MIT CSAIL

6.869: Advances in Computer Vision

Antonio Torralba, 2013

MIT COMPUTER VISION

Lecture 2

Linear filters

• 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.

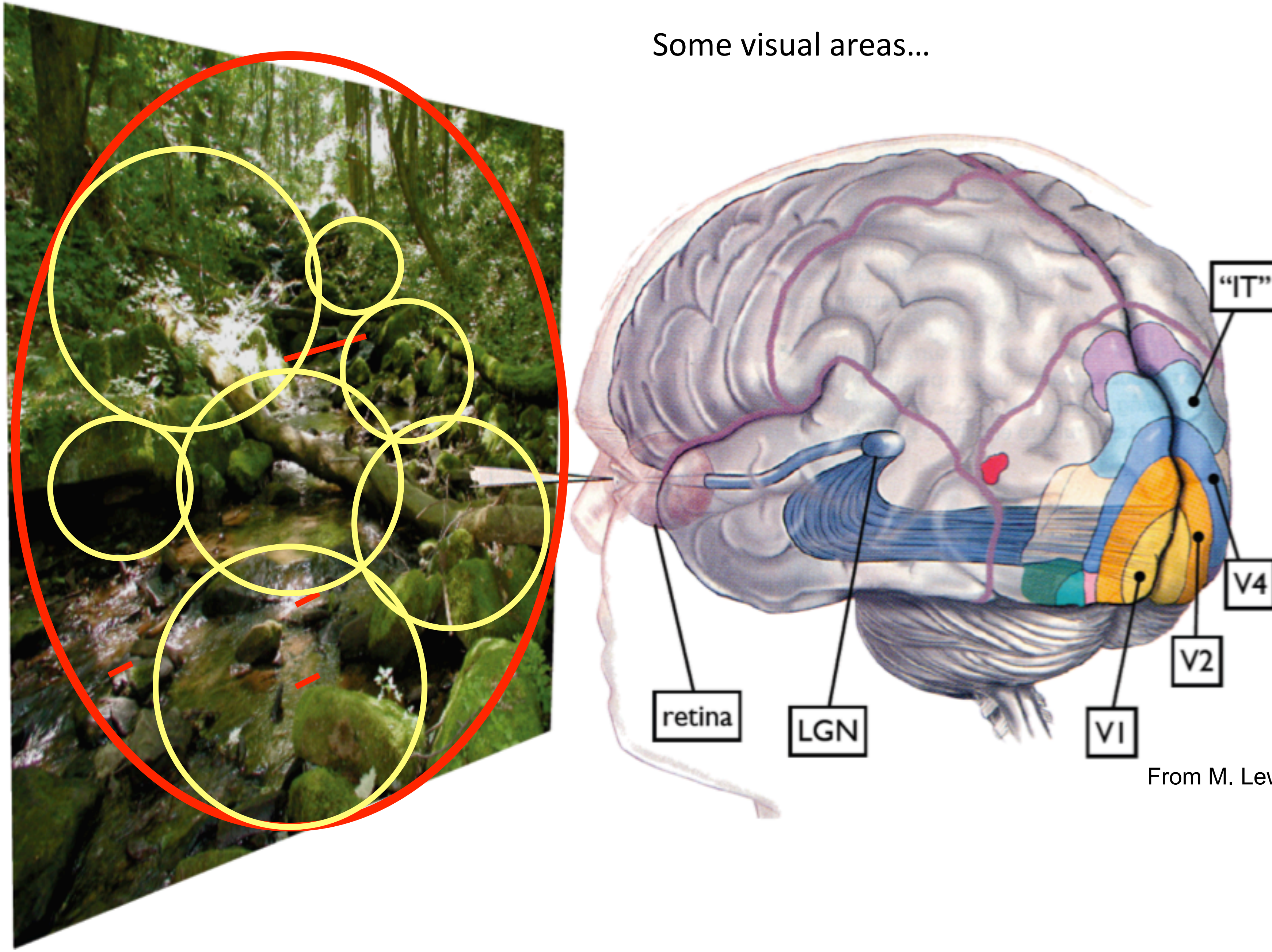
• Goal: to transform the image into other representations (rather than pixel values) that makes scene information more explicit

What we think we see

What we really see

Cavanagh, Perception 95

Some visual areas...

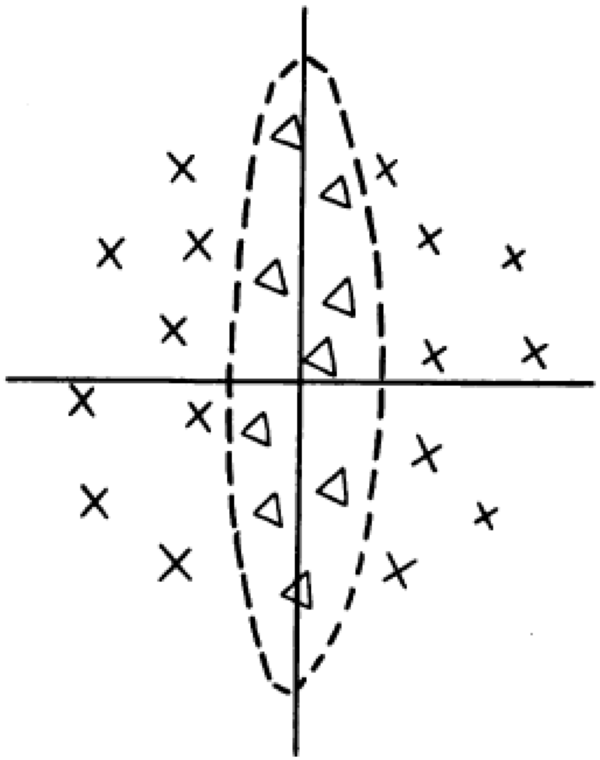


From M. Lewicky

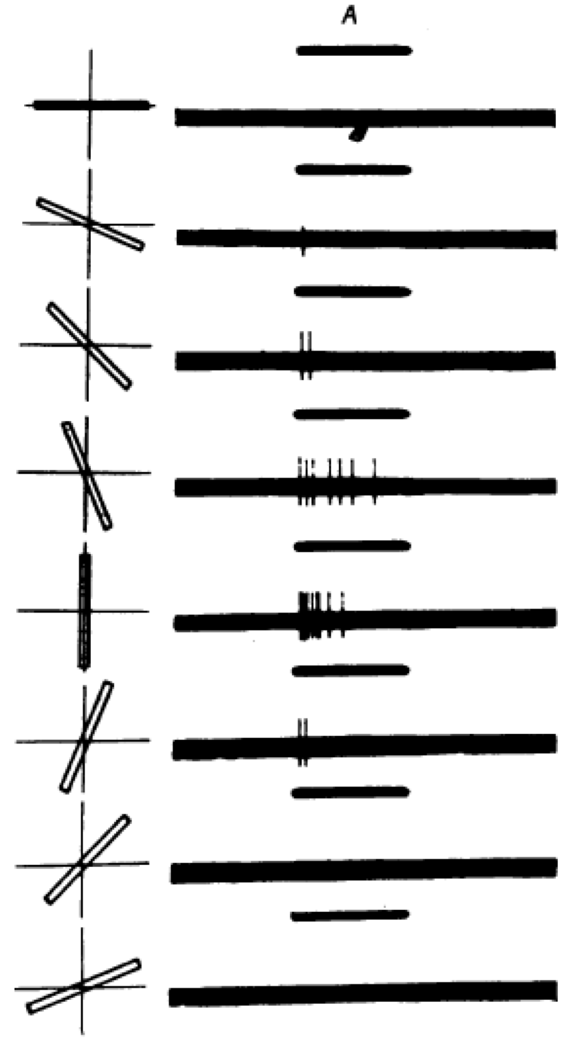
J. Physiol. (1959) 148, 574–591

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

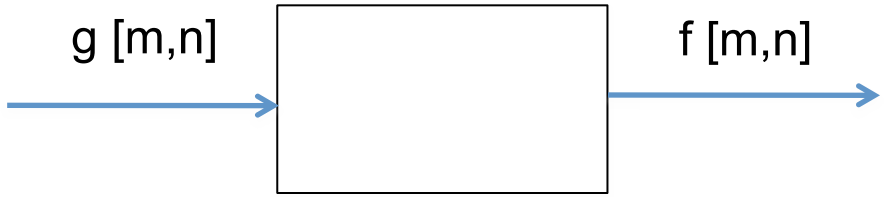


Responses to an oriented bar

Outline

- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

Filtering



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

Linear filtering

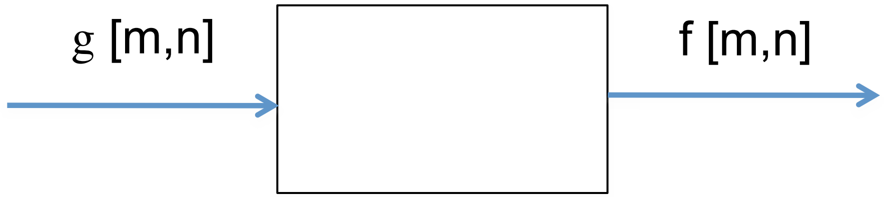
For a linear system, each output is a linear combination of all the input values:

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

In matrix form:

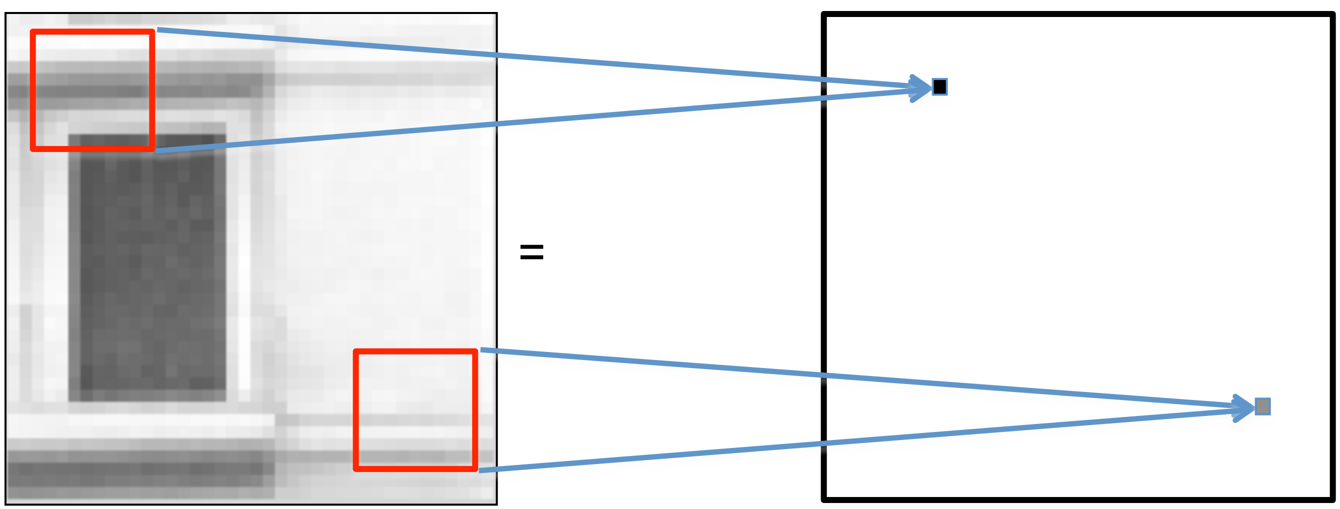
f = H g

Linear filtering



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

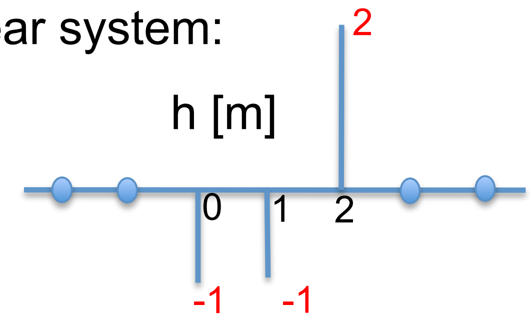
$$f[m,n] = h \otimes g = \sum_{k,l} h[m-k, n-l] g[k,l]$$



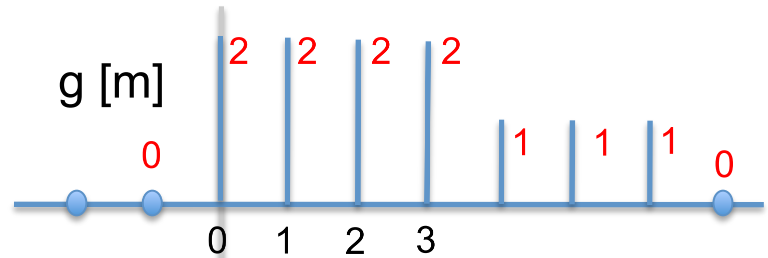
Linear filtering

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

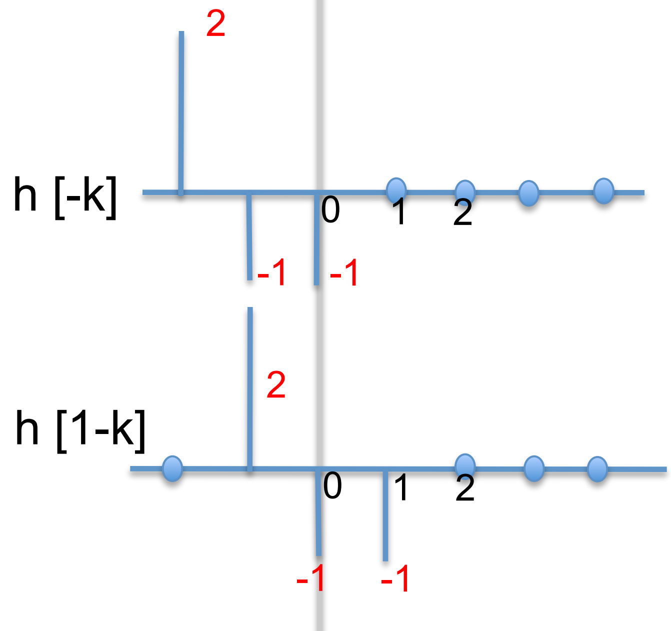
Linear system:



Input:



Output?



$$f[m=0] = \sum_{k} h[-k]\,g[k]$$

f [m=0]=-2

$$f[m=1] = \sum_{k} h[1-k]\,g[k]$$

f [m=1]=-4

$$f[m=2] = \sum_{k} h[2-k]\,g[k]$$

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

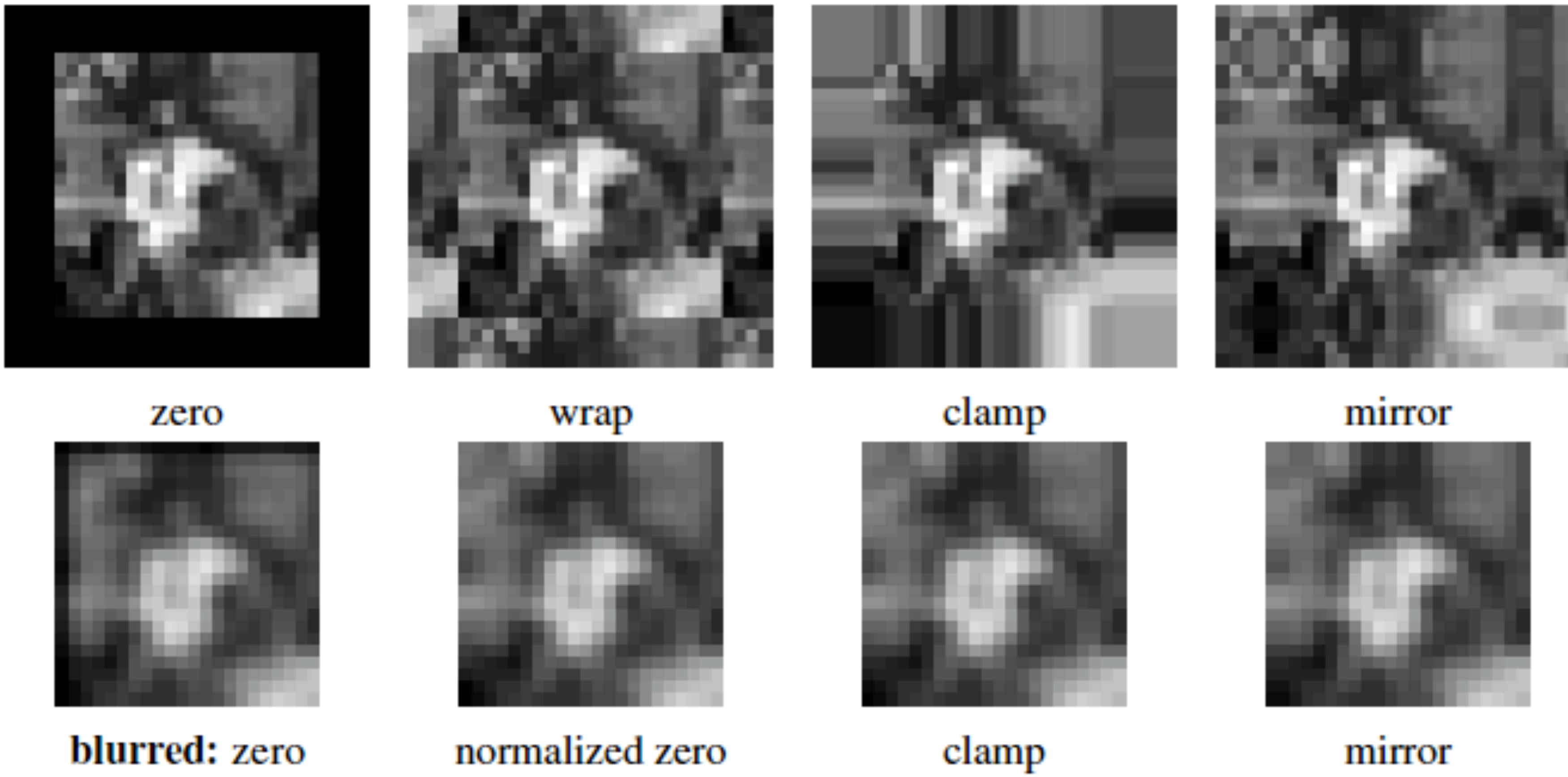
h[m,n]

=

?	?	?	?	?	?	?	?
?	-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]

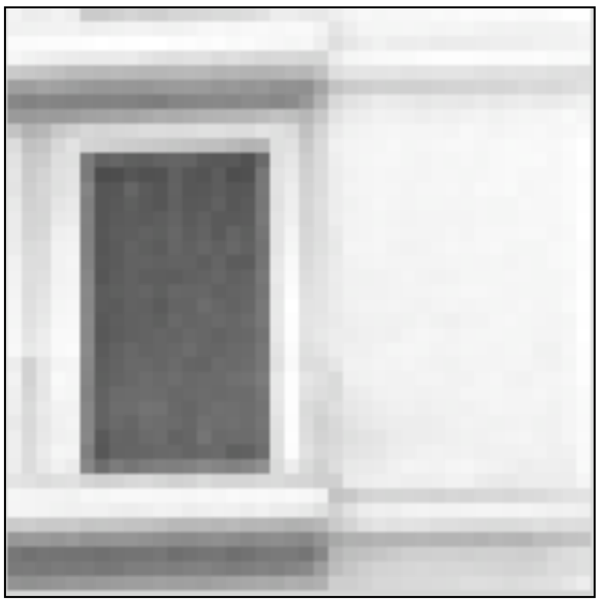
Borders



From Szeliski, Computer Vision, 2010

Impulse

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



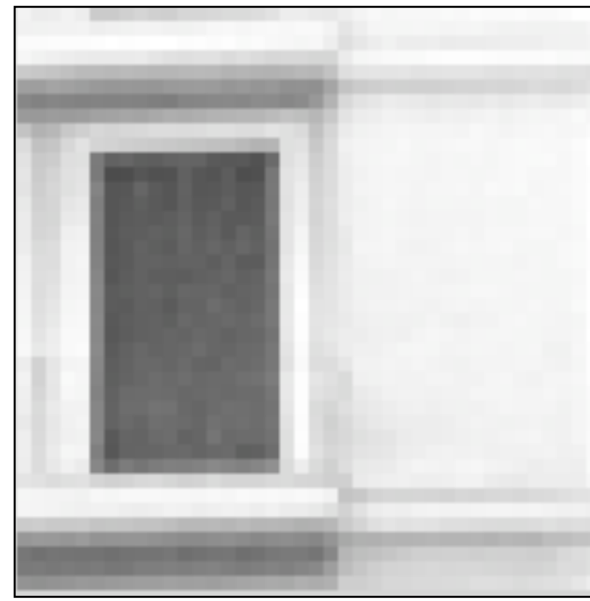
g[m,n]

$\otimes$

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]

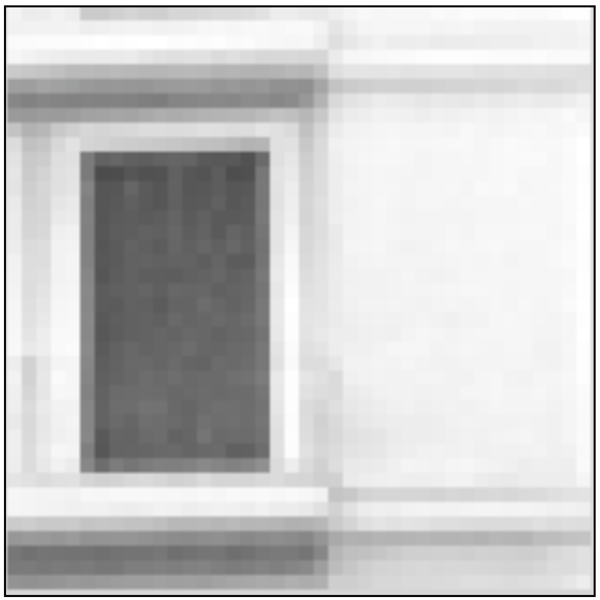
=



f[m,n]

Shifts

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



g[m,n]

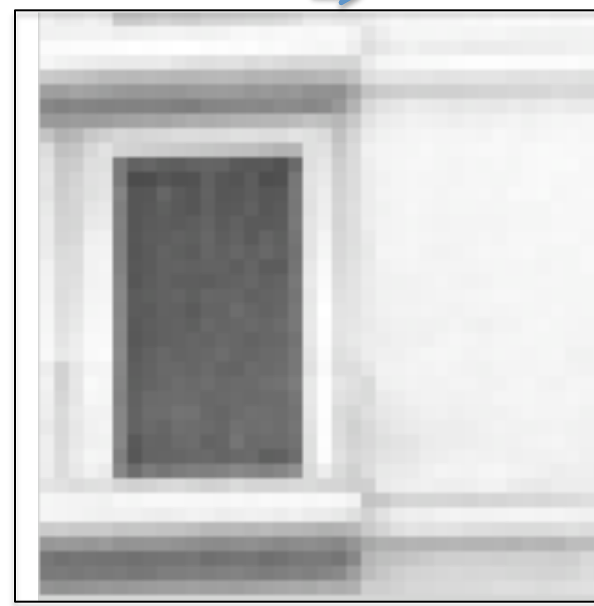
$\otimes$

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

h[m,n]

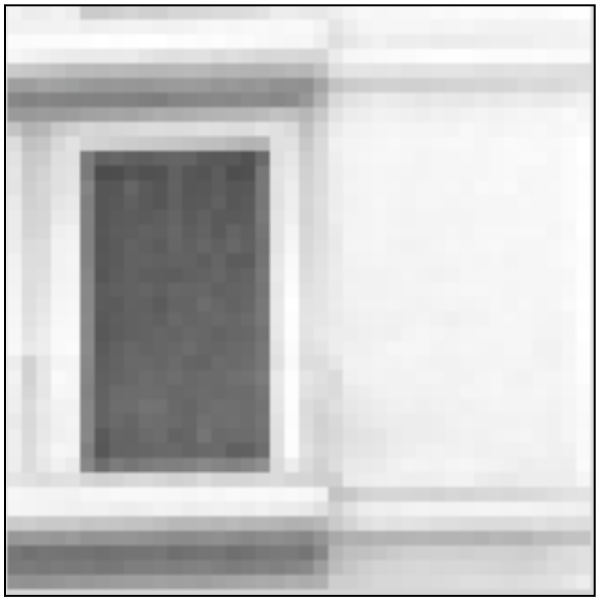
=

2pixels



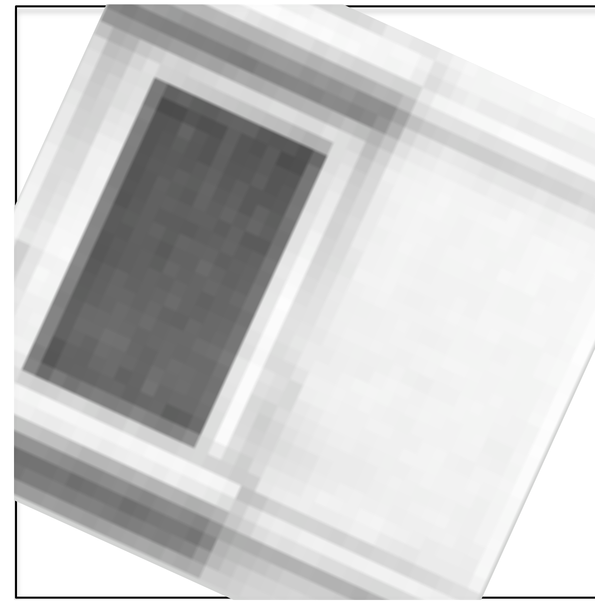
f[m,n]

Image rotation



$\otimes$? =

h[m,n]



g[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]

$\otimes$

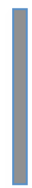
h[m,n]

=

f[m,n]

Rectangular filter

$\otimes$



h[m,n]

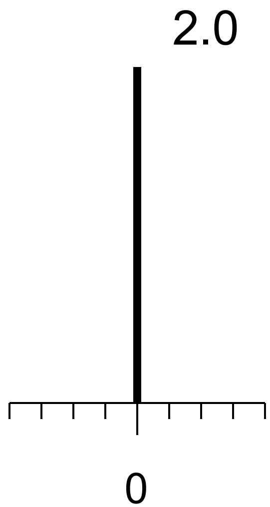
=

f[m,n]

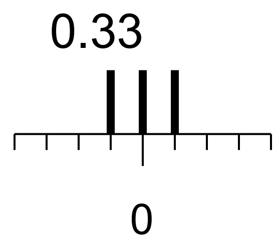
g[m,n]

Sharpening

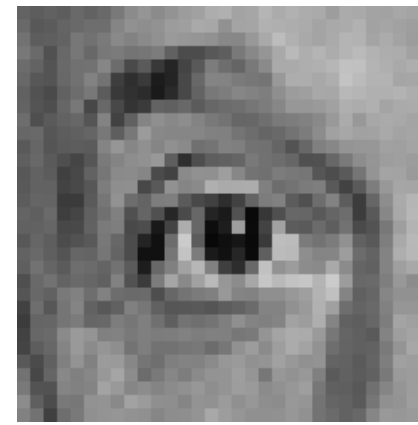
original



−

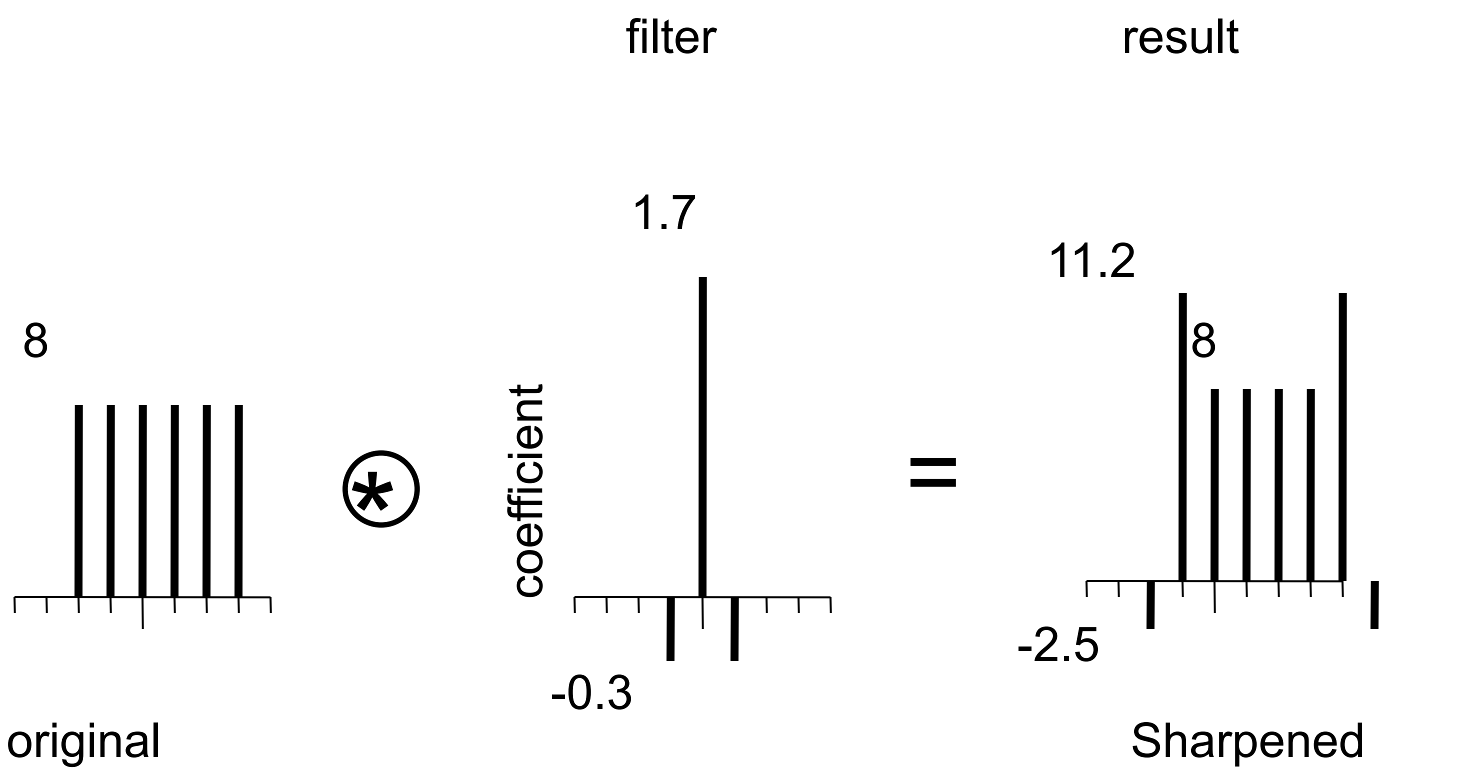


=



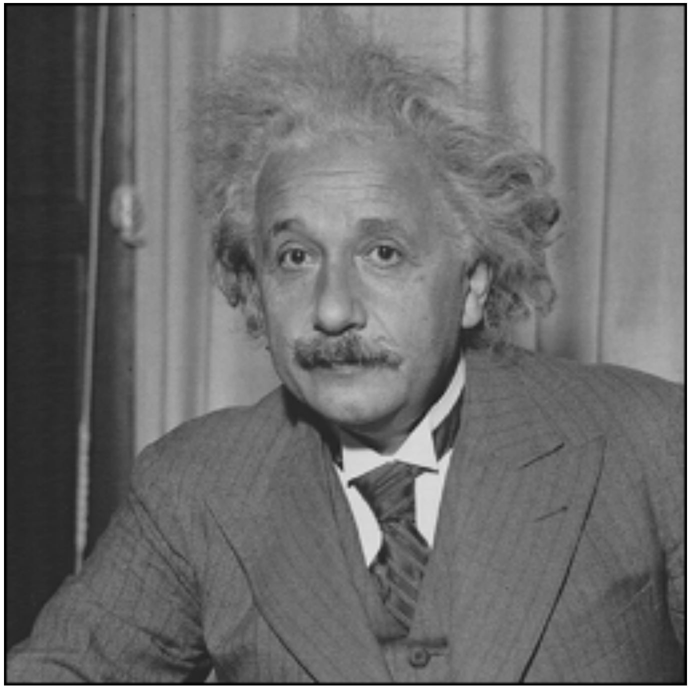
Sharpened
original

Sharpening example

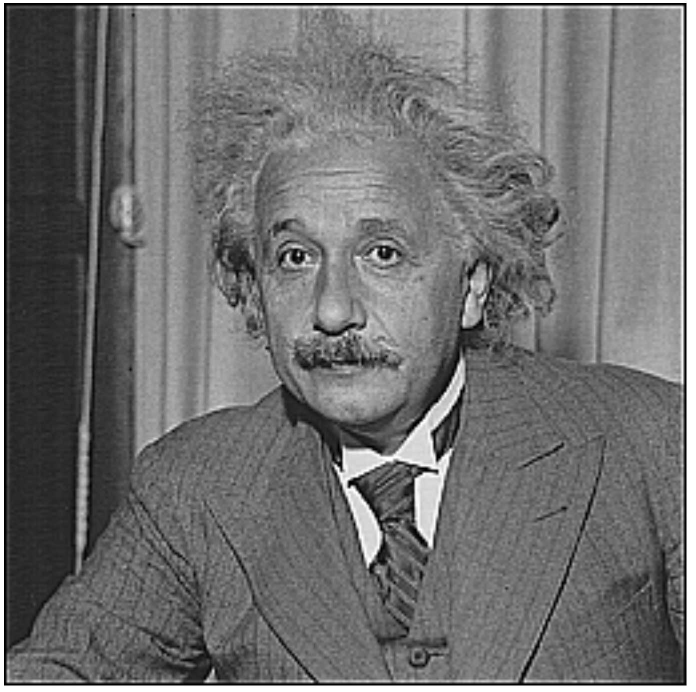


Sharpened
(differences are
accentuated; constant
areas are left untouched).

Sharpening



before



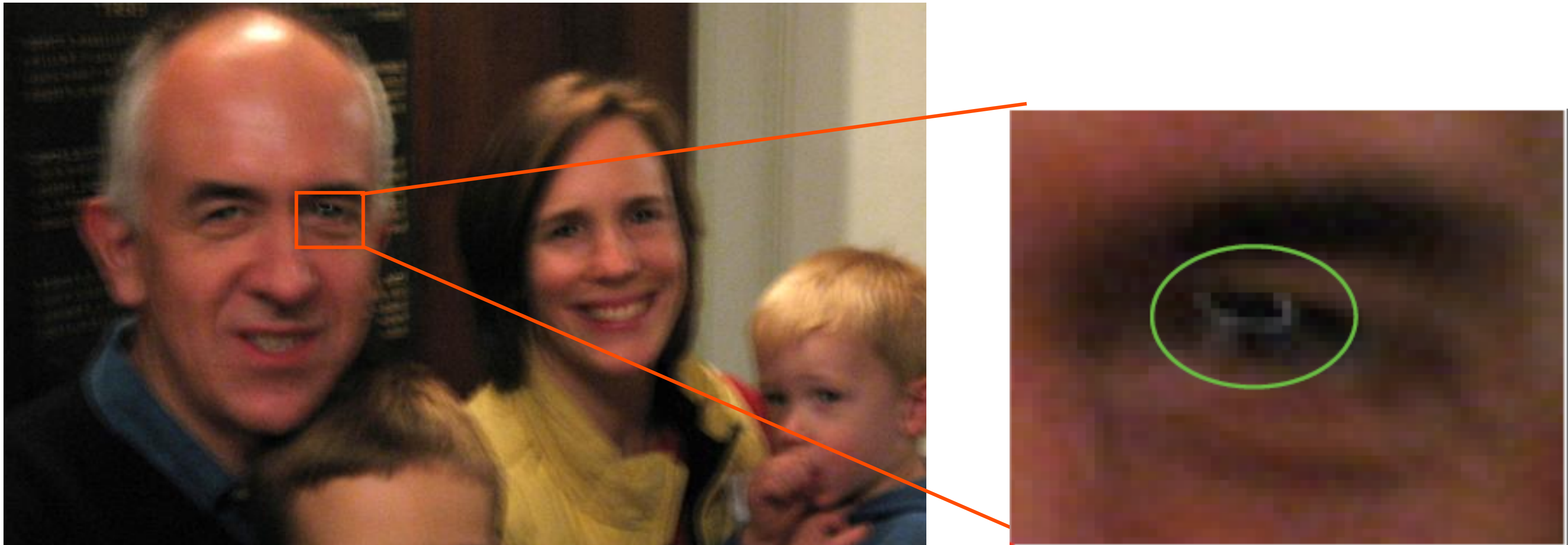
after

A taxonomy of useful filters

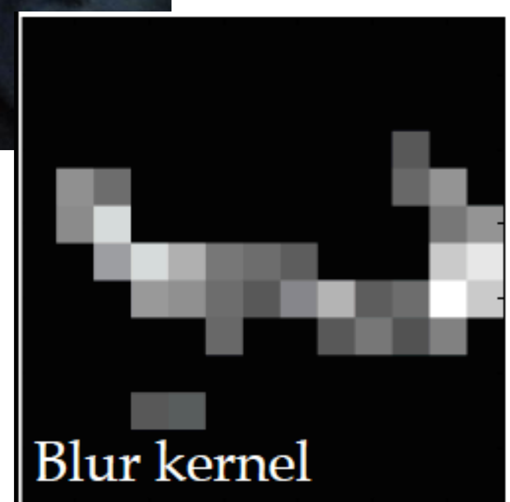
- Impulse, Shifts,
- Blur
  - Rectangular blur (see artifacts)
  - Gaussian
  - Bilateral exponential
  - Asymmetrical filter: motion blur
- Edges
  - [-1 1]
  - Derivative filter
  - Derivative of a gaussian
  - Oriented filters
  - Gabor filter
  - Quadrature filters: phase and magnitude.
  - Elongated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

BLUR

Linear blur occurs under many natural situations



(from Fergus et al, 2007)

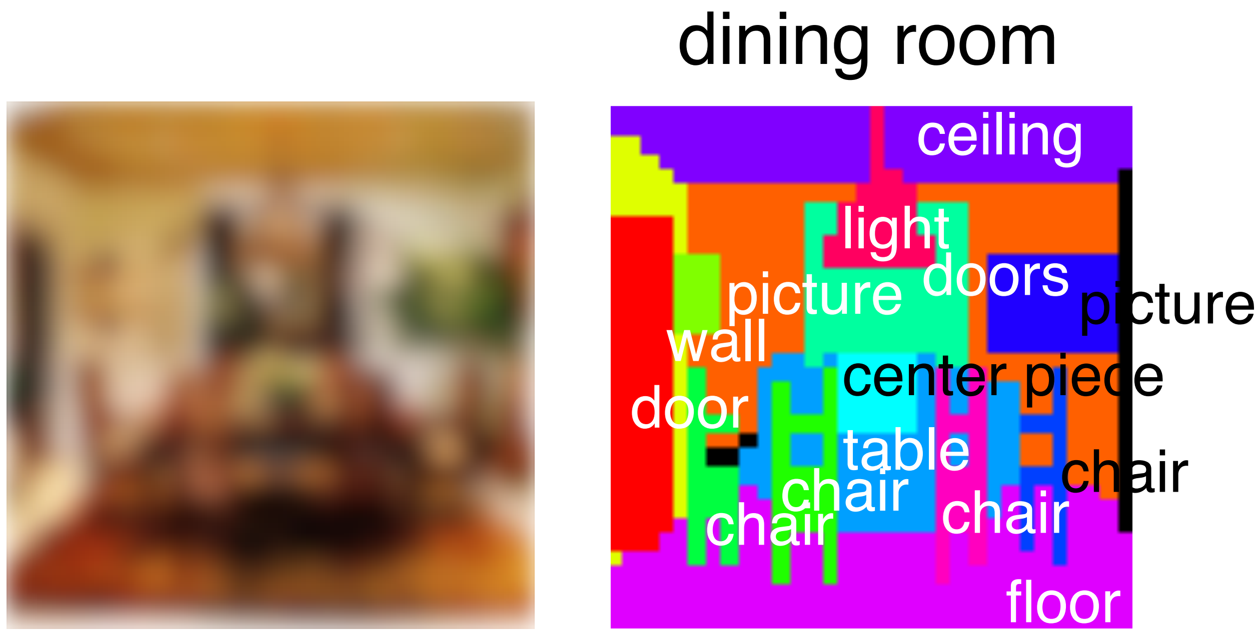


This is not a Gaussian kernel...

