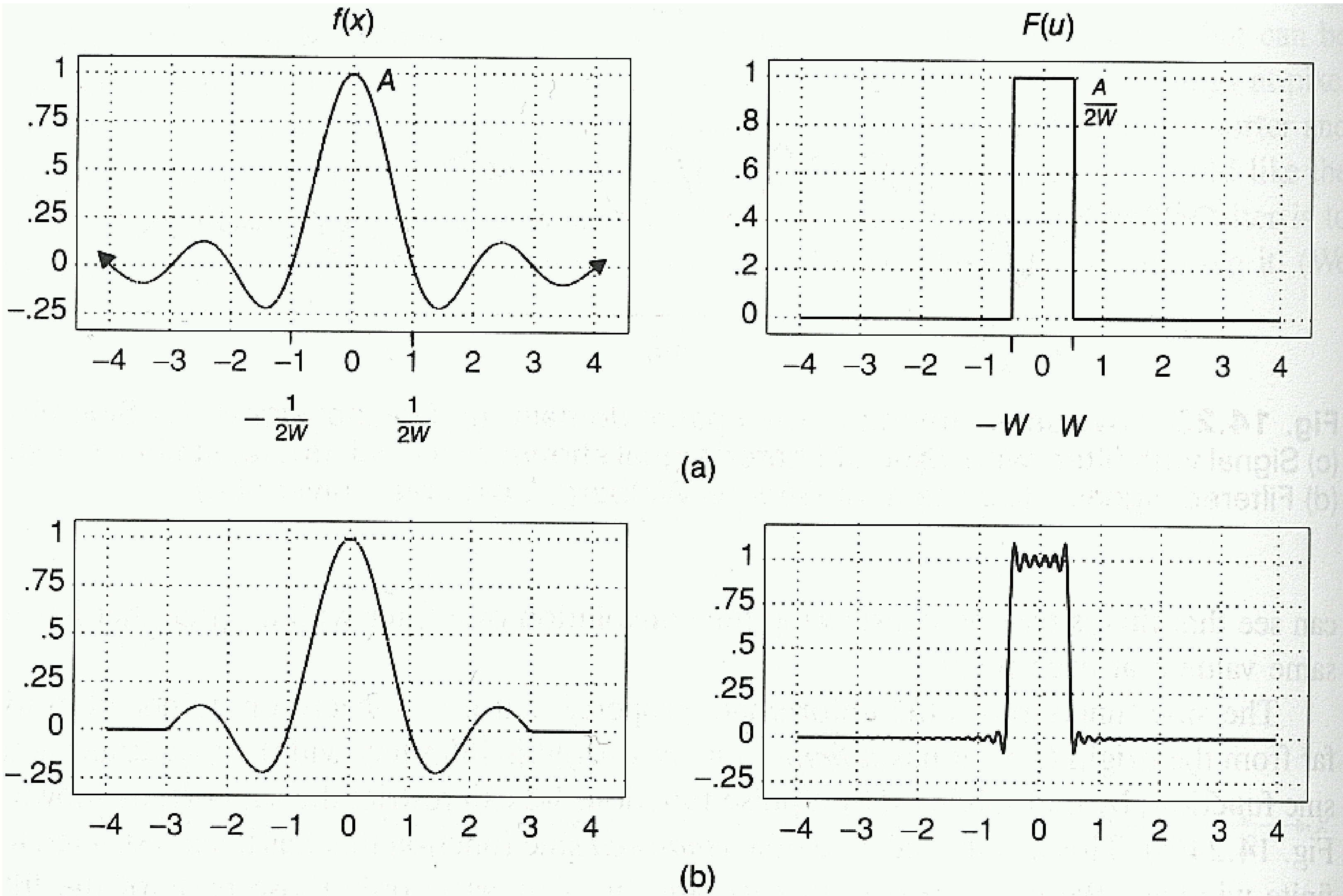
- In Fourier domain: multiplication with windowing function around origin
- In spatial domain: convolution with Fourier transform of windowing function

## Optimal filtering function

- Box function in Fourier domain
- Corresponds to $sinc()$ in space domain
  - Unlimited region of support
- Spatial domain only allows approximations (limited support)



## Cutting off the support is *not* a good solution



Original function and its band-limited frequency spectrum

Signal sampling:

Multiplication / convolution with Shah

Shah dense enough (sampling≥2*bandlimit)

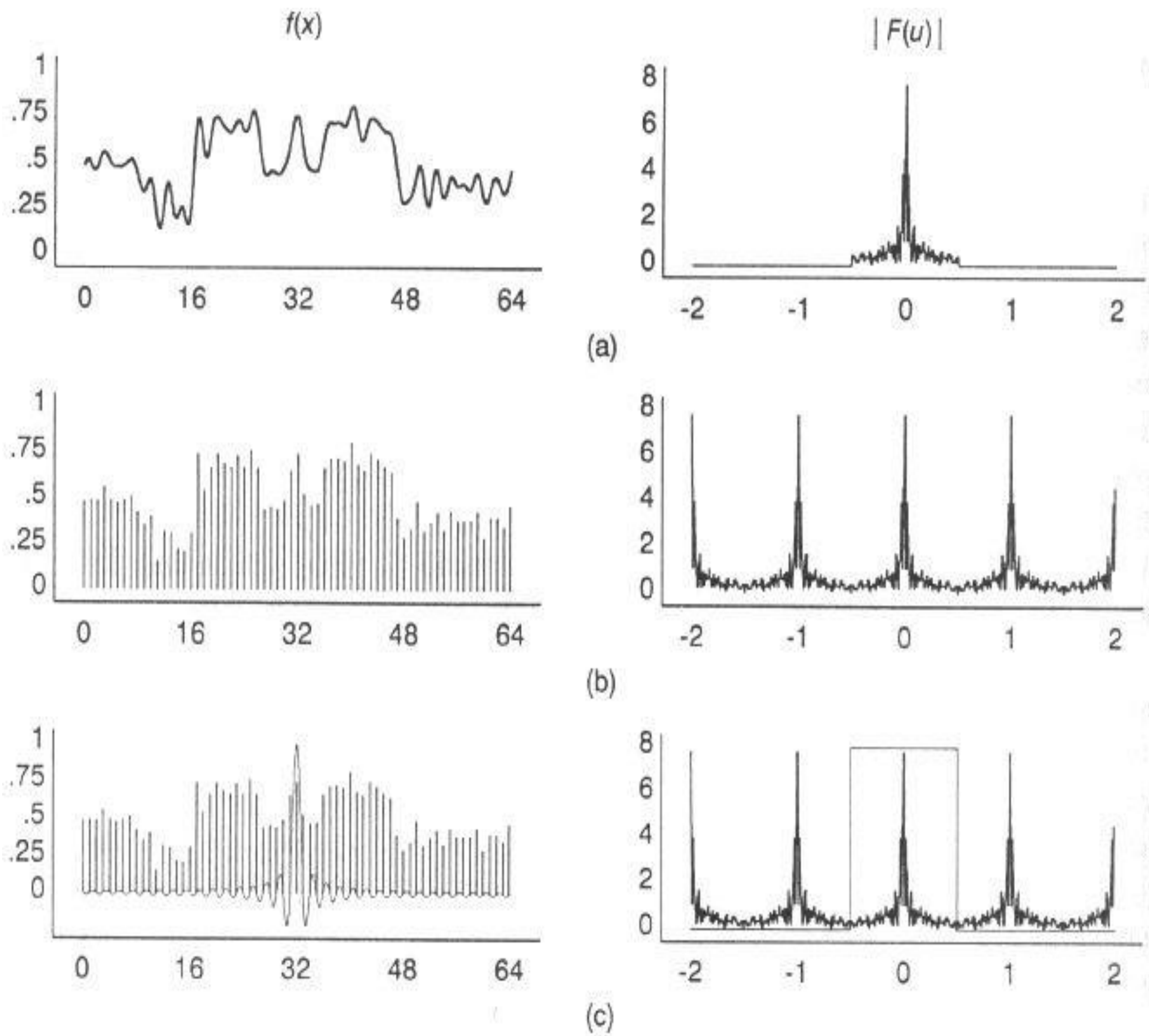
Frequency spectrum is replicated

Bands do not overlap

Correct filtering

Fourier: Box (mult.)
Space: $sinc$ (conv.)

