C280, Computer Vision

Prof. Trevor Darrell
trevor@eecs.berkeley.edu

Lecture 5: Pyramids

Last time: Image Filters

- Filters allow local image neighborhood to influence our description and features
  - Smoothing to reduce noise
  - Derivatives to locate contrast, gradient
- Filters have highest response on neighborhoods that “look like” it; can be thought of as template matching.
- Convolution properties will influence the efficiency with which we can process images.
  - Associative
  - Filter separability
- Edge detection processes the image gradient to find curves, or chains of edgels.

Today

- Review of Fourier Transform
- Sampling and Aliasing
- Image Pyramids
- Applications: Blending and noise removal

Background: Fourier Analysis

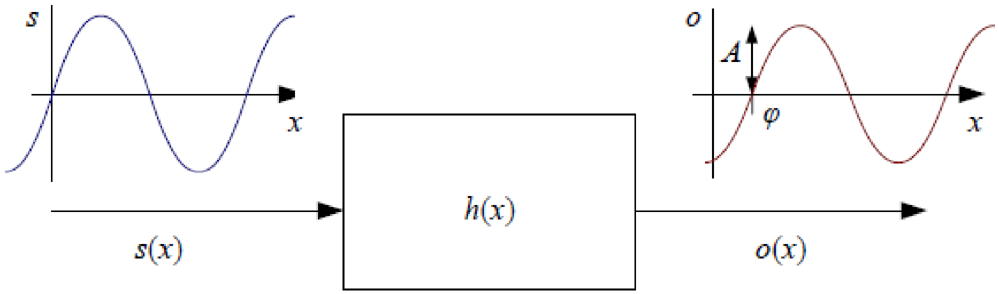
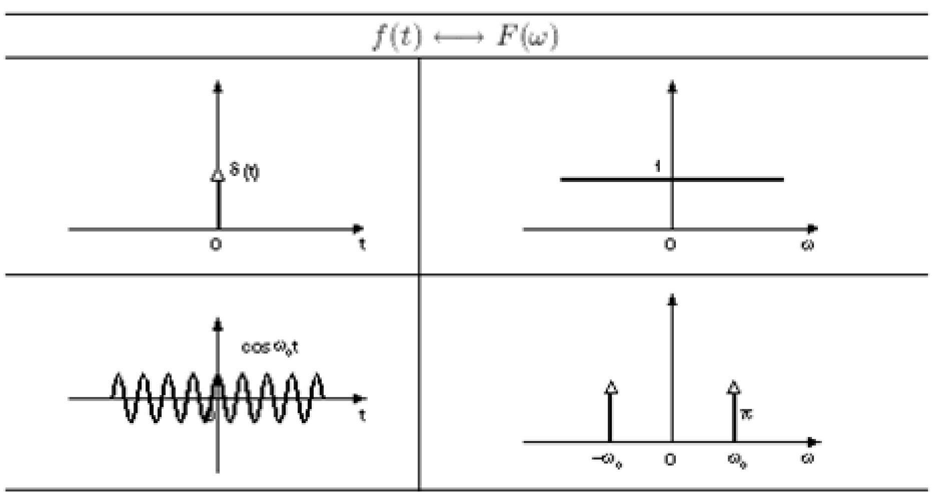


Figure 3.23: *The Fourier Transform as the response of a filter* $h(x)$ *to an input sinusoid* $s(x) = e^{j\omega x}$ *yielding an output sinusoid* $o(x) = h(x) * s(x) = Ae^{j\omega x+\phi}$.



Note symmetry in magnitude F(w)=F(-w)

Background: Fourier Analysis

Property	Signal	Transform
superposition	$f_1(x) + f_2(x)$	$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$	$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$	$F^*(\omega)$
convolution	$f(x) * h(x)$	$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$	$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$	$F(\omega) * H(\omega)$
differentiation	$f'(x)$	$j\omega F(\omega)$
domain scaling	$f(ax)$	$1/aF(\omega/a)$
real images	$f(x) = f^*(x) \Leftrightarrow F(\omega) = F(-\omega)$	
Parseval's Thm.	$\sum_x [f(x)]^2 = \sum_\omega [F(\omega)]^2$	

Table 3.1: *Some useful properties of Fourier transforms. The original transform pair is* $F(\omega) = \mathcal{F}\{f(x)\}$.

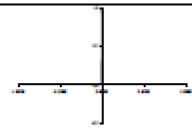
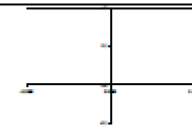
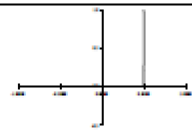
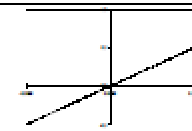
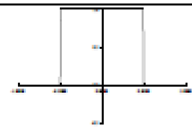
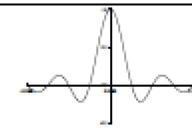
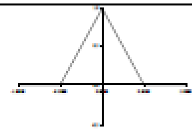
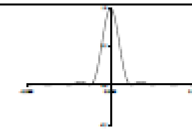
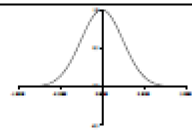
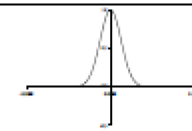
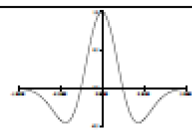
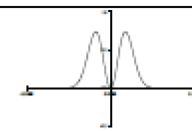
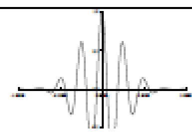
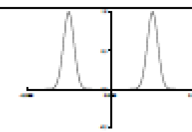
Name	Signal	Transform
impulse	 $\delta(x)$	1 
shifted impulse	 $\delta(x-u)$	$e^{-j\omega u}$ 
box filter	 $\mathrm{box}(x/a)$	$a\mathrm{sinc}(a\omega)$ 
tent	 $\mathrm{tent}(x/a)$	$a\mathrm{sinc}^2(a\omega)$ 
Gaussian	 $G(x;\sigma)$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$ 
Lapl. of Gauss.	 $(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$ 
Gabor	 $\cos(\omega_0 x)G(x;\sigma)$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega\pm\omega_0;\sigma^{-1})$ 

Table 3.2: *Some useful (continuous) Fourier transforms 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 rather drawn to illustrate the general shape and characteristics of the filter or its response. In particular, the Laplacian of a Gaussian is drawn inverted because it resembles more the "Mexican Hat" it is sometimes called.*

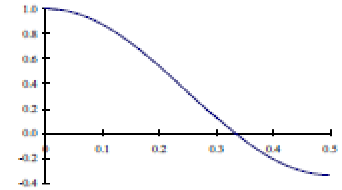
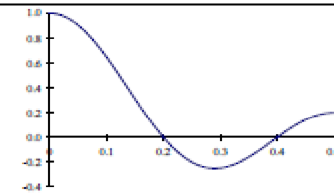
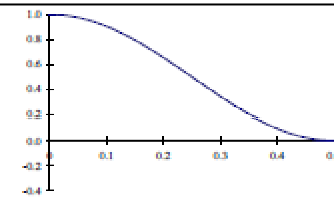
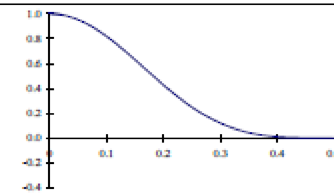
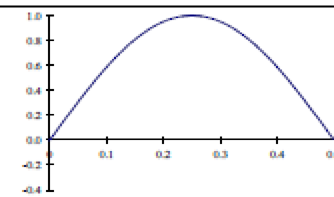
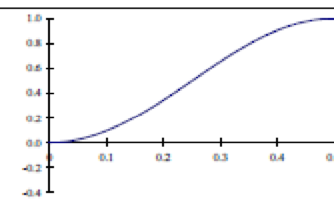
Name	Kernel	Transform	Plot
box-3	$\frac{1}{3}$ | 1 | 1 | 1 |	$\frac{1}{3}(1 + 2\cos\omega)$	
box-5	$\frac{1}{5}$ | 1 | 1 | 1 | 1 | 1 |	$\frac{1}{5}(1 + 2\cos\omega + 2\cos 2\omega)$	
linear	$\frac{1}{4}$ | 1 | 2 | 1 |	$\frac{1}{2}(1 + \cos\omega)$	
binomial	$\frac{1}{16}$ | 1 | 4 | 6 | 4 | 1 |	$\frac{1}{4}(1 + \cos\omega)^2$	
Sobel	$\frac{1}{2}$ | −1 | 0 | 1 |	$\sin\omega$	
"Laplacian"	$\frac{1}{2}$ | −1 | 2 | −1 |	$\frac{1}{2}(1 - \cos\omega)$	

Table 3.3: *Fourier transforms of the separable kernels shown in Figure 3.13.*

Background: high/low pass

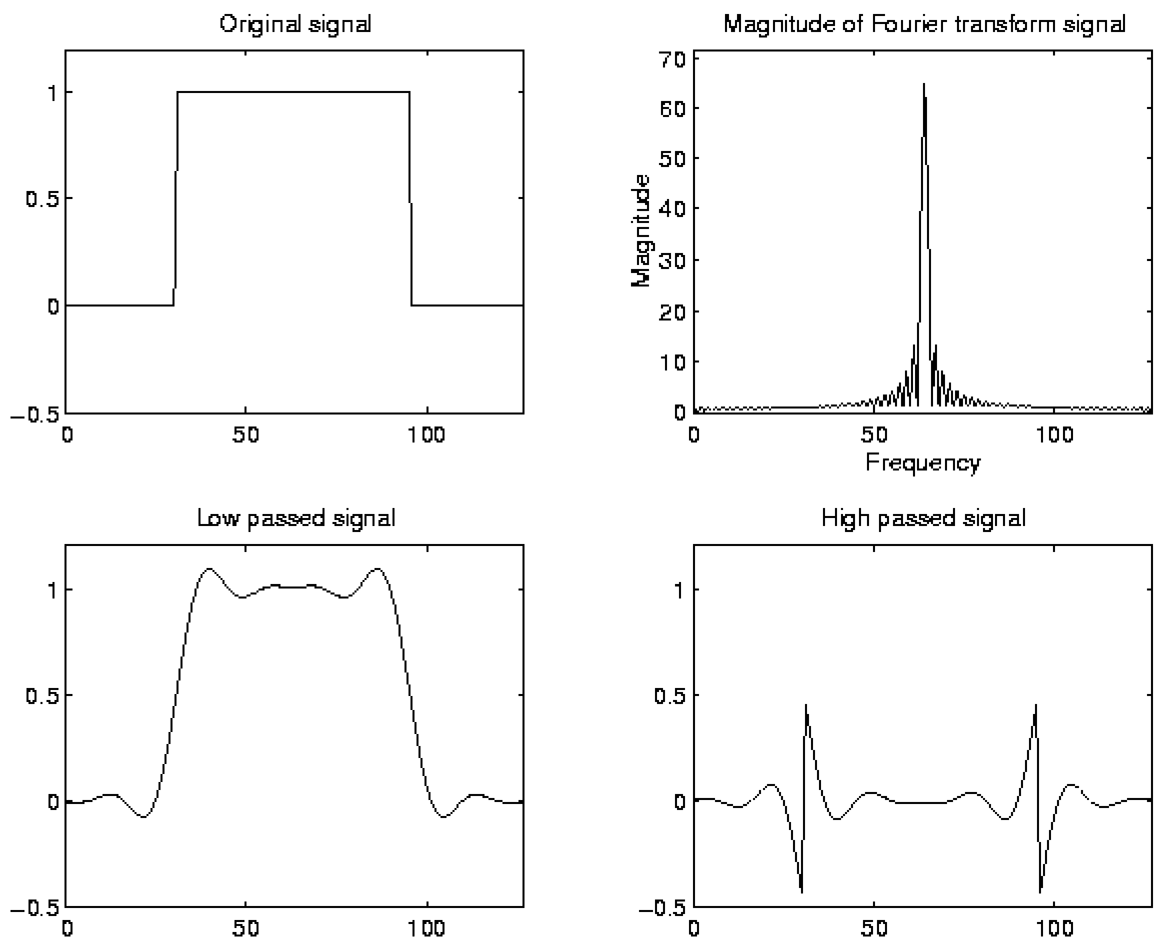


Image credit: Sandberg, UC Boulder

Background: 2D FT Example

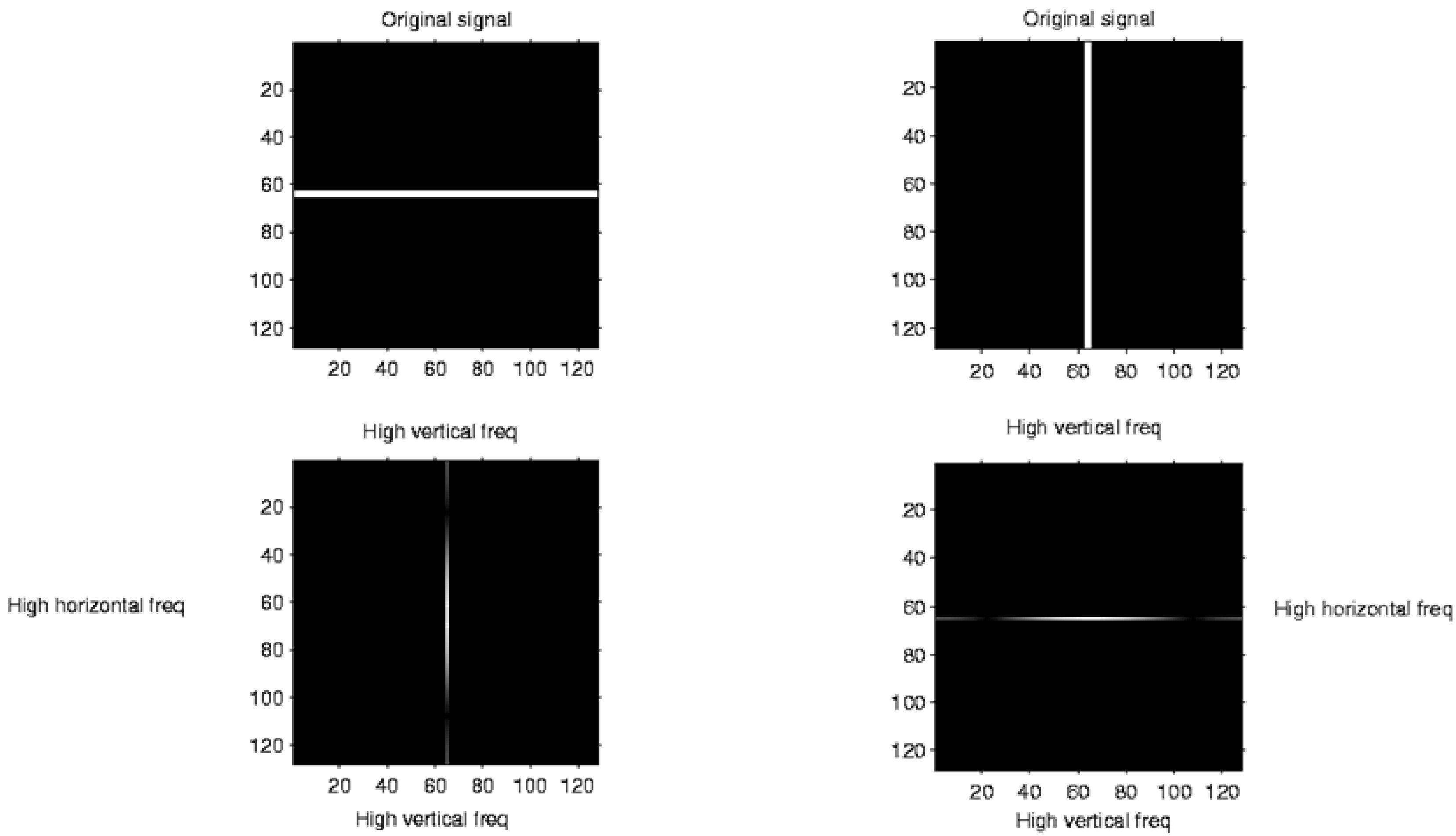


Image credit: Sandberg, UC Boulder

Background: high/low pass

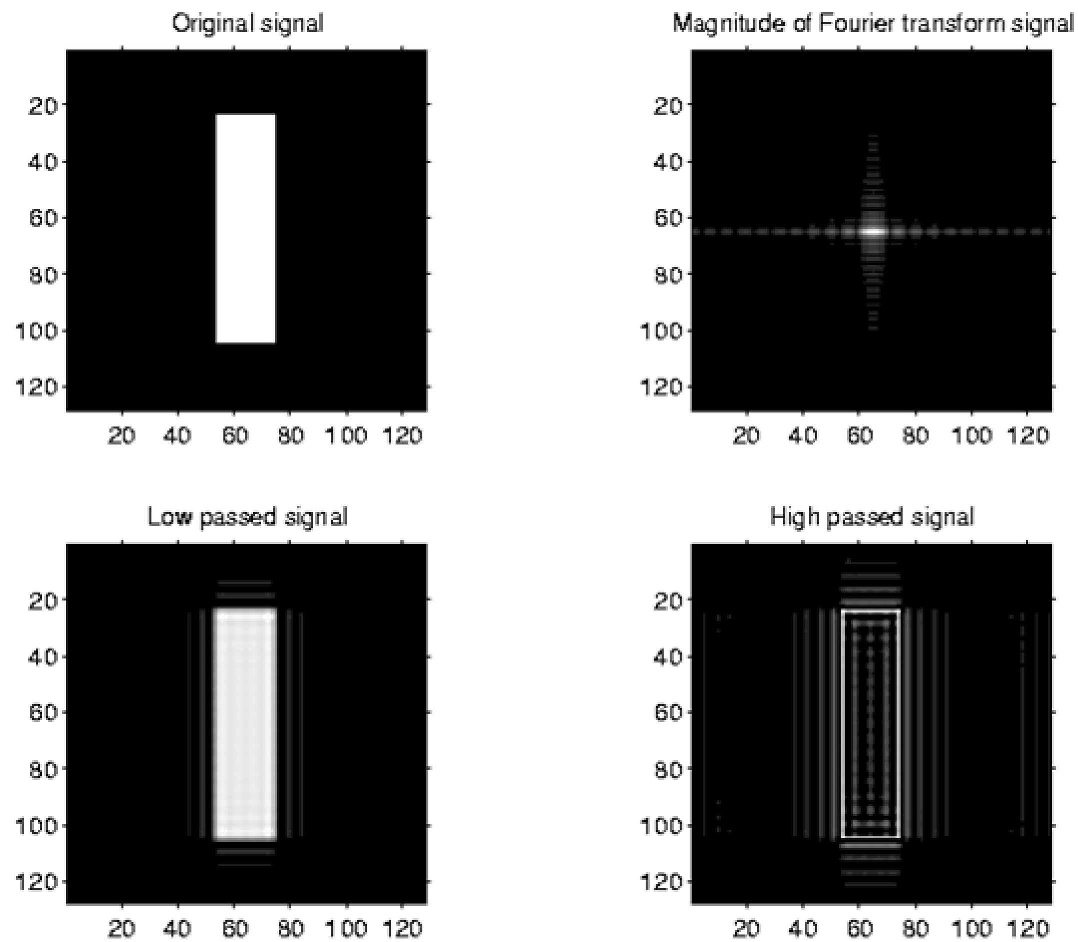


Image credit: Sandberg, UC Boulder

Background: more examples

- http://mathworld.wolfram.com/FourierTransform.html
- http://en.wikipedia.org/wiki/Discrete-time_Fourier_transform
- http://www.cs.unm.edu/~brayer/vision/fourier.html
- …

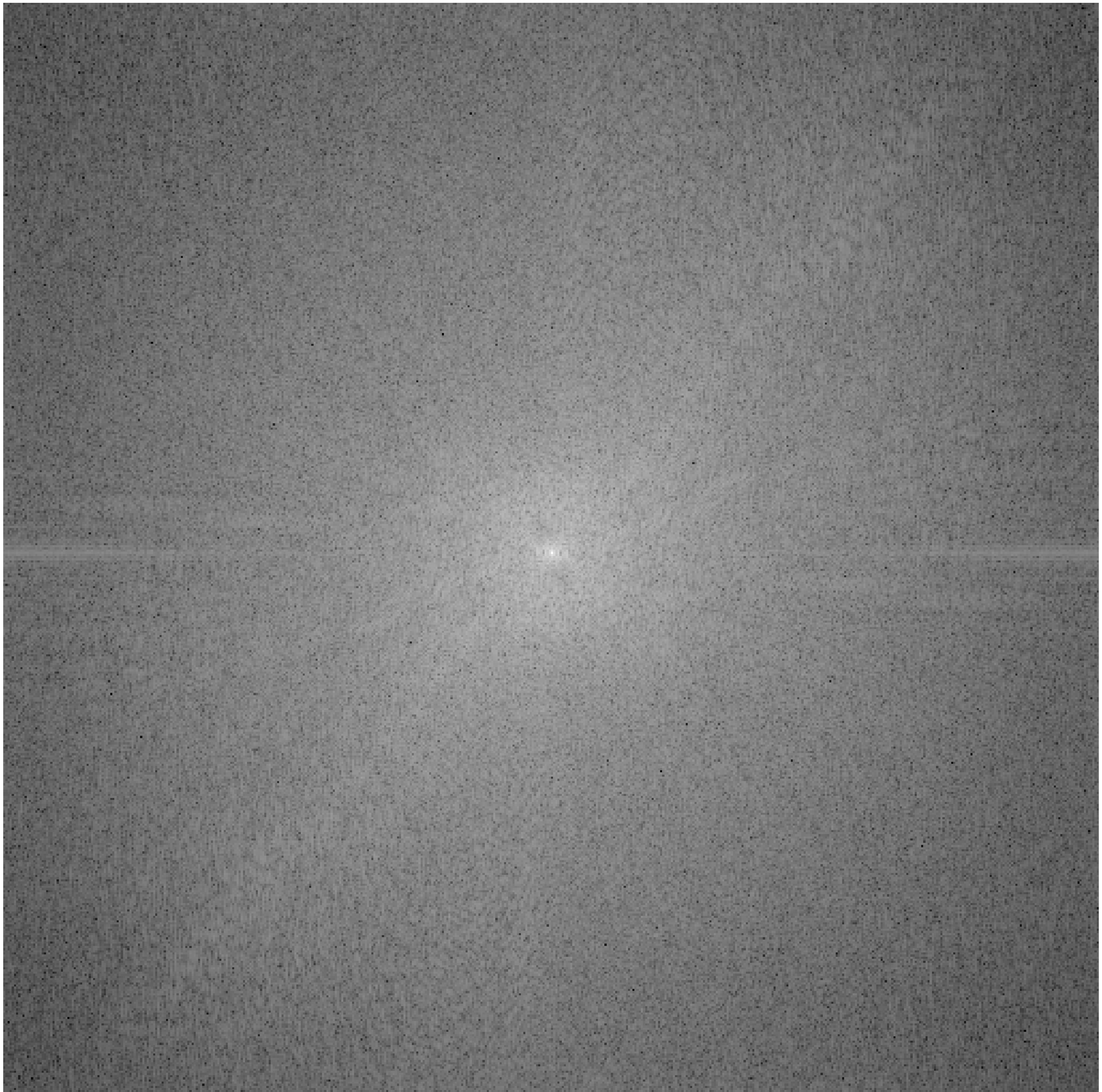
Magnitude vs Phase…?

- Mostly considered Magnitude spectra so far
- Sufficient for many vision methods:
  - high-pass/low-pass channel coding later in lecture.
  - simple edge detection, focus/defocus models
  - certain texture models
- May discard perceptually significant structure!

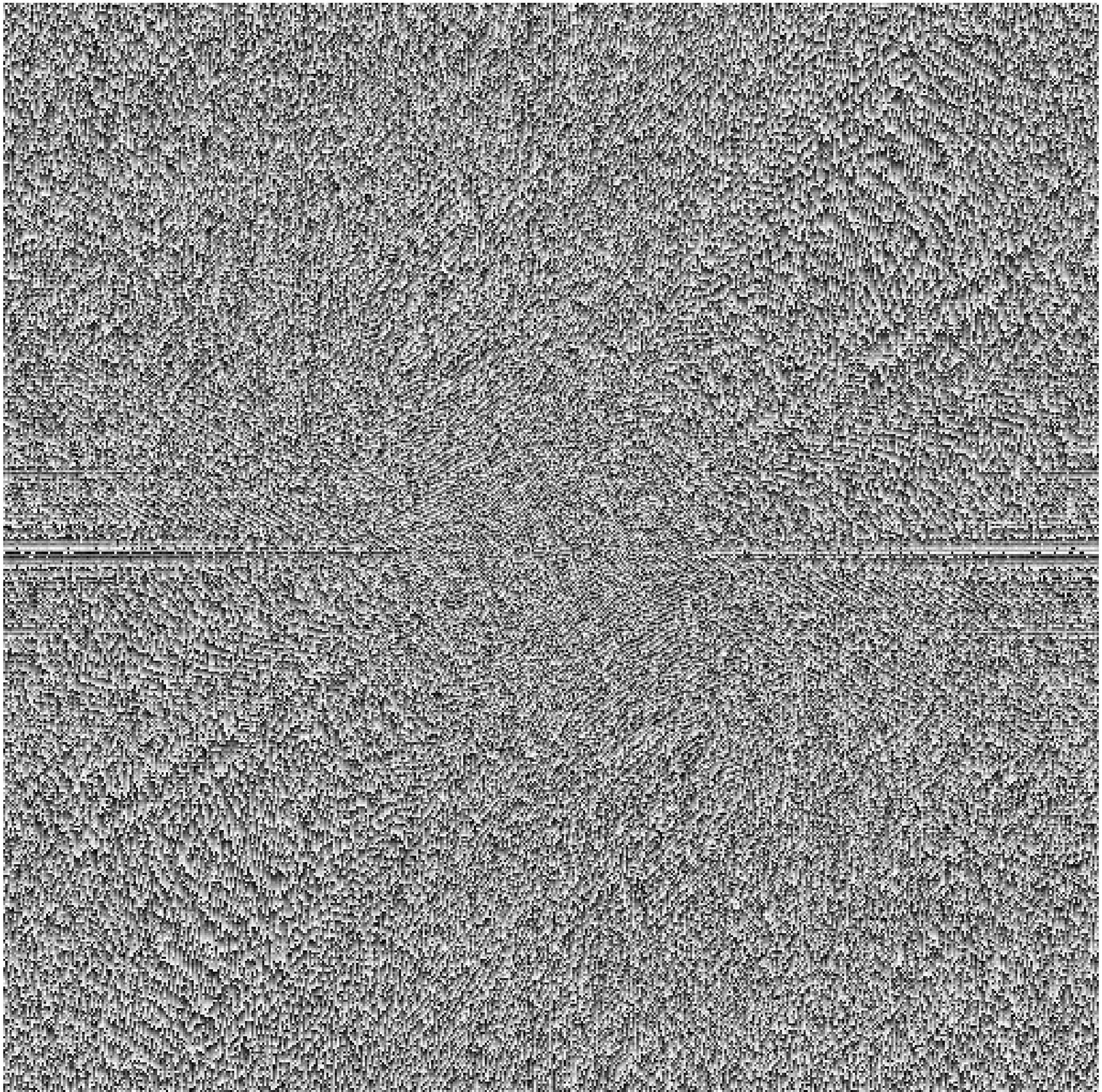
Phase and Magnitude

- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform
- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn't
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

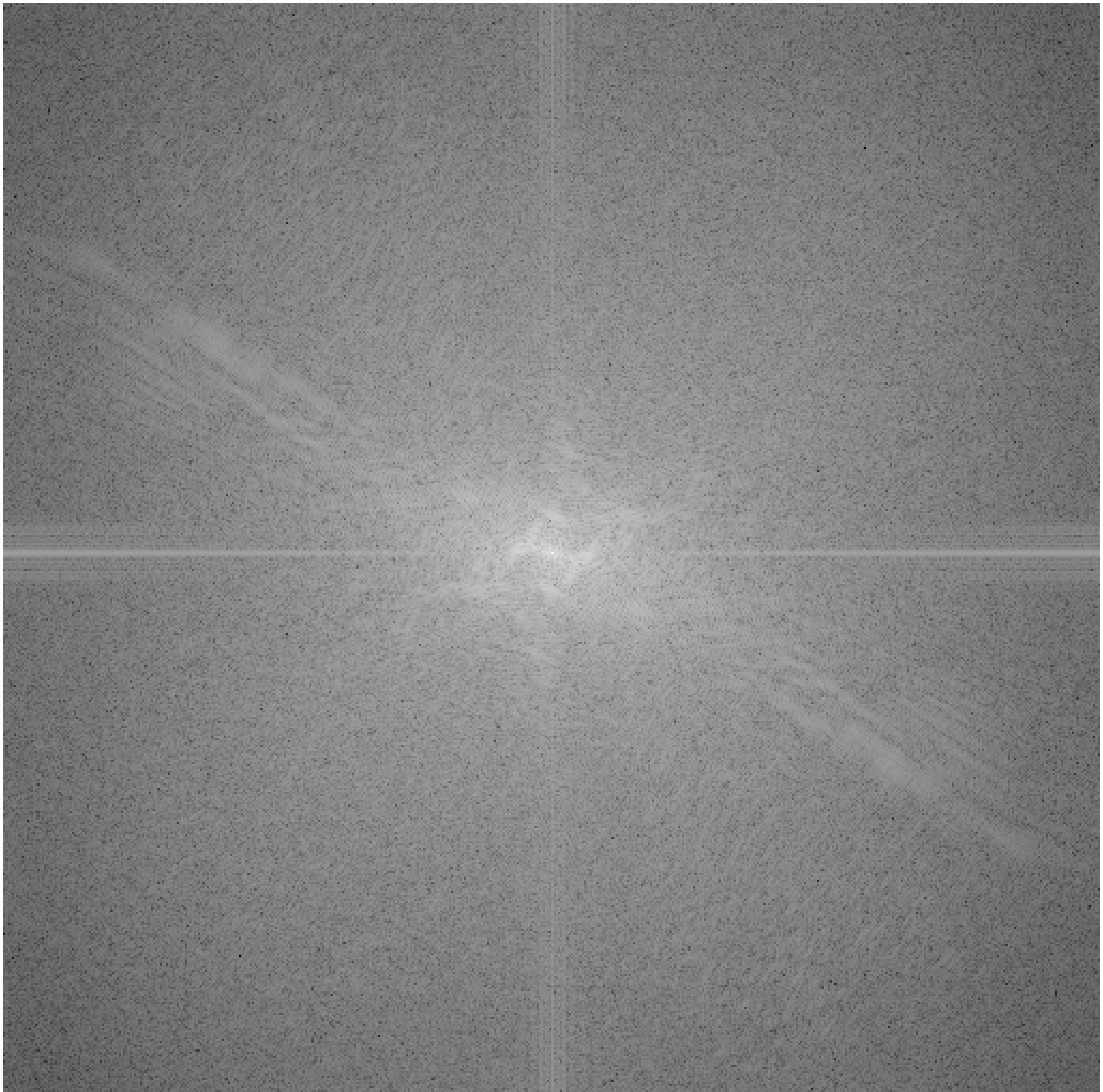
This is the magnitude transform of the cheetah pic



