Computer Vision

Edit By *Dr. Khawla Hussein*

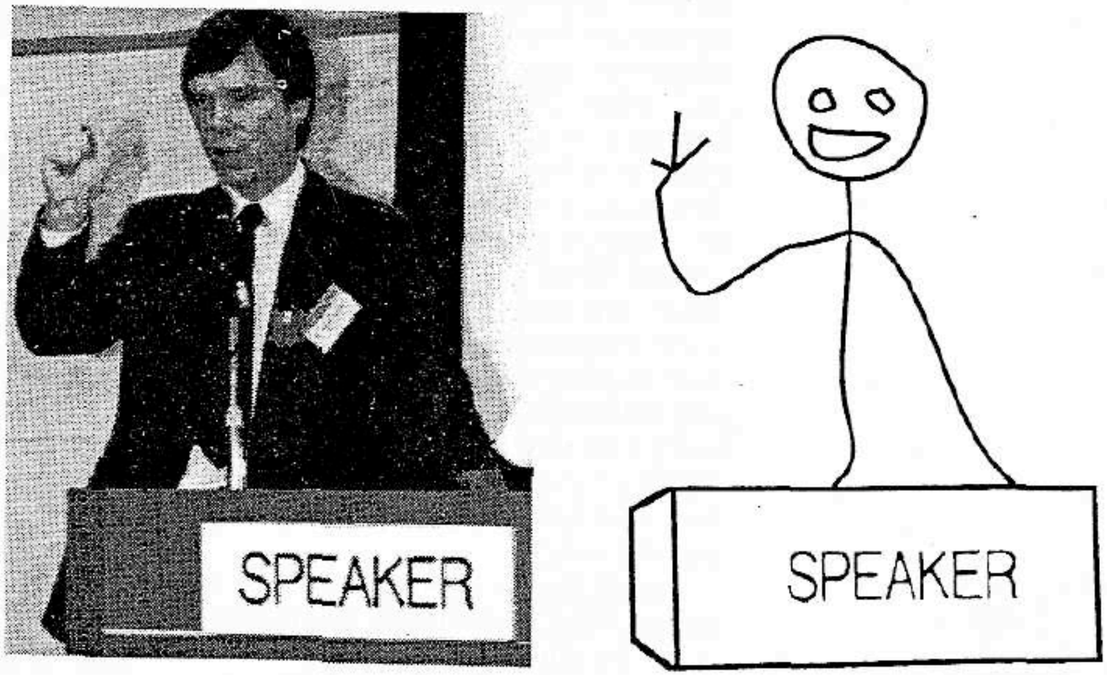
Lecture 3 & 4

Linear filters and Edge Detection

• Proposition 1. The primary task of early vision is to **deliver a small set of useful measurements about each observable location in the plenoptic function.**

• Proposition 2. The elemental operations of early vision involve **the measurement of local change along various directions within the plenoptic function.**

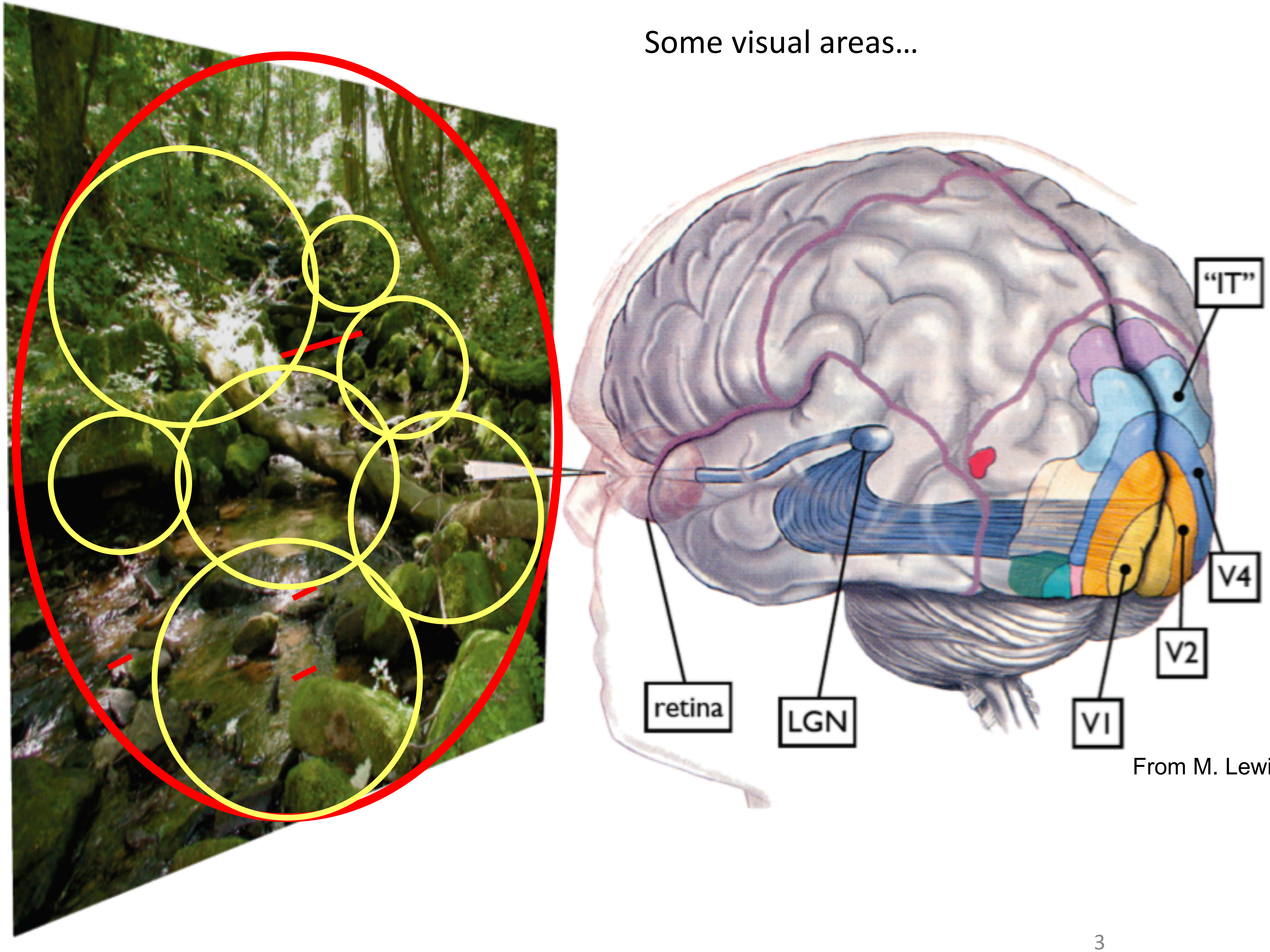
• Goal: to transform the image into other representations (rather than pixel values) that makes scene information **more explicit**



What we think we see   What we really see

Cavanagh, Perception 95

Some visual areas...

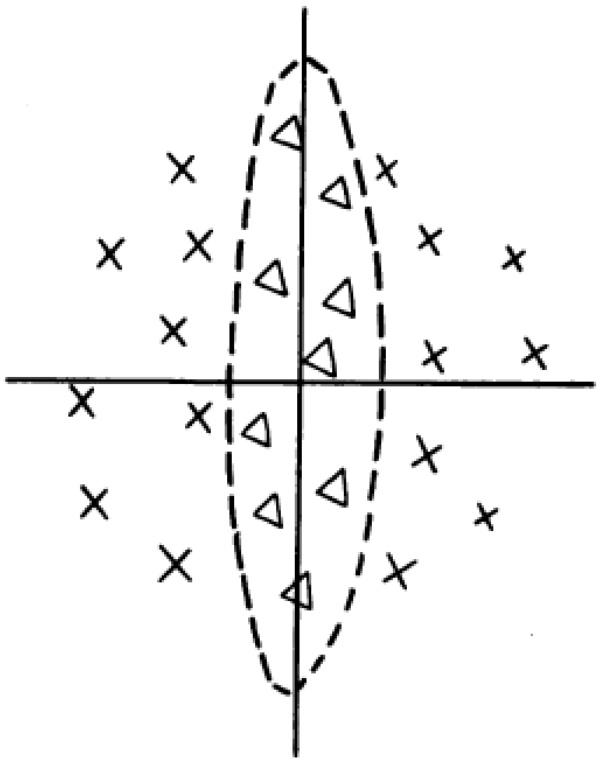


From M. Lewicky

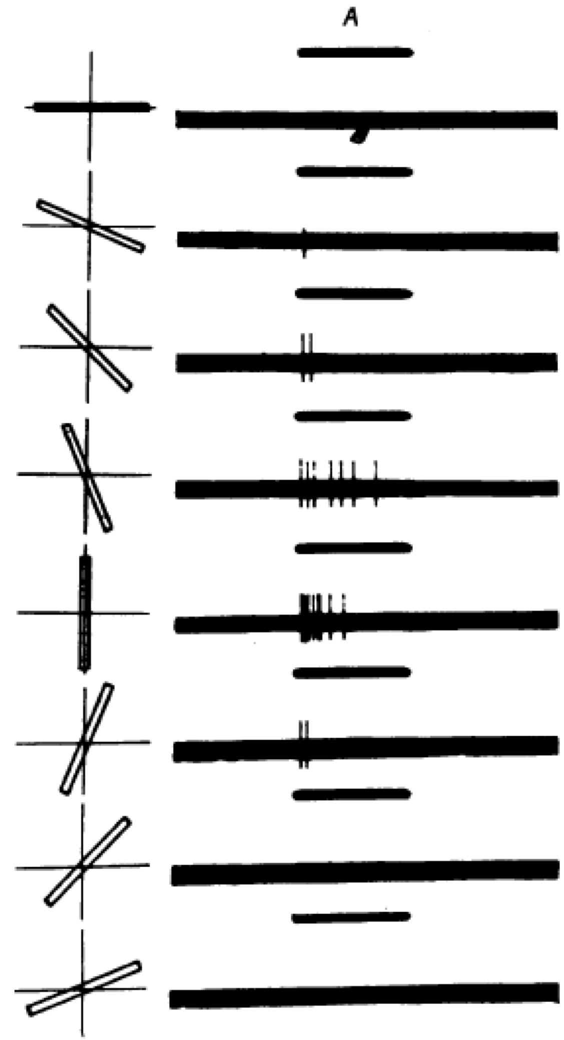
J. Physiol. (1959) 148, 574–591

RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

By D. H. HUBEL* and T. N. WIESEL*
From the Wilmer Institute, The Johns Hopkins Hospital and University, Baltimore, Maryland, U.S.A.



Receptive field
of a cell in the cat's cortex

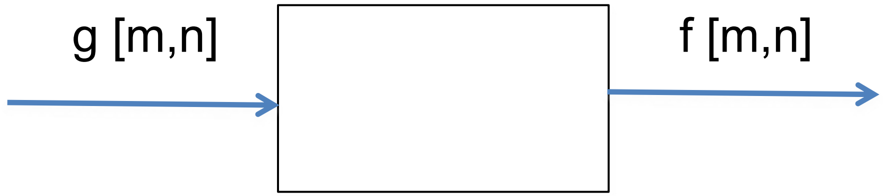


Responses to an oriented bar

Outline

- Linear filtering
- Fourier Transform

Filtering



We want to remove unwanted sources of variation, and keep the information relevant for whatever task we need to solve

Linear filtering

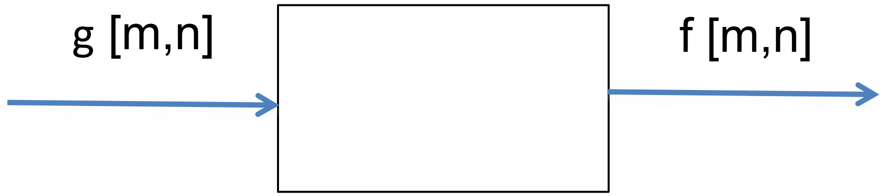
For a linear system, each output is a linear combination of all the input values:

$$f[m,n] = \sum_{k,l} h[m,n,k,l]\, g[k,l]$$

In matrix form:

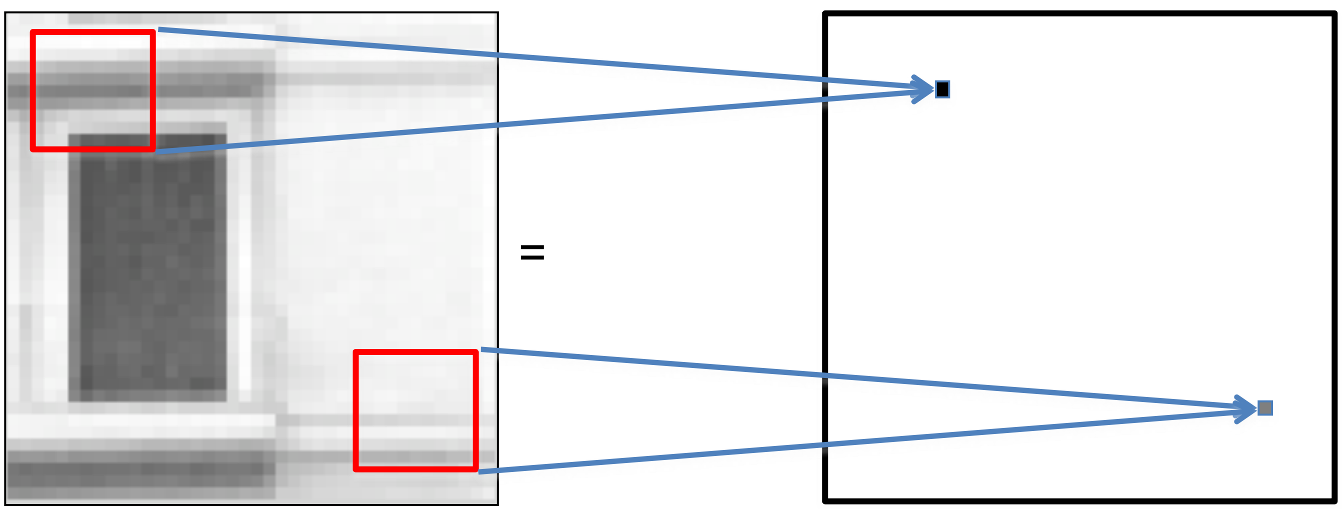
f = H g

Linear filtering



In vision, many times, we are interested in operations that are spatially invariant.
For a linear spatially invariant system:

