Advances in Computer Vision
Feb. 7, 2011

Bill Freeman and Antonio Torralba
Massachusetts Institute of Technology

Linear filtering.

Readings:

slide notes for lecture 2

Szeliski, sects. 3.2, 3.3 for linear filtering.

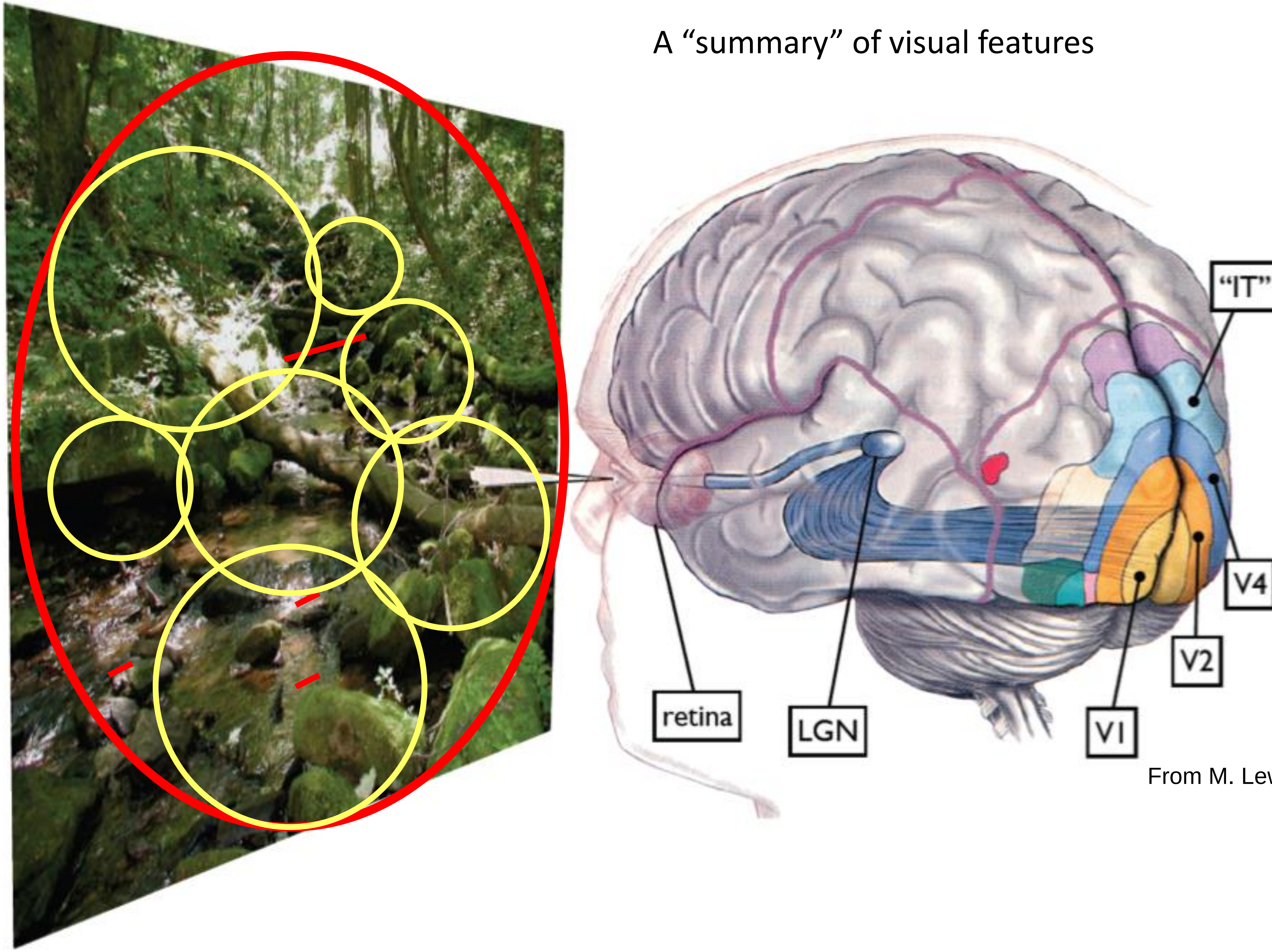
Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

Outline

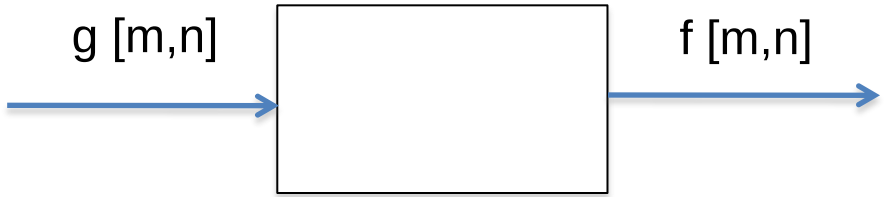
- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

A “summary” of visual features



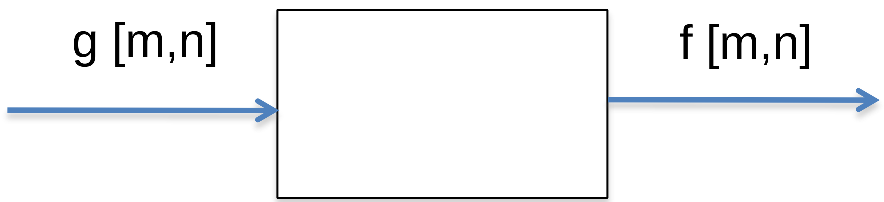
From M. Lewicky

Filtering



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

Linear filtering



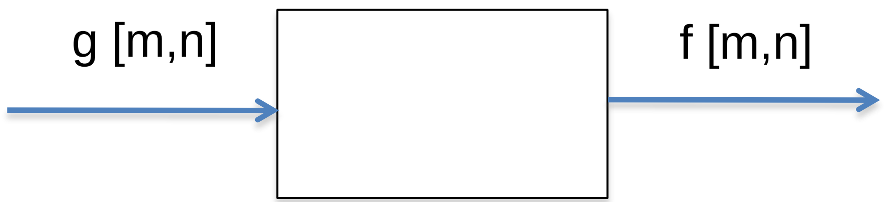
For a linear system, each output is a linear combination of all the input values:

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

In matrix form:

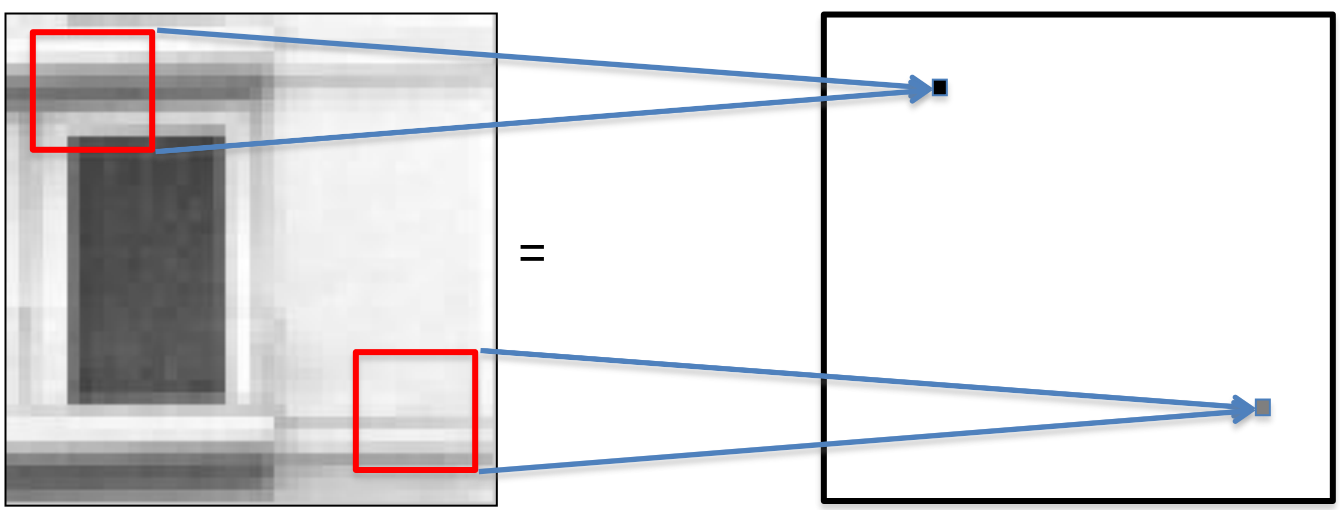
f = H g

Linear filtering



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

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

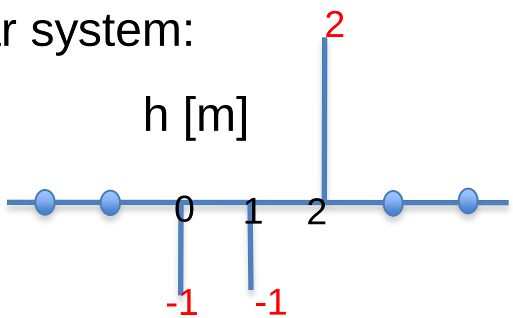


Linear filtering

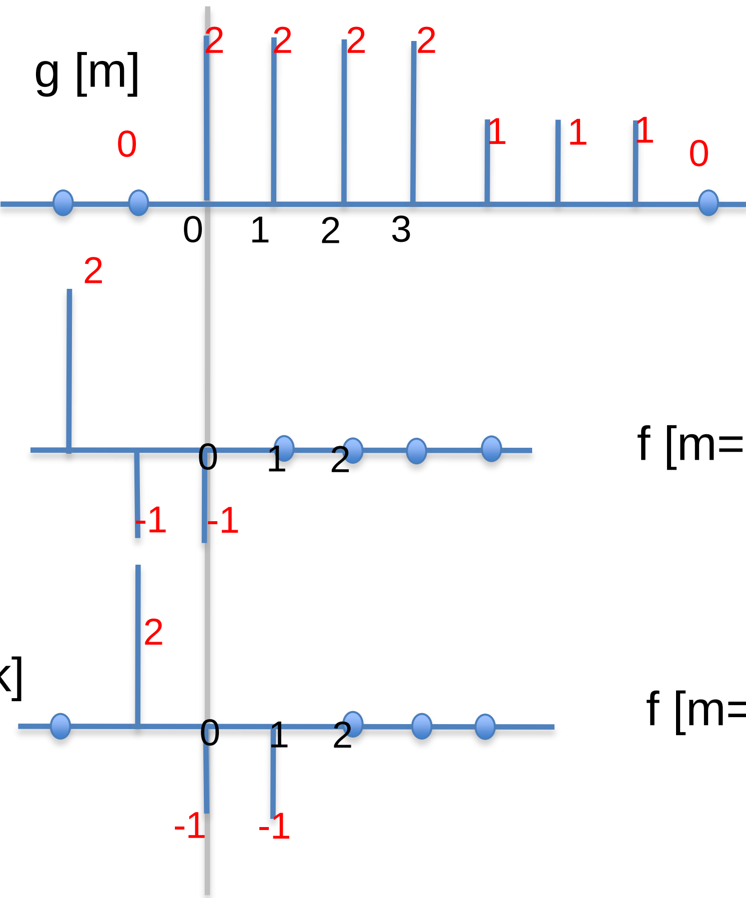
$$f[m,n] = h \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$

Linear system:

h [m]



Input:



h [-k]

h [1-k]

Output?

$$f[m=0] = \sum_k h[-k]\, g[k]$$

f [m=0]=-2

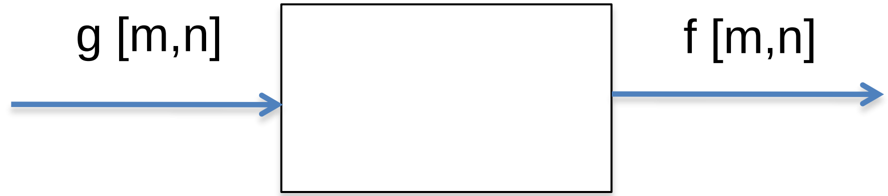
$$f[m=1] = \sum_k h[1-k]\, g[k]$$

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

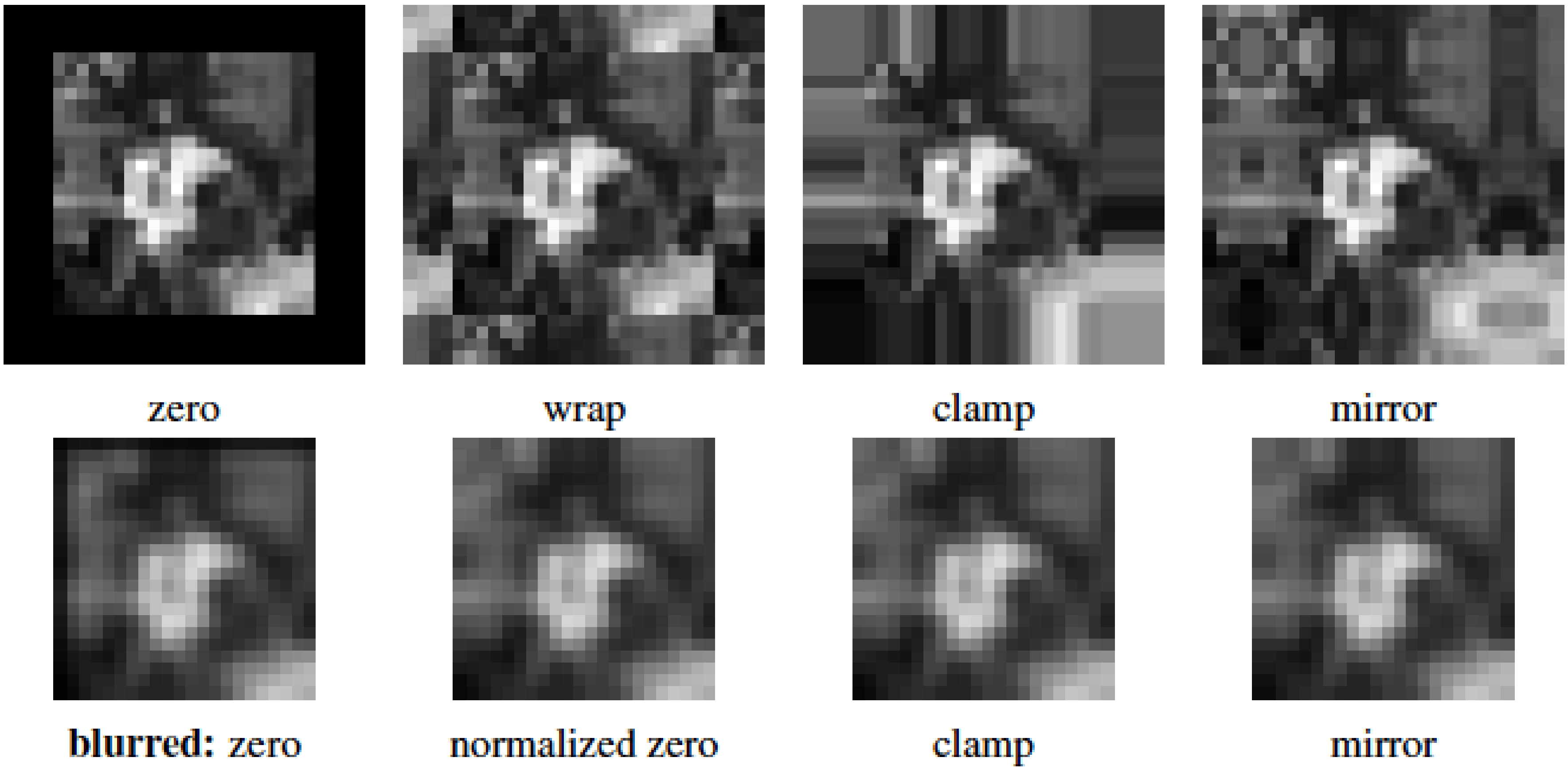
h[m,n]

=

?	?	?	?	?	?	?	?
?	-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



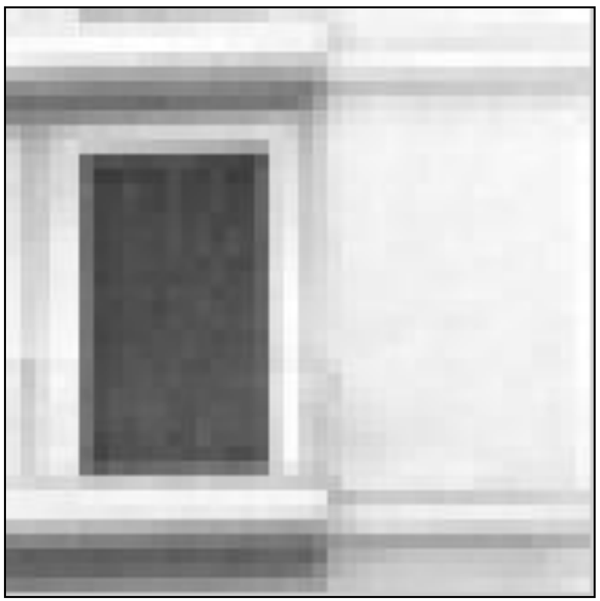
From Szeliski, Computer Vision, 2010

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…

Impulse

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

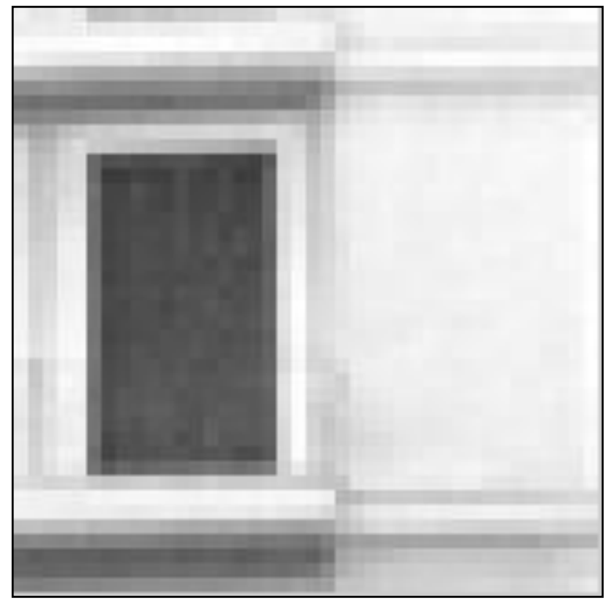


$\otimes$

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]

=

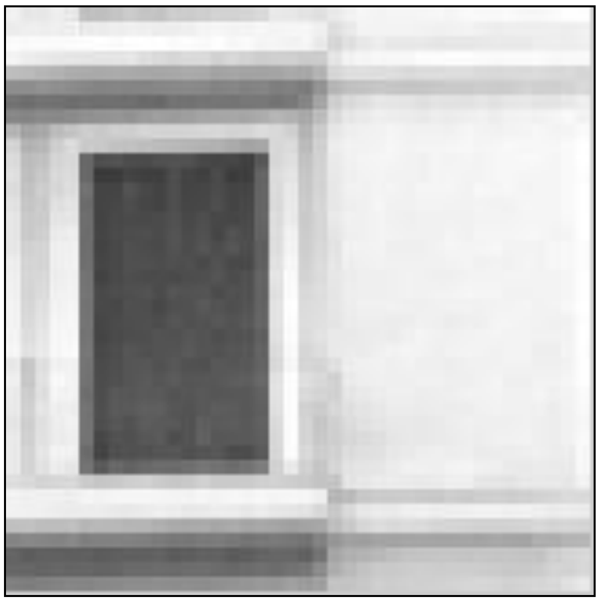


g[m,n]

f[m,n]

Shifts

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



g[m,n]

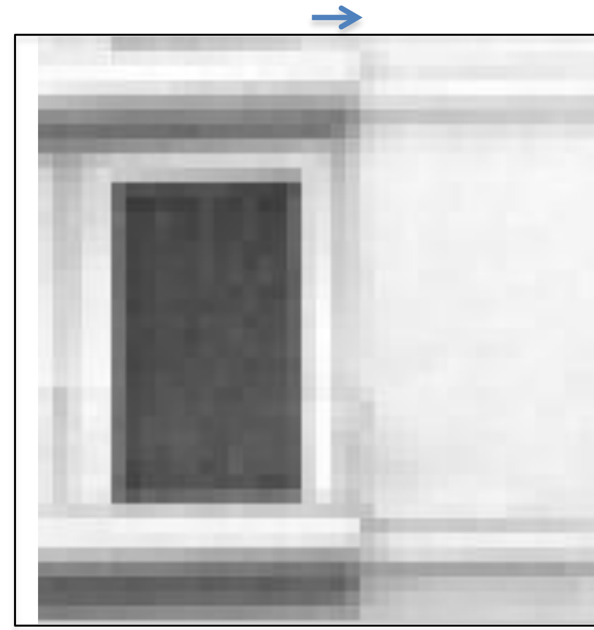
$\otimes$

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]

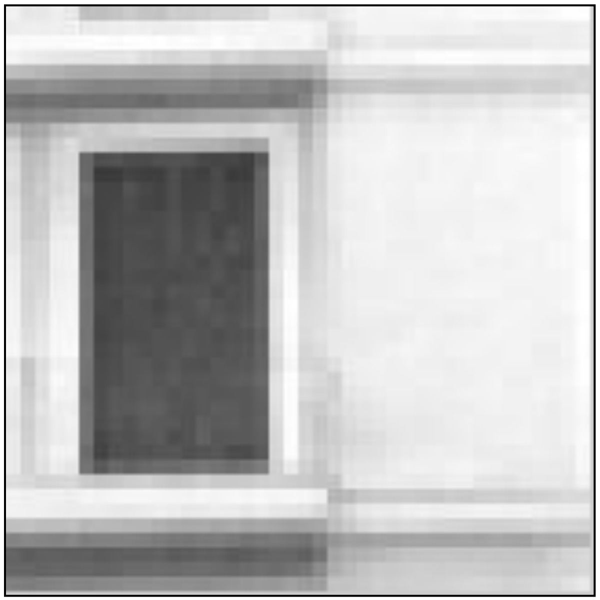
=

2pixels



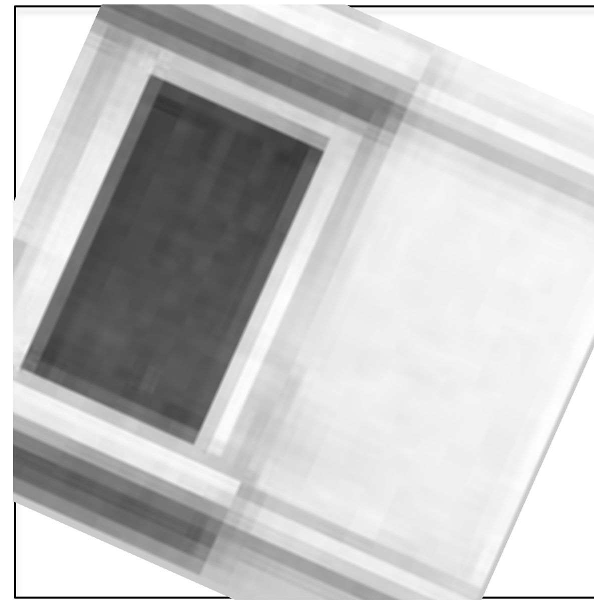
f[m,n]

Image rotation



$\otimes$? =

h[m,n]



f[m,n]

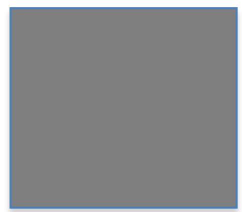
g[m,n]

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

Rectangular filter

g[m,n]

$\otimes$



h[m,n]

=

f[m,n]

Rectangular filter

g[m,n]

$\otimes$

h[m,n]

=

f[m,n]

Rectangular filter

$\otimes$

h[m,n]

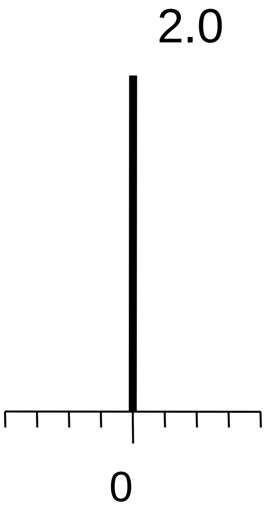
=

f[m,n]

g[m,n]

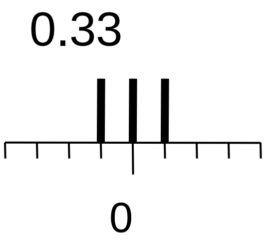
Sharpening

original

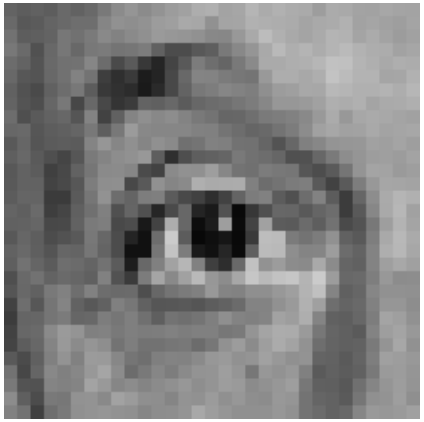


2 times (the sharp plus the blurred parts)

−



1 times the blurred parts



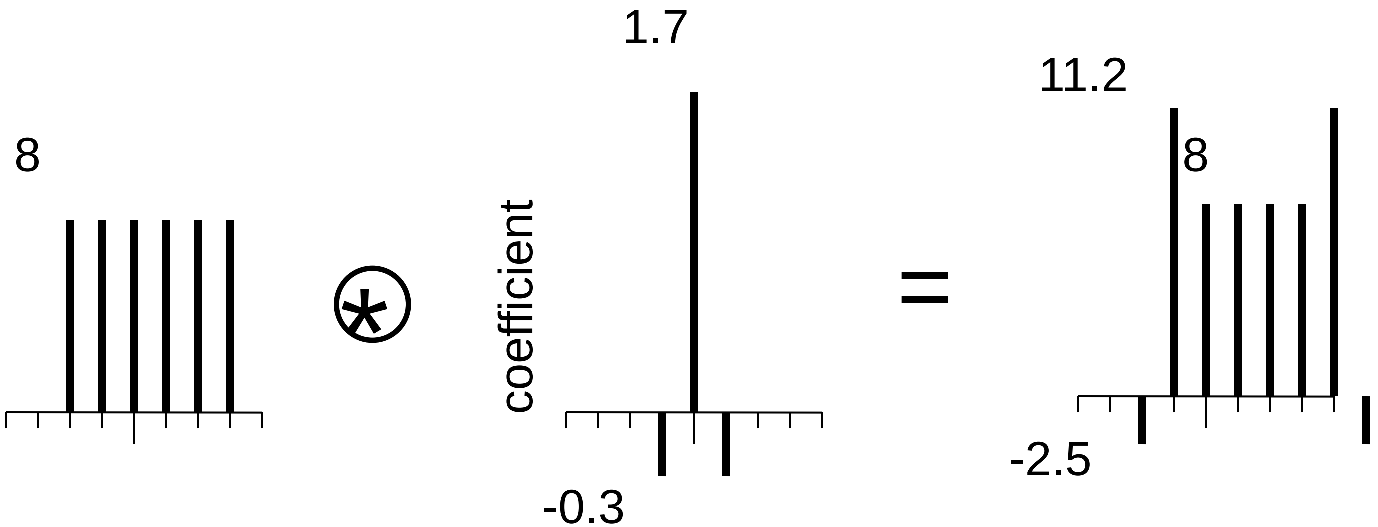
Sharpened original

a sharpened image--the blurred parts plus twice the sharp parts.

Sharpening example

filter

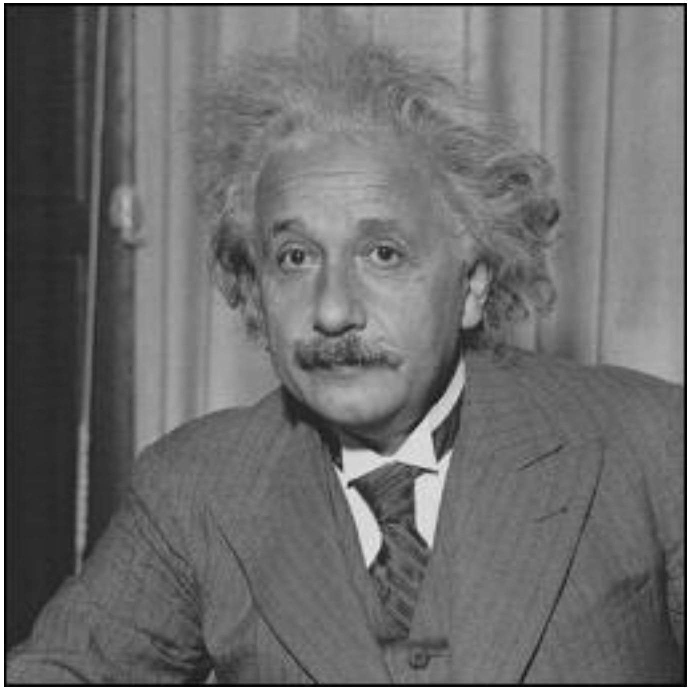
result



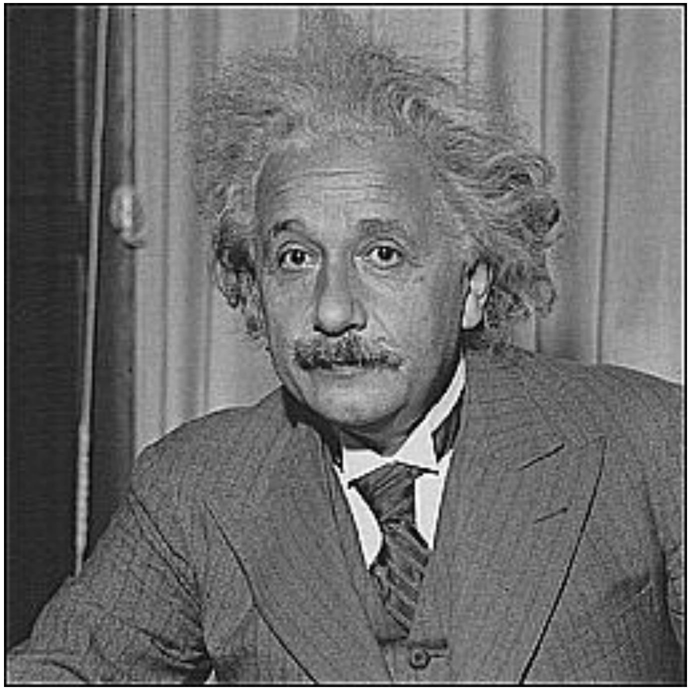
original

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

Sharpening



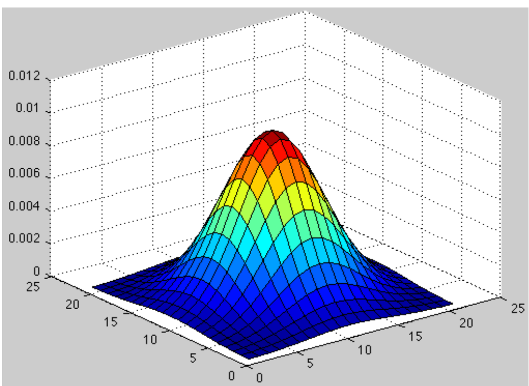
before

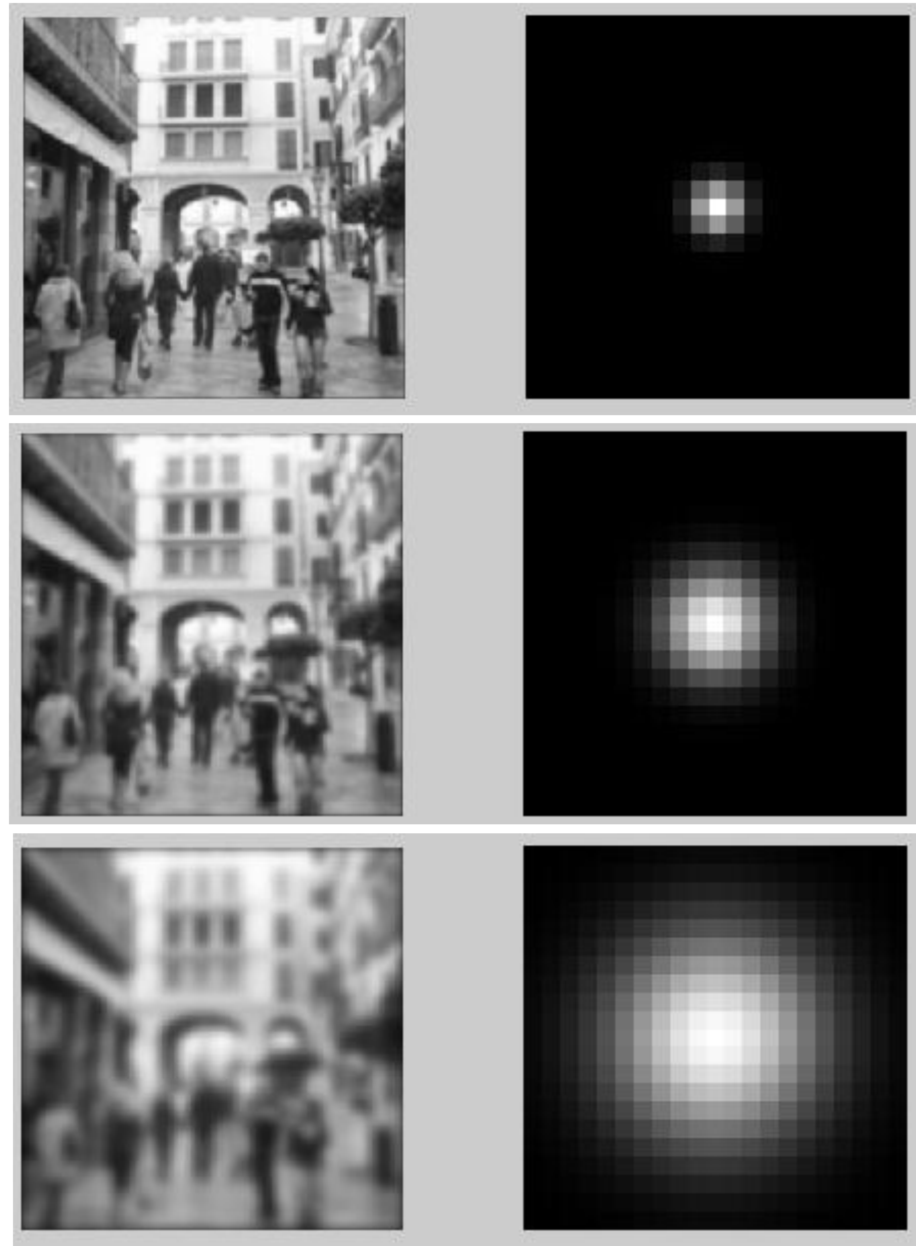


after

Gaussian filter

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



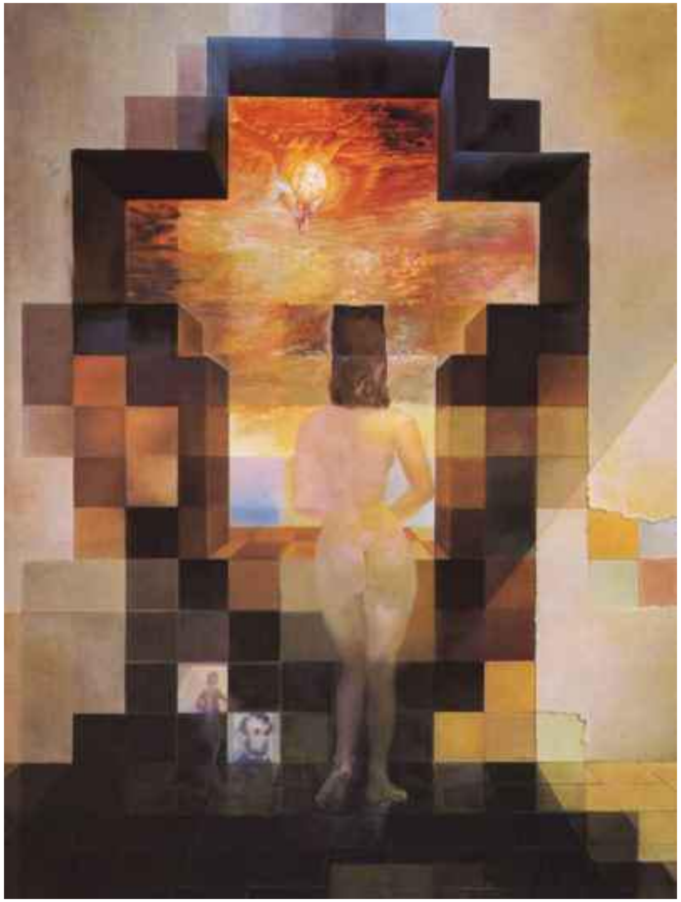


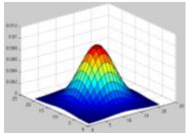
σ=1

σ=2

σ=4

Gaussian filtering allows analysis at different spatial scales


Dali



Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

The Discrete Fourier transform

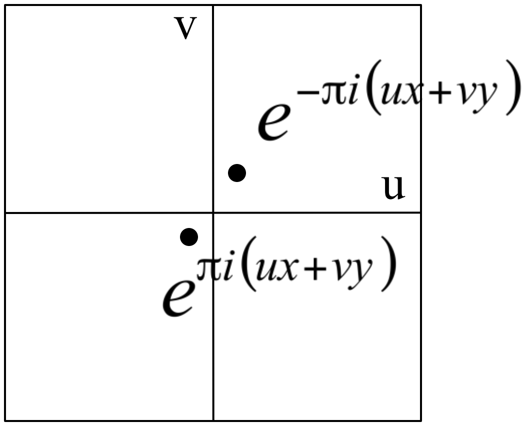
Forward transform

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

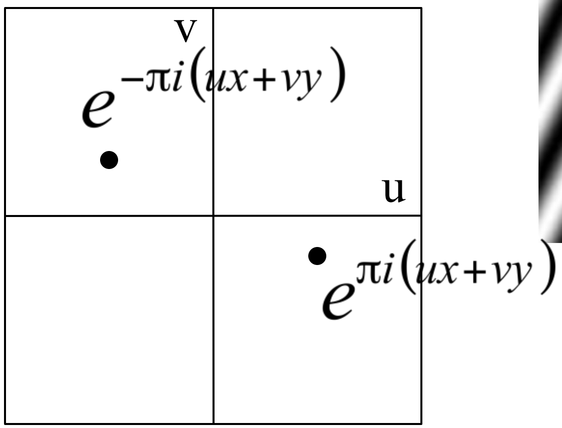
Inverse transform

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

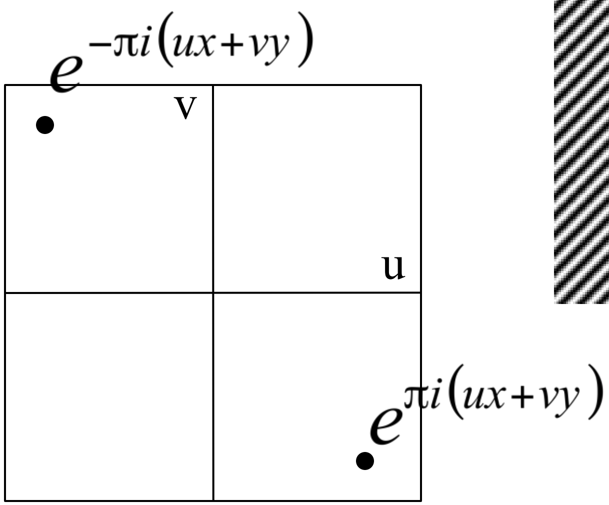
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation.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

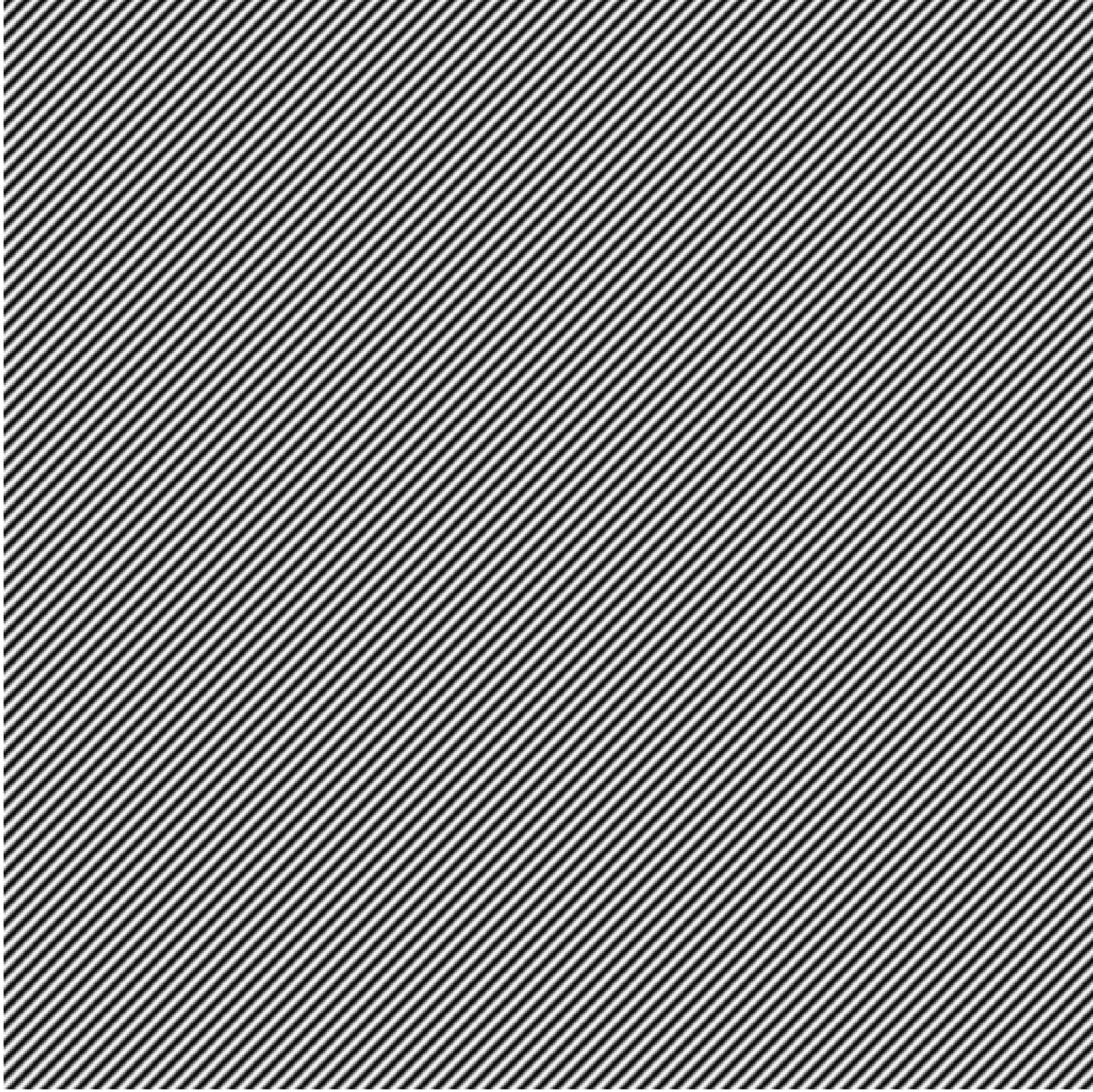


Here u and v are larger than in the previous slide.



And larger still...





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

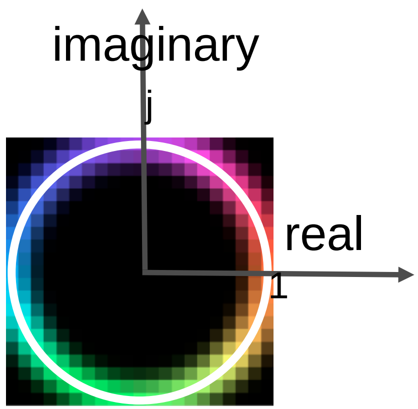
Same basis functions are used for the inverse transform

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

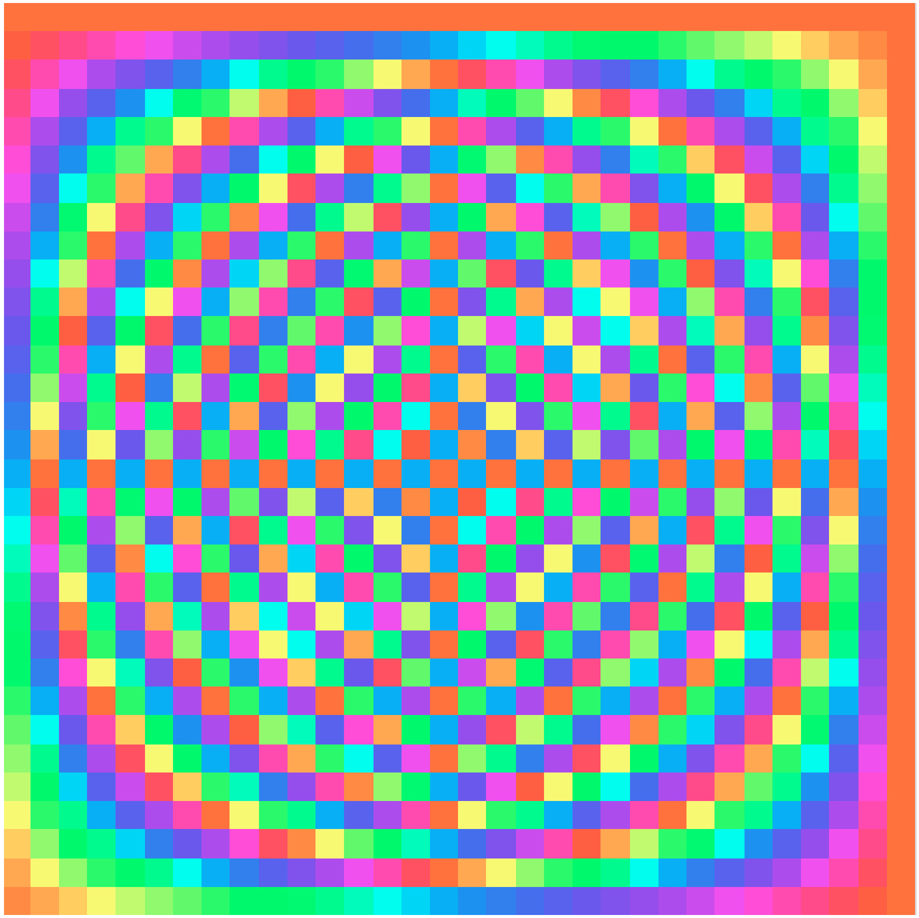
$$= U^{+}\vec{F}$$

U transpose and complex conjugate

Fourier transform visualization



color key



Fourier transform matrix

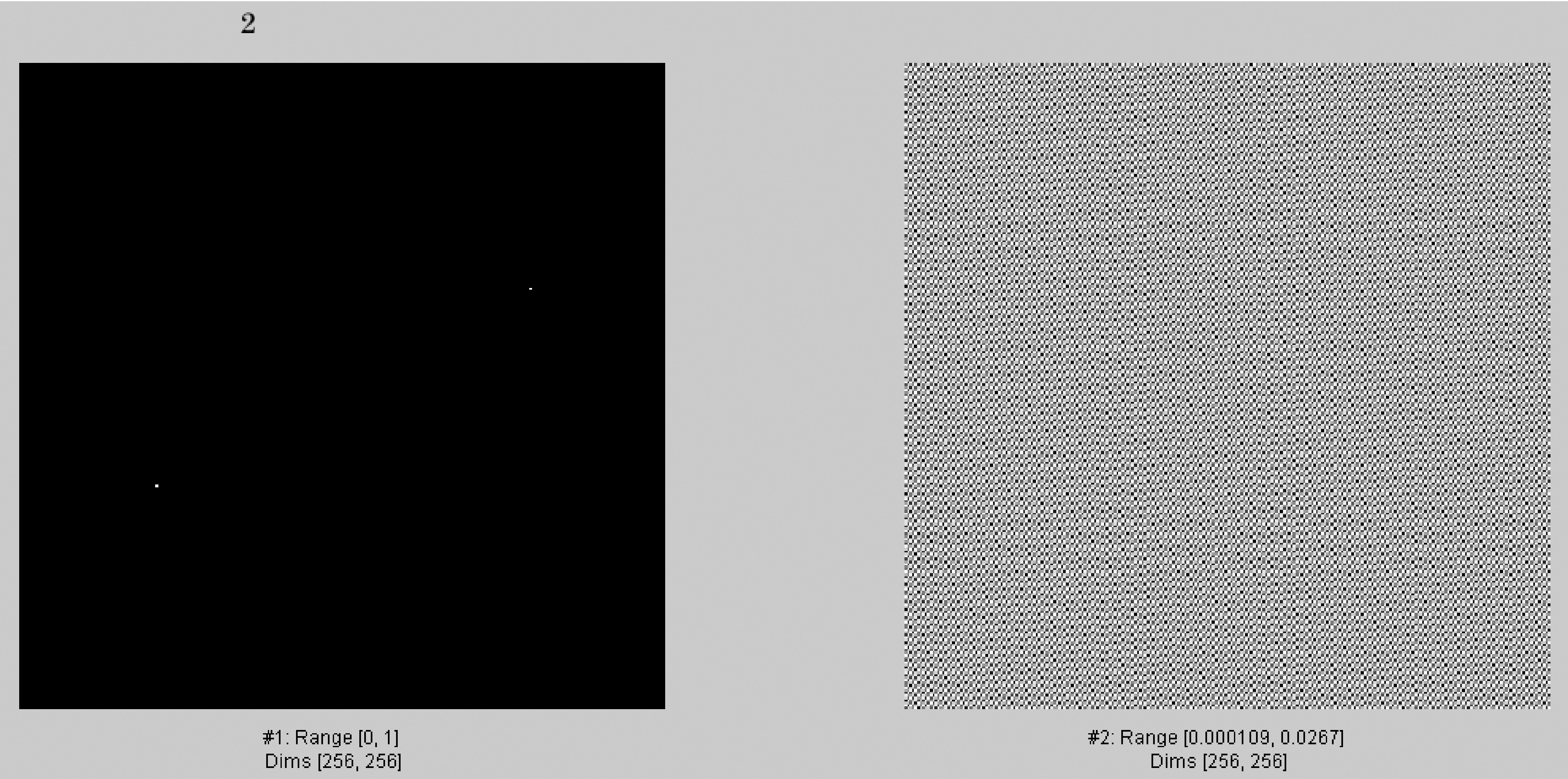
input signal

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

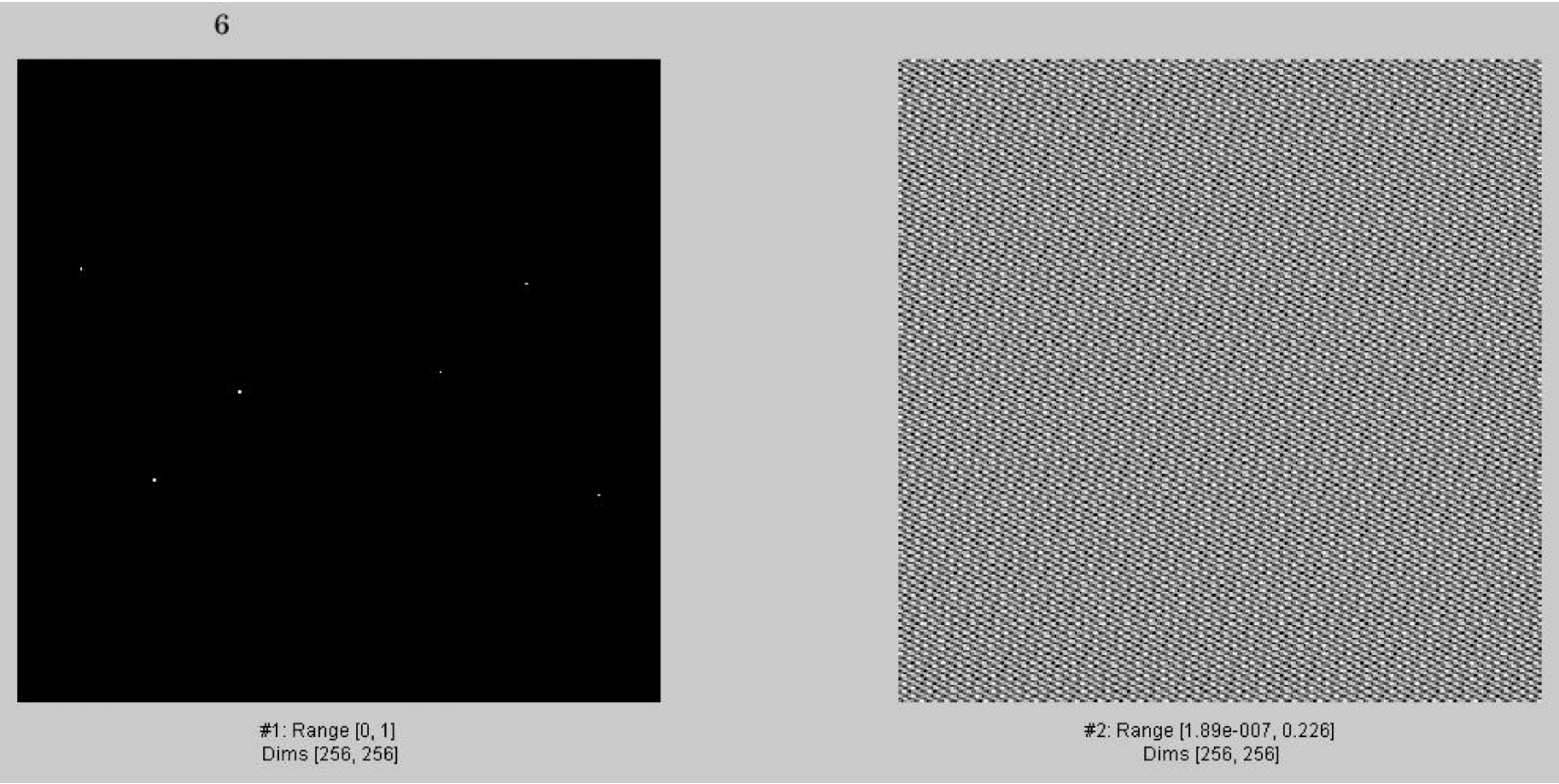
Example image synthesis with fourier basis.

