MIT CSAIL

6.869: Advances in Computer Vision

Antonio Torralba and Bill Freeman, 2017

MIT COMPUTER VISION

Lecture 2

Linear filters

Class web page

http://6.869.csail.mit.edu/fa17/

About me…

Bill Freeman

Thomas and Gerd Perkins Professor,

Electrical Engineering and Computer Science

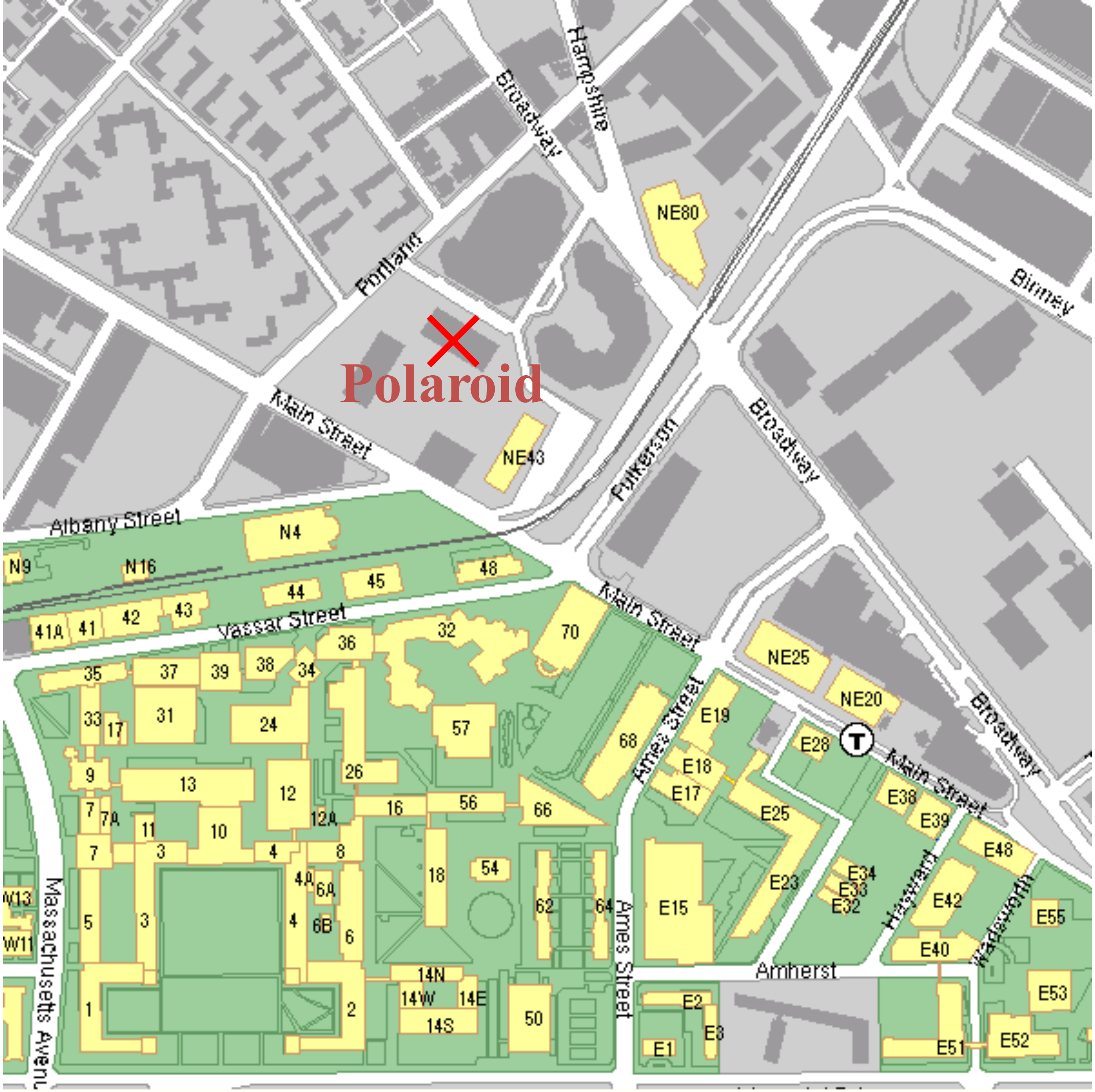
MIT CSAIL

Stata Center, 32-D476

Cambridge, MA 02139

e-mail: billf@mit.edu

web: http://www.ai.mit.edu/people/wtf


Polaroid
Broadway
Hampshire
Portland
NE80
Binney
Main Street
NE43
Fulkerson
Broadway
Albany Street
N4
N9
N16
48
44
45
41A
41
42
43
Vassar Street
36
32
70
Main Street
NE25
NE20
35
37
39
38
34
33
17
31
24
57
68
Ames Street
E19
E28
Main Street
Broadway
9
13
12
26
E18
E17
E38
E39
7
7A
11
10
12A
16
56
66
E25
E48
7
3
4
8
18
54
4A
6A
E34
E33
E32
E23
Hayward
E42
Wadsworth
E55
W13
5
3
4
6B
6
62
64
E15
E40
Massachusetts Avenue
W11
14N
Amherst
1
2
14W
14E
50
Ames Street
E2
E53
14S
E3
E1
E51
E52

Polaroid work

solutions for my

explore solutions with the power to...

product catalo

- Locate a Dealer
- View FAQs
- Sales Contact
- Buy an Extended Warranty

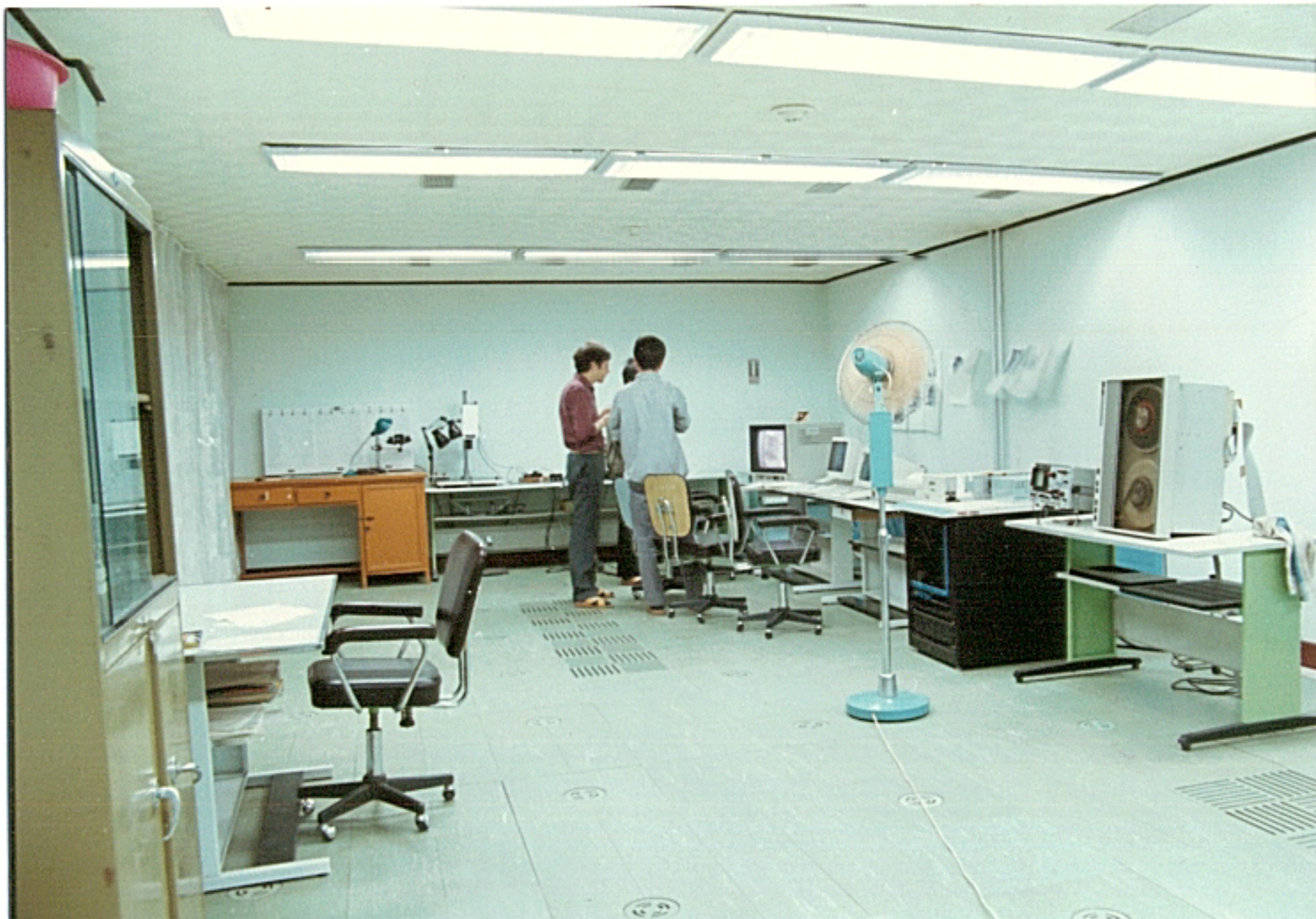
ProPalette 8000 Digital Film Recorder


Polaroid
Farbbilder
vom Bildschirm
Instant Color
Video Hard Copy
Polaroid
RD GR BL GY YE VT BG ET

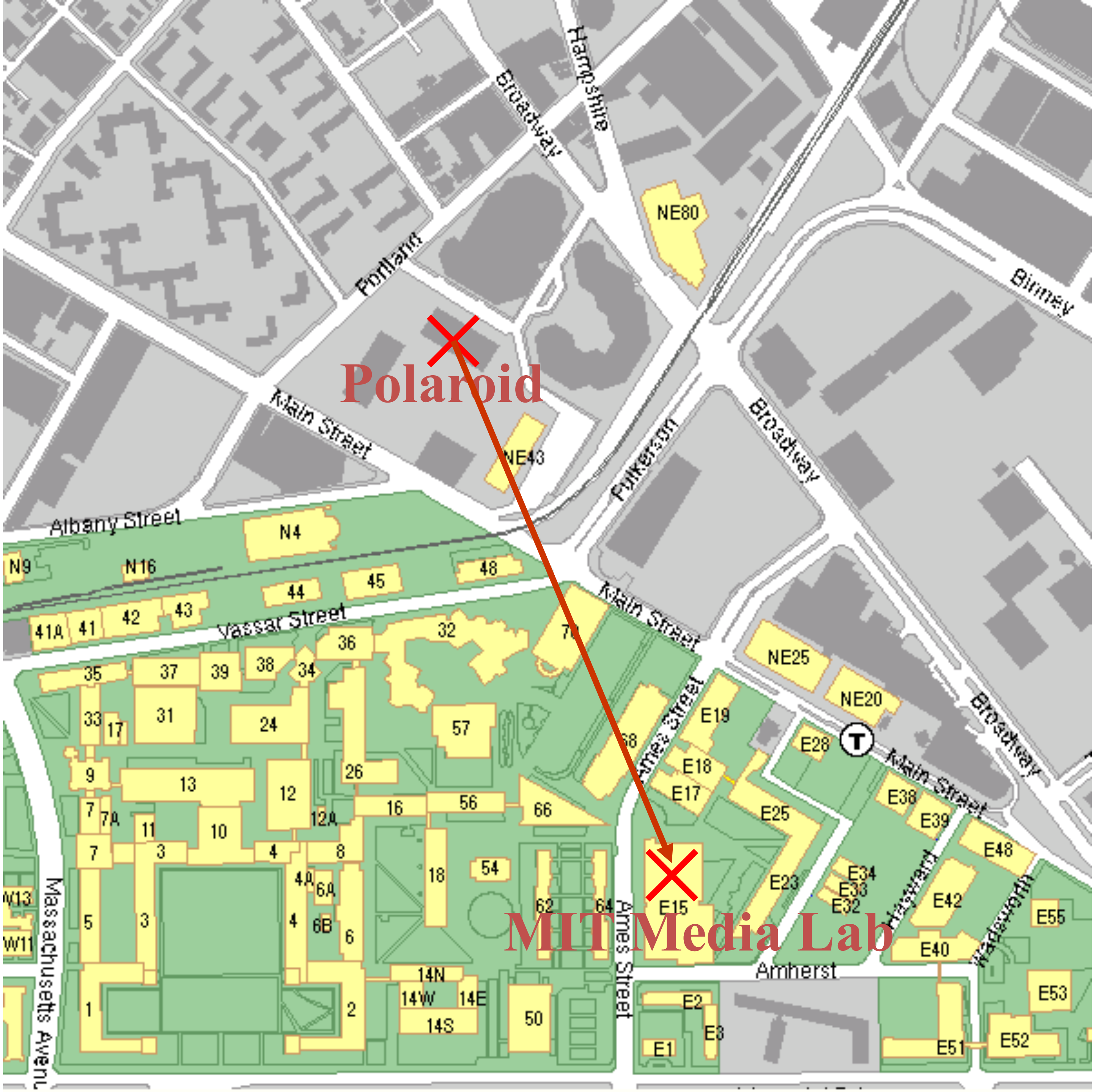
PRODUCT *catalog*

Taiyuan, China, 1987

In my office at the Computer Center, 1987




计算中心
COMPUTER CENTER


Polaroid
MIT Media Lab
Main Street
Broadway
Hampshire
Portland
Binney
Fulkerson
Albany Street
Vassar Street
Ames Street
Amherst
Hayward
Wadsworth
Massachusetts Avenue
NE80
NE43
NE25
NE20
N4
N9
N16
48
45
44
41A
41
42
43
36
32
70
35
37
39
38
34
33
17
31
24
57
68
E19
E18
E17
E28
E25
E38
E39
E48
E15
E23
E34
E33
E32
E42
E55
E40
E2
E3
E1
E53
E51
E52
9
13
12
26
16
56
66
7
7A
11
10
12A
3
4
8
18
54
62
64
4A
6A
5
4
6B
6
W13
W11
14N
14W
14E
14S
50
1
2

Steerable Filter Architecture

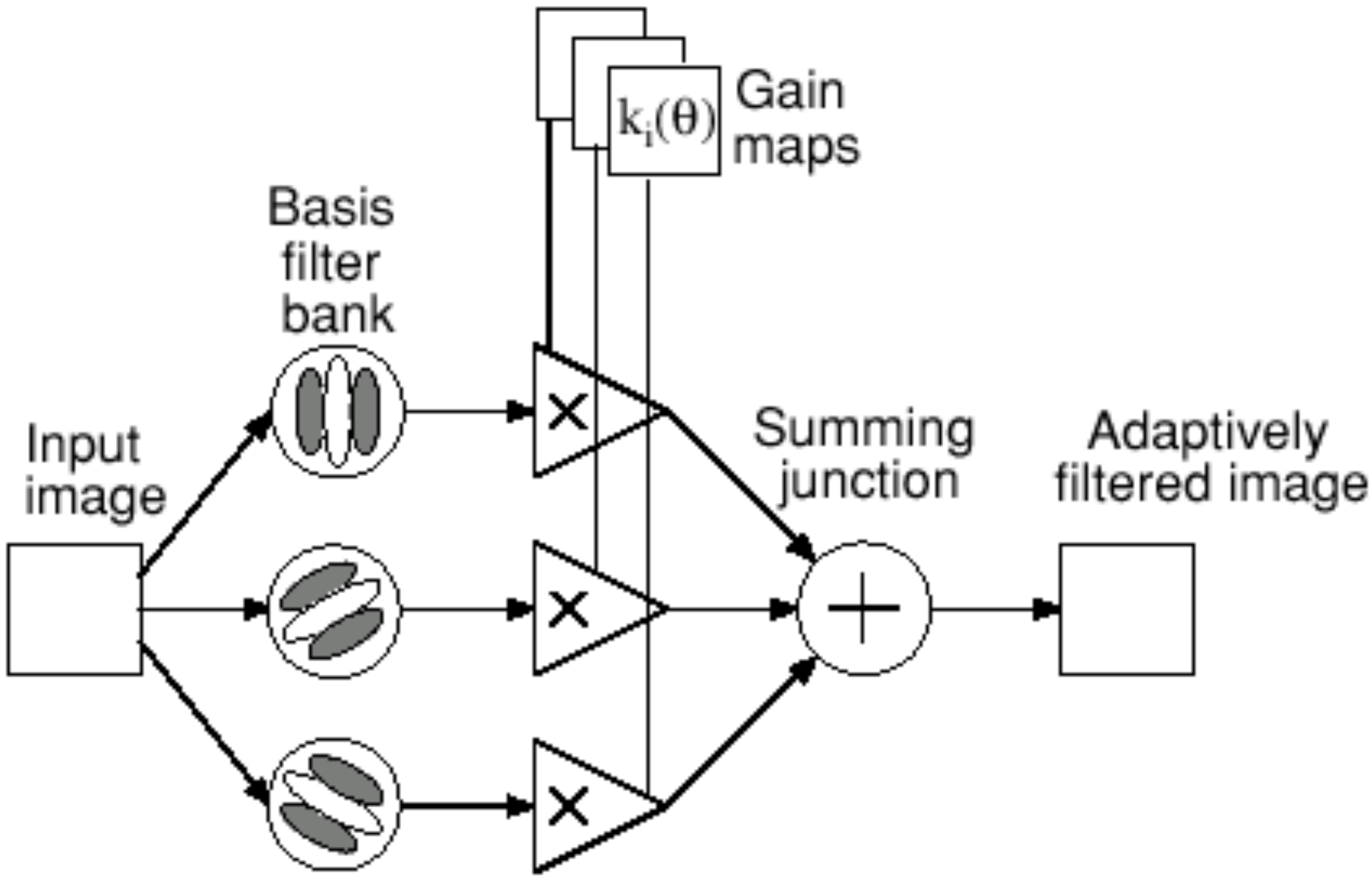
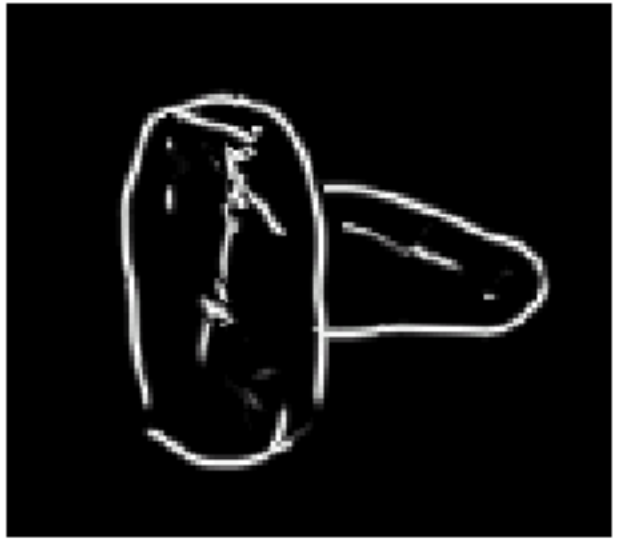
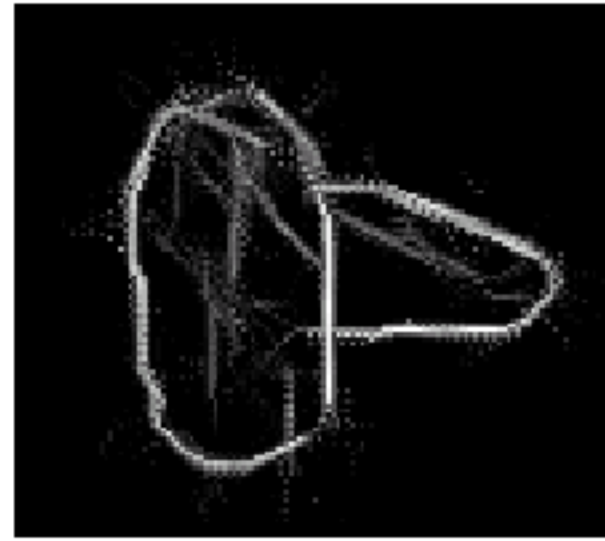