Linear blur occurs under many natural situations

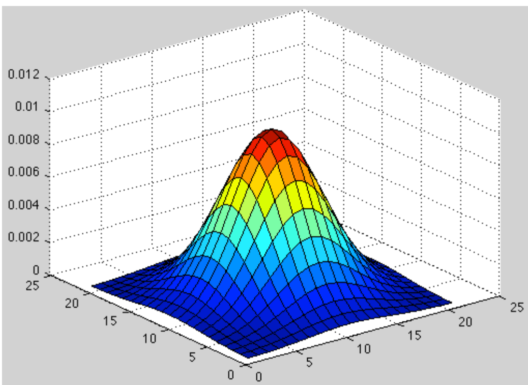
Linear blur occurs under many natural situations

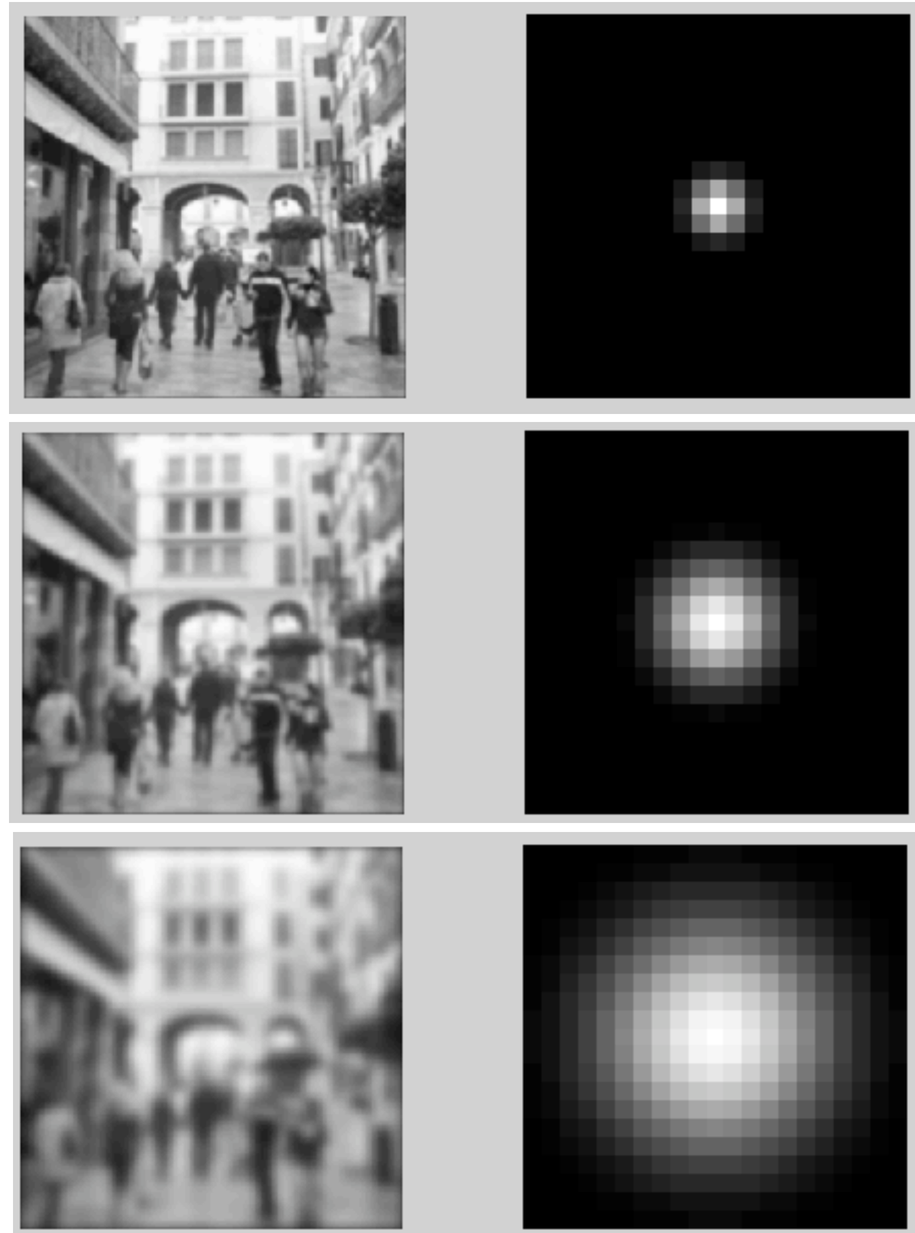
Linear blur occurs under many natural situations



Gaussian filter

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



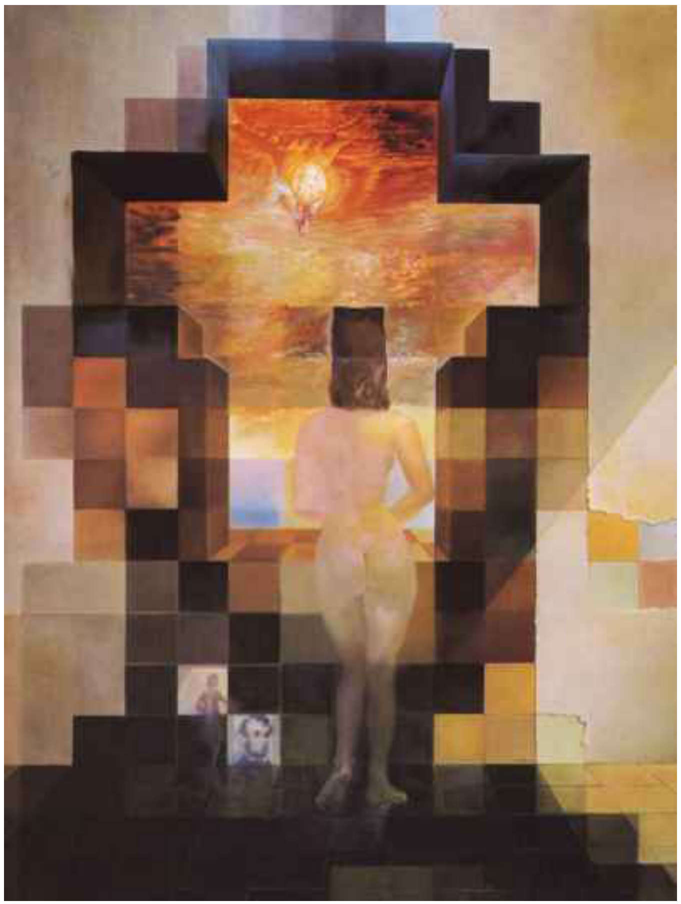


σ=1

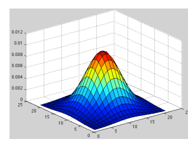
σ=2

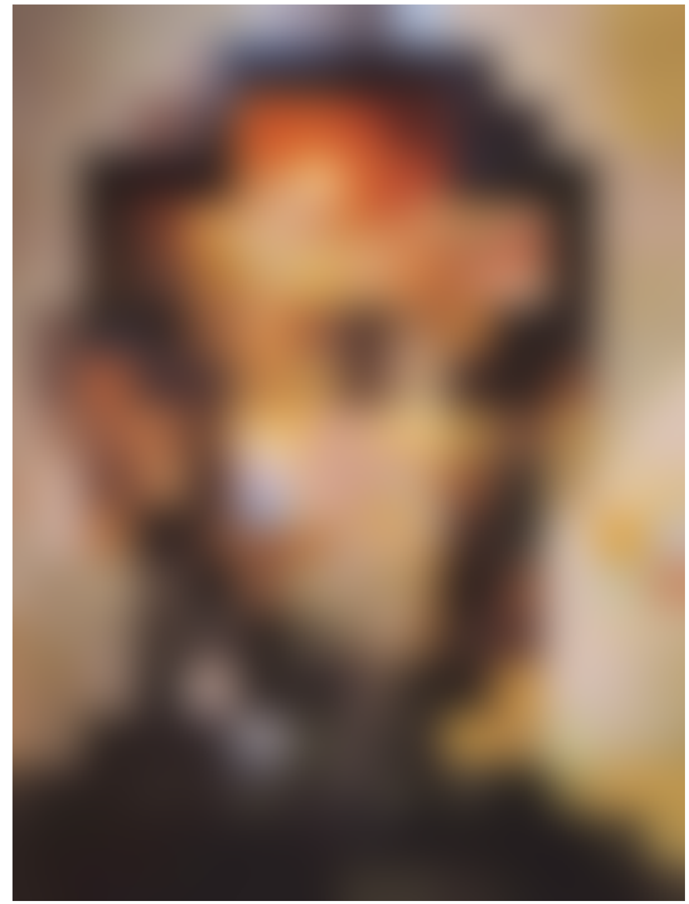
σ=4

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]

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

g[m,n]

$\otimes$

[-1, 1]

h[m,n]

=

f[m,n]

$[-1\ 1]^T$

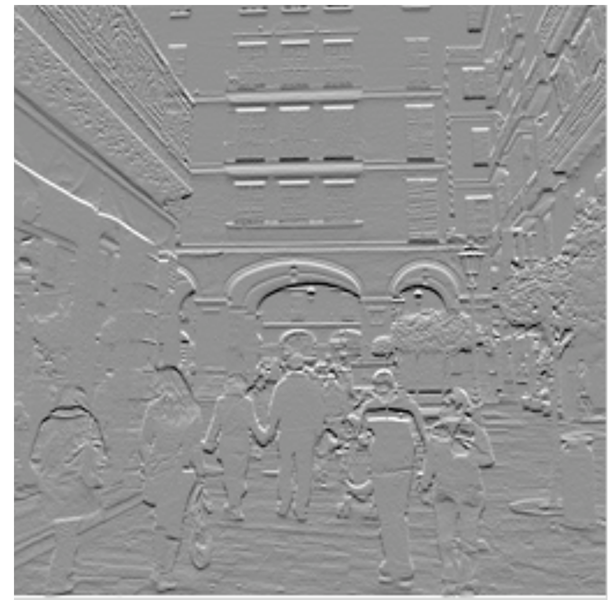
g[m,n]

$\otimes$

$[-1, 1]^T$

h[m,n]

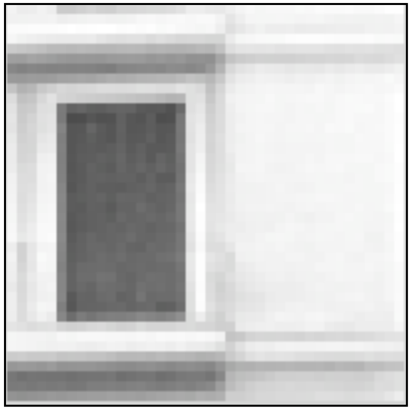
$=$

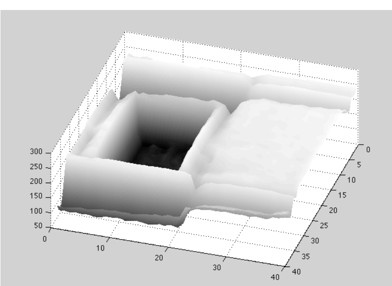


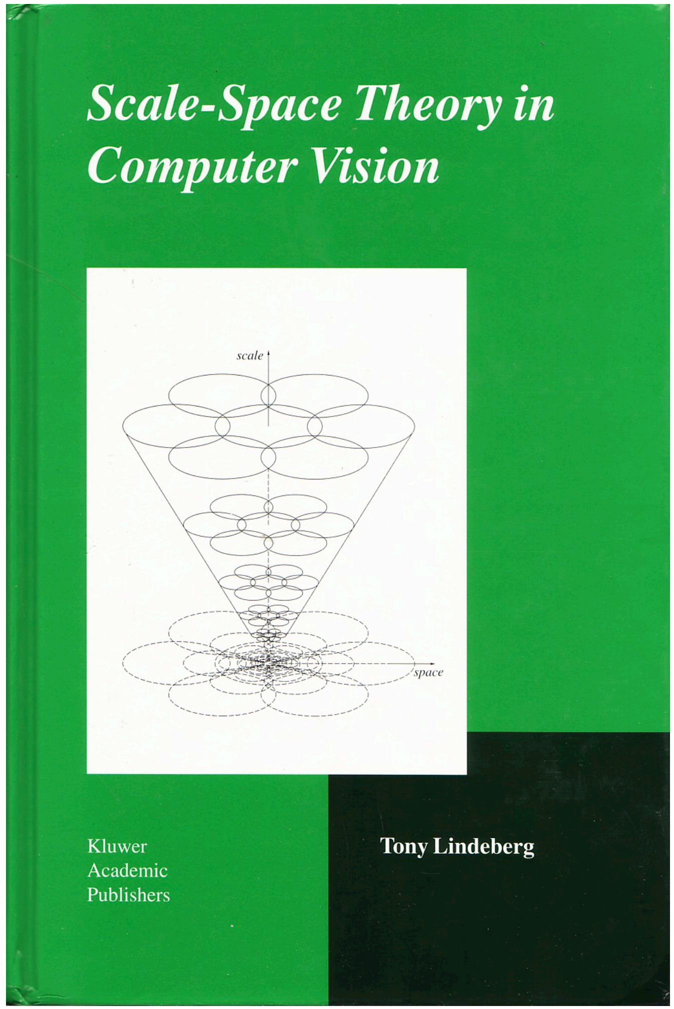
f[m,n]

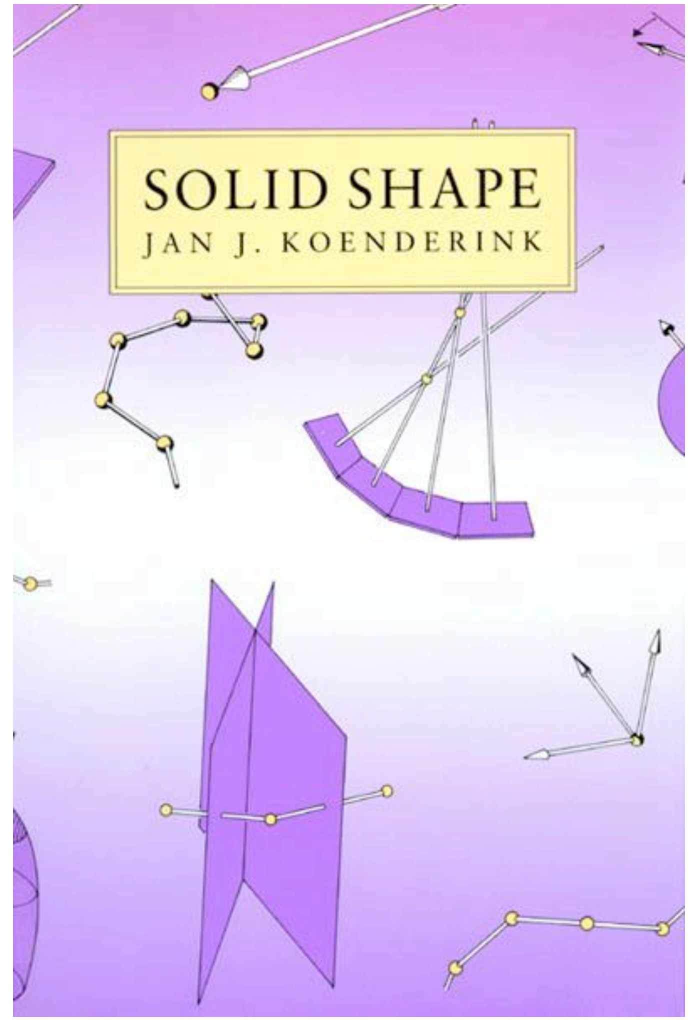
Differential Geometry Descriptors

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:

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

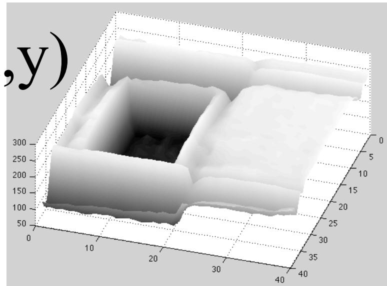
Edge strength $\quad E(x, y) = |\nabla \mathbf{I}(x, y)|$

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

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

Differential Geometry Descriptors

I(x,y)



If we think of the image as a continuous function

Image gradient:

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

Directional gradient:

$|u|=1$ α

$$u^T \nabla I = \cos(\alpha) \frac{\partial I(x,y)}{\partial x} + \sin(\alpha) \frac{\partial I(x,y)}{\partial y}$$

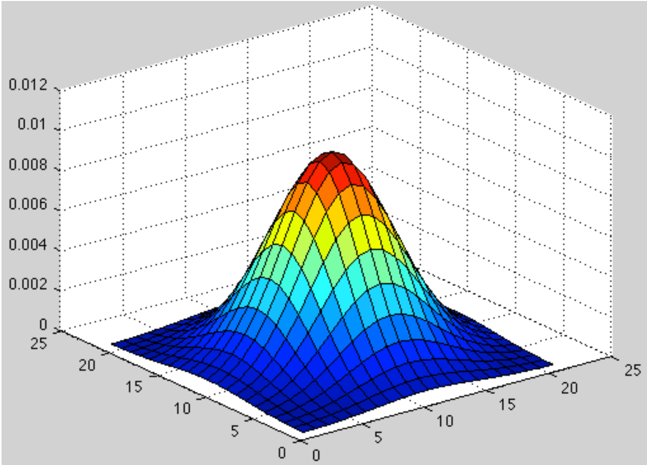
Laplacian:

$$\nabla^2 I = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2}$$

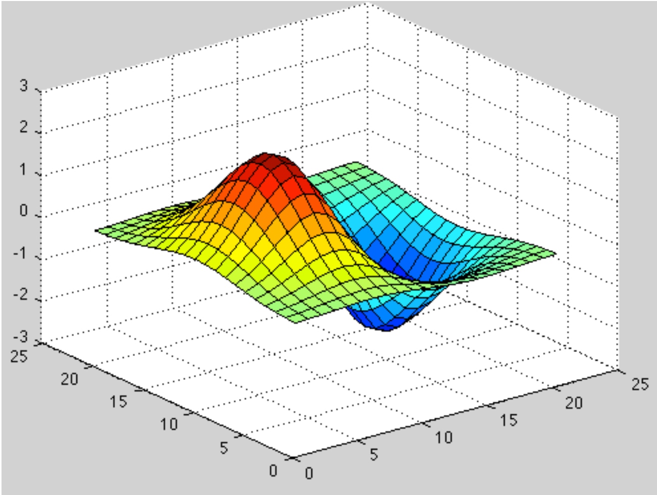
Problem: *dI/dx* might not be defined around discontinuities.

Gaussian derivative

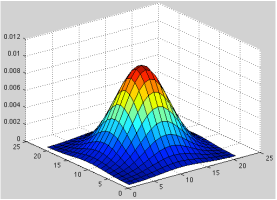
$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$\frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



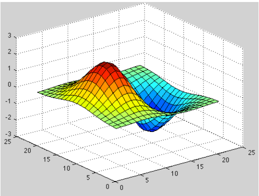
$$g(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$\frac{\partial I(x,y)}{\partial x}\otimes g(x,y)=$$

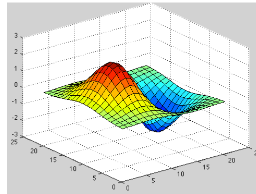
$$=\frac{\partial I(x,y)\otimes g(x,y)}{\partial x}=$$

$$=I(x,y)\otimes\frac{\partial g(x,y)}{\partial x}$$

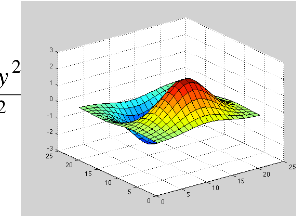


$$\frac{\partial g(x,y)}{\partial x}=\frac{-x}{2\pi\sigma^4}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$g_x(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$g_y(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



The smoothed directional gradient is a linear combination of two kernels

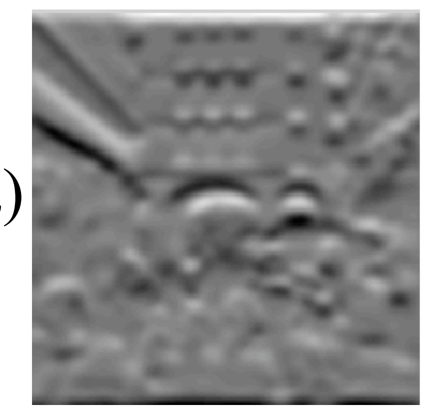
$$u^T \nabla g \otimes I = \left(\cos(\alpha) g_x(x,y) + \sin(\alpha) g_y(x,y)\right) \otimes I(x,y) =$$

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

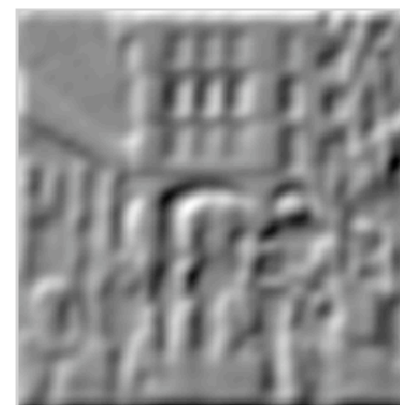
$$= \cos(\alpha) g_x(x,y) \otimes I(x,y) + \sin(\alpha) g_y(x,y) \otimes I(x,y) =$$

$= \cos(\alpha)$

$+\sin(\alpha)$



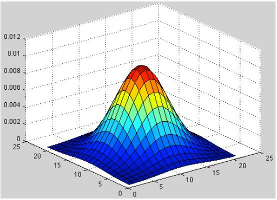
$=$



Steereability of gaussian derivatives, Freeman & Adelson 92

Laplacian

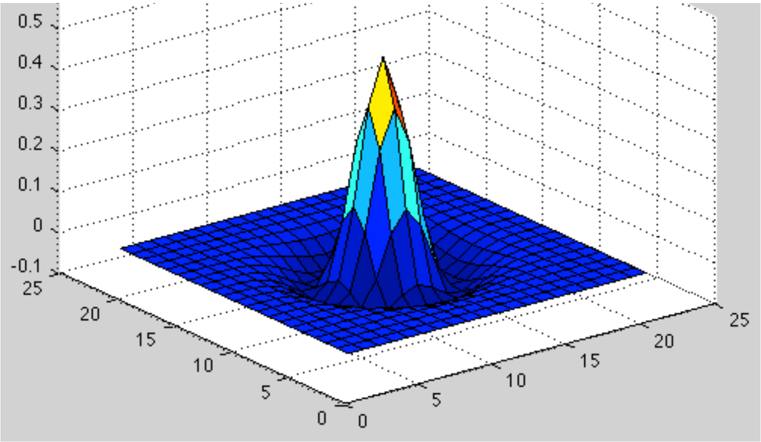
$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



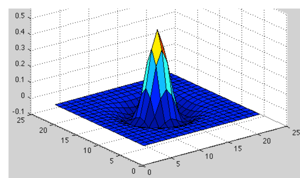
$$\nabla^2 I \otimes g = \left(\frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \right) \otimes g(x,y)$$

$$\nabla^2 I \otimes g = I \otimes \nabla^2 g$$

$$\nabla^2 g(x,y) = \left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) g(x,y)$$



Laplacian



Outline

- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

Linear image transformations

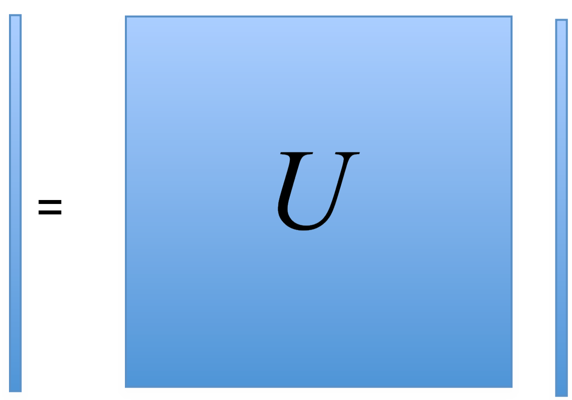
- In analyzing images, it's often useful to make a change of basis.

Transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



Self-inverting transforms

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

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

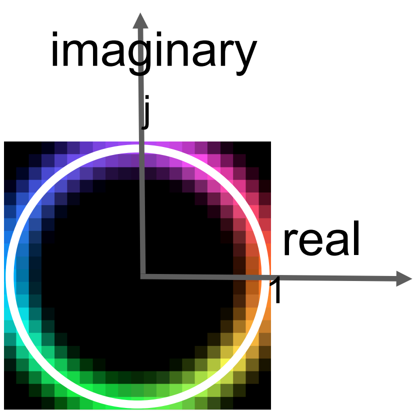
Forward transform

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

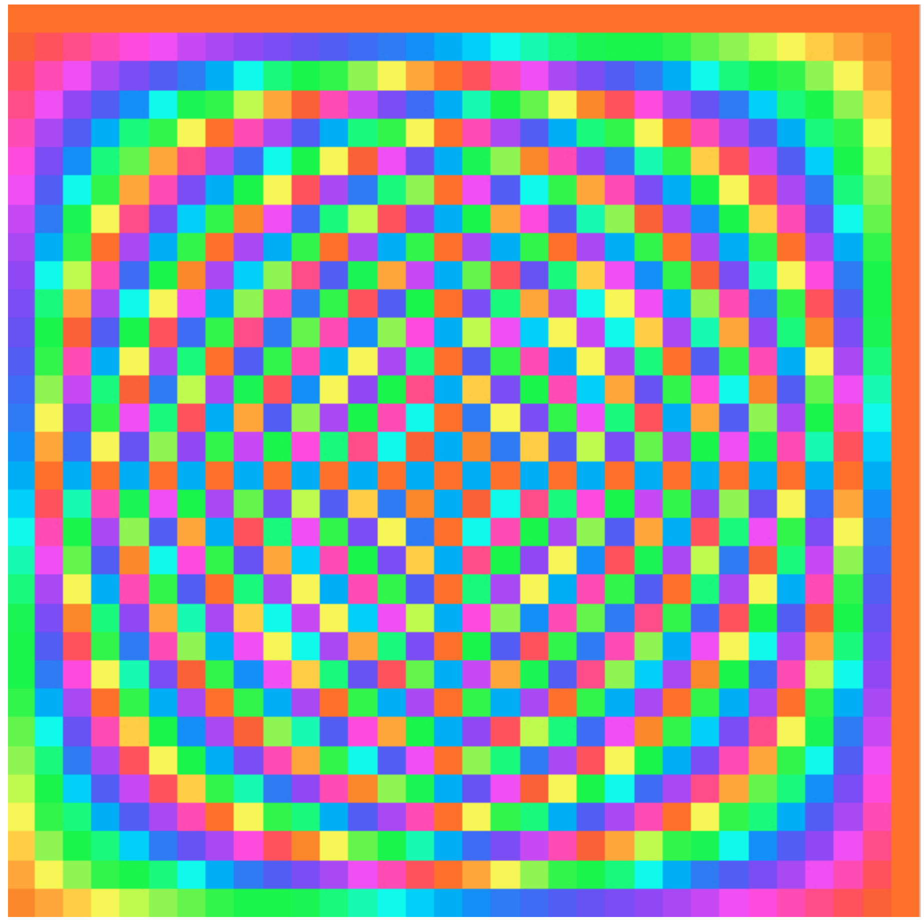
Inverse transform

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

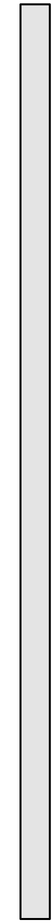
Fourier transform visualization



color key



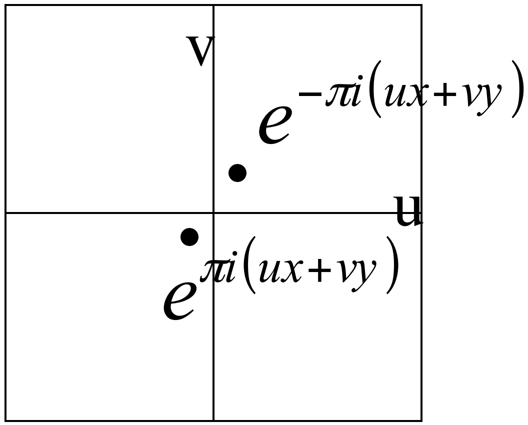
Fourier transform matrix



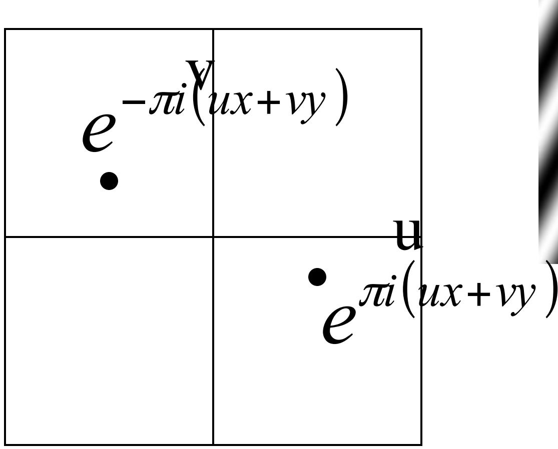
input signal

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

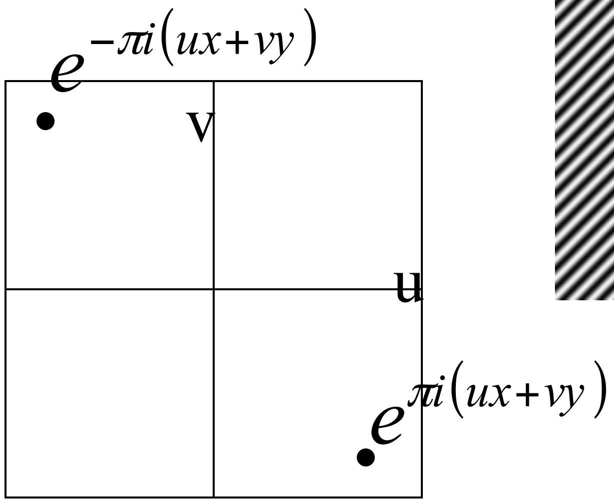
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

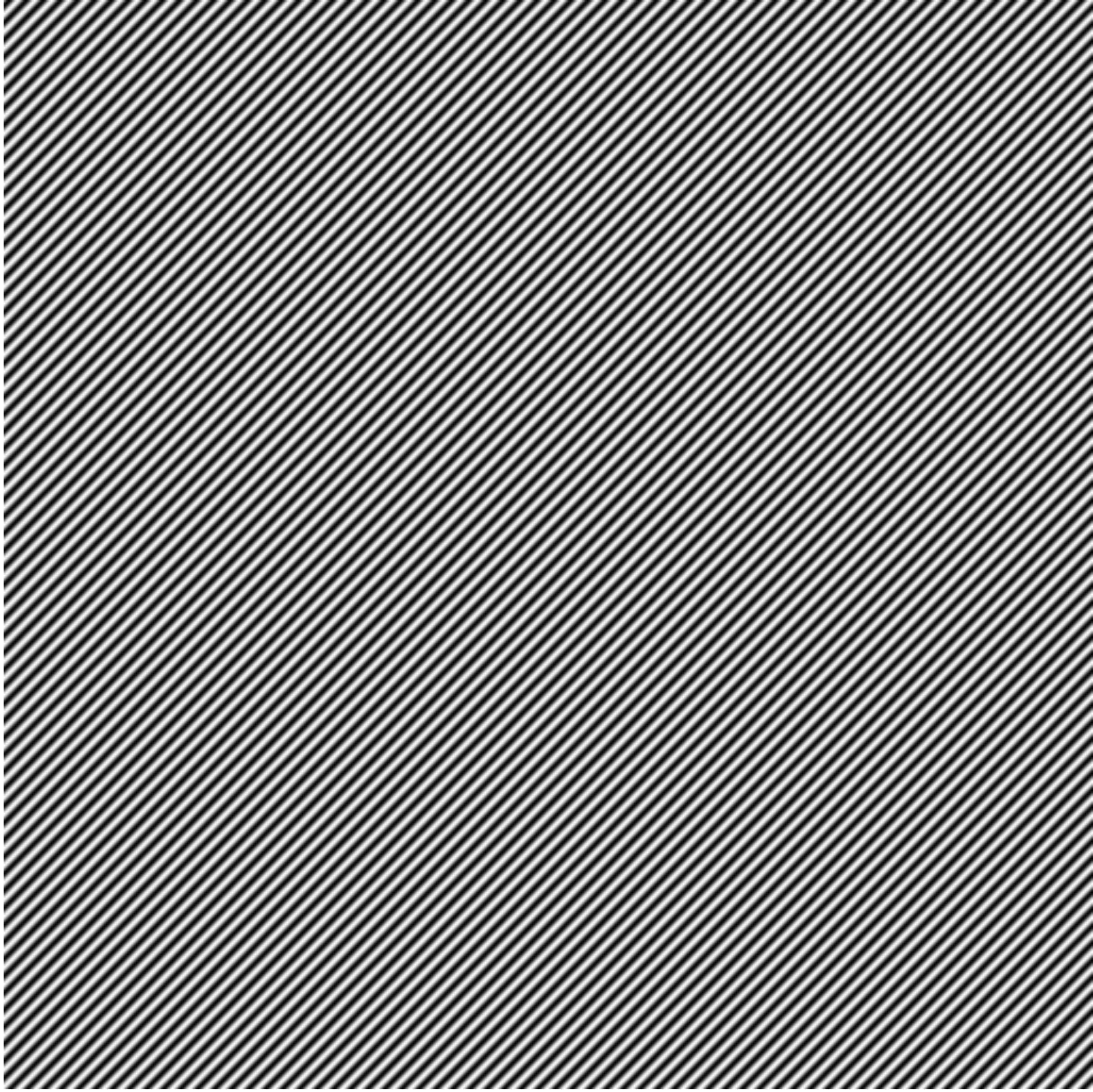


Here u and v are larger than in the previous slide.



And larger still...





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

$$\begin{aligned} F[m,n] &= DFT\left(g \otimes h\right) \\ &= G\left[m,n\right]H\left[m,n\right] \end{aligned}$$

Fourier transform of convolution

$$f = g \otimes h$$
Consider a (circular) convolution of g and h

$$F[m,n] = DFT\left(g \otimes h\right)$$
Take DFT of both sides

$$F\left[m,n\right] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k,v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$$
Write the DFT and convolution explicitly

$$= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k,v-l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}h[k,l]$$
Move the exponent in

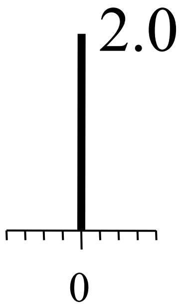
$$= \sum_{\mu=-k}^{M-k-1}\sum_{\upsilon=-l}^{N-l-1}\sum_{k,l} g[\mu,\upsilon]e^{-\pi i\left(\frac{(k+\mu)m}{M}+\frac{(l+\upsilon)n}{N}\right)}h[k,l]$$
Change variables in the sum

$$= \sum_{k,l} G\left[m,n\right]e^{-\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}h[k,l]$$
Perform the DFT (circular boundary conditions)

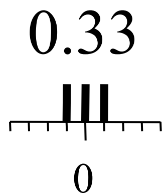
$$= G\left[m,n\right]H\left[m,n\right]$$
Perform the other DFT (circular boundary conditions)

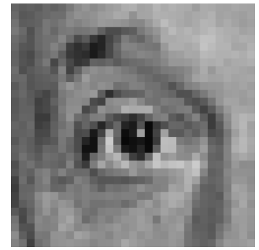
Analysis of a simple sharpening filter

original



−

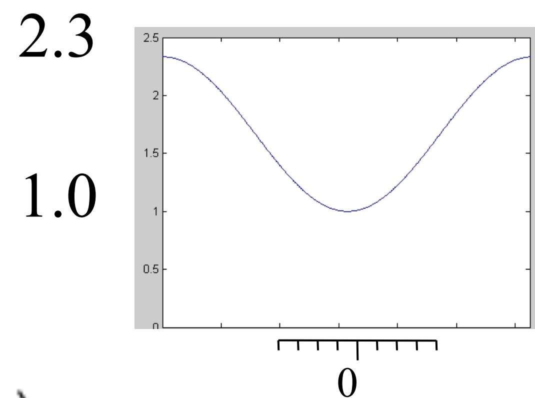




sharpened

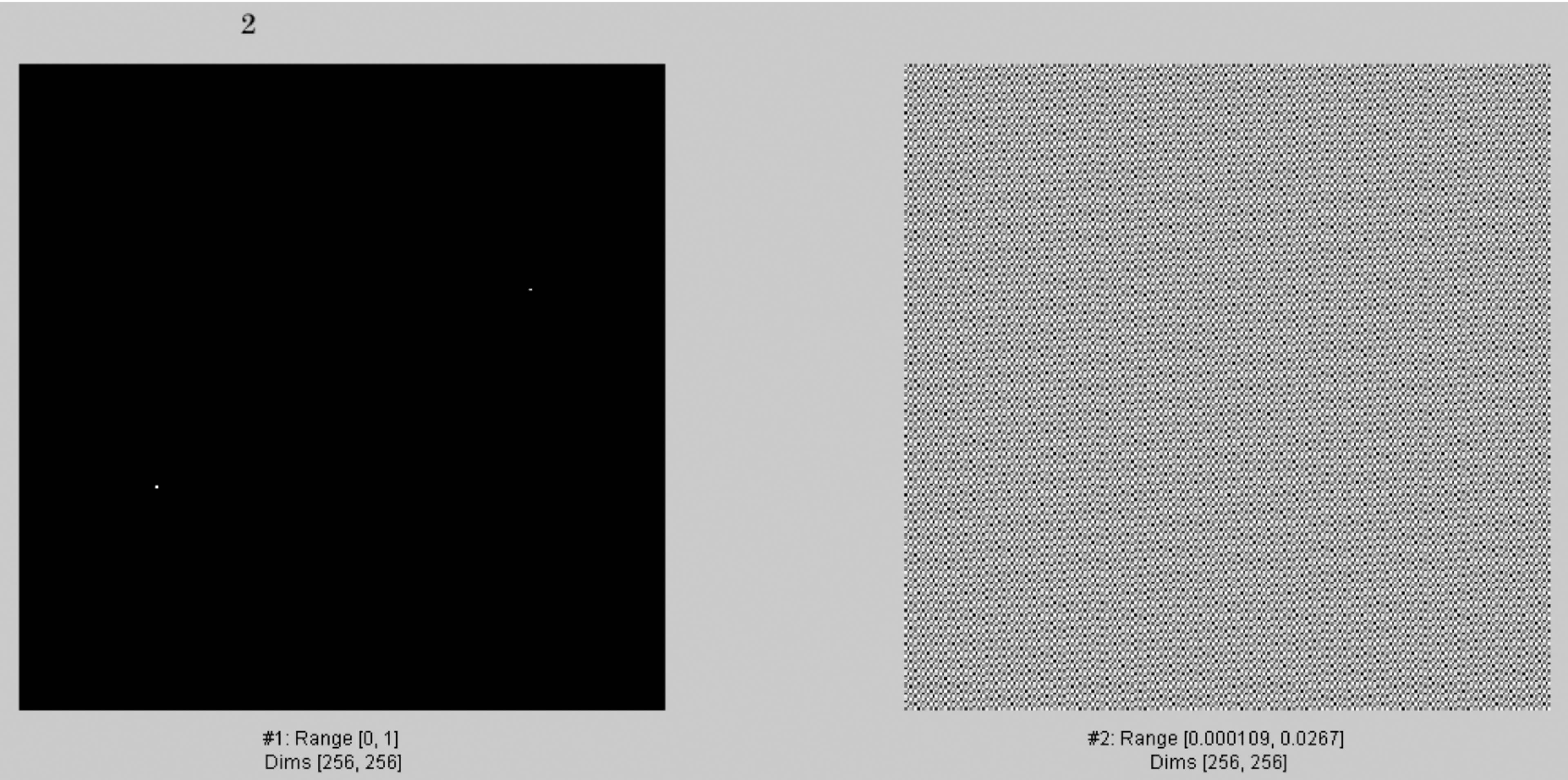
$$F[m] = \sum_{k=0}^{M-1} f[k] e^{-\pi i \left(\frac{km}{M}\right)}$$

