Lecture 6
Motion Filtering and Sampling

6.869/6.819 Advances in Computer Vision

Fall 2019

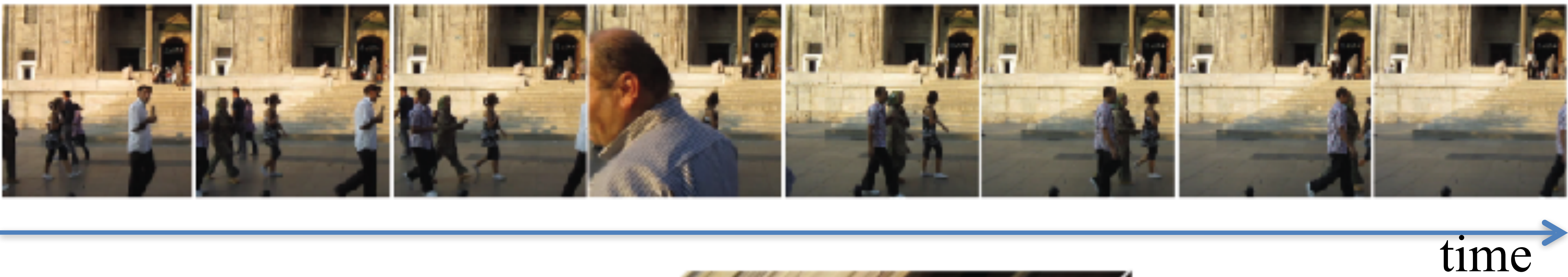
Bill Freeman, Antonio Torralba, Phillip Isola

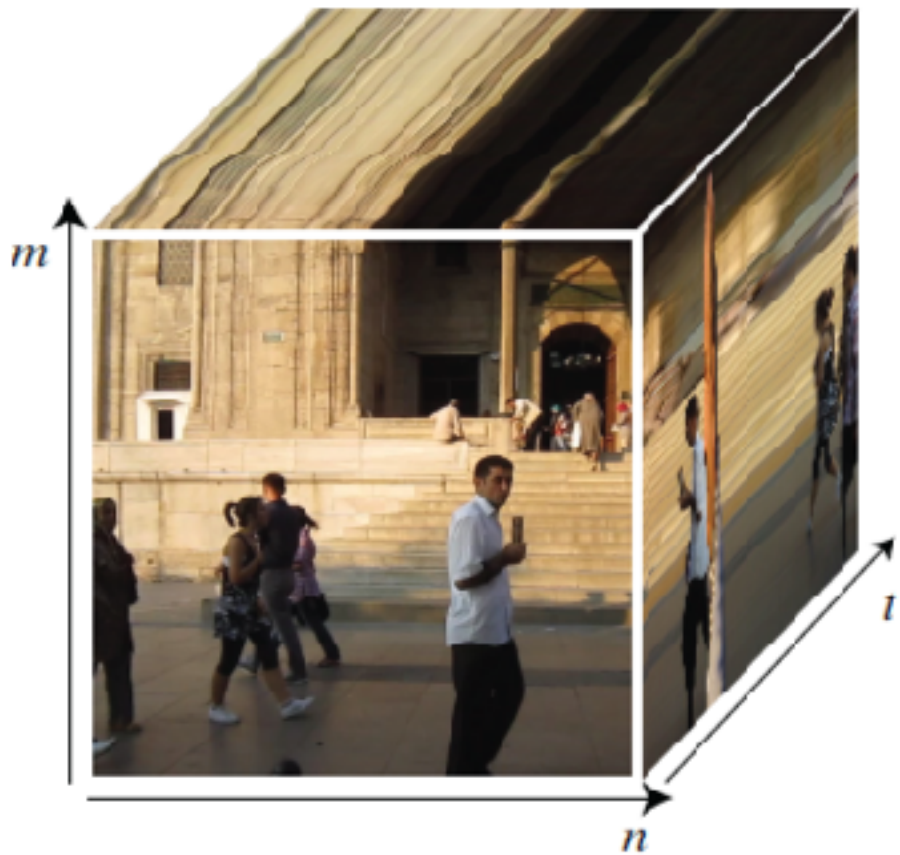
Sept. 24, 2019

Temporal filtering

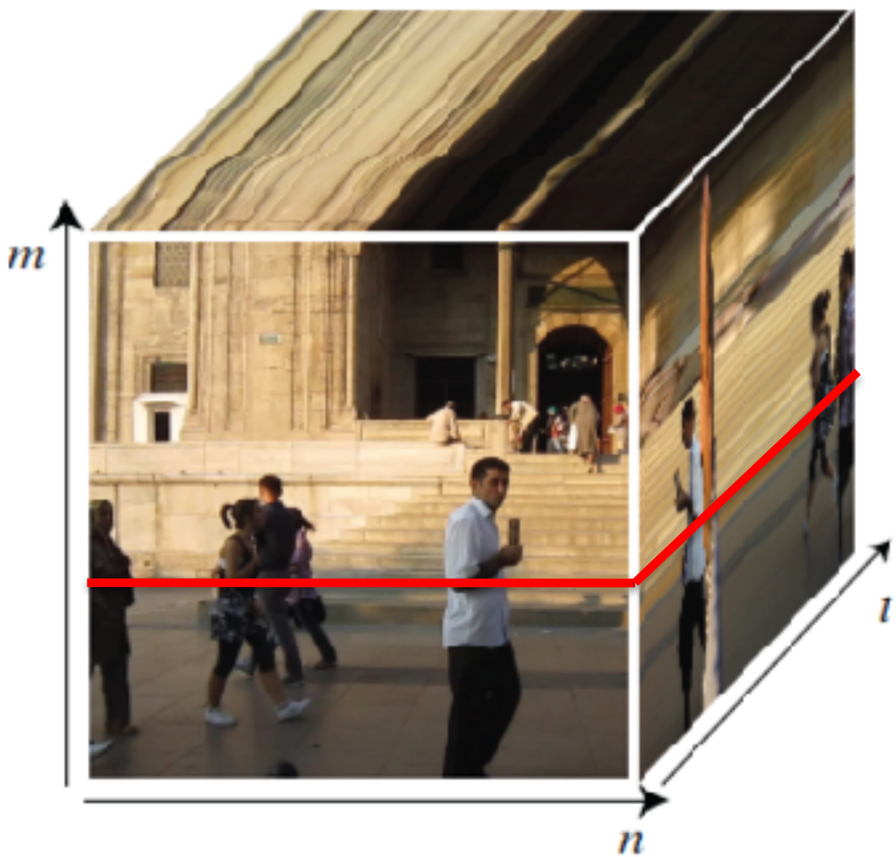
why filter videos?

Sequences

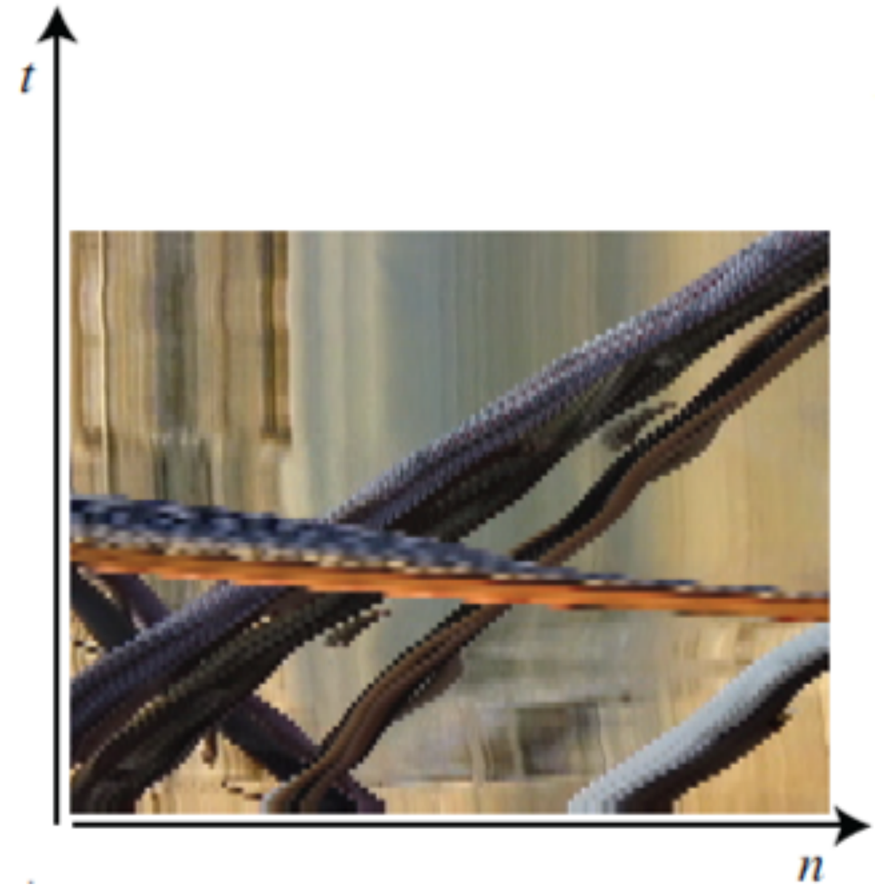




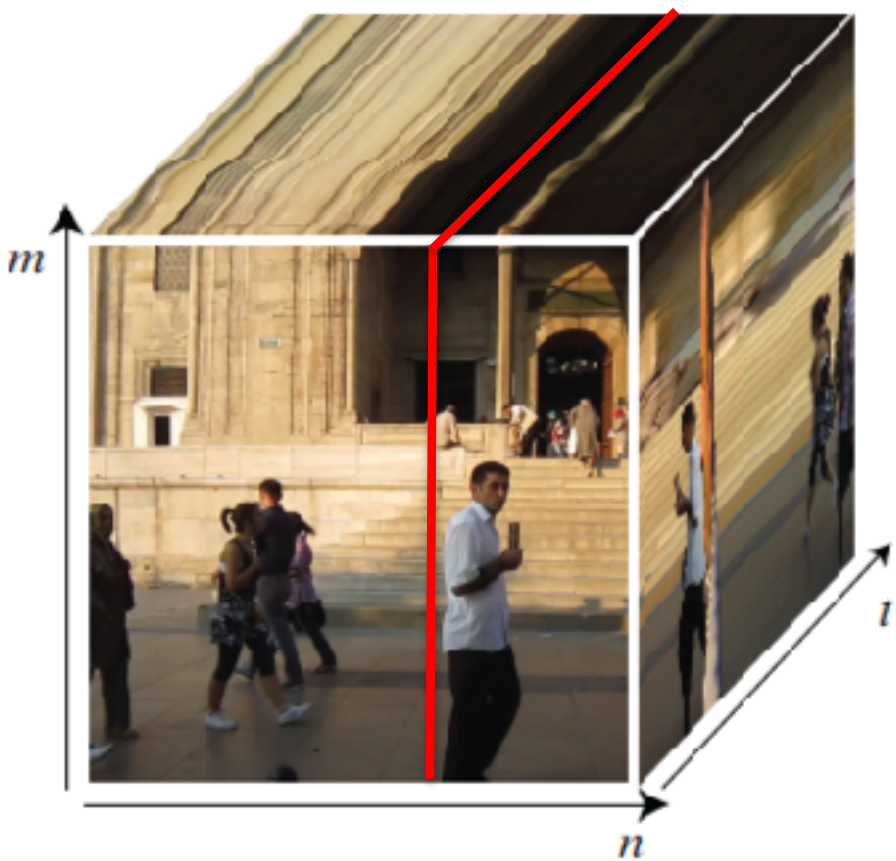
Sequences



Cube size = 128x128x90



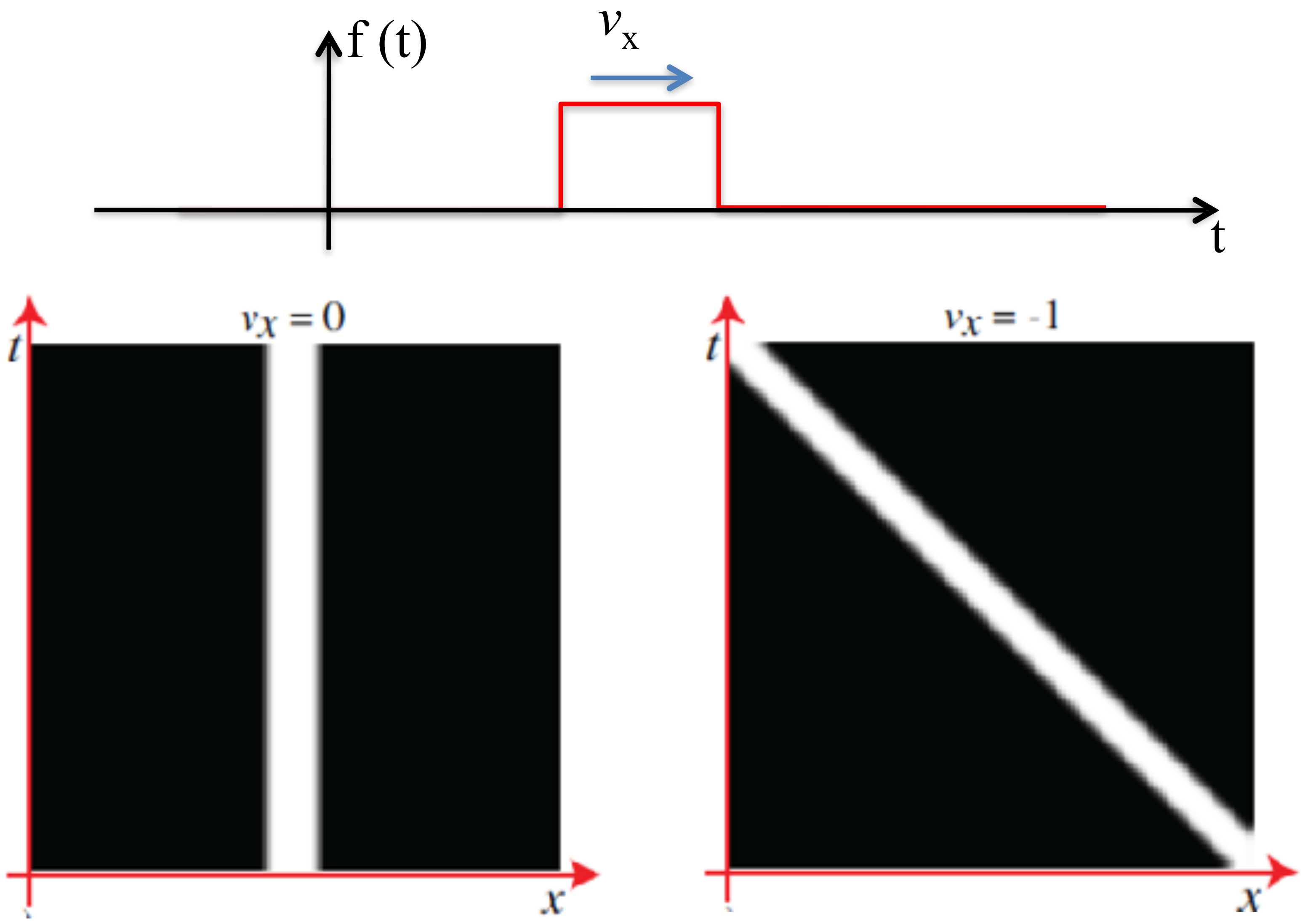
Sequences



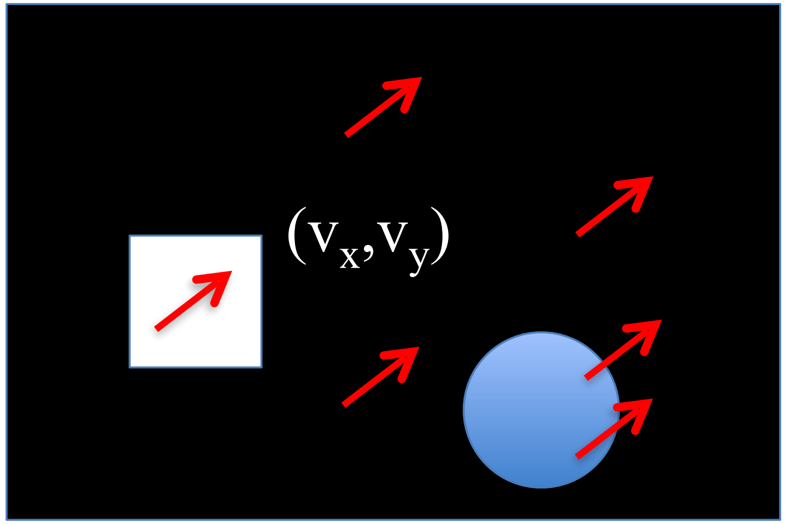
Cube size = 128x128x90

Globally constant motion

Let's work on the continuous space-time domain…



Global constant motion



A global motion can be written as:

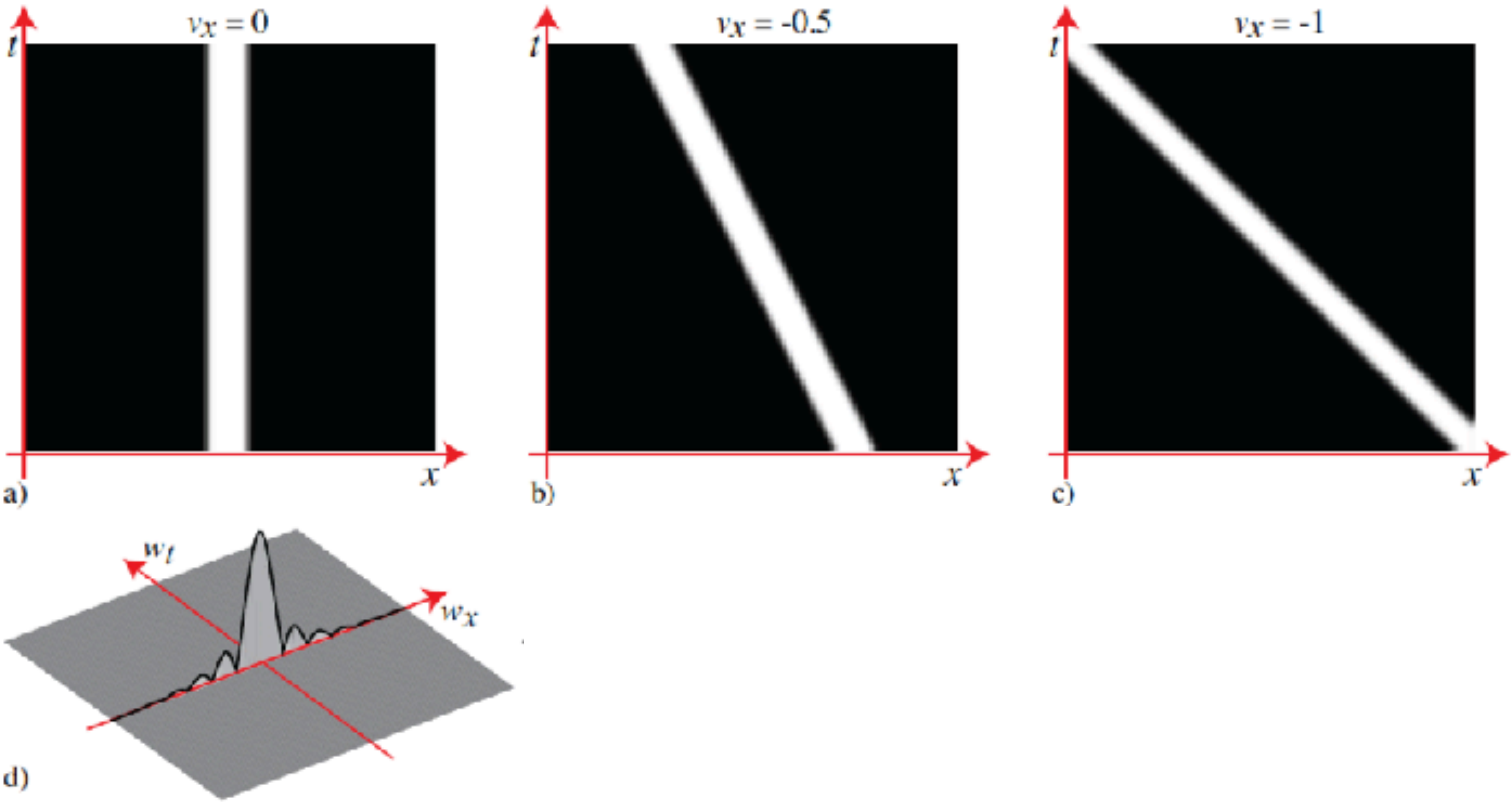
$$f(x, y, t) = f_0(x - v_x t, y - v_y t)$$

Where:

