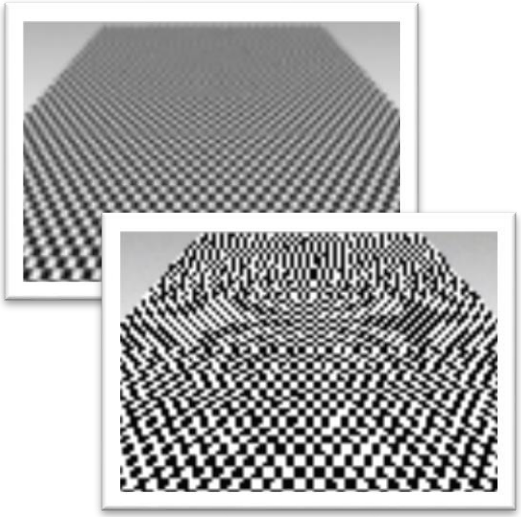
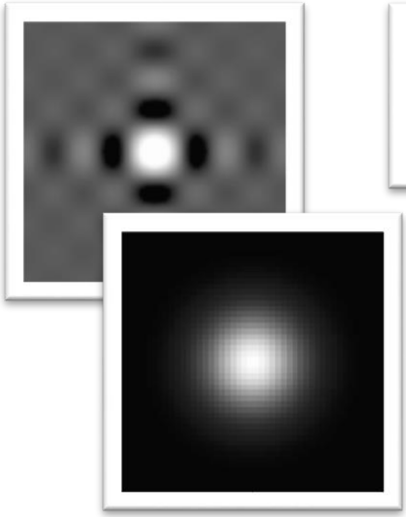
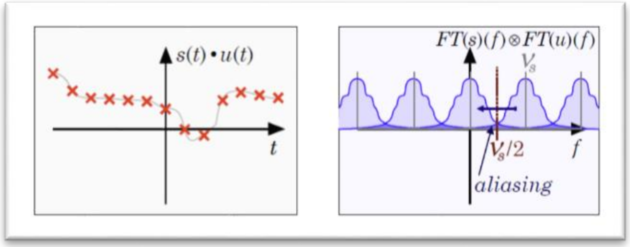
Modelling I

SUMMER TERM 2020





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



Lecture 18

Signal Theory & Sampling

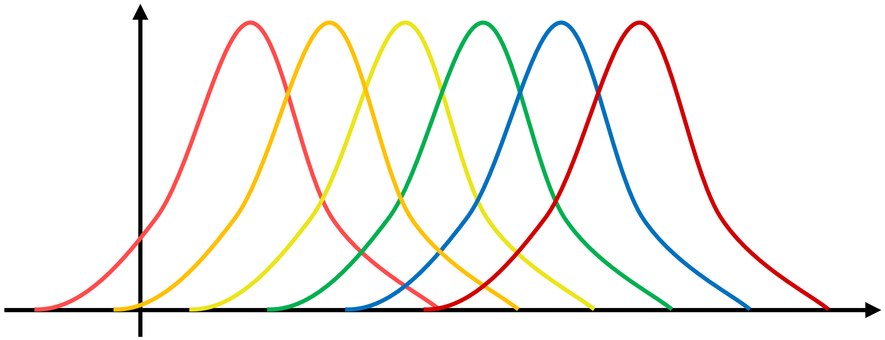
Michael Wand · Institut für Informatik · michael.wand@uni-mainz.de

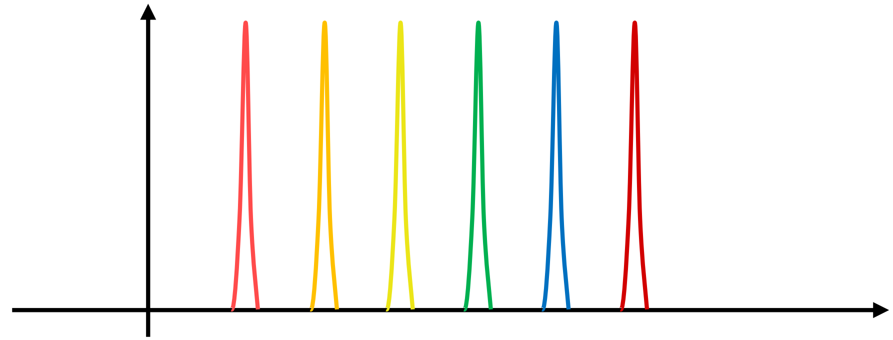
Reminder:

Constructing Bases

Frequency Space Analysis

Which of the following two is better?

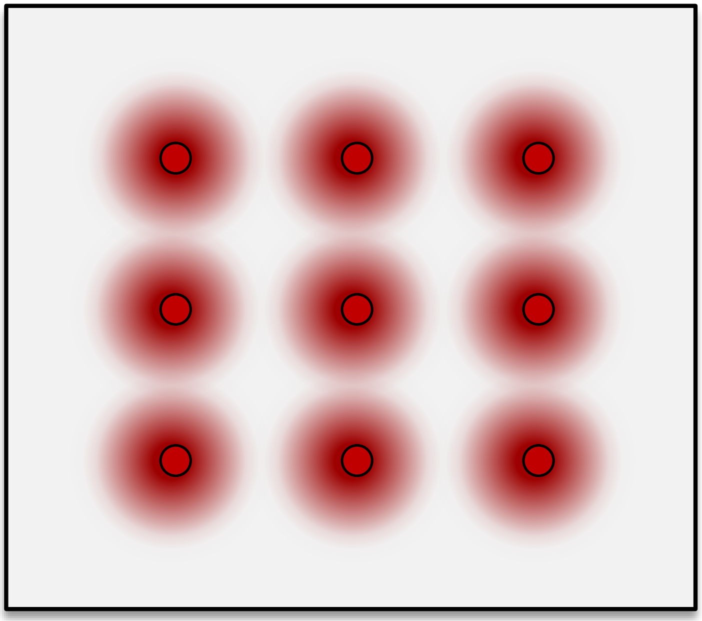




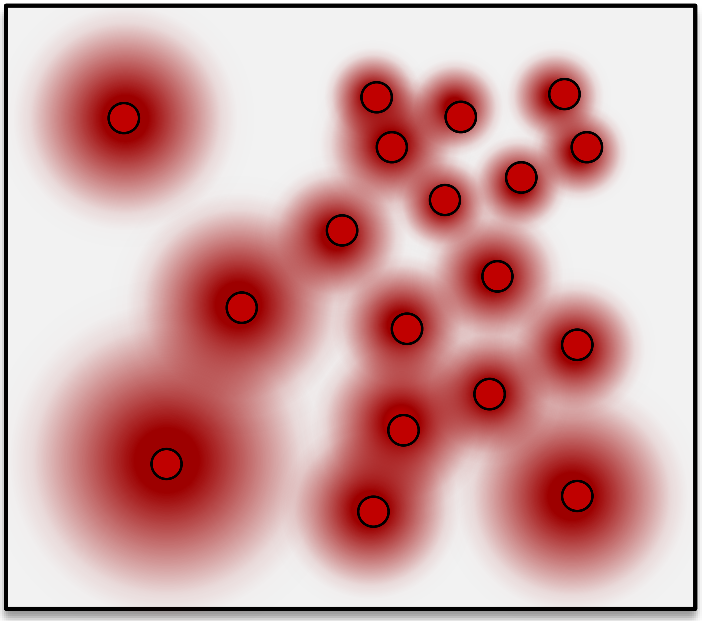
Obvious, but why?

- Long story…
  - Sampling theory
  - Fourier transforms involved
- We'll look at this ~~later~~ now.
- Also: why the "Sombrero"-style shape?

Radial Basis Functions



Regular grids



Irregular (w/scaling)

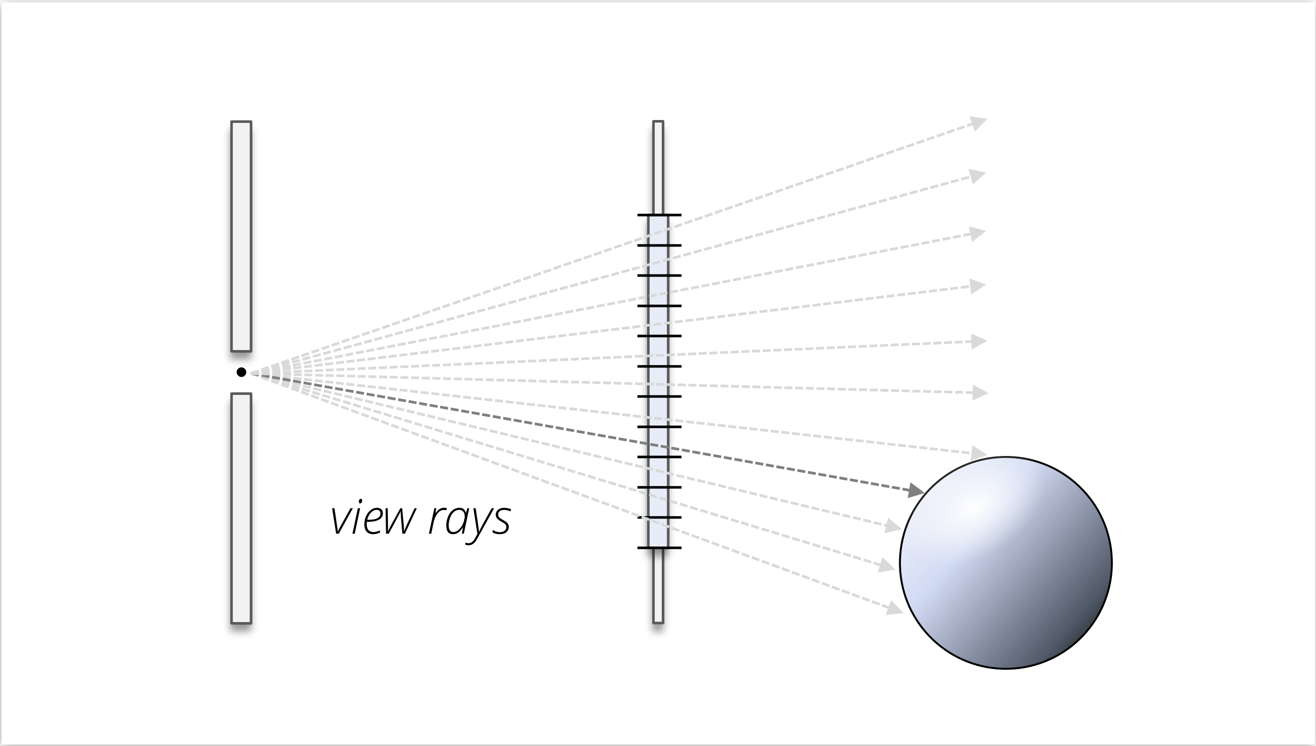
Signal Theory & Sampling

Topics

Topics

- What is the problem?
- Fourier transform
- Theorems
- Analysis of regularly sampled signals
- Irregular sampling