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

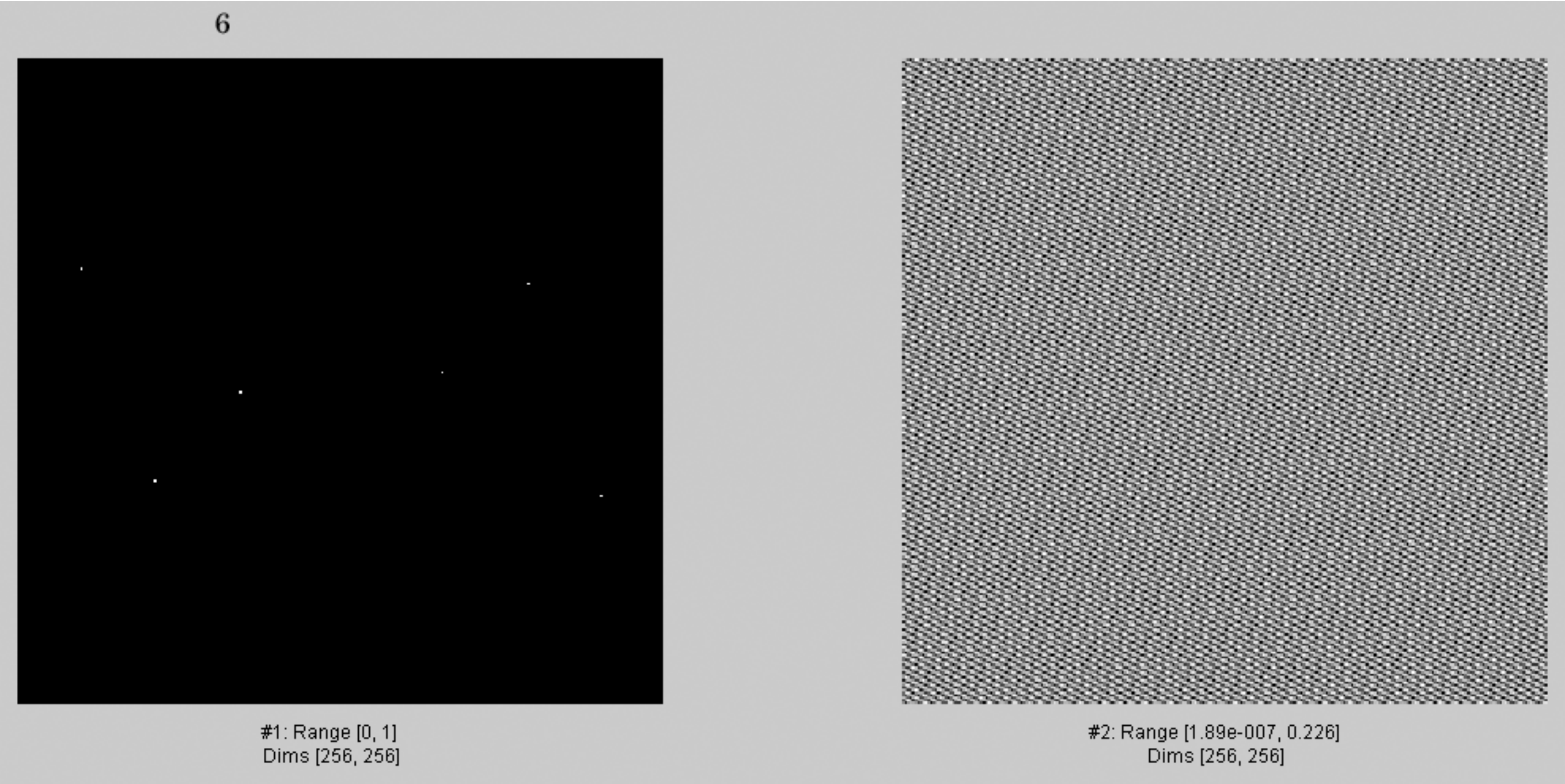


high-pass filter

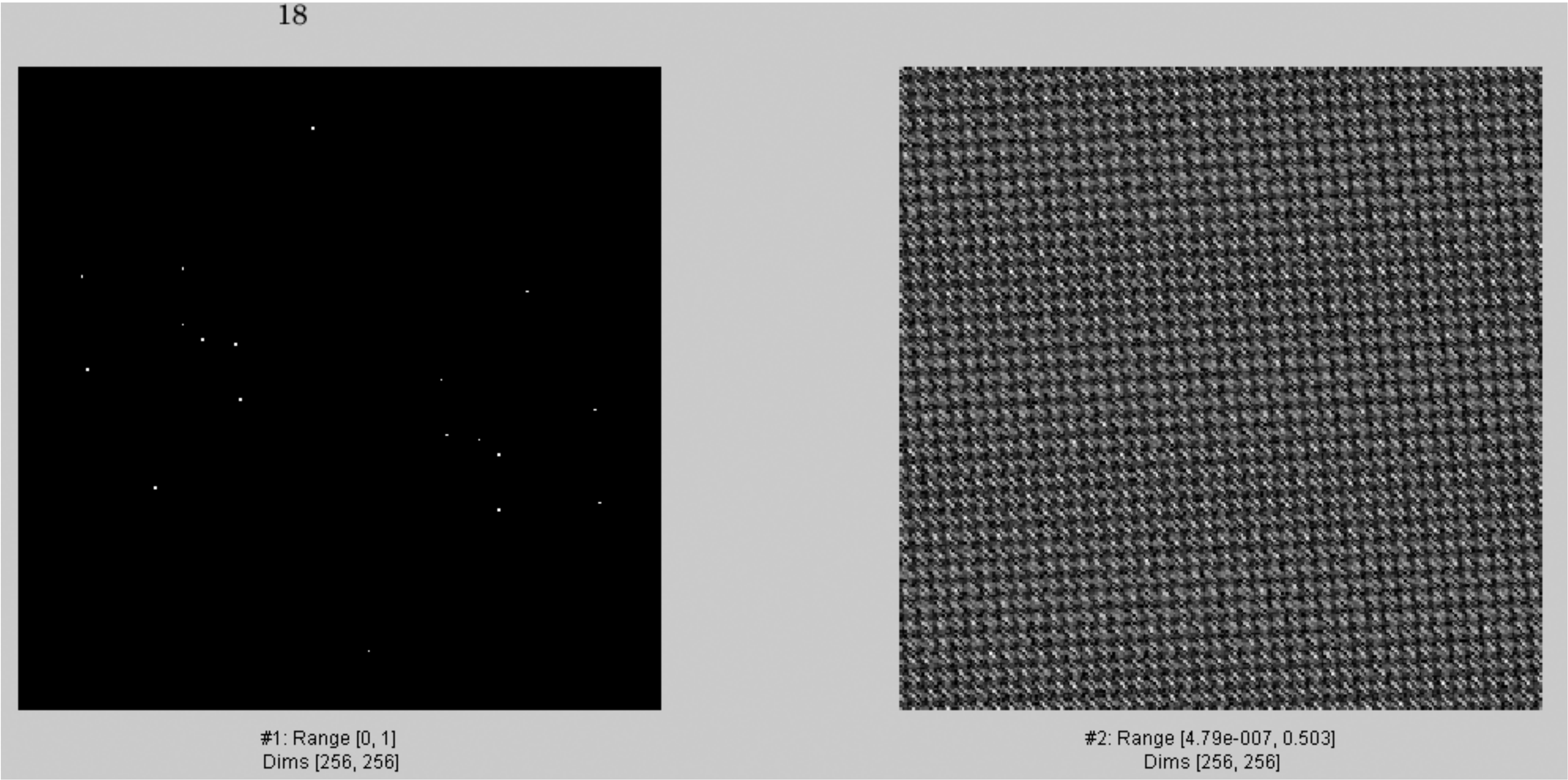
2



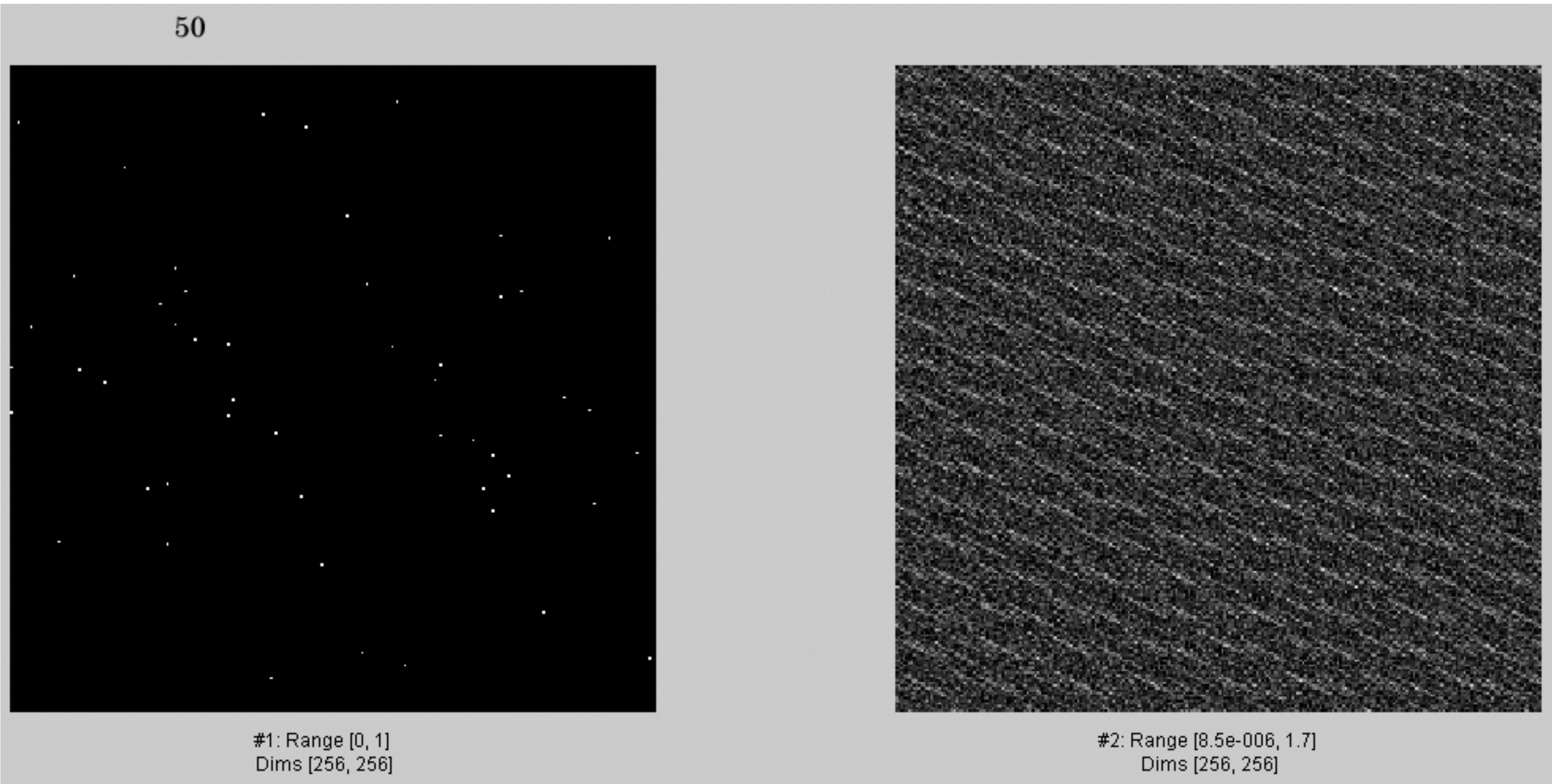
6



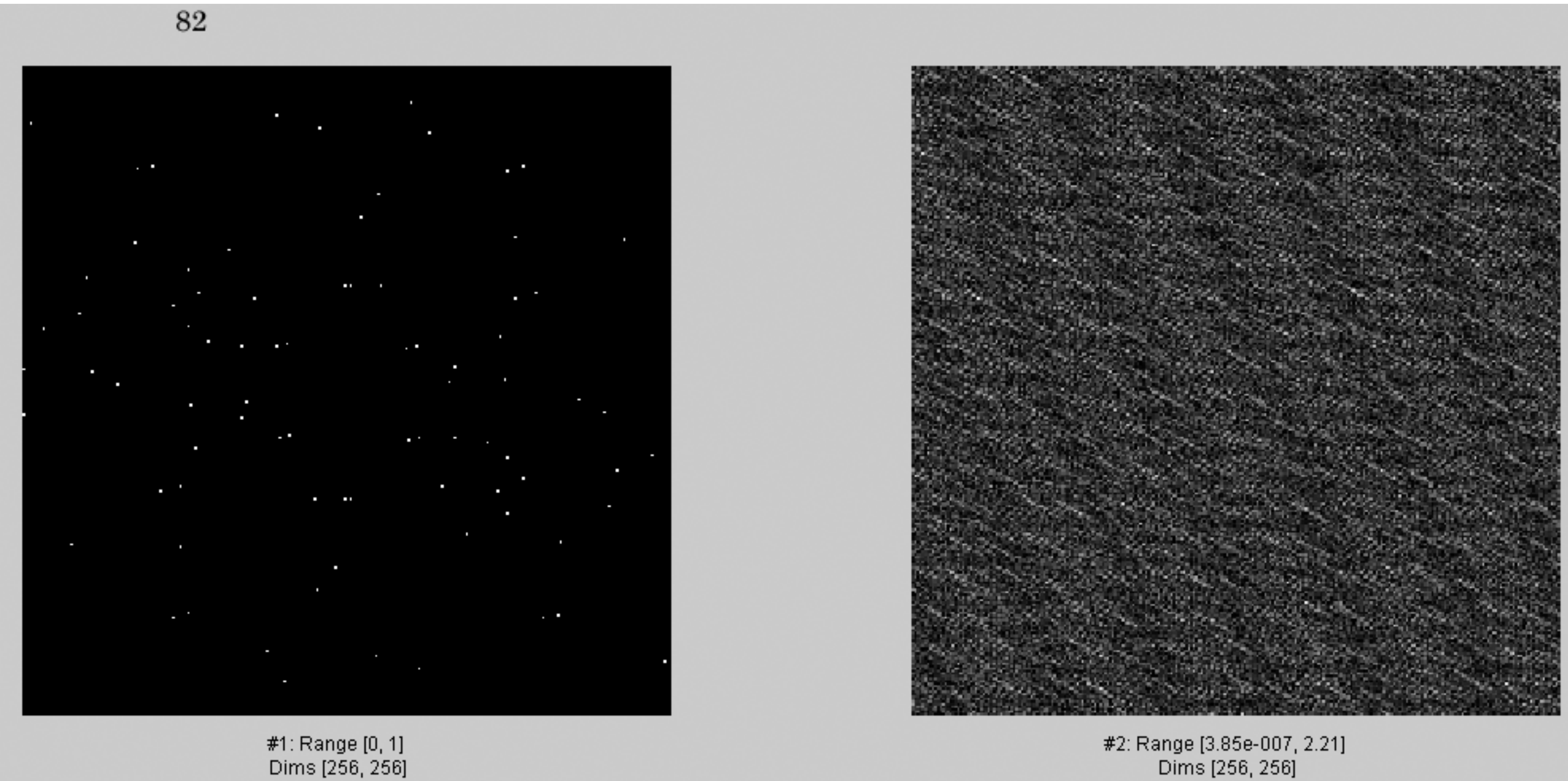
18



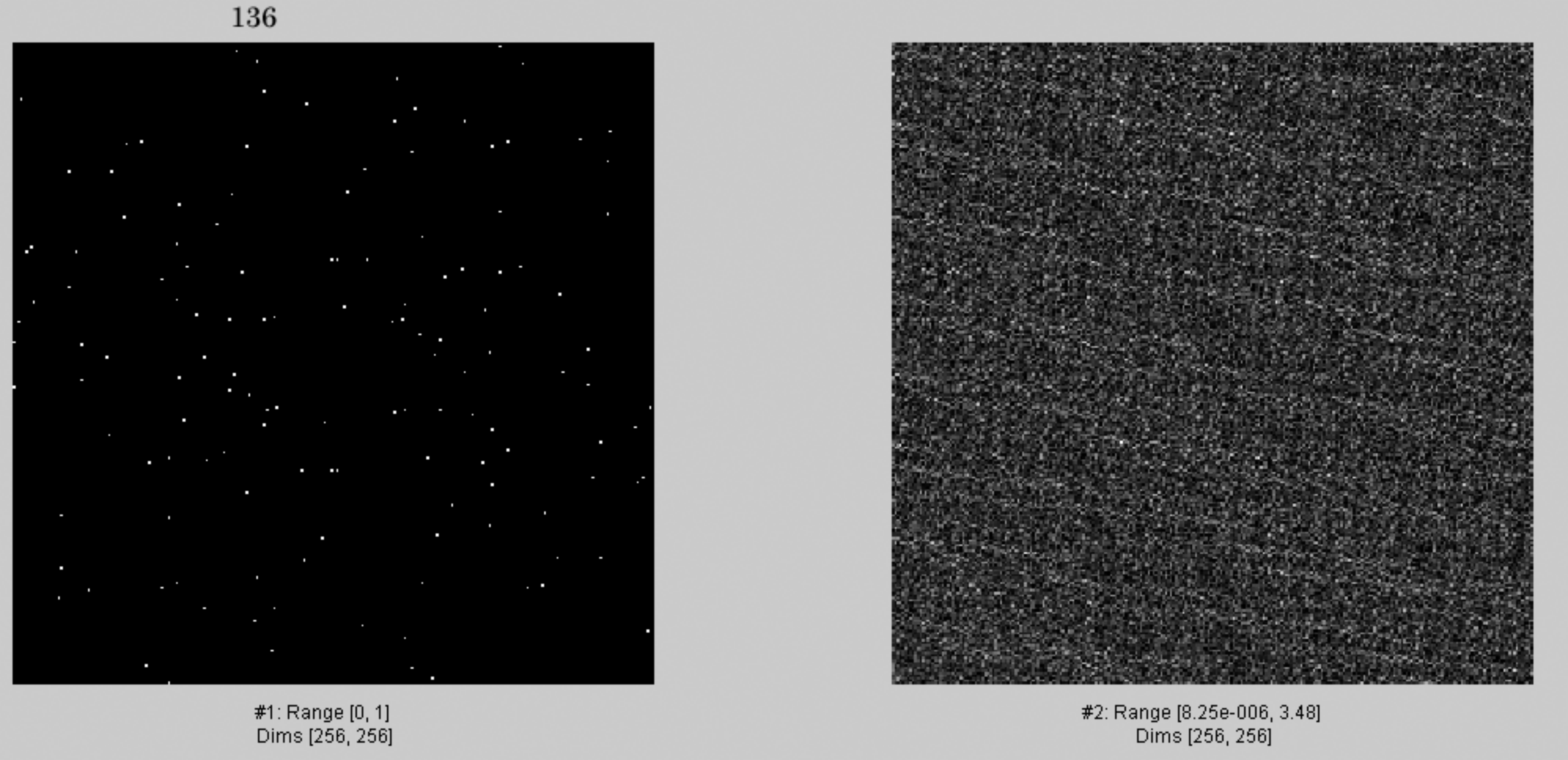
50



82

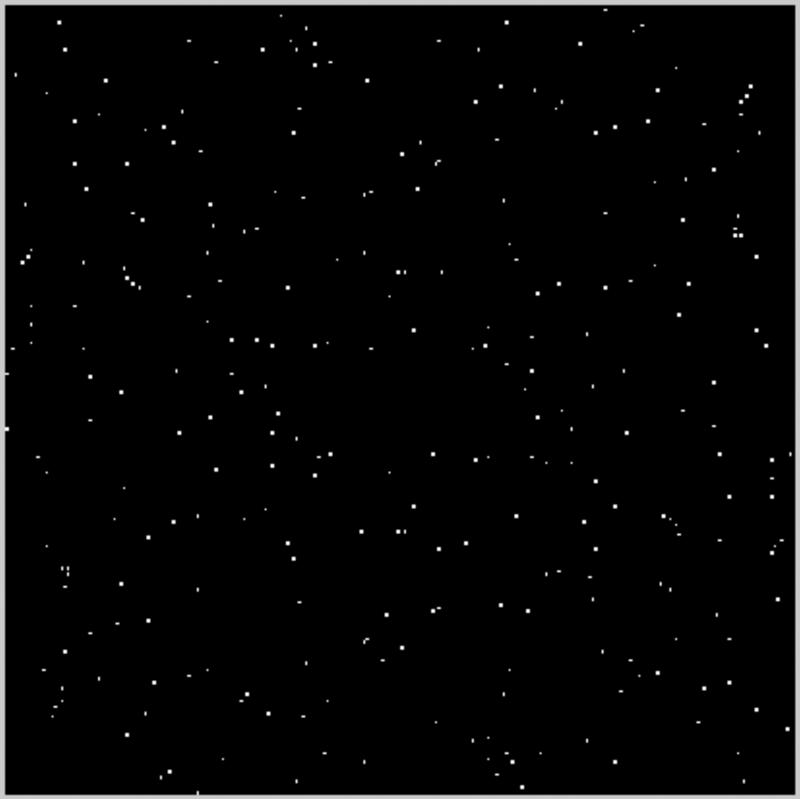


136

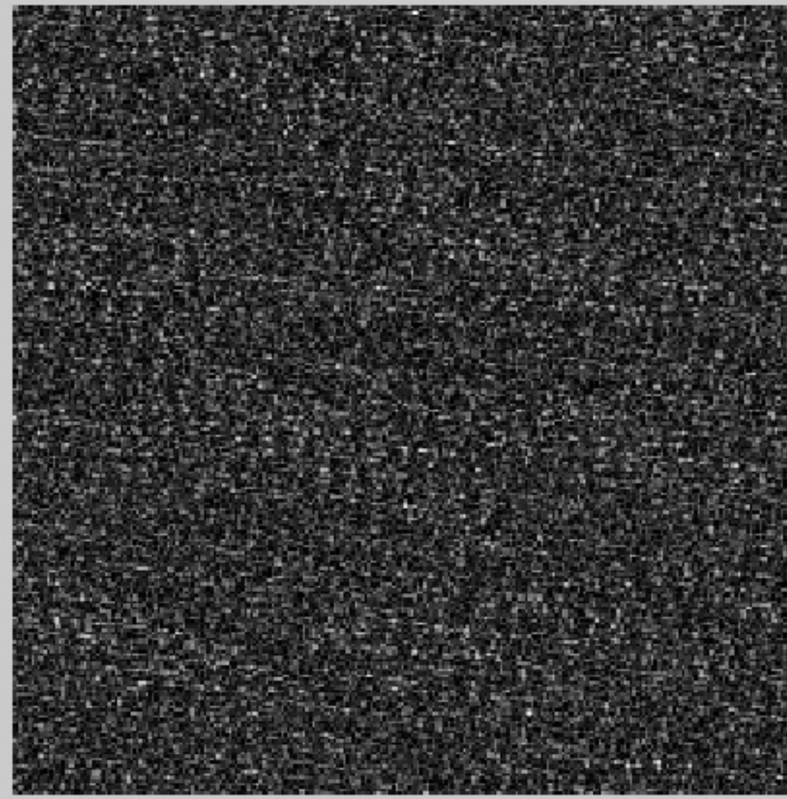


282

282

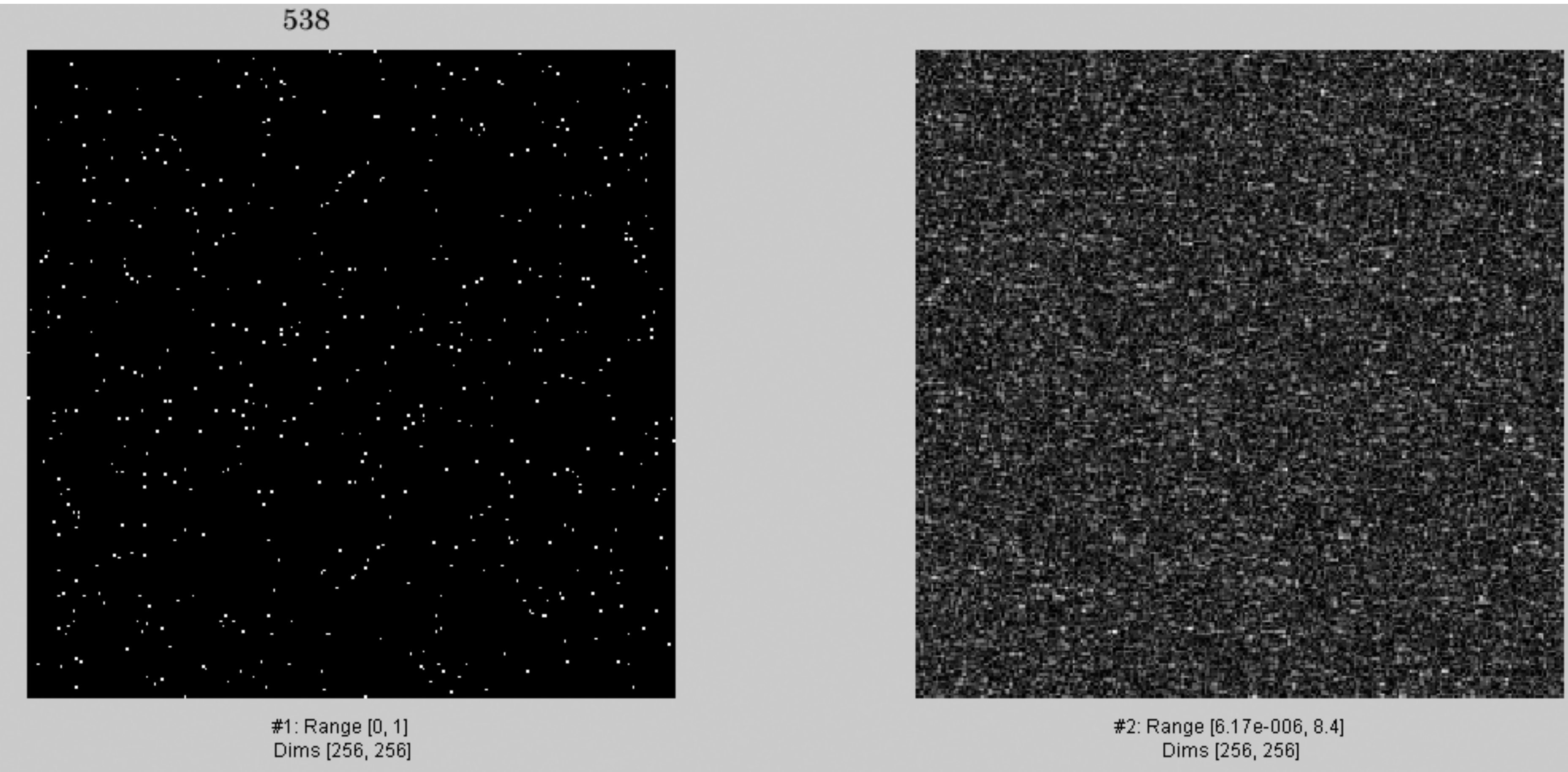


#1: Range [0, 1]
Dims [256, 256]

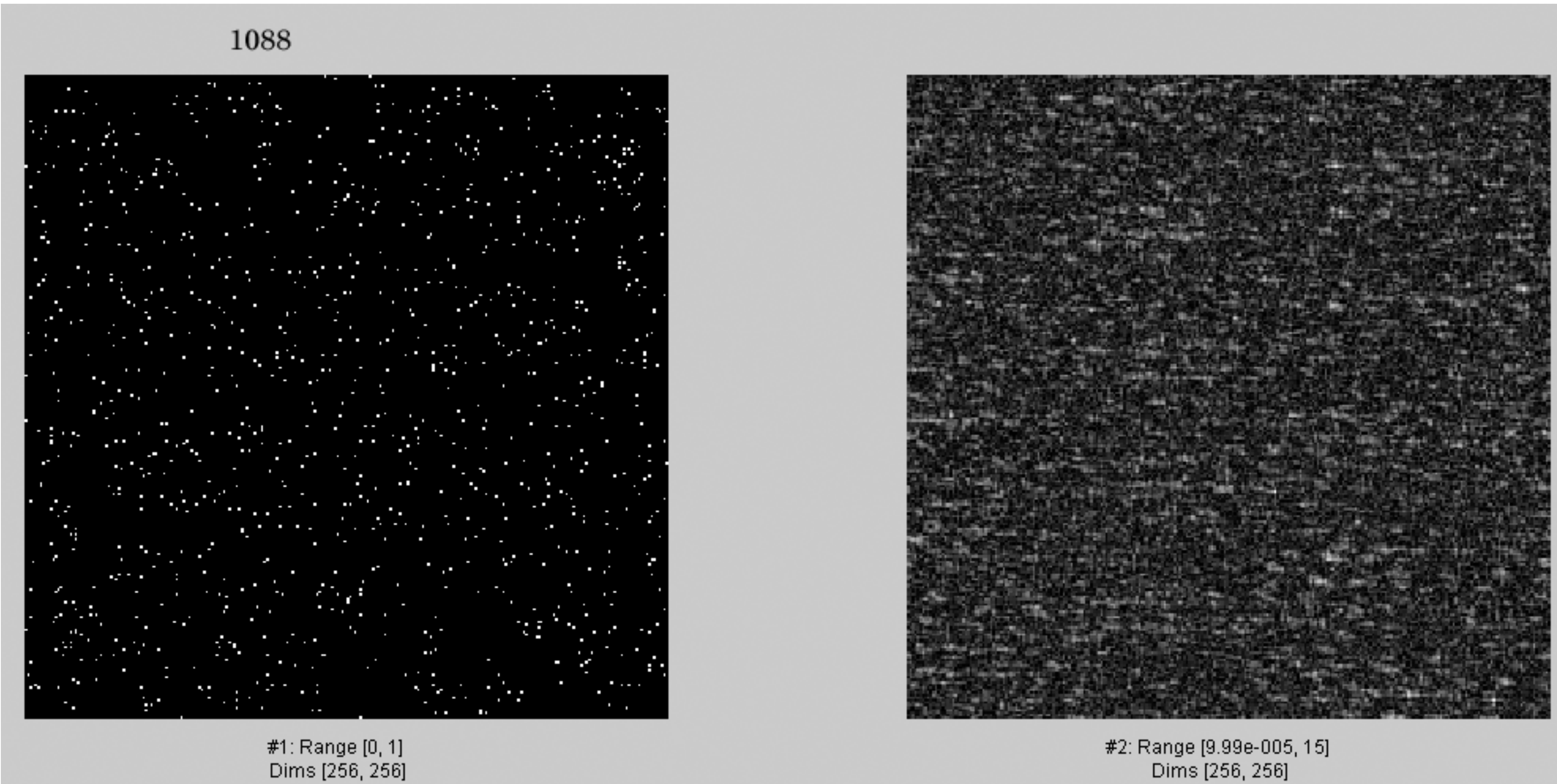


#2: Range [1.39e-005, 5.88]
Dims [256, 256]

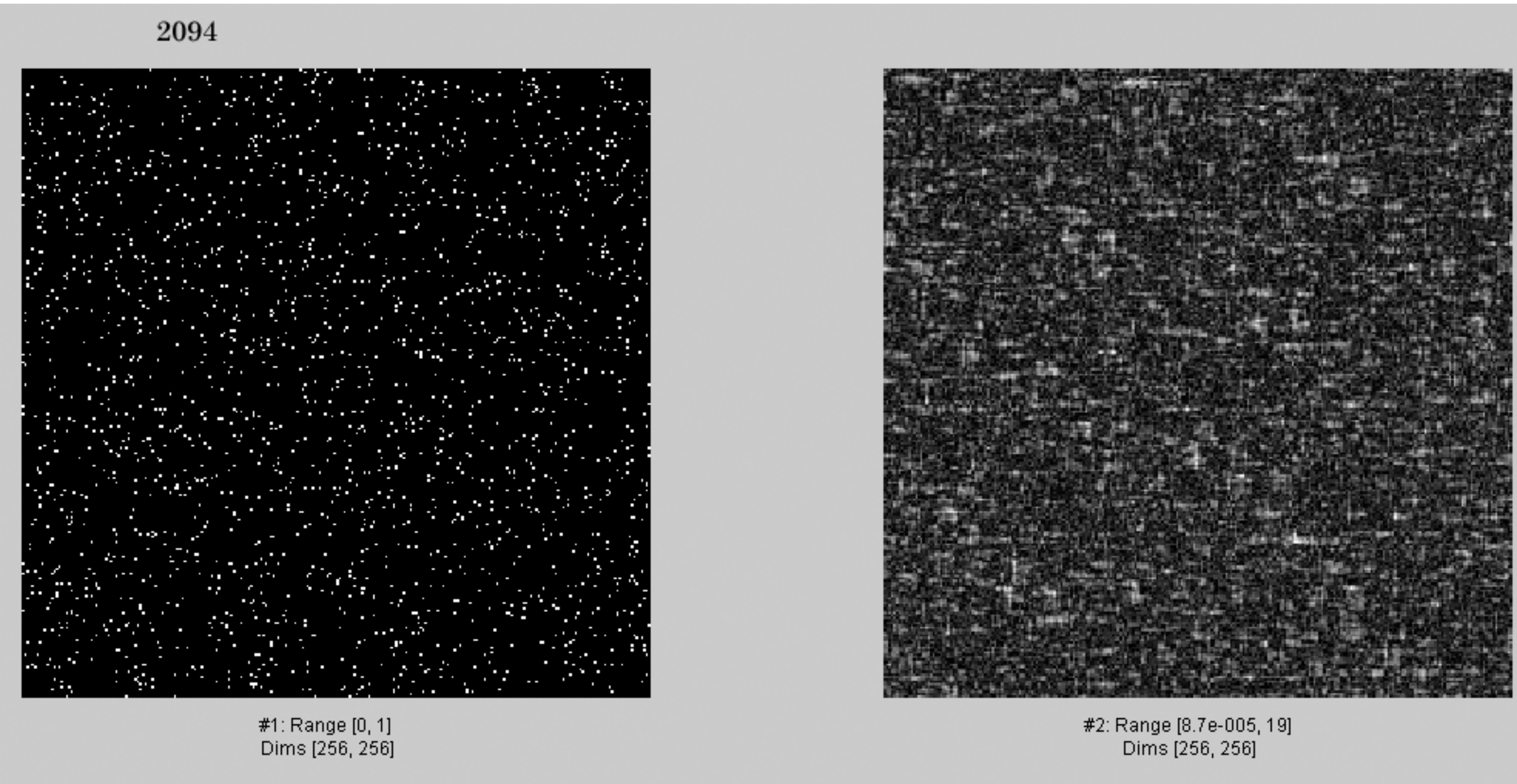
538



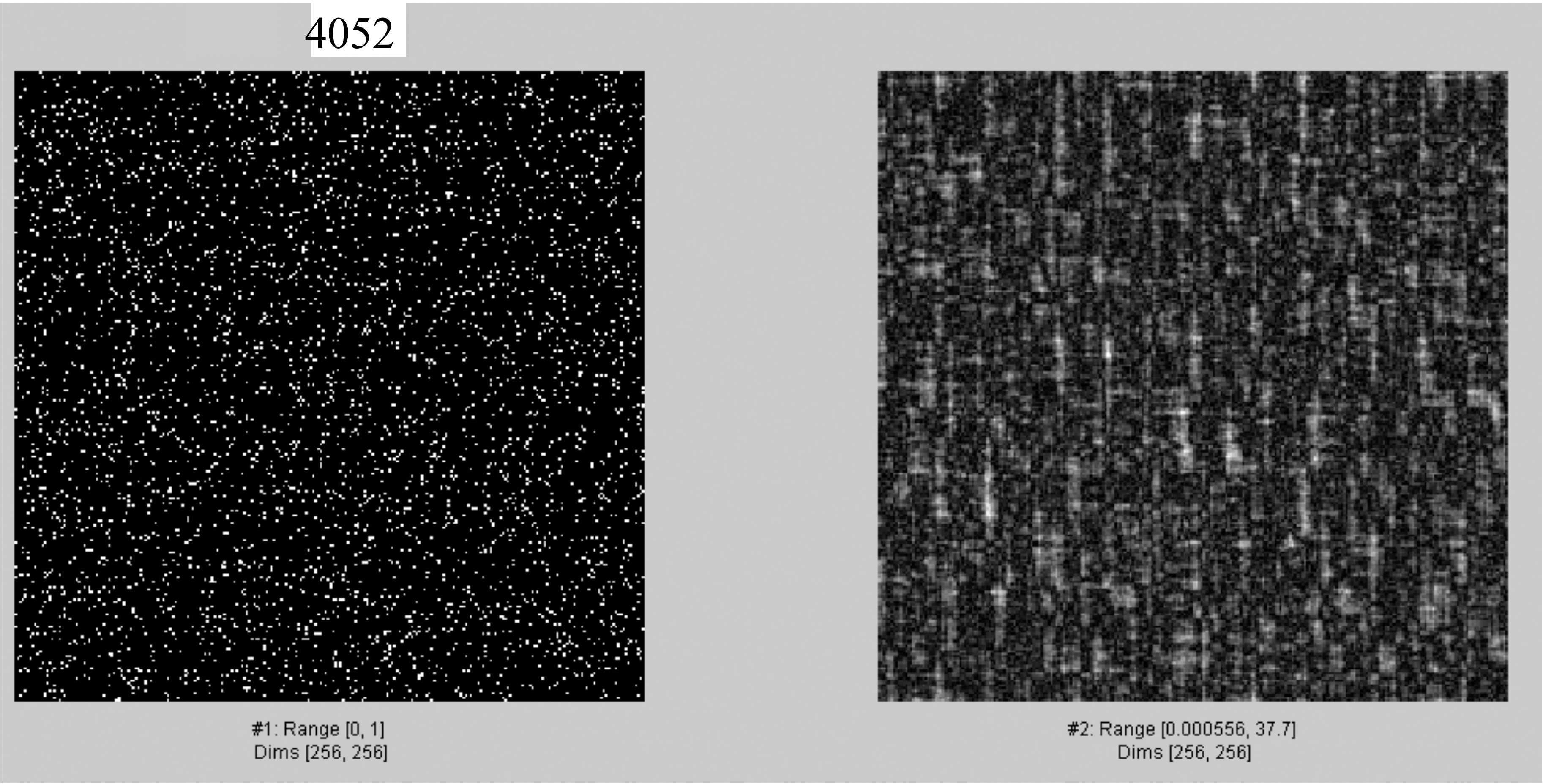
1088



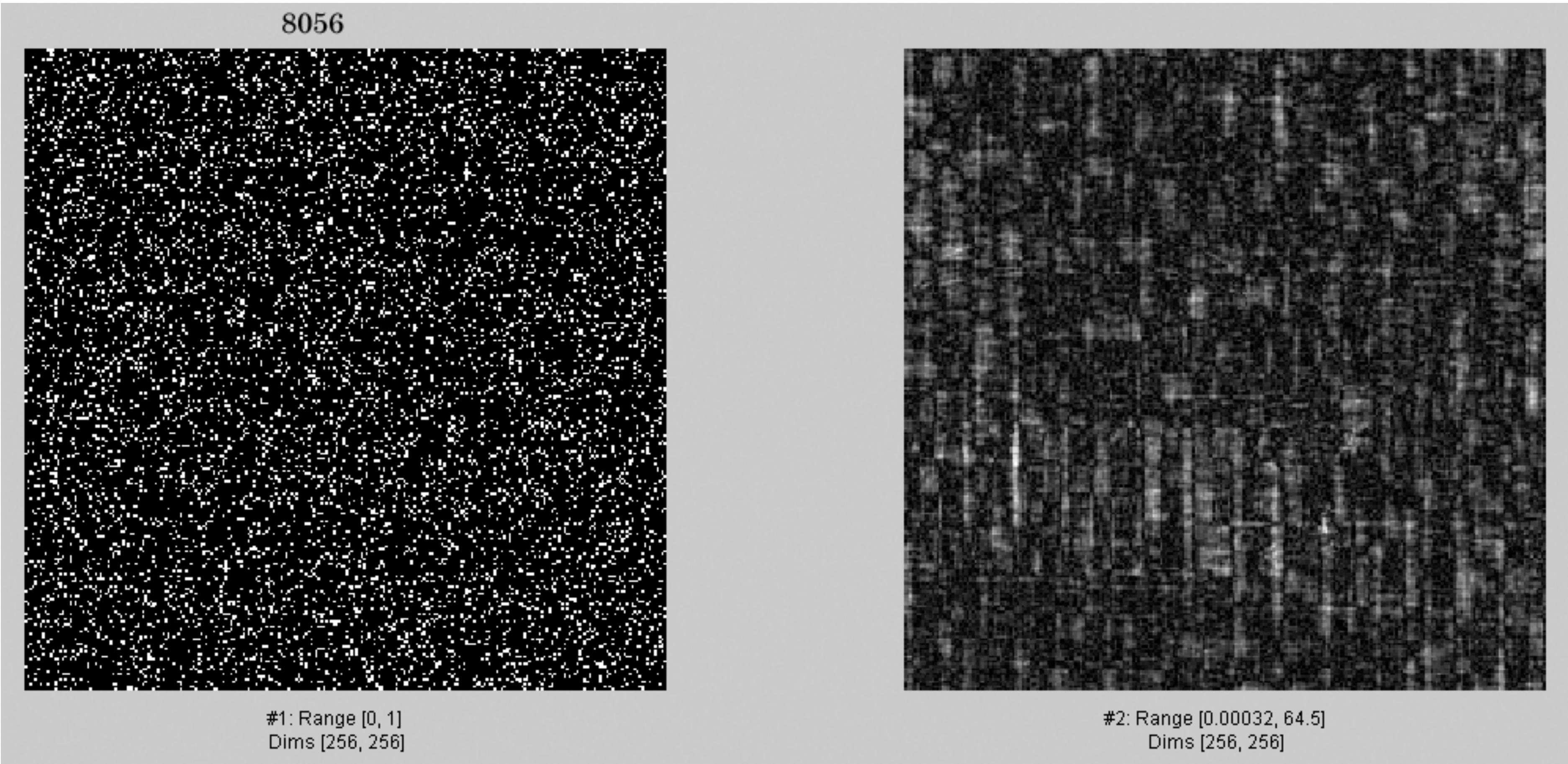
2094



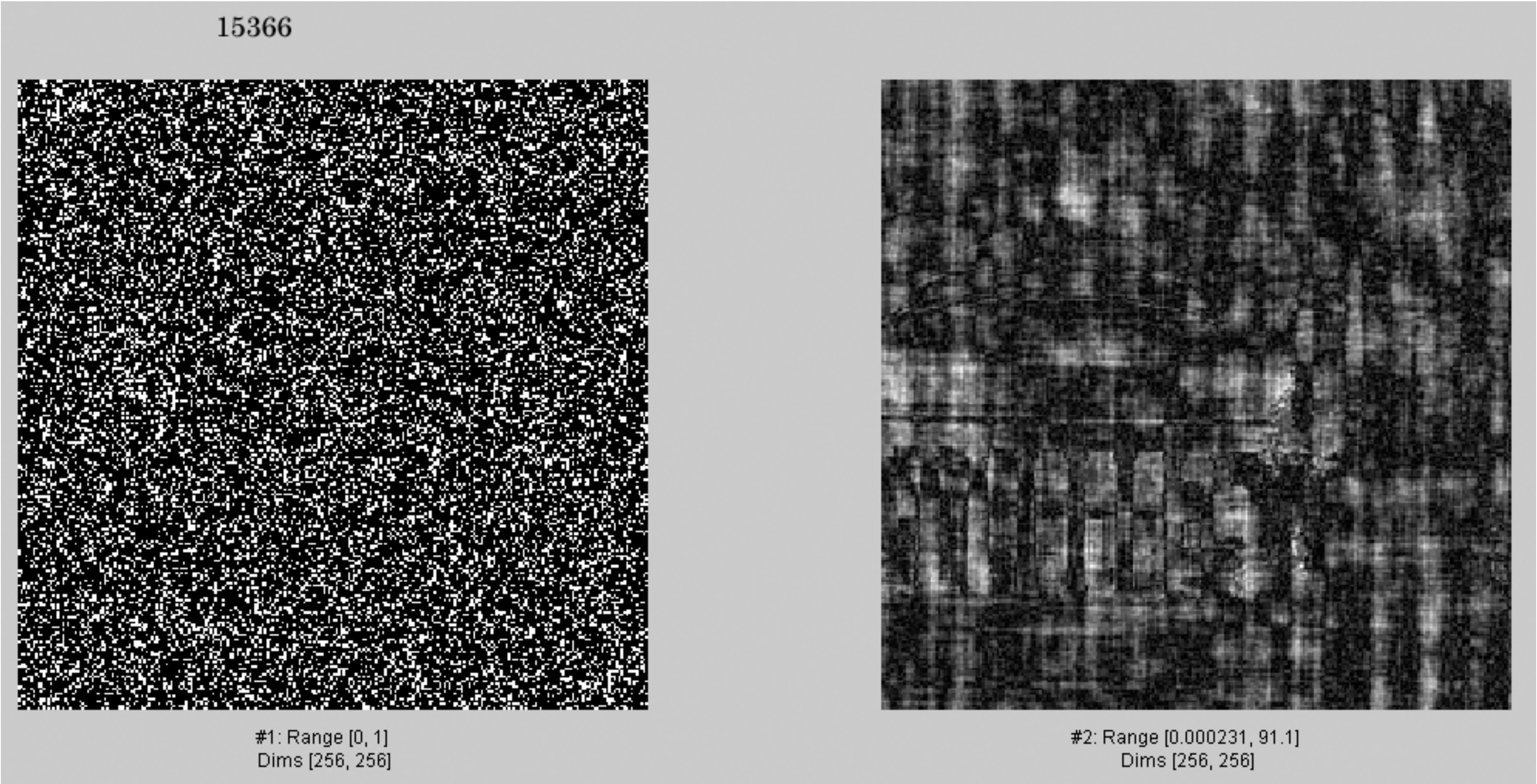
4052.



