

# Computer Graphics

## Sampling Theory & Anti-Aliasing

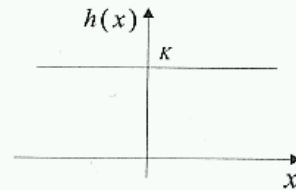
**Philipp Slusallek**

# Dirac Comb (1)

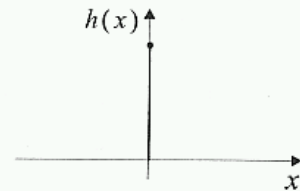
- **Constant &  $\delta$ -function**

– flash

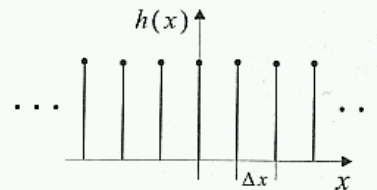
Ortsbereich



Konstante Funktion

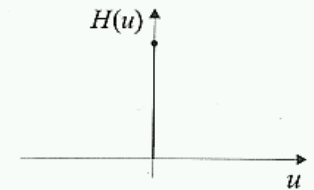


Delta-Funktion

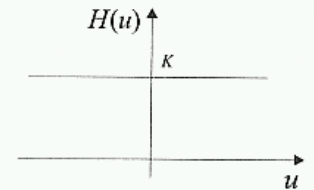


Kamm-Funktion

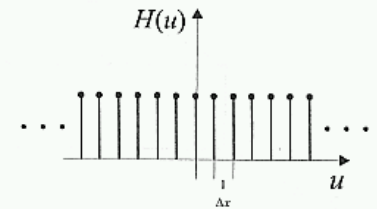
Ortsfrequenzbereich



Delta-Funktion



Konstante Funktion



Kamm-Funktion

$$h(x) = K$$

$$H(u) = K\delta(u)$$

$$h(x) = K\delta(x)$$

$$H(u) = K$$

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

$$H(u) = \frac{1}{\Delta x} \sum_{k=-\infty}^{\infty} \delta(u - \frac{k}{\Delta x})$$

- **Comb/Shah function**

# Dirac Comb (2)

- **Constant &  $\delta$ -Function**

- Duality

$$f(x) = K$$

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

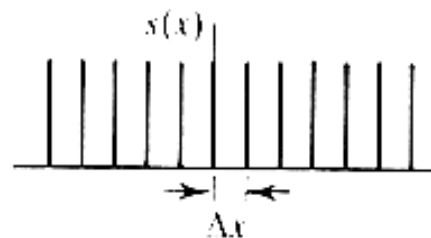
- And vice versa

- **Comb function**

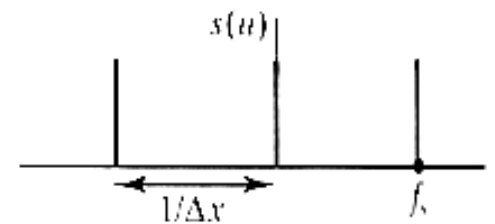
- Duality: the dual of a comb function is again a comb function

- Inverse wavelength
- Amplitude scales with inverse wavelength

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

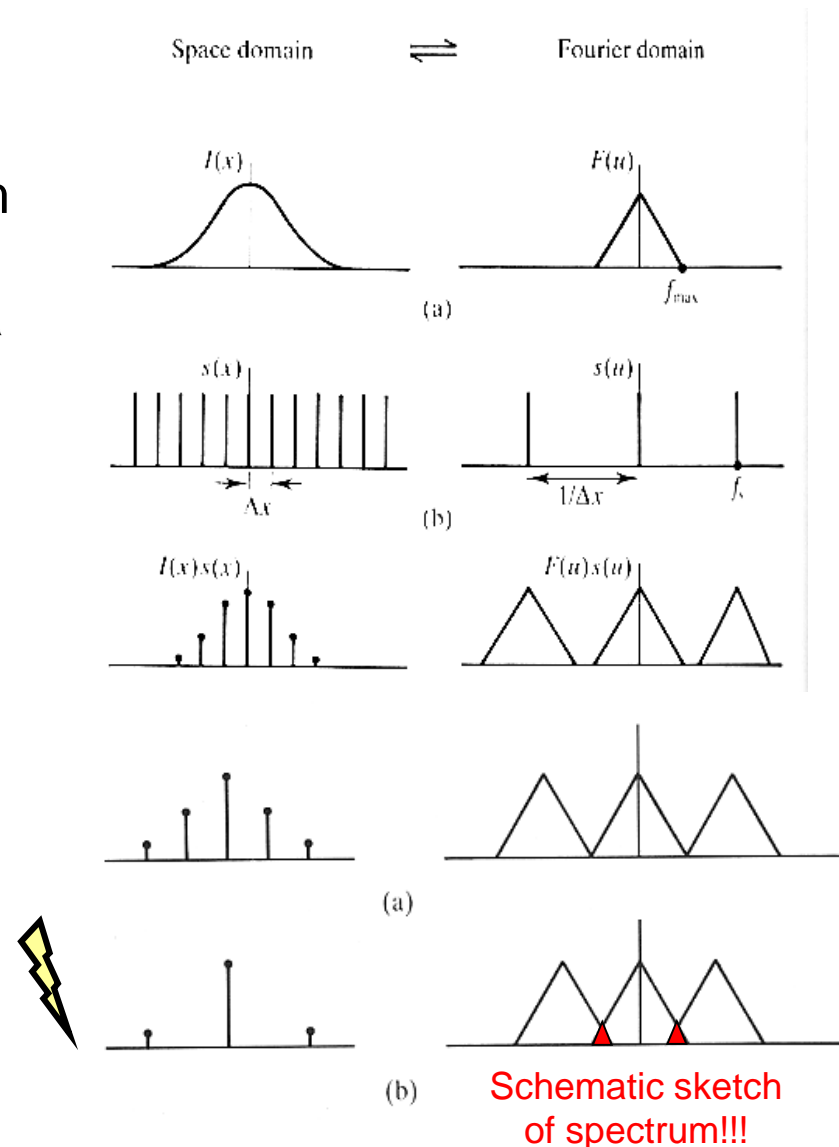


(b)



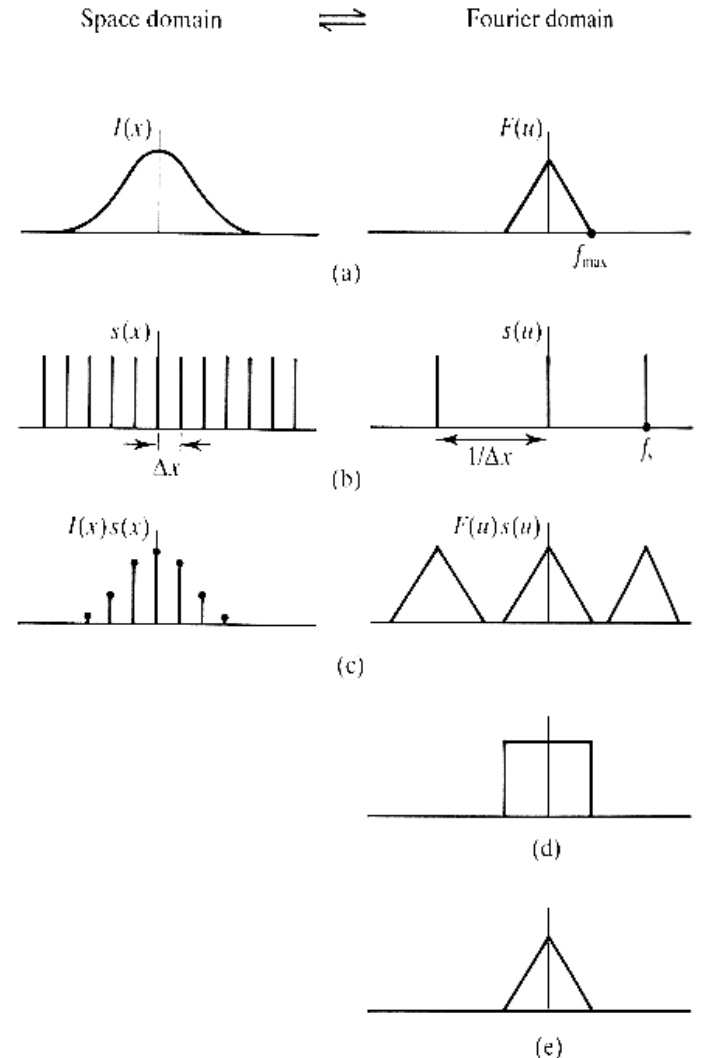
# Sampling

- **Continuous function**
  - Assume band-limited
  - Finite support of Fourier transform
    - Depicted here as triangle-shaped finite spectrum (not meant to be a tent function)
- **Sampling at discrete points**
  - Multiplication with Comb function in spatial domain
  - Corresponds to convolution in Fourier domain
    - ⇒ Multiple copies of the original spectrum (convolution theorem!)
- **Frequency bands overlap ?**
  - No : good
  - Yes: aliasing artifacts



