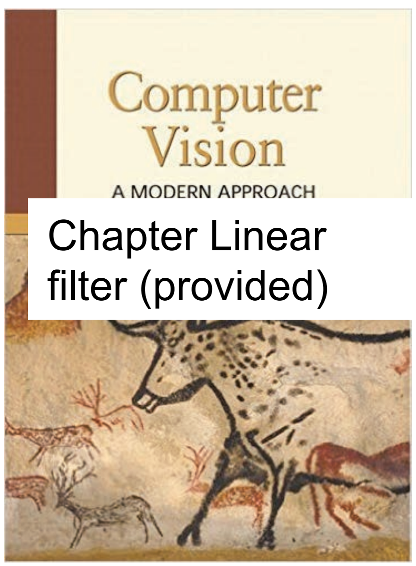
6.819 / 6.869: Advances in Computer Vision

Basics of Image Processing I:

Points operators; linear filtering; fourier transform

Website:

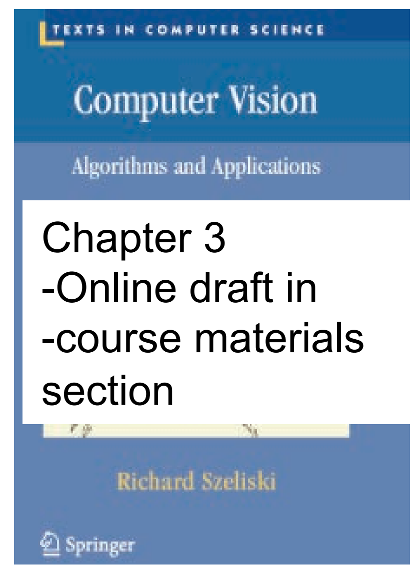
http://6.869.csail.mit.edu/fa15/



Chapter Linear filter (provided)

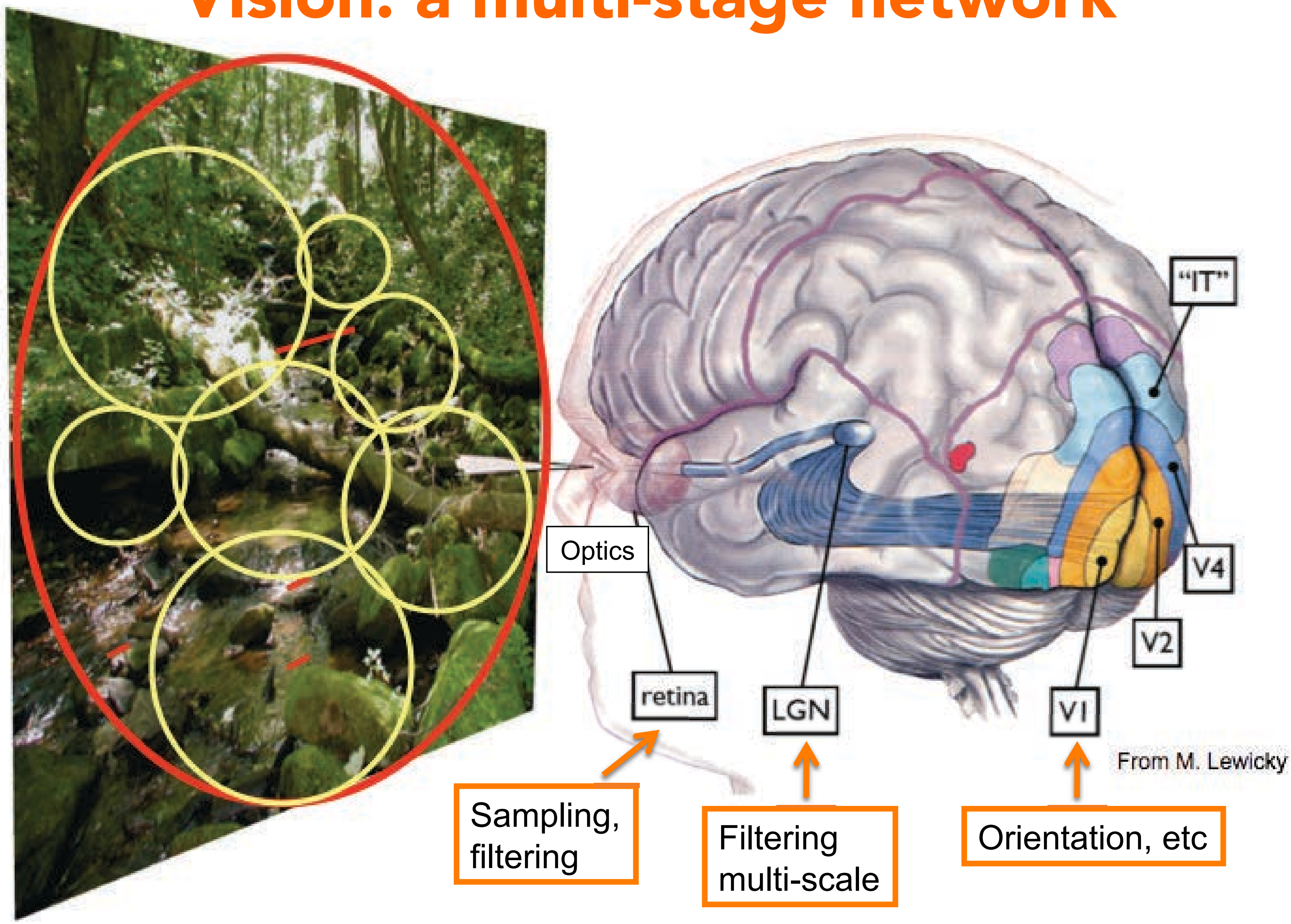
Instructor: Aude Oliva

Lecture TR 9:30AM – 11:00AM
(Room 34-101)



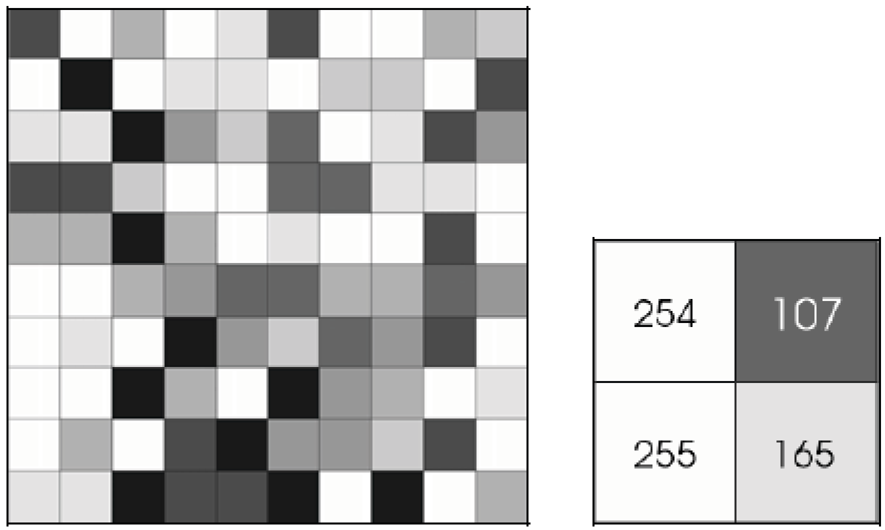
Chapter 3
-Online draft in
-course materials section

Vision: a multi-stage network



What is an image?

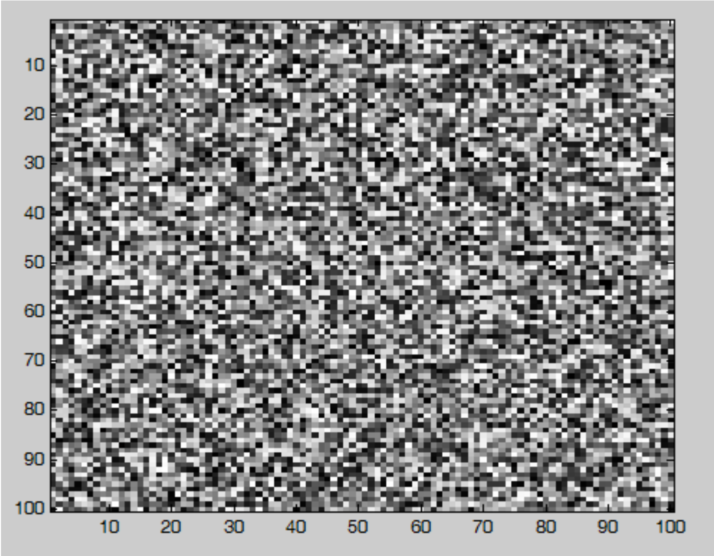
In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

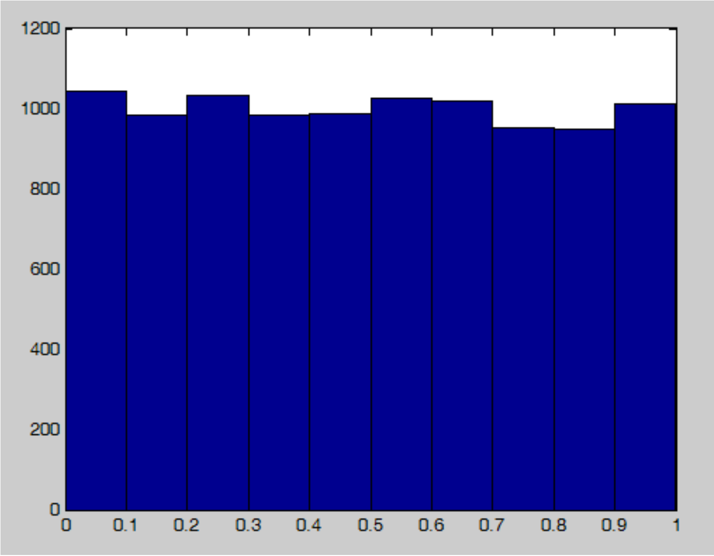


Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 tones or greyscales.

A normal greyscale image has 8 bit colour depth = 256 greyscales. A "true colour" image has 24 bit colour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours.

A random visual world: Noise Image

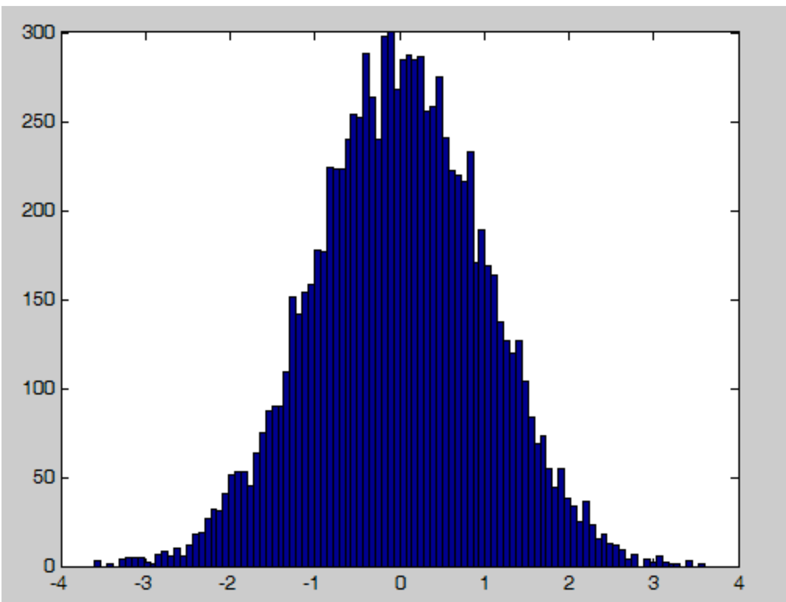




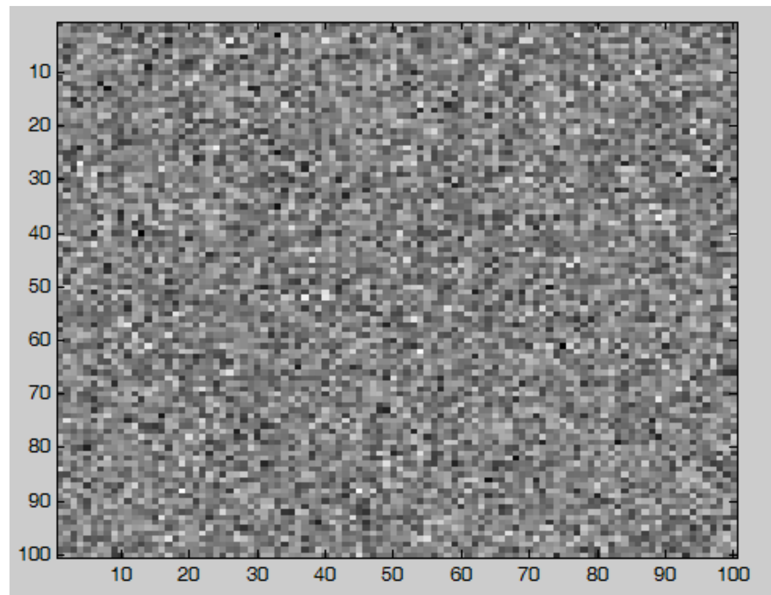

```
noise=rand(100,100);
imagesc(noise)
colormap(gray(256))
```

```
noise1d=noise(:);
size(noise1d)
Figure; hist(noise1d)
```

A prior-based world: Gaussian Noise



Gaussian noise

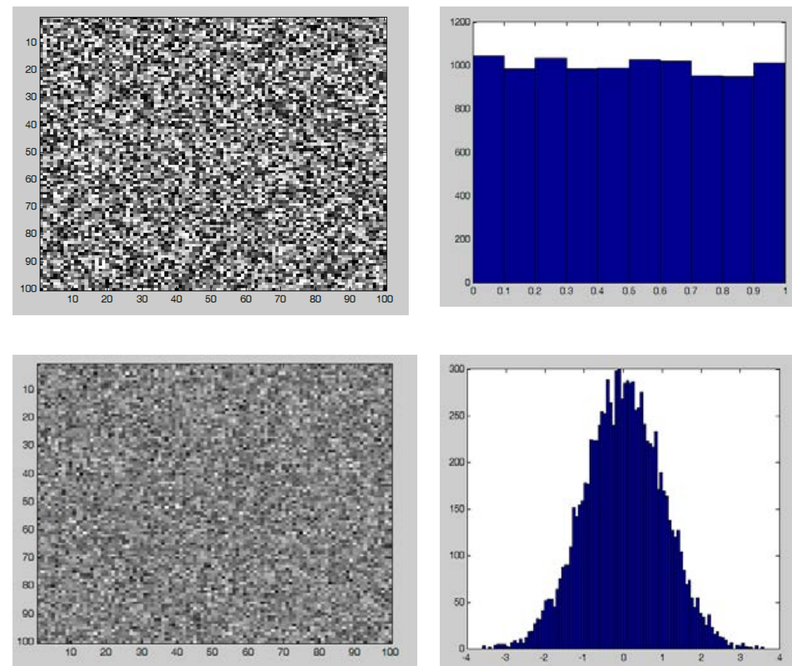


randomgenerator = **randn**(10000,1);
hist(randomgenerator, 100);

randomimage = **randn**(100,100);
imagesc(randomimage);
colormap(gray(256))

Random noise and Gaussian noise are *White* noise

- White noise is a source of random numbers, uniformly distributed with no correlation whatsoever between successive numbers (pixels).
- White noise is never the same twice
- In some applications (e.g. generating textures in computer graphics), pseudo-random noise is desirable



Properties of Noise

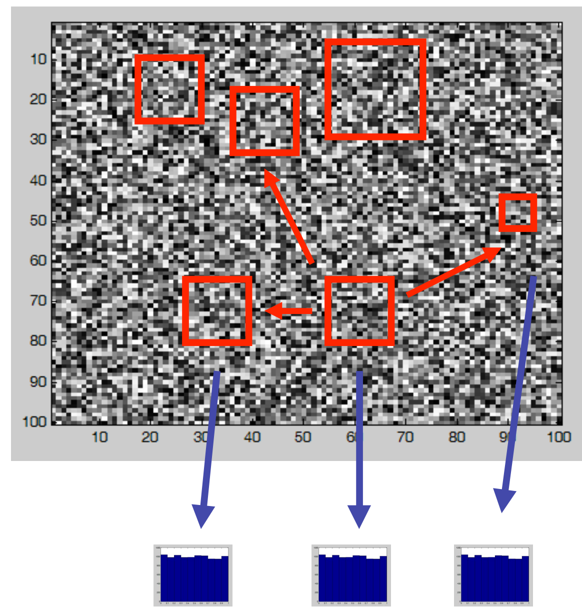
- Noise is **stationary**: its statistical character is translationally invariant.

  Stationarity is the property of a random process which guarantees that its statistical properties, such as the **mean** value, its **moments** and **variance**, will not change over time or space.

  A stationary process is one whose probability distribution is the same at all times/location.

- Noise is **isotropic**: its statistical character should be rotationally invariant. A noise is is said to have rotational invariance if its value does not change when arbitrary rotations are applied to it

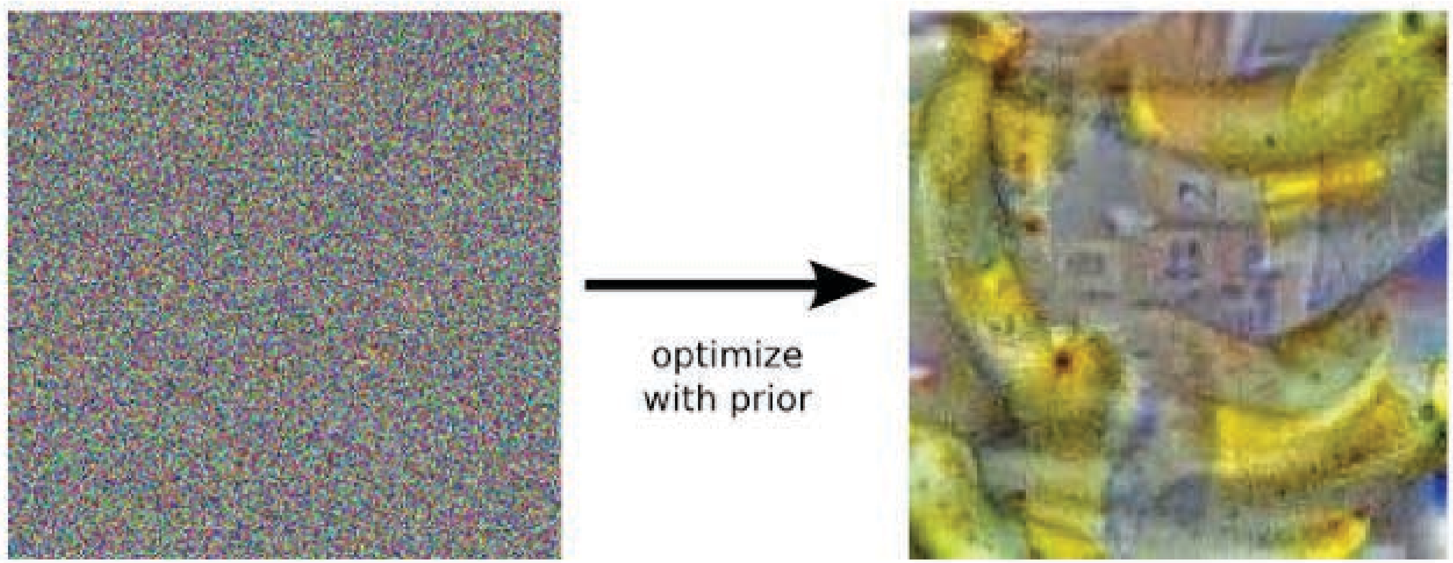
  Isotropy is uniformity in all directions. Precise definitions depend on the subject area. The word is made up from Greek *iso* (equal) and *tropos* (direction).



Why is noise image an important concept ?

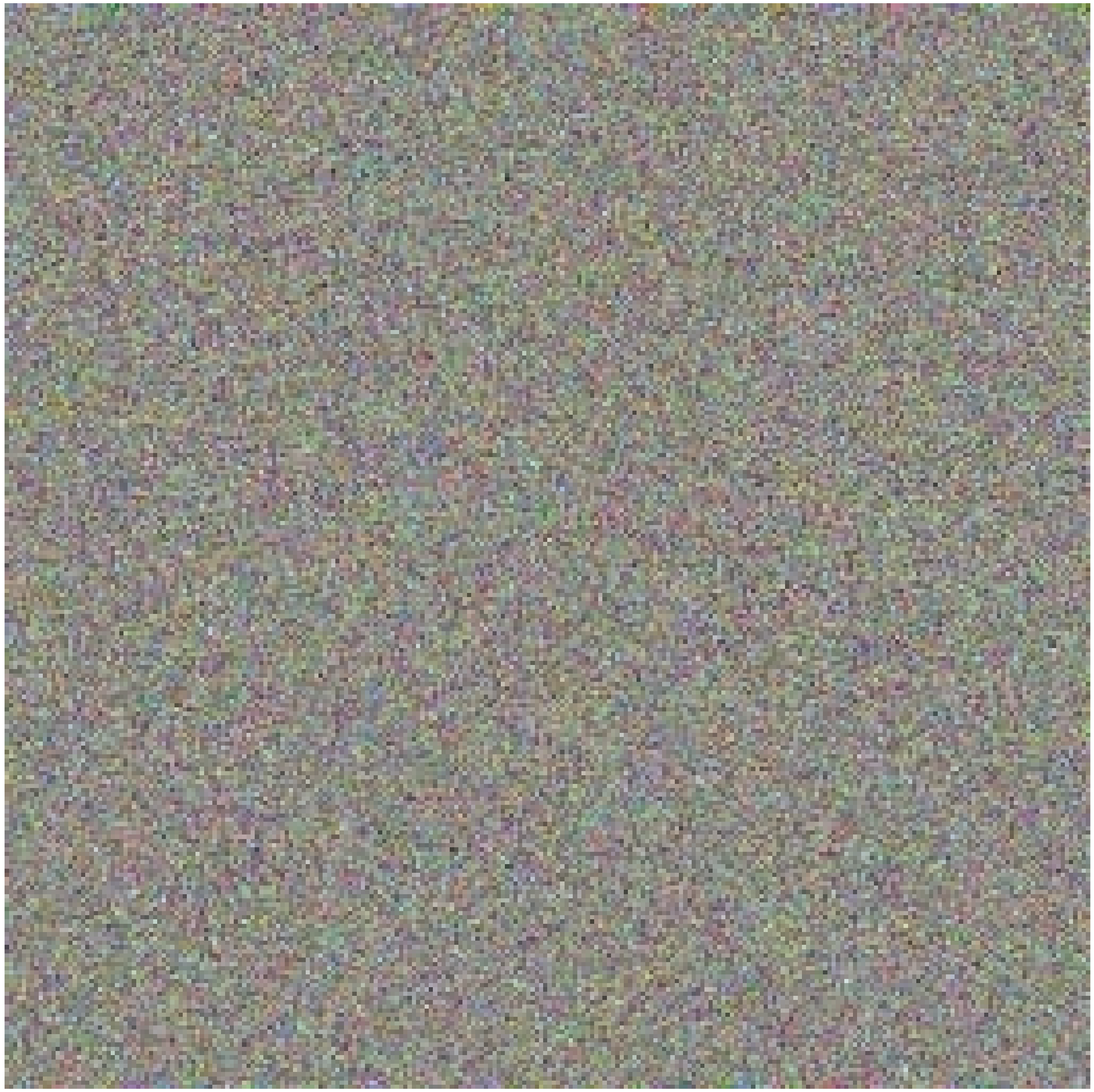
Inceptionism: Reconstructing what a neural network "imagines"

image/texture synthesis, image regeneration



How does a deep learning network see a "banana"? Start with a random noise image, then gradually tweak the image towards what the neural net considers a banana. It works "well enough" if we impose a prior constraint that the image should have similar statistics to natural images, such as neighboring pixels needing to be correlated.

http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html



Slide from Zisserman/Vedaldi

Simonyan, Vedaldi, Zisserman, ICLR 2014

What happens if you start from a different noise image?

Slide from Zisserman/Vedaldi

I- Points Operators

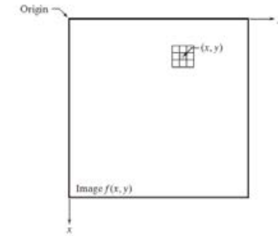
The simplest kinds of image processing transforms:

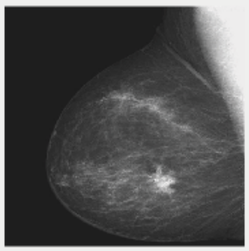
Each **output pixel**'s value depends only on the corresponding **input pixel value** (brightness, contrast adjustments, color correction and transformations)

Intensity Transformation

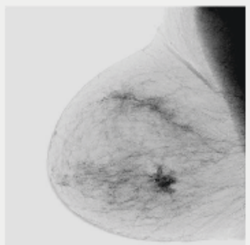
Intensity of gray level transformation function

IntensityEqualization/demoIntensity.m

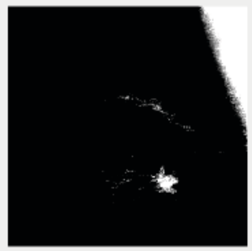




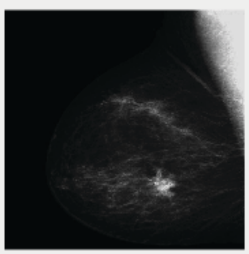
Original digital mammogram



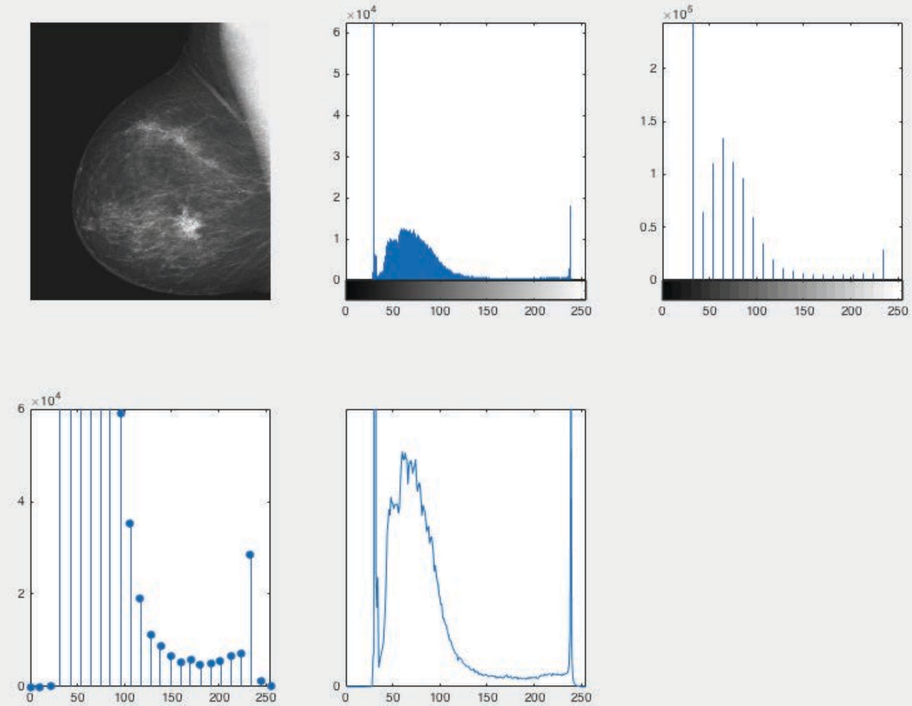
Negative image



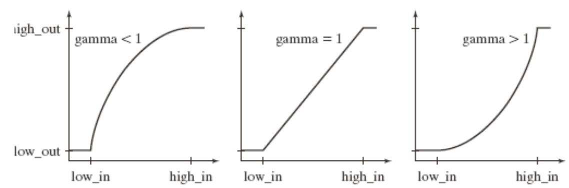
Result: expanding Intensities in the range [0.5,0.75]



Result: enhancing the image with gamma -2



gamma specifies the shape of the curve that maps the intensity. gamma is less than 1, the mapping is weighted toward higher (b output values. If gamma is greater than 1, the mapping is weigh toward lower (darker) output values.

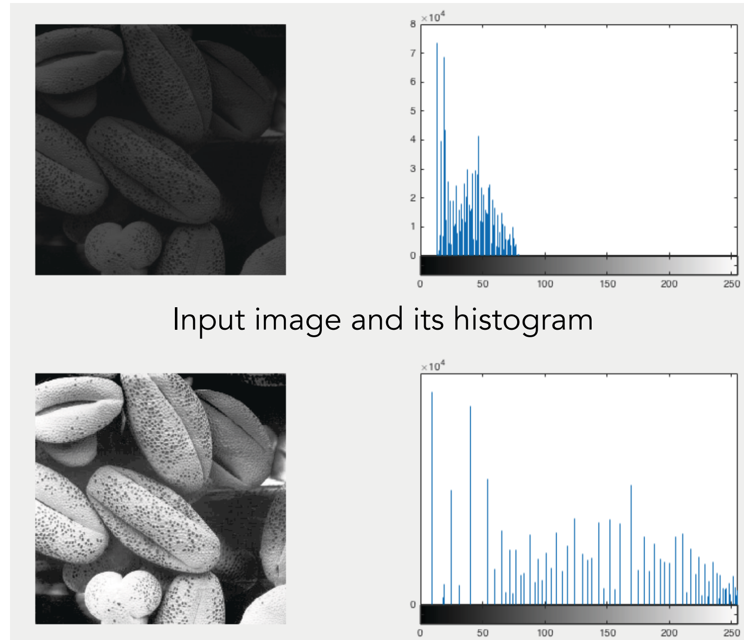


Histogram Equalization & Scaling

- **Intensity level equalization** process is an image with <u>increased dynamic range</u> which will tend to have a <u>higher contrast</u>.

  The process creates an image whose intensity cover the entire range [0 1] (or 0-255).

IntensityEqualization/demoIntensity.m, Part II

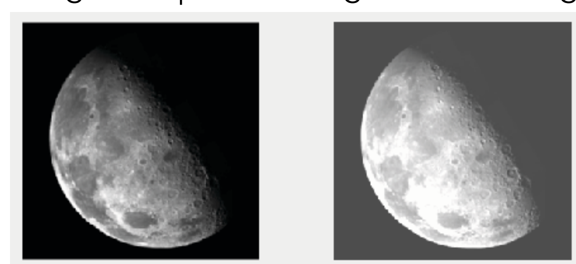


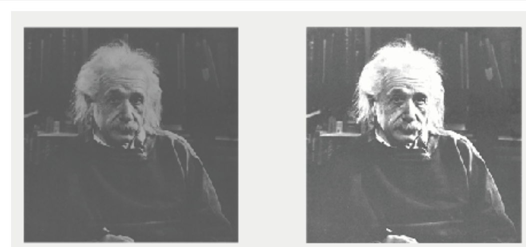
Input image and its histogram

Histogram equalized image and its histogram

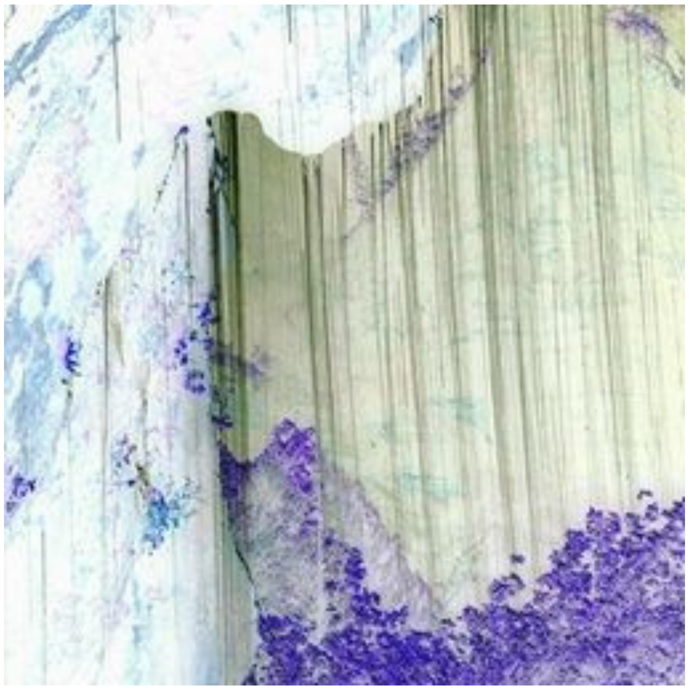
- **Intensity Scaling** is a less drastic intensity transformation that works for most images

IntensityScaling





For human vision, pixels inversion may change the entire interpretation of the image ..



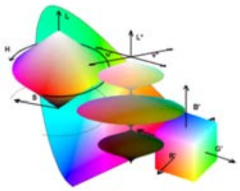
Textile, cloth, curtain
Indoor, close up view

Forest, waterfall
Outdoor, distant view

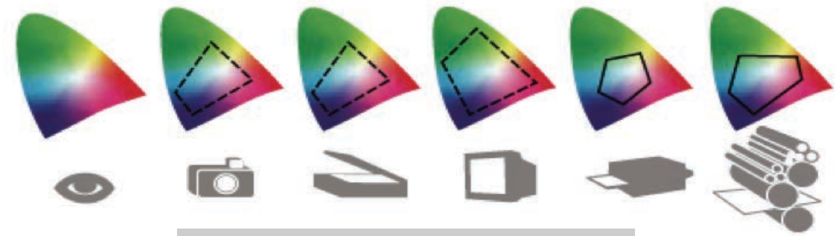
Image Enhancement

Images courtesy of Tobey Thorn

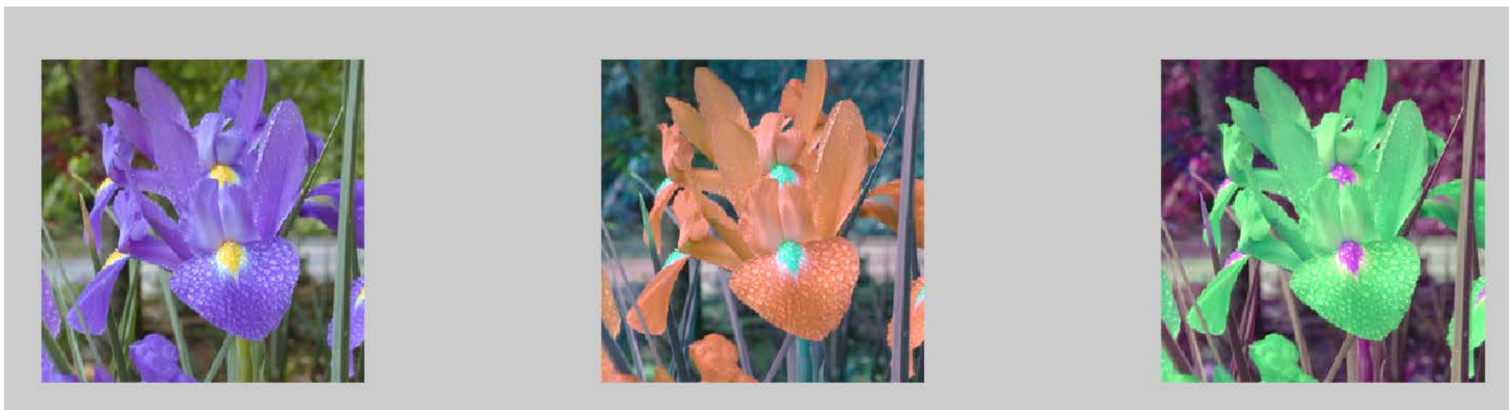
- Often used to increase the contrast in images that are overly dark or light
- Enhancement algorithms often play to humans' sensitivity to contrast
- More sophisticated algorithms enhance images in a small neighborhood, allowing overall better enhancement.