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

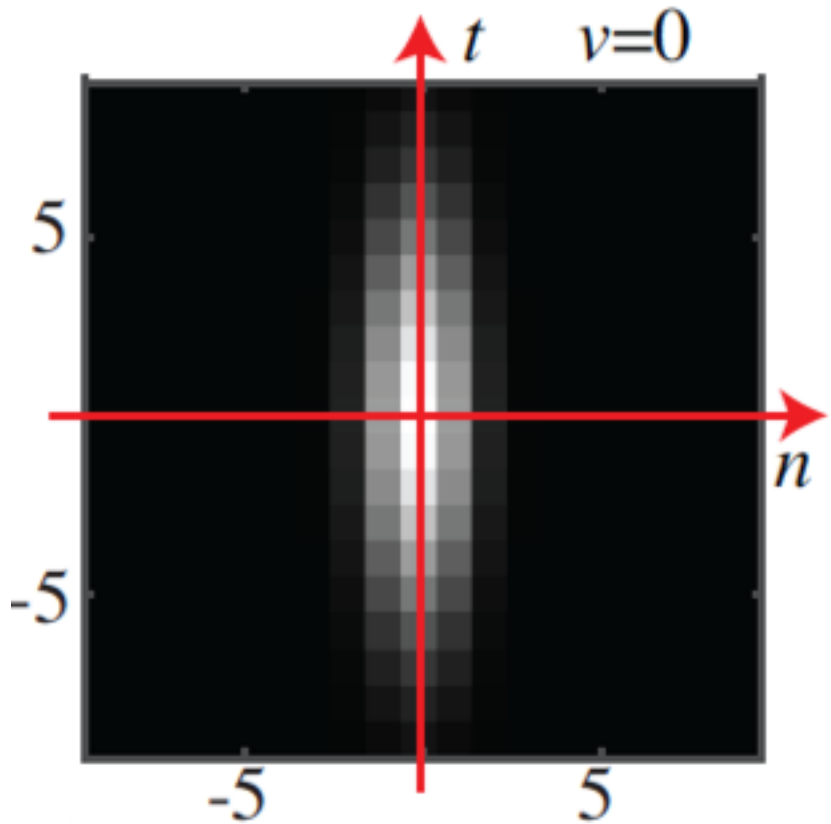
$$f(x,y,t) = f_0(x - v_x t, y - v_y t)$$

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

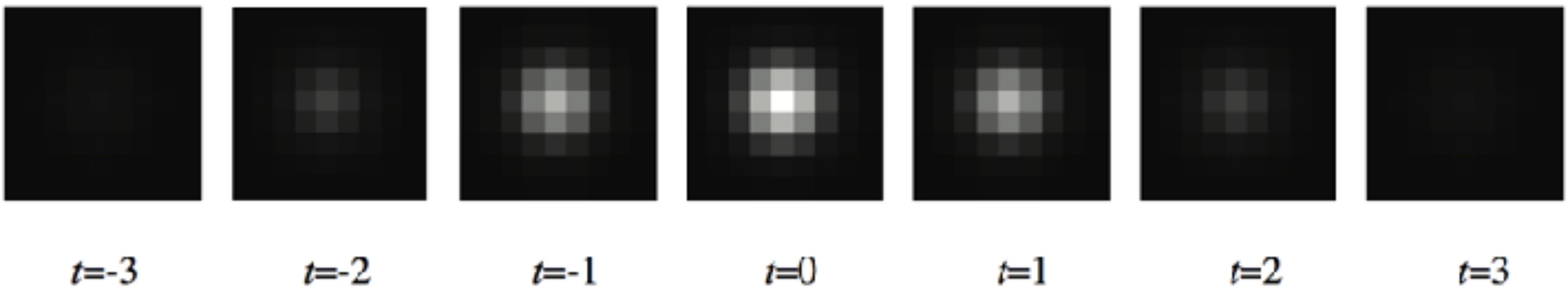


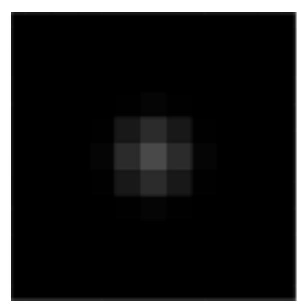
Temporal Gaussian

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



Spatio-temporal Gaussian

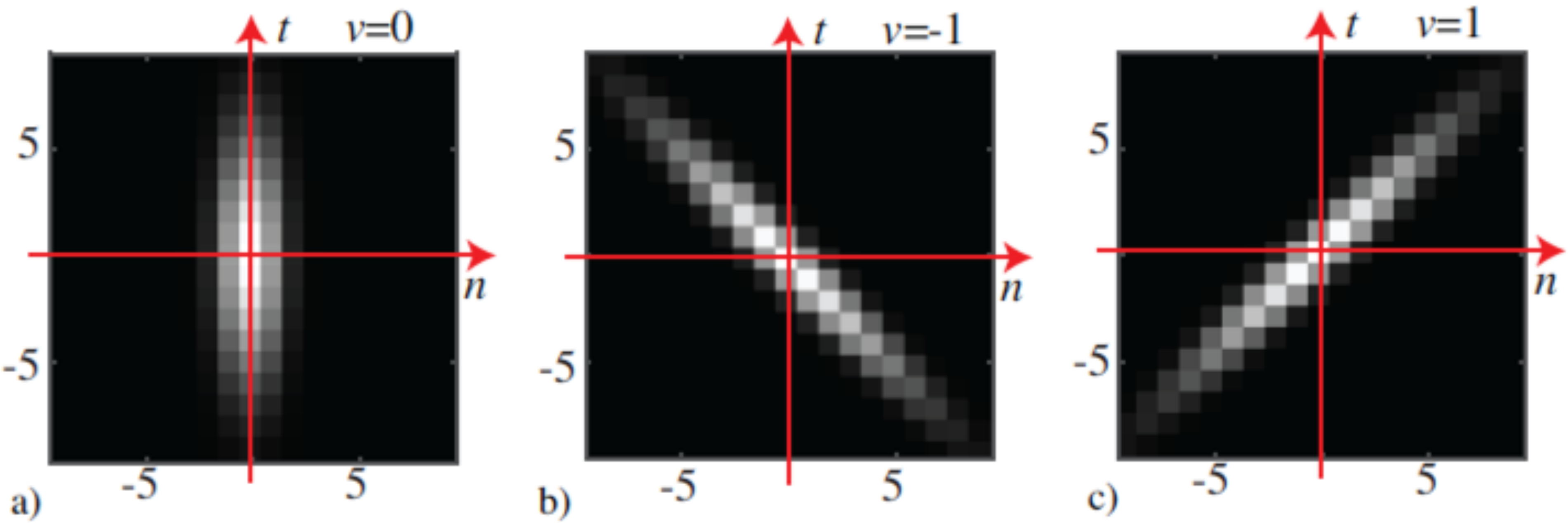




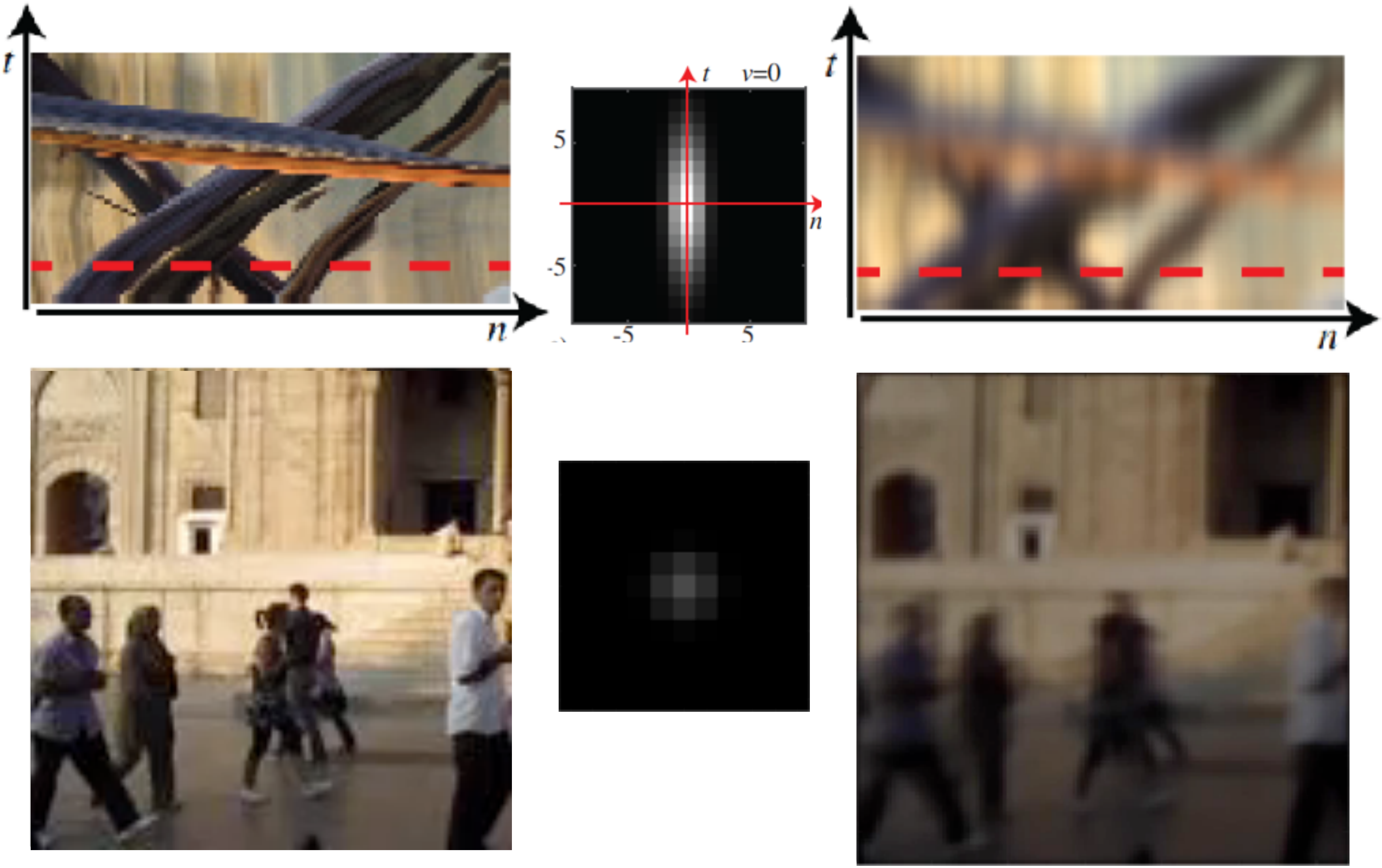
Spatio-temporal Gaussian

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

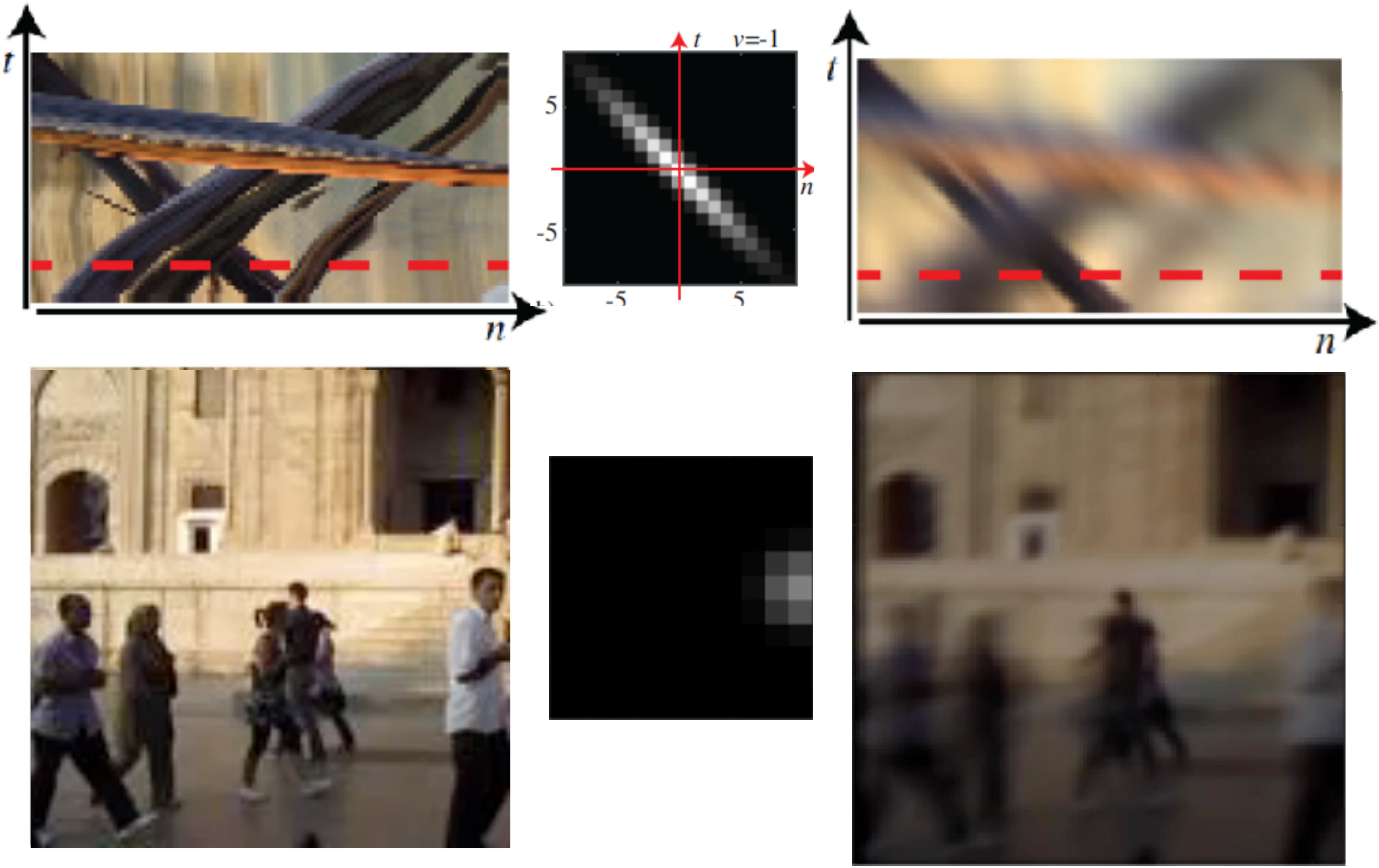
$$g_{v_x,v_y}(x,y,t) = g(x - v_x t, y - v_y t, t)$$

