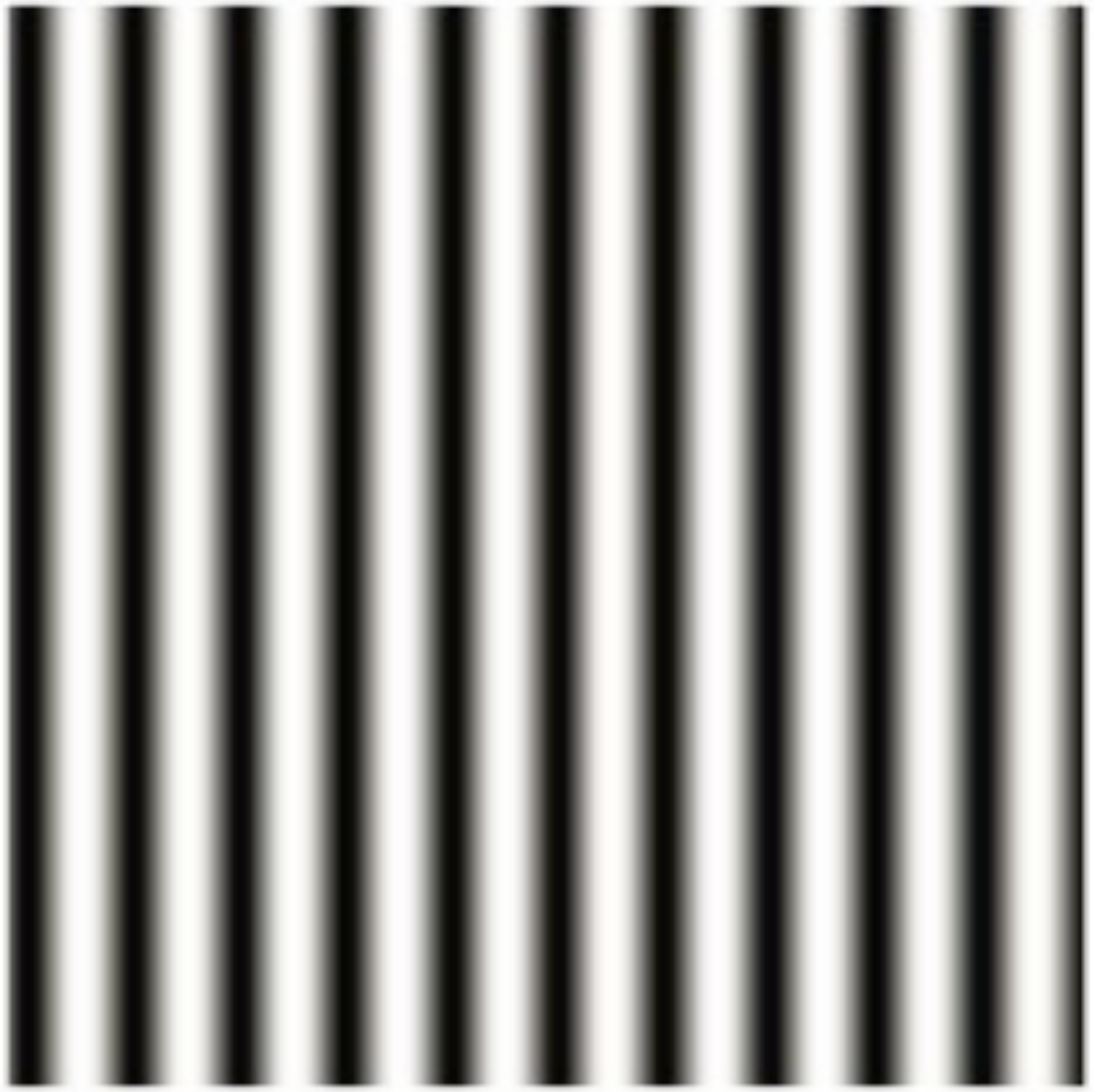
Lecture 4

Spatial and Temporal Linear Filters

6.8300/6.8301 Advances in Computer Vision

Spring 2023

Vincent Sitzmann, Mina Lukovic Konacovic, Bill Freeman



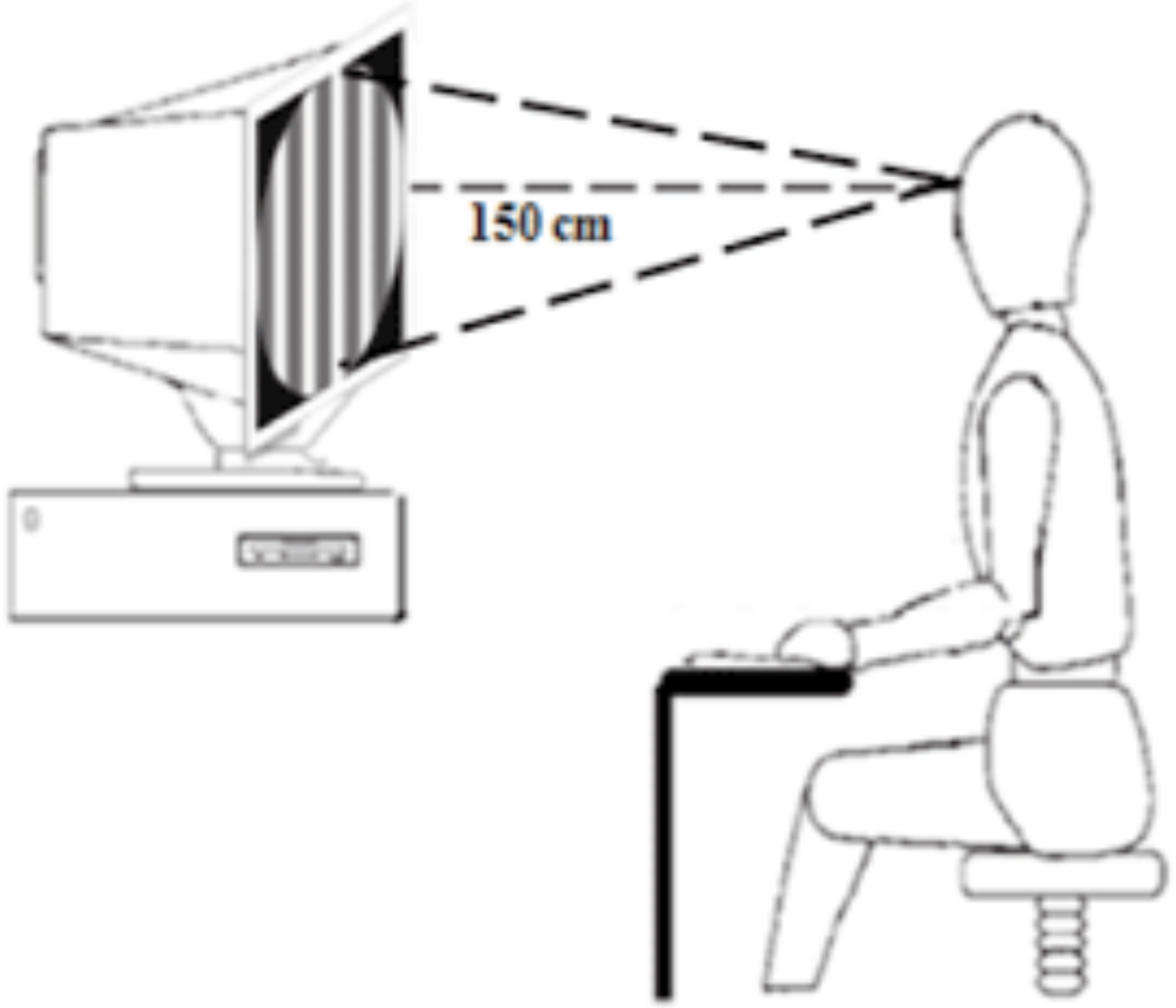
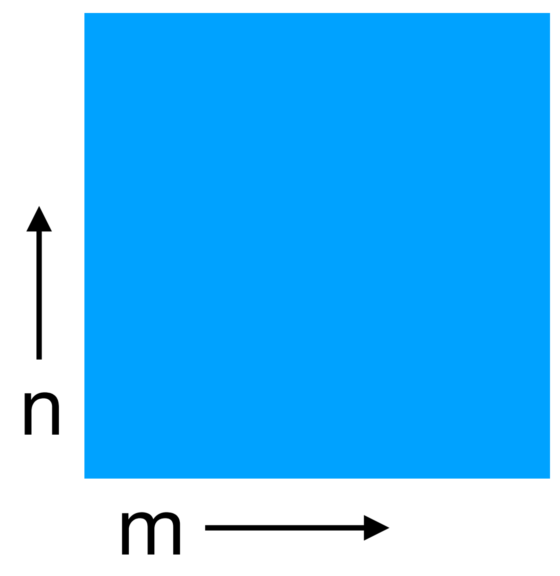


Figure 1. Stimulus presentation scheme. The stimuli were originally calibrated to be seen at a distance of 150 cm in a 19″ display.

Campbell & Robson chart

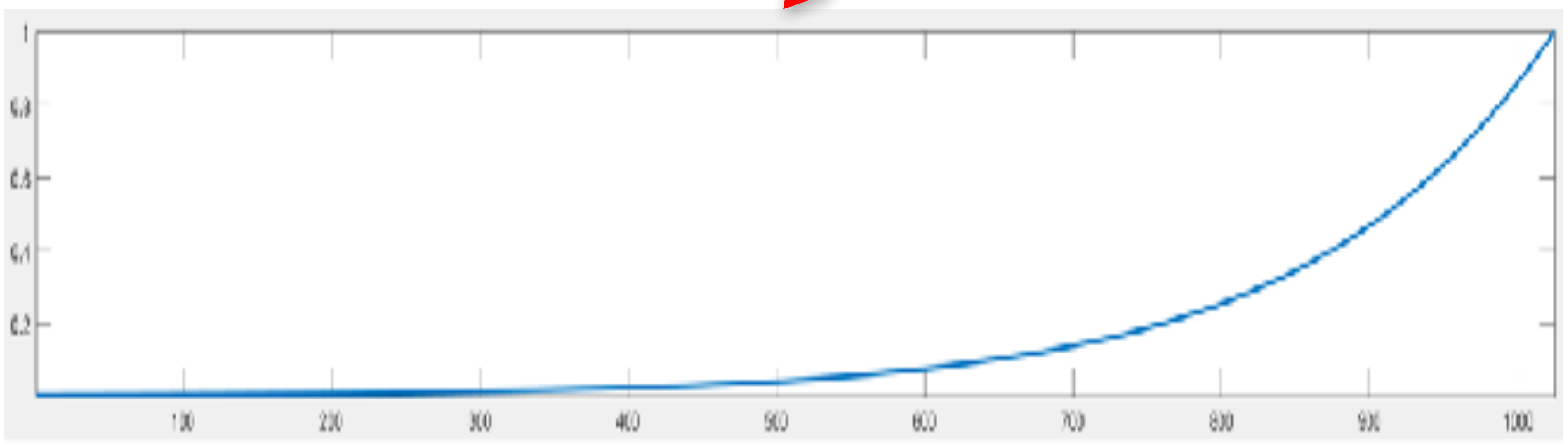


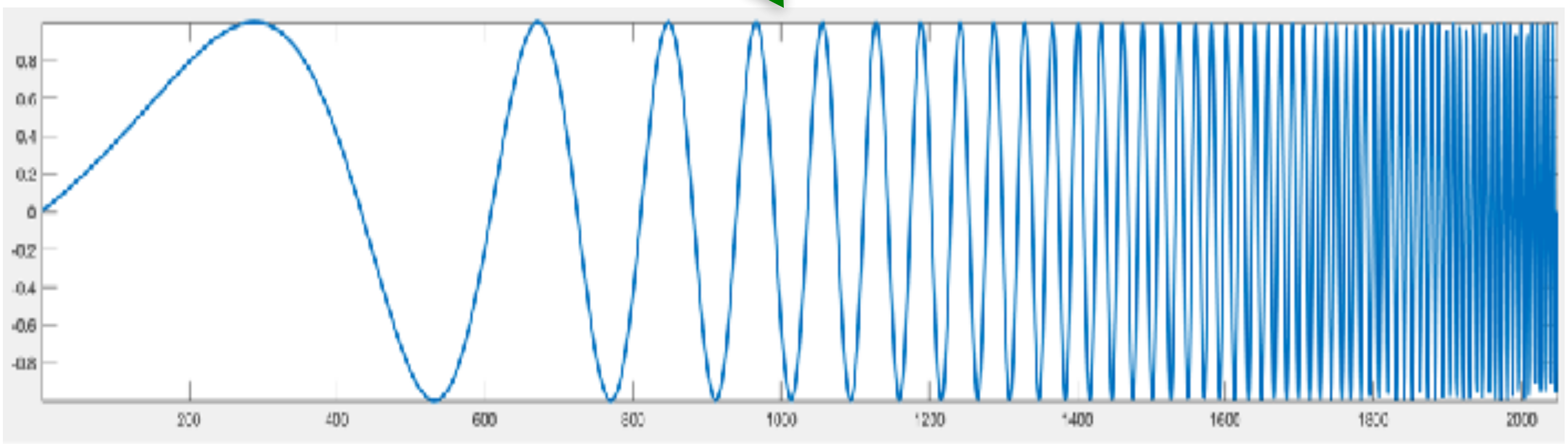
Let's define the following image:

$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\, m/M)$$

With:

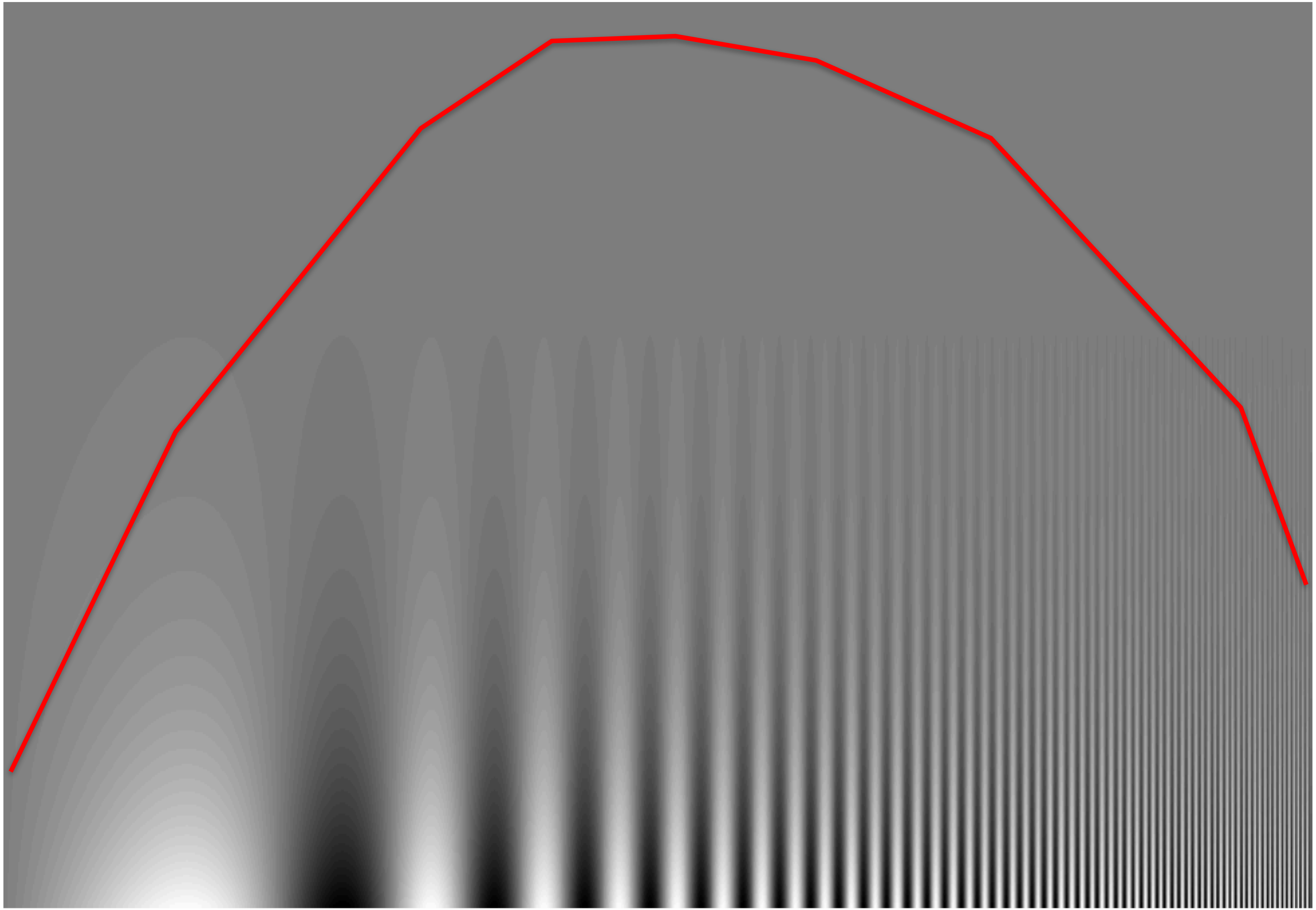
$$A[n] = A_{min}\left(\frac{A_{max}}{A_{min}}\right)^{n/N} \qquad f[m] = f_{min}\left(\frac{f_{max}}{f_{min}}\right)^{m/M}$$





What do you think you should see when looking at this image?

$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\,m/M)$$

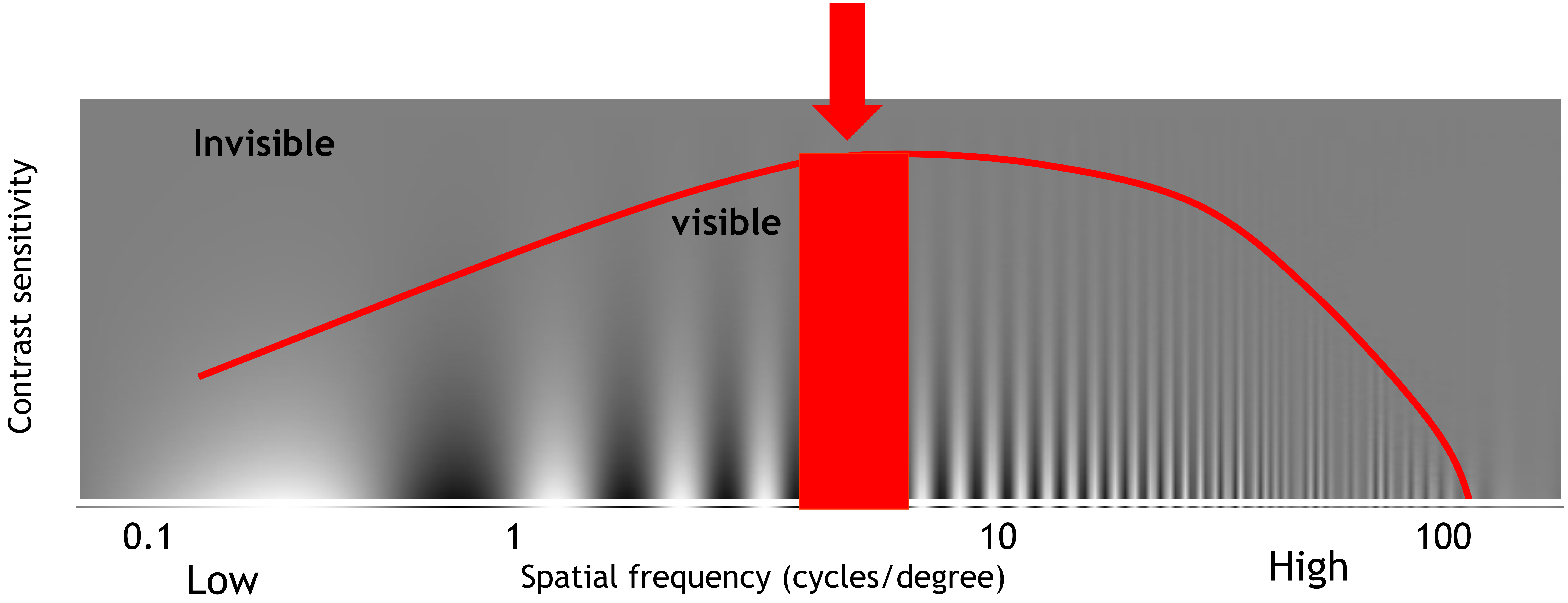


Contrast Sensitivity Function

Blackmore & Campbell (1969)

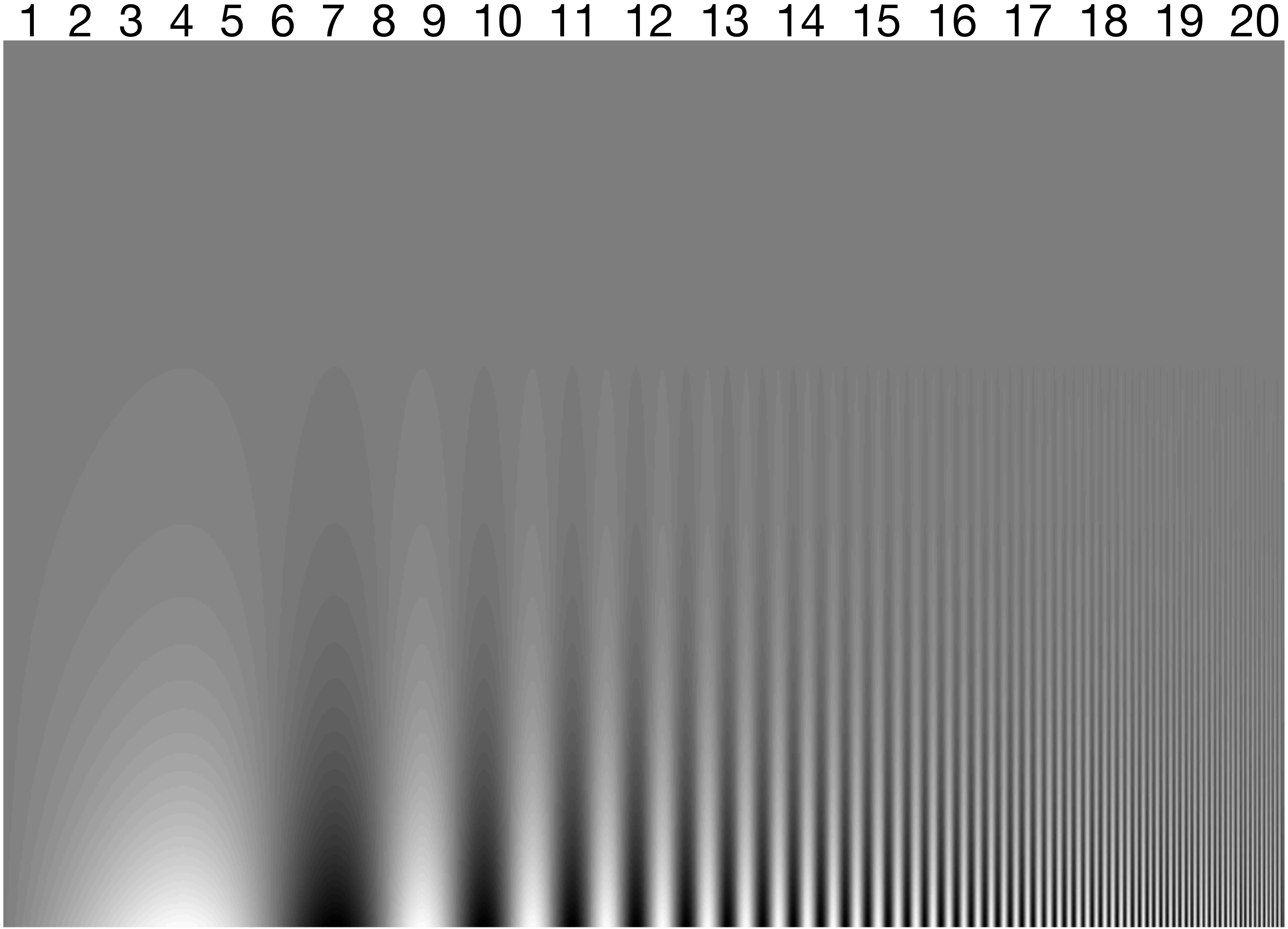
Maximum sensitivity

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

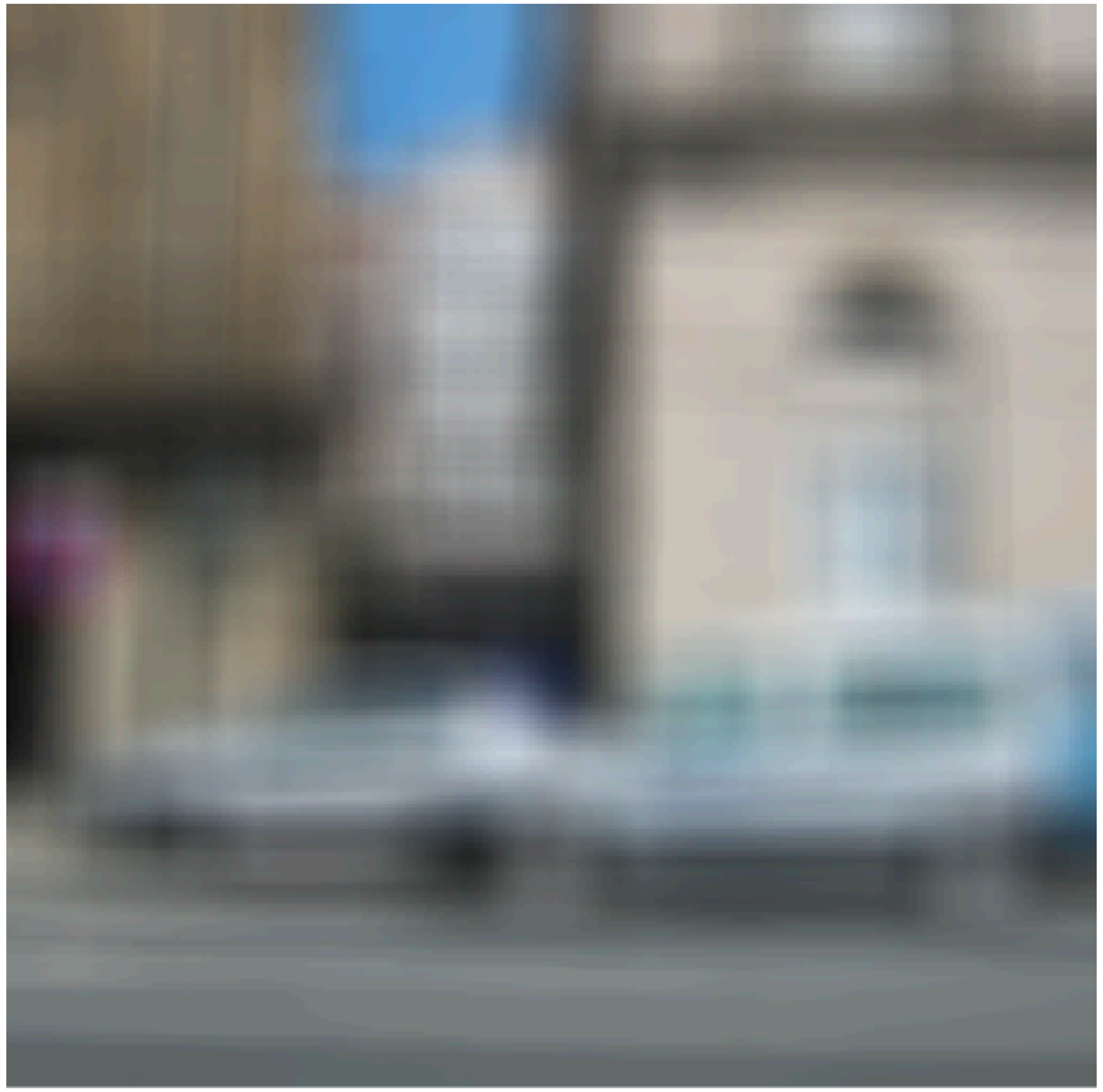


Things that are very close and/or large are hard to see

Things far away are hard to see


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Today: A collection of useful filters in space and time, and aliasing.



Low-pass filters

Band-pass filters

Low pass-filters

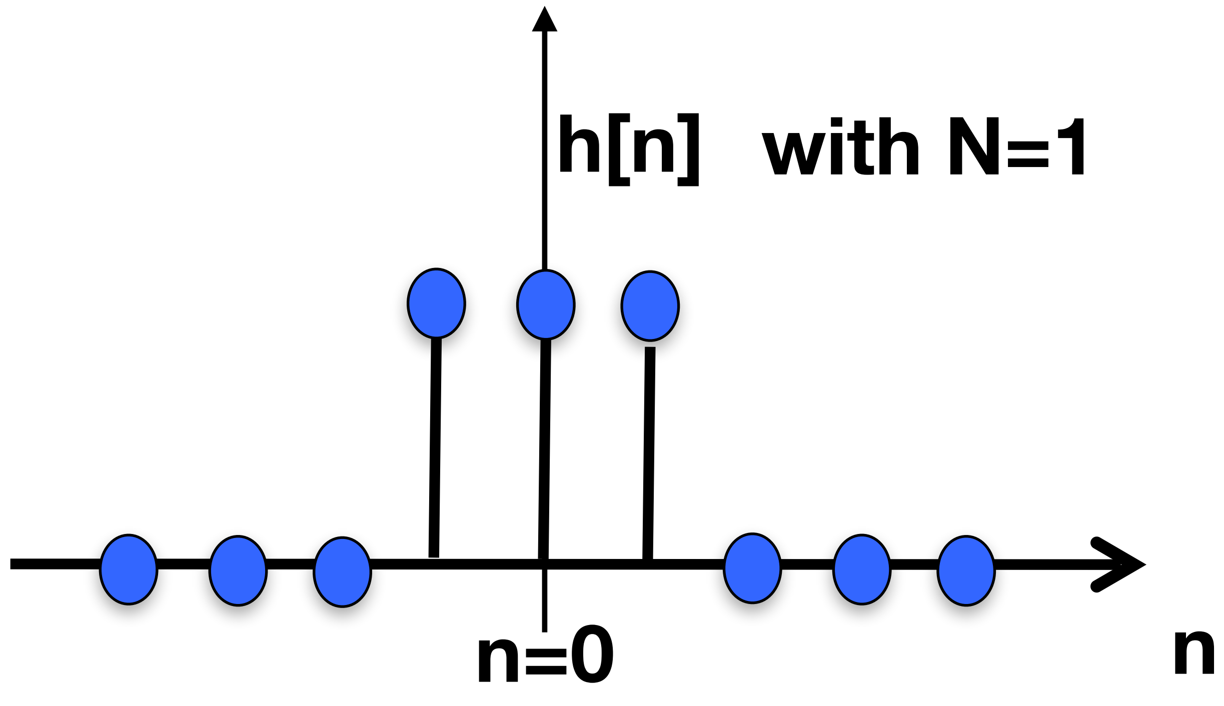
Box filter

2N+1

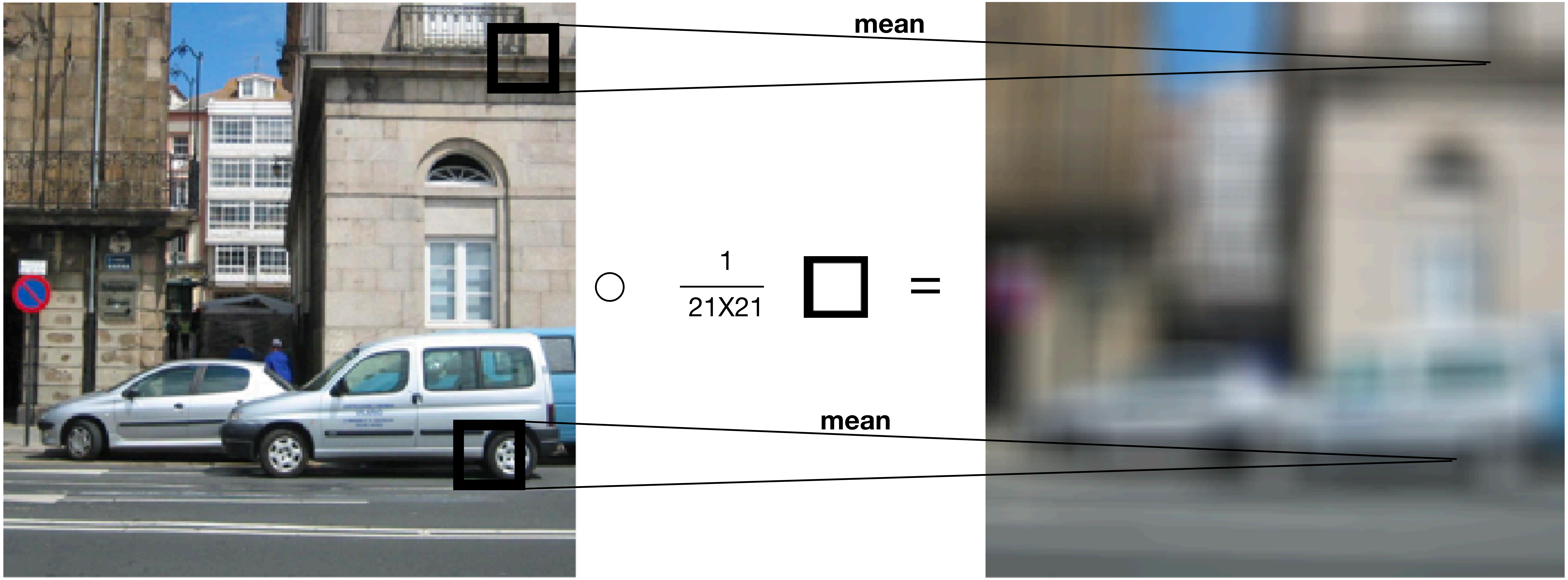
1	1	…	1
1	1		1
1	1		1
…			
1	1	1	1

2M+1

$$h_{N,M}[n,m] = \begin{cases} 1 & \text{if } -N \leq n \leq N \text{ and } -M \leq m \leq M \\ 0 & \text{otherwise} \end{cases}$$



Box filter



256X256 **256X256**

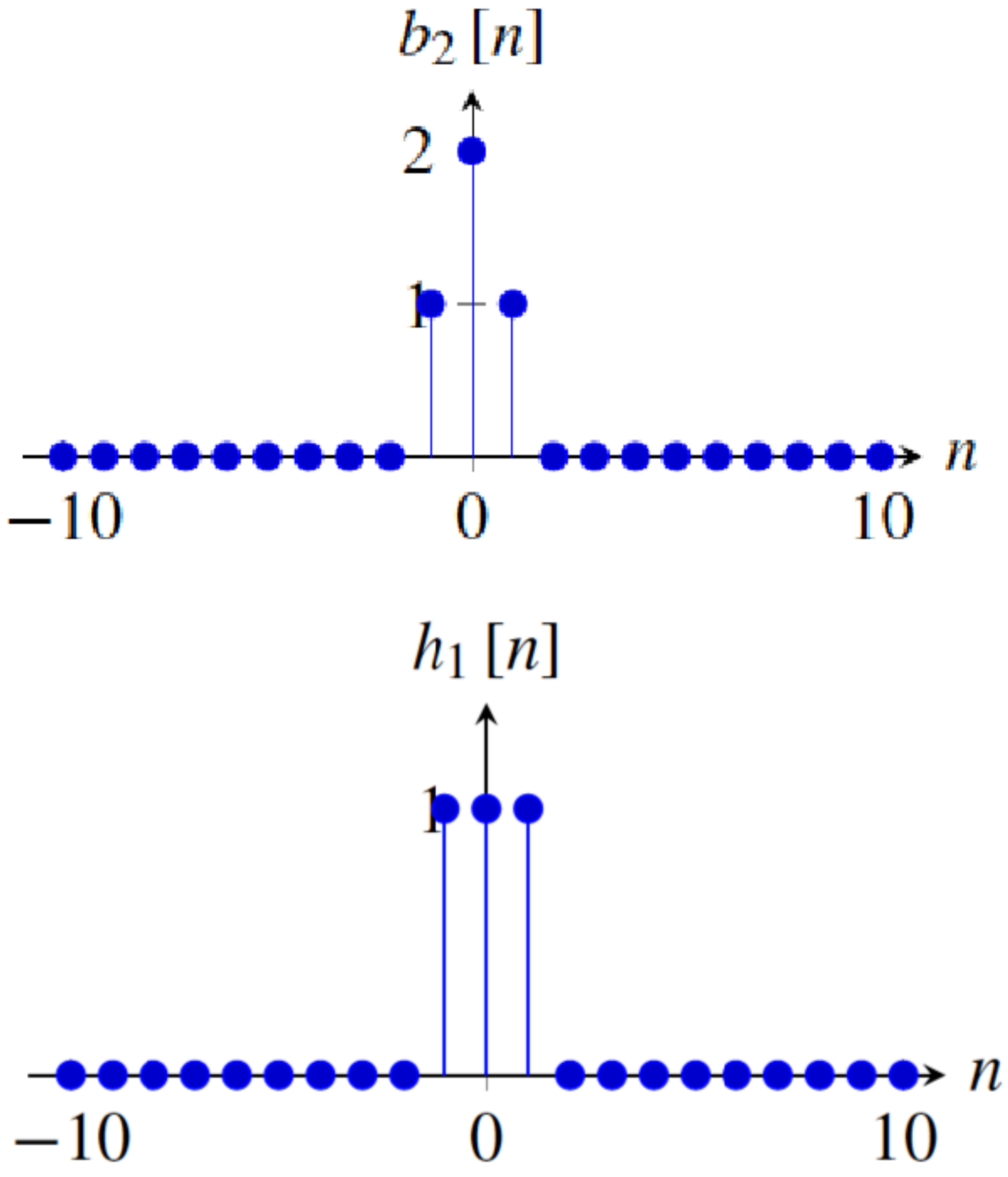
What does it do?

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

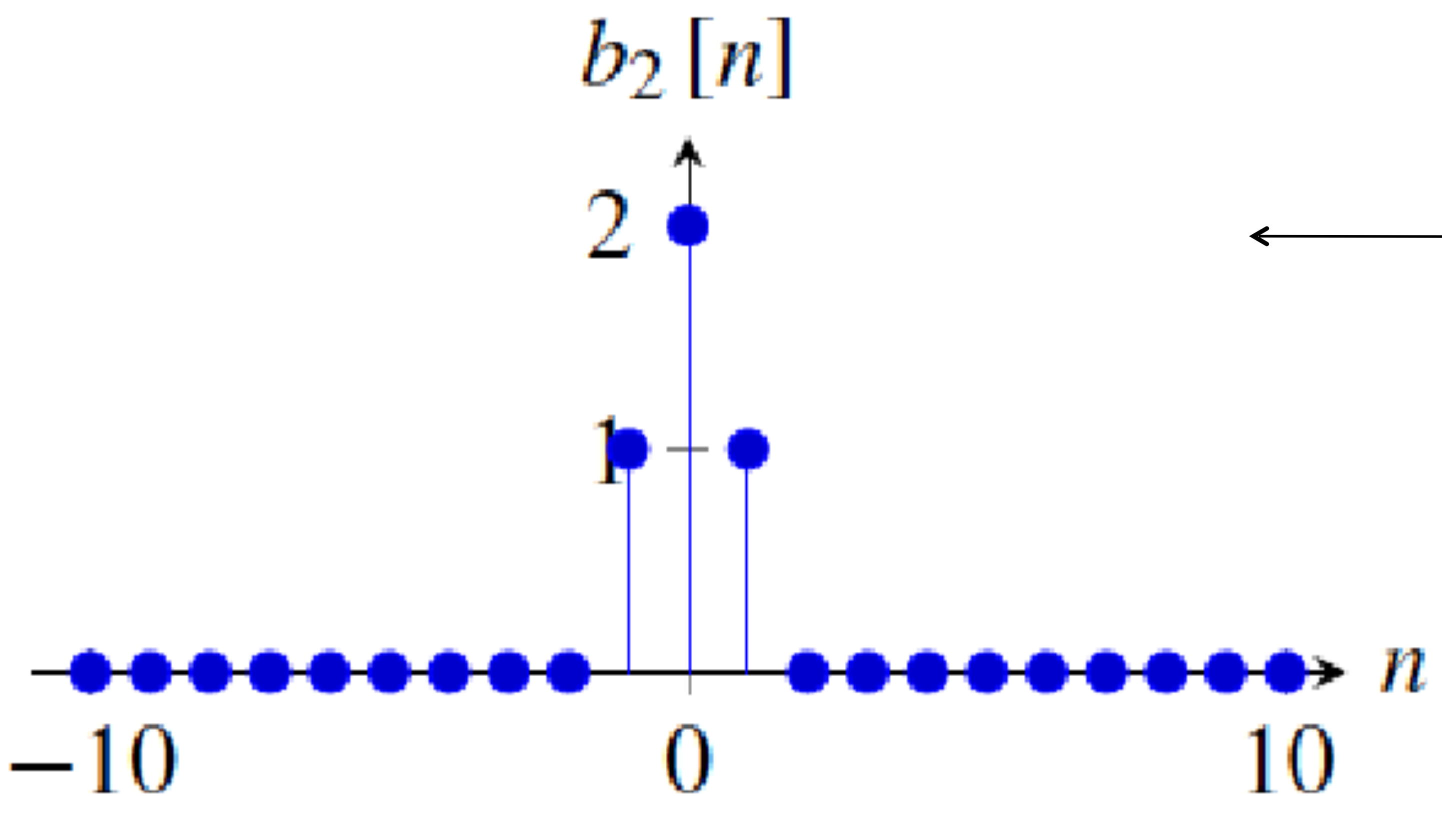
b2[n] versus the 3-tap box filter

[1 2 1]

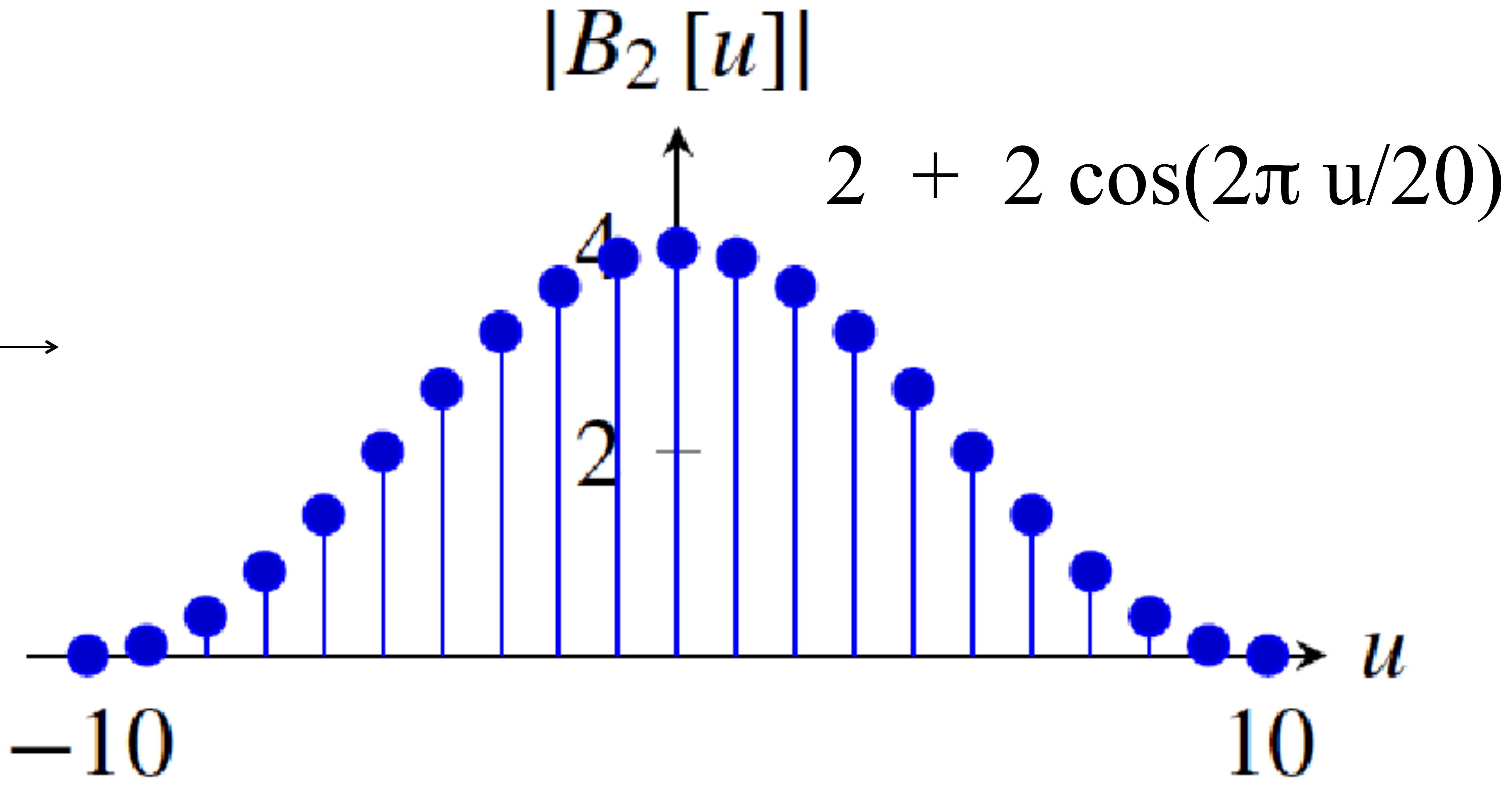
[1 1 1]



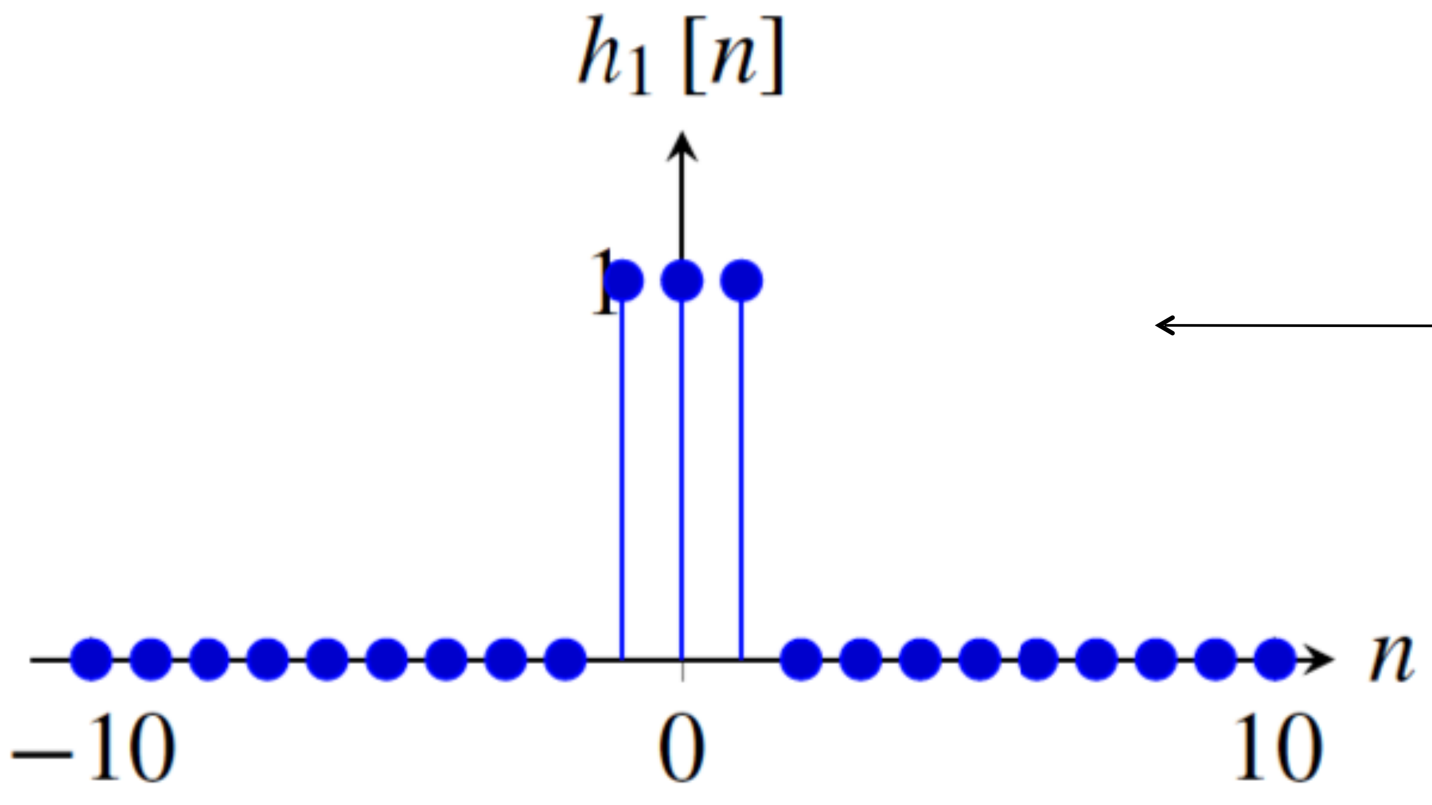
Which one is a better low-pass filter?