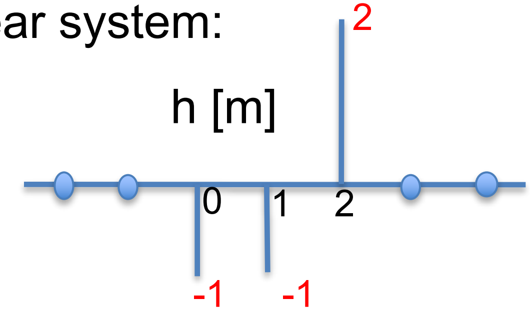
$$f[m,n] \quad h \quad g \quad \underset{k,l}{} \quad h[m \quad k,n \quad l]g[k,l]$$



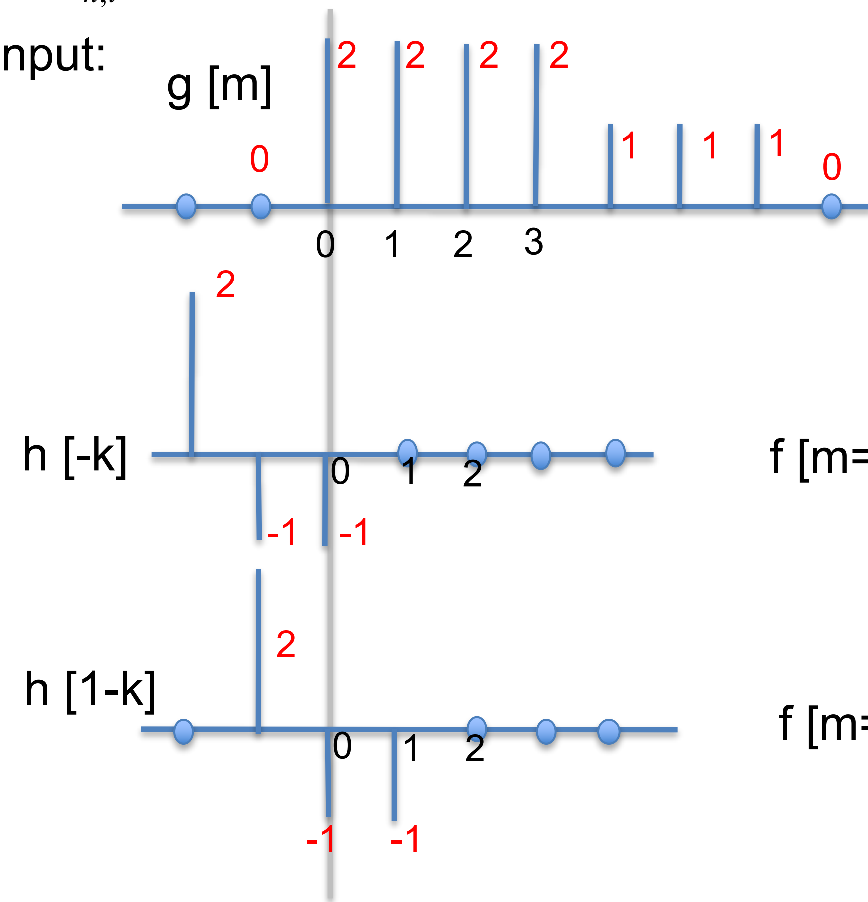
Linear filtering

$$f[m,n] \quad h \quad g \quad \sum_{k,l} h[m \quad k,n \quad l]g[k,l]$$

Linear system:



Input:



Output?

$$f[m \quad 0] \quad \sum_{k} h[\quad k]g[k]$$

f [m=0]=-2

$$f[m \quad 1] \quad \sum_{k} h[1 \quad k]g[k]$$

f [m=1]=-4

$$f[m \quad 2] \quad \sum_{k} h[2 \quad k]g[k]$$

f [m=2]=0

Linear filtering

For a linear spatially invariant system

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$

m=0 1 2 …

111	115	113	111	112	111	112	111
135	138	137	139	145	146	149	147
163	168	188	196	206	202	206	207
180	184	206	219	202	200	195	193
189	193	214	216	104	79	83	77
191	201	217	220	103	59	60	68
195	205	216	222	113	68	69	83
199	203	223	228	108	68	71	77

g[m,n]

$\otimes$

-1	2	-1
-1	2	-1
-1	2	-1

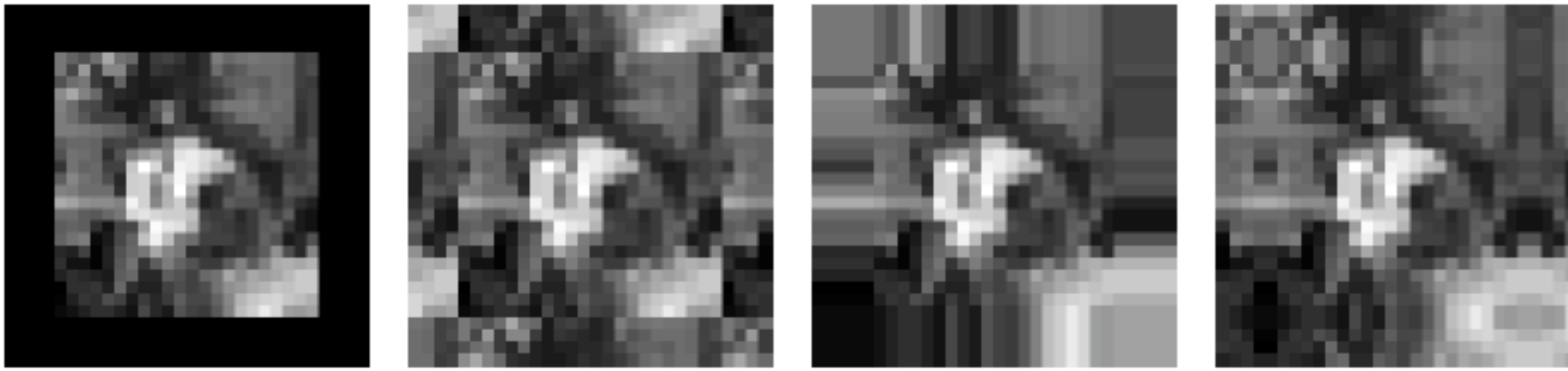
h[m,n]

=

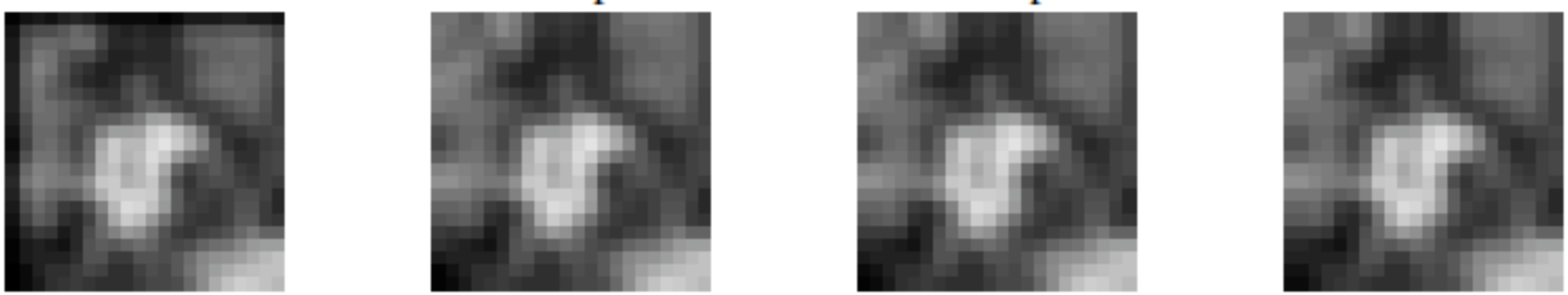
?	?	?	?	?	?	?	?
?	-5	9	-9	21	-12	10	?
?	-29	18	24	4	-7	5	?
?	-50	40	142	-88	-34	10	?
?	-41	41	264	-175	-71	0	?
?	-24	37	349	-224	-120	-10	?
?	-23	33	360	-217	-134	-23	?
?	?	?	?	?	?	?	?

f[m,n]

Borders



zero wrap clamp mirror

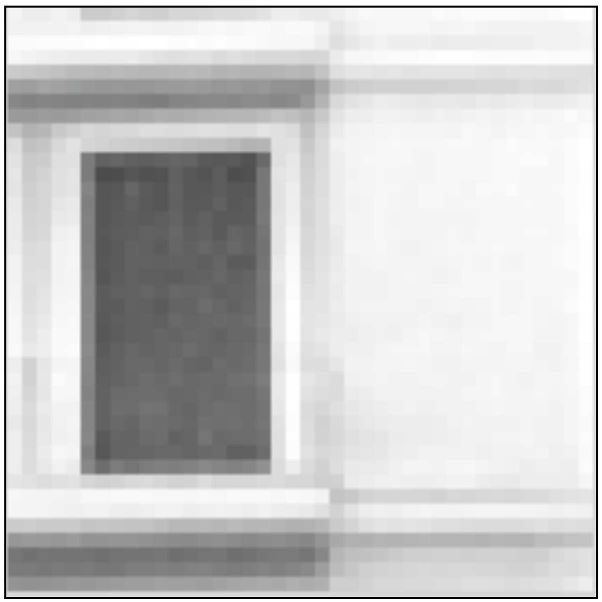


blurred: zero normalized zero clamp mirror

From Szeliski, Computer Vision, 2010

Impulse

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k,n-l]g[k,l]$$



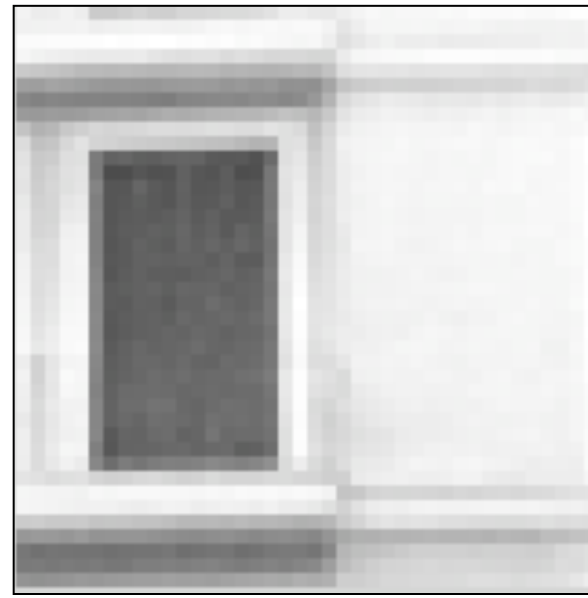
g[m,n]

$\otimes$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

h[m,n]

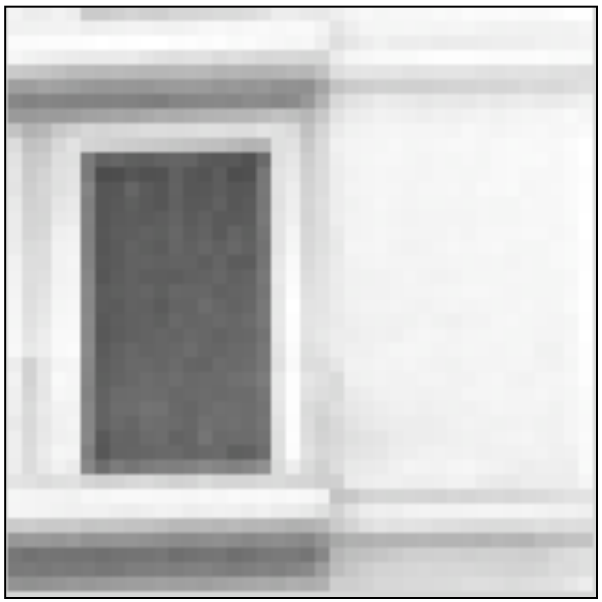
=



f[m,n]

Shifts

$$f[m,n] \quad I \quad g \quad \underset{k,l}{} \quad h[m \quad k,n \quad l]g[k,l]$$



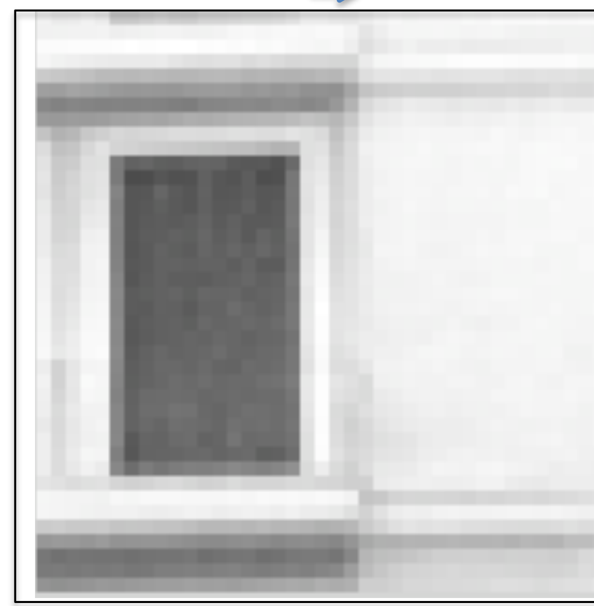
g[m,n]

0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0

h[m,n]

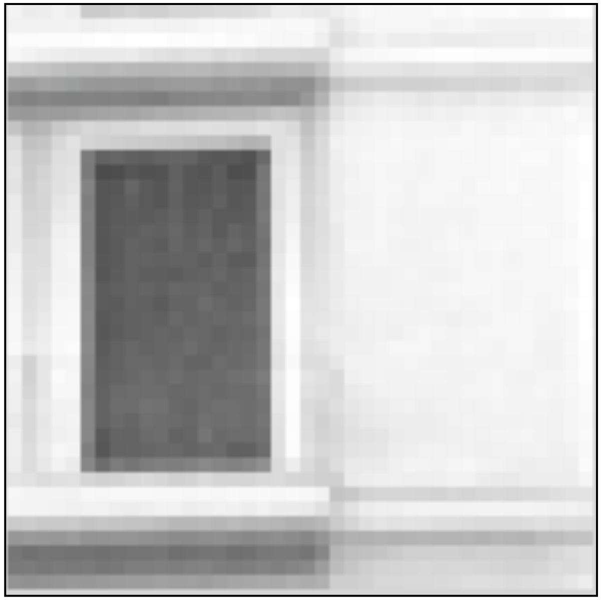
=

2pixels



f[m,n]

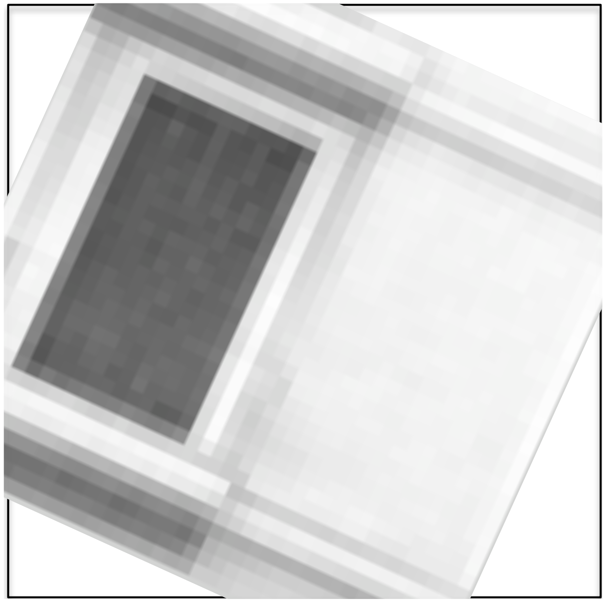
Image rotation



g[m,n]

? =

h[m,n]



f[m,n]

It is linear, but not a spatially invariant operation. There is not convolution.