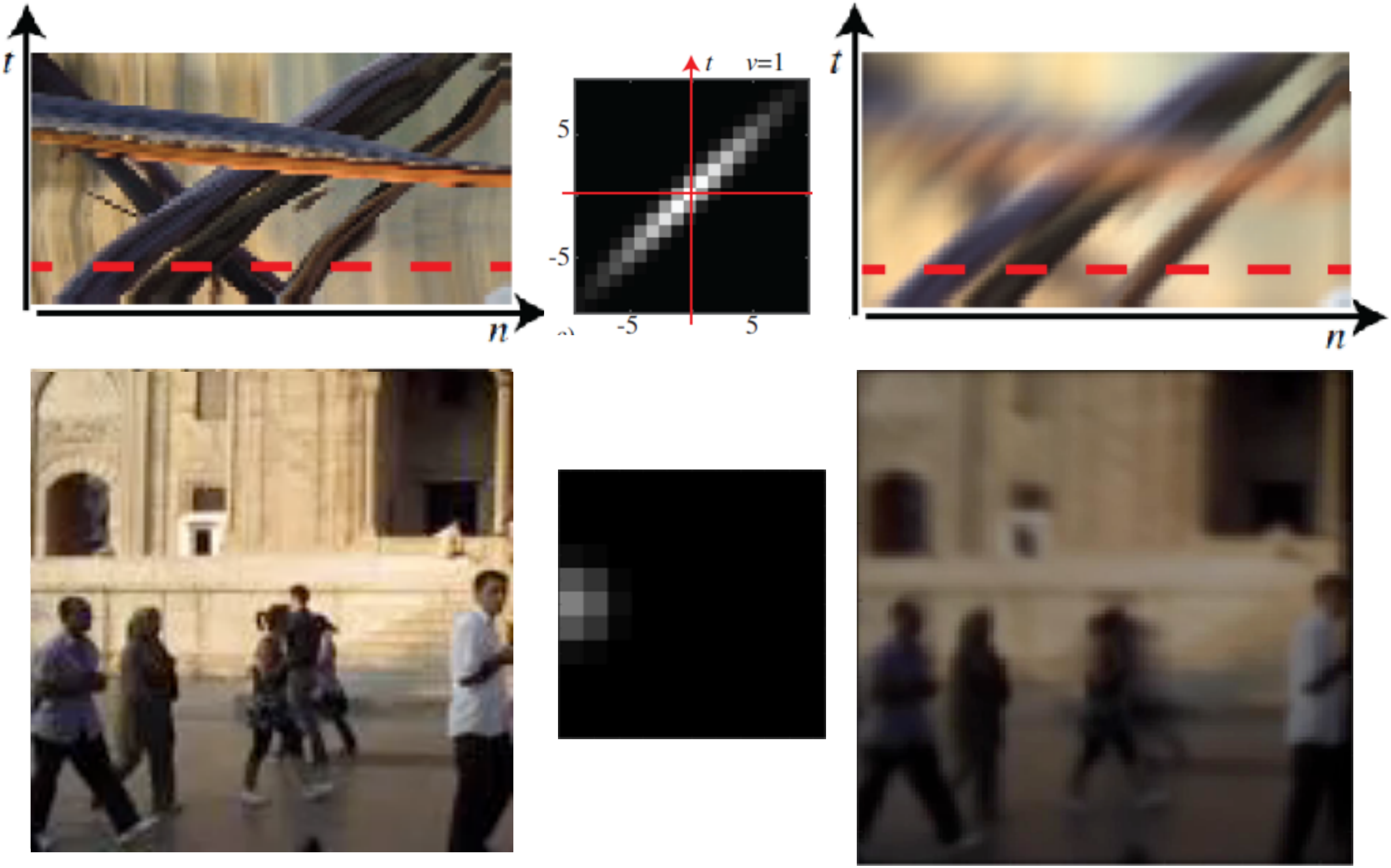


examine in frequency domain…



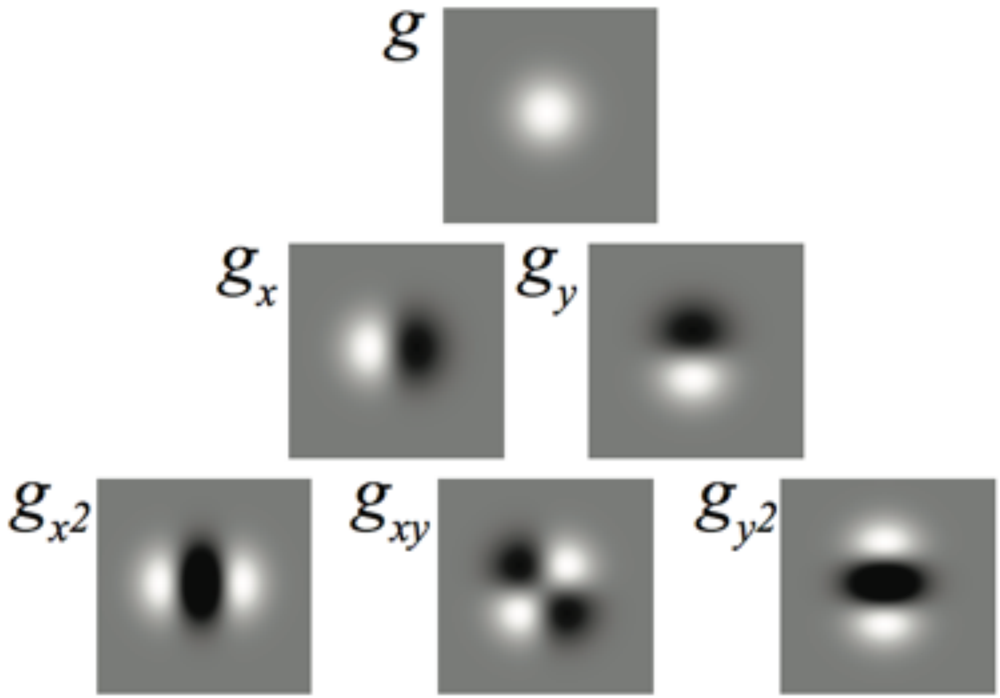
t
n
t
v=0
5
-5
-5
5
n
t
n


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

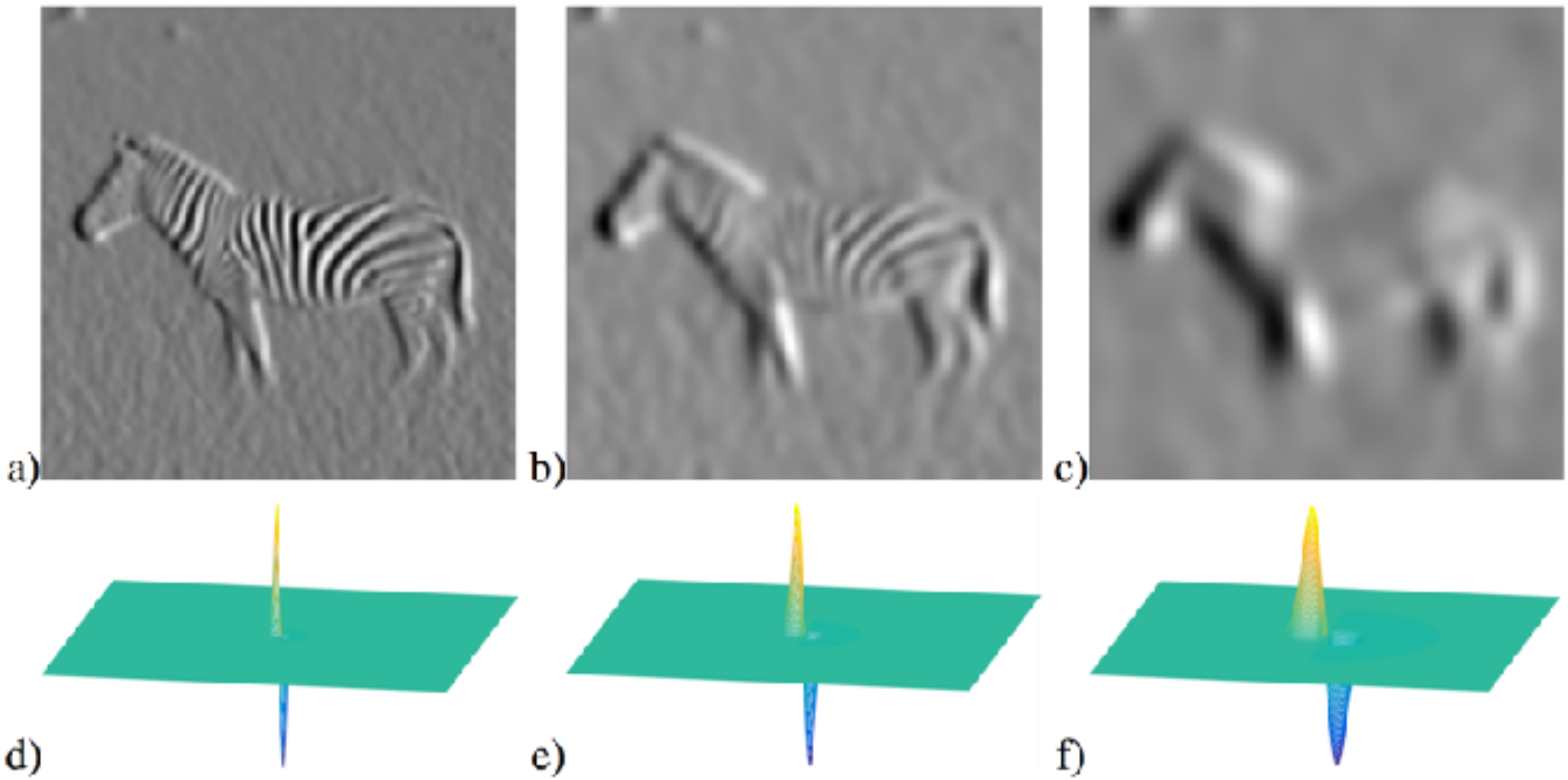


t
n
t
v=1
5
-5
-5
5
t
n

derivatives of Gaussians



derivatives of Gaussians



Space-time Gaussian derivatives

$$\frac{\partial g}{\partial t} = \frac{-t}{\sigma_t^2} g(x, y, t)$$

$$\nabla g = \left(g_x(x,y,t), g_y(x,y,t), g_t(x,y,t)\right) =$$
$$= \left(-x/\sigma^2, -y/\sigma^2, -t/\sigma_t^2\right) g(x,y,t)$$

Note: we can discretize time derivatives in the same way we discretized spatial derivatives. For instance:

$$f\,[m,n,t] - f\,[m,n,t-1]$$

Cancelling moving objects

Can we create a filter that *removes* objects that move at some velocity **(vx , vy)** while keeping the rest?

Space-time Gaussian derivatives

For a global translation, we can write:

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

Therefore, we can write the temporal derivative of f as a function of the spatial derivatives of f_0 :

$$\frac{\partial f}{\partial t} = \frac{\partial f_0}{\partial t} = -v_x \frac{\partial f_0}{\partial x} - v_y \frac{\partial f_0}{\partial y}$$

And from here (using derivatives of f):

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} = 0$$

This relation is known as the "Brightness change constraint equation", introduced by Horn & Schunck in 1981

Space-time Gaussian derivatives

Can could we create a filter that removes objects that move at some velocity **(vx , vy)** while keeping the rest?

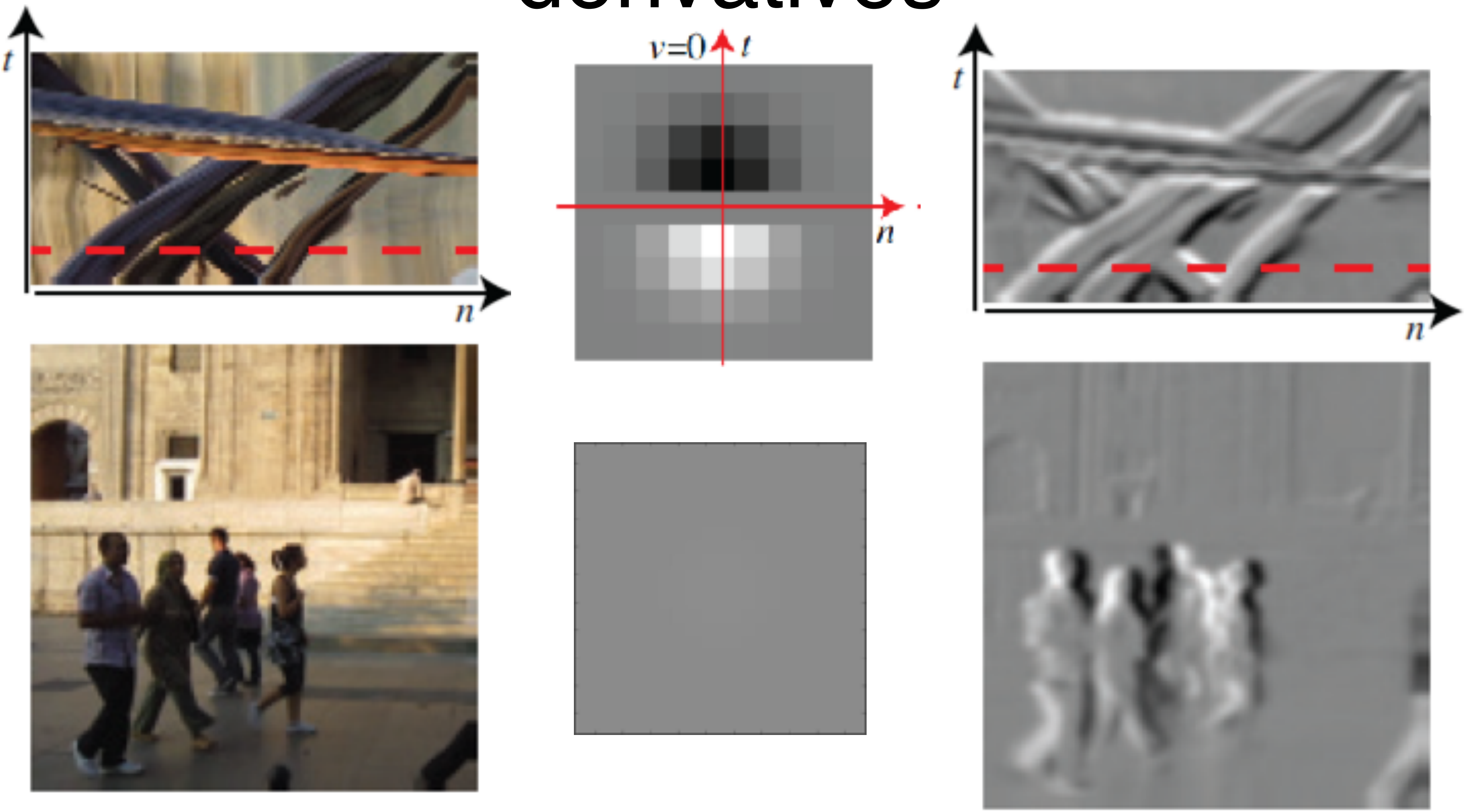
Yes, we could create a filter that implements this constraint:

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} = 0$$

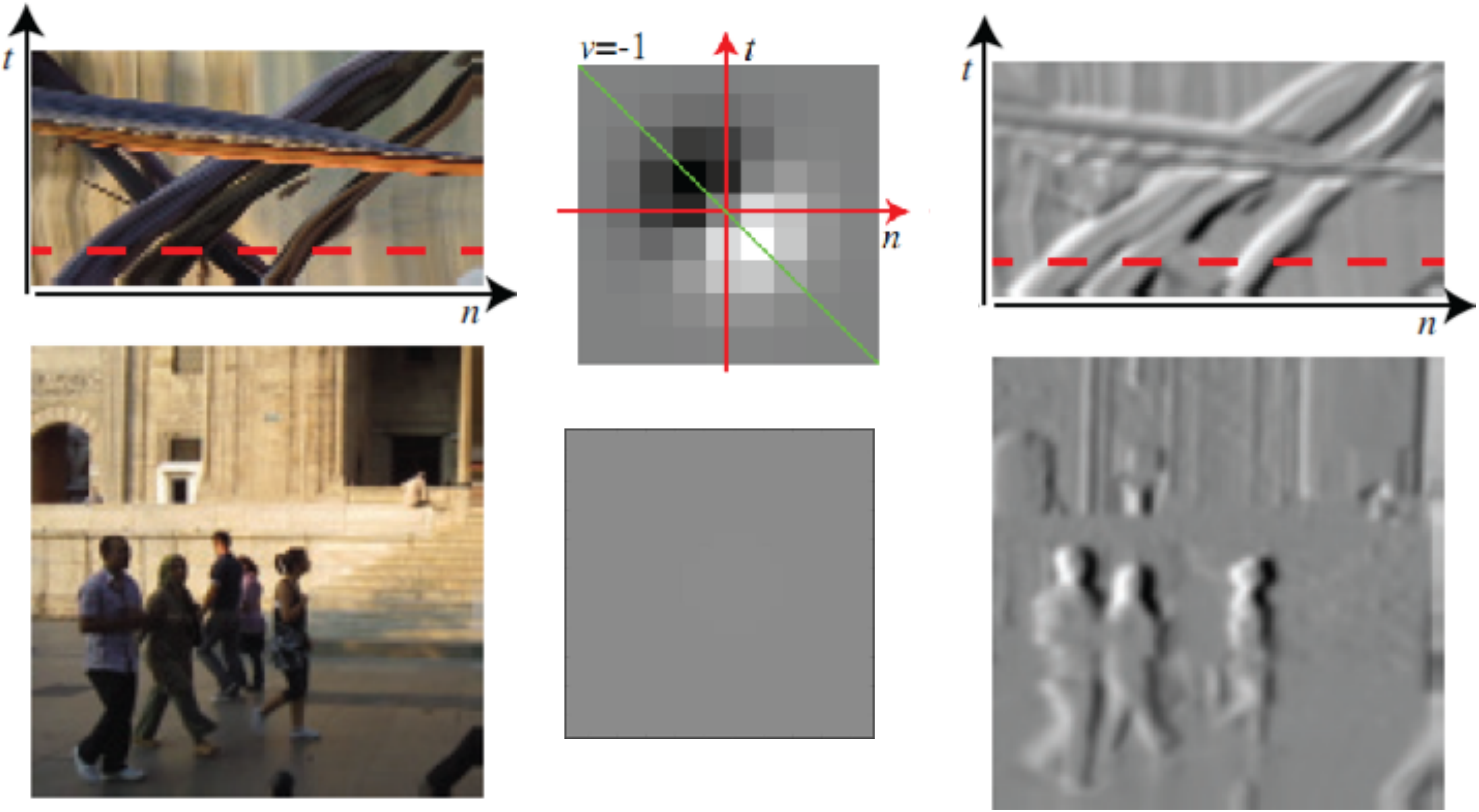
We can create this filter as a combination of Gaussian derivatives:

$$\begin{aligned} h(x, y, t; v_x, v_y) &= g_t + v_x g_x + v_y g_y \\ &= \nabla g \left(1, v_x, v_y\right)^T \end{aligned}$$

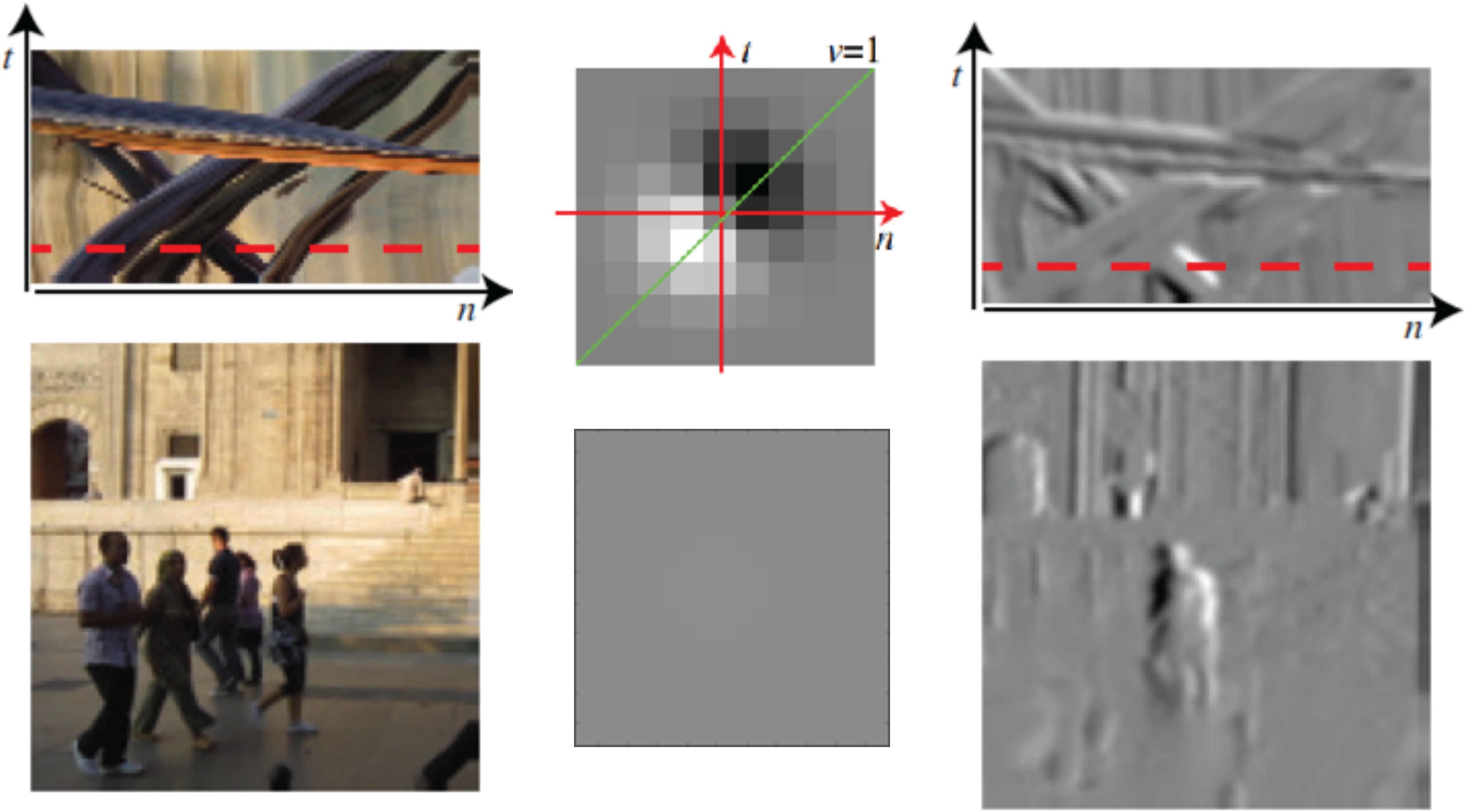
Space-time Gaussian derivatives



Nulling-out $v_x = 0$, $v_y = 0$ motion



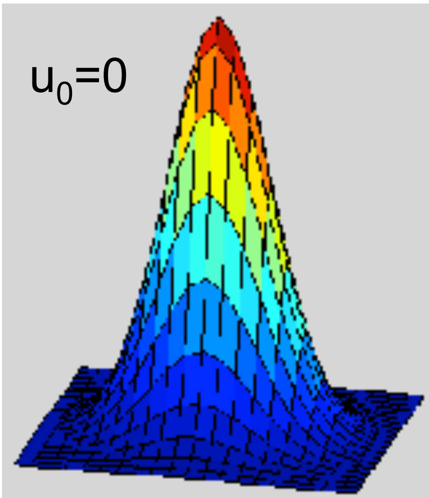
Nulling-out v_x =-1, v_y=0 motion

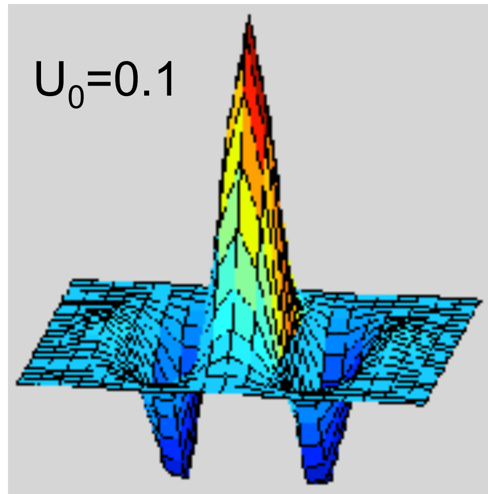


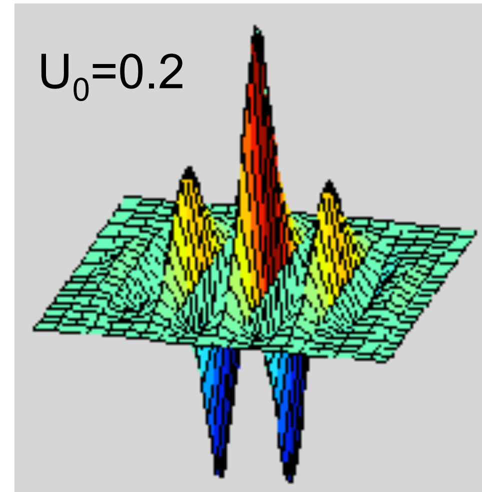
Nulling-out $v_x = 1$, $v_y = 0$ motion

Gabor wavelets

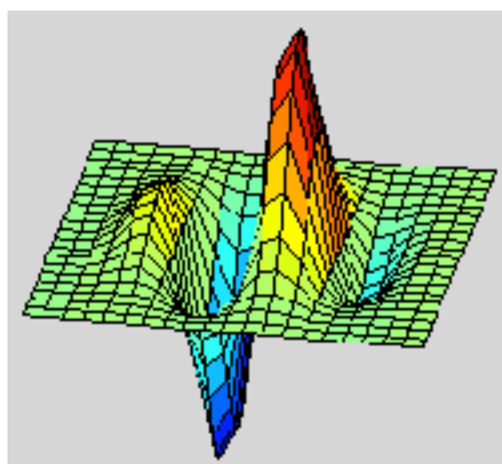
$$\psi_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi u_0 x)$$

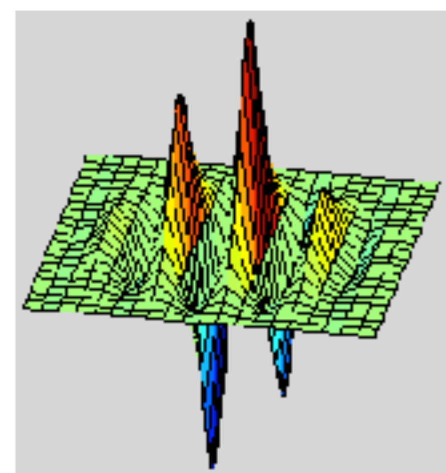


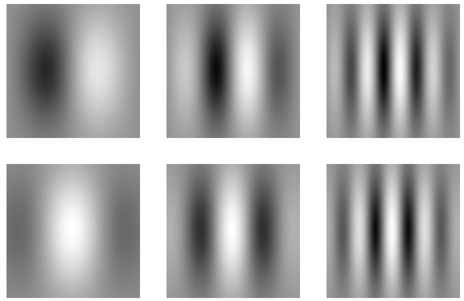




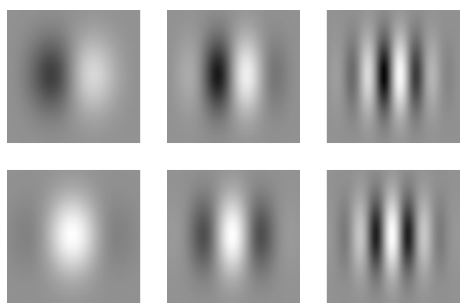
$$\psi_s(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \sin(2\pi u_0 x)$$





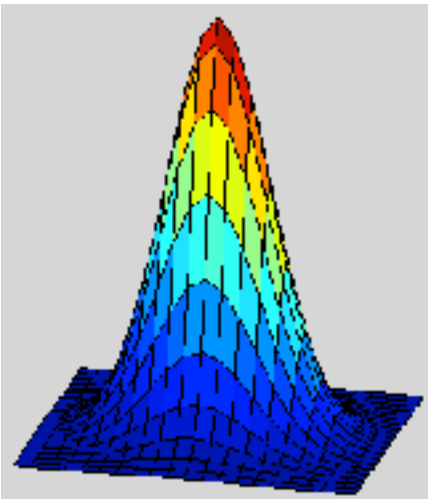


Gabor filters at different scales and spatial frequencies

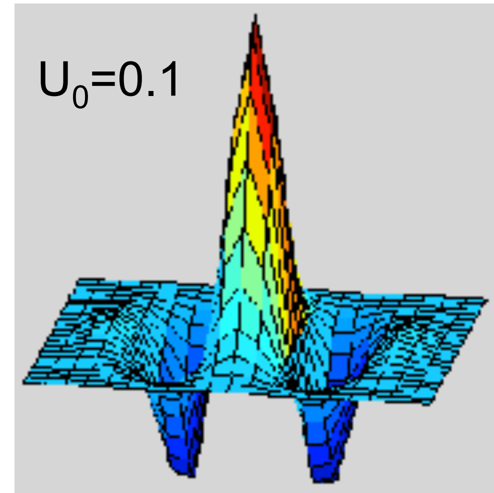


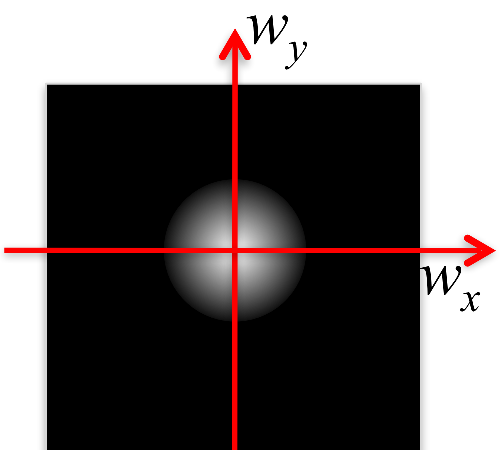
Top row shows anti-symmetric (or odd) filters; these are good for detecting odd-phase structures like edges.
Bottom row shows the symmetric (or even) filters, good for detecting line phase contours.