Only one copy



Reconstruction with ideal *sinc*

Identical signal

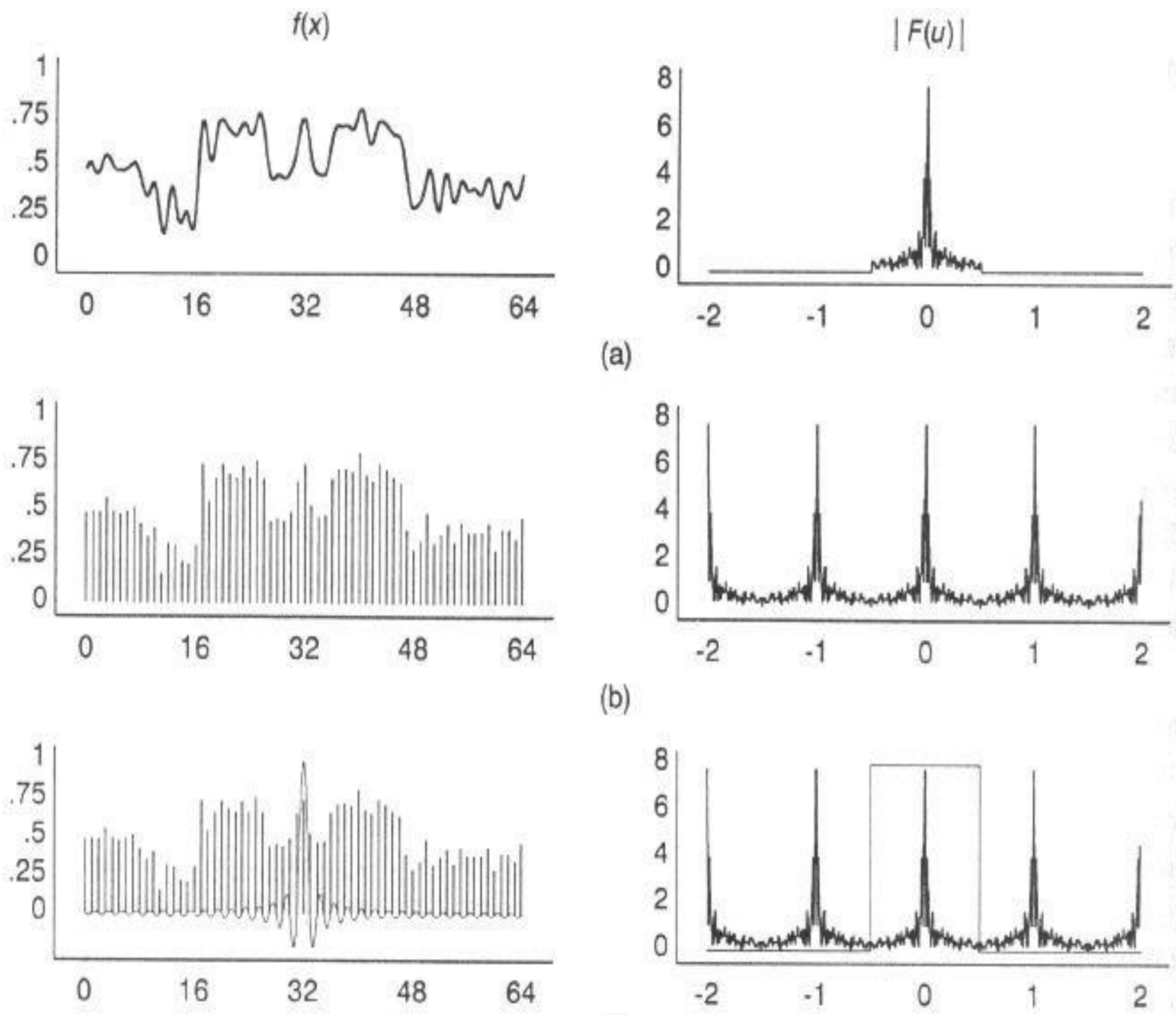
Approximate filtering

Space: triangle (conv.)
Fourier: $\mathrm{sinc}^2$ (mult.)

High frequencies are not ignored

⇒Aliasing

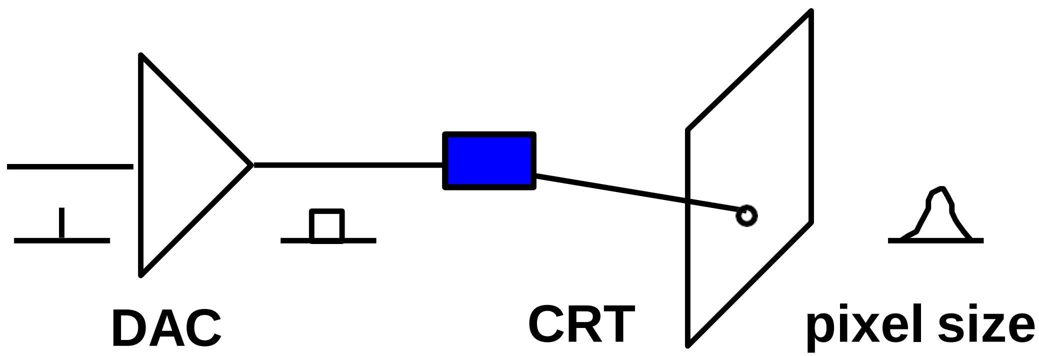
Reconstruction with triangle function (= piecewise linear interpolation)



(a)
(b)
(c)

## Reconstruction

- Physical output devices generate a continuous signal, even for discrete input, *e.g.*, on a computer monitor




## Example

- DAC (Digital-to-Analog Converter): Sample and hold
  - Capacities and inductivities
- CRT: Phosphor and light spot
  - Afterglow on screen

Original function

Sampling below Nyquist:

Shah spaced to far (sampling<2*bandlimit)

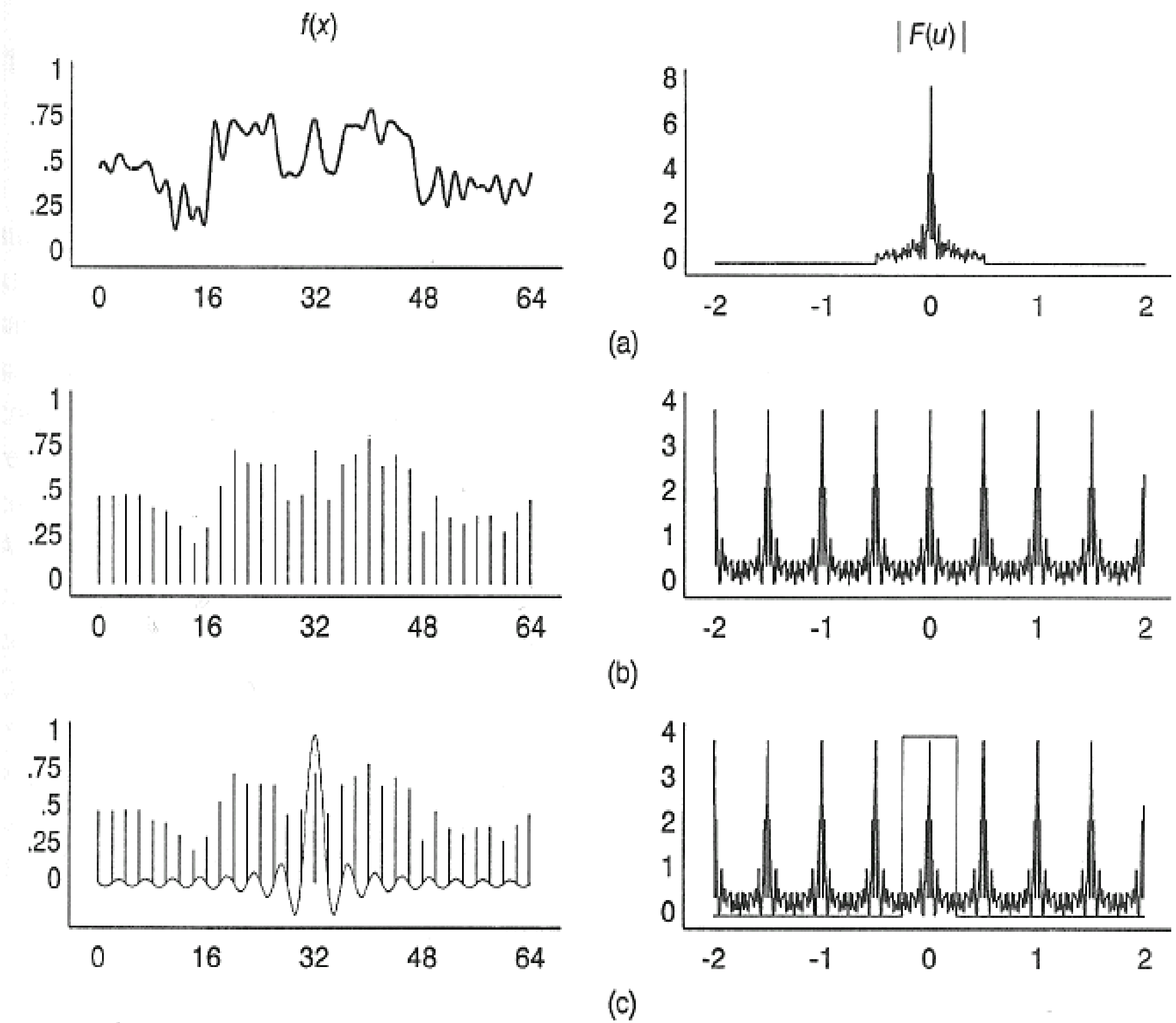
Frequency bands overlap

Correct filtering

Space: $sinc$ (conv.)

Fourier: hat (mult.)