Rectangular filter

$\otimes$

h[m,n]

=

f[m,n]

g[m,n]

Rectangular filter

g[m,n]

$\otimes$

h[m,n]

=

f[m,n]

Rectangular filter

$g[m,n]$

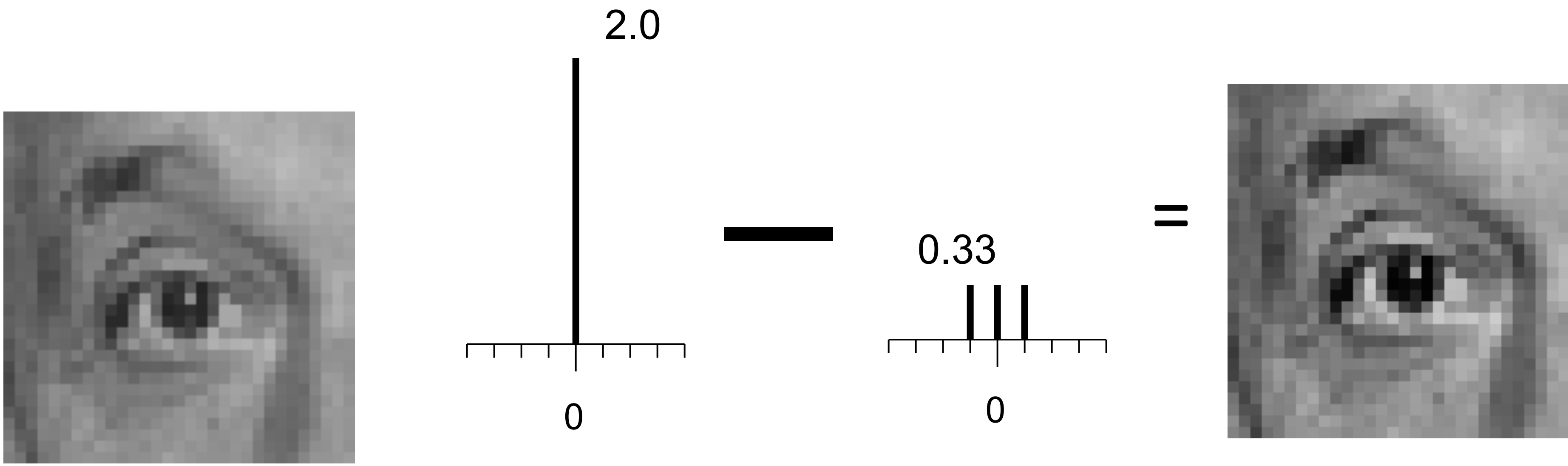
$\otimes$

$h[m,n]$

$=$

$f[m,n]$

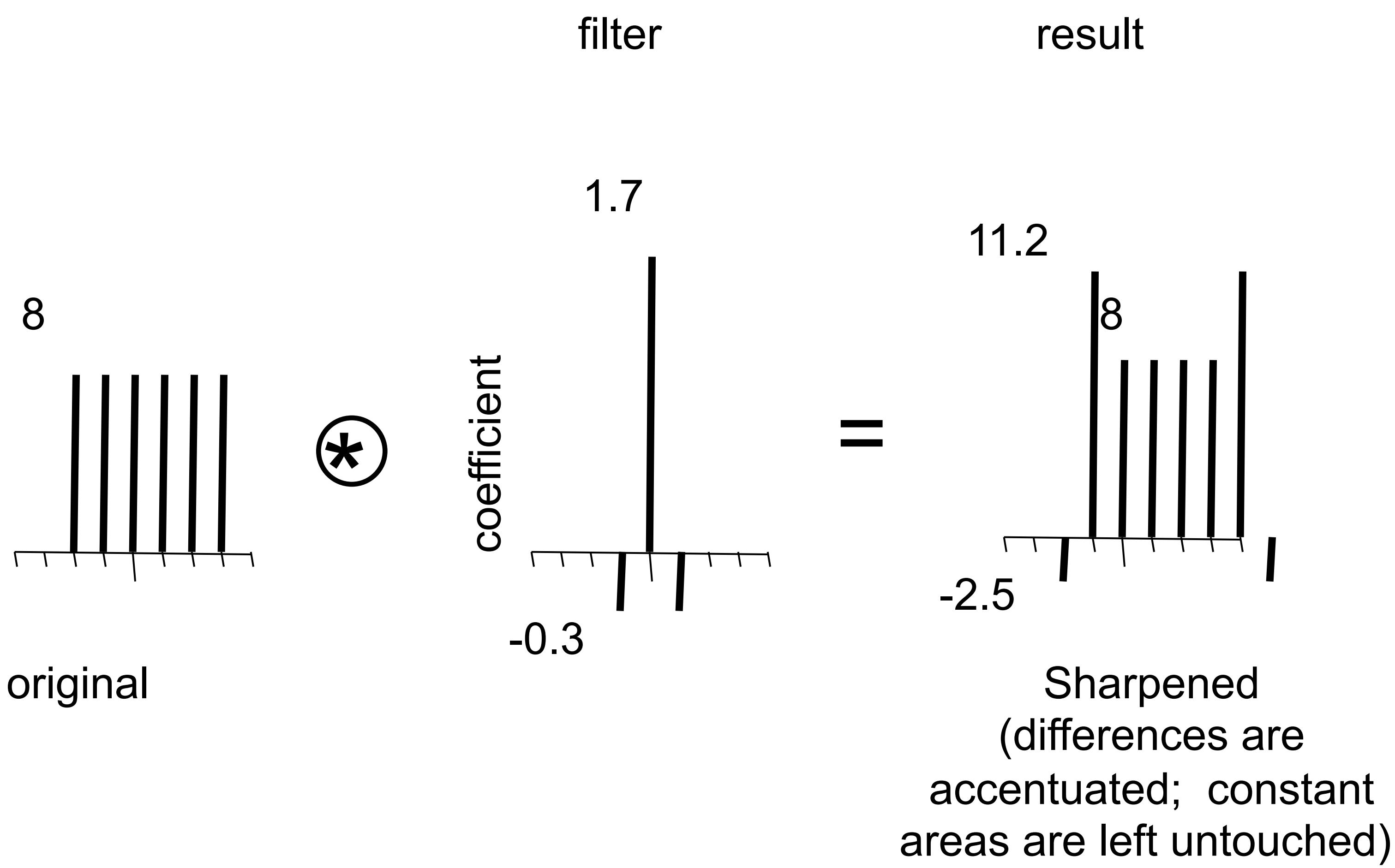
Sharpening



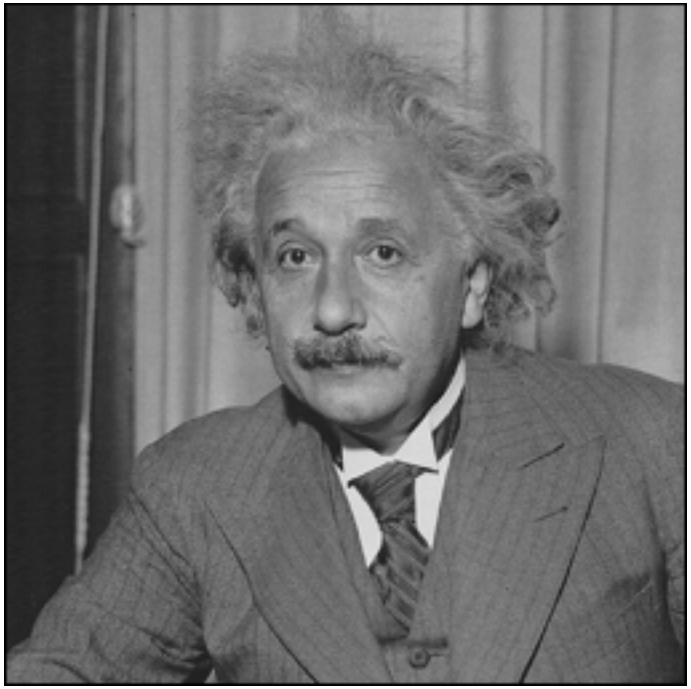
original

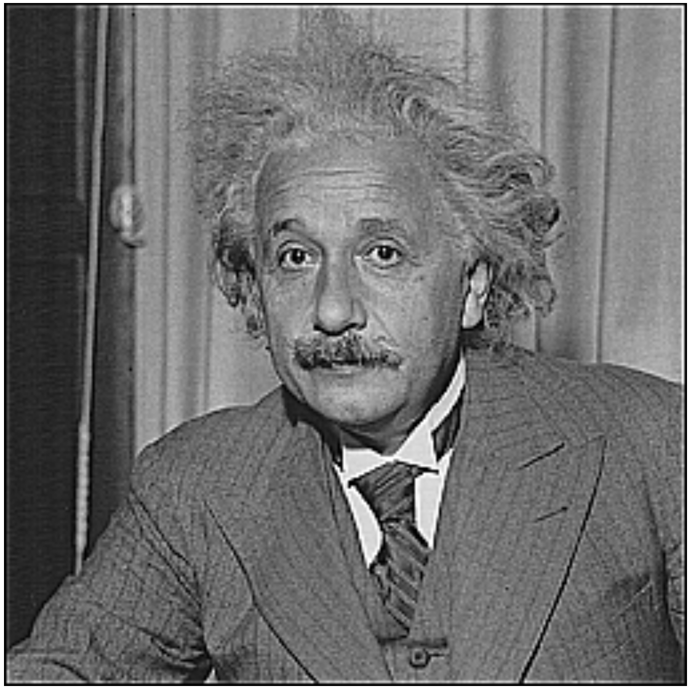
Sharpened
original

Sharpening example



Sharpening





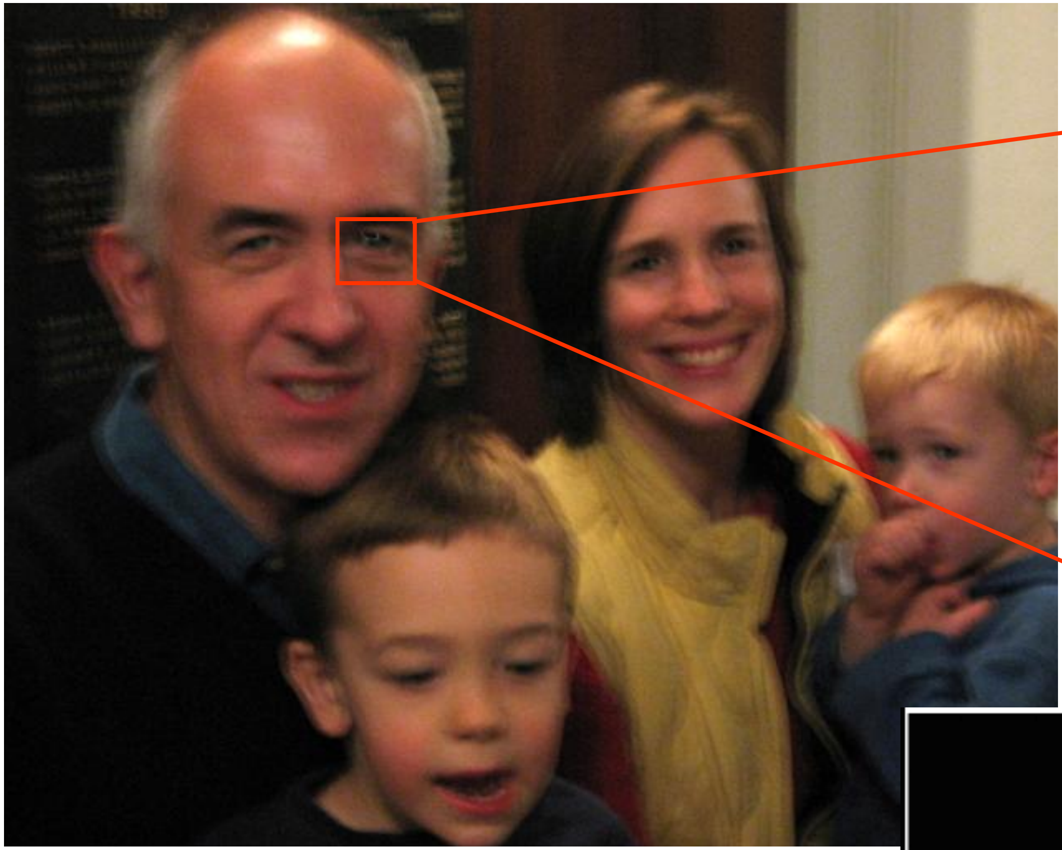
before **after**

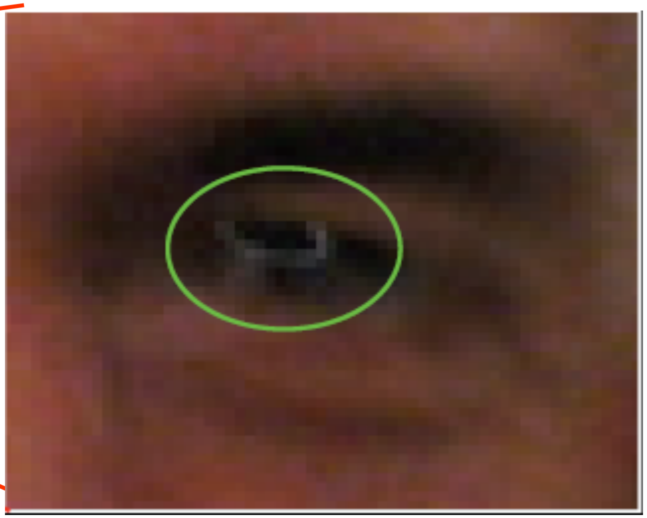
A taxonomy of useful filters

- Impulse, Shifts,
- Blur
  - Rectangular blur (see artifacts)
  - Gaussian
  - Bilateral exponential
  - Asymmetrical filter: motion blur
- Edges
  - [-1 1]
  - Derivative filter
  - Derivative of a gaussian
  - Oriented filters
  - Gabor filter
  - Quadrature filters: phase and magnitude.
  - Elongated edges: filling gaps…

