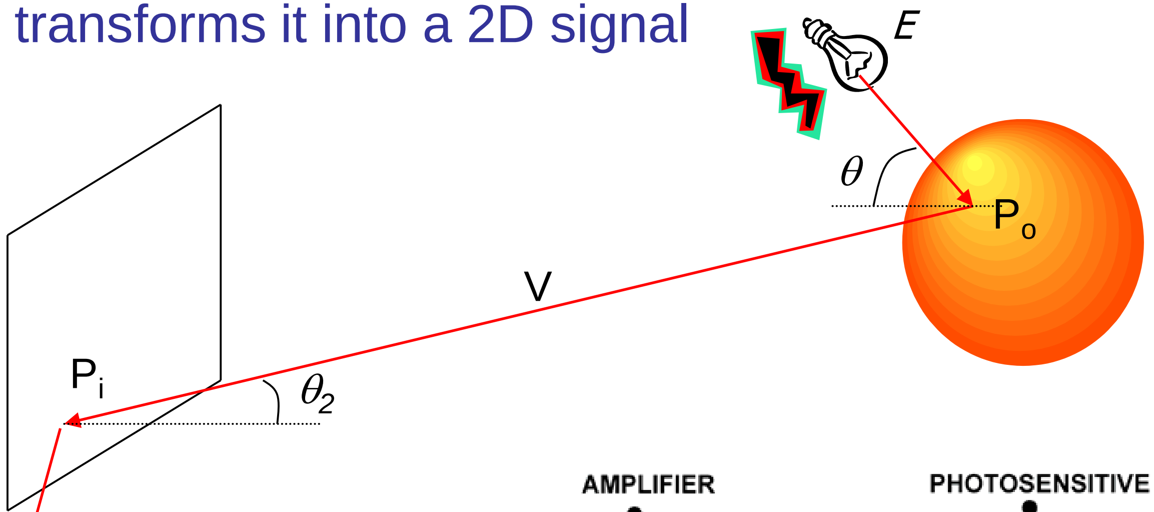
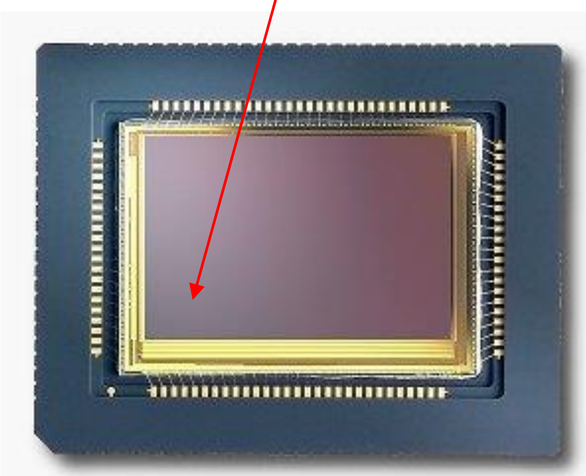
2D DSP

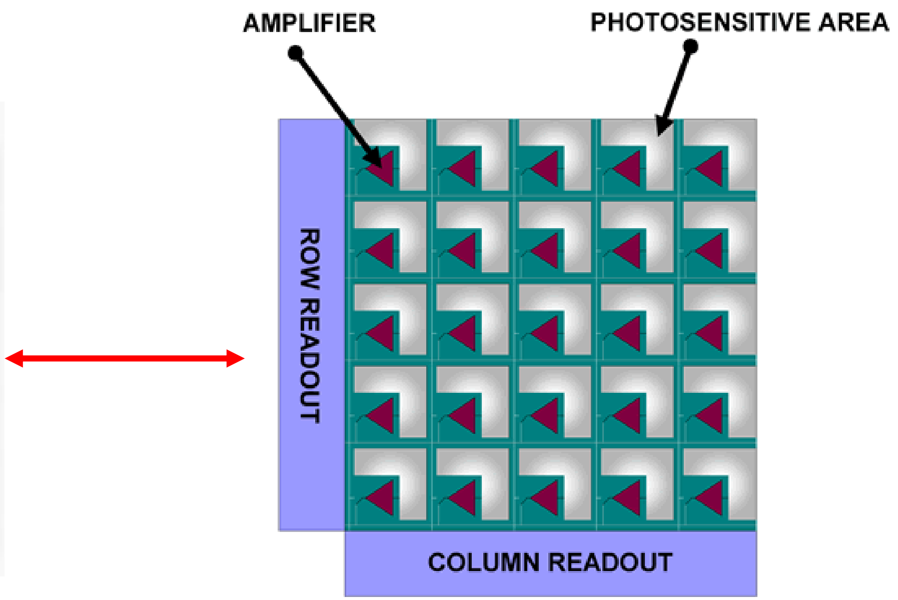
Nuno Vasconcelos
UCSD

Images

- the incident light is collected by an image sensor
  - that transforms it into a 2D signal







2D-DSP

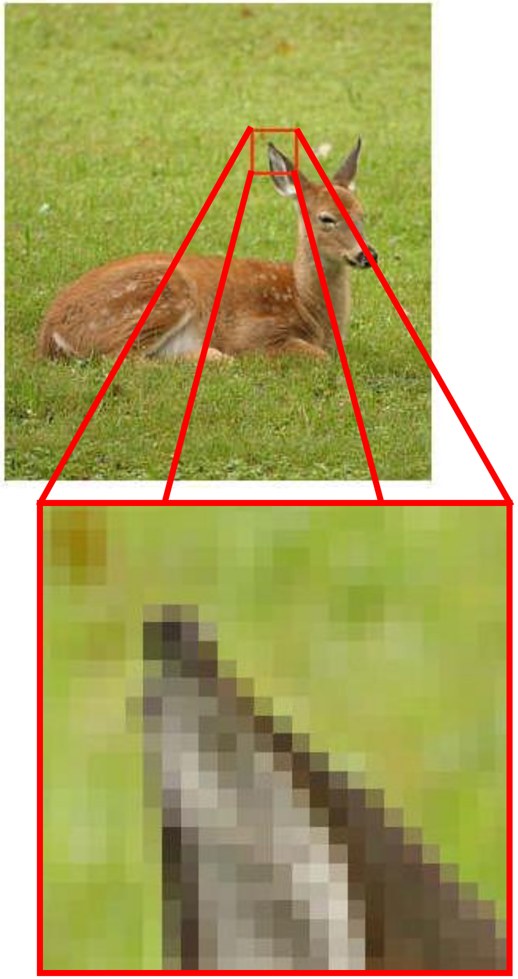
- in summary:
  - image is a *N x M* array of pixels
  - each pixel contains three colors
  - overall, the image is a 2D discrete-space signal
  - each entry is a 3D vector

$$x[n_1, n_2] = (r, g, b), \; n_1 \in \{0, \ldots, N\}$$
$$n_2 \in \{0, \ldots, M\}$$

  - for simplicity, we consider only single channel images

$$x[n_1, n_2], \; n_1 \in \{0, \ldots, N\}$$
$$n_2 \in \{0, \ldots, M\}$$

  - but everything extends to color in a straightforward manner



Separable sequences

- a trivial concept,
  - but probably the only real novelty in this lecture
  - very important in practice, because it reduces 2D problem to collection on 1D problems

- **Definition:** a sequence is separable if and only if

$$x[n_1, n_2] = f[n_1] \times g[n_2]$$

where *f[.]* and *g[.]* are 1D functions

- note: there are many examples of separable sequences
- but most sequences are not separable

Linear Shift Invariant (LSI) systems

- straightforward extension of LTI systems

- **Definition:** a system T that maps $x[n_1,n_2]$ into $y[n_1,n_2]$ is LSI if and only if
  - it is linear

$$
\begin{aligned}
T\{ax_1[n_1,n_2]+bx_2[n_1,n_2]\} &= \\
&= aT\{x_1[n_1,n_2]\}+bT\{x_2[n_1,n_2]\} \\
&= ay_1[n_1,n_2]+by_2[n_1,n_2]
\end{aligned}
$$

  - it is shift invariant

$$
T\{x[n_1-m_1,n_2-m_2]\}=y[n_1-m_1,n_2-m_2]
$$

2D convolution

- the operation

$$y[n_1,n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\, h[n_1-k_1, n_2-k_2]$$

  is the 2D convolution of x and h
  - we will denote it by

$$y[n_1,n_2] = x[n_1,n_2] * h[n_1,n_2]$$

- this is of great practical importance:
  - for an LSI system the response to any input can be obtained by the convolution with this impulse response
  - the IR fully characterizes the system
  - it is all that I need to measure

2D convolution

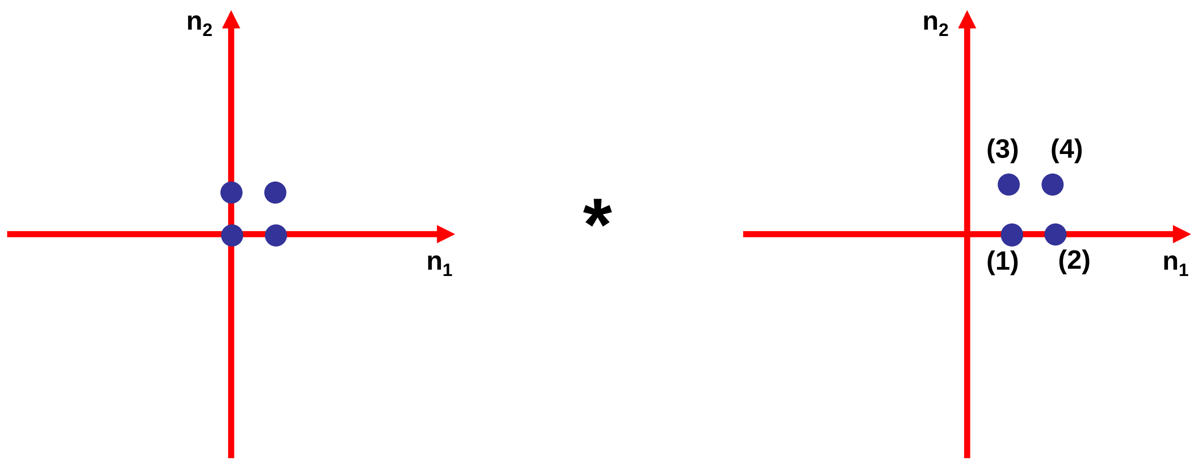
- has various properties of interest
- but these are the ones that you have already seen in 1D (check handout)
- some of the more important:
  - commutative: $x * y = y * x$
  - associative: $(x * y) * z = x * (y * z)$
  - distributive: $x * (y + z) = x * y + x * z$
  - convolution with impulse:

$$x[n_1, n_2] * \delta[n_1 - m_1, n_2 - m_2] = x[n_1 - m_1, n_2 - m_2]$$

2D convolution

- as in 1D, it is most easily done in graphical form
- e.g. how do we convolve these two sequences?

$$y[n_1,n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\, h[n_1-k_1, n_2-k_2]$$



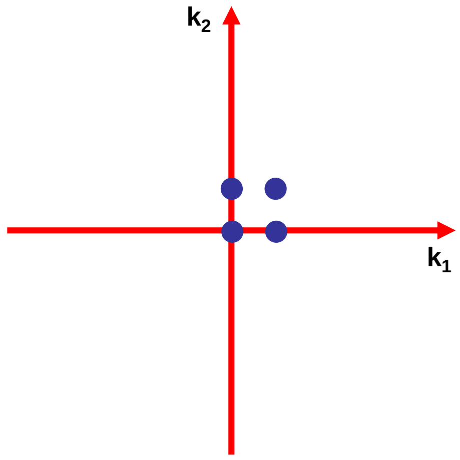
$x[n_1,n_2]$ $h[n_1,n_2]$

- we need four steps

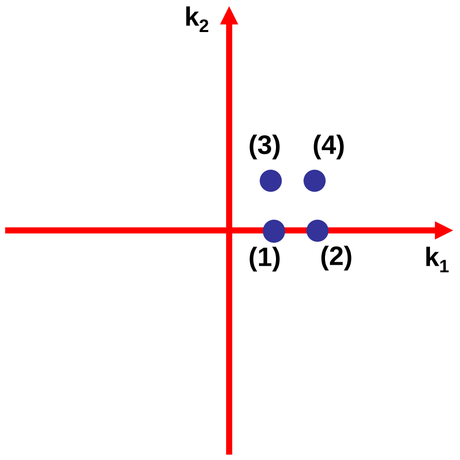
2D convolution

- **step 1):** express sequences in terms of *(k_1, k_2)*

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]\,h[n_1-k_1,n_2-k_2]$$



$x[k_1,k_2]$

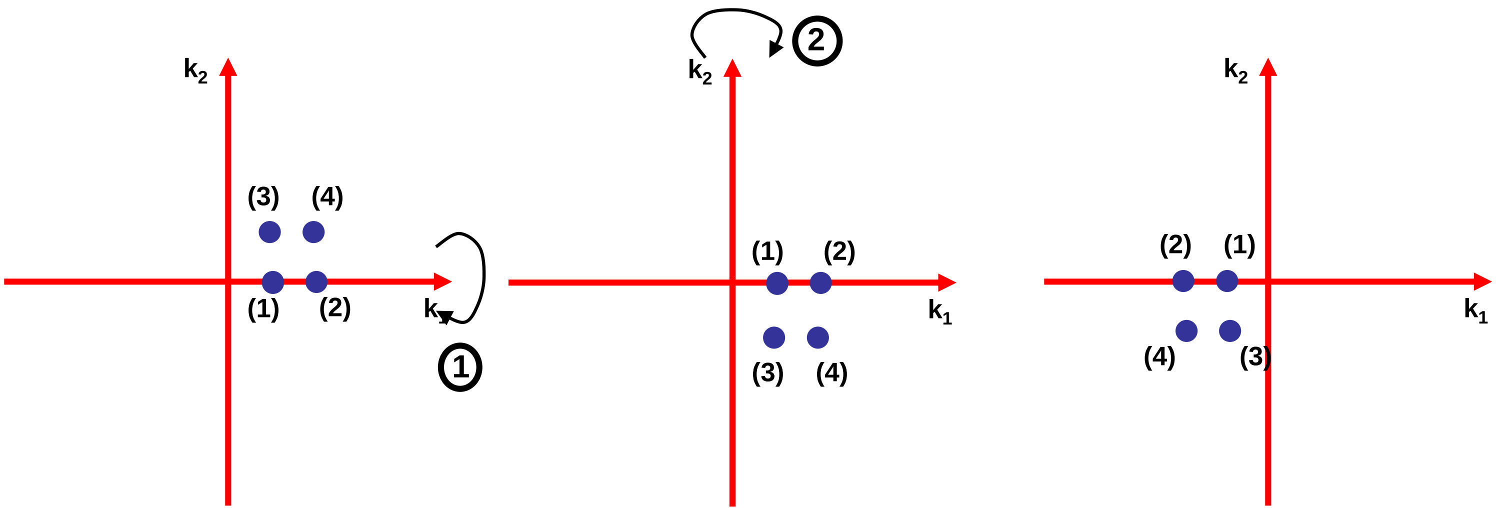


$h[k_1,k_2]$

2D convolution

- **step 2):** invert $h(k_1, k_2)$

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]\,h[n_1-k_1,n_2-k_2]$$

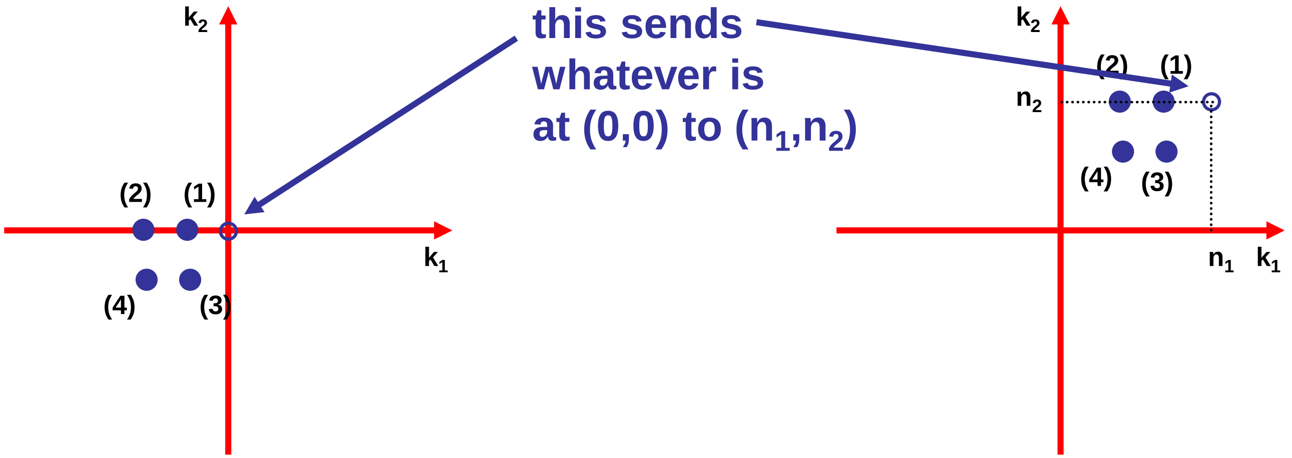


$h[k_1,k_2]$ $\qquad$ $h[k_1,-k_2]$ $\qquad$ $g[k_1,k_2]=h[-k_1,-k_2]$

2D convolution

- **step 3):** shift *$g(k_1, k_2)$* by *(n_1, n_2)*

$$y[n_1,n_2] = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\, h[n_1-k_1, n_2-k_2]$$



