Computer Graphics

- Antialiasing -

Hans-Peter Seidel

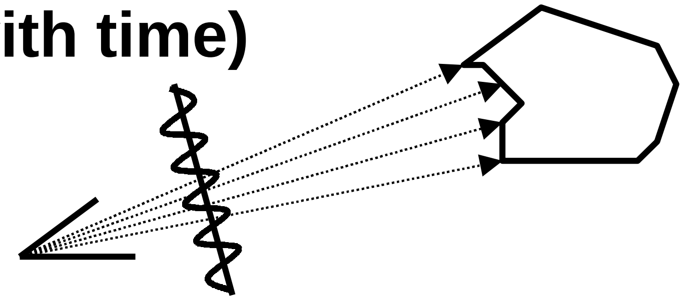
Karol Myszkowski

Overview

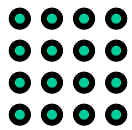
- **So far:**
  - ...
  - Ray tracing, radiosity
  - Signal processing
- **Today:**
  - Antialiasing
    - Prefiltering
    - Supersampling
    - Adaptive Sampling
- **Soon:**
  - Color
  - Imaging
  - Animation

Overview

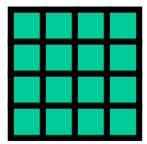
- **Continuous signal (2D/3D/4D with time)**
  - Defined for all points



- **Sampling**
  - Rays, pixel/texel, spectral values, frames, ...
- **Discrete image / image sequence**
  - Defined at isolated points, not over surfaces



, not



- **Reconstruction**
  - Spot of electronic beam on screen and human visual system
- **Impression of a continuous signal**
  - Should be similar to the original signal, no artefacts

Fourier Transformation

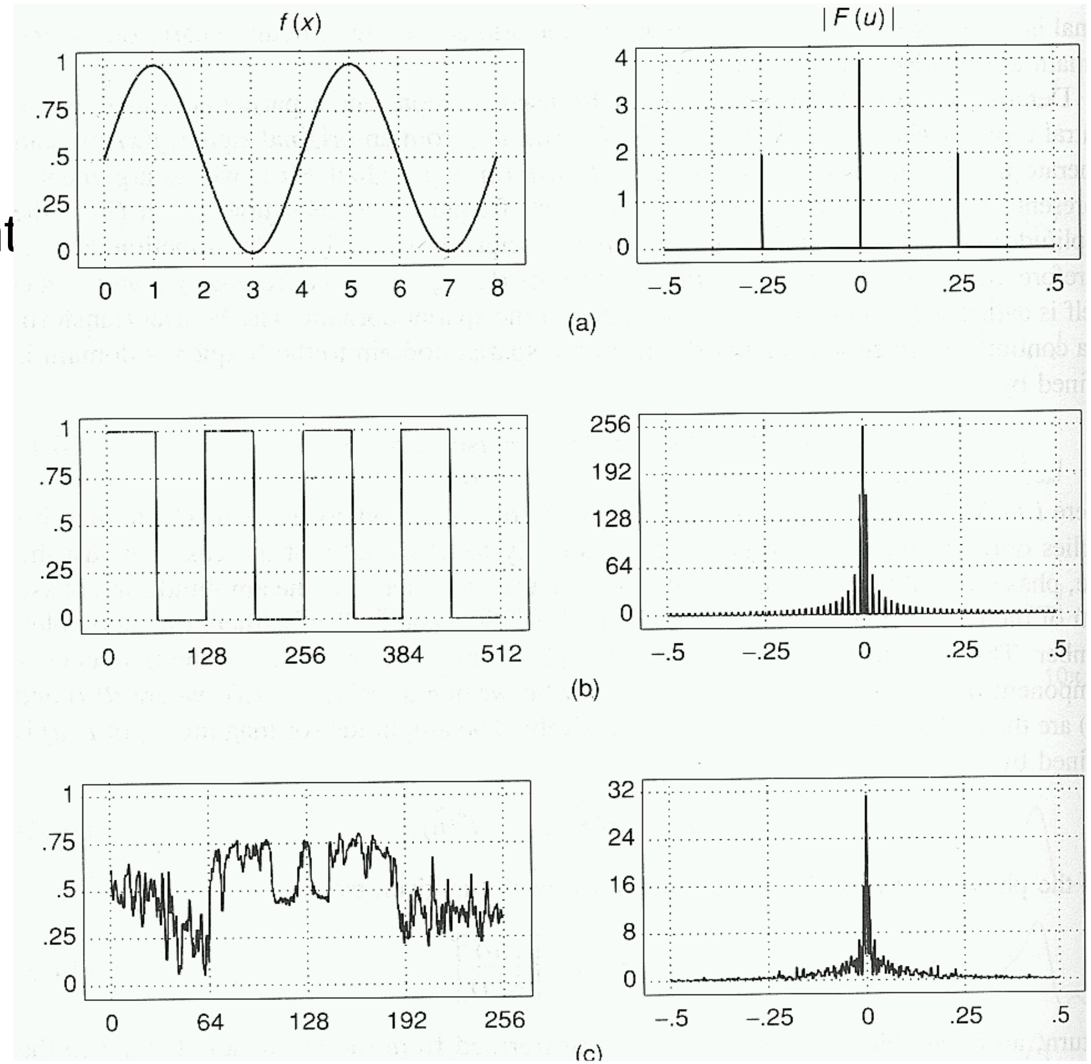
- **Spectral analysis**
  - Decomposition of a signal in different frequency bands
  - Representation of a function as weighted sum of sine and cosine functions (as orthonormal basis)
  - Two representations
    - Spatial/temporal domain: f(x)
    - Frequency domain: F(ω), spectral representation
- **Fourier transformation**
  - Conversion between the two representations
  - Functional: Convolution with complex exponential function
    - Corresponds to separate convolution with sine and cosine

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx, \quad \text{mit } e^{i\omega x} = \cos(\omega x) + i\sin(\omega x)$$

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

Spatial and Frequency Domains

- **Examples** (pixel vs cycles per pixel)
  - Positive sine with DC (= direct current) part
  - Square wave
  - Scanline of an image

