Introduction to Linear Image Processing

IPAM - UCLA

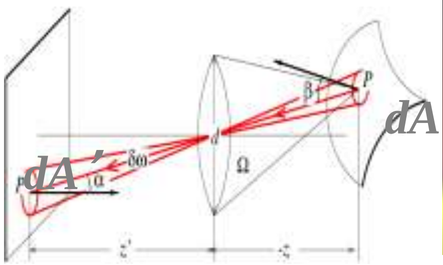
July 22, 2013

Iasonas Kokkinos
Center for Visual Computing
Ecole Centrale Paris / INRIA Saclay

Image Sciences in a nutshell

Image Processing

Image to Image



Imaging

Physics to Image

Computer Graphics

Symbols to Image

Computer Vision

Image to Symbols

Images as functions

Continuous

$$f : R^2 \to R^d$$

$$x = (h, v)$$

Discrete

$$f : Z^2 \to R^d$$

$$n = (n_1, n_2)$$

d=1: Gray
d=3: Color

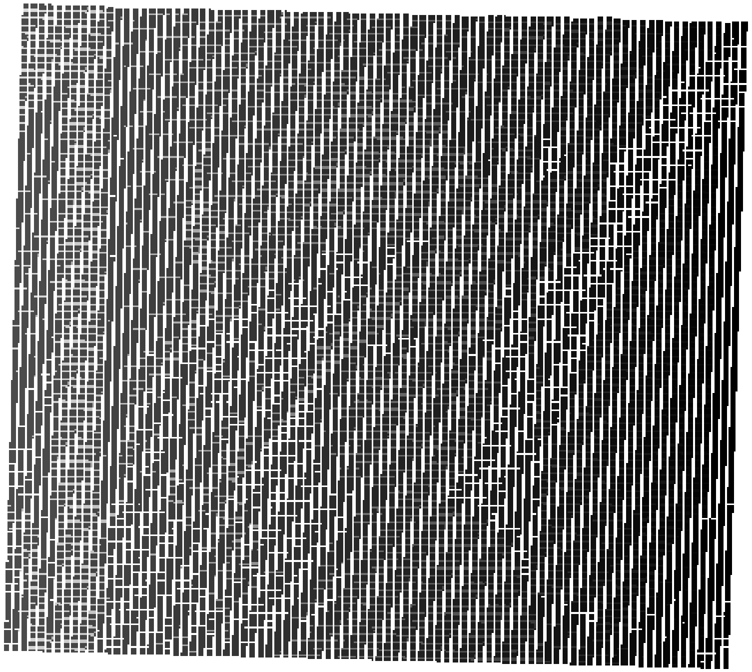


Image Denoising

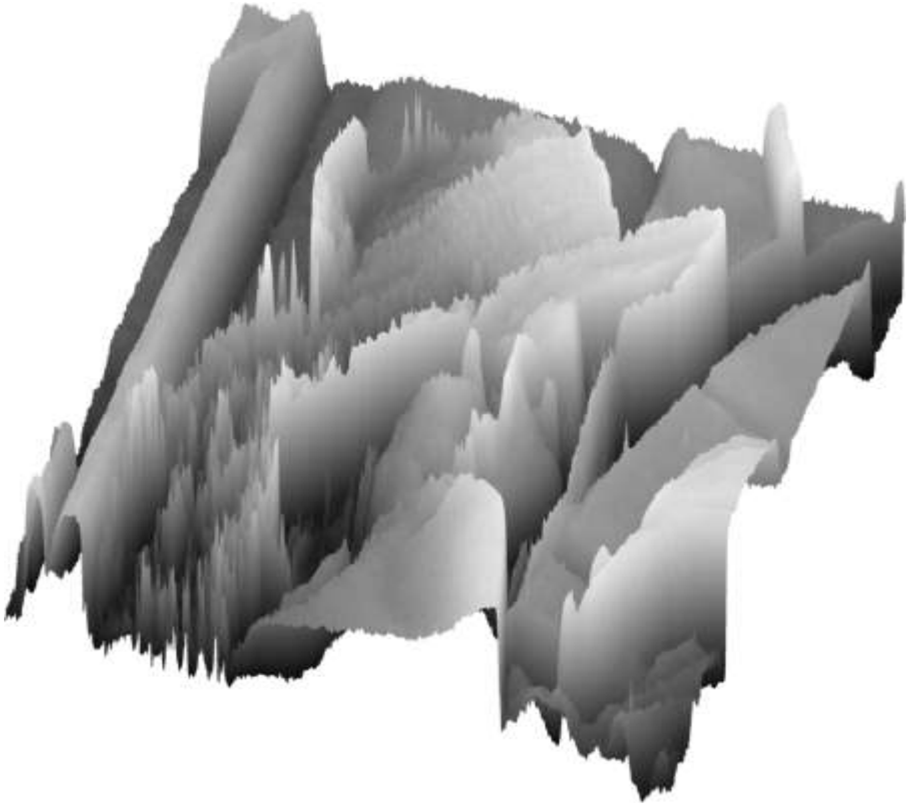
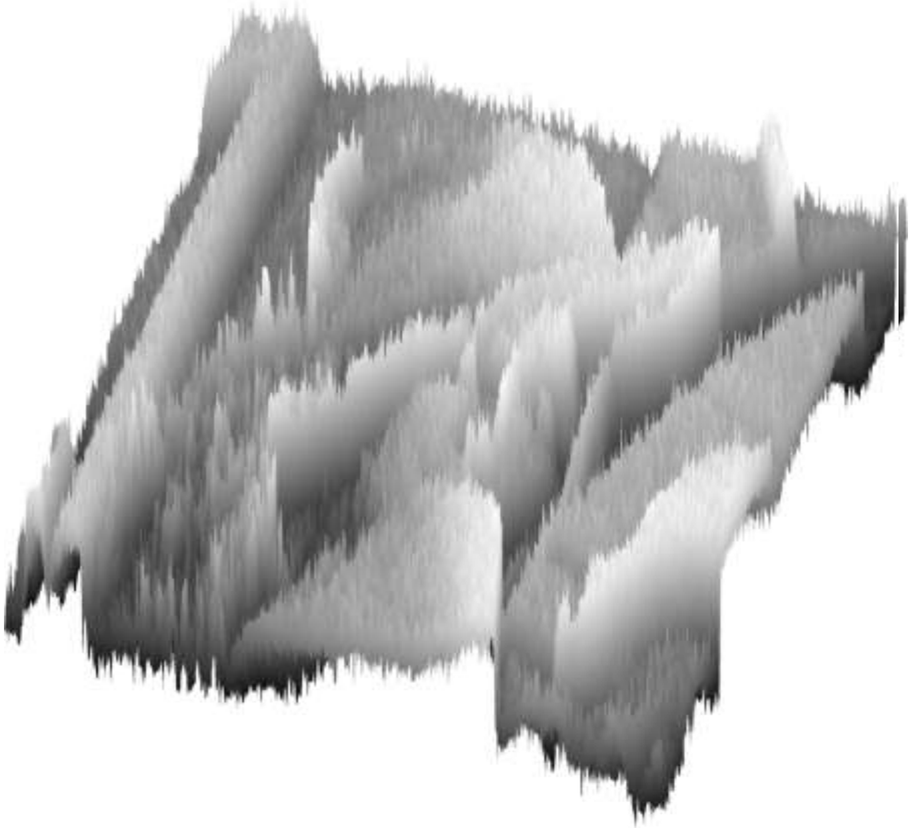


Image Denoising



Key assumption: clean image is smooth

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0								

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10							

Slide Source: S. Seitz

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20						

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30					

Slide Source: S. Seitz

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30				

Slide Source: S. Seitz

Moving Average in 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20				20	10	
	0	20	40	60	60	60	40	20	
	0		60	90	90	90	60		
	0		50	80	80	90	60		
	0		50	80	80	90	60		
	0	20		50	50	60	40	20	
	10	20					20	10	
	10	10	10	0	0	0	0	0	