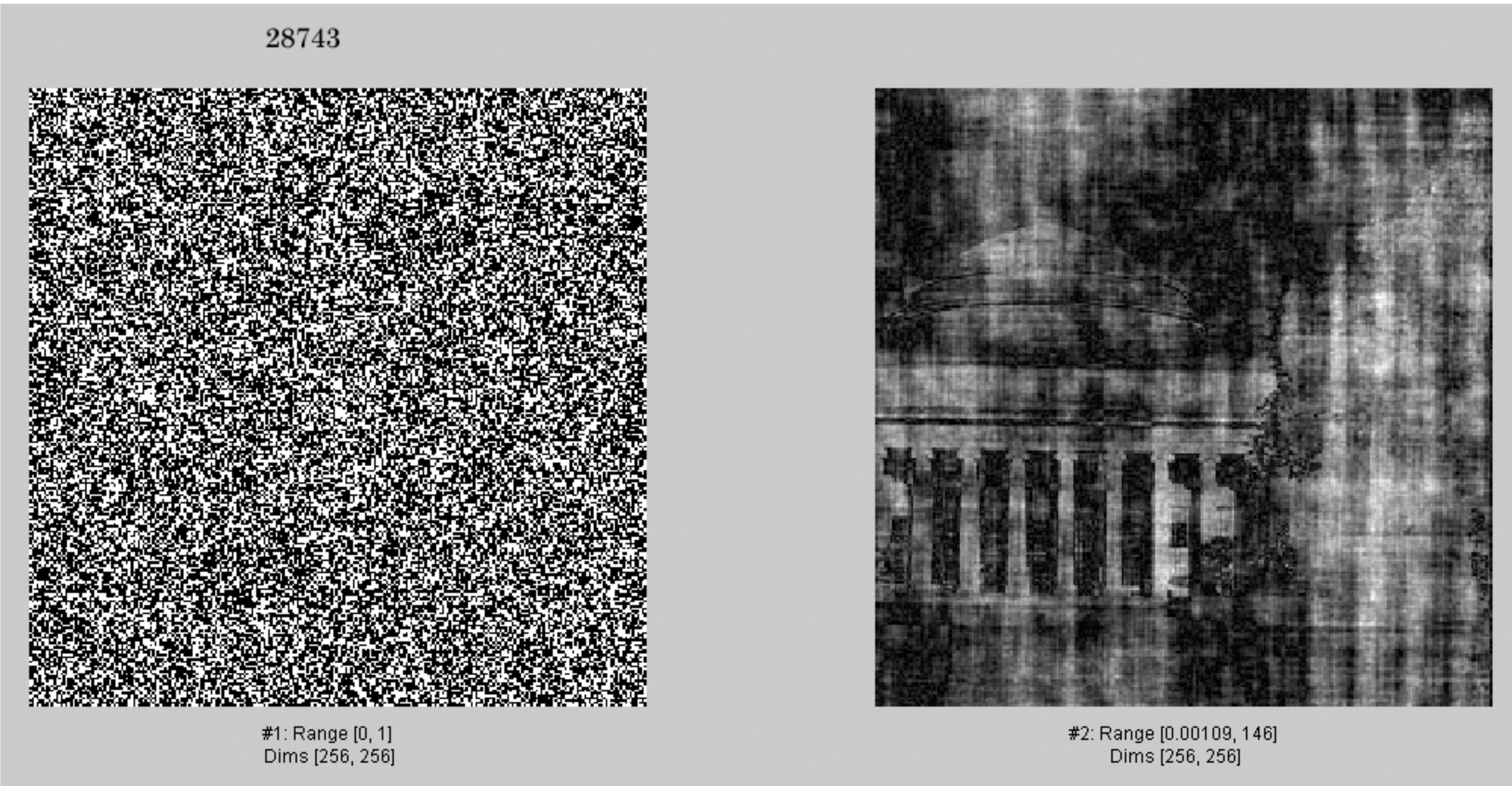
8056.



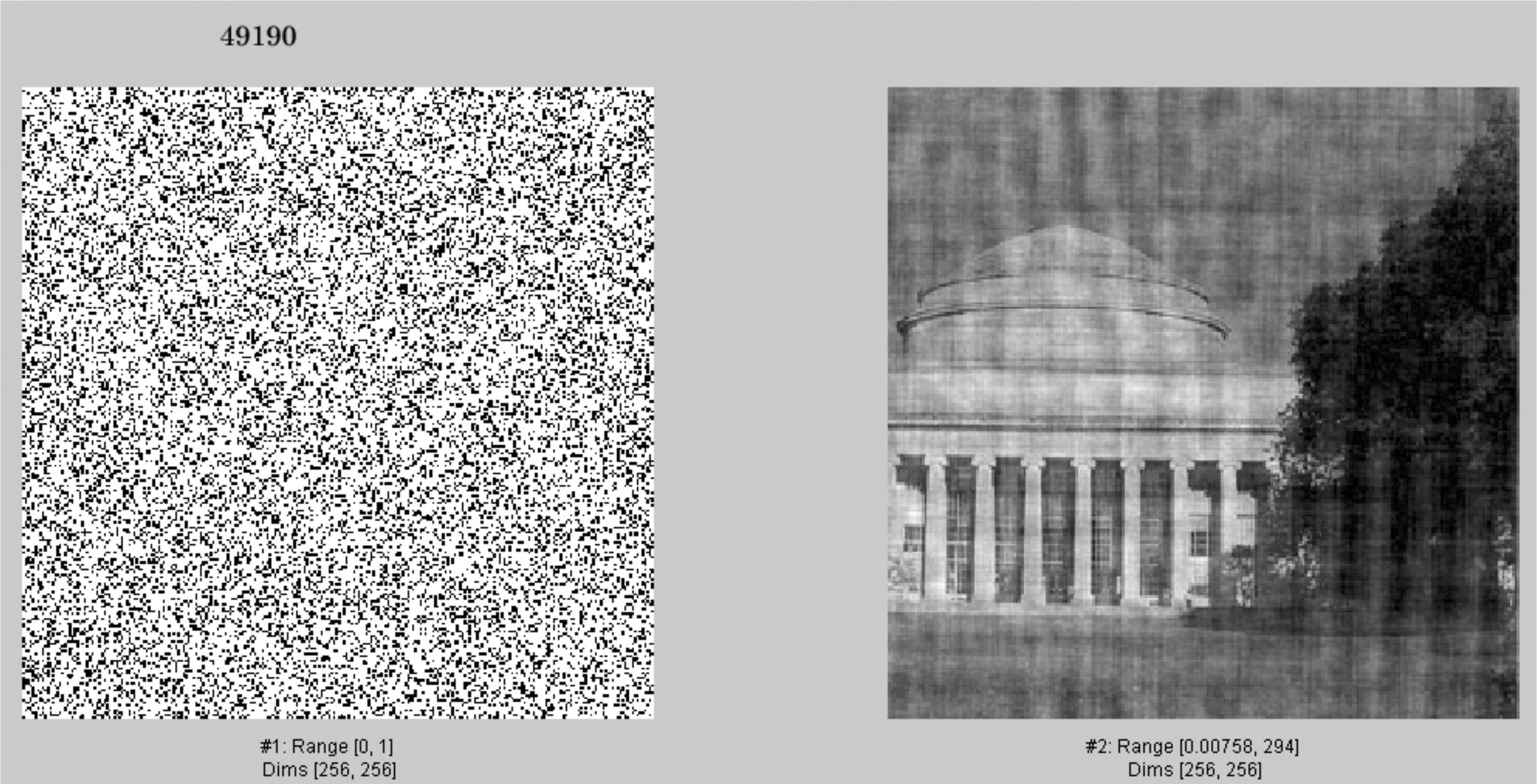
15366



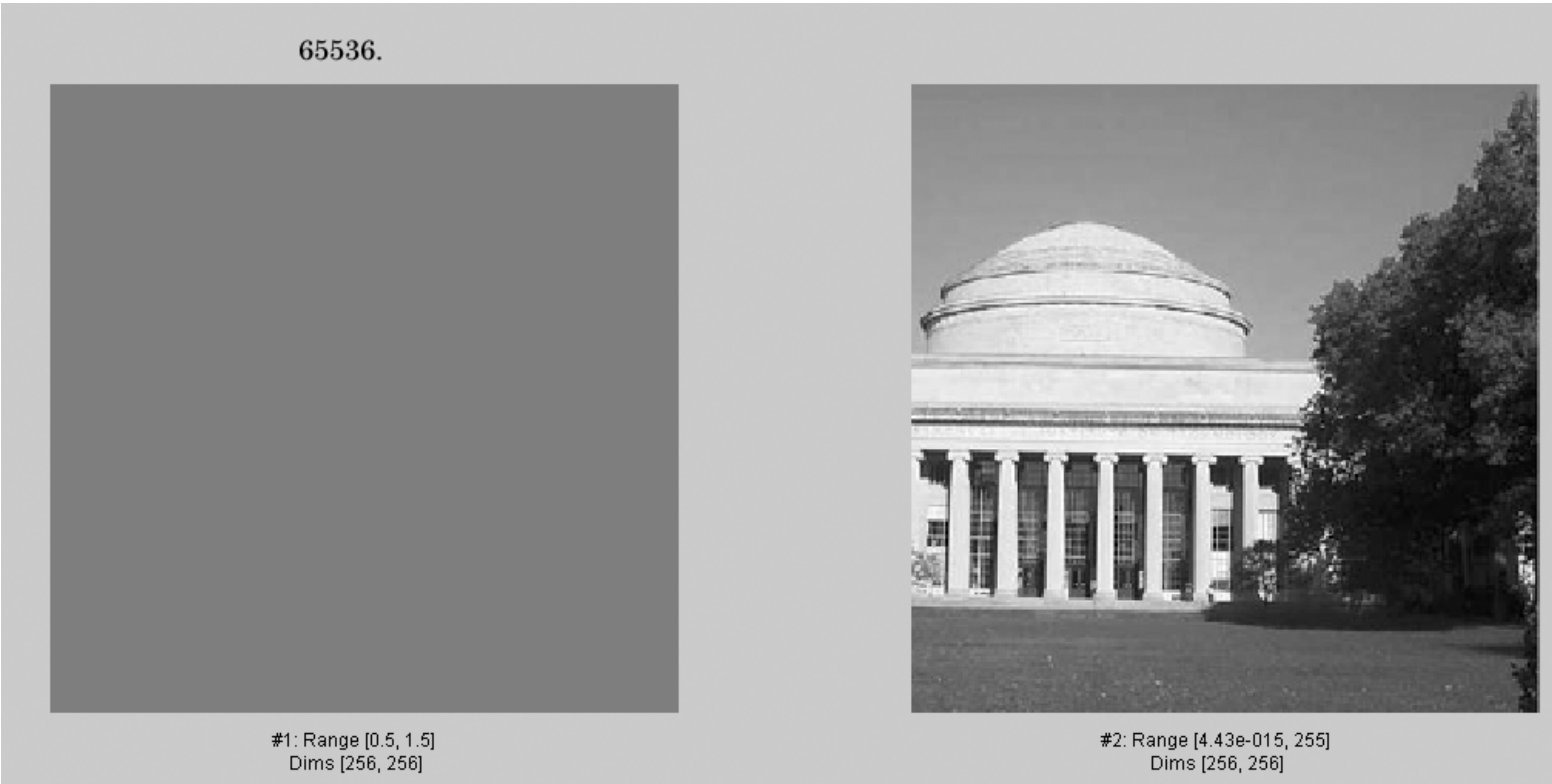
28743



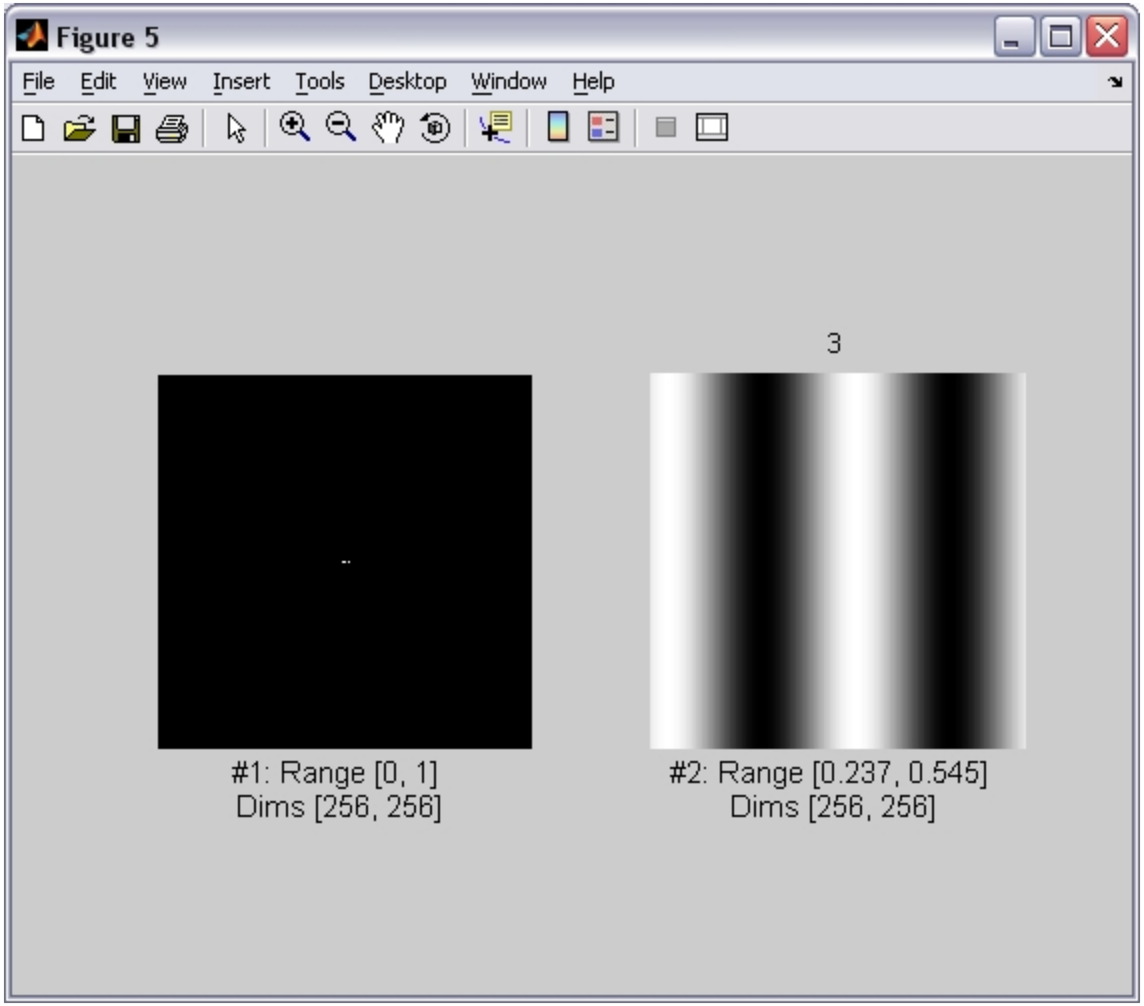
49190.



65536.


65536.
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [4.43e-015, 255]
Dims [256, 256]

3



Now, an analogous sequence of images, but selecting Fourier components in descending order of magnitude.

5

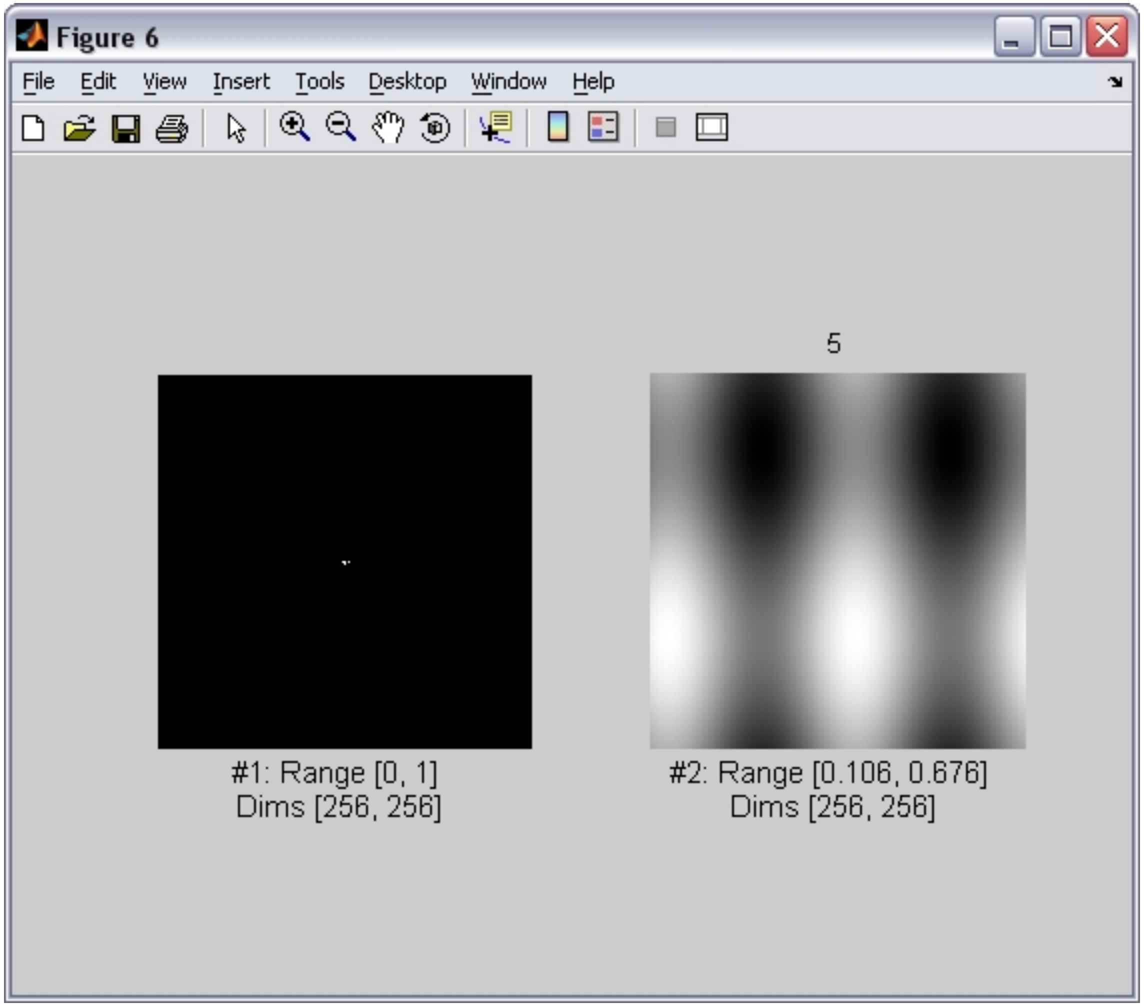


Figure 6
File
Edit
View
Insert
Tools
Desktop
Window
Help
5
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.106, 0.676]
Dims [256, 256]

9

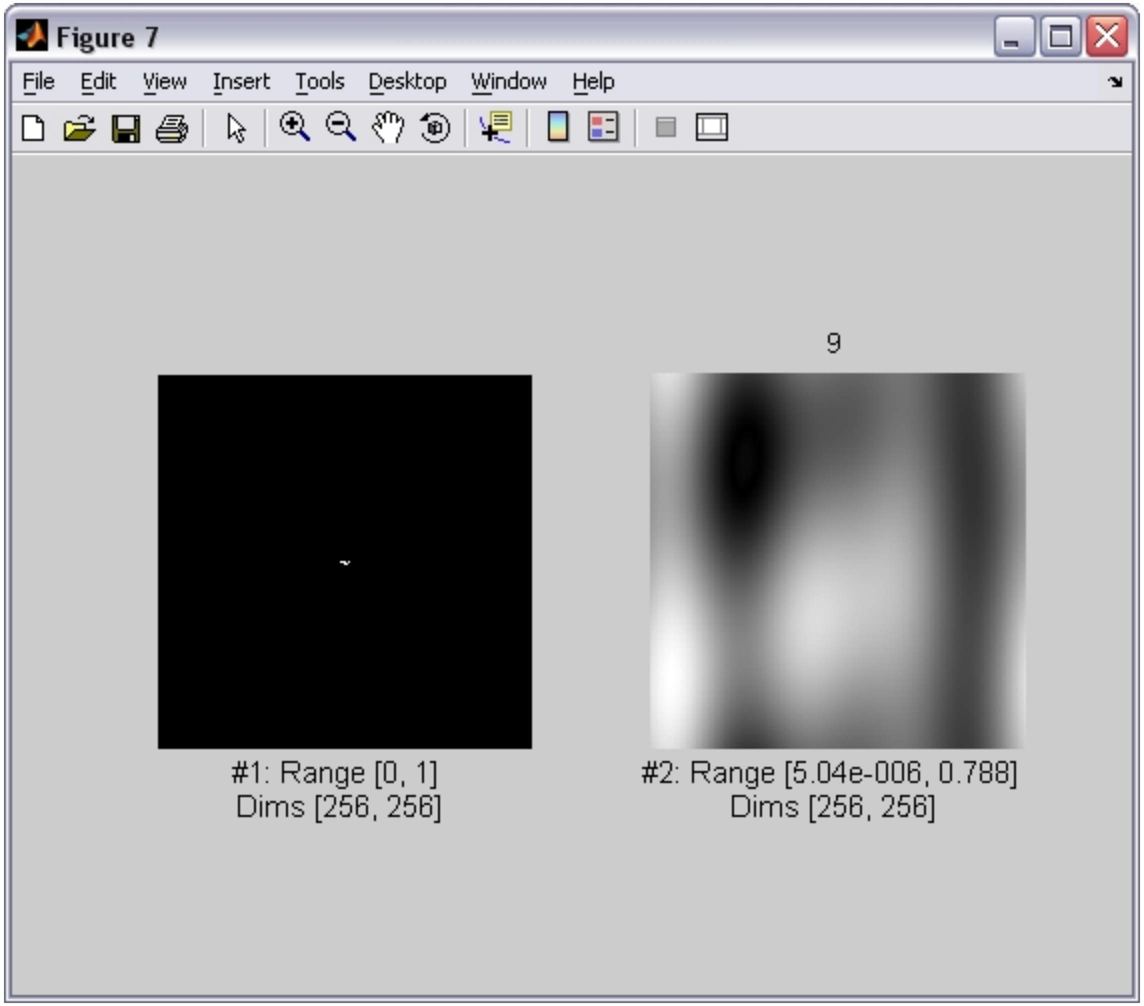


Figure 7
File Edit View Insert Tools Desktop Window Help
9
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.04e-006, 0.788]
Dims [256, 256]

17

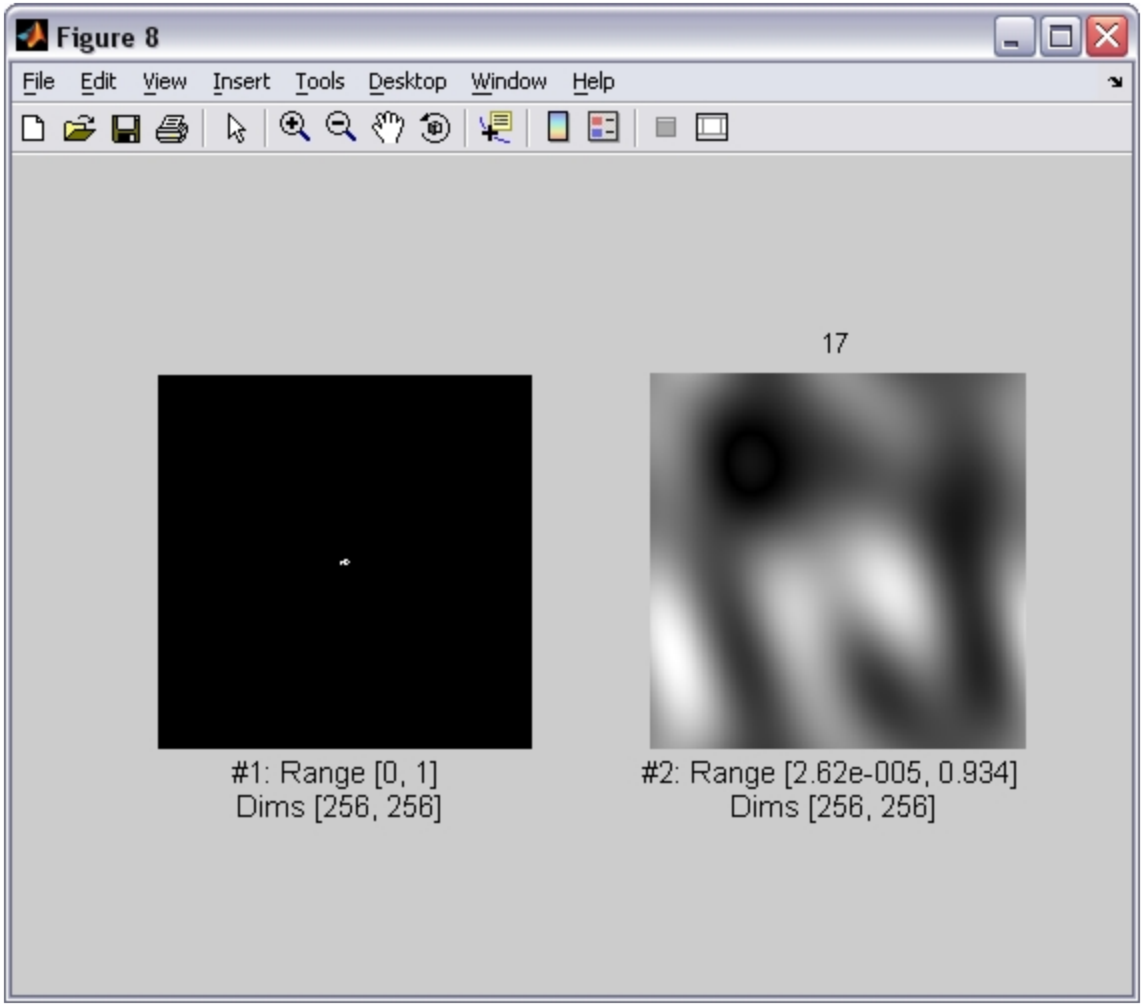


Figure 8
File Edit View Insert Tools Desktop Window Help
17
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.62e-005, 0.934]
Dims [256, 256]

33

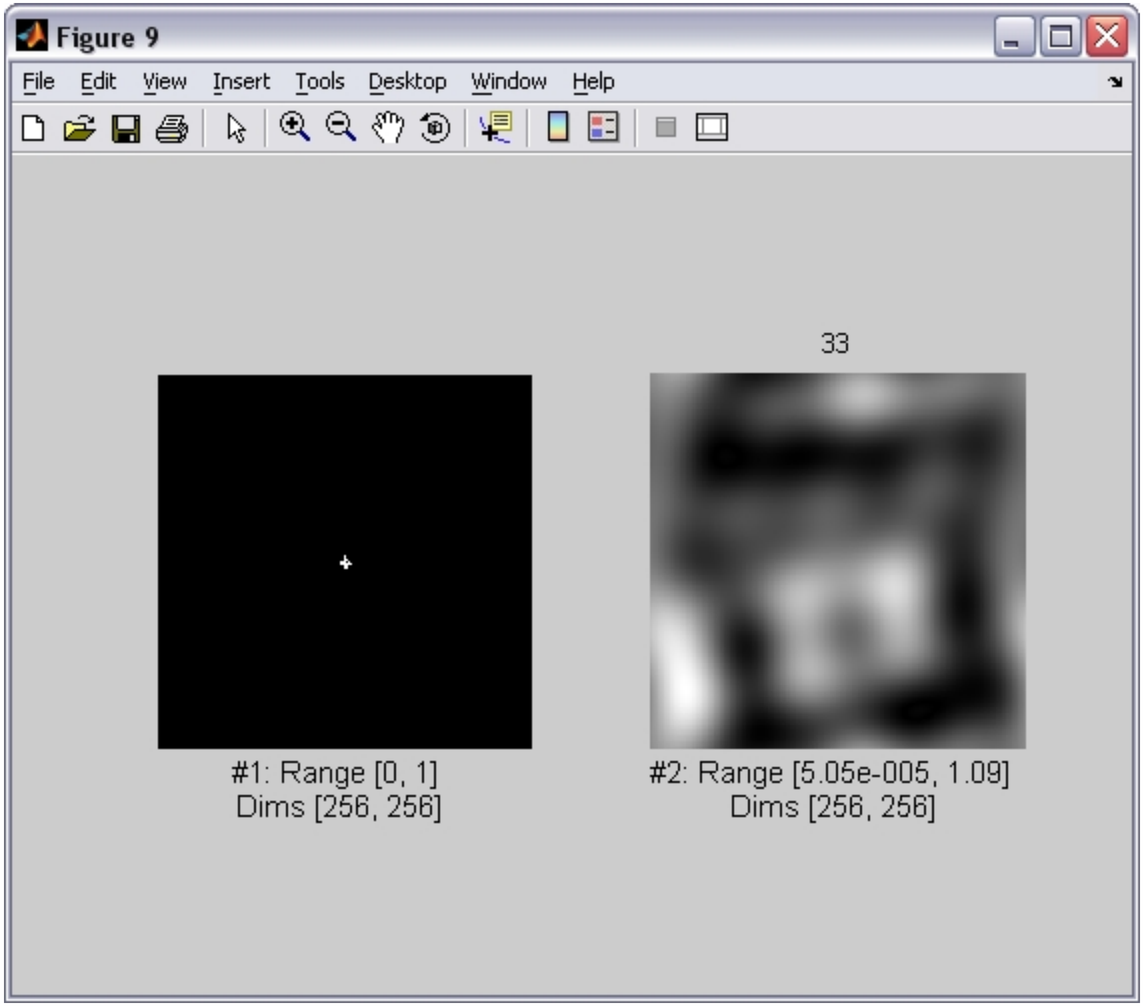


Figure 9
File Edit View Insert Tools Desktop Window Help
33
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.05e-005, 1.09]
Dims [256, 256]

65

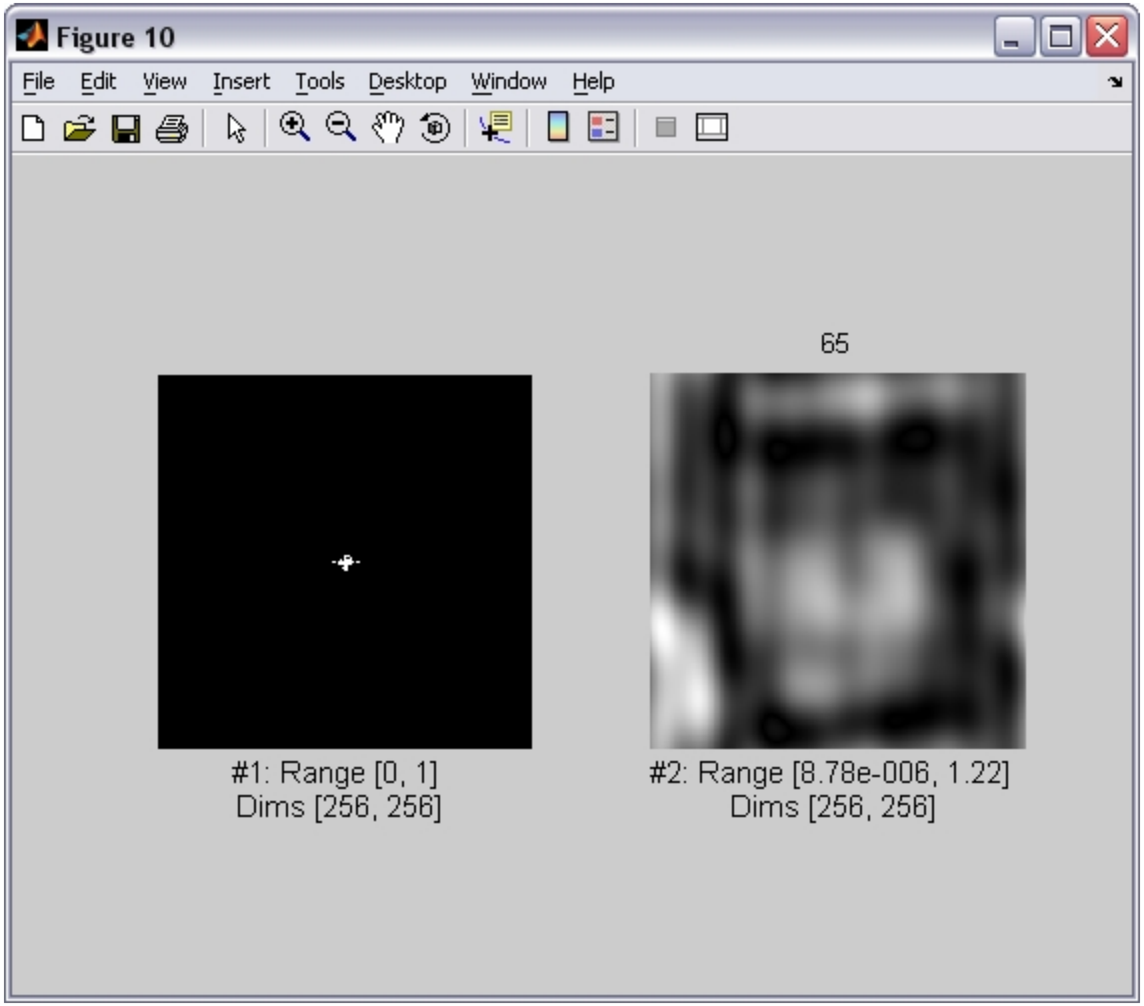


Figure 10
File Edit View Insert Tools Desktop Window Help
65
#1: Range [0, 1]
Dims [256, 256]
#2: Range [8.78e-006, 1.22]
Dims [256, 256]

129

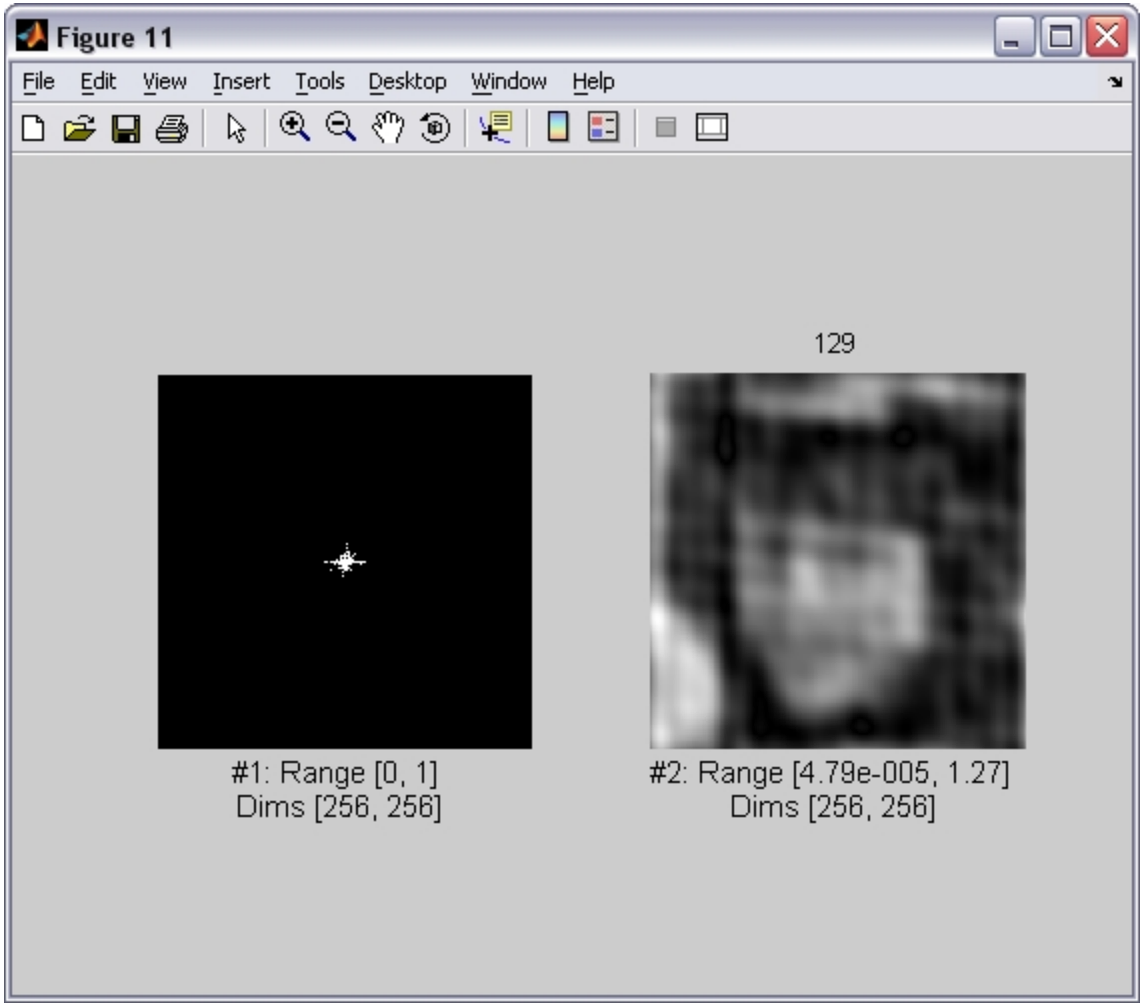


Figure 11
File Edit View Insert Tools Desktop Window Help
129
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.79e-005, 1.27]
Dims [256, 256]

257

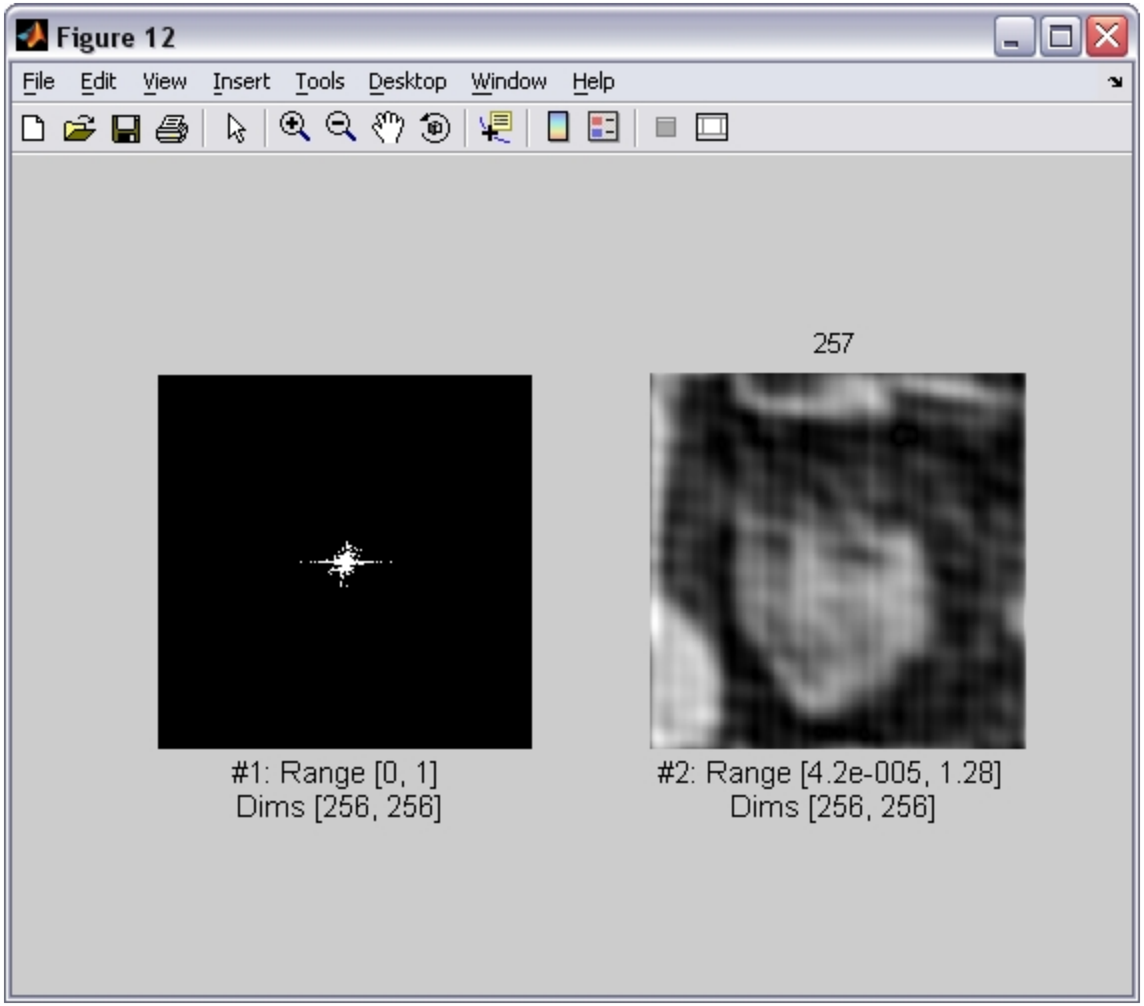


Figure 12
File Edit View Insert Tools Desktop Window Help
257
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.2e-005, 1.28]
Dims [256, 256]