- 16 images

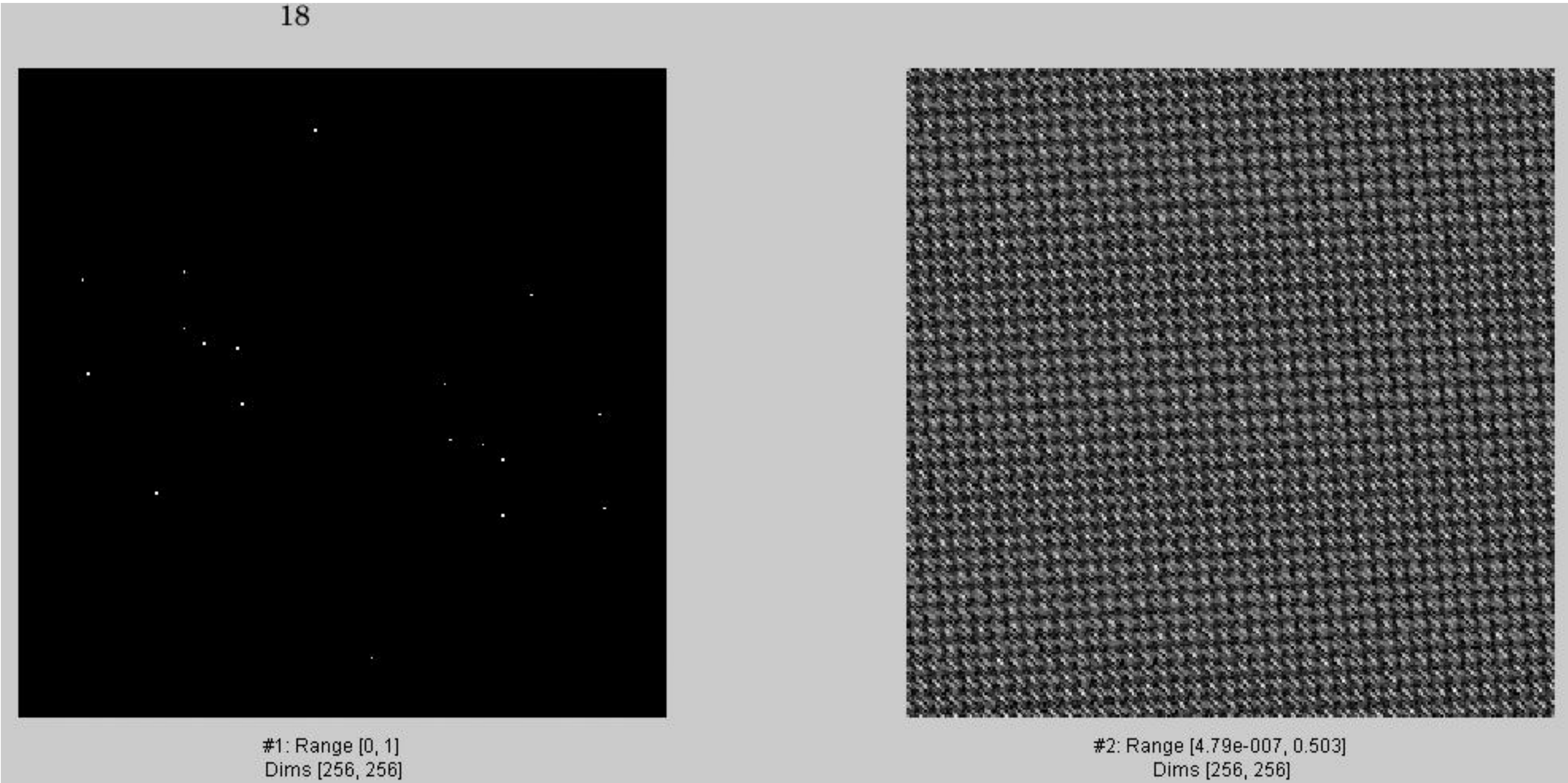
2



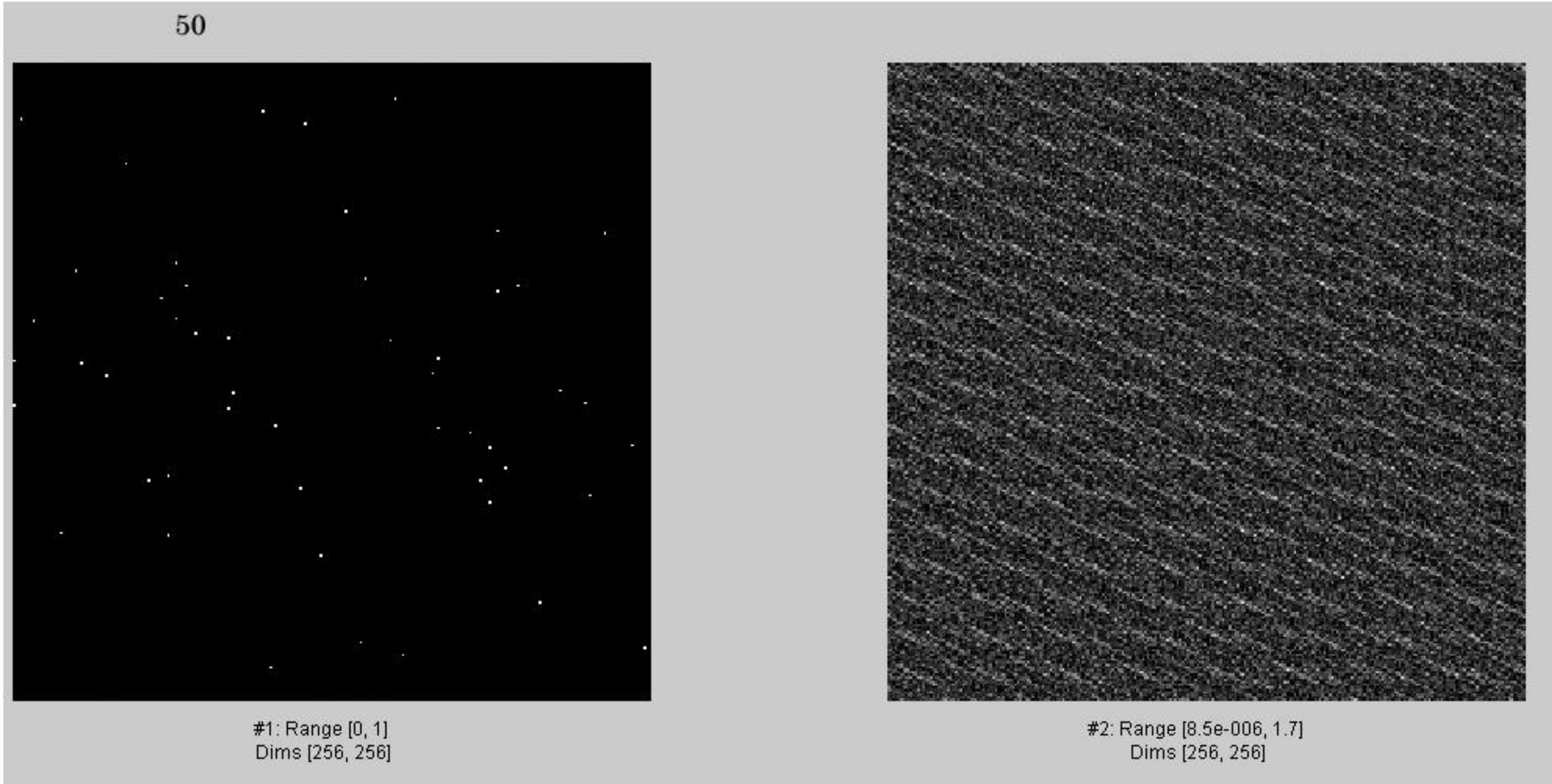
6



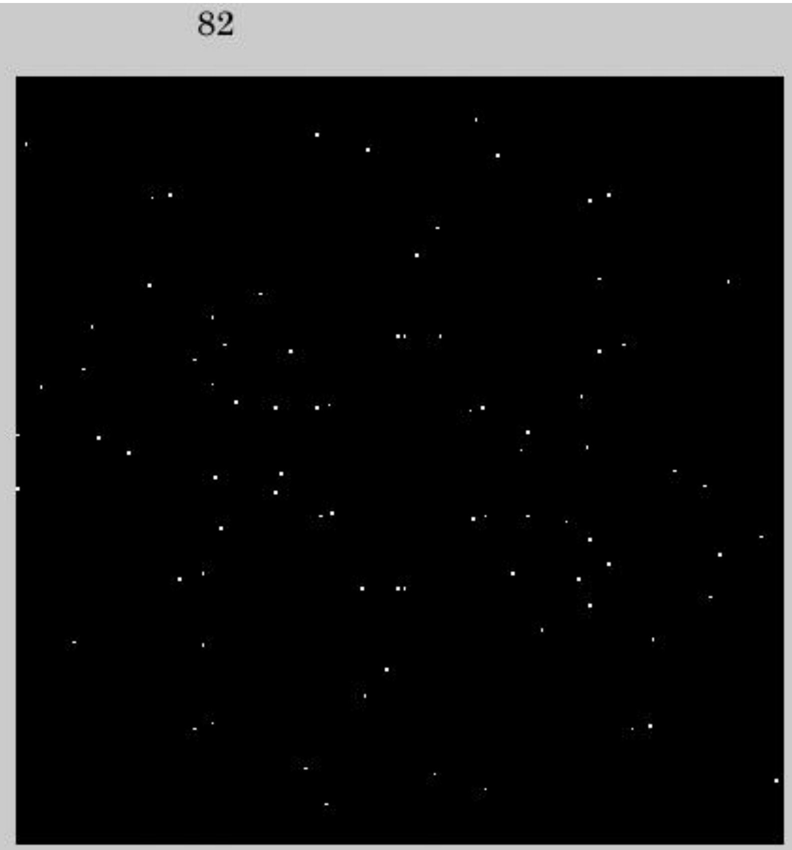
18



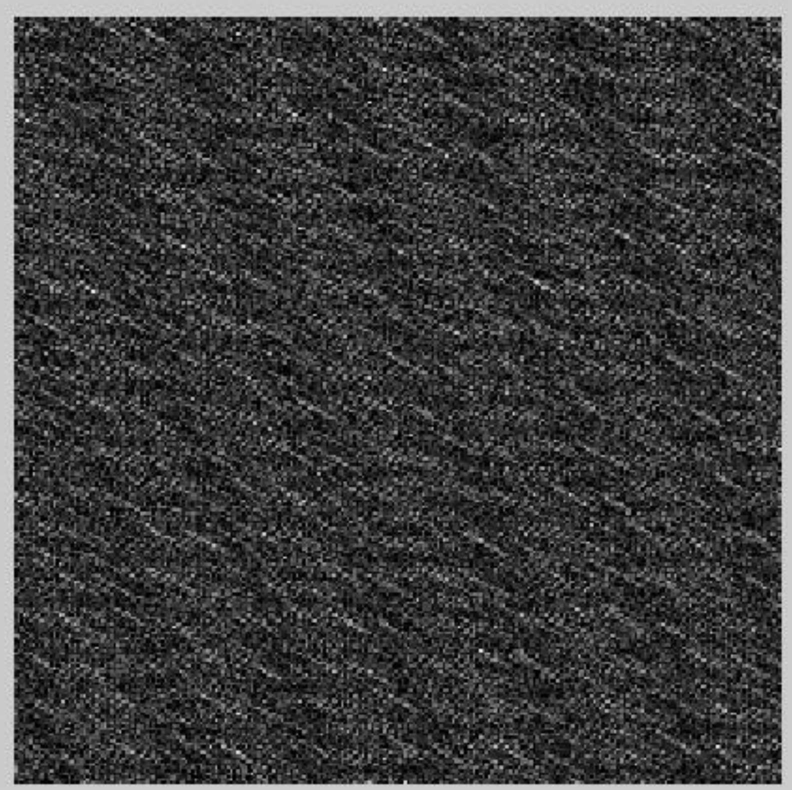
50



82

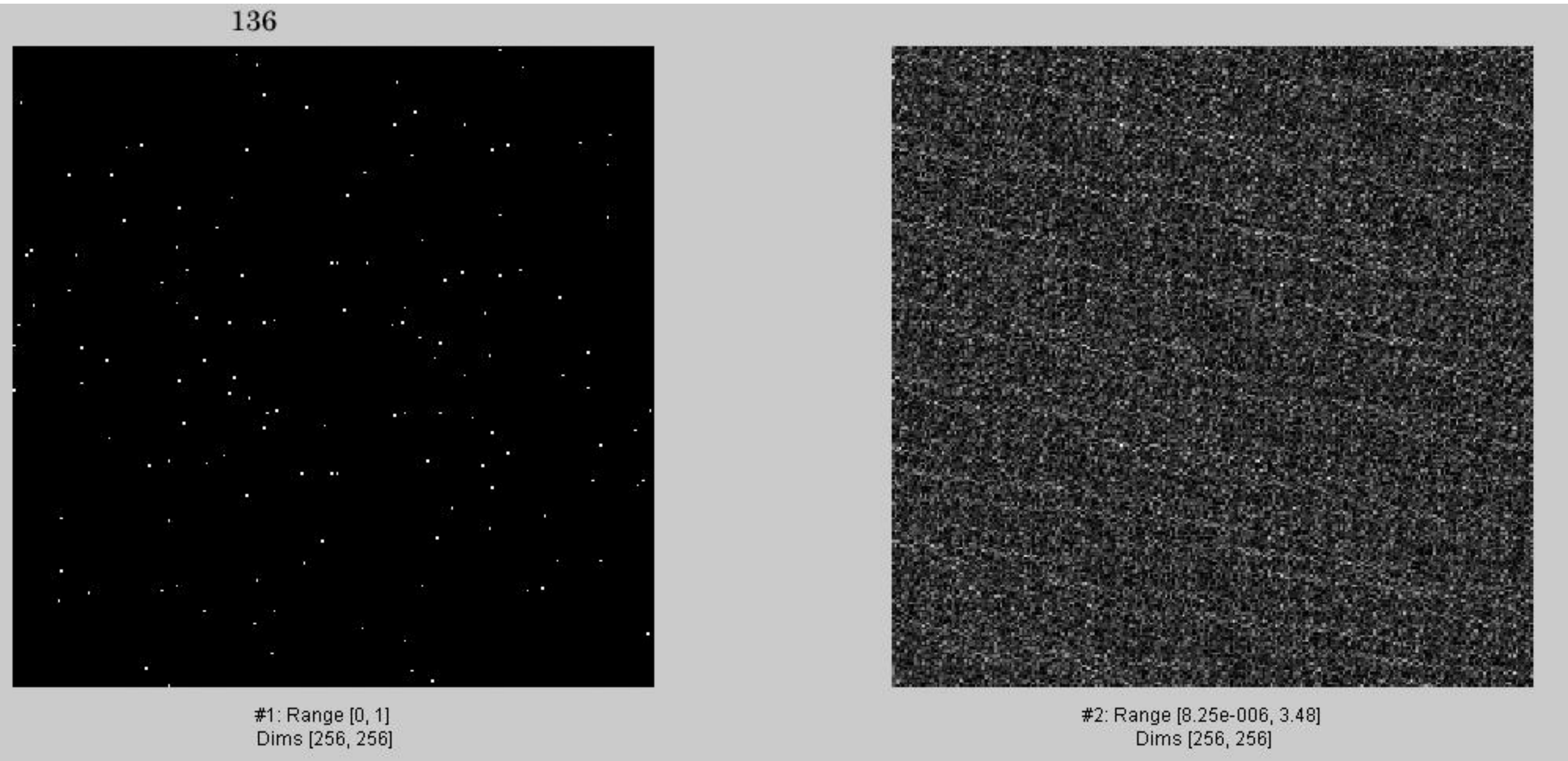


#1: Range [0, 1]
Dims [256, 256]

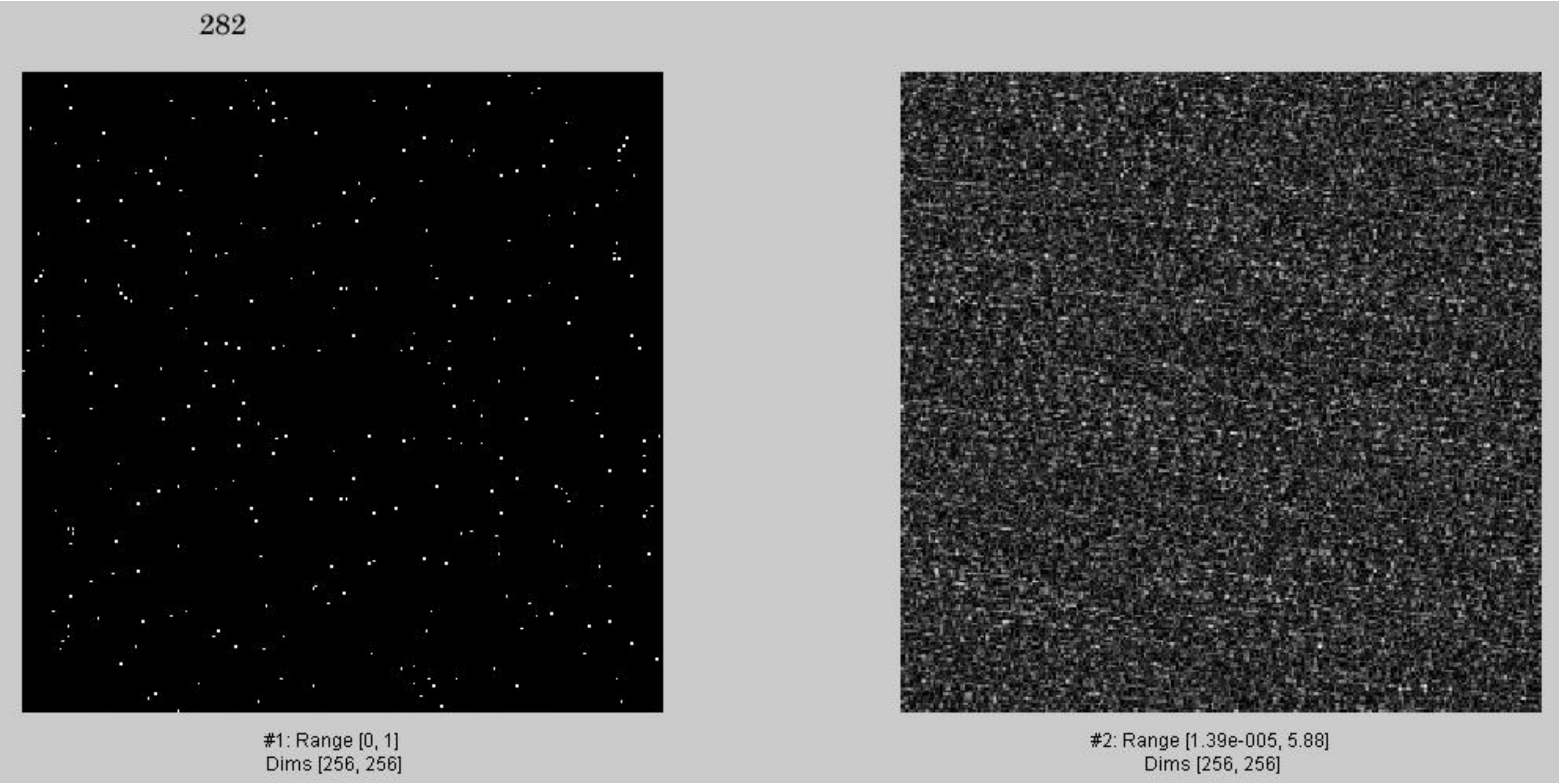


#2: Range [3.85e-007, 2.21]
Dims [256, 256]

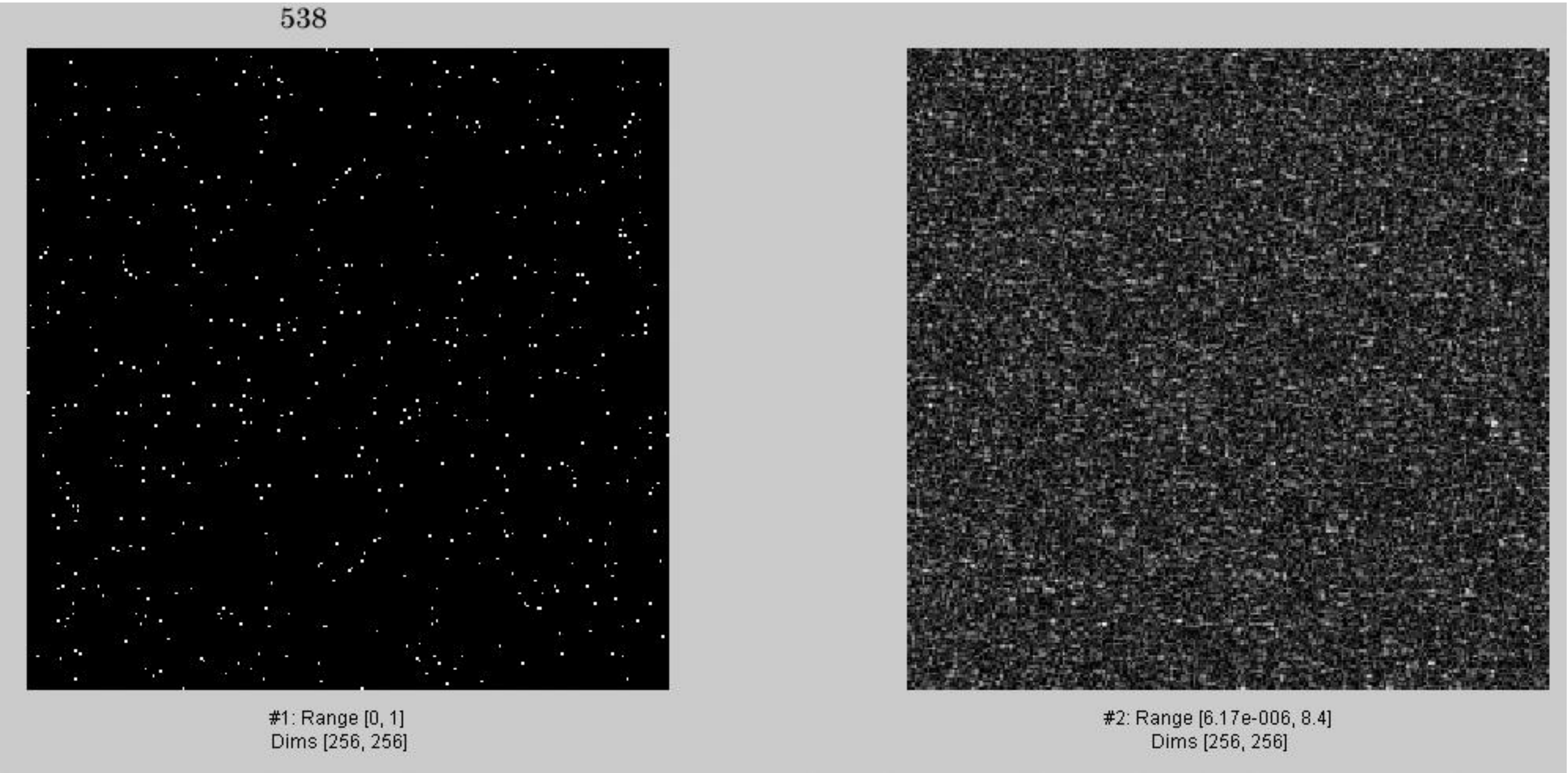
136



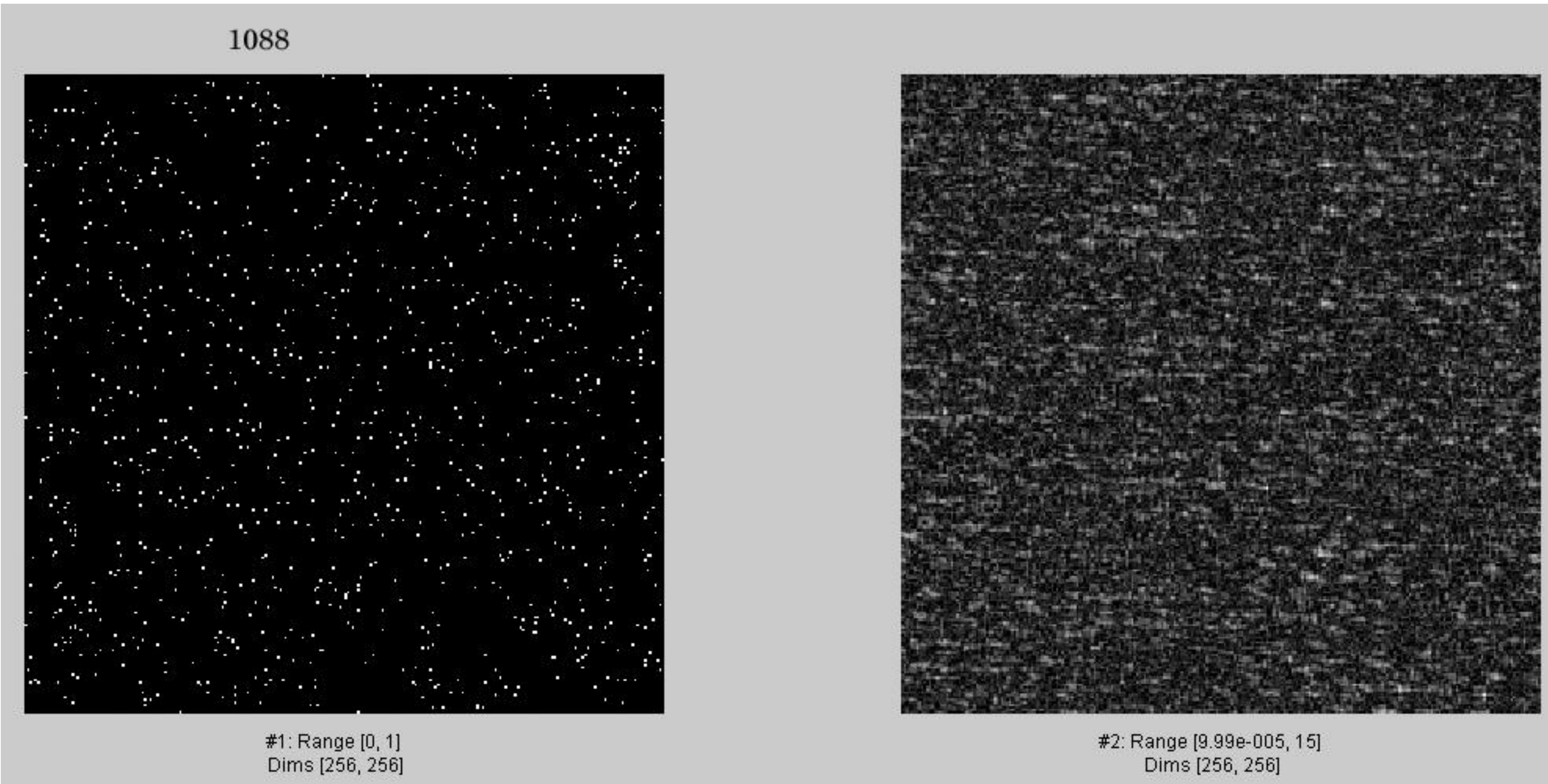
282



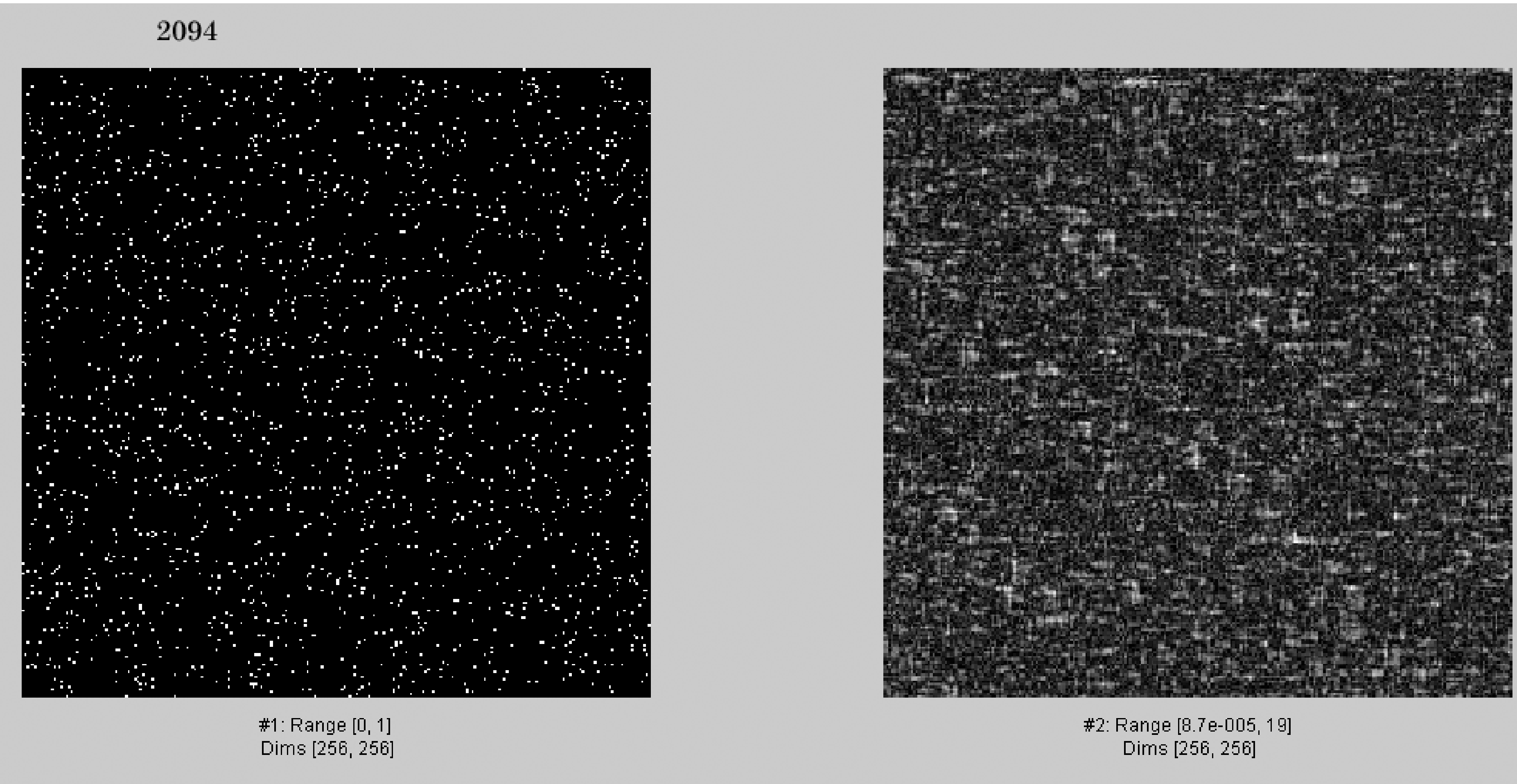
538



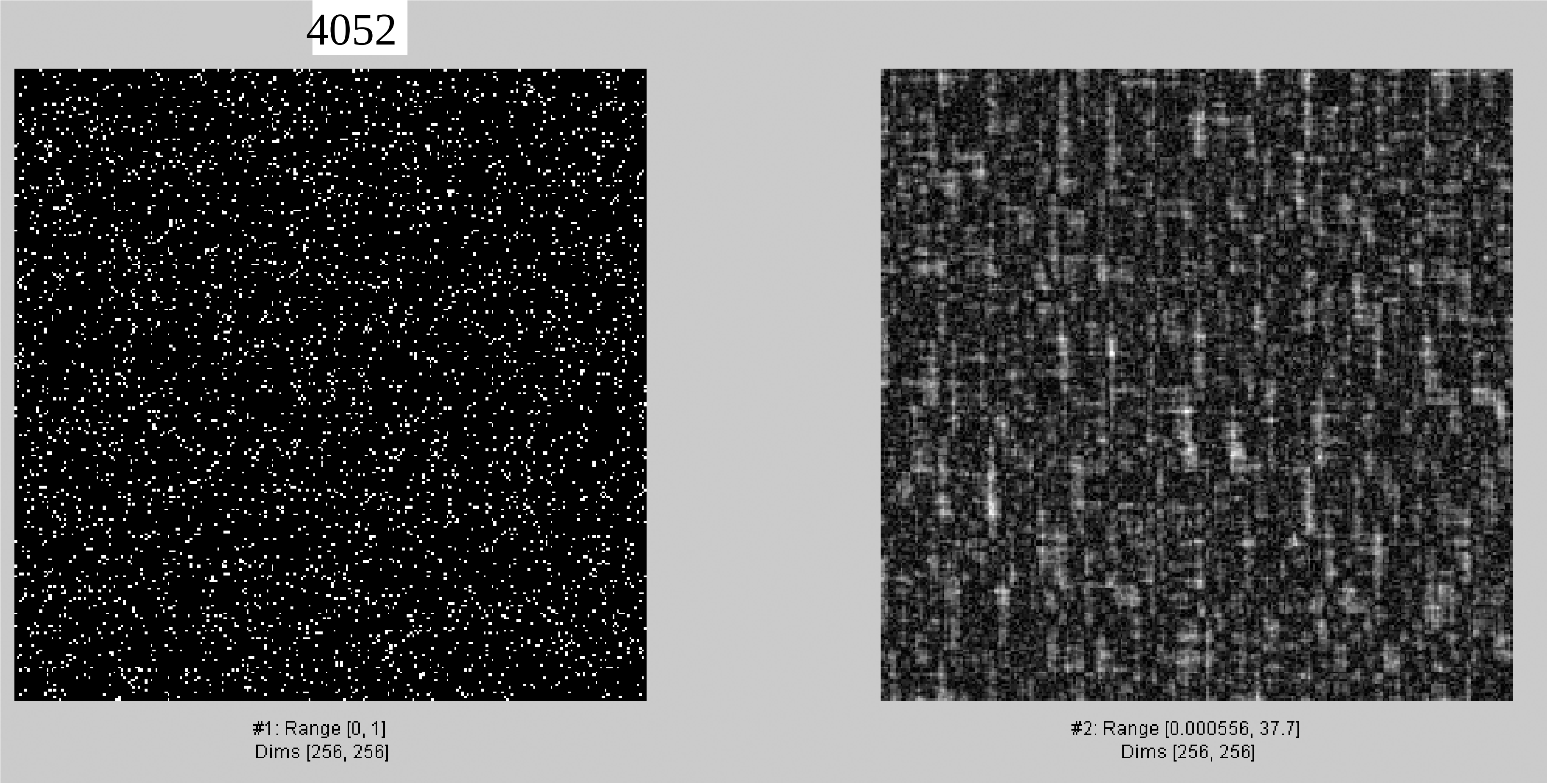
1088



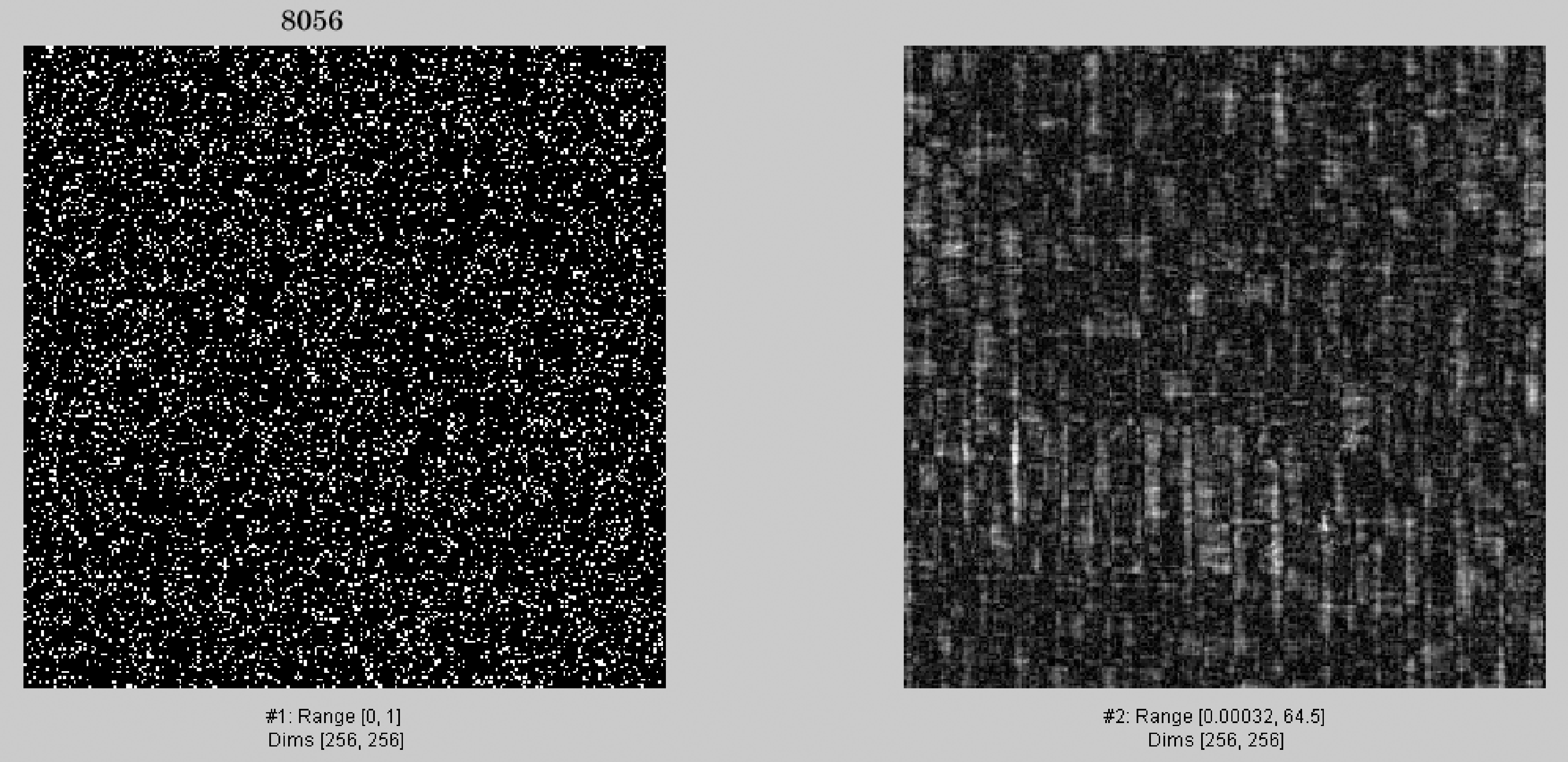
2094



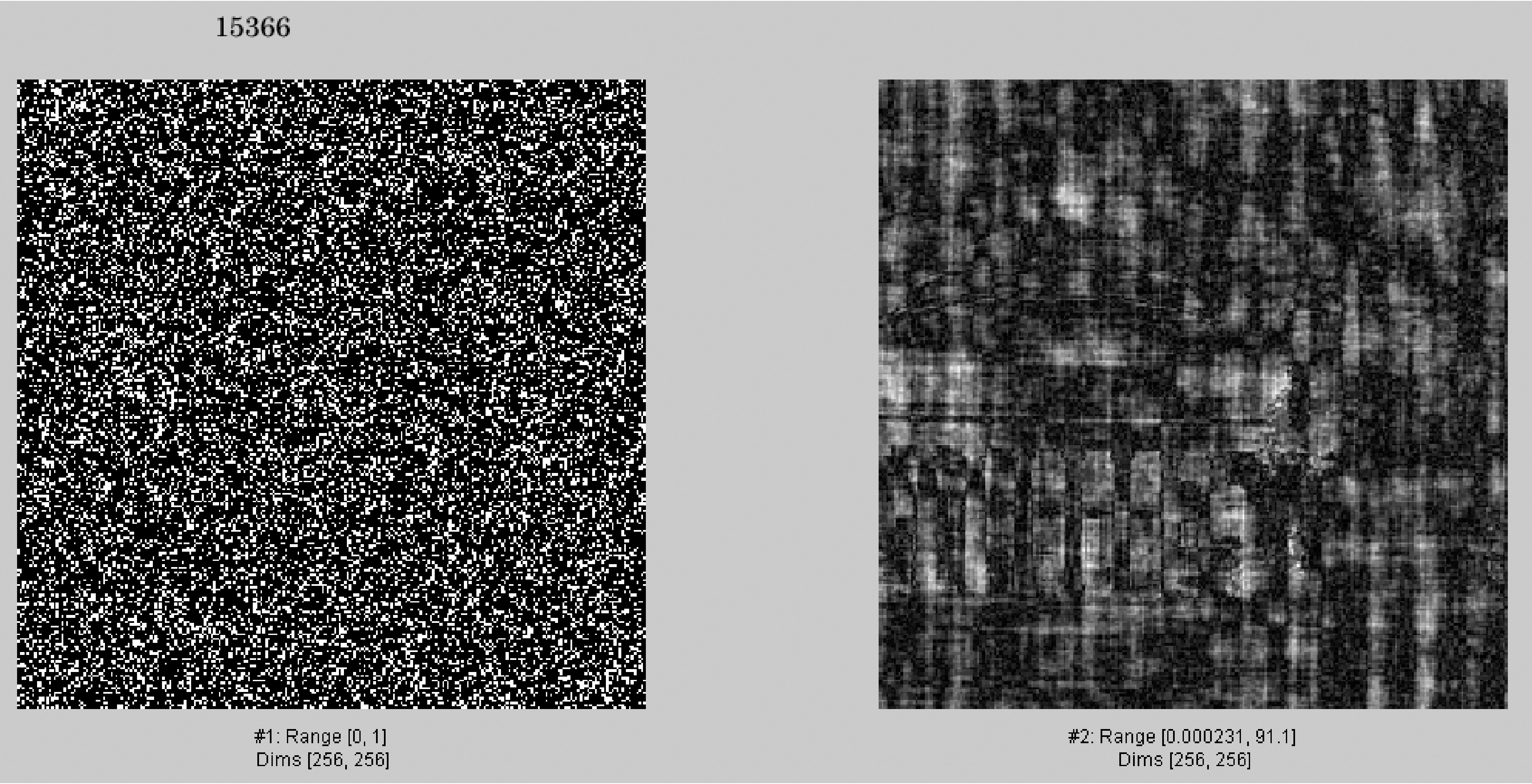
4052.



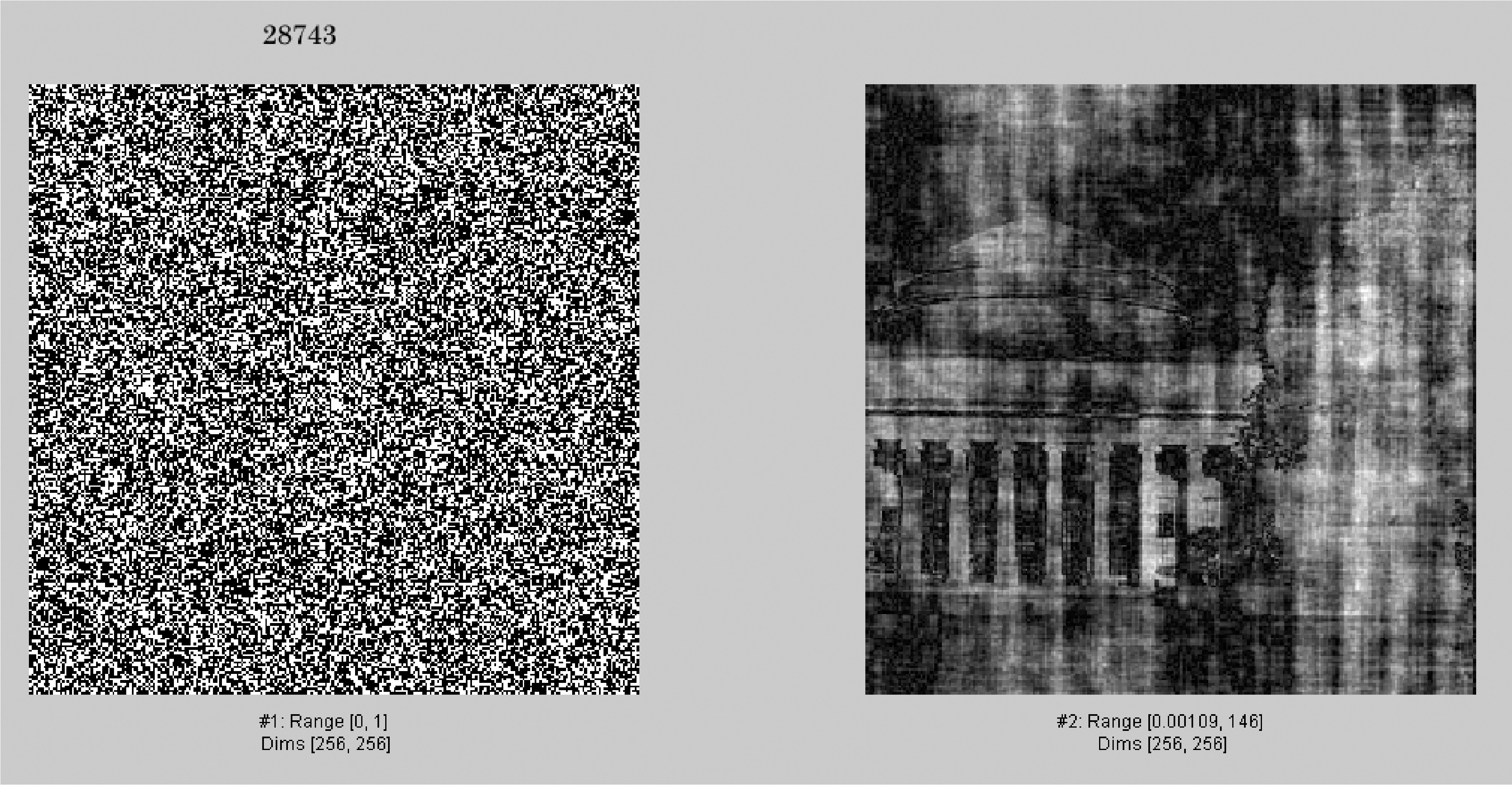
8056.



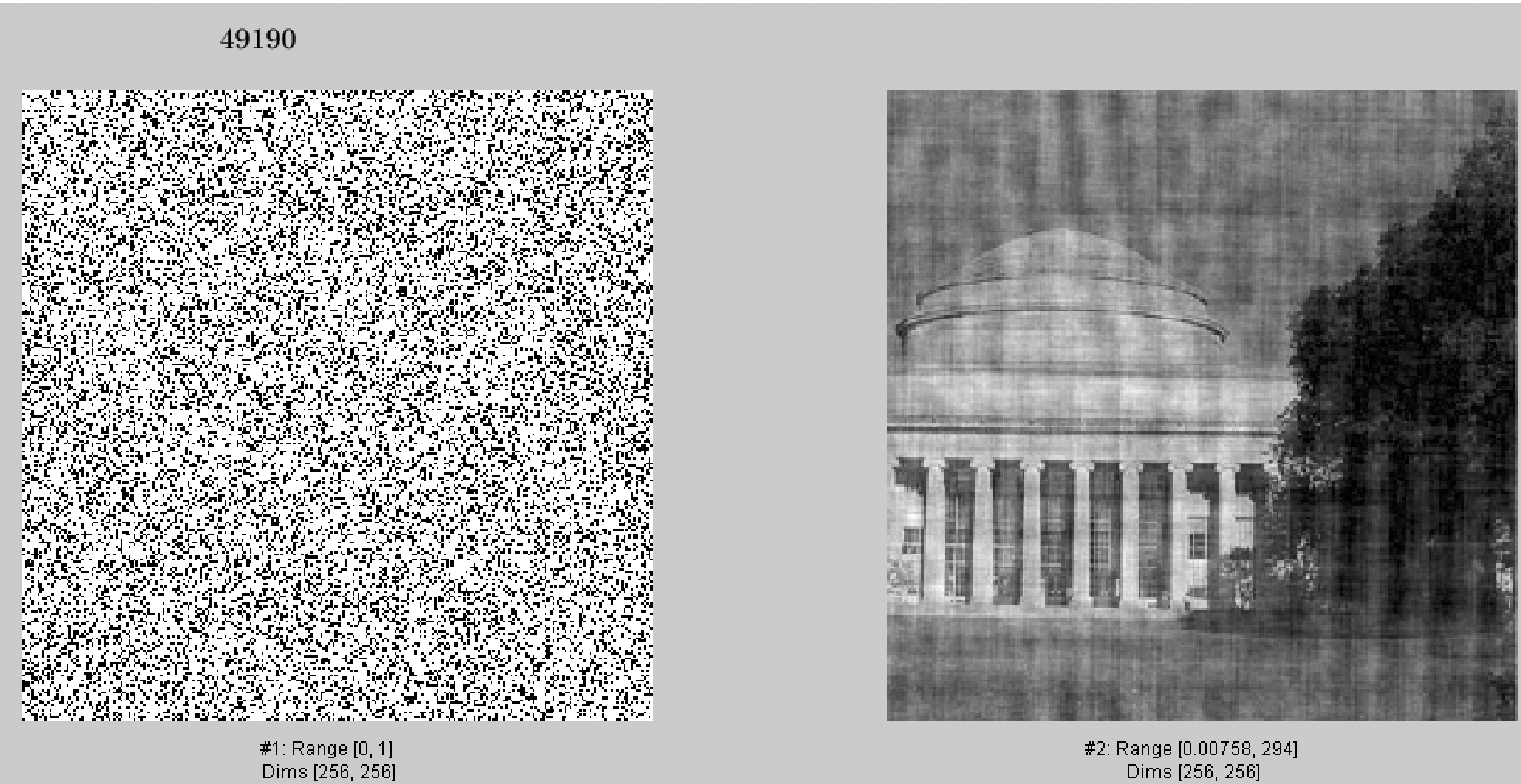
15366



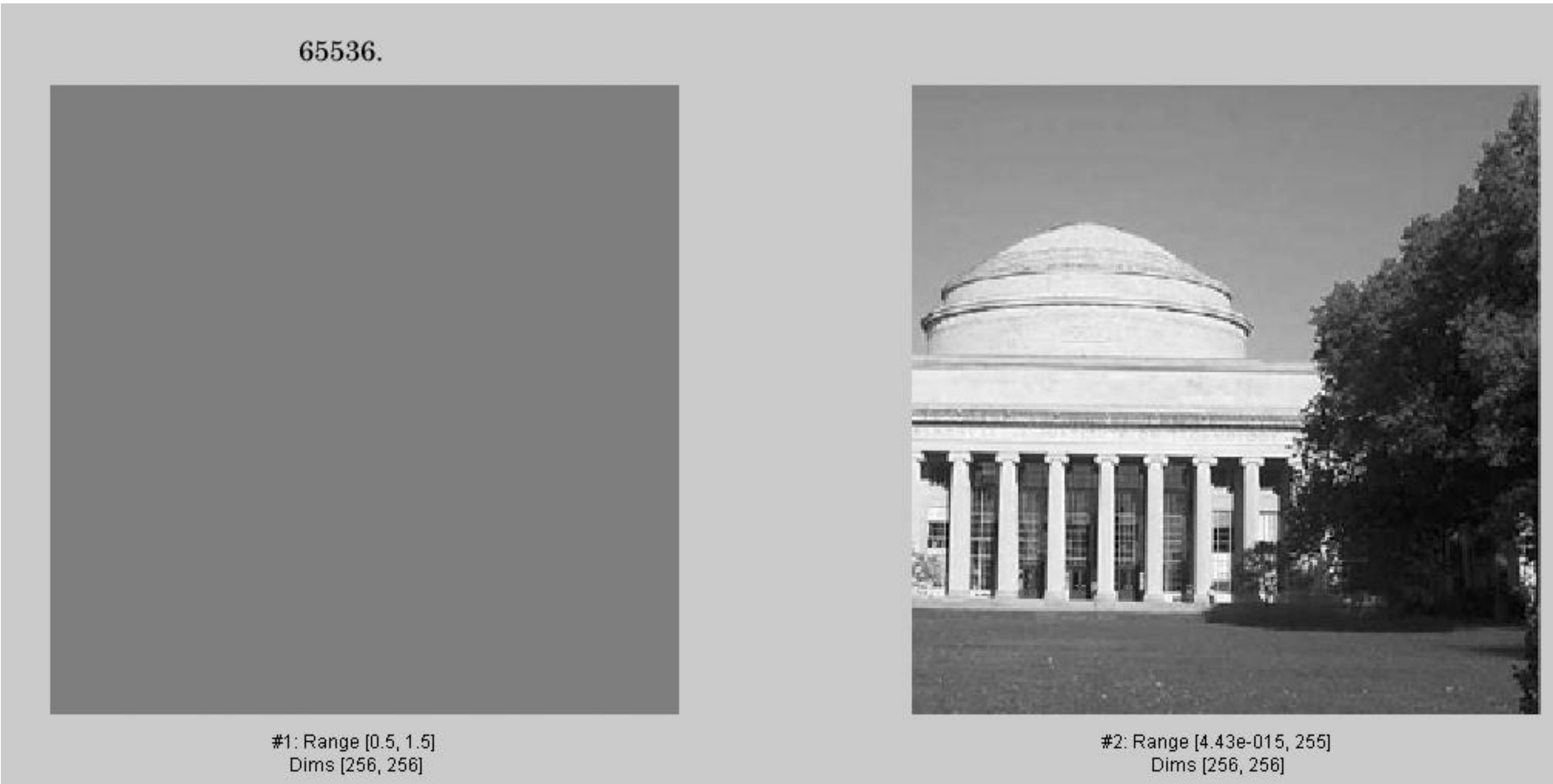
28743



49190.

