6.869 Advances in Computer Vision

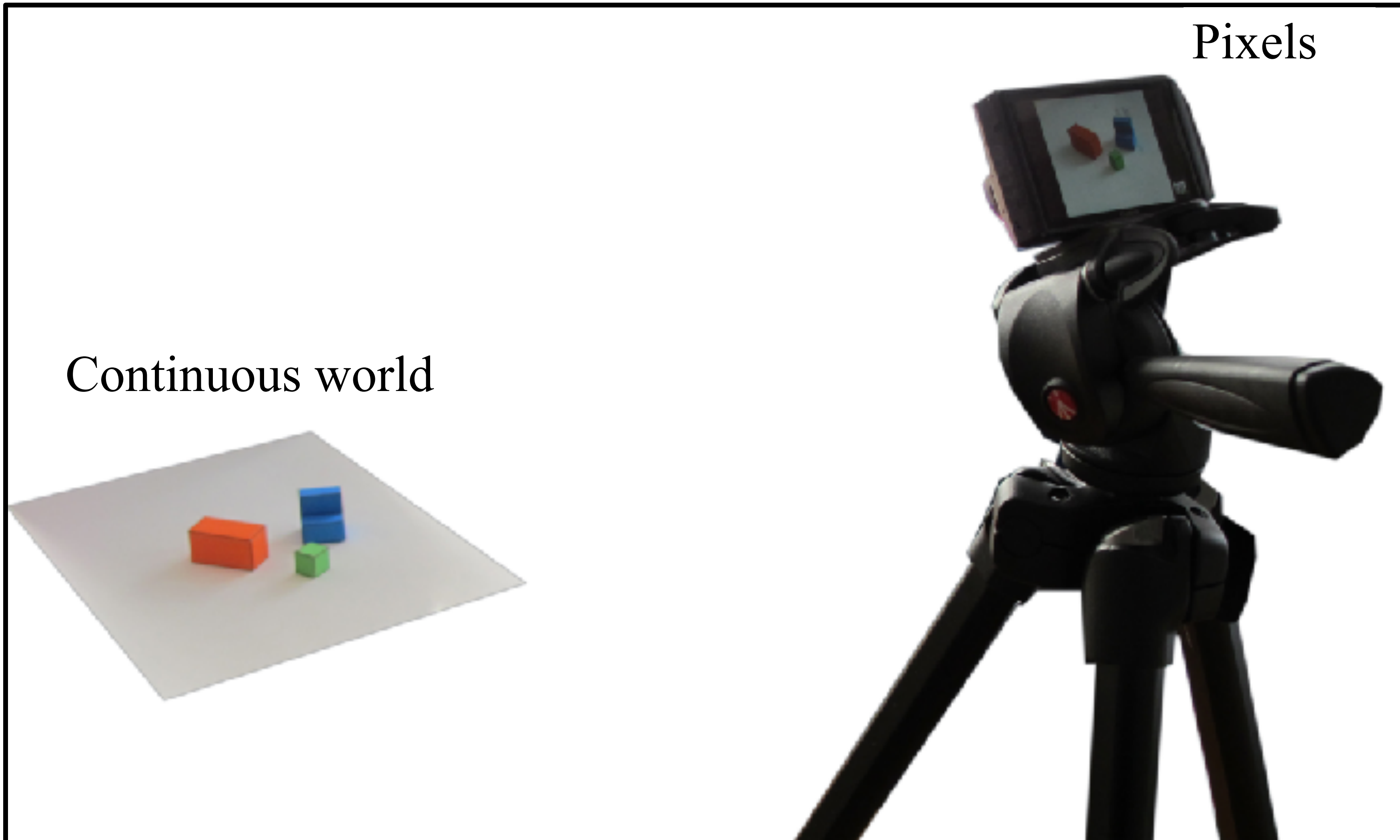
Bill Freeman, Antonio Torralba and Phillip Isola

MIT

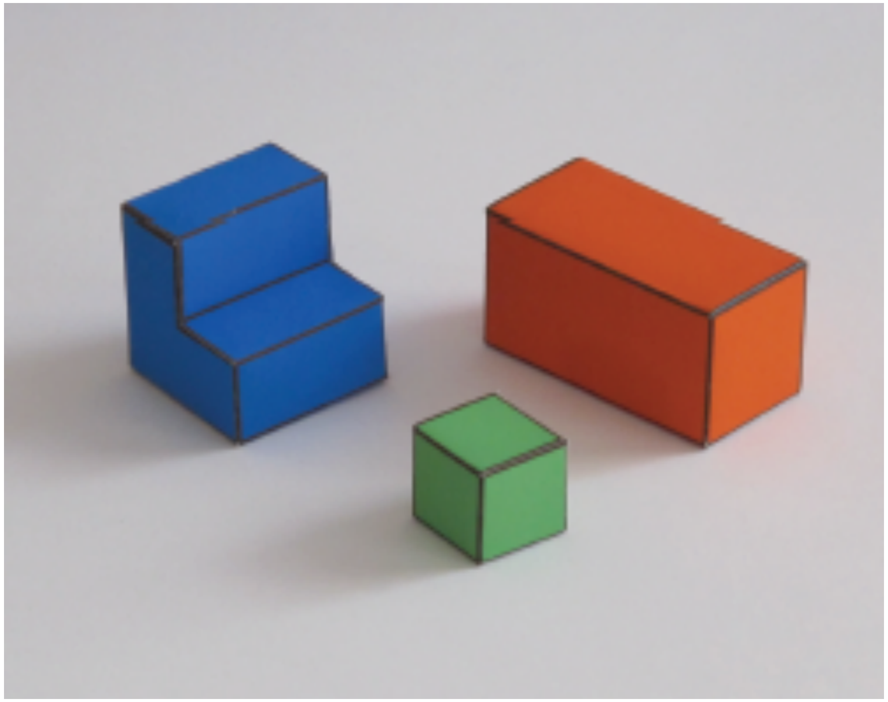
Oct. 3, 2018

Sampling

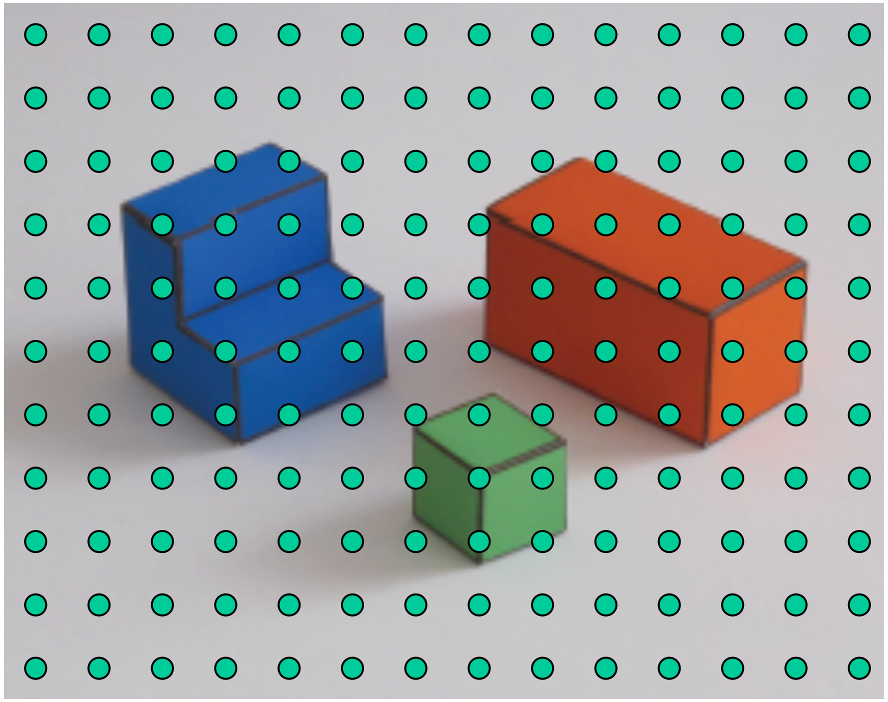
Sampling



Sampling



Sampling

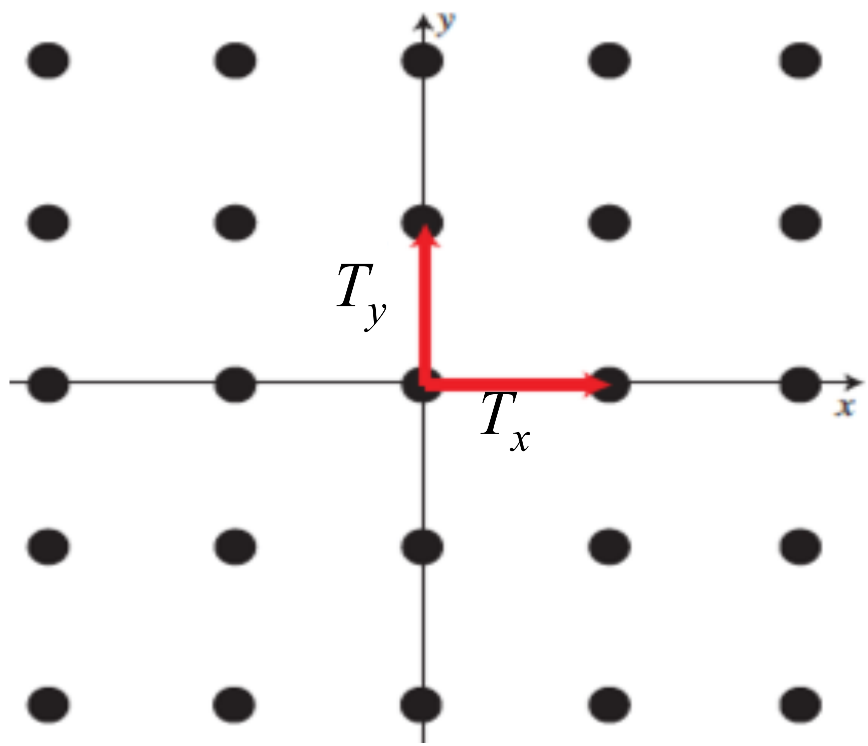


Sampling

Continuous image $f(x, y)$

We can sample it using a rectangular grid as

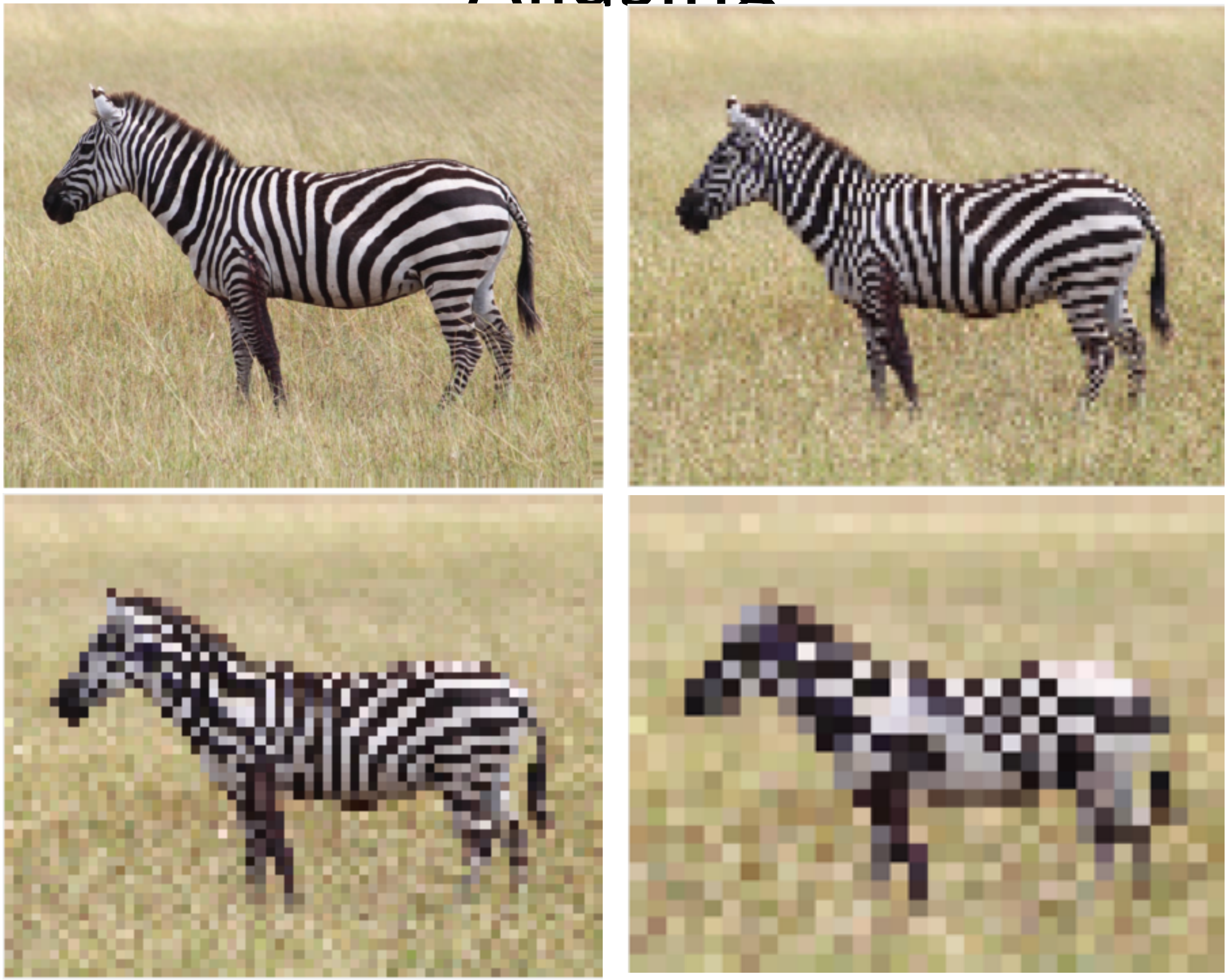
$$f[n, m] = f\left(nT_x, mT_y\right)$$



Aliasing

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

Aliasing



103×128

52×64

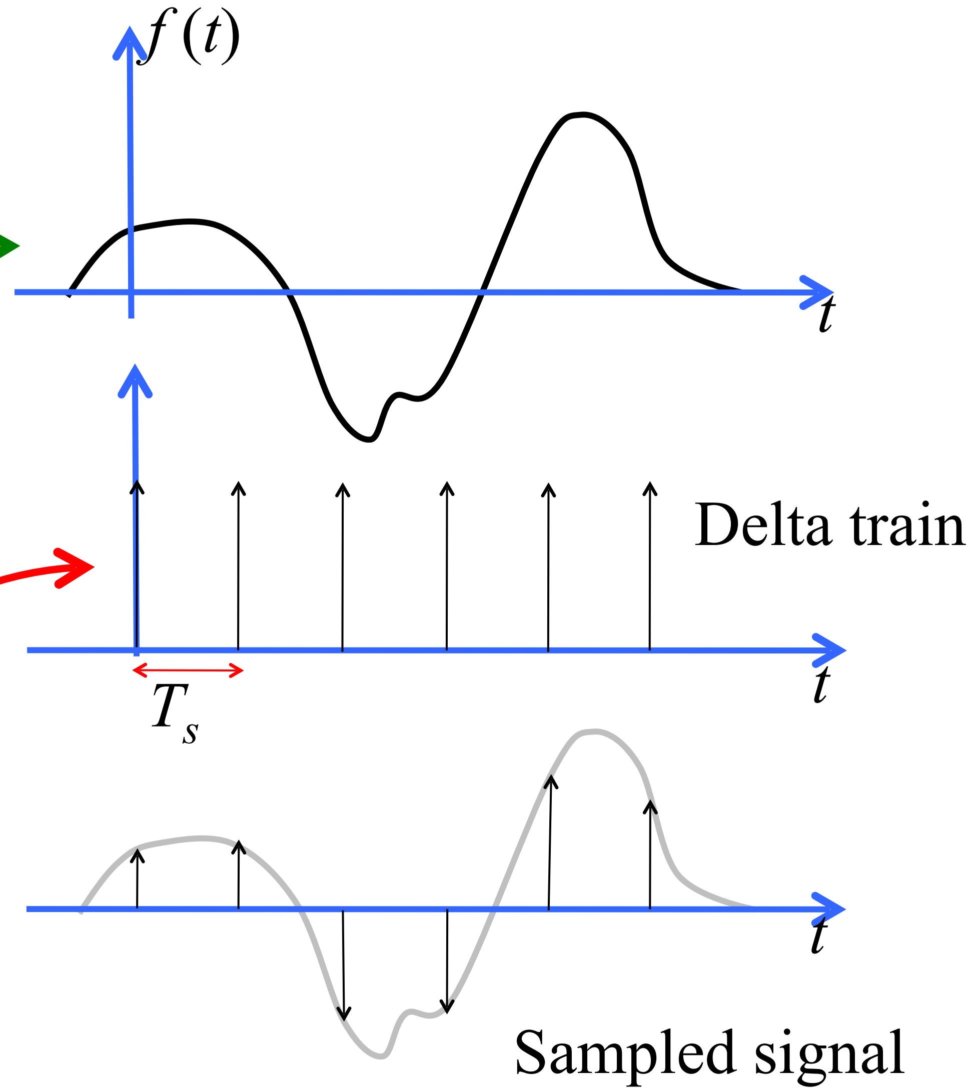
26×32

Modeling the sampling process

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

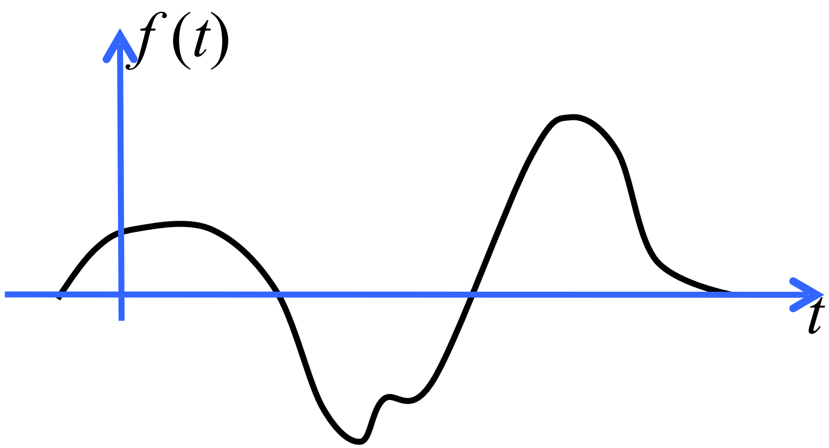
A convenient writing:

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

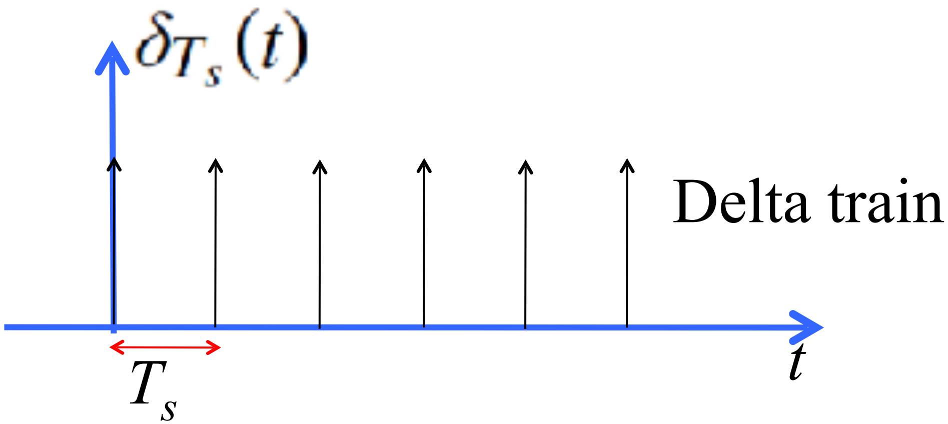


Modeling the sampling process

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



*

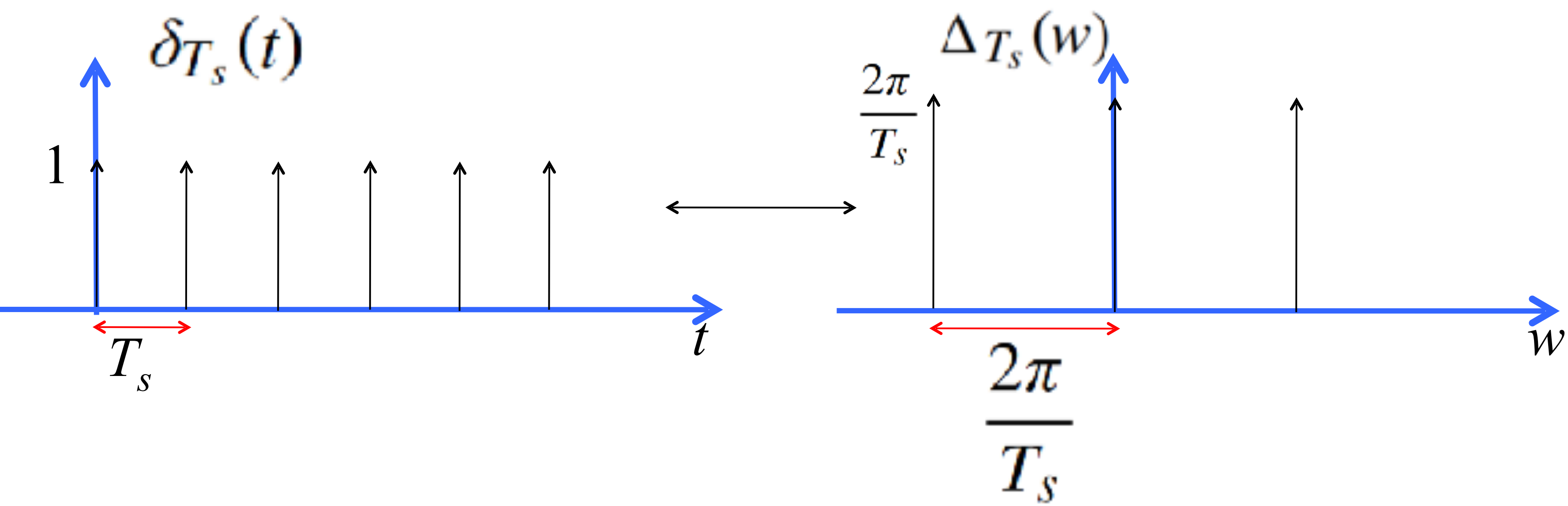


The Fourier transform is a convolution…

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

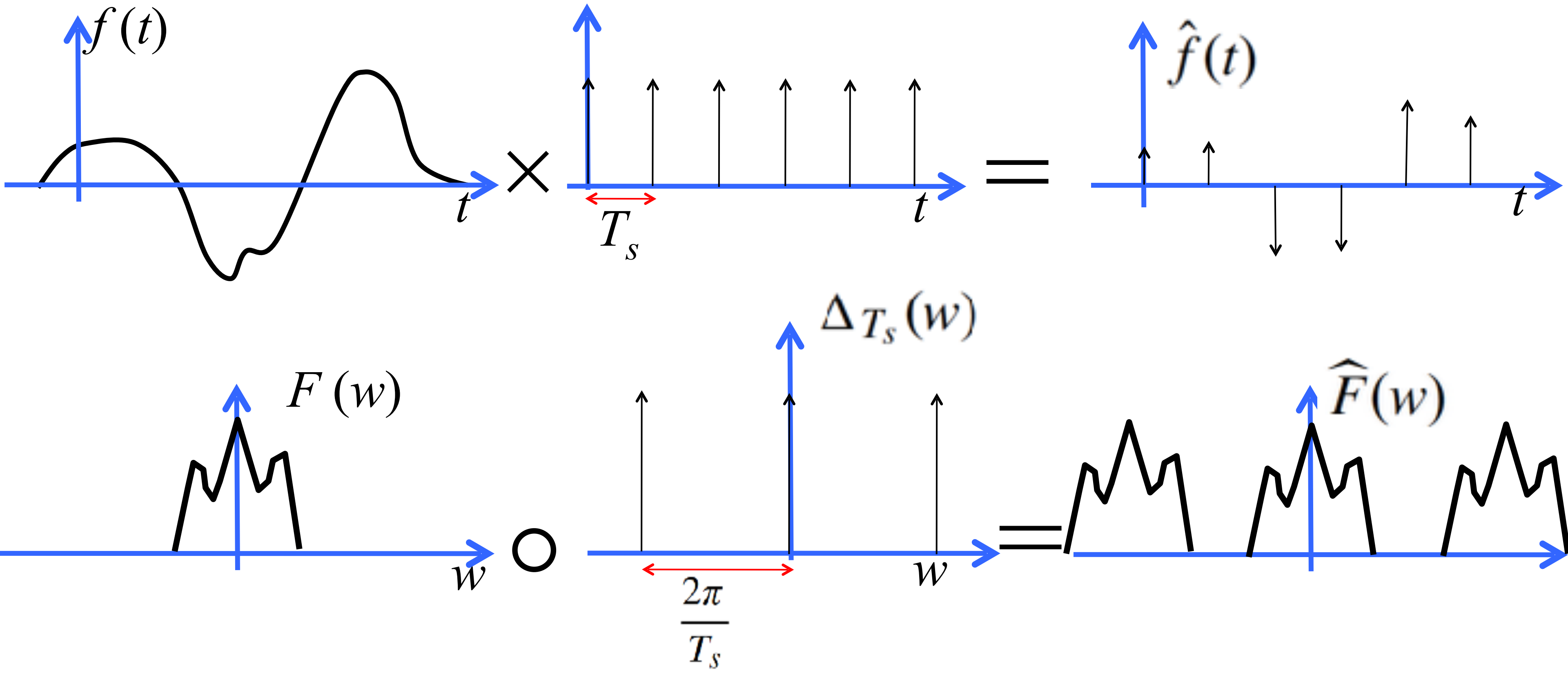
Modeling the sampling process

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



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.