Band overlap in frequency domain cannot be corrected - aliasing



# Sampling with Low Frequency

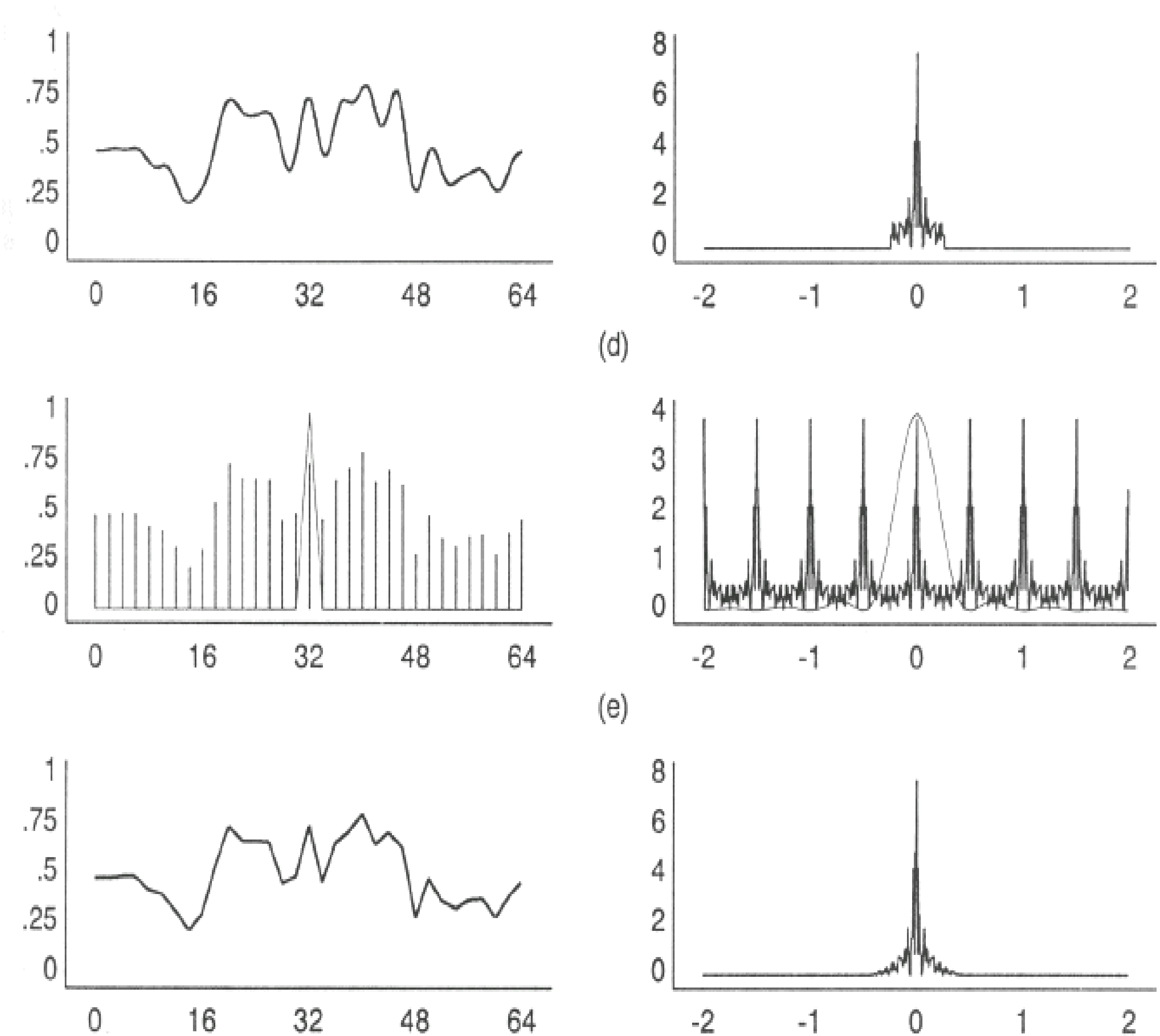
Reconstruction with ideal $sinc$

Reconstruction fails (frequency components wrong due to aliasing!)

Fitlering with $sinc^2$ function

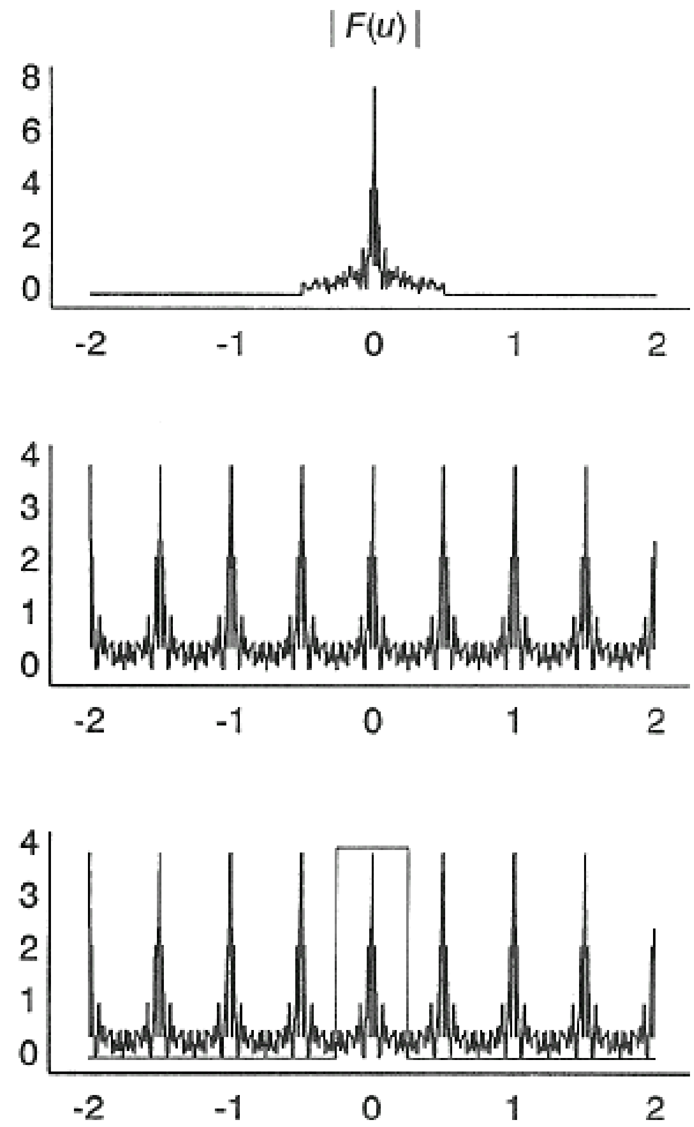
Reconstruction with triangle function (= piecewise linear interpolation)

Even worse reconstruction



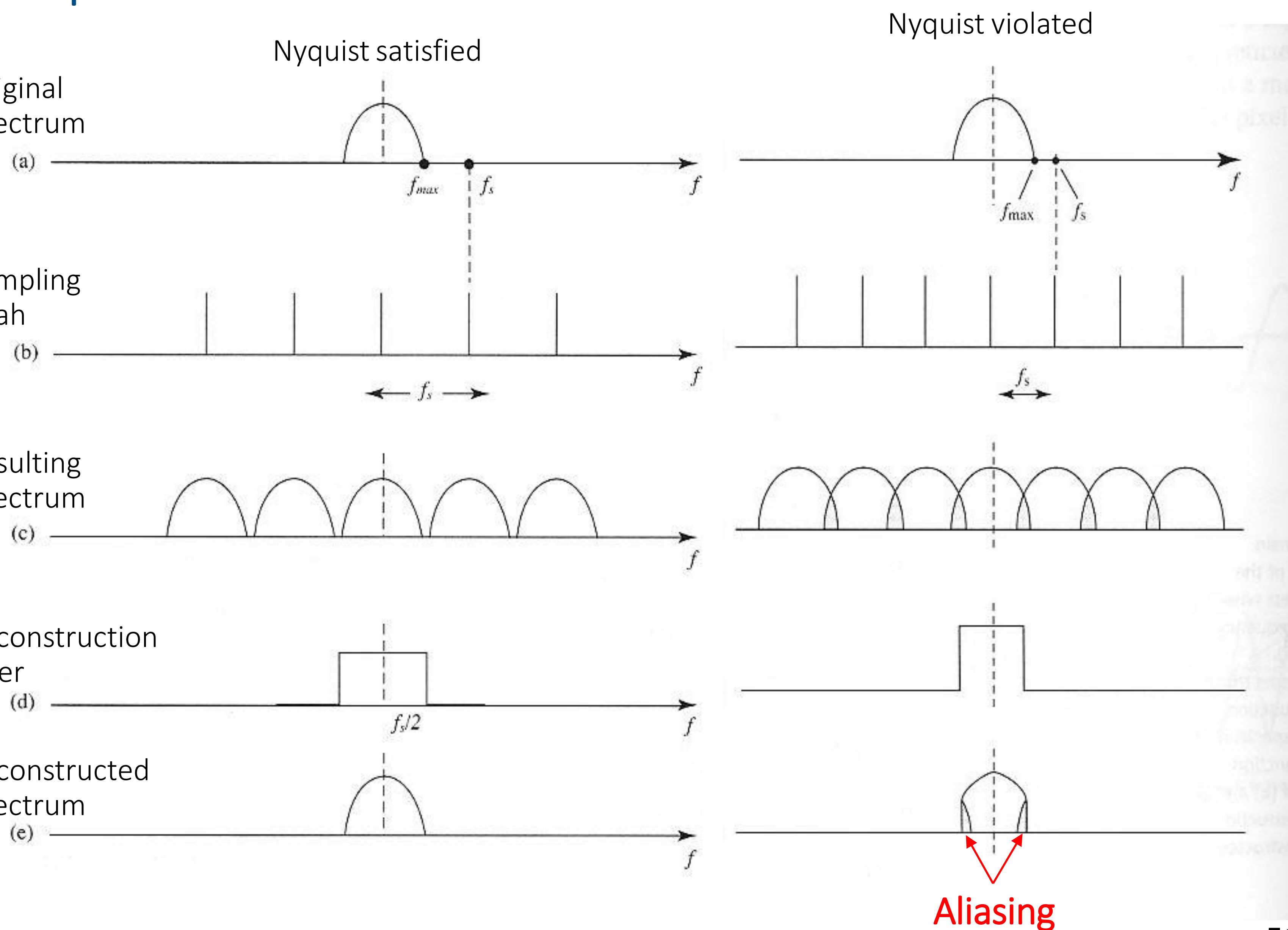
# Overlap between replicated copies in frequency spectrum