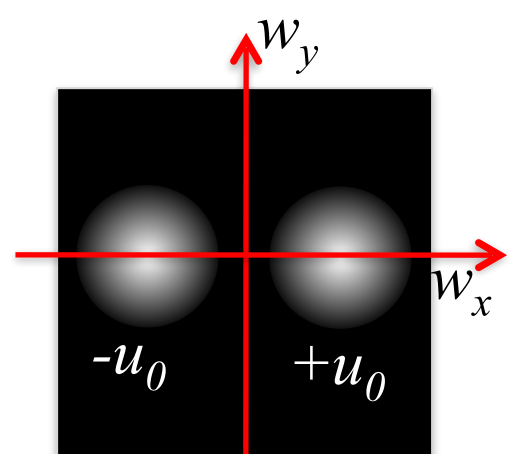
Fourier transform of a Gabor wavelet

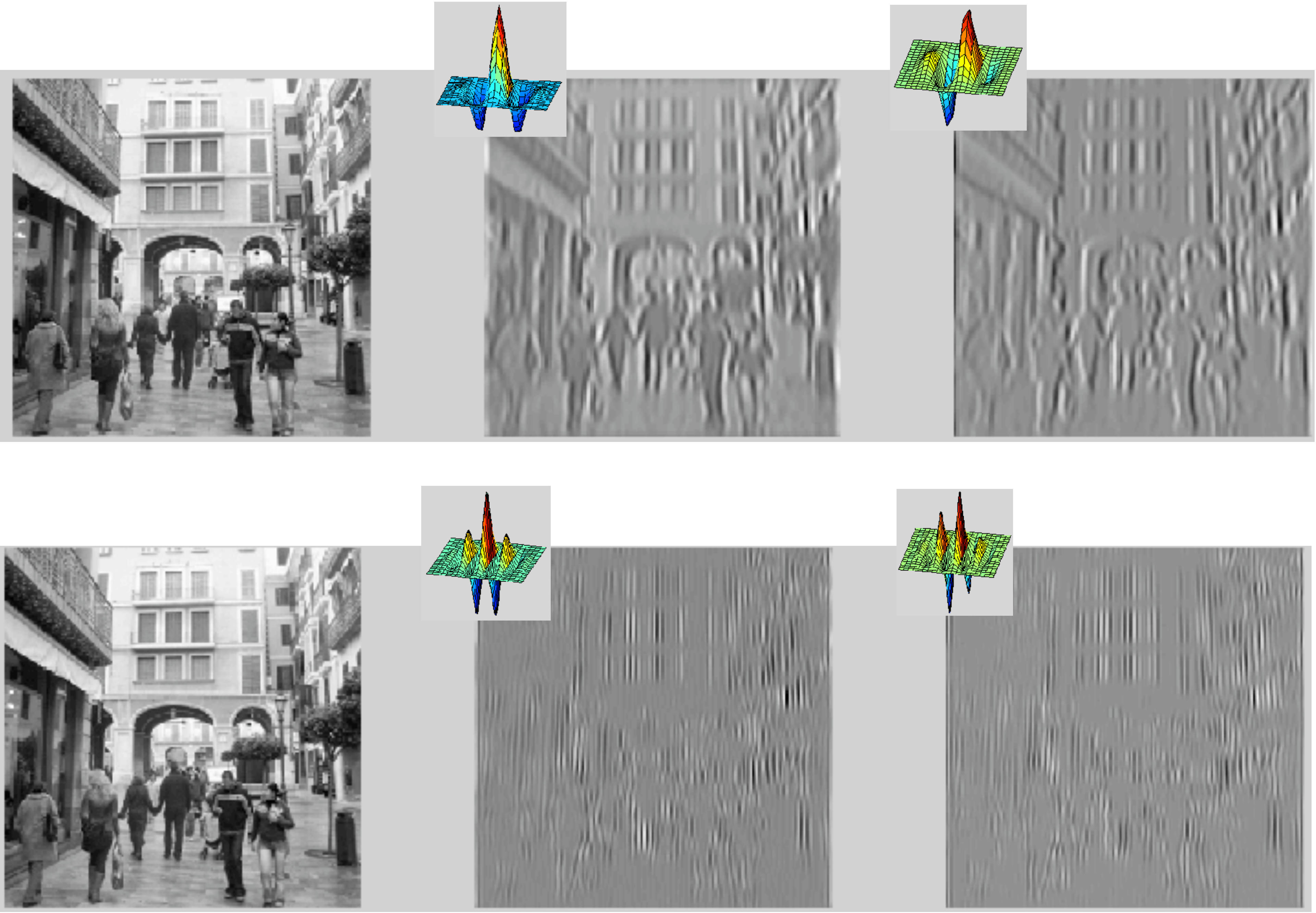


$$\psi_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi u_0 x)$$



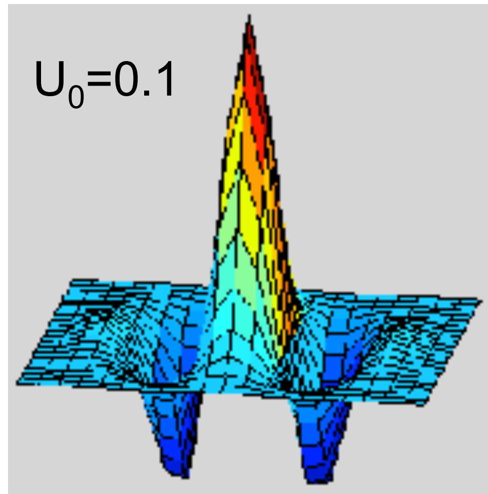




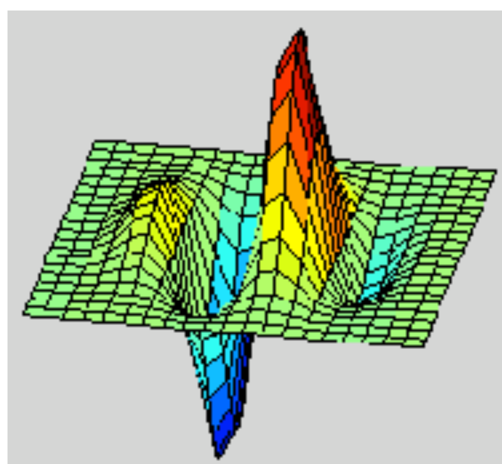


Quadrature pair

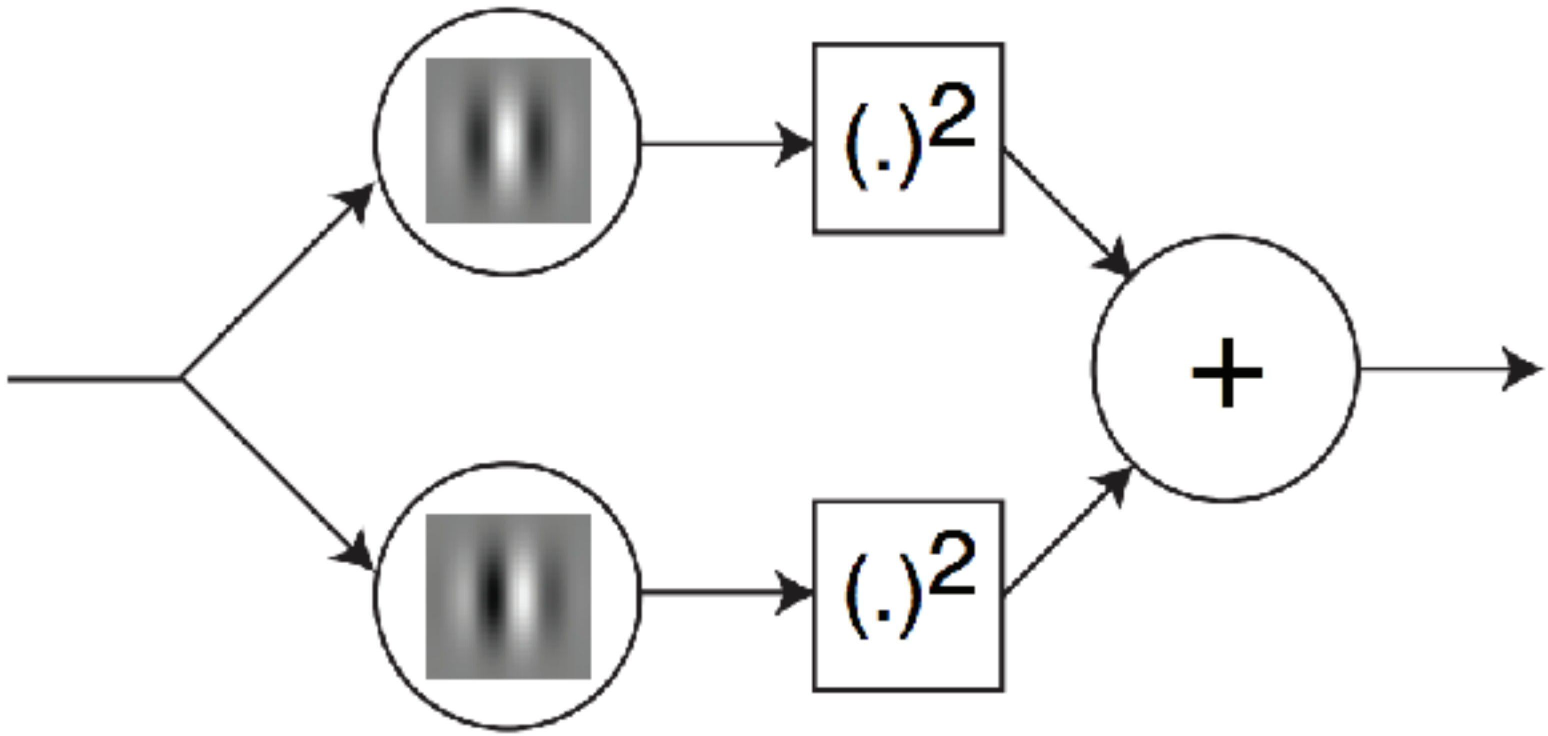
$$\psi_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi u_0 x)$$



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

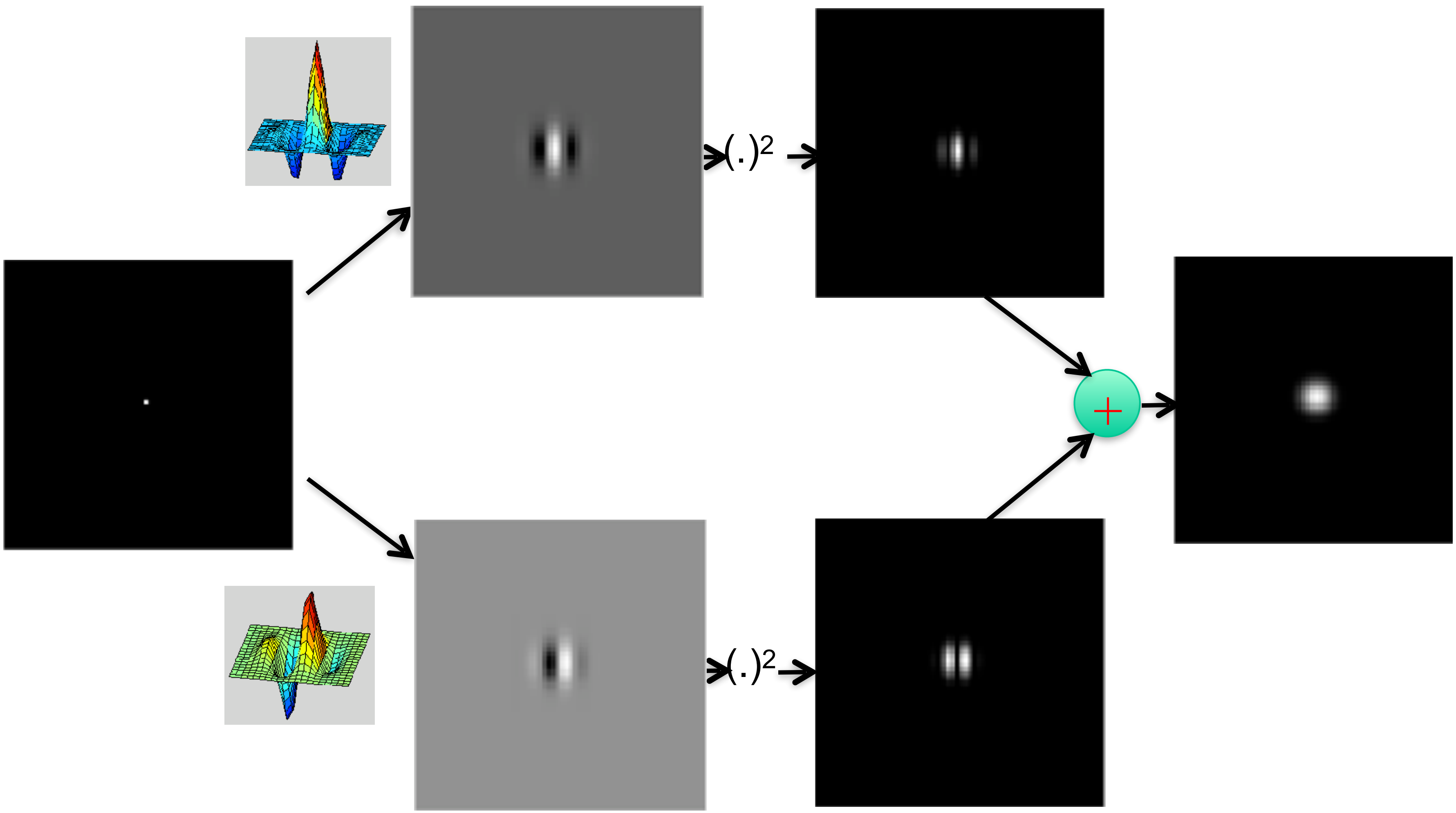


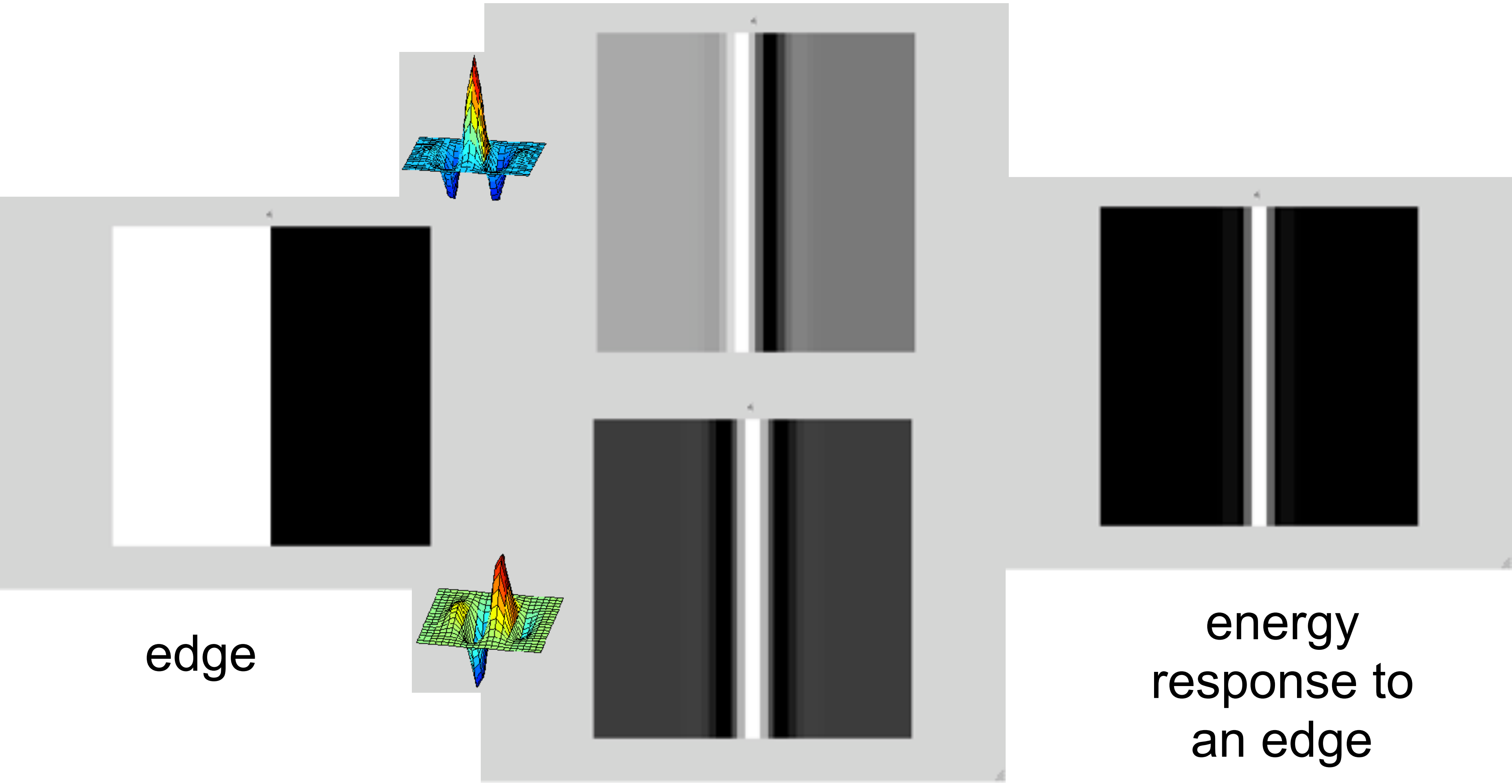
“oriented energy” from a quadrature pair

