Visual Computing:

Fourier Transform

Prof. Marc Pollefeys

Prof. Markus Gross

ETH

Informatik
Computer Science

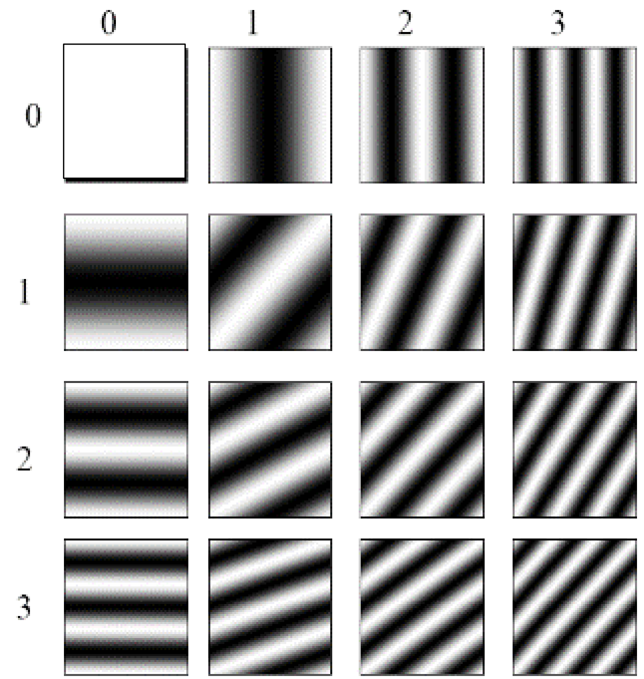
Last lecture

Fourier Transform

$$\boxed{F(g(x,y))(u,v)=\iint_{\mathbb{R}^2} g(x,y)e^{-i2\pi(ux+vy)}dxdy}$$

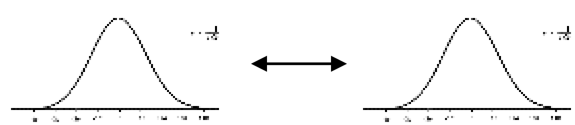
$$e^{-i2\pi(ux+vy)}$$

$= \cos 2\pi(ux+vy) - i \sin 2\pi\,(ux+vy)$

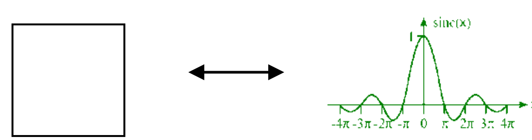


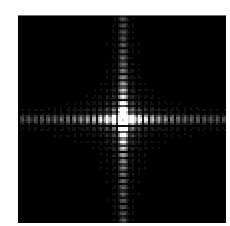
Various Fourier Transform Pairs

- Important facts
  - The Fourier transform is linear
  - There is an inverse FT $$f = \mathbf{U}^{-1}F$$
  - scale function down ⇔ scale transform up
    i.e. high frequency = small details
  - The FT of a Gaussian is a Gaussian.



compare to box function transform

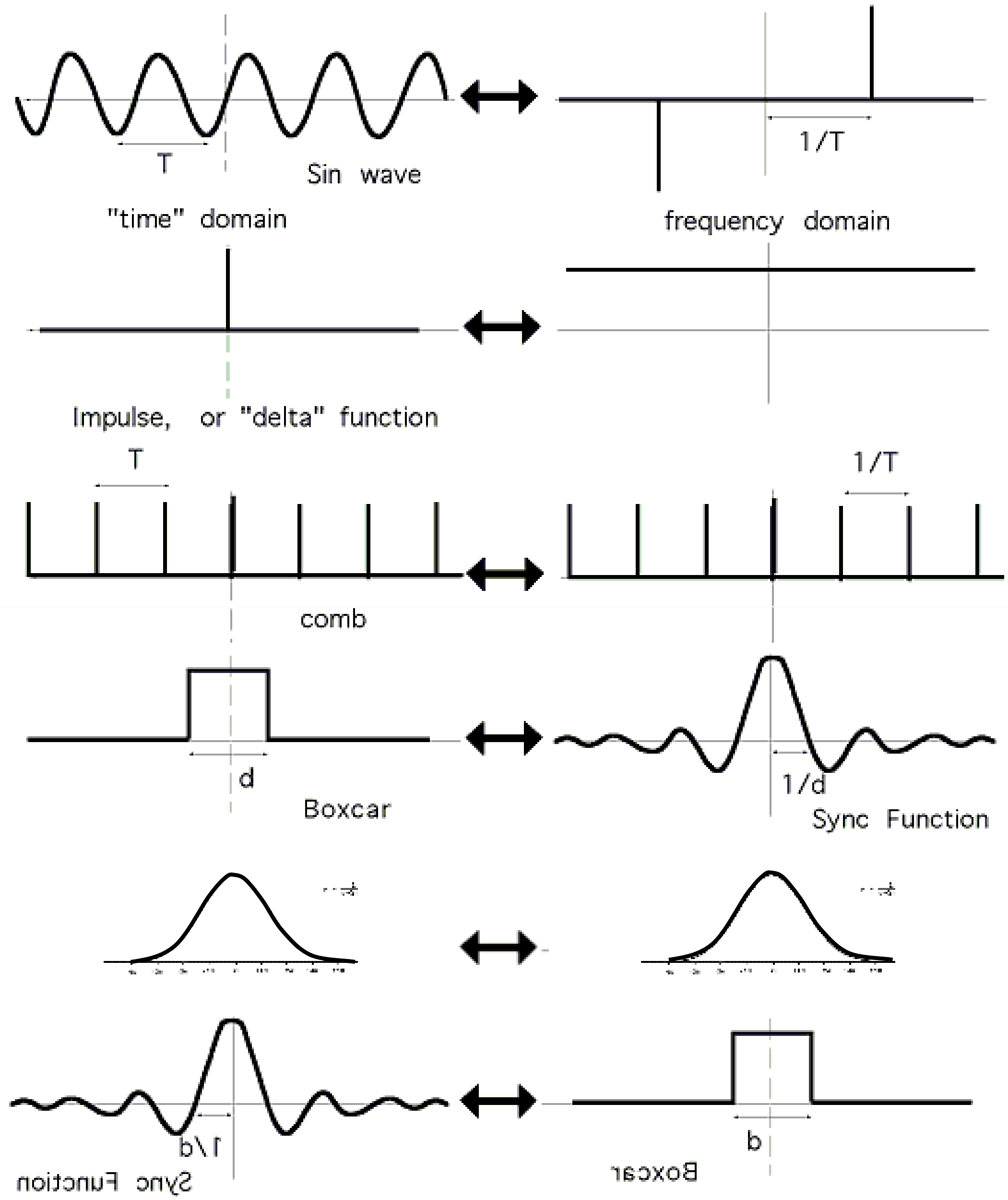




Fourier Transform of important functions

spatial domain

frequency domain



Convolution theorem

- The convolution theorem
  - The Fourier transform of the convolution of two functions is the product of their Fourier transforms

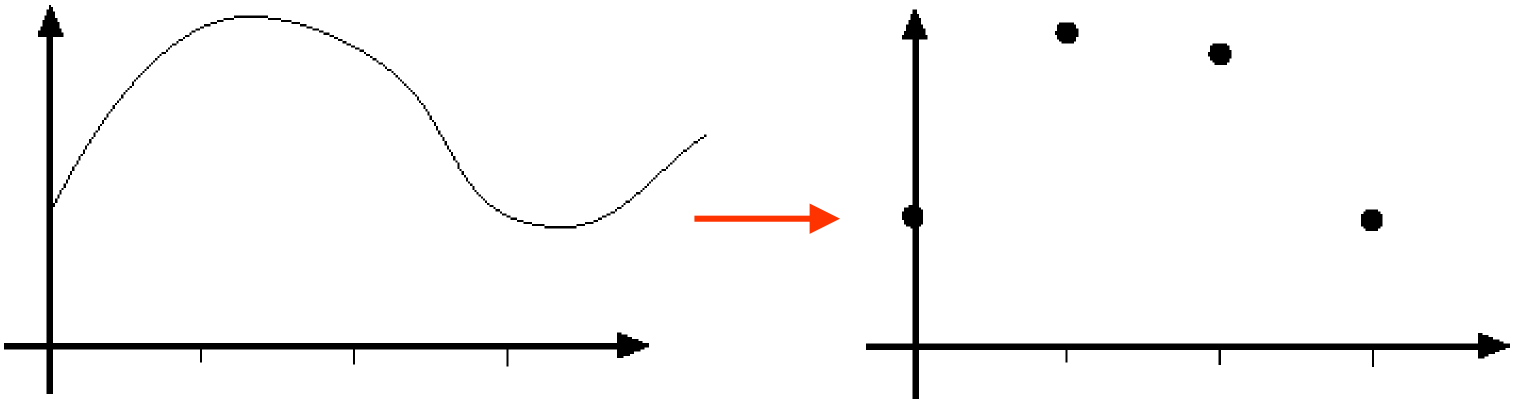
$$F.G = \mathbf{U}\left(f ** g\right)$$ (cfr. filtering)

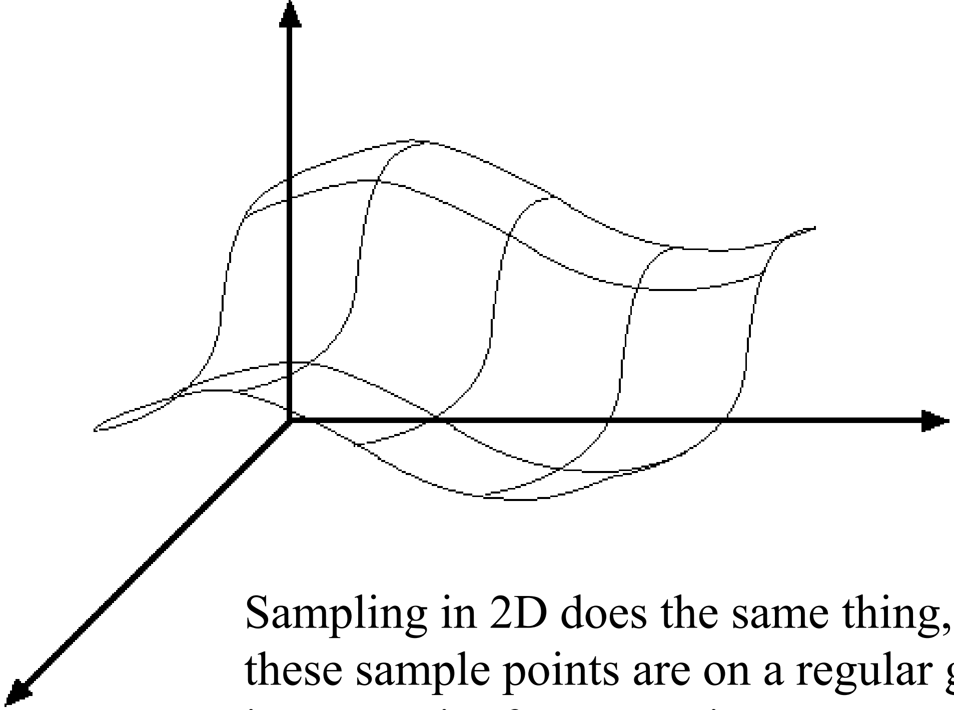
  - The Fourier transform of the product of two functions is the convolution of the Fourier transforms

$$F ** G = \mathbf{U}\left(f . g\right)$$ (cfr. sampling)