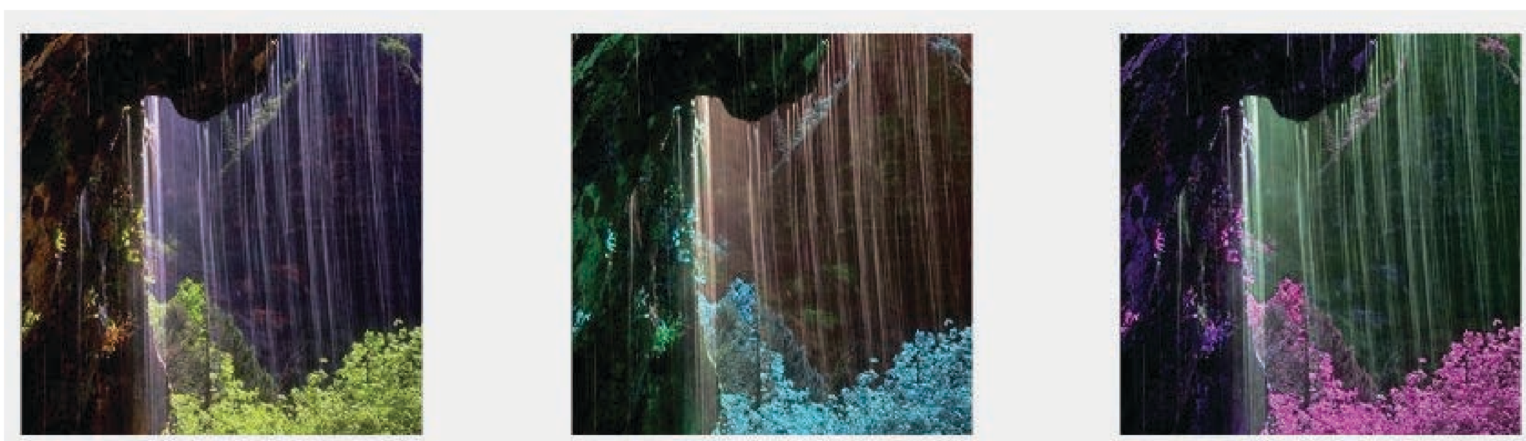


Color Spaces



ColorTransformation/SwapColor.m





II - Linear Filtering

Goal: Remove unwanted sources of variation, and keep the information relevant for whatever task we need to solve

Approach: Modify the pixels in an image based on some function of the local neighborhood around each pixel

What can filters do?

- Smooth or sharpen
- Remove noise
- Increase/decrease image contrast
- Enhance edges, detect particular orientations
- Detect image regions that match a template

Linear filtering

For a general 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

H is usually called the kernel

Operation is called convolution

Convolutional (Linear) Filtering

Operations that are <u>spatially invariant</u>

Rectangular Filter (box)

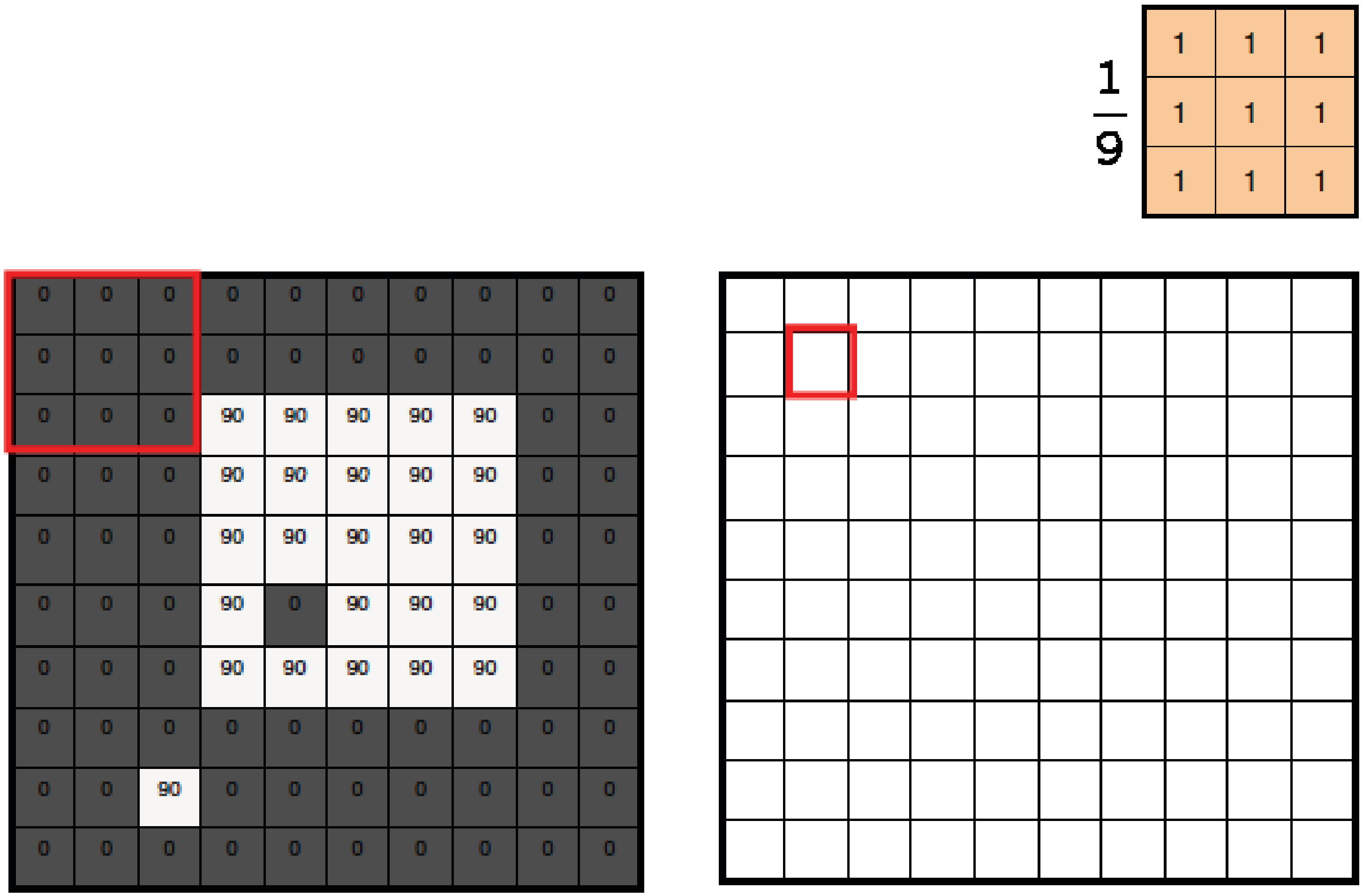
smoothing by averaging

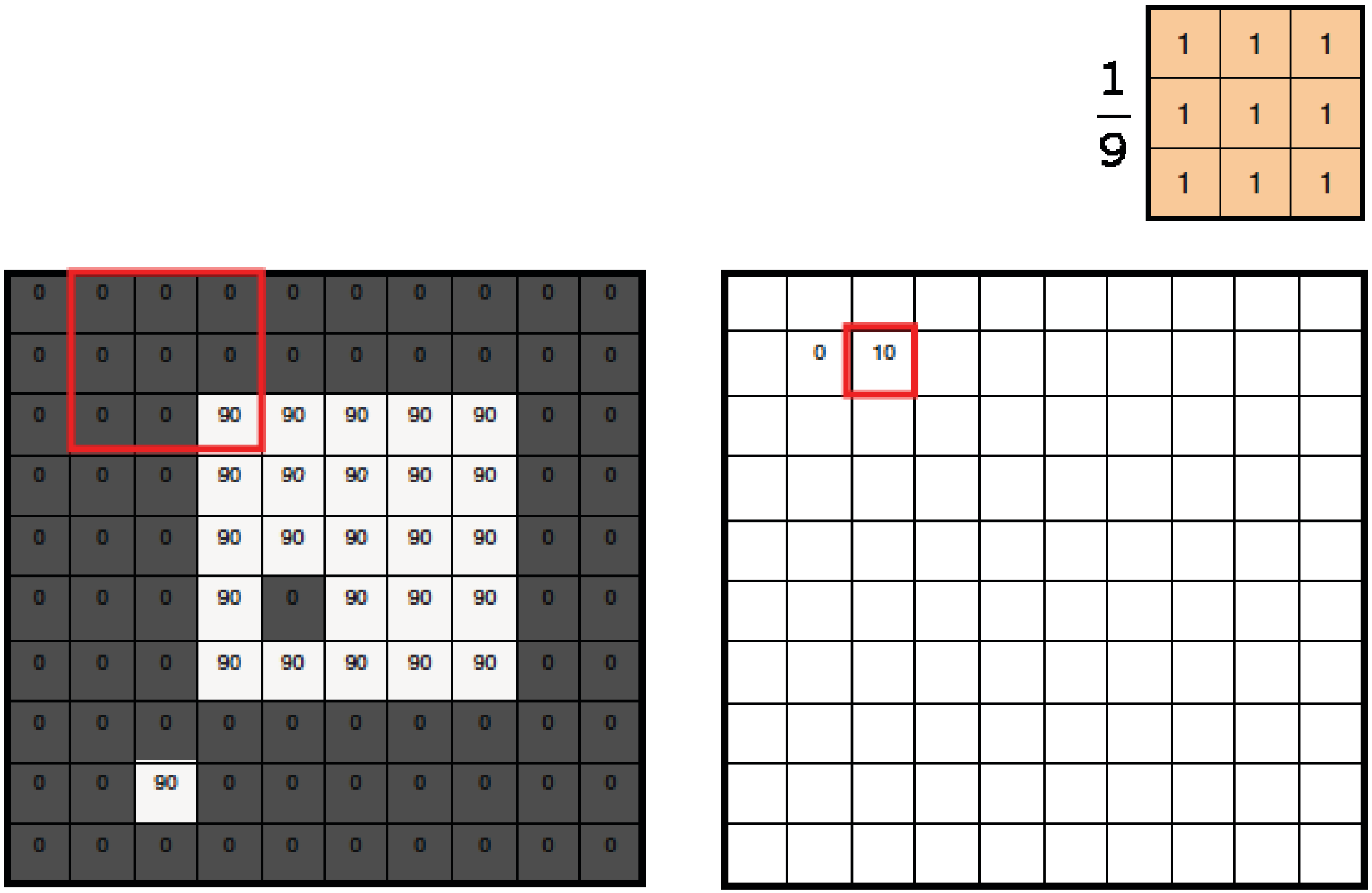
g[m,n]

⊗ ? =

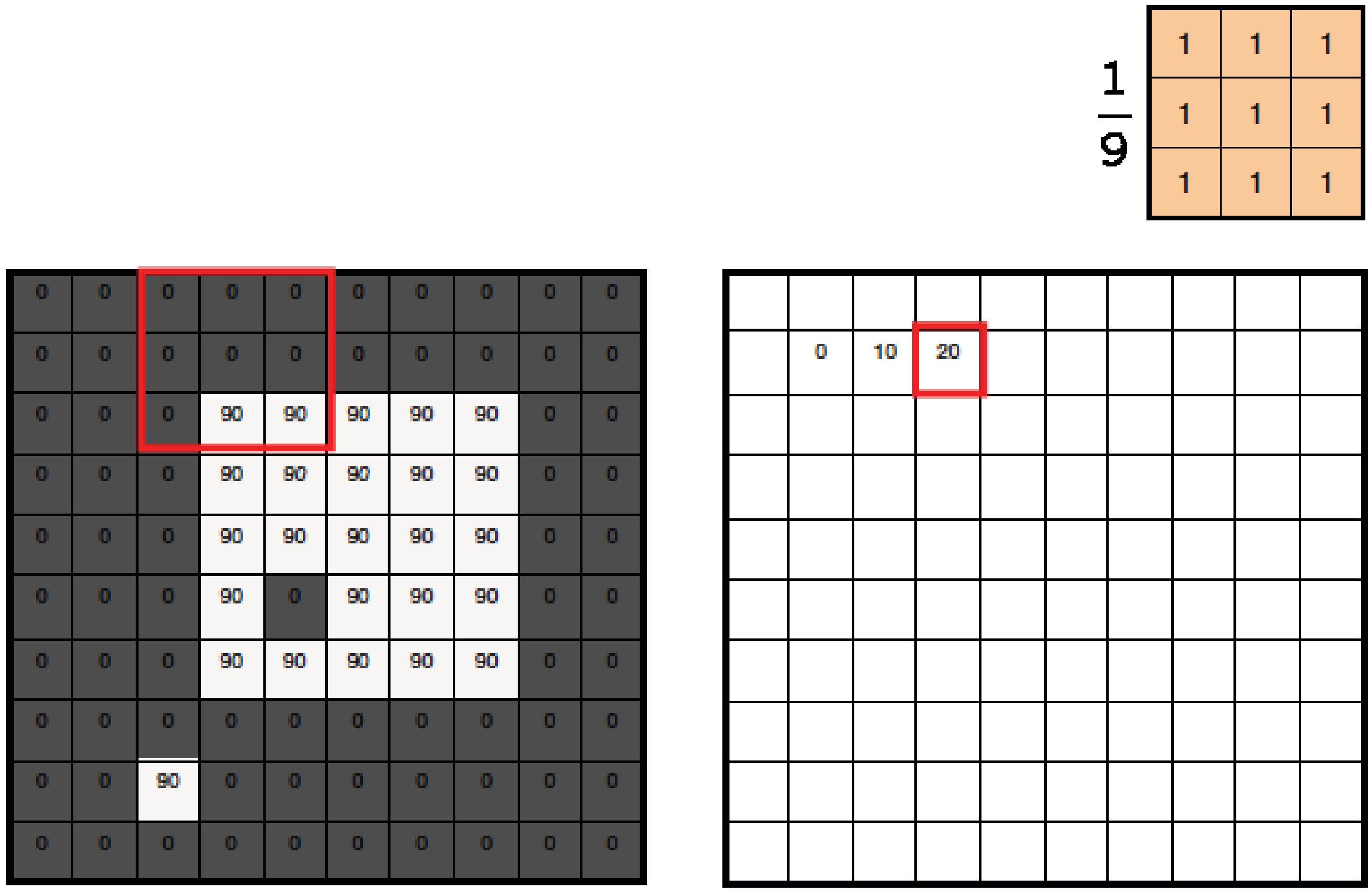
f[m,n]

How does convolution work?


1/9
1 1 1
1 1 1
1 1 1
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



1/9
1 1 1
1 1 1
1 1 1
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



1/9
1 1 1
1 1 1
1 1 1
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

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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					

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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				

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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				
				?					

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

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				
						?			
				50					

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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

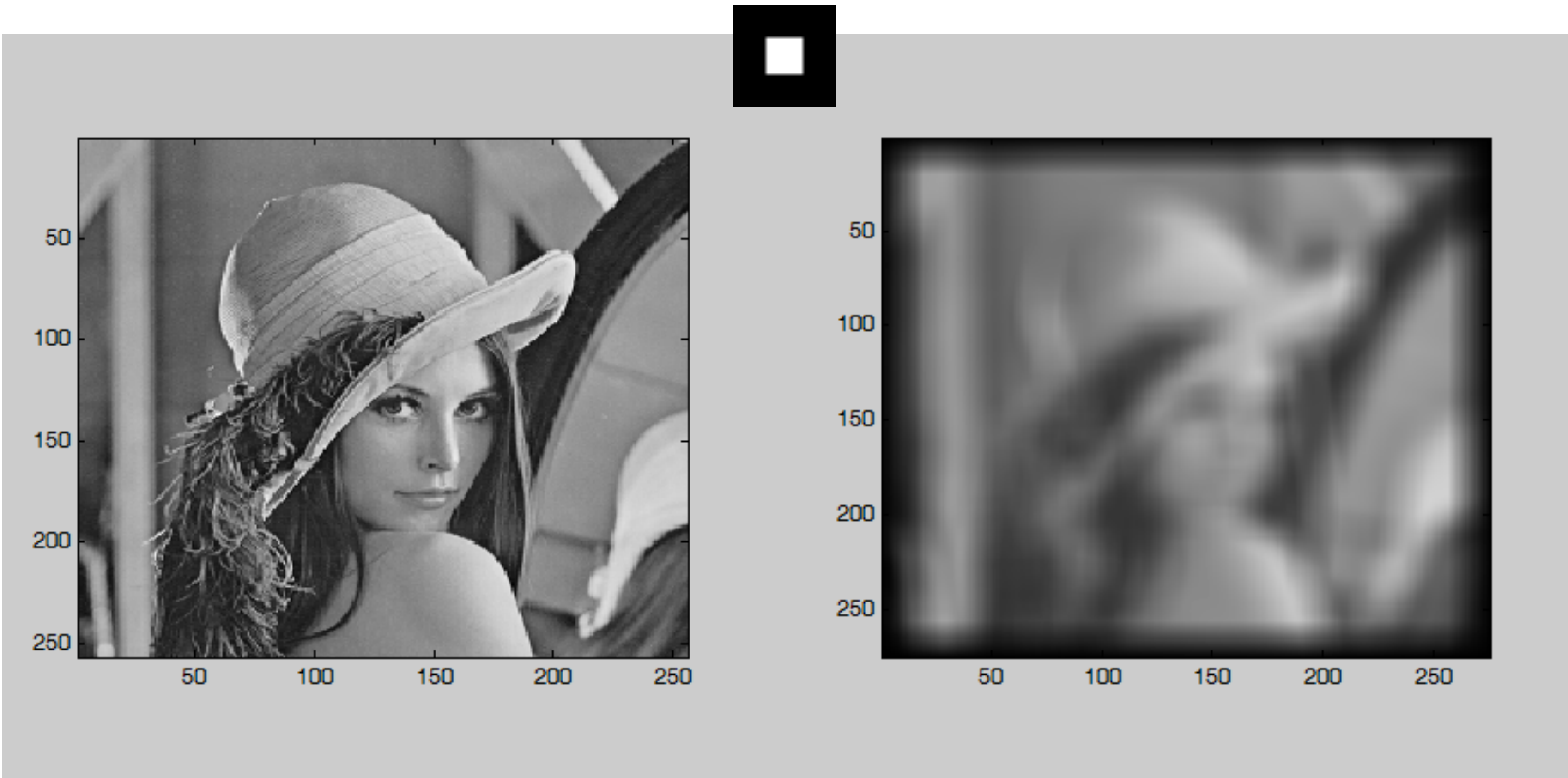
	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

Smoothing by Averaging

Convolution\ConvolutionAverage.m



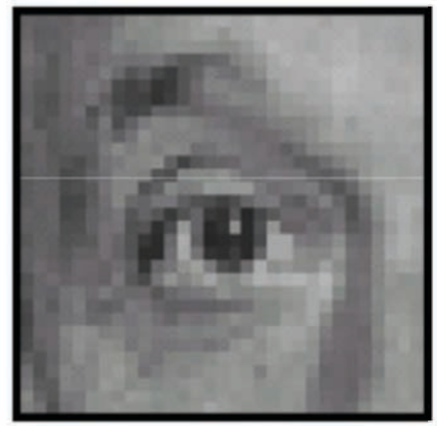
With a kernel of 20 x 20. This image is blur: you arrive at the blurry image by blurring some pixels together.

This image contains the “low spatial frequency” information

Impulse

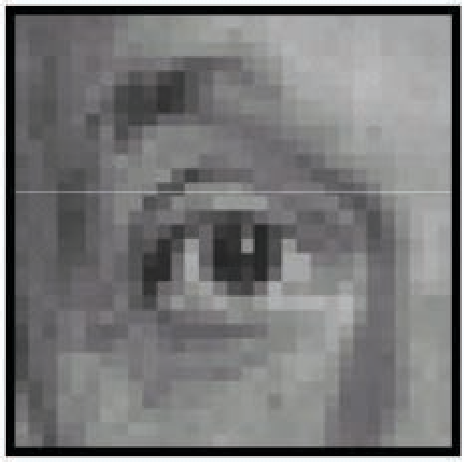
Original

0	0	0
0	1	0
0	0	0



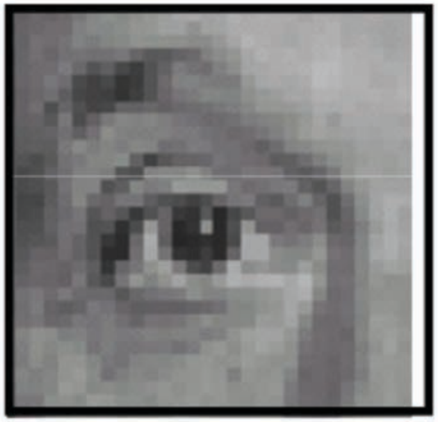
Filtered
(no change)

Shift



Original

0	0	0
0	0	1
0	0	0

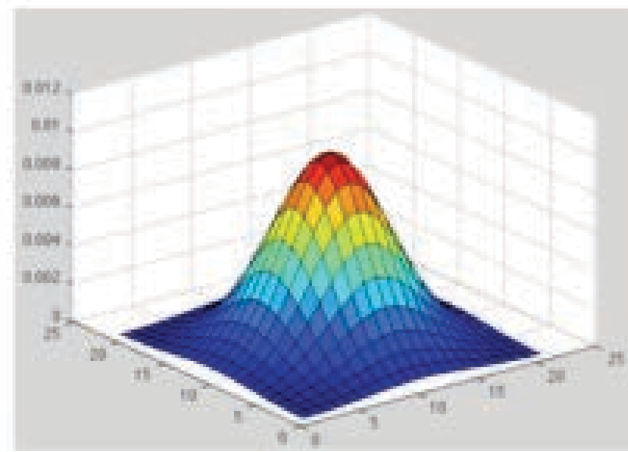


Shifted left
By 1 pixel

Smoothing with a Gaussian

- Smoothing with an average actually doesn't compare at all well with a defocused lens
- Most obvious difference is that a single point of light viewed in a defocused lens looks like a fuzzy blob; but the averaging process would give a little square.

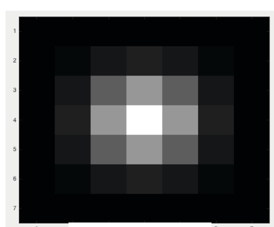
$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



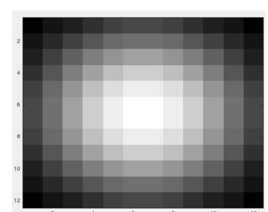
A Gaussian gives a good model of a fuzzy blob

Gaussian filter

Convolution\GaussianFiltering.m

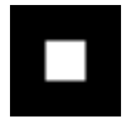


$\sigma=1$



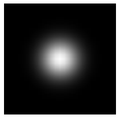
$\sigma=4$

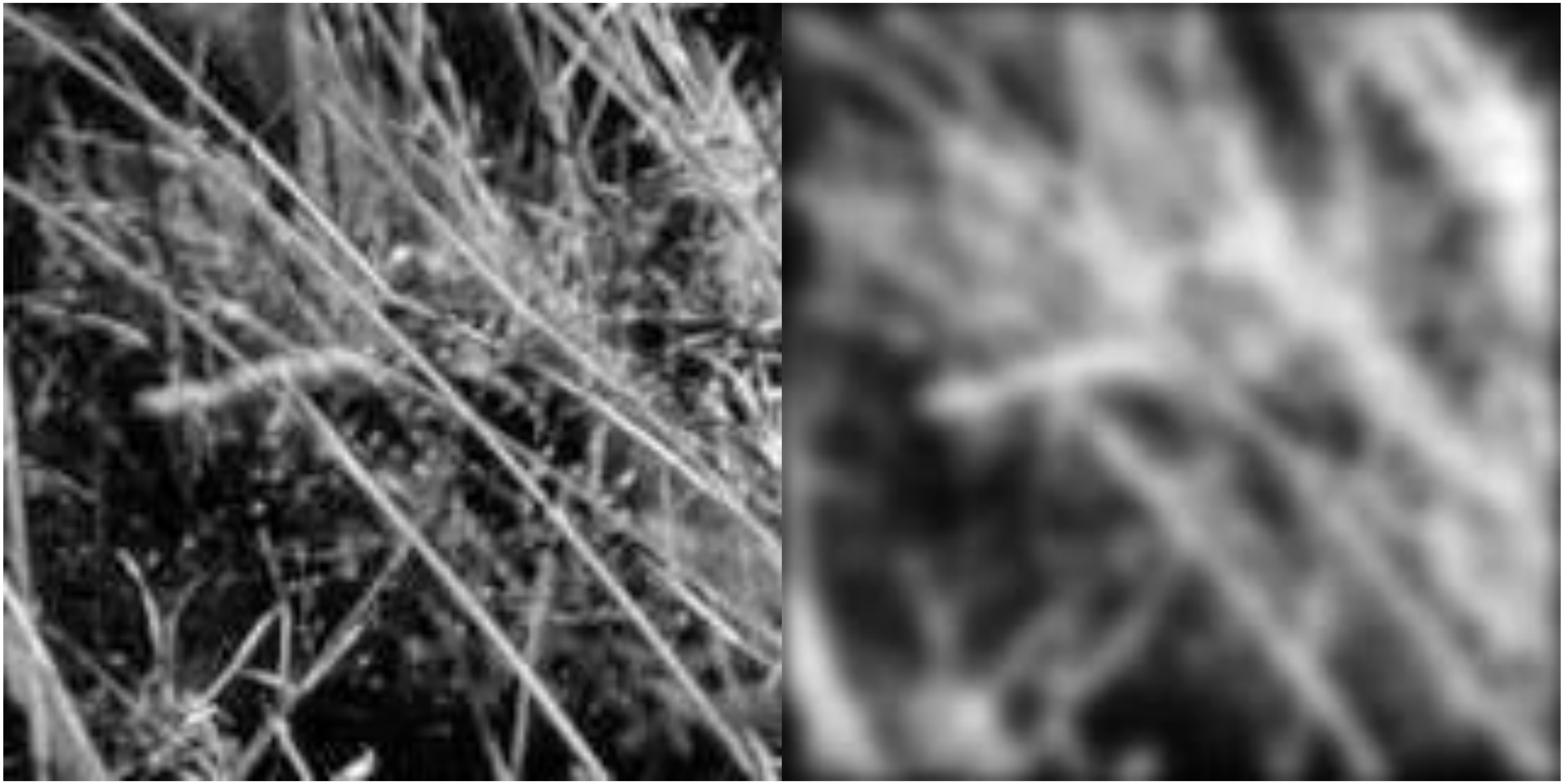
Smoothing by Averaging



Smoothing with a Gaussian

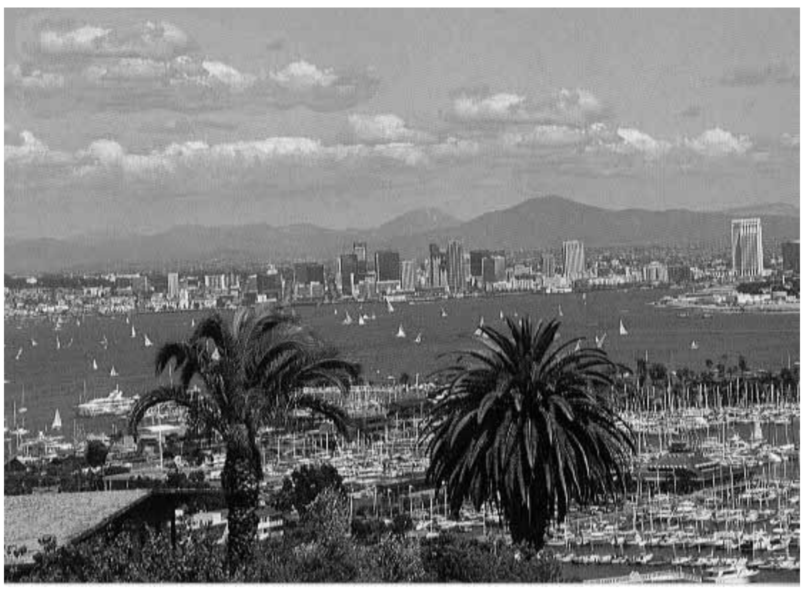
No more "ringing" effect





Human vision: fovea and periphery

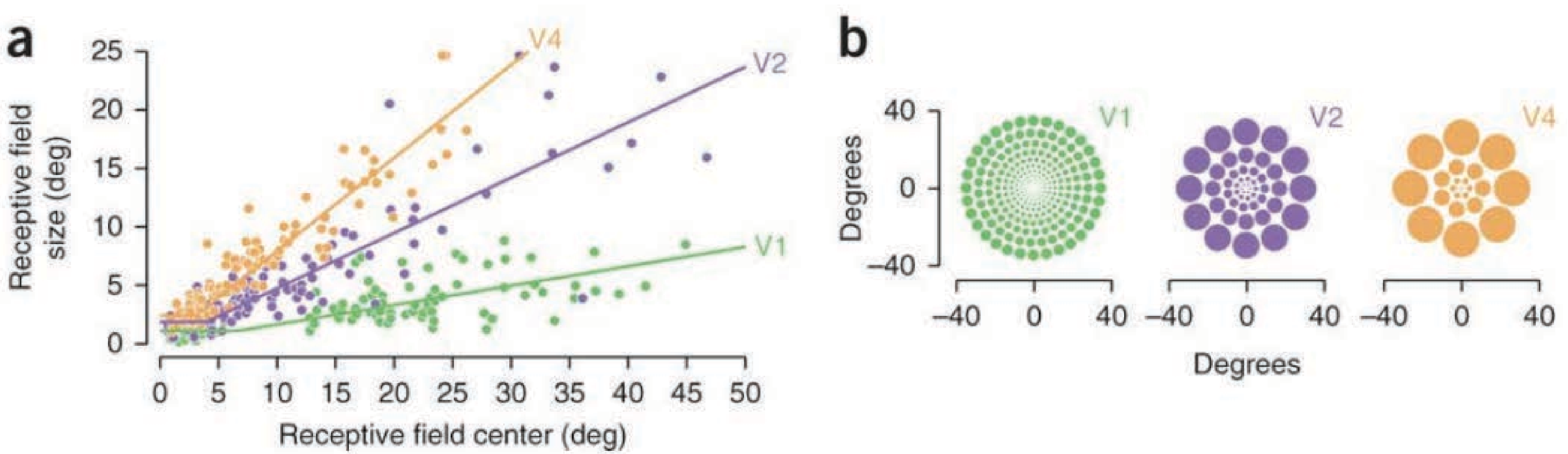
Some properties of image encoding like blurring, color representation set the stage for what is available to the neural system



Camera

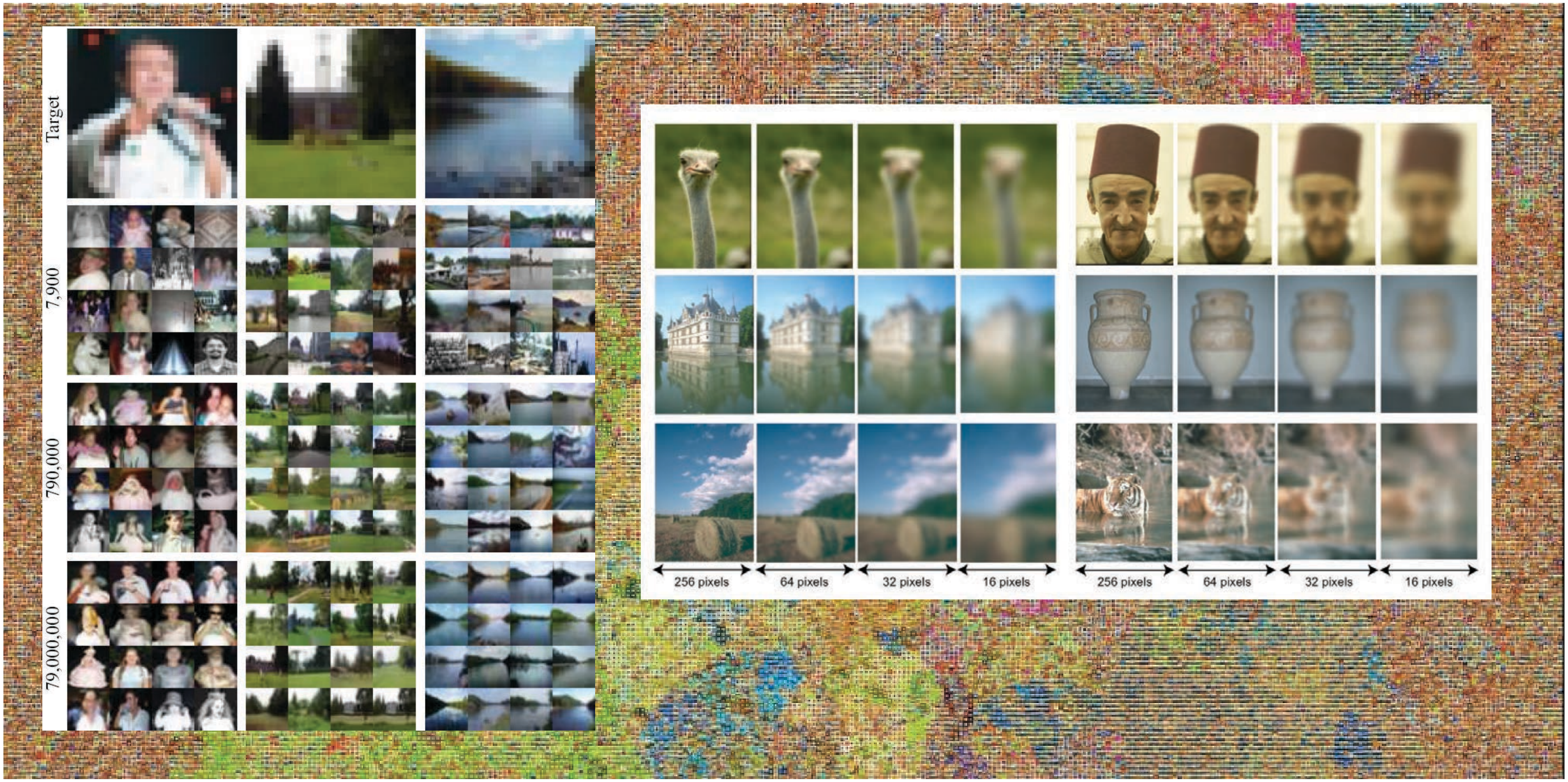
Human: Acuity decreases with eccentricity

Human vision: Receptive fields size scale with eccentricity



Freeman & Simoncelli (2011)

80 millions tiny images



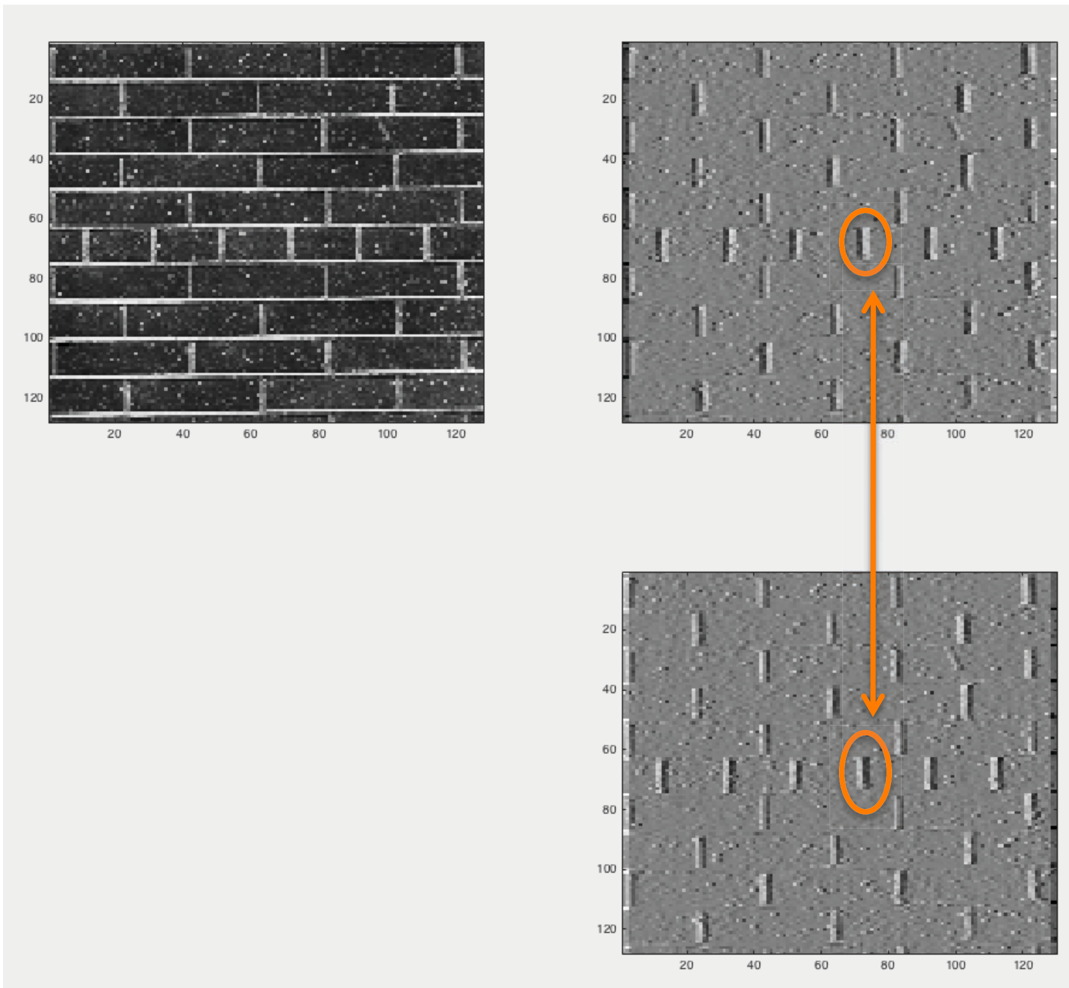
A. Torralba, R. Fergus, W.T. Freeman. PAMI 2008

Torralba (2009). How many pixels make an image?