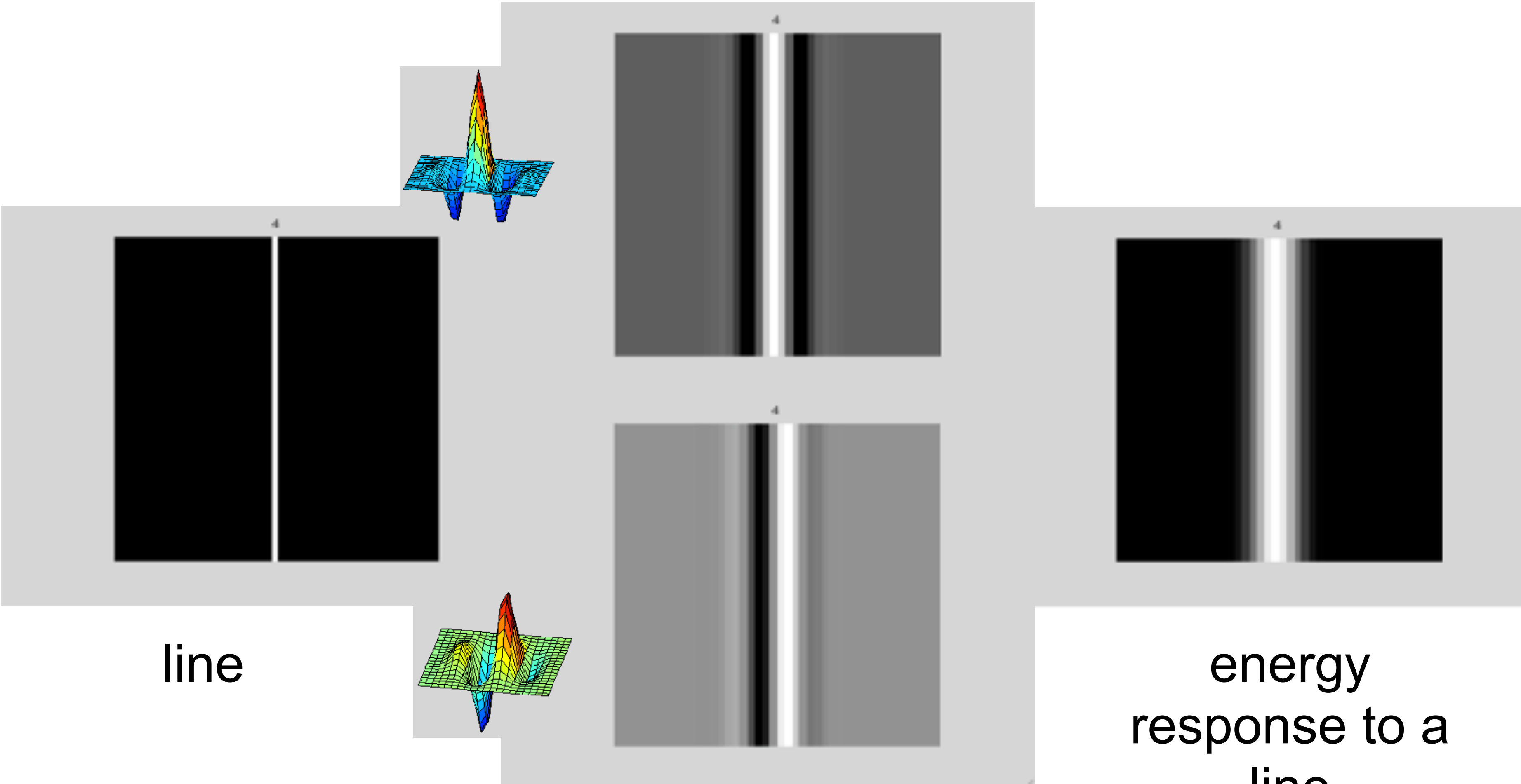


Quadrature filter pairs

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




edge
energy
response to
an edge

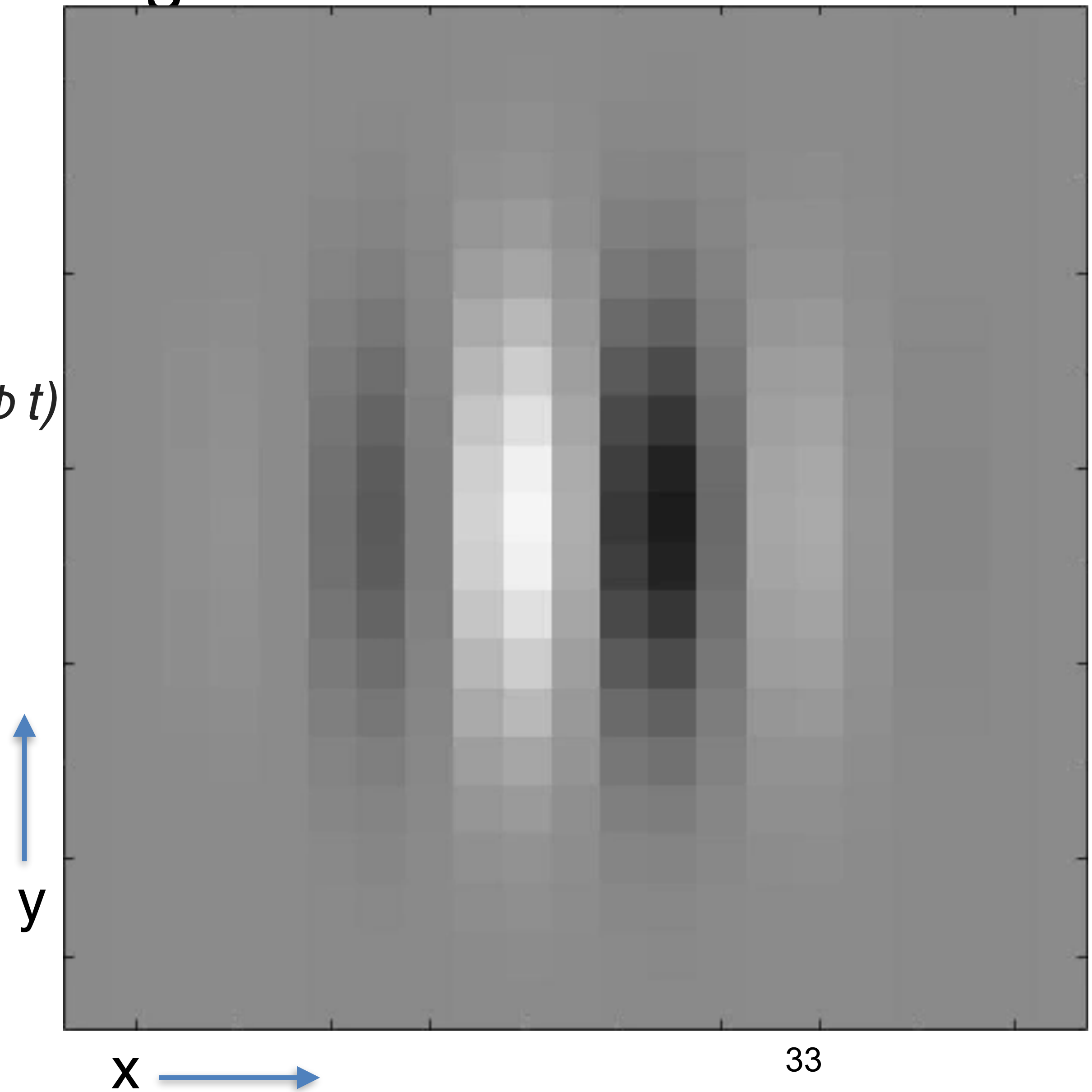


line

energy response to a line

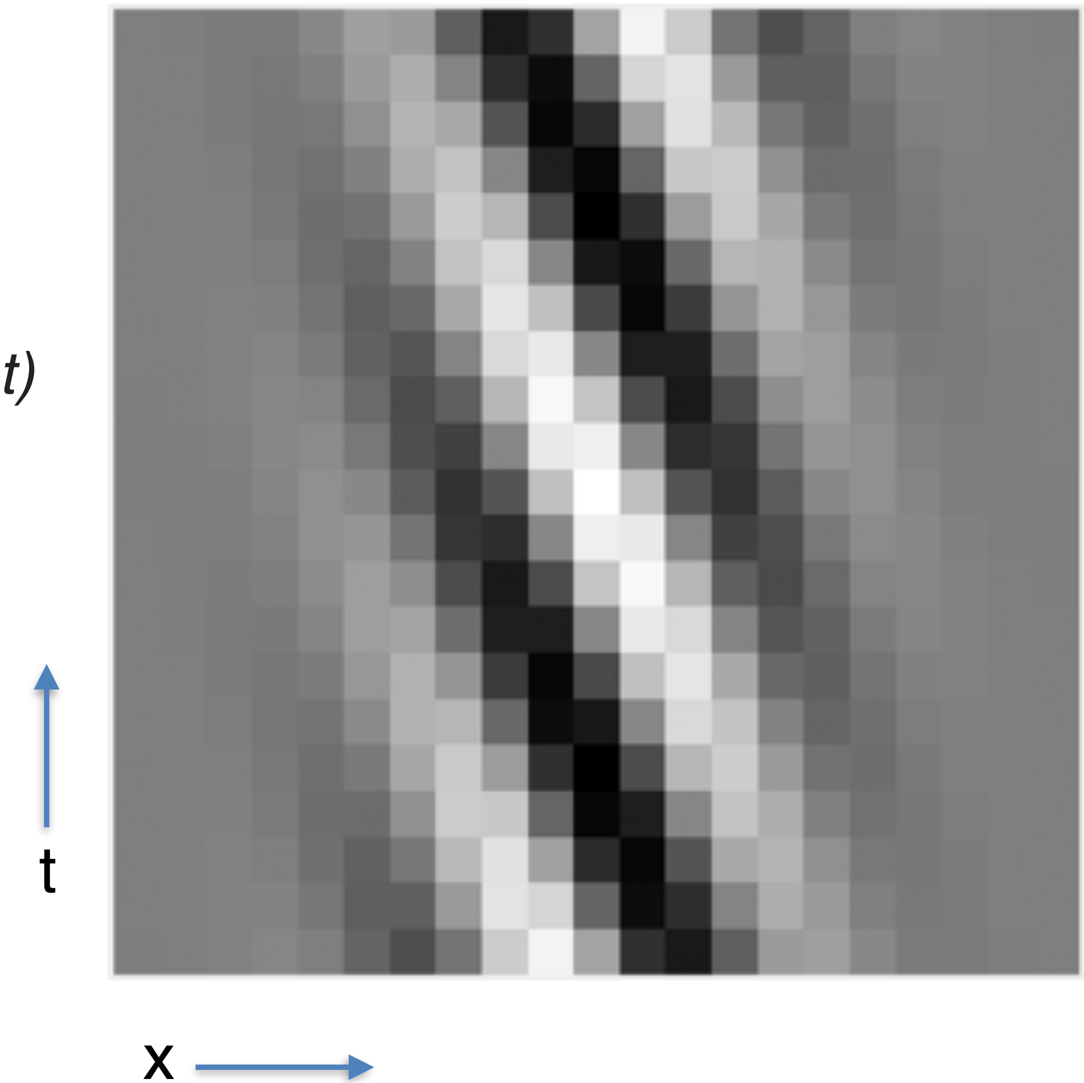
Using phase changes of local Gabor filters to analyze or generate motion

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

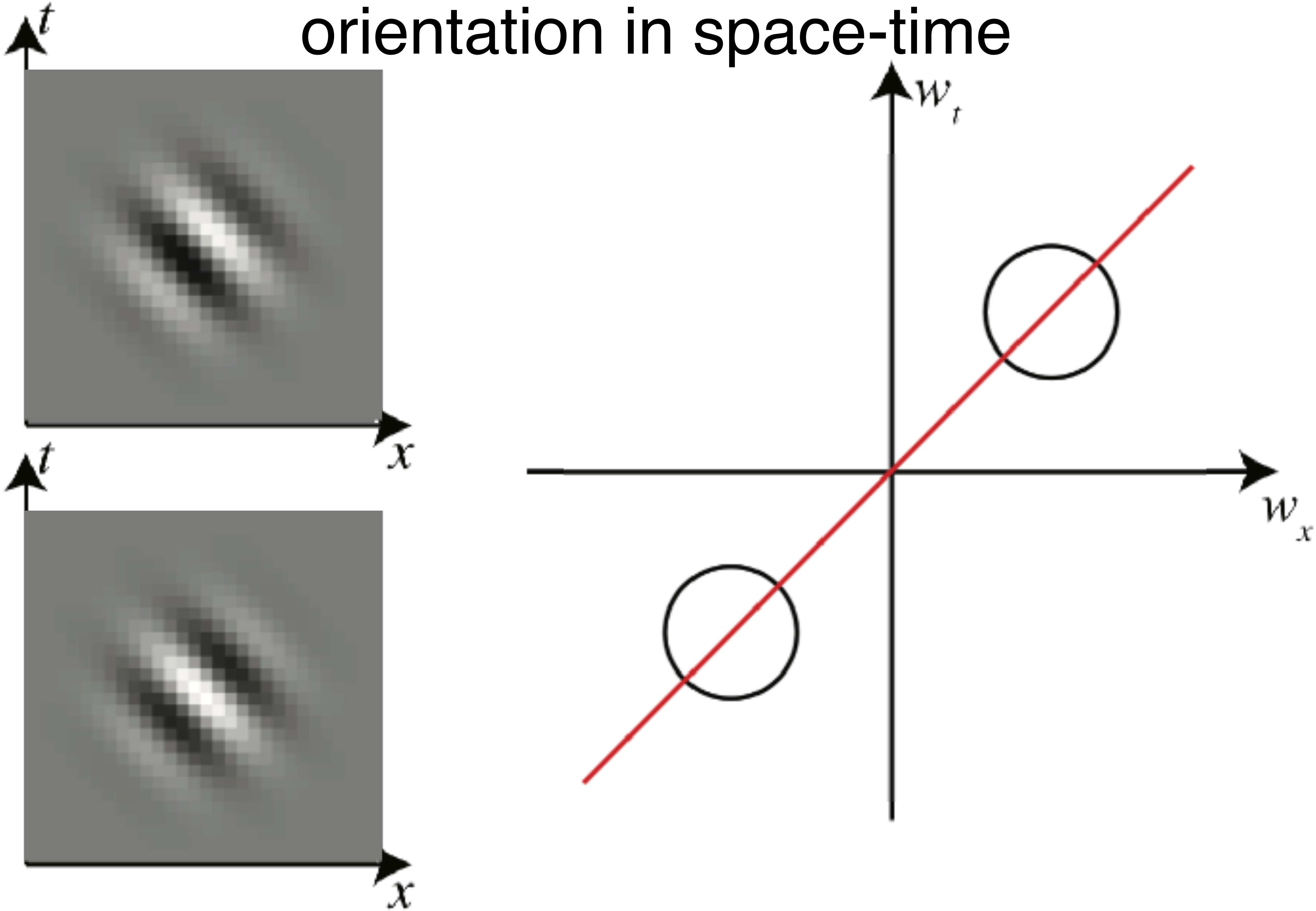


Space-time plot of the a slice through the patio-temporal filter of the previous slide

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



Gabor filters for analyzing motion as orientation in space-time



Motion without movement

SIGGRAPH '91 Las Vegas, 28 July 2 August 1991

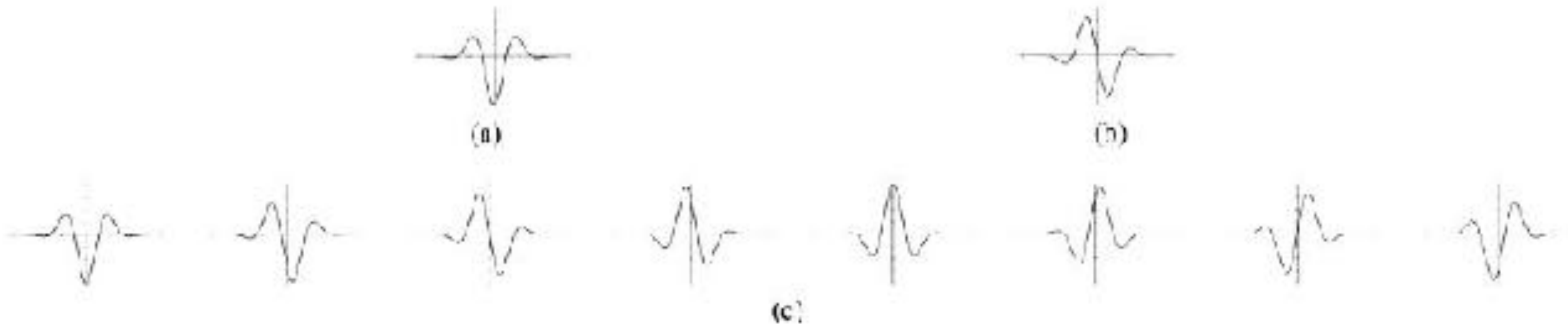


Figure 1: 1-d cross sections of filters. (a) Even phase (G_2). (b) Odd phase (H_2). (c) Filters modulated in phase according to Eq. (1). Note the apparent rightward motion of the filter ripples.

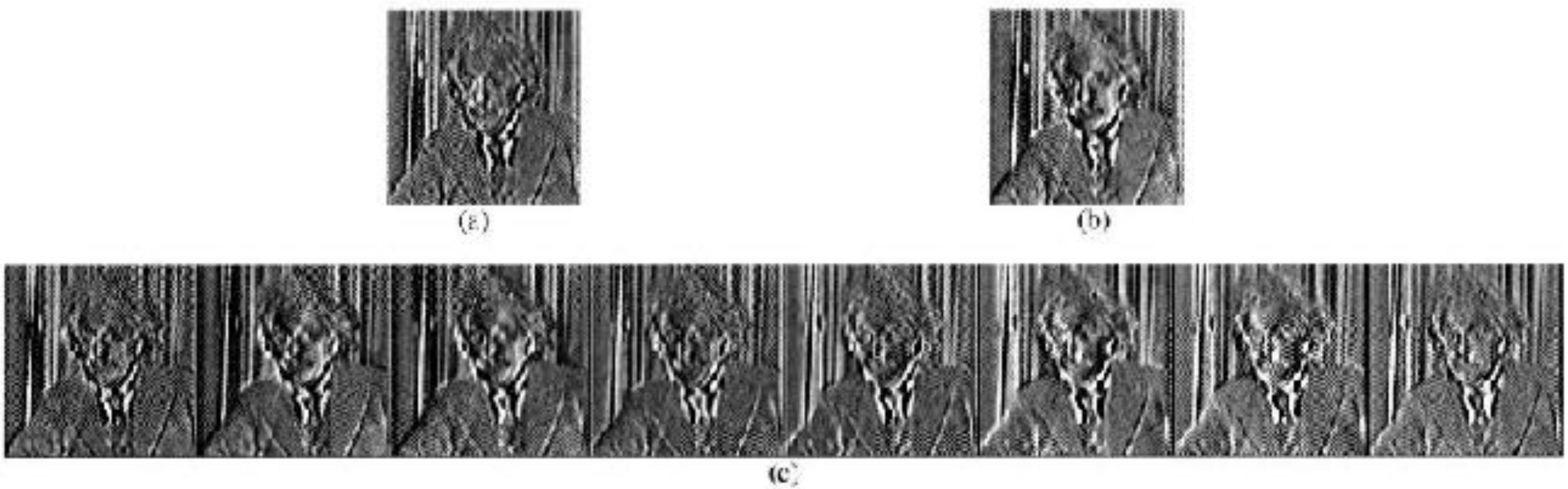
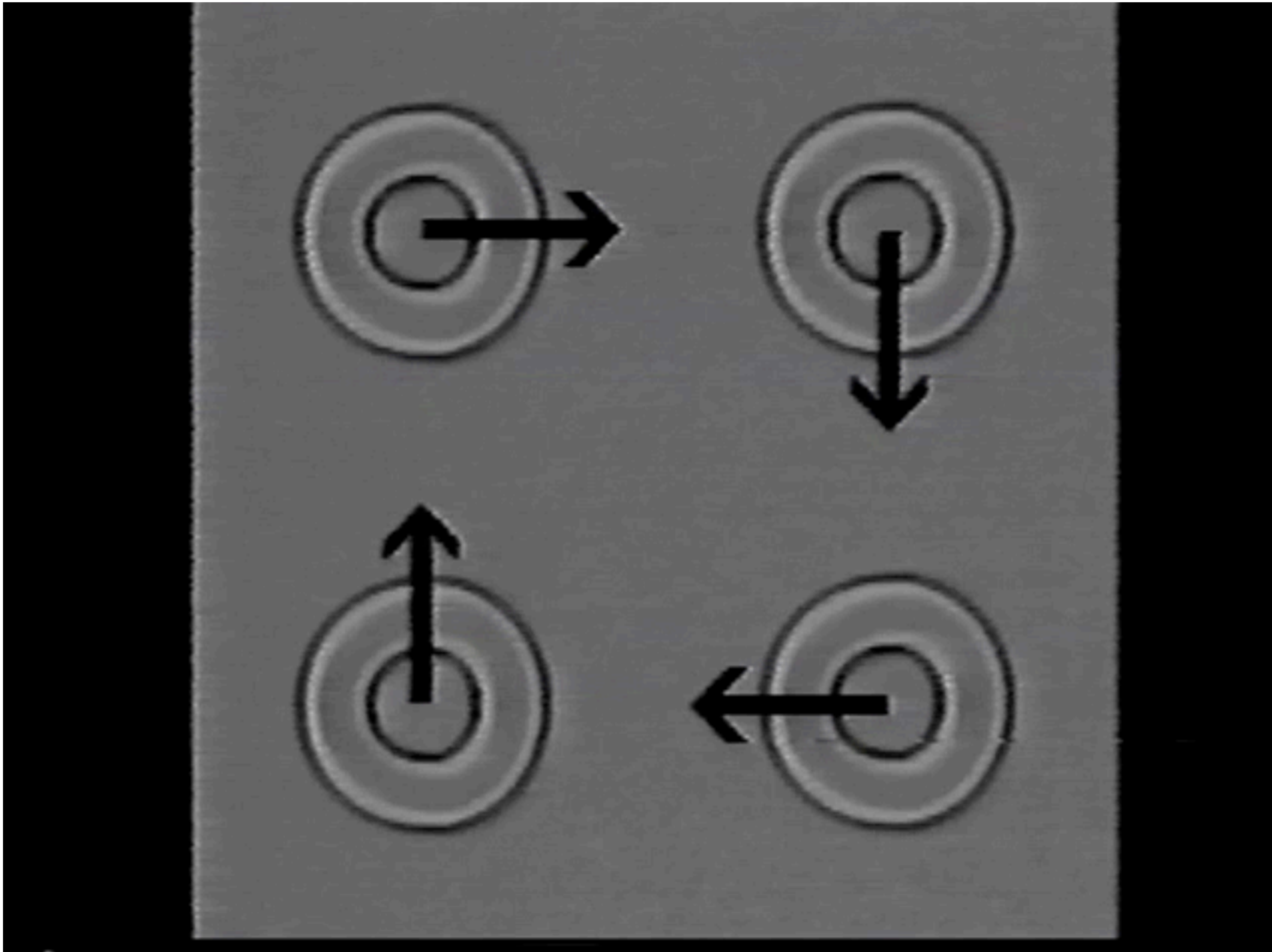