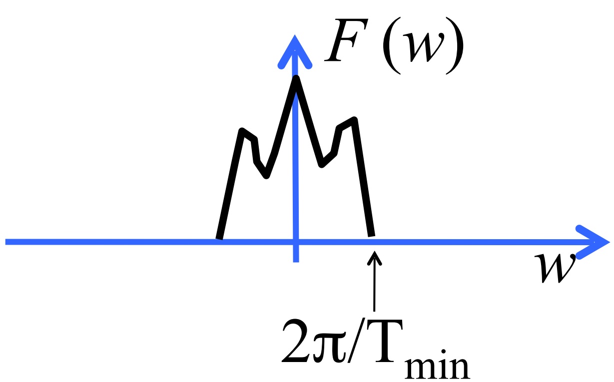
Modeling the sampling process

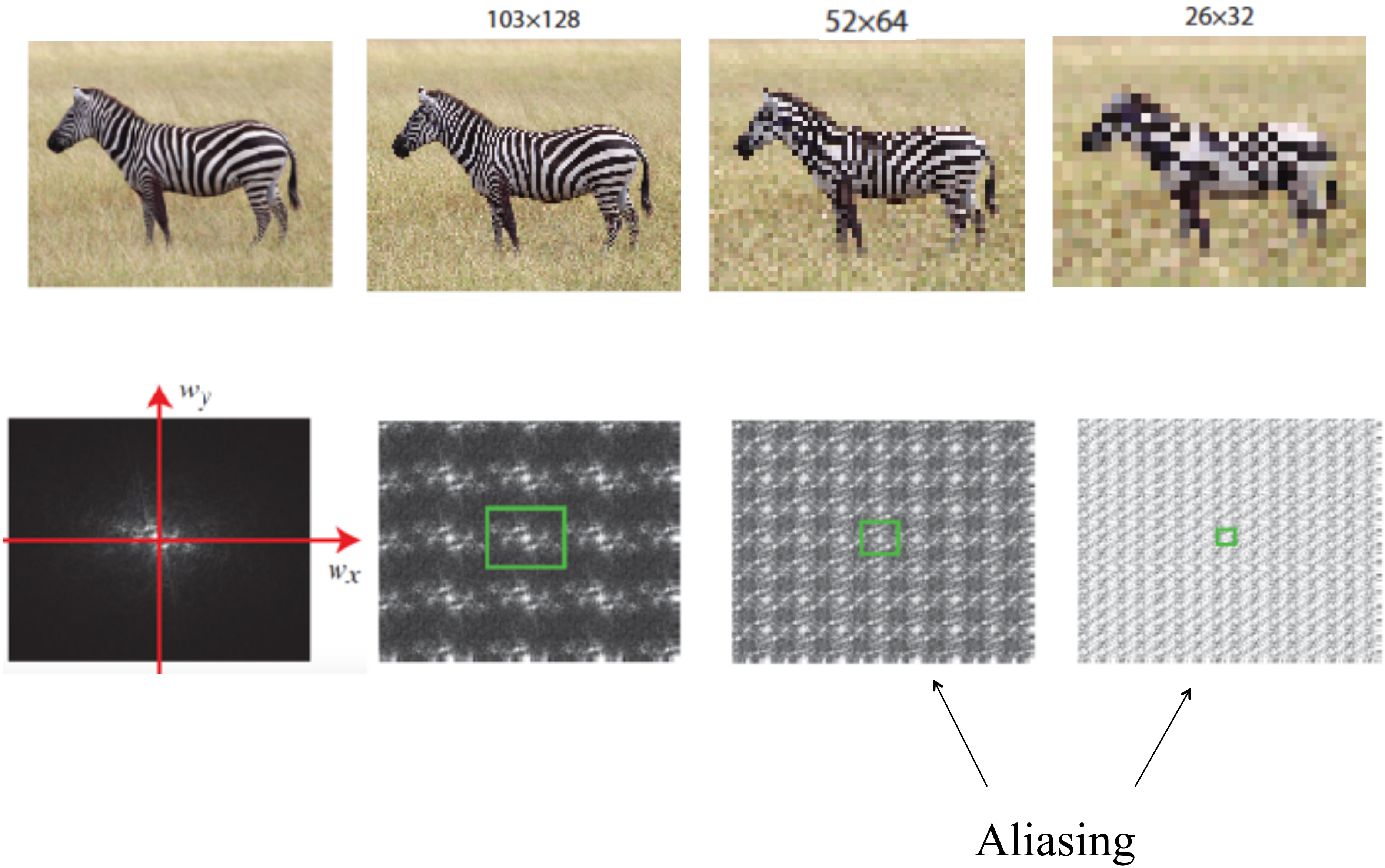


What happens when the repetitions overlap?

Sampling theorem

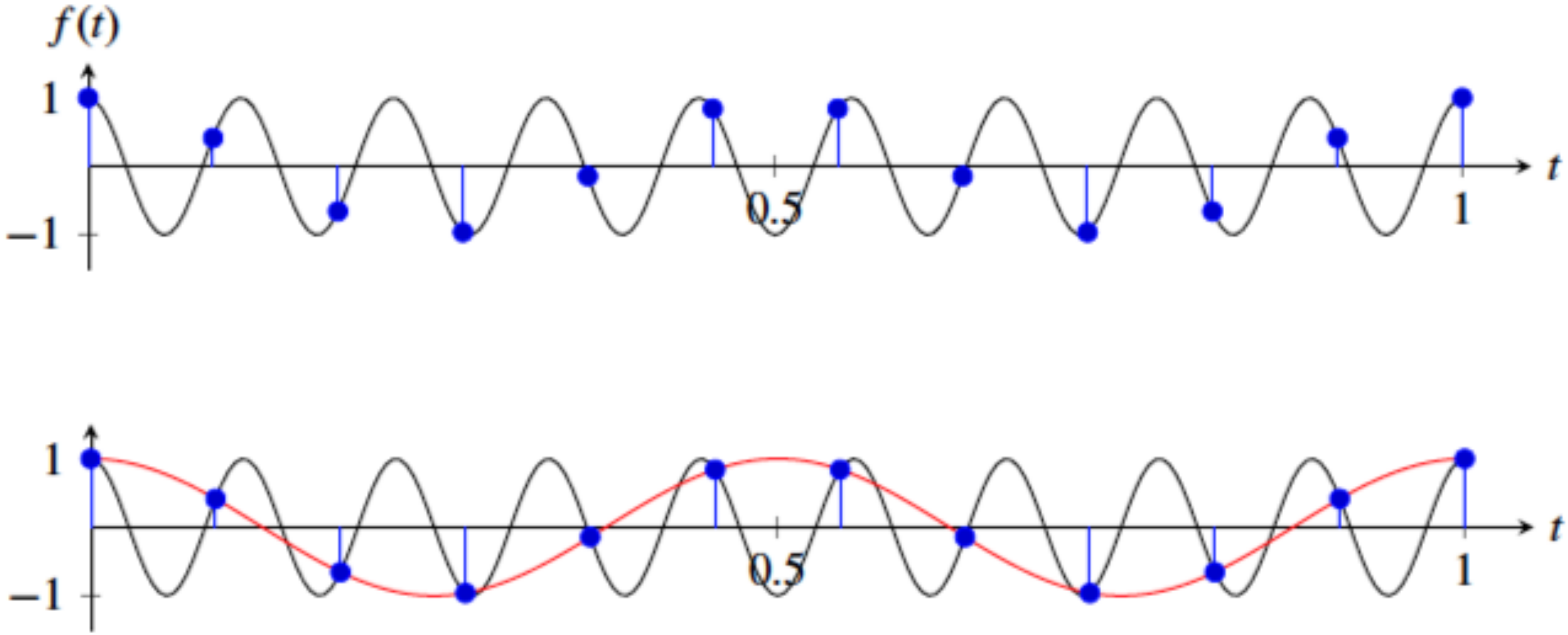
The sampling theorem (also known as Nyquist theorem) states that for a signal to be perfectly reconstructed from it samples, the sampling period T_s has to be $T_s < T_{min}/2$ where T_{min} is the period of the highest frequency present in the input signal.





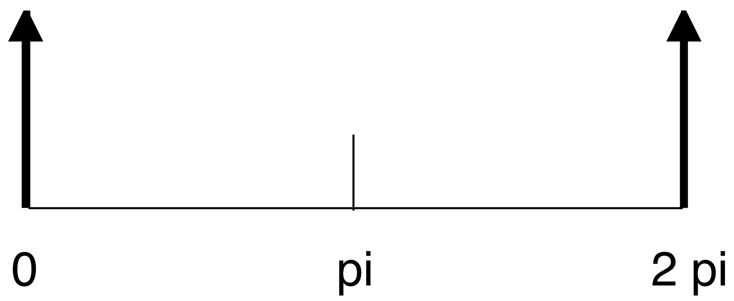
103×128
52×64
26×32
w_y
w_x
Aliasing

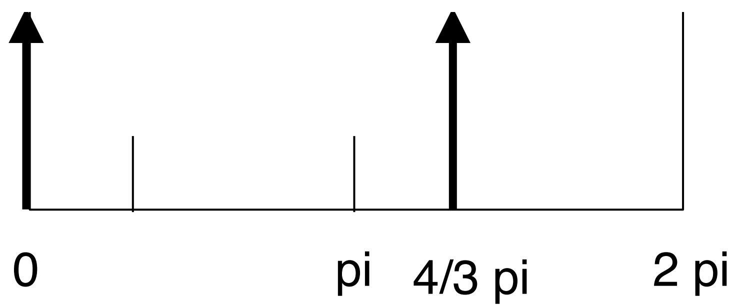
Aliasing



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

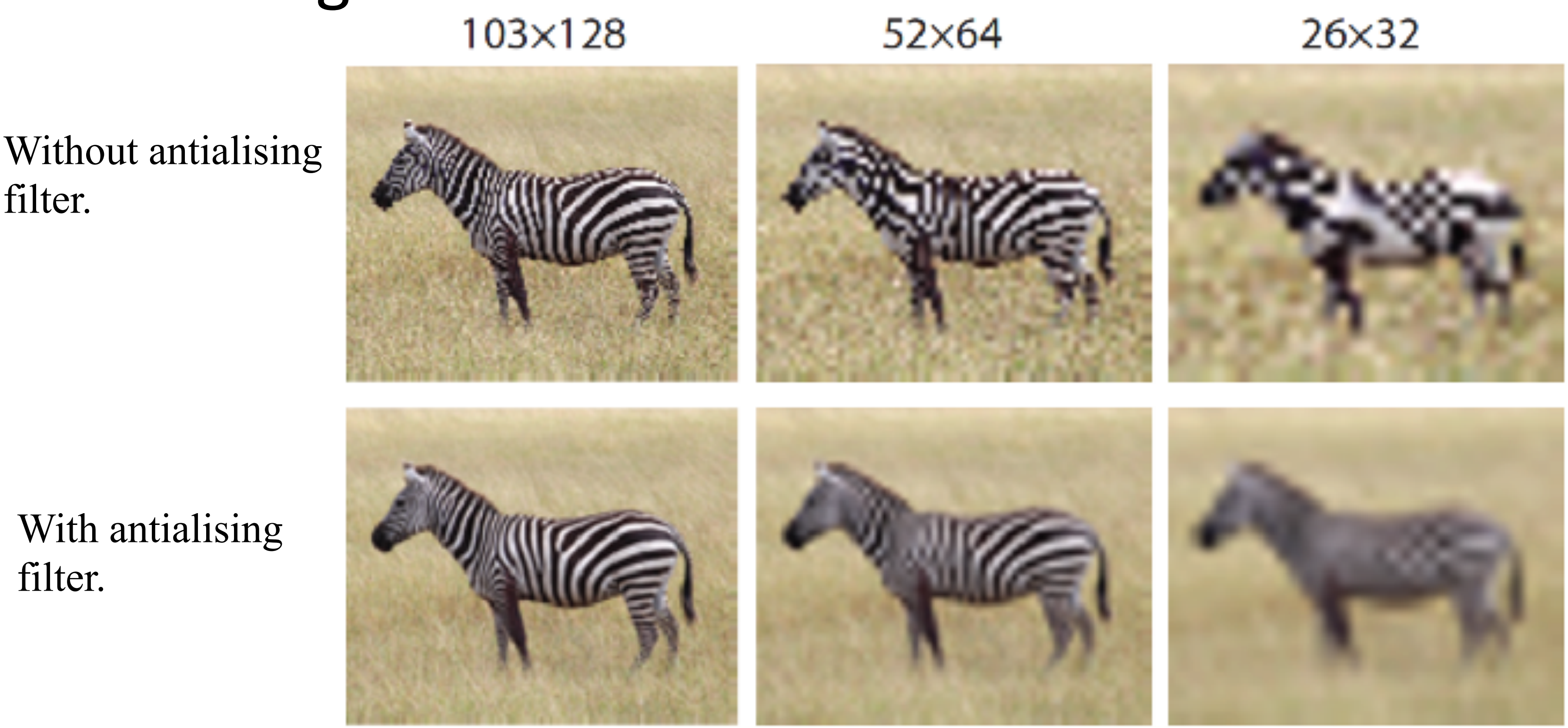
aliasing





Antialising filtering

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



Spatio-temporal sampling illusion

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 patio-temporal filters to measure spatio-temporal energy in order to measure local motion.

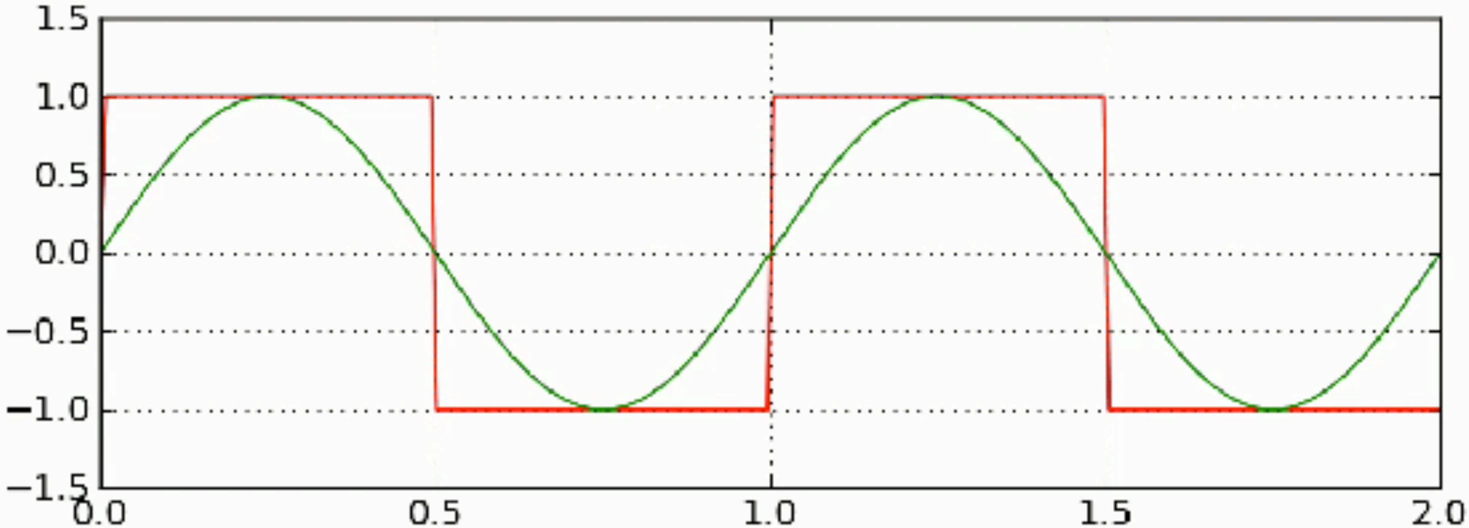
Square wave Fourier components

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

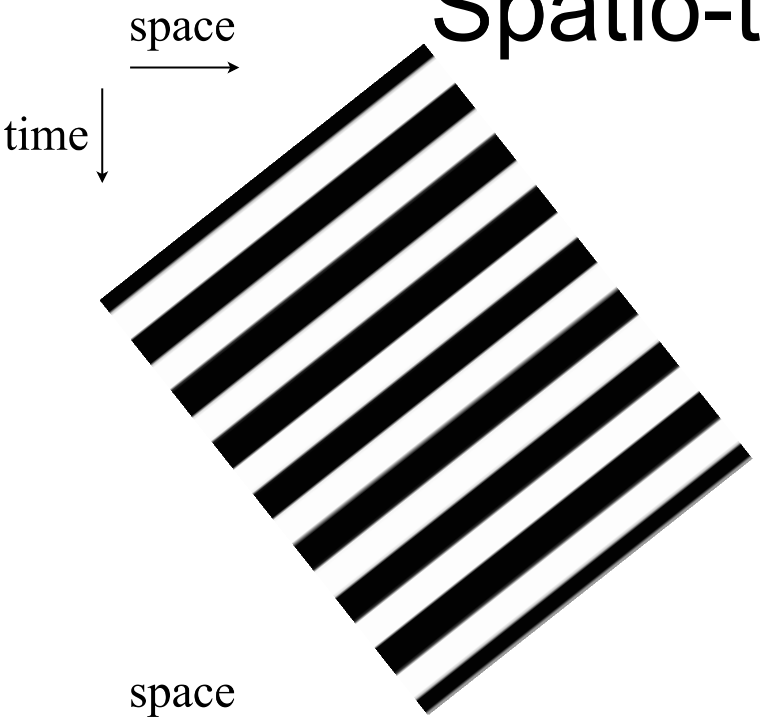
$$x_{\text{square}}(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin\left((2k-1)2\pi ft\right)}{(2k-1)}$$

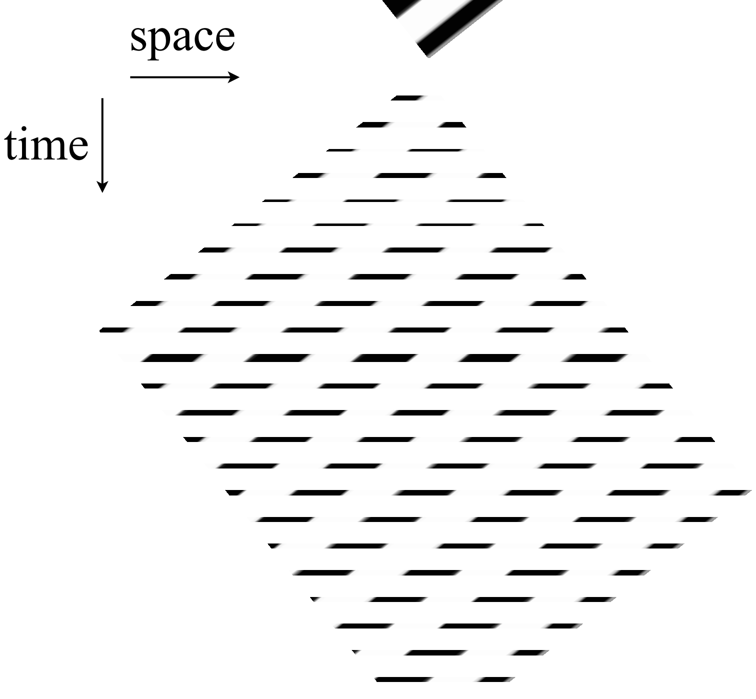
$$= \frac{4}{\pi} \left(\sin(2\pi ft) + \frac{1}{3}\sin(6\pi ft) + \frac{1}{5}\sin(10\pi ft) + \cdots \right).$$

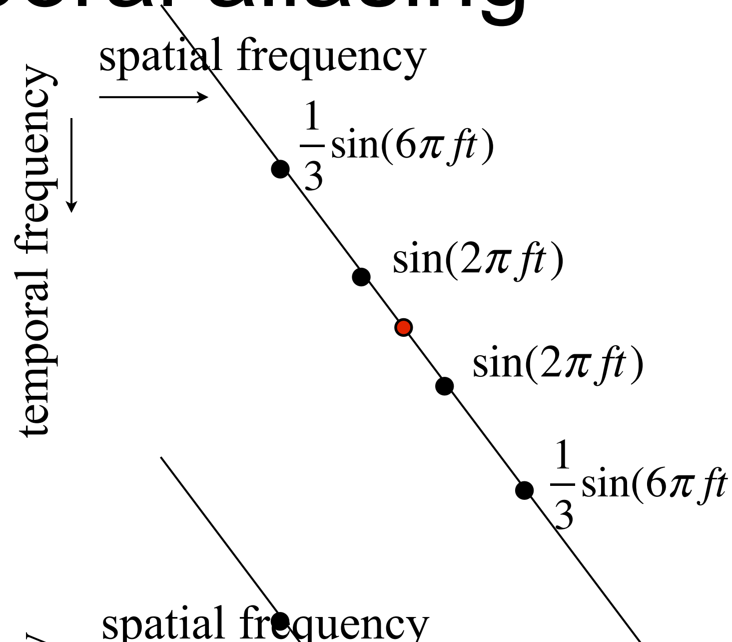
http://en.wikipedia.org/wiki/Square_wave