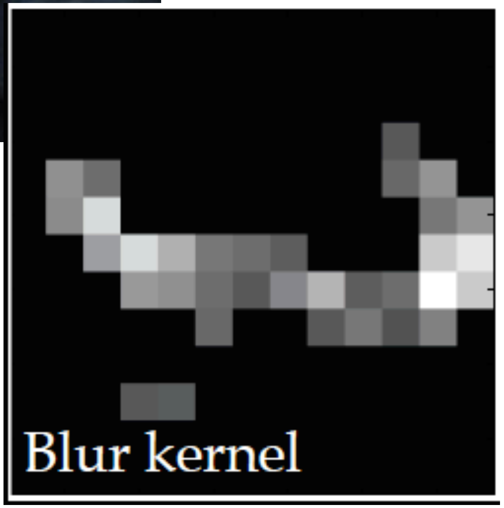
BLUR

Linear blur occurs under many natural situations





(from Fergus et al, 2007)

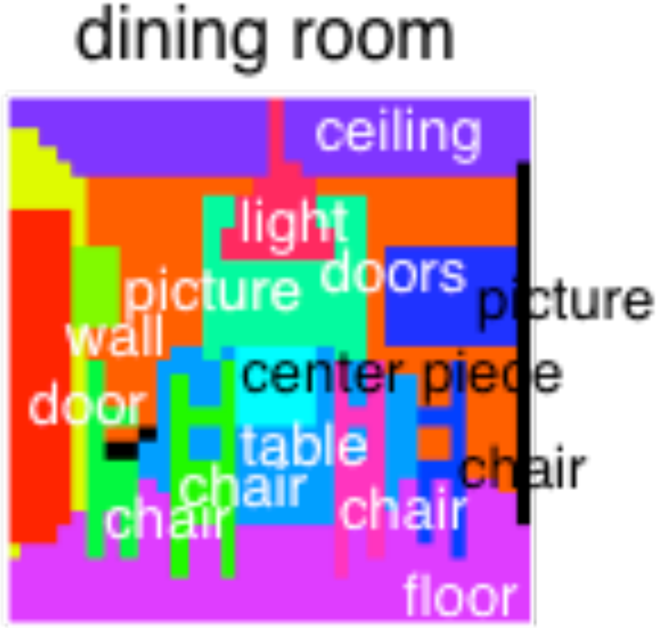


This is not a Gaussian kernel...

Linear blur occurs under many natural situations

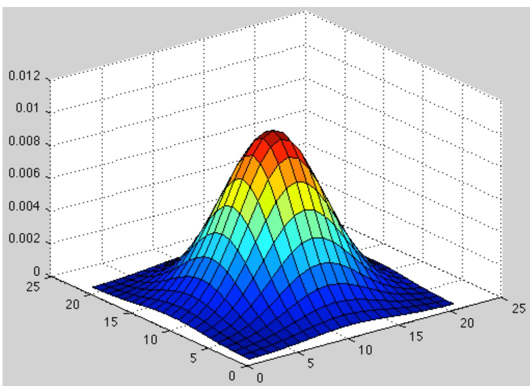
Linear blur occurs under many natural situations

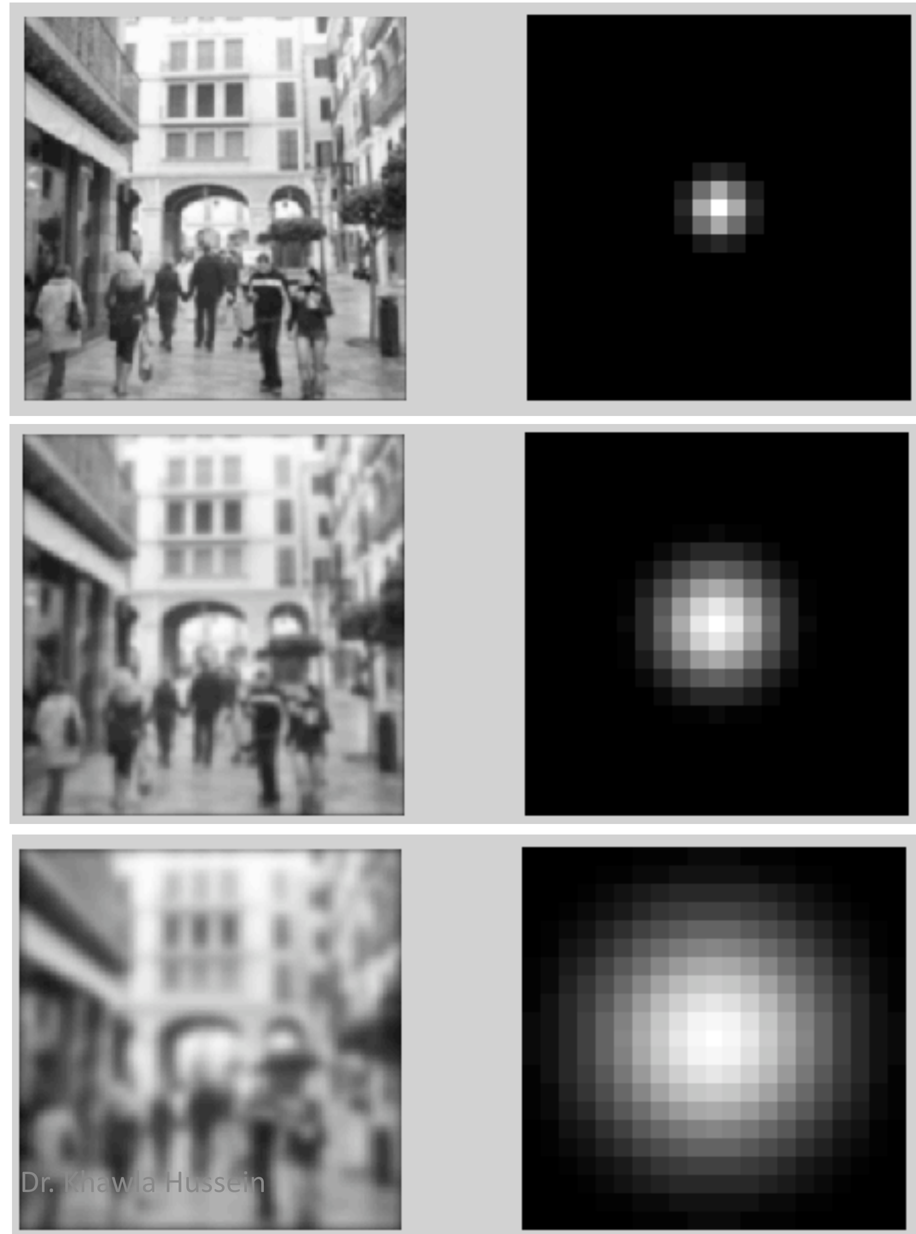
Linear blur occurs under many natural situations



Gaussian filter

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



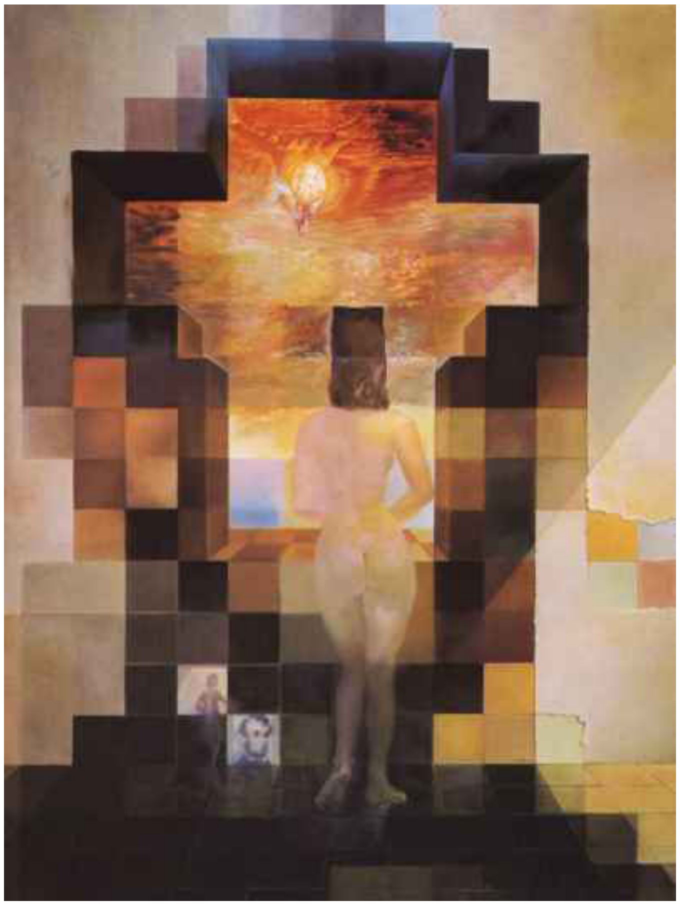


σ=1

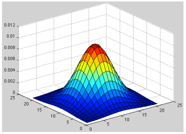
σ=2

σ=4

Gaussian filter



Dali



Some desirable properties for a blur kernel

- Positivity: $h(m) >= 0$
- Symmetry: $h(m) = h(-m)$
- Unimodality: $h(m) >= h(m+1)$ for $m >= 0$
- Normalized: $\Sigma\, h(m) = 1$
- Equal contribution: $\Sigma\, h(2m) = \Sigma\, h(2m+1)$

Some kernels that verify this are:

$$[\tfrac{1}{2}\ \tfrac{1}{2}]$$

$$[\tfrac{1}{4}\ \tfrac{1}{2}\ \tfrac{1}{4}]$$


DERIVATIVES


DERIVATIVES

[-1 1]

$$\frac{\partial \mathbf{I}}{\partial x} \simeq \mathbf{I}(x, y) - \mathbf{I}(x-1, y)$$

[-1, 1]

h[m,n]

=

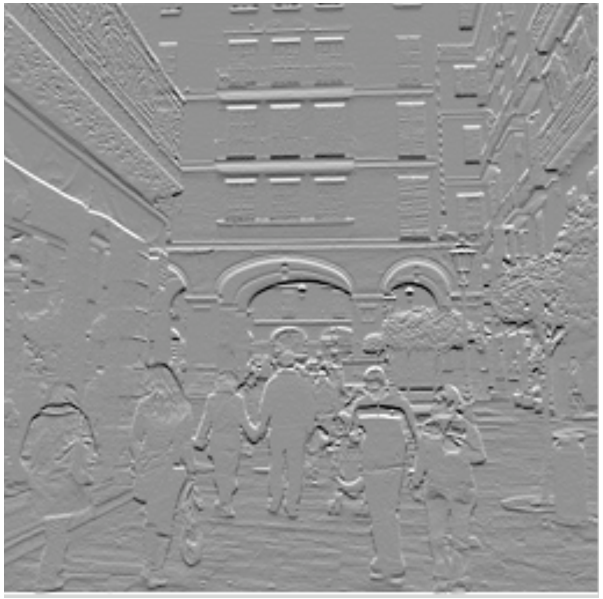
g[m,n]

f[m,n]

$[-1\ 1]^T$

$[-1, 1]^T$ =

h[m,n]

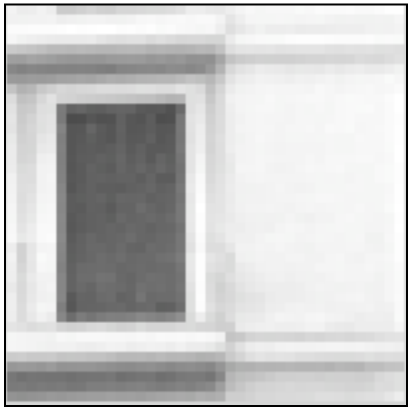


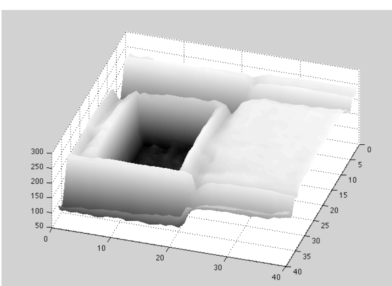
g[m,n]