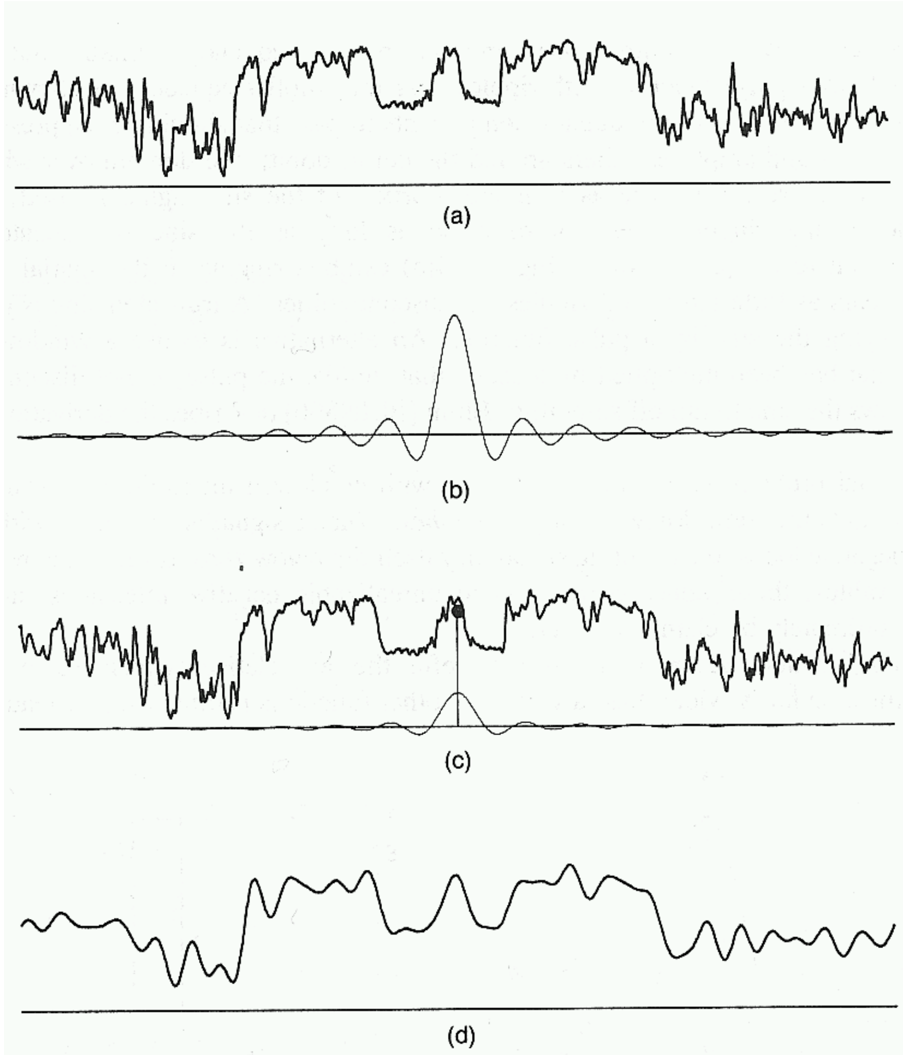


Convolution and Filtering

- **Convolution (Faltung)**

$$f(x) \otimes g(x) = \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau$$

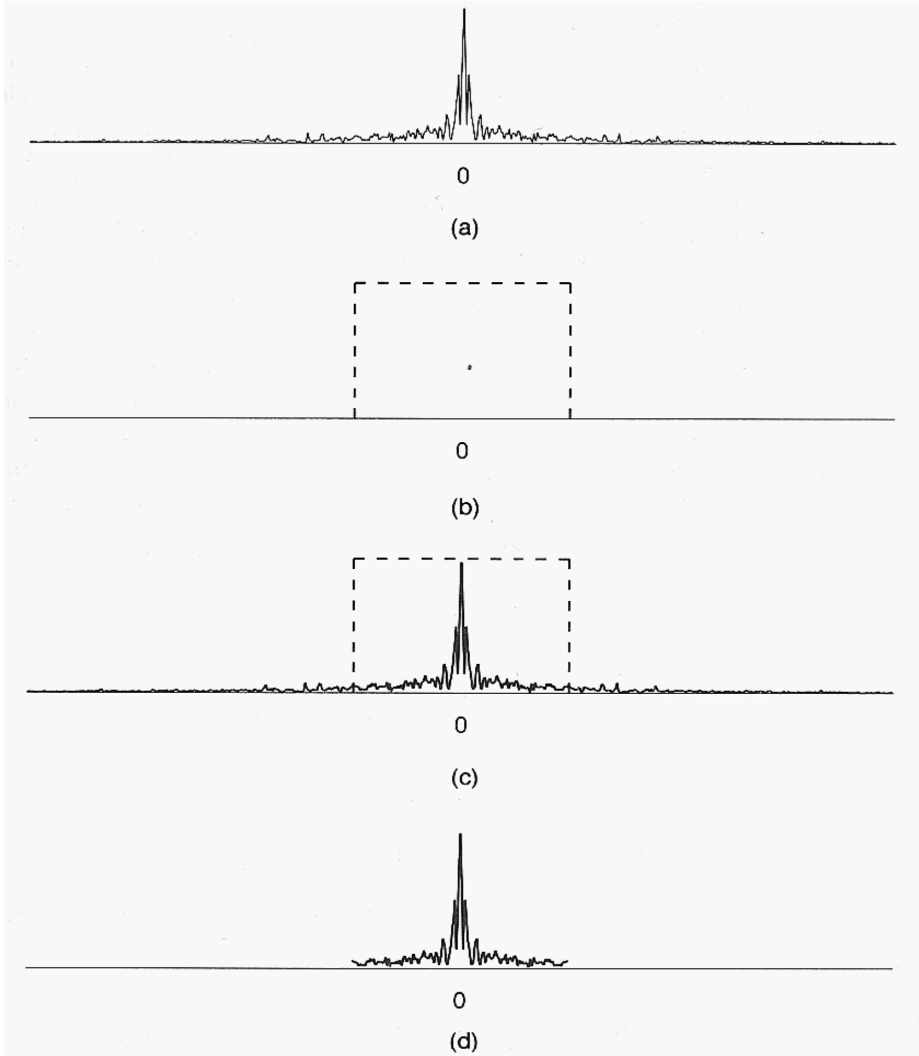
- **Convolution Theorem**
  - Multiplication in spatial domain corresponds to convolution in frequency domain
    - f • g <-> F ⊗ G
  - And vice versa (for symmetry reasons)
    - F • G <-> f ⊗ g



Low-pass filtering in spatial domain: convolution with sinc

Convolution and Filtering

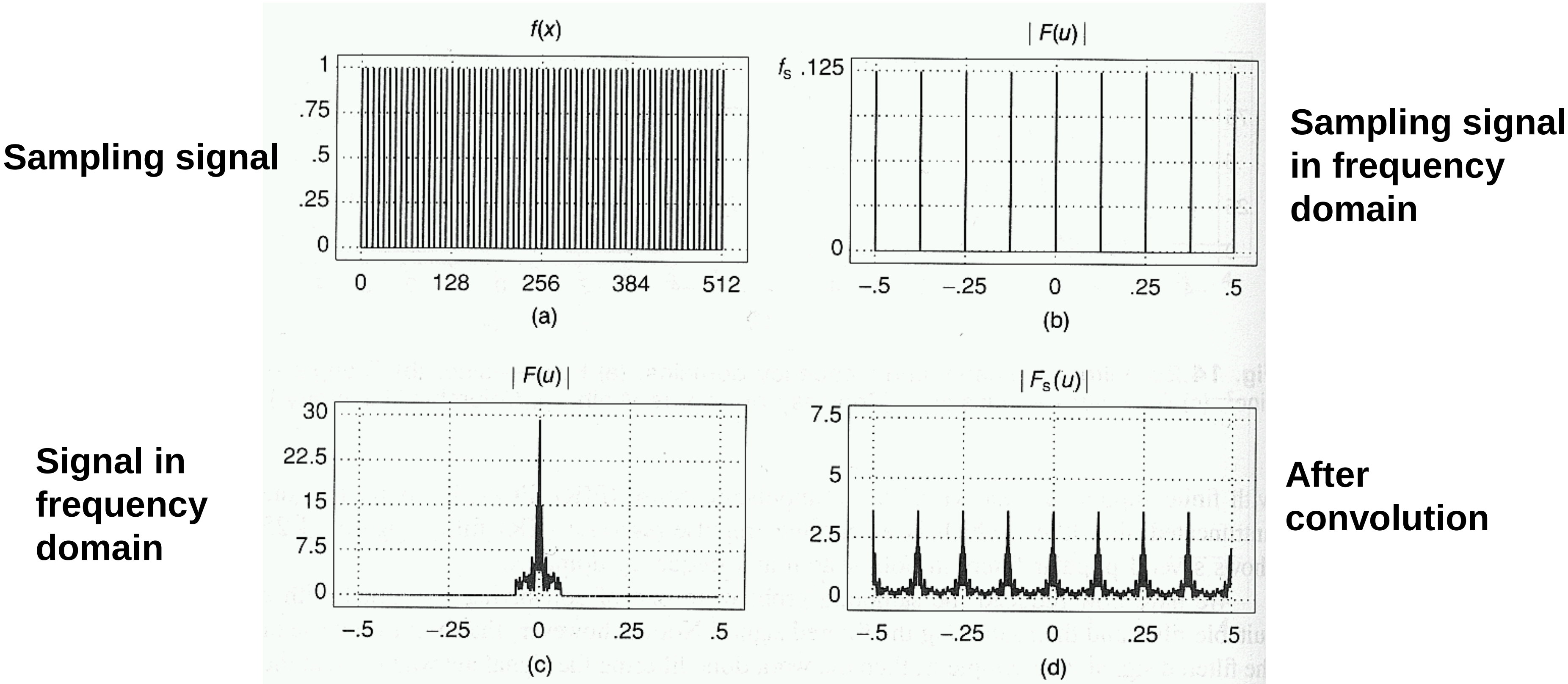
- **Low-pass filtering**
  - Convolution with sinc in spatial domain, or
  - multiplication with box in frequency domain
- **High-pass filtering**
  - Only high frequencies
- **Band-pass filtering**
  - Only intermediate frequencies



Low-pass filtering in frequency domain: multiplication with box

Sampling and Reconstruction

- **Sampling**
  - corresponds to multiplication with a comb function
    - Convolution with F(ω) in frequency domain
  - Regular delta-funktions (e.g. Pixel center)