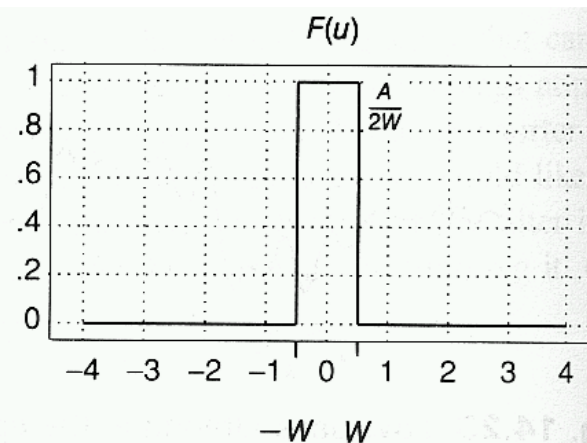
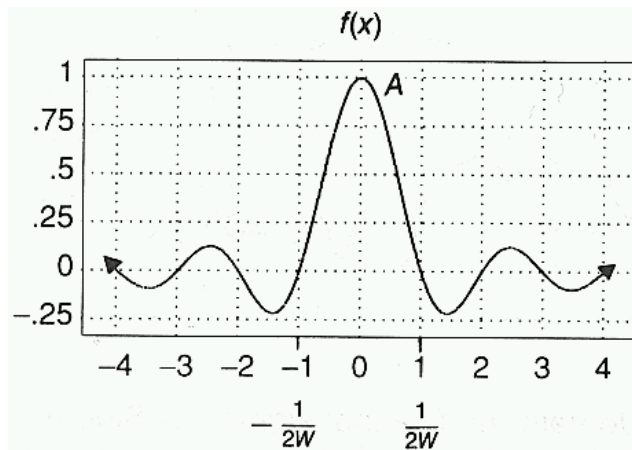
# Reconstruction

- **Only original frequency band desired**
- **Filtering**
  - In Fourier domain:
    - Multiplication with windowing function around origin
  - In spatial domain
    - Convolution with inverse 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 due to finite support.

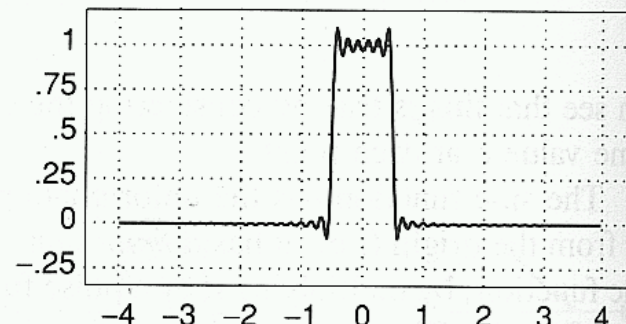
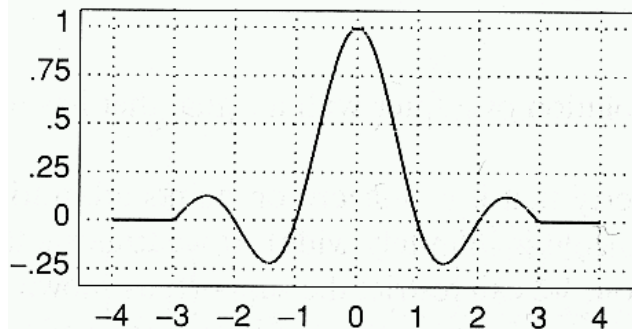


# Reconstruction Filter

- Cutting off the spatial support of the *sinc* function is **NOT** a good solution
  - Re-introduces high-frequencies  $\Rightarrow$  spatial ringing



(a)



(b)

# Sampling and Reconstruction

Original function and its band-limited frequency spectrum

Signal sampling beyond Nyquist:

Mult./conv. with comb

Frequency spectrum is replicated

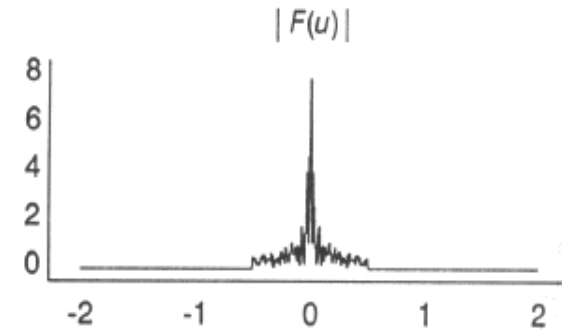
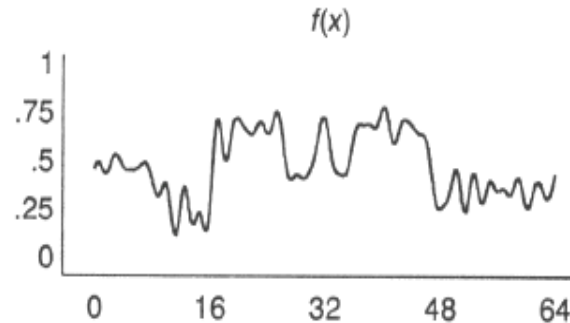
Comb dense enough (sampling rate  $> 2 \cdot \text{bandlimit}$ )

Bands do not overlap

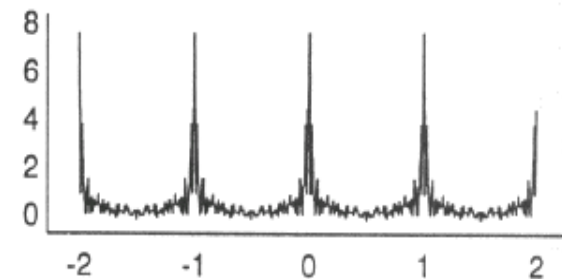
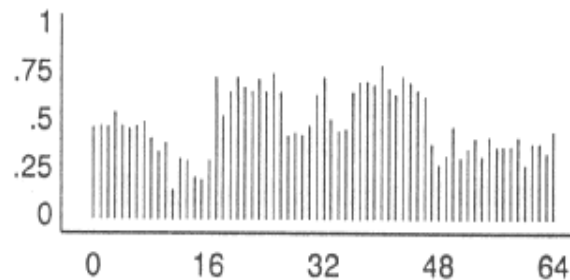
Ideal filtering

Fourier: box (mult.)  
Space: *sinc* (conv.)

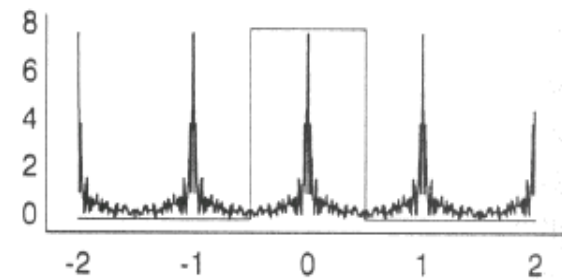
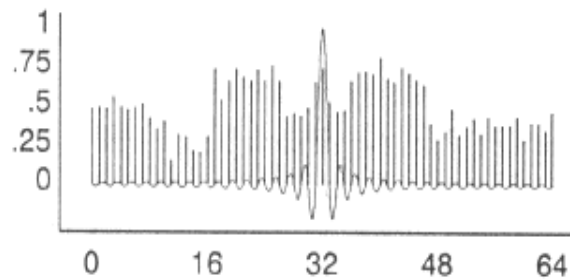
Only one copy



(a)



(b)