This is the phase transform of the cheetah pic

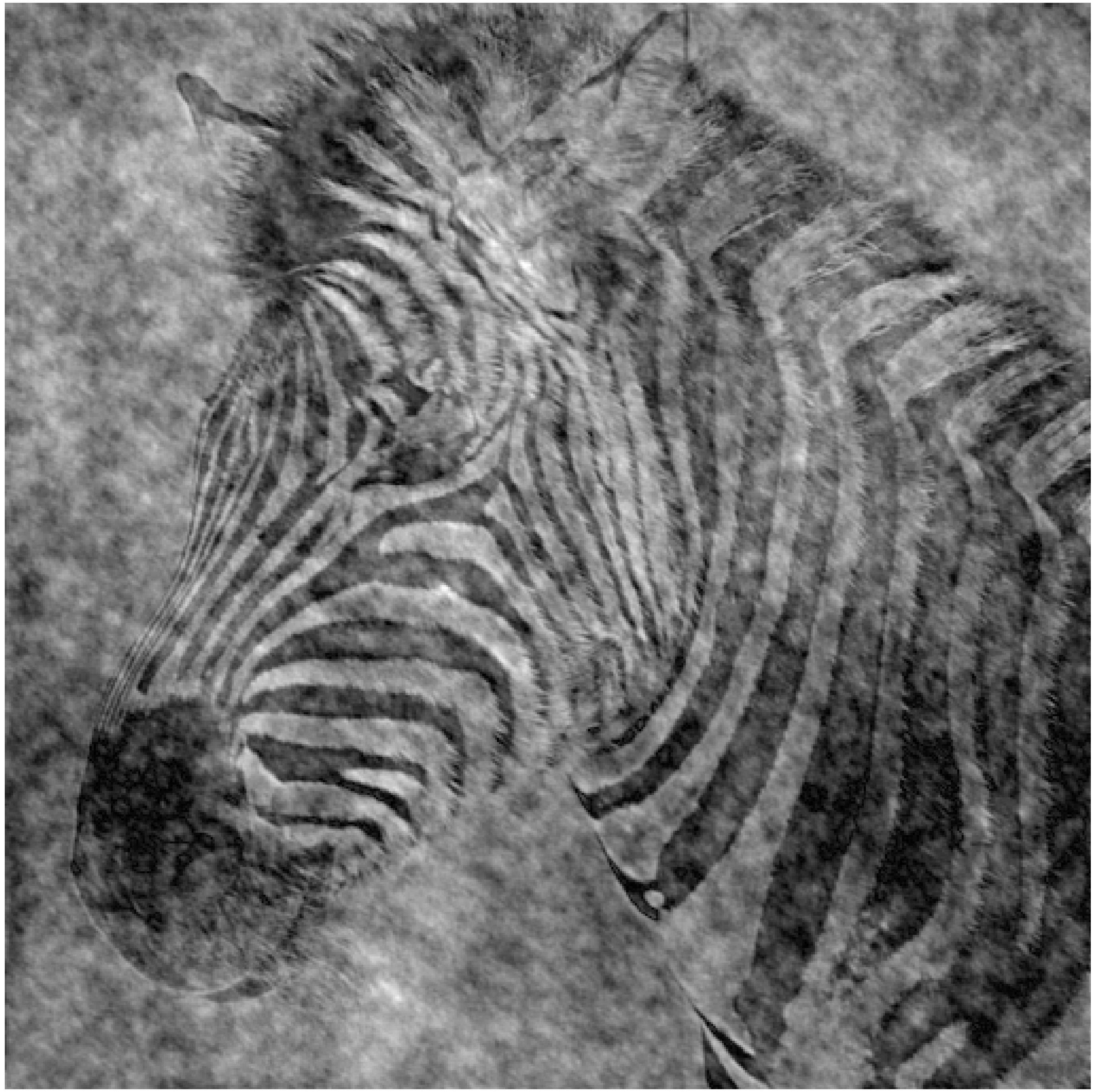


This is the magnitude transform of the zebra pic

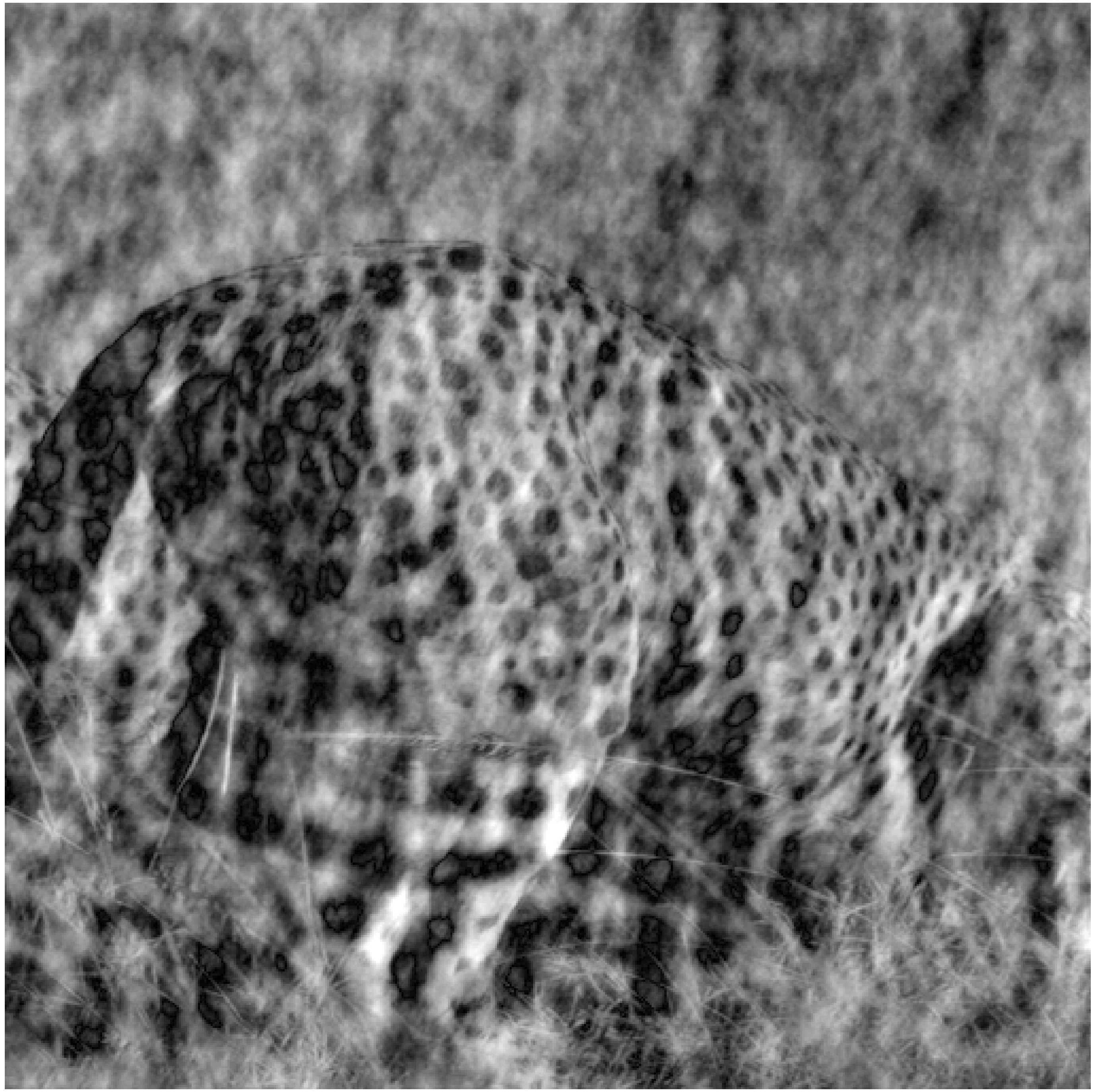


This is the phase transform of the zebra pic

Reconstruction with zebra phase, cheetah magnitude



Reconstruction with cheetah phase, zebra magnitude



1D D.O.G.

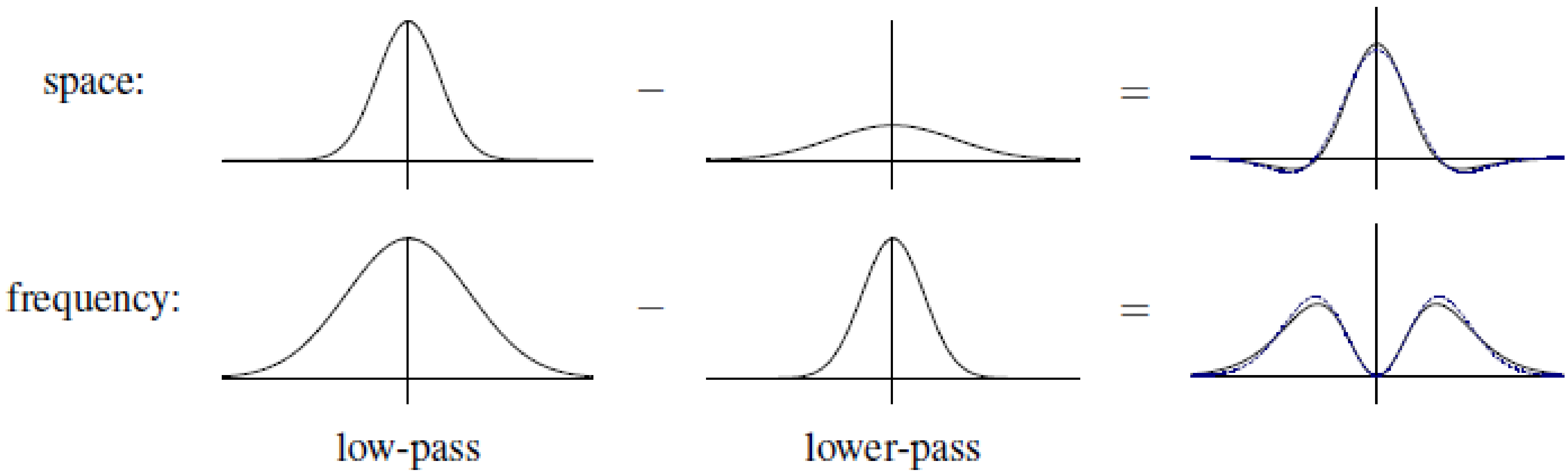
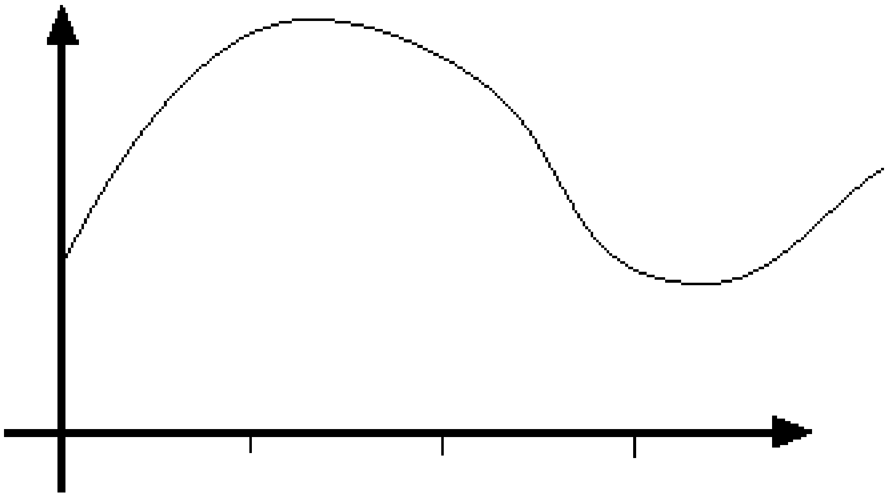
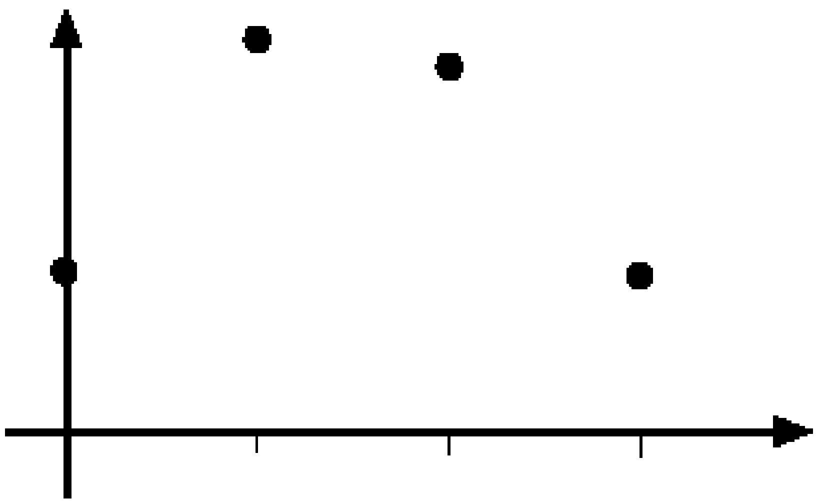


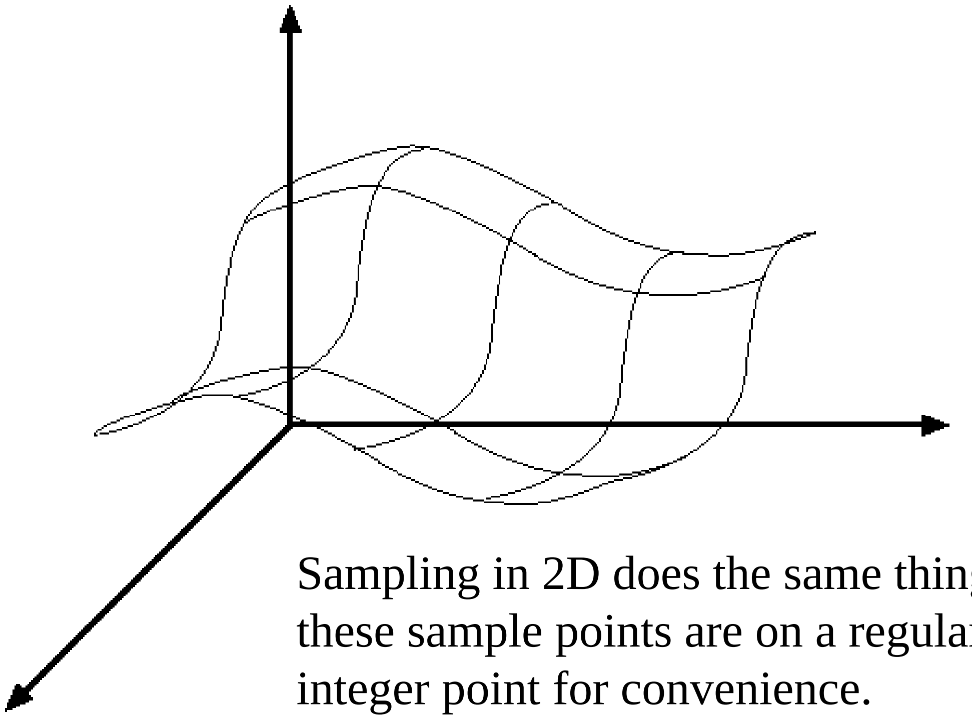
Figure 3.37: *The difference of two low-pass filters results in a band-pass filter. The dashed blue lines show the close fit to a half-octave Laplacian of Gaussian.*

Sampling and aliasing

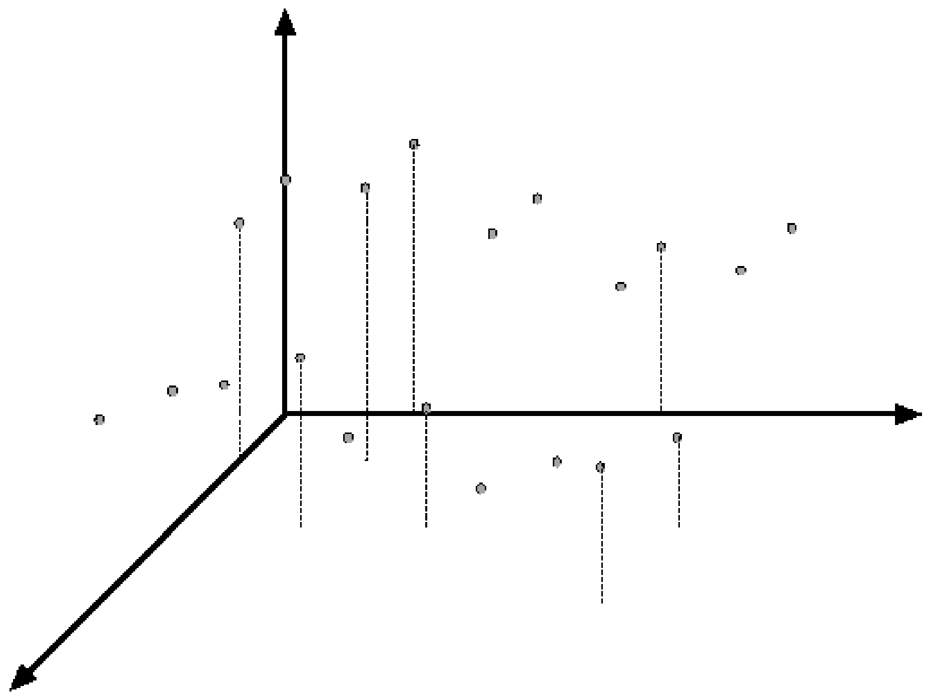


Sampling in 1D takes a continuous function and replaces it with a vector of values, consisting of the function’s values at a set of sample points. We’ll assume that these sample points are on a regular grid, and can place one at each integer for convenience.





Sampling in 2D does the same thing, only in 2D. We'll assume that these sample points are on a regular grid, and can place one at each integer point for convenience.

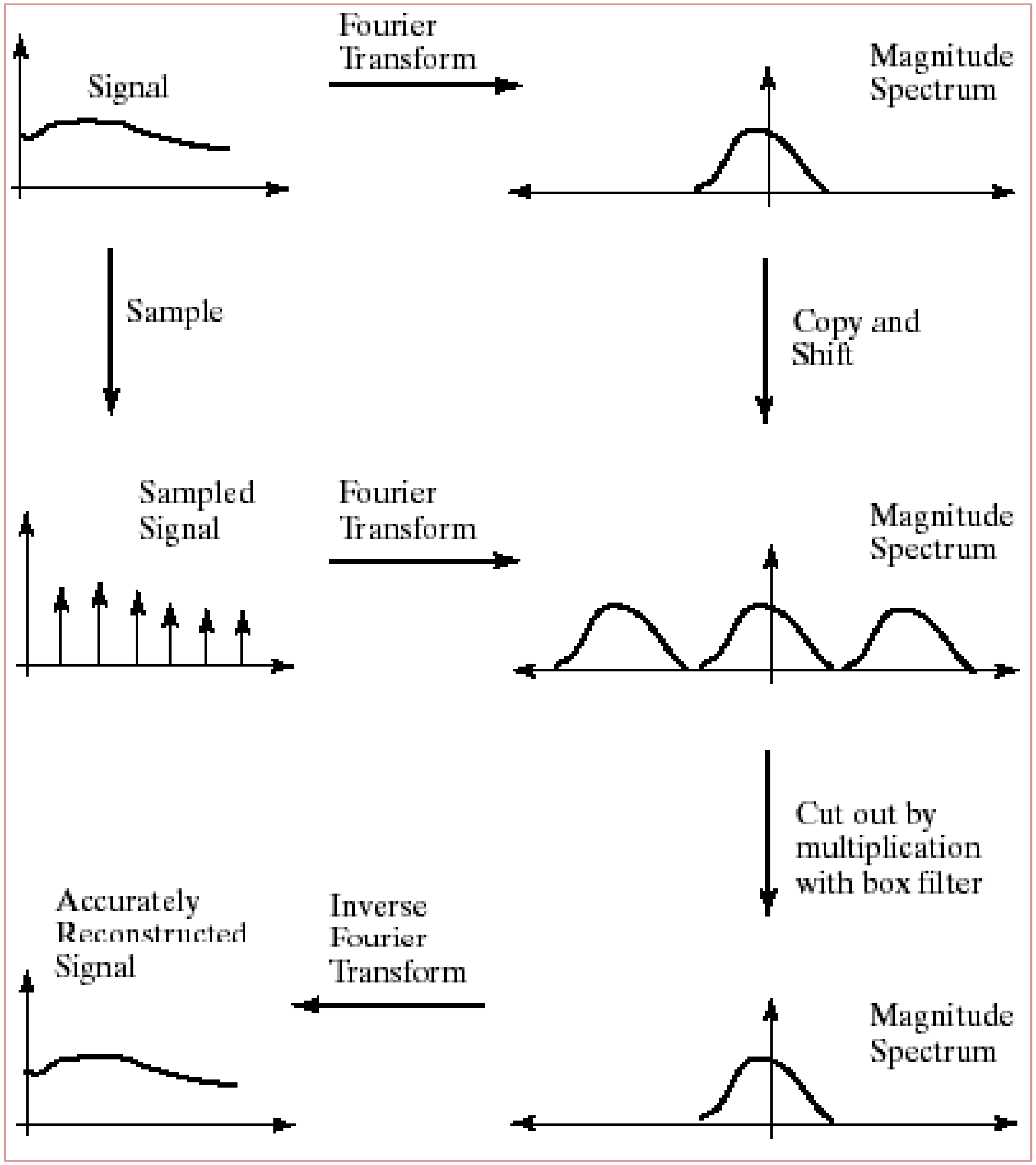


The Fourier transform of a sampled signal

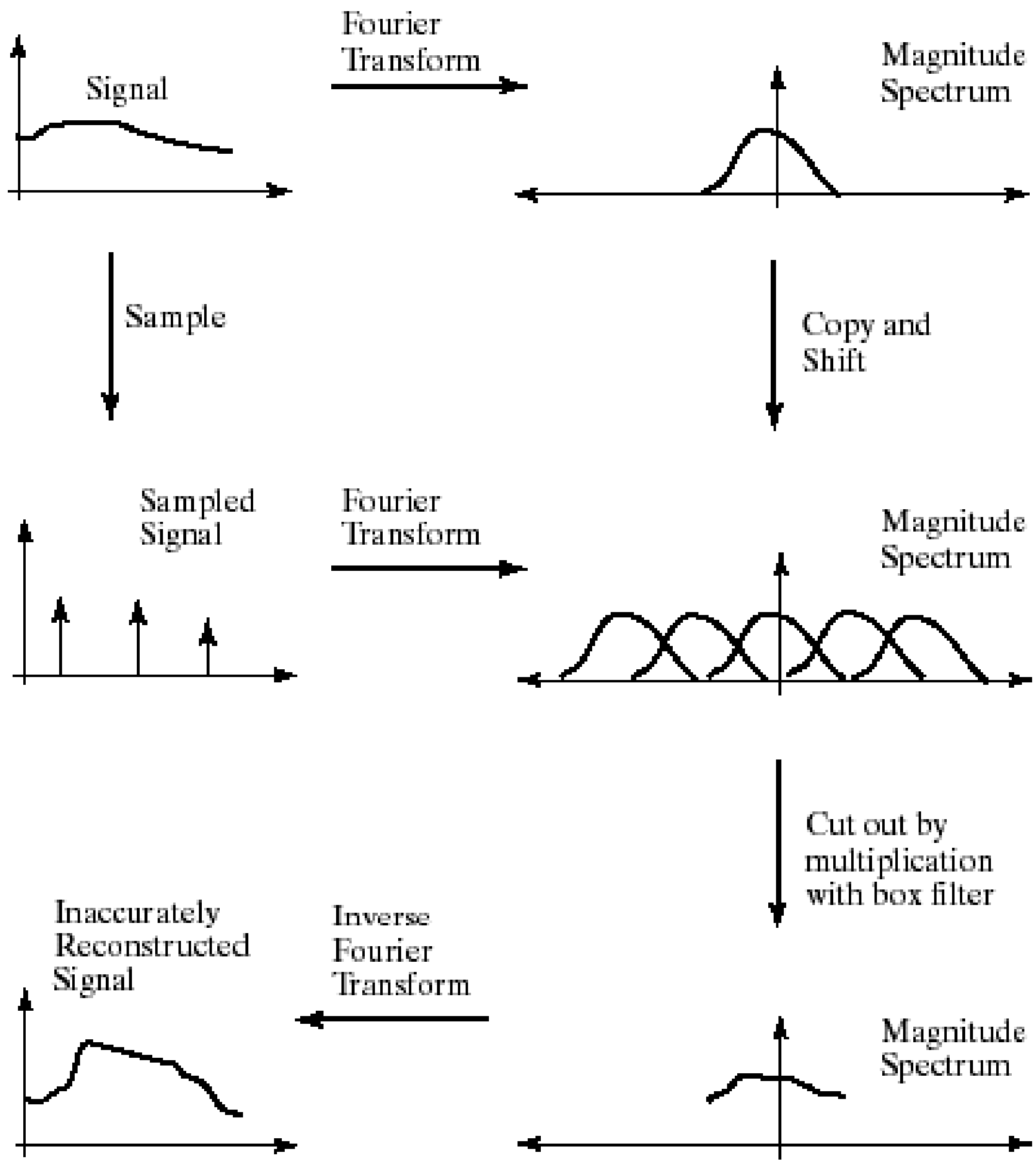
$$F\left(\mathrm{Sample}_{2\mathrm{D}}(f(x,y))\right) = F\left(f(x,y) \sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i, y-j) \right)$$

$$= F\left(f(x,y)\right) * F\left(\sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i, y-j) \right)$$

$$= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F\left(u-i, v-j\right)$$


Signal
Fourier Transform
Magnitude Spectrum
Sample
Copy and Shift
Sampled Signal
Fourier Transform
Magnitude Spectrum
Cut out by multiplication with box filter
Accurately Reconstructed Signal
Inverse Fourier Transform
Magnitude Spectrum

Freeman

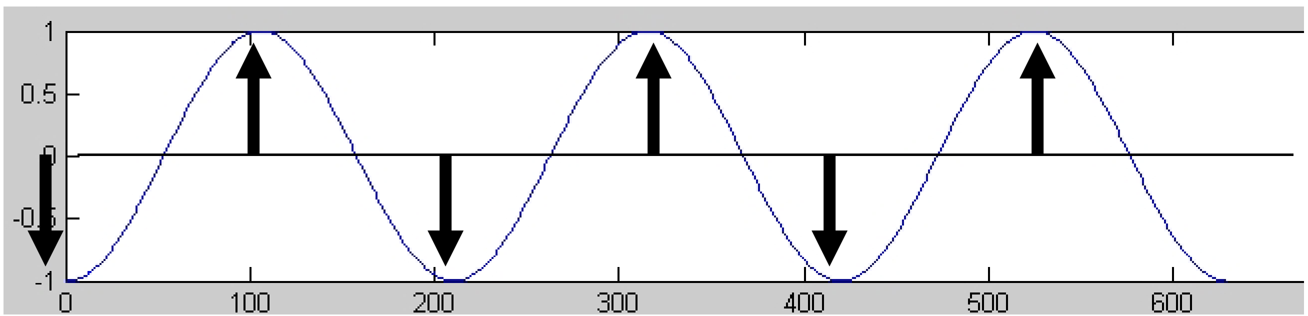

Signal
Fourier
Transform
Magnitude
Spectrum
Sample
Copy and
Shift
Sampled
Signal
Fourier
Transform
Magnitude
Spectrum
Cut out by
multiplication
with box filter
Inaccurately
Reconstructed
Signal
Inverse
Fourier
Transform
Magnitude
Spectrum

Freeman

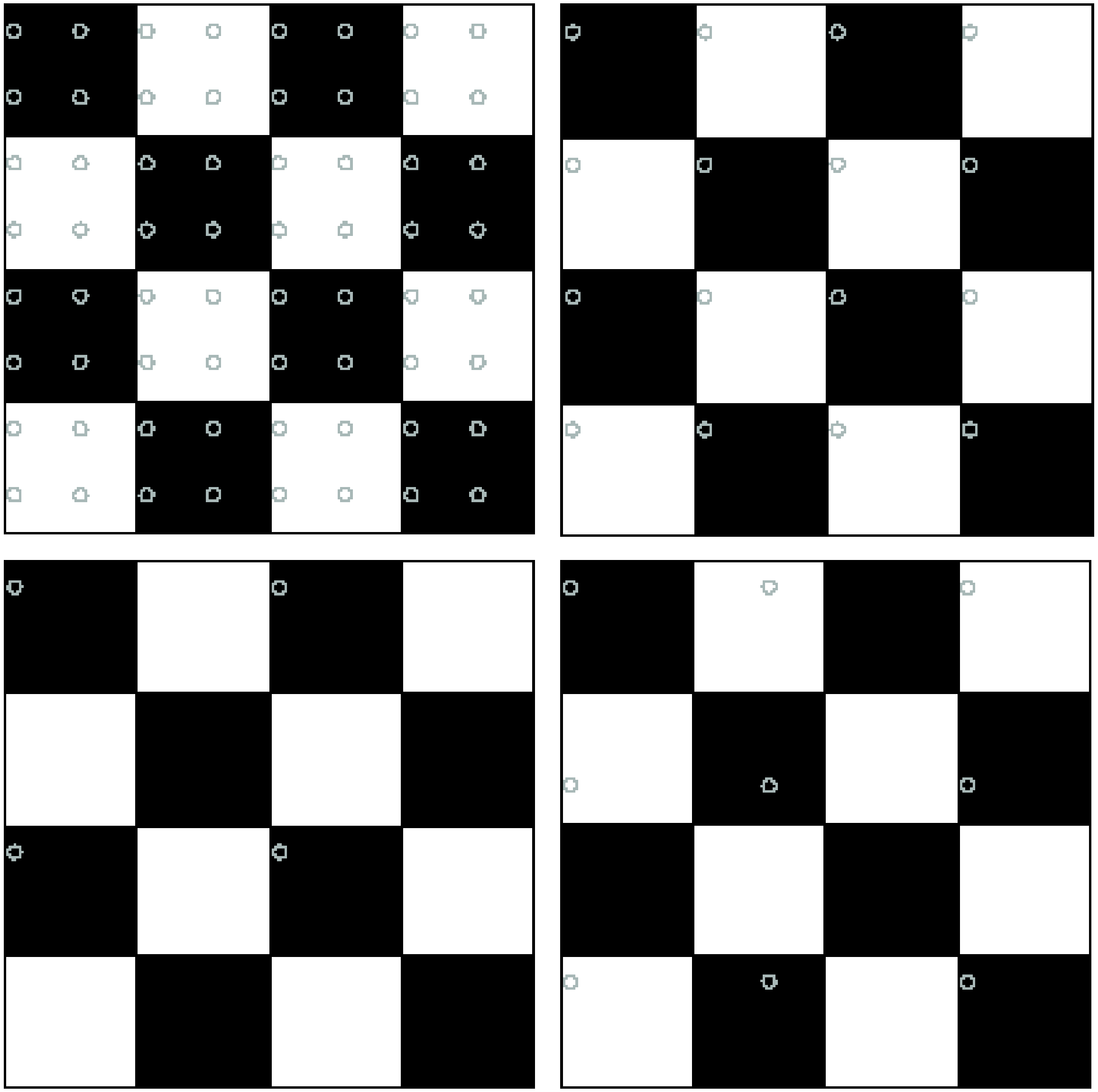
Aliasing

- Can't shrink an image by taking every second pixel
- If we do, characteristic errors appear
  - In the next few slides
  - Typically, small phenomena look bigger; fast phenomena can look slower
  - Common phenomenon
    - Wagon wheels rolling the wrong way in movies
    - Checkerboards misrepresented in ray tracing

Space domain explanation of Nyquist sampling



You need to have at least two samples per sinusoid cycle to represent that sinusoid.