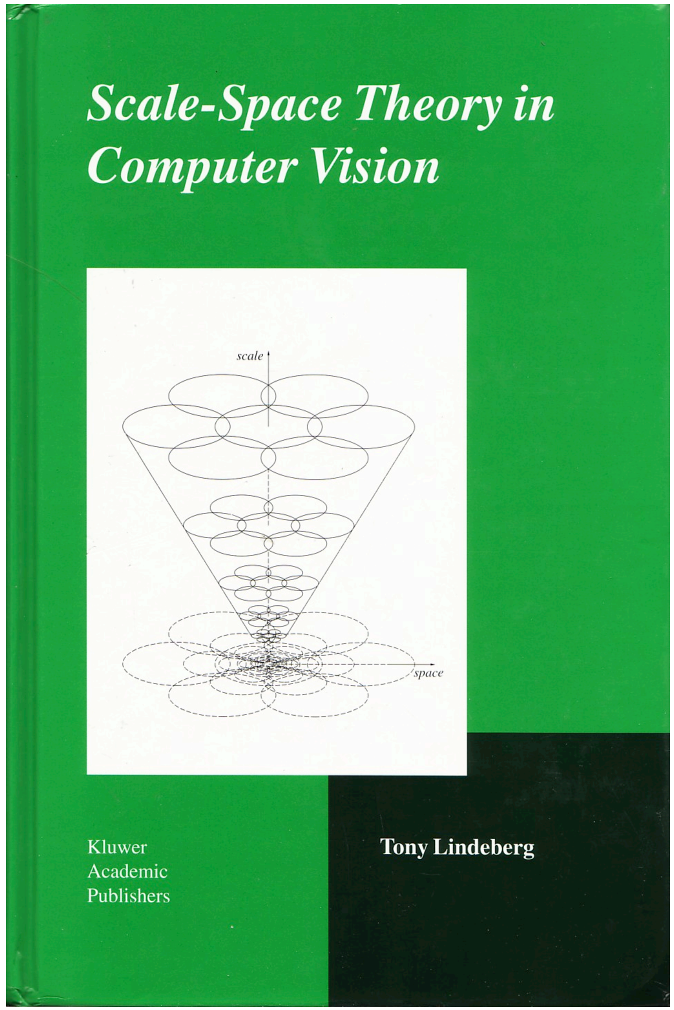
f[m,n]

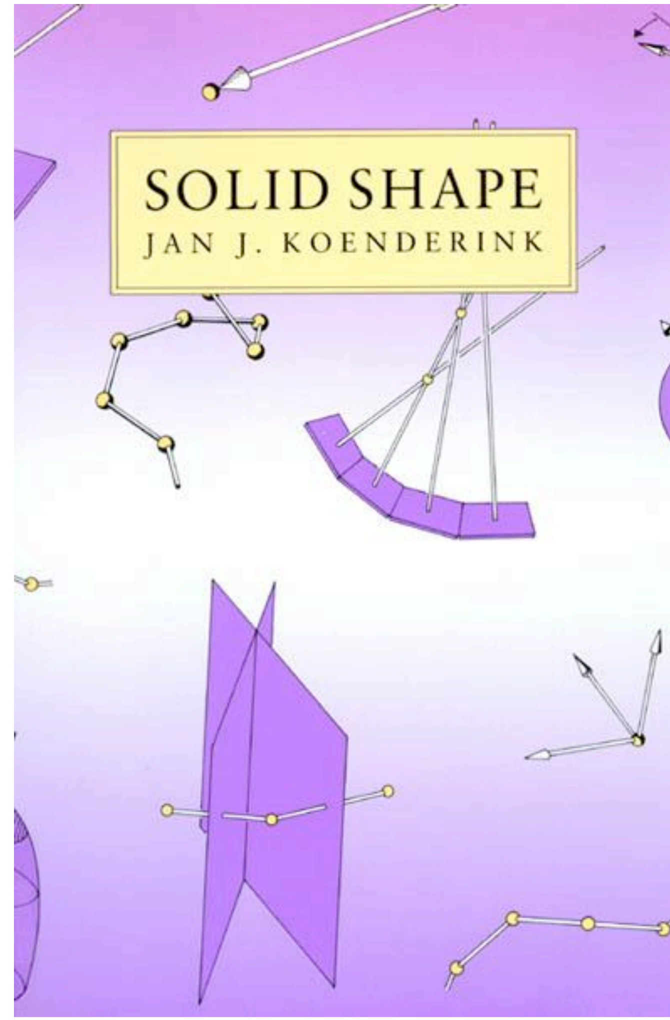
Differential Geometry Descriptors

I(x,y)









Finding edges in the image

Image gradient:

$$\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y} \right)$$

Approximation image derivative:

$$\frac{\partial \mathbf{I}}{\partial x} \simeq \mathbf{I}(x, y) - \mathbf{I}(x-1, y)$$

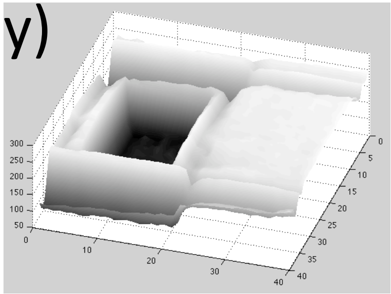
Edge strength $E(x, y) = |\nabla \mathbf{I}(x, y)|$

Edge orientation: $\theta(x, y) = \angle \nabla \mathbf{I} = \arctan \frac{\partial \mathbf{I}/\partial y}{\partial \mathbf{I}/\partial x}$

Edge normal: $\mathbf{n} = \frac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|}$

Differential Geometry Descriptors

I(x,y)



If we think of the image as a continuous function

Image gradient:

$$I \qquad \frac{I(x,y)}{x}, \frac{I(x,y)}{y}$$

Directional gradient:

|u|=1

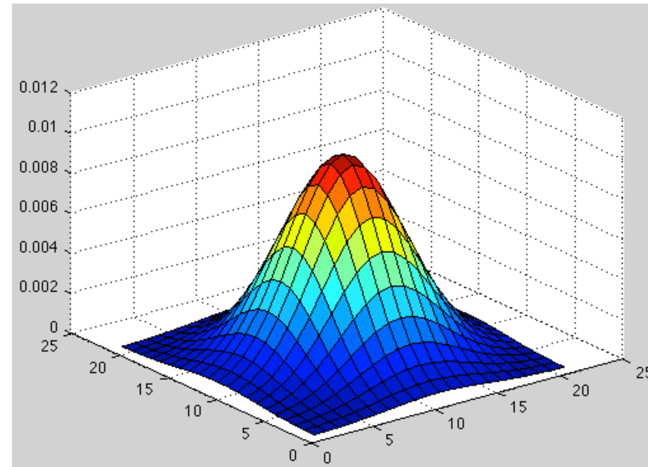
$$u^T \quad I \quad \cos \qquad \frac{I(x,y)}{x} \quad \sin \qquad \frac{I(x,y)}{y}$$

Laplacian:

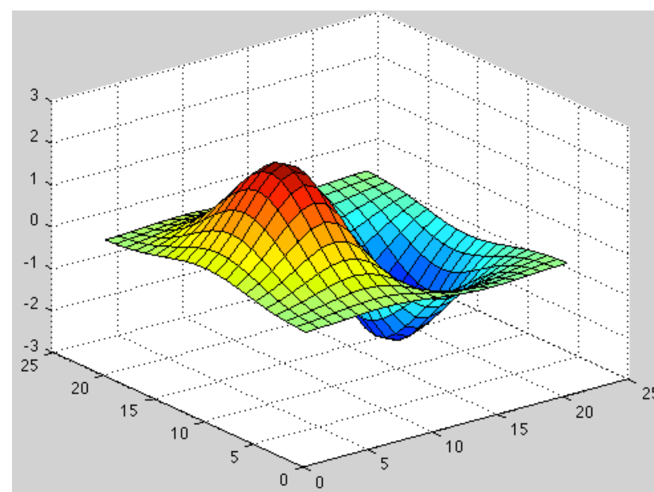
$$^2 I \qquad \frac{^2 I(x,y)}{x^2} \qquad \frac{^2 I(x,y)}{y^2}$$

Gaussian derivative

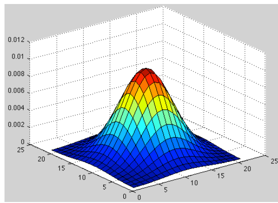
$$g(x,y) \quad \frac{1}{2 \quad ^{2}} e^{\frac{x^{2} \quad y^{2}}{2 \quad ^{2}}}$$



$$\frac{g(x,y)}{x} \quad \frac{x}{2 \quad ^{4}} e^{\frac{x^{2} \quad y^{2}}{2 \quad ^{2}}}$$



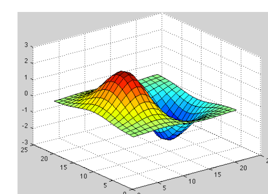
$$g(x,y) \quad \frac{1}{2 \quad ^{2}} e^{\frac{x^{2} \quad y^{2}}{2 \quad ^{2}}}$$



$$\frac{I(x,y)}{x} \quad g(x,y)$$

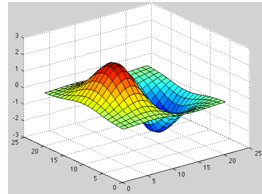
$$\frac{I(x,y) \quad g(x,y)}{x}$$

$$I(x,y) \quad \frac{g(x,y)}{x}$$

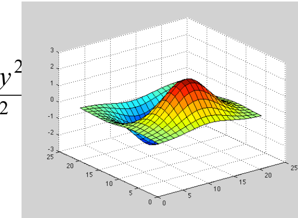


$$\frac{g(x,y)}{x} \quad \frac{x}{2 \quad ^{4}} e^{\frac{x^{2} \quad y^{2}}{2 \quad ^{2}}}$$

$$g_x(x,y) \quad \frac{g(x,y)}{x} \quad \frac{x}{2 \quad {}^4} e^{\frac{x^2 \quad y^2}{2 \quad {}^2}}$$



$$g_y(x,y) \quad \frac{g(x,y)}{x} \quad \frac{x}{2 \quad {}^4} e^{\frac{x^2 \quad y^2}{2 \quad {}^2}}$$

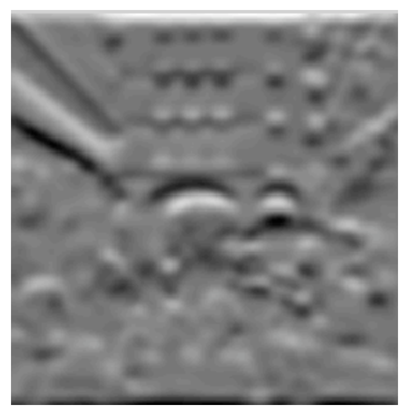
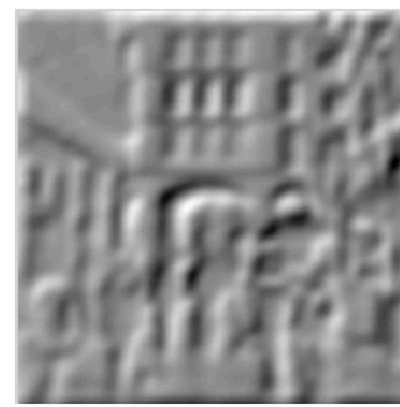


The smoothed directional gradient is a linear combination of two kernels

$$u^T \quad g \quad I \quad \cos \quad g_x(x,y) \quad \sin \quad g_y(x,y) \quad I(x,y)$$

Any orientation can be computed as a linear combination of two filtered images

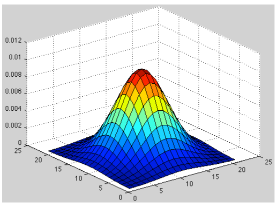
$$\cos \quad g_x(x,y) \quad I(x,y) \quad \sin \quad g_y(x,y) \quad I(x,y)$$

= cos(α)  +sin(α)  = 

Steereability of gaussian derivatives, Freeman & Adelson 92

Laplacian

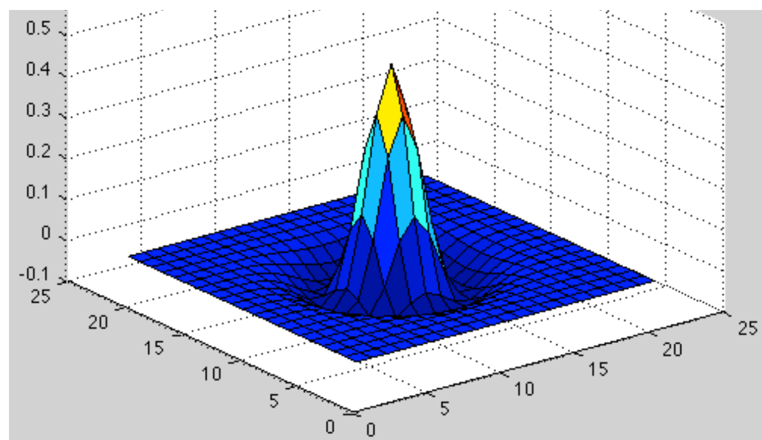
$$g(x,y) \quad \frac{1}{2 \quad ^2} e^{\frac{x^2 \quad y^2}{2 \quad ^2}}$$



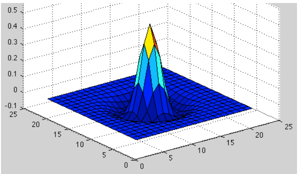
$$^2 I \quad g \qquad \frac{^2 I(x,y)}{x^2} \quad \frac{^2 I(x,y)}{y^2} \qquad g(x,y)$$

$$^2 I \quad g \quad I \qquad ^2 g$$

$$^2 g(x,y) \qquad \frac{x^2 \quad y^2}{4} \quad \frac{2}{\quad 2} \; g(x,y)$$



Laplacian



Outline

- Linear filtering
- Fourier Transform

Linear image transformations

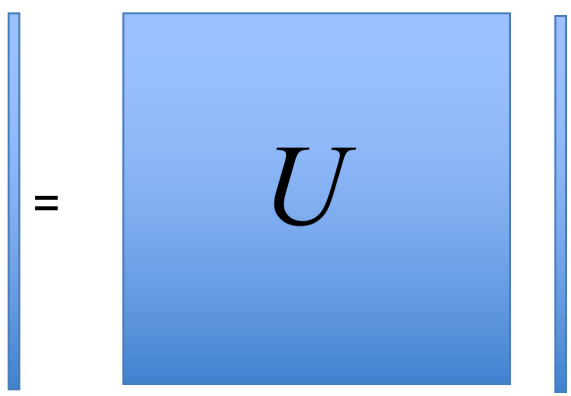
- In analyzing images, it's often useful to make a change of basis.

