Aliasing And Anti-Aliasing
Sampling and Reconstruction

An Introduction

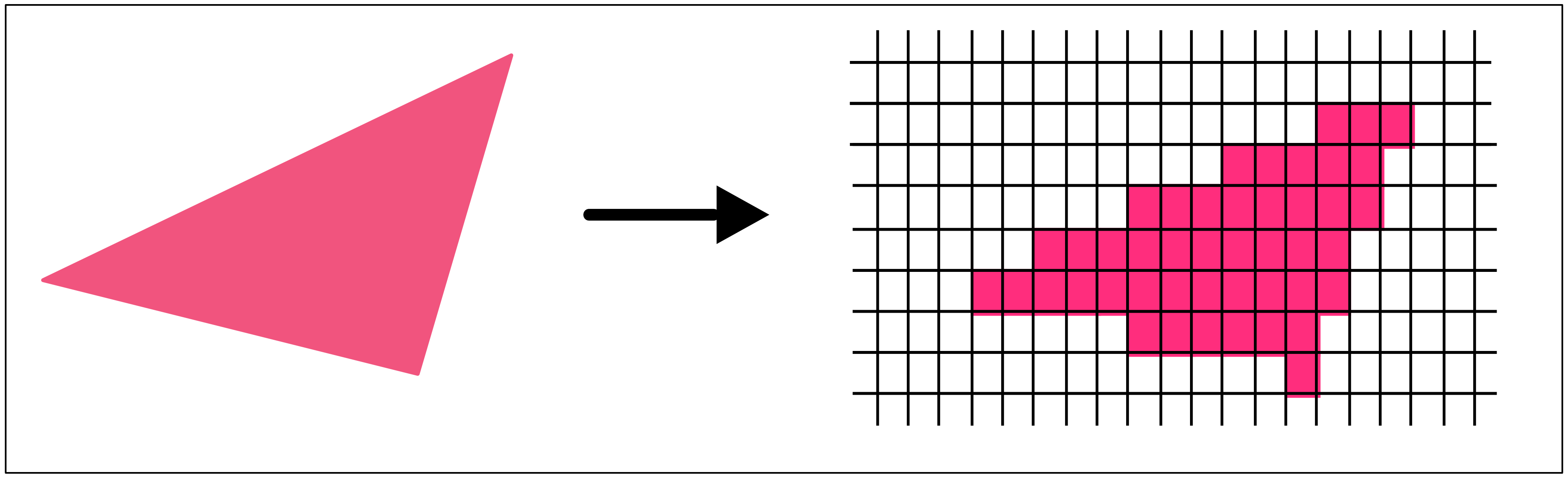
Computer
Graphics
Charles
University

Computer Graphics Charles University

Overview

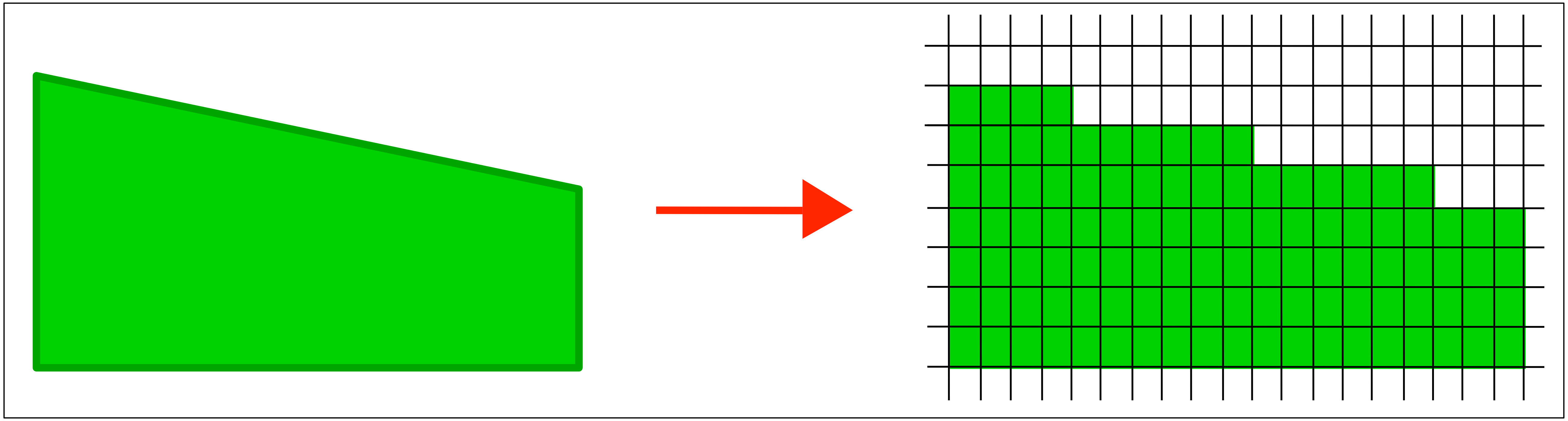
- Intro - Aliasing
  - Problem definition, Examples
- Ad-hoc Solutions
- Sampling theory
  - Fourier transform
  - Convolution
- Reconstruction
  - Sampling theorem
  - Reconstruction in theory and practice

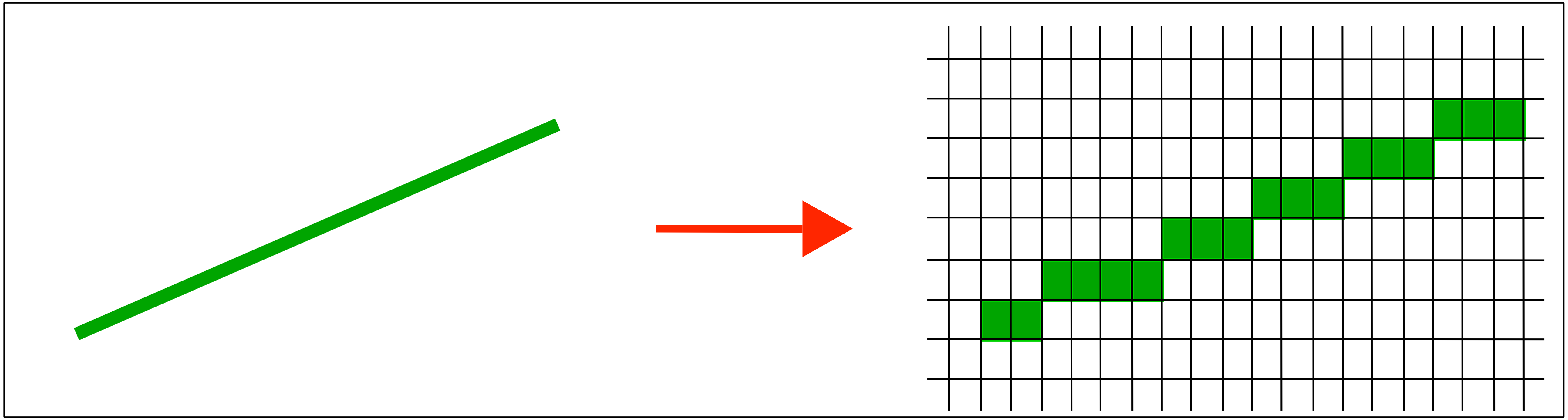
Aliasing - a Common Problem



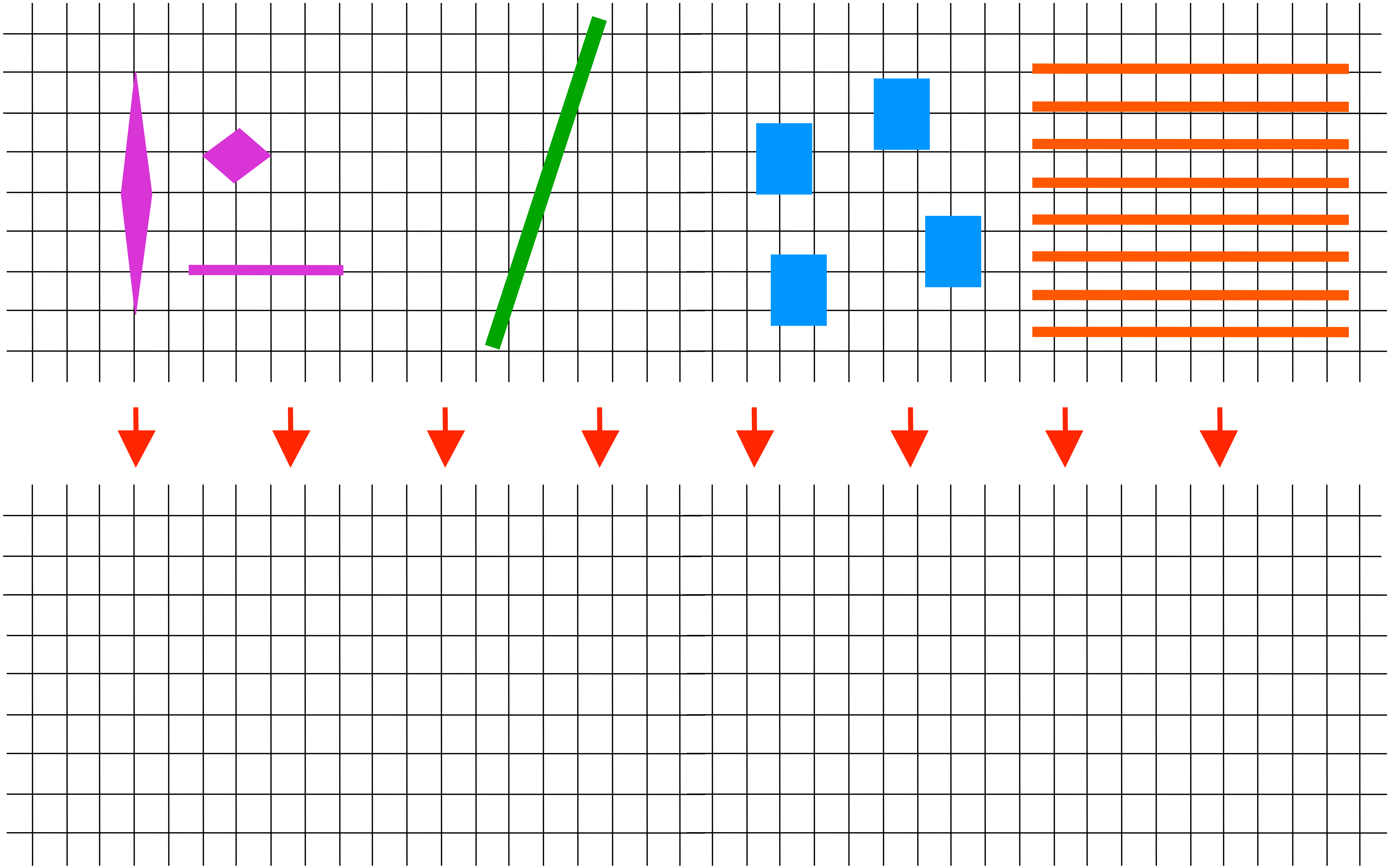
- Errors incurred during analog-digital conversions:
  - Geometric (pixel artefacts, „jaggies“)
  - Colour quality
  - Errors in animation sequences