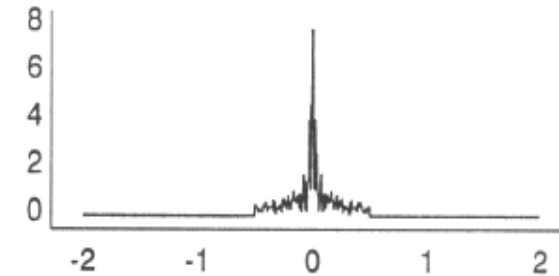
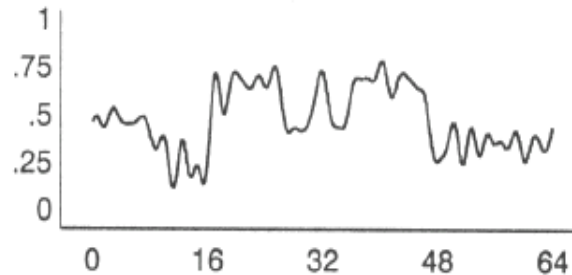


(c)

# Sampling and Reconstruction

Reconstruction  
with ideal *sinc*

Identical signal

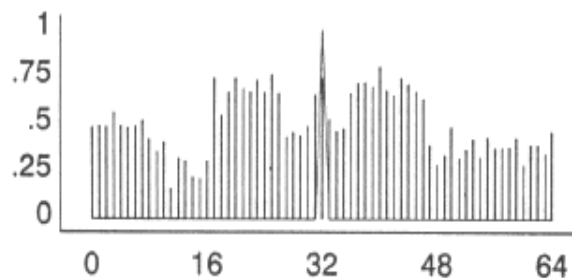


Non-ideal filtering

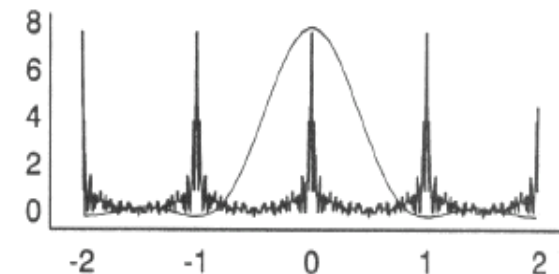
Fourier:  $\text{sinc}^2$  (mult.)  
Space: tent (conv.)

Artificial high frequen.  
are not cut off

⇒ Aliasing artifacts

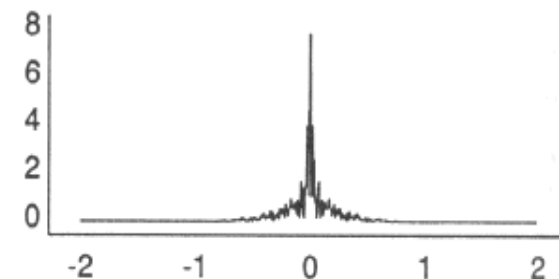
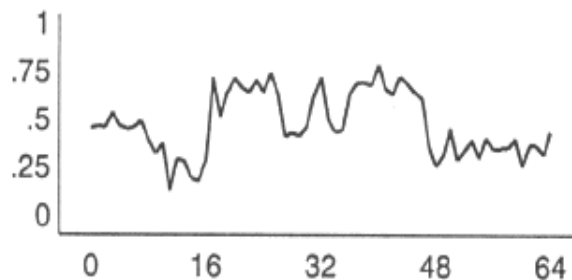


(d)



(e)

Reconstruction with  
tent function  
(= piecewise linear  
interpolation)





# Sampling at Too Low Frequency

Original function and its band-limited frequency spectrum

Signal sampling below Nyquist:

Mult./conv. with comb

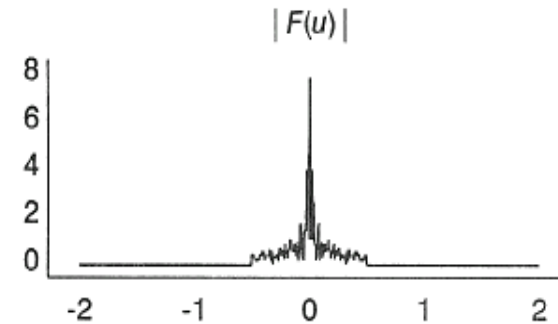
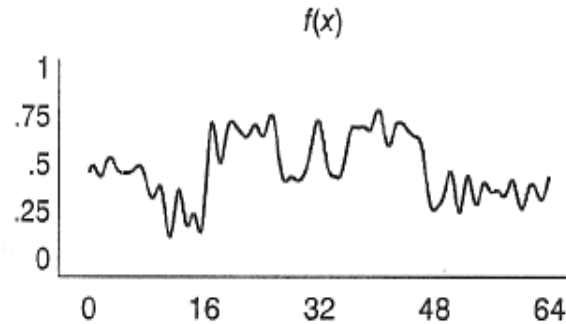
Comb spaced too far (sampling rate  $\leq 2 \cdot \text{bandlimit}$ )

**Spectral band overlap: artificial low frequenci.**

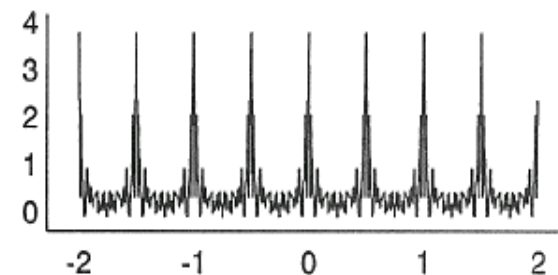
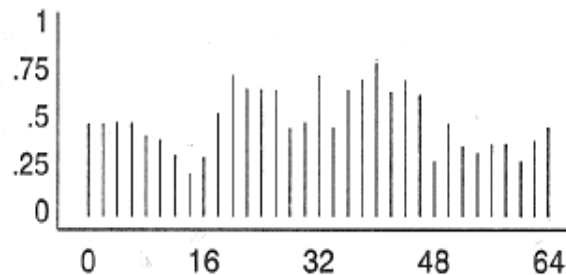
Ideal filtering

Fourier: box (mult.)  
Space: *sinc* (conv.)

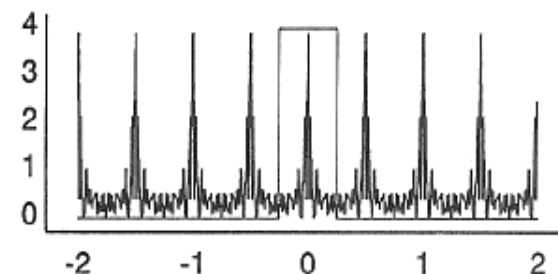
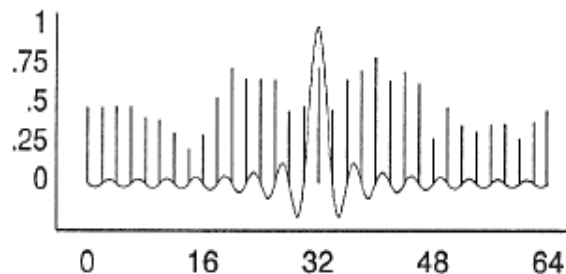
**Band overlap in frequency domain cannot be corrected**  
 $\Rightarrow$  **Aliasing**



(a)



(b)



(c)

# Sampling at Too Low Frequency

Reconstruction  
with ideal *sinc*

Reconstruction fails  
(frequency  
components wrong  
due to aliasing !)

Non-ideal filtering

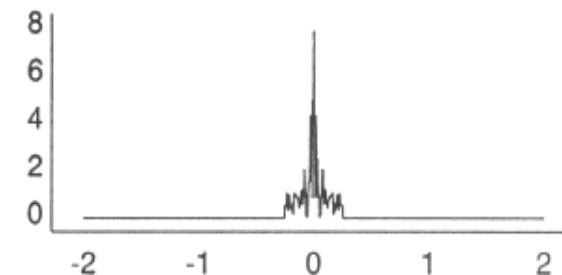
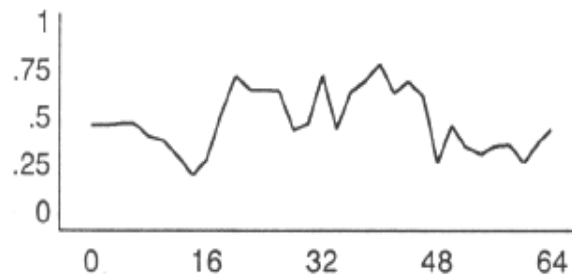
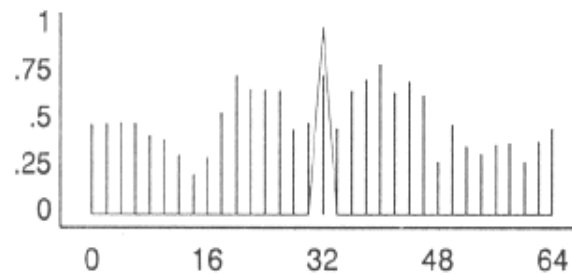
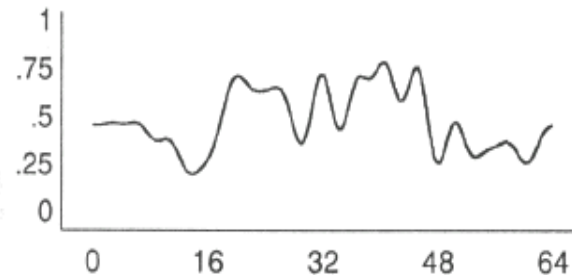
Fourier:  $\text{sinc}^2$  (mult.)  
Space: tent (conv.)