- High frequency components from the replicated copies are treated like low frequencies during the reconstruction process



## In Fourier space

- Original spectrum
- Sampling Shah
- Resulting spectrum
- Reconstruction filter
- Reconstructed spectrum



## It all comes from sampling at discrete points

- Multiplied with Shah function, no smoothly weighted filters
- Shah function: repeats frequency spectrum

## Or, from using non band limited primitives

- Hard edges ⇒ infinitely high frequencies

## In reality, integration over finite region necessary

- *E.g.*, finite CCD pixel size

## Computer: Analytic integration often not possible

- No analytic description of radiance or visible geometry available

## Only way: numerical integration

- Estimate integral by taking multiple point samples, average
    - Leads to aliasing
- Computationally expensive
- Approximate

# Prefiltering

Filter (*i.e.*, blur) the original function so that no high frequencies remain that can't be captured accurately at the sampling rate being used. While this technique changes the character of the function being sampled by removing information from it, blurring is generally less objectionable than aliasing.

# Nonuniform Sampling

Although the image functions that we will be sampling are known to have infinite-frequency components and thus can't be perfectly reconstructed from point samples, it is possible to reduce the visual impact of aliasing by varying the spacing between samples in a nonuniform way.

# Adaptive Sampling

If we can identify the regions of the signal with frequencies higher than the Nyquist limit, we can take additional samples in those regions without needing to incur the computational expense of increasing the sampling frequency everywhere. It can be difficult to get this approach to work well in practice, because finding all of the places where supersampling is needed is difficult. Most techniques for doing so are based on examining adjacent sample values and finding places where there is a significant change in value between the two; the assumption is that the signal has high frequencies in that region.

## Filtering before sampling

- Band-limiting signal
- Analog / analytic or
- Reduce Nyquist frequency for chosen sampling-rate

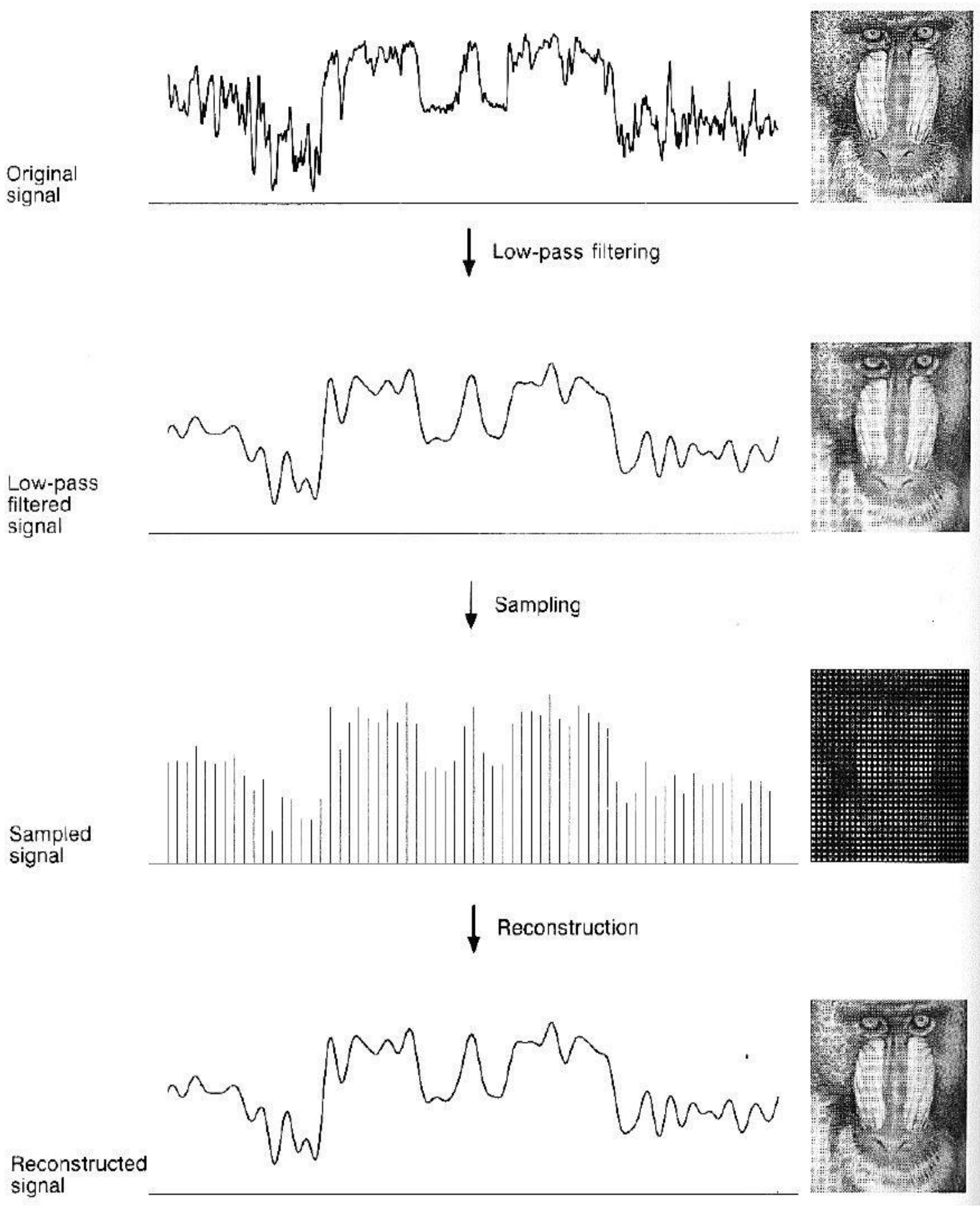
## Ideal reconstruction

- Convolution with $sinc$

## Practical reconstruction

- Convolution with
  - Box filter, Bartlett (Tent)

## ⇒ Reconstruction error

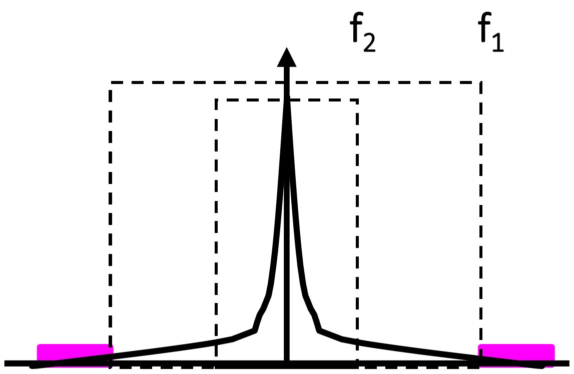


## Assumption

- Energy in high frequencies decreases quickly
- Reduced aliasing by intermediate sampling with higher frequencies

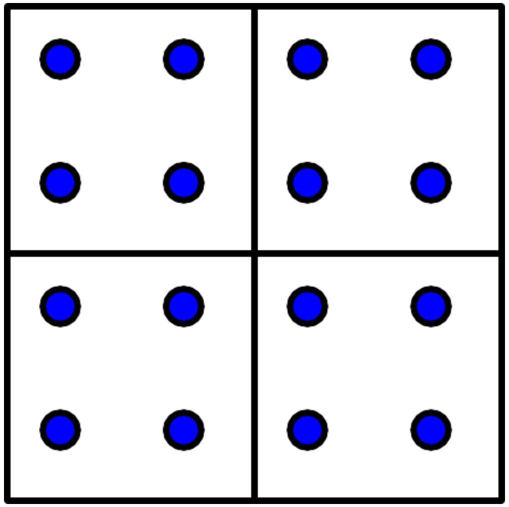
## Algorithm

- Super-sampling
  - Sample continuous signal with boundary frequency **$f_1$**
  - Aliasing with energy beyond **$f_1$** (assumed to be small)
- Reconstruction of signal
  - Filtering with $g_1(x)$: *e.g.* convolution with $sinc(\mathrm{f}_1)$
- Analytic low-pass filtering of signal
  - Filtering with filter $g_2(x)$ with **$f_2$** << **$f_1$**
  - Signal is now band limited w.r.t. **$f_2$**
- Re-sampling with a sampling frequency that is compatible with **$f_2$**
  - No additional aliasing
- Filters $g_1(x)$ and $g_2(x)$ can be combined
- Hardware support (OpenGL multisampling extension)



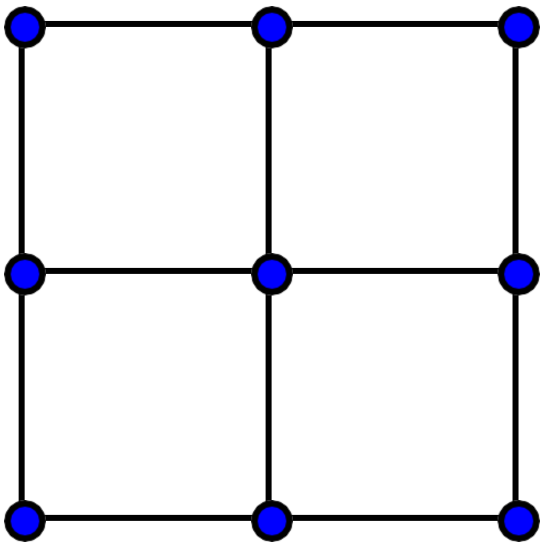
# Regular super-sampling

- Averaging of $N$ samples per pixel on a grid
- $N$ :
  - 4 quite good
  - 16 almost always sufficient
- Samples
  - Rays, z-buffer, reflection, motion, ...
- Averaging
  - Box filter
  - Others: Pyramid (Bartlett), B-spline, Hexagonal, ...
- Regular super-sampling
  - Nyquist frequency for aliasing only shifted
    - $\Rightarrow$ Irregular sampling patterns



## Popular mistake

- Sampling at the corners of every pixel
- Pixel color by averaging
- Free super-sampling ???