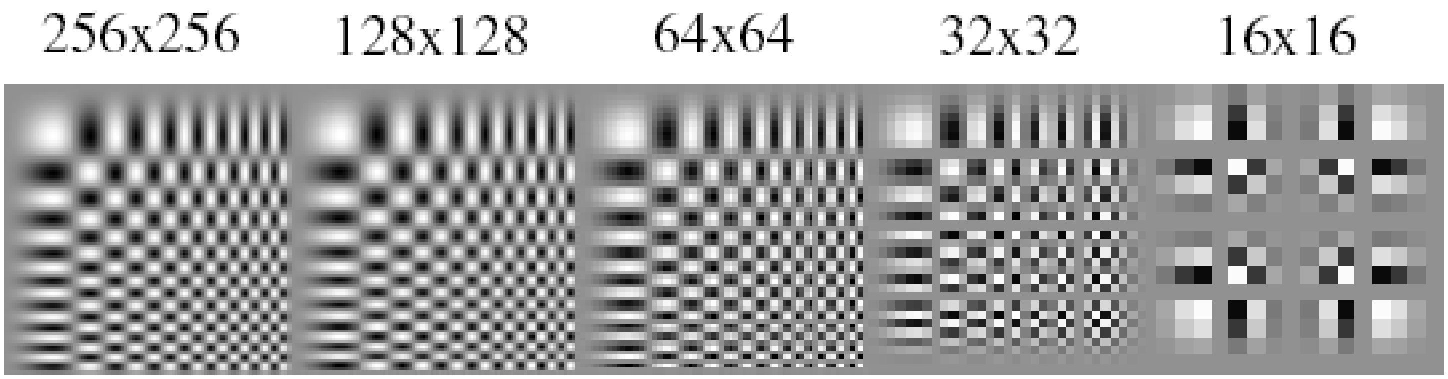
Resample the checkerboard by taking one sample at each circle. In the case of the top left board, new representation is reasonable. Top right also yields a reasonable representation. Bottom left is all black (dubious) and bottom right has checks that are too big.

Smoothing as low-pass filtering

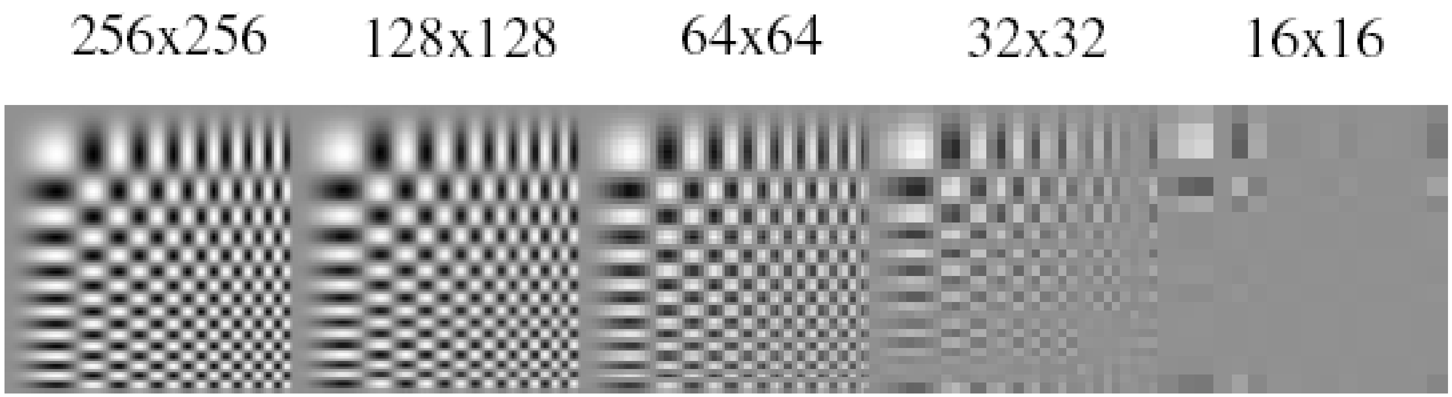
- The message of the FT is that high frequencies lead to trouble with sampling.
- Solution: suppress high frequencies before sampling
  – multiply the FT of the signal with something that suppresses high frequencies
  – or convolve with a low-pass filter
- A filter whose FT is a box is bad, because the filter kernel has infinite support
- Common solution: use a Gaussian
  – multiplying FT by Gaussian is equivalent to convolving image with Gaussian.

Freeman

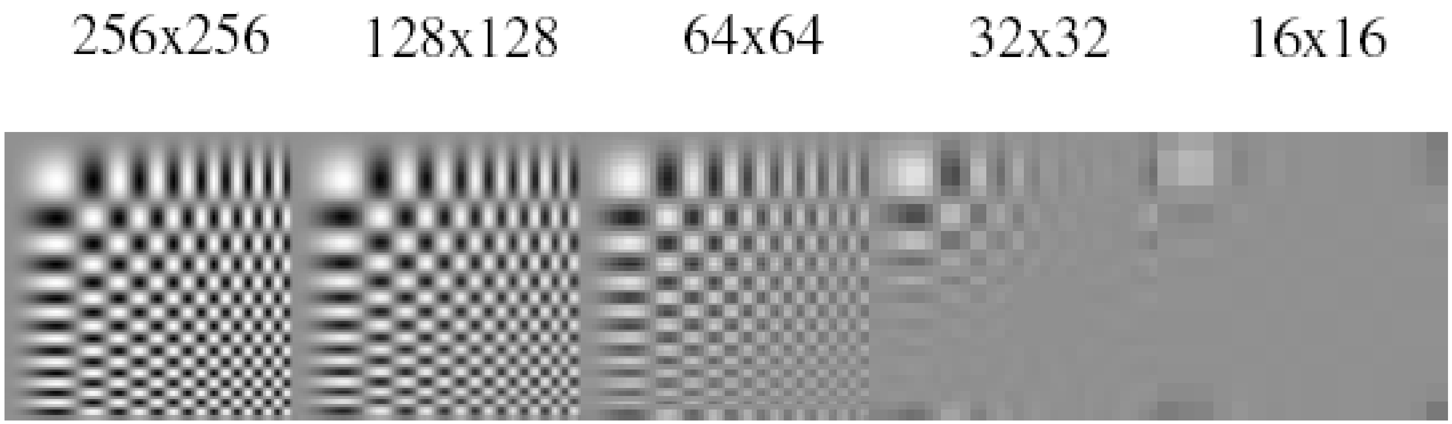
Sampling without smoothing. Top row shows the images, sampled at every second pixel to get the next.



Sampling with smoothing. Top row shows the images. We
get the next image by smoothing the image with a Gaussian with sigma 1 pixel,
then sampling at every second pixel to get the next.

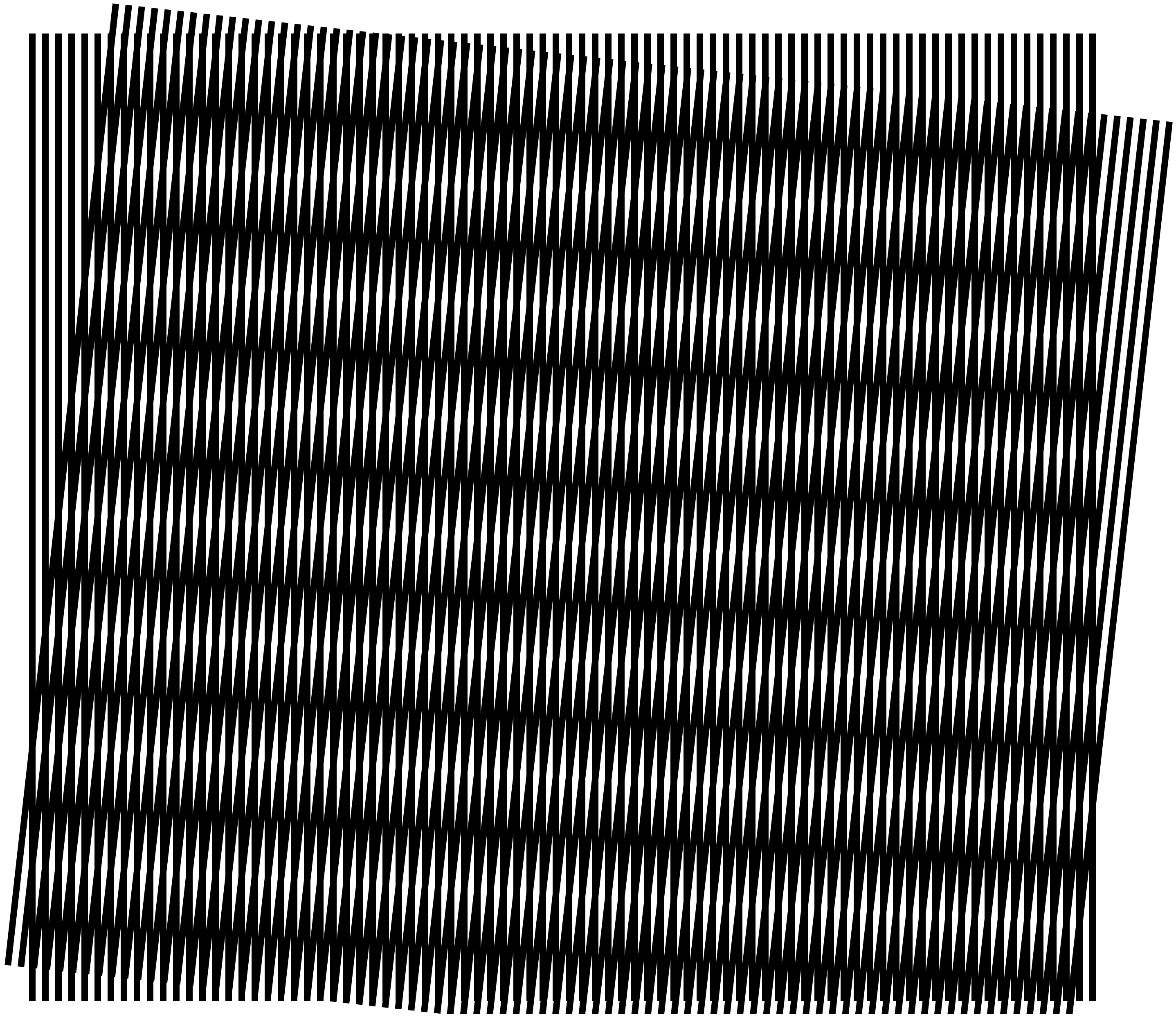


Sampling with smoothing. Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1.4 pixels, then sampling at every second pixel to get the next.



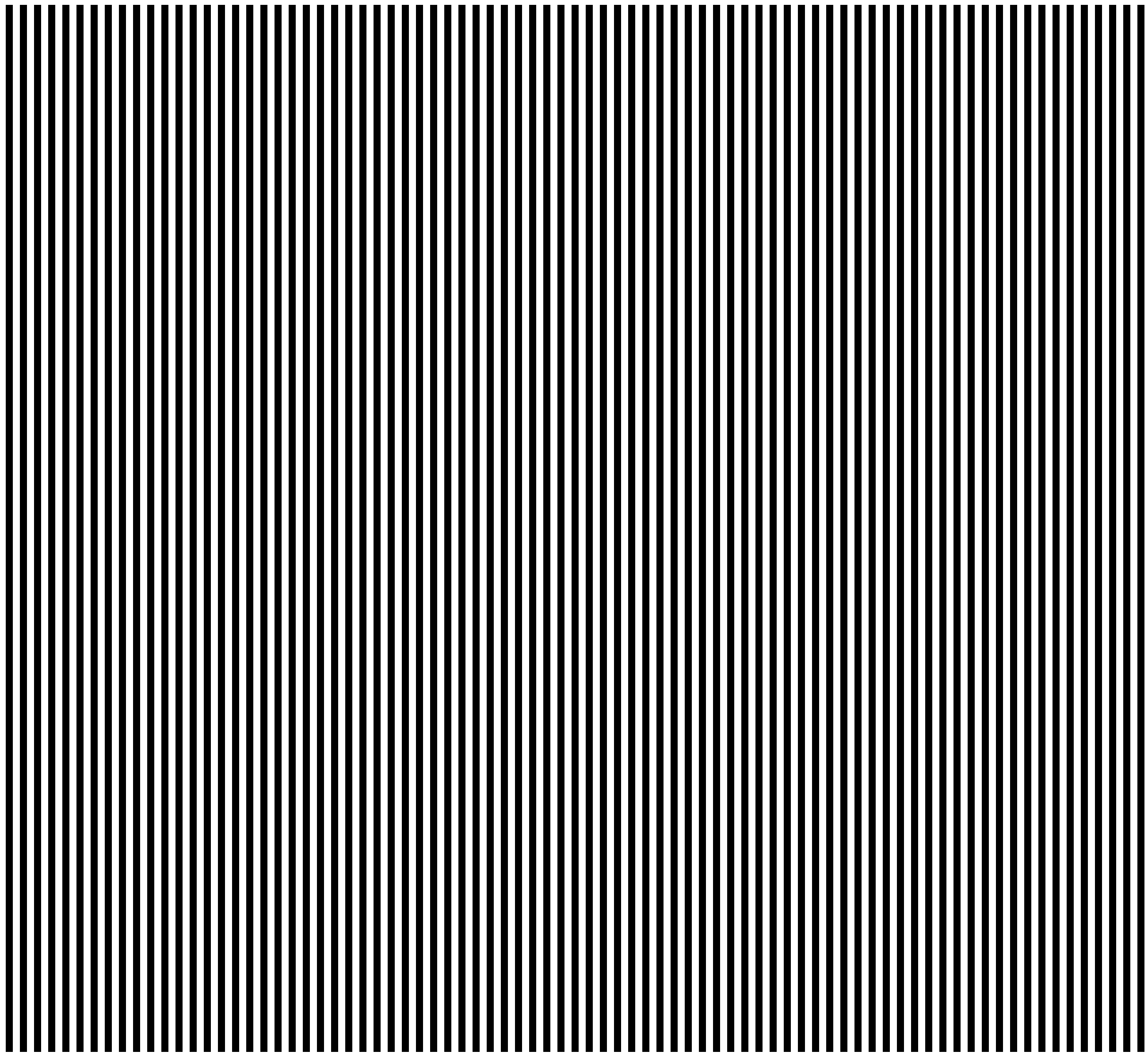
Sampling example

Analyze crossed gratings…



Sampling example

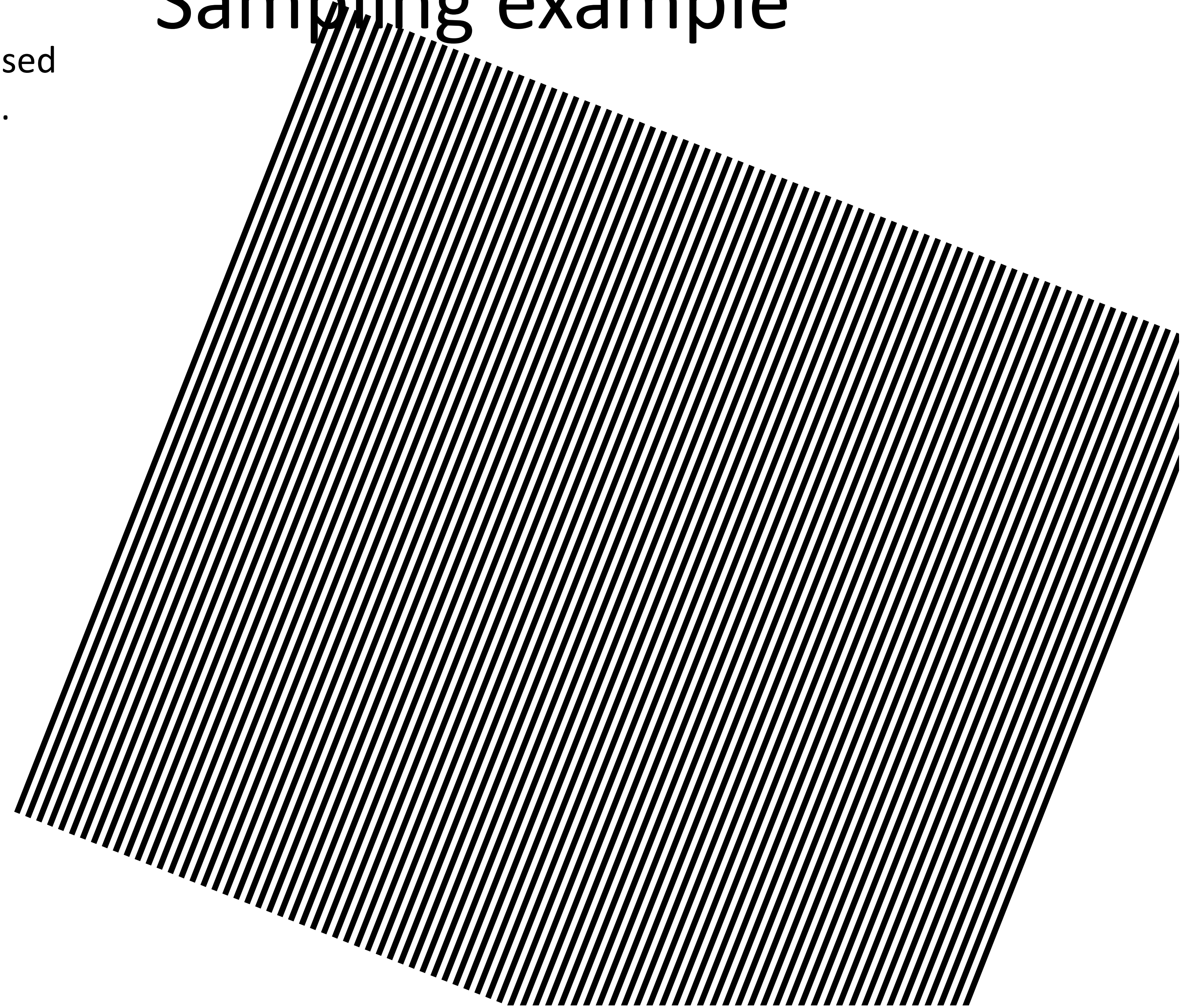
Analyze crossed gratings…



Freeman

Sampling example

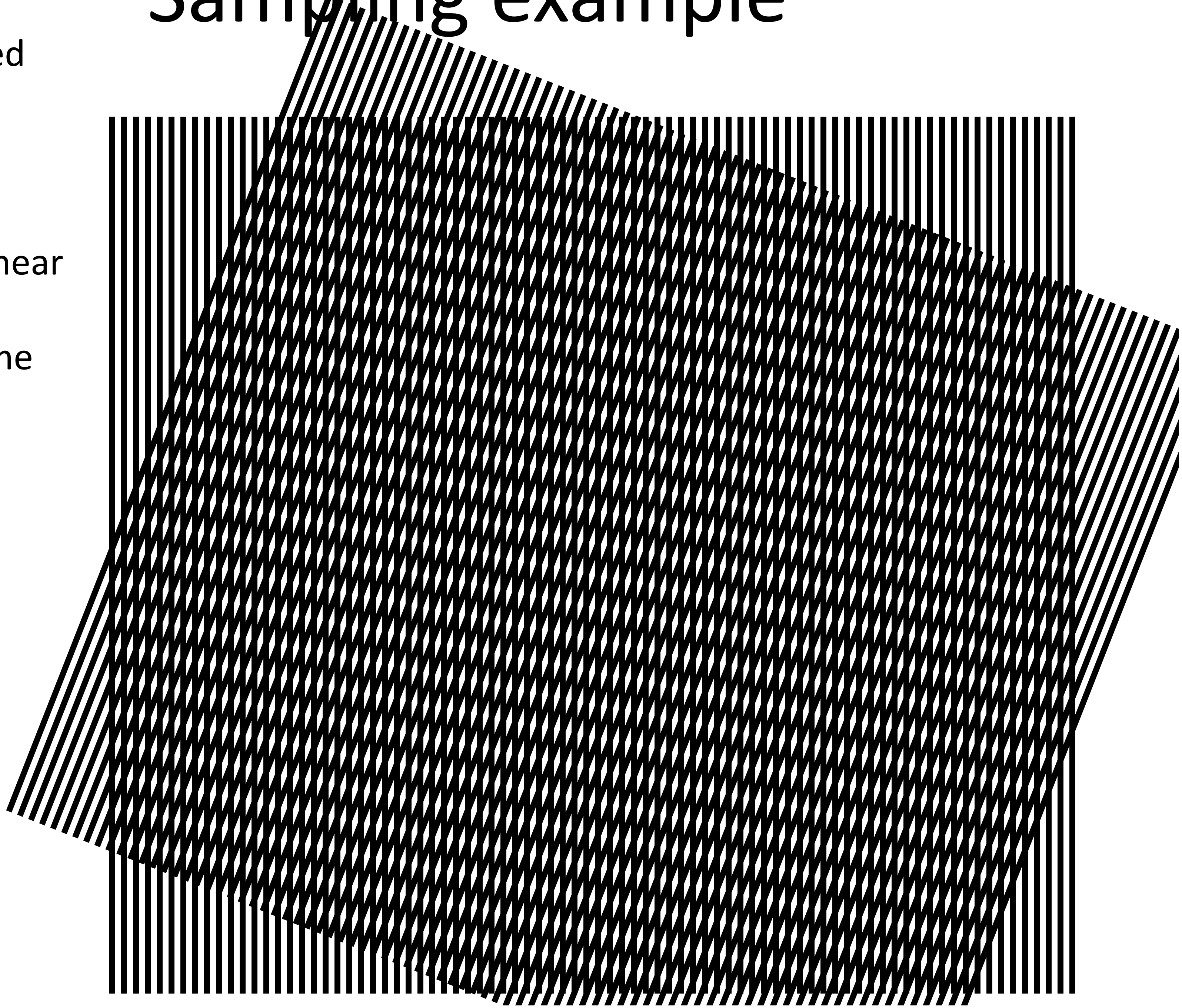
Analyze crossed gratings…



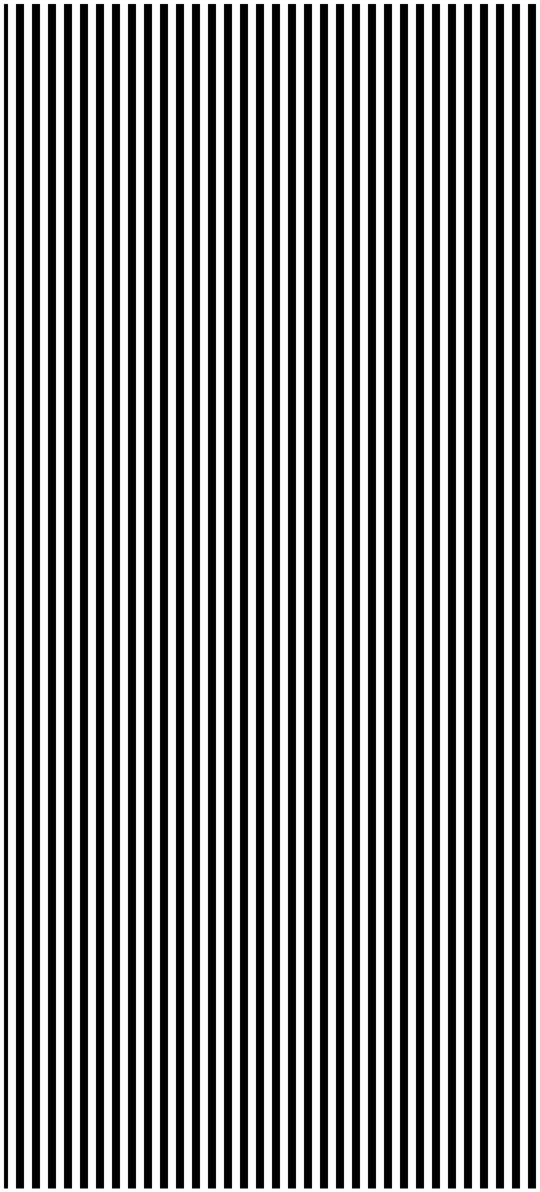
Sampling example

Analyze crossed gratings…

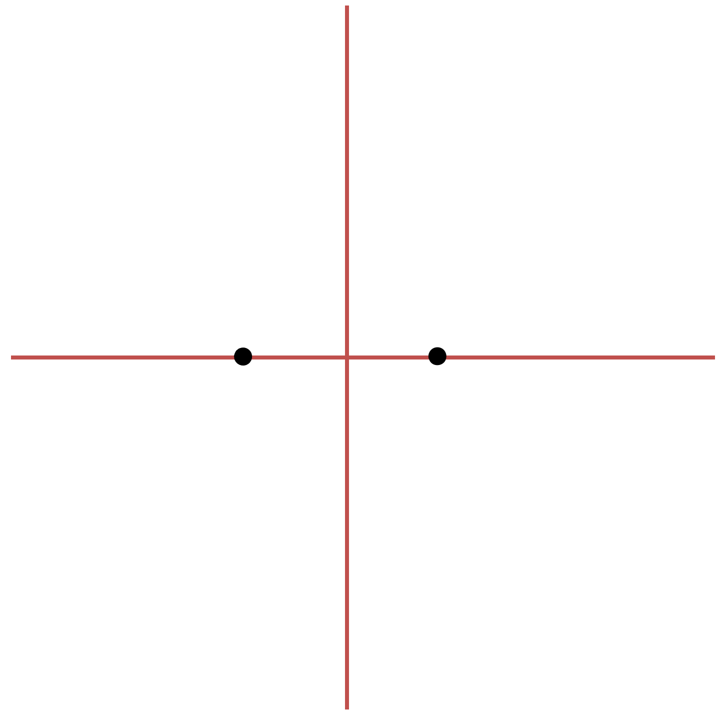
Where does perceived near horizontal grating come from?