Aliasing: „Jaggies“

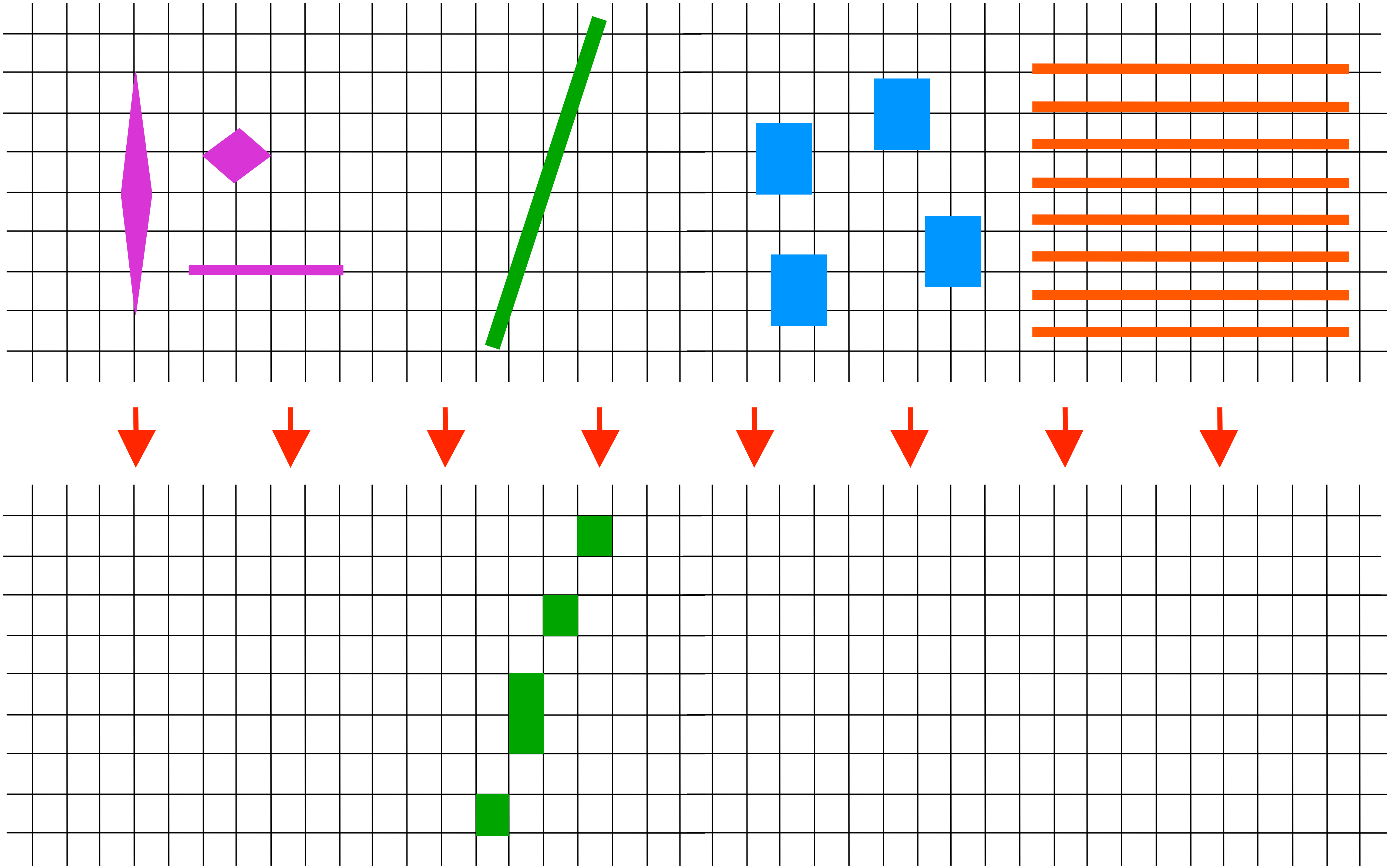




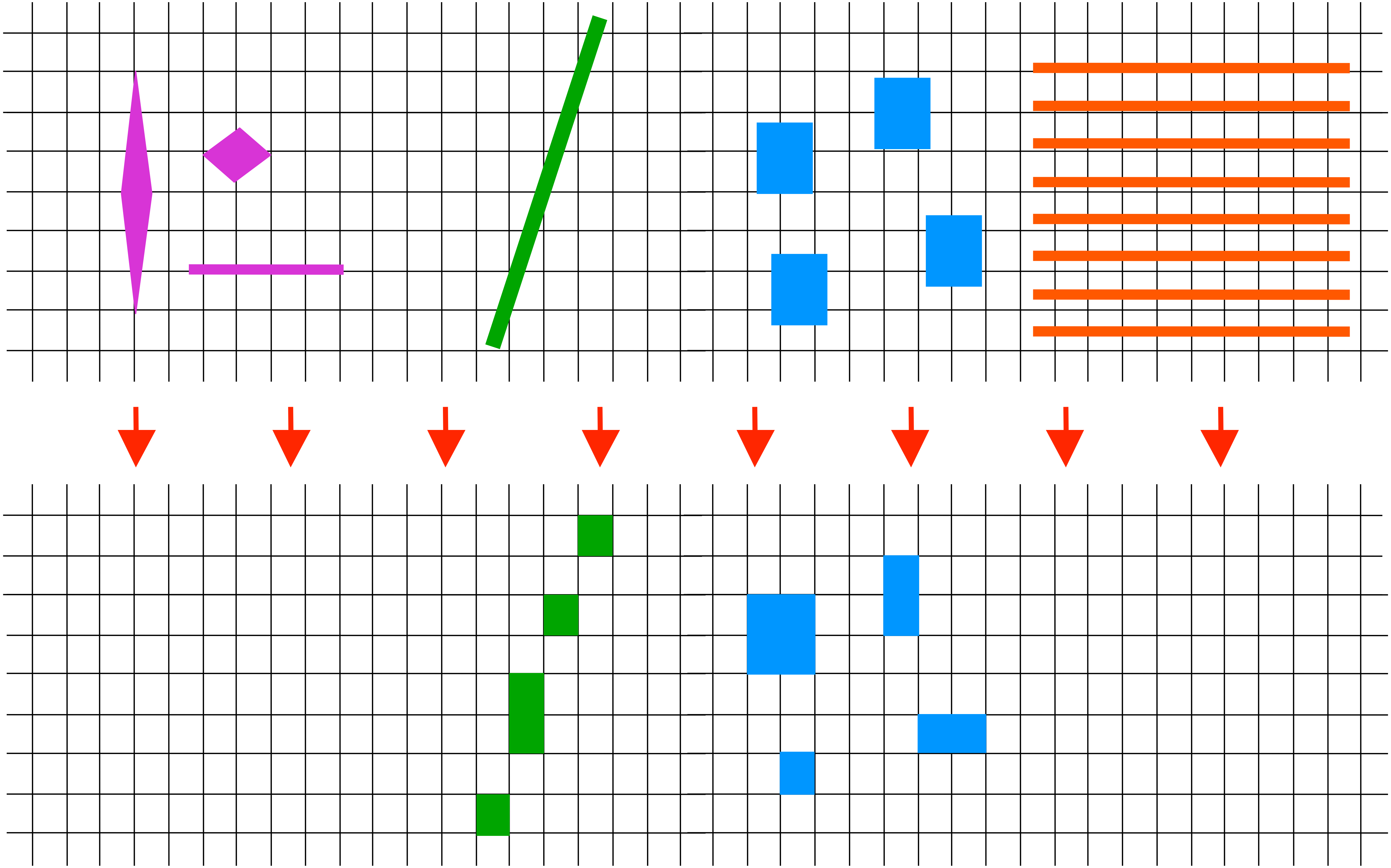
Aliasing: Sampling Errors



Aliasing: Sampling Errors

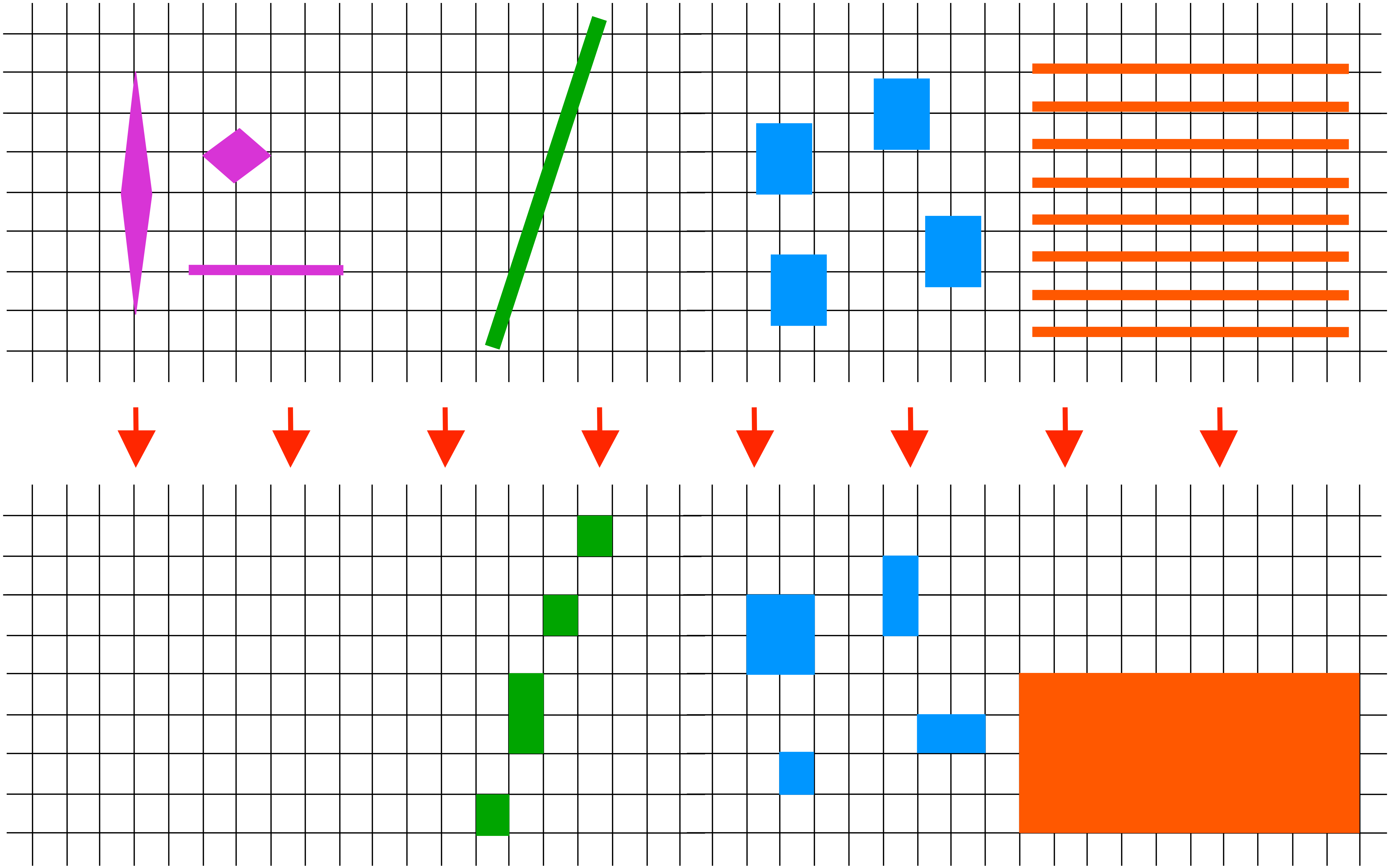


Aliasing: Sampling Errors

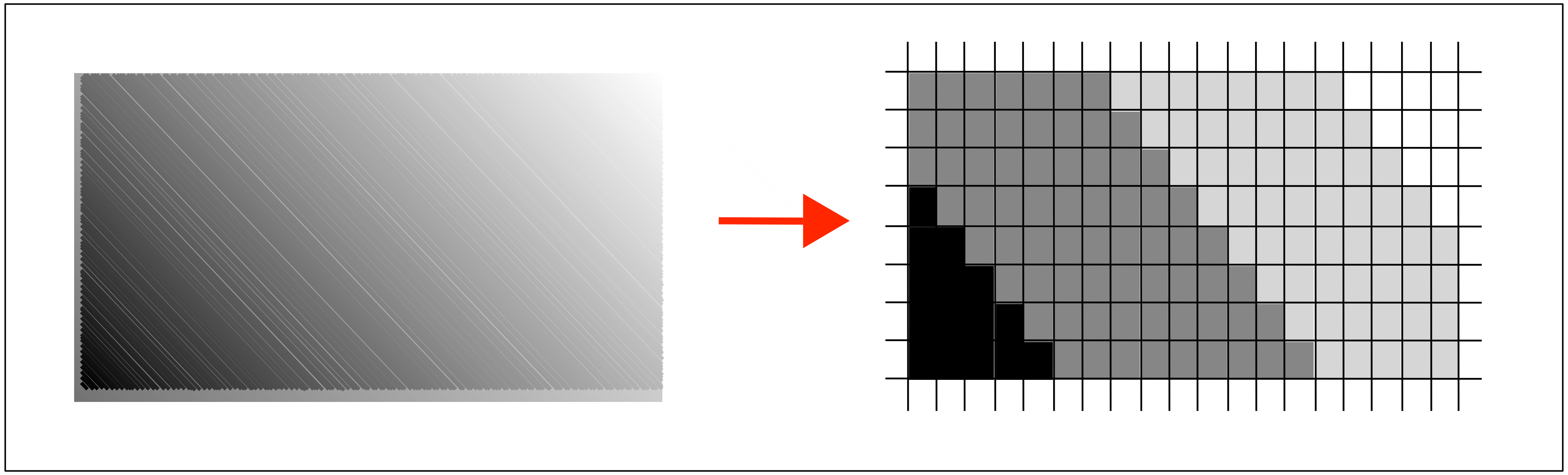


Computer Graphics Charles University

Aliasing: Sampling Errors

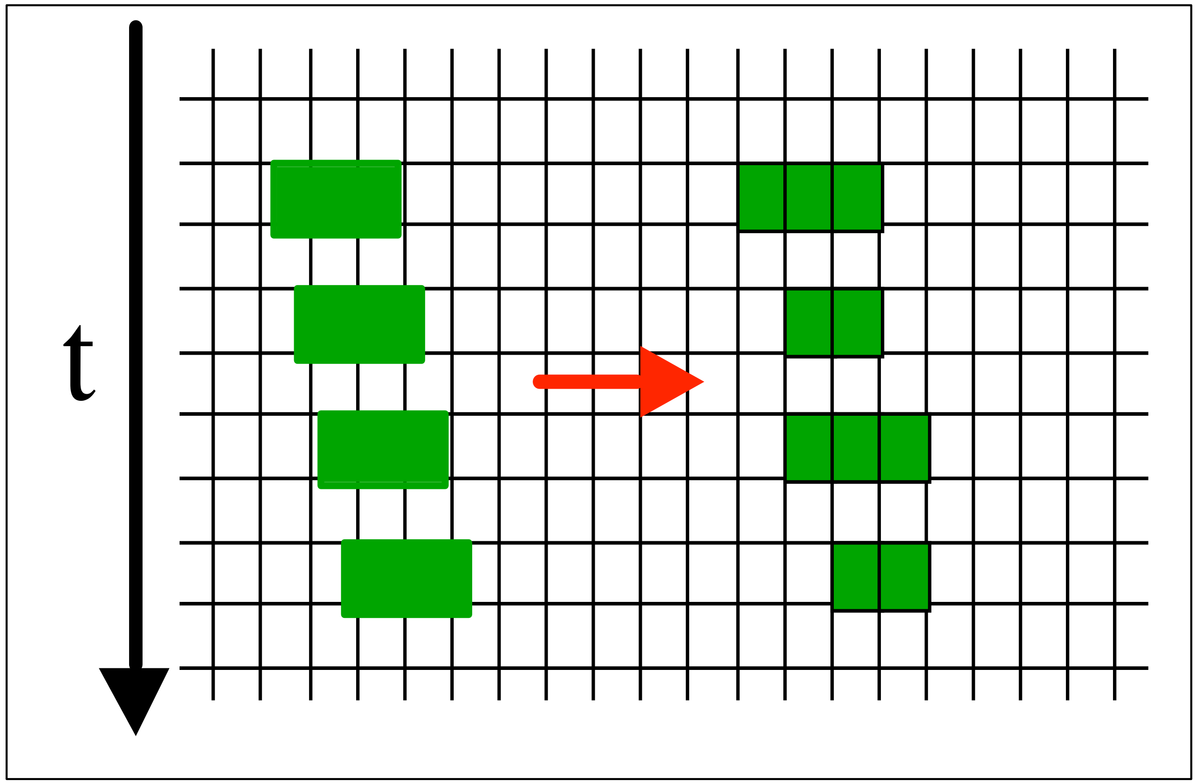


Aliasing: Colour Ramps

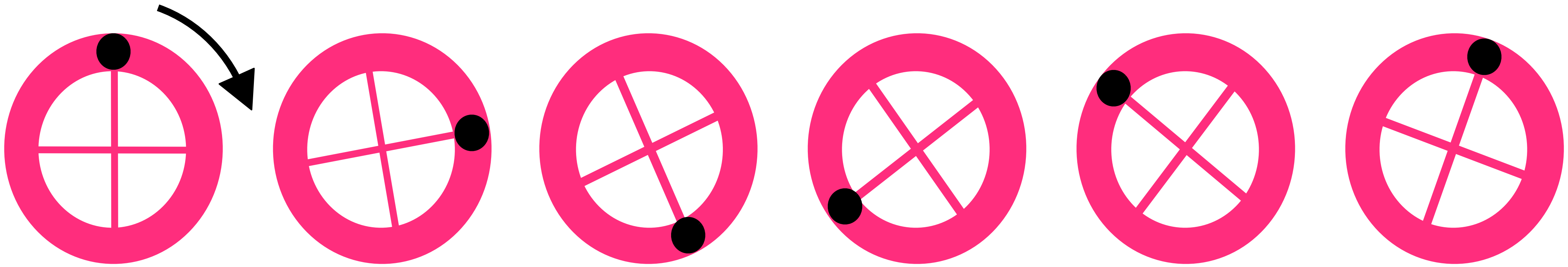


Animation Aliasing

- Jumpy images
- "worming“

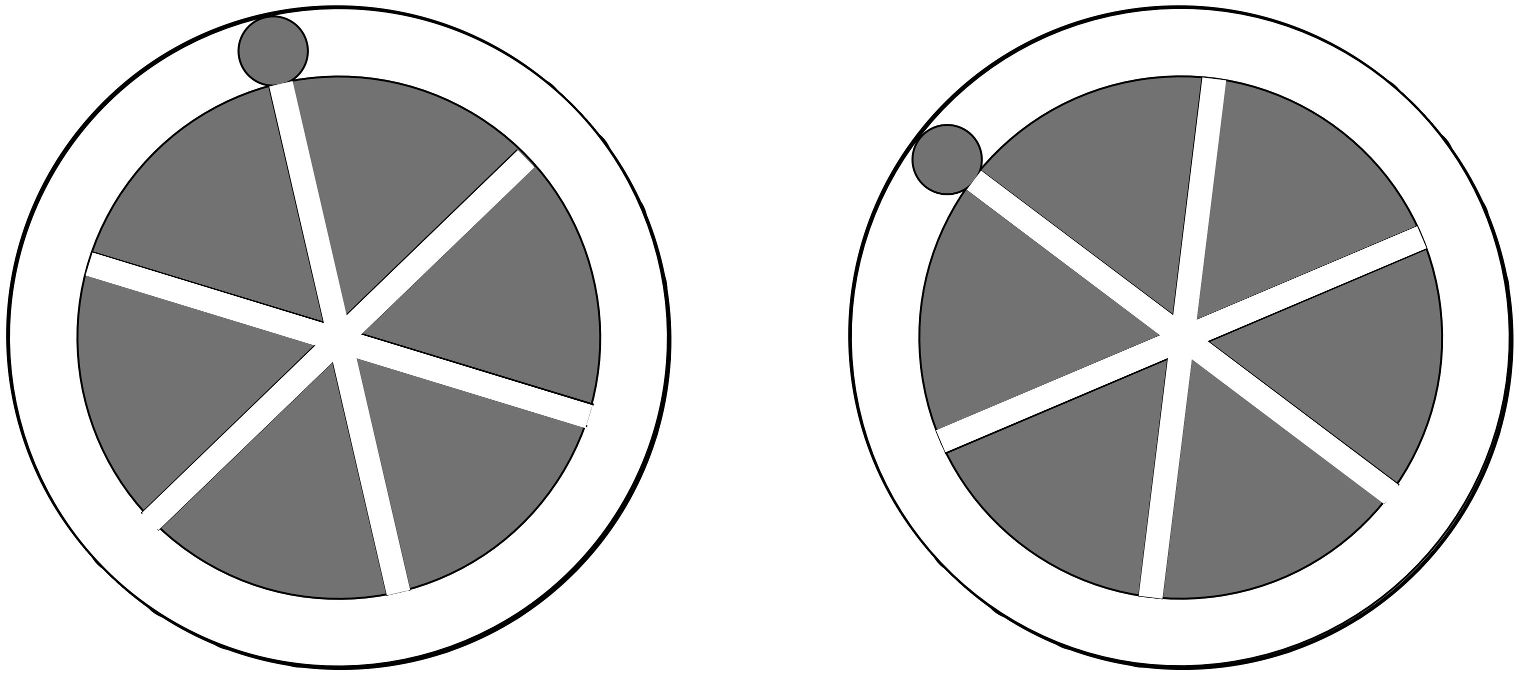


- Wheels turning backwards



Wheels turning backwards

Wheels turning backwards



Solution Strategies

- “More effort”
  - Ad-hoc solutions
  - Higher resolution
  - Higher colour depth
  - Faster image refresh
- “Use your brain”
  - Understanding the problem
  - Efficient counter-measures

Solution Strategies

Oft aufwendig,
nicht immer
zielführend,
manchmal
unmöglich

- “More effort”
  - Ad-hoc solutions
  - Higher resolution
  - Higher colour depth
  - Faster image refresh
- “Use your brain”
  - Understanding the problem
  - Efficient counter-measures

Computer
Graphics
Charles
University

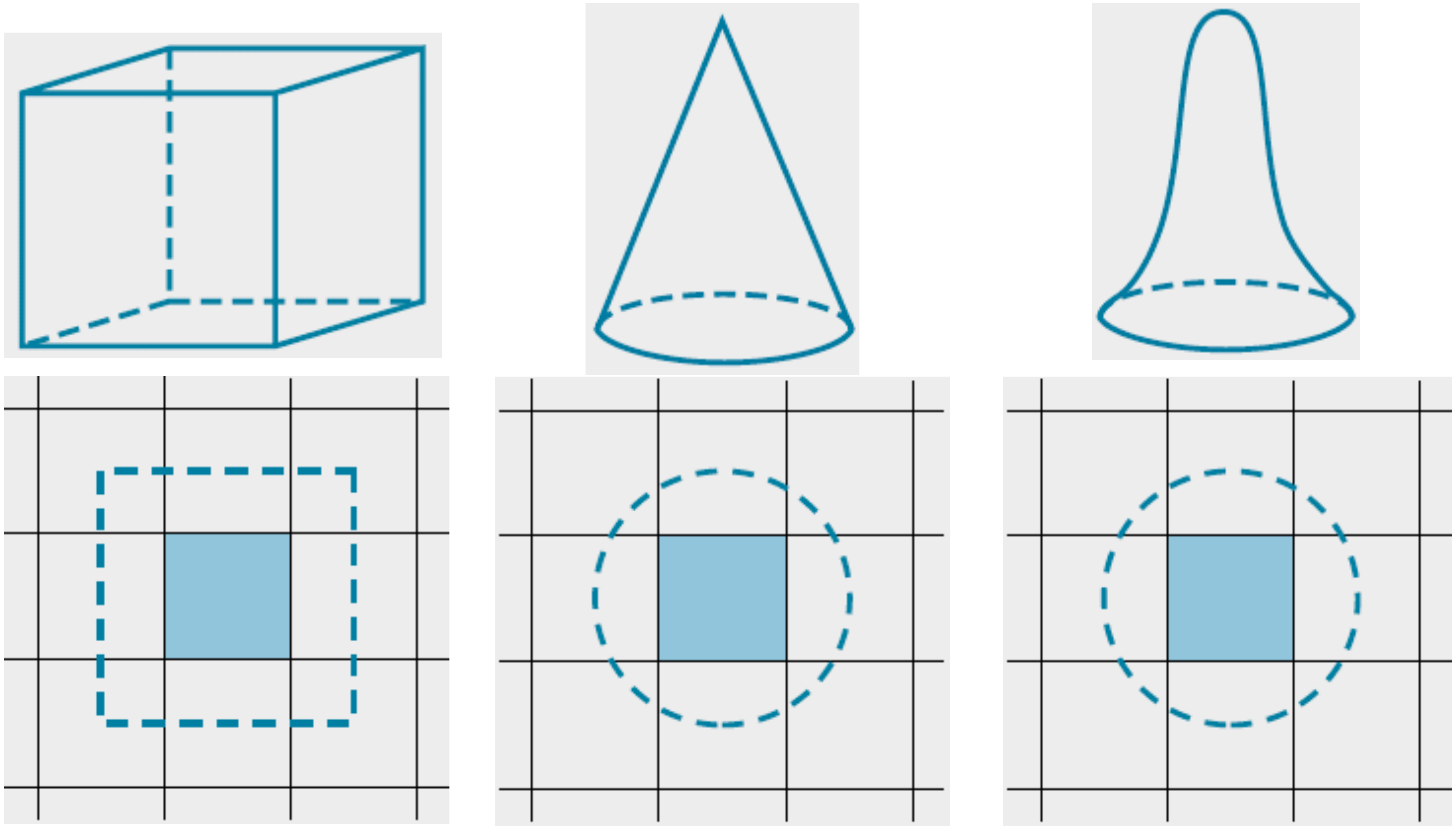
Solution Strategies

Oft aufwendig,
nicht immer
zielführend,
manchmal
unmöglich

- "More effort"
  - Ad-hoc solutions
  - Higher resolution
  - Higher colour depth
  - Faster image refresh
- "Use your brain"
  - Understanding the problem
  - Efficient counter-measures