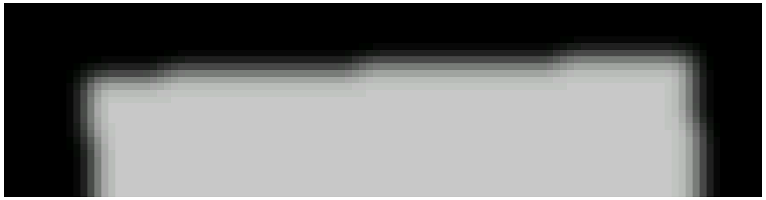


## Problem

- Wrong reconstruction filter!!!
- Same sampling frequency, but post-filtering with a hat function
- Blurring: Loss of information

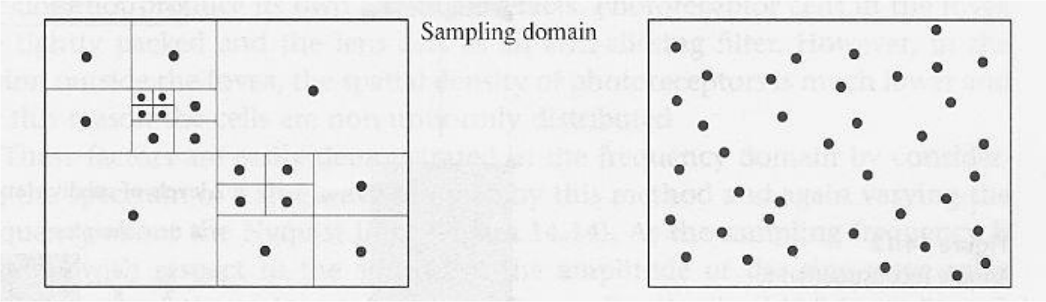
## Post-Reconstruction Blur

1x1 Sampling, 3x3 Blur



1x1 Sampling, 7x7 Blur

⇒ „Super-sampling“ does not come for free

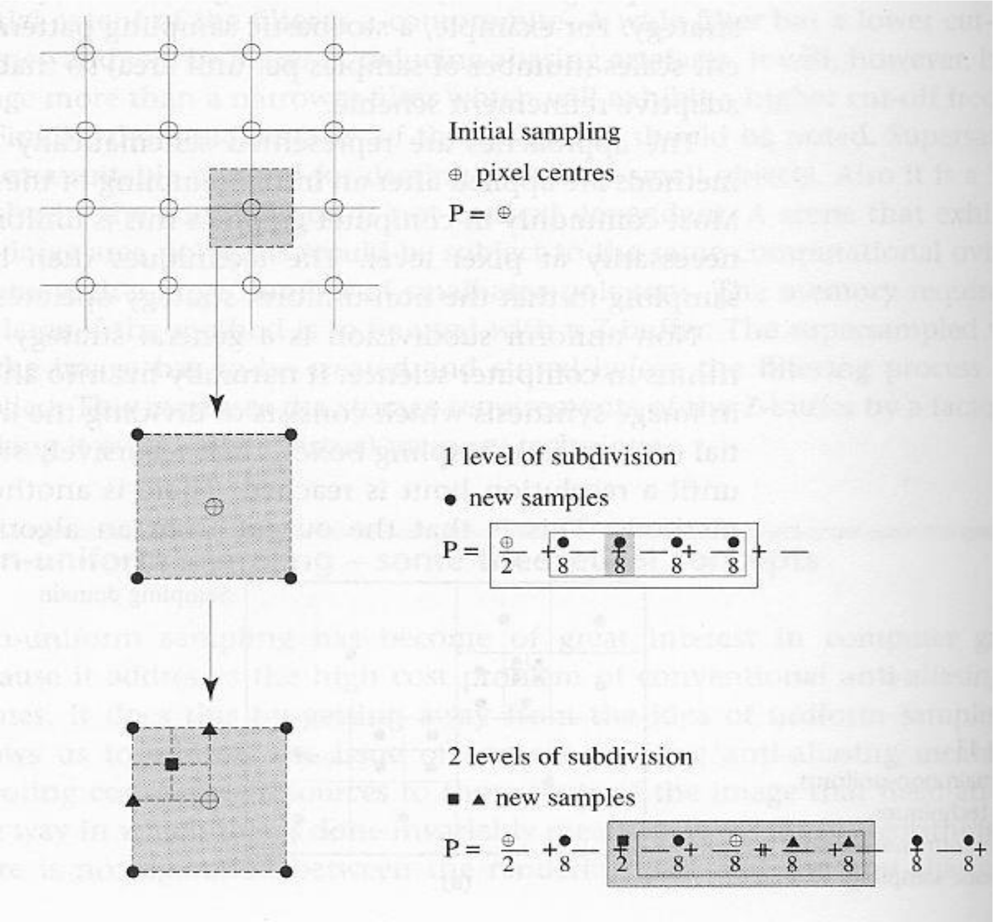



## Adaptive super-sampling

- Idea: locally adapt sampling density
    - Slowly varying signal:

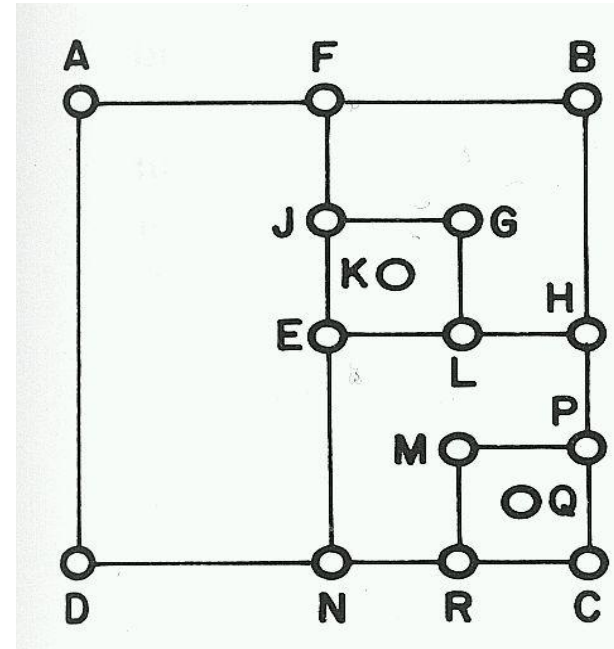
      low sampling rate
    - Strong changes:

      high sampling rate
- Decide sampling density locally
- Decision criterion needed
    - Differences of pixel values
    - Contrast (relative difference)
        - $|A-B| / |A|+|B|$

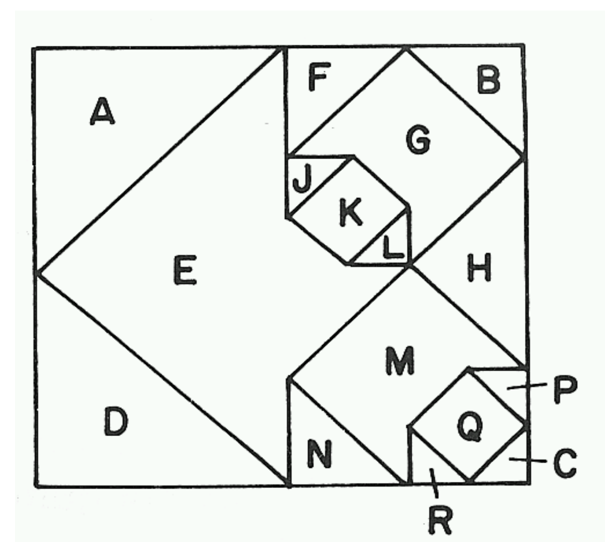


# Algorithm

- Sampling at corners and mid points
- Recursive subdivision of each quadrant
- Decision criterion
  - Differences, contrast, object-IDs, ray trees, ...
- Filtering with weighted averaging
  - ¼ from each quadrant
  - Quadrant: ½ (midpoint + corner)
    - Recursion




$$\frac{1}{4}\left(\frac{A+E}{2}+\frac{D+E}{2}+\frac{1}{4}\left[\frac{F+G}{2}+\frac{B+G}{2}+\frac{H+G}{2}+\frac{1}{4}\left\{\frac{J+K}{2}+\frac{G+K}{2}+\frac{L+K}{2}+\frac{E+K}{2}\right\}\right]\right.$$
$$\left.+\frac{1}{4}\left[\frac{E+M}{2}+\frac{H+M}{2}+\frac{N+M}{2}+\frac{1}{4}\left\{\frac{M+Q}{2}+\frac{P+Q}{2}+\frac{C+Q}{2}+\frac{R+Q}{2}\right\}\right]\right)$$




# Extension

- Jittering of sample points

## Geometry

- Edges, vertices, sharp boundaries
- Silhouettes (view dependent)
- ...