Model Problem: Raytracing

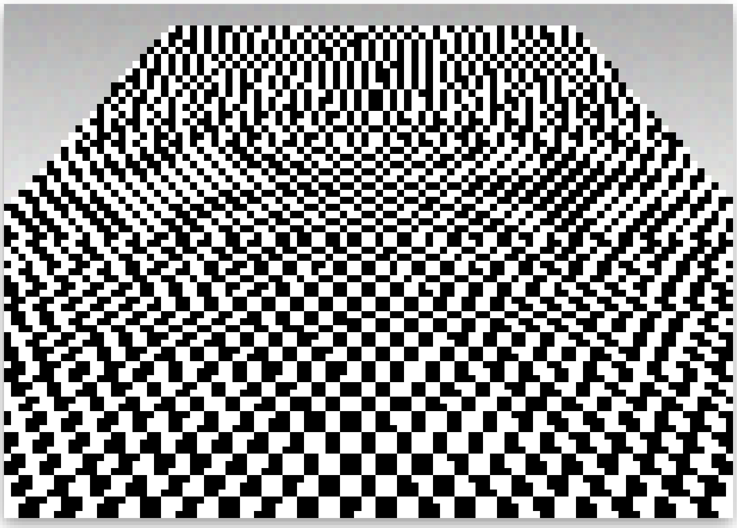


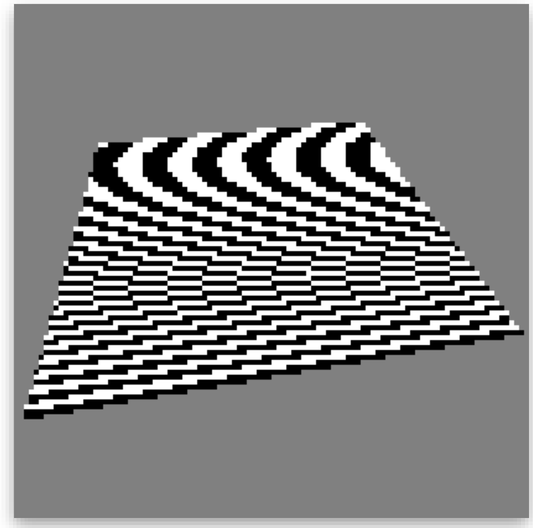
Raytracing

- Sample 3D scenes with "view rays" through each pixel

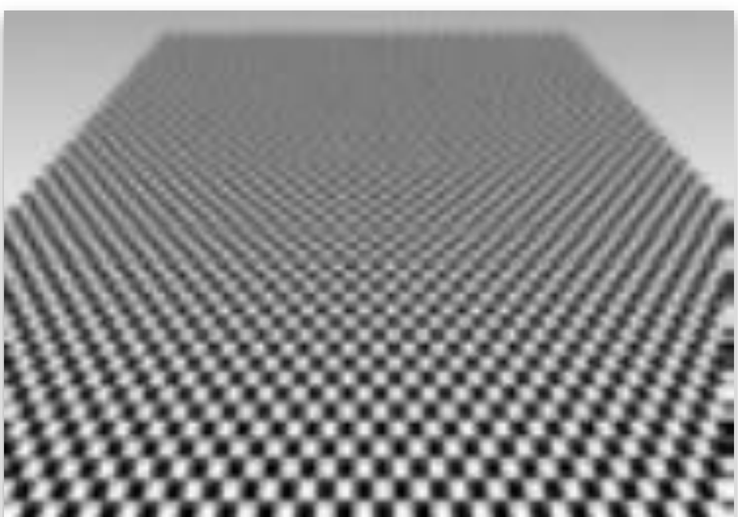
Sampling Aliasing

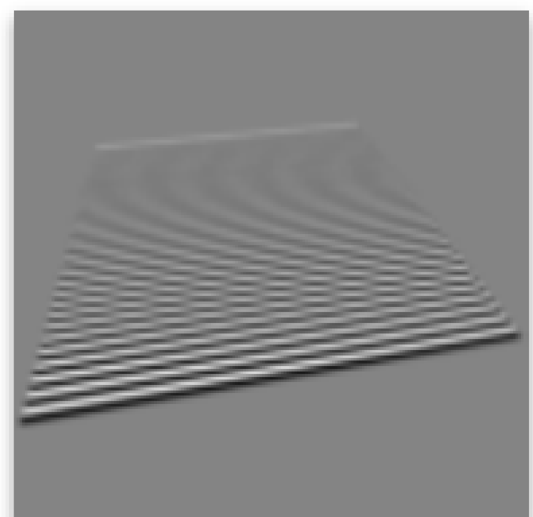
1 sample / pixel



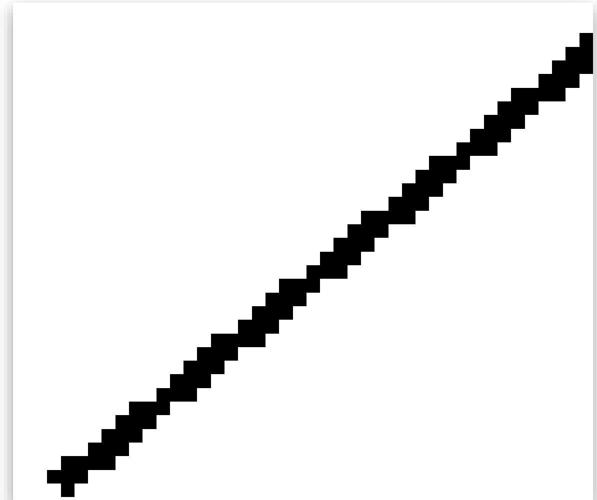


Integrating with
Gaussian weights

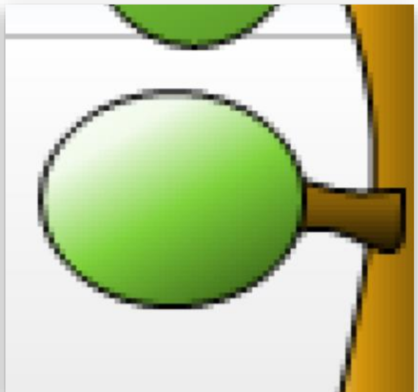




Sampling Text



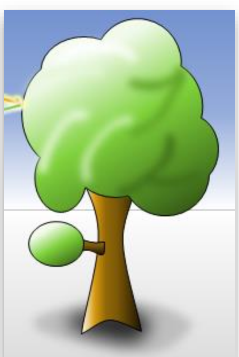
Reconstruction Aliasing



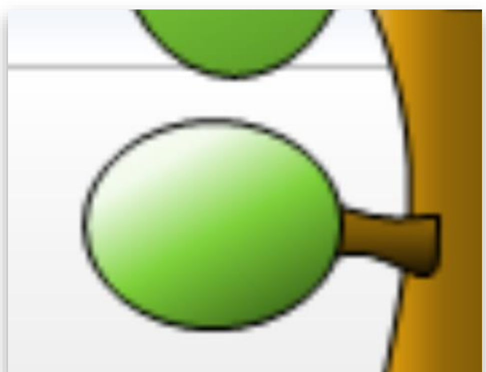
pixel

Gaussian

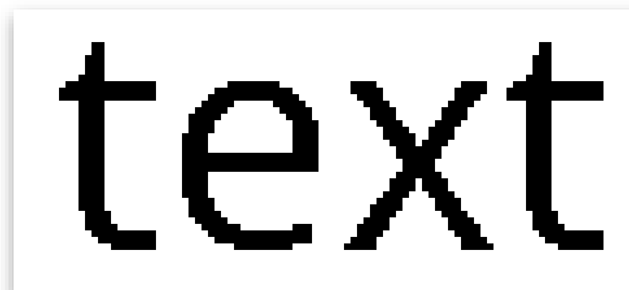
Lanczos



image



bilinear



magnified pixels

Aliasing – the Short Story

Sampling Aliasing

- Sampling a signal inadequately
  - Detail information shows up “under false name”
  - Too-high-frequency details → low-frequency moiré
- Need to understand sampling requirements

Reconstruction Aliasing

- Low-detail signal is reconstructed with unwarranted high-frequency details
- Need to understand reconstruction process

Rendering: Crucial for quality + efficiency

Underlying Question

Deeper question underlying this

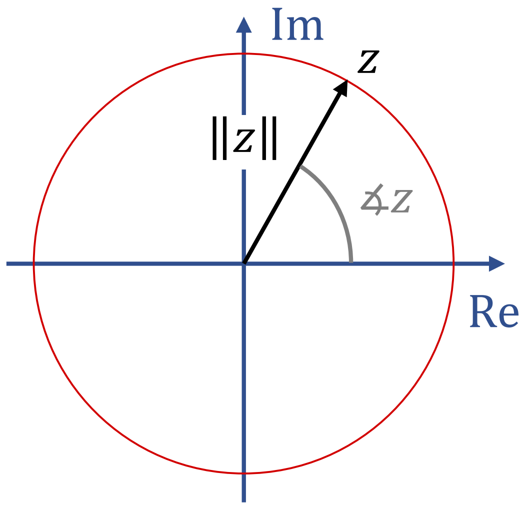
- How much information is in a function?

Complex Numbers

Complex Numbers

Vector space $\mathbb{R}^2$:

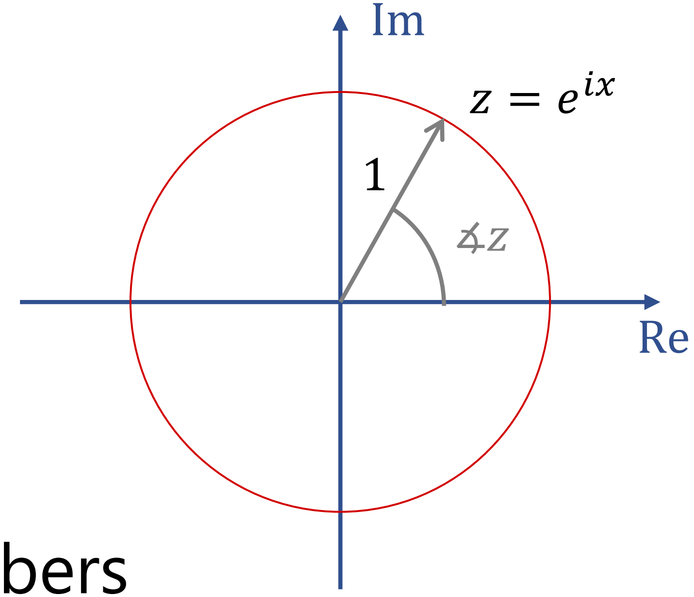
- $z \in \mathbb{C} \rightarrow z = (x, y) =: x + iy$
- i is the upward basis vector (0,1)
  - i introduces the y-axis
- Unlike $\mathbb{R}$: Unordered!



Additional multiplication

- Multiplying complex numbers $z_1 \cdot z_2$:
  - Multiply length
  - Add angles
  - This makes $i = \sqrt{-1}$

Complex exponential



Complex exponentials:

- Powers of imaginary numbers
  = rotating vectors
- Euler's formula:

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

Real Fourier Series

Fourier Basis

Fourier basis (orthonormal)

$$B = \left\{1, \sqrt{2}\sin 2\pi kx, \sqrt{2}\cos 2\pi kx \;\middle|\; k \in \mathbb{N}^{\geq 1}\right\}$$

Fourier series

- Periodic functions $f: [0,1] \to \mathbb{R}$
- Fourier series approximation:

$$\tilde{f}(x) = b_0 + \sqrt{2}\sum_{k=0}^{\infty}[a_k \sin 2\pi kx + b_k \cos 2\pi kx]$$

- Coefficients?

Fourier Basis

Fourier series

- Fourier series:

$$\tilde{f}(x) = b_0 + \sqrt{2} \sum_{k=0}^{\infty} [a_k \sin 2\pi kx \ + b_k \cos 2\pi kx]$$

- Coefficients?

$$a_k = \langle f(x), \sqrt{2} \sin 2\pi kx \rangle = \sqrt{2} \int_0^1 f(x) \cdot \sin 2\pi kx \, dx$$

$$b_k = \langle f(x), \sqrt{2} \cos 2\pi kx \rangle = \sqrt{2} \int_0^1 f(x) \cdot \cos 2\pi kx \, dx$$

$$b_0 = \langle f(x), 1 \rangle = \int_0^1 f(x) dx$$

- Convergence?

Fourier Series

Fourier Series

- Converges for functions
  - Finite variation
  - Lipschitz-smooth
- Convergence means:

$$\lim_{k\to\infty}\left\|f-\tilde{f}\right\|^2=\lim_{k\to\infty}\left\langle f-\tilde{f},f-\tilde{f}\right\rangle=0$$

Complex
Fourier Series

Fourier Basis

Fourier basis (real):

$$B_{\mathbb{R}} = \left\{1, \sqrt{2}\sin 2\pi kx, \sqrt{2}\cos 2\pi kx \,\middle|\, k \in \mathbb{N}\right\}$$

Fourier basis (complex):

$$B_{\mathbb{C}} = \{\exp(2\pi ikx) \,|\, k \in \mathbb{Z}\}$$

Complex Series

Fourier series

- Fourier series:

$$\tilde{f}(x) = \sum_{k=-\infty}^{\infty} z_k \exp(2\pi i k x)$$

- Coefficients?

$$z_k = \langle f(x),\ \exp(-2\pi i k x)\rangle$$

$$= \int_0^1 f(x) \cdot \exp(-2\pi i k x)\, dx$$

Hermetian space

Tip: 3BLUE1BROWN – But what is a Fourier series? From heat flow to circle drawings
https://www.youtube.com/watch?v=r6sGWTCMz2k

Scalar Product on Real Function Spaces

Real (finite-dim.) Vector Spaces

- For $\mathbf{z}, \mathbf{q} \in \mathbb{R}^d$: $\langle \mathbf{z}, \mathbf{q} \rangle \coloneqq \mathbf{z}^T \mathbf{q}$

Real Function Spaces

- For suitable*) functions

$$f, g: \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$$

the *standard scalar product* is defined as:

$$f \cdot g = \langle f, g \rangle \coloneqq \int_\Omega f(x) \cdot g(x)\, dx$$

- Measures an *norm* and *angle* in an abstract sense

*) square-integrable

Complex Function Spaces

Hermetian Vector Space

- For $\mathbf{z}, \mathbf{q} \in \mathbb{C}^d$: $\langle \mathbf{z}, \mathbf{q} \rangle := \mathbf{z}^T \overline{\mathbf{q}}$

$$z \;= a + ib$$
$$\bar{z} := a - ib$$

Hermetian Function Space

- For suitable functions

$$f, g: \Omega \subset \mathbb{R} \to \mathbb{C}$$

the *standard scalar product* is defined as:

$$f \cdot g = \langle f, g \rangle := \int_\Omega f(x) \cdot \overline{g(x)}\, dx$$

- Measures an *norm* and *angle* in an abstract sense

Fourier Transform

Fourier Transform

Continuous transform:

- Continuous function set: $\{e^{-i2\pi\omega x} \mid \omega \in \mathbb{R}\}$
  - Orthogonal on $\mathbb{R}$
  - Projection via scalar products $\Rightarrow$ Fourier transform
- Fourier transform: $(\mathrm{f}: \mathbb{R} \to \mathbb{C}) \;\to\; (\mathrm{F}: \mathbb{R} \to \mathbb{C})$

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

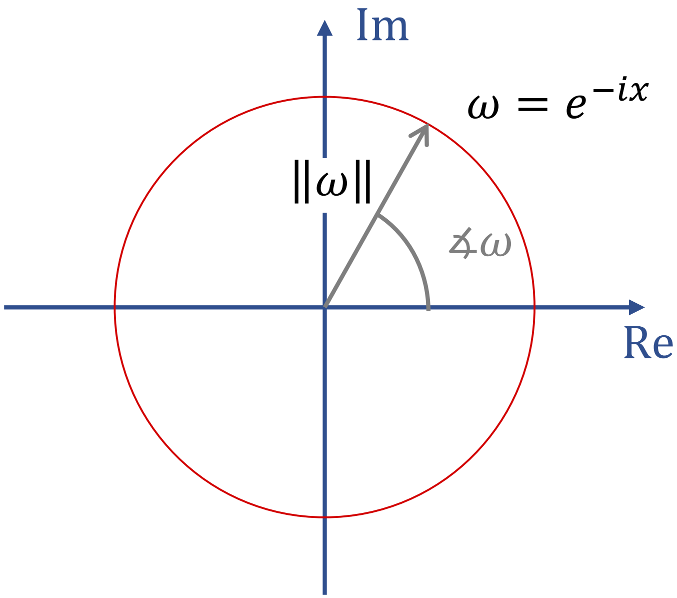
- Inverse Fourier transform: $(\mathrm{F}: \mathbb{R} \to \mathbb{C}) \to (\mathrm{f}: \mathbb{R} \to \mathbb{C})$

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

Fourier Transform

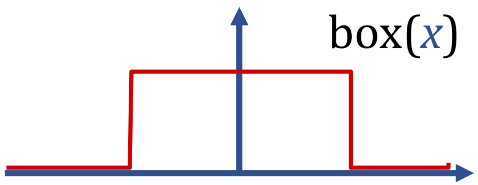
Interpreting the result:

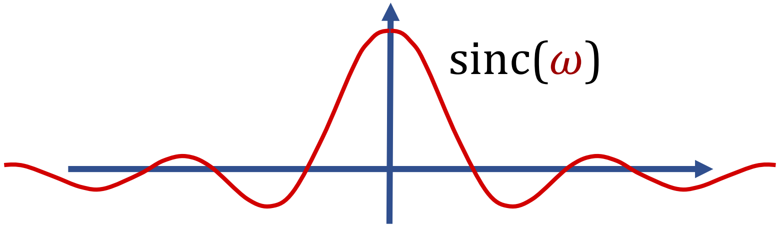
- Transforming a real function
  $f(x)\colon \mathbb{R} \to \mathbb{R}$
- Result: $\mathrm{F}(\omega)\colon \mathbb{R} \to \mathbb{C}$
  - ω are frequencies (real)
- Real input f: Symmetric result
  $\mathrm{F}(-\omega) = \mathrm{F}(\omega)$
- Output are complex numbers
  - Magnitude: “power spectrum” (frequency content)
  - Phase: phase spectrum (encodes shifts)



Important Functions

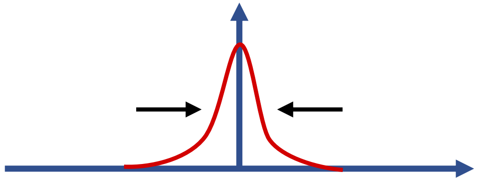
Some important Fourier-transform pairs

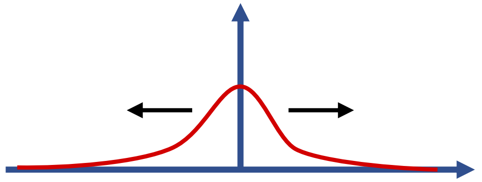




- Box function:

$$f(x) = \text{box}(x) \quad \rightarrow \quad F(\omega) = \frac{\sin \omega}{\omega} := \text{sinc}(\omega)$$



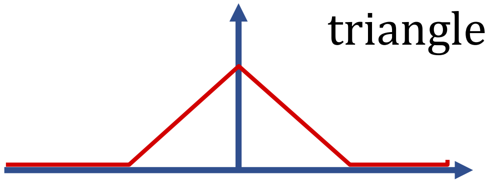


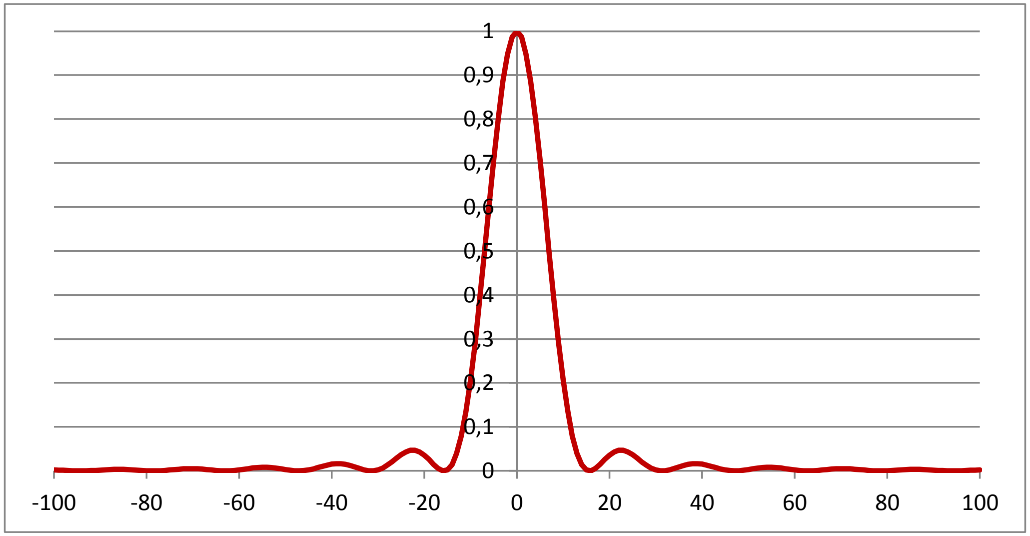
- Gaussian:

$$f(x) = e^{-ax^2} \quad \rightarrow \quad F(\omega) = \sqrt{\frac{\pi}{a}} \cdot e^{-\frac{(\pi\omega)^2}{a}}$$