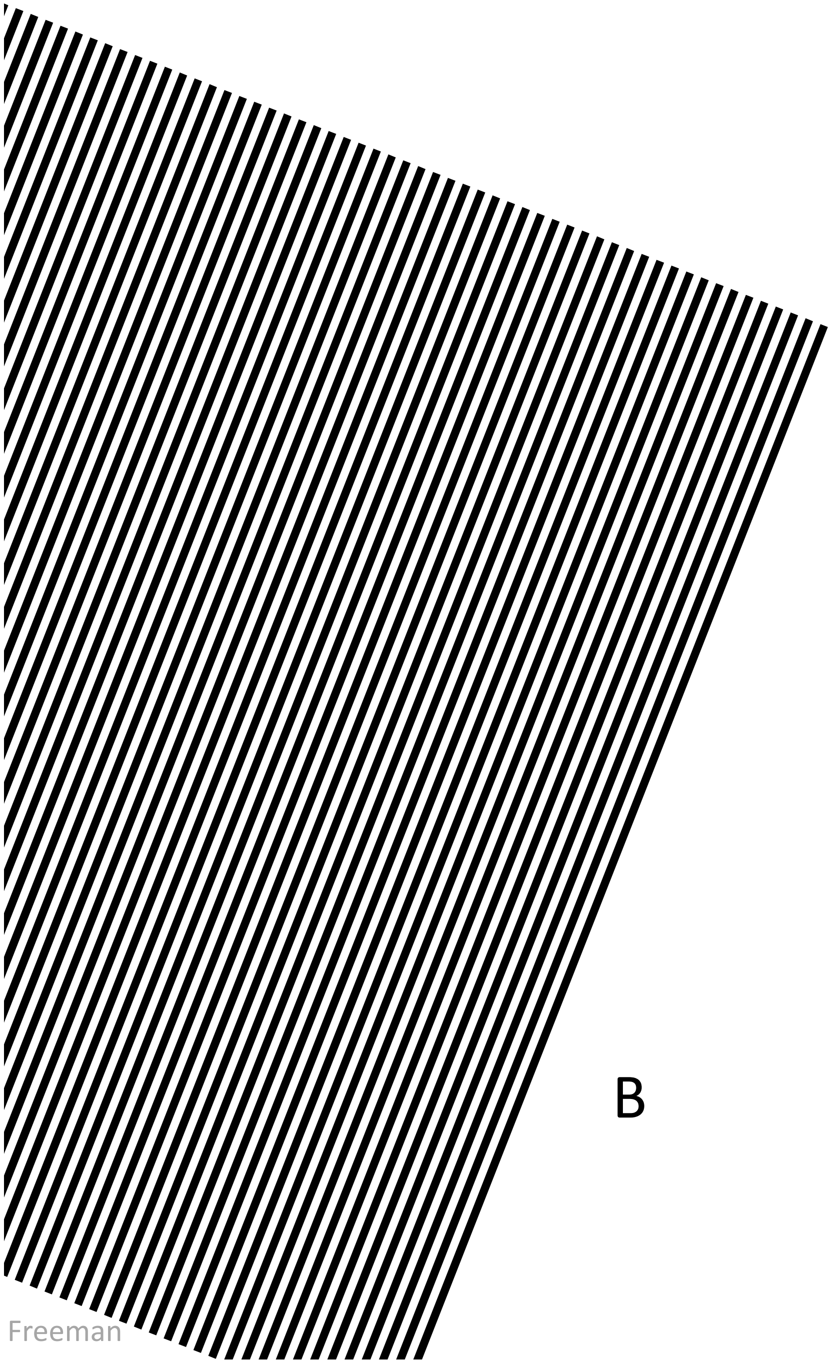
Freeman



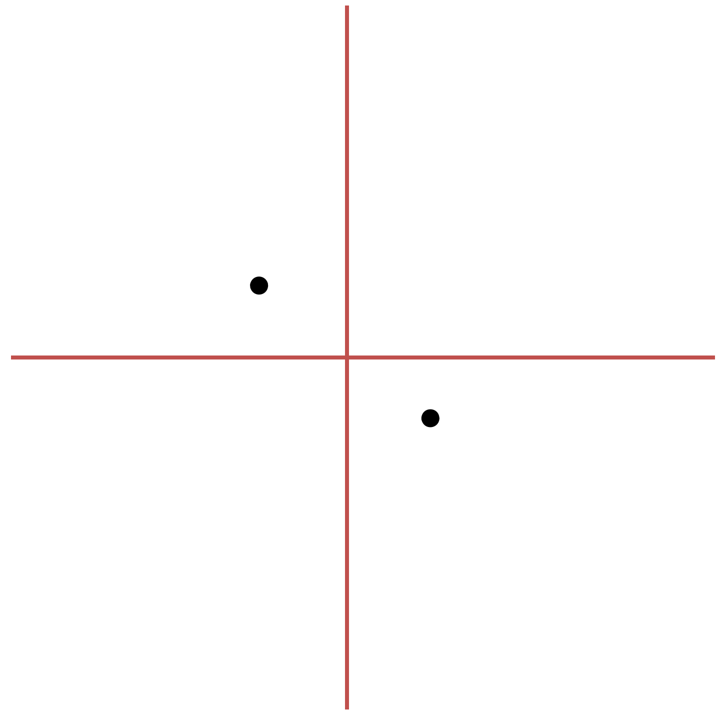
A



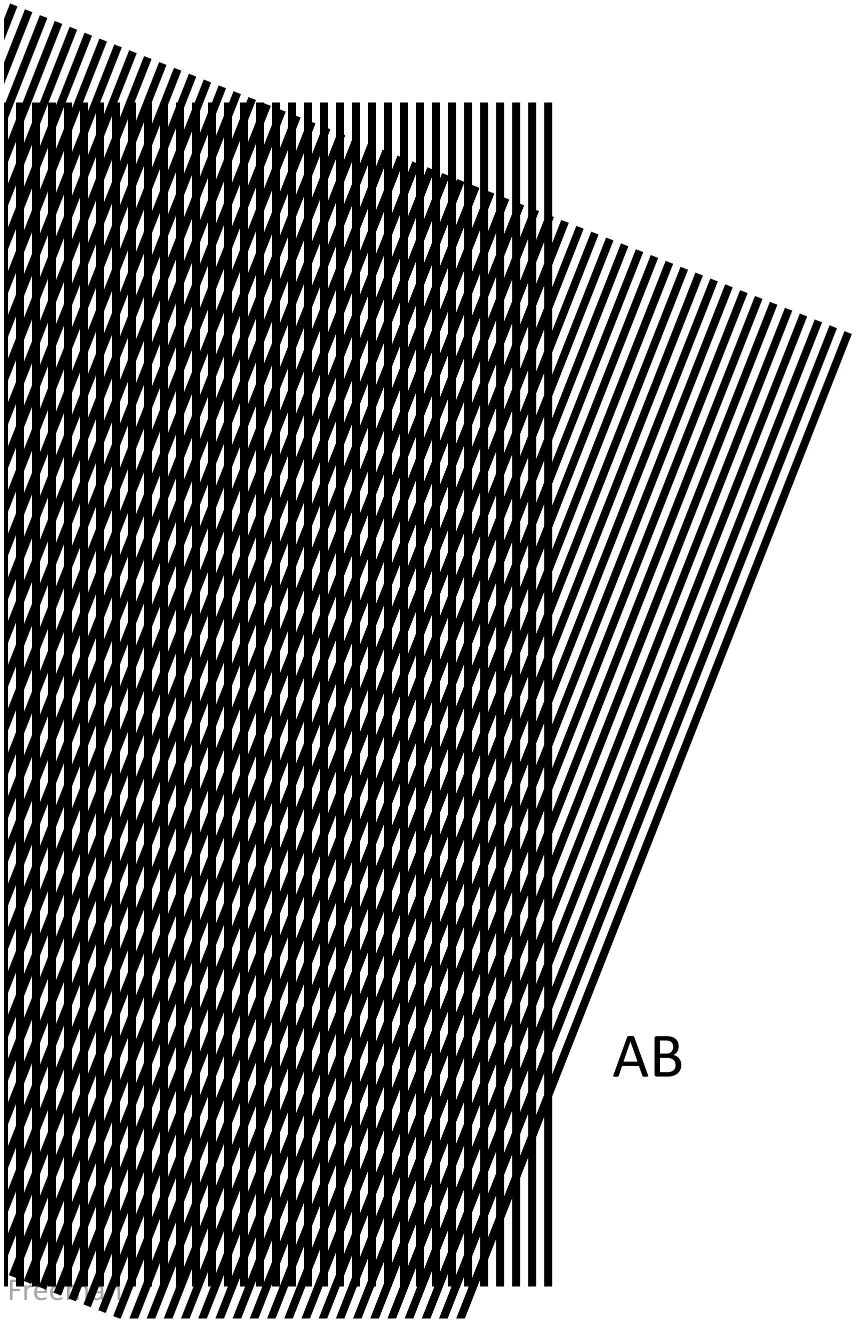
F(A)



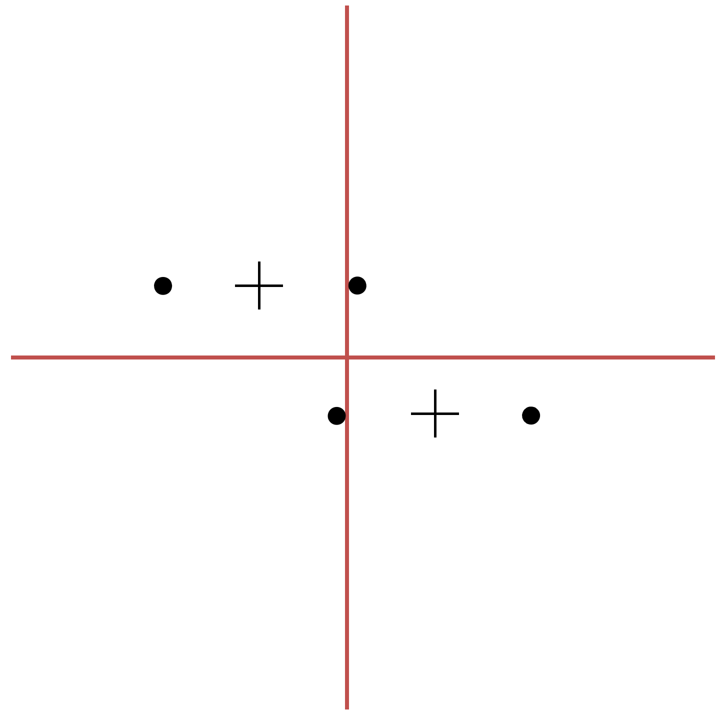
B



F(B)



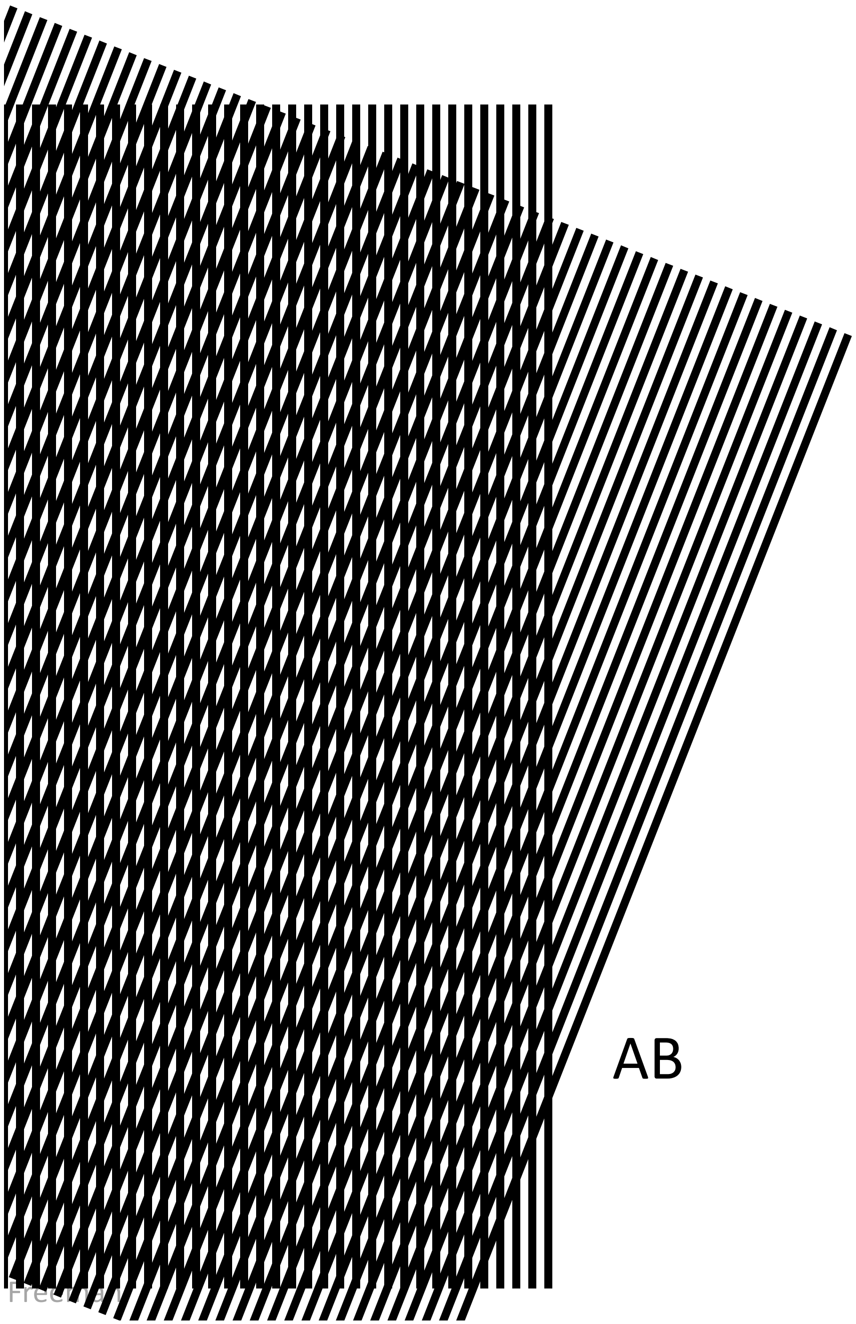
AB



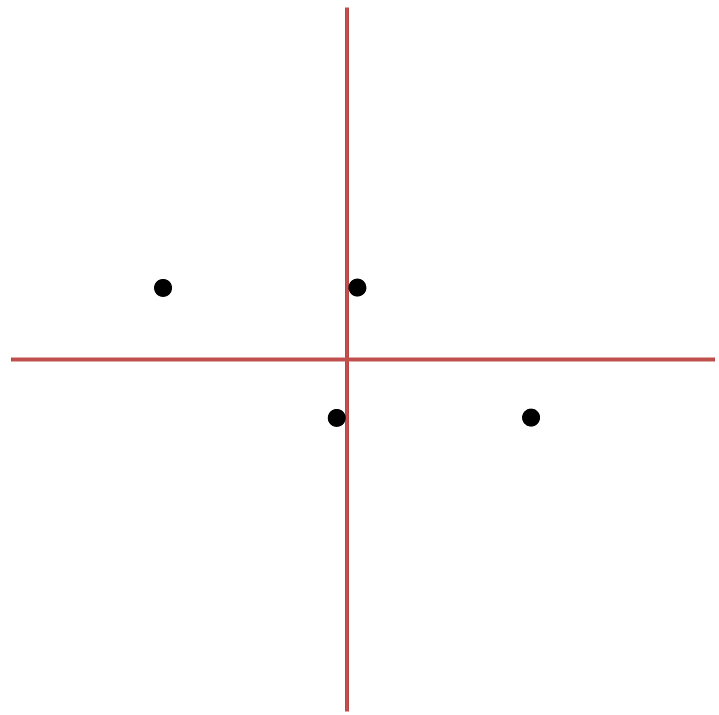
F(A) * F(B)

(using Szeliski notation, ‘*’ is convolution)

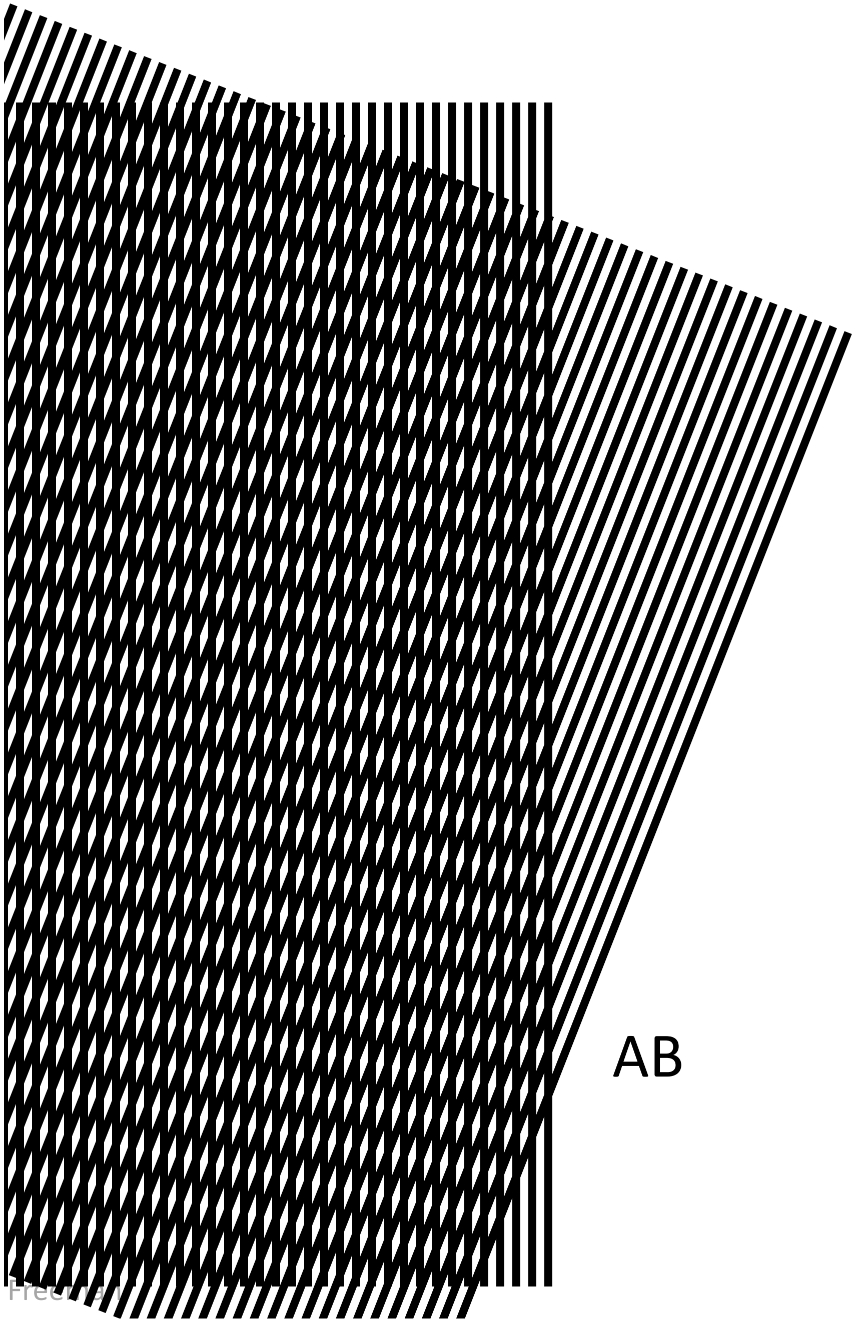
Freeman



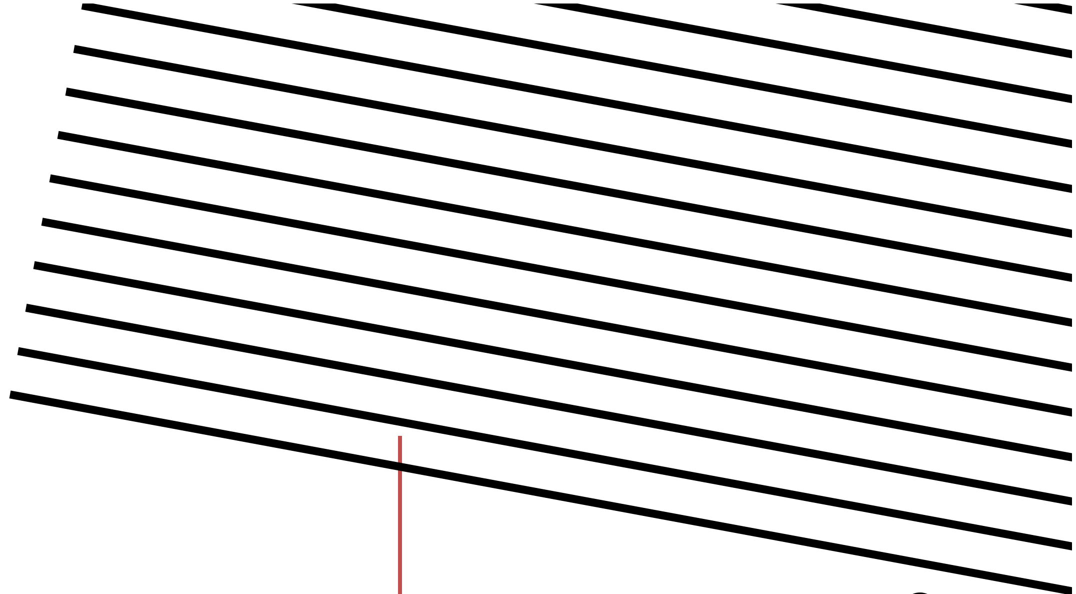
AB



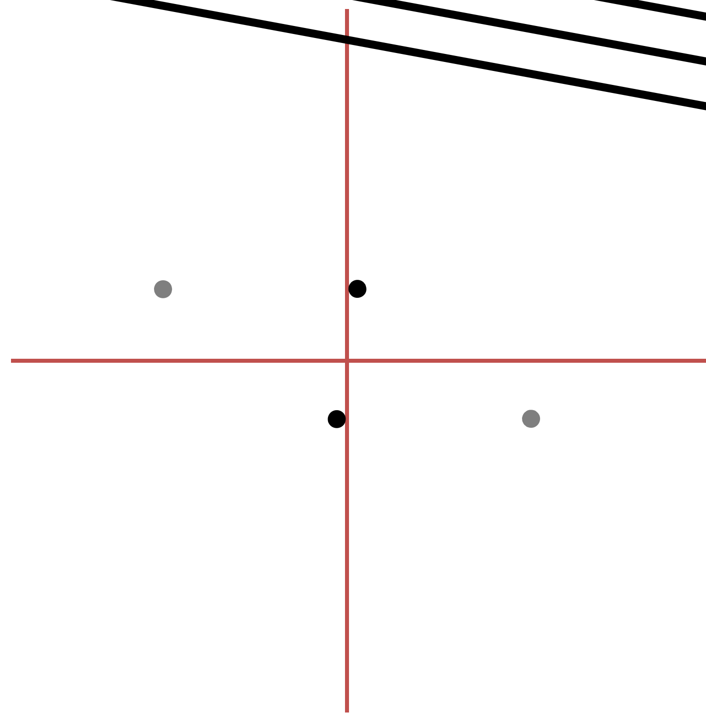
F(A) * F(B)



AB



C

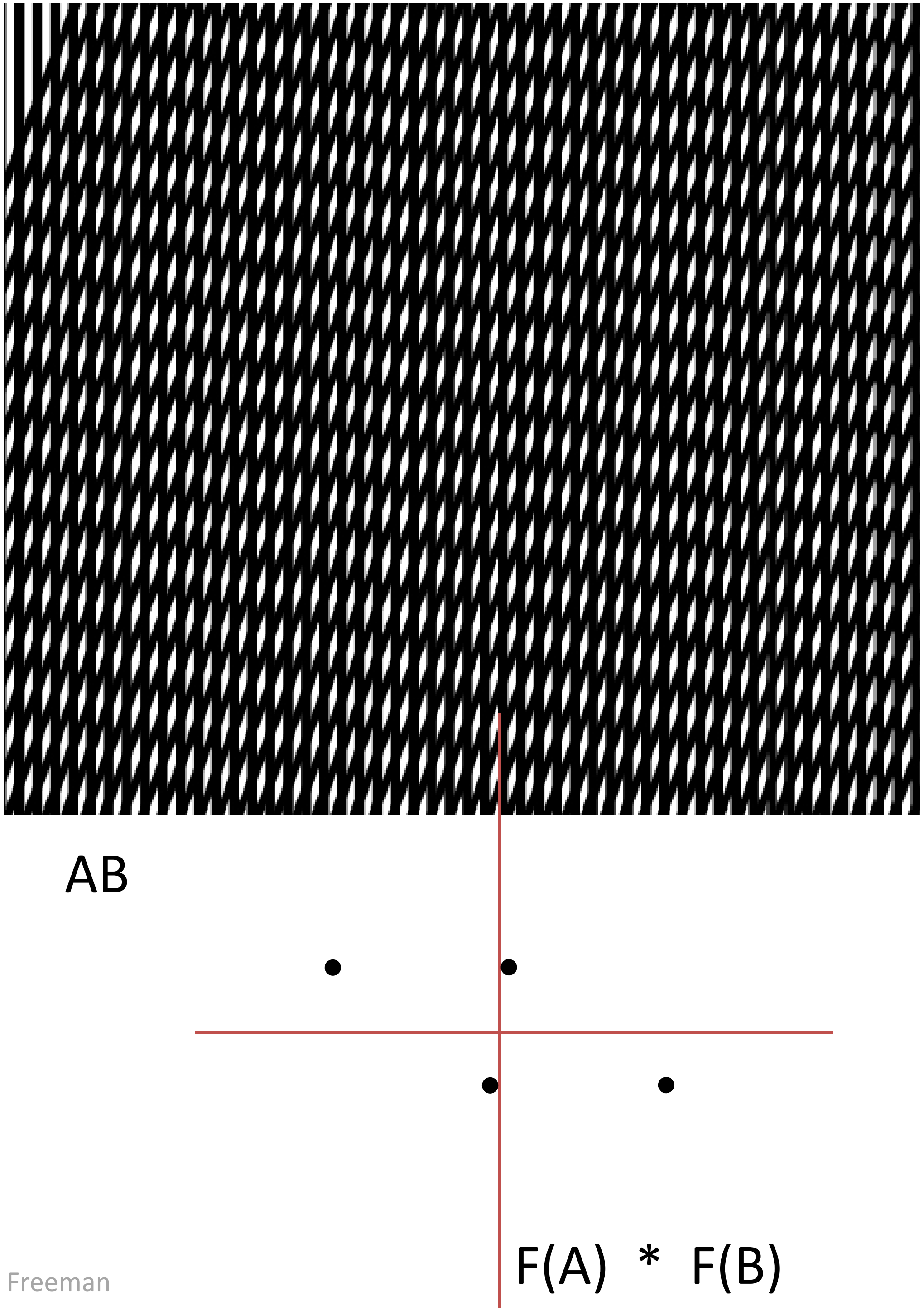


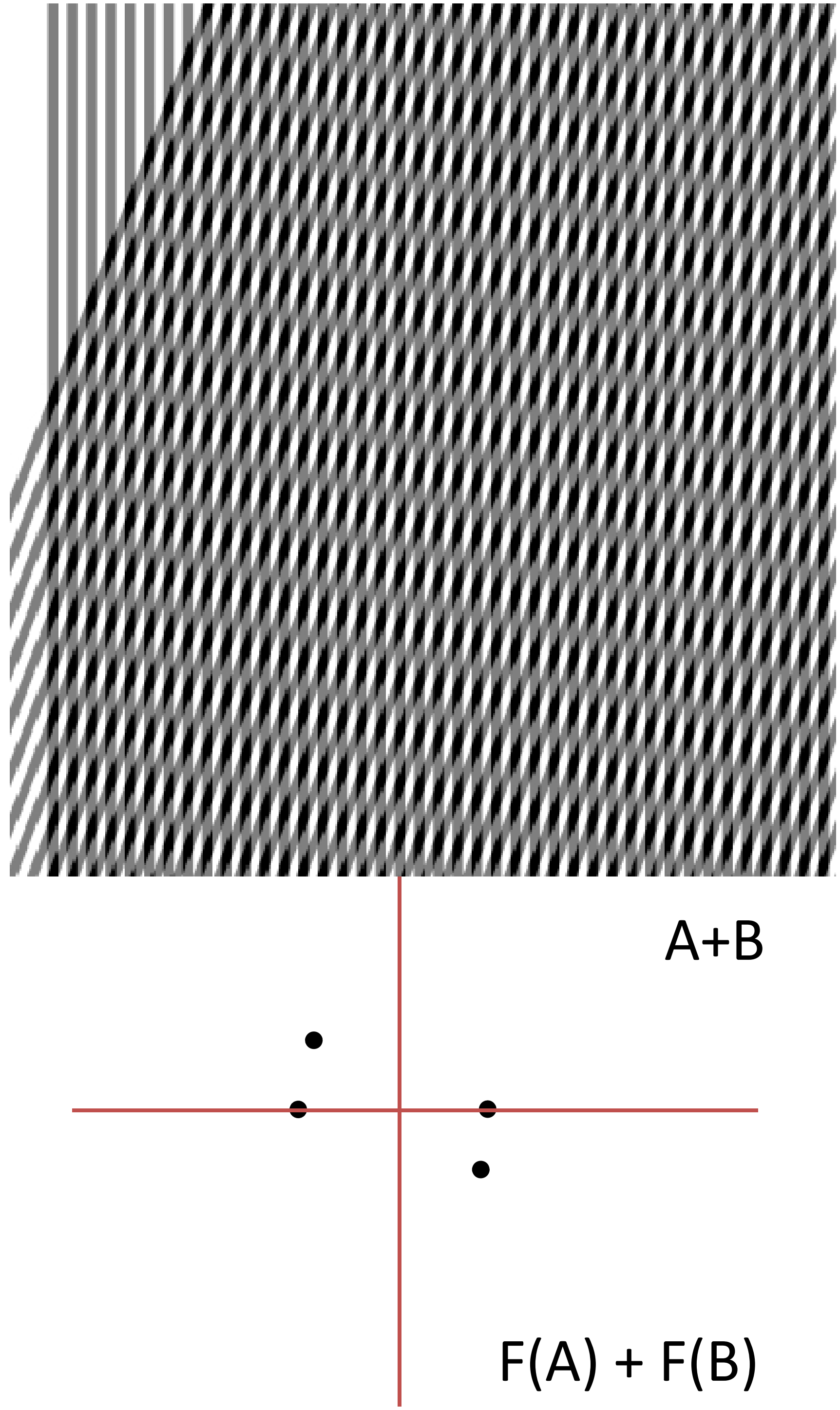
Lowpass(F(A) * F(B))

~=F(C)

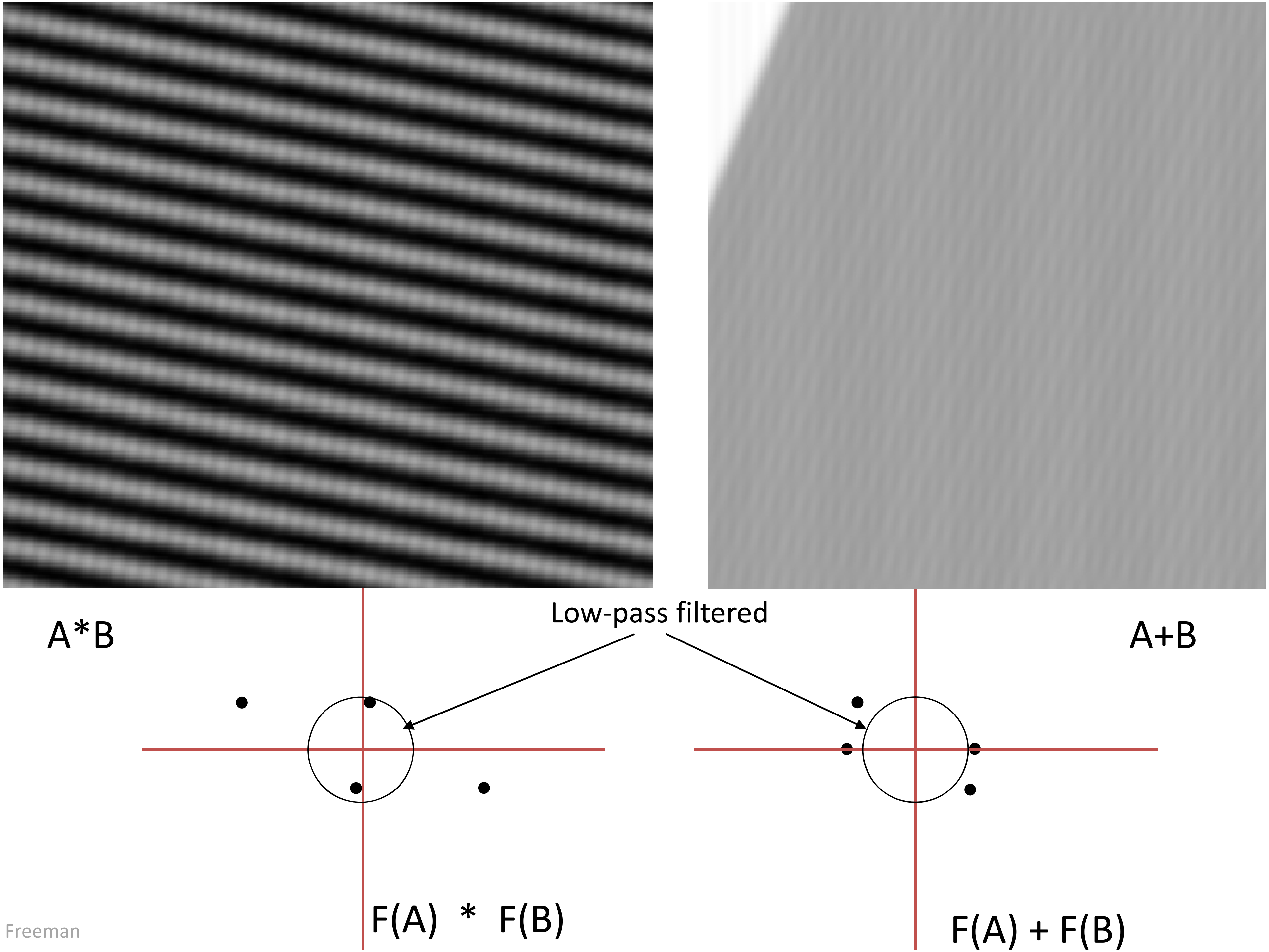
Control test

- If our analysis is correct, if we *add* those two sinusoids (or square waves), and if there is no non-linearity in the display of the sum, then there should only be summing, not convolution, in the frequency domain.