513

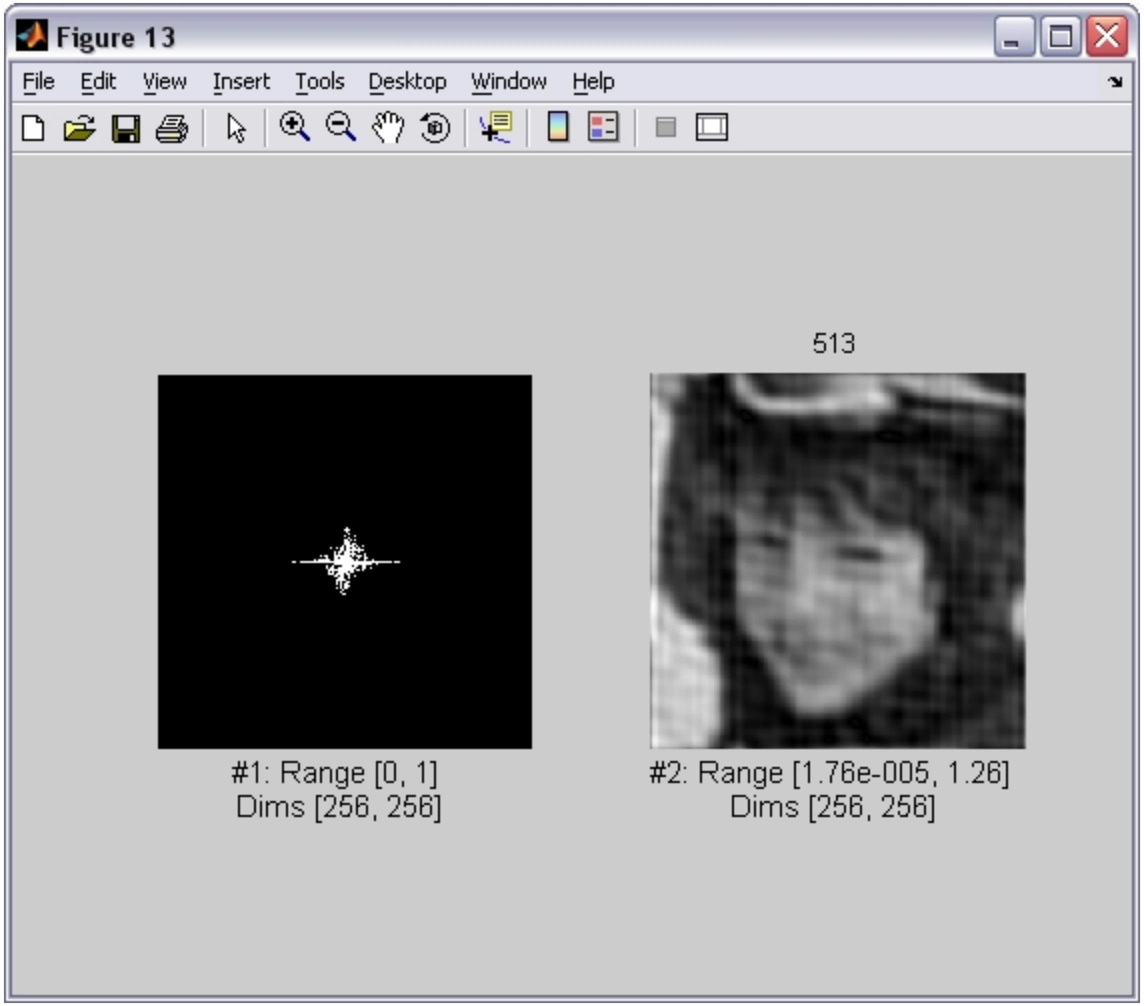


Figure 13
File Edit View Insert Tools Desktop Window Help
513
#1: Range [0, 1]
Dims [256, 256]
#2: Range [1.76e-005, 1.26]
Dims [256, 256]

1025

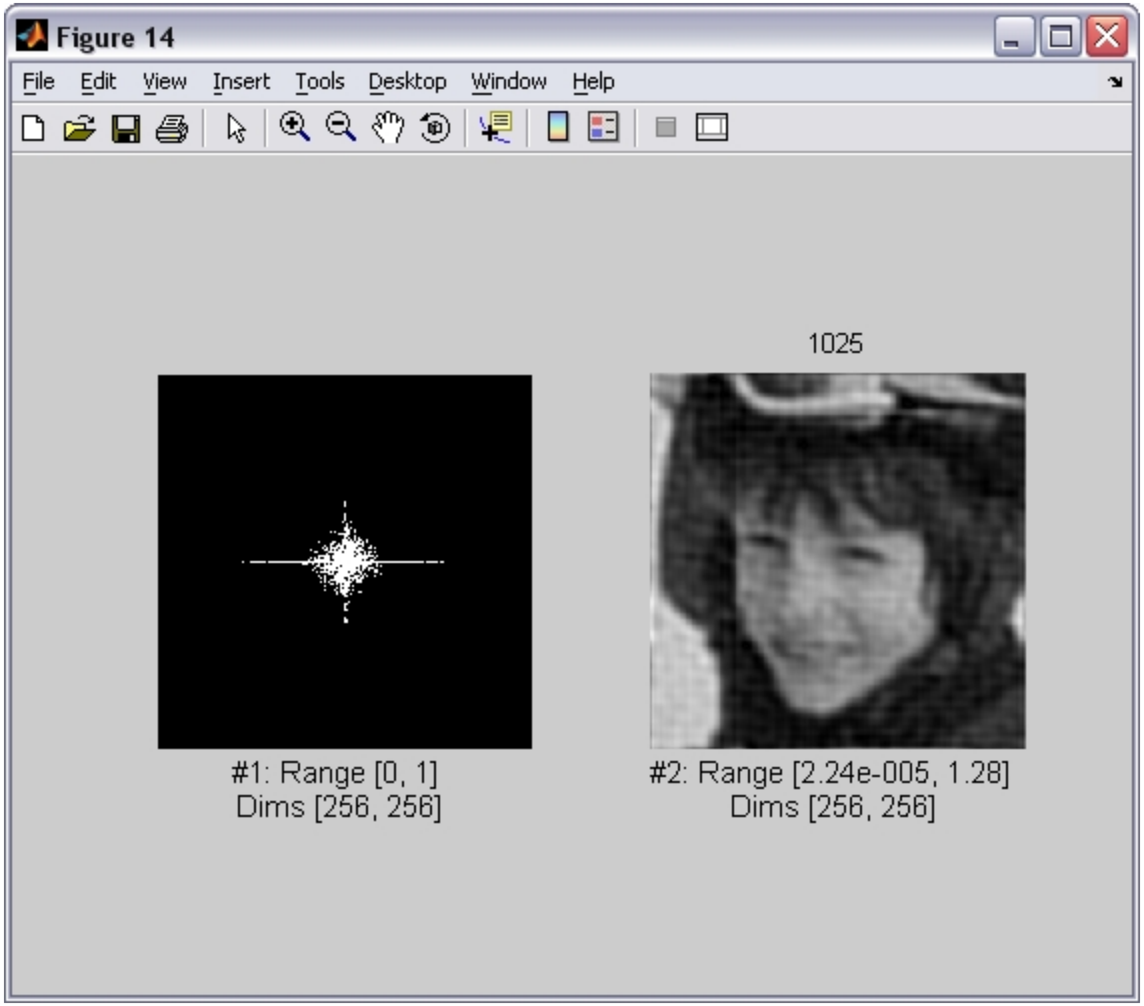


Figure 14
File Edit View Insert Tools Desktop Window Help
1025
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.24e-005, 1.28]
Dims [256, 256]

2049

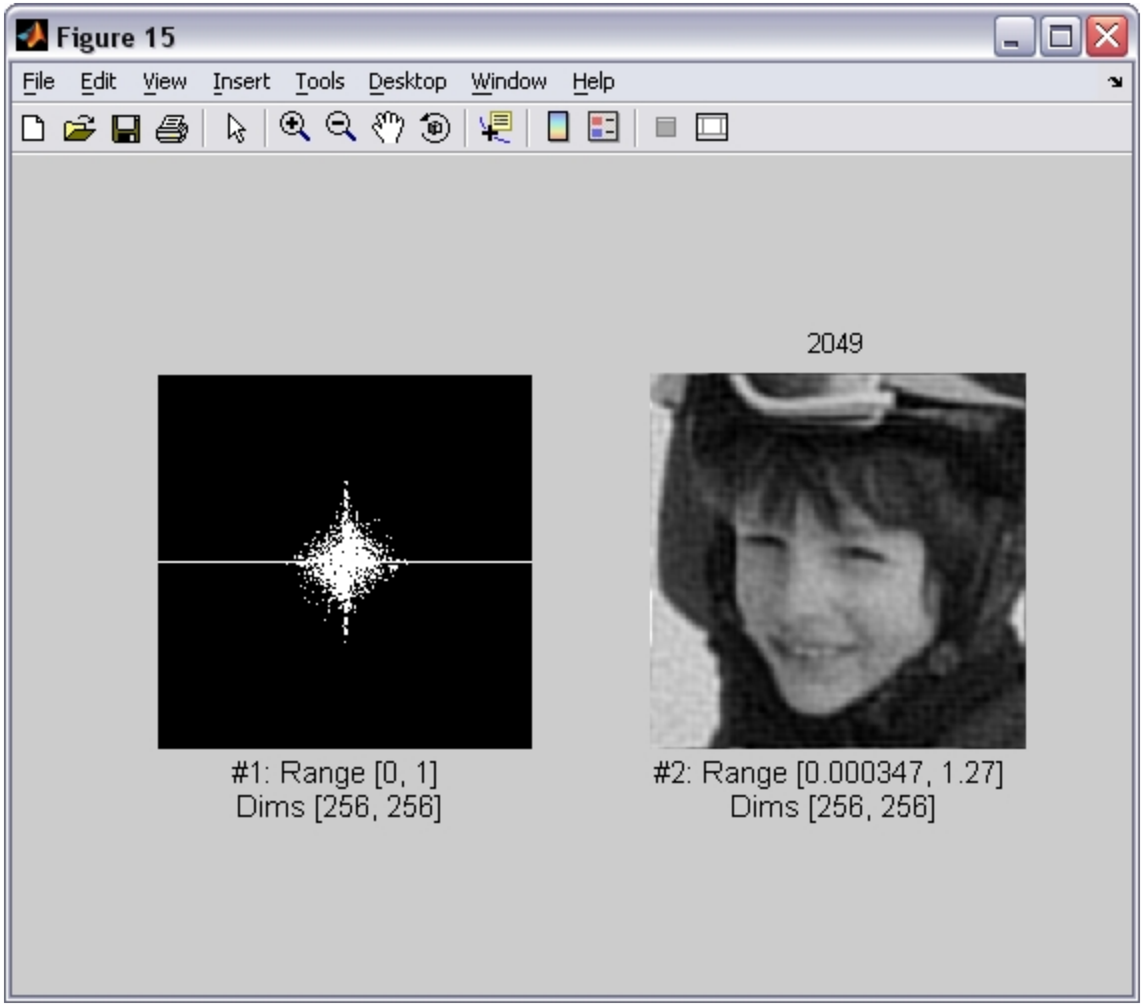


Figure 15
File Edit View Insert Tools Desktop Window Help
2049
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000347, 1.27]
Dims [256, 256]

4097

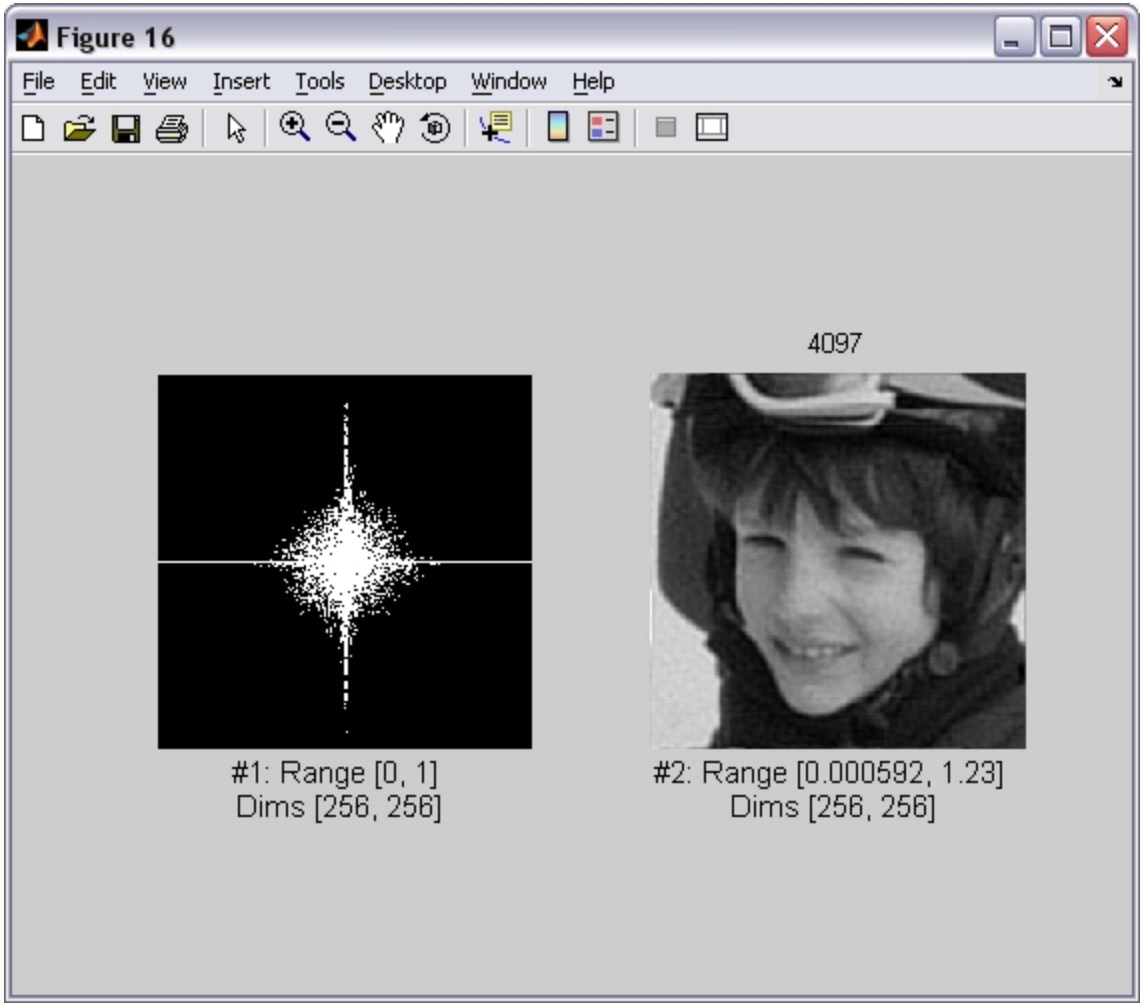


Figure 16
File
Edit
View
Insert
Tools
Desktop
Window
Help
4097
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000592, 1.23]
Dims [256, 256]

8193

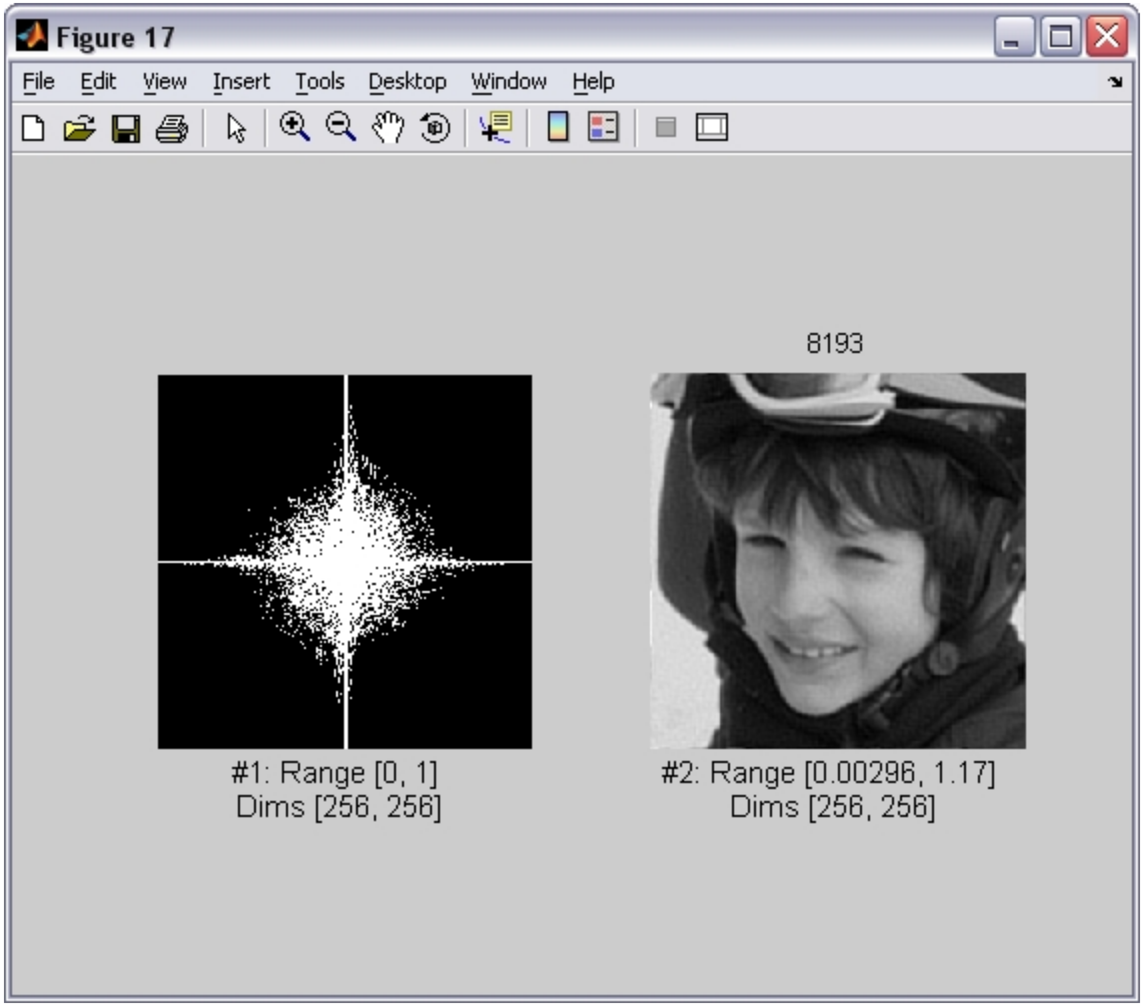


Figure 17
File Edit View Insert Tools Desktop Window Help
8193
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.00296, 1.17]
Dims [256, 256]

16385

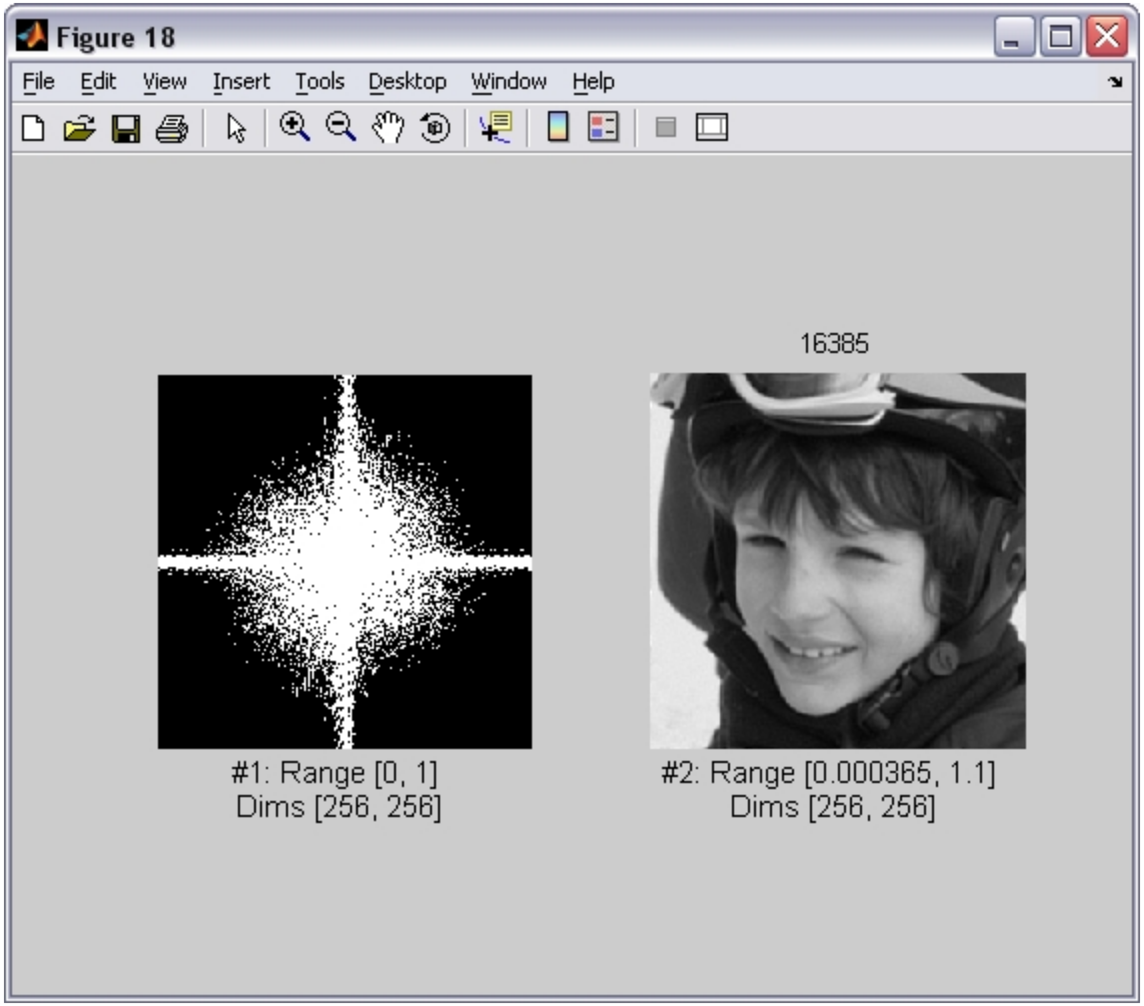


Figure 18
File Edit View Insert Tools Desktop Window Help
16385
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000365, 1.1]
Dims [256, 256]

32769

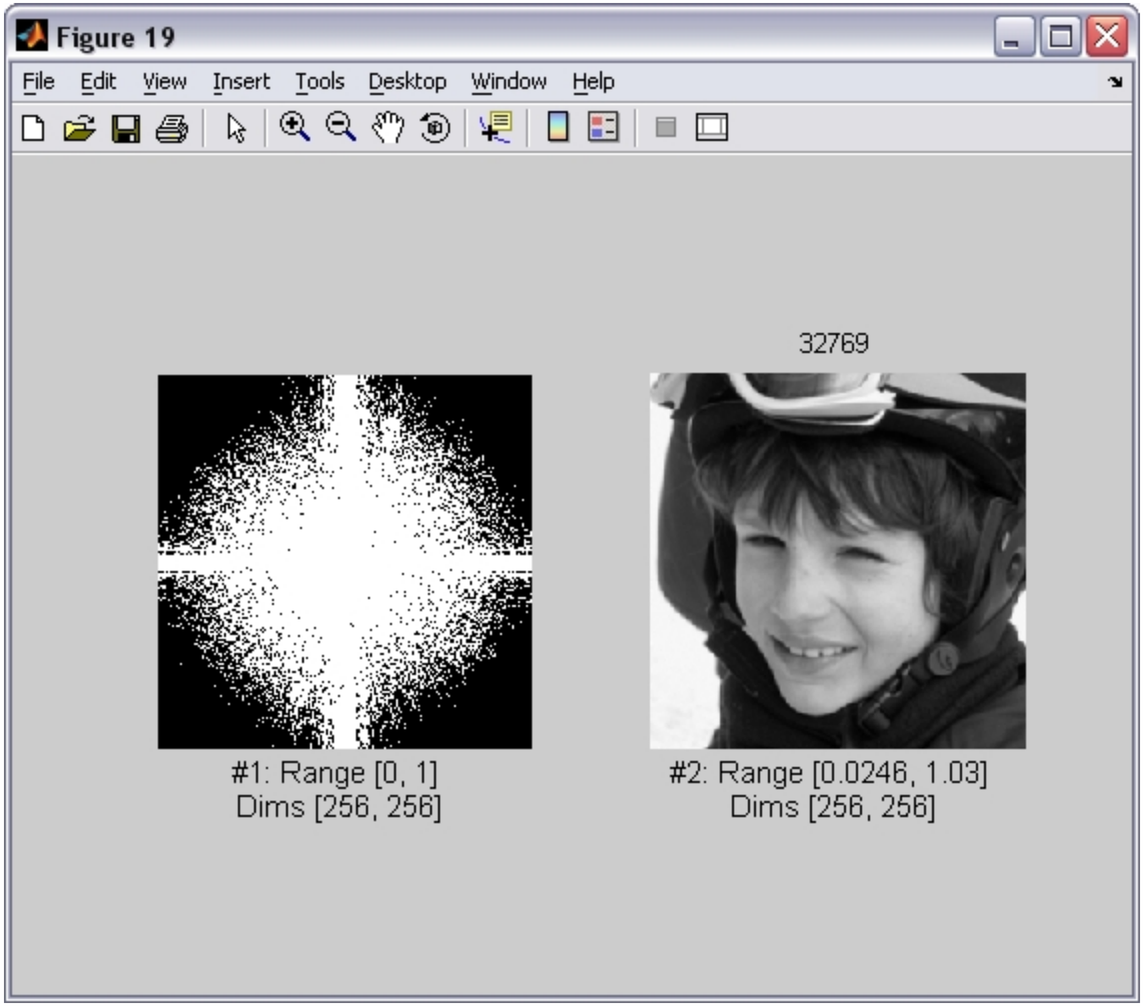


Figure 19
File Edit View Insert Tools Desktop Window Help
32769
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.0246, 1.03]
Dims [256, 256]

65536

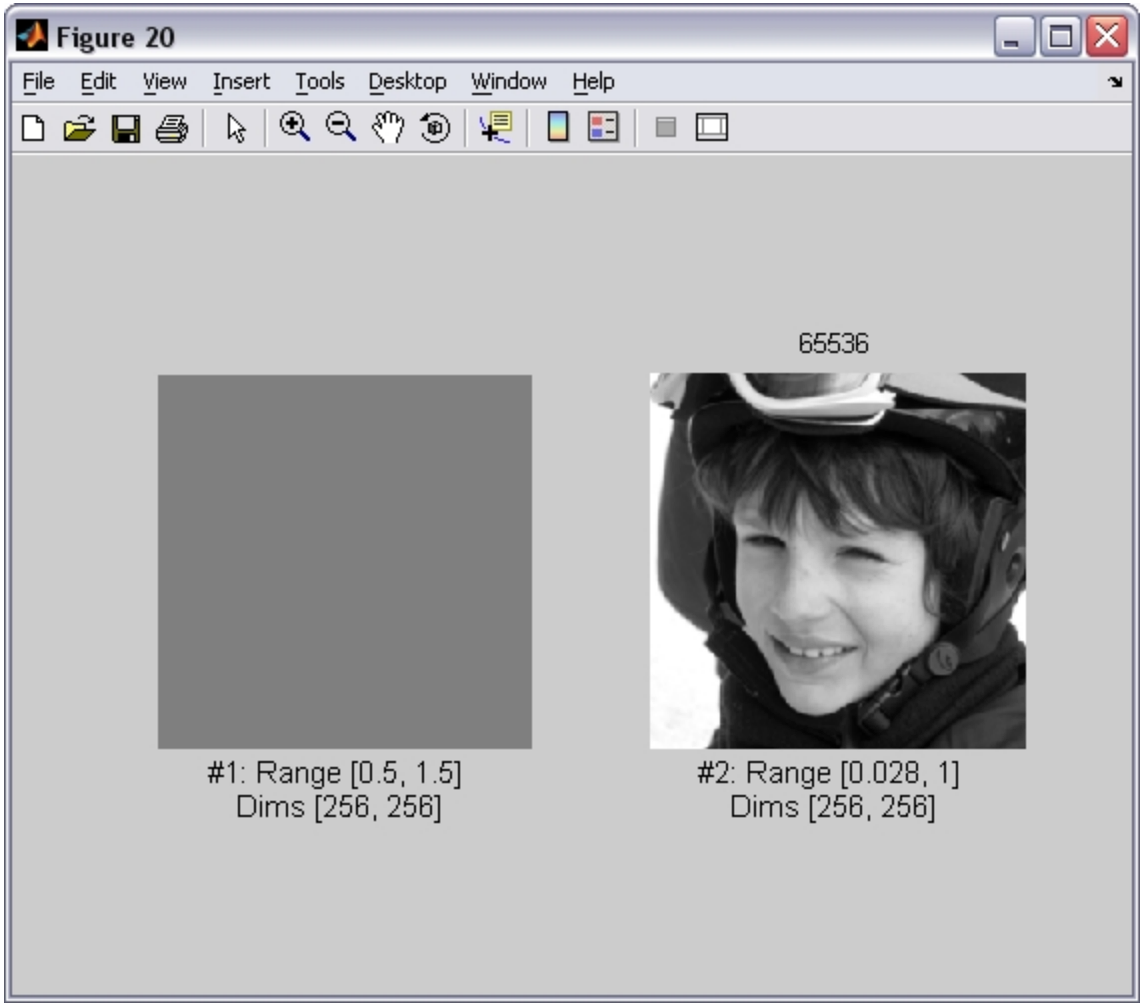
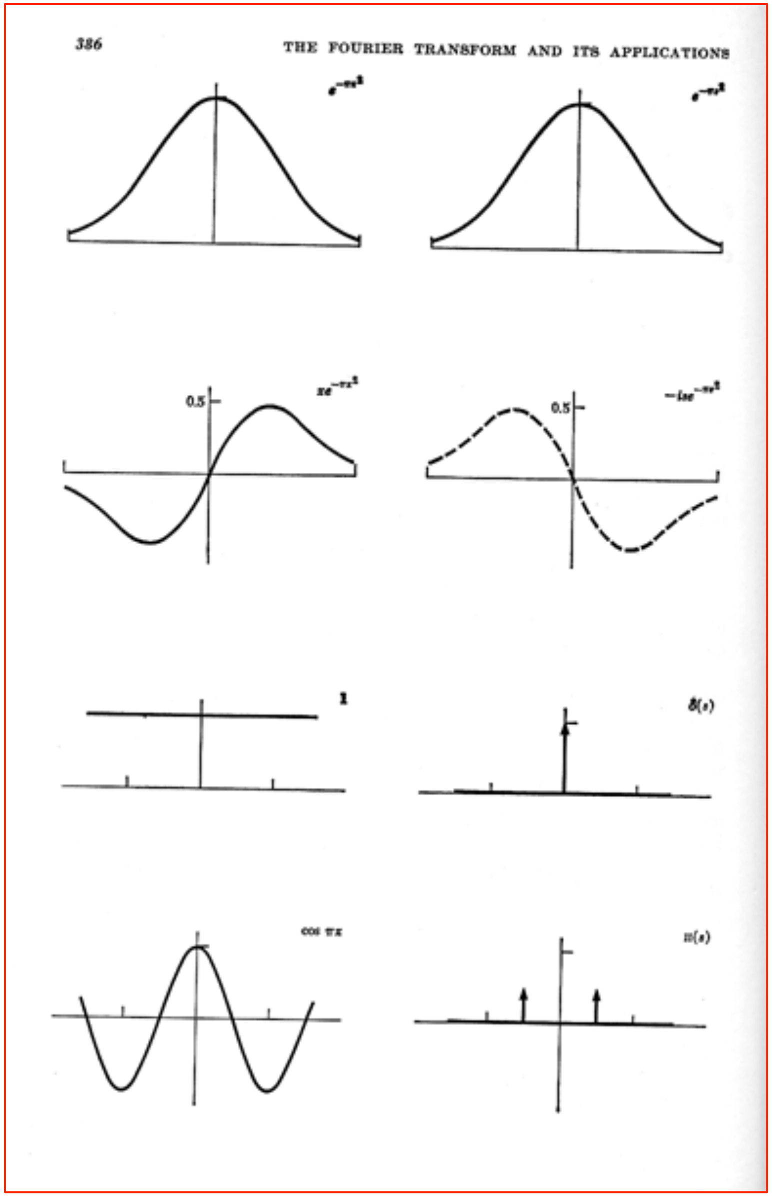
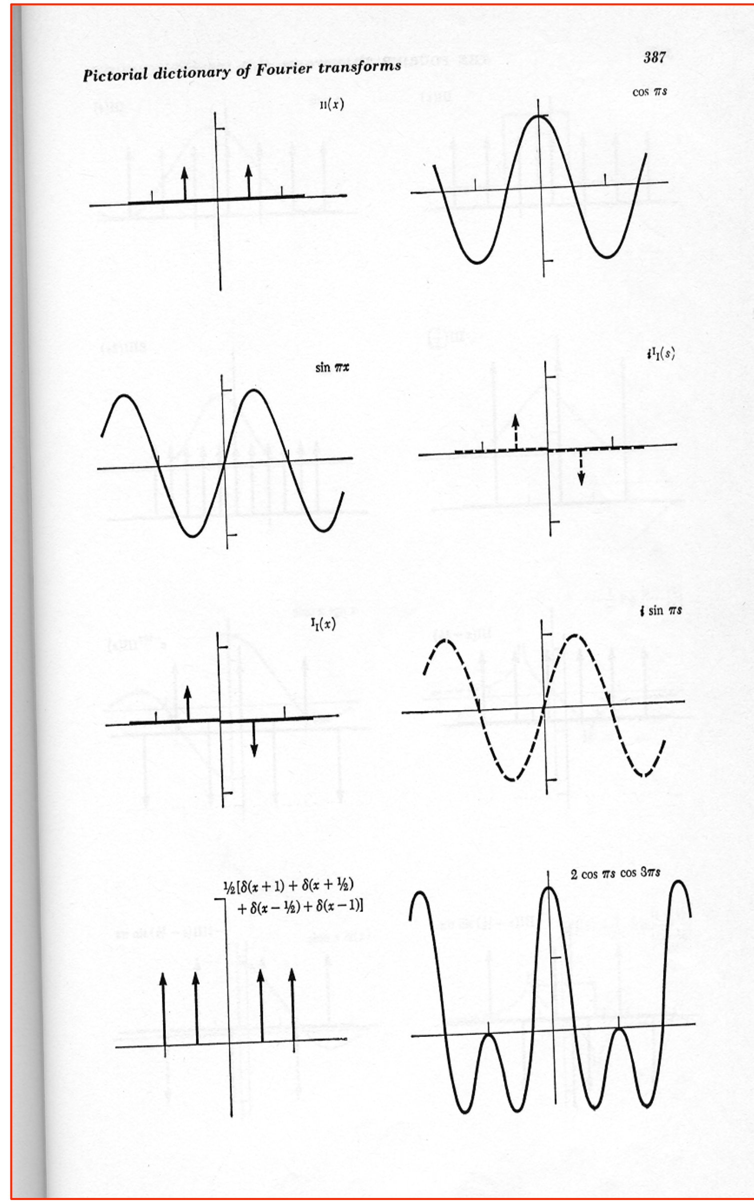


Figure 20
File
Edit
View
Insert
Tools
Desktop
Window
Help
65536
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [0.028, 1]
Dims [256, 256]

Some important Fourier Transforms

Bracewell's pictorial dictionary of Fourier transform pairs





Bracewell, The Fourier Transform and its Applications, McGraw Hill 1978

Szeliski, Computer Vision, 2010

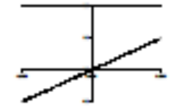
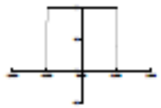
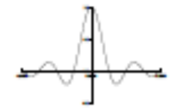
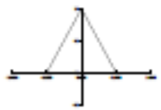
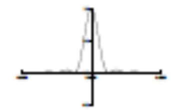
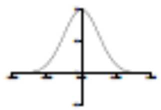
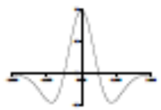
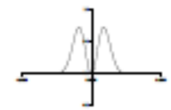
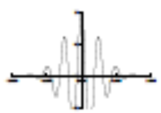
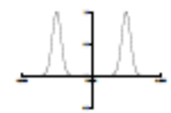
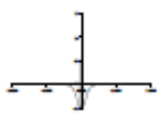
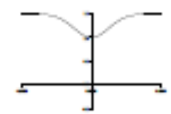
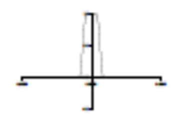
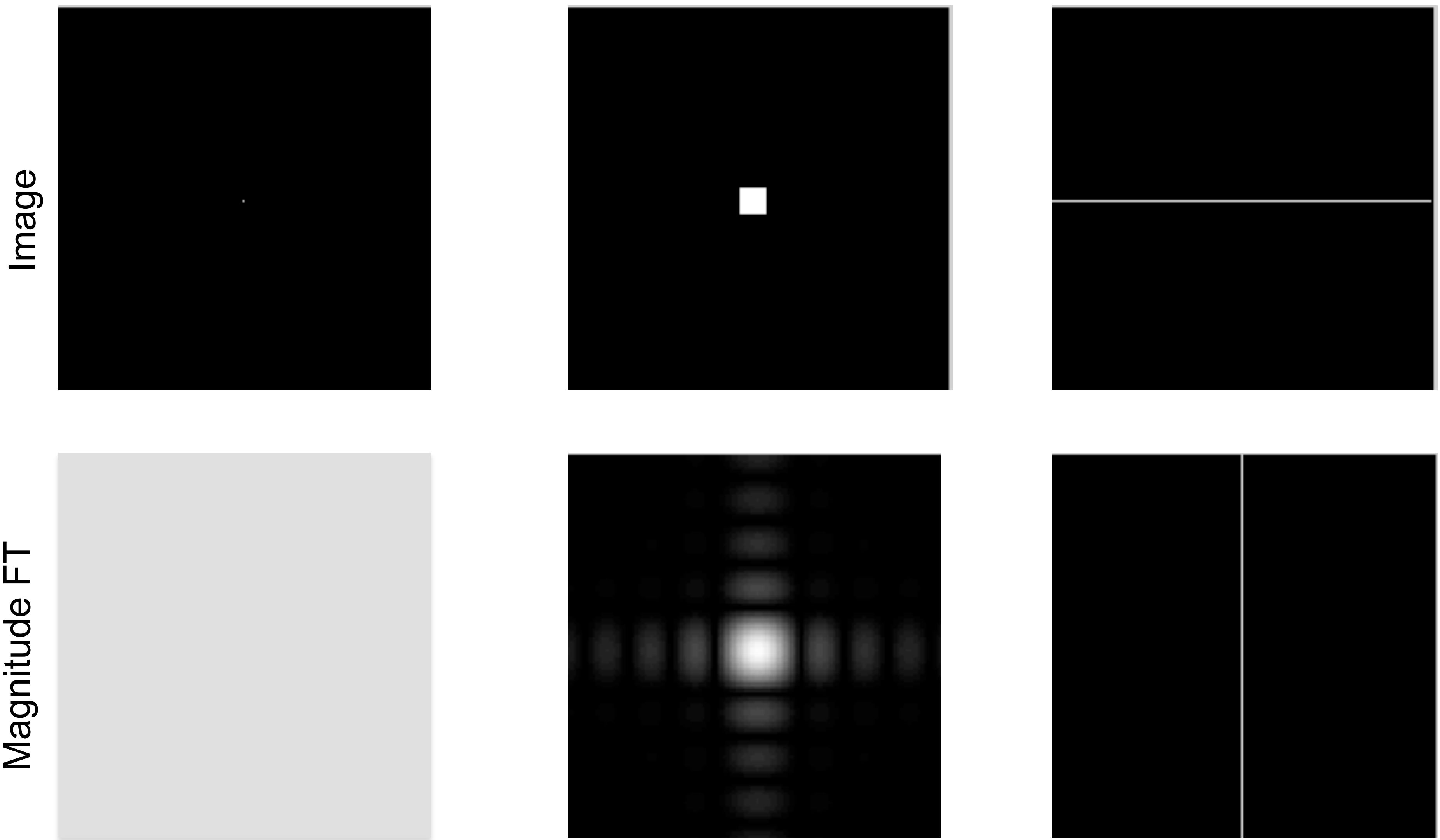
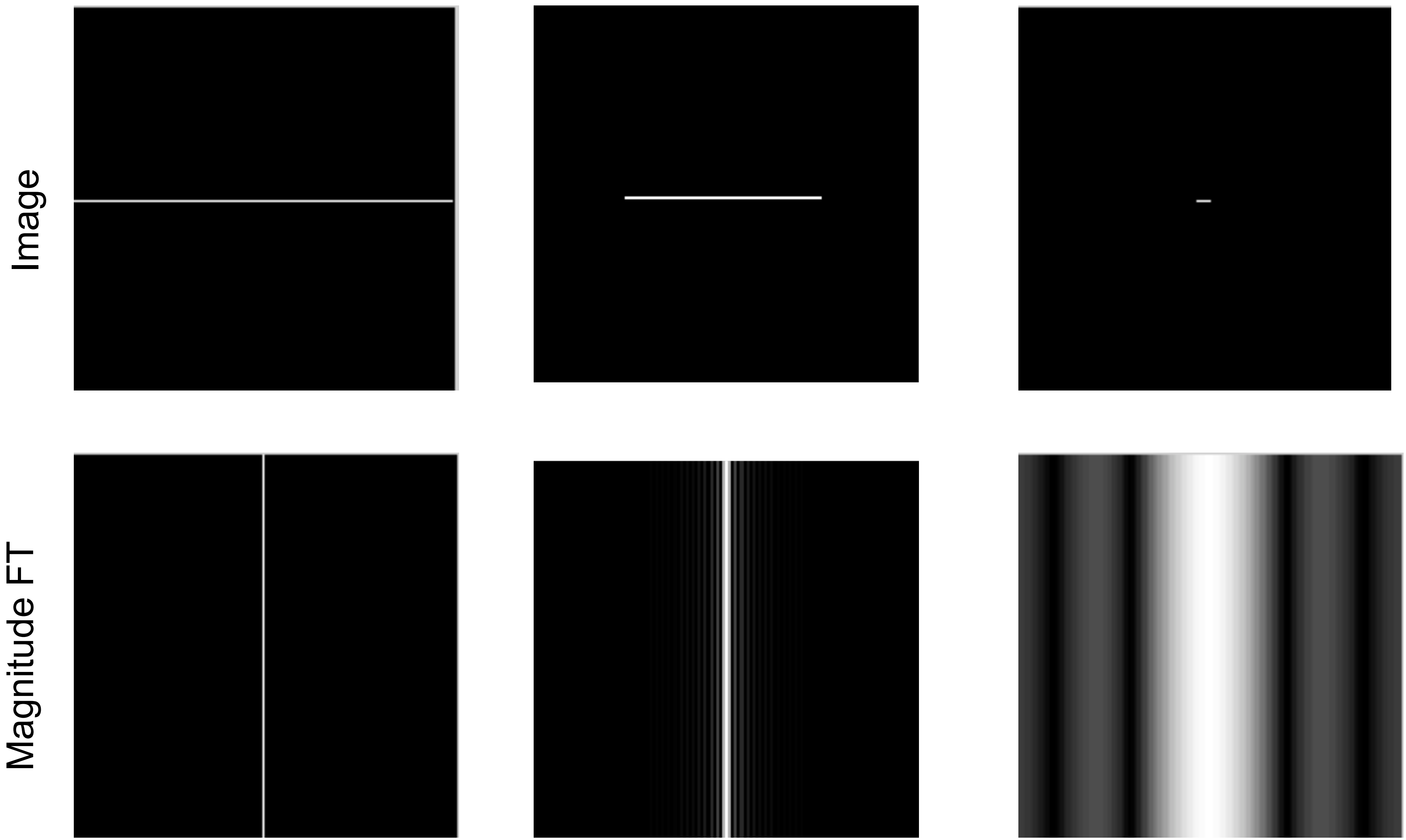
Name		Signal		Transform	
impulse		$\delta(x)$	$\Leftrightarrow$	1	
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$	
box filter		$\mathrm{box}(x/a)$	$\Leftrightarrow$	$a\mathrm{sinc}(a\omega)$	
tent		$\mathrm{tent}(x/a)$	$\Leftrightarrow$	$a\mathrm{sinc}^2(a\omega)$	
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$	
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$	
Gabor		$\cos(\omega_0 x)G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega\pm\omega_0;\sigma^{-1})$	
unsharp mask		$(1+\gamma)\delta(x)$ $-\gamma G(x;\sigma)$	$\Leftrightarrow$	$(1+\gamma)-$ $\frac{\sqrt{2\pi}\gamma}{\sigma}G(\omega;\sigma^{-1})$	
windowed sinc		$\mathrm{rcos}(x/(aW))$ $\mathrm{sinc}(x/a)$	$\Leftrightarrow$	(see Figure 3.29)	

Table 3.2 Some useful (continuous) Fourier transform pairs: The dashed line in the Fourier transform of the shifted impulse indicates its (linear) phase. All other transforms have zero phase (they are real-valued). Note that the figures are not necessarily drawn to scale but are drawn to illustrate the general shape and characteristics of the filter or its response. In particular, the Laplacian of Gaussian is drawn inverted because it resembles more a "Mexican hat", as it is sometimes called.