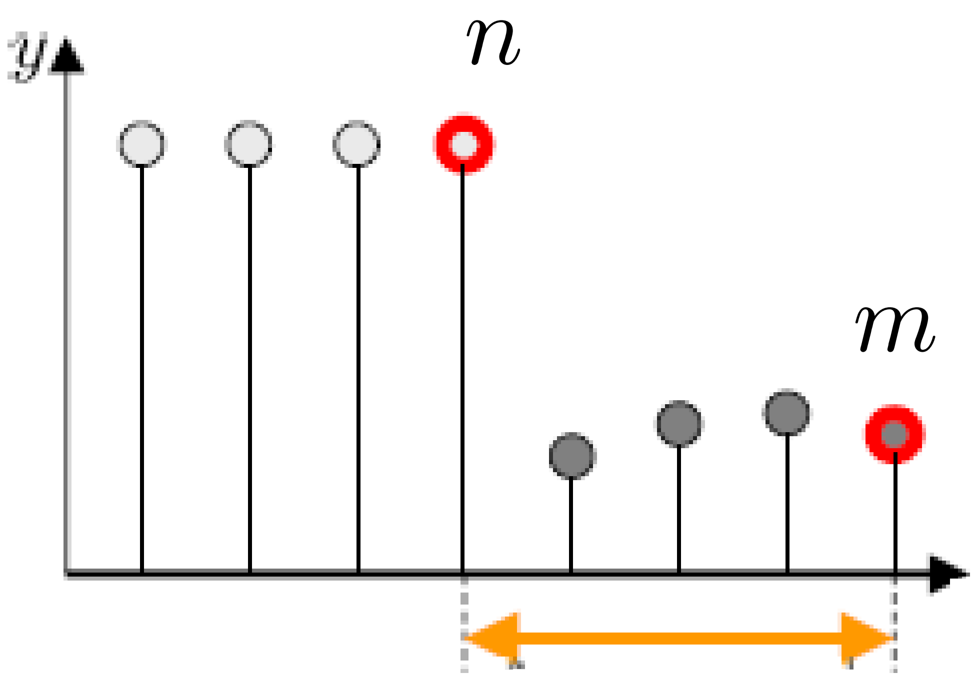
Denoising: input

Denoising: first application of averaring filter

Denoising: tenth application of denoising filter

Denoising: application of larger box filter

Weighted averaging



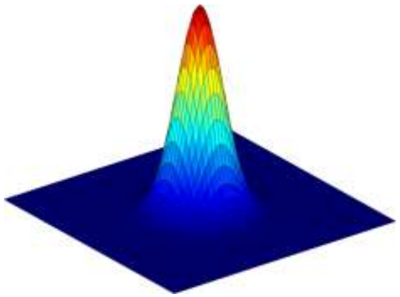
$$g[n] = \sum_m w(|n-m|) f(m)$$

$$w[k] = \begin{cases} \frac{1}{2l+1}, & k \le l \\ 0, & \text{otherwise} \end{cases}$$

Weighting kernel

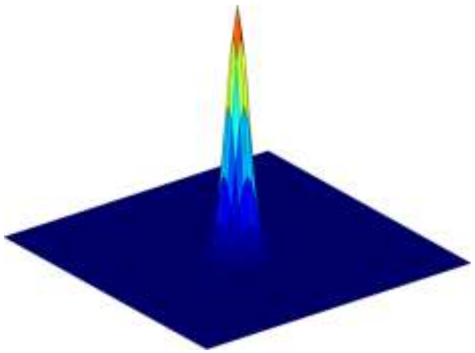
Gaussian function:

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

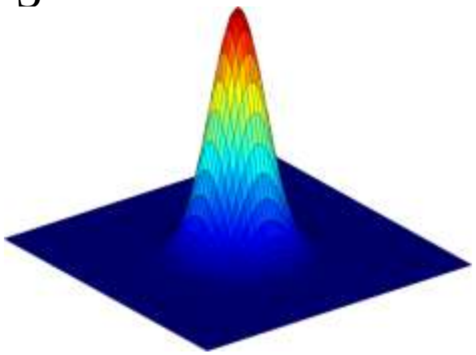


Standard deviation, σ: determines spatial support

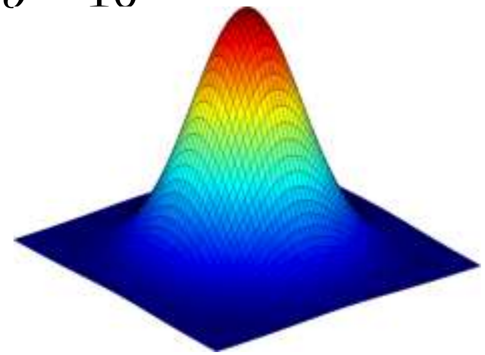
$\sigma = 2$



$\sigma = 5$



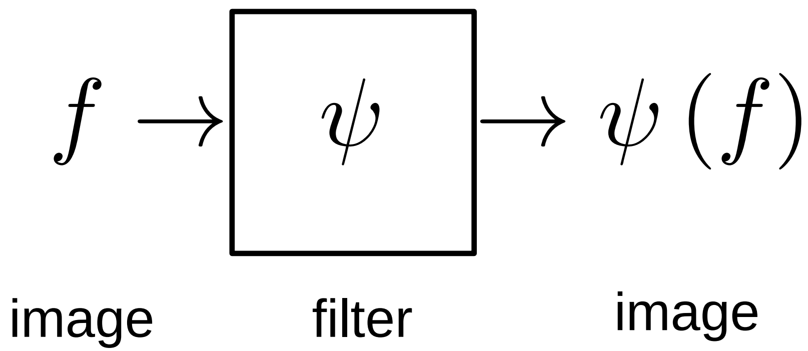
$\sigma = 10$



Moving average

Gaussian blur

Image Processing



Linear Image Processing

Linearity

$$\psi\left(\alpha f+\beta h\right)=\alpha\psi\left(f\right)+\beta\psi\left(h\right)$$

$$\psi\left(\sum_k \alpha_k f_k\right)=\sum_k \alpha_k\psi\left(f_k\right)$$

Translation Invariance

$$f^c(x)\doteq f(x-c)$$

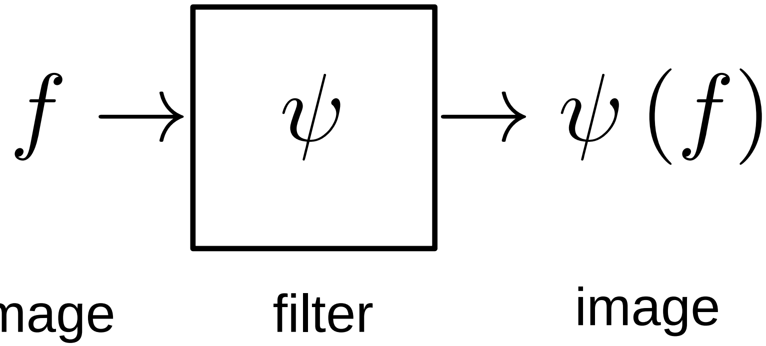
$$\psi\left(f^c\right)=\left[\psi\left(f\right)\right]^c$$

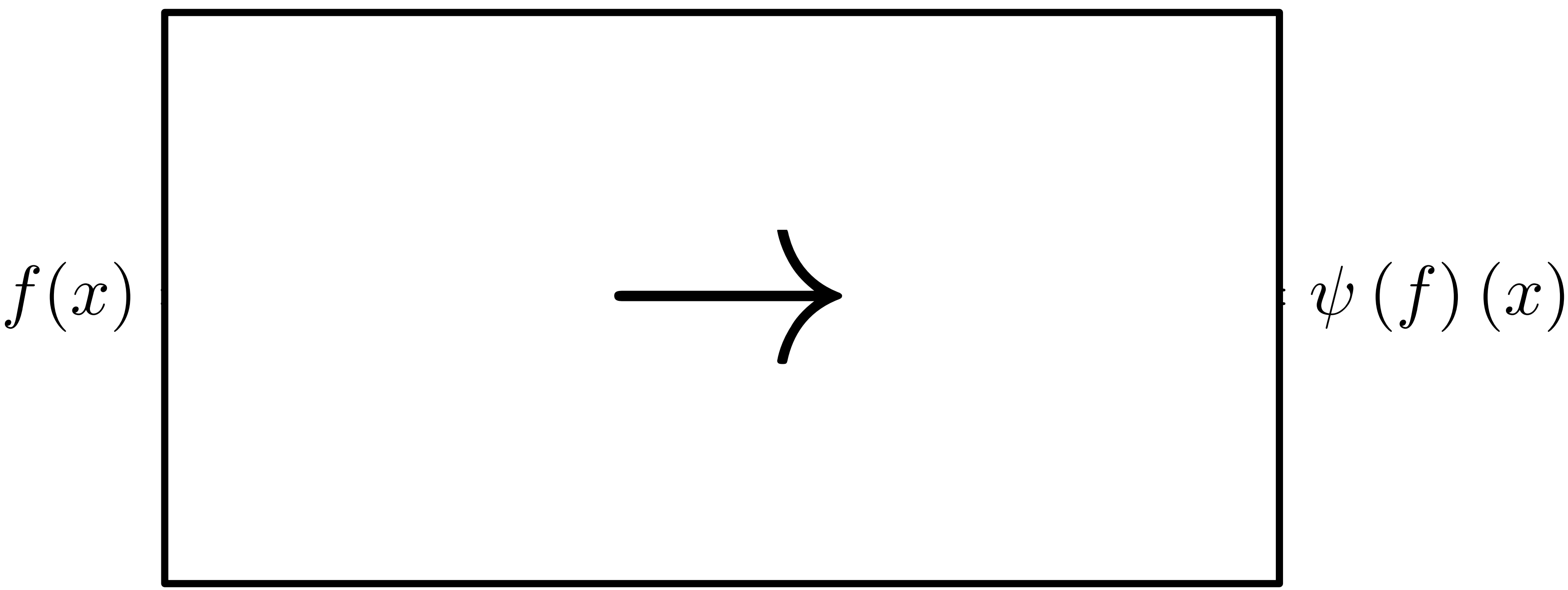
Linear, Translation-Invariant (LTI) system