Sampling and Reconstruction

Original function and its frequency spectrum

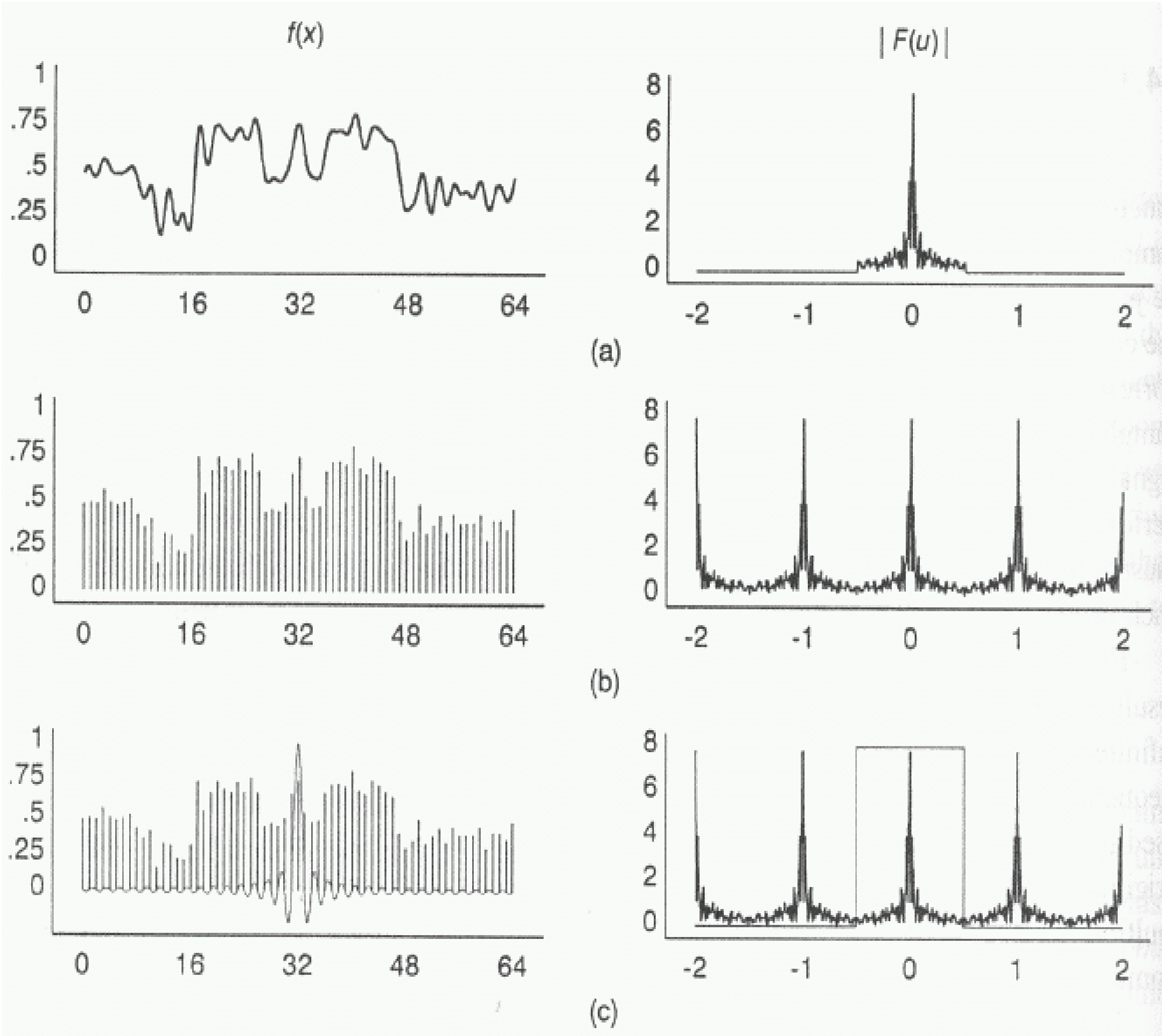
Sampled signal

Convolution of comb with F(u)

Signal is replicated!!

Before Reconstruction

Only a copy!!!



Sampling and Reconstruction

Reconstruction with ideal sinc

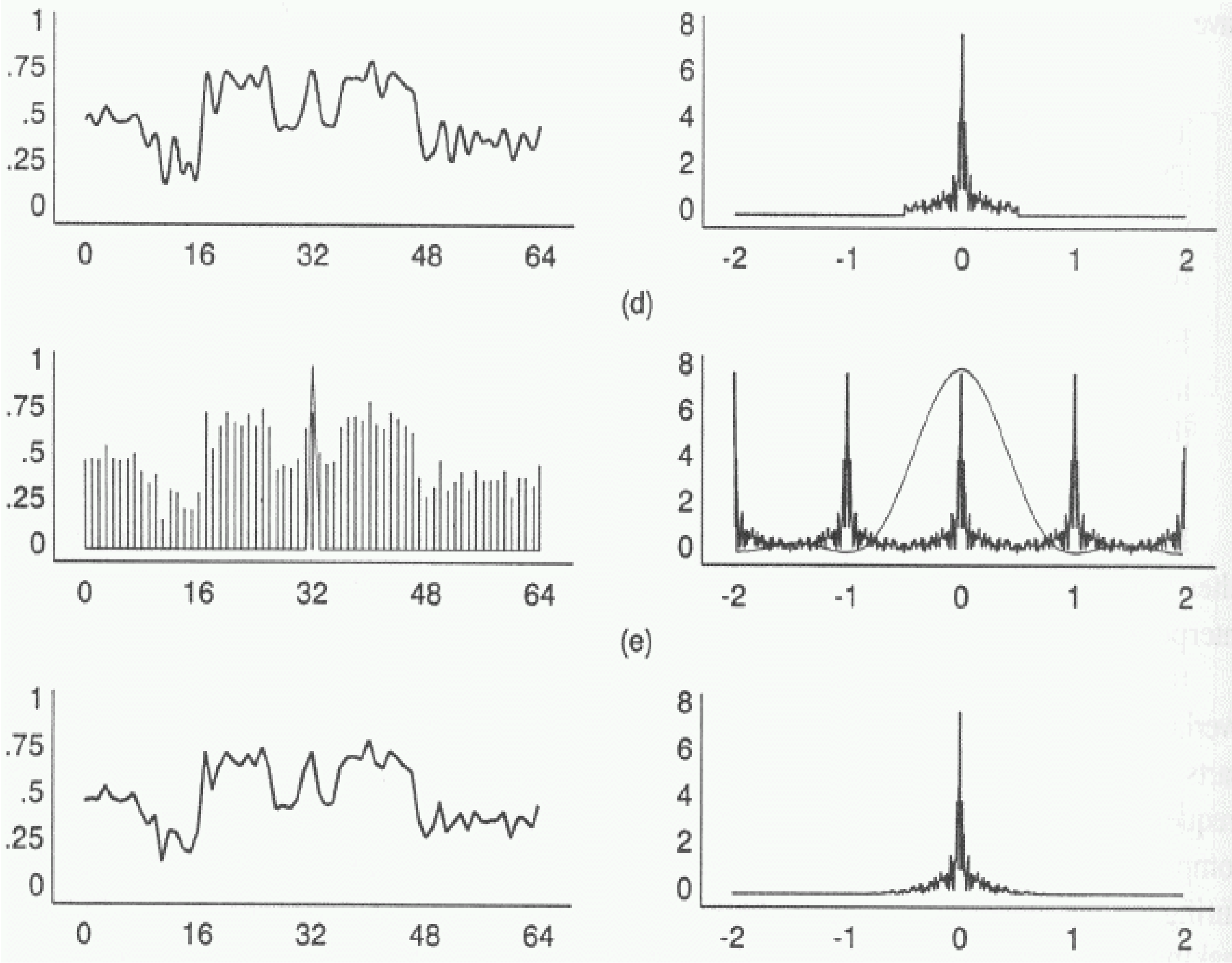
Identical signal

Before reconstruction with hat function

High frequencies are not ignored

➔ Aliasing

Reconstruction with hat function (= piecewise linear interpolation)