Figure 2: (a) and (b): G_2 and H_2 filters were applied to an image of Einstein. (c) Images modulated as in Eq. (1). When viewed as a temporal sequence, this generates the perception of rightward motion, yet image remains stationary

Motion without movement

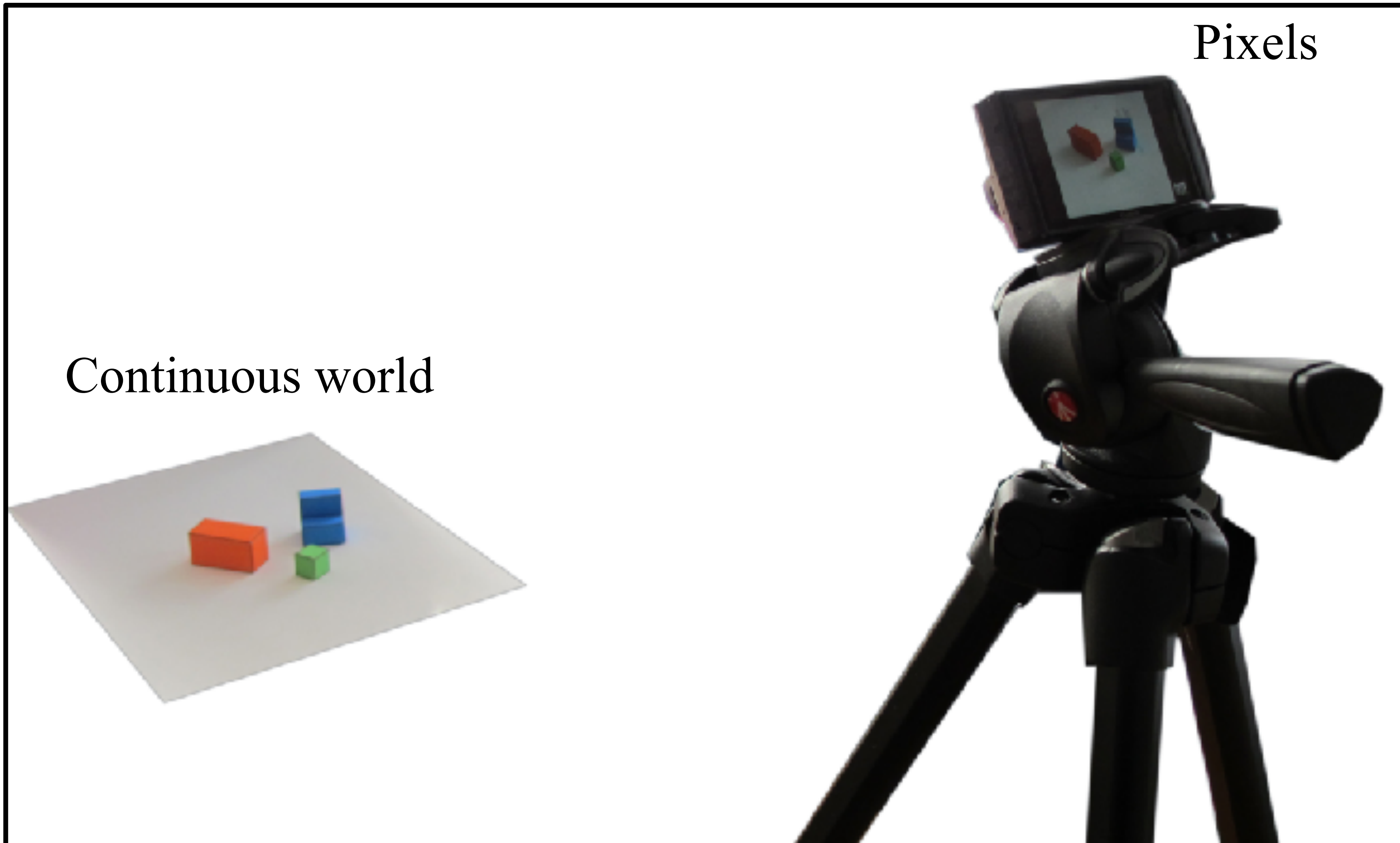


original

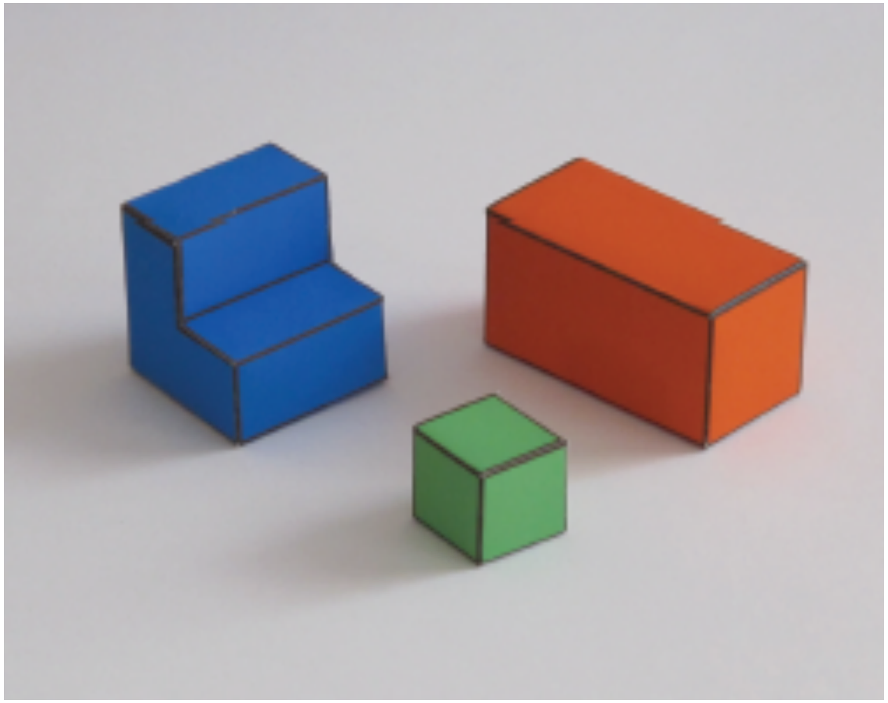
original

Sampling

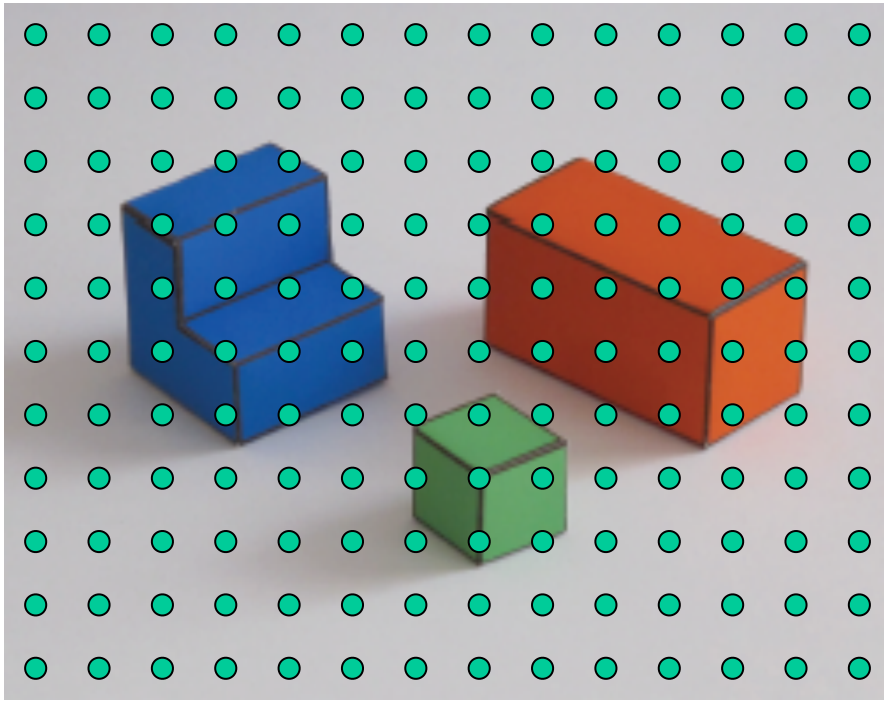
Sampling



Sampling



Sampling

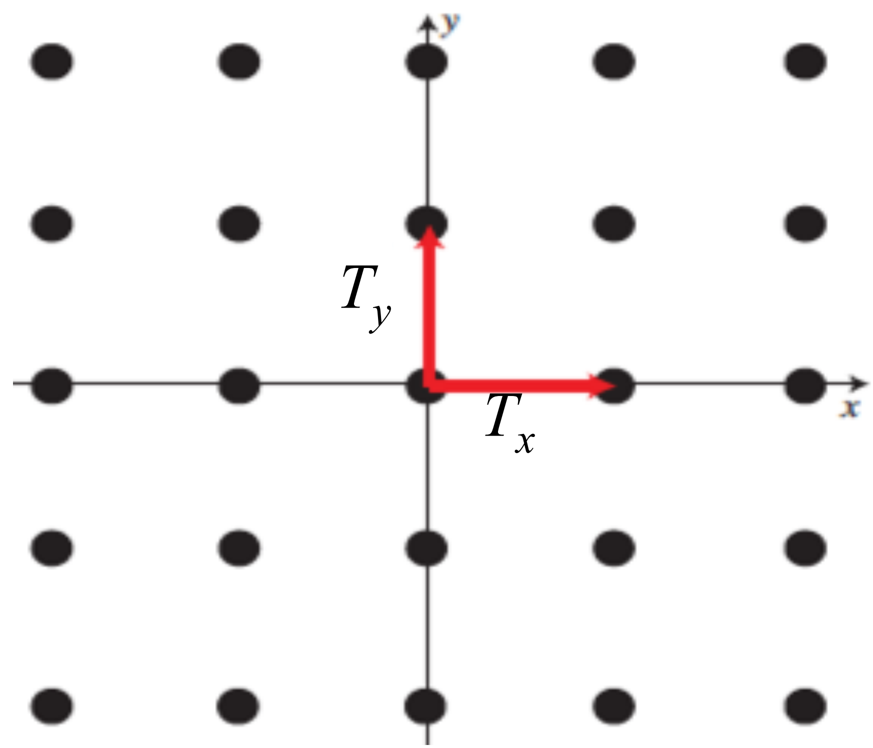


Sampling

Continuous image $f(x, y)$

We can sample it using a rectangular grid as

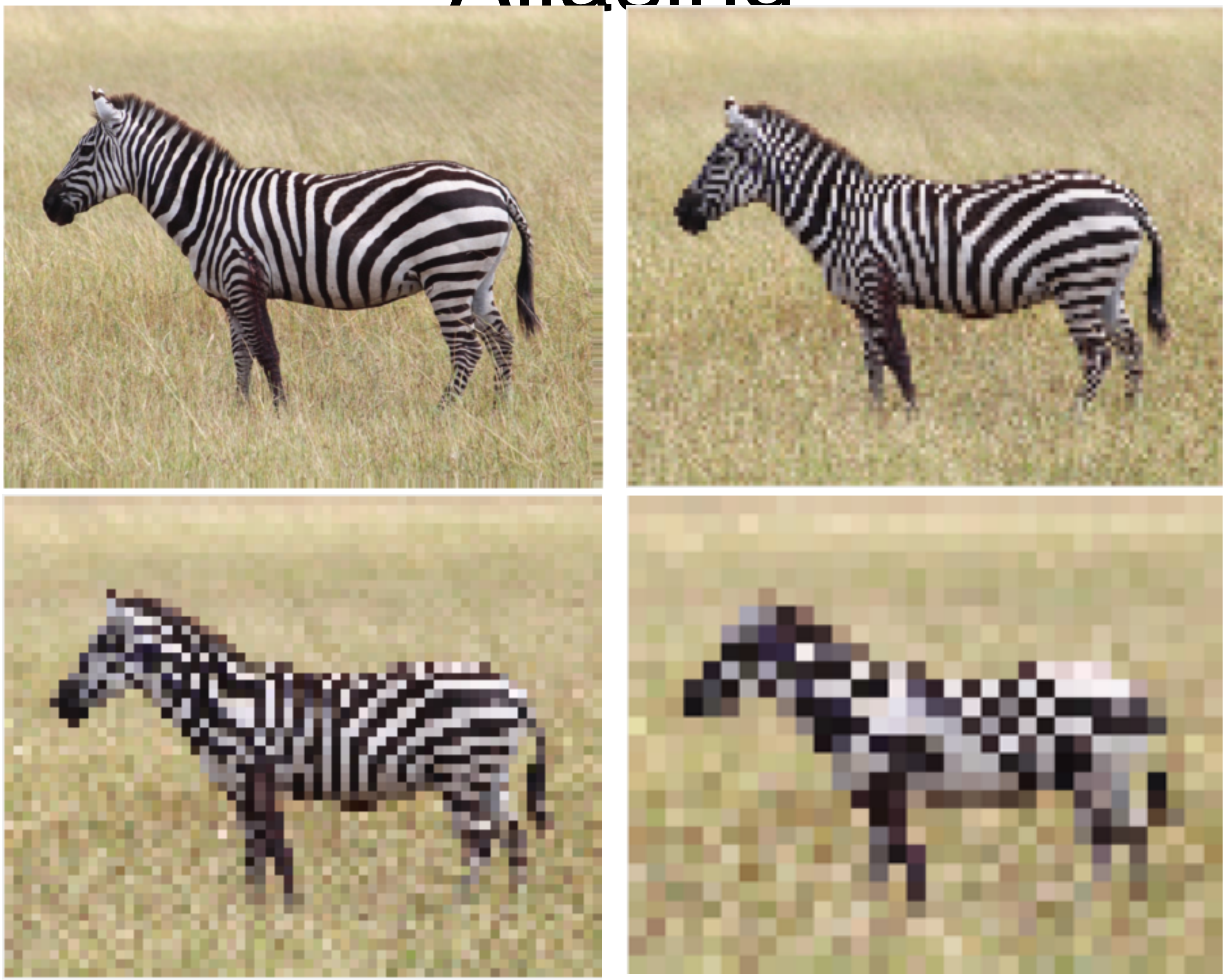
$$f[n,m] = f\left(nT_x, mT_y\right)$$



Aliasing

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

Aliasing



103×128

52×64

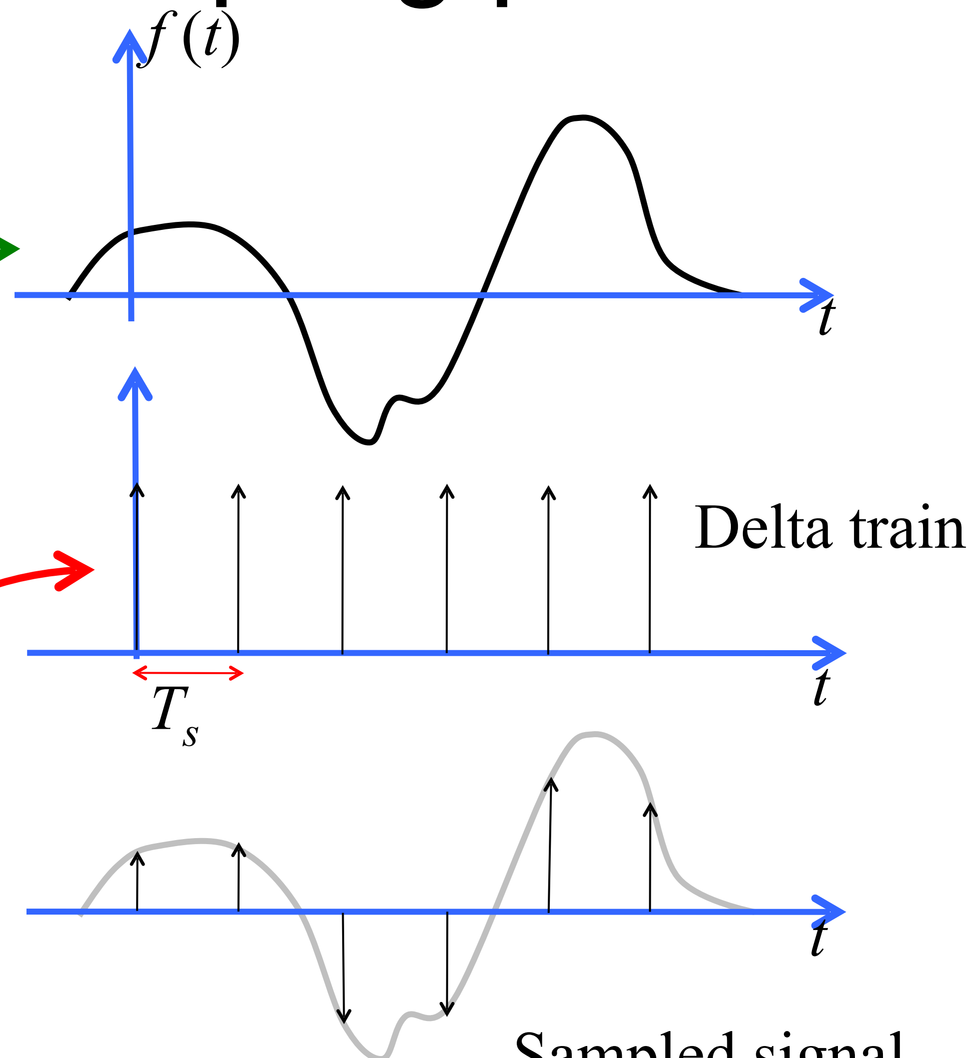
26×32

Modeling the sampling process

$$f[n] = f(nT_s)$$

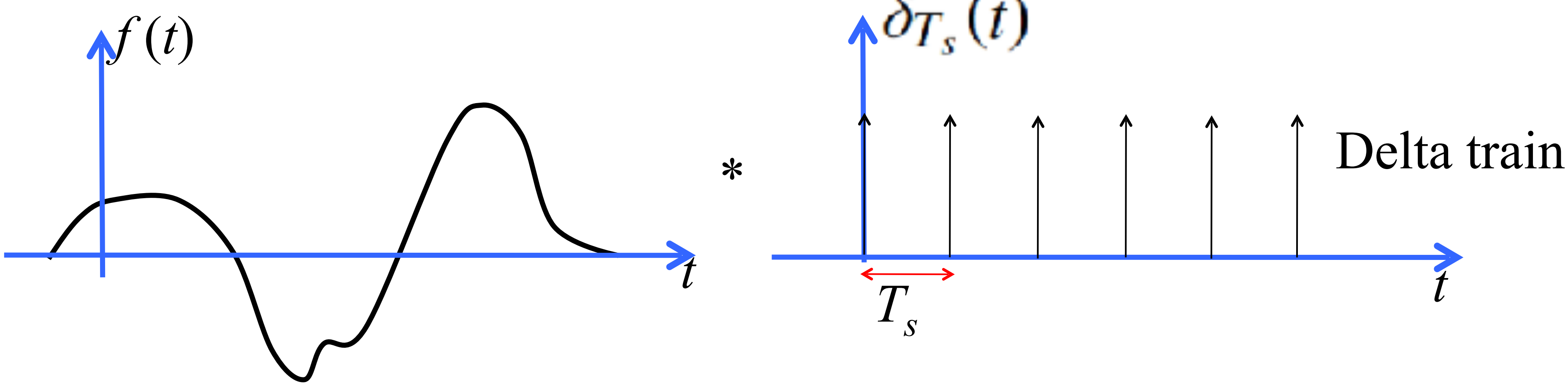
A convenient writing:

$$\hat{f}(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$



Modeling the sampling process

$$\hat{f}(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = f(t)\delta_{T_s}(t)$$