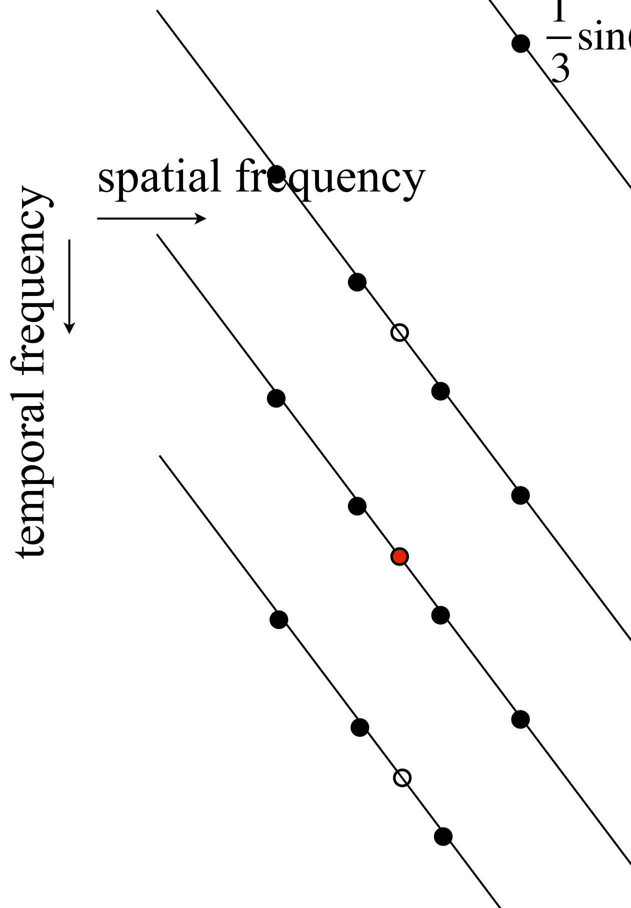

1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.5
1.0
1.5
2.0

Spatio-temporal aliasing

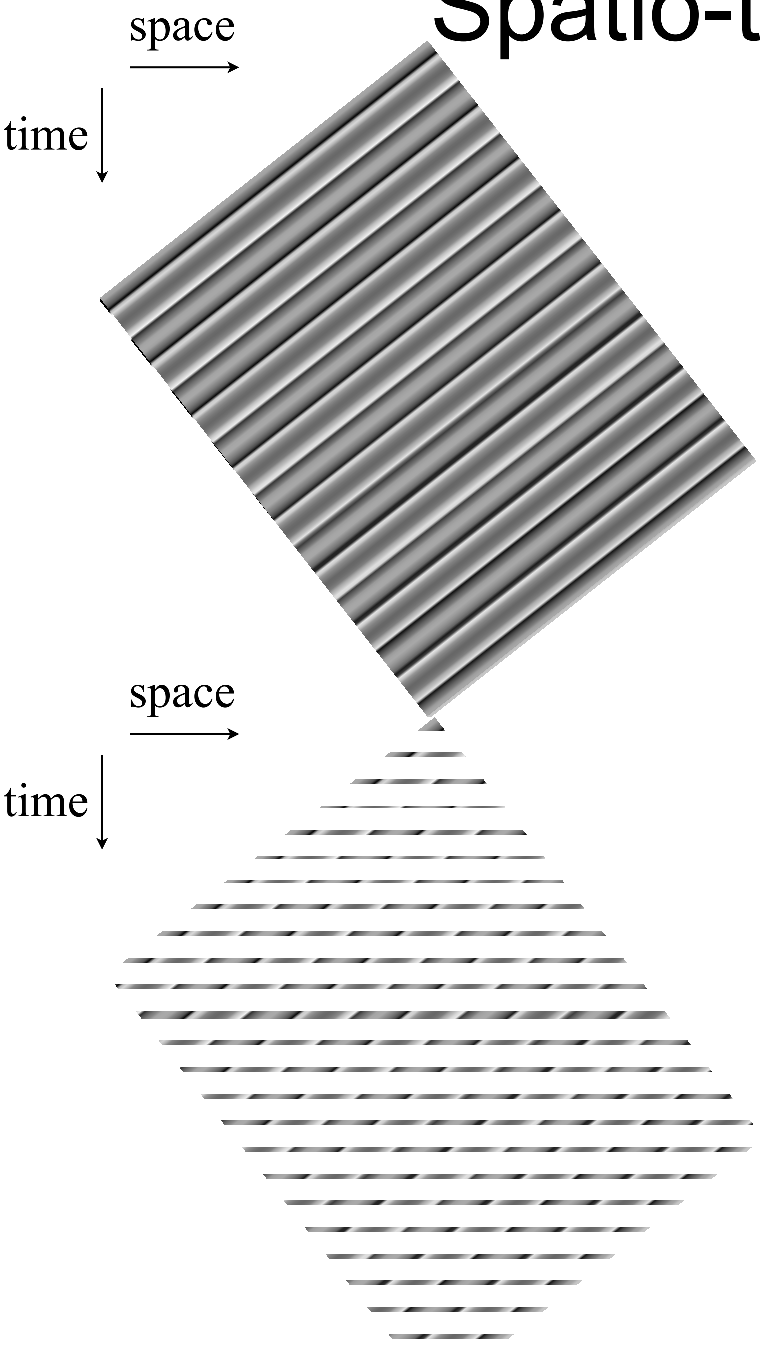


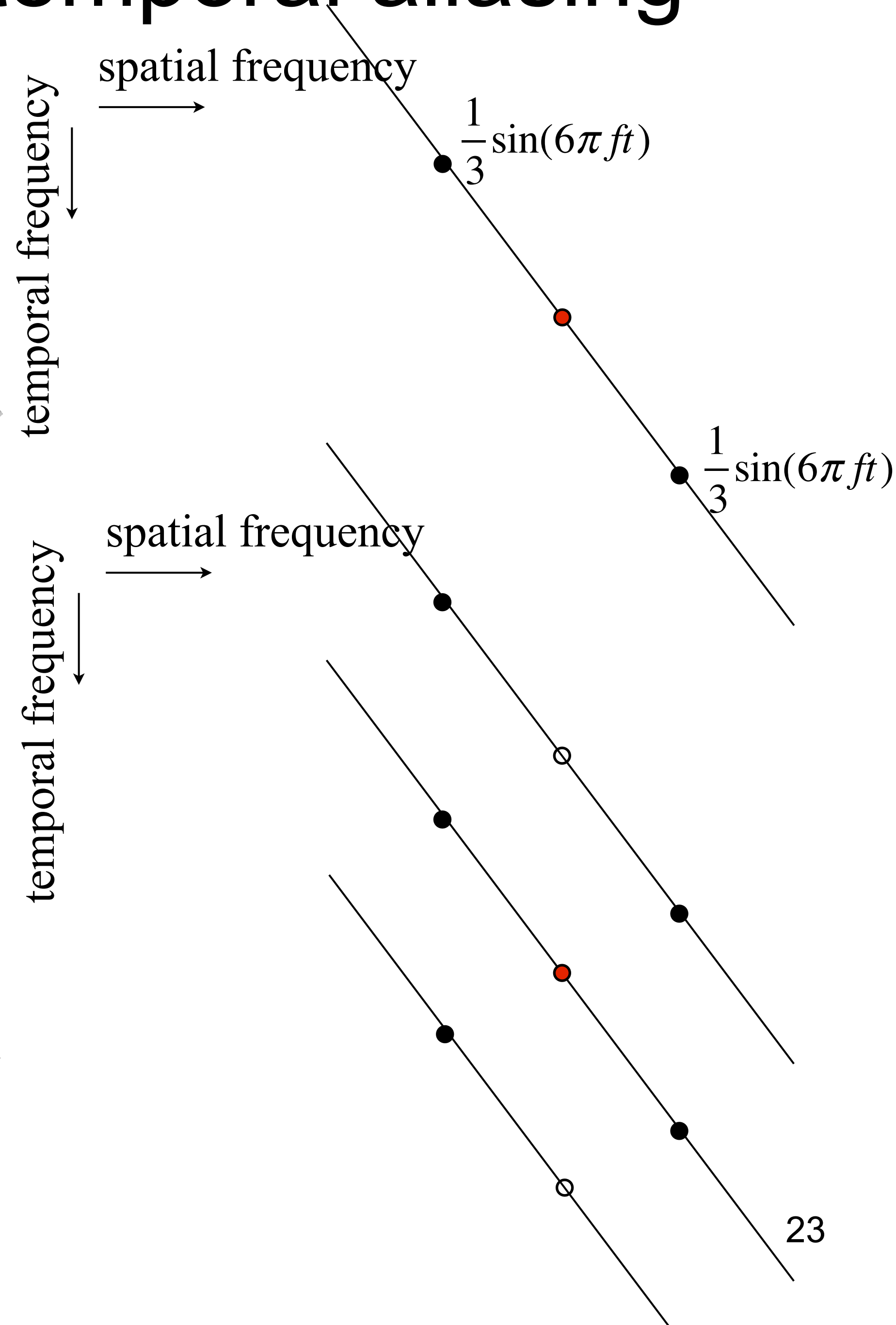


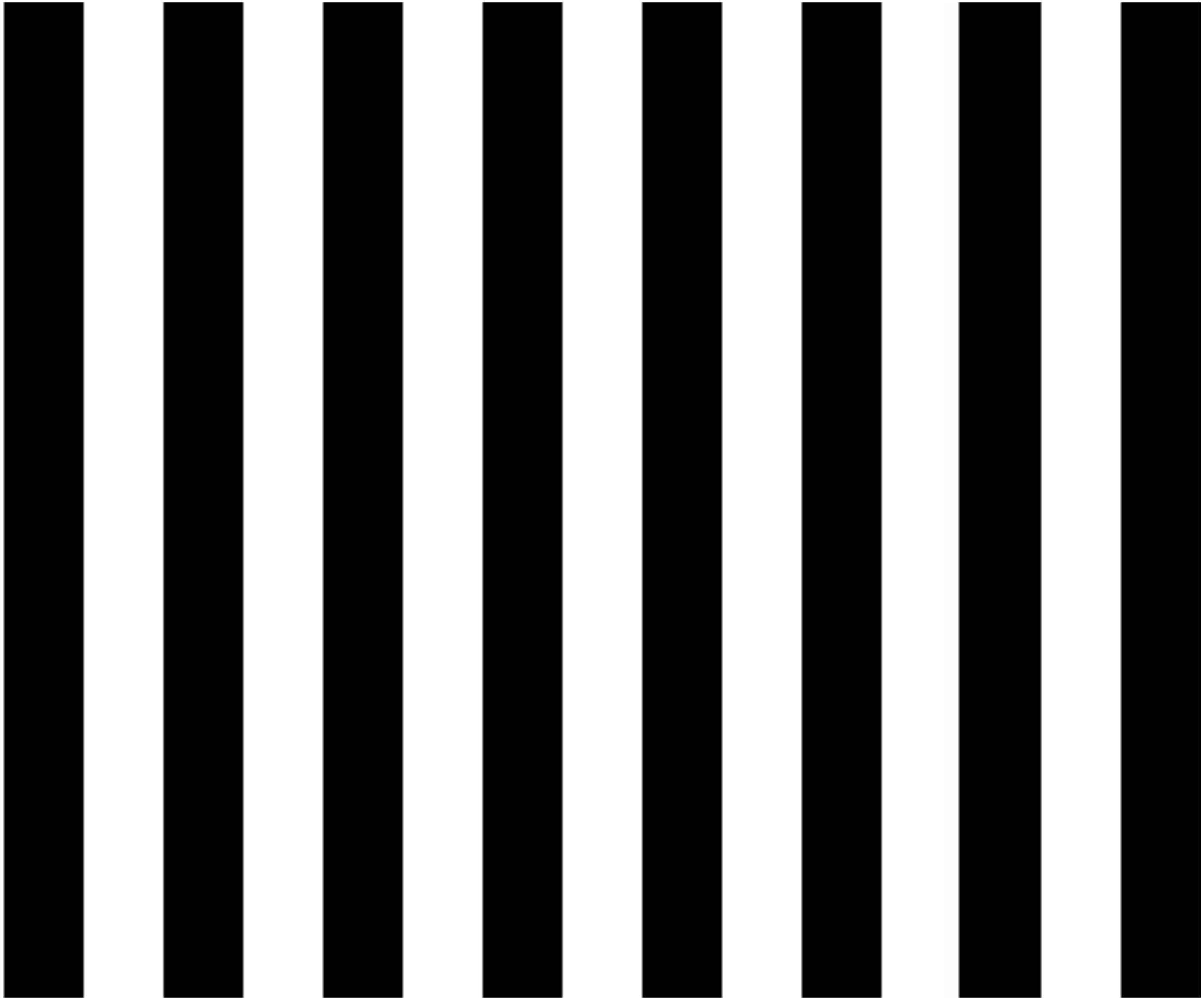


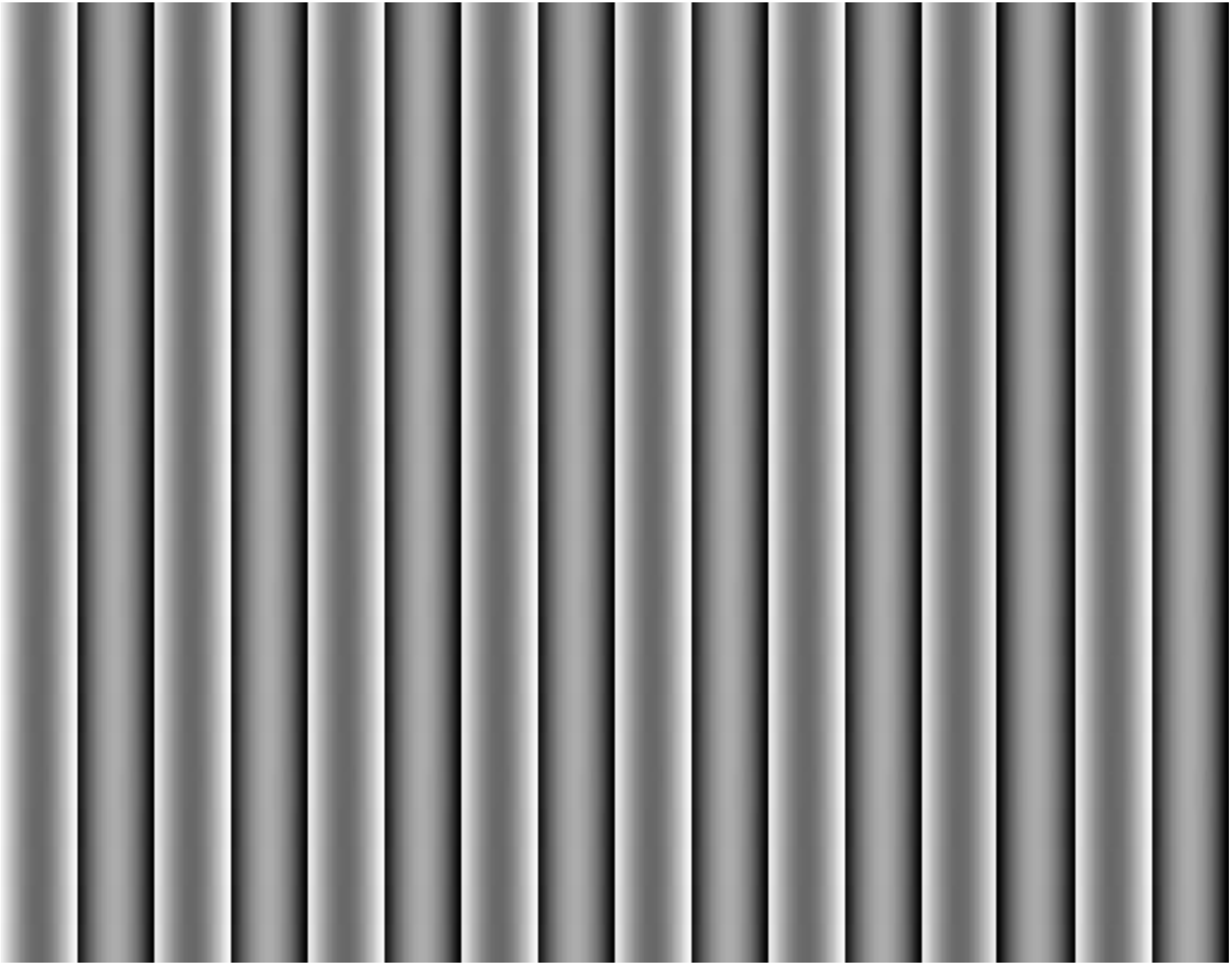


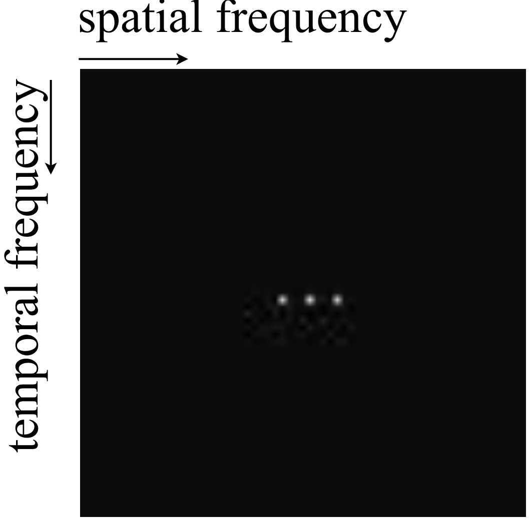
Spatio-temporal aliasing

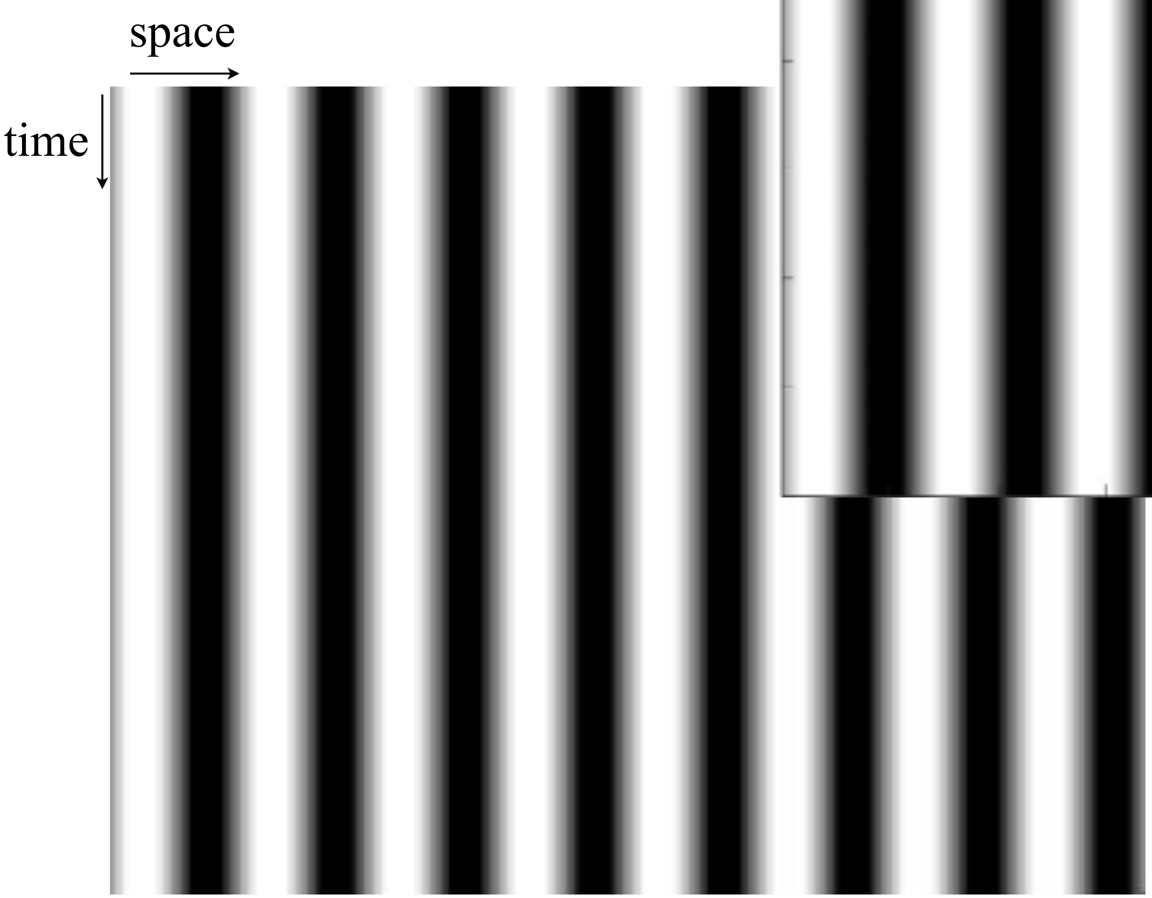




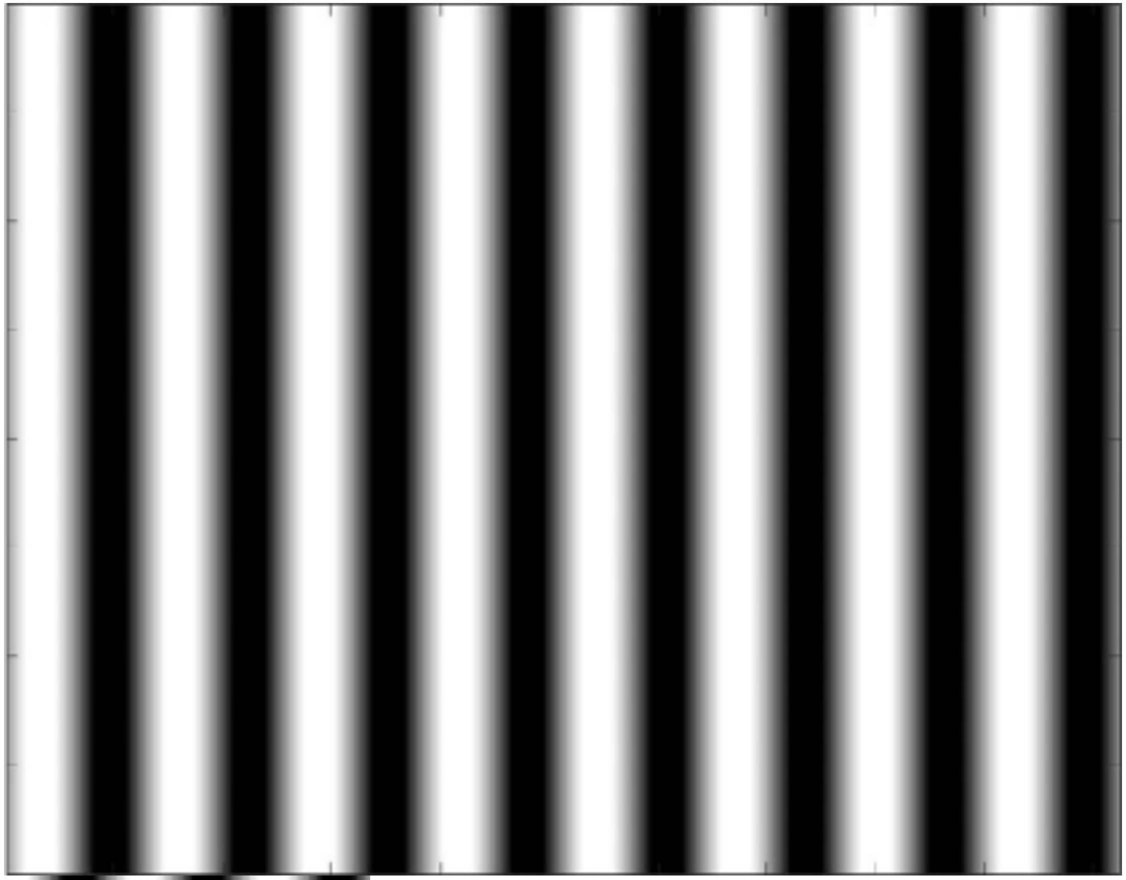


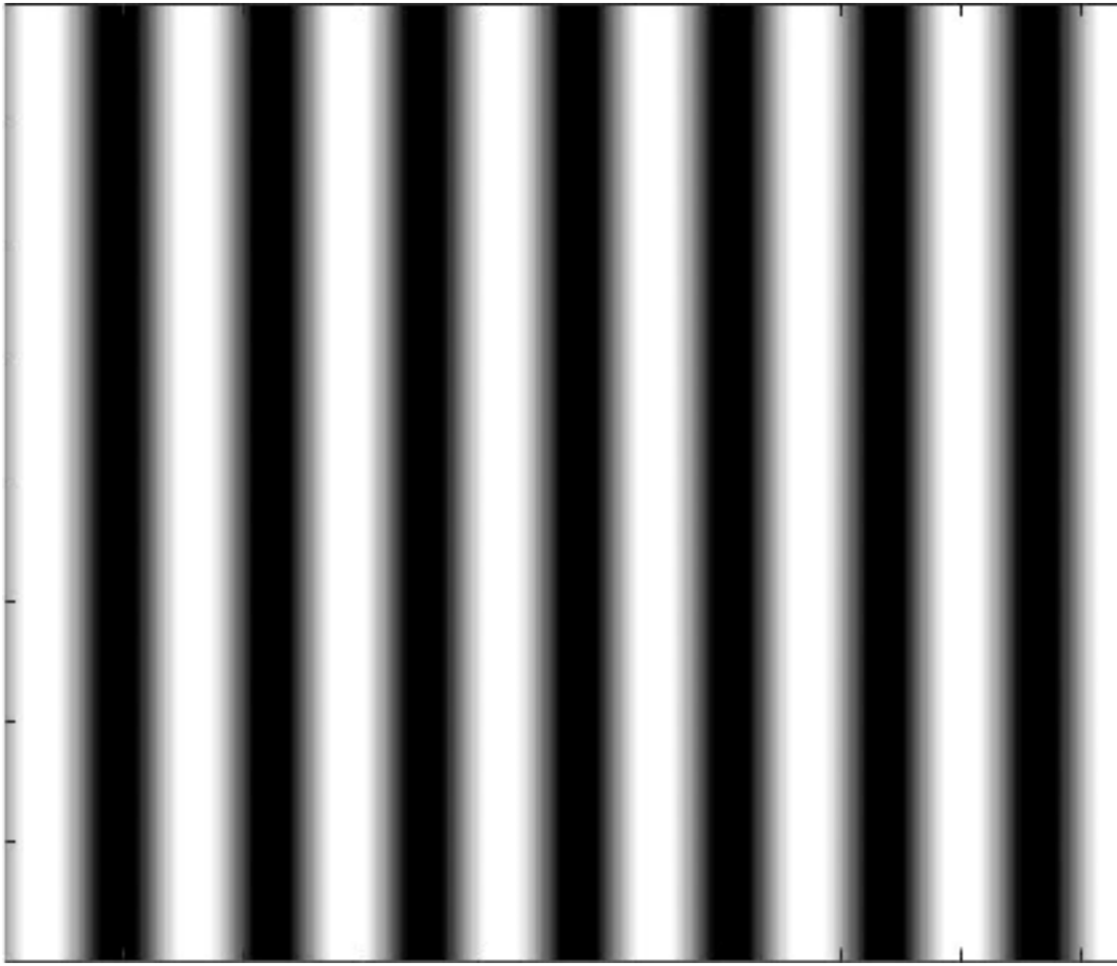


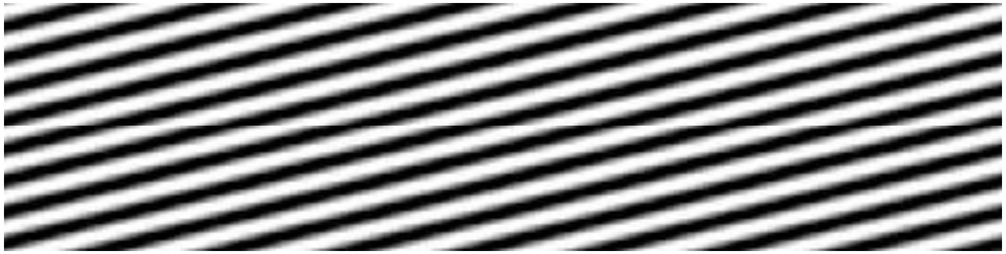




Visual signal





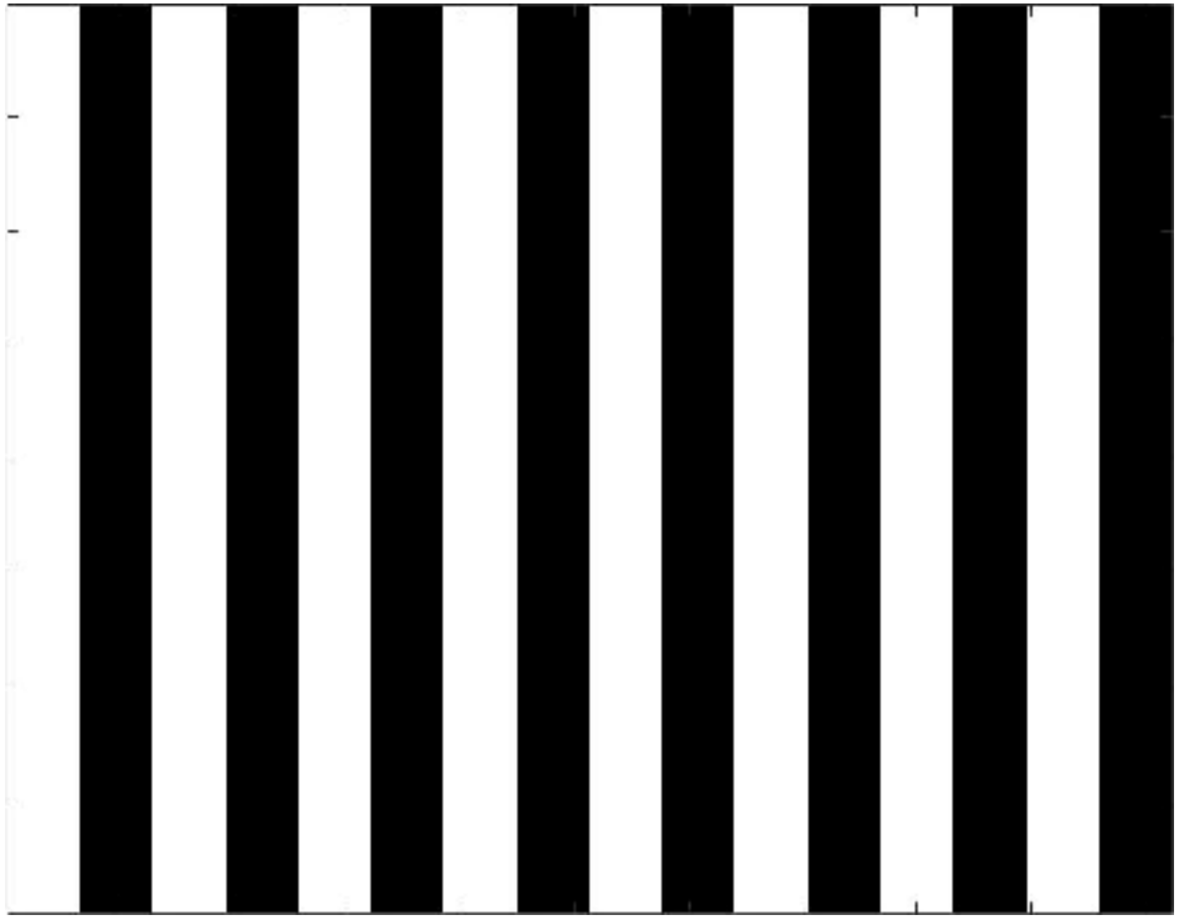


alpha: 1 squareFlag: 0 offset: 4

spatial frequency

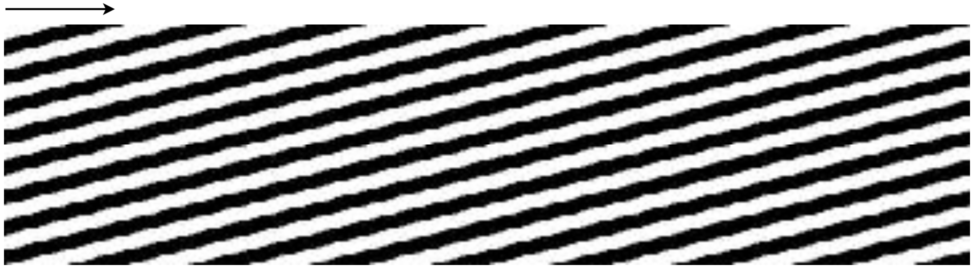
temporal frequency

Visual signal

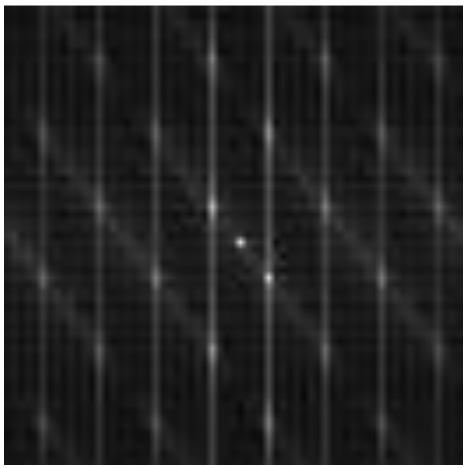


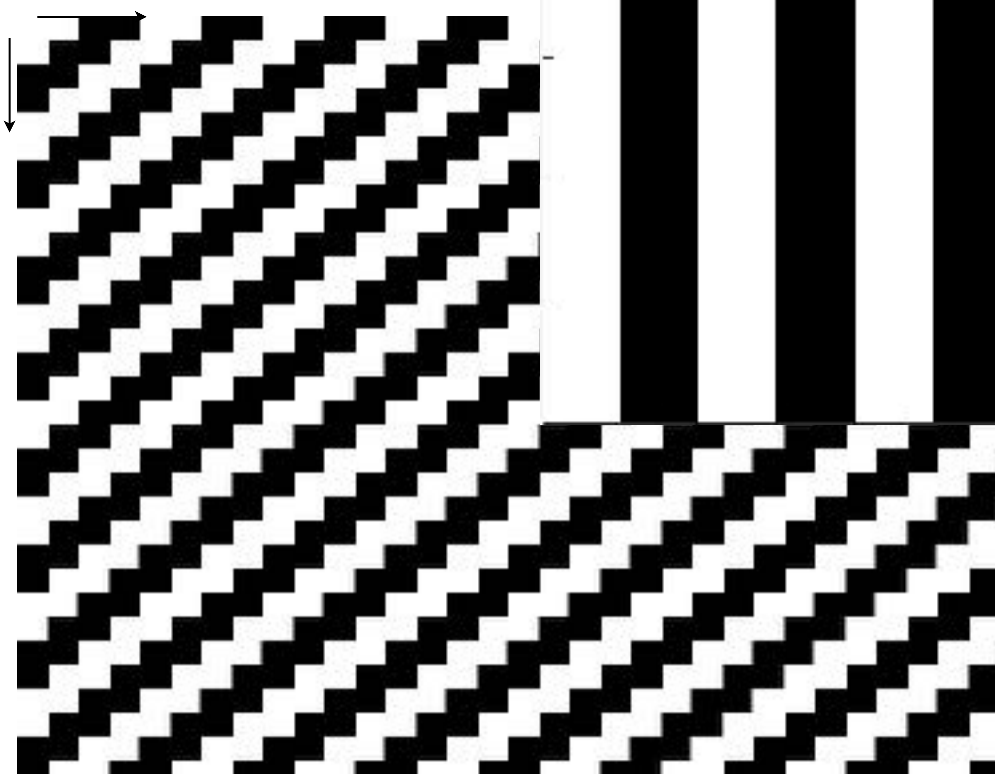
space

time

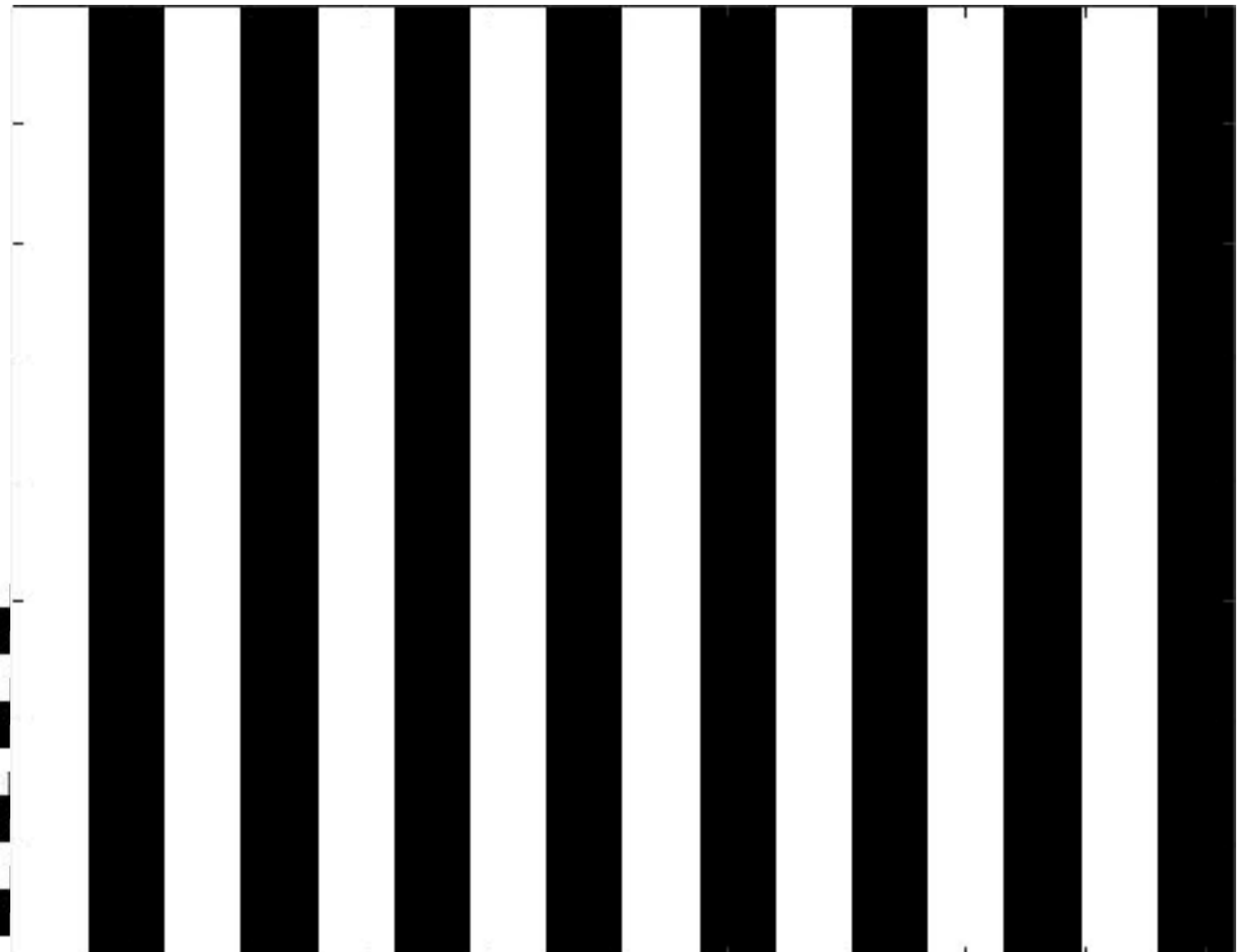