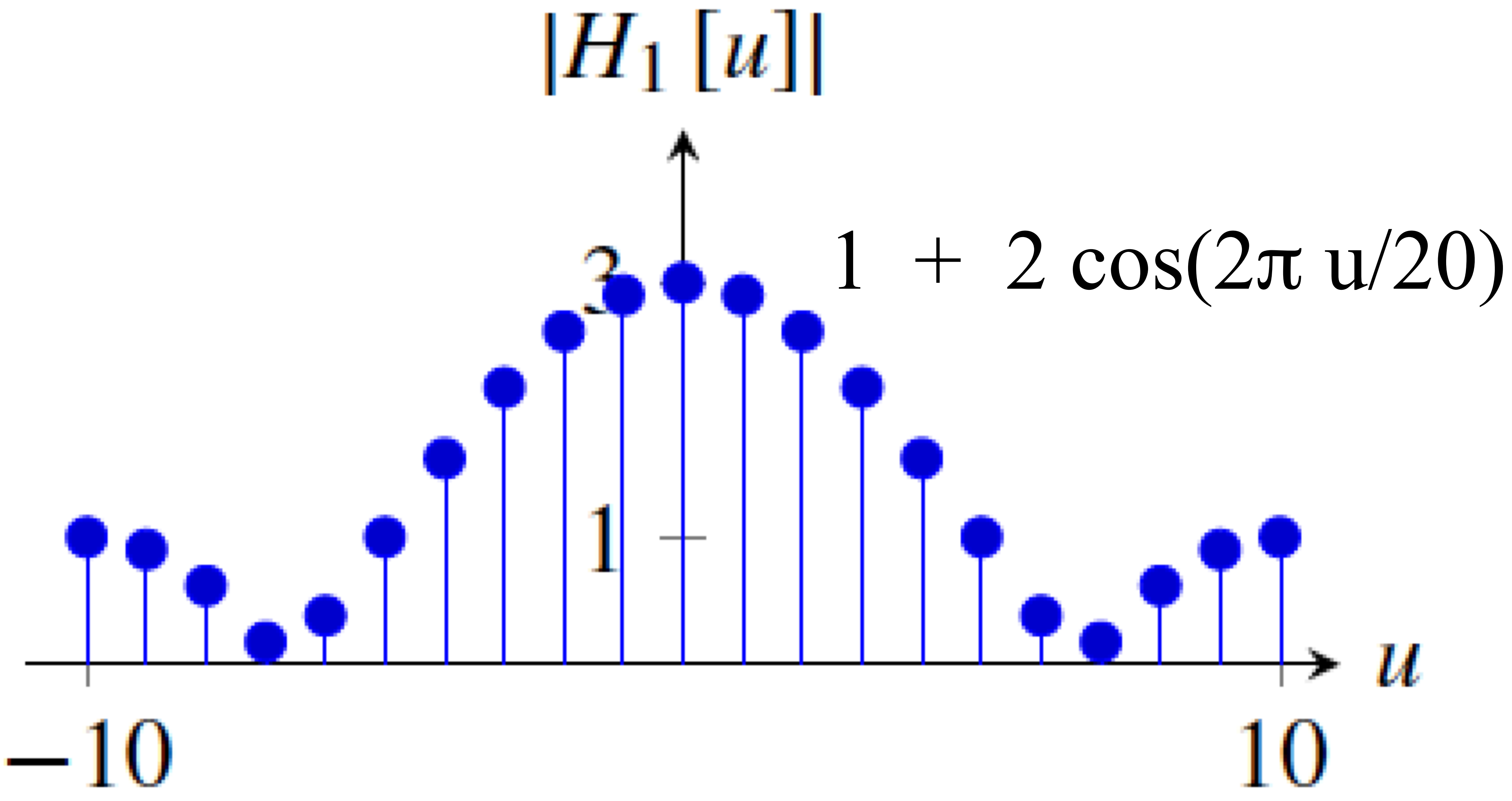
$b_2[n]$
2
1
n
−10
0
10



$|B_2[u]|$
2 + 2 cos(2π u/20)
4
2
u
−10
10



$h_1[n]$
1
n
−10
0
10



$|H_1[u]|$
1 + 2 cos(2π u/20)
3
1
u
−10
10

b2[n] vs h1[n]

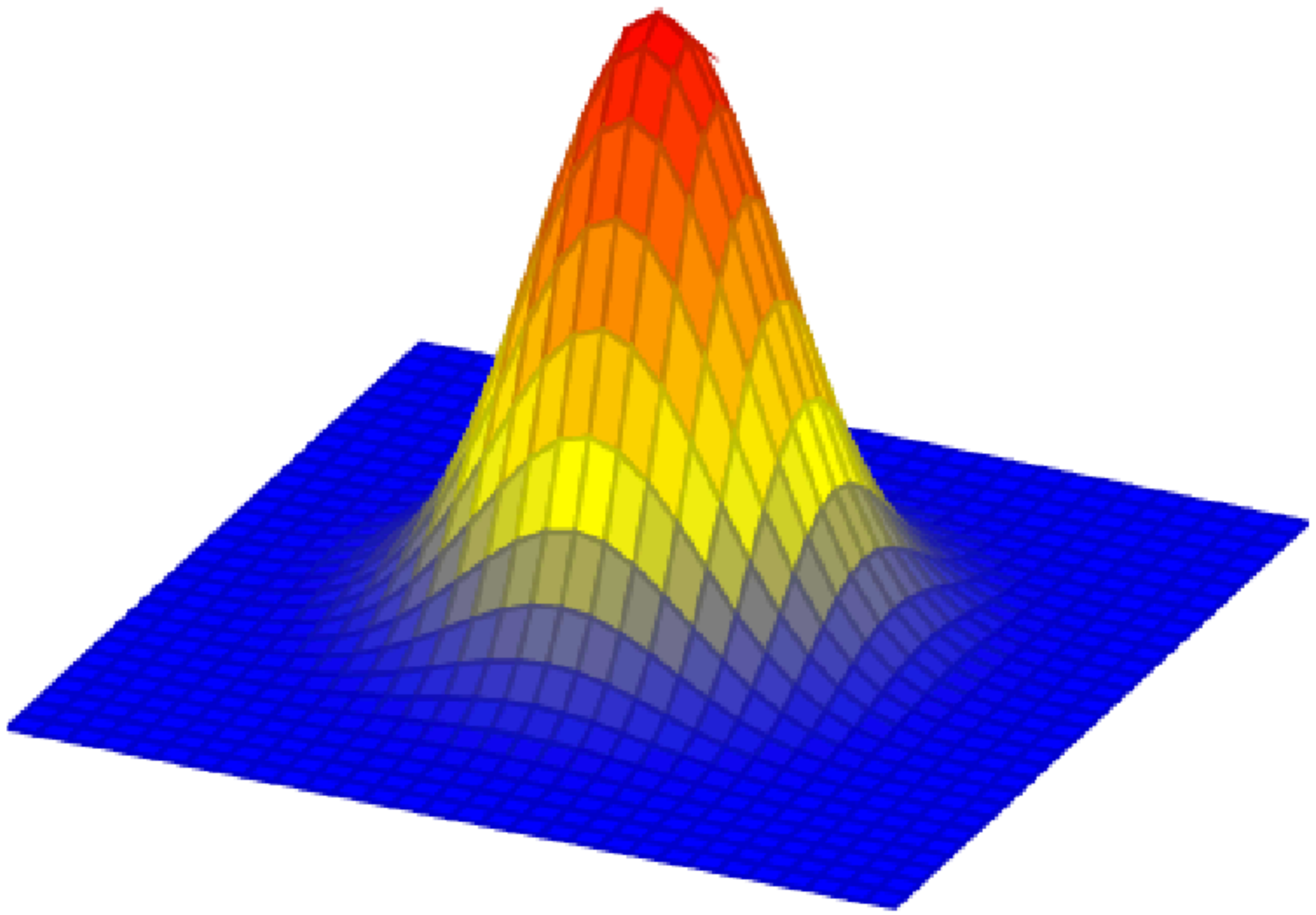
$$[1, 1, 1] \circ [\ldots, 1, -1, 1, -1, 1, -1, \ldots] = [\ldots, -1, 1, -1, 1, -1, 1, \ldots]$$

$$[1, 2, 1] \circ [\ldots, 1, -1, 1, -1, 1, -1, \ldots] = [\ldots, 0, 0, 0, 0, 0, 0, \ldots]$$

Gaussian filter

In the continuous domain:

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



Gaussian filter

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

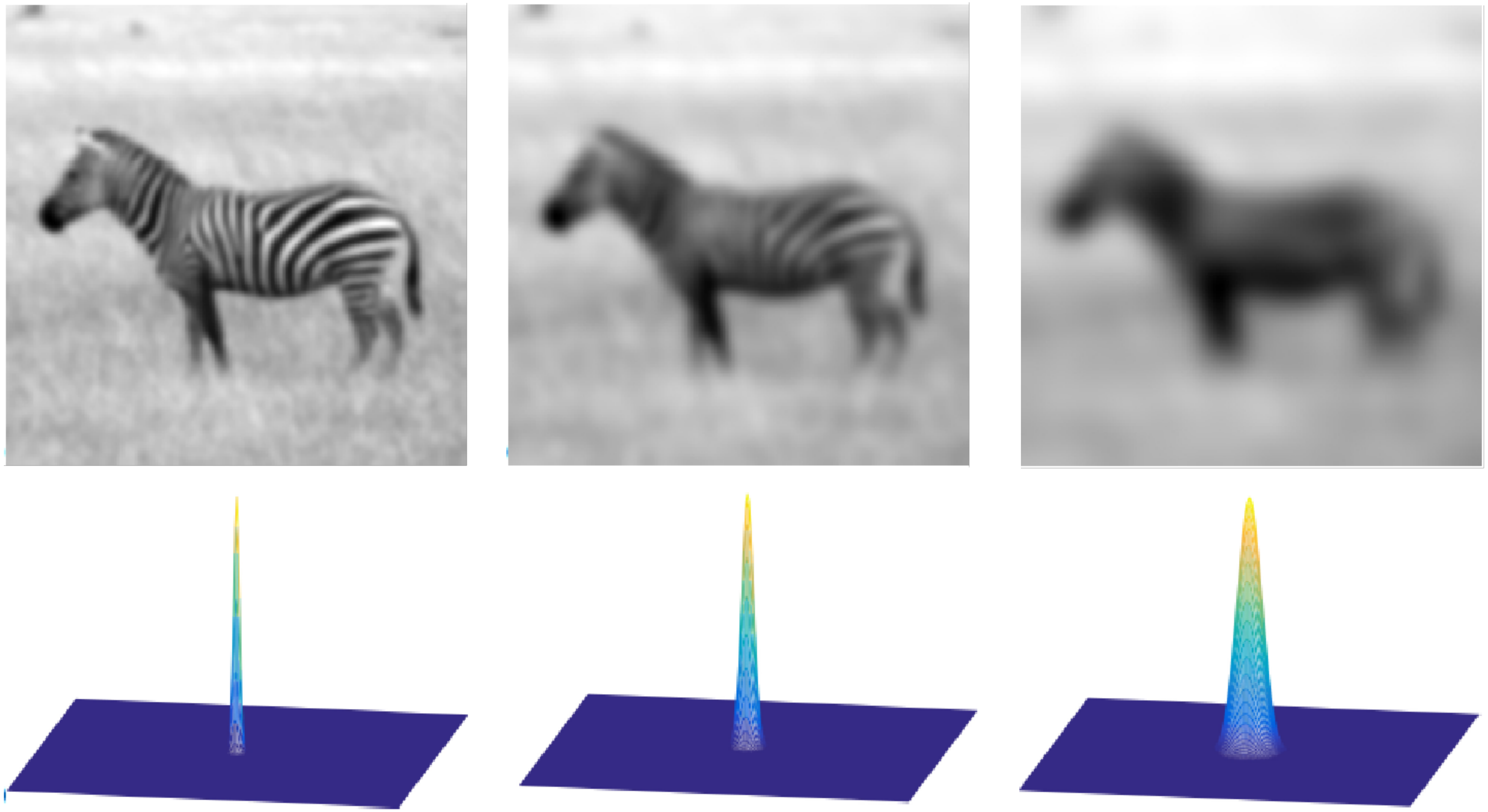
Discretization of the Gaussian:

At 3σ the amplitude of the Gaussian is around 1% of its central value

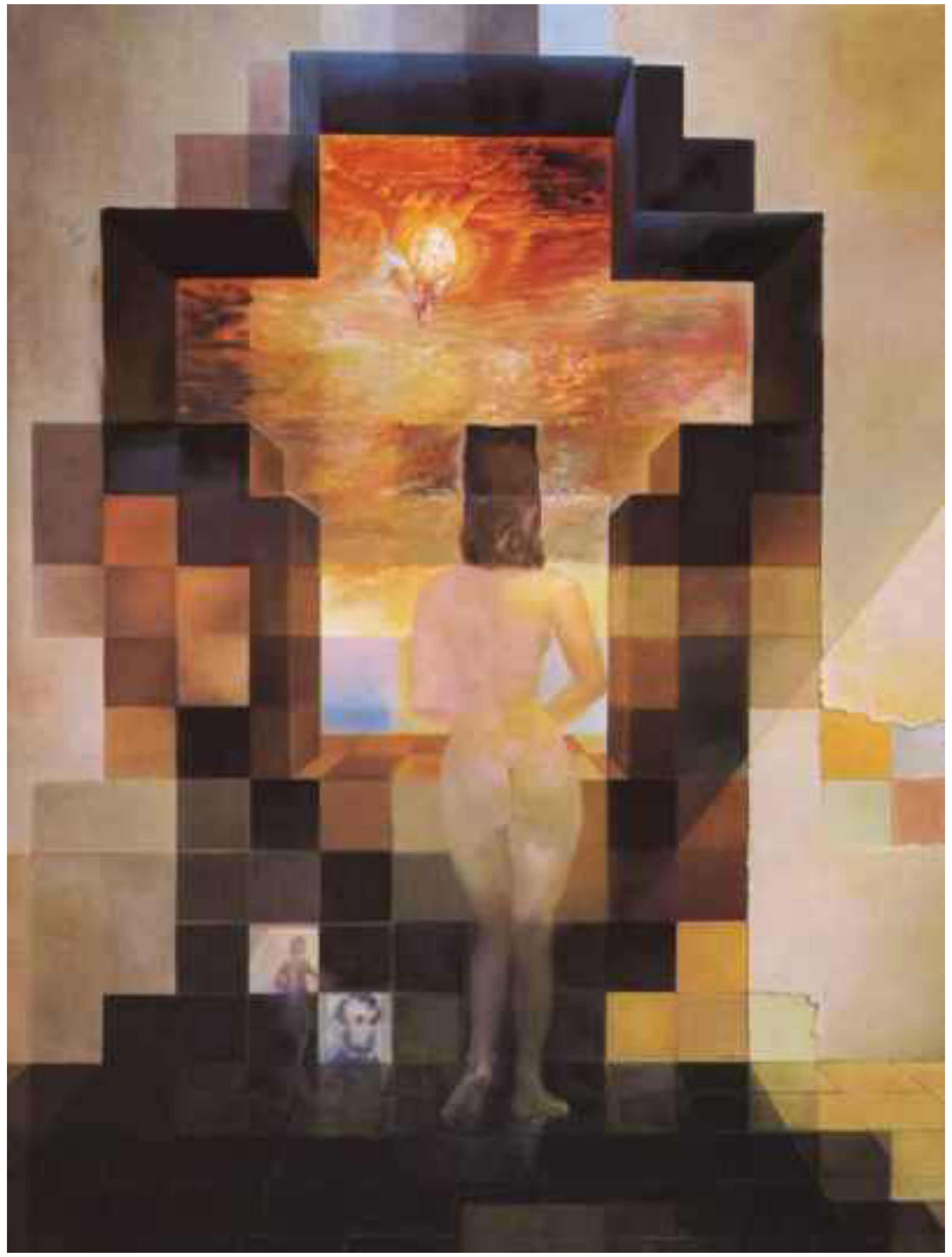
$$g[m,n;\sigma] = \exp -\frac{m^2+n^2}{2\sigma^2}$$

Scale

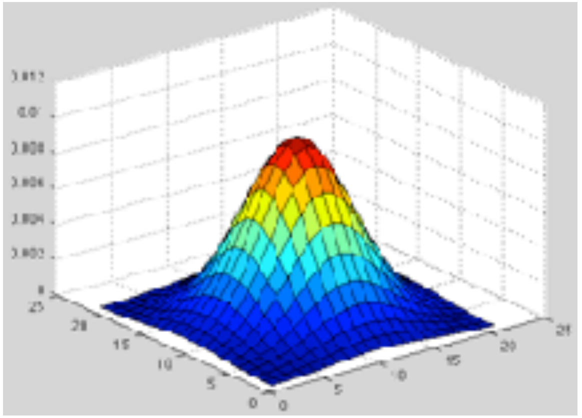
$$g[m,n;\sigma] = \exp - \frac{m^2 + n^2}{2\sigma^2}$$



Gaussian filter for low-pass filtering



Dali



Properties of the Gaussian filter

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

- The n-dimensional Gaussian is the only completely circularly symmetric operator that is separable.

- The (continuous) Fourier transform of a Gaussian is another Gaussian

$$G(u, v; \sigma) = \exp -2\pi^2(u^2 + v^2)\sigma^2$$

Properties of the Gaussian filter

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

- The convolution of two n-dimensional Gaussians is an n-dimensional Gaussian.

$$g(x, y; \sigma_1) \circ g(x, y; \sigma_2) = g(x, y; \sigma_3)$$

where the variance of the result is the sum

$$\sigma_3^2 = \sigma_1^2 + \sigma_2^2$$

(it is easy to prove this using the FT of the Gaussian)

Binomial filter

Binomial coefficients provide a compact approximation of the gaussian coefficients using only integers.

The simplest blur filter (low pass) is

$$[1 \quad 1]$$

Binomial filters in the family of filters obtained as successive convolutions of [1 1]

Binomial filter

$$b_1 = [1 \;\; 1]$$

$$b_2 = [1 \; 1] \circ [1 \; 1] = [1 \; 2 \; 1]$$

$$b_3 = [1 \; 1] \circ [1 \; 1] \circ [1 \; 1] = [1 \; 3 \; 3 \; 1]$$

Binomial filter

b_1	1 1	$\sigma_1^2 = 1/4$
b_2	1 2 1	$\sigma_2^2 = 1/2$
b_3	1 3 3 1	$\sigma_3^2 = 3/4$
b_4	1 4 6 4 1	$\sigma_4^2 = 1$
b_5	1 5 10 10 5 1	$\sigma_5^2 = 5/4$
b_6	1 6 15 20 15 6 1	$\sigma_6^2 = 3/2$
b_7	1 7 21 35 35 21 7 1	$\sigma_7^2 = 7/4$
b_8	1 8 28 56 70 56 28 8 1	$\sigma_8^2 = 2$

Properties of binomial filters

- Sum of the values is 2^n
- The variance of b_n is $\sigma^2 = n/4$
- The convolution of two binomial filters is also a binomial filter

$$b_n \circ b_m = b_{n+m}$$

With a variance:

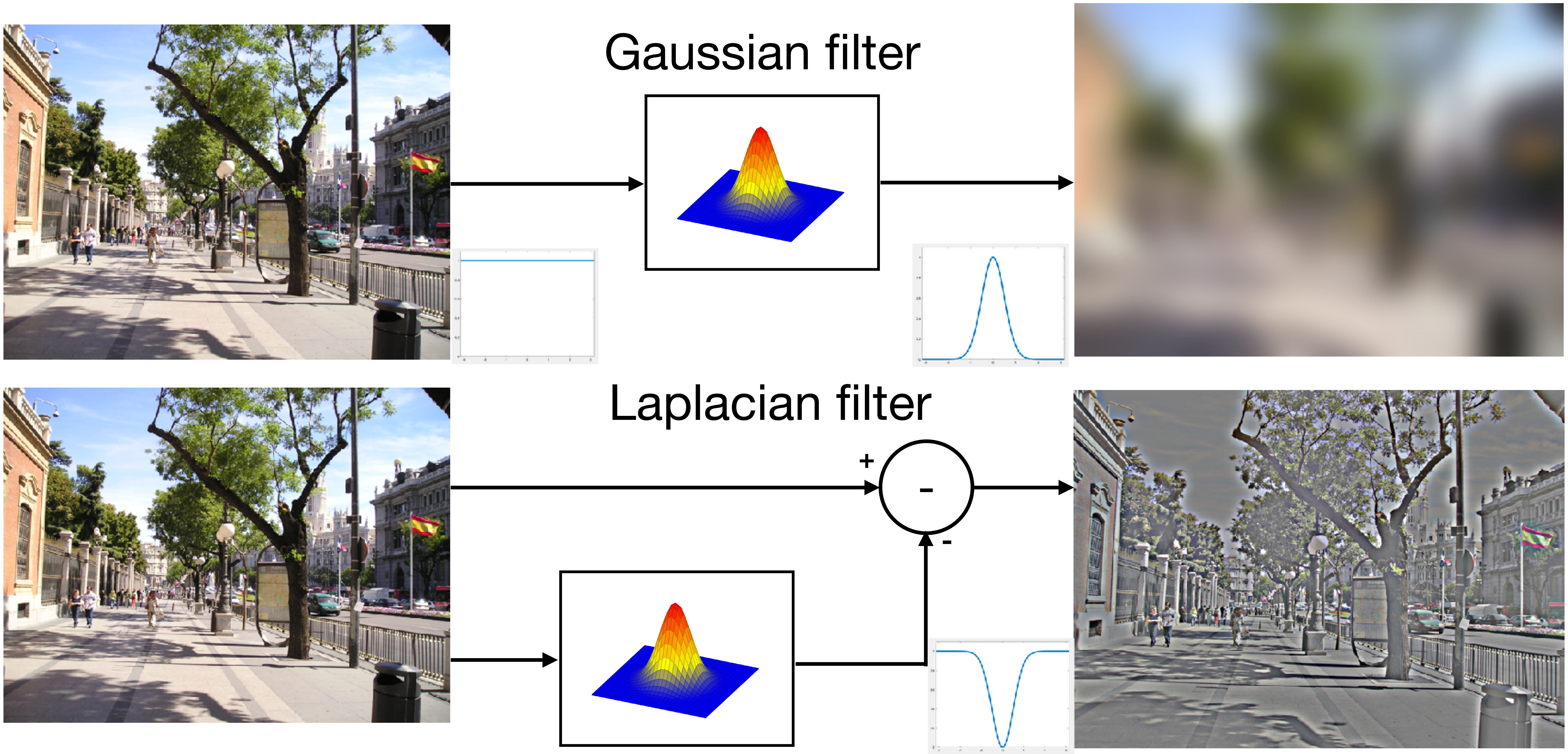
$$\sigma_n^2 + \sigma_m^2 = \sigma_{n+m}^2$$

These properties are analogous to the gaussian property in the continuous domain (but the binomial filter is different than a discretization of a gaussian)

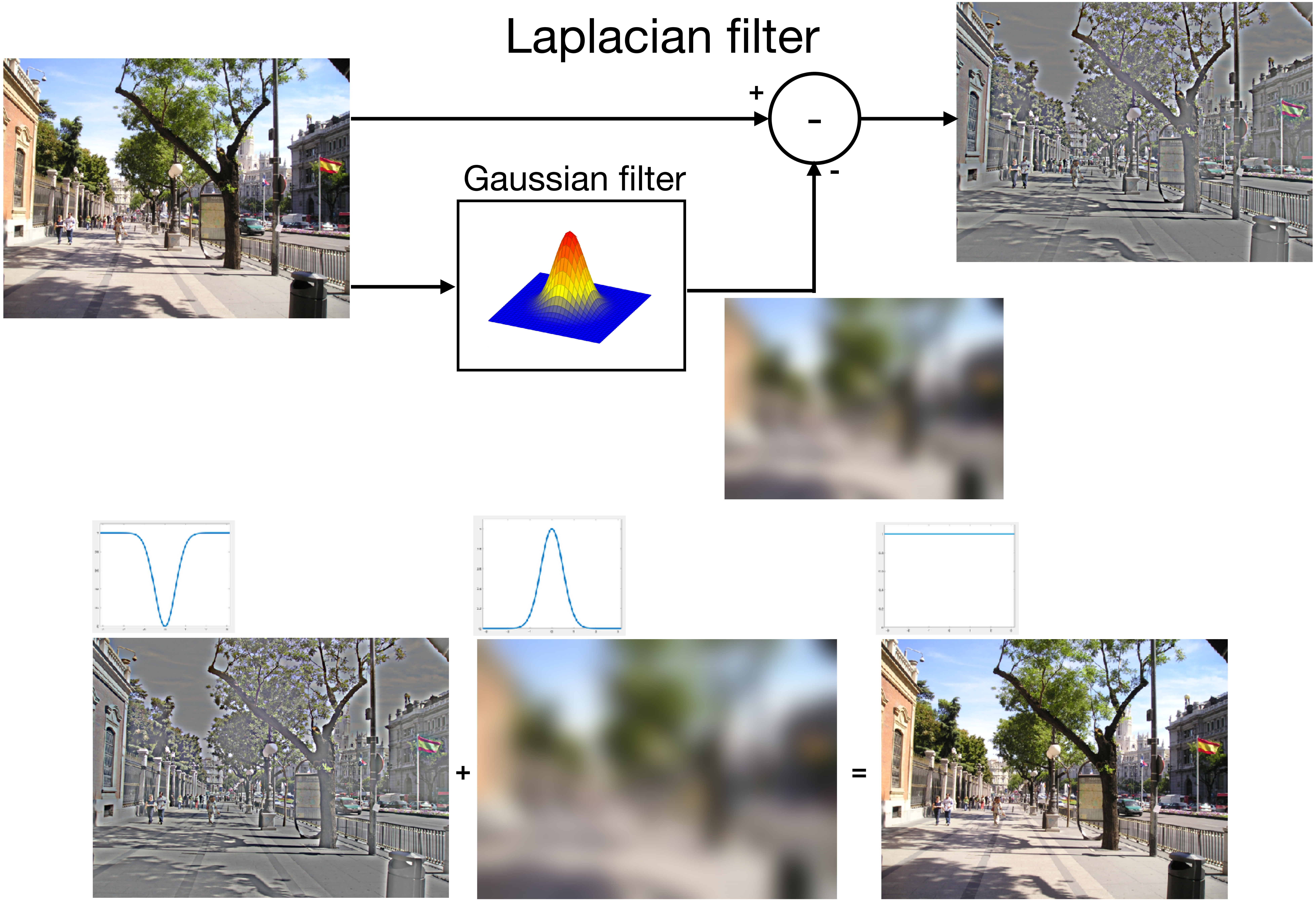
B2[n]

$$b_{2,2} = b_{2,0} \circ b_{0,2} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

What about the opposite of blurring?