$$g[k_1,k_2] = h[-k_1,-k_2]$$

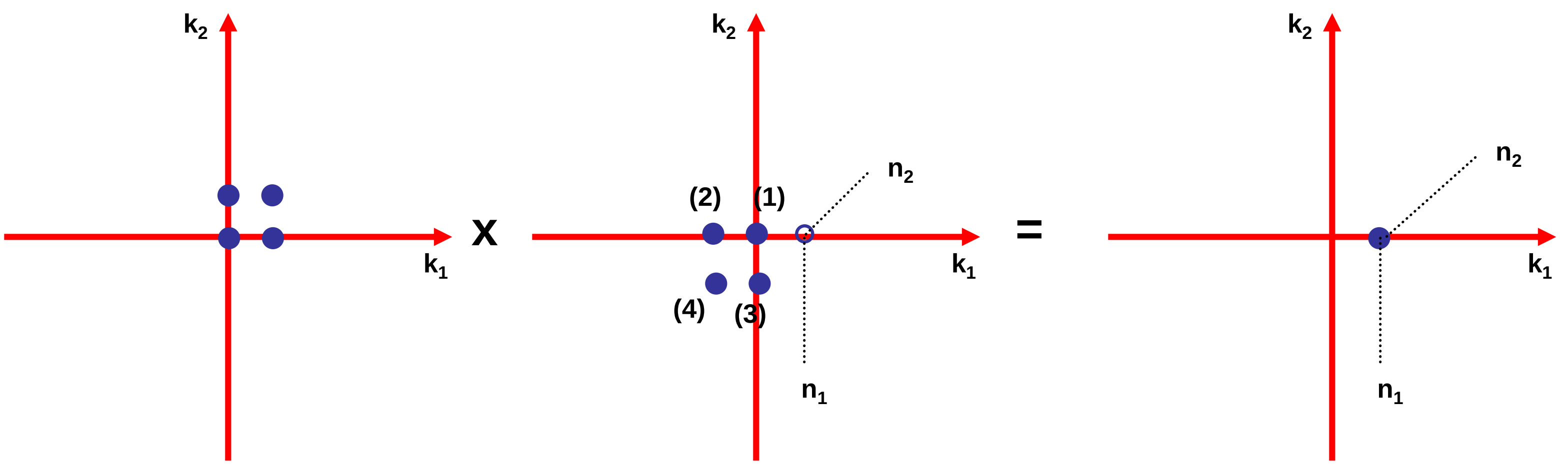
$$g[k_1-n_1, k_2-n_2] = h[n_1-k_1, n_2-k_2]$$

2D convolution

- **step 4):** point-wise multiply the two signals and sum

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]\,h[n_1-k_1,n_2-k_2]$$

  – e.g. for $(n_1,n_2) = (1,0)$



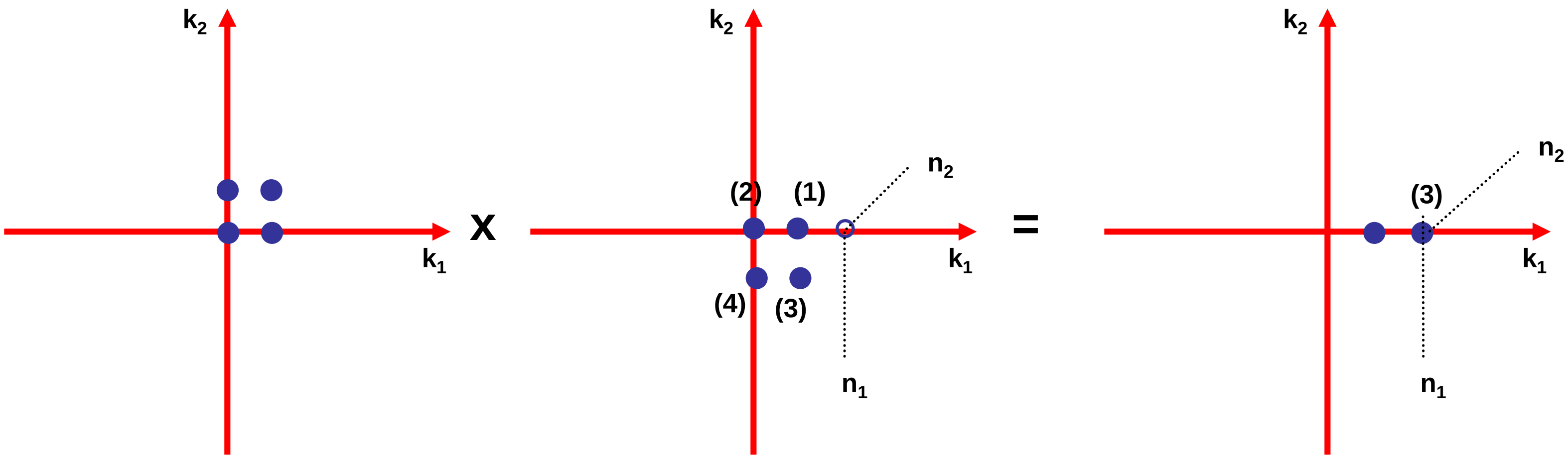
$x[k_1,k_2]$ $\quad$ $h[n_1-k_1,n_2-k_2]$ $\quad$ $y[n_1,n_2]$

2D convolution

- **step 4):** point-wise multiply the two signals and sum

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h[n_1 - k_1, n_2 - k_2]$$

  - e.g. for $(n_1,n_2) = (2,0)$



$x[k_1, k_2]$ $h[n_1 - k_1, n_2 - k_2]$ $y[n_1, n_2]$

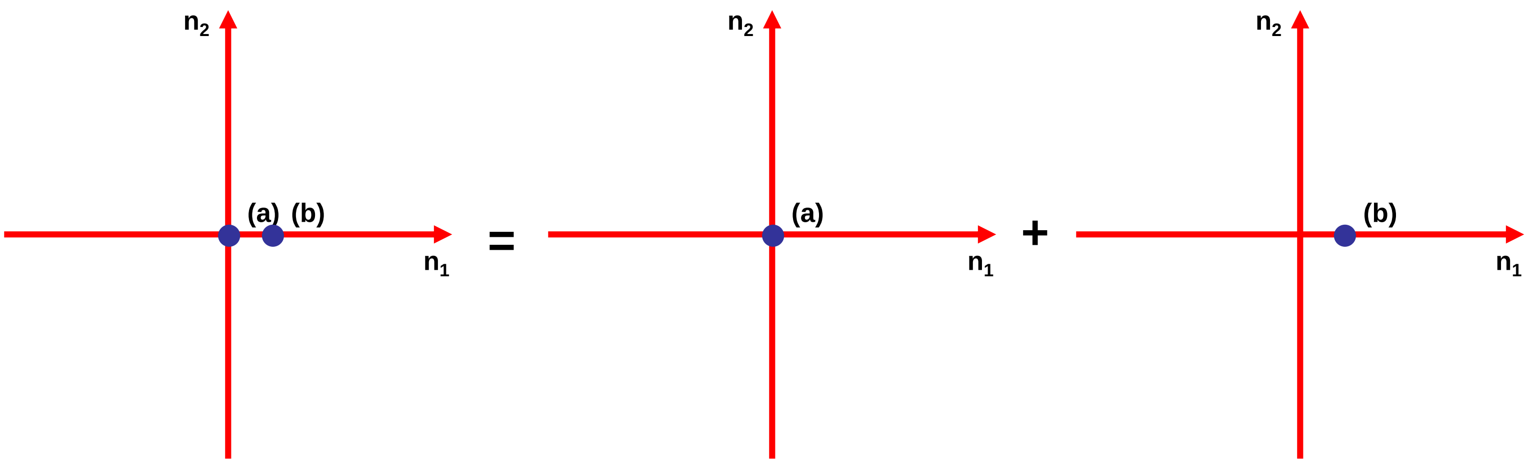
etc.

2D convolution

- is this the only way to look at convolution?
  - any signal can be written as

$$x[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2] \delta[n_1 - k_1, n_2 - k_2]$$

  - e.g.



$$a\delta[n_1, n_2] + b\delta[n_1 - 1, n_2]$$

$$a = x[0,0] \qquad b = x[1,0]$$

2D convolution

- we combine this

$$x[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\delta[n_1 - k_1, n_2 - k_2]$$

  with the properties of the convolution
  - commutative: $x * y = y * x$
  - associative: $(x * y) * z = x * (y * z)$
  - distributive: $x * (y + z) = x * y + x * z$
  - convolution with impulse:

$$x[n_1, n_2] * \delta[n_1 - m_1, n_2 - m_2] = x[n_1 - m_1, n_2 - m_2]$$

- to obtain another interpretation

2D convolution

- it is done like this

$$y[n_1,n_2] = x[n_1,n_2] * h[n_1,n_2]$$

$$= \left(\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\delta[n_1-k_1,n_2-k_2] \right) * h[n_1,n_2]$$

$$y[n_1,n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\left(\delta[n_1-k_1,n_2-k_2] * h[n_1,n_2]\right)$$

- note that this is just our definition of convolution

$$x[n_1,n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\delta[n_1-k_1,n_2-k_2]$$

(no surprises here)

- but we want to think about it like this, not like this

2D convolution

- we can think of

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]\left(\delta[n_1-k_1,n_2-k_2]*h[n_1,n_2]\right)$$

as the following sequence of operations

1. set $y[n_1, n_2] = 0$, for all (n_1,n_2)
2. for each (k_1,k_2) such that $x[k_1,k_2]$ is not zero
   - set $\alpha = x[k_1,k_2]$
   - shift $h[n_1,n_2]$ by (k_1,k_2)
   - multiply the entire sequence by α

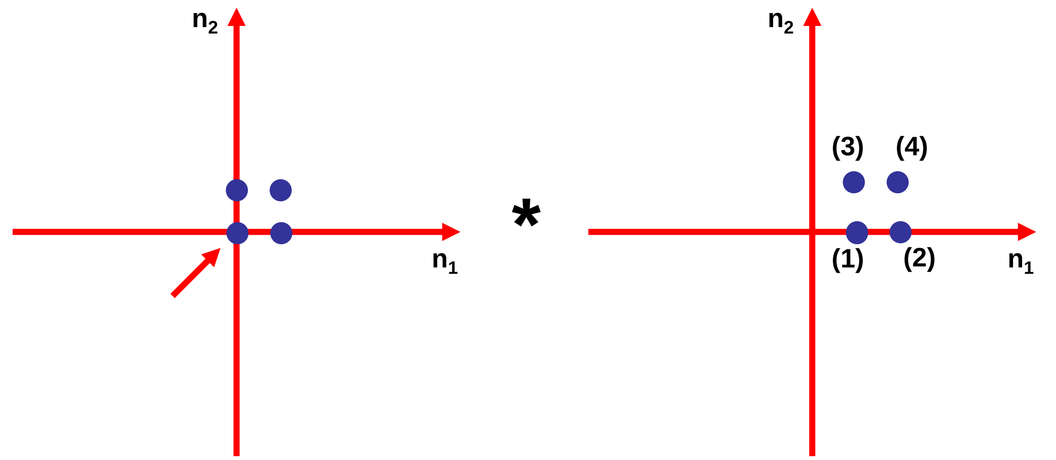
$$z[n_1,n_2]=\alpha\left(\delta[n_1-k_1,n_2-k_2]*h[n_1,n_2]\right)$$