Triangle Function

Bilinear Interpolation





$$f(x) = \text{triangle}(x) \quad \rightarrow \quad F(\omega) = \frac{\sin^2 \omega}{\omega^2} := \text{sinc}^2(\omega)$$

Higher Dimensional FT

Multi-dimensional Fourier Basis:

- Functions $f: \mathbb{R}^d \to \mathbb{C}$
- 2D Fourier basis:

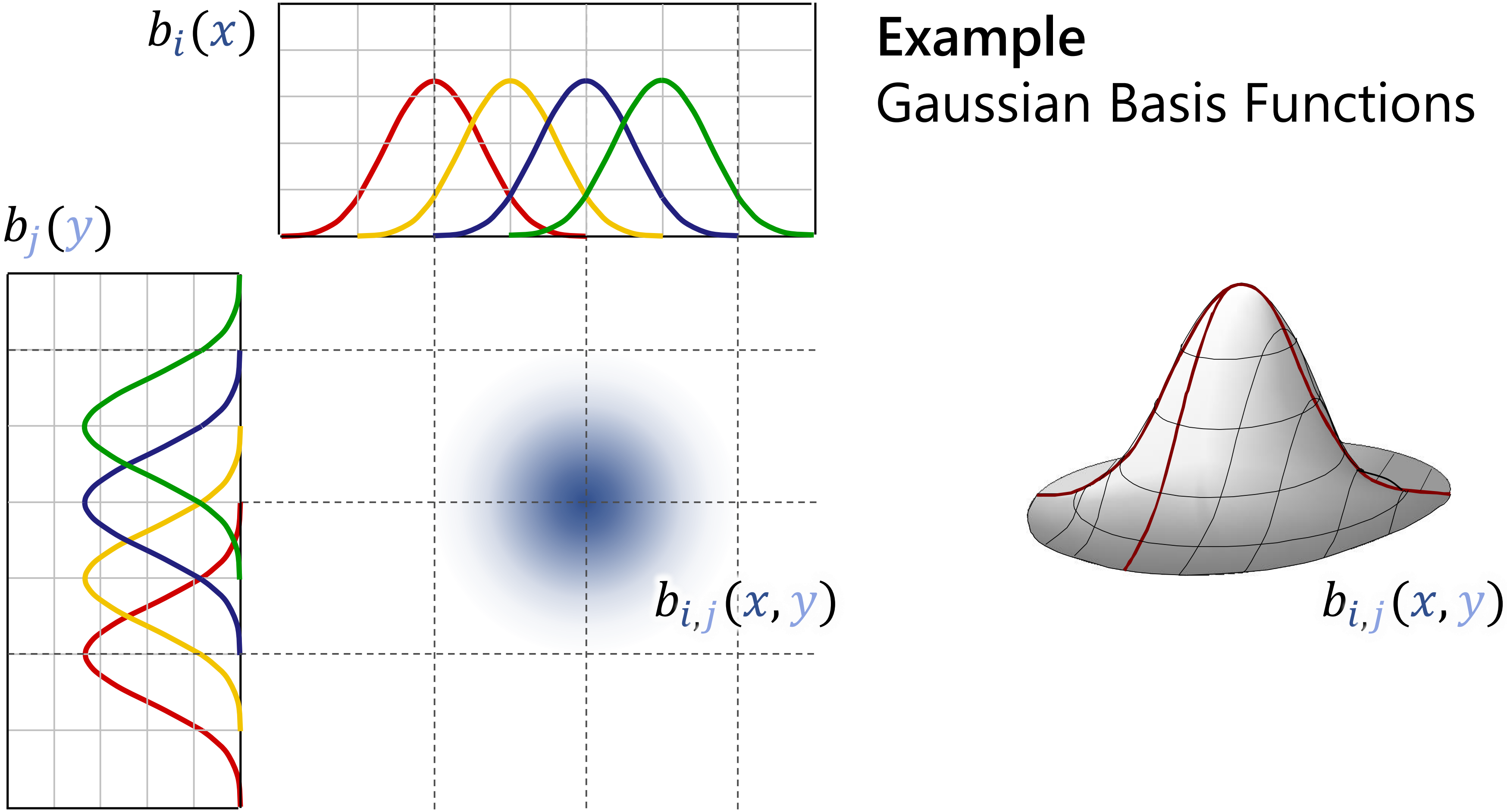
$f(x, y)$ represented
as combination of

$$\{e^{-i2\pi\omega_x x} \cdot e^{-i2\pi\omega_y y} \mid \omega_x, \omega_y \in \mathbb{R}\}$$

- In general:
  - All combinations of 1D functions
  - „Tensor product basis“
  - $b_{i,j}(x, y) = b_i(x) \cdot b_j(y)$

Tensor Product

Example
Gaussian Basis Functions

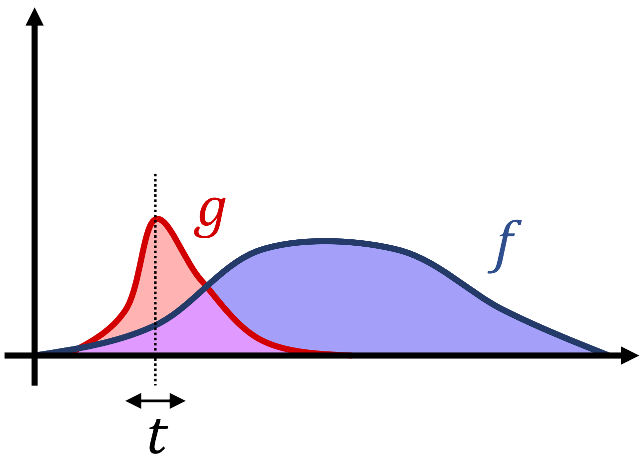


Convolution

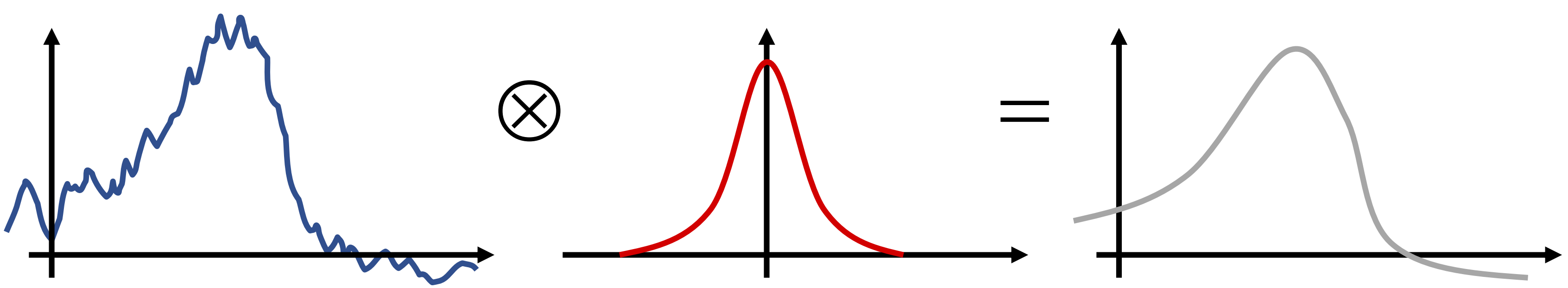
Convolution:

- Weighted average of functions
- Definition:

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



Example:



Theorems

Convolution theorem:

- Fourier Transform converts convolution into multiplication

$$FT(f \otimes g) = F \cdot G$$

Theorems

Convolution theorem:

- Fourier Transform converts convolution into multiplication

$$FT(f \otimes g) = F \cdot \mathrm{G}$$

All other cases as well

- $FT^{-1}(F \cdot G) = f \otimes g$
- $FT(f \cdot g) = F \otimes \mathrm{G}$
- $FT^{-1}(F \otimes G) = f \cdot g$
- (Formally: Fourier basis diagonalizes shift-invariant linear operators)

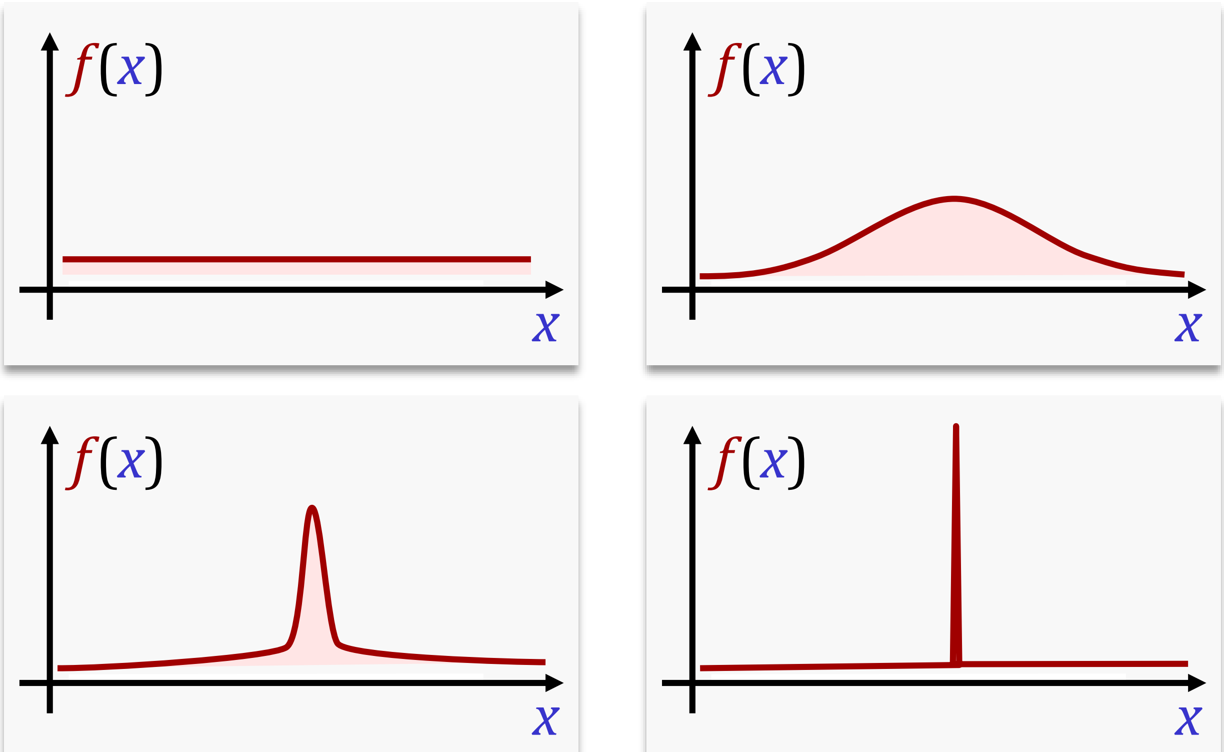
Signal Theory

Sampling a Signal

Given:

- Signal $f: \mathbb{R} \to \mathbb{R}$
- Store digitally:
  - Sample regularly … $f(0.3), f(0.4), f(0.5)$ …
- Question: what information is lost?

Delta Function

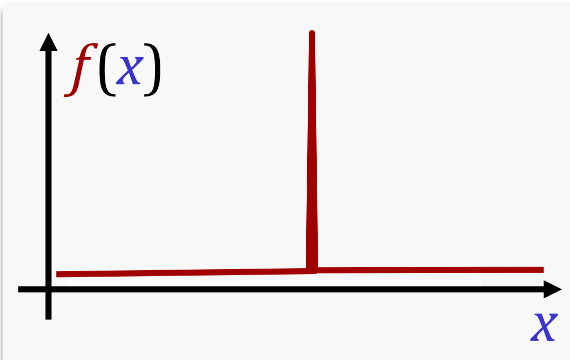


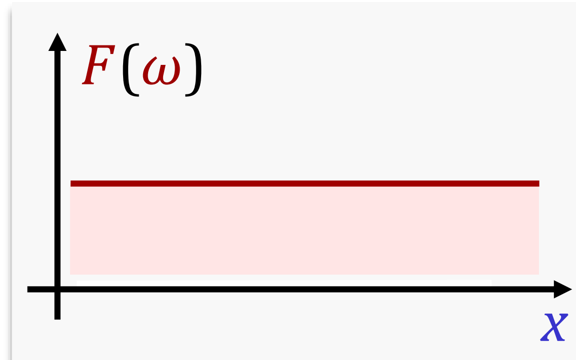
$$\int_{\Omega} f(x)dx = 1$$

Dirac Delta "Function"

- $\int_{\mathbb{R}} \delta(x)dx = 1$, zero everywhere but at $x = 0$
- Idealization ("distribution") – think of very sharp peak

Fourier Transform



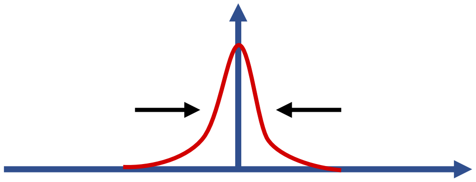


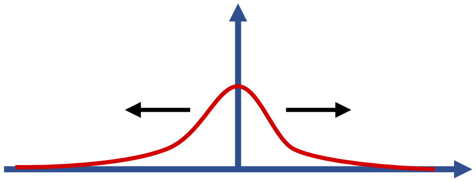
Fourier Transform Pair

- Dirac delta function ↔ uniform spectrum…
- …and vice versa.