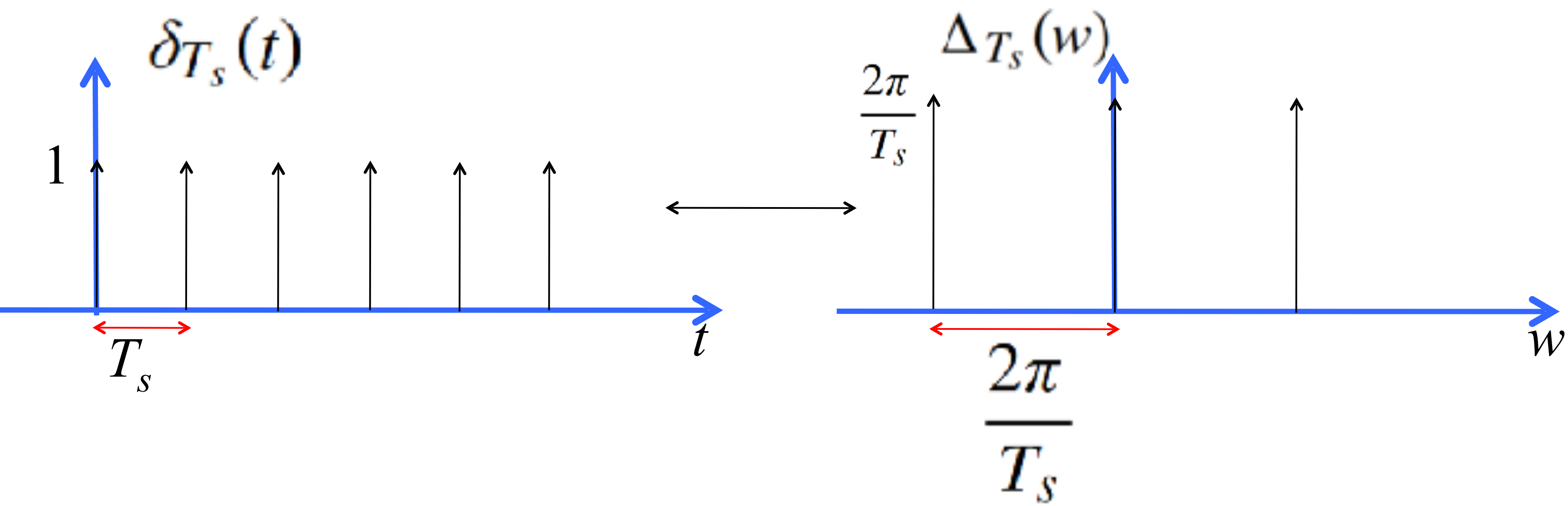


The Fourier transform is a convolution…

Interesting property of the delta train: the Fourier transform of a delta train of period T is another delta train with period 2π/T

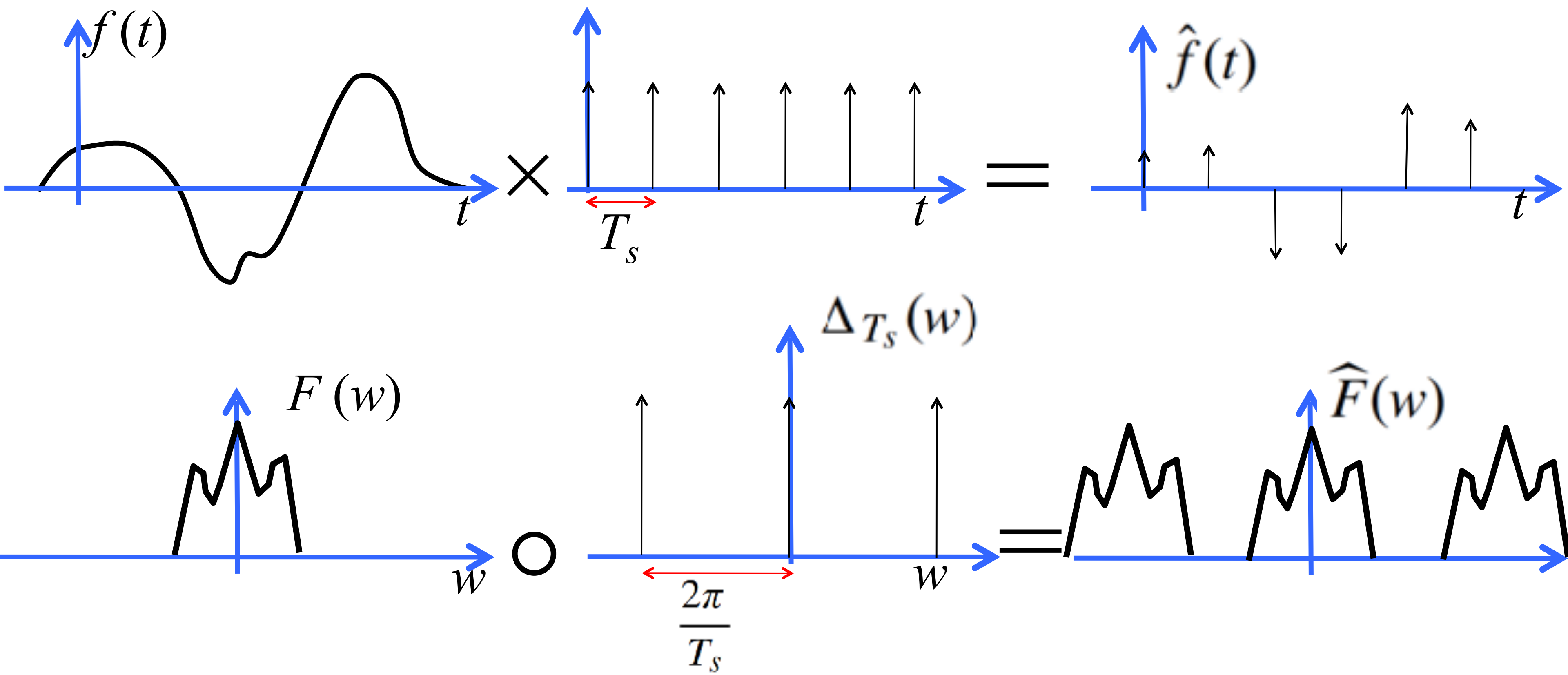
Modeling the sampling process

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \longleftrightarrow \Delta_{T_s}(w) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta\left(w - k\frac{2\pi}{T_s}\right)$$

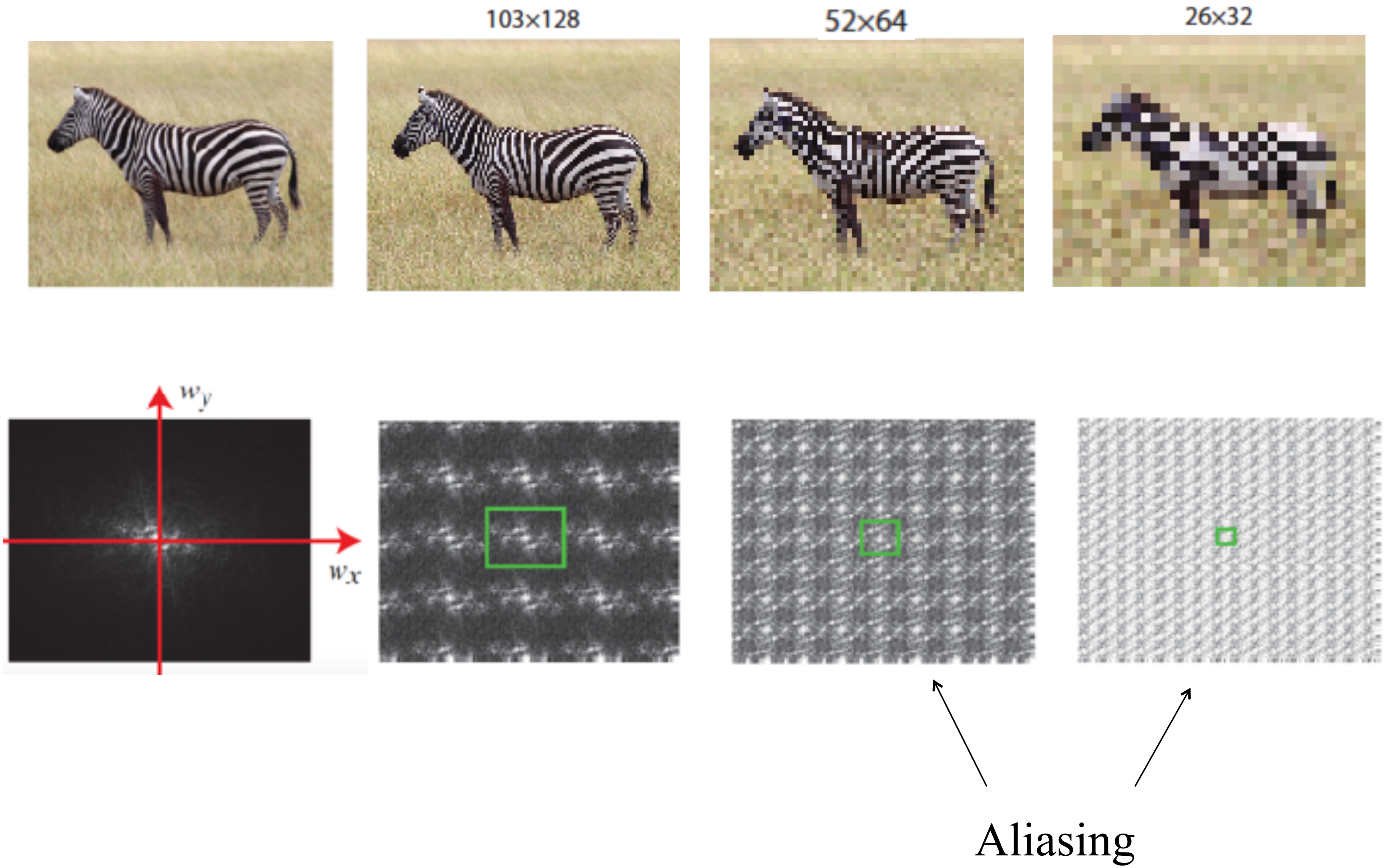


Interesting property of the delta train: the Fourier transform of a delta train of period T is another delta train with period 2π/T. Demo in the class notes.

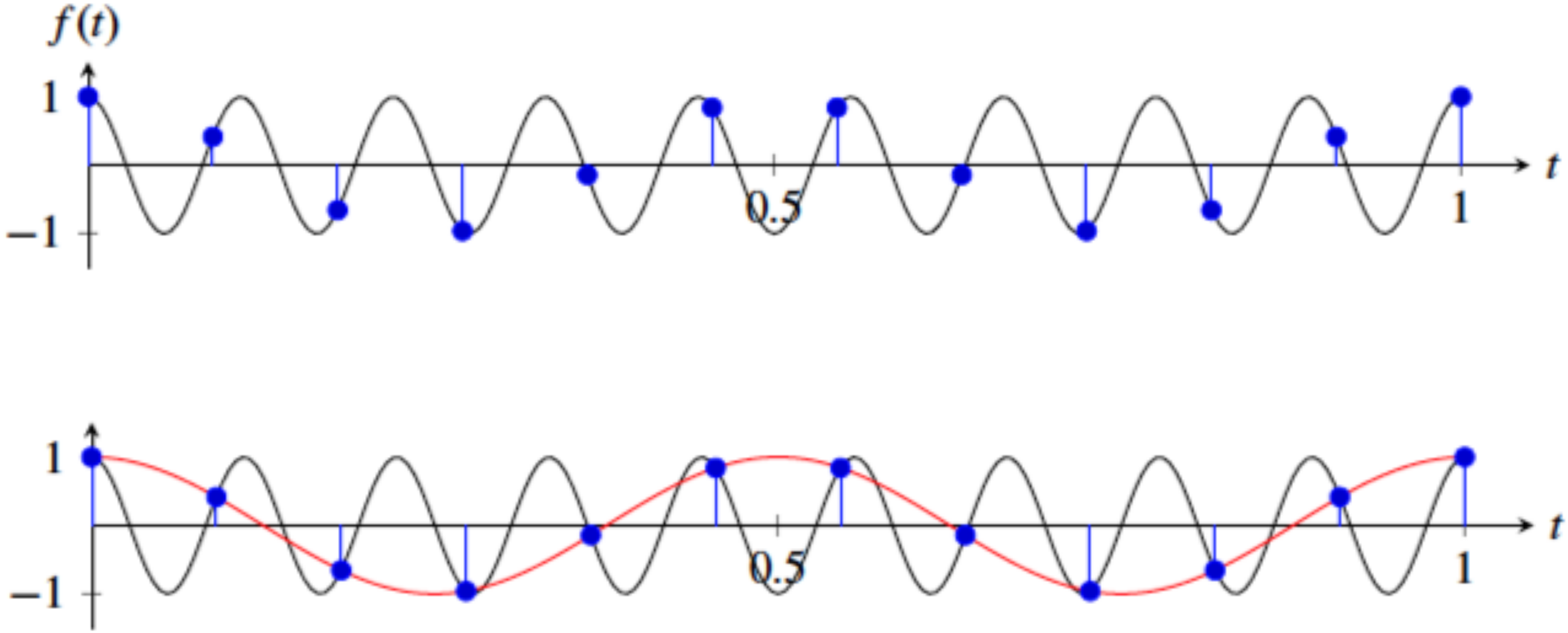
Modeling the sampling process



What happens when the repetitions overlap?


103×128
52×64
26×32
w_y
w_x
Aliasing

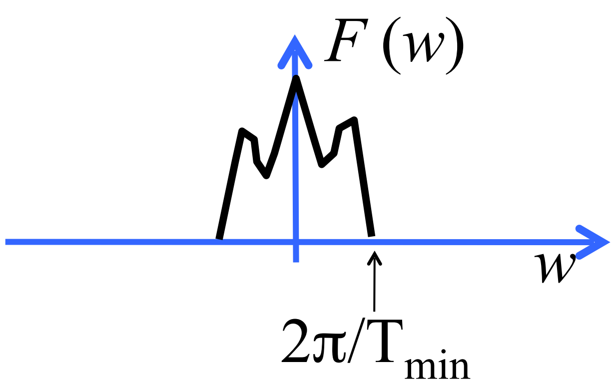
Aliasing



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

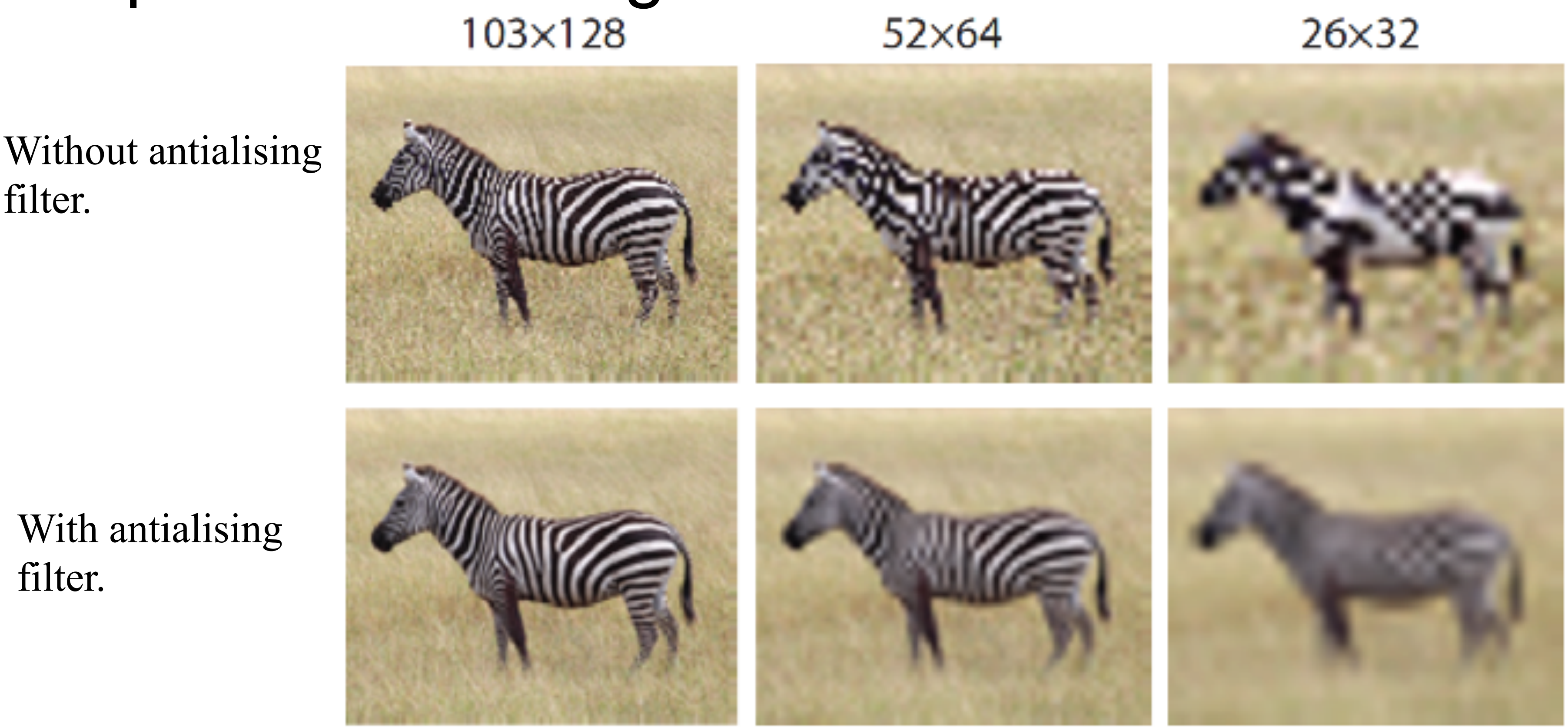
Sampling theorem

The sampling theorem (also known as Nyquist theorem) states that for a signal to be perfectly reconstructed from it samples, the sampling period T_s has to be $T_s > T_{min}/2$ where T_{min} is the period of the highest frequency present in the input signal.



Antialising filtering

Before sampling, apply a low pass-filter to remove all the frequencies that will produce aliasing.