$$f_k(x)\quad\rightarrow\quad g_k(x)$$

$$\sum_k \alpha_k f_k(x-c)\quad\rightarrow\quad\sum_k \alpha_k g_k(x-c)$$

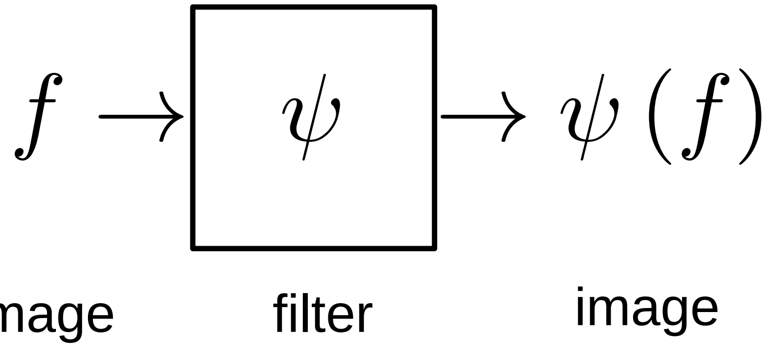
Linear Image Processing





From time-invariance: useful bases.

Linear Image Processing



$$f(x) = \left\{ \begin{array}{rcr} a_0 b_0(x) & \rightarrow & a_0 h_0(x) \\ +a_1 b_1(x) & \rightarrow & +a_1 h_1(x) \\ \vdots & & \vdots \\ +a_k b_k(x) & \rightarrow & +a_k h_k(x) \end{array} \right\} = \psi(f)(x)$$

From time-invariance: useful bases.

Linear algebra reminder

$$\mathbf{u} \in R^N$$

Basis: N linearly independent vectors $\{\mathbf{v}_i\},\ i = 1, \ldots, N$

Expansion on basis: $$\mathbf{u} = \sum_i c_i \mathbf{v}_i$$

Orthonormal basis: $$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases}$$

Expansion coefficients: $$\langle \mathbf{v}_i, \mathbf{u} \rangle = c_i$$

Expansion: $$\mathbf{u} = \sum_i \langle \mathbf{v}_i, \mathbf{u} \rangle \mathbf{v}_i$$

Canonical basis

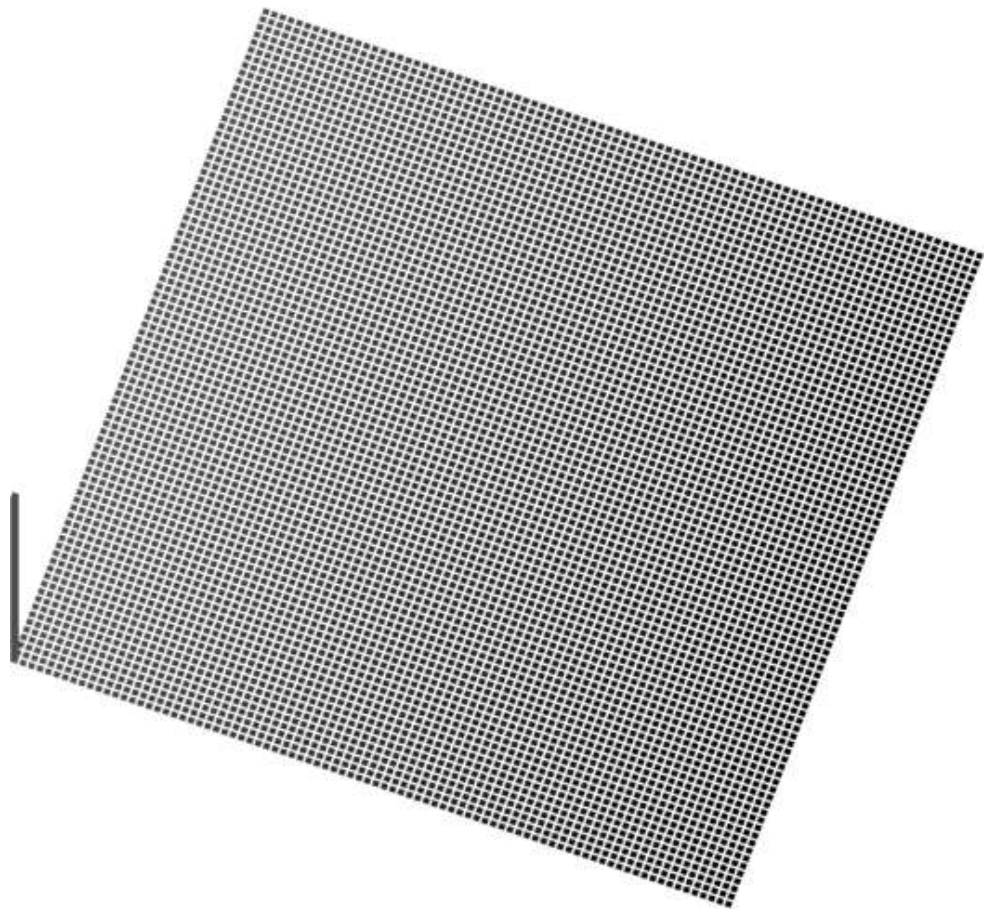
$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_1 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{e}_1} + u_2 \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{\mathbf{e}_2} + u_3 \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{e}_3}$$

$$\mathbf{u} = \sum_i u_i \mathbf{e}_i = \sum_i \langle \mathbf{e}_i, \mathbf{u} \rangle \mathbf{e}_i$$

Canonical basis for 2D signals

Kronecker delta

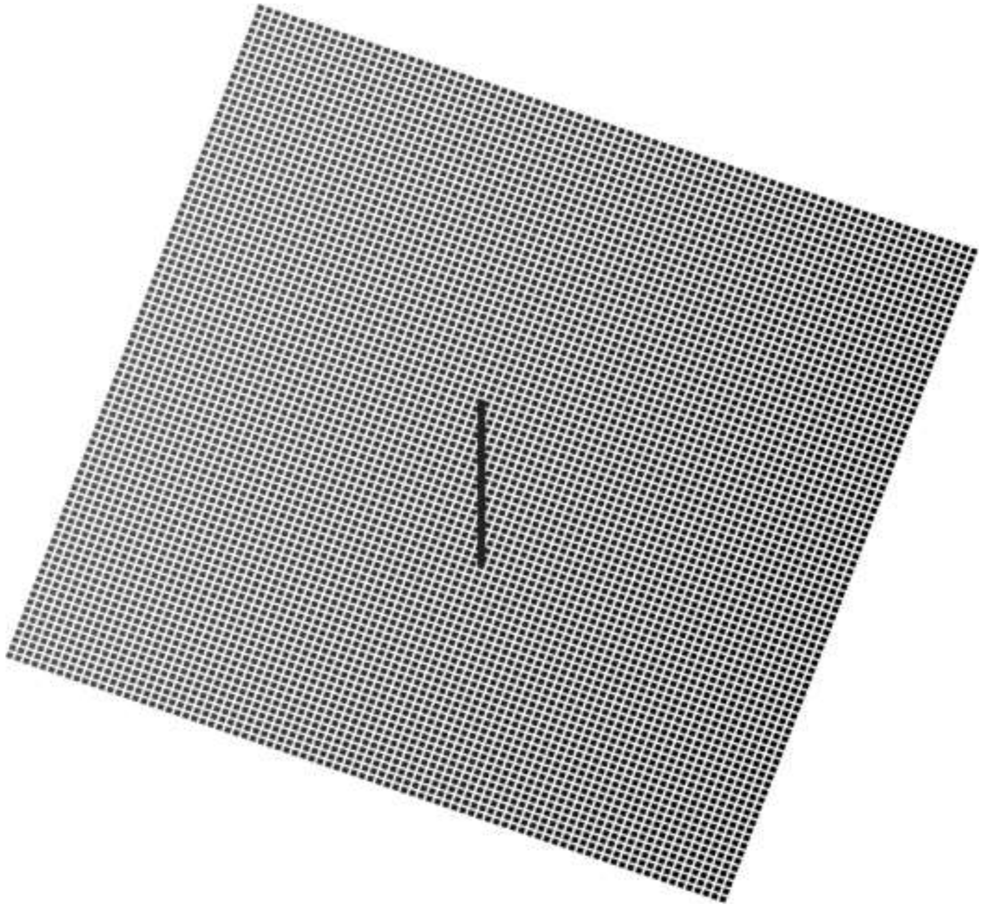
$$d_k[n] = \begin{cases} 1, & n = k \\ 0, & \text{otherwise} \end{cases}$$