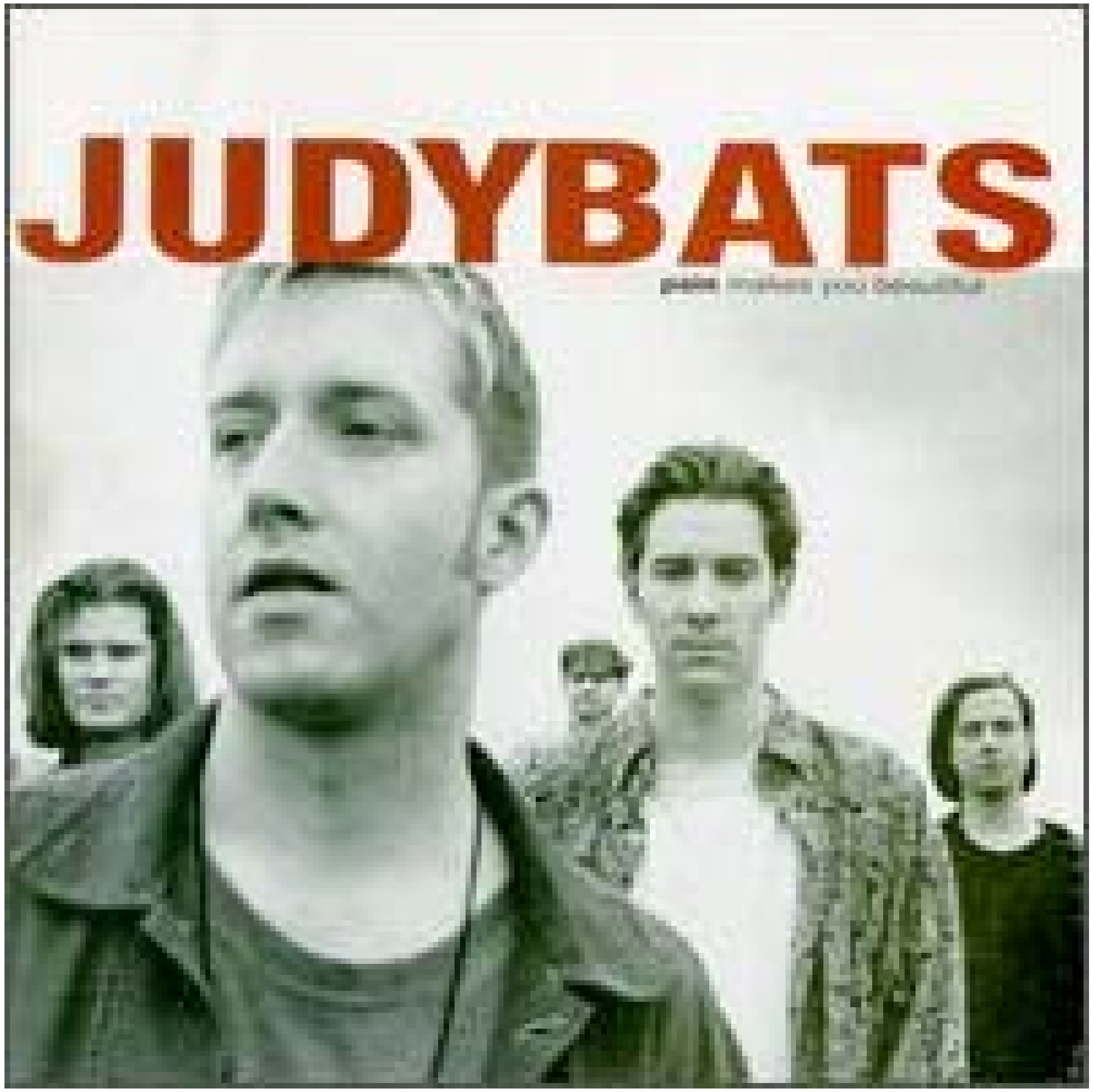


Freeman



Freeman

Image information occurs at all spatial scales



Freeman

Image pyramids

- Gaussian pyramid
- Laplacian pyramid
- Wavelet/QMF pyramid
- Steerable pyramid

Image pyramids

- Gaussian pyramid
- Laplacian pyramid
- Wavelet/QMF pyramid
- Steerable pyramid

Freeman

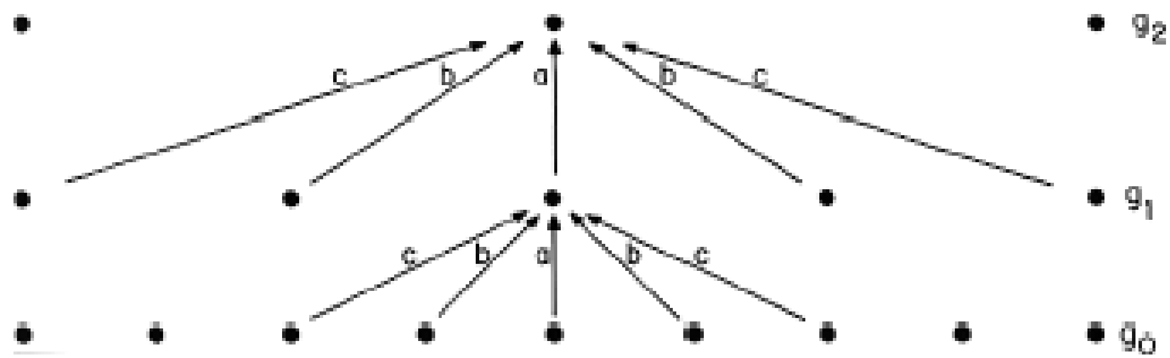
The Gaussian pyramid

- Smooth with gaussians, because
  - a gaussian*gaussian=another gaussian
- Gaussians are low pass filters, so representation is redundant.

Freeman

The computational advantage of pyramids

GAUSSIAN PYRAMID



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.

From: http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

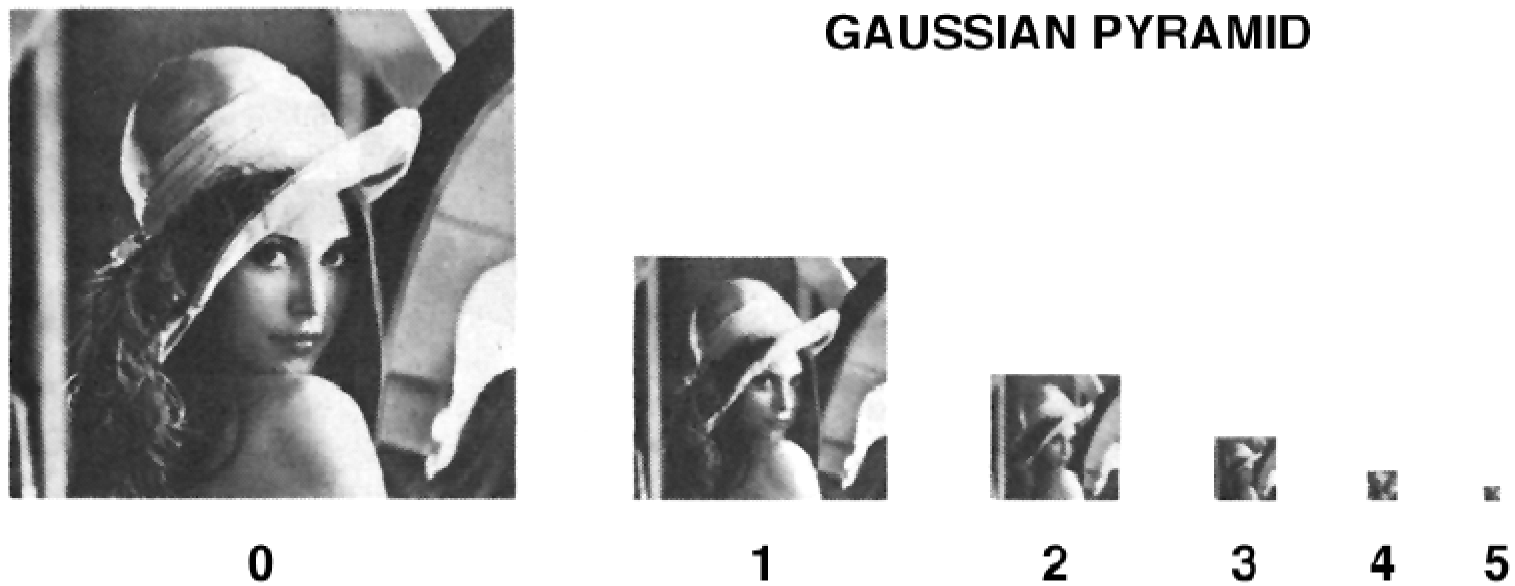


Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image The original image, level 0, meusures 257 by 257 pixels and each higher level array is roughly half the dimensdons of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

512 256 128 64 32 16 8

Freeman

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

$$x_2 = G_1 x_1$$

$$G_1 =$$

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

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

Next pyramid level

$$x_3 = G_2 x_2$$

$$G_2 =$$

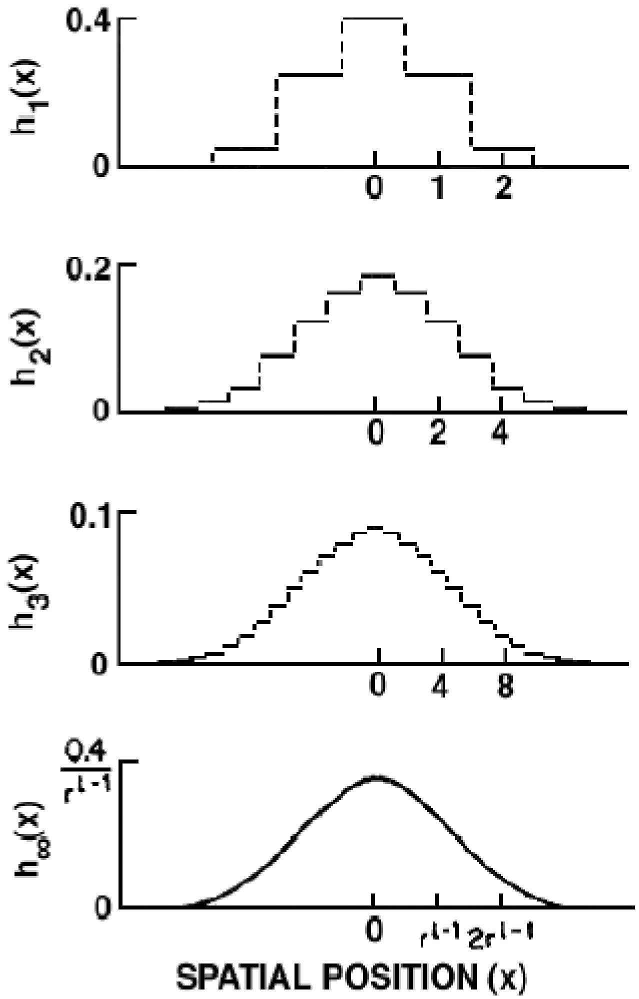
1	4	6	4	1	0	0	0
0	0	1	4	6	4	1	0
0	0	0	0	1	4	6	4
0	0	0	0	0	0	1	4

The combined effect of the two pyramid levels

$$x_3 = G_2 G_1 x_1$$

$$G_2 G_1 =$$

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



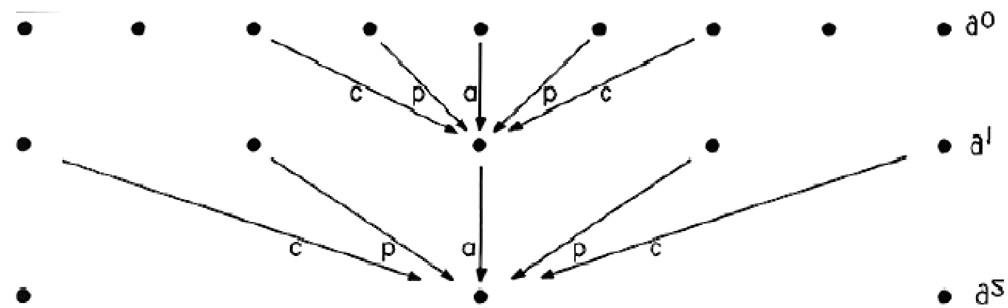


Fig. 2. The equivalent weighting functions $h_l(x)$ for nodes in levels 1, 2, 3, and infinity of the Gaussian pyramid. Note that axis scales have been adjusted by factors of 2 to aid comparison Here the parameter a of the generating kernel is 0.4, and the resulting equivalent weighting functions closely resemble the Gaussian probability density functions.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

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.

Freeman

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Image pyramids

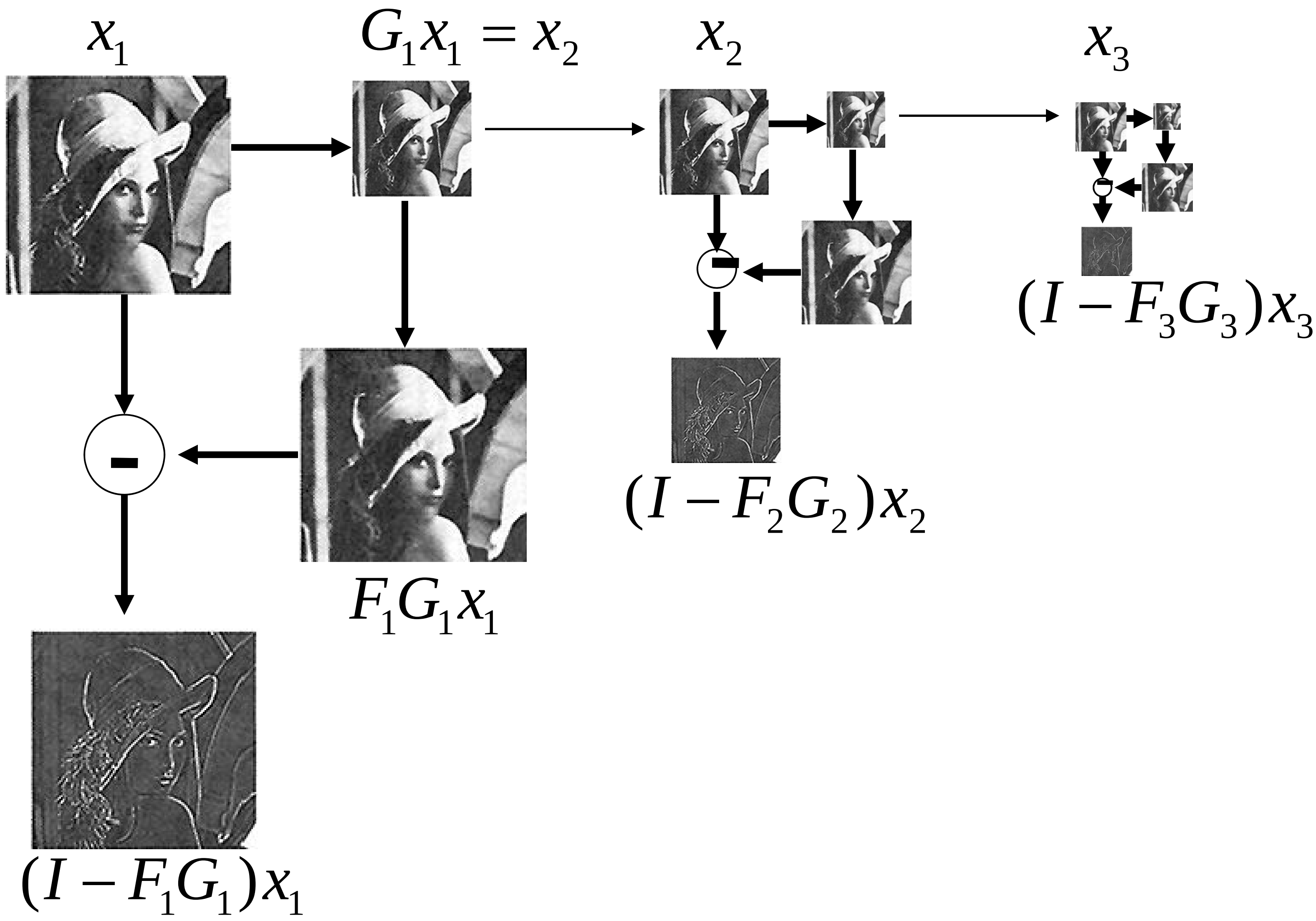
- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Freeman

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



Freeman

Upsampling

$$y_2 = F_3 x_3$$

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

Showing, at full resolution, the information captured at each level of a Gaussian (top) and Laplacian (bottom) pyramid.

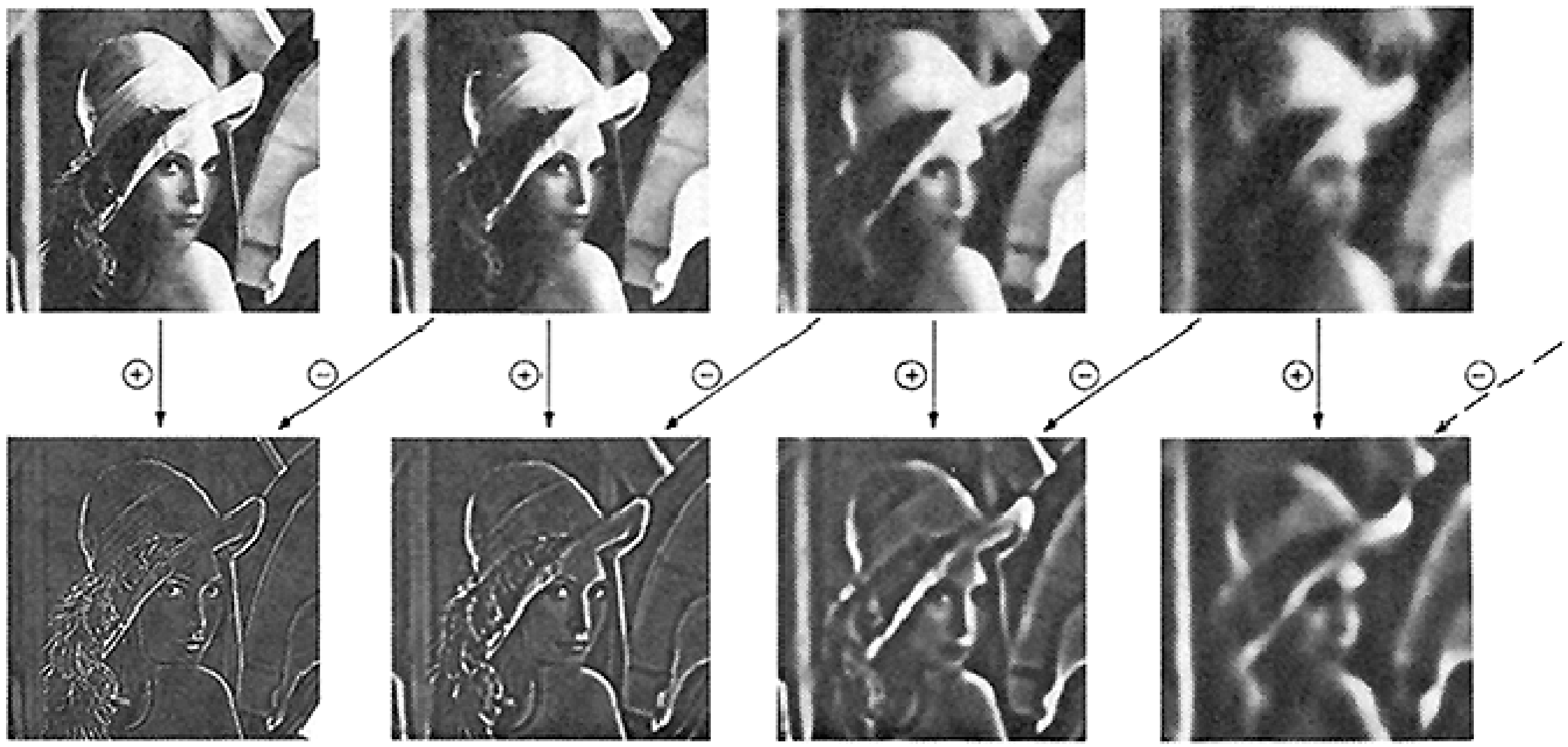


Fig 5. First four levels of the Gaussian and Laplacian pyramid. Gaussian images, upper row, were obtained by expanding pyramid arrays (Fig. 4) through Gaussian interpolation. Each level of the Laplacian pyramid is the difference between the corresponding and next higher levels of the Gaussian pyramid.

From: http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

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

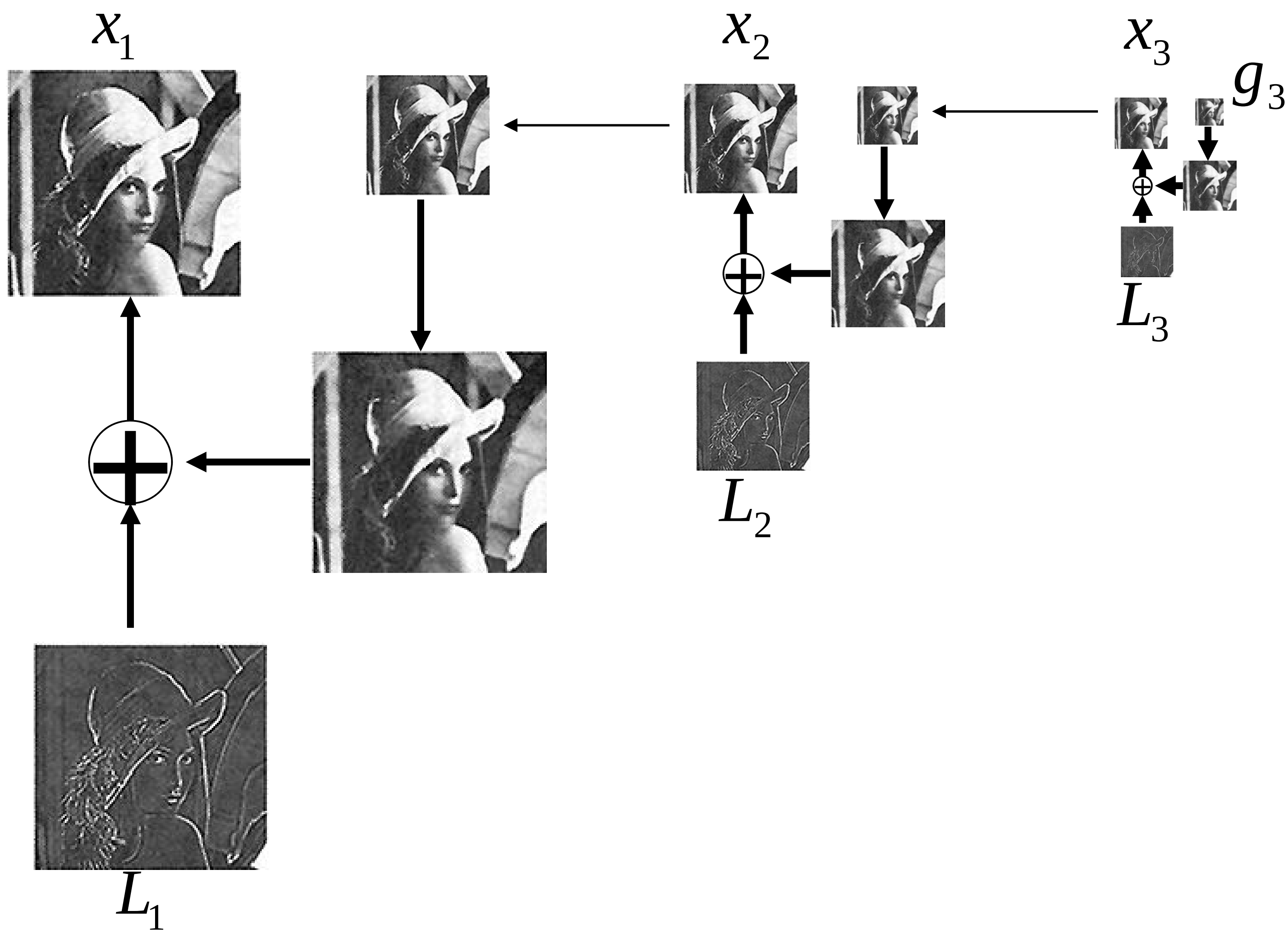
G# is the blur-and-downsample operator at pyramid level #
F# is the blur-and-upsample operator at pyramid level #

Laplacian pyramid elements:
L1 = (I – F1 G1) x1
L2 = (I – F2 G2) x2
L3 = (I – F3 G3) x3
x2 = G1 x1
x3 = G2 x2
x4 = G3 x3

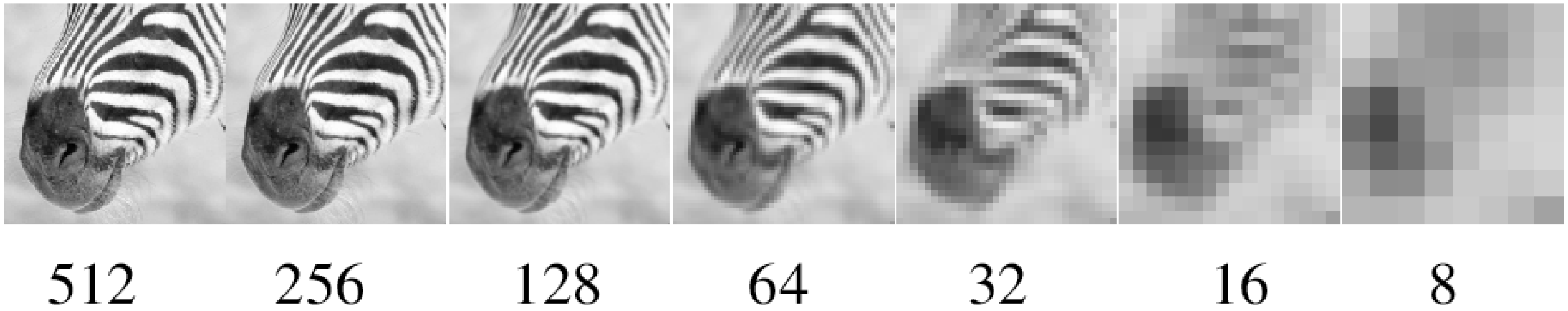
Reconstruction of original image (x1) from Laplacian pyramid elements:
x3 = L3 + F3 x4
x2 = L2 + F2 x3
x1 = L1 + F1 x2