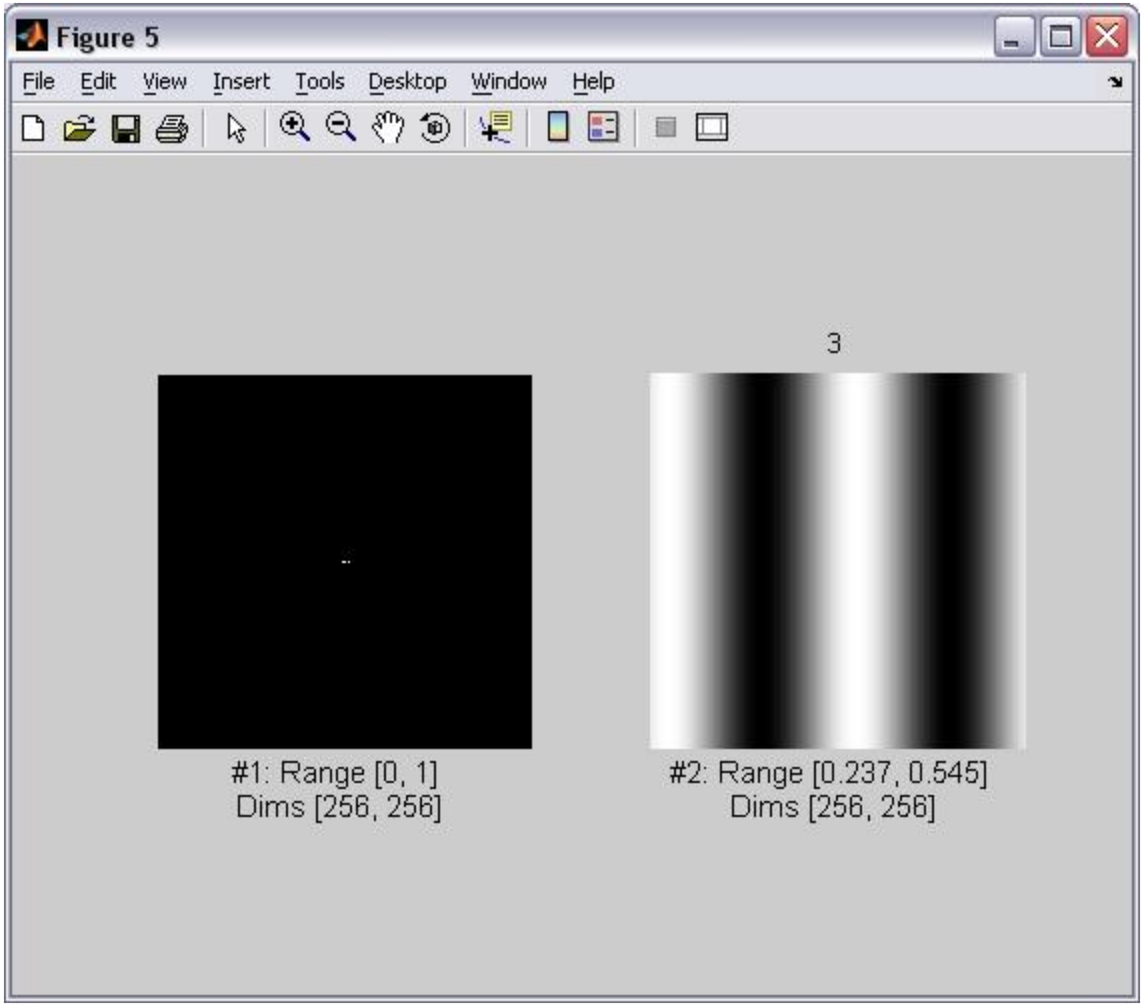


65536.



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

3



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

5

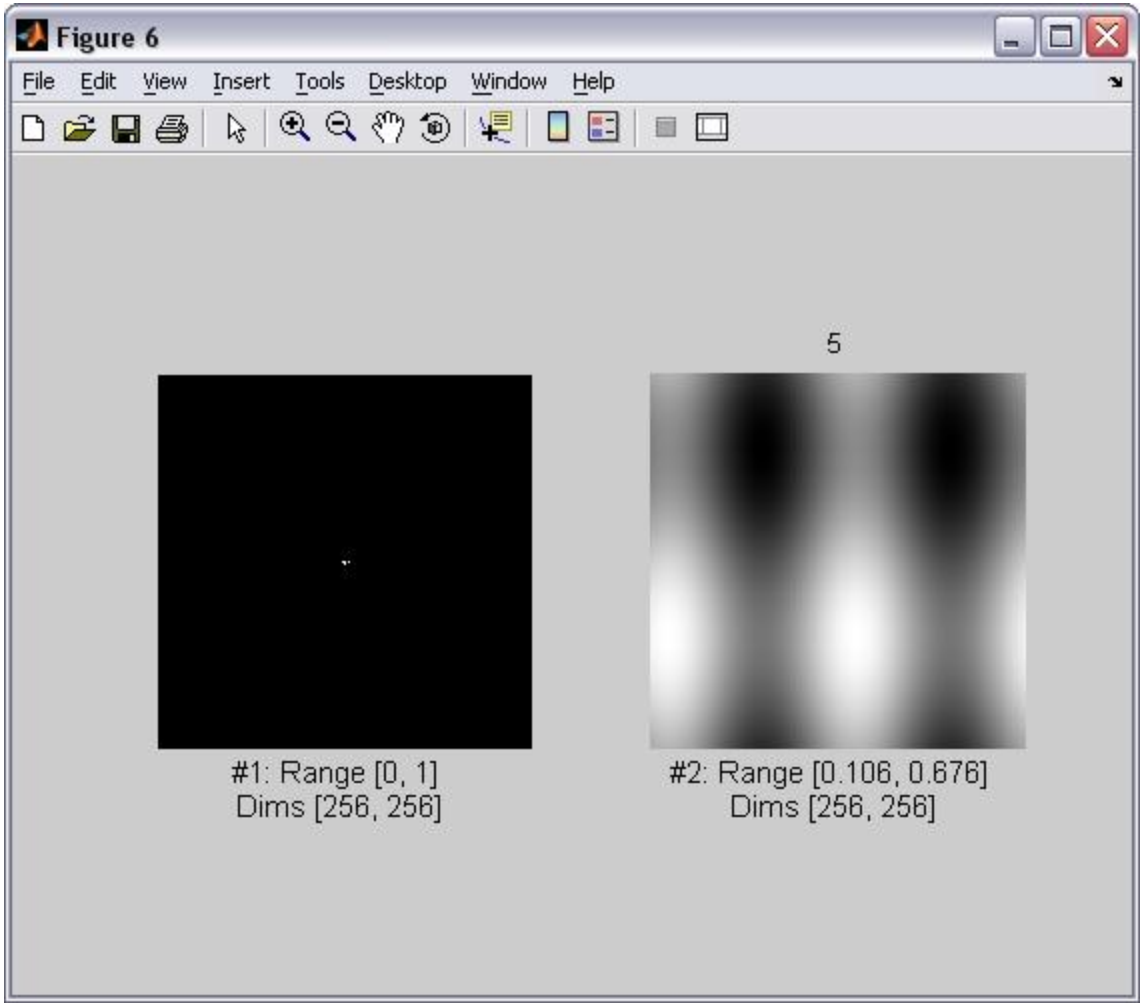


Figure 6
File Edit View Insert Tools Desktop Window Help
5
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.106, 0.676]
Dims [256, 256]

9

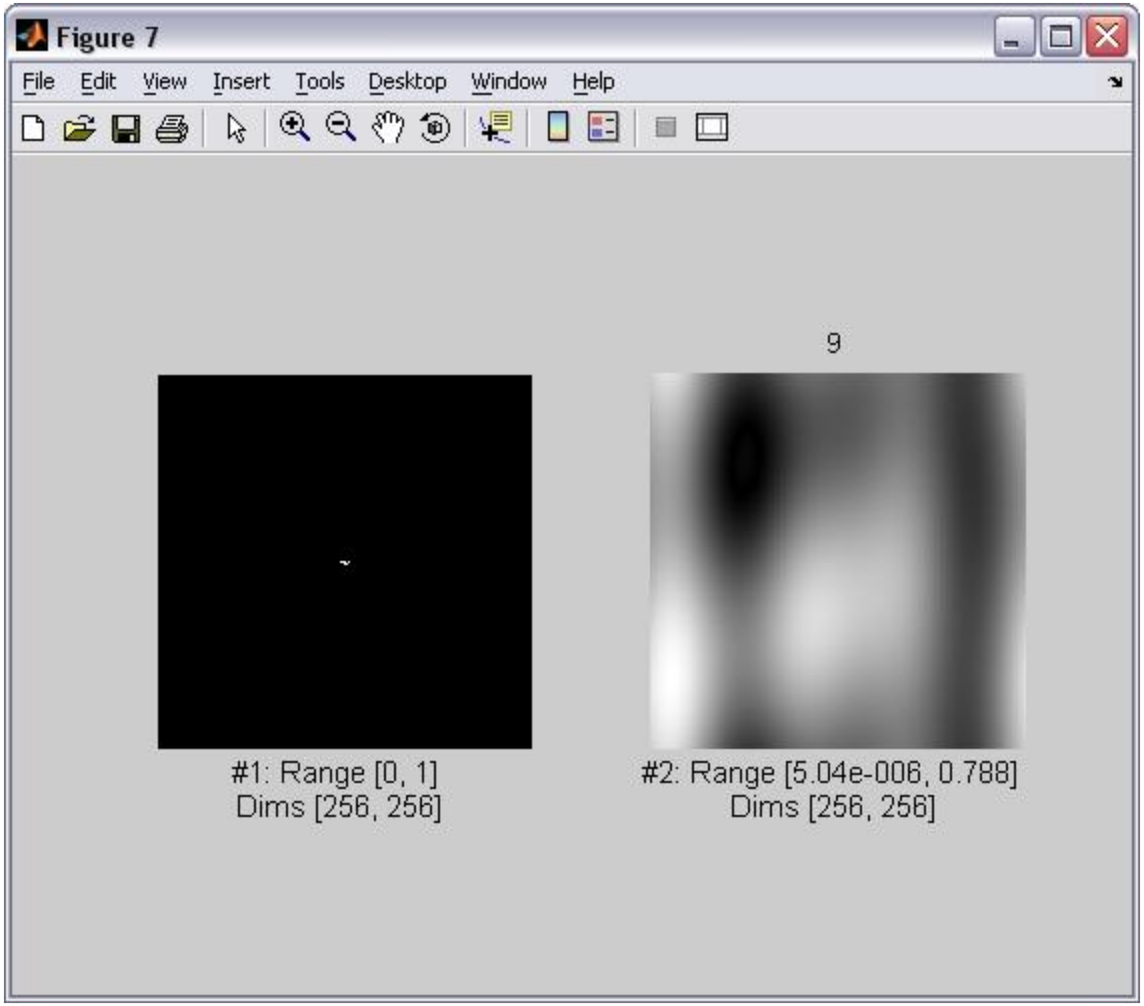


Figure 7
File Edit View Insert Tools Desktop Window Help
9
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.04e-006, 0.788]
Dims [256, 256]

17

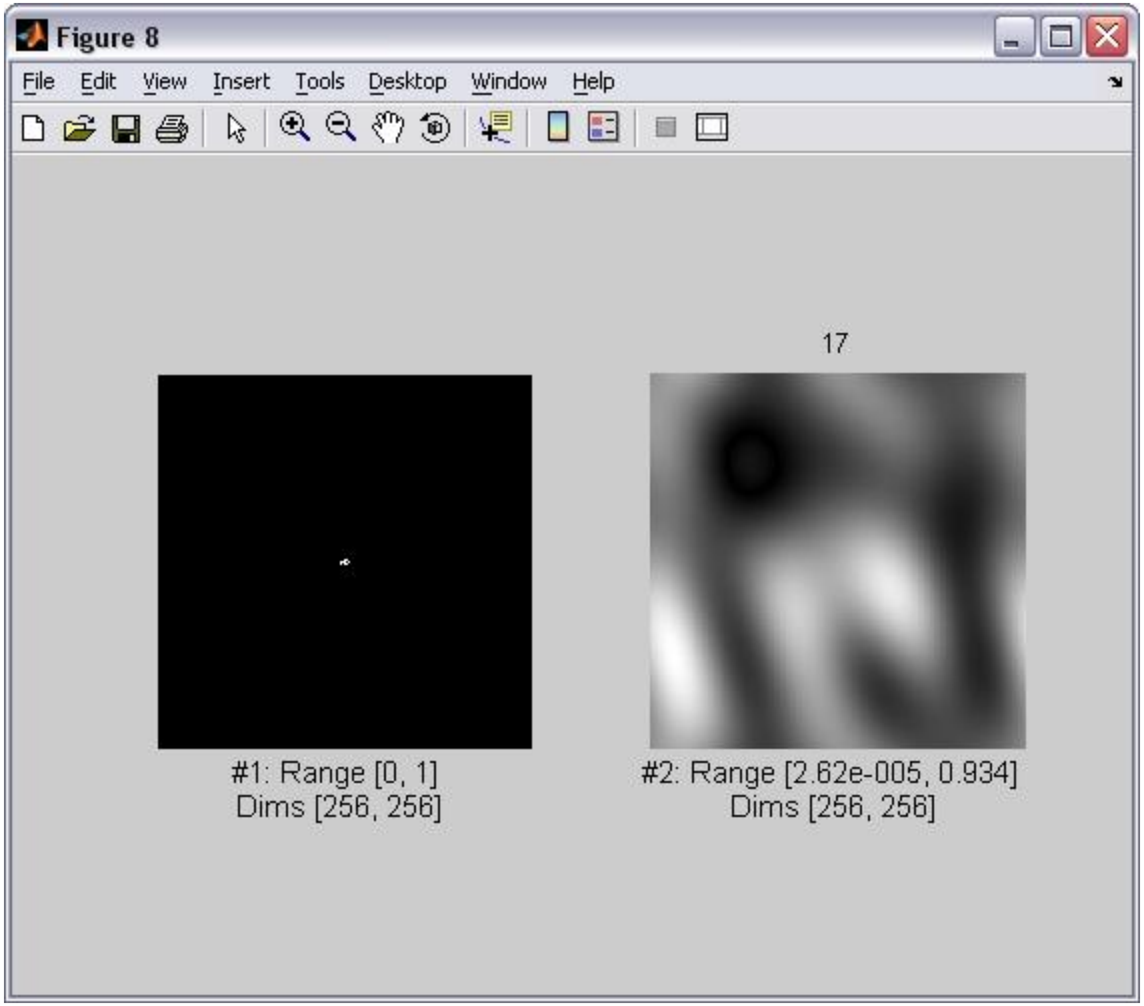


Figure 8
File Edit View Insert Tools Desktop Window Help
17
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.62e-005, 0.934]
Dims [256, 256]

33

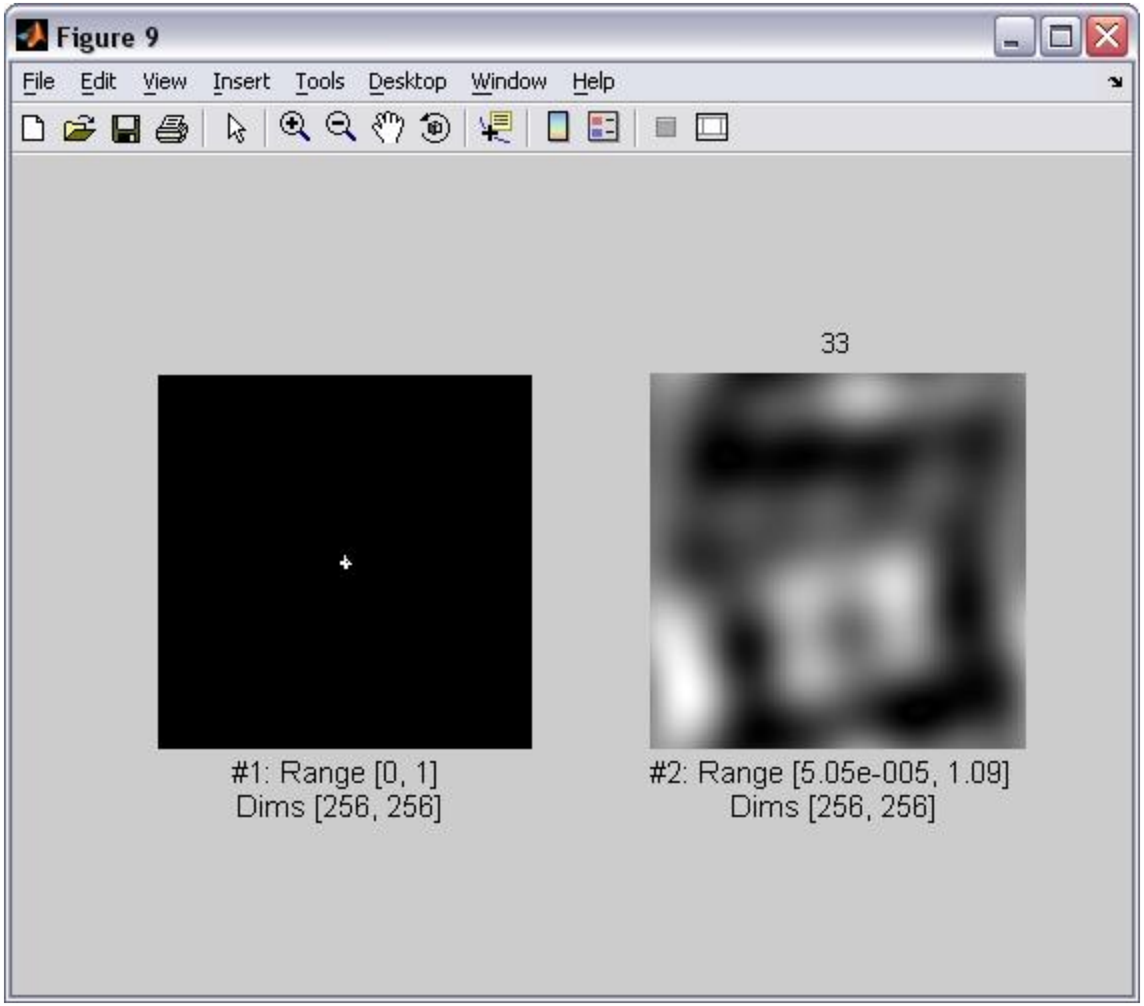


Figure 9
File Edit View Insert Tools Desktop Window Help
33
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.05e-005, 1.09]
Dims [256, 256]

65

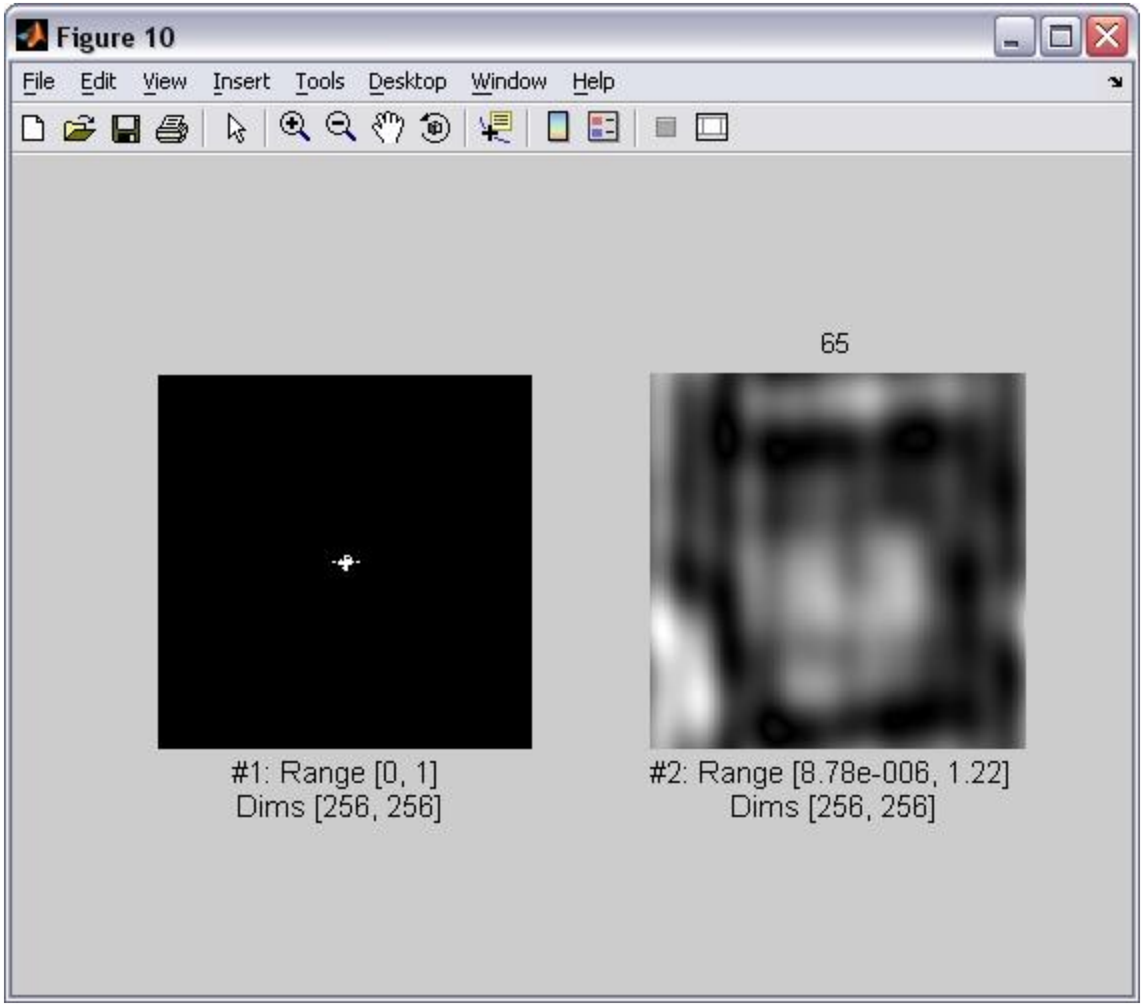


Figure 10
File Edit View Insert Tools Desktop Window Help
65
#1: Range [0, 1]
Dims [256, 256]
#2: Range [8.78e-006, 1.22]
Dims [256, 256]

129

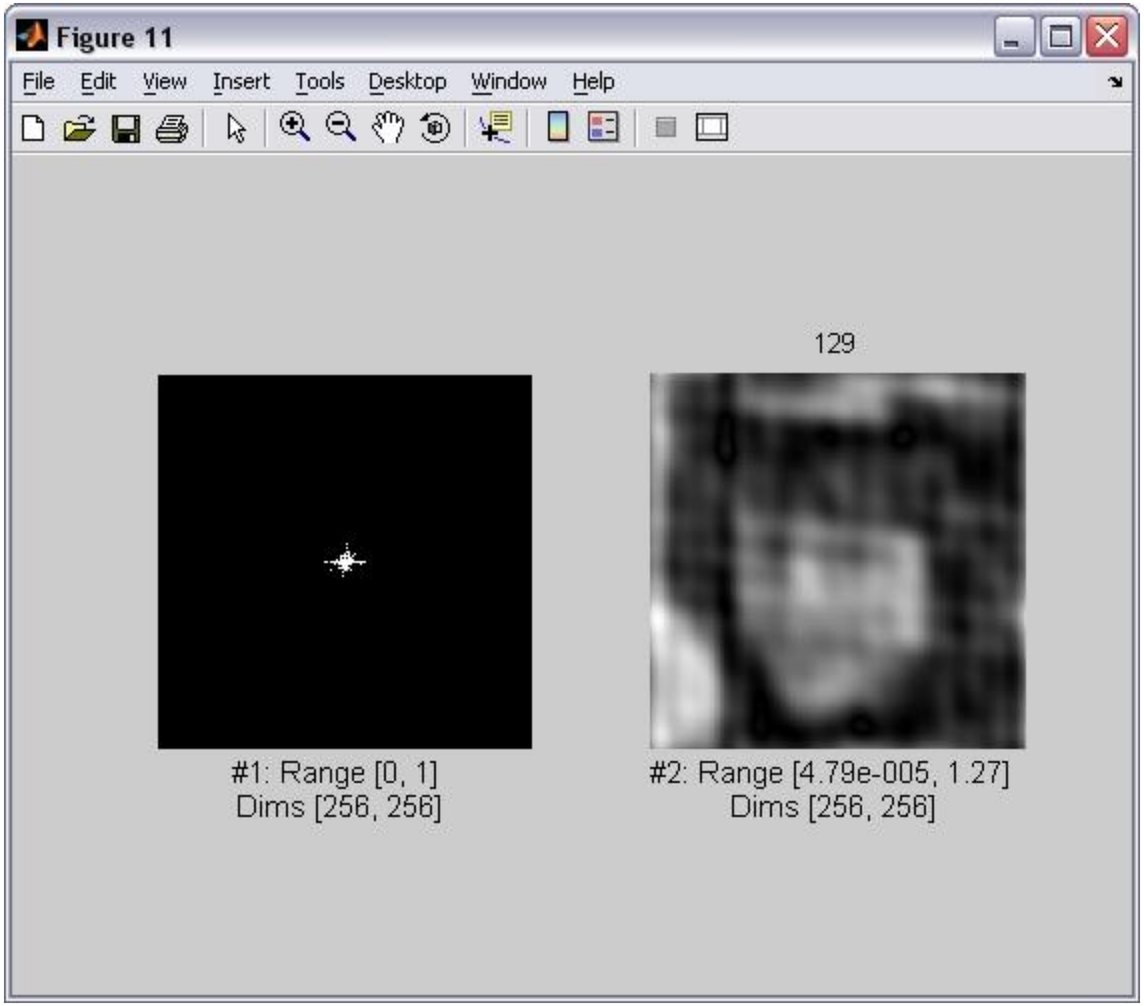


Figure 11
File Edit View Insert Tools Desktop Window Help
129
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.79e-005, 1.27]
Dims [256, 256]

257

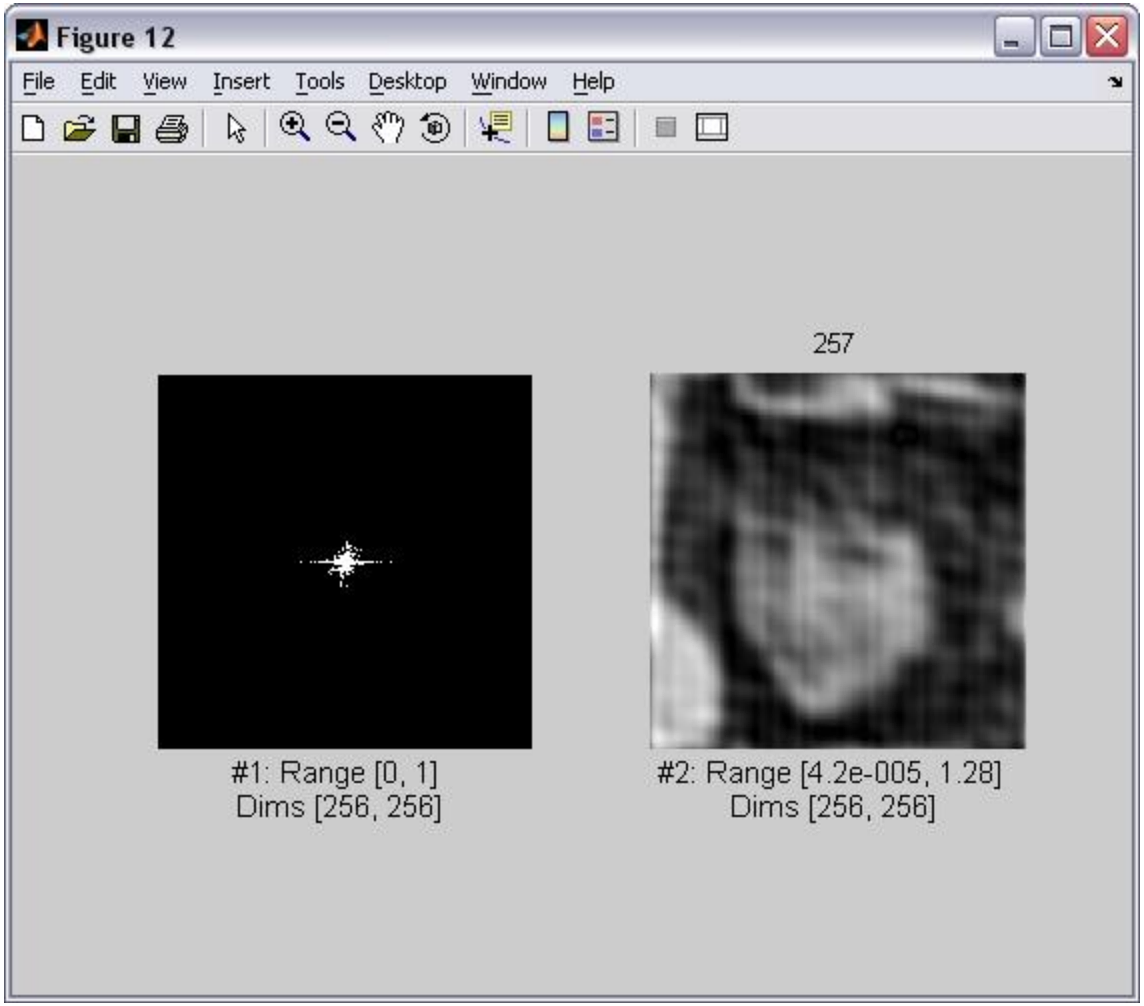


Figure 12
File Edit View Insert Tools Desktop Window Help
257
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.2e-005, 1.28]
Dims [256, 256]

513

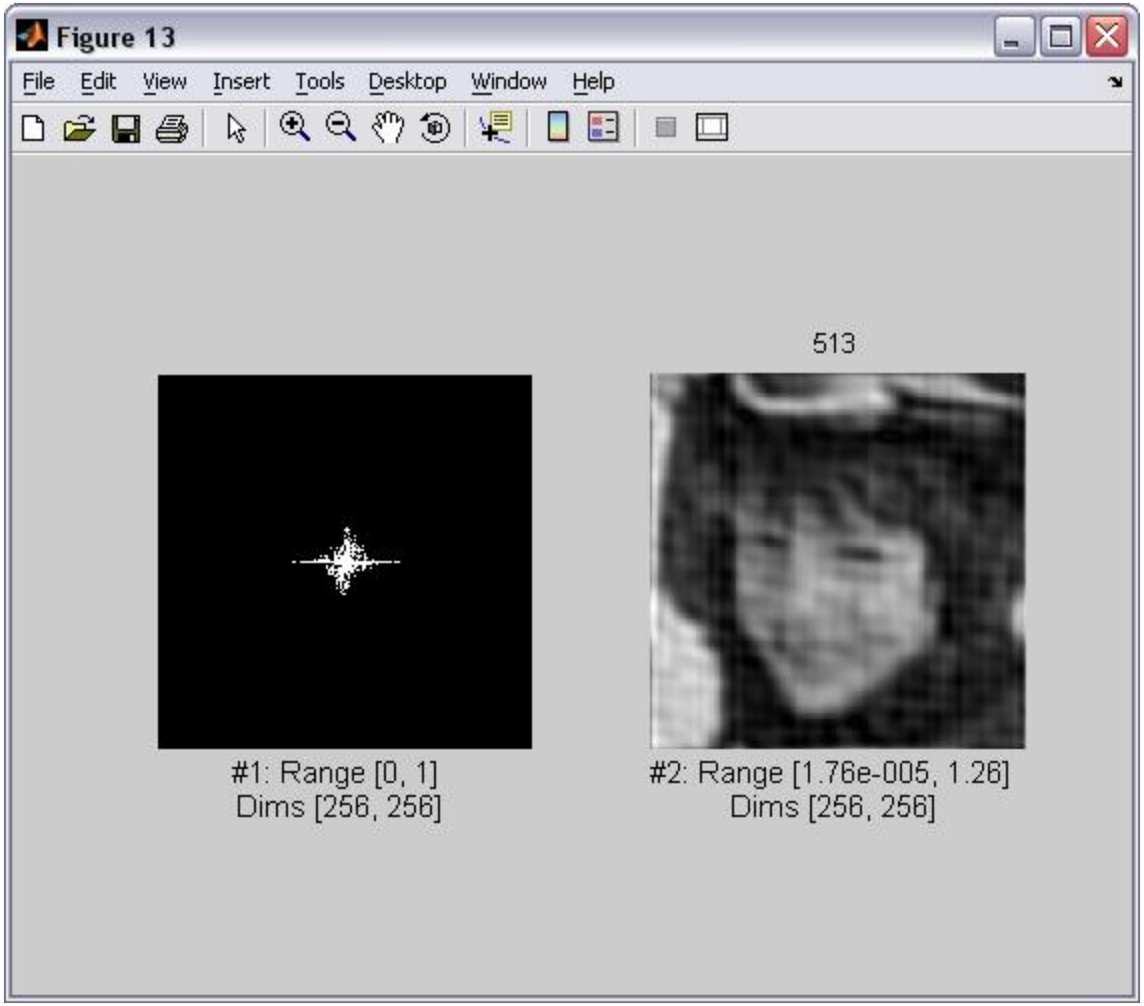


Figure 13
File Edit View Insert Tools Desktop Window Help
513
#1: Range [0, 1]
Dims [256, 256]
#2: Range [1.76e-005, 1.26]
Dims [256, 256]

1025

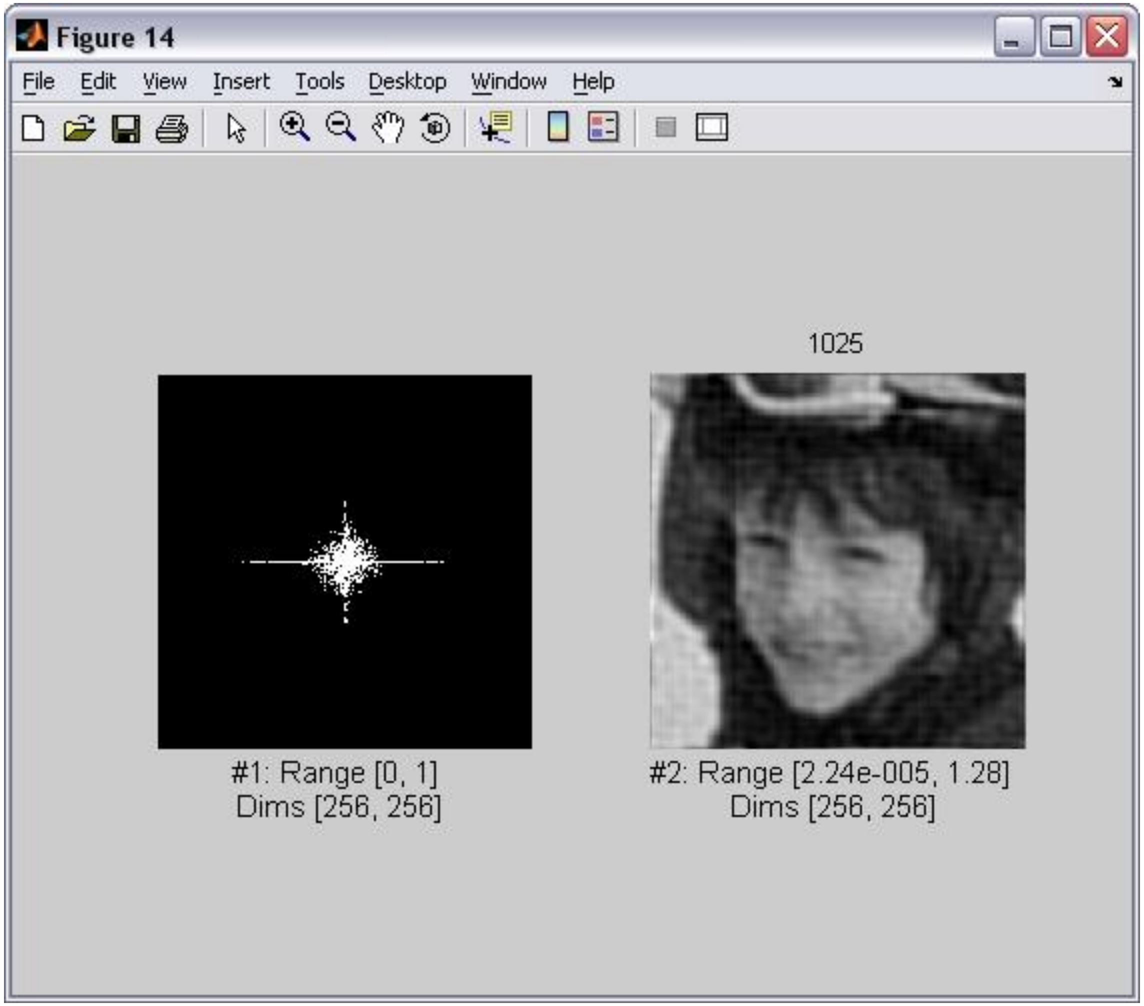


Figure 14
File Edit View Insert Tools Desktop Window Help
1025
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.24e-005, 1.28]
Dims [256, 256]

2049

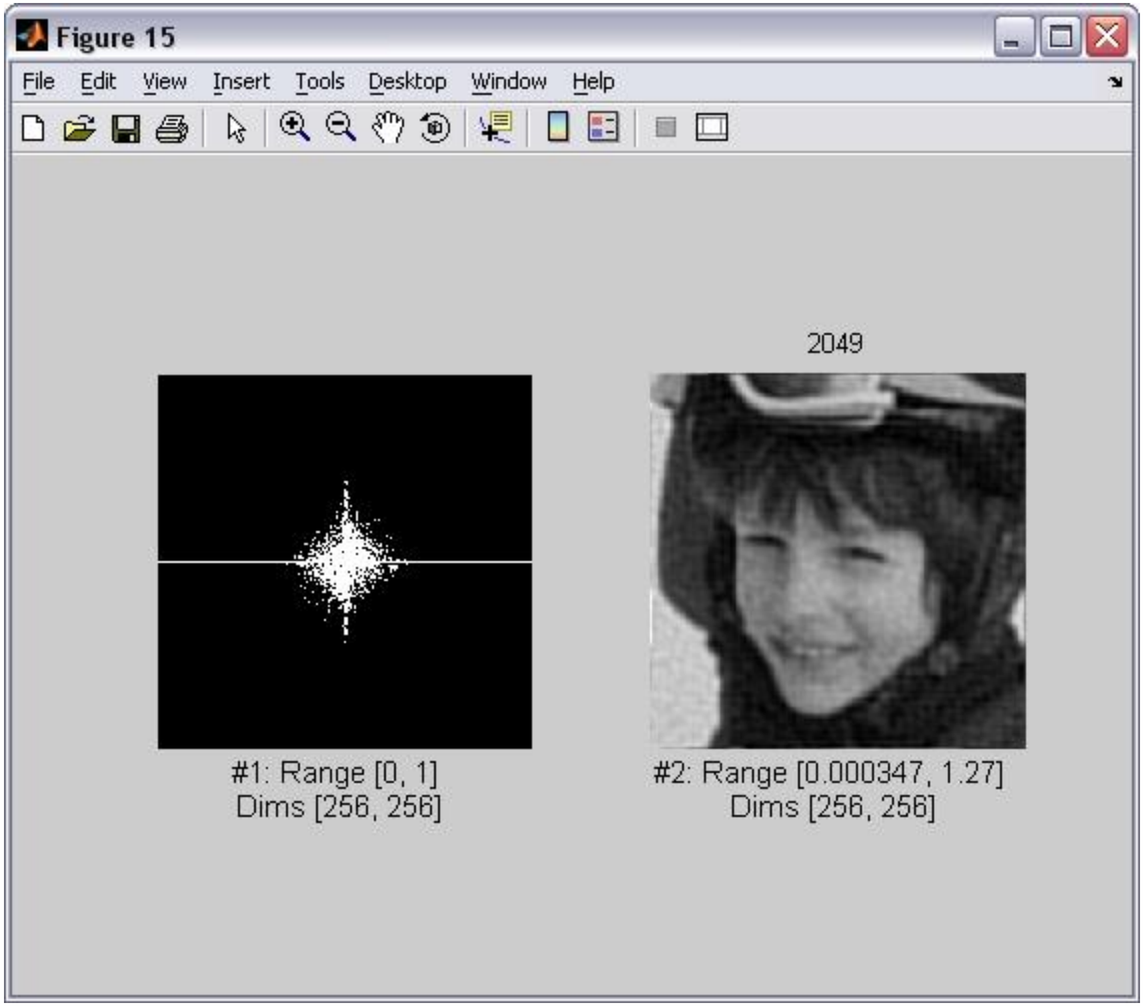


Figure 15
File
Edit
View
Insert
Tools
Desktop
Window
Help
2049
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000347, 1.27]
Dims [256, 256]

4097

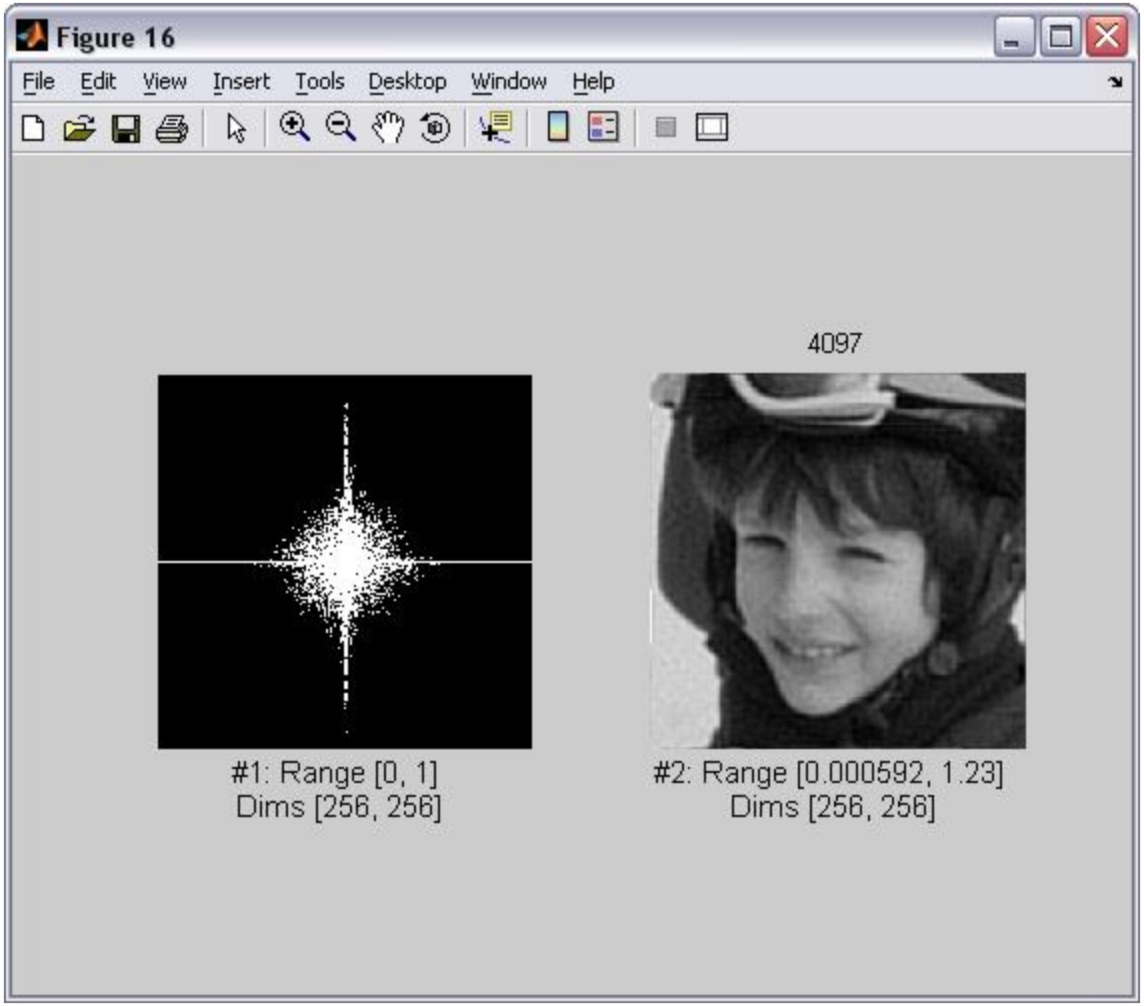


Figure 16
File
Edit
View
Insert
Tools
Desktop
Window
Help
4097
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000592, 1.23]
Dims [256, 256]

8193

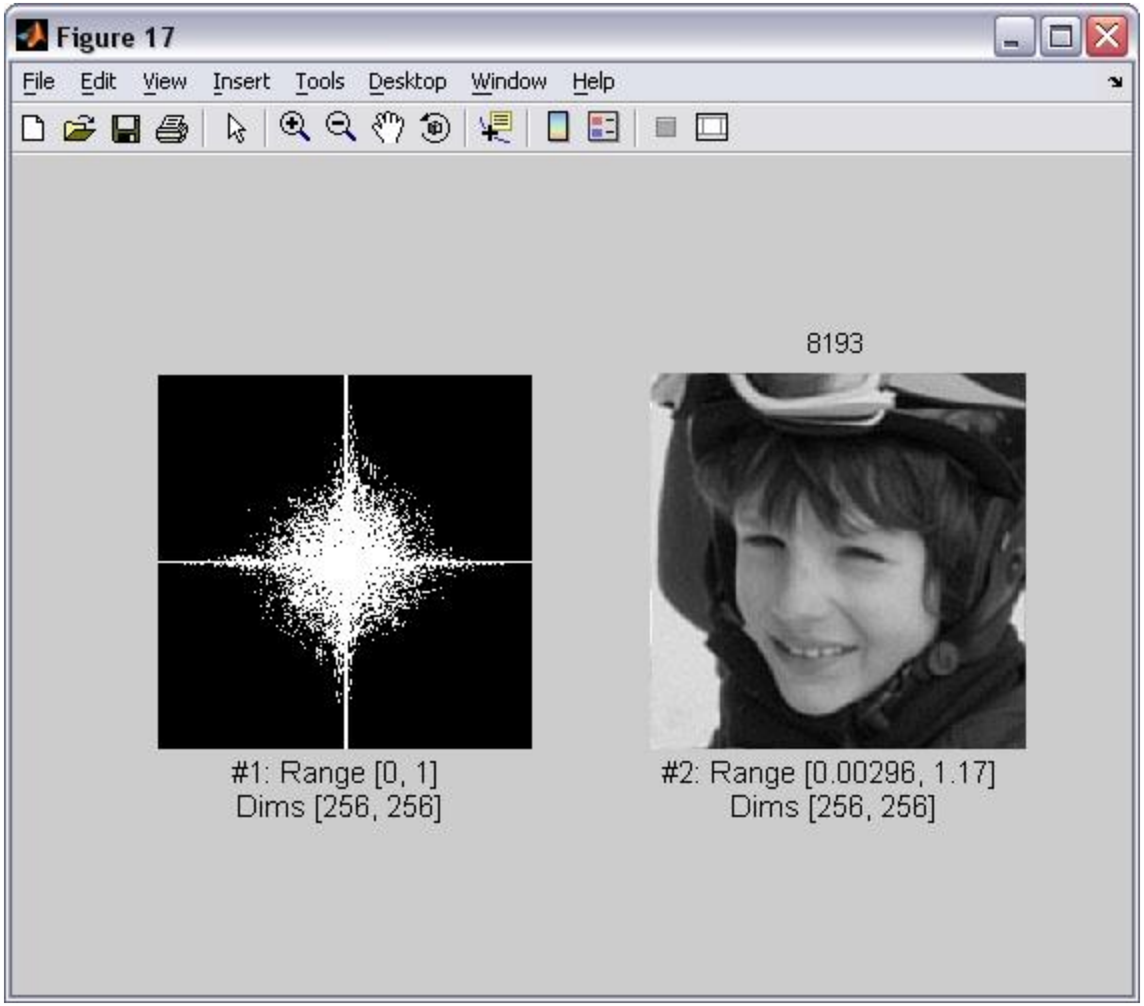


Figure 17
File Edit View Insert Tools Desktop Window Help
8193
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.00296, 1.17]
Dims [256, 256]

16385

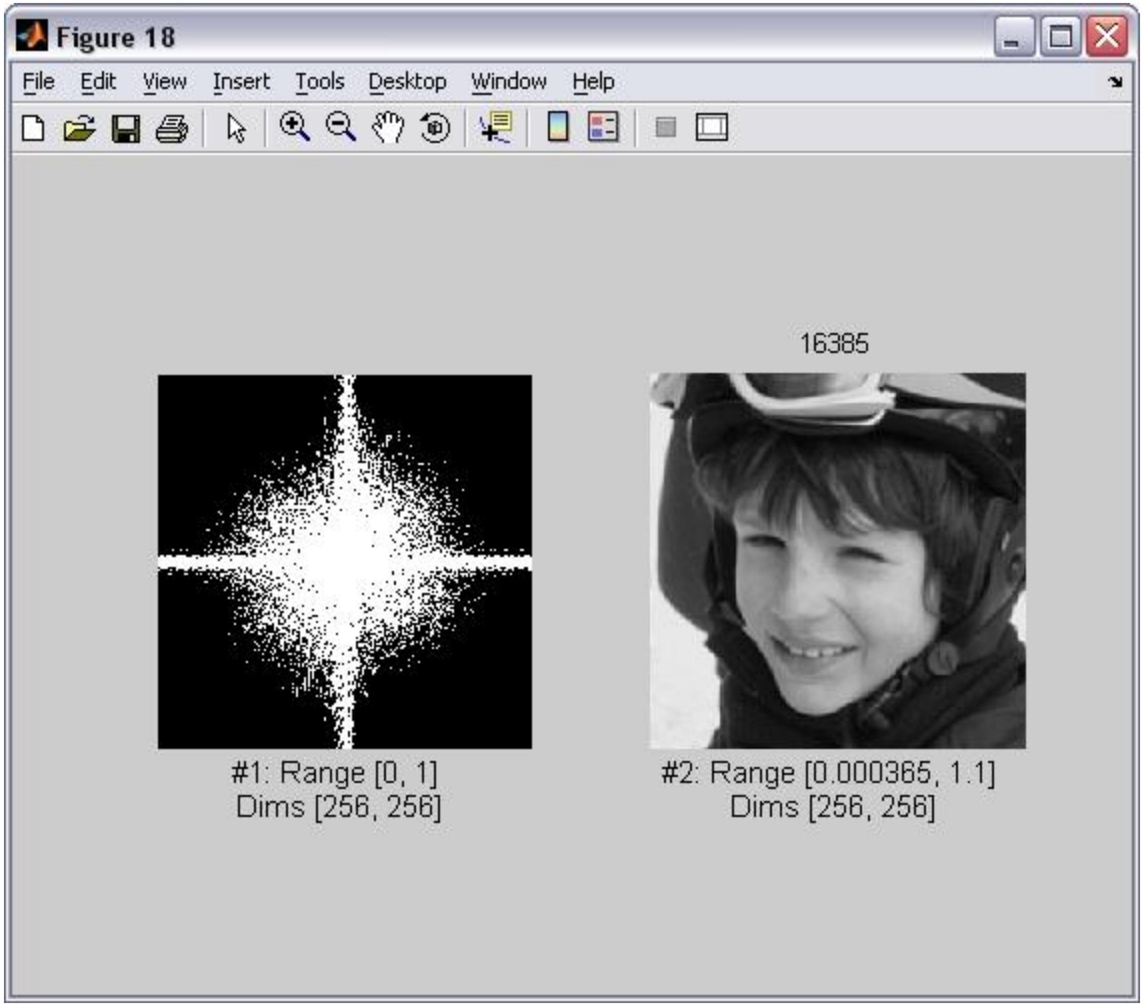


Figure 18
File Edit View Insert Tools Desktop Window Help
16385
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000365, 1.1]
Dims [256, 256]