Artificial high frequen.  
are not cut off

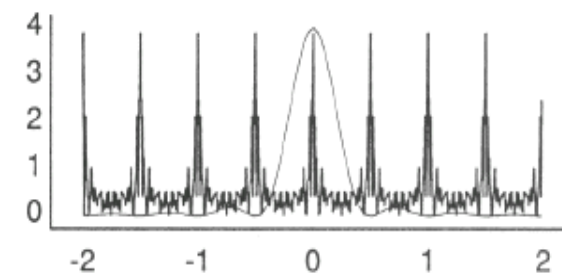
⇒ Aliasing artifacts

Reconstruction with  
tent function  
(= piecewise linear  
interpolation)

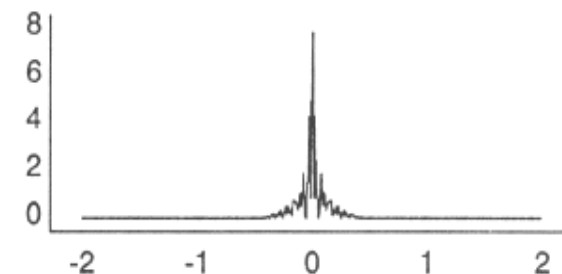
Even worse  
reconstruction



(d)



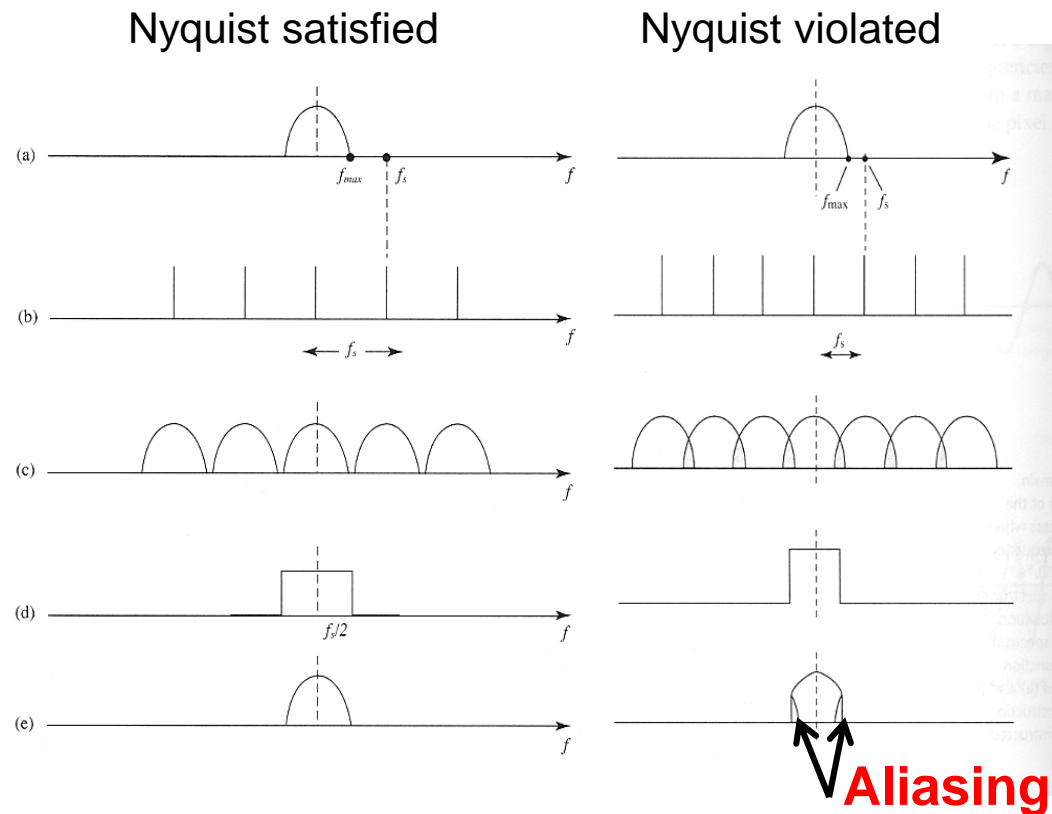
(e)



# Aliasing

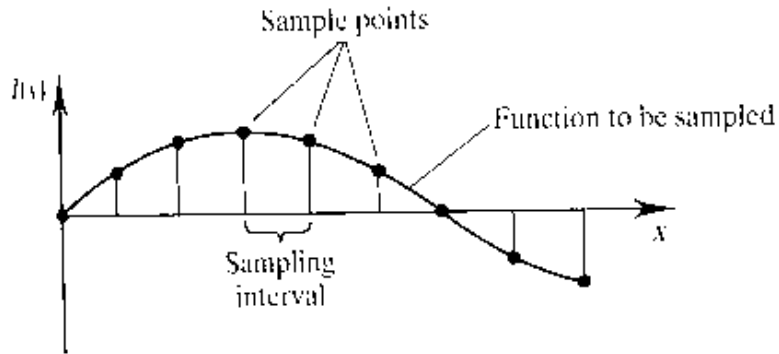
- High frequency components from the replicated copies are treated like low frequencies during the reconstruction process
- In Fourier space:

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

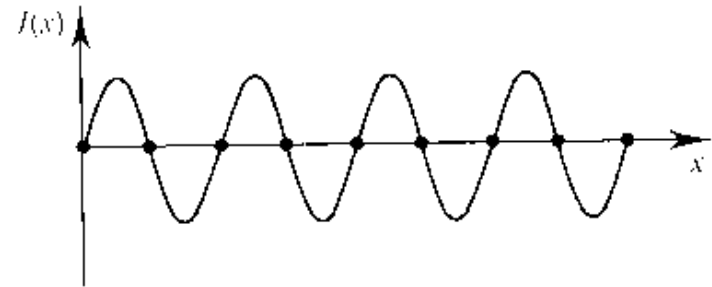


- Different signals become “aliases” when sampled

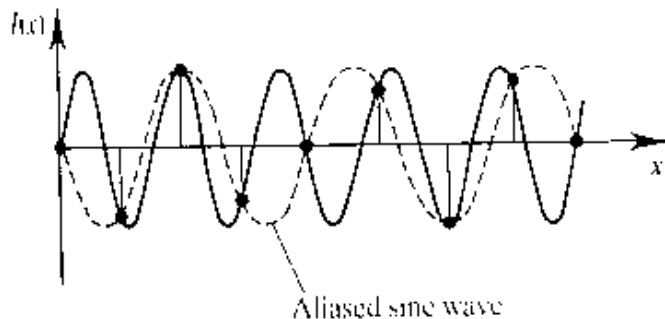
# Aliasing in 1D



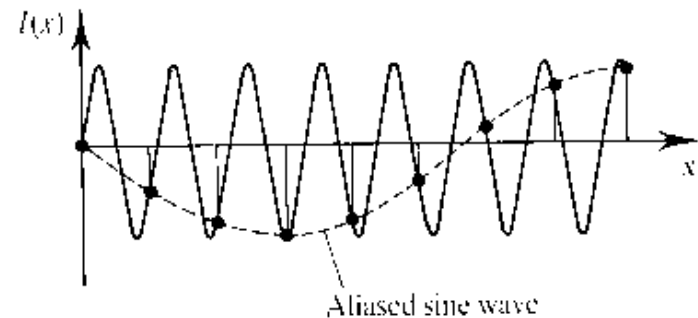
**Spatial frequency < Nyquist**



**Spatial frequency = Nyquist  
2 samples / period**



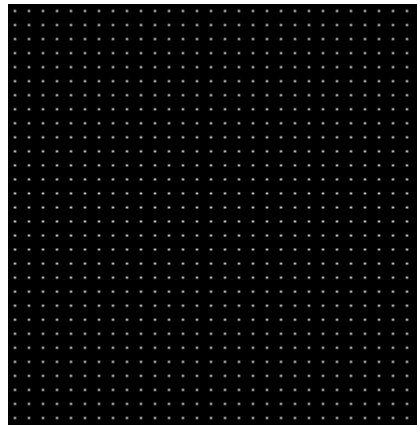
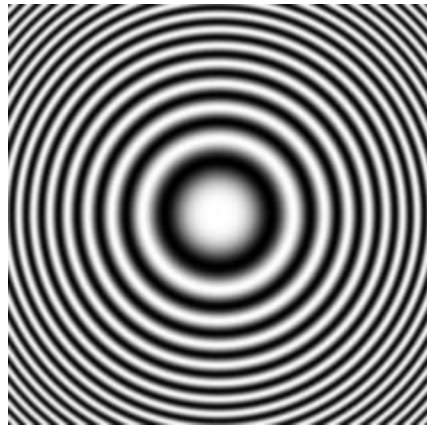
**Spatial frequency > Nyquist**