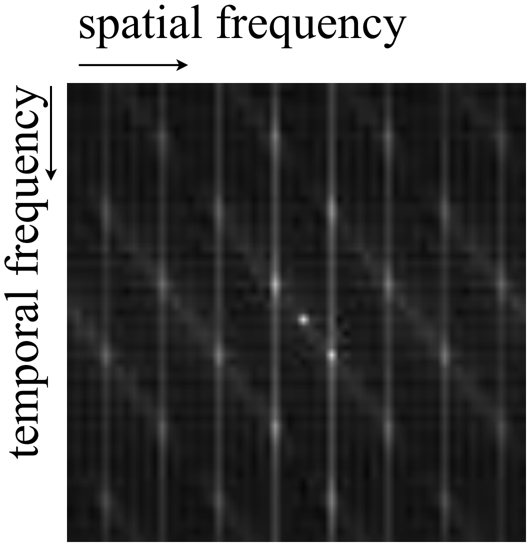
alpha: 1 squareFlag: 1 offset: 4

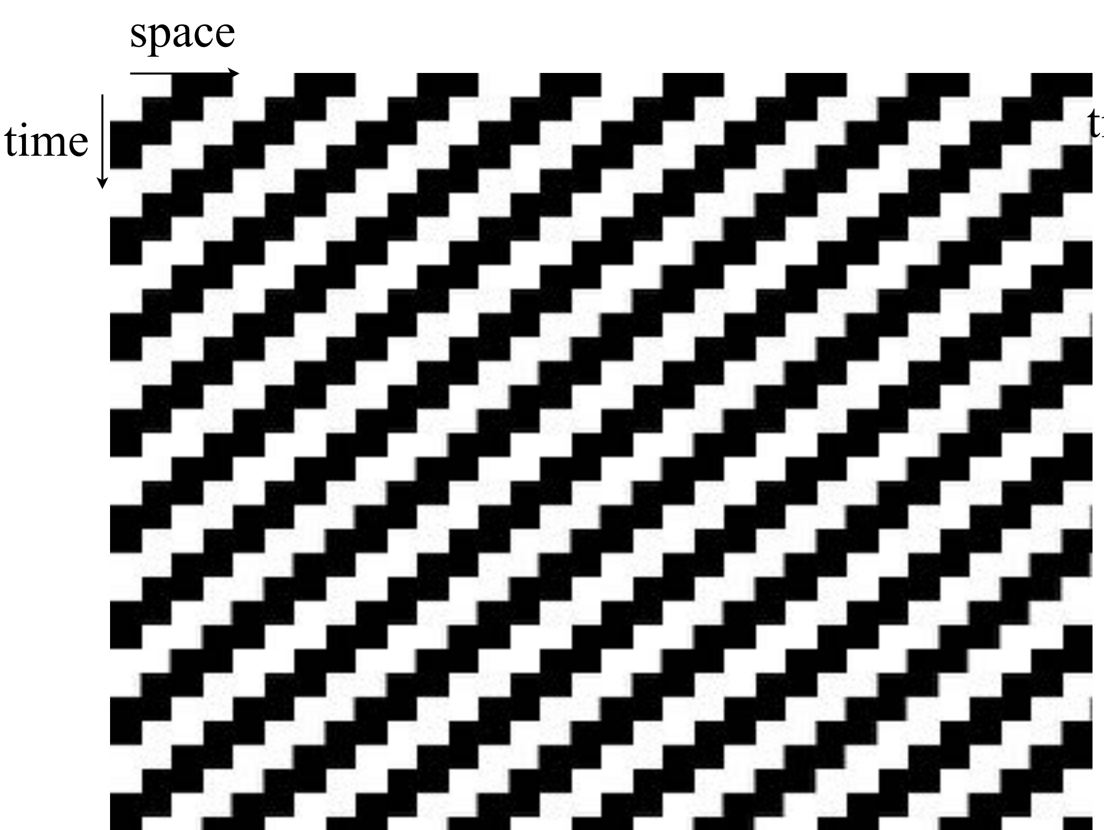


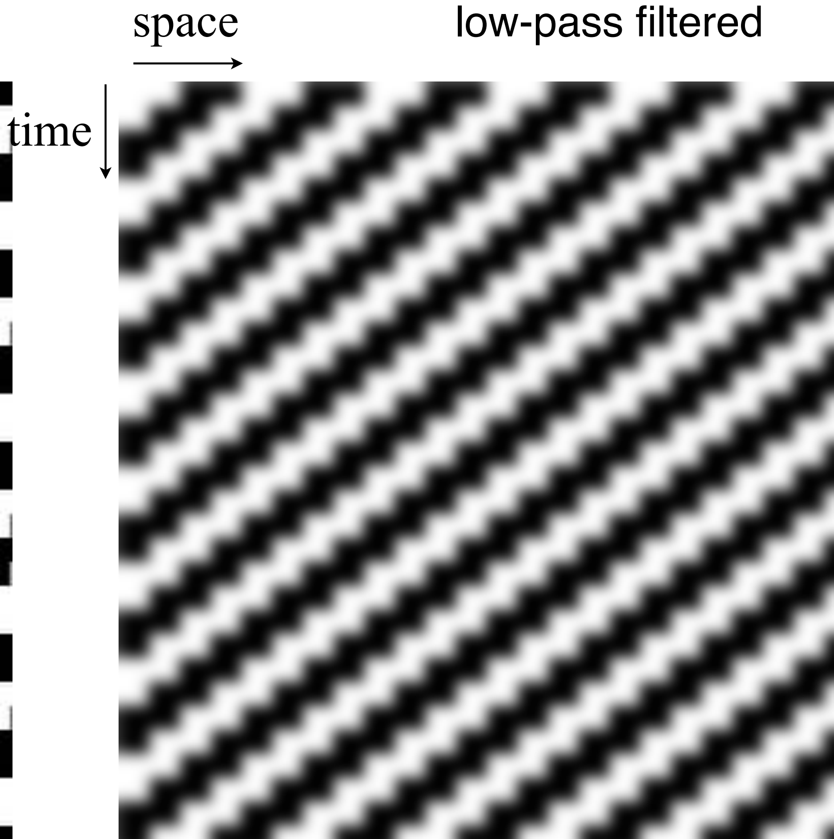


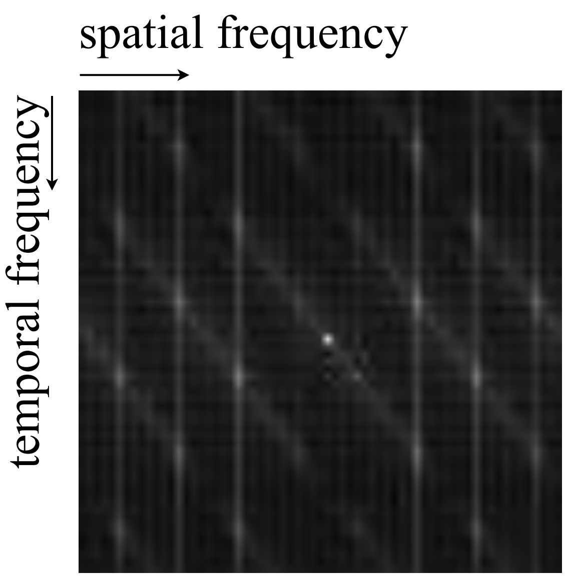
Visual signal



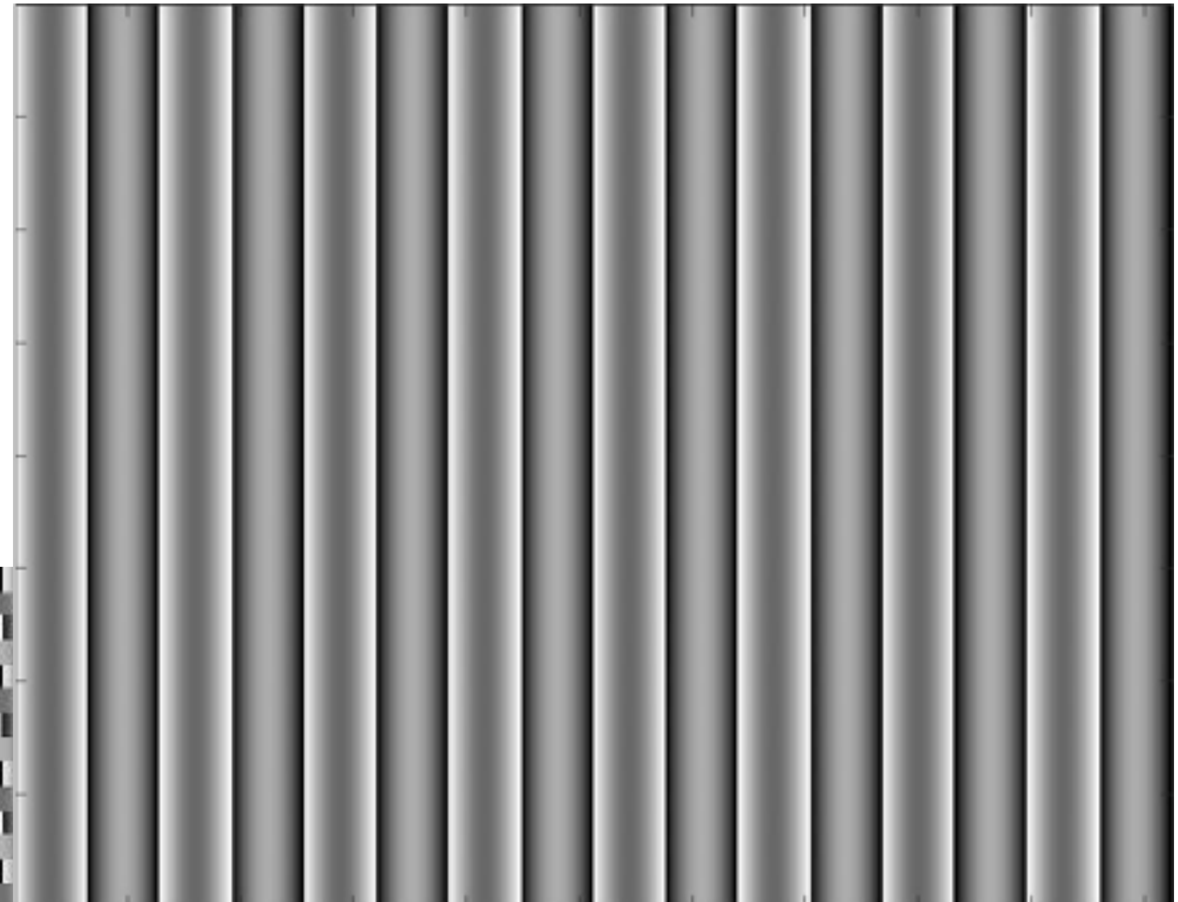

spatial frequency
temporal frequency

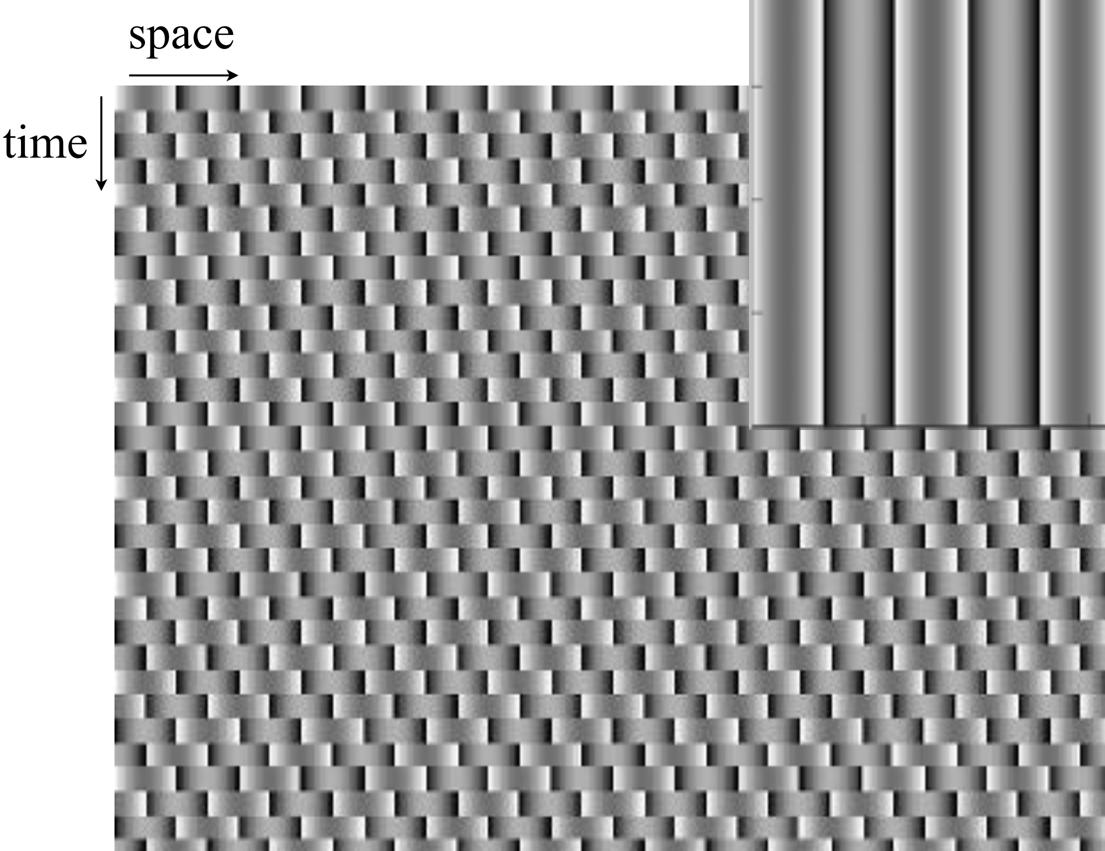

space
time

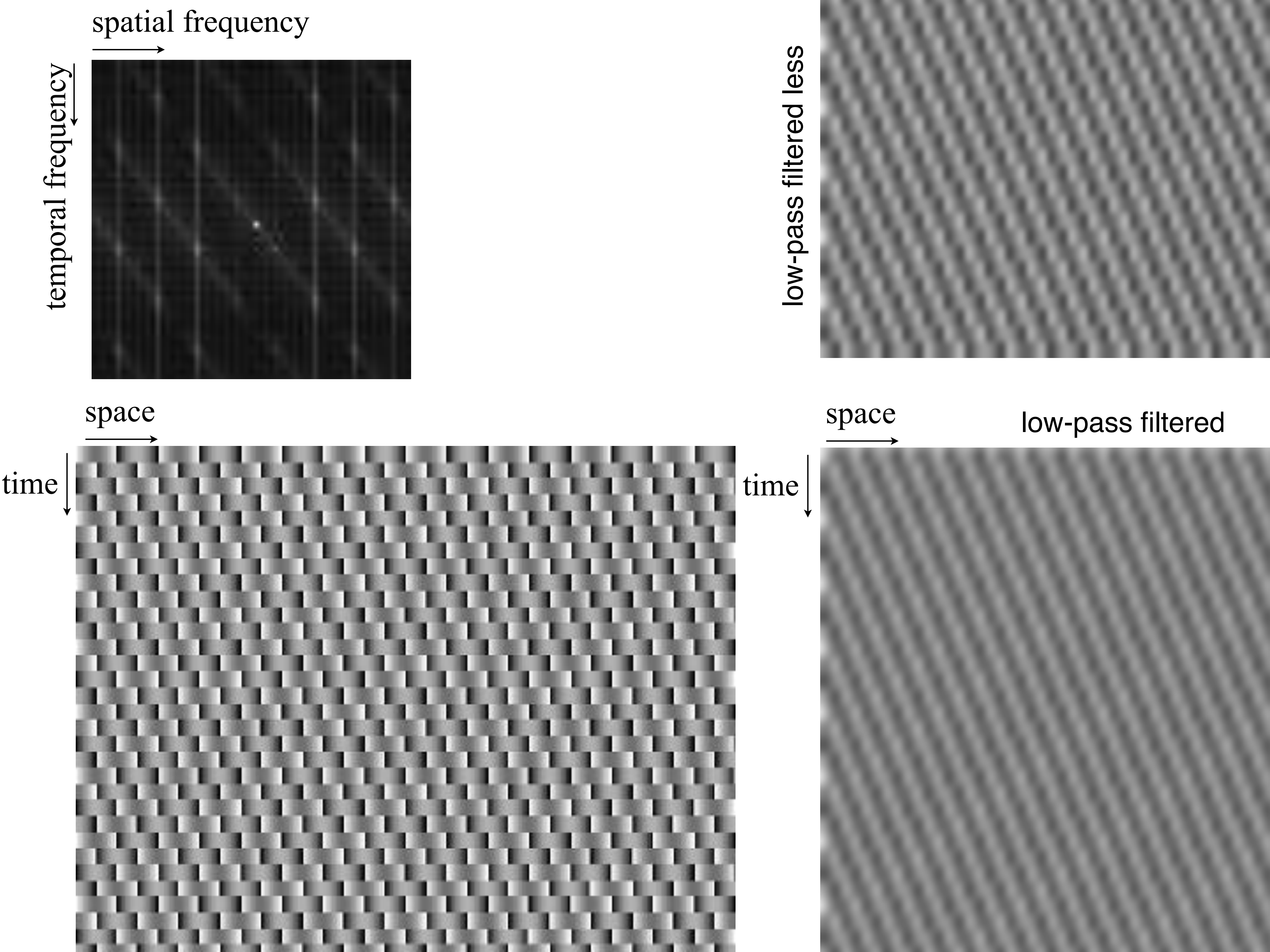

space
low-pass filtered
time



Visual signal

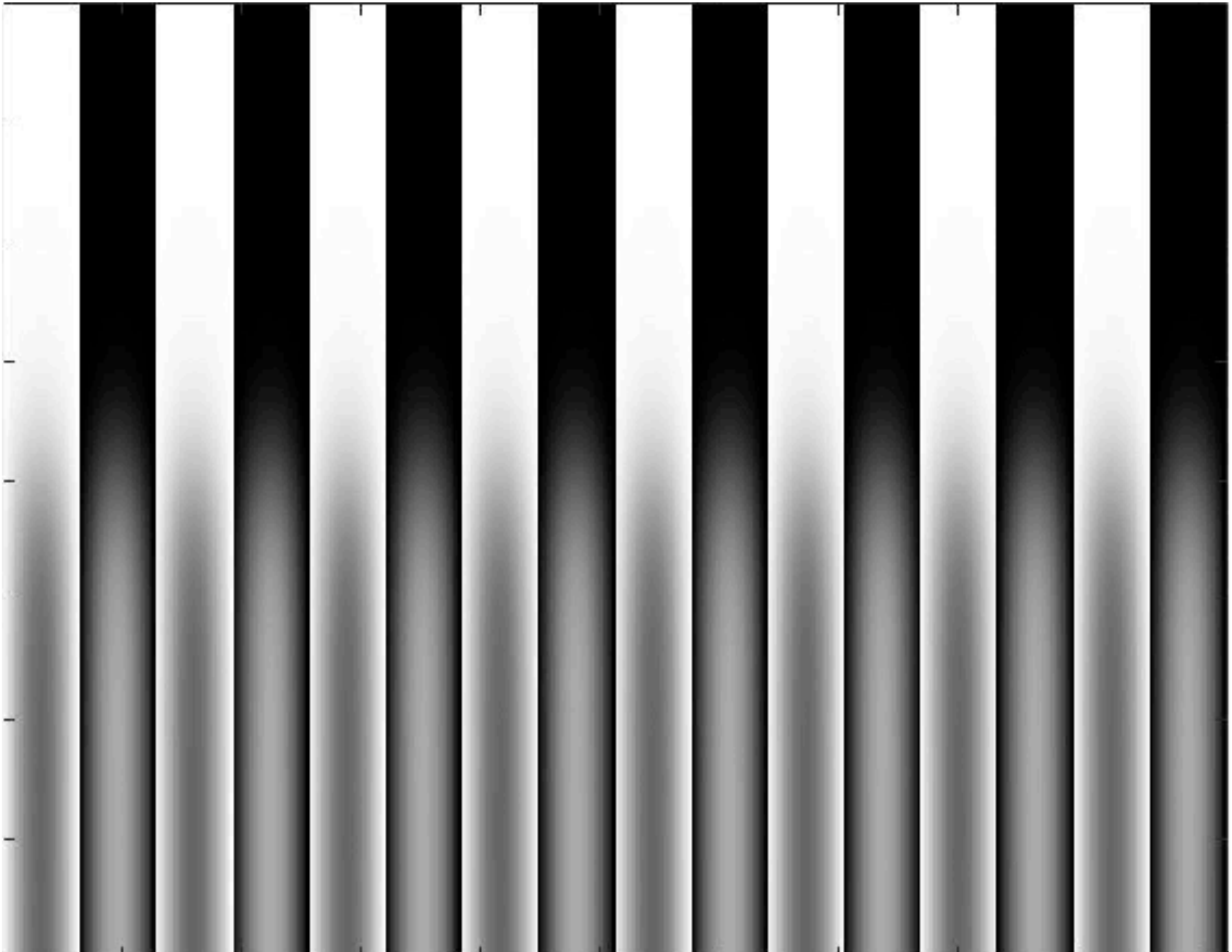




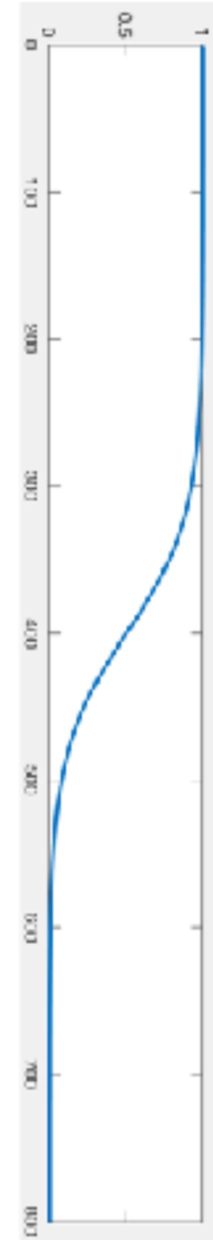


spatial frequency
temporal frequency
low-pass filtered less
space
time
space
time
low-pass filtered

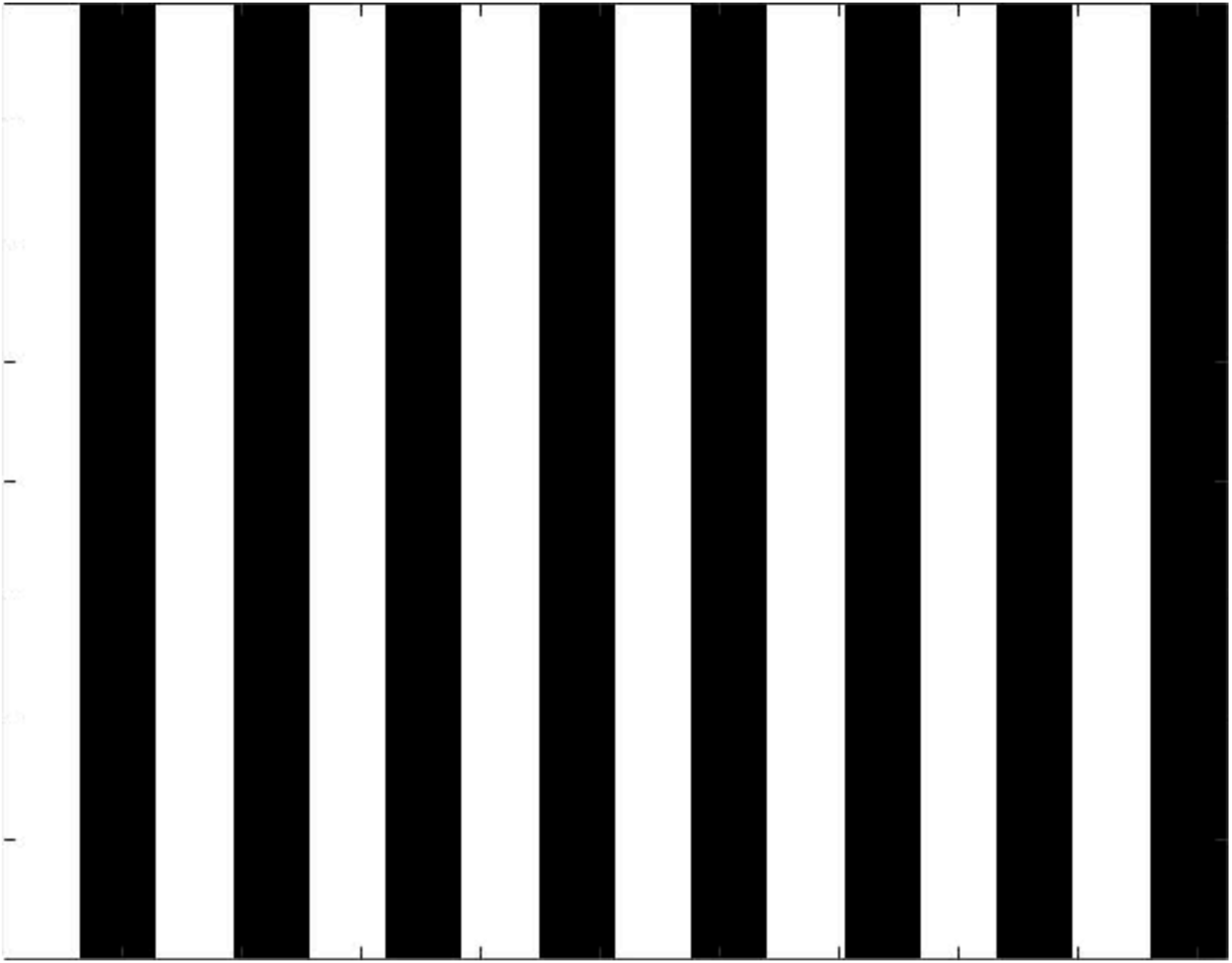
blend over the two conditions



fraction of square wave
fundamental frequency

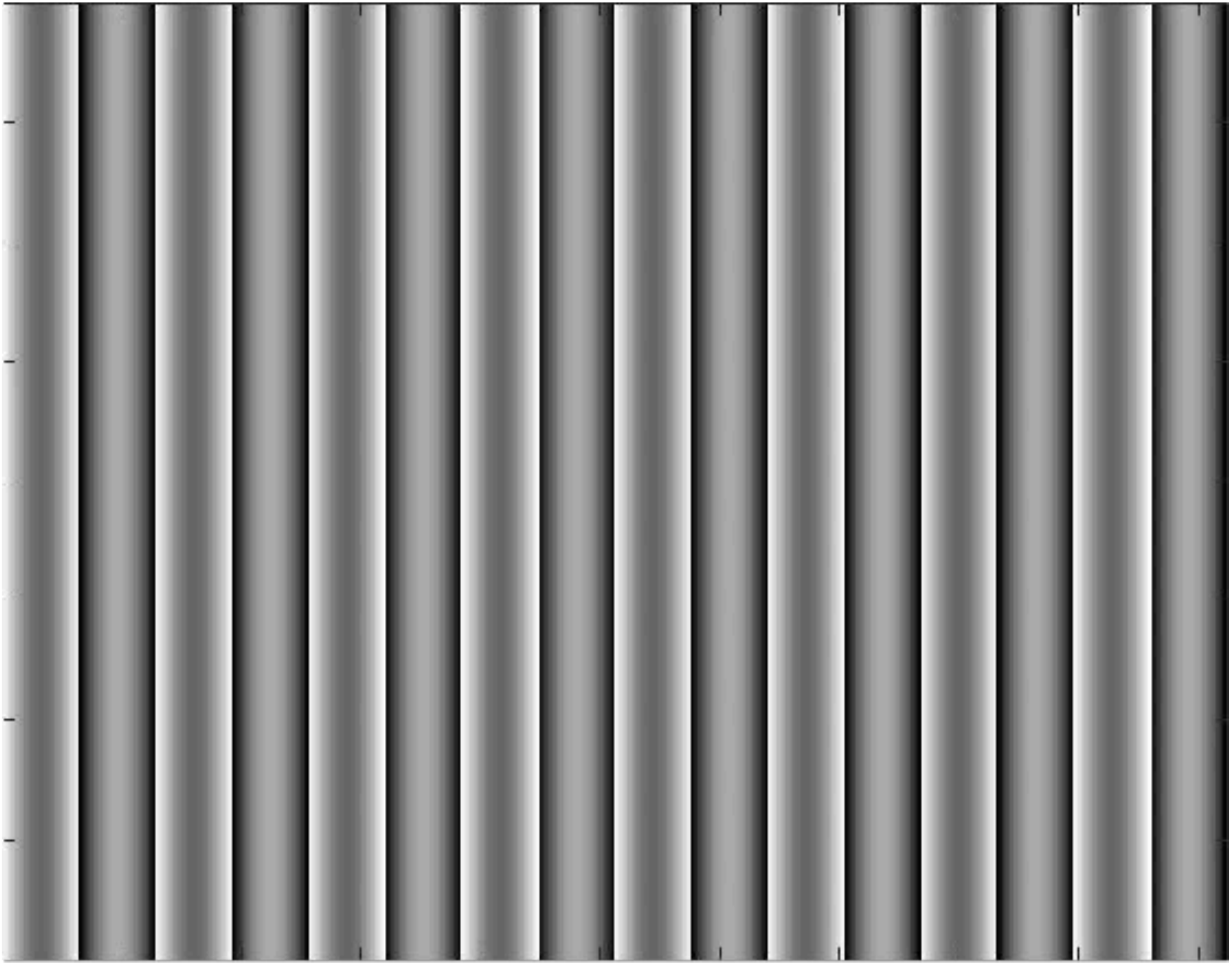


faster display speed



alpha: 1 squareFlag: 1 offset: period/4