Transformed image

$$\vec{F} = U\vec{f}$$

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



Self-inverting transforms

$$\vec{F} = U\vec{f} \longleftrightarrow \vec{f} = U^{-1}\vec{F}$$

Same basis functions are used for the inverse transform

$$\vec{f} = U^{-1}\vec{F}$$

$$= U^{+}\vec{F}$$

U transpose and complex conjugate

An example of such a transform: the Discrete Fourier transform

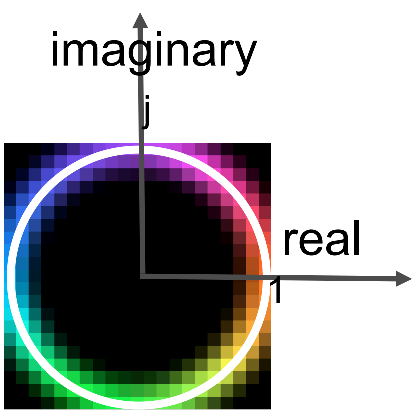
Forward transform

$$F[m,n] \quad \mathop{}_{k\ \ 0\ l\ \ 0}^{M\ \ 1N\ \ 1} \quad f[k,l]e^{\,i\ \frac{km}{M}\ \ \frac{\ln}{N}}$$

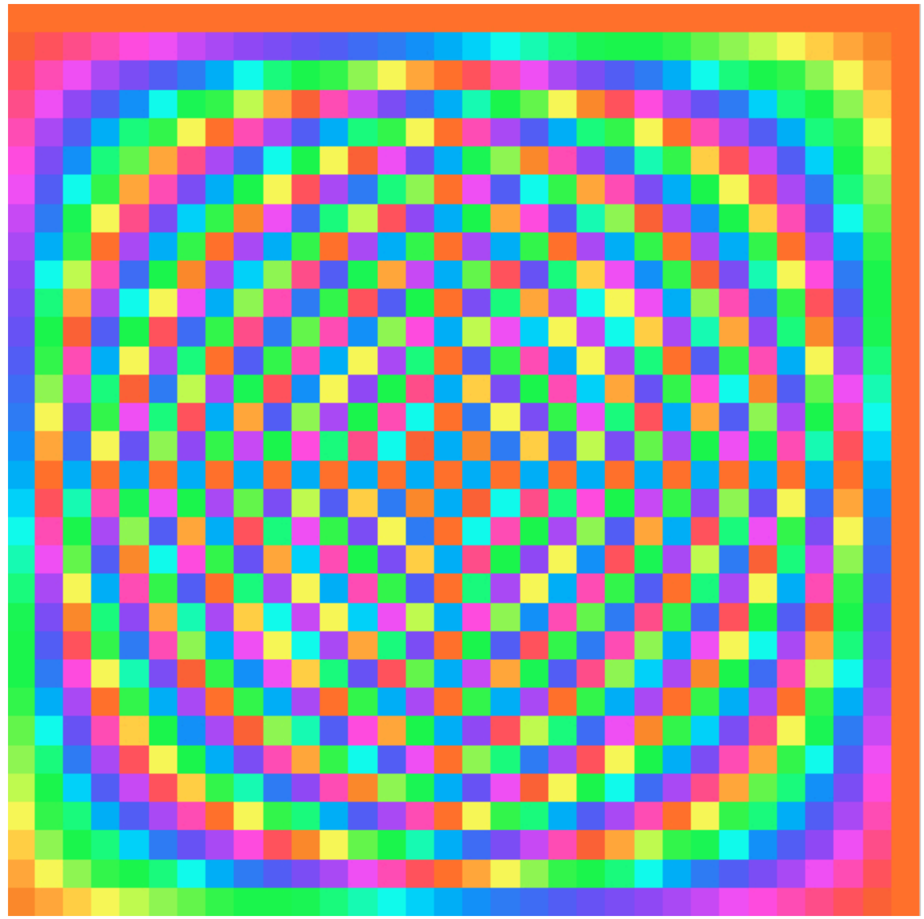
Inverse transform

$$f[k,l] \quad \frac{1}{MN} \mathop{}_{k\ \ 0\ l\ \ 0}^{M\ \ 1N\ \ 1} F[m,n]e^{\,i\ \frac{km}{M}\ \ \frac{\ln}{N}}$$

Fourier transform visualization



color key



Fourier transform matrix

input signal

$$F[m,n] = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} f[k,l]e^{-\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

Why is the Fourier domain particularly useful?

- Linear, space invariant operations are just **diagonal operations** in the frequency domain.

- Ie, linear convolution is multiplication in the frequency domain.

Fourier transform of convolution

Consider a (circular) convolution of g and h

$$f = g \otimes h$$

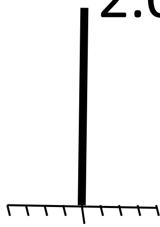
In the transform domain, this just modulates the transform amplitudes

$$F[m,n] = DFT\left(g \otimes h\right)$$
$$= G\left[m,n\right]H\left[m,n\right]$$