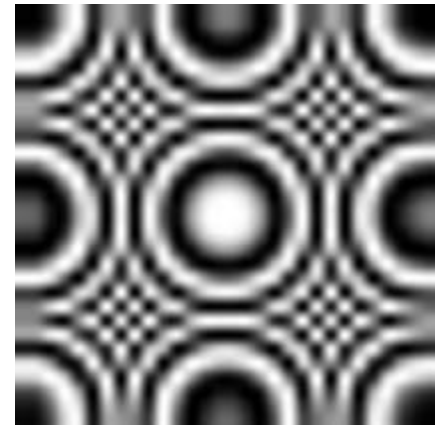
**Spatial frequency >> Nyquist**

# Aliasing in 2D

---



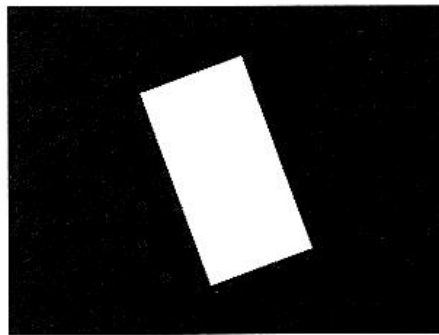
[wikipedia]



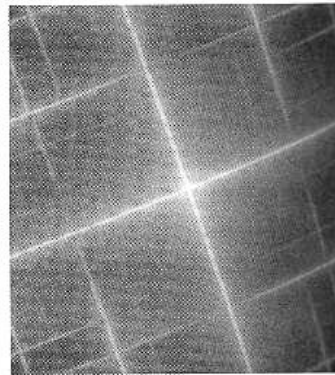
This original image sampled at these locations yields this reconstruction.

# Aliasing in 2D

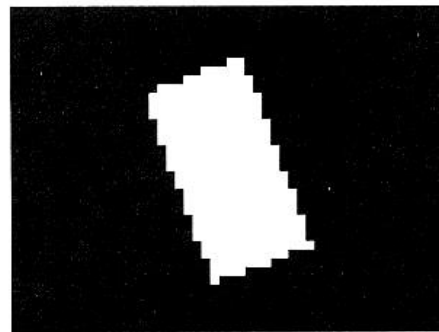
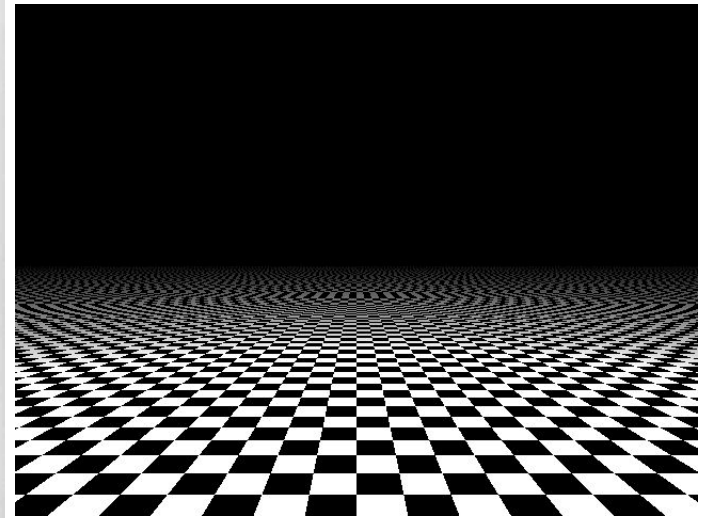
- Spatial sampling  $\Rightarrow$  repeated frequency spectrum
- Spatial conv. with box filter  $\Rightarrow$  spectral mult. with *sinc*



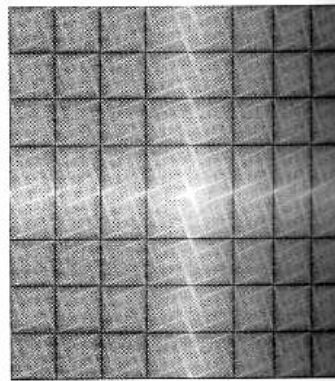
(a) Simulation of a perfect line



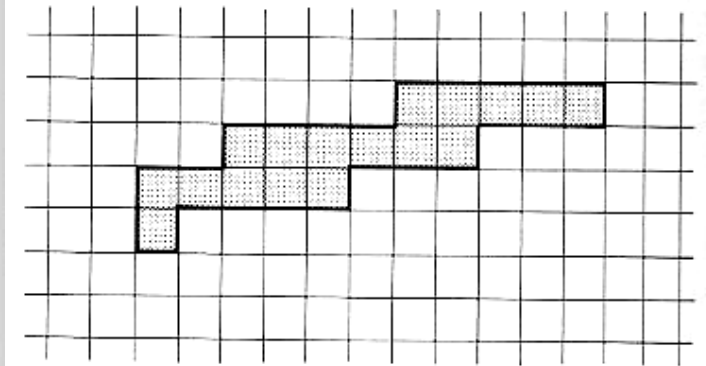
(b) Fourier transform of (a)



(c) Simulation of a jagged line



(d) Fourier transform of (c)



# Causes for Aliasing

---

- **It all comes from sampling at discrete points**
  - Multiplied with comb function
  - Comb function: repeats frequency spectrum
- **Issue when using non-band-limited primitives**
  - Hard edges → infinitely high frequencies
- **In reality, integration over finite region necessary**
  - E.g., finite CCD pixel size, integrates in the *analog domain*
- **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
- **Important:**
  - Distinction between **sampling errors** and **reconstruction errors**

# Sampling Artifacts

---

- **Spatial aliasing**
  - Stair cases, Moiré patterns (interference), etc...
- **Solutions**
  - Increasing the sampling rate
    - OK, but infinite frequencies at sharp edges
  - Post-filtering (after reconstruction)
    - Too late, does not work - only leads to blurred stair cases
  - Pre-filtering (blurring) of sharp geometry features
    - Slowly make geometry “fade out” at the edges?
    - Correct solution in principle, but blurred images might not be useful
    - Analytic low-pass filtering hard to implement
  - Super-sampling (see later)
    - On the fly re-sampling: densely sample, filter, down sample

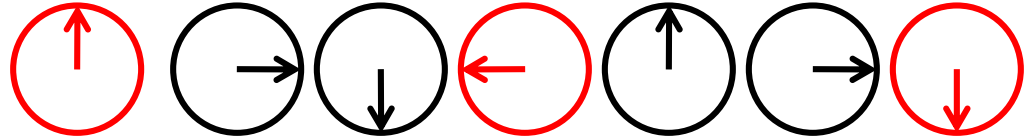


# Sampling Artifacts in 4D

---

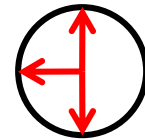
- **Temporal aliasing**

- Video of cart wheel, ...