Canonical basis for 2D signals

Kronecker delta

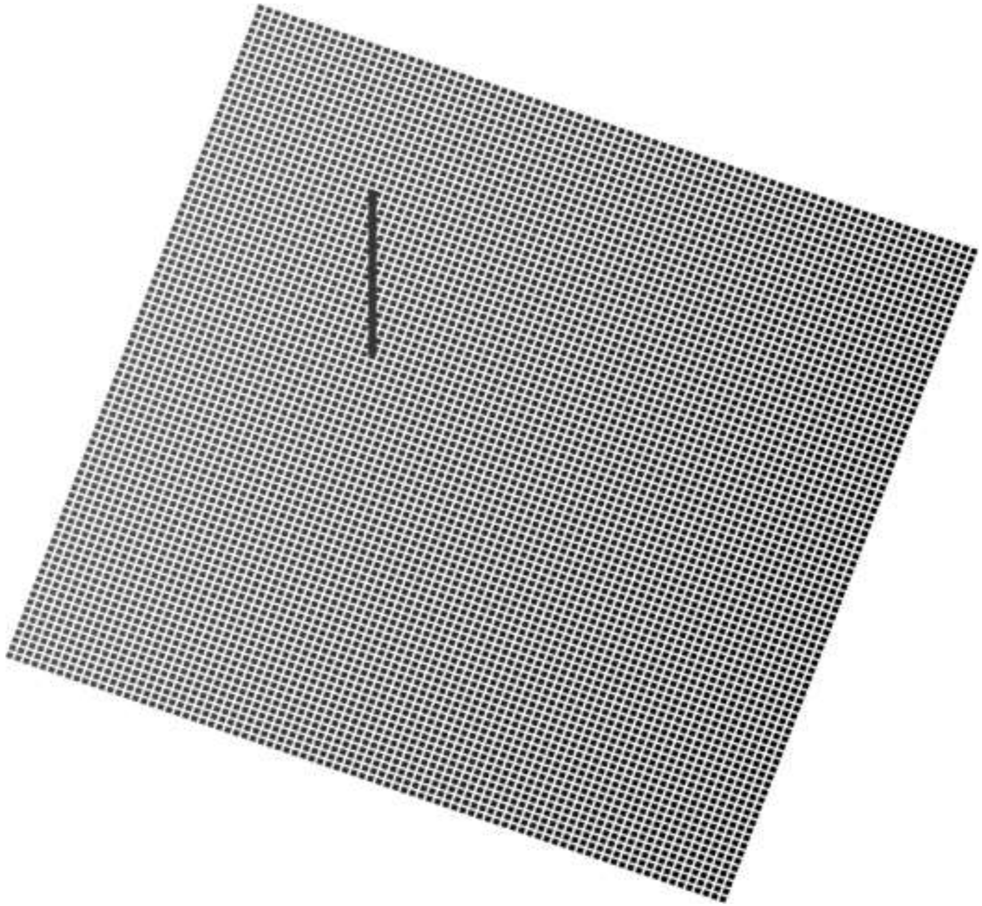
$$d_k[n] = \begin{cases} 1, & n = k \\ 0, & \text{otherwise} \end{cases}$$



Canonical basis for 2D signals

Kronecker delta

$$d_k[n] = \begin{cases} 1, & n = k \\ 0, & \text{otherwise} \end{cases}$$



Canonical basis for signals: expansion

Signal expansion: $$g[n] = \sum_k c_k d_k[n]$$

Identify terms: $$g[k] = c_k$$

Rewrite: $$d_k[n] = d[n-k]$$

$$d[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases}$$

Unit sample function

Sifting property: $$g[n] = \sum_k g[k] d[n-k]$$

Canonical basis for signals and LTI filters

unit sample

$$d[n] \rightarrow h[n]$$

impulse response

$$d[n-k] \rightarrow h[n-k]$$

Translation-invariance

Any signal: $$g[n] = \sum_k g[k] d[n-k]$$

By linearity:

$$\psi(g) = \sum_k g[k] h[n-k] \doteq g[n] * h[n]$$

Convolution sum

Output of **any** LSI filter for **any** input:
convolution of input with filter's impulse response

Convolution – discrete and continuous

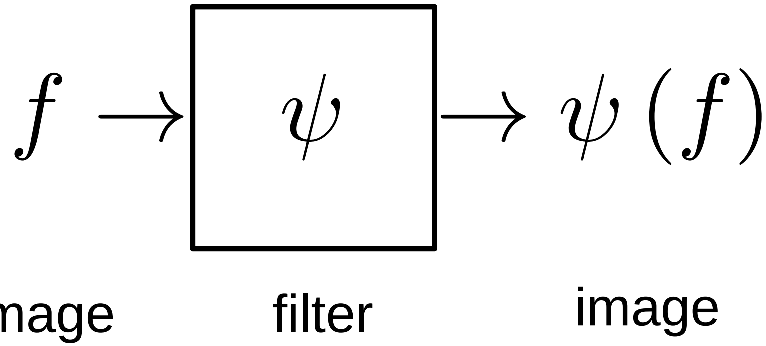
2D convolution sum:

$$\begin{aligned} f[n_1, n_2] &= \sum_{k_1, k_2} g[k_1, k_2] h[n_1 - k_1, n_2 - k_2] \\ &= g[n_1, n_2] * h[n_1, n_2] \end{aligned}$$

2D convolution integral:

$$\begin{aligned} f(x, y) &= \iint g(a, b) h(x - a, y - b) da db \\ &= g(x, y) * h(x, y) \end{aligned}$$

Linear Image Processing



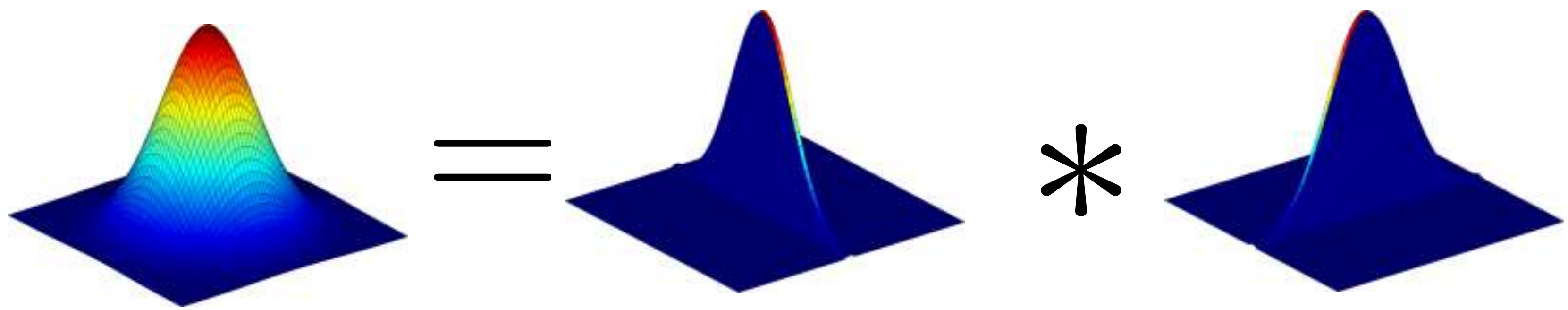
$$f(x) = \left\{ \begin{array}{rcr} a_0 b_0(x) & \rightarrow & a_0 h_0(x) \\ +a_1 b_1(x) & \rightarrow & +a_1 h_1(x) \\ \vdots & & \vdots \\ +a_k b_k(x) & \rightarrow & +a_k h_k(x) \end{array} \right\} = \psi(f)(x)$$

From time-invariance: useful bases.