Analysis of a simple sharpening filter

original

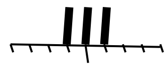
2.0



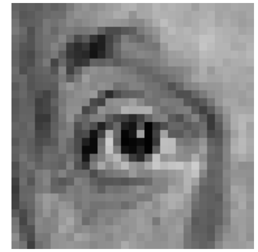
0

−

0.33



0



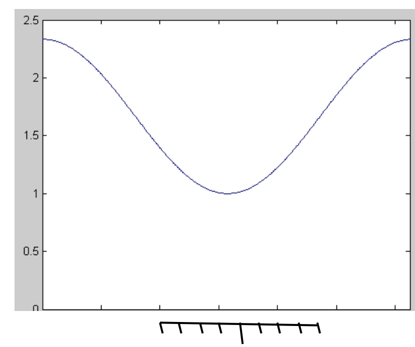
sharpened

$$F[m] = \sum_{k=0}^{M-1} f[k] e^{-\pi i \left(\frac{km}{M} \right)}$$

$$= 2 - \frac{1}{3}\left(1 + 2\cos\left(\frac{\pi m}{M} \right) \right)$$

2.3

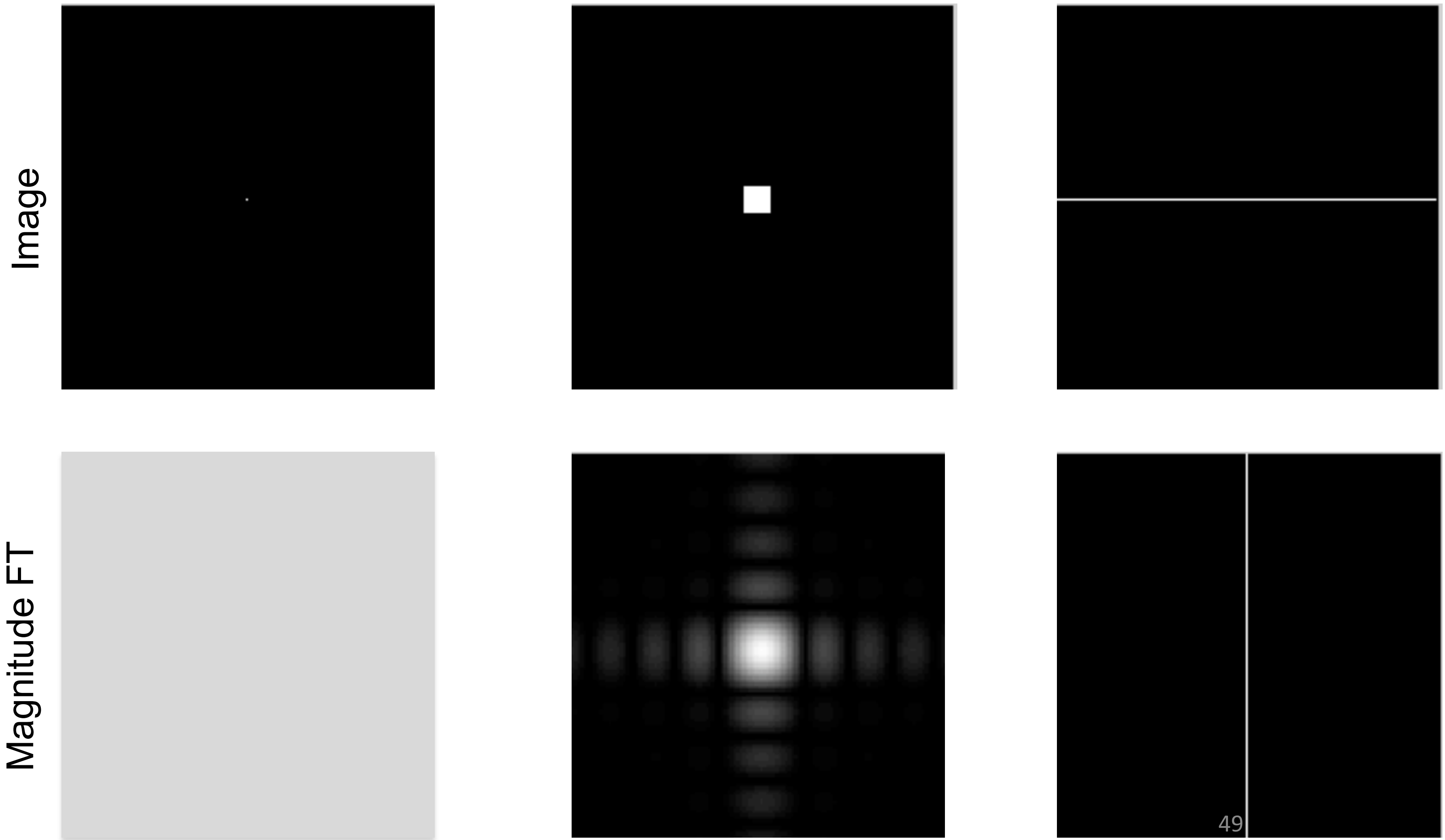
1.0



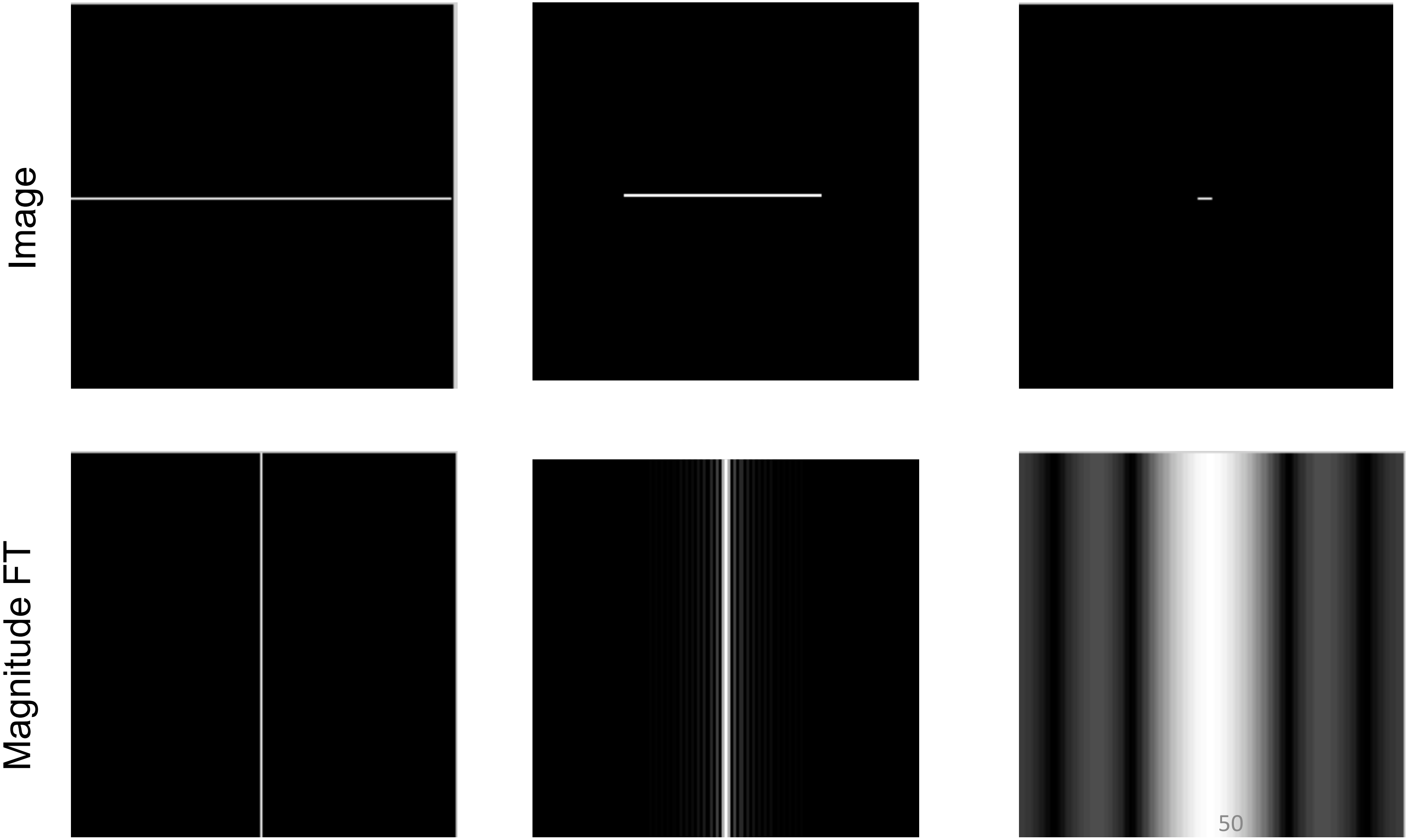
0

high-pass filter

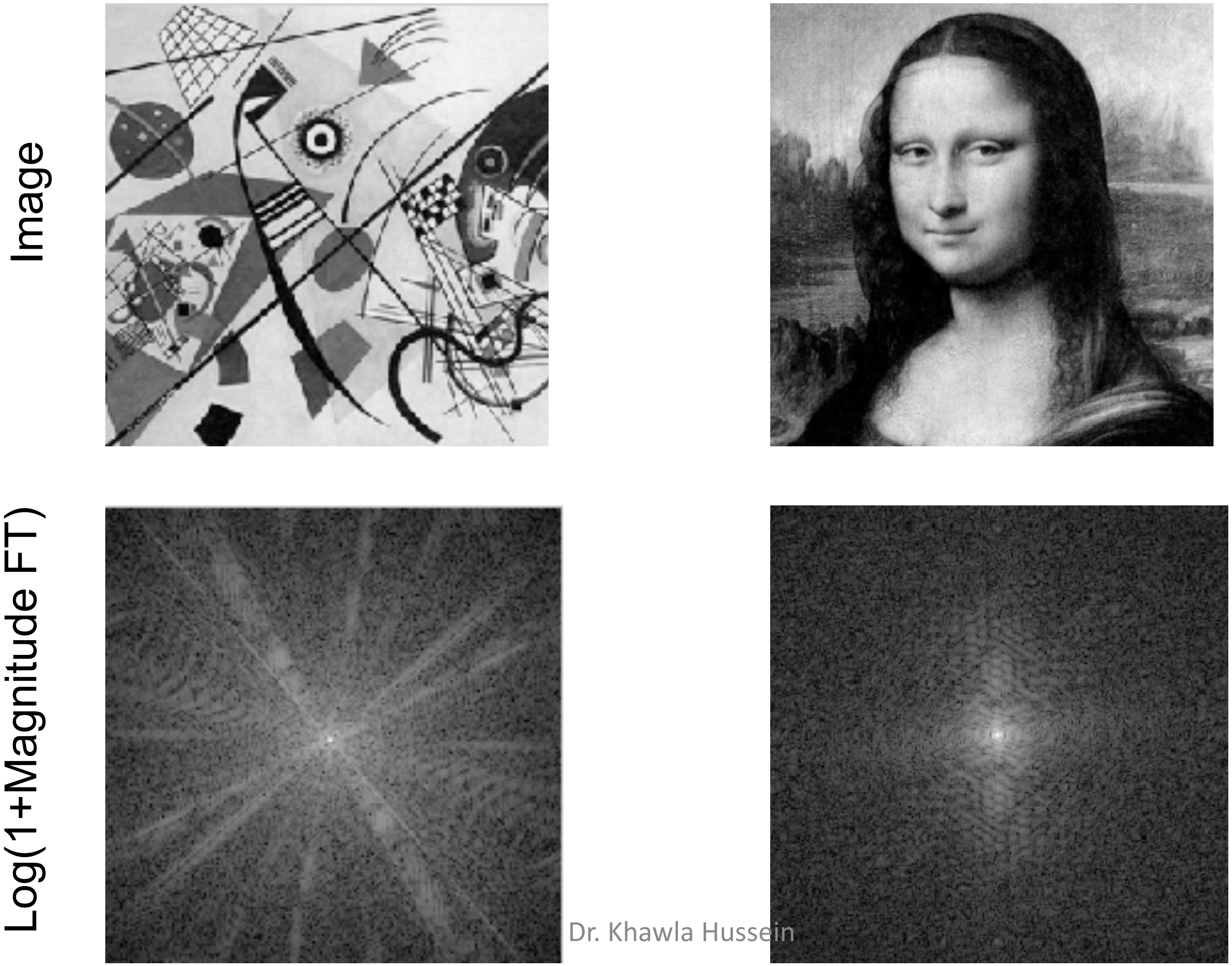
Some important Fourier Transforms



Some important Fourier Transforms



The Fourier Transform of some important images

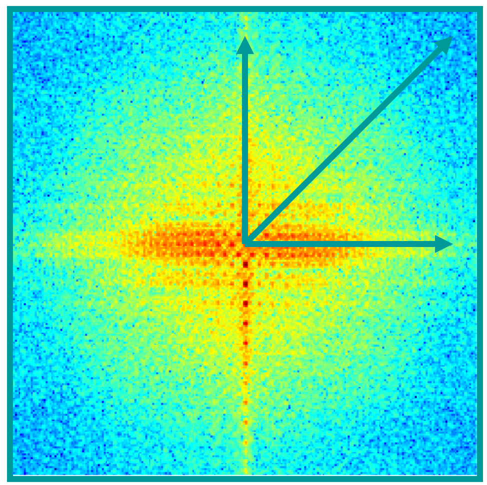


How to interpret a Fourier Spectrum

Vertical orientation

45 deg.

0

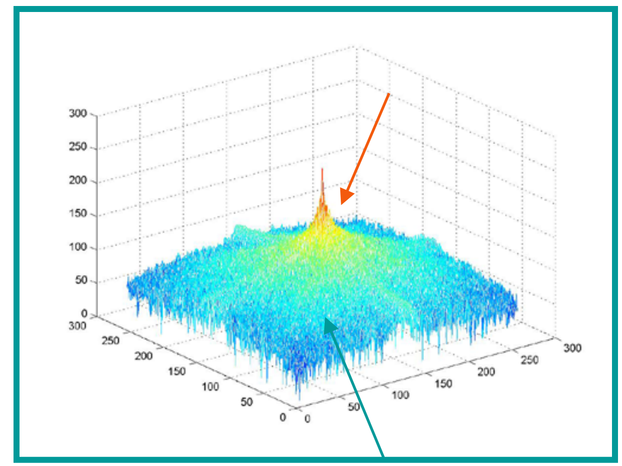


Horizontal orientation

0 f_{max}

fx in cycles/image

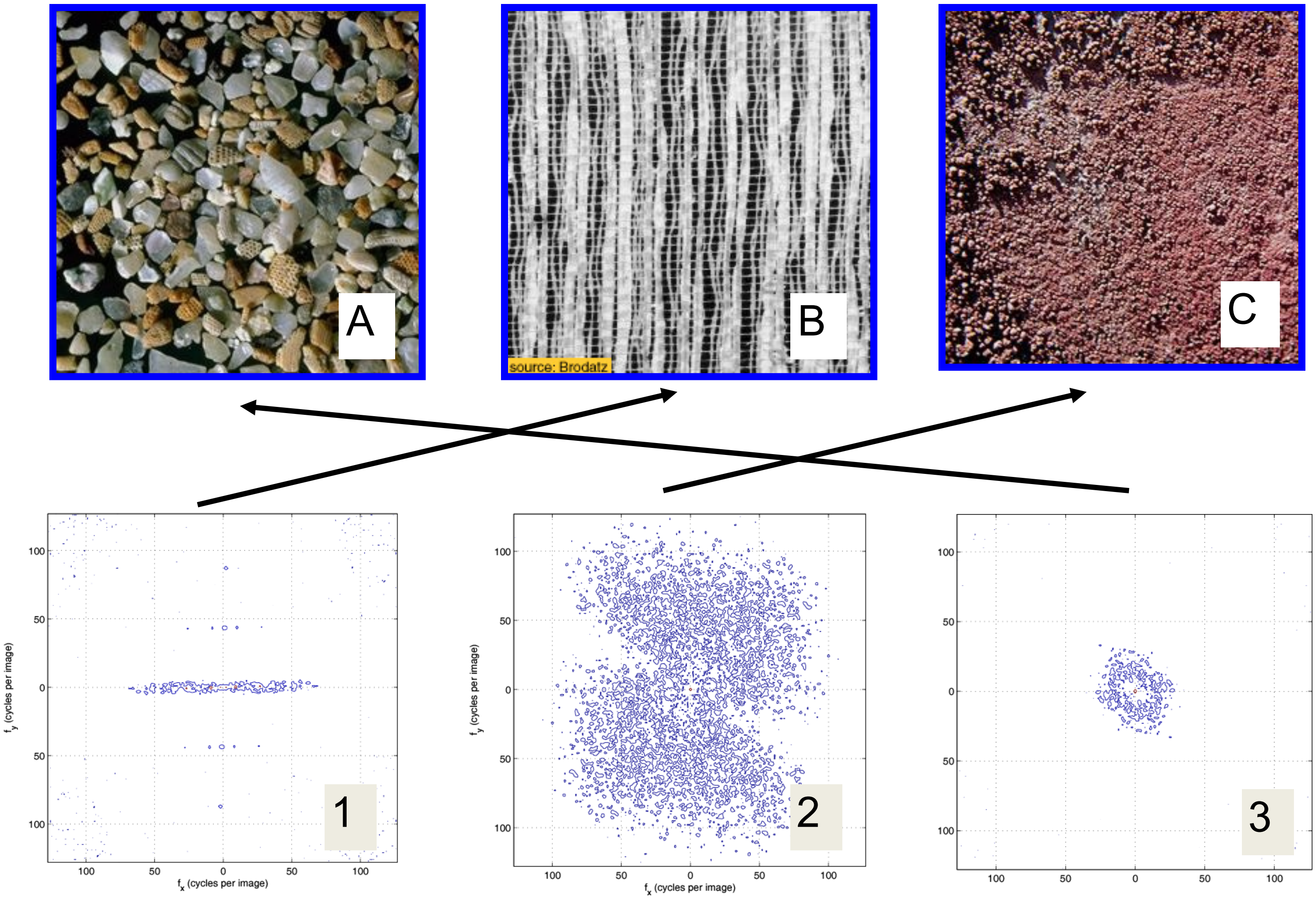
Low spatial frequencies



High spatial frequencies

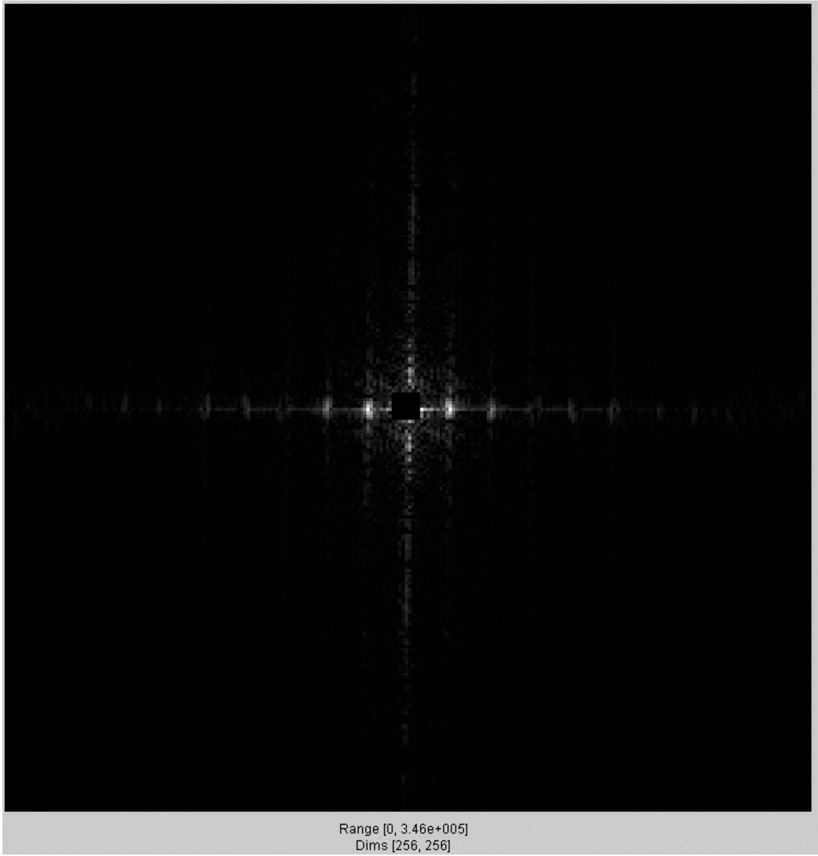
Log power spectrum

Fourier Amplitude Spectrum

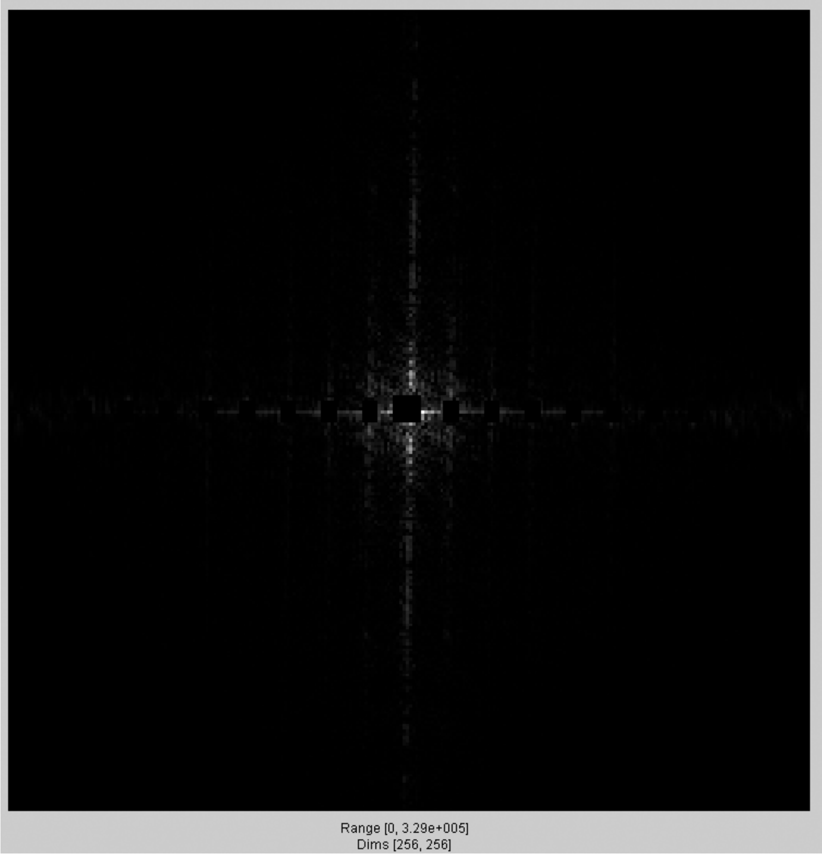


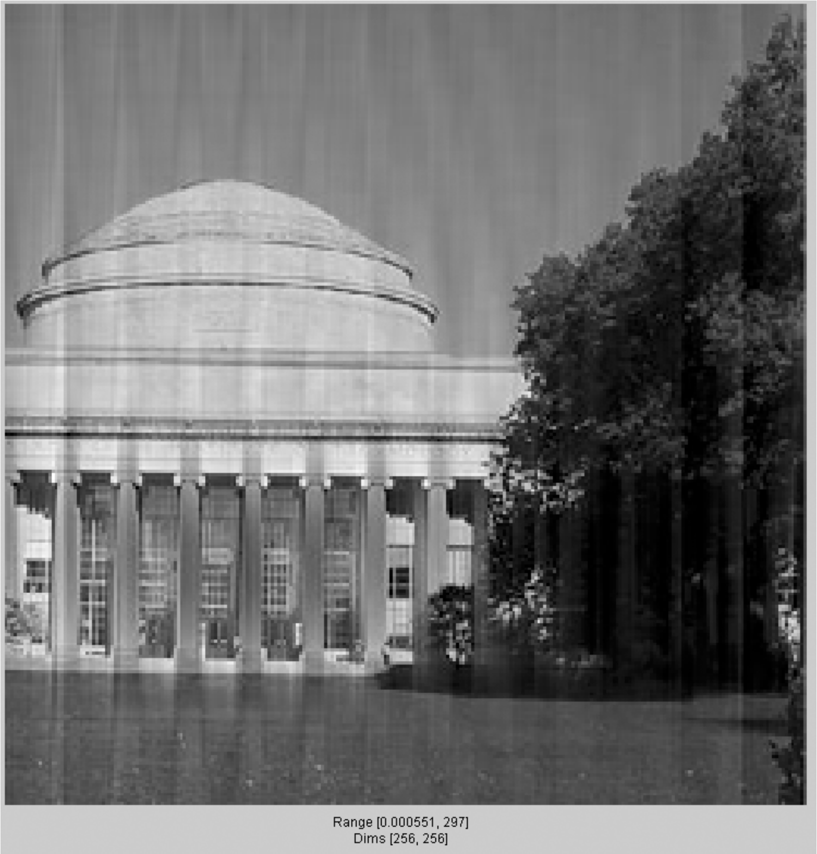
fx(cycles/image pixel size) fx(cycles/image pixel size) fx(cycles/image pixel size)