   - add the entire sequence to $y[n_1,n_2]$

$$y[n_1,n_2]=y[n_1,n_2]+z[n_1,n_2]$$

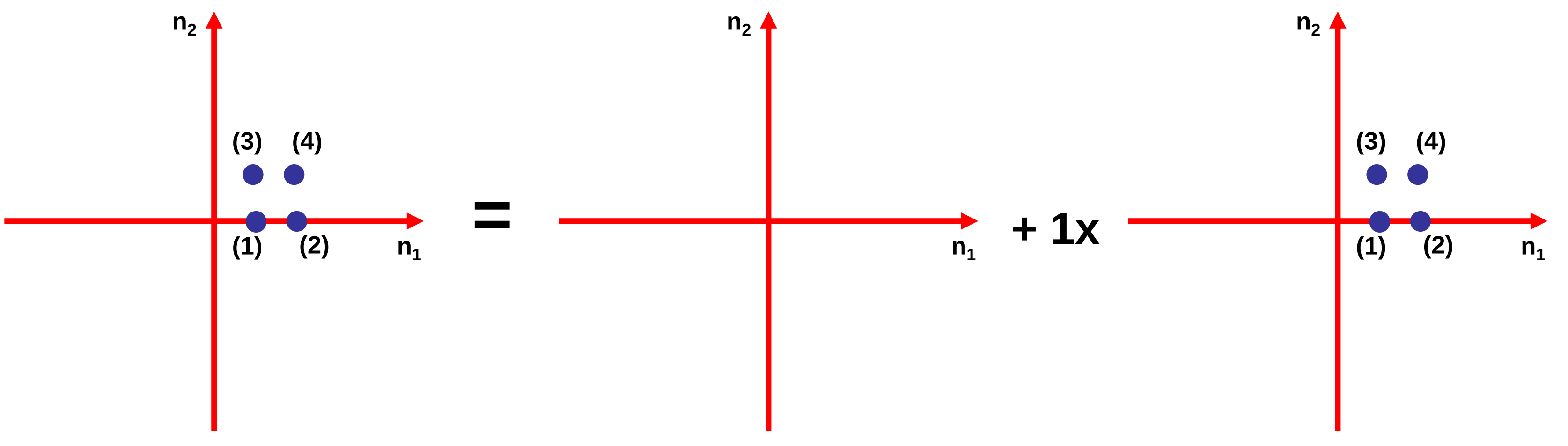
2D convolution

- example



$x[k_1, k_2]$ $\quad$ $h[n_1, n_2]$

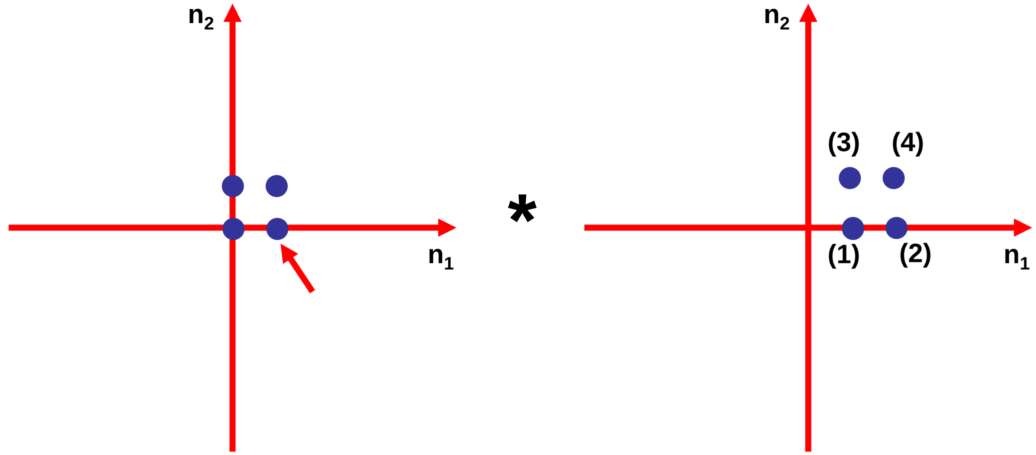
- $(k_1,k_2) = (0,0)$



$y[n_1, n_2]$ $\quad$ $y[n_1, n_2]$ $\quad$ $h[n_1 - k_1, n_2 - k_2]$

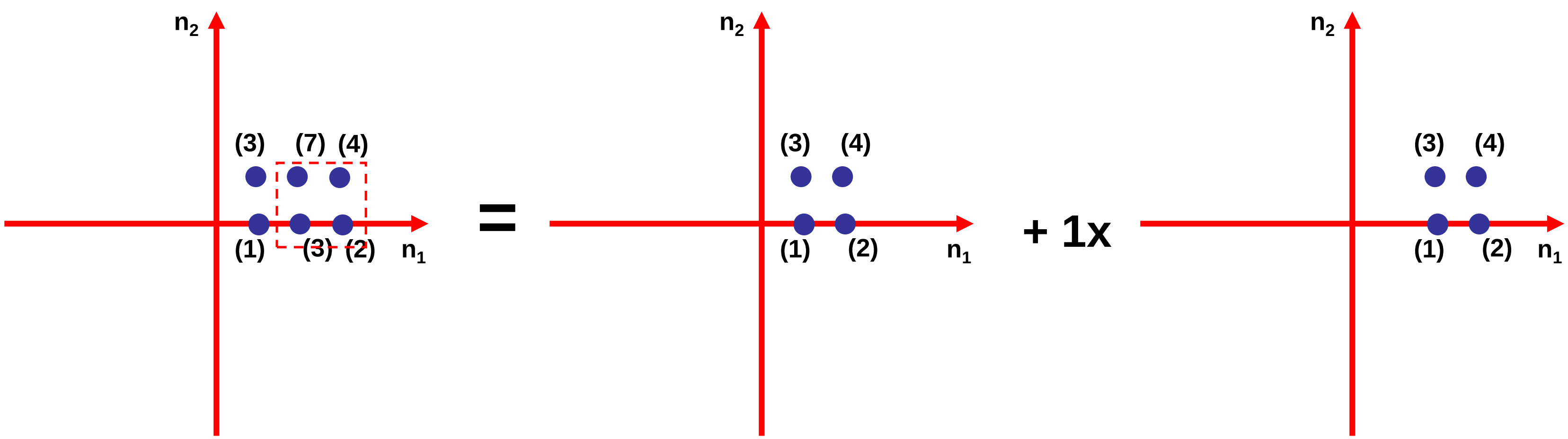
2D convolution

- example



$x[k_1, k_2]$ $\quad$ $h[n_1, n_2]$

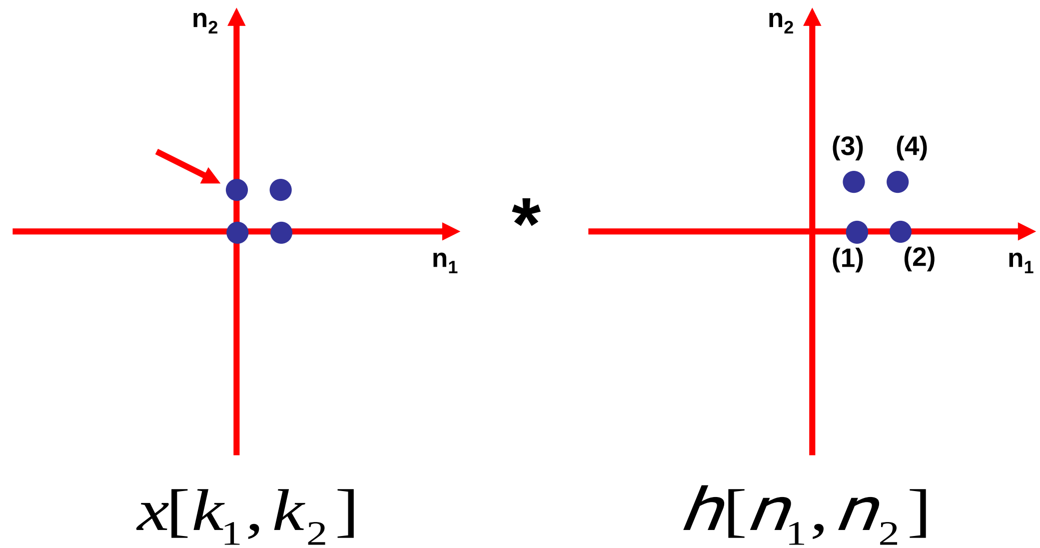
- $(k_1,k_2) = (1,0)$



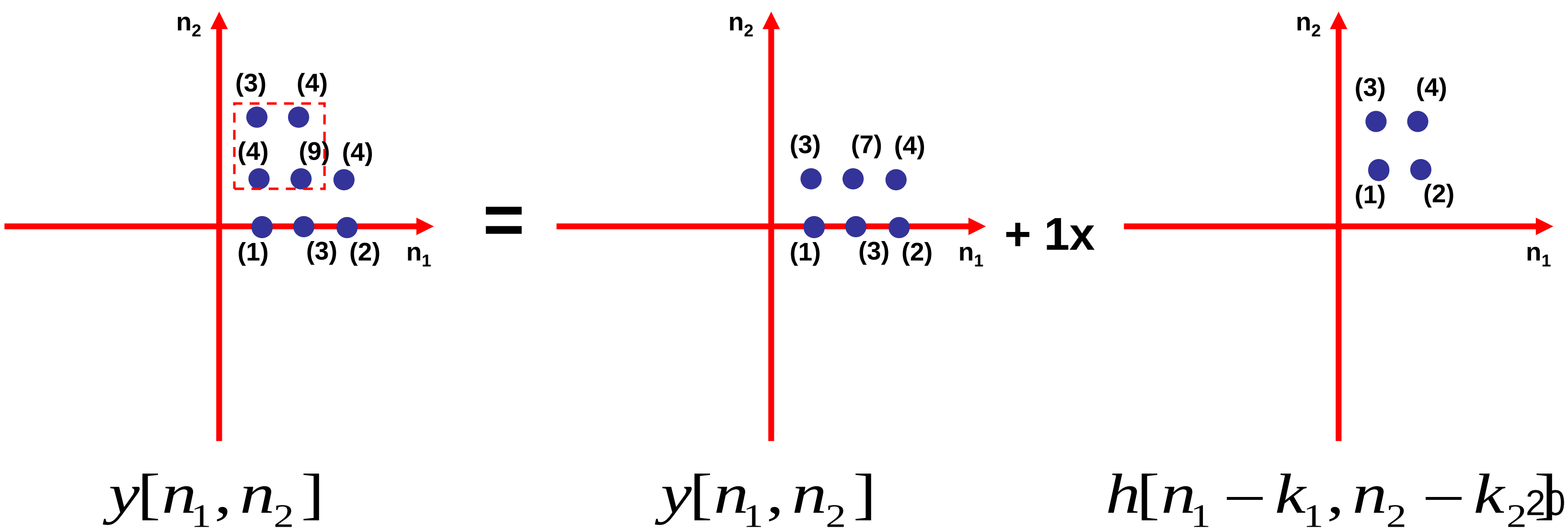
$y[n_1, n_2]$ $\quad$ $y[n_1, n_2]$ $\quad$ $h[n_1 - k_1, n_2 - k_2]$

2D convolution

- example

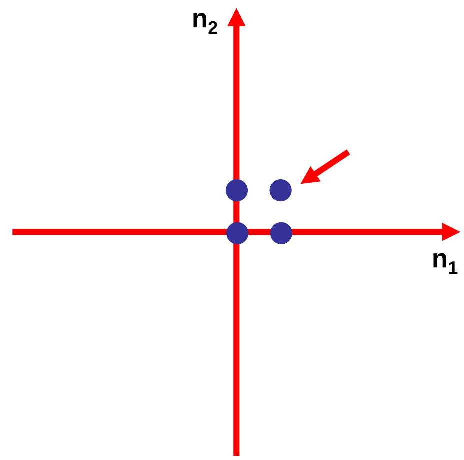


- $(k_1, k_2) = (0,1)$



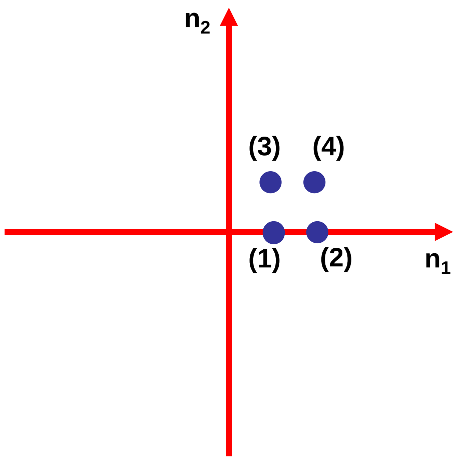
2D convolution

- example



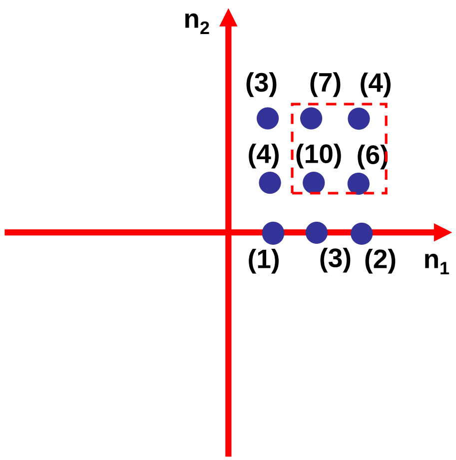
$x[k_1, k_2]$

*



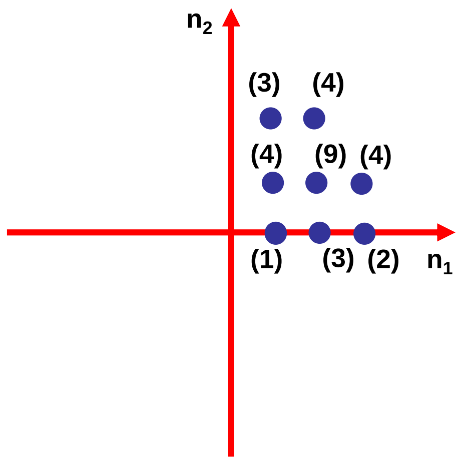
$h[n_1, n_2]$

- $(k_1,k_2) = (1,1)$



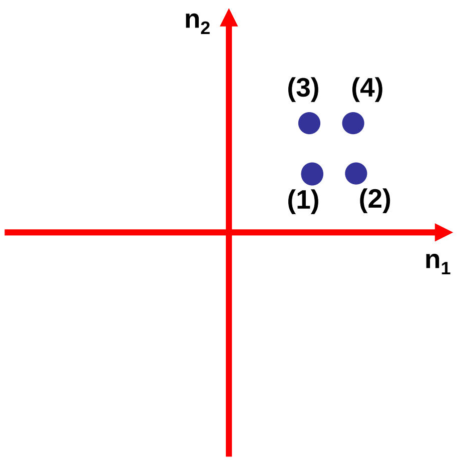
$y[n_1, n_2]$

=



$y[n_1, n_2]$

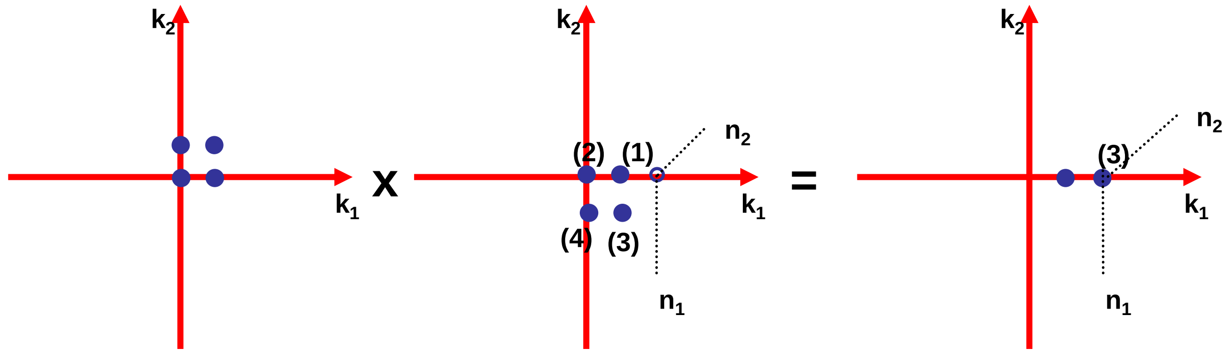
+ 1x



$h[n_1 - k_1, n_2 - k_2]$

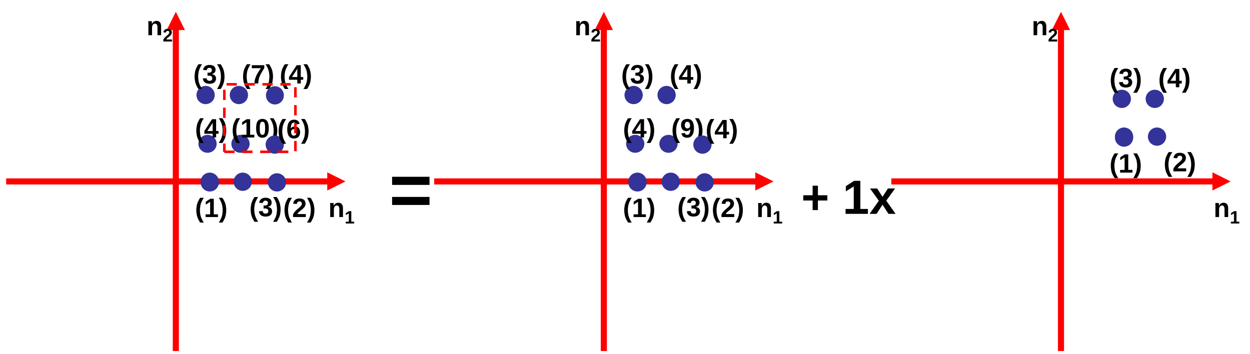
2D convolution

- note the differences with the previous convolution
  - before we were computing one $y[n_1,n_2]$ at a time