Important Functions

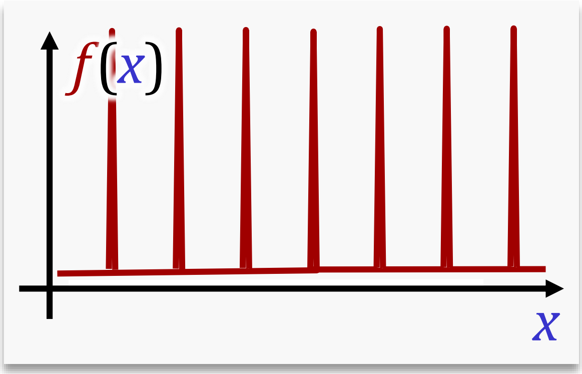
Intuition: Gaussians

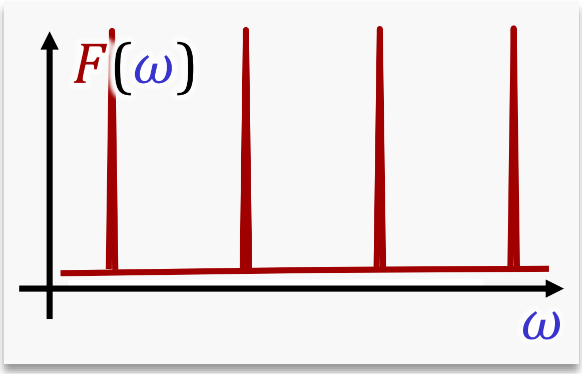




$$f(x) = e^{-ax^2} \quad \rightarrow \quad F(\omega) = \sqrt{\frac{\pi}{a}} \cdot e^{-\frac{(\pi\omega)^2}{a}}$$

Dirac Comb (Impulse Train)





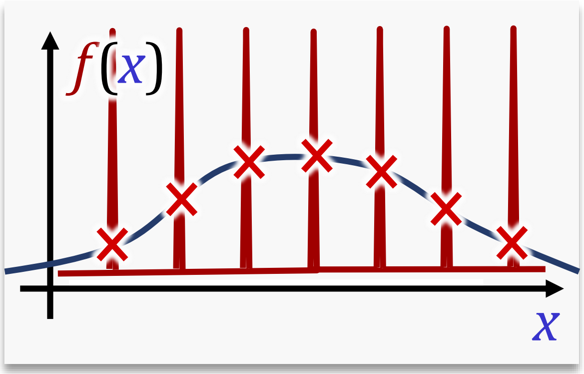
Impulse Train

$$\text{III}_T(x) = \sum_{k=-\infty}^{\infty} \delta(x - k \cdot T)$$

Fourier Transform

$$FT(\text{III}_T) = \frac{1}{T}\text{III}_{1/T}$$

Sampling

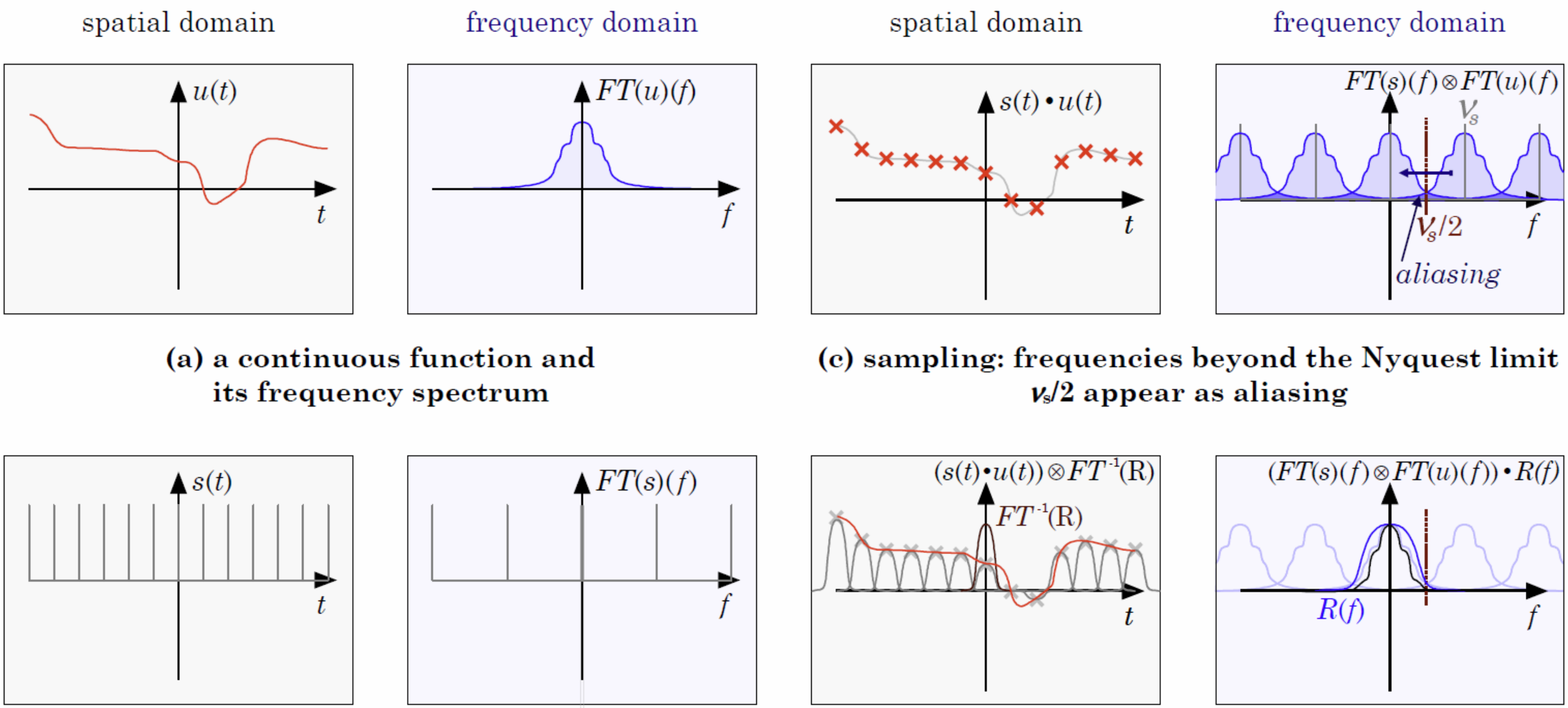


Sampling a function

- Multiplication with impulse train

$$f_{sampled}(x) = f(x) \cdot \text{III}_T(x)$$

Sampling & Reconstruction



(a) a continuous function and its frequency spectrum

(c) sampling: frequencies beyond the Nyquest limit $\nu_s/2$ appear as aliasing

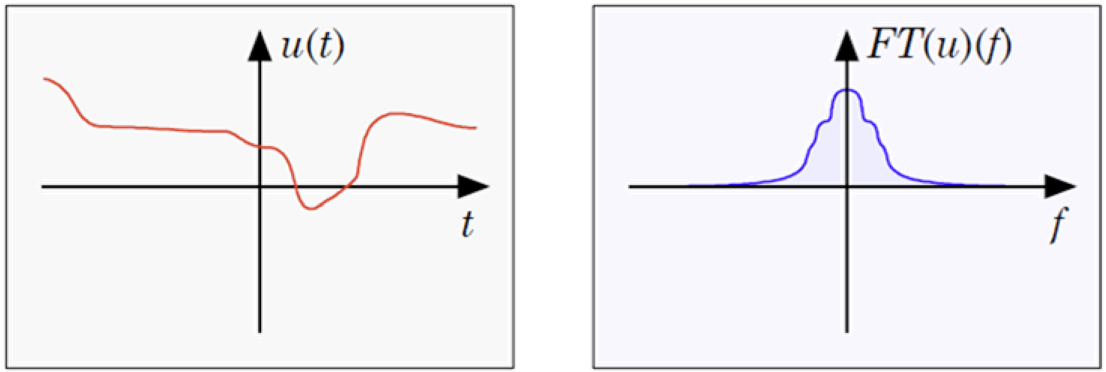
(b) a regular sampling pattern (impulse train) and its frequency spectrum

(d) reconstruction: filtering with a low-pass filter *R* to remove replicated spectra

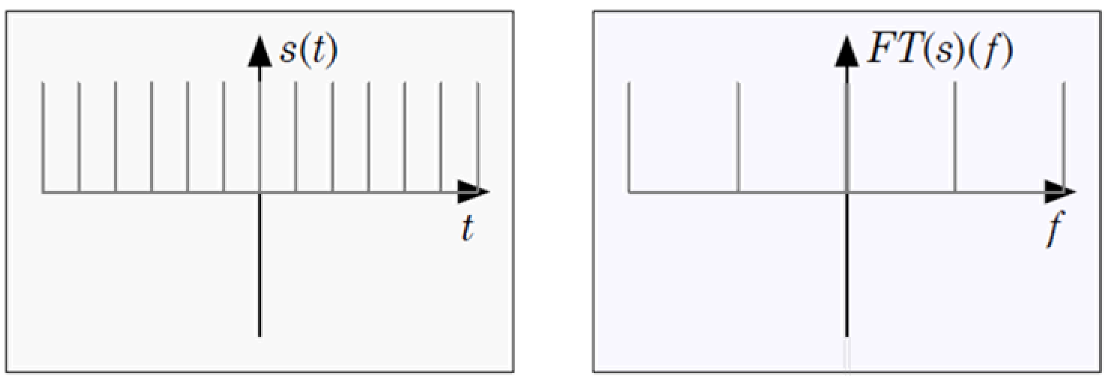
Reference: Foley, van Dam, Feiner, Hughes
Computer Graphics - Principles & Practice, 2nd Edition, Addisson-Wesley, 1996
Chapter 14.10 "Aliasing and Antialiasing"

Sampling a Signal

spatial domain frequency domain

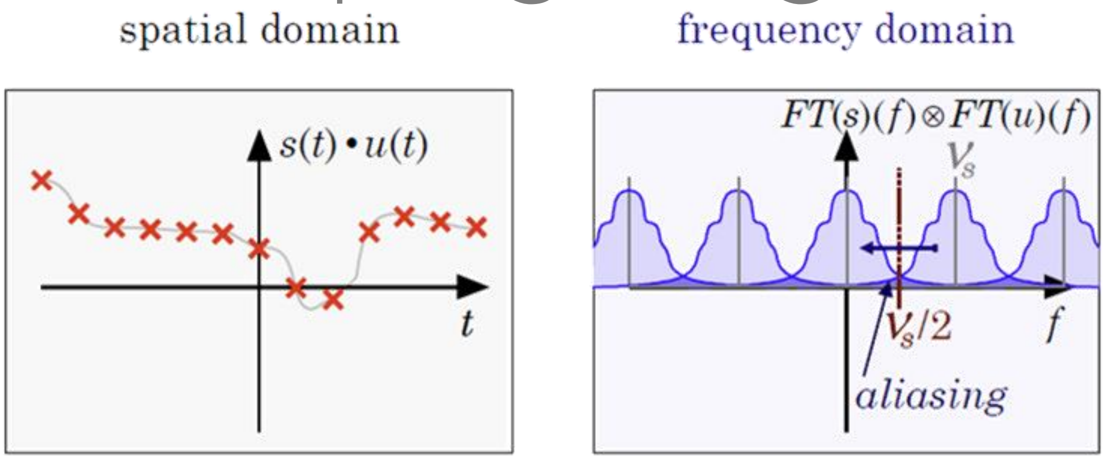


(a) a continuous function and its frequency spectrum



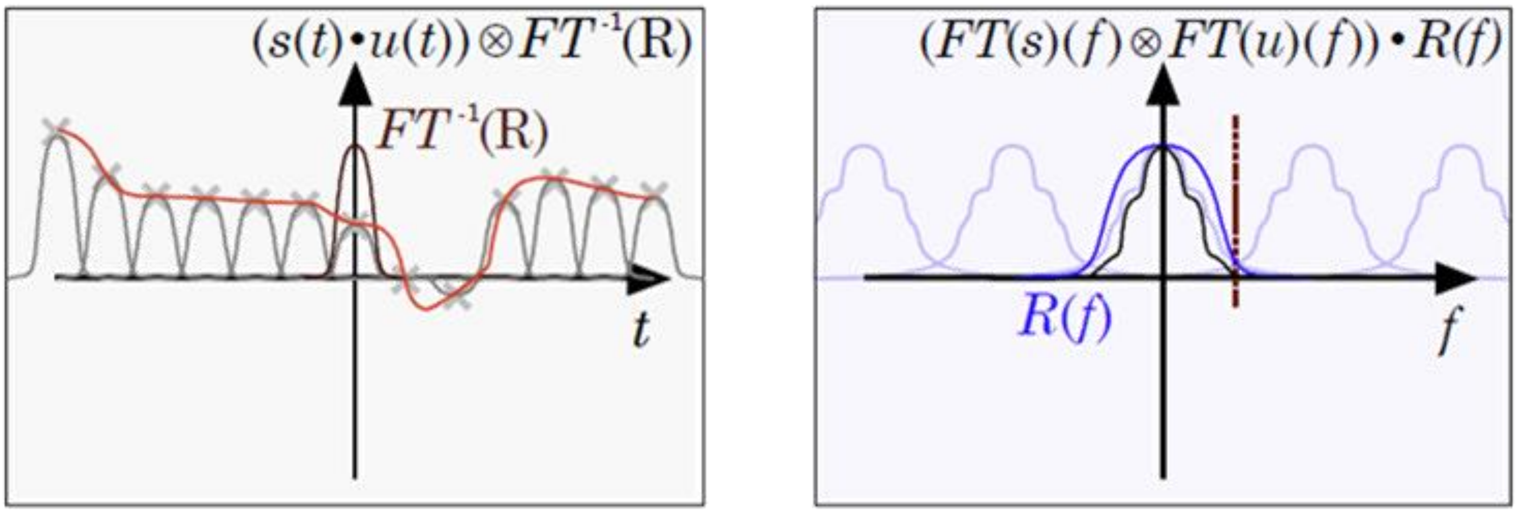
(b) a regular sampling pattern (impulse train) and its frequency spectrum

Sampling a Signal



(c) sampling: frequencies beyond the Nyquest limit $\nu_s/2$ appear as aliasing

Reconstructing a Signal

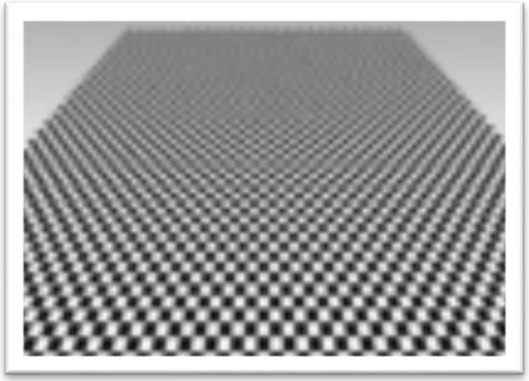


(d) reconstruction: filtering with a low-pass filter *R* to remove replicated spectra

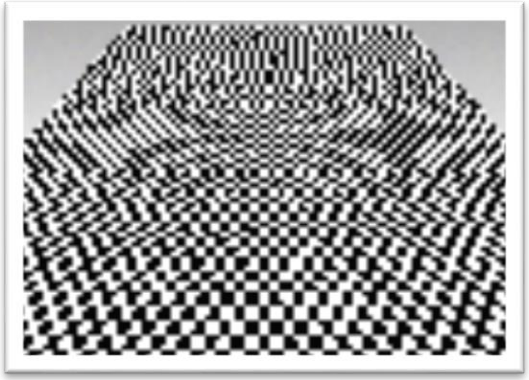
Regular Sampling

Results: Sampling

- Band-limited signals can be represented exactly
  - Sampling with frequency ν_s:
    Highest frequency in Fourier spectrum $\leq \nu_s/2$
- Higher frequencies *alias*
  - Aliasing artifacts (low-frequency patterns)
  - Cannot be removed after sampling (loss of information)



band-limited

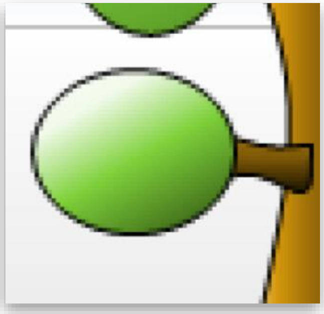


aliasing

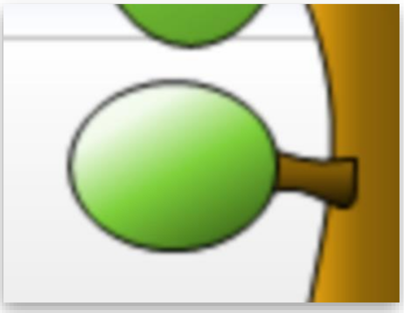
Regular Sampling

Result: Reconstruction

- When reconstructing from discrete samples
- Use band-limited basis functions
  - Highest frequency in Fourier spectrum $\leq \nu_s/2$
  - Otherwise: Reconstruction aliasing



pixel

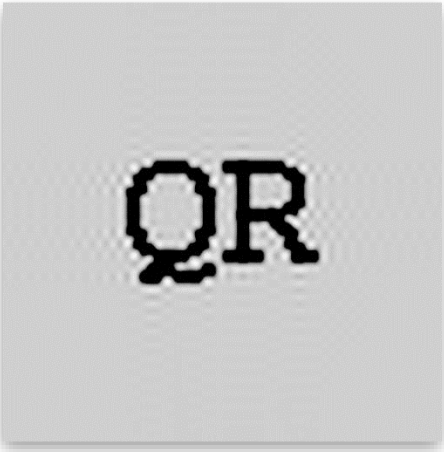


Gaussian

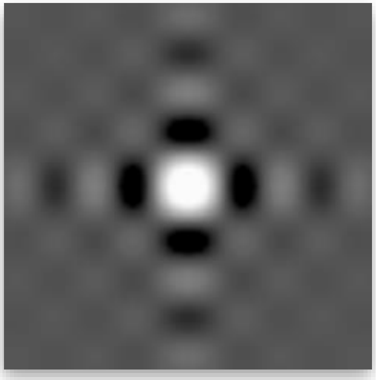
Regular Sampling

Reconstruction Filters

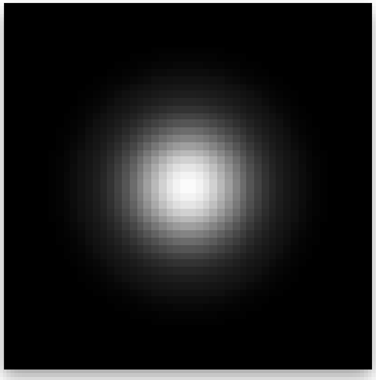
- Optimal filter: sinc
  (no frequencies discarded)
- However:
  - Ringing artifacts in spatial domain
  - Not useful for images
    (better for audio)
- Compromise
  - Gaussian filter
    (most frequently used)
  - There exist better ones,
    such as Mitchell-Netravalli,
    Lancos, etc...