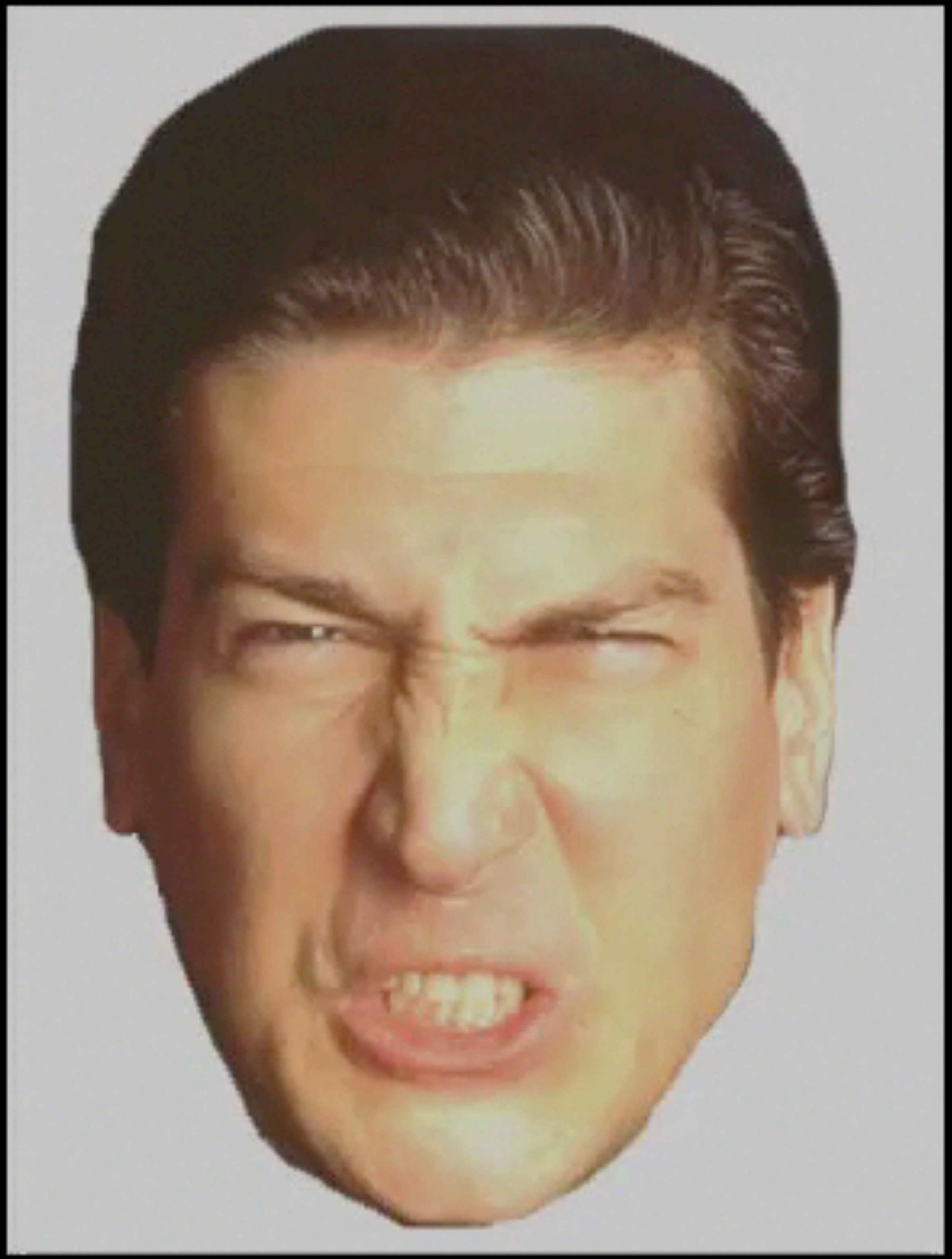


Laplacian filter

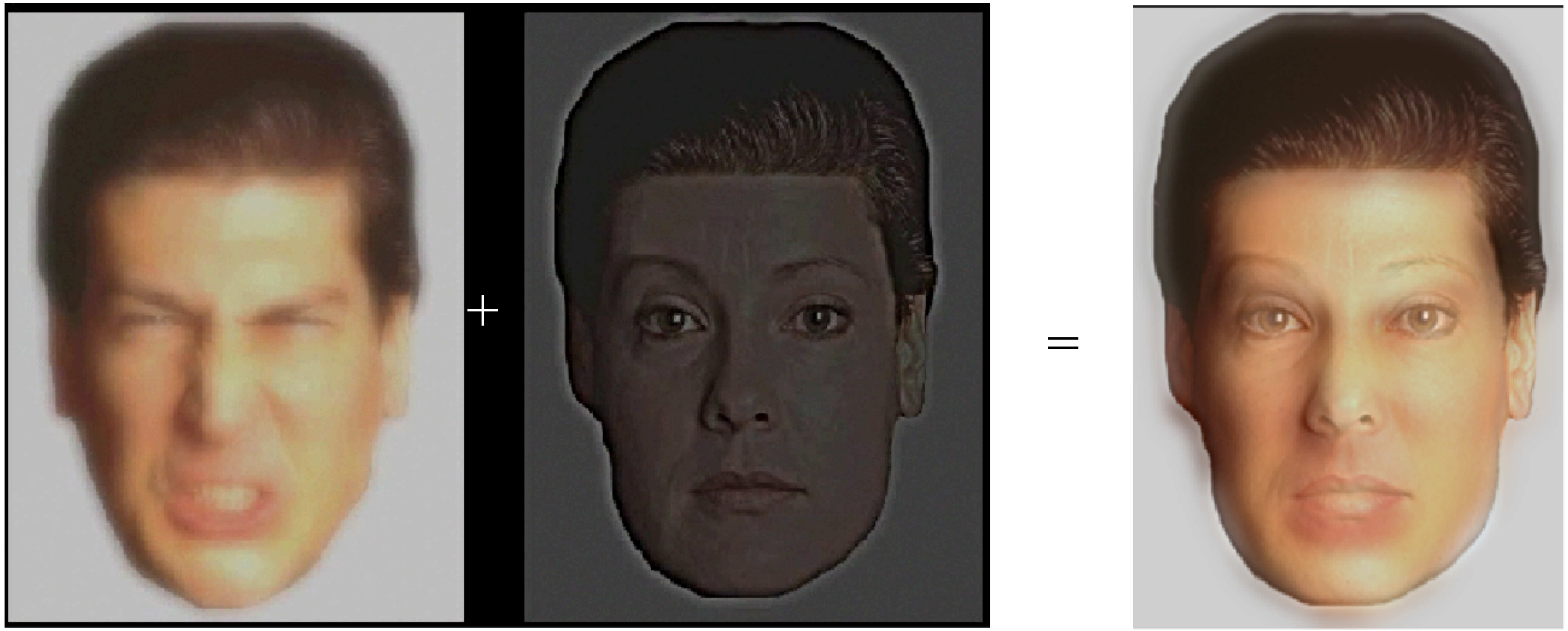


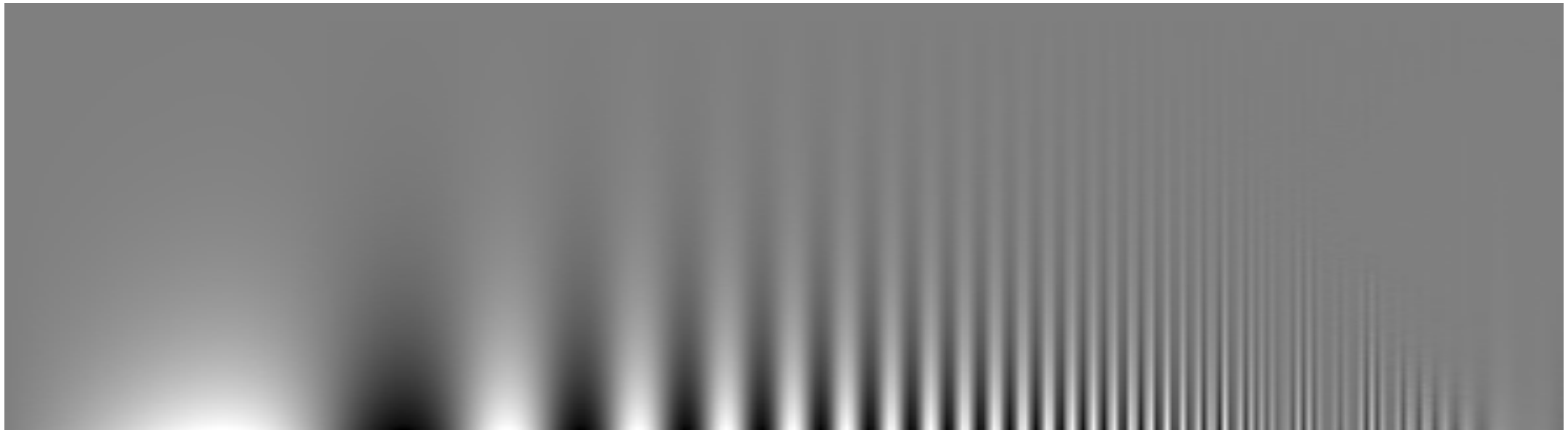
Hybrid Images

Oliva & Schyns

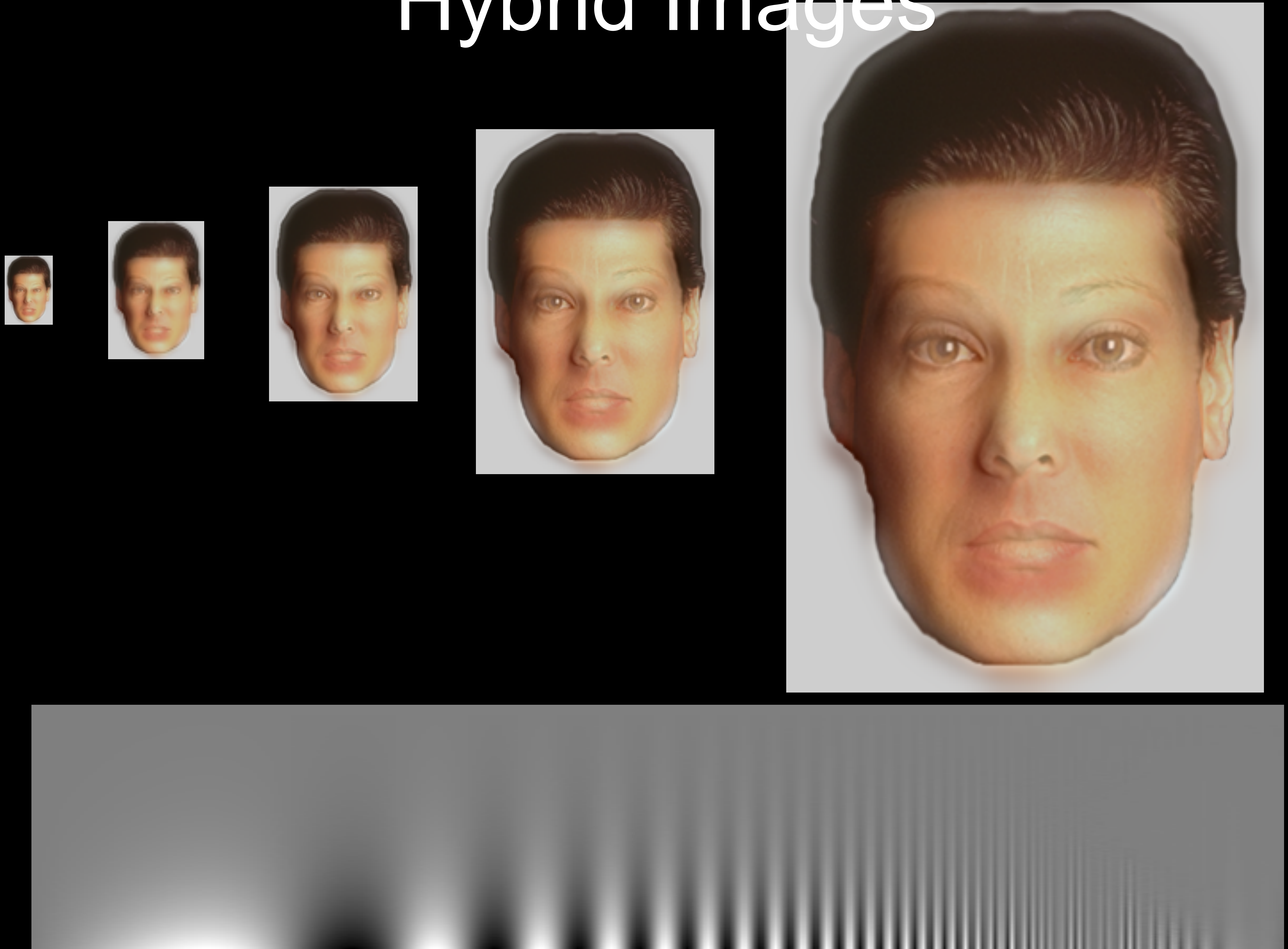


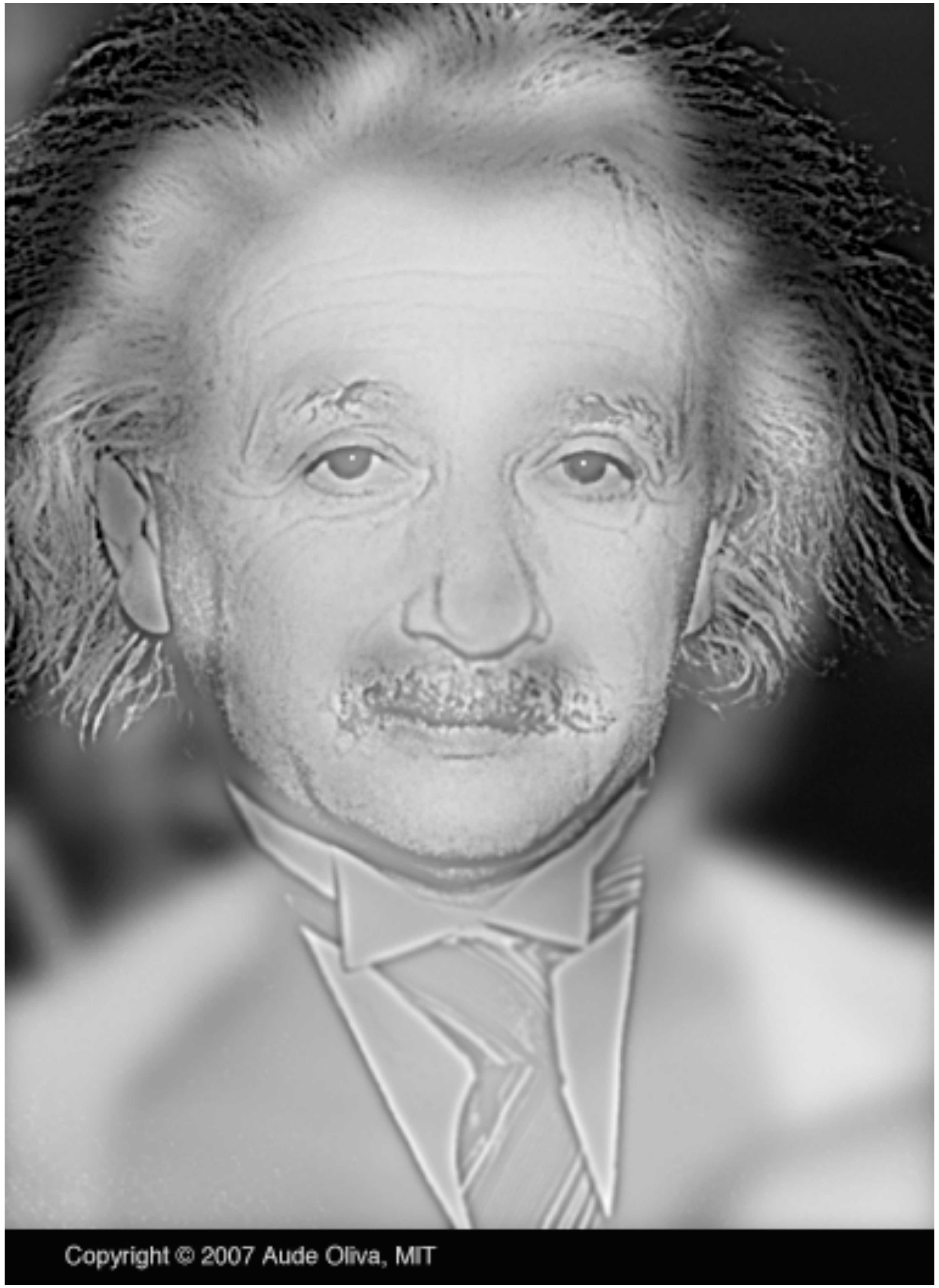
Hybrid Images

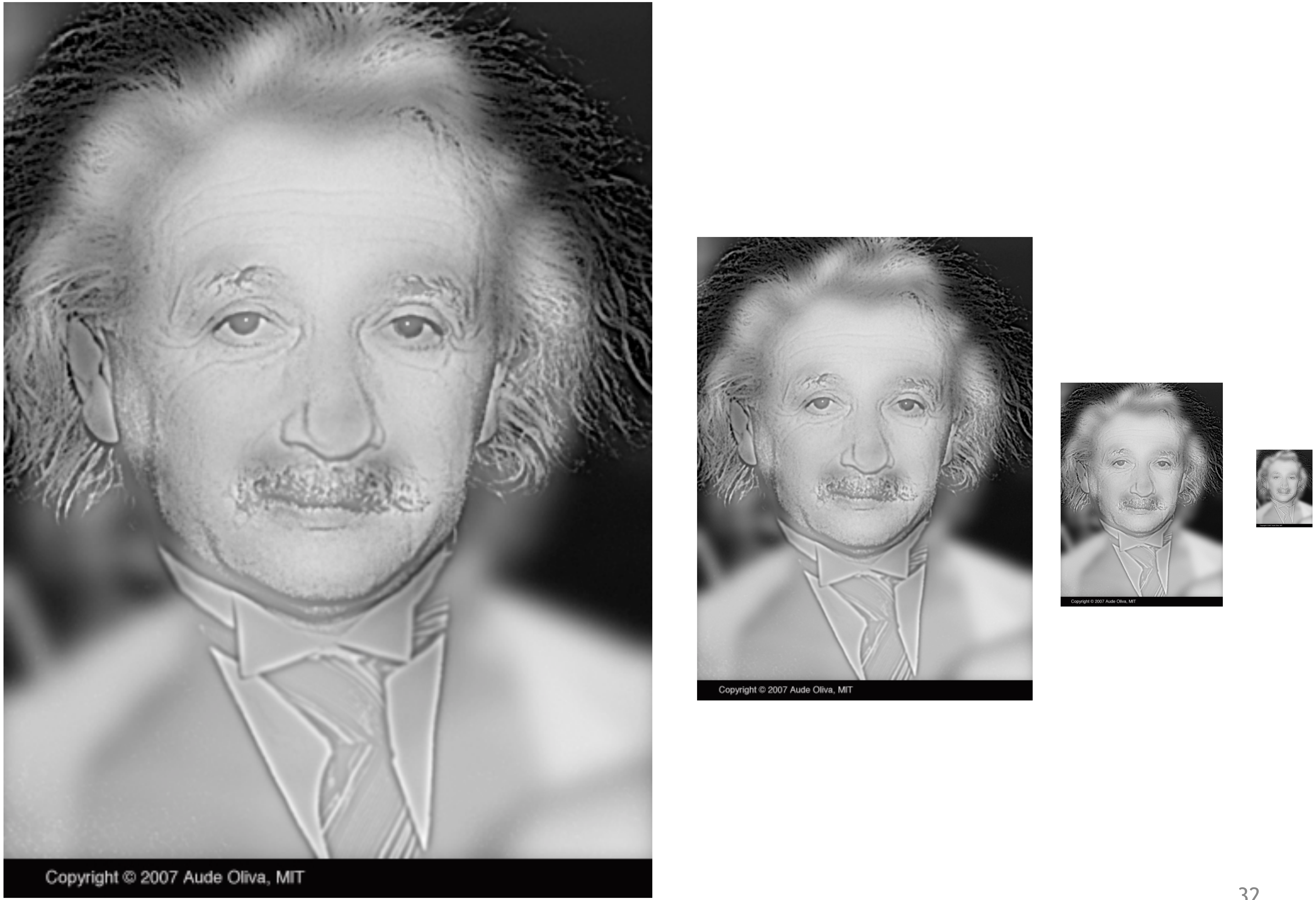




Hybrid Images




Copyright © 2007 Aude Oliva, MIT



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

High pass-filters

Finding edges in the image

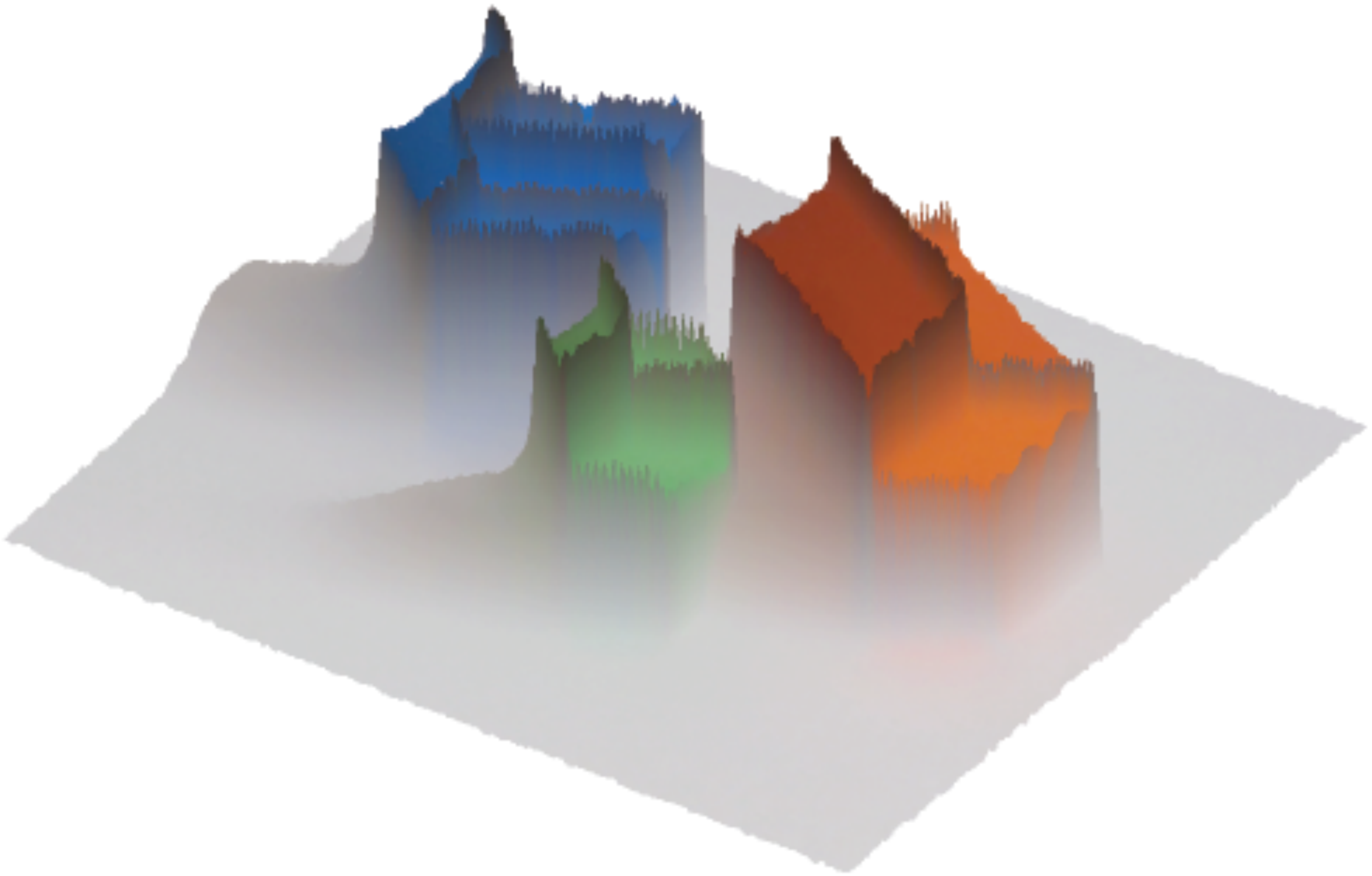


Image gradient:

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

Approximation image derivative:

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

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

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

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

[-1 1]

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

$\otimes$

[-1, 1]

h[m,n]

=

f[m,n]

g[m,n]

$[-1\ 1]^T$

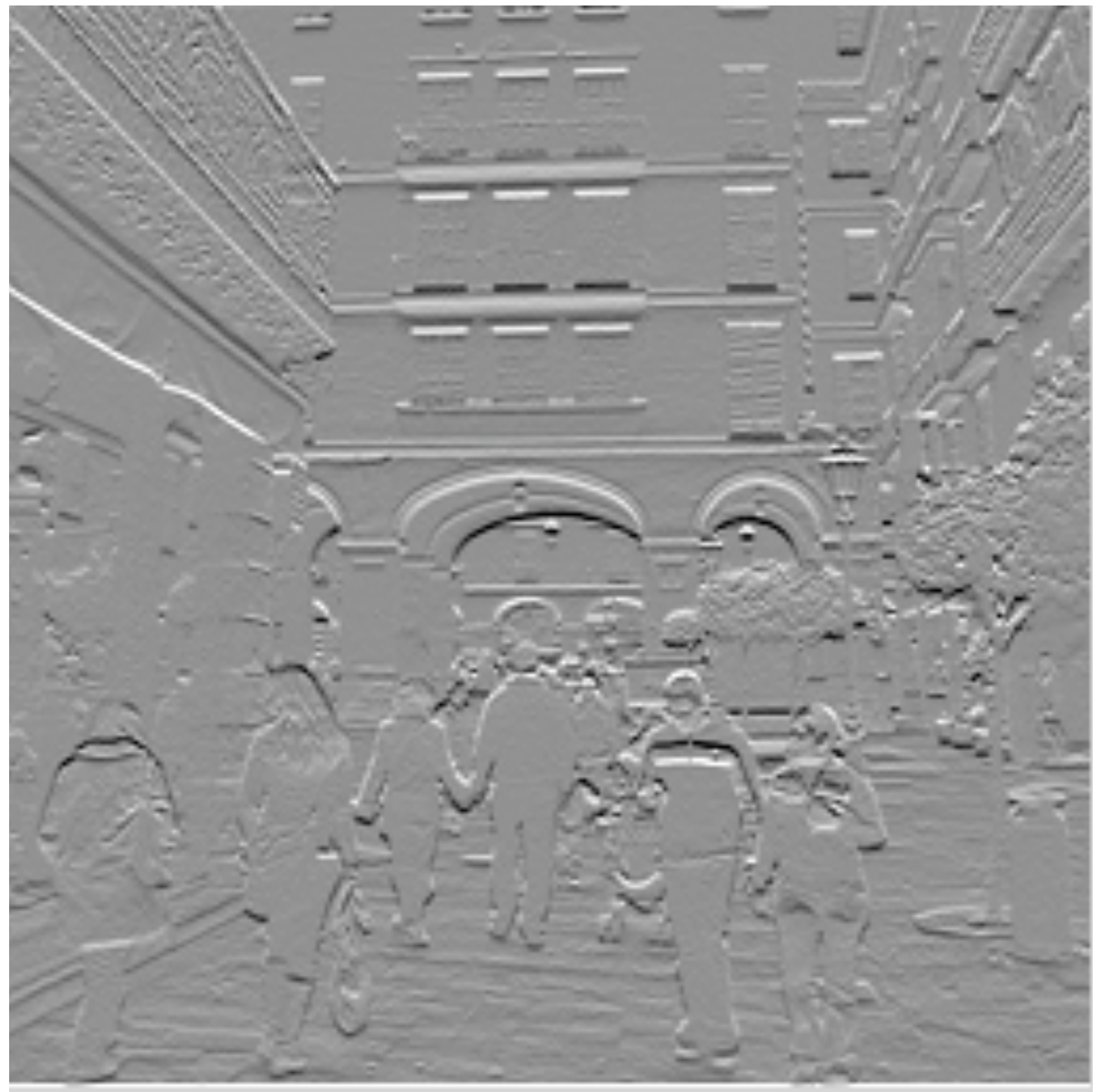
g[m,n]

$\otimes$

$[-1, 1]^T$

h[m,n]

=



f[m,n]

Discrete derivatives

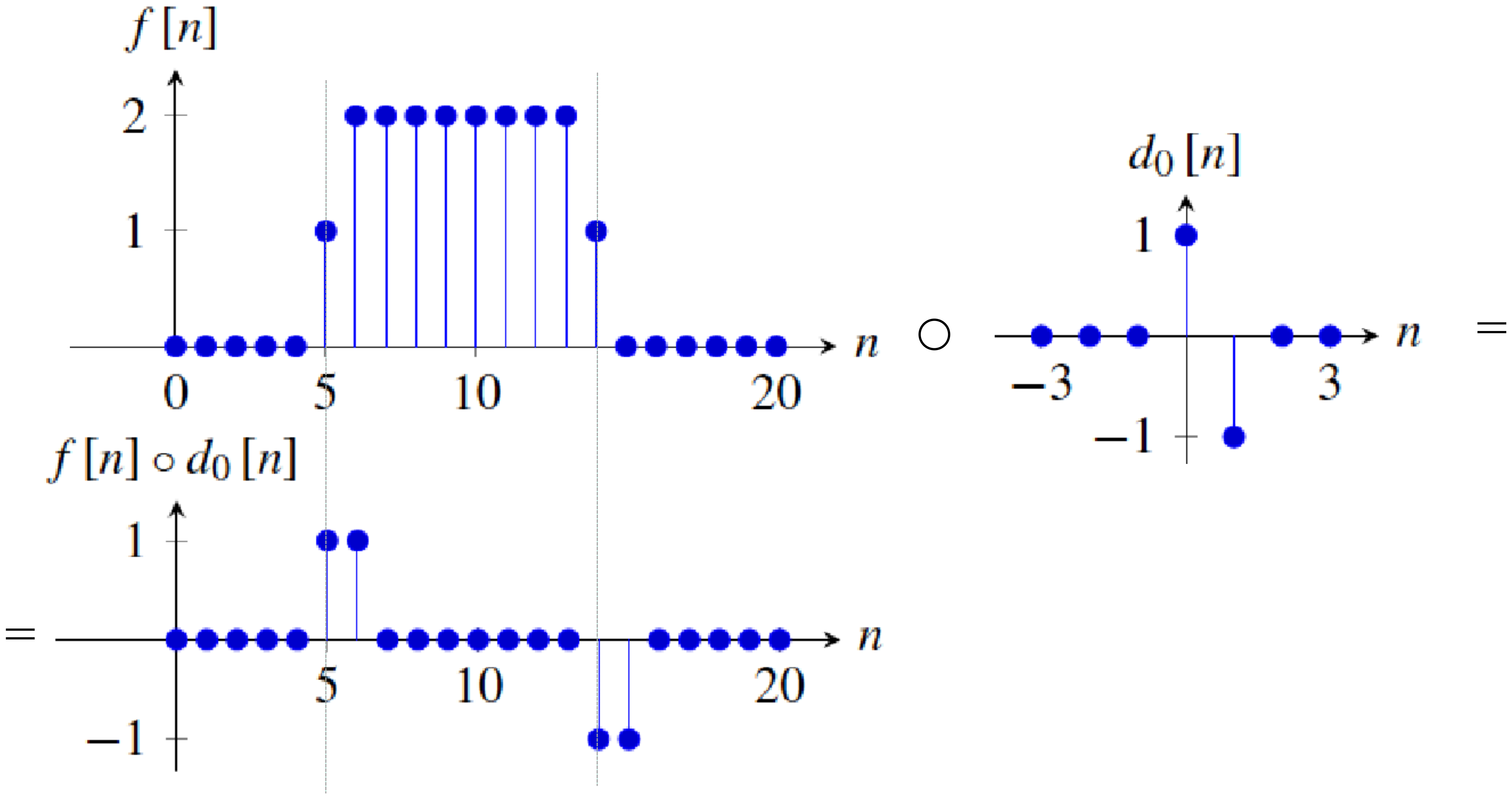
$$d_0 = [1, -1]$$

$$f \circ d_0 = f[n] - f[n-1]$$

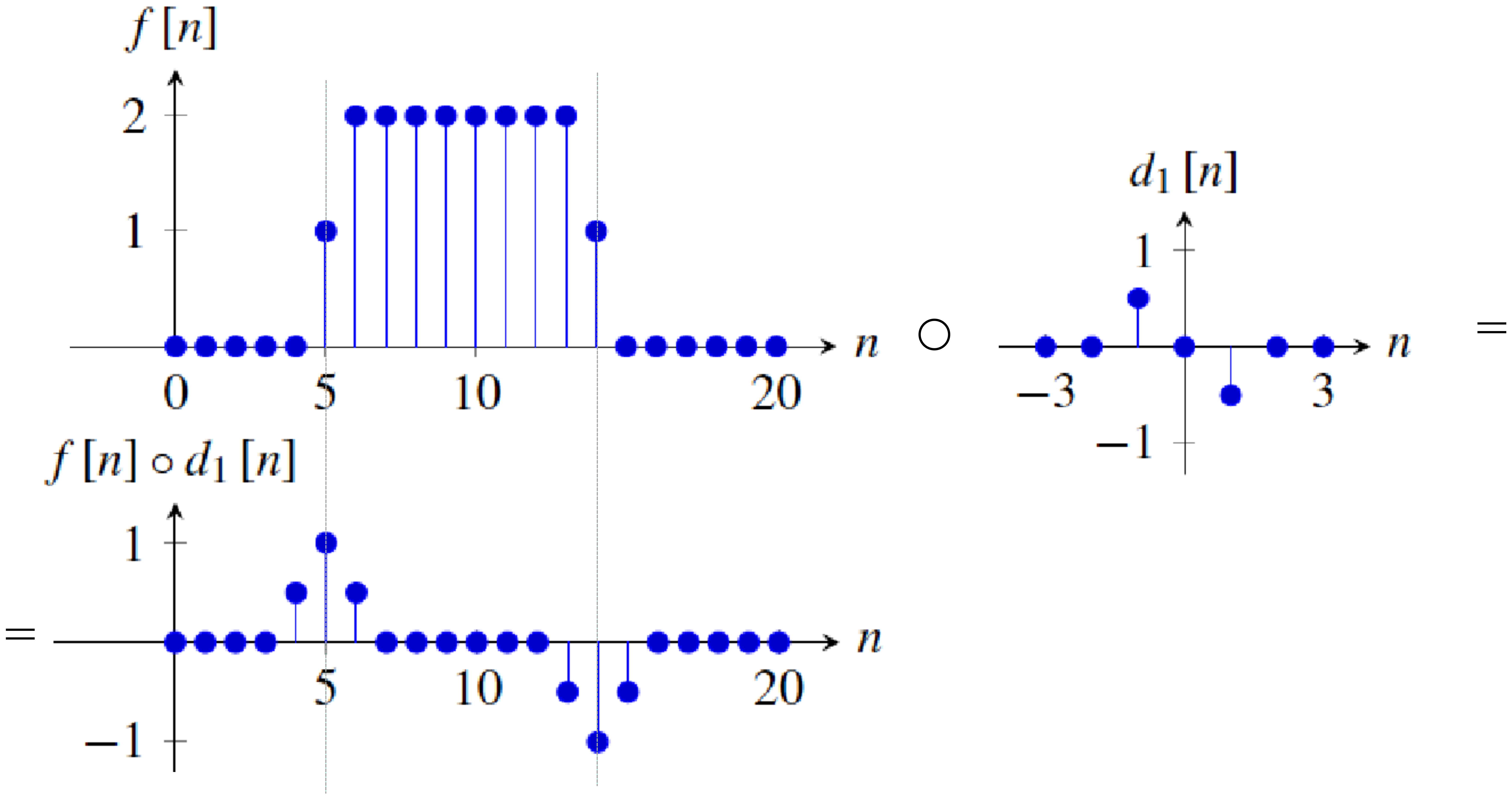
$$d_1 = [1, 0, -1]/2$$

$$f \circ d_1 = \frac{f[n+1] - f[n-1]}{2}$$

Discrete derivatives



Discrete derivatives



Derivatives

We want to compute the image derivative:

$$\frac{\partial f(x,y)}{\partial x}$$

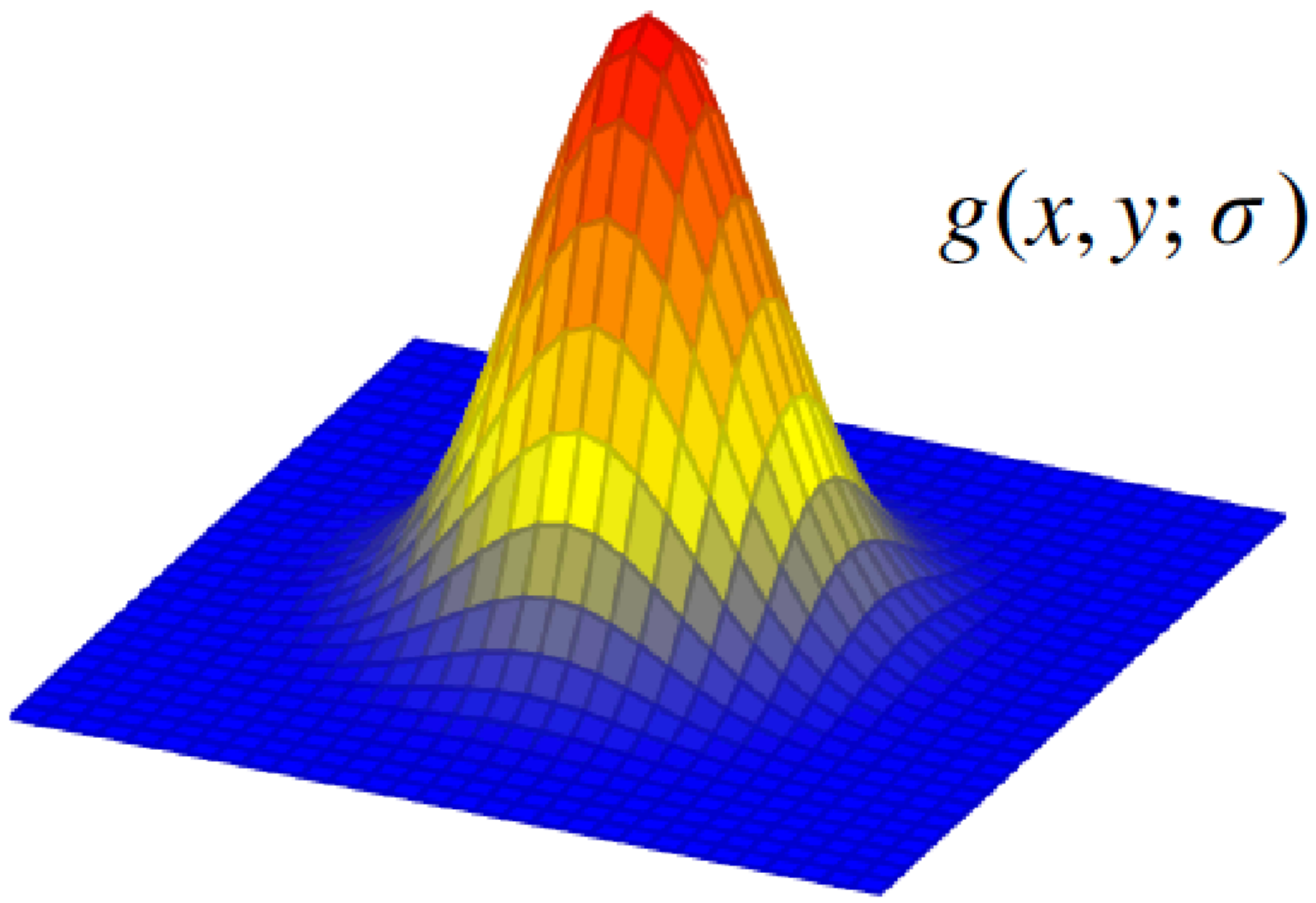
If there is noise, we might want to “smooth” it with a blurring filter

$$\frac{\partial f(x,y)}{\partial x} \circ g(x,y)$$

But derivatives and convolutions are linear and we can move them around:

$$\frac{\partial f(x,y)}{\partial x} \circ g(x,y) = f(x,y) \circ \frac{\partial g(x,y)}{\partial x}$$

Gaussian derivatives



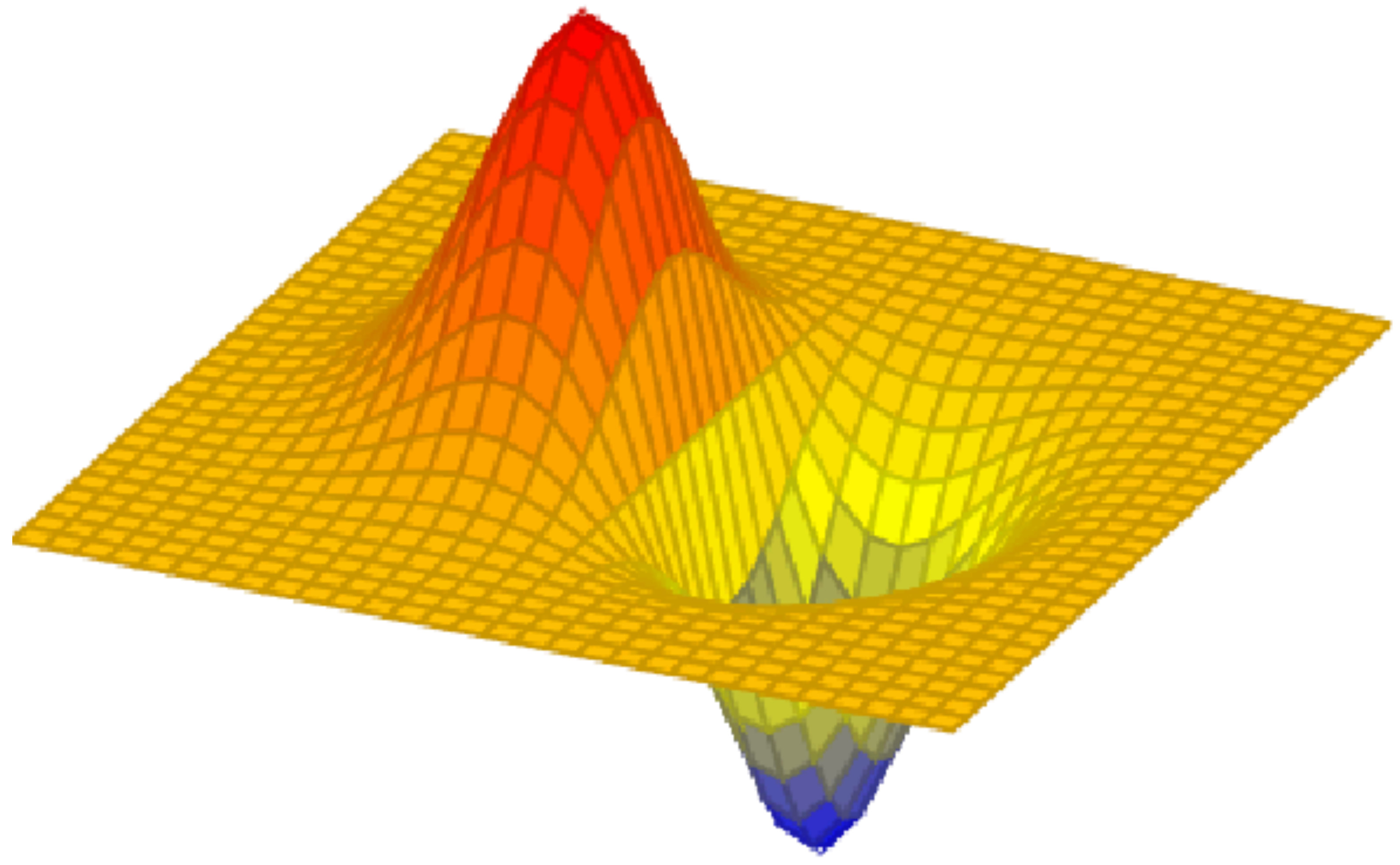
$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp - \frac{x^2 + y^2}{2\sigma^2}$$

The continuous derivative is:

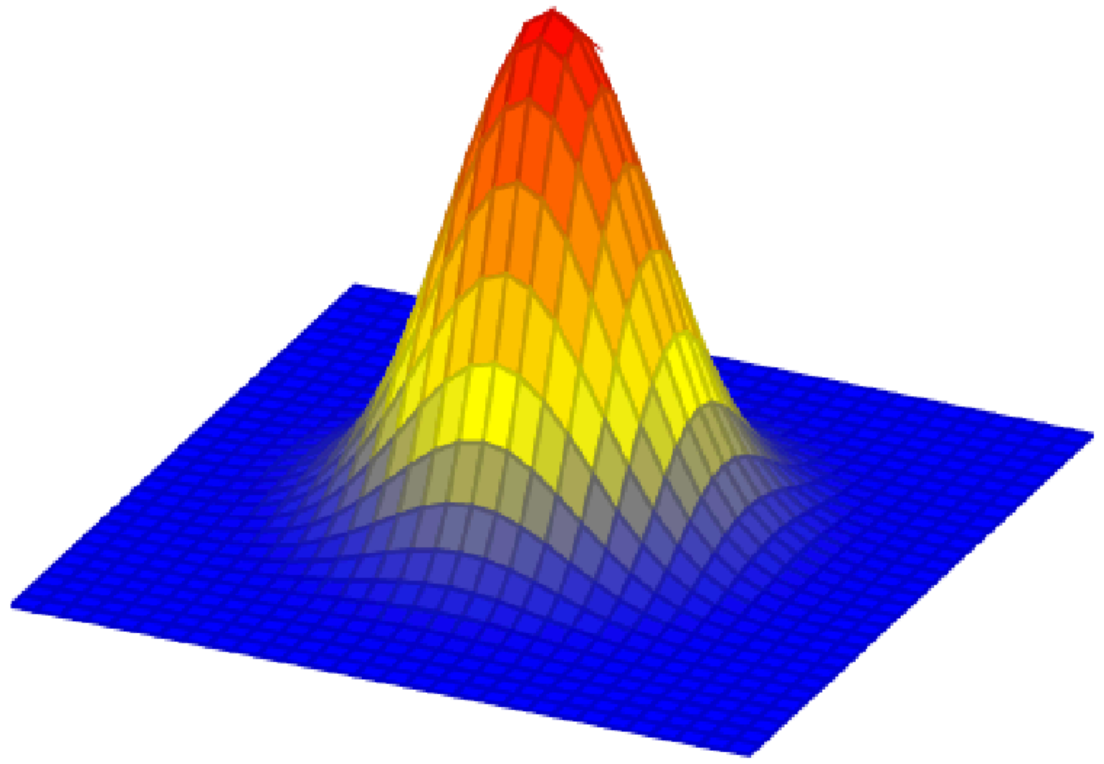
$$g_x(x, y; \sigma) = \frac{\partial g(x, y; \sigma)}{\partial x} =$$

$$= \frac{-x}{2\pi\sigma^4} \exp - \frac{x^2 + y^2}{2\sigma^2}$$

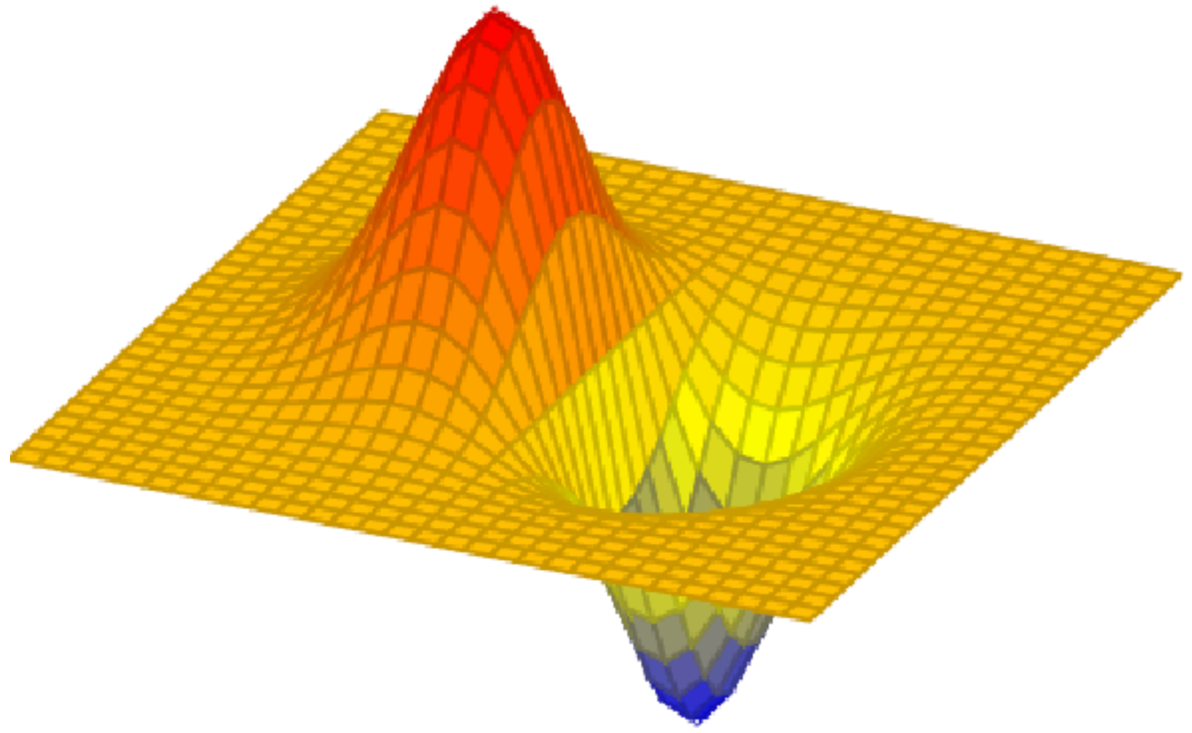
$$= \frac{-x}{\sigma^2} g(x, y; \sigma)$$



Gaussian derivatives



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

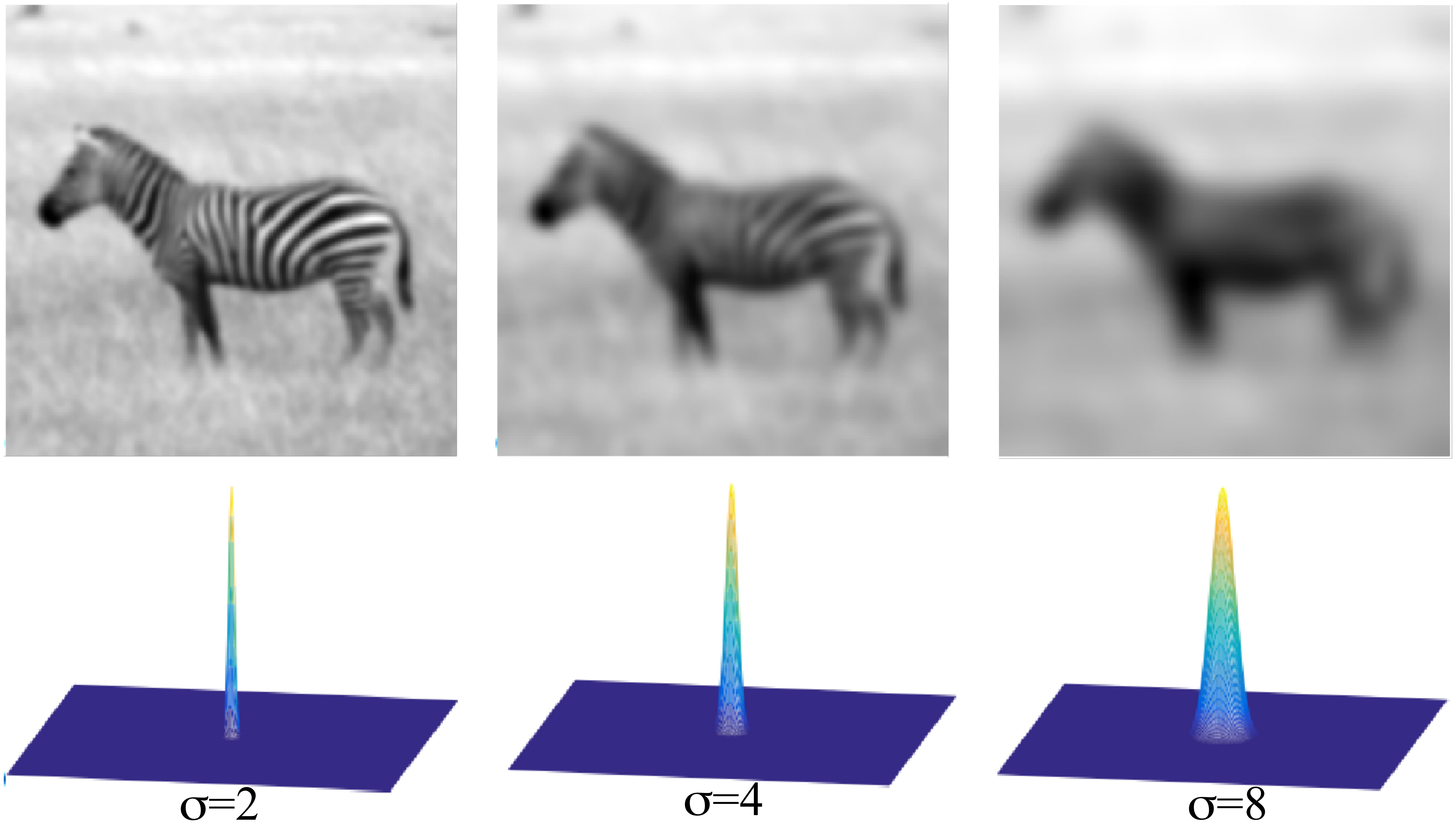


$$g_x(x,y) = \frac{-x}{2\pi\sigma^4}\exp -\frac{x^2+y^2}{2\sigma^2}$$

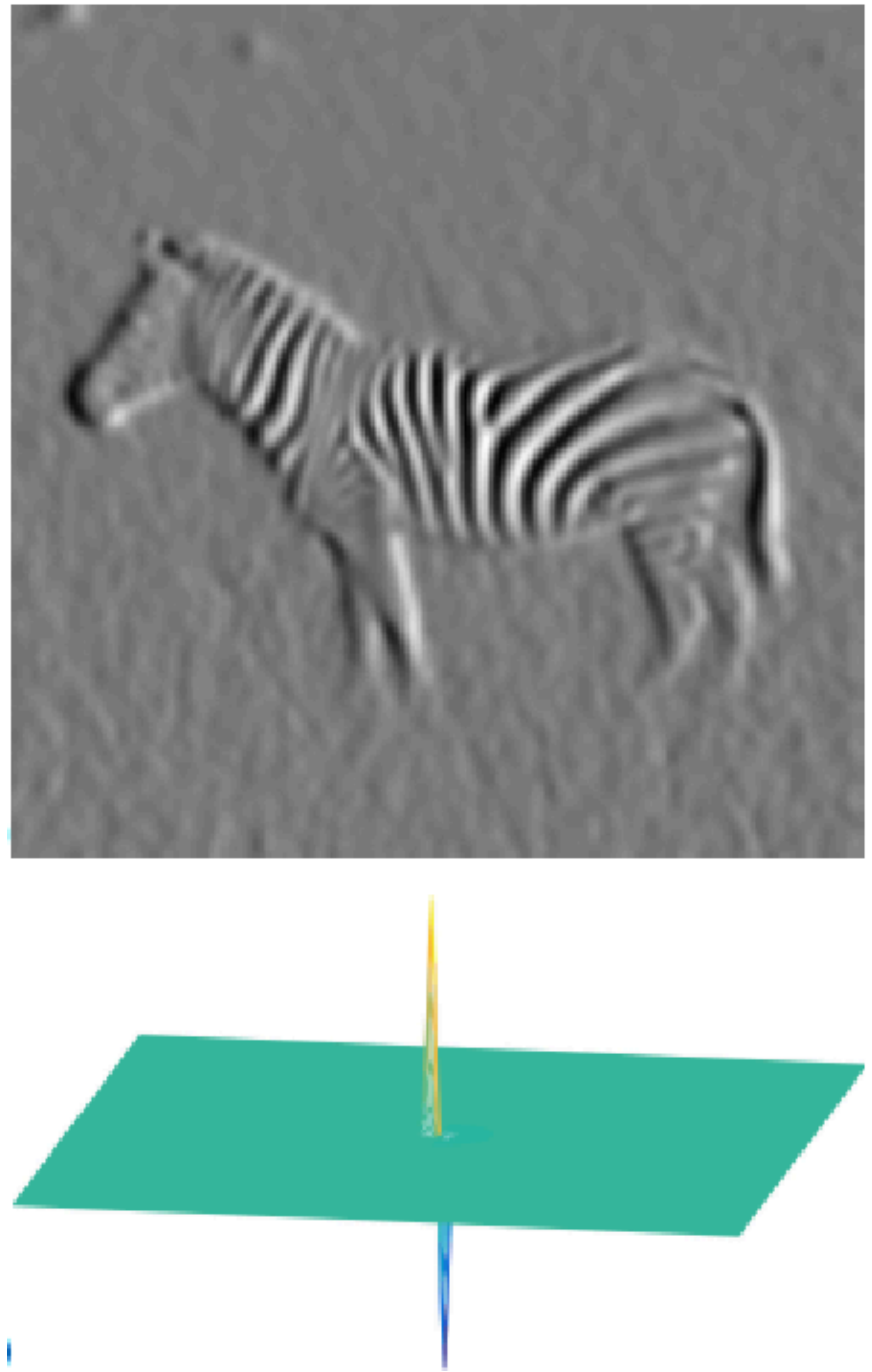
In general:

$$g_{x^n,y^m}(x,y;\sigma) = \frac{\partial^{n+m} g(x,y)}{\partial x^n \partial y^m} = \left(\frac{-1}{\sigma\sqrt{2}}\right)^{n+m} H_n\left(\frac{x}{\sigma\sqrt{2}}\right) H_m\left(\frac{y}{\sigma\sqrt{2}}\right) g(x,y;\sigma)$$