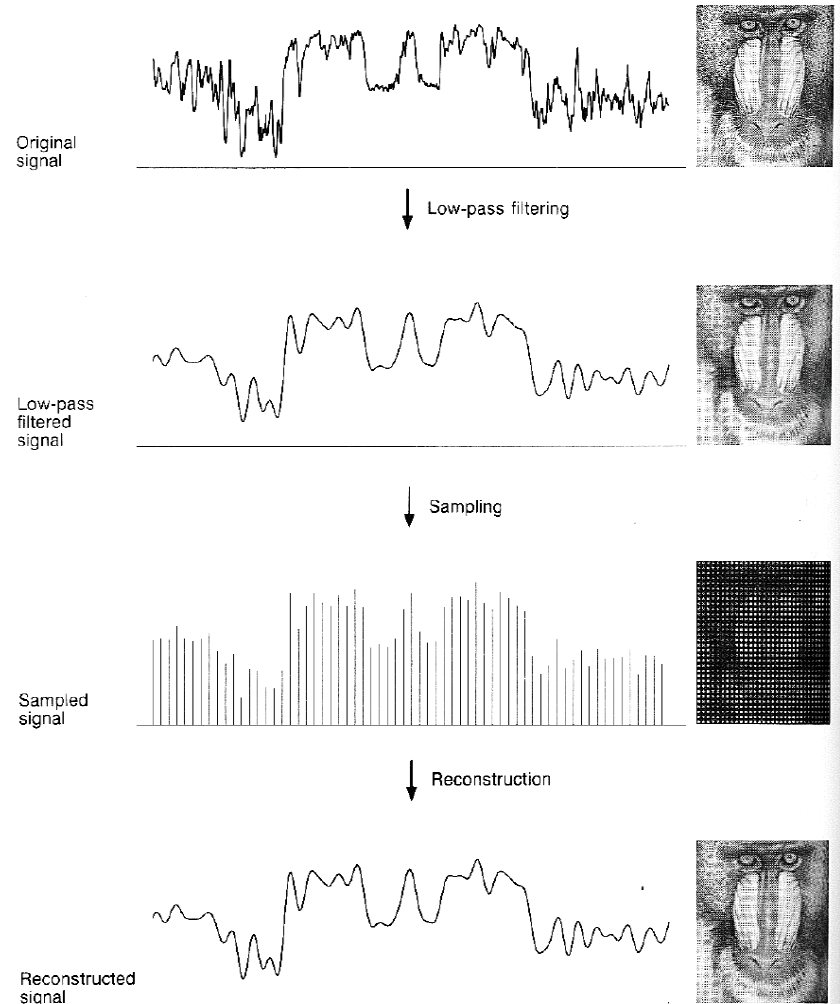
- **Solutions**

- Increasing the frame rate
  - OK
- Post-filtering (averaging several frames)
  - Does not work – only multiple details
- Pre-filtering (motion blur)
  - Should be done on the original analog signal
  - Possible for simple geometry (e.g., cartoons)
  - Problems with texture, etc...
- Super-sampling (see later)



# Antialiasing by Pre-Filtering

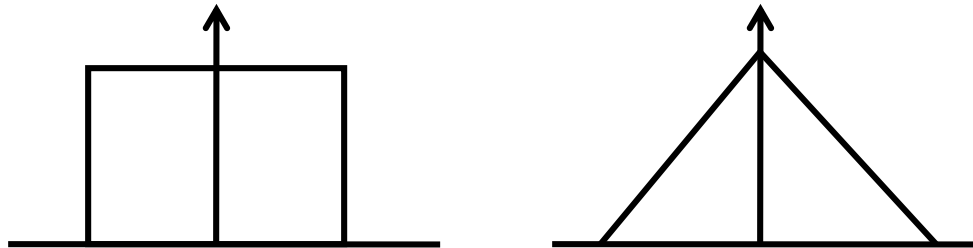
- **Filtering before sampling**
  - Analog/analytic original signal
  - Band-limiting the signal
  - Reduce Nyquist frequency for chosen sampling-rate
- **Ideal reconstruction**
  - Convolution with *sinc*
- **Practical reconstruction**
  - Convolution with
    - Box filter, Bartlett (tent)
  - Reconstruction error



# Sources of High Frequencies

---

- **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

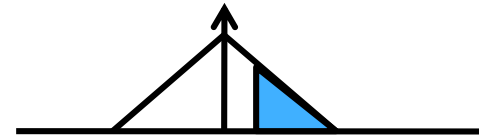
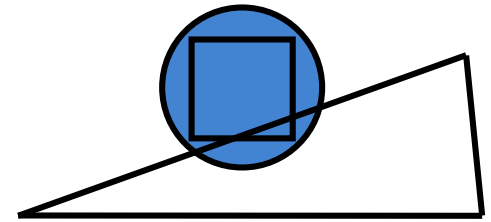


# Comparison

---

- **Analytic low-pass filtering (pixel/triangle overlap)**

- Ideally eliminates aliasing completely
- Hard to implement
  - Weighted or unweighted area evaluation
  - 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
  - Does not work at corners



- **Over-/Super-sampling**

- Very easy to implement
- Does not eliminate aliasing completely
  - Sharp edges contain **infinitely** high frequencies
- But it helps: ...

# Re-Sampling Pipeline

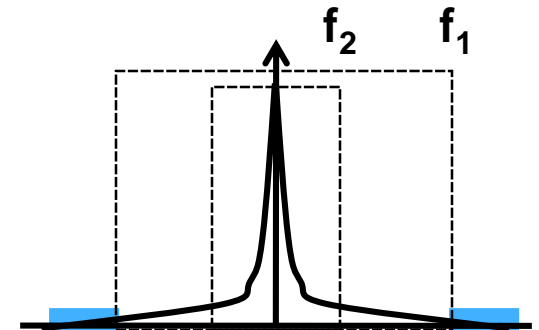
---

- **Assumption**

- Energy in higher frequencies typically decreases quickly
- Reduced aliasing by intermediate sampling at higher frequency

- **Algorithm**

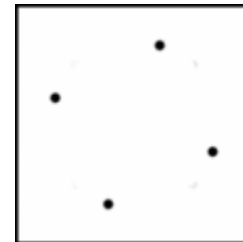
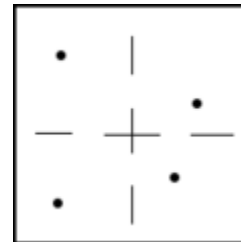
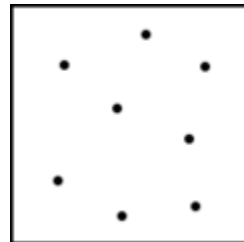
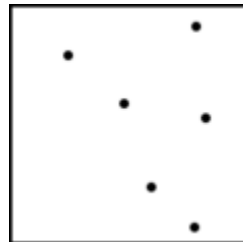
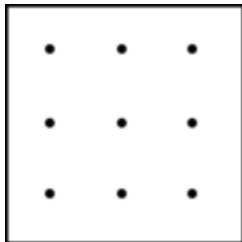
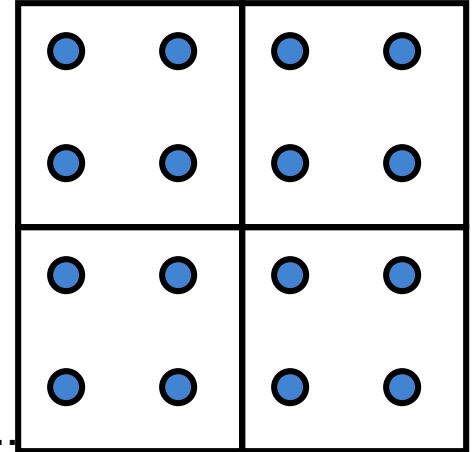
- Super-sampling
  - Sample continuous signal with high frequency  $f_1$
  - Aliasing with energy beyond  $f_1$  (assumed to be small)
- Reconstruction of signal
  - Filtering with  $g_1(x)$ : e.g. convolution with  $\text{sinc}_{f_1}$
  - Exact representation with sampled values !!
- Analytic low-pass filtering of signal
  - Filtering with filter  $g_2(x)$  where  $f_2 \ll f_1$
  - 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



# Super-Sampling in Practice

- **Regular super-sampling**

- Averaging of  $N$  samples per pixel
- $N$ : 4 (quite good), 16 (often sufficient)
- Samples: rays, z-buffer, motion, reflection, ...
- Filter weights
  - Box filter
  - Others: B-spline, pyramid (Bartlett), hexagonal, ...
- Sampling Patterns (left to right)
  - Regular: aliasing likely
  - Random: often clumps, incomplete coverage
  - Poisson Disc: close to perfect, but costly
  - Jittered: randomized regular sampling
  - Most often: rotated grid pattern