Image interpretation from local cues

Image

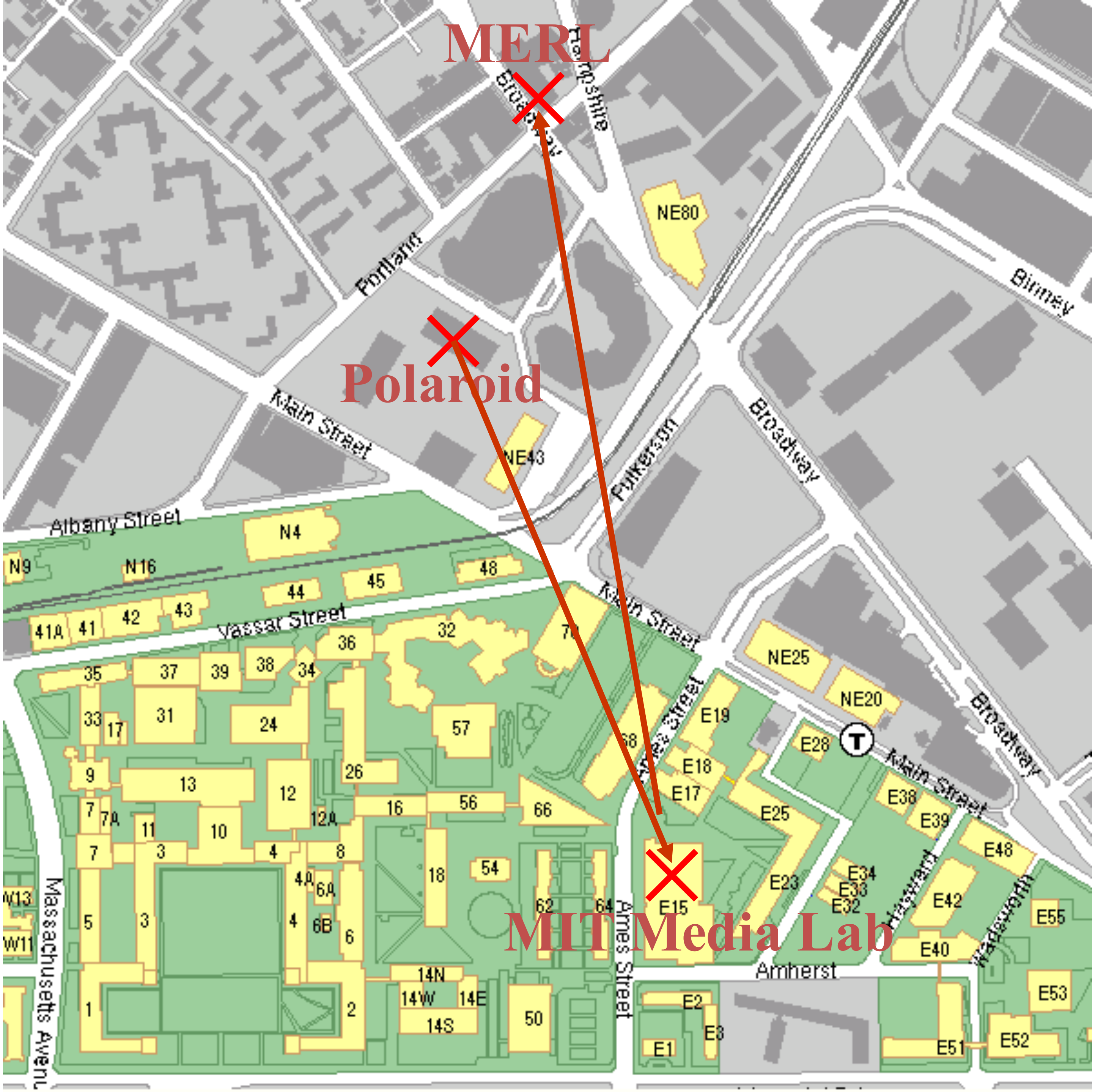


Oriented energy

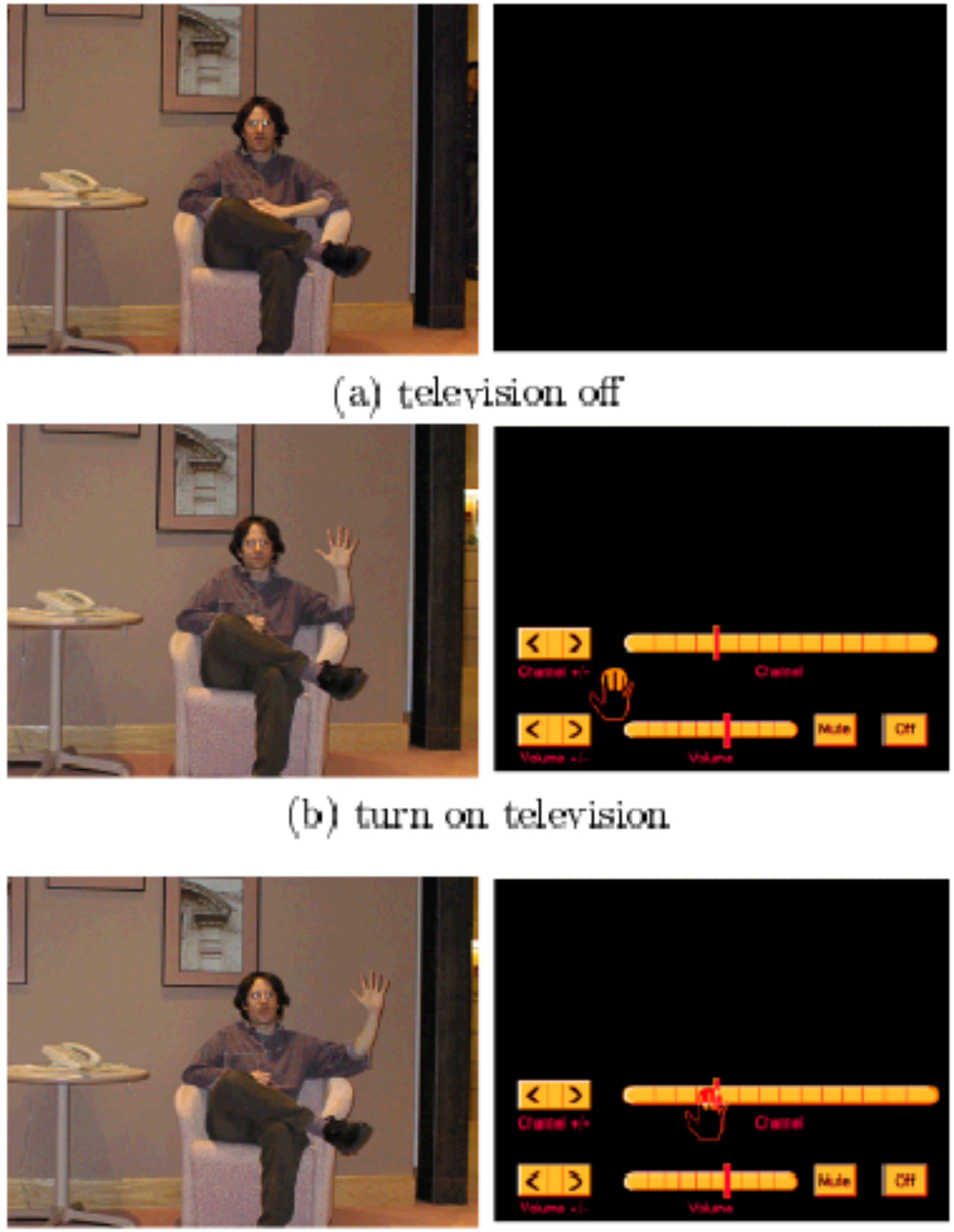


Contours

Occluding contour


MERL
Polaroid
MIT Media Lab
Portland
Main Street
Albany Street
Vassar Street
Fulkerson
Broadway
Binney
Ames Street
Amherst
Massachusetts Avenu
Hayward
Wadsworth
NE80
NE43
NE25
NE20
N4
N9
N16
48
44
45
41A
41
42
43
32
36
70
35
37
39
38
34
33
17
31
24
57
68
E19
E28
E18
E17
9
13
12
26
16
56
66
7
7A
11
10
12A
3
4
8
E25
E38
E39
E48
E34
E33
E32
E23
E42
E55
E15
18
54
62
64
4A
6A
6B
6
5
W13
W11
E40
14N
14W
14E
14S
50
1
2
E2
E3
E1
E51
E52
E53

camera input television overlay



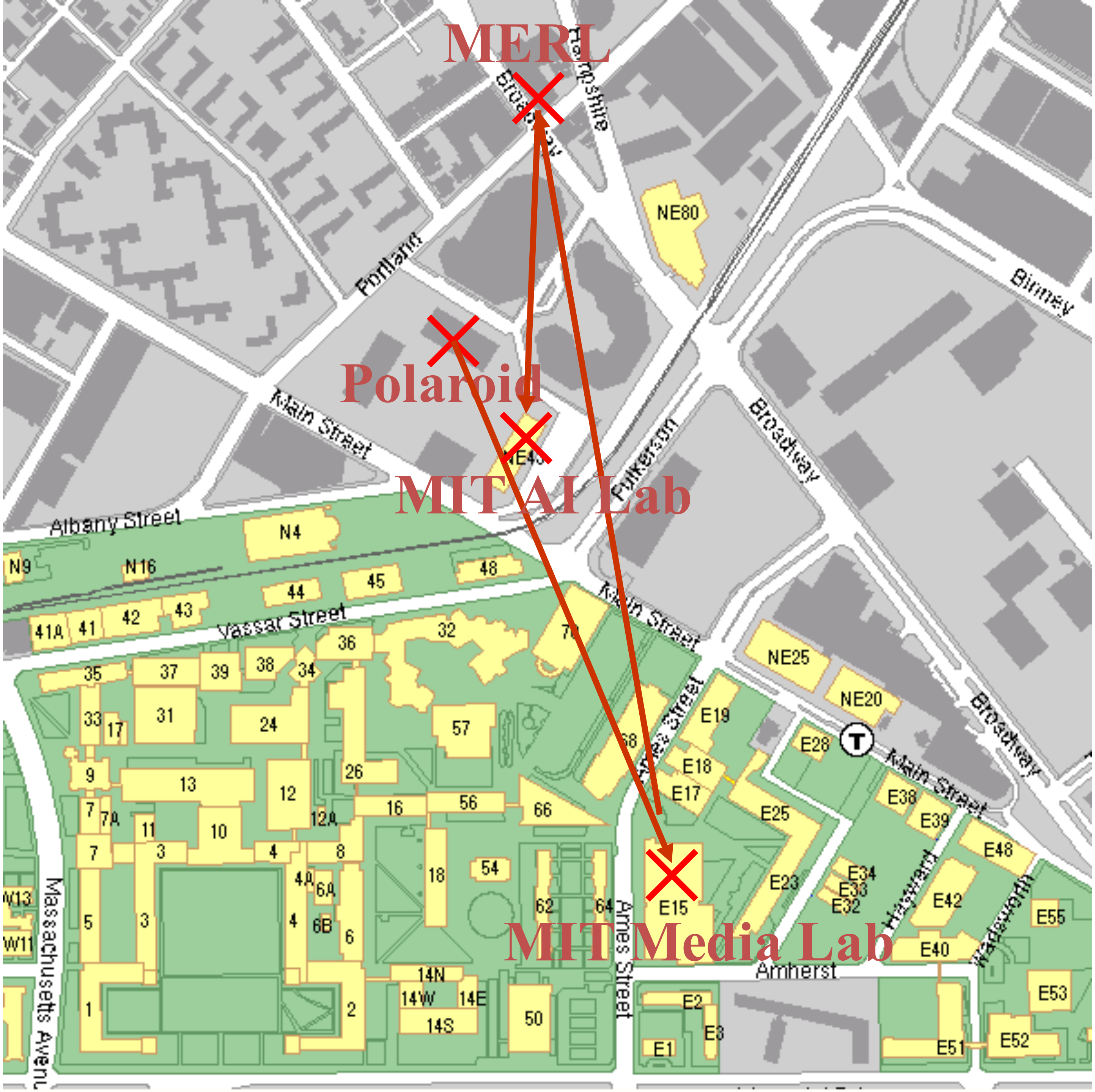
(a) television off

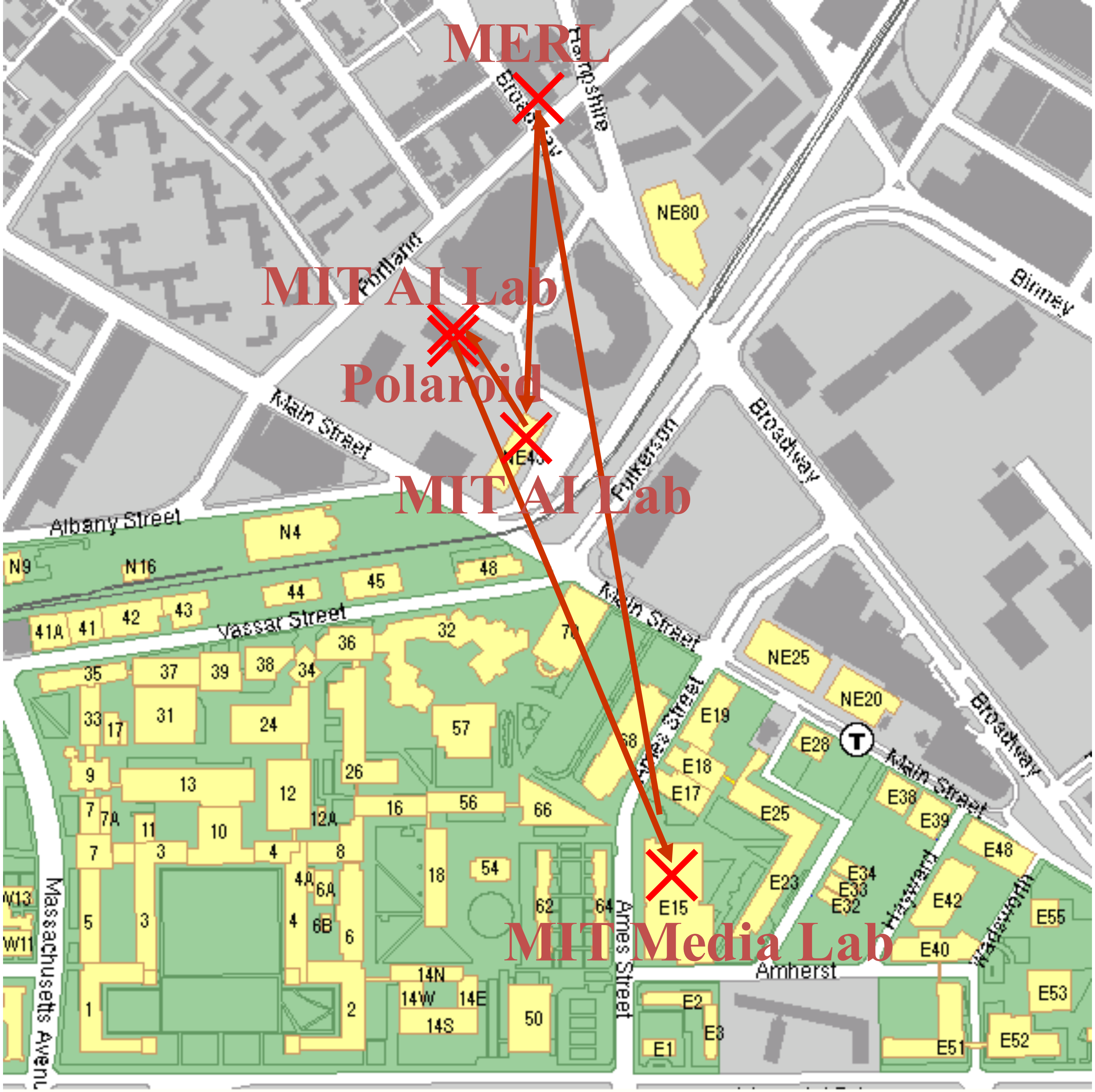
(b) turn on television

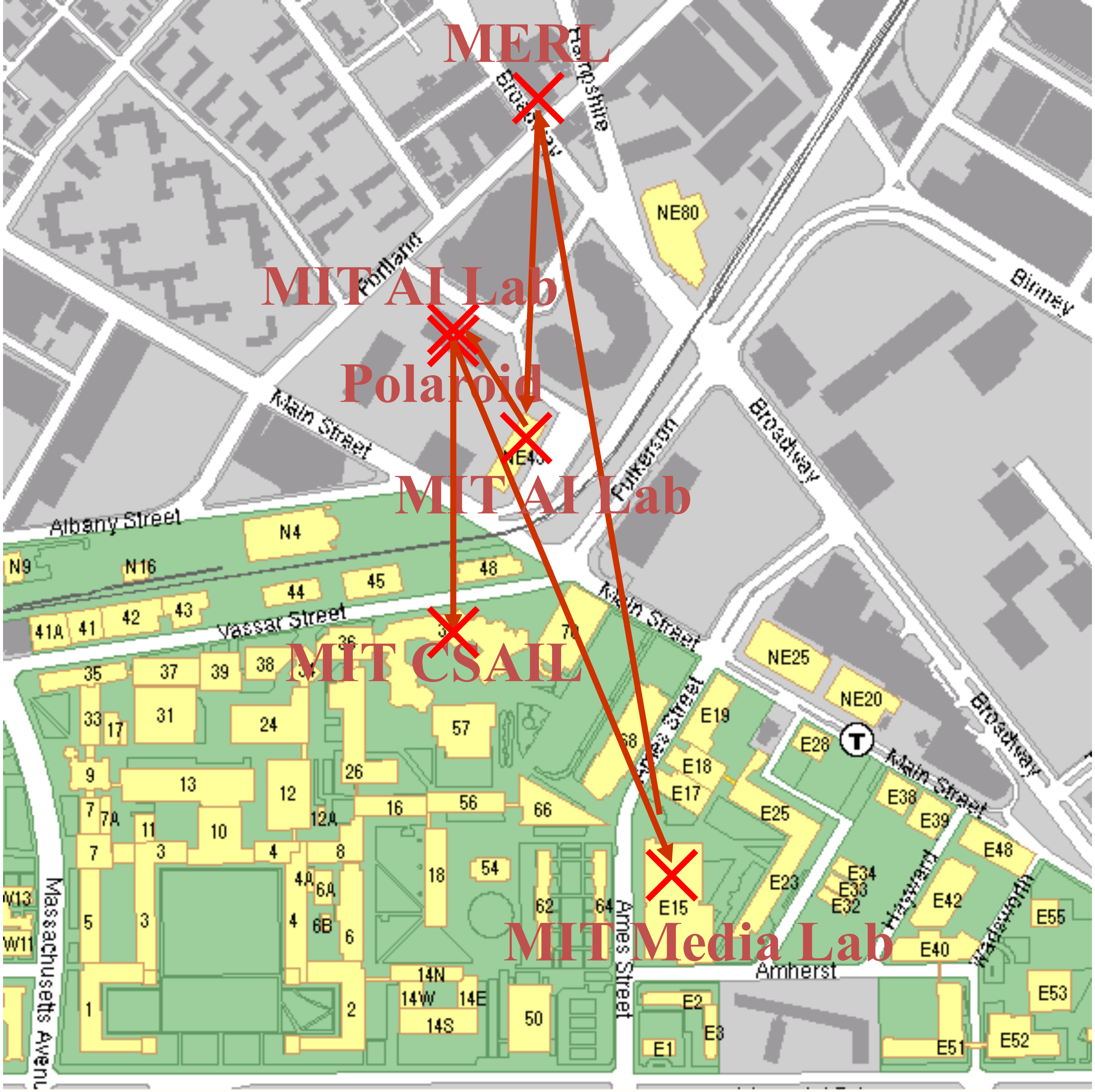
(c) channel control

Comdex 1994

Decathlete 100m hurdles

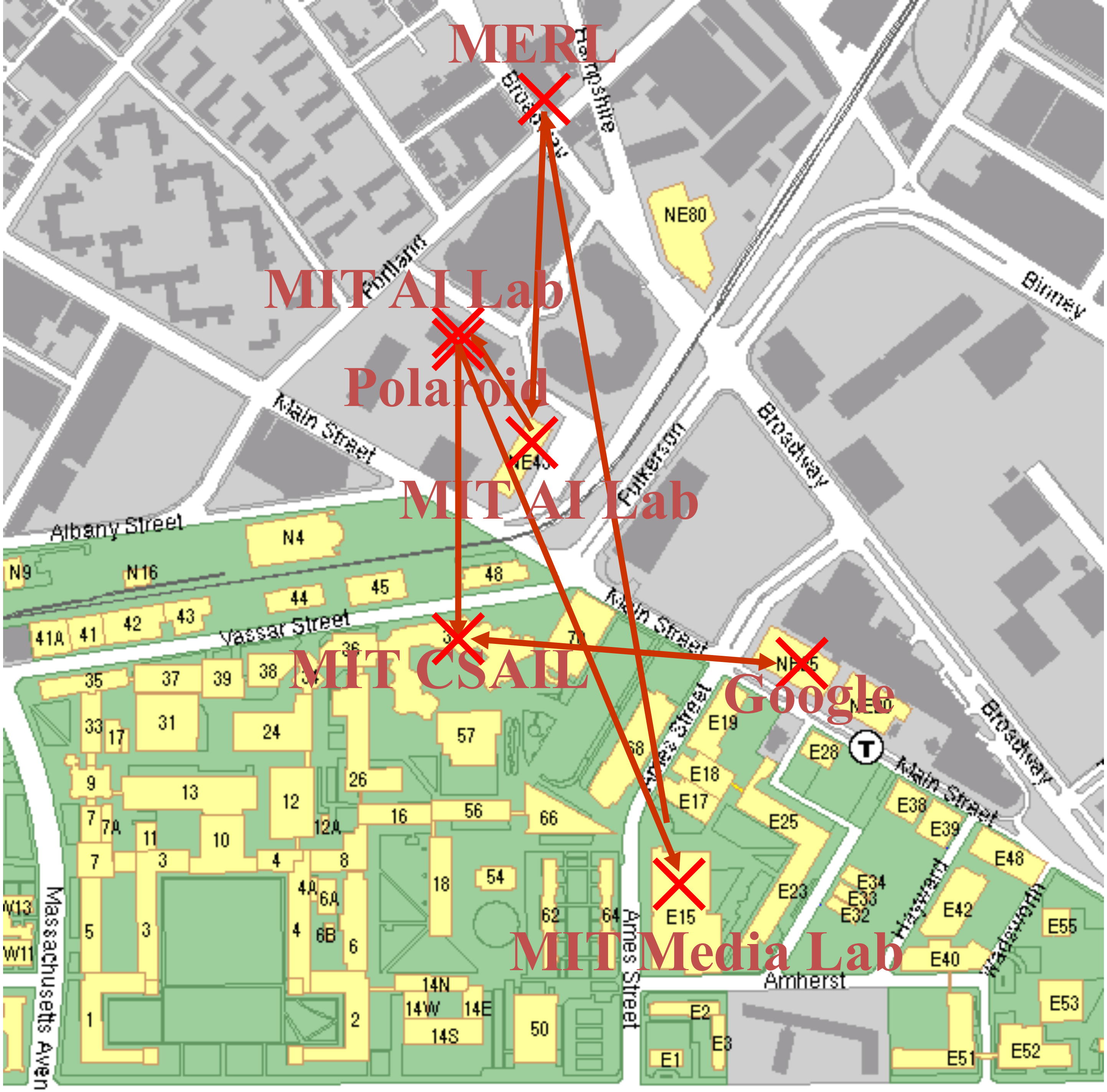

MERL
Polaroid
MIT AI Lab
MIT Media Lab
Portland
Hampshire
Broadway
Main Street
Albany Street
Vassar Street
Fulkerson
Binney
Ames Street
Amherst
Hayward
Wadsworth
Massachusetts Avenue
NE80
NE43
NE25
NE20
N4
N9
N16
41A
41
42
43
44
45
48
32
70
36
35
37
39
38
34
33
17
31
24
57
68
E19
E28
E18
E17
E25
E38
E39
E48
9
13
12
26
16
56
66
7
7A
11
10
12A
3
4
8
18
54
62
64
E15
E23
E34
E33
E32
E42
E55
W13
W11
5
4A
6A
6B
6
14N
14W
14E
14S
1
2
50
E2
E3
E1
E40
E53
E51
E52


MERL
MIT AI Lab
Polaroid
MIT AI Lab
MIT Media Lab
Broadway
Hampshire
Portland
Binney
Main Street
Fulkerson
Albany Street
Vassar Street
Ames Street
Amherst
Hayward
Wadsworth
Massachusetts Avenu
NE80
NE43
NE25
NE20
N4
N9
N16
48
44
45
41A
41
42
43
36
32
70
35
37
39
38
34
33
17
31
24
57
68
E19
E28
E18
E17
E25
E38
E39
E48
9
13
12
26
16
56
66
7
7A
11
10
12A
3
4
8
E23
E34
E33
E32
E42
E55
W13
W11
5
4A
6A
18
54
62
64
E15
6B
6
E40
14N
14W
14E
14S
1
2
50
E2
E3
E1
E53
E52
E51

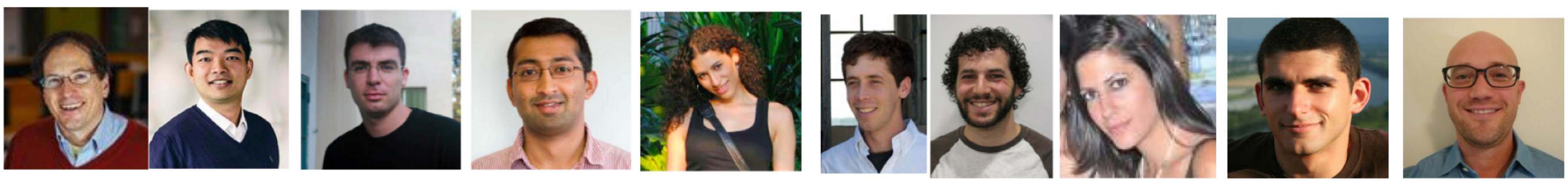


MERL
MIT AI Lab
Polaroid
MIT AI Lab
MIT CSAIL
MIT Media Lab
Broadway
Hampshire
Portland
Main Street
Albany Street
Vassar Street
Fulkerson
Binney
Ames Street
Amherst
Hayward
Wadsworth
Massachusetts Avenue
NE80
NE43
N4
N9
N16
41A
41
42
43
44
45
48
36
70
NE25
NE20
E19
E28
E18
E17
E25
E38
E39
E48
E34
E33
E32
E23
E42
E55
E15
E40
E53
E52
E51
E2
E3
E1
35
37
39
38
33
17
31
24
57
68
9
13
12
26
7
7A
11
10
12A
16
56
66
3
4
8
18
54
62
64
4A
6A
6B
6
5
1
2
14N
14W
14E
14S
50
W13
W11


IVORY


MERL
MIT AI Lab
Polaroid
MIT AI Lab
MIT CSAIL
Google
MIT Media Lab
Portland
Main Street
Albany Street
Vassar Street
Fulkerson
Broadway
Binney
Amherst
Ames Street
Hayward
Massachusetts Avenu
NE80
NE43
N4
N9
N16
48
45
44
43
42
41
41A
NE25
E19
E28
E18
E17
E25
E15
E23
E38
E39
E48
E42
E40
E55
E53
E52
E51
E2
E3
E1
50

Google Cambridge Vision Team



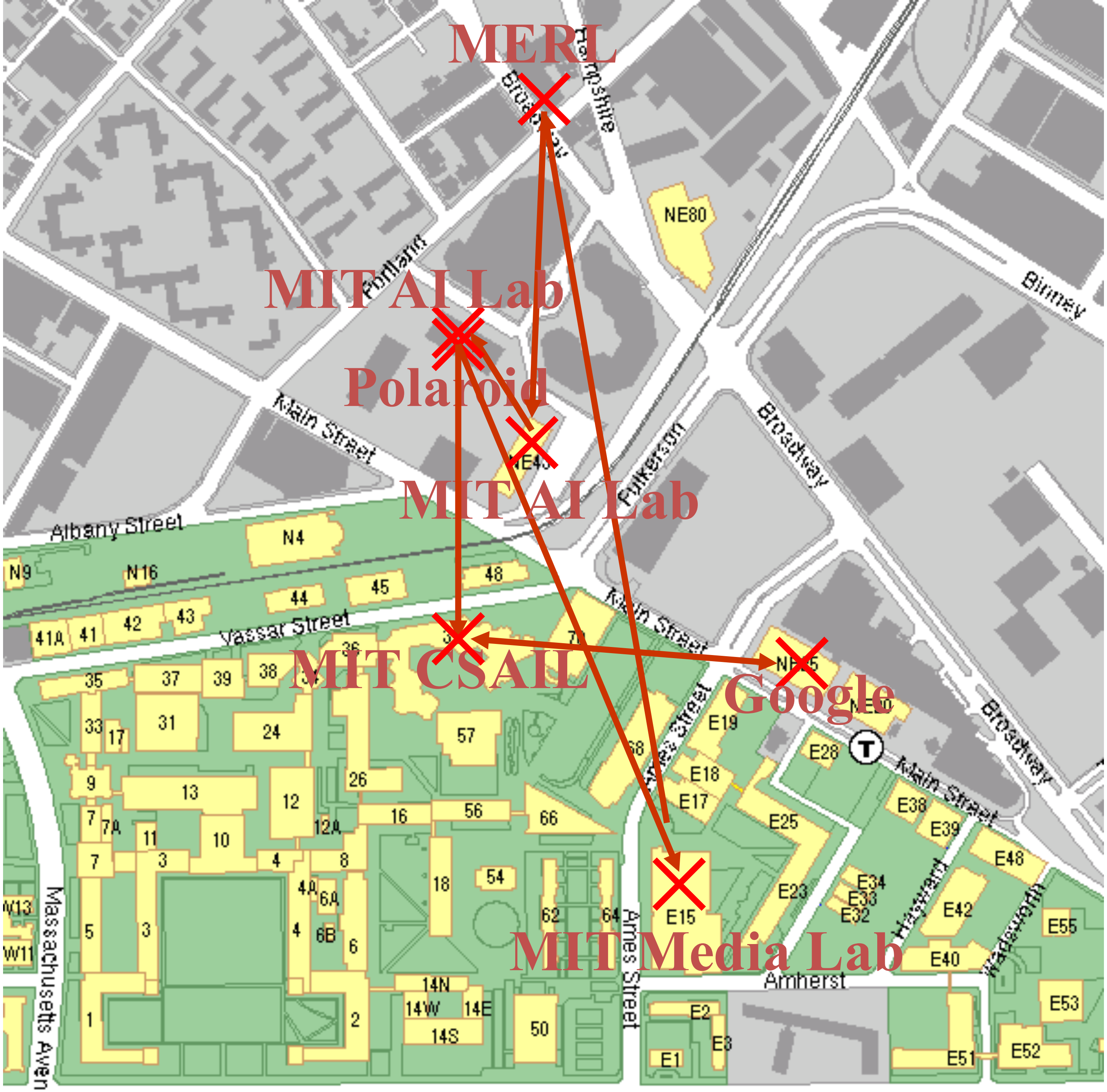
team members:
Bill Freeman, Ce Liu, Miki Rubinstein, Dilip Krishnan, Inbar Mosseri, Forrester Cole, Aaron Sarna, Tali Dekel, Mike Krainin, Aaron Maschinot

We take summer interns!