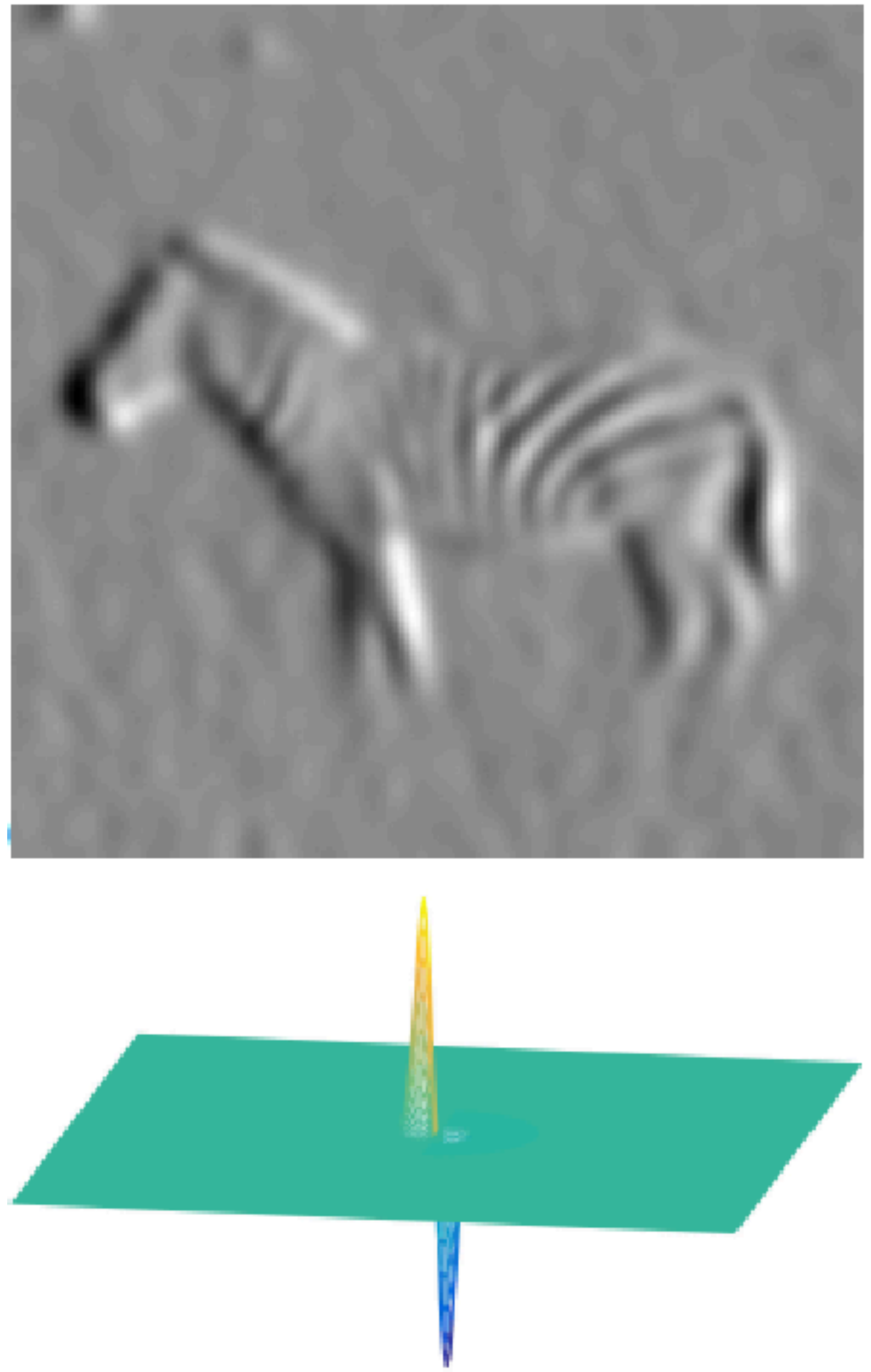
Gaussian Scale



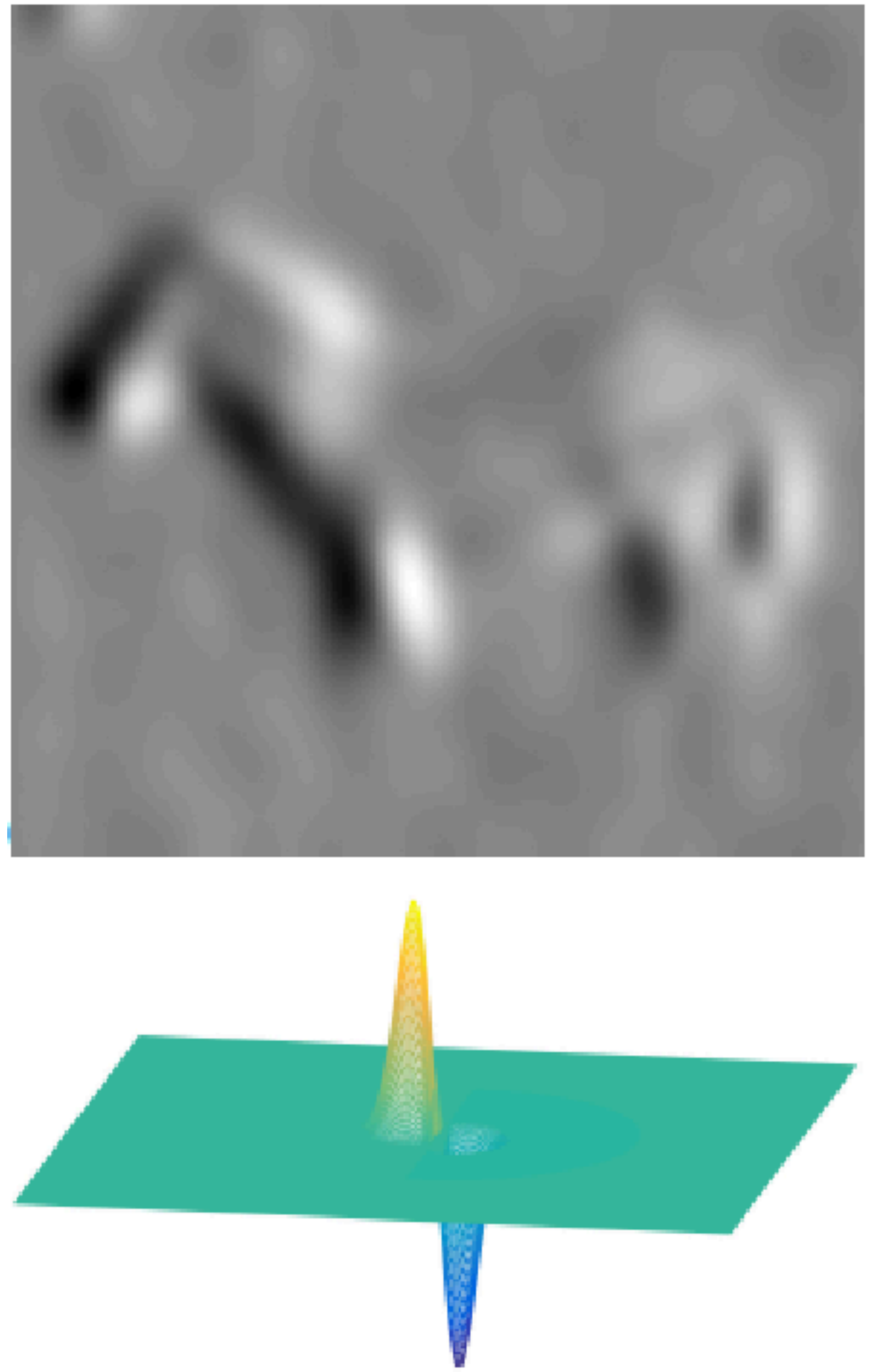
Derivatives of Gaussians: Scale



σ=2



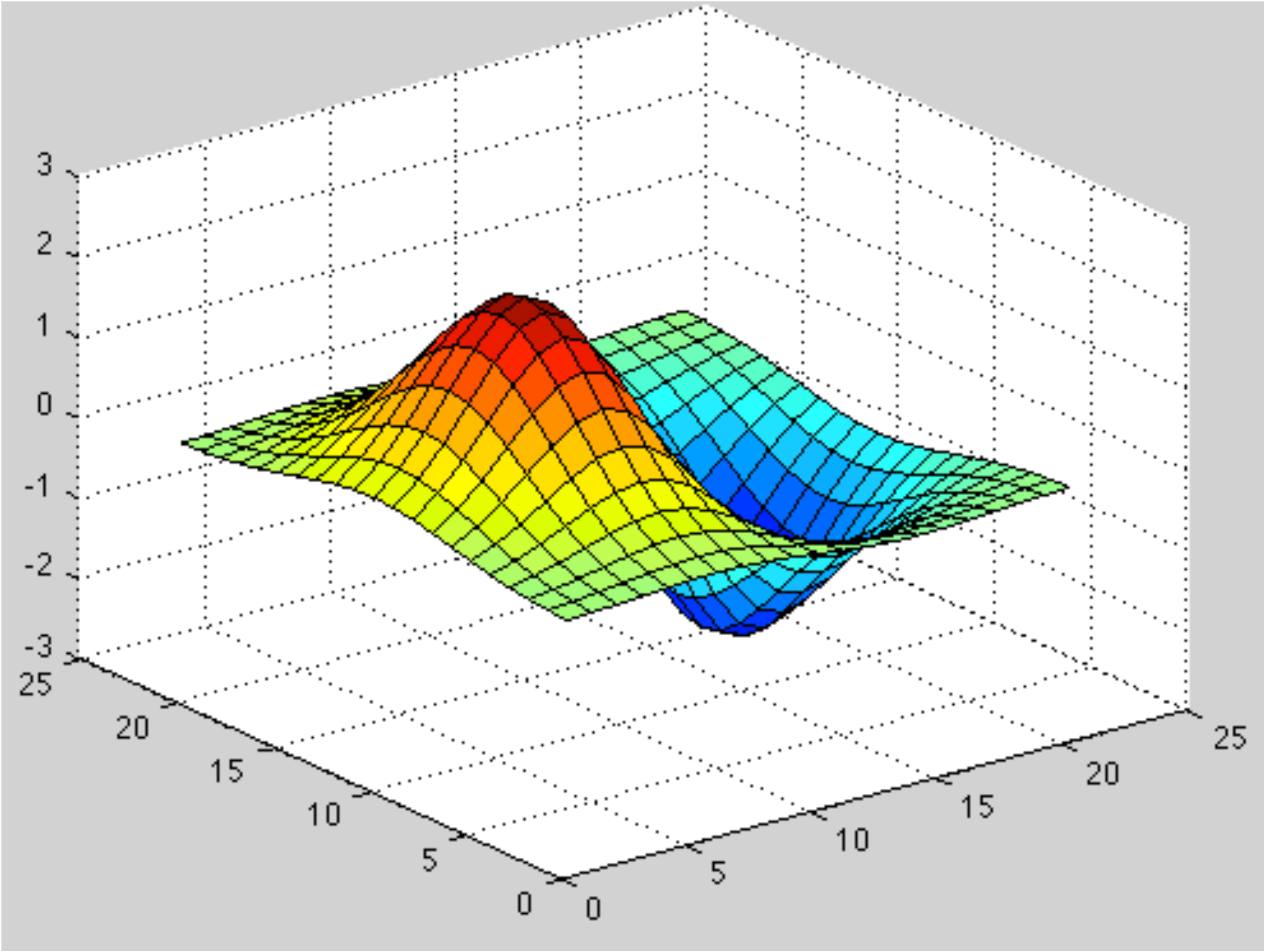
σ=4



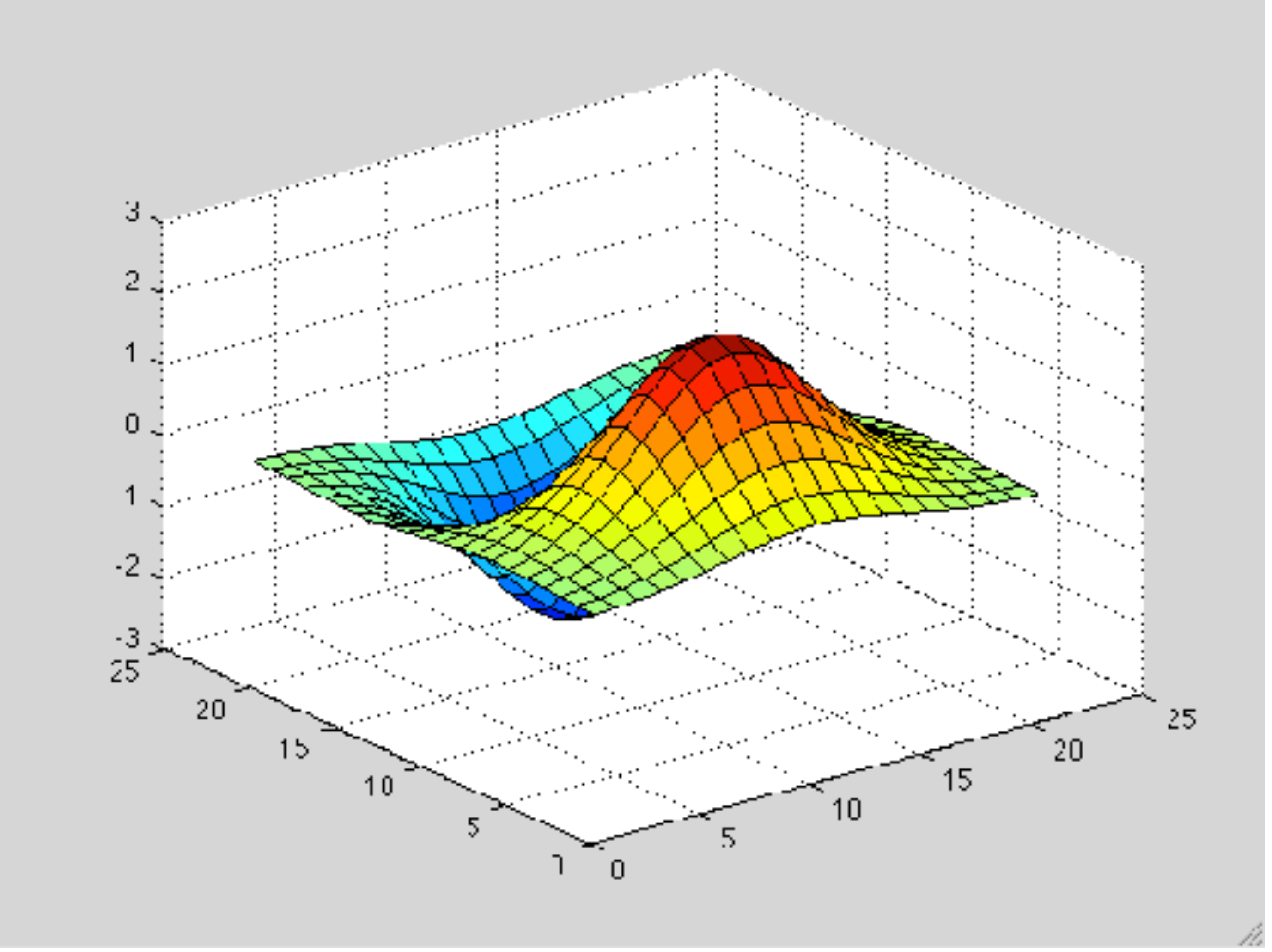
σ=8

Orientation

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

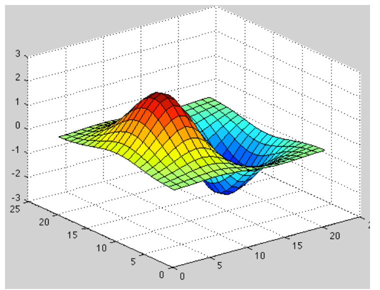


$$g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

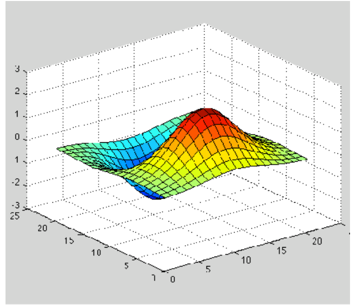


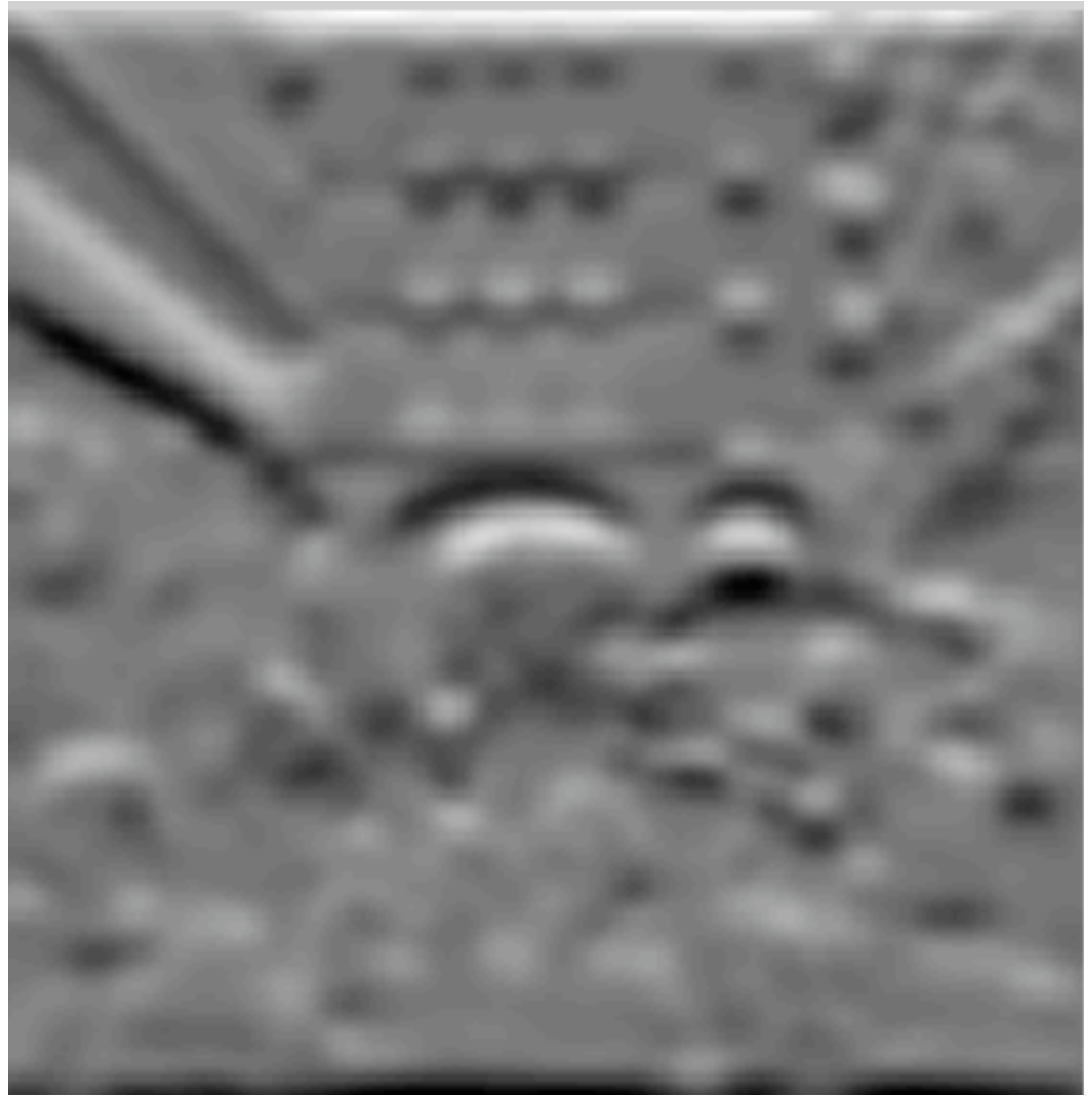
Orientation

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



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

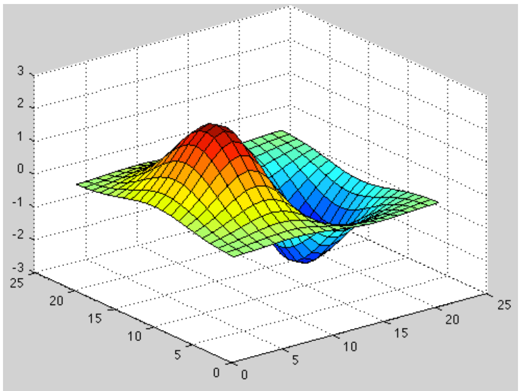




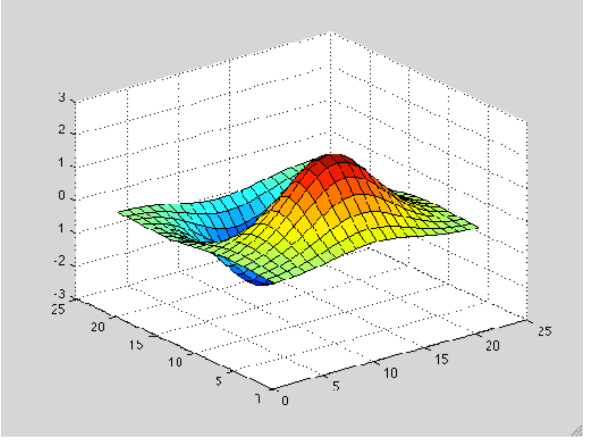
What about other orientations not axis aligned?

Orientation

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



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



The smoothed directional gradient is a linear combination of two kernels

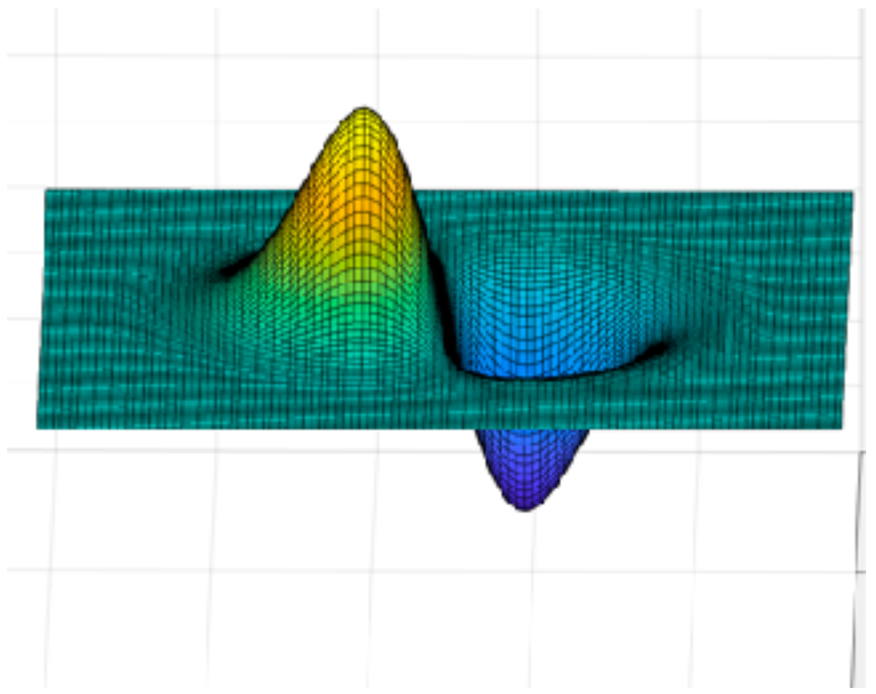
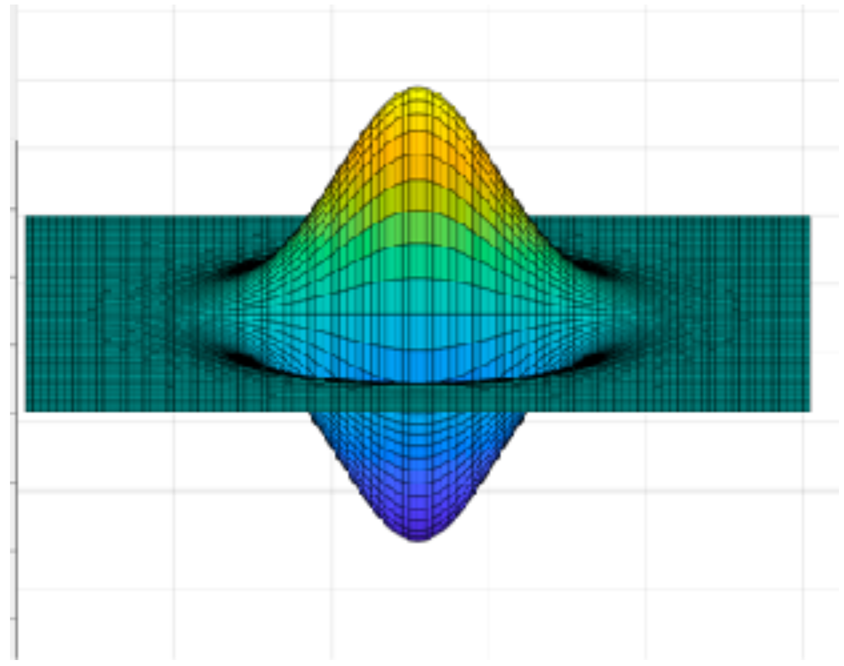
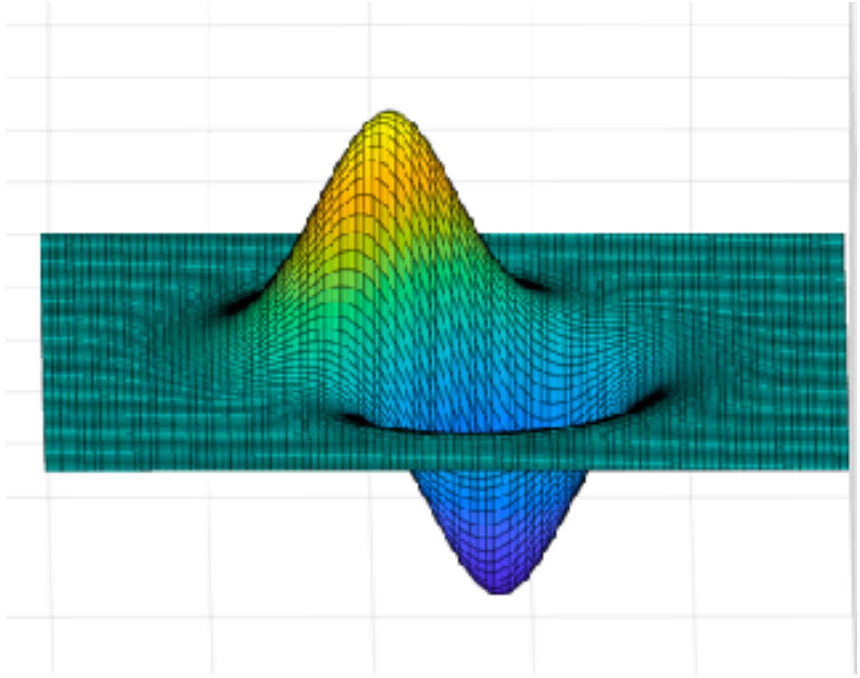
$$u^T \nabla g \otimes I = \left(\cos(\alpha) g_x(x,y) + \sin(\alpha) g_y(x,y)\right) \otimes I(x,y) =$$

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

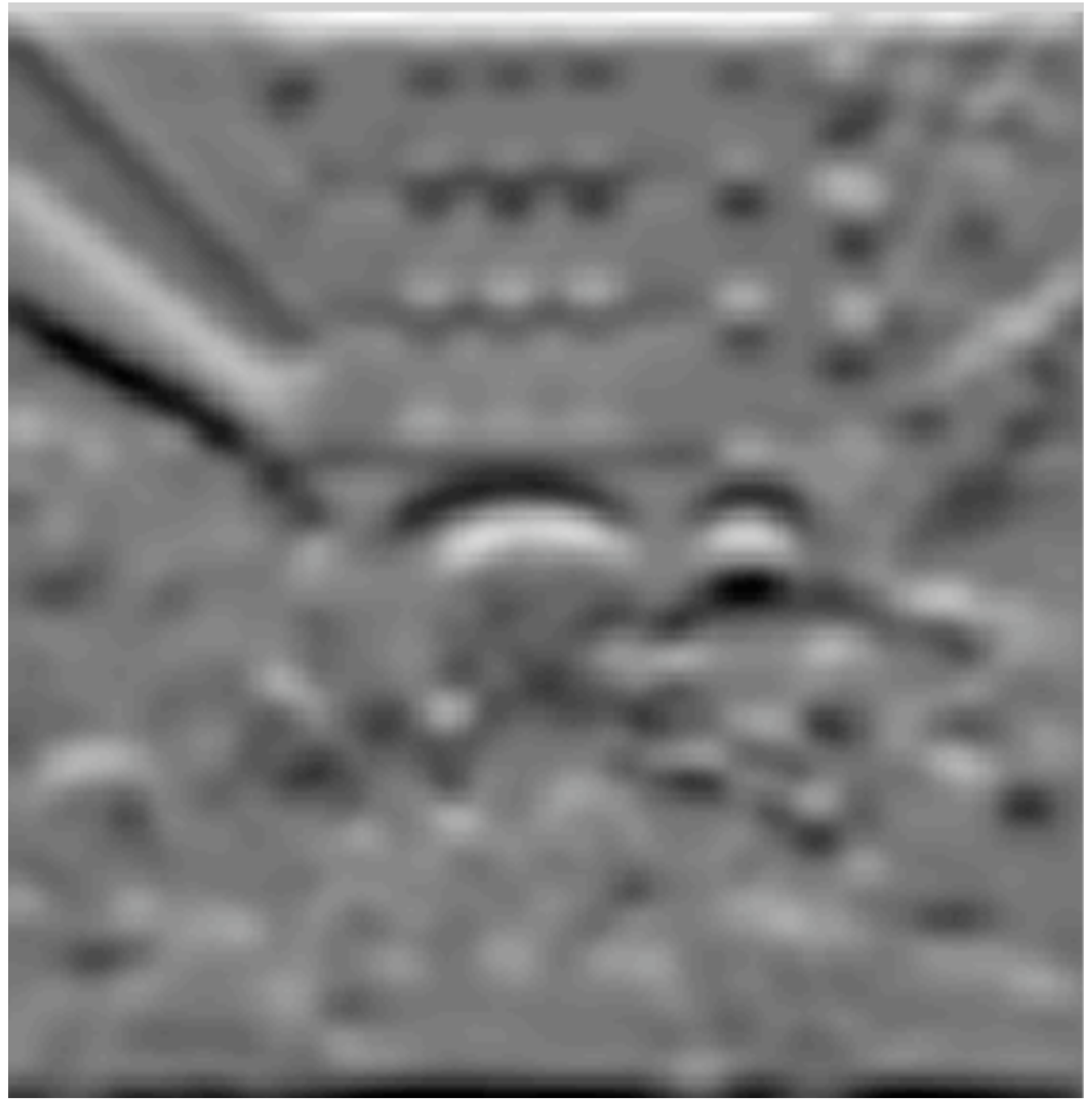
$$= \cos(\alpha) g_x(x,y) \otimes I(x,y) + \sin(\alpha) g_y(x,y) \otimes I(x,y)$$

Steereability of gaussian derivatives, Freeman & Adelson 92

Example: “steering” to 45°

$\cos(\alpha)$

+
$\sin(\alpha)$

=


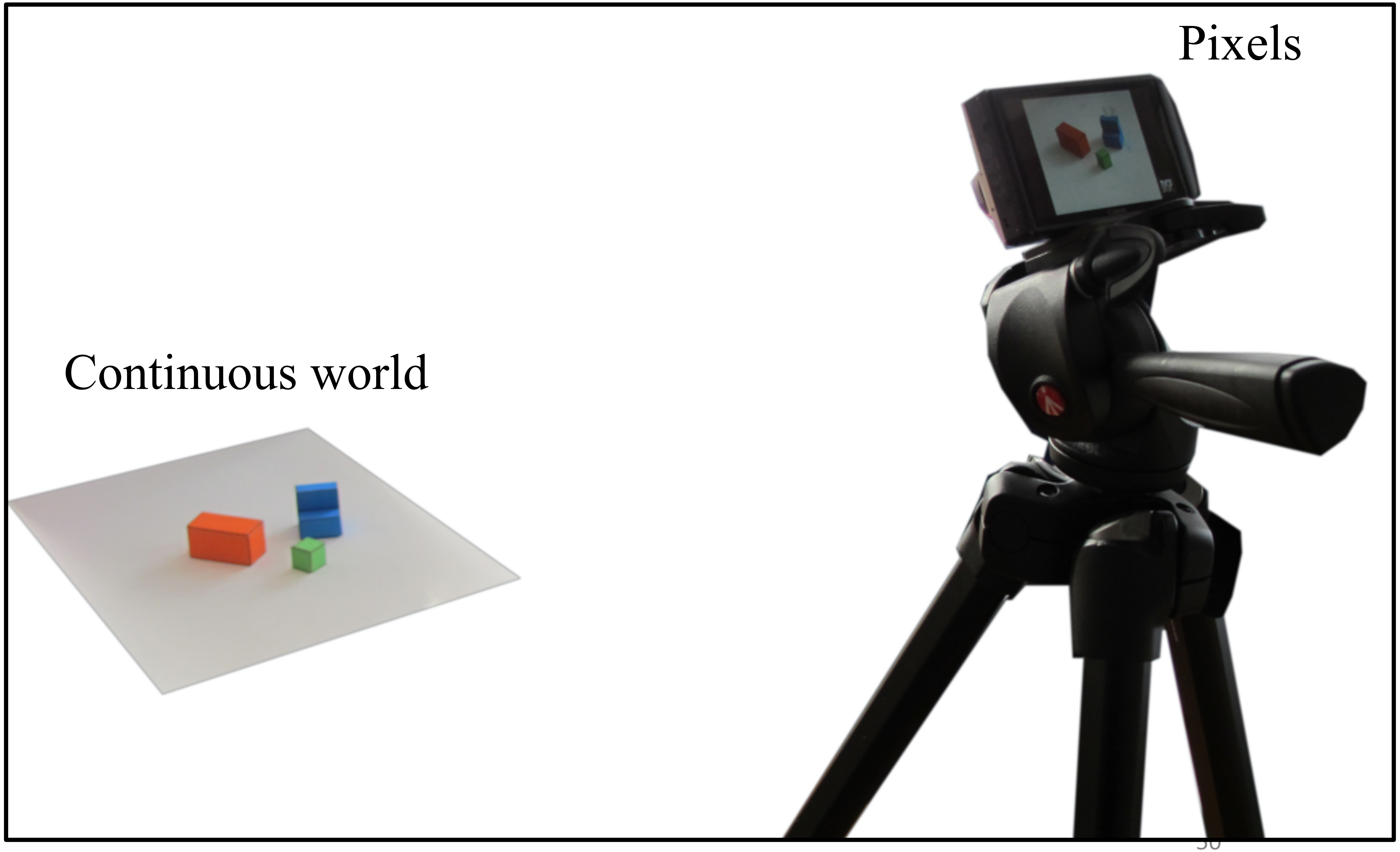
$\cos(\alpha)$

$+\sin(\alpha)$

=

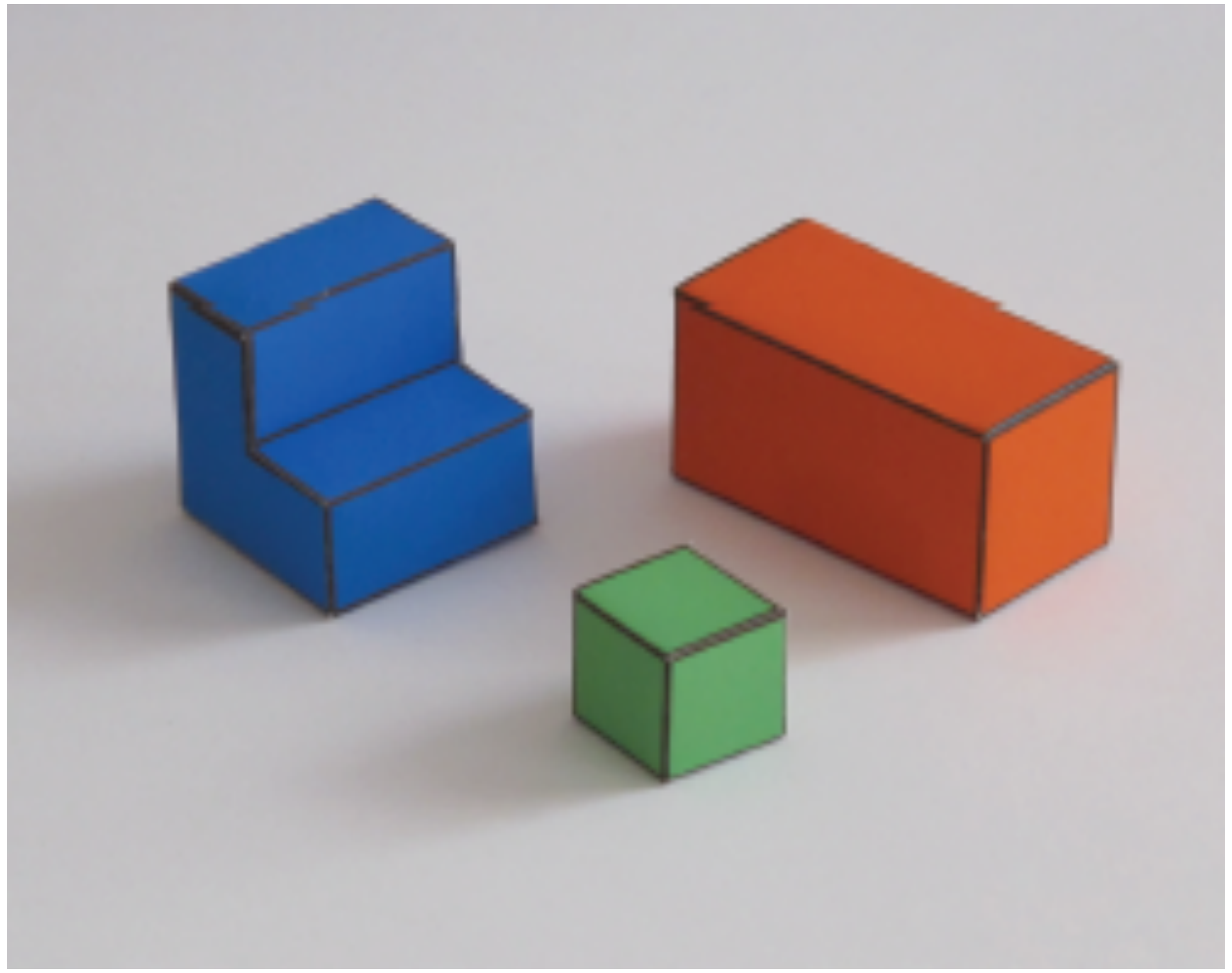

Steereability of gaussian derivatives, Freeman & Adelson 92

Sampling

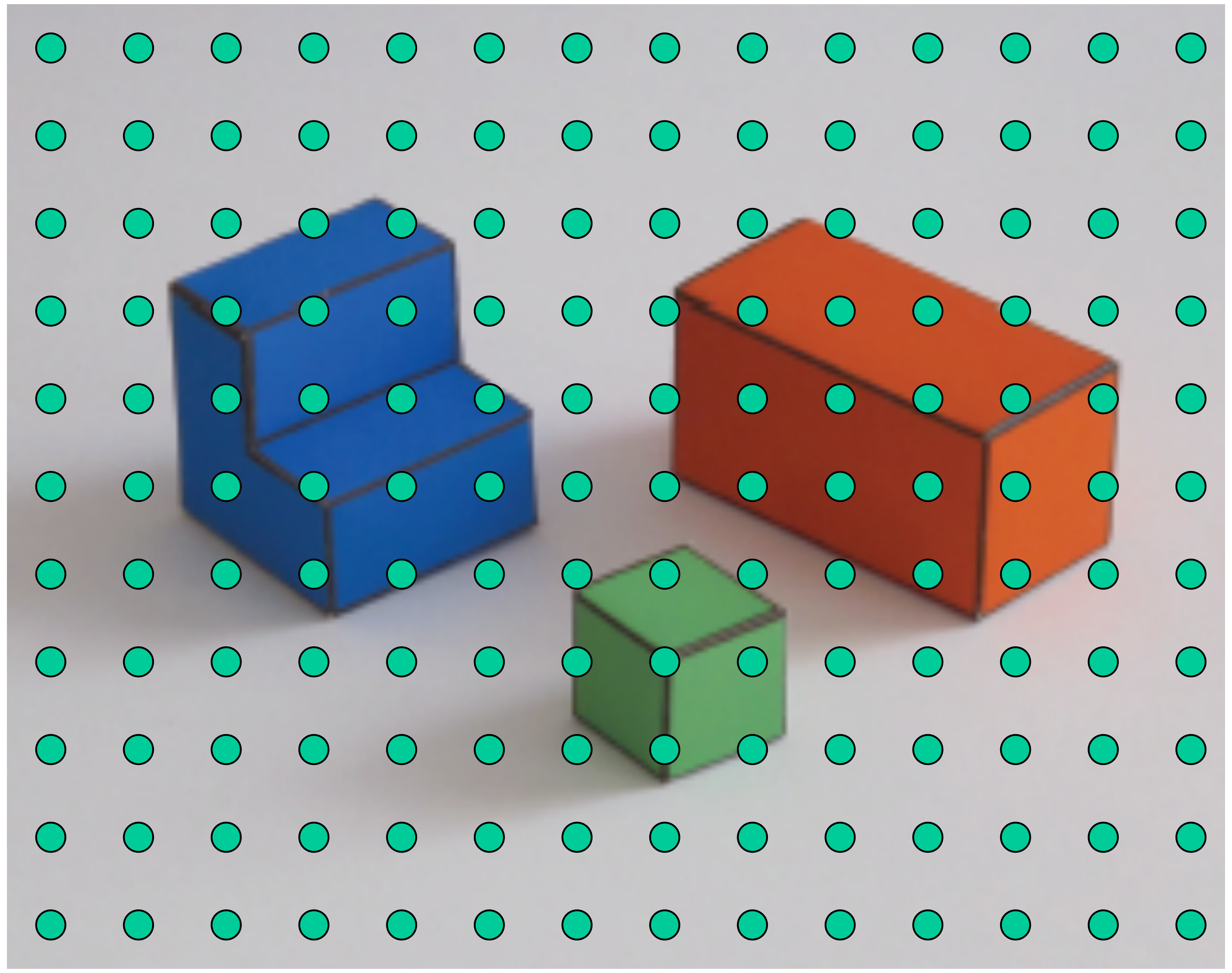
Sampling



Sampling



Sampling



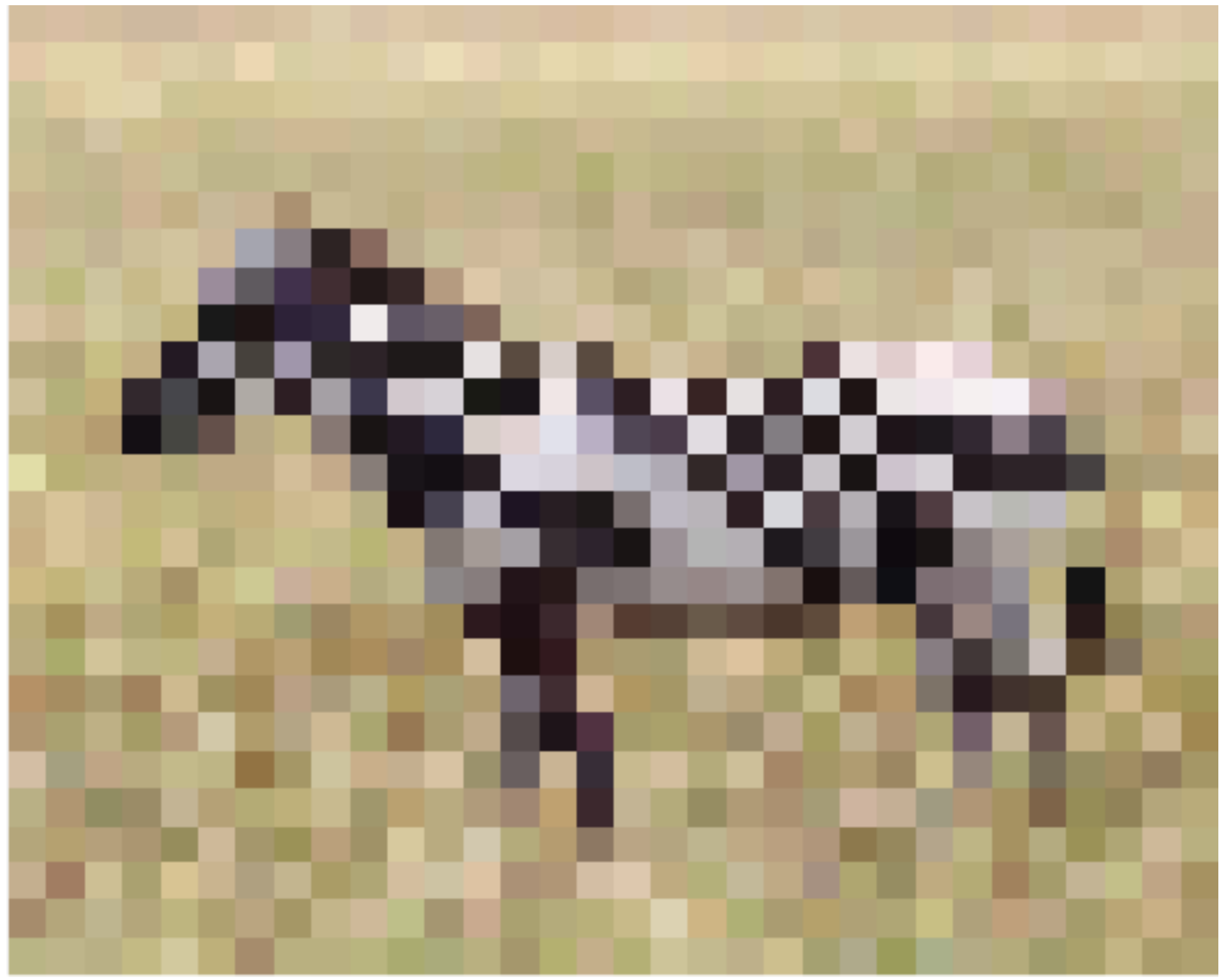
Aliasing

Let's start with this continuous image (it is not really continuous…)