Ringing by sinc reconstruction from [Mitchell & Netravali, Siggraph 1988]



2D sinc



2D Gaussian

Irregular Sampling

Irregular Sampling

Irregular Sampling

- No comparable formal theory
- However: similar idea
  - Band-limited by "sampling frequency"
  - Sampling frequency = mean sample spacing
    - Not as clearly defined as in regular grids
    - May vary locally (adaptive sampling)
- Aliasing
  - Random sampling creates noise as aliasing artifacts
  - Evenly distributed sample concentrate noise in higher frequency bands in comparison to purely random sampling

Consequences

When designing bases for function spaces

- Use band-limited functions
- Typical scenario:
  - Regular grid with spacing σ
  - Grid points $\mathbf{g}_i$
  - Use functions: $\exp\left(-\frac{(\mathbf{x}-\mathbf{g}_i)^2}{\sigma^2}\right)$
- Irregular sampling:
  - Same idea
  - Use estimated sample spacing instead of grid width
  - Set σ to average sample spacing to neighbors

Random Sampling

Random sampling

- Aliasing gets replaced by noise
- Can we optimize this? – Yes!

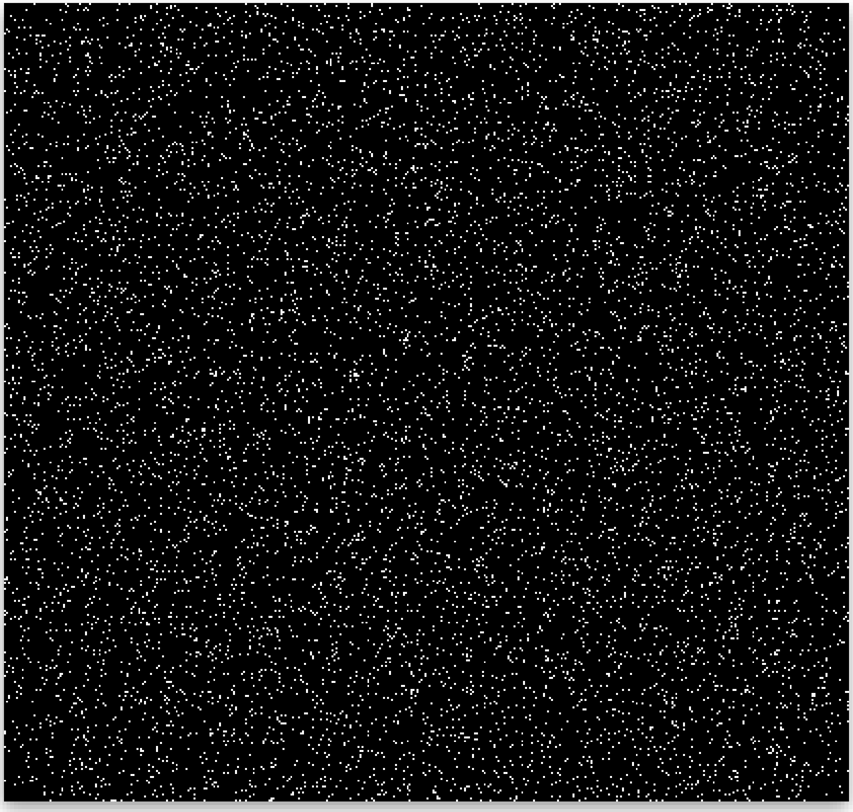
Different types of noise

- “White noise”: All frequencies equally likely
- “Blue noise”: Pronounced high-frequency content

Depends on sampling

- Random sampling is “white”
- Poisson-disc sampling (uniform spacing) is “blue”

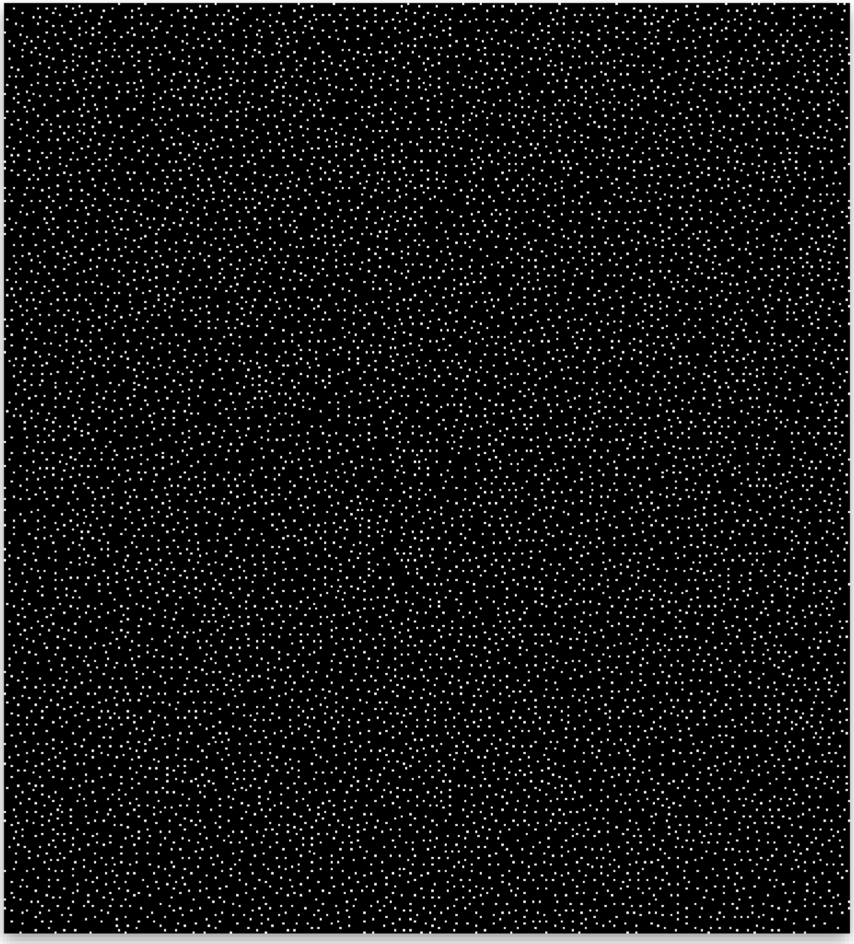
Random Noise



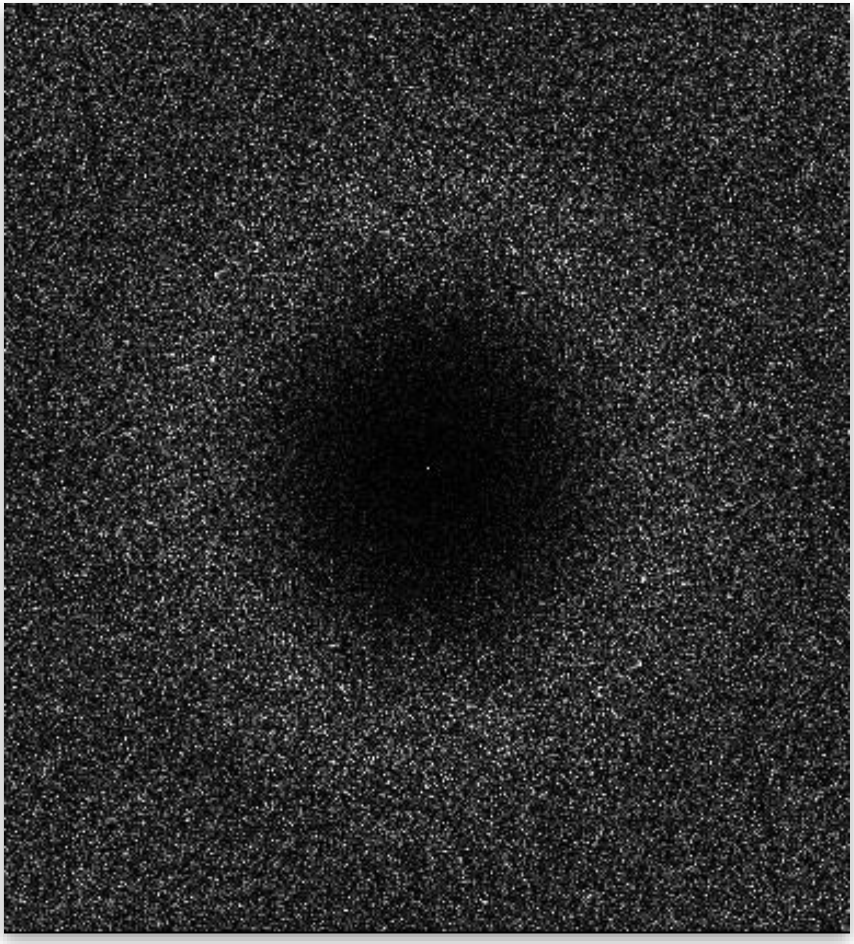
pixel image (b/w)

discrete Fourier transform
(power-spectrum)

Poisson Disc Sampling

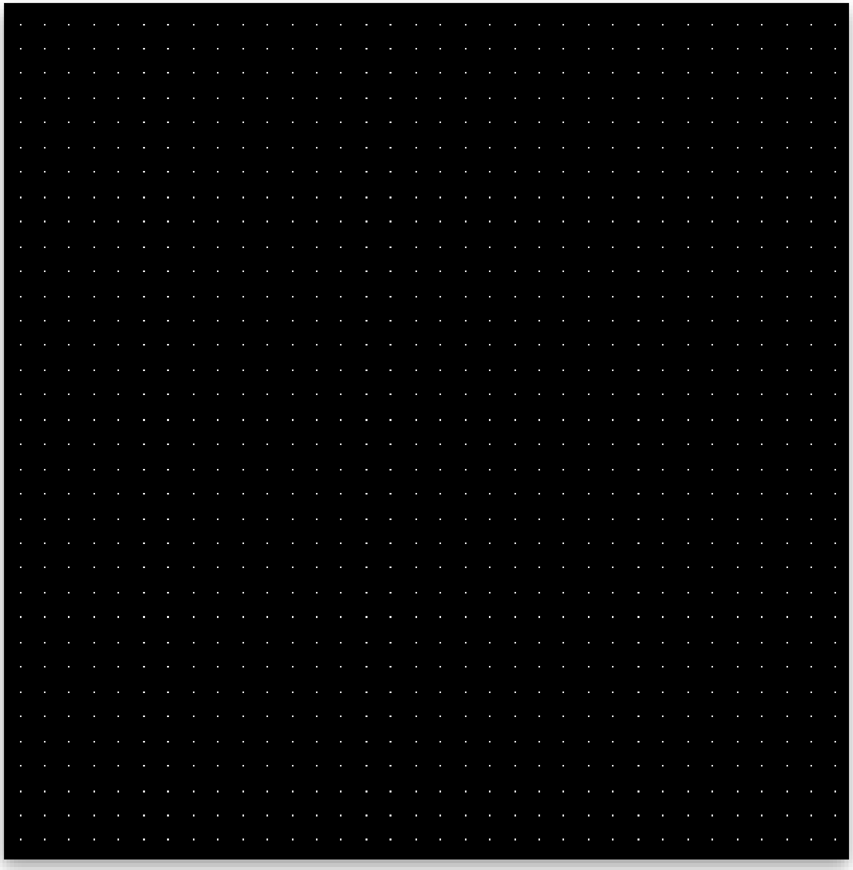


pixel image (b/w)

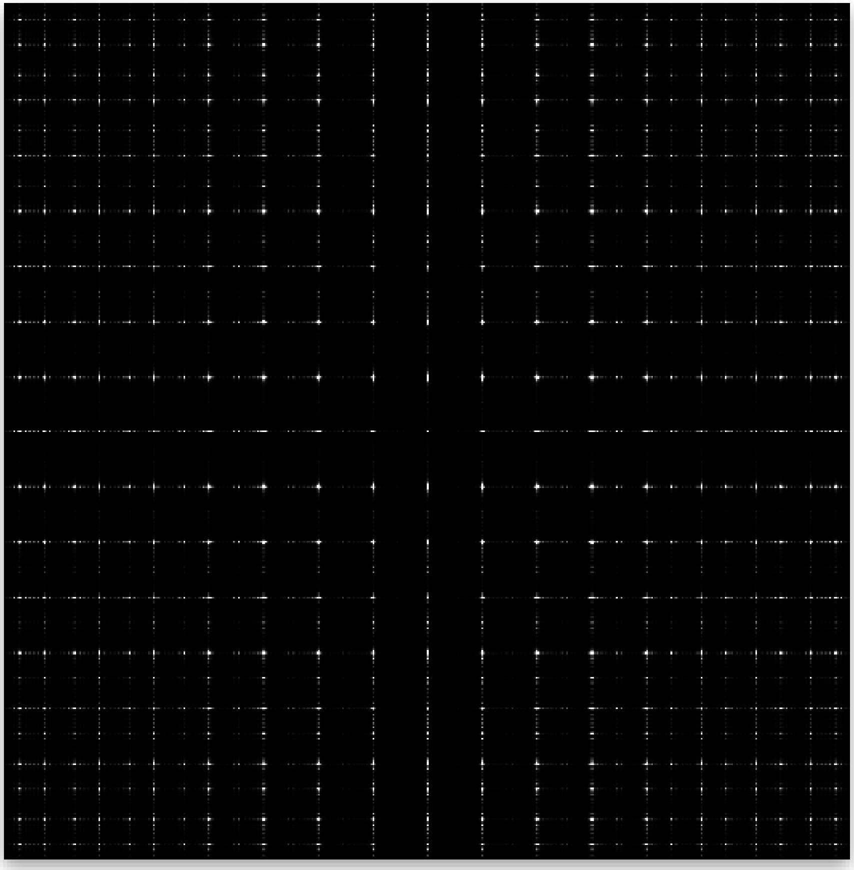


discrete Fourier transform (power-spectrum)

Regular Sampling



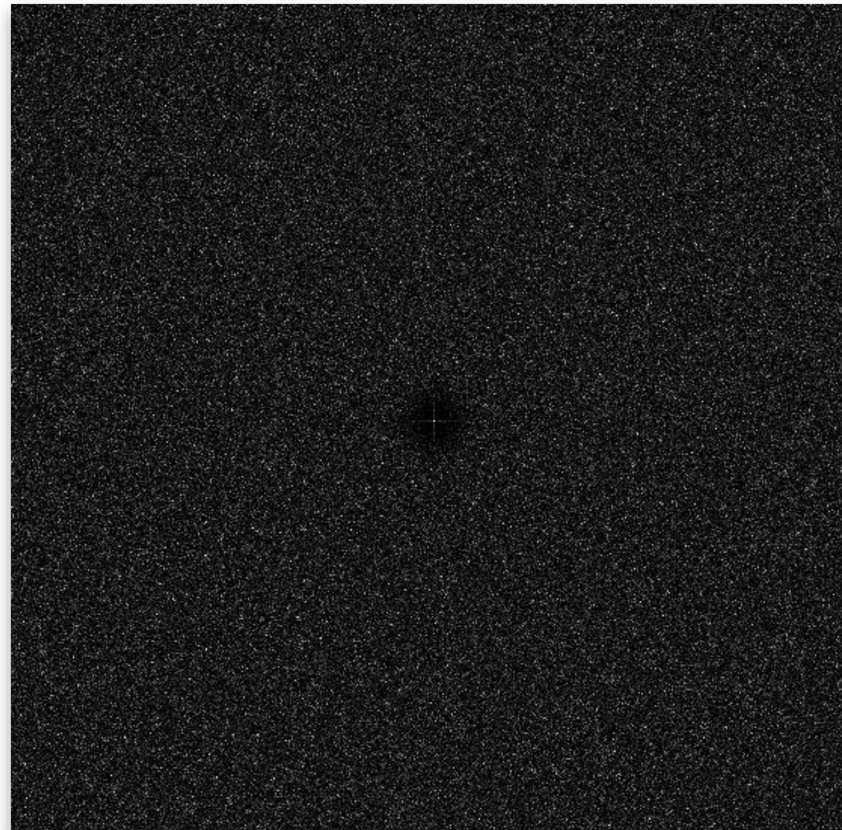
pixel image (b/w)



discrete Fourier transform
(power-spectrum)

Jittered Grid (Uniform Displacem.)