## Texture

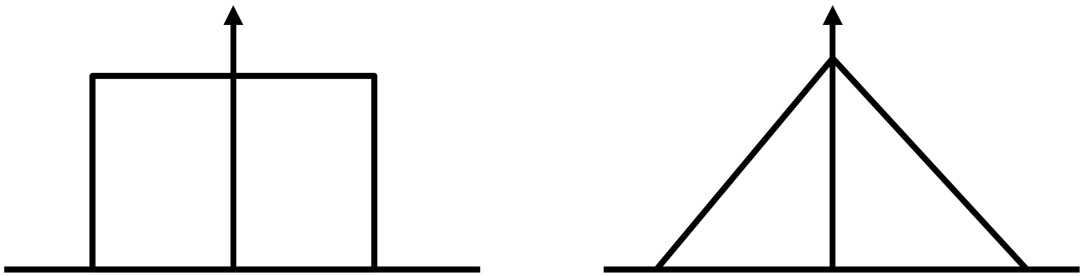
- *E.g.*, checkerboard pattern, other discontinuities, ...

## Illumination

- Shadows, lighting effects, projections, ...

## ⇒ Analytic filtering almost impossible

- Even with the most simple filters

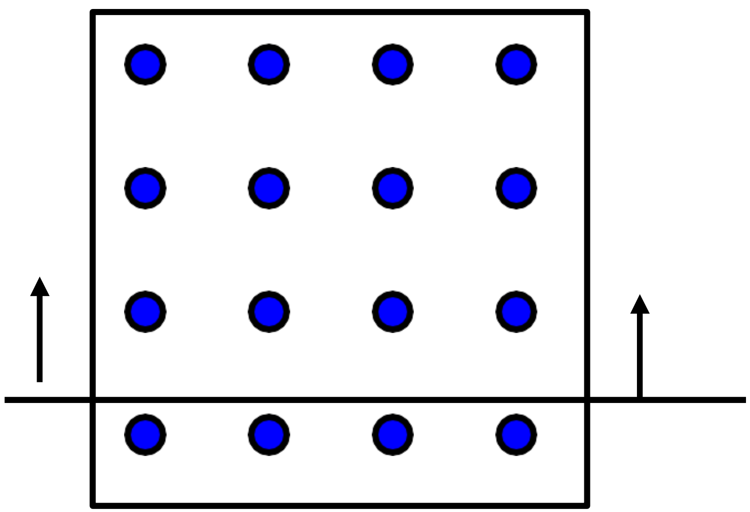


# Stratified Sampling

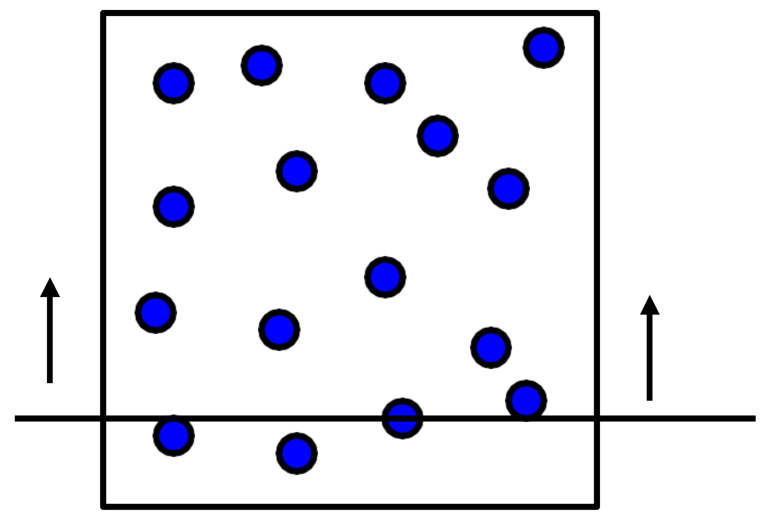
## Problems with regular super-sampling

- Expensive: 4-fold to 16-fold effort
- Non-adaptive: Same effort everywhere
- Too regular: Reduced number of levels

## Introduce irregular sampling pattern



0 → 4/16 → 8/16 → 12/16 → 16/16



Better, but noisy

## Stochastic super-sampling

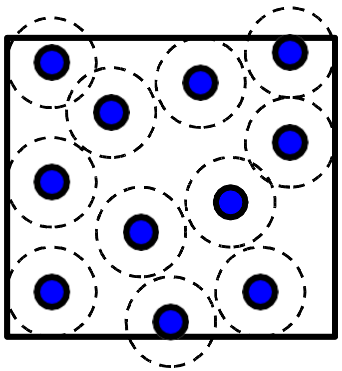
- Or analytic computation of pixel coverage and pixel mask

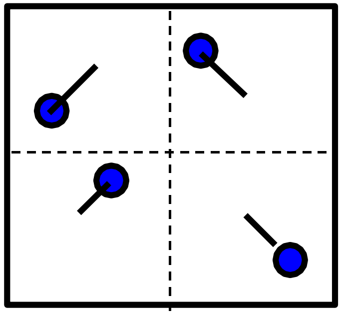
## Requirements

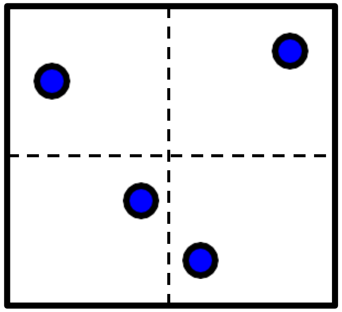
- Even distribution
- Little correlation between samples
- Incremental generation

## Generation of samples

- Poisson-disk sampling
    - Fixes a minimum distance between samples
    - Random generation of samples
        - Rejection, if too close to other samples
- Jittered sampling
    - Random perturbation from regular positions
- Stratified Sampling
    - Subdivision into areas with one random sample each
- Quasi-random numbers (Quasi-Monte Carlo)
    - *E.g.* Halton Sequence
    - Advanced feature



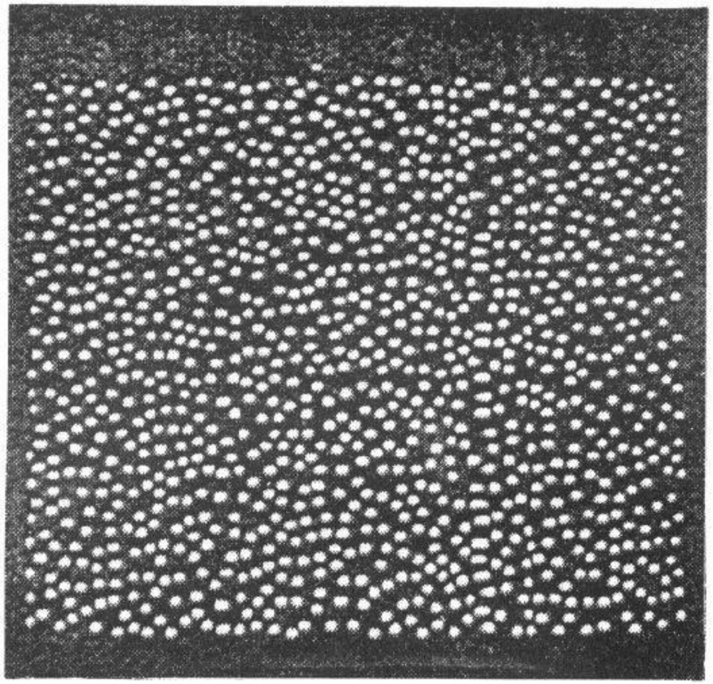




# Motivation

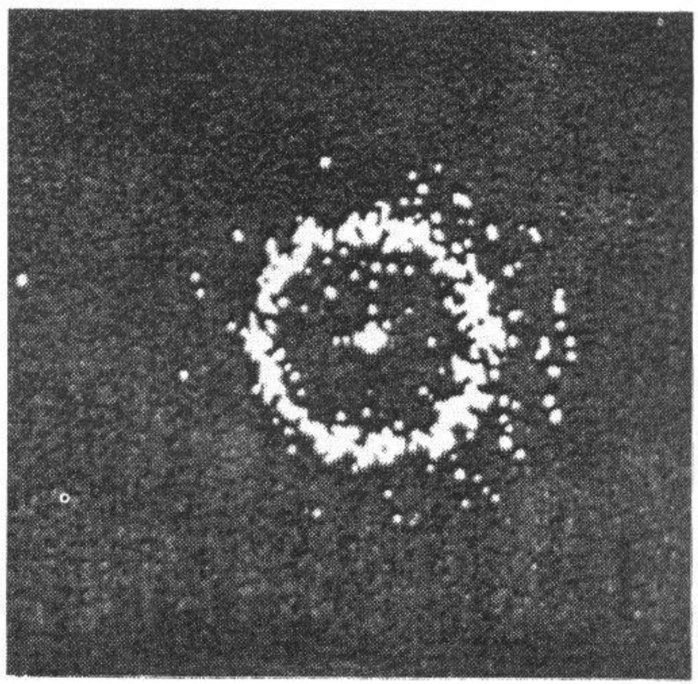
- Distribution of the optical receptors on the retina (here: ape)