Associative property & efficiency

Associative Property: $f * [g * h] = [f * g] * h$

Separability of Gaussian:



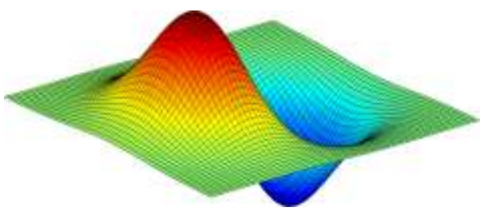
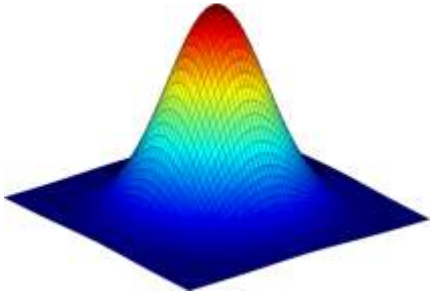
Slow

Fast

Associative property & accuracy

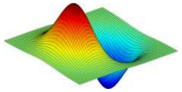
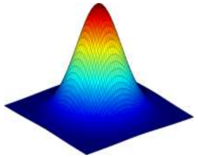
Associative Property: $f * [g * h] = [f * g] * h$

Derivative of Gaussian:

 $=$  $* \dfrac{\partial}{\partial x}$

exact

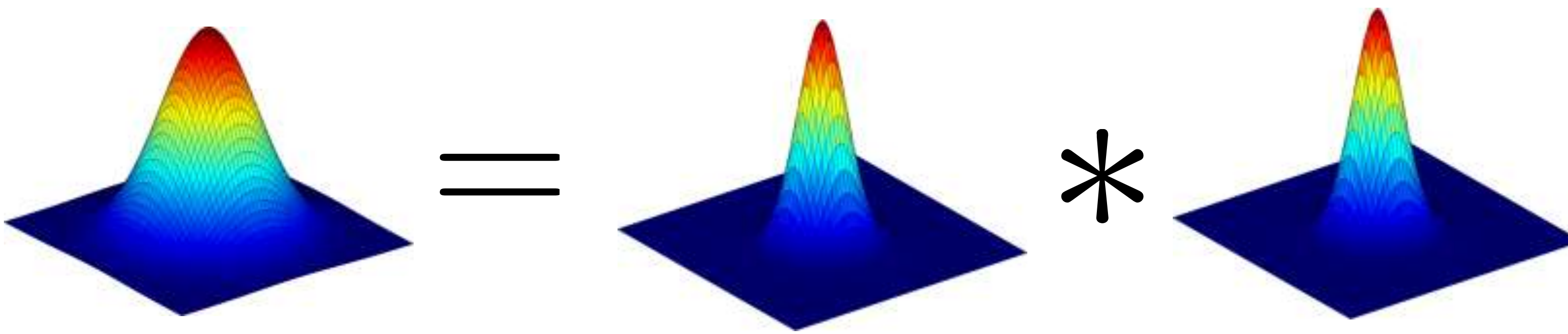
approximate

 $*$  $=$  $*$  $* \dfrac{\partial}{\partial x}$

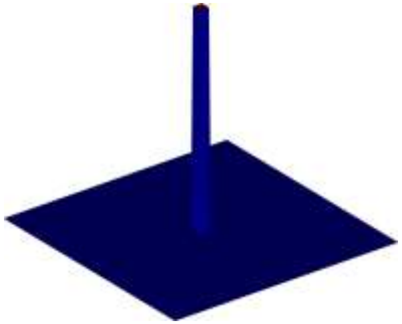
Associative property & multi-scale processing

Associative Property: $f * [g * h] = [f * g] * h$

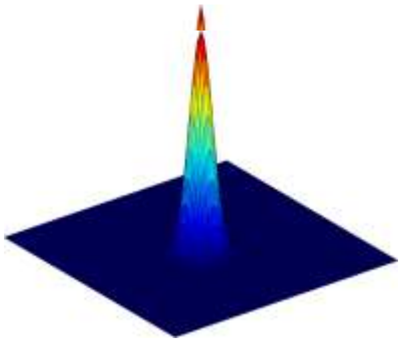
Semi-group property of Gaussian:



Denoising: first application of averaging kernel



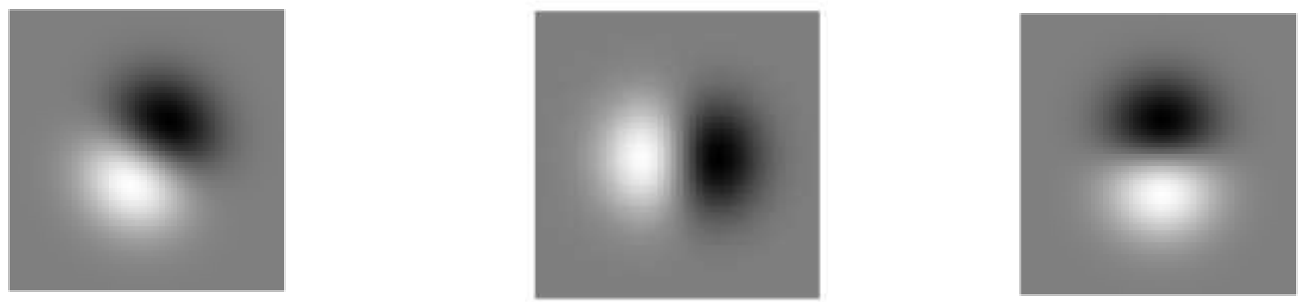
Denoising: 10^{th} application of denoising kernel



Distributive property & efficiency

Distributive property: $$f * (g + h) = f * g + f * h$$

Steerable fliter: $$g_\theta(x, y) = \cos(\theta) g_0(x, y) + \sin(\theta) g_{\pi/2}(x, y)$$



$$I \qquad I * g_\theta = \cos(\theta)(I * g_0) + \sin(\theta)(I * g_{\pi/2})$$

W. Freeman and E. Adelson, ‘The design and use of steerable filters’, PAMI, 1991

Linear algebra reminder: eigenvectors

$$\mathbf{M} : N \times N$$

Eigenvectors: $$\mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \; i = 1, \ldots, N$$

Full-rank, real and symmetric: eigenbasis

$$\mathbf{u} = \sum_k \underbrace{\langle \mathbf{v}_k, \mathbf{u} \rangle}_{c_k} \mathbf{v}_k$$

$$\mathbf{M}\mathbf{u} = \sum_k c_k \mathbf{M}\mathbf{v}_k = \sum_k \underbrace{c_k \lambda_k}_{c'_k} \mathbf{v}_k$$

$$\mathbf{M}\left(\mathbf{M}\mathbf{u}\right) = \sum_k \underbrace{c_k \lambda_k^2}_{c''_k} \mathbf{v}_k$$

Eigenvectors and eigenfunctions

Eigenvector: $$\mathbf{M}\mathbf{v} = \lambda \mathbf{v}$$

Eigenfunction: $$\psi(b) = \lambda b$$

Input: $$f = \sum_k a_k b_k$$

Output: $$\psi(f) = \sum_k a_k \psi(b_k)$$

$$f \leftrightarrow \{a_k\} \quad \psi(f) \leftrightarrow \{a_k \lambda_k\}$$

Eigenfunctions for LTI filters

LTI filter: $$\psi\left(g\right)[n] = \sum_k h[k]g[n-k]$$

Let's guess: $$b_\omega[n] = \exp\left(j\omega n\right) = \cos(\omega n) + j\sin(\omega n)$$

It works: $$\begin{aligned}\psi\left(b_\omega\right)[n] &= \sum_k h[n]b_\omega[n-k] \\ &= \sum_k h[k]\exp\left(j\omega[n-k]\right) \\ &= \sum_k h[k]\exp\left(-j\omega k\right)\exp\left(j\omega n\right) \\ &= H(\omega)b_\omega[n]\end{aligned}$$

Frequency response: $$H(\omega) \doteq \sum_k h[k]\exp\left(-j\omega k\right)$$

Expansion on harmonic basis

From orthonormality: $$\mathbf{u} = \sum_k \langle \mathbf{u}, \mathbf{v}_k \rangle \mathbf{v}_k$$

Inner product for complex functions: $$\langle f, g \rangle = \sum_n f[n] g^*[n]$$

Discrete-time: $$F(\omega) \doteq \langle f, b_\omega \rangle = \sum_n f[n] e^{-j\omega n}$$

Continuous-time: $$F(\omega) \doteq \int_{-\infty}^{\infty} f(t) \exp(-j\omega t) dt$$

Change of basis

Canonical expansion: $$\mathbf{u} = \sum_k u_k \mathbf{e}_k$$

Eigenbasis expansion: $$\mathbf{u} = \sum_k \underbrace{\langle \mathbf{u}, \mathbf{v}_k \rangle}_{c_k} \mathbf{v}_k$$

Rotation matrix from eigenbasis:

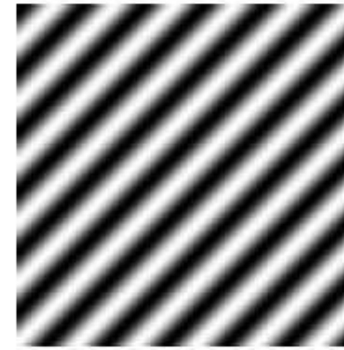
$$\mathbf{c}^T = \mathbf{u}^T \underbrace{\left[\begin{array}{c|c|c} \mathbf{v}_1 & \ldots & \mathbf{v}_N \end{array} \right]}_{\mathbf{V}}$$