PhotoScan

Input: move phone over the print to be digitized

Output: a cropped glare-free image

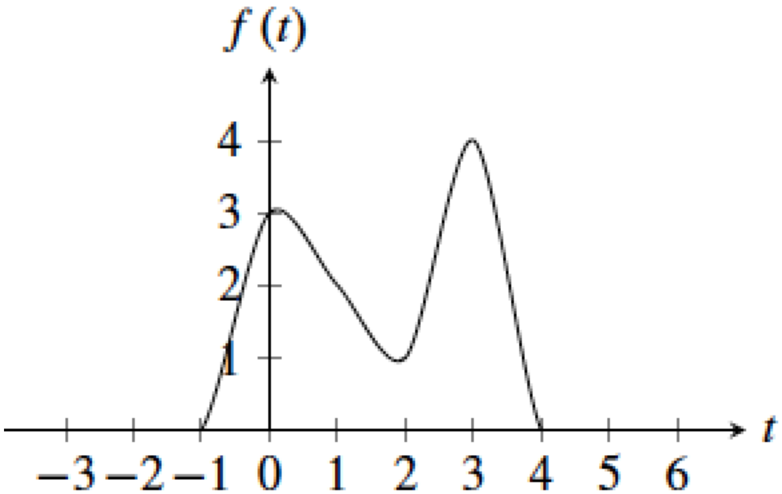

MERL
MIT AI Lab
Polaroid
MIT AI Lab
MIT CSAIL
Google
MIT Media Lab
Main Street
Albany Street
Vassar Street
Broadway
Portland
Fulkerson
Binney
Ames Street
Amherst
Massachusetts Avenue
Hayward
Wadsworth

Two offerings of a Matlab tutorial Sep. 13 & Sep. 14

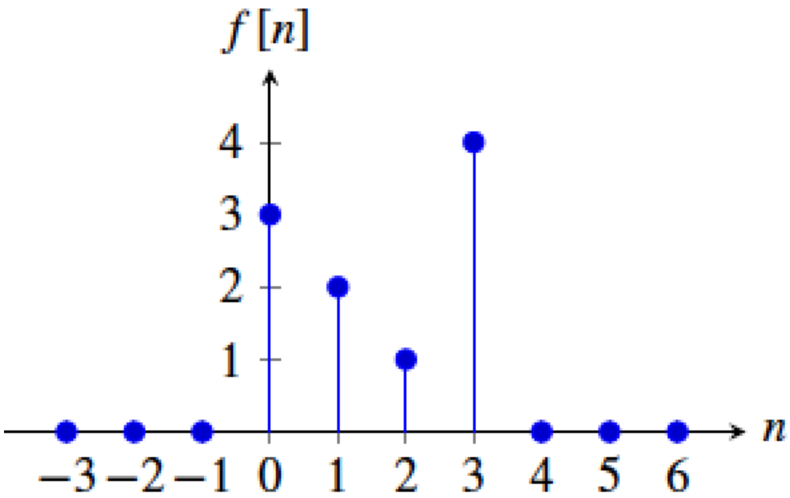
- Intended for people with no Matlab exposure.

- Weds 9/13/2017 11:00 am 32-D507 Zhoutong
- Thurs 9/14/2017 3:00 pm 32-D507 Jiajun

Signals and systems

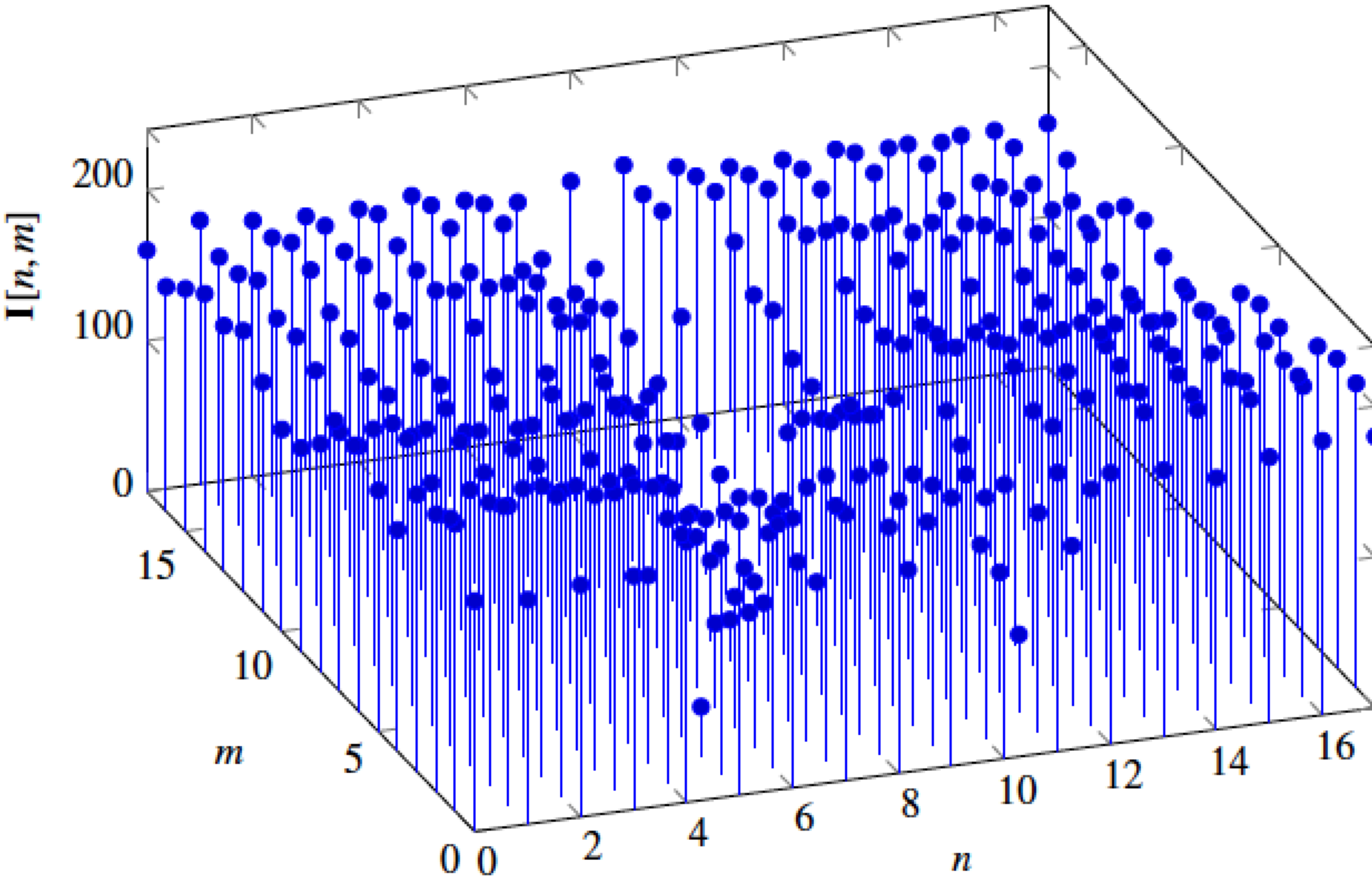


Time continuous signal

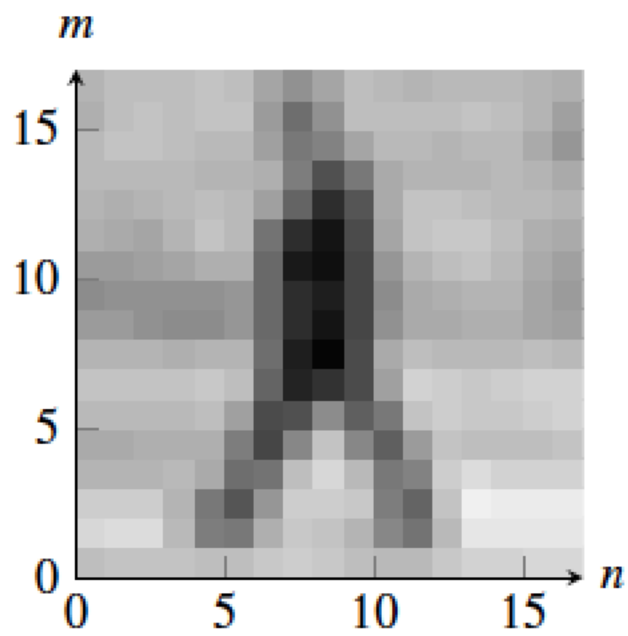


Time discrete signal

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 :

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

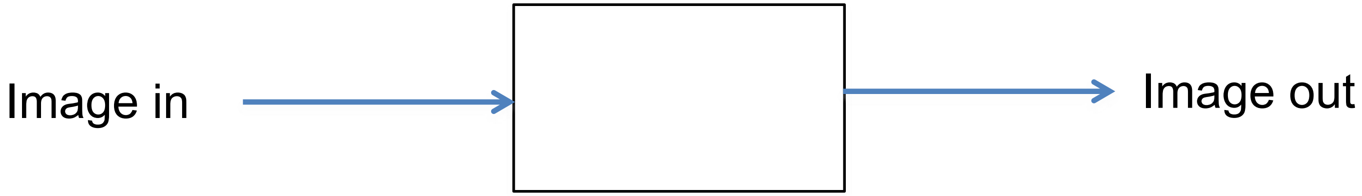
L2 norm of f :

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

Distance between two signals f, g :

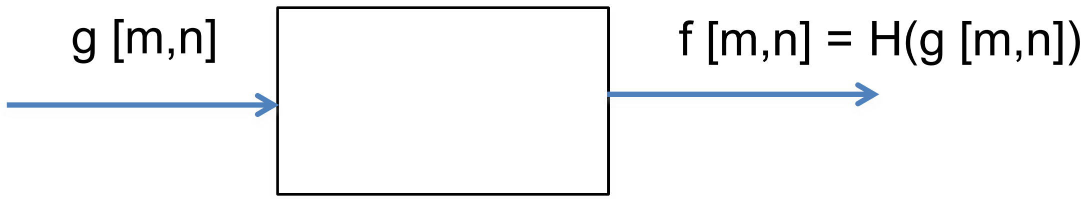
$$d_{f,g}^2 = \|f - g\|^2 = \sum_{n=0}^{N-1} |f[n] - g[n]|^2 = E_f + E_g - 2\langle f, g \rangle$$

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, 1D

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:

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

Linear filtering, 1D

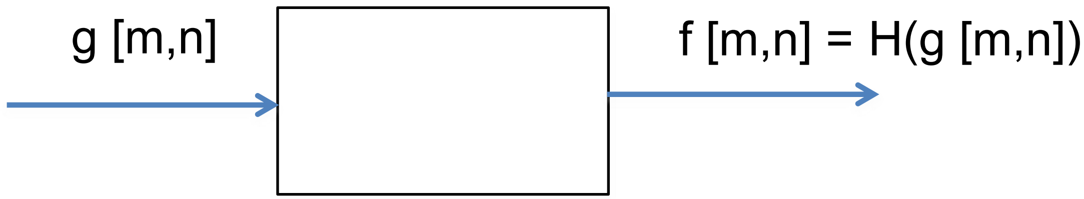
A linear filter in its most general form can be written as, in 1D for a signal of length N:

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

It is useful to make it more explicit by writing:

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

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

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

Why should one pixel be treated differently than any another?

Credit picture: Fredo Durand

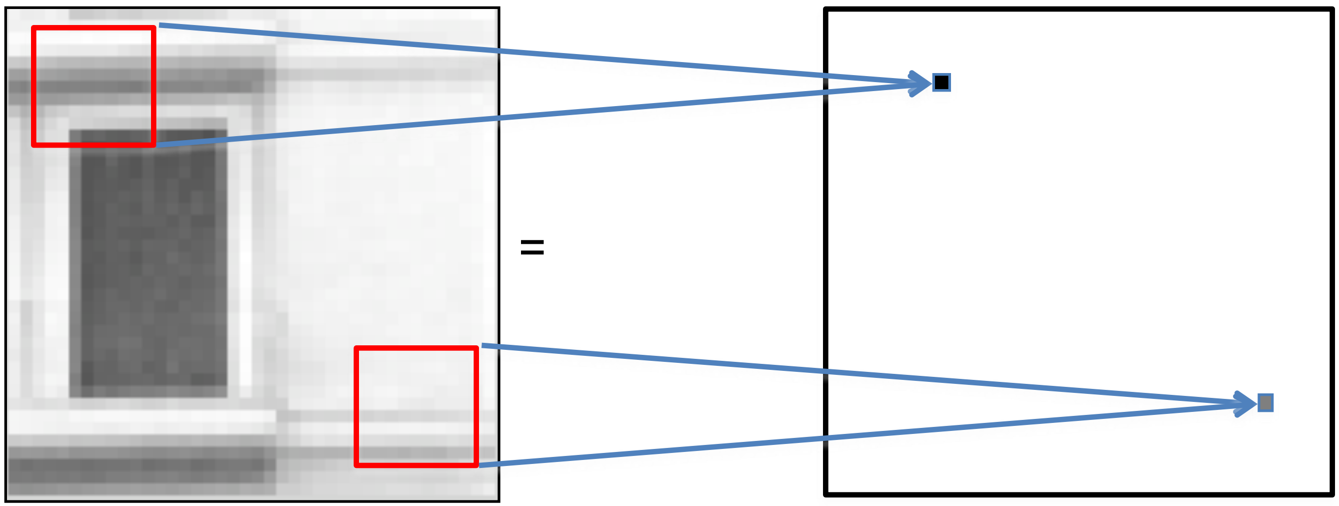
A translation invariant filter

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

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

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

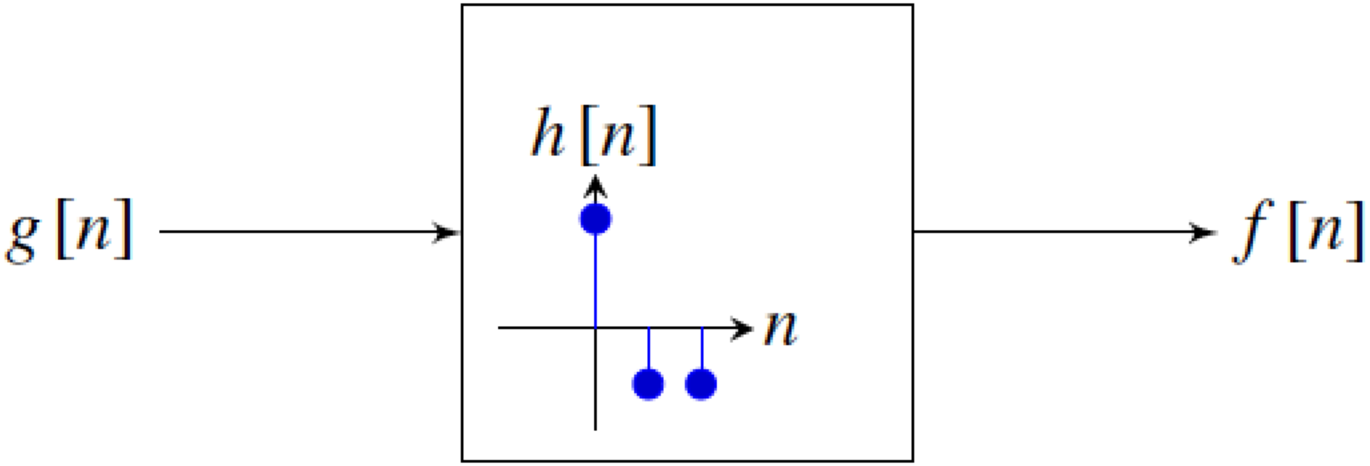


The same weighting occurs within each window

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:

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

Convolution

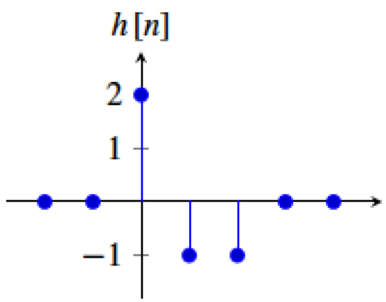
This image cannot currently be displayed.

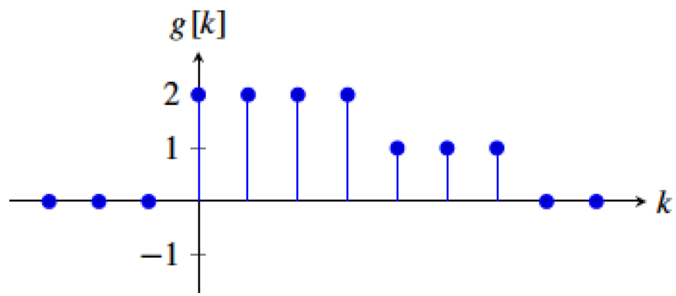
In the 1D case, it helps to make explicit the structure of the matrix:

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

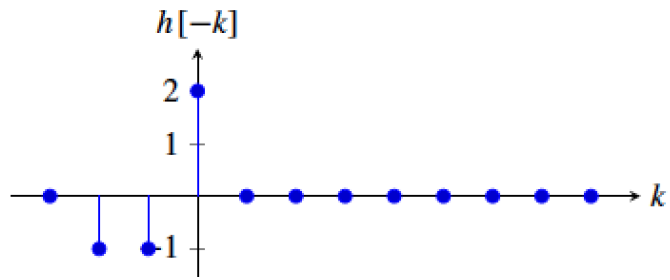
Convolution

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



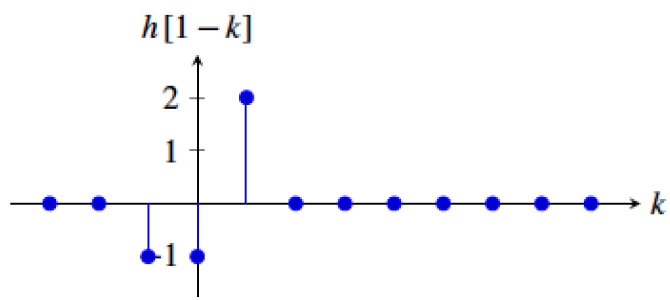


$$f[0] = \sum_k h[-k]\, g[k]$$



$$f[0] = 4$$

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

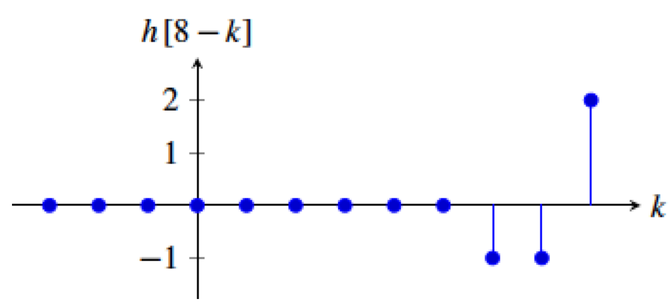


$$f[1] = 2$$

⋮

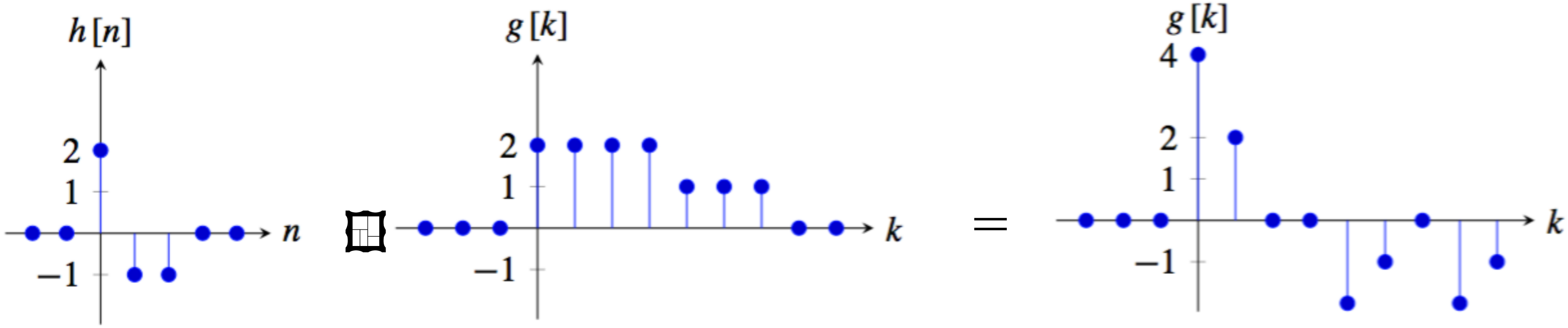
⋮

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



$$f[8] = \text{-}1$$

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

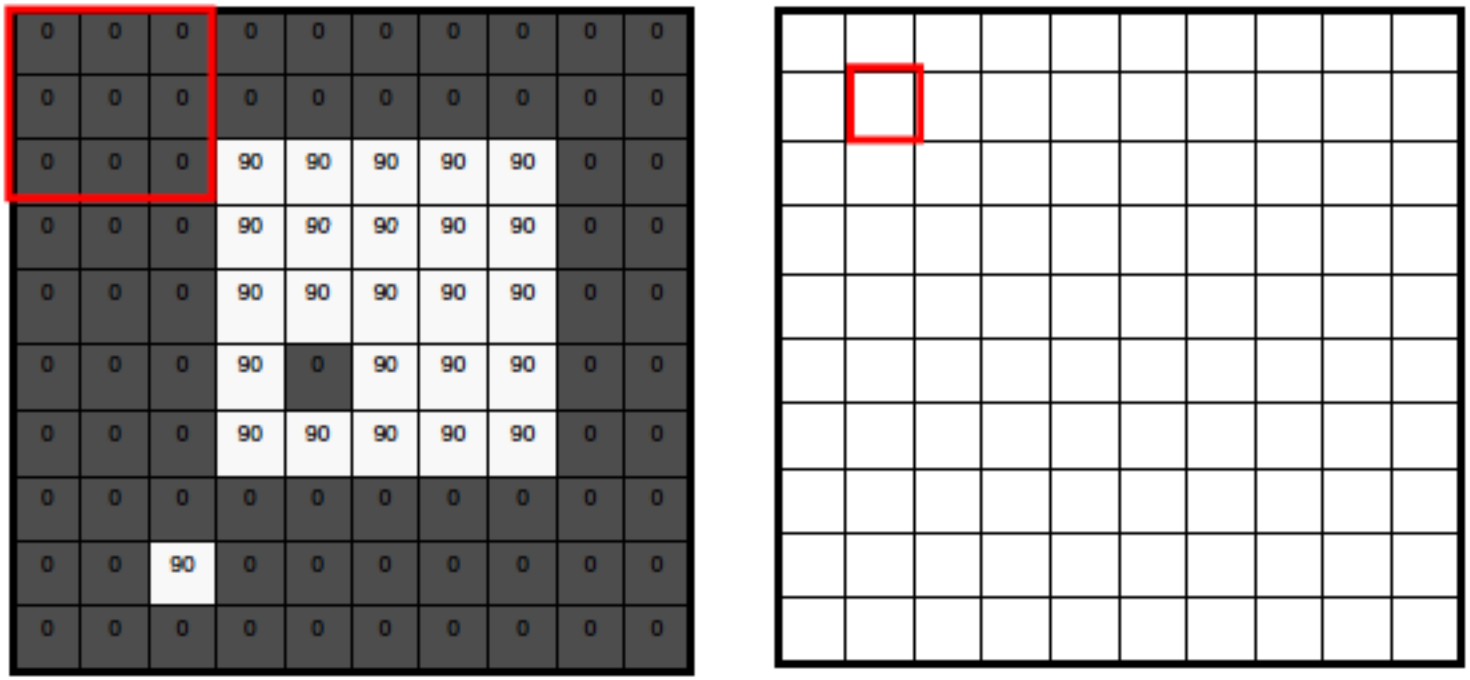
$$h[n] \circ (f[n] + g[n]) = h[n] \circ f[n] + h[n] \circ g[n]$$