(a)



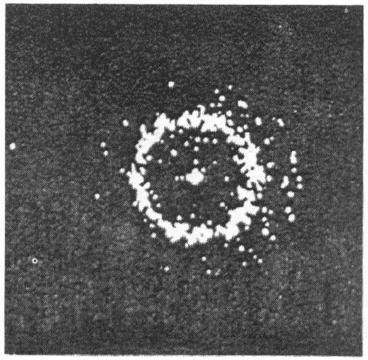
Distribution of the receptors

(b)

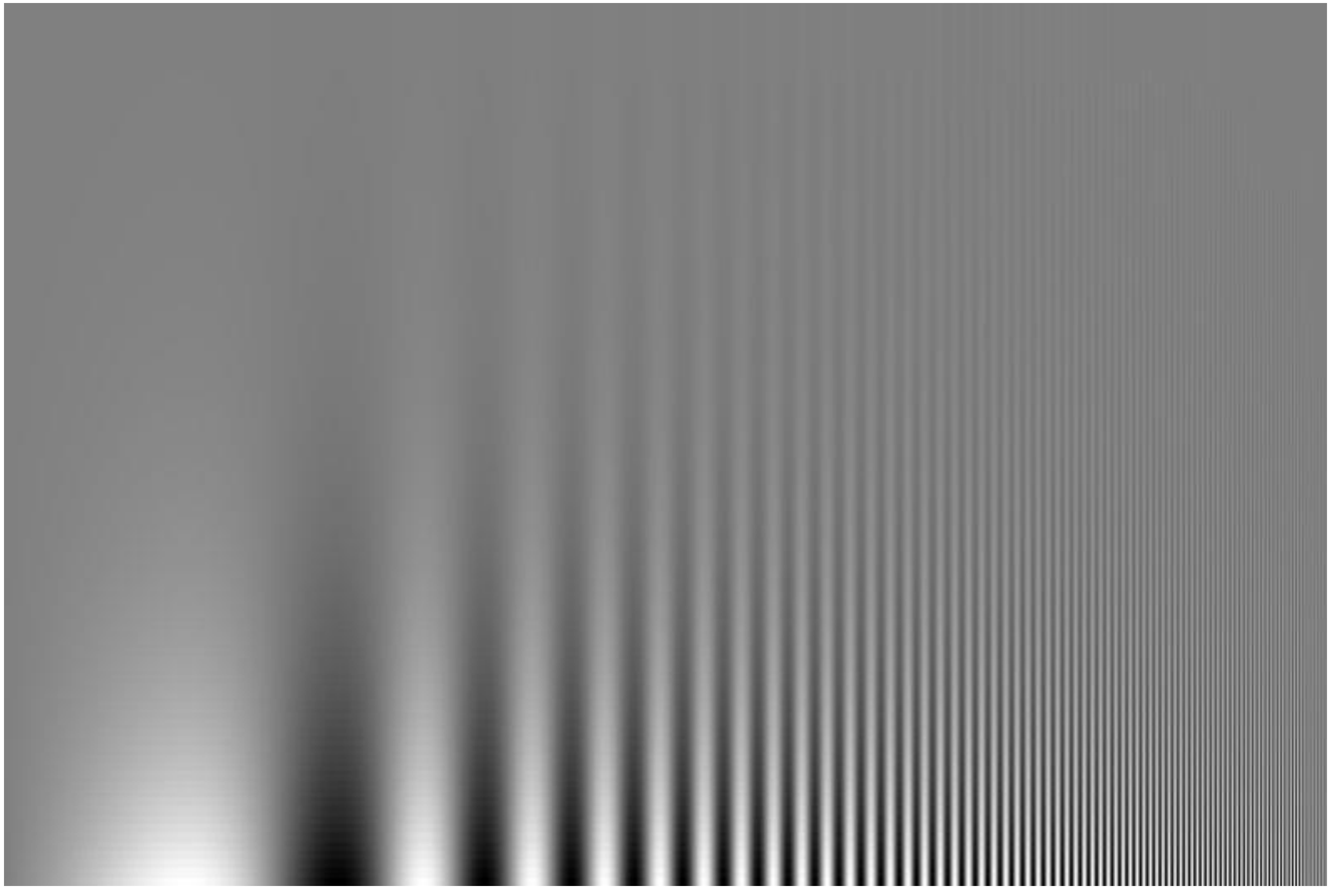


Fourier analysis

## Human perception



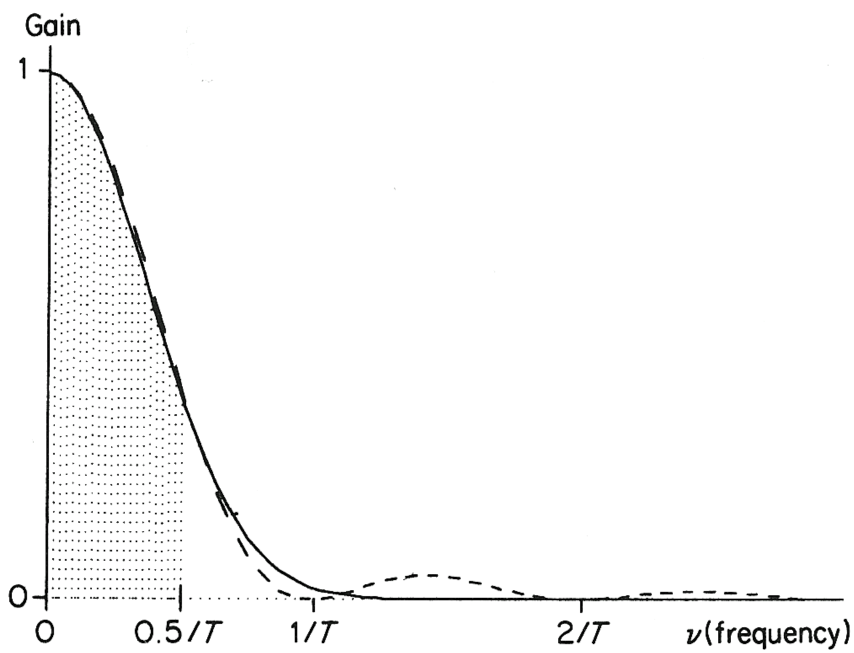
- Very sensitive to regular structures
- Insensitive against (high frequency) noise