Derivatives (contours)

Convolution\Highpassfilter.m
(see exercise for different orientations)

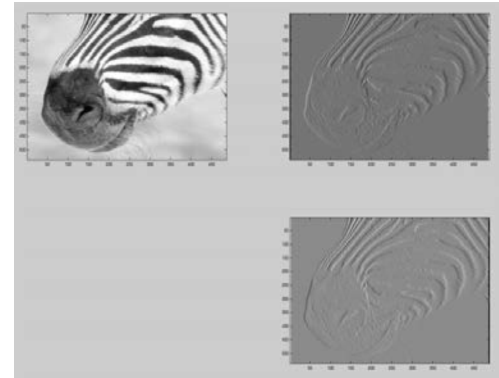
The result is “signed”

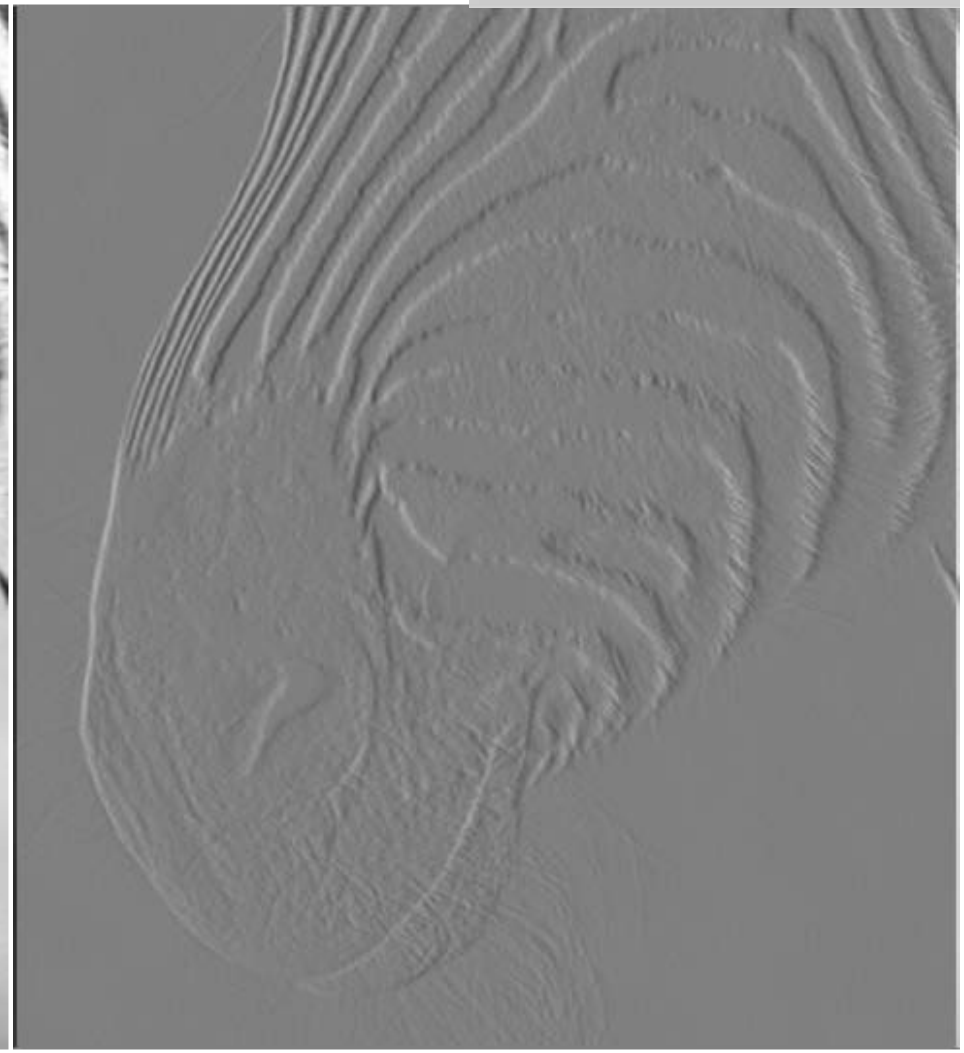


hk=[-1 0 1];

hk=[1 0 -1];

darker is negative,
lighter is positive,
mid grey is zero.



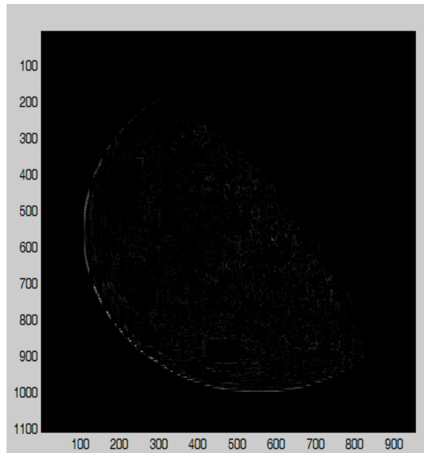


Laplacian filter

Convolution\Highpassfilter.m

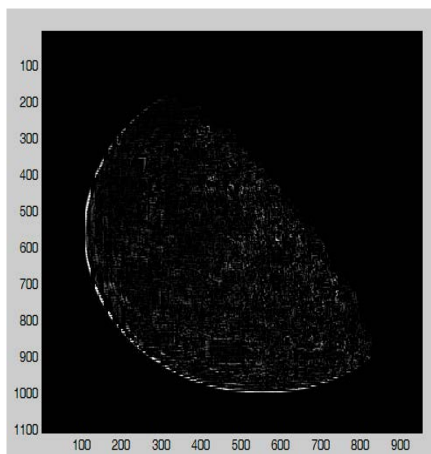
- kernel 1 =

0	1	0
1	-4	1
0	1	0



- Kernel 2 =

1	1	1
1	-8	1
1	1	1



What is the difference between the two kernels ?

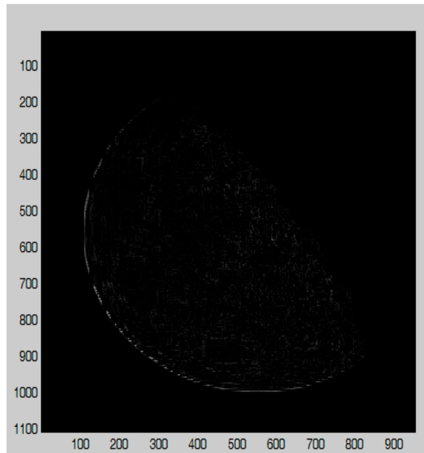
The Laplacian operator is implemented as a convolution between an image and a kernel (shown here)

Laplacian filter

Convolution\Highpassfilter.m

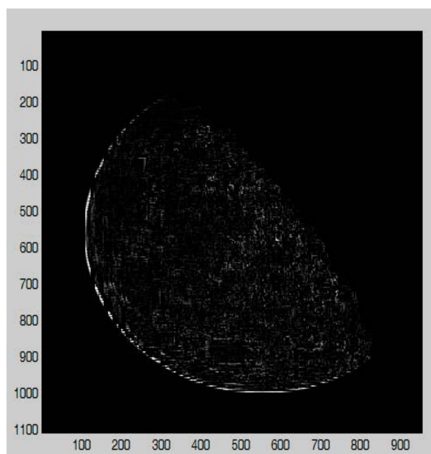
- kernel 1 =

0	1	0
1	-4	1
0	1	0



- Kernel 2 =

1	1	1
1	-8	1
1	1	1



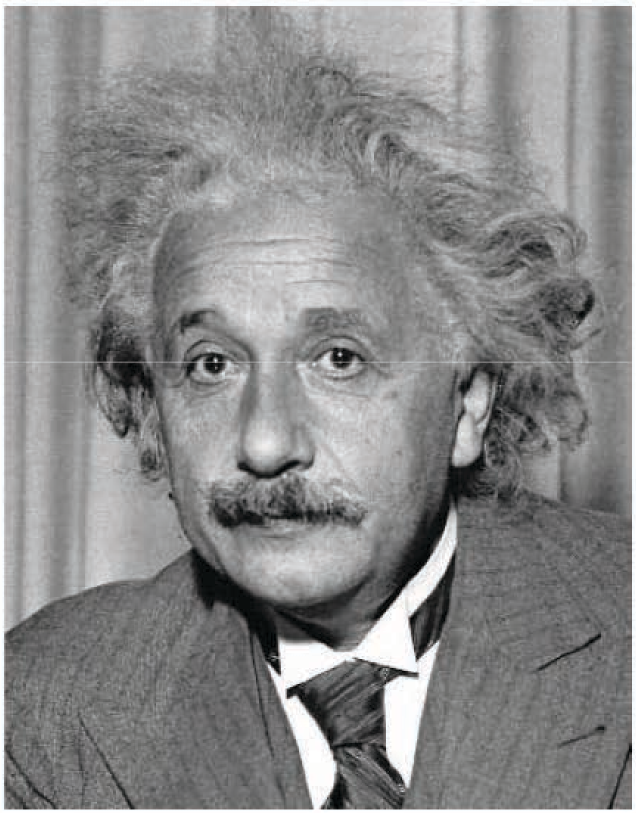
-1	-1	-1
-1	8	-1
-1	-1	-1

In image convolution, the kernel is centered on each pixel in turn, and the pixel value is replaced by the sum of the kernel multiplied by the image values. In this particular kernel we are using here, **we are counting the contributions of the diagonal pixels as well as the orthogonal pixels in the filter operation.**

What can the laplacien filter be used for ? Image sharpening

Image Sharpening with a Laplacian kernel

Sobel

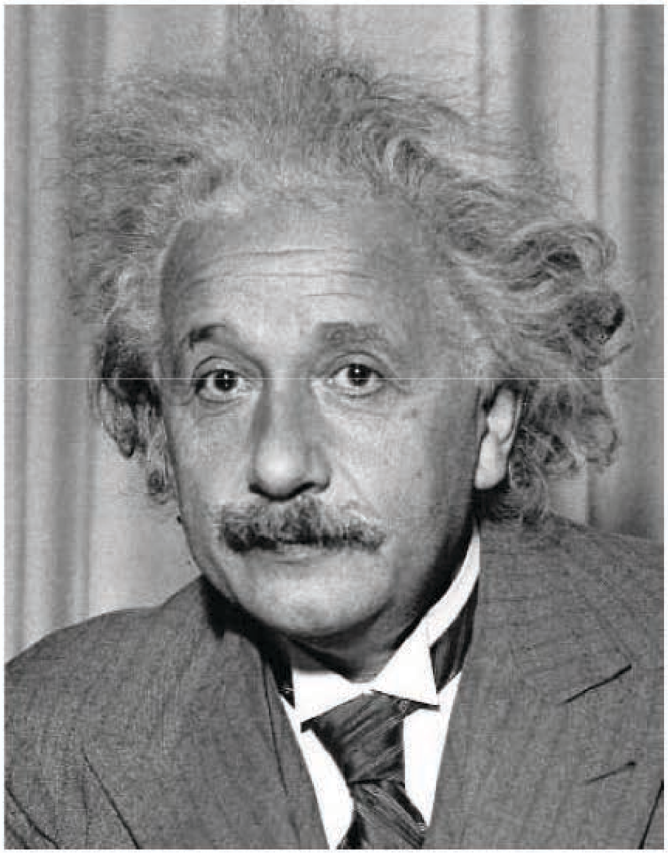


1	0	-1
2	0	-2
1	0	-1

?

Convolution\Highpassfilter.m

Sobel



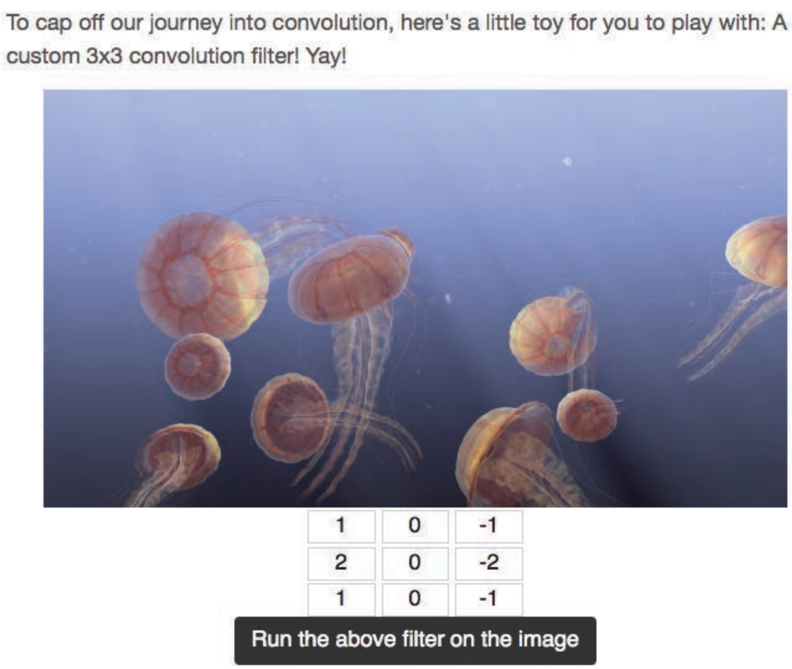
1	2	1
0	0	0
-1	-2	-1

?

Convolution\Highpassfilter.m

Filtering on the web

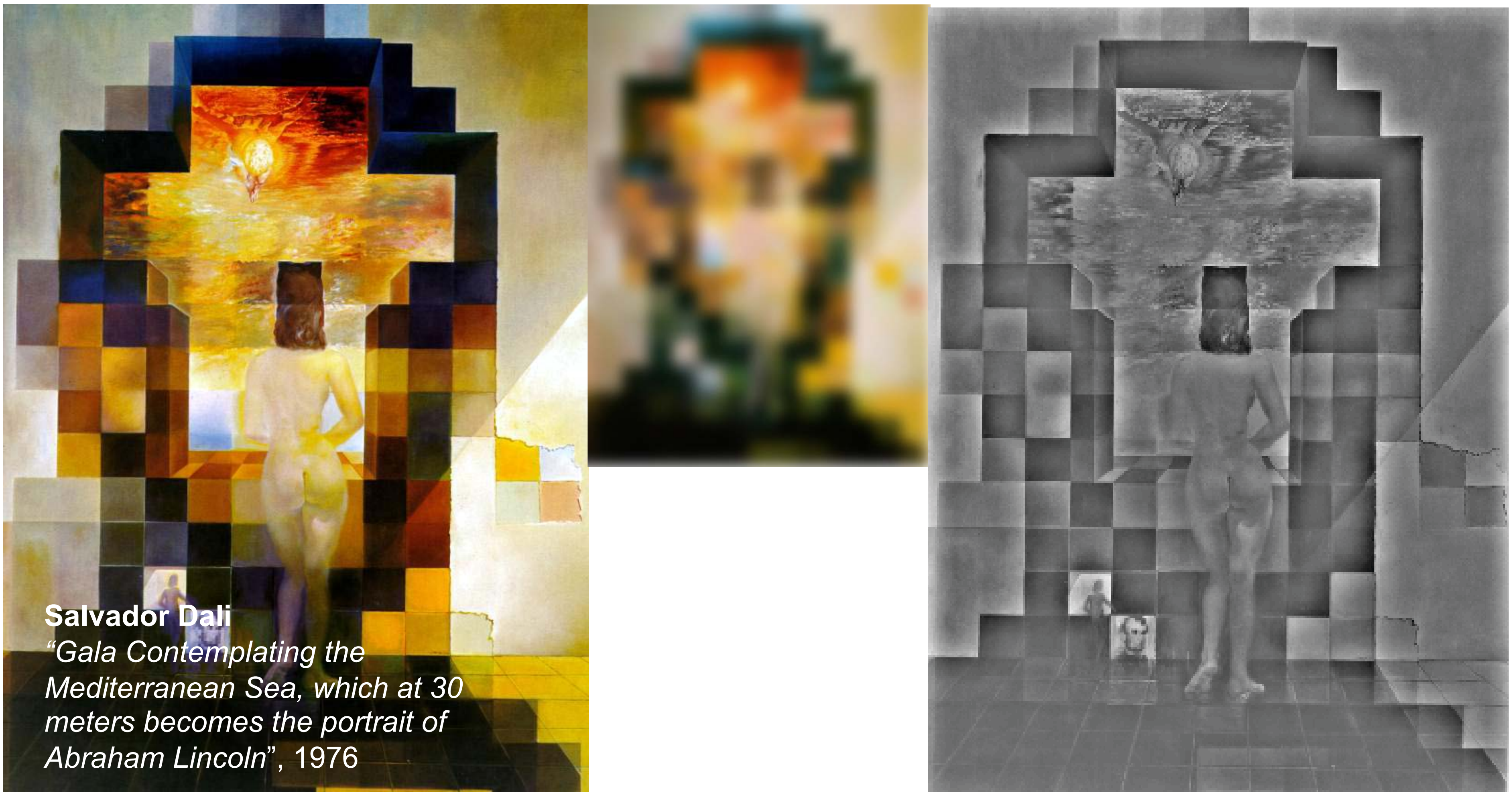
- http://www.html5rocks.com/en/tutorials/canvas/imagefilters/
- http://setosa.io/ev/image-kernels/



Thanks to Lea Verou and Jon Gjengset

III. Fourier Transform

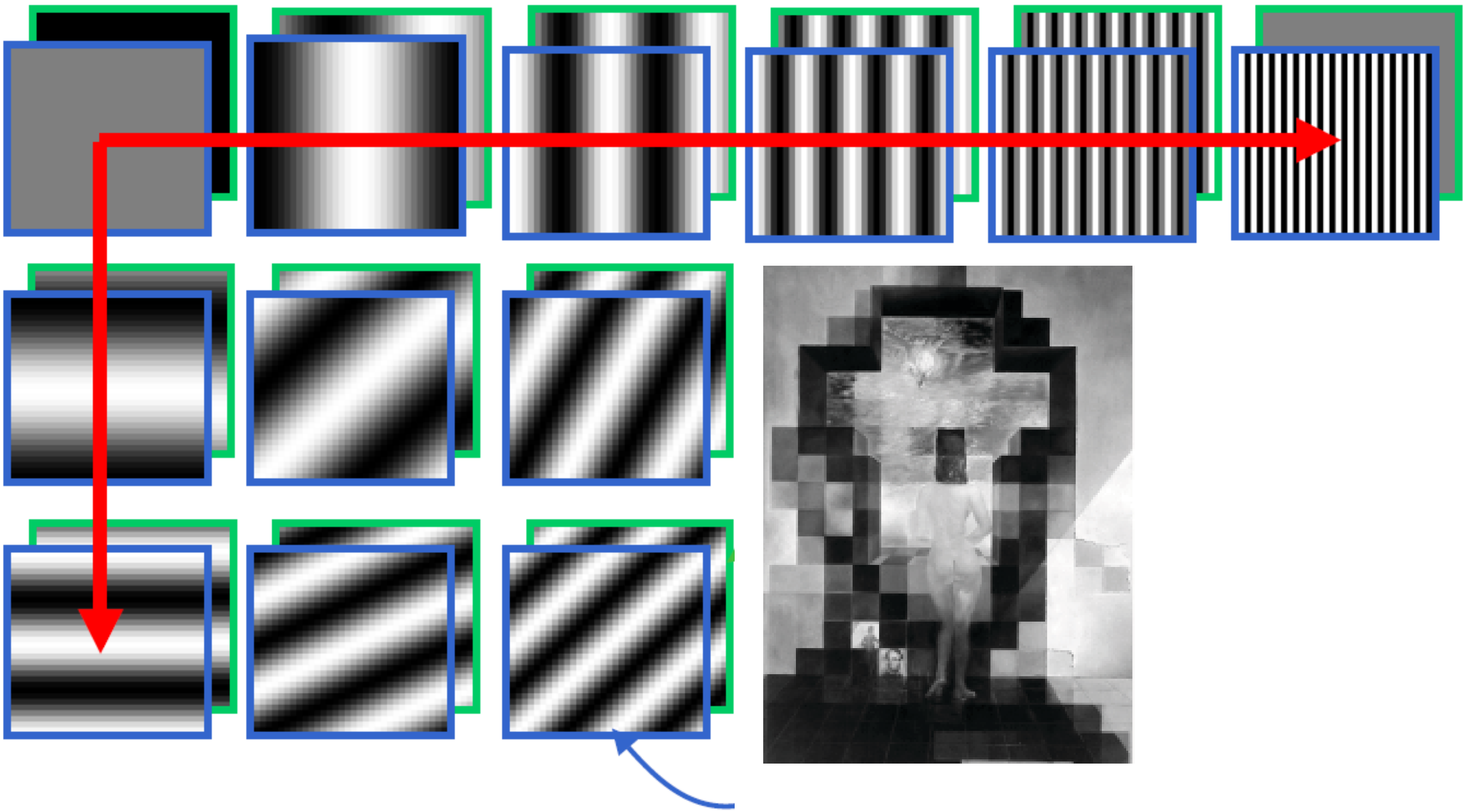
Fourier analysis is a method by which any two dimensional luminance image can be analyzed into the sum of a set of sinusoidal gratings that differ in **spatial frequency**, **orientation**, **amplitude** and **phase**.



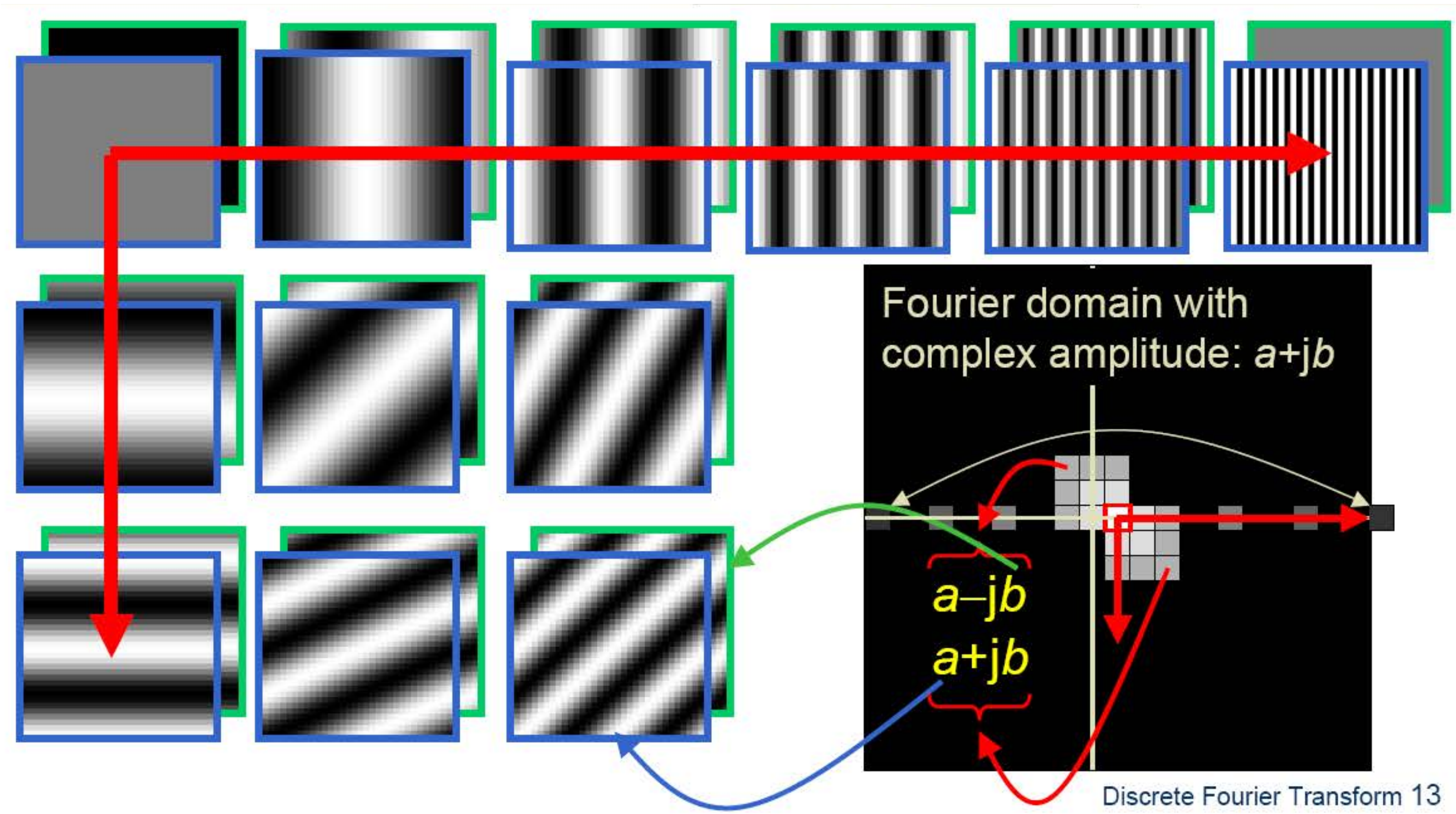
Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976

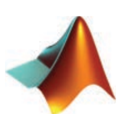
Sinusoidal gratings as the "primitives" of an image

A nice set of basis: Teases away fast vs. slow changes in the image.



Sinusoidal gratings as the “primitives” of an image

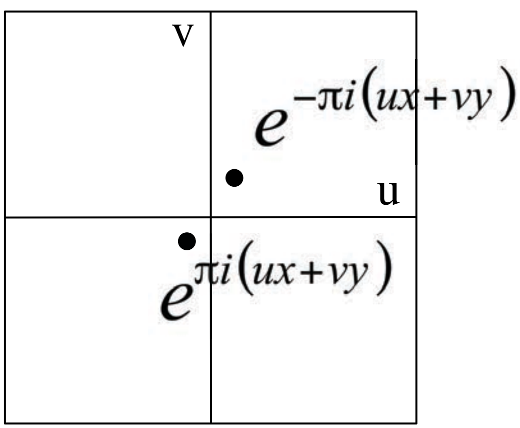


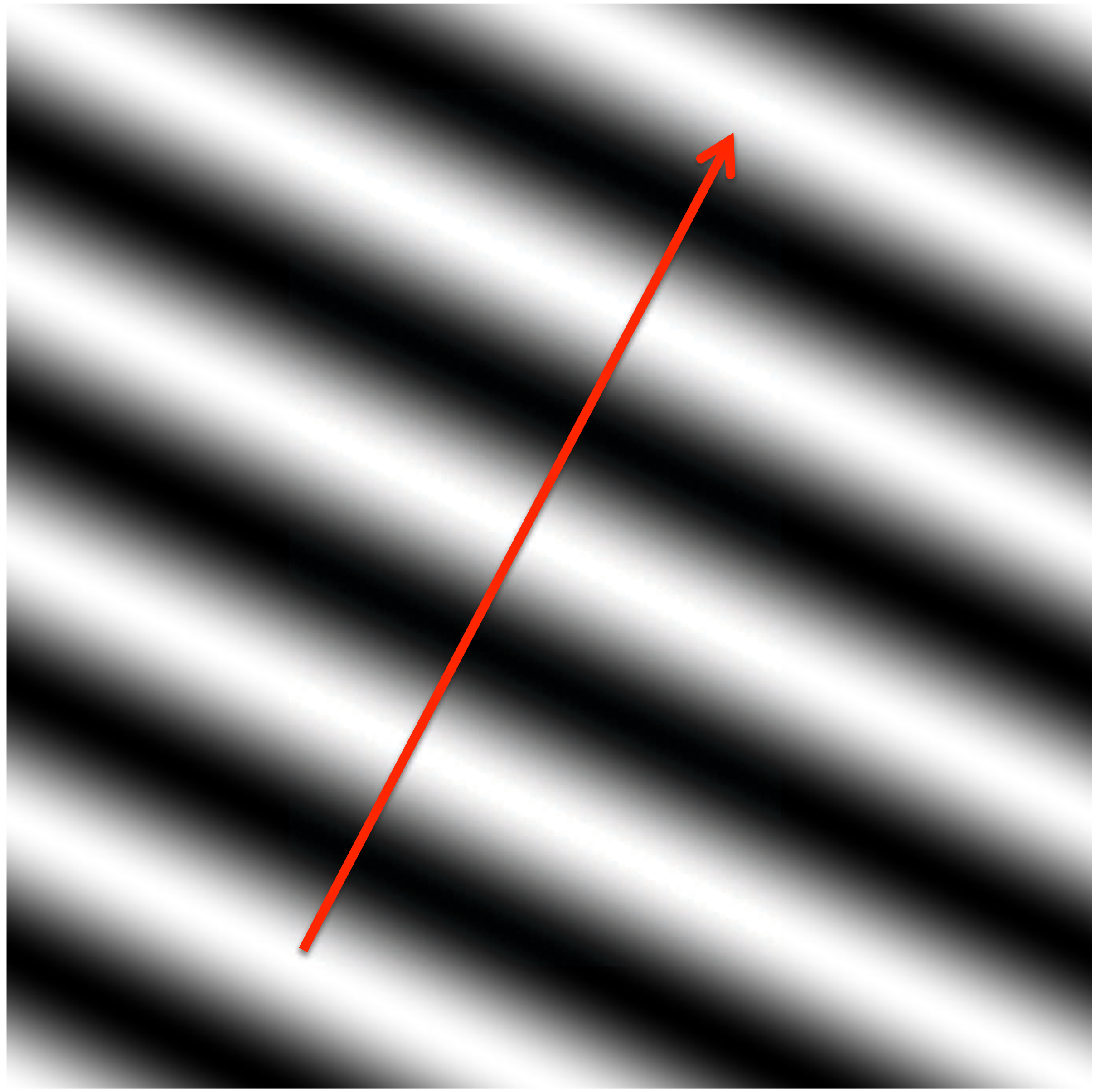


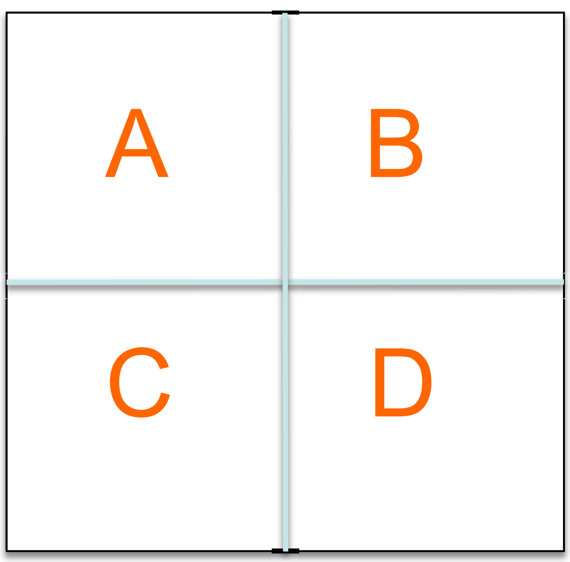
```
imagesc(log(abs(fftshift(fft2(im)))));
```

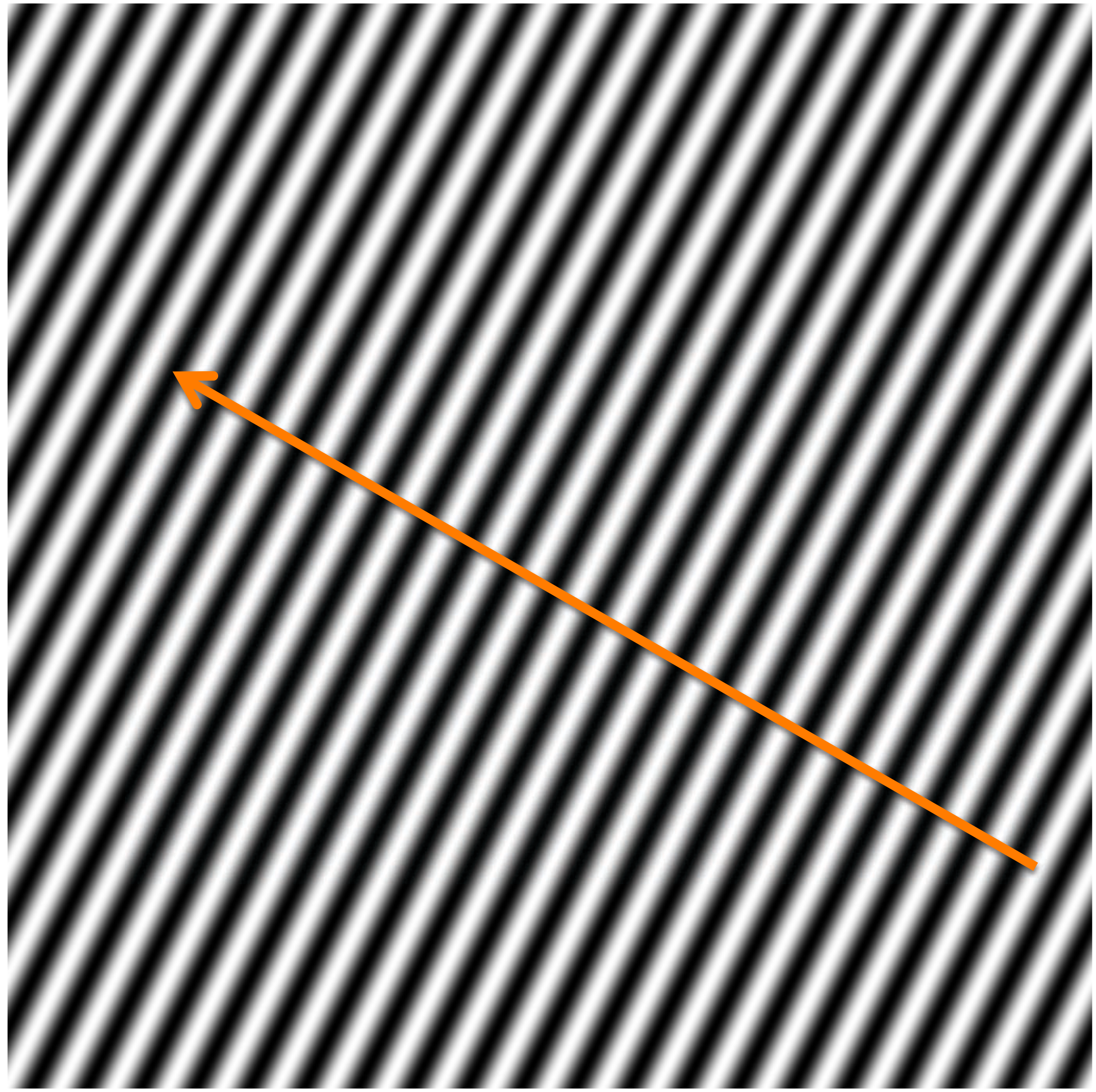
Slide from A. Efros

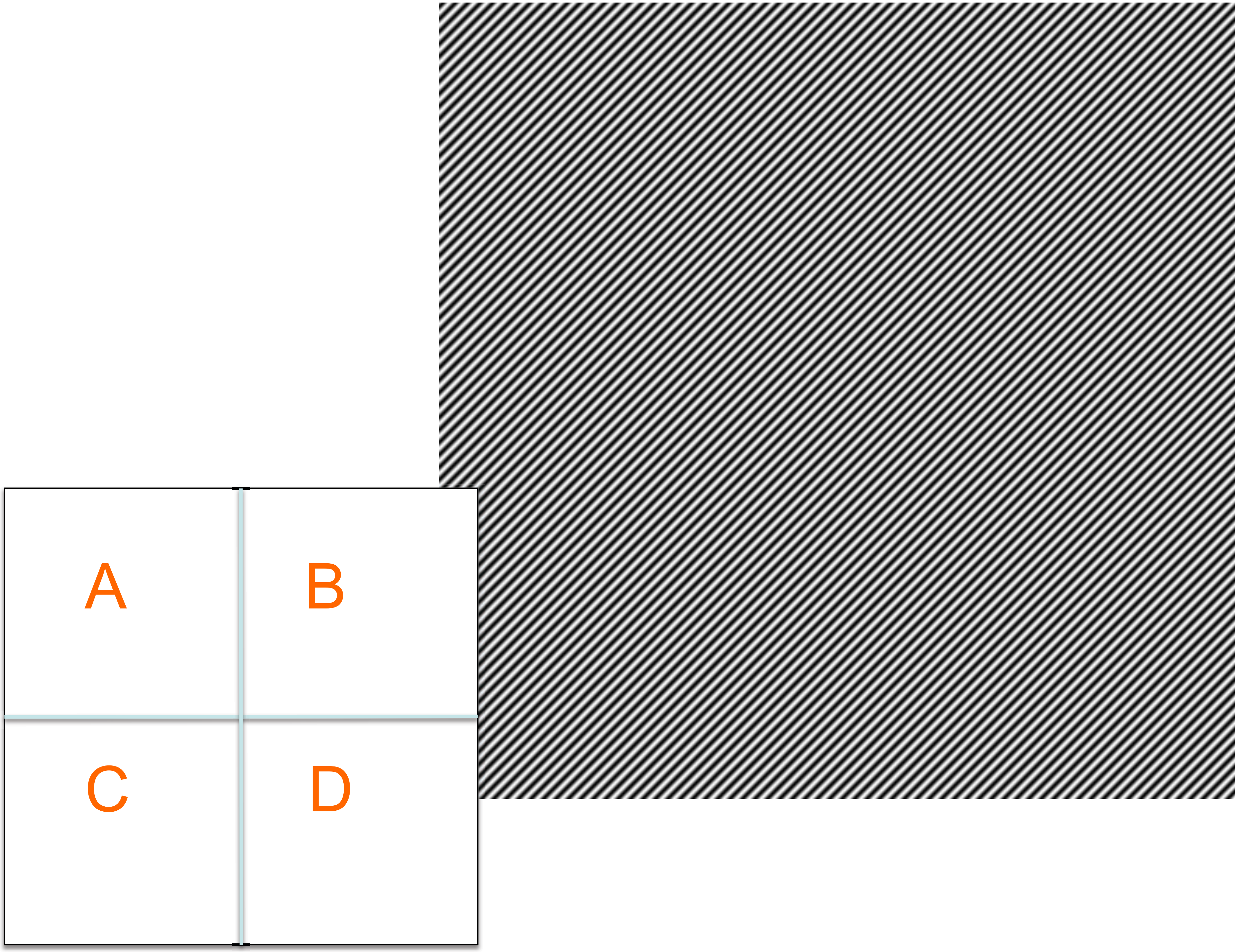
A basic element: a sinusoid with a frequency along a direction, with alternating dark and light in a certain direction.