Freeman

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

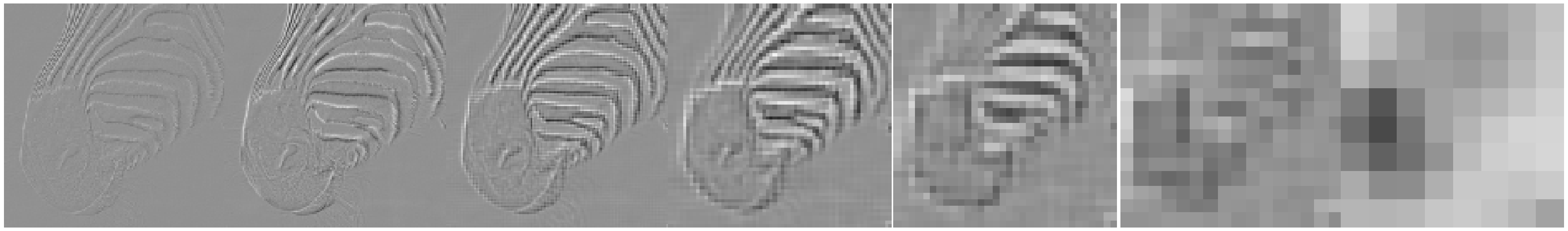


Freeman

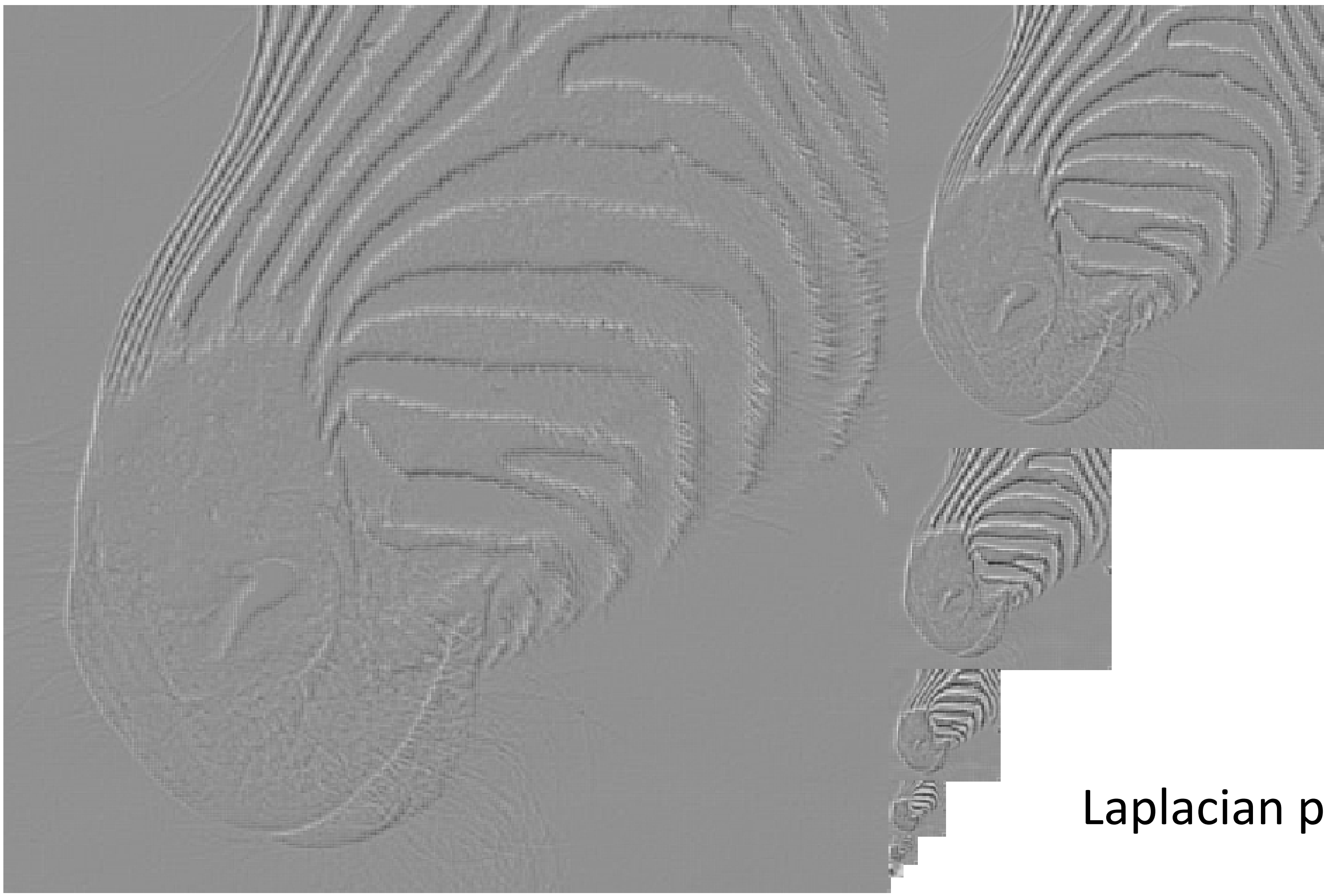


Gaussian pyramid

Freeman



512 256 128 64 32 16 8



Laplacian pyramid

Freeman

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Freeman

Wavelets/QMF's

transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

The simplest wavelet transform: the Haar transform

$$U = \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}$$

The inverse transform for the Haar wavelet

```
>> inv(U)

ans =

    0.5000    0.5000
    0.5000   -0.5000
```

Apply this over multiple spatial positions

U =

1	1	0	0	0	0	0	0
1	-1	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	-1	0	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	-1	0	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	-1

The high frequencies

U =

1	1	0	0	0	0	0	0
1	**-1**	**0**	**0**	**0**	**0**	**0**	**0**
0	0	1	1	0	0	0	0
0	**0**	**1**	**-1**	**0**	**0**	**0**	**0**
0	0	0	0	1	1	0	0
0	**0**	**0**	**0**	**1**	**-1**	**0**	**0**
0	0	0	0	0	0	1	1
0	**0**	**0**	**0**	**0**	**0**	**1**	**-1**

The low frequencies

U =

1	**1**	**0**	**0**	**0**	**0**	**0**	**0**
1	-1	0	0	0	0	0	0
0	**0**	**1**	**1**	**0**	**0**	**0**	**0**
0	0	1	-1	0	0	0	0
0	**0**	**0**	**0**	**1**	**1**	**0**	**0**
0	0	0	0	1	-1	0	0
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**
0	0	0	0	0	0	1	-1

The inverse transform

```
>> inv(U)

ans =
```

L	H	L	H	L	H	L	H
0.5000	0.5000	0	0	0	0	0	0
0.5000	-0.5000	0	0	0	0	0	0
0	0	0.5000	0.5000	0	0	0	0
0	0	0.5000	-0.5000	0	0	0	0
0	0	0	0	0.5000	0.5000	0	0
0	0	0	0	0.5000	-0.5000	0	0
0	0	0	0	0	0	0.5000	0.5000
0	0	0	0	0	0	0.5000	-0.5000

Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

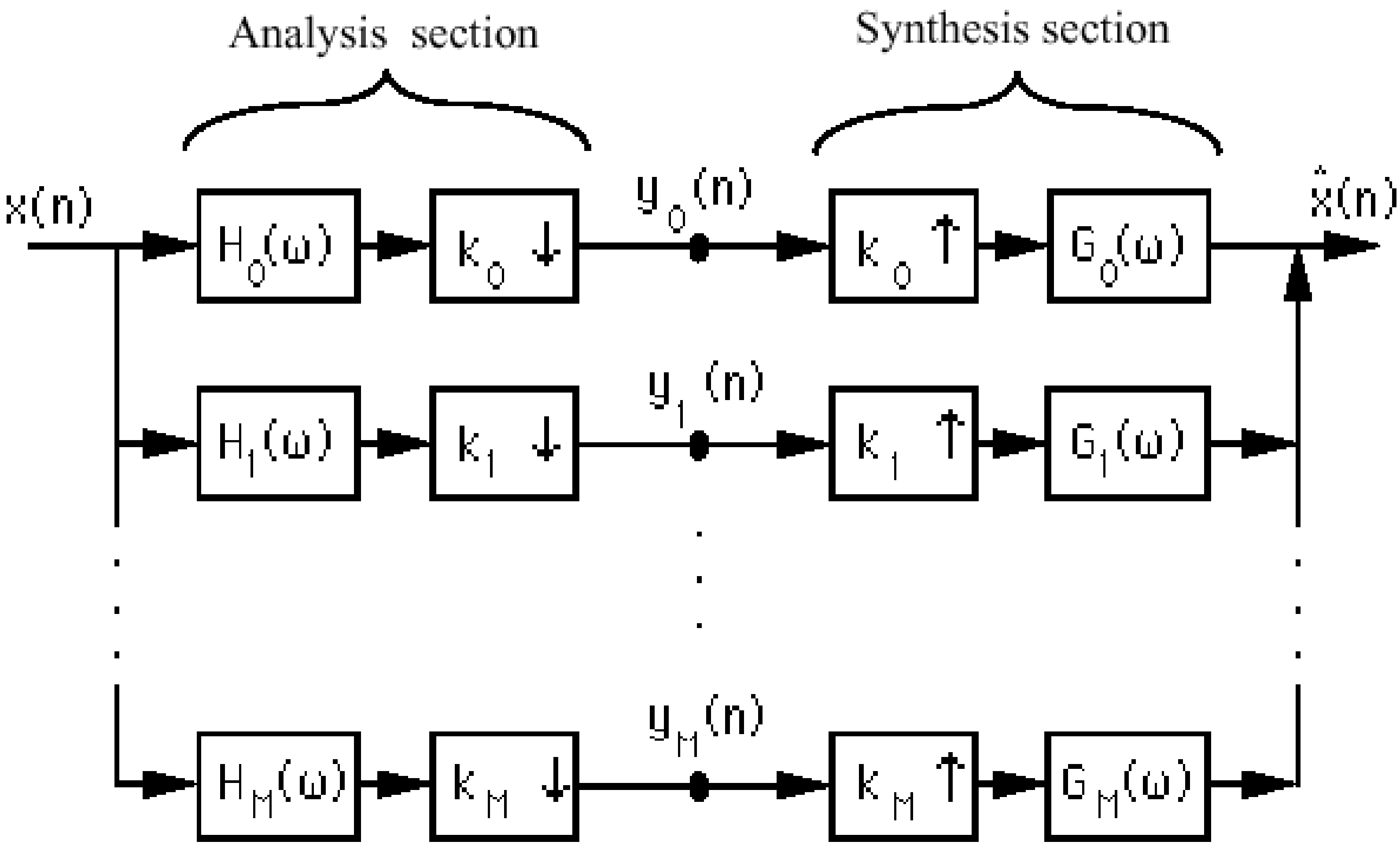


Figure 4.2: An analysis/synthesis filter bank.

Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

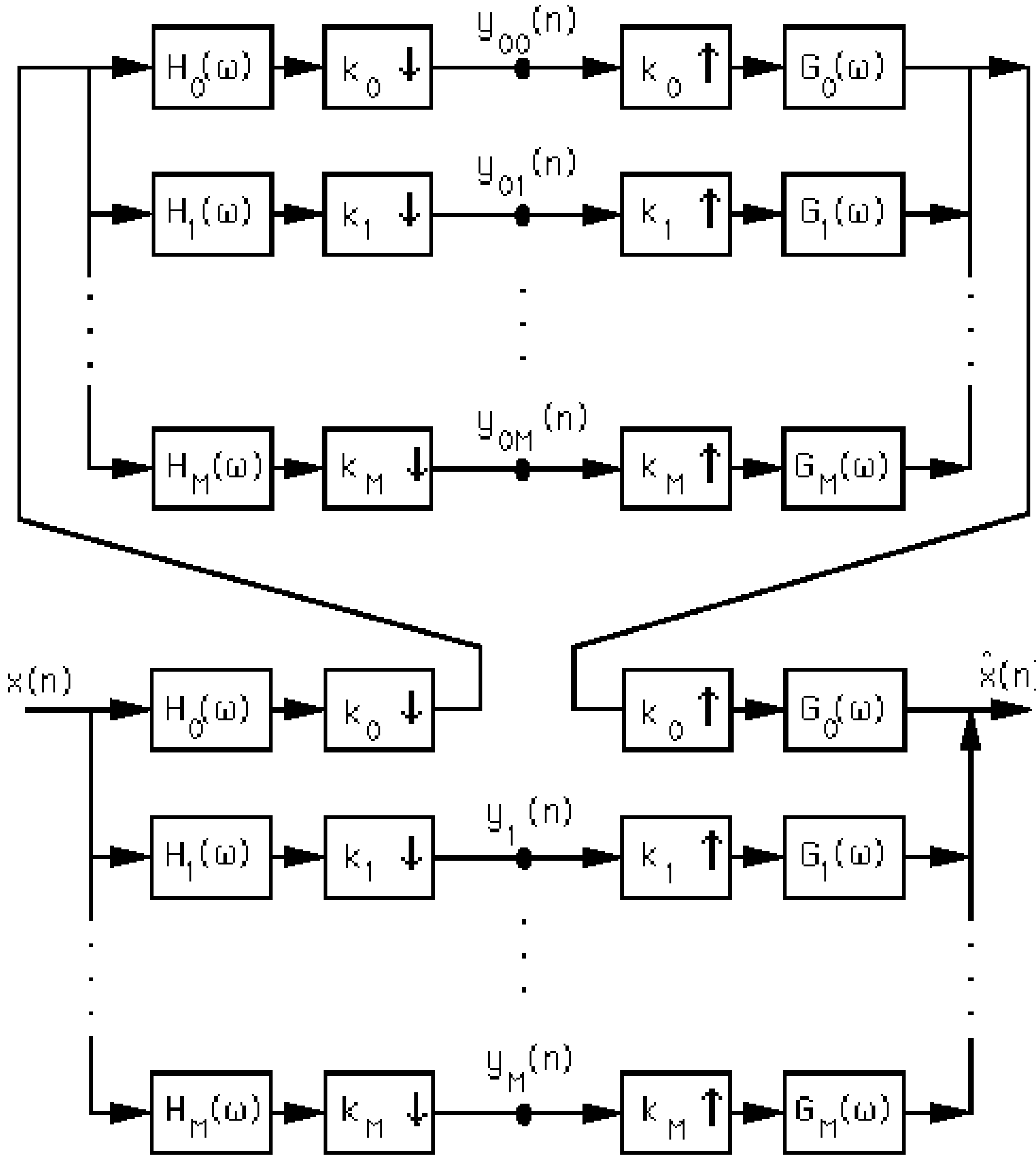


Figure 4.3: A non-uniformly cascaded analysis/synthesis filter bank.

Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

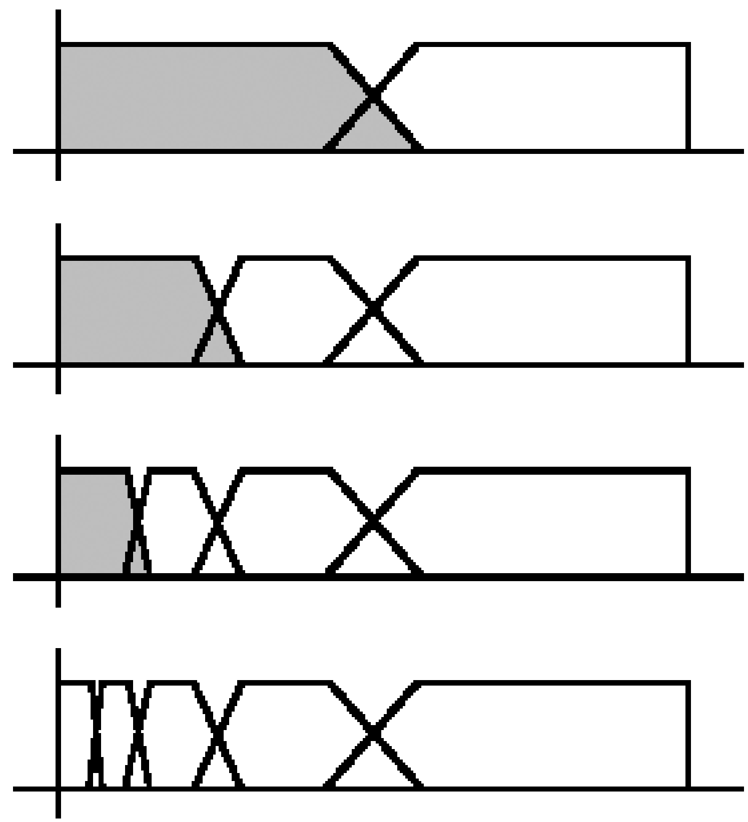


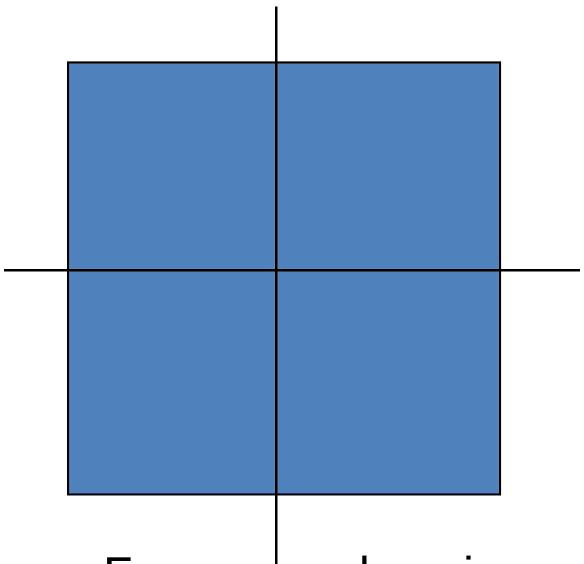
Figure 4.4: Octave band splitting produced by a four-level pyramid cascade of a two-band A/S system. The top picture represents the splitting of the two-band A/S system. Each successive picture shows the effect of re-applying the system to the lowpass subband (indicated in grey) of the previous picture. The bottom picture gives the final four-level partition of the frequency domain. All frequency axes cover the range from 0 to π.

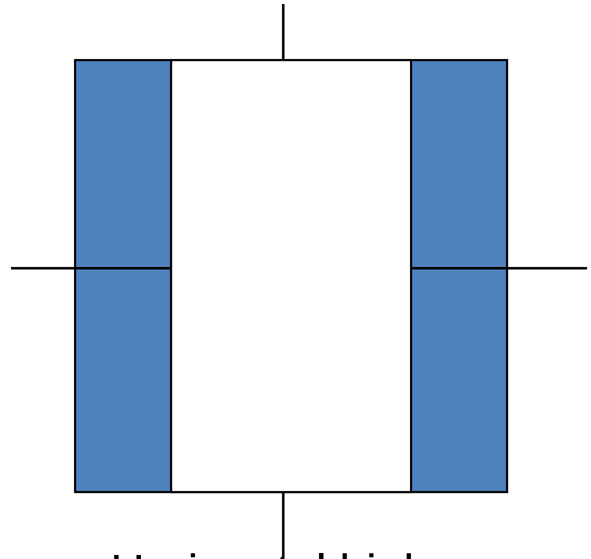
Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

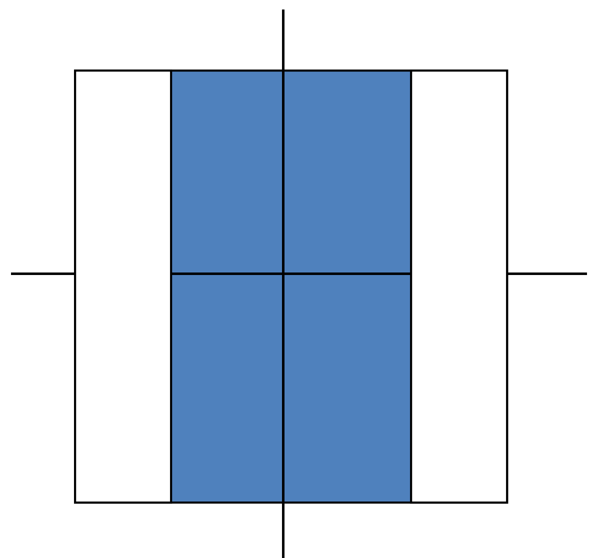
n	QMF-5	QMF-9	QMF-13
0	0.8593118	0.7973934	0.7737113
1	0.3535534	0.41472545	0.42995453
2	-0.0761025	-0.073386624	-0.057827797
3		-0.060944743	-0.09800052
4		0.02807382	0.039045125
5			0.021651438
6			-0.014556438

Table 4.1: Odd-length QMF kernels. Half of the impulse response sample values are shown for each of the normalized lowpass QMF filters (All filters are symmetric about $n = 0$). The appropriate highpass filters are obtained by delaying by one sample and multiplying with the sequence $(-1)^n$.

Now, in 2 dimensions…

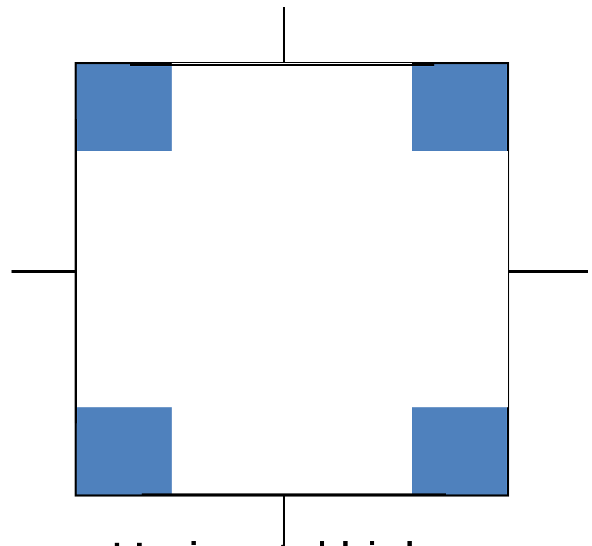




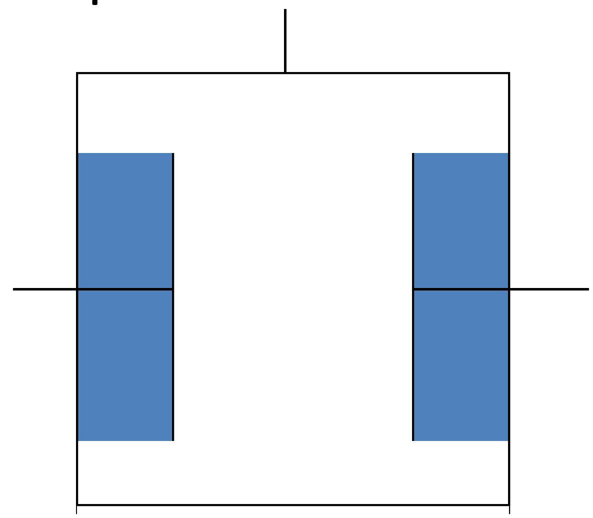


Freeman

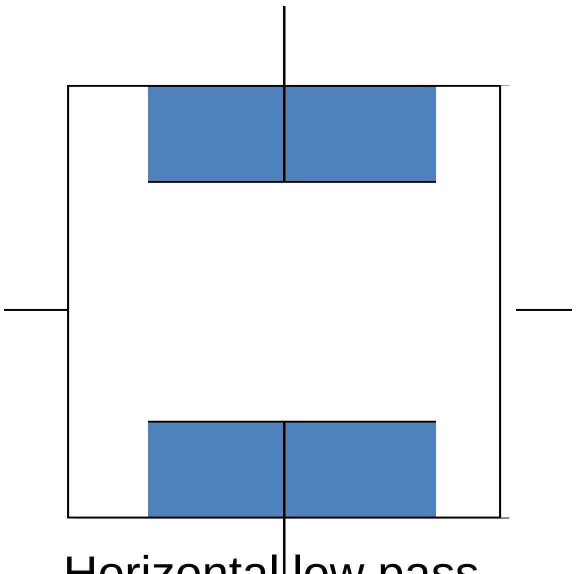
Apply the wavelet transform separable in both dimensions



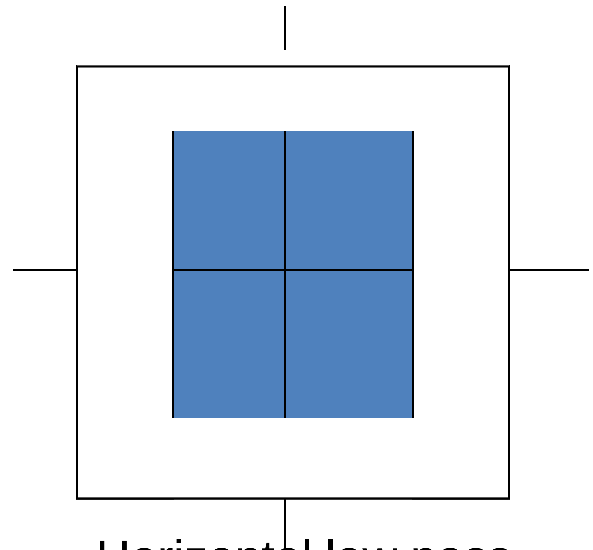
Horizontal high pass,
vertical high pass



Horizontal high pass,
vertical low-pass



Horizontal low pass,
vertical high-pass



Horizontal low pass,
Vertical low-pass

Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

To create 2-d filters, apply the 1-d filters separably in the two spatial dimensions

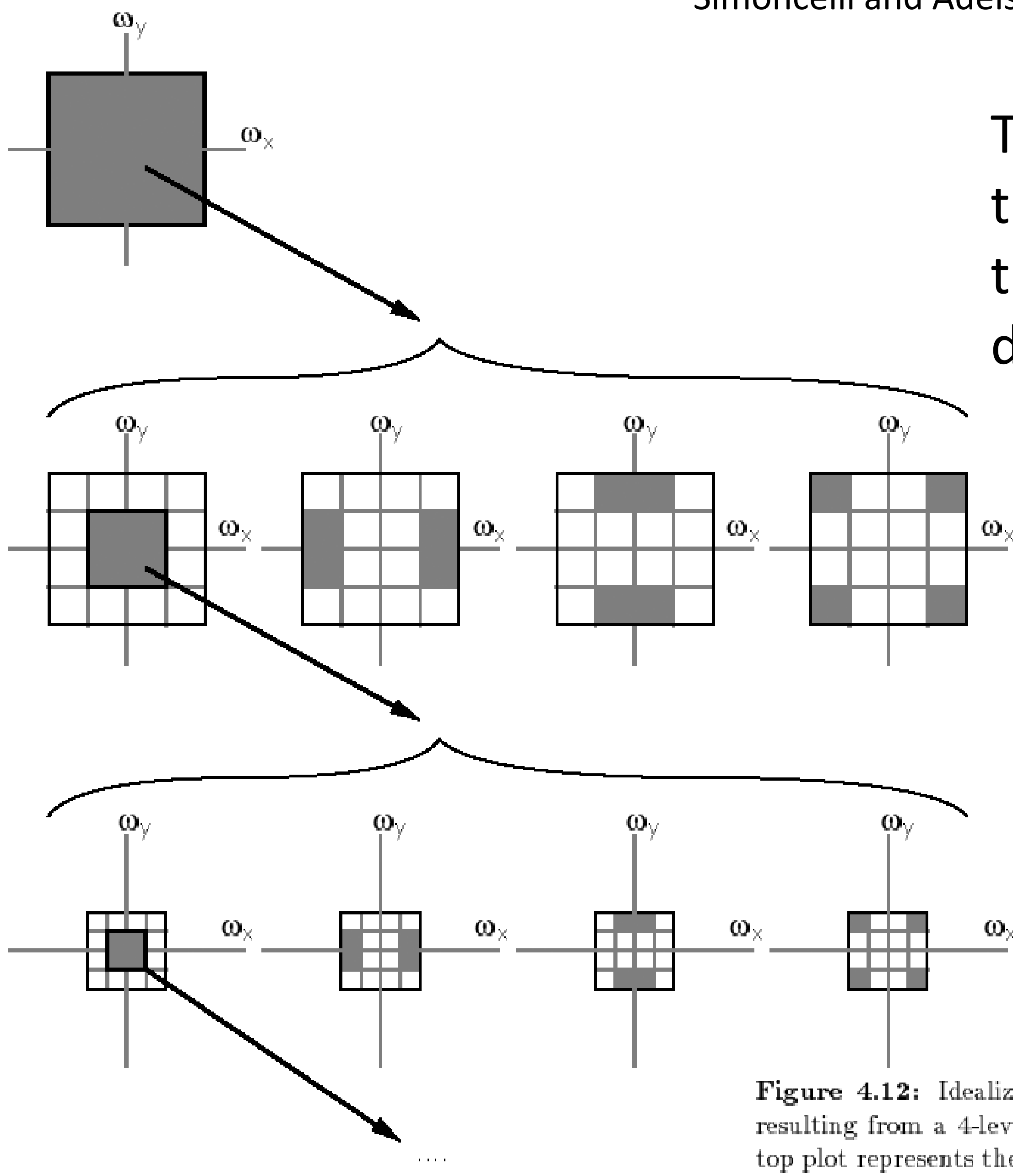
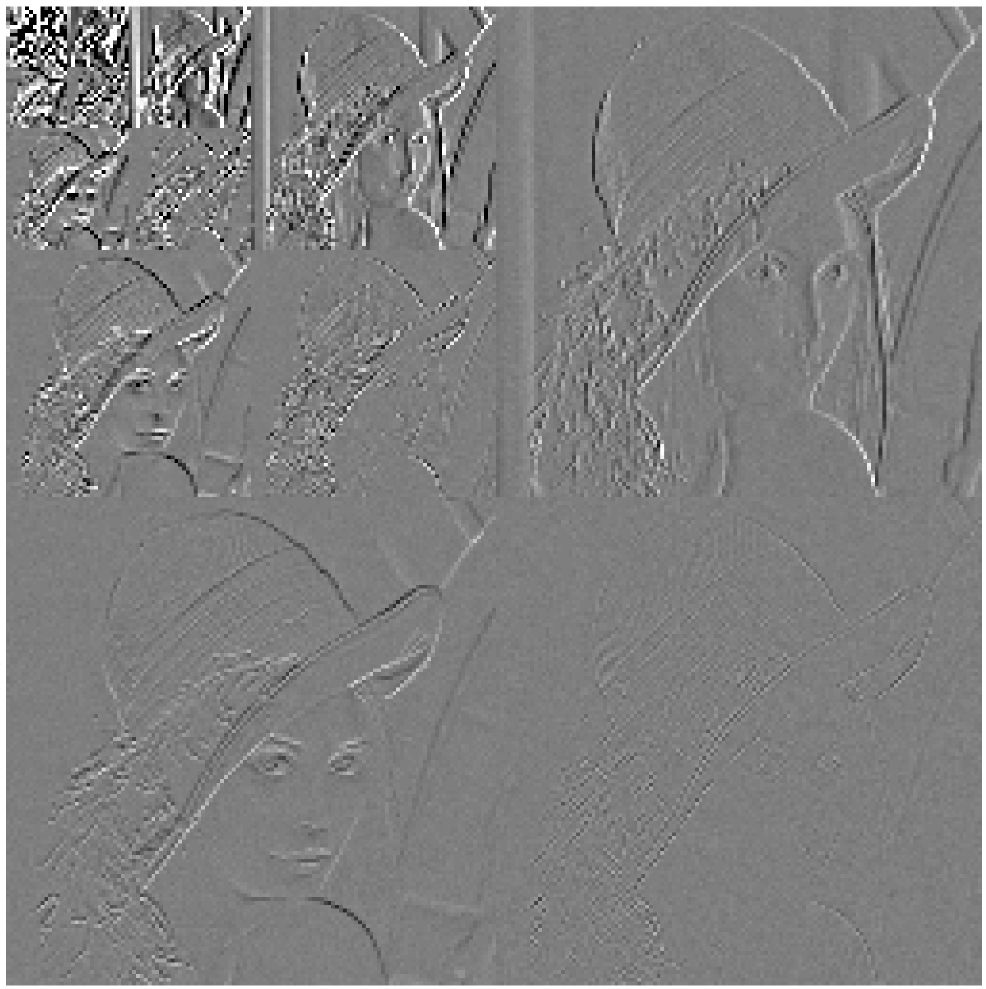


Figure 4.12: Idealized diagram of the partition of the frequency plane resulting from a 4-level pyramid cascade of separable 2-band filters. The top plot represents the frequency spectrum of the original image, with axes ranging from $-\pi$ to π. This is divided into four subbands at the next level. On each subsequent level, the lowpass subband (outlined in bold) is subdivided further.