Fourier transform: change of basis

Rotation from canonical basis to eigenfunction basis

Fourier Analysis

$$F(\omega_1) \cdot e^{j\omega_1 x}$$

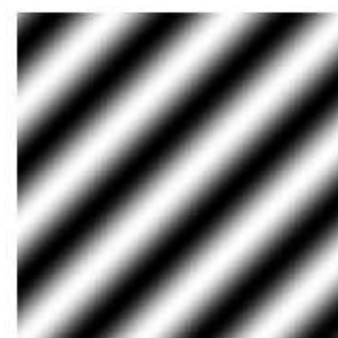


$$+$$

$$= \quad F(\omega_2) \cdot e^{j\omega_2 x}$$

$$+$$

$$F(\omega_K) \cdot e^{j\omega_K x}$$



Fourier synthesis equation

Continuous-time:

$$f(x,y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_1, \omega_2) e^{j(\omega_1 x + \omega_2 x)} dx dy$$

Discrete-time:

$$f[n,m] = \frac{1}{4\pi^2} \int_{0}^{2\pi} \int_{0}^{2\pi} F(\omega_1, \omega_2) e^{j(\omega_1 n + \omega_2 m)} d\omega_1 d\omega_2$$

Convolution theorem of Fourier transform

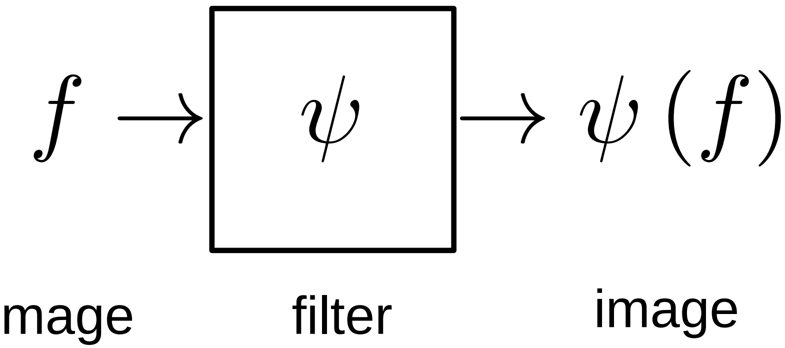
Input expansion:
$$f[n] = \int_\omega F(\omega) e^{j\omega n} \mathrm{d}\omega$$

Output:
$$\psi\left(f\right)[n] = \int_\omega F(\omega)\psi\left(e^{\omega n}\right)\mathrm{d}\omega$$
$$= \int_\omega F(\omega) H(\omega) e^{j\omega n} \mathrm{d}\omega$$

$$H(\omega) \doteq \sum_k h[k] e^{-j\omega k}$$

Expansions:
$$f[n] \quad \leftrightarrow \quad F(\omega)$$
$$\psi(f)[n] \quad \leftrightarrow \quad F(\omega) H(\omega)$$
$$f[n] * h[n] \quad \leftrightarrow \quad F(\omega) \cdot H(\omega)$$

Linear Image Processing



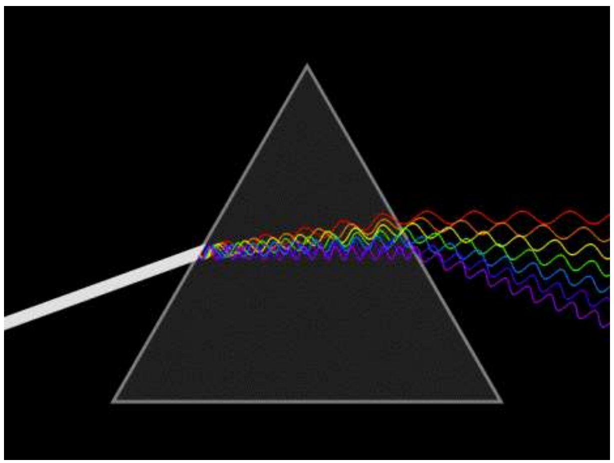
$$f(x) = \left\{ \begin{array}{rcr} a_0 b_0(x) & \rightarrow & a_0 h_0(x) \\ +a_1 b_1(x) & \rightarrow & +a_1 h_1(x) \\ \vdots & & \vdots \\ +a_k b_k(x) & \rightarrow & +a_k h_k(x) \end{array} \right\} = \psi(f)(x)$$

From time-invariance: useful bases.

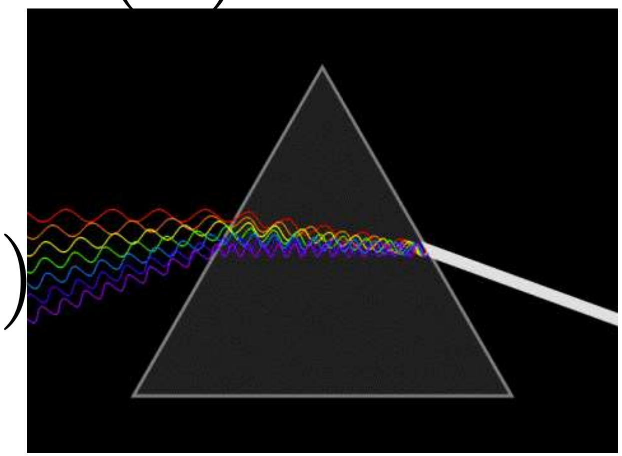
Convolution theorem

$X(\omega)$ $Y(\omega)$

$x(t)$



$H(\omega)$



$y(t)$

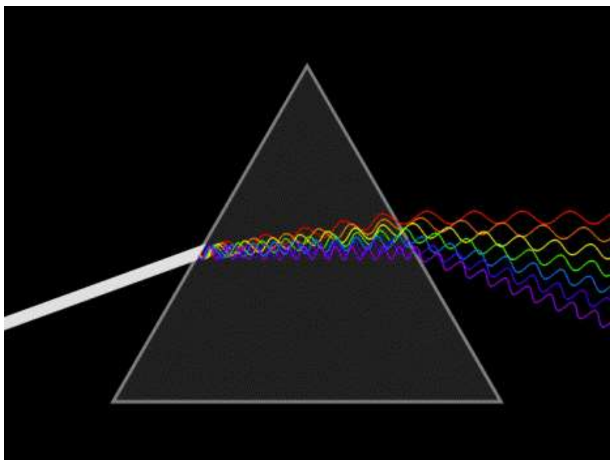
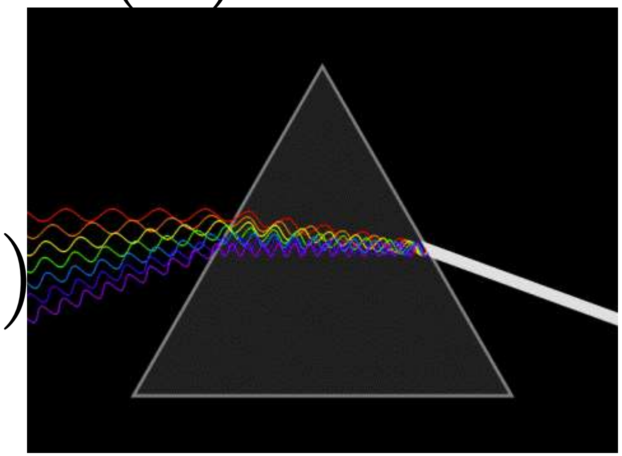
Fourier Analysis