32769

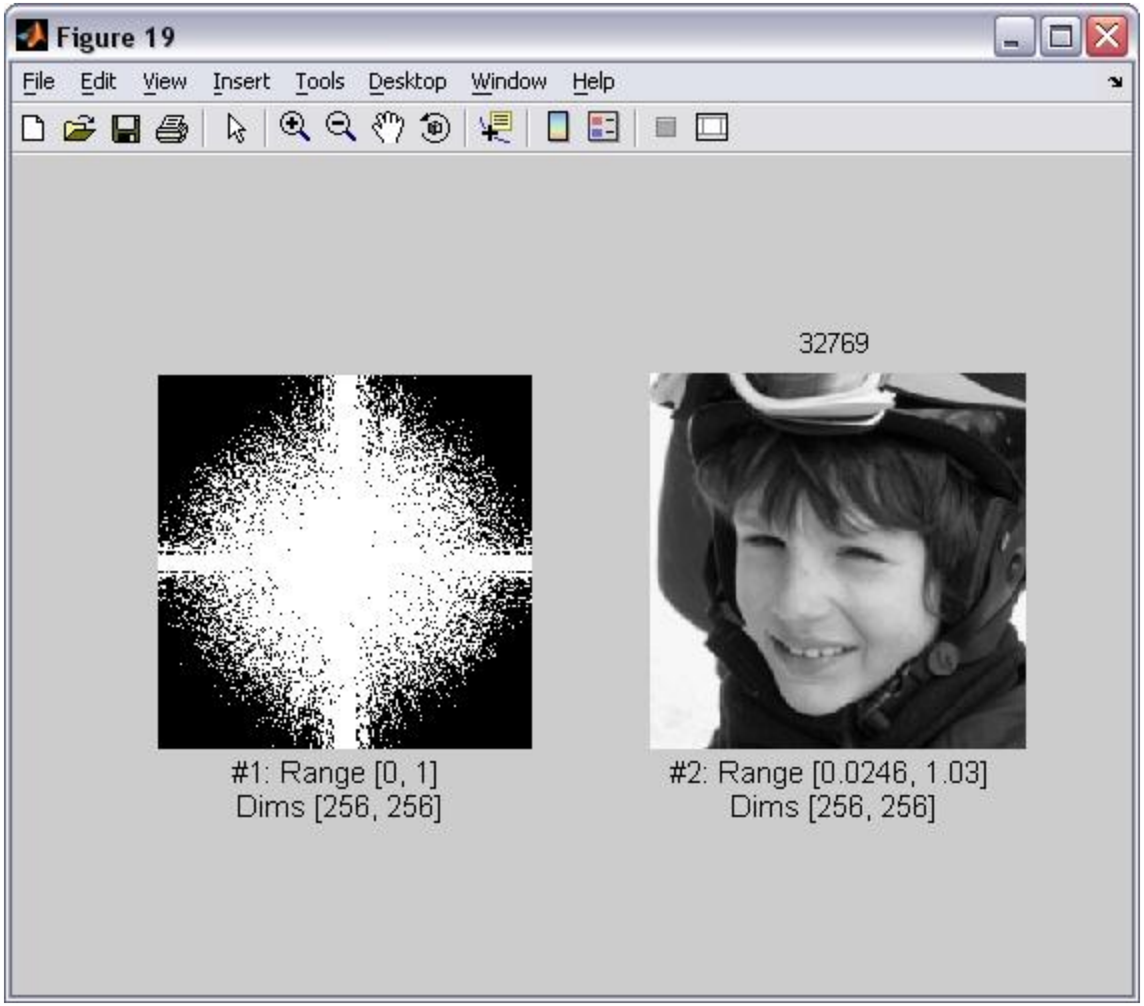


Figure 19
File Edit View Insert Tools Desktop Window Help
32769
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.0246, 1.03]
Dims [256, 256]

65536

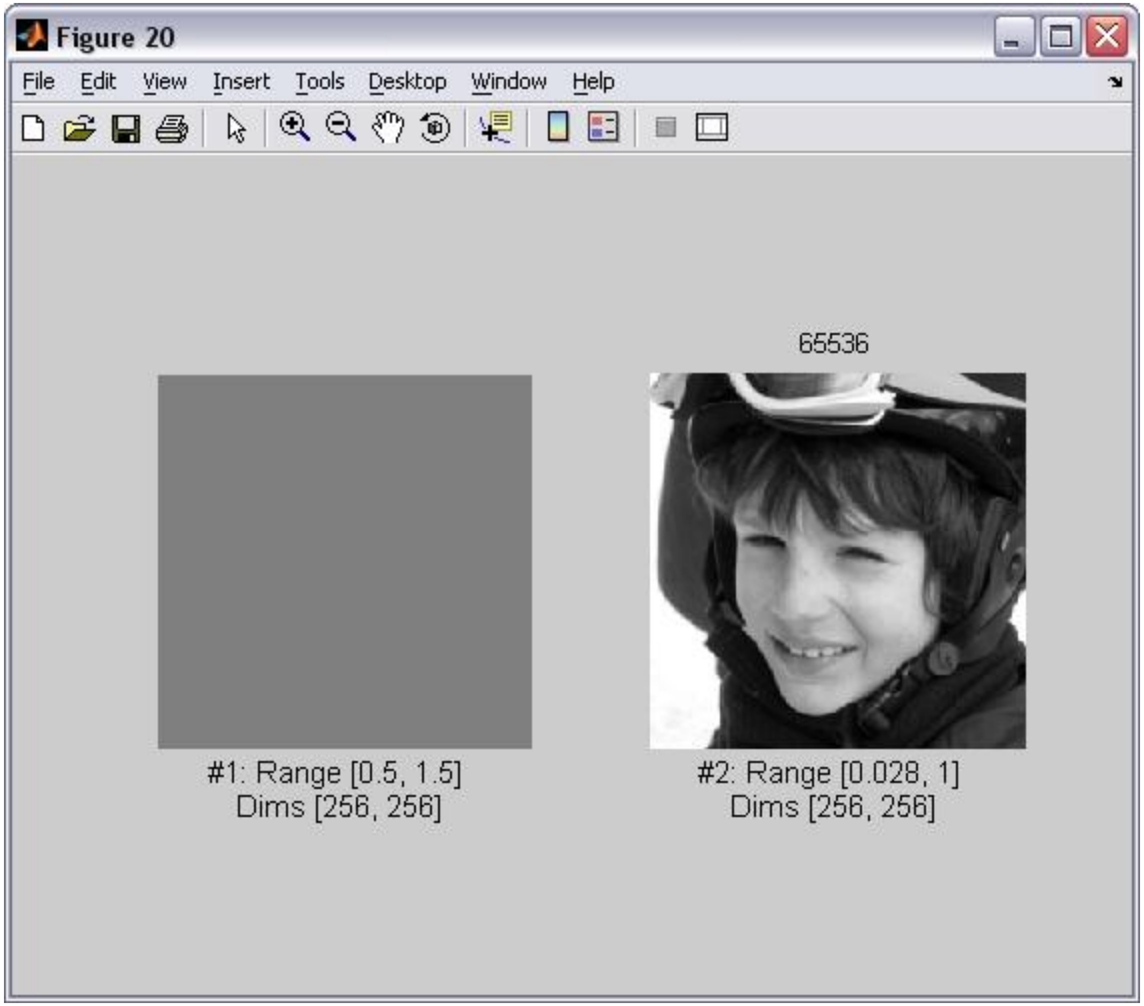
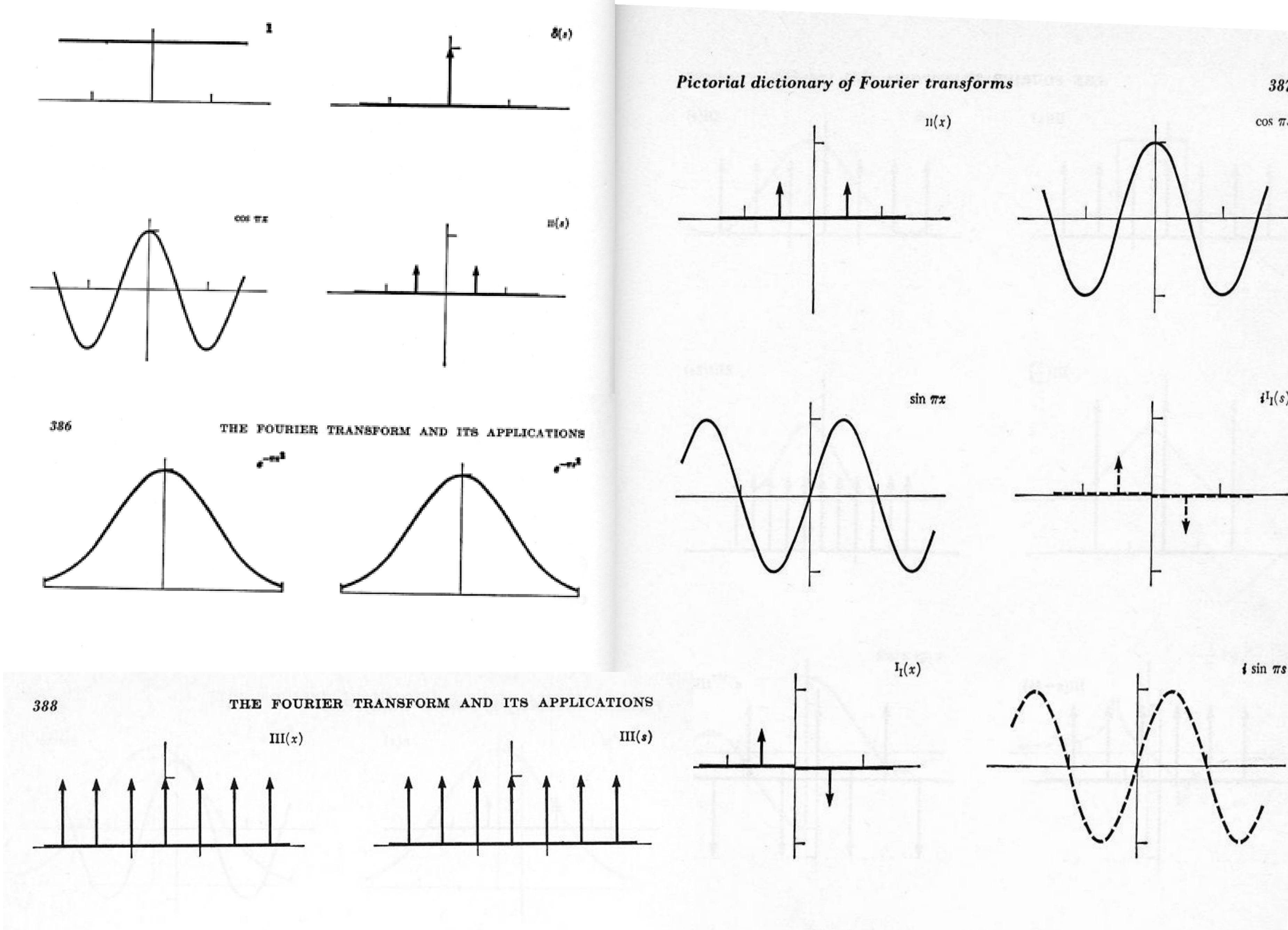


Figure 20
File Edit View Insert Tools Desktop Window Help
65536
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [0.028, 1]
Dims [256, 256]

Some important Fourier Transforms

Bracewell's pictorial dictionary of Fourier transform pairs



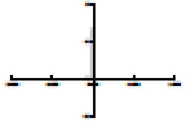
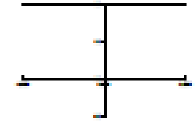
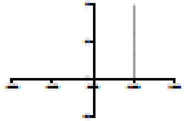
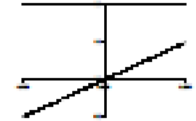
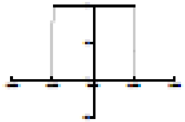
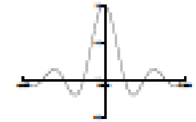
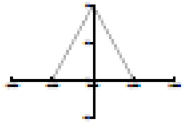
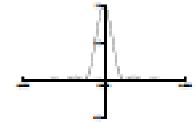
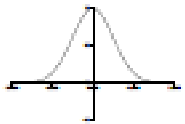
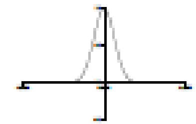
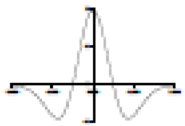
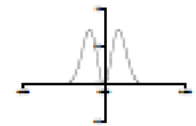
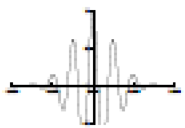
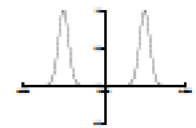
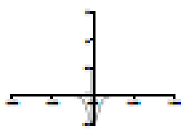
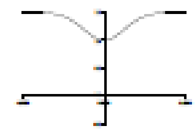
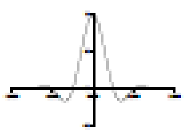
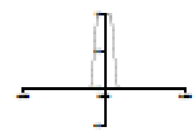
Name	Signal	Formula		Fourier transform	Spectrum
impulse		$\delta(x)$	$\Leftrightarrow$	1	
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$	
box filter		$\text{box}(x/a)$	$\Leftrightarrow$	$a\text{sinc}(a\omega)$	
tent		$\text{tent}(x/a)$	$\Leftrightarrow$	$a\text{sinc}^2(a\omega)$	
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$	
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2G(\omega;\sigma^{-1})$	
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		$\text{rcos}(x/(aW))$ $\text{sinc}(x/a)$	$\Leftrightarrow$	(see Figure 3.29)	

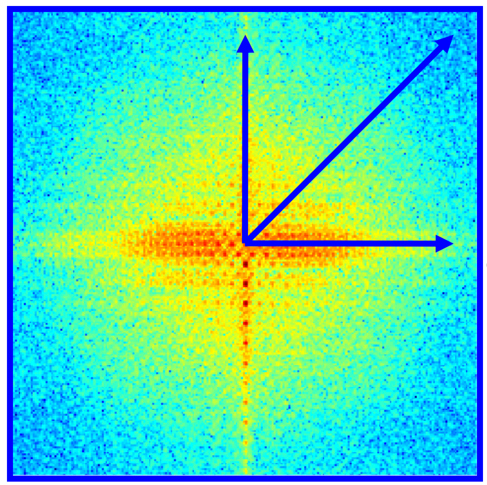
Table 3.2 Some useful (continuous) Fourier transform pairs: The dashed line in the Fourier transform of the shifted impulse indicates its (linear) phase. All other transforms have zero phase (they are real-valued). Note that the figures are not necessarily drawn to scale but are drawn to illustrate the general shape and characteristics of the filter or its response. In

How to interpret a 2-d Fourier Spectrum

Vertical orientation

45 deg.

0

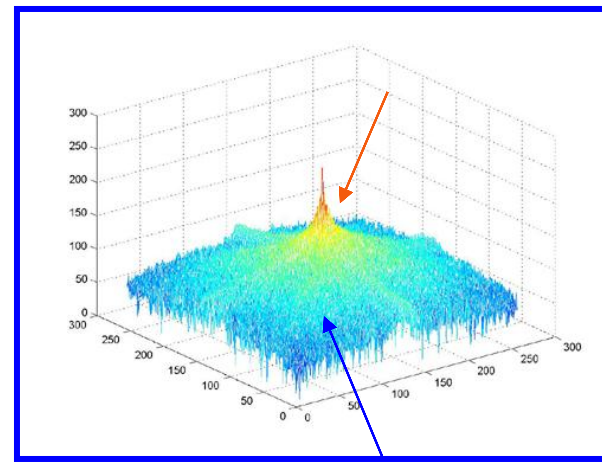


Horizontal orientation

0 f_{max}

fx in cycles/image

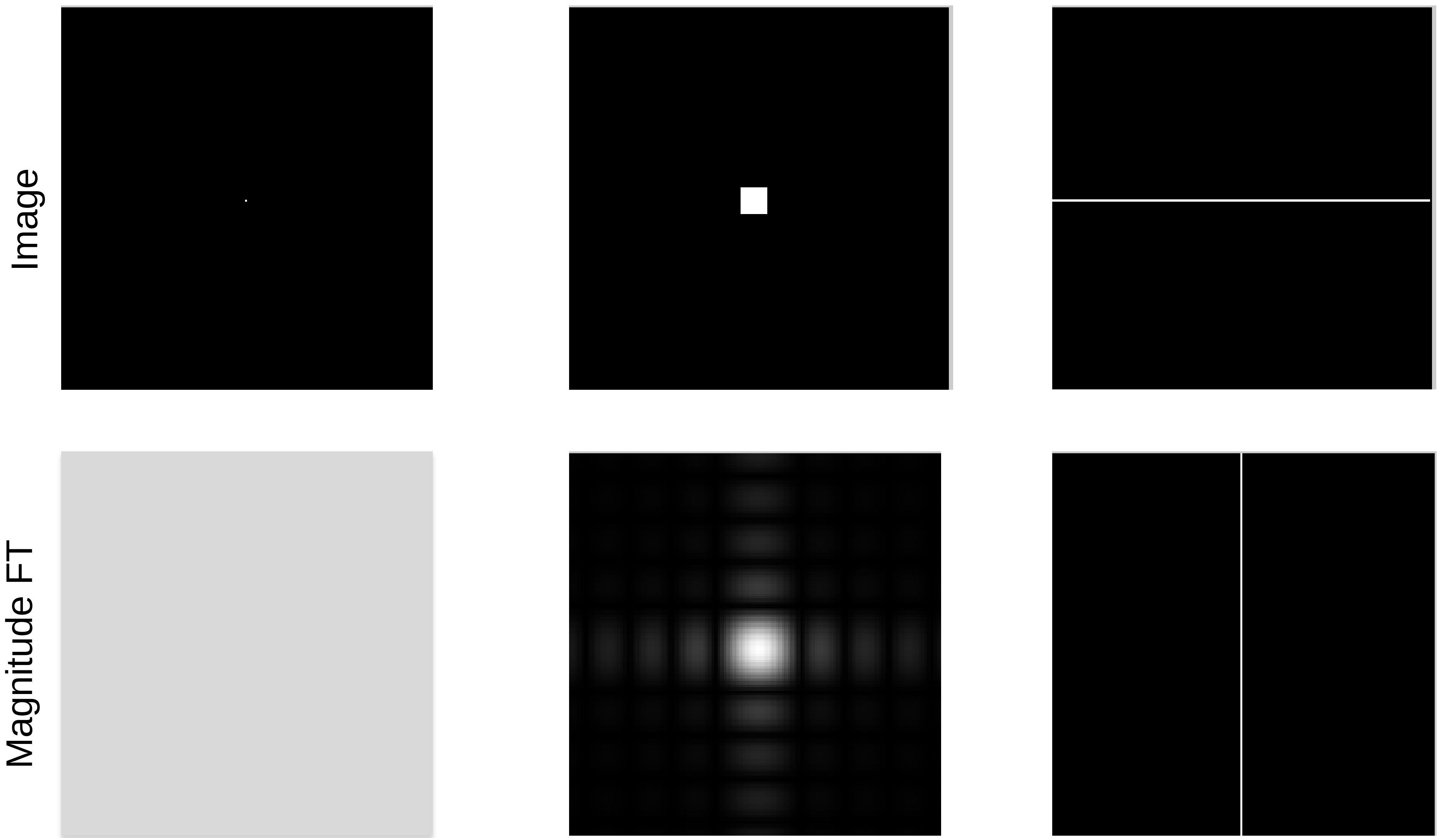
Low spatial frequencies



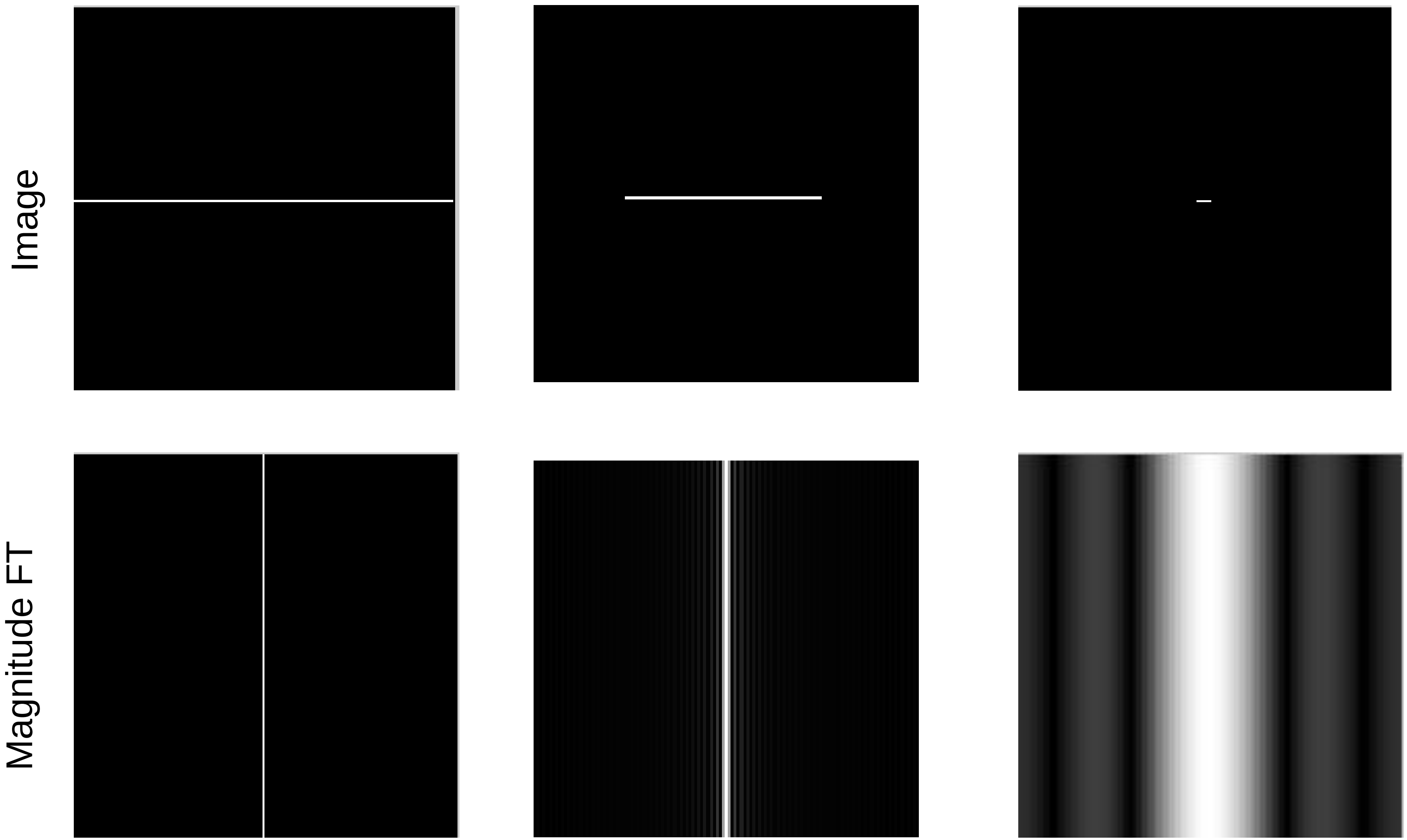
High spatial frequencies

Log power spectrum

Some important Fourier Transforms

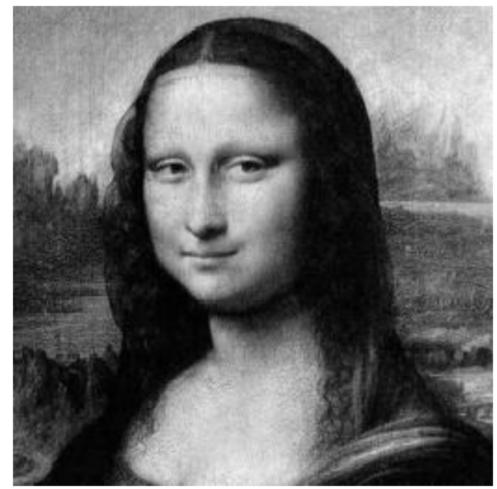


Some important Fourier Transforms

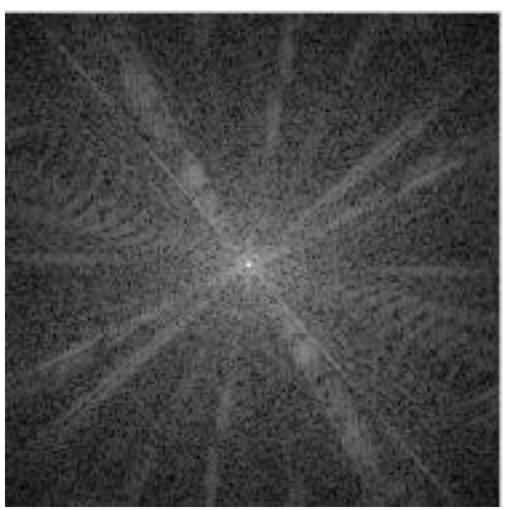


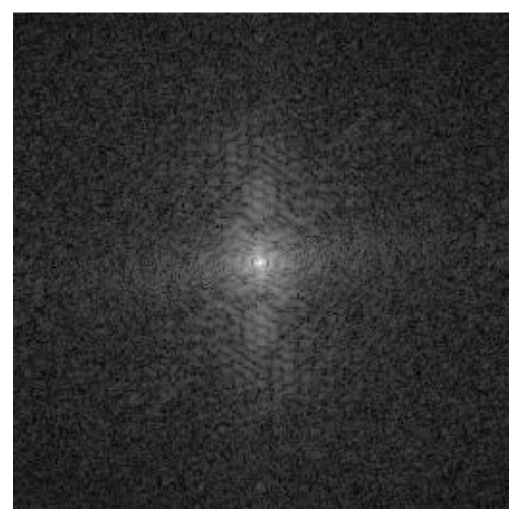
The Fourier Transform of some important images

Image

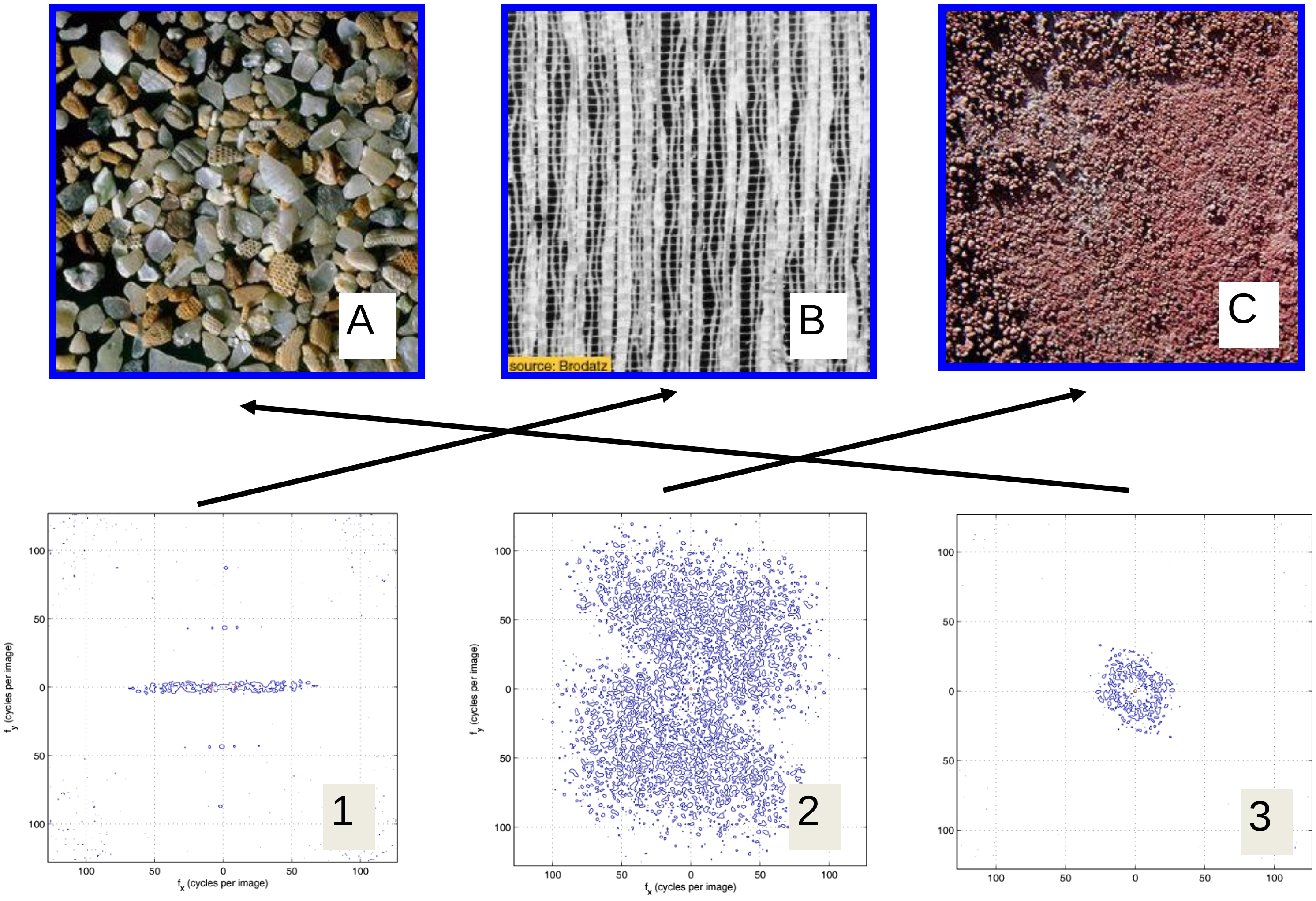


Log(1+Magnitude FT)





Fourier Amplitude Spectrum

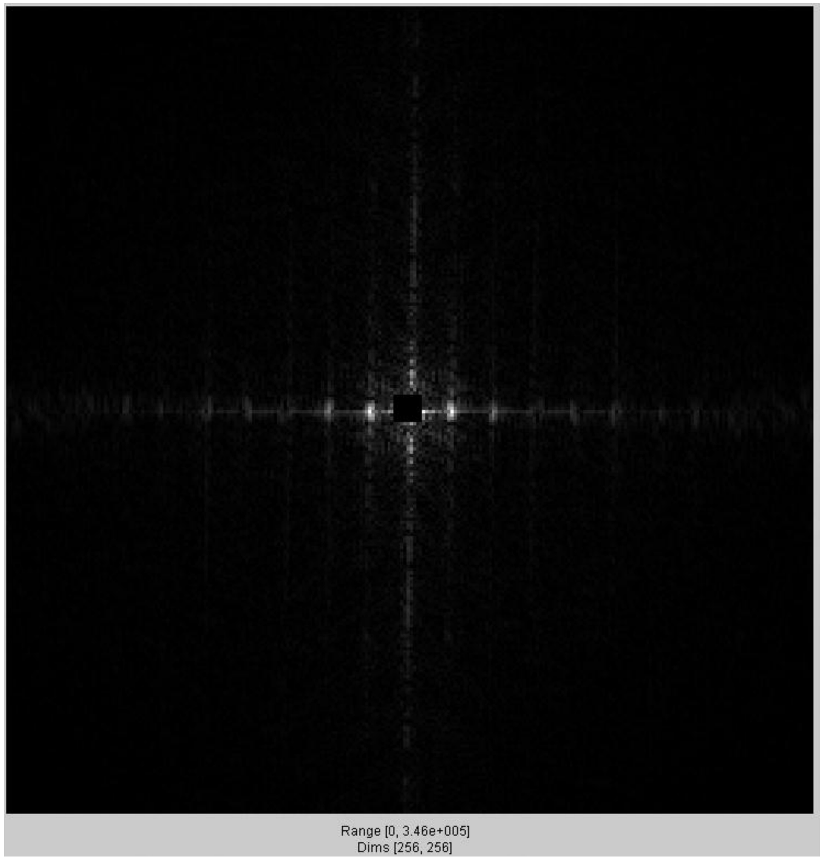


fx(cycles/image pixel size)

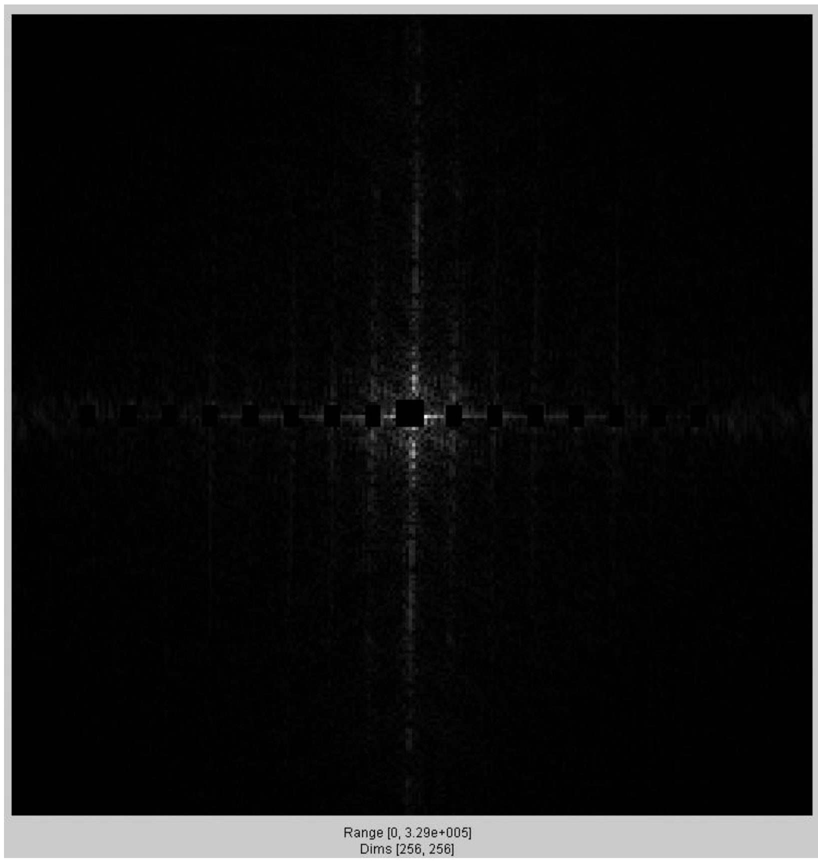
fx(cycles/image pixel size)

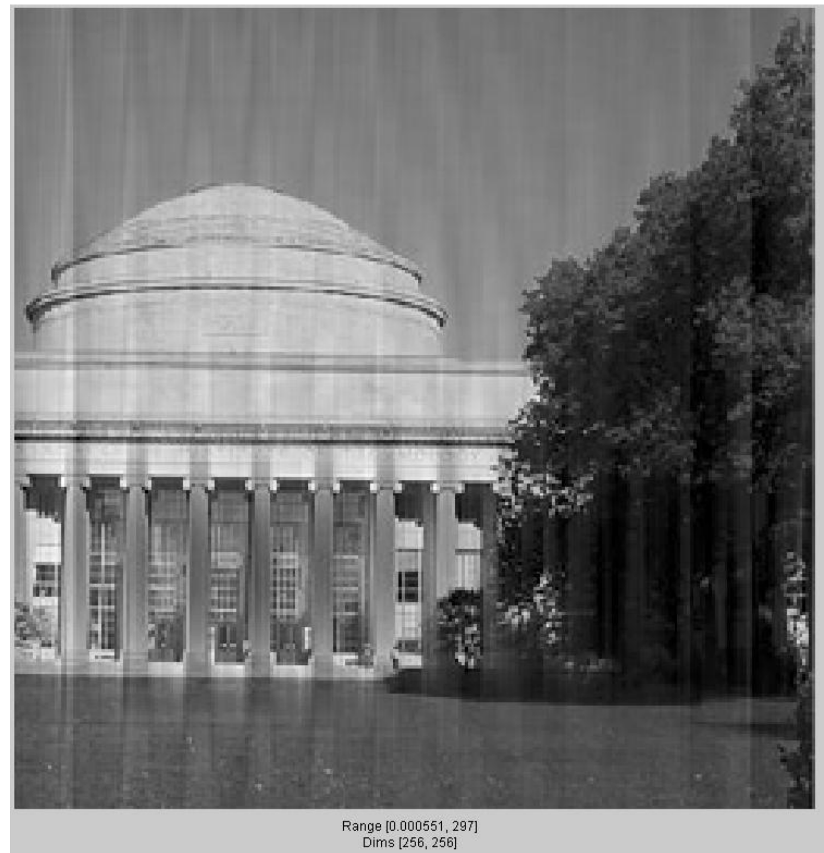
fx(cycles/image pixel size)

Fourier transform magnitude



Masking out the fundamental and harmonics from periodic pillars





Why is the Fourier domain particularly useful?

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

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

Fourier transform of convolution

Consider a (circular) convolution of g and h

$$f = g \otimes h$$

In the transform domain, this just modulates the transform amplitudes

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

Fourier transform of convolution

$$f = g \otimes h$$

Consider a (circular) convolution of g and h

$$F[m,n] = DFT(g \otimes h)$$

Take DFT of both sides

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

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

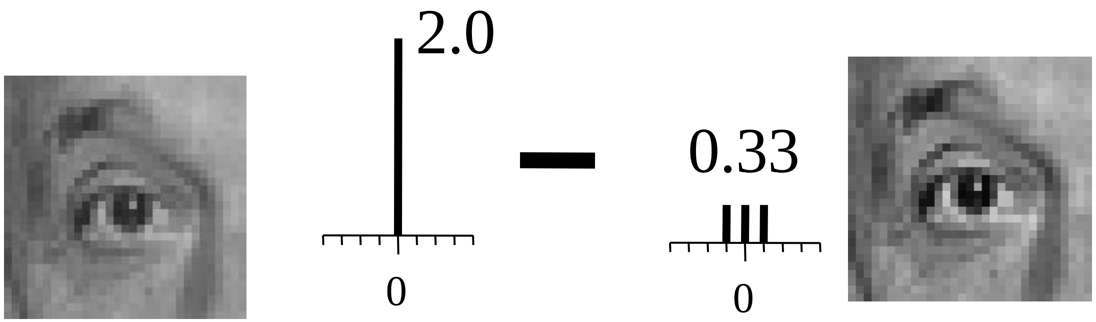
$$= \sum_{k,l} G[m,n] e^{-\pi i \left(\frac{km}{M} + \frac{\ln}{N} \right)} h[k,l]$$

Perform the DFT (circular boundary conditions)

$$= G[m,n] H[m,n]$$

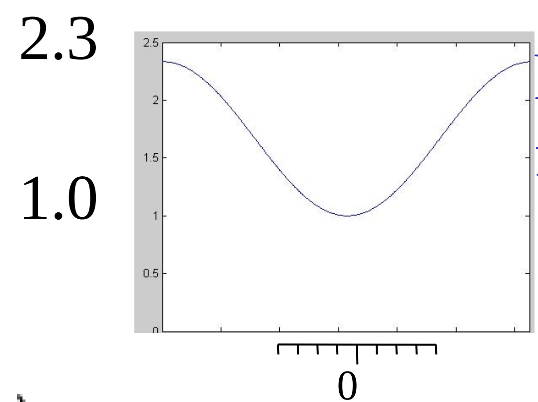
Perform the other DFT (circular boundary conditions)

Analysis of a simple sharpening filter



original

sharpened



high-pass filter

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

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

Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

Phase and Magnitude

Image with cheetah phase
(and zebra magnitude)

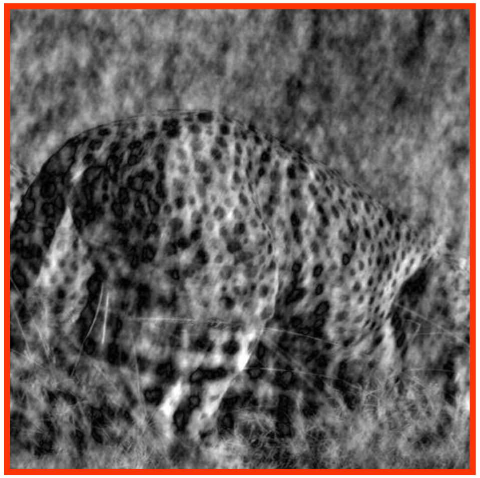
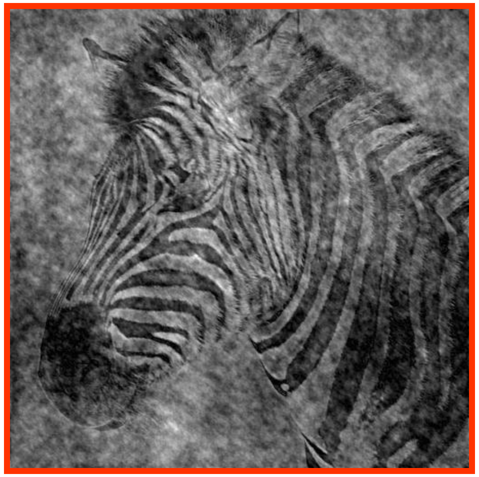


Image with zebra phase
(and cheetah magnitude)



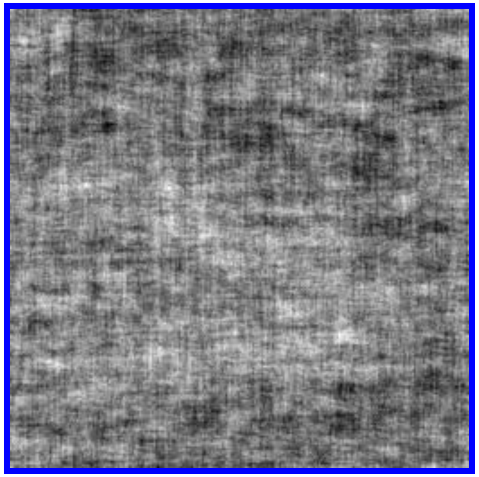
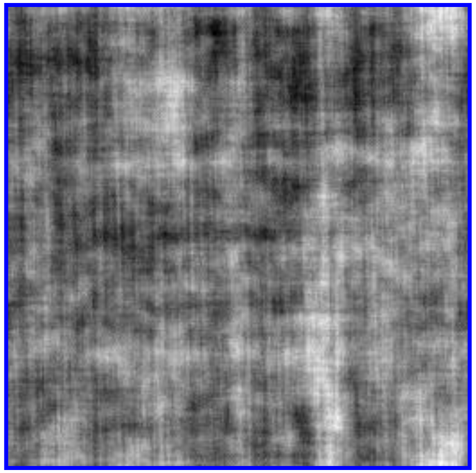
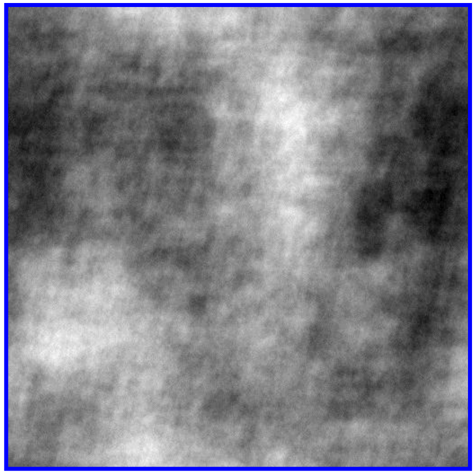
Phase and Magnitude

- 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?

Randomizing the phase



Fourier transform, randomize the phase, inverse transform

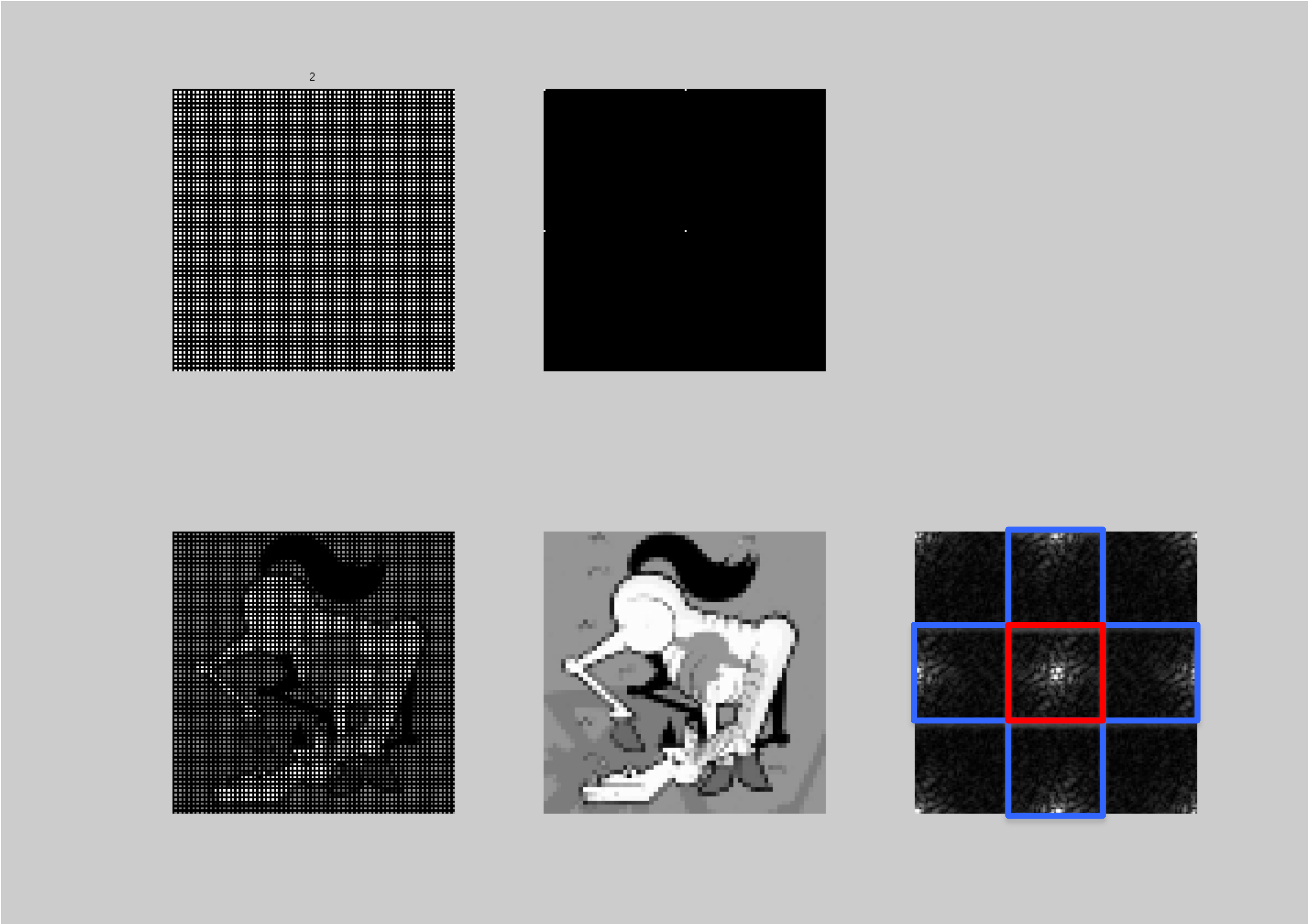




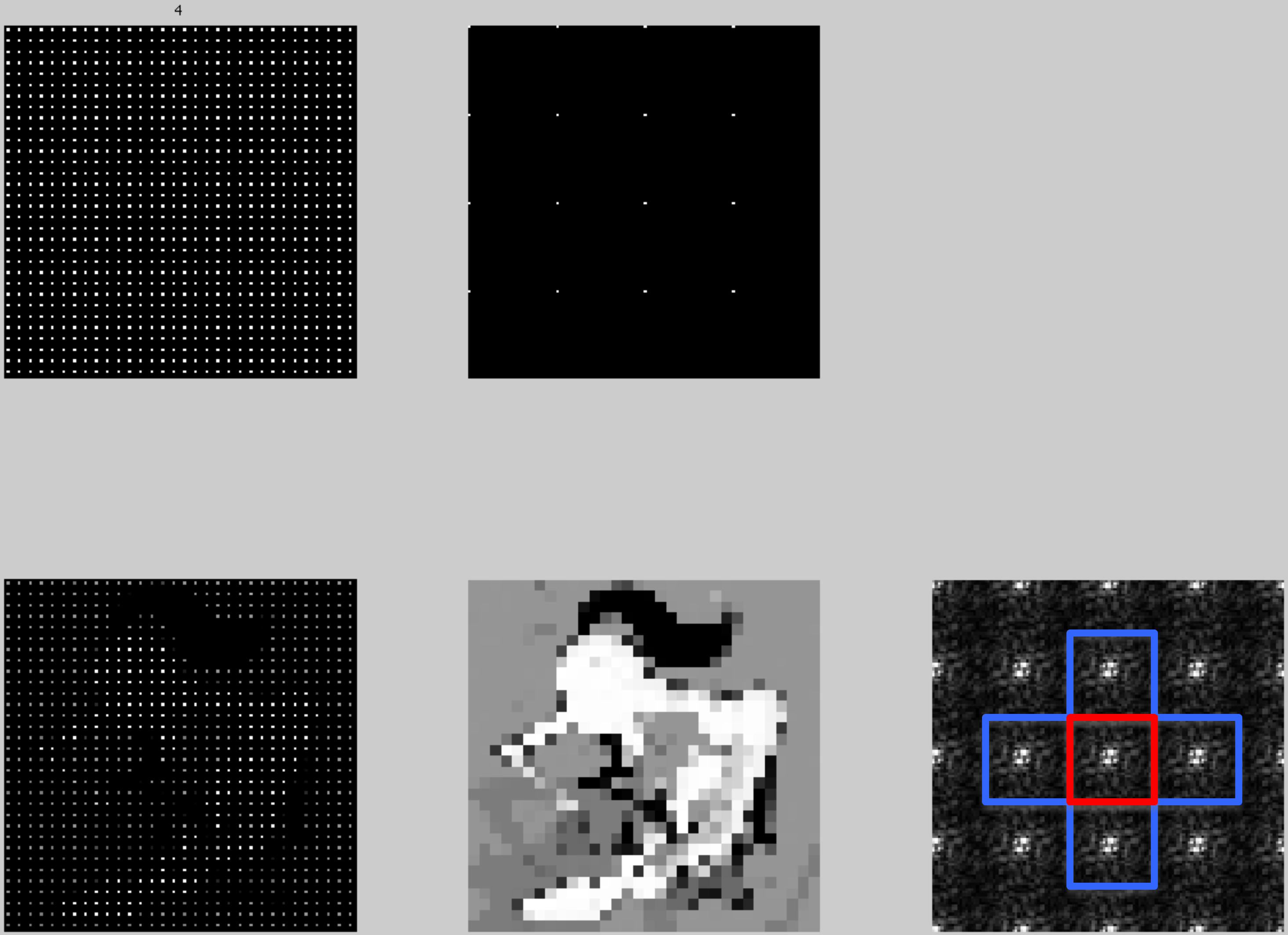
Outline

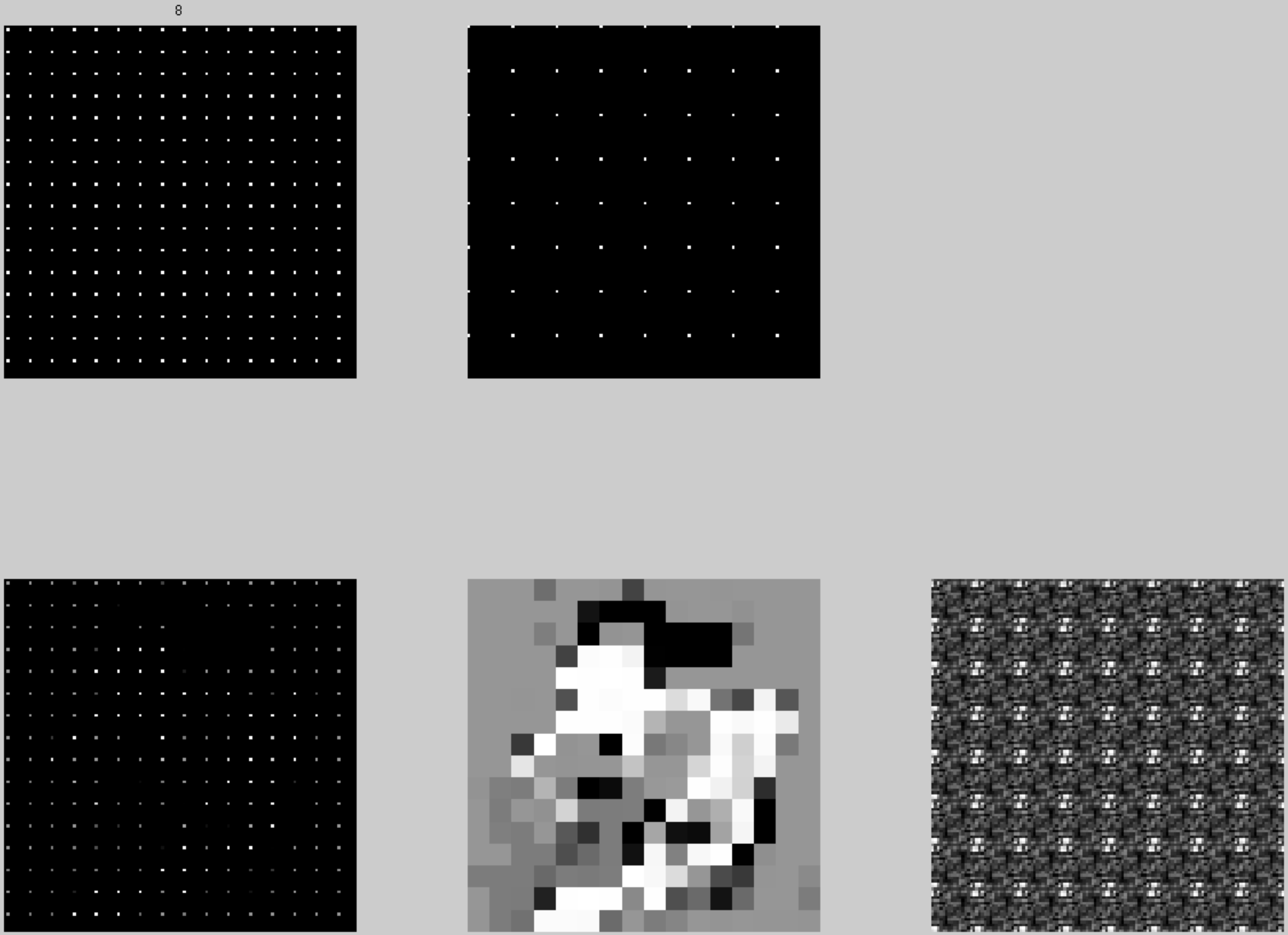
- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