Freeman

Wavelet/QMF representation



Freeman

What is a good representation for image analysis?

(Goldilocks and the three representations)

- Fourier transform domain tells you “what” (textural properties), but not “where”. In space, this representation is too spread out.
- Pixel domain representation tells you “where” (pixel location), but not “what”. In space, this representation is too localized
- Want an image representation that gives you a local description of image events—what is happening where. That representation might be “just right”.

Good and bad features of wavelet/QMF filters

- Bad:
  - Aliased subbands
  - Non-oriented diagonal subband
- Good:
  - Not overcomplete (so same number of coefficients as image pixels).
  - Good for image compression (JPEG 2000).
  - Separable computation, so it's fast.

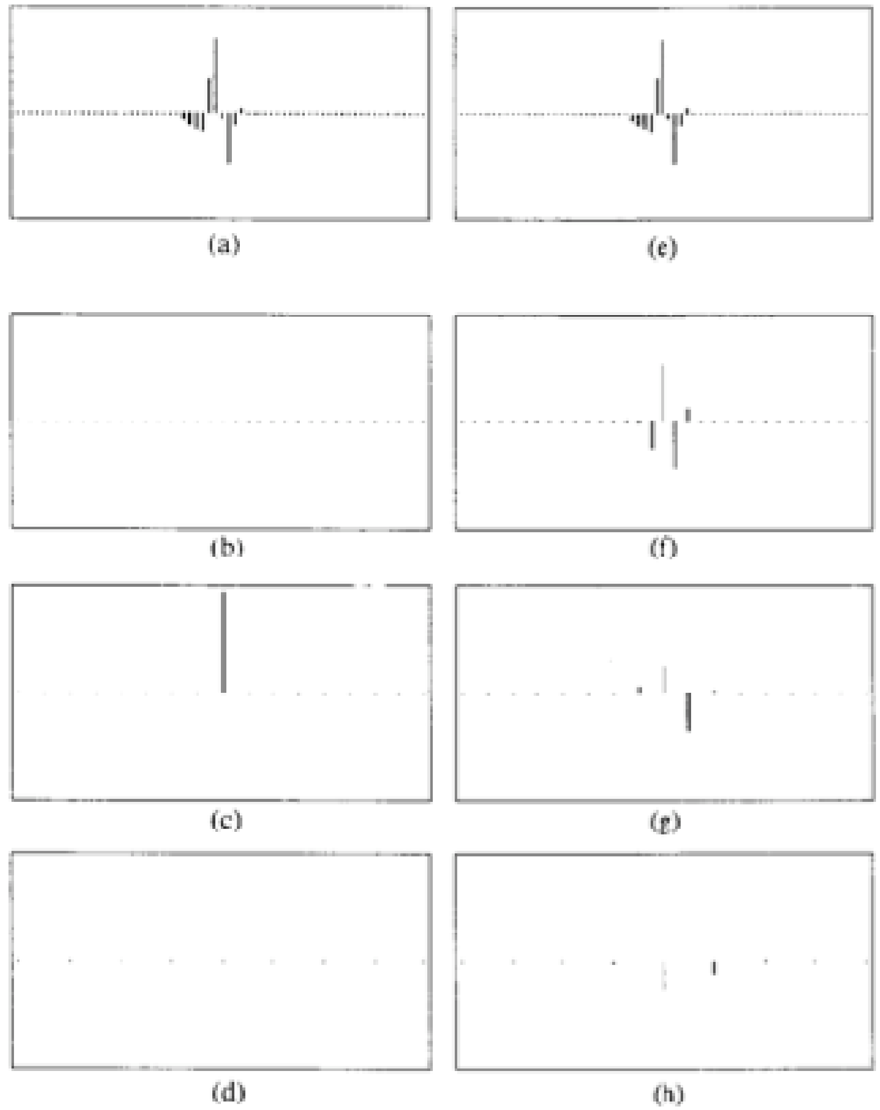


Fig. 1. Effect of translation on the wavelet representation of a signal. (a) Input signal, which is equal to one of the wavelet basis functions. (b)-(d) Decomposition of the signal into three wavelet subbands. Plotted are the coefficients of each subband. Dots correspond to zero-value coefficients. (e) Same input signal, translated one sample to to the right. (f)-(h) Decomposition of the shifted signal into three wavelet subbands. Note the drastic

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Steerable filters

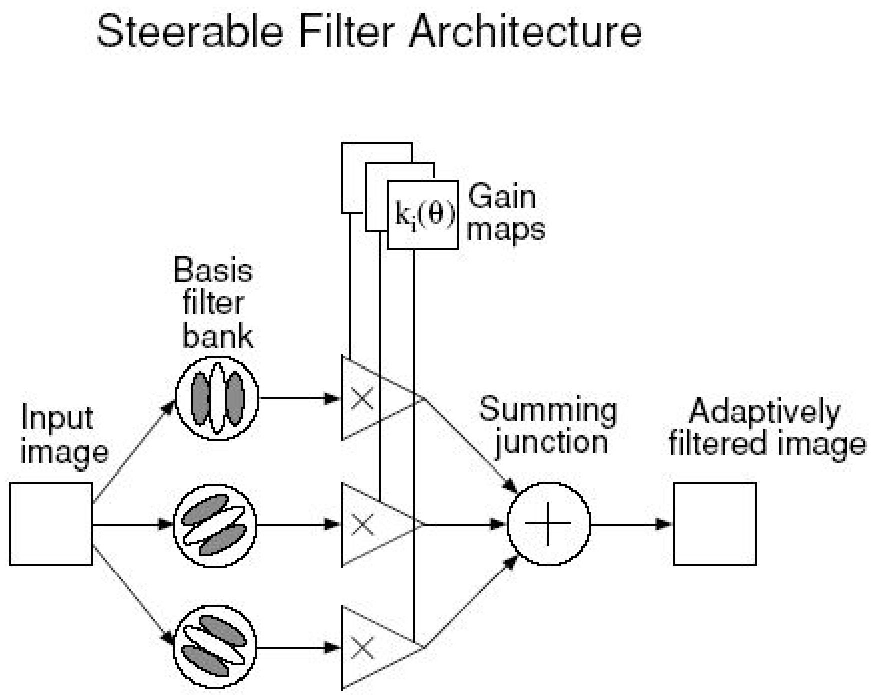


Figure 2-3: Steerable filter system block diagram. A bank of dedicated filters process the image. Their outputs are multiplied by a set of gain maps which adaptively control the orientation of the synthesized filter.

http://people.csail.mit.edu/billf/freemanThesis.pdf

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

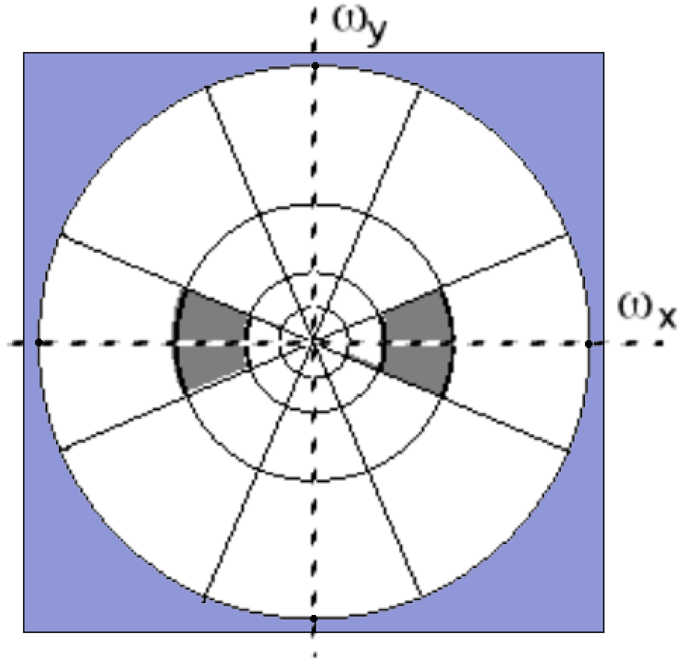


Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For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

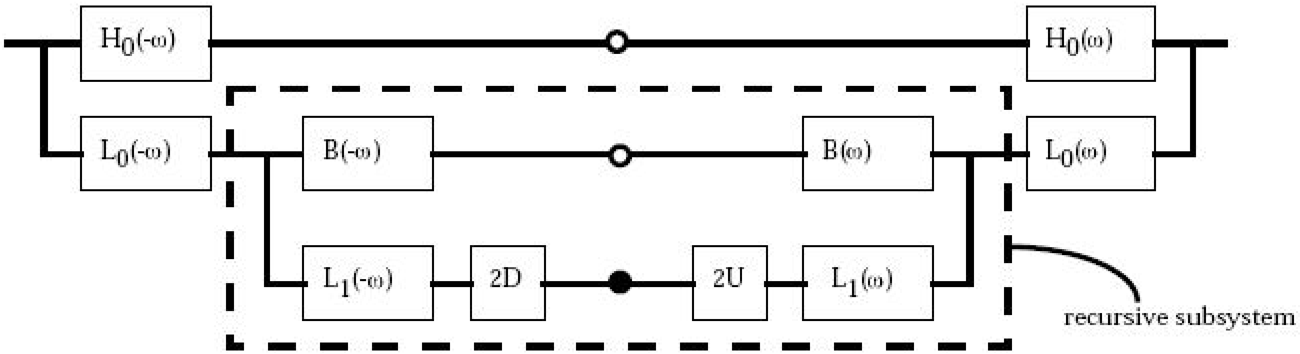


Figure 2: System diagram for the radial portion of the steerable pyramid, illustrating the filtering and sampling operations, and the recursive construction. Boxes containing "2D" and "2U" correspond to downsampling and upsampling by a factor of 2. All other boxes correspond to standard 2D convolution. The circles correspond to the transform coefficients. The recursive construction of a pyramid is achieved by inserting a copy of the diagram contents enclosed by the dashed rectangle at the location of the *solid* circle (i.e., the lowpass branch).

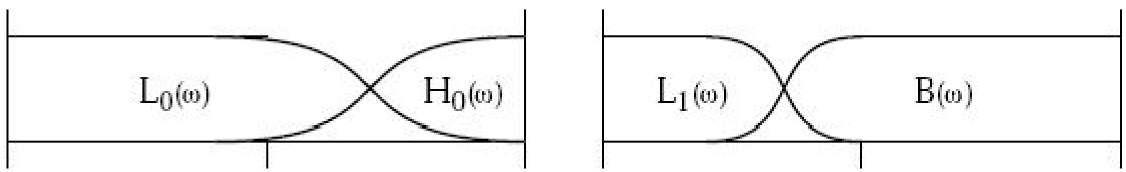
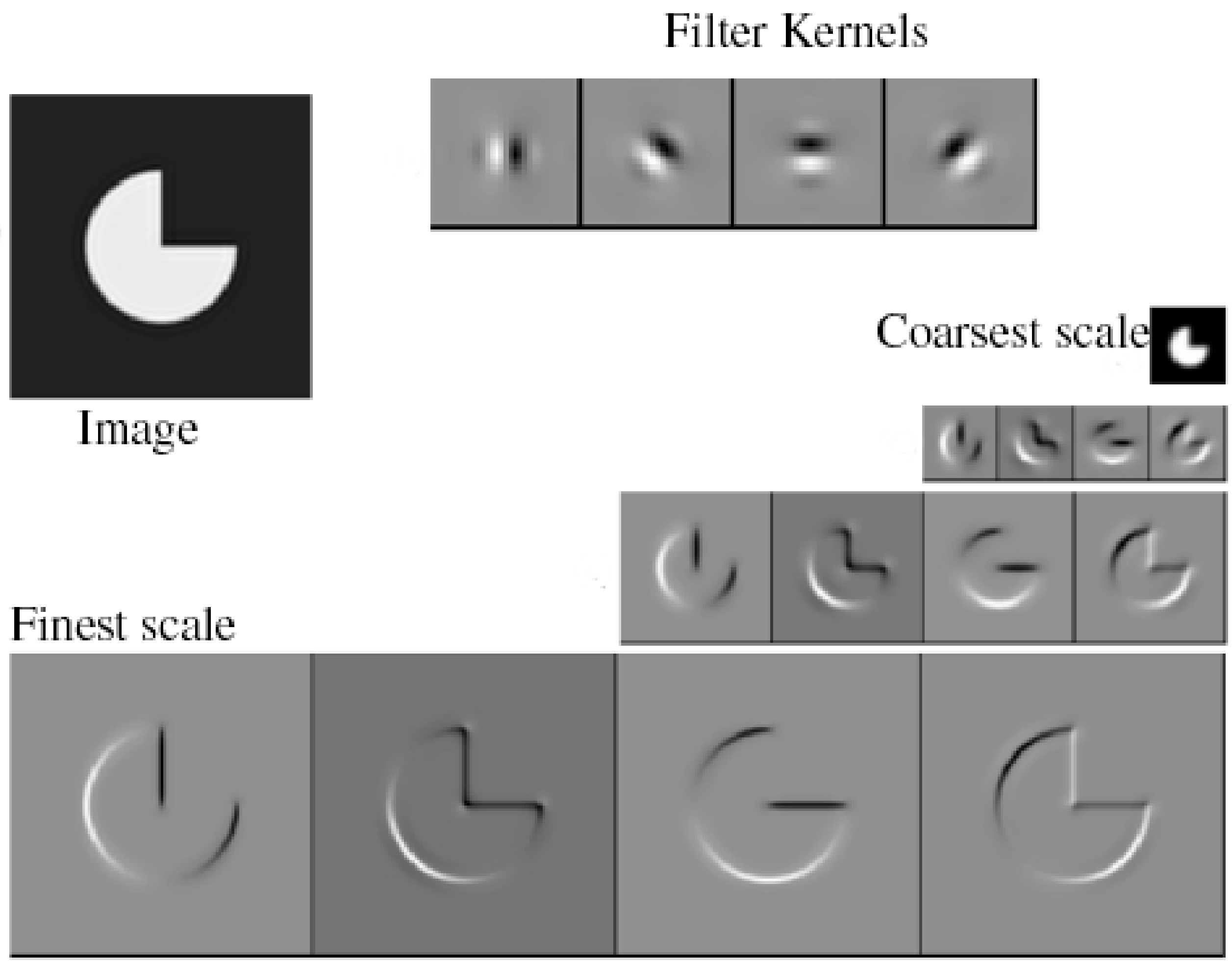


Figure 3: Idealized depiction of filters satisfying the constraints of the block diagram in figure 2. Plots show Fourier spectra over the range $[0, \pi]$.

http://www.merl.com/reports/docs/TR95-15.pdf



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

Non-oriented steerable pyramid

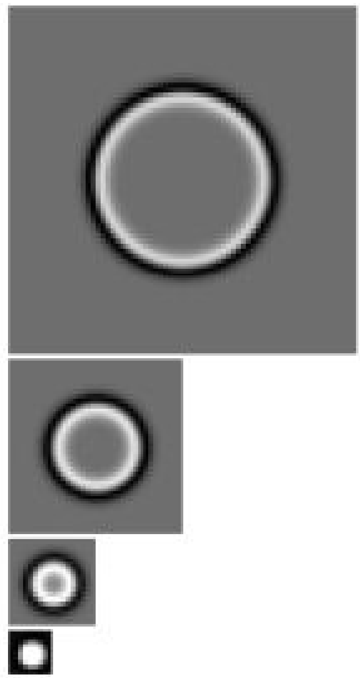


Figure 4: A 3-level $k = 1$ (non-oriented) steerable pyramid. Shown are the bandpass images and the final lowpass image.

http://www.merl.com/reports/docs/TR95-15.pdf

Freeman

3-orientation steerable pyramid

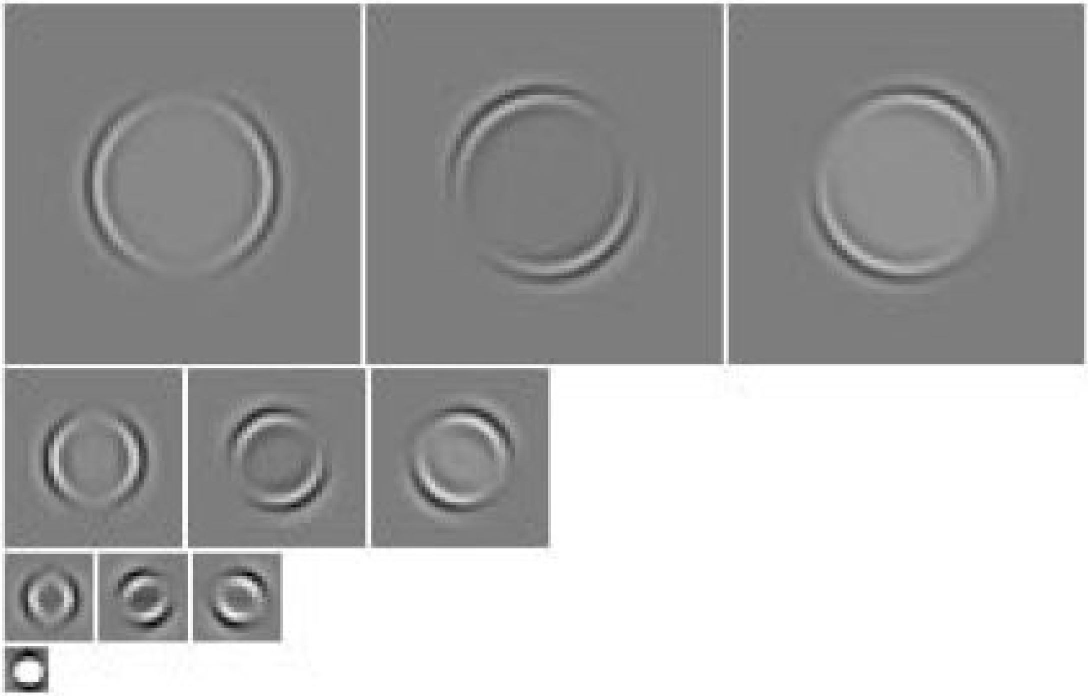


Figure 5: A 3-level $k = 3$ (second derivative) steerable pyramid. Shown are the three band-pass images at each scale and the final lowpass image.

http://www.merl.com/reports/docs/TR95-15.pdf

Freeman

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.

	Laplacian Pyramid	Dyadic QMF/Wavelet	Steerable Pyramid
self-inverting (tight frame)	no	yes	yes
overcompleteness	$4/3$	1	$4k/3$
aliasing in subbands	perhaps	yes	no
rotated orientation bands	no	only on hex lattice [9]	yes

Table 1: Properties of the Steerable Pyramid relative to two other well-known multi-scale representations.

http://www.cns.nyu.edu/ftp/eero/simoncelli95b.pdf Simoncelli and Freeman, ICIP 1995

- Summary of pyramid representations

Image pyramids

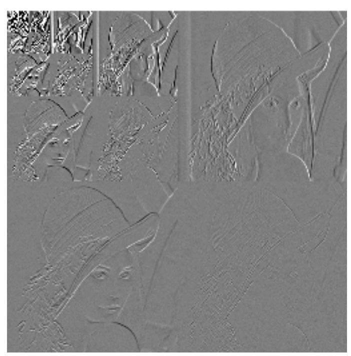
- Gaussian

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

- Laplacian

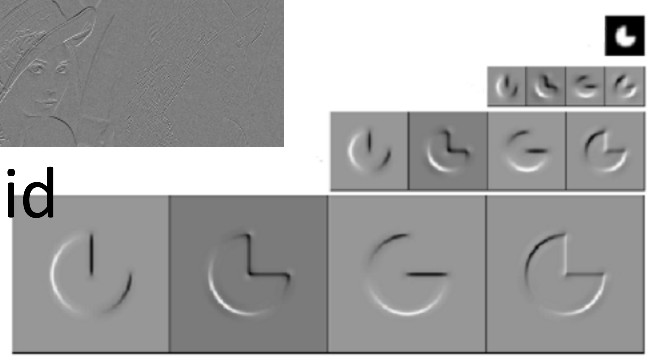
Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- Wavelet/QMF



Bandpassed representation, complete, but with aliasing and some non-oriented subbands.

- 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.

Freeman

Schematic pictures of each matrix transform

Shown for 1-d images

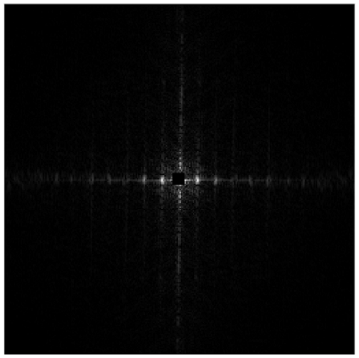
The matrices for 2-d images are the same idea, but more complicated, to account for vertical, as well as horizontal, neighbor relationships.

transformed image

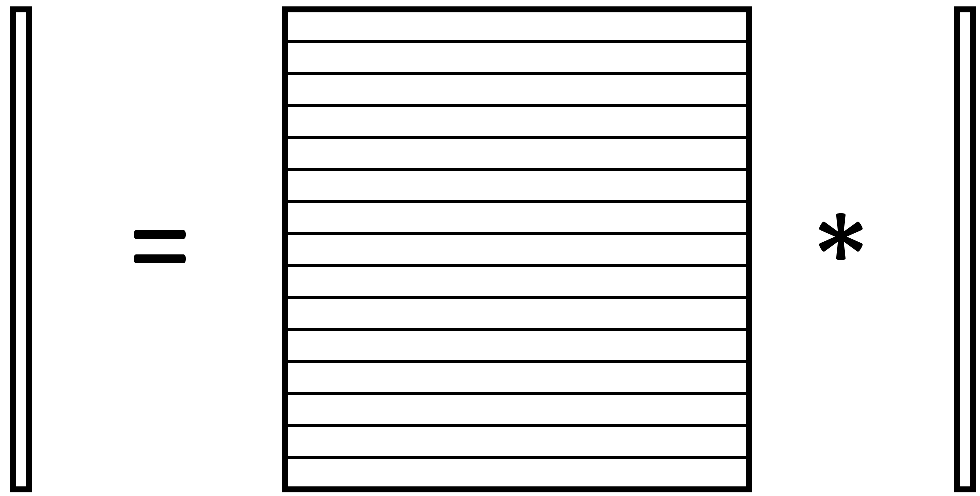
$$\vec{F} = U\vec{f}$$

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



Fourier transform



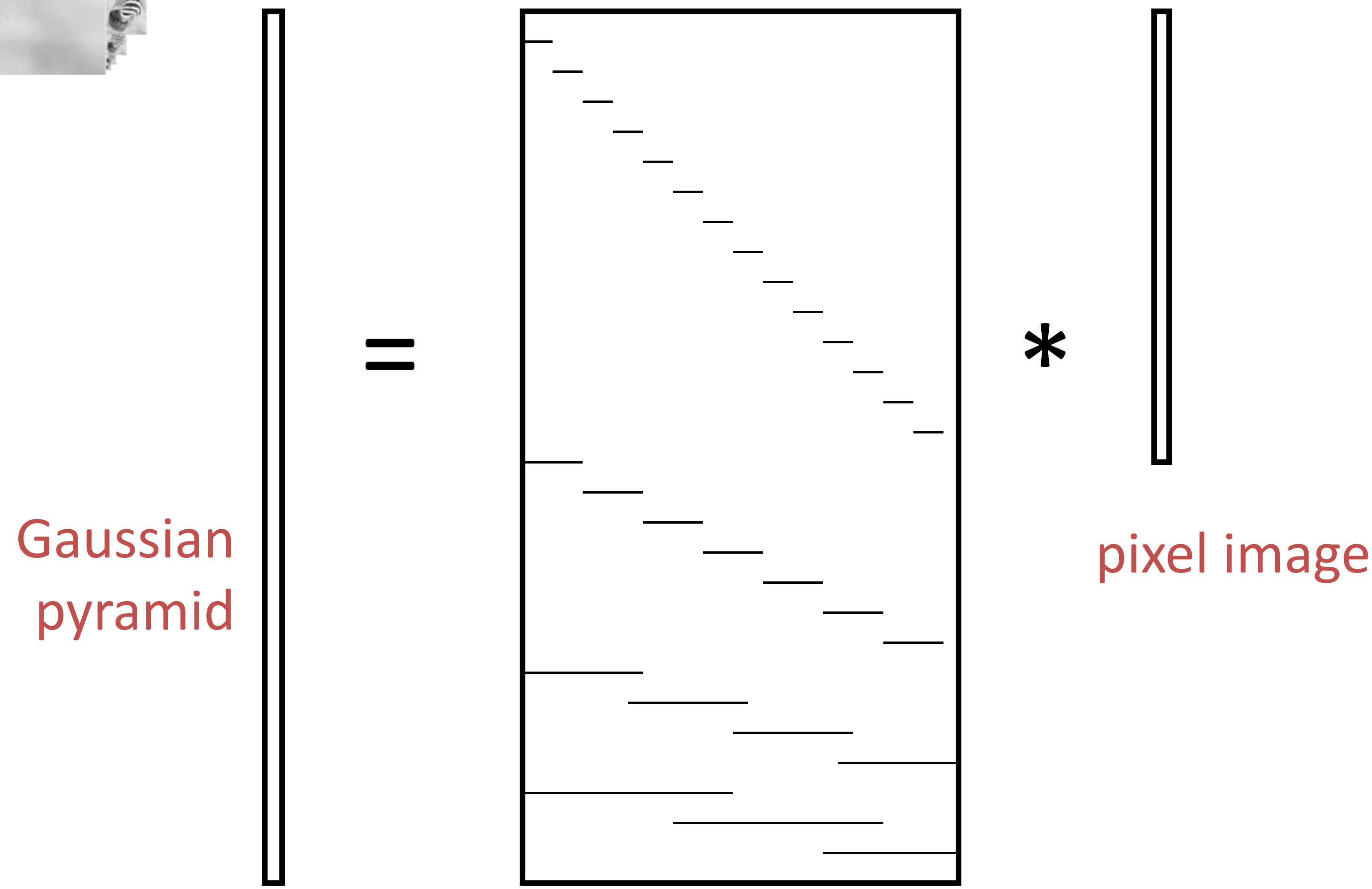
Fourier transform

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

pixel domain image

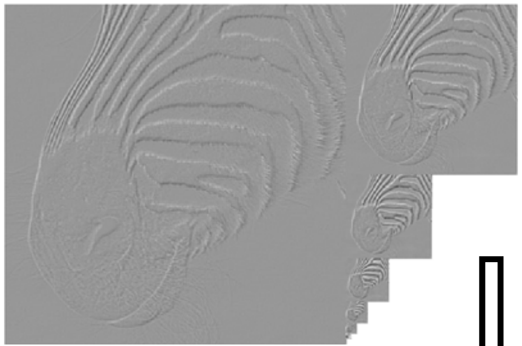
Freeman

Gaussian pyramid



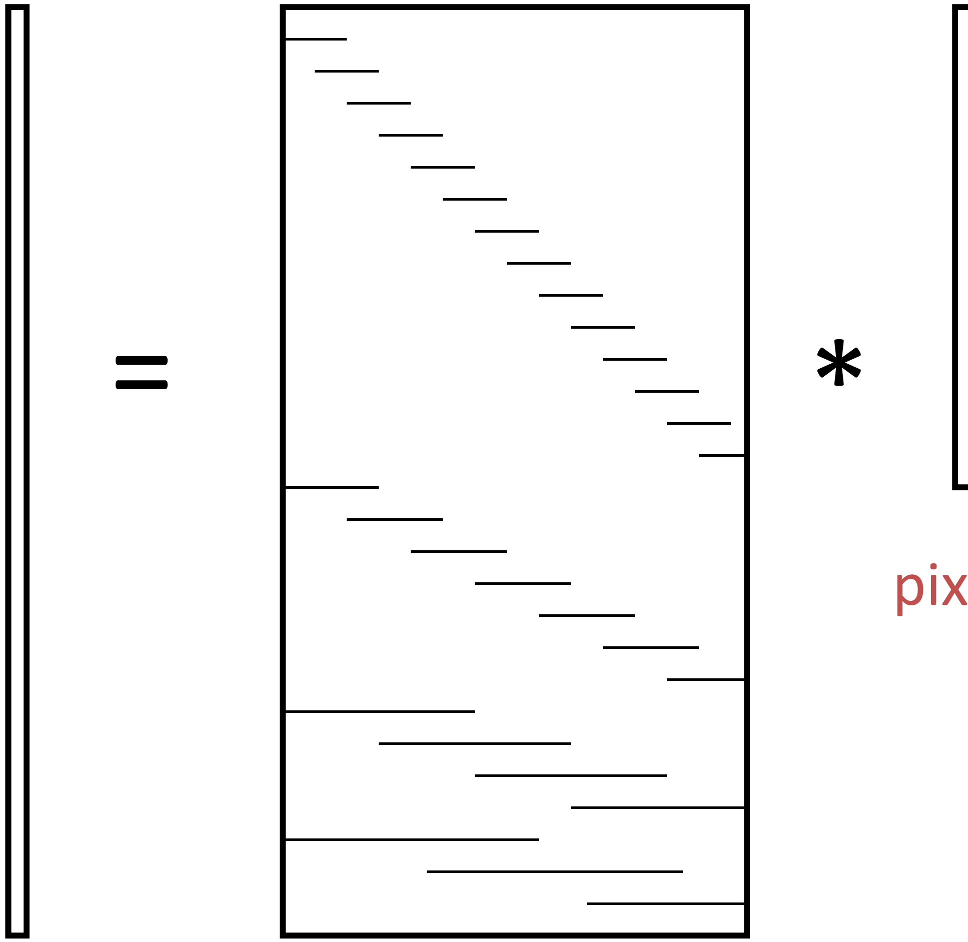
Overcomplete representation.
Low-pass filters, sampled
appropriately for their blur.