$x[k_1,k_2]$ $h[n_1-k_1,n_2-k_2]$ $y[n_1,n_2]$

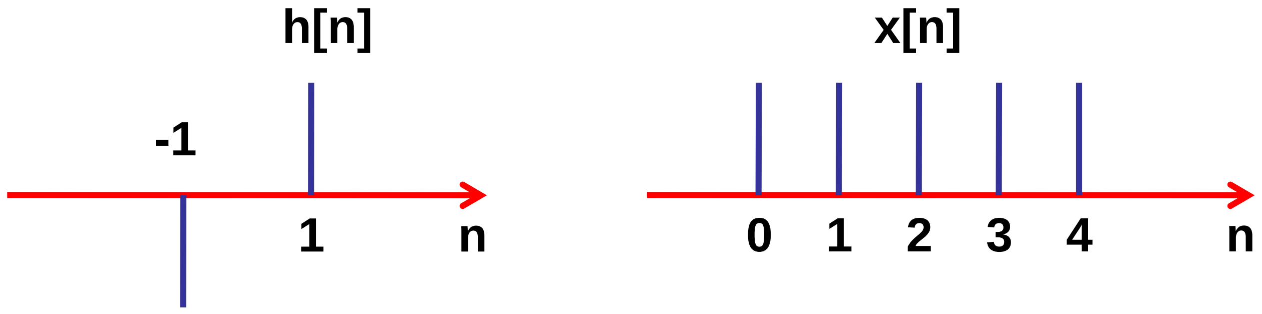
  - now we update the entire sequence at a time



$y[n_1,n_2]$ $y[n_1,n_2]$ $h[n_1-k_1,n_2-k_2]$

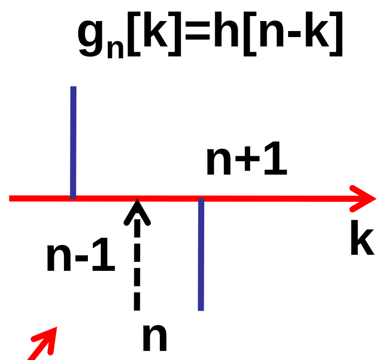
Convolution

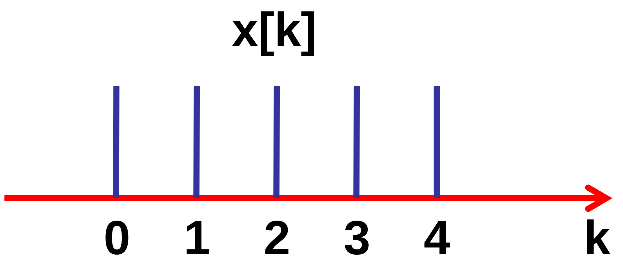
- When do I use the serial vs parallel method?
  - serial always works
  - parallel is useful when one of the sequences is small
  - example

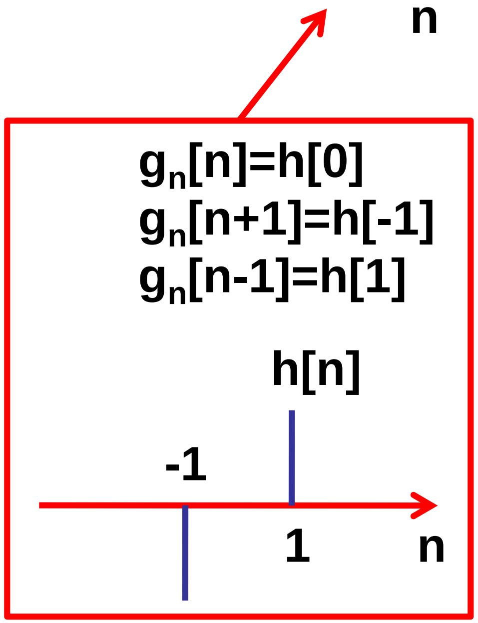


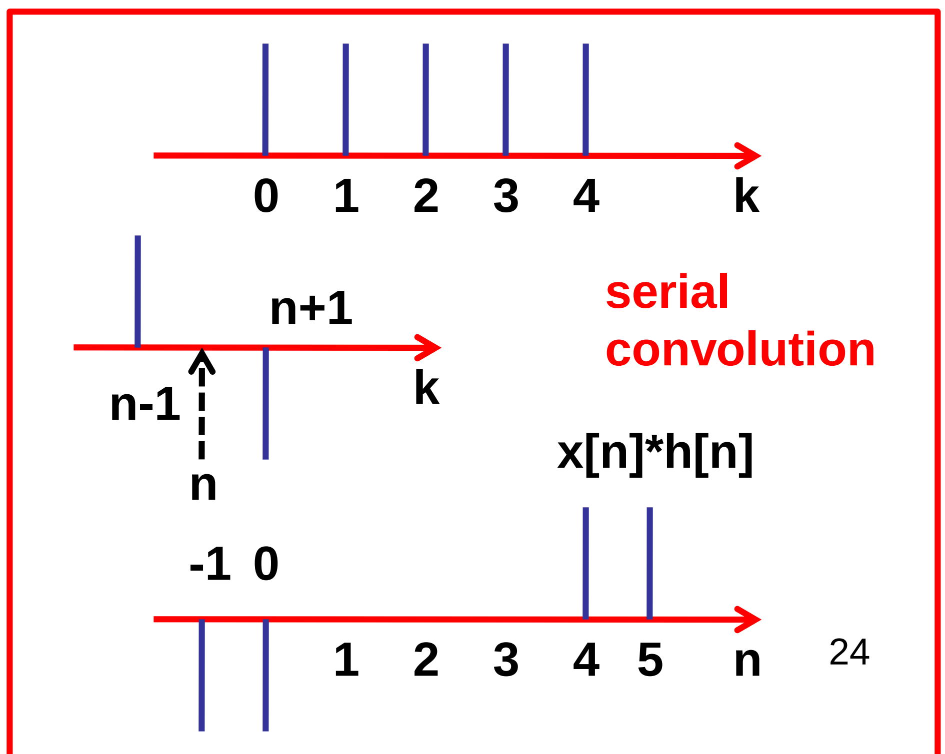
Convolution

- serial



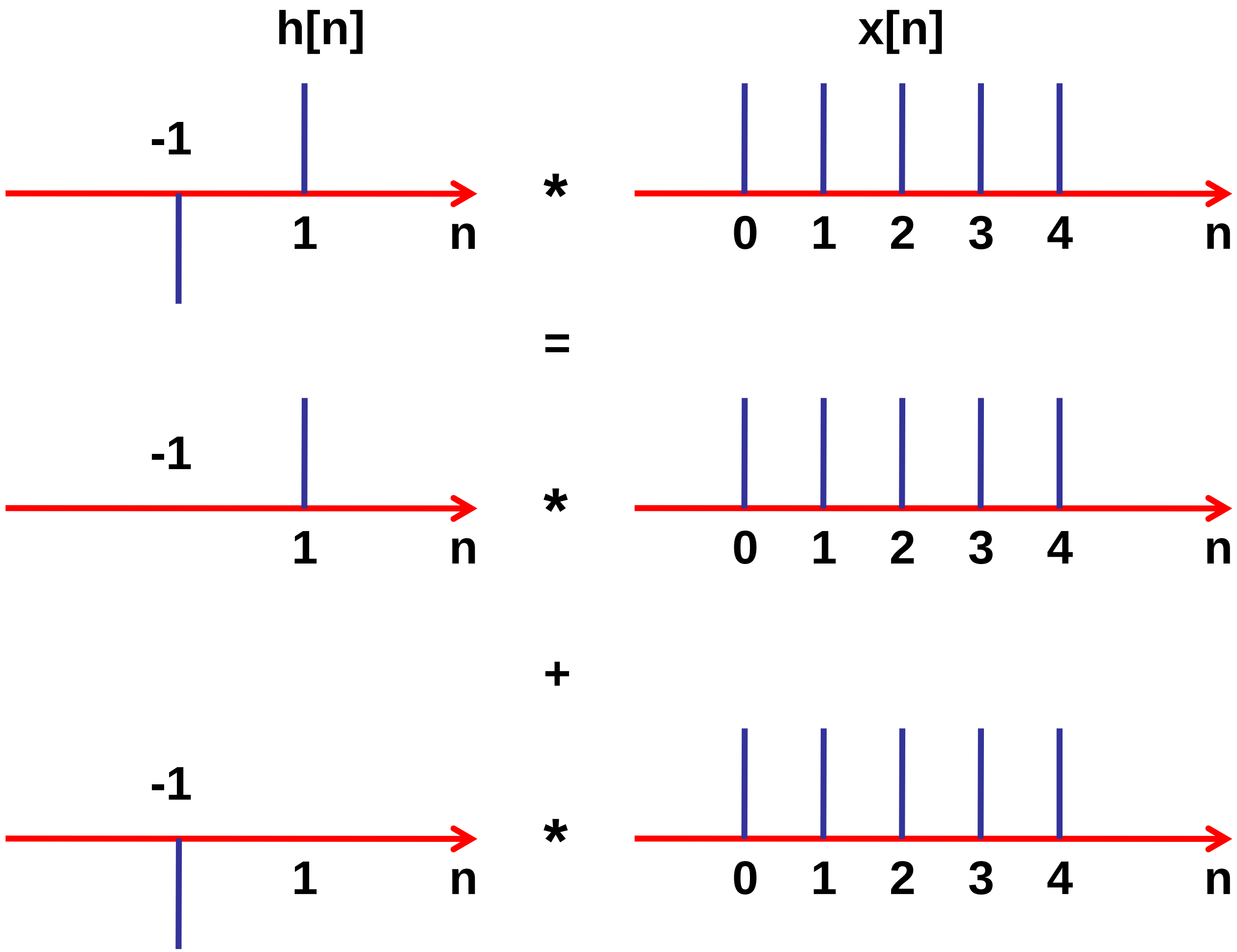






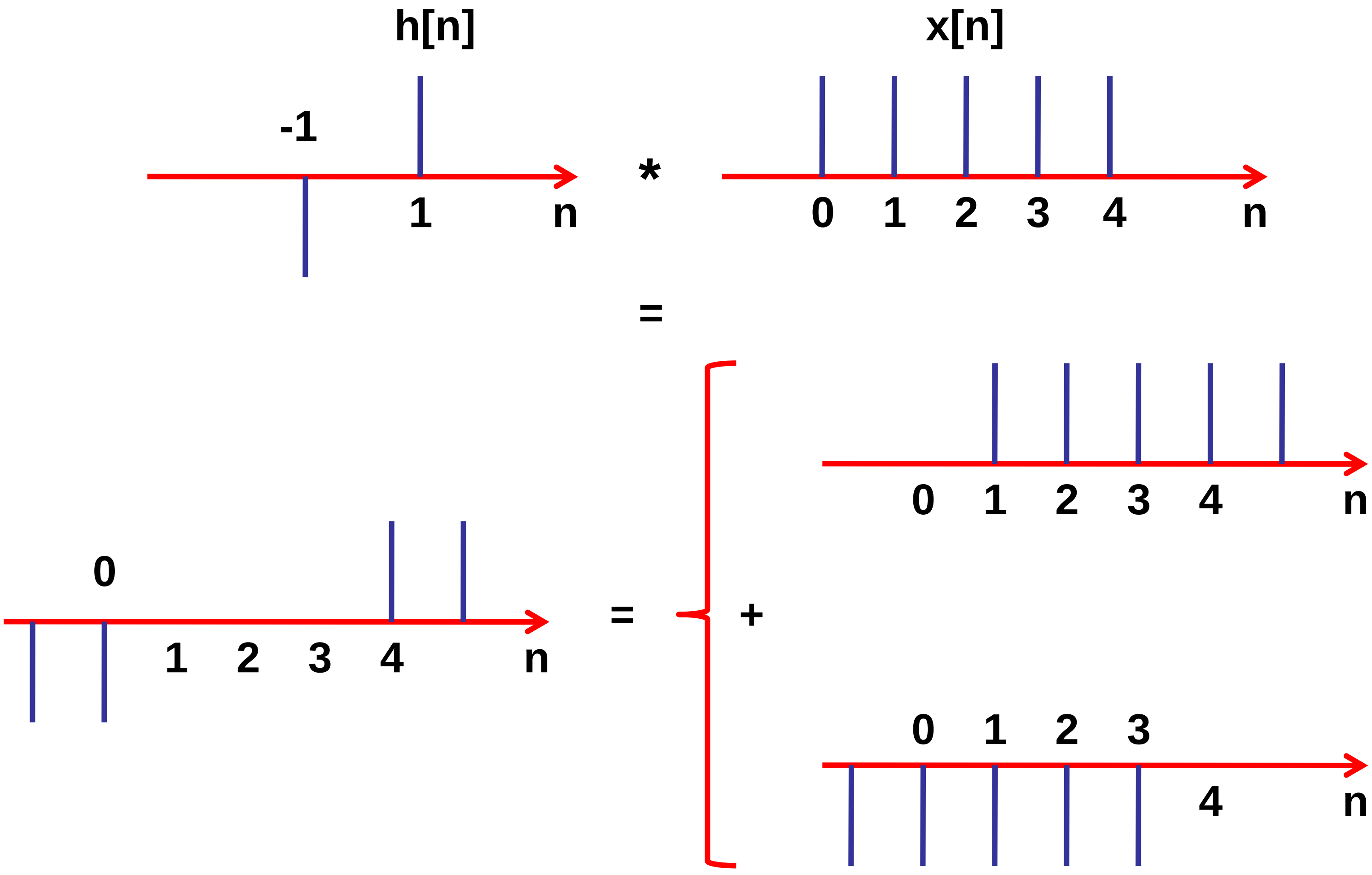
Convolution

- parallel



Convolution

- parallel



Separable systems

- **Definition:** a system is separable if and only if its impulse response is a separable sequence

$$h[n_1, n_2] = h_1[n_1] \times h_2[n_2]$$

- note that, in this case the convolution simplifies

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h[n_1 - k_1, n_2 - k_2]$$

$$= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h_1[n_1 - k_1] h_2[n_2 - k_2]$$

$$= \sum_{k_1=-\infty}^{\infty} h_1[n_1 - k_1] \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h_2[n_2 - k_2]$$

$$= \sum_{k_1=-\infty}^{\infty} h_1[n_1 - k_1] f[k_1, n_2]$$

Separable systems

- the convolution simplifies to

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}h_1[n_1-k_1]f[k_1,n_2]$$

with

$$f[k_1,n_2]=\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]h_2[n_2-k_2]$$