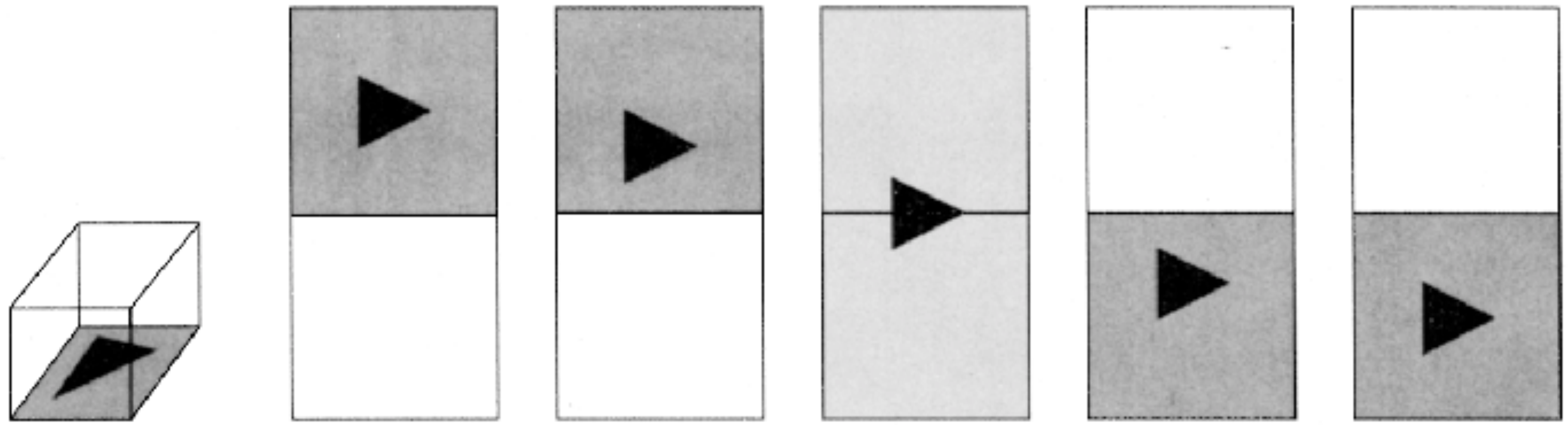
machbar

Simple Splatting Kernels

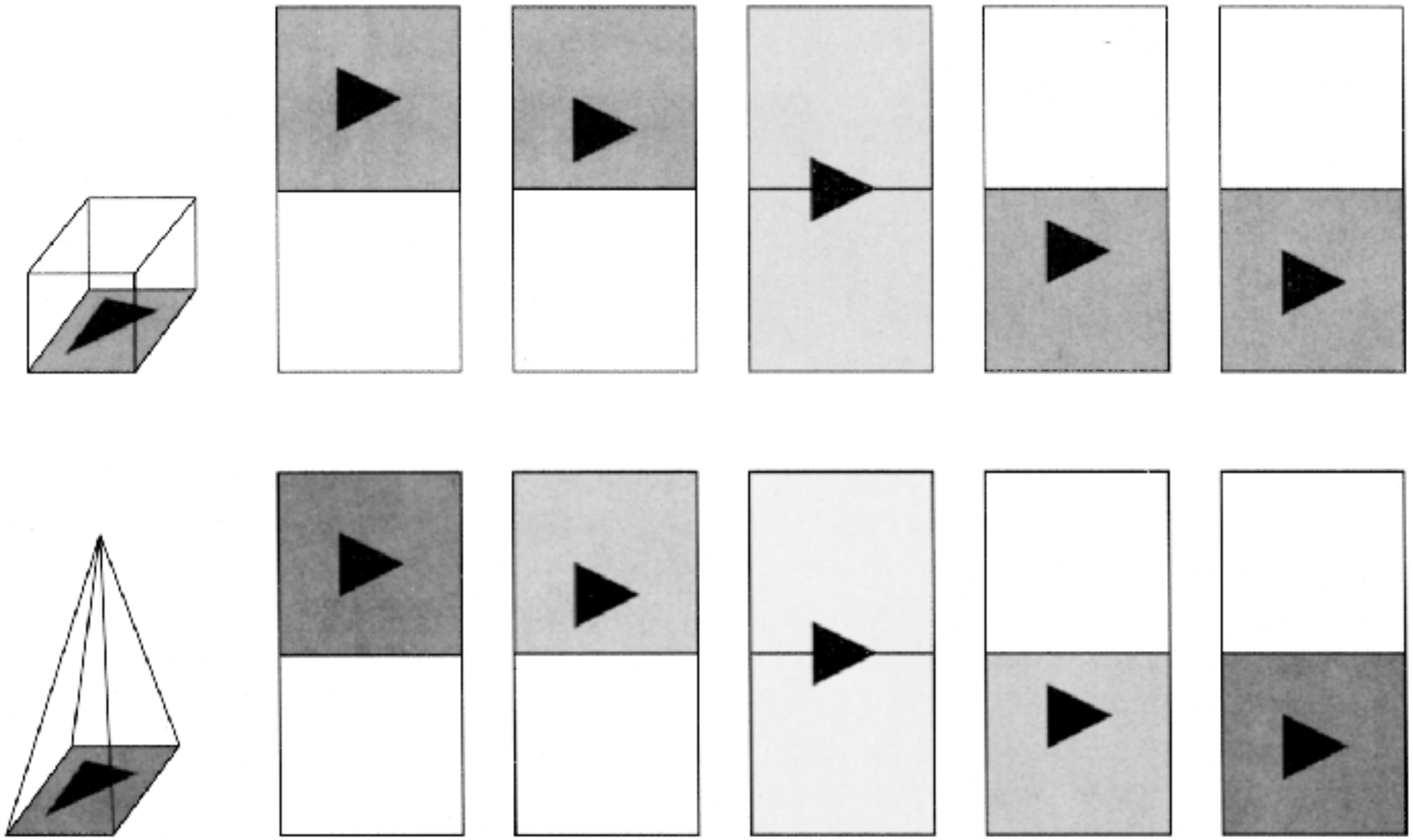


box filter cone Gaussian filter

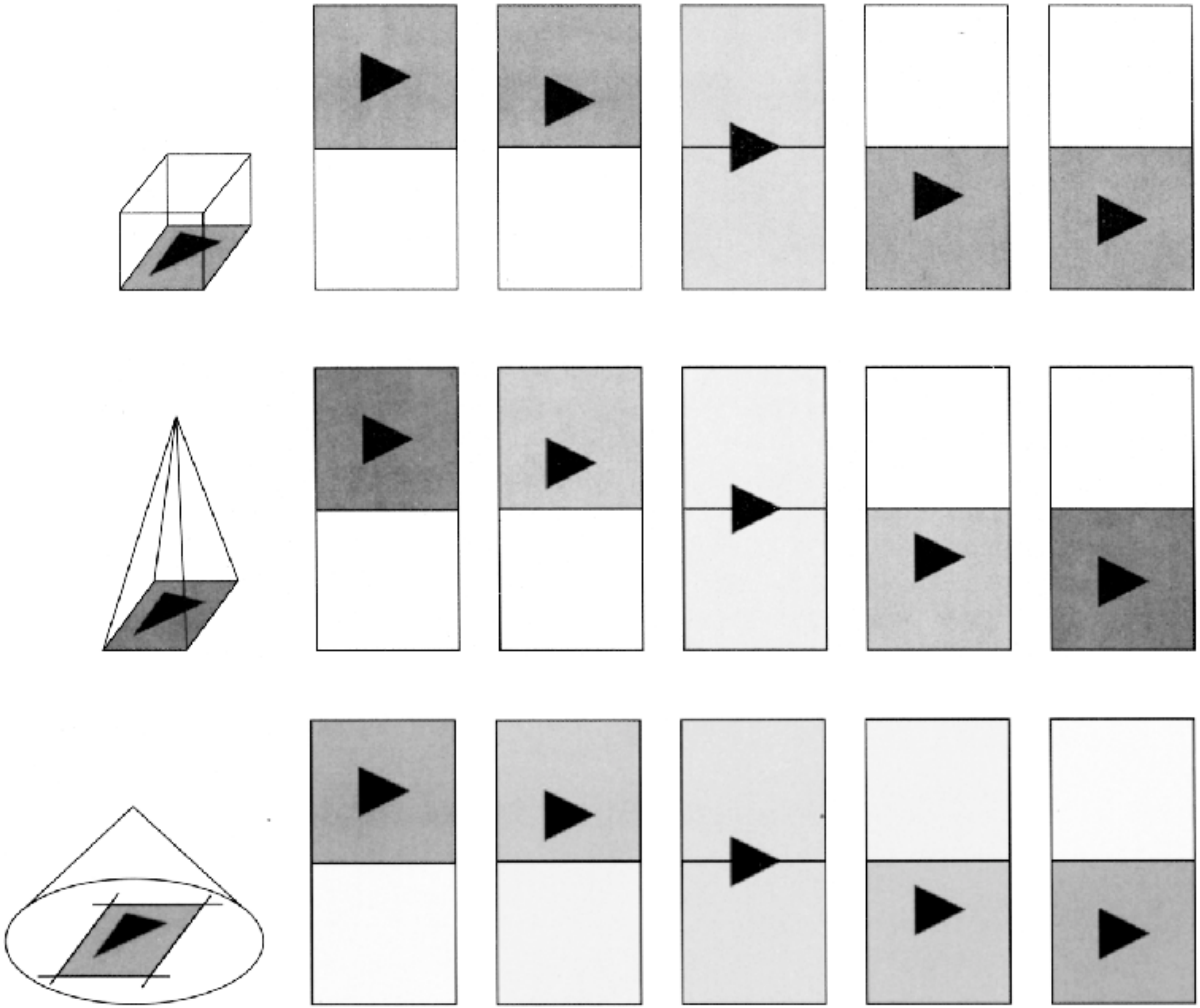
Filter Kernels in Action



Filter Kernels in Action

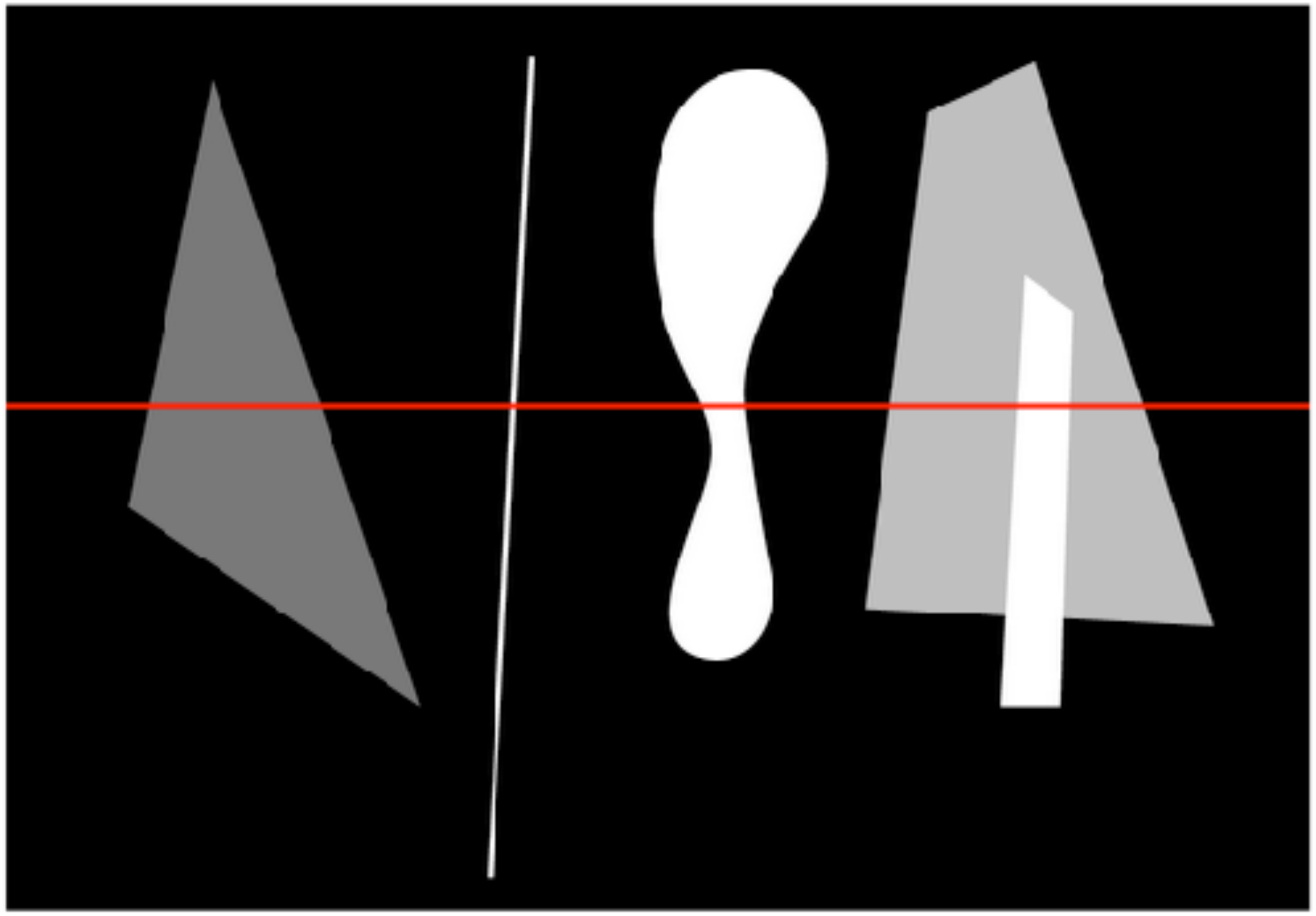


Filter Kernels in Action

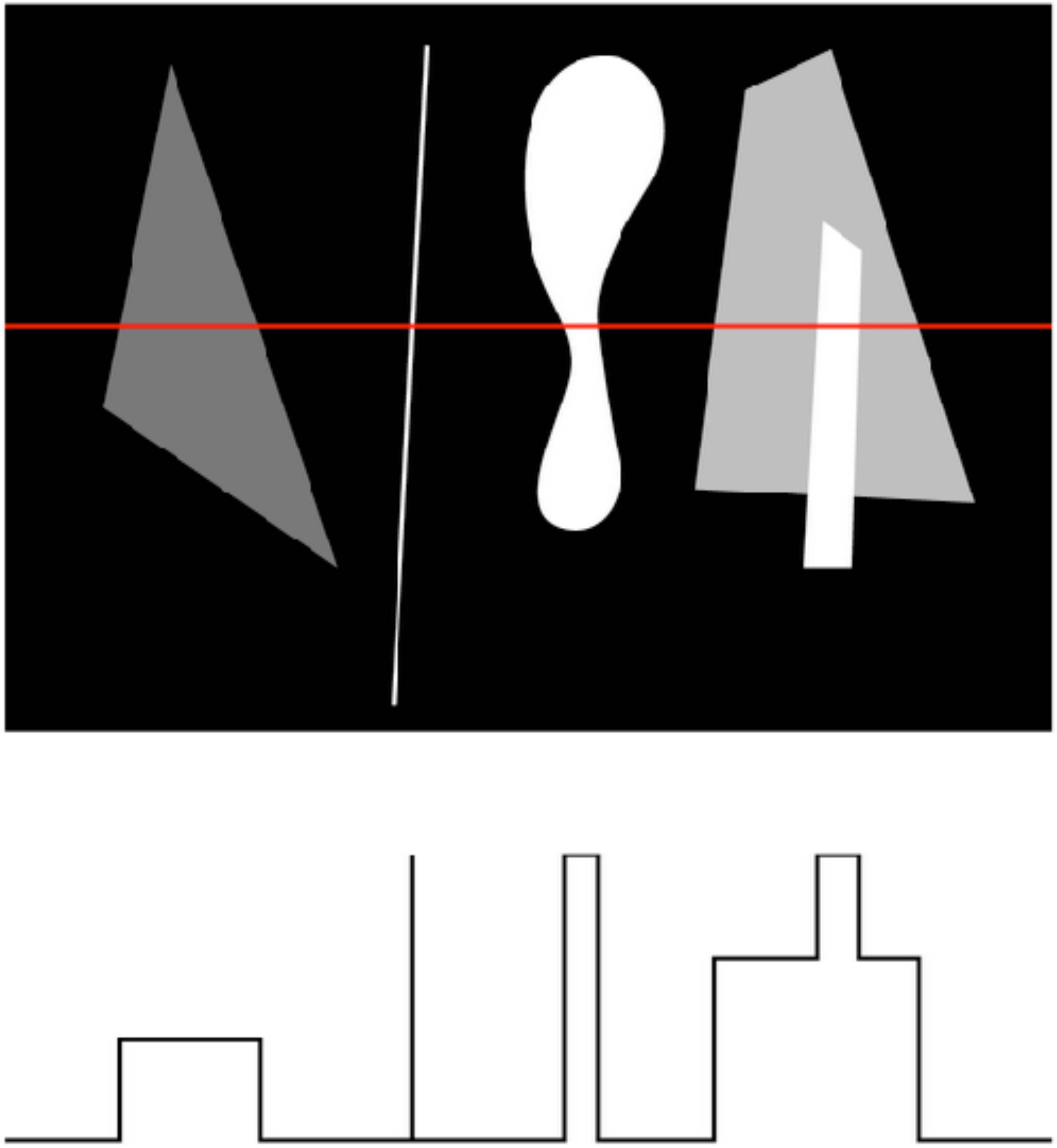


Computer Graphics Charles University

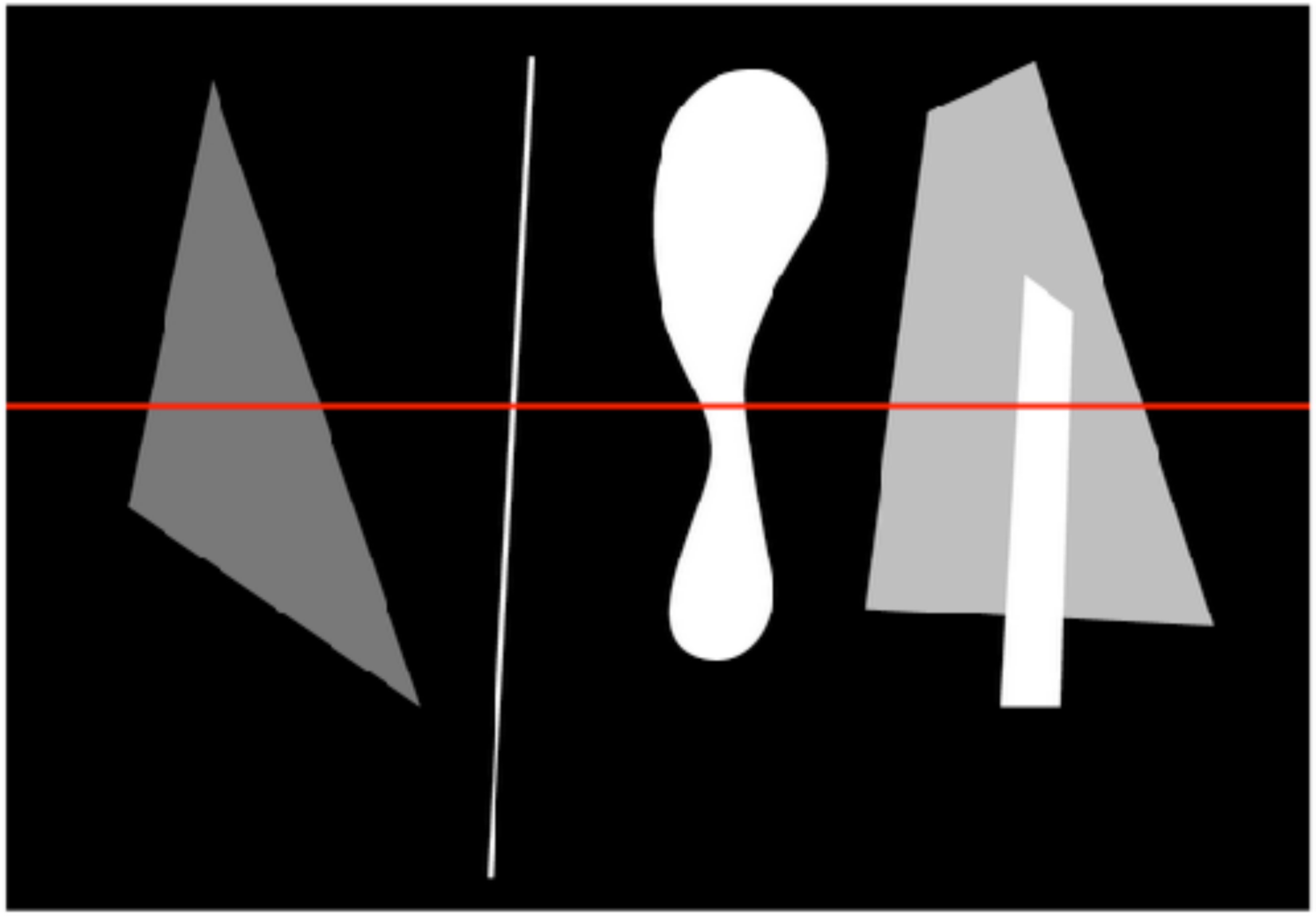
Scanline as Signal

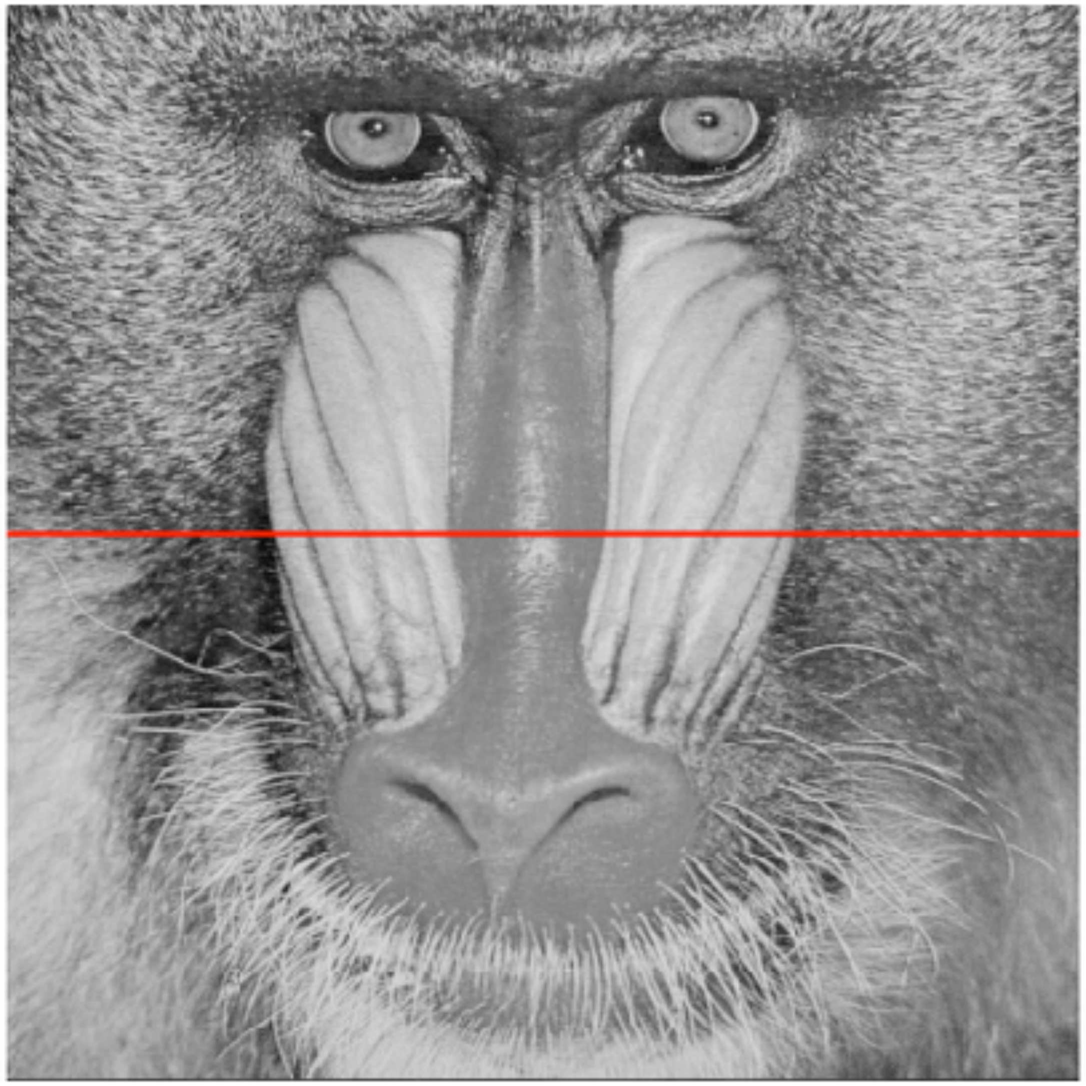


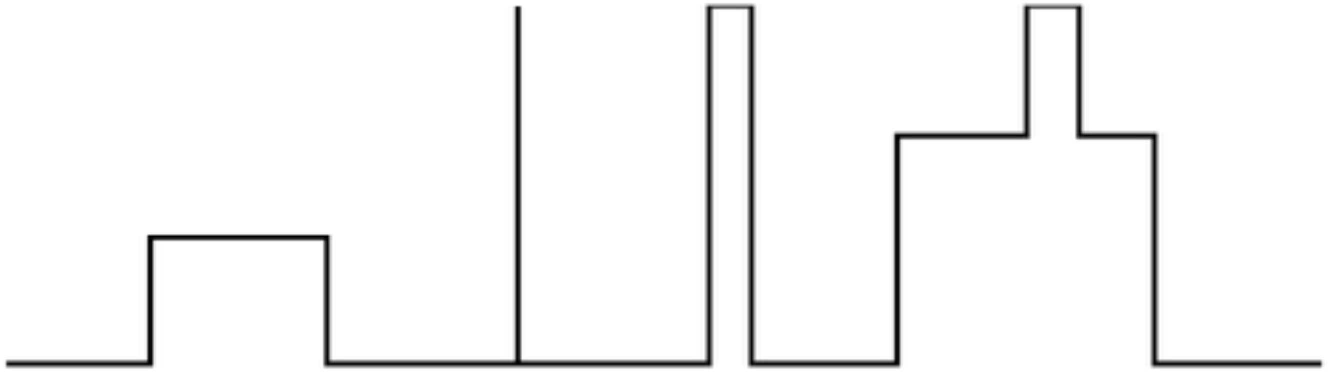
Scanline as Signal



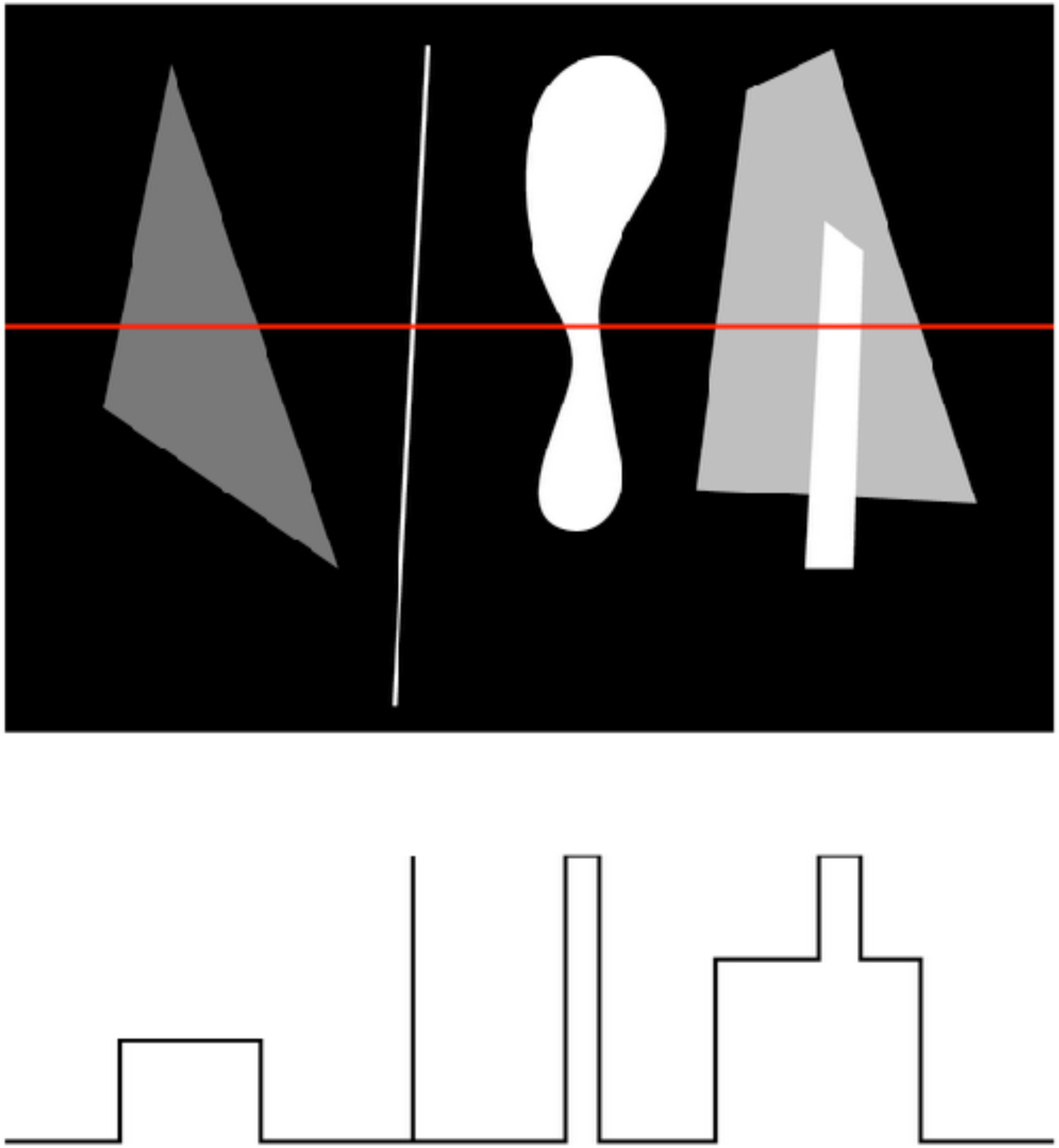
Scanline as Signal

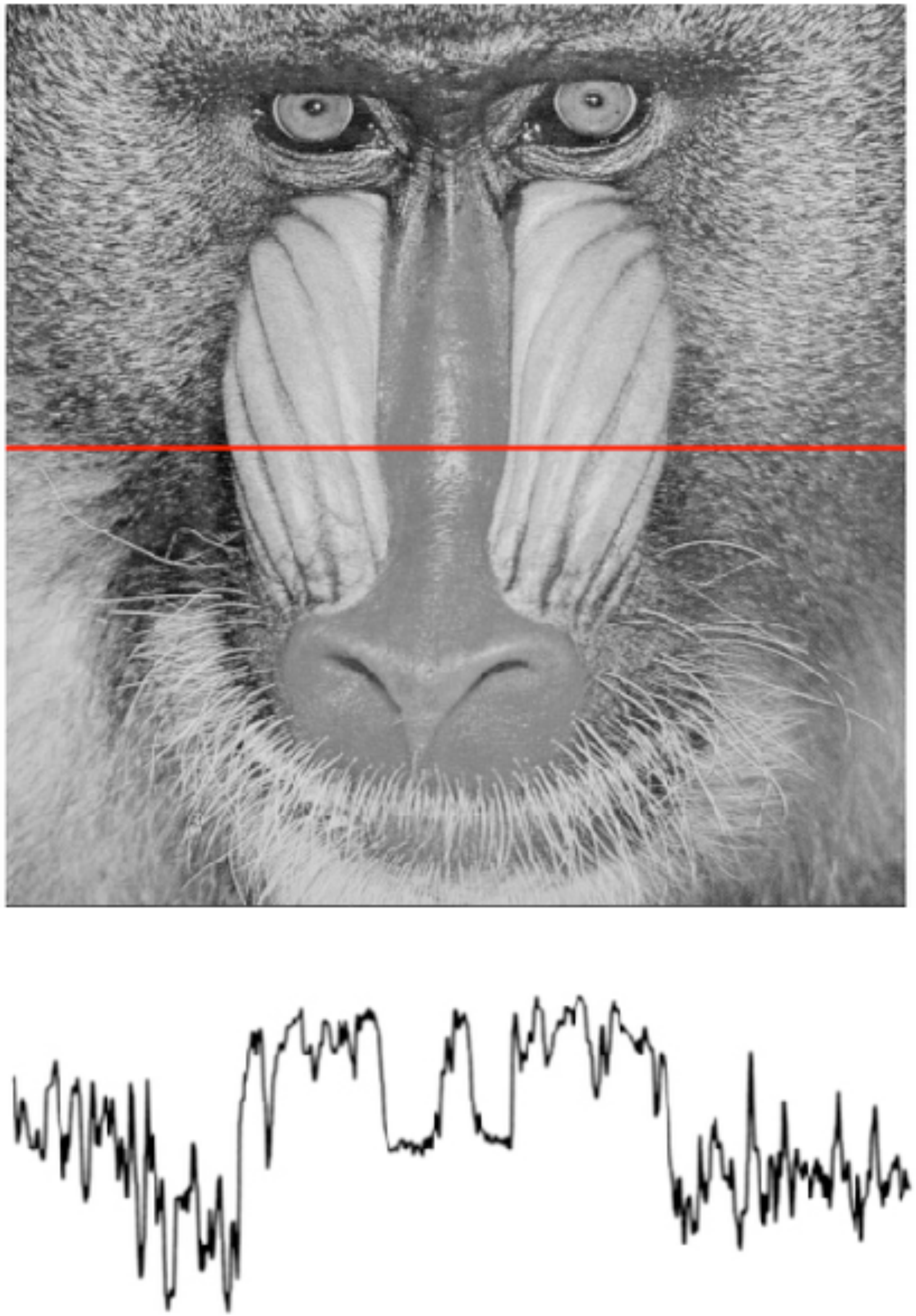




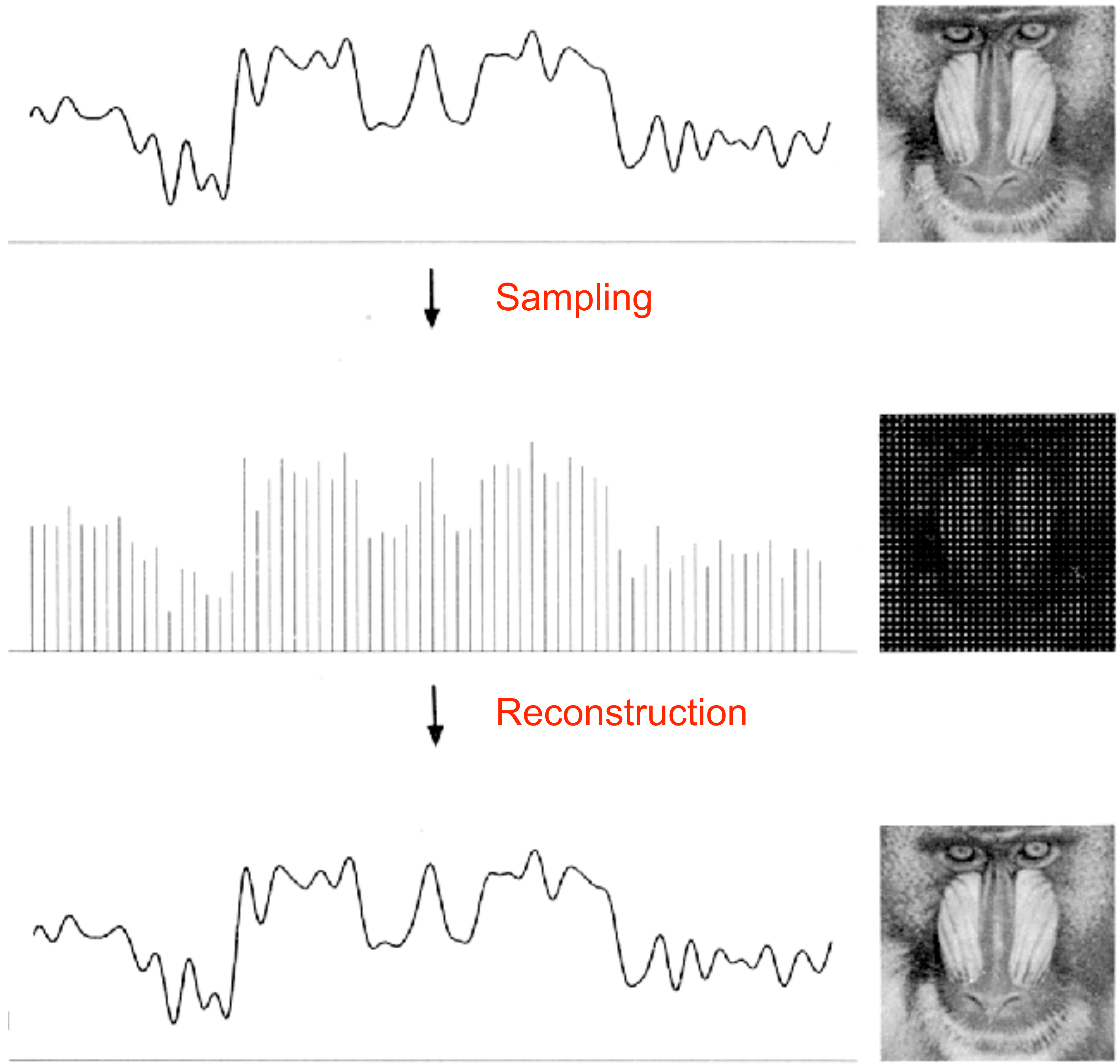


Scanline as Signal



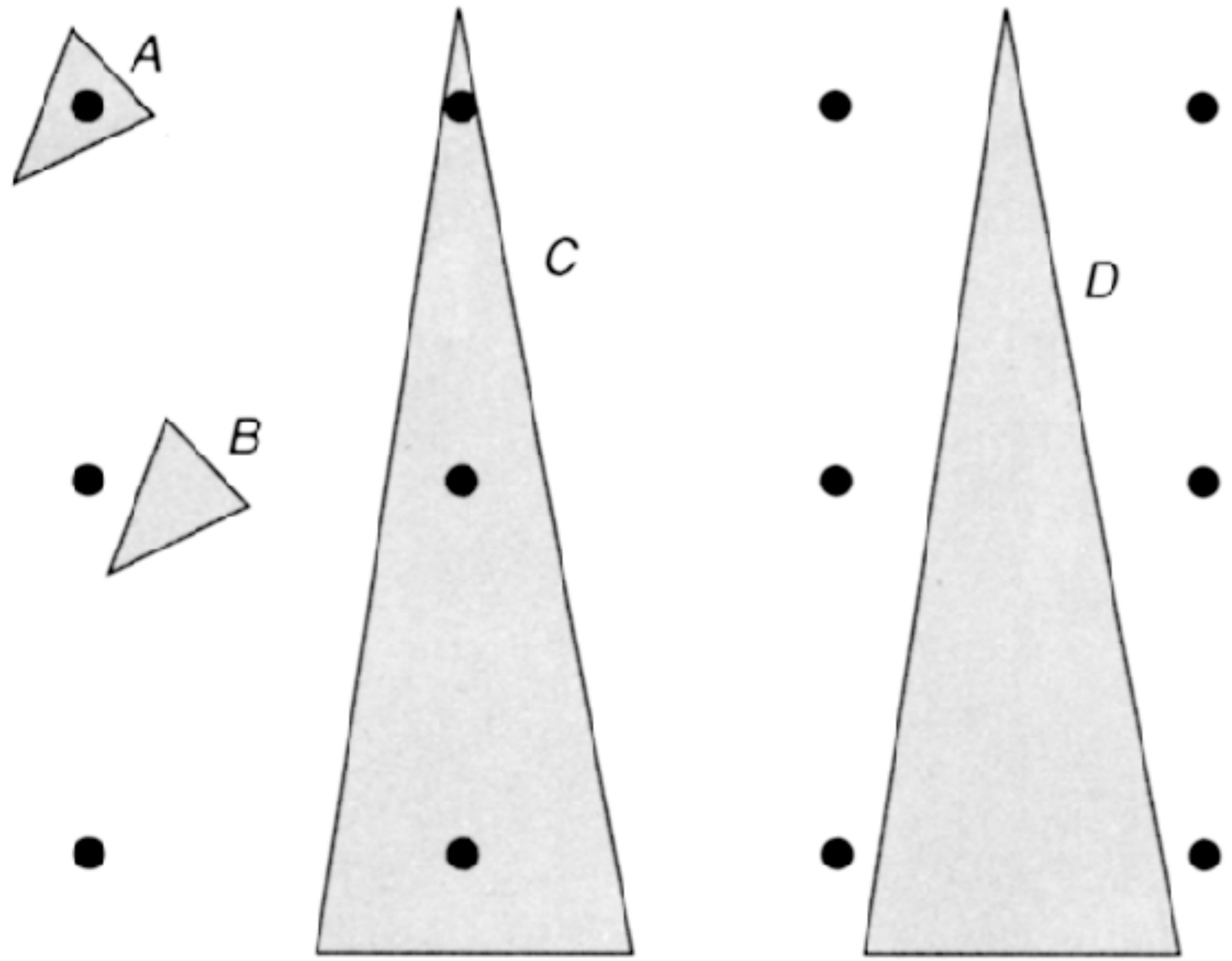


Storing a Scanline



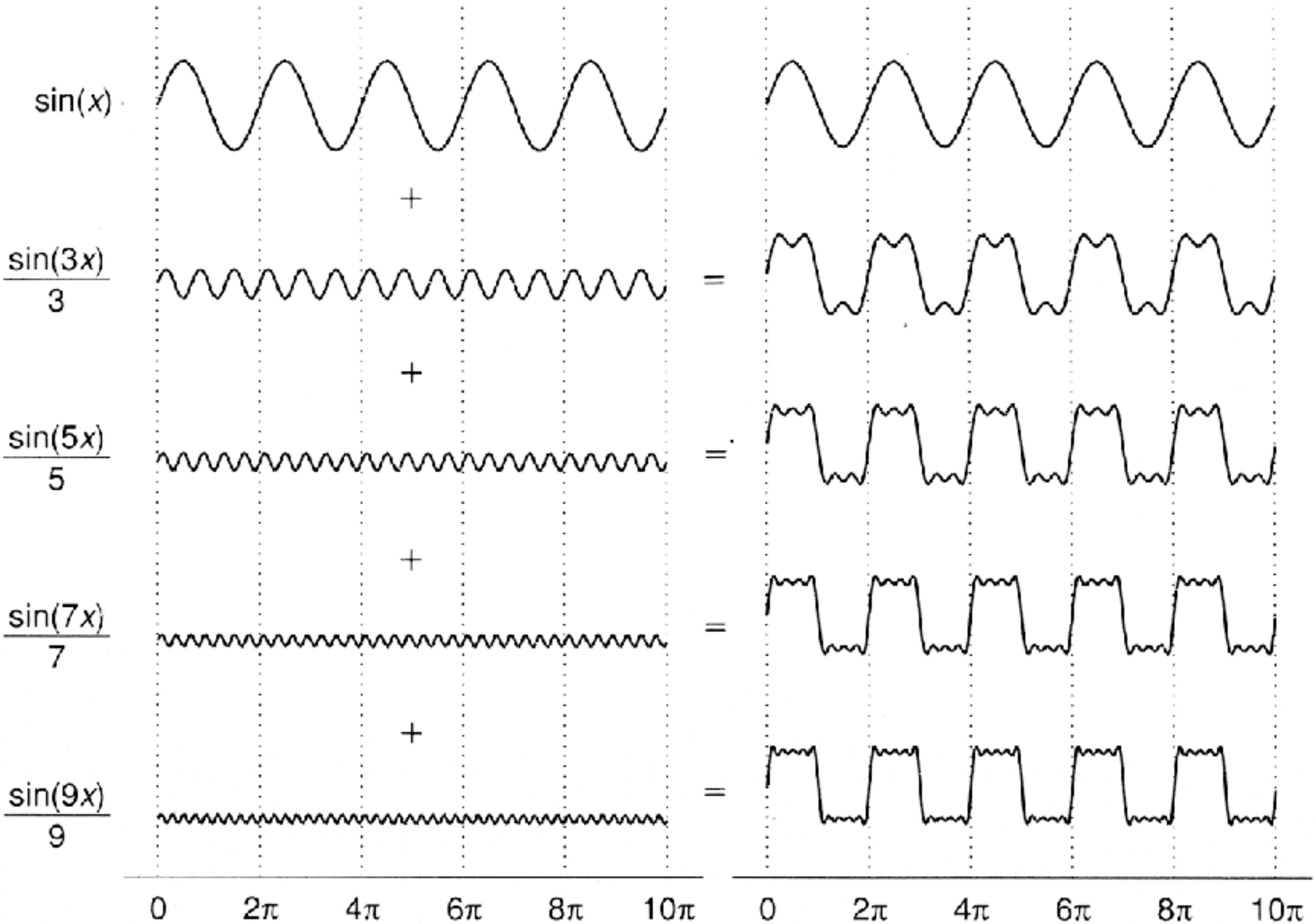
Computer Graphics Charles University

Sampling: Basic Problem

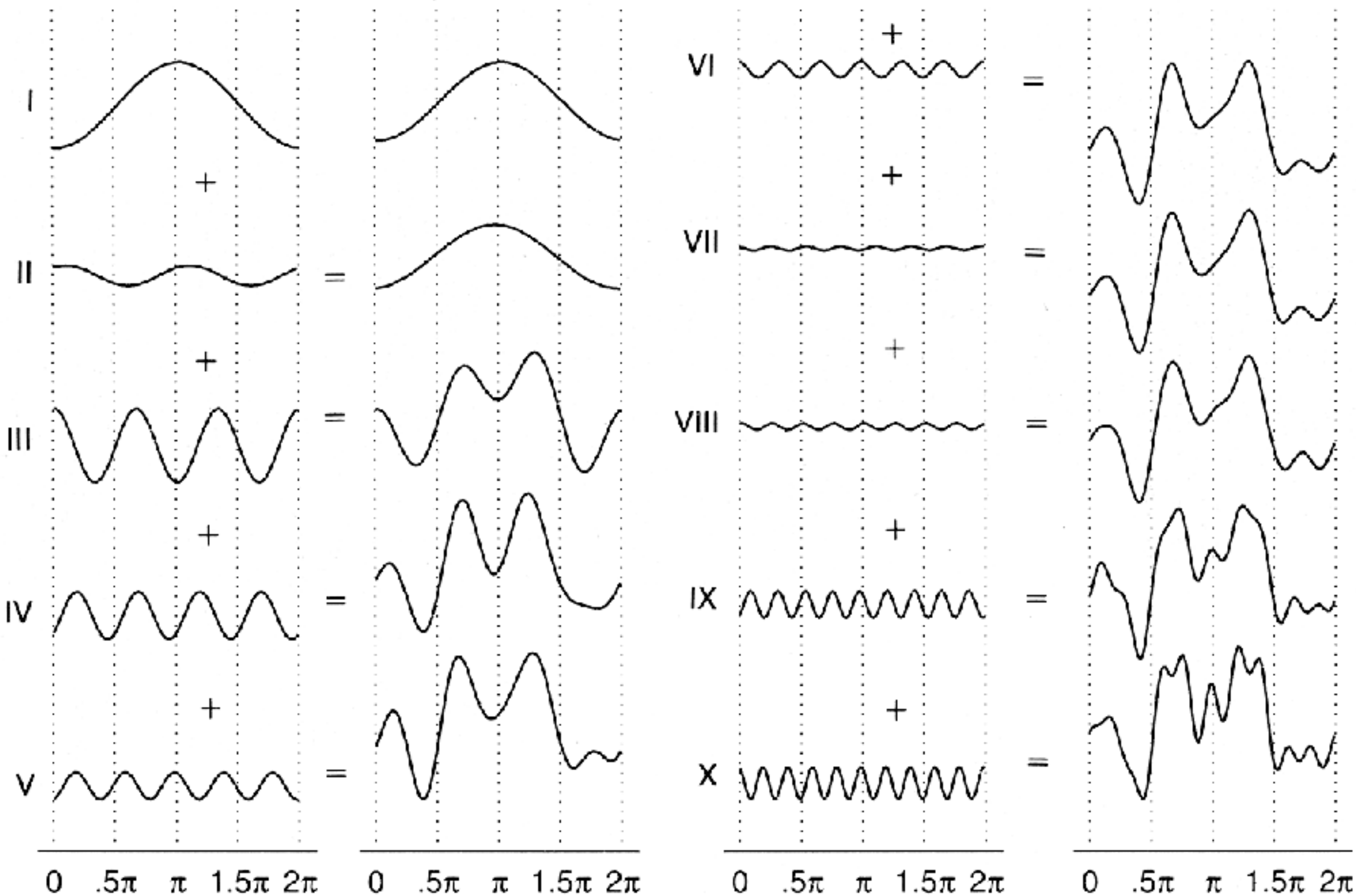


- **Without at least some knowledge about the sampled signal, we cannot guarantee that this will work!**
  - **At all!**