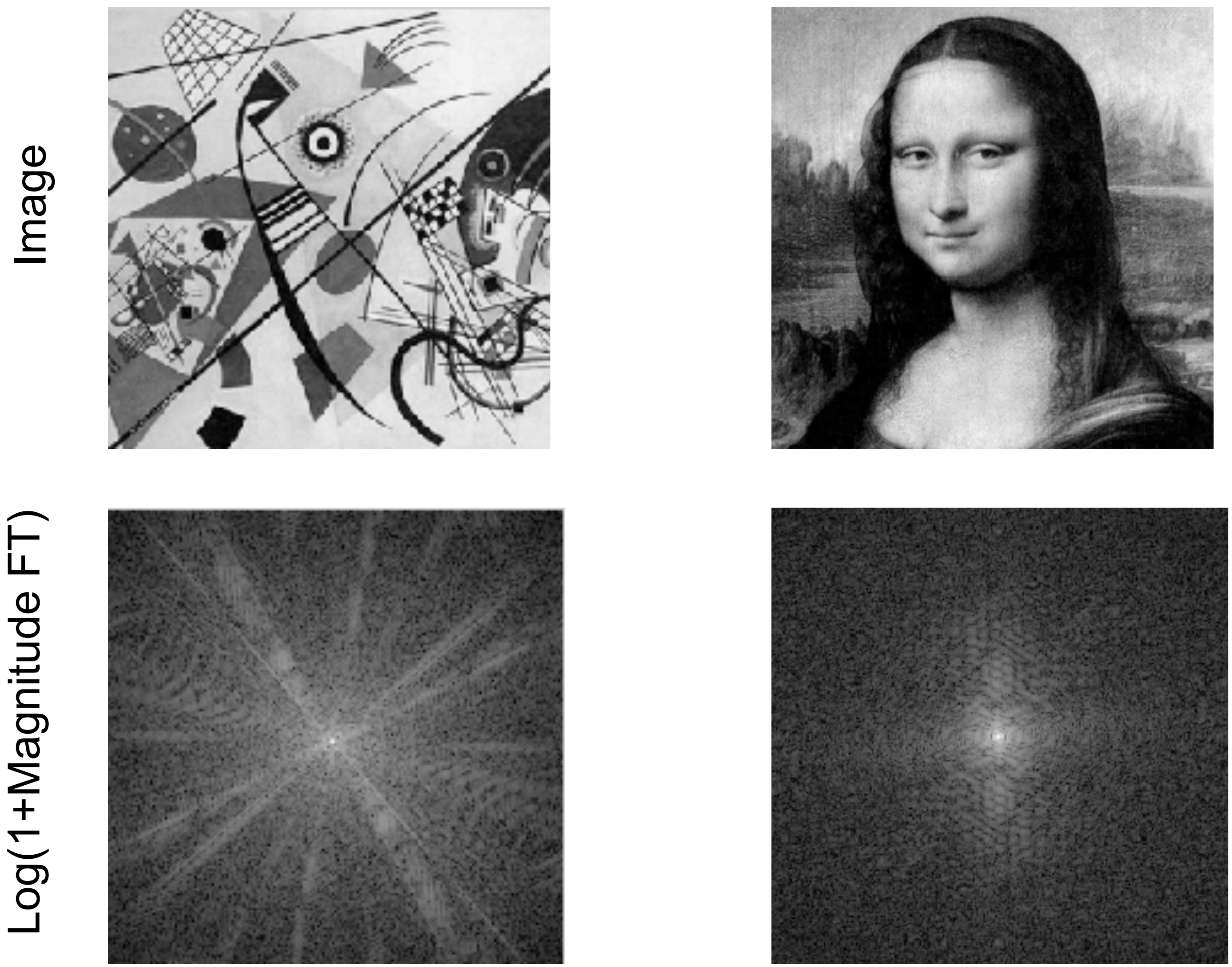
Some important Fourier Transforms



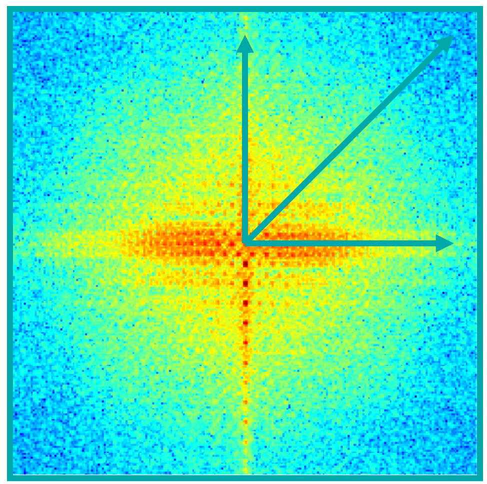
Some important Fourier Transforms

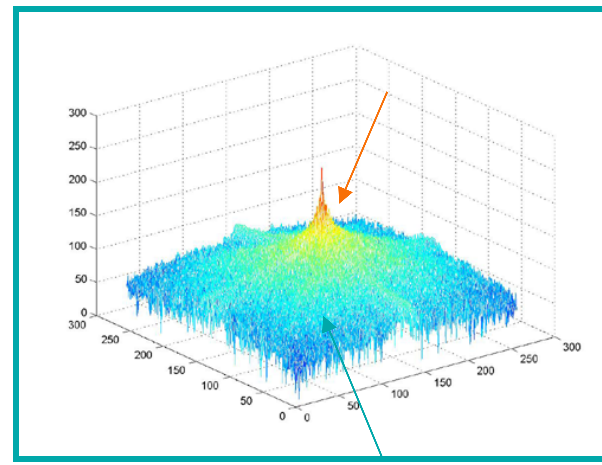


The Fourier Transform of some important images



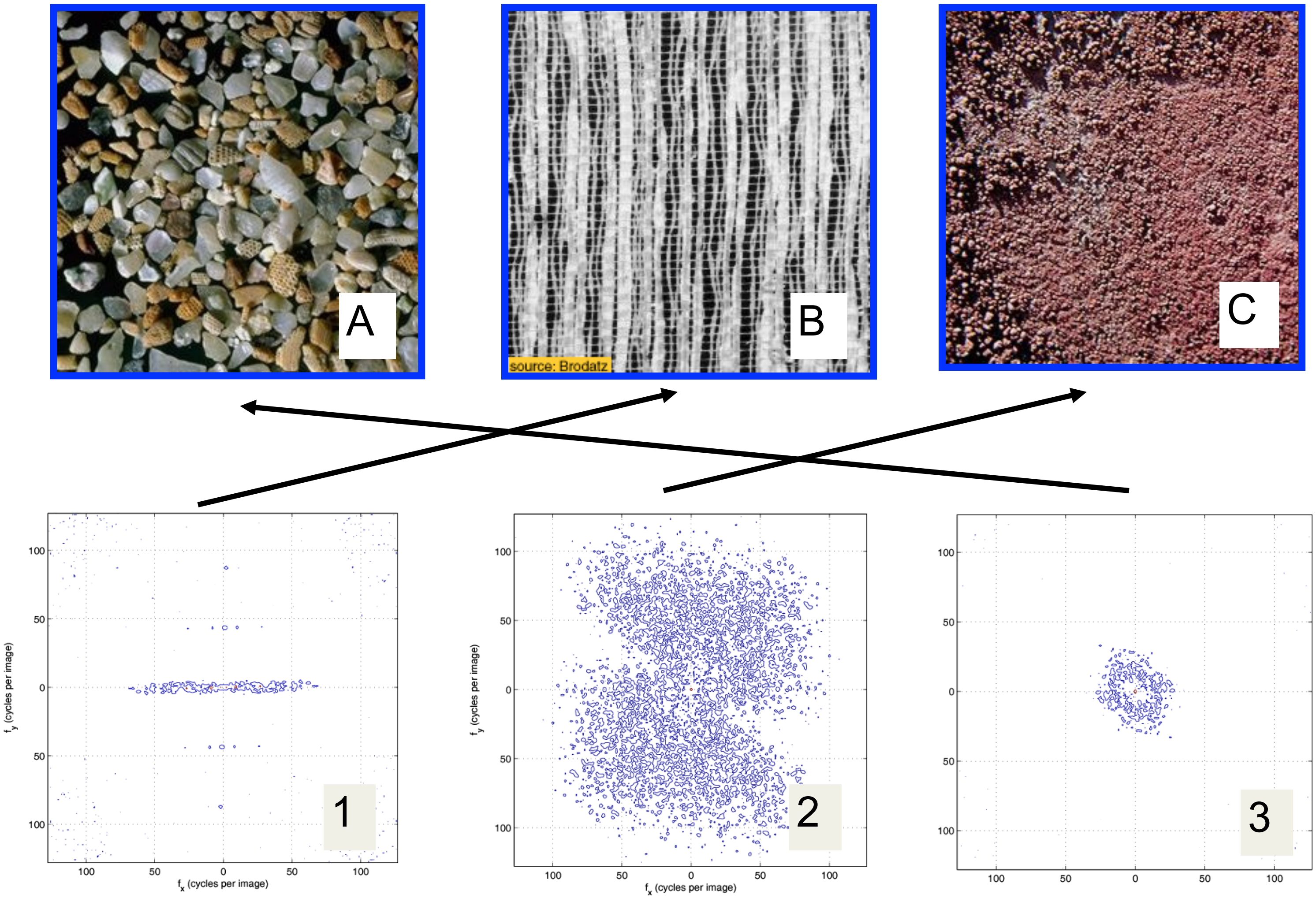
How to interpret a Fourier Spectrum





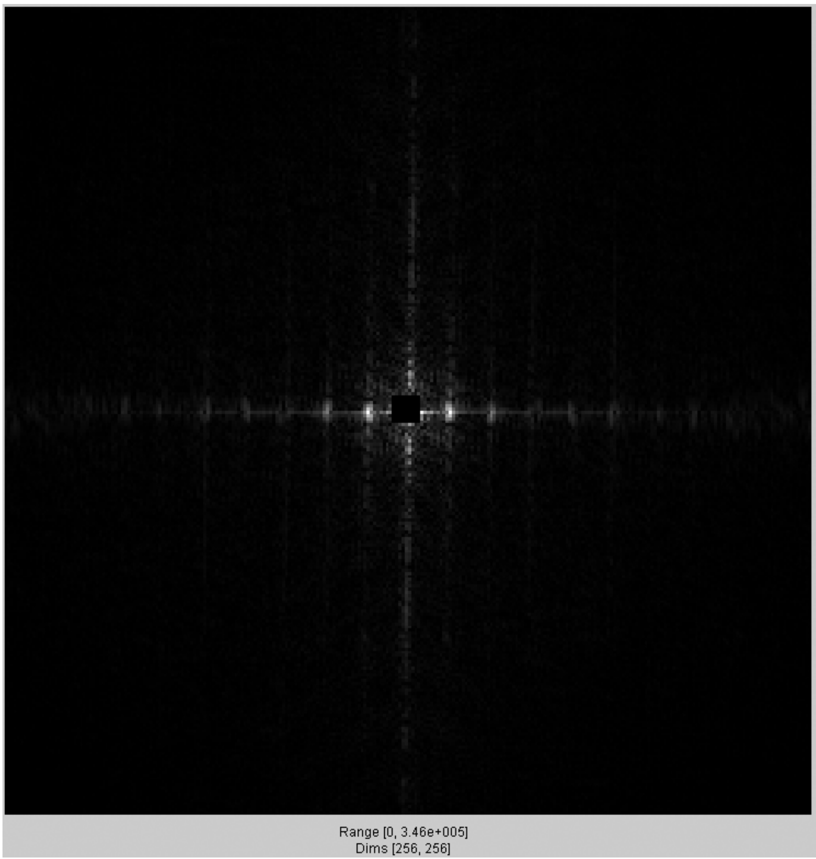
Log power spectrum

Fourier Amplitude Spectrum

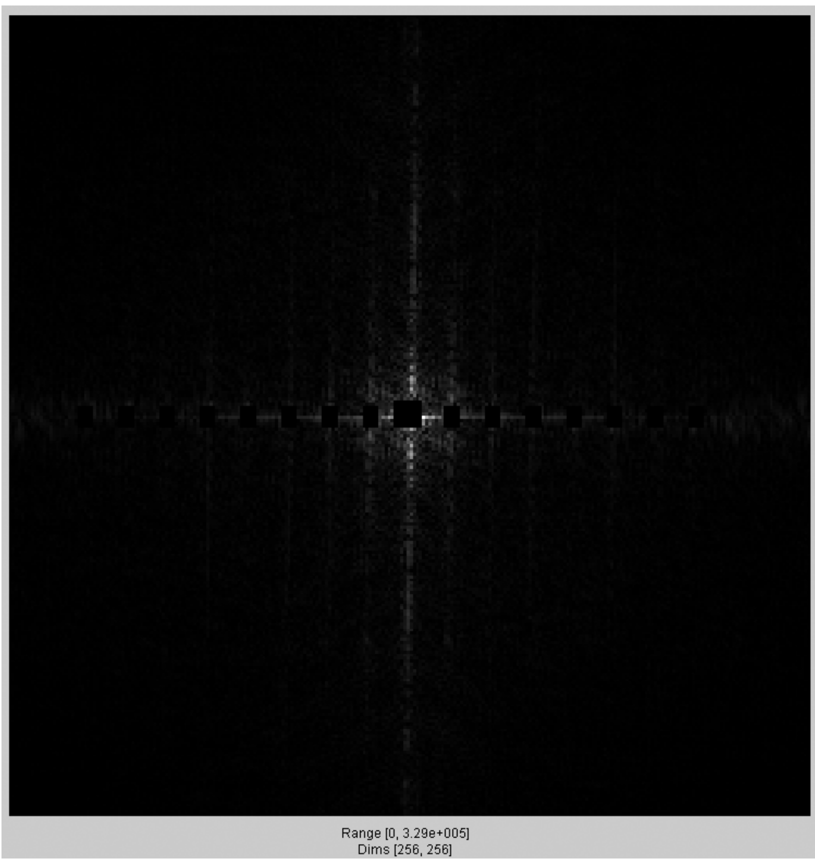


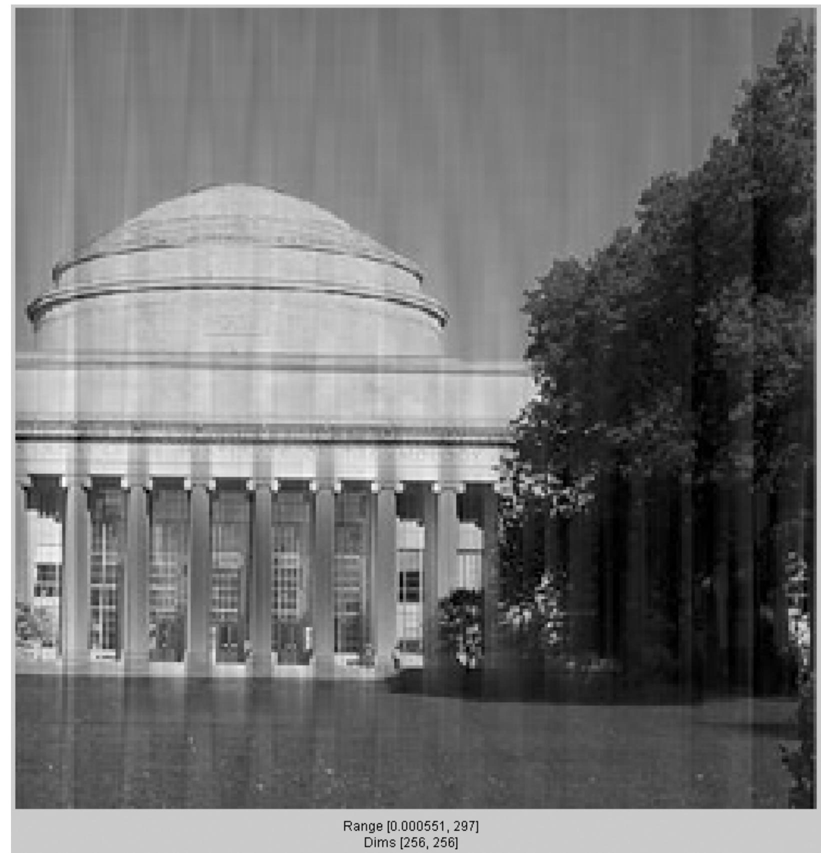
fx(cycles/image pixel size) fx(cycles/image pixel size) fx(cycles/image pixel size)

Fourier transform magnitude



Masking out the fundamental and harmonics from periodic pillars

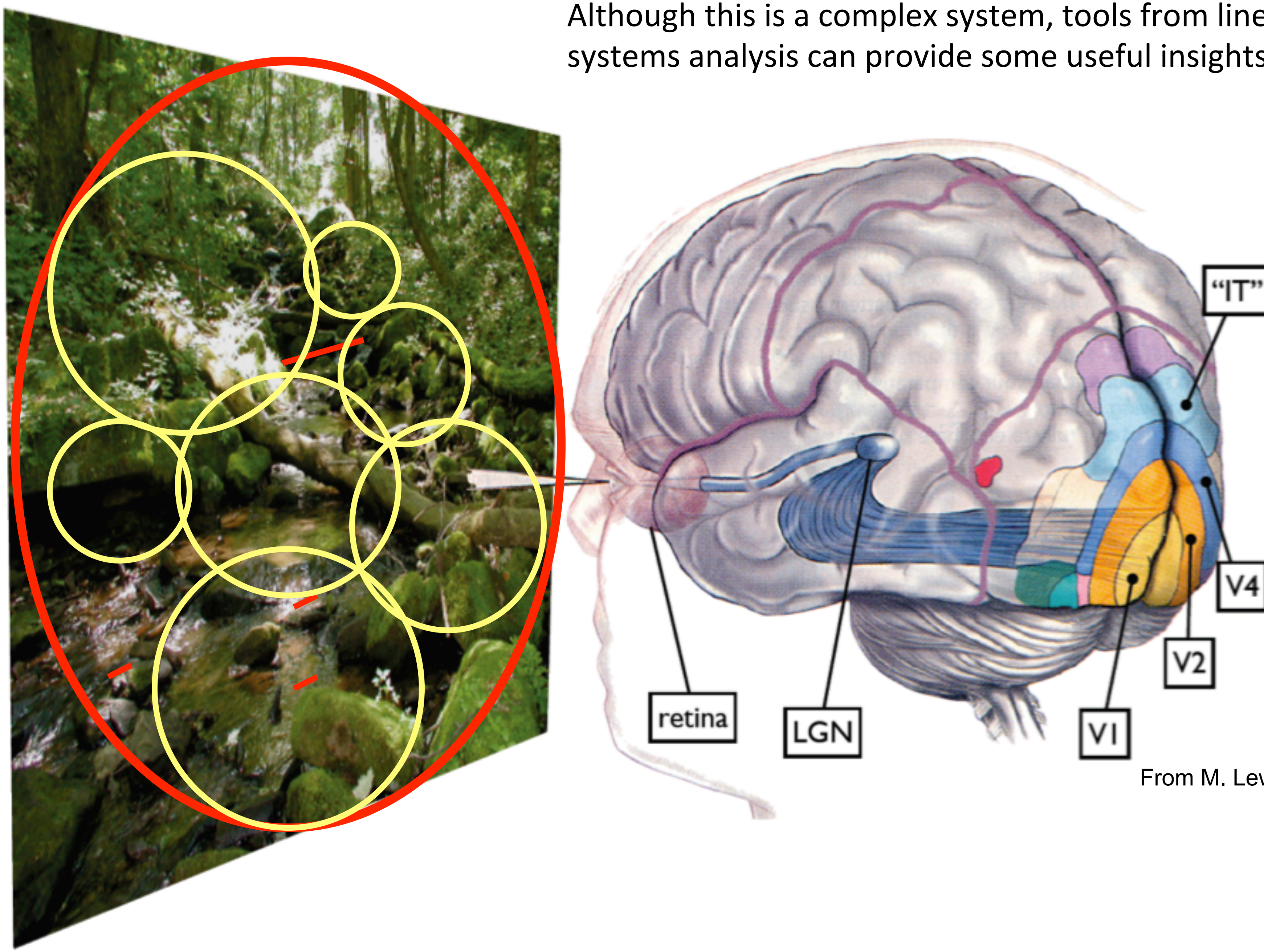




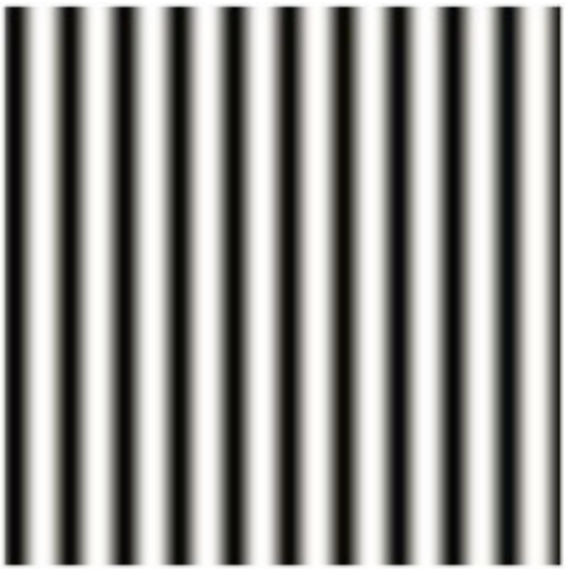
Outline

- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

Although this is a complex system, tools from linear systems analysis can provide some useful insights…



From M. Lewicky



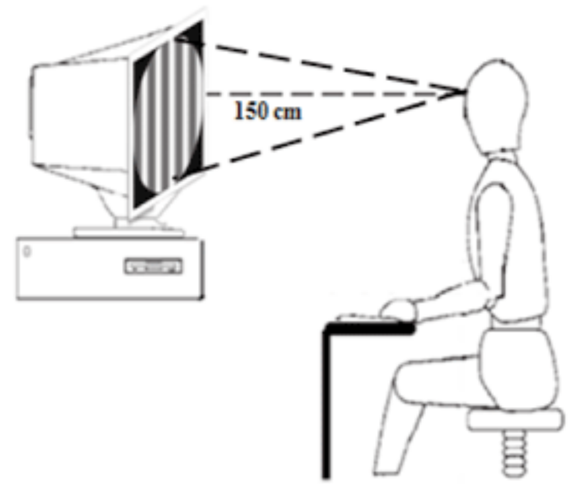
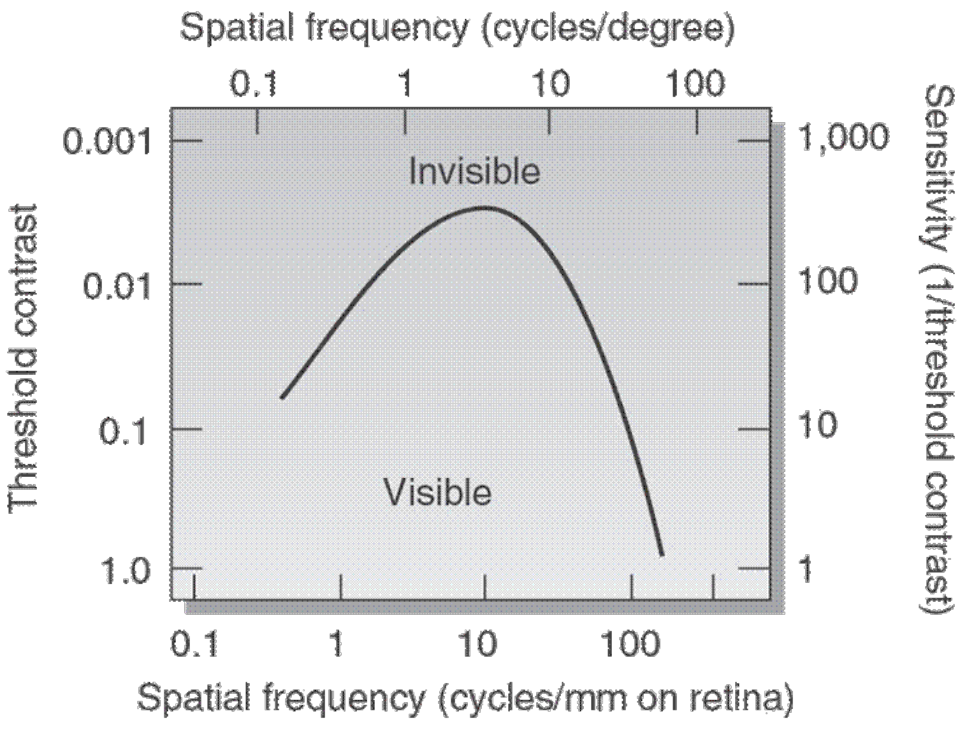
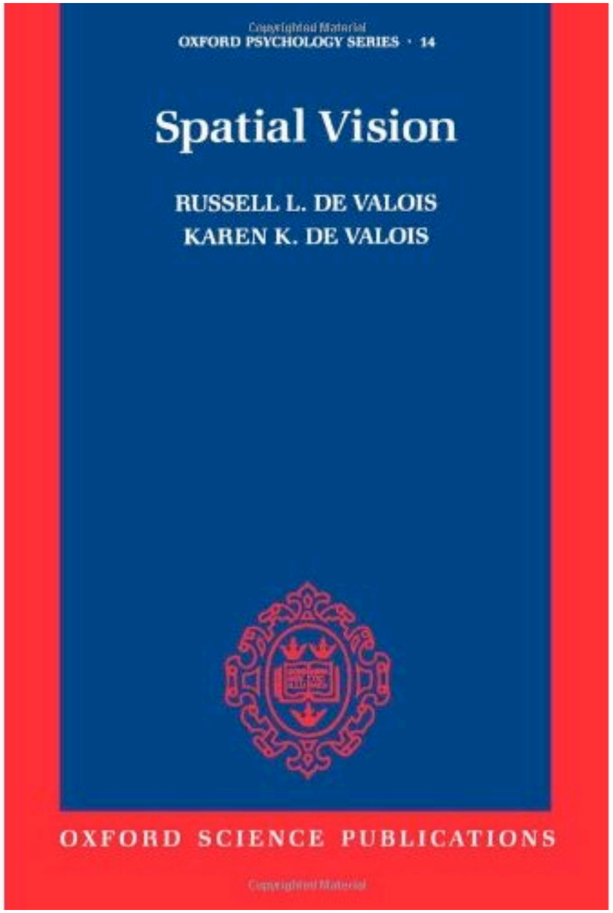


Figure 1. Stimulus presentation scheme. The stimuli were originally calibrated to be seen at a distance of 150 cm in a 19″ display.

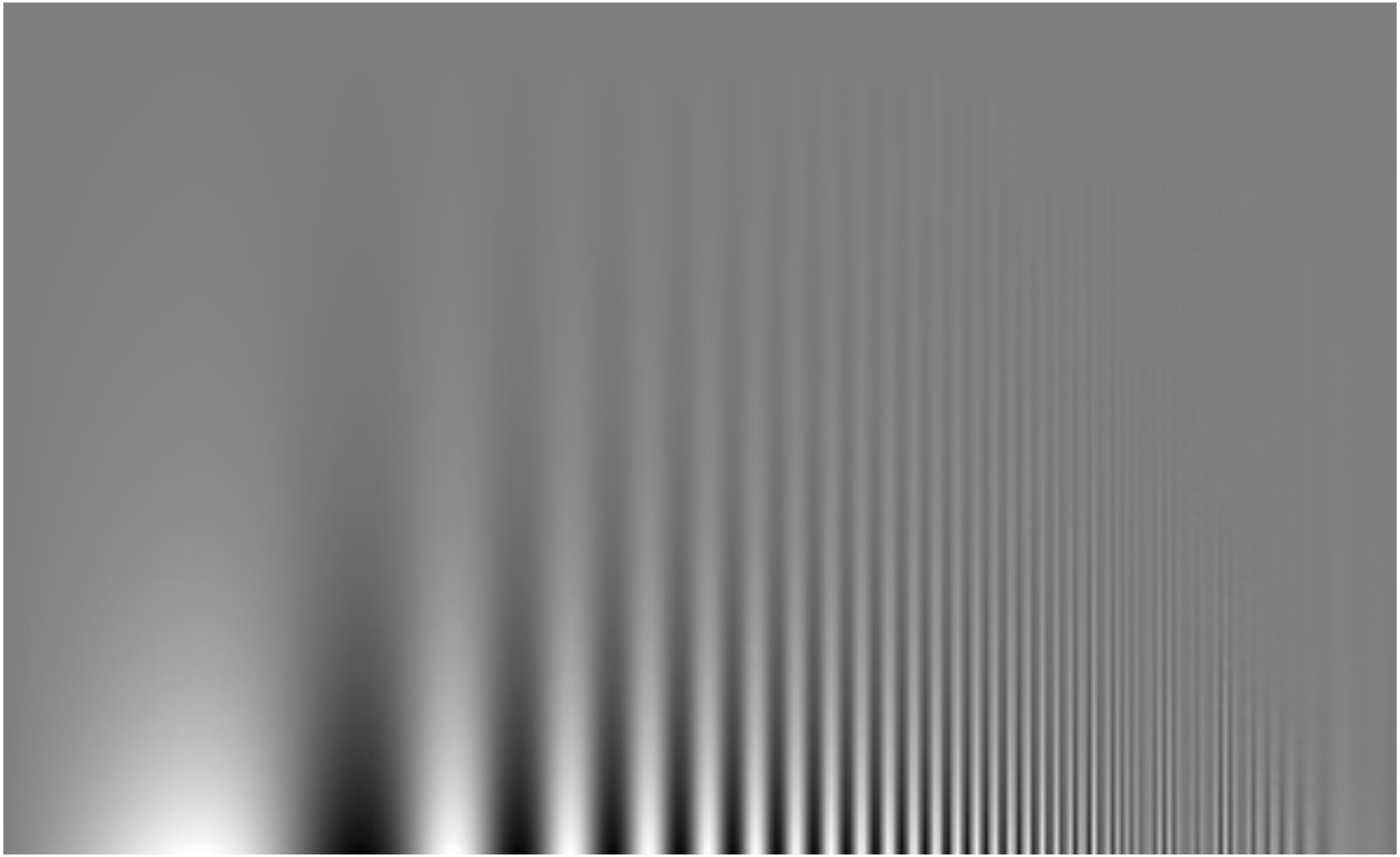
Contrast Sensitivity Function





From:
http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/channels/channels.html

Contrast Sensitivity Function



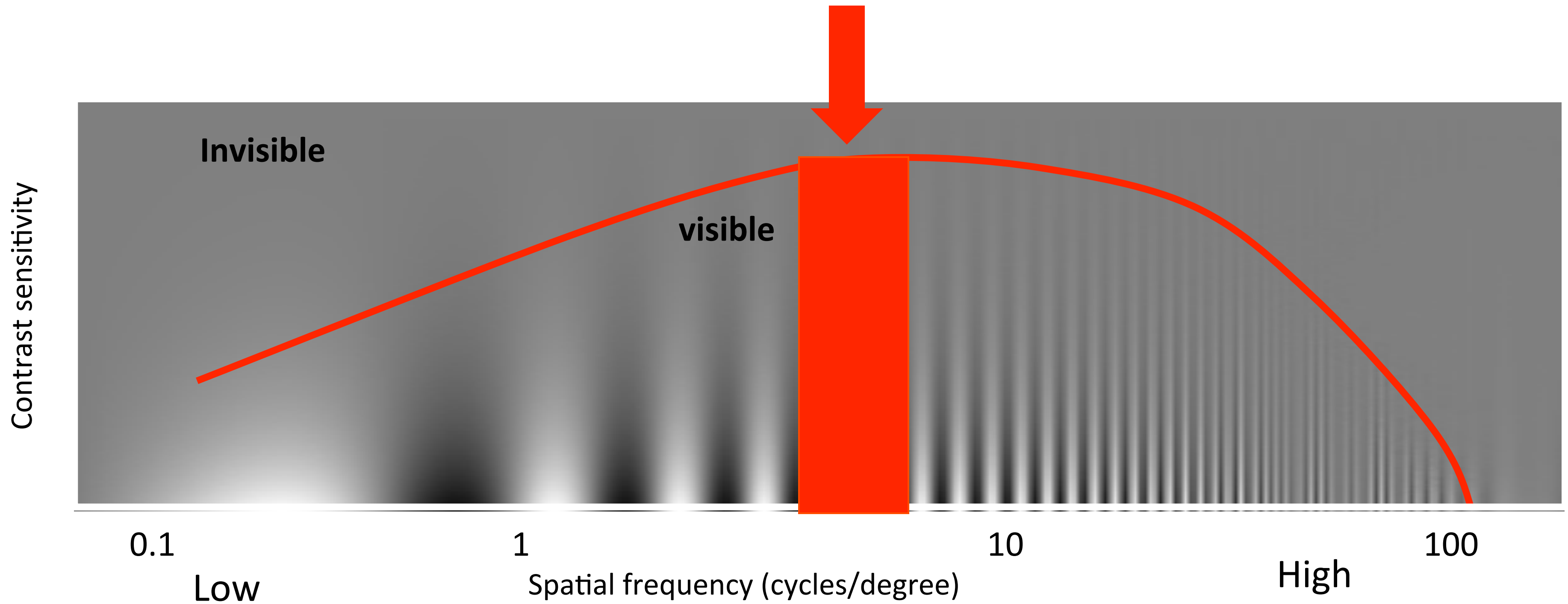
A demo of human contrast sensitivity as a function of spatial frequency. Frequency rises from left to right at a constant rate. Contrast drops from bottom to top at a constant rate. The bars are visible further up for middle frequencies, showing these are more salient to the human visual system.

Contrast Sensitivity Function

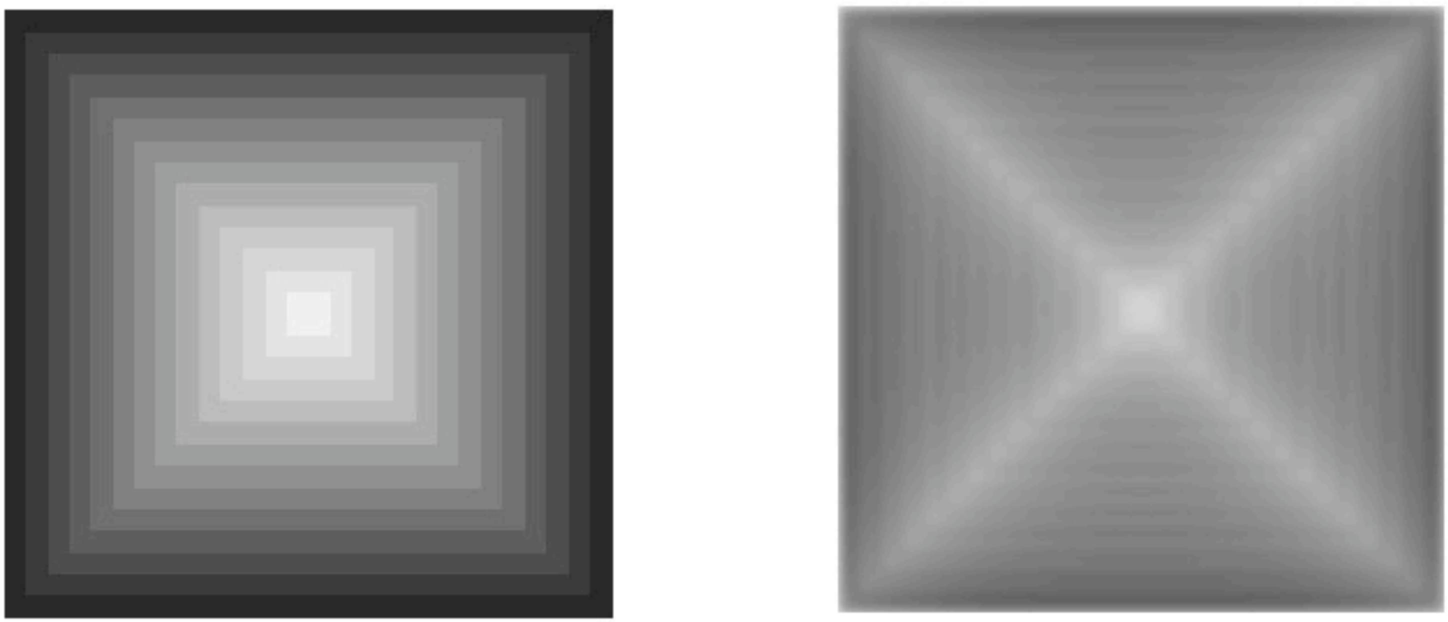
Blackmore & Campbell (1969)

Maximum sensitivity

~ **6** cycles / degree of visual angle



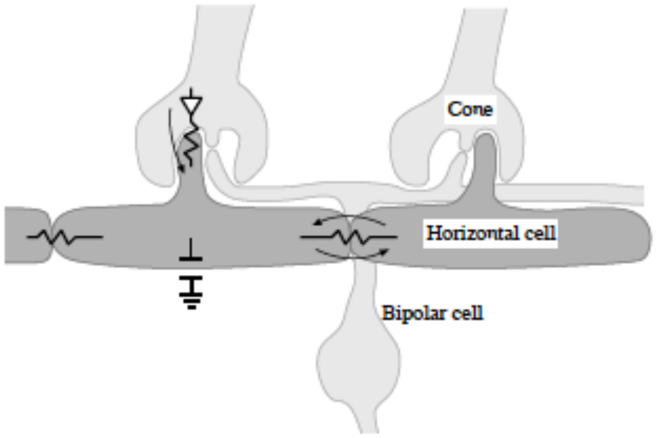
Laplacian



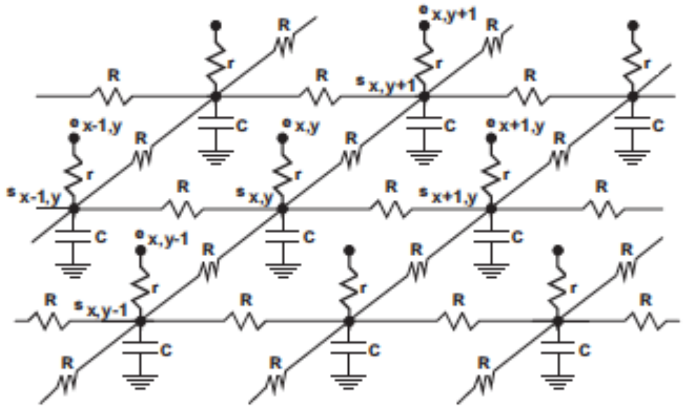
a b

An illusion by Vasarely, left, and a bandpass filtered version, right.

http://web.mit.edu/persci/people/adelson/publications/gazzan.dir/vasarely.html

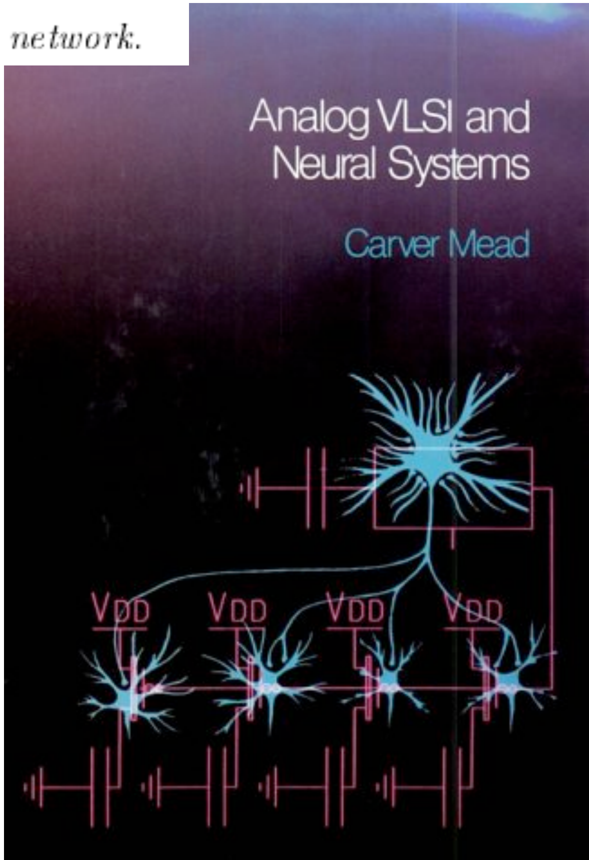


a)



b)

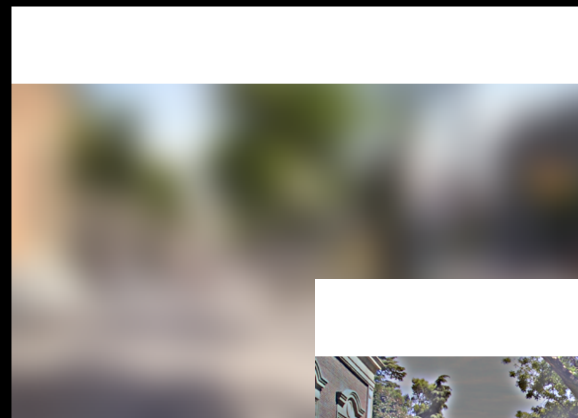
Figure 1.2: *a) Schema of the horizontal cell layer of the retina. b) RC analog network.*