faster display speed



alpha: 0 squareFlag: 1 offset: period/4 fast blended…

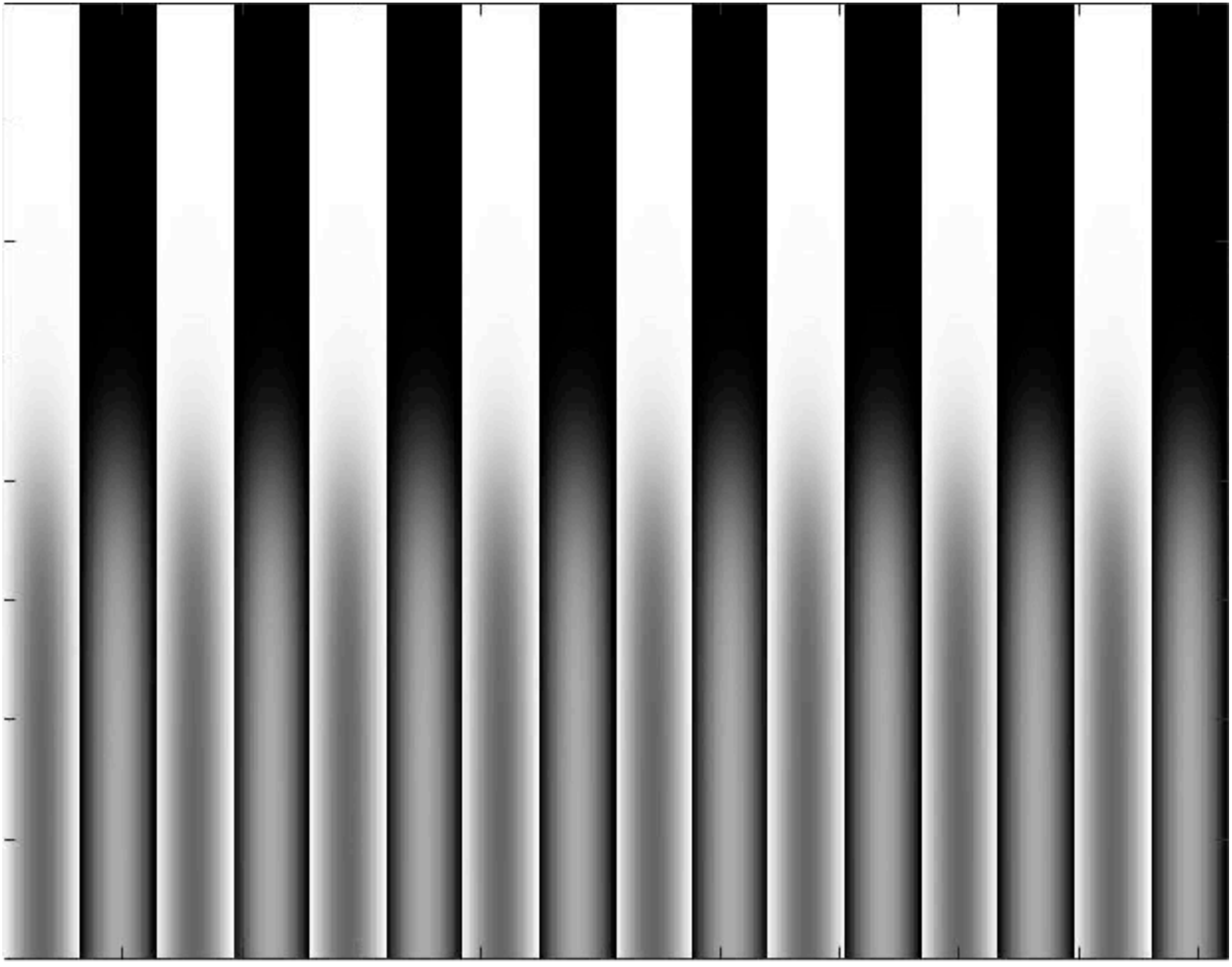


Image pyramids

Image information occurs at all spatial scales

Image pyramids

- Gaussian pyramid
- Laplacian pyramid
- Steerable pyramid

Image pyramids

- **Gaussian pyramid**
- Laplacian pyramid
- Steerable pyramid

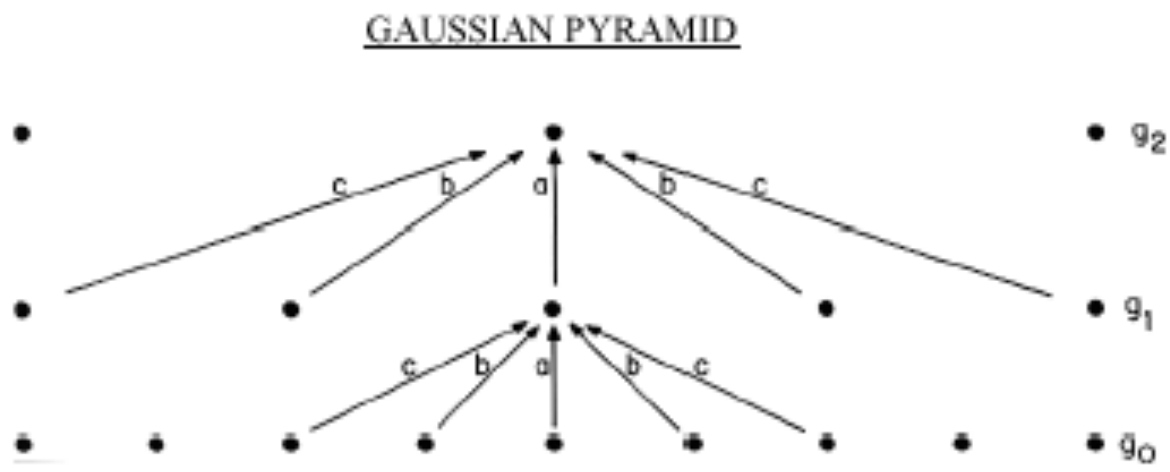
E. H. Adelson | C. H. Anderson | J. R. Bergen | P. J. Burt | J. M. Ogden

Pyramid methods in image processing

http://persci.mit.edu/pub_pdfs/RCA84.pdf

The Gaussian pyramid

Smooth with gaussians, because a gaussian*gaussian=another gaussian



g_0 = IMAGE

g_L = REDUCE [g_{L-1}]

Fig 1. A one-dimensional graphic representation of the process which generates a Gaussian pyramid Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or "generating kernel" is used to generate all levels.