Aliasing

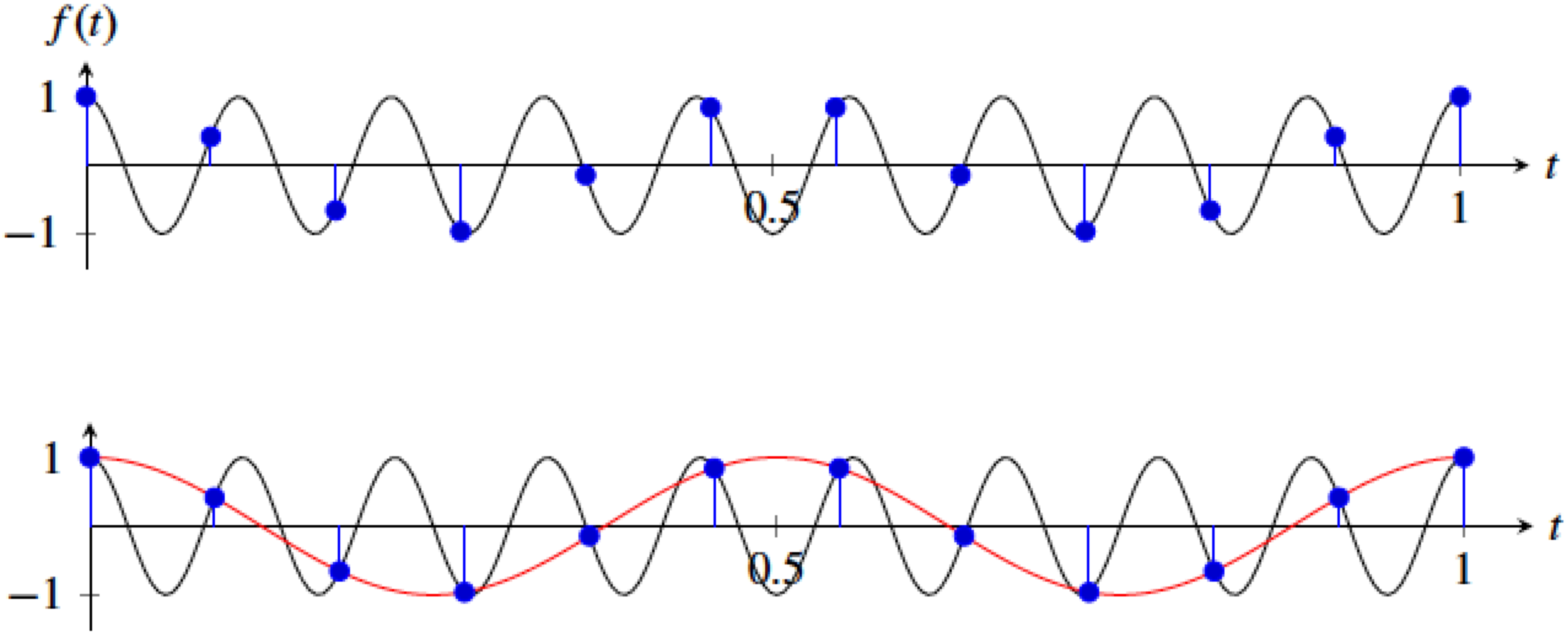
103×128

52×64



26×32

Aliasing

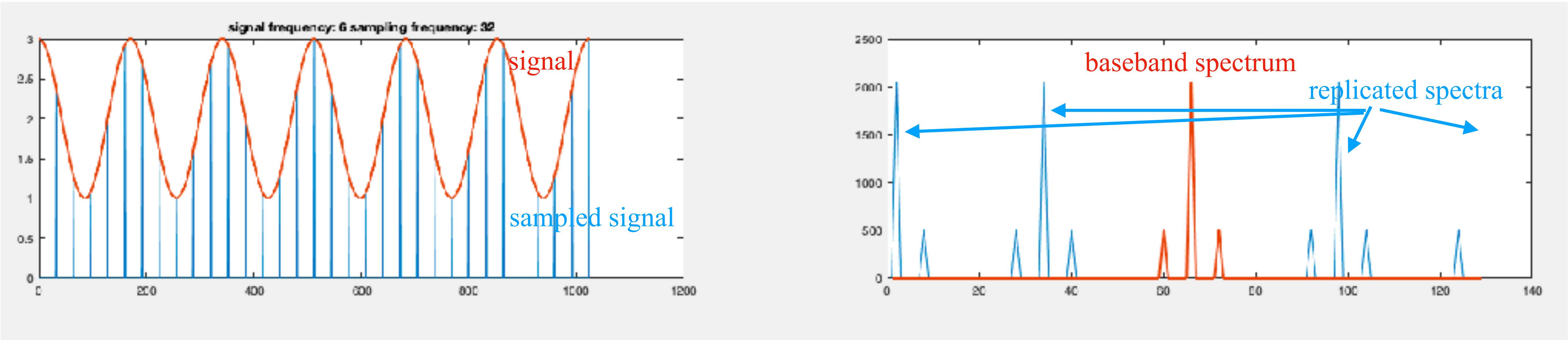


Both waves fit the same samples. Aliasing consists in “perceiving” the red wave when the actual input was the blue wave.

Red curve is the signal: sinusoid + constant

Blue shows sampled signal

spatial domain

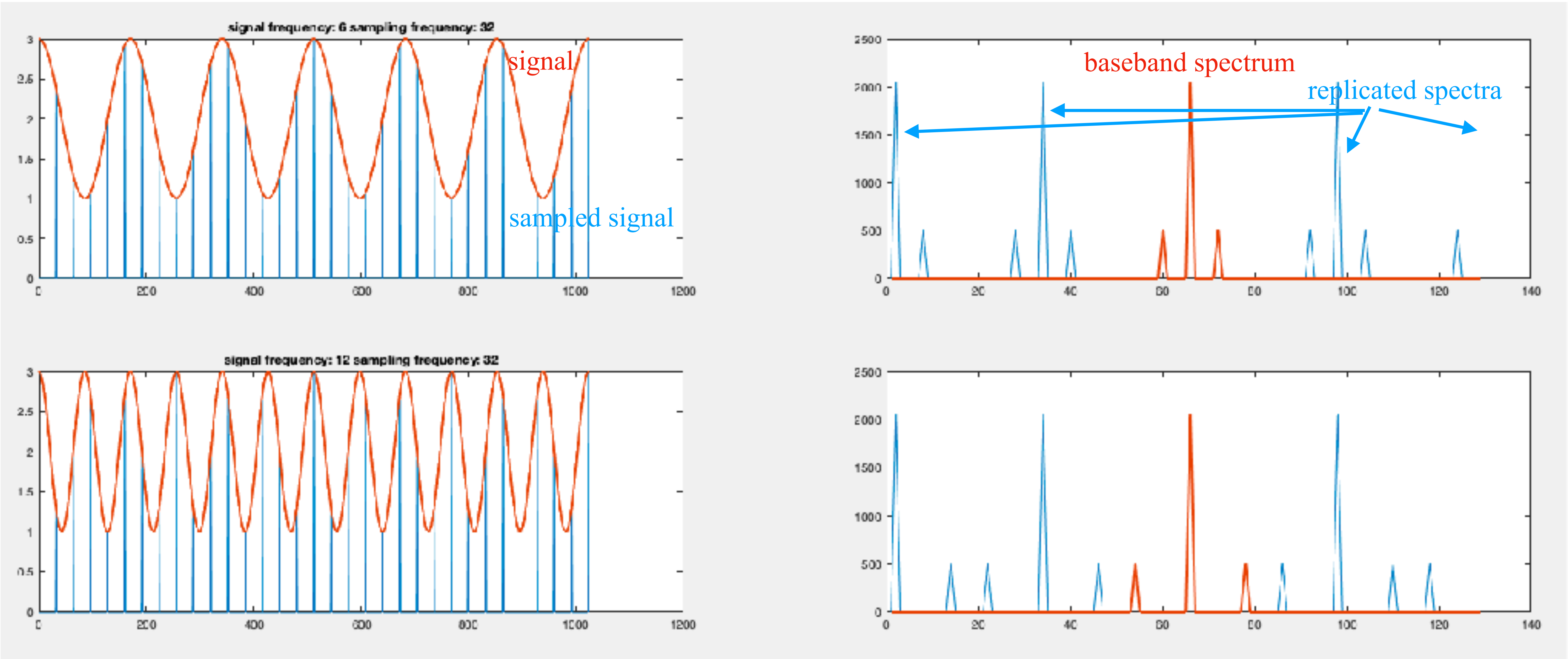


frequency domain

Red curve is the signal: sinusoid + constant

Blue shows sampled signal

spatial domain



frequency domain

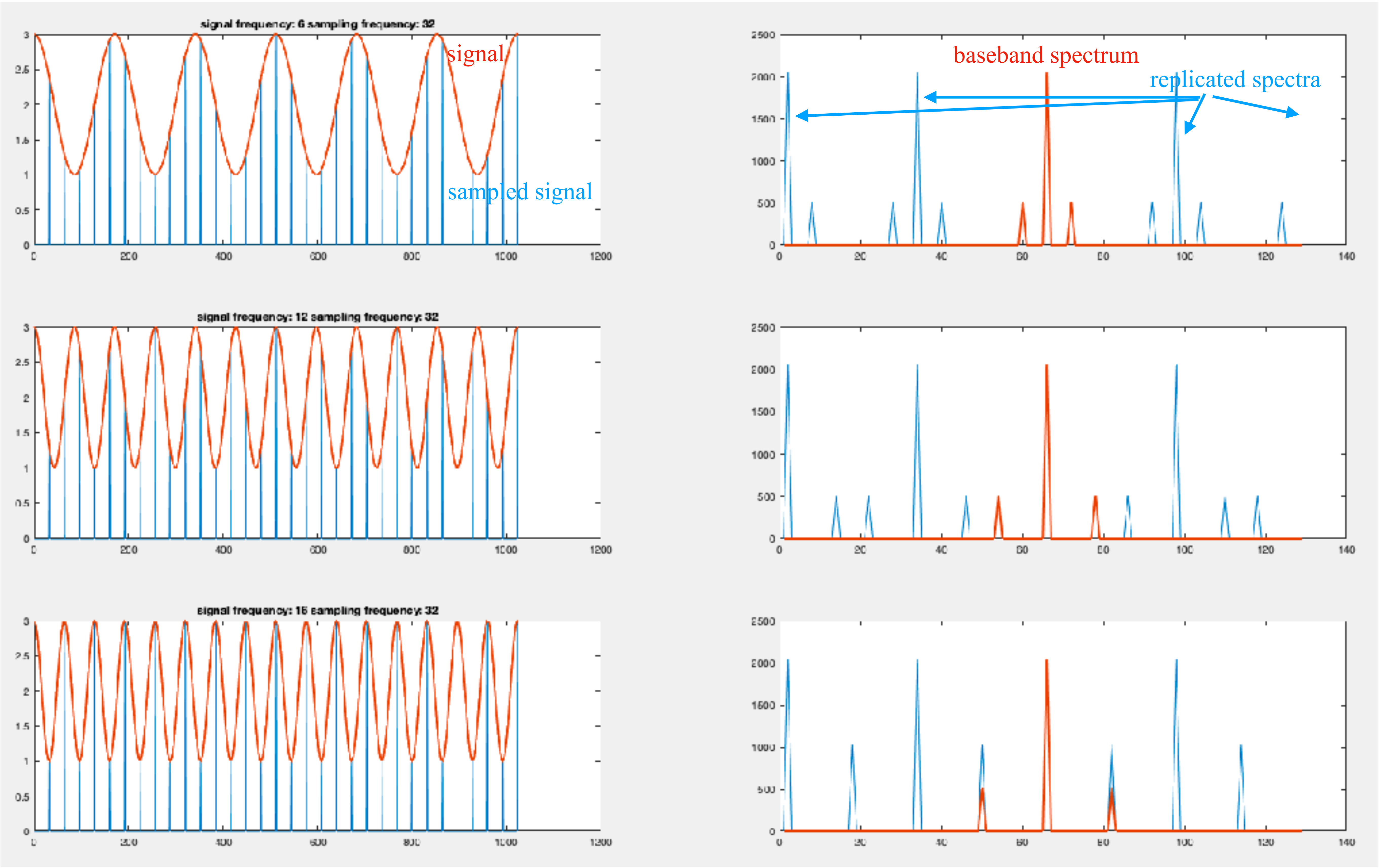
Red curve is the signal: sinusoid + constant

Blue shows sampled signal

spatial domain

frequency domain

sampled at Nyquist frequency

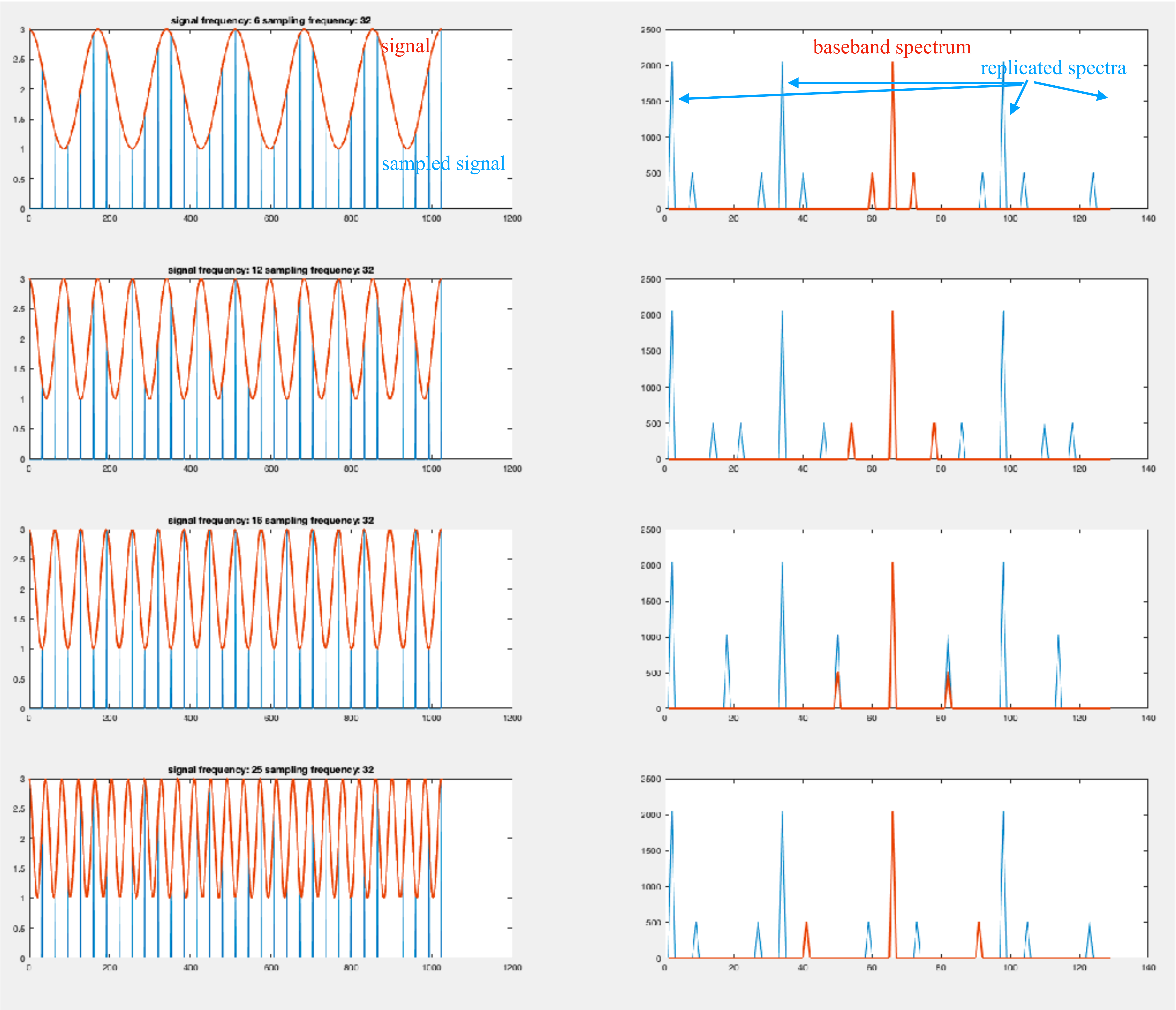


Red curve is the signal: sinusoid + constant

Blue shows sampled signal

spatial domain

sampled at Nyquist frequency



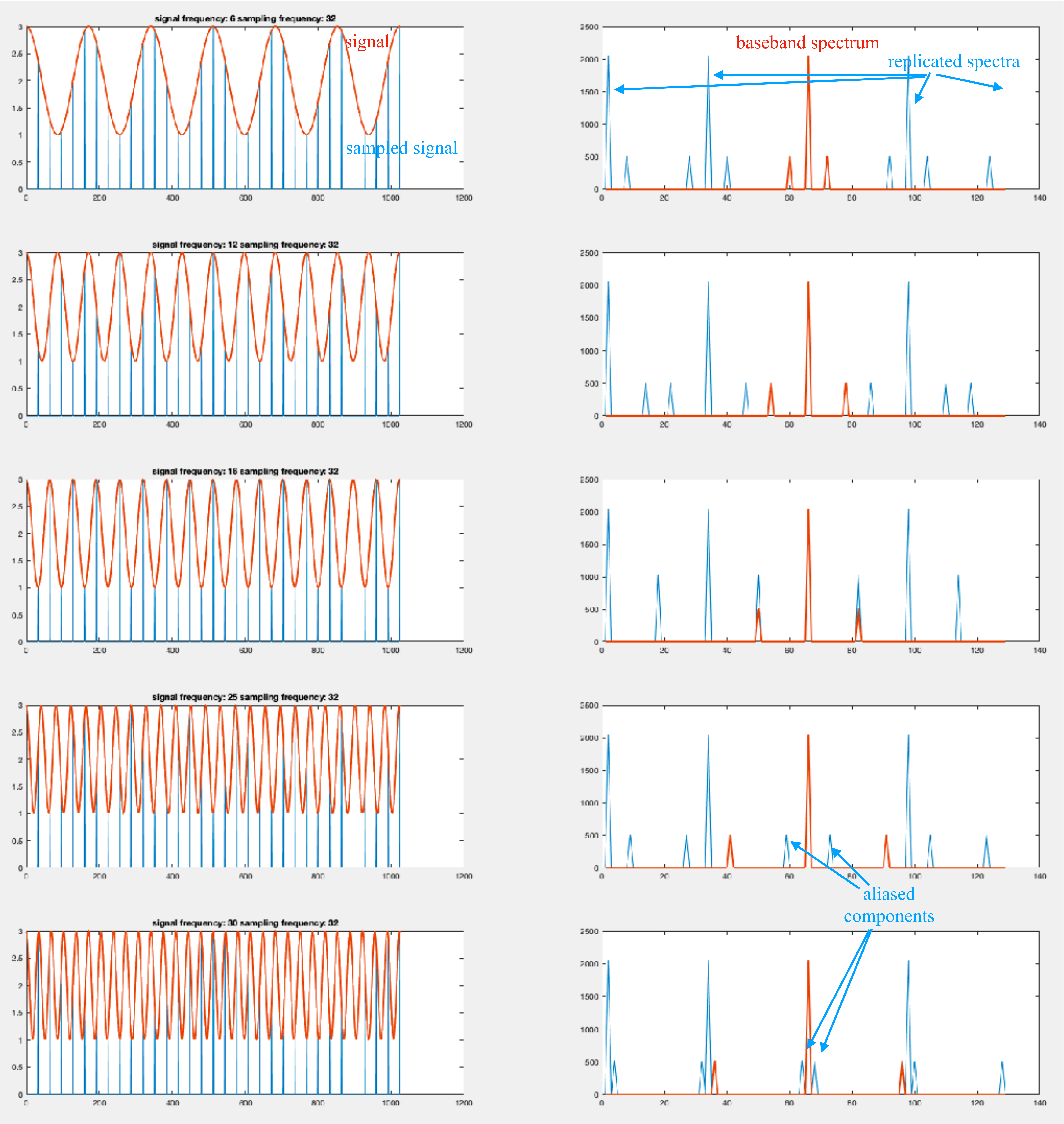
frequency domain

Red curve is the signal: sinusoid + constant

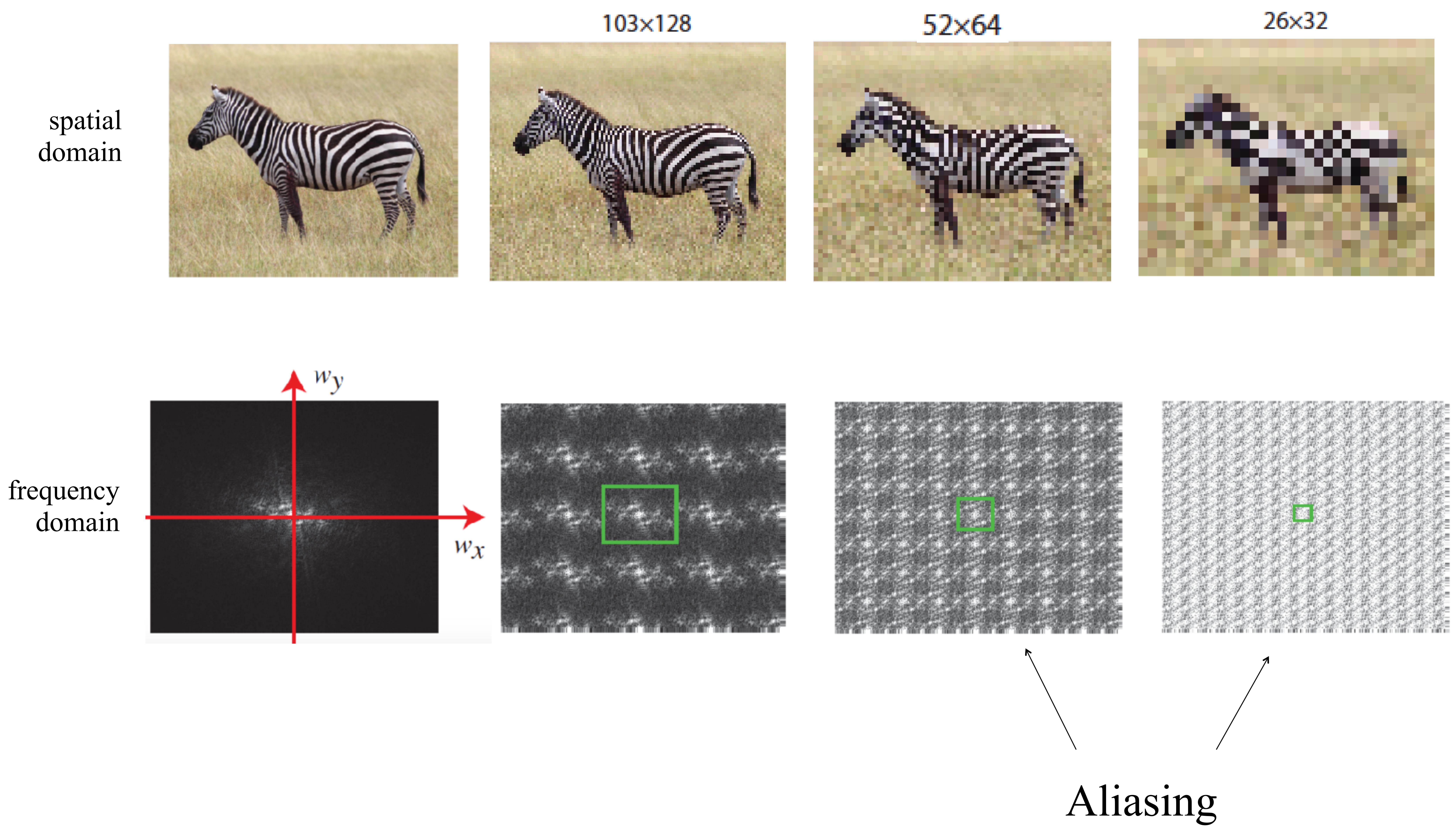
Blue shows sampled signal

spatial domain

sampled at Nyquist frequency



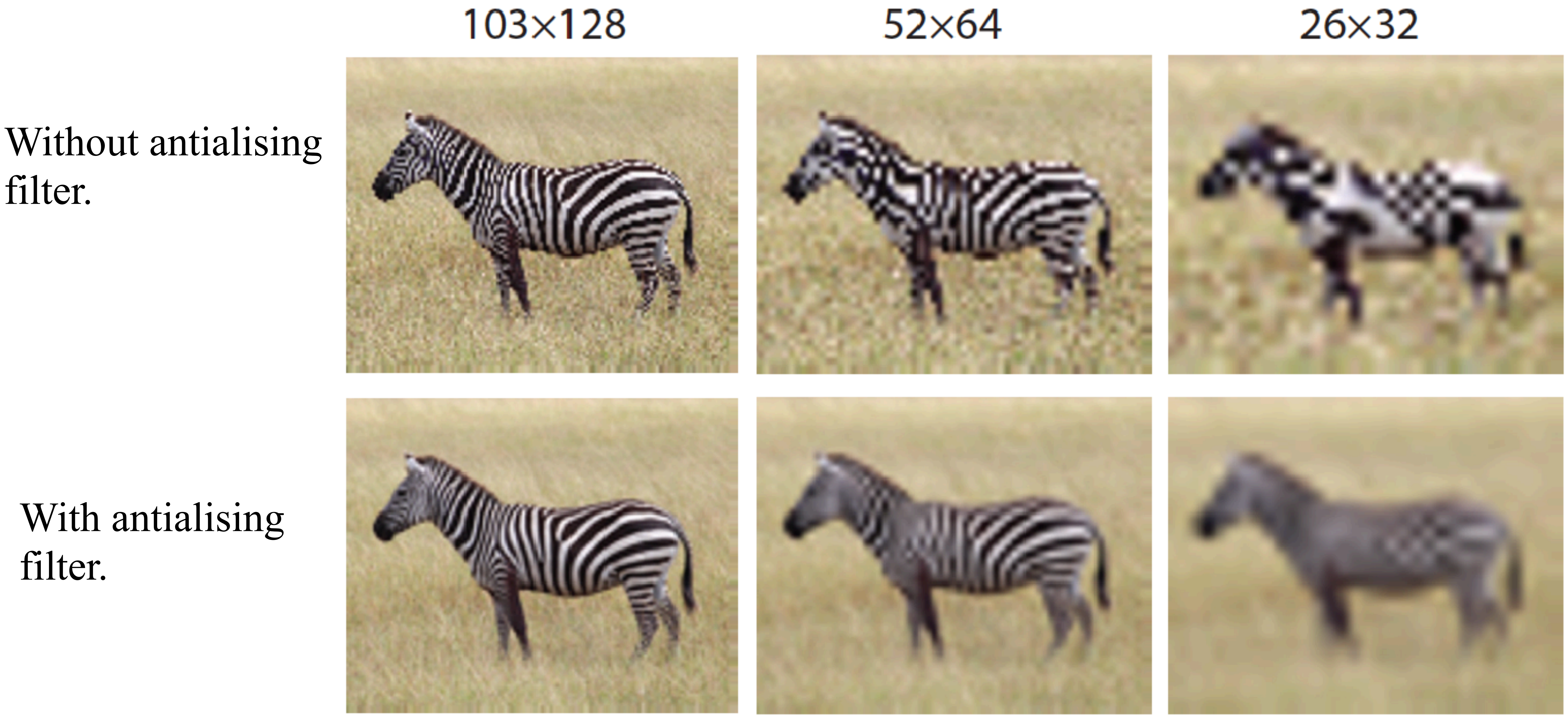
frequency domain



103x128
52x64
26x32
spatial domain
frequency domain
w_y
w_x
Aliasing

Antialising filtering

Before sampling, apply a low pass-filter to remove all the frequencies that will produce aliasing: “blur before you subsample"

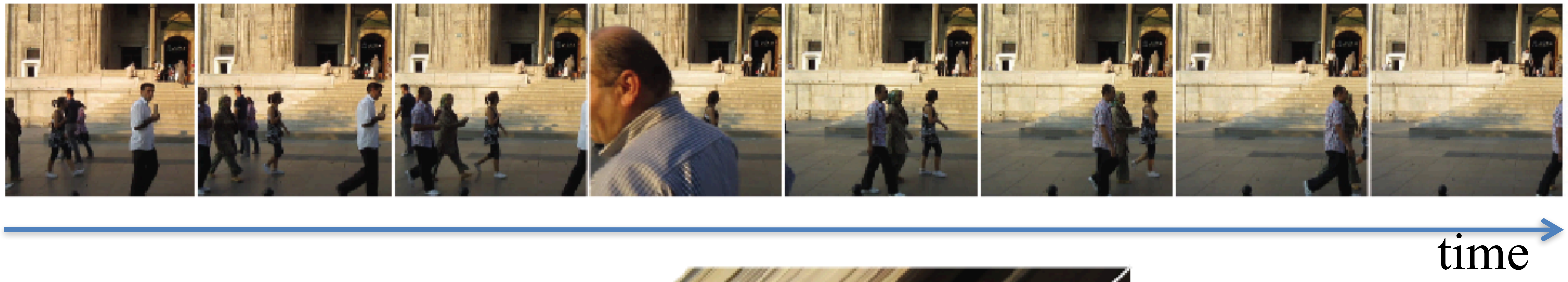


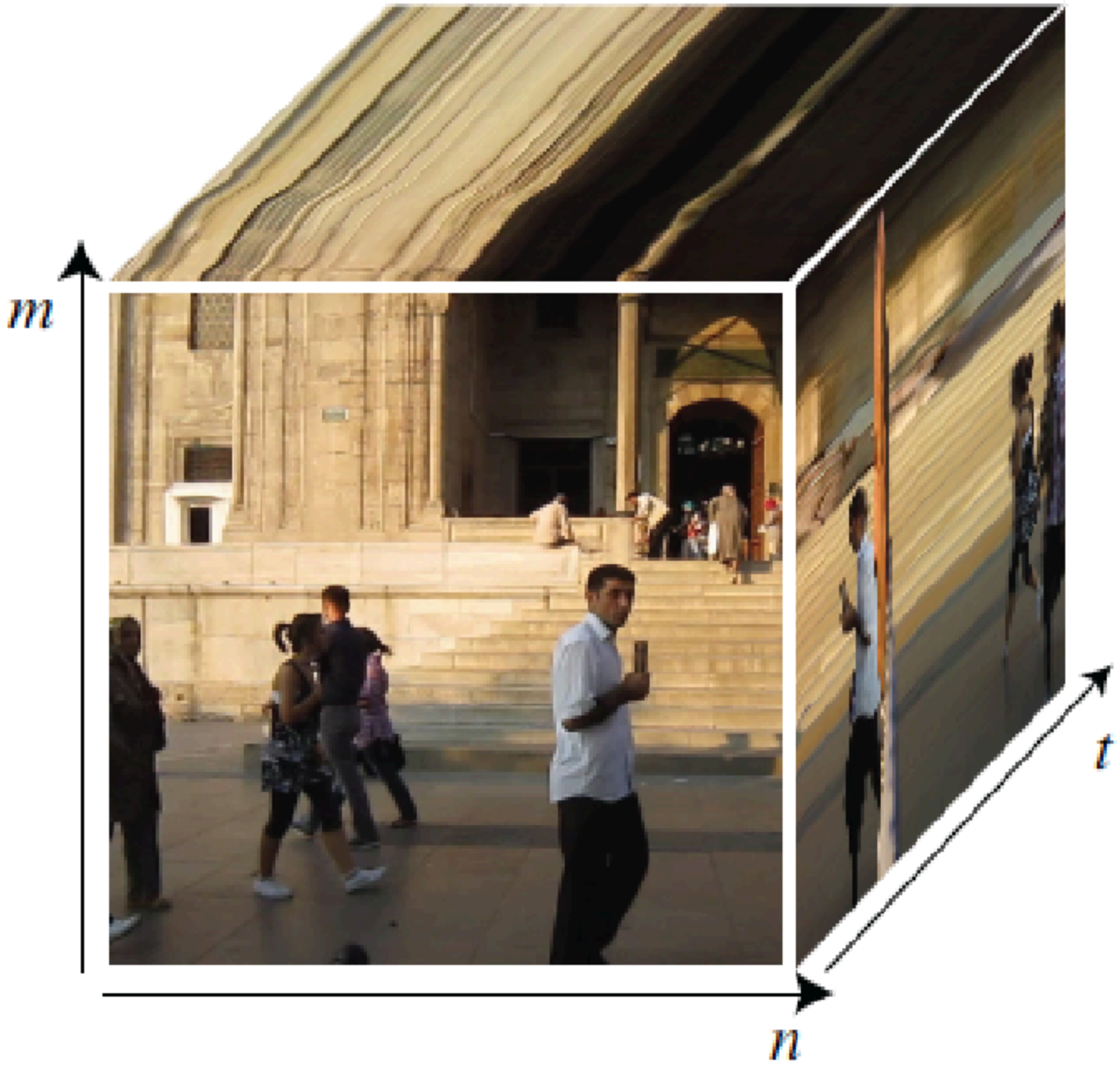
- **Temporal filtering**
- **Motion illusion**, involving aliasing, addressing whether humans match spatial patterns, or use temporal filters, to measure motion.

Temporal filtering

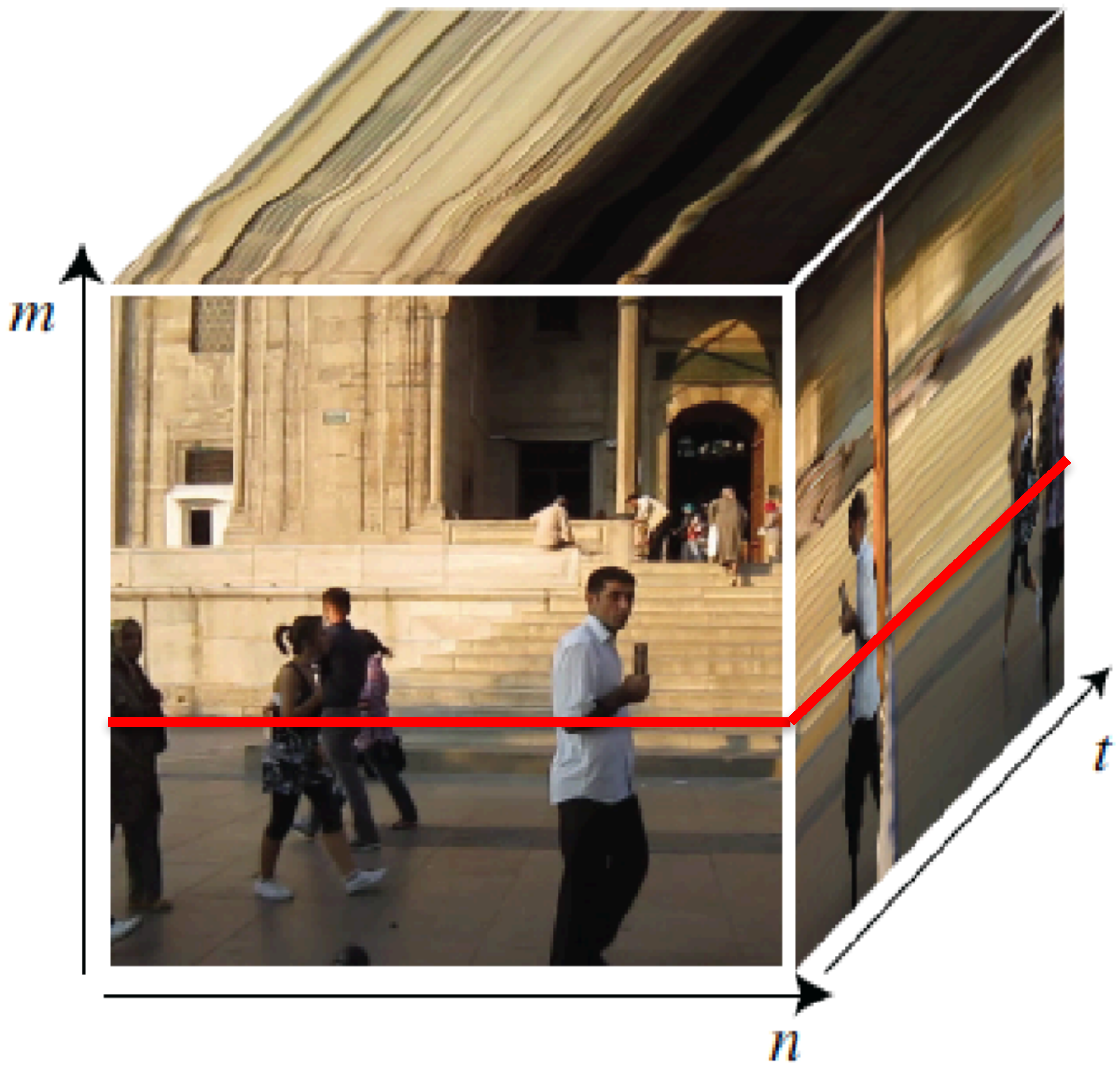
why filter videos over time?