- note that:
  - for a fixed k_1, $f[k_1,n_2]$ is 1D convolution of $x[k_1,n_2]$ and $h_2[n_2]$

$$f[k_1,n_2]=x[k_1,n_2]*h_2[n_2]$$

  - for a fixed n_2, $y[n_1,n_2]$ is 1D convolution of $f[n_1,n_2]$ and $h_1[n_1]$

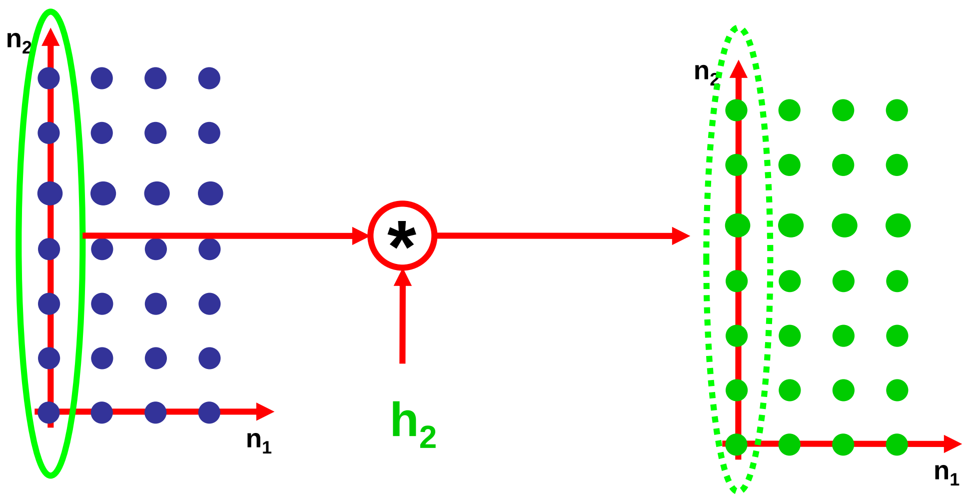
$$y[n_1,n_2]=f[n_1,n_2]*h_1[n_1]$$

Separable systems

- the convolution simplifies to a sequence of 1D steps
- **step1)** for every k_1,
  - $f[k_1,n_2]$ is 1D convolution of $x[k_1,n_2]$ and $h_2[n_2]$

$$f[k_1,n_2] = x[k_1,n_2] * h_2[n_2]$$

  - which means: "convolve the columns of x with h_2 to obtain columns of f"



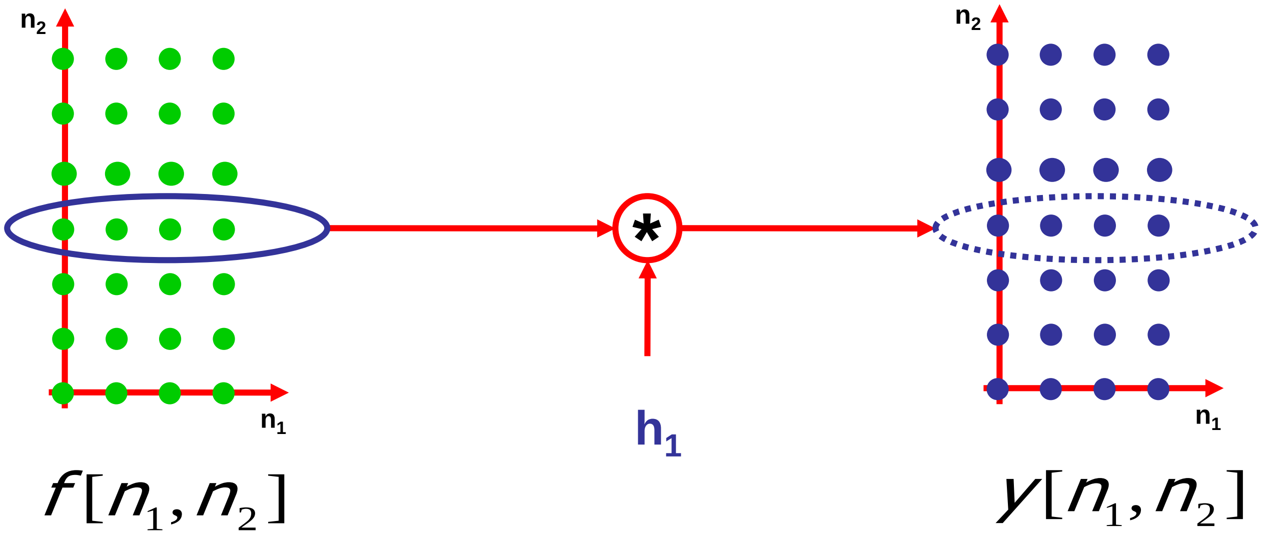
$x[n_1,n_2]$ $f[n_1,n_2]$

Separable systems

- **step2)** for every n_2,
  - $y[n_1,n_2]$ is 1D convolution of $f[n_1,n_2]$ and $h_1[n_1]$

$$y[n_1,n_2] = f[n_1,n_2] * h_1[n_1]$$

  - which means: "convolve the rows of *f* with h_1 to obtain rows of *y*"



Separable systems

- in summary, if we have a separable system

$$h[n_1, n_2] = h_1[n_1] \times h_2[n_2]$$

to convolve with $x[n_1,n_2]$ we:

  - 1) "convolve the columns of x with h_2 to create f"

$$f[k_1, n_2] = x[k_1, n_2] * h_2[n_2]$$

  - 2) "convolve the rows of f with h_1 to obtain y"

$$y[n_1, n_2] = f[n_1, n_2] * h_1[n_1]$$

The Discrete-Space Fourier Transform

- as in 1D, an important concept in linear system analysis is that of the Fourier transform
- the Discrete-Space Fourier Transform is the 2D extension of the Discrete-Time Fourier Transform

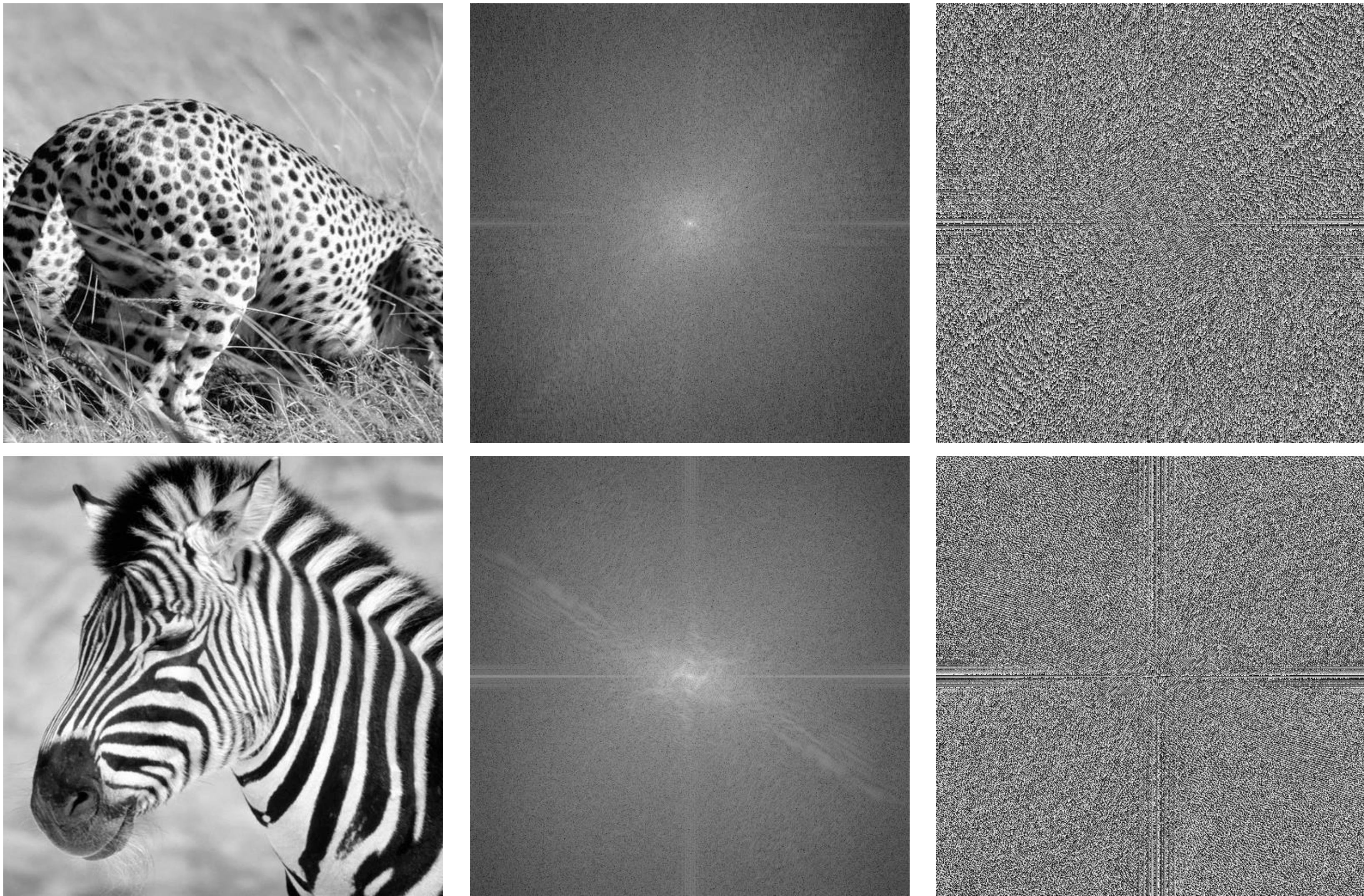
$$X(\omega_1,\omega_2)=\sum_{n_1}\sum_{n_2}x[n_1,n_2]e^{-j\omega_1 n_1}e^{-j\omega_2 n_2}$$

$$x[n_1,n_2]=\frac{1}{(2\pi)^2}\iint X(\omega_1,\omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}d\omega_1 d\omega_2$$

- note that this is a continuous function of frequency
- the nomenclature distinguishes it from the 2D Discrete Fourier transform (we will get back to this)
- what does the DSFT of an image look like?

Image spectrum

- two images, the magnitude, and phase of their FTs



Phase and Magnitude

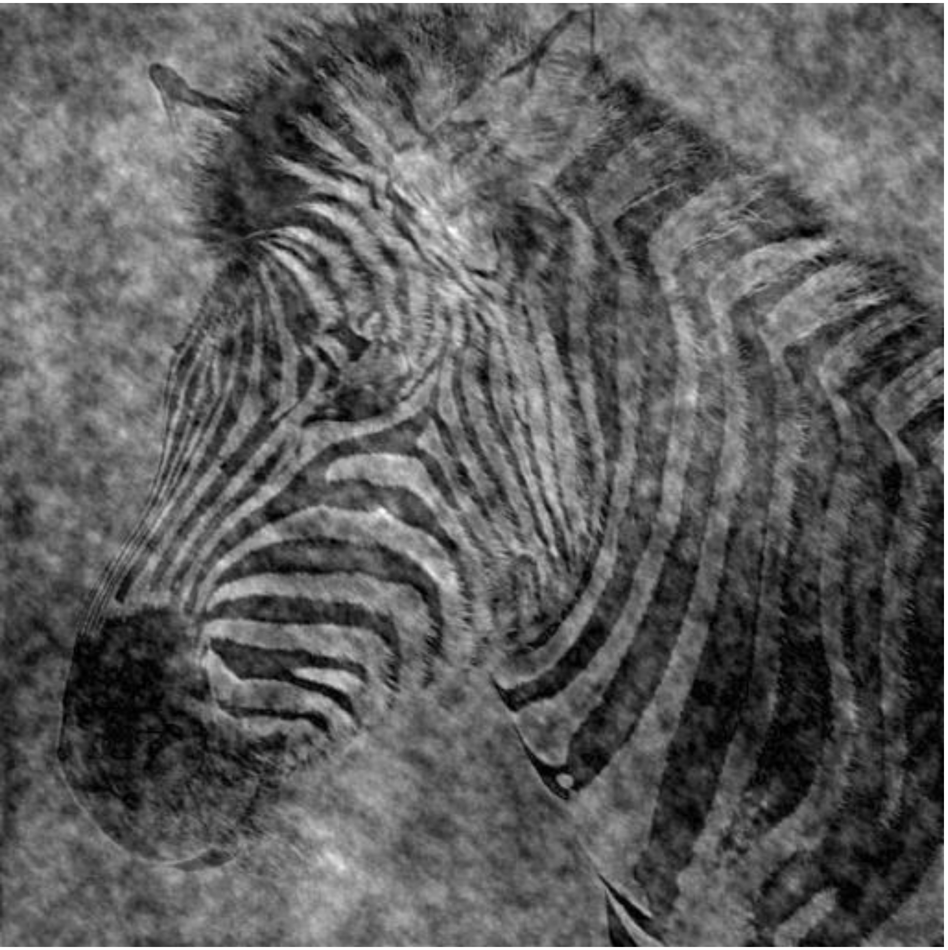
- curious fact
  - all natural images have about the same magnitude transform
  - monotonically decaying with frequency

$$X(\omega_1, \omega_2) \propto \frac{1}{\omega_1^2 + \omega_2^2}$$

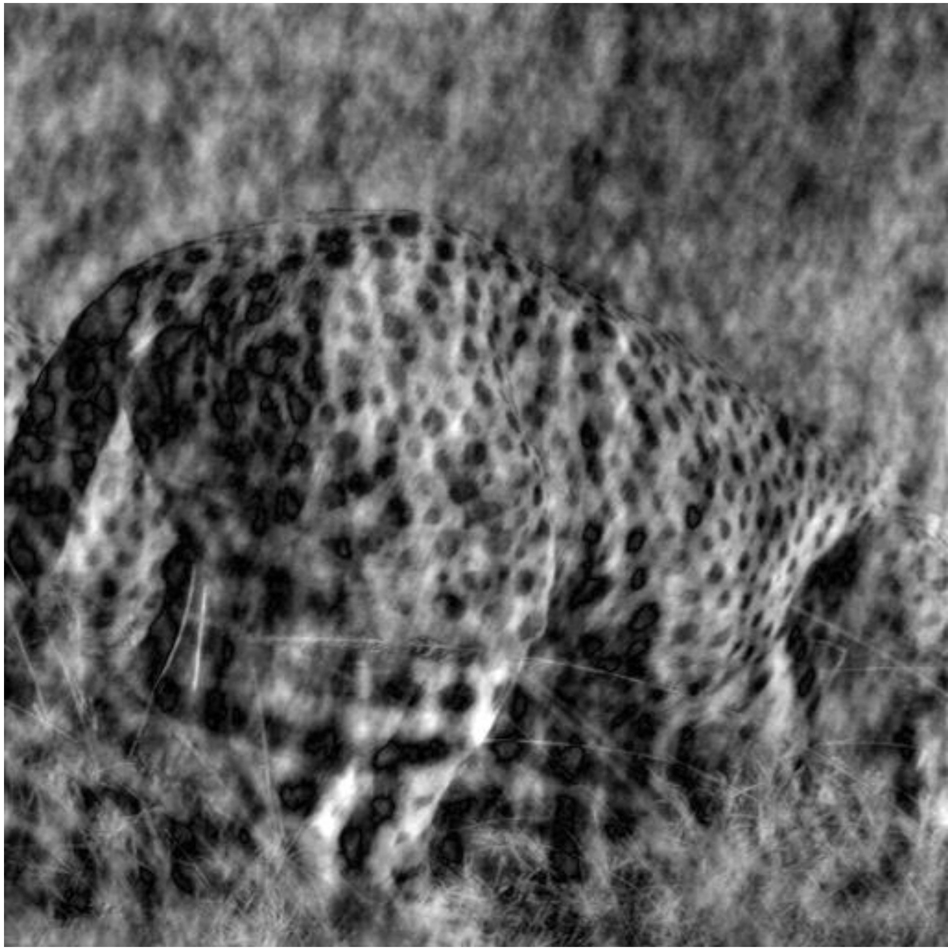
- hence, phase seems to matter, but magnitude largely doesn't
- we can see this through the following experiment
- take two pictures, swap the phase transforms, compute the inverse
- here is what you get

The importance of phase

Reconstruction with zebra phase, cheetah magnitude



Reconstruction with cheetah phase, zebra magnitude



LSI systems

- why do we care so much about Fourier transforms?
- note that when we convolve a sequence with a complex exponential,

$$x[n_1, n_2] = e^{j\varpi_1 n_1} e^{j\varpi_2 n_2}$$

we get

$$\begin{aligned} y[n_1, n_2] &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h[k_1, k_2]\, x[n_1 - k_1, n_2 - k_2] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h[k_1, k_2]\, \mathrm{e}^{\mathrm{j}\varpi_1 (n_1 - k_1)}\, \mathrm{e}^{\mathrm{j}\varpi_2 (n_2 - k_2)} \\ &= \mathrm{e}^{\mathrm{j}\varpi_1 n_1}\, \mathrm{e}^{\mathrm{j}\varpi_2 n_2} H(\varpi_1, \varpi_2) \\ &= x[n_1, n_2] H(\varpi_1, \varpi_2) \end{aligned}$$