Sampling with Low Frequency

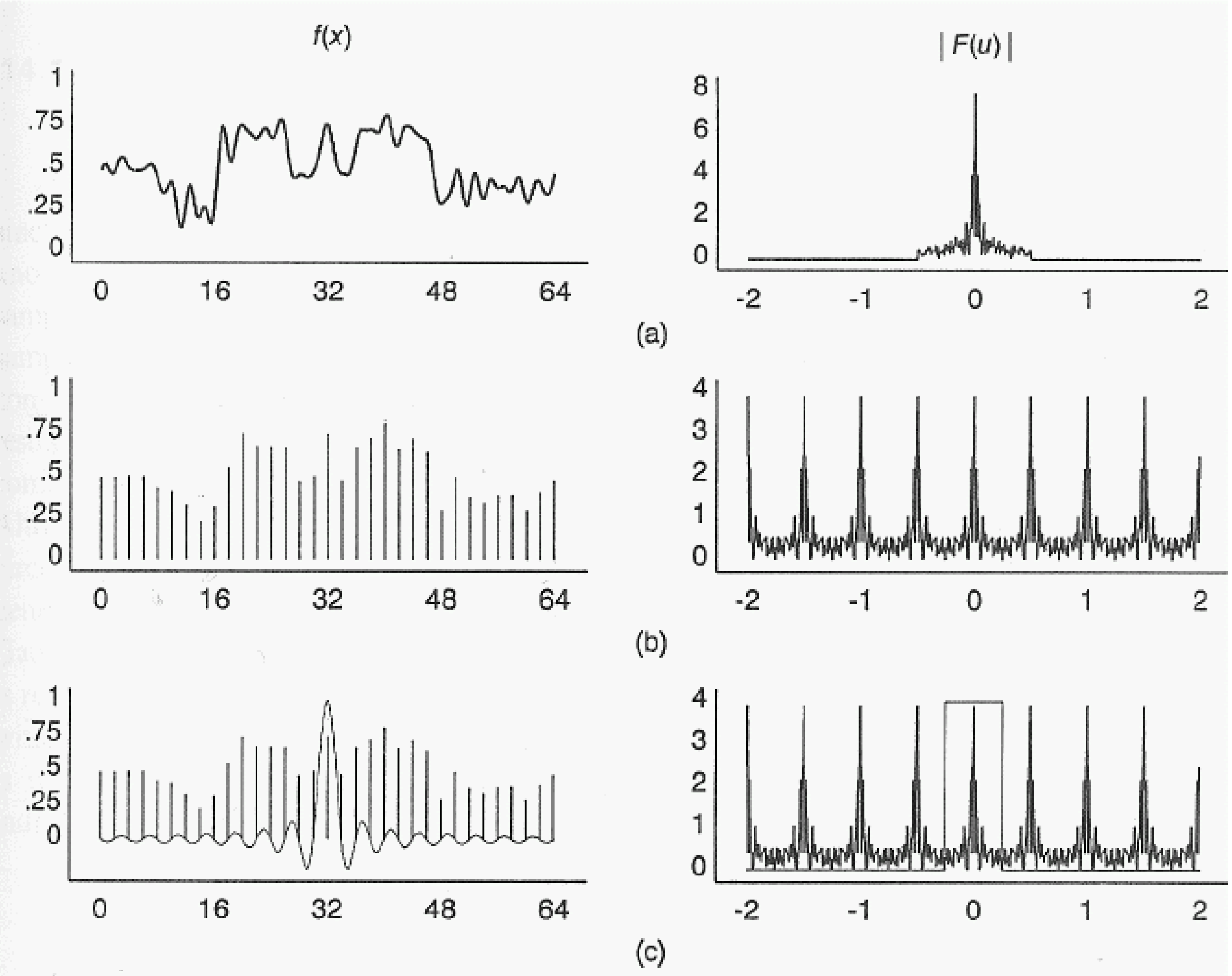
Function and its frequency spectrum

Sampling with low frequency

Convolution of comb with F(u)

Before reconstruction with suitable sinc

Reconstruction fails (overlap in frequency domain)

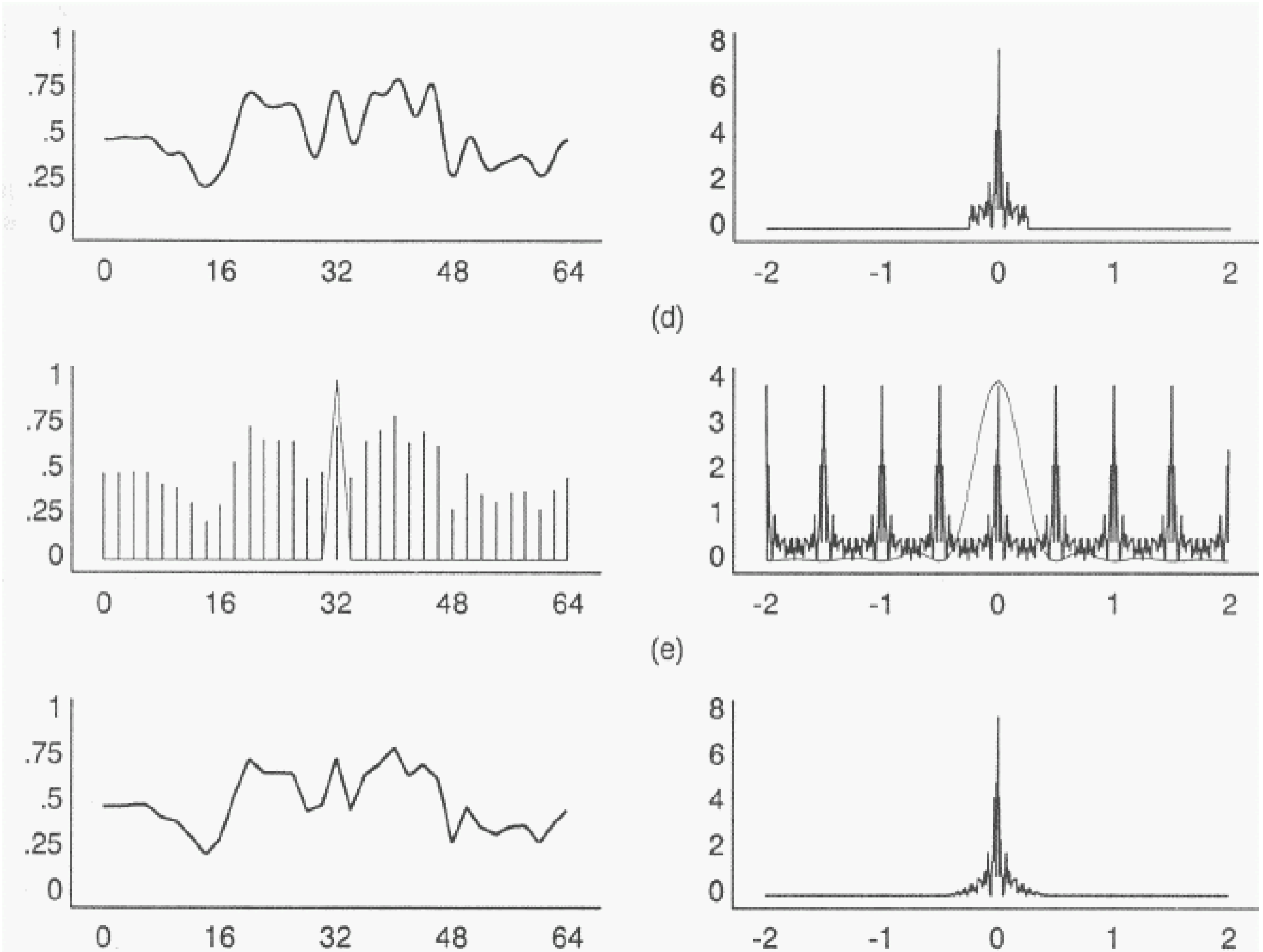


Sampling with Low Frequency

Reconstruction with suitable sinc

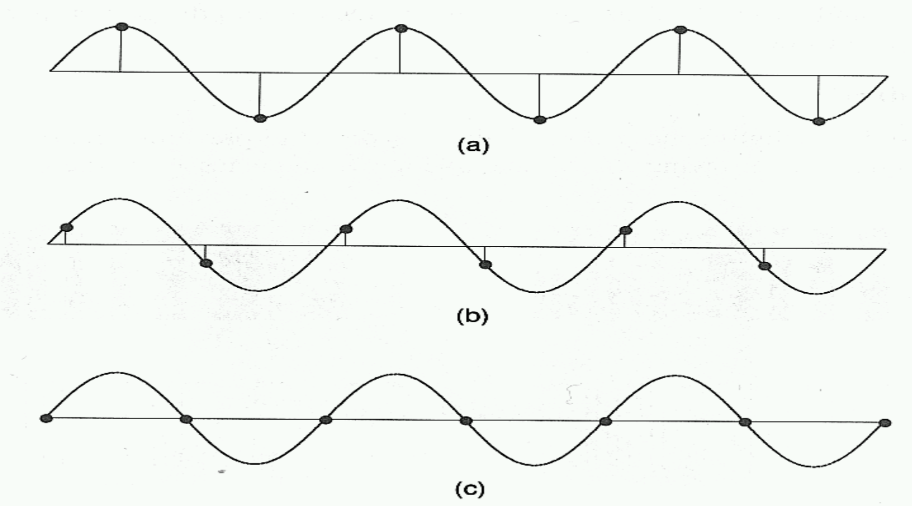
Before reconstruction with hat function

Reconstruction with hat function (= piecewise linear interpolation)