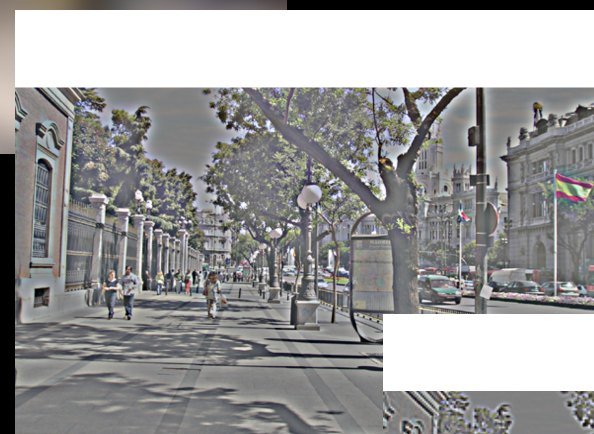
Neuromorphic circuits

Human Visual Perception

Blur image



Sharp image



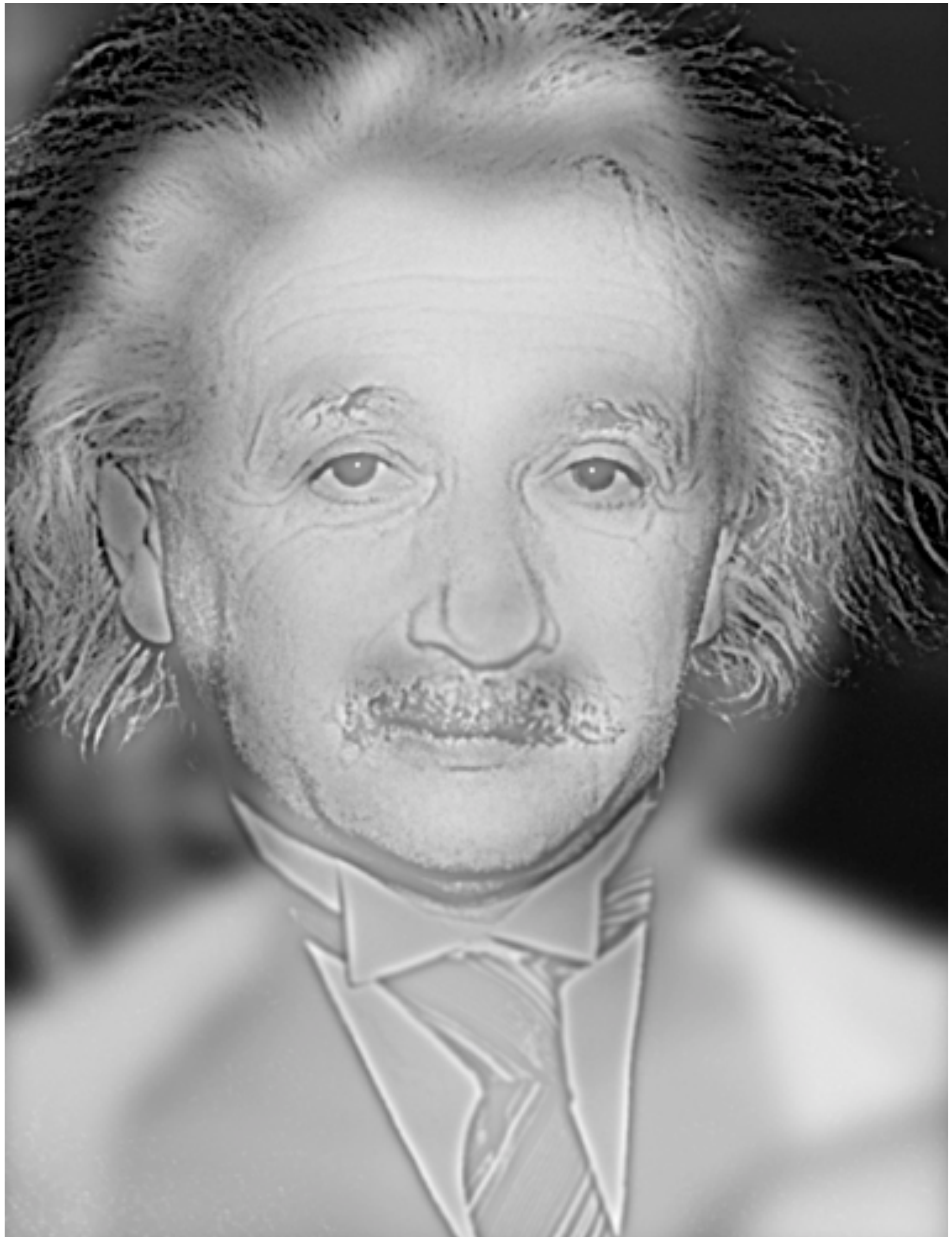
Spatial frequency channels

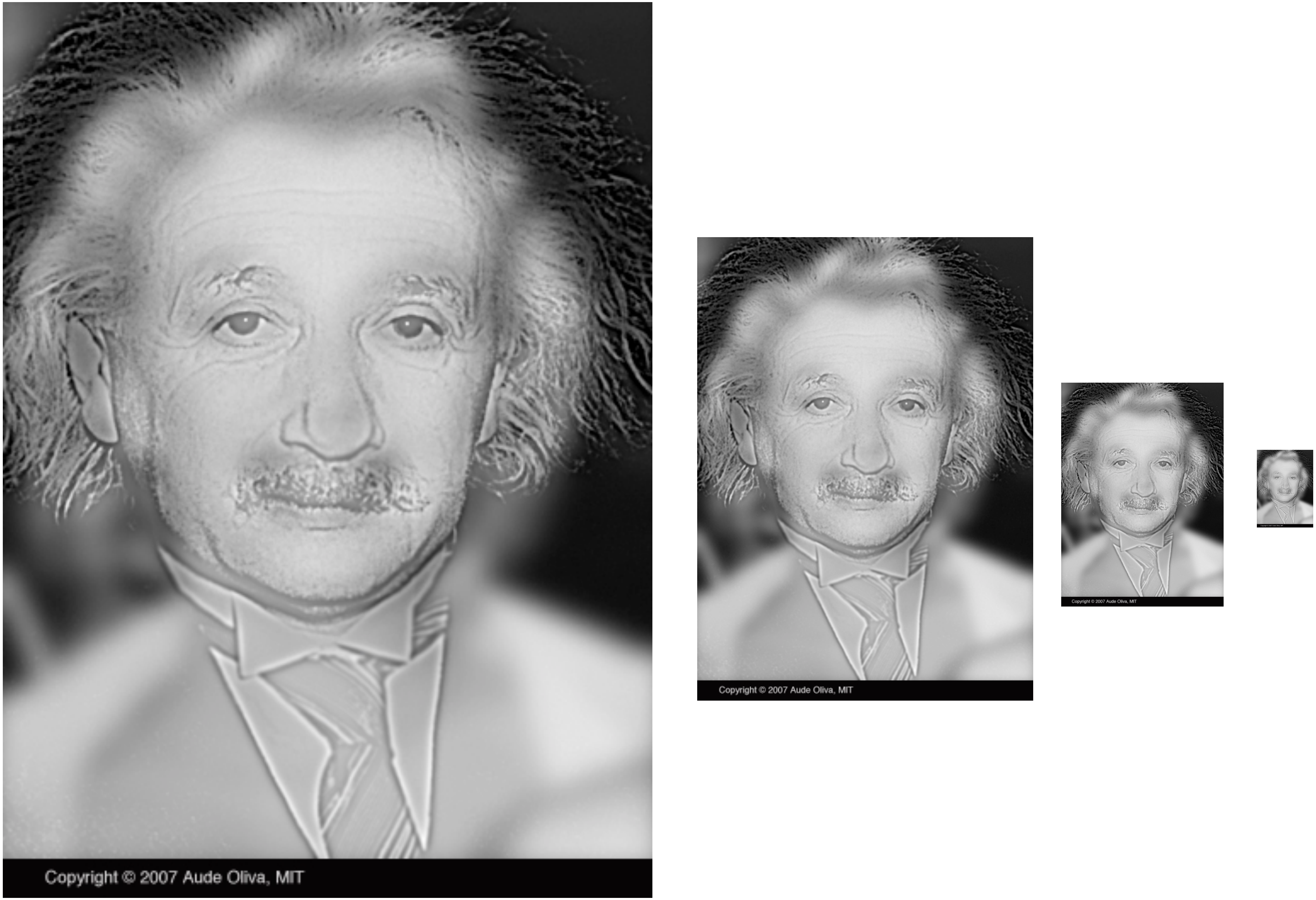
Hybrid Images

Oliva & Schyns

Hybrid Images

Hybrid Images





http://cvcl.mit.edu/hybrid_gallery/gallery.html

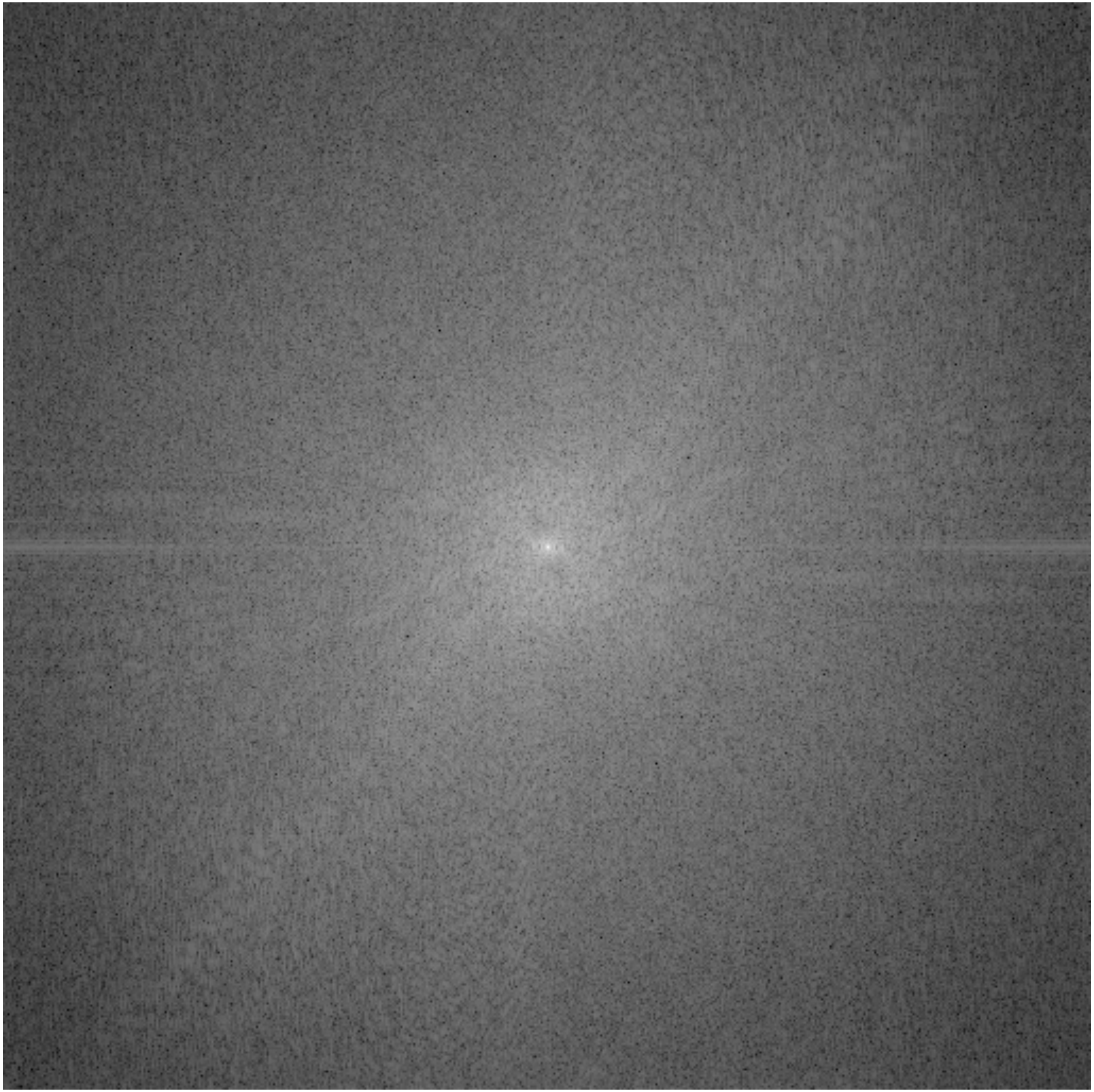
Outline

- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

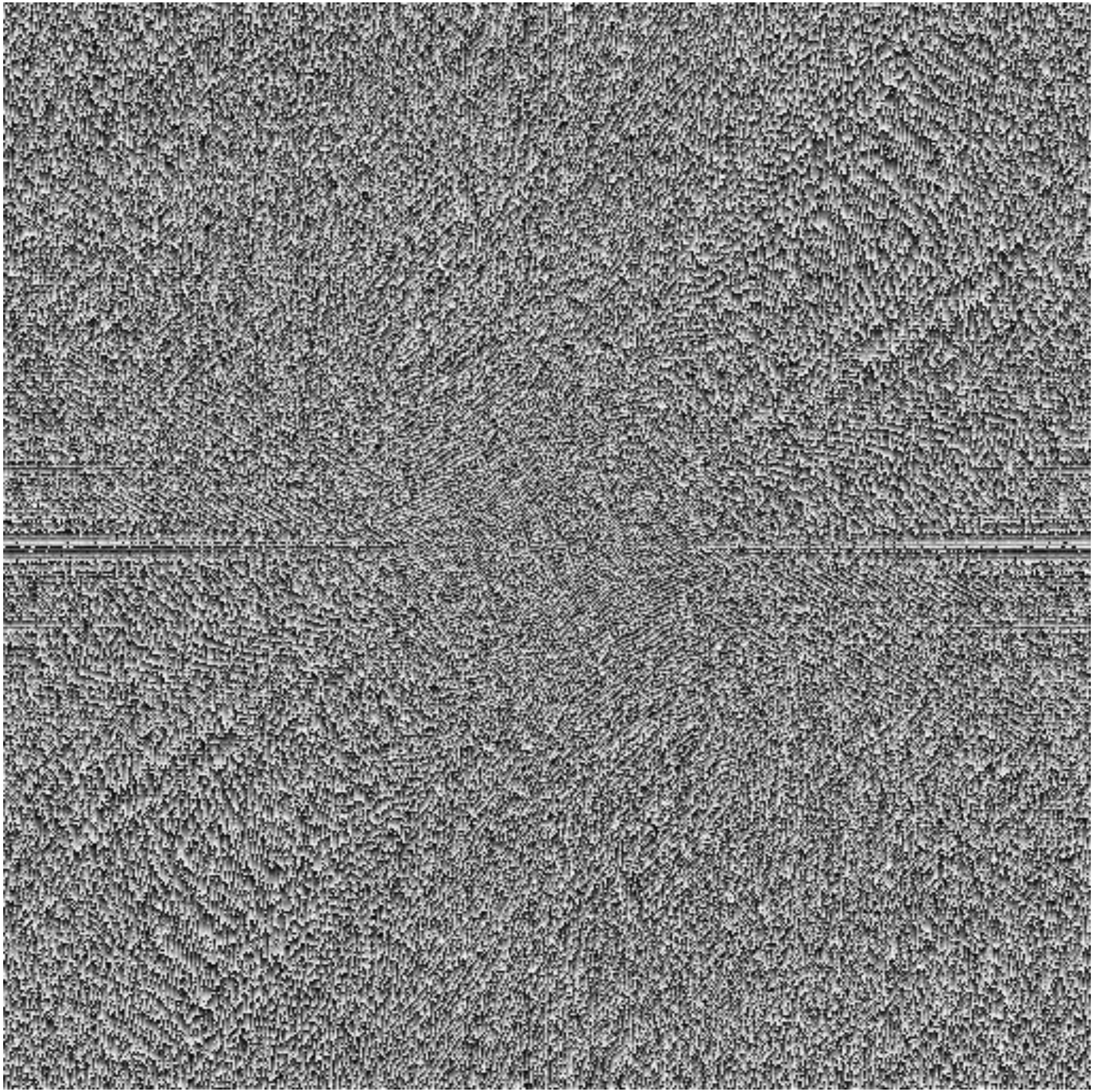
Phase and Magnitude

- 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?

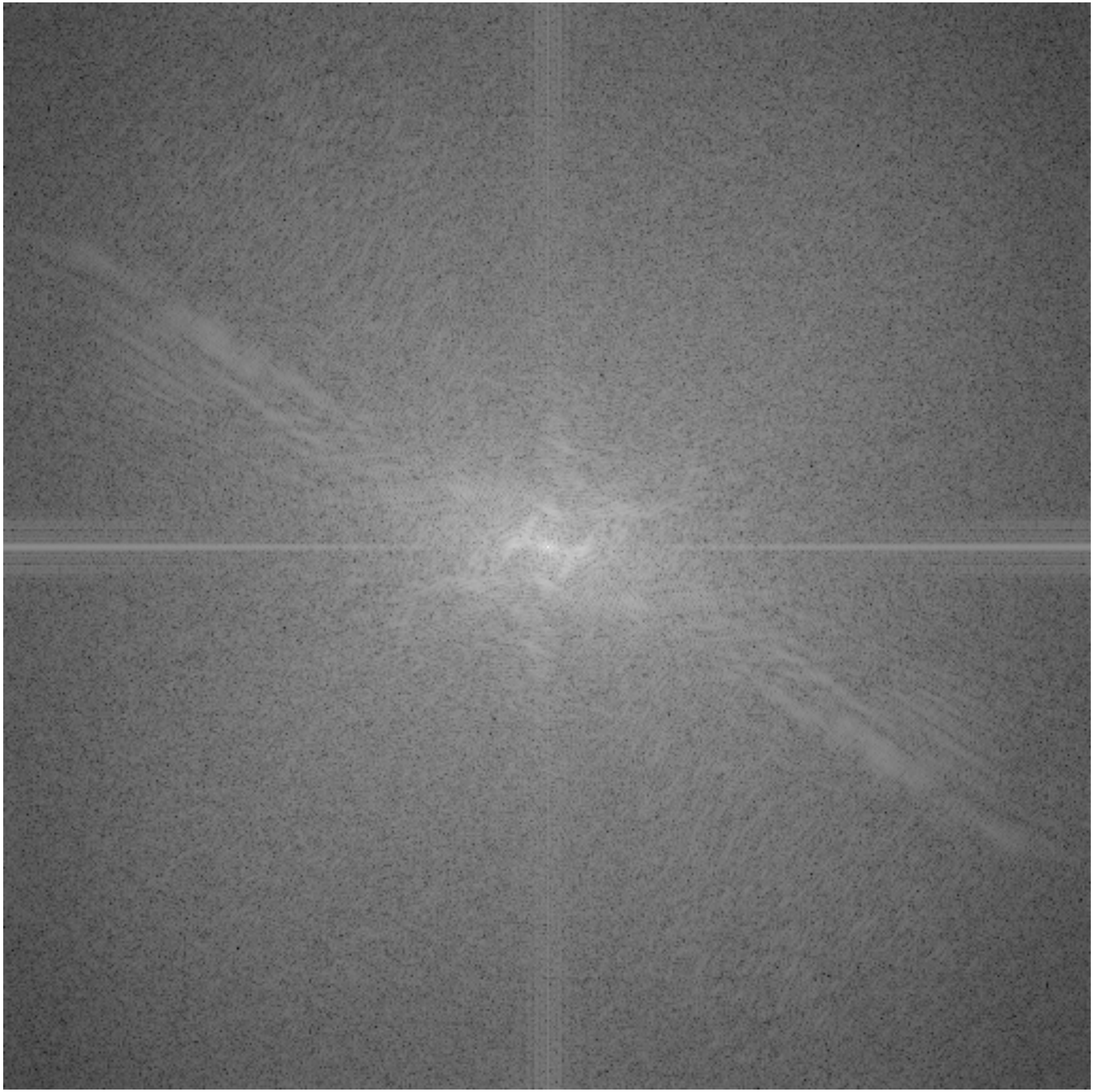
This is the magnitude transform of the cheetah pic



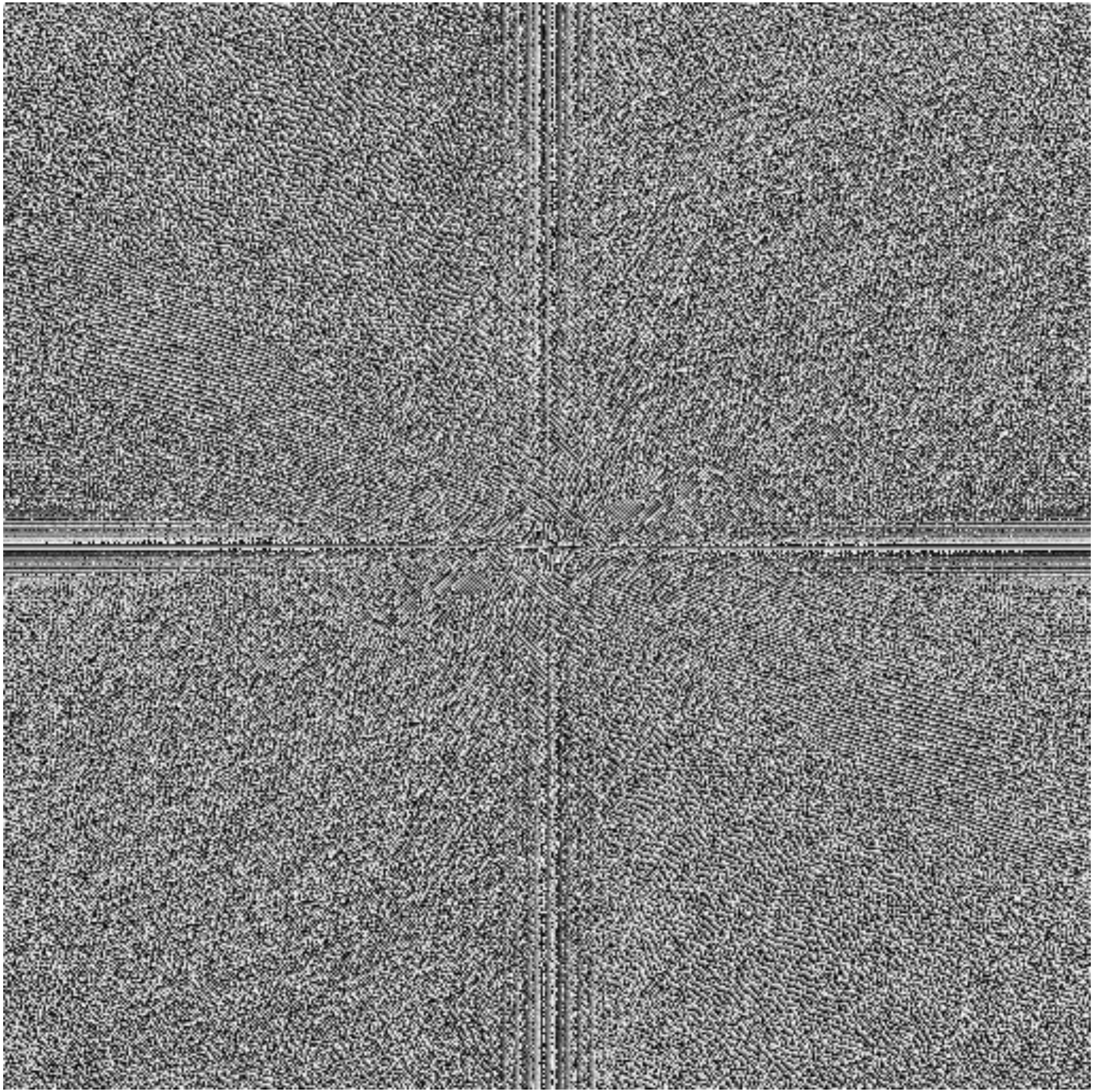
This is the phase transform of the cheetah pic



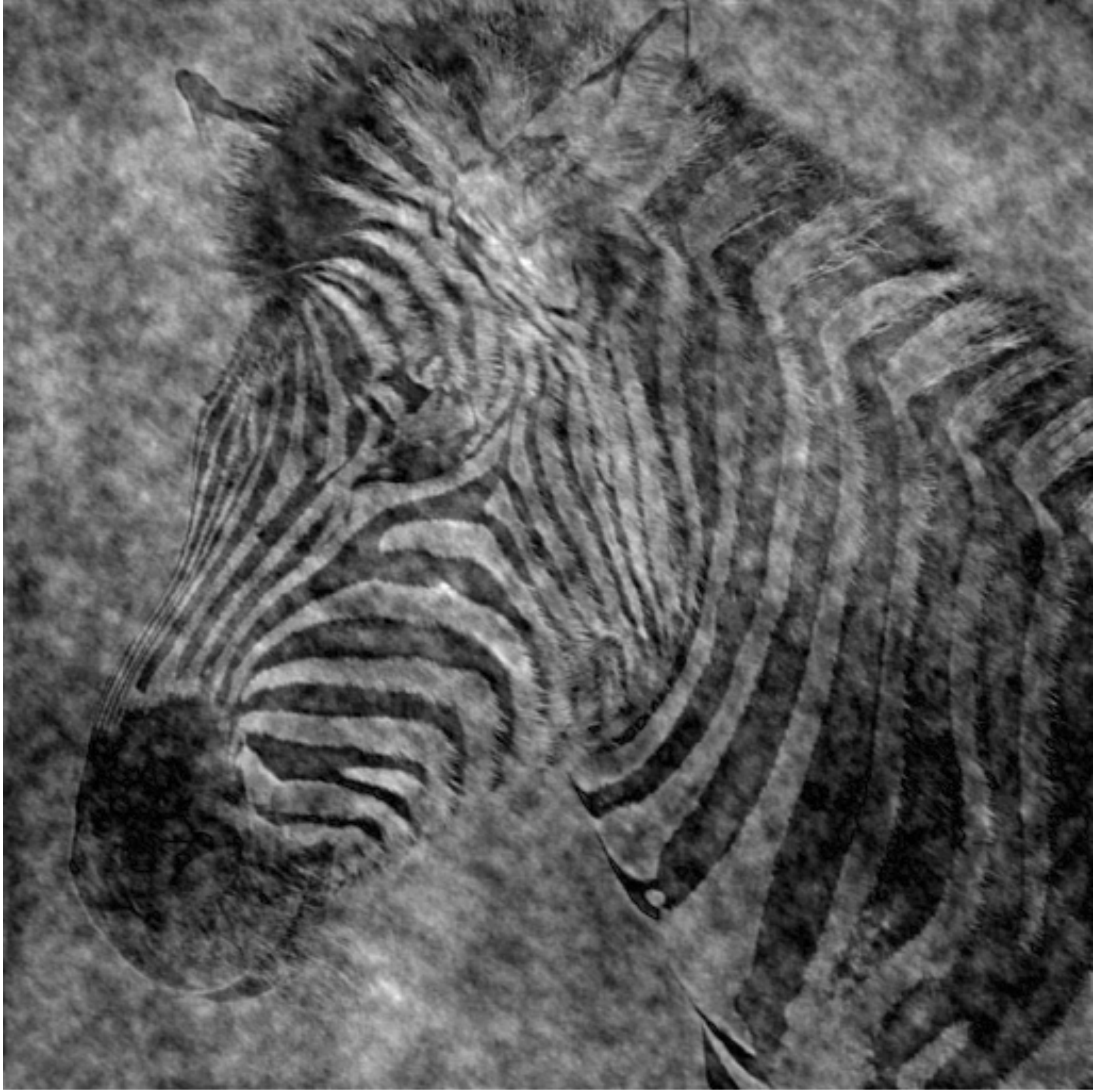
This is the magnitude transform of the zebra pic



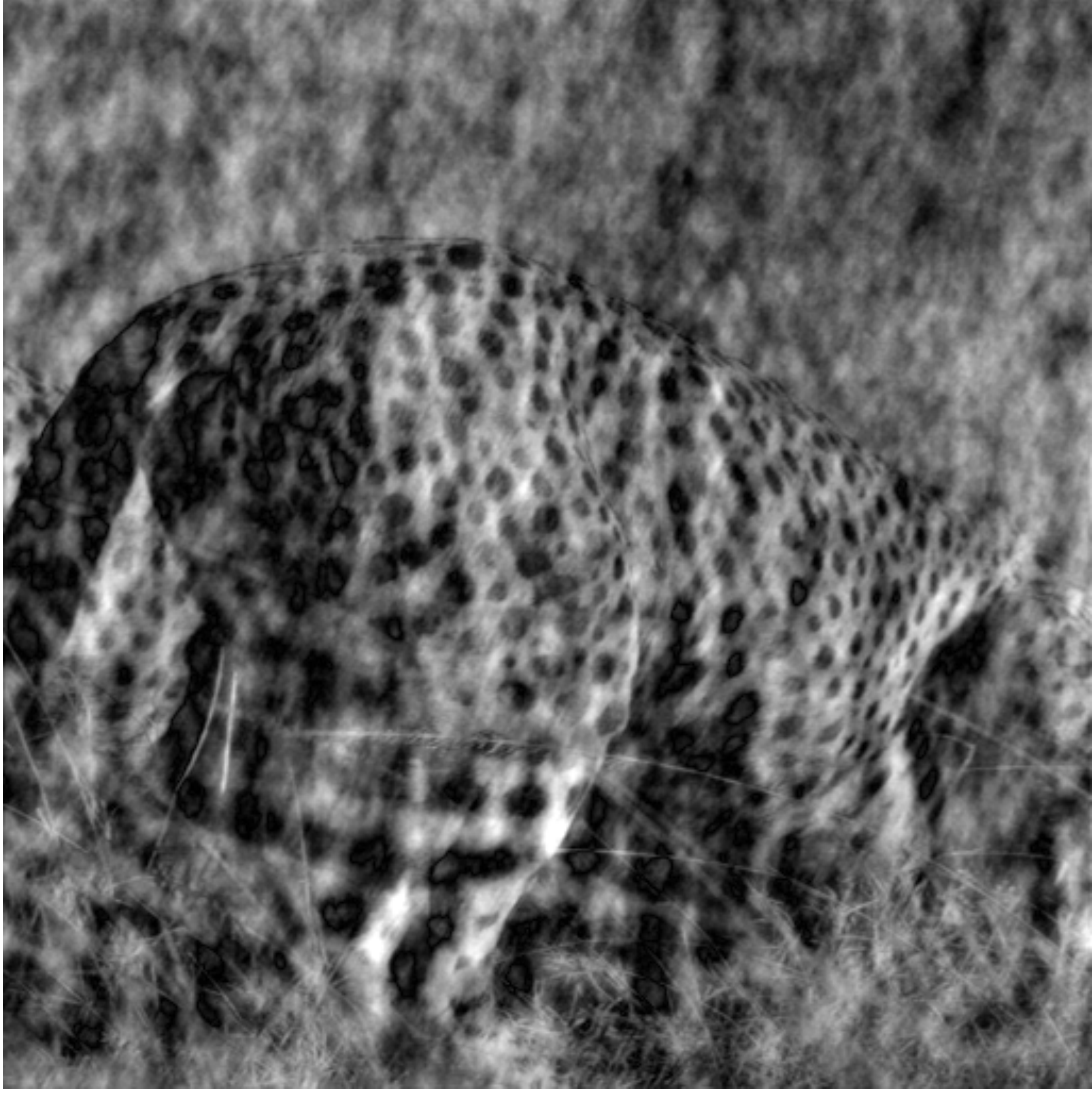
This is the phase transform of the zebra pic



Reconstruction with zebra phase, cheetah magnitude



Reconstruction
with cheetah phase,
zebra magnitude



Phase and Magnitude

Image with cheetah phase
(and zebra magnitude)

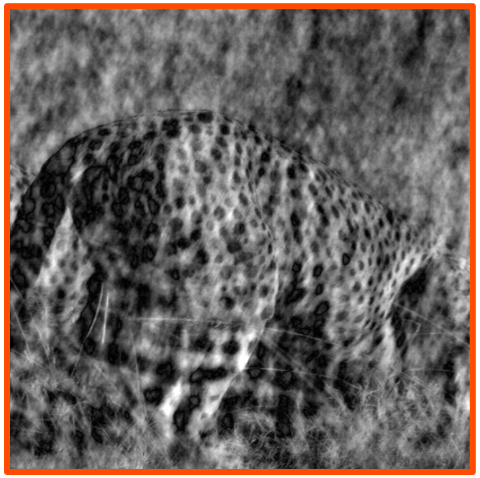
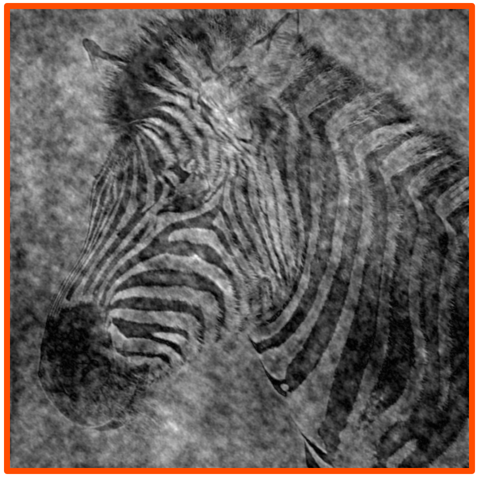
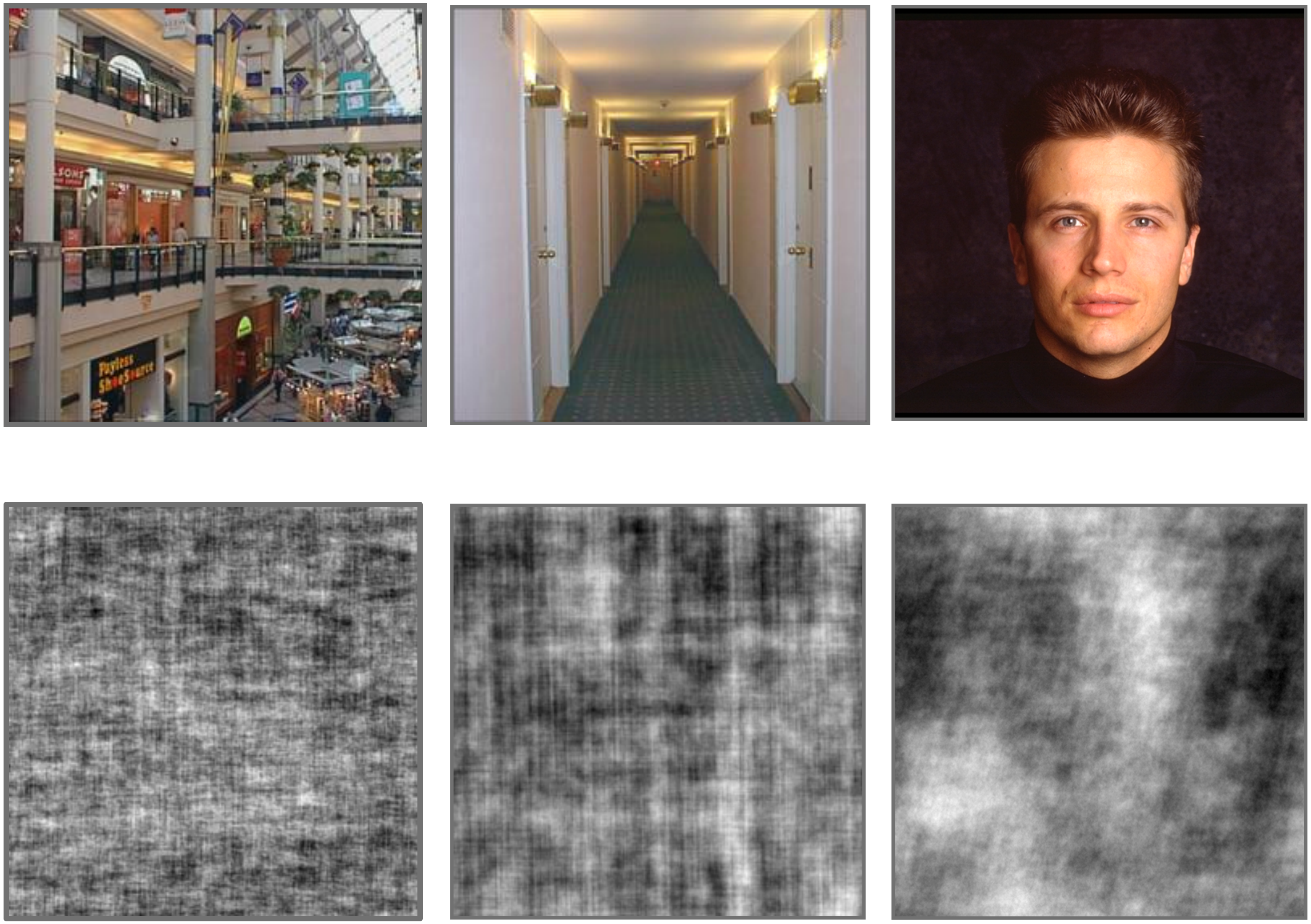


Image with zebra phase
(and cheetah magnitude)



Randomizing the phase

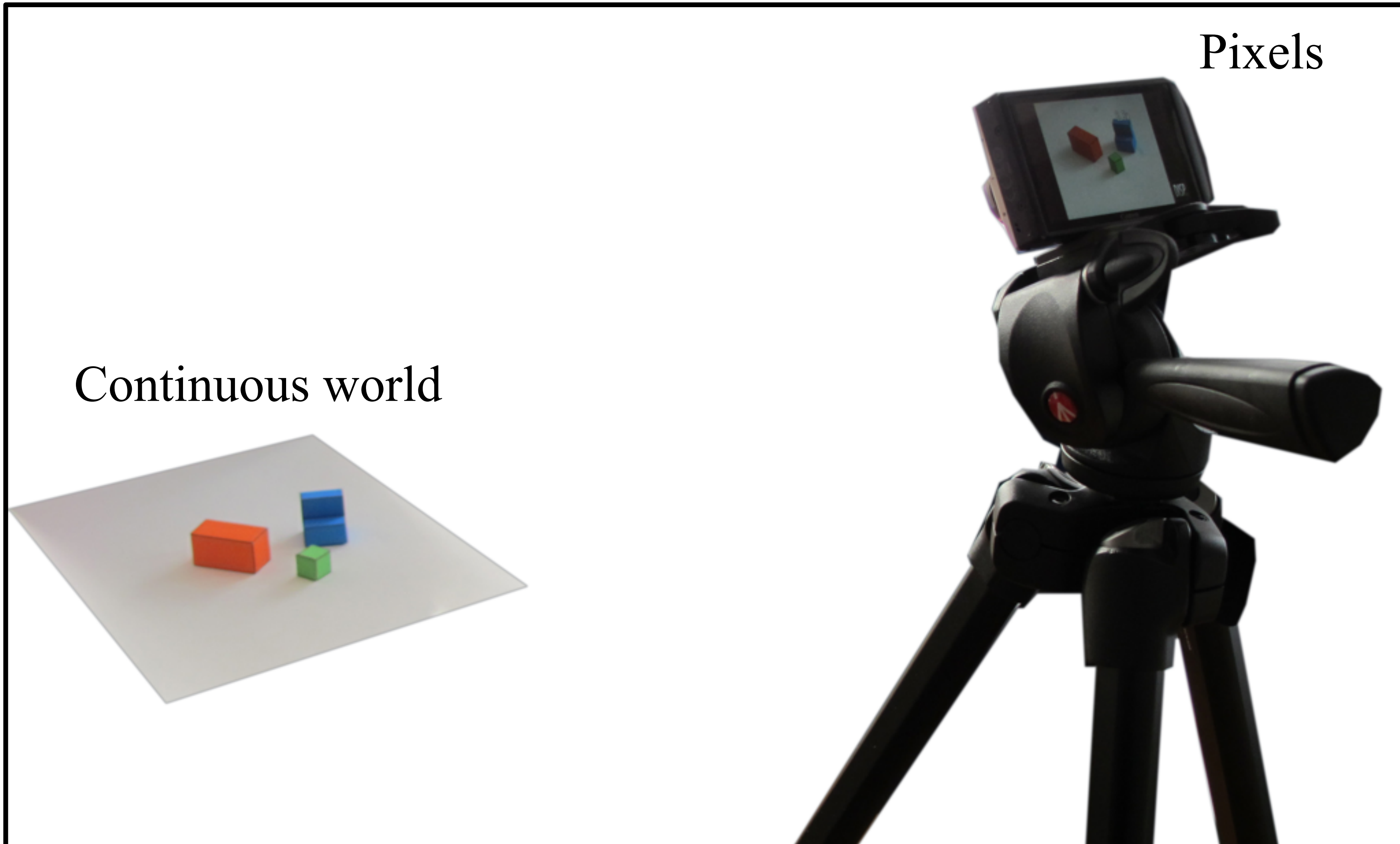


Outline

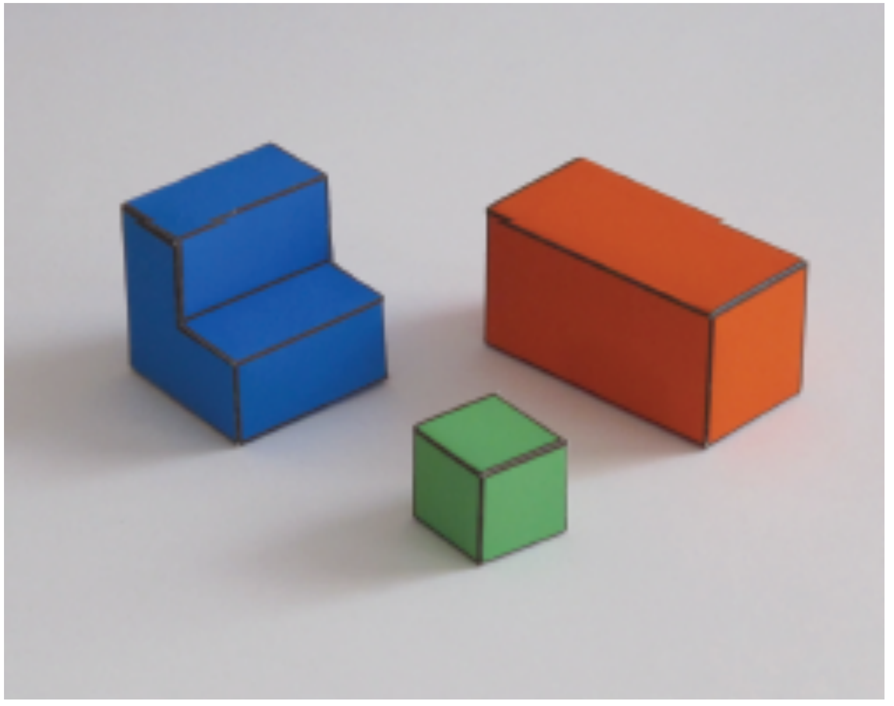
- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