Fourier transform magnitude



Masking out the fundamental and harmonics from periodic pillars

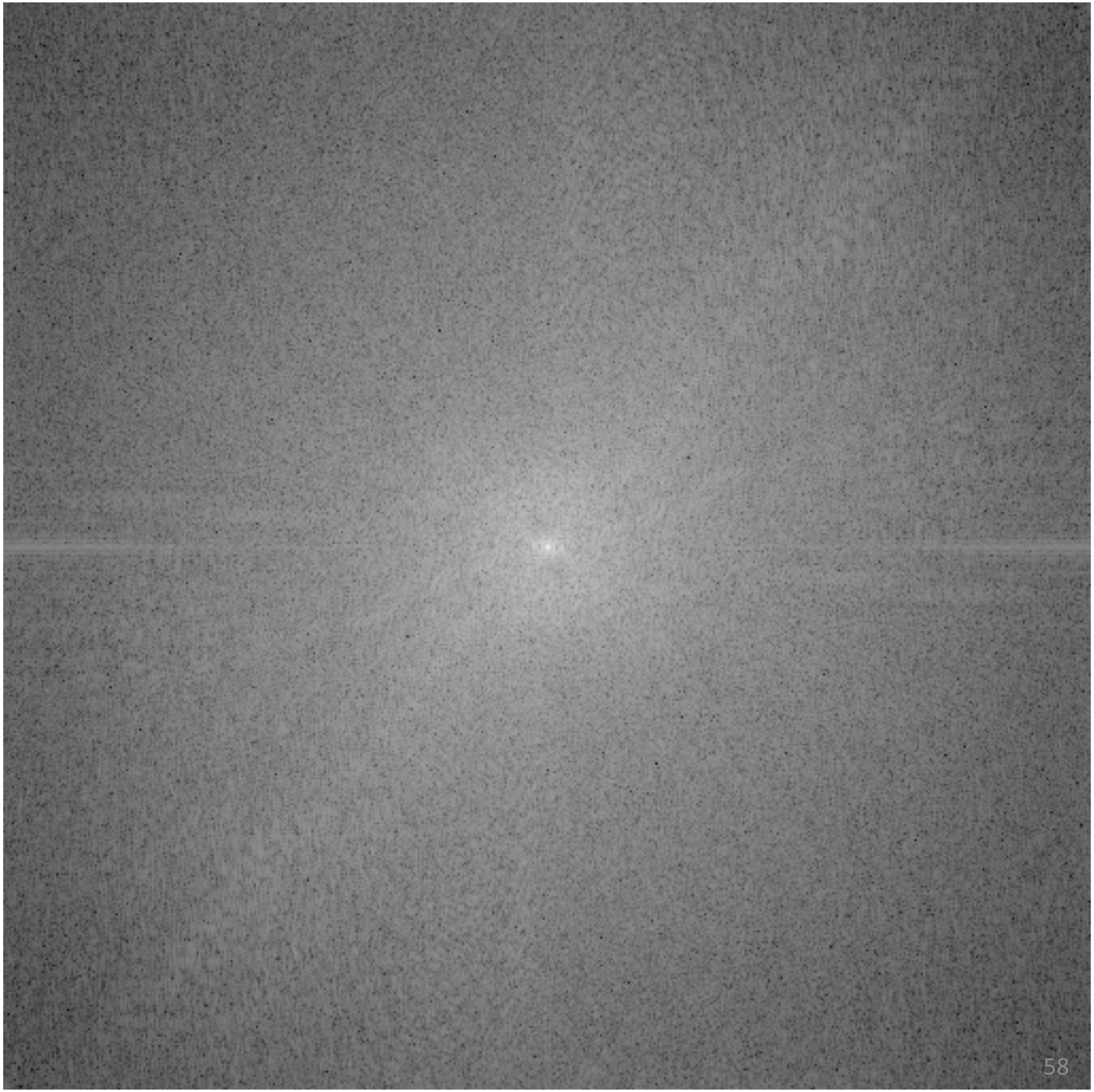




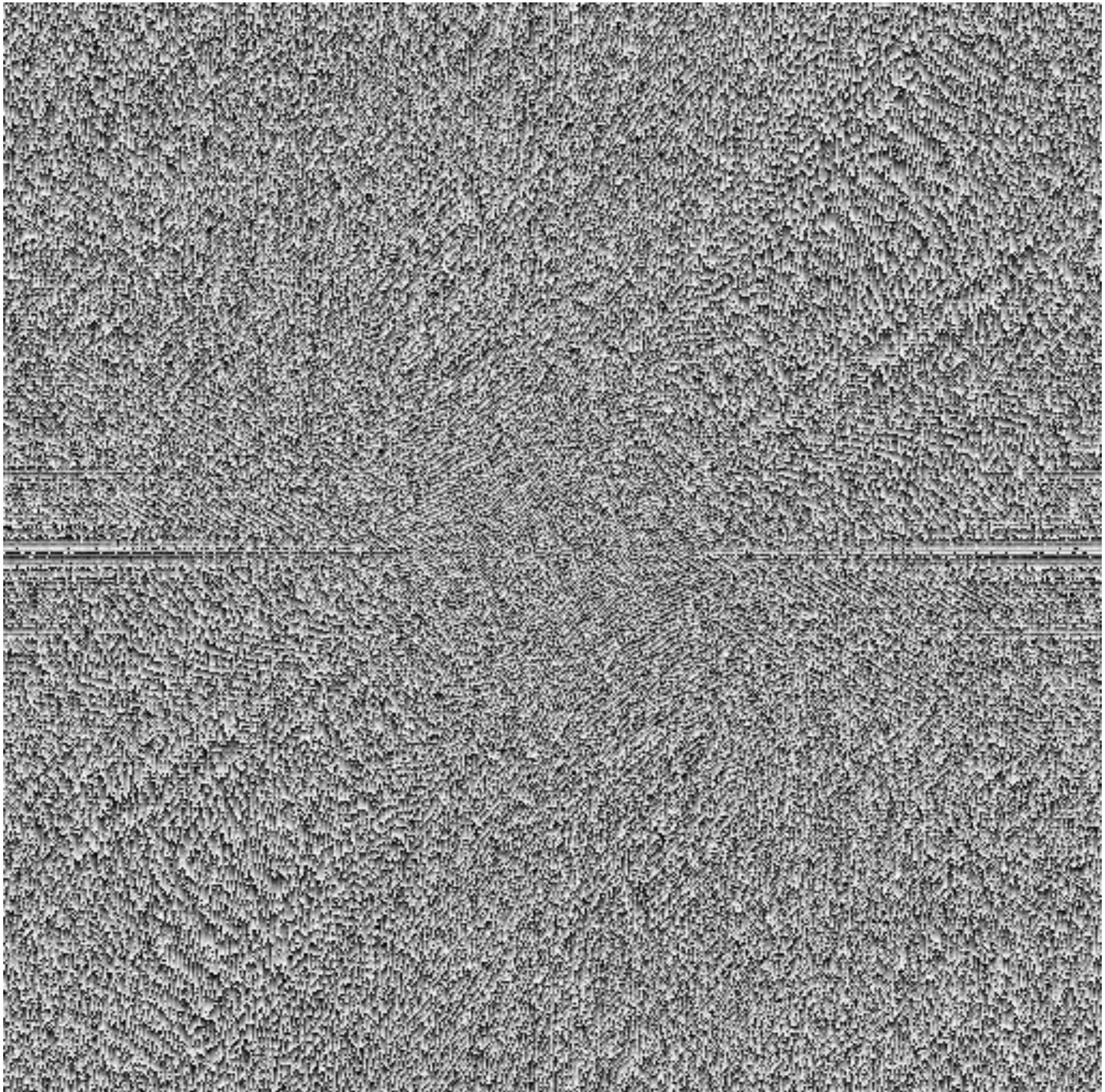
Phase and Magnitude

- Curious fact
  - all natural images have about **the same magnitude transform**
  - hence, phase seems to matter, but magnitude largely doesn't
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

This is the magnitude transform of the cheetah pic

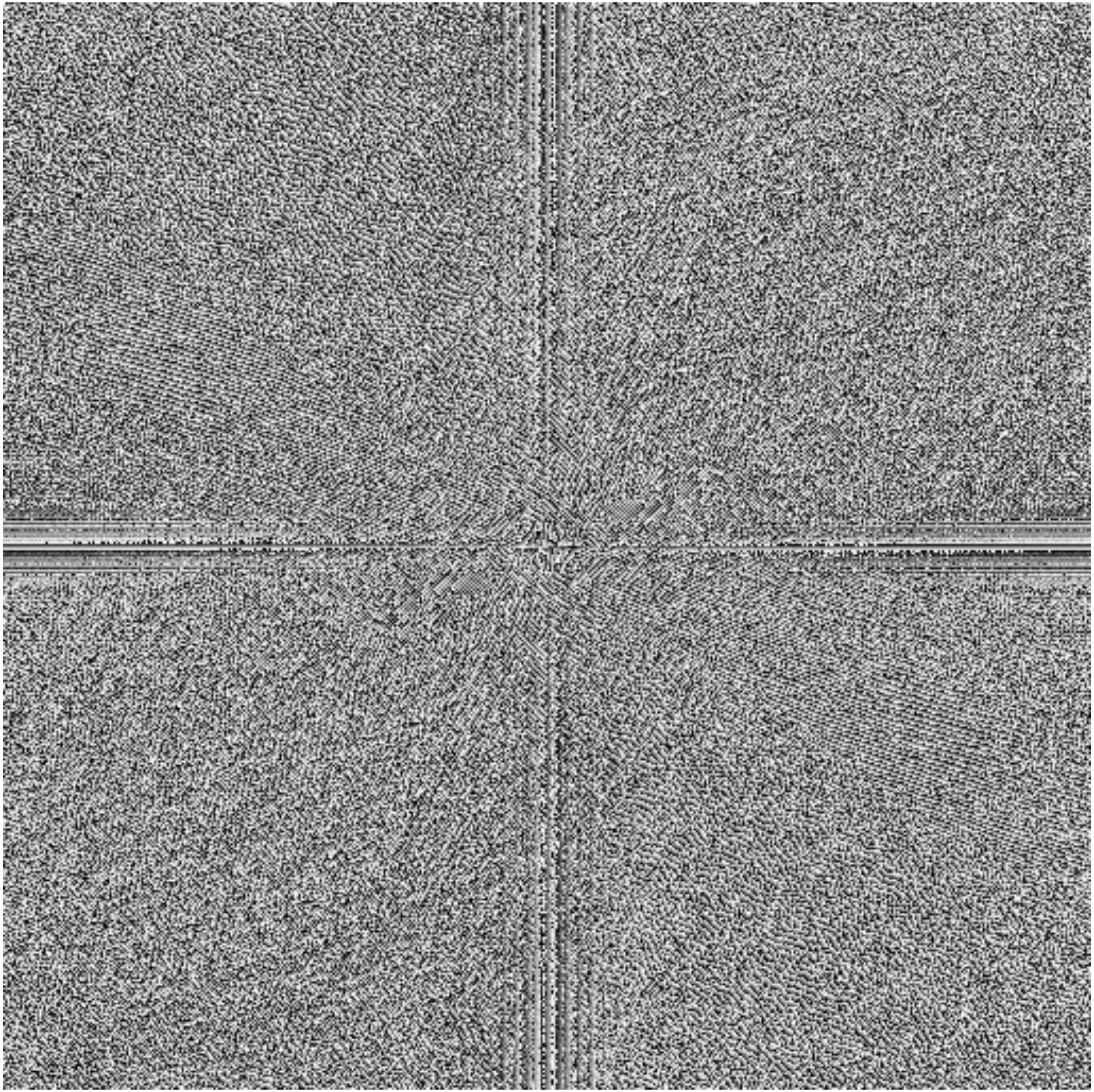


This is the phase transform of the cheetah pic



This is the magnitude transform of the zebra pic

This is the phase transform of the zebra pic



What is a good representation for image analysis?

- Fourier transform domain tells you "what" (textural properties), but not "where".
- Pixel domain representation tells you "where" (pixel location), but not "what".
- Want an image representation that gives you a local description of image events—what is happening where.