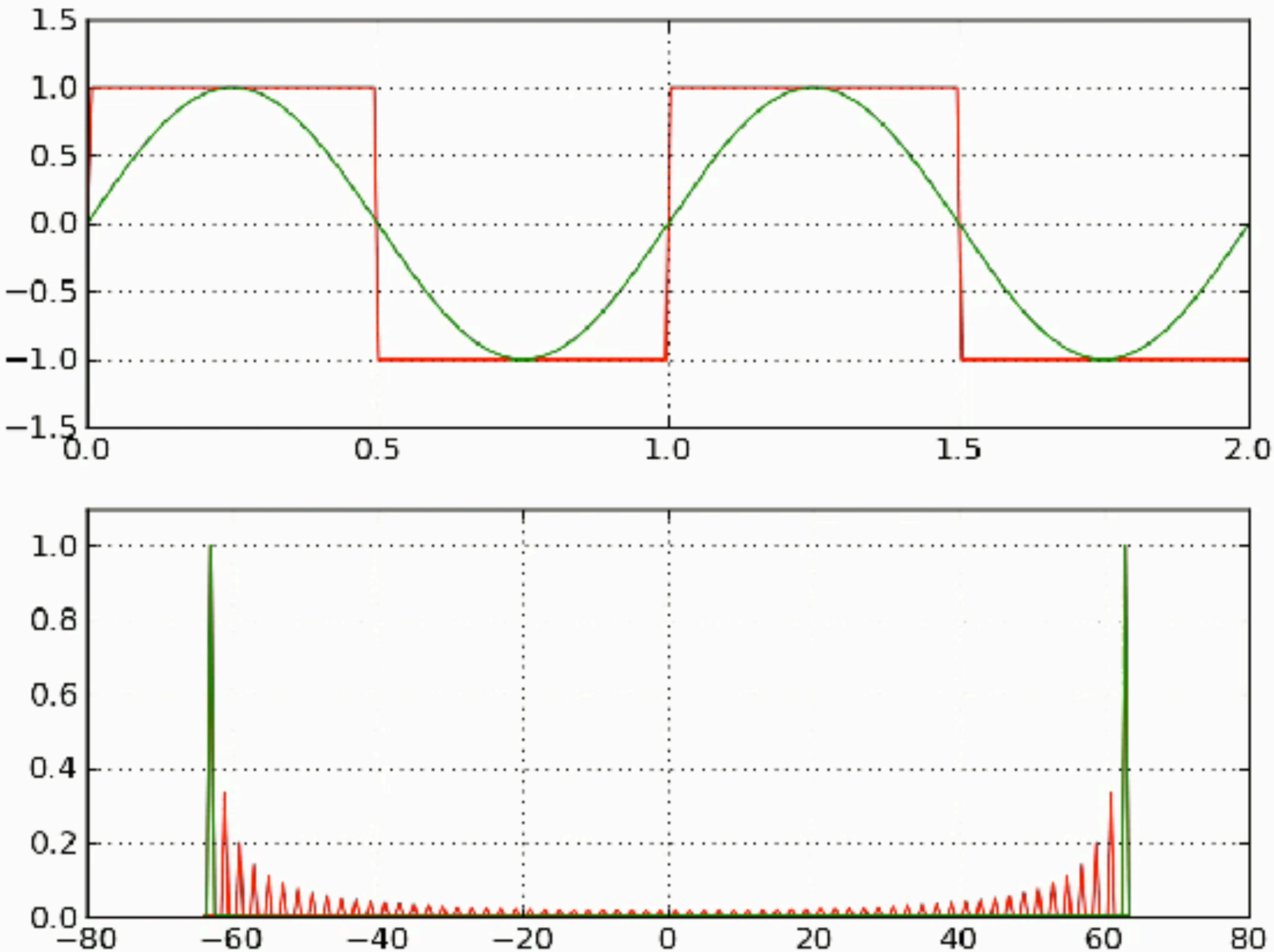
Evidence for filter-based analysis of motion in the human visual system

Square wave Fourier components

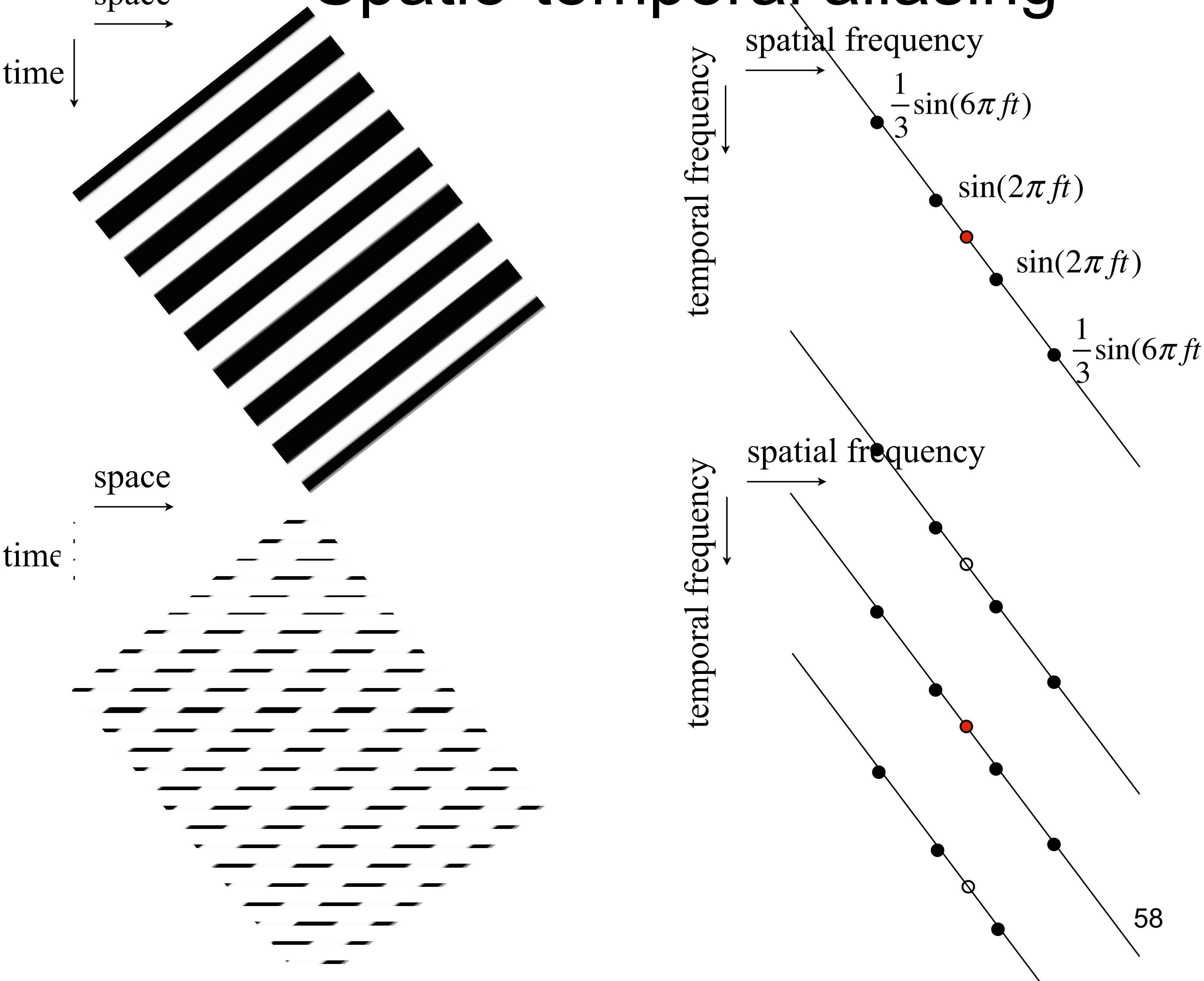
Using Fourier series we can write an ideal square wave as an infinite series of the form

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

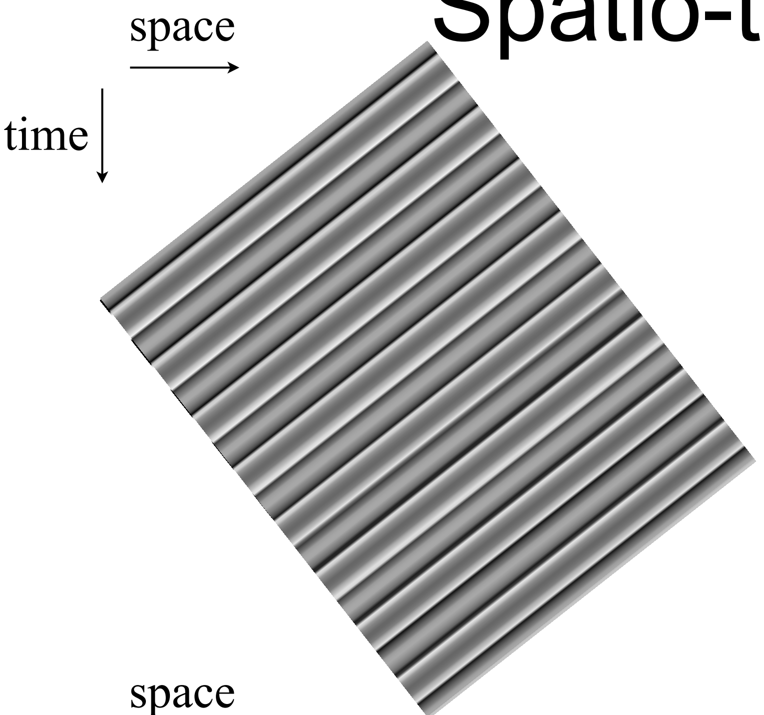
http://en.wikipedia.org/wiki/Square_wave

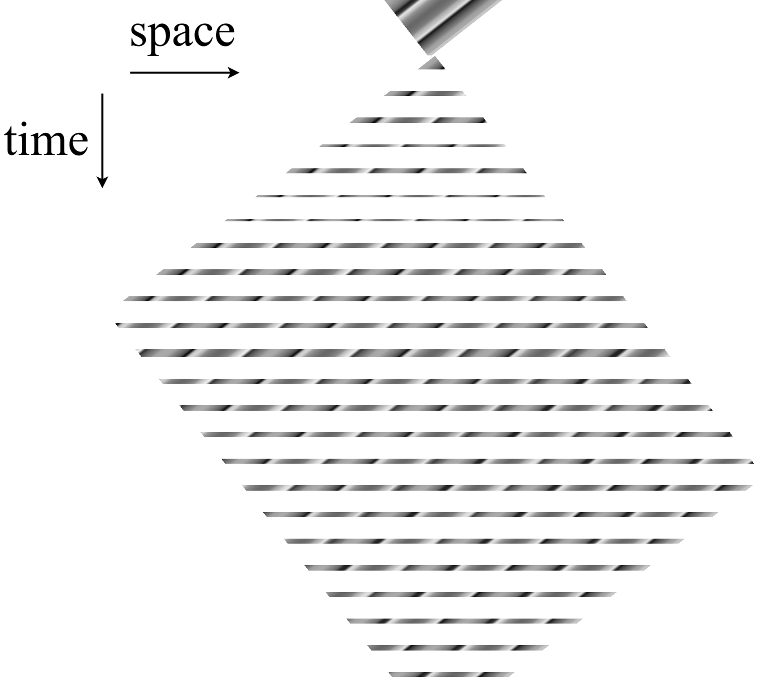

1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.5
1.0
1.5
2.0
1.0
0.8
0.6
0.4
0.2
0.0
−80
−60
−40
−20
0
20
40
60
80

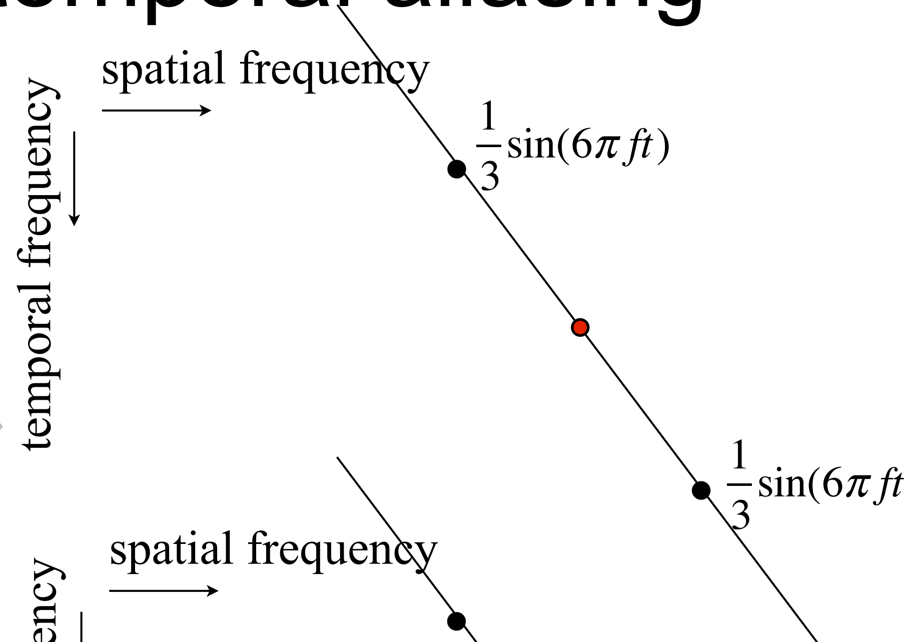
Spatio-temporal aliasing

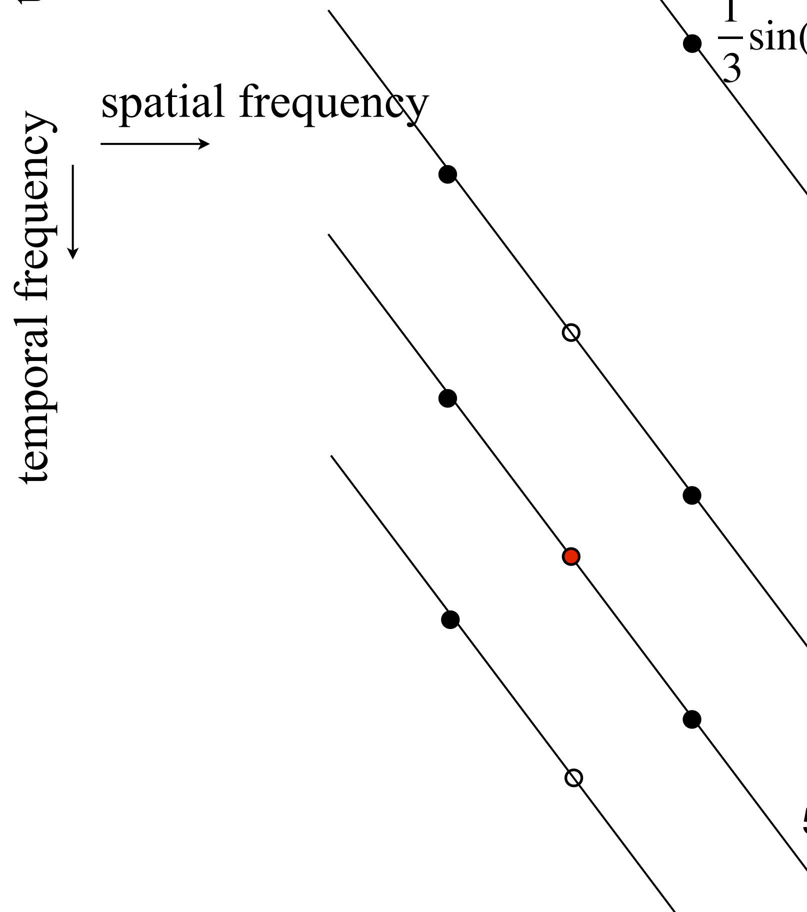


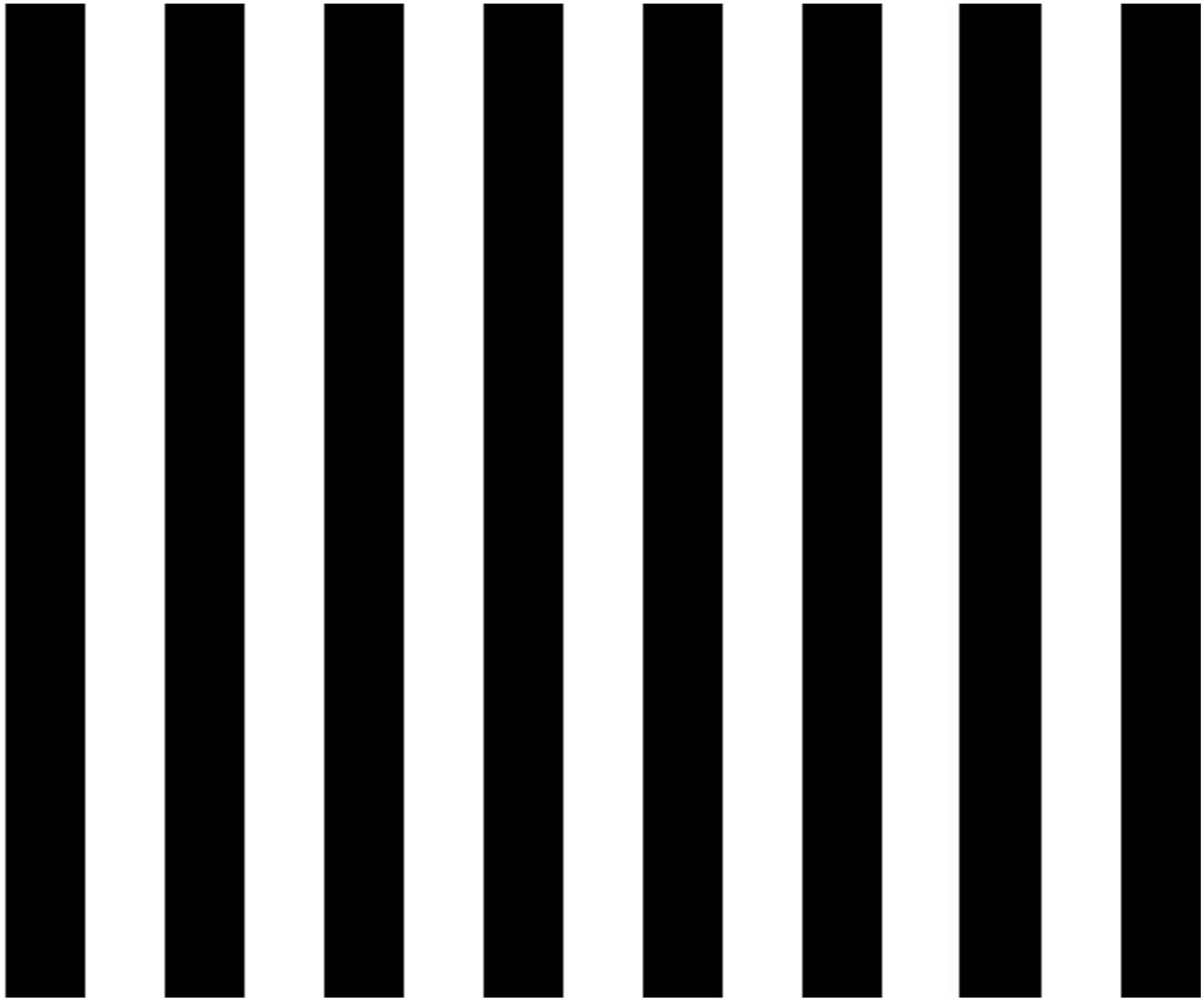
Spatio-temporal aliasing

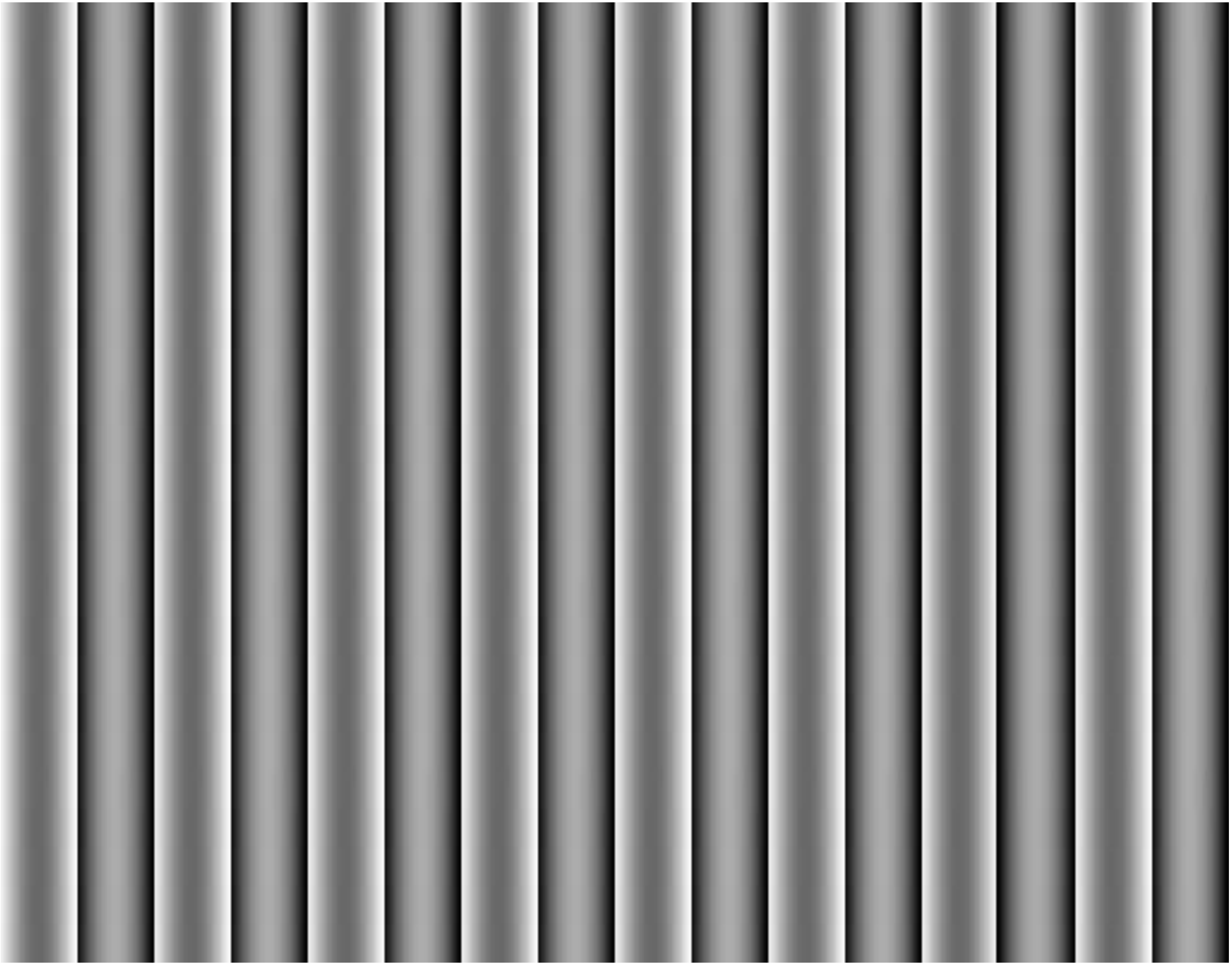












end