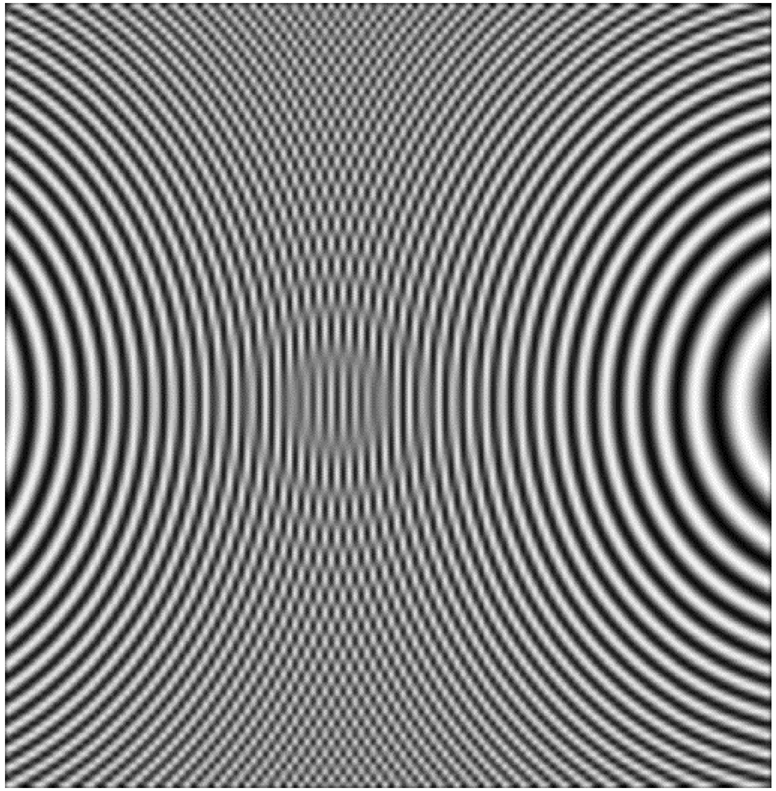
Nyquist Frequency

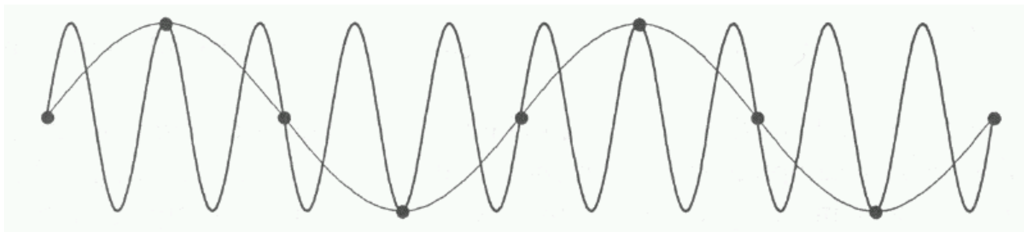
- **Shannon: Sampling-Theorem `49**
  - „A signal can be properly reconstructed from its samples if the original signal is sampled at a frequency that is greater than twice the highest frequency component in its spectrum“
  - In short: Sampling frequency > 2 * highest frequency component (in particular: signal must be band limited)
- **Nyquist**
  - Critical frequency is called Nyquist frequency



Aliasing Artefacts

- **Moiré Patterns**
- **Aliasing**



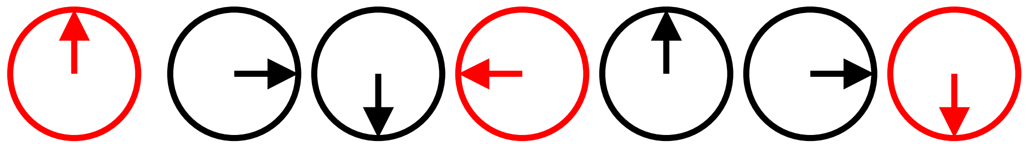


Sampling Artefacts

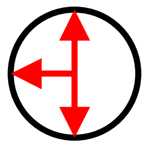
- **Spatial aliasing:**
  - Stair cases, Moiré patterns, etc.
- **Solutions:**
  - Increasing the sampling rate
    - Ok, but infinite frequencies at sharp edges
  - Postfiltering (after reconstruction)
    - Doesn‘t work - only leads to blurred stair cases
  - Prefiltering (Blurring) of sharp geometry features
    - Slowly make geometry transparent at the edges
    - Correct solution in principal
    - Analytic low-pass filtering hard to implement
    - Supersampling

Sampling Artefacts

- **Temporal Aliasing**
  - Car wheels, ...



- **Solutions**
  - Increasing the frame rate
    - OK
  - Prefiltering (Motion Blur)
    - Yes, possible for simple geometry (e.g.. Comic)
    - Problems with texture, etc.
  - Postfilterung (Averaging several frames)
    - Doesn‘t work – only multiple detail

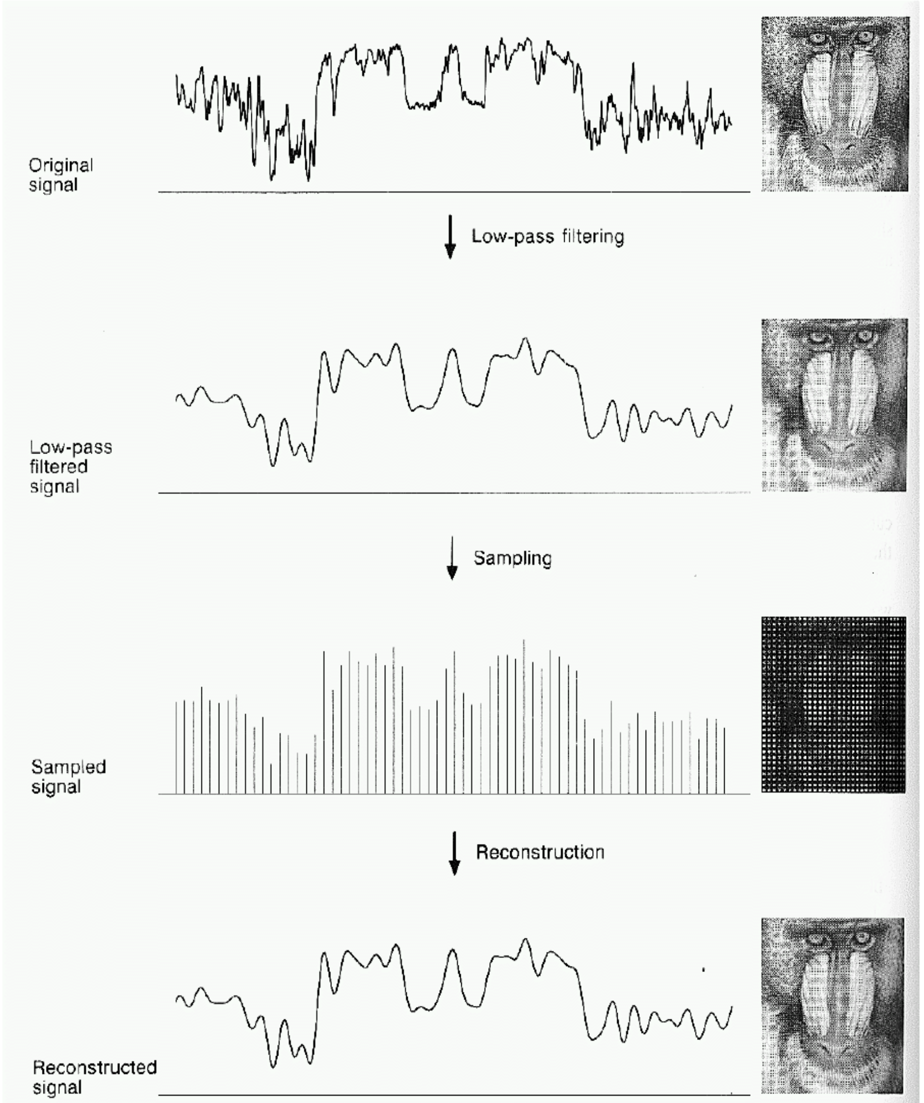


- **Important**
  - Distinction between aliasing errors and reconstruction errors