# Super-Sampling Caveats

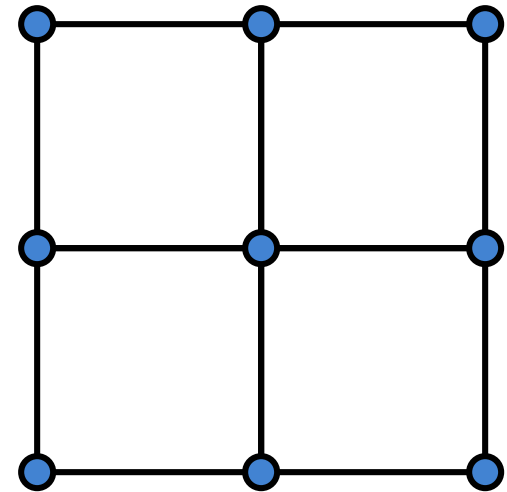
---

- **Popular mistake**

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

- **Problem**

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



- **Post-reconstruction blur**



1x1 Sampling, 3x3 Blur



1x1 Sampling, 7x7 Blur

- **There is no “free” Super-sampling**

# 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)
    - $|A-B| / (|A|+|B|)$



# Adaptive Super-Sampling

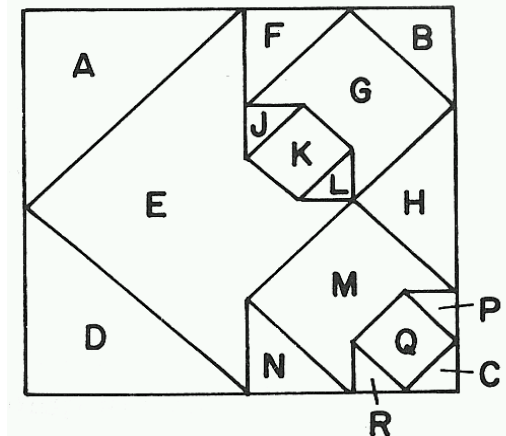
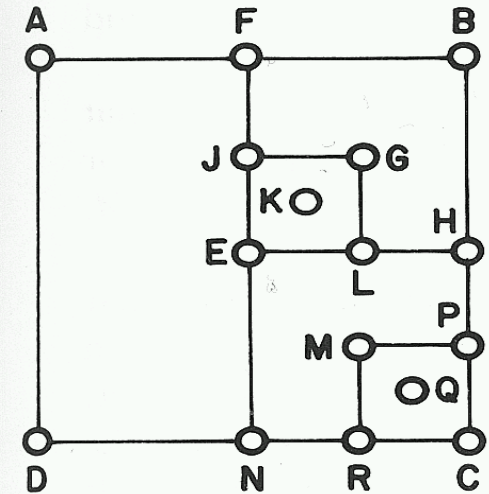
- **Recursive algorithm**

- Sampling at pixel corners and center
- Decision criterion for corner-center pairs
  - Differences, contrast, object-IDs, ray trees, ...
- Subdivide quadrant by adding 3 diag. points
- Filtering with weighted averaging
  - Tile:  $\frac{1}{4}$  from each quadrant
  - Leaf quadrant:  $\frac{1}{2}$  (center + corner)
- Box filter with final weight proport. to area  $\rightarrow$

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

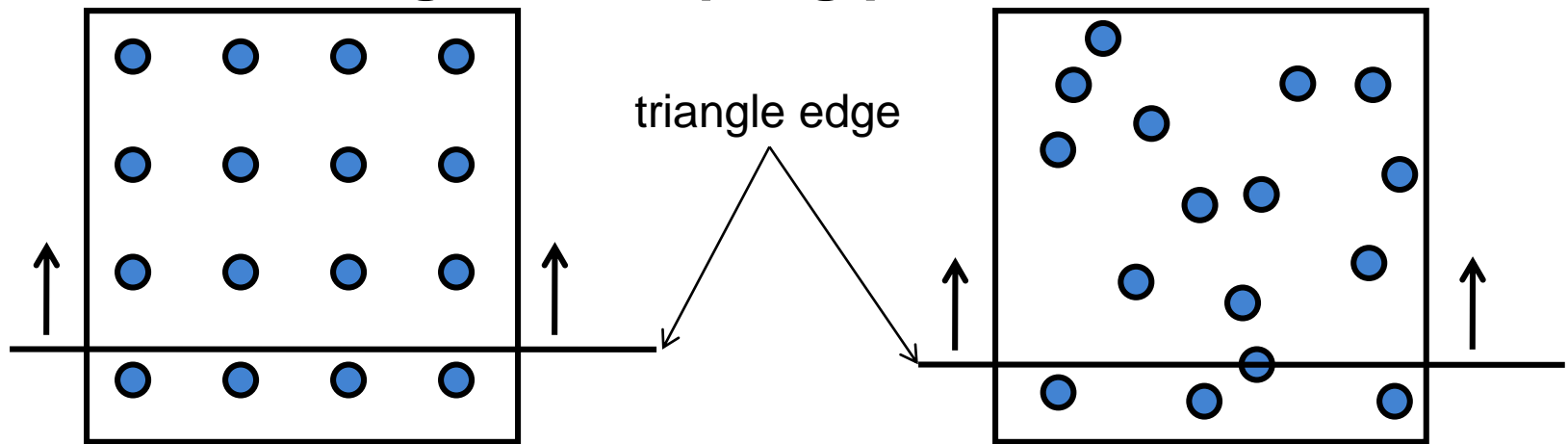
- **Extension**

- Jittering of sample points



# Stochastic Super-Sampling

- **Problems with regular super-sampling**
  - Nyquist frequency for aliasing only shifted
  - Expensive: 4-fold to 16-fold effort
  - Non-adaptive: same effort everywhere
  - Too regular: reduction of effective number of axis-aligned levels
- **Introduce irregular sampling pattern**



0  $\rightarrow$  4/16  $\rightarrow$  8/16  $\rightarrow$  12/16  $\rightarrow$  16/16: 5 levels

17 levels: better, but noisy

- **Stochastic super-sampling**
  - Or analytic computation of pixel coverage and pixel mask

# Stochastic Sampling

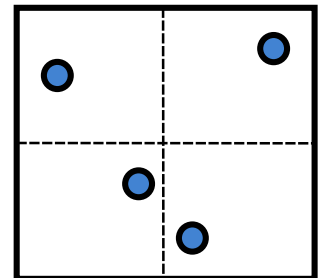
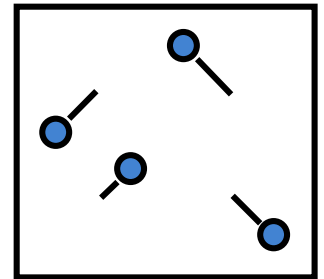
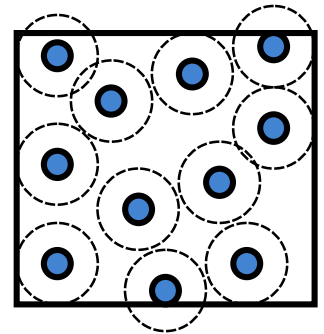
---

- **Requirements**

- Even sample distribution: no clustering
- Little correlation between positions: no alignment
- Incremental generation: on demand as needed

- **Generation of samples**

- Poisson-disk sampling
  - Random generation of samples
  - Rejection if closer than min distance to other samples
- Jittered sampling
  - Random perturbation from regular positions
- Stratified sampling
  - Subdivision into areas with one random sample in each
  - Improves even distribution
- Quasi-random numbers (Quasi-Monte Carlo)
  - E.g. Halton sequence
  - Advanced feature: see RIS course for more details



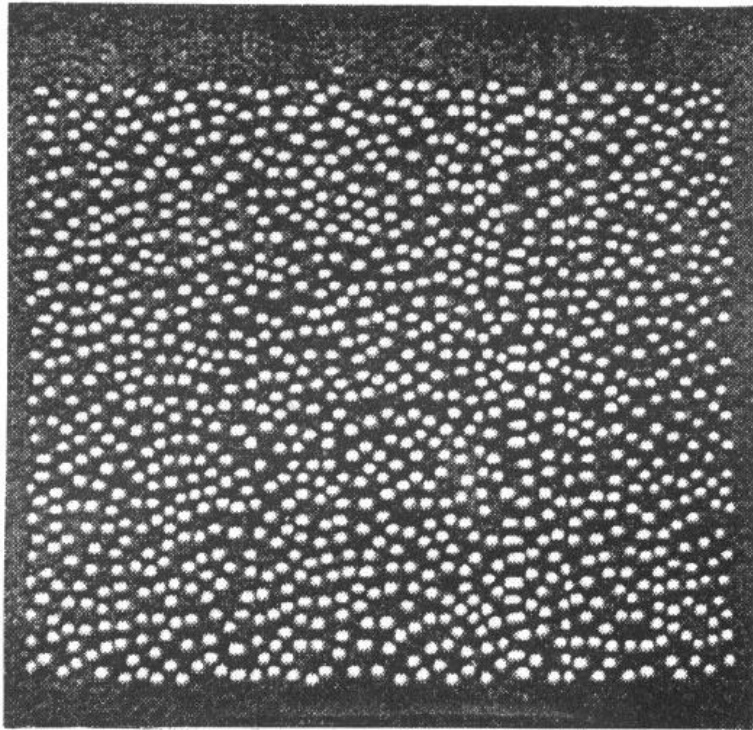
# Poisson-Disk Sample Distribut.