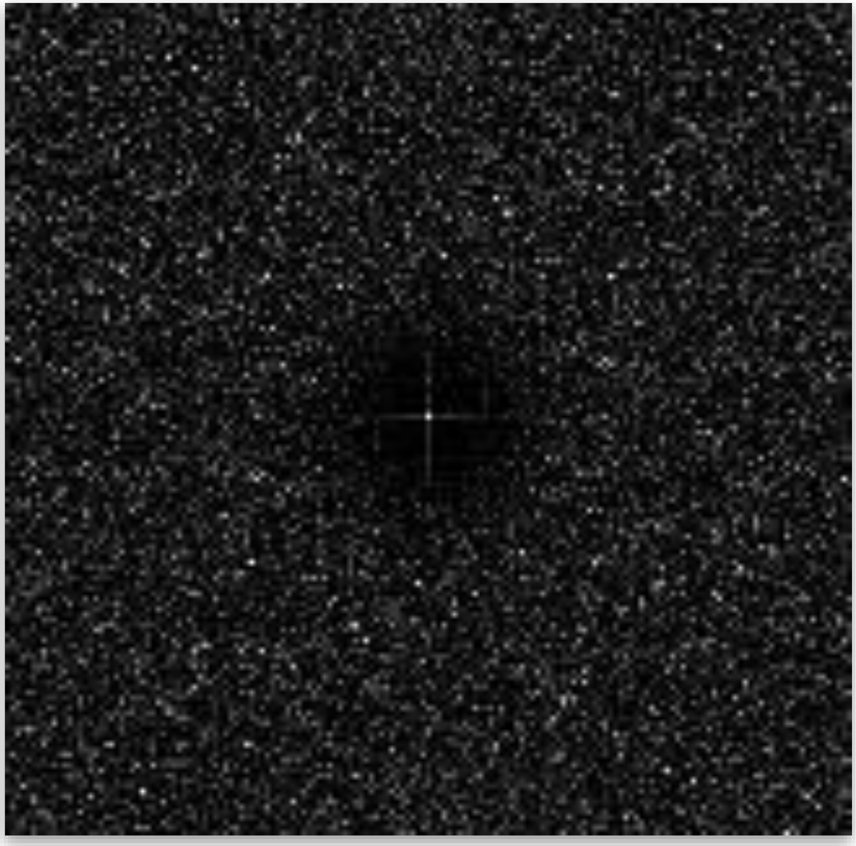
pixel image (b/w)



discrete Fourier transform (power-spectrum)

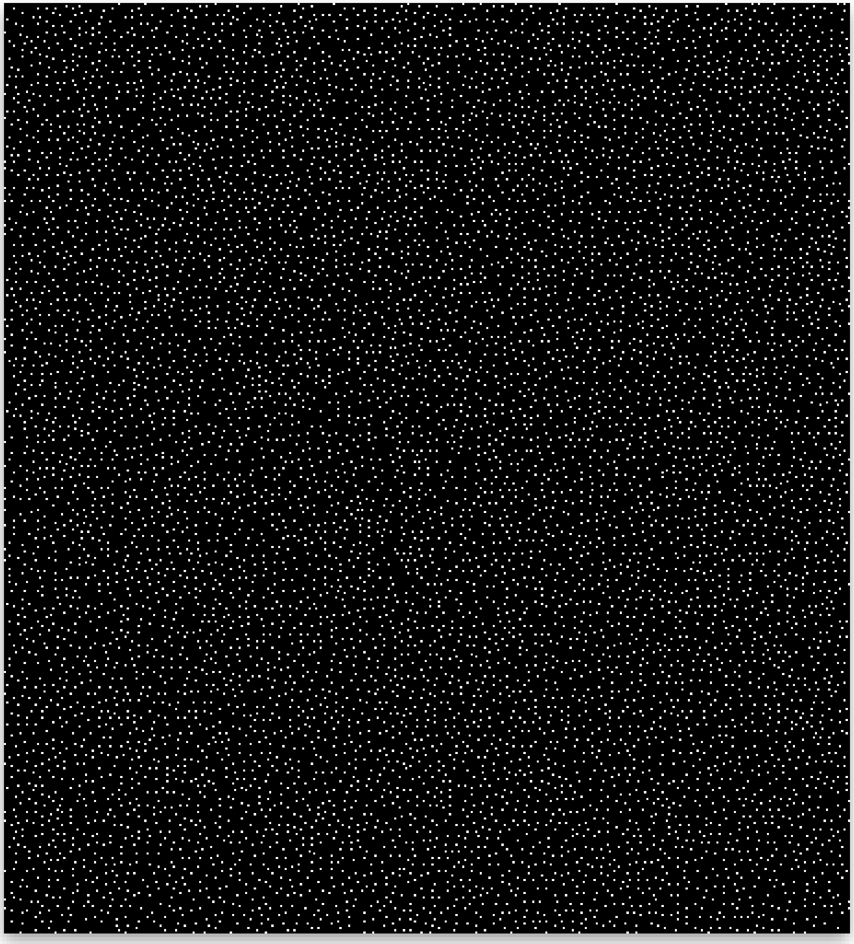
Jittered Grid (same density)

pixel image (b/w)

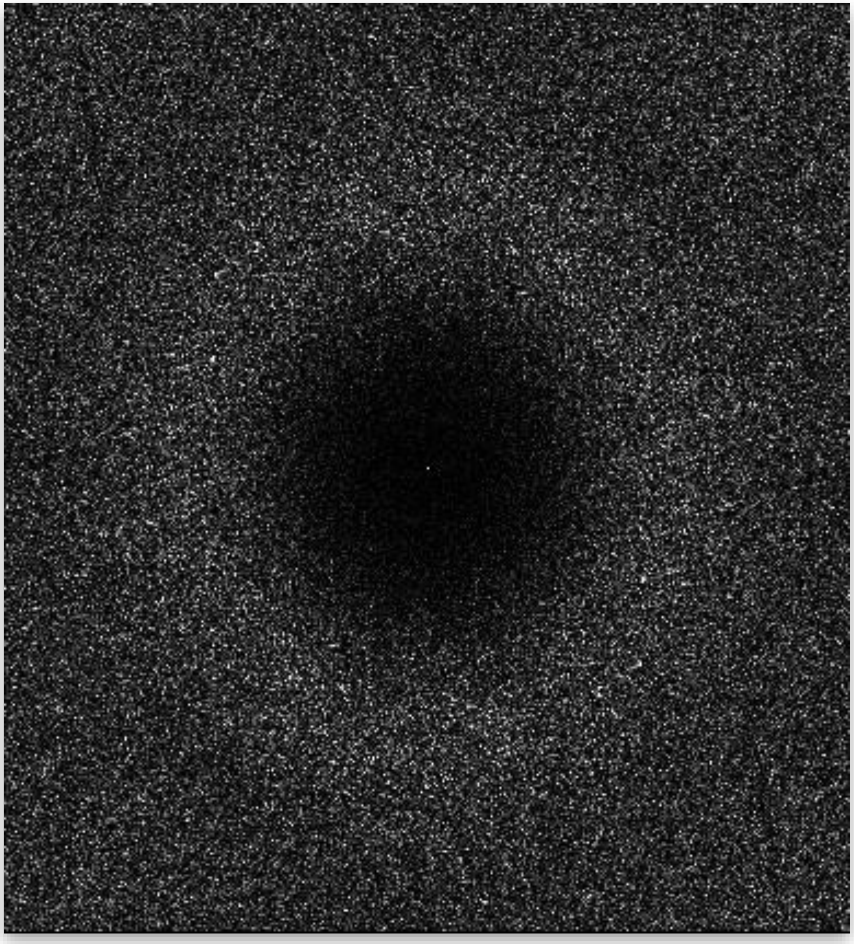


discrete Fourier transform (power-spectrum)

Examples

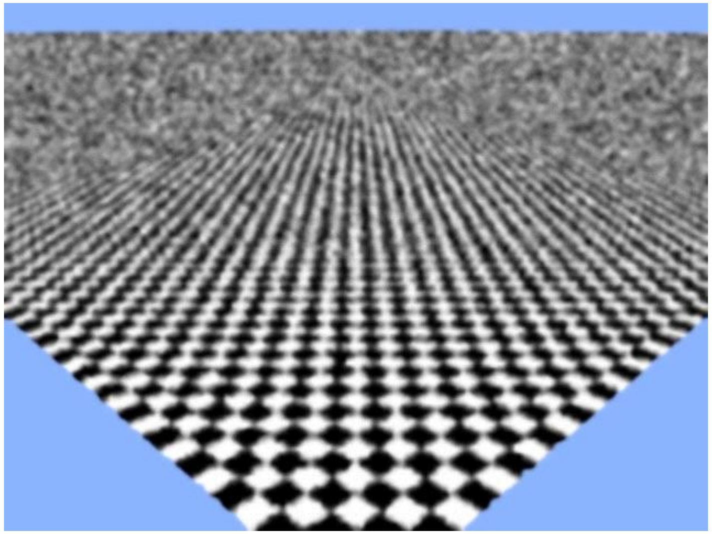


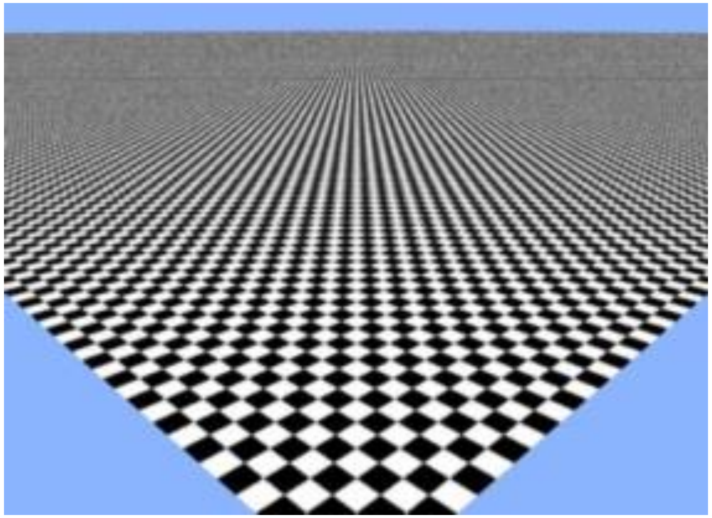
pixel image (b/w)



discrete Fourier transform
(power-spectrum)

Why should we care?





Exampe: Stochastic Raytracing

- Shoot random rays → random noise
- Low-pass filter → less noise
  - Low-frequency noise persists
  - LF-noise is particularly ugly!
  - Need many samples

Recipe:
Sampling Signals

How to Sample

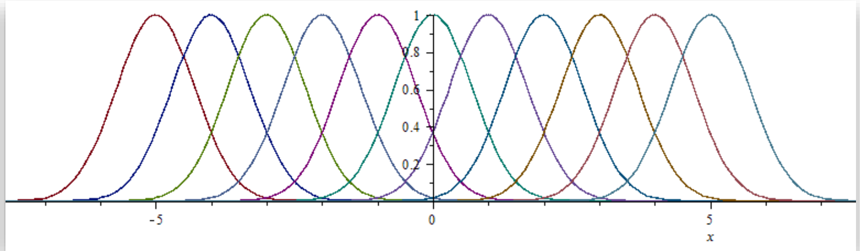
Given

- Function $f: \mathbb{R} \to \mathbb{R}$

Uniform sampling

- Sample spacing δ (given)

Choose filter kernel



- In case of doubt, try:

$$\omega(\mathrm{x}) = \exp(-\delta^{-1} x^2)$$

- Sample $\left(f \otimes \omega(\mathrm{x})\right)$ regularly
  - For example: Monte-Carlo integration

How to Sample

Given

- Function $f: \mathbb{R}^n \to \mathbb{R}^m$

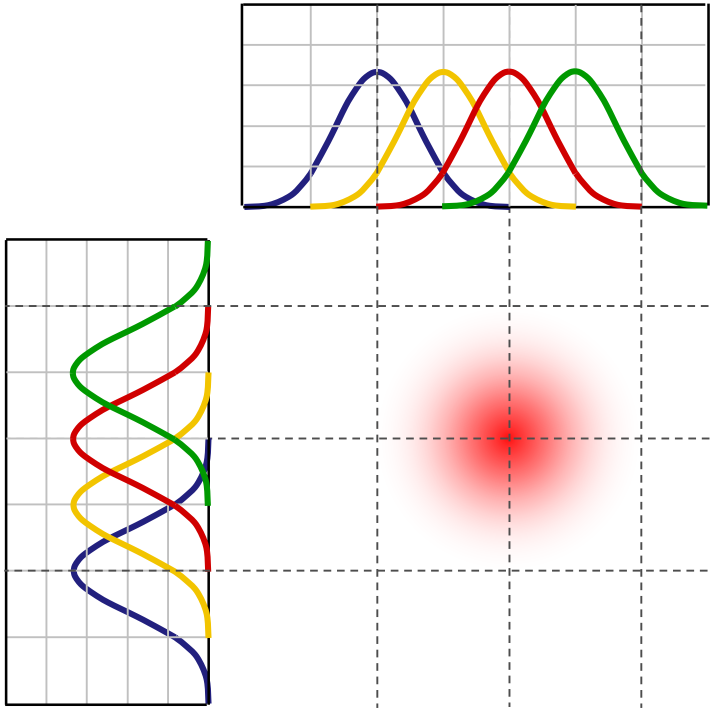
Multi-dimensional Gaussian

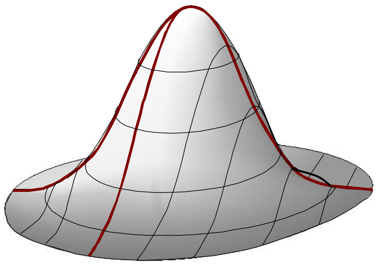
- In case of doubt, try:

$$\omega(\mathrm{x}) = \prod_{d=1}^{\mathrm{n}} \exp\left(-\frac{1}{\delta} x_d^2\right)$$

- Same procedure otherwise…

How to Sample





Multi-dimensional Gaussian

$$\omega(\mathrm{x}) = \prod_{d=1}^{n} \exp\left(-\frac{1}{\delta} x_d^2\right)$$

