MIT CSAIL

6.869: Advances in Computer Vision

Antonio Torralba, 2013

MIT COMPUTER VISION

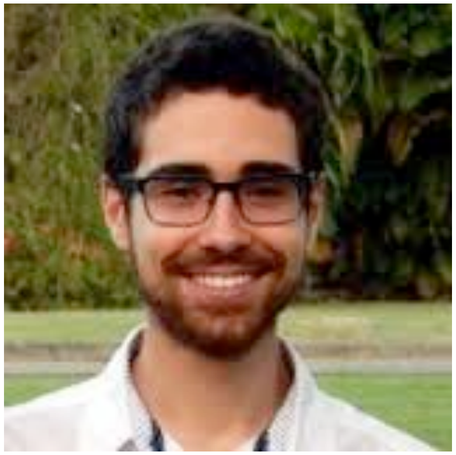
Lecture 2

Linear filters

Teaching Assistants

Adrià Recasens

Hang Zhao



Nicholas Hynes

Anying Li

Readings

Class notes

Please, send me comments about

-Anything wrong you might find
-Improving the clarity
-Suggestions for examples
-Suggestions for additional material

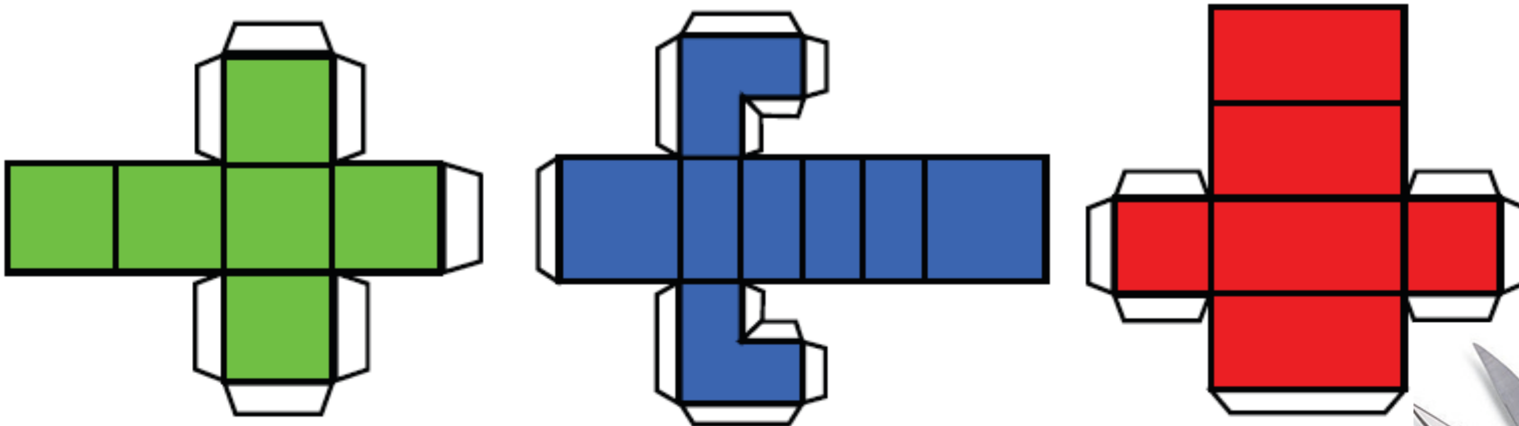
Matlab Tutorial

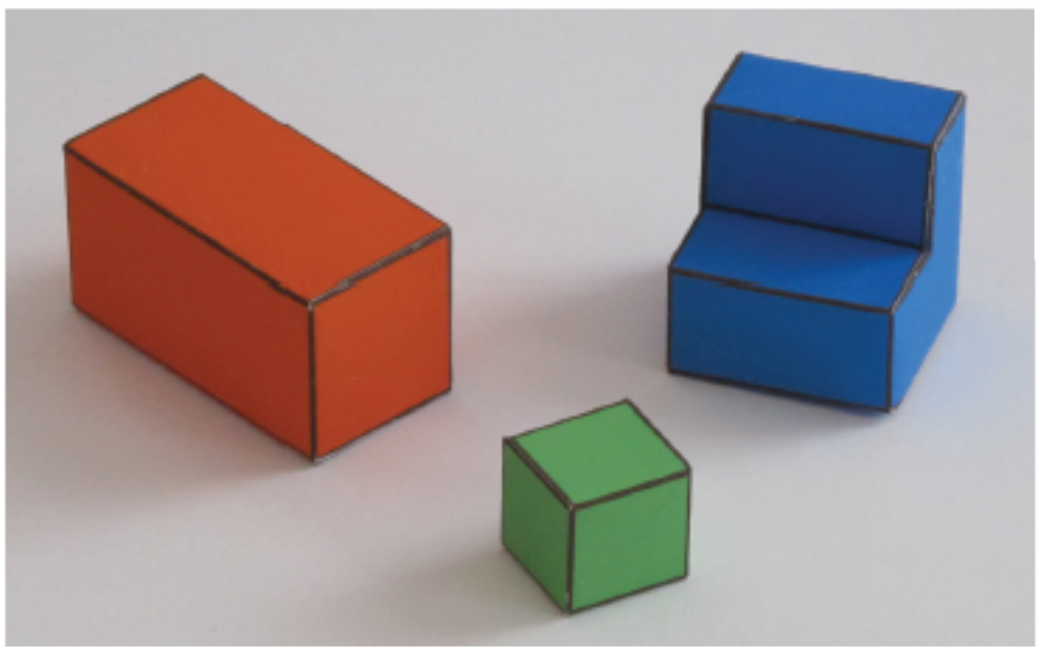
If you are a Matlab expert, no need to attend.

Wednesday, from 2pm to 3pm, 32-D463

Thursday, 5-6pm

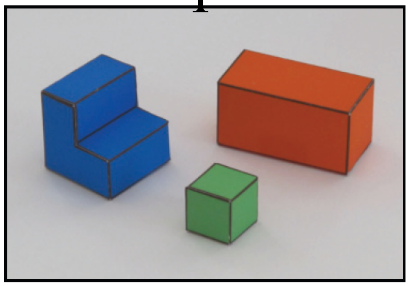
A Simple World



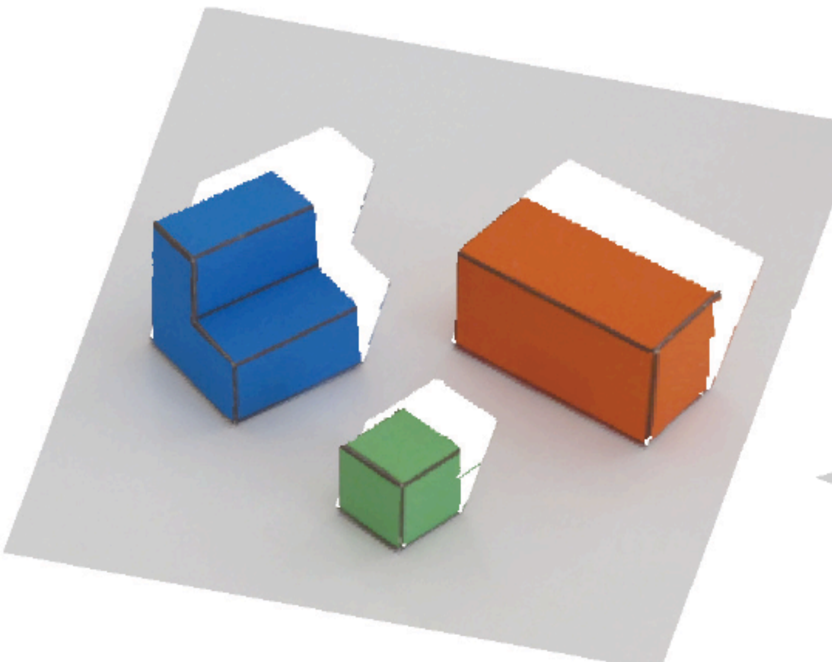


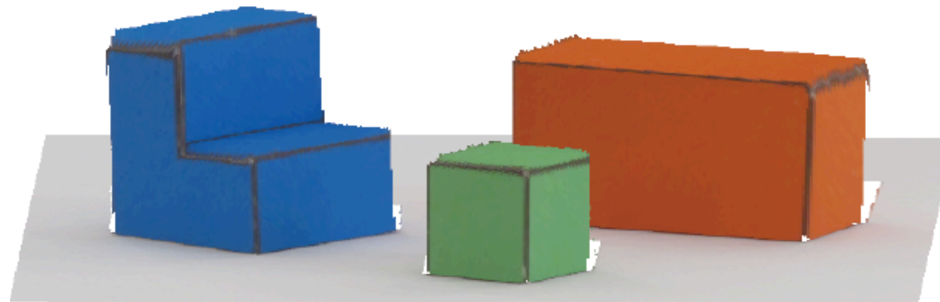
Changing view point

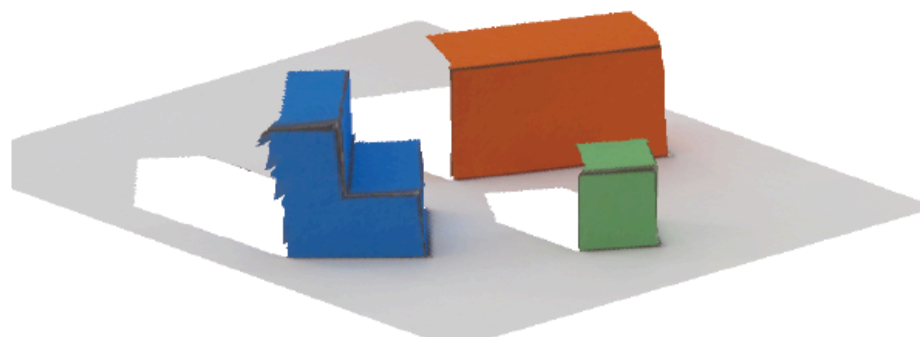
Input



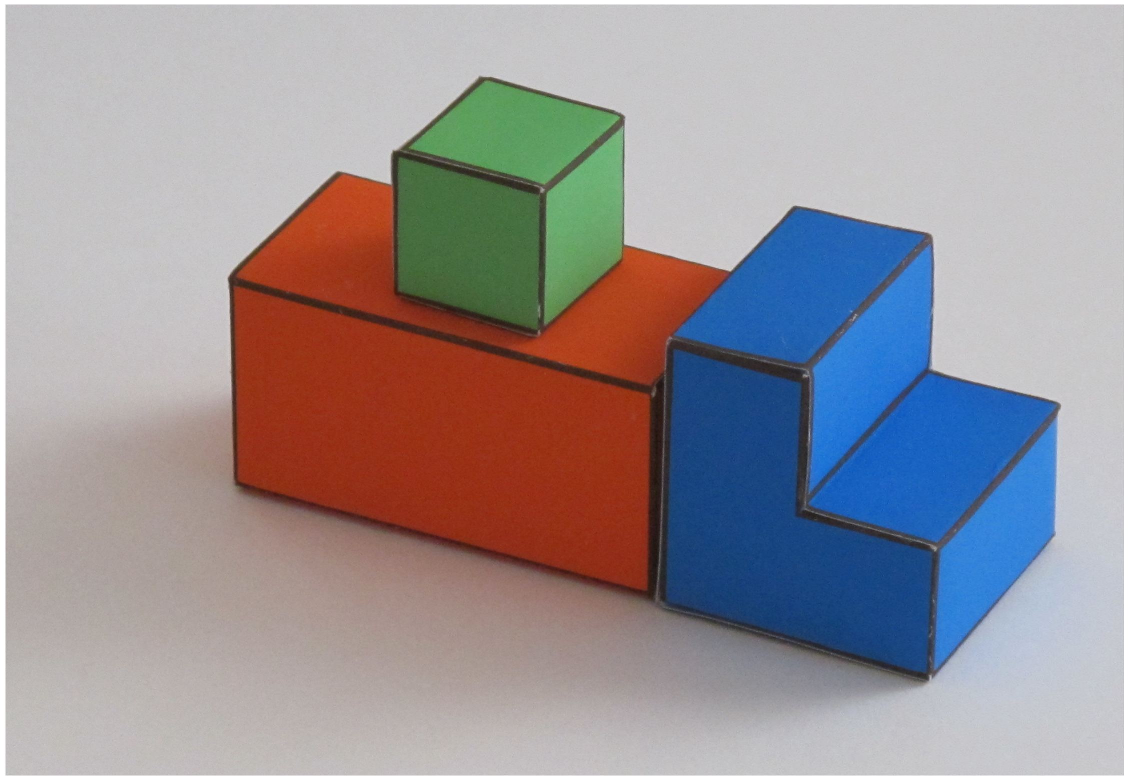
New view points:



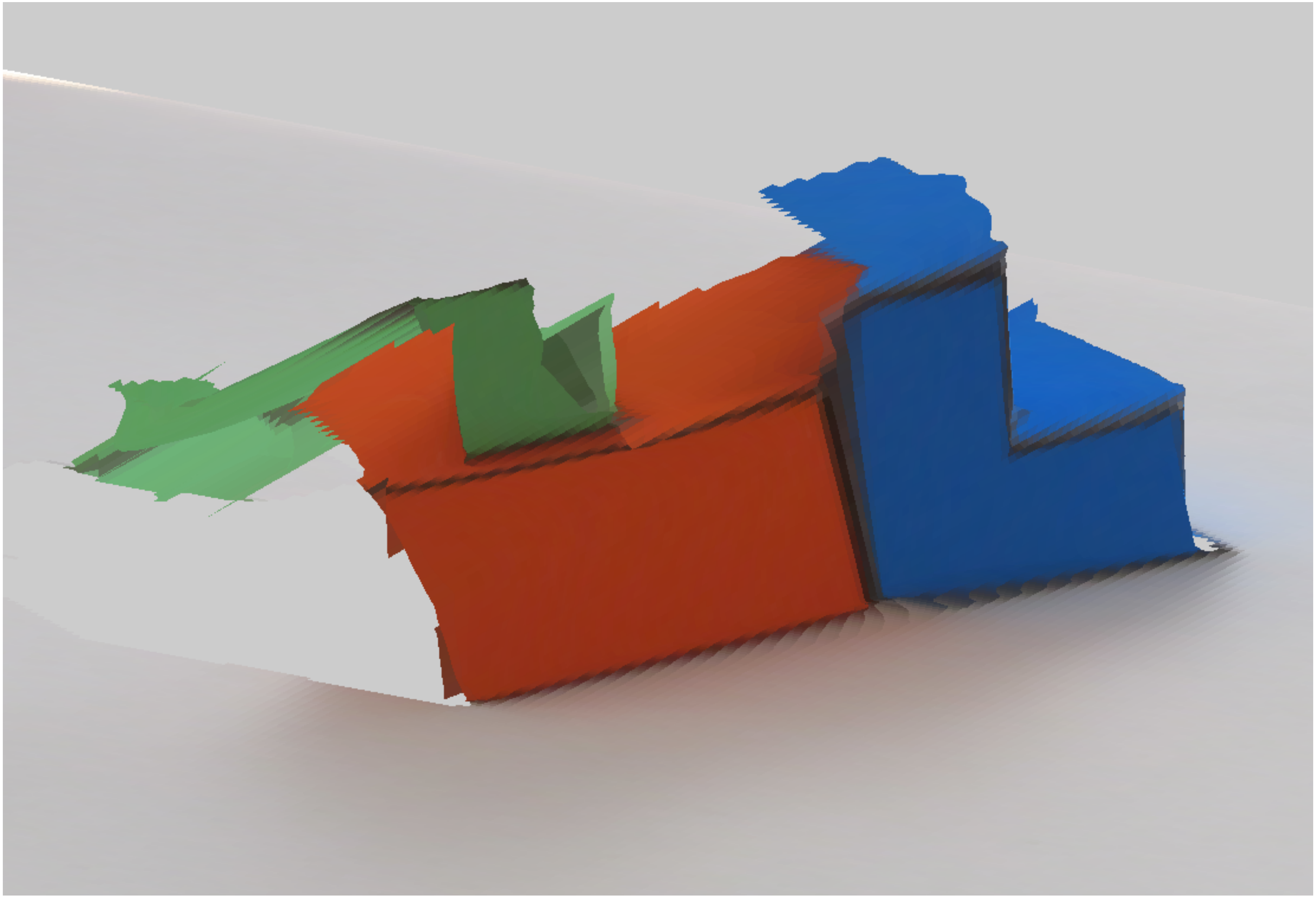




An inconvenient truth



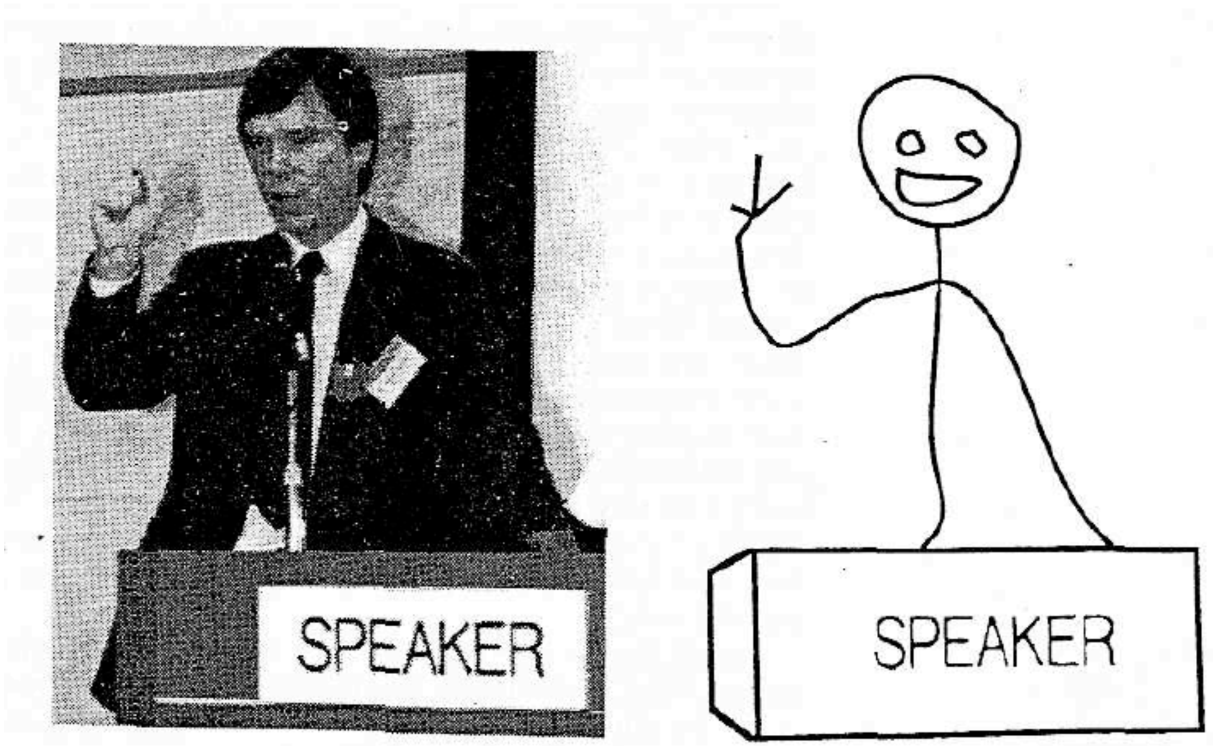
An inconvenient truth



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

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

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

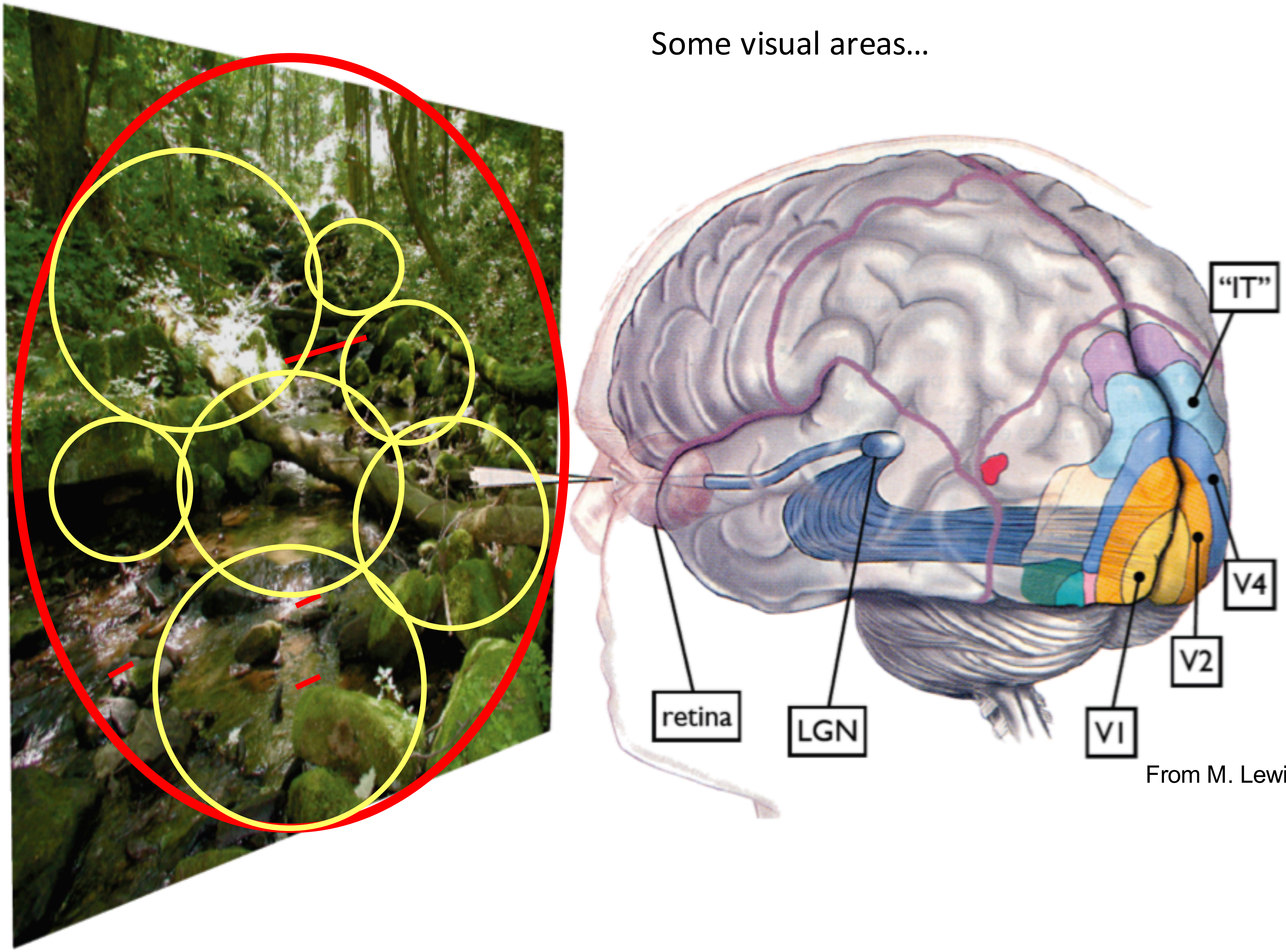


What we think we see

What we really see

Cavanagh, Perception 95

Some visual areas...

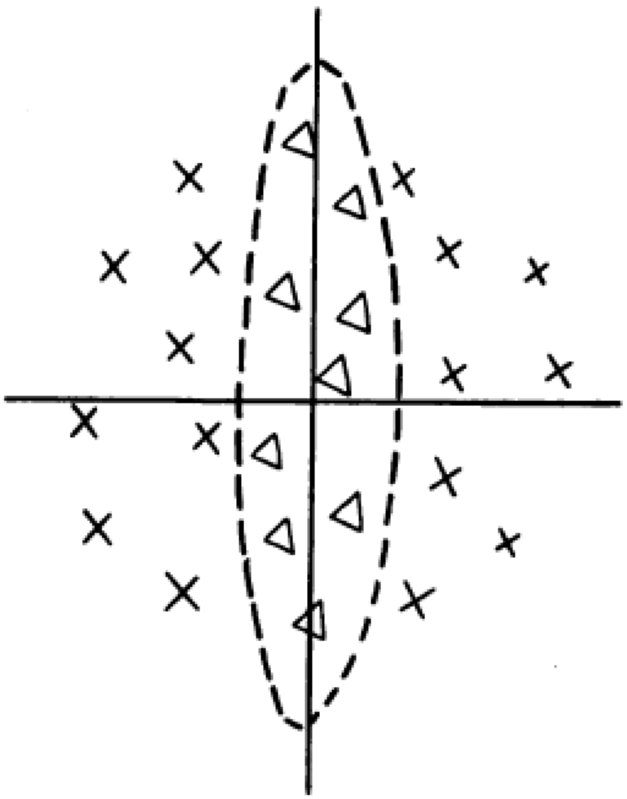


From M. Lewicky

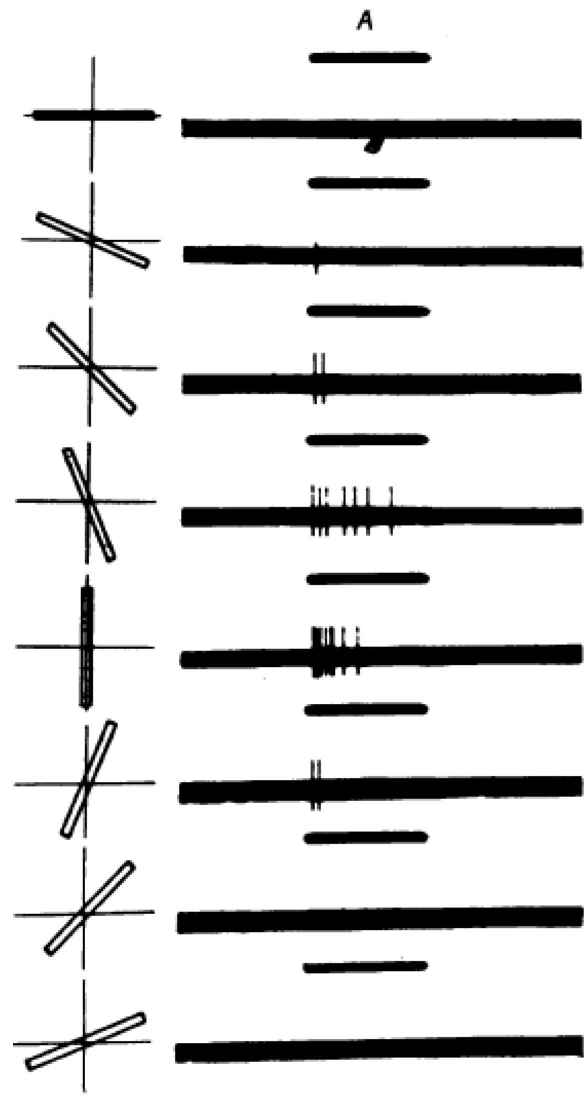
J. Physiol. (1959) 148, 574–591

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

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

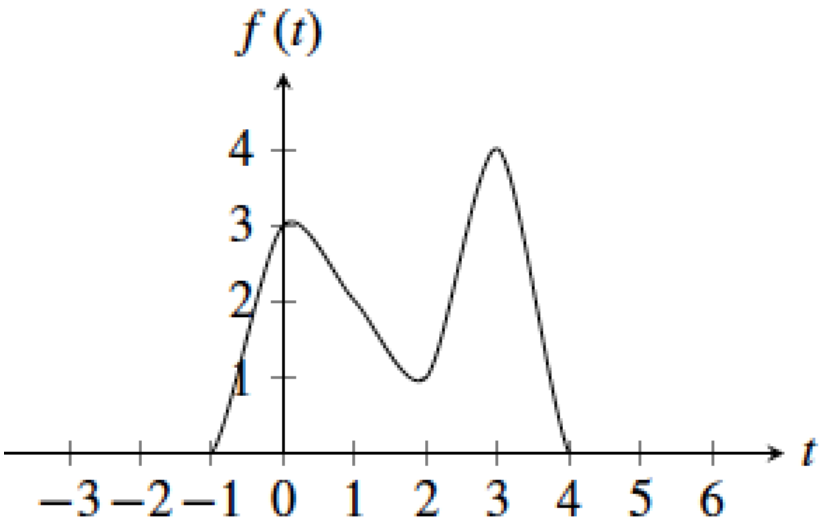


Receptive field
of a cell in the cat's cortex

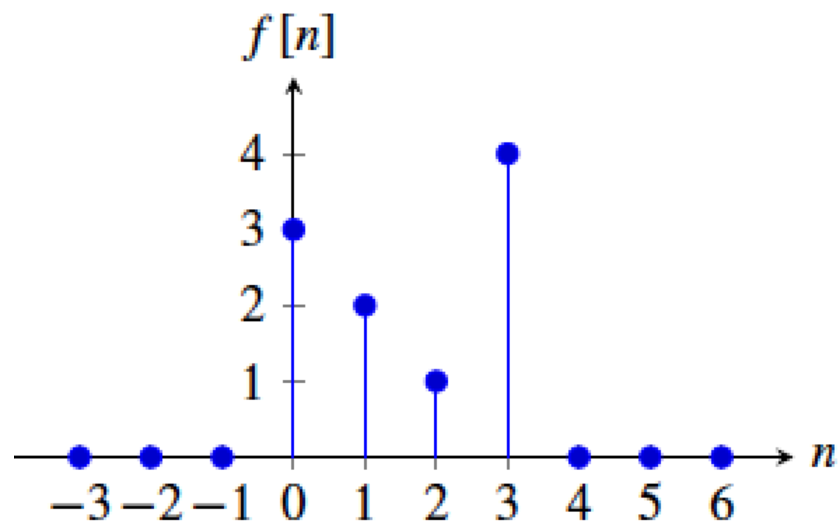


Responses to an oriented bar

Signals and systems

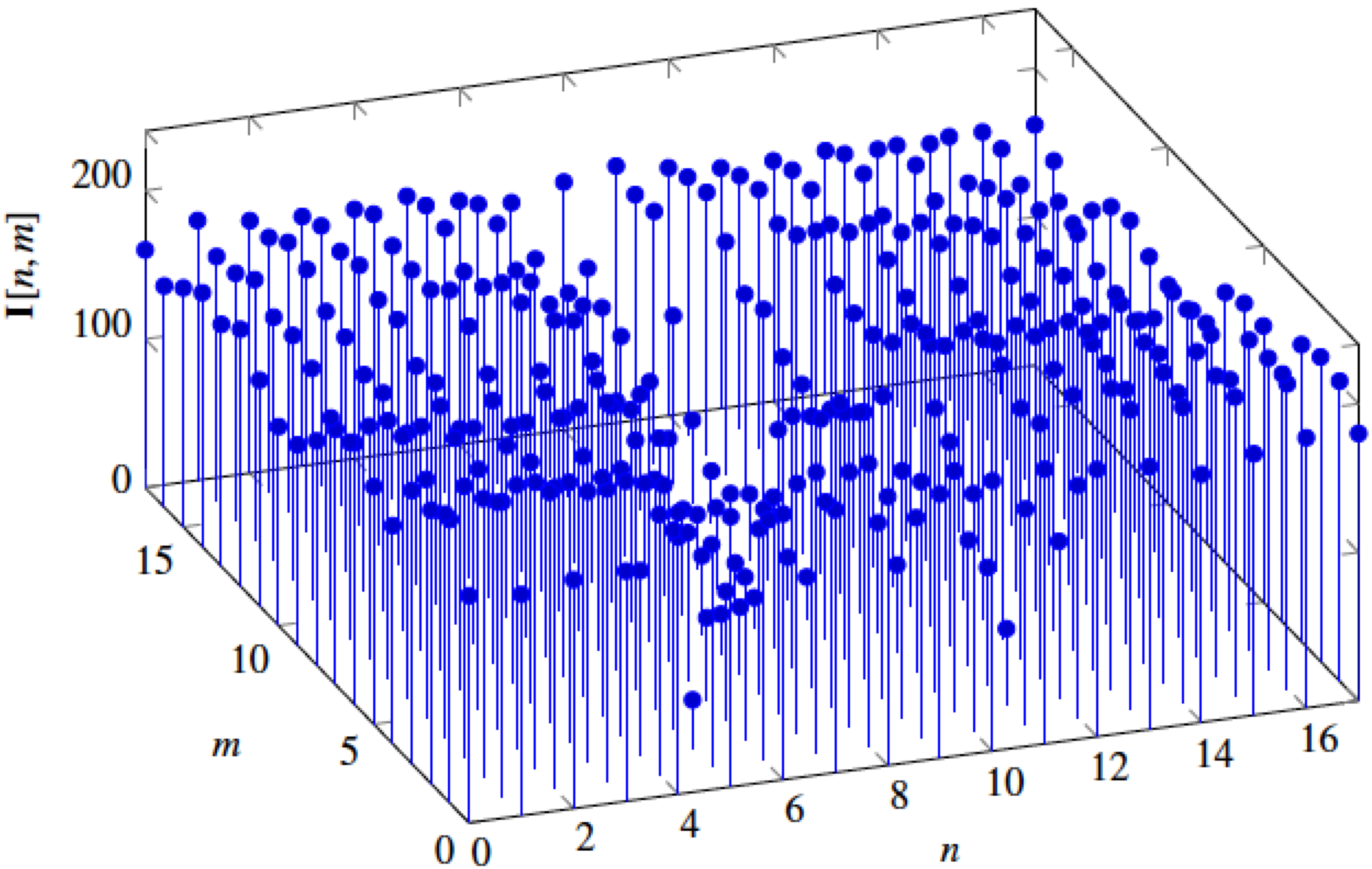


Time continuous signal

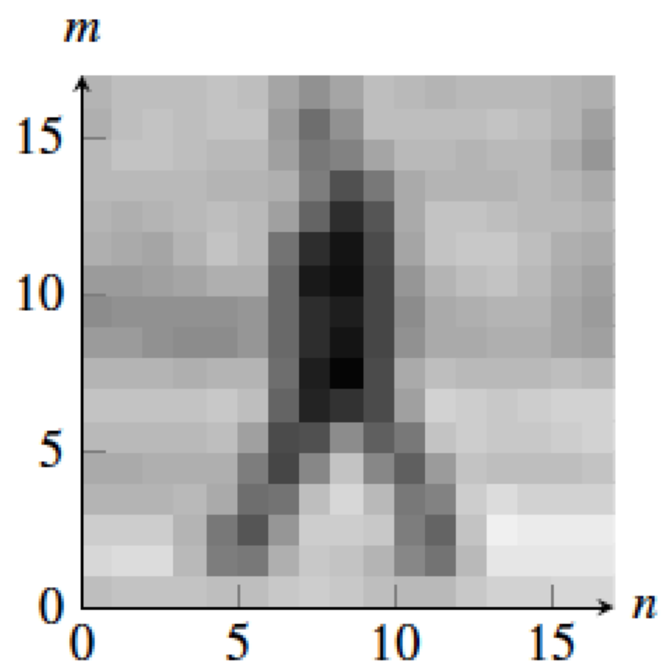


Time discrete signal

A 2D discrete signal



$$
\mathbf{I} = \begin{bmatrix}
160 & 175 & 171 & 168 & 168 & 172 & 164 & 158 & 167 & 173 & 167 & 163 & 162 & 164 & 160 & 159 & 163 & 162 \\
149 & 164 & 172 & 175 & 178 & 179 & 176 & 118 & 97 & 168 & 175 & 171 & 169 & 175 & 176 & 177 & 165 & 152 \\
161 & 166 & 182 & 171 & 170 & 177 & 175 & 116 & 109 & 169 & 177 & 173 & 168 & 175 & 175 & 159 & 153 & 123 \\
171 & 174 & 177 & 175 & 167 & 161 & 157 & 138 & 103 & 112 & 157 & 164 & 159 & 160 & 165 & 169 & 148 & 144 \\
163 & 163 & 162 & 165 & 167 & 164 & 178 & 167 & 77 & 55 & 134 & 170 & 167 & 162 & 164 & 175 & 168 & 160 \\
173 & 164 & 158 & 165 & 180 & 180 & 150 & 89 & 61 & 34 & 137 & 186 & 186 & 182 & 175 & 165 & 160 & 164 \\
152 & 155 & 146 & 147 & 169 & 180 & 163 & 51 & 24 & 32 & 119 & 163 & 175 & 182 & 181 & 162 & 148 & 153 \\
134 & 135 & 147 & 149 & 150 & 147 & 148 & 62 & 36 & 46 & 114 & 157 & 163 & 167 & 169 & 163 & 146 & 147 \\
135 & 132 & 131 & 125 & 115 & 129 & 132 & 74 & 54 & 41 & 104 & 156 & 152 & 156 & 164 & 156 & 141 & 144 \\
151 & 155 & 151 & 145 & 144 & 149 & 143 & 71 & 31 & 29 & 129 & 164 & 157 & 155 & 159 & 158 & 156 & 148 \\
172 & 174 & 178 & 177 & 177 & 181 & 174 & 54 & 21 & 29 & 136 & 190 & 180 & 179 & 176 & 184 & 187 & 182 \\
177 & 178 & 176 & 173 & 174 & 180 & 150 & 27 & 101 & 94 & 74 & 189 & 188 & 186 & 183 & 186 & 188 & 187 \\
160 & 160 & 163 & 163 & 161 & 167 & 100 & 45 & 169 & 166 & 59 & 136 & 184 & 176 & 175 & 177 & 185 & 186 \\
147 & 150 & 153 & 155 & 160 & 155 & 56 & 111 & 182 & 180 & 104 & 84 & 168 & 172 & 171 & 164 & 168 & 167 \\
184 & 182 & 178 & 175 & 179 & 133 & 86 & 191 & 201 & 204 & 191 & 79 & 172 & 220 & 217 & 205 & 209 & 200 \\
184 & 187 & 192 & 182 & 124 & 32 & 109 & 168 & 171 & 167 & 163 & 51 & 105 & 203 & 209 & 203 & 210 & 205 \\
191 & 198 & 203 & 197 & 175 & 149 & 169 & 189 & 190 & 173 & 160 & 145 & 156 & 202 & 199 & 201 & 205 & 202 \\
153 & 149 & 153 & 155 & 173 & 182 & 179 & 177 & 182 & 177 & 182 & 185 & 179 & 177 & 167 & 176 & 182 & 180
\end{bmatrix}
$$



A tiny person of 18 x 18 pixels

Signal / image space

Scalar product between two signals f, g :

$$\langle f, g \rangle = \sum_{n=0}^{N-1} f[n]\, g^*[n] = f^T g^*$$

L2 norm of f :

$$E_f = \|f\|^2 = \langle f, f \rangle = \sum_{n=0}^{N-1} |f[n]|^2 = f^T f^*$$

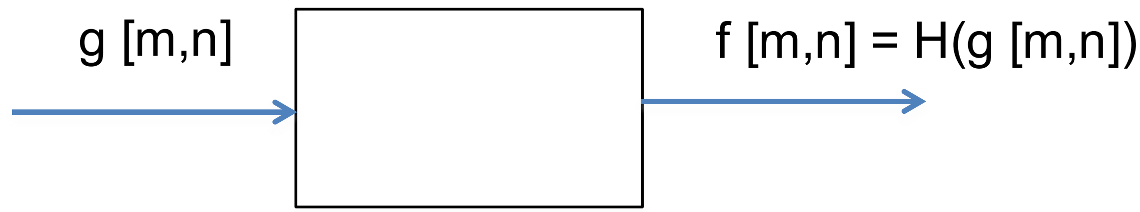
Distance between two signals f, g :

$$d_{f,g}^2 = \|f - g\|^2 = \sum_{n=0}^{N-1} |f[n] - g[n]|^2 = E_f + E_g - 2\langle f, g \rangle$$

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 filter to be linear, it has to verify:

$$f[m,n] = H(a[m,n] + b[m,n]) = H(a[m,n]) + H(b[m,n])$$

$$f[m,n] = H(C\,a[m,n]) = C\,H(a[m,n])$$

Linear filtering

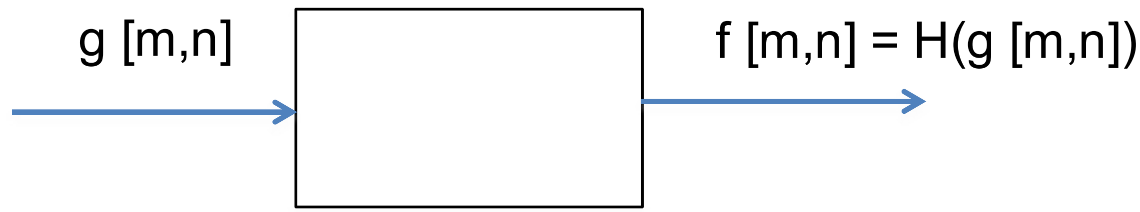
A linear filter in its most general form can be written as, in 1D for a signal of length N:

$$f[n] = \sum_{k=0}^{N-1} h[n,k]\, g[k]$$

It is useful to make it more explicit by writing:

$$\begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[M] \end{bmatrix} = \begin{bmatrix} h[0,0] & h[0,1] & \ldots & h[0,N] \\ h[1,0] & h[1,1] & \ldots & h[1,N] \\ \vdots & \vdots & \vdots & \vdots \\ h[M,N] & h[M,1] & \ldots & h[M,N] \end{bmatrix} \begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[N] \end{bmatrix}$$

Linear filtering



In 2D:

$$f[n,m] = \sum_{k,\,l=0}^{N-1,\,M-1} h[n,m,k,l]\, g[k,l]$$

Which can also be written in matrix form as in the 1D case:

$$\begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[M] \end{bmatrix} = \begin{bmatrix} h[0,0] & h[0,1] & \dots & h[0,N] \\ h[1,0] & h[1,1] & \dots & h[1,N] \\ \vdots & \vdots & \vdots & \vdots \\ h[M,N] & h[M,1] & \dots & h[M,N] \end{bmatrix} \begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[N] \end{bmatrix}$$

Why should one pixel be treated differently than any another?