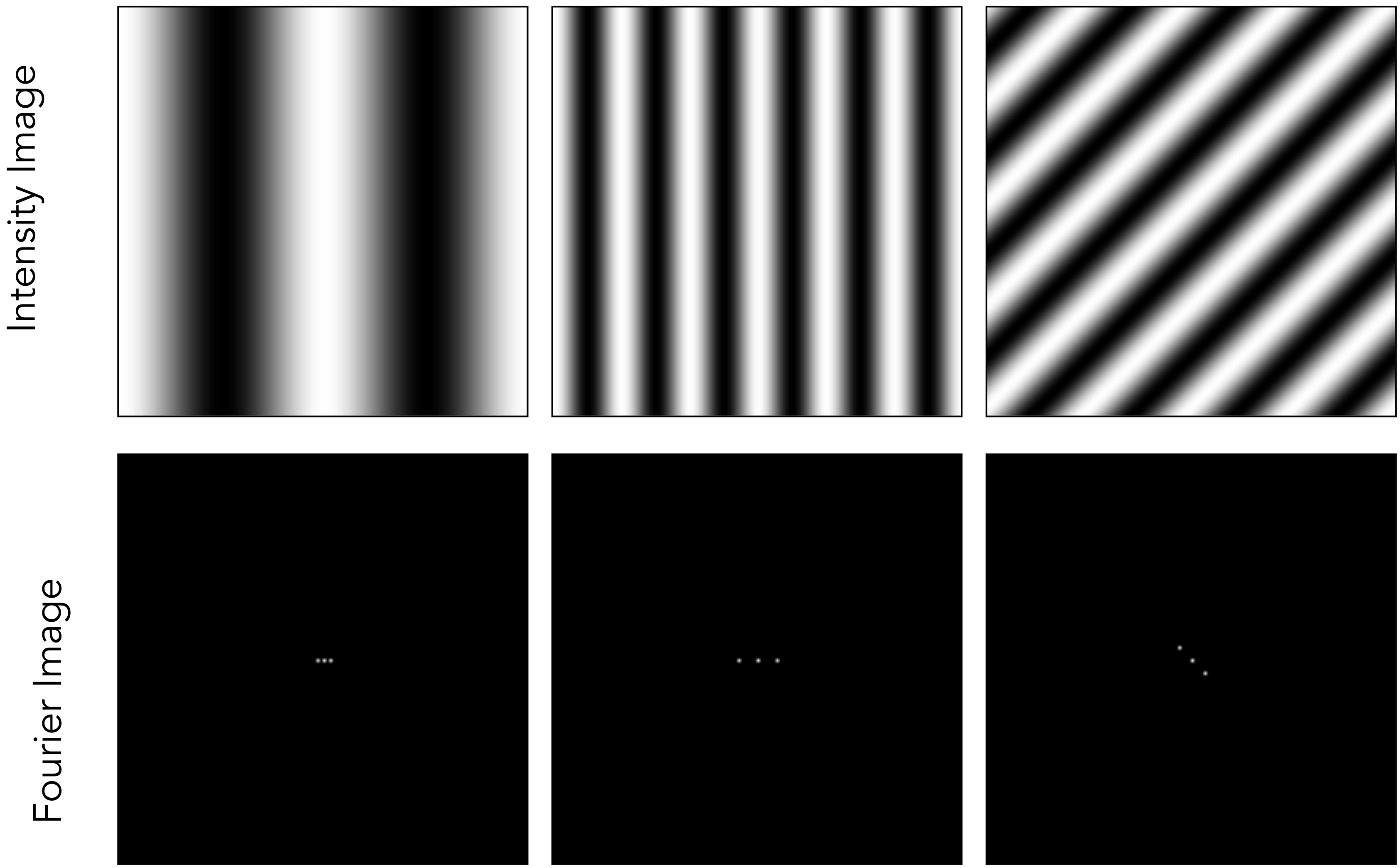



A
B
C
D




A
B
C
D

Fourier analysis in images

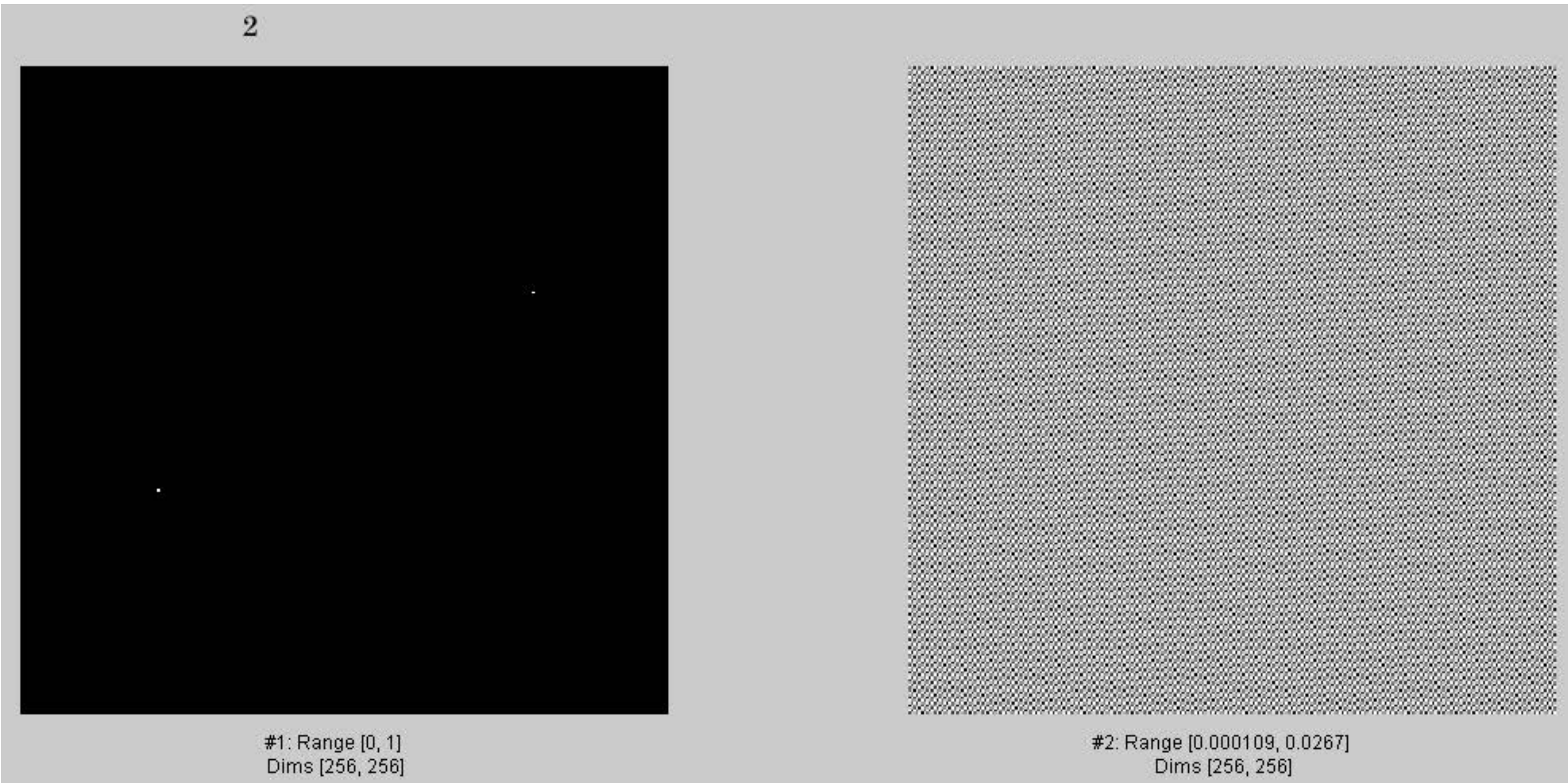


Two examples of image synthesis with Fourier basis

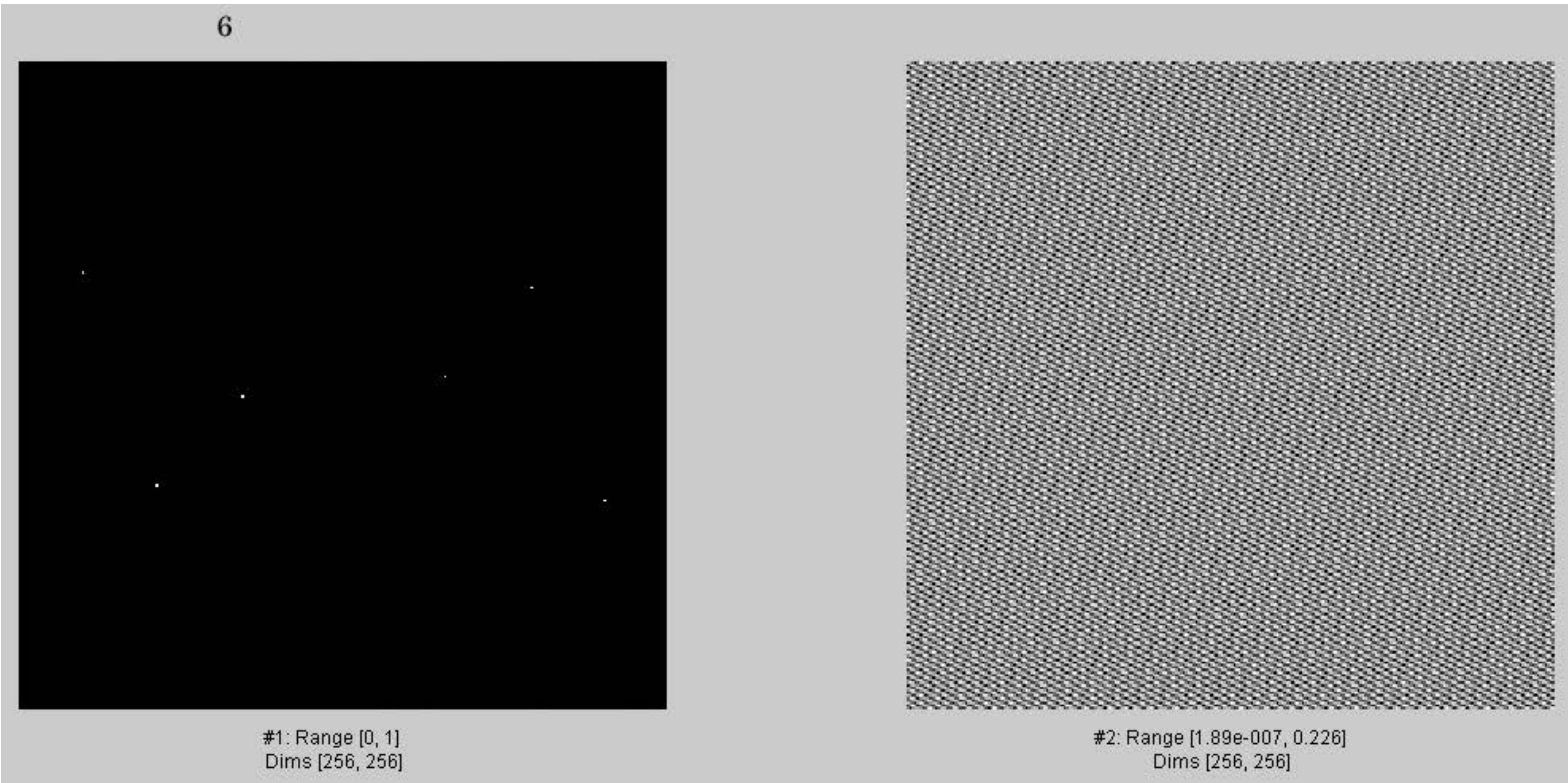
First: randomly sample the Fourier coefficients of an image and reconstruct from those.

Second: sample Fourier coefficients in descending order of amplitude.