Fourier Synthesis

$$Y(\omega) = H(\omega)X(\omega)$$

Convolution theorem and efficiency

$X(\omega)$ $Y(\omega)$

$x(t)$  $H(\omega)$  $y(t)$

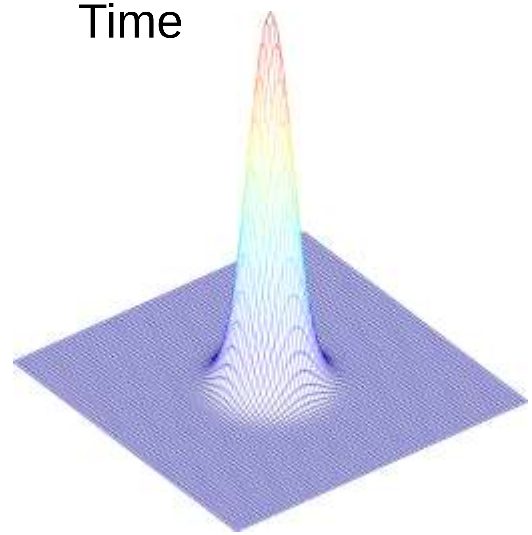
Fast Fourier Transform   Fast Fourier Transform

$$Y(\omega) = H(\omega)X(\omega)$$

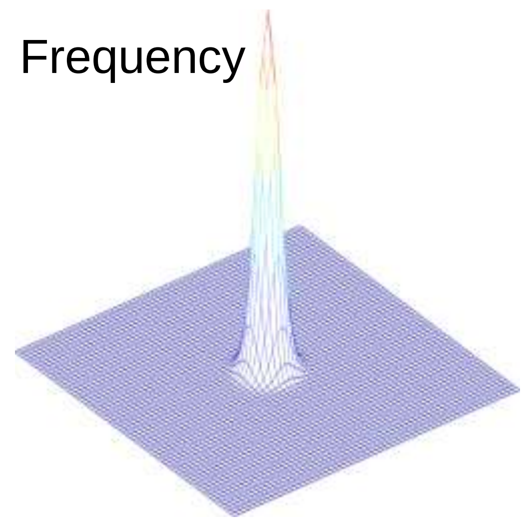
$$O(NK) \rightarrow O(N \log N)$$

Gaussian blur

Time

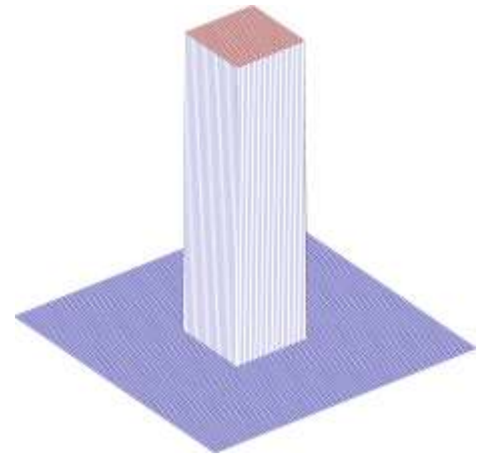


Frequency

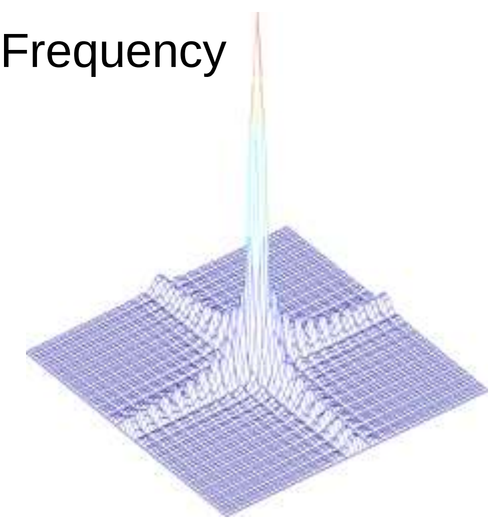


Moving average

Time



Frequency



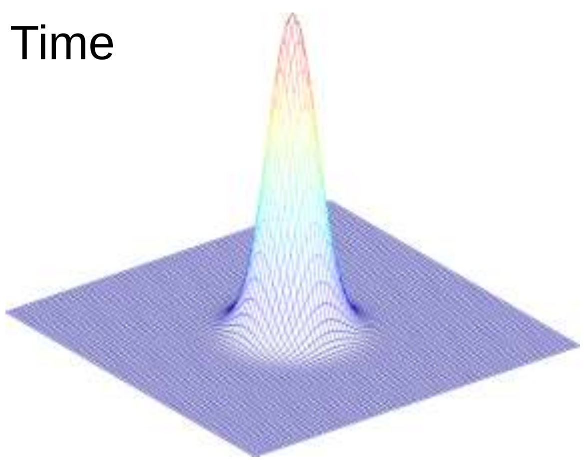
Modulation property and Gabor filters

Modulation property: $$f(x) \leftrightarrow F(\omega)$$

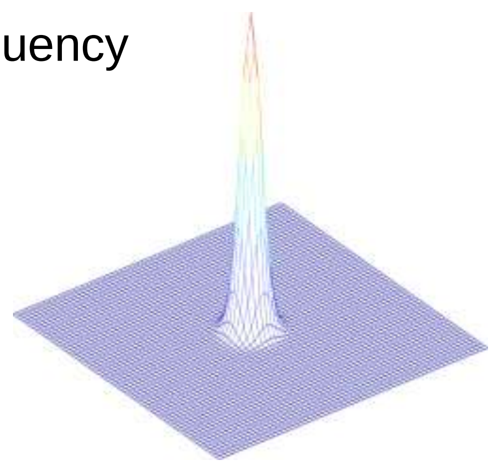
$$f(x)e^{j\omega_c x} \leftrightarrow F(\omega - \omega_c)$$

Gaussian: $$\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \leftrightarrow e^{-\frac{(\omega_x^2+\omega_y^2)\sigma^2}{2}}$$

Time



Frequency



Modulation property and Gabor filters

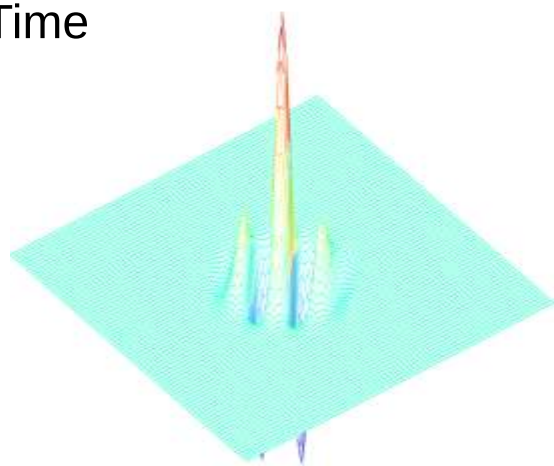
Modulation property: $$f(x) \leftrightarrow F(\omega)$$

$$f(x)e^{j\omega_c x} \leftrightarrow F(\omega - \omega_c)$$

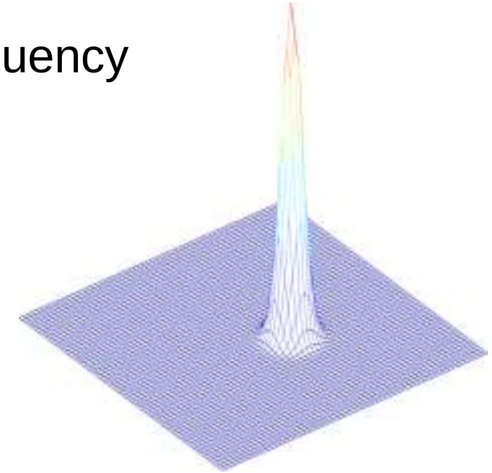
Gaussian: $$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \leftrightarrow e^{-\frac{(\omega_x^2+\omega_y^2)\sigma^2}{2}}$$

Gabor: $$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j(\omega_x^c x+\omega_y^c y)} \leftrightarrow e^{-j\frac{\sigma^2((\omega_x-\omega_x^c)^2+(\omega_y-\omega_y^c)^2)}{2}}$$