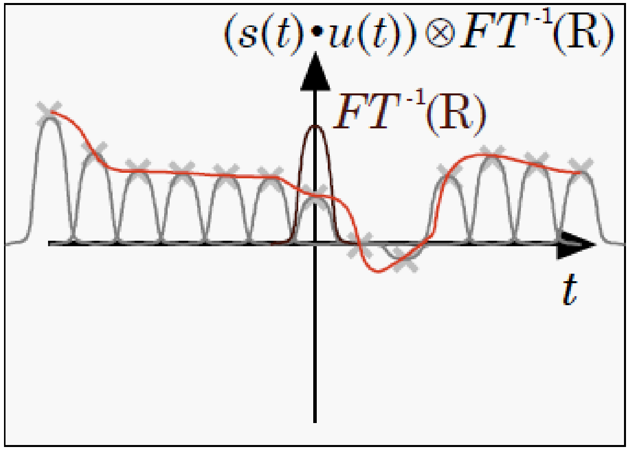
How to Sample

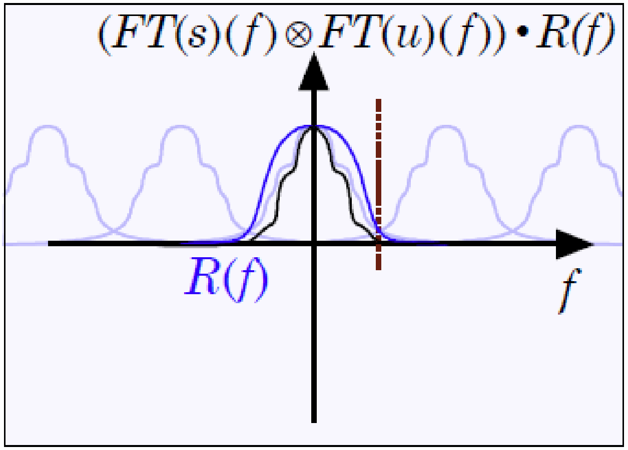
Non-Uniform Sampling

- Choose sample spacing $\delta(x)$
- Match level of detail
  - Nyquest limit
  - Spacing between two "ups" = frequency
- Filter adaptively
  - Varying filter width
- Sample adaptively
  - Sampling width varies accordingly

Recipe:
Reconstructing Signals

Signal Rec





Uniform

- Given samples $y_i = f(x_i), i = 1, \ldots, n$, spacing δ
- Chose reconstruction filter
- Try: $\omega(\mathrm{x}) = \exp(-\delta^{-2}x^2)$

Reconstruction: $\tilde{f} = \sum_{i=1}^{n} y_i \cdot \omega(\mathrm{x} - x_i)$

Non-Uniform

Non-Uniform

- Samples $y_i = f(x_i), i = 1, \dots, n,$
- Varying spacing δ_i
  - If unknown: average spacing of k-nearest neighbors
- Chose reconstruction filter
- Try: $\omega_i(\mathrm{x}) = \exp\left(-\delta_i^{-2}(x - x_i)^2\right)$

Reconstruction:

$$\tilde{f} = \frac{\sum_{i=1}^{n} y_i \cdot \omega_i(\mathrm{x} - x_i)}{\sum_{i=1}^{n} \omega_i(\mathrm{x} - x_i)}$$

"Partition of Unity" just to be save…

Reconstruction: Implementation

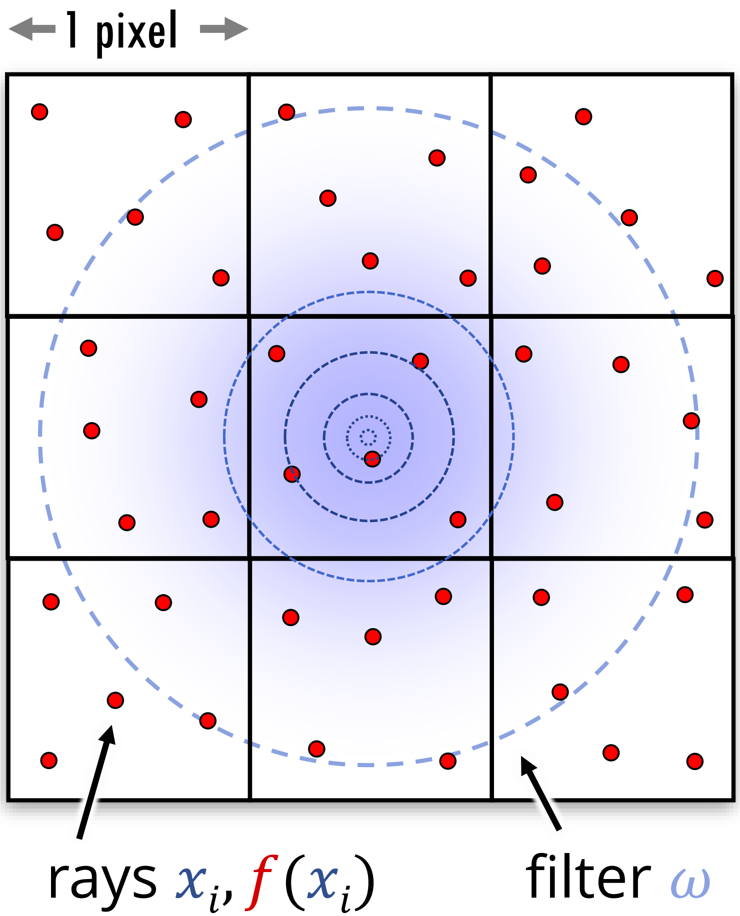
Variant 1: Gathering

- Record *samples in list* (plus kD Tree, Octree, grid)
- For each *pixel*:
  - Range query: kernel support radius
  - Compute weighted sum (last slide)

Variant 2: Splatting

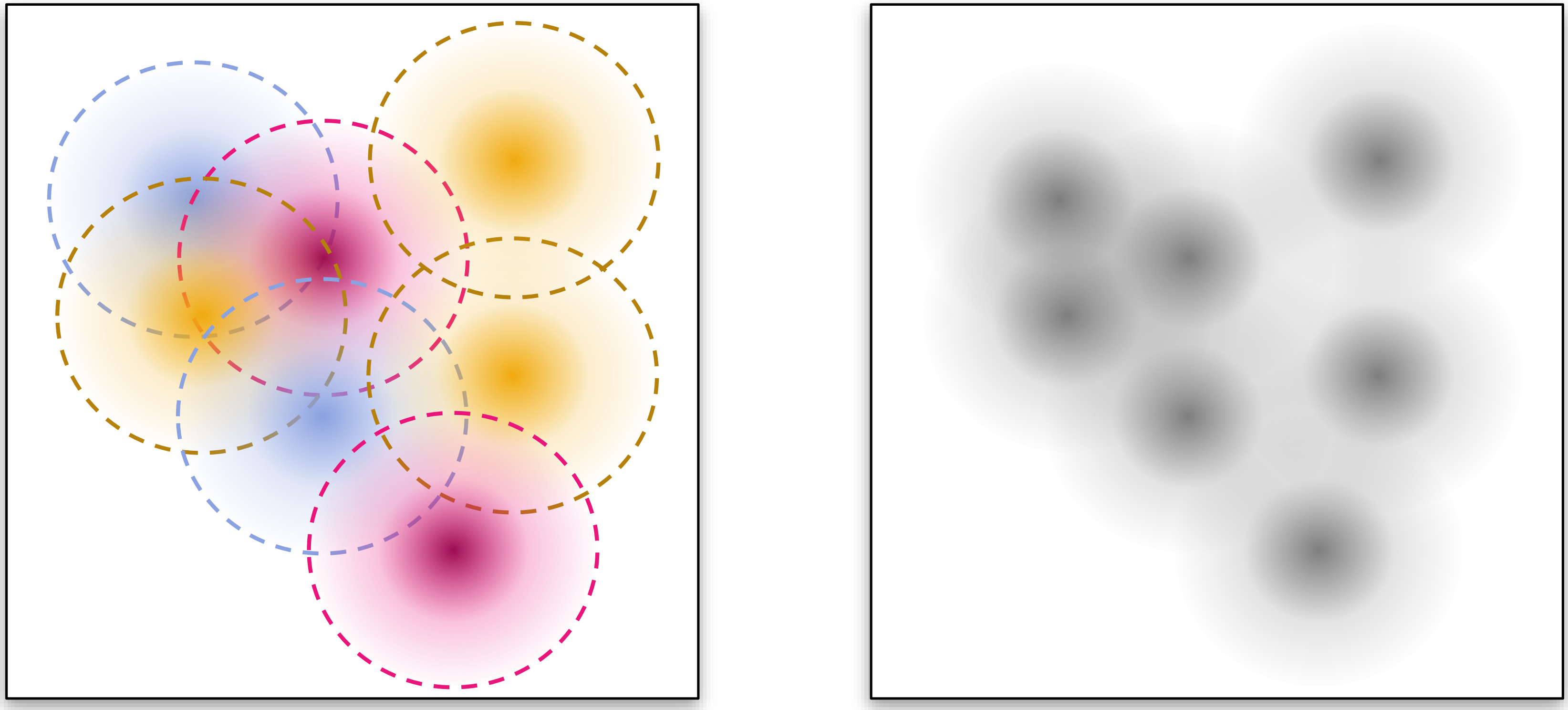
- Two *pixel* buffers: Color (3D), weight (1D)
- Iterate over *samples*:
  - Add Gaussian splat to weight buffer
  - Add 3× Gaussian splat scaled by RGB to color buffer
- In the end: Divide *color buffer* by *weight buffer*.

Gathering



$$\tilde{f} = \frac{\sum_{i=1}^{n} y_i \cdot \omega(\mathrm{x} - x_i)}{\sum_{i=1}^{n} \omega(\mathrm{x} - x_i)}$$

Splatting

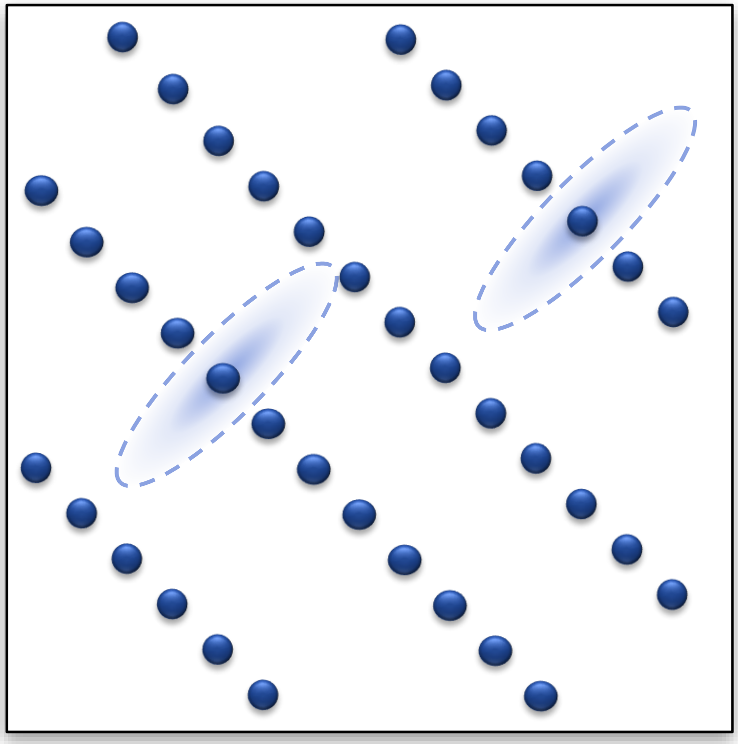


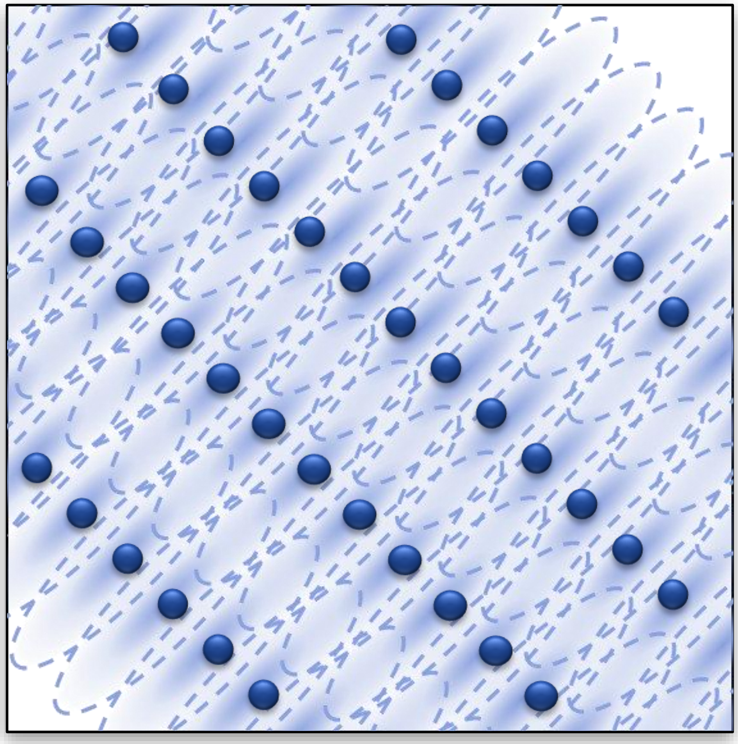
color buffer

weight buffer

$$\tilde{f} = \frac{\sum_{i=1}^{n} y_i \cdot \omega(\mathrm{x} - x_i)}{\sum_{i=1}^{n} \omega(\mathrm{x} - x_i)}$$

Remark: Anisotropic Filtering

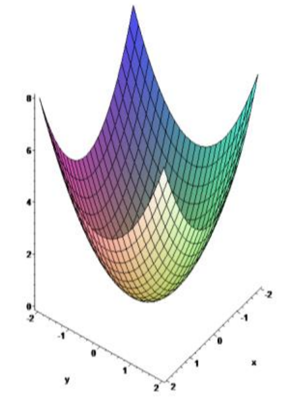


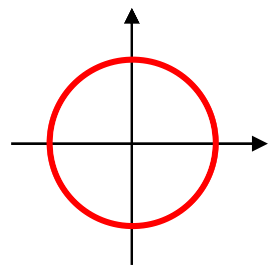


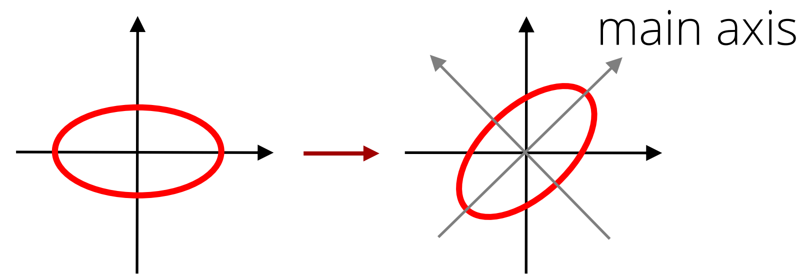
$$\tilde{f} = \frac{\sum_{i=1}^{n} y_i \cdot \omega(\mathrm{x} - x_i)}{\sum_{i=1}^{n} \omega(\mathrm{x} - x_i)}$$

Building Anisotropic Filters

$\mathbf{x}^{\mathrm{T}}\mathbf{x}$







$$\mathbf{x}^{\mathrm{T}}[\mathbf{T}^{\mathrm{T}} \cdot \mathbf{T}]\mathbf{x}$$

How to construct?