Sequences

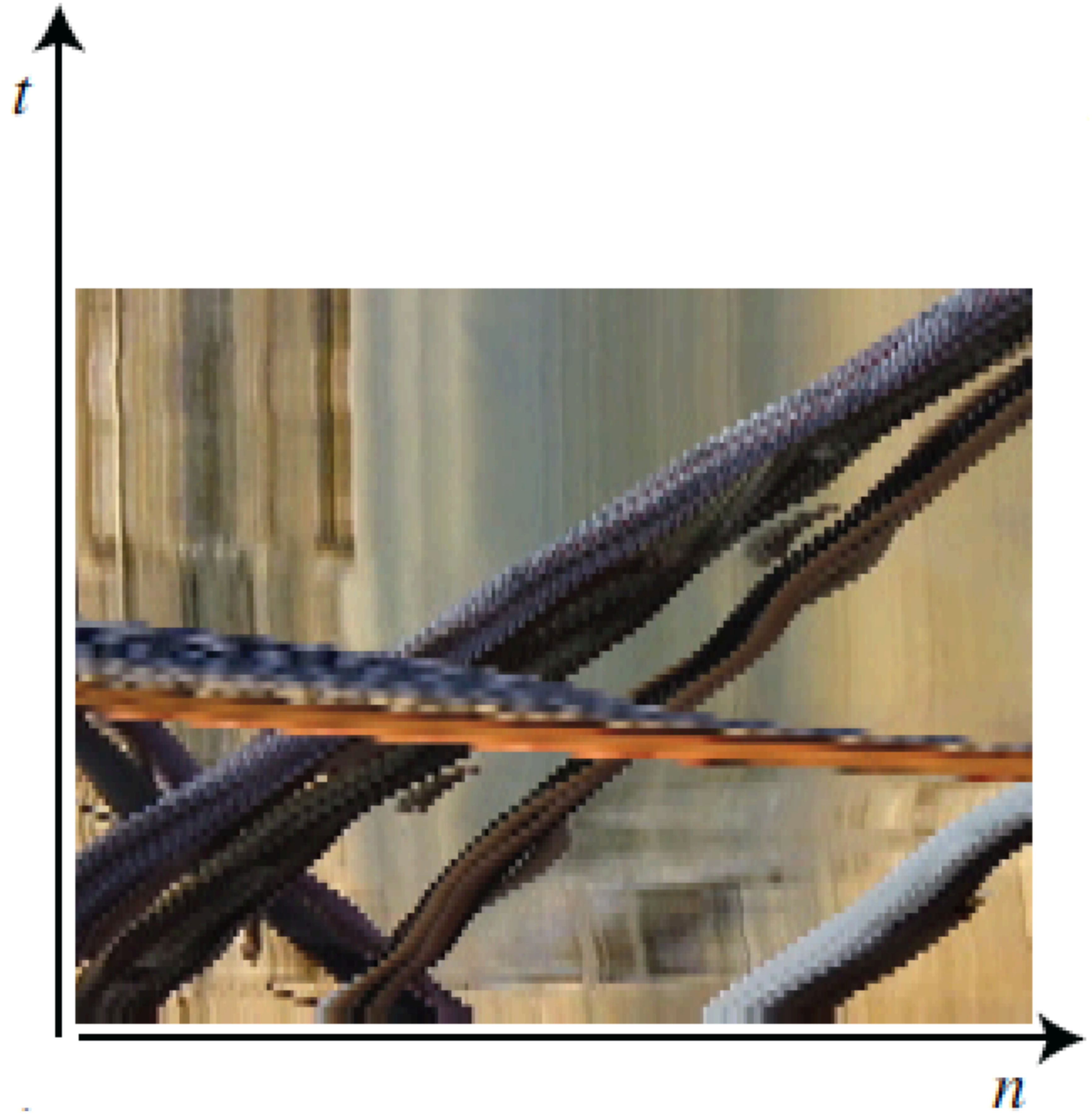




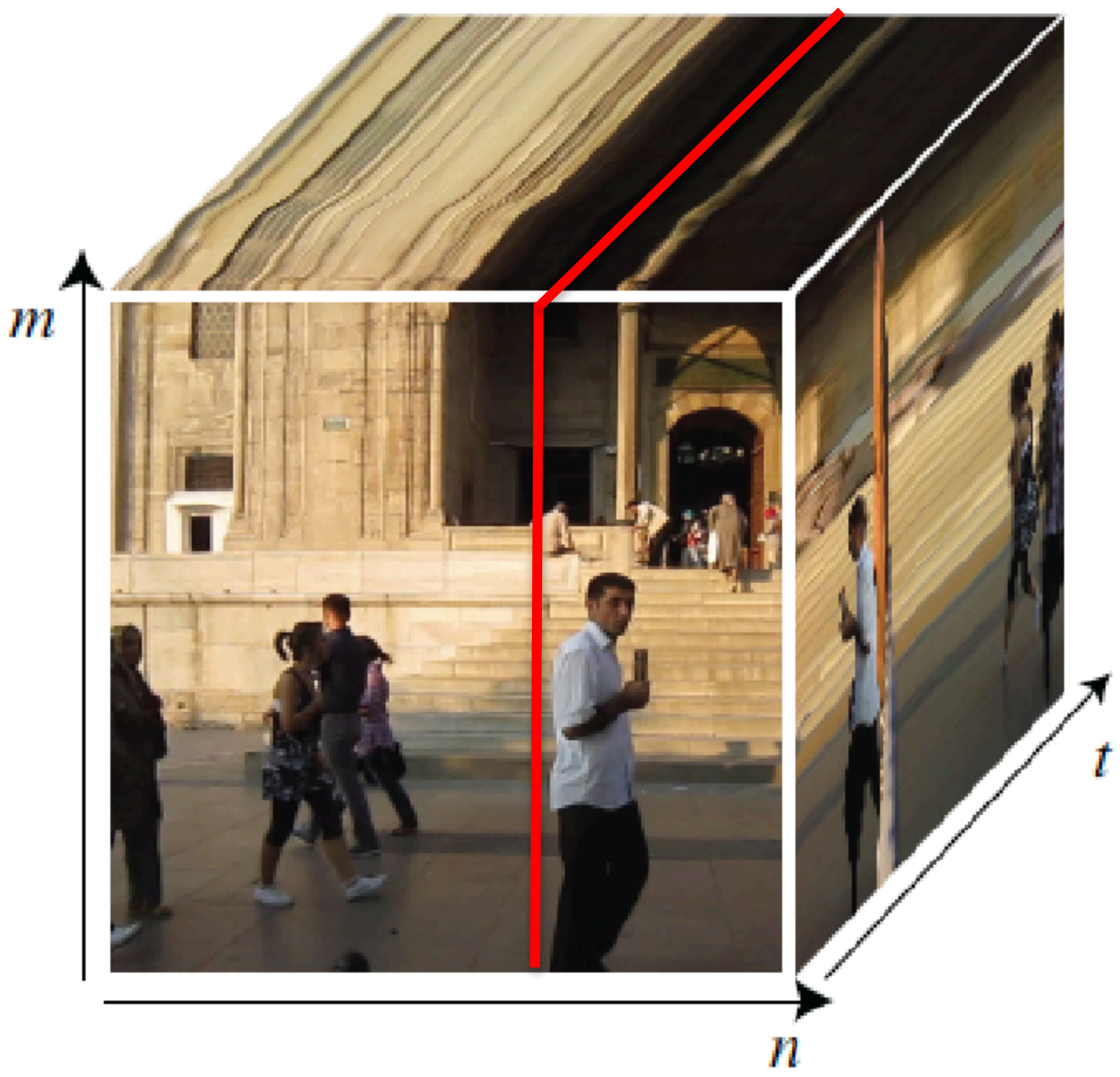
Sequences



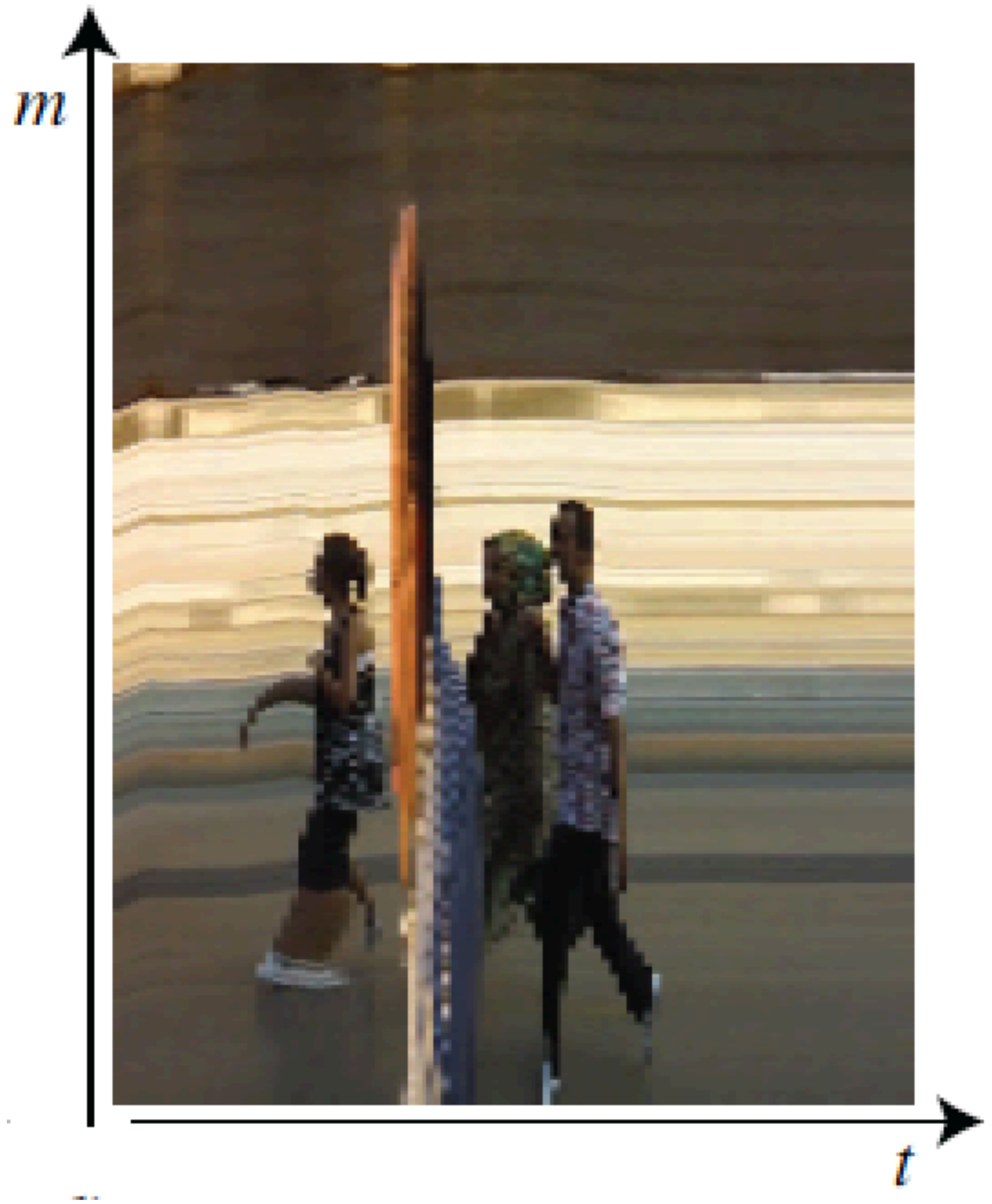
Cube size = 128x128x90



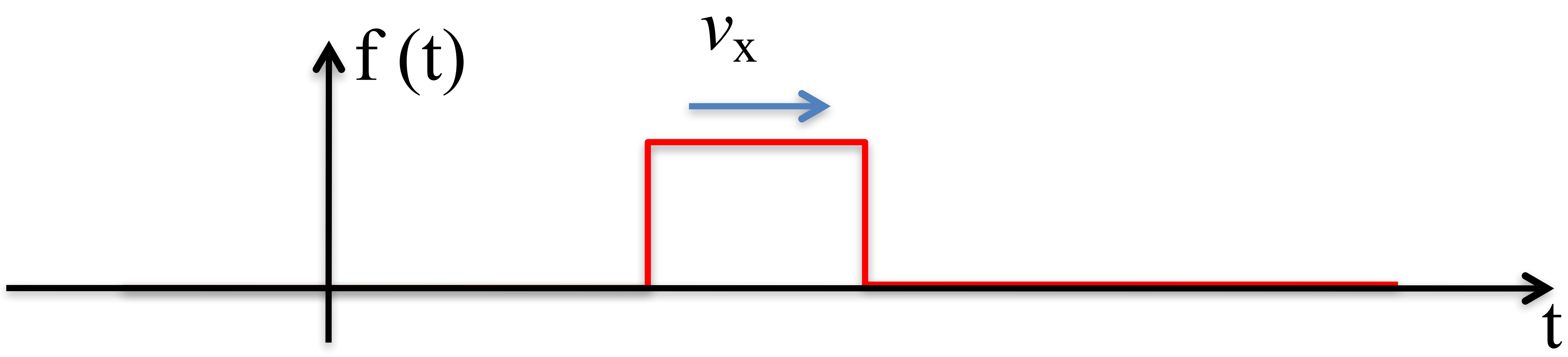
Sequences

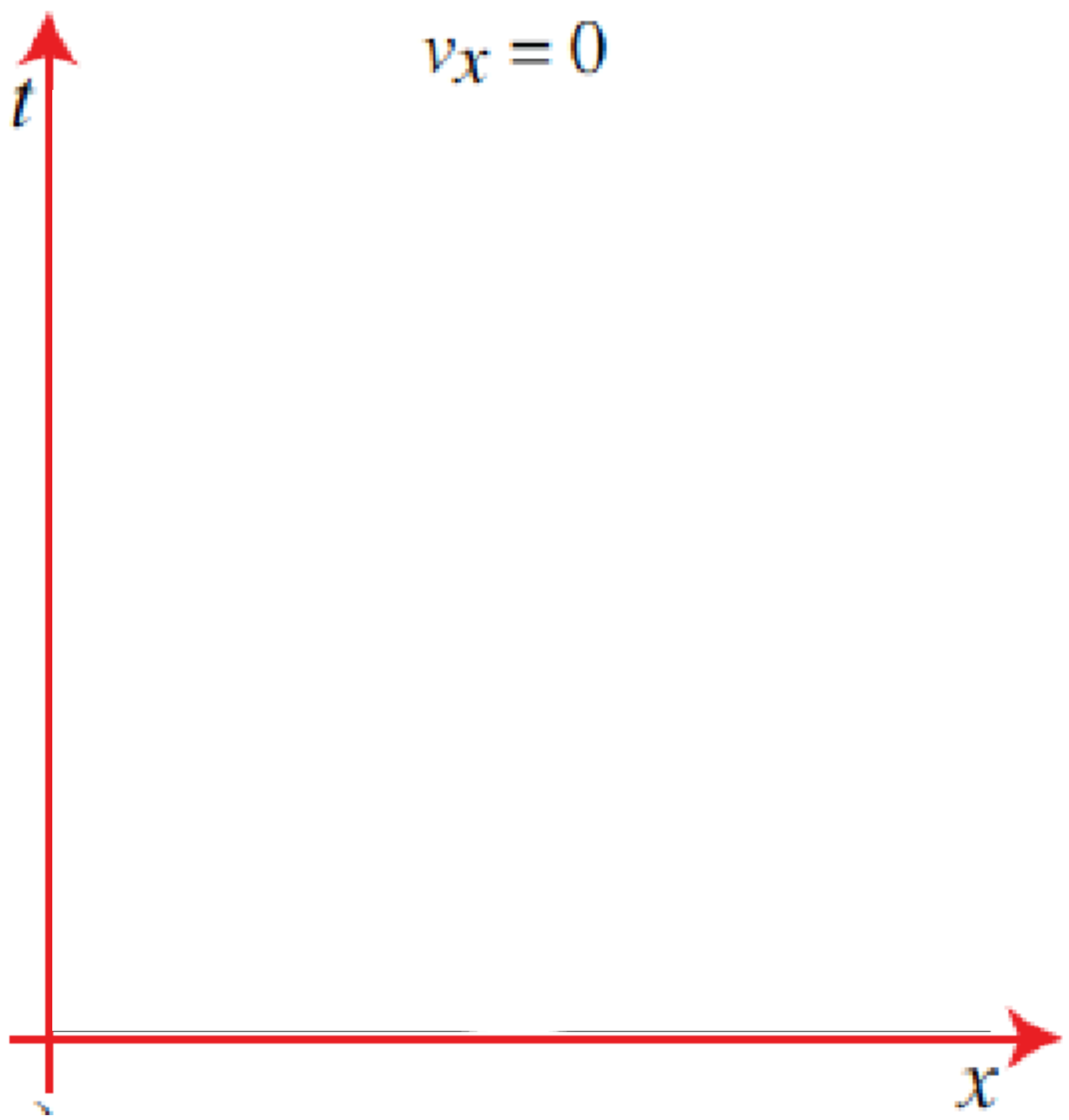


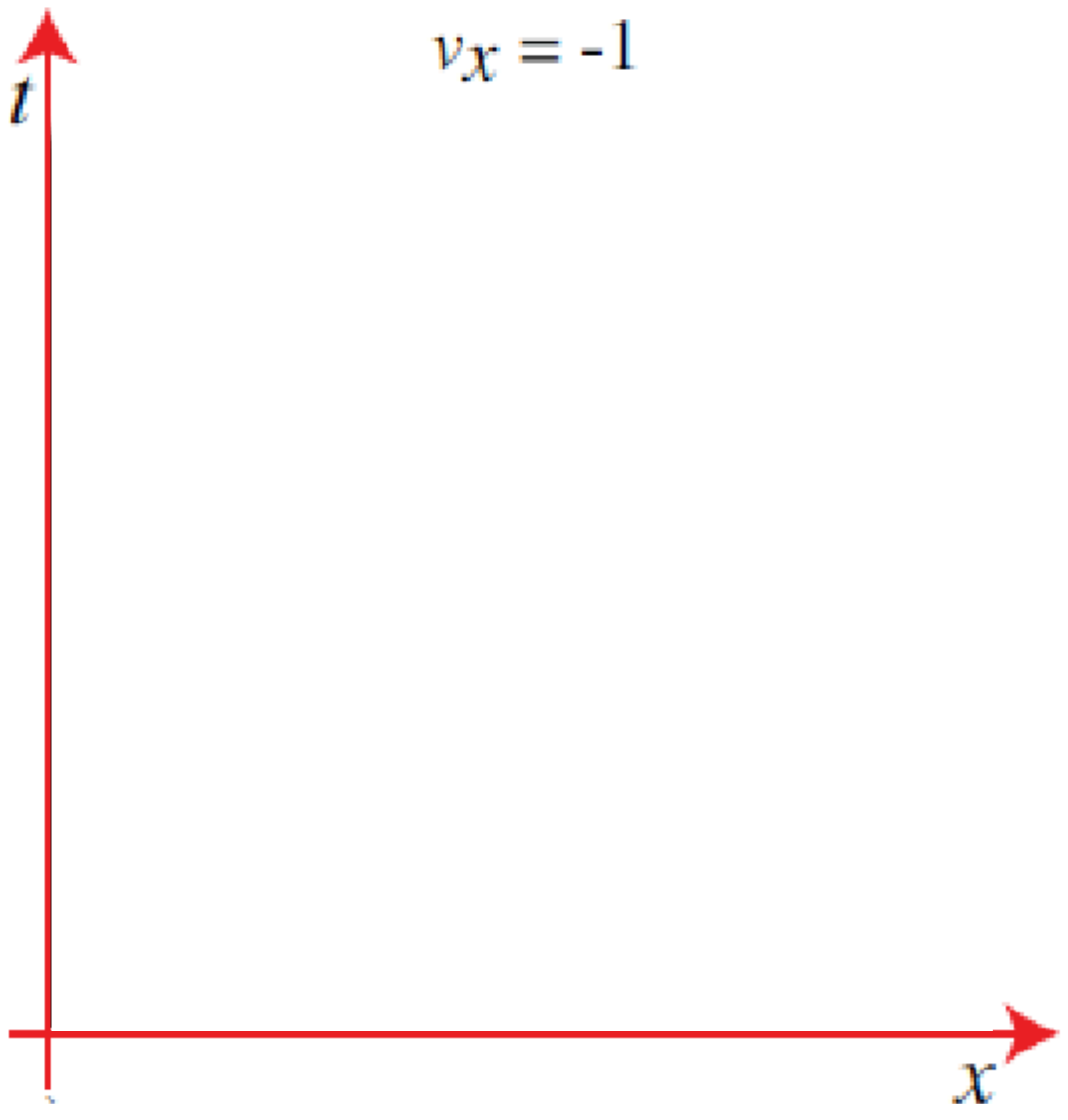
Cube size = 128x128x90



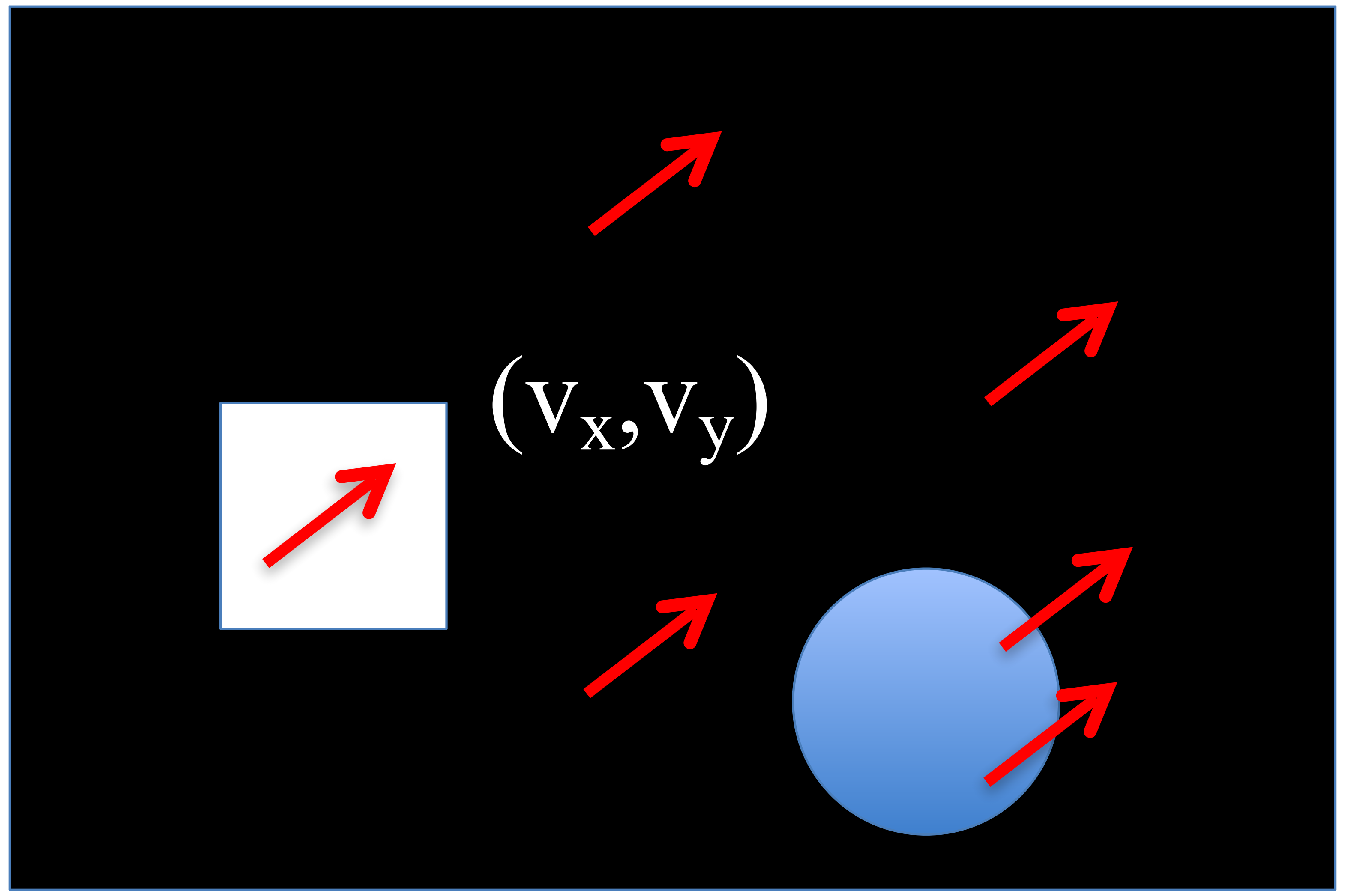
A box moving with speed v_x







Global constant motion



A global motion of the image can be written as:

$$f(x,y,t) = f_0(x - v_x t, y - v_y t)$$

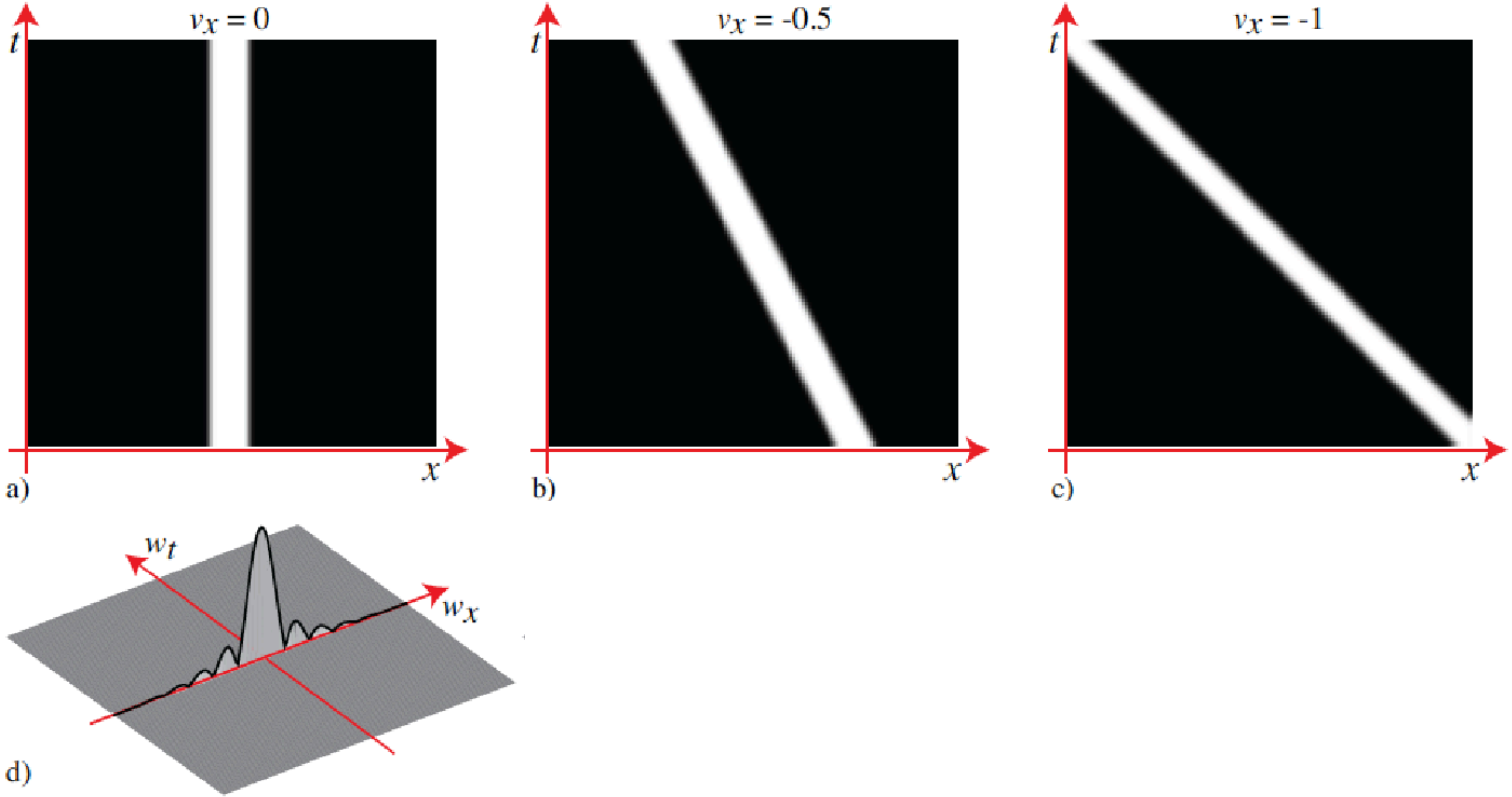
Where:

$$f_0(x,y) = f(x,y,0)$$

$$f(x,y,t) = f_0(x - v_x t, y - v_y t)$$

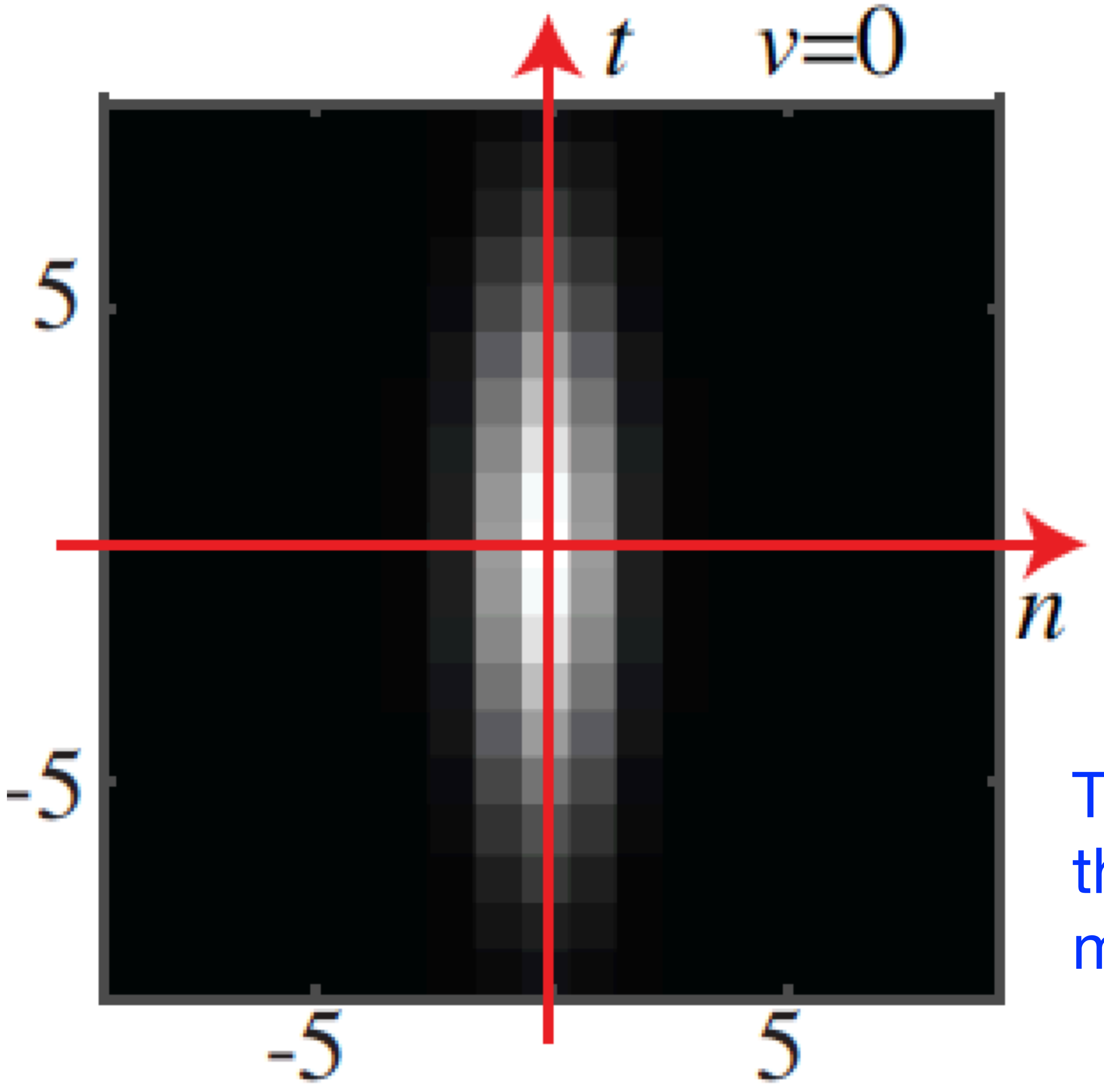
$$\updownarrow$$

$$F(w_x, w_y, w_t) = F_0(w_x, w_y)\delta(w_t + v_x w_x + v_y w_y)$$



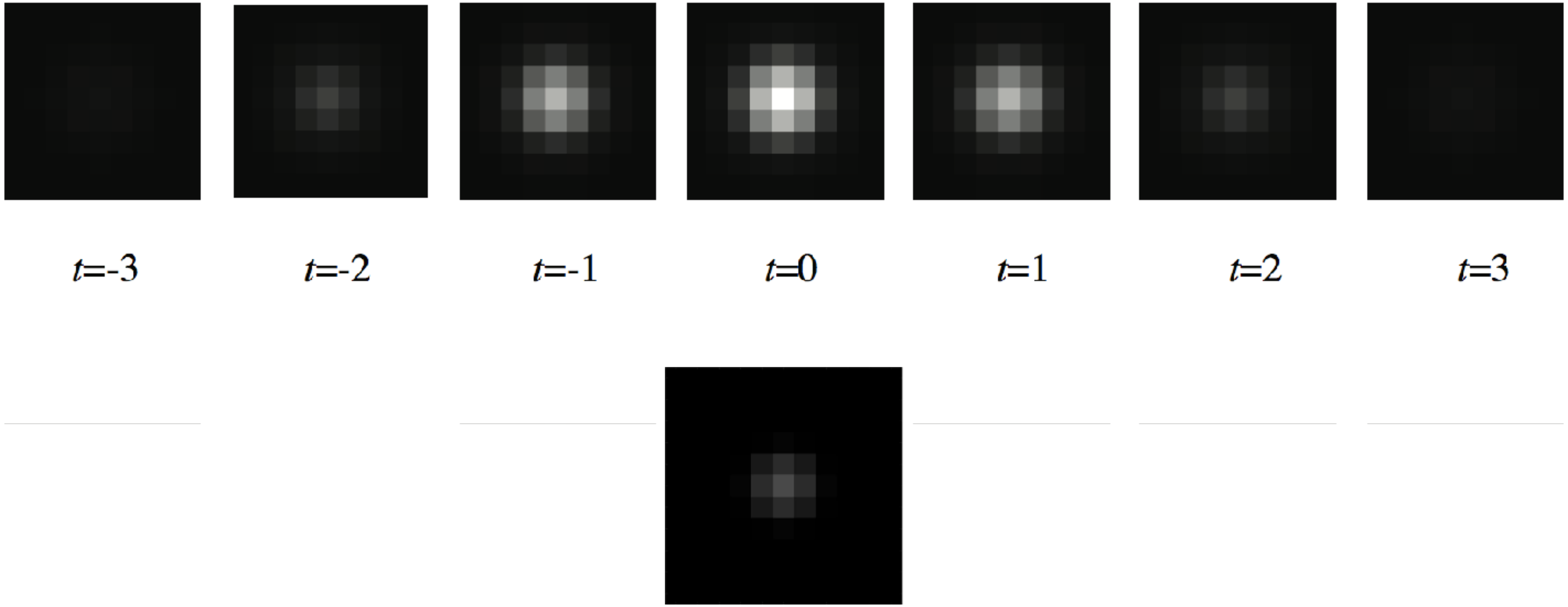
Temporal Gaussian

$$g(x, y, t; \sigma_x, \sigma_t) = \frac{1}{(2\pi)^{3/2}\sigma_x^2\sigma_t} \exp - \frac{x^2 + y^2}{2\sigma_x^2} \exp - \frac{t^2}{2\sigma_t^2}$$



This filter keeps stationary things sharp, and blurs moving things.

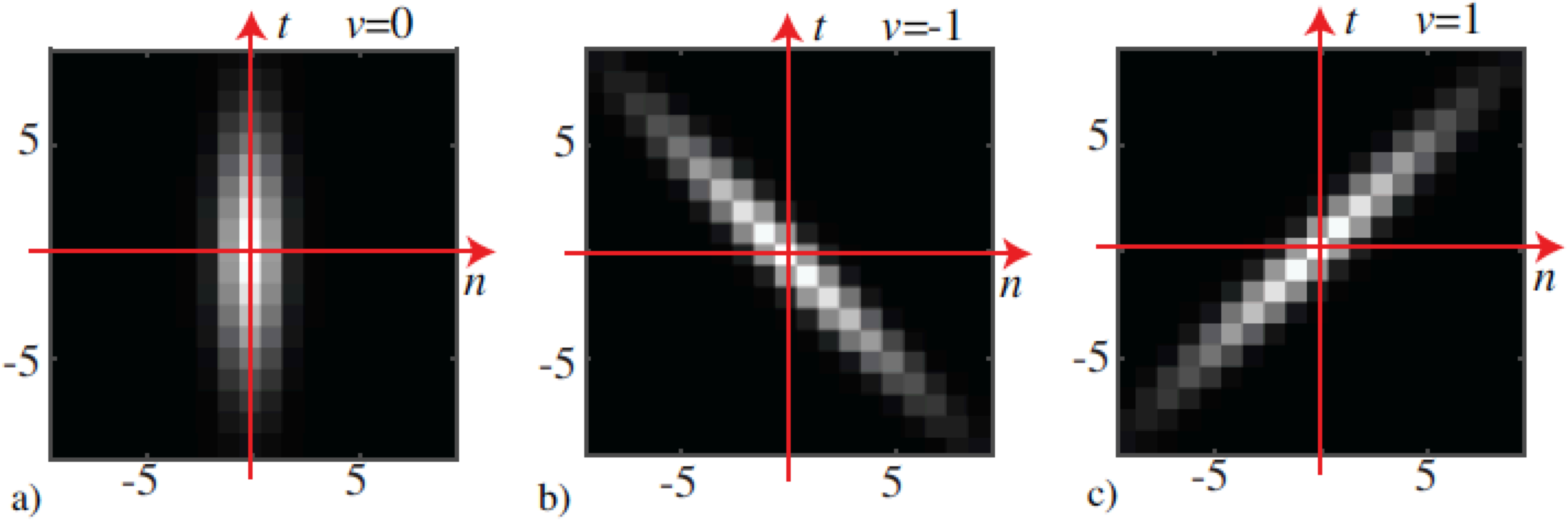
Spatio-temporal Gaussian

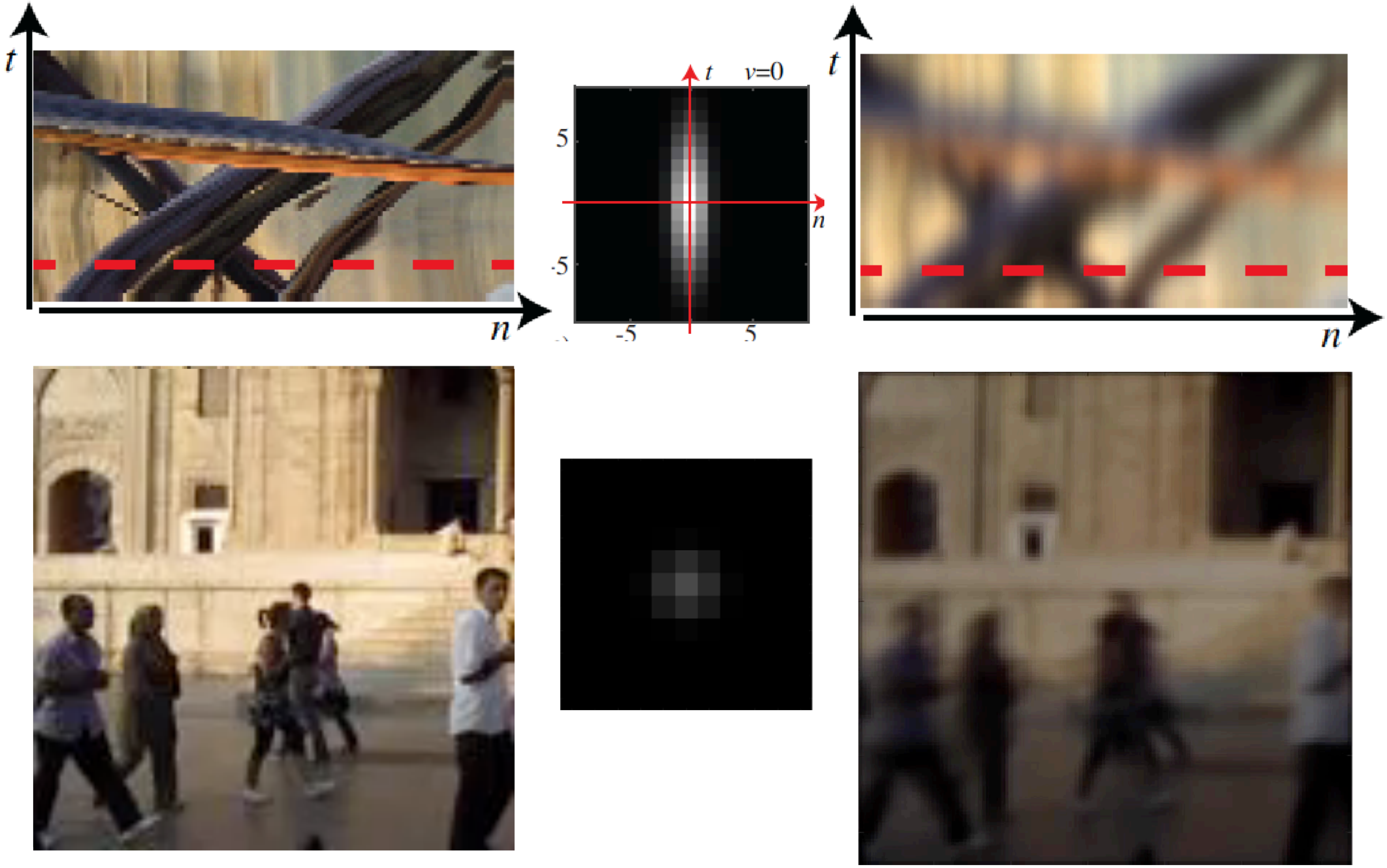


Spatio-temporal Gaussian

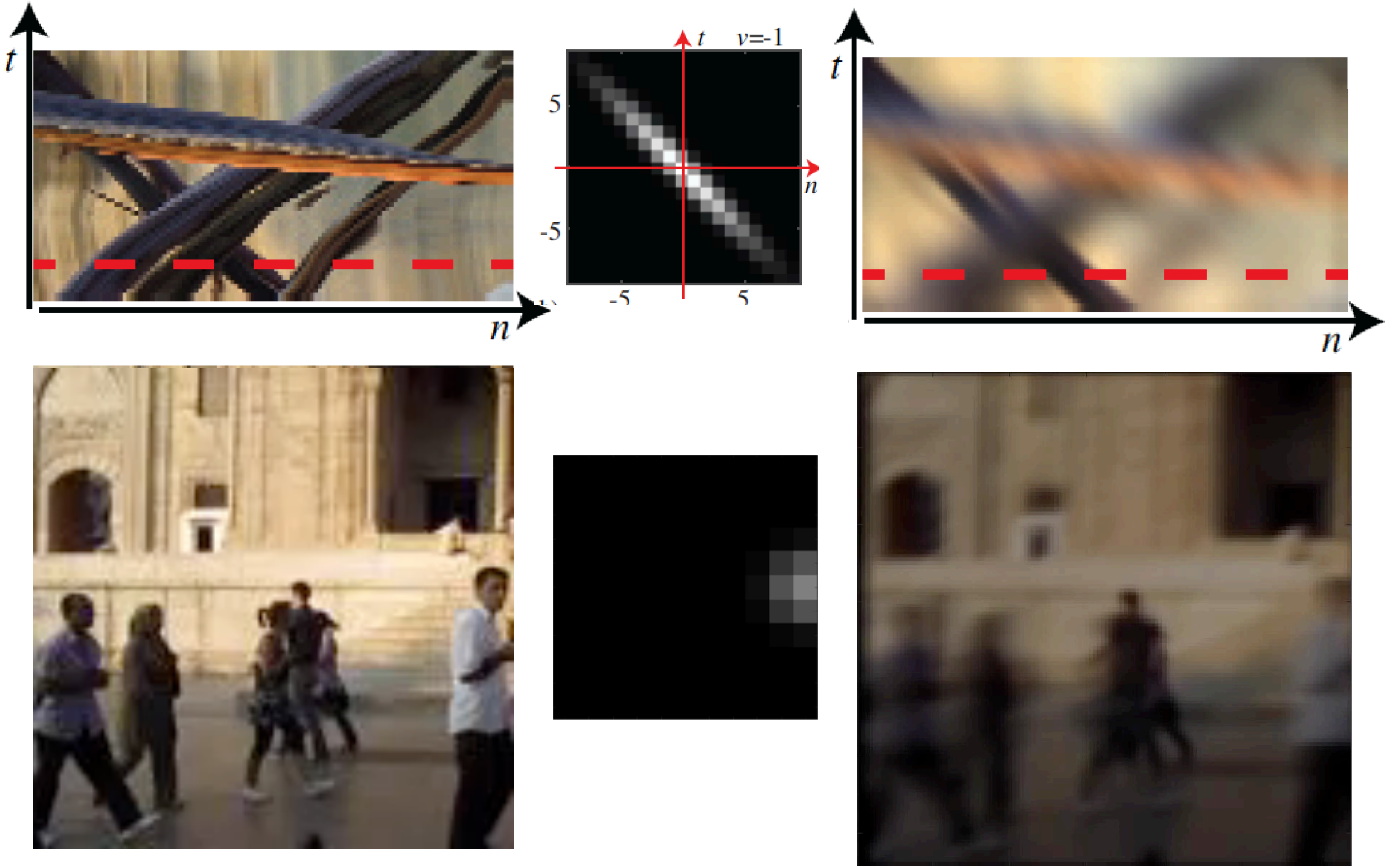
How could we create a filter that keeps sharp objects that move at some velocity **(vx , vy)** while blurring the rest?