121

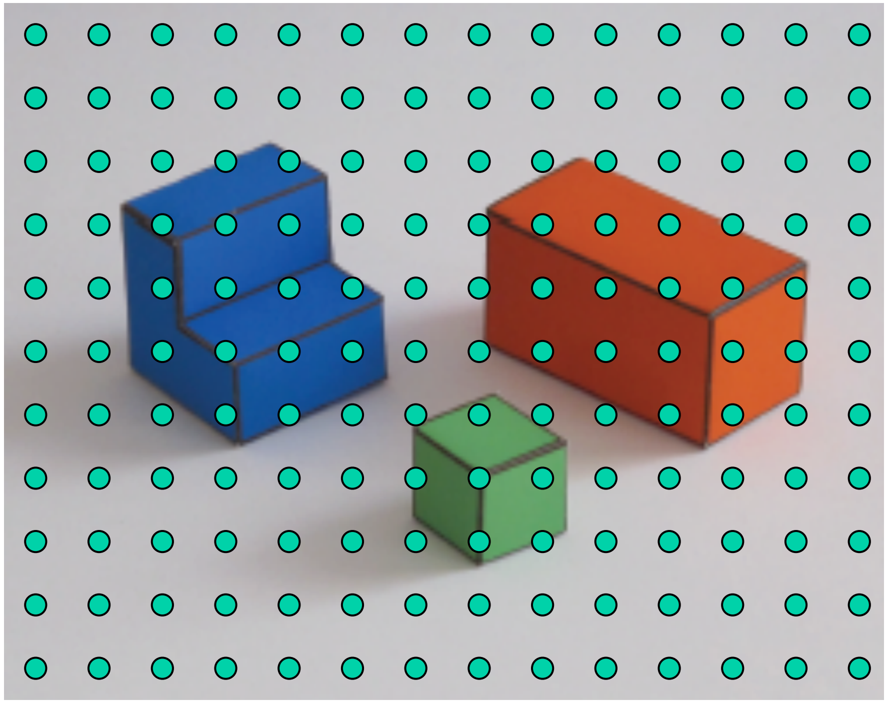
Sampling



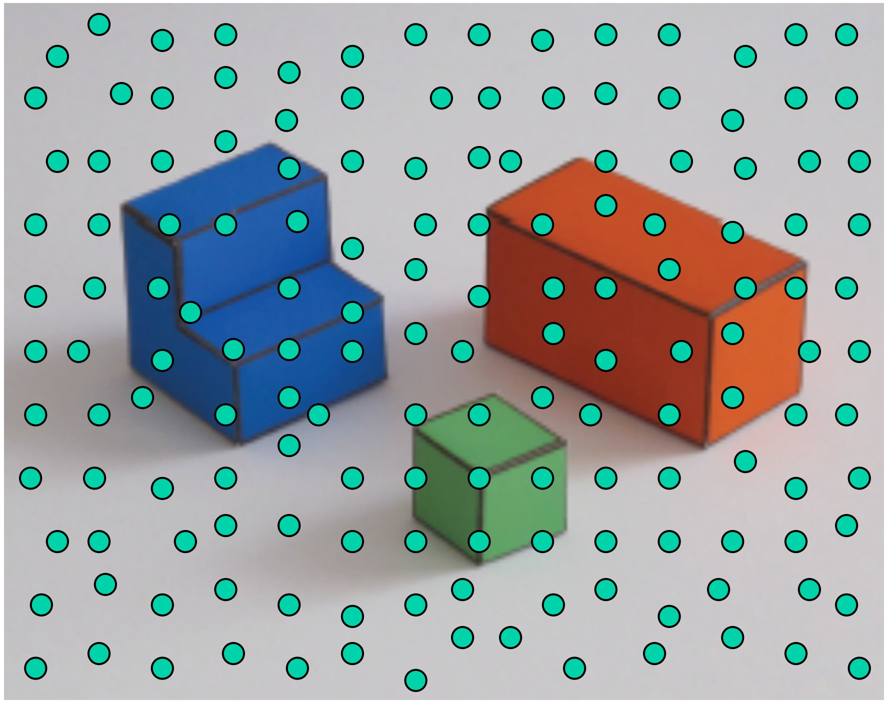
Sampling



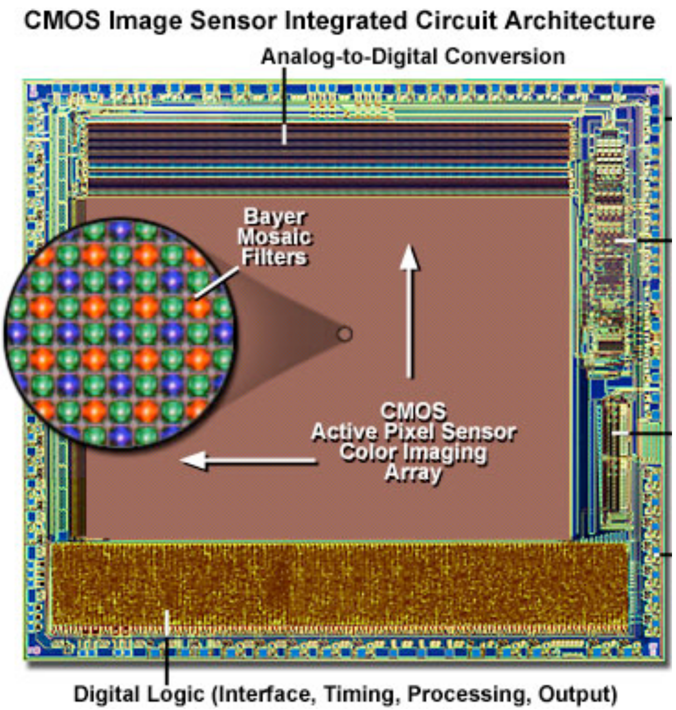
Sampling

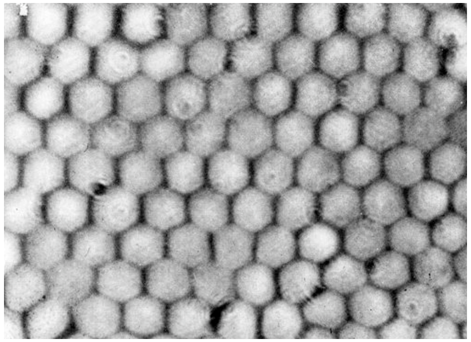


Sampling



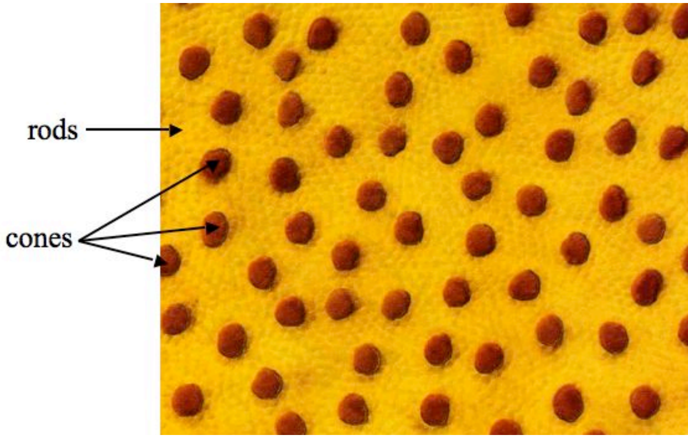
What will be the best sampling pattern in 2D?





Retinal fovea

Hexagonal



Retina periphery

Random

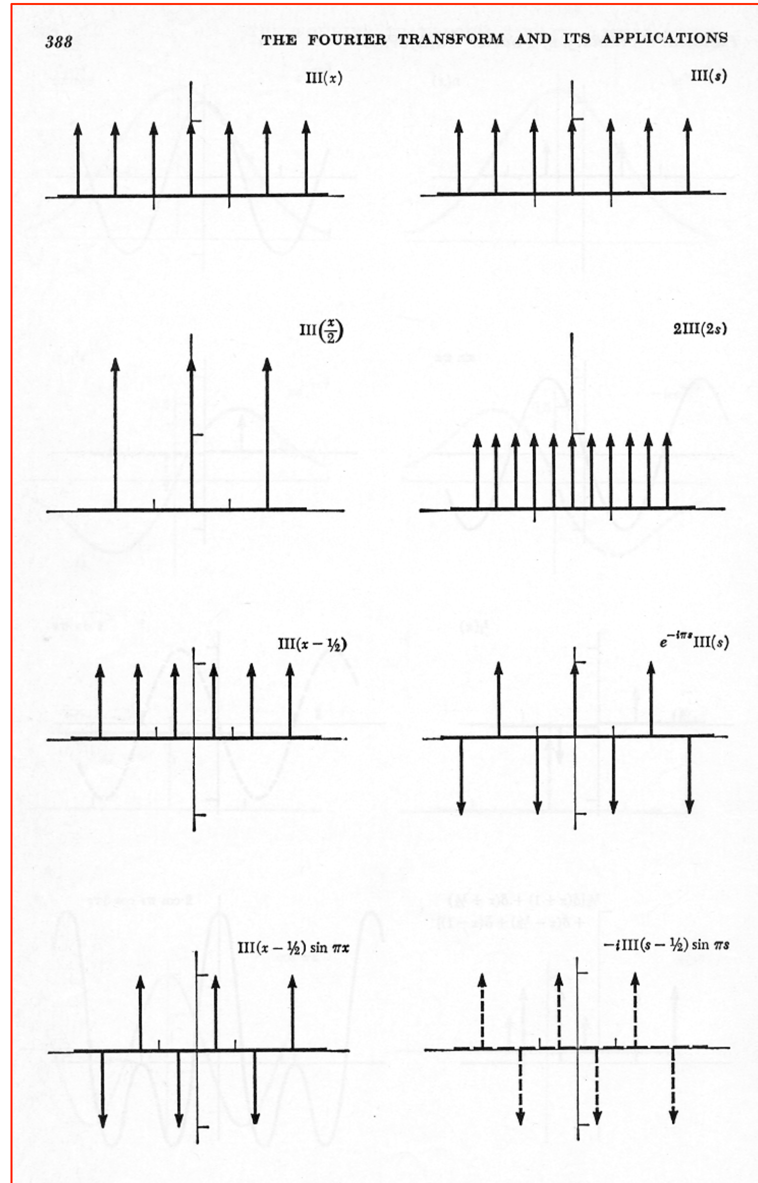
Images from: http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/retina/retina.html

The Fourier transform of a sampled signal

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

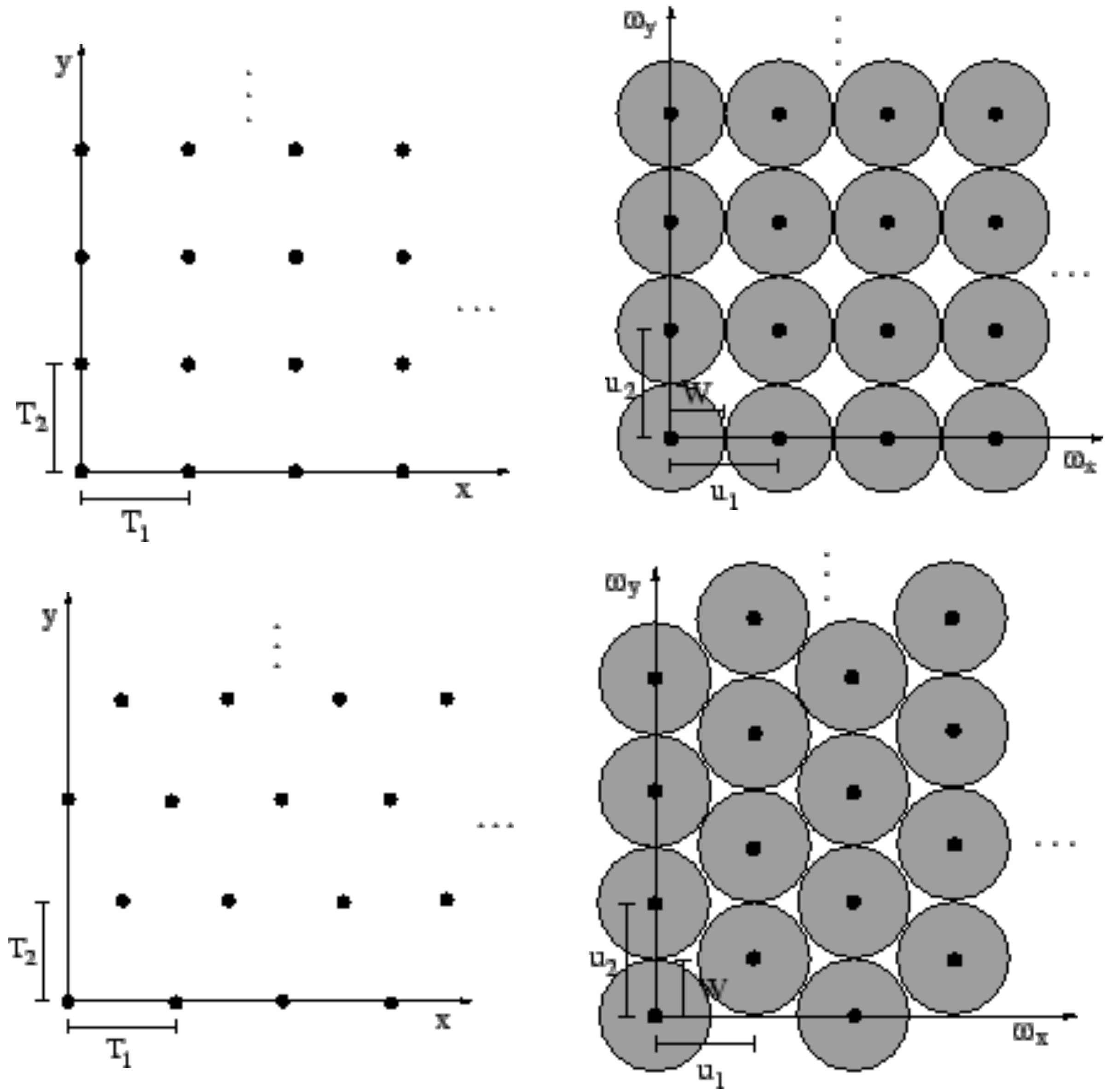
$$= F\left(f(x,y)\right) ** F\left(\sum_{i=-\infty}^{\infty}\sum_{i=-\infty}^{\infty}\delta(x-i,y-j)\right)$$

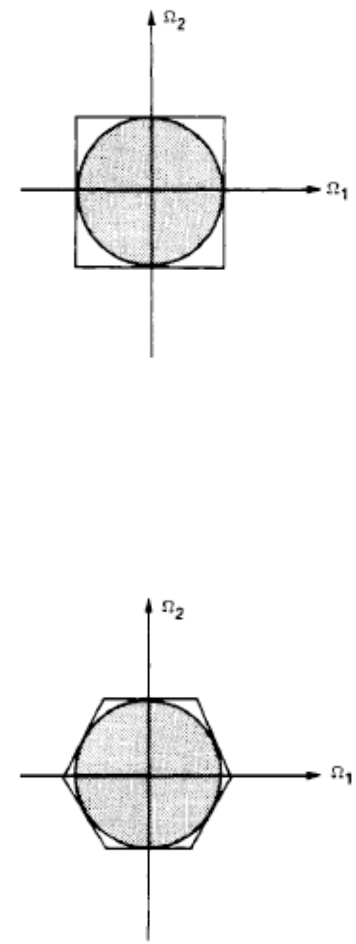
$$= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}F\left(u-i,v-j\right)$$



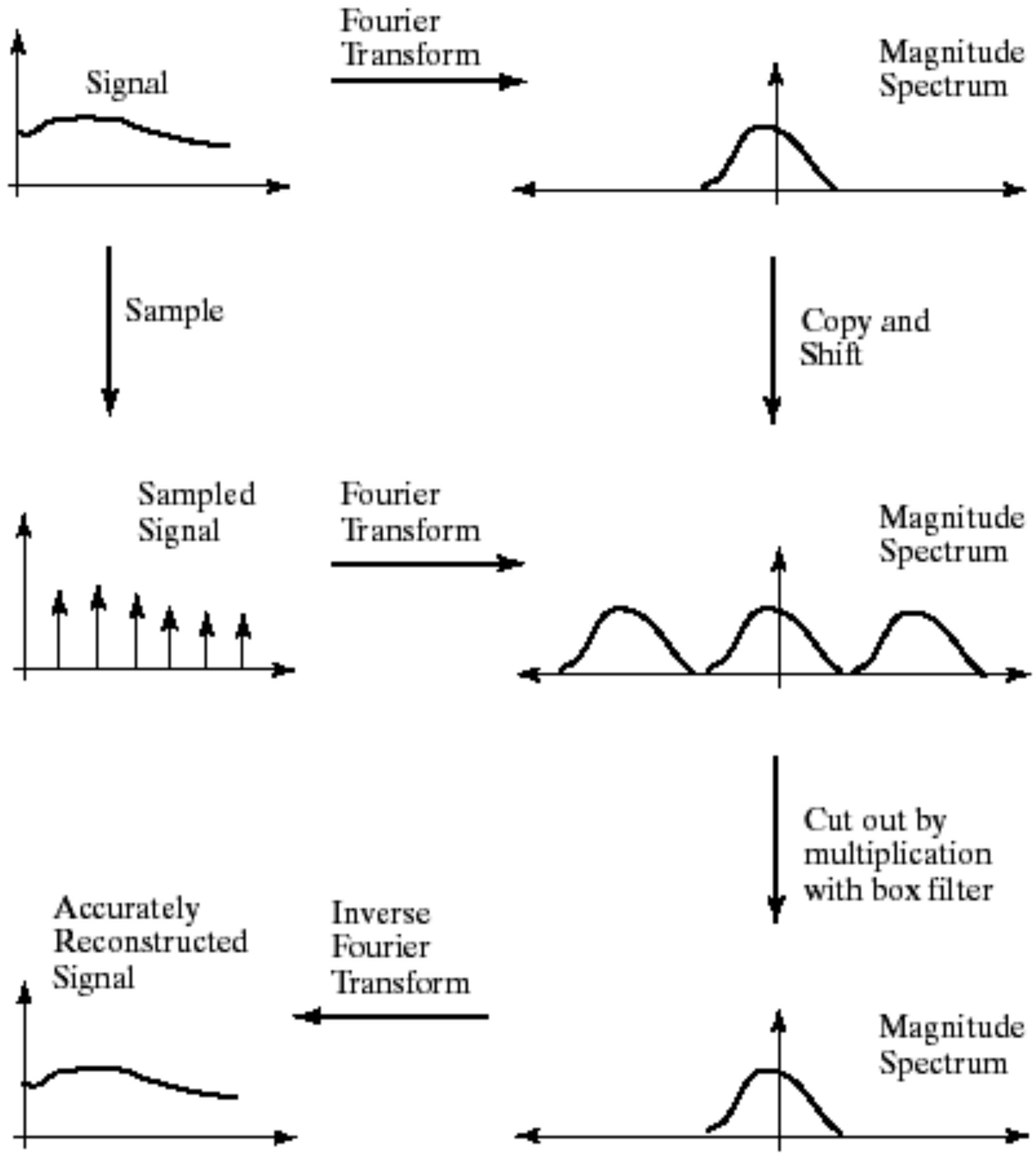
Bracewell, The Fourier Transform and its Applications, McGraw Hill 1978

Sampling

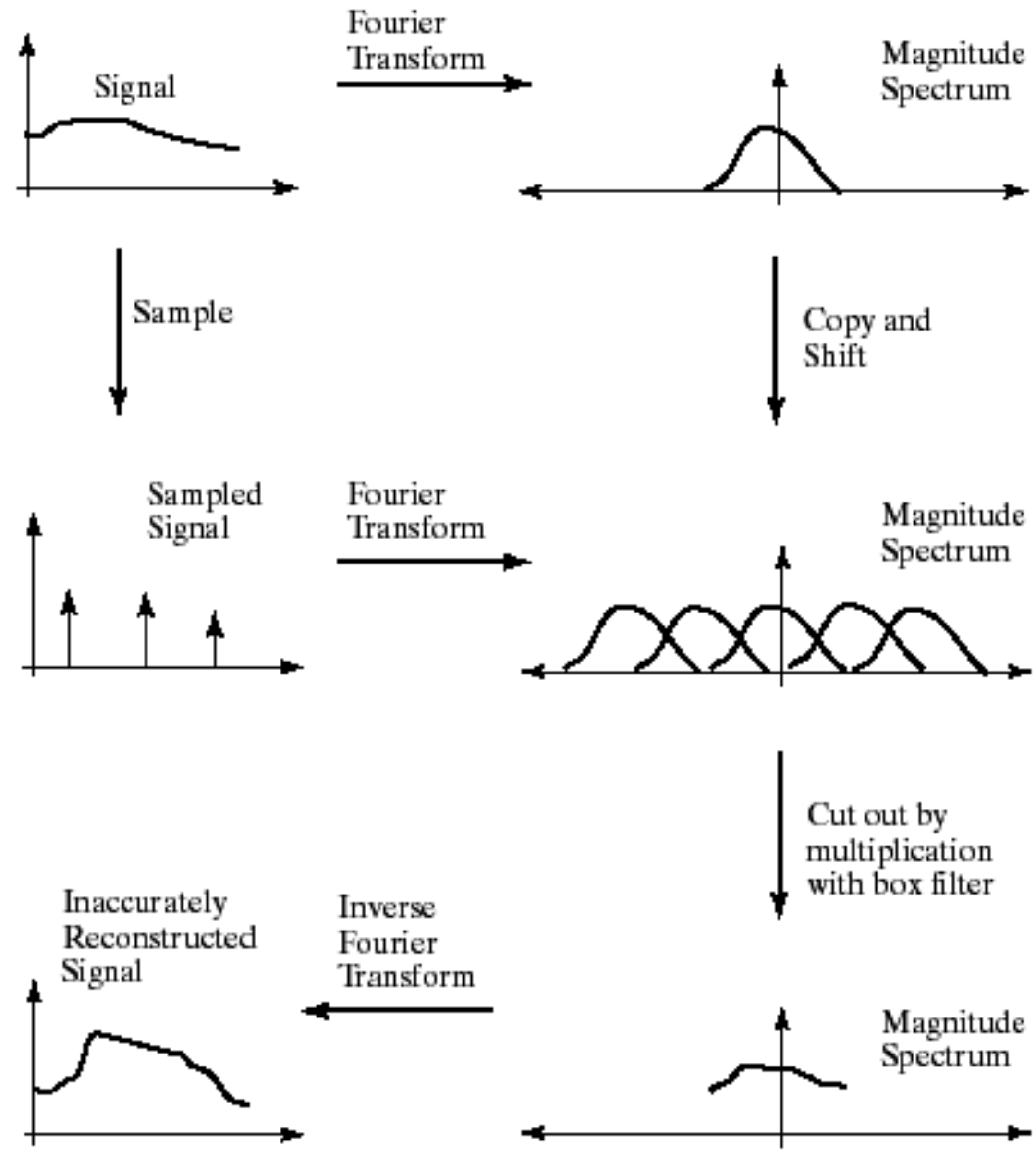




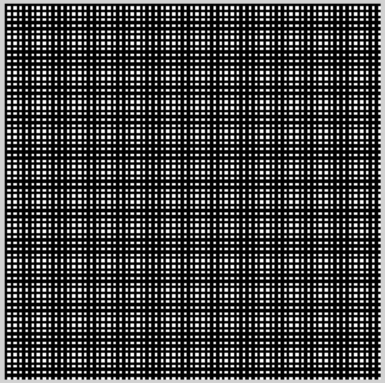
Mersereau, 1979



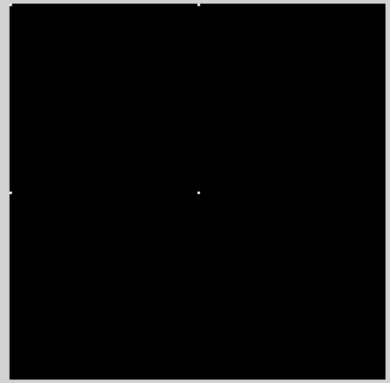
Signal
Fourier Transform
Magnitude Spectrum
Sample
Copy and Shift
Sampled Signal
Fourier Transform
Magnitude Spectrum
Cut out by multiplication with box filter
Accurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum


Signal
Fourier Transform
Magnitude Spectrum
Sample
Copy and Shift
Sampled Signal
Fourier Transform
Magnitude Spectrum
Cut out by multiplication with box filter
Inaccurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum

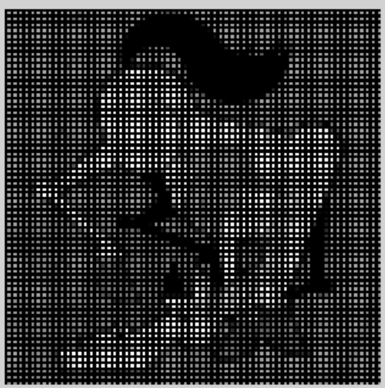
Sampling function



FT(Sampling function)

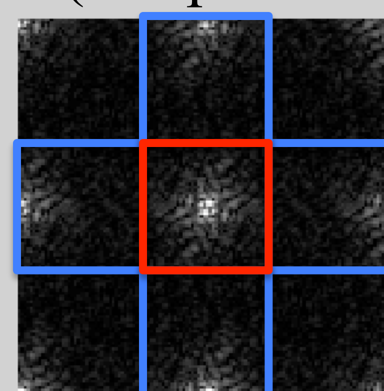


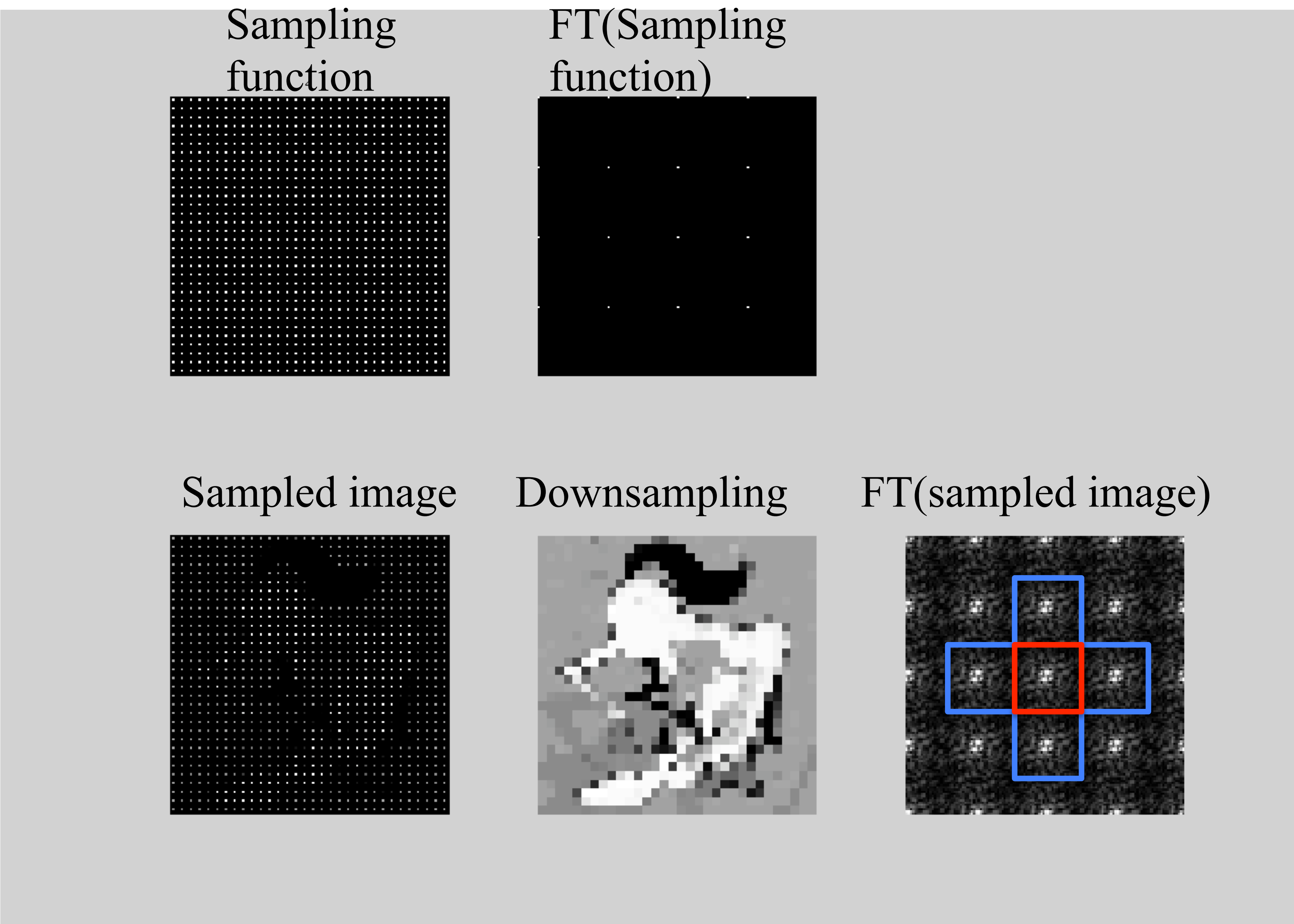
Sampled image

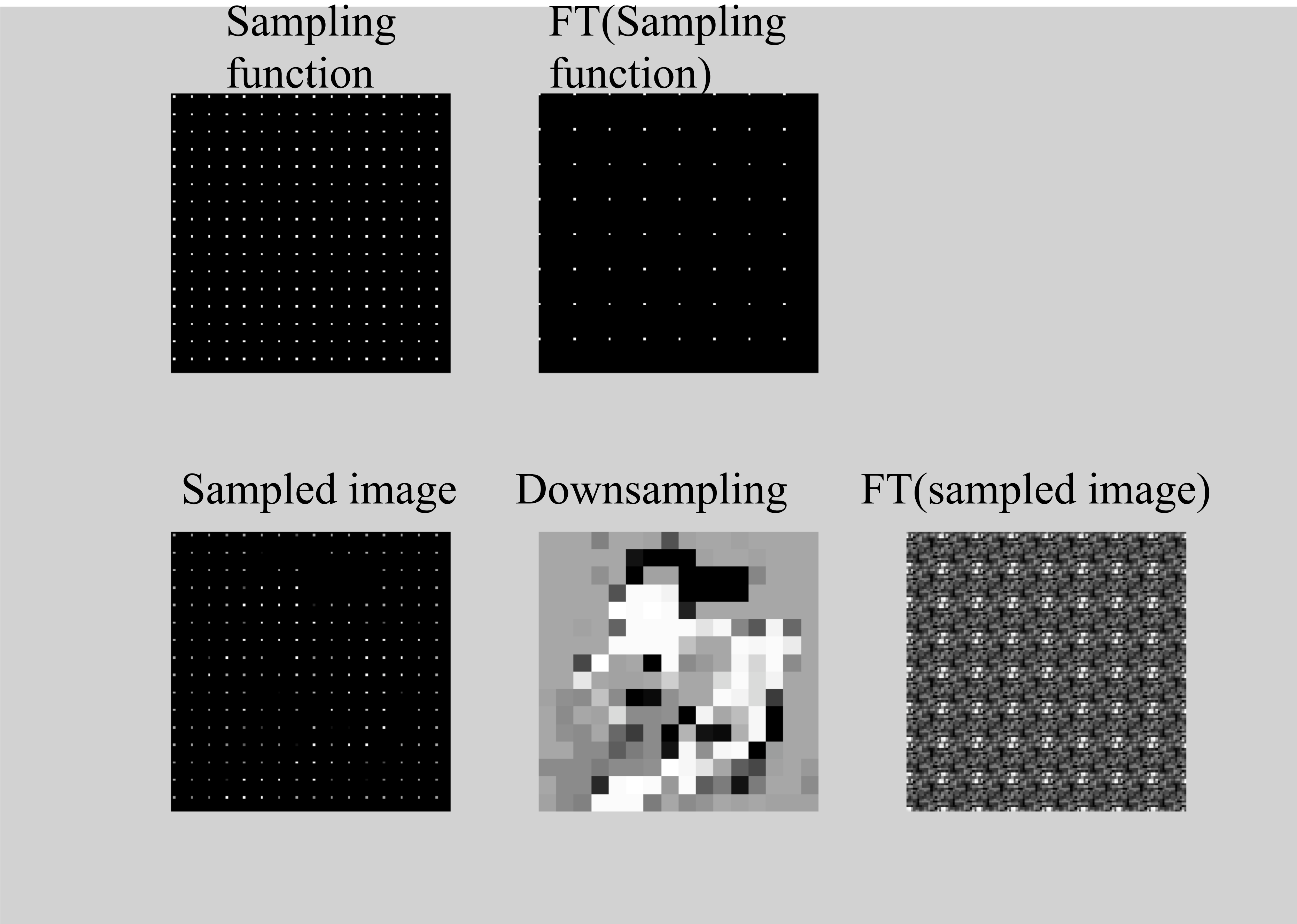


Downsampling

FT(sampled image)

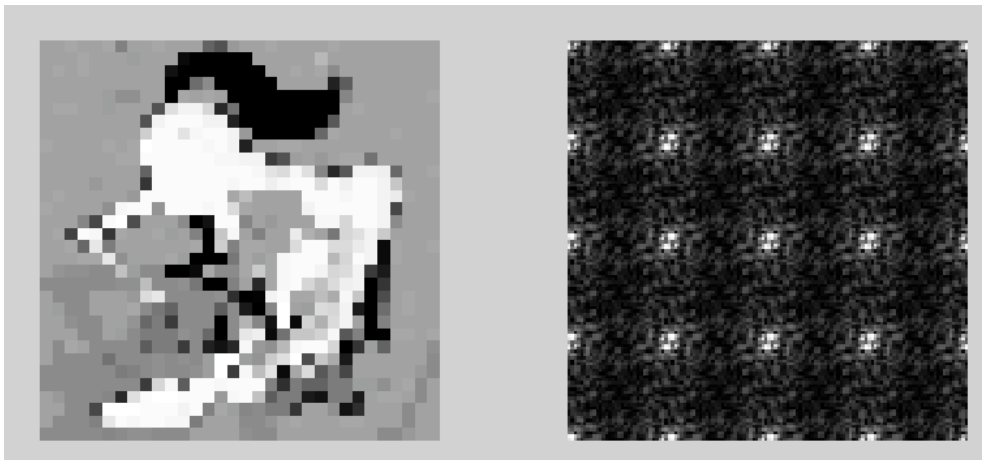



Sampling function
FT(Sampling function)
Sampled image
Downsampling
FT(sampled image)

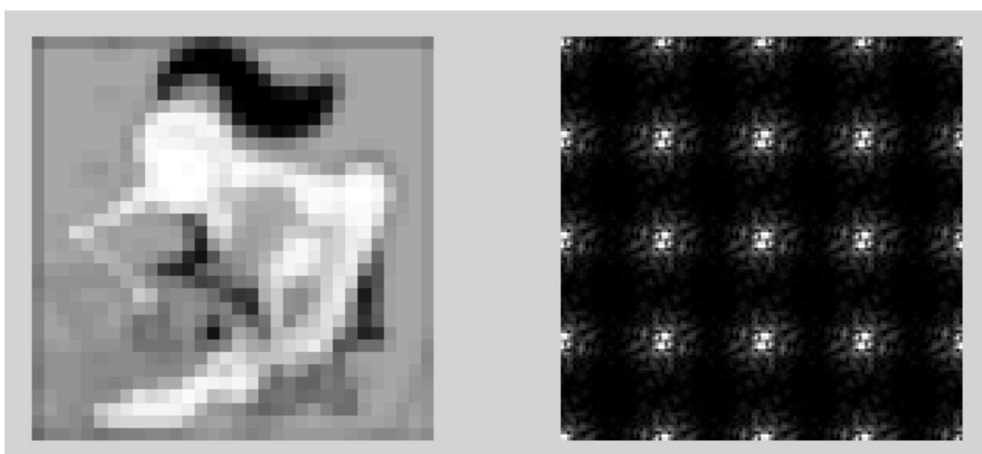

Sampling function
FT(Sampling function)
Sampled image
Downsampling
FT(sampled image)

Antialiasing filter

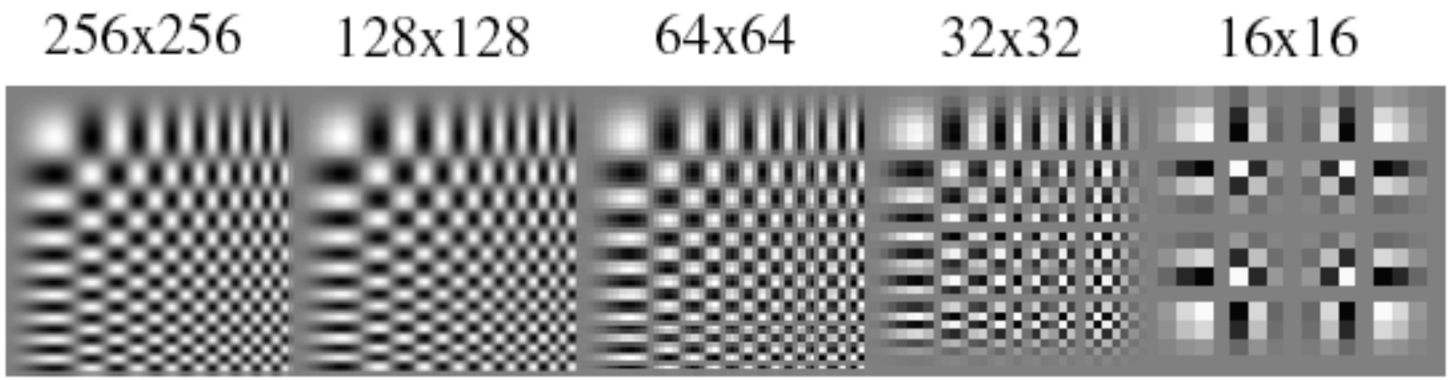
Without prefiltering



With prefiltering

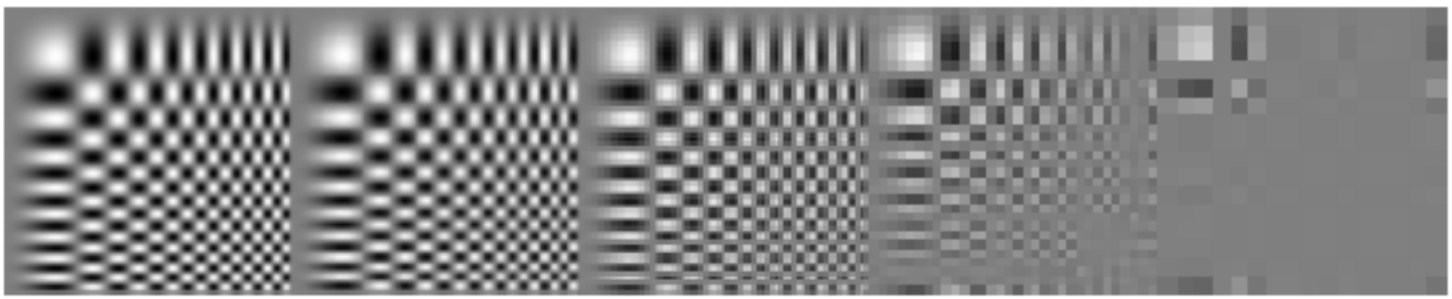


Sampling without smoothing. Top row shows the images, sampled at every second pixel to get the next.

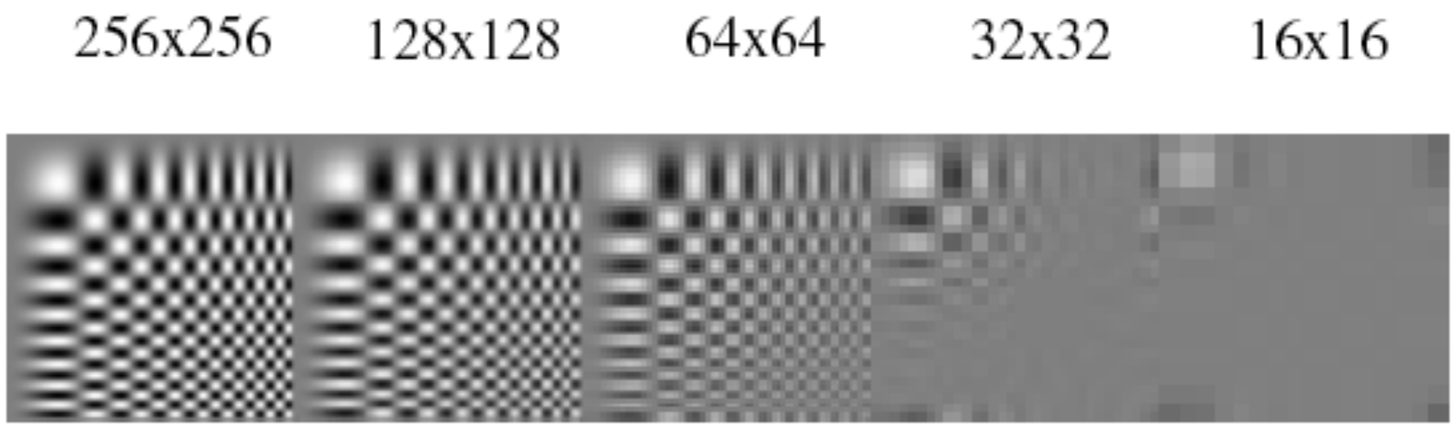


Sampling with smoothing. Top row shows the images. We
get the next image by smoothing the image with a Gaussian with sigma 1 pixel,
then sampling at every second pixel to get the next.

256x256 128x128 64x64 32x32 16x16



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.



Outline

- Linear filtering
- Fourier Transform
- Human spatial frequency sensitivity
- Phase
- Sampling and Aliasing
- Spatially localized analysis

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.