Antialiasing by prefiltering

- **Filterung before sampling**
  - Analog/analytic or
  - Sampling with higher frequency (super sampling)

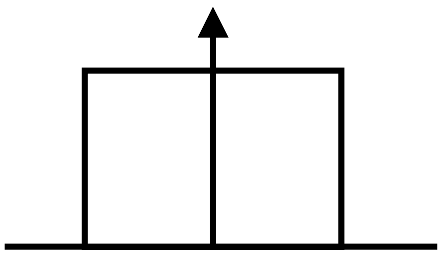
- **Ideal reconstruction**
  - Convolution with sinc
- **Practical reconstruction**
  - Convolution with
    - Box filter or
    - Circle filter
  - Sampling-Rate must then be significantly higher than Nyquist frequency

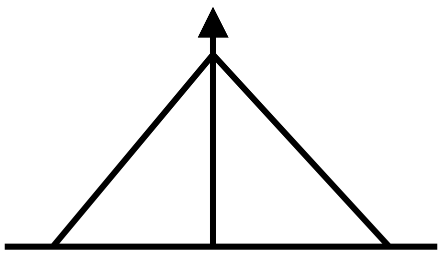


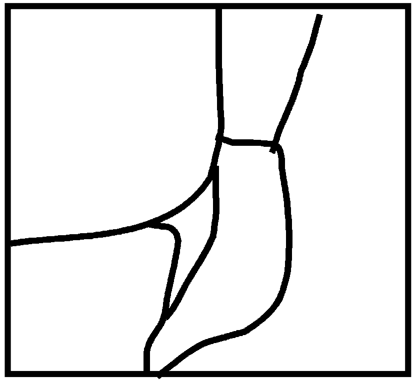
Sources of High Frequencies

- **Geometry**
  - Edges, Vertices, sharp boundaries
  - Silhouettes (view dependent)
- **Texture**
  - Chess board patterns, lots of detail
- **Illumination**
  - Shadows, lighting effects, projections

➔**Analytic filtering almost impossible**
  - Even with the most simple filters

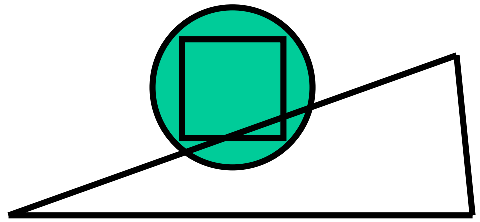


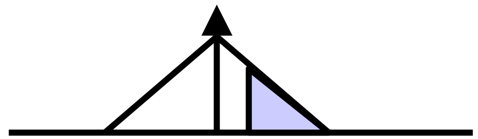




Comparison

- **Analytic low-pass filtering**
  - Ideally eliminates aliasing completely
  - Hard to implement
    - Only works for polygon edges with constant color
    - Weighted or unweighted area sampling
    - Compute distance from pixel to a line
    - Filter values can be stored in look-up tables
      - Possibly taking into account slope
      - Distance correction
      - Non rotationally symmetric filters
    - Doesn‘t work for corners

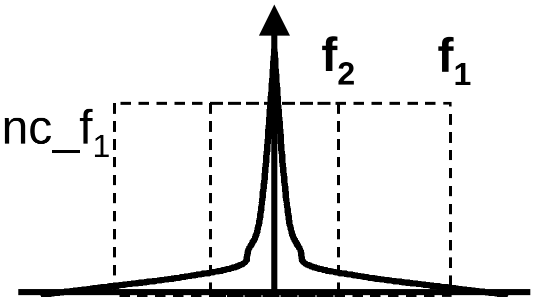




- **Over-/Supersampling**
  - Very easy to implement
  - Doesn‘t eliminate aliasing completely
    - Sharp edges contain infinitely high frequencies

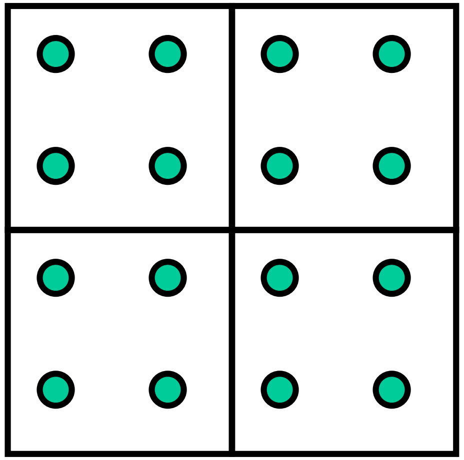
Resampling Pipeline

- **Assumption**
  - Energy in high frequencies decreases
  - Reduced aliasing by sampling with higher frequencies
- **Algorithm**
  - Supersampling
    - Sample continuous signal with boundary frequency f_1
    - Aliasing with energy beyound f_1
  - Reconstruction of signal
    - Filtering with $g_1(x)$: e.g. convolution with sinc_f_1
  - Analytic low-pass filtering of signal
    - Filtering with filter $g_2(x)$ with $f_2 << f_1$
    - Signal becomes band limited w.r.t. f_2
  - Resampling 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 (SGI Reality Engine, multisampling)



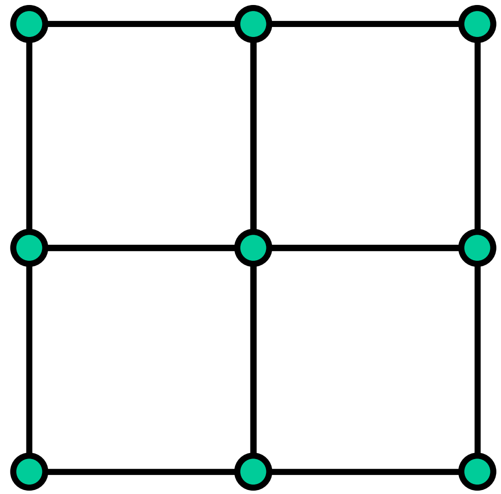
Supersampling in Praxis

- **Regular supersampling**
  - Averaging of N samples per pixel on a grid
  - N:
    - 4 quite good
    - 16 almost always sufficient
  - Samples
    - Rays, z-buffer, reflexion, motion, ...
  - Averaging
    - Box filter
      - Up to 5 resp. 17 intensity levels
    - Others: Pyramid (Bartlett), B-Spline, Hexagonal, ...
  - Regular supersampling
    - Nyquist frequency for aliasing only shifted
      - ➔ Irregular sampling patterns



Supersampling Caveats

- **Popular mistake**
  - Sampling at the corners of every pixel
  - Pixel color by averaging
- **Problem**
  - Wrong reconstruction filter !!!
  - Same sampling frequency, but postfiltering with a box
  - Blurring: Loss of information
- **Post-Rekonstructions-Blur**



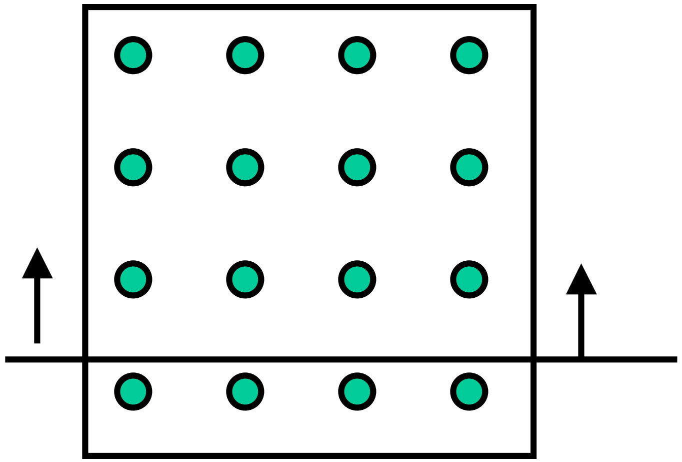
1x1 Sampling, 3x3 Blur

1x1 Sampling, 7x7 Blur

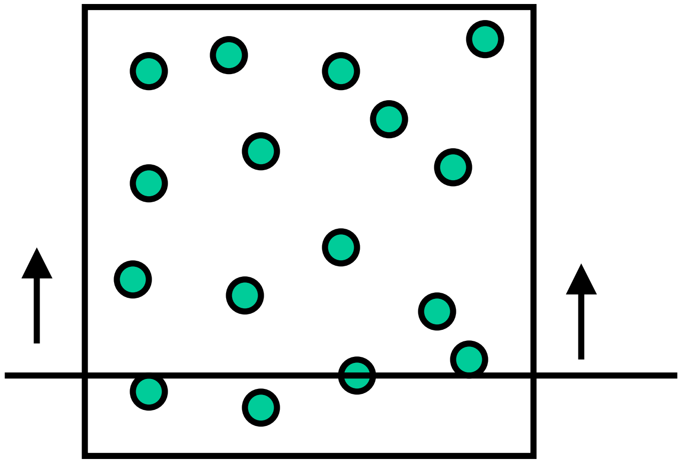
➔„Supersampling“ doesn‘t come for free

Supersampling in Praxis

- **Problems with regular supersampling**
  - 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