Square Wave Decomposition



Signal Decomposition



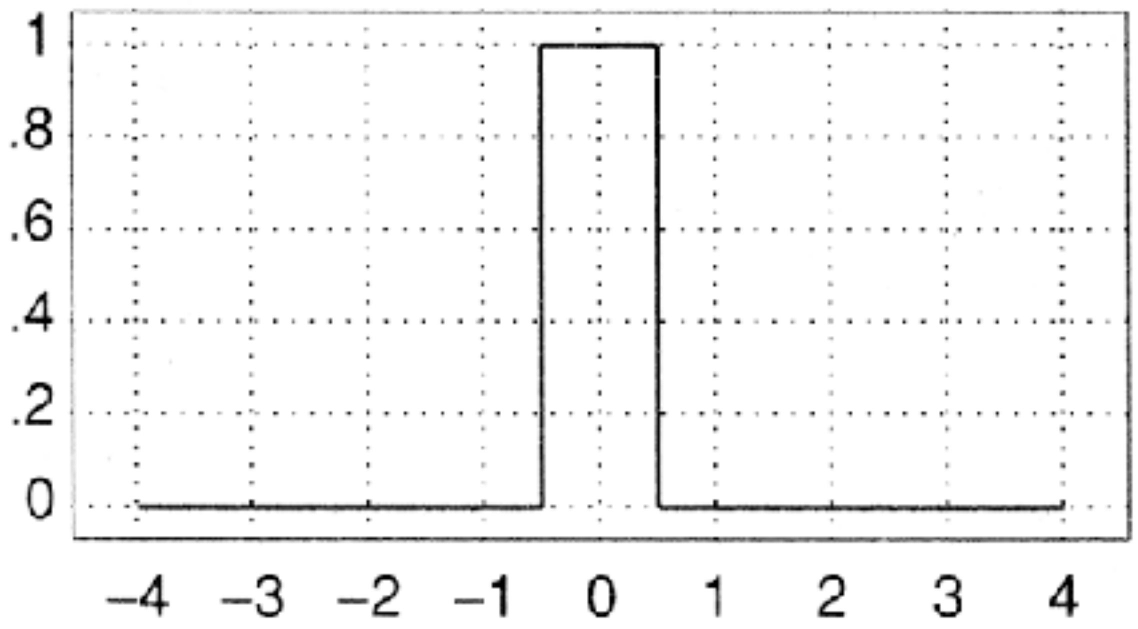
Fourier Transform

- **Connects time and frequency domain**
- **Applicable for arbitrary signals**
  - **f(x) time domain**
  - **F(u) frequency domain**

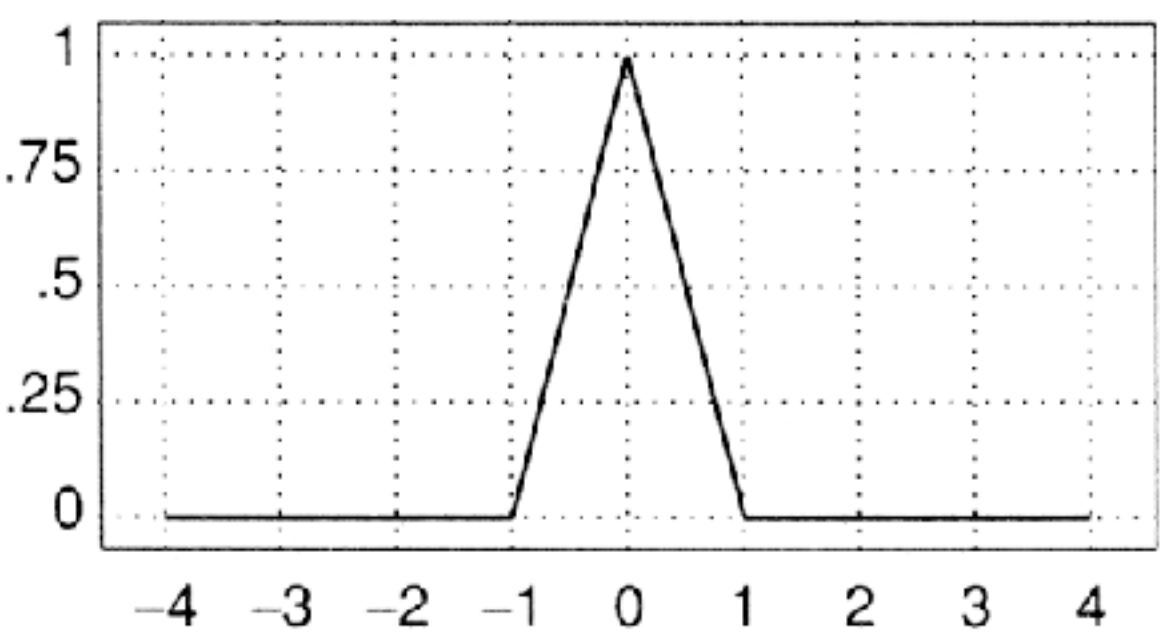
$$F(u) = \int_{-\infty}^{+\infty} f(x)[\cos 2\pi ux - i \sin 2\pi\, ux]\, dx$$