Campbell-Robson contrast sensitivity chart

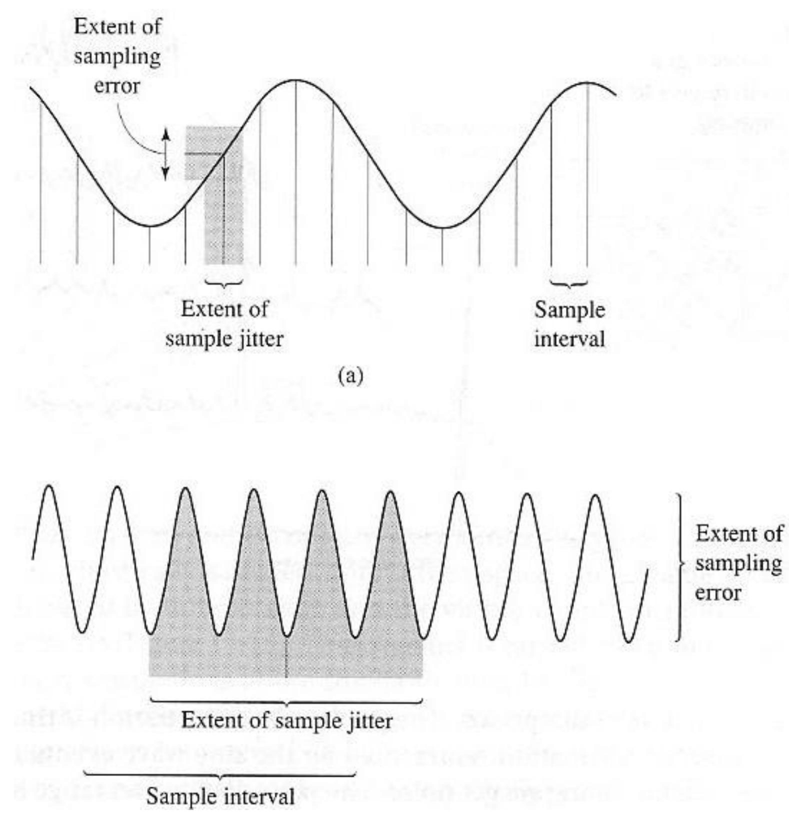
# Stochastic Sampling

- Transforms energy in high frequency bands into noise




Gauss Jittering

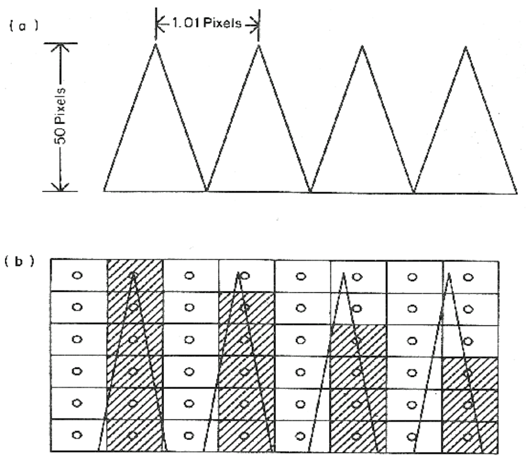
White-Noise Jittering

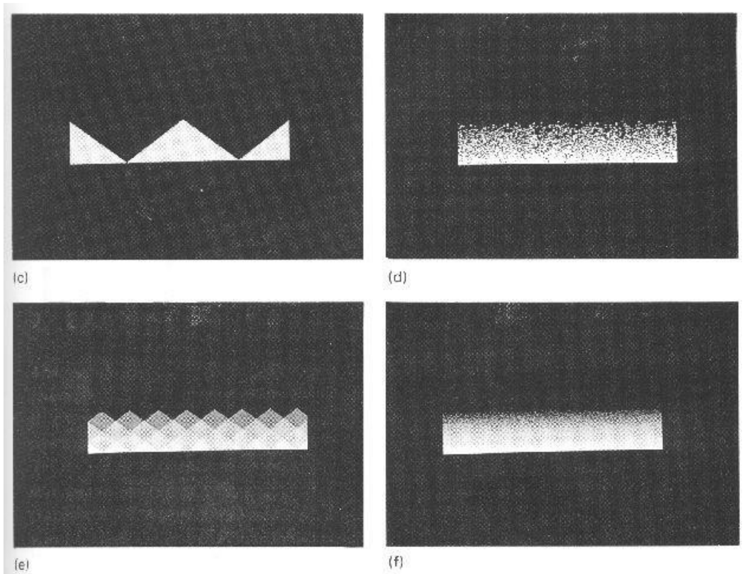


## Triangle Shah

## (Width: 1.01 pix, Heigth: 50 pix):

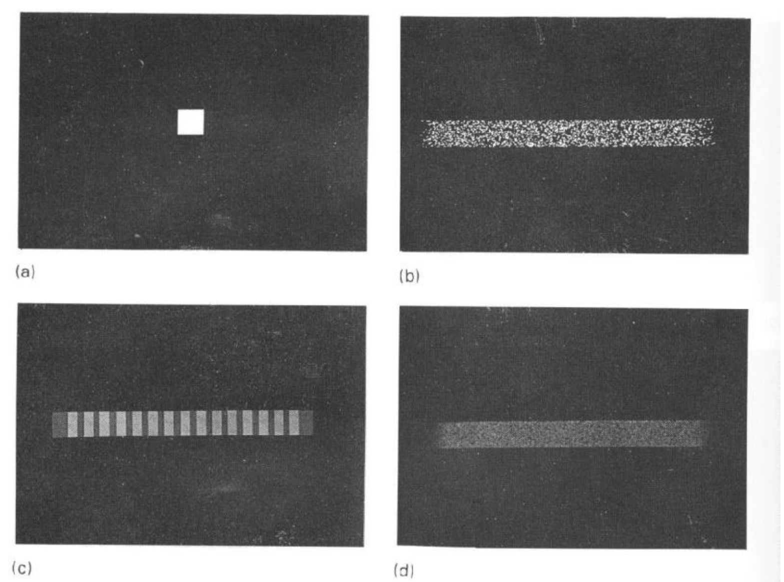
- 1 sample, no jittering
- 1 sample, jittering
- 16 samples, no jittering
- 16 samples, jittering




## Motion Blur:

- 1 sample, no jittering
- 1 sample, jittering
- 16 samples, no jittering
- 16 samples, jittering



# Comparison

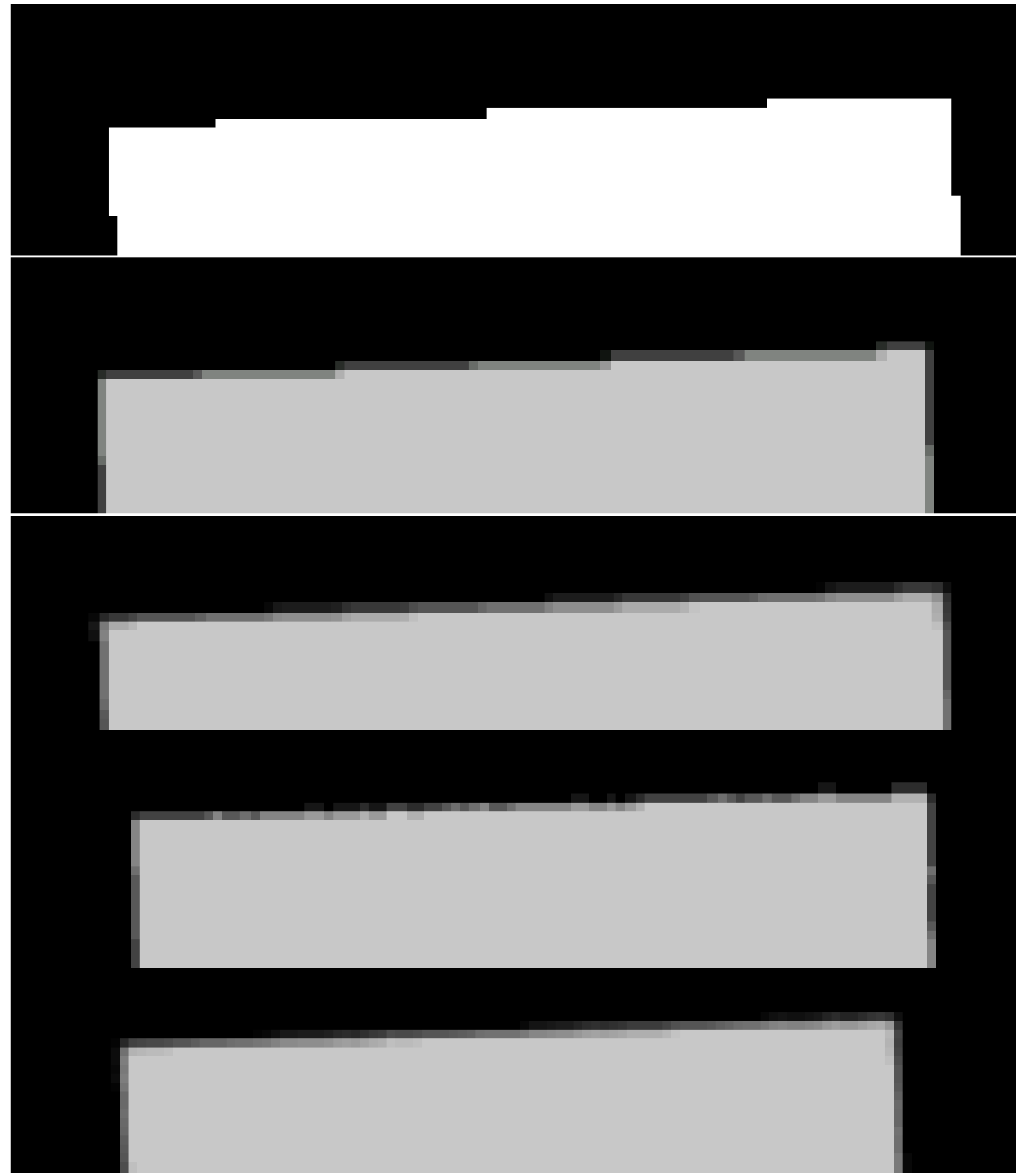
Regular, 1x1

Regular 3x3

Regular, 7x7

Jittered, 3x3

Jittered, 7x7



# Image Reconstruction

## Filter Functions for econstruction

- Box Filter
- Triangle Filter
- Gaussian Filter
- Mitchell Filter
- Windowed Sinc Filter

## Read more

- http://www.pbr-book.org/3ed-2018/Sampling_and_Reconstruction/Image_Reconstruction.html

## Fourier transformation

- Equivalent representation of transformed signal
- Spectral analysis: shows signal’s frequency components

## Convolution

- Filtering

## Sampling

- Multiplication with Shah function
- Only at discrete points: no integration over signal
- Frequency spectrum replicated
- Replication distance = sampling rate

## Aliasing

- Replicated spectra overlap
- Cannot be separated by filtering anymore
- Erroneous frequency amplitudes – wrong function!

## Spatial aliasing solutions:

- Increasing the sampling rate
  - Ok, but infinite frequencies at sharp edges
- Post-filtering (after reconstruction)
  - Does not work - only leads to blurred stair cases
- Pre-filtering (Blurring) of sharp geometry features
  - Slowly make geometry transparent at the edges
  - Correct solution in principal
  - Analytic low-pass filtering hard to implement
  - Super-sampling

## Temporal aliasing solutions:

- Increasing the frame rate
  - OK
- Pre-filtering (Motion Blur)
  - Yes, possible for simple geometry (*e.g.*, Cartoons)
  - Problems with texture, *etc*.
- Post-filtering (Averaging several frames)
  - Does not work – only multiple detail