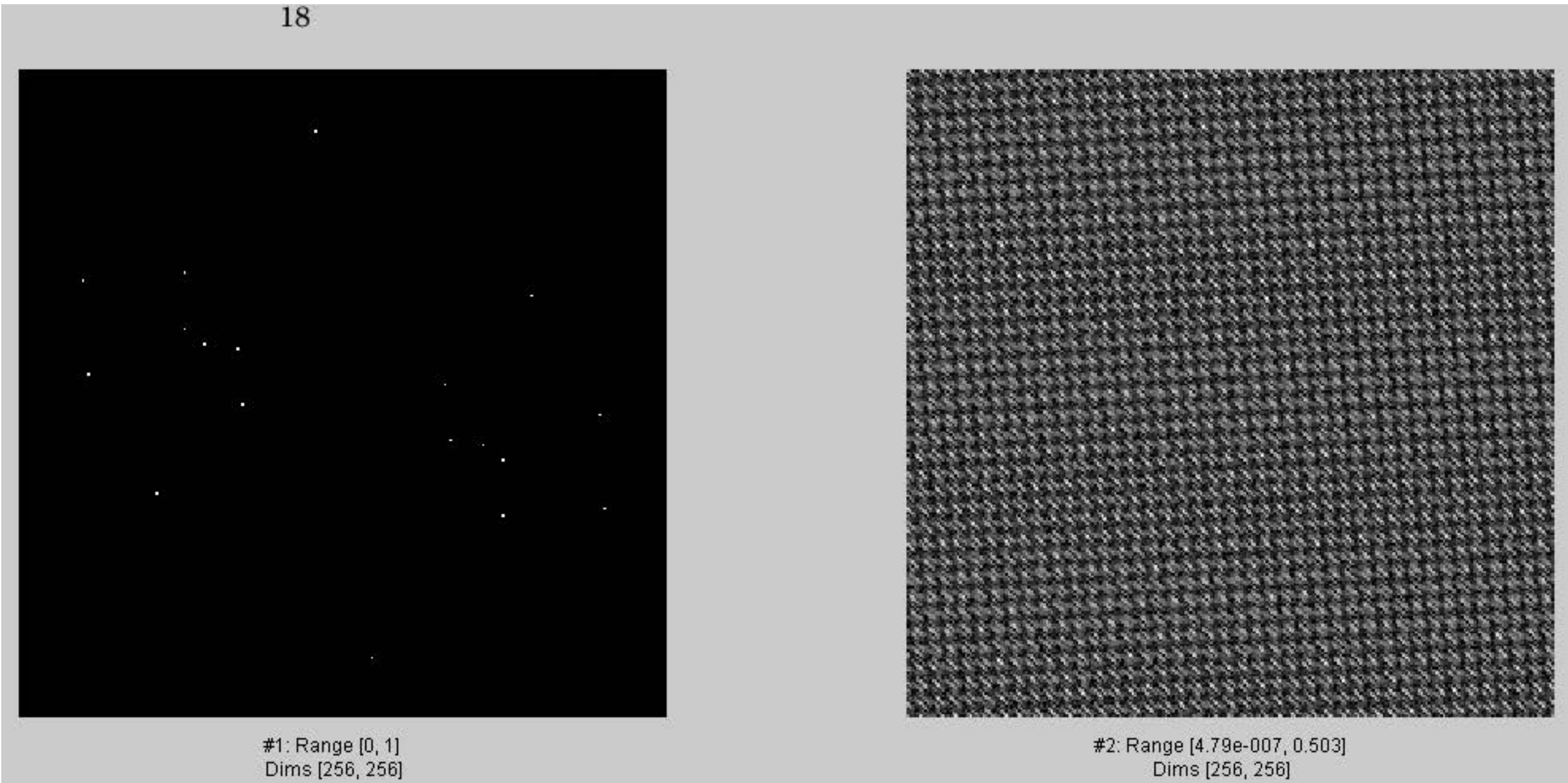
2



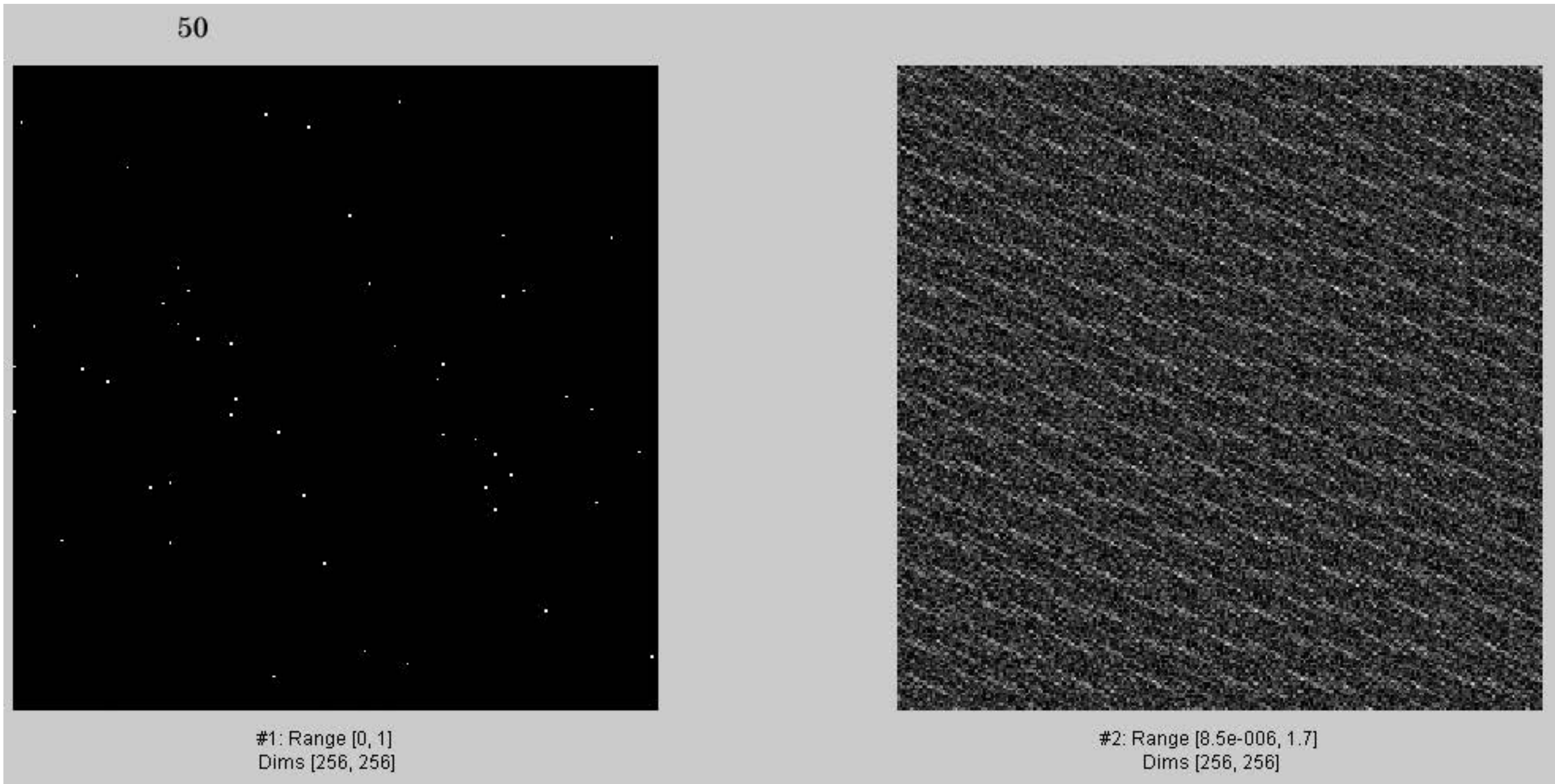
6



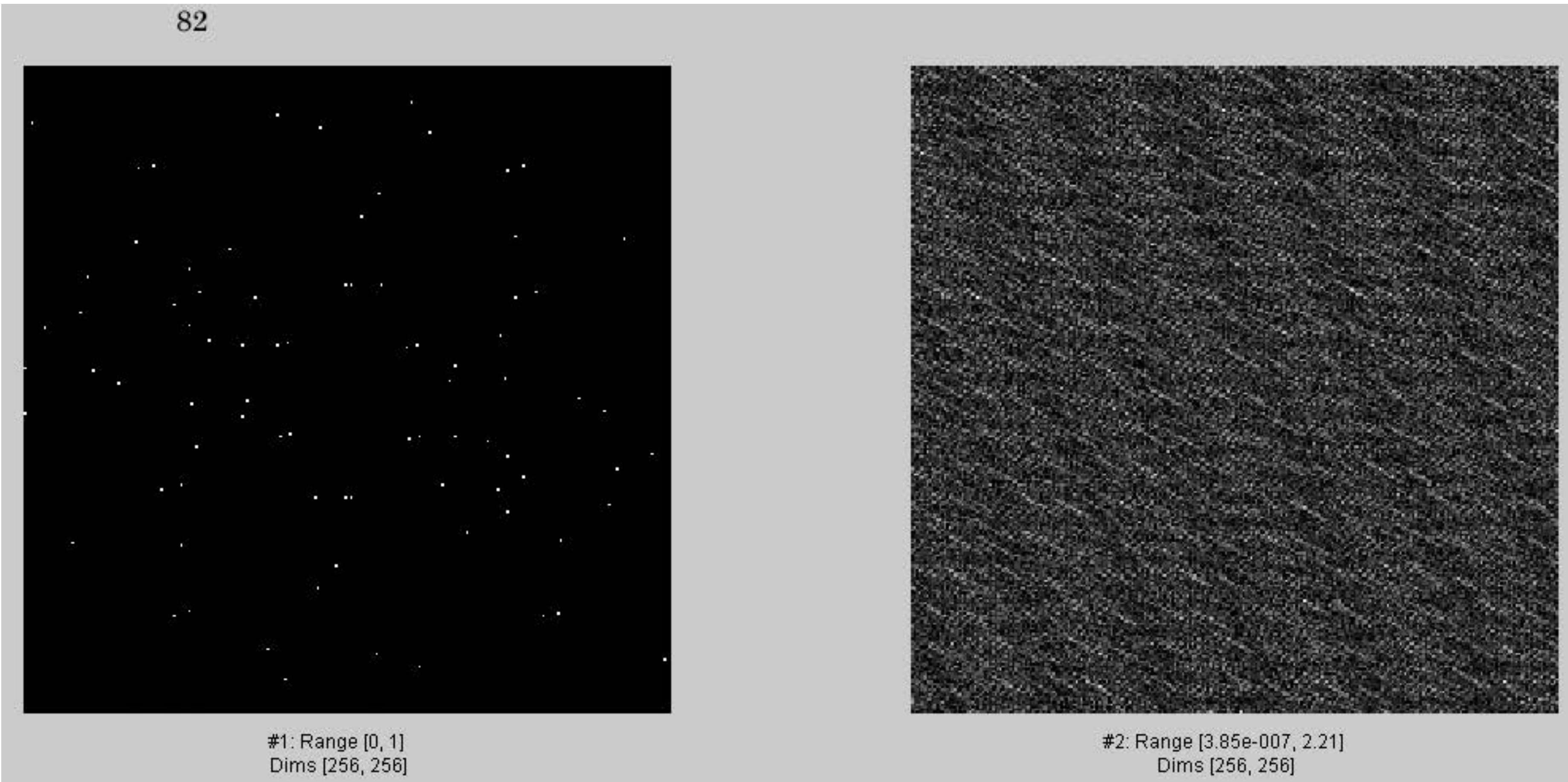
18



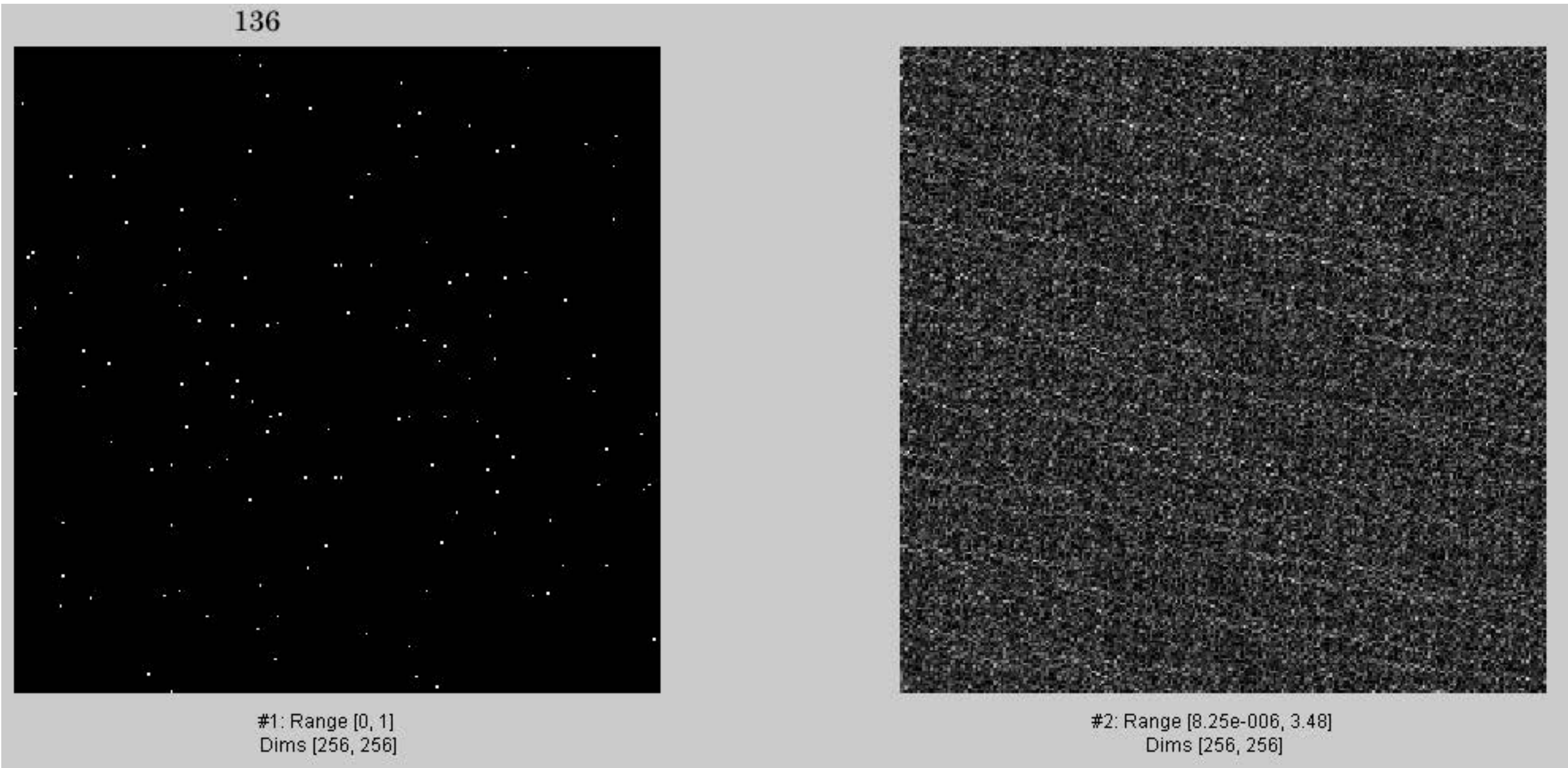
50



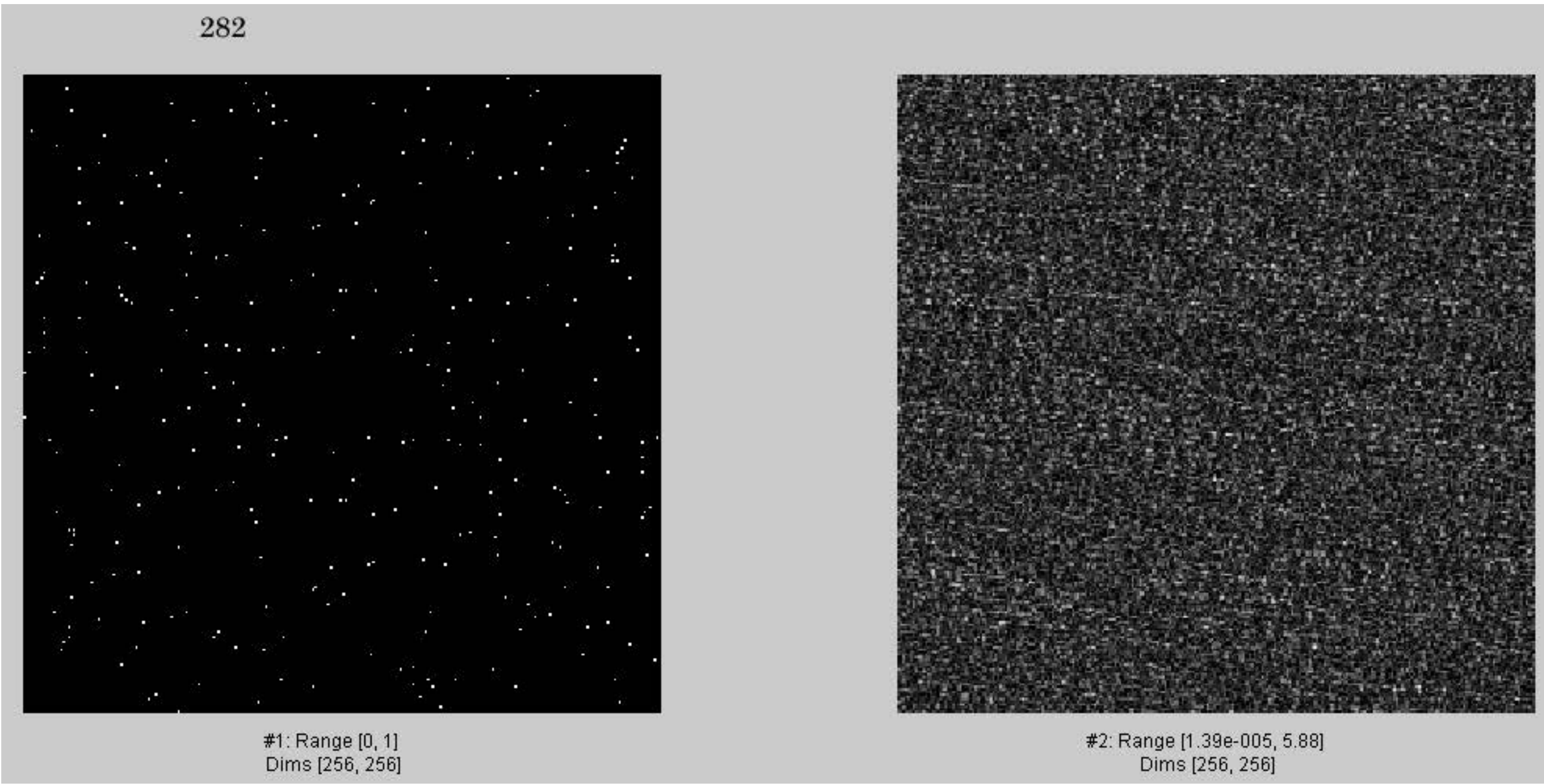
82



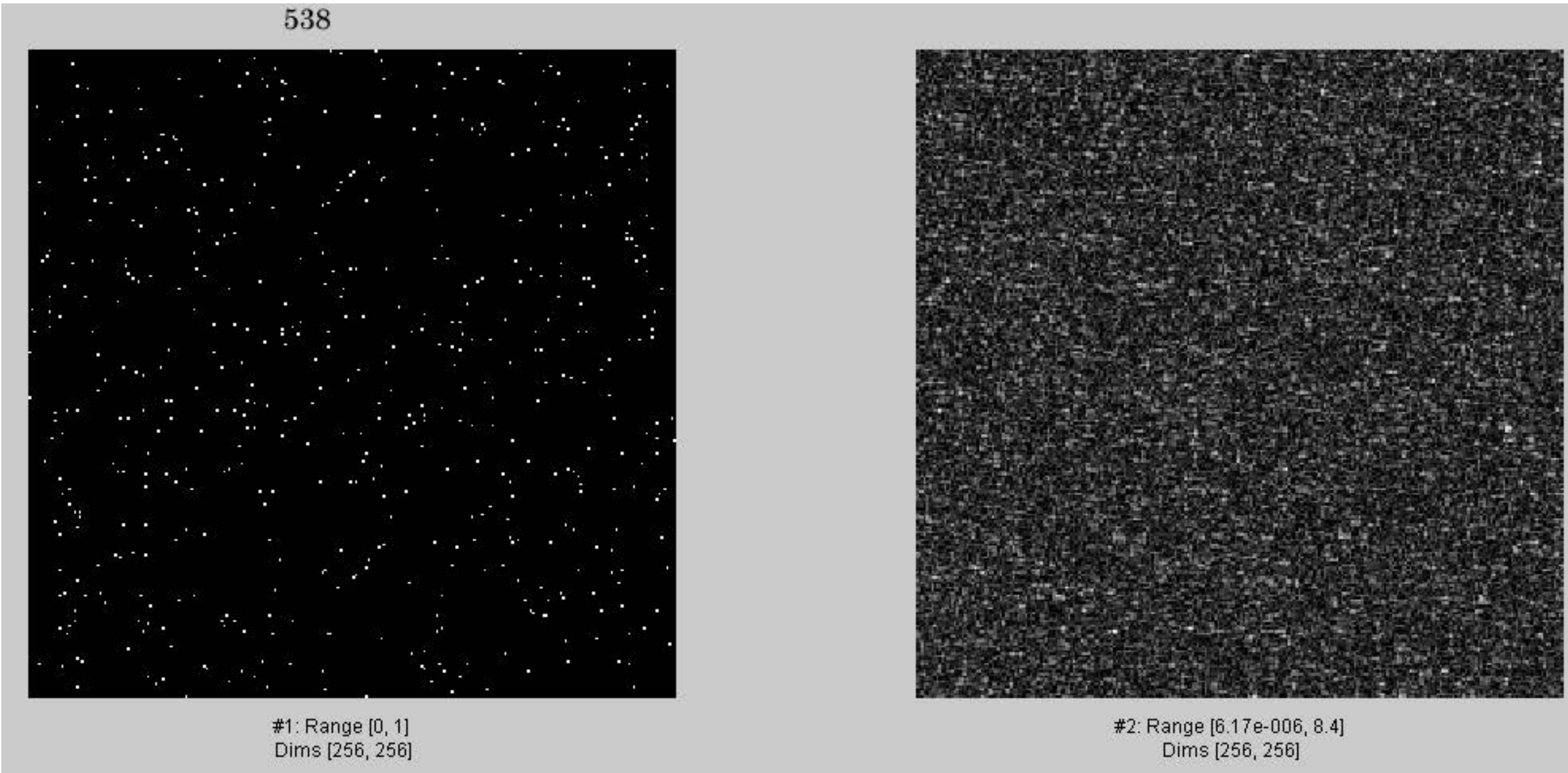
136



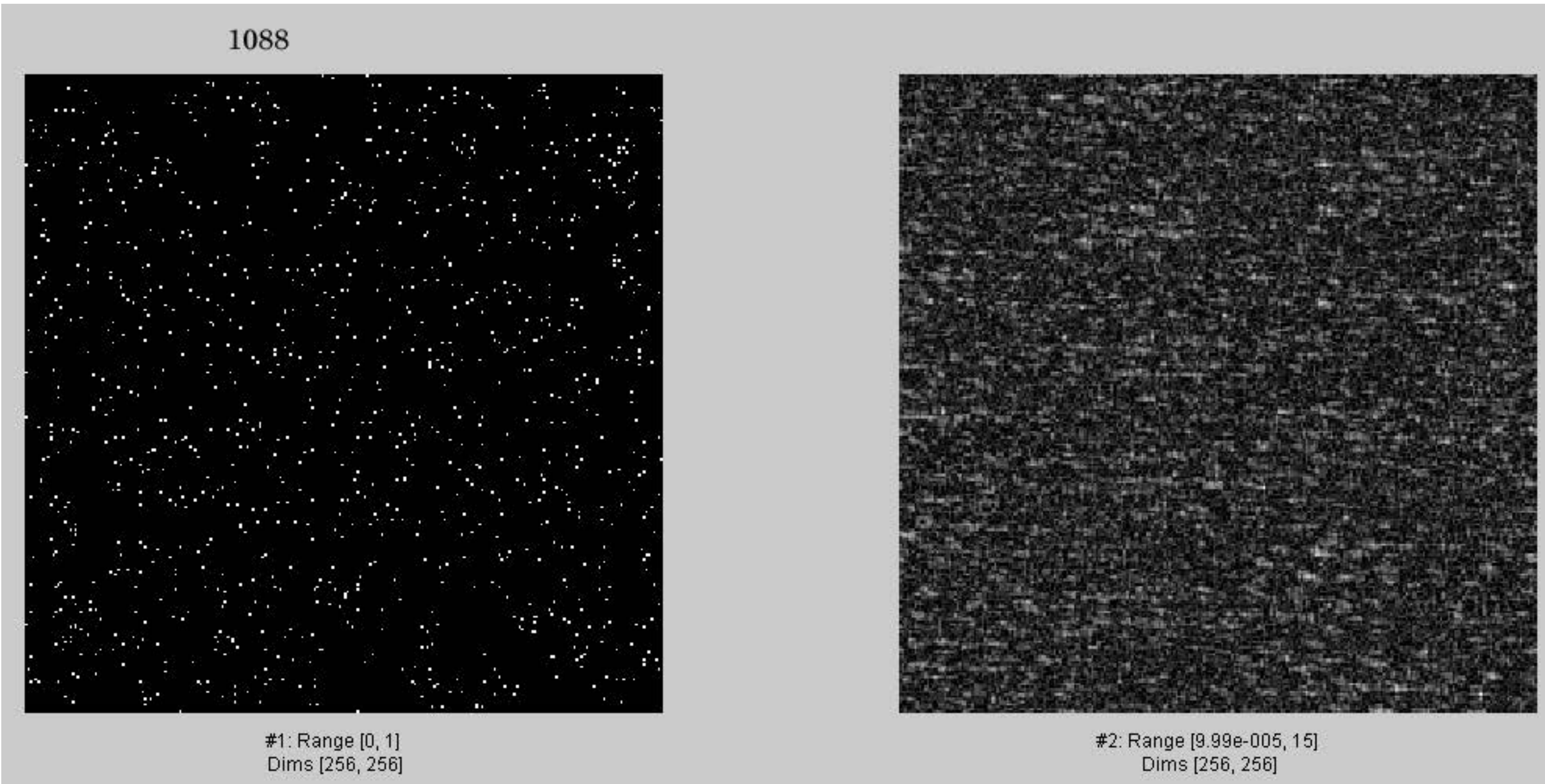
282



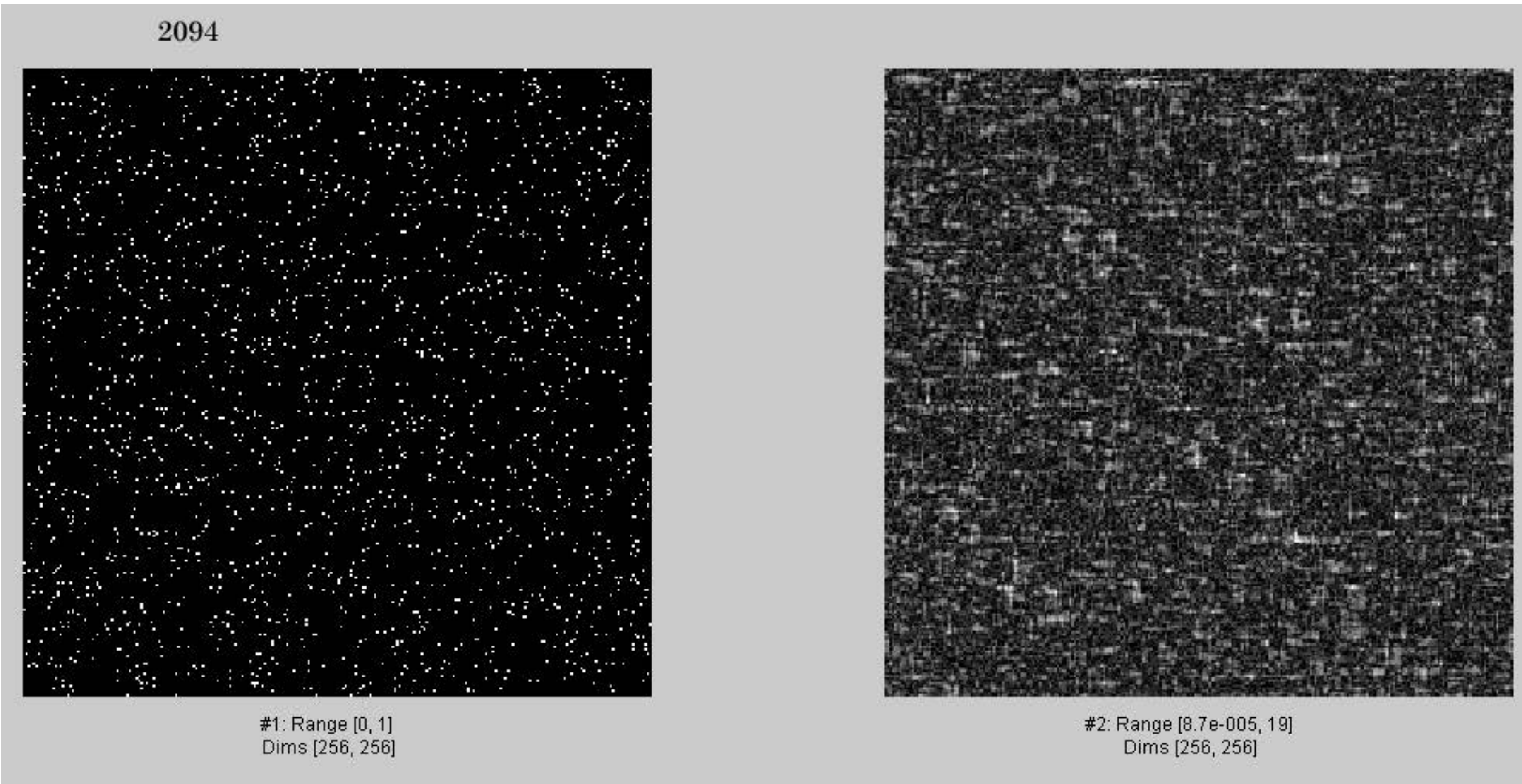
538



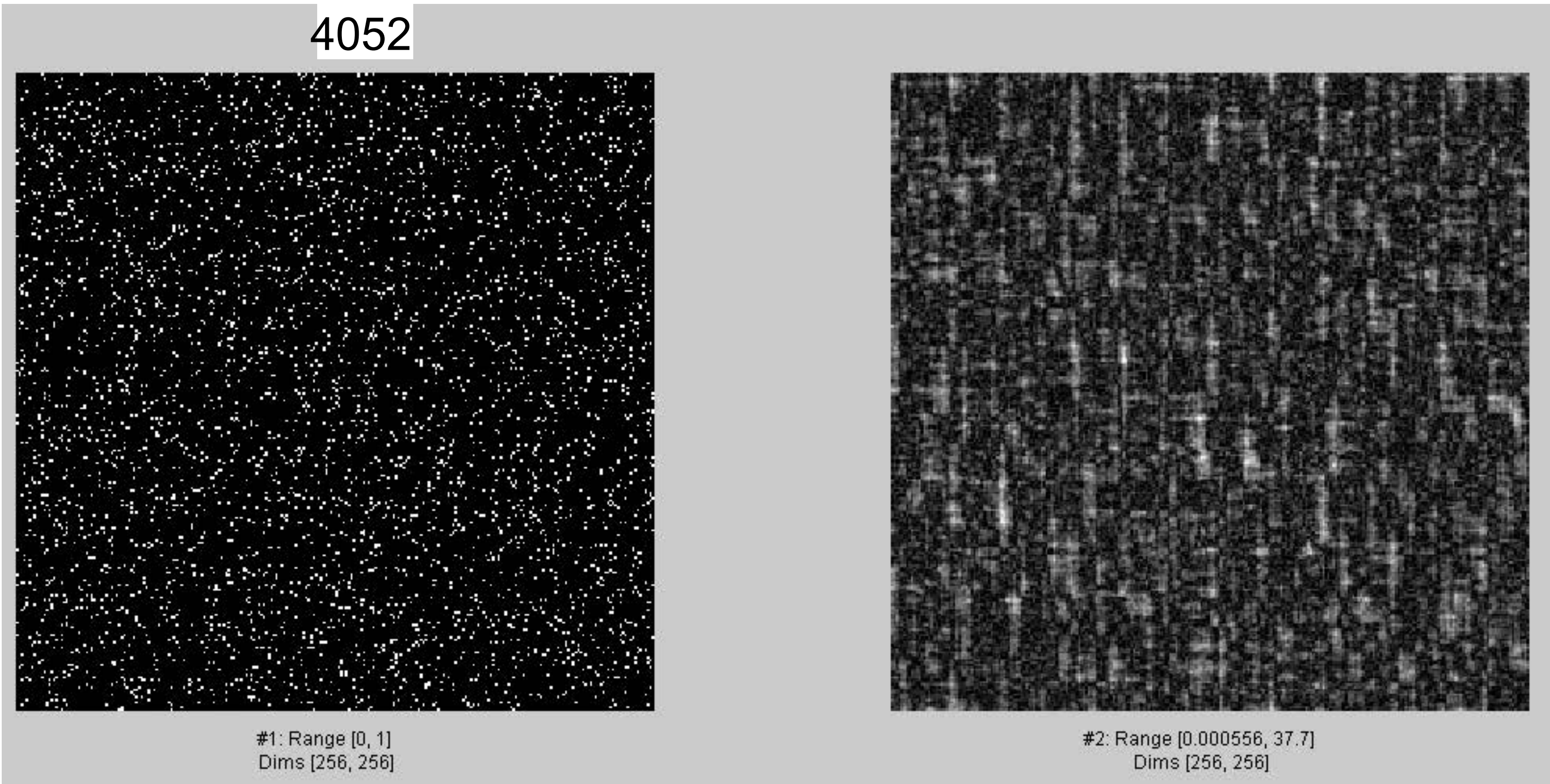
1088



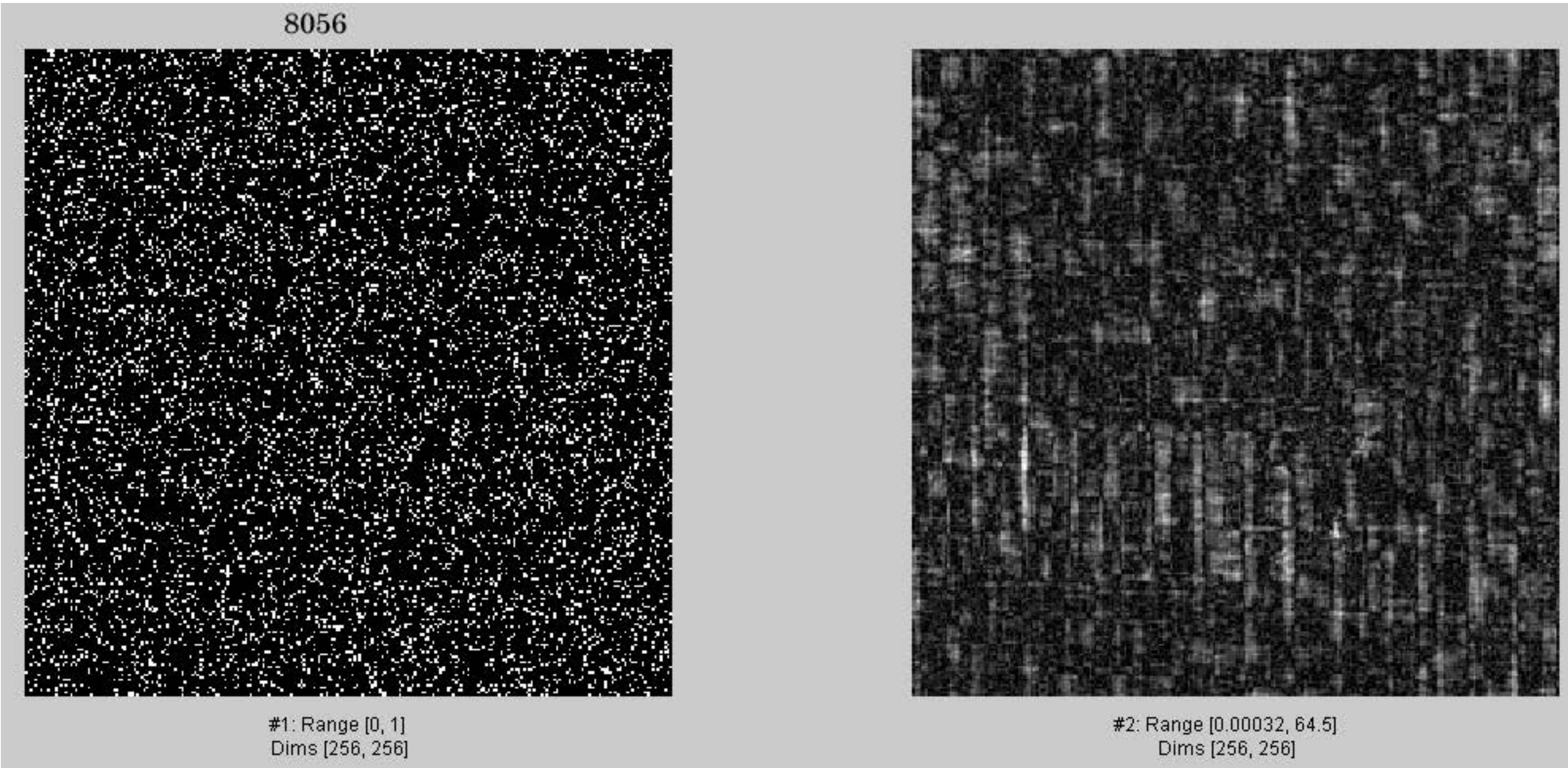
2094



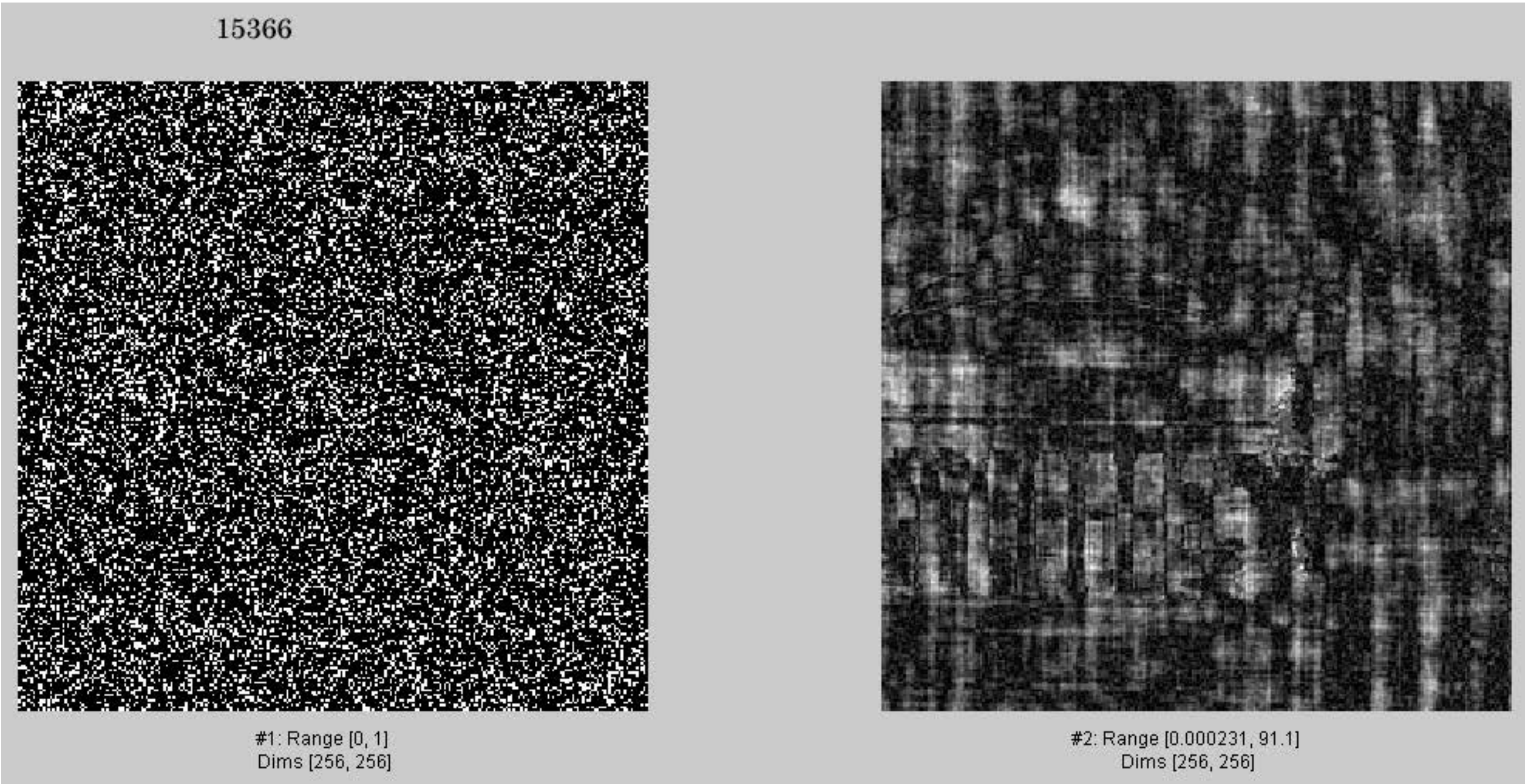
4052.



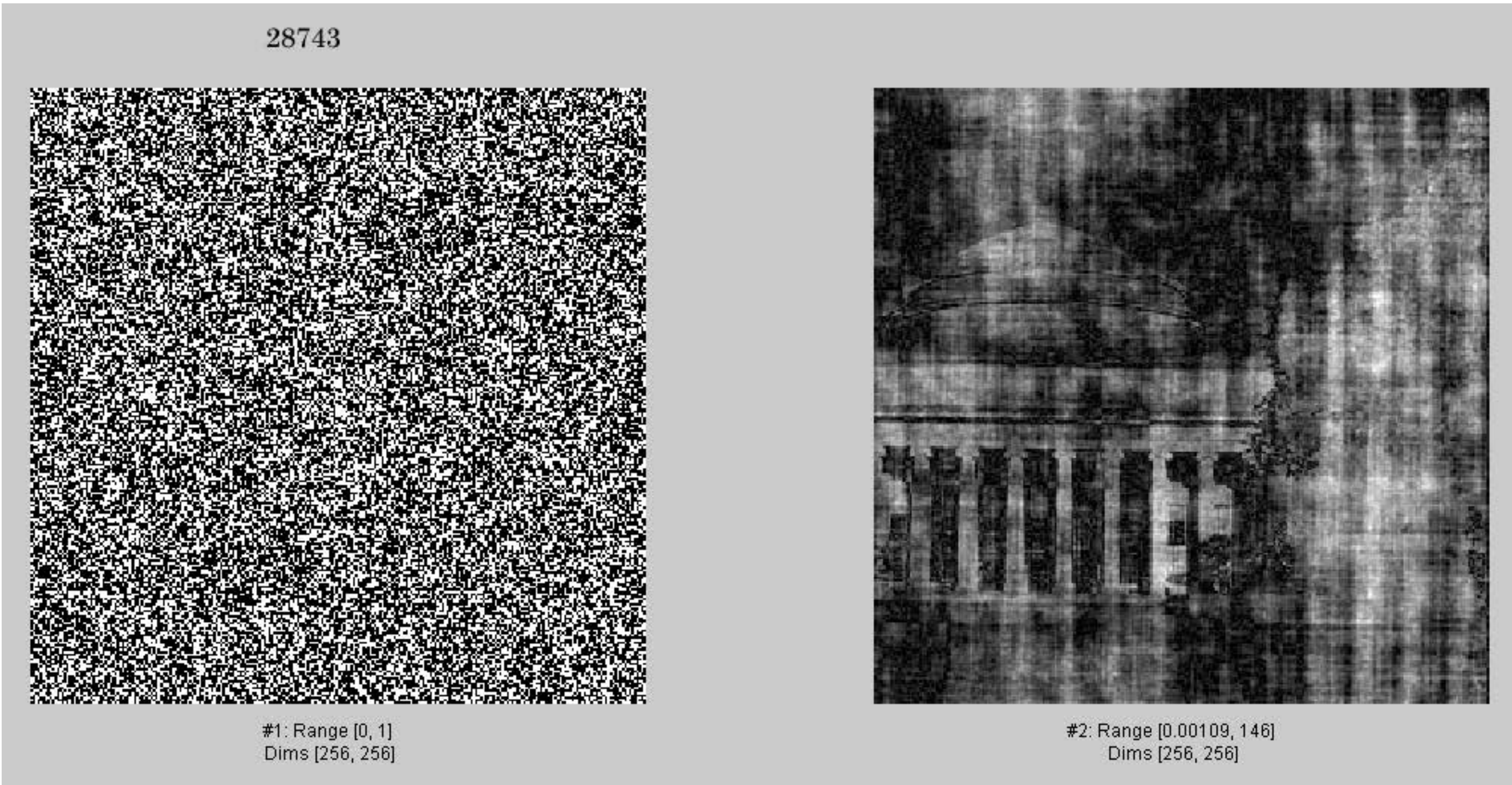
8056.



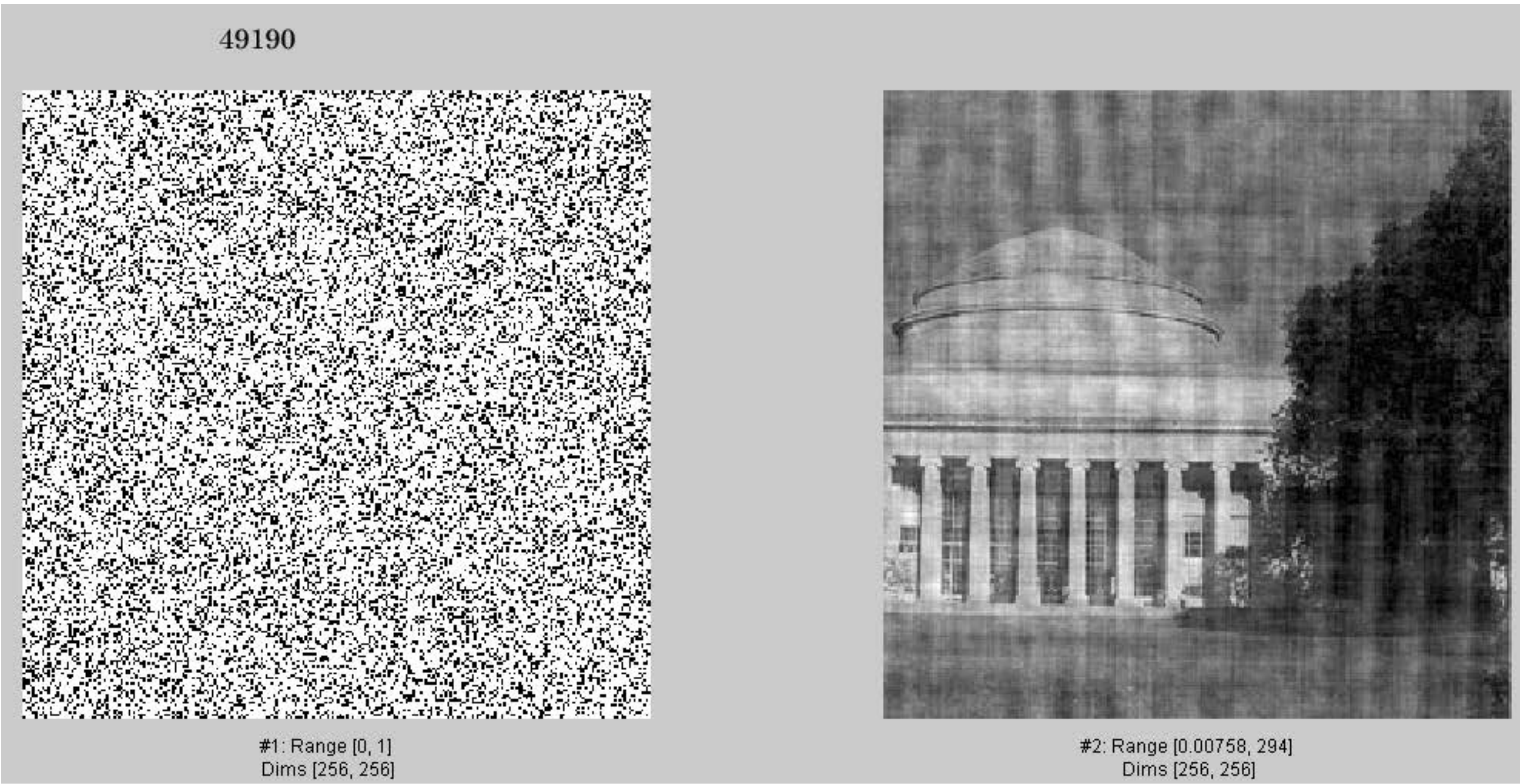
15366



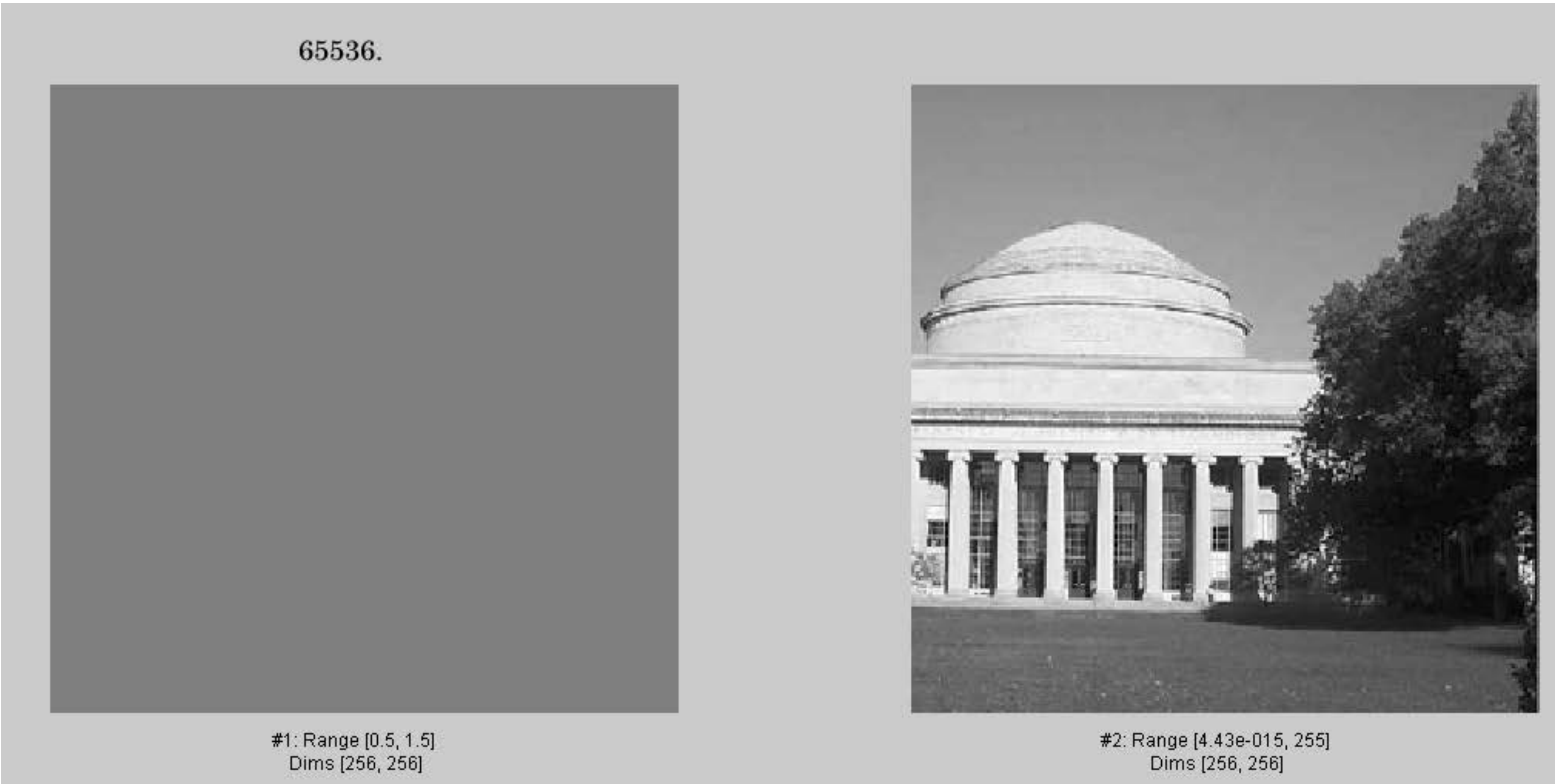
28743



49190.



65536.

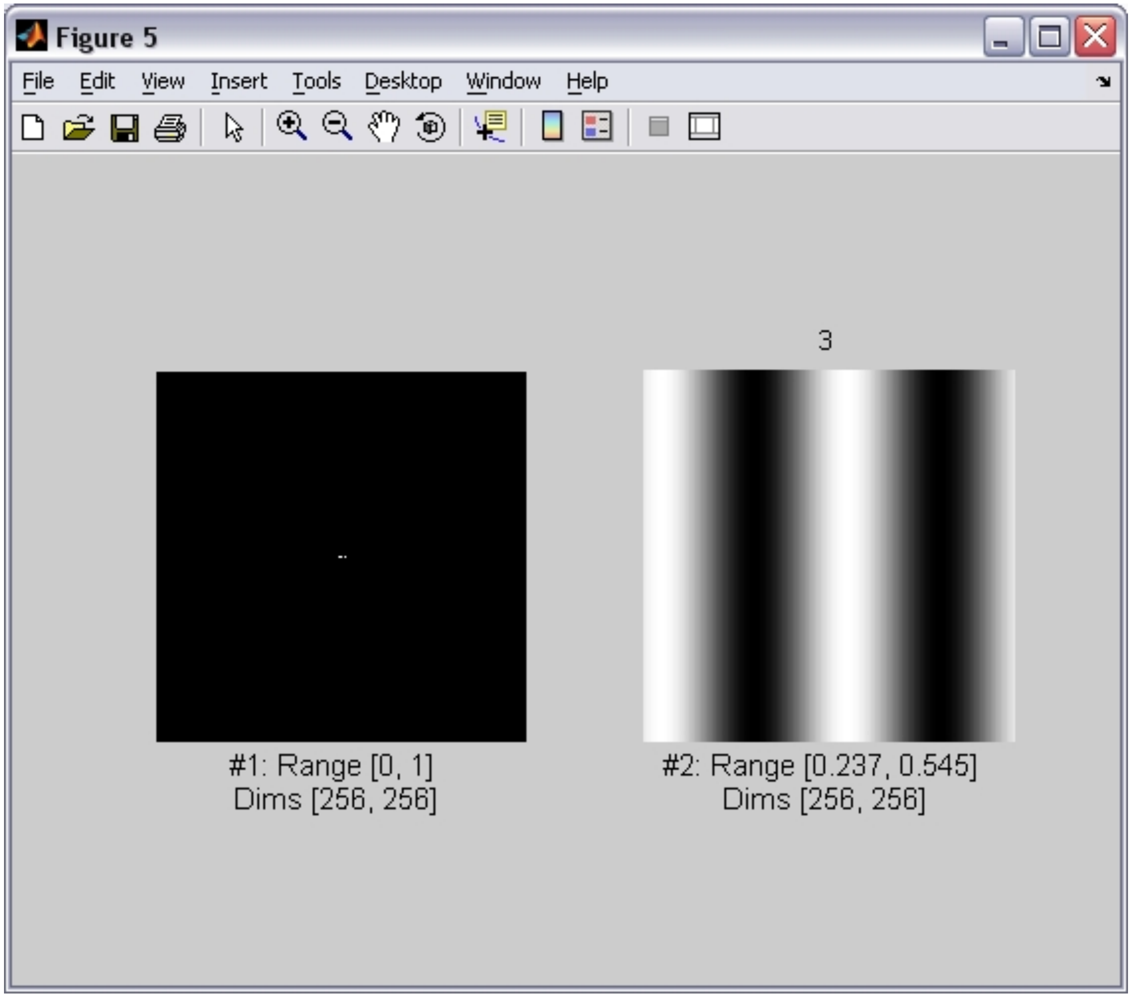

65536.
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [4.43e-015, 255]
Dims [256, 256]

Two examples of image synthesis with Fourier basis

First: randomly sample the Fourier coefficients of an image and reconstruct from those.