Shift property

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

2D convolution

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

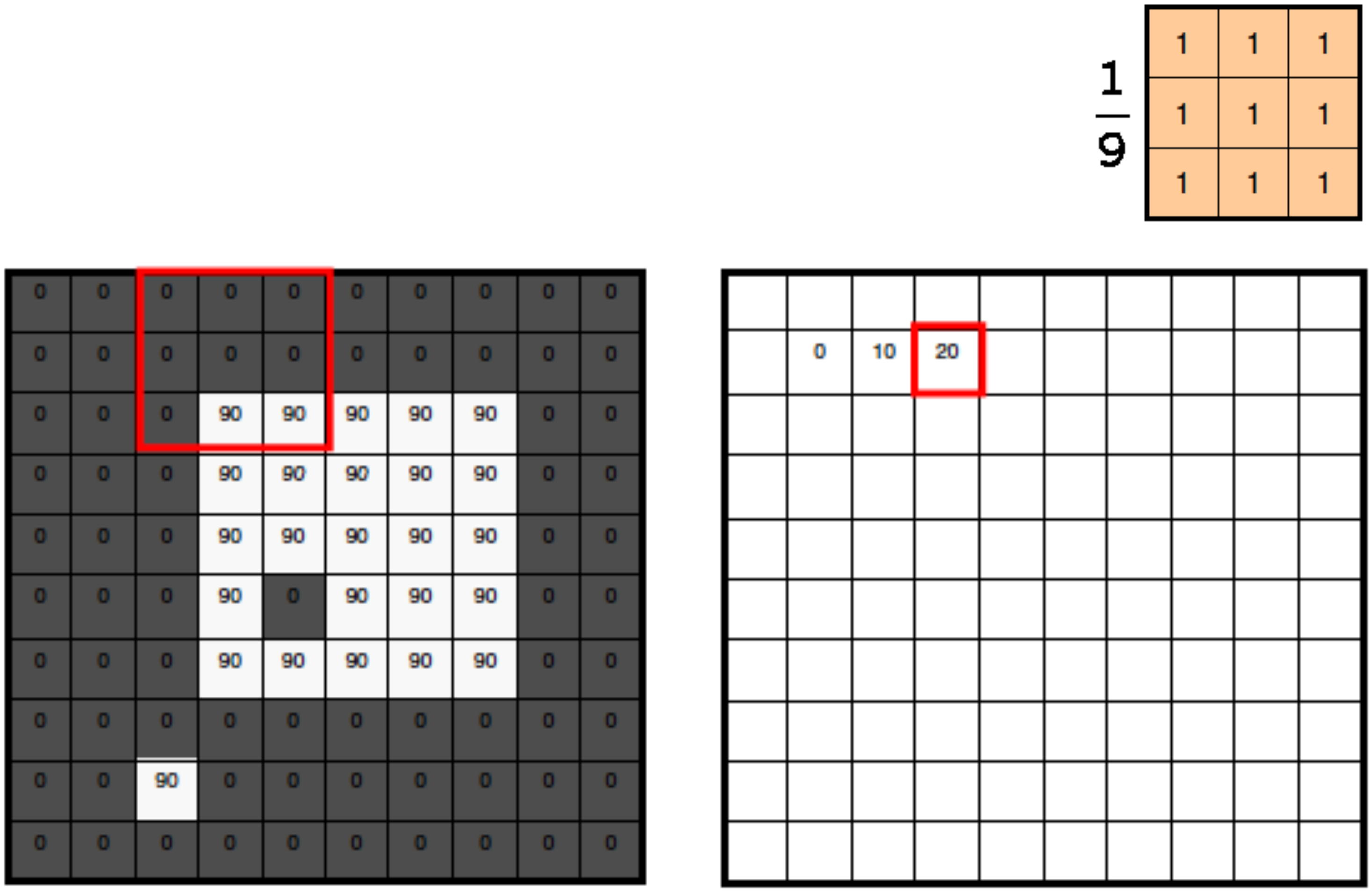


$$f[m,n] = h \circ g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$

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

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							



1/9
1 1 1
1 1 1
1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 0 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 0 0 0 0 0 0 0
0 0 90 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 10 20

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

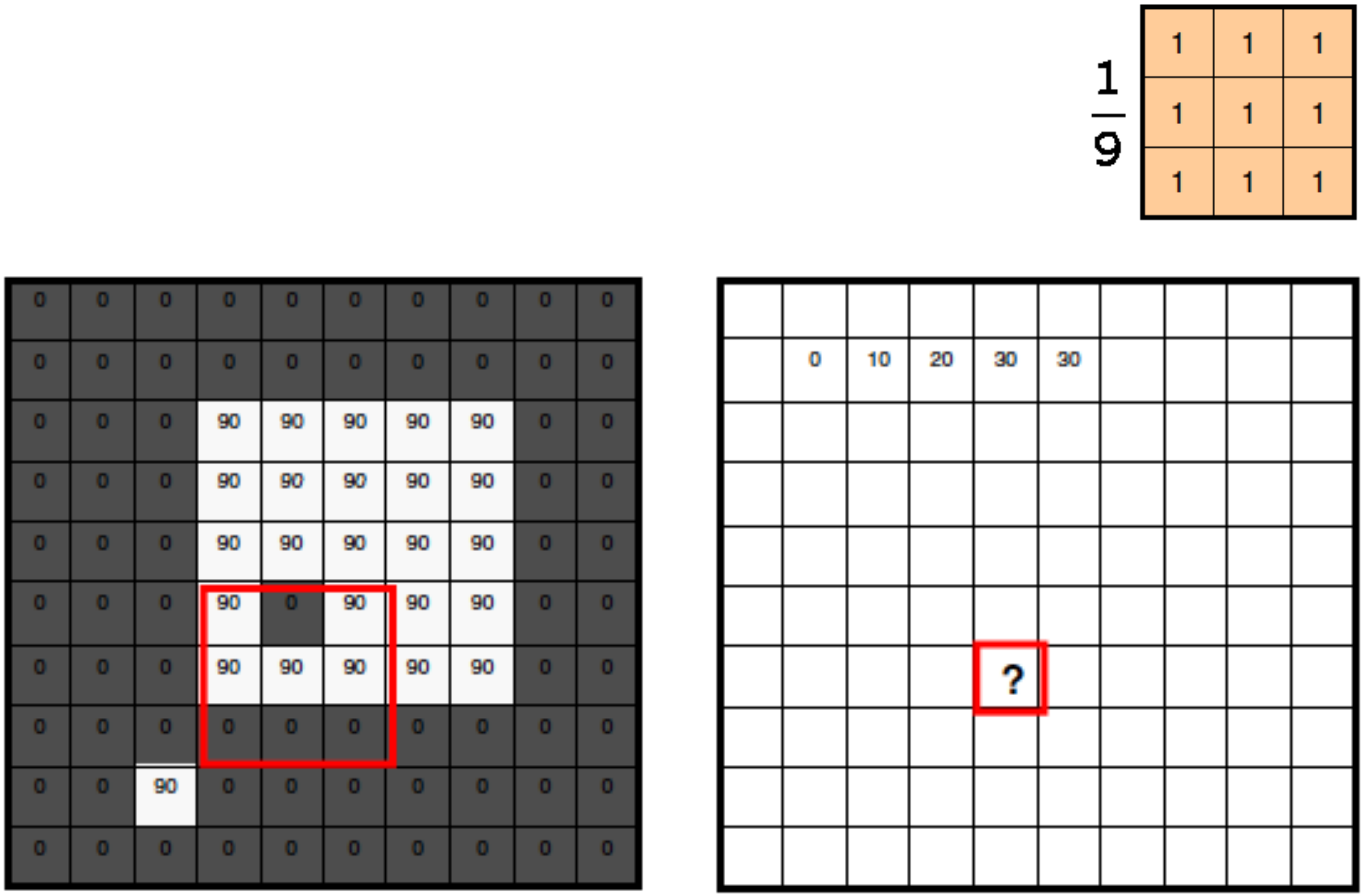
	0	10	20	30					

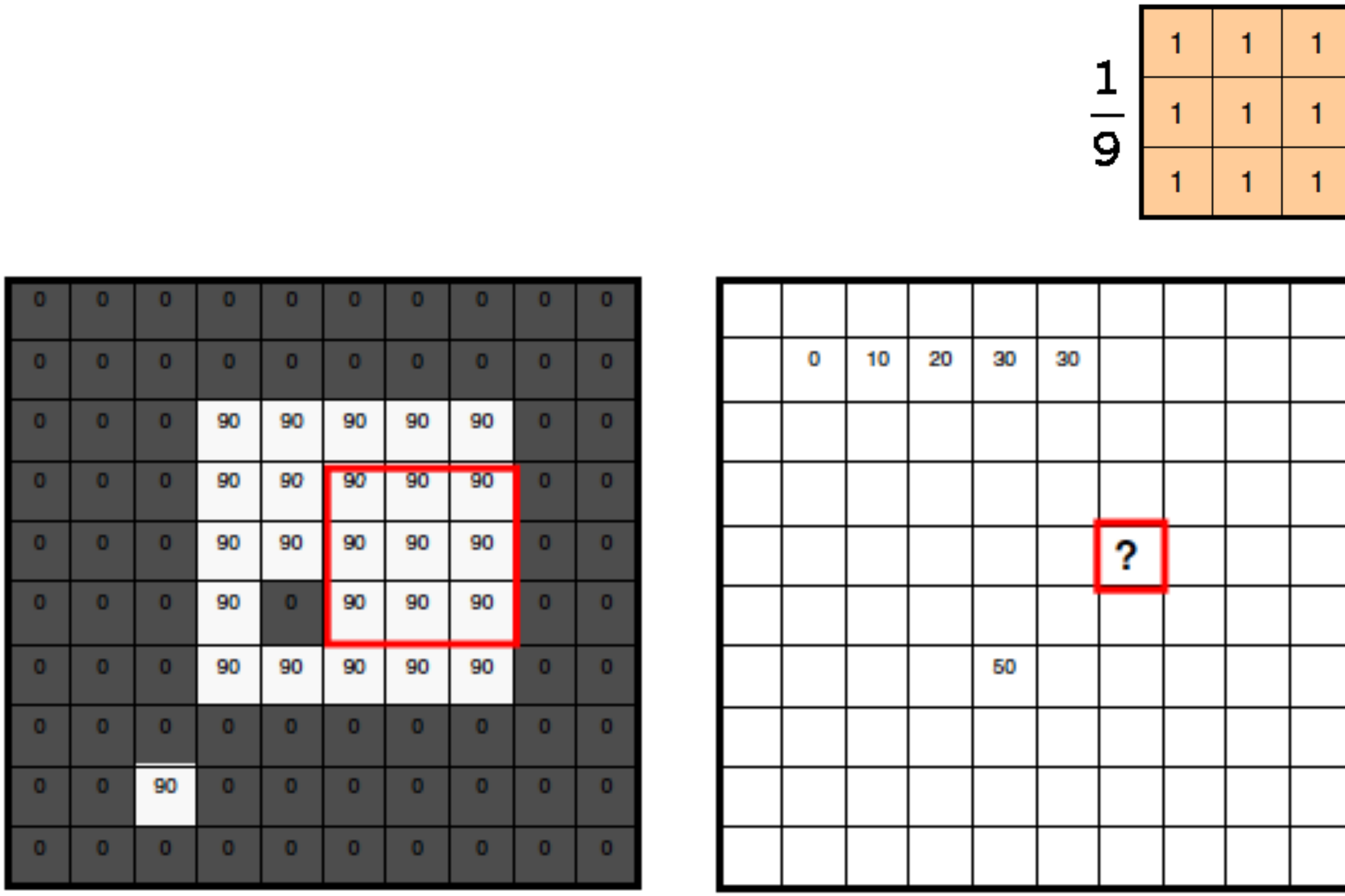
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

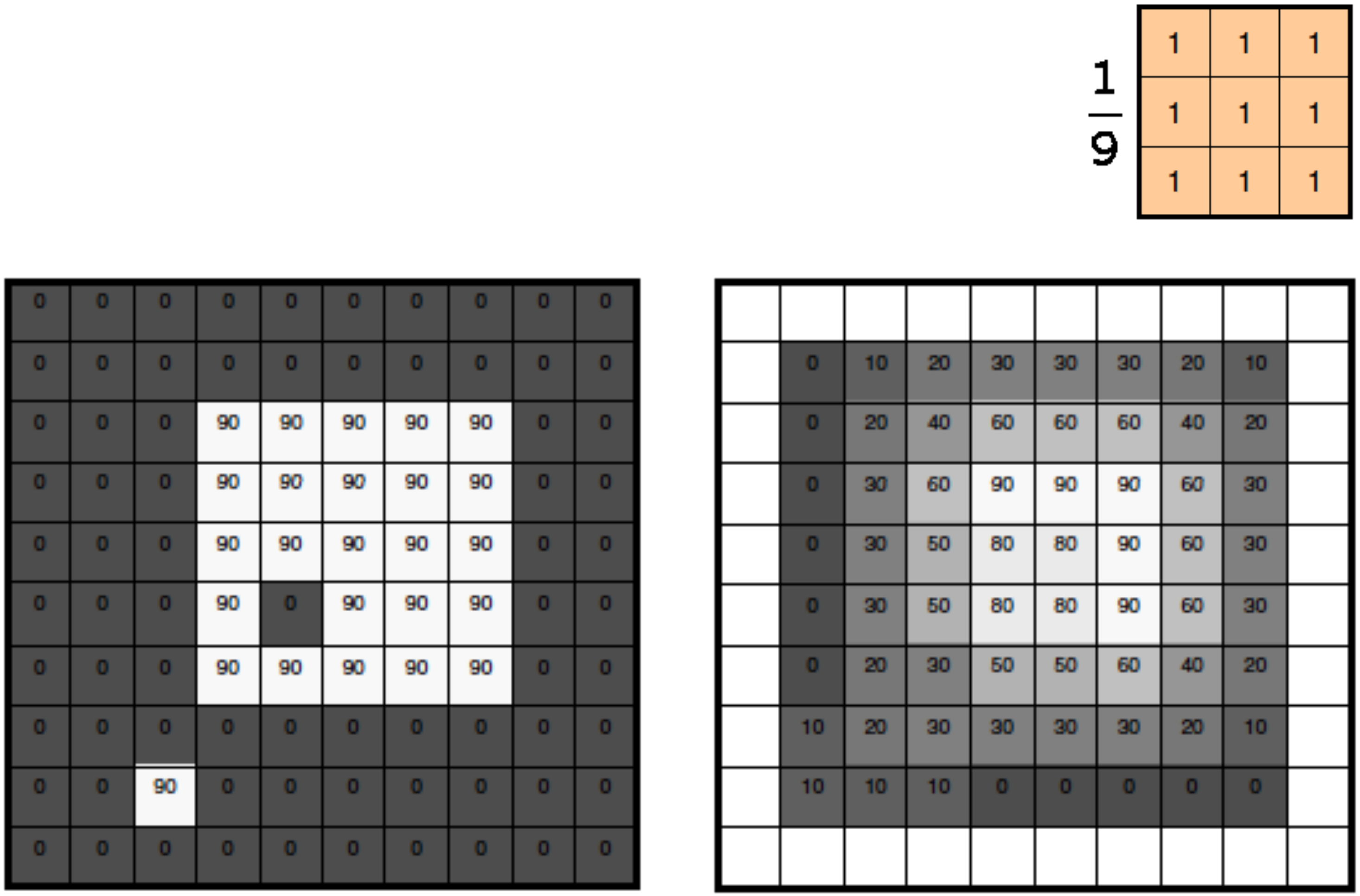
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30				


1/9
1 1 1
1 1 1
1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 0 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 0 0 0 0 0 0 0
0 0 90 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 10 20 30 30
?



1/9
1 1 1
1 1 1
1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 90 0 90 90 90 0 0
0 0 0 90 90 90 90 90 0 0
0 0 0 0 0 0 0 0 0 0
0 0 90 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 10 20 30 30
?
50



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

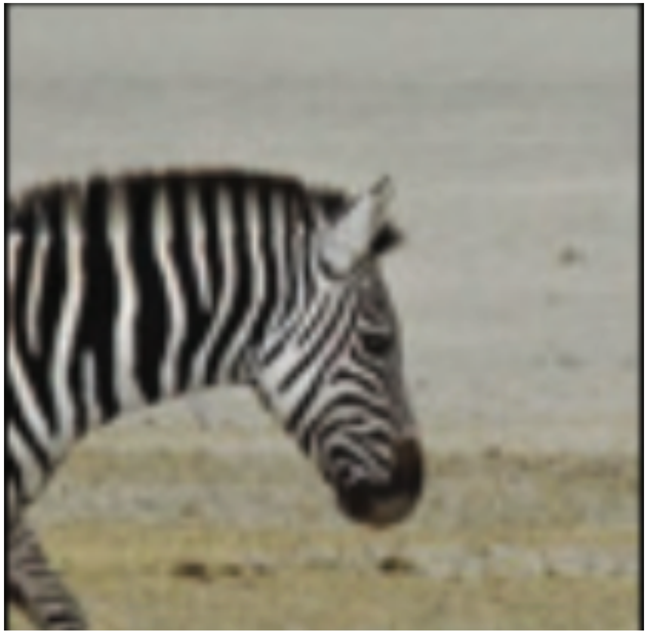
h[m,n]

=

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

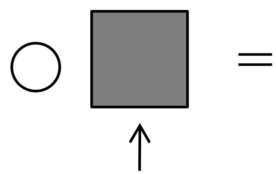
f[m,n]