$$g_{v_x,v_y}(x,y,t) = g(x - v_x t, y - v_y t, t)$$

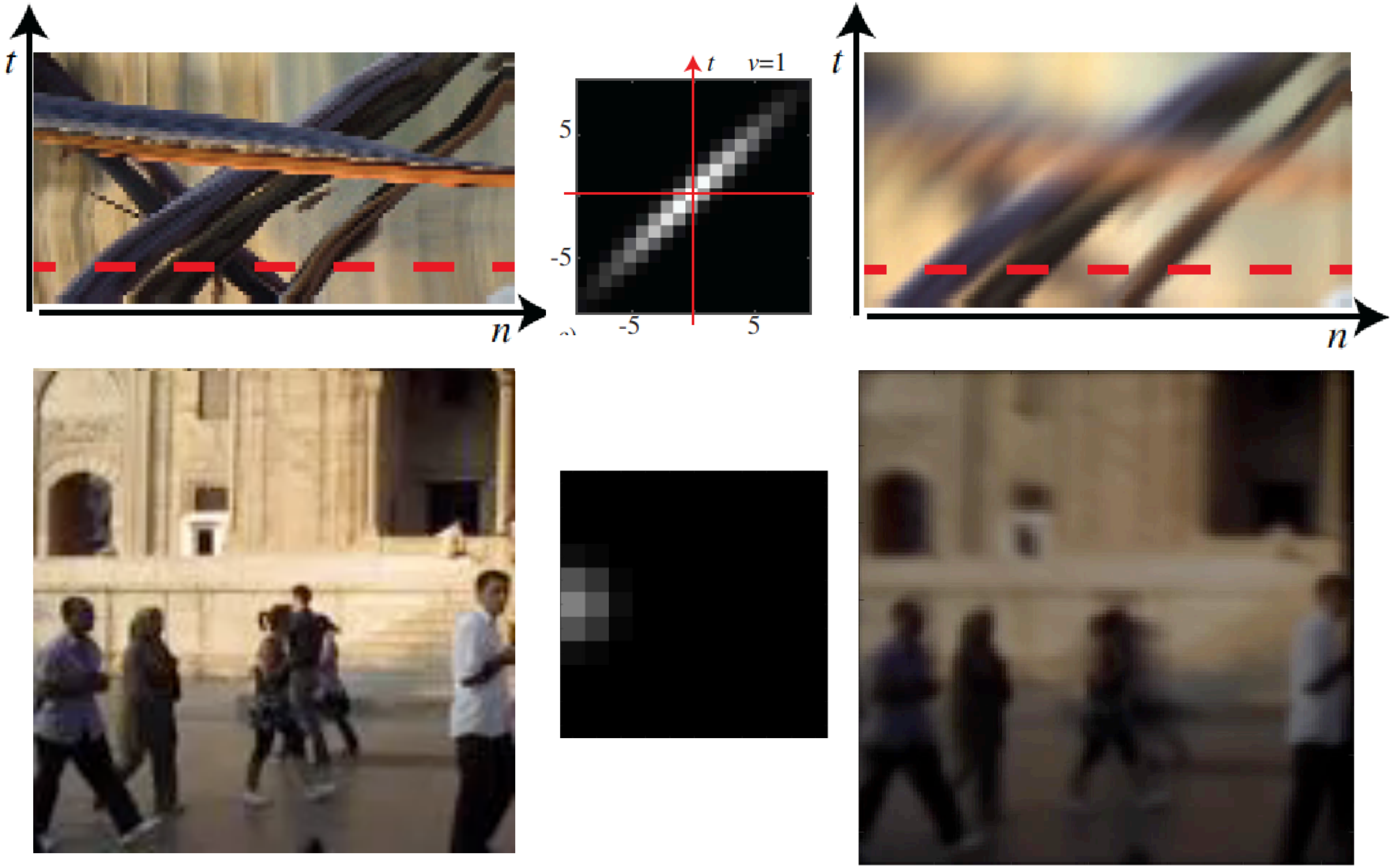




t
n
t
v=0
5
-5
n
-5
5
t
n



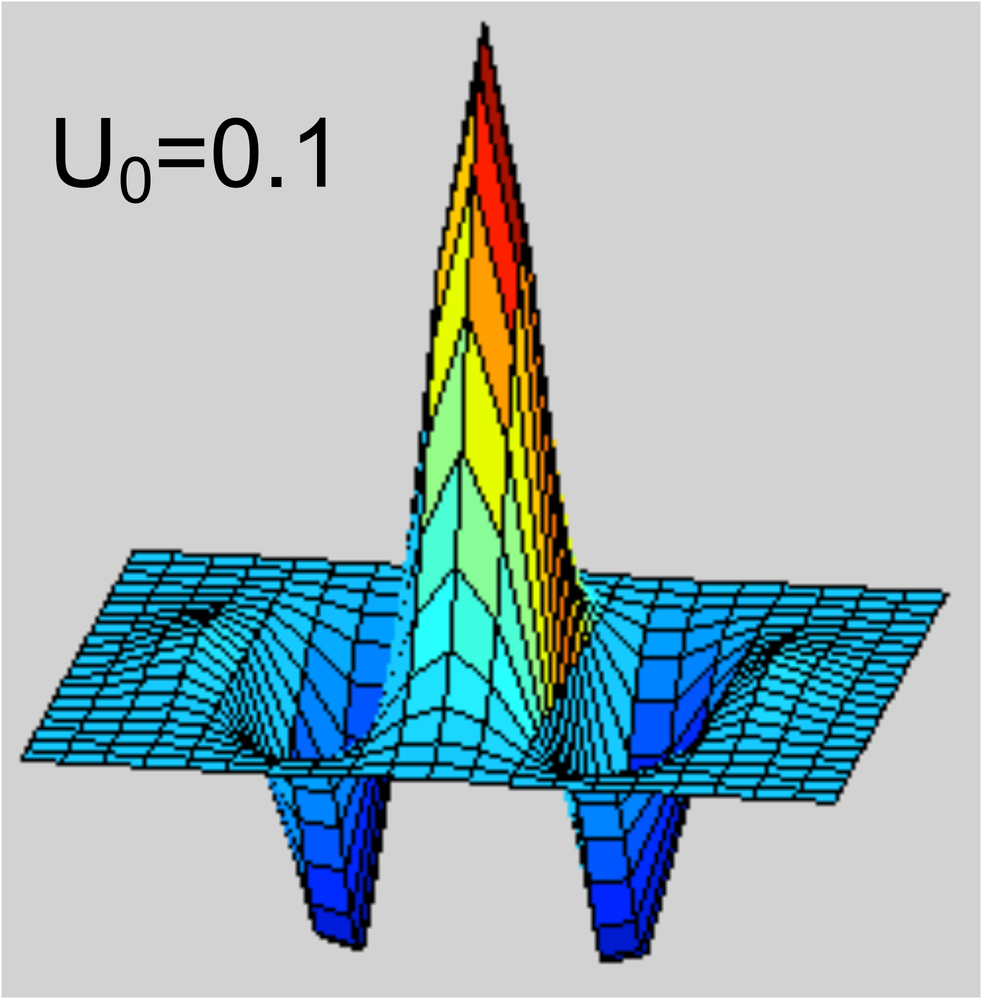
t
n
t
v=-1
5
-5
-5
5
n
t
n



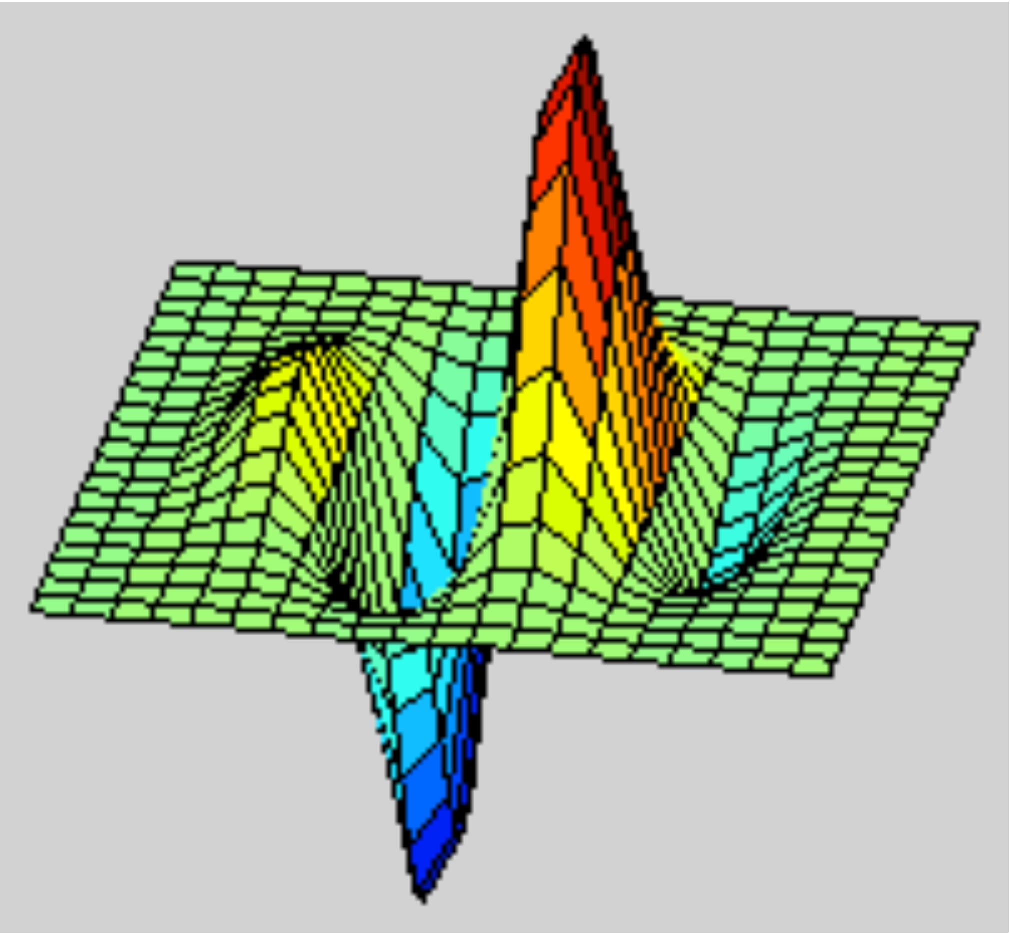
t
n
t
v=1
5
-5
-5
5
t
n

Quadrature pair of Gabor filters

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

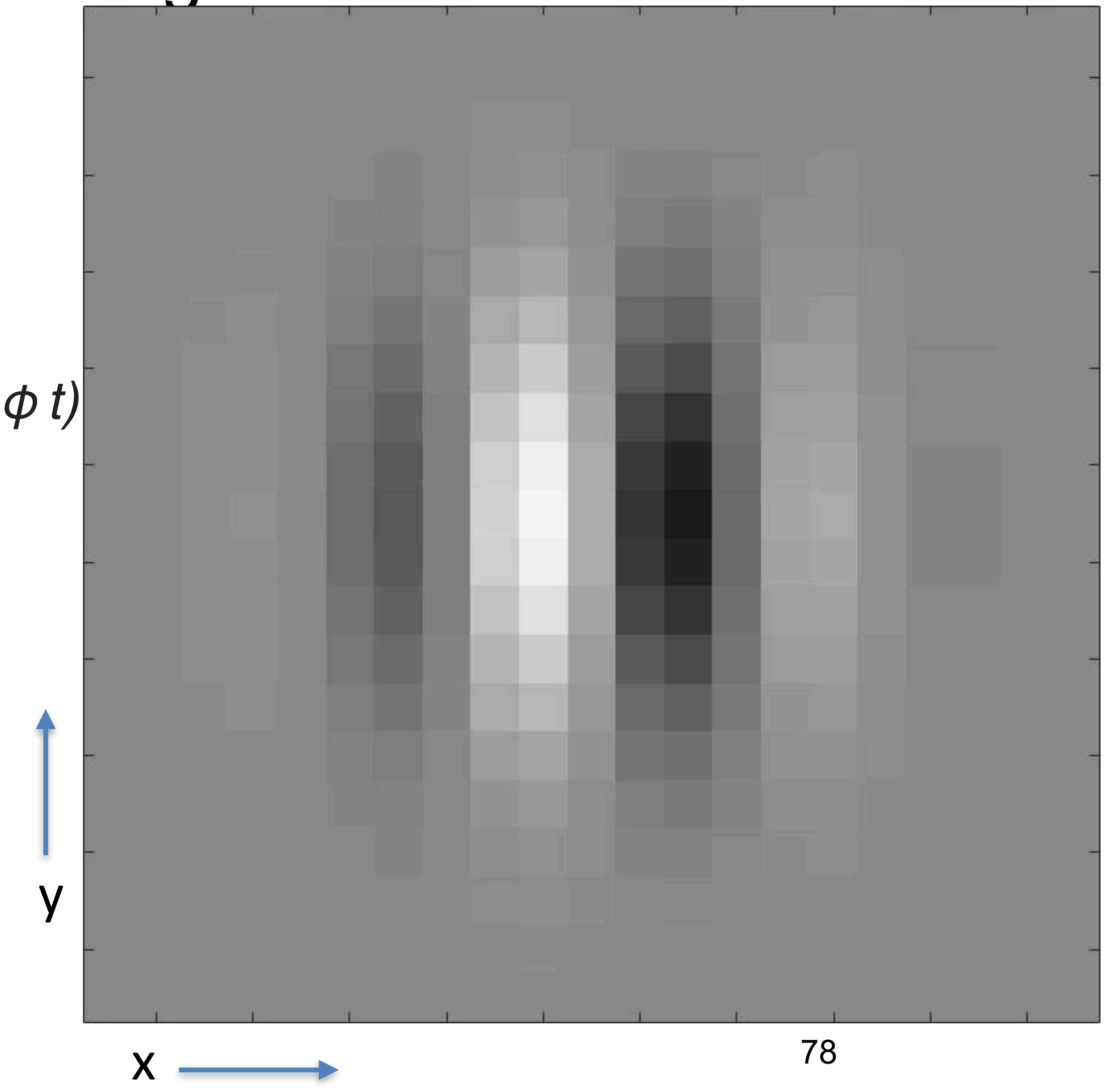


$$\psi_s(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \sin(2\pi u_0 x)$$



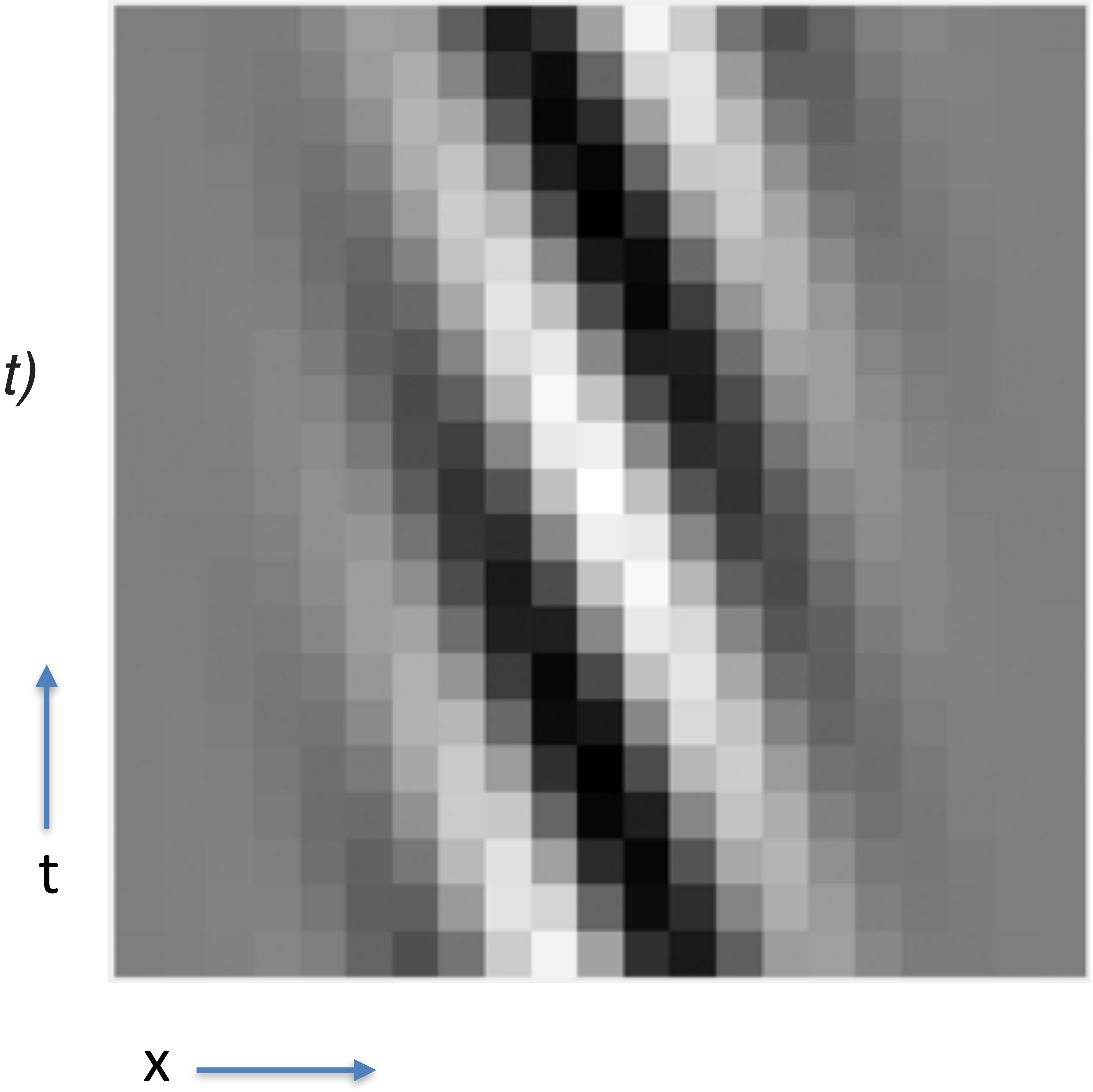
Using phase changes of local Gabor filters to analyze or generate motion

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



Space-time plot of the a slice through the patio-temporal filter of the previous slide

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



Spatio-temporal sampling illusion, due to Edward Adelson and Jim Bergen

Evidence for filter-based analysis of motion in the human visual system shown via spatio-temporal visual illusion based on sampling

Two potential theories for how humans compute our motion perceptions:

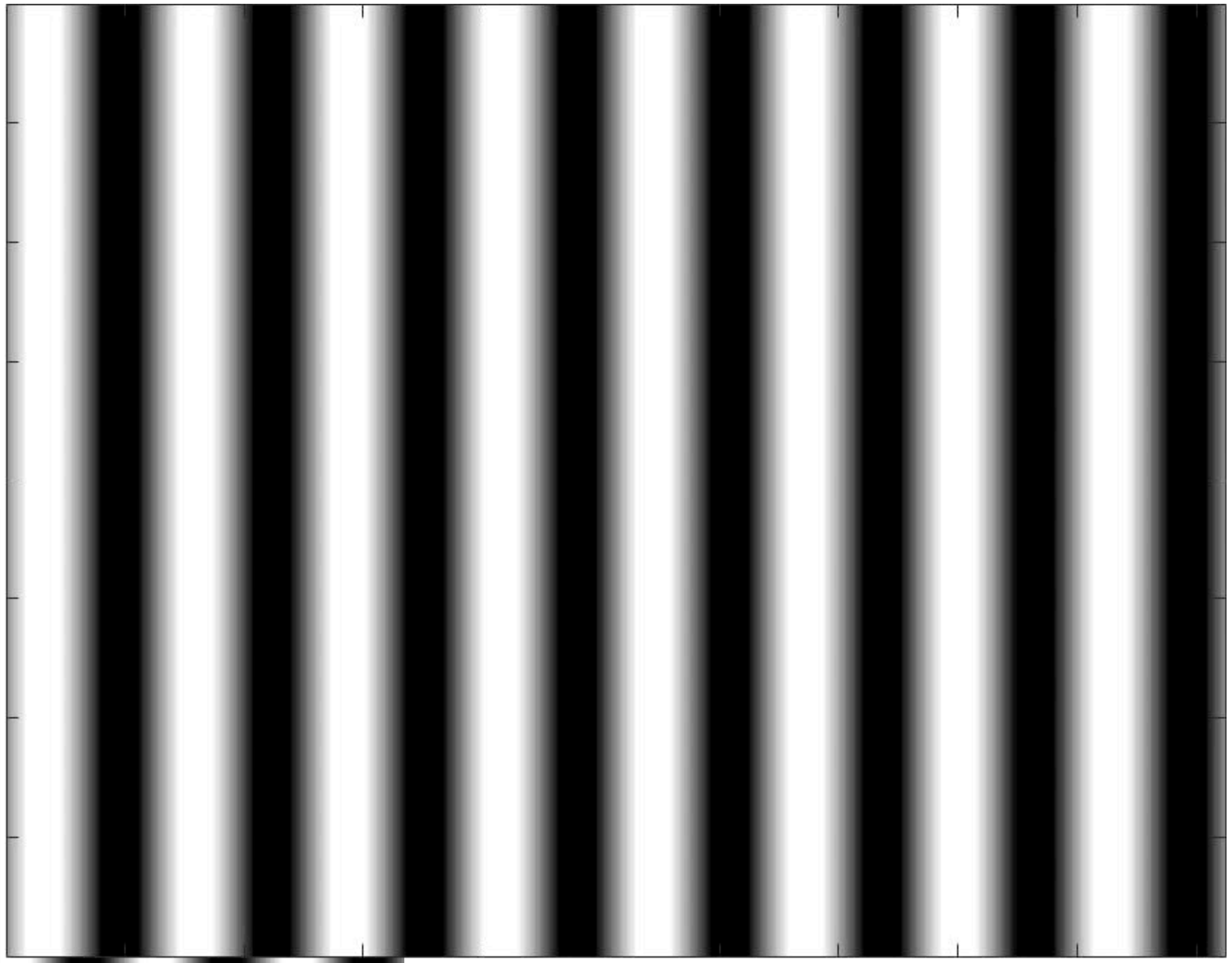
(a) We match the pattern in the image that we see at one moment and compare it with what we see at subsequent times.
(b) We use spatio-temporal filters to measure spatio-temporal energy in order to measure local motion.

This illusion favors one theory over the other.

spatial frequency

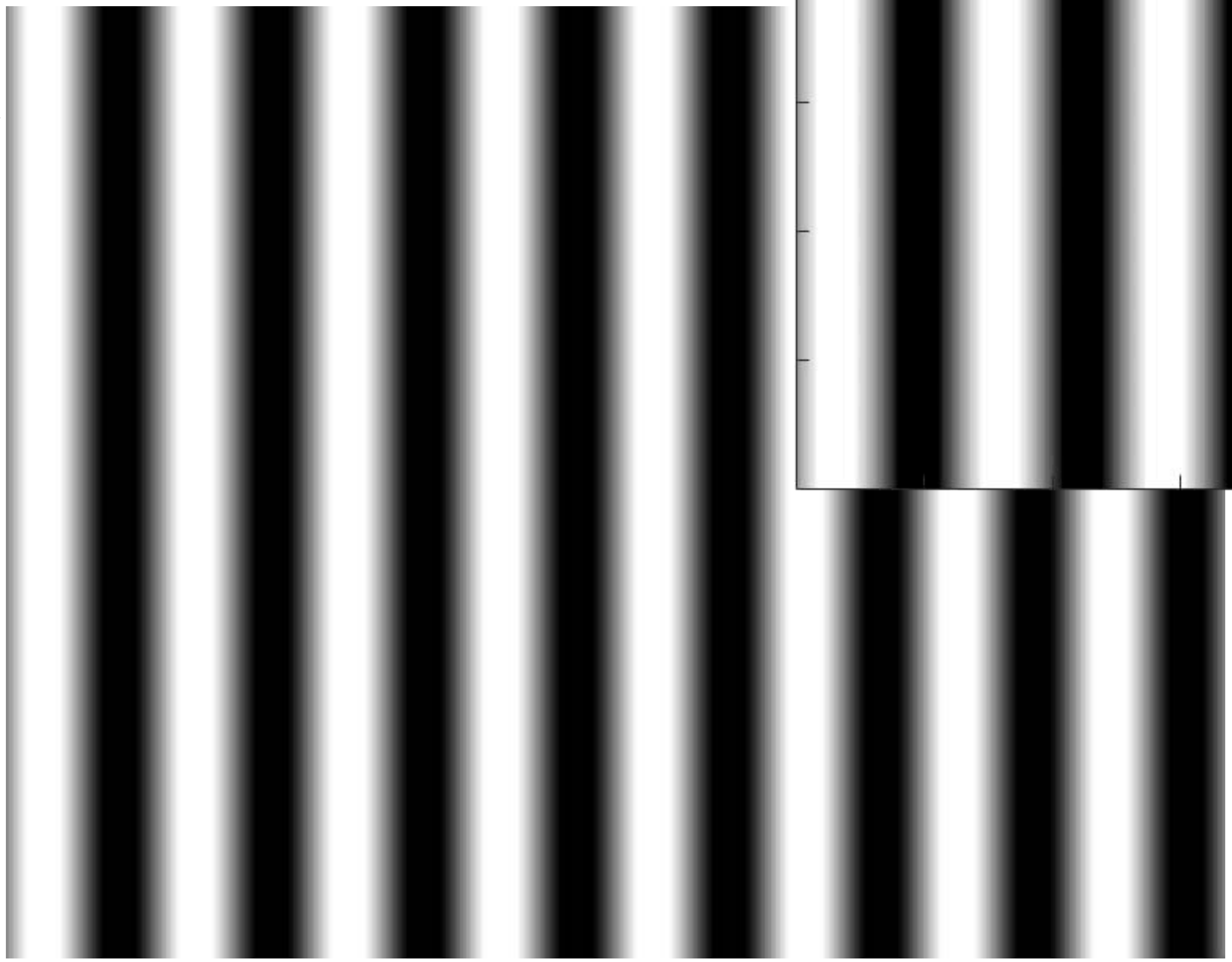
temporal frequency

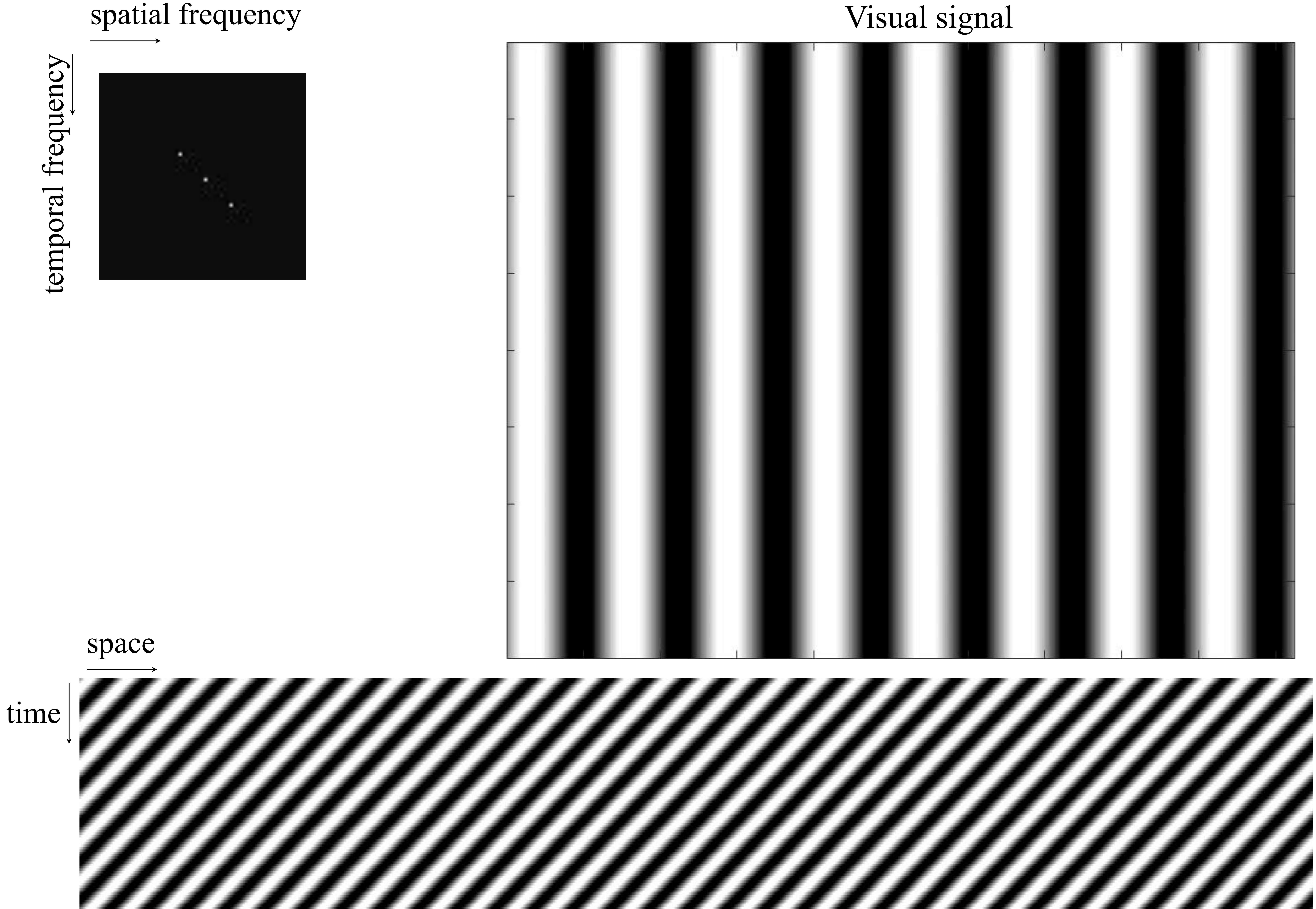
Visual signal (this “video” is static)



space

time





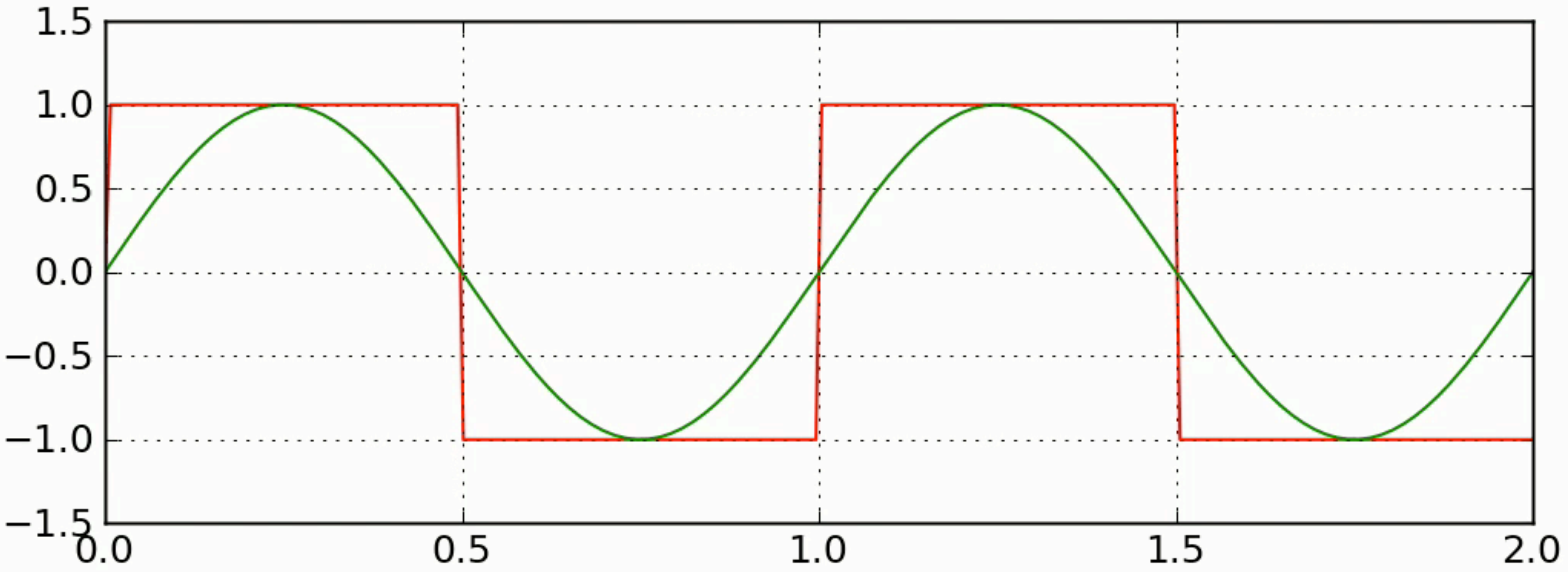
spatial frequency
temporal frequency
Visual signal
space
time

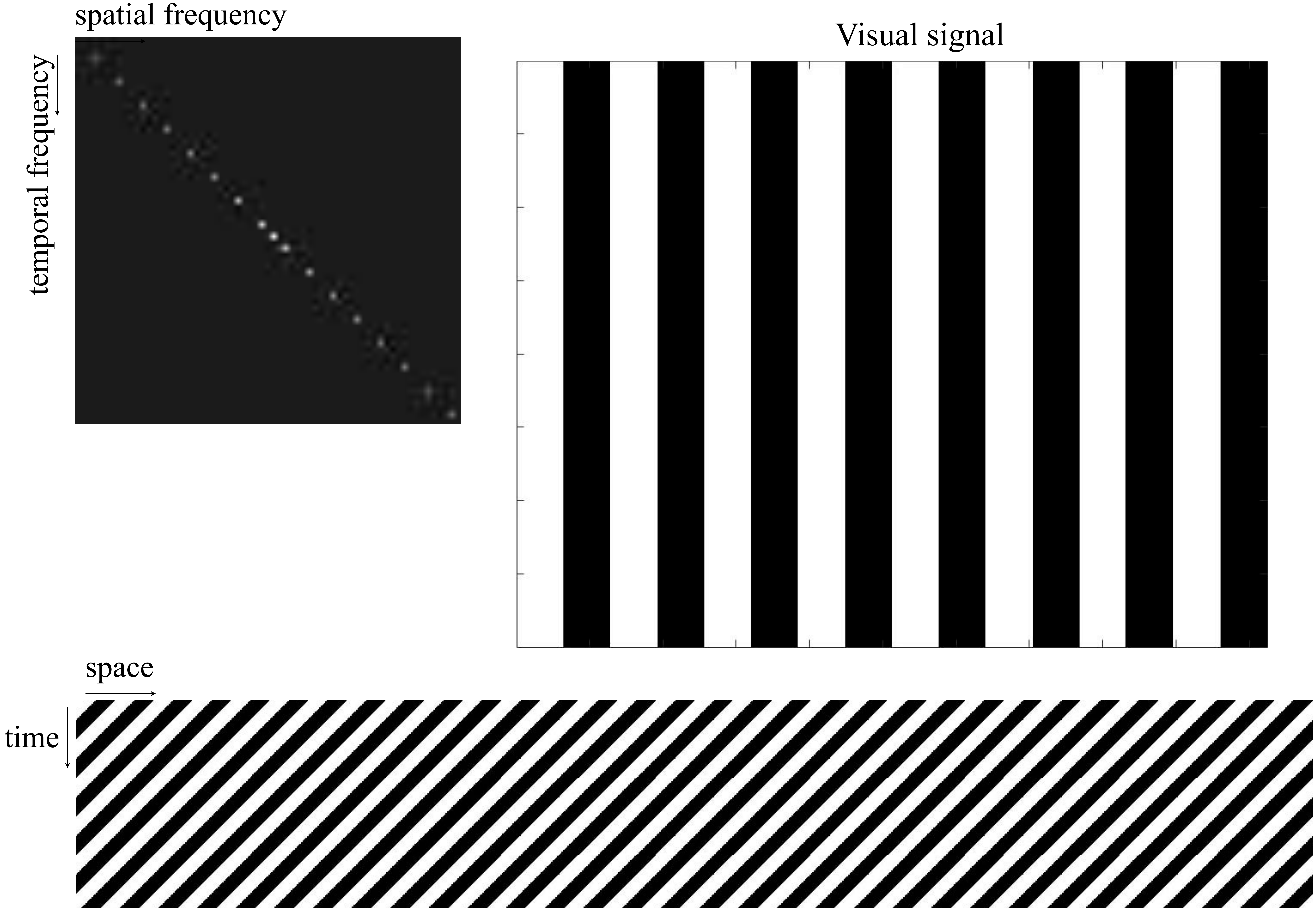
Square wave Fourier components

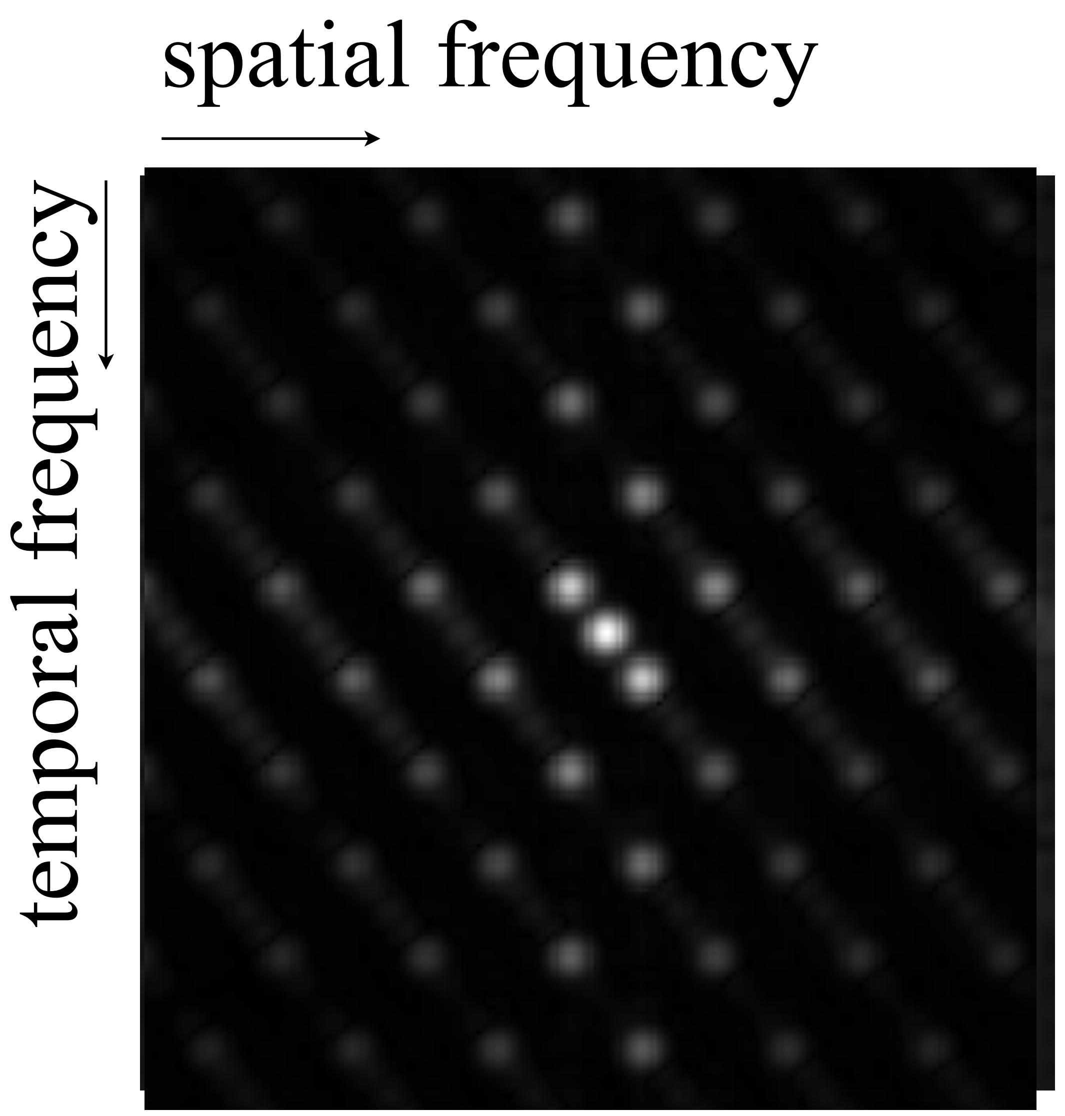
Using Fourier series we can write an ideal square wave as an infinite series of the form

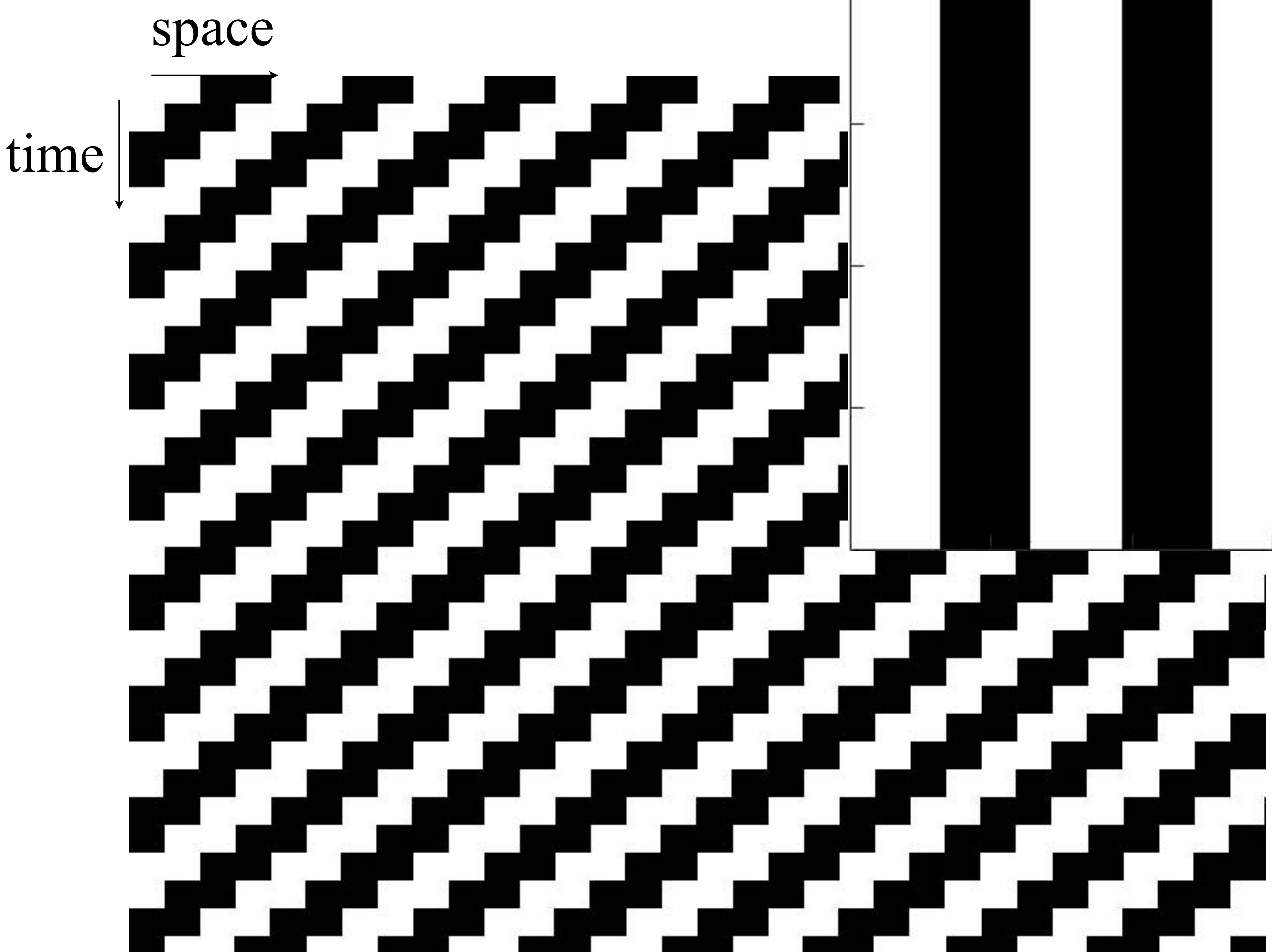
$$\begin{aligned} x_{\text{square}}(t) &= \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin\left((2k-1)2\pi f t\right)}{(2k-1)} \\ &= \frac{4}{\pi}\left(\sin(2\pi f t) + \frac{1}{3}\sin(6\pi f t) + \frac{1}{5}\sin(10\pi f t) + \cdots\right). \end{aligned}$$