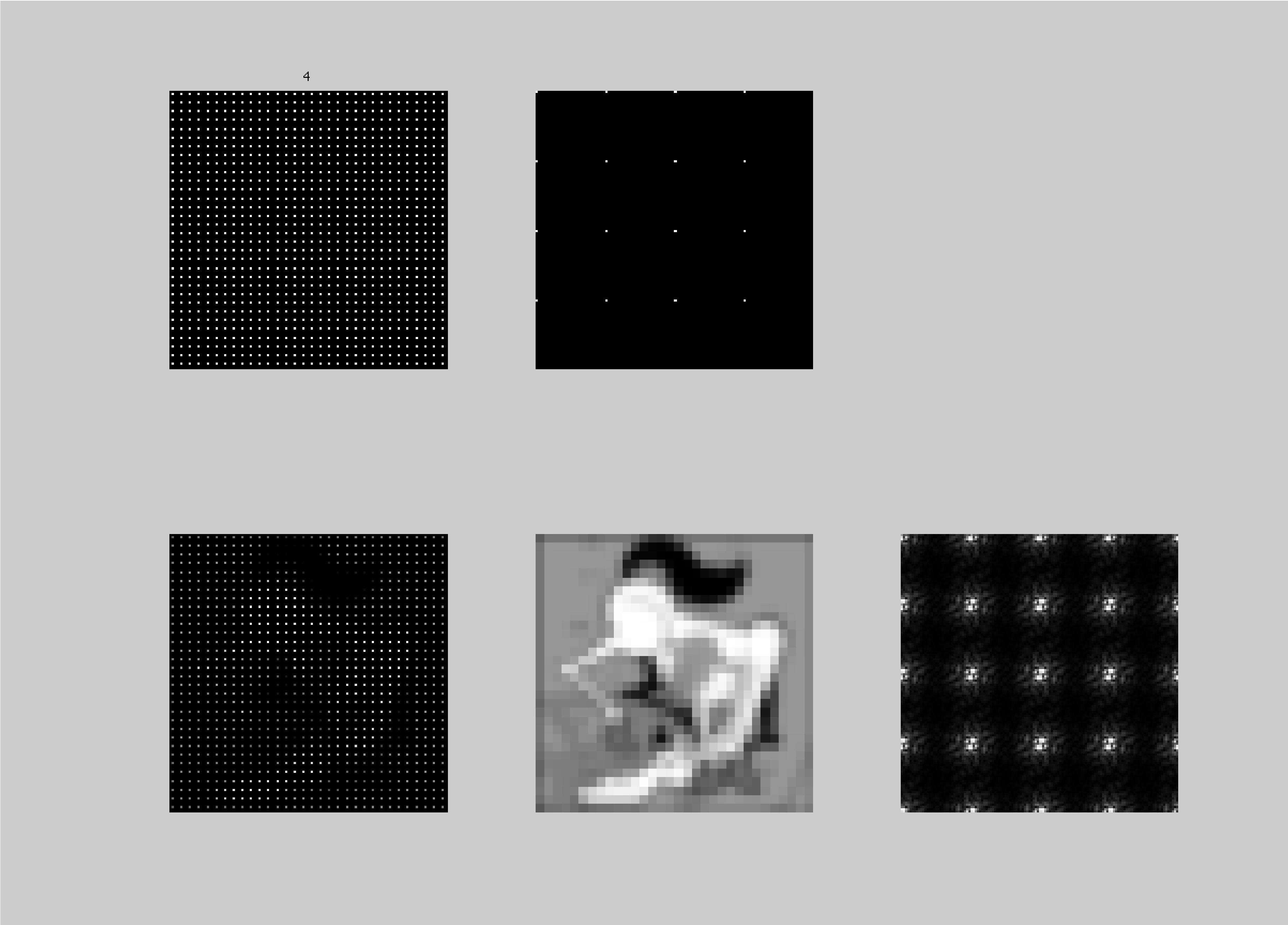
The Fourier transform of a sampled signal

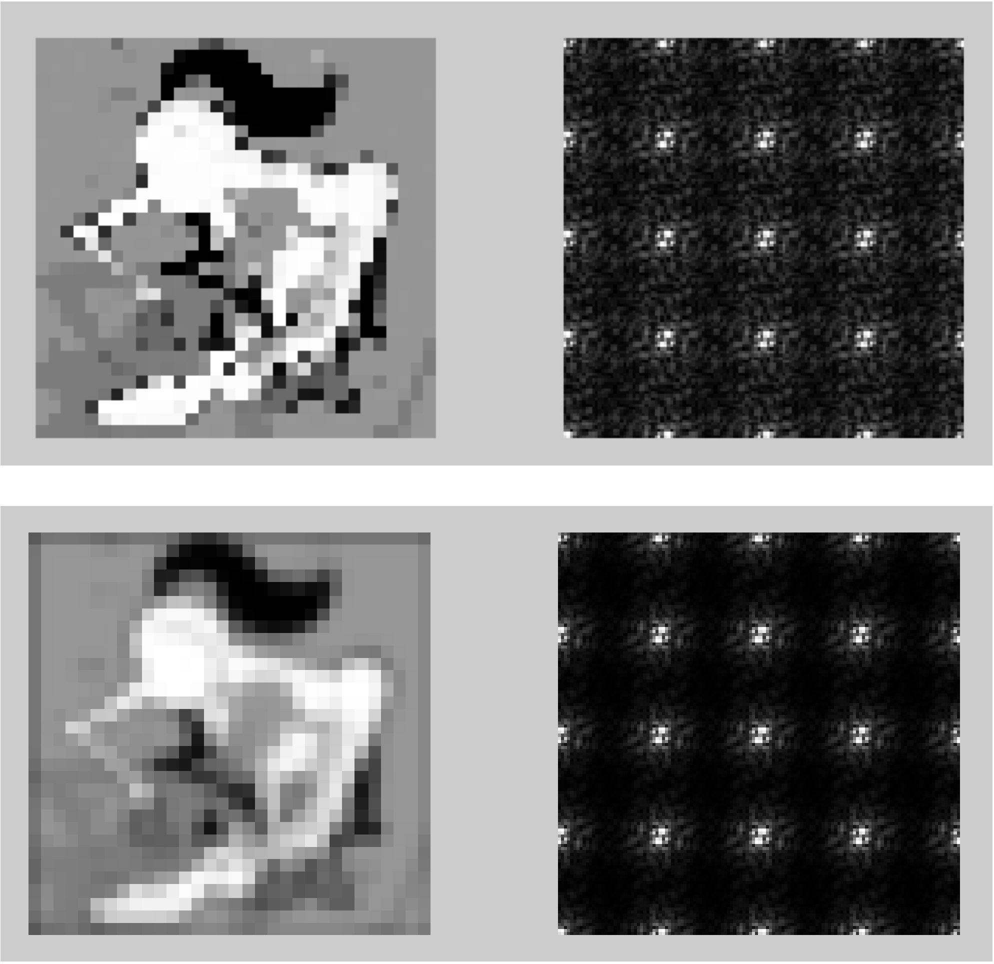
$$
\begin{aligned}
F\left(\text{Sample}_{2D}\left(f(x,y)\right)\right) &= F\left(f(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \right) \\
&= F\left(f(x,y)\right) ** F\left(\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \right) \\
&= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F\left(u-i, v-j\right)
\end{aligned}
$$


2


4

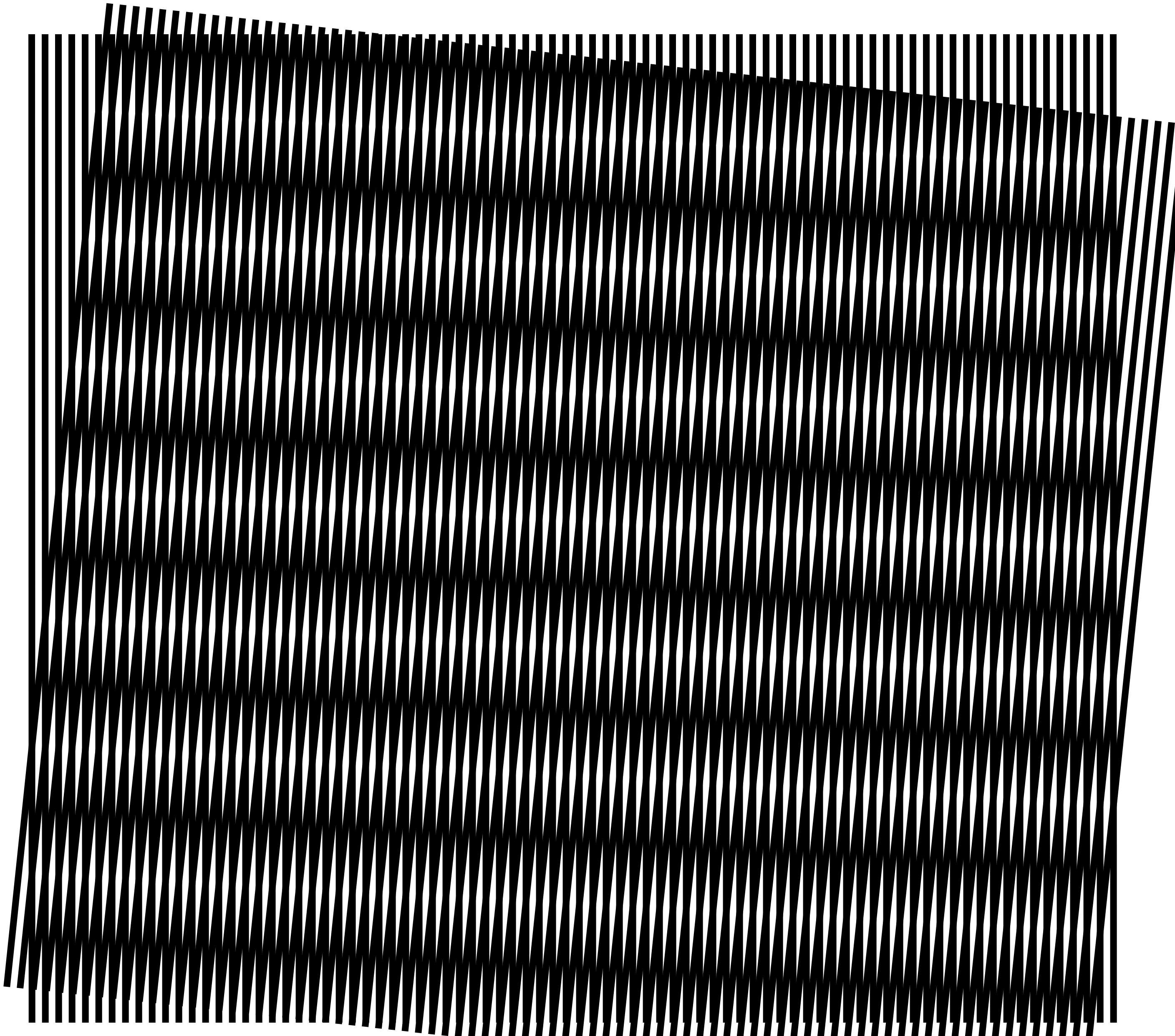

8


4



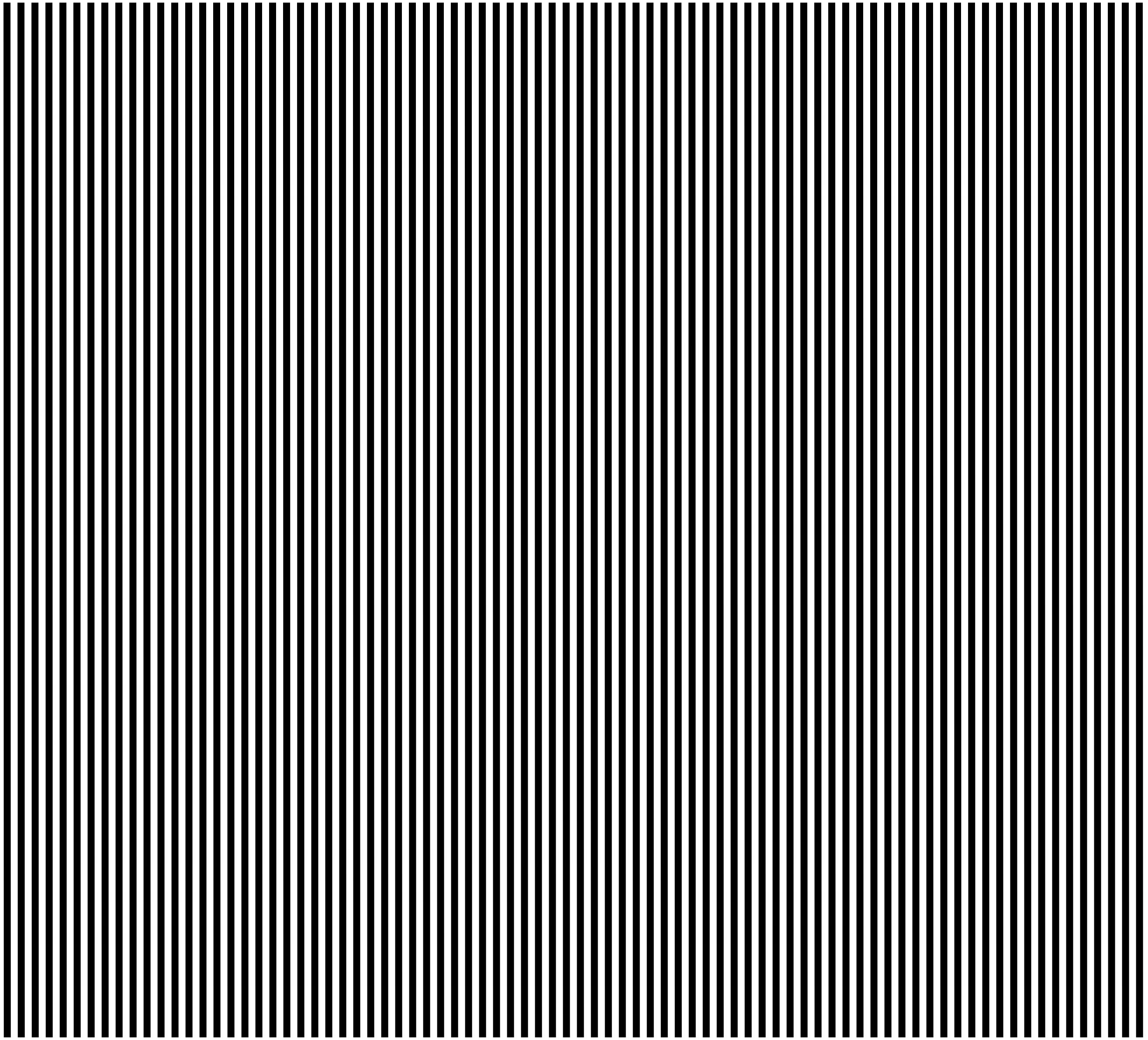
Sampling example

Analyze crossed gratings…



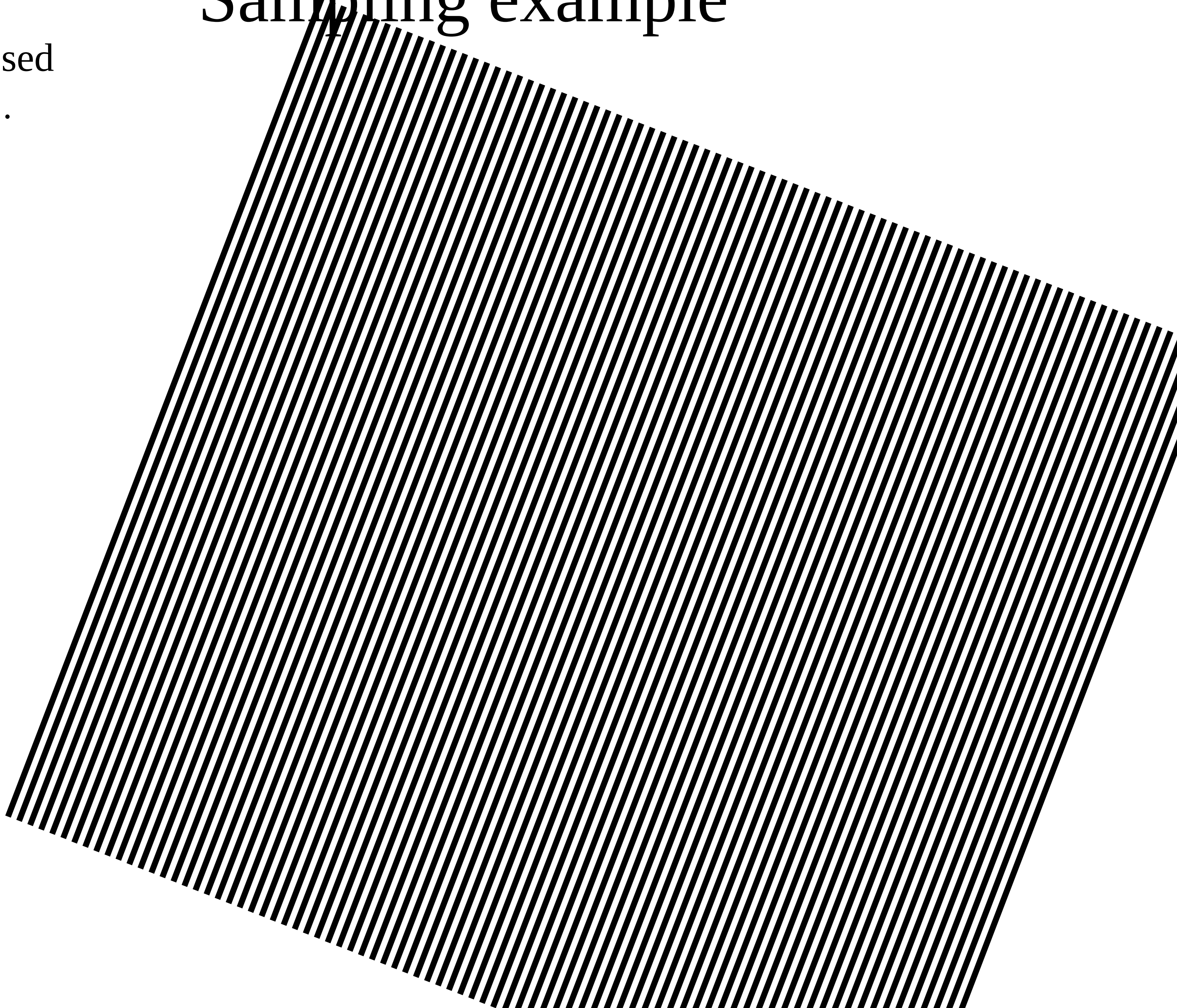
Sampling example

Analyze crossed gratings…



Sampling example

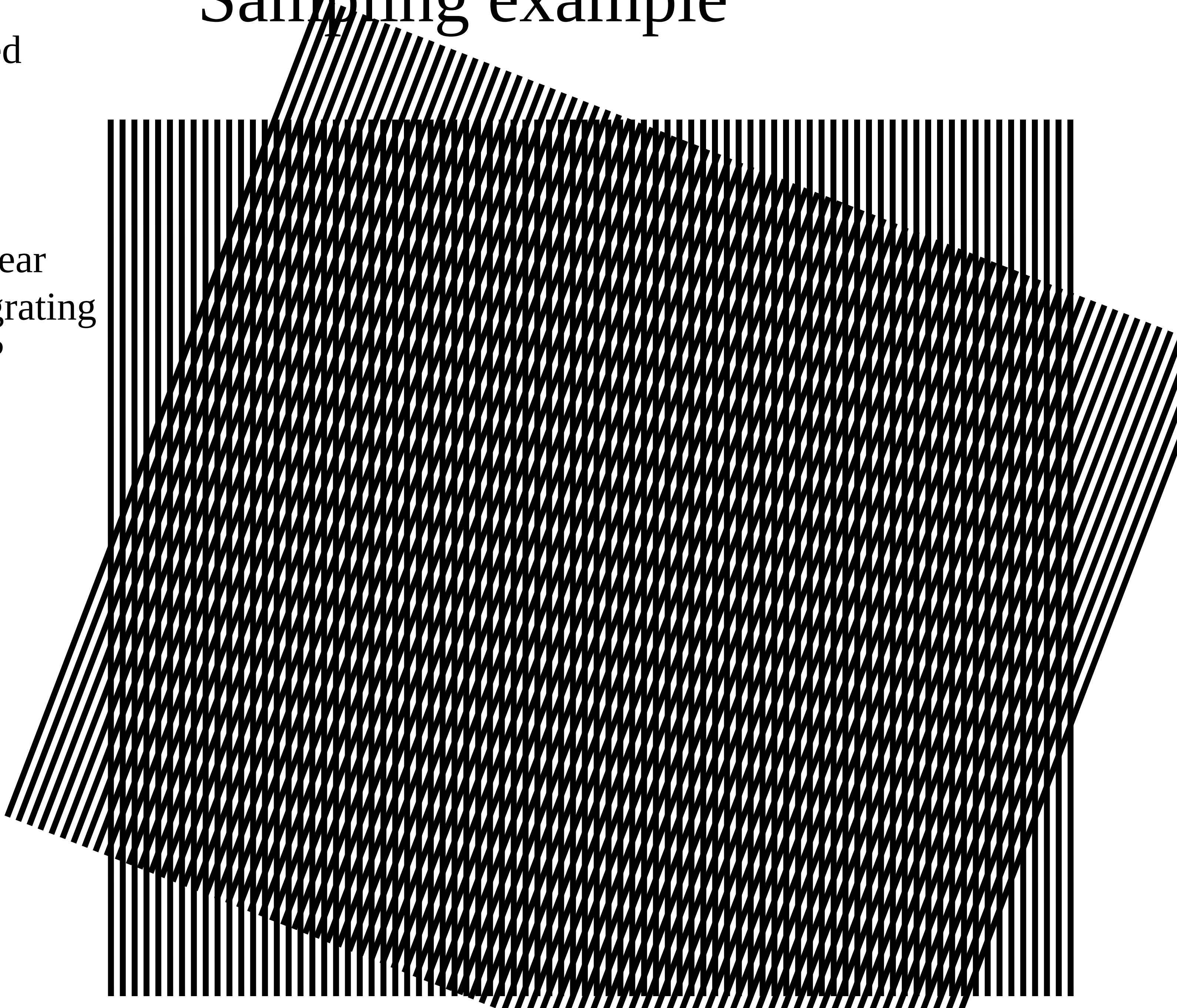
Analyze crossed gratings…

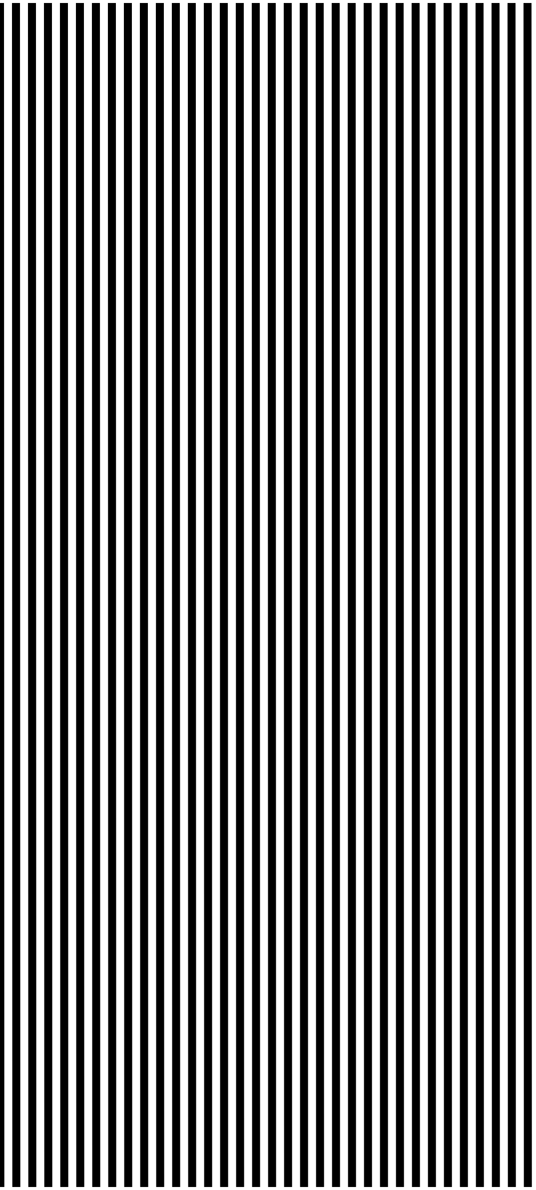


Sampling example

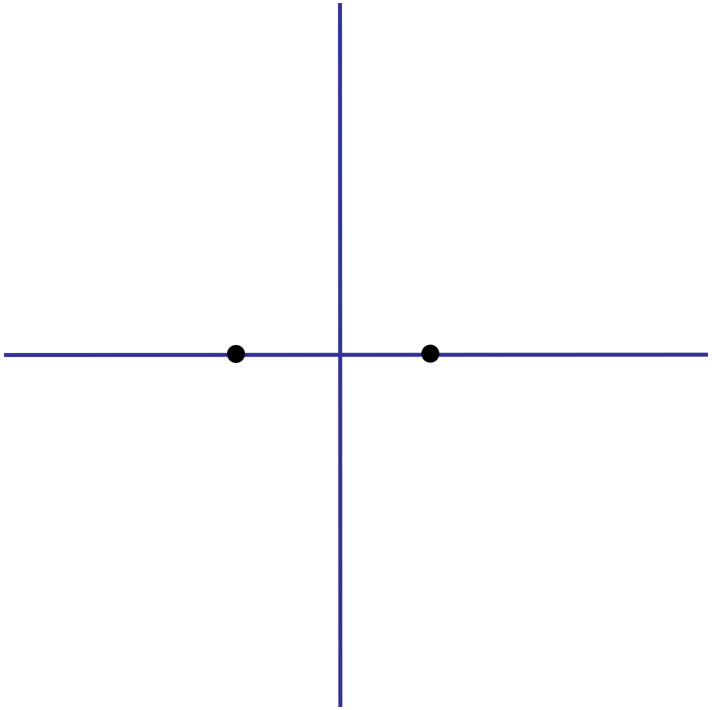
Analyze crossed gratings…

Where does perceived near horizontal grating come from?

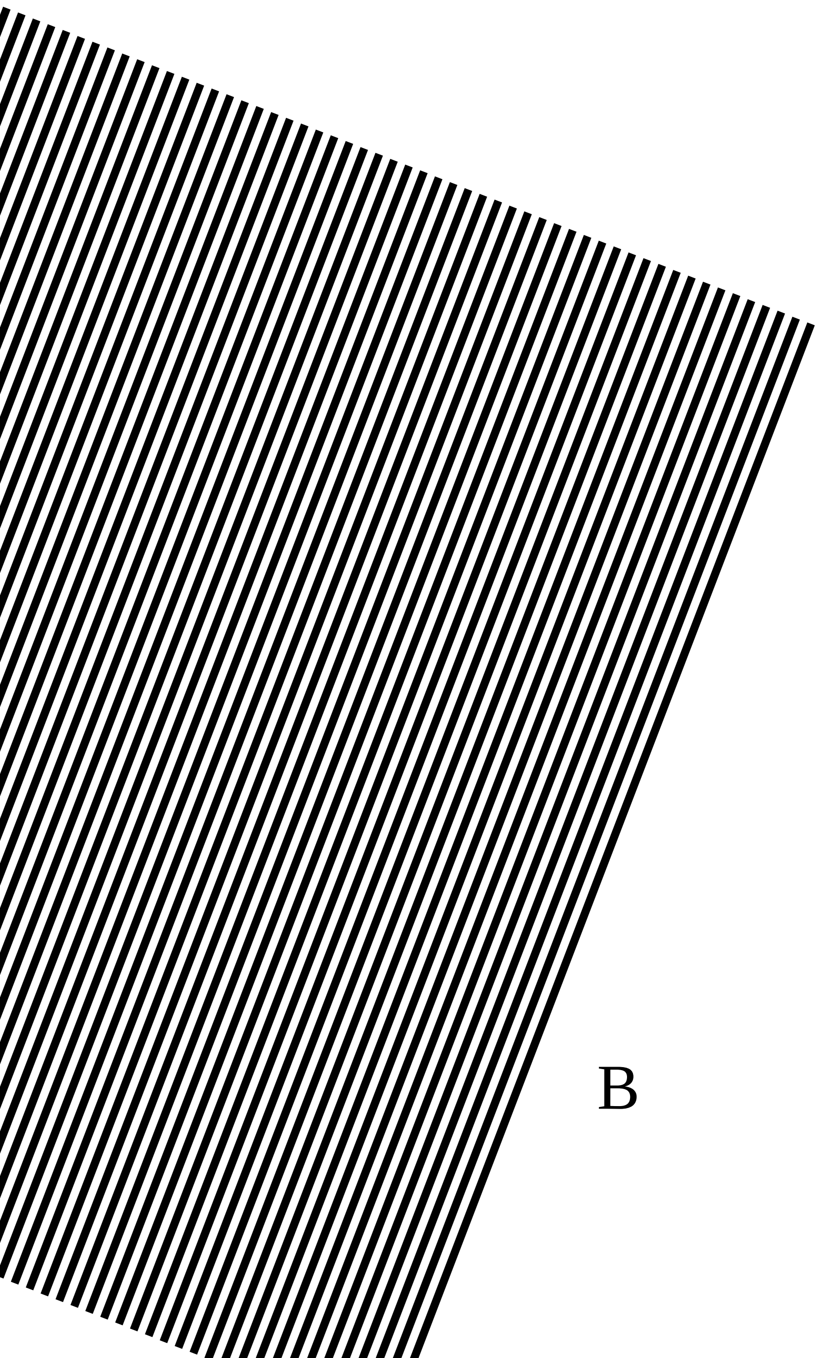




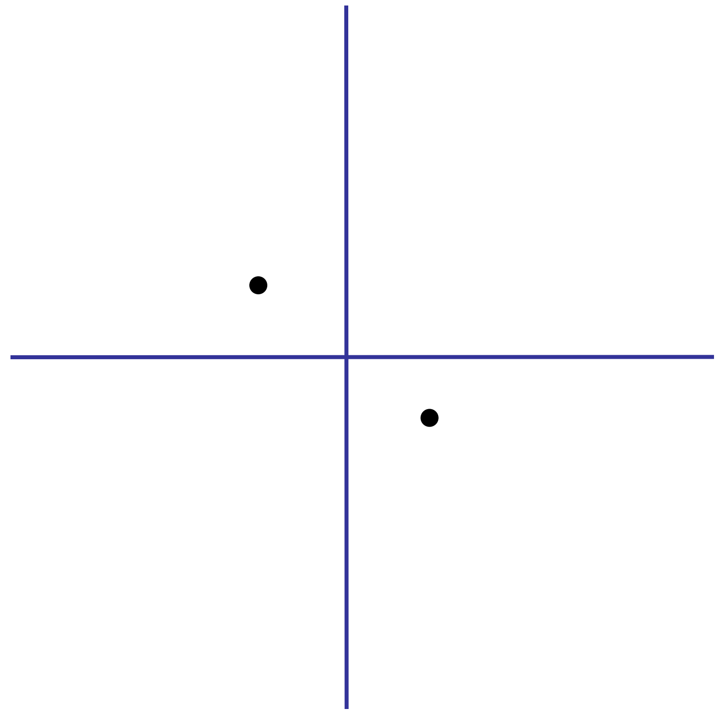
A



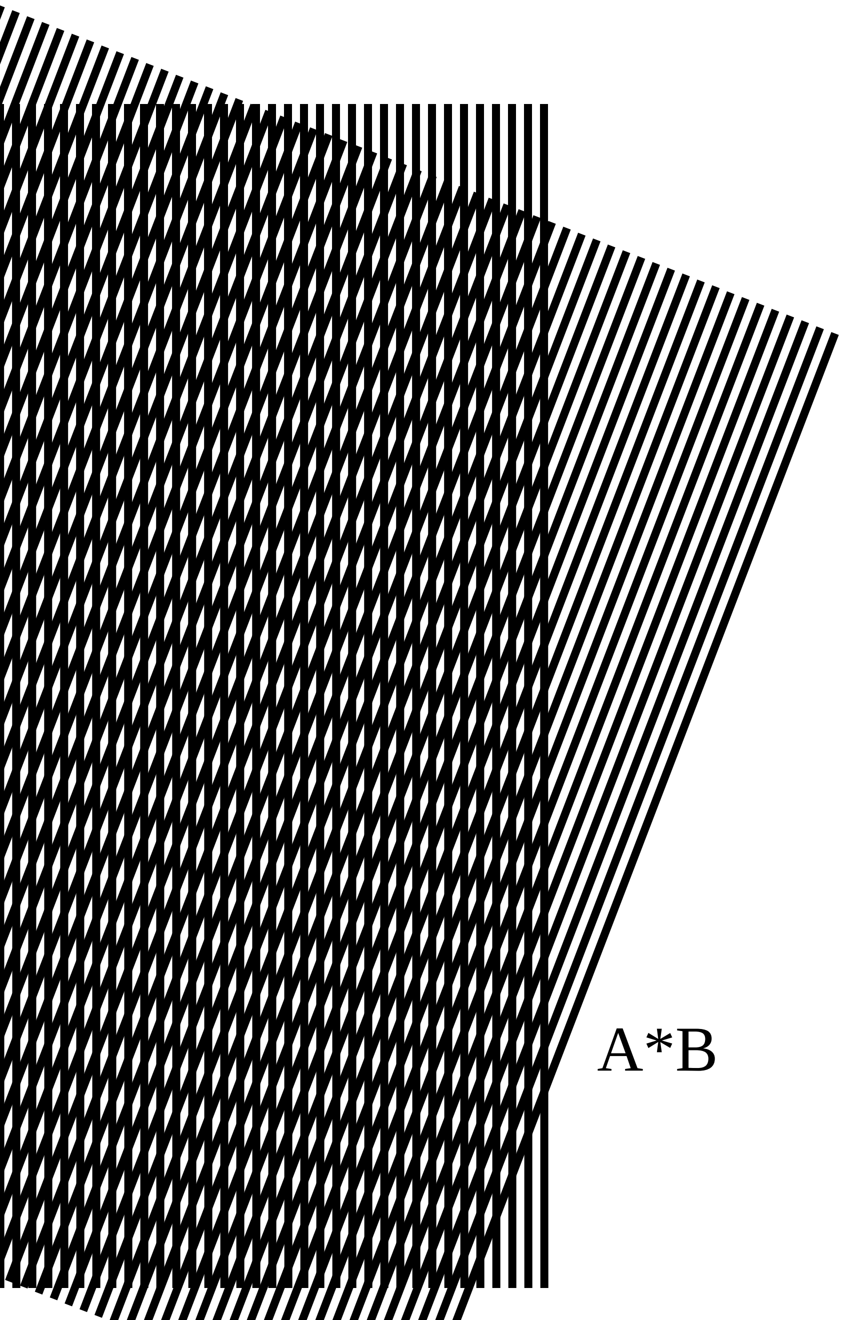
F(A)



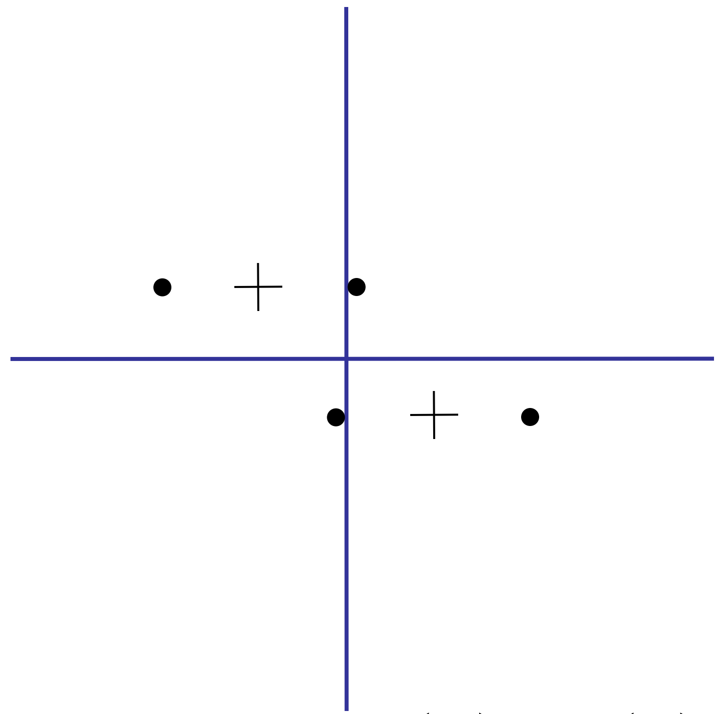
B



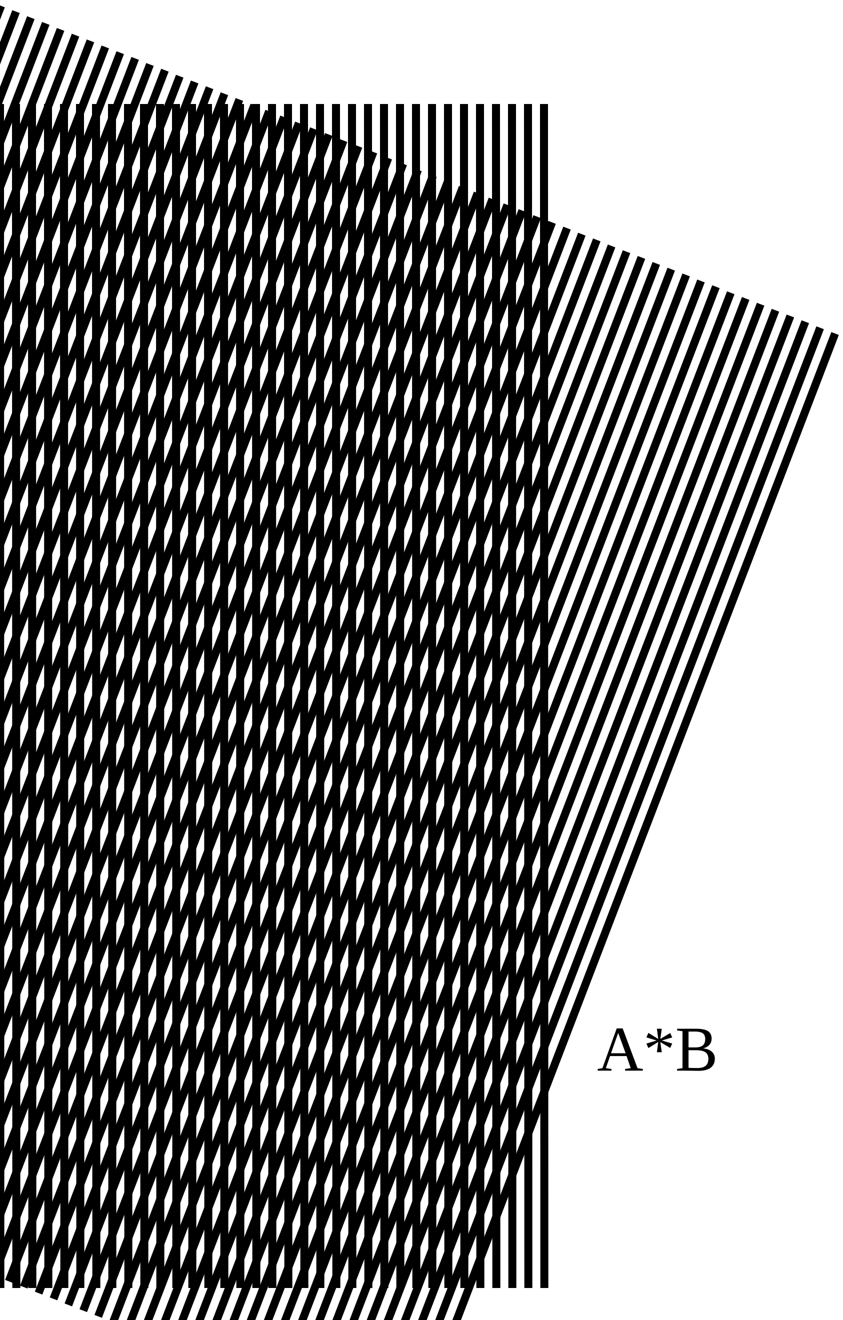
F(B)



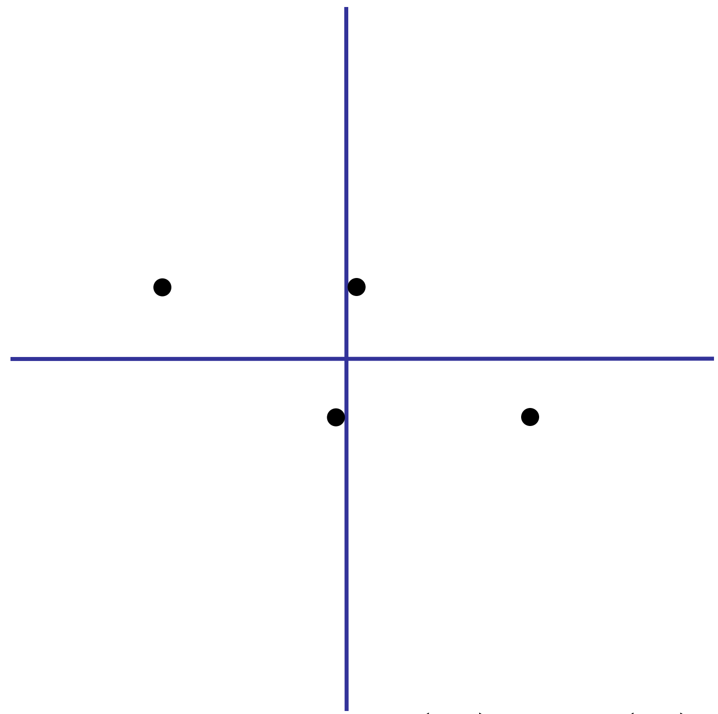
A*B



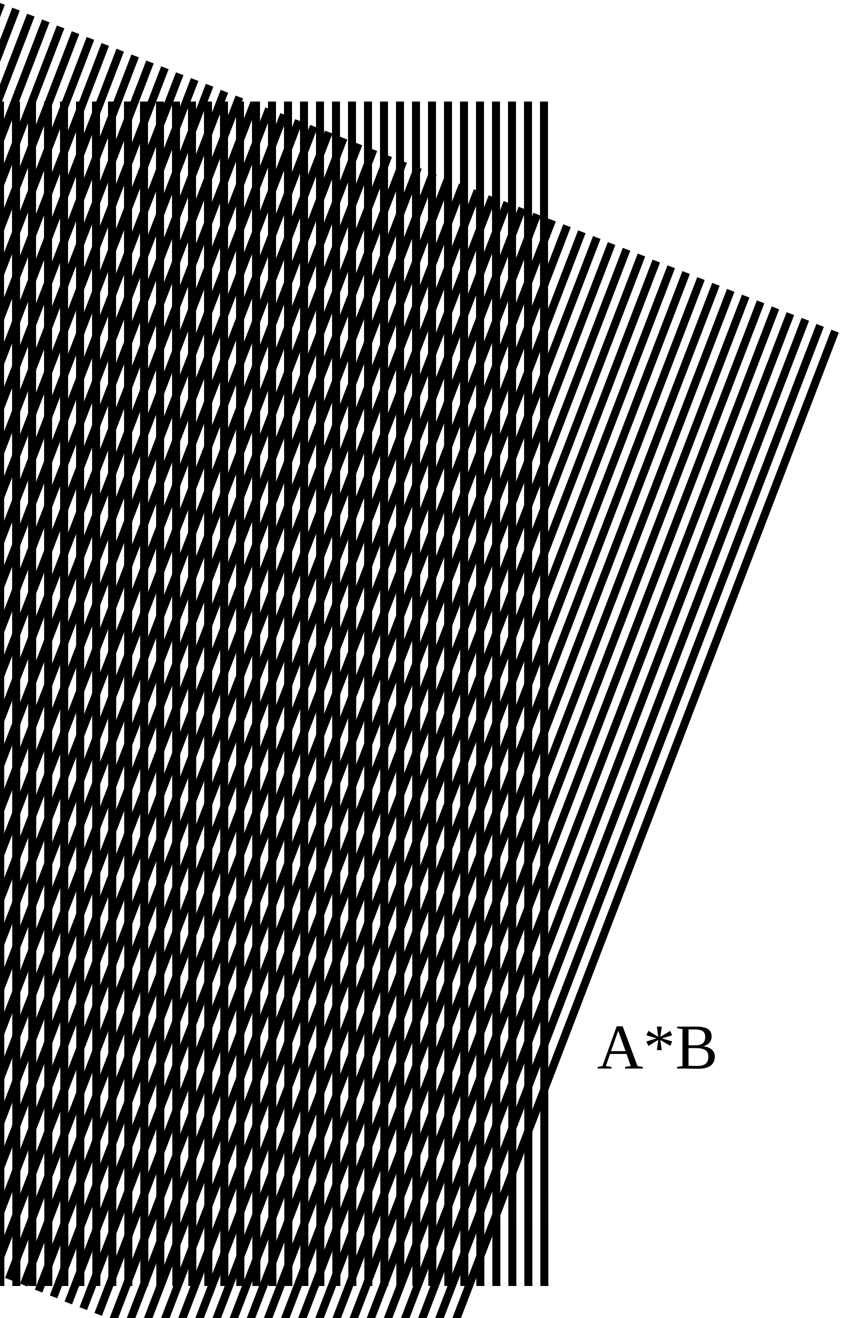
F(A) $\otimes$ F(B)



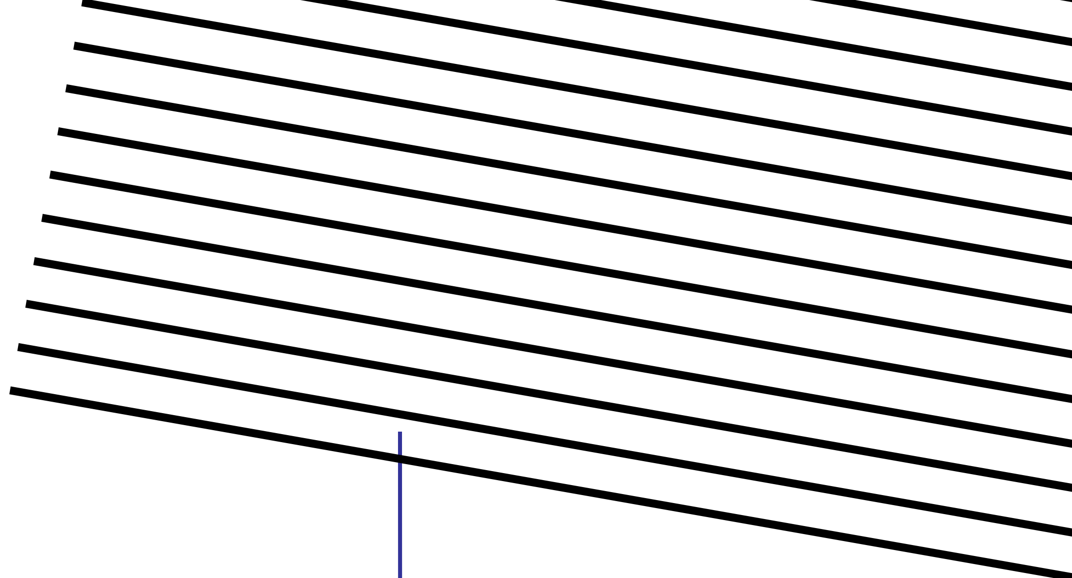
A*B



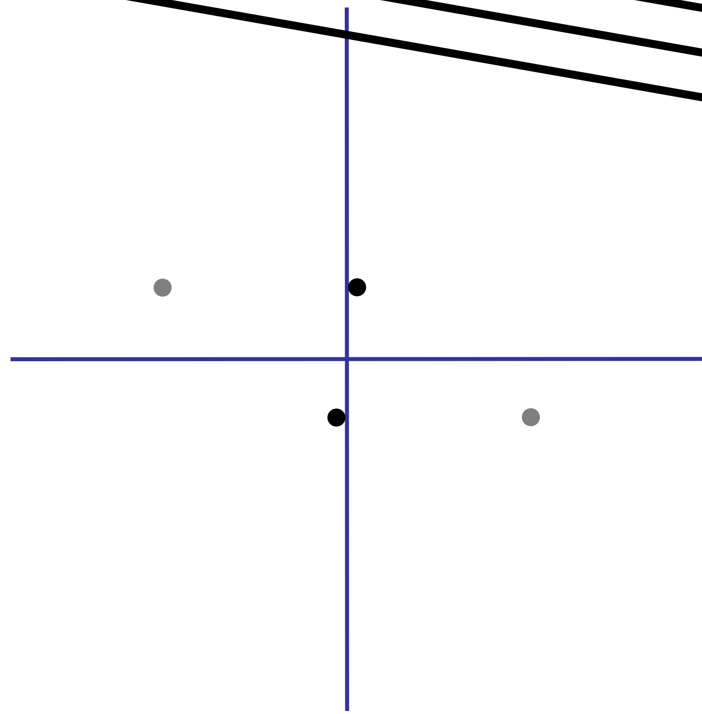
$F(A) \otimes F(B)$



A*B



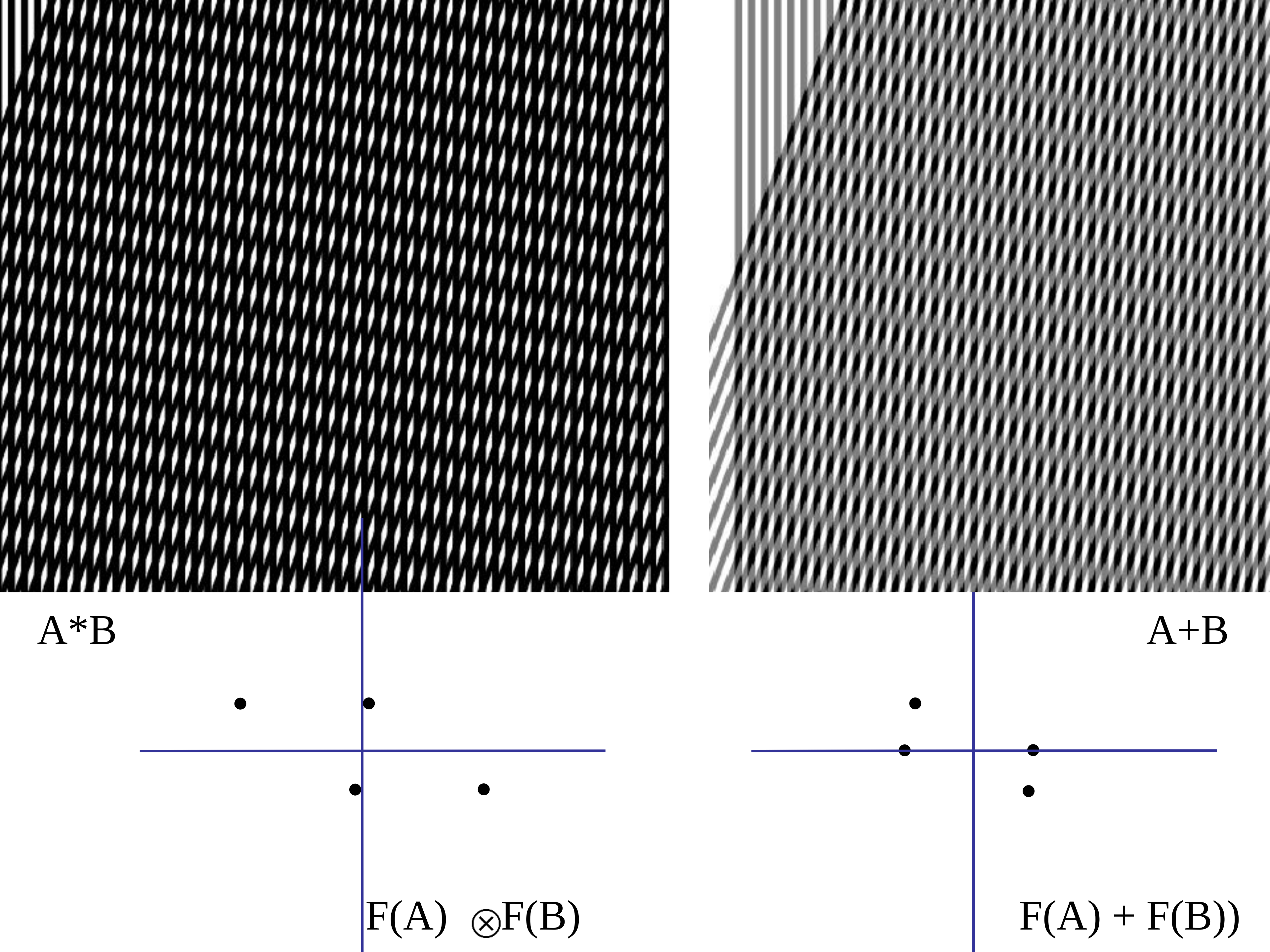
C



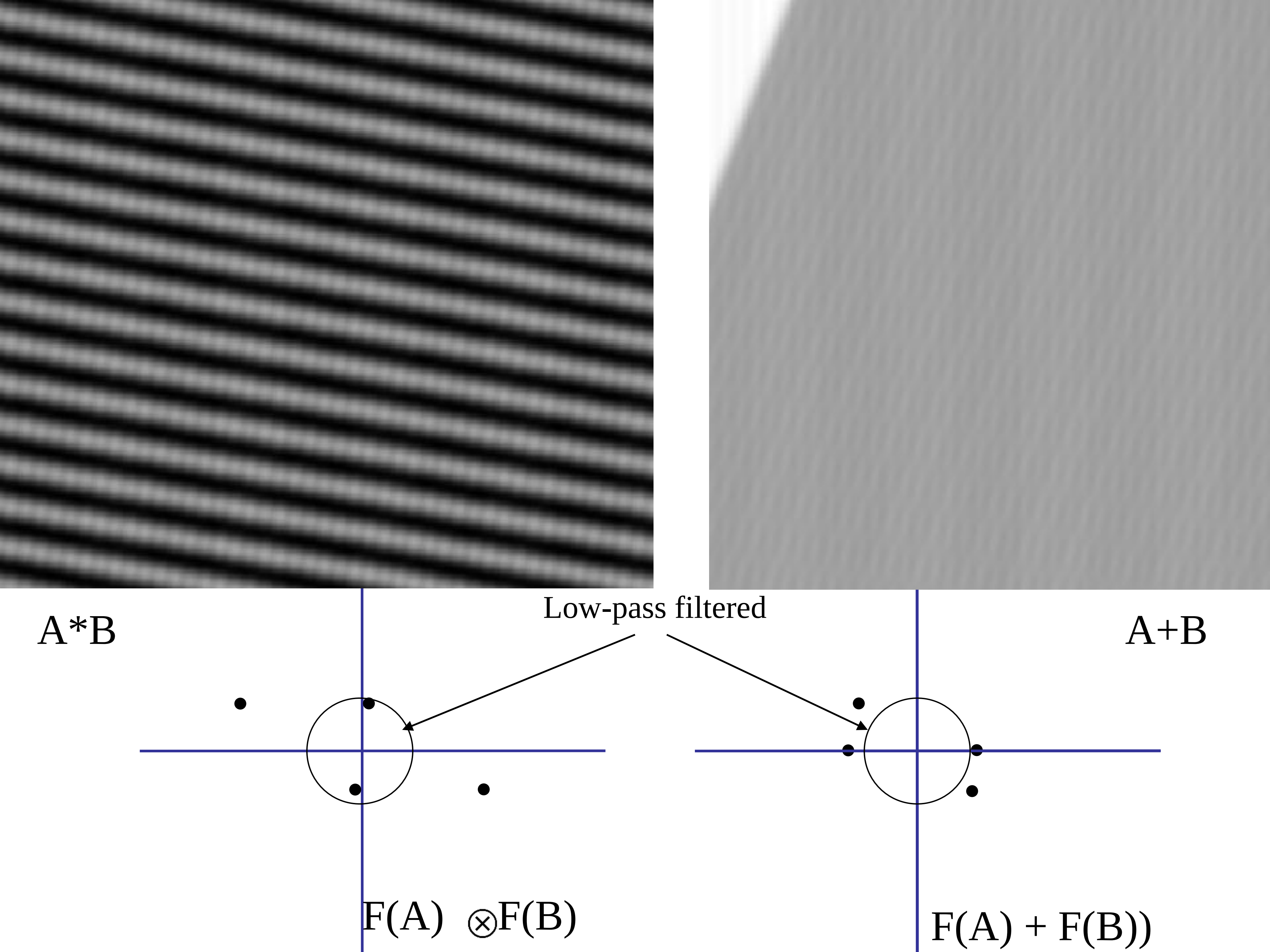
Lowpass(F(A) $\otimes$F(B))
~=F(C)

Control test

- If our analysis is correct, if we *add* those two sinusoids (or square waves), and if there is no non-linearity in the display of the sum, then there should only be summing, not convolution, in the frequency domain.