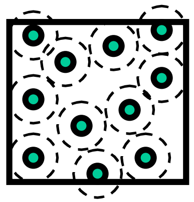
Better, but noisy

➔ **Stochastic supersampling**

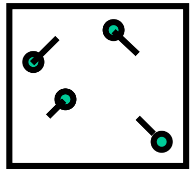
- Or analytic computation of pixel coverade and pixel mask

Stochastic Sampling

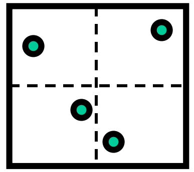
- **Requirements**
  - Even distribution
  - Little correlation between samples
  - Incremental generation
- **Generation of samples**
  - Poisson-disk-sampling
    - Fixes a minimum distance between samples
    - Random generation of samples
      - Rejection, if too close to other samples



  - Jittered sampling
    - Random perturbation from regular positions



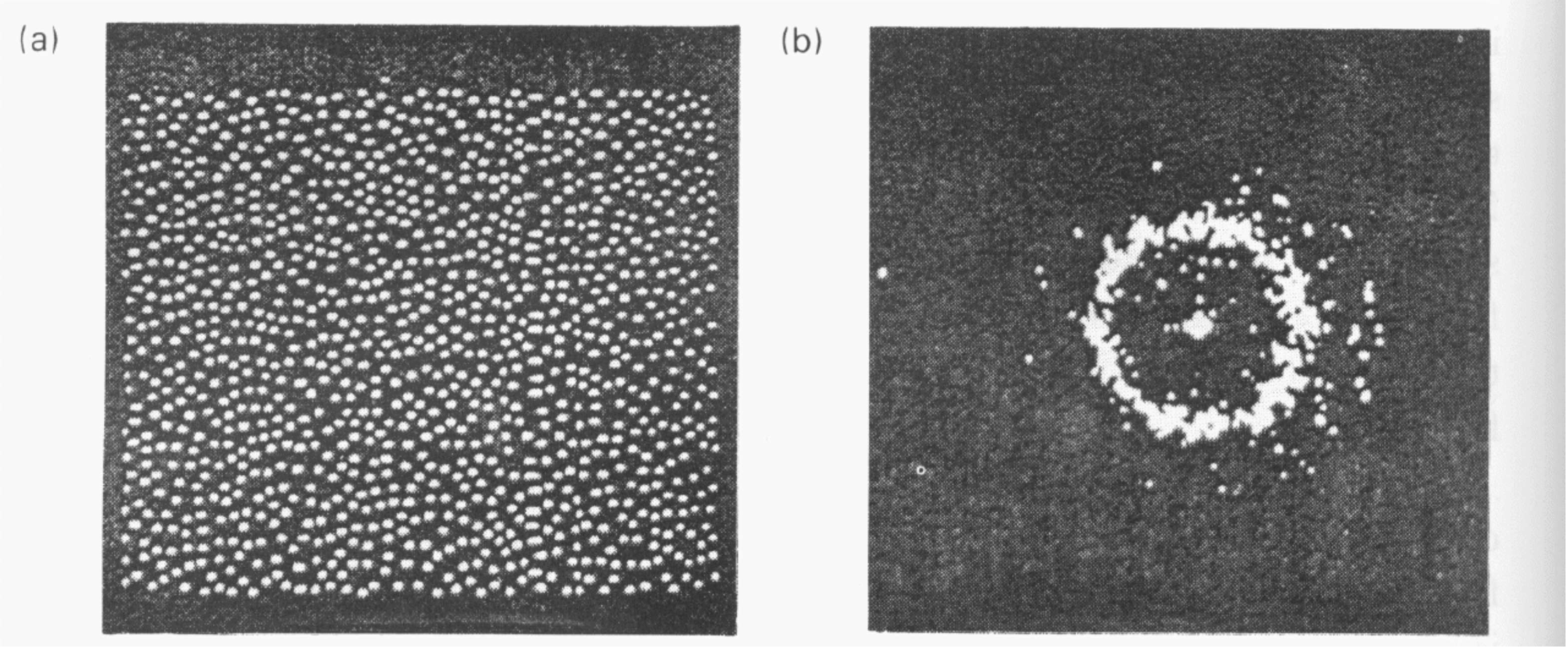
  - Stratified Sampling
    - Subdivision in areas with one random sample each



  - Quasi-random numbers (Quasi-Monte-Carlo)

Poisson-Disk-Sampling

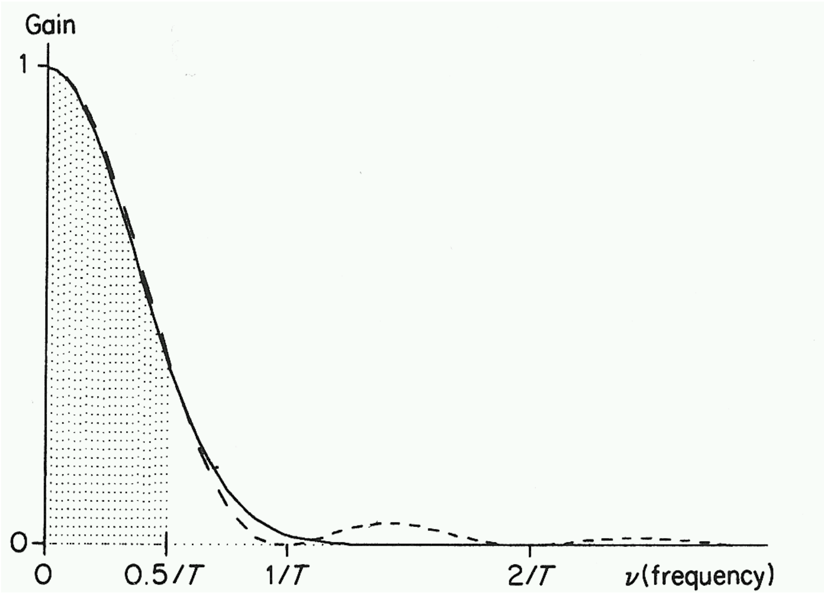
- **Motivation**
  – Distribution of the optical receptors on the retina (here: ape)

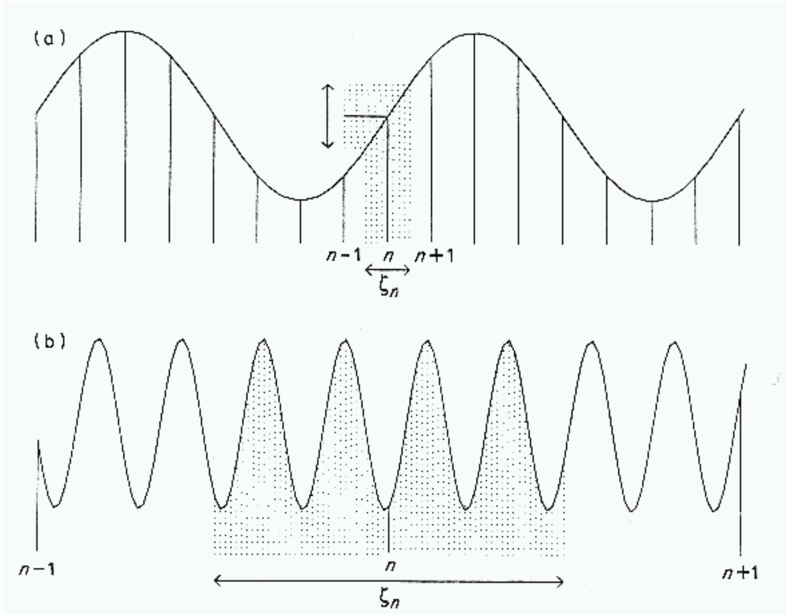


Distribution of the receptors **Fourier analysis**

Why does it work?