Credit picture: Fredo Durand

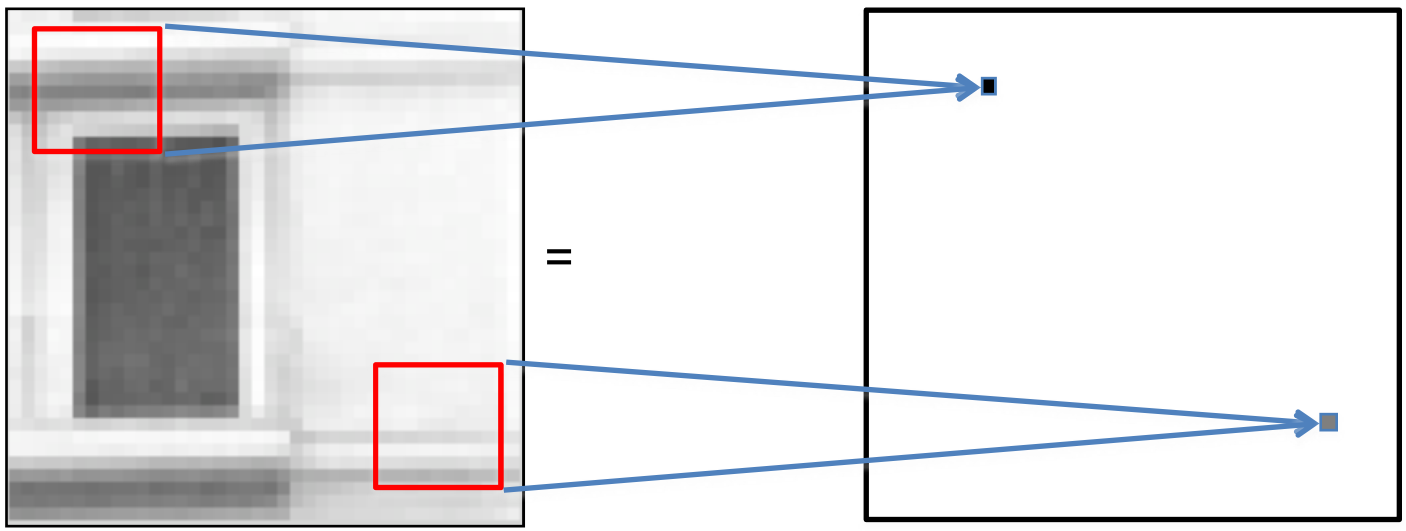
A translation invariant filter

Example: The output for the sample n is twice the value of the input at that same time minus the sum of the two previous time steps

$$f[0] = 2g[0] - g[-1] - g[-2]$$
$$f[1] = 2g[1] - g[0] - g[-1]$$
$$f[2] = 2g[2] - g[1] - g[0]$$
$$\ldots$$
$$f[n] = 2g[n] - g[n-1] - g[n-2]$$

A filter is linear translation invariant (LTI) if it is linear and when we translate the input signal by m samples, the output is also translated by m samples.

A translation invariant filter

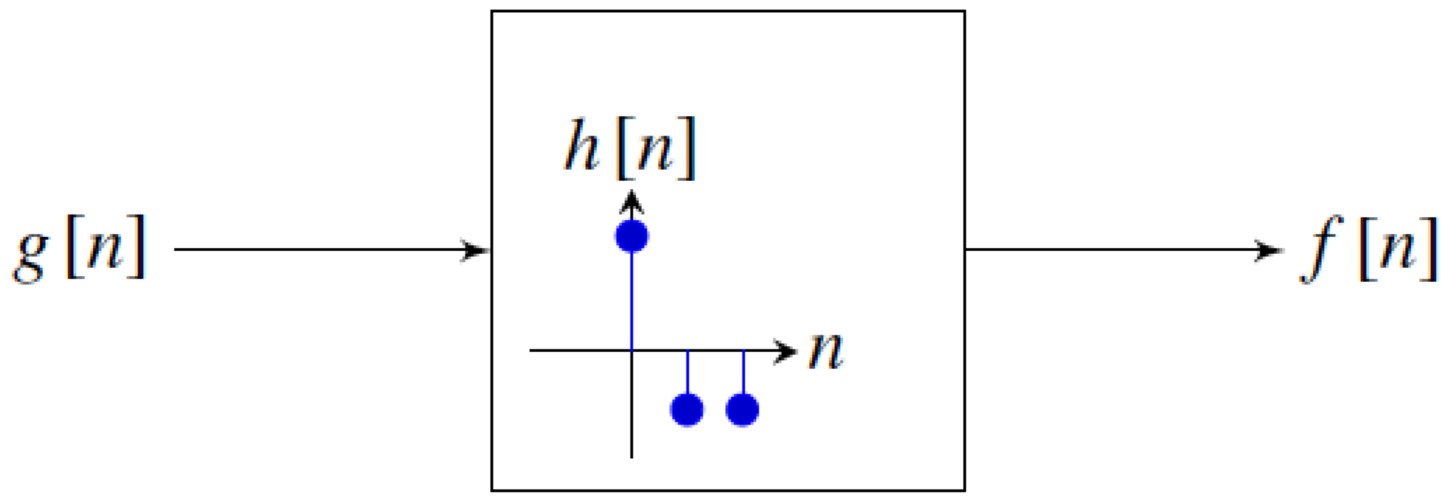


The same weighting occurs within each window

Convolution

$$f[n] = h \circ g = \sum_{k=0}^{N-1} h[n-k]\, g[k]$$

For the previous example: h = [2, -1, -1]

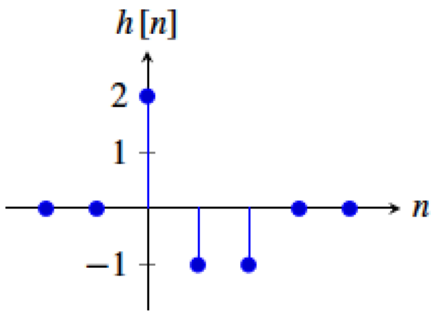


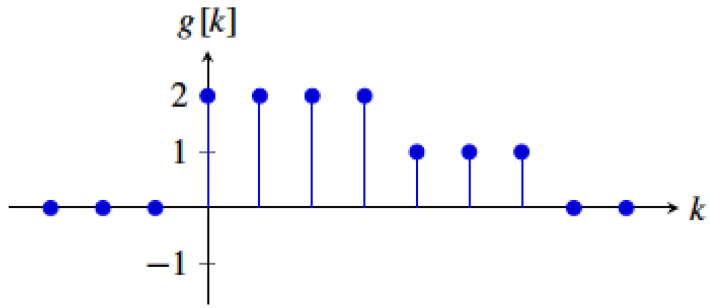
Convolution

In the 1D case, it helps to make explicit the structure of the matrix:

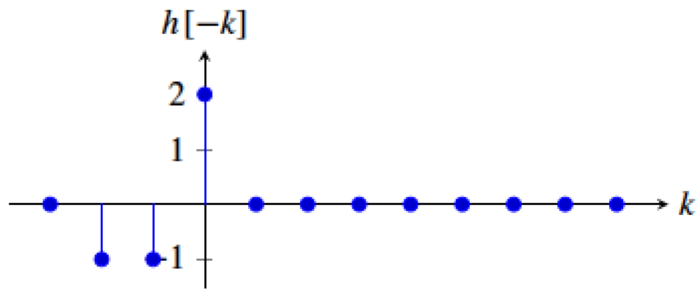
$$\begin{bmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N] \end{bmatrix} = \begin{bmatrix} h[0] & h[-1] & h[-2] & \dots & h[-N] \\ h[1] & h[0] & h[-1] & \dots & h[1-N] \\ h[2] & h[1] & h[0] & \dots & h[2-N] \\ \vdots & \vdots & \vdots & \vdots & \\ h[N] & h[N-1] & h[N-2] & \dots & h[0] \end{bmatrix} \begin{bmatrix} g[0] \\ g[1] \\ g[2] \\ \vdots \\ g[N] \end{bmatrix}$$

Convolution



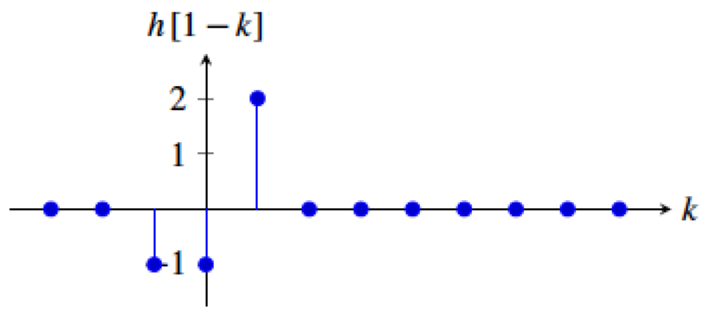


$f[0] = \sum_k h[-k]\, g[k]$



$f[0] = 4$

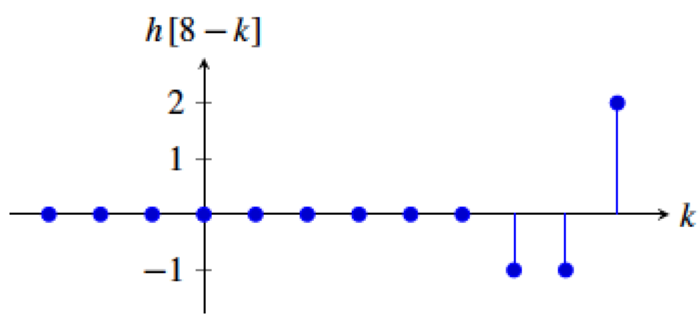
$f[1] = \sum_k h[1-k]\, g[k]$



$f[1] = 2$

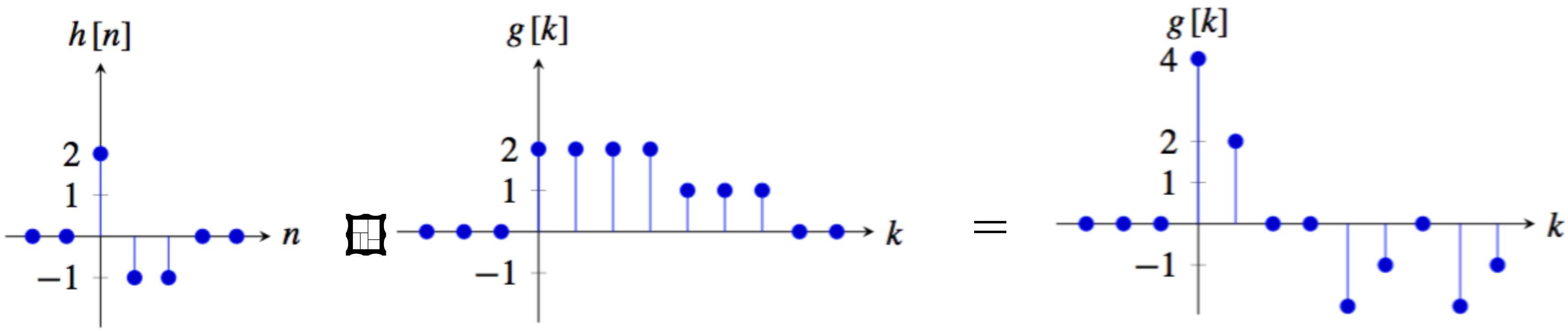
$\vdots$

$f[8] = \sum_k h[8-k]\, g[k]$



$f[8] = -1$

Convolution



Properties of the convolution

Commutative

$$h[n] \circ g[n] = g[n] \circ h[n]$$

Associative

$$h[n] \circ g[n] \circ q[n] = h[n] \circ (g[n] \circ q[n]) = (h[n] \circ g[n]) \circ q[n]$$

Distributive with respect to the sum

$$h[n] \circ (f[n] + g[n]) = h[n] \circ f[n] + h[n] \circ g[n]$$

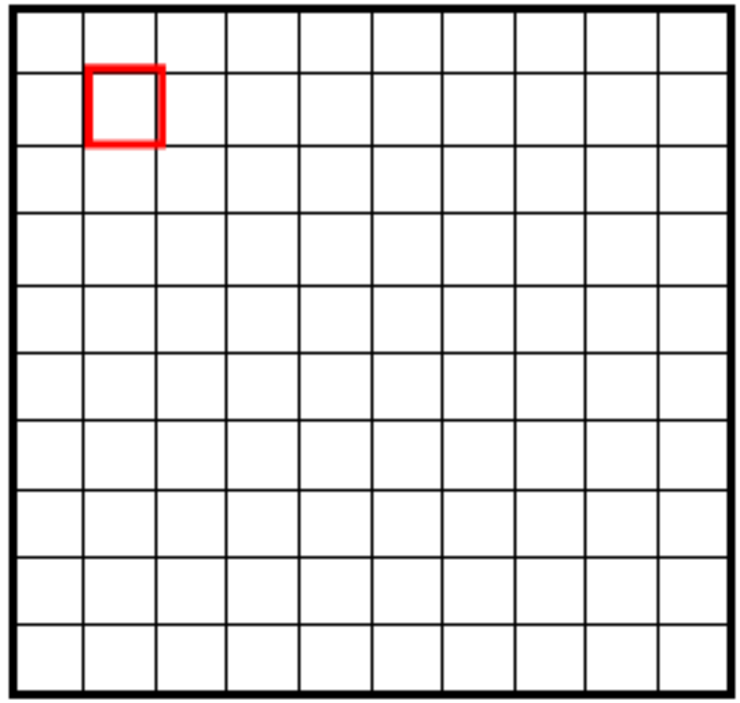
Shift property

$$f[n - n_0] = h[n] \circ g[n - n_0] = h[n - n_0] \circ g[n]$$

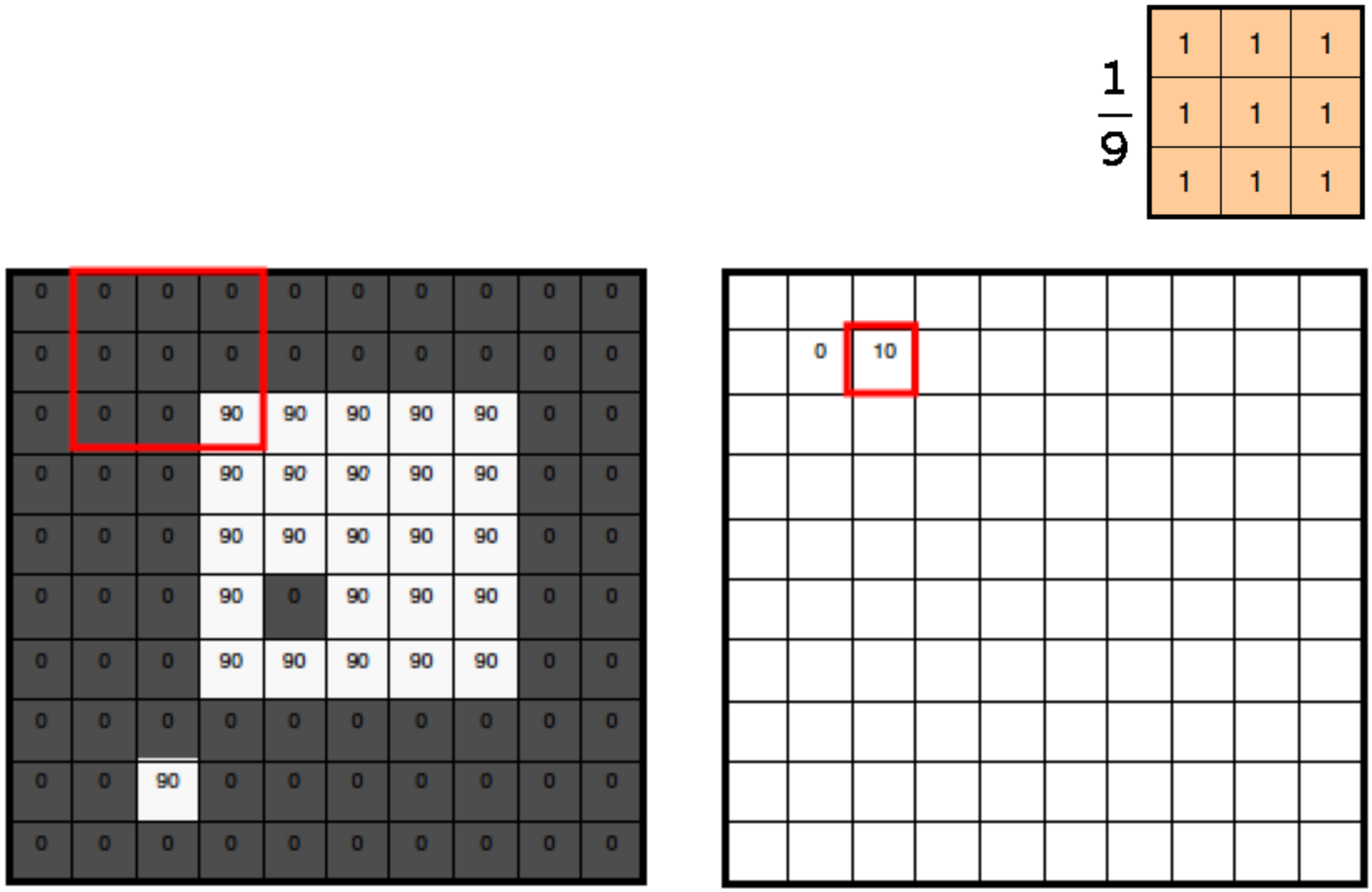
2D convolution

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

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



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



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

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

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						

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

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				
				?					

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

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				
						?			
				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)

2D convolution

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

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]

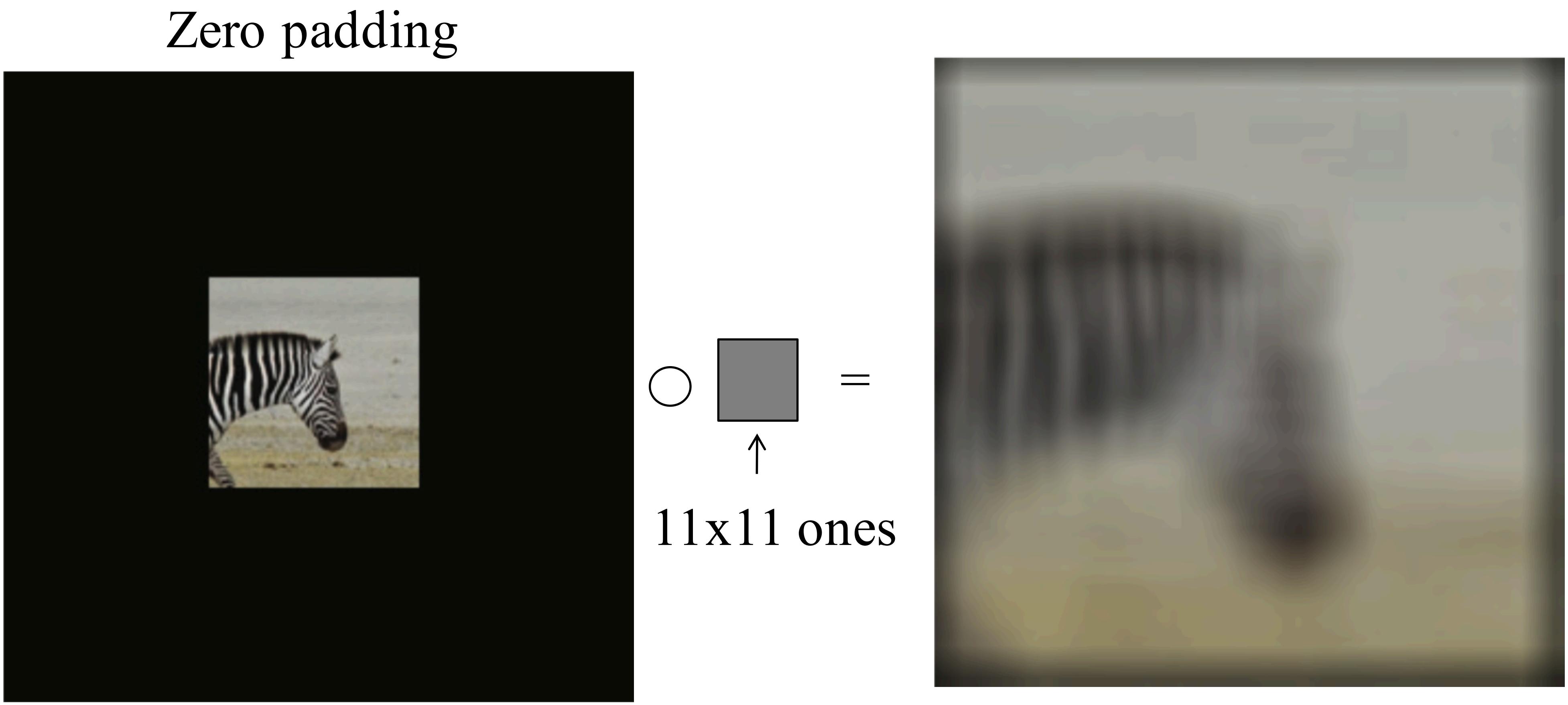
=

?	?	?	?	?	?	?	?
?	-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	?
?	?	?	?	?	?	?	?

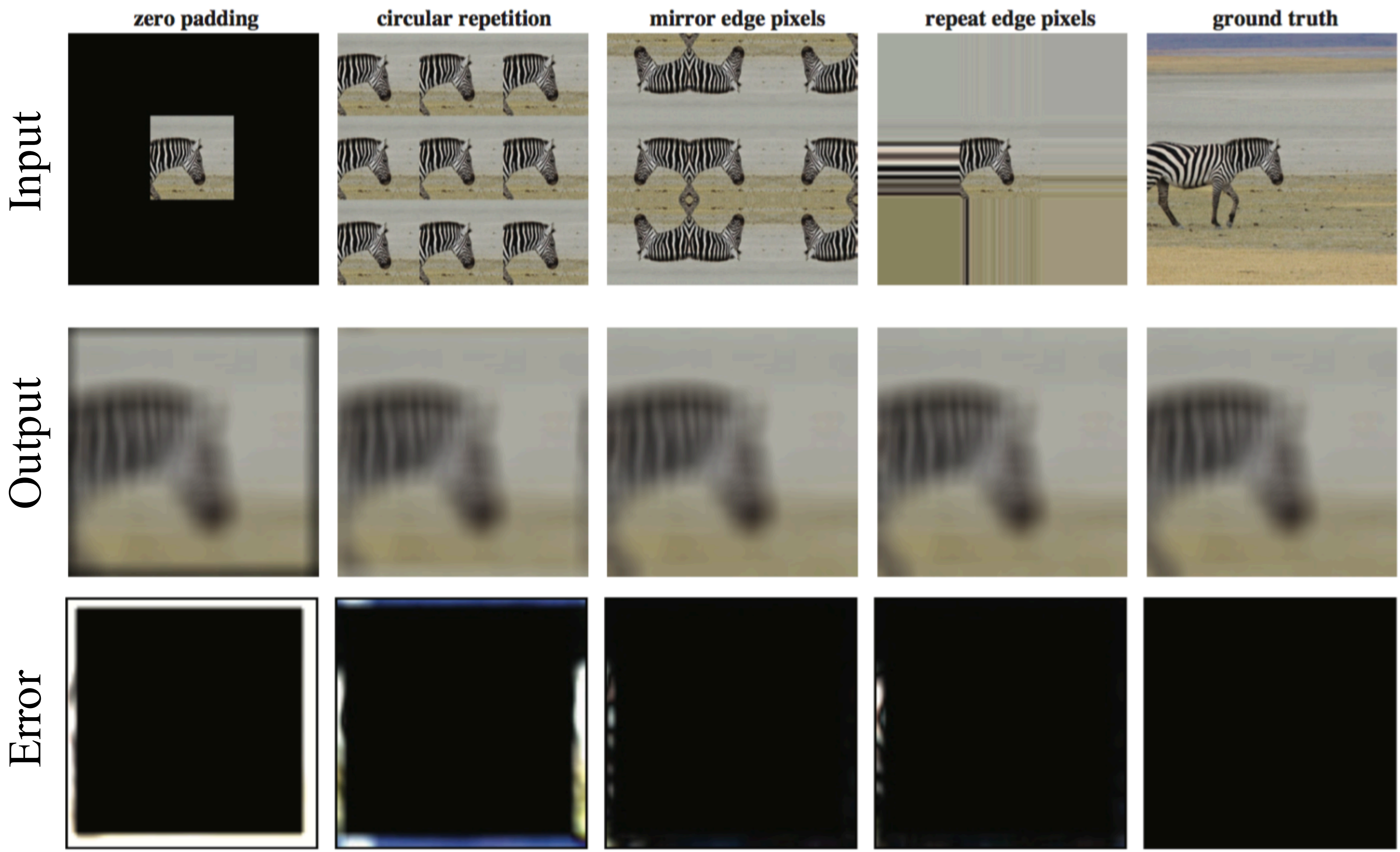
f[m,n]

Handling boundaries

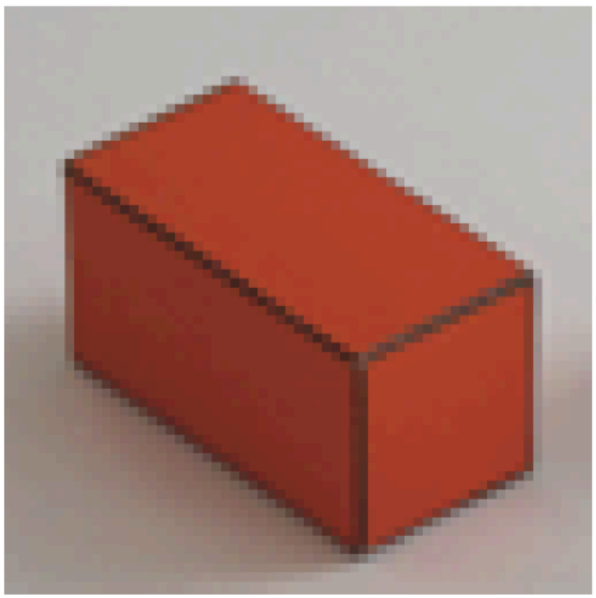
Handling boundaries



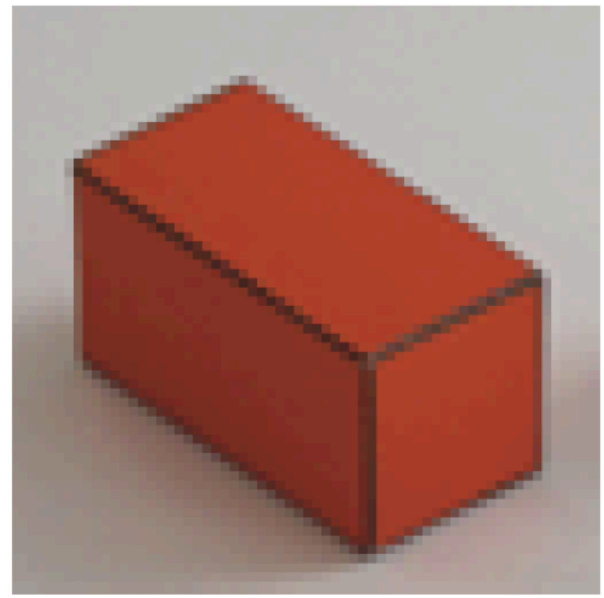
Handling boundaries



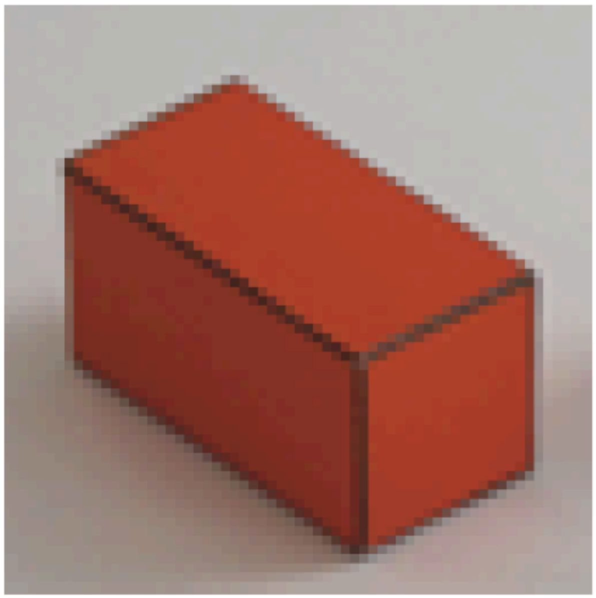
Examples



$$\circ \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} =$$

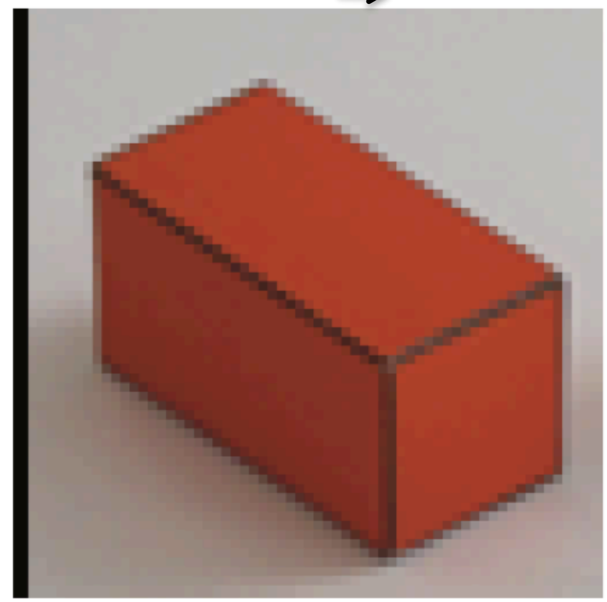


Examples



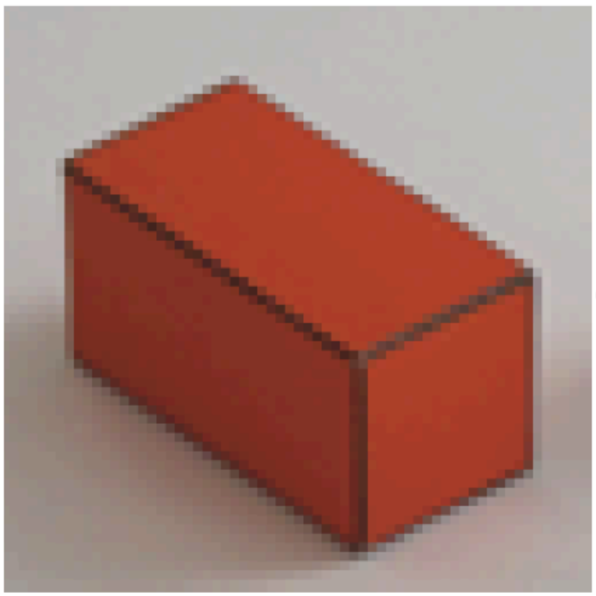
$$\circ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} =$$

2 pixels



(using zero padding)

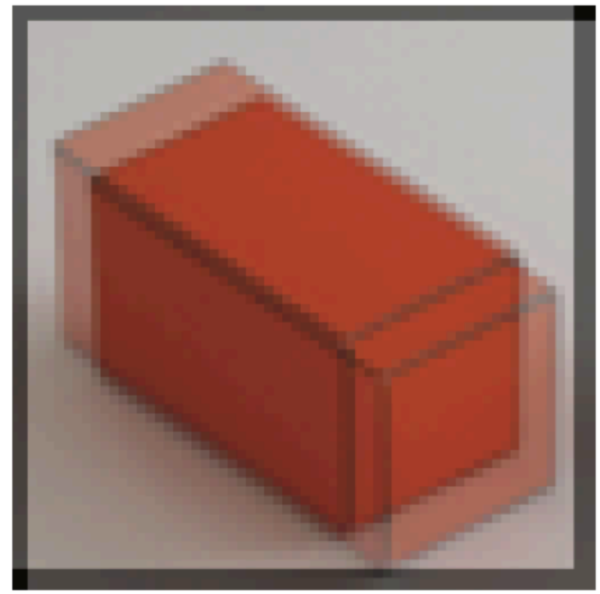
Examples



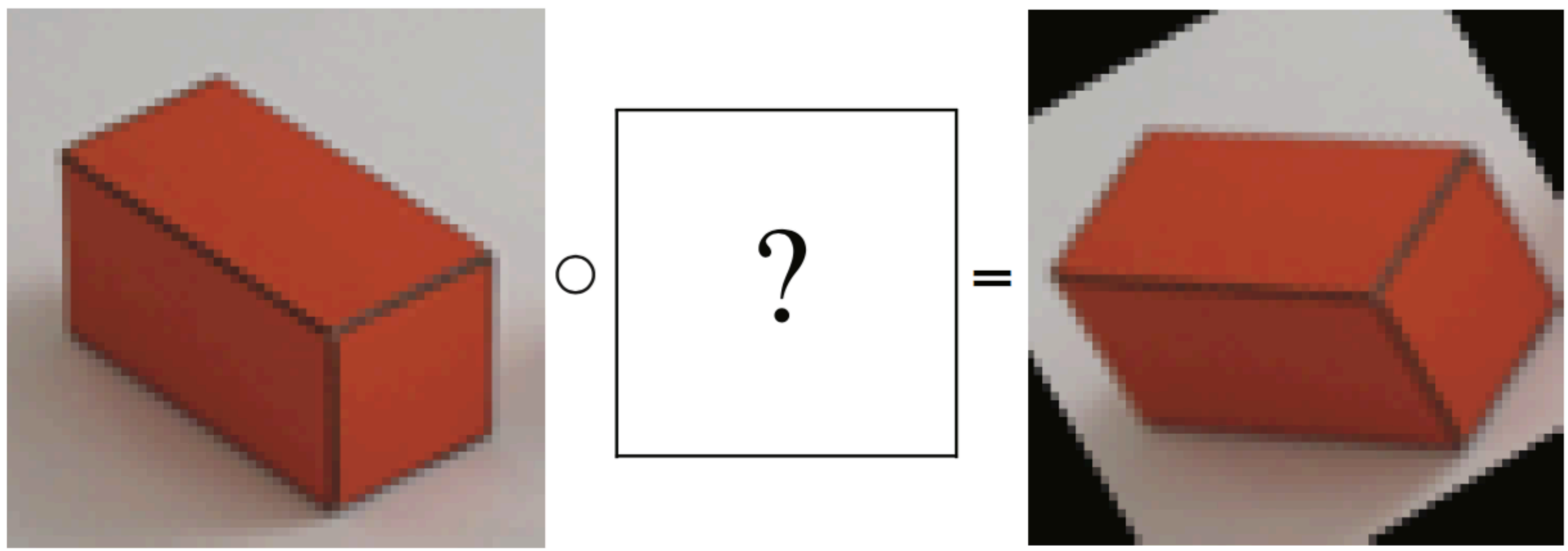
○

.5	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	.5

=



Examples



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

g[m,n]

$\otimes$

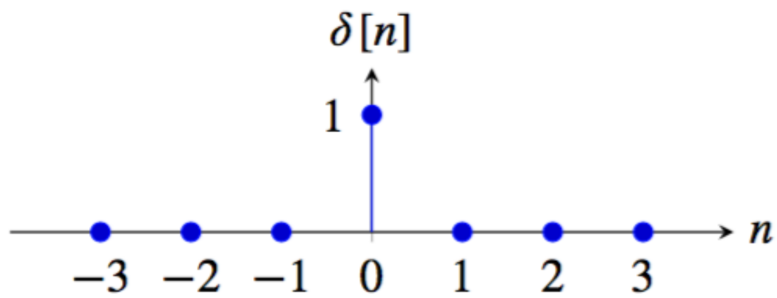
h[m,n]

=

f[m,n]

Important signals

The impulse



The result of convolving a signal g[n] with the impulse signal is the same signal:

$$f[n] = \delta \circ g = \sum_k \delta[n-k]\, g[k] = g[n]$$

Convolving a signal f with a translated impulse $\delta\,[n - n_0]$ results in a translated signal:

$$f[n - n_0] = \delta[n - n_0] \circ f[n]$$

Important signals

Cosine and sine waves

$$s(t) = A\sin(w\,t - \theta)$$

A discrete signal $f[n]$ is periodic, if there exists $T \in \mathrm{N}$ such that $f[n] = f[n + mT]$ for all $m \in \mathrm{Z}$. For the discrete sine (and cosine) wave to be periodic the frequency has to be $w = 2\pi K/N$ for $K,N \in \mathrm{N}$. If K/N is an irreducible fraction, then the period of the wave will be $T = N$ samples.