Second: sample Fourier coefficients in descending order of amplitude.

3



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

5

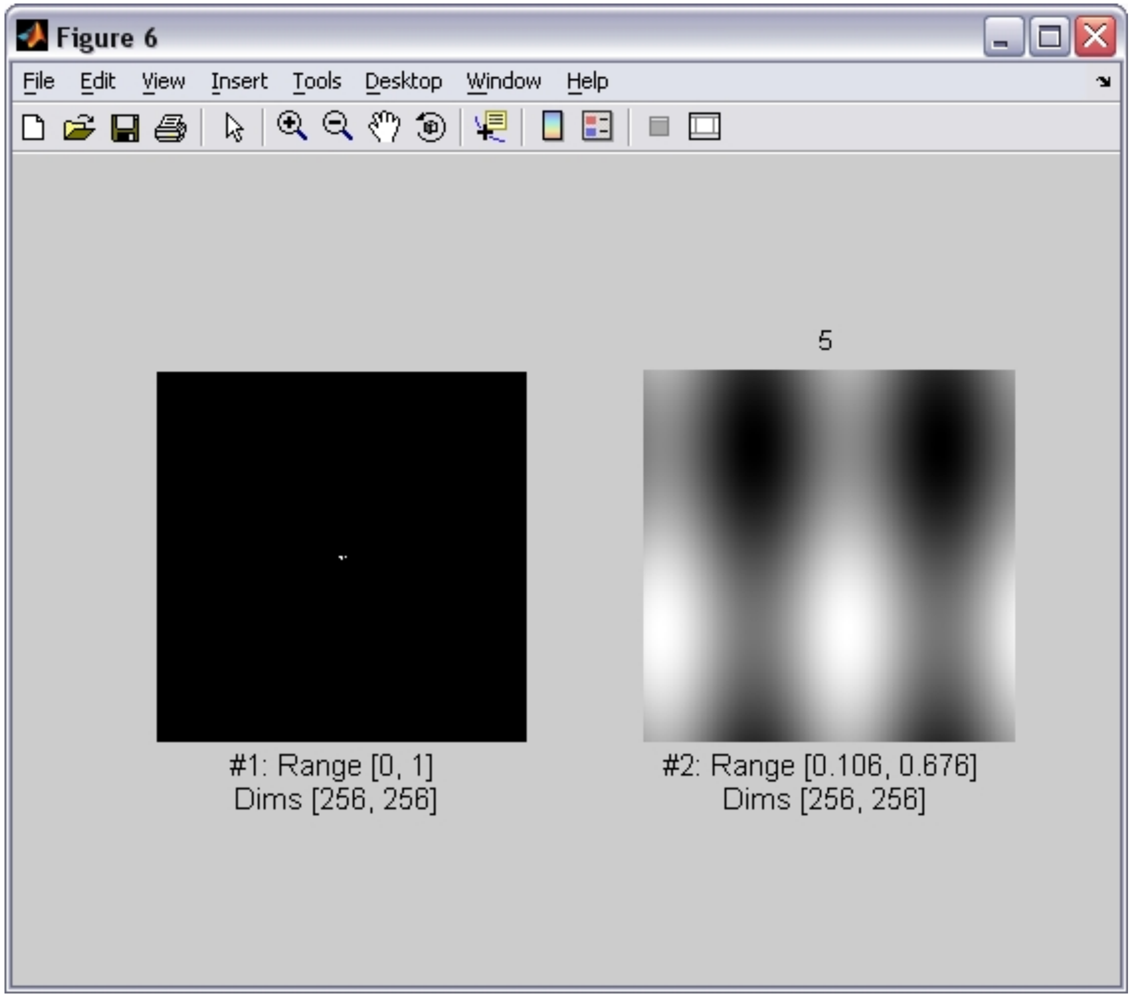


Figure 6
File
Edit
View
Insert
Tools
Desktop
Window
Help
5
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.106, 0.676]
Dims [256, 256]

9

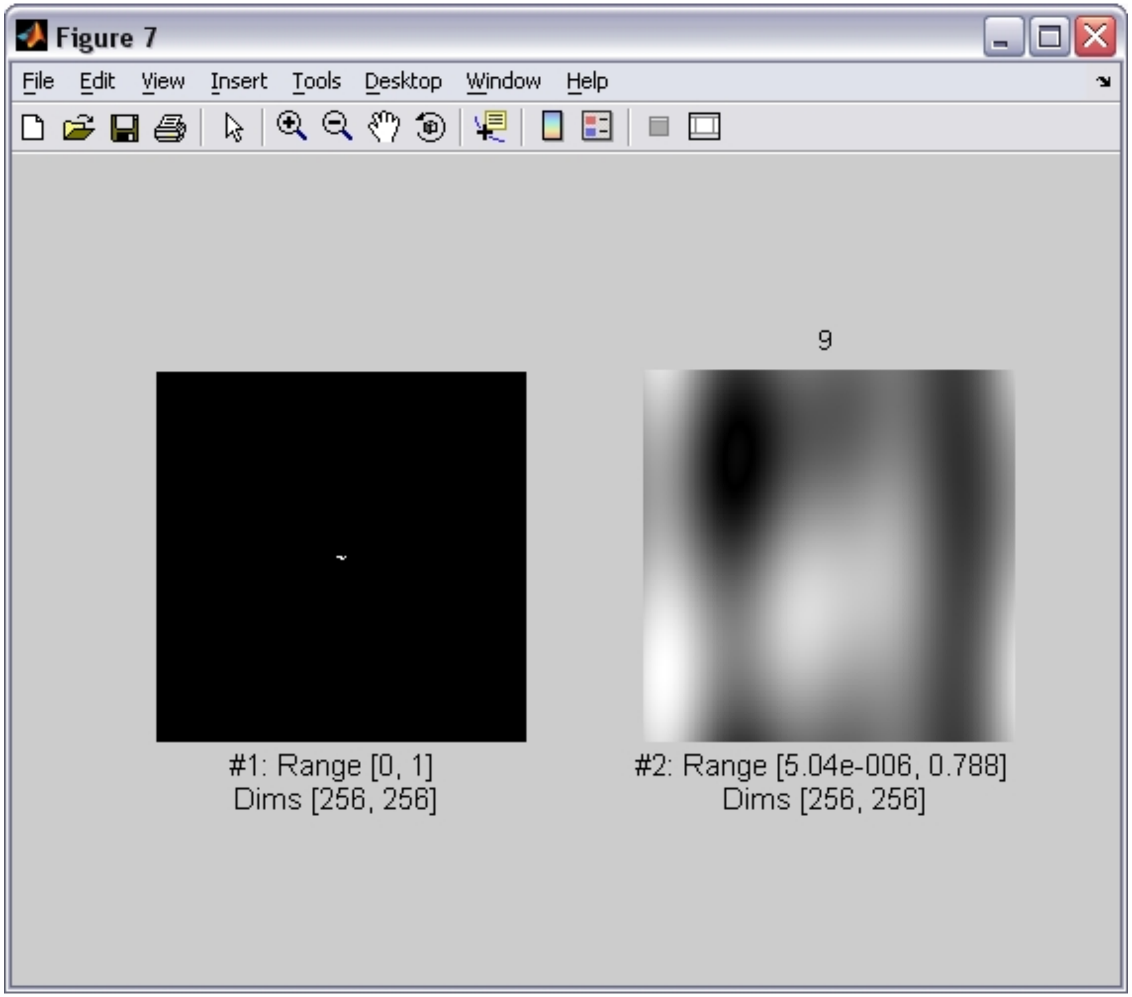


Figure 7
File Edit View Insert Tools Desktop Window Help
9
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.04e-006, 0.788]
Dims [256, 256]

17

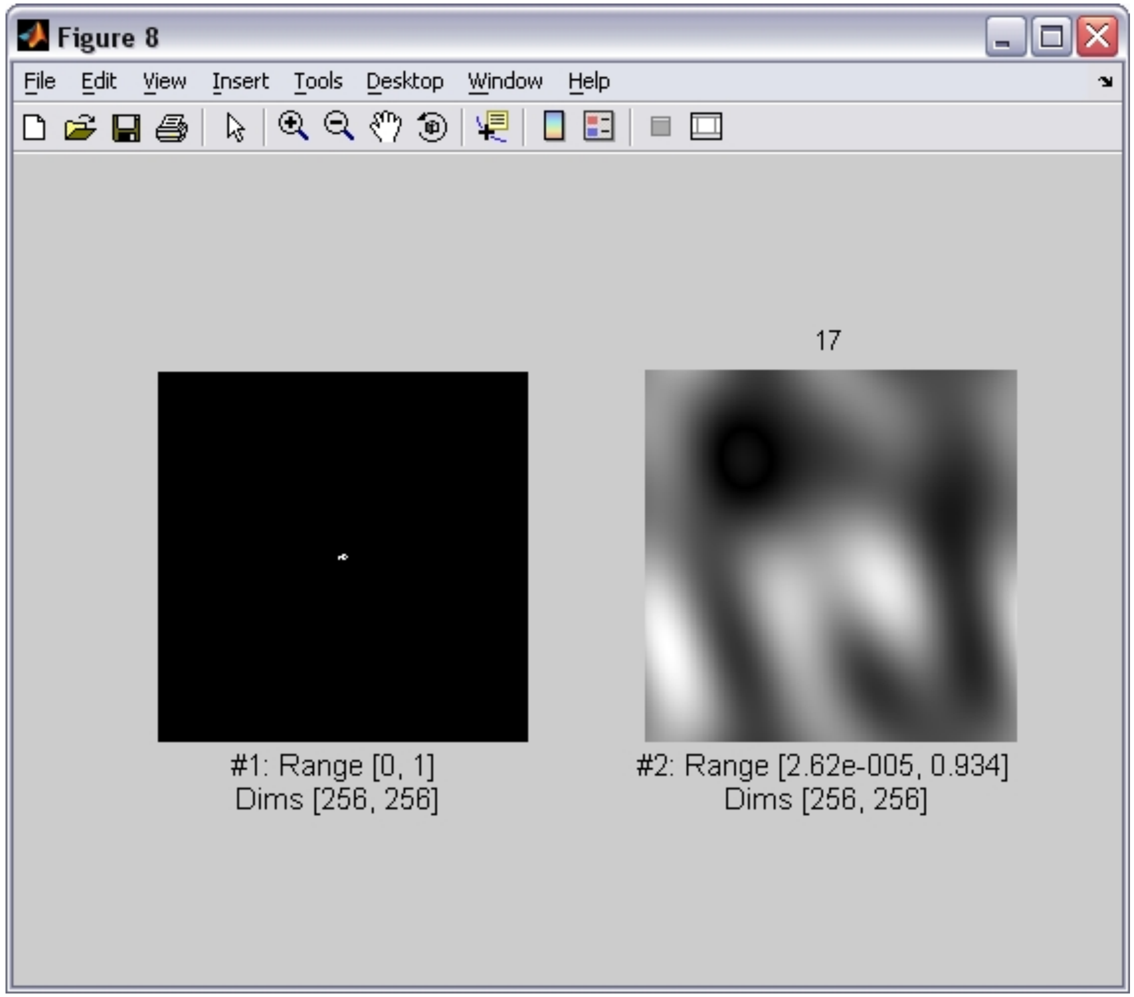


Figure 8
File
Edit
View
Insert
Tools
Desktop
Window
Help
17
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.62e-005, 0.934]
Dims [256, 256]

33

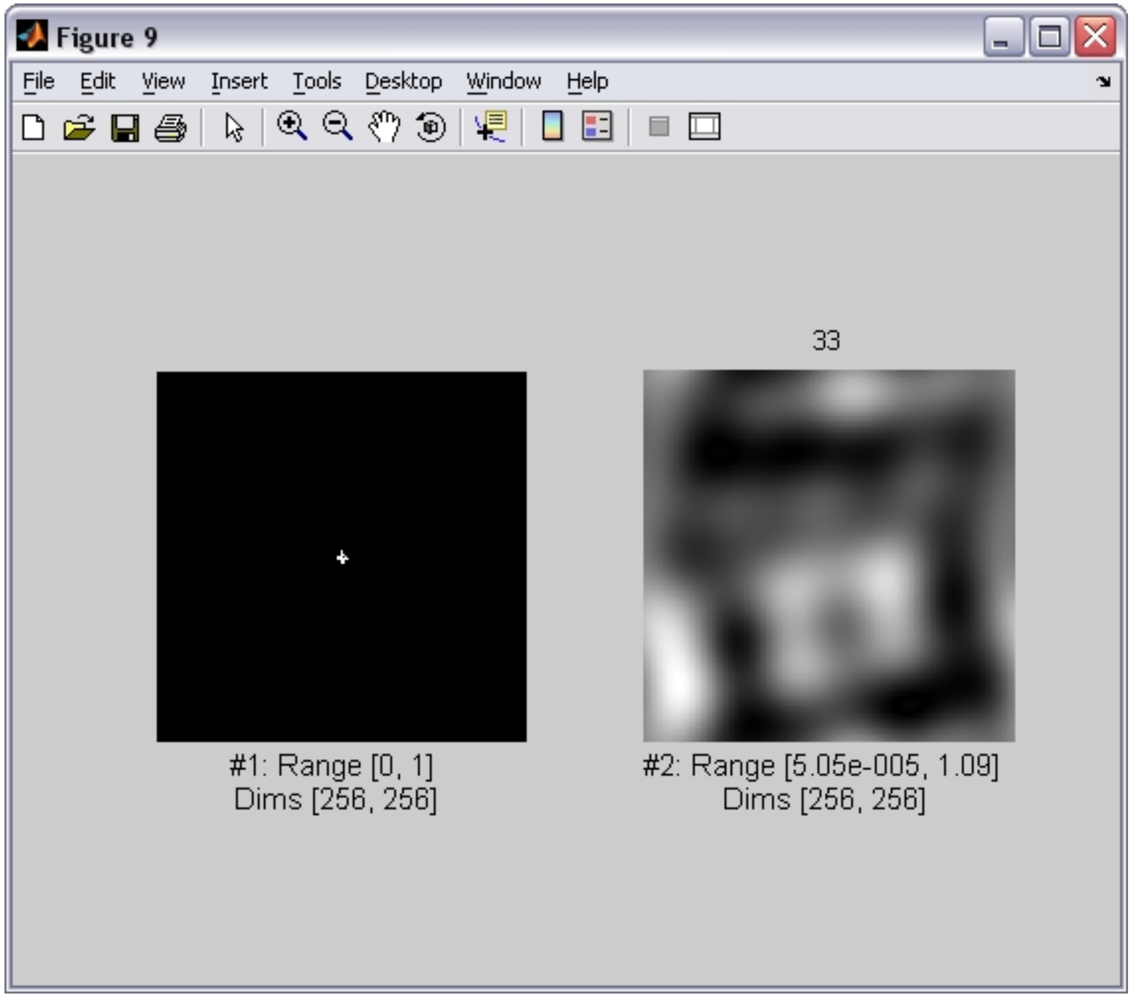


Figure 9
File
Edit
View
Insert
Tools
Desktop
Window
Help
33
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.05e-005, 1.09]
Dims [256, 256]

65

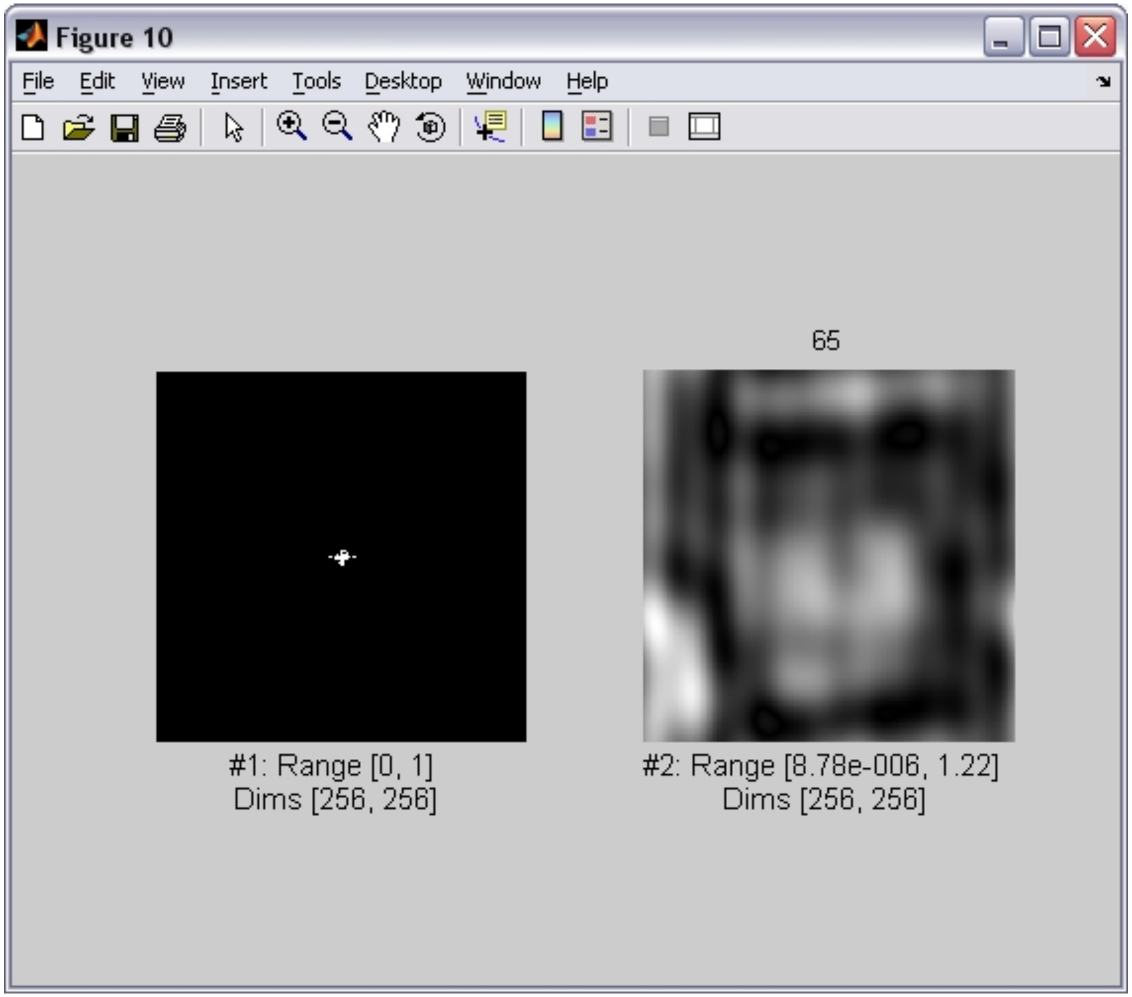


Figure 10
File Edit View Insert Tools Desktop Window Help
65
#1: Range [0, 1]
Dims [256, 256]
#2: Range [8.78e-006, 1.22]
Dims [256, 256]

129

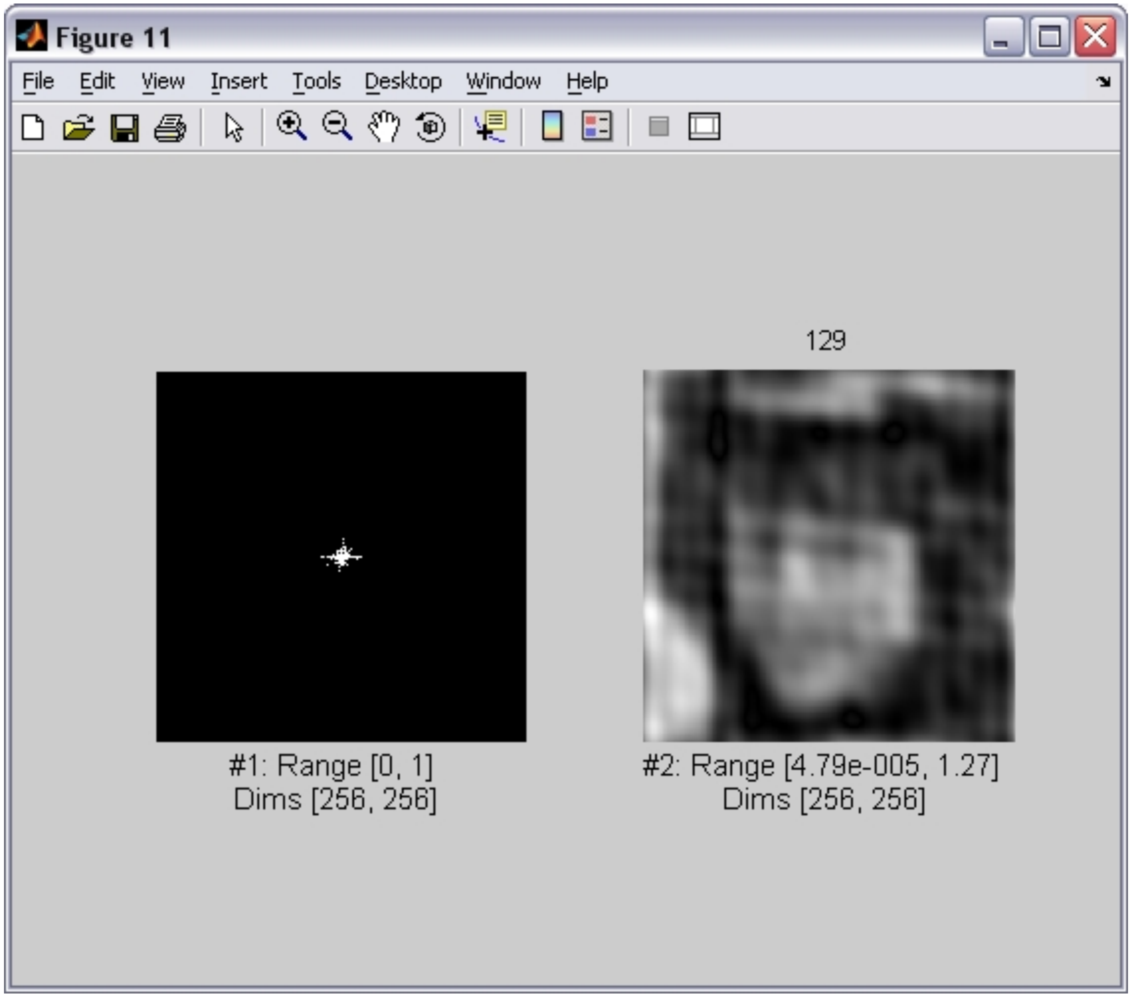


Figure 11
File Edit View Insert Tools Desktop Window Help
129
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.79e-005, 1.27]
Dims [256, 256]

257

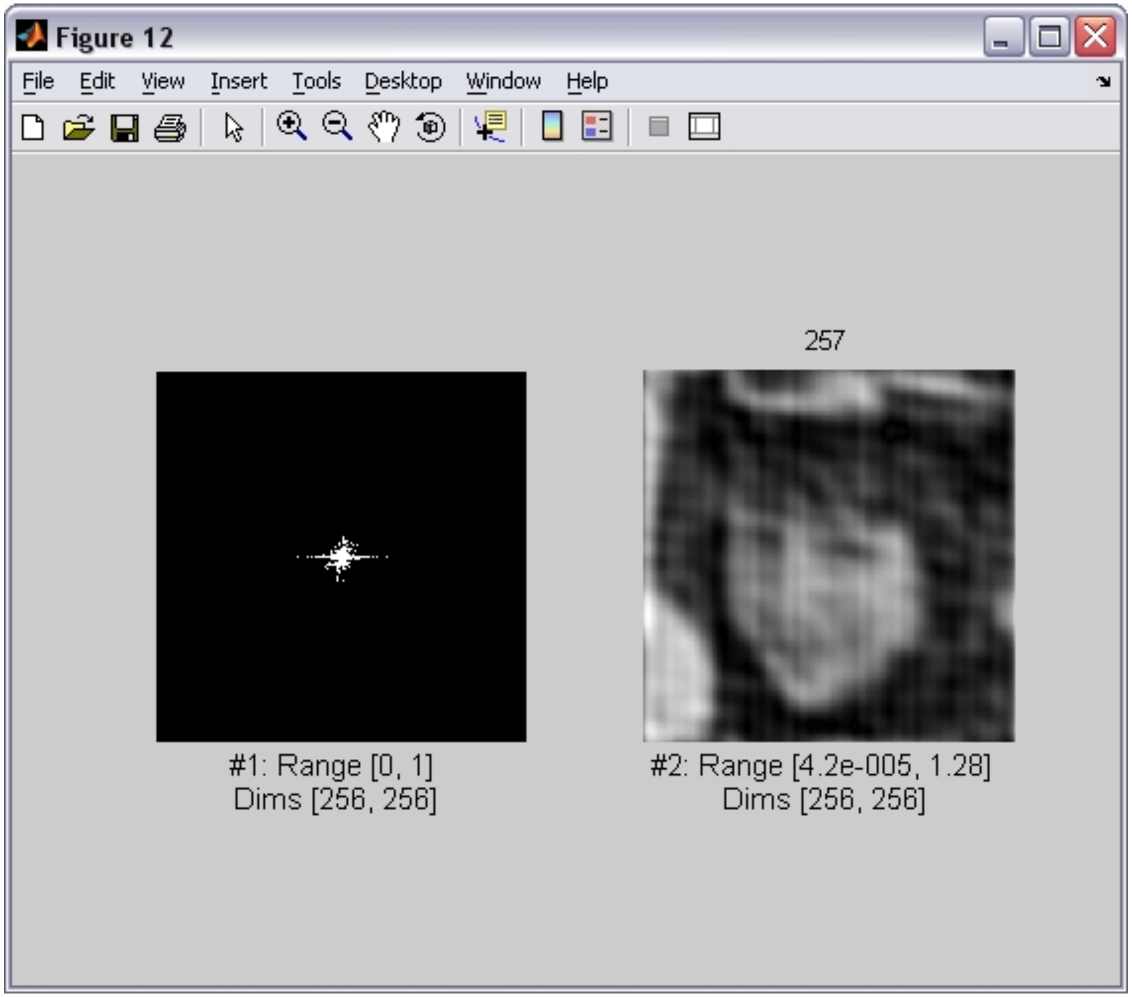


Figure 12
File Edit View Insert Tools Desktop Window Help
257
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.2e-005, 1.28]
Dims [256, 256]

513

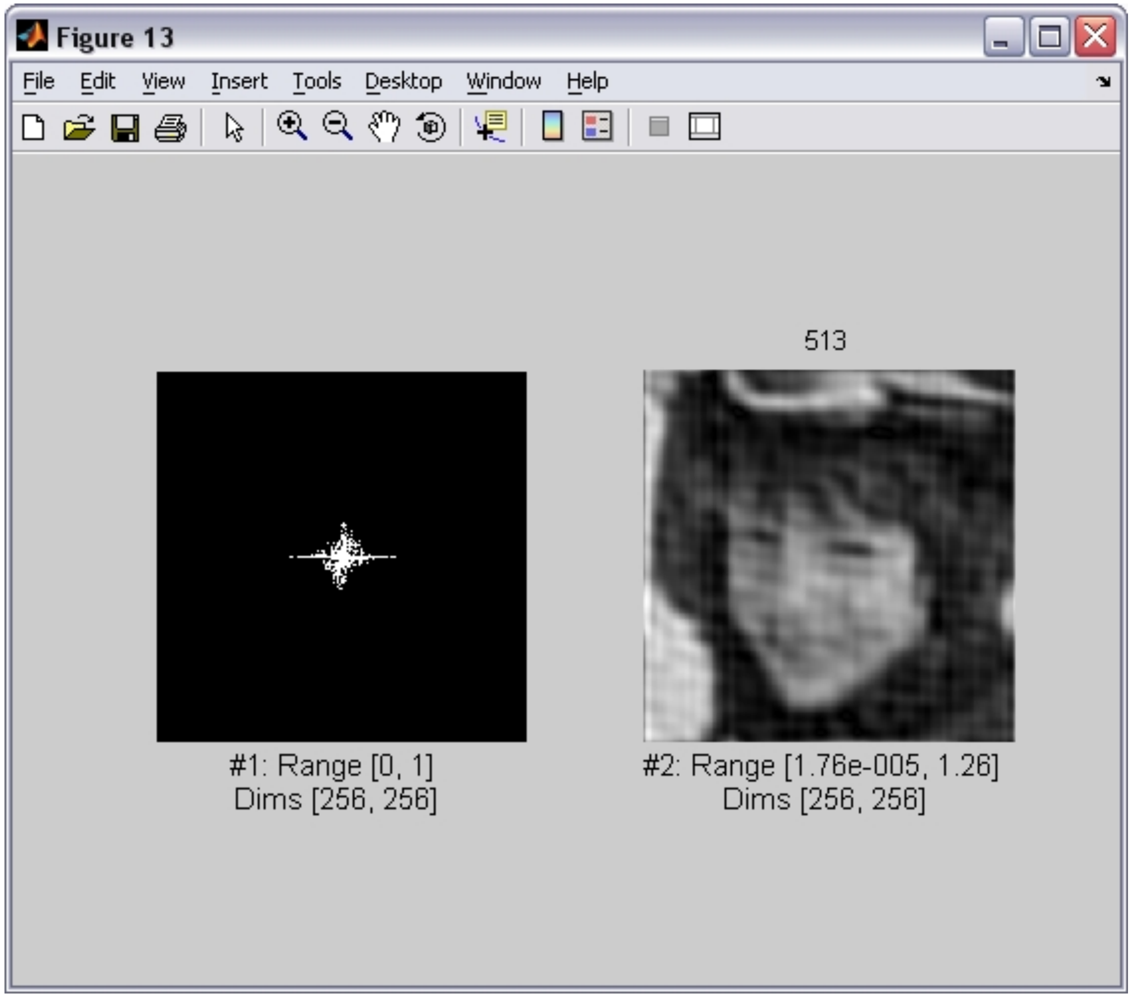


Figure 13
File
Edit
View
Insert
Tools
Desktop
Window
Help
513
#1: Range [0, 1]
Dims [256, 256]
#2: Range [1.76e-005, 1.26]
Dims [256, 256]

1025

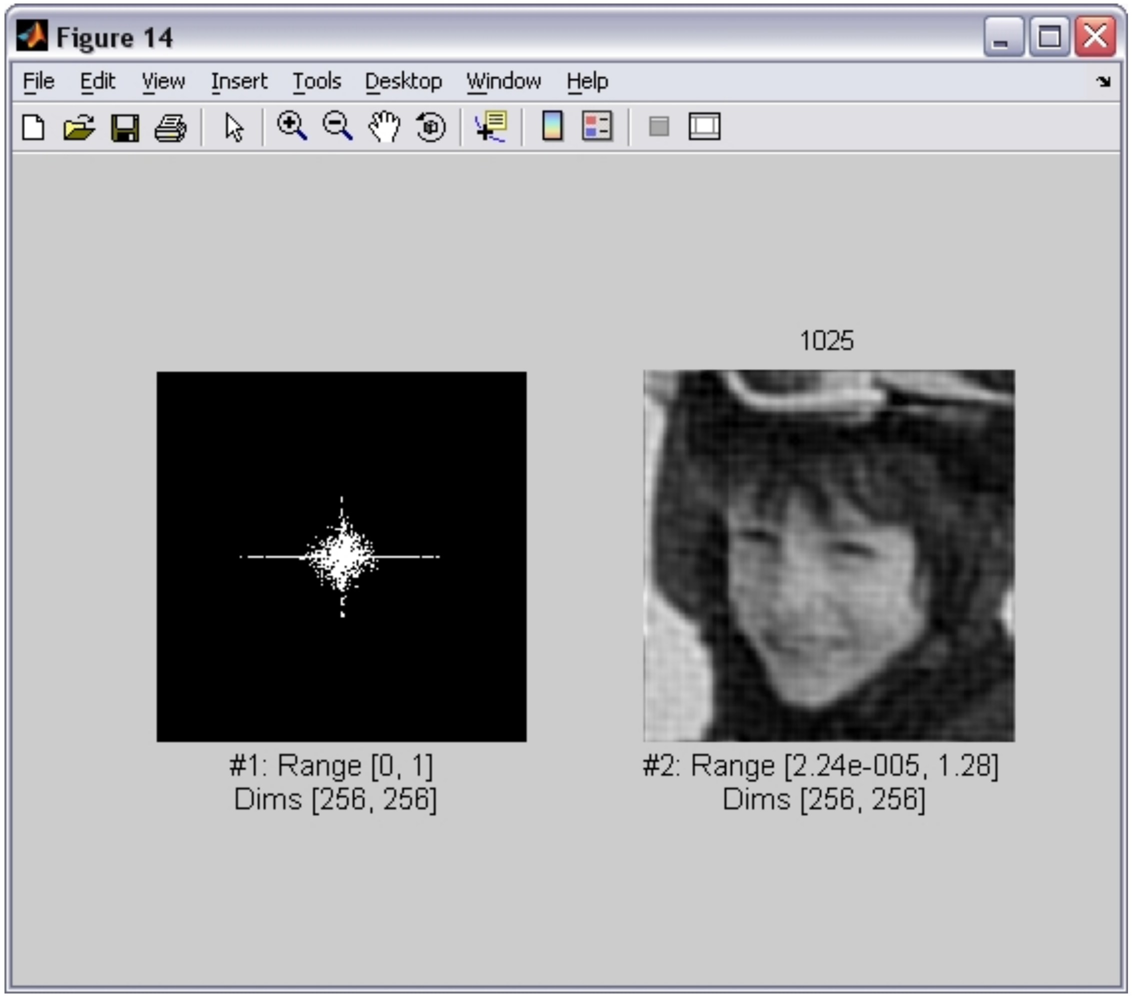


Figure 14
File Edit View Insert Tools Desktop Window Help
1025
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.24e-005, 1.28]
Dims [256, 256]