---

- **Motivation**

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

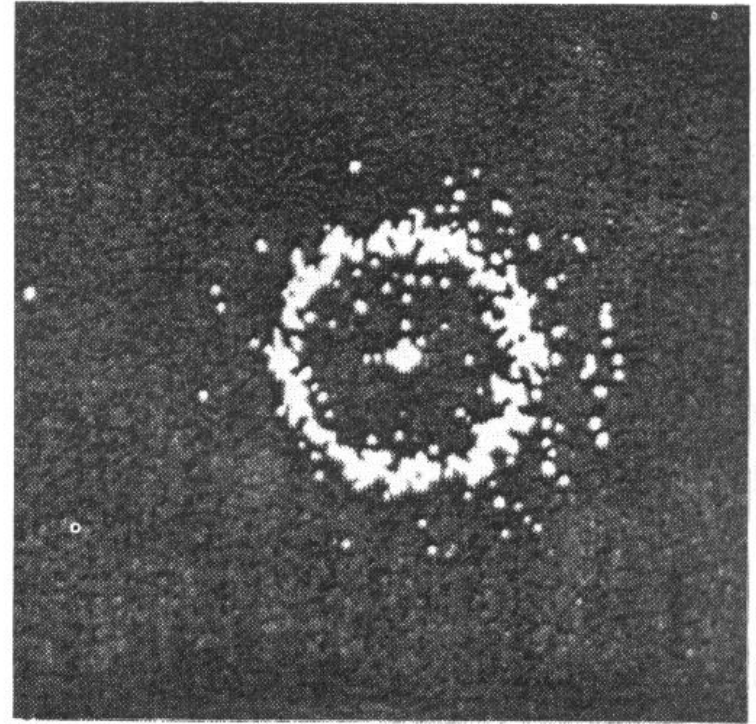
(a)



Distribution of the photo-receptors

© Andrew Glassner, Intro to Raytracing

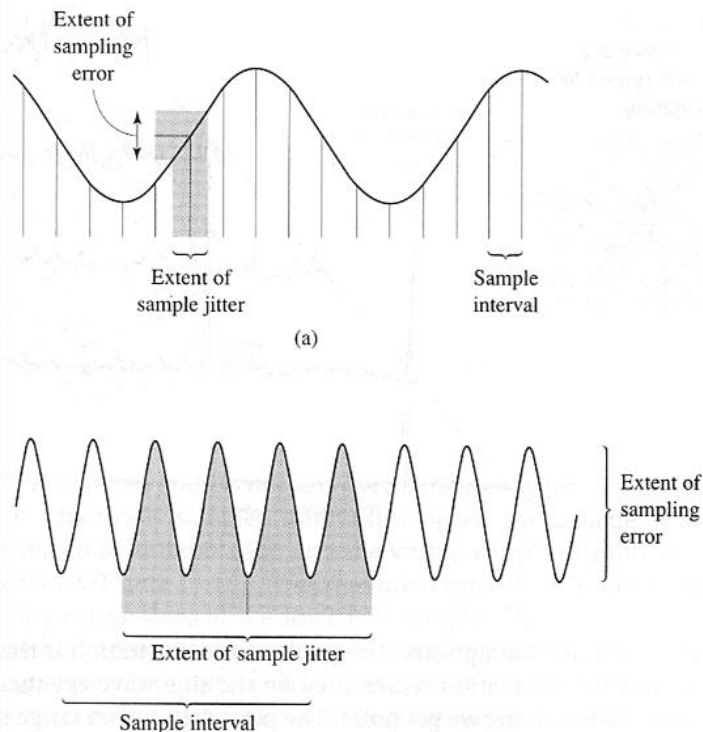
(b)



Fourier analysis

# Stochastic Sampling

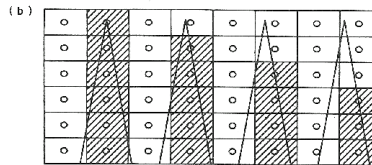
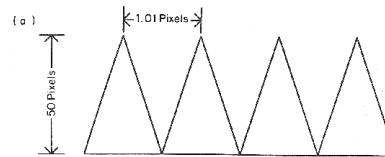
- **Slowly varying function in sample domain**
  - Closely reconstructs target value with few samples
- **Quickly varying function in sample domain**
  - Transforms energy in high-frequency bands into noise
  - Reconstructs average value as sample count increases



# Examples

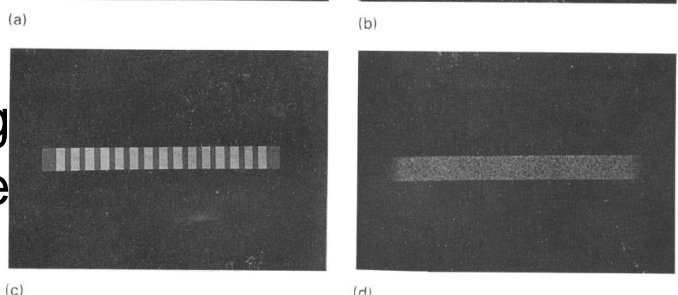
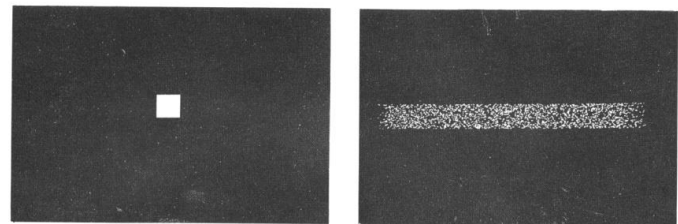
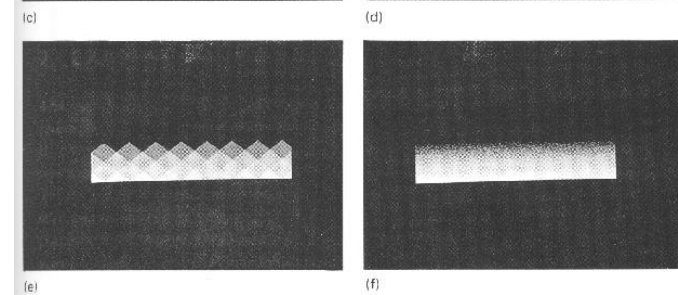
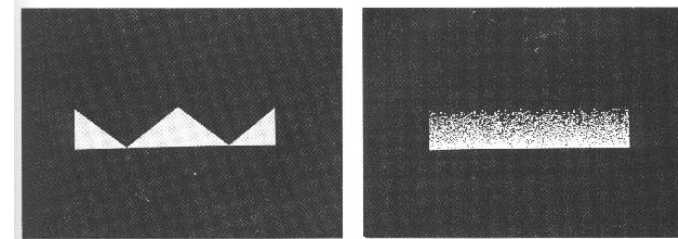
- **Spatial sampling: triangle comb**

- (c) 1 sample/pixel, no jittering: aliasing
- (d) 1 spp, jittering: noise
- (e) 16 spp, no jittering: less aliasing
- (f) 16 spp, jittering: less noise



- **Temporal sampling: motion blur**

- (a) 1 time sample, no jittering: aliasing
- (b) 1 time sample, jittering/pixel: noise
- (c) 16 samples, no jittering: less aliasing
- (d) 16 samples, jittering/pixel: less noise



(c)

(d)

(e)

(f)

(a)

(b)

(c)

(d)

# Comparison

---

- **Regular, 1x1**
- **Regular, 3x3**
- **Regular, 7x7**
- **Jittered, 3x3**
- **Jittered, 7x7**