$$f(x) = \int_{-\infty}^{+\infty} F(u)[\cos 2\pi ux - i \sin 2\pi\, ux]\, du$$

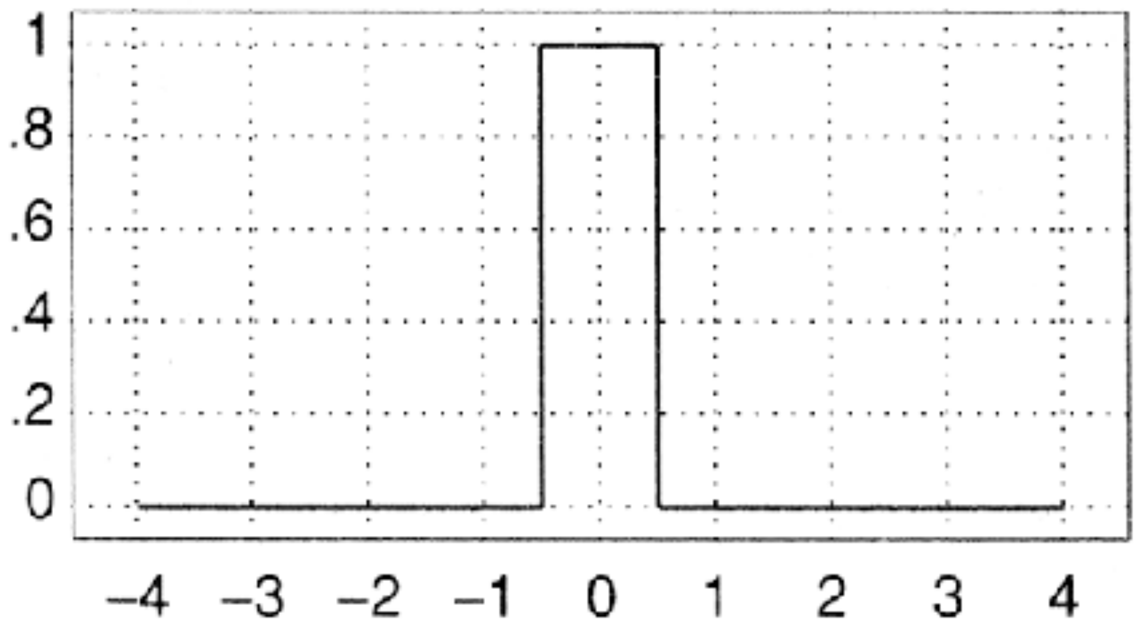
Box & Sawtooth



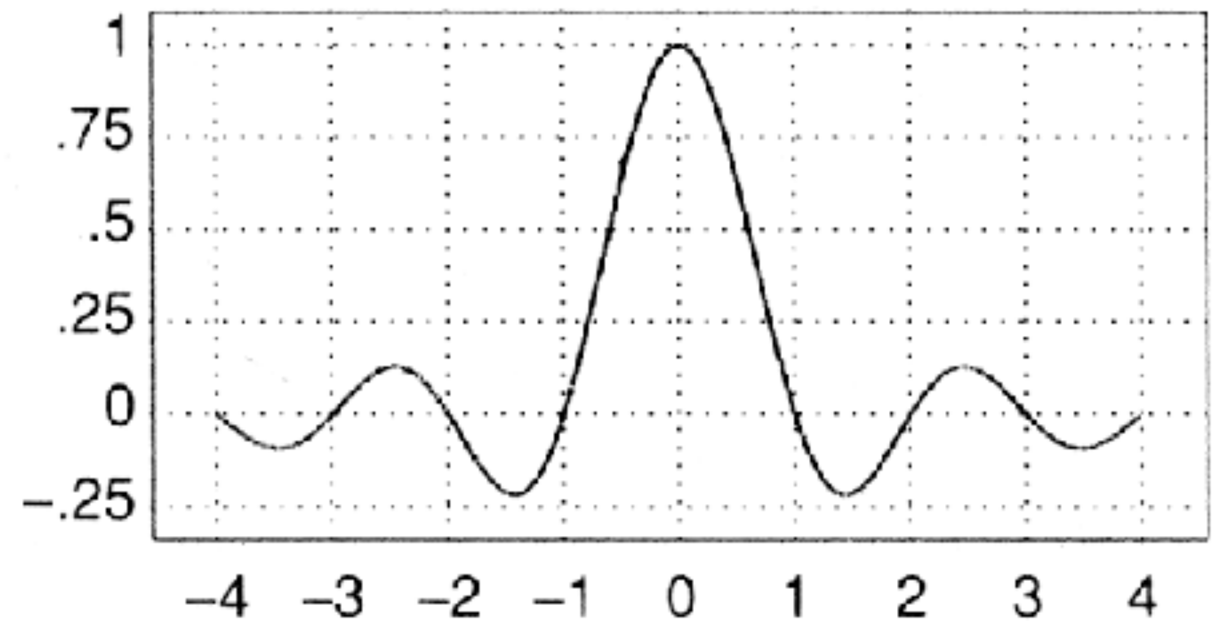
$f(x)$ $\quad$ $F(u)$



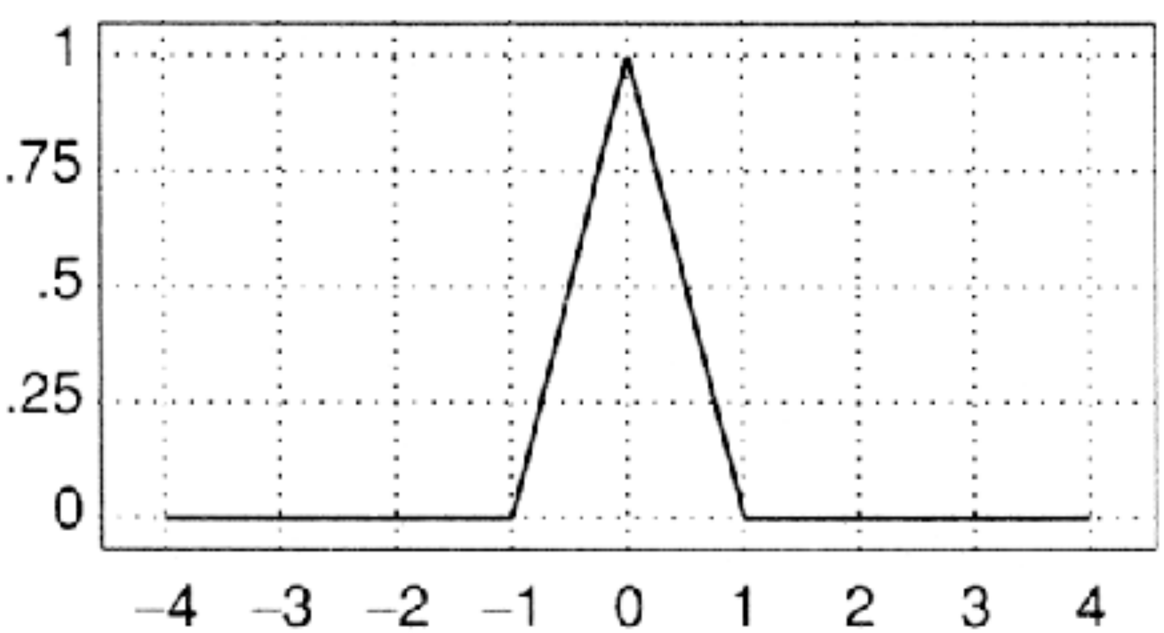
Box & Sawtooth



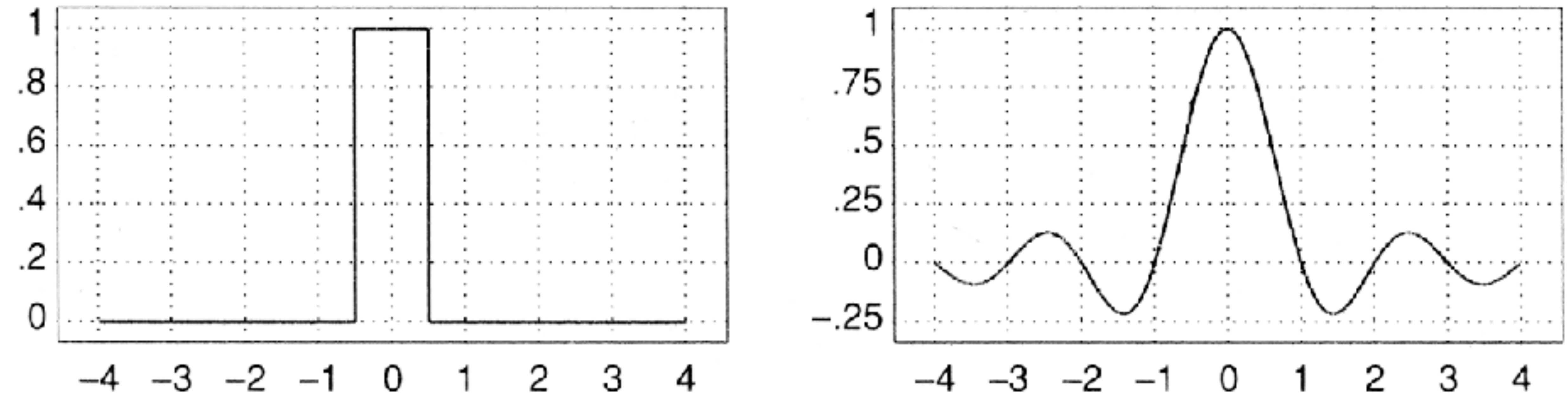
$f(x)$



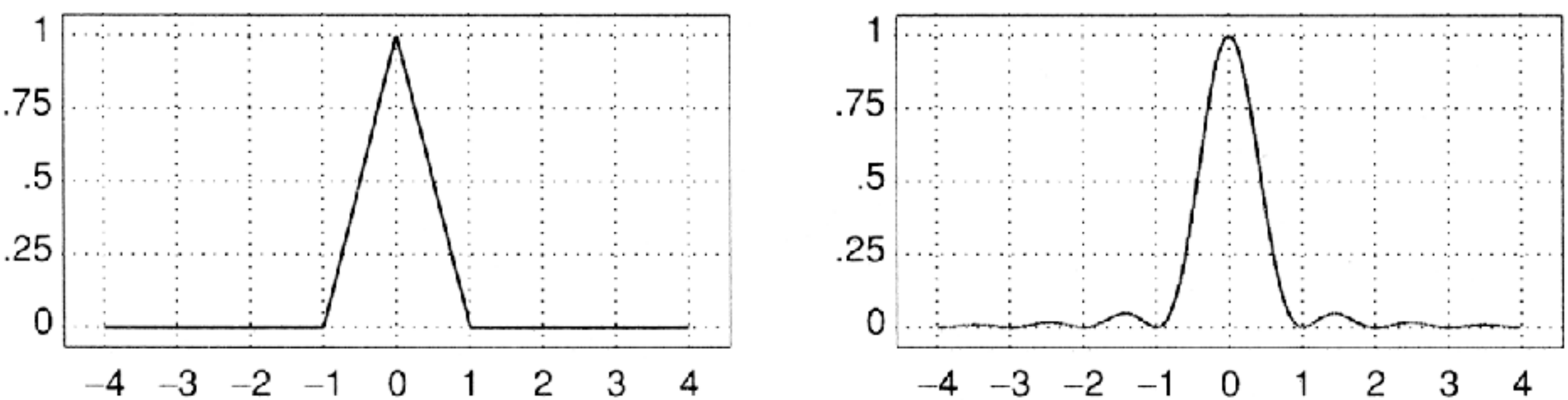
$F(u)$



Box & Sawtooth

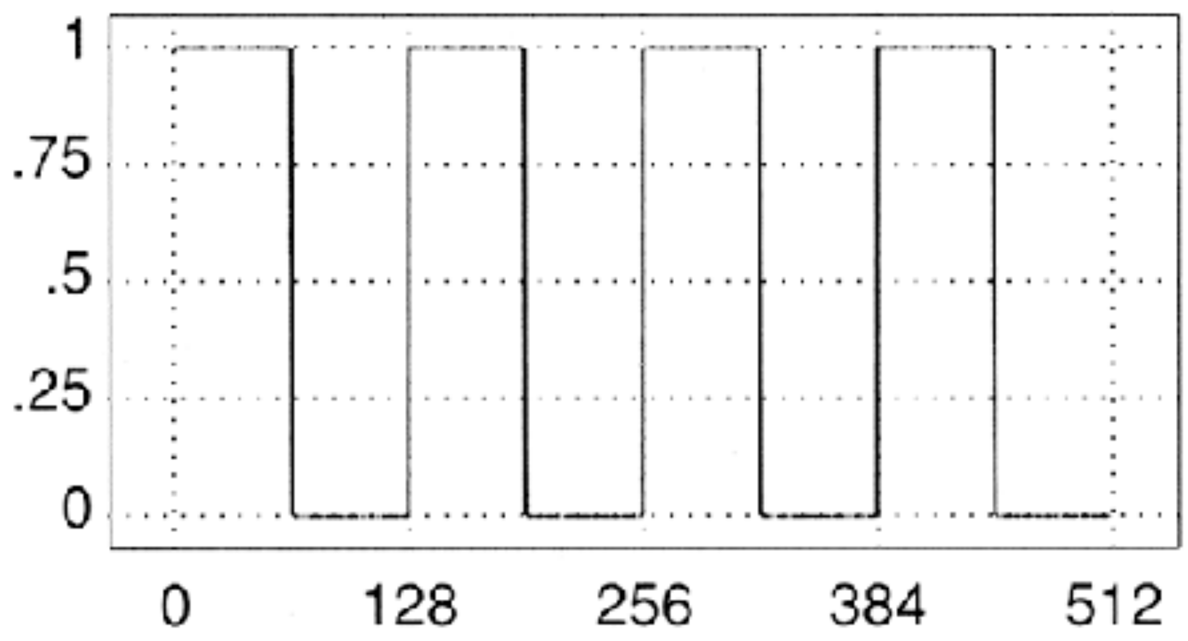


$f(x)$ $F(u)$



Computer Graphics Charles University

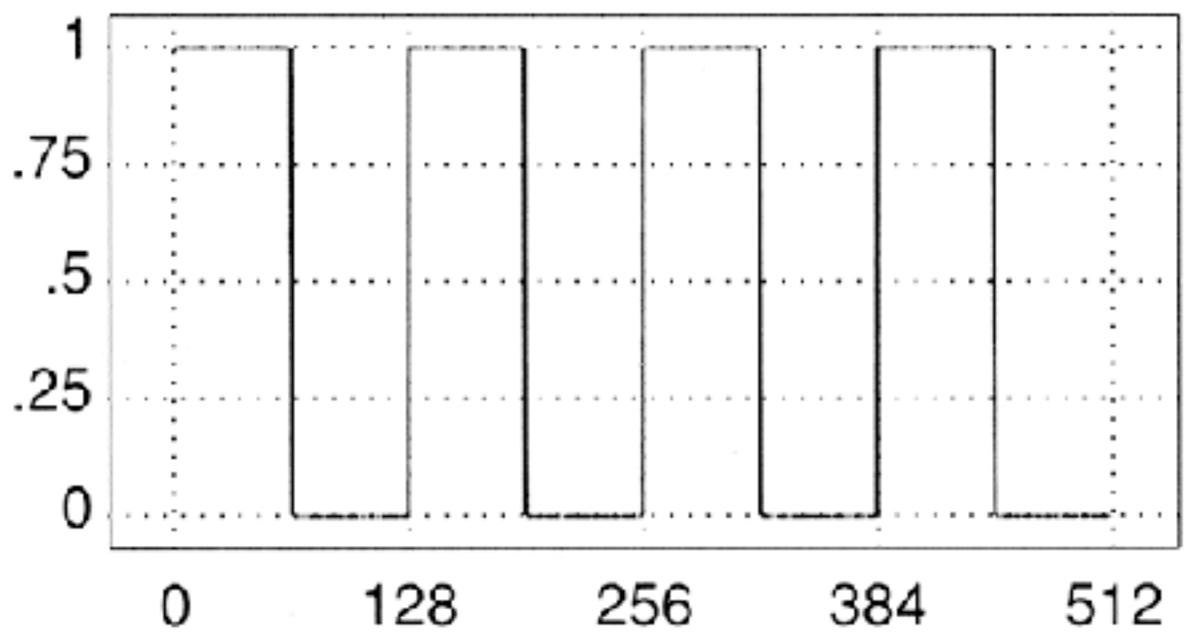
Square Wave & Scanline



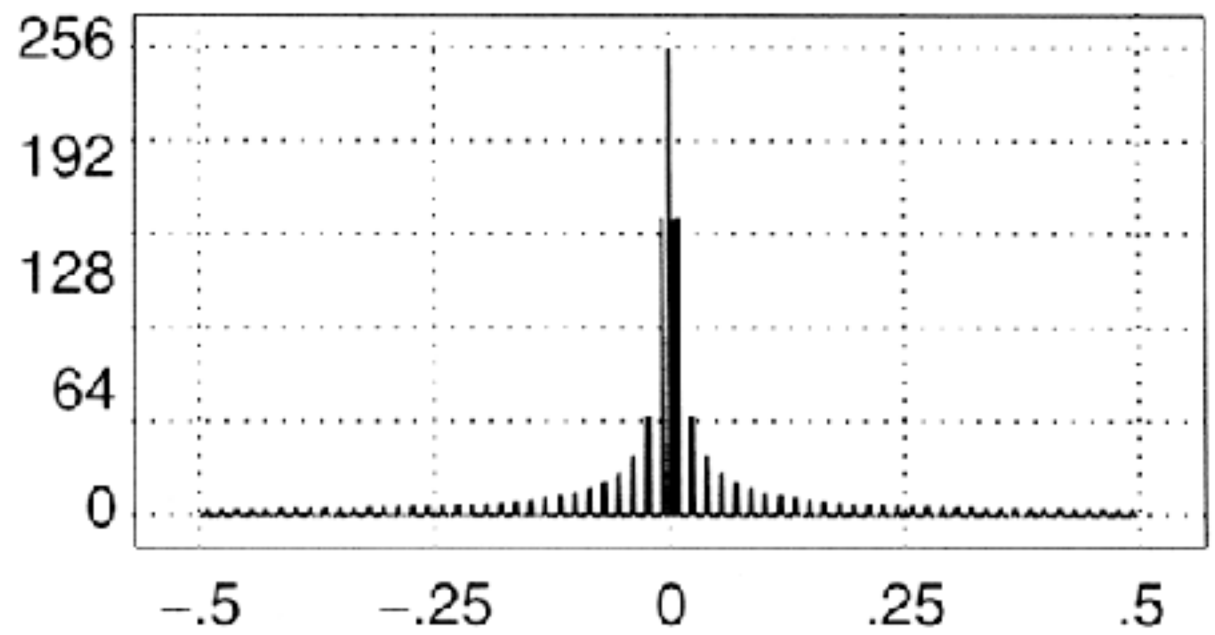
$f(x)$

$F(u)$

Square Wave & Scanline

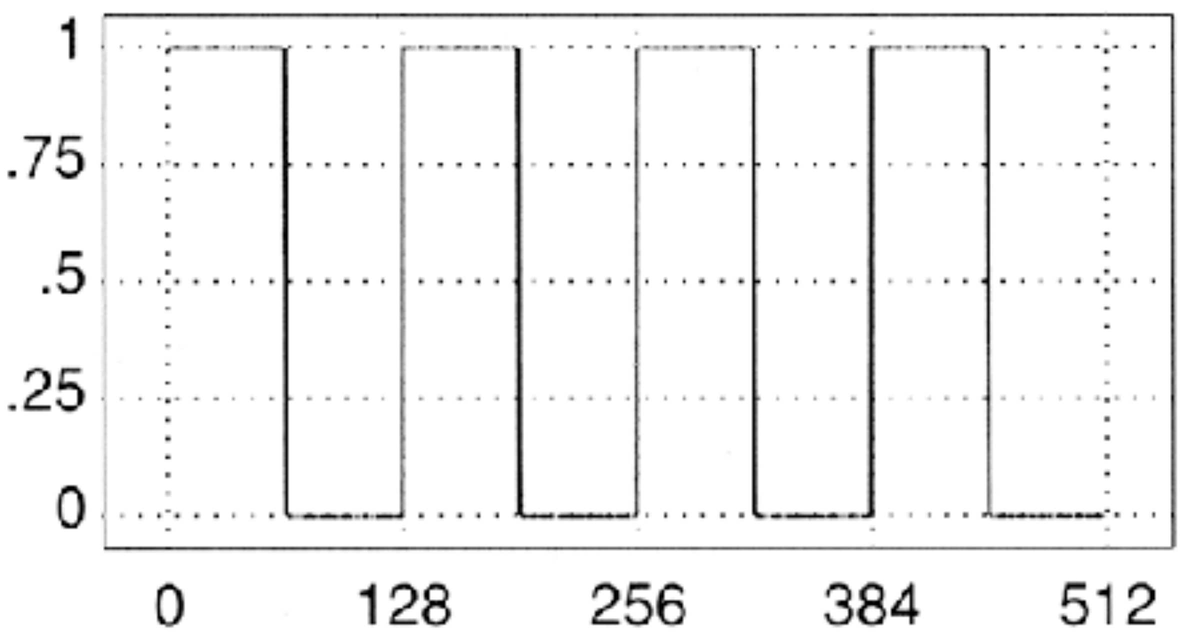


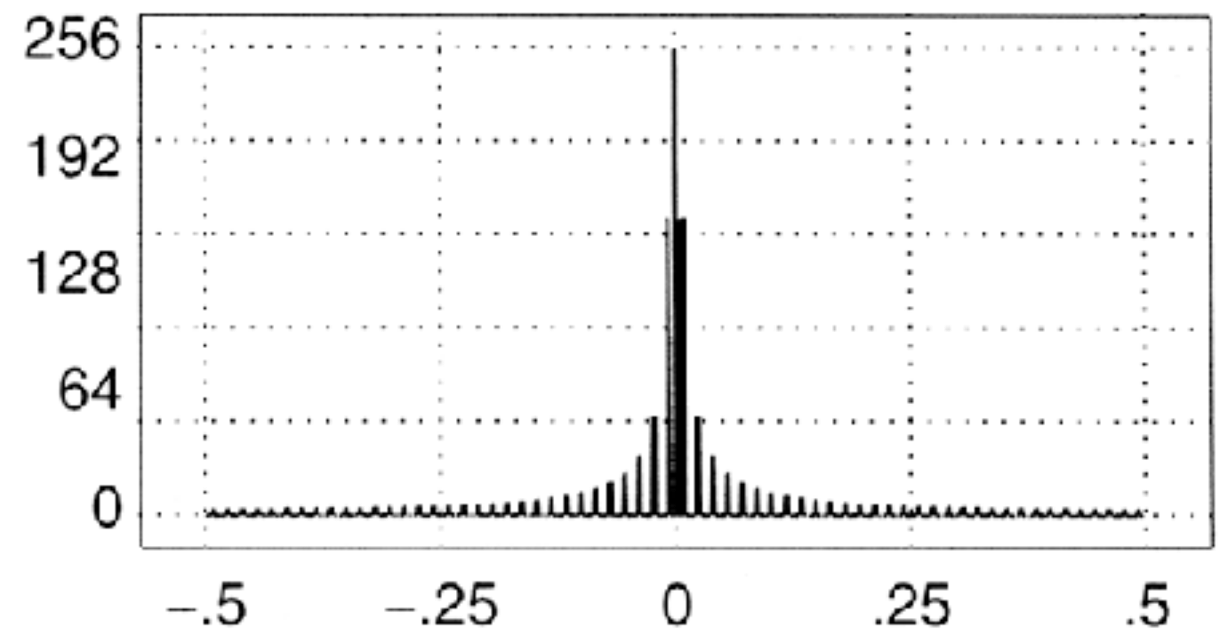
$f(x)$



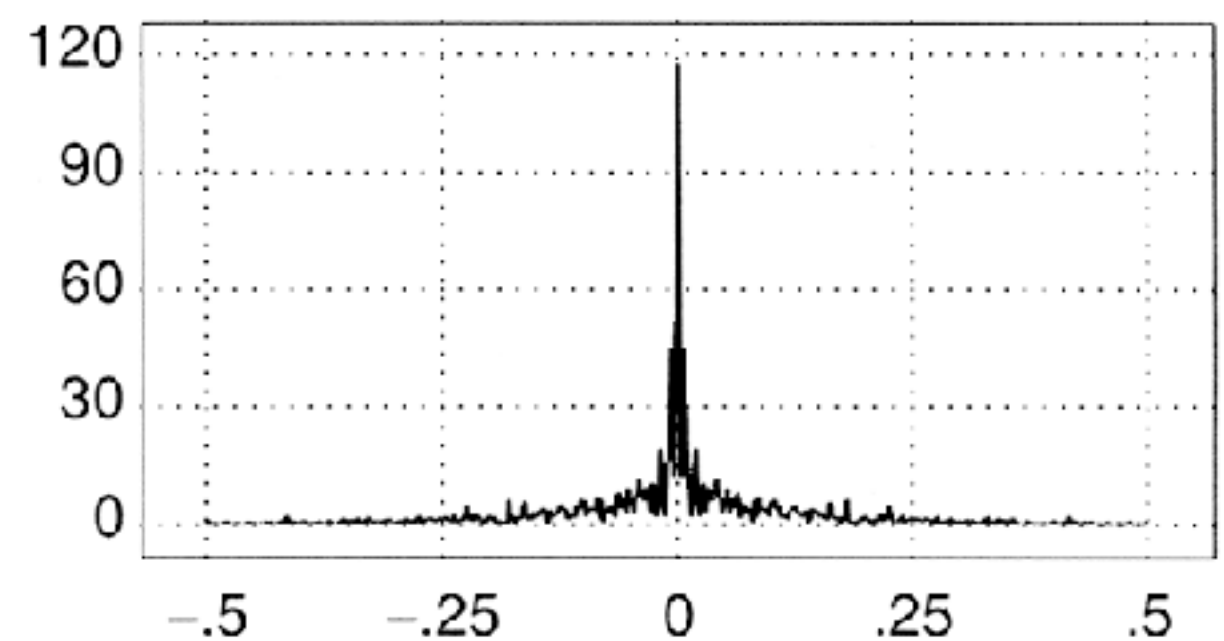
$F(u)$

Square Wave & Scanline





$f(x)$ $F(u)$



Computer Graphics Charles University

Fourier Transform

- Yields complex-valued frequency space functions
  - The imaginary part contains phase information - this is usually omitted
- Can be generalised to higher dimensions (2D, 3D)
- Alternatives, such as the Hartley transform, exist

Discrete Fourier Transform

- **For discrete signals**

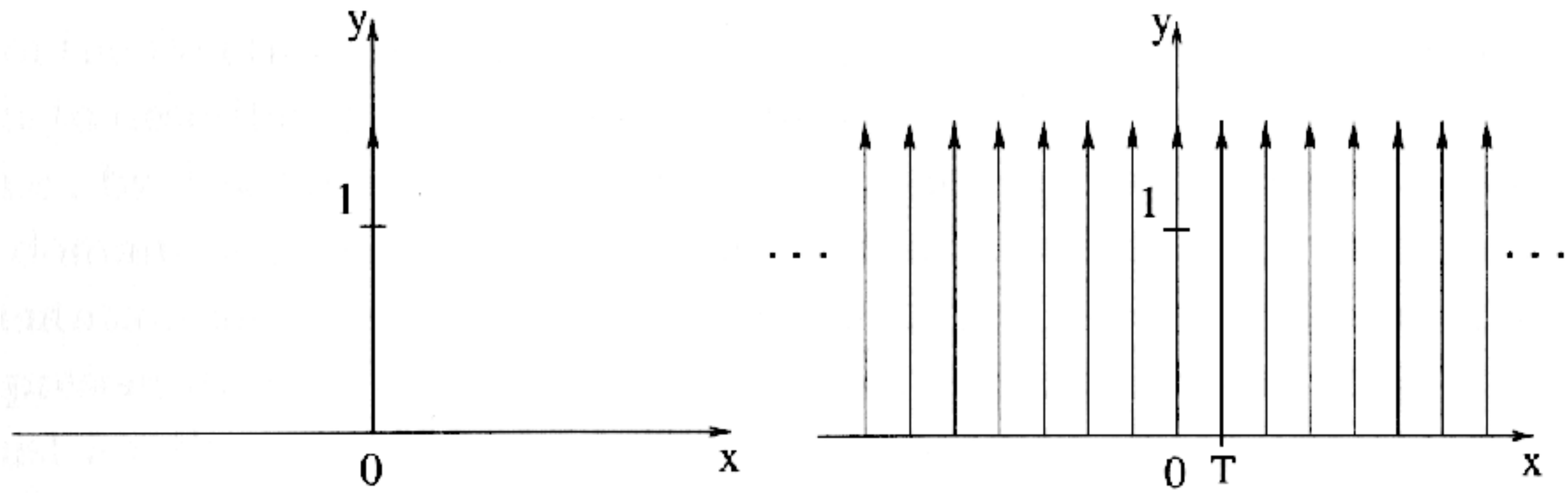
$$F(u) = \sum_{0}^{N-1} f(x)[\cos(2\pi ux/N) - i\sin(2\pi ux/N)], 0 \le u \le N-1$$

$$f(x) = \sum_{0}^{N-1} F(u)[\cos(2\pi ux/N) - i\sin(2\pi ux/N)], 0 \le x \le N-1$$

- **N samples: complexity O(N^2)**
- **Fast FT (FFT): O(N log N)**

Sampling Function: Comb

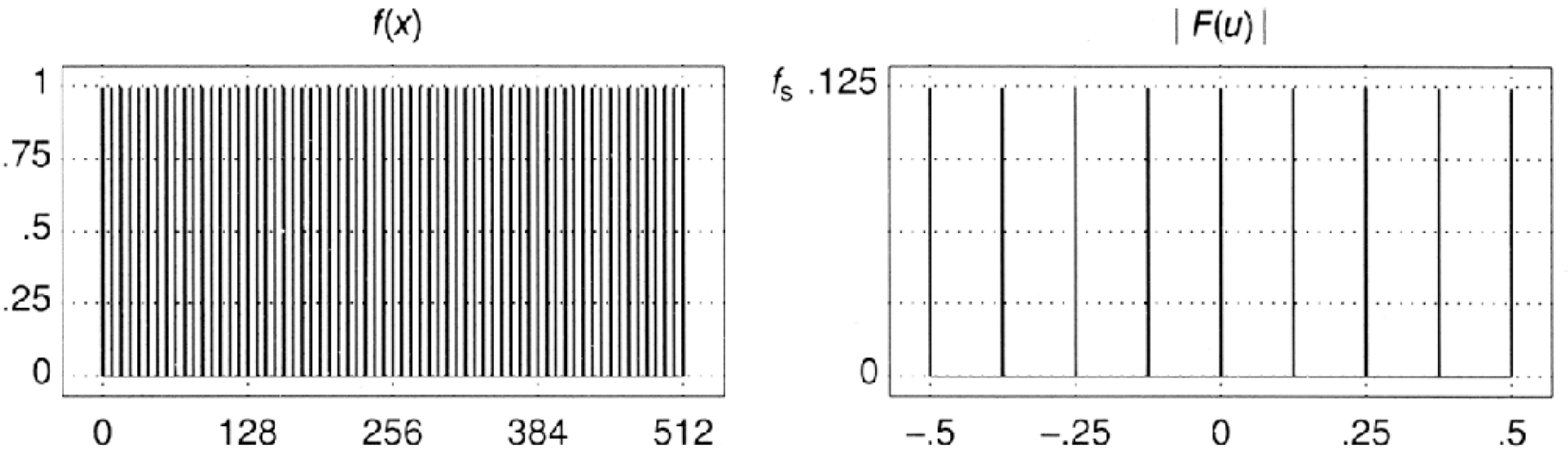
- A single Dirac pulse corresponds to sampling an image at a single location
- Regular sampling can be seen as multiplication with n evenly spaced Dirac pulses (comb function)



FT of a Comb

- **A time domain comb corresponds to a frequency domain comb with inverse pulse distance**

$$\mathrm{comb}_T(x) \equiv \mathrm{comb}_{1/T}(\omega)$$



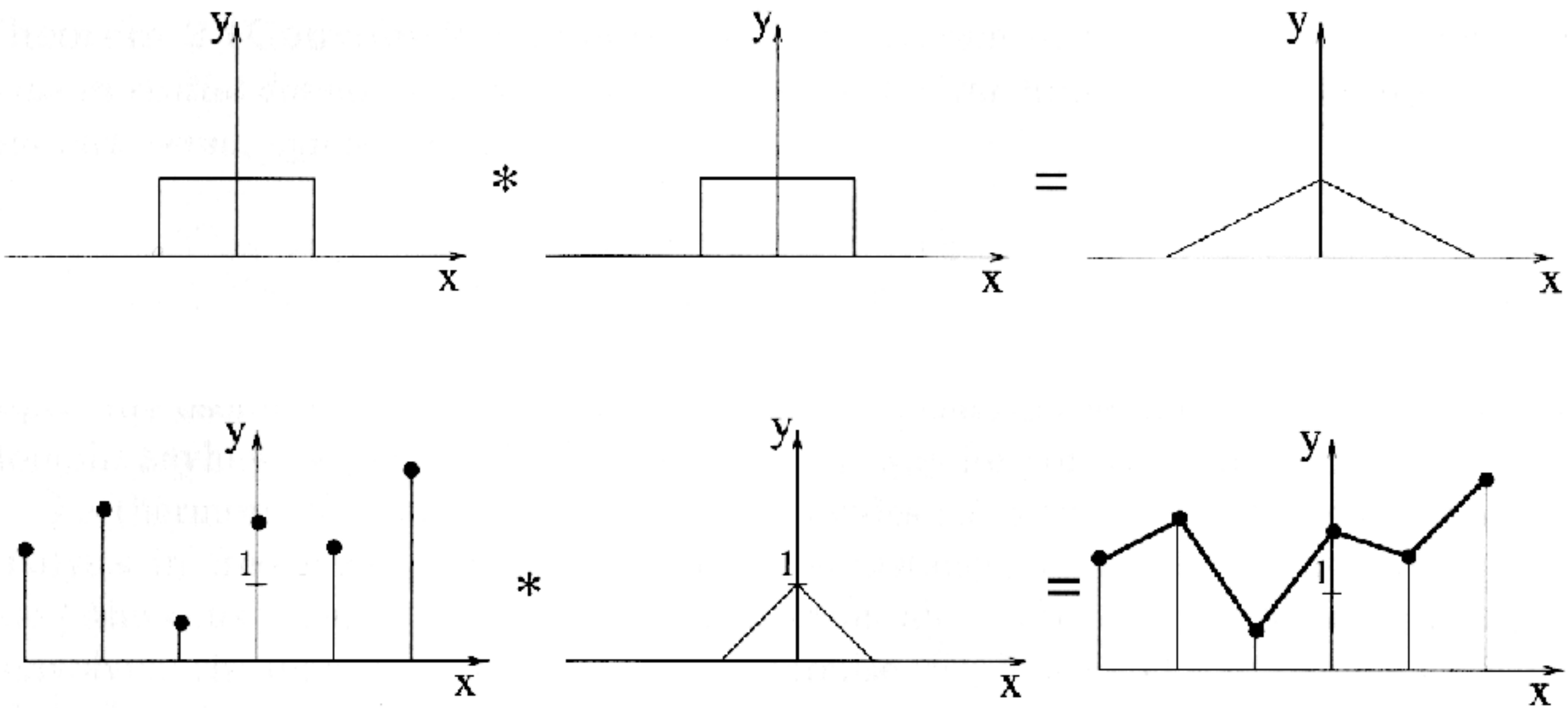
Computer Graphics Charles University

Convolution

- Mathematical operator:
  - Two functions as input
  - A new function as output
- “Weighing the first function with the second”

$$(f * g)(t) = \int_D f(\tau)g(t-\tau)d\tau$$

Convolution Examples



Convolution Theorem

- Convolution in the time domain corresponds to point wise multiplication in the frequency domain
- And vice versa

$$f * g \equiv F \cdot G$$

$$f \cdot g \equiv F * G$$

Computer Graphics Charles University

Low Pass Filter

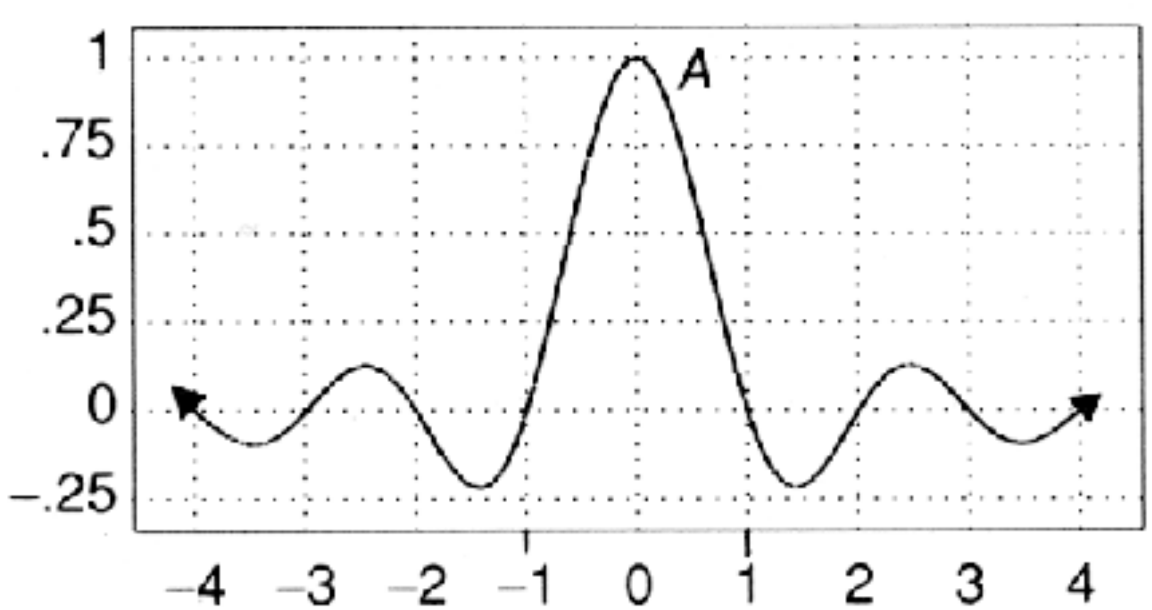
- Goal: low pass filtering of a scanline
- Time domain: convolution with a Sinc()
- Frequency domain: cutting off high frequencies - multiplication with a box function

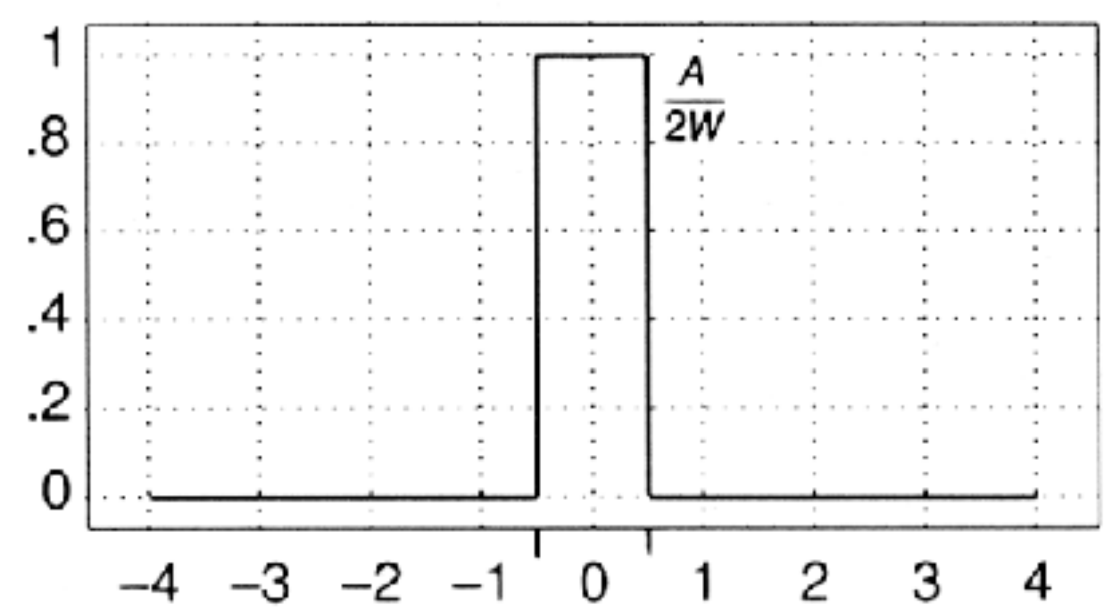
- Sinc() in the time domain corresponds to a box in the frequency domain

Sinc()

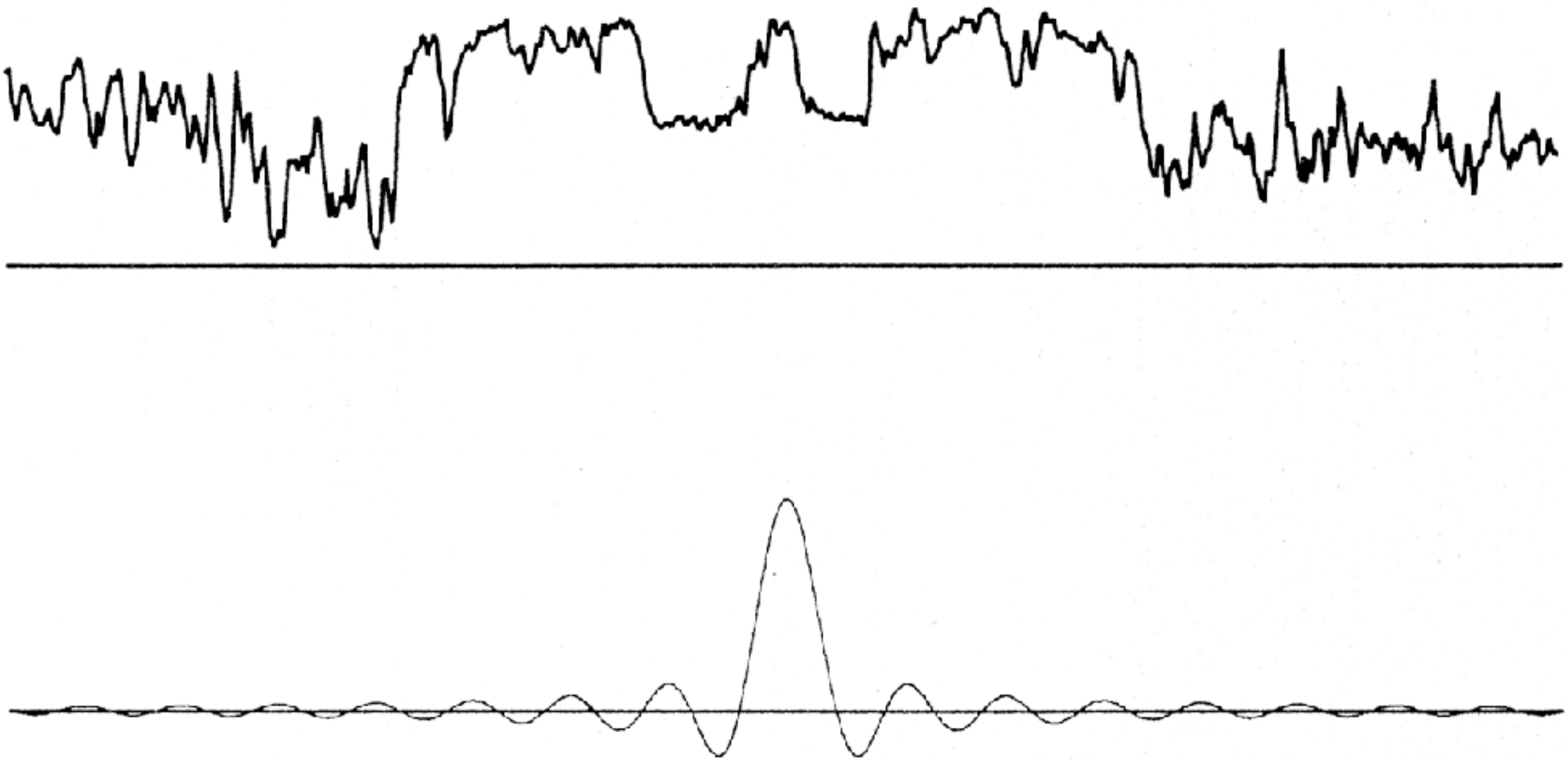
- Unlimited domain
- Perfect reconstruction filter
- Fourier transform of a box function

$$\mathrm{Sinc}(x) = \begin{cases} \dfrac{\mathrm{Sin}(x)}{x} & \mathit{falls}\ x \neq 0 \\ 0 & \mathit{falls}\ x = 0 \end{cases}$$

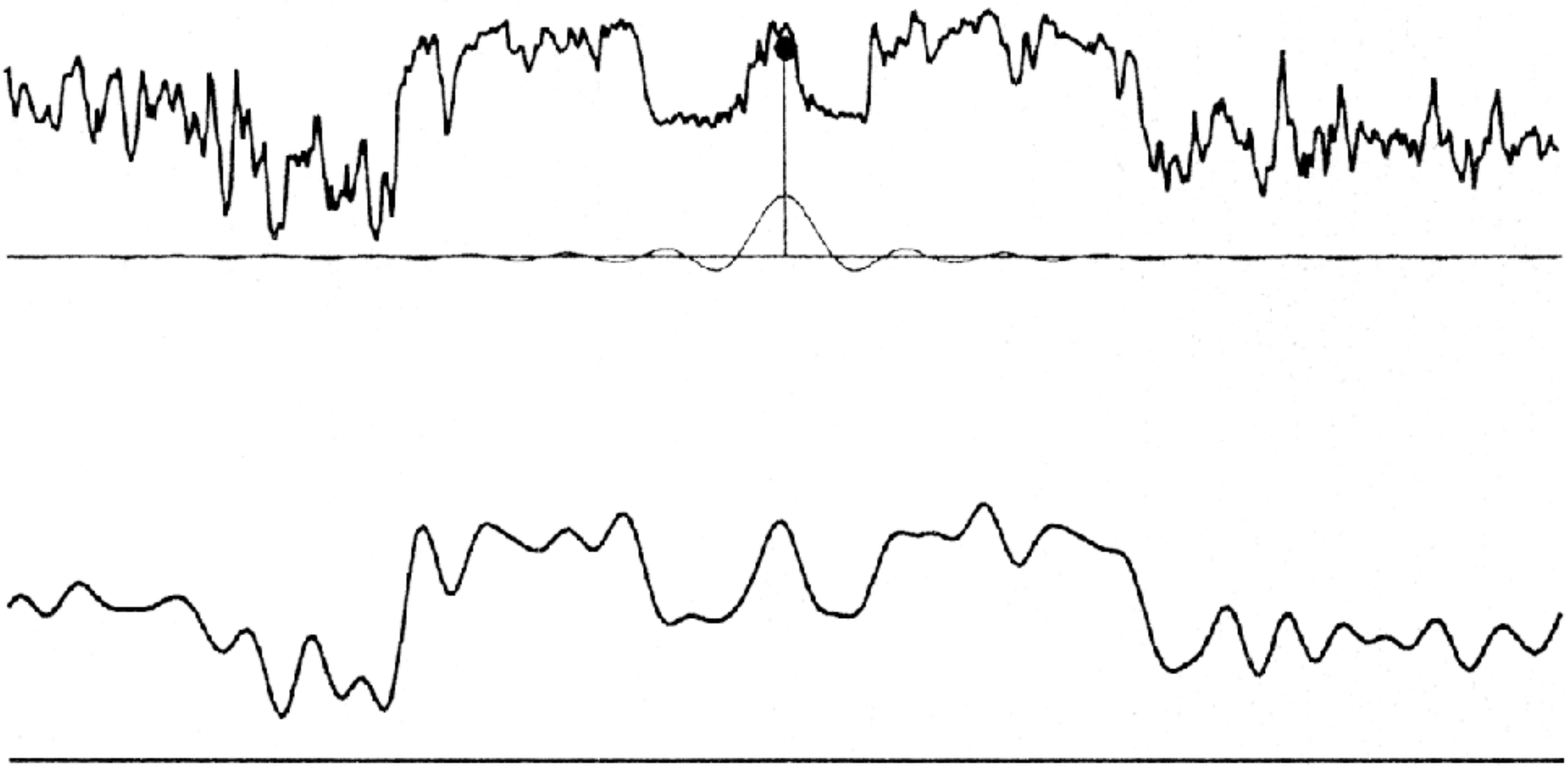




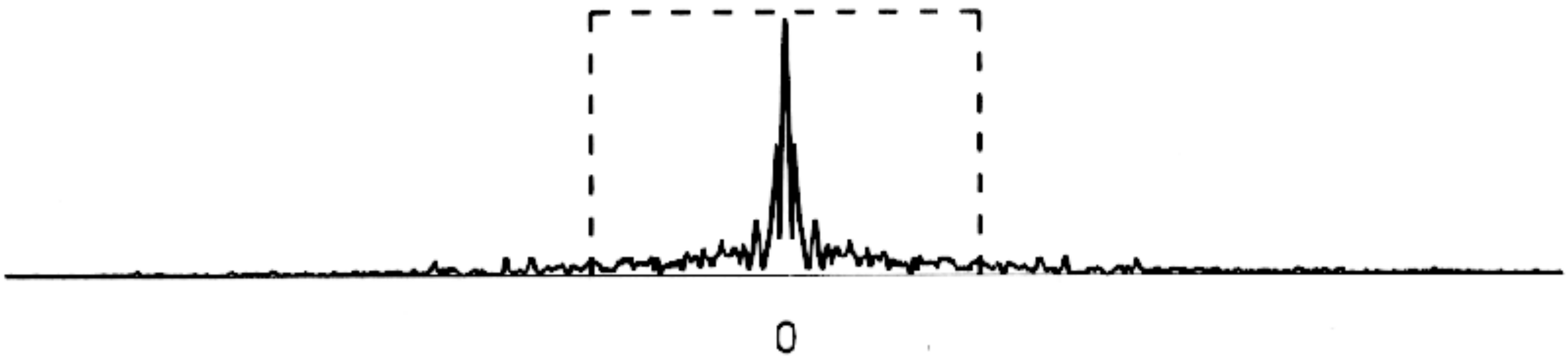
Low Pass (time domain)