- Given: Kernel $w(\mathbf{x})$
  - For example: $w(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma}\mathbf{x}^{\mathrm{T}}\mathbf{x}\right)$
- Coordinate transformation:
  - $w(\mathbf{x}) \rightarrow w(\mathbf{T}\mathbf{x})$
  - Gaussian: $w(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma}\mathbf{x}^{\mathrm{T}}[\mathbf{T}^{\mathrm{T}} \cdot \mathbf{T}]\mathbf{x}\right)$

Advanced Reconstruction

Push-Pull Algorithm

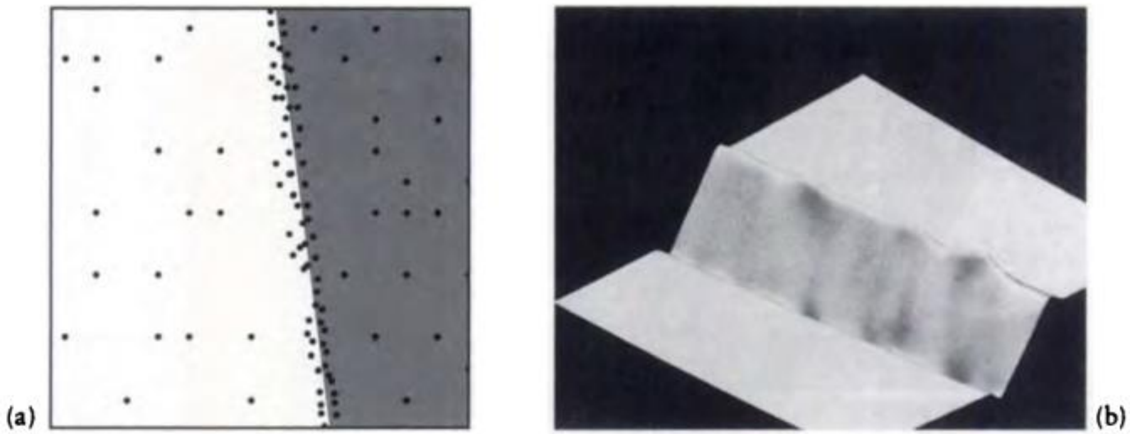


FIGURE 10.101
(a) The test situation: a straight edge between black and white regions. (b) A failure of weighted-average reconstruction. Reprinted, by permission, from Mitchell in *Computer Graphics (Proc. Siggraph '87)*, fig. 11, p. 72.

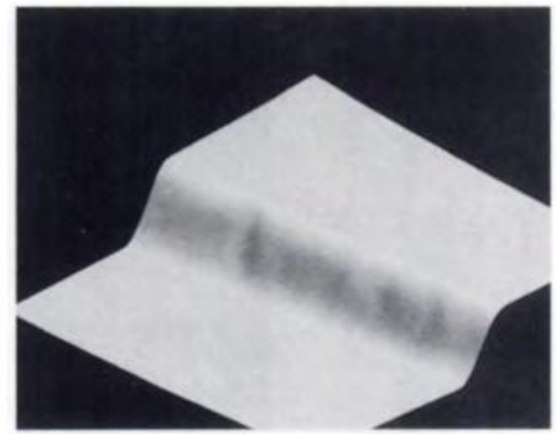


FIGURE 10.103
Reconstruction with the Mitchell multistage filter. Reprinted, by permission, from Mitchell in *Computer Graphics (Proc. Siggraph '87)*, fig. 14, p. 72.

Source: [Glassner 1995, Principles of digital image synthesis, CC license]

Problem with partition-of unity:

Artifacts at boundaries of sampling

Remedy

Push-Pull-Algorithm

- Reconstruct at multiple levels (stratification)
  - Build quadtree
  - Keep one sample per cell
  - Creates different levels
- Add results together
  - Do not reconstruct in empty cells

Reduced bias

Advanced Reconstruction

Moving Least-Squares

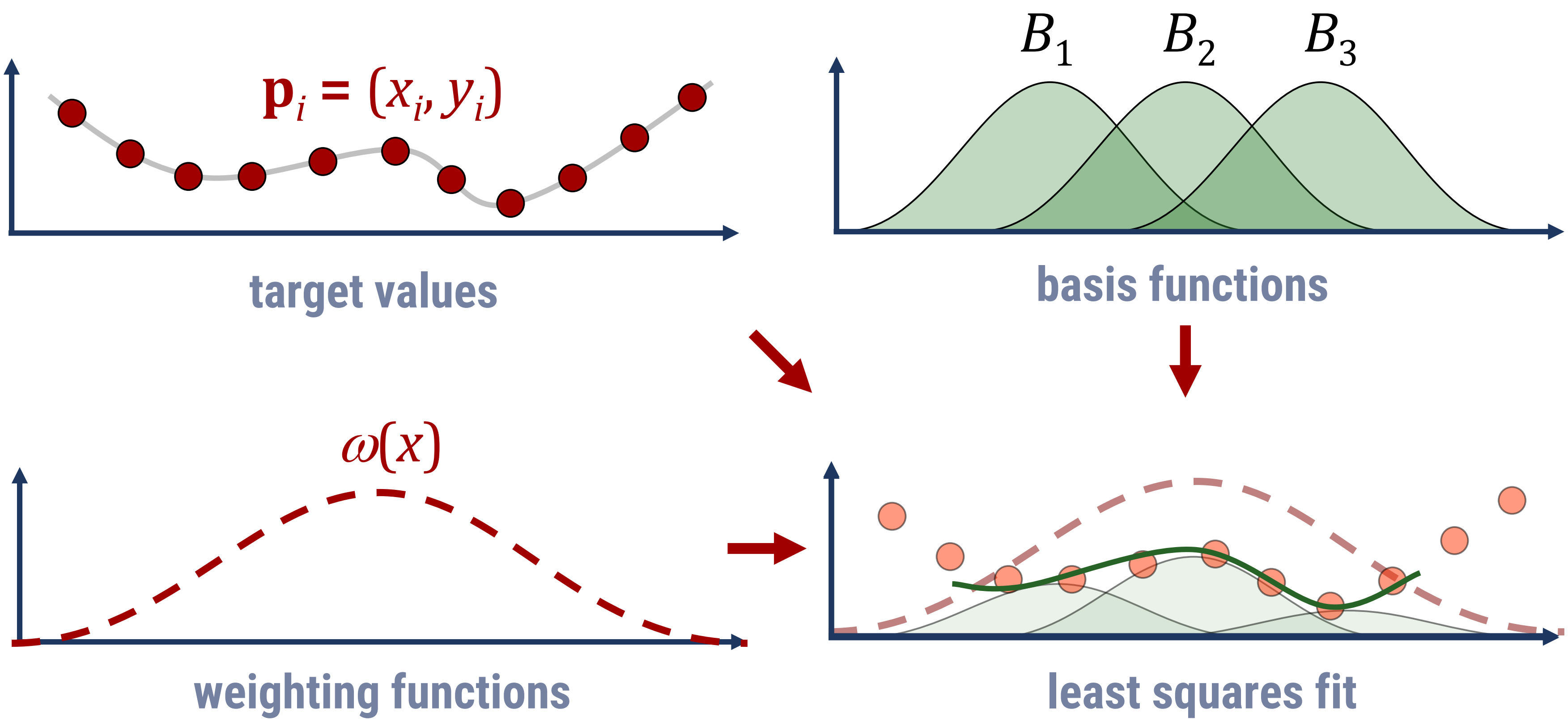
Moving Least Squares

Moving least squares (MLS):

- MLS is a standard technique for scattered data interpolation.
- Generalization of partition-of-unity method

Weighted Least-Squares

Least Squares Approximation:



Least-Squares

Least Squares Approximation:

$$\tilde{y}(x) = \sum_{i=1}^{n} \lambda_i B_i(x)$$

Best Fit (weighted):

$$\underset{c_i}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \left(\tilde{y}(x_i) - y_i \right) \omega(x_i) \right\|^2$$

Least-Squares

Normal Equations: $\left(\mathbf{B}^T\mathbf{W}^2\mathbf{B}\right)\boldsymbol{\lambda} = \left(\mathbf{B}^T\mathbf{W}^2\right)\mathbf{y}$

Solution: $\boldsymbol{\lambda} = \left(\mathbf{B}^T\mathbf{W}^2\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{W}^2\mathbf{y}$

Evaluation: $\tilde{y}(x) = \langle \mathbf{b}(x), \boldsymbol{\lambda} \rangle = \mathbf{b}(x)^{\mathrm{T}}\left(\mathbf{B}^{\mathrm{T}}\mathbf{W}^2\mathbf{B}\right)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{W}^2\mathbf{y}$

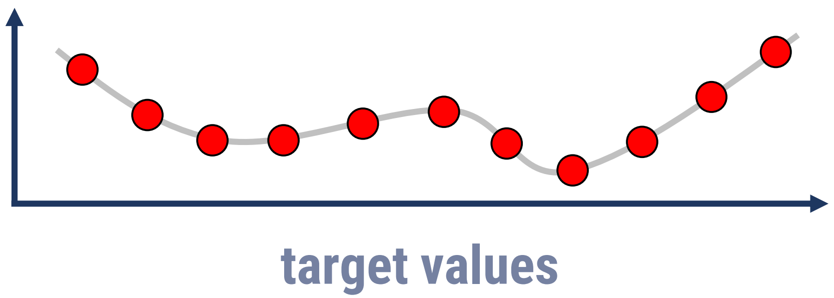
MLS approximation

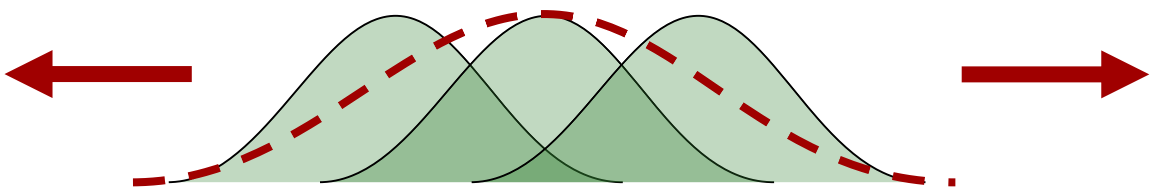
$$\mathbf{b} := \left[B_1, \ldots, B_n\right]$$

$$\mathbf{B} := \begin{bmatrix} -\mathbf{b}(x_1)- \\ \vdots \\ -\mathbf{b}(x_n)- \end{bmatrix} \qquad \mathbf{y} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{W} := \begin{bmatrix} \omega(x_1) & & \\ & \ddots & \\ & & \omega(x_n) \end{bmatrix}$$

Moving Least-Squares

Moving Least Squares Approximation:

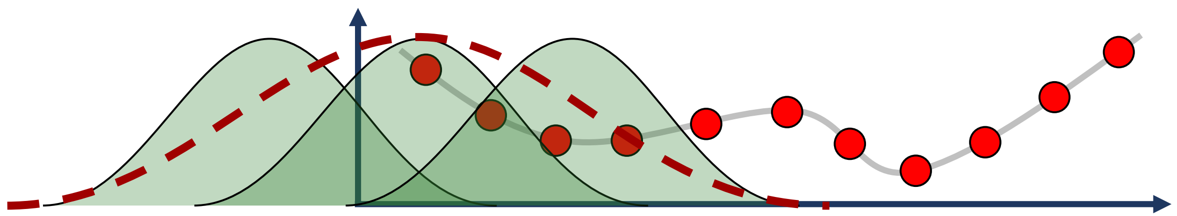




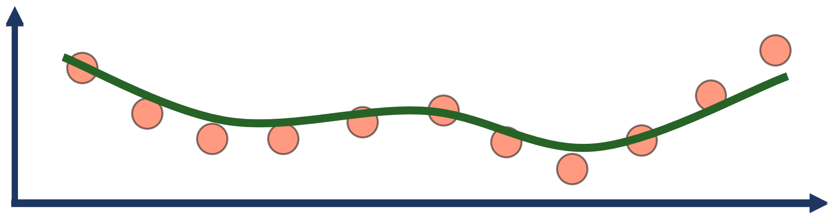
move basis and weighting function, recompute approximation $\tilde{y}(x)$

Moving Least-Squares

Moving Least Squares Approximation:



target values



approximation

Summary: MLS

Standard MLS approximation:

- Choose set of basis functions
  - Typically monomials of degree 0,1,2
- Choose weighting function
  - Typical choices: Gaussian, Wendland function, B-Splines
  - Solution will have the same continuity as the weighting function.
- Solve a weighted least squares problem at each point:

$$\tilde{y}(x) = \mathbf{b}(x)^{\mathrm{T}} \left(\mathbf{B}(x)^{\mathrm{T}} \mathbf{W}(x)^2 \mathbf{B}(x) \right)^{-1} \mathbf{B}(x)^{\mathrm{T}} \mathbf{W}(x)^2 \mathbf{y}$$

moment matrix

- Need to invert the "moment matrix" at each evaluation.
- Use SVD if sampling requirements are not guaranteed.

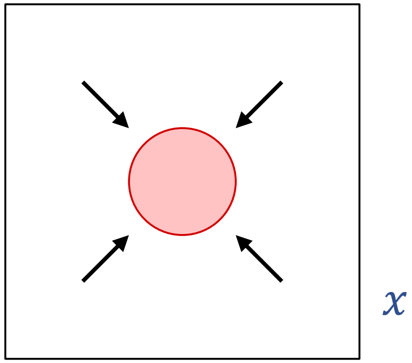
Remark

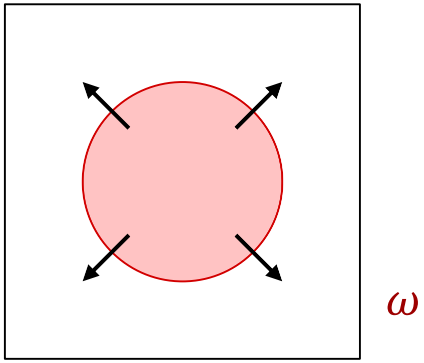
Uncertainty Relation(s)

Fourier Transform Pairs

Gaussians

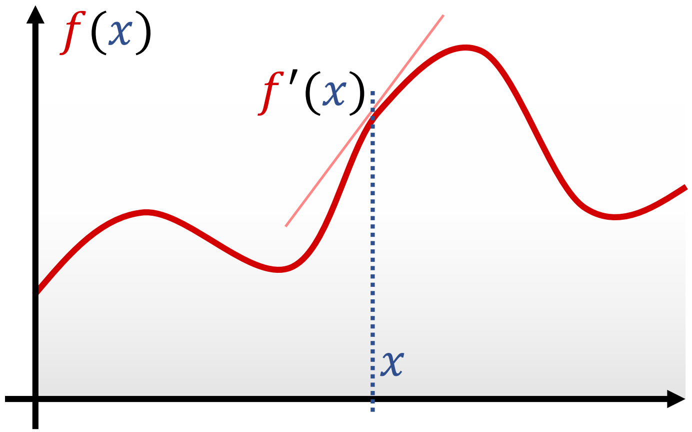
$$f(x) = e^{-ax^2} \quad \rightarrow \quad F(\omega) = \sqrt{\frac{\pi}{a}} \cdot e^{-\frac{(\pi\omega)^2}{a}}$$





Taylor-Approximation

Function f

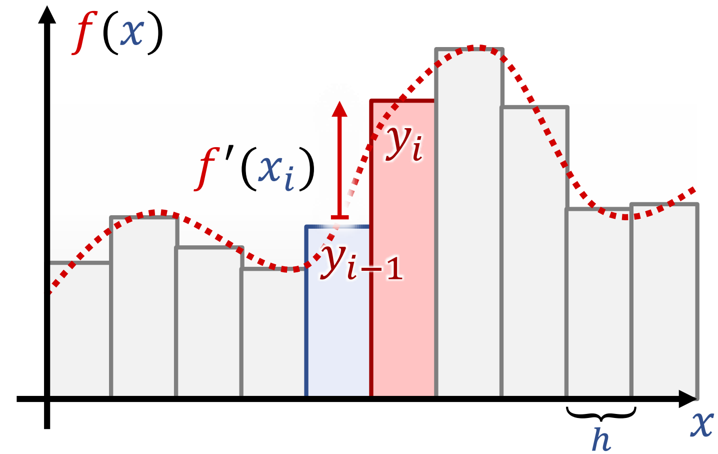


tangent slope

$$f : \mathbb{R} \to \mathbb{R}$$

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Think of this:



neighborhood differences

$$f = (y_1, \ldots, y_n)$$

$$f'(x_i) \approx \frac{y_i - y_{i-1}}{h}$$