A*B
F(A) ⊗F(B)
A+B
F(A) + F(B))


Low-pass filtered
A*B
A+B
F(A) ⊗F(B)
F(A) + F(B))

Problems with Fourier transform as an image representation

Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

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.

Analyzing local image structures

Too much

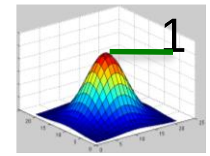
Too little

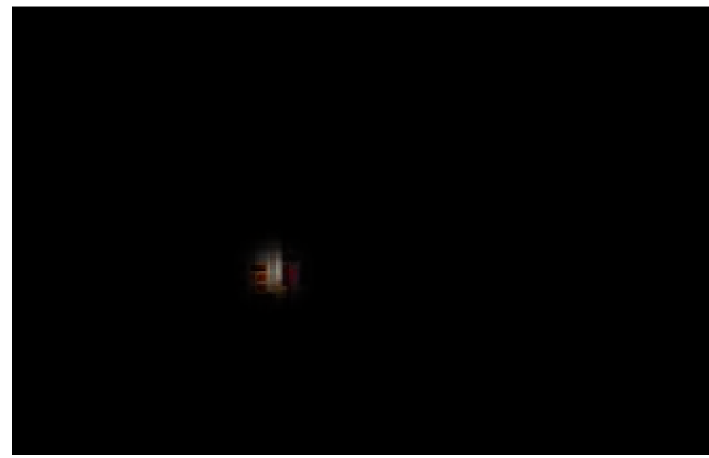
The image through the Gaussian window

Too much

Too little

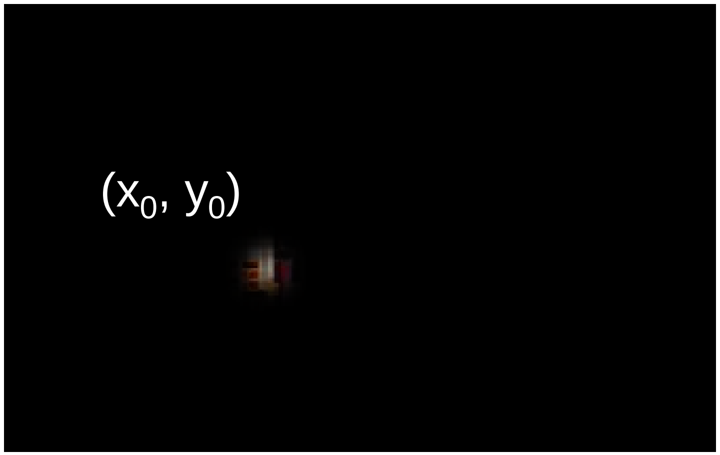
$$h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$



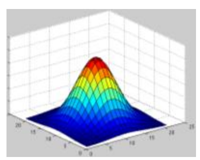


Probably still too little…
…but hard enough for now

Analysis of local frequency



$$h(x,y;x_0,y_0) = e^{-\frac{(x-x_o)^2+(y-y_o)^2}{2\sigma^2}}$$



Fourier basis:

$$e^{j2\pi u_0 x}$$

Gabor wavelet:

$$\psi(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j2\pi u_0 x}$$

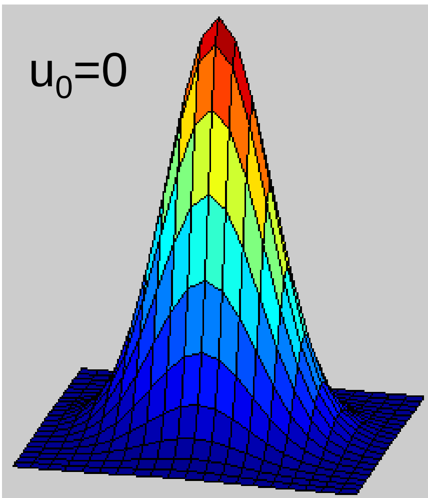
We can look at the real and imaginary parts:

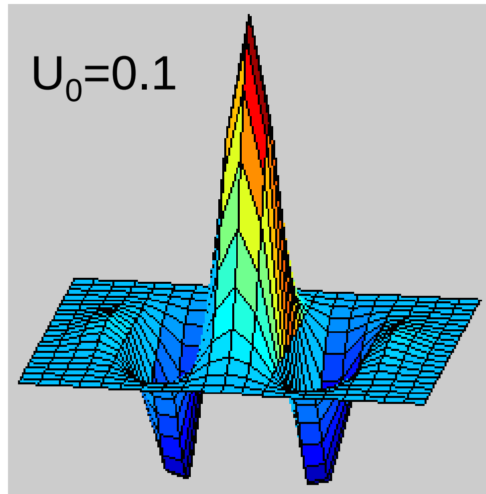
$$\psi_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi u_0 x)$$

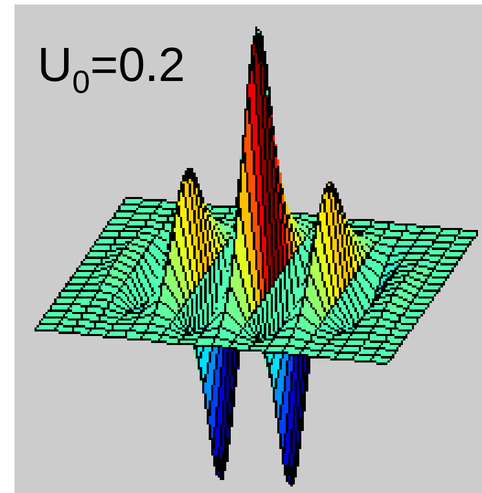
$$\psi_s(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \sin(2\pi u_0 x)$$

Gabor wavelets

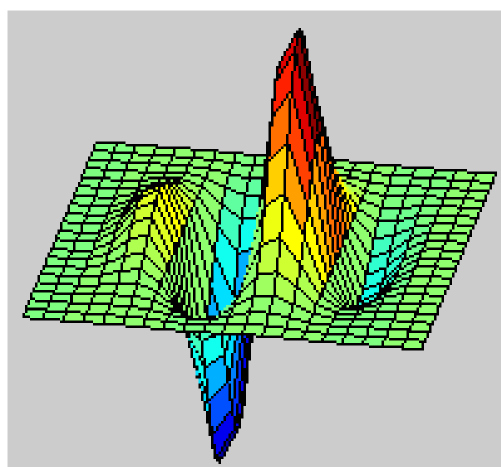
$$\psi_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi u_0 x)$$

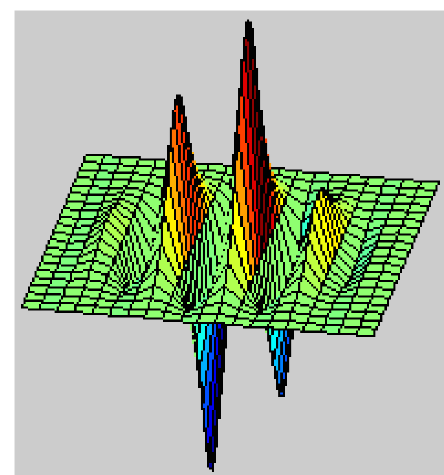


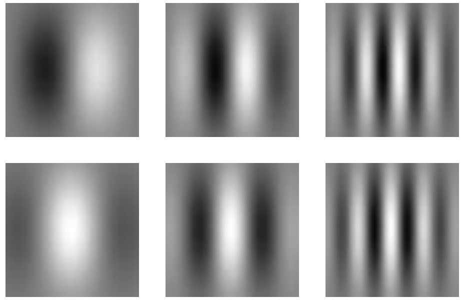




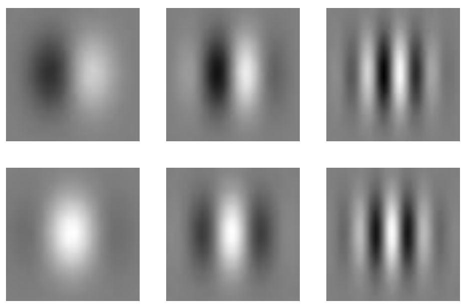
$$\psi_s(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \sin(2\pi u_0 x)$$







Gabor filters at different scales and spatial frequencies



Top row shows anti-symmetric (or odd) filters; these are good for detecting odd-phase structures like edges.
Bottom row shows the symmetric (or even) filters, good for detecting line phase contours.

John Daugman, 1988

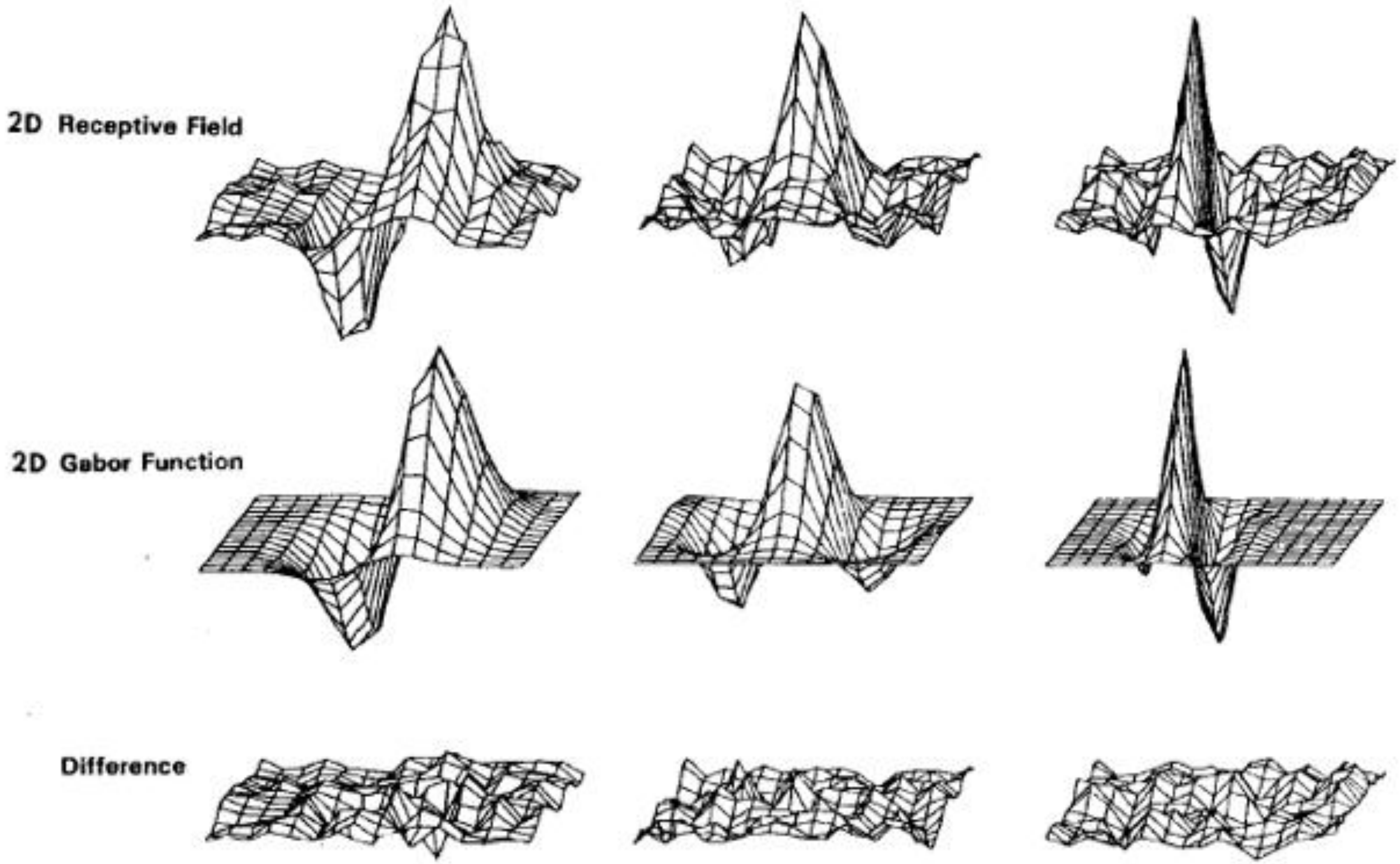
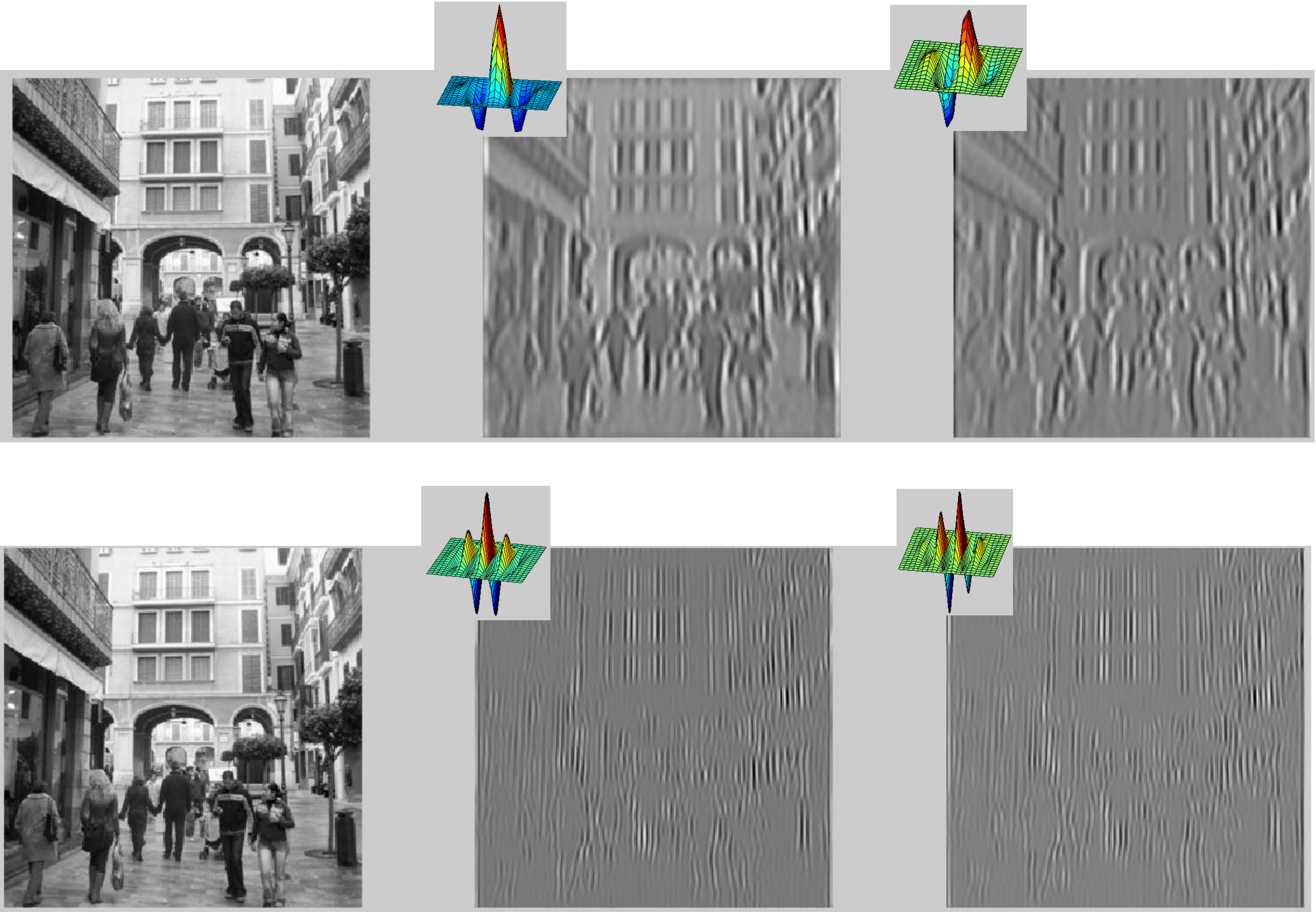


Fig. 5. Top row: illustrations of empirical 2-D receptive field profiles measured by J. P. Jones and L. A. Palmer (personal communication) in simple cells of the cat visual cortex. Middle row: best-fitting 2-D Gabor elementary function for each neuron, described by (10). Bottom row: residual error of the fit, indistinguishable from random error in the Chi-squared sense for 97 percent of the cells studied.



Outline

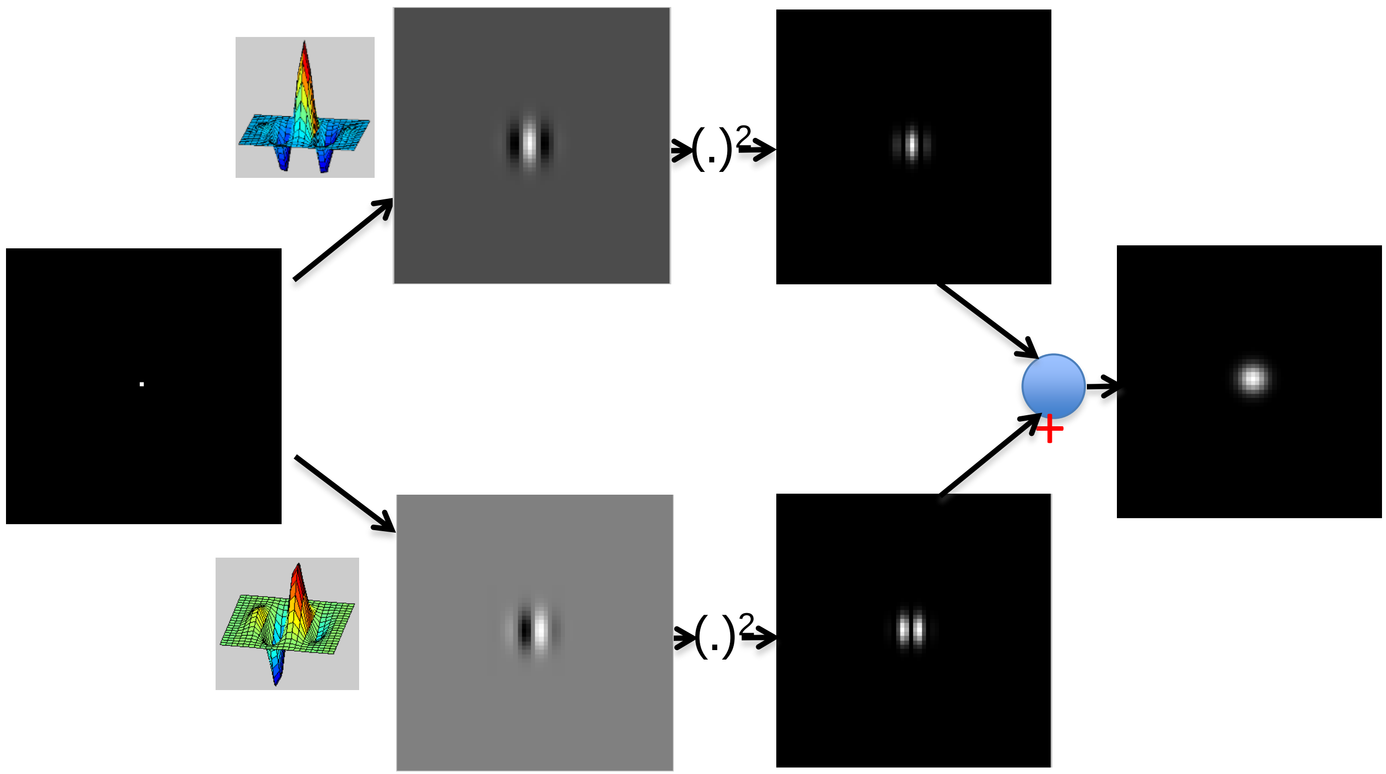
- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

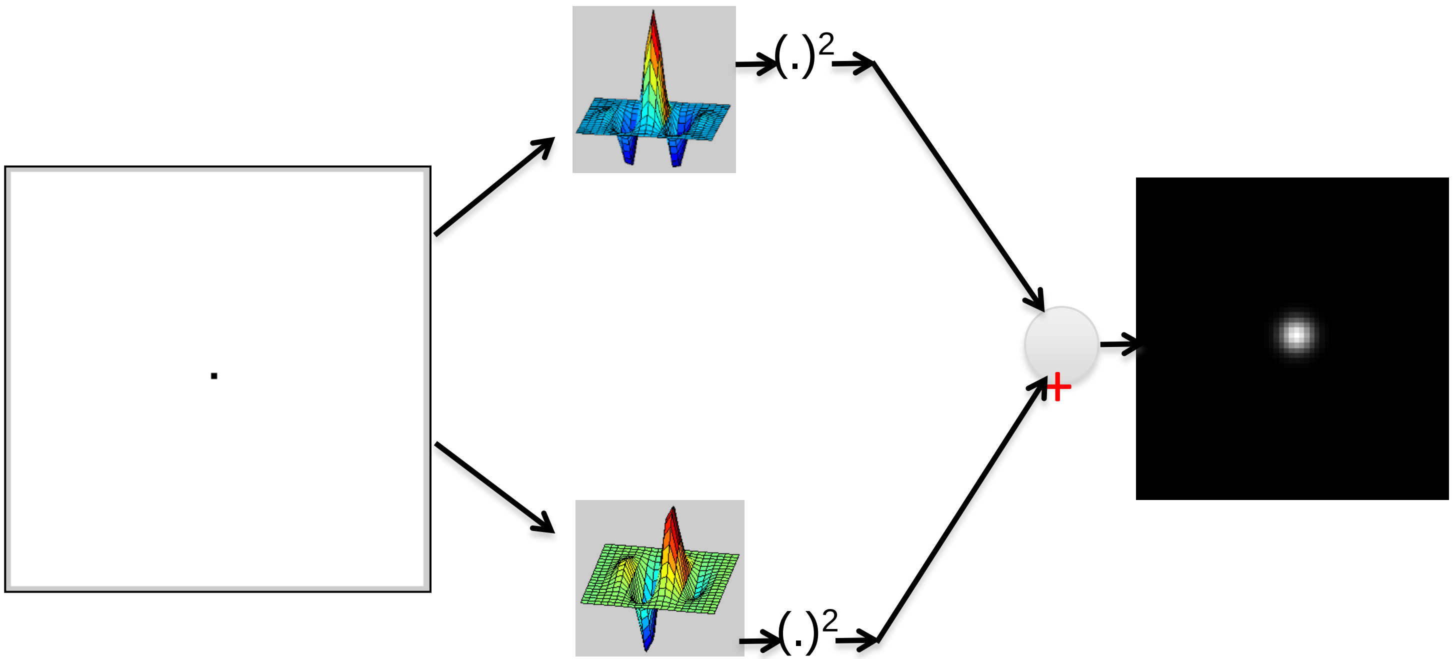
Quadrature filter pairs

A quadrature filter is a complex filter whose real part is related to its imaginary part via a Hilbert transform along a particular axis through the origin

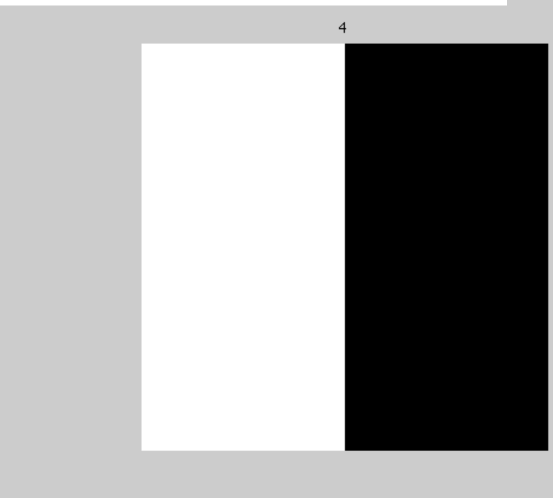
Gabor wavelet:

$$\psi(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j 2\pi u_0 x}$$

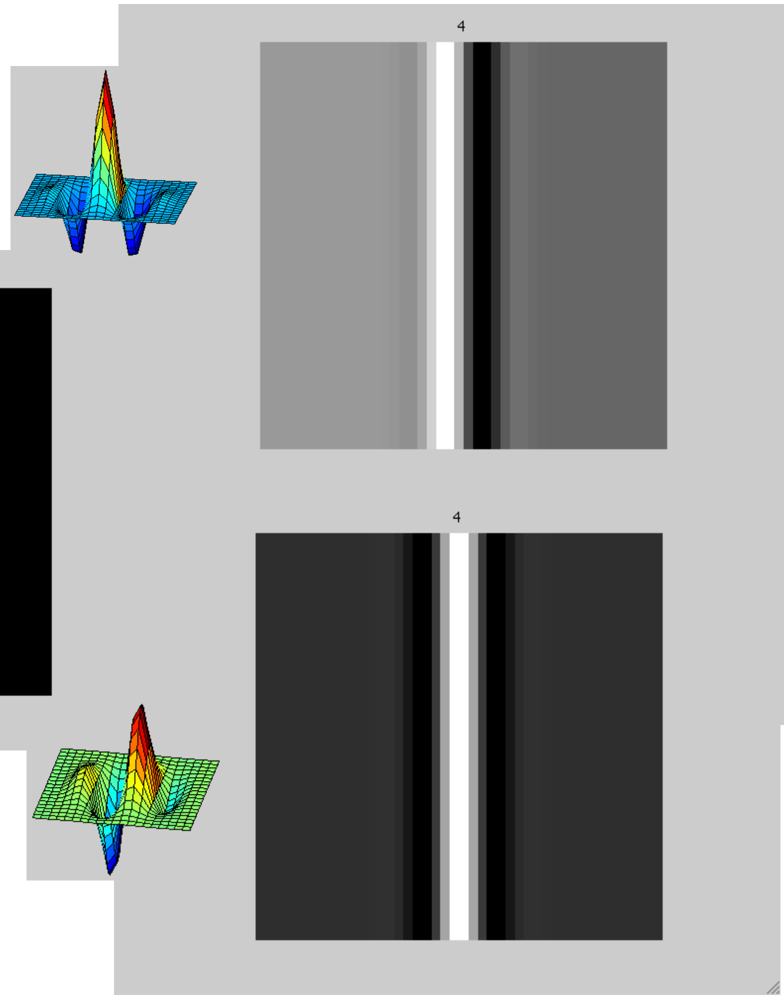


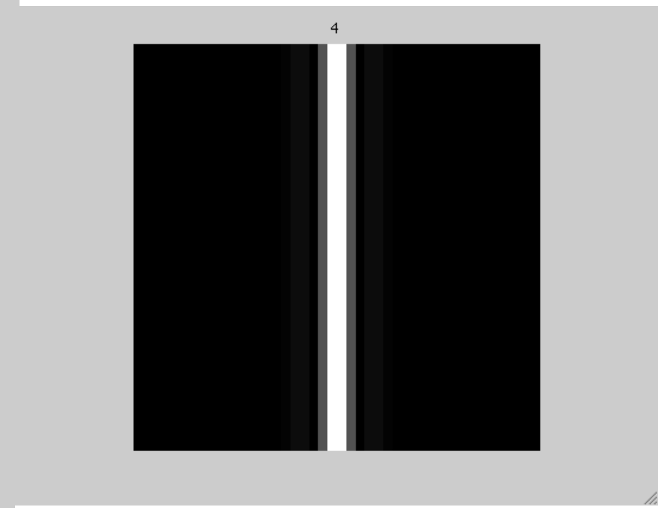


Contrast invariance! (same energy response for white dot on black background as for a black dot on a white background).

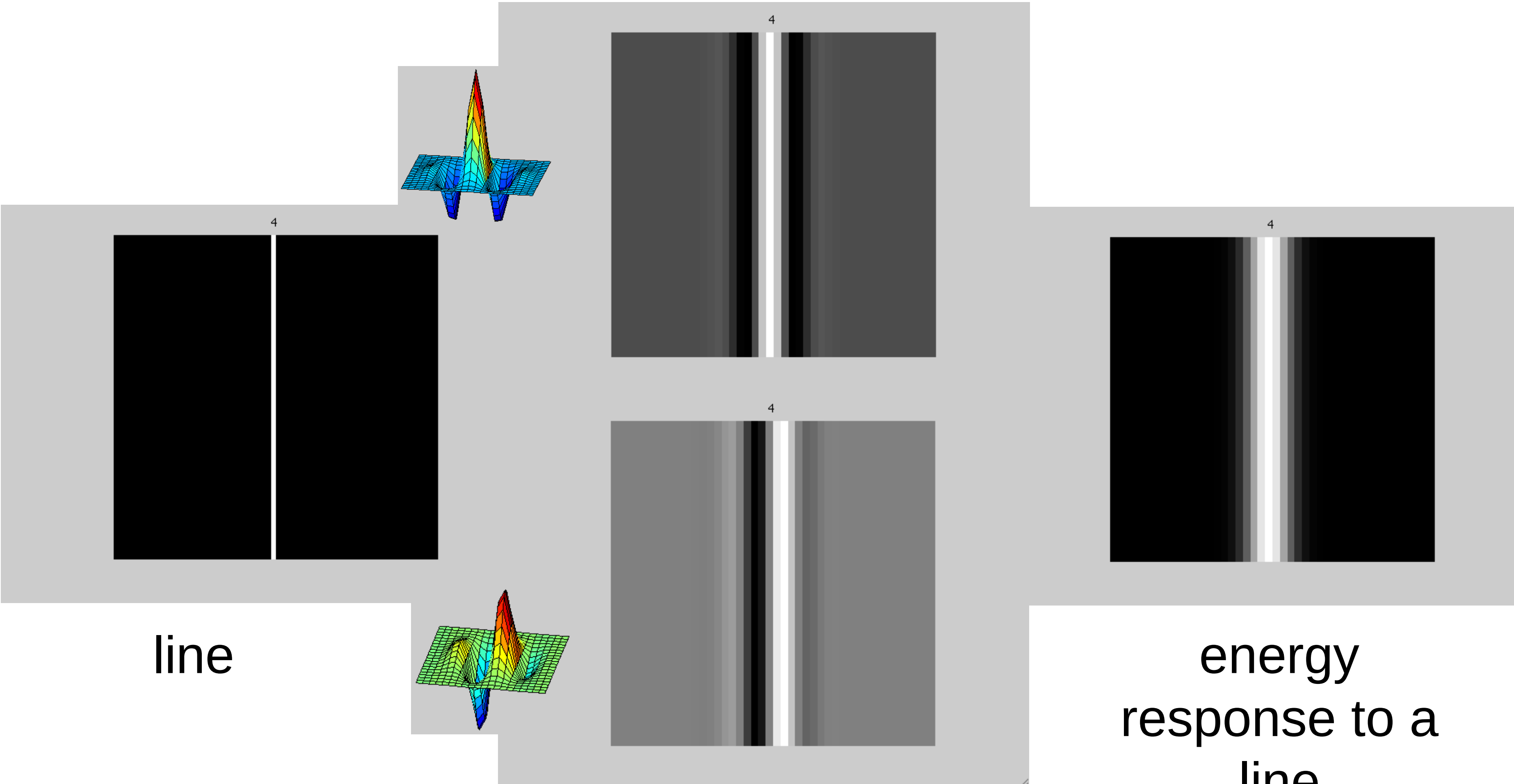


edge





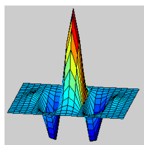
energy response to an edge

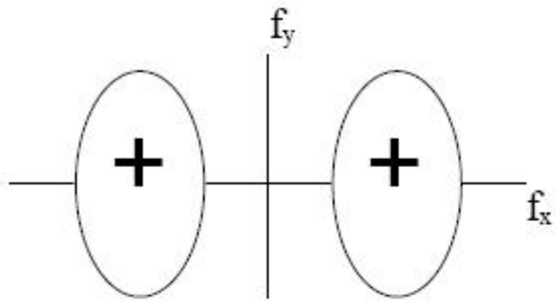


line

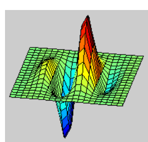
energy response to a line

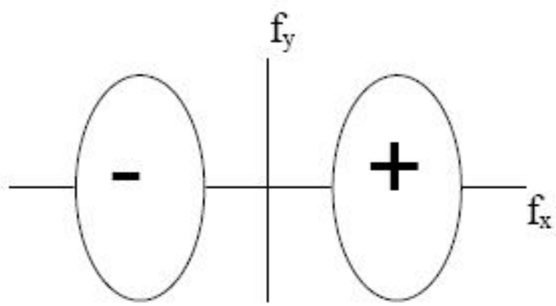
How quadrature pair filters work





(a) Frequency response of even filter, G
(real)

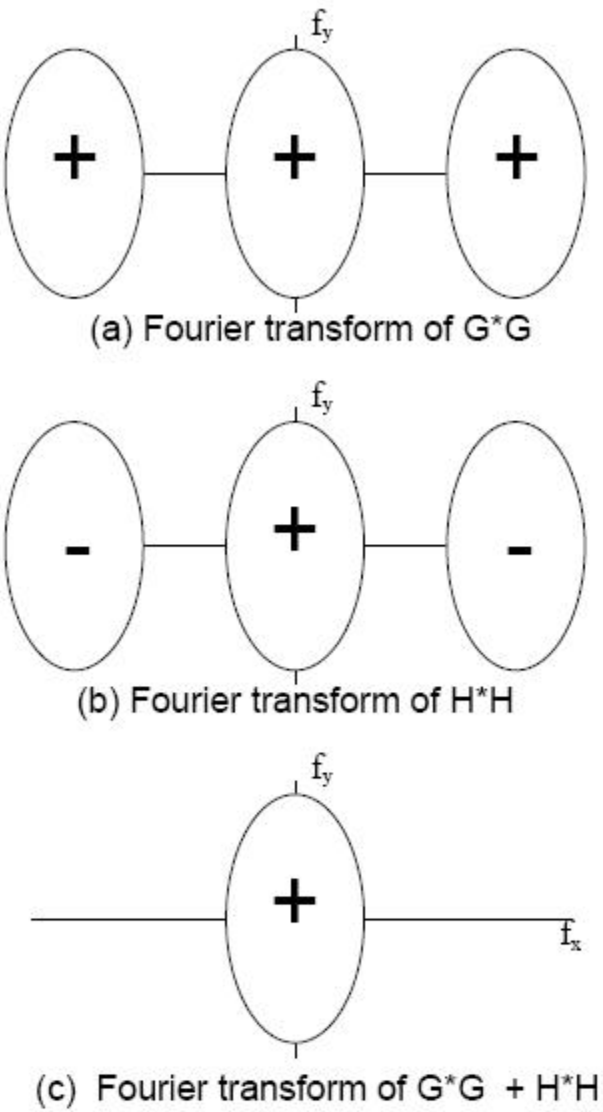




(b) Frequency response of odd filter, H
(imaginary)

Figure 3-5: Frequency content of two bandpass filters in quadrature. (a) even phase filter, called G in text, and (b) odd phase filter, H. Plus and minus sign illustrate relative sign of regions in the frequency domain. See Fig. 3-6 for calculation of the frequency content of the energy measure derived from these two filters.

How quadrature pair filters work



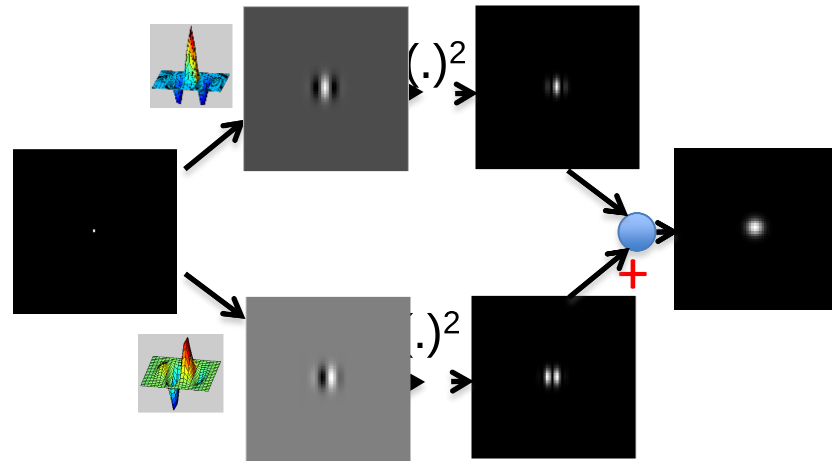
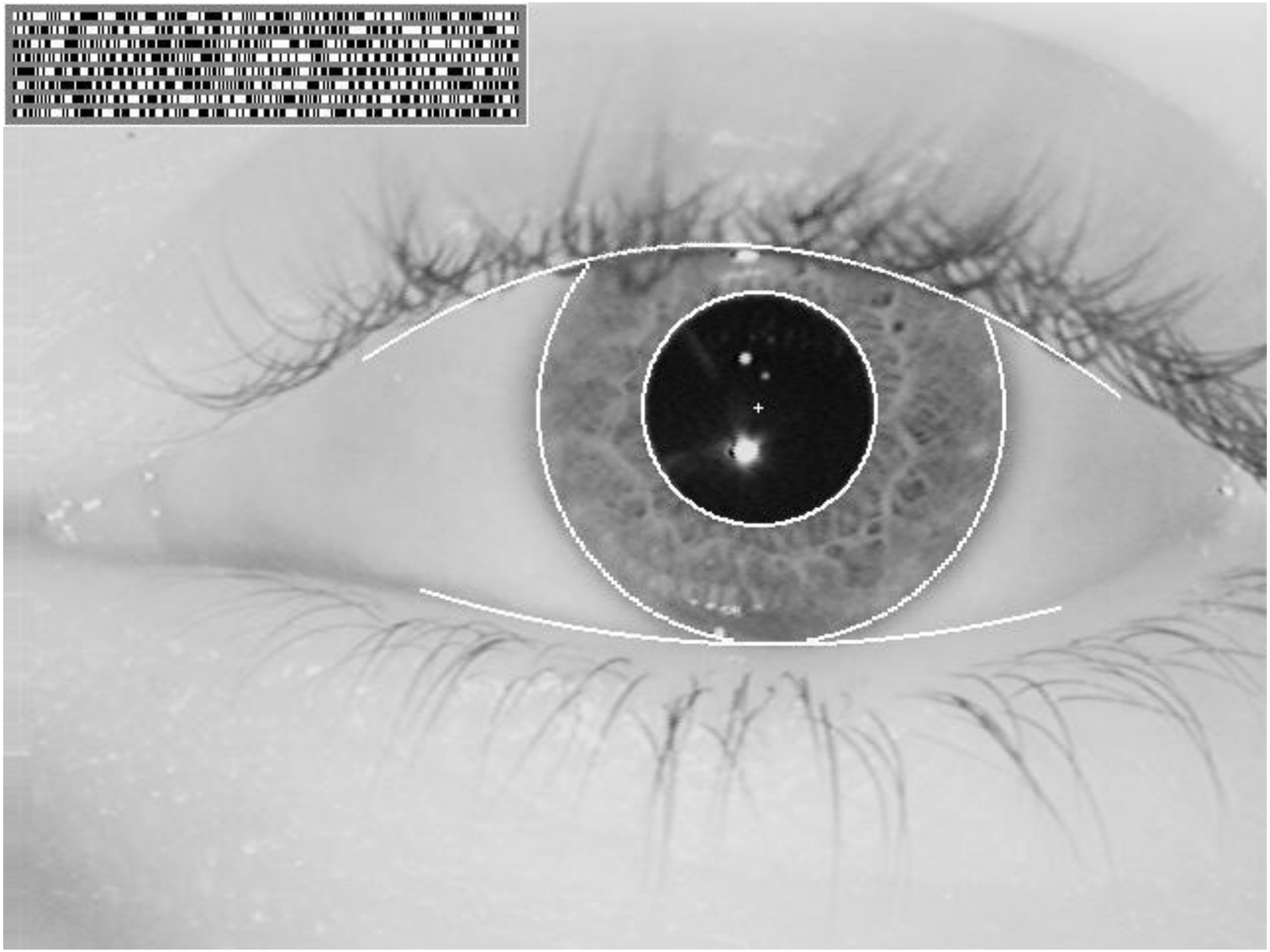


Figure 3-6: Derivation of energy measure frequency content for the filters of Fig. 3-5. (a) Fourier transform of $G * G$. (b) Fourier transform of $H * H$. Each squared response has 3 lobes in the frequency domain, arising from convolution of the frequency domain responses. The center lobe is modulated down in frequency while the two outer lobes are modulated up. (There are two sign changes which combine to give the signs shown in (b). To convolve H with itself, we flip it in f_x and f_y, which interchanges the + and − lobes of Fig. 3-5 (b). Then we slide it over an unflipped version of itself, and integrate the product of the two. That operation will give positive outer lobes, and a negative inner lobe. However, H has an imaginary frequency response, so multiplying it by itself gives an extra factor of -1, which yields the signs shown in (b)). (c) Fourier transform of the energy measure, $G * G + H * H$. The high frequency lobes cancel, leaving only the baseband spectrum, which has been demodulated in frequency from the original bandpass response. This spectrum is proportional to the sum of the auto-correlation functions of either lobe of Fig. 3-5 (a) and either lobe of Fig. 3-5 (b).

Gabor filter measurements for iris recognition code



John Daugman, http://www.cl.cam.ac.uk/~jgd1000/

Setting the Bits in an IrisCode

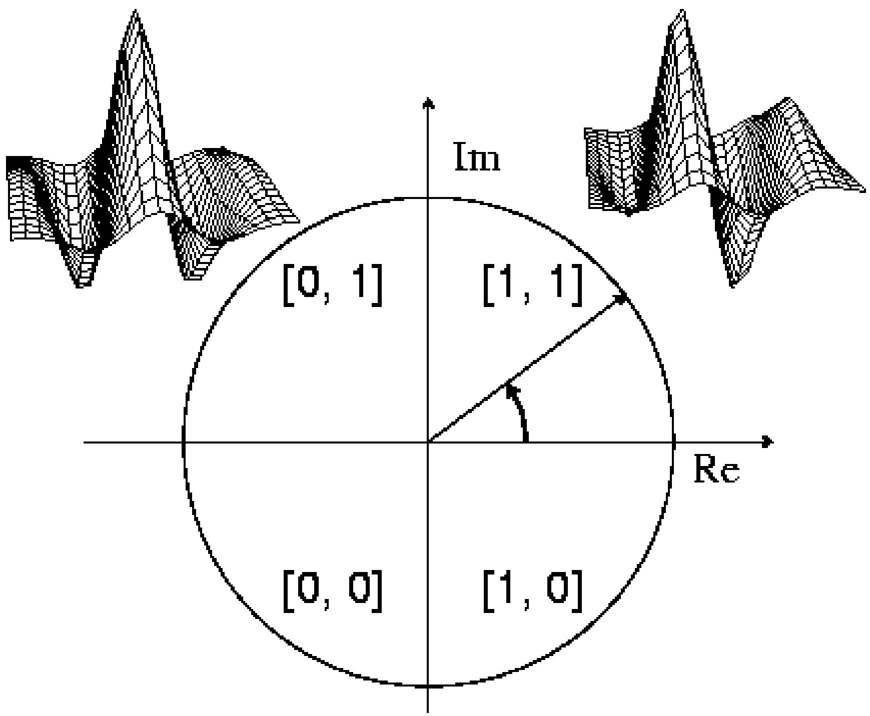
$$h_{Re} = 1 \text{ if } \mathrm{Re} \int_\rho \int_\phi e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi \geq 0$$

$$h_{Re} = 0 \text{ if } \mathrm{Re} \int_\rho \int_\phi e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi < 0$$

$$h_{Im} = 1 \text{ if } \mathrm{Im} \int_\rho \int_\phi e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi \geq 0$$

$$h_{Im} = 0 \text{ if } \mathrm{Im} \int_\rho \int_\phi e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi < 0$$

Phase-Quadrant Iris Demodulation Code



Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

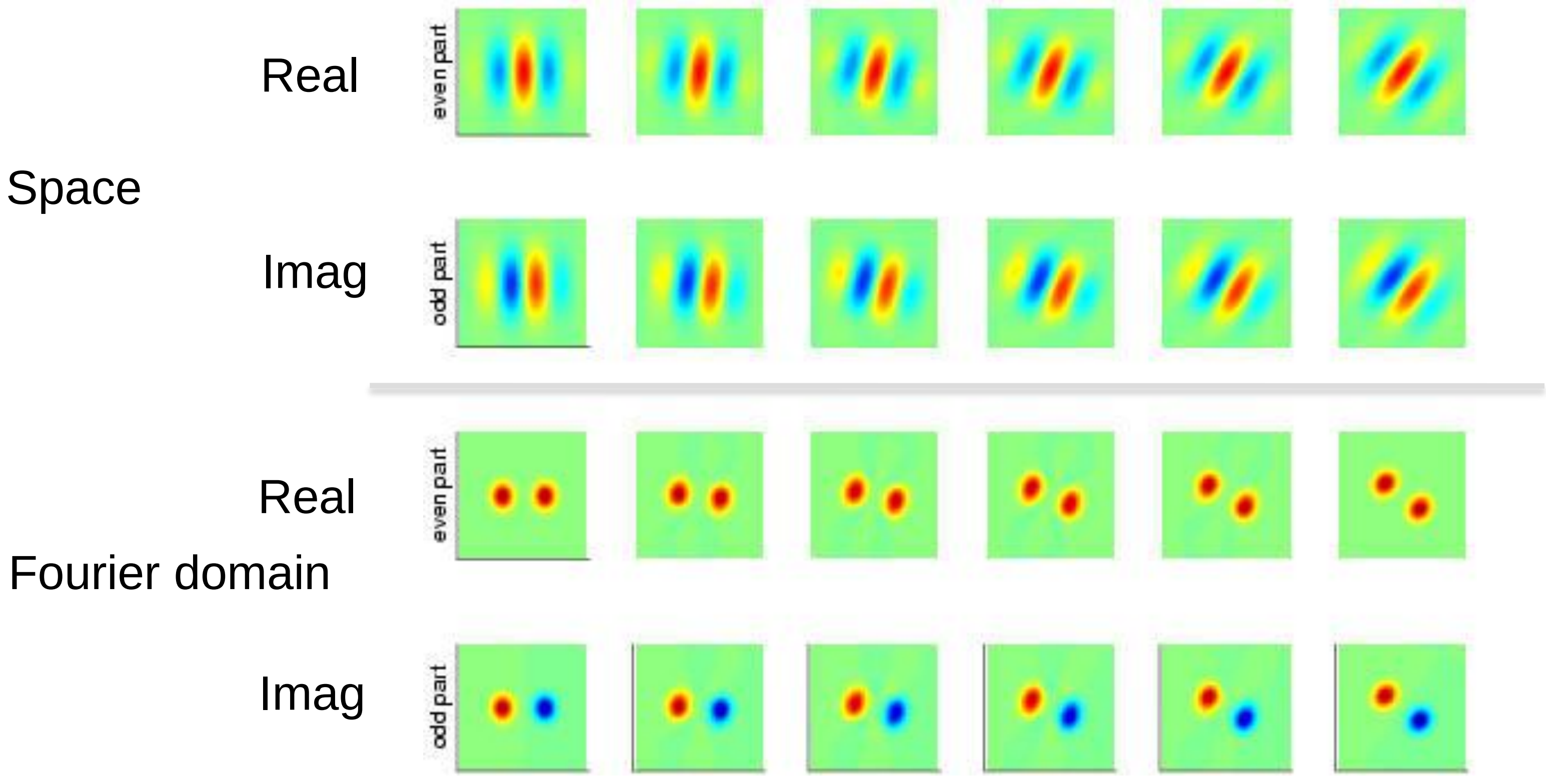
Gabor wavelet:

$$\psi(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j 2\pi u_0 x}$$

Tuning filter orientation:

$$x' = \cos(\alpha)x + \sin(\alpha)y$$

$$y' = -\sin(\alpha)x + \cos(\alpha)y$$



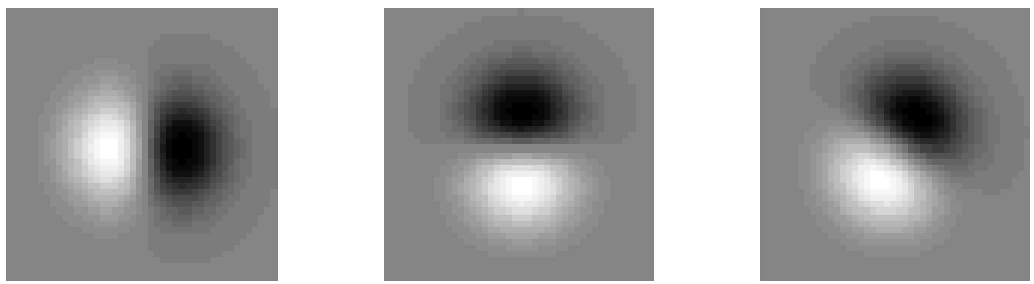
Simple example

“Steerability”-- the ability to synthesize a filter of any orientation from a linear combination of filters at fixed orientations.