Burt and Adelson

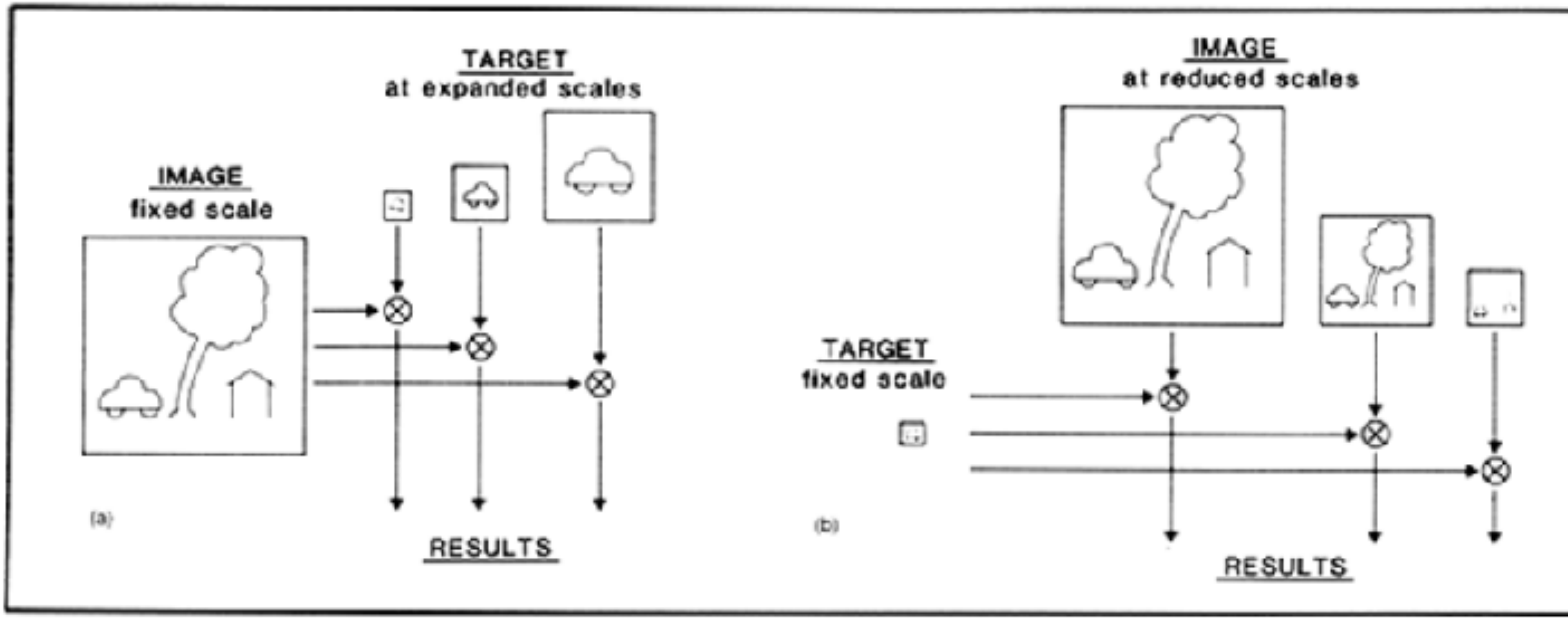


Fig. 1. Two methods of searching for a target pattern over many scales. In the first approach, (a), copies of the target pattern are constructed at several expanded scales, and each is convolved with the original image. In the second approach, (b), a single copy of the target is convolved with copies of the image reduced in scale. The target should be just large enough to resolve critical details. The two approaches should give equivalent results, but the second is more efficient by the fourth power of the scale factor (image convolutions are represented by 'O').

Fig. 2a. *The Gaussian pyramid. The original image, G_0 is repeatedly filtered and subsampled to generate the sequence of reduced resolution image G_1, G_2, etc. These comprise a set of lowpass-filtered copies of the original image in which the bandwidth decreases in one-octave steps.*

G_{00} G_{11} G_{22}

Fig. 2b. *Levels of the Gaussian pyramid expanded to the size of the original image. The effects of lowpass filtering are now clearly apparent.*

512 256 128 64 32 16 8

Convolution and subsampling as a matrix multiply (1-d case)

$$x_2 = G_1 x_1$$

$$G_1 =$$

$$\begin{matrix}
1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0
\end{matrix}$$

(Normalization constant of 1/16 omitted for visual clarity.)

Next pyramid level

$$x_3 = G_2 x_2$$

$$G_2 =$$

$$\begin{matrix} 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{matrix}$$

The combined effect of the two pyramid levels

$$x_3 = G_2 G_1 x_1$$

$G_2 G_1 =$

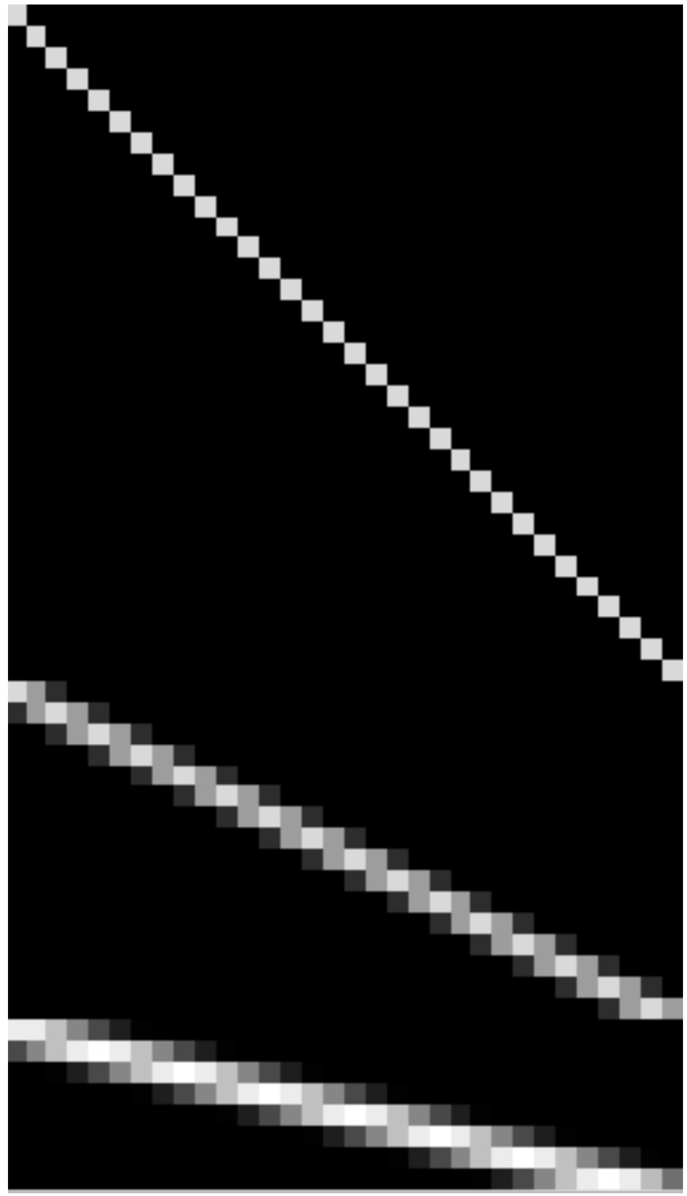
$$\begin{matrix}
1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 30 & 16 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 25 & 16 & 4 & 0
\end{matrix}$$

1-d Gaussian pyramid matrix, for [1 4 6 4 1] low-pass filter

full-band image,
highest resolution

lower-resolution
image

lowest resolution
image



Gaussian pyramids used for

- up- or down- sampling images.
- Multi-resolution image analysis
  - Look for an object over various spatial scales
  - Coarse-to-fine image processing: form blur estimate or the motion analysis on very low-resolution image, upsample and repeat. Often a successful strategy for avoiding local minima in complicated estimation tasks.

Image pyramids

- Gaussian pyramid
- **Laplacian pyramid**
- Steerable pyramid

Down-sampling

Original

Blurred

Downsampled

Down-sampling and Up-sampling

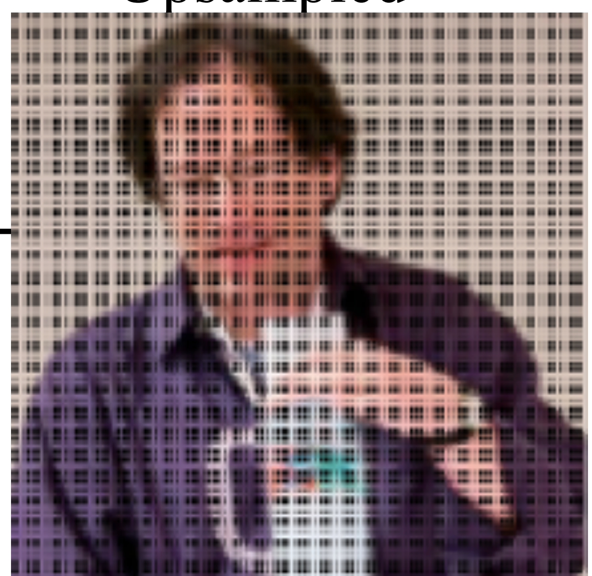
Original

Blurred

Downsampled

Original

Upsampled



Blurred

Upsampling

$$y_2 = F_3 x_3$$

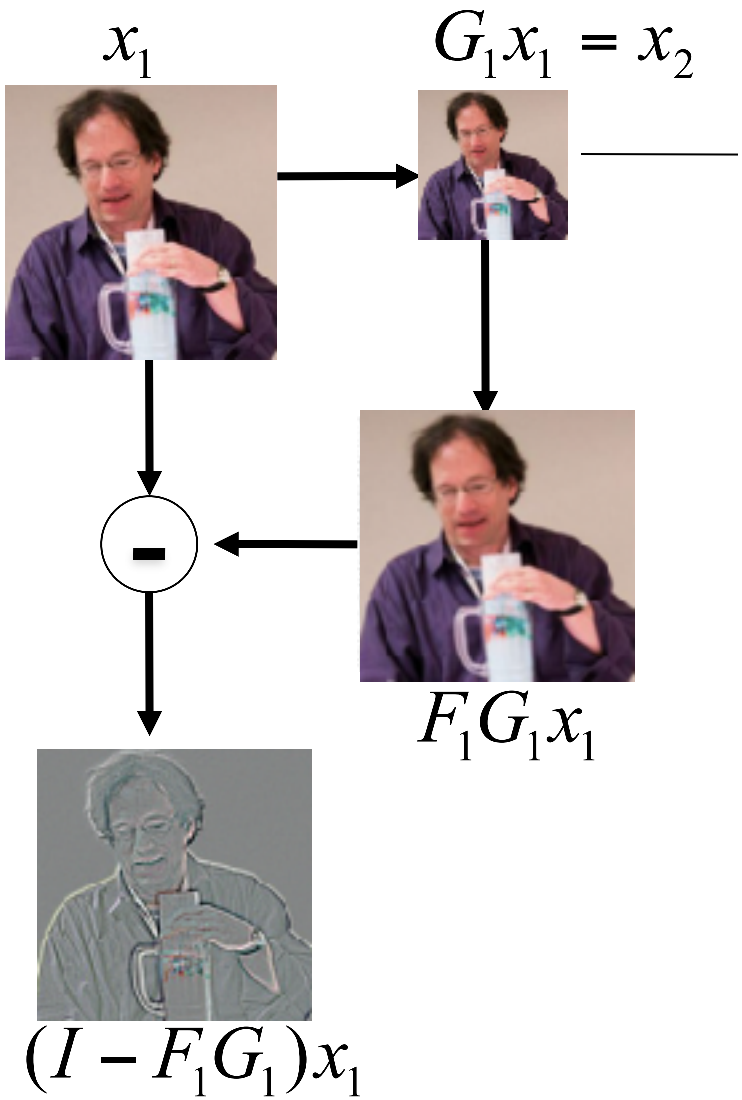
Insert zeros between pixels, then apply a low-pass filter, [1 4 6 4 1]

$$F_3 = \begin{matrix} 6 & 1 & 0 & 0 \\ 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 4 \end{matrix}$$

The Laplacian Pyramid

- Synthesis
  - Compute the difference between upsampled Gaussian pyramid level and Gaussian pyramid level.
  - band pass filter - each level represents spatial frequencies (largely) unrepresented at other level.

Laplacian pyramid algorithm



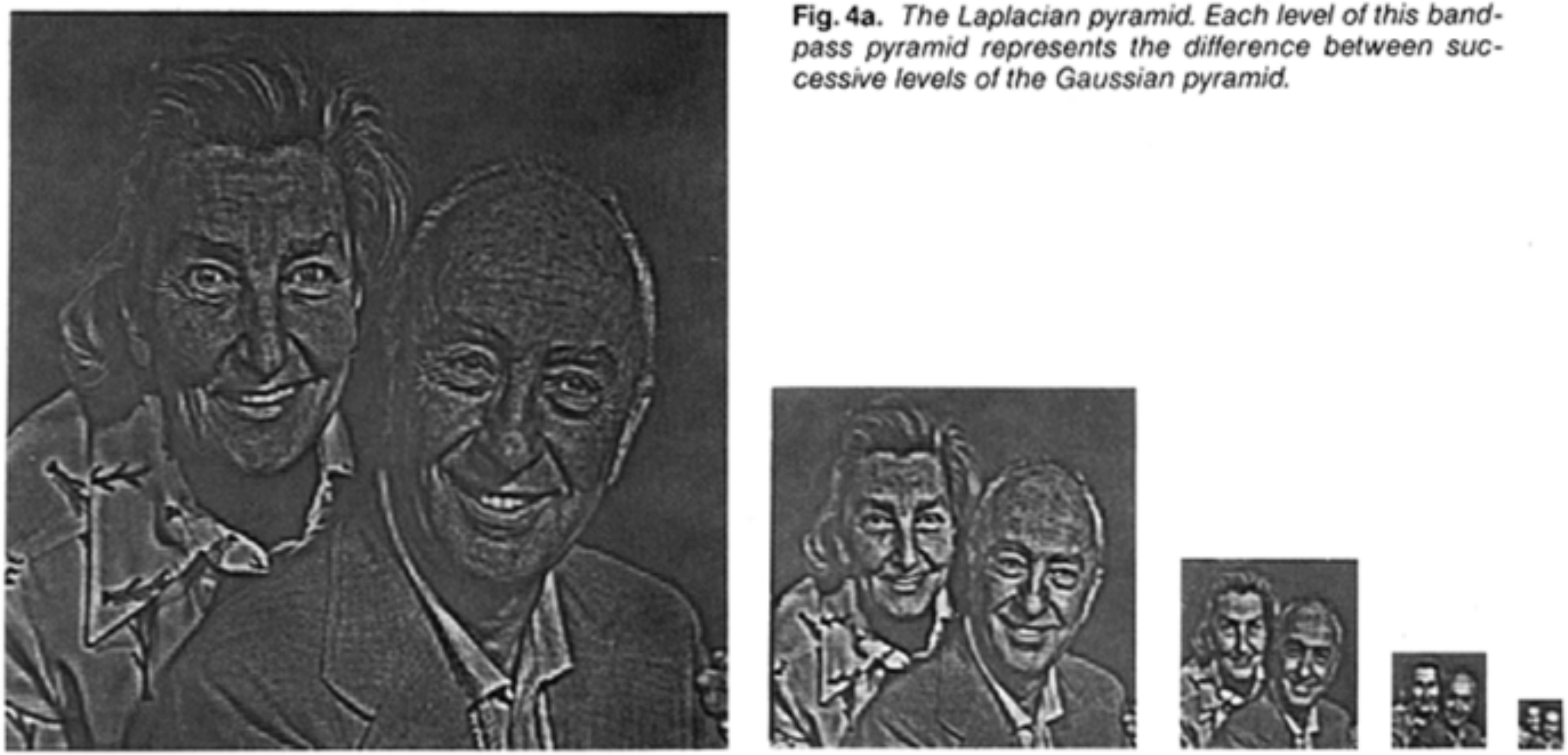


Fig. 4a. *The Laplacian pyramid. Each level of this band-pass pyramid represents the difference between successive levels of the Gaussian pyramid.*

$\mathbf{L}_{0.0}$ $\mathbf{L}_{1.0}$ $\mathbf{L}_{2.2}$

Fig. 4b. *Levels of the Laplacian pyramid expanded to the size of the original image. Note that edge and bar features are enhanced and segregated by size.*

Laplacian pyramid reconstruction algorithm: recover x_1 from L_1, L_2, L_3 and x_4

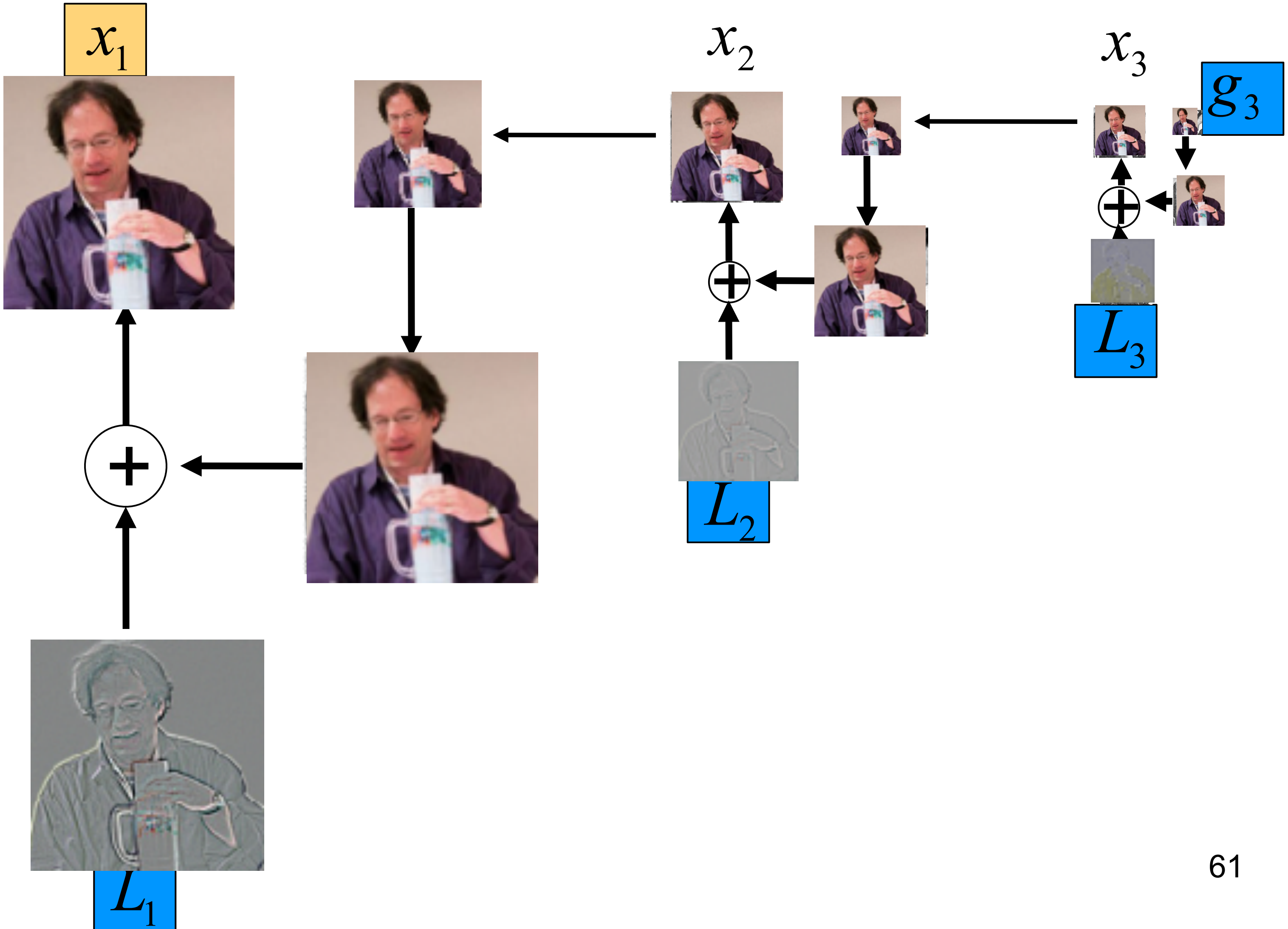
G# is the blur-and-downsample operator at pyramid level #
F# is the blur-and-upsample operator at pyramid level #

First, form Gaussian pyramid:
x2 = G1 x1
x3 = G2 x2
x4 = G3 x3

Then the Laplacian pyramid elements are:
L1 = (I – F1 G1) x1
L2 = (I – F2 G2) x2
L3 = (I – F3 G3) x3

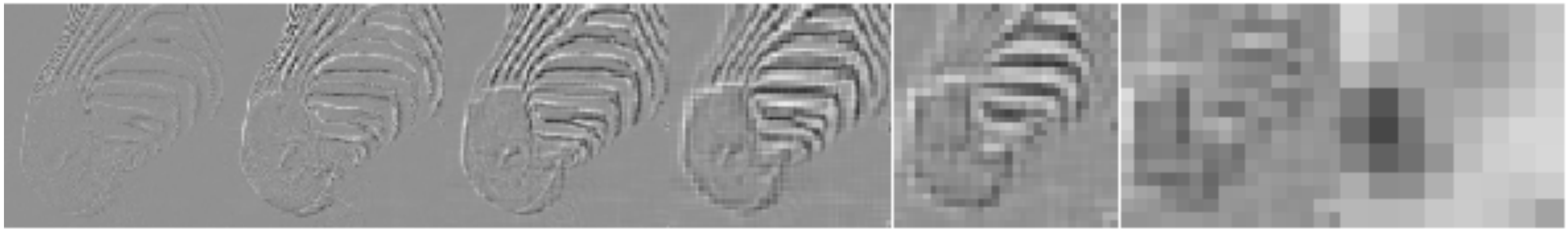
Reconstruction of original image (x1) from Laplacian pyramid elements and the smallest level of the Gaussian pyramid, x4:
x3 = L3 + F3 x4
x2 = L2 + F2 x3
x1 = L1 + F1 x2