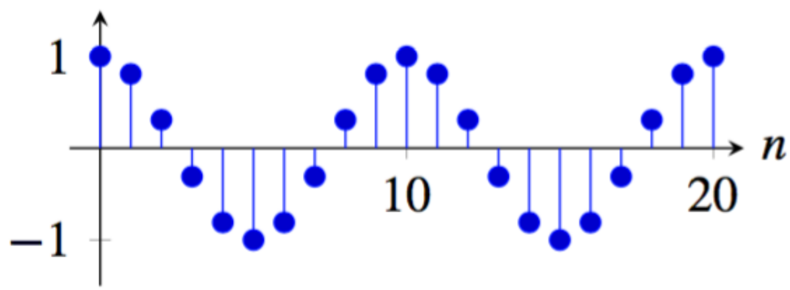
$$s_k[n] = \sin\left(\frac{2\pi}{N}k\,n\right) \qquad c_k[n] = \cos\left(\frac{2\pi}{N}k\,n\right)$$

$k \in [1, N/2]$ denotes the number of wave cycles that will occur within the region of support

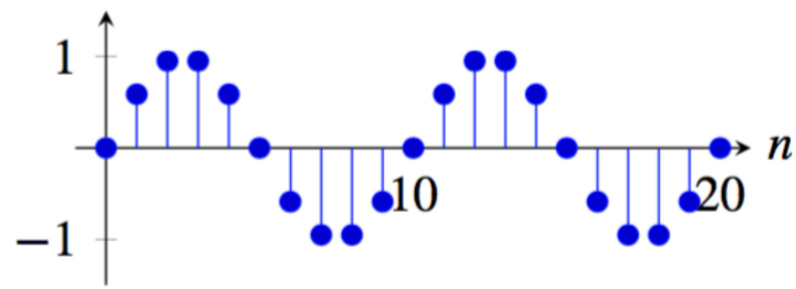
Important signals

Cosine and sine waves

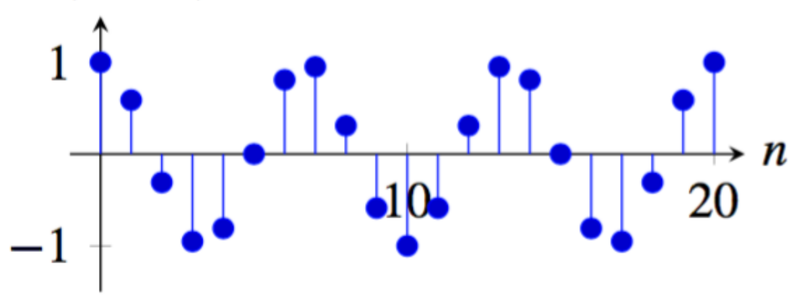
$\cos\left(\frac{2\pi}{N}kn\right)$ k=2



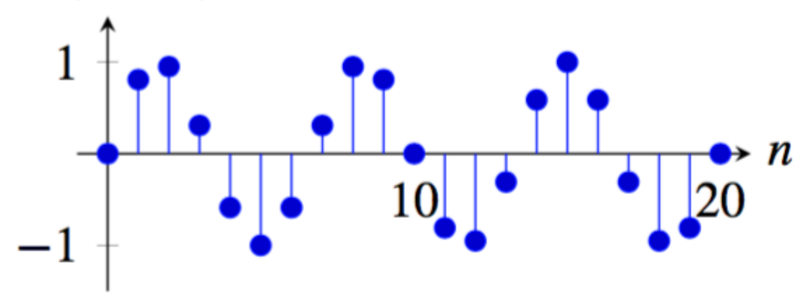
$\sin\left(\frac{2\pi}{N}kn\right)$ k=2



$\cos\left(\frac{2\pi}{N}kn\right)$ k=3

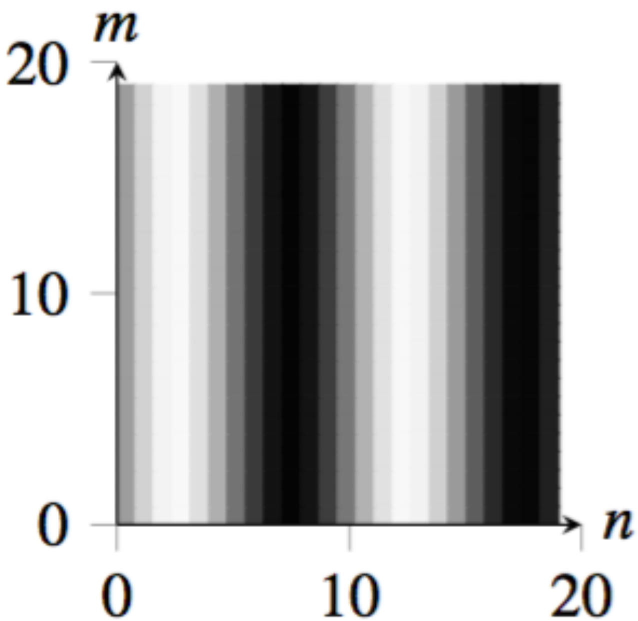


$\sin\left(\frac{2\pi}{N}kn\right)$ k=3

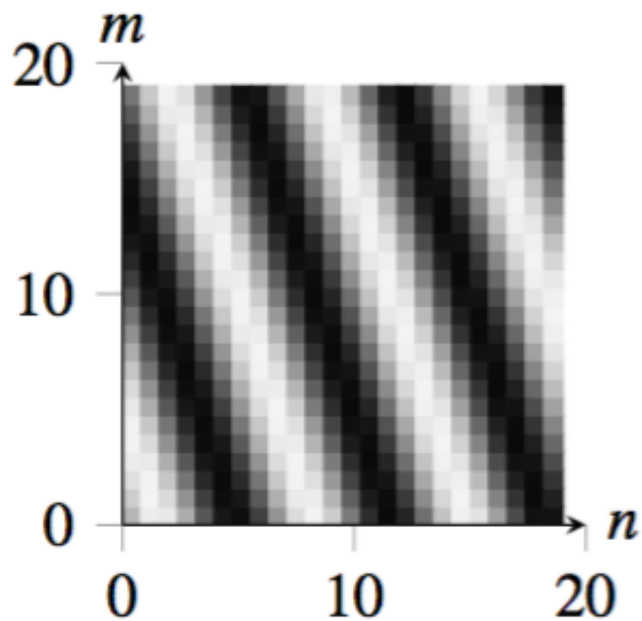


Waves in 2D

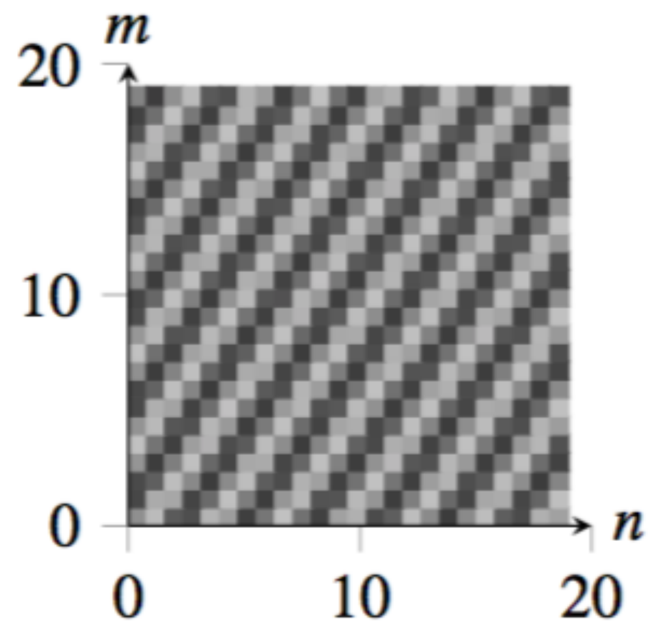
$$s_{u,v}[n,m] = A\sin\left(2\pi\left(\frac{u\,n}{N} + \frac{v\,m}{M}\right)\right) \qquad c_{u,v}[n,m] = A\cos\left(2\pi\left(\frac{u\,n}{N} + \frac{v\,m}{M}\right)\right)$$



u = 2, v = 0



u = 3, v = 1



u = 7, v = −5

Important signals

Complex exponential

$$s(t) = A\exp(jwt)$$

In discrete time (setting $A = 1$), we can write:

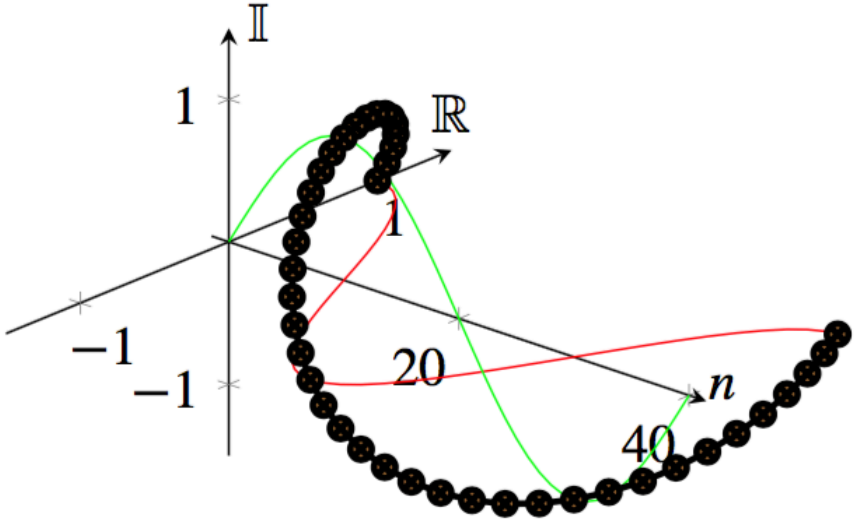
$$e_k[n] = \exp\left(j\frac{2\pi}{N}kn\right) = \cos\left(\frac{2\pi}{N}kn\right) + j\sin\left(\frac{2\pi}{N}kn\right)$$

And in 2D, the complex exponential wave is:

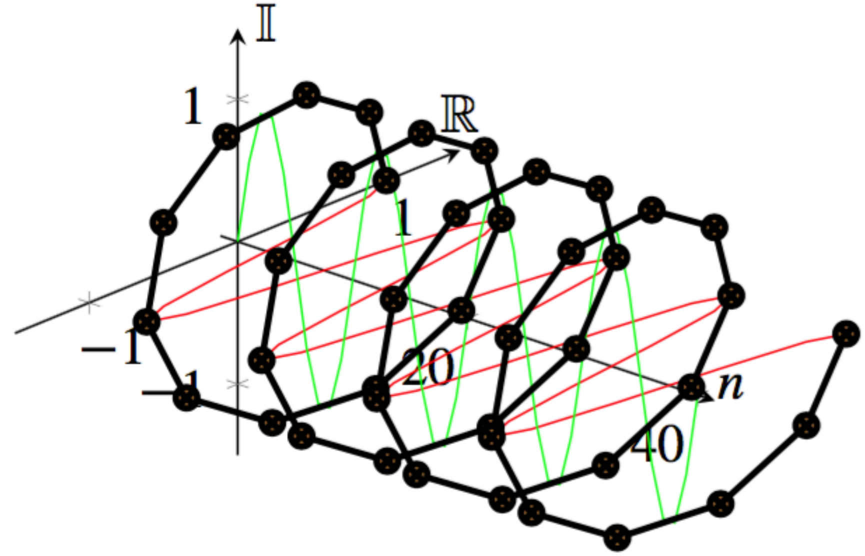
$$e_{u,v}[n,m] = \exp\left(2\pi j\left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

Important signals

Complex exponential



$N = 40, k = 1$



$N = 40, k = 3$

Impulses, sine and cosine waves or complex exponentials form each an orthogonal basis for signals of length N

Linear image transformations

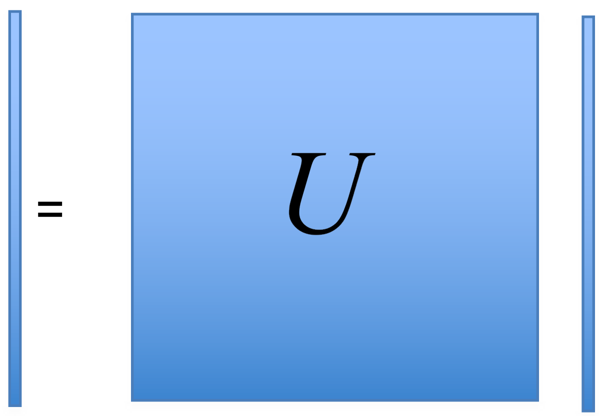
- In analyzing images, it's often useful to make a change of basis.

Transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



Self-inverting transforms

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

Same basis functions are used for the inverse transform

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

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

U transpose and complex conjugate

The Discrete Fourier transform

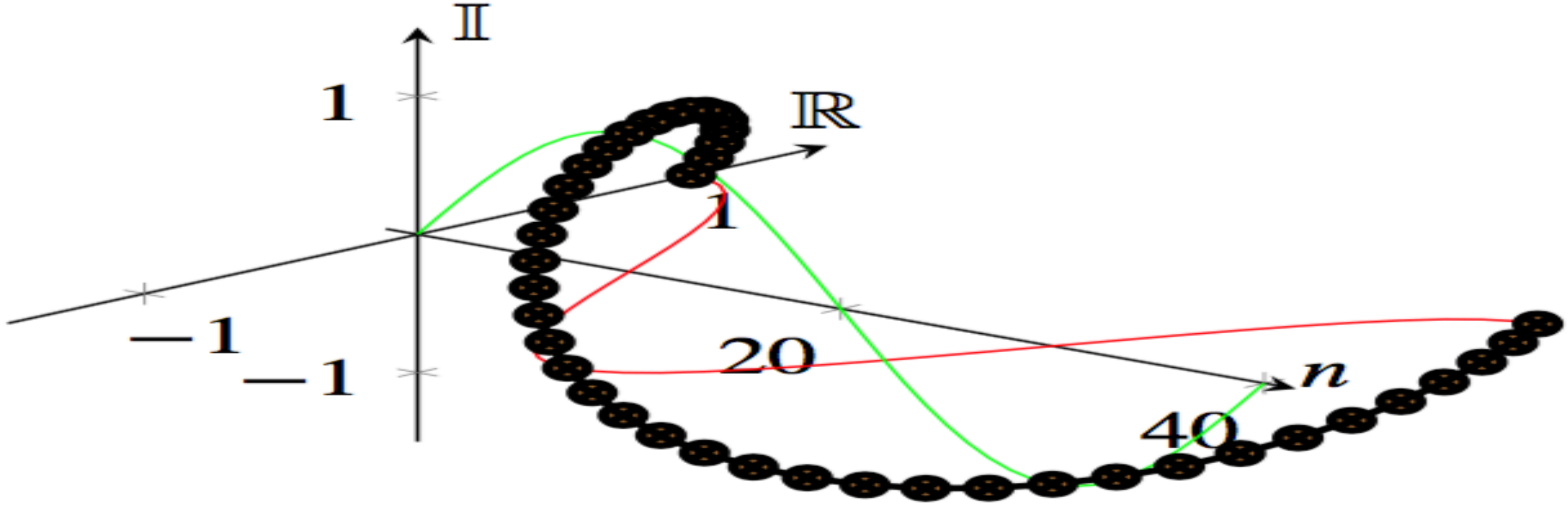
Discrete Fourier Transform (DFT) transforms an image $f\,[m, m]$ into the complex image Fourier transform $F\,[u, v]$ as:

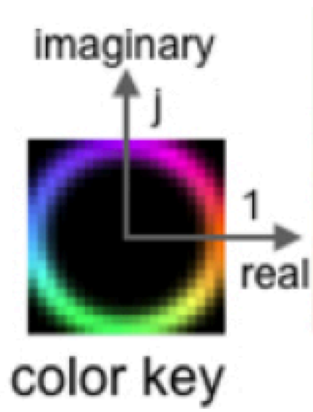
$$F[u,v] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

The inverse Fourier transform is:

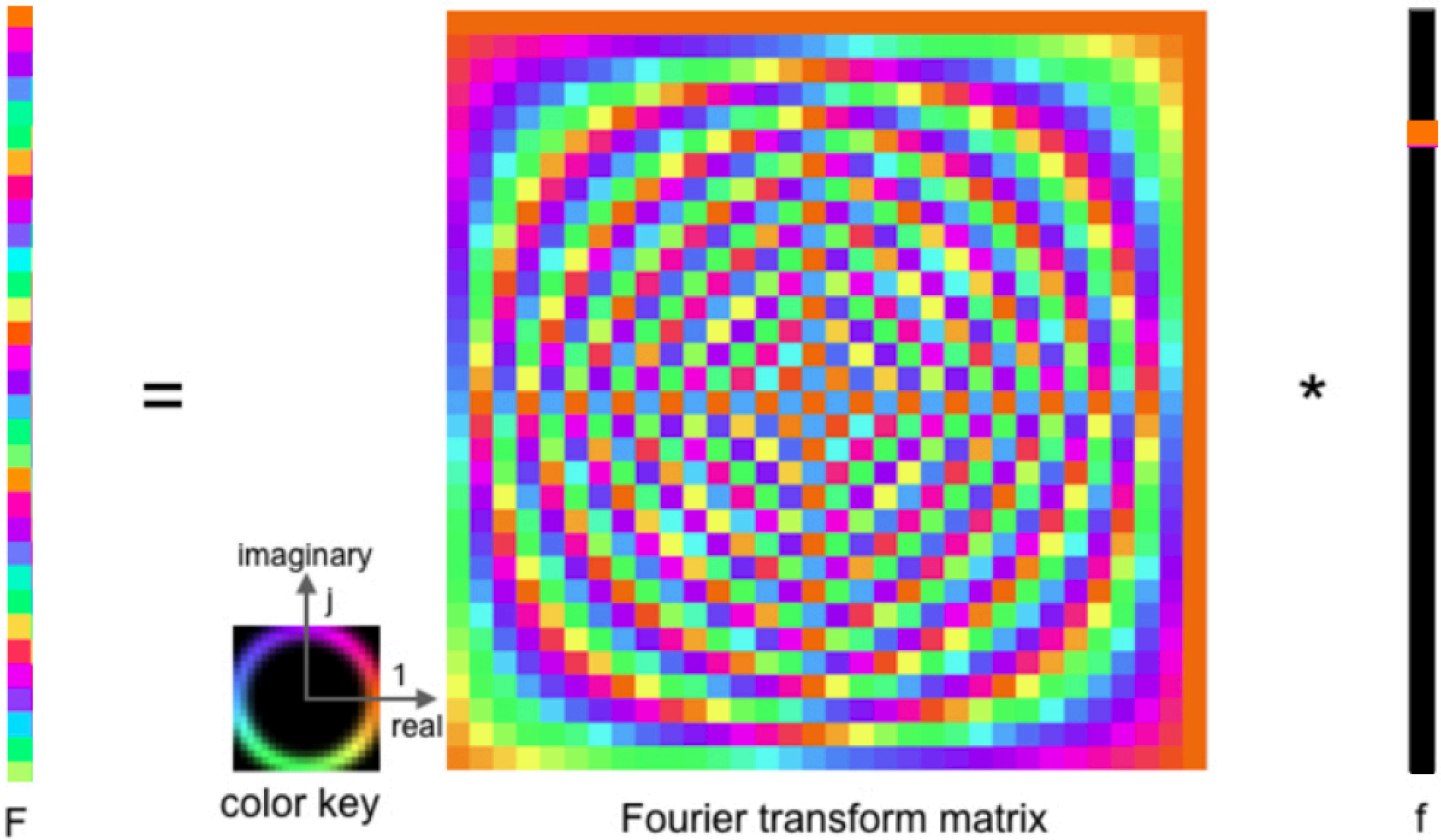
$$f[n,m] = \frac{1}{NM}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F[u,v] \exp\left(+2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

Discrete Fourier transform visualization





Fourier transform visualization



Some useful transforms

Fourier transform of the Delta function δ[n ,m]**:**

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \delta[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right) = 1$$

Observation: if we think in terms of the inverse DFT, this means that:

$$\delta[n,m] = \frac{1}{NM} \sum_{u=-N/2}^{N/2} \sum_{v=-M/2}^{M/2} \exp\left(2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

Some useful transforms

The Fourier transform of the cosine wave

$$\cos\left(2\pi\left(\frac{u_0\, n}{N} + \frac{v_0\, m}{M}\right)\right)$$

is:

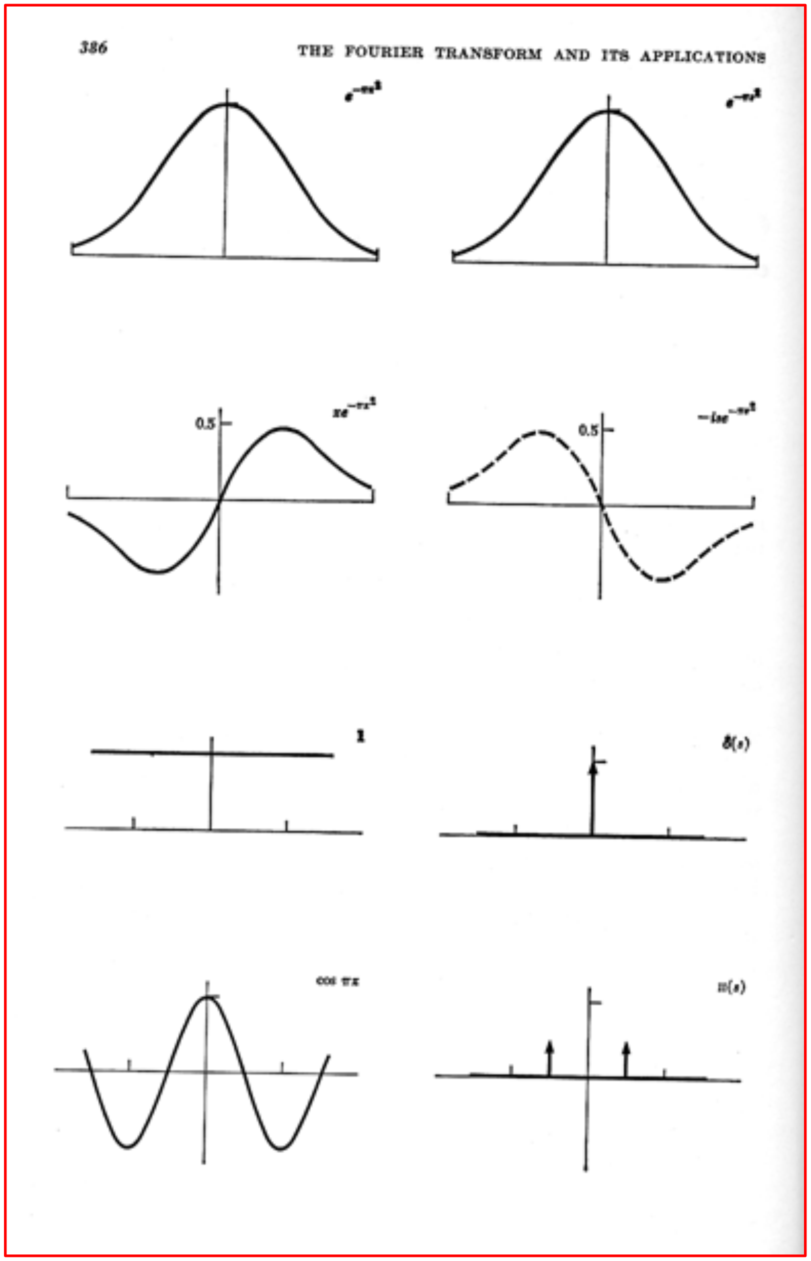
$$F[u,v] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} \cos\left(2\pi\left(\frac{u_0\, n}{N} + \frac{v_0\, m}{M}\right)\right) \exp\left(-2\pi j\left(\frac{u\, n}{N} + \frac{v\, m}{M}\right)\right) =$$

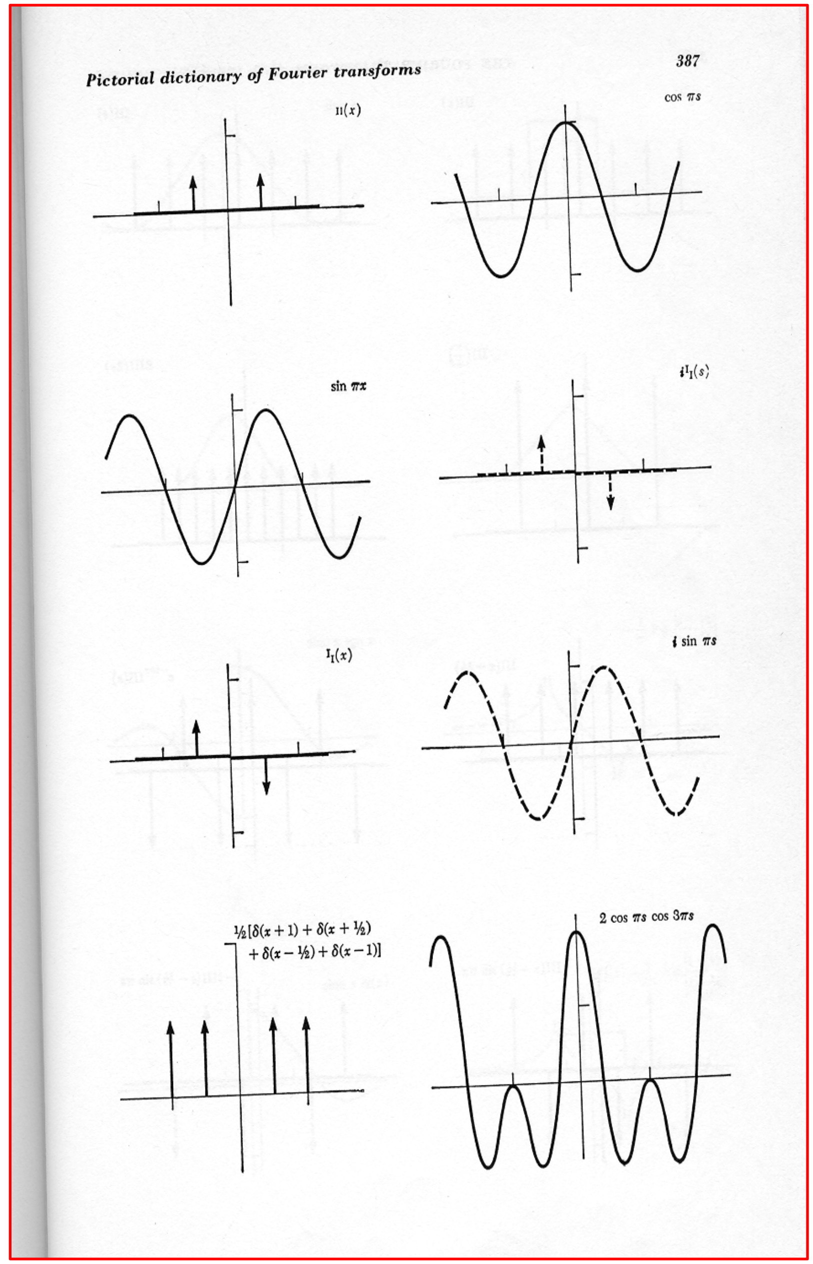
$$= \frac{1}{2}\left(\delta[u-u_0, v-v_0] + \delta[u+u_0, v+v_0]\right)$$

Same for the sine wave:

$$\sin\left(2\pi\left(\frac{u_0\, n}{N} + \frac{v_0 m}{M}\right)\right) \leftrightarrow F[u,v] = \frac{1}{2j}\left(\delta[u-u_0, v-v_0] - \delta[u+u_0, v+v_0]\right)$$

Bracewell's pictorial dictionary of Fourier transform pairs





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

Szeliski, Computer Vision, 2010

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

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

2D Discrete Fourier Transform

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

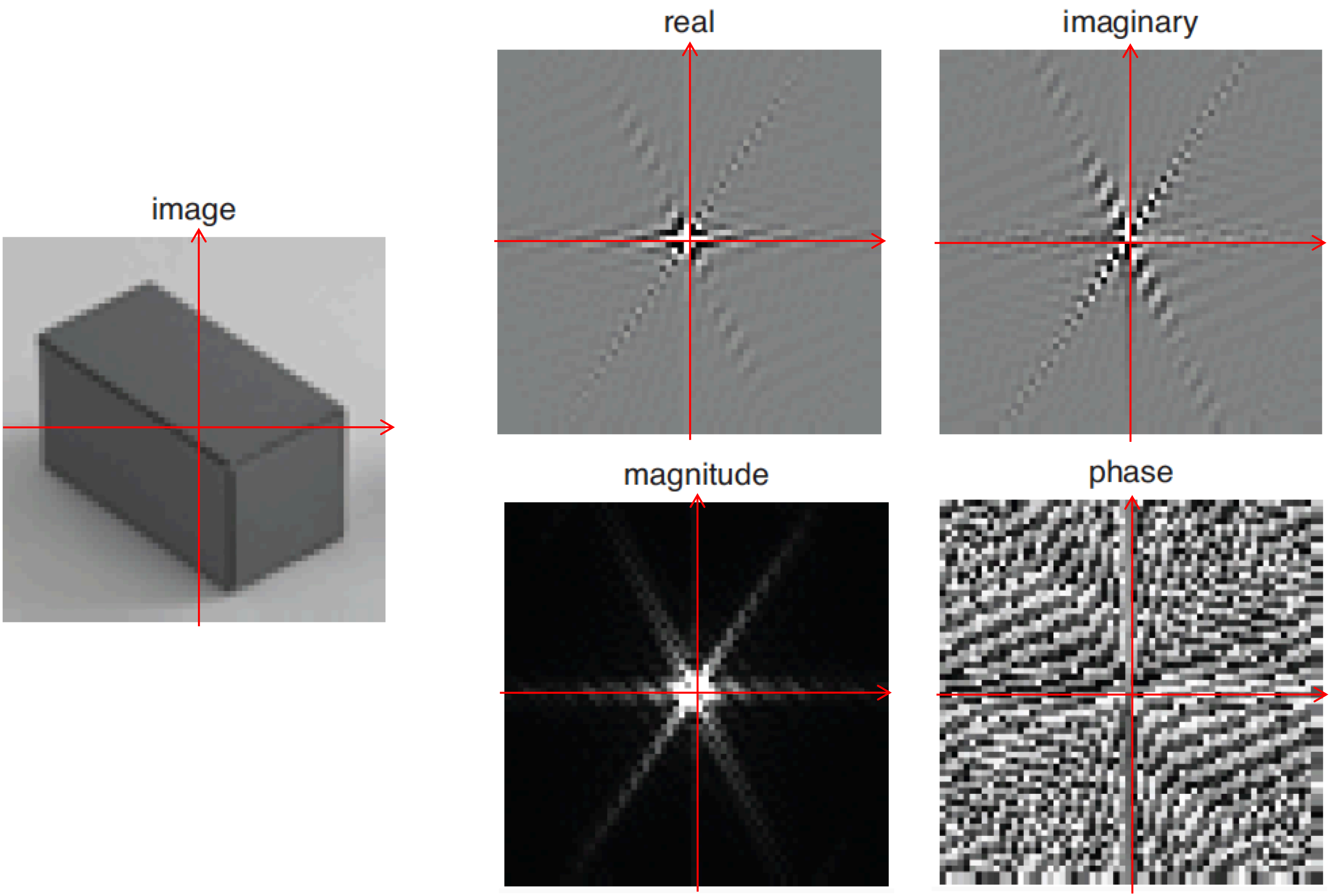
Using the real and imaginary components:

$$F[u,v] = Re\{F[u,v]\} + j\,Imag\{F[u,v]\}$$

Or using a polar decomposition:

$$F[u,v] = A[u,v] \exp(j\theta[u,v])$$

2D Discrete Fourier Transform



Properties for the DFT

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

- Linearity
- Symmetry: Fourier transform of a real signal has coefficients that come in pairs, with *F* [*u*, *v*] being the complex conjugate of *F* [-*u*, -v].

Properties for the DFT

- Shift in space

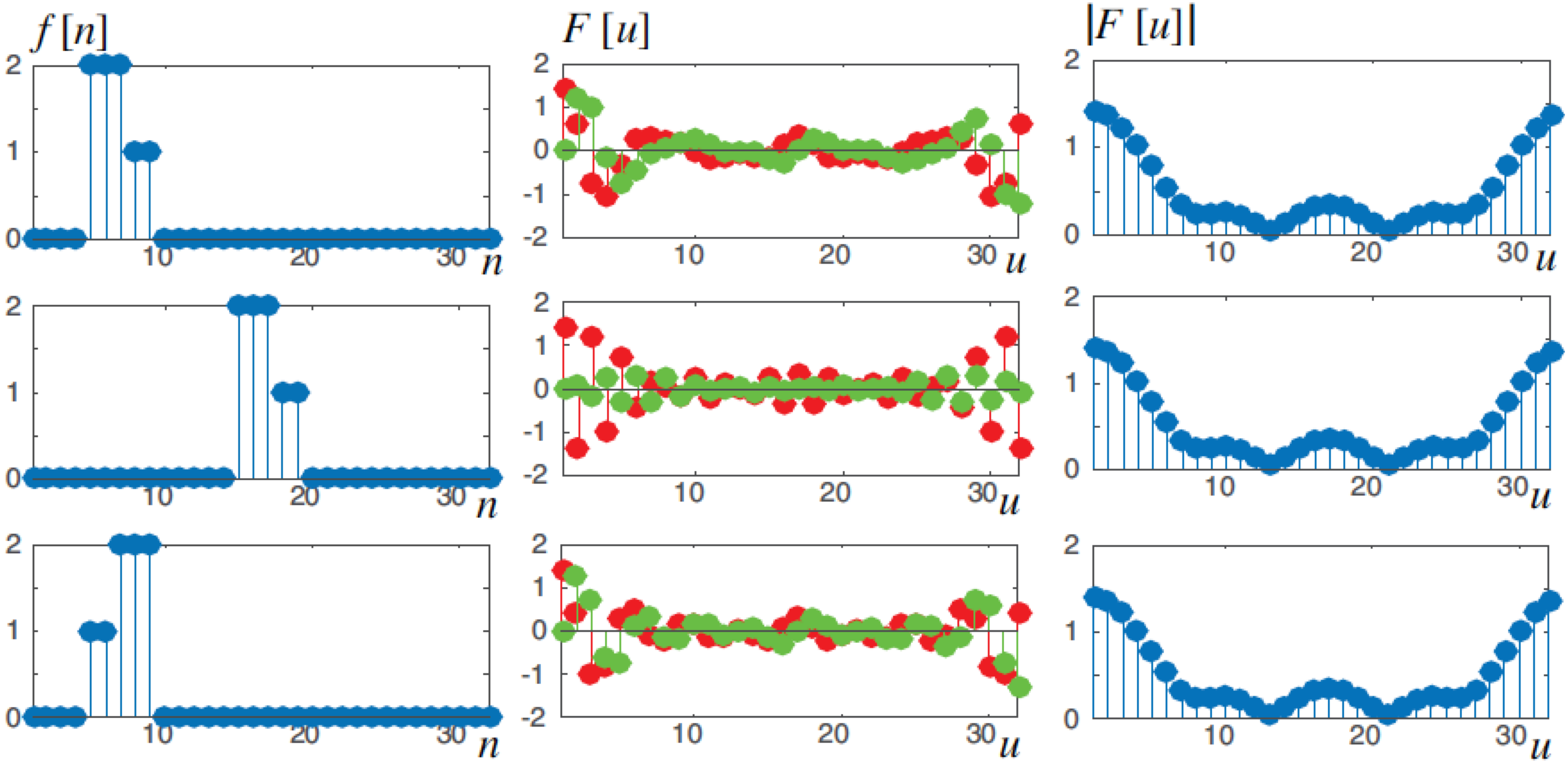
$$DFT\{f[n-n_0, m-m_0]\} =$$

$$= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f[n-n_0, m-m_0] \exp\left(-2\pi j\left(\frac{un}{N}+\frac{vm}{M}\right)\right) =$$

$$= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f[n, m] \exp\left(-2\pi j\left(\frac{u(n+n_0)}{N}+\frac{v(m+m_0)}{M}\right)\right) =$$

$$= \boxed{F[u,v] \exp\left(-2\pi j\left(\frac{u n_0}{N}+\frac{v m_0}{M}\right)\right)}$$

Properties for the DFT



Only the phase changes! The magnitude is translation invariant.