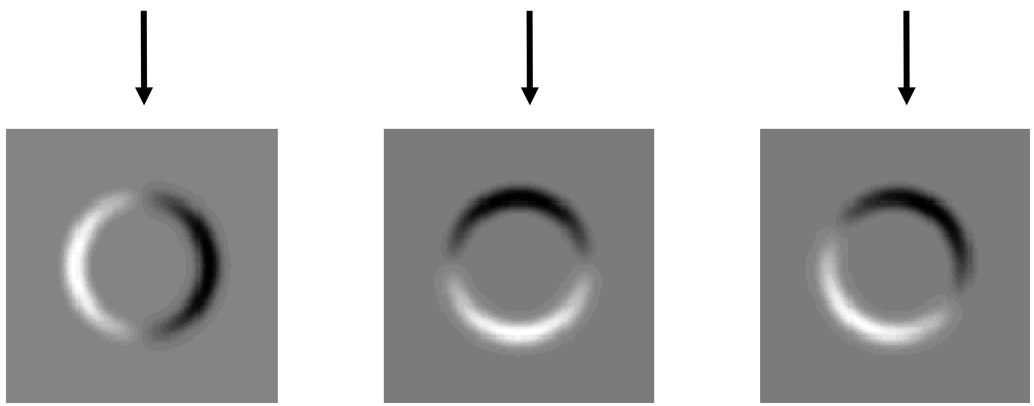
$$G_{\theta}^{1} = \cos(\theta) G_{0}^{1} + \sin(\theta) G_{90}^{1}$$

	0°	90°	Synthesized 30°

Filter Set:



Response:



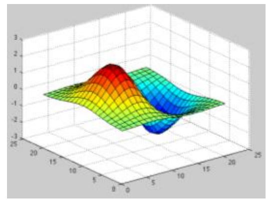
Raw Image

Taken from:
W. Freeman, T. Adelson, “The Design and Use of Sterrable Filters”, IEEE Trans. Patt, Anal. and Machine Intell., vol 13, #9, pp 891-900, Sept 1991

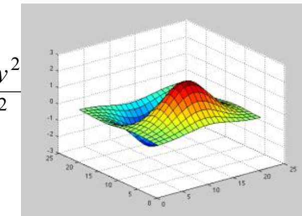
Steerable filters

Derivatives of a Gaussian:

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



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



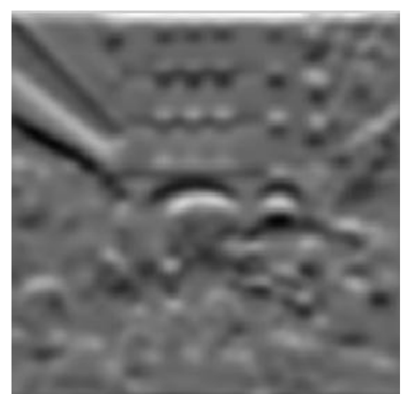
An arbitrary orientation can be computed as a linear combination of those two basis functions:

$$h_\alpha(x,y) = \cos(\alpha) h_x(x,y) + \sin(\alpha) h_y(x,y)$$

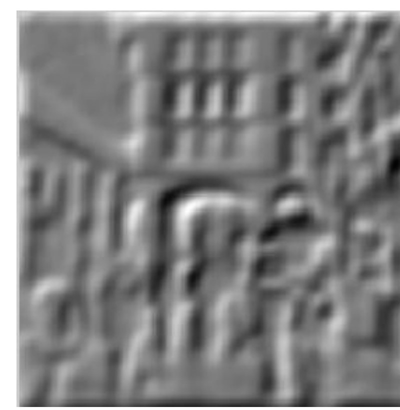
The representation is “shiftable” on orientation: We can interpolate any other orientation from a finite set of basis functions.

cos(〈)

+sin(〈)



=



Freeman & Adelson 92

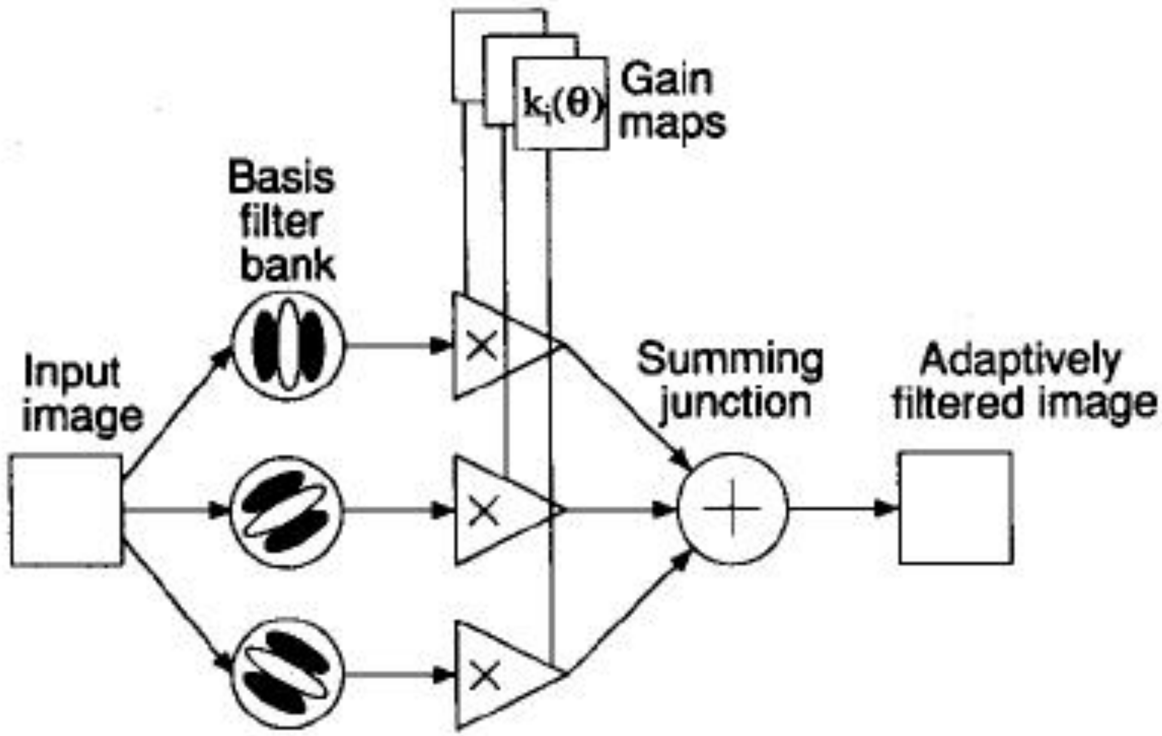


Fig. 3. Steerable filter system block diagram. A bank of dedicated filters process the image. Their outputs are multiplied by a set of gain maps that adaptively control the orientation of the synthesized filter.

Steering theorem

Change from Cartesian to polar coordinates

f(x,y) ⟷ H(r,⎝)

A convolution kernel can be written using Fourier series in polar angle as:

$$f(r,\phi) = \sum_{n=-N}^{N} a_n(r) e^{in\phi}$$

Theorem: Let T be the number of nonzero coefficients $a_n(r)$. Then, the function f can be steered with T functions.

Steering theorem for polynomials

$$f(x,y) = W(r)\, P(x,y)$$

Theorem 3: Let $f(x,y) = W(r)P_N(x,y)$, where $W(r)$ is an arbitrary windowing function, and $P_N(x,y)$ is an Nth order polynomial in x and y, whose coefficients may depend on r. Linear combinations of $2N+1$ basis functions are sufficient to synthesize $f(x,y) = W(r)P_N(x,y)$ rotated to any angle. Equation (10) gives the interpolation functions $k_j(\theta)$. If $P_N(x,y)$ contains only even [odd] order terms (terms $x^n y^m$ for $n+m$ even [odd]), then $N+1$ basis functions are sufficient, and (10) can be modified to contain only the even [odd] numbered rows (counting from zero) of the left-hand side column vector and the right-hand side matrix.

For an Nth order polynomial with even or odd symmetry N+1 basis functions are sufficient.

Steerability and Separability

Important example is 2nd derivative of Gaussian $G_2^{0^\circ} = (4x^2 - 2)e^{-(x^2+y^2)}$ (~Laplacian):

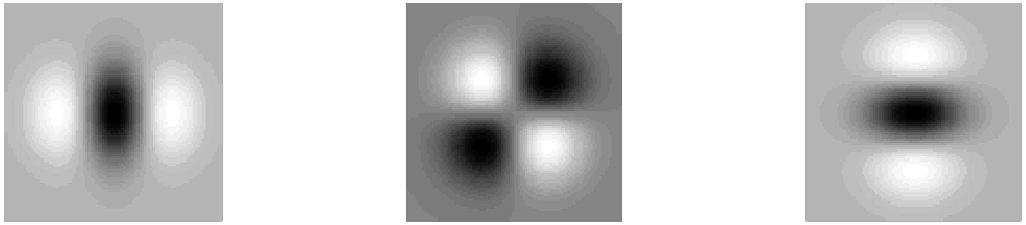


Figure 16: X-Y separable basis filters for G_2, listed in Tables 3 and 4.

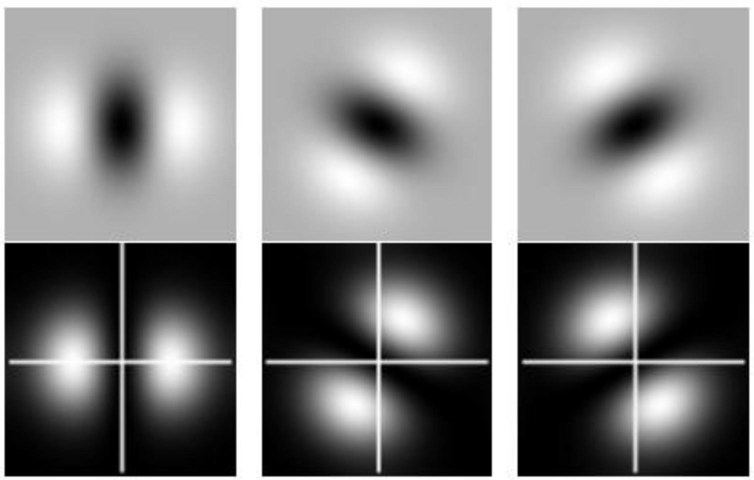
G_{2a}	=	$0.9213(2x^2 - 1)e^{-(x^2+y^2)}$	$k_a(\theta)$	=	$\cos^2(\theta)$
G_{2b}	=	$1.843xye^{-(x^2+y^2)}$	$k_b(\theta)$	=	$-2\cos(\theta)\sin(\theta)$
G_{2c}	=	$0.9213(2y^2 - 1)e^{-(x^2+y^2)}$	$k_c(\theta)$	=	$\sin^2(\theta)$

Table 3: X-Y separable basis set and interpolation functions for second derivative of Gaussian. To create a second derivative of a Gaussian rotated along to an angle θ, use: $G_2^\theta = (k_a(\theta)\ G_{2a} + k_b(\theta)\ G_{2b} + k_c(\theta)\ G_{2c})$. The minus sign in $k_b(\theta)$ selects the direction of positive θ to be counter-clockwise.

Taken from: W. Freeman, T. Adelson, "The Design and Use of Steerable Filters", IEEE Trans. Patt, Anal. and Machine Intell., vol 13, #9, pp 891-900, Sept 1991

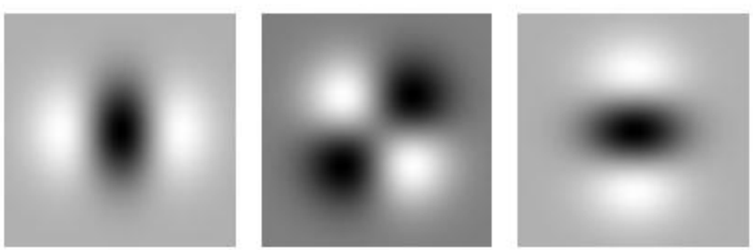
Two equivalent basis

These two basis can use to steer 2nd order Gaussian derivatives



(a) G_2 Basis Set

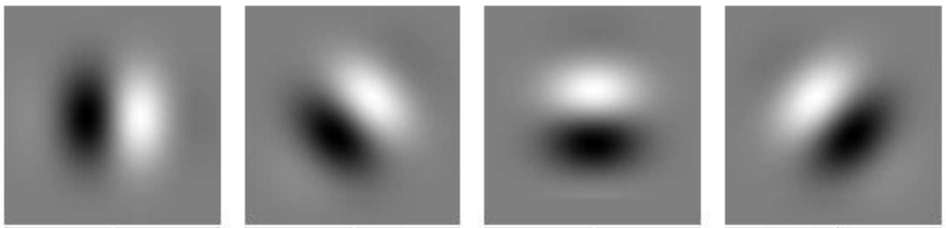
(b) G_2 Amplitude Spectra



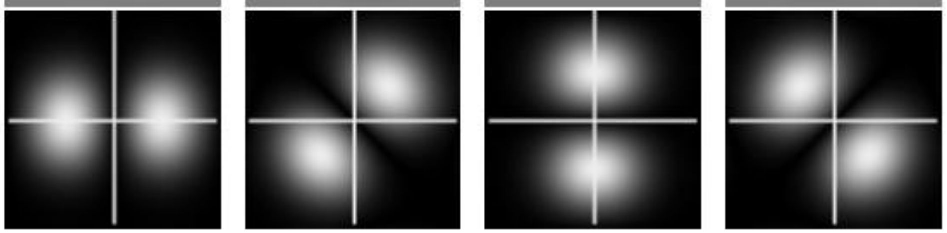
(c) G_2 X-Y Separable Basis Set

Approximated quadrature filters for 2nd order Gaussian derivatives

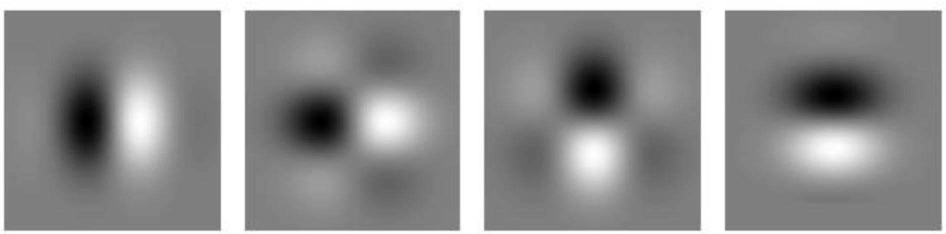
(this approximation requires 4 basis to be steerable)



(d) H_2 Basis Set



(e) H_2 Amplitude Spectra

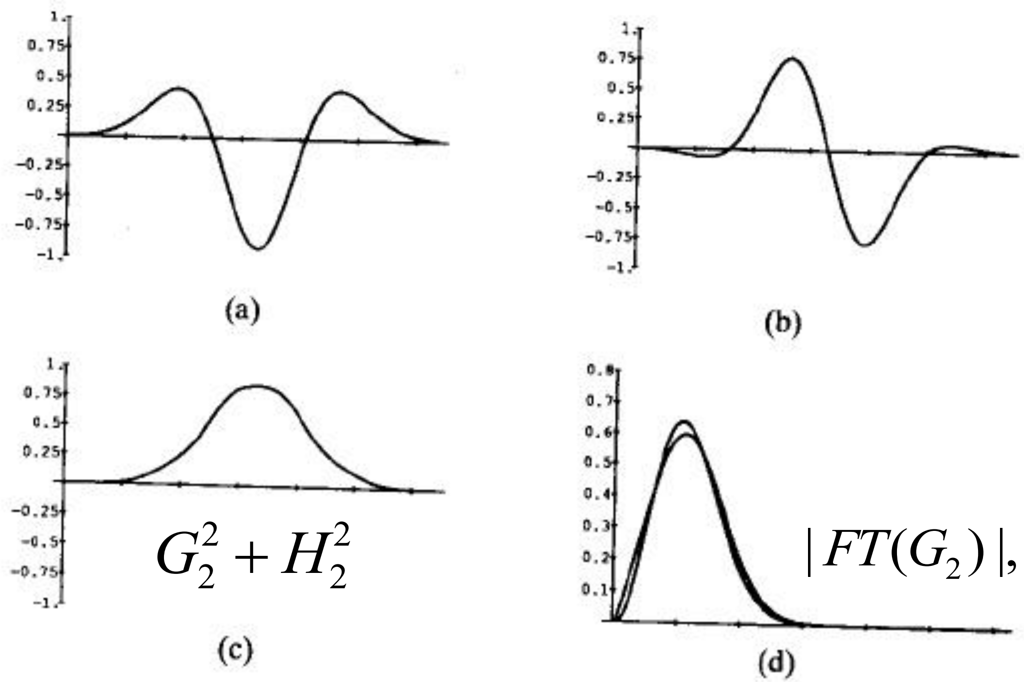


(f) H_2 X-Y Separable Basis Set

Steerable quadrature pairs

For the Gaussian derivatives we can approximate a quadrature pair

G_2 H_2



$G_2^2 + H_2^2$

$|FT(G_2)|, \quad |FT(H_2)|$

Fig. 4. (a) G_2, second derivative of Gaussian (in one dimension); (b) H_2, fit of third order polynomial (times Gaussian) to the Hilbert transform of (a); (c) energy measure: $(G_2)^2 + (H_2)^2$; (d) magnitudes of Fourier transforms of (a) and (b).

Orientation analysis

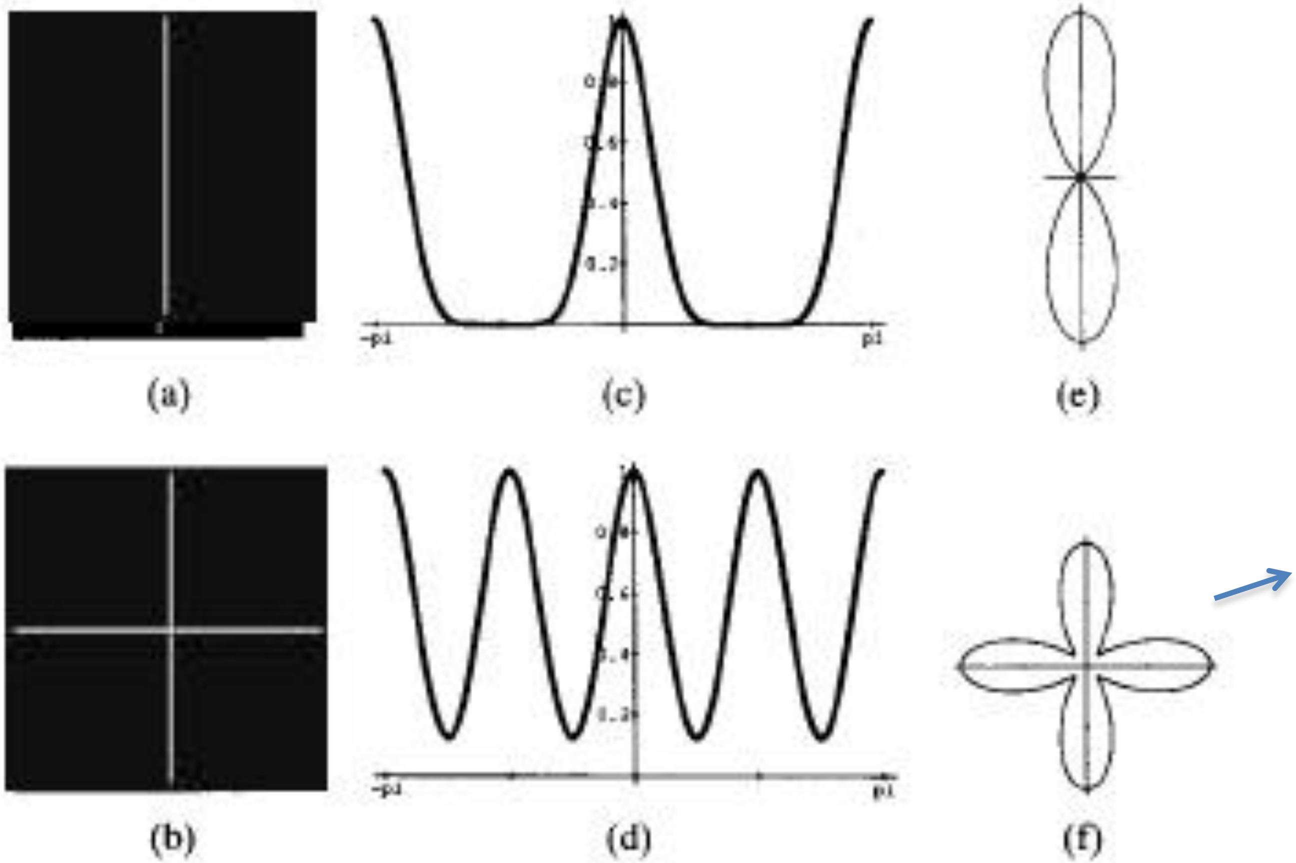
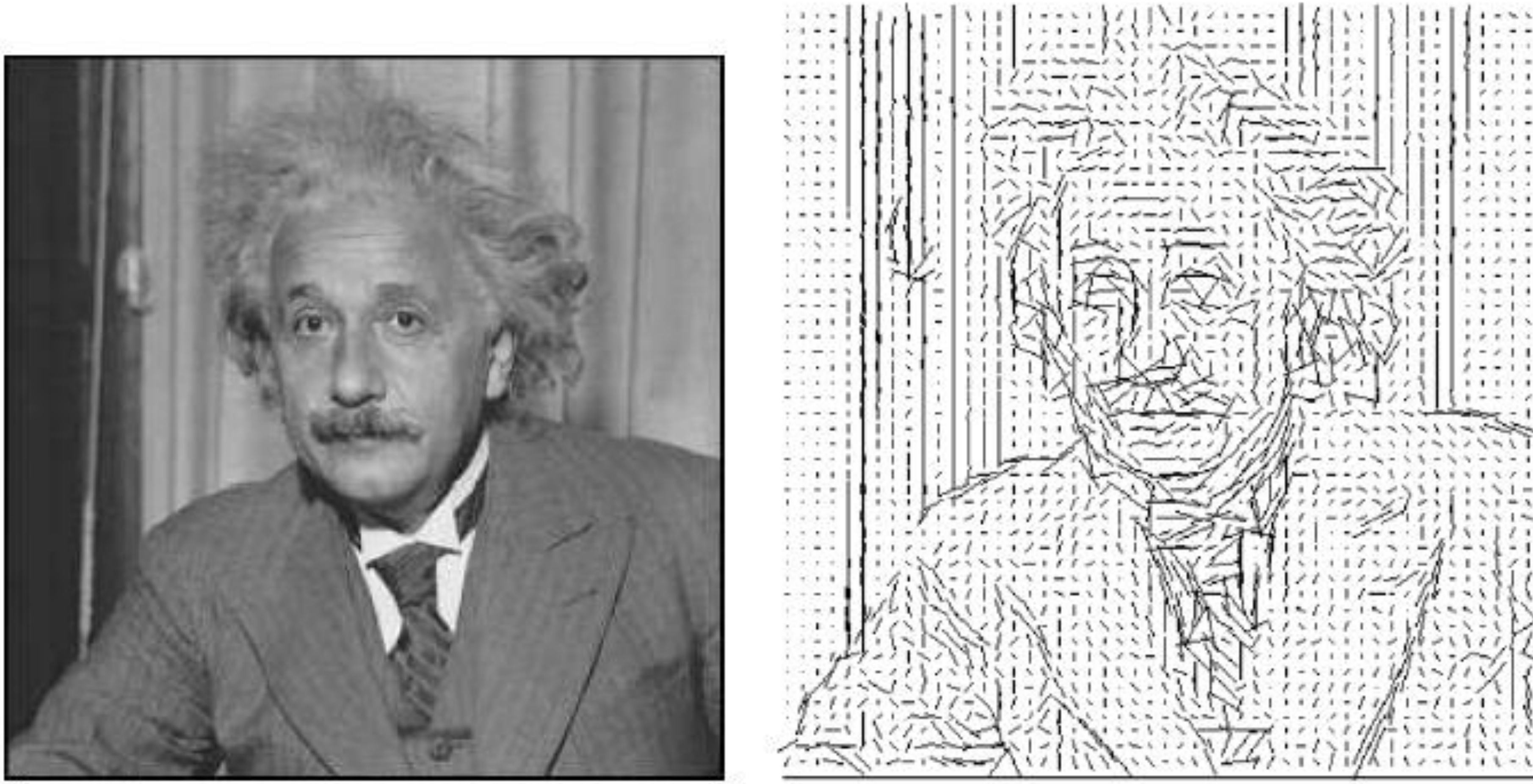


Fig. 9. Test images of (a) vertical line and (b) intersecting lines; (c) and (d) oriented energy as a function of angle at the centers of test images (a) and (b). Oriented energy was measured using the G_4, H_4 quadrature steerable pair; (e) and (f) polar plots of (c) and (d).

High resolution in orientation requires many oriented filters as basis (high order gaussian derivatives).

Orientation analysis



(a) (b)

Fig. 8. (a) Original image of Einstein; (b) orientation map of (a) made using the lowest order terms in a Fourier series expansion for the oriented energy as measured with G_2 and H_2. Table XI gives the formulas for these terms.

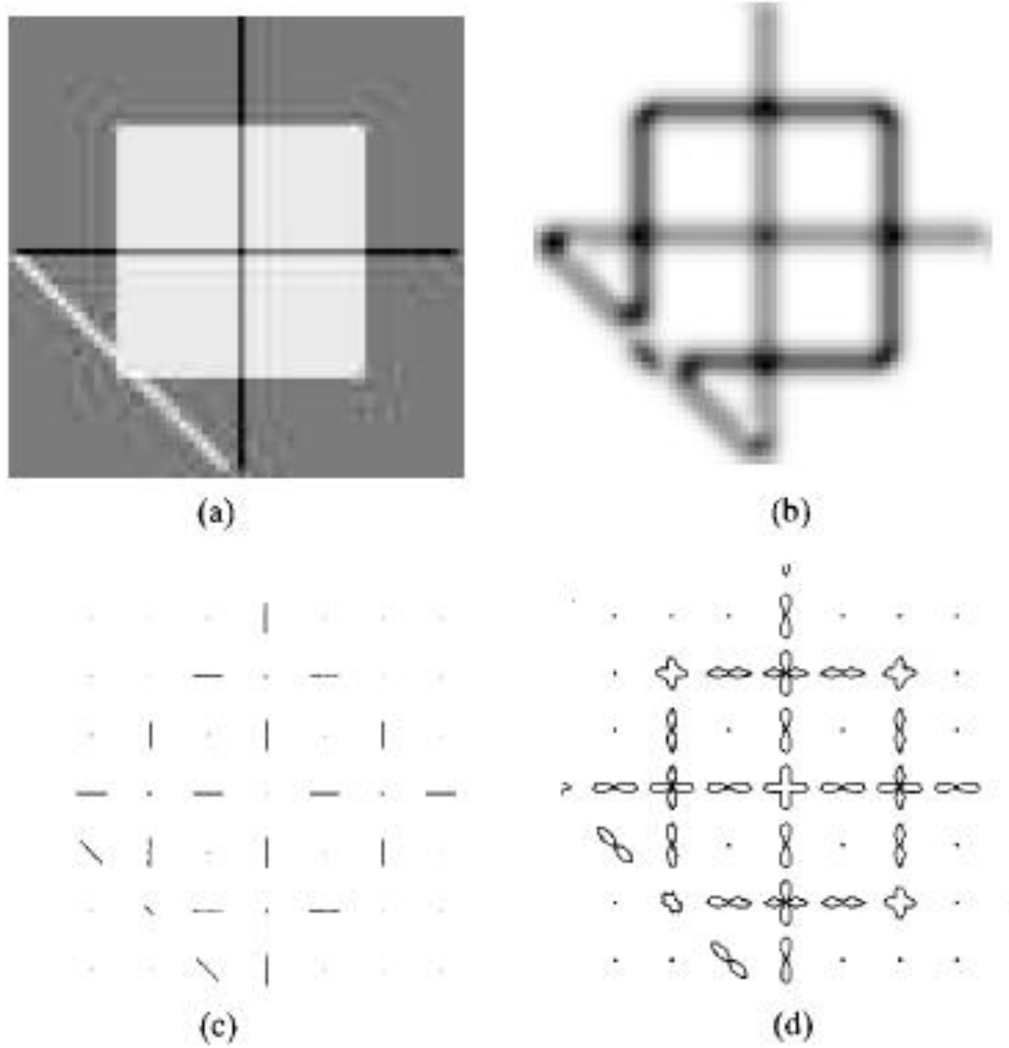
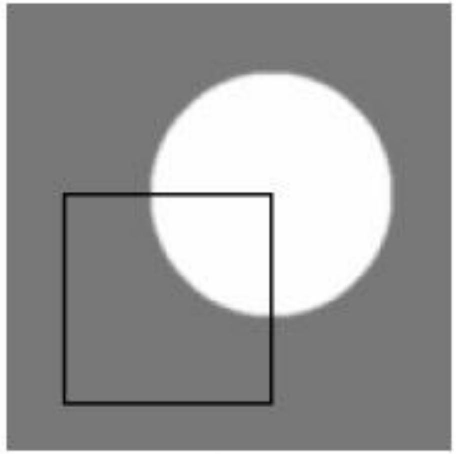


Fig. 10. Measures of orientation derived from G_4 and H_4 steerable filter outputs: (a) Input image for orientation analysis; (b) angular average of oriented energy as measured by G_4, H_4 quadrature pair. This is an oriented features detector; (c) conventional measure of orientation: dominant orientation plotted at each point. No dominant orientation is found at the line intersection or corners; (d) oriented energy as a function of angle, shown as a polar plot for a sampling of points in the image (a). Note the multiple orientations found at intersection points of lines or edges and at corners, shown by the florets there.



(a)

A contour detector

(b)

(c)

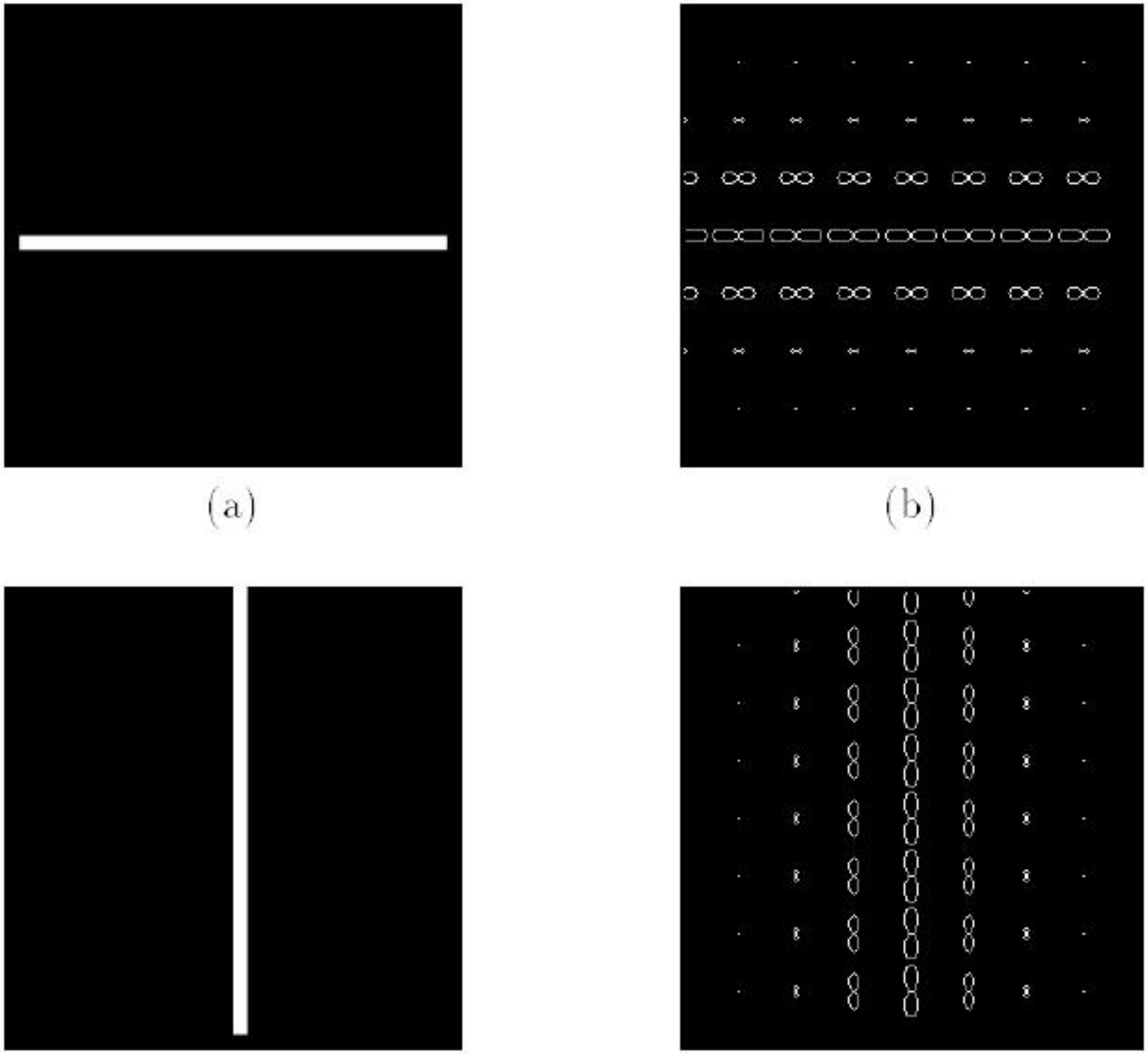
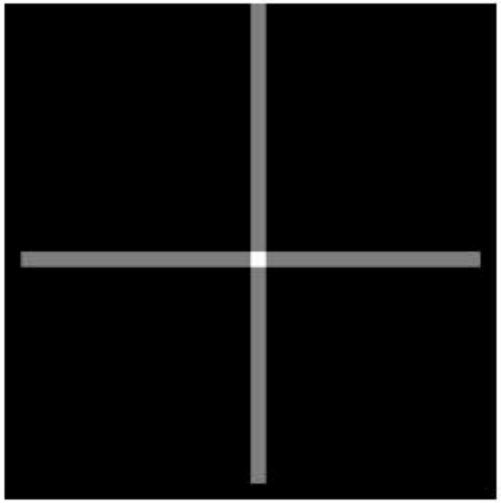
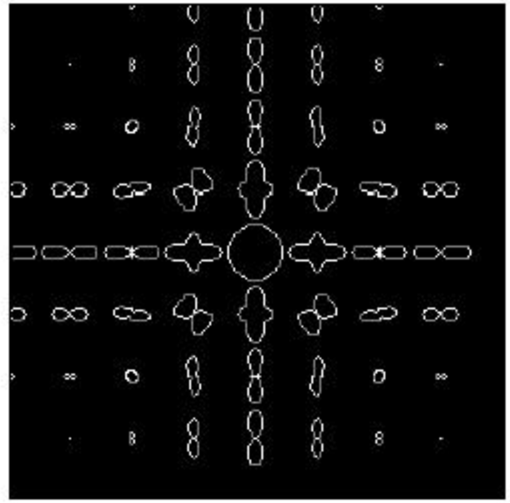


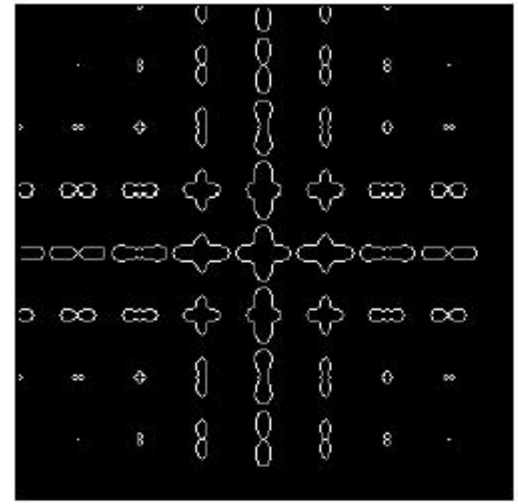
Figure 3-8: The problem with using energy measures to analyze a structure of multiple orientations, and how to solve it (part one). (a) Horizontal line and (b) floret polar plot of G_2 and H_2 quadrature pair oriented energies as a function of angle and position. The same for a vertical line are shown in (c) and (d). Continued in Fig. 3-9



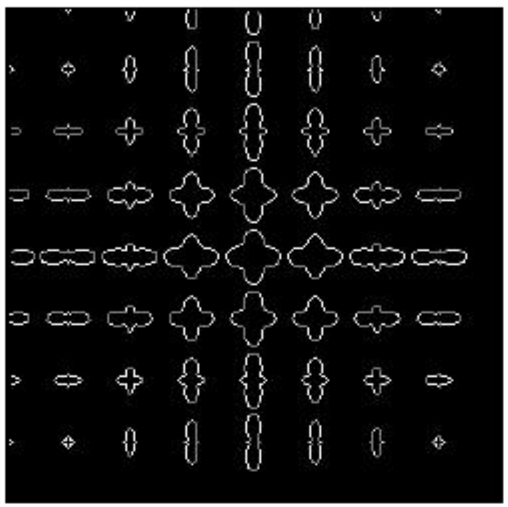
(a)



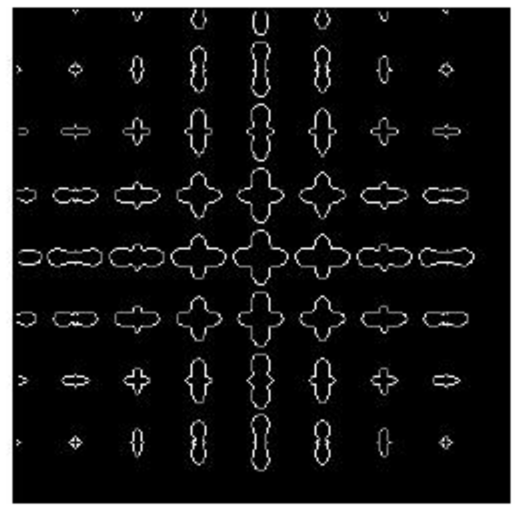
(b)



(c)



(d)



(e)

Figure 3-9: The problem with using energy measures to analyze a structure of multiple orientations, and how to solve it (part two). (a) Cross image (the sum of Fig. 3-8 (a) and (c)). The oriented energy (b) of the cross is not the sum of the energies of the horizontal and vertical lines, Fig. 3-8 (b) and (d), due to an effect analogous to optical interference. Many of the florets do not show the two orientations which are present; several show angularly uniform responses. For comparison, (c) shows the sum of energies Fig. 3-8 (b) and (d). Floret polar plot of energies after spatial blurring, (d), are predicted to remove interference effects, as described in text. Note that the energy local maxima correspond to image structure orientations. These florets are nearly identical to the sum of blurred energies of the horizontal and vertical lines, (e), showing that superposition nearly holds. (The agreement is not exact because the low-pass filter used for the blurring was not perfect).

Interference: why you should blur the oriented energy image

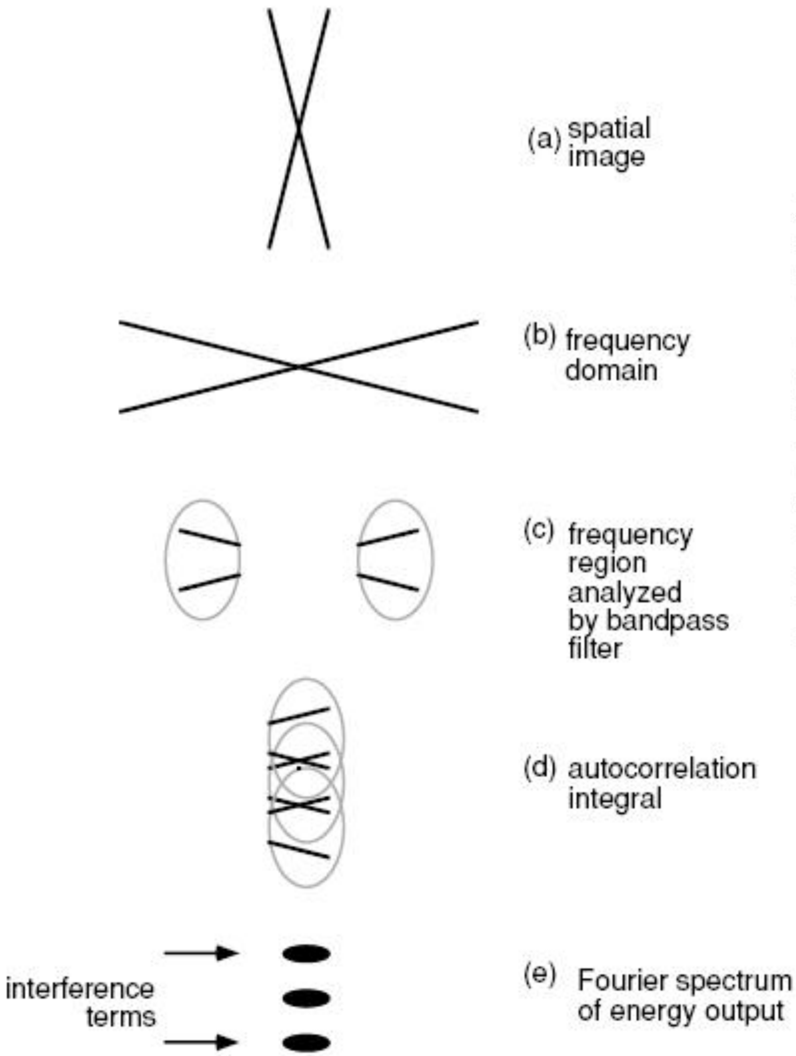


Figure 3-7: Showing the origin of interference effects when using energy measures to analyze regions of multiple orientations. (a) Test image of two intersecting lines. (b) Fourier transform of (a). (c) Part of (b) seen by the bandpass filters. (d) Frequency spectrum of energy measure applied to image (a). This is proportional to the auto-correlation of either one of the two lobes of (b). The result has 3 dominant contributions. The middle blob at DC is the integral of the squared frequency response over the bandpass region. For this term, superposition holds, and the energy of the sum of two images (non-overlapping in the frequency domain) will be the sum of the energies of each individual image. The other two terms are interference terms, arising from interactions between the Fourier transforms of the two images. Low-pass filtering the squared energy output can remove those terms while retaining the term for which superposition holds. Note this is not the same as low-pass filtering the linear filters before taking the energy.

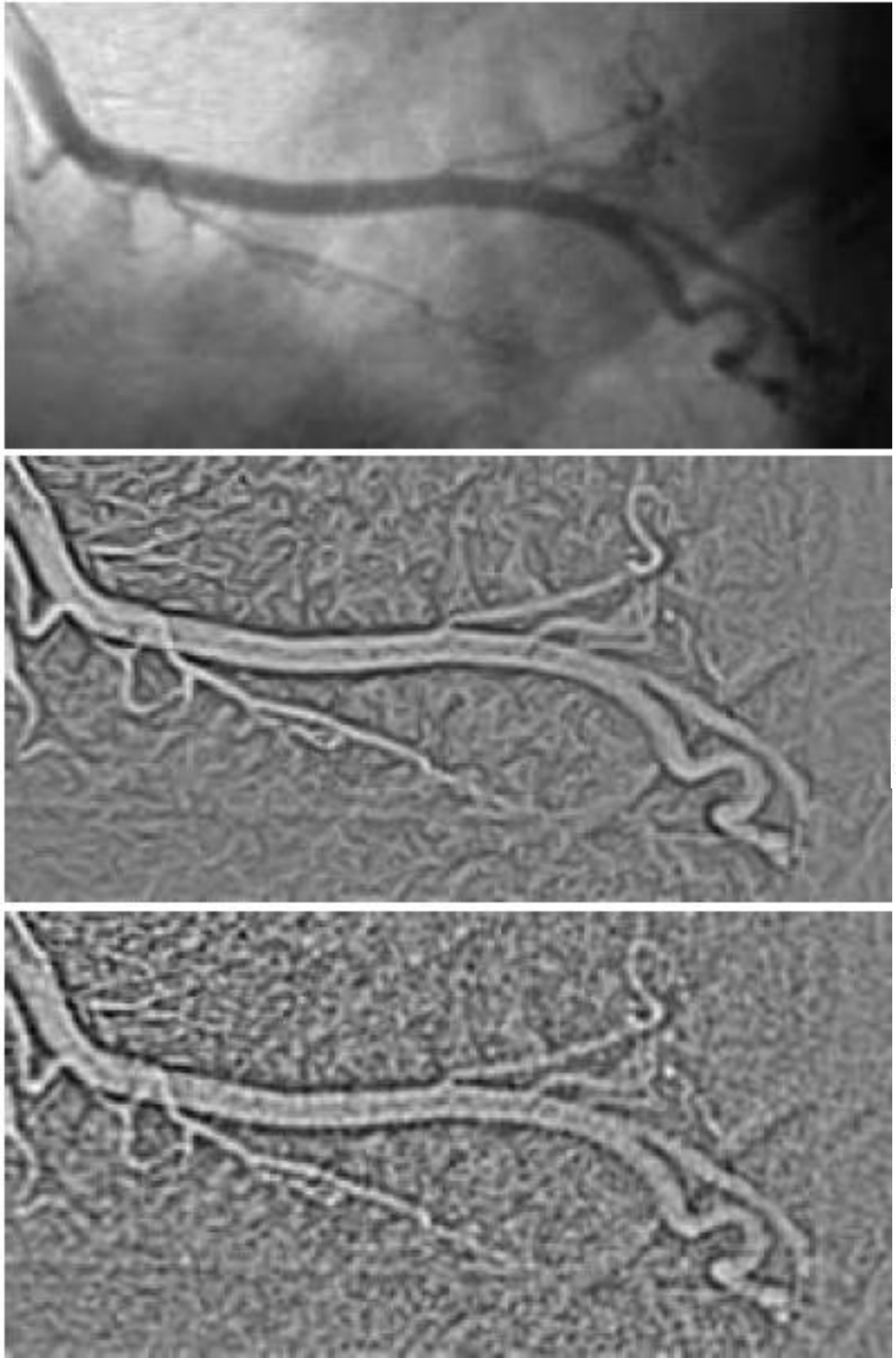


Fig. 12. (a) Digital cardiac angiogram; (b) result of filtering (a) with G_2 oriented along the local direction of dominant orientation, shown after local contrast enhancement (division by the image's blurred absolute value). The oriented vascular structures of (a) are enhanced; (c) isotropic bandpass filtering of (a) after local contrast enhancement. Note the increased noise relative to the oriented filtering results.

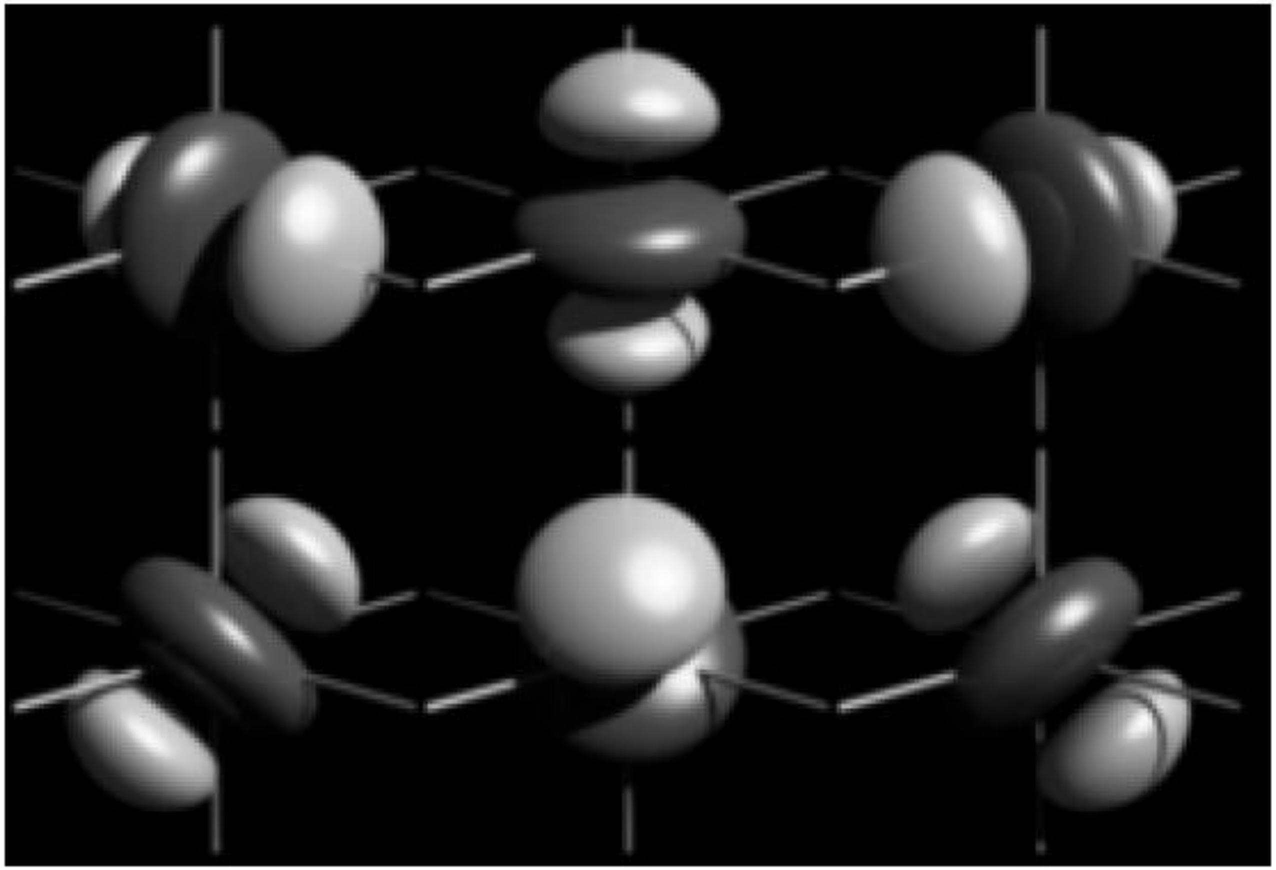


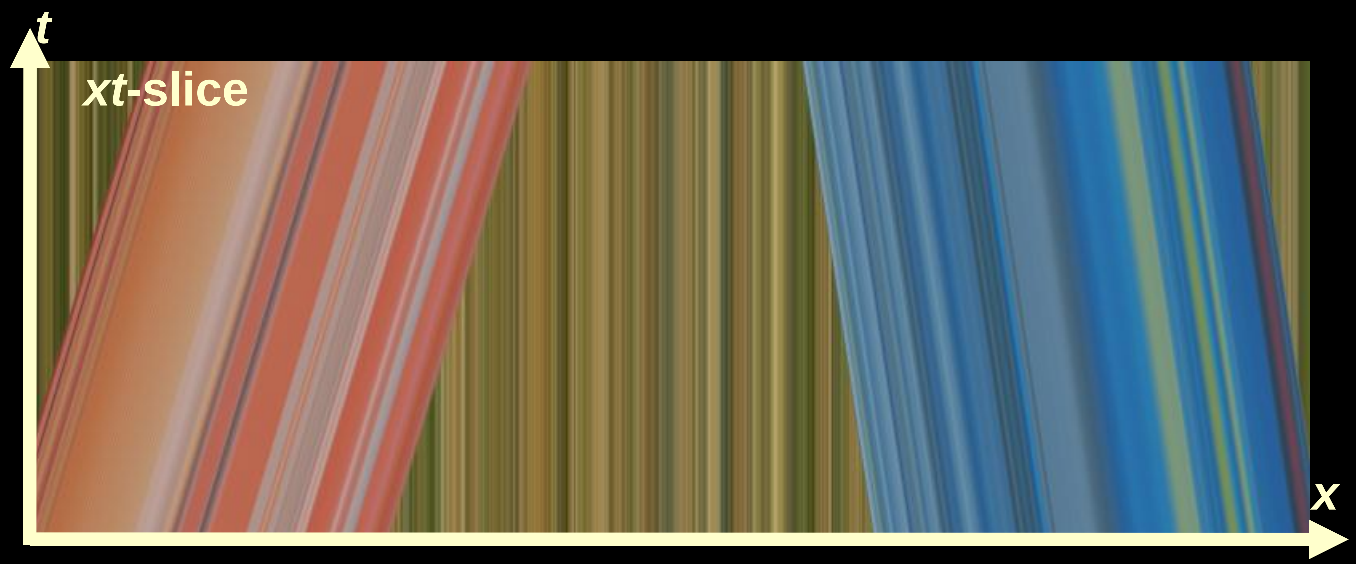
Figure 2-10: Example of a three-dimensional steerable filter. Surfaces of constant value are shown for the six basis filters of a second derivative of a three-dimensional Gaussian. Linear combinations of these six filters can synthesize the filter rotated to any orientation in three-space. Such three-dimensional steerable filters are useful for analysis and enhancement of motion sequences or volumetric image data, such as MRI or CT data. For discussions of steerable filters in three or more dimensions, see [59, 58, 33, 89]. (Martin Friedmann rendered this image with the Thingworld program).