Laplacian pyramid reconstruction algorithm: recover x_1 from L_1, L_2, L_3 and g_3



512 256 128 64 32 16 8

Gaussian pyramid



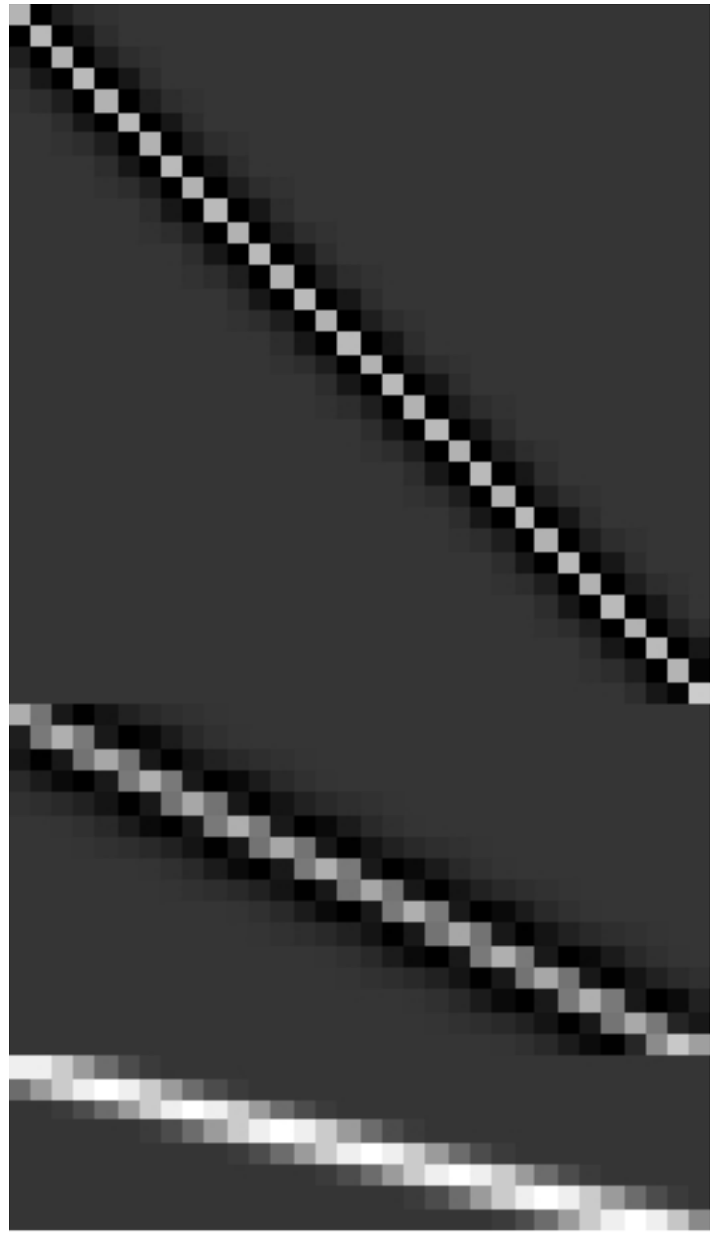
Laplacian pyramid

1-d Laplacian pyramid matrix, for [1 4 6 4 1] low-pass filter

high frequencies

mid-band
frequencies

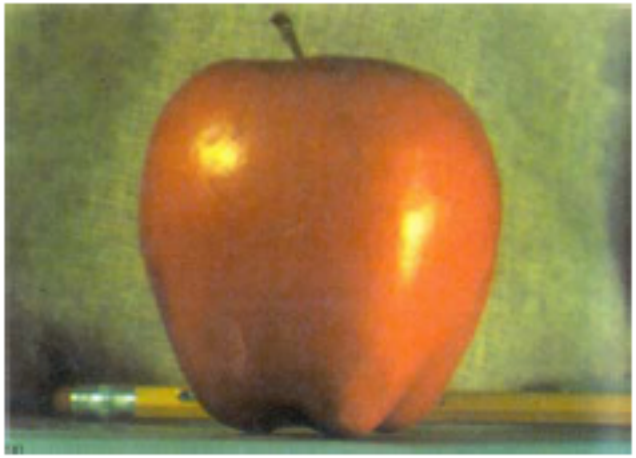
low frequencies



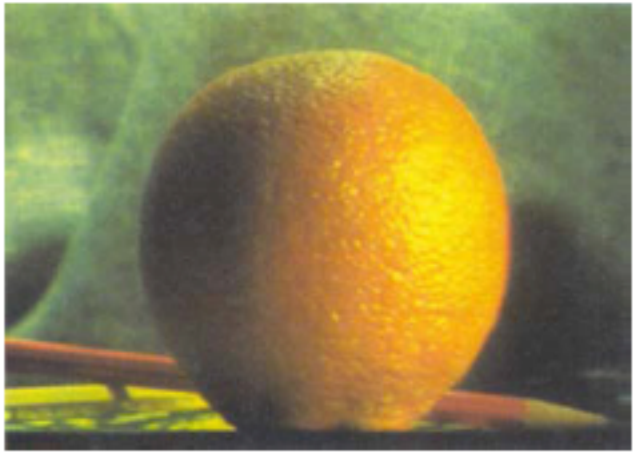
Laplacian pyramid applications

- Texture synthesis
- Image compression
- Noise removal

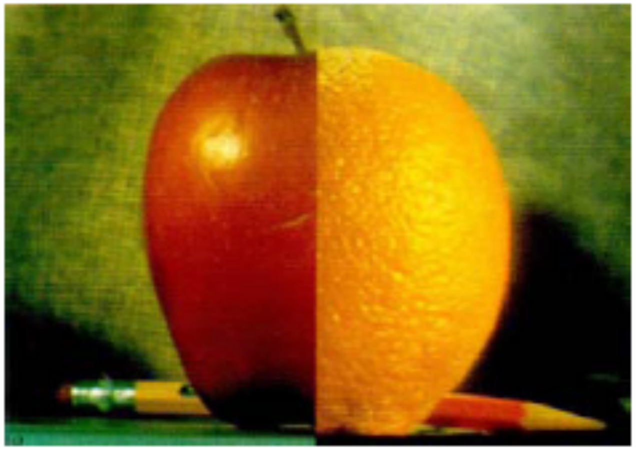
Image blending

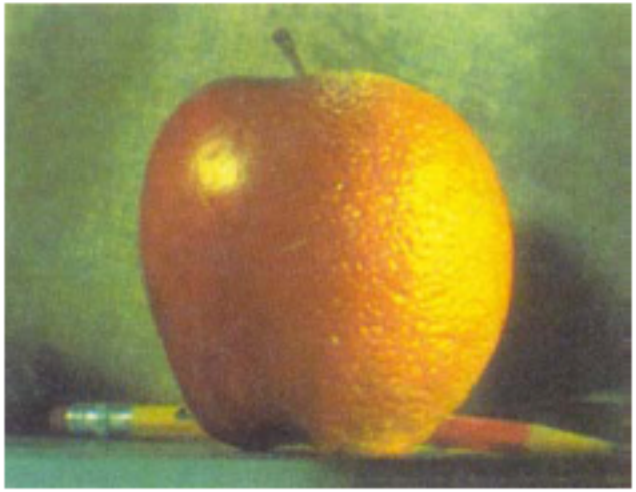


(a)



(b)





Szeliski, Computer Vision, 2010

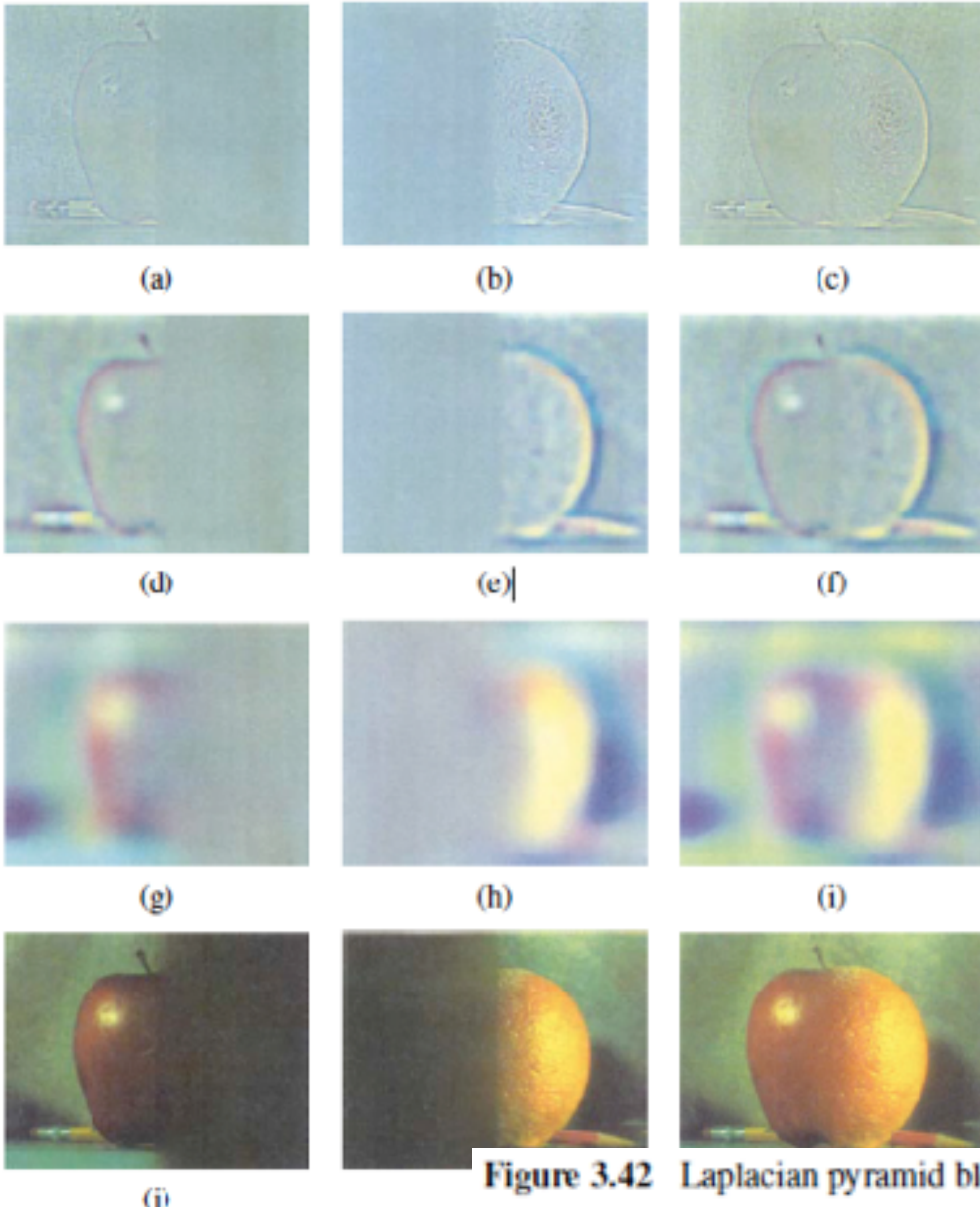
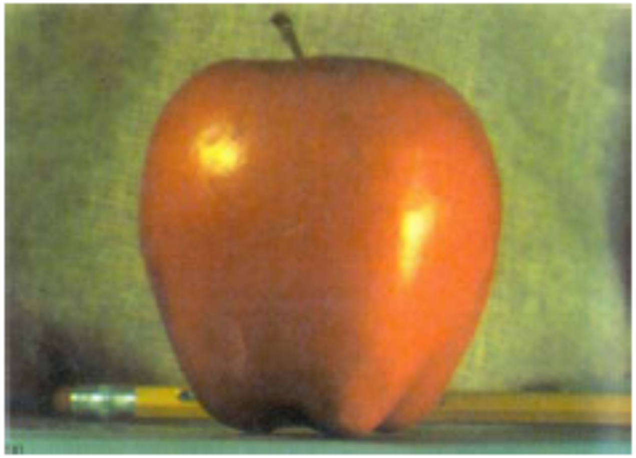
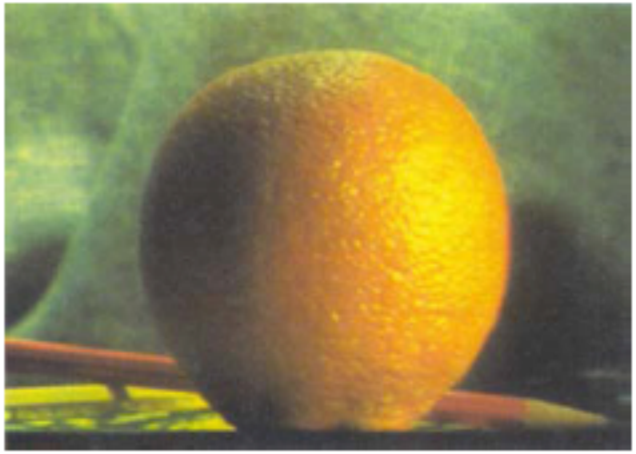


Figure 3.42 Laplacian pyramid blending details (Burt and Adelson 1983b) © 1983 ACM. The first three rows show the high, medium, and low frequency parts of the Laplacian pyramid (taken from levels 0, 2, and 4). The left and middle columns show the original apple and orange images weighted by the smooth interpolation functions, while the right column shows the averaged contributions.

Image blending





- Build Laplacian pyramid for both images: LA, LB
- Build Gaussian pyramid for mask: G
- Build a combined Laplacian pyramid: L(j) = G(j) LA(j) + (1-G(j)) LB(j)
- Collapse L to obtain the blended image

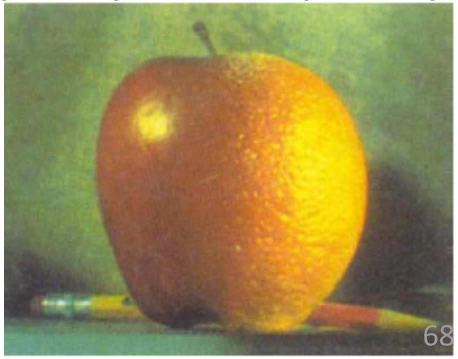


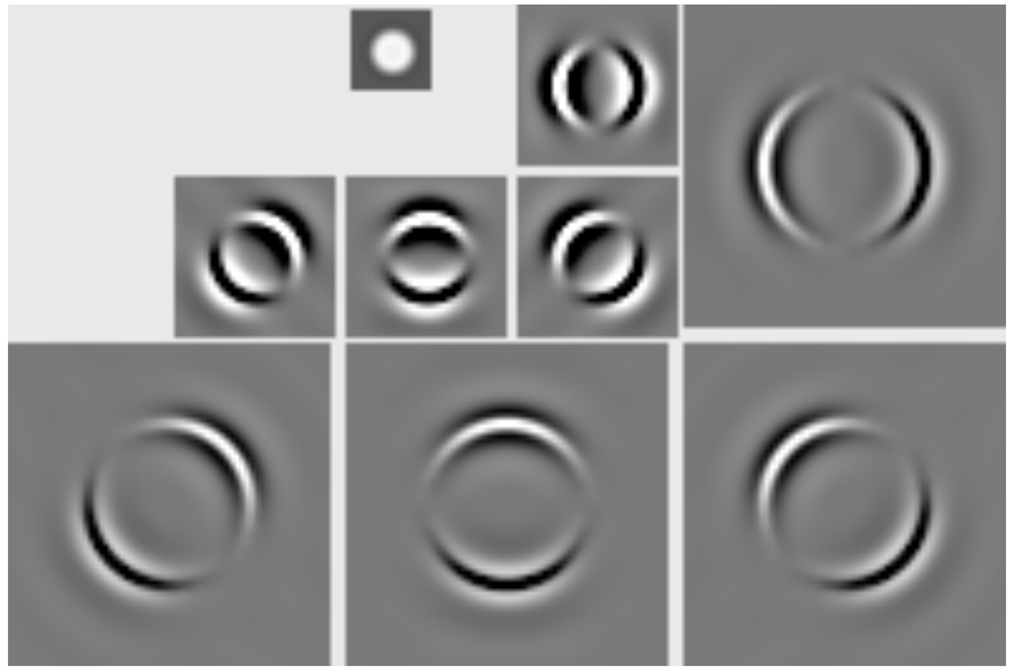
Image pyramids

- Gaussian pyramid
- Laplacian pyramid
- **Steerable pyramid**

Steerable Pyramid

2 Level decomposition of white circle example:

Low pass residual

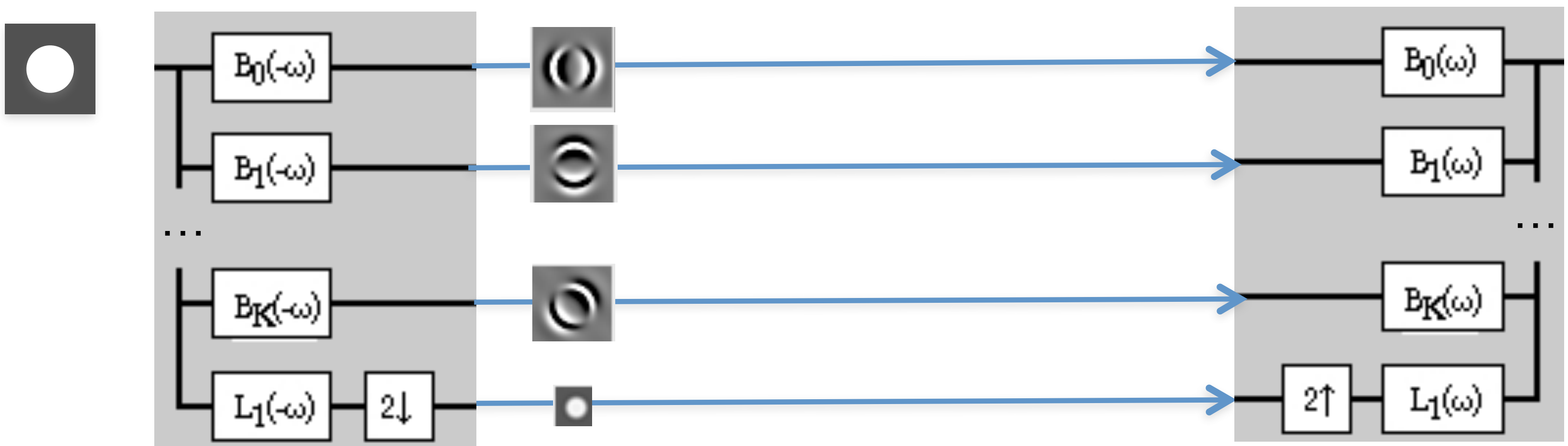


Subbands

Images from: http://www.cis.upenn.edu/~eero/steerpyr.html

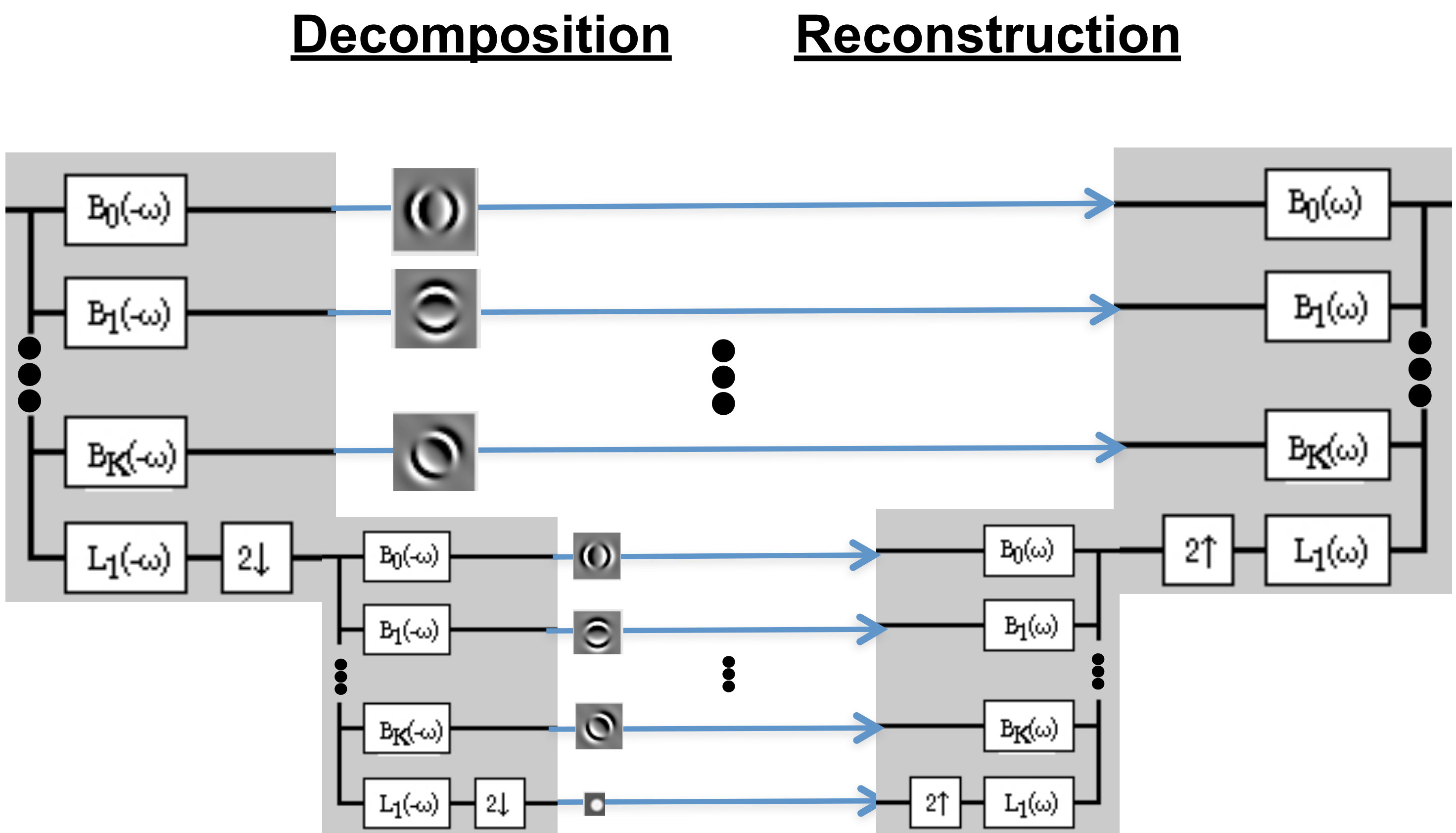
Steerable Pyramid

Decomposition **Reconstruction**

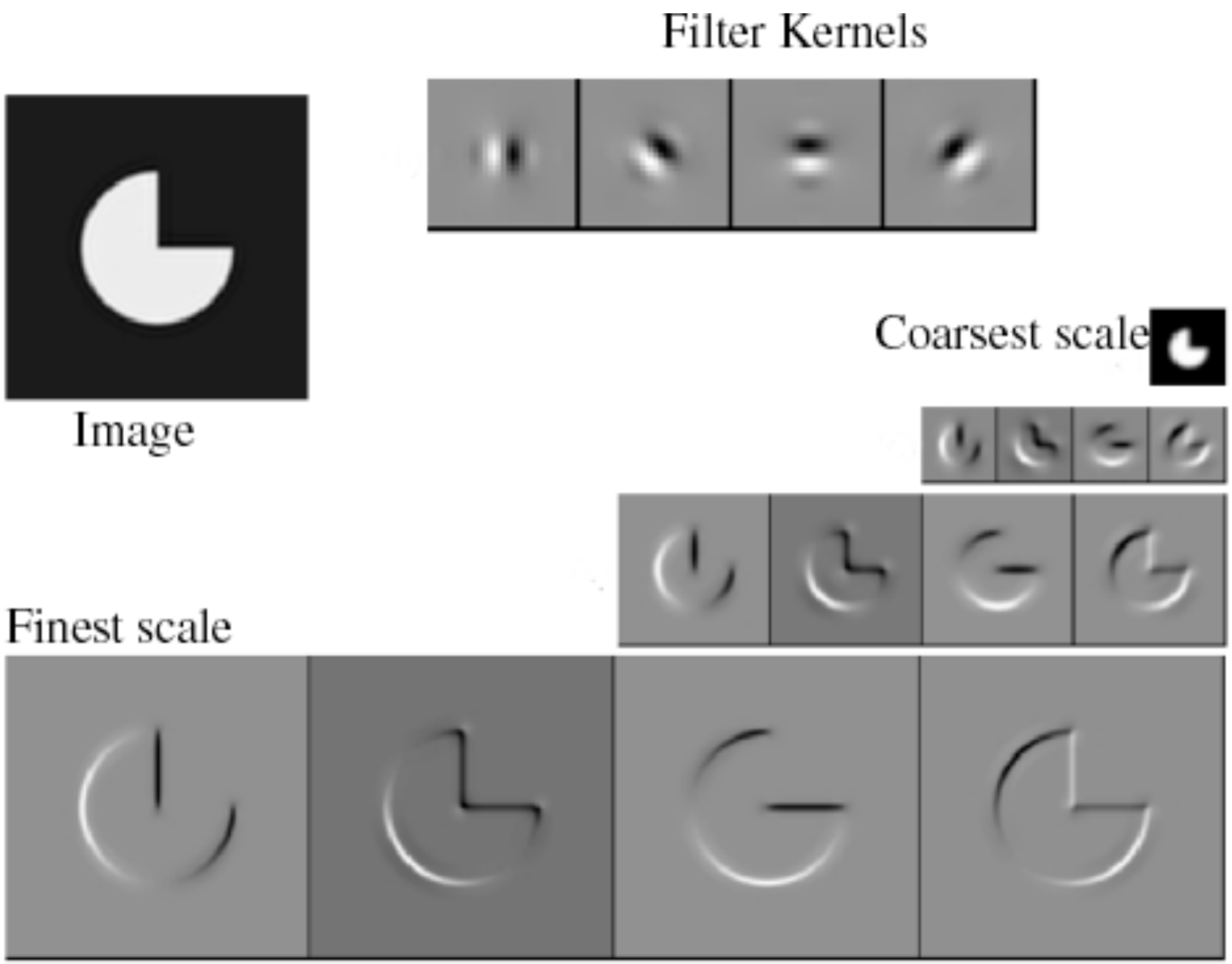


Images from: http://www.cis.upenn.edu/~eero/steerpyr.html

Steerable Pyramid



Images from: http://www.cis.upenn.edu/~eero/steerpyr.html



Reprinted from “Shiftable MultiScale Transforms,” by Simoncelli et al., IEEE Transactions on Information Theory, 1992, copyright 1992, IEEE