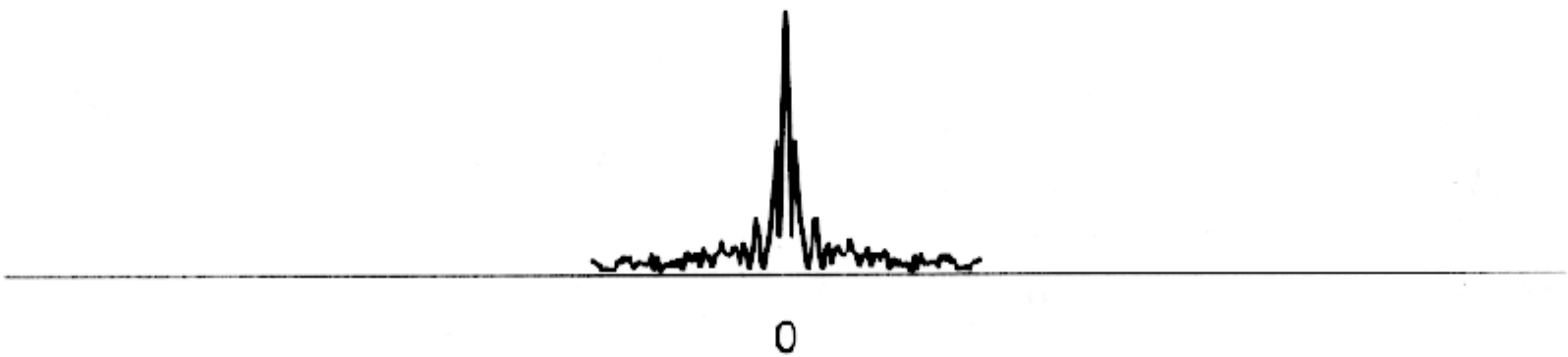
Low Pass (time domain)



Low Pass (freqency domain)



(c)



(d)

Computer Graphics Charles University

Sampling

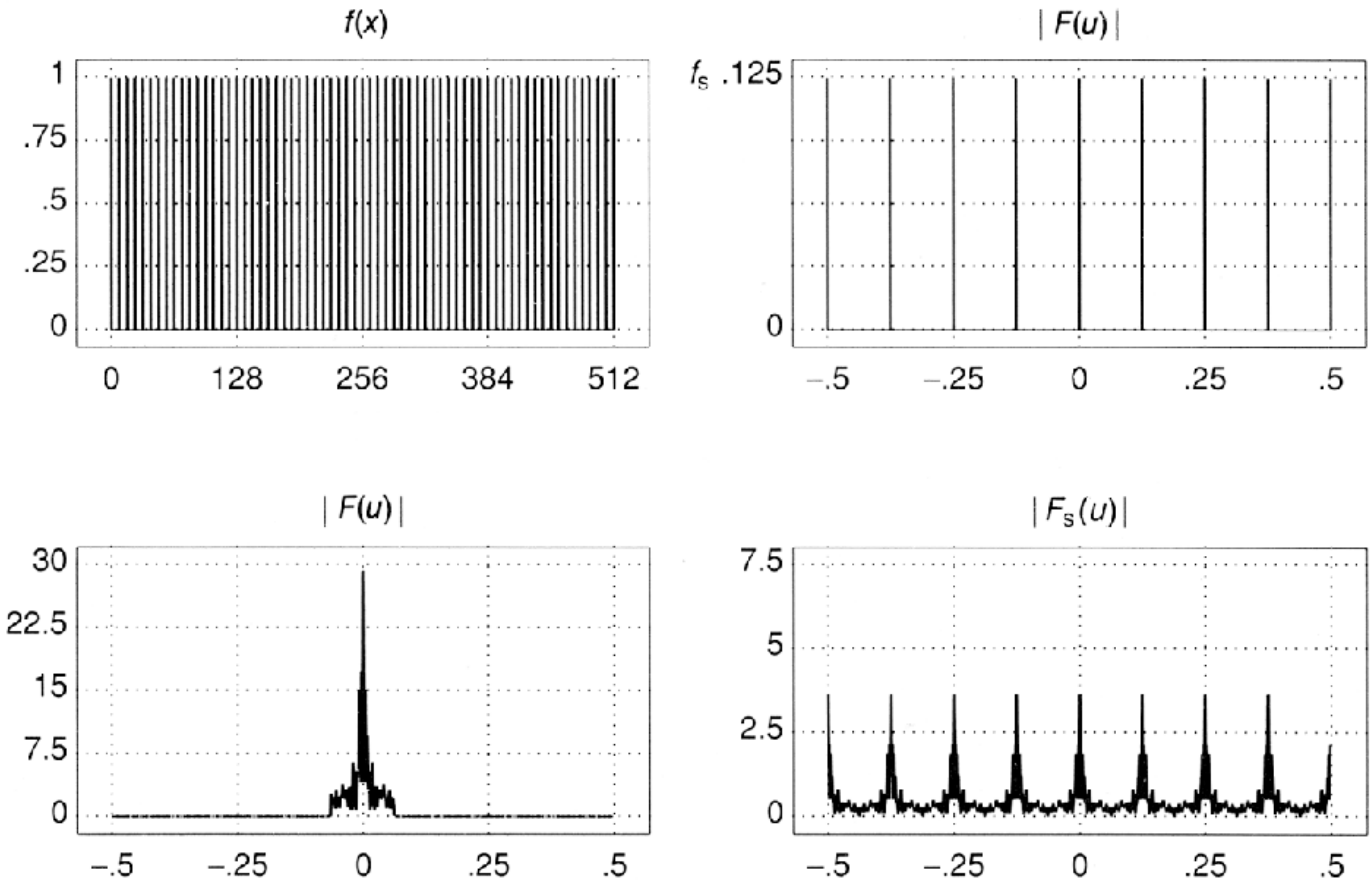
- **Sampling is a multiplication of the source signal with a comb function:**

$$f_s(x) = f(x) \cdot \mathrm{comb}_T(x)$$

- **In the frequency domain, this corresponds to a convolution (!) with an inversely spaced comb:**

$$F_s(\omega) = F(\omega) * \mathrm{comb}_{1/T}(\omega)$$

Spectrum of a Sampled Function



Perfect Reconstruction

- The copies of the original frequency spectrum must not overlap in the frequency domain
- Multiplication of the spectrum with a box function is equivalent to „cutting out" of the original spectrum
- This corresponds to a convolution with a Sinc function in the time domain

Computer Graphics Charles University

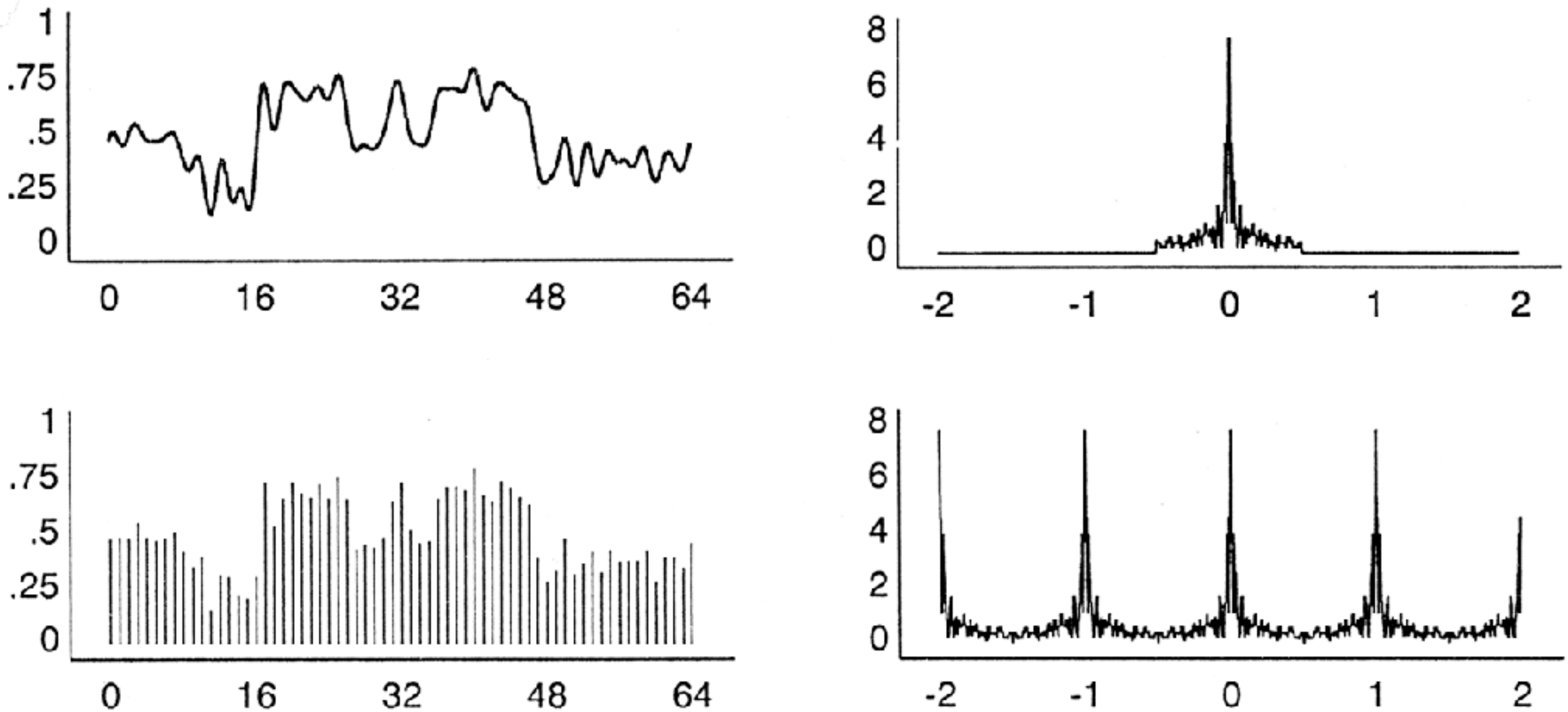
Reconstruction Examples

- Sampling and Reconstruction of a scanline

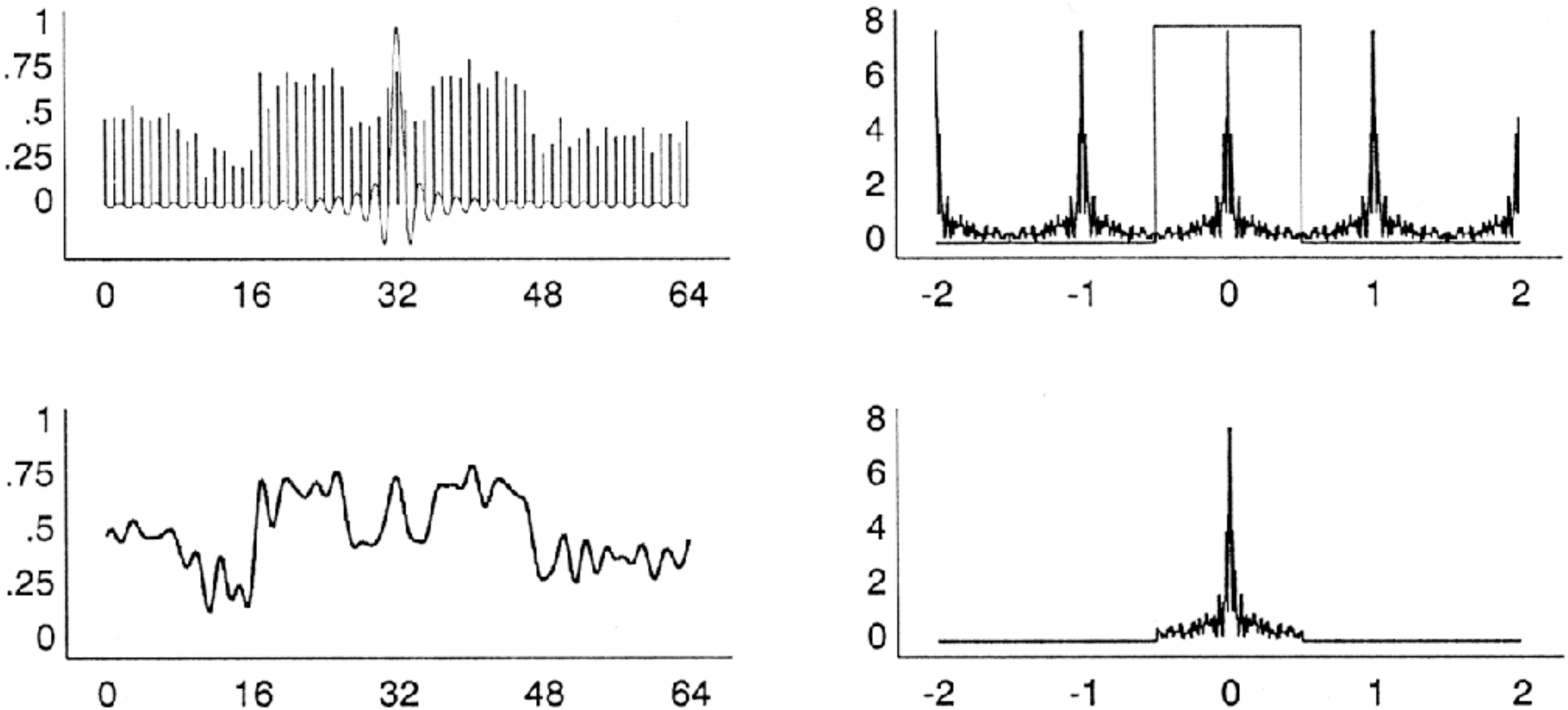
- With sufficient bandwidth
- With insufficient bandwidth
- With band limiting