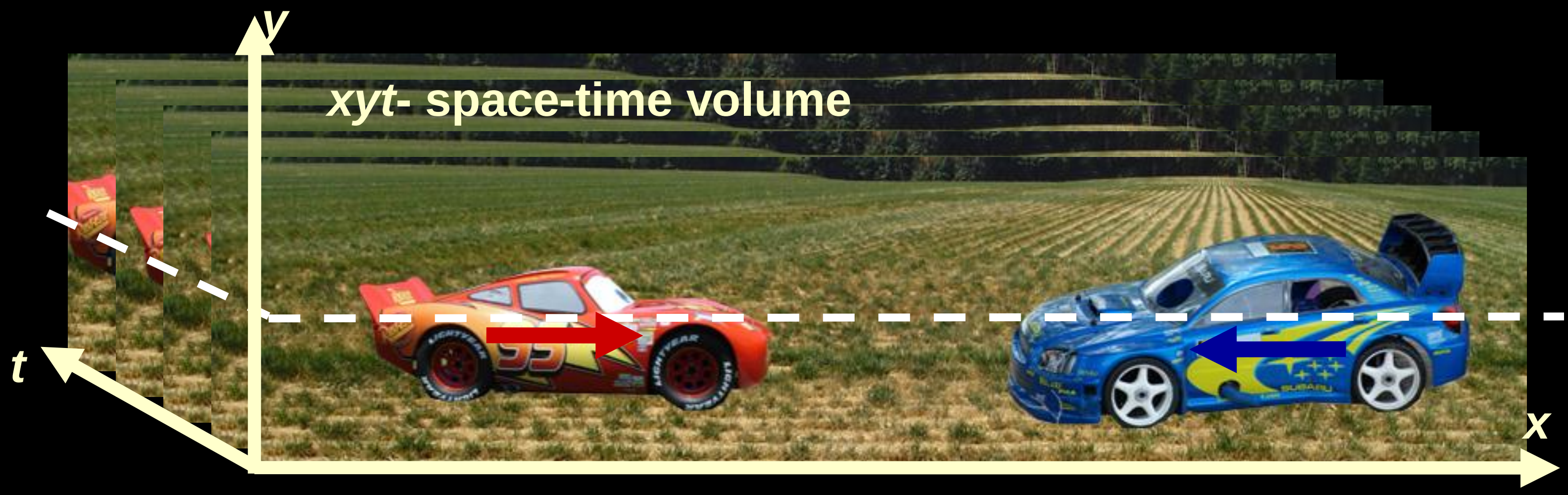
Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

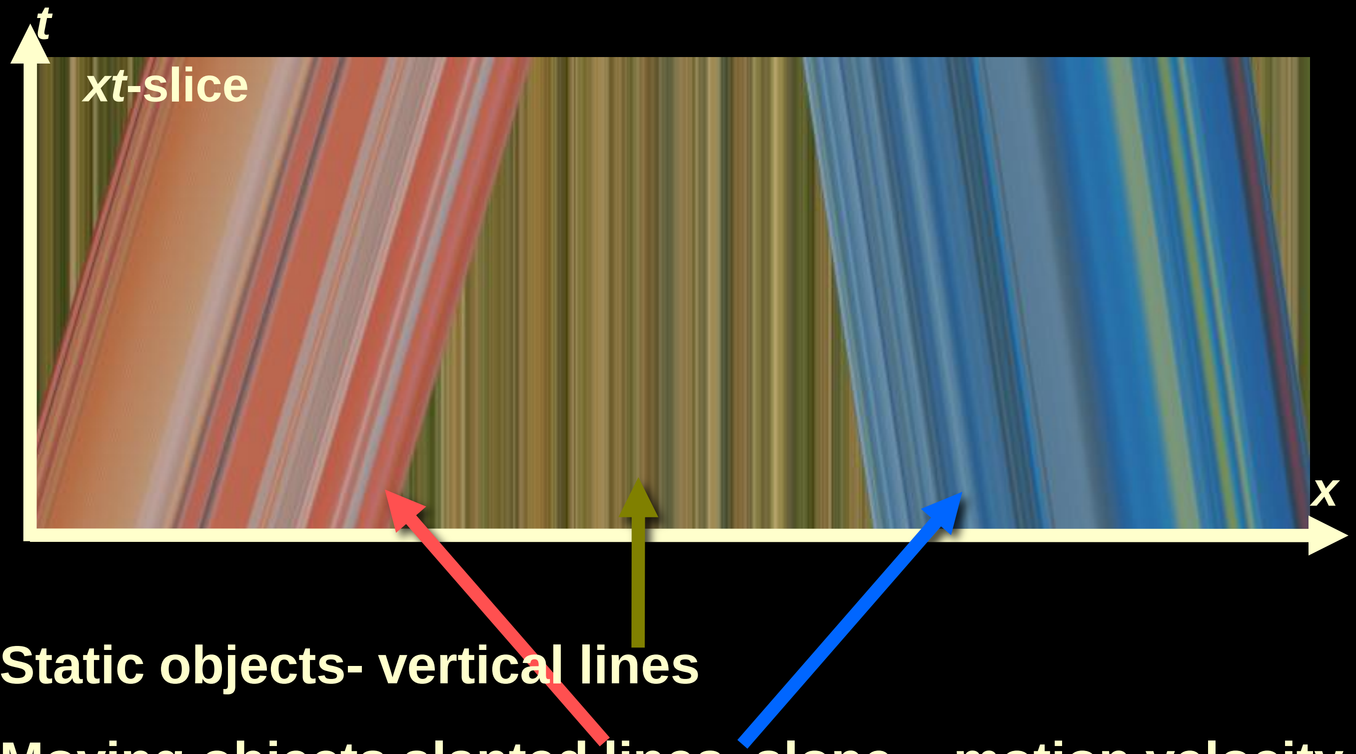
The space time volume

The space time volume





The space time volume

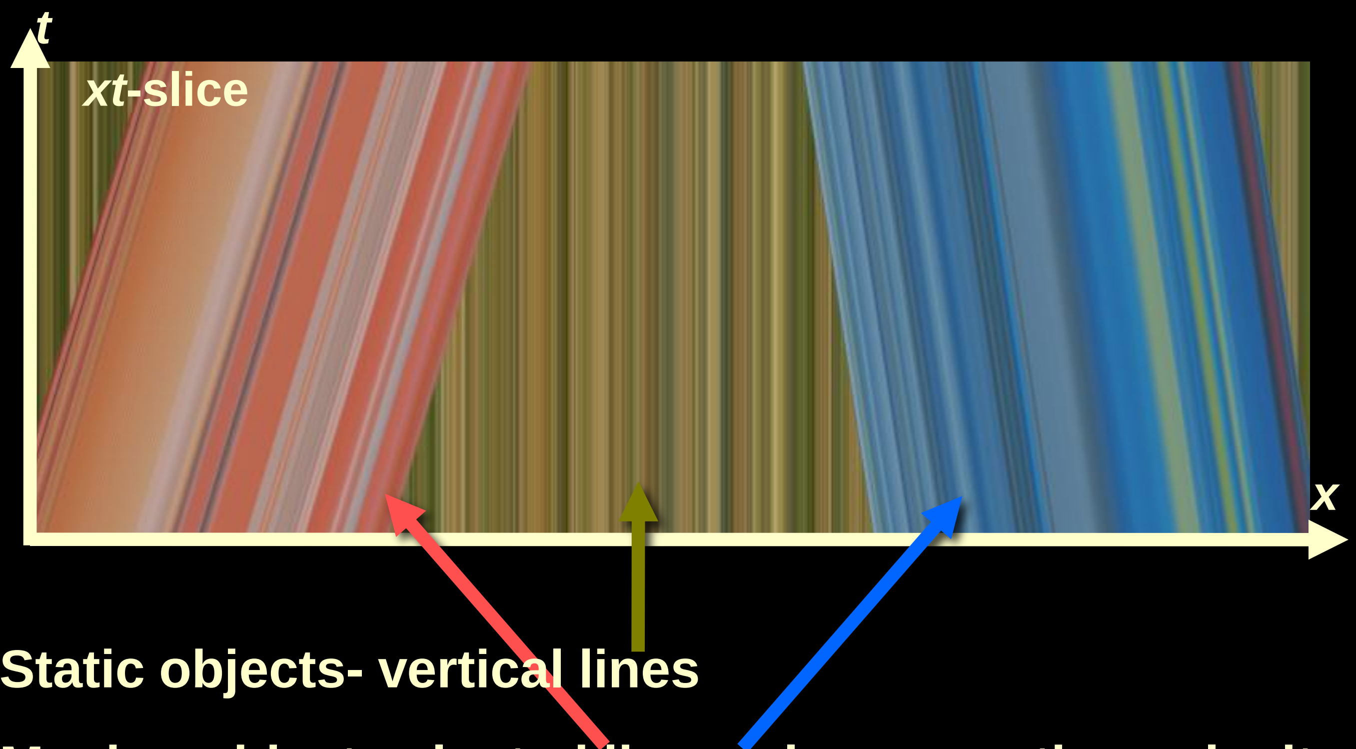


Static objects- vertical lines

Moving objects slanted lines, slope ~ motion velocity

x

The space time volume

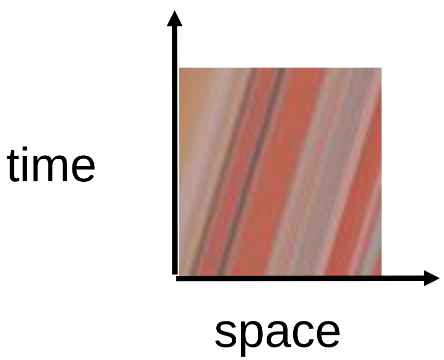


Static objects- vertical lines

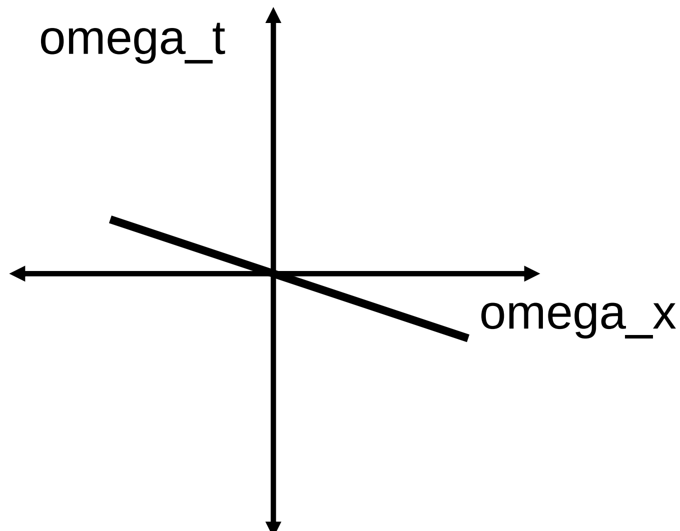
Moving objects slanted lines, slope ~ motion velocity

x

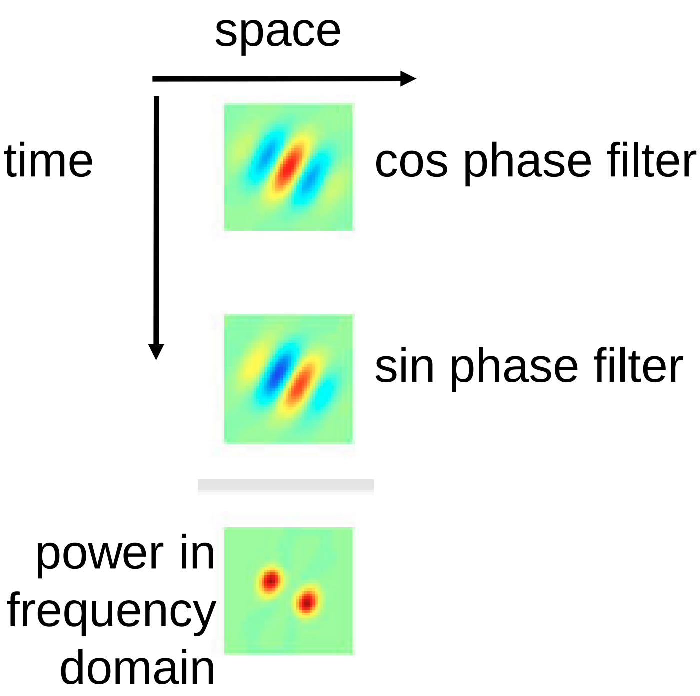
space-time domain



spatio-temporal Fourier transform domain



note locus of energy


space
time
cos phase filter
sin phase filter
power in
frequency
domain

Evidence for filter-based analysis of motion in the human visual system

Square wave Fourier components

Using Fourier series we can write an ideal square wave as an infinite series of the form

$$x_{\mathrm{square}}(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin\left((2k-1)2\pi f t\right)}{(2k-1)}$$

$$= \frac{4}{\pi} \left(\sin(2\pi f t) + \frac{1}{3} \sin(6\pi f t) + \frac{1}{5} \sin(10\pi f t) + \cdots \right).$$

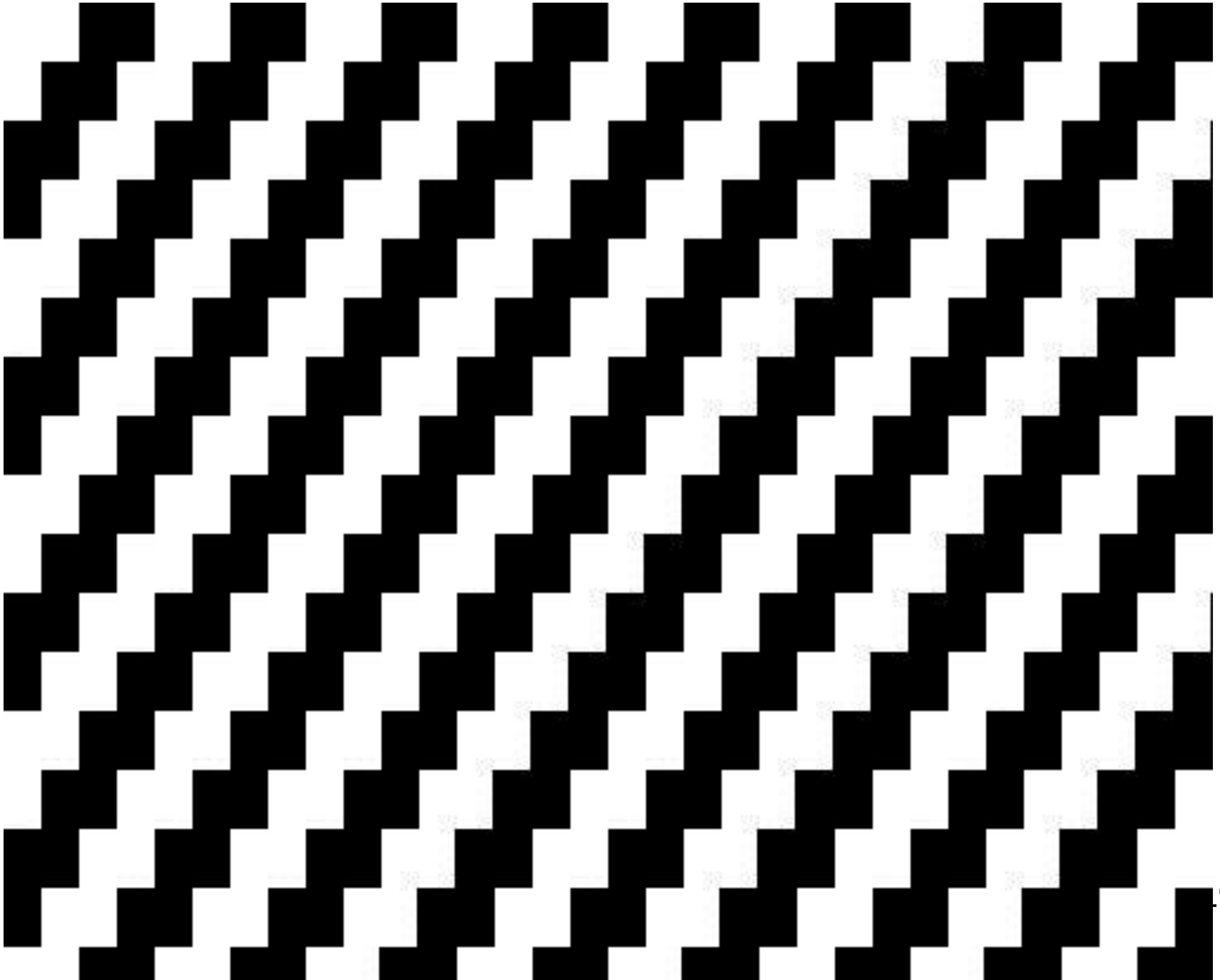
http://en.wikipedia.org/wiki/Square_wave

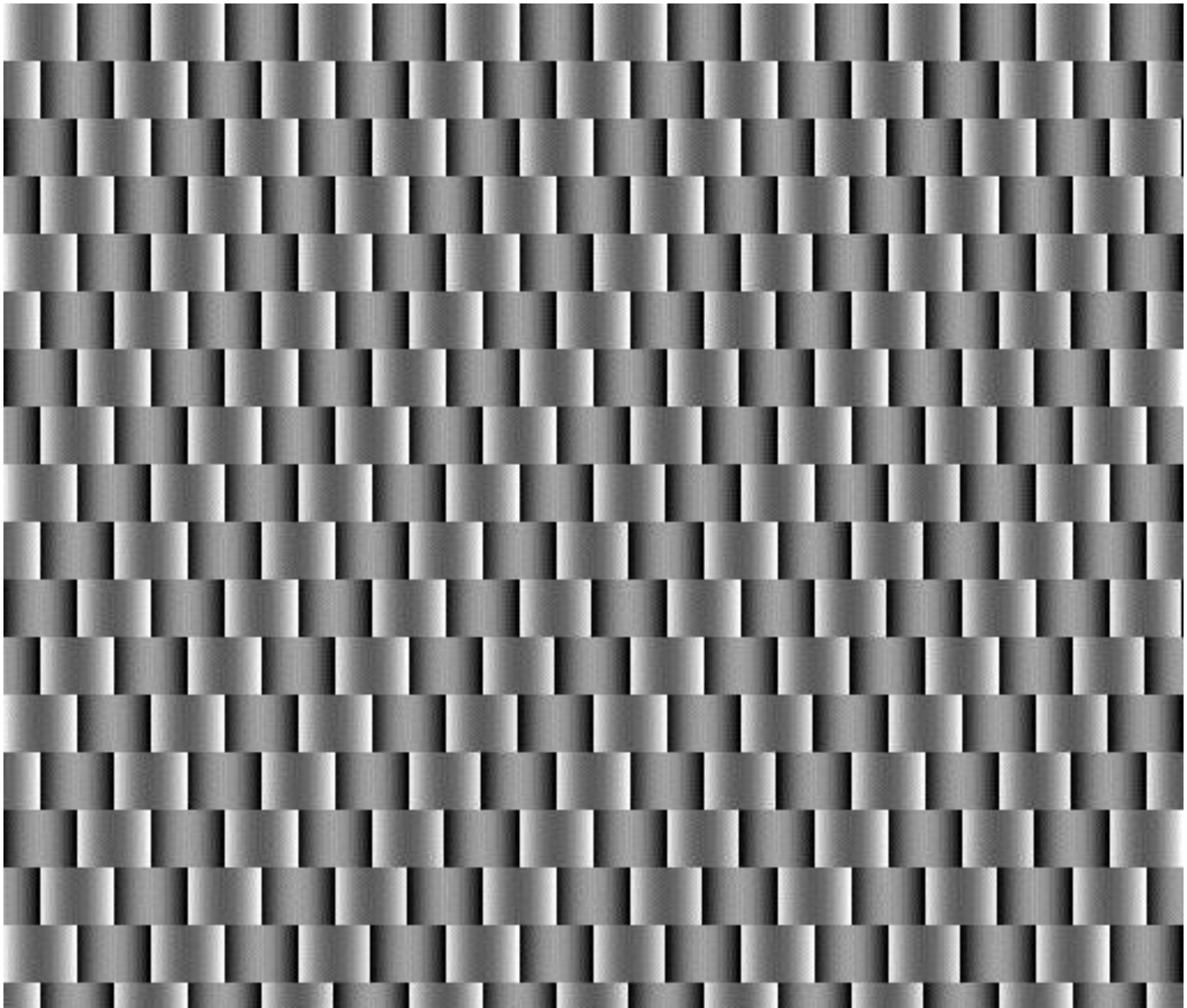
filters to analyze motion

QuickTime™ and a
GIF decompressor
are needed to see this picture.

QuickTime™ and a
decompressor
are needed to see this picture.

QuickTime™ and a
decompressor
are needed to see this picture.





Motion without movement

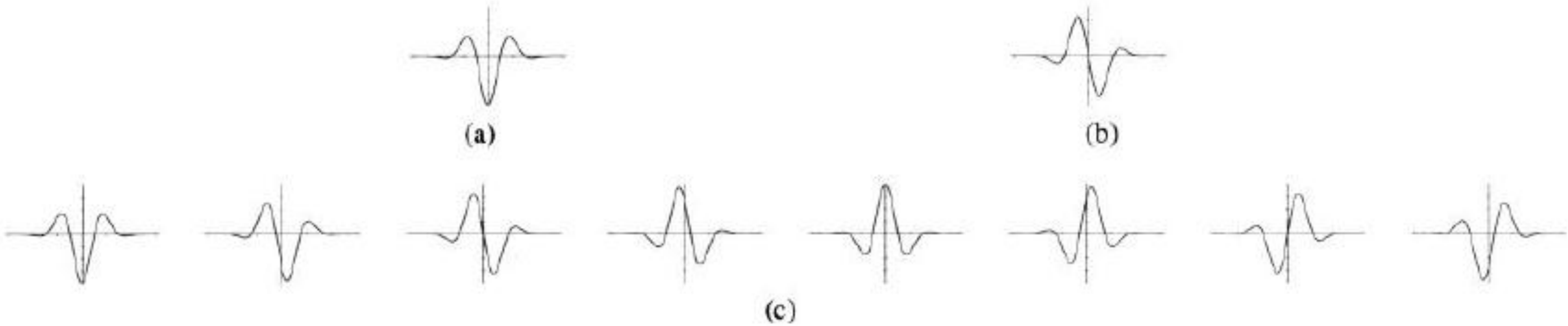


Figure 1: 1-d cross-sections of filters. (a) Even phase (G_2). (b) Odd phase (H_2). (c) Filters modulated in phase according to Eq. (1). Note the apparent rightward motion of the filter ripples.

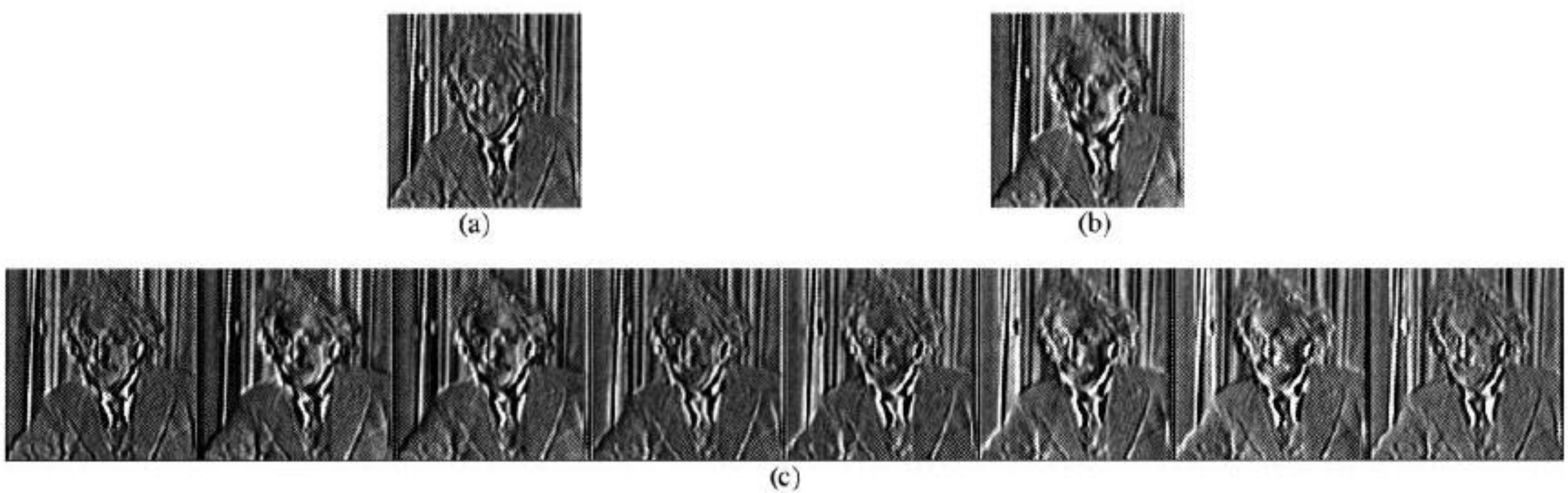


Figure 2: (a) and (b): G_2 and H_2 filters were applied to an image of Einstein. (c) Images modulated as in Eq. (1). When viewed as a temporal sequence, this generates the perception of rightward motion, yet image remains stationary.

G_2, H_2 basis filters

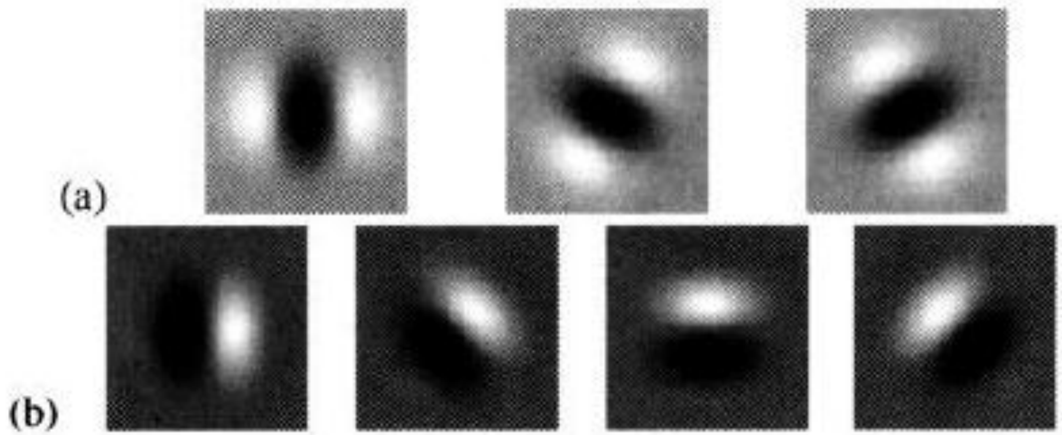
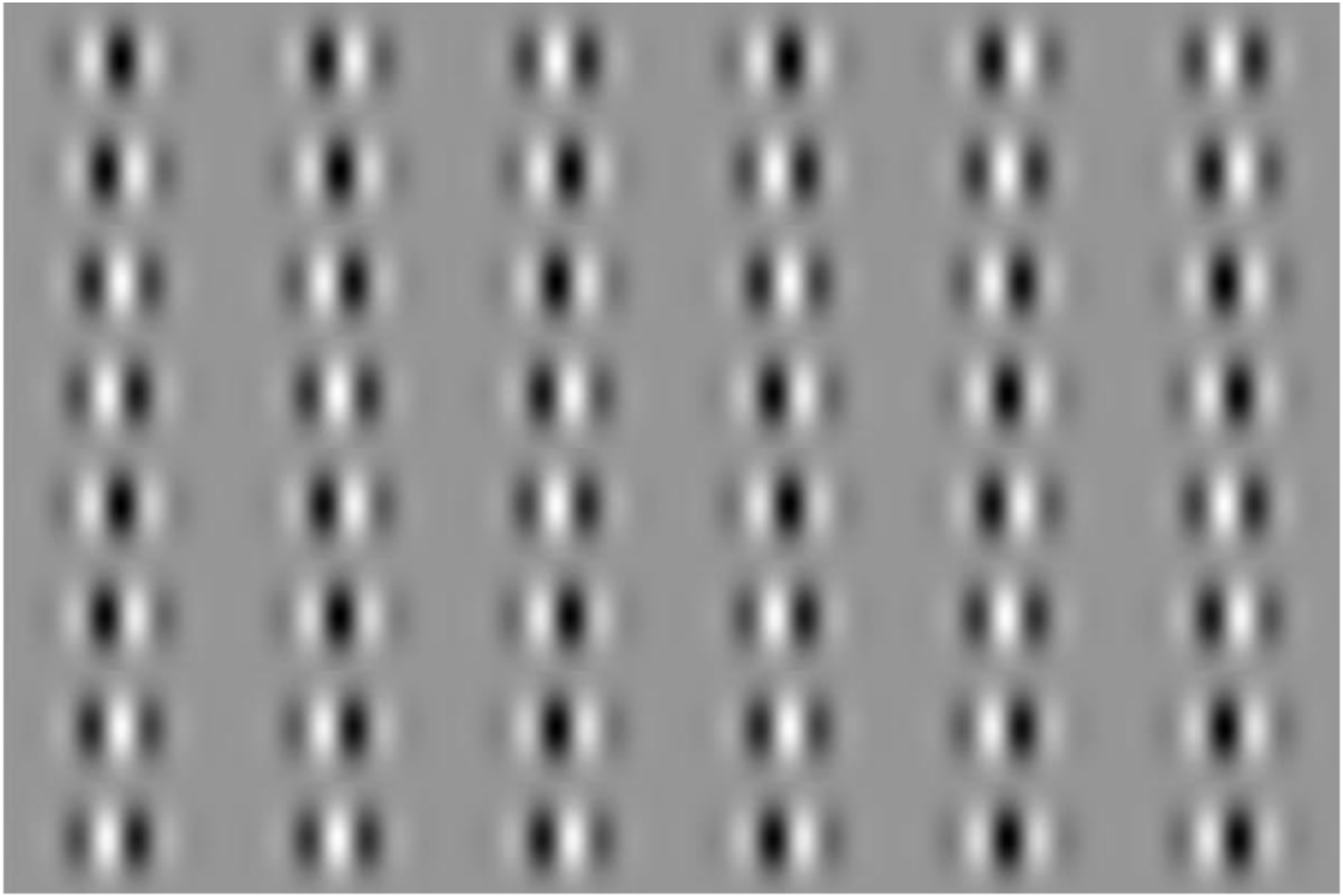


Figure 3: (a) G_2 and (b) H_2 quadrature pair steerable basis filters. The filter sets (a) and (b) span the space of all rotations of their respective filters.

C:\Documents and Settings\billf\My Documents\presentations\comp photog workshop lecture

Motion without movement

QuickTime™ and a
decompressor
are needed to see this picture.

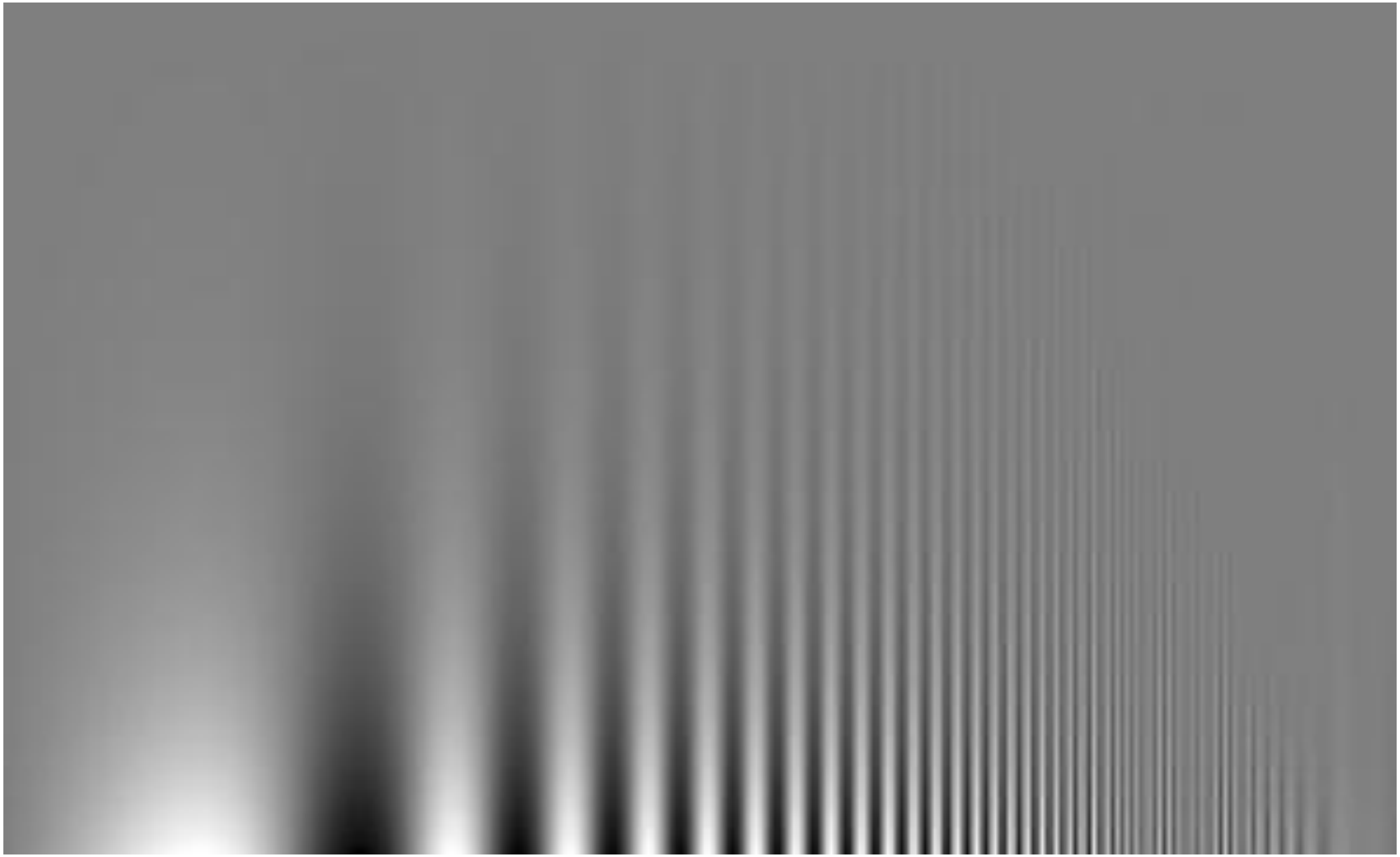


http://www.psy.ritsumei.ac.jp/~akitaoka/popple-e.html

Outline

- Linear filtering
- Fourier Transform
- Phase
- Sampling and Aliasing
- Spatially localized analysis
- Quadrature phase
- Oriented filters
- Motion analysis
- Human spatial frequency sensitivity

Contrast Sensitivity Function



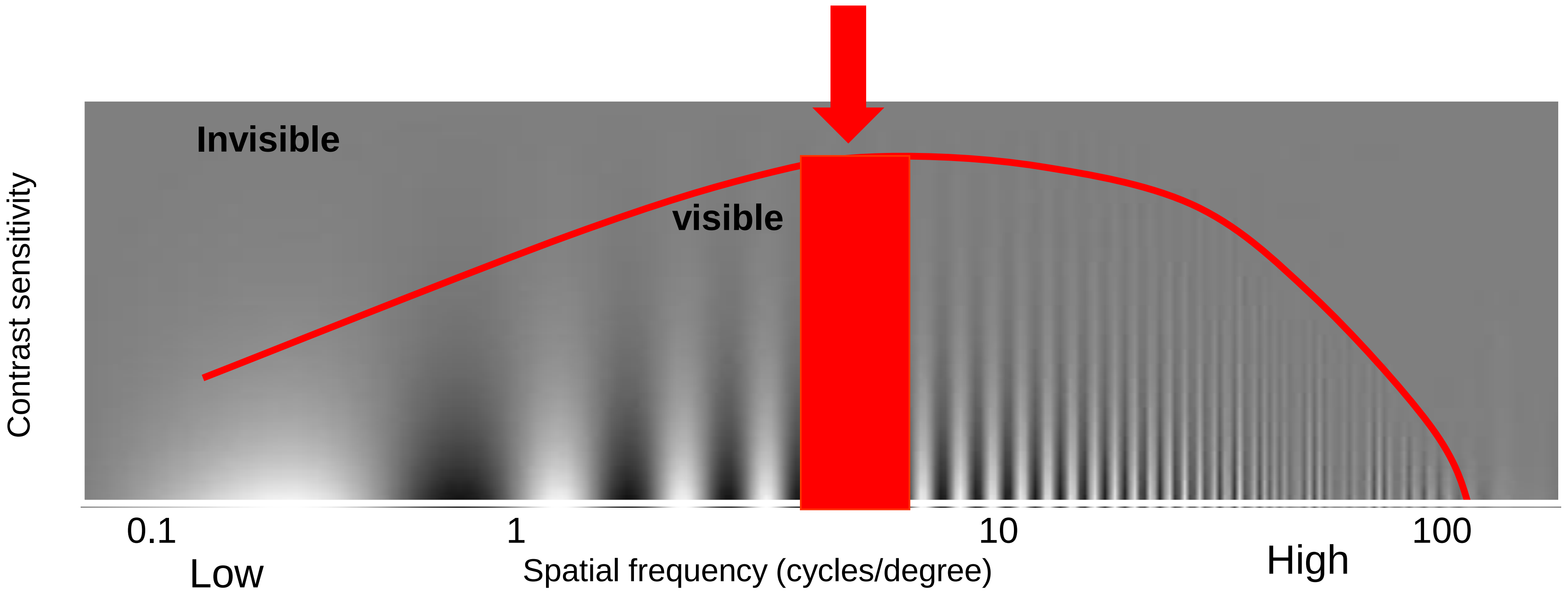
A demo of human contrast sensitivity as a function of spatial frequency. Frequency rises from left to right at a constant rate. Contrast drops from bottom to top at a constant rate. The bars are visible further up for middle frequencies, showing these are more salient to the human visual system.

Contrast Sensitivity Function

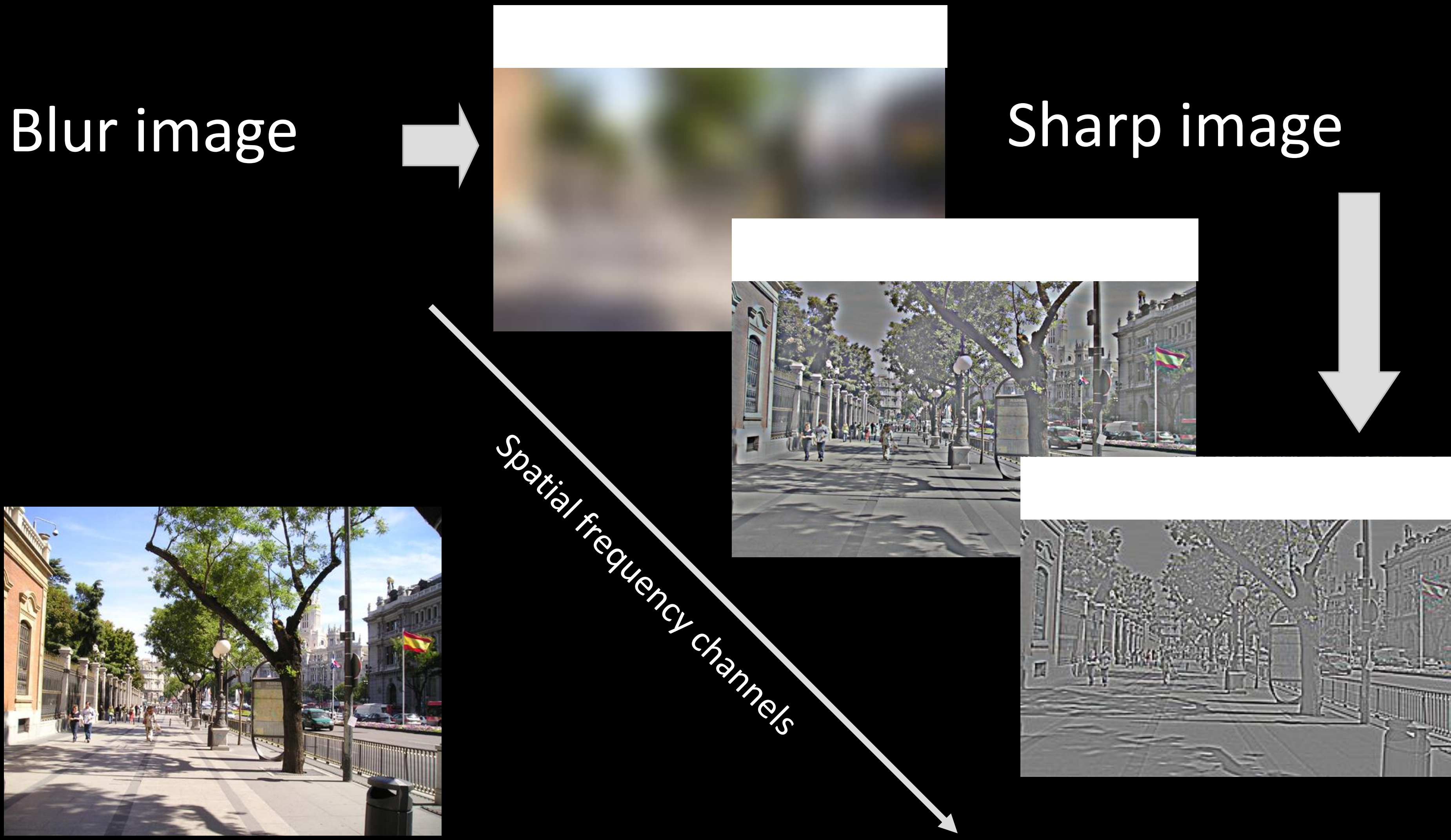
Blackmore & Campbell (1969)

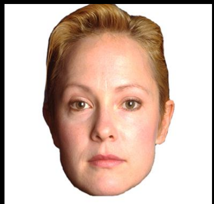
Maximum sensitivity

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

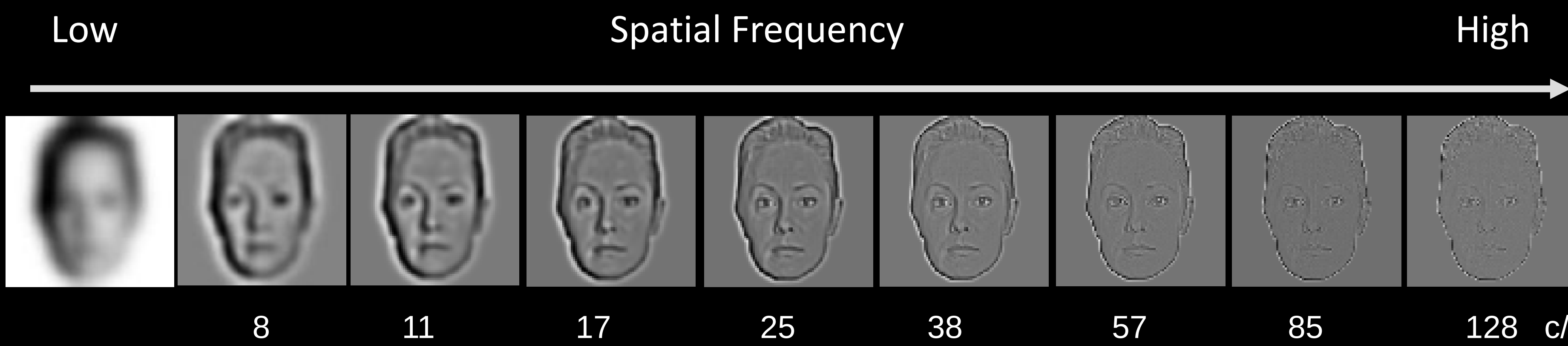


Human Visual Perception





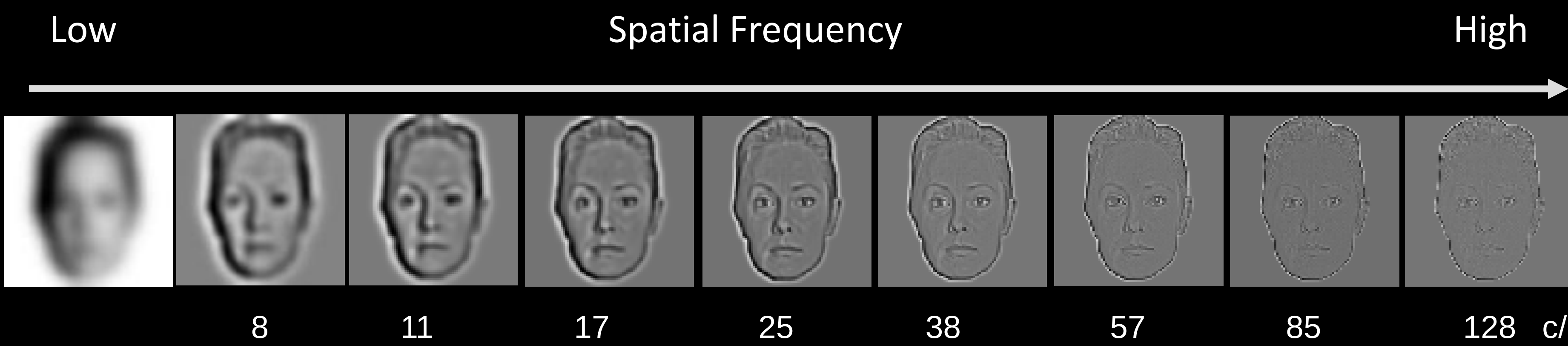
Multiscale subband decomposition

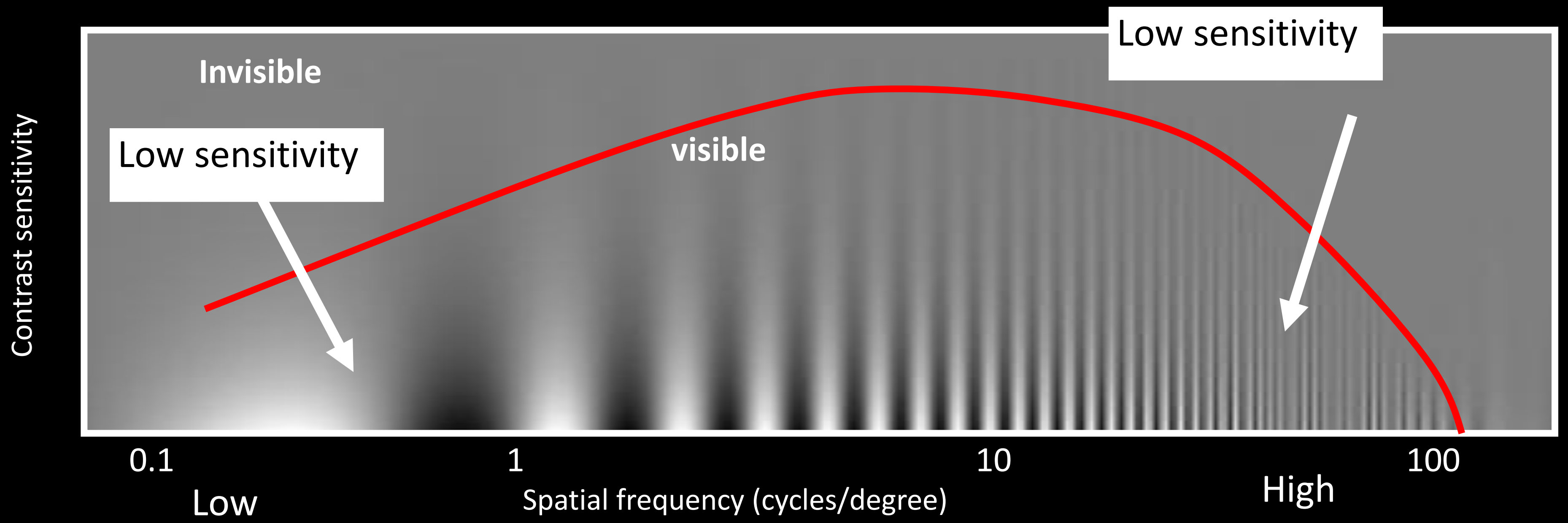


Burt, D.C. & Adelson E. (1983) *IEEE Trans. Com.*

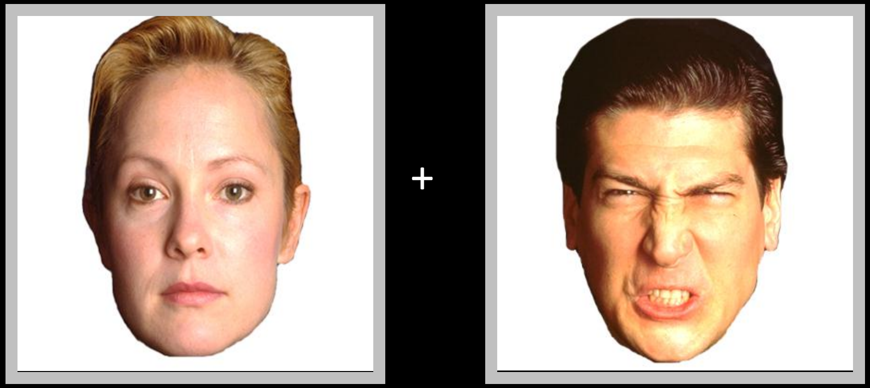
Contrast Sensitivity Function

Blackmore & Campbell (1969)





Perception of hybrid images



Perception of hybrid images

Oliva & Schyns

Male dominance

Female dominance

A man or a woman ?

Hybrid Images

Oliva & Schyns

Hybrid Images

Hybrid Images

Hybrid Images

Hybrid Images