Time



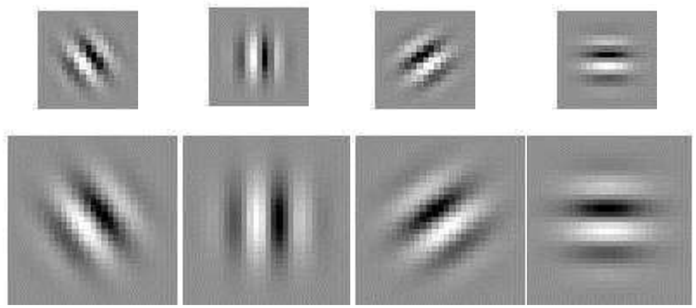
Frequency



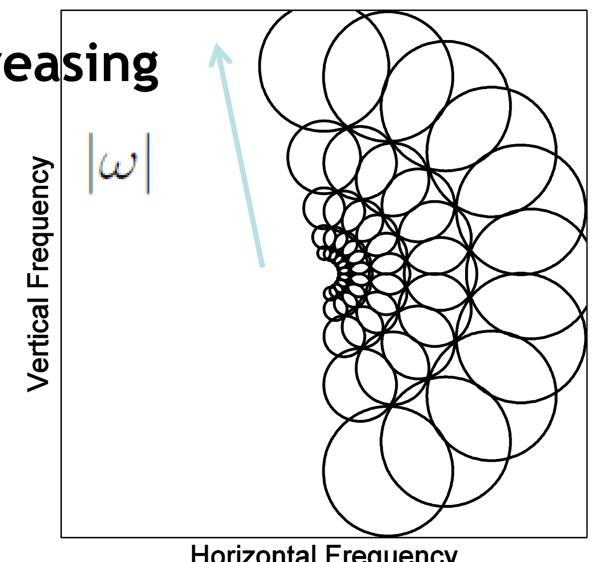
2D Gabor filterbank

Consider many combinations of $|\omega|$ and $\angle\omega$

Increasing $|\omega|$

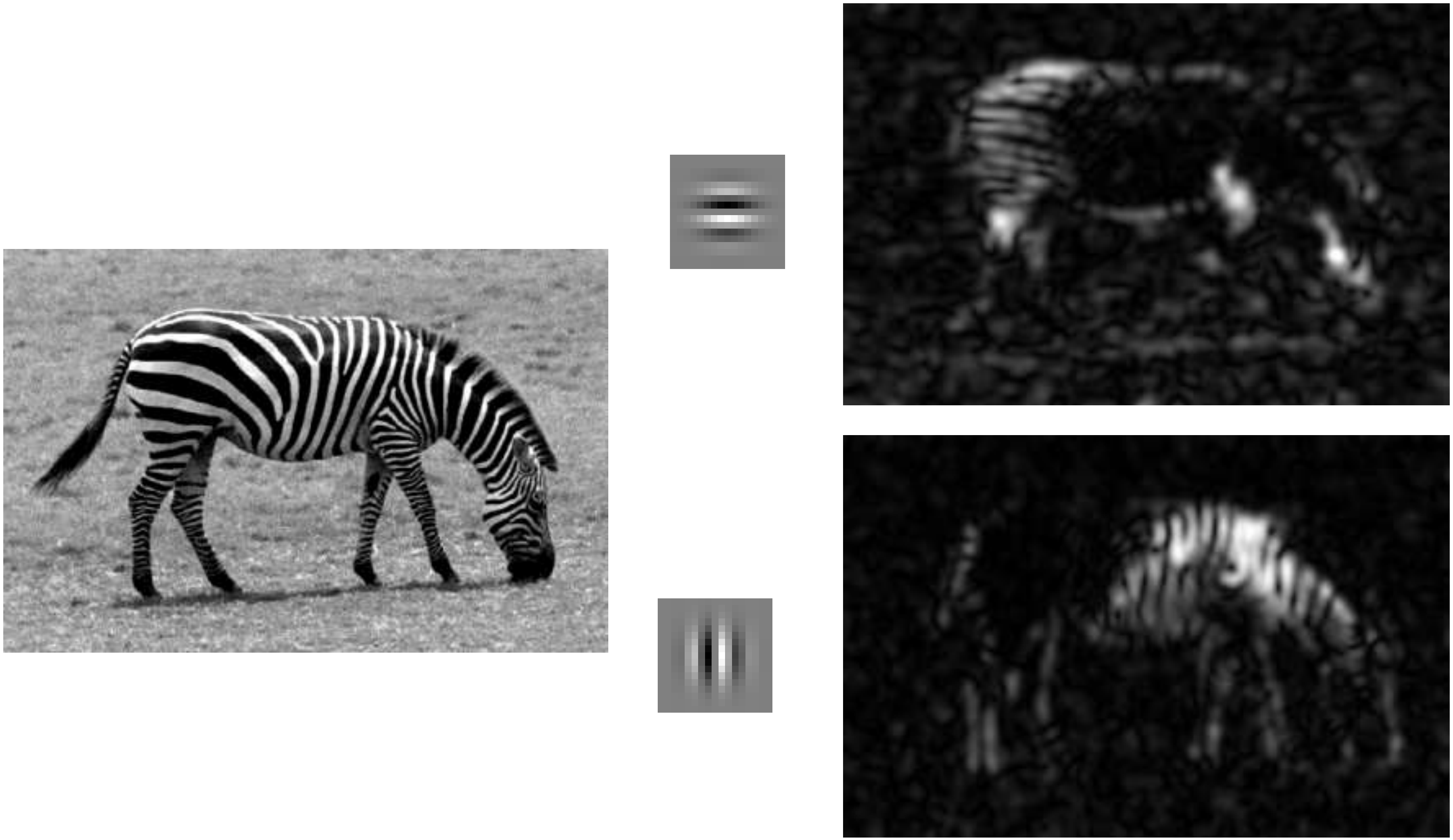


Increasing $|\omega|$

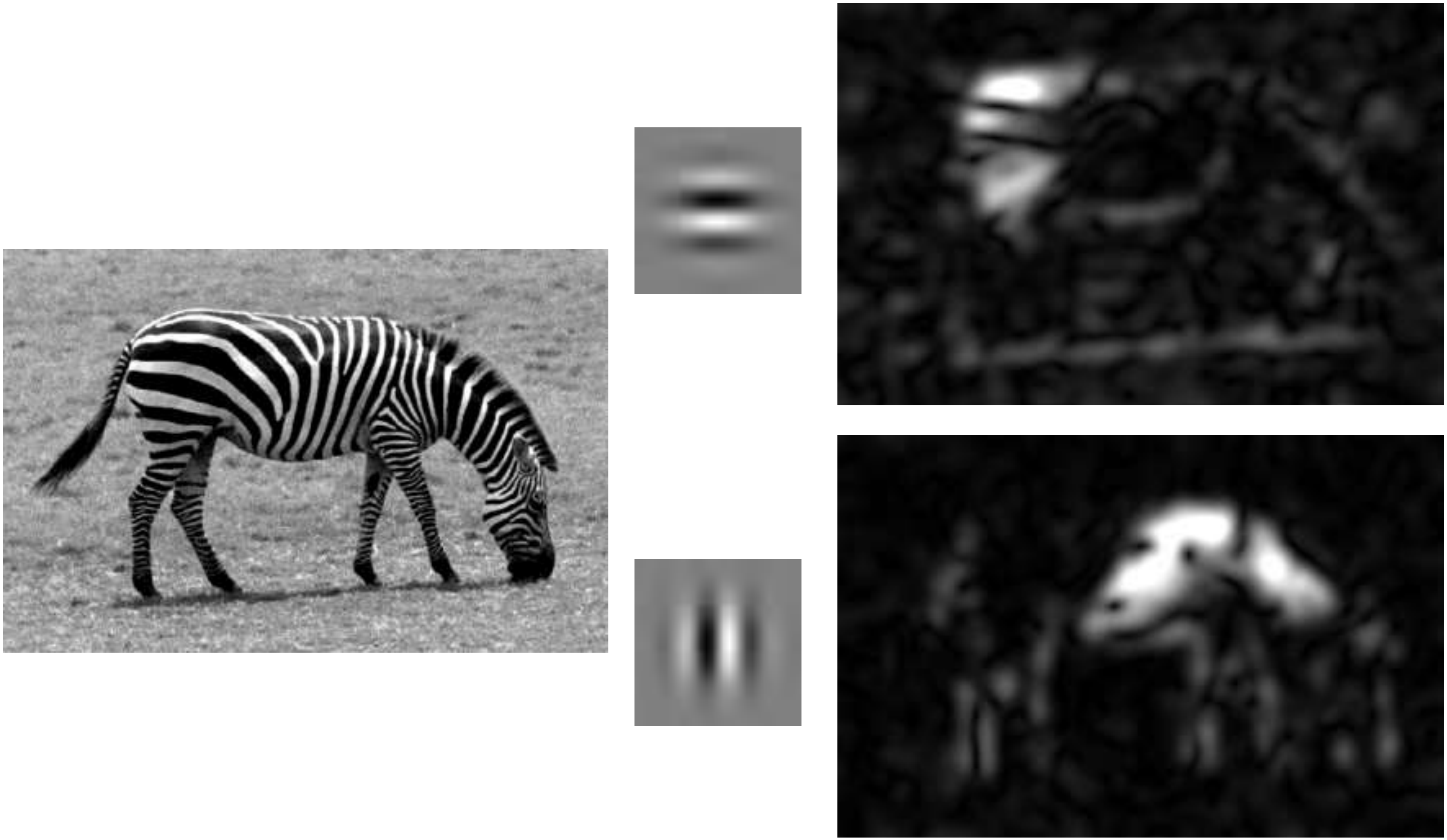


Frequency responce isocurves

2D Gabor filterbank and texture analysis



2D Gabor filterbank and texture analysis



Summary

- Linear Time-Invariant filters
- Convolution
- Fourier Transform
- (Derivative-of) Gaussian filters
- Steerable filters
- Gabor filters

Thursday's lecture: Pyramids, Scale-Invariant Blobs/Ridges, SIFT, HOG, Log-polar features, Harmonic analysis on surfaces…

Further reading:

Fast recursive filters:
Recursively implementing the Gaussian and its Derivatives - R. Deriche, 1993
Recursive implementation of the Gaussian filter. I. Young, L. Vliet, 1995
Fast IIR Isotropic 2D Complex Gabor Filters with Boundary Initialization,
A Bernardino, J. Santos-Victor, TIP, 2006
Wavelets:
A Wavelet Tour of Signal Processing, S. Mallat, 2008