Handling boundaries

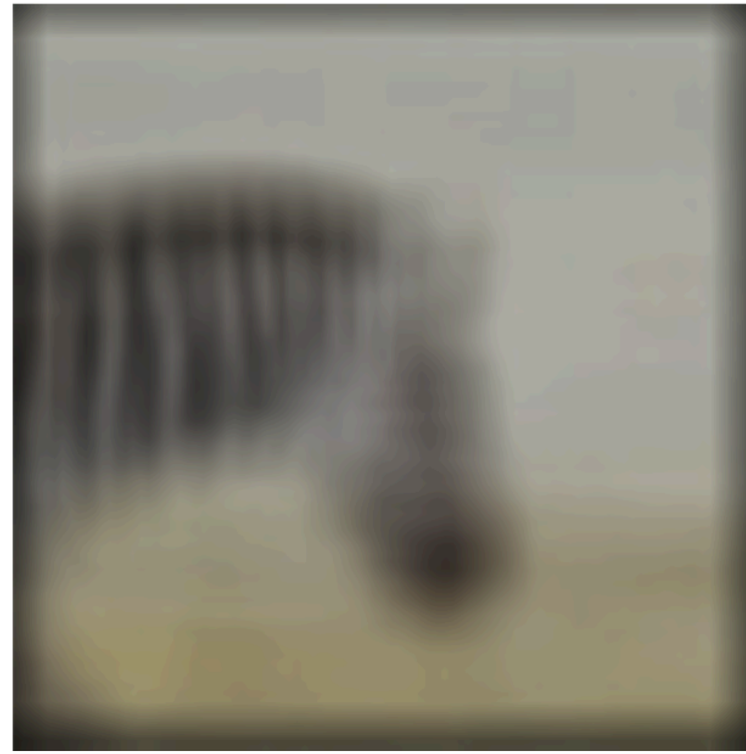


Handling boundaries

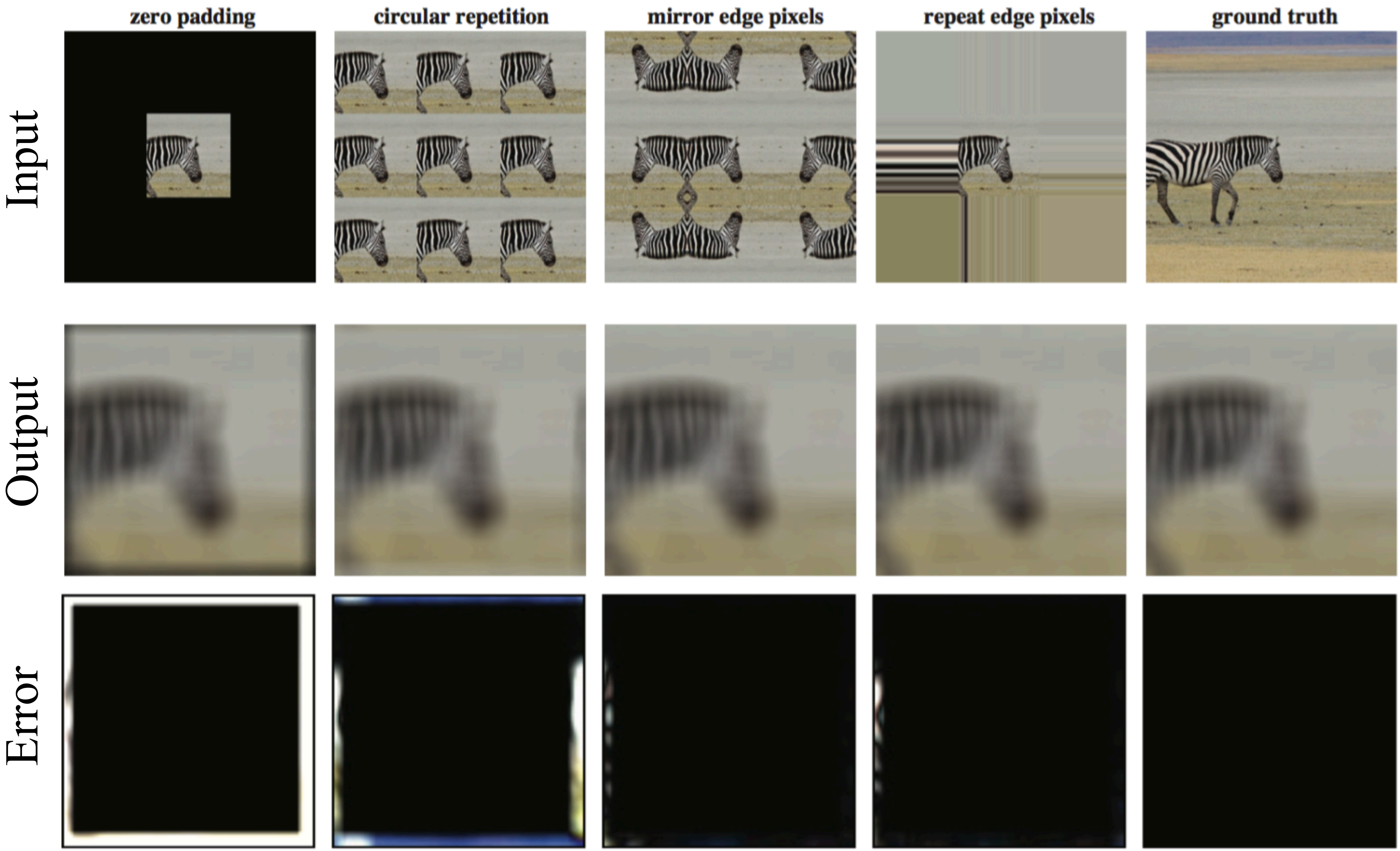
Zero padding



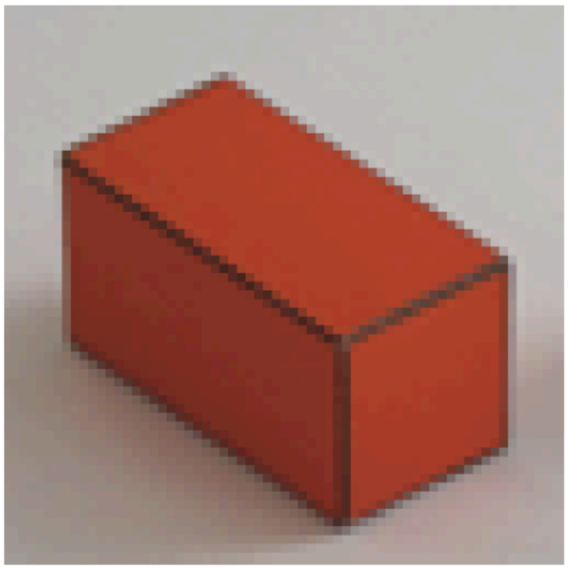
11x11 ones



Handling boundaries



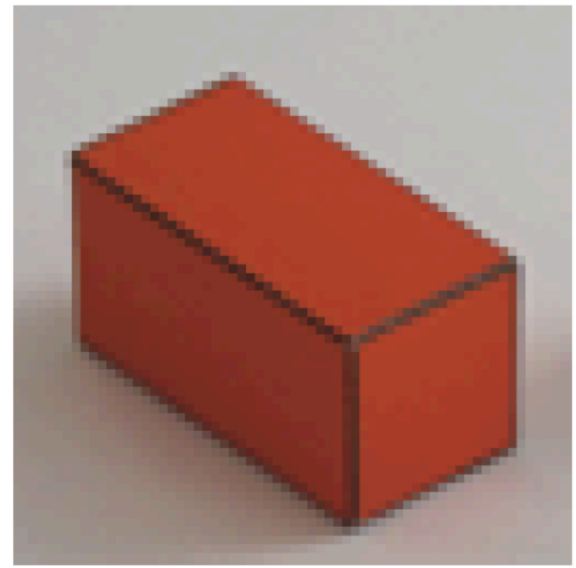
Examples



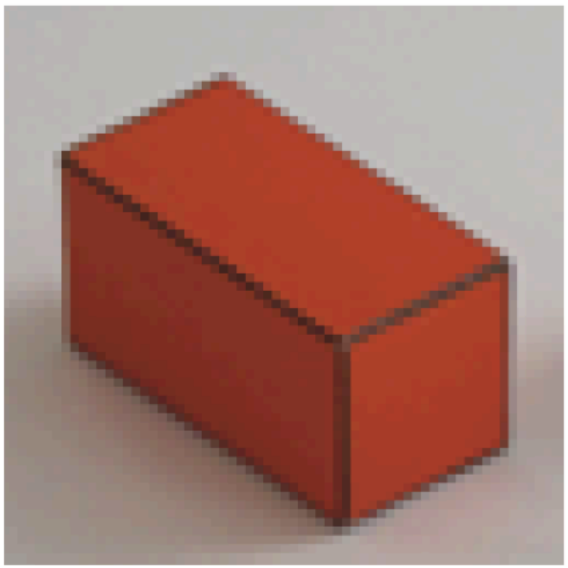
$\circ$

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

$=$



Examples

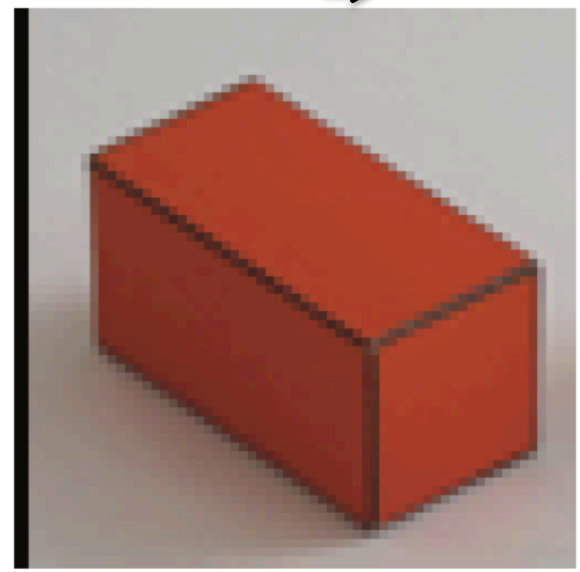


○

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

=

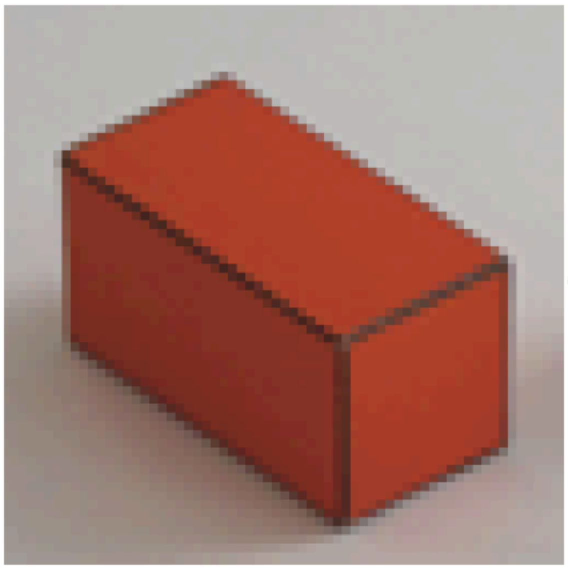
2 pixels →



↑

(using zero padding)

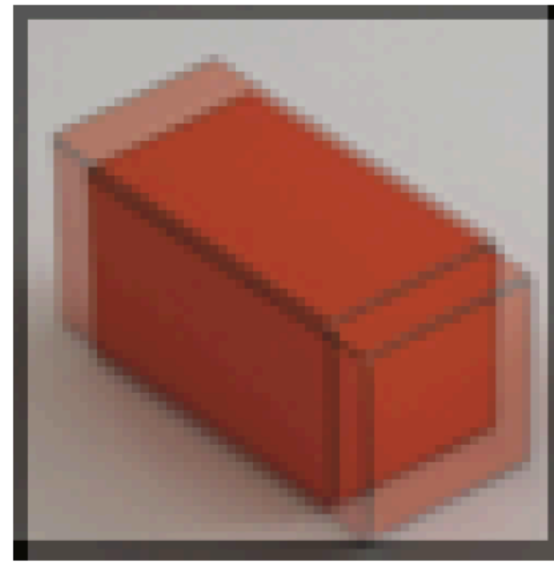
Examples



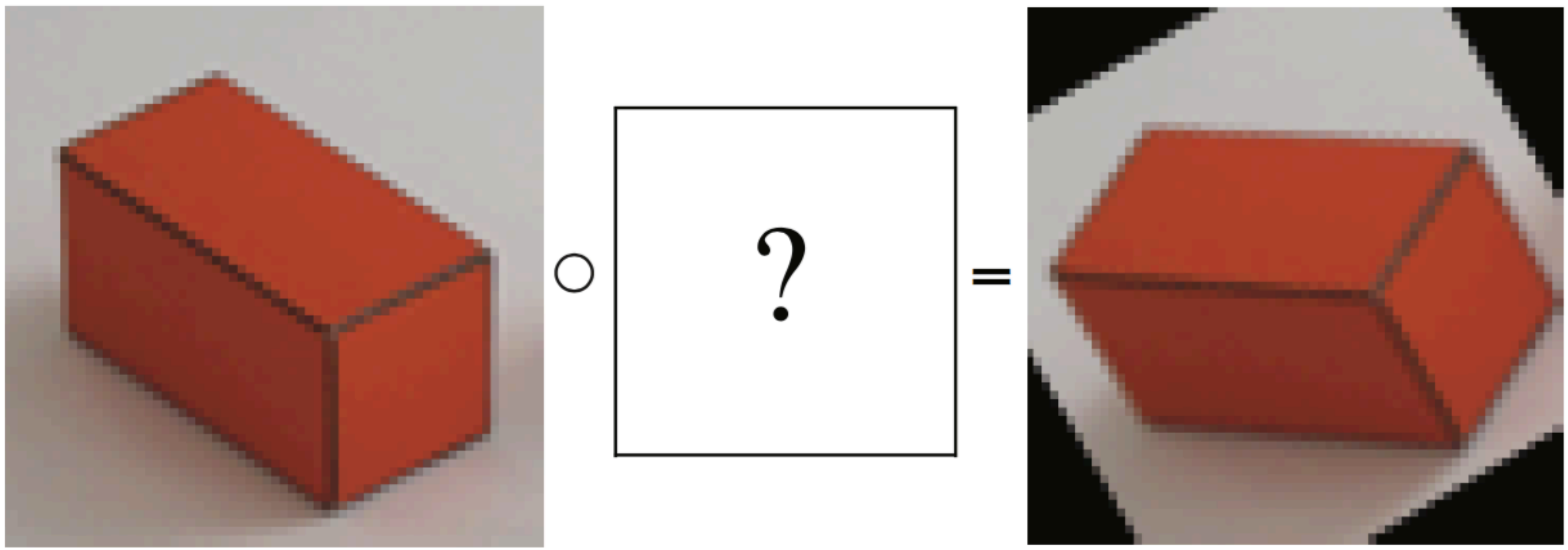
$\circ$

.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]

$\otimes$

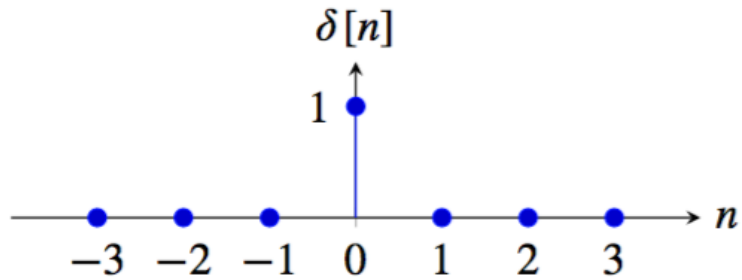
h[m,n]

=

f[m,n]

Important signals

The impulse



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

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

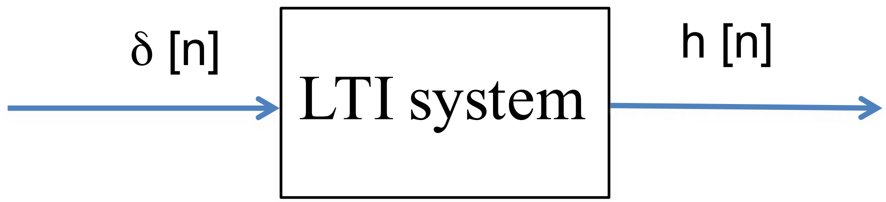
Convolving a signal f with a translated impulse $\delta\,[n - n_0]$ results in a translated signal:

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

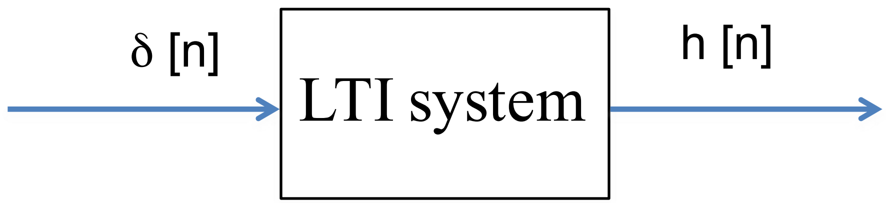
Why the impulse is so important

$$f[n] = \sum_k f[k]\delta[n-k]$$

Write the input signal as a sum of impulses

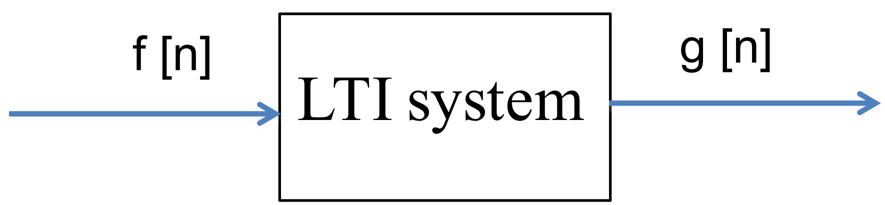


Why the impulse is so important



Passing f[n] through the LTI system, replace every δ[n] in f[n] with h[n]

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



Then the output of an LTI system is the corresponding sum of impulse responses.

Important signals

Cosine and sine waves

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

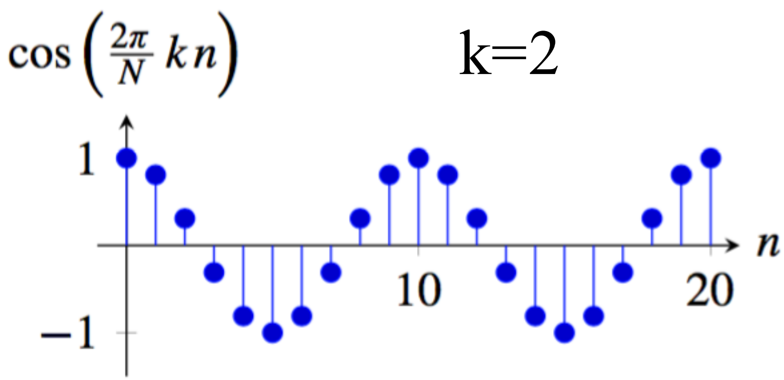
A discrete signal $f[n]$ is periodic, if there exists $T \in$ integers such that $f[n] = f[n + mT]$ for all $m \in$ integers. For the discrete sine (and cosine) wave to be periodic the frequency has to be $w = 2\pi K/N$ for $K,N \in$ integers. If K/N is an irreducible fraction, then the period of the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

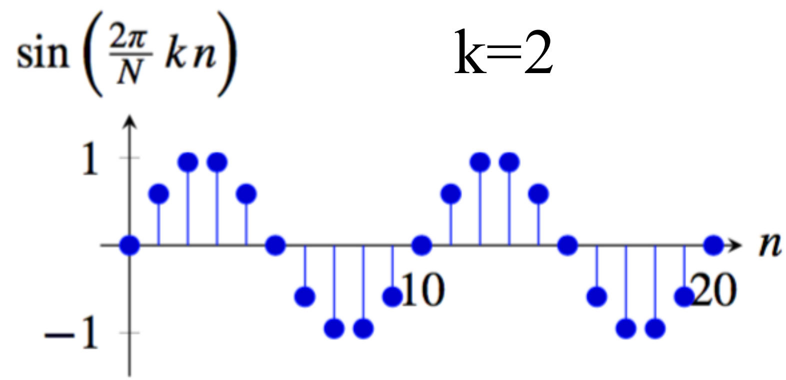
$$s_k[n] = \sin\left(\frac{2\pi}{N} k n\right) \qquad c_k[n] = \cos\left(\frac{2\pi}{N} k n\right)$$

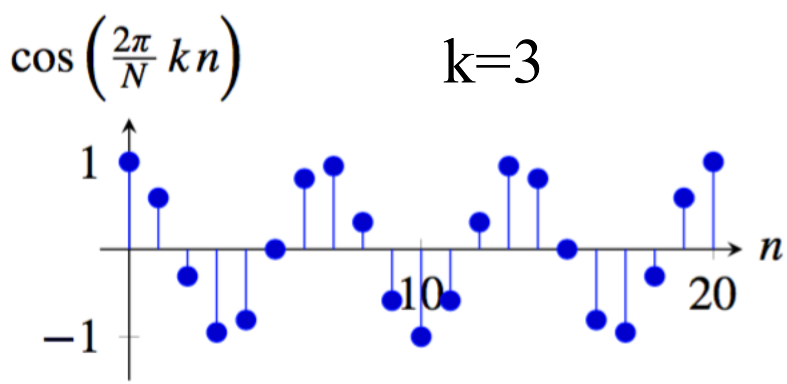
$k \in [1, N/2]$ denotes the number of wave cycles that will occur within the region of support

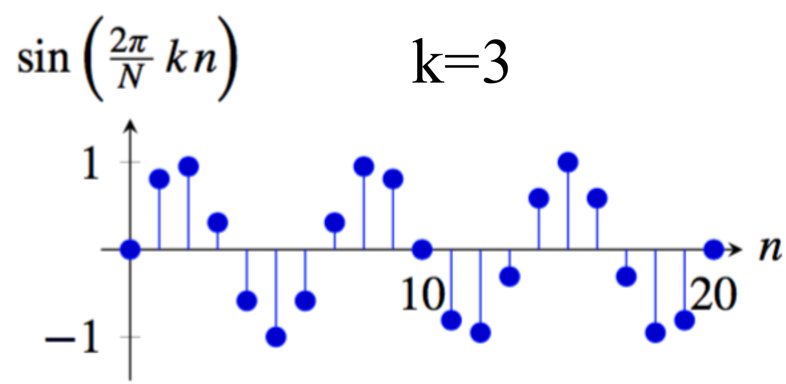
Important signals

Cosine and sine waves, N=20



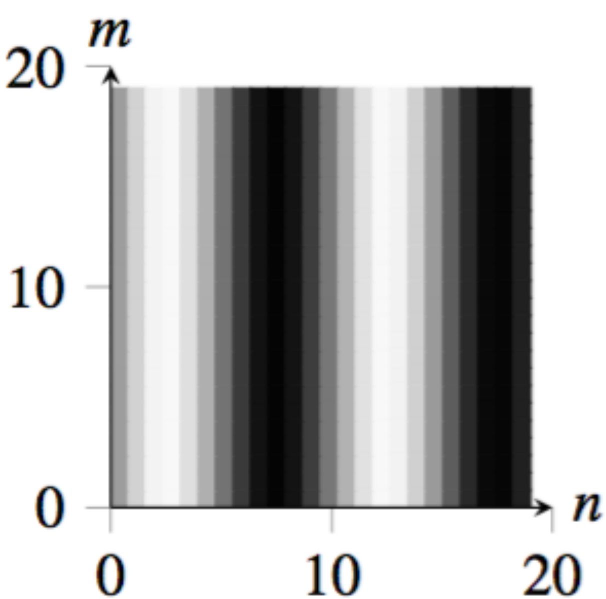




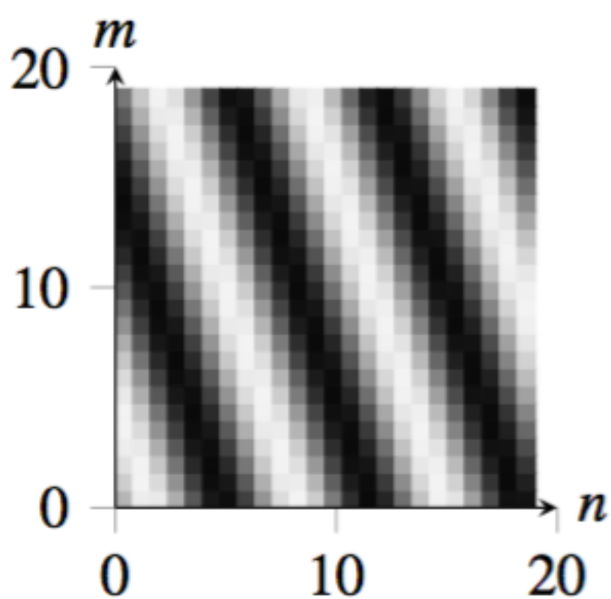


Waves in 2D

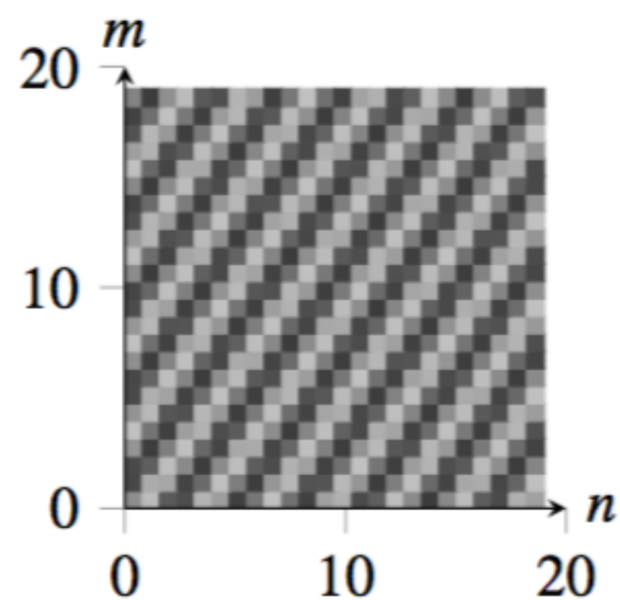
$$s_{u,v}[n,m] = A\sin\left(2\pi\left(\frac{un}{N}+\frac{vm}{M}\right)\right) \qquad c_{u,v}[n,m] = A\cos\left(2\pi\left(\frac{un}{N}+\frac{vm}{M}\right)\right)$$



u = 2, v = 0



u = 3, v = 1



u = 7, v = −5

Important signals

Complex exponential

$$s(t) = A \exp(j w t)$$

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

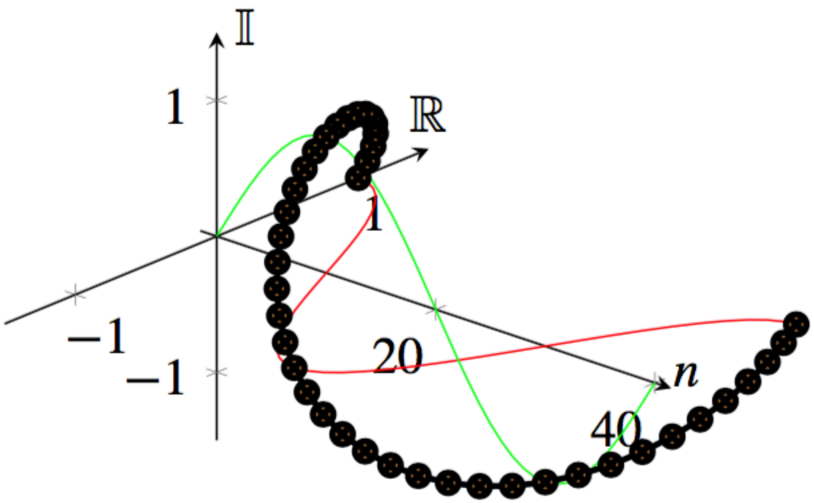
$$e_k[n] = \exp\left(j \frac{2\pi}{N} k n\right) = \cos\left(\frac{2\pi}{N} k n\right) + j \sin\left(\frac{2\pi}{N} k n\right)$$

And in 2D, the complex exponential wave is:

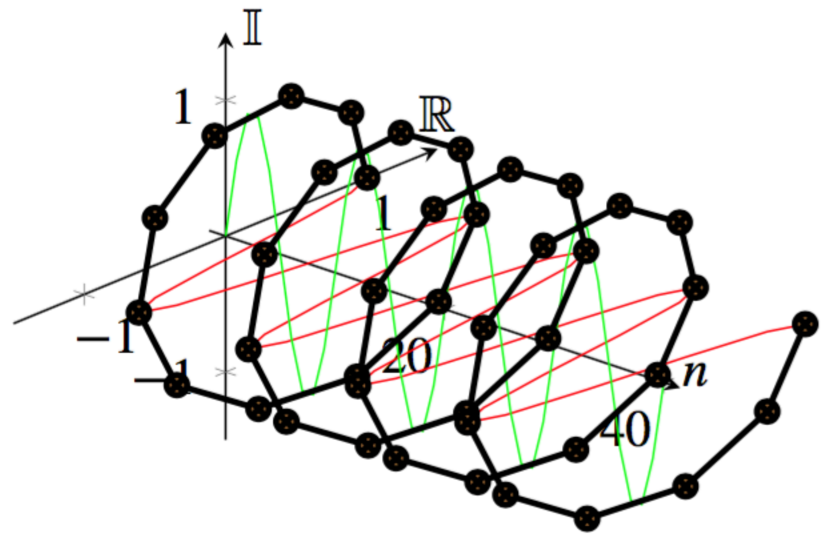
$$e_{u,v}[n, m] = \exp\left(2\pi j \left(\frac{u n}{N} + \frac{v m}{M}\right)\right)$$

Important signals

Complex exponential



$N = 40,\ k = 1$



$N = 40,\ k = 4$

Each of impulses, sine and cosine waves or complex exponentials can form an orthogonal basis for signals of length N

Linear image transformations

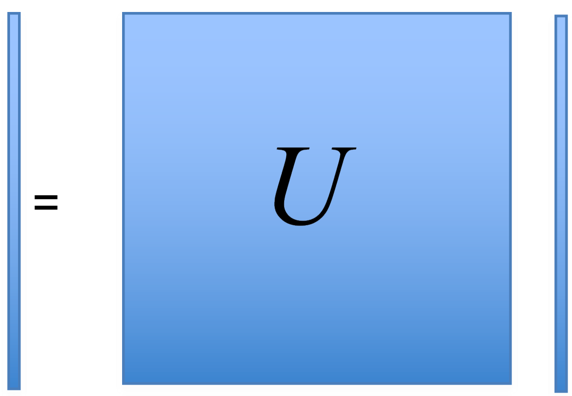
In analyzing images, it's often useful to make a change of basis.

Transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



Self-inverting transforms

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

Same basis functions are used for the inverse transform

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

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

U transpose and complex conjugate

The Discrete Fourier transform

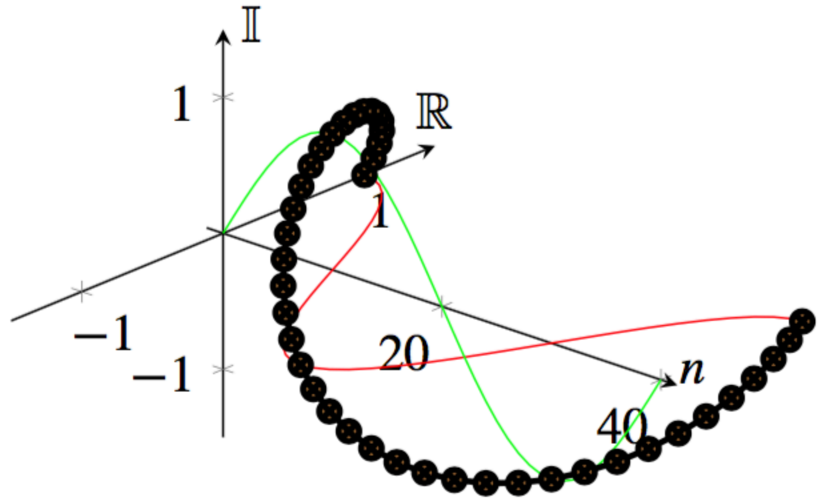
Discrete Fourier Transform (DFT) transforms an image $f[m, m]$ into the complex image Fourier transform $F[u, v]$ as:

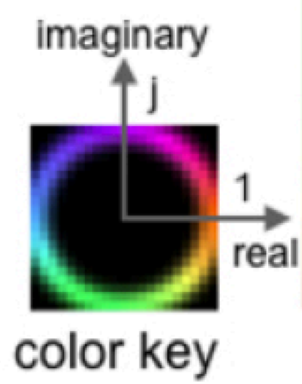
$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{u\,n}{N} + \frac{v\,m}{M}\right)\right)$$

The inverse Fourier transform is:

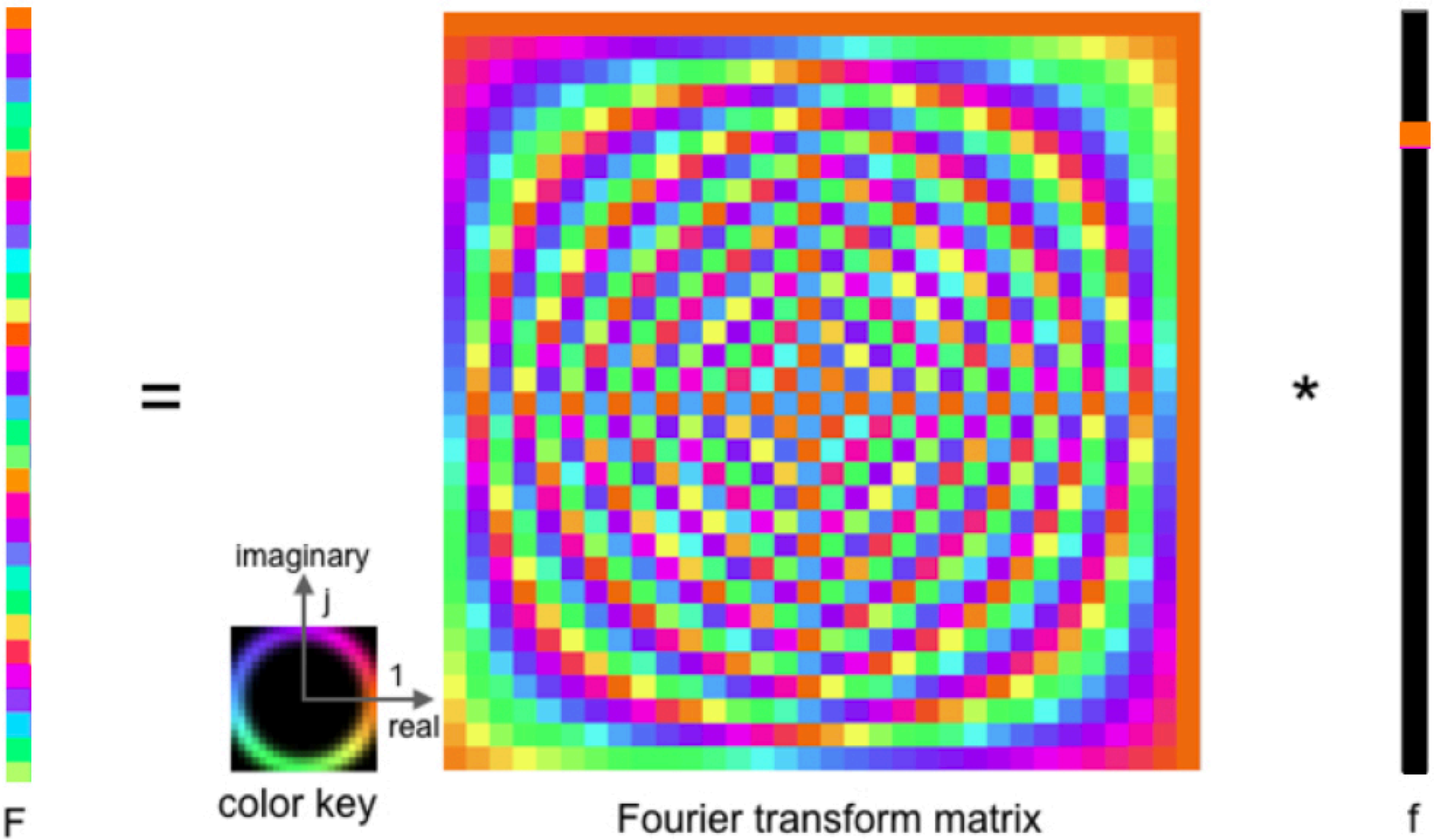
$$f[n,m] = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F[u,v] \exp\left(+2\pi j \left(\frac{u\,n}{N} + \frac{v\,m}{M}\right)\right)$$

Discrete Fourier transform visualization





Fourier transform visualization



Some useful transforms

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

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

If we apply the inverse DFT to both sides, we have:

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

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:

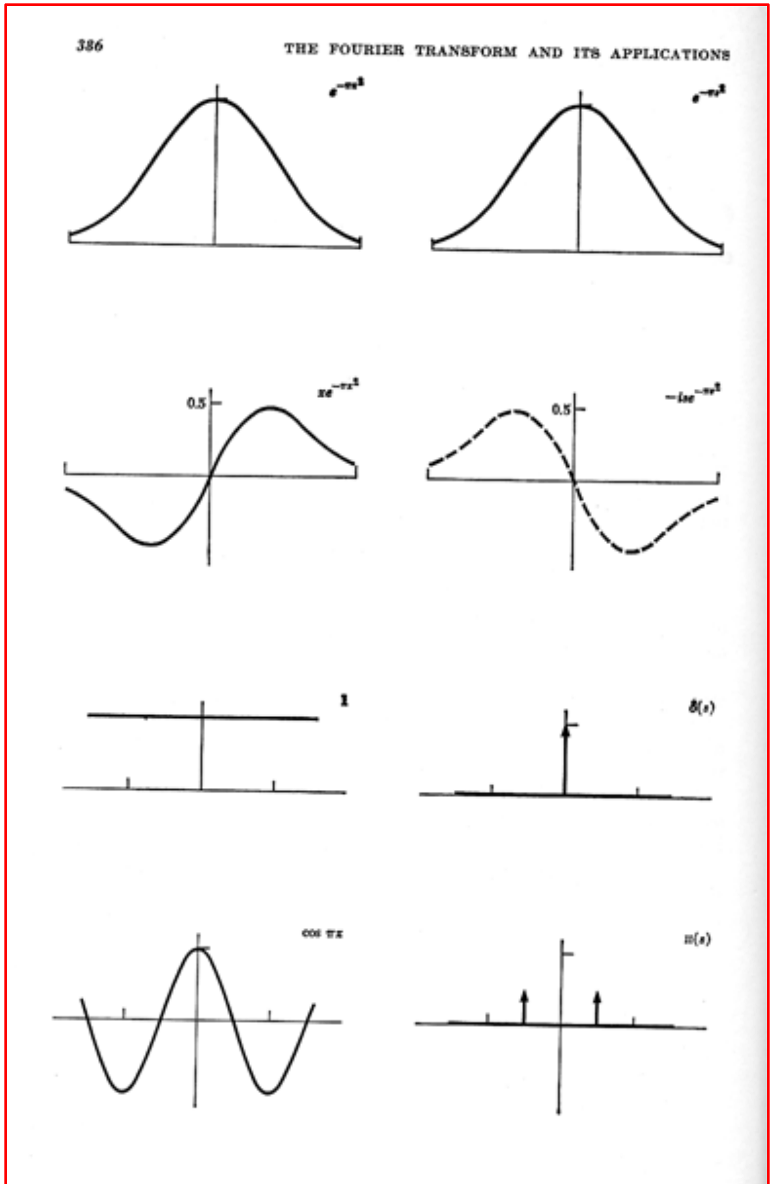
$$F[u,v]=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\cos\left(2\pi\left(\frac{u_0 n}{N}+\frac{v_0 m}{M}\right)\right)\exp\left(-2\pi j\left(\frac{u n}{N}+\frac{v m}{M}\right)\right)=$$

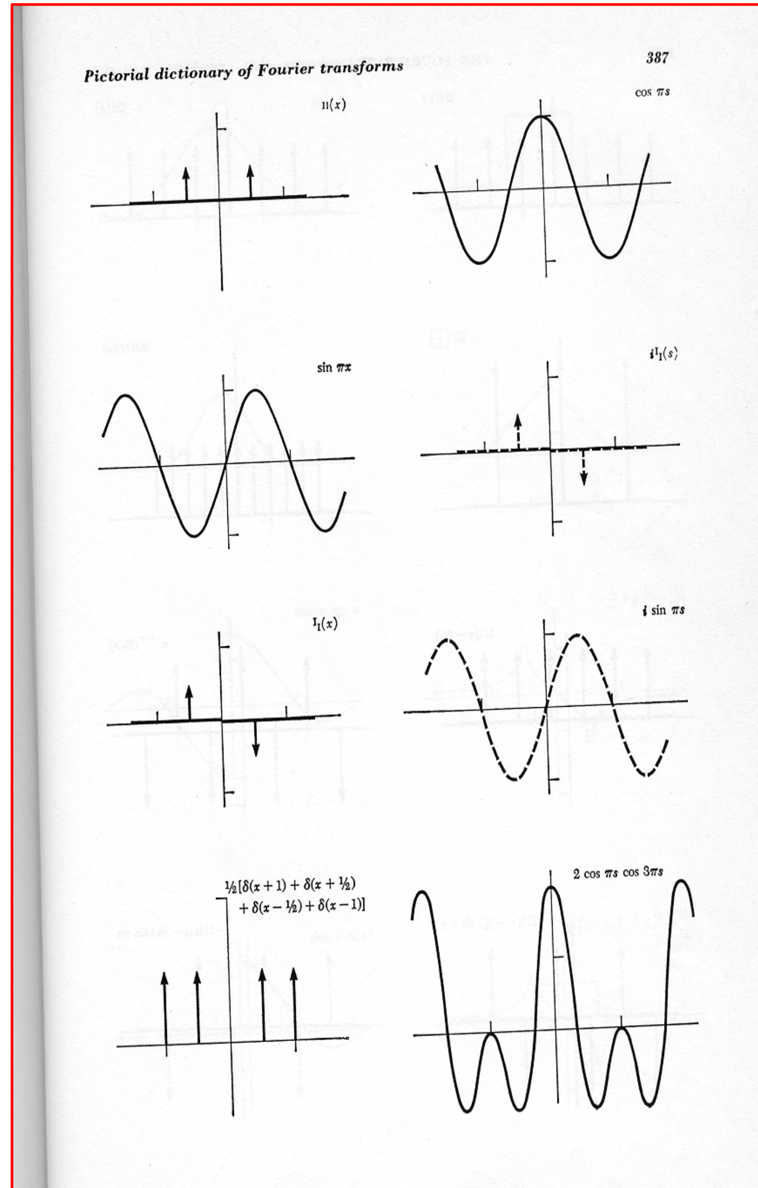
$$=\frac{1}{2}\left(\delta[u-u_0,v-v_0]+\delta[u+u_0,v+v_0]\right)$$

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

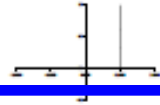
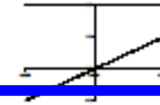
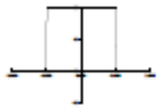
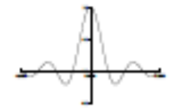
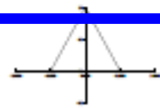
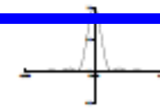
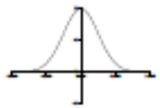
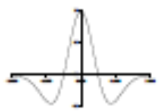
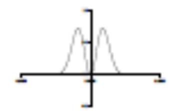
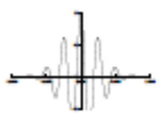
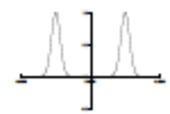
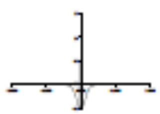
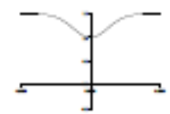
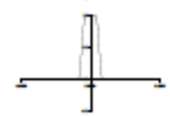
Name		Signal		Transform	
impulse		$\delta(x)$	$\Leftrightarrow$	1	
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$	
box filter		$\text{box}(x/a)$	$\Leftrightarrow$	$a\text{sinc}(a\omega)$	
tent		$\text{tent}(x/a)$	$\Leftrightarrow$	$a\text{sinc}^2(a\omega)$	
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$	
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$	
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		$\text{rcos}(x/(aW))$ $\text{sinc}(x/a)$	$\Leftrightarrow$	(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 particular, the Laplacian of Gaussian is drawn inverted because it resembles more a "Mexican hat", as it is sometimes called.

2D Discrete Fourier Transform

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

Note that 2D (and higher-D) DFT's are separable:

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

This is a 1D DFT over m, followed by 1D DFT over n.

2D Discrete Fourier Transform

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

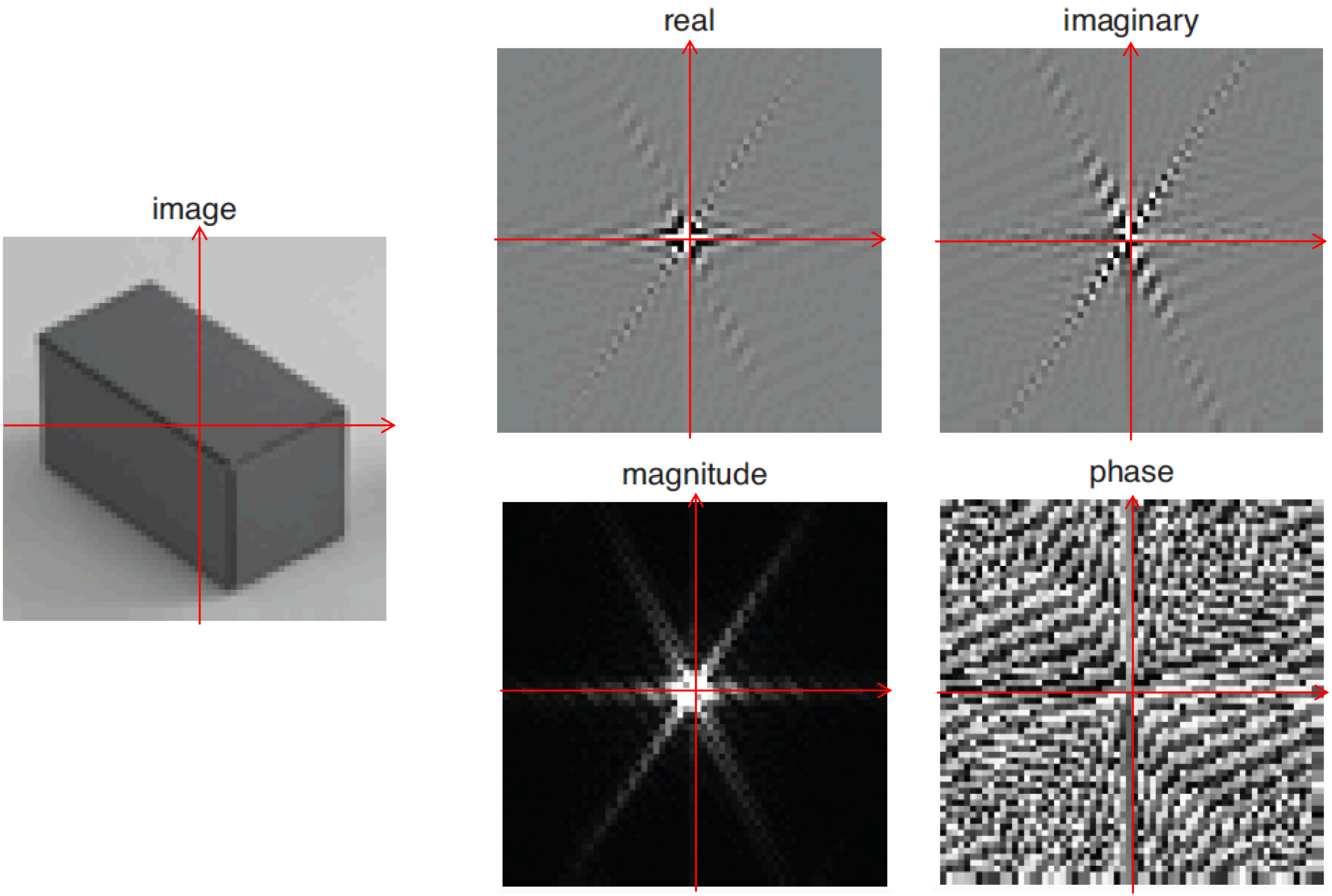
Using the real and imaginary components:

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

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

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

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

- Both the DFT and its inverse are periodic

As F [u, v] is obtained as a sum of complex exponential with a common period of N , M samples, the function F [u, v] is also periodic: F [u + aN, v + bM] = f [u, v] for any a, b $\in$ Z. Also the result of the inverse DFT is a periodic image: f [n + aN, m + bM] = f [n, m] for any a, b $\in$ Z.

Properties for the DFT

- Shift in space

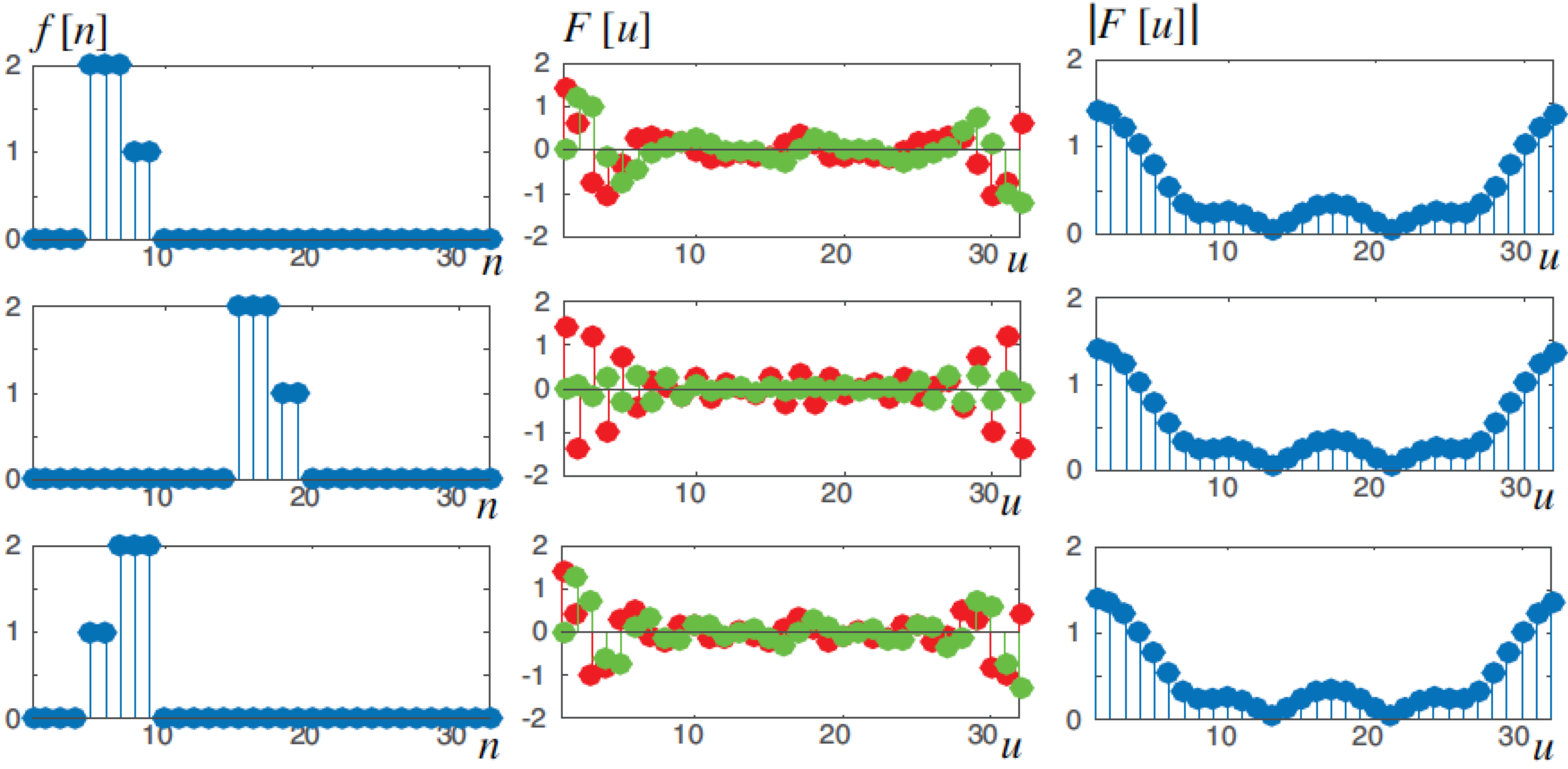
$$DFT\{f[n-n_0, m-m_0]\} =$$

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

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

$$= \boxed{F[u, v] \exp\left(-2\pi j \left(\frac{un_0}{N} + \frac{vm_0}{M}\right)\right)}$$

Properties for the DFT



Only the phase changes! The magnitude is translation invariant.