2049

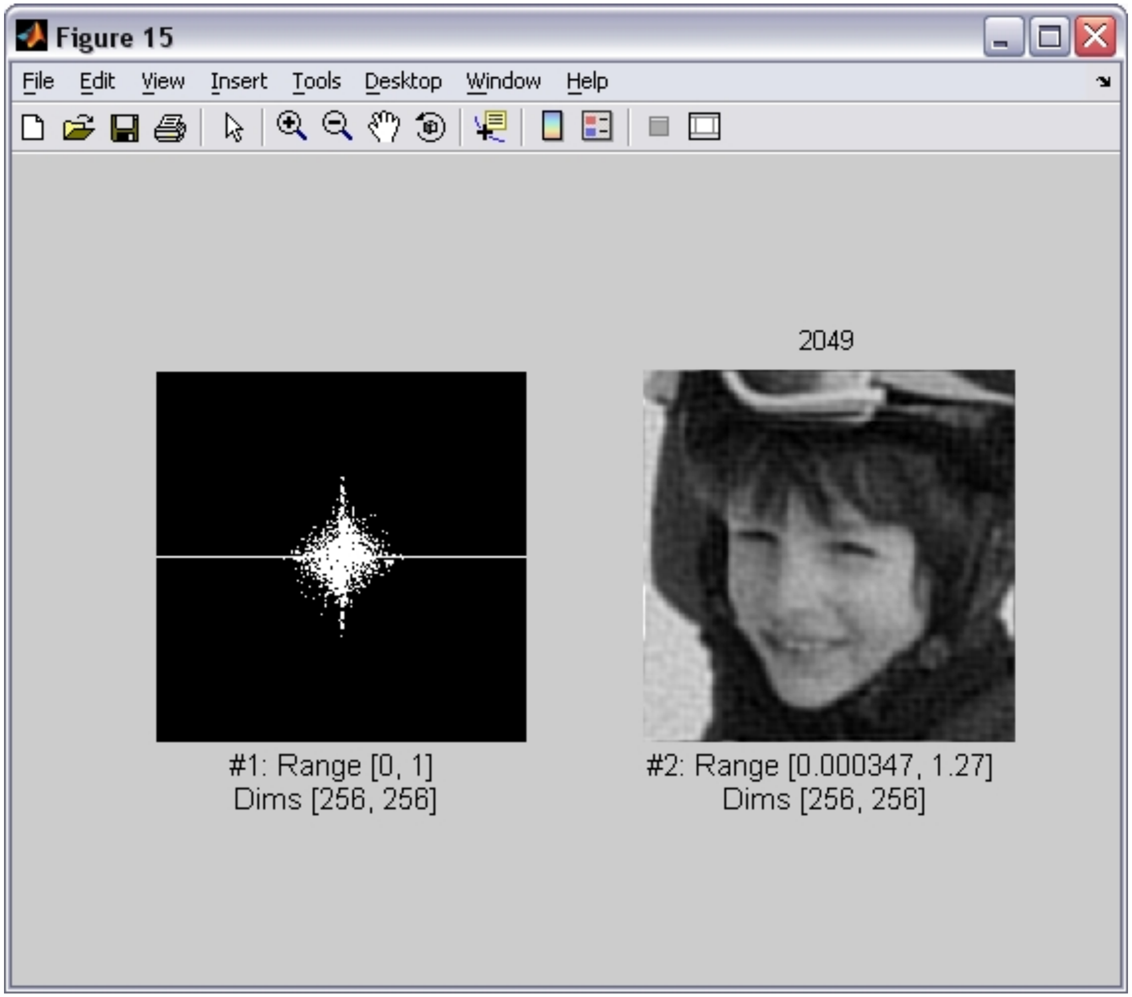


Figure 15
File Edit View Insert Tools Desktop Window Help
2049
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000347, 1.27]
Dims [256, 256]

4097

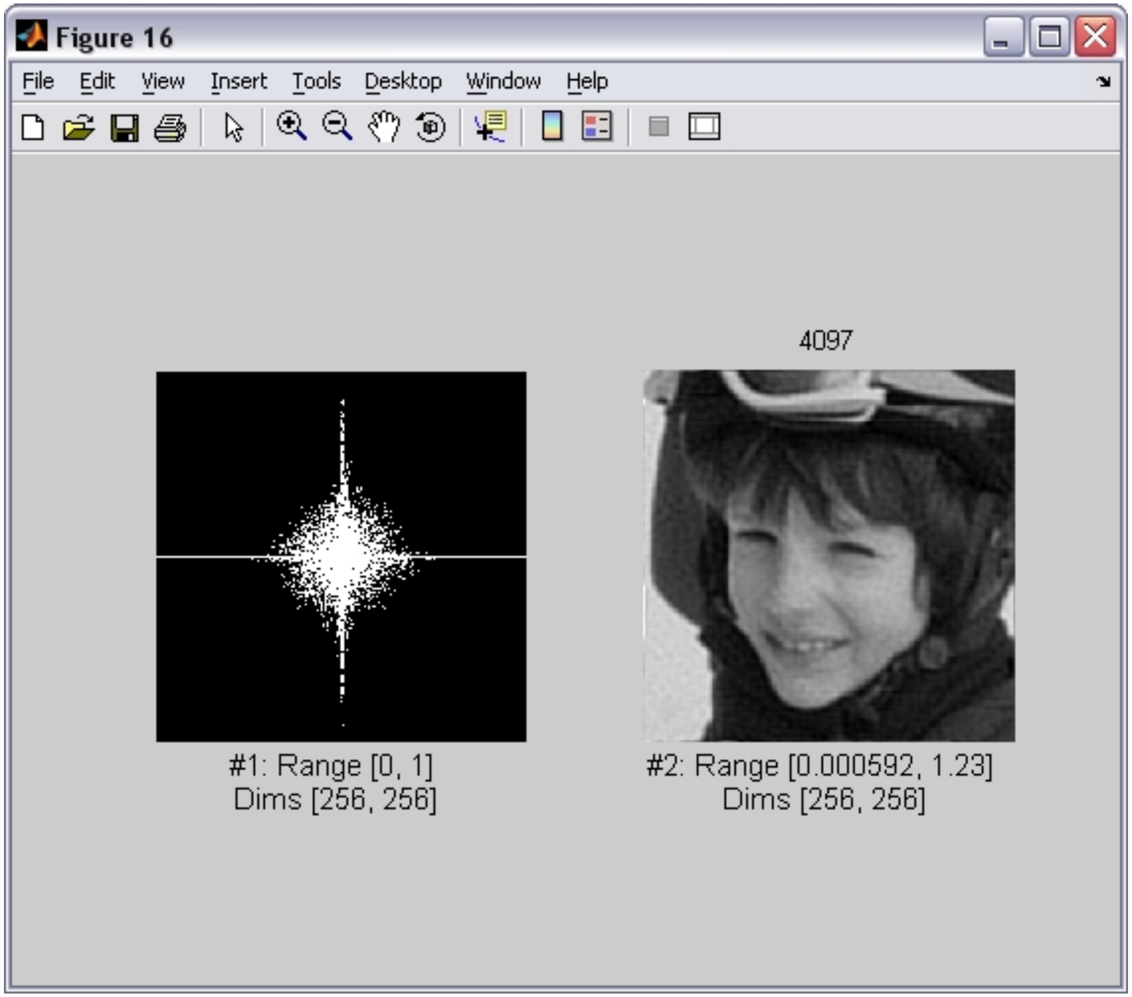


Figure 16
File Edit View Insert Tools Desktop Window Help
4097
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000592, 1.23]
Dims [256, 256]

8193

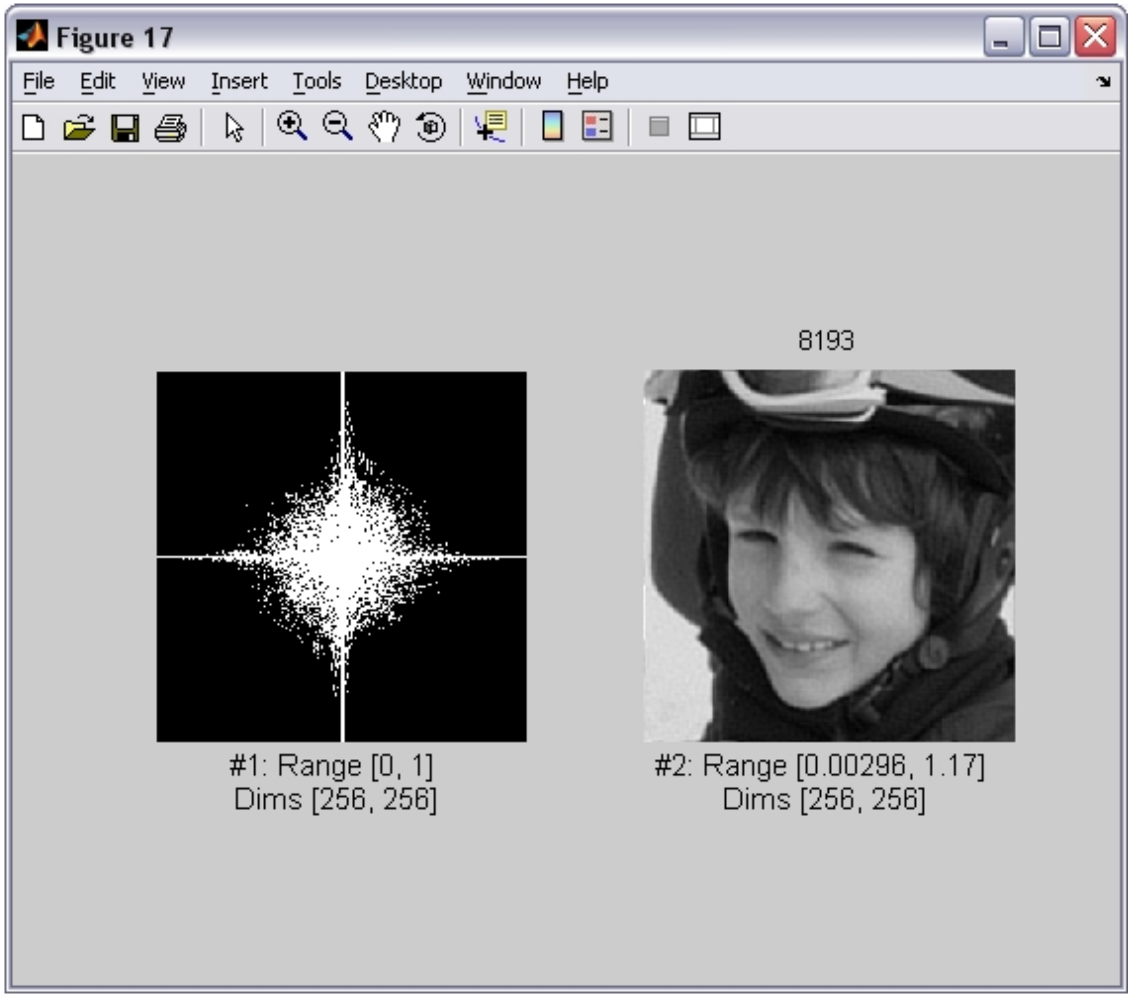


Figure 17
File
Edit
View
Insert
Tools
Desktop
Window
Help
8193
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.00296, 1.17]
Dims [256, 256]

16385

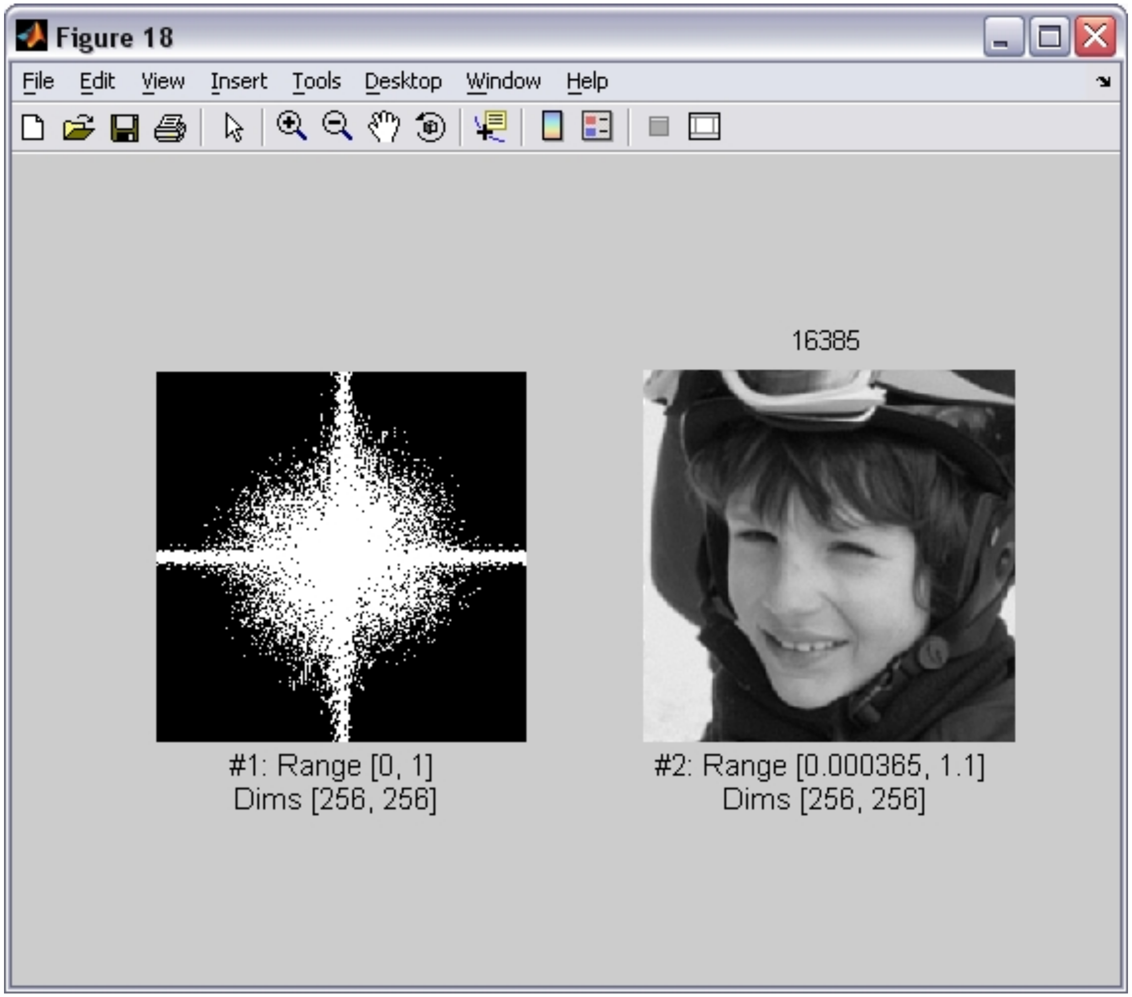


Figure 18
File Edit View Insert Tools Desktop Window Help
16385
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000365, 1.1]
Dims [256, 256]

32769

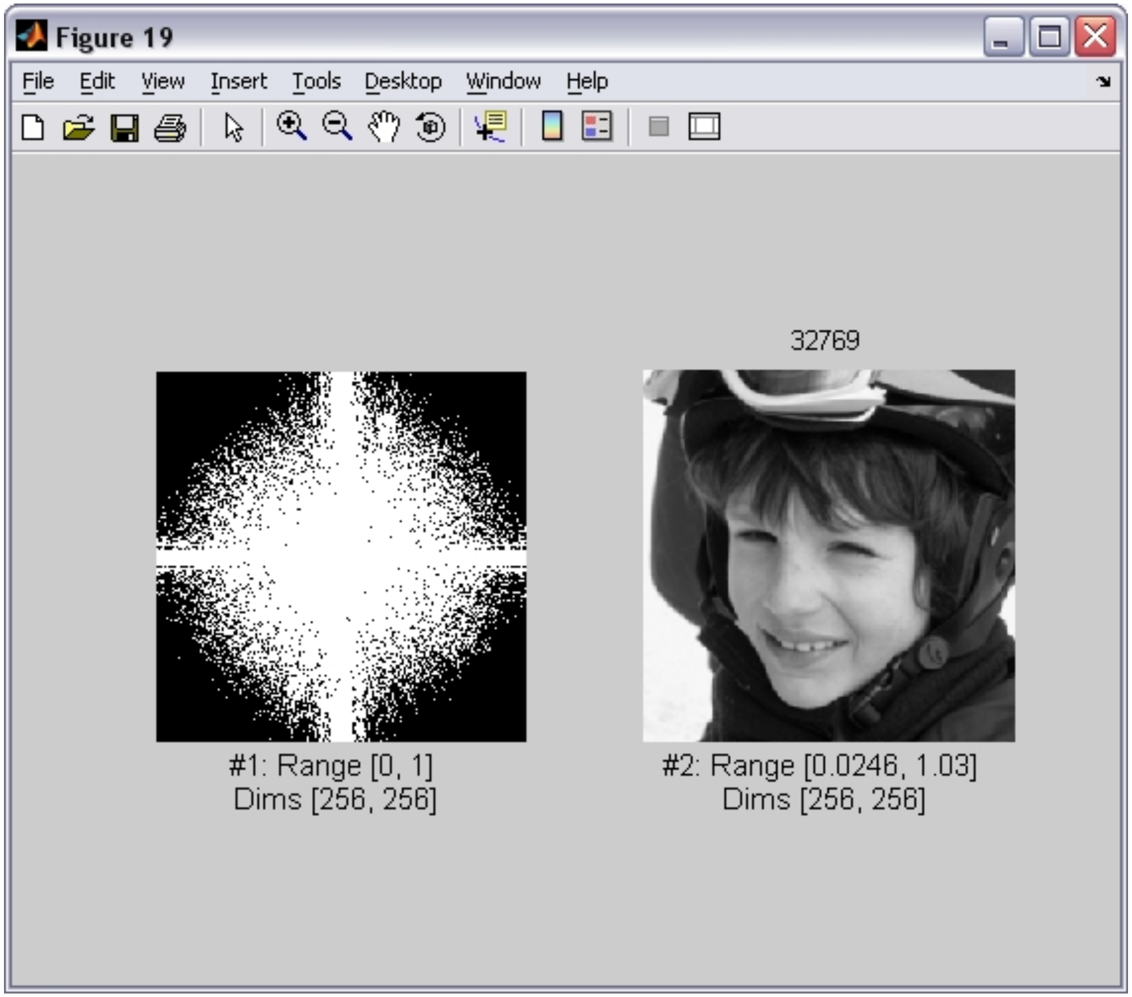


Figure 19
File Edit View Insert Tools Desktop Window Help
32769
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.0246, 1.03]
Dims [256, 256]

65536

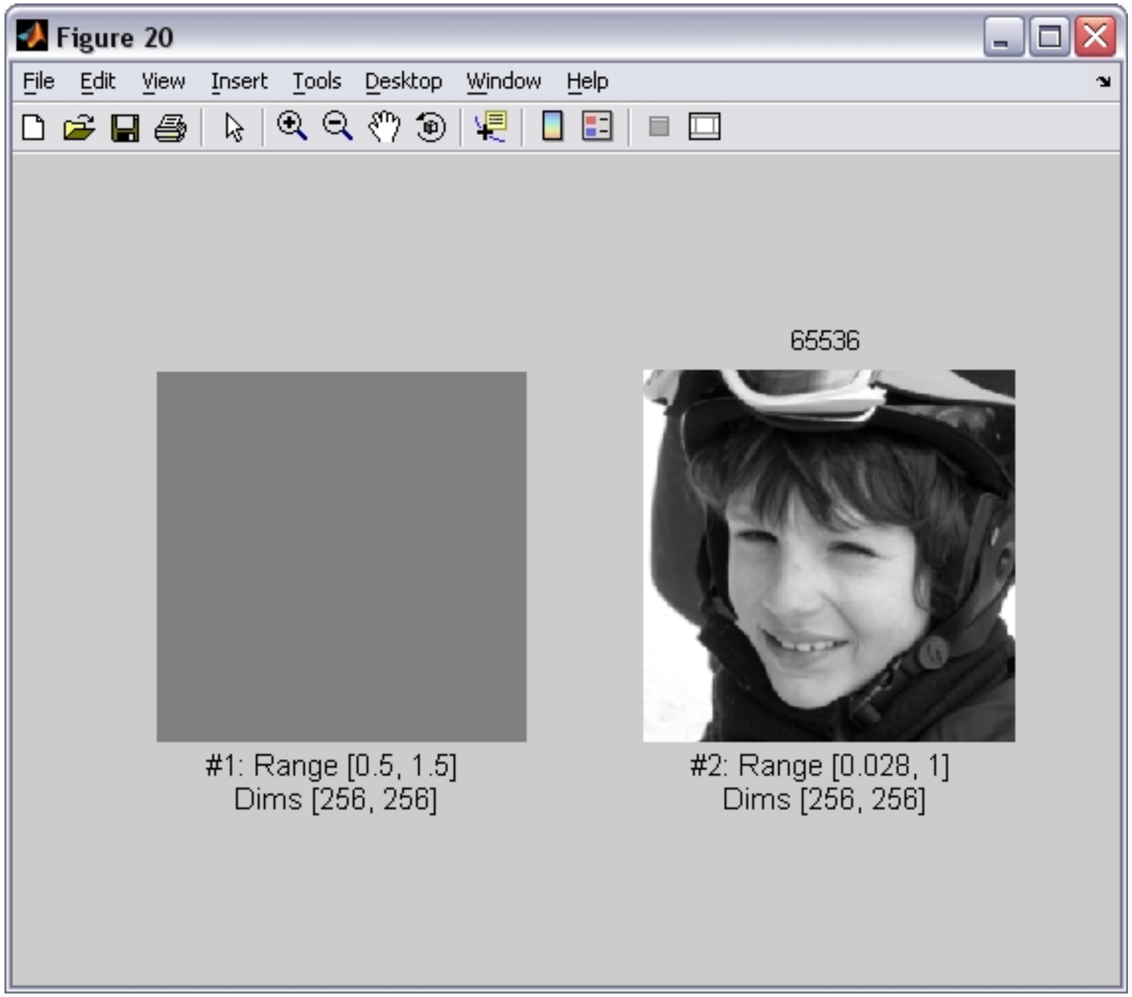
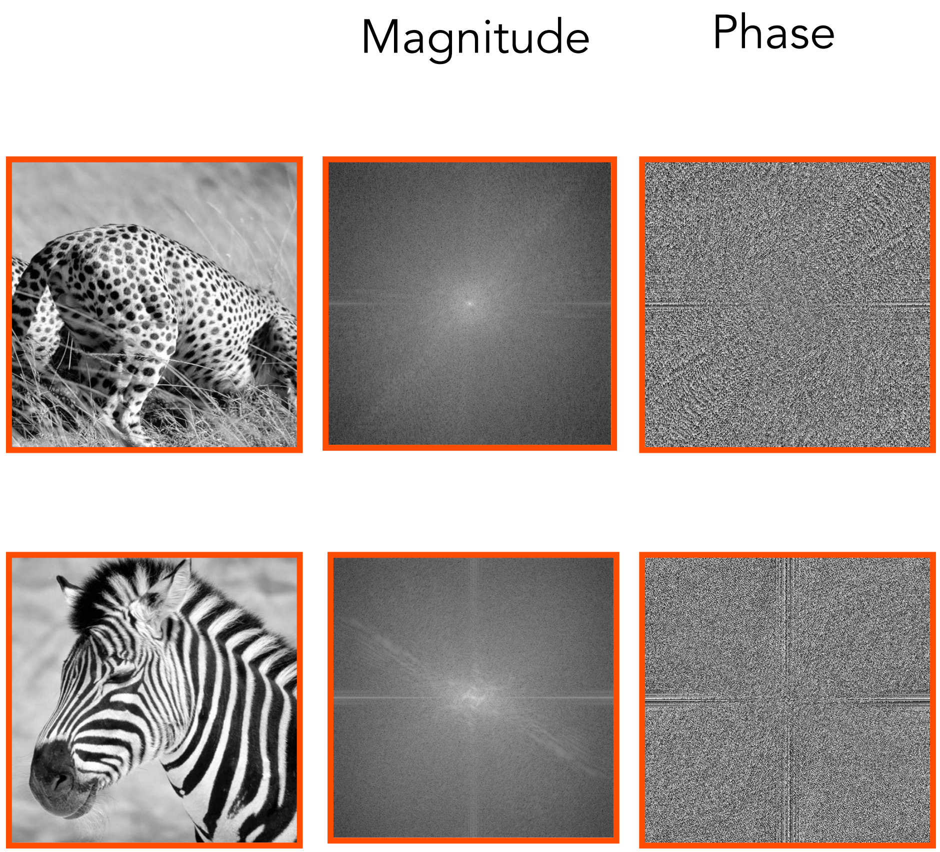


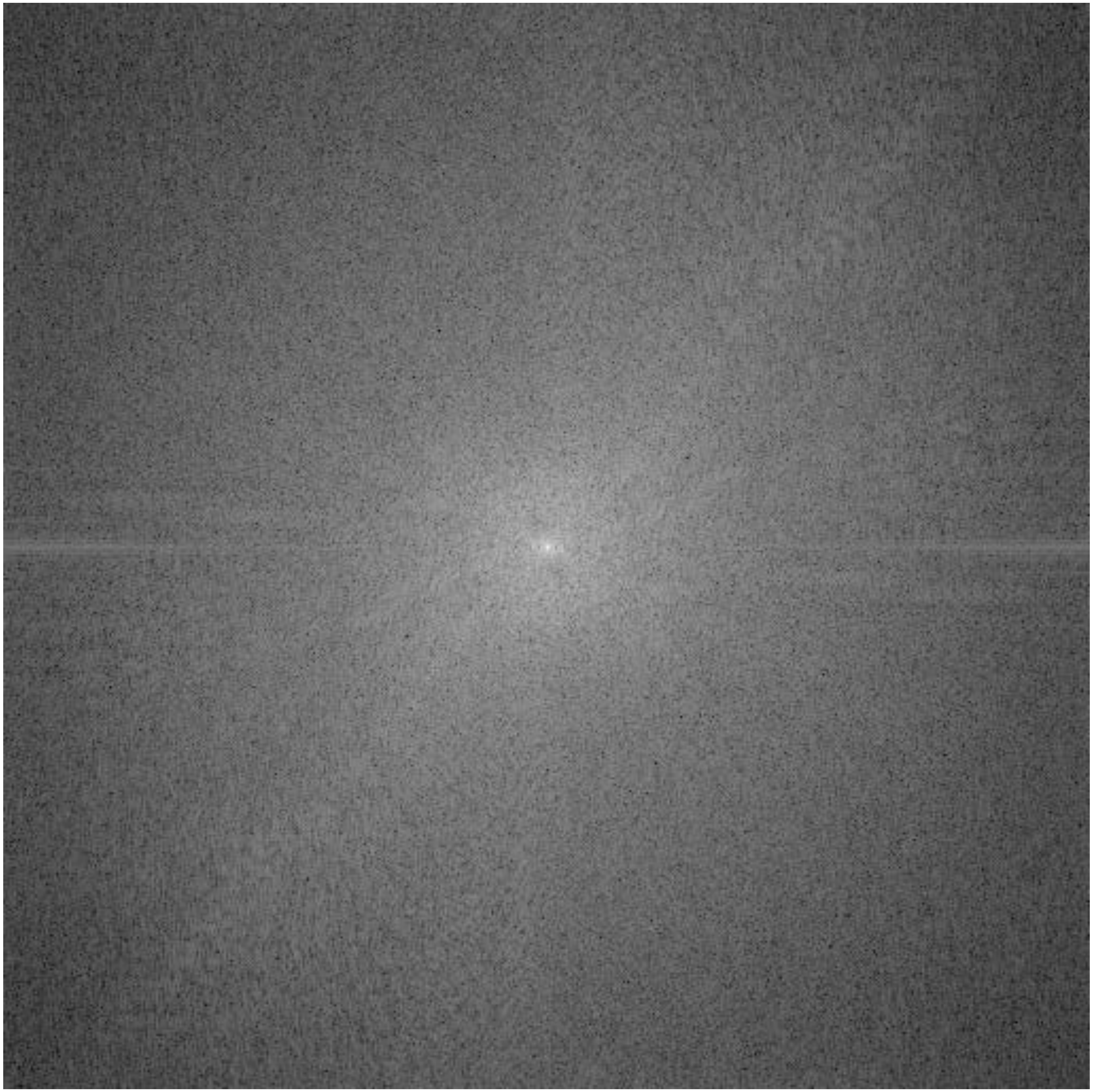
Figure 20
File Edit View Insert Tools Desktop Window Help
65536
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [0.028, 1]
Dims [256, 256]

Fourier Transform

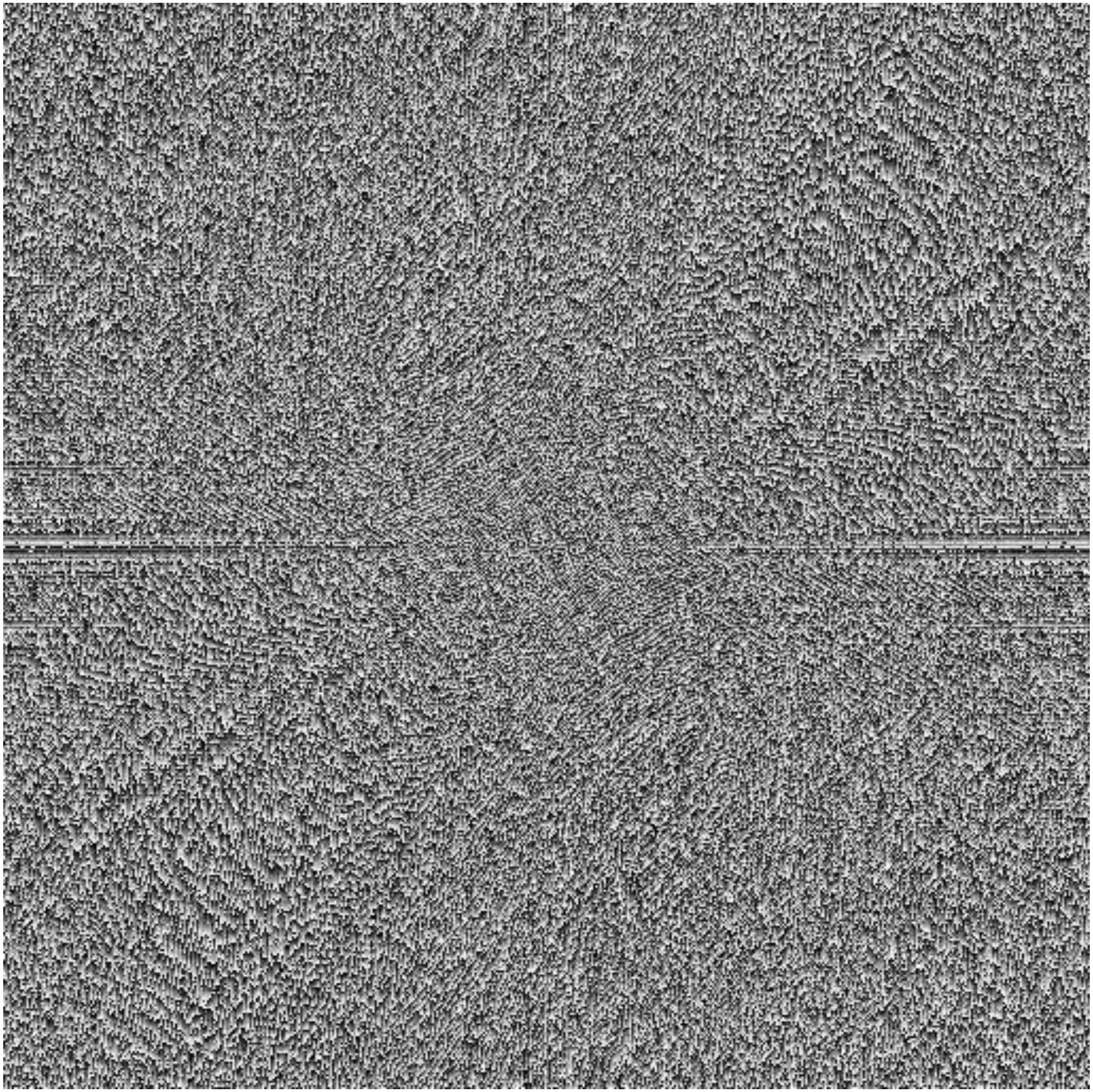
- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- The magnitude of natural images can often be quite similar, one to another. But magnitude **encodes statistics of orientation at all spatial scales.**
- The phase carry the information of **where the image contours are**, by specifying how the phases of the sinusoids must line up in order to create the observed contours and edges.