- With sinc() and tent kernels

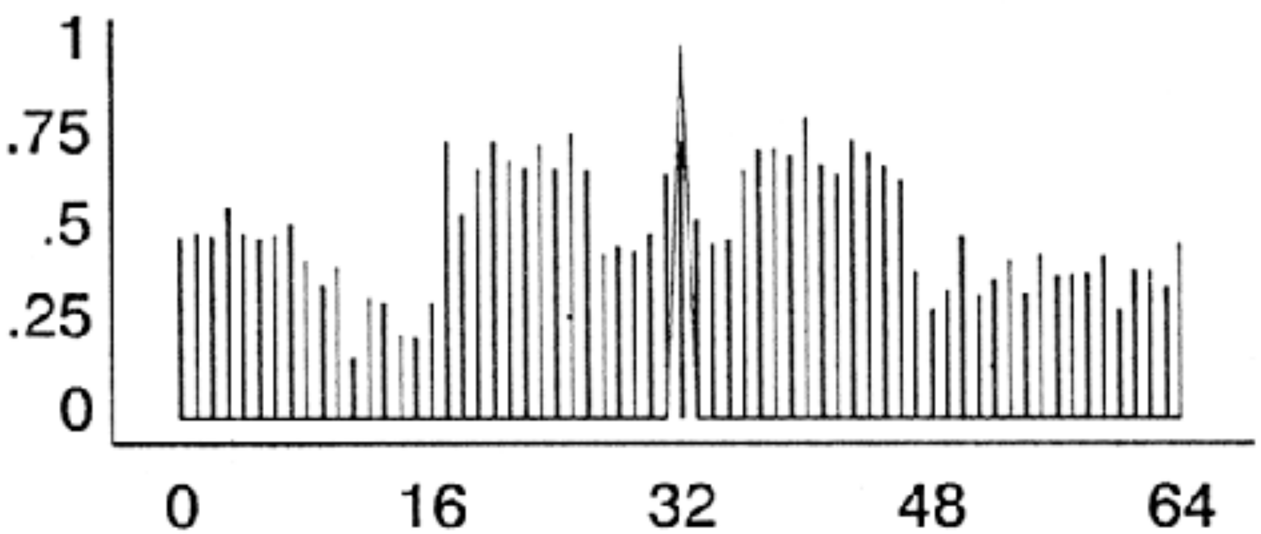
Sampling > w

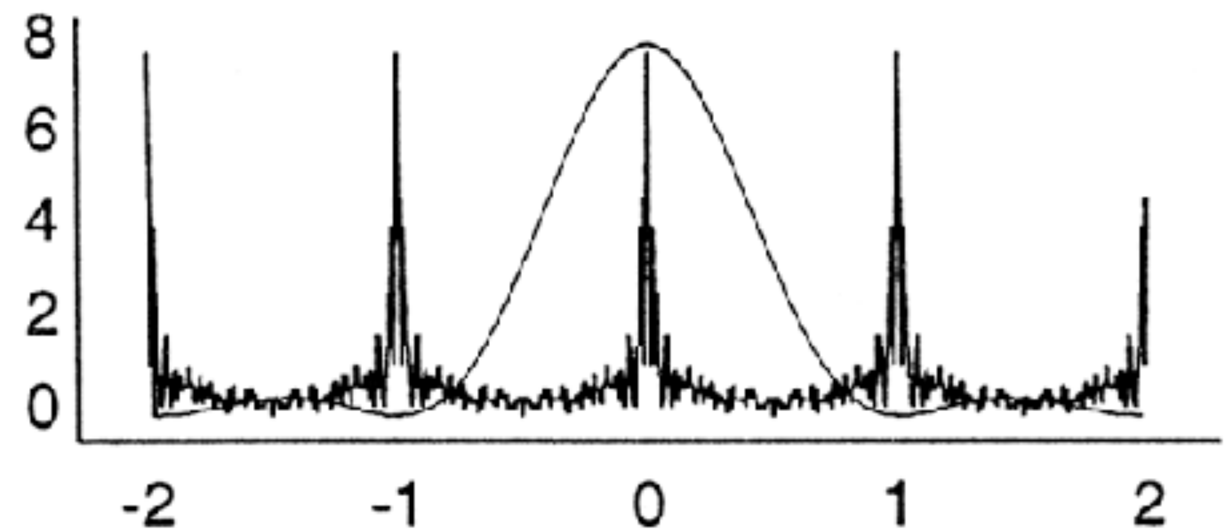


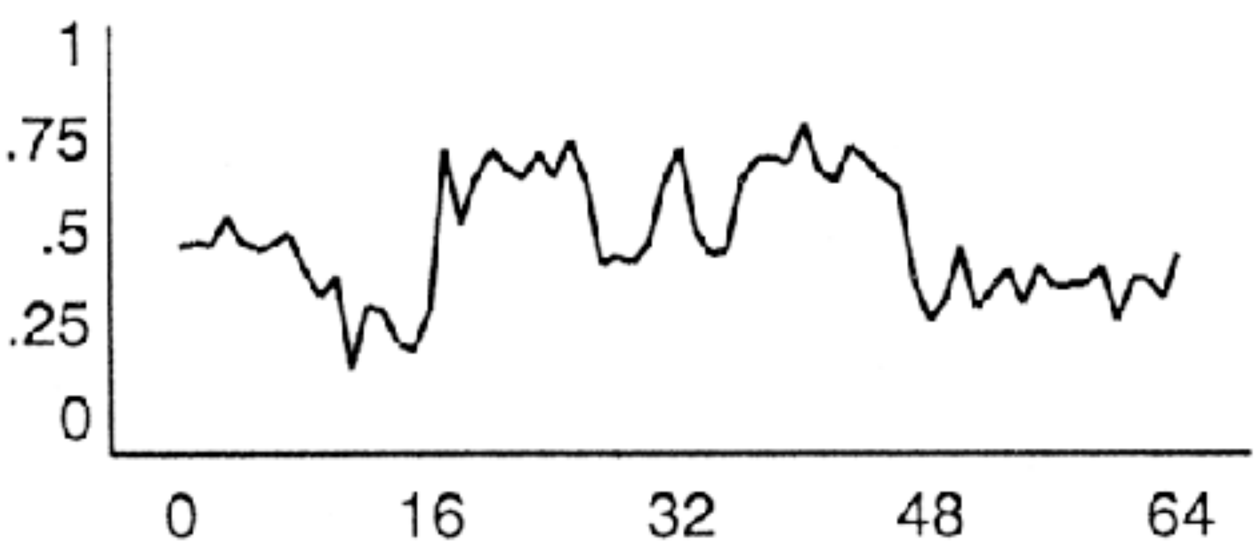
Sampling > w: Sinc() kernel

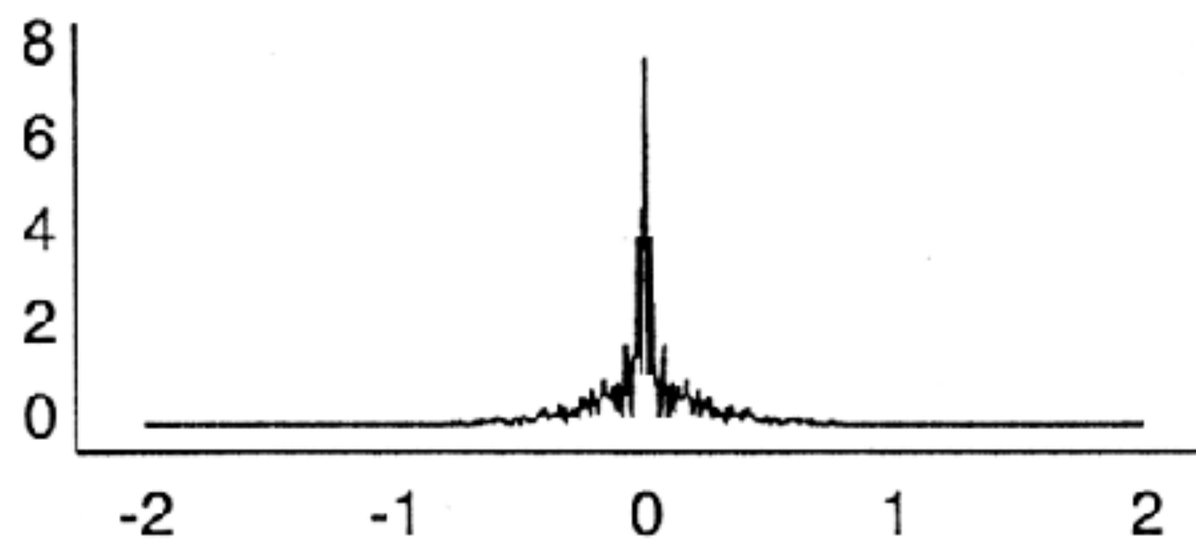


Sampling > w: Tent Kernel

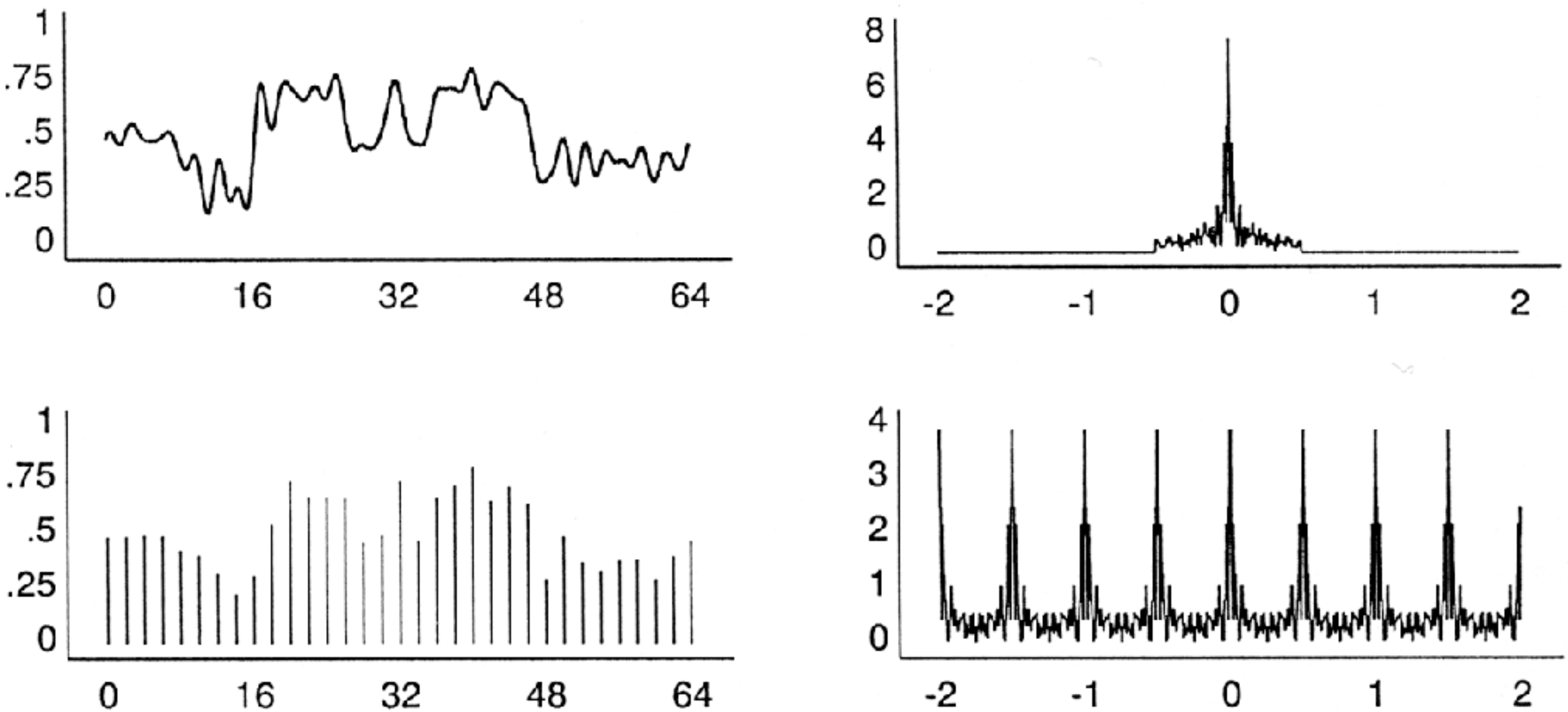




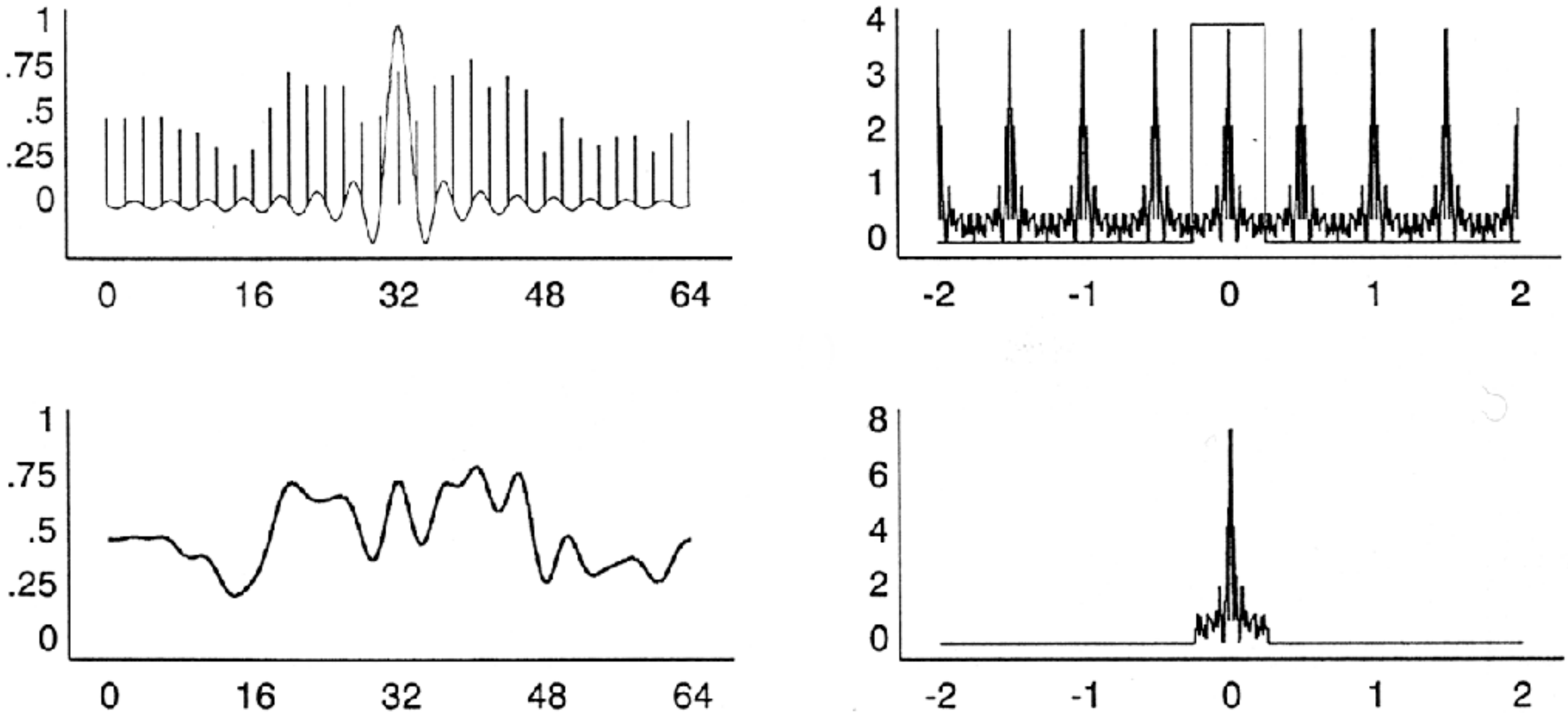




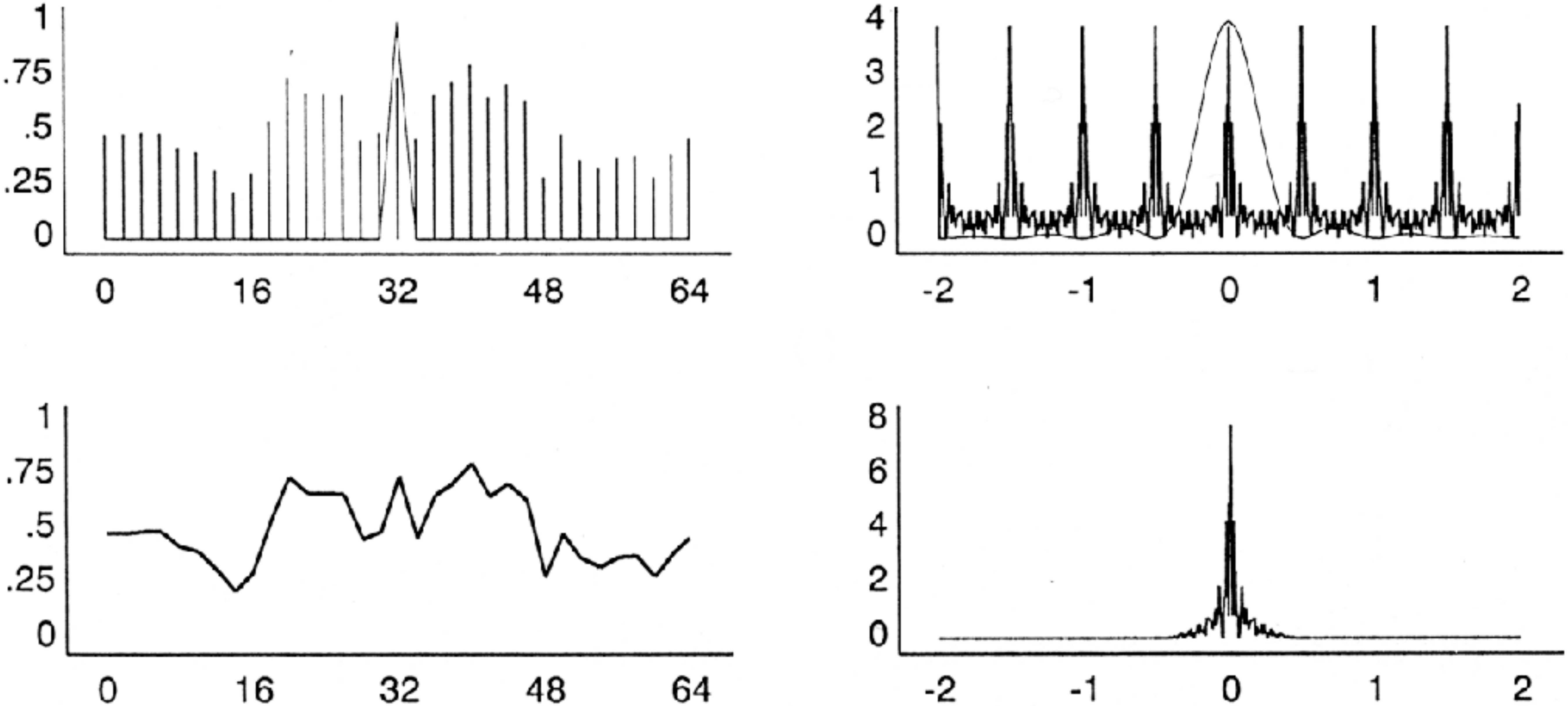
Sampling < w


1
.75
.5
.25
0
0
16
32
48
64
8
6
4
2
0
-2
-1
0
1
2
1
.75
.5
.25
0
0
16
32
48
64
4
3
2
1
0
-2
-1
0
1
2

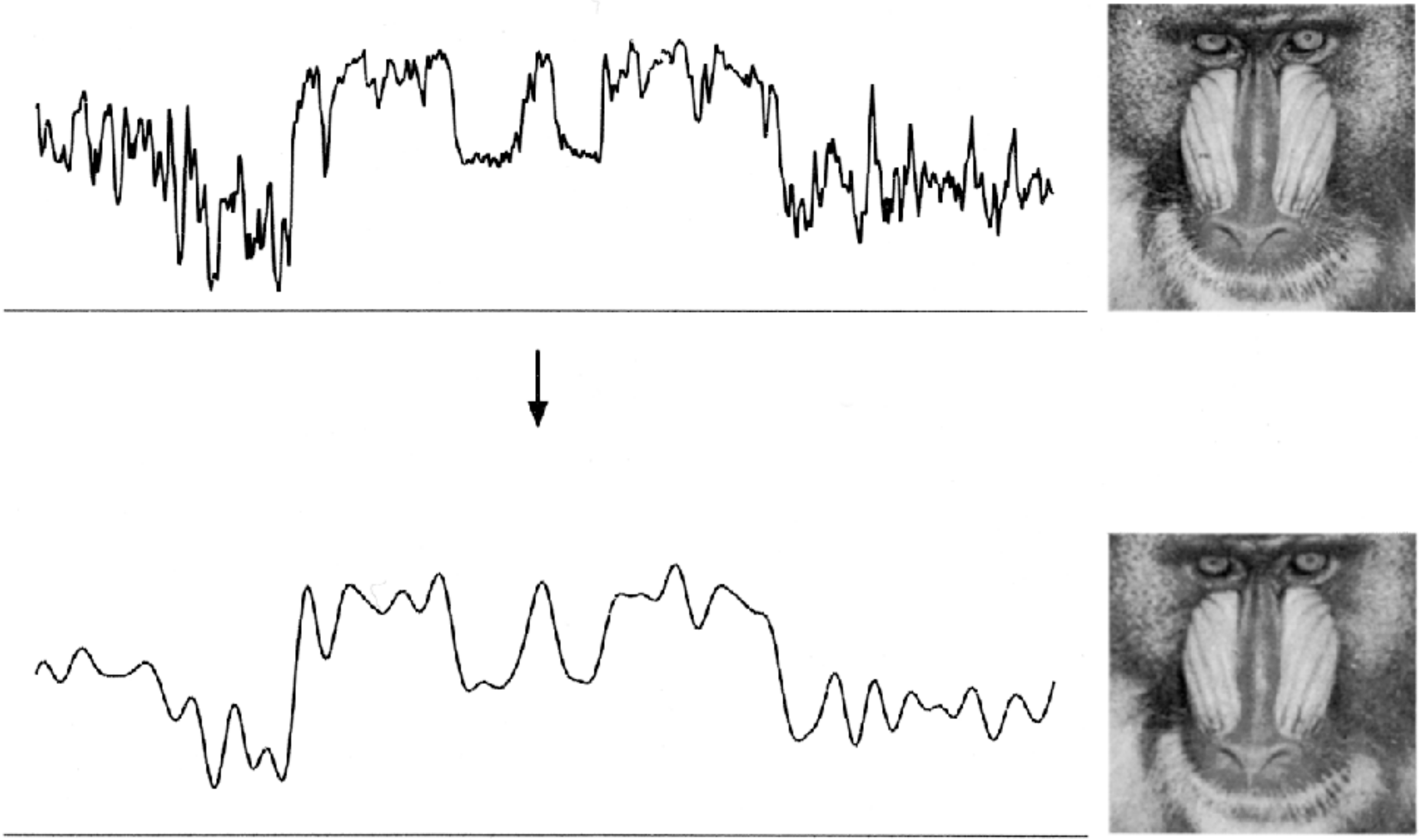
Sampling < w: Sinc()