Properties for the DFT

- Modulation

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

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

DFT amplitude

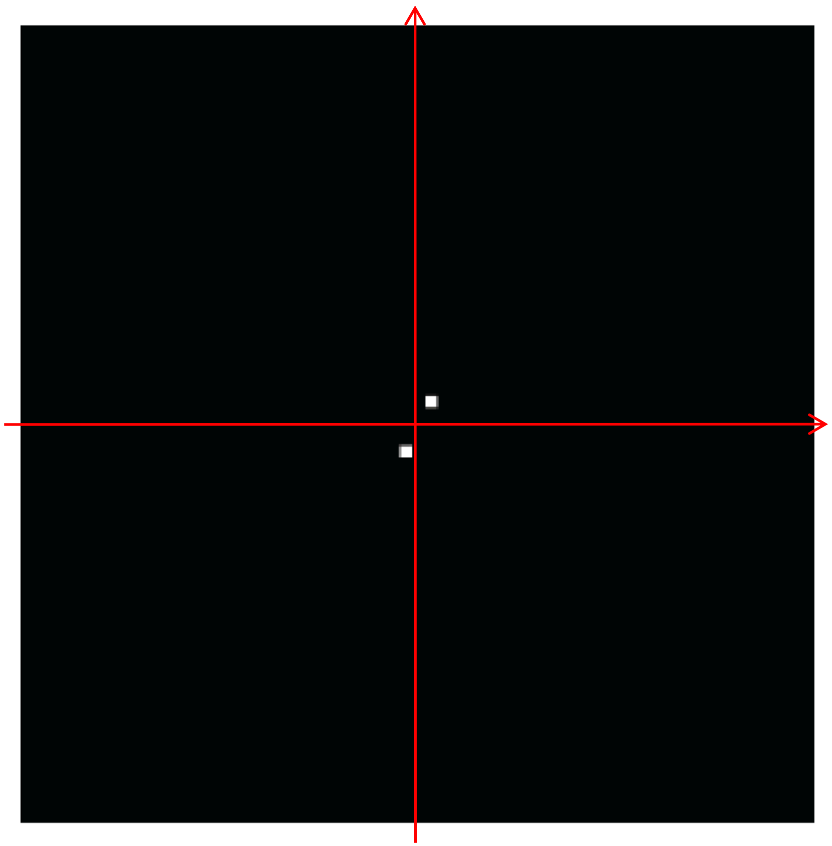
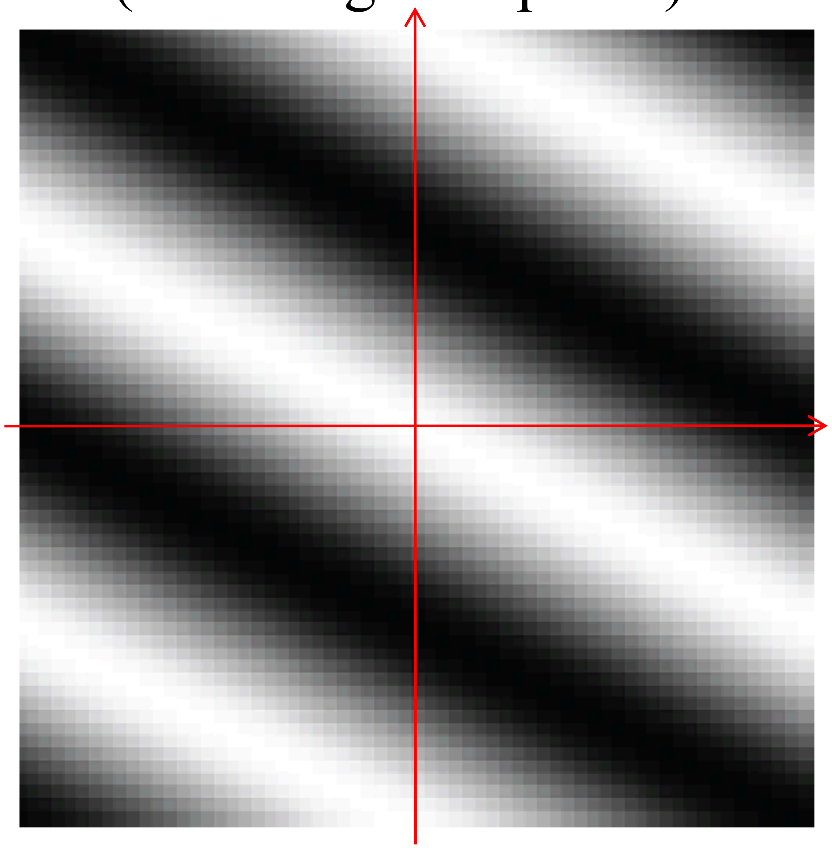
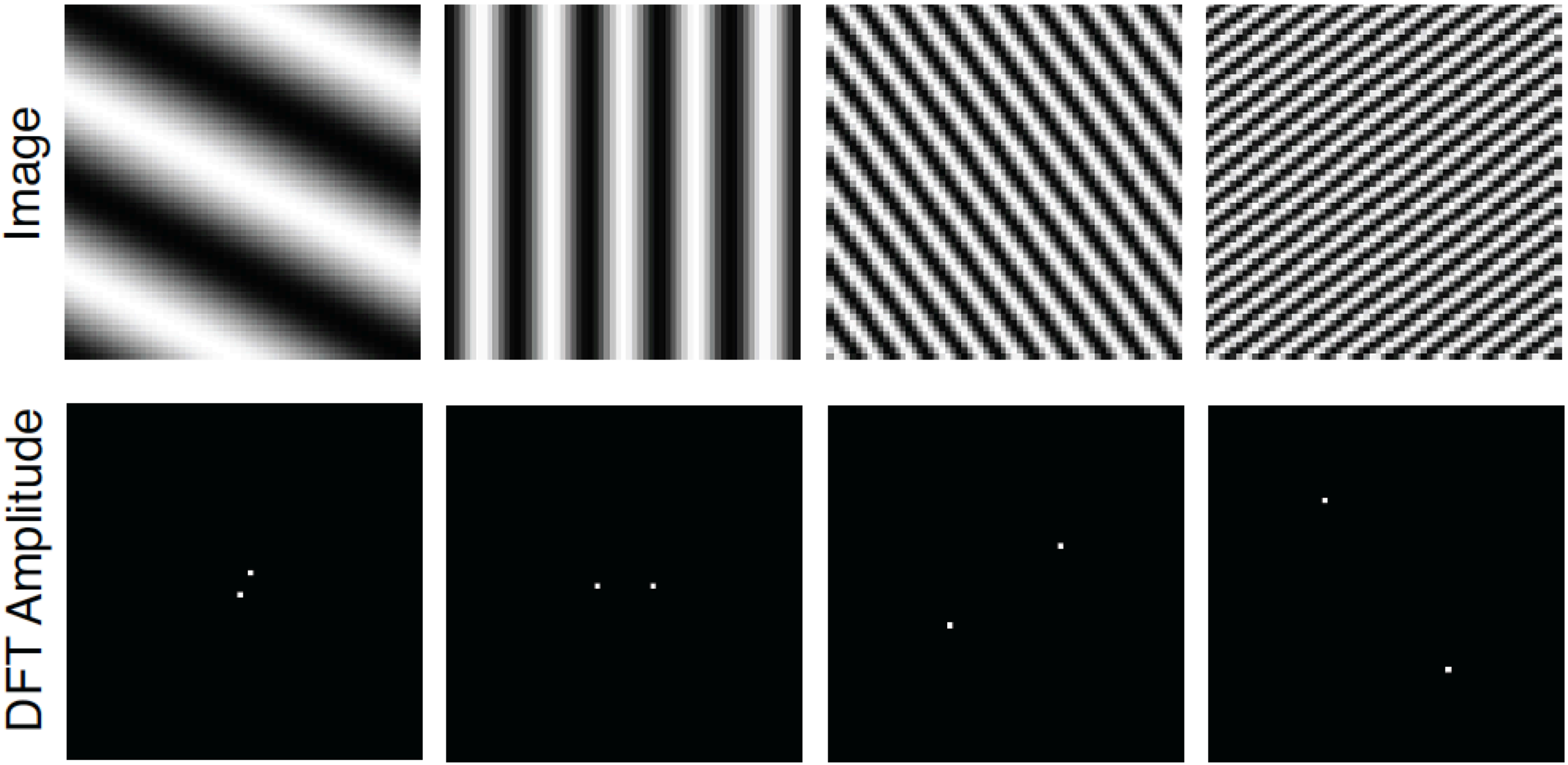


Image
(assuming zero phase)

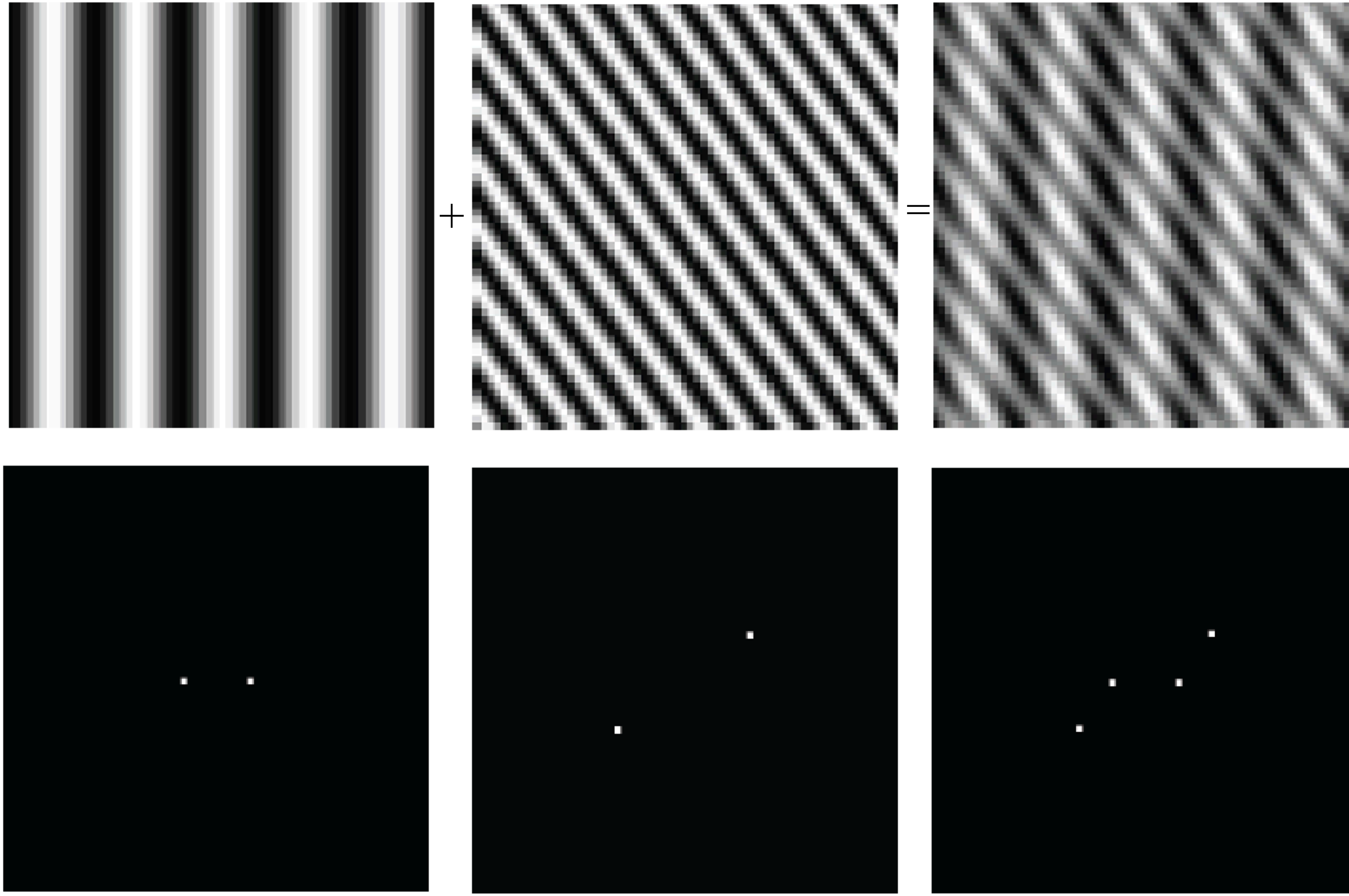


Images are 64x64 pixels. The wave is a cosine (if phase is zero).

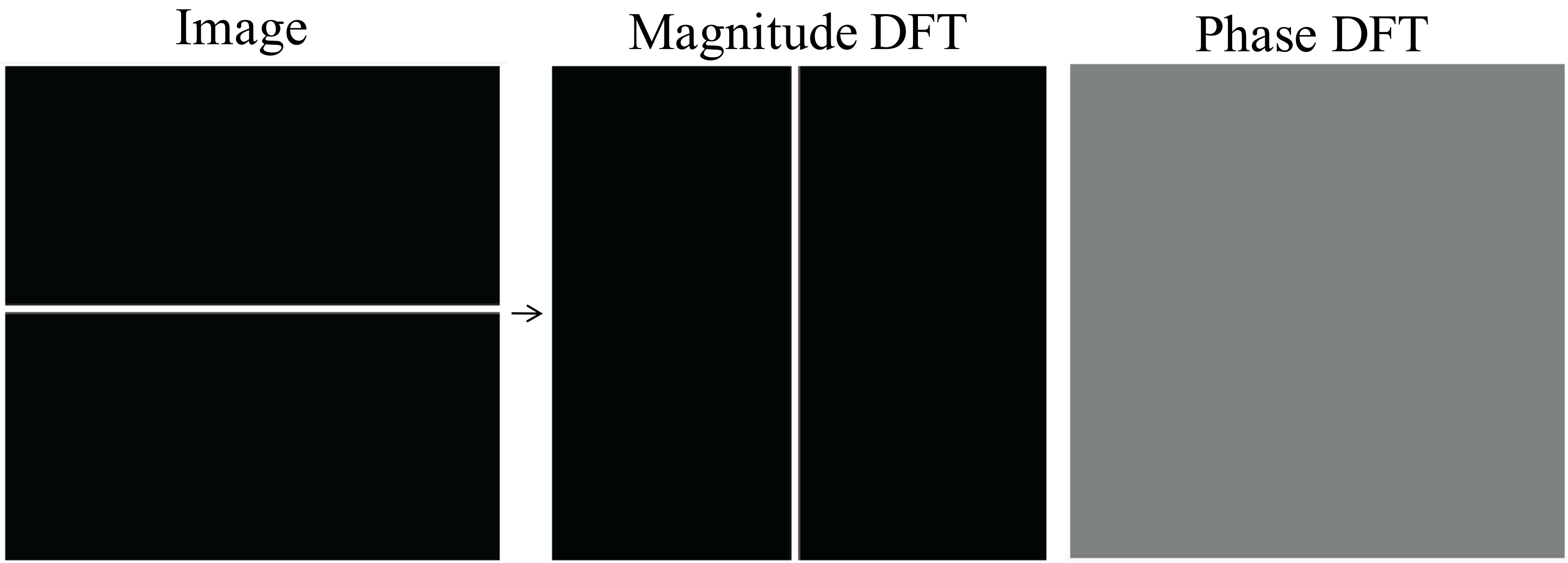
Frequencies



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

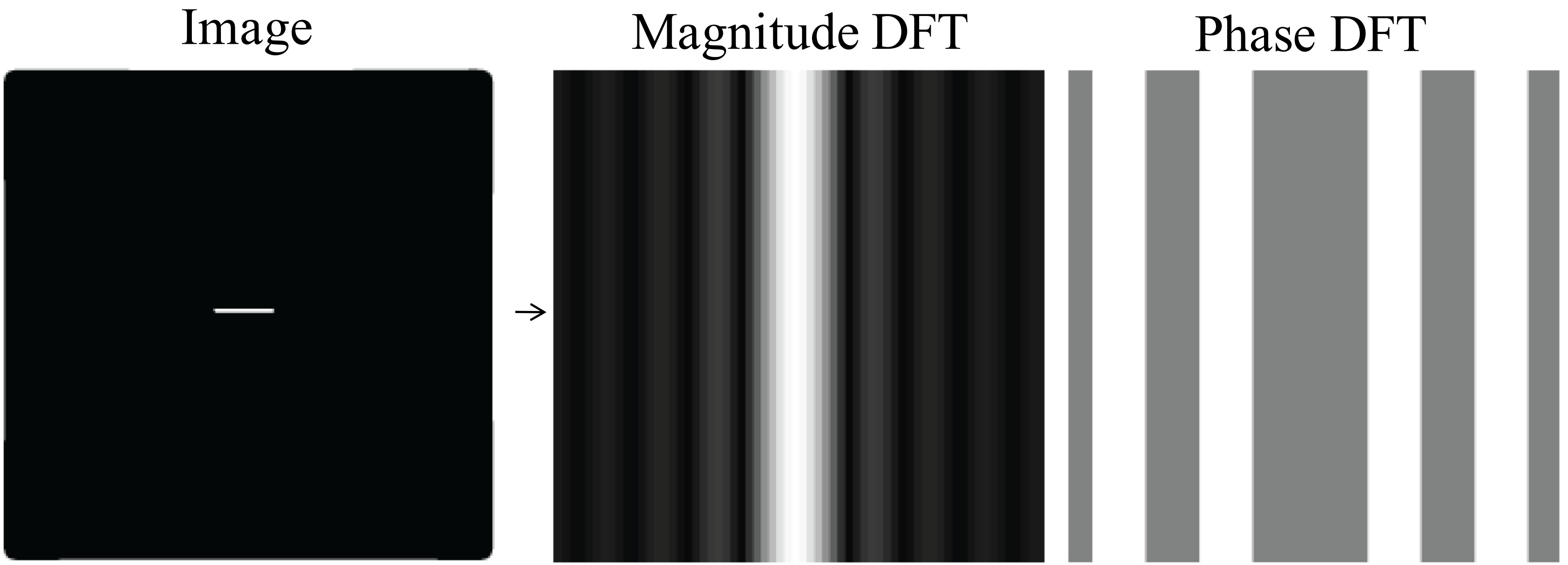


Some important Fourier transforms

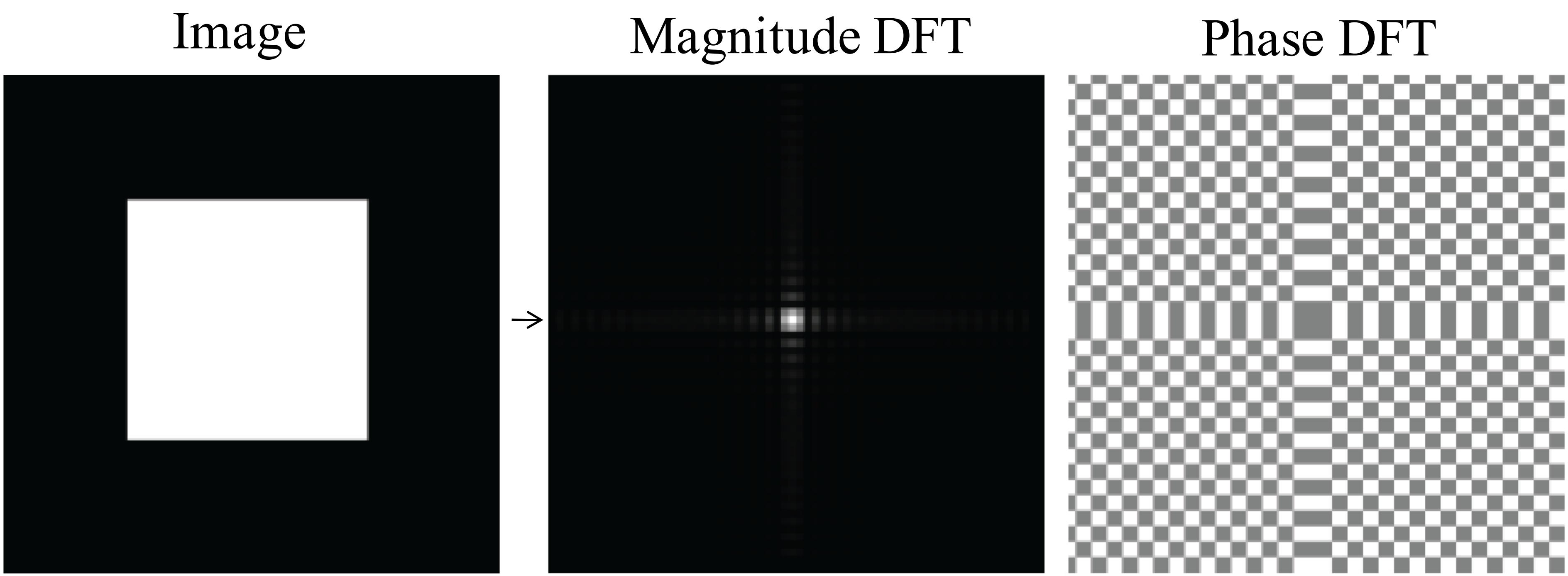


Images are 64x64 pixels.