- **Exploits human perception**
  - Very sensitive to regular structures
  - Insensitive against (high frequency) noise
- **Stochastic Sampling**
  - Transforms ignored high frequency bands into noise

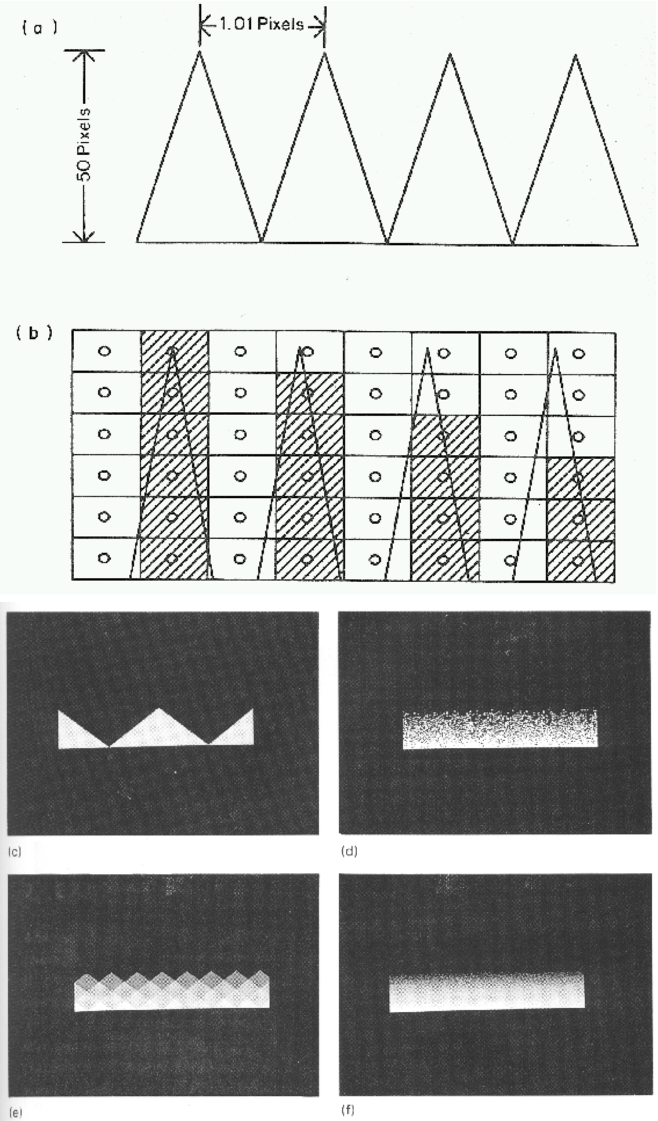




—— Gaußsches Jittering

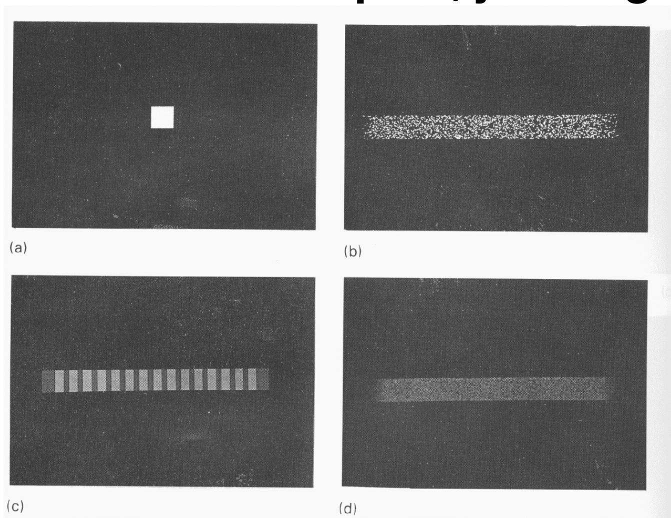
– – – White-Noise Jittering

Examples



Triangle comb
(B: 1.01 pix, H: 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

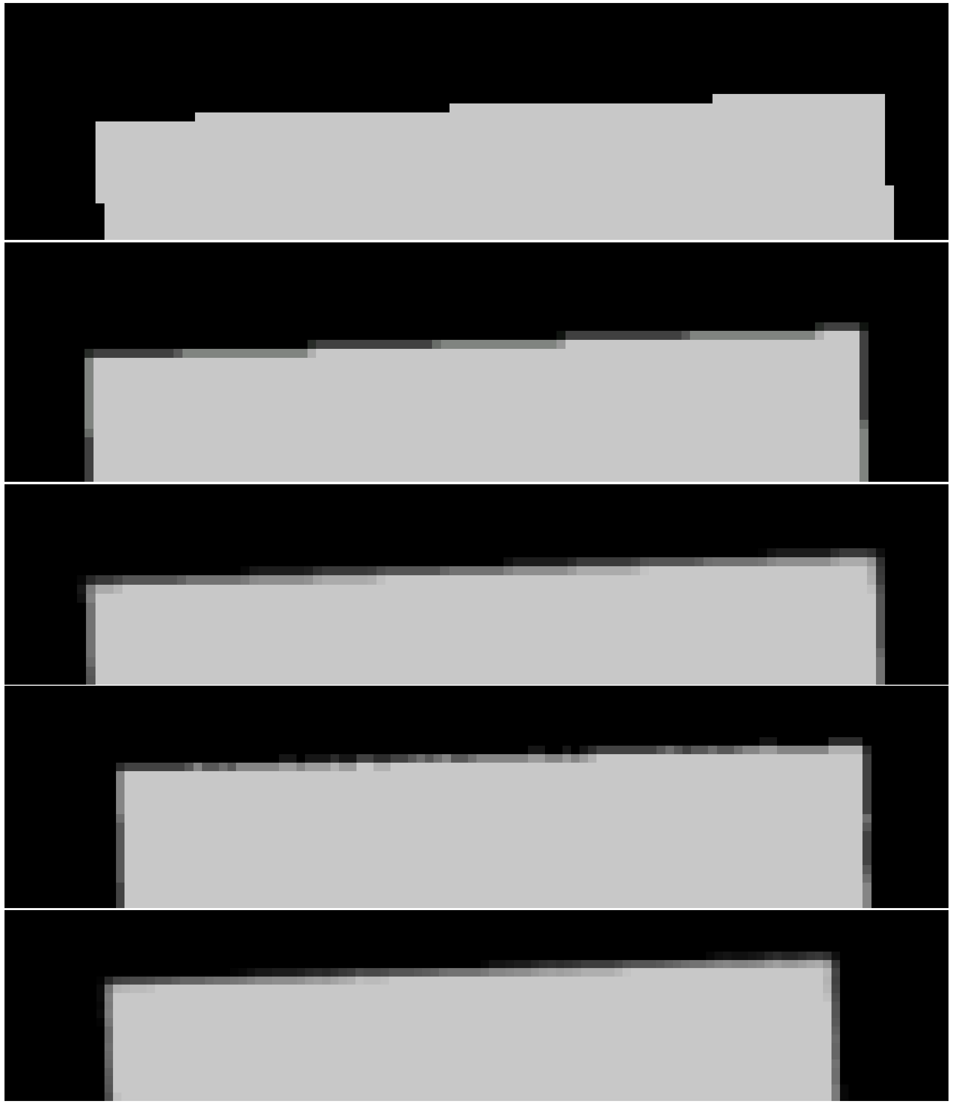
Regular, 1x1

Regular 3x3

Regular, 7x7

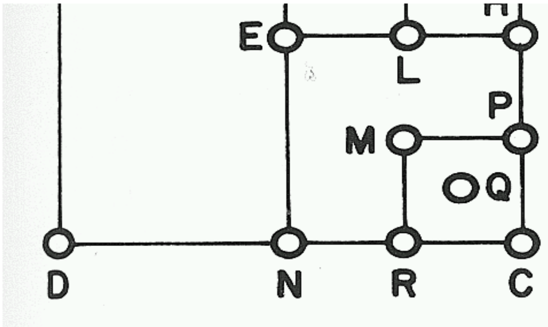
Jittered, 3x3

Jittered, 7x7



Adaptive Supersampling

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



- **Extension**
  - Jittering of the samples