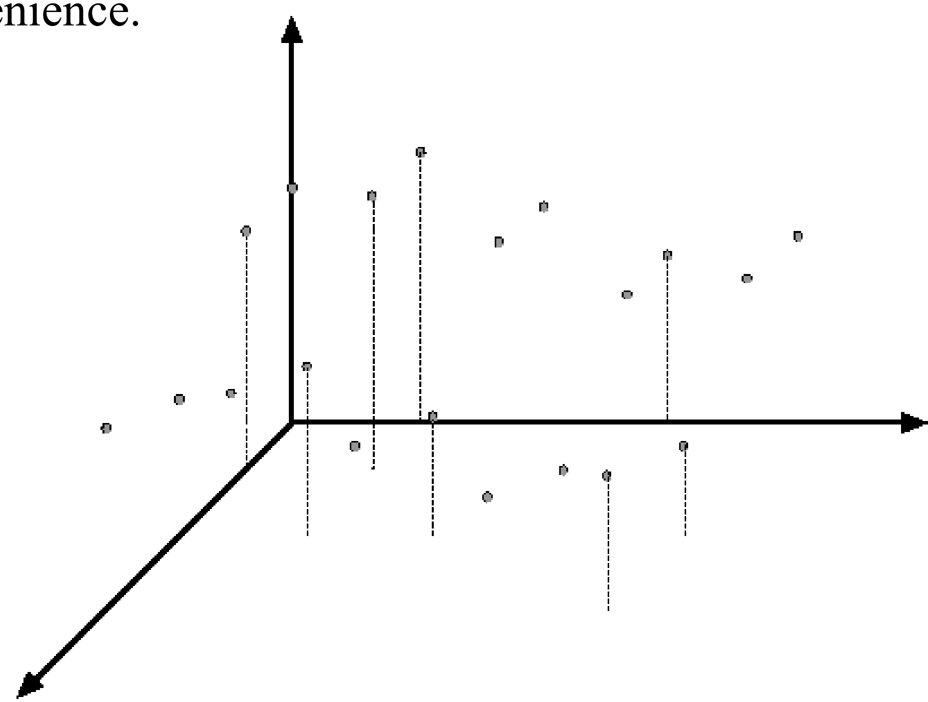
Sampling

- Go from continuous world to discrete world, from function to vector
- Samples are typically measured on regular grid





Sampling in 2D does the same thing, only in 2D. We' ll assume that these sample points are on a regular grid, and can place one at each integer point for convenience.



A continuous model for a sampled function

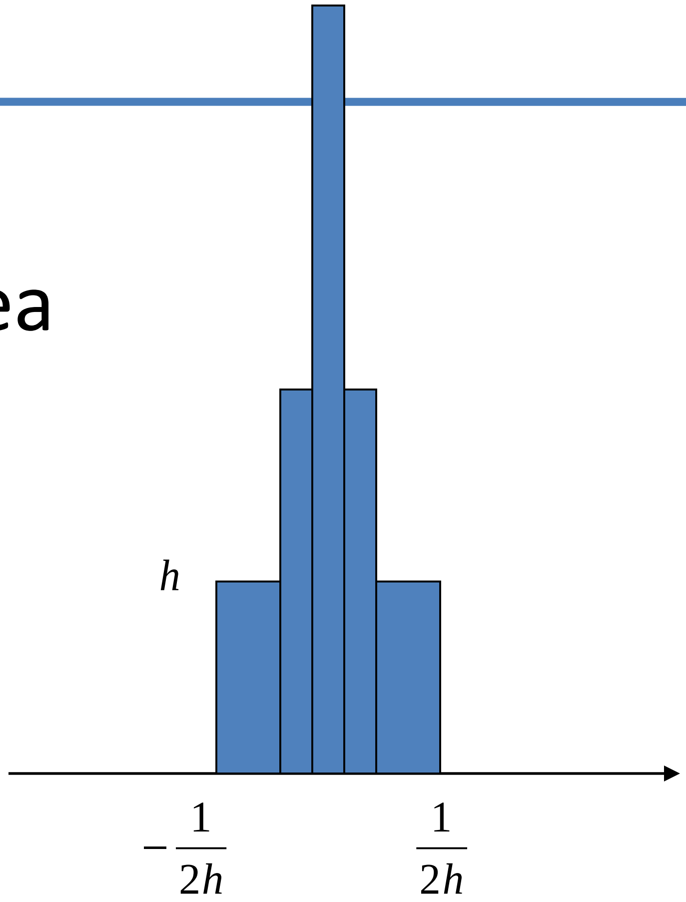
- We want to be able to approximate integrals sensibly
- Leads to
  - the delta function
  - model on right

$$\mathrm{Sample}_{2D}\left(f(x,y)\right) = \sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} f(x,y)\delta(x-i, y-j)$$

$$= f(x,y) \sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i, y-j)$$

Delta function

- limit to infinity of constant area function:



A continuous model for a sampled function

- We want to be able to approximate integrals sensibly
- Leads to
  - the delta function
  - model on right

$$\text{Sample}_{2\mathrm{D}}\big(f(x,y)\big) = \sum_{i=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} f(x,y)\,\delta(x-i, y-j)$$

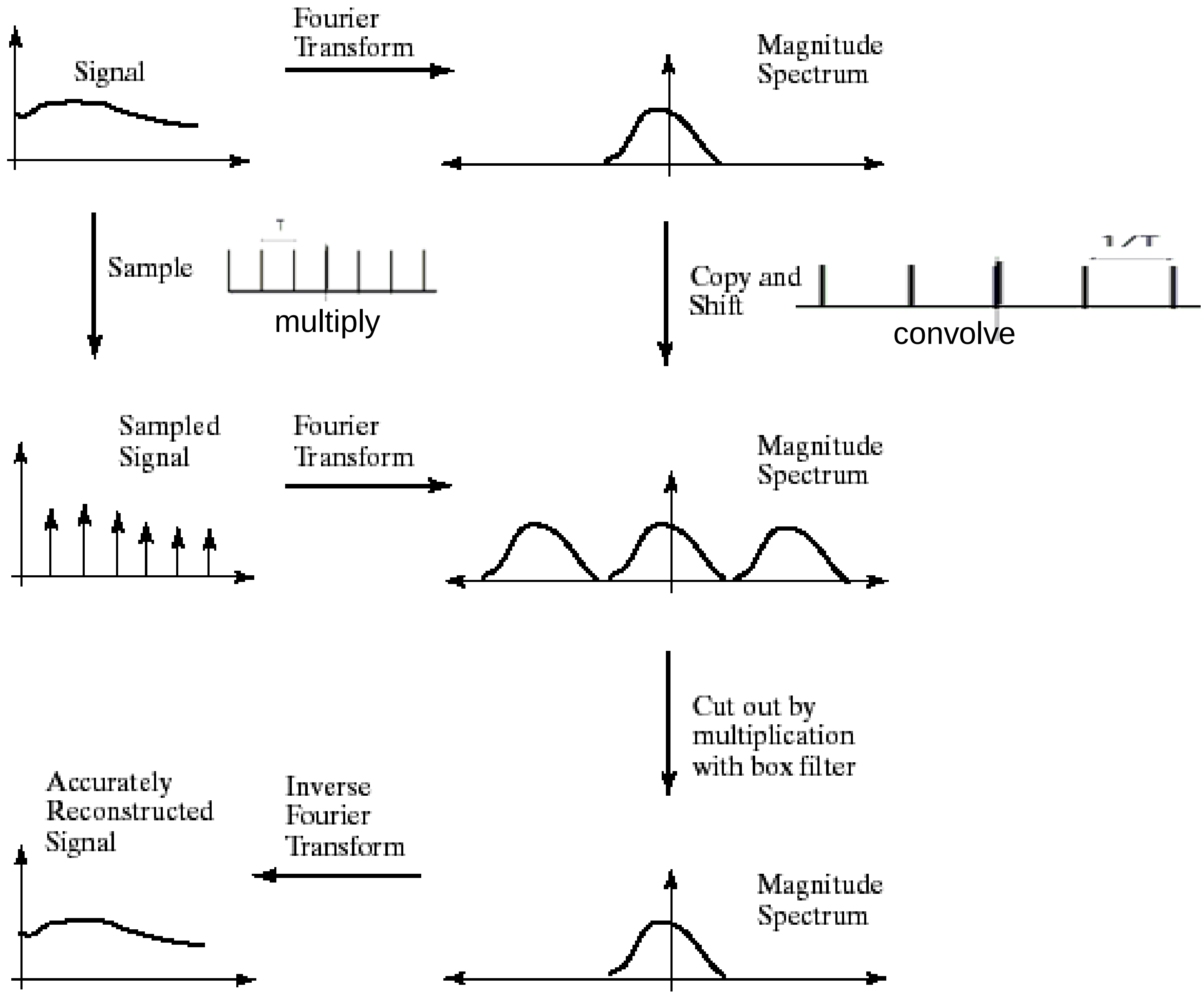
$$= f(x,y)\sum_{i=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} \delta(x-i, y-j)$$

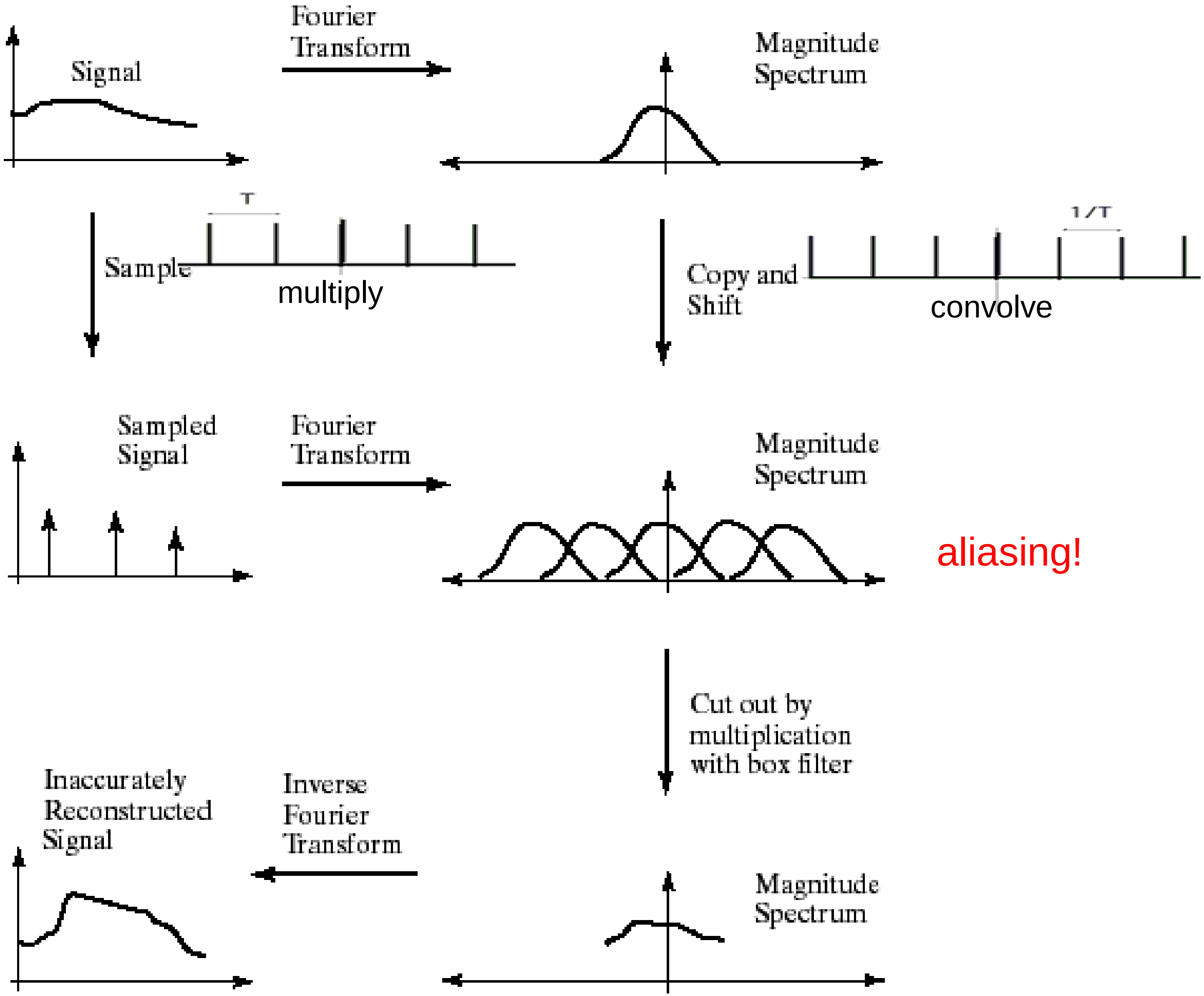
The Fourier transform of a sampled signal