LSI systems

- but we have seen that, for an LSI system
  - the output in response to $x[n_1,n_2]$
  - is the convolution with the impulse response $h[n_1,n_2]$
  - hence, the response to

$$x[n_1, n_2] = e^{j\varpi_1 n_1} e^{j\varpi_2 n_2}$$

  - is

$$y[n_1, n_2] = x[n_1, n_2] H(\varpi_1, \varpi_2)$$

  - this means that complex exponentials are the eigenfunctions of LSI systems
  - when we input an eigenfunction, we get back the same function
  - but scaled by $H(\omega_1, \omega_2)$
  - this is called the frequency response of the system

LSI systems

- this is remarkable, since
  - we know that any signal can be represented as a weighted sum of complex exponentials

$$X(\omega_1,\omega_2) = \sum_{n_1}\sum_{n_2} x[n_1,n_2]e^{-j\omega_1 n_1}e^{-j\omega_2 n_2}$$

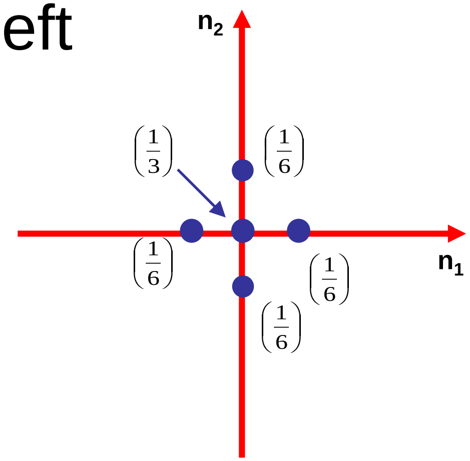
$$x[n_1,n_2] = \frac{1}{(2\pi)^2}\iint X(\omega_1,\omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}d\omega_1 d\omega_2$$

  - when the signal is fed to an LSI system, each exponential is scaled by $H(\omega_1,\omega_2)$
  - hence, the frequency response completely characterizes the system
  - and the DSFT of the output is just the product of the two

$$Y(\omega_1,\omega_2) = H(\omega_1,\omega_2)X(\omega_1,\omega_2)$$

Example

- the system with impulse response on the left
- has frequency response

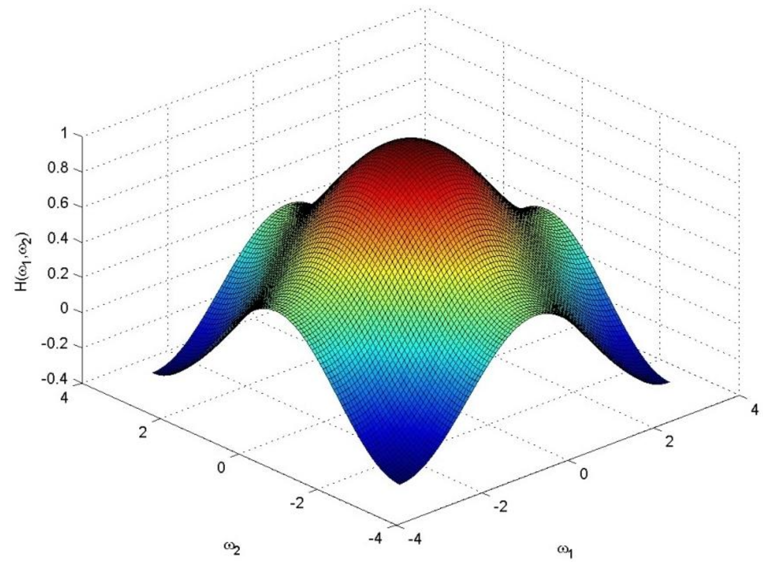


$$H(\omega_1,\omega_2)=\sum_{n_1}\sum_{n_2} h[n_1,n_2]e^{-j\omega_1 n_1}e^{-j\omega_2 n_2}$$

$$=\frac{1}{3}+\frac{1}{6}e^{-j\varpi_1}+\frac{1}{6}e^{-j\varpi_2}+\frac{1}{6}e^{j\varpi_1}+\frac{1}{6}e^{j\varpi_2}$$

$$=\frac{1}{3}+\frac{1}{3}\cos\varpi_1+\frac{1}{3}\cos\varpi_2$$

- note that:
  - the response is 1 at DC
  - lower for high frequencies
  - this system is a low-pass filter



WARNING

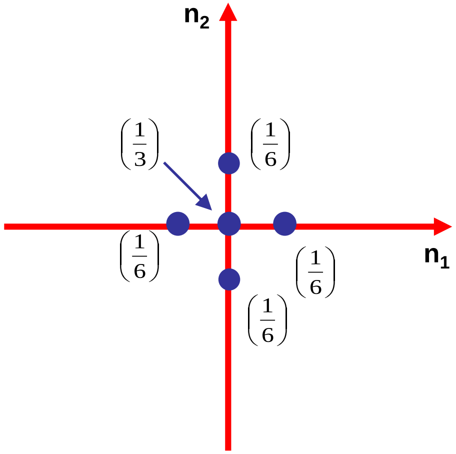
- WARNING, WARNING, WARNING!
- the equivalence

$$\frac{e^{-jn\varpi_1}+e^{jn\varpi_1}}{2}=\cos n\varpi_1$$

- is the oldest trick in the DSP book!
- please do not fall for it
  - you can “read” the sequence that has this DSFT

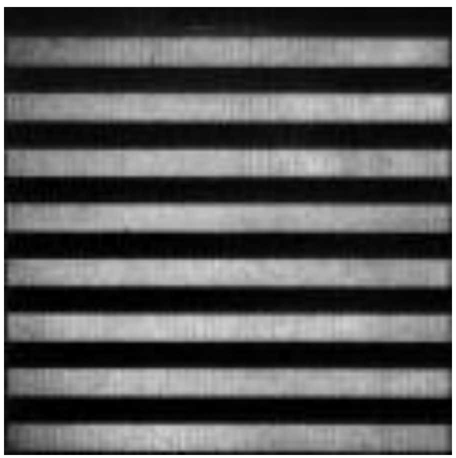
$$H(\omega_1,\omega_2)=\frac{1}{3}+\frac{1}{3}\cos\varpi_1+\frac{1}{3}\cos\varpi_2$$

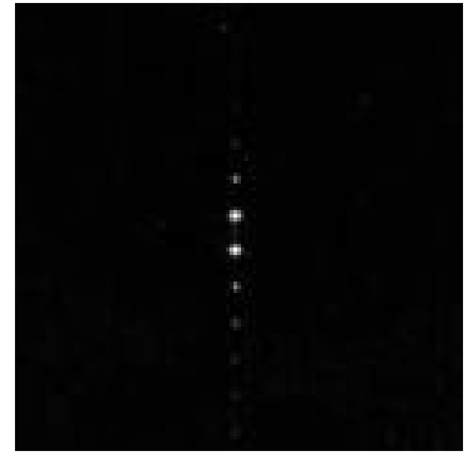
by applying this trick and the definition of DSFT!



WARNING

- Quizz: which on the left is the DSFT of this image?

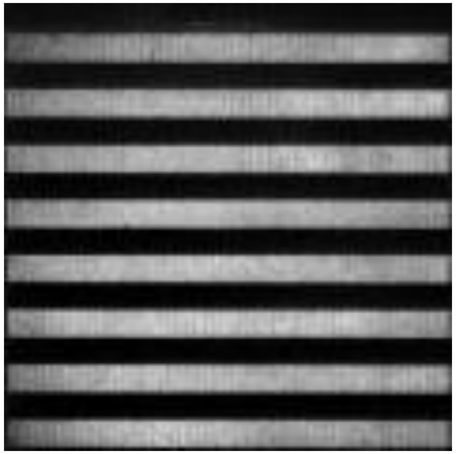




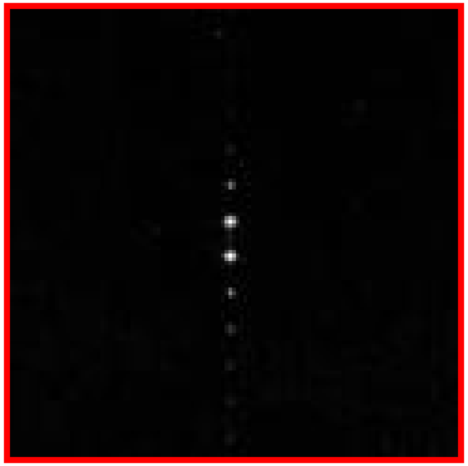
?

WARNING

- the way to think about this is the following:



  - this image has low frequencies horizontally (ω_1)
  - high frequencies vertically (ω_2)
  - the spectrum is a delta function along (ω_1) and harmonics along (ω_2)
  - the spectrum is this
  - wrong way: “because image is horizontal spectrum must be too”



Properties of the DSFT

- these are extremely important, but straightforward extension of what you have seen in 1D
- only novelty is separability (homework):
  - the DSFT of a separable sequence is itself separable
  - it is the product of the DTFTs of the 1D sequences that make up the 2D sequence

$$x[n_1, n_2] = x_1[n_1]x_2[n_2] \leftrightarrow X(\omega_1, \omega_2) = X_1(\omega_1)X_2(\omega_2)$$

- all other properties carry from 1D to 2D

Properties of the DSFT

$$x(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2)$$
$$y(n_1, n_2) \longleftrightarrow Y(\omega_1, \omega_2)$$

Property 1. <u>*Linearity*</u>

$$ax(n_1, n_2) + by(n_1, n_2) \longleftrightarrow aX(\omega_1, \omega_2) + bY(\omega_1, \omega_2)$$

Property 2. <u>*Convolution*</u>

$$x(n_1, n_2) * y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2)Y(\omega_1, \omega_2)$$

Property 3. <u>*Multiplication*</u>

$$x(n_1, n_2)y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2) \circledast Y(\omega_1, \omega_2)$$

$$= \frac{1}{(2\pi)^2} \int_{\theta_1=-\pi}^{\pi} \int_{\theta_2=-\pi}^{\pi} X(\theta_1, \theta_2)Y(\omega_1 - \theta_1, \omega_2 - \theta_2)\, d\theta_1\, d\theta_2$$

Property 4. <u>*Separable Sequence*</u>

$$x(n_1, n_2) = x_1(n_1)x_2(n_2) \longleftrightarrow X(\omega_1, \omega_2) = X_1(\omega_1)X_2(\omega_2)$$

Property 5. <u>*Shift of a Sequence and a Fourier Transform*</u>

(a) $x(n_1 - m_1, n_2 - m_2) \longleftrightarrow X(\omega_1, \omega_2)e^{-j\omega_1 m_1}e^{-j\omega_2 m_2}$

(b) $e^{j\nu_1 n_1}e^{j\nu_2 n_2}x(n_1, n_2) \longleftrightarrow X(\omega_1 - \nu_1, \omega_2 - \nu_2)$

Property 6. <u>*Differentiation*</u>

(a) $-jn_1x(n_1, n_2) \longleftrightarrow \dfrac{\partial X(\omega_1, \omega_2)}{\partial \omega_1}$

(b) $-jn_2x(n_1, n_2) \longleftrightarrow \dfrac{\partial X(\omega_1, \omega_2)}{\partial \omega_2}$

Properties of the DSFT

Property 7. *Initial Value and DC Value Theorem*

(a) $$x(0, 0) = \frac{1}{(2\pi)^2} \int_{\omega 1 = -\pi}^{\pi} \int_{\omega 2 = -\pi}^{\pi} X(\omega_1, \omega_2)\, d\omega_1\, d\omega_2$$

(b) $$X(0, 0) = \sum_{n1 = -\infty}^{\infty} \sum_{n2 = -\infty}^{\infty} x(n_1, n_2)$$

Property 8. *Parseval's Theorem*

(a) $$\sum_{n1 = -\infty}^{\infty} \sum_{n2 = -\infty}^{\infty} x(n_1, n_2) y^*(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{\omega 1 = -\pi}^{\pi} \int_{\omega 2 = -\pi}^{\pi} X(\omega_1, \omega_2) Y^*(\omega_1, \omega_2)\, d\omega_1\, d\omega_2$$

(b) $$\sum_{n1 = -\infty}^{\infty} \sum_{n2 = -\infty}^{\infty} |x(n_1, n_2)|^2 = \frac{1}{(2\pi)^2} \int_{\omega 1 = -\pi}^{\pi} \int_{\omega 2 = -\pi}^{\pi} |X(\omega_1, \omega_2)|^2\, d\omega_1\, d\omega_2$$

Properties of the DSFT

Property 9. <u>*Symmetry Properties*</u>

(a) $x(-n_1, n_2) \longleftrightarrow X(-\omega_1, \omega_2)$

(b) $x(n_1, -n_2) \longleftrightarrow X(\omega_1, -\omega_2)$

(c) $x(-n_1, -n_2) \longleftrightarrow X(-\omega_1, -\omega_2)$

(d) $x^*(n_1, n_2) \longleftrightarrow X^*(-\omega_1, -\omega_2)$

(e) $x(n_1, n_2)$: real $\longleftrightarrow X(\omega_1, \omega_2) = X^*(-\omega_1, -\omega_2)$

$X_R(\omega_1, \omega_2)$, $|X(\omega_1, \omega_2)|$: even (symmetric with respect to the origin)

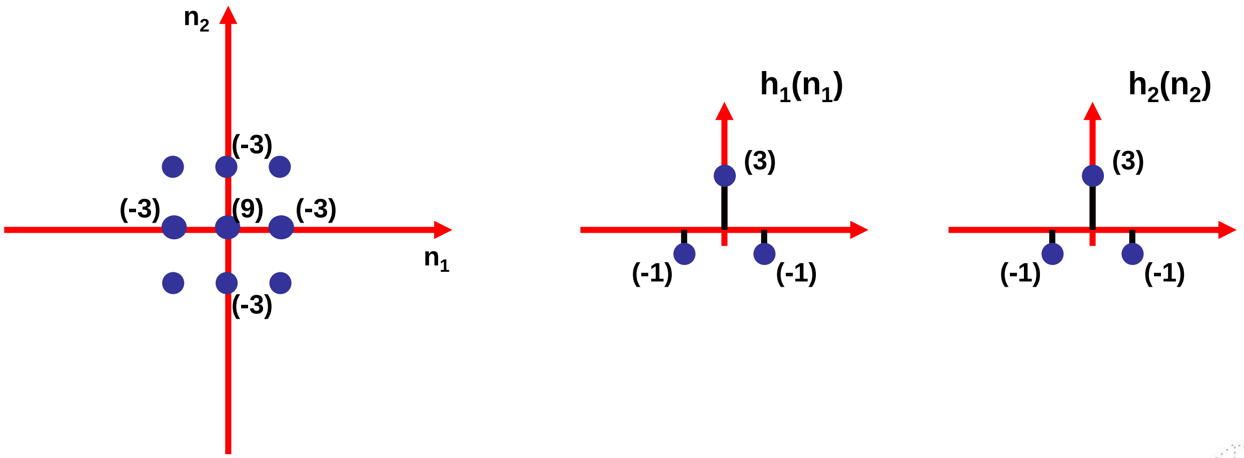
$X_I(\omega_1, \omega_2)$, $\theta_x(\omega_1, \omega_2)$: odd (antisymmetric with respect to the origin)