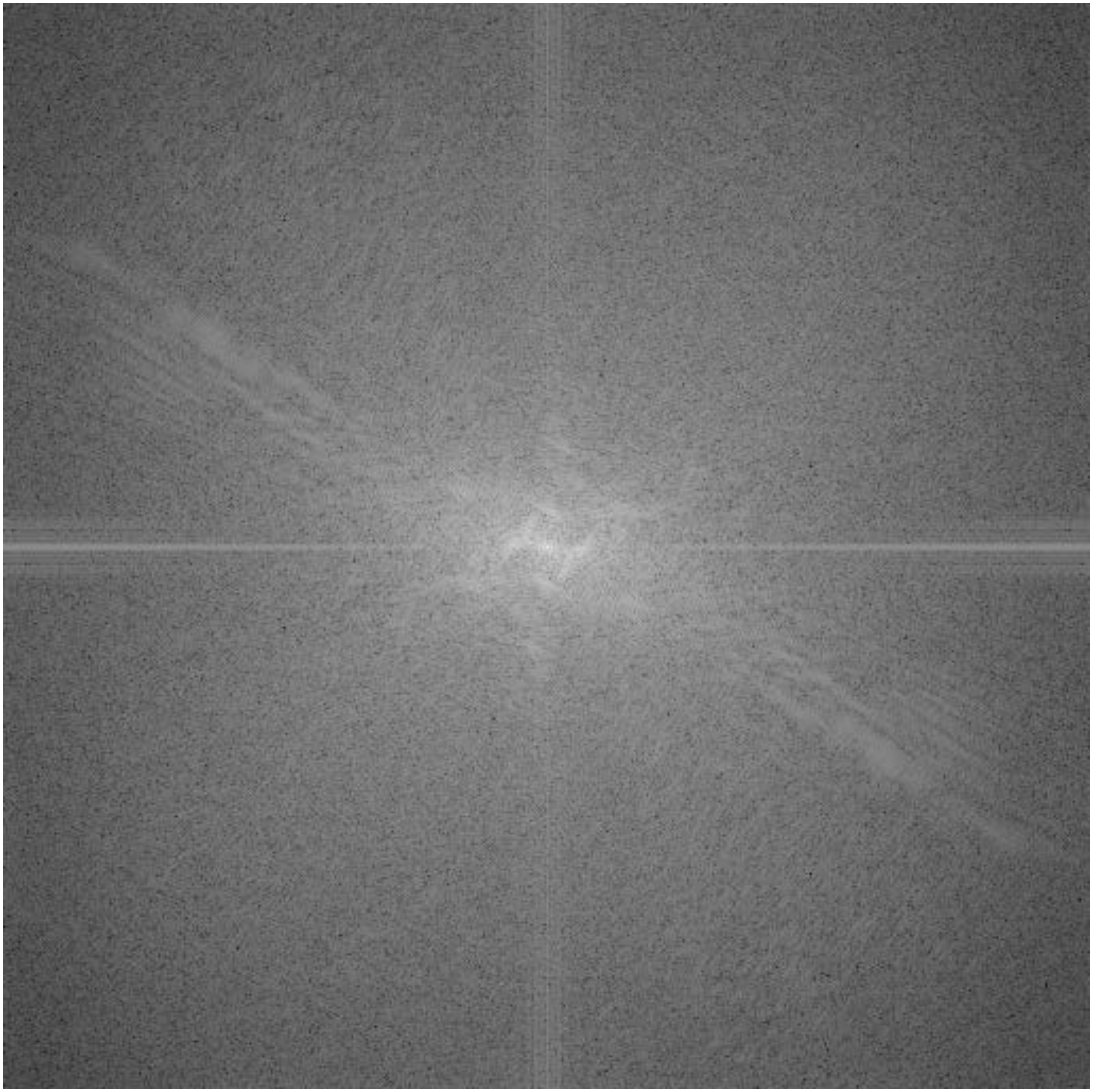
This is the magnitude transform of the cheetah pic



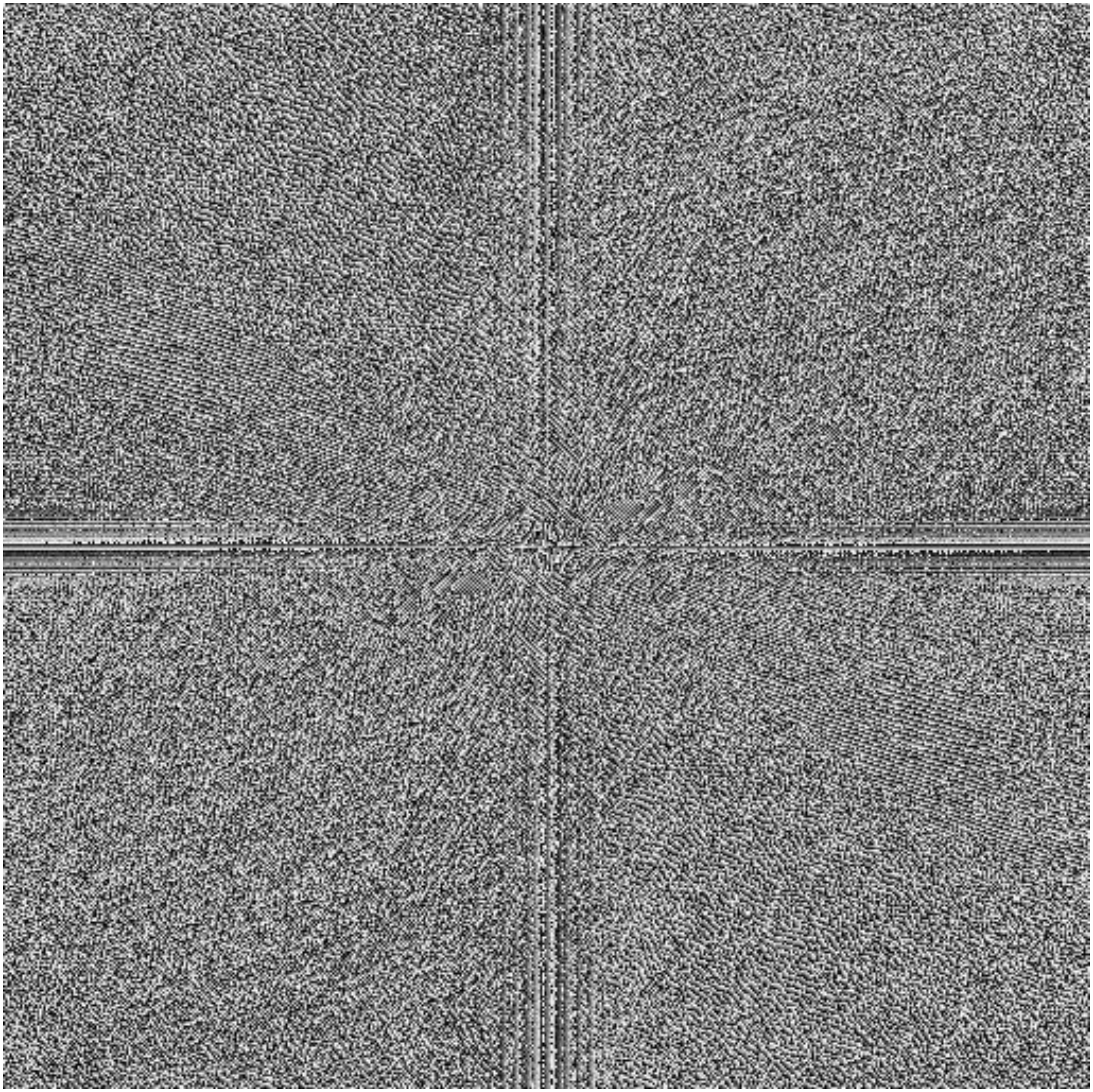
This is the phase transform of the cheetah pic



This is the magnitude transform of the zebra pic



This is the phase transform of the zebra pic



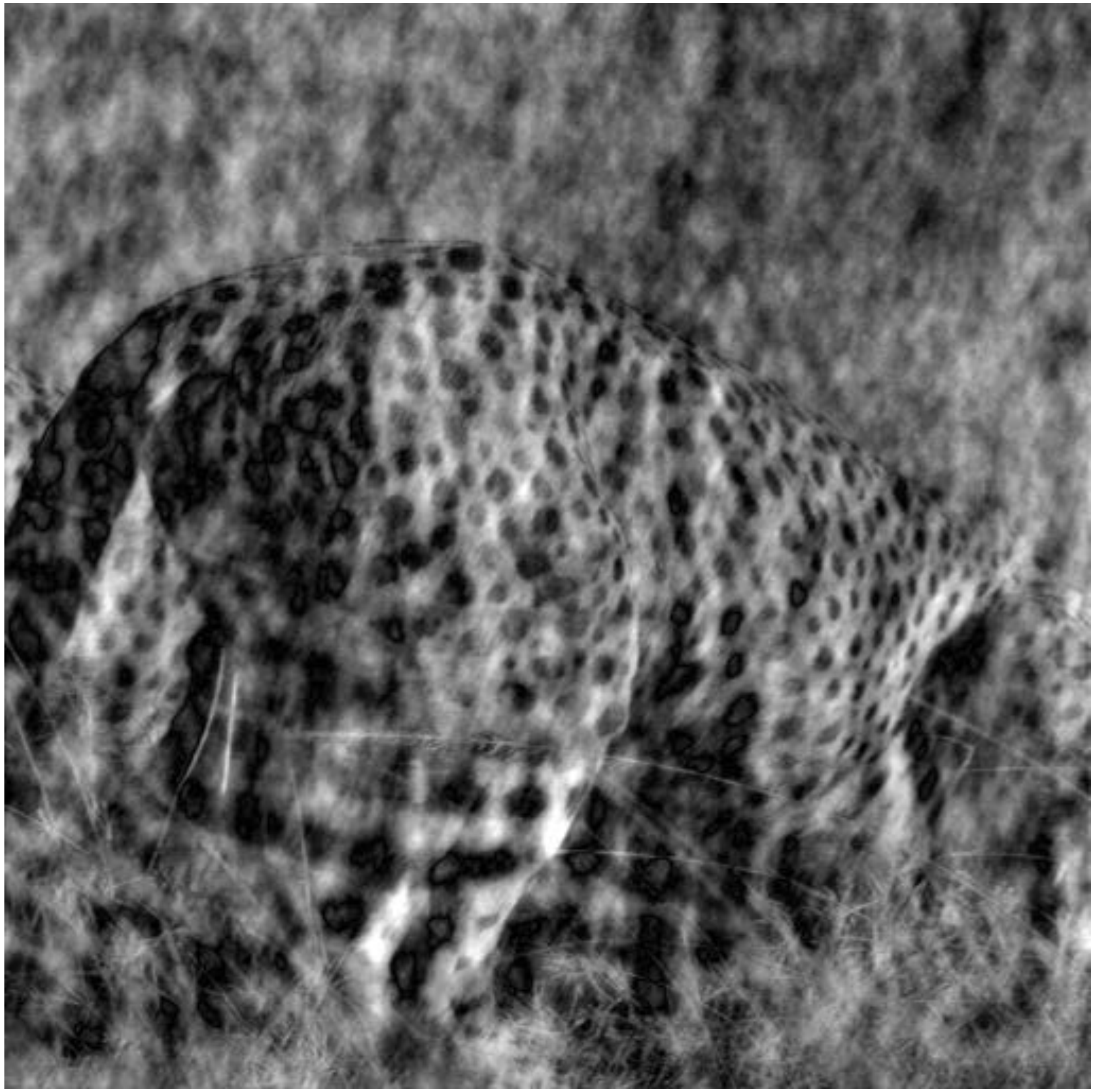
Phase and Magnitude

Demonstration

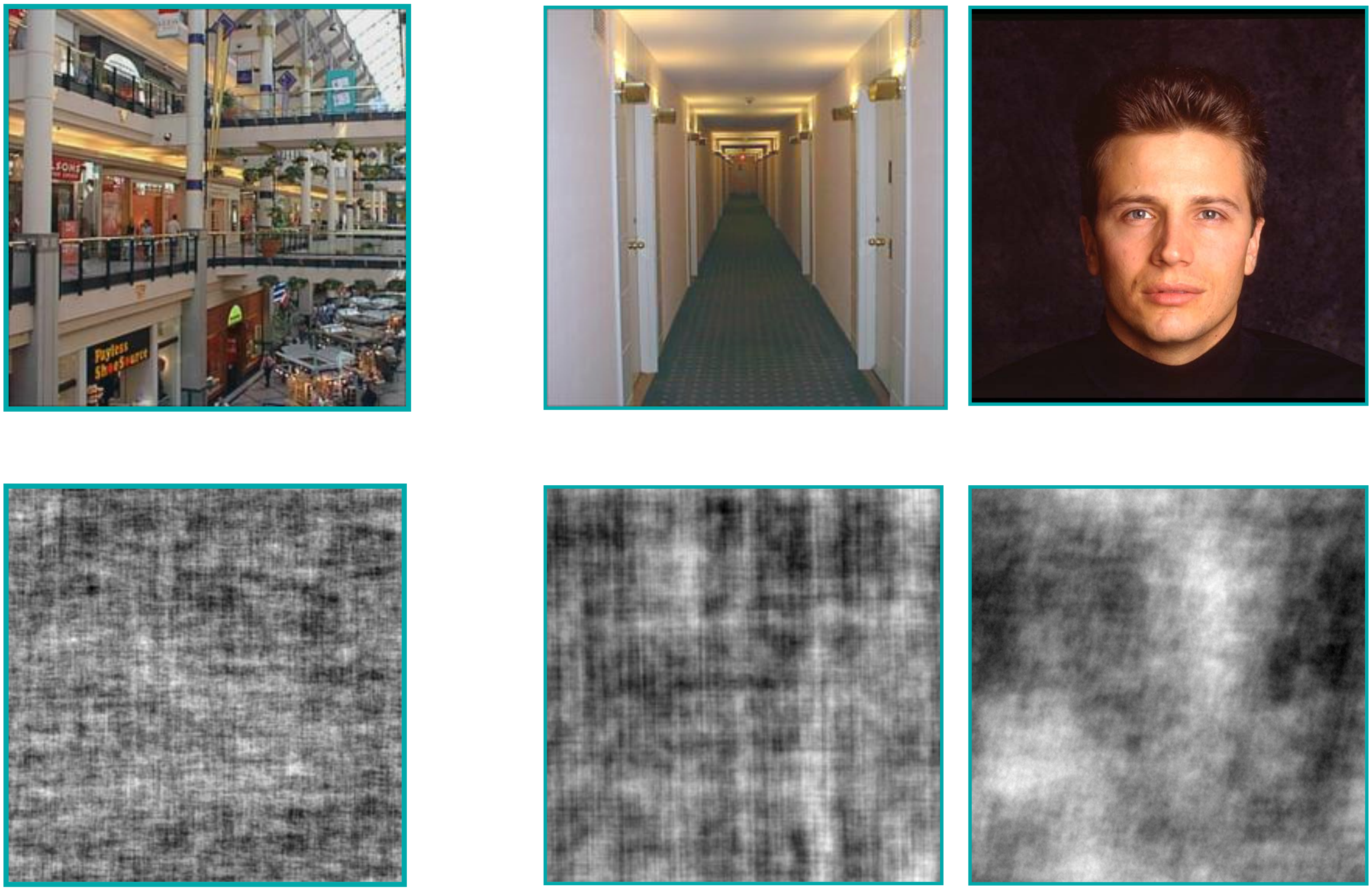
- Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

Reconstruction with zebra phase, cheetah magnitude

Reconstruction with cheetah phase, zebra magnitude

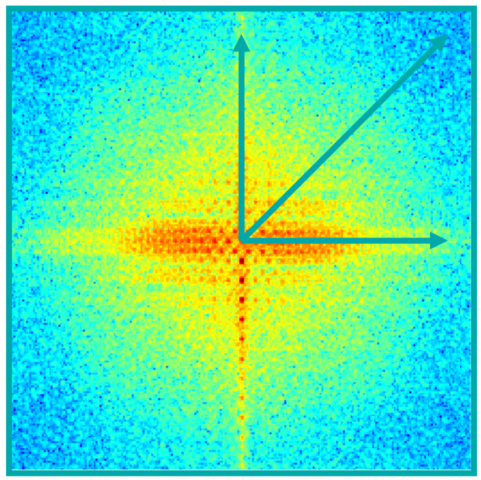


Randomizing the phase



How to interpret a Fourier Spectrum

Vertical orientation



45 deg.

Horizontal orientation

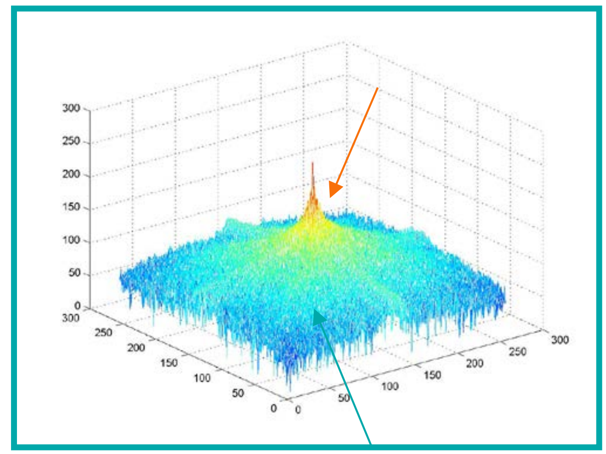
0

0

f_{max}

fx in cycles/image

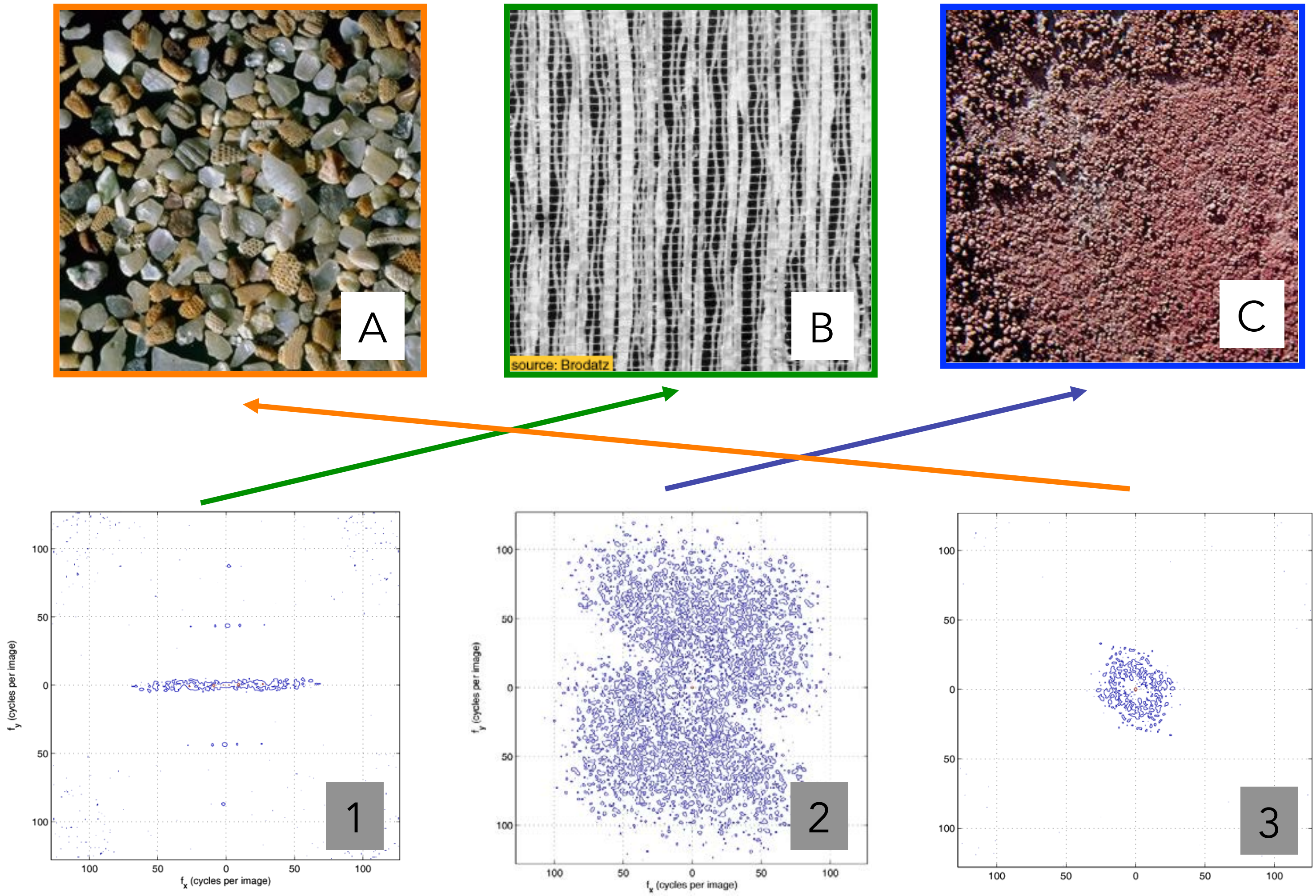
Low spatial frequencies



High spatial frequencies

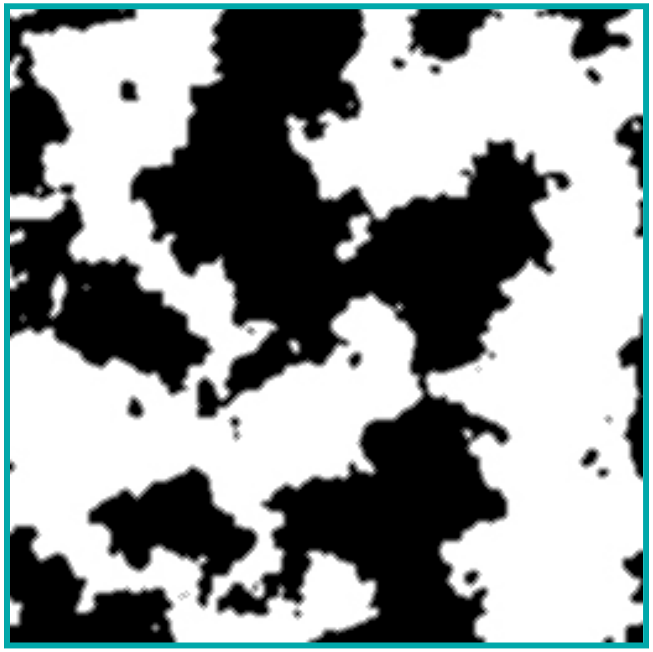
Log power spectrum

Which Fourier for which image?

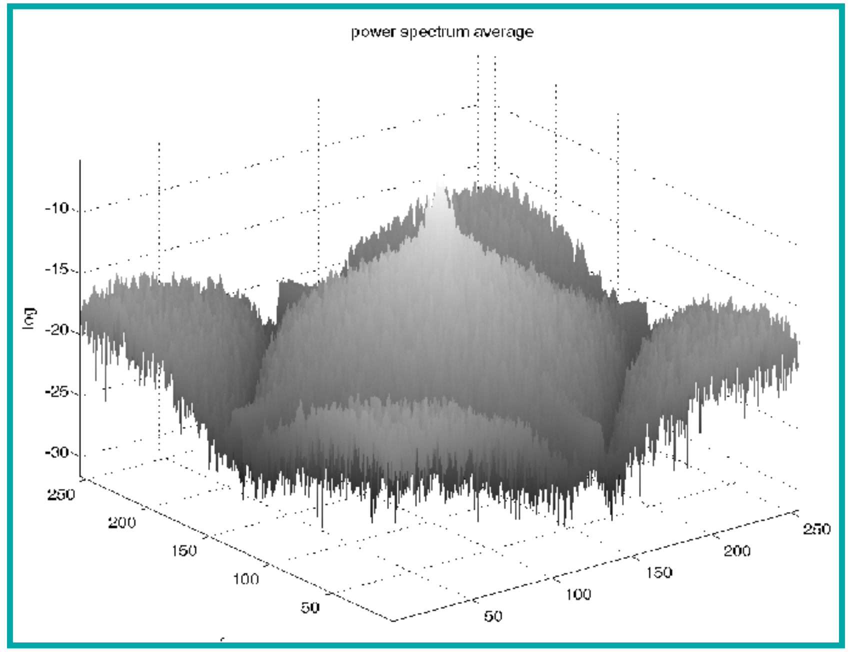


fx(cycles/image pixel size) fx(cycles/image pixel size) fx(cycles/image pixel size)

Some bizarre things in nature …

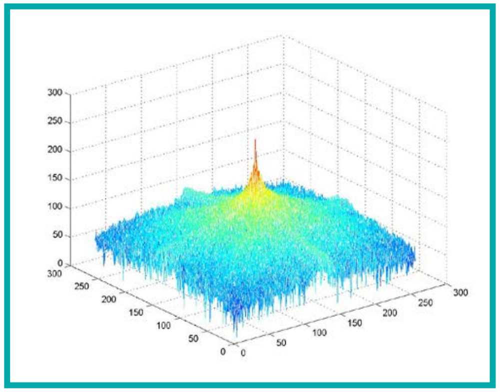


Cow skin



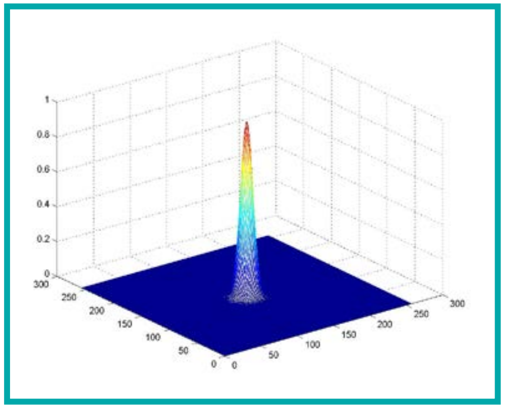
Use of Fourier Spectrum : Filtering

Fourier space



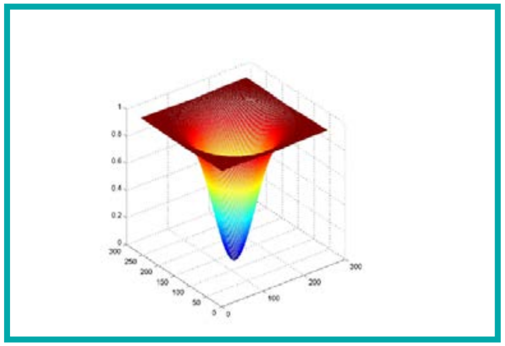
*

Low pass Filter

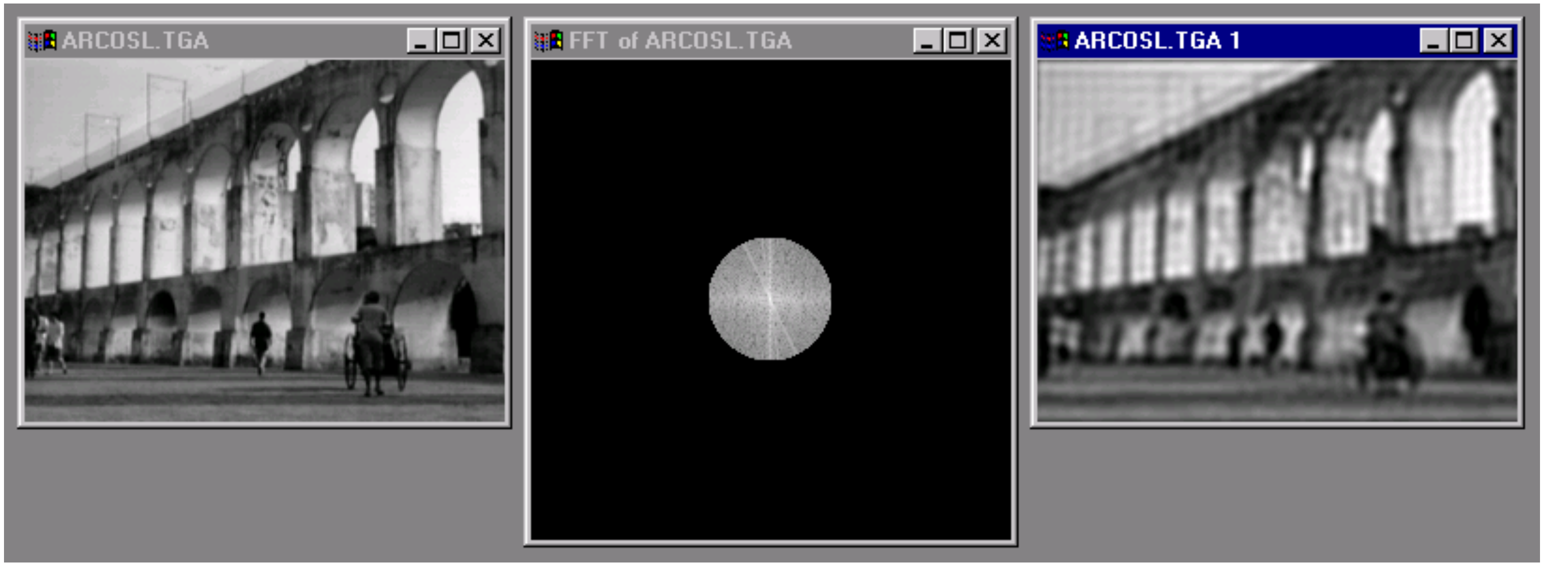


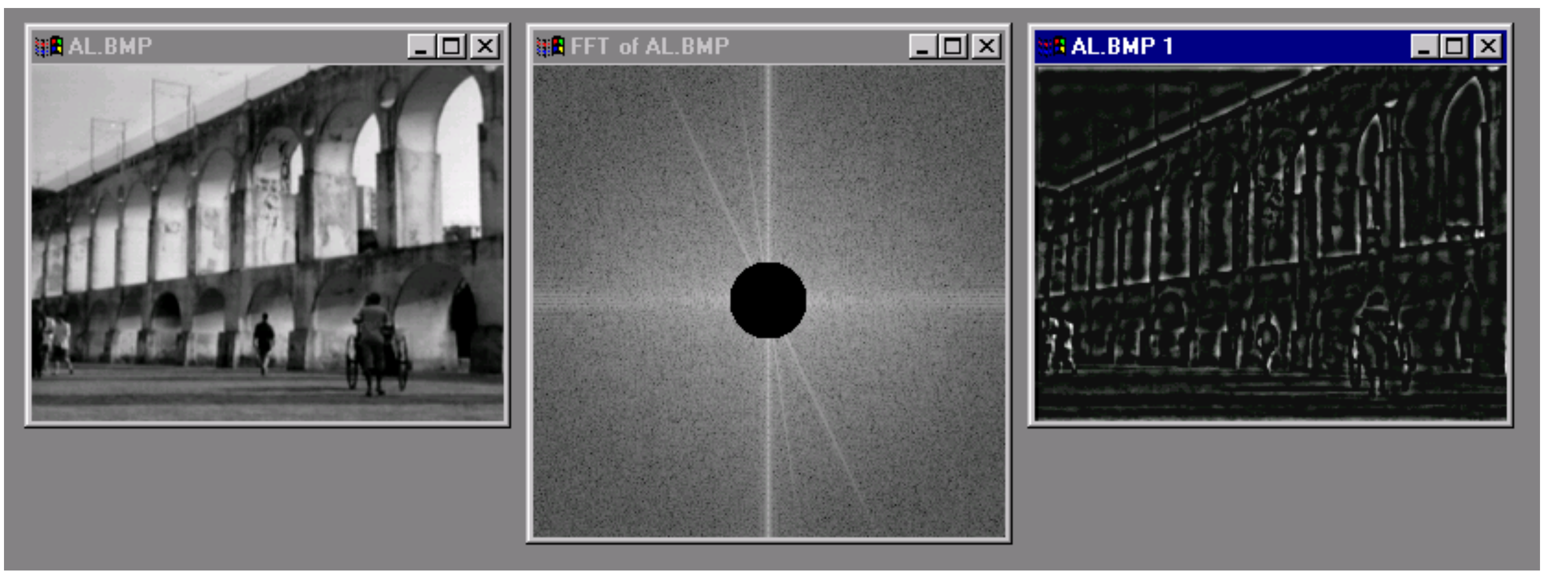
*

High pass filter



Low and High Pass filtering





The Convolution Theorem

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

$$\mathrm{F}[g * h] = \mathrm{F}[g]\mathrm{F}[h]$$

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

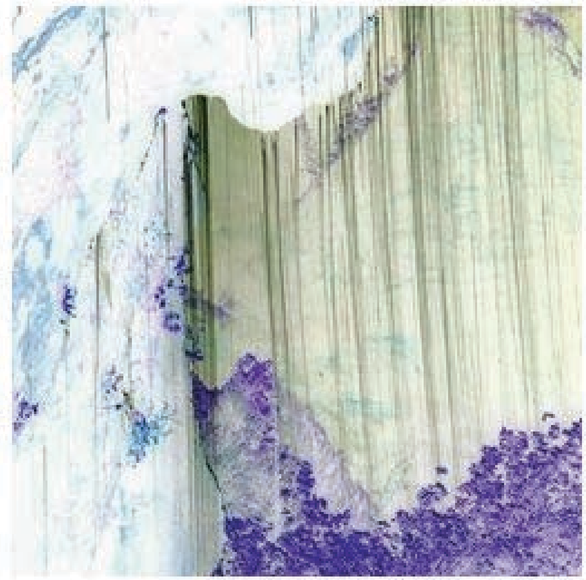
$$\mathrm{F}^{-1}[gh] = \mathrm{F}^{-1}[g] * \mathrm{F}^{-1}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

Places2: Demo

Predicted scene categories:

barndoor (0.398), waterfall - block (0.142), waterfall - plunge (0.125), bamboo forest (0.07), waterfall - fan (0.056)



Predicted scene categories:

barndoor (0.25), ice shelf (0.097), childs room (0.074), clothing store (0.061), bow window - indoor (0.058)

Additional Slides

Principles of Spatial Convolution

- The linear operation consists in multiplying each pixel in the neighborhood by a corresponding coefficient and summing the results to obtain a response at each point (x,y)
- If the neighborhood is a size (m,n), nm coefficients are required
- The *coefficients* are arranged as a matrix called *filter*, mask, filter mask, kernel, template
- The figure illustrates the mechanics of linear spatial filtering: it consists in moving the center of the filter mask, *w*, from point to point in an image *f*.

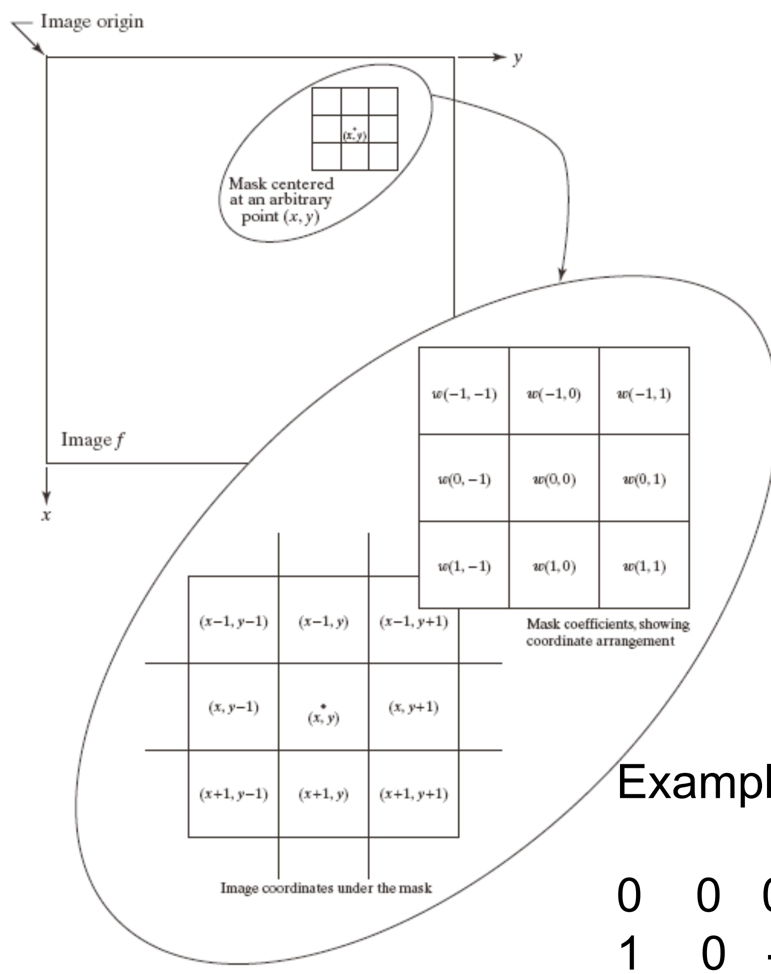


FIGURE 3.13 The mechanics of linear spatial filtering. The magnified drawing shows a 3×3 filter mask and the corresponding image neighborhood directly under it. The image neighborhood is shown displaced out from under the mask for ease of readability.

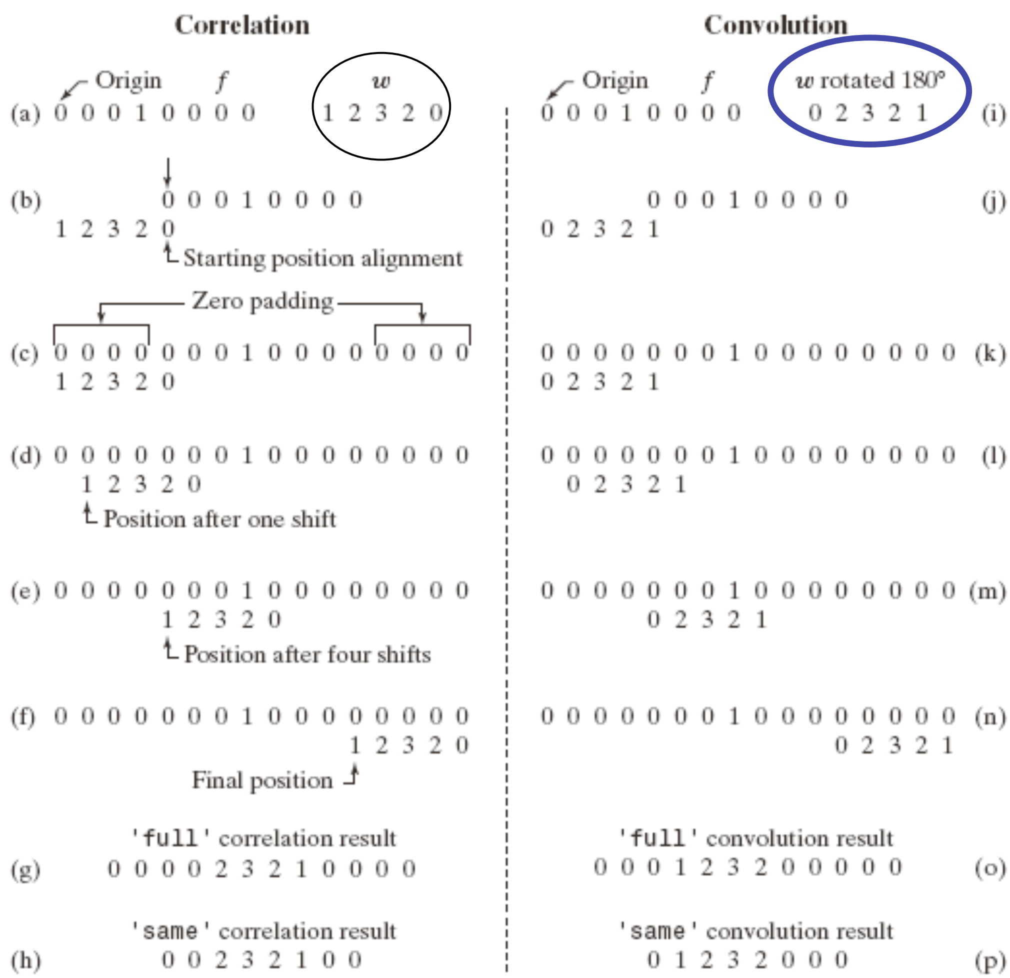
Example mask

0	0	0
1	0	-1
0	0	0

Convolution is correlation with a rotated filter mask

See the pdf on stellar Explaining_Convolution.pdf

FIGURE 3.14 Illustration of one-dimensional correlation and convolution.



A 2 d correlation and convolution

See the pdf Explaining_Convolution.pdf

Origin of $f(x, y)$

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

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	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	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

`'full'` convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(g)