There is also a high pass residual…

But we need to get rid of the corner regions before starting the recursive circular filtering

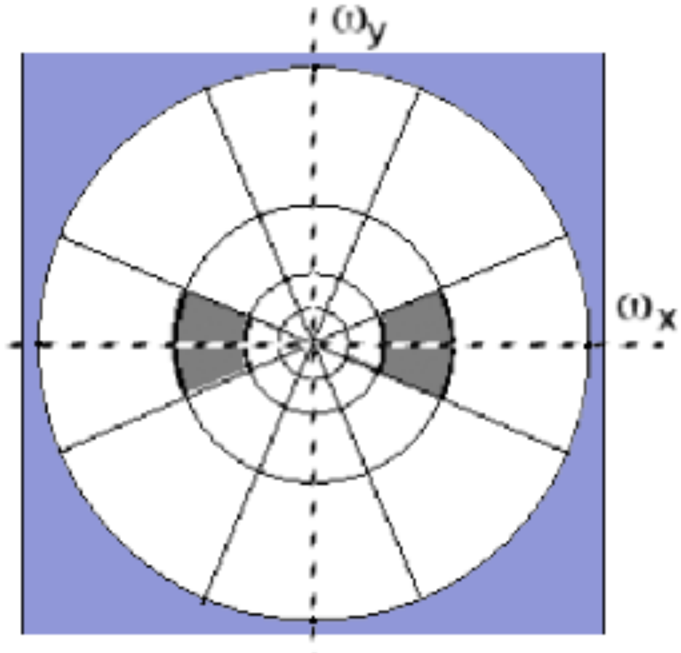


Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For example, the shaded region corresponds to the spectral support of a single (vertically-oriented) subband.

http://www.cns.nyu.edu/ftp/eero/simoncelli95b.pdf

Simoncelli and Freeman, ICIP 1995

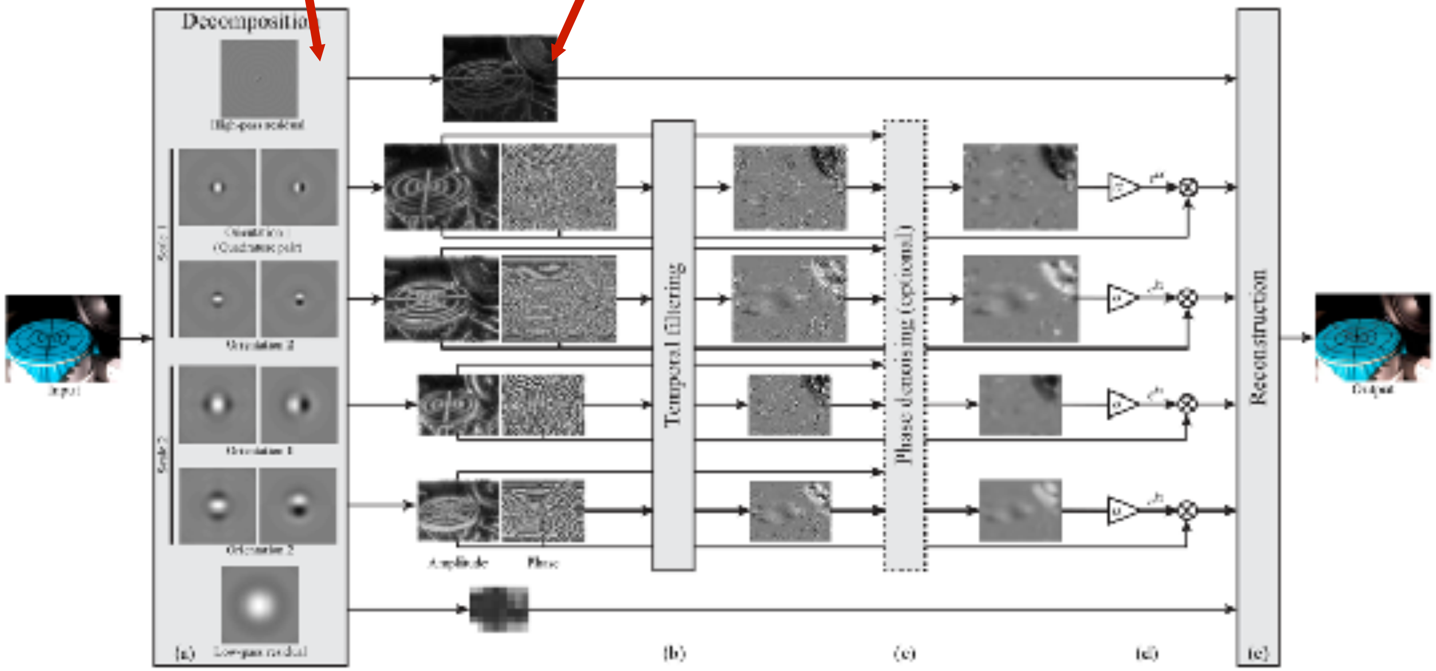
Steerable pyramids

- Good:
  - Oriented subbands
  - Non-aliased subbands
  - Steerable filters
  - Used for: noise removal, texture analysis and synthesis, super-resolution, shading/paint discrimination.
- Bad:
  - Overcomplete
  - Have one high frequency residual subband, required in order to form a circular region of analysis in frequency from a square region of support in frequency.

Phase-based Pipeline (SIGGRAPH'13)

Amplitude

Phase



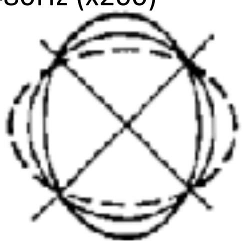
Complex steerable pyramid
[Simoncelli and Freeman 1995]

Temporal filtering on **phases**

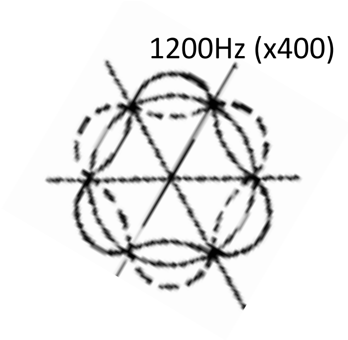
Vibration Modes of PVC pipe

Source (20000 FPS)

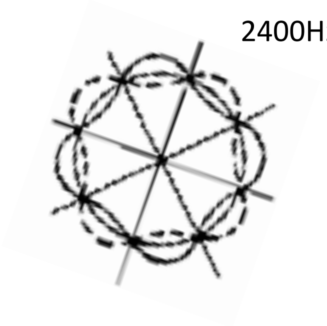
480Hz (x200)



1200Hz (x400)



2400Hz (x1200)



Sequences courtesy of Justin Chen,
Civil Engineering, MIT

"Piping Vibration Analysis"
[Wachel et al. 1990]

Image pyramids

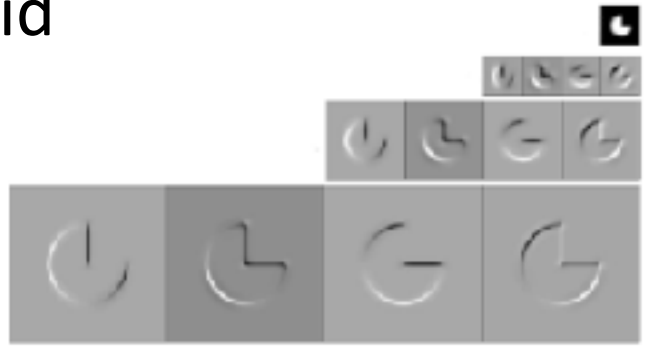
- Gaussian

Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

- Laplacian

Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- Steerable pyramid



Shows components at each scale and orientation separately. Non-aliased subbands. Good for texture and feature analysis. But overcomplete and with HF residual.

Schematic pictures of each matrix transform

Shown for 1-d images

The matrices for 2-d images are the same idea, but more complicated, to account for vertical, as well as horizontal, neighbor relationships.

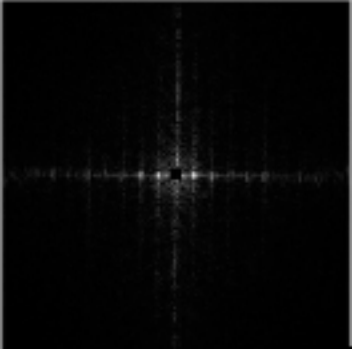
transformed image

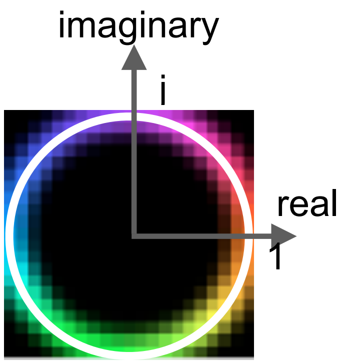
$$\vec{F} = U\vec{f}$$

Vectorized image

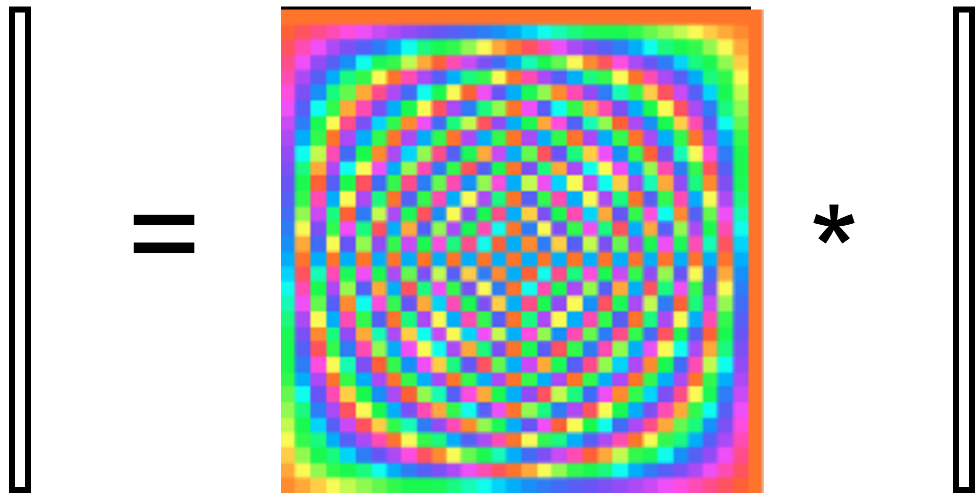
Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

Fourier transform





color key



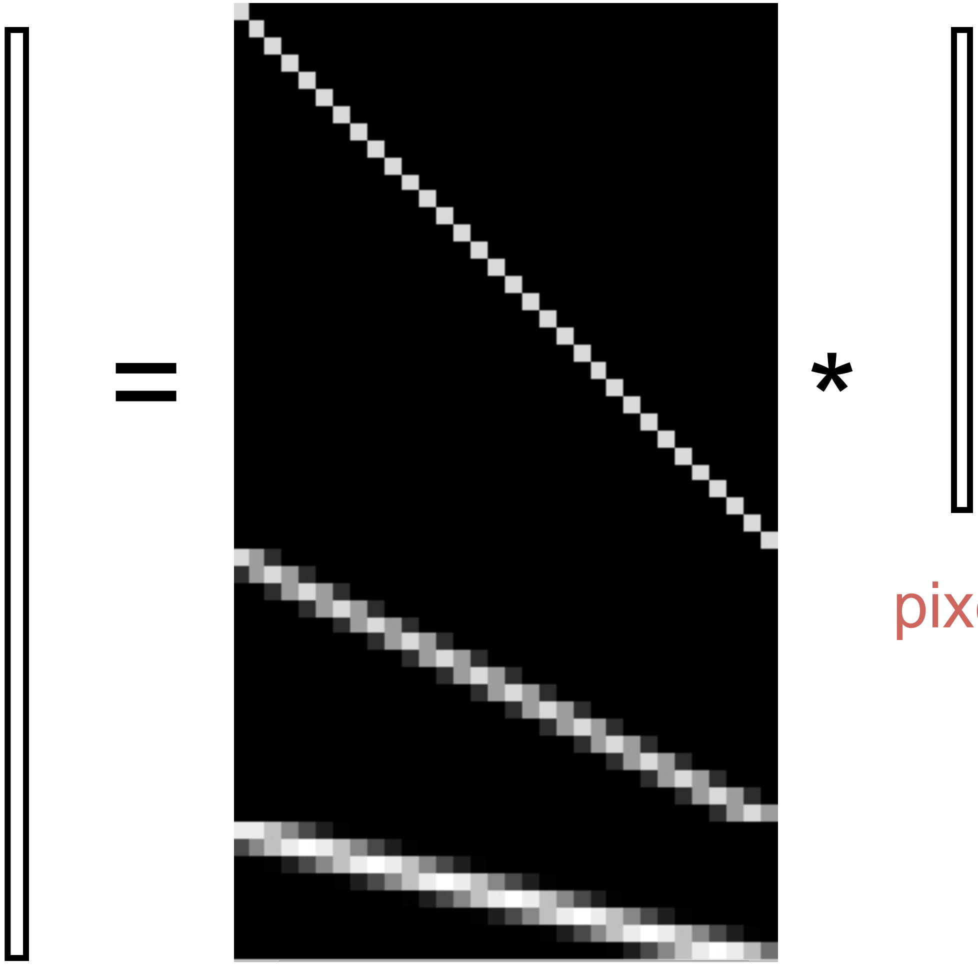
Fourier transform

Fourier bases are global: each transform coefficient depends on all pixel

pixel domain image

Gaussian pyramid

Gaussian pyramid



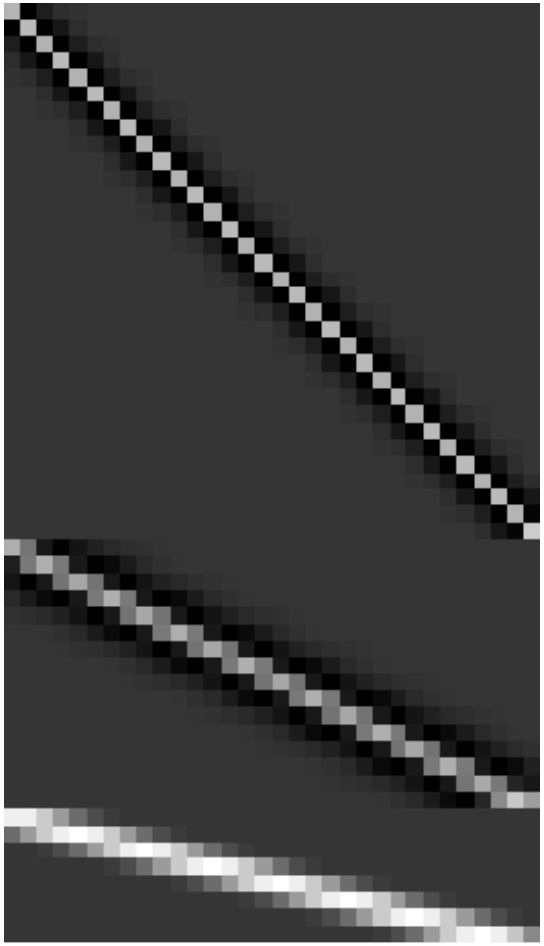
pixel image

Overcomplete representation. Low-pass filters, sampled

Laplacian pyramid

Laplacian pyramid

=

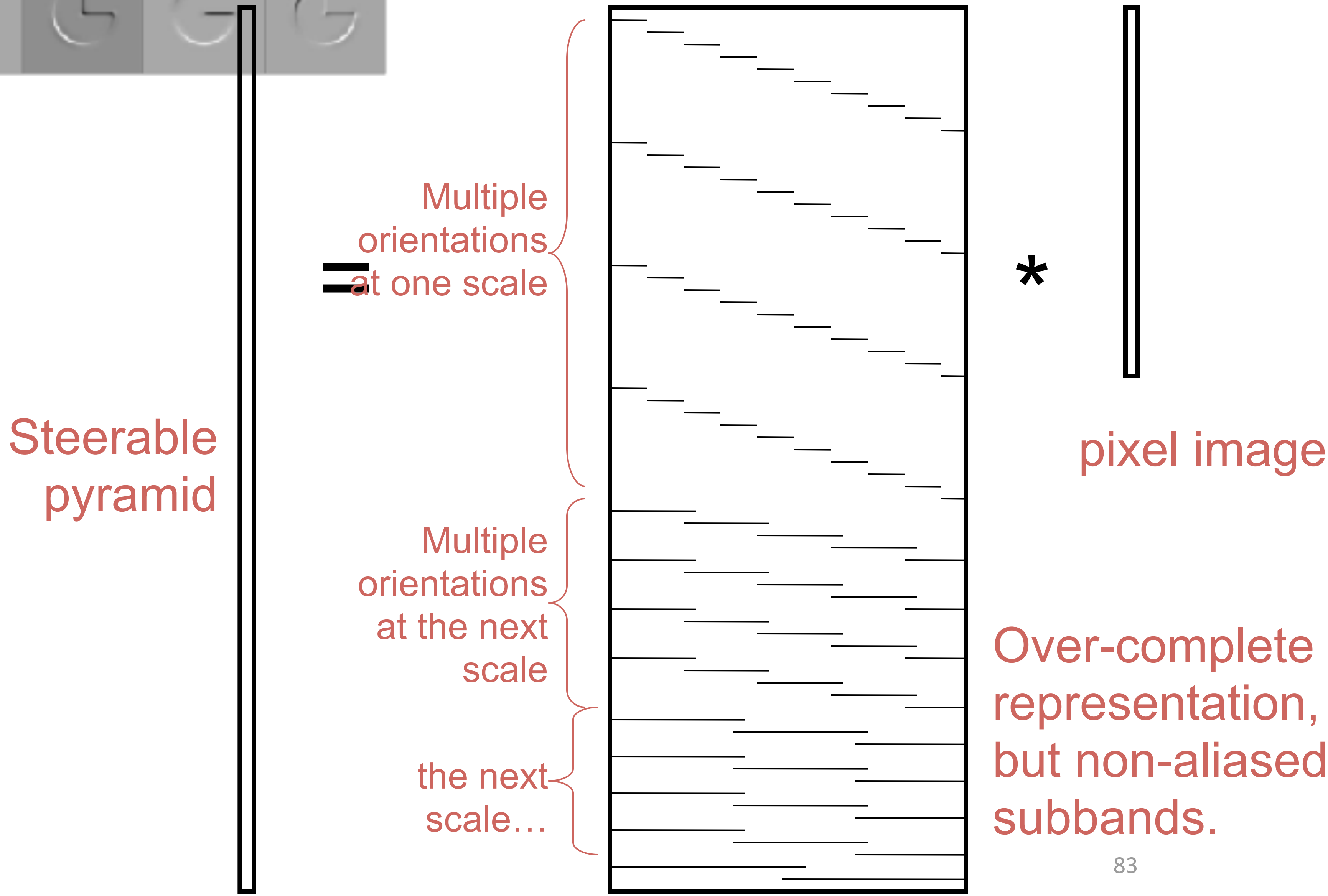


*

pixel image

Overcomplete representation. Transformed pixels

Steerable pyramid



Matlab resources for pyramids (with tutorial)

http://www.cns.nyu.edu/~eero/software.html

Eero P. Simoncelli

Associate Investigator,
Howard Hughes Medical Institute

Associate Professor,
Neural Science and Mathematics,
New York University

Matlab resources for pyramids (with tutorial)

http://www.cns.nyu.edu/~eero/software.html

lcv

Laboratory for Computational Vision

Home | People | Research | Publications | Software

Publicly Available Software Packages

- Texture Analysis/Synthesis - Matlab code is available for analyzing and synthesizing visual textures. README | Contents | ChangeLog | Source code (UNIX/PC, gzip'ed tar file)
- EPWIC - Embedded Progressive Wavelet Image Coder. C source code available.
- **matlabPyrTools** - Matlab source code for multi-scale image processing. Includes tools for building and manipulating Laplacian pyramids, QMF/Wavelets, and steerable pyramids. Data structures are compatible with the Matlab wavelet toolbox, but the convolution code (in C) is faster and has many boundary-handling options. README, Contents, Modification list, UNIX/PC source or Macintosh source.
- The Steerable Pyramid, an (approximately) translation- and rotation-invariant multi-scale image decomposition. MatLab (see above) and C implementations are available.
- Computational Models of cortical neurons. Macintosh program available.
- EPIC - Efficient Pyramid (Wavelet) Image Coder. C source code available.
- OBVIUS [Object-Based Vision & Image Understanding System]: README / ChangeLog / Doc (225k) / Source Code (2.25M).
- CL-SHELL [Gnu Emacs <-> Common Lisp Interface]: README / Change Log / Source Code (119k).

Why use these representations?

- Handle real-world size variations with a constant-size vision algorithm.
- Remove noise
- Analyze texture
- Recognize objects
- Label image features
- Image priors can be specified naturally in terms of wavelet pyramids.