(f) $x(n_1, n_2)$: real and even $\longleftrightarrow X(\omega_1, \omega_2)$: real and even

(g) $x(n_1, n_2)$: real and odd $\longleftrightarrow X(\omega_1, \omega_2)$: pure imaginary and odd

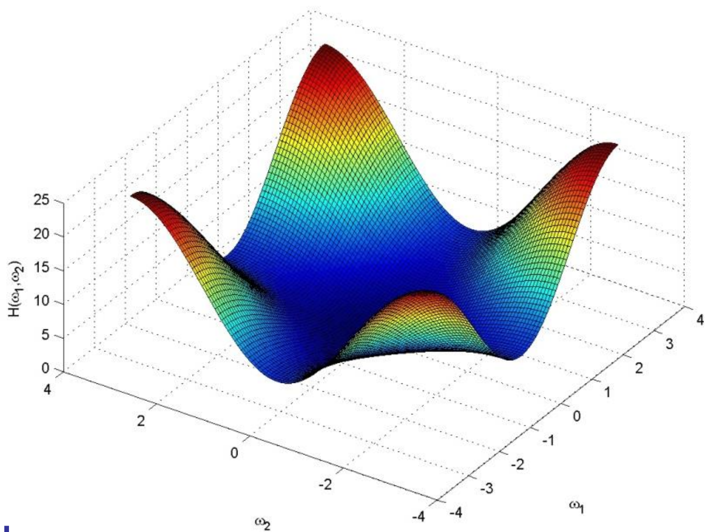
Example

- consider the separable impulse response



- frequency response

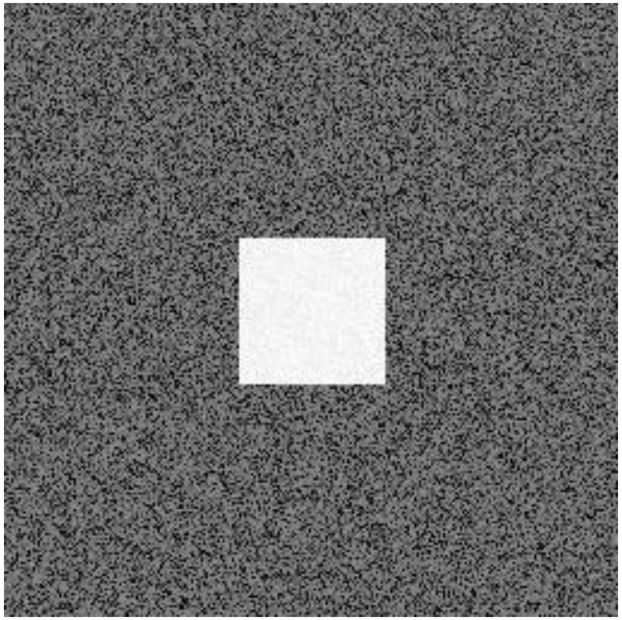
$$H(\omega_1,\omega_2) = H_1(\omega_1)H_2(\omega_2)$$
$$= (3-2\cos\varpi_1)(3-2\cos\varpi_2)$$

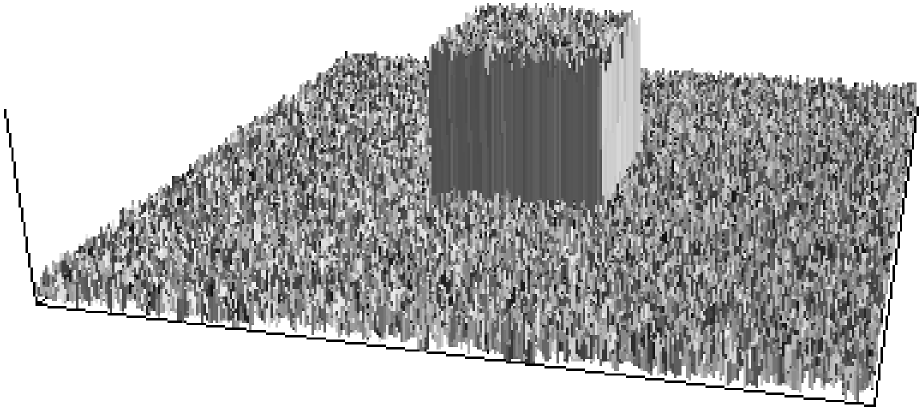


- note that:
  - this system is a high-pass filter
  - "diagonal" frequencies are enhanced

Examples

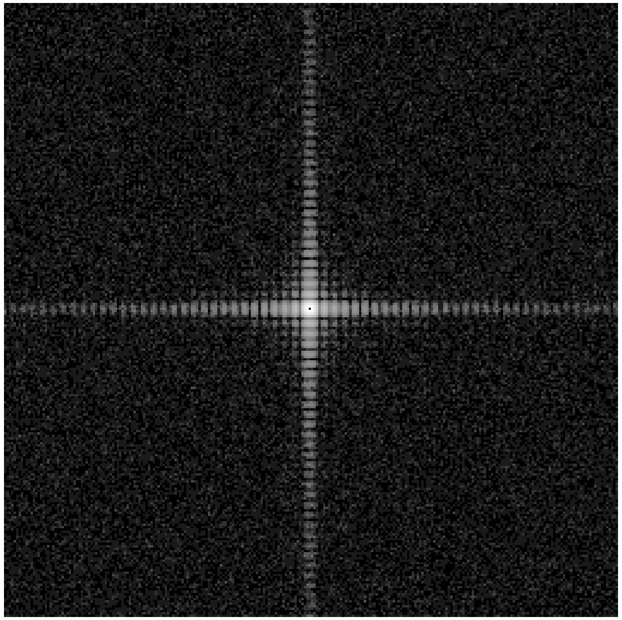
- what do filtered images look like?
  - here is a noisy image
  - a light square against dark background, plus noise

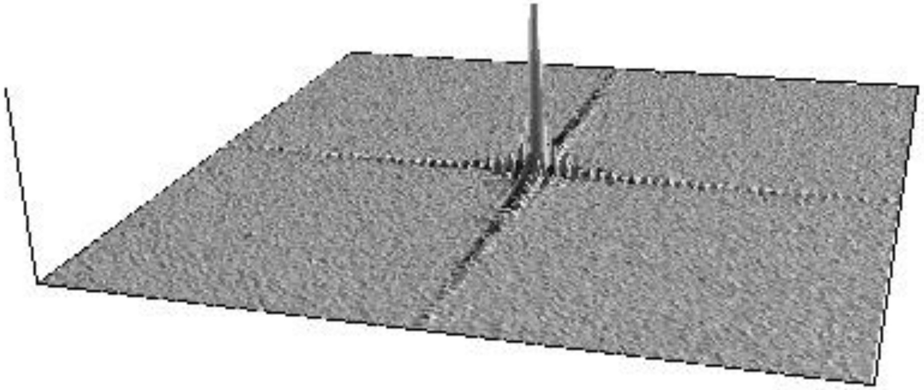




Examples

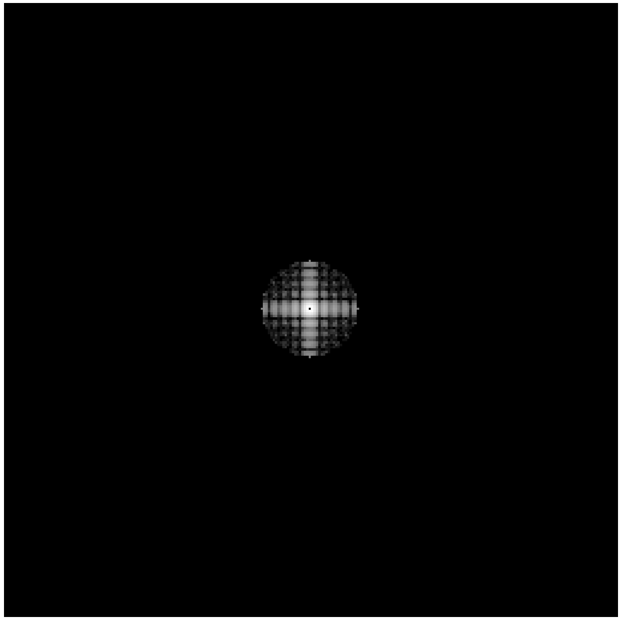
- what do filtered images look like?
  - here is the magnitude of its DSFT (origin at center), it contains:
  - a peak at the center,
  - some background signal at all frequencies,
  - a cross-like pattern that goes from low to high frequencies
  - why does it look like this?

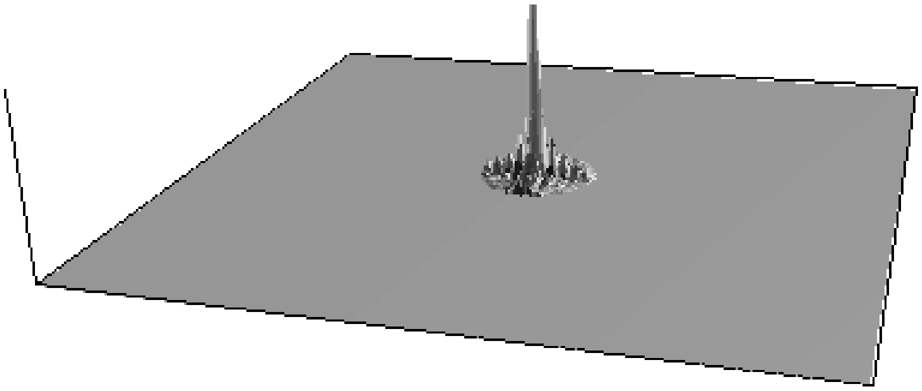




Examples

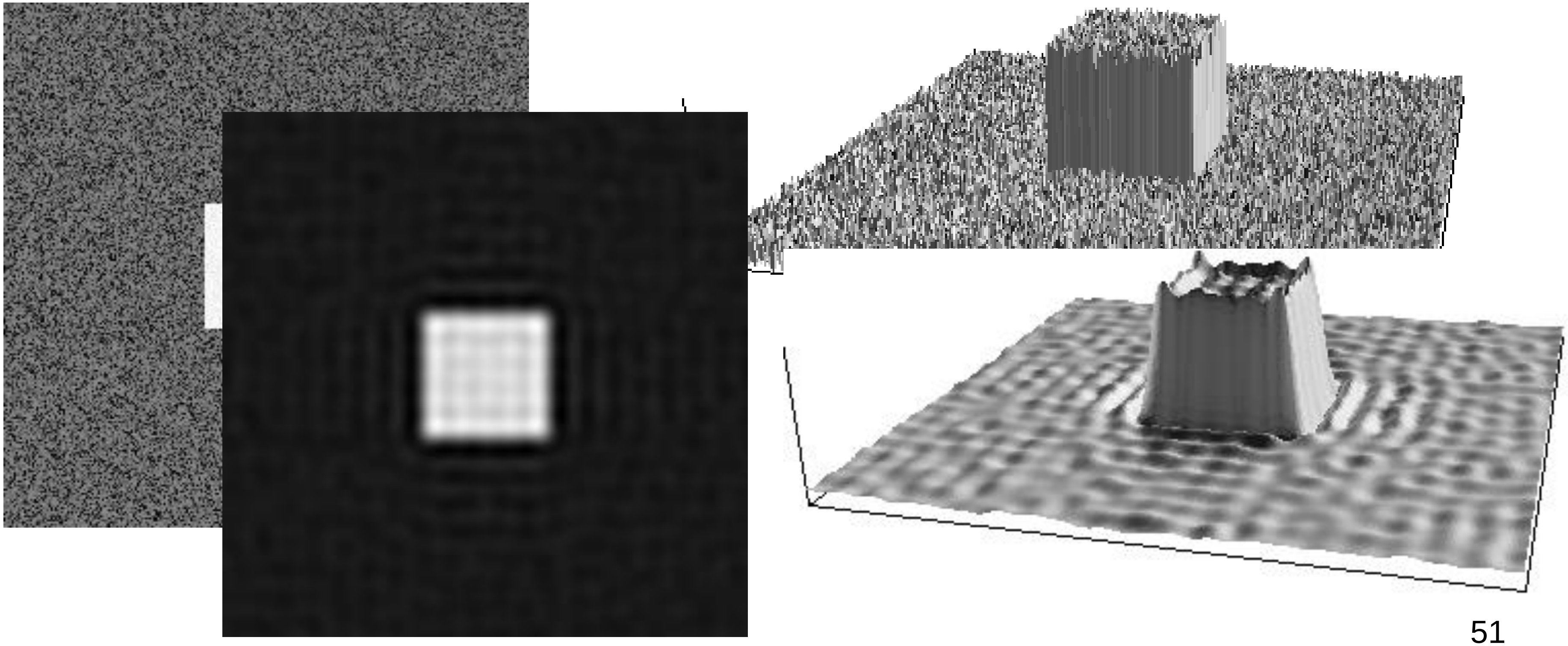
- one way to find out is to filter and reconstruct the image
  - we simulate the ideal low-pass filter by
  - removing all signal components outside a circle in the frequency domain
  - this is what the spectrum looks like
  - this gets rid of the background signal that covers all frequencies





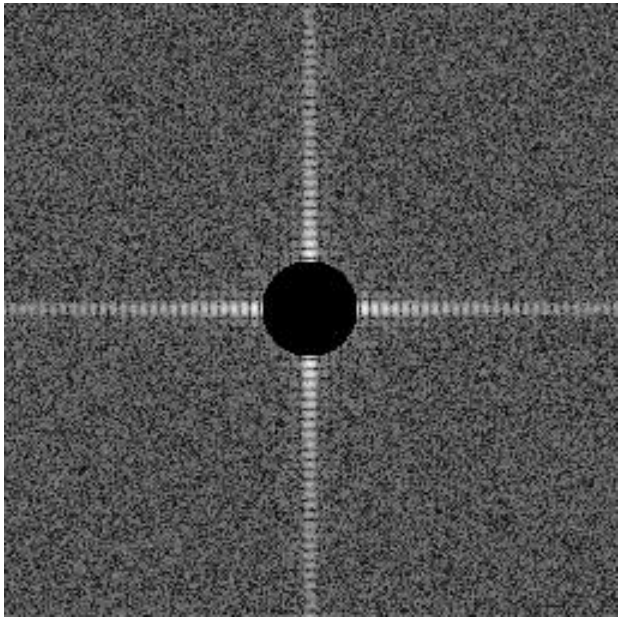
Examples

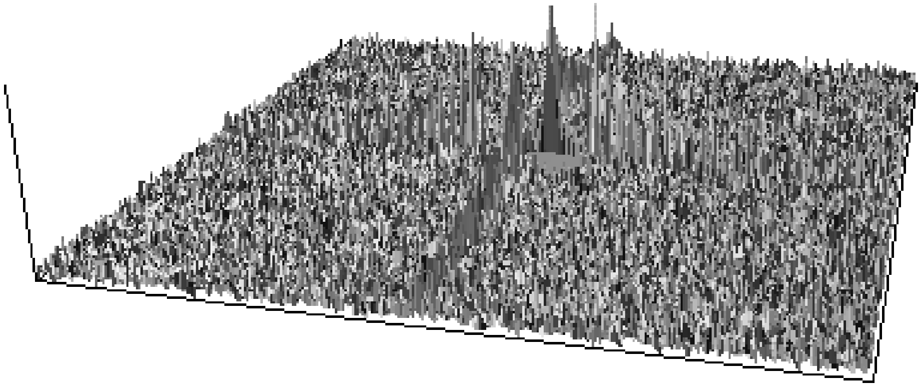
- this is the resulting image
  - the component we removed was due to the noise
  - "white" noise has energy at all frequencies
  - notice that there are some artifacts (i.e. ringing) in the reconstructed image