Freeman



Laplacian pyramid

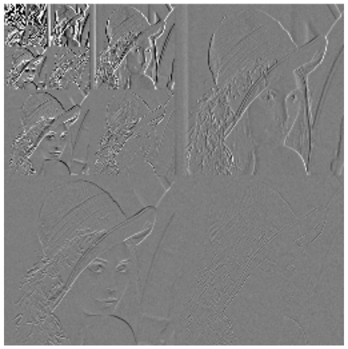
Laplacian pyramid



pixel image

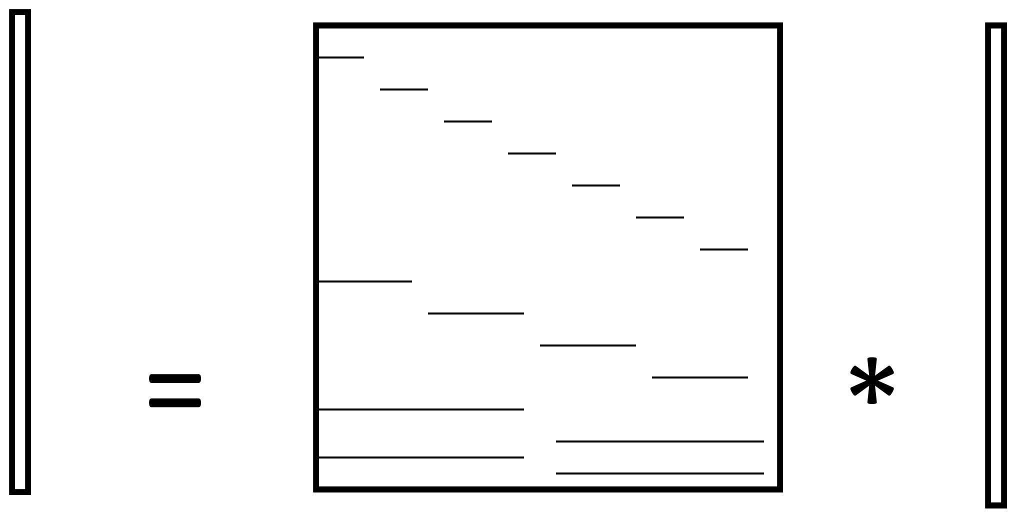
Overcomplete representation.
Transformed pixels represent bandpassed image information.

Freeman



Wavelet (QMF) transform

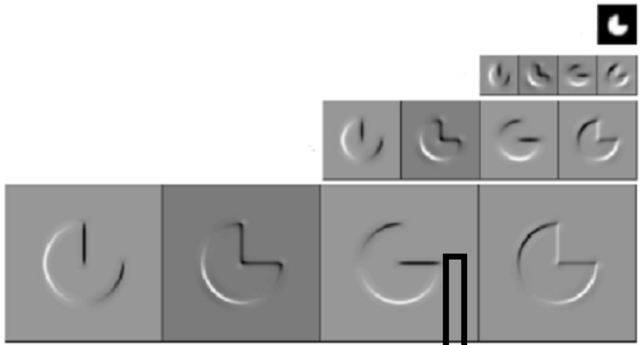
Wavelet pyramid

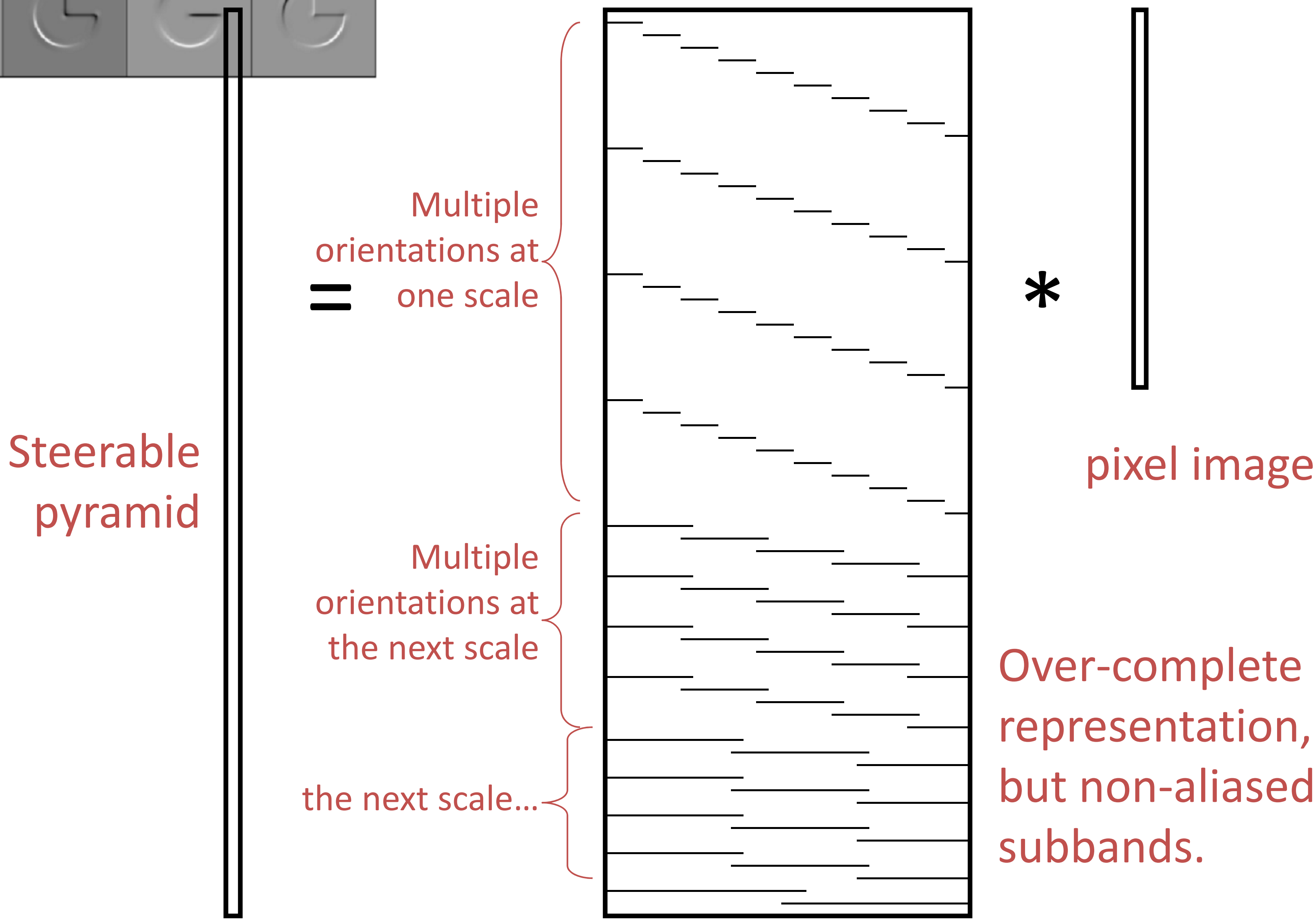


Ortho-normal transform (like Fourier transform), but with localized basis functions.

pixel image

Steerable pyramid





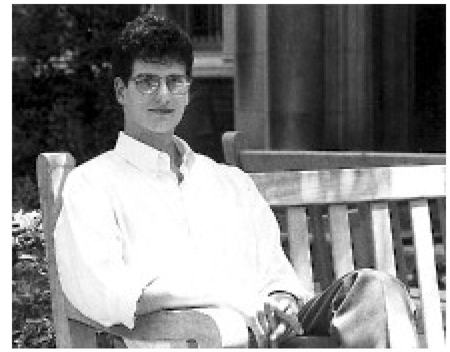
Matlab resources for pyramids (with tutorial)

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

Freeman

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

Pyramid methods in image processing

The image pyramid offers a flexible, convenient multiresolution format that mirrors the multiple scales of processing in the human visual system.

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

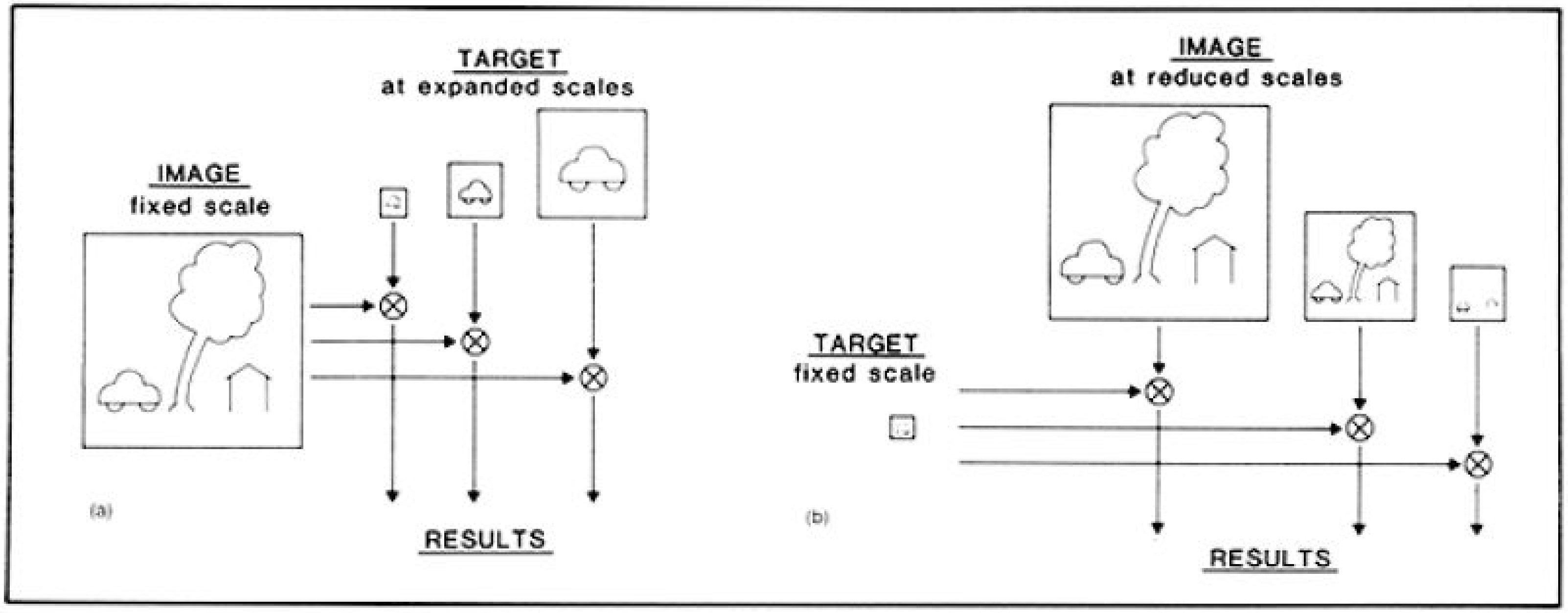


Fig. 1. Two methods of searching for a target pattern over many scales. In the first approach, (a), copies of the target pattern are constructed at several expanded scales, and each is convolved with the original image. In the second approach, (b), a single copy of the target is convolved with copies 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').

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

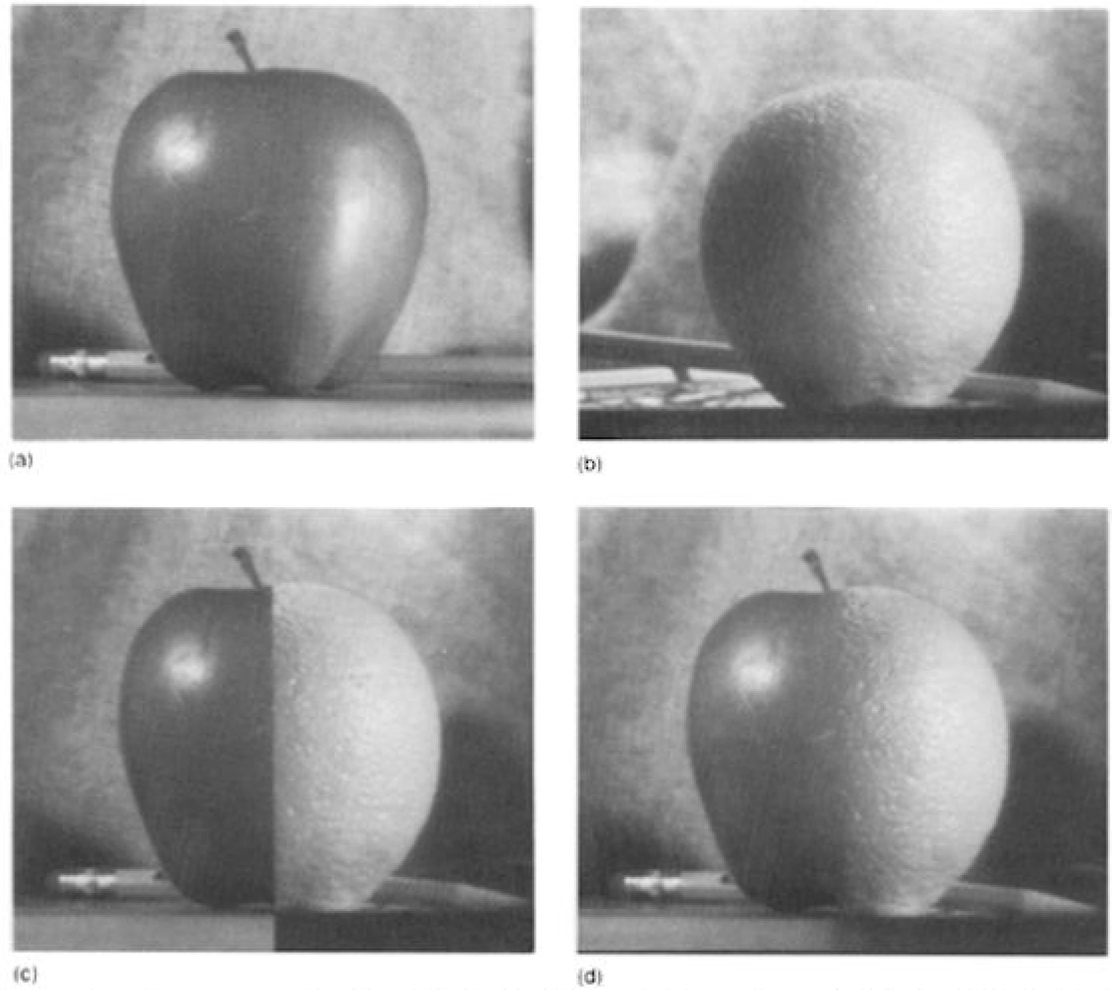


Fig. 10. *Image mosaics. The left half of image (a) is catinated with the right half of image (b) to give the mosaic in (c). Note that the boundary between regions is clearly visible. The mosaic in (d) was obtained by combining images separately in each spatial frequency band of their pyramid representations then expanding and summing these bandpass mosaics.*

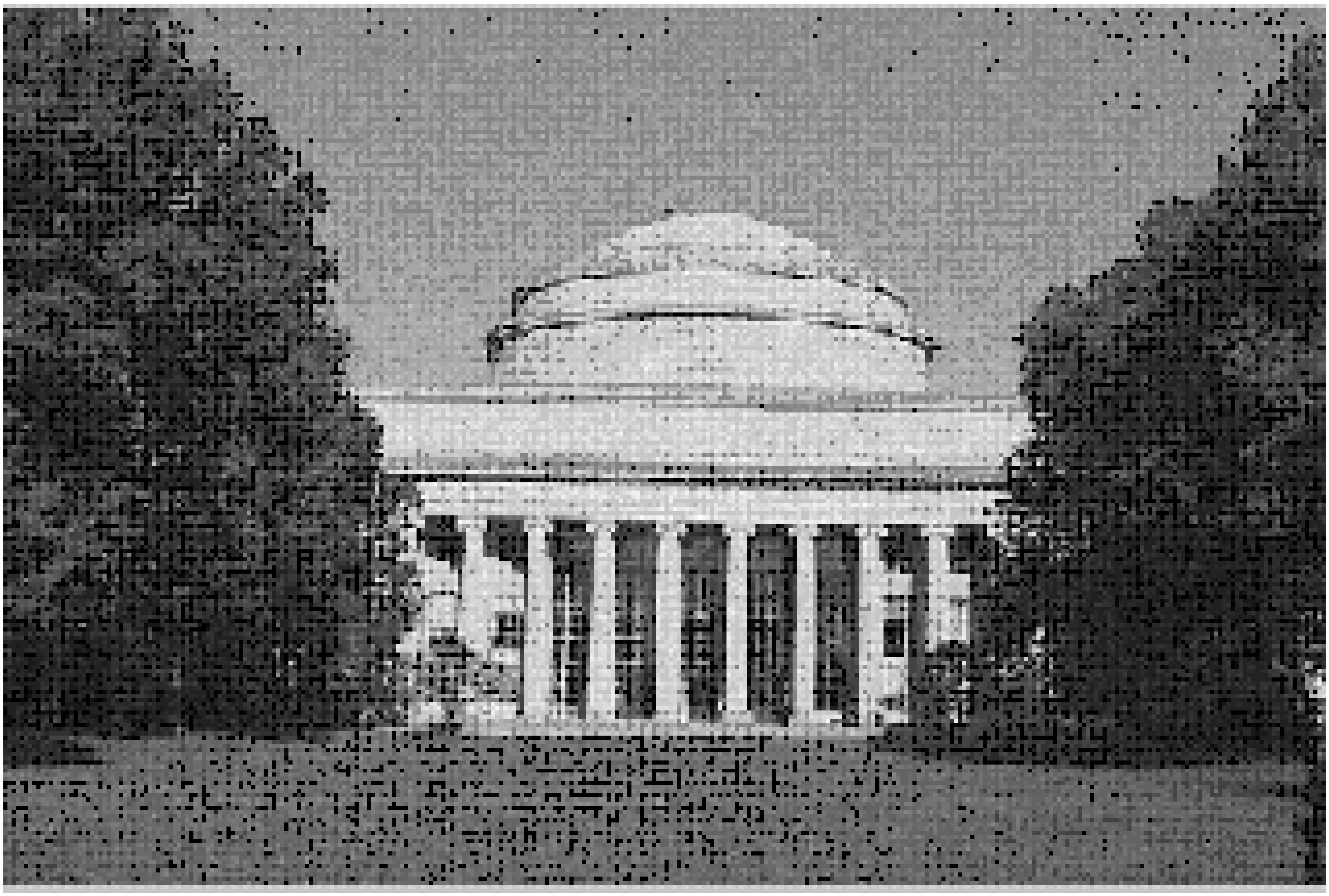
http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

An application of image pyramids: noise removal

Freeman

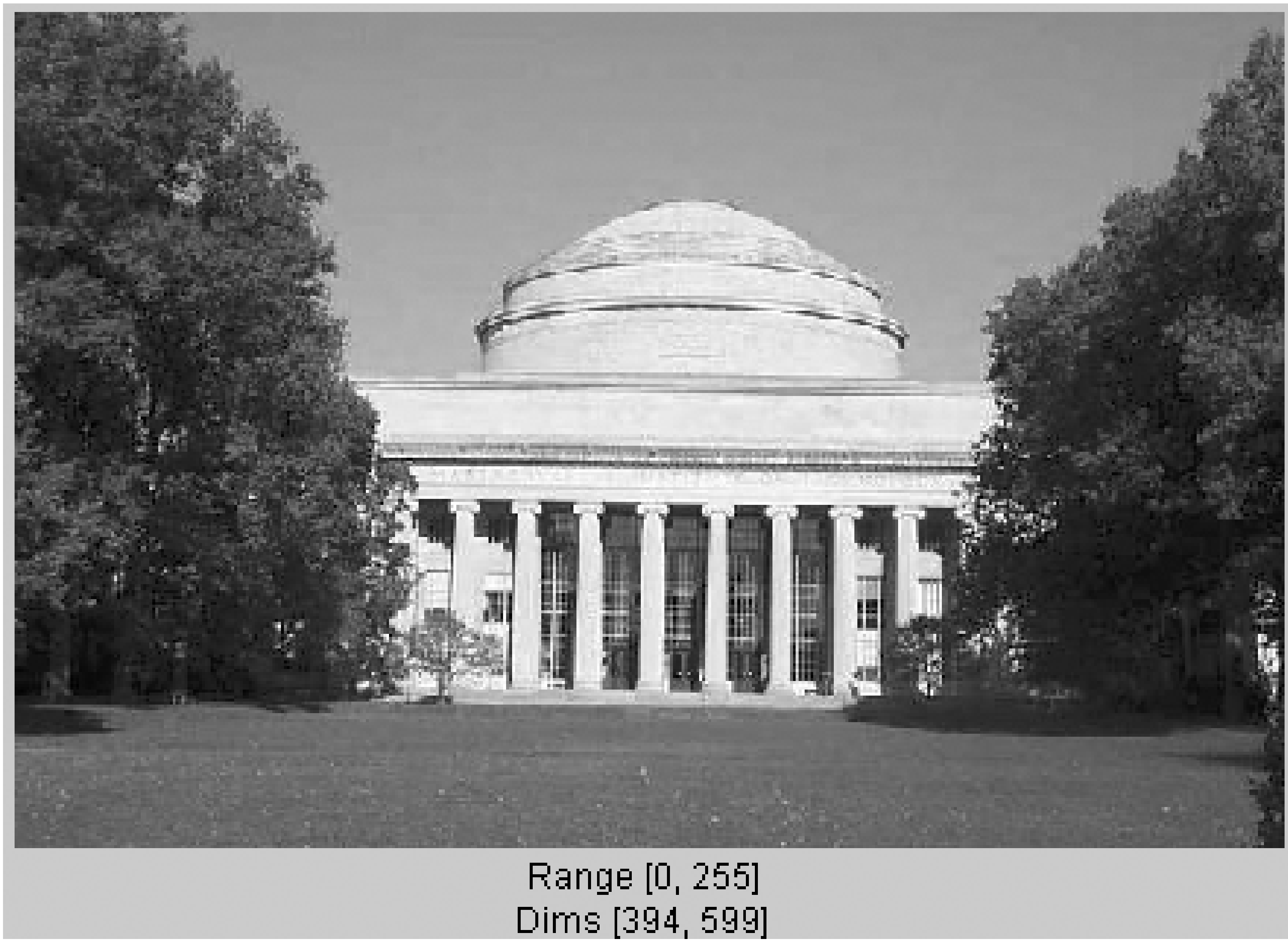
Image statistics (or, mathematically, how can you tell image from noise?)

Noisy image

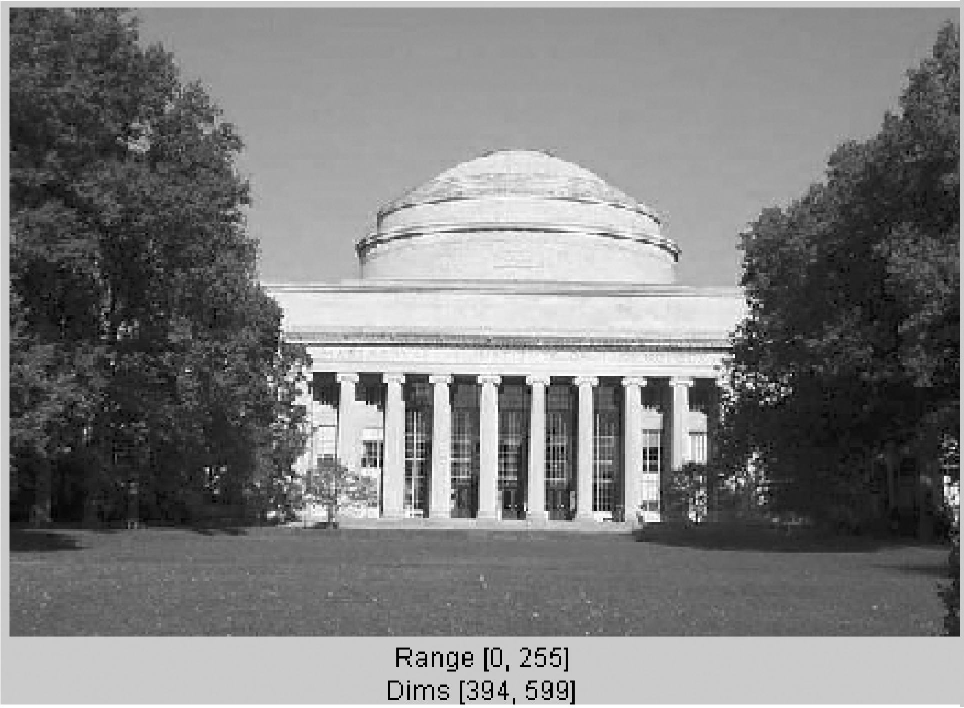


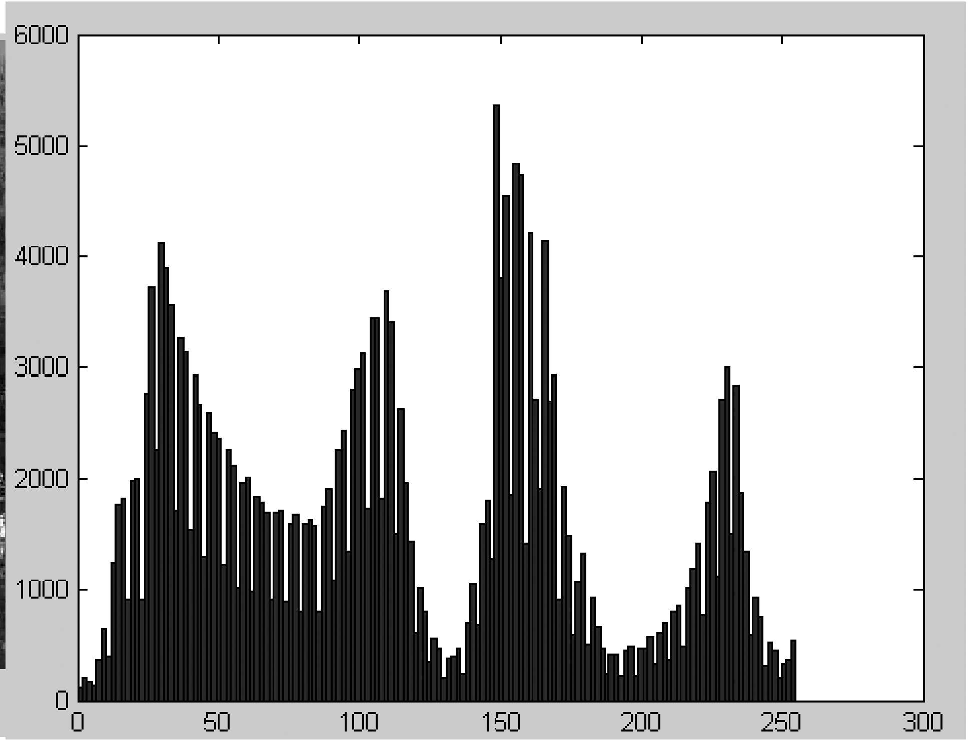
Freeman

Clean image