$$F\left(\mathrm{Sample}_{2\mathrm{D}}(f(x,y))\right) = F\left(f(x,y) \sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i, y-j) \right)$$

$$= F\left(f(x,y)\right) ** F\left(\sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i, y-j) \right)$$

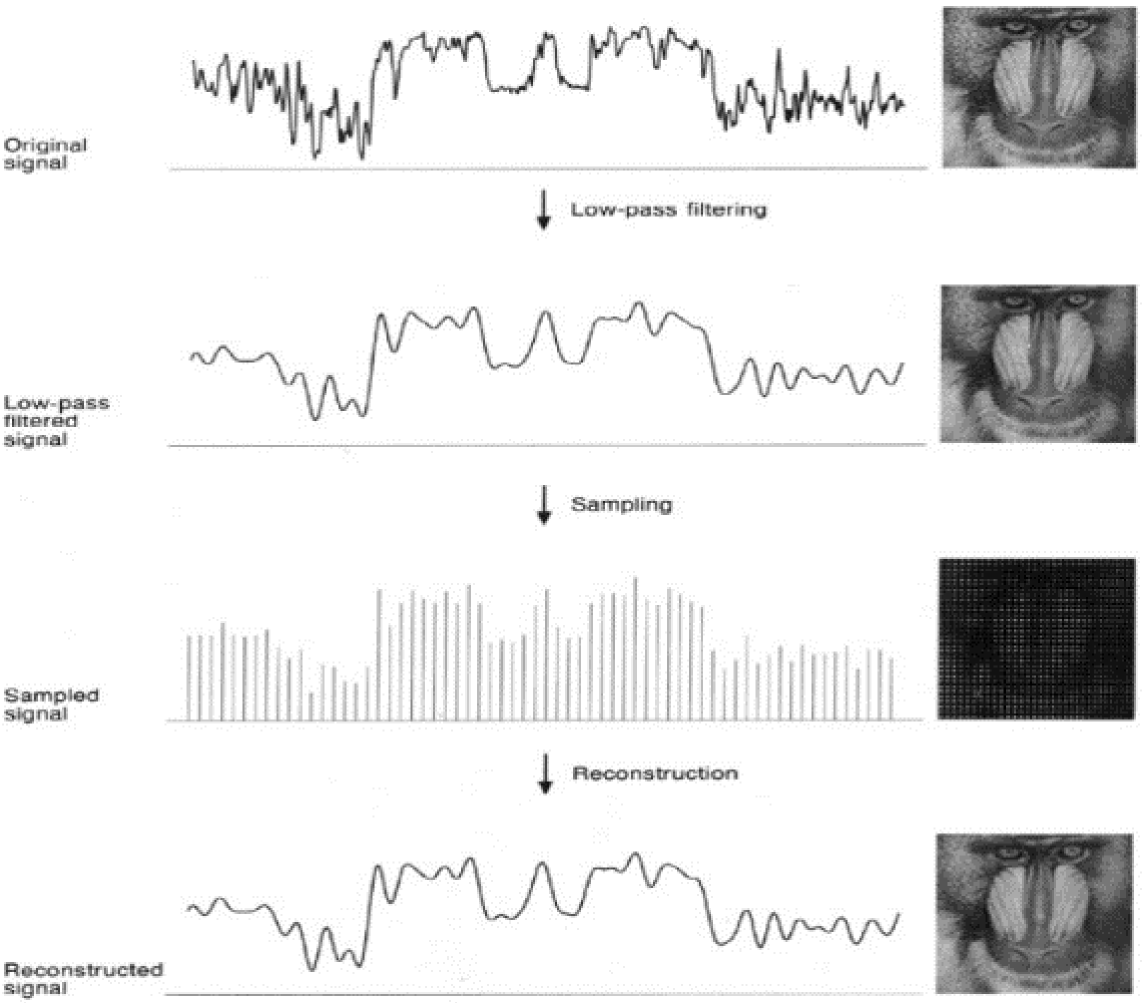
$$= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F\left(u-i, v-j\right)$$


Signal
Fourier Transform
Magnitude Spectrum
Sample
T
multiply
Copy and Shift
convolve
Sampled Signal
Fourier Transform
Magnitude Spectrum
Cut out by multiplication with box filter
Accurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum



Fourier Transform
Signal
Magnitude Spectrum
T
Sample
multiply
1/T
Copy and Shift
convolve
Sampled Signal
Fourier Transform
Magnitude Spectrum
aliasing!
Cut out by multiplication with box filter
Inaccurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum

Proper sampling

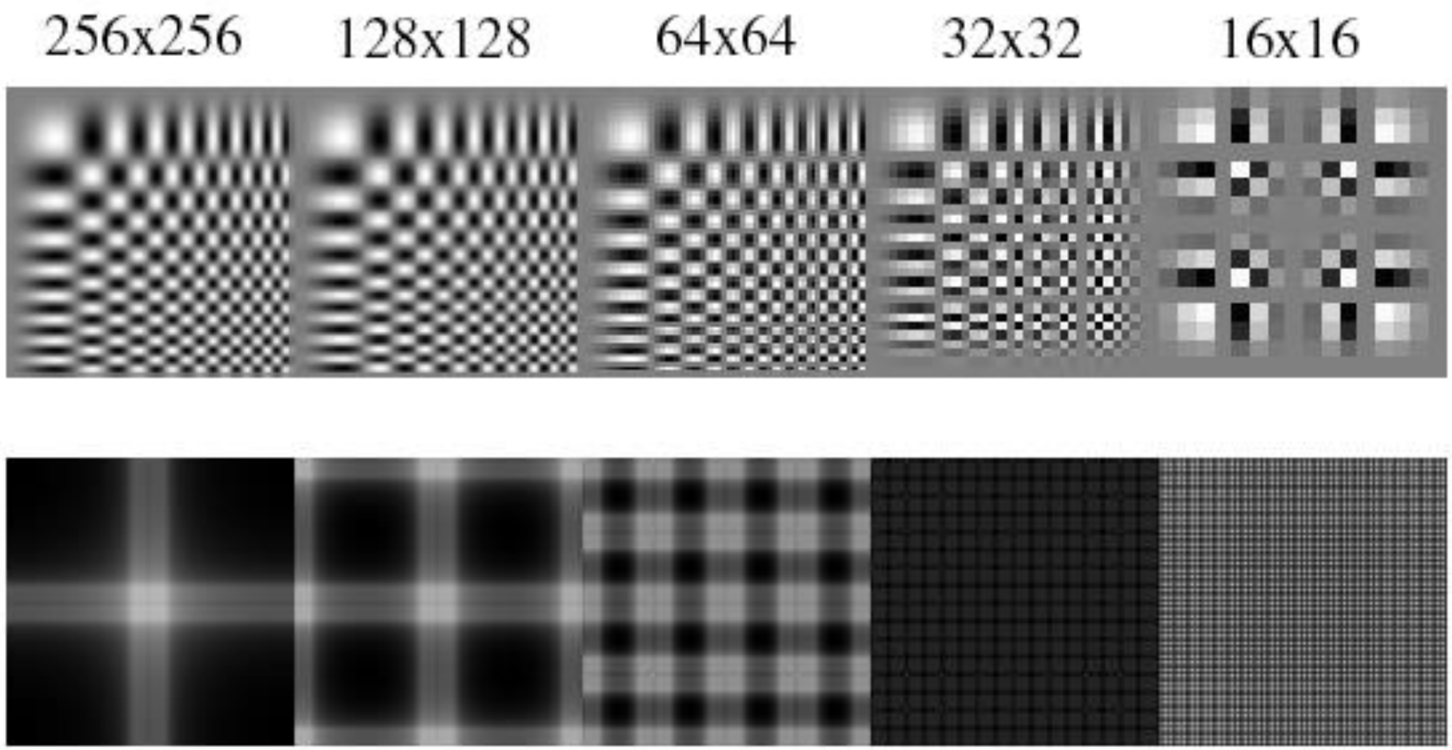


Slide from Dani Lischinski

Smoothing as low-pass filtering

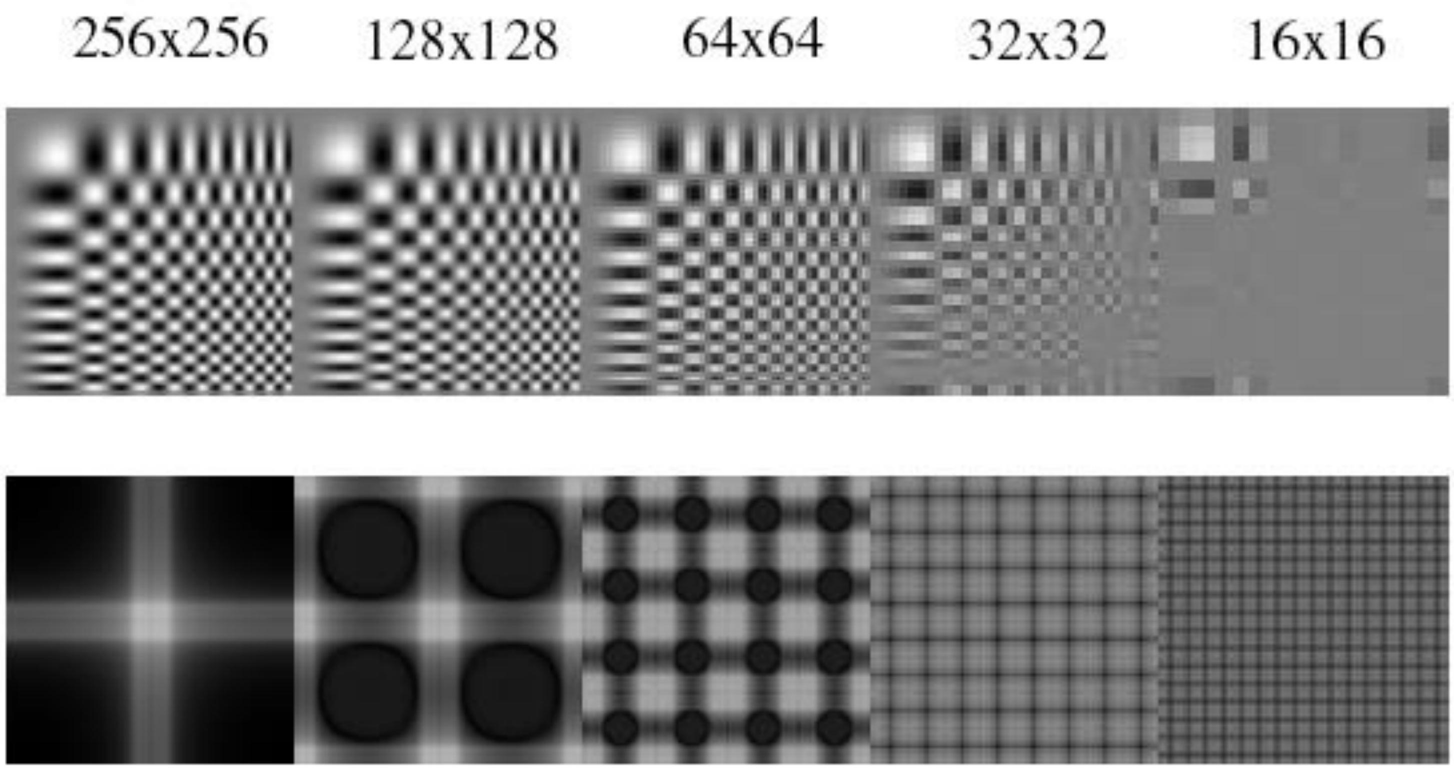
- The message of the FT is that high frequencies lead to trouble with sampling.
- Solution: suppress high frequencies before sampling
  - multiply the FT of the signal with something that suppresses high frequencies
  - or convolve with a low-pass filter
- A filter whose FT is a box is bad, because the filter kernel has infinite support
- Common solution: use a Gaussian
  - multiplying FT by Gaussian is equivalent to convolving image with Gaussian.

Sampling without smoothing.
Top row shows the images, sampled at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.



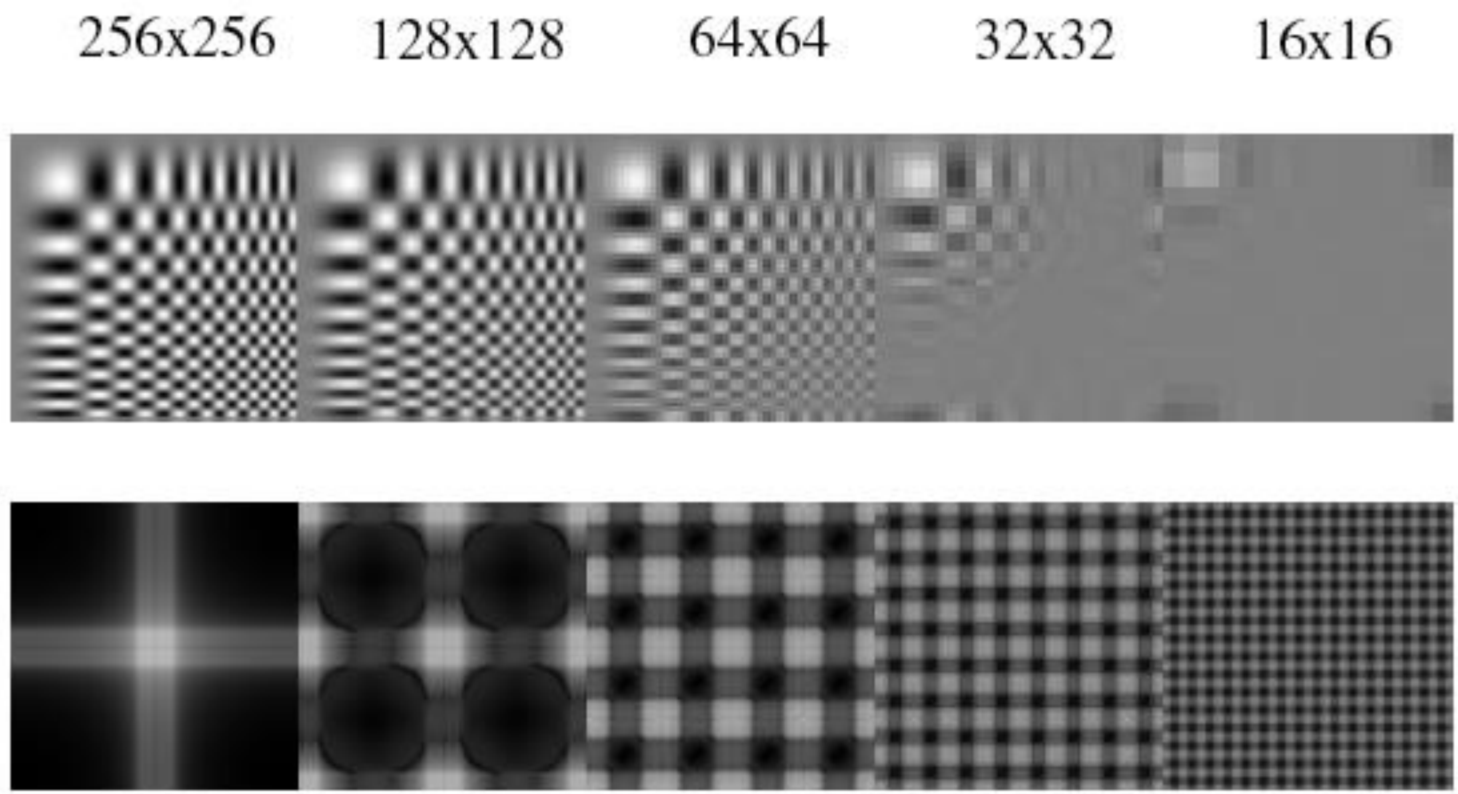
Sampling with smoothing.

Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1 pixel, then sampling at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.



Sampling with smoothing.

Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1.4 pixels, then sampling at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.

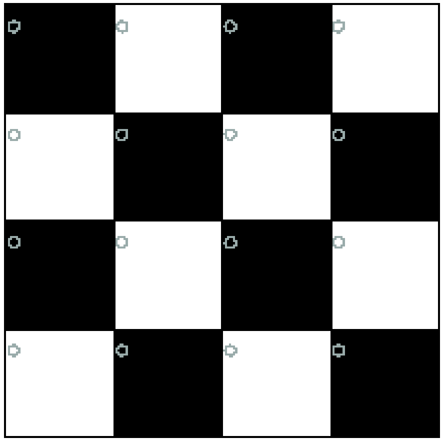


Nyquist sampling theorem

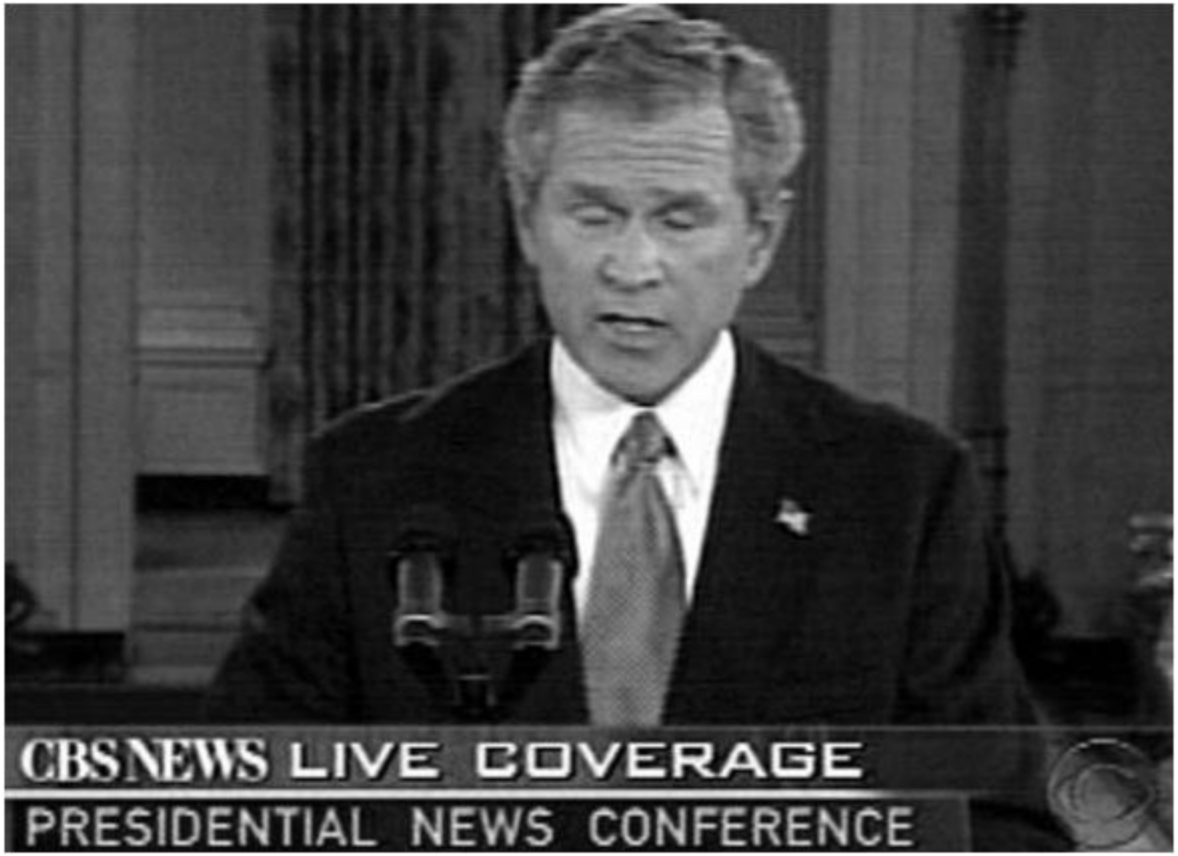
- Nyquist theorem: The sampling frequency must be at least twice the highest frequency

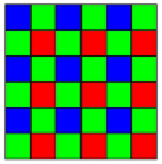
$$\omega_s \geq 2\omega$$

- If this is not the case the signal needs to be bandlimited before sampling, e.g. with a low-pass filter



What went wrong?





color sampled at half-resolution!