Properties for the DFT

- Modulation

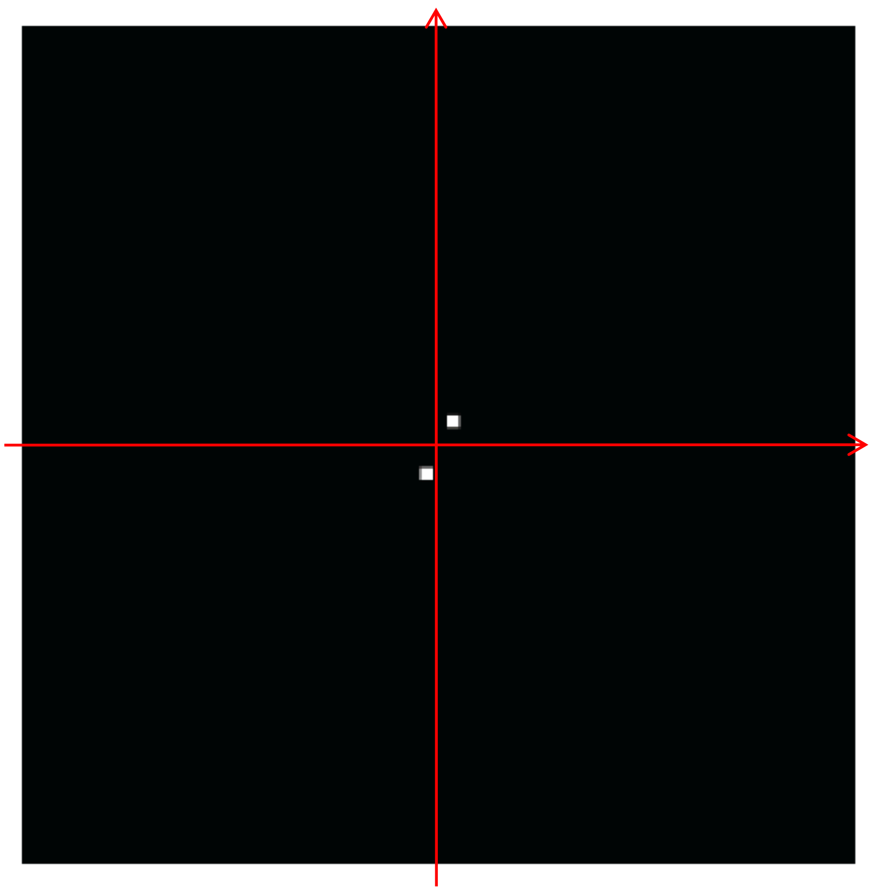
$$f[n,m]\cos\left(2\pi j\left(\frac{u_0 n}{N}+\frac{v_0 m}{M}\right)\right)$$

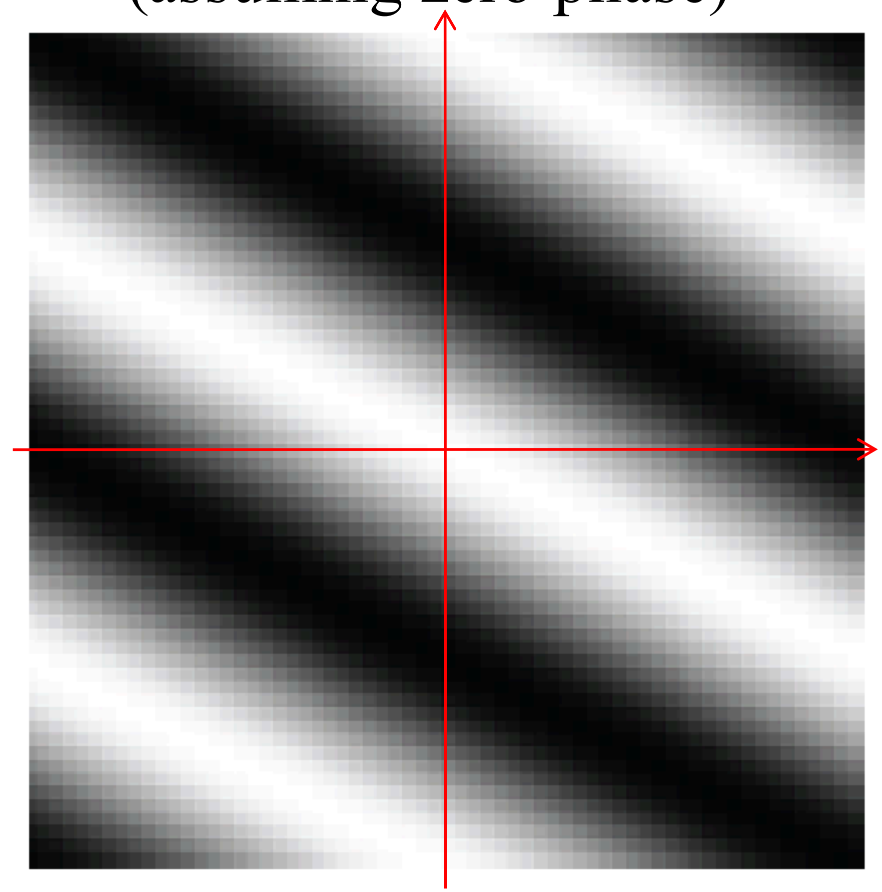
$$f[n,m]\exp\left(-2\pi j\left(\frac{u_0 n}{N}+\frac{v_0 m}{M}\right)\right)$$

Multiplying by a complex exponential results in a translation of the DFT

$$DFT\left\{f[n,m]\exp\left(-2\pi j\left(\frac{u_0 n}{N}+\frac{v_0 m}{M}\right)\right)\right\}=F[u-u_0,v-v_0]$$

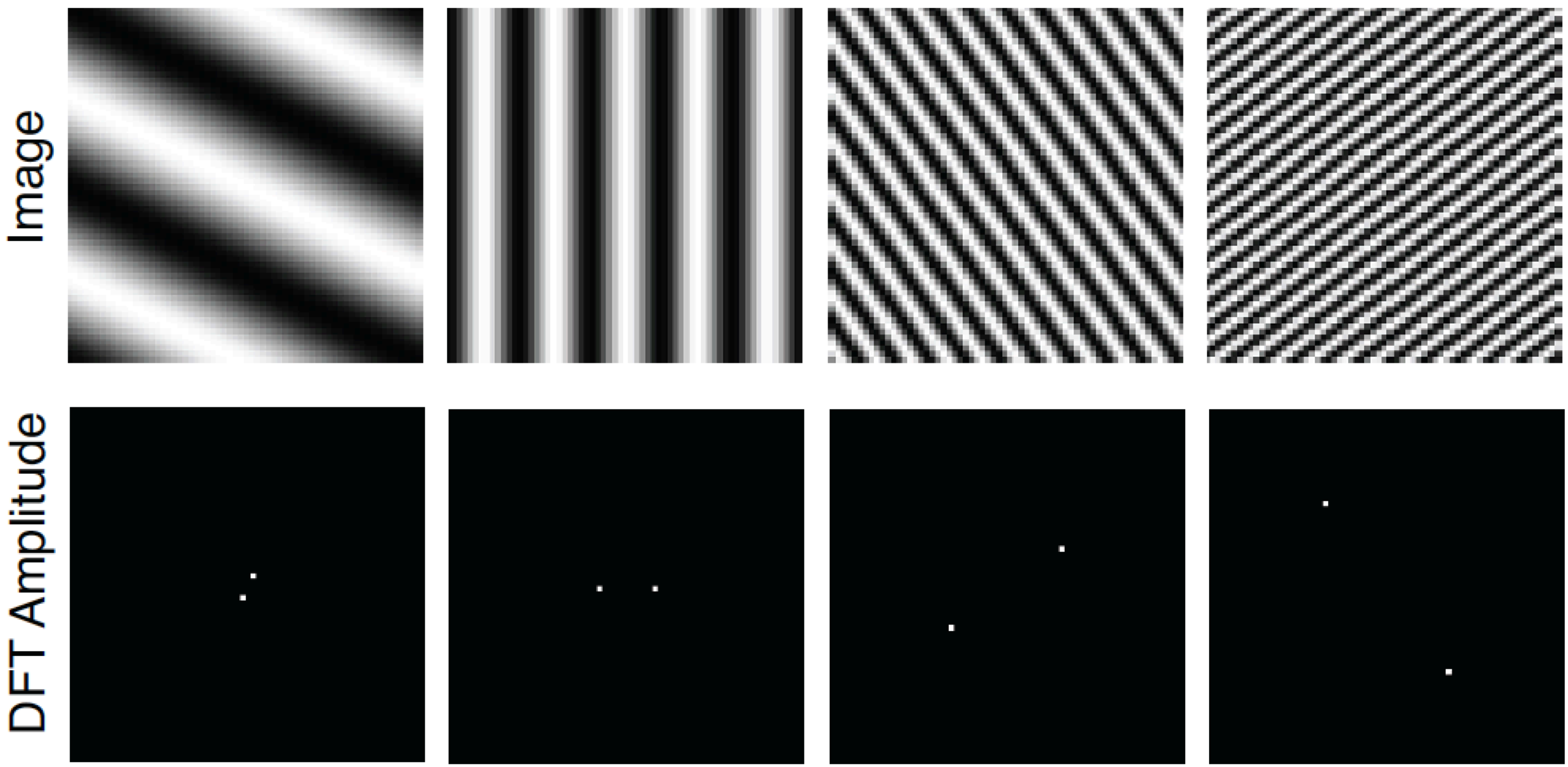
Frequencies



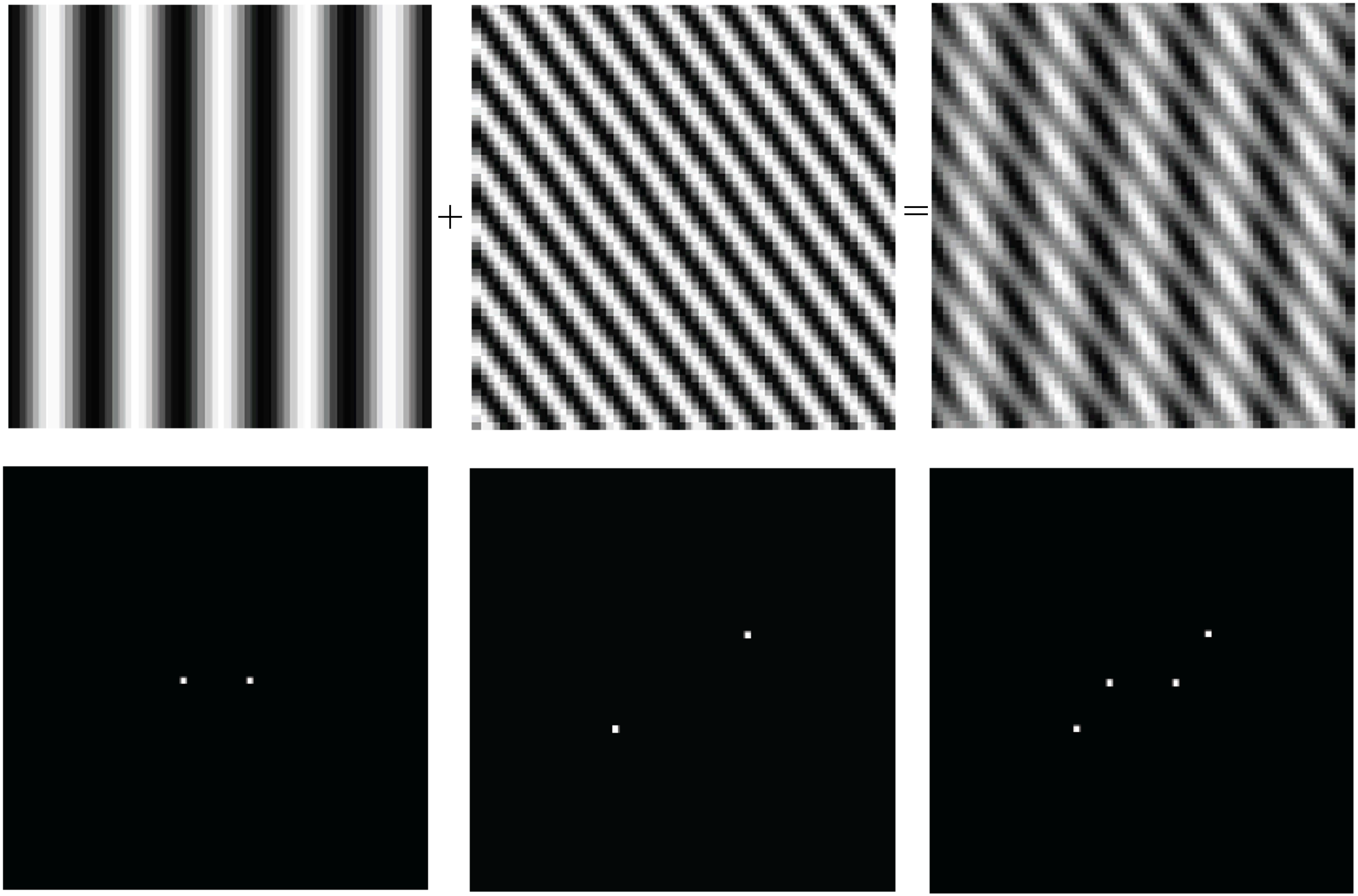


Images are 64x64 pixels. The wave is a cosine (if phase is zero).

Frequencies

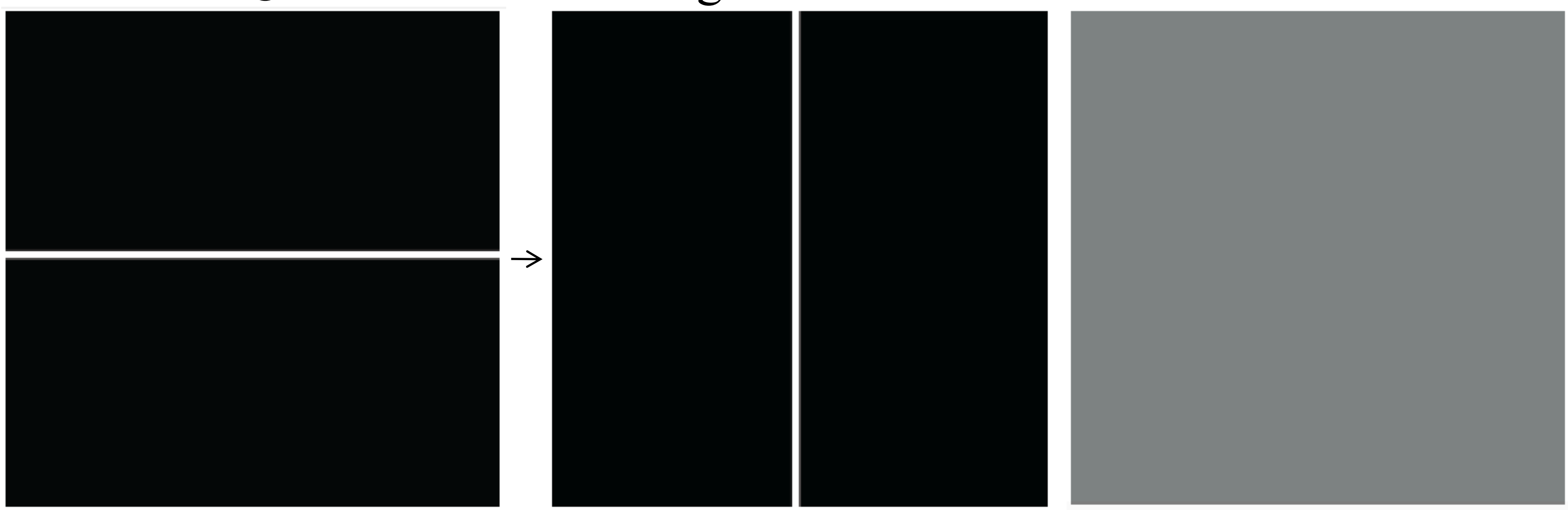


Images are 64x64 pixels. The wave is a cosine (if phase is zero).



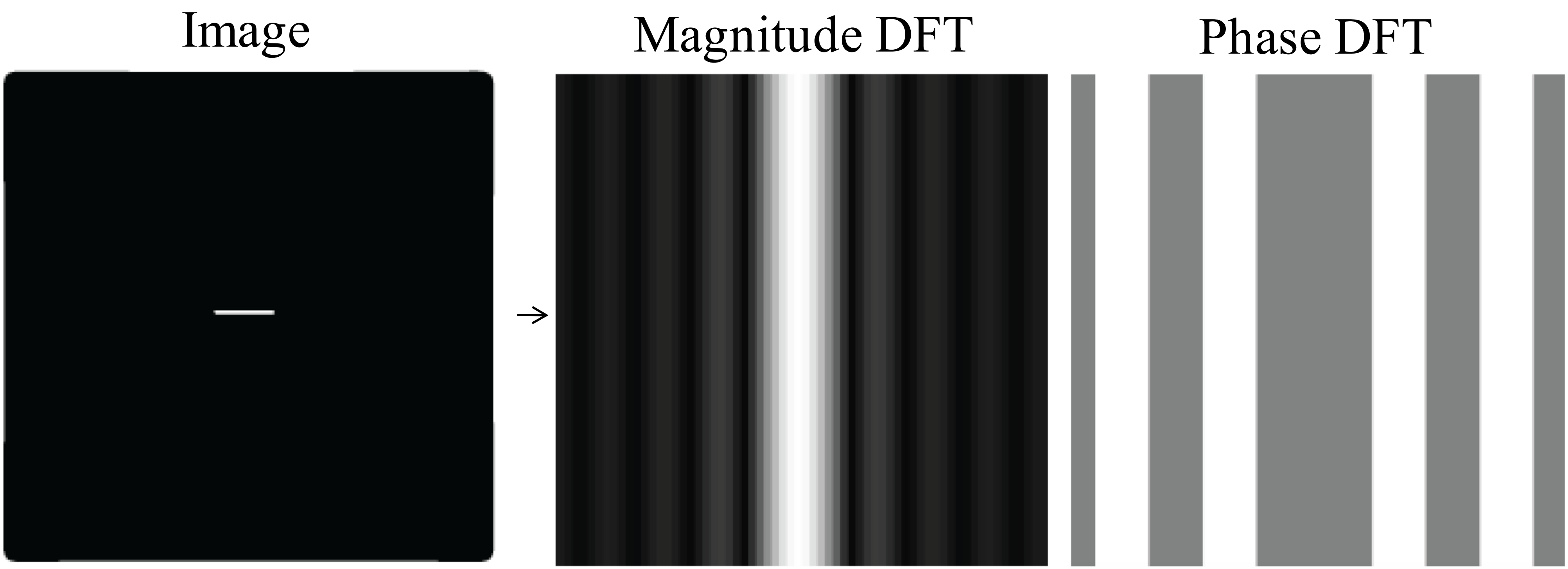
Some important Fourier transforms

Image Magnitude DFT Phase DFT

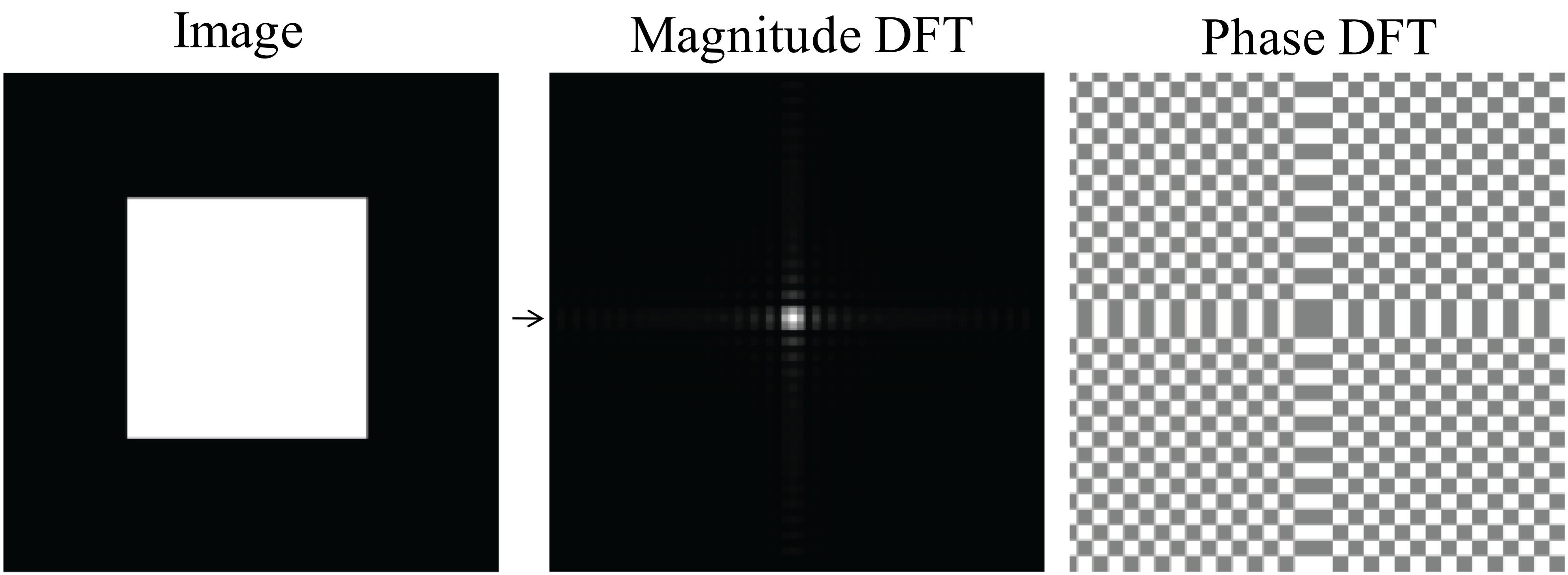


Images are 64x64 pixels.

Some important Fourier transforms

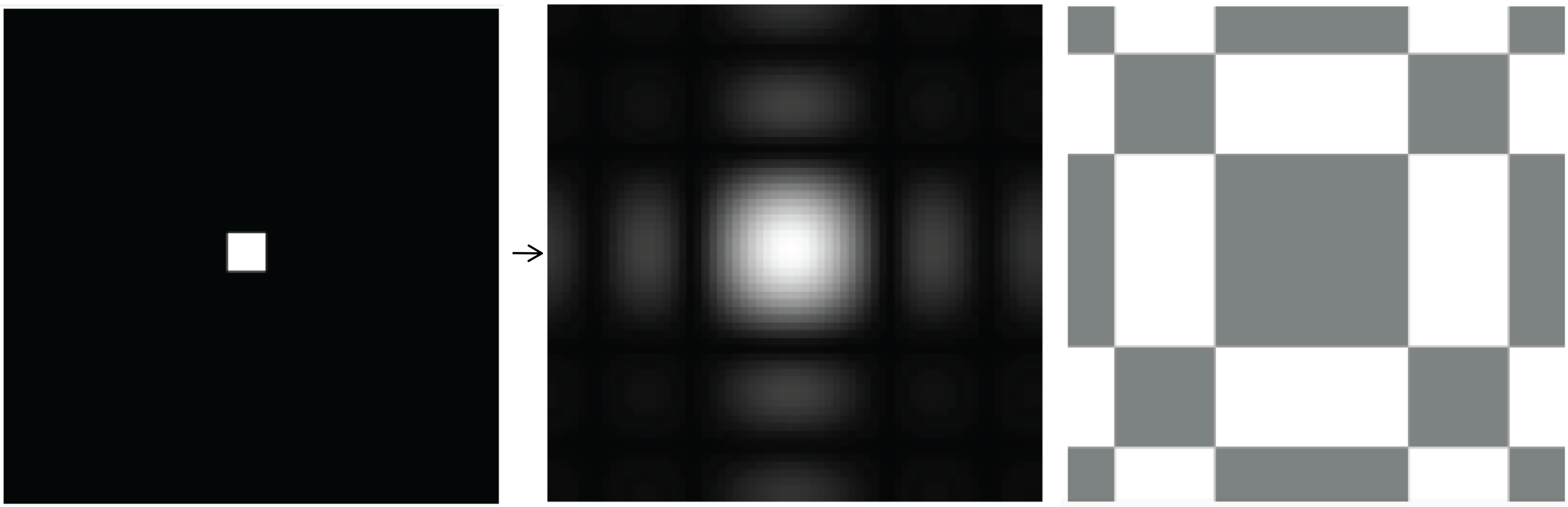


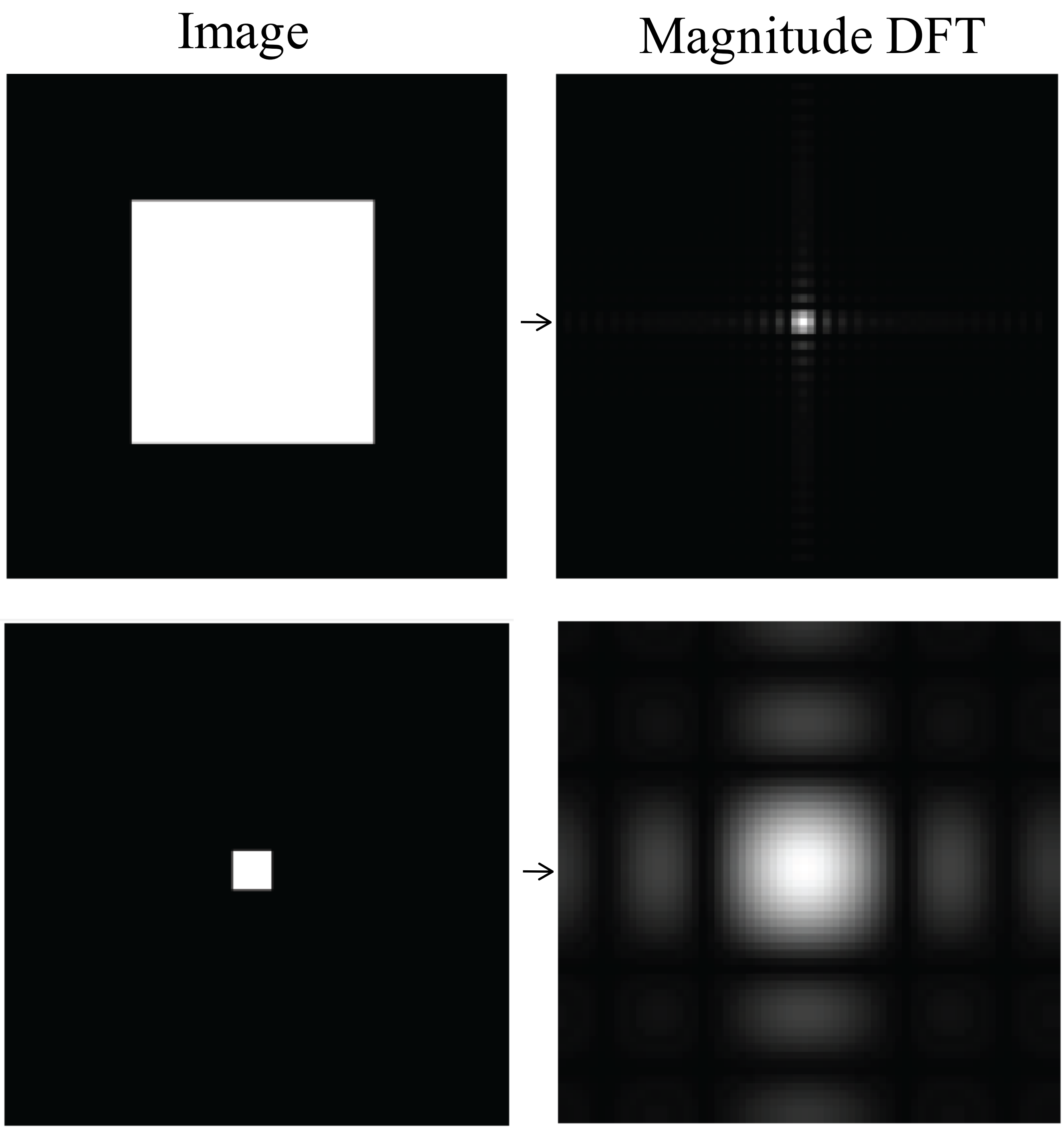
Some important Fourier transforms



Some important Fourier transforms

Image | Magnitude DFT | Phase DFT

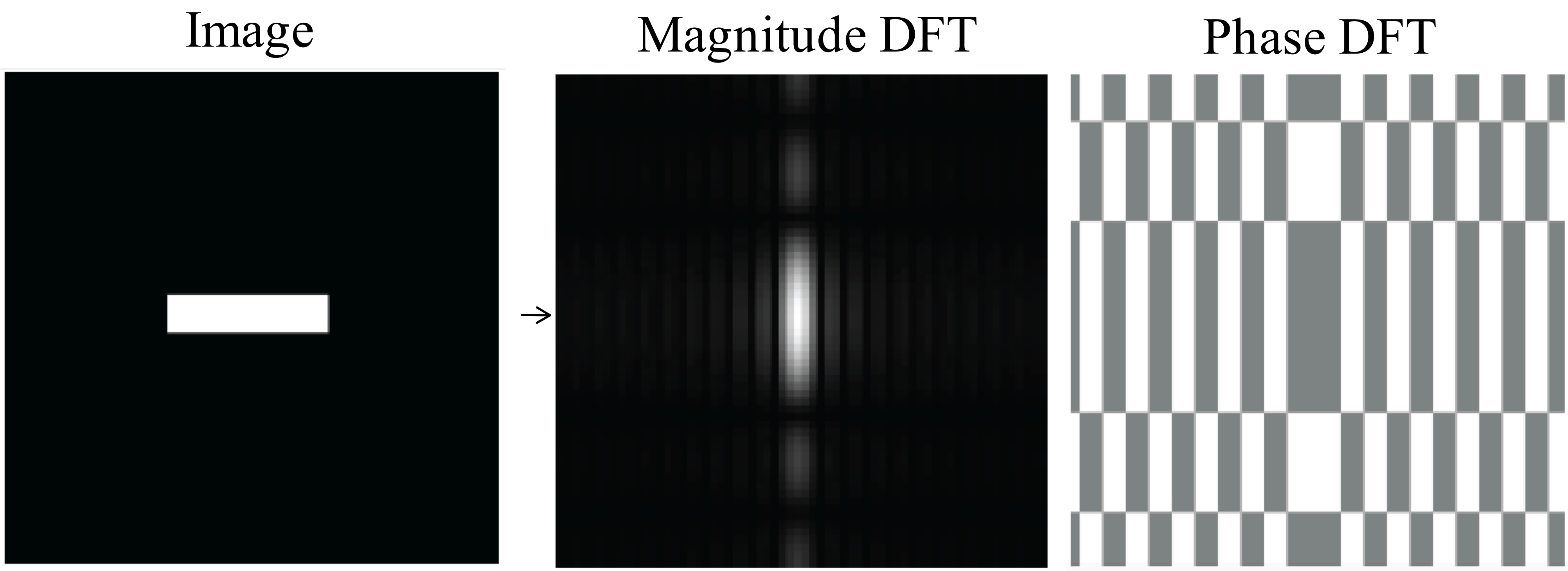




Scale

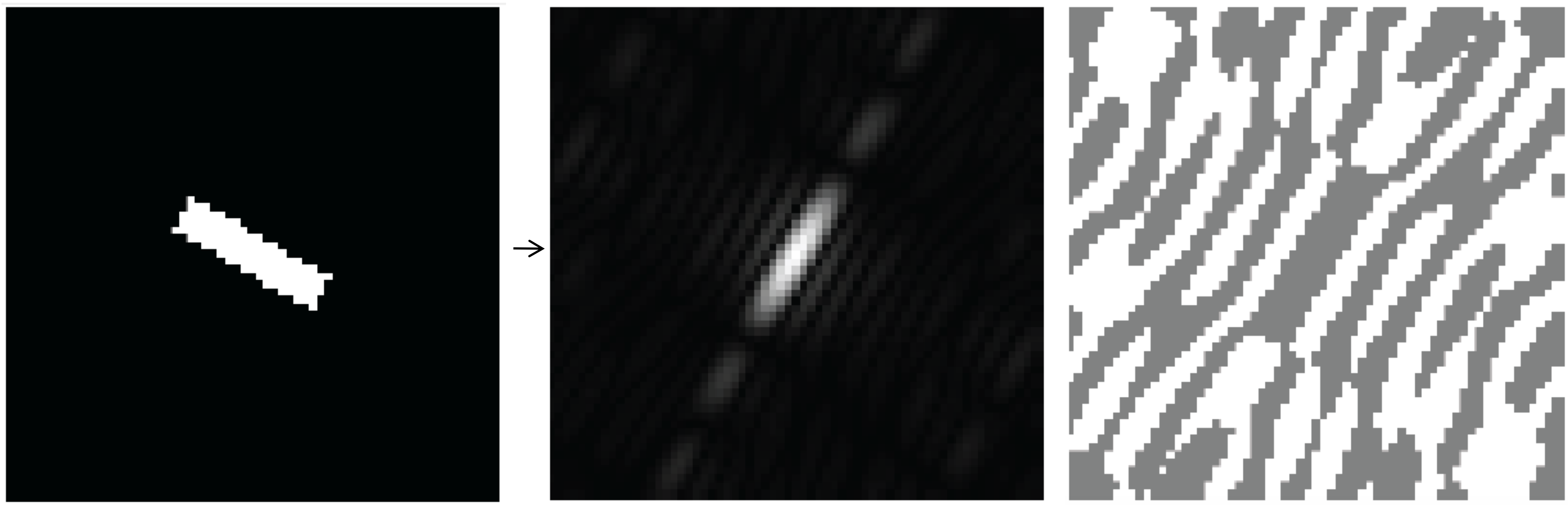
Small image details produce content in high spatial frequencies

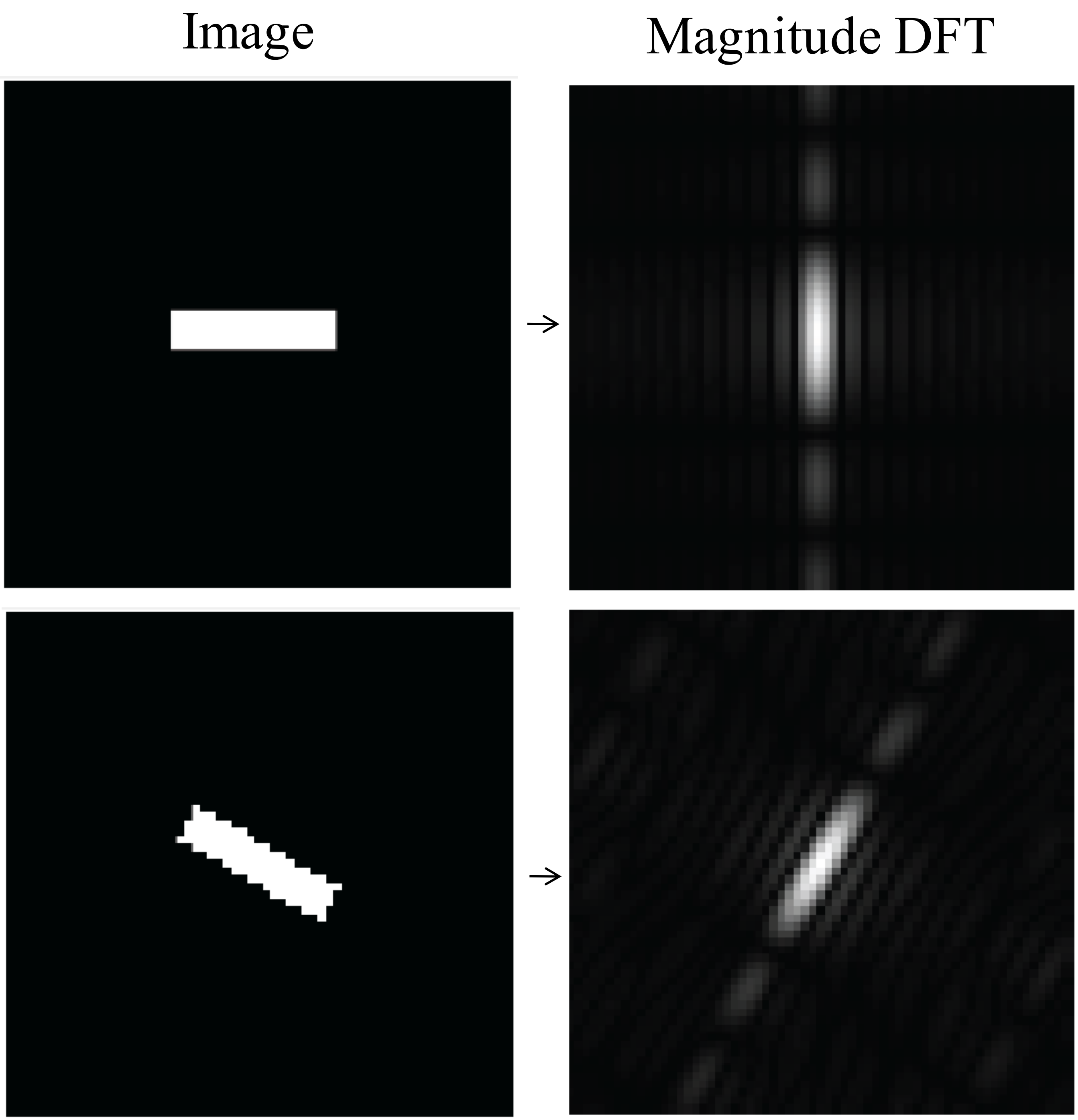
Some important Fourier transforms



Some important Fourier transforms

Image | Magnitude DFT | Phase DFT

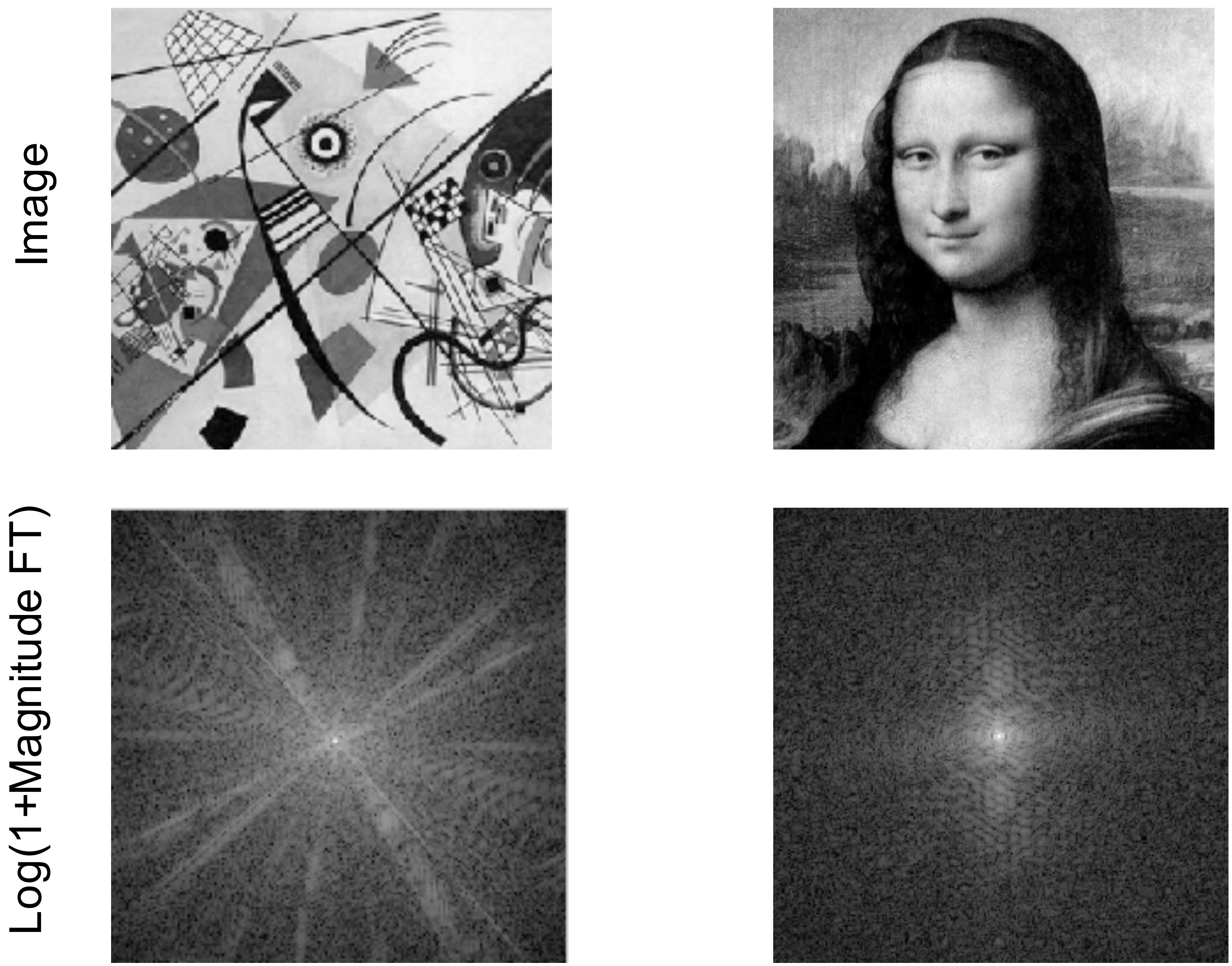




Orientation

A line transforms to a line oriented perpendicularly to the first.