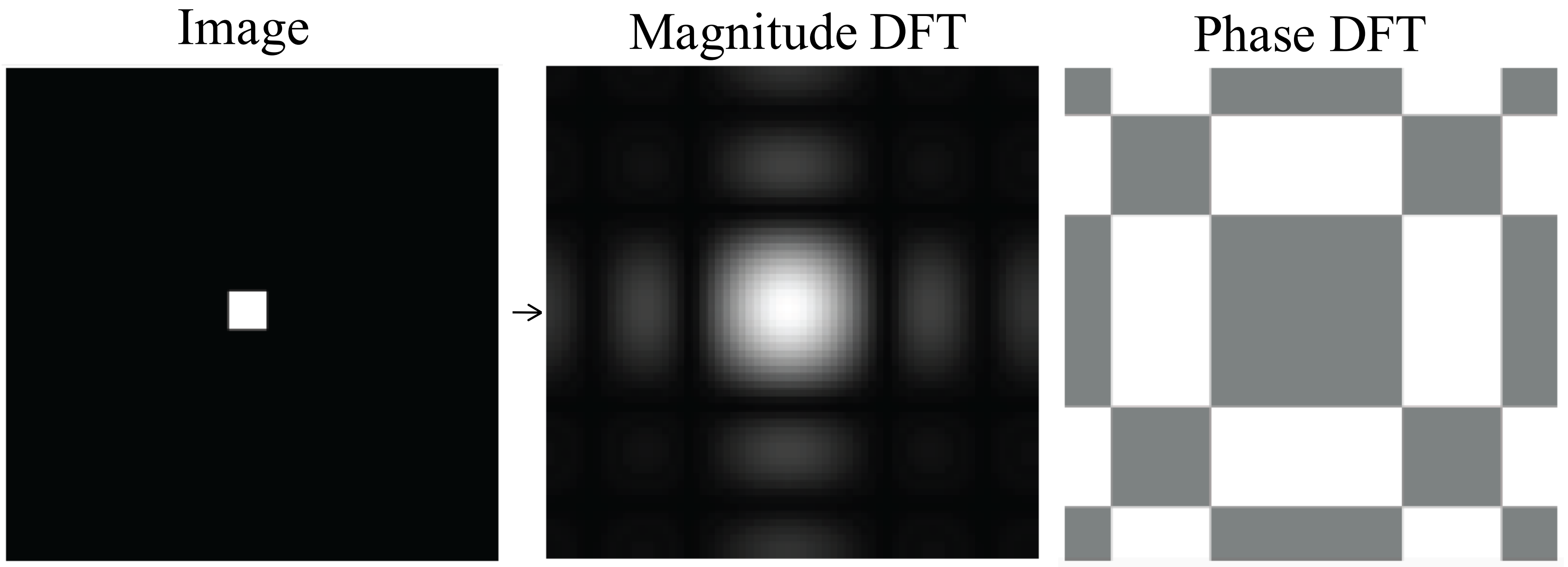
Some important Fourier transforms

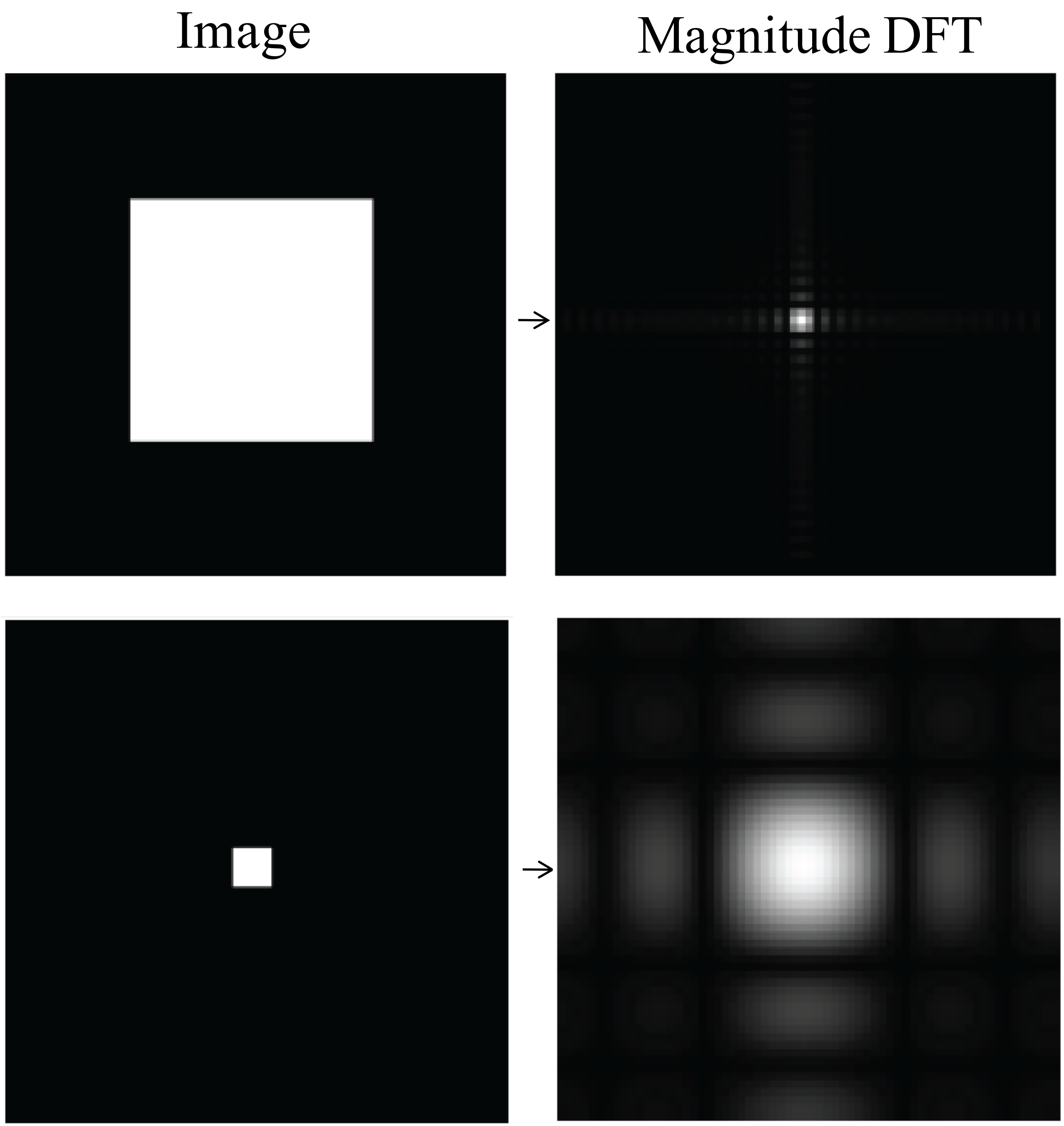


Some important Fourier transforms



Some important Fourier transforms

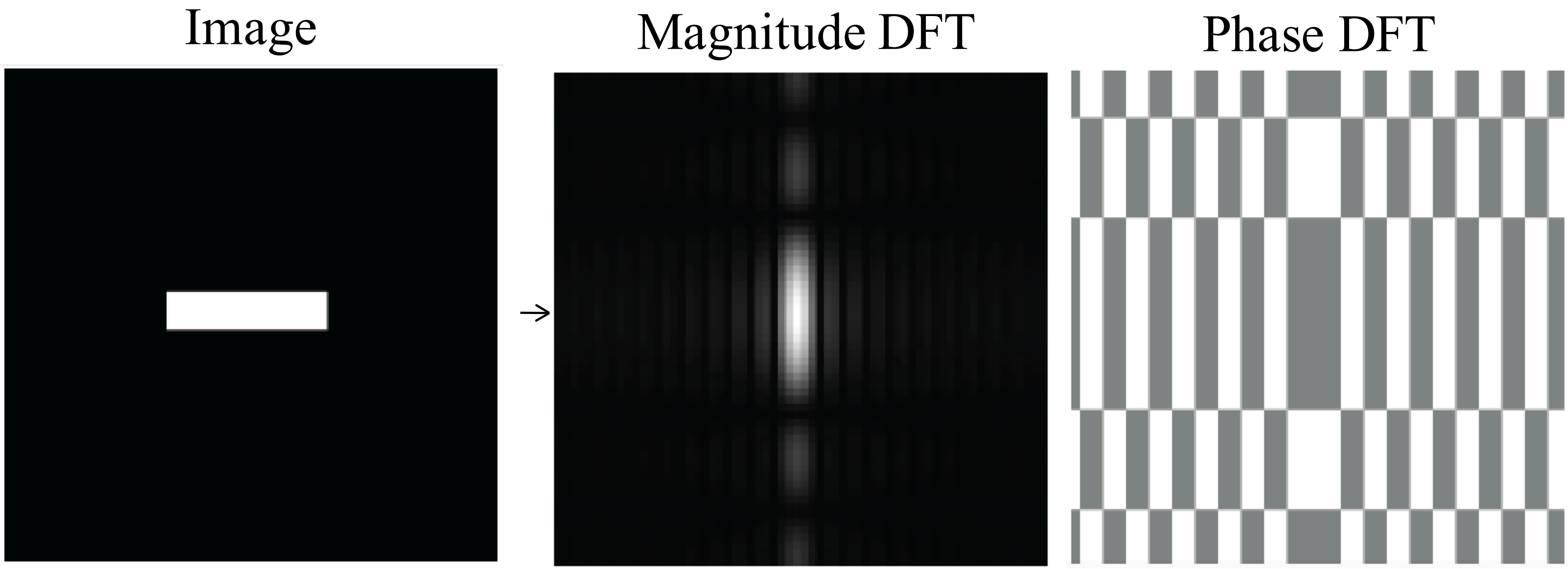




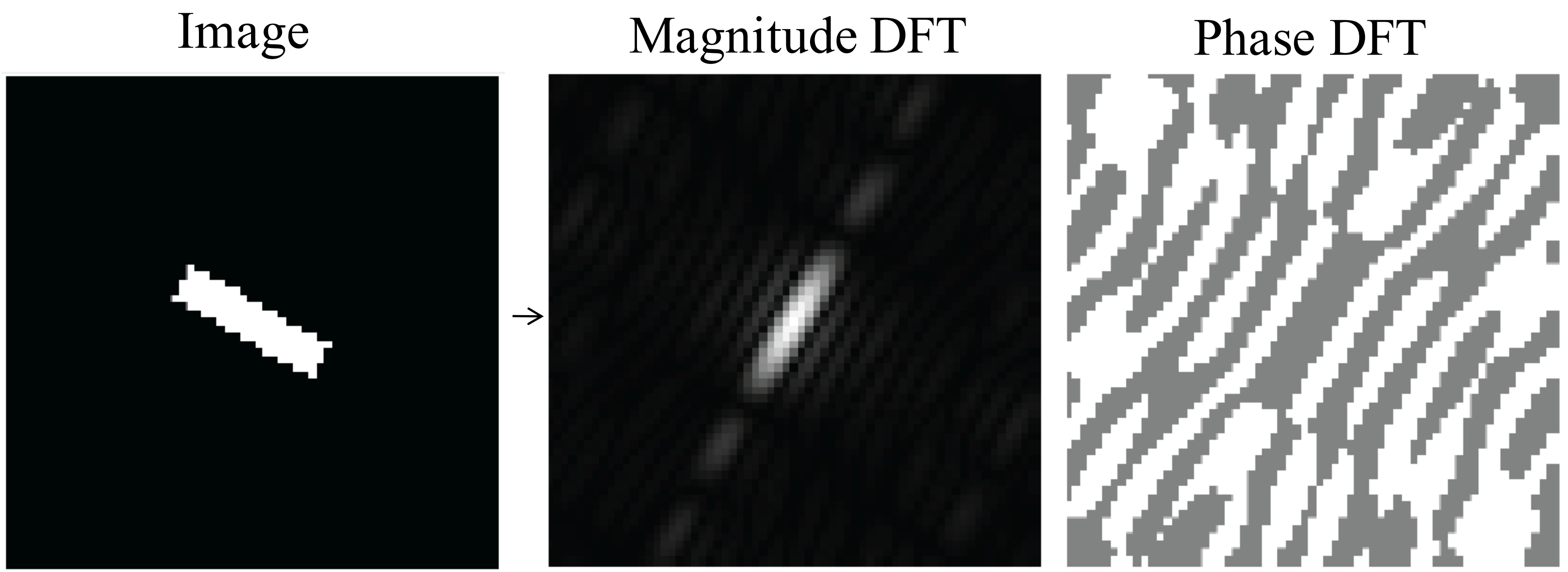
Scale

Small image details produce content in high spatial frequencies

Some important Fourier transforms

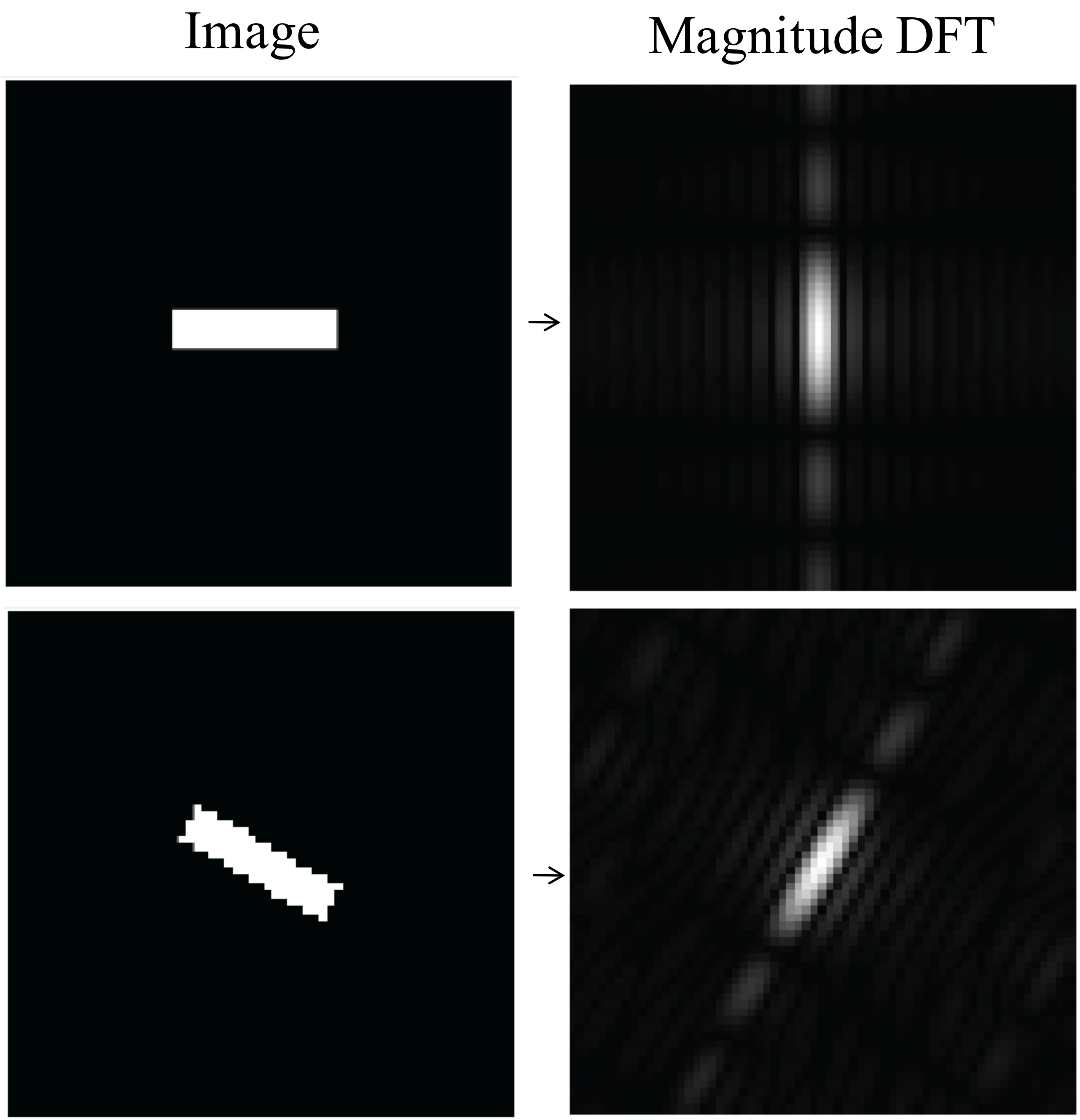


Some important Fourier transforms

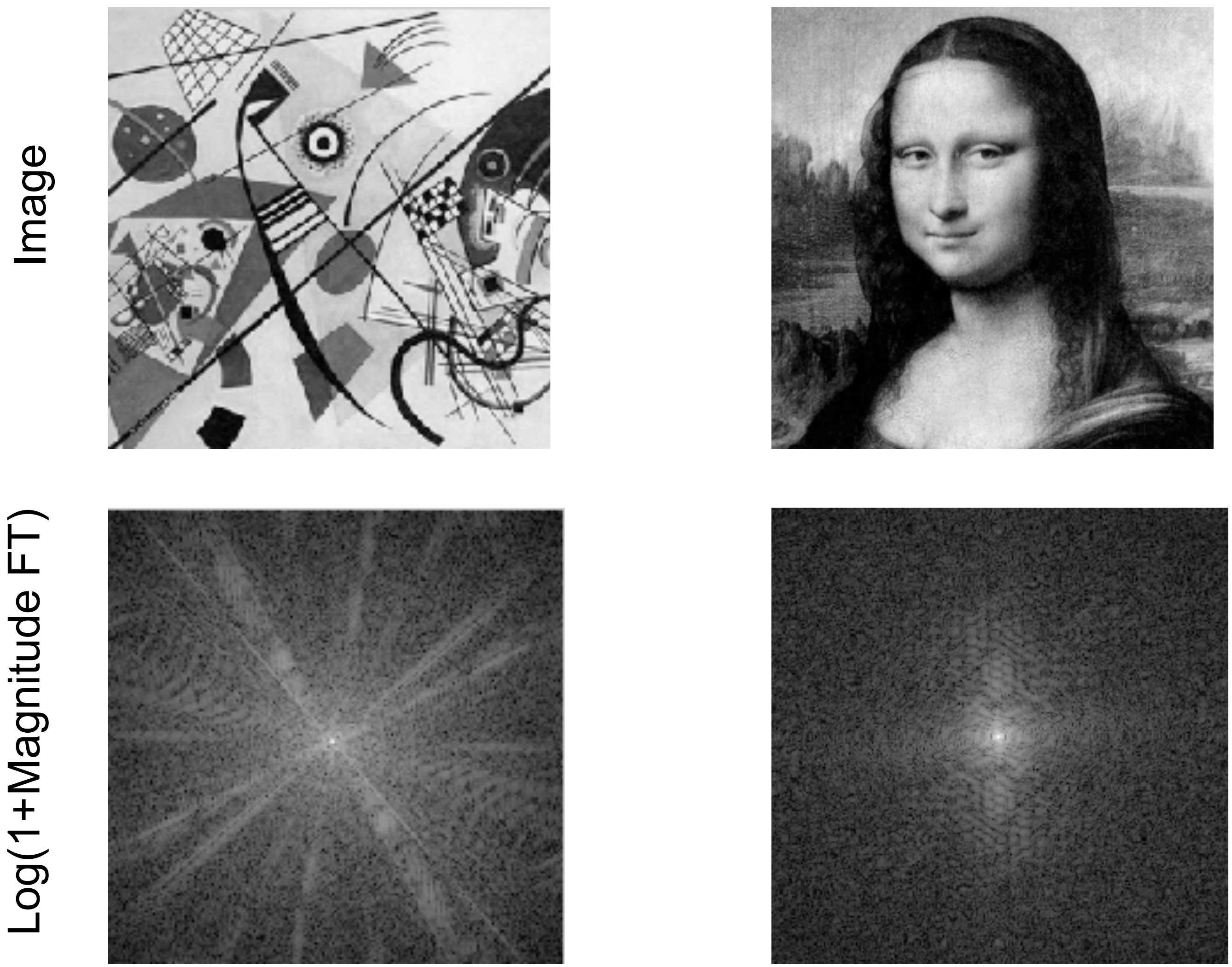


Orientation

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



The Fourier Transform of some important images



More properties for the DFT

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

DFT of the convolution

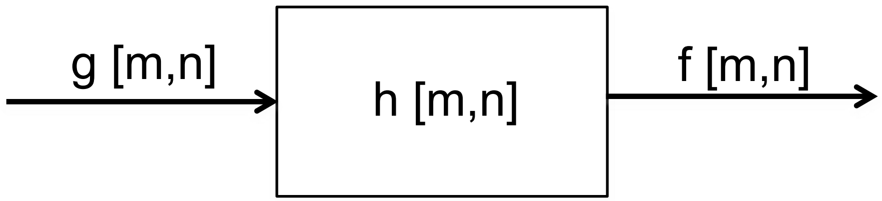
$$f = g \circ h \longleftrightarrow F[u,v] = G[u,v]\,H[u,v]$$

$$\begin{aligned}F[u,v] &= DFT\{g \circ h\} \\ &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\boxed{\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} g[m-k,n-l]\,h[k,l]}\exp\left(-2\pi j\left(\frac{mu}{M}+\frac{nv}{N}\right)\right)\end{aligned}$$

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

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

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:

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

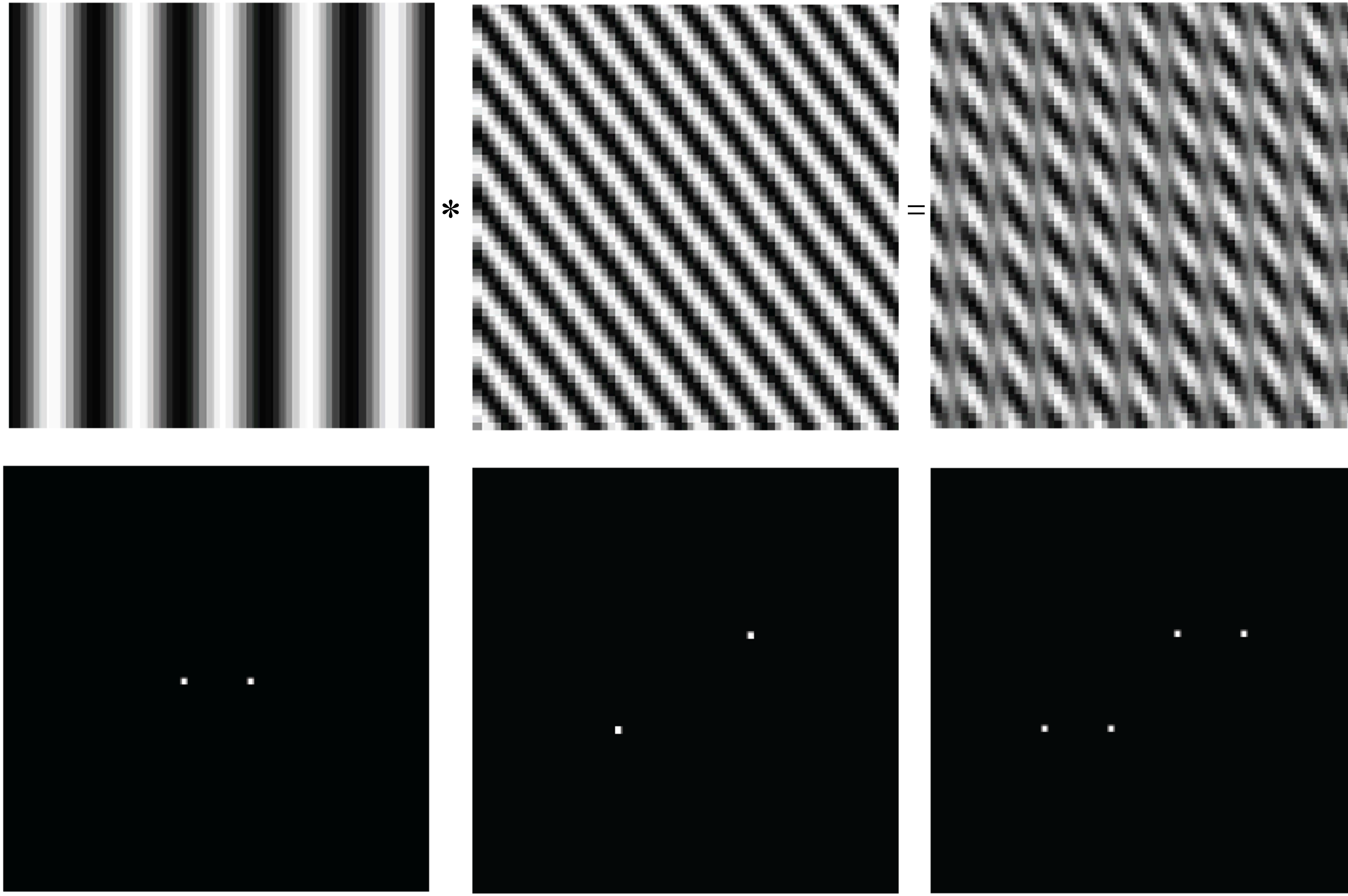
Product of images

The Fourier transform of the product of two images

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

is the convolution of their DFTs:

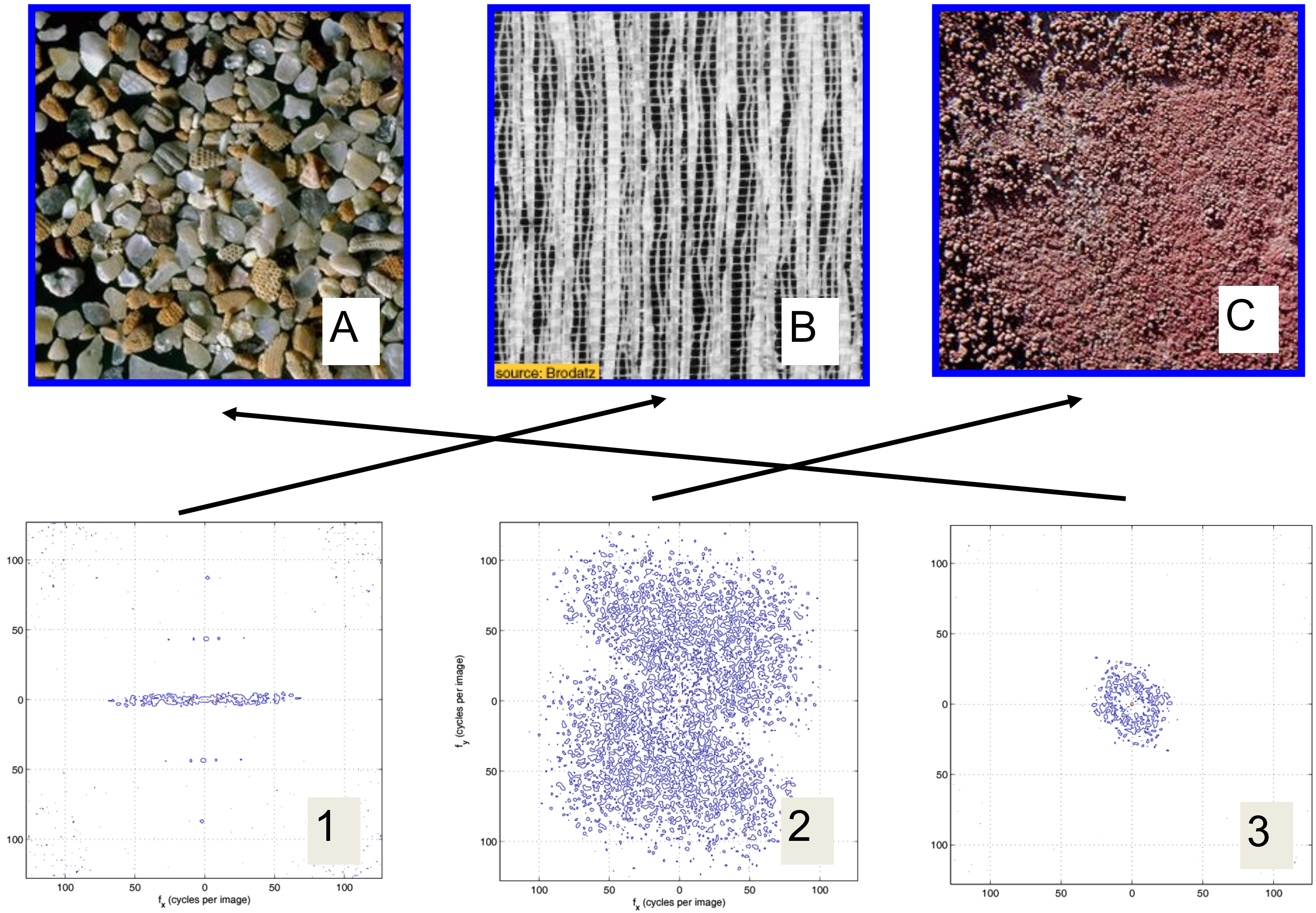
$$F[u,v] = \frac{1}{NM} G[u,v] \circ H[u,v]$$



Game: find the right pairs

Images

DFT magnitude



fx(cycles/image pixel size)

fx(cycles/image pixel size)

fx(cycles/image pixel size)