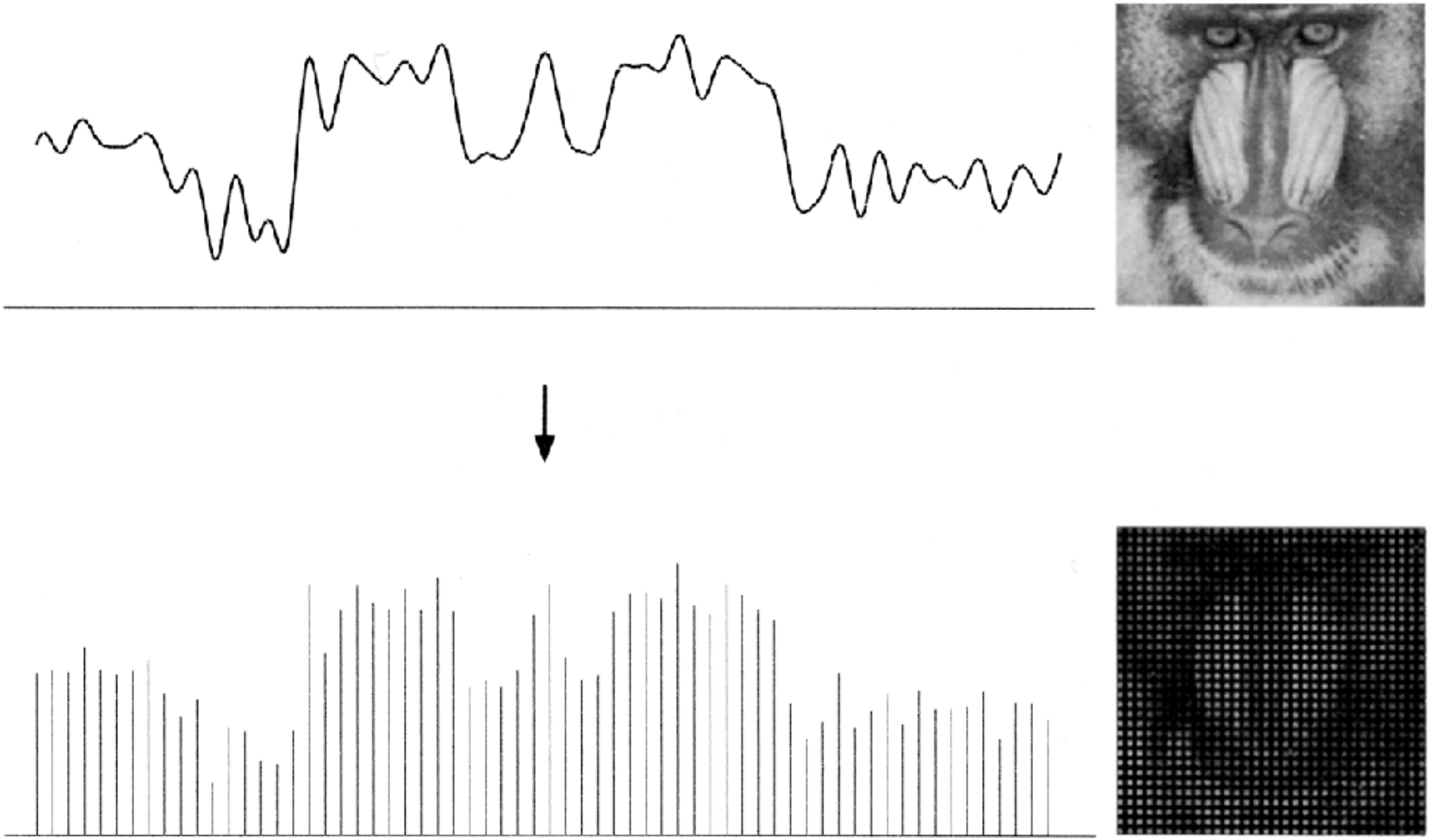
Sampling < w: Tent Kernel



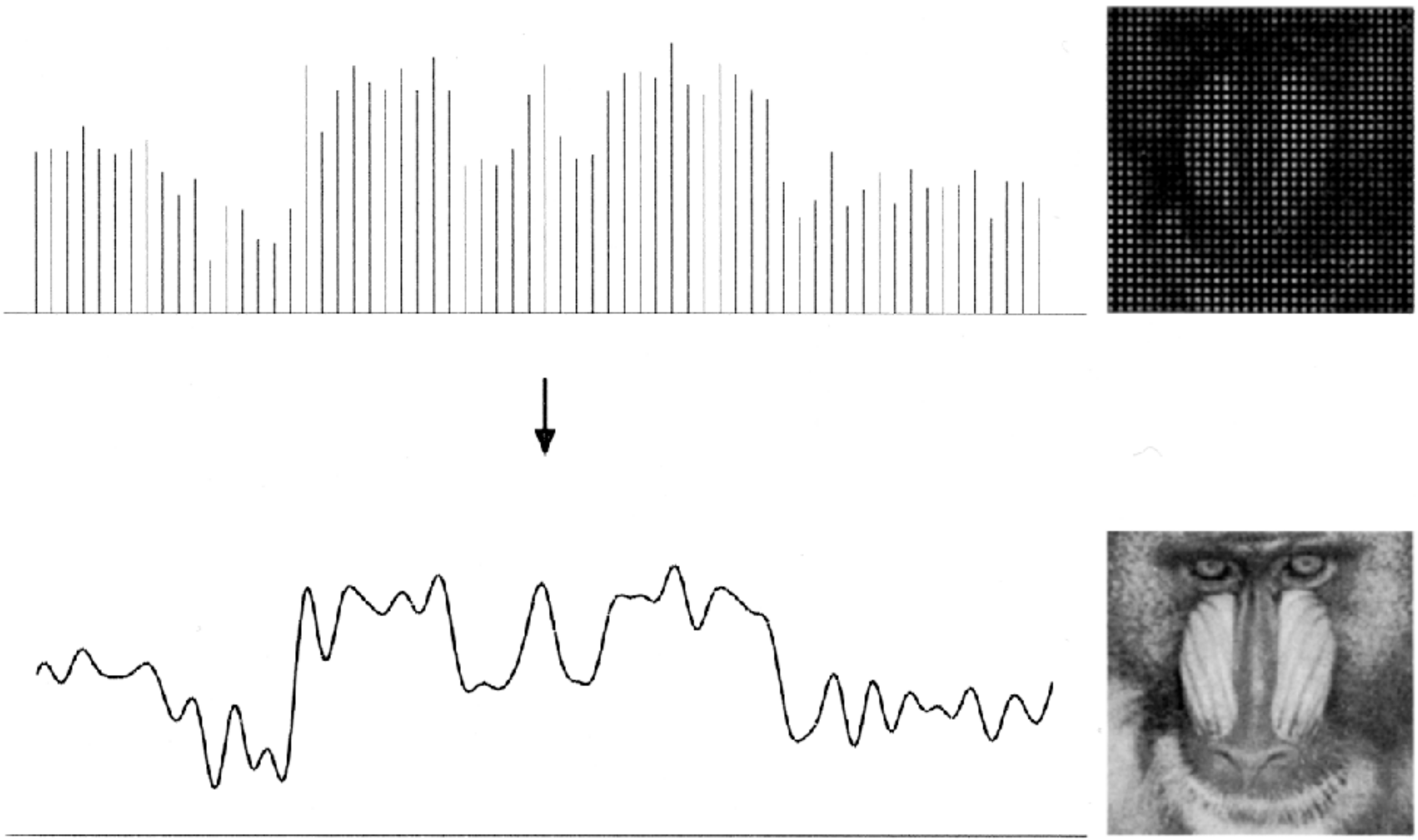
Band Limiting the Signal



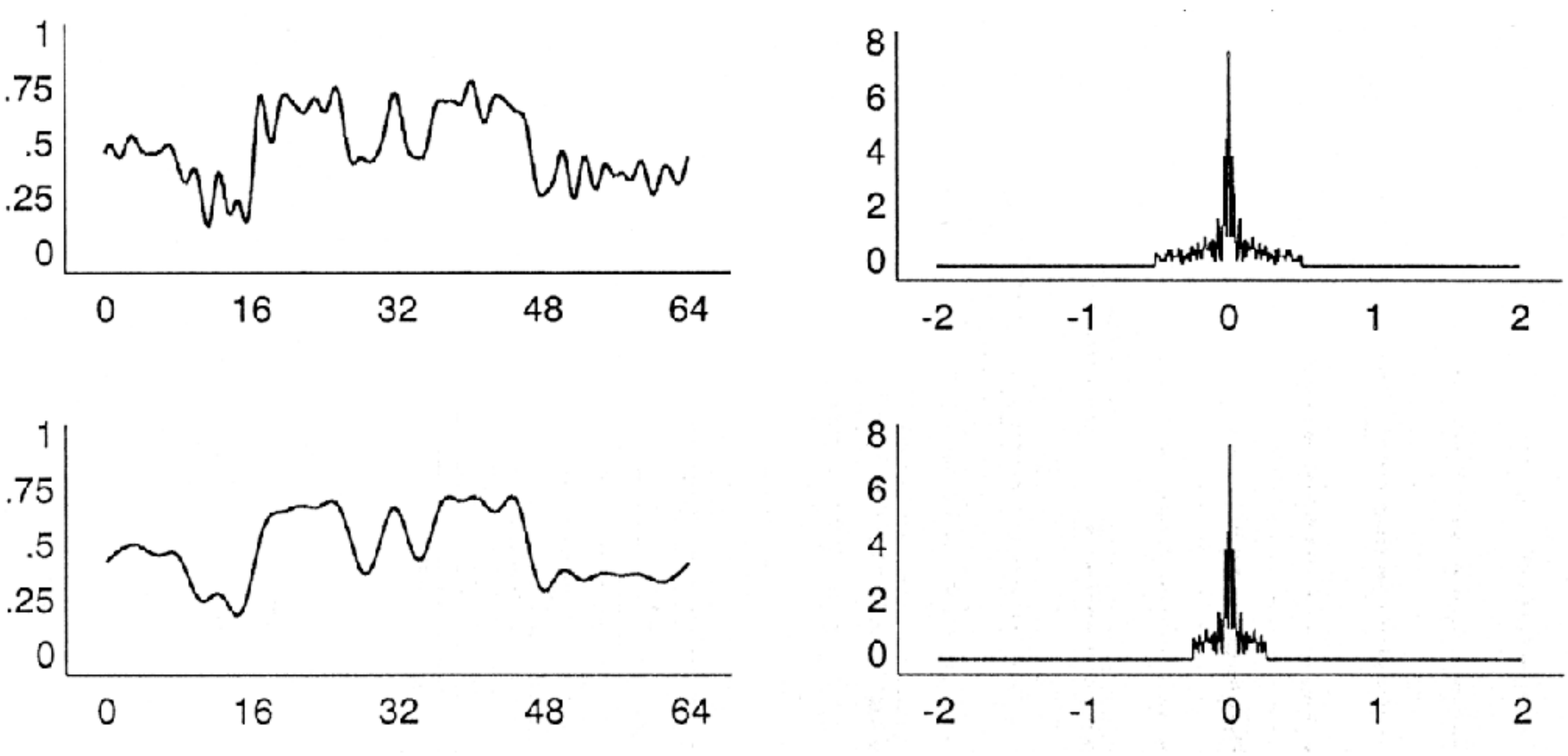
Band Limiting the Signal



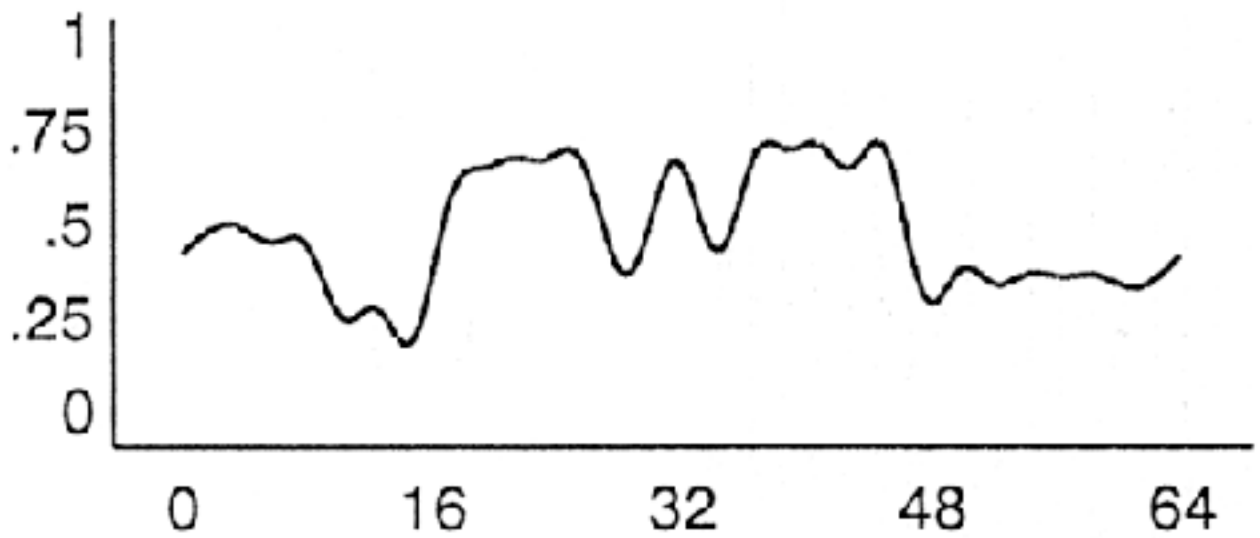
Band Limiting the Signal

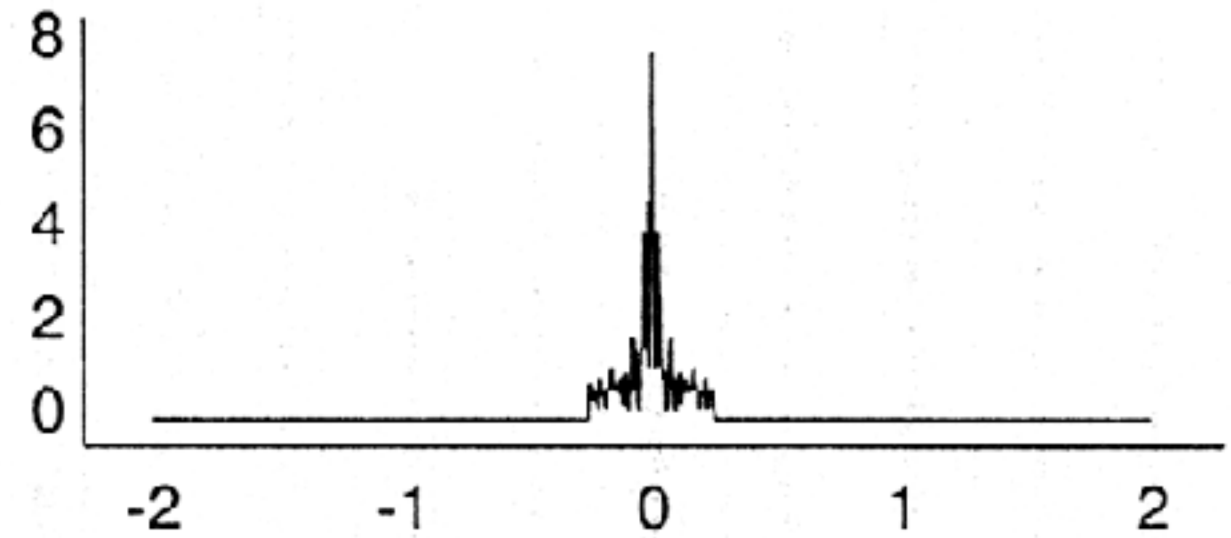


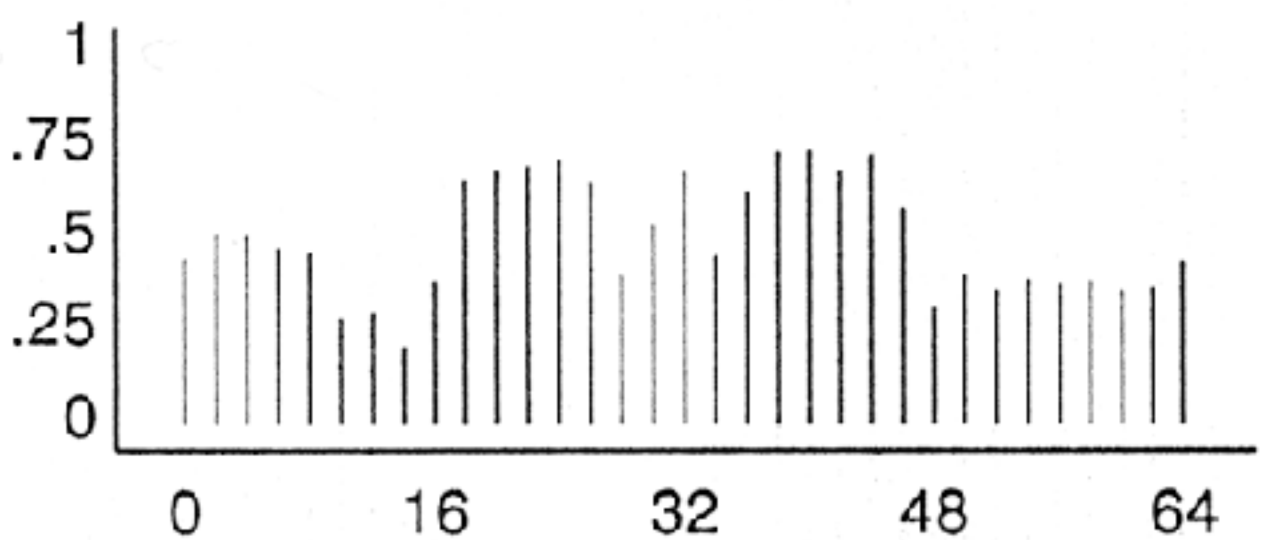
Band Limiting

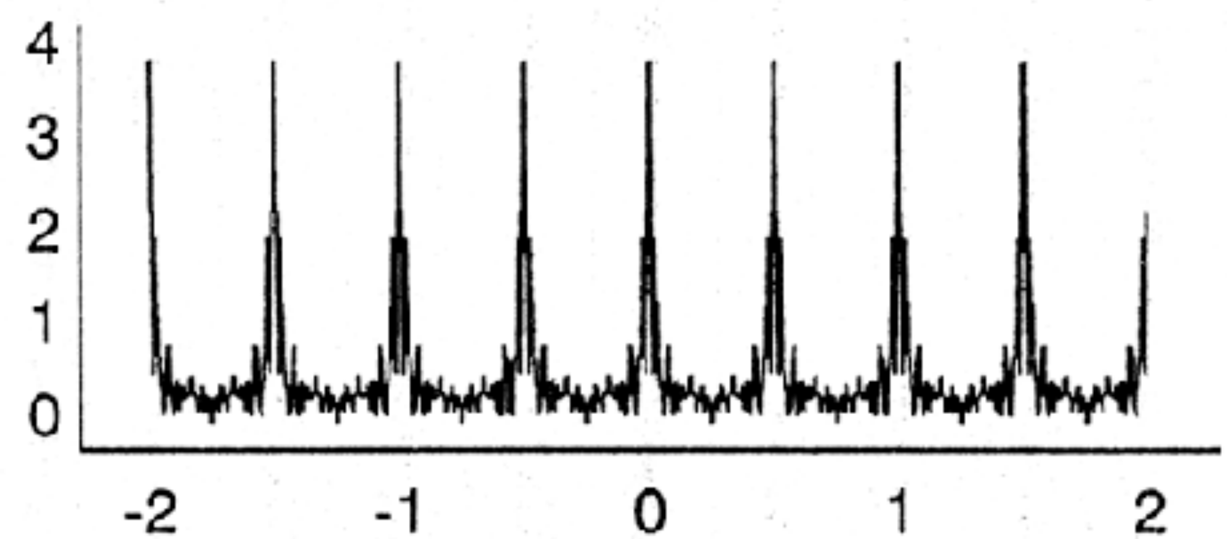


Sampling

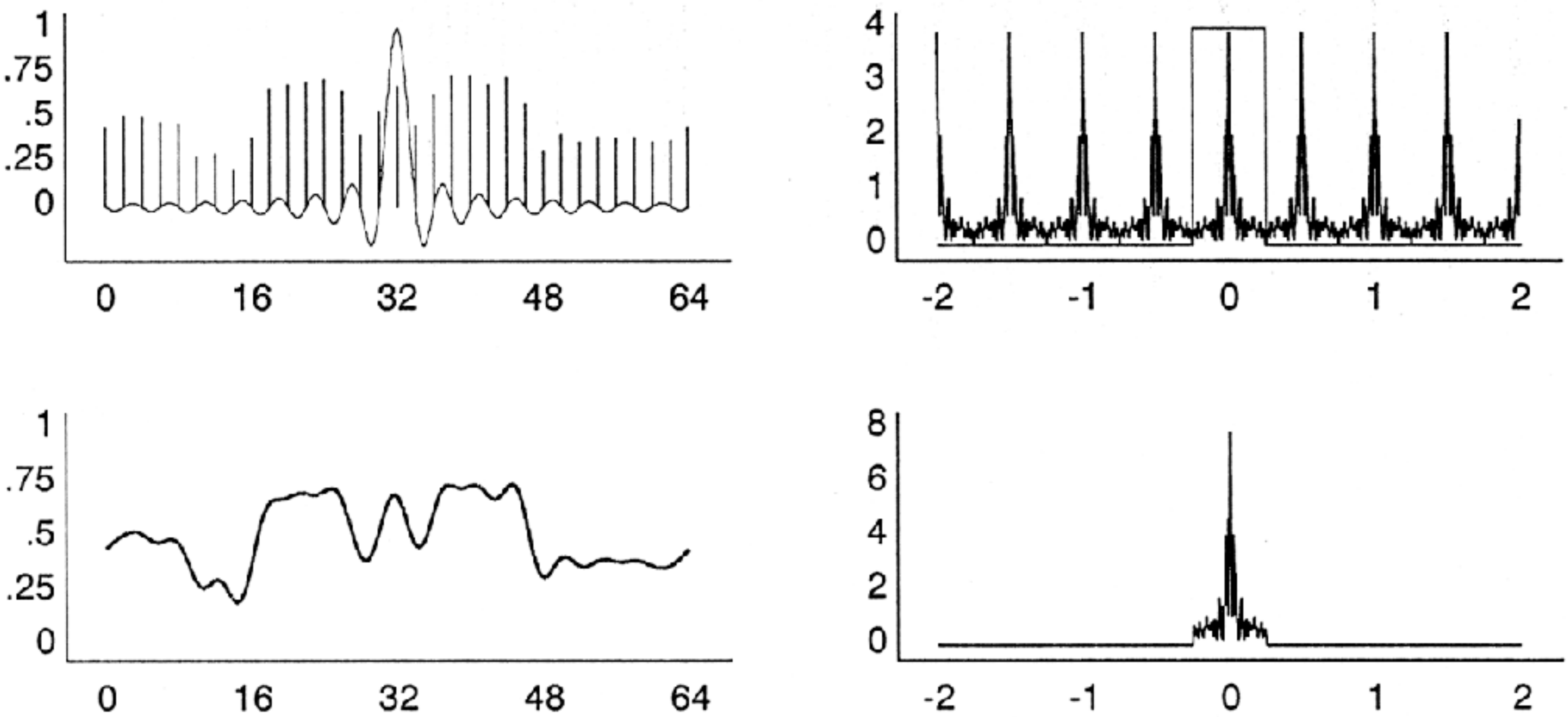

1
.75
.5
.25
0
0
16
32
48
64


8
6
4
2
0
-2
-1
0
1
2


1
.75
.5
.25
0
0
16
32
48
64


4
3
2
1
0
-2
-1
0
1
2

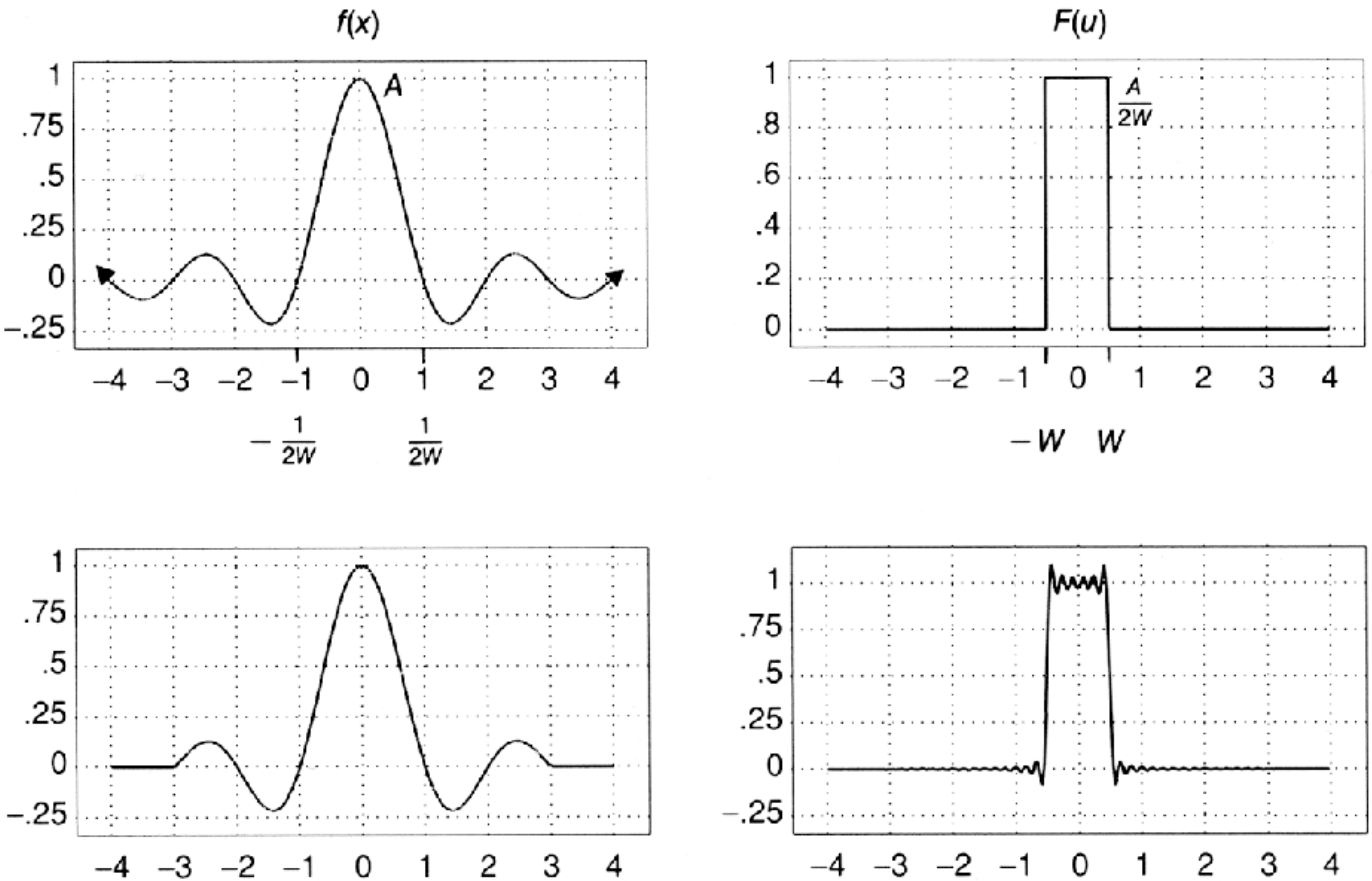
Reconstruction



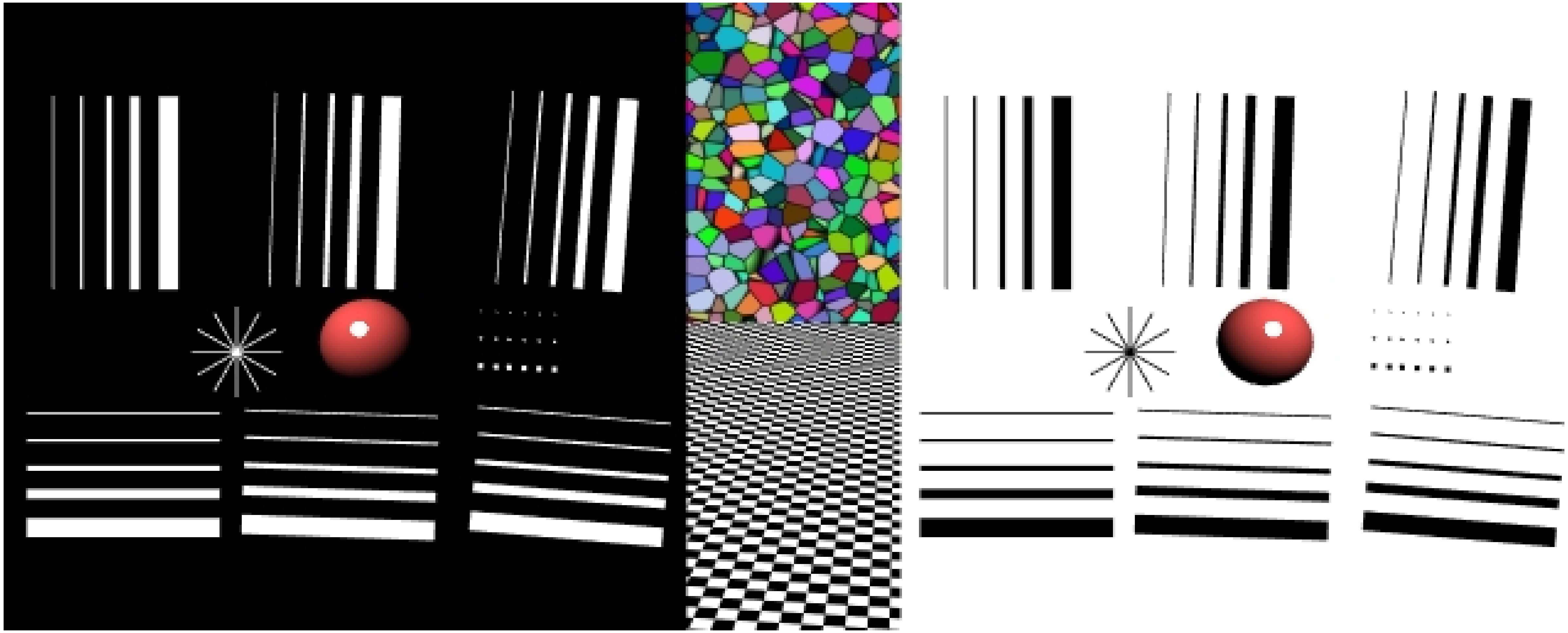
Reconstruction in Practice

- An ideal reconstruction filter exists:
  Sinc(x)
- Problem: Sinc(x) is unusable in practice, as it has infinite extent
- And to add insult to injury, truncated Sinc(x) is actually worse than many alternatives
- In practice, one uses a number of sub-optimal filters, depending on application

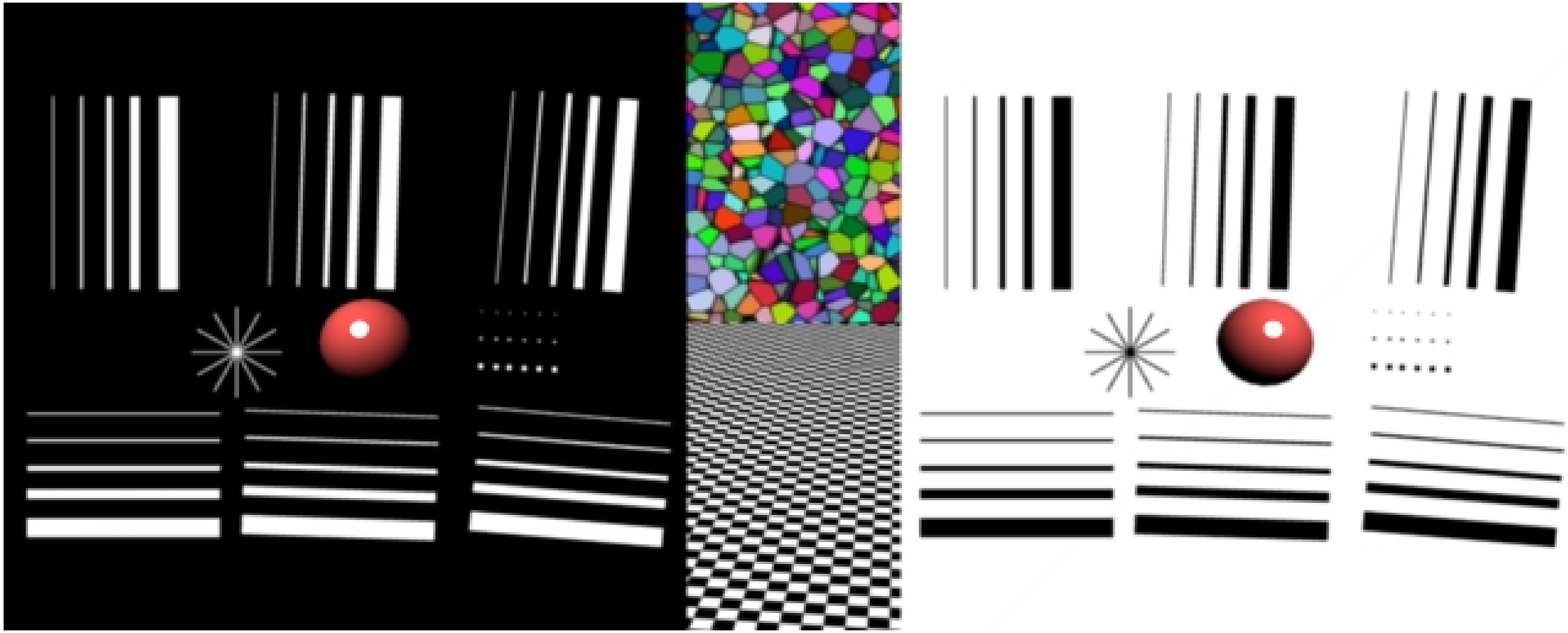
Sinc() & cut-off Sinc()



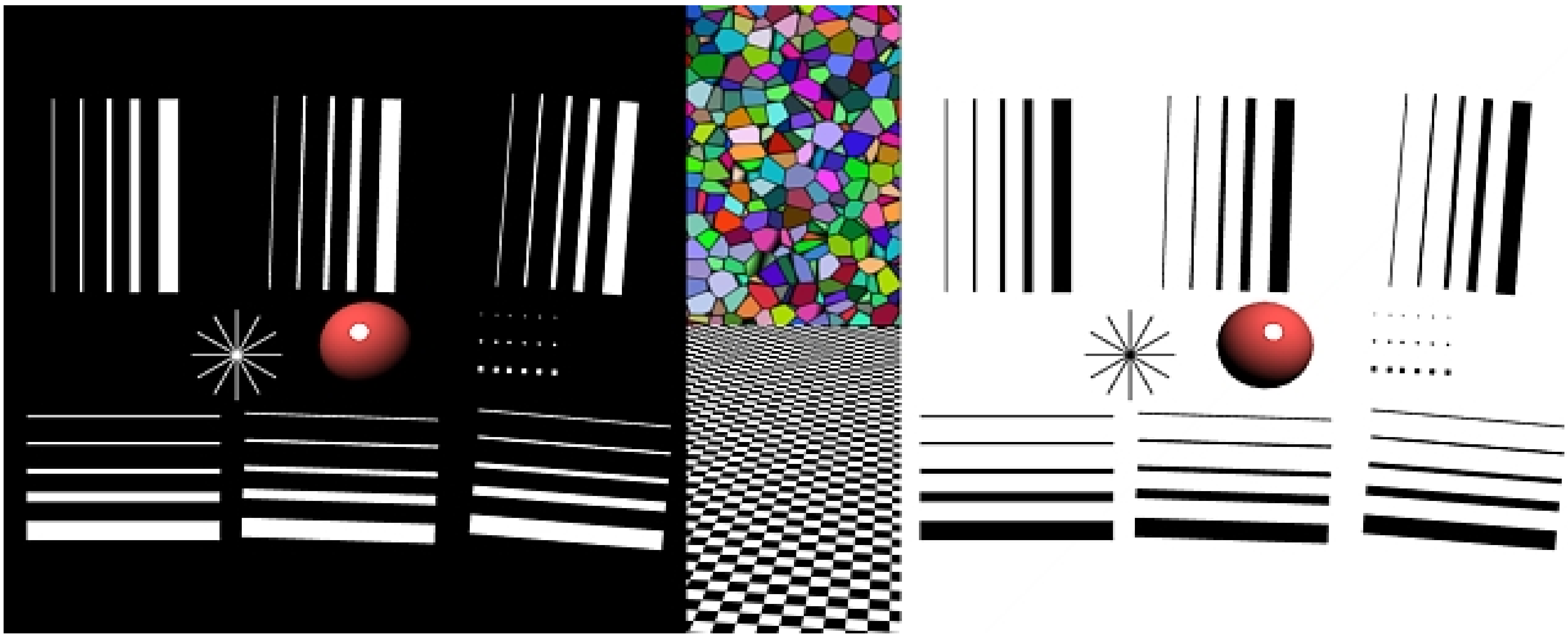
Blender



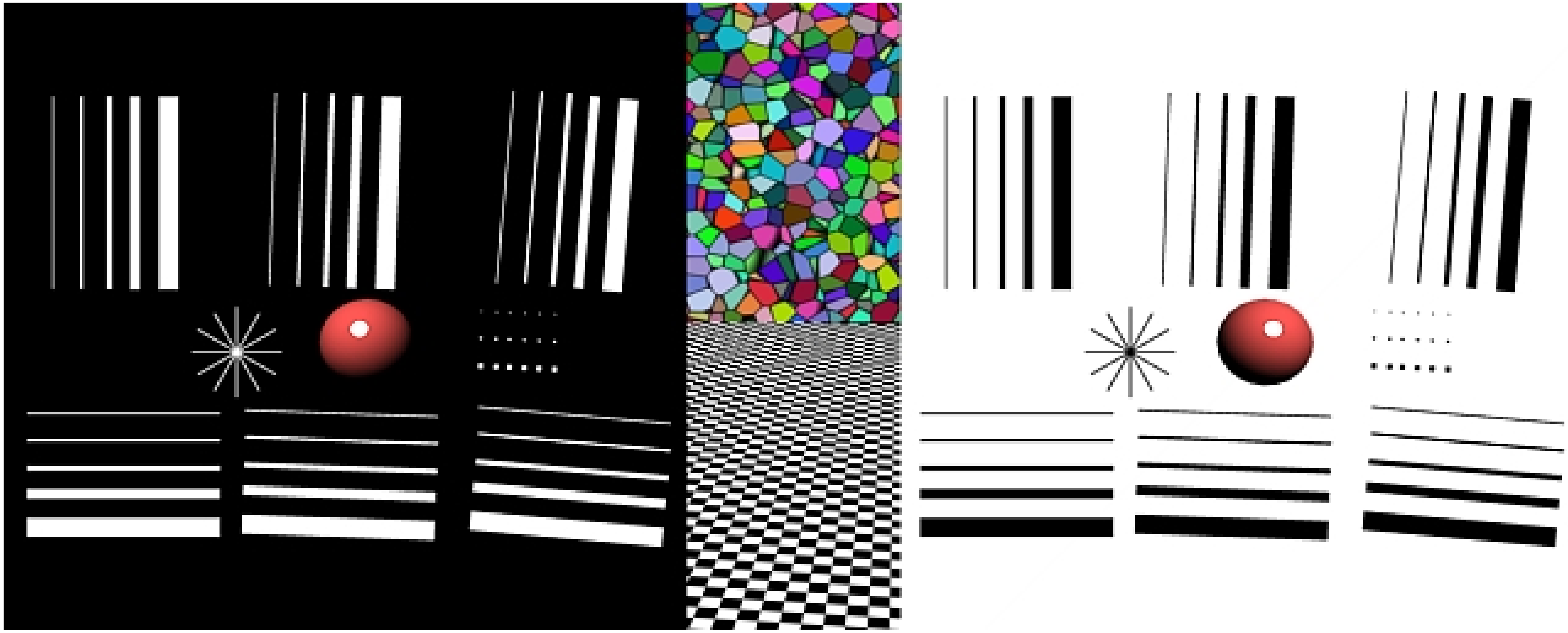
Blender



Blender

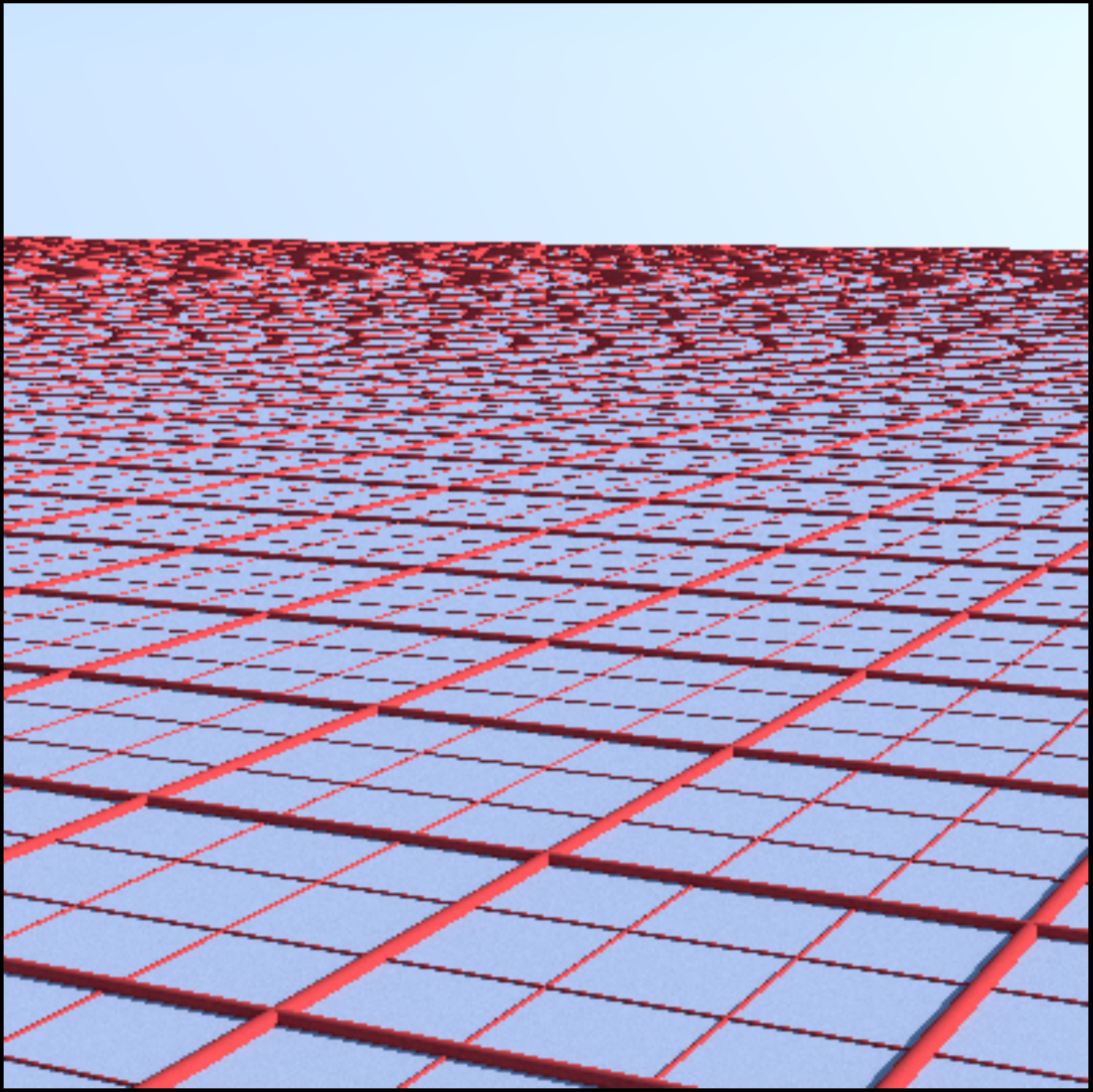


Blender



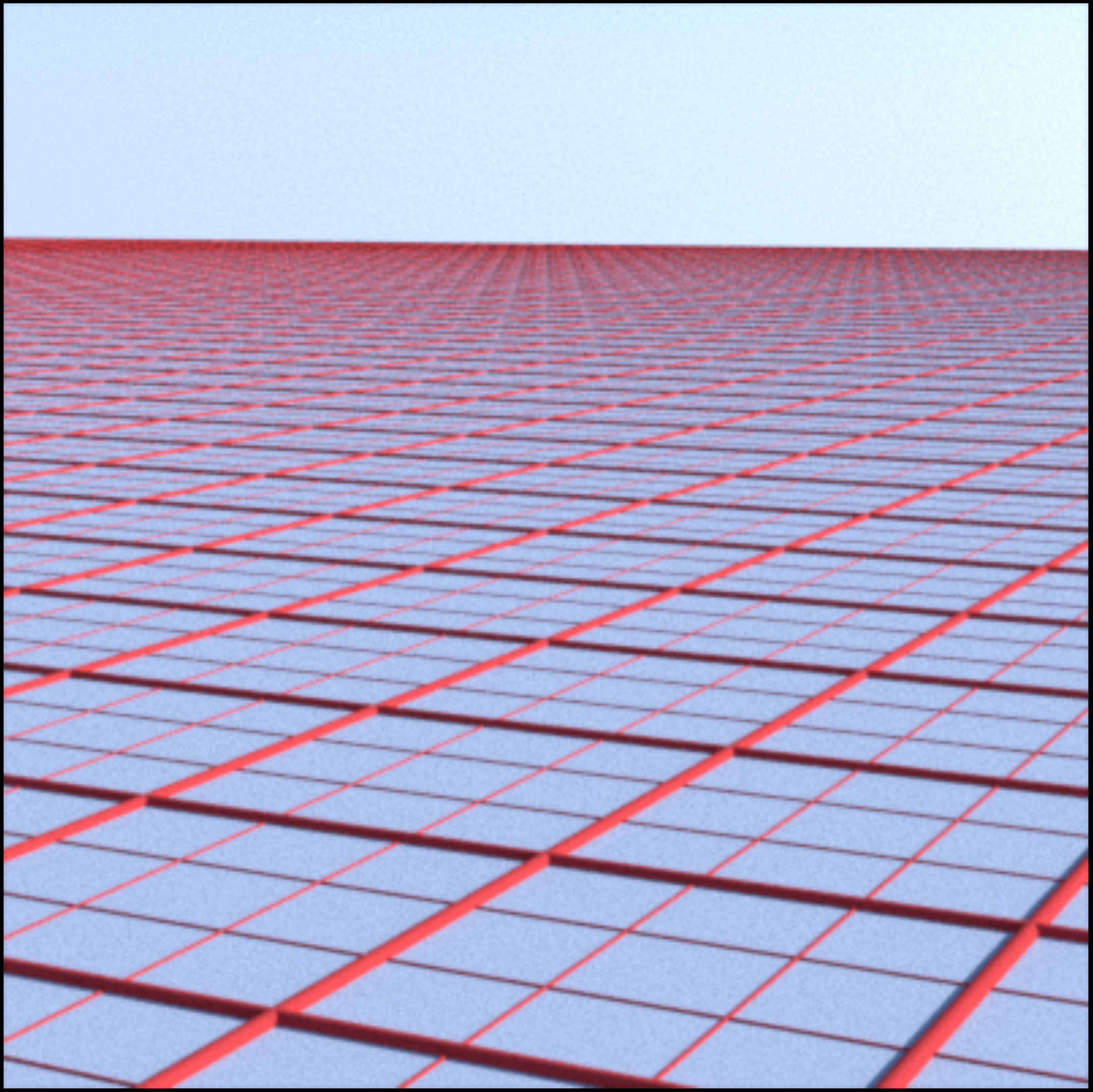
Computer Graphics Charles University

Grid Example



Computer
Graphics
Charles
University

Grid Example



Computer
Graphics
Charles
University

Grid Example