Signal reconstruction

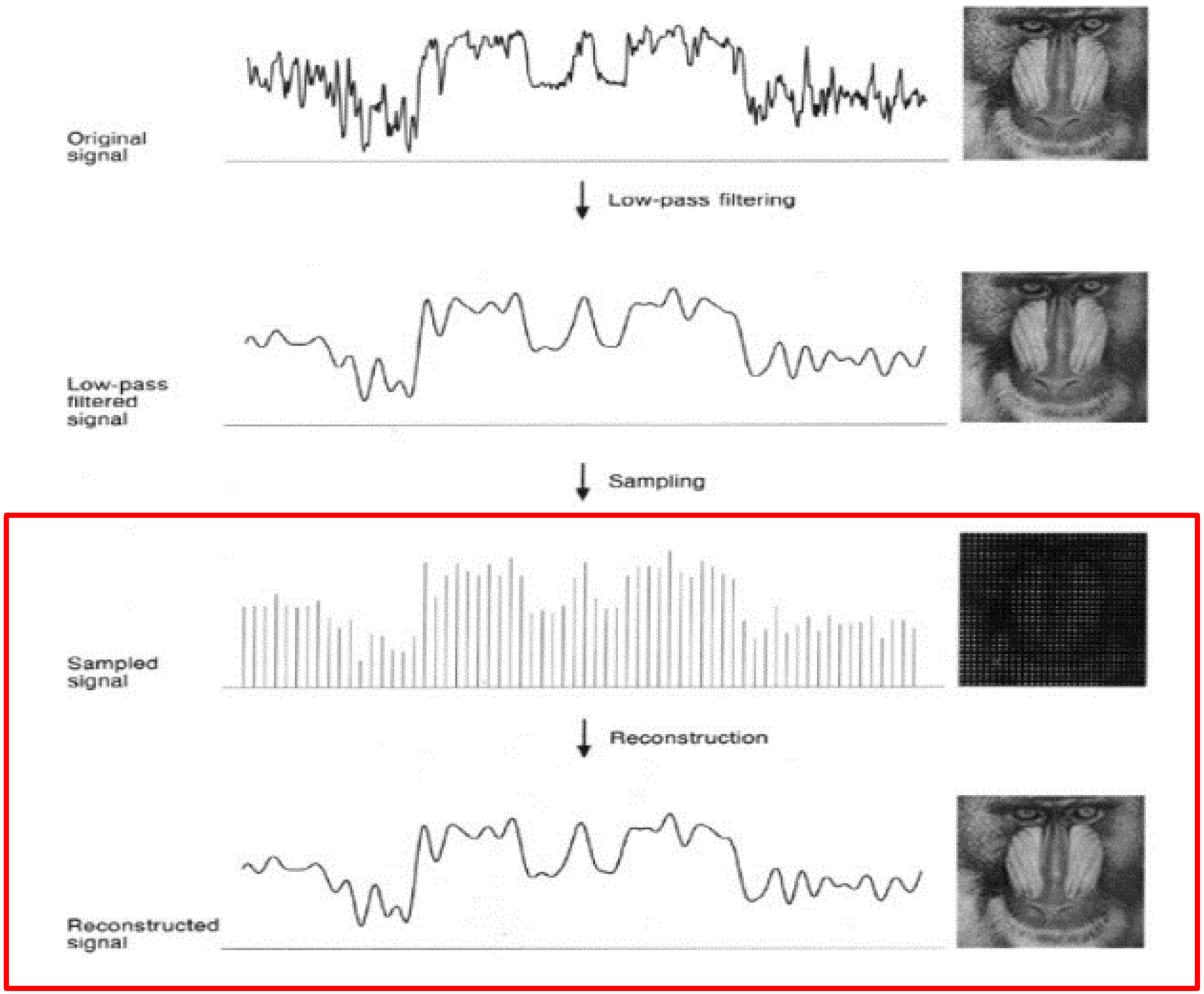
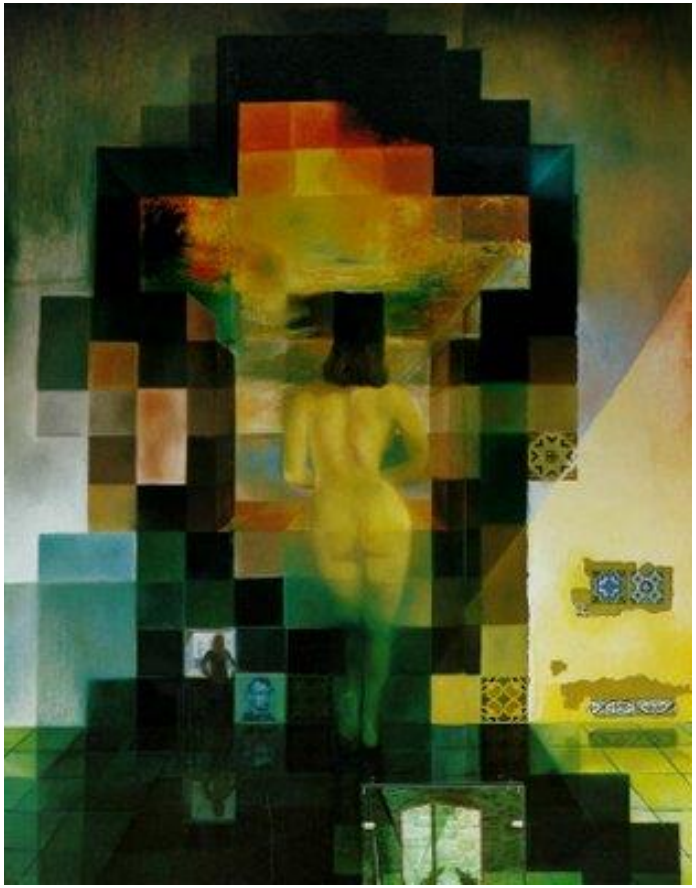


Image reconstruction: pixelization

- Who is this?

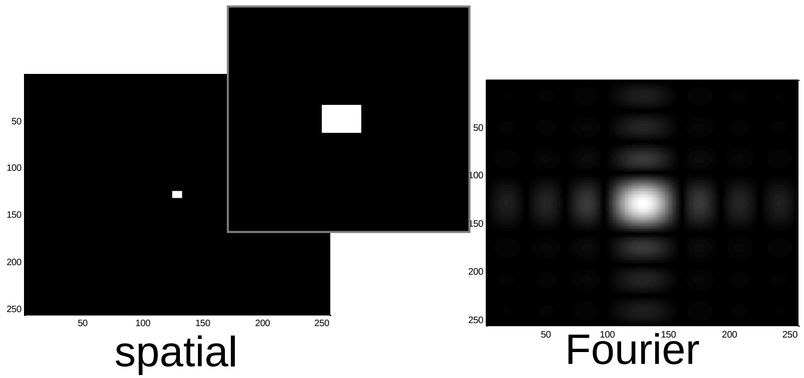
Harmon & Julesz 1973



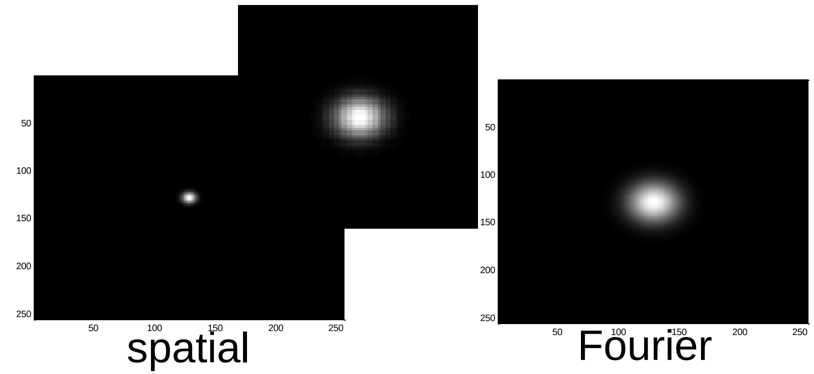
Dali 1976

Reconstruction filters

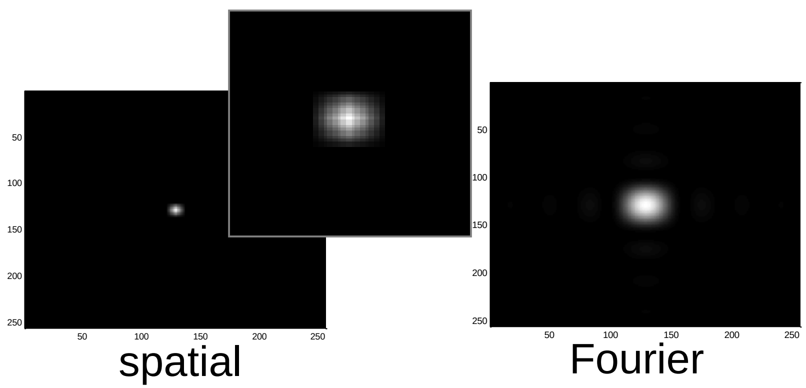
Square pixels



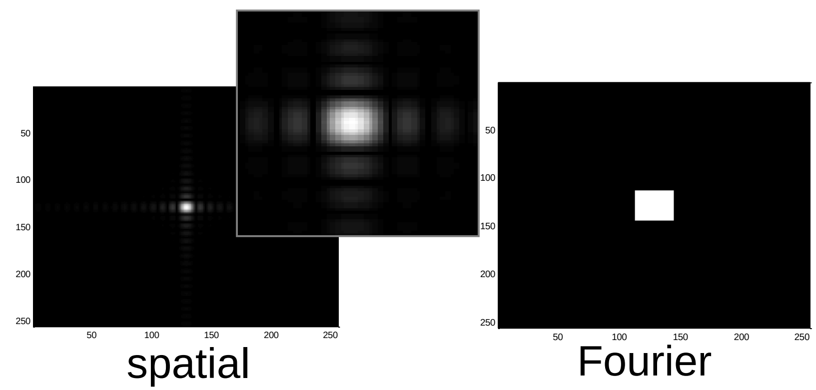
Gaussian reconstruction filter



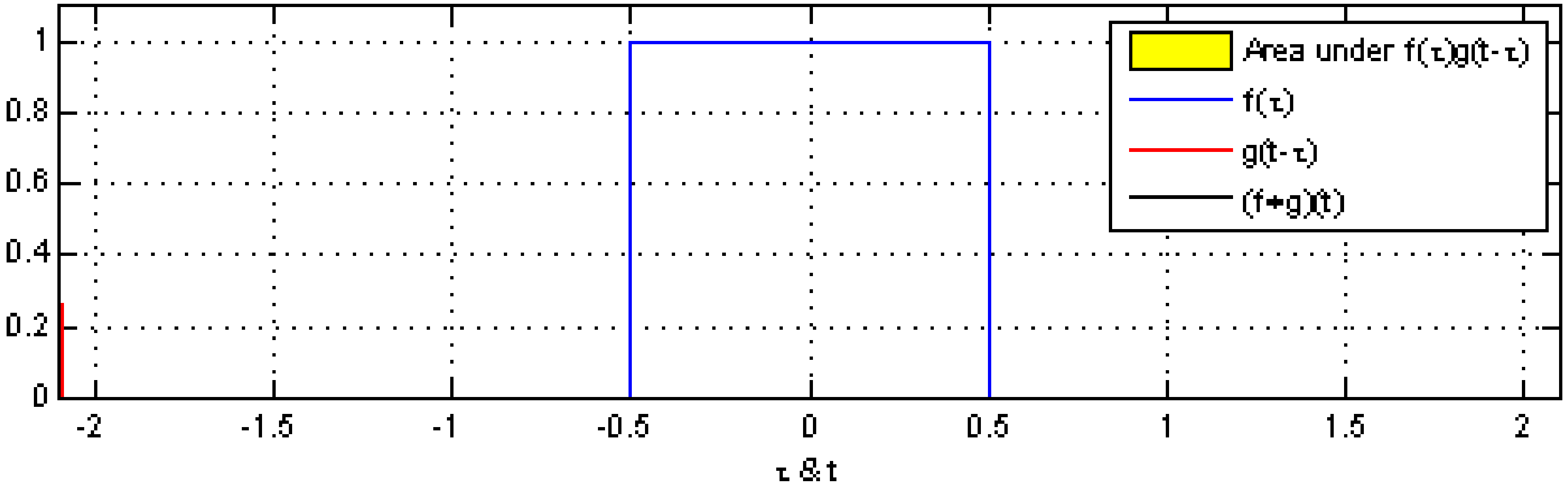
Bilinear interpolation



Perfect reconstruction filter

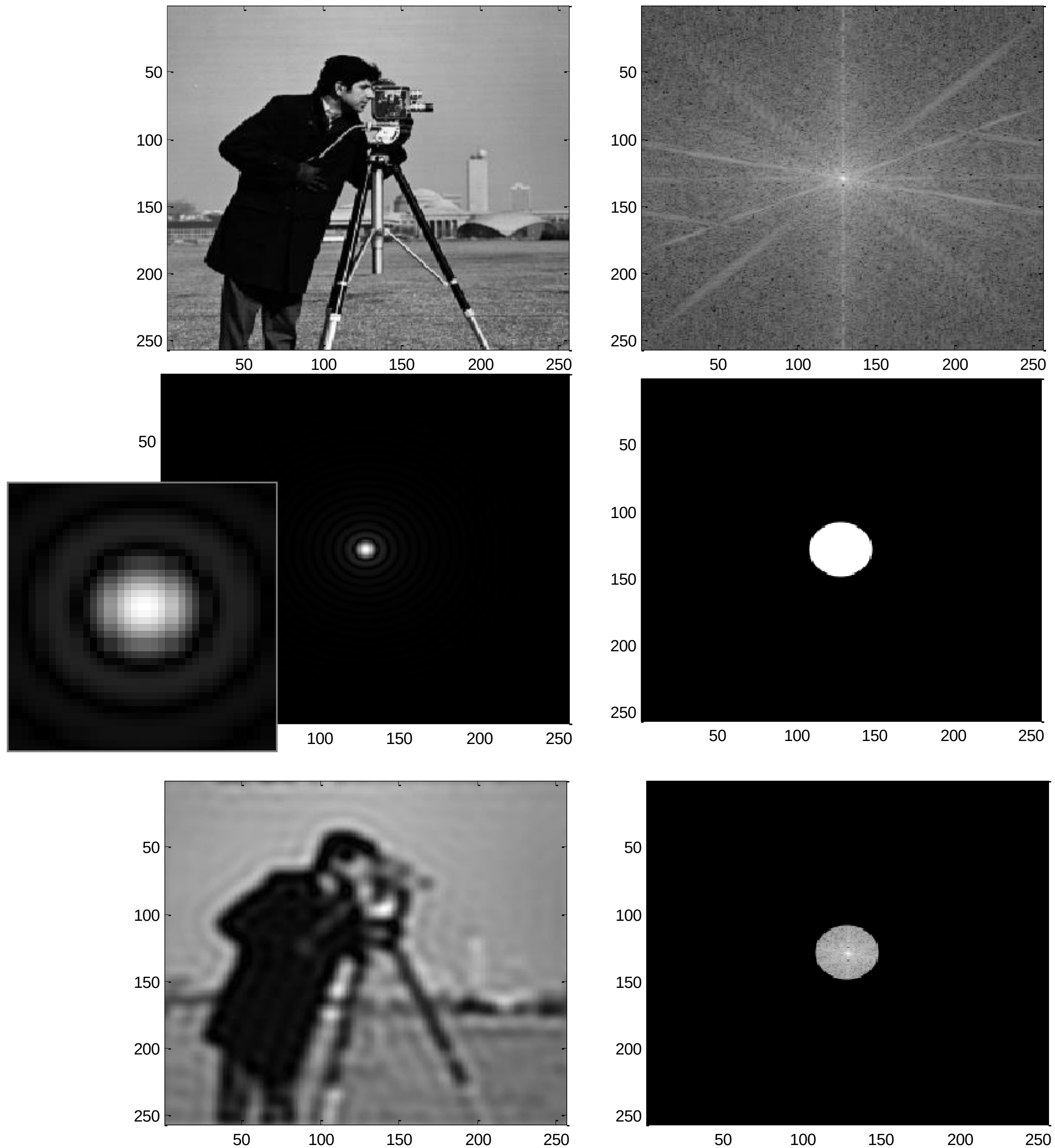


Convolution of Box with Box

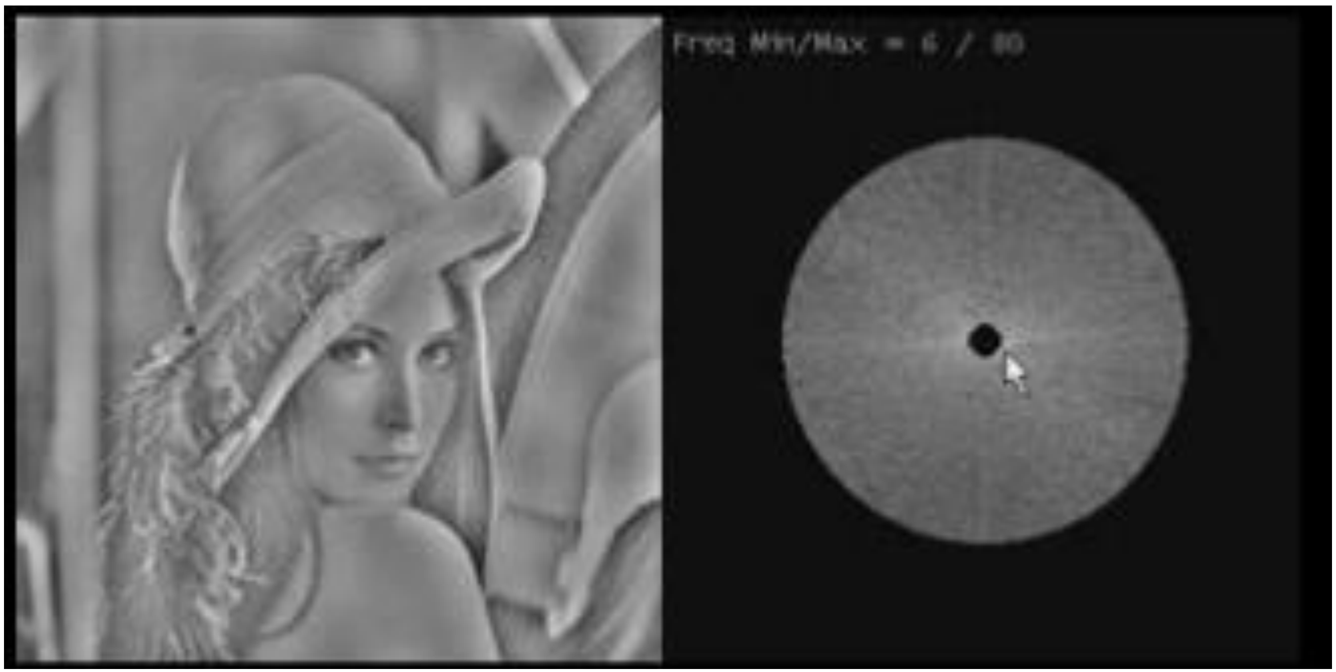


https://commons.wikimedia.org/wiki/File:Convolution_of_box_signal_with_itself2.gif

Designing the ‘perfect’ low-pass filter

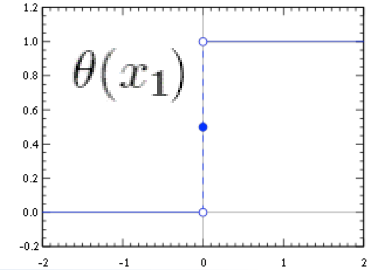


Filtering in Fourier domain



Defocus blurring

Motion blurring



Each light dot is transformed into a short line along the x_1-axis:

$$h(x_1, x_2) = \frac{1}{2l}\left[\theta(x_1 + l) - \theta(x_1 - l)\right]\delta(x_2)$$

Lena with blurring and noise

Gaussian blurring kernel:

$$h(x_1, x_2) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x_1^2 + x_2^2}{2\sigma^2})$$

Image restoration problem

$$f(\mathbf{x}) \longrightarrow \boxed{h(\mathbf{x})} \longrightarrow g(\mathbf{x}) \longrightarrow \boxed{\tilde{h}(\mathbf{x})} \longrightarrow f(\mathbf{x})$$

The 'inverse' kernel $\tilde{h}(\mathbf{x})$ should compensate the effect of the image degradation $h(\mathbf{x})$, i.e.,

$$(\tilde{h} * h)(\mathbf{x}) = \delta(\mathbf{x})$$

$\tilde{h}$ may be determined more easily in Fourier space:

$$\mathcal{F}[\tilde{h}](u,v) \cdot \mathcal{F}[h](u,v) = 1$$

To determine $\mathcal{F}[\tilde{h}]$ we need to estimate

1. the distortion model $h(\mathbf{x})$ (point spread function) or $\mathcal{F}[h](u,v)$ (modulation transfer function)
2. the parameters of $h(\mathbf{x})$, e.g. r for defocussing.

Image Restoration: Motion Blur

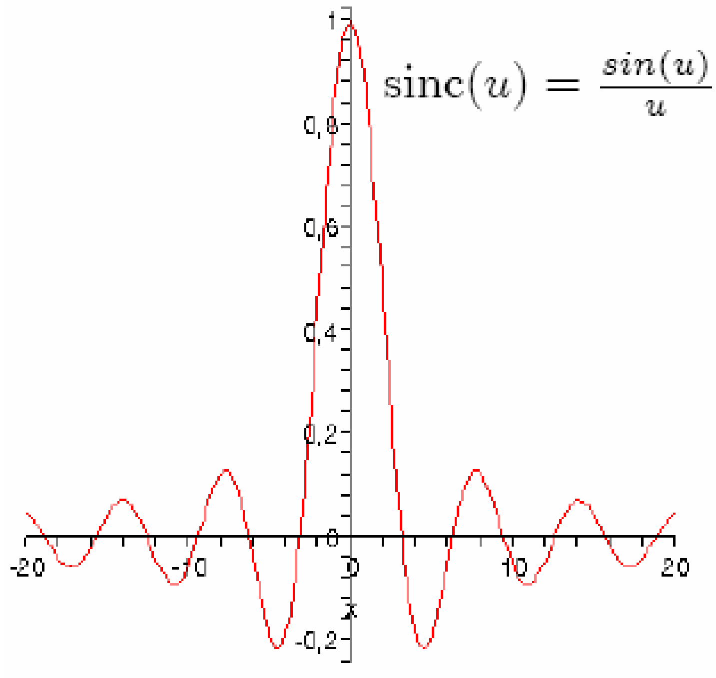
Kernel for motion blur $h(\mathbf{x}) = \frac{1}{2l}\big(\theta(x_1 + l) - \theta(x_1 - l)\big)\delta(x_2)$

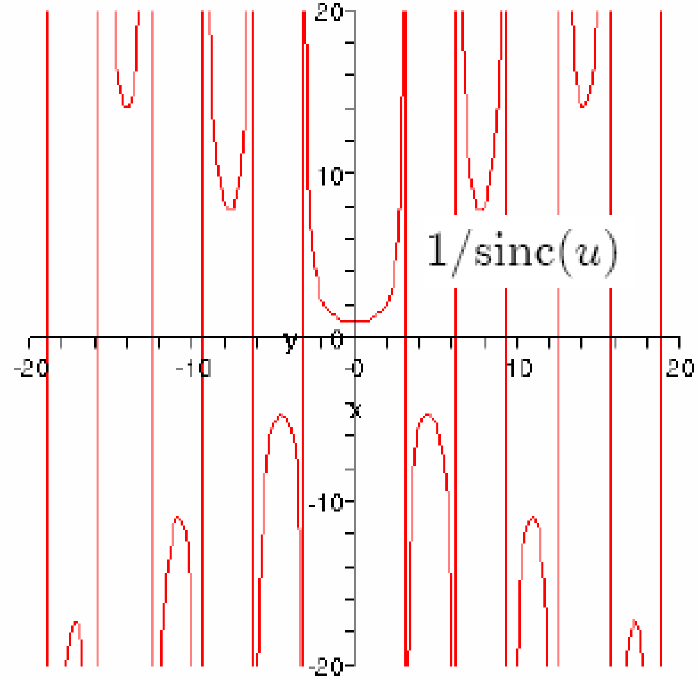
(a light dot is transformed into a small line in x_1 direction).

Fourier transformation:

$$\begin{aligned}\mathcal{F}[h](u,v) &= \frac{1}{2l}\int_{-l}^{+l} \exp(-\imath 2\pi u x_1) \underbrace{\int_{-\infty}^{+\infty} \delta(x_2)\exp(-\imath 2\pi v x_2) dx_2}_{=1} dx_1 \\ &= \frac{\sin(2\pi u l)}{2\pi u l} =: \operatorname{sinc}(2\pi u l)\end{aligned}$$

ETH





$$\hat{h}(u) = \mathcal{F}[h](u) = \mathrm{sinc}(2\pi u l) \qquad \mathcal{F}[\tilde{h}](u) = 1/\hat{h}(u)$$

Problems:

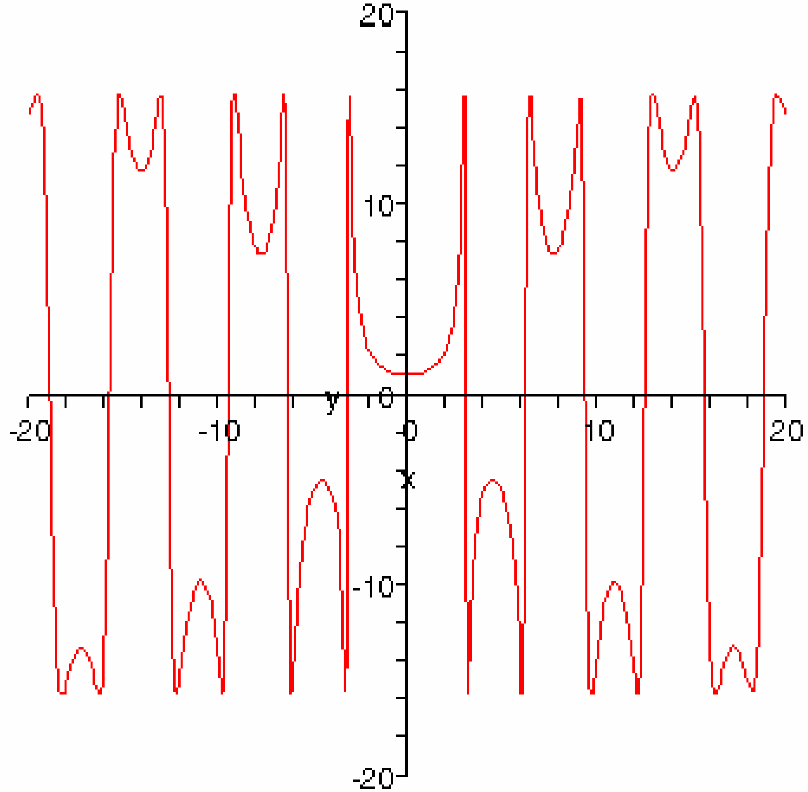
- Convolution with the kernel h completely cancels the frequencies $\frac{\nu}{2l}$ for $\nu \in \mathcal{Z}$. Vanishing frequencies cannot be recovered!
- Noise amplification for $\mathcal{F}[h](u, v) \ll 1$.

Avoiding noise amplification

Regularized reconstruction filter:

$$\tilde{\mathcal{F}}[\tilde{h}](u,v) = \frac{\mathcal{F}[h]}{|\mathcal{F}[h]|^2 + \epsilon}$$

Singularities are avoided by the regularization ϵ.



The size of ϵ implicitly determines an estimate of the noise level in the image, since we discard signals which are dampened below the size ϵ.

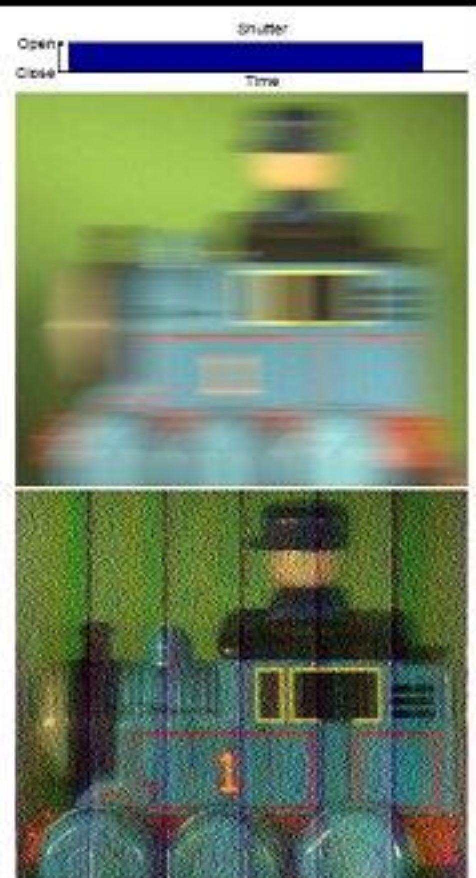
Coded Exposure Photography:
Assisting Motion Deblurring using Fluttered Shutter

Raskar, Agrawal, Tumblin (Siggraph2006)

Short Exposure

Image is dark and noisy

Traditional



Result has Banding Artifacts and some spatial frequencies are lost

← Shutter →

← Captured Photos →

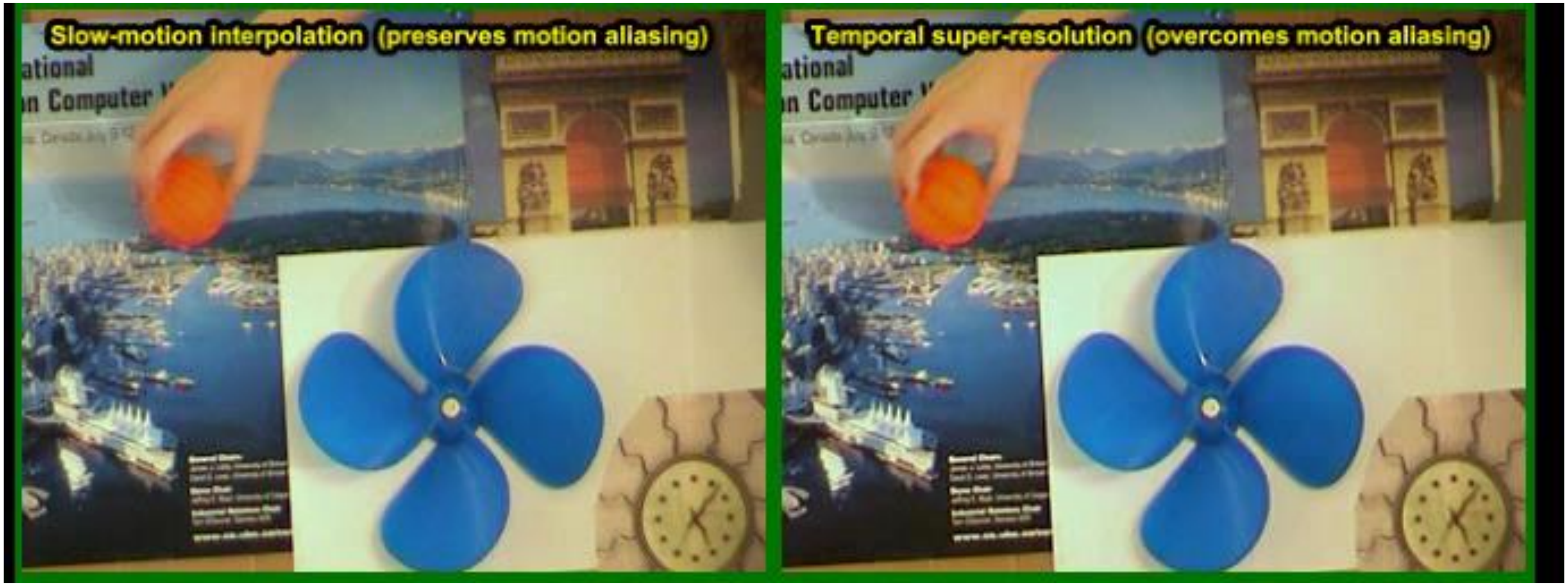
← Deblurred Results →

Coded

Decoded image is as good as image of a static scene

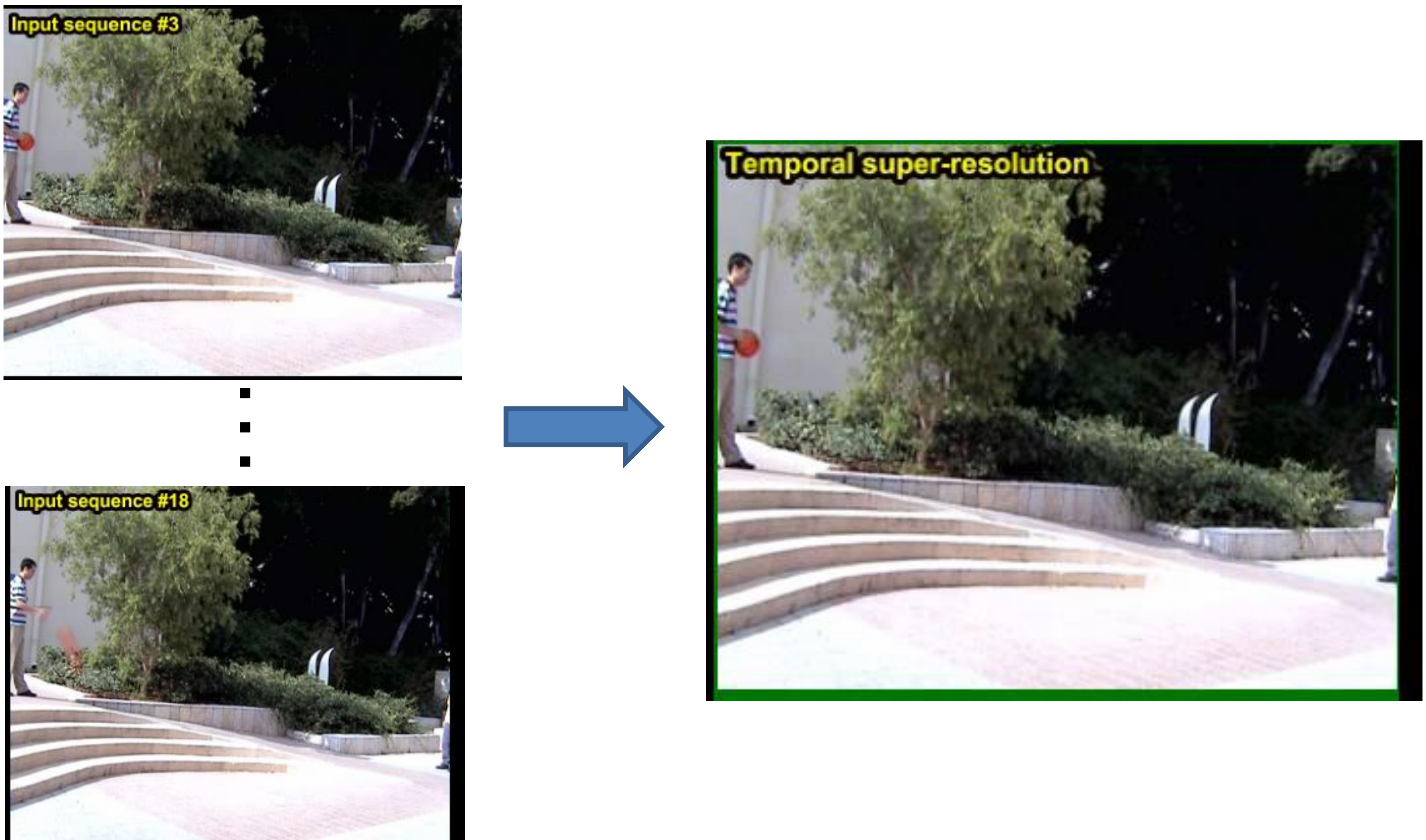
Space-time super-resolution

Shechtman et al. PAMI05



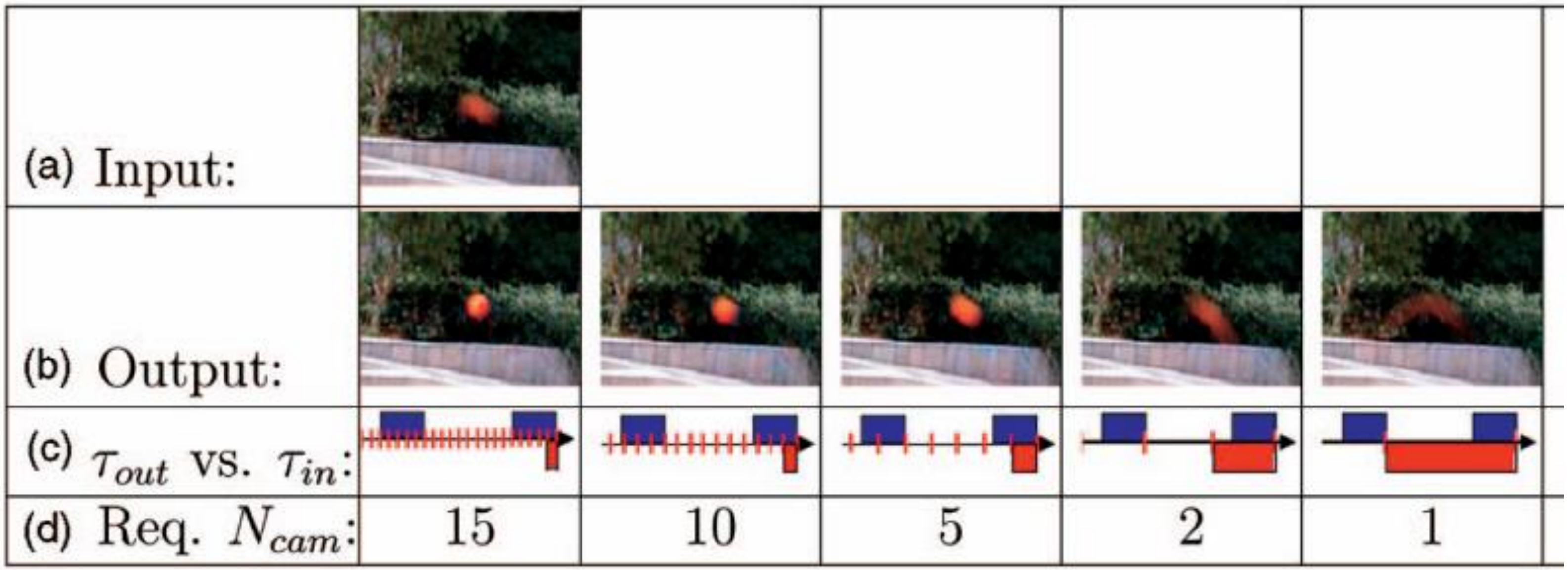
Space-time super-resolution

Shechtman et al. PAMI05

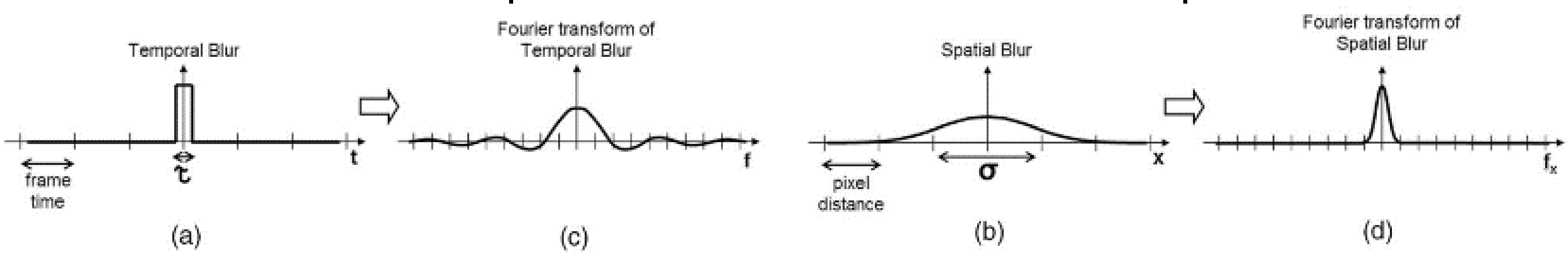


Space-time super-resolution

Shechtman et al. PAMI05

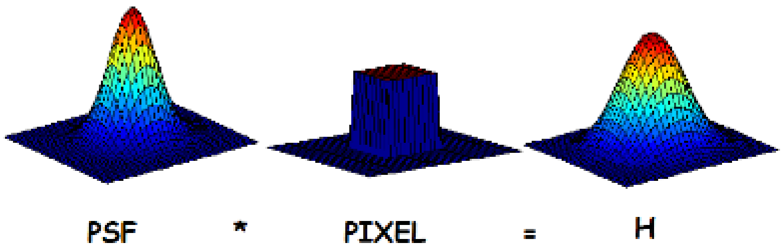


time super-resolution works better than space



Spatial super-resolution

- lens+pixel=low-pass filter (desired to avoid aliasing)



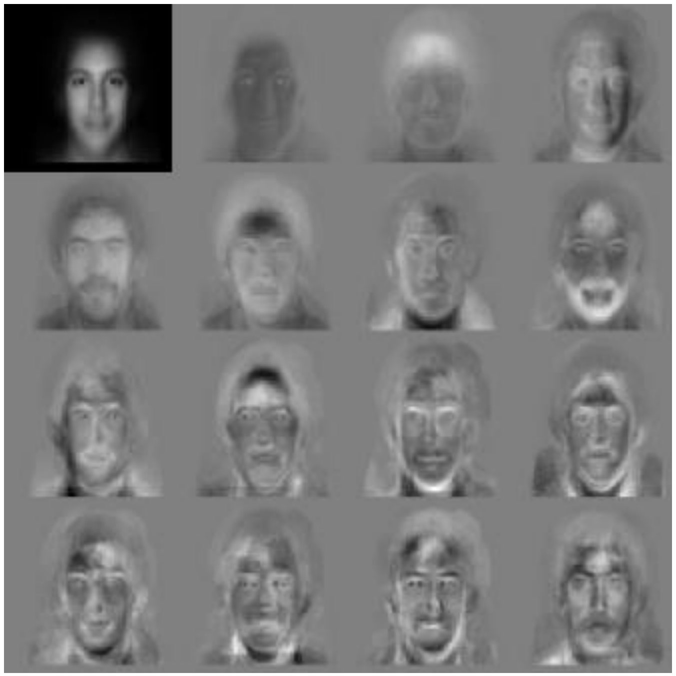
- Low-res images = D*H*G*(desired high-res image)
  - D: decimate, H:lens+pixel, G: Geometric warp
- Simplified case for translation: LR=(D*G)*(H*HR)
  - G is shift-invariant and commutes with H
  - First compute H*HR, then deconvolve HR with H
- Super-resolution needs to restore attenuated frequencies
  - Many images improve S/N ratio (~sqrt(n)), which helps
  - Eventually Gaussian's double exponential always dominates

ETH

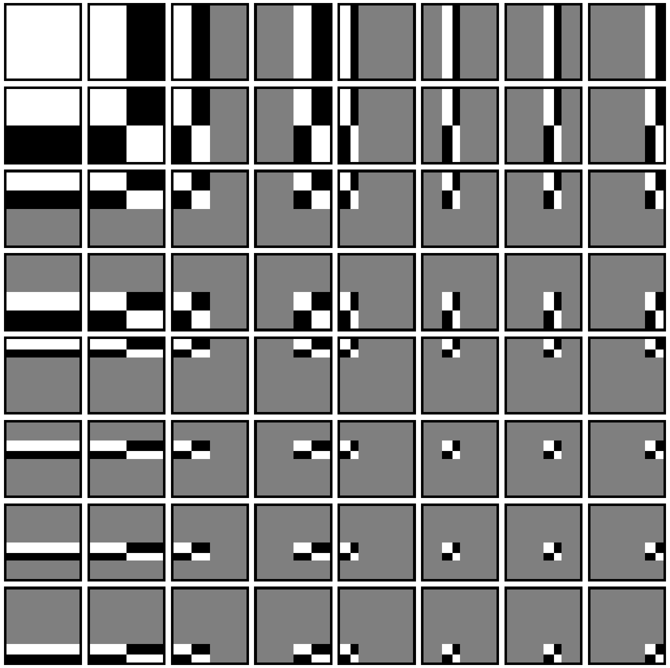
Next week:

More Image Transformations

Eigenfaces



Wavelets



- Maybe a bit too short, explain better super-reslution with added noise, etc. Lena+gaussian noise slide is funny…
- Defocus bluring slide also a bit funny…
- Make slides to explain reconstruction kernels as convolution of boxes…