The Fourier Transform of some important images



More properties for the DFT

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{u\,n}{N} + \frac{v\,m}{M}\right)\right)$$

DFT of the convolution

$$f = g \circ h \longleftrightarrow F[u,v] = G[u,v]\,H[u,v]$$

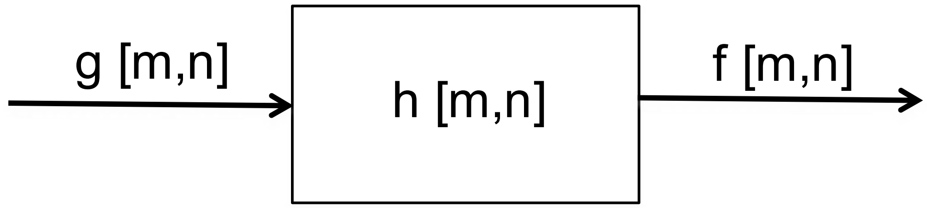
$$F[u,v] = DFT\{g \circ h\}$$

$$= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \boxed{\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} g[m-k, n-l]\,h[k,l]} \exp\left(-2\pi j\left(\frac{mu}{M}+\frac{nv}{N}\right)\right)$$

$$F[u,v] = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} h[k,l] \sum_{m'=-k}^{M-k-1}\sum_{n'=-l}^{N-l-1} g[m',n'] \exp\left(-2\pi j\left(\frac{(m'+k)u}{M}+\frac{(n'+l)v}{N}\right)\right)$$

$$F[u,v] = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} G[u,v] \exp\left(-2\pi j\left(\frac{ku}{M}+\frac{lv}{N}\right)\right) h[k,l]$$

Linear filtering



In the spatial domain:

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

In the frequency domain:

$$F[u,v] = G[u,v]\, H[u,v]$$

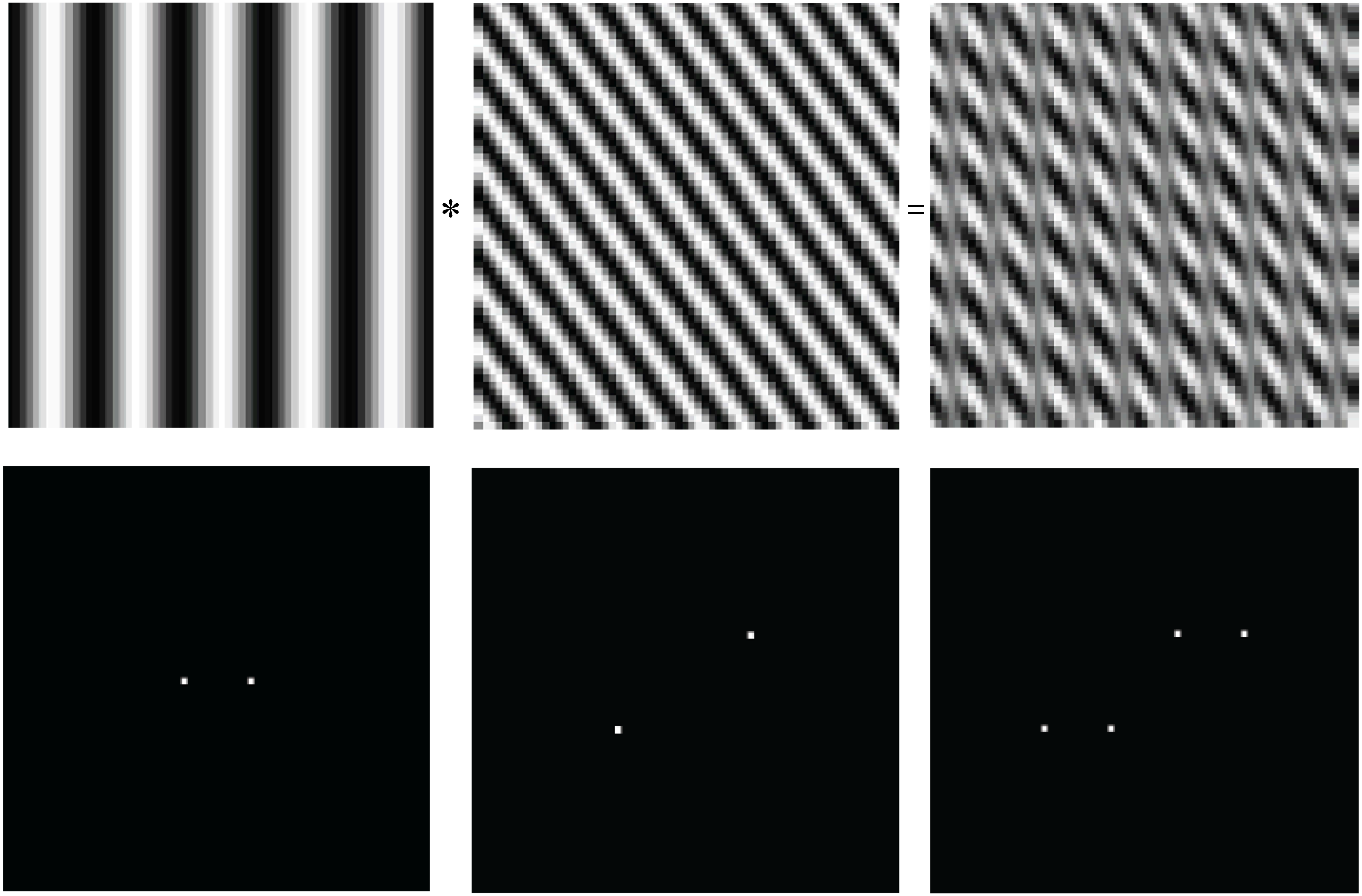
Product of images

The Fourier transform of the product of two images

$$f[n,m] = g[n,m]\, h[n,m]$$

is the convolution of their DFTs:

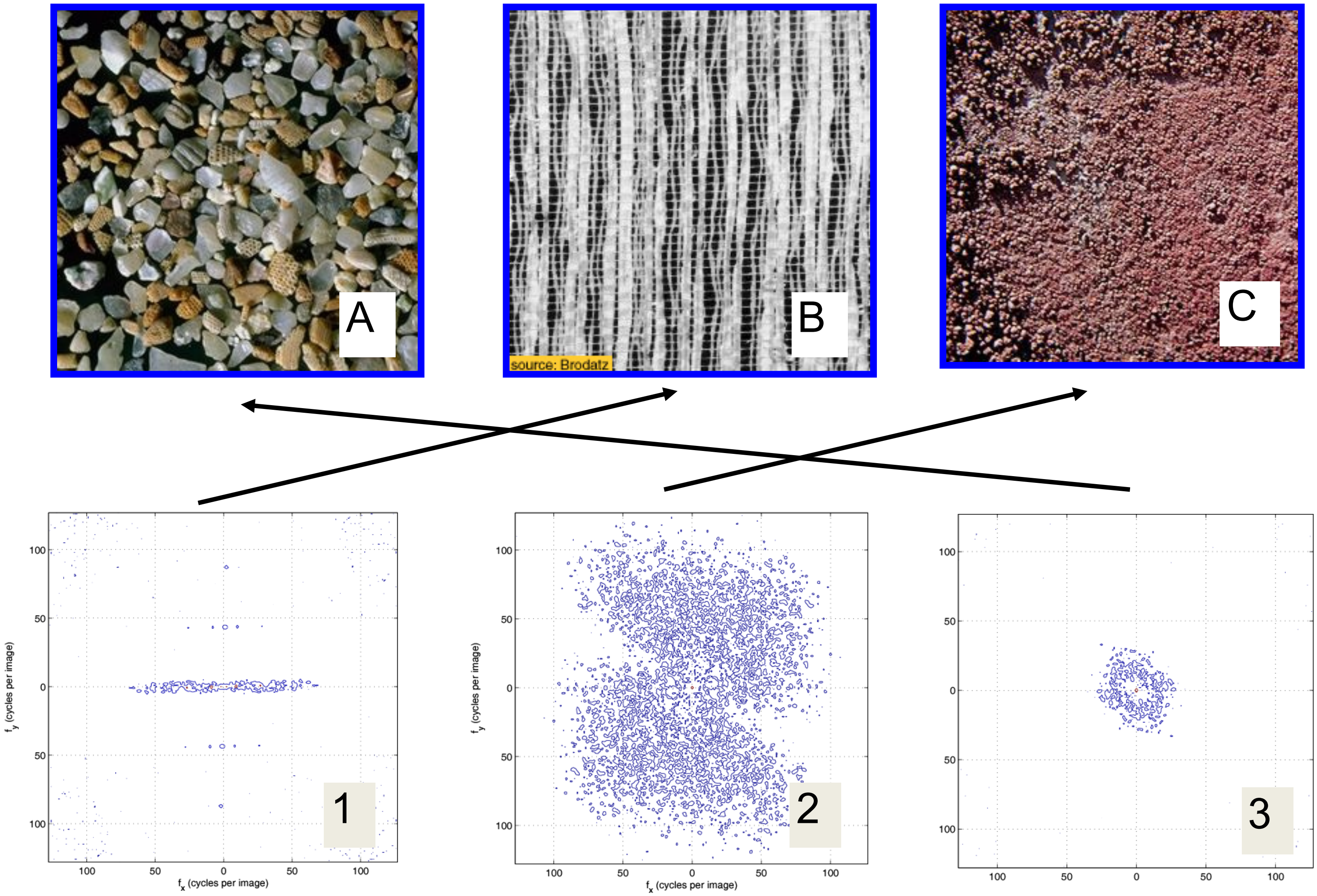
$$F[u,v] = \frac{1}{NM} G[u,v] \circ H[u,v]$$



Game: find the right pairs

Images

DFT magnitude



fx(cycles/image pixel size)

fx(cycles/image pixel size)

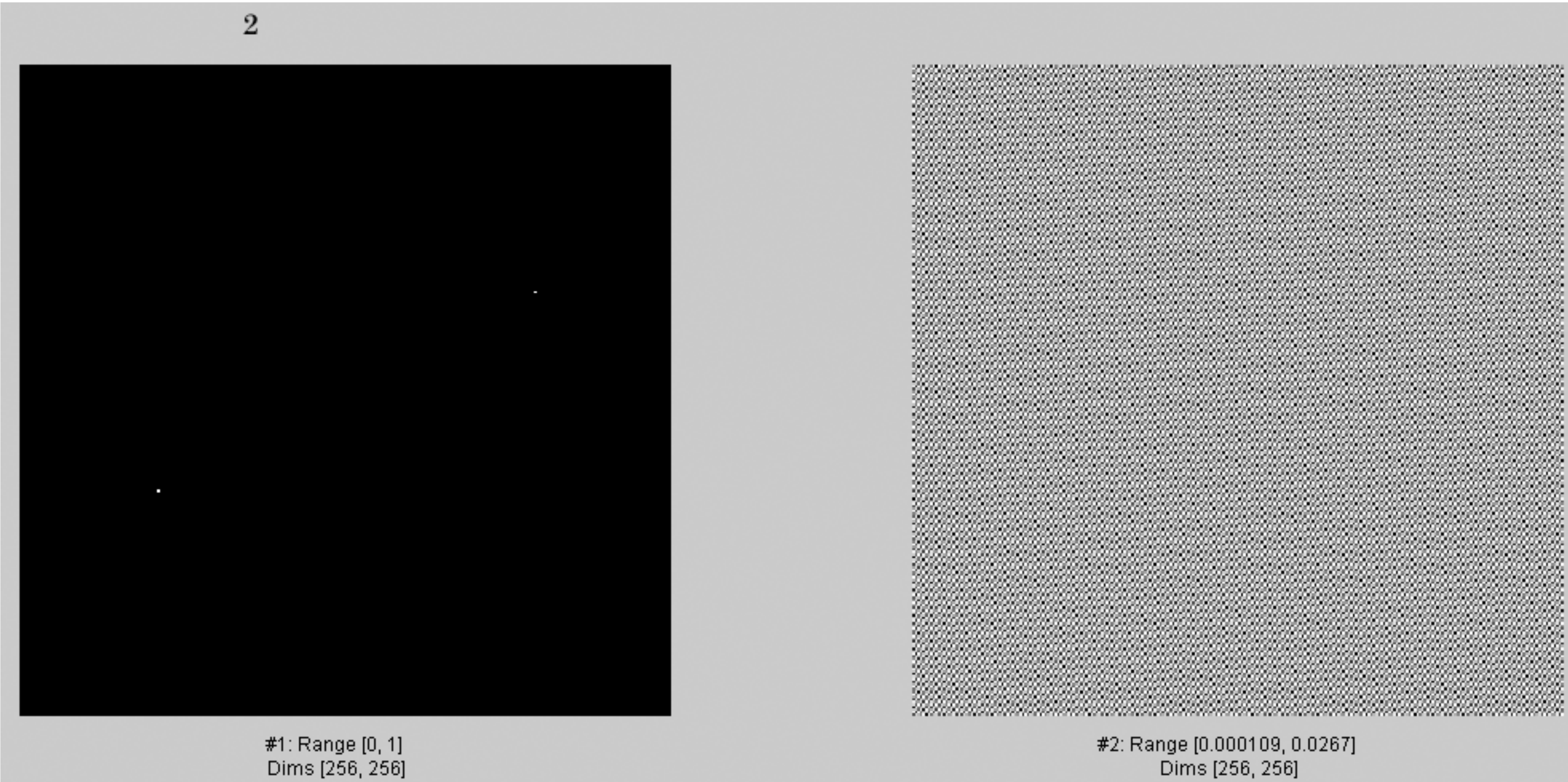
fx(cycles/image pixel size)

The inverse Discrete Fourier transform

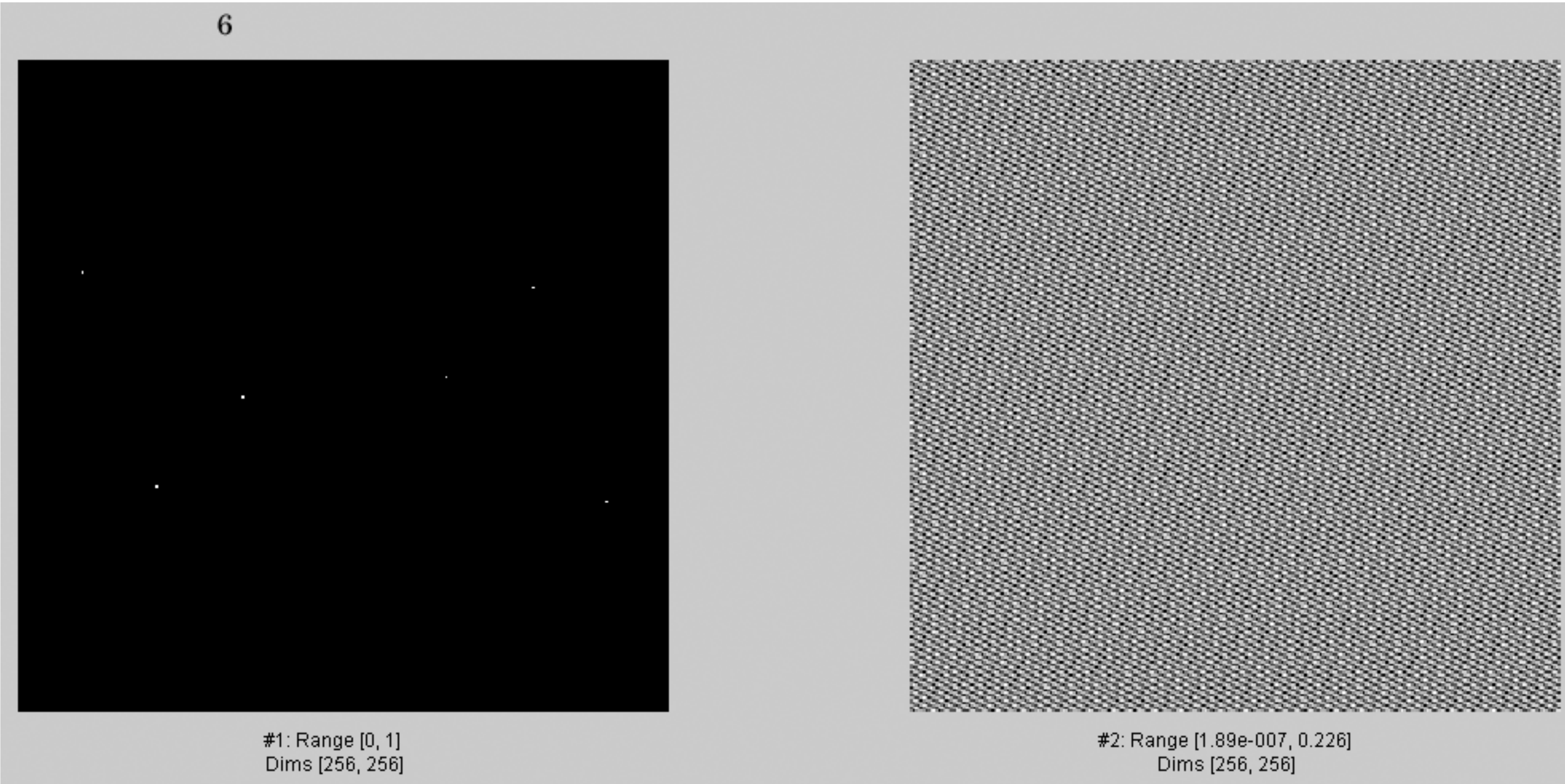
$$f[n,m] = \frac{1}{NM}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F[u,v]\exp\left(+2\pi j\left(\frac{u\,n}{N}+\frac{v\,m}{M}\right)\right)$$

How does summing waves ends up giving back a picture?

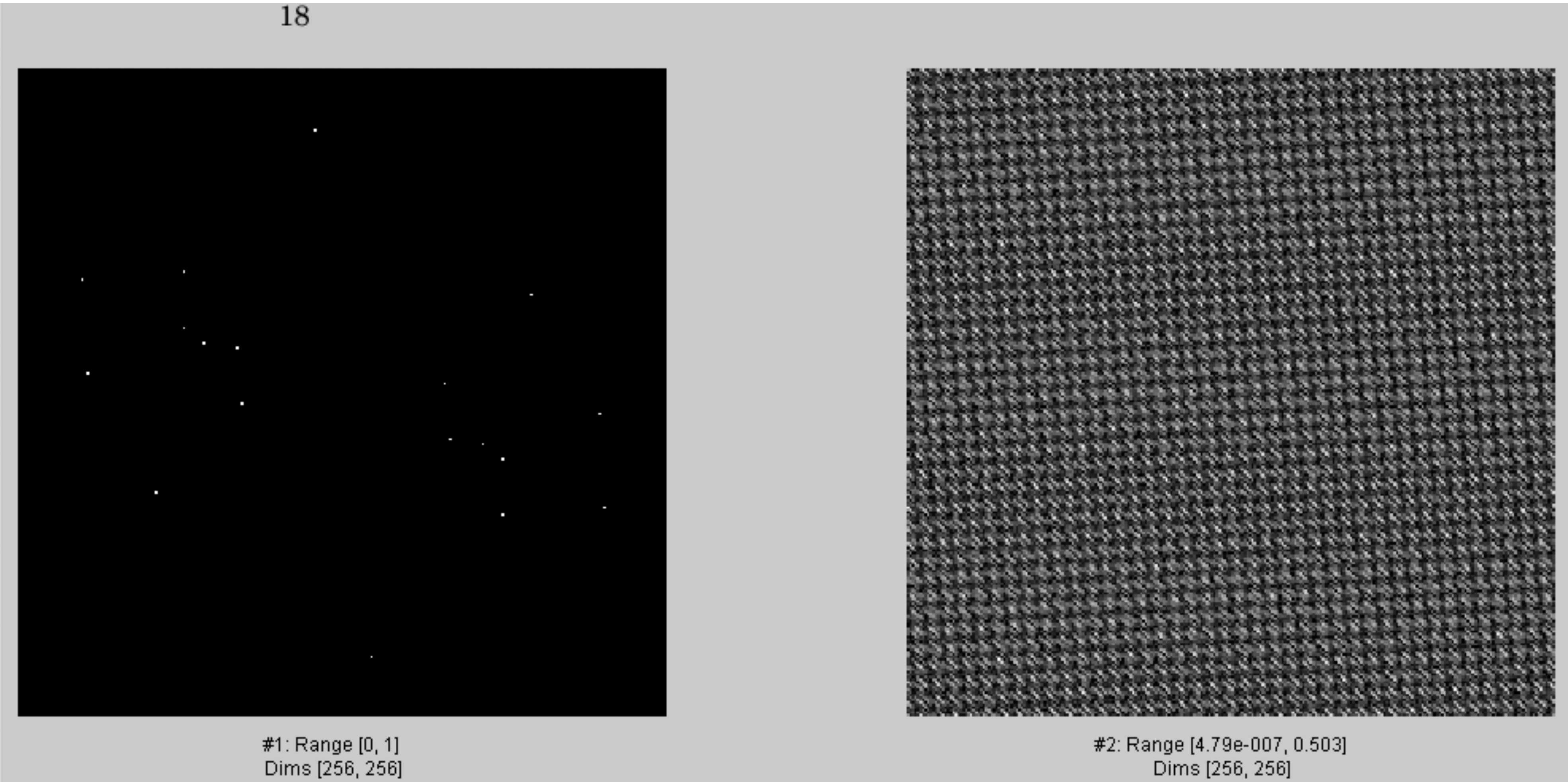
2



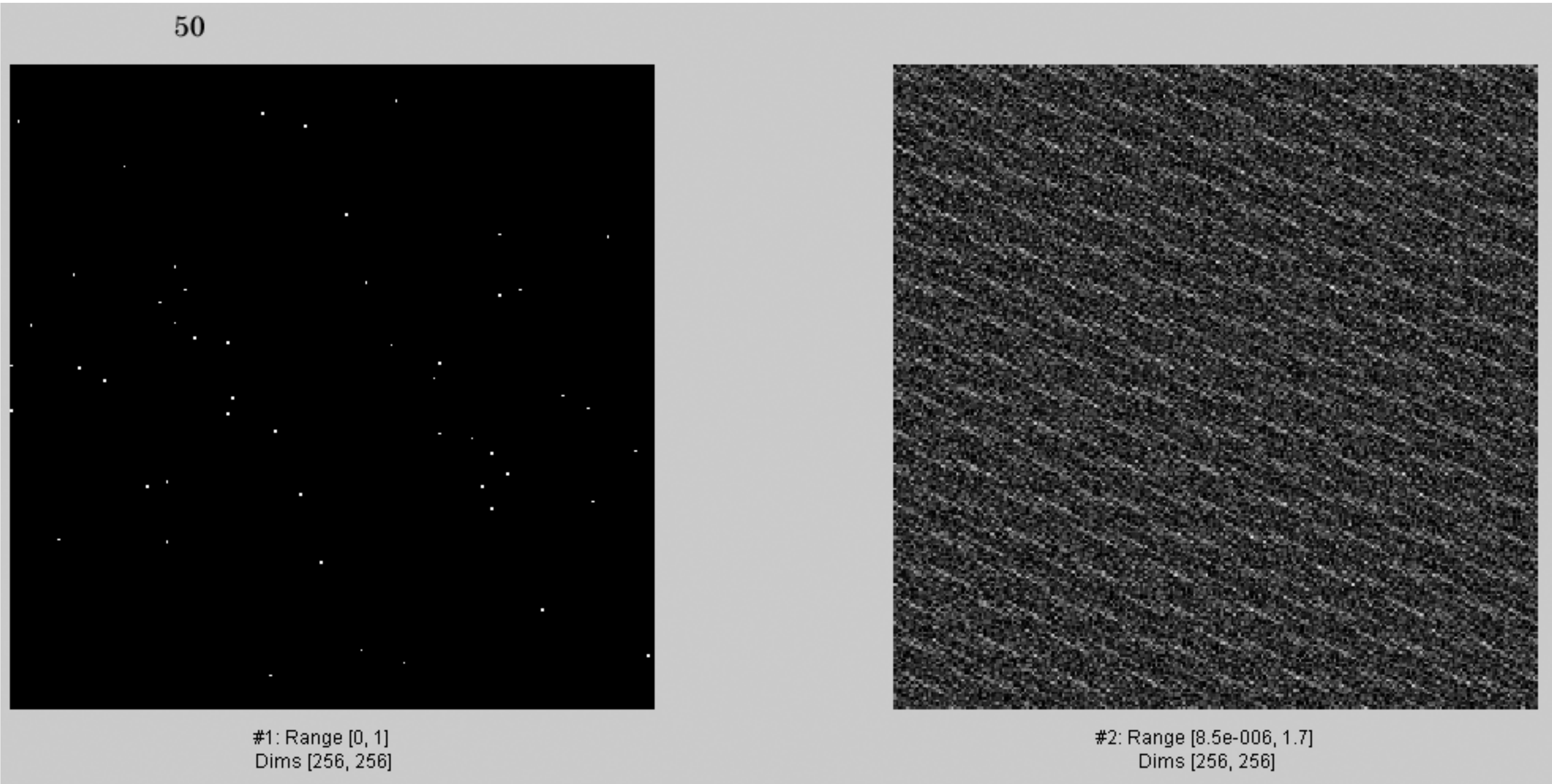
6



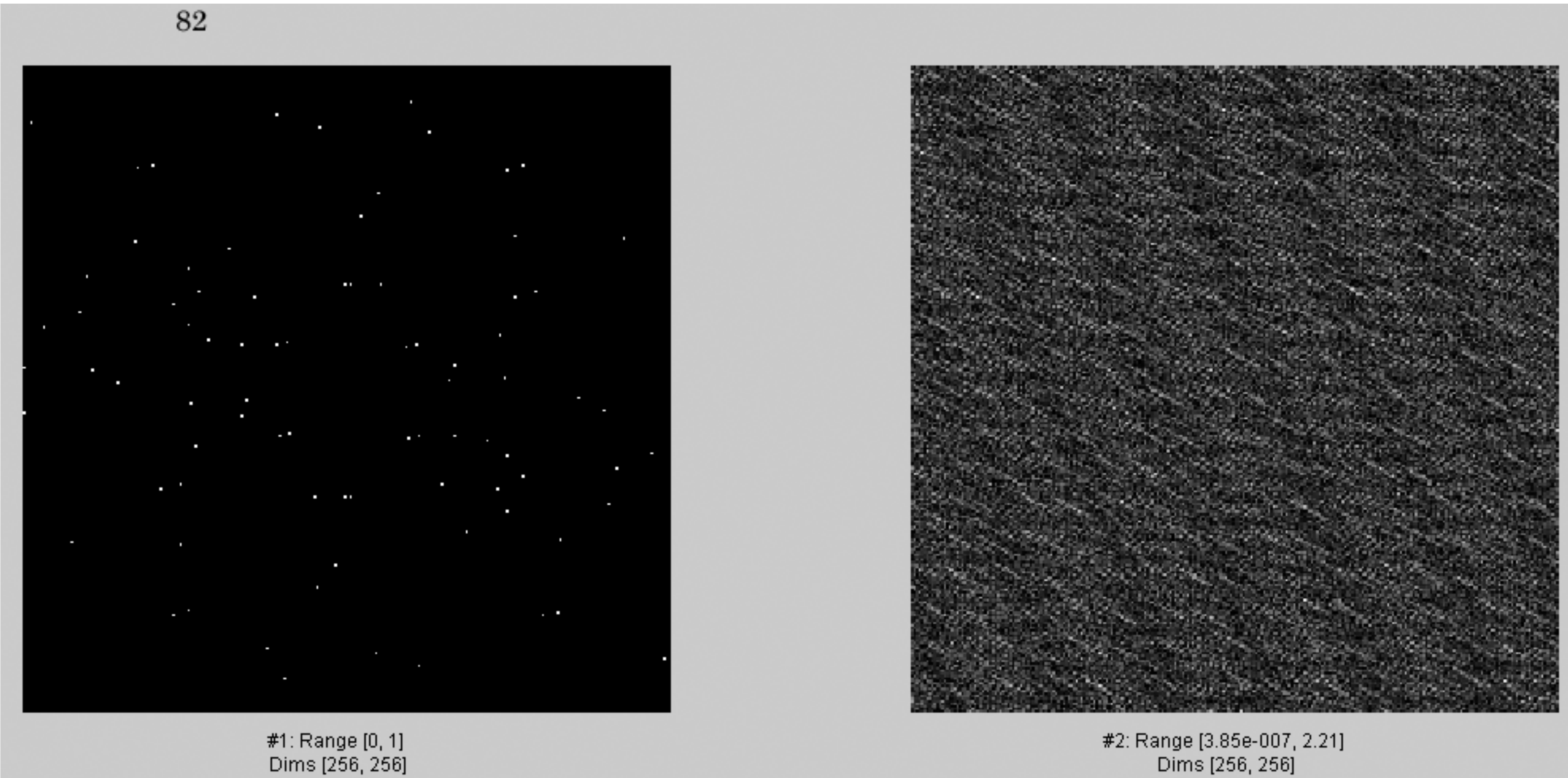
18



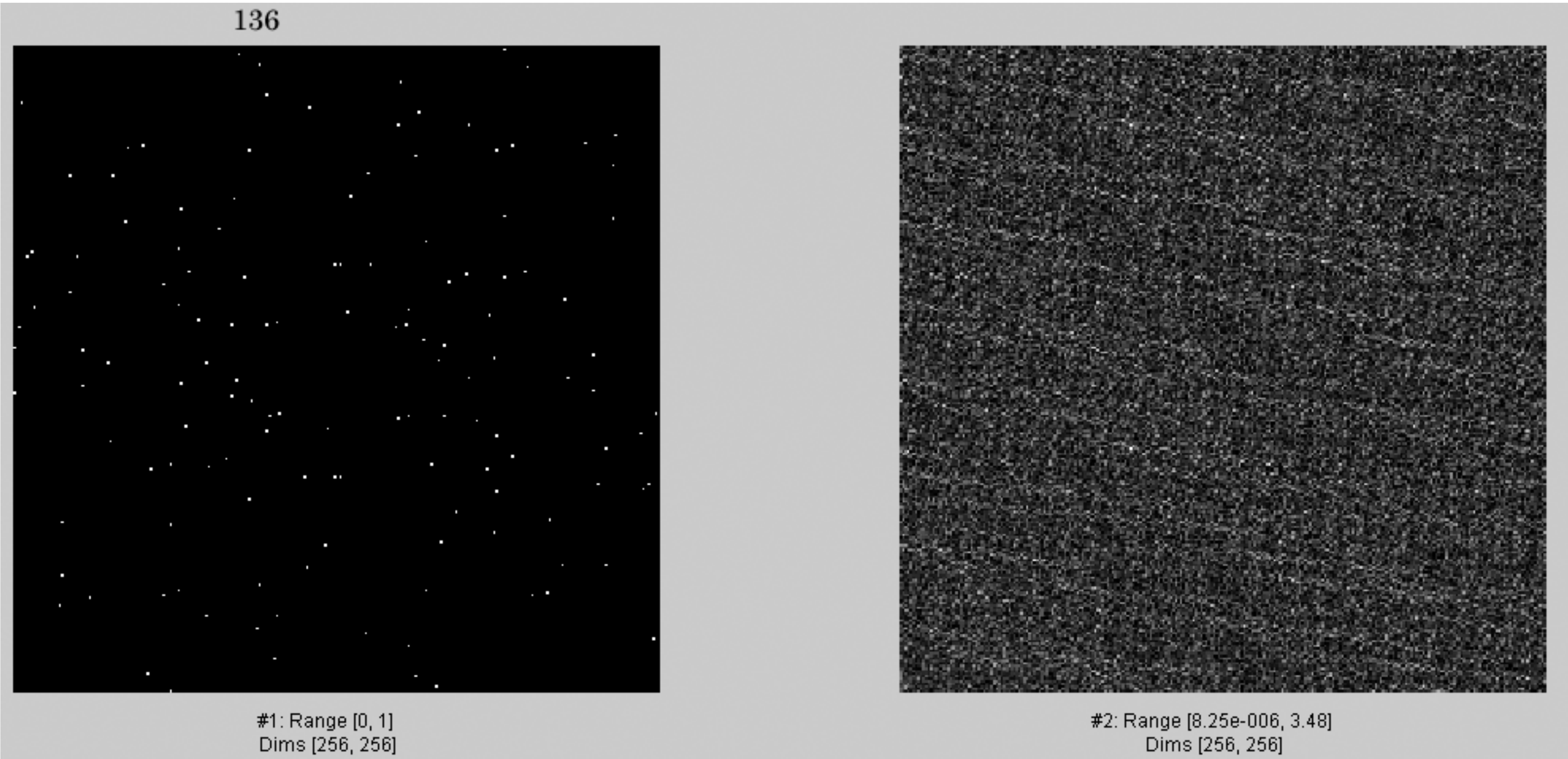
50



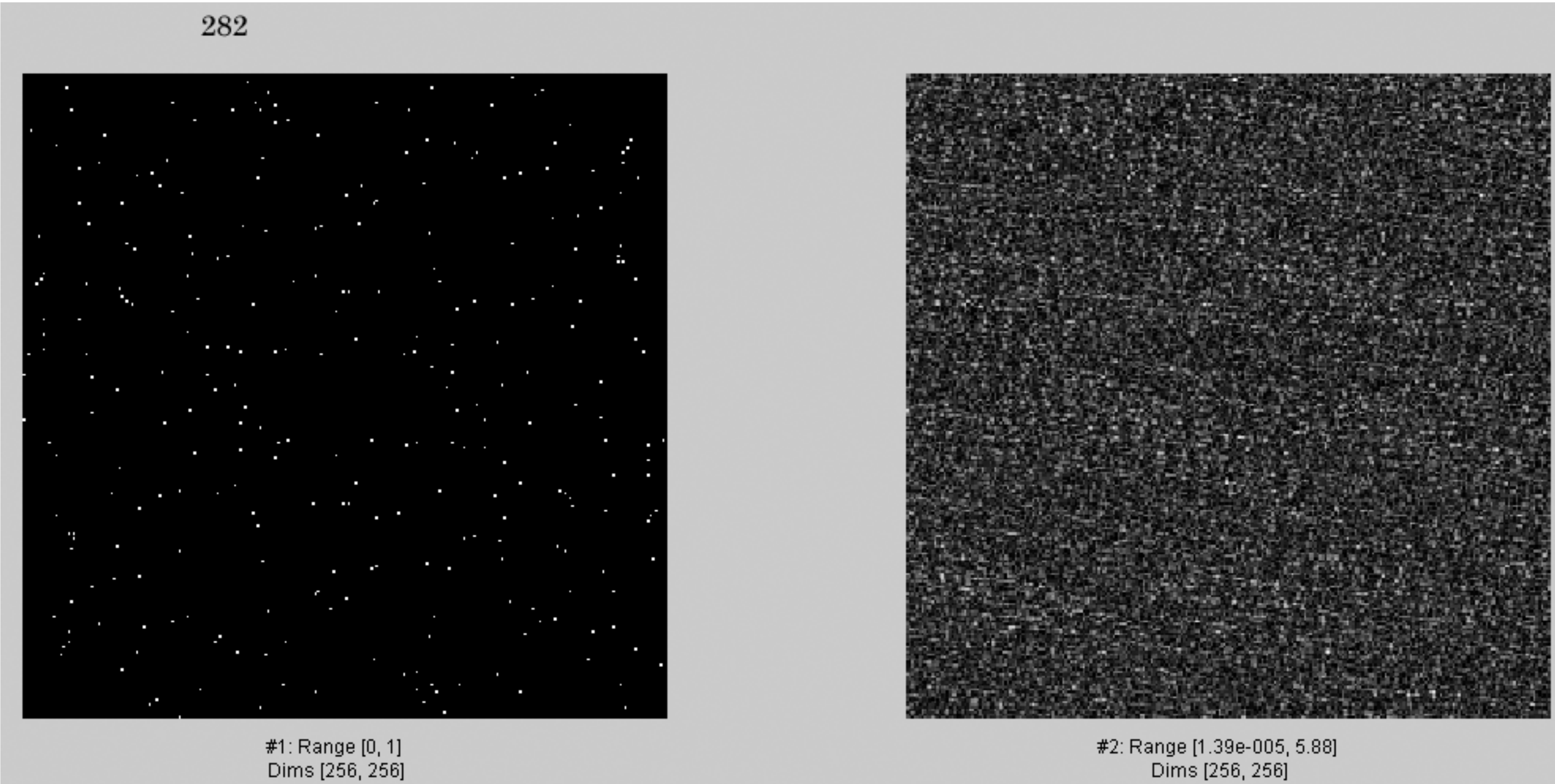
82



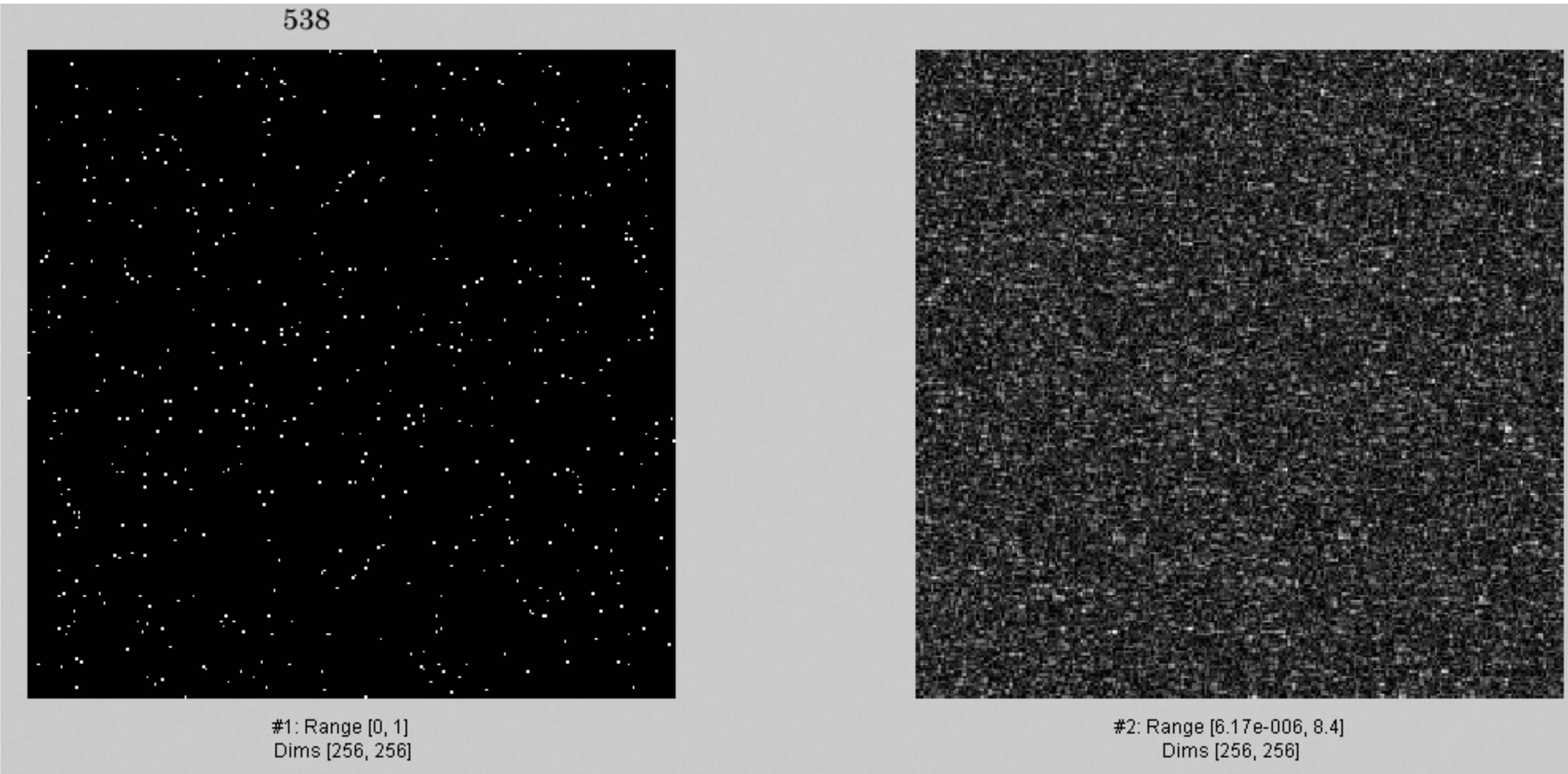
136



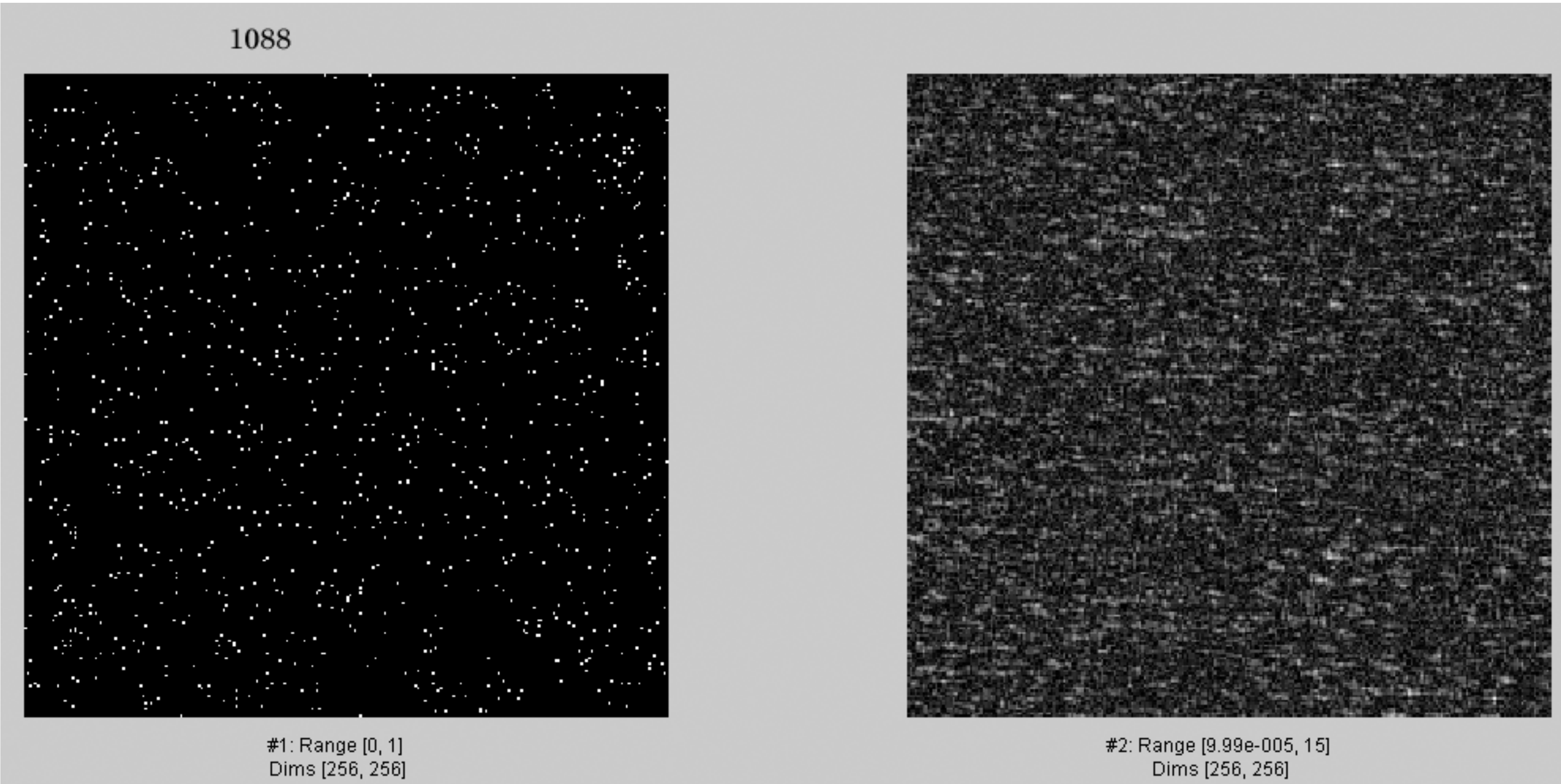
282



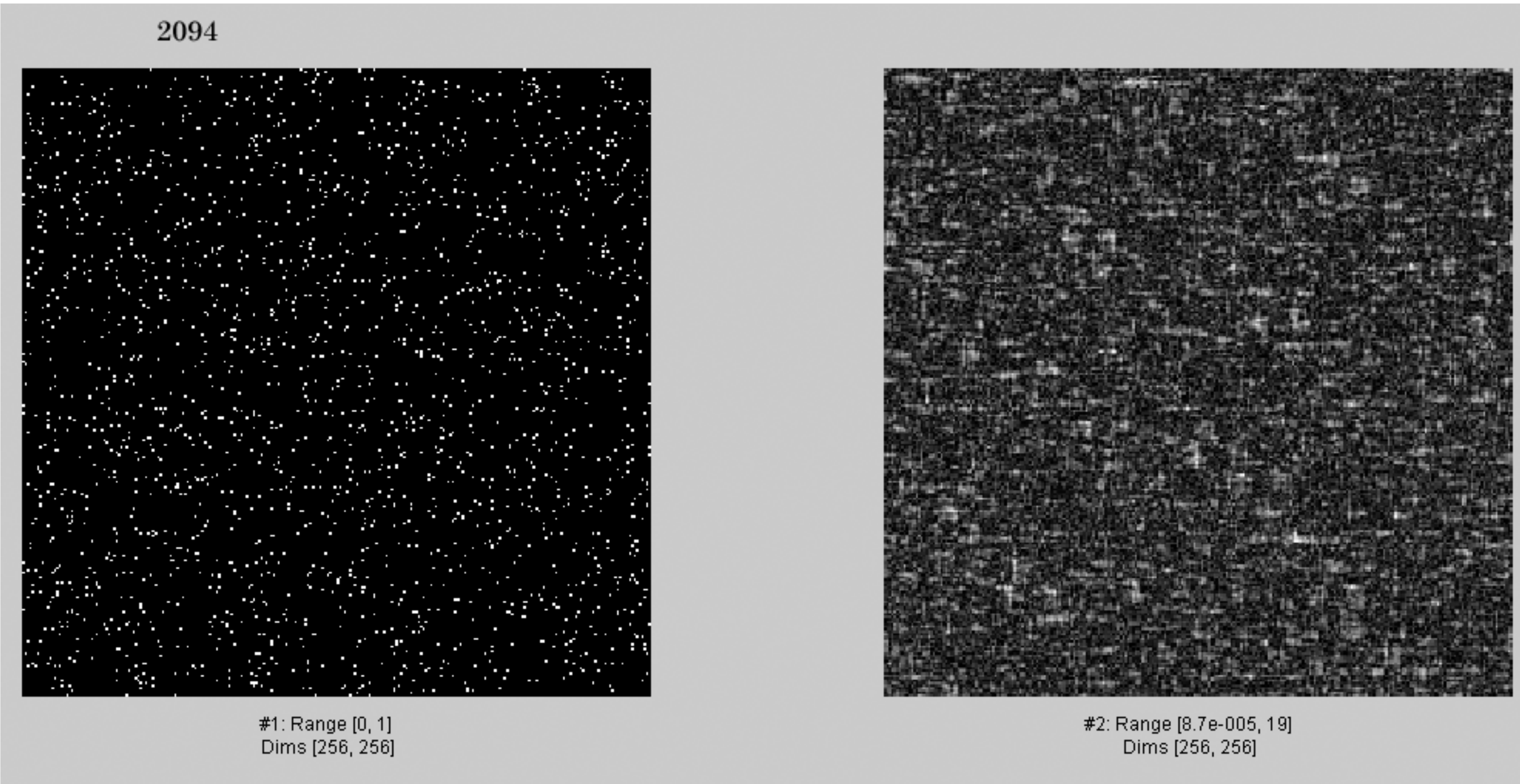
538



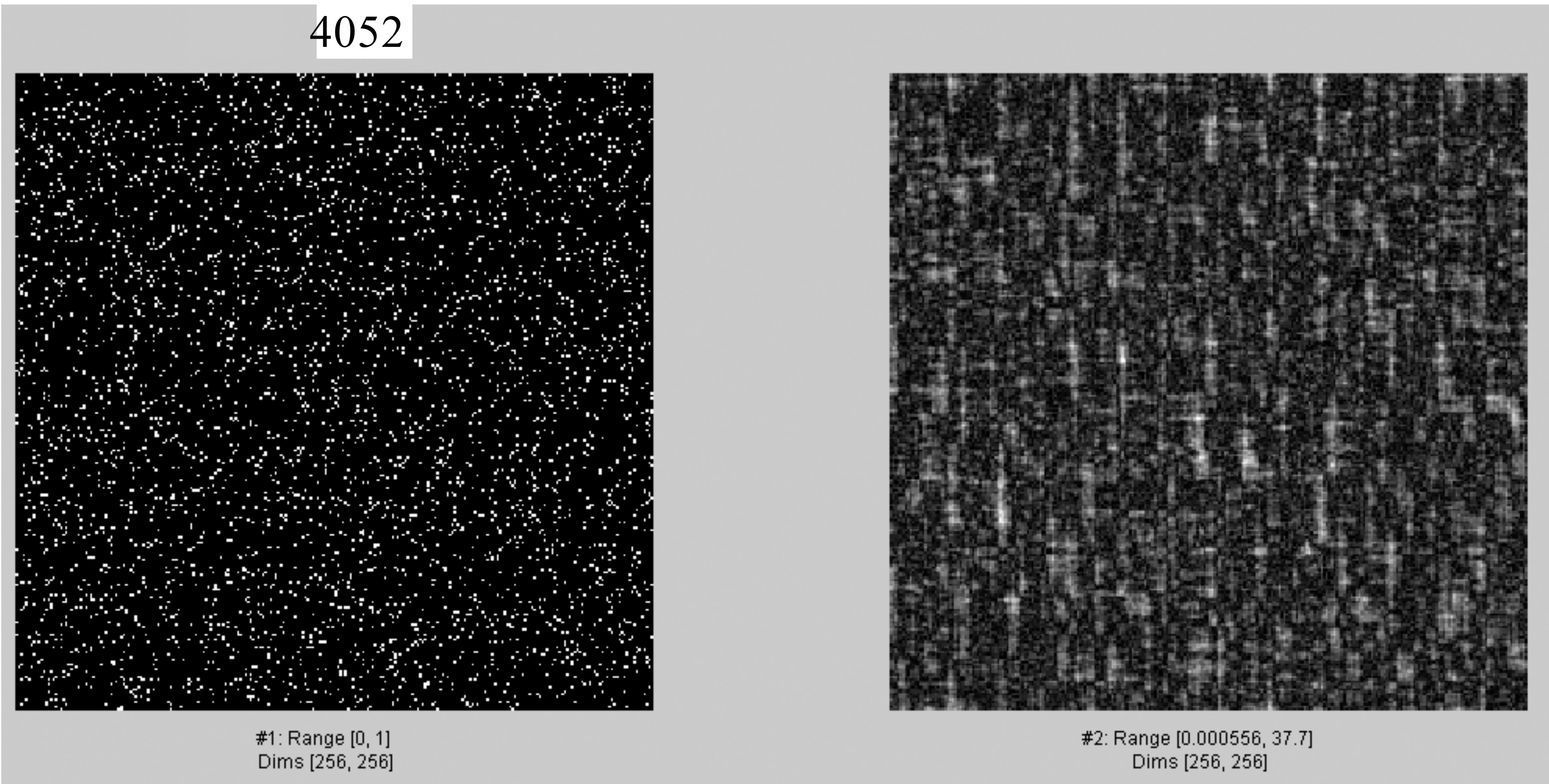
1088



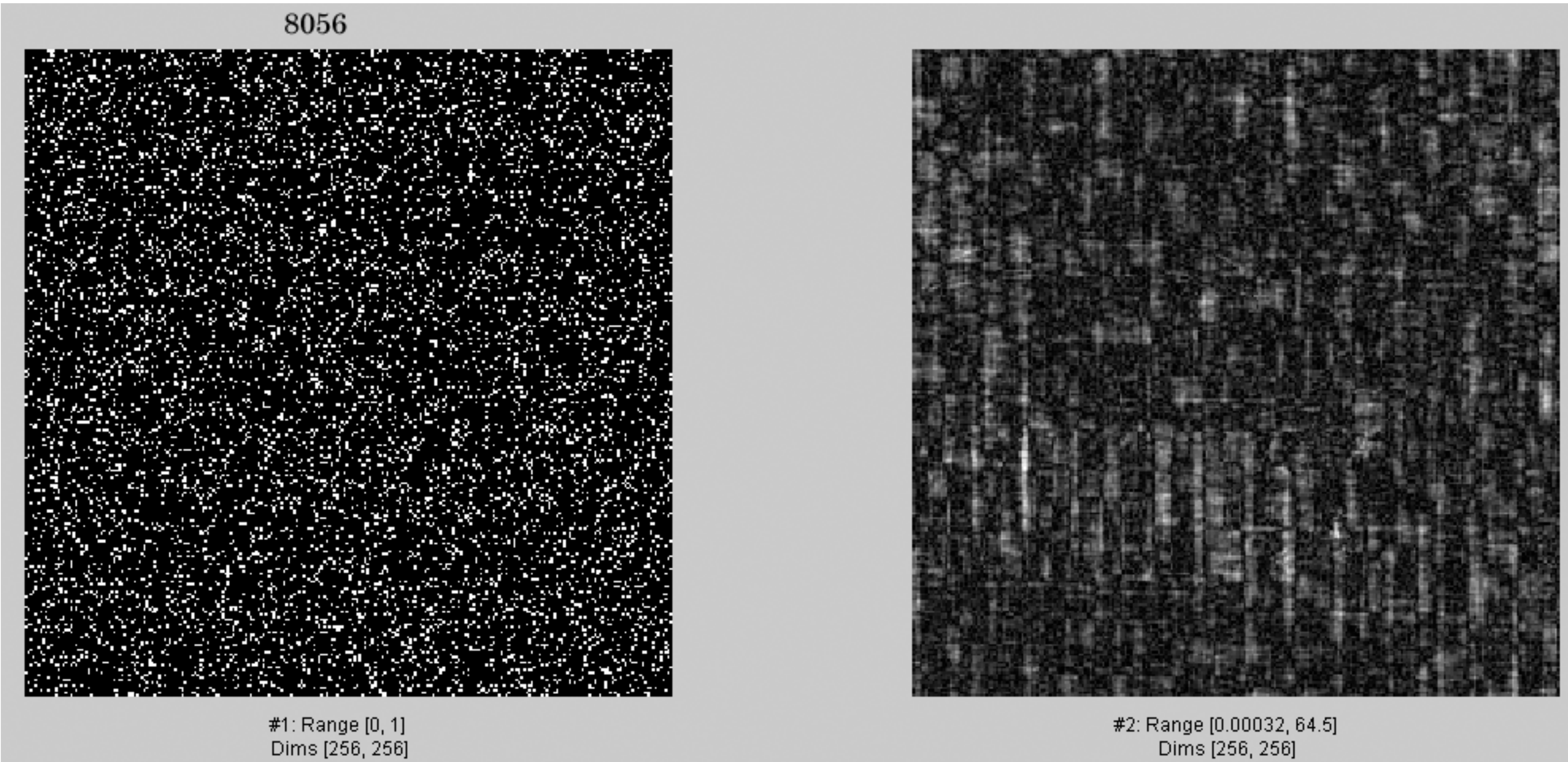
2094



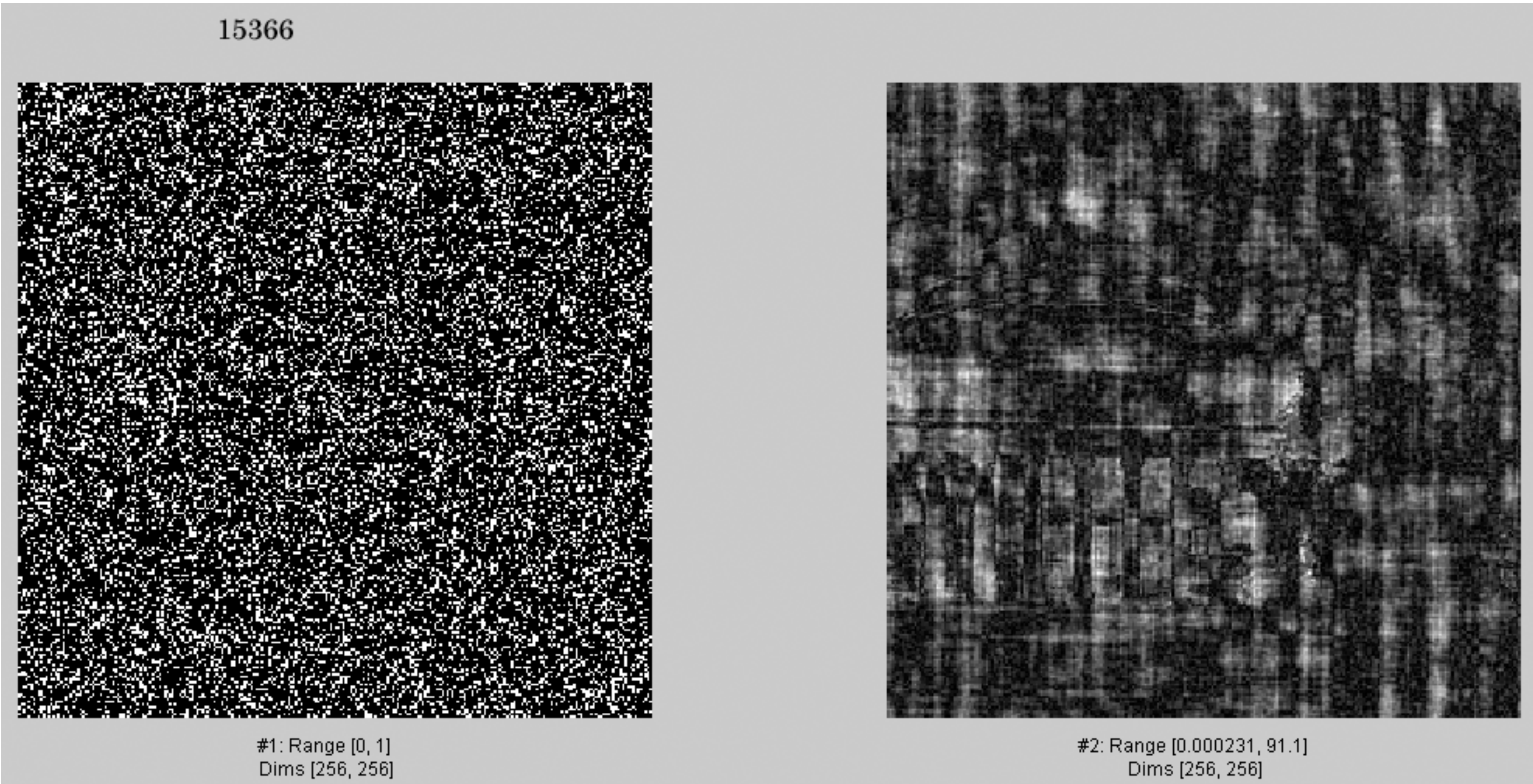
4052.



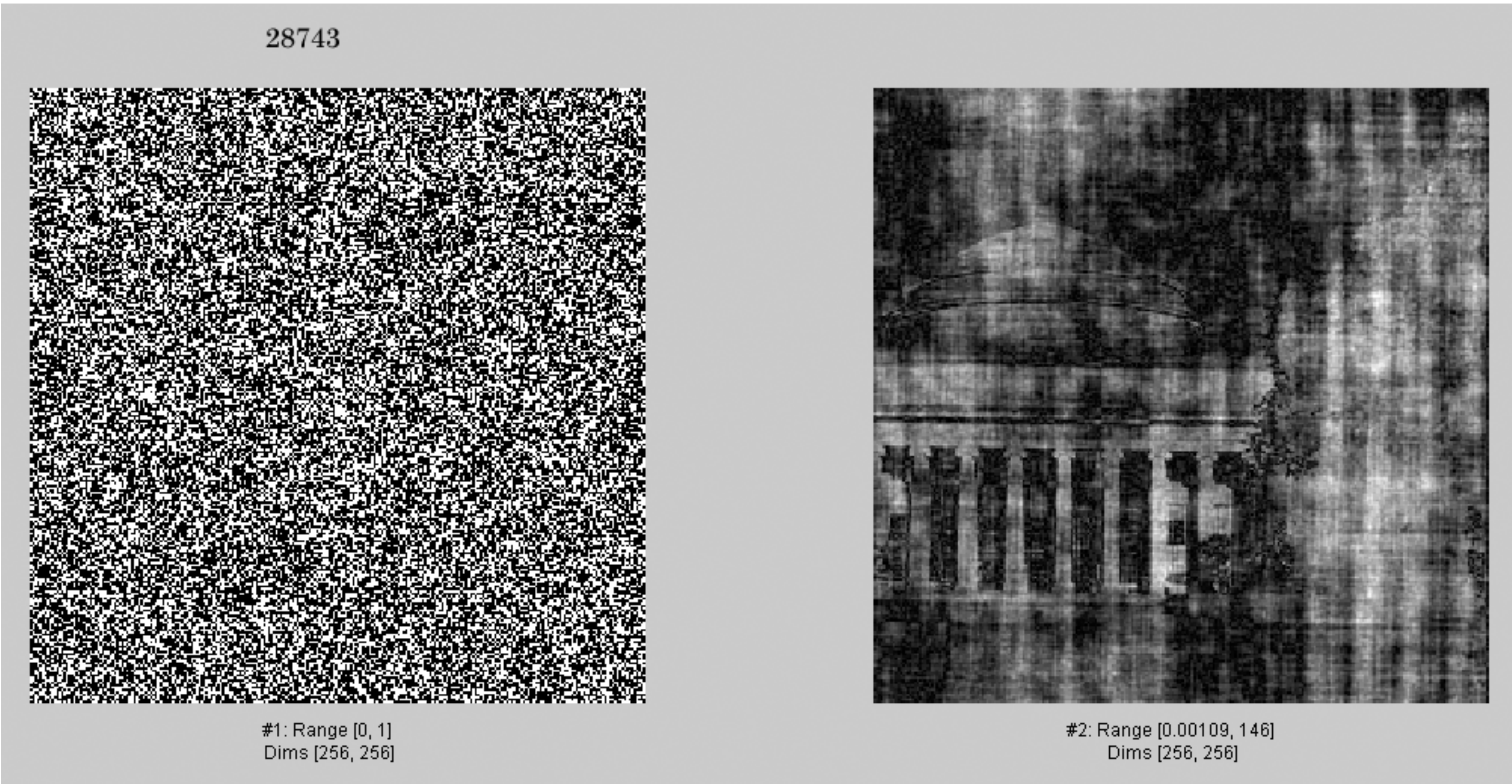
8056.



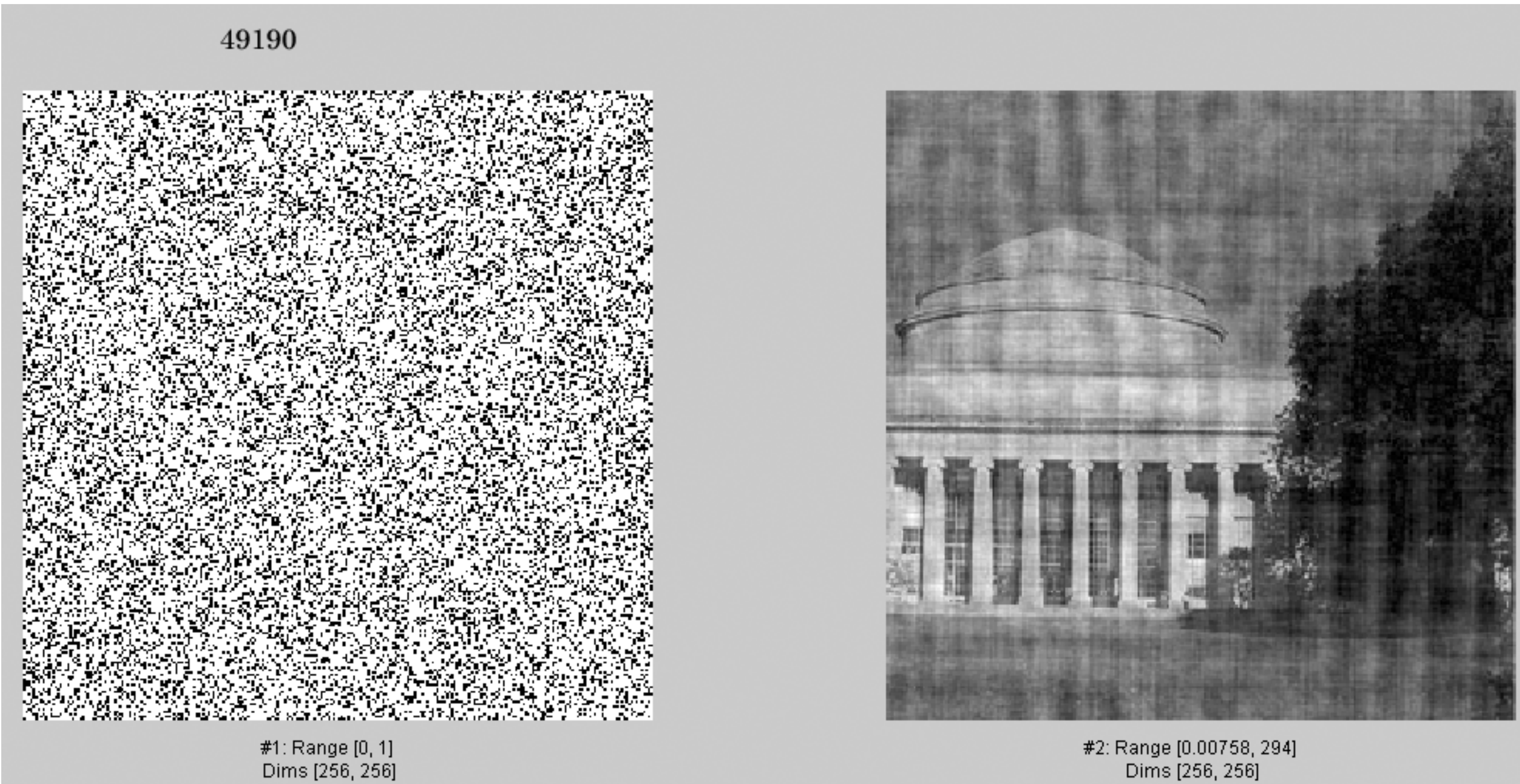
15366



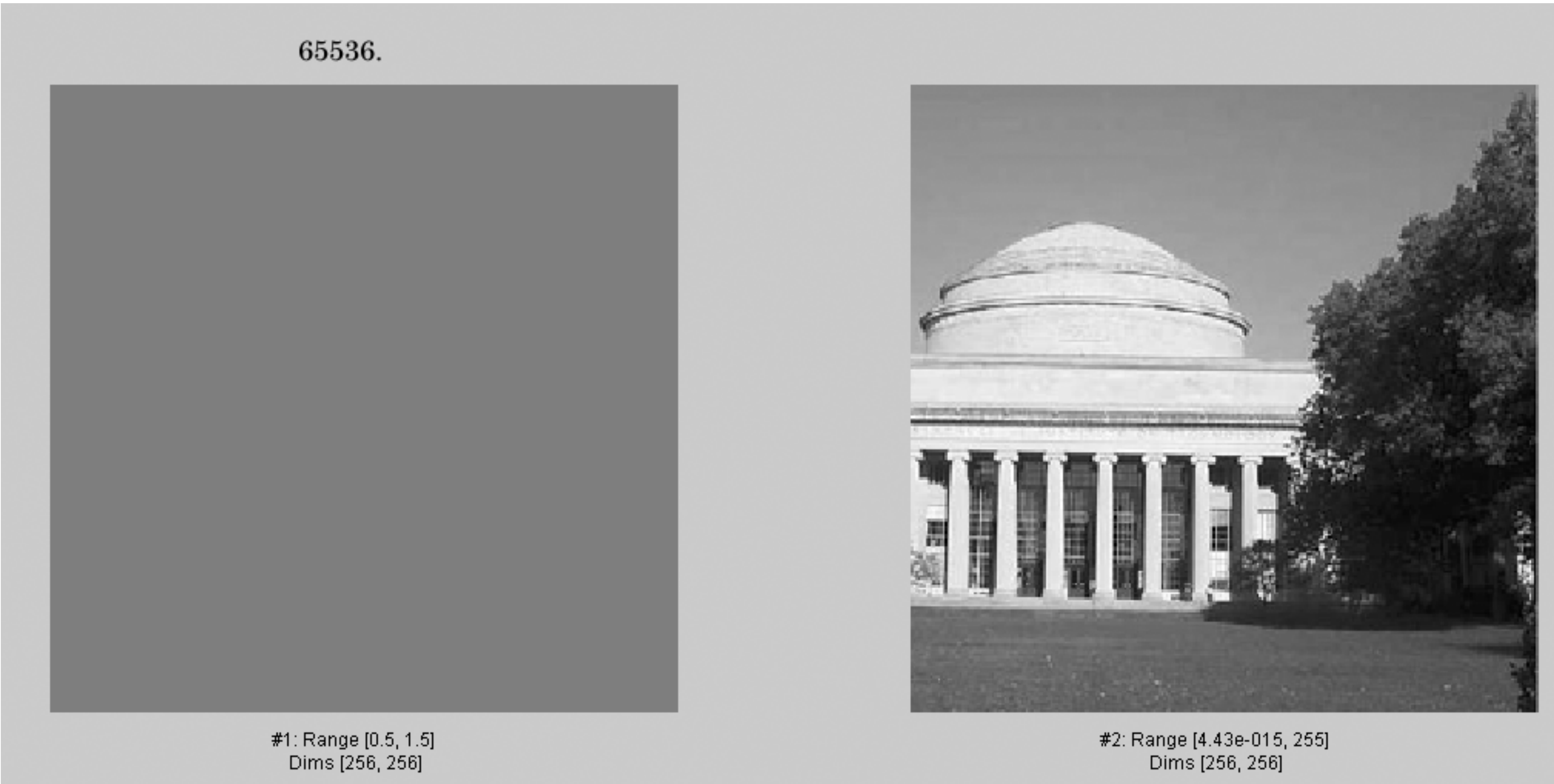
28743



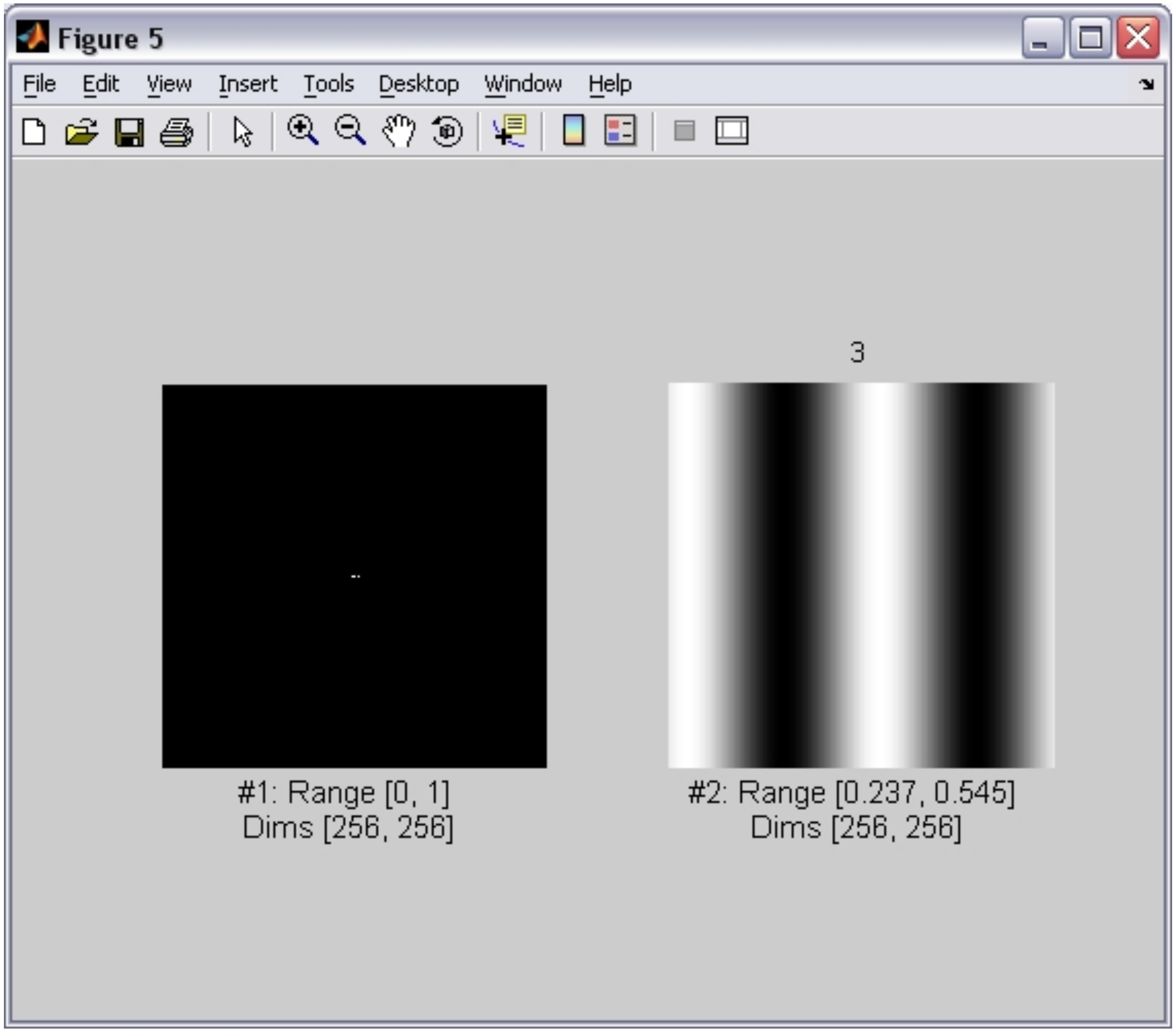
49190.



65536.


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

3



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

5

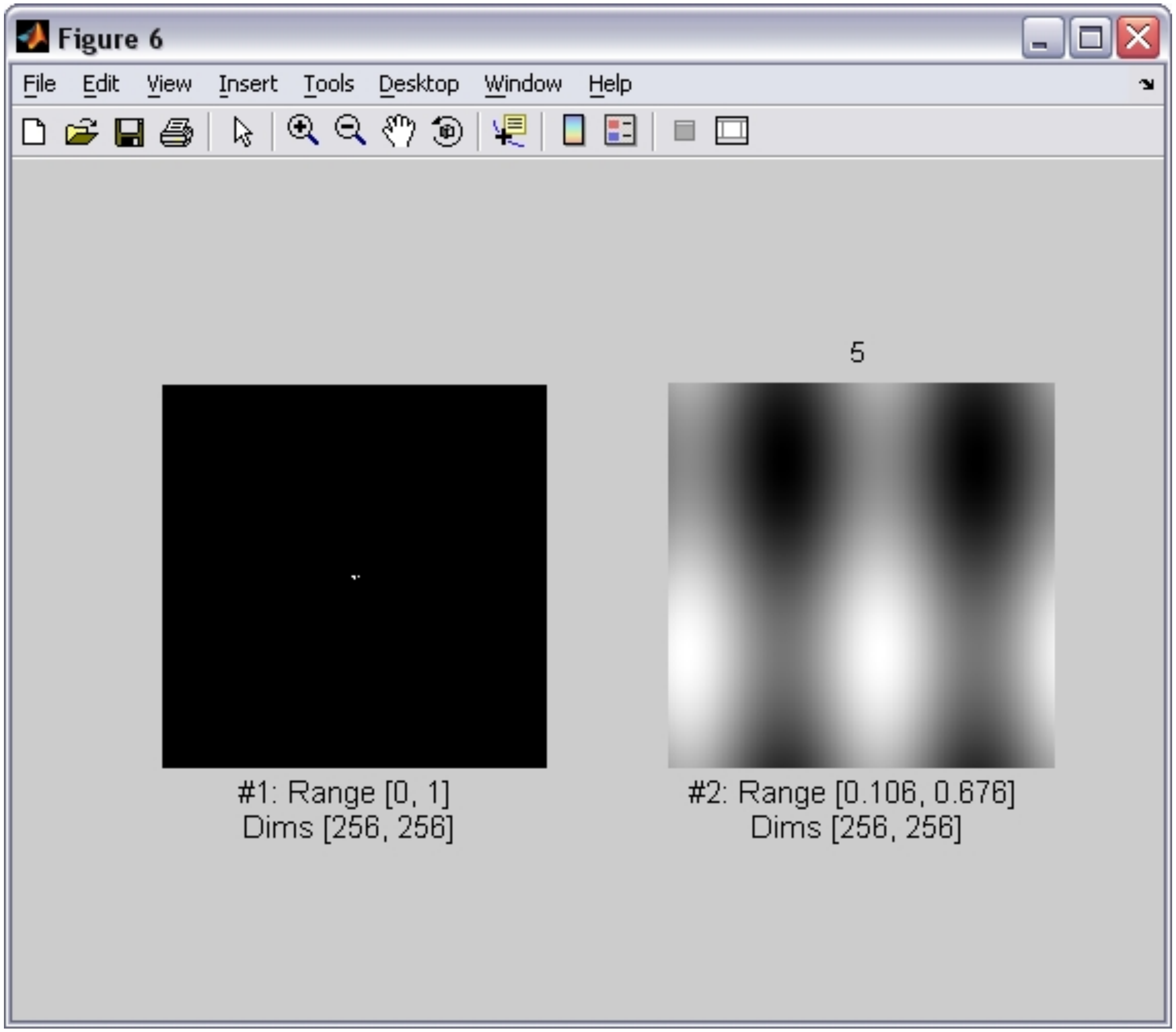


Figure 6
File Edit View Insert Tools Desktop Window Help
5
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.106, 0.676]
Dims [256, 256]

9

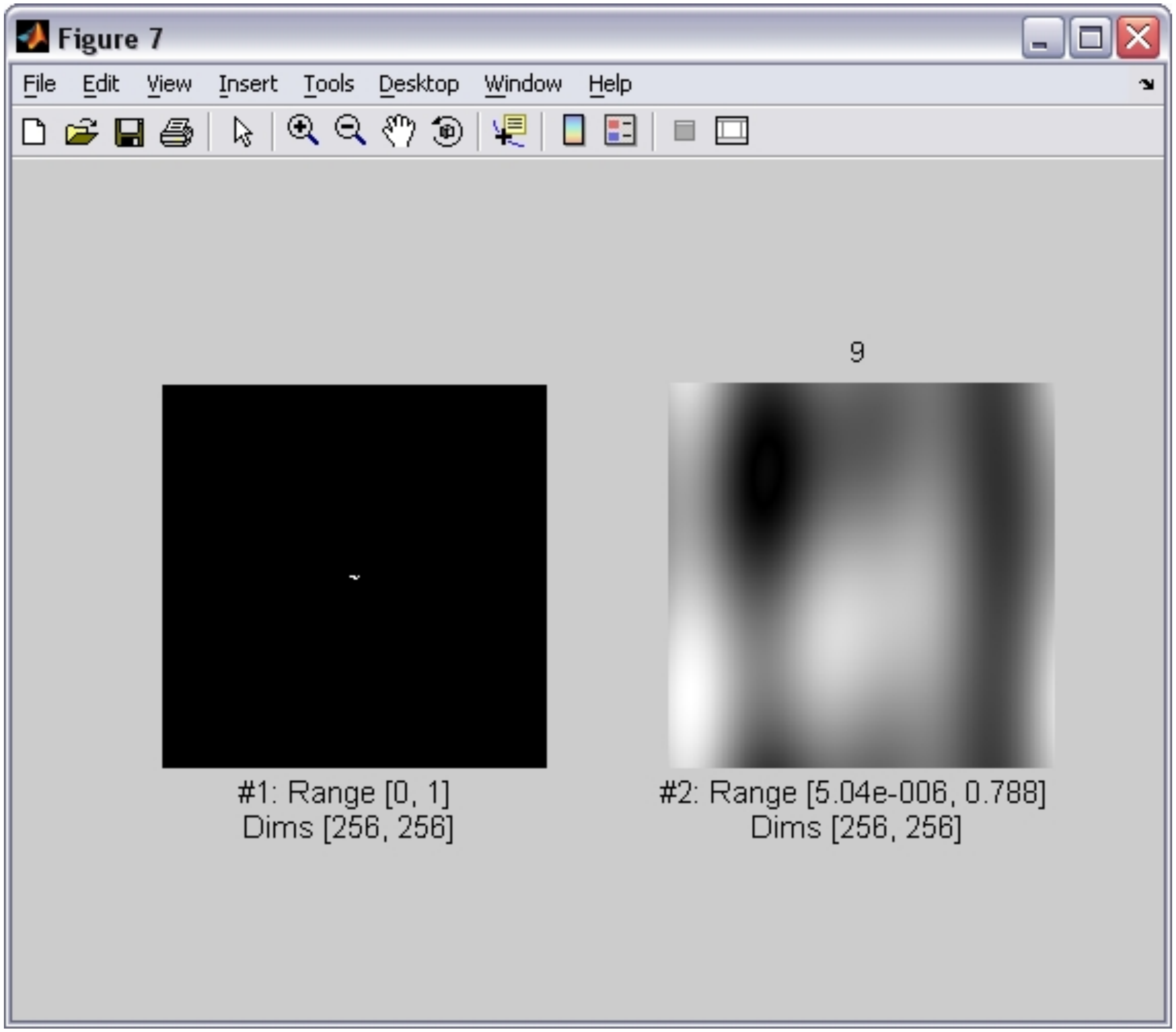


Figure 7
File Edit View Insert Tools Desktop Window Help
9
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.04e-006, 0.788]
Dims [256, 256]

17

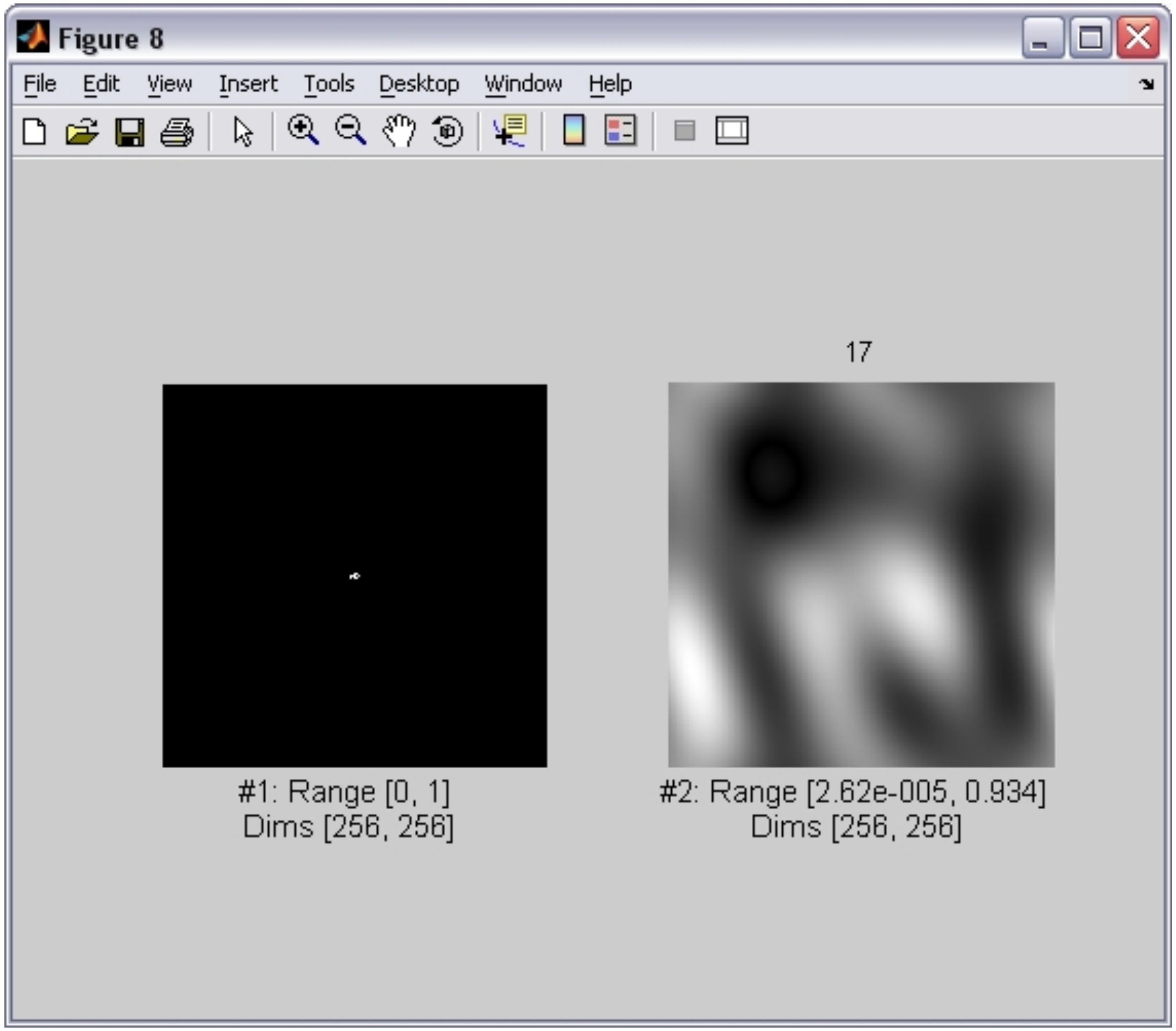


Figure 8
File
Edit
View
Insert
Tools
Desktop
Window
Help
17
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.62e-005, 0.934]
Dims [256, 256]

33

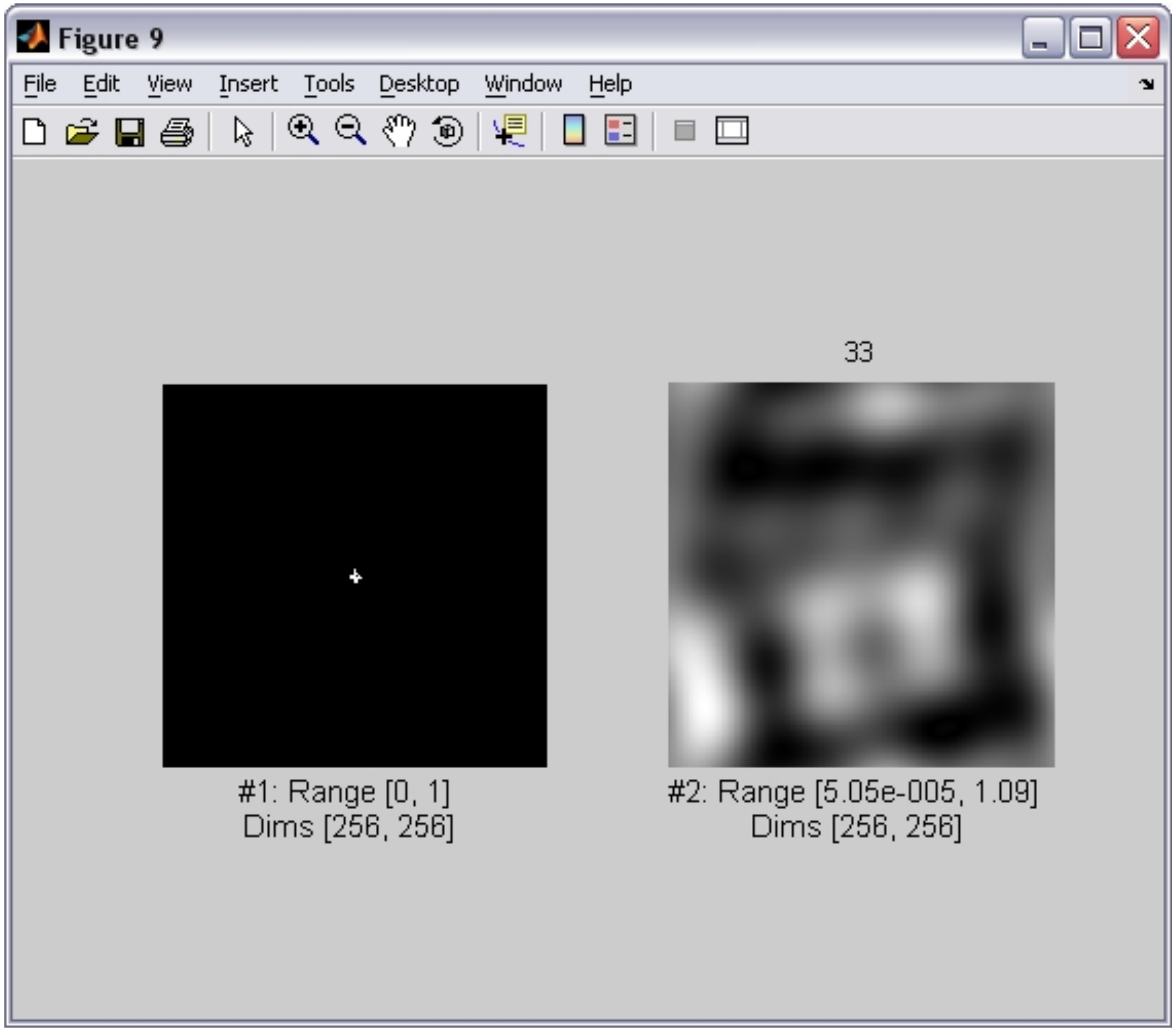


Figure 9
File Edit View Insert Tools Desktop Window Help
33
#1: Range [0, 1]
Dims [256, 256]
#2: Range [5.05e-005, 1.09]
Dims [256, 256]

65

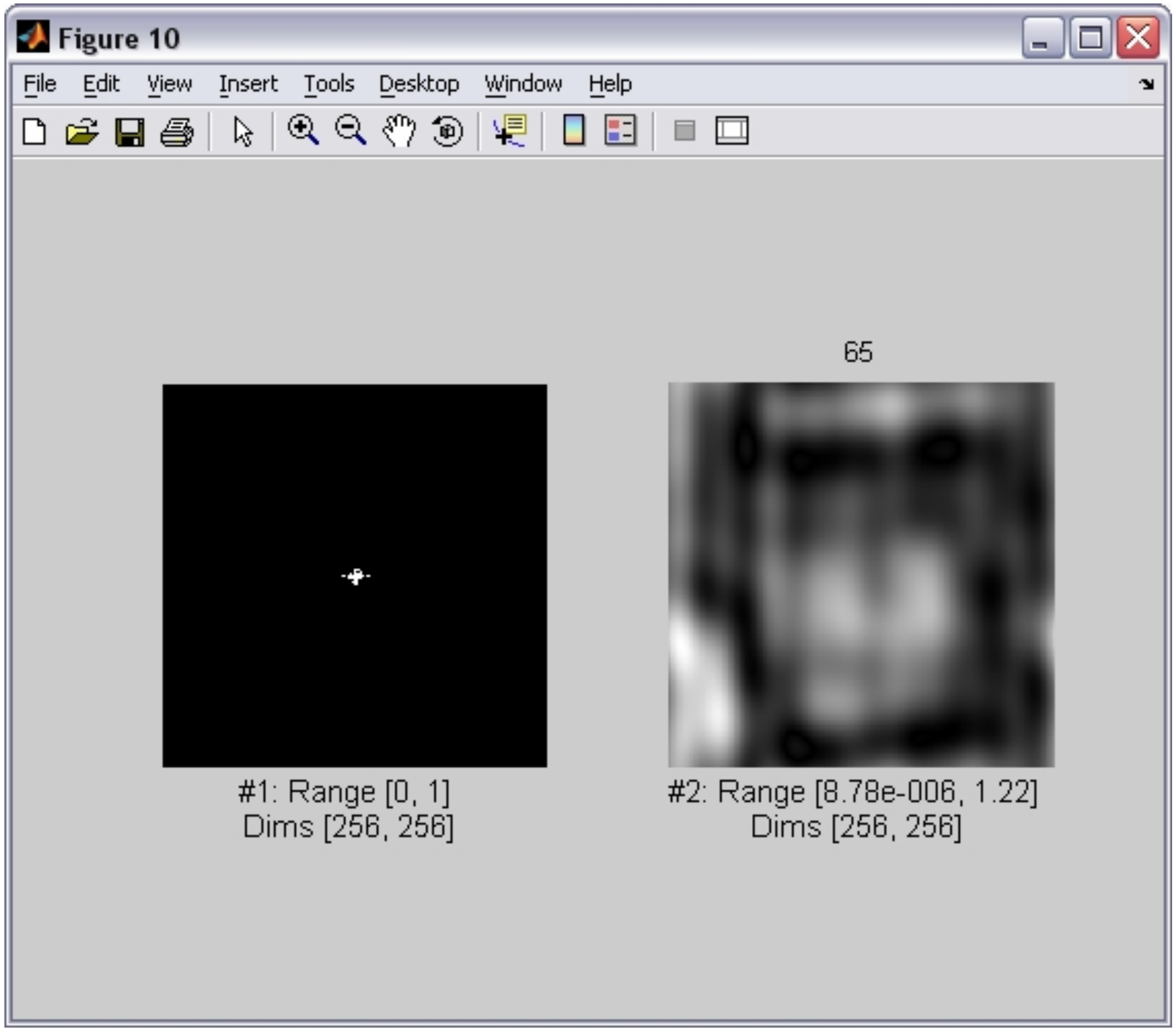


Figure 10
File Edit View Insert Tools Desktop Window Help
65
#1: Range [0, 1]
Dims [256, 256]
#2: Range [8.78e-006, 1.22]
Dims [256, 256]

129

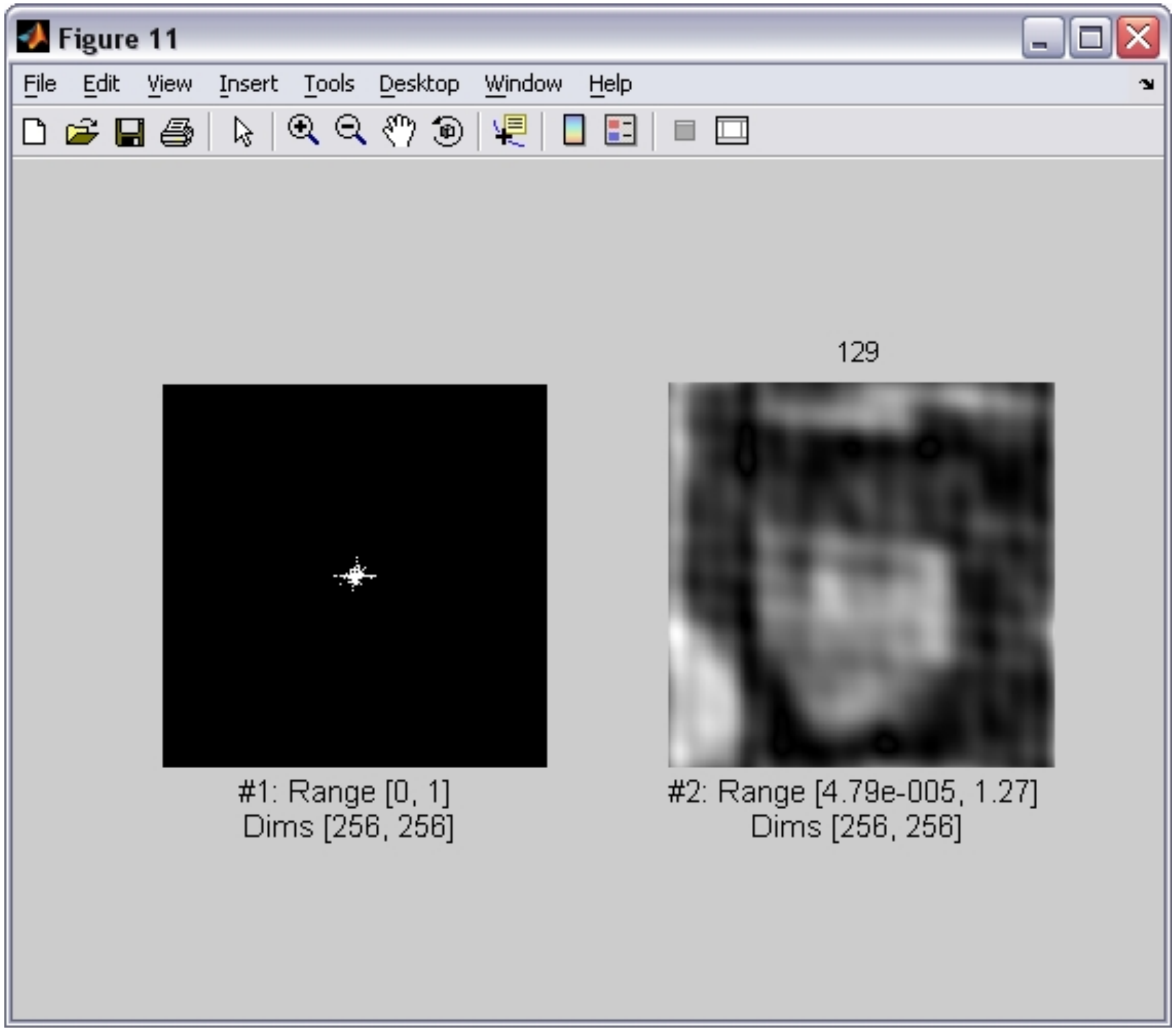


Figure 11
File
Edit
View
Insert
Tools
Desktop
Window
Help
129
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.79e-005, 1.27]
Dims [256, 256]

257

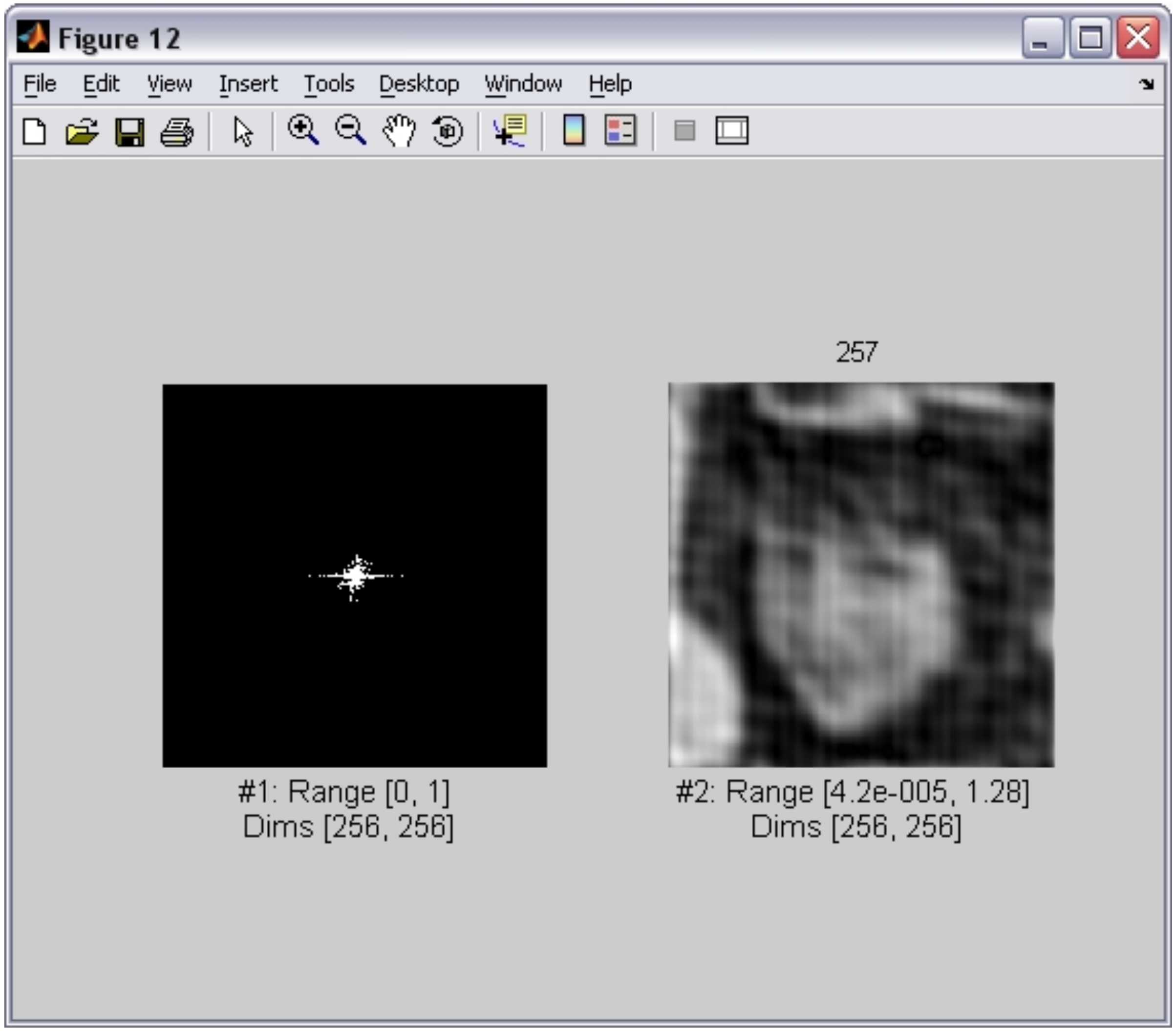


Figure 12
File Edit View Insert Tools Desktop Window Help
257
#1: Range [0, 1]
Dims [256, 256]
#2: Range [4.2e-005, 1.28]
Dims [256, 256]

513

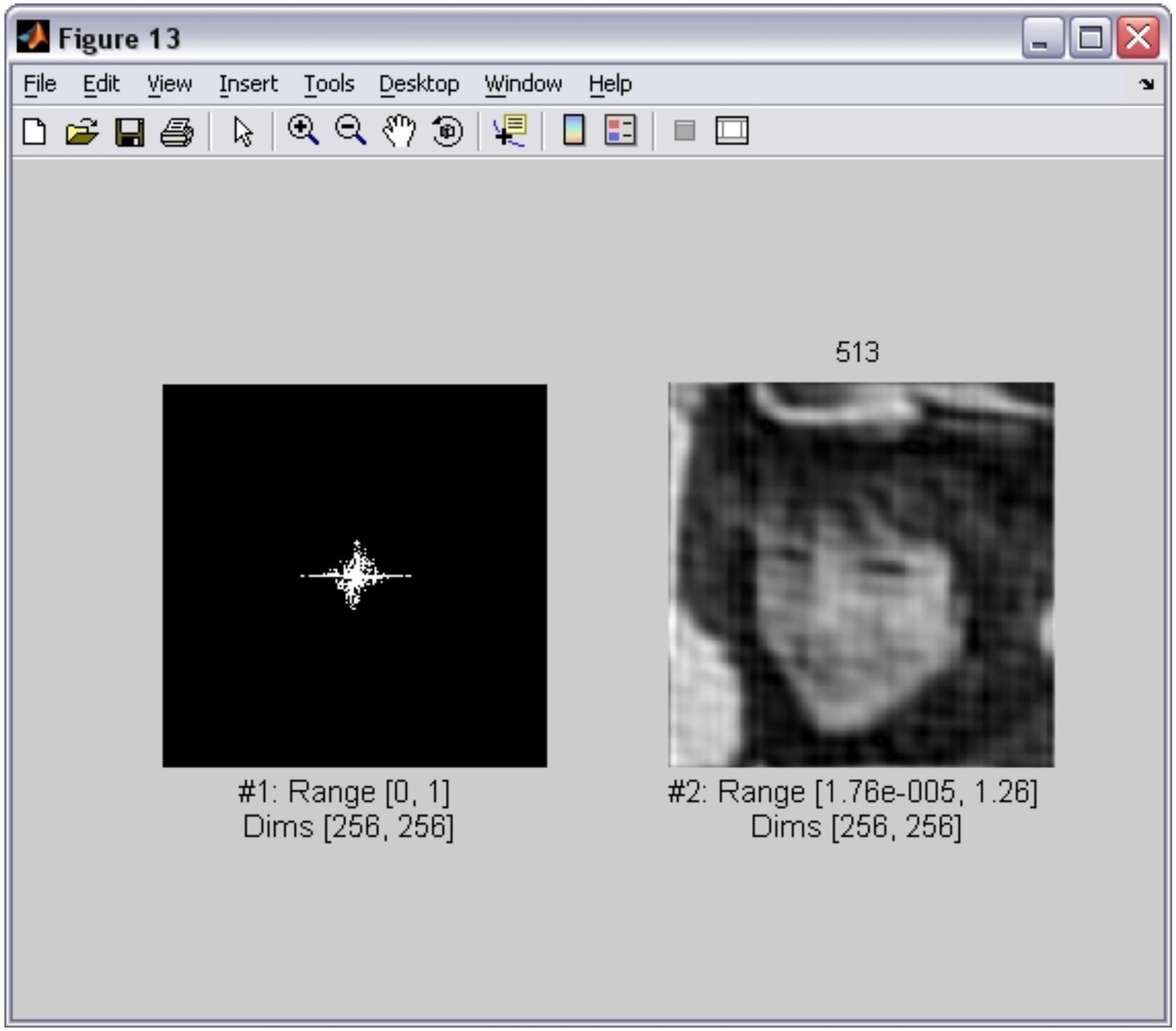


Figure 13
File Edit View Insert Tools Desktop Window Help
513
#1: Range [0, 1]
Dims [256, 256]
#2: Range [1.76e-005, 1.26]
Dims [256, 256]

1025

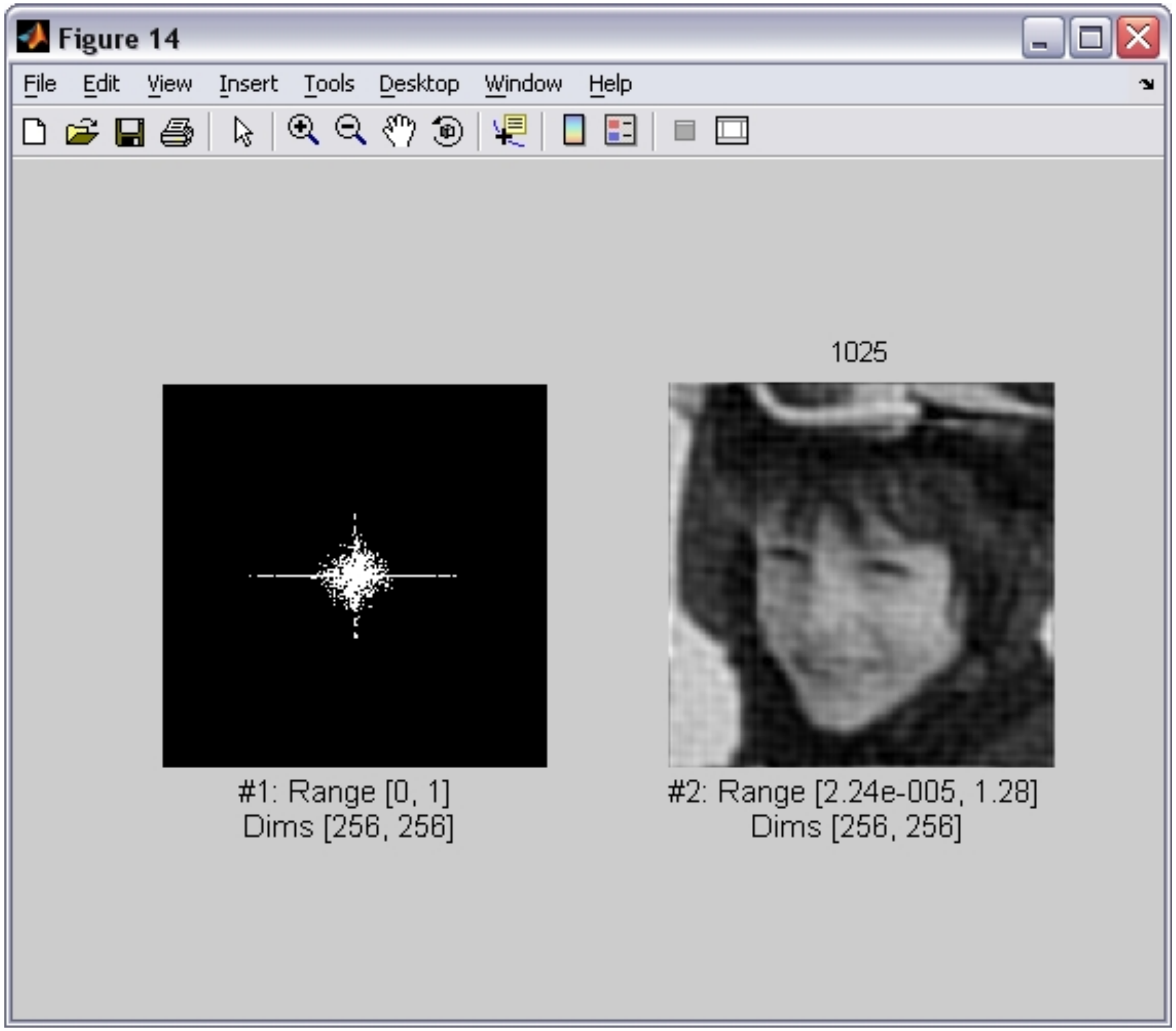


Figure 14
File Edit View Insert Tools Desktop Window Help
1025
#1: Range [0, 1]
Dims [256, 256]
#2: Range [2.24e-005, 1.28]
Dims [256, 256]

2049

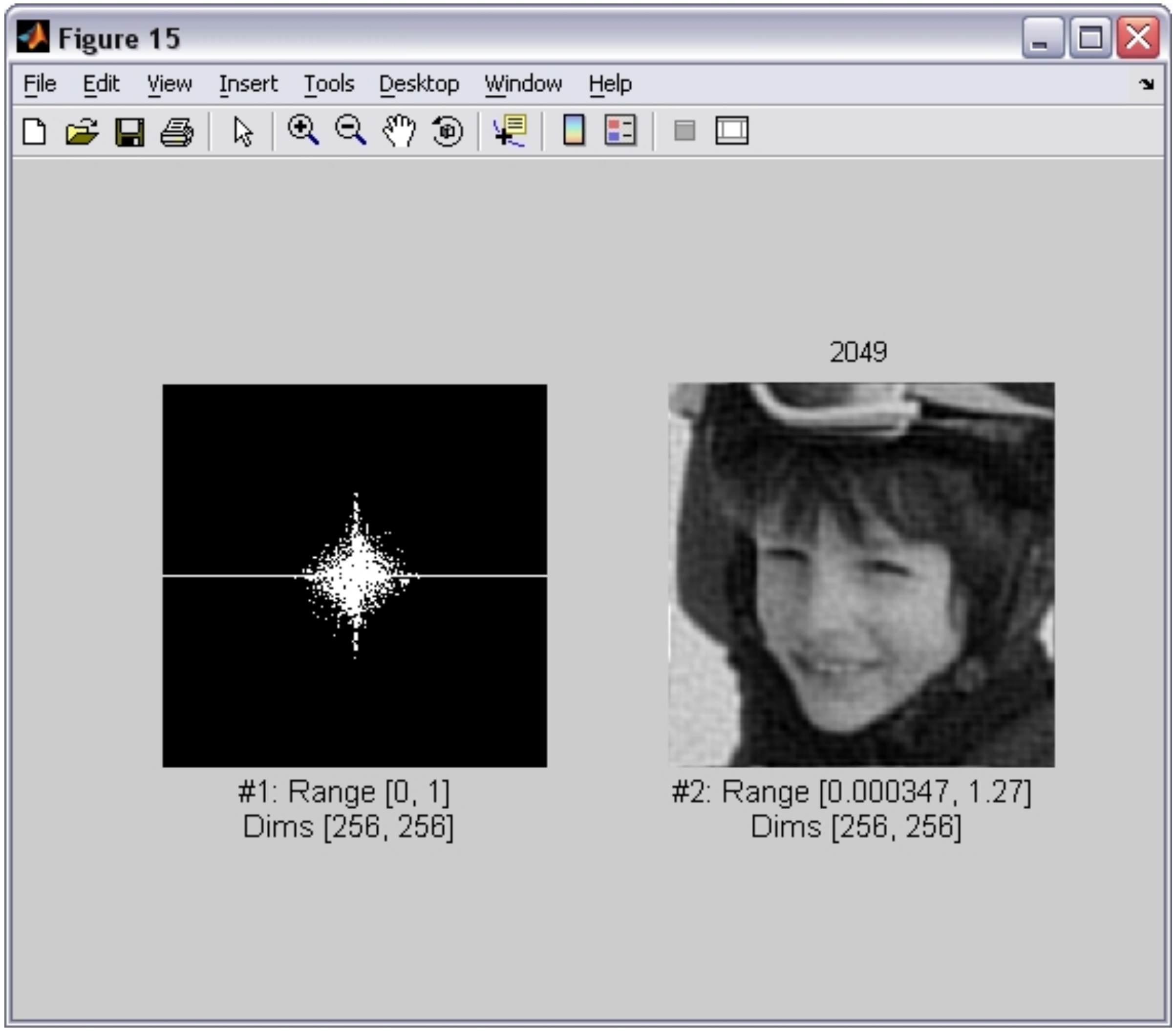


Figure 15
File Edit View Insert Tools Desktop Window Help
2049
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000347, 1.27]
Dims [256, 256]

4097

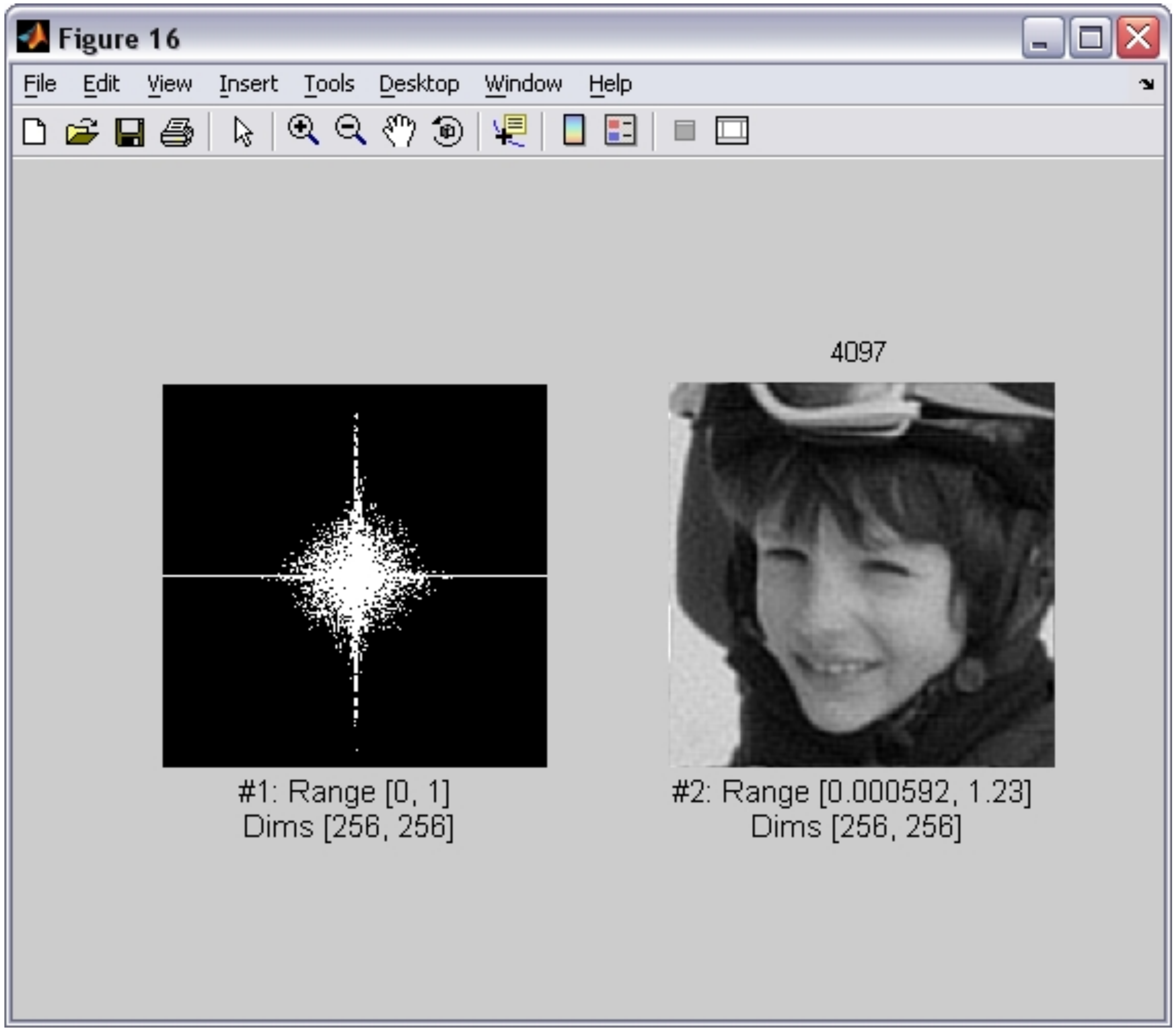


Figure 16
File Edit View Insert Tools Desktop Window Help
4097
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000592, 1.23]
Dims [256, 256]

8193

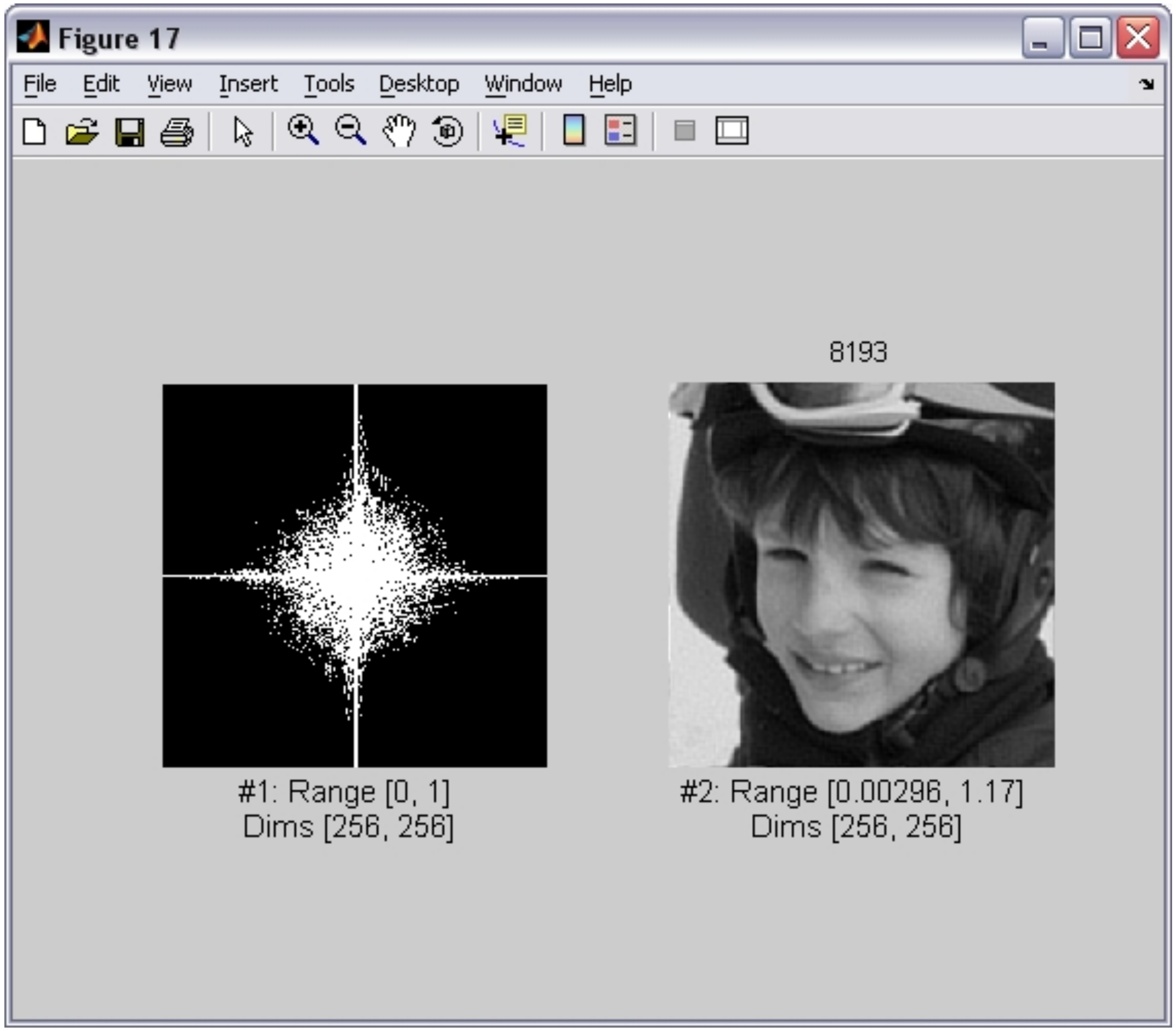


Figure 17
File Edit View Insert Tools Desktop Window Help
8193
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.00296, 1.17]
Dims [256, 256]

16385

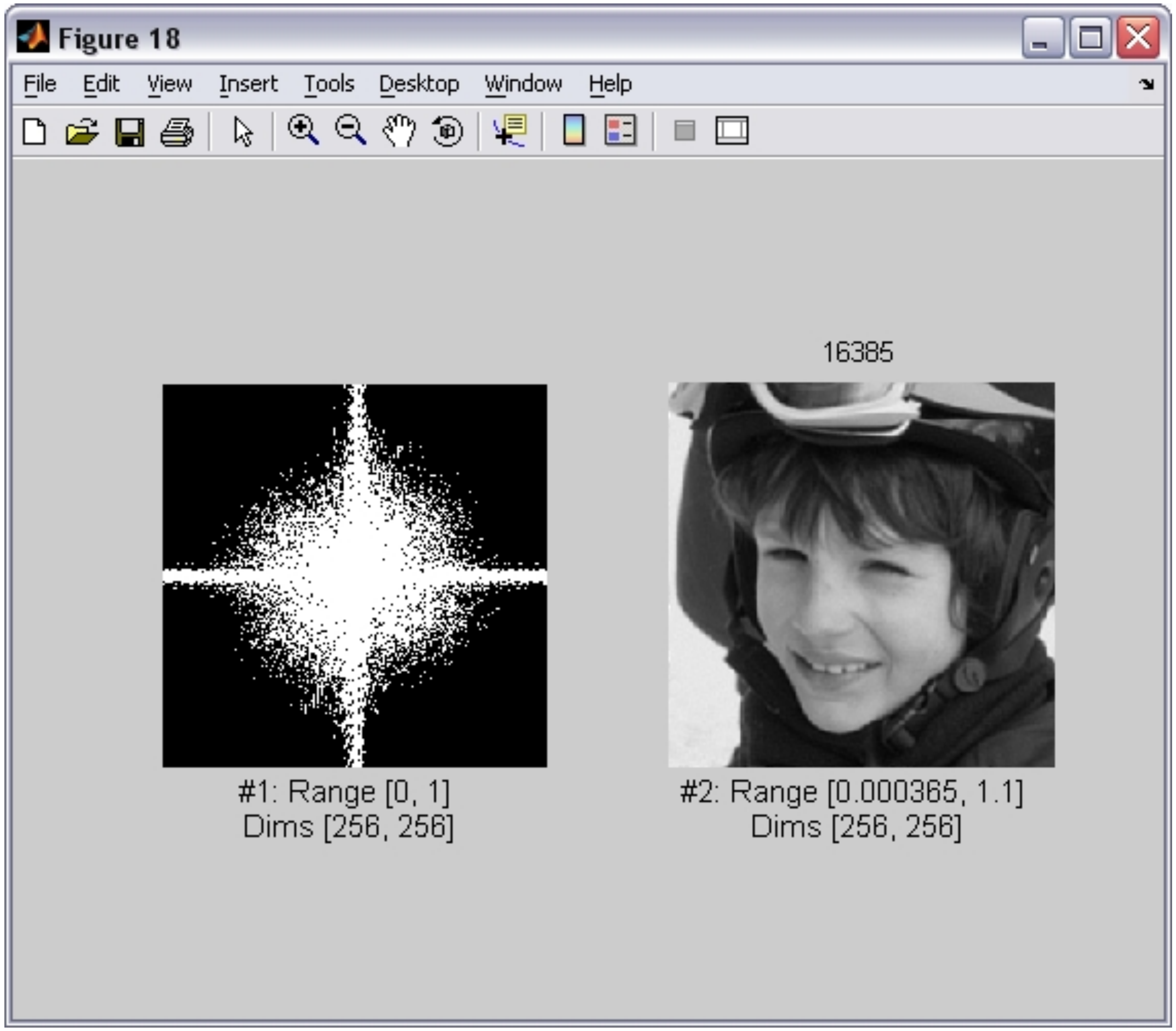


Figure 18
File
Edit
View
Insert
Tools
Desktop
Window
Help
16385
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.000365, 1.1]
Dims [256, 256]

32769

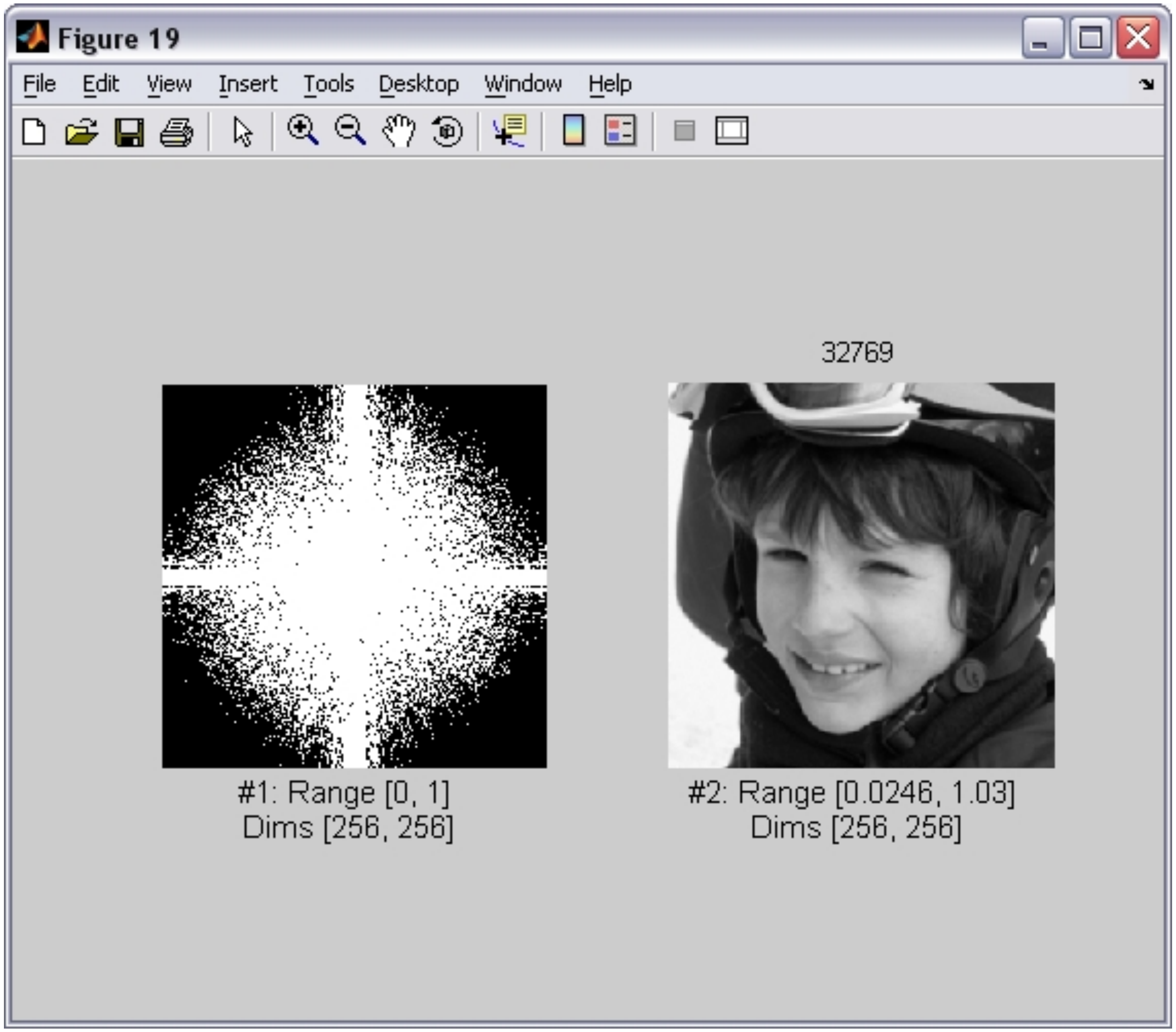


Figure 19
File Edit View Insert Tools Desktop Window Help
32769
#1: Range [0, 1]
Dims [256, 256]
#2: Range [0.0246, 1.03]
Dims [256, 256]

65536

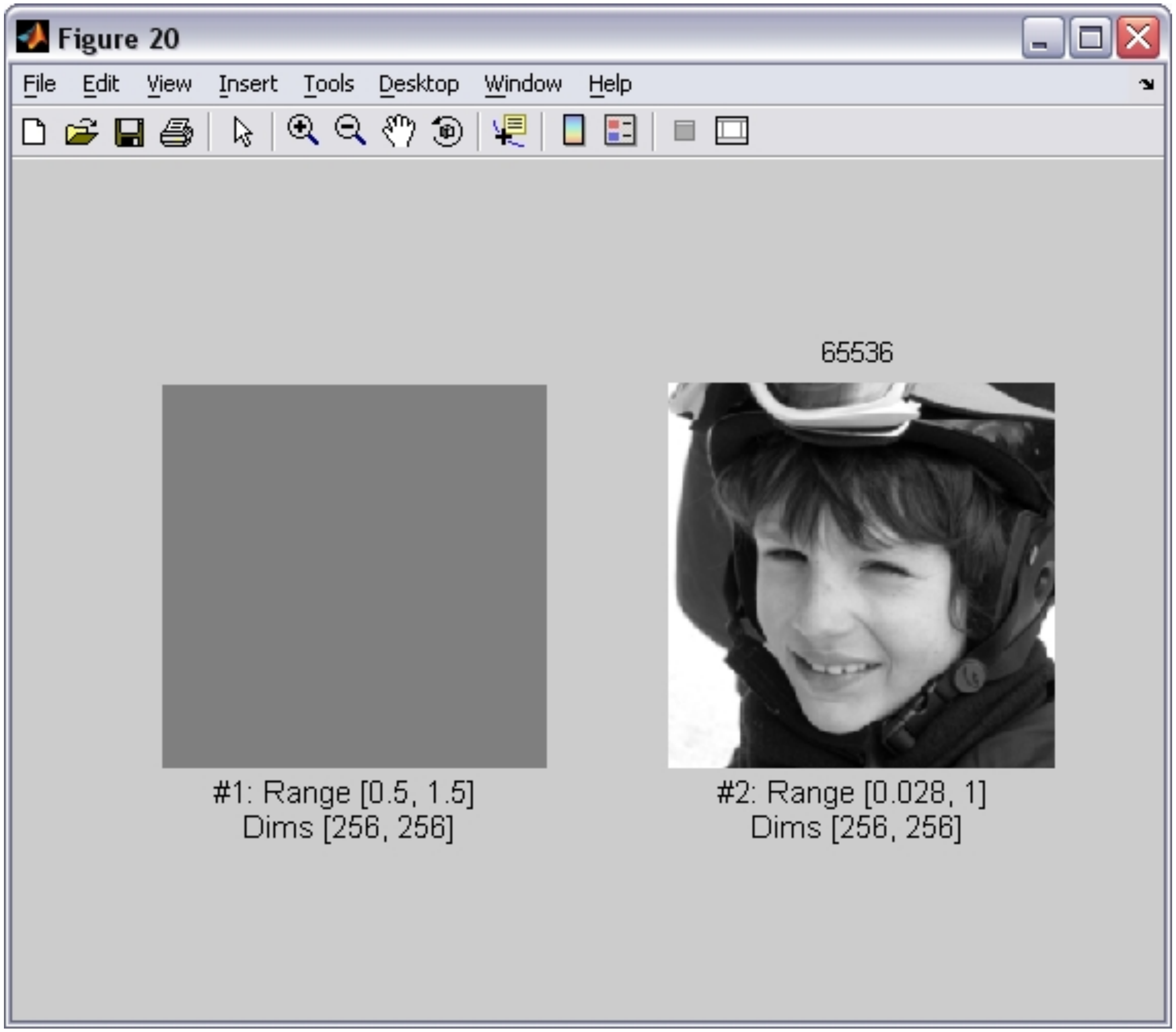
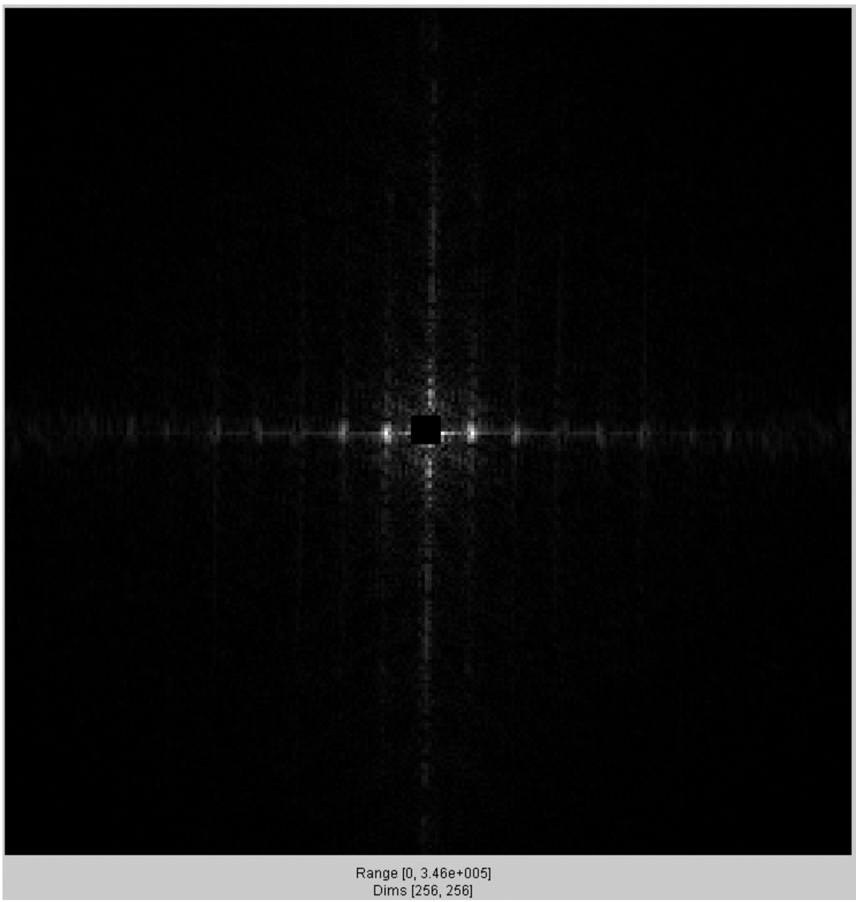
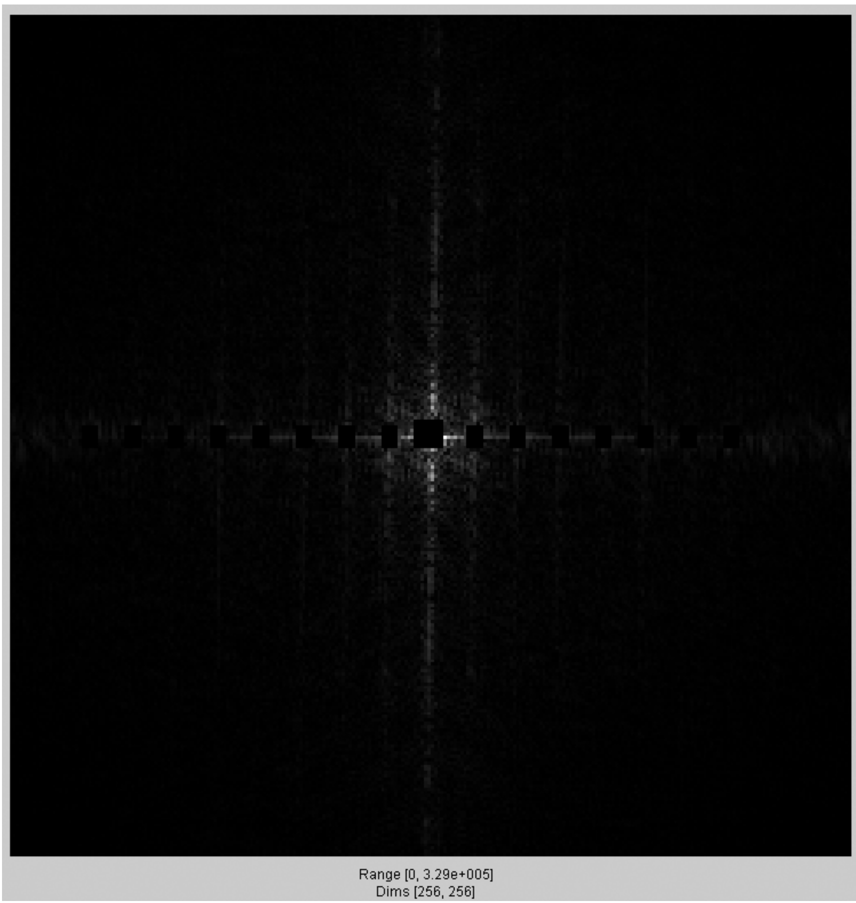


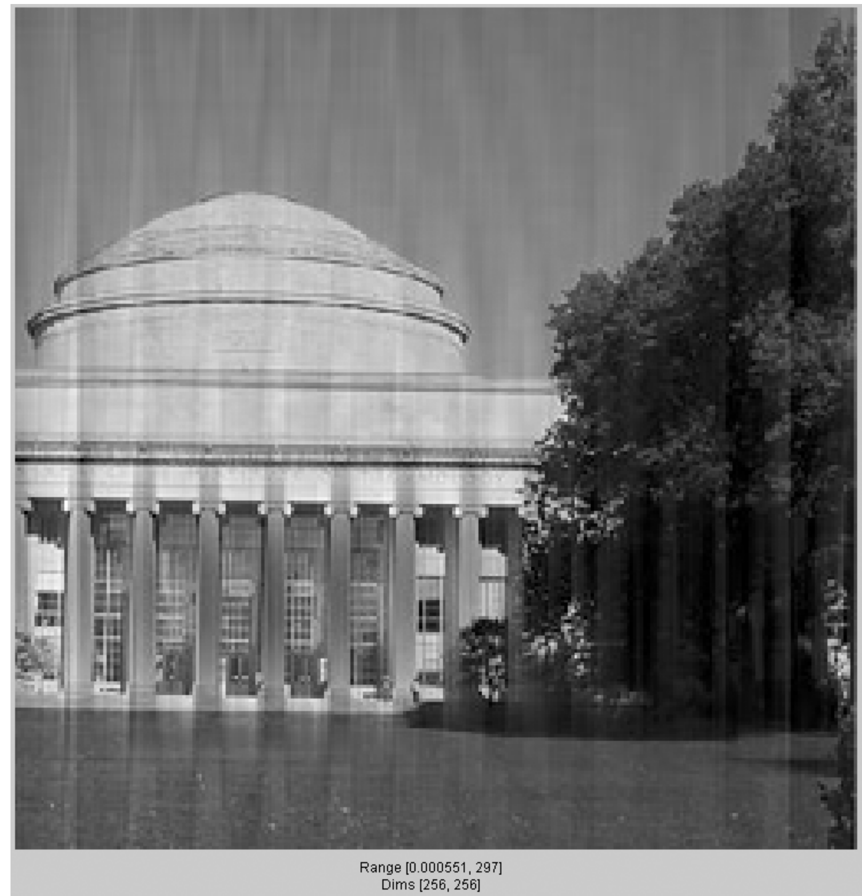
Figure 20
File Edit View Insert Tools Desktop Window Help
65536
#1: Range [0.5, 1.5]
Dims [256, 256]
#2: Range [0.028, 1]
Dims [256, 256]

Fourier transform magnitude



Masking out the fundamental and harmonics from periodic pillars





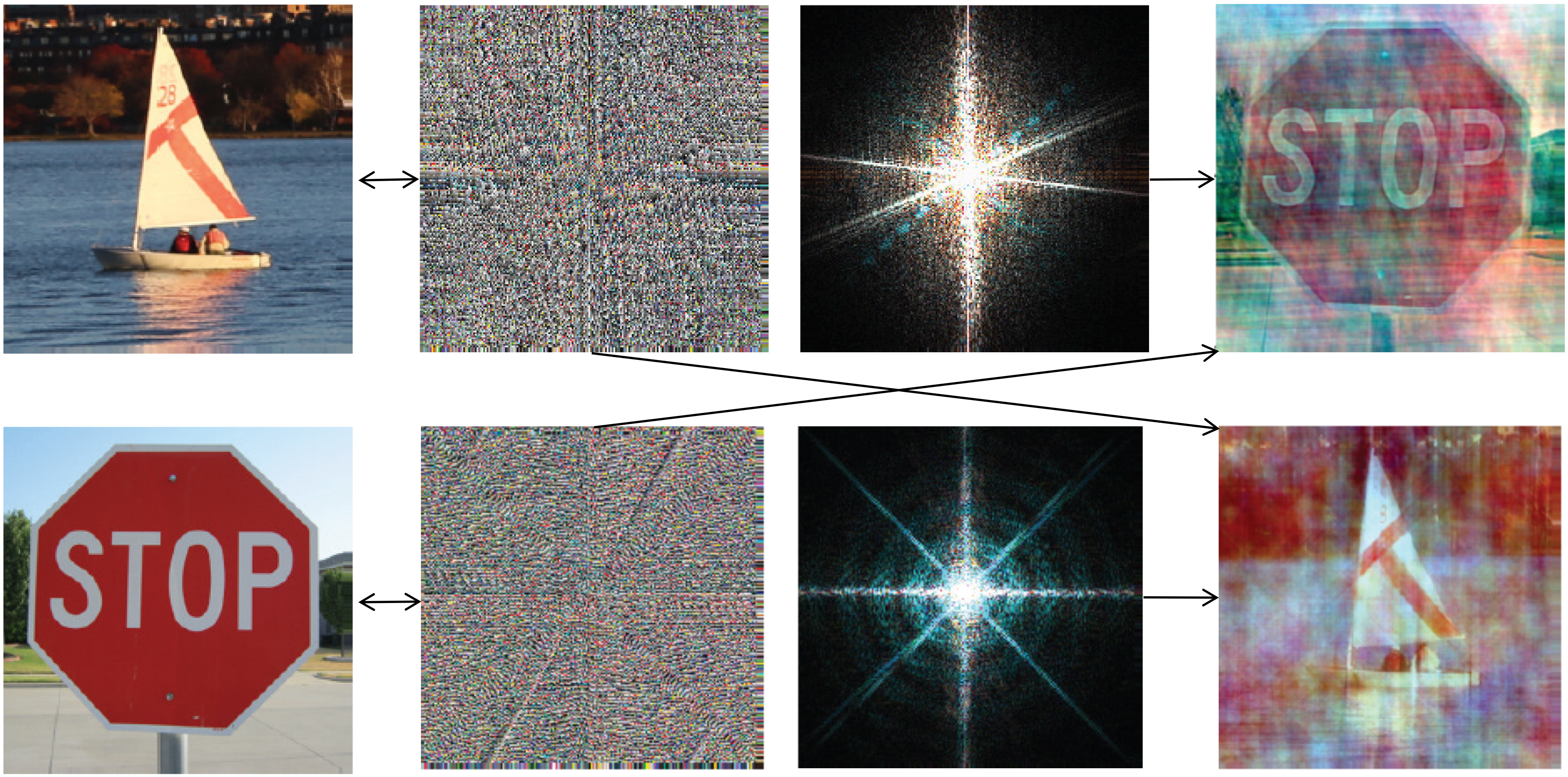
Phase and Magnitude

- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn't

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

Phase and magnitude

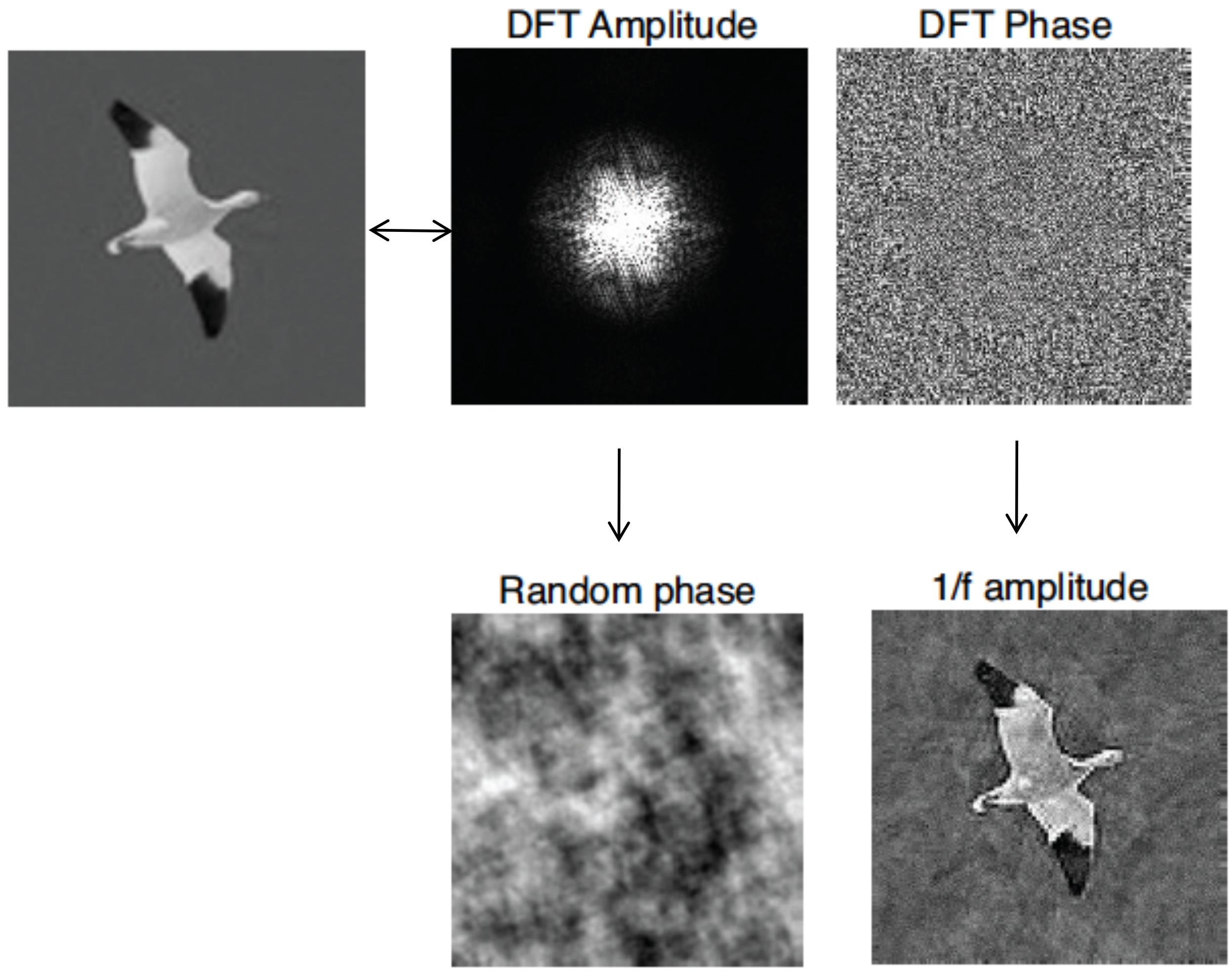
$$F[u,v] = A[u,v]\exp(j\theta[u,v])$$



Each color channel is processed in the same way.

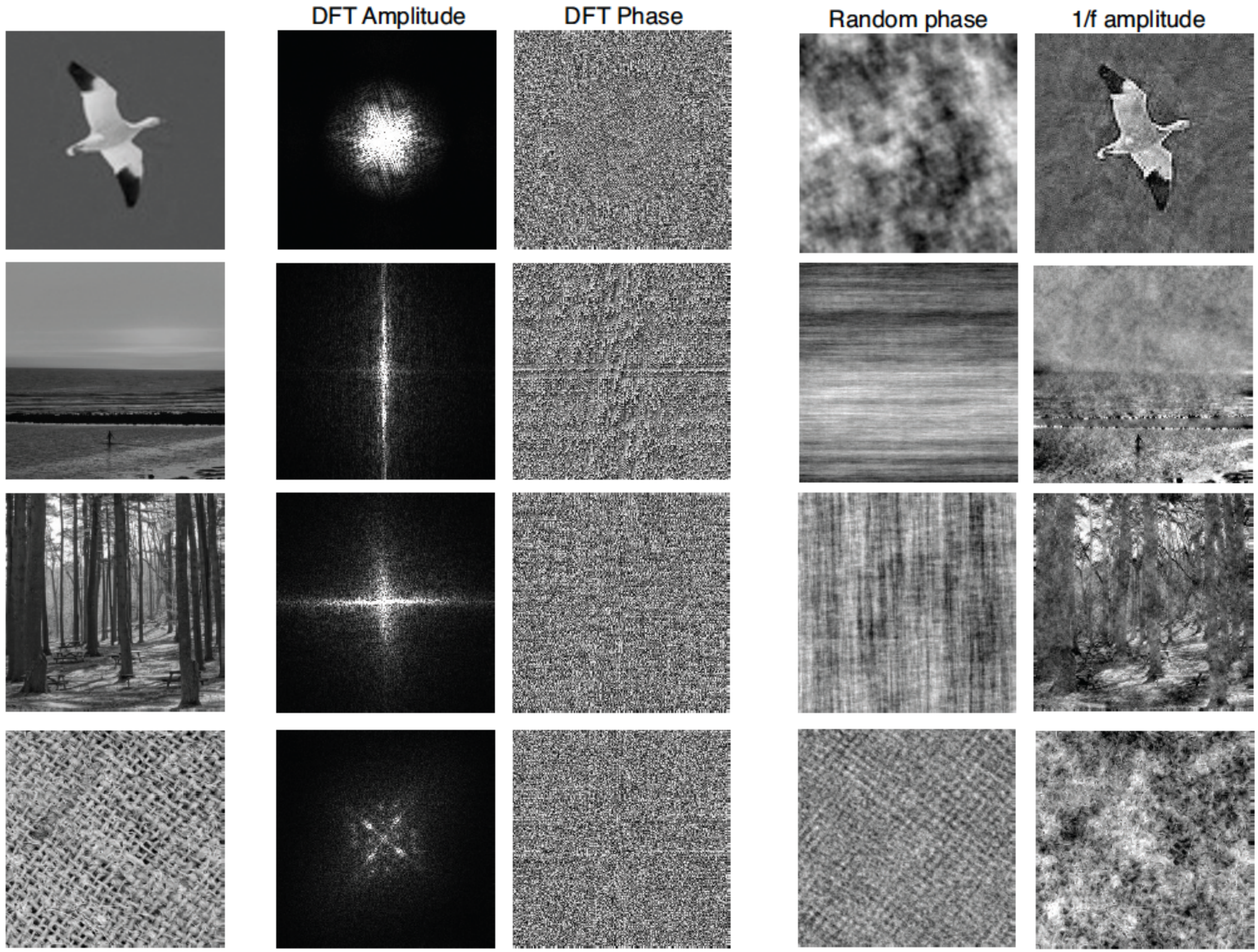
Phase and magnitude

$$F[u,v] = A[u,v] \exp(j\theta[u,v])$$



Using random amplitude does not look good.

Does phase always win?



Randomizing the phase