A square wave is an infinite sum of sinusoids

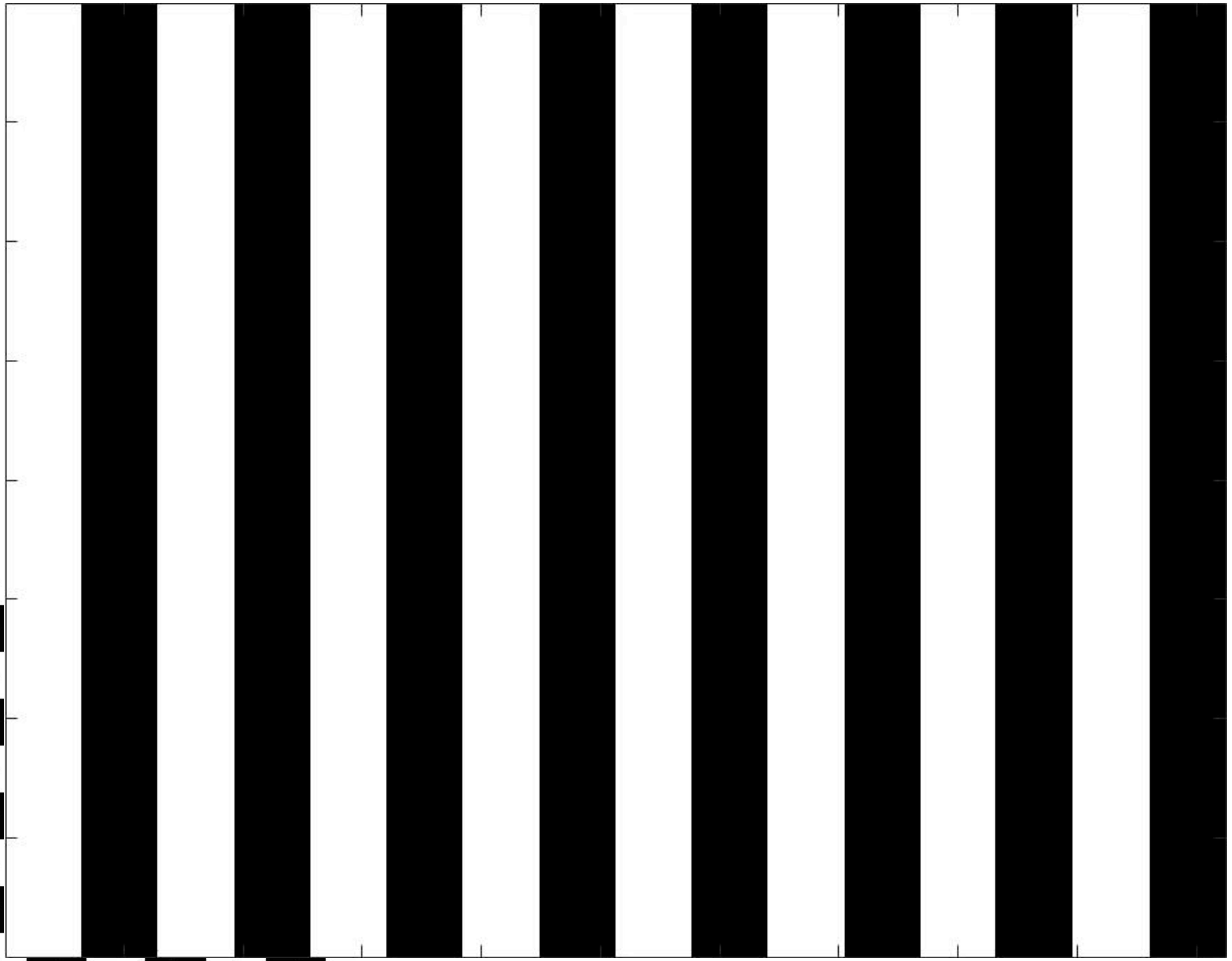


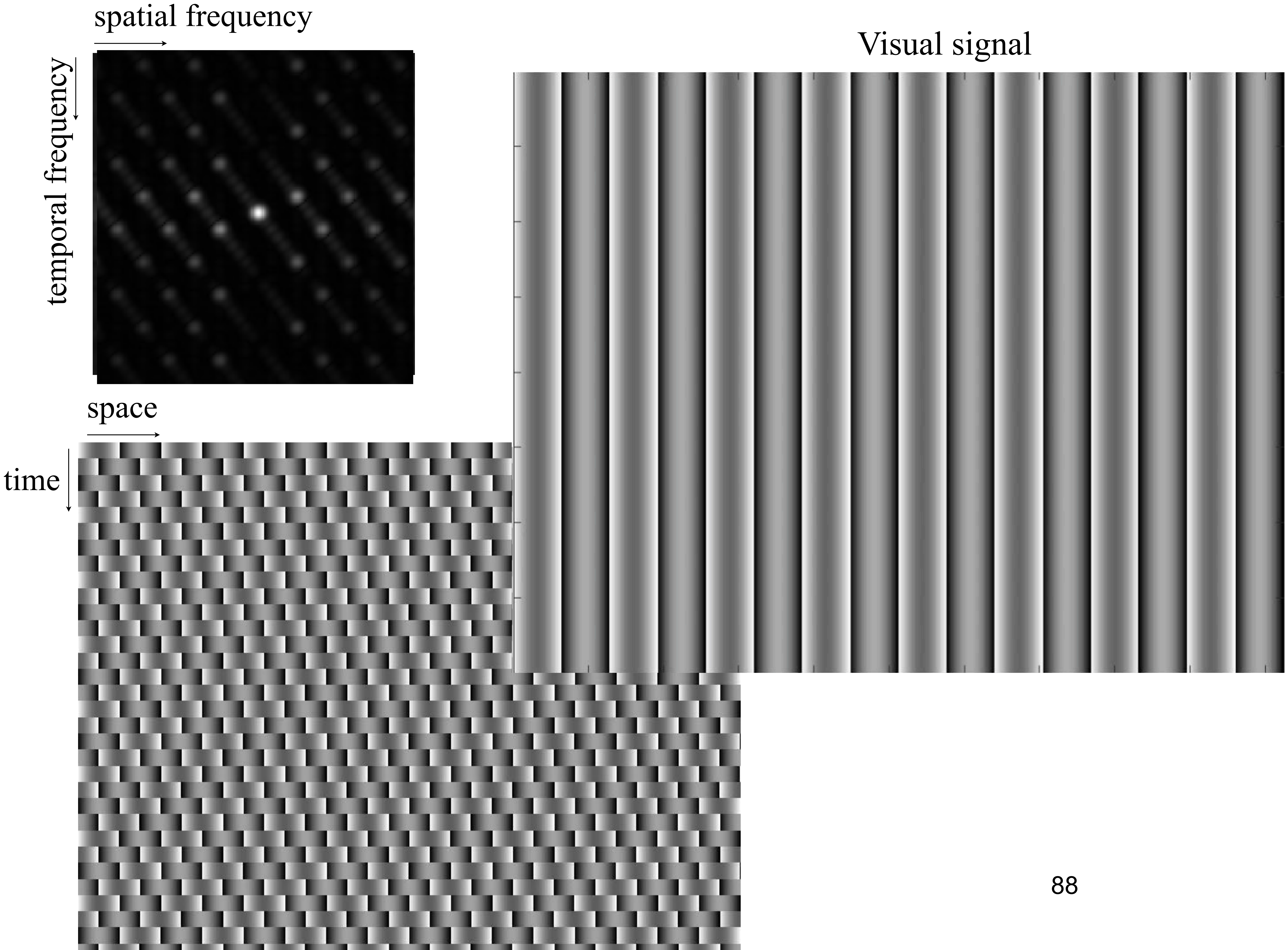

spatial frequency
temporal frequency
Visual signal
space
time



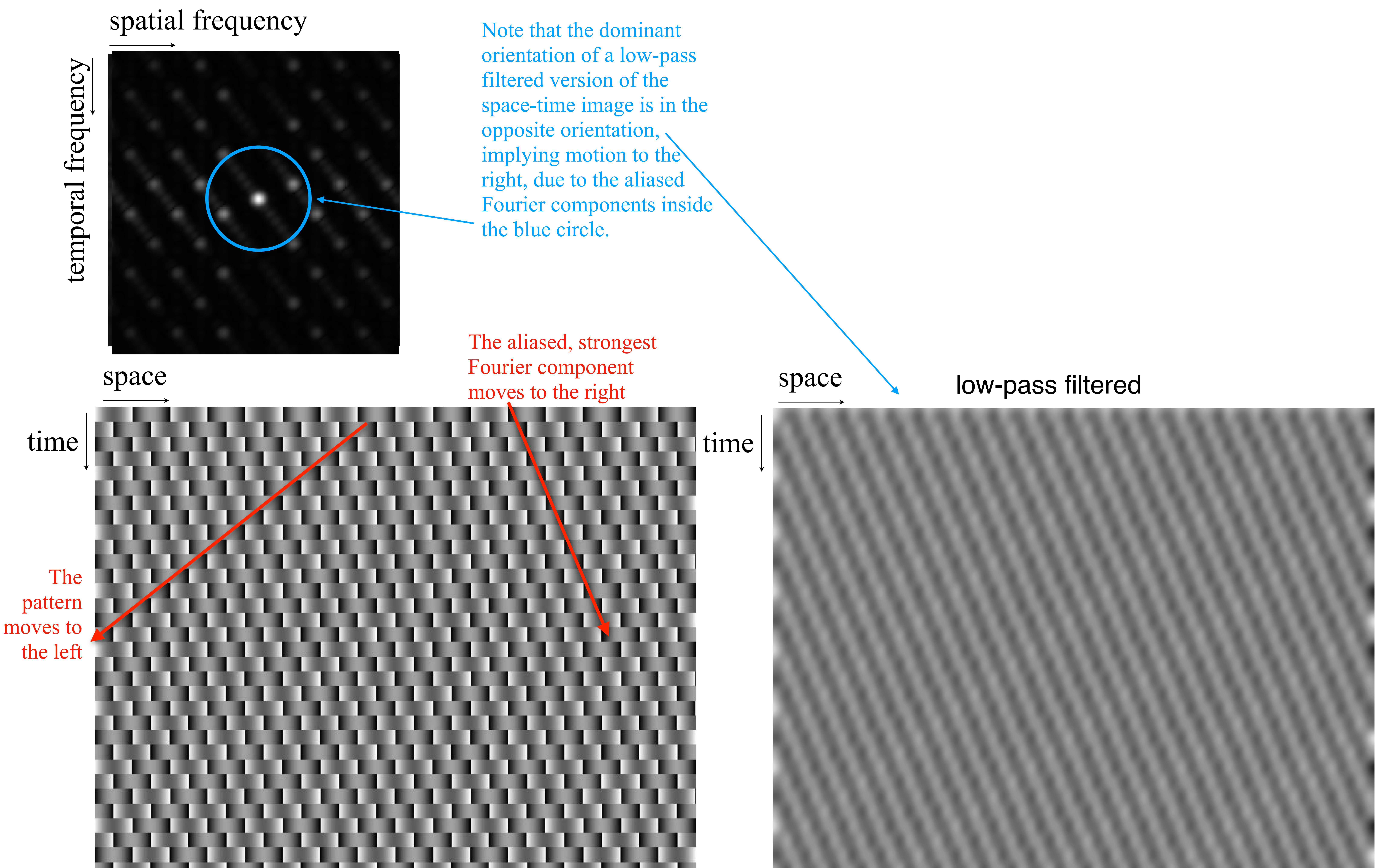


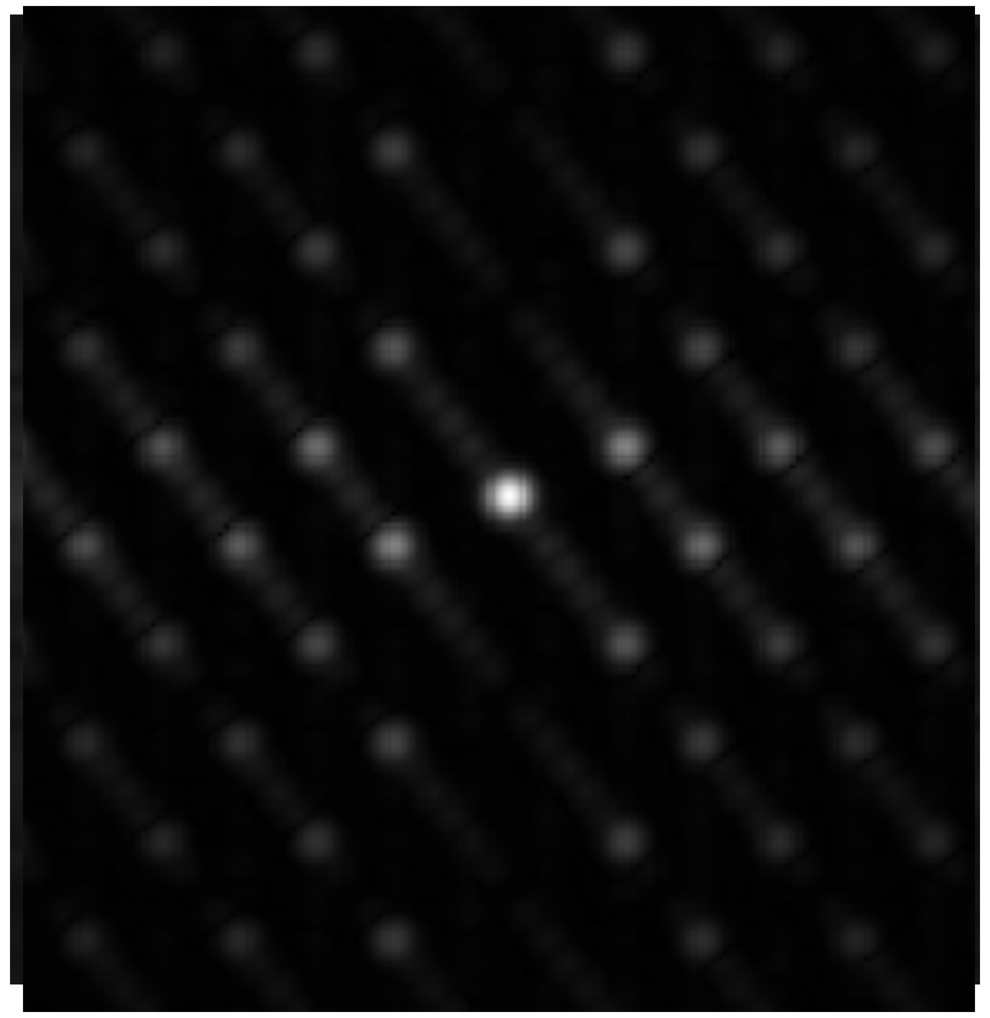
Visual signal

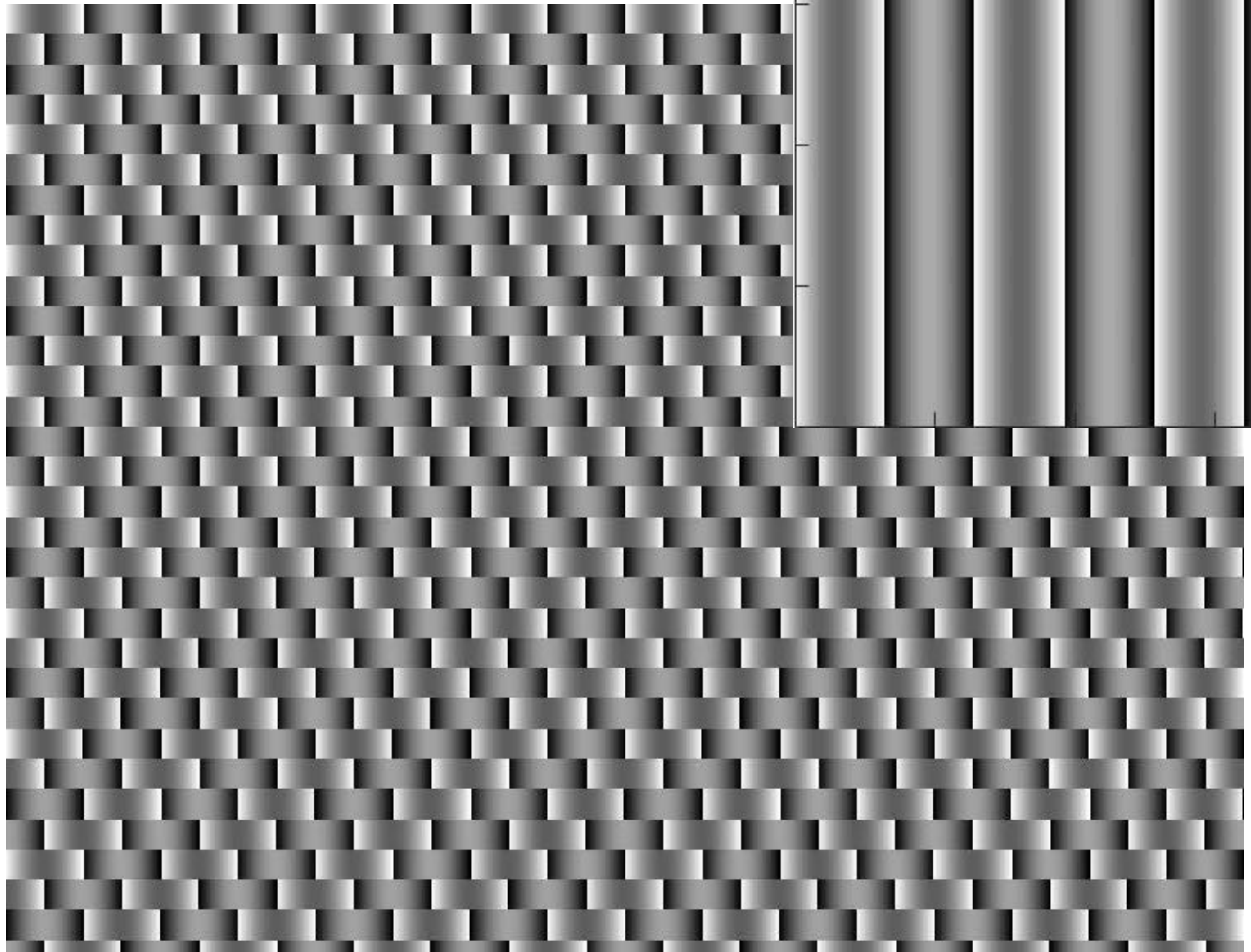




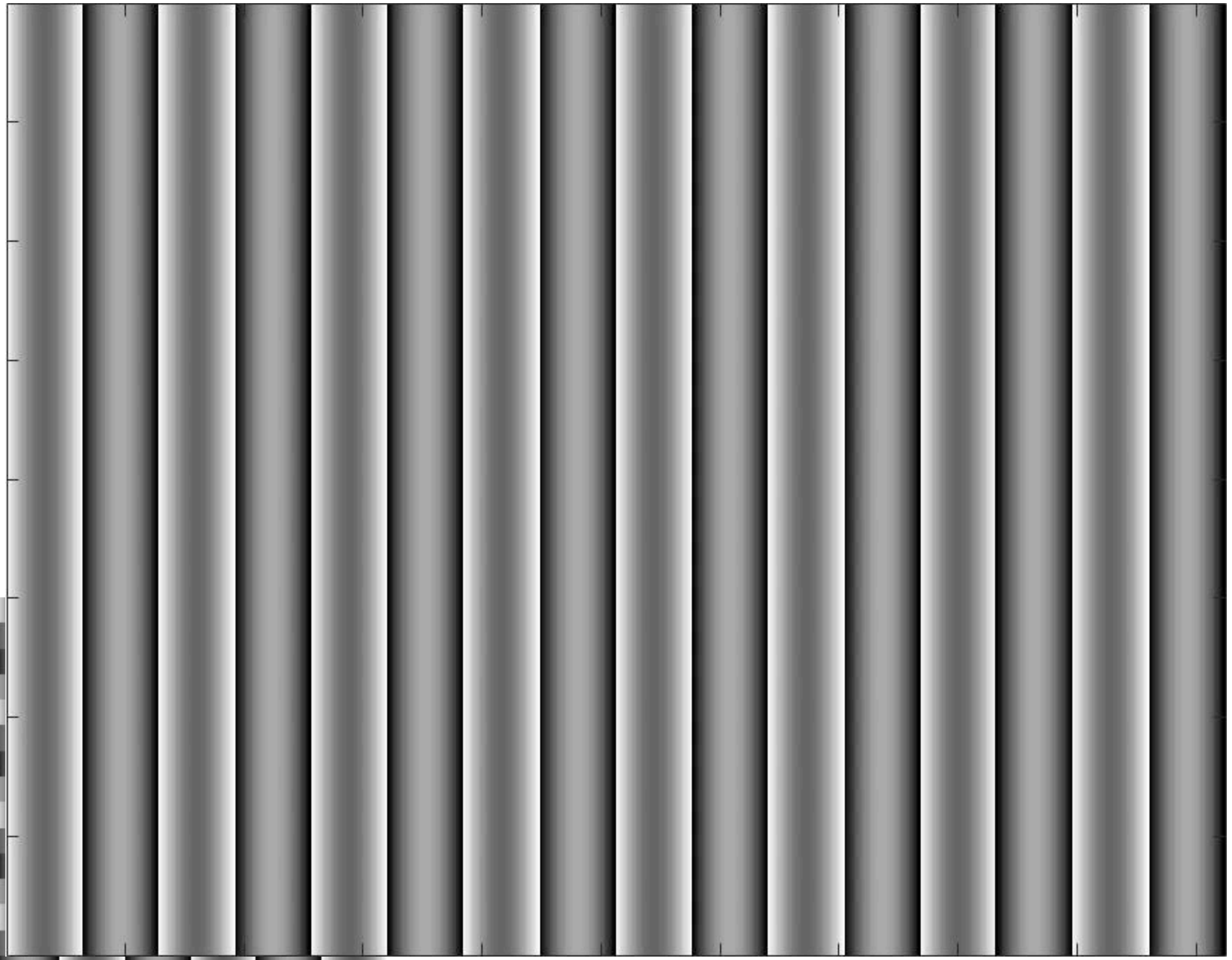
spatial frequency
temporal frequency
Visual signal
space
time

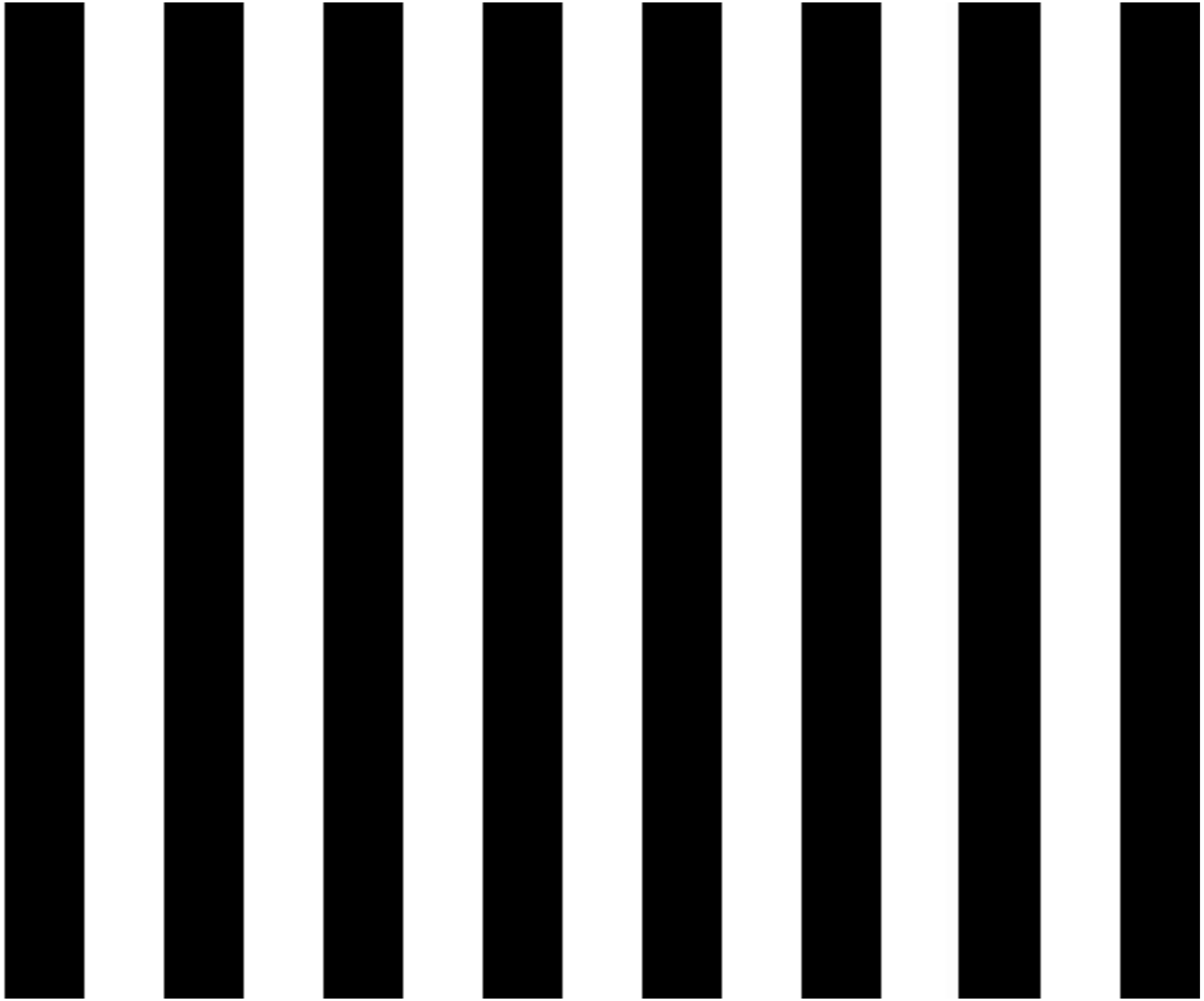

spatial frequency
temporal frequency
Note that the dominant orientation of a low-pass filtered version of the space-time image is in the opposite orientation, implying motion to the right, due to the aliased Fourier components inside the blue circle.
The aliased, strongest Fourier component moves to the right
space
time
The pattern moves to the left
space
low-pass filtered
time

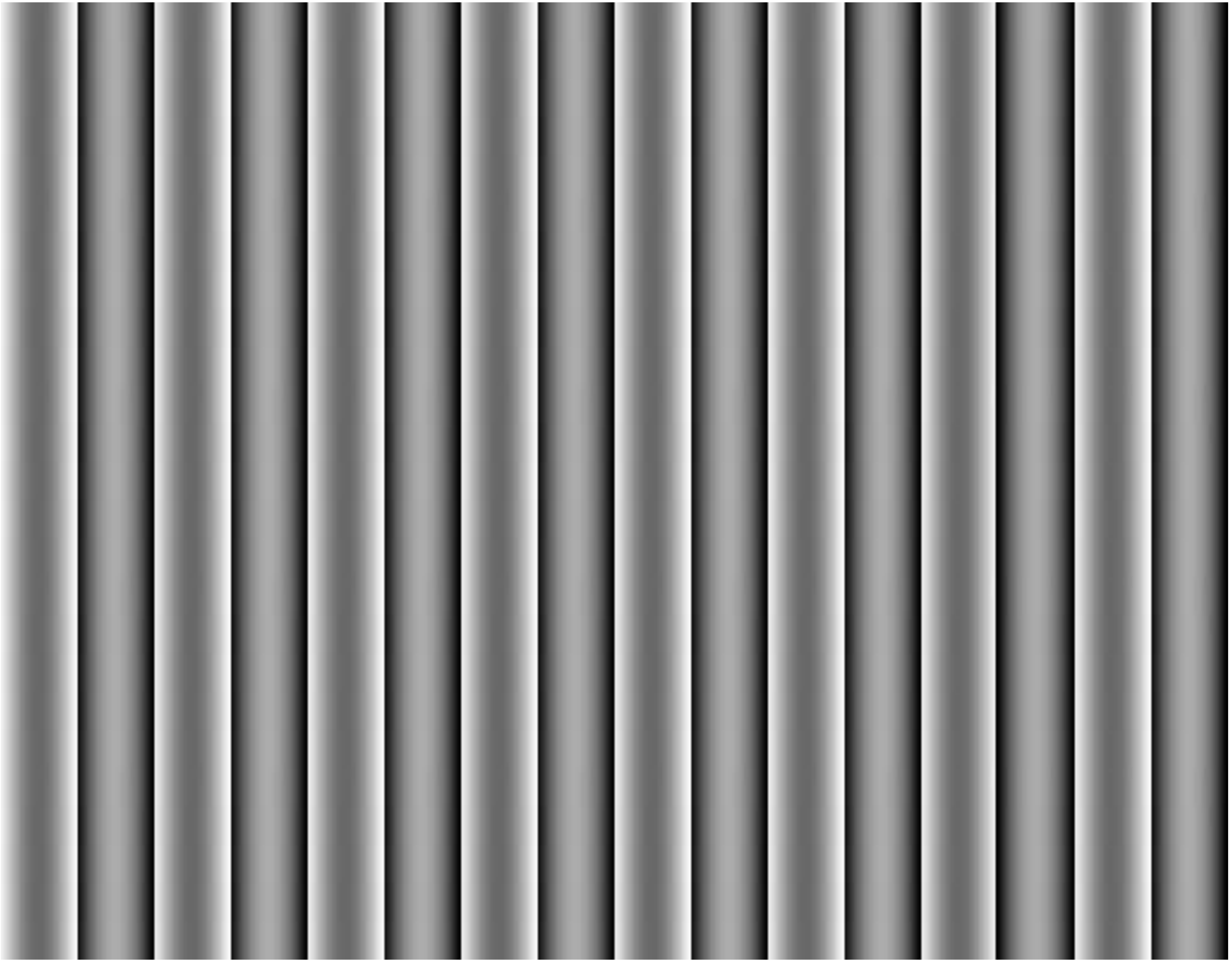




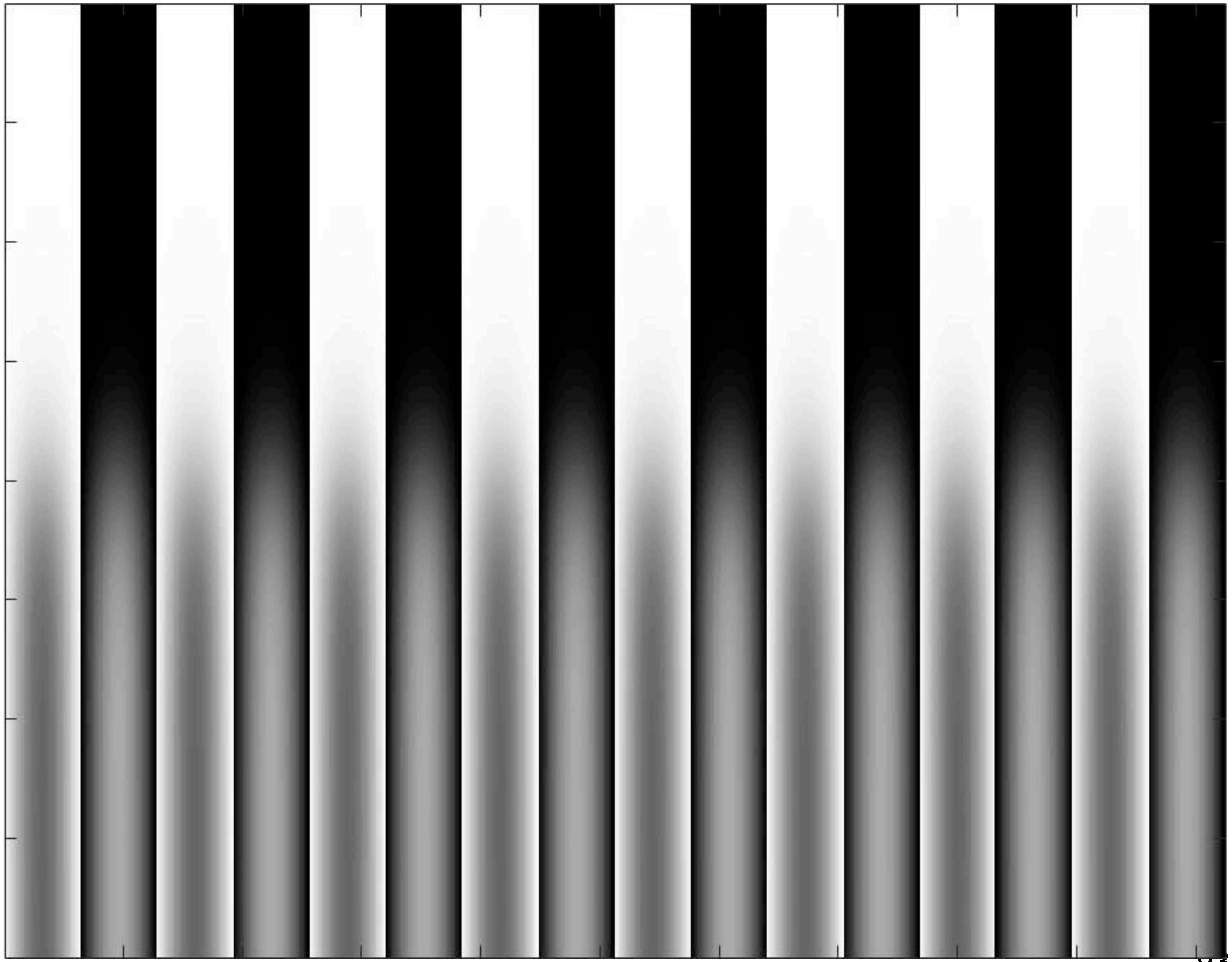
Visual signal



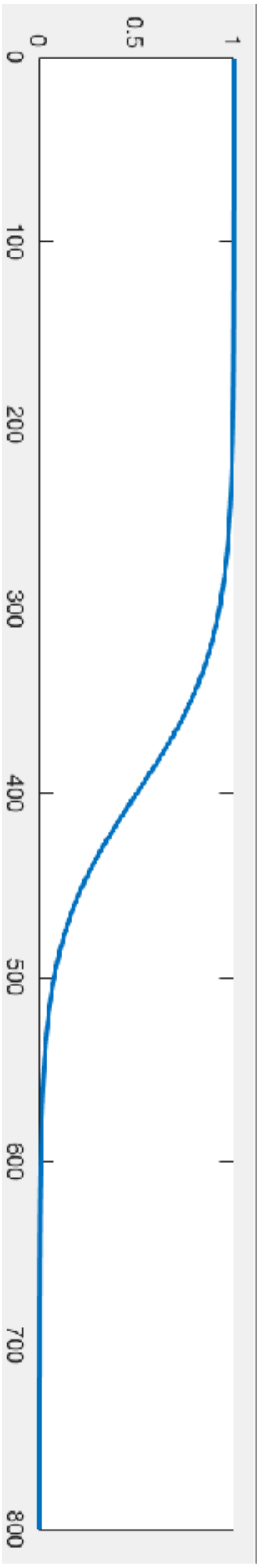




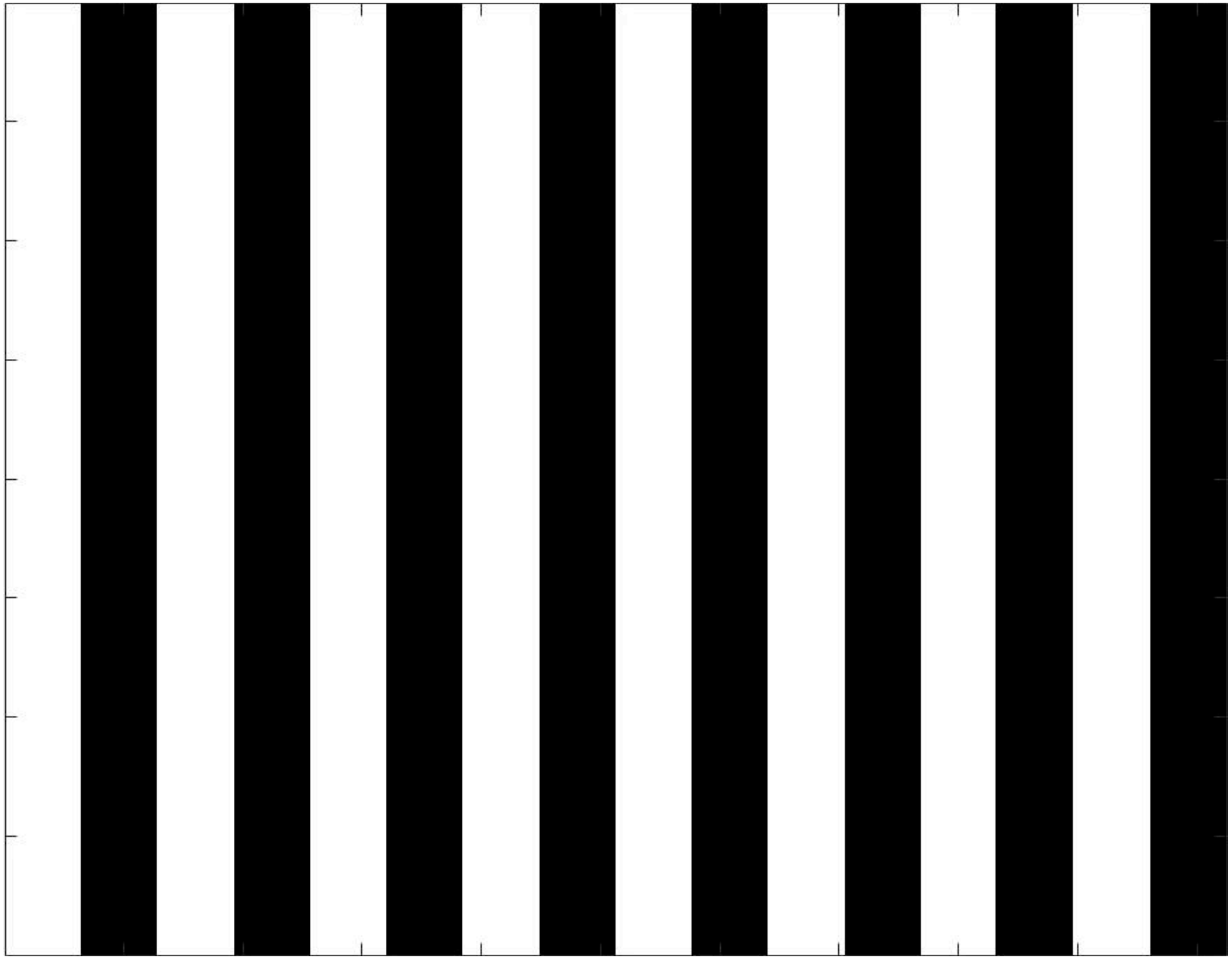
blend over the two conditions



fraction of square wave
fundamental frequency

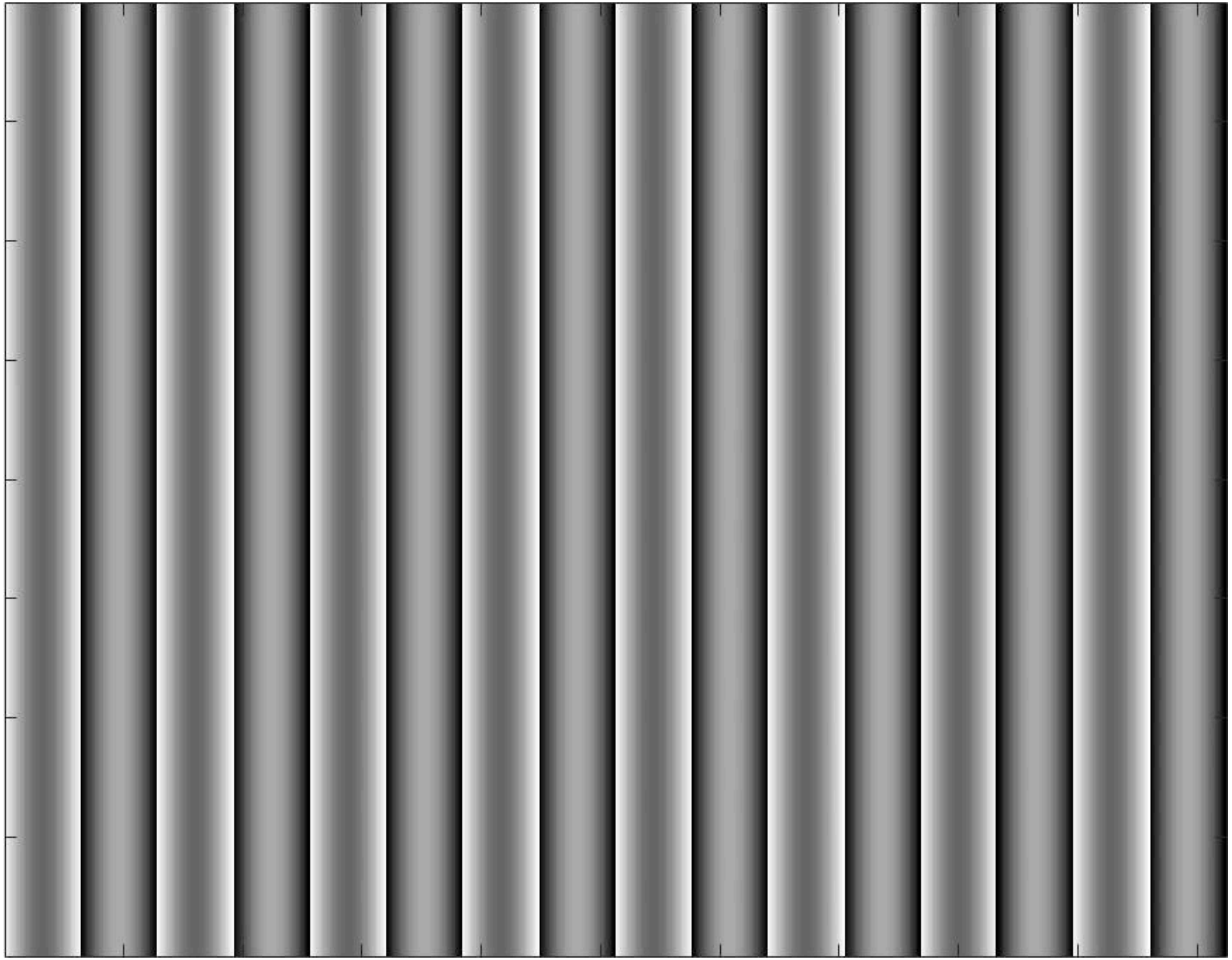


faster display speed



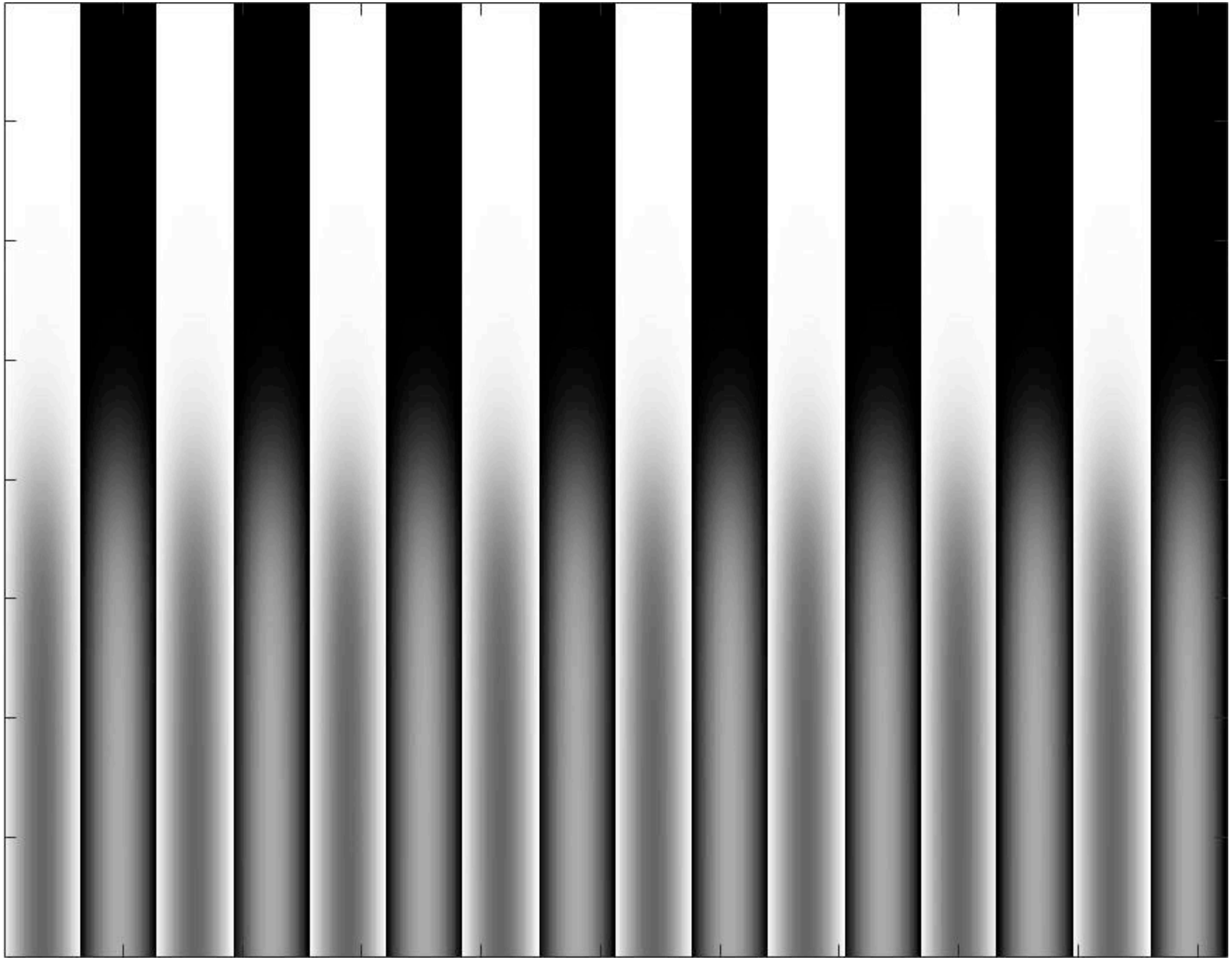
alpha: 1 squareFlag: 1 offset: period/4

faster display speed



alpha: 0 squareFlag: 1 offset: period/4

fast blended…



lecture summary

- We have "inverted U shaped" sensitivity to spatial frequencies, peaking at 6 cycles per degree.
- We discussed ways to filter out different spatial frequency components of an image.
- Aliasing: "blur before you subsample".
- Spatio-temporal filtering enables motion analysis.
- Motion illusion gives evidence some temporal filtering mechanisms are involved in our motion processing.