Examples

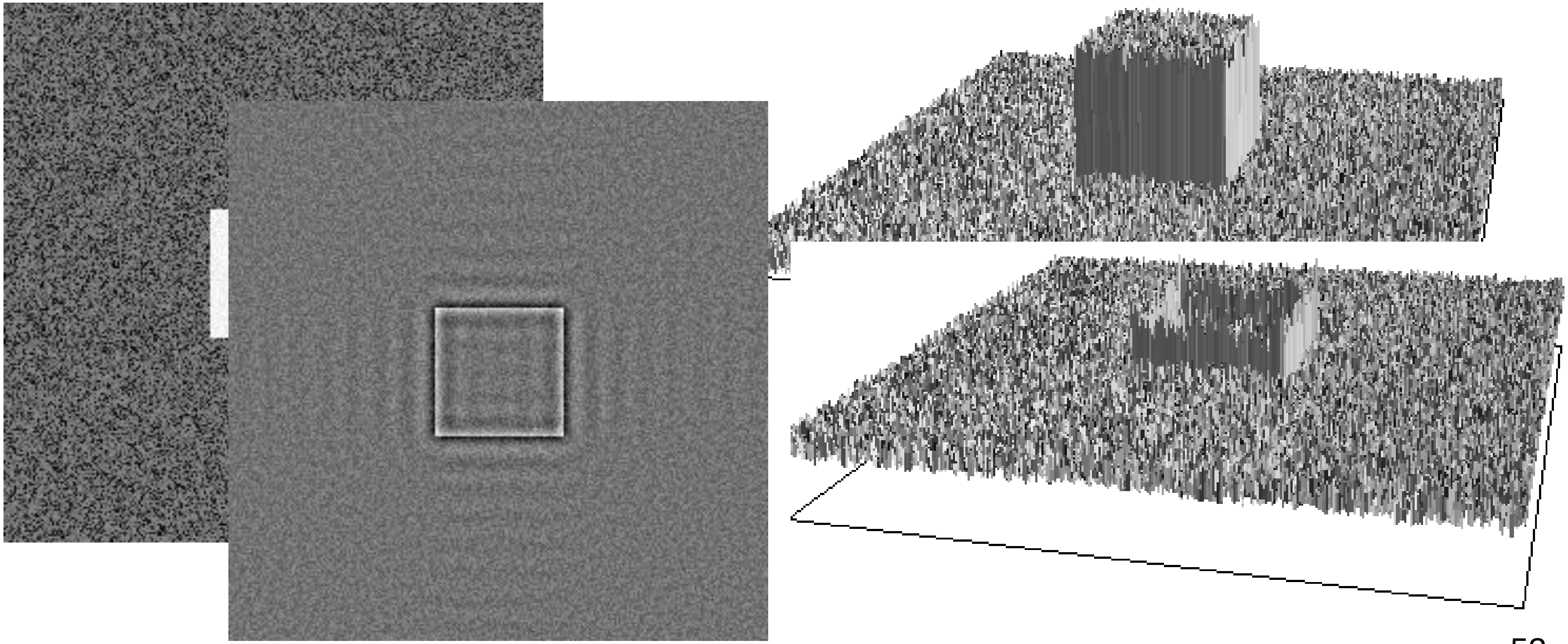
- what about the stuff other than noise?
  - let's high-pass by removing everything inside the circle





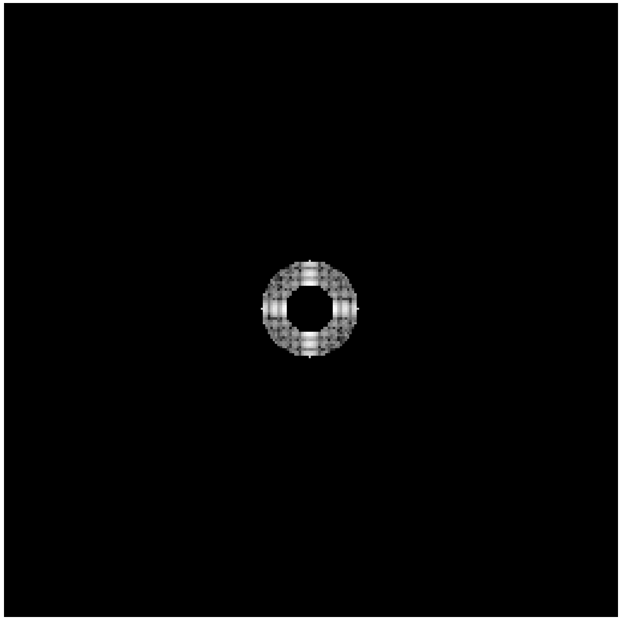
Examples

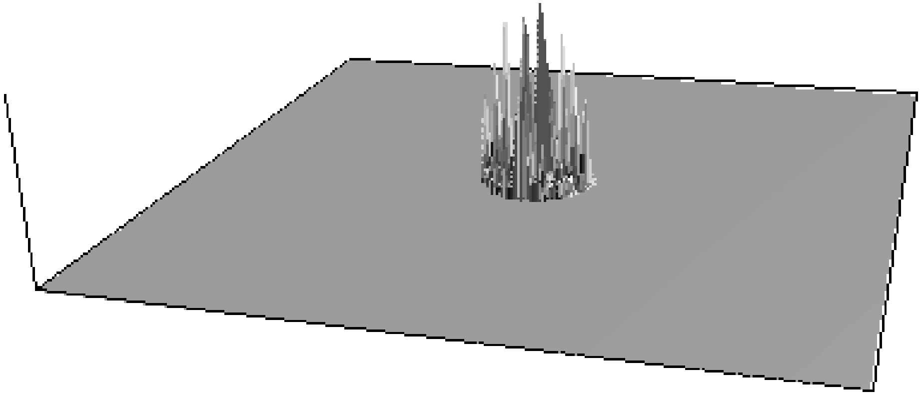
- this is the resulting image
  - we now get mostly noise, as expected
  - note that the square has mostly gone away
  - this means that the flat part is low-frequency
  - but we can still see the edges



Examples

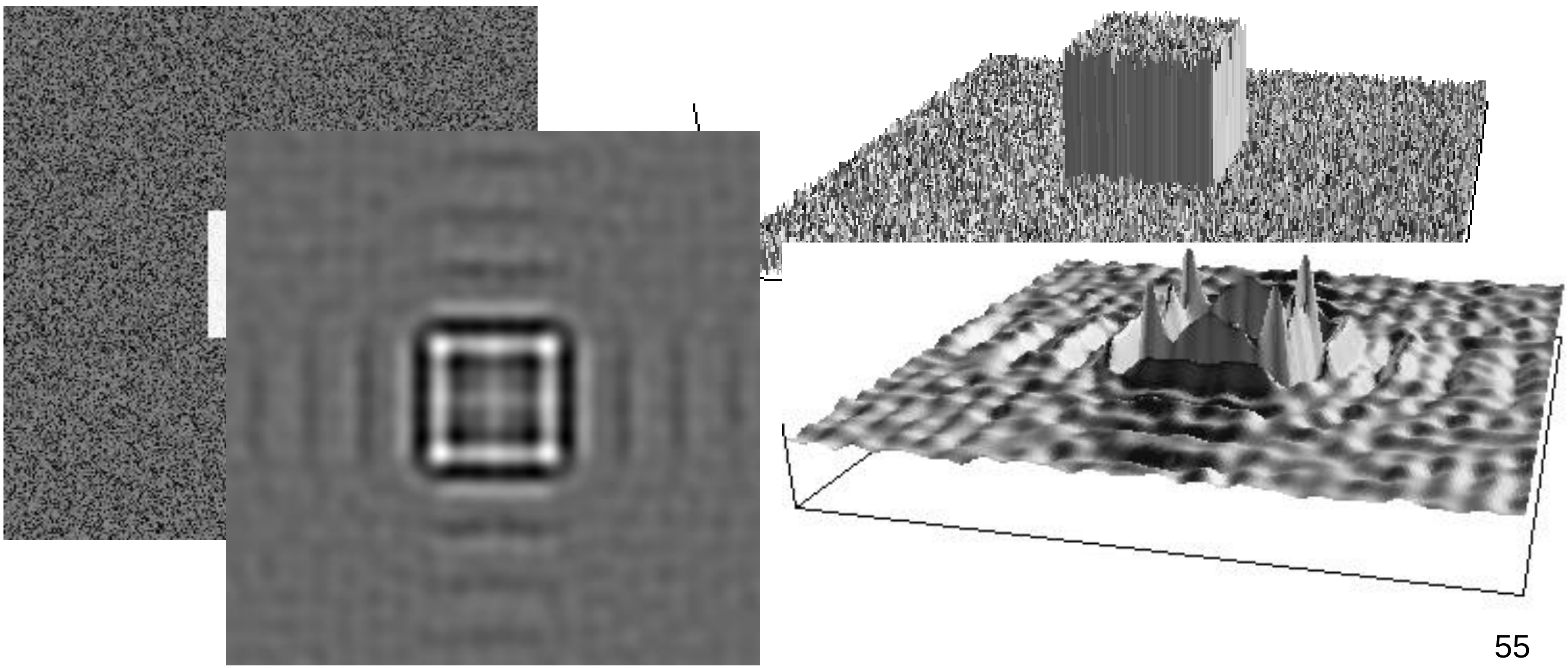
- this is interesting
  - the edges are not only low-pass
  - maybe they are the reason for the cross-shaped pattern
  - to check we band-pass





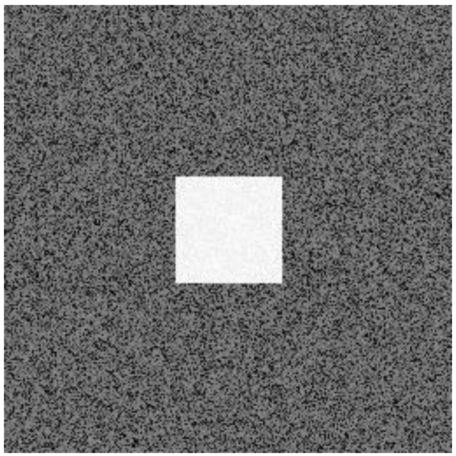
Examples

- this is the resulting image
  - we now get mostly the edges
  - we were right, the edges cause the cross-shaped pattern
  - note that the edges are very hard to filter out

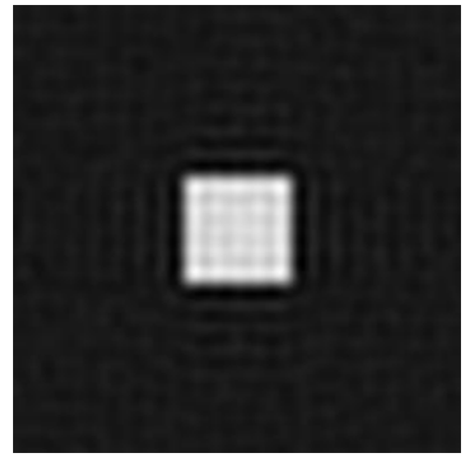


Examples

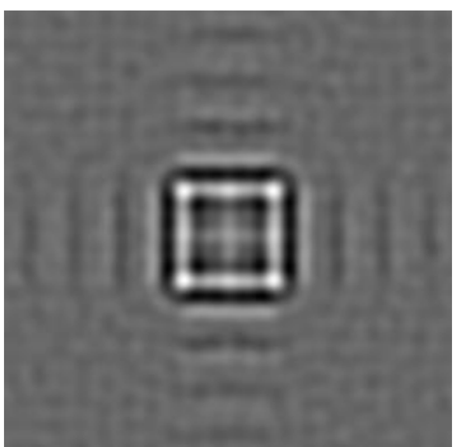
- this is one of the fundamental properties of images:
  - edges have energy at all frequencies



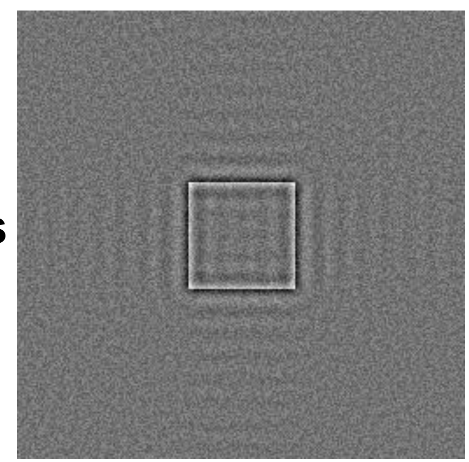
original



low-pass



band-pass



high-pass

The z transform

- once again, it is a straightforward extension of 1D
- **Definition:** the z-transform of the sequence $x[n_1,n_2]$ is

$$X(z_1,z_2) = \sum_{n_1}\sum_{n_2} x[n_1,n_2] z_1^{-n_1} z_2^{-n_2}$$

- the region of the (z_1,z_2) plane where this sum is finite is called the Region of Convergence (ROC)
- it turns out that:
  - in 2D the ROC is much more complicated than in 1D
  - while in 1D the ROC is bounded by poles (0D subspace of the 2D complex plane)
  - in 2D is bounded by pole surfaces (2D subspaces of the 4D space of two complex variables)

The z-transform

- computation is also much harder:
  - as you might remember from 1D
  - most useful tool in computing z-transforms is polynomial factorization
  - z-transform is a ratio of two polynomials

$$Y(z) = \frac{N(z)}{D(z)}$$

  - we factor in to a sum of low order terms, e.g.

$$Y(z) = \sum_i \frac{1}{1 - a_i z^{-1}}$$

  - and then invert each of the terms to get y[n]

z-transform

- in 2D we only have one of two situations
- 1) the sequence is separable, in which case everything reduces to the 1D case

$$x[n_1, n_2] = x_1[n_1]x_2[n_2] \leftrightarrow X(z_1, z_2) = X_1(z_1)X_2(z_2)$$

$$ROC: |z_1| \in ROC \text{ of } X_1(z_1) \text{ and}$$

$$|z_2| \in ROC \text{ of } X_2(z_2)$$

  the proof is identical to that of the DSFT

- 2) the signal is not separable
  – here our polynomials are of the form $z_1^m z_2^n$ and, in general, it is not know how to factor them
  – we can solve only if sequence is simple enough that we can do it by inspection (from the definition of the z-transform)

Example

- consider the sequence

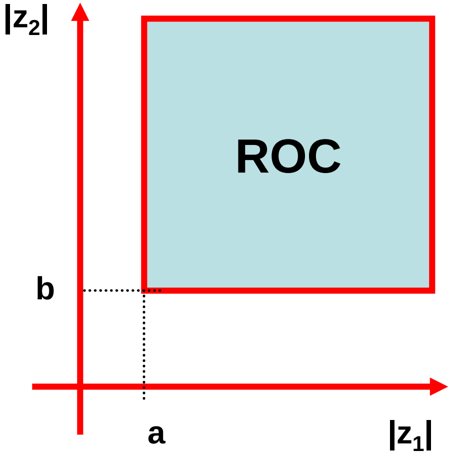
$$x[n_1,n_2]=a^{n_1}b^{n_2}u[n_1,n_2]$$

- the z-transform is

$$X(z_1,z_2)=\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\left(az_1^{-1}\right)^{n_1}\left(az_2^{-1}\right)^{n_2}$$

$$=\sum_{n_1=0}^{\infty}\left(az_1^{-1}\right)^{n_1}\sum_{n_2=0}^{\infty}\left(az_2^{-1}\right)^{n_2}$$

$$=\frac{1}{1-az_1^{-1}}\frac{1}{1-bz_2^{-1}},\quad |z_1|>a,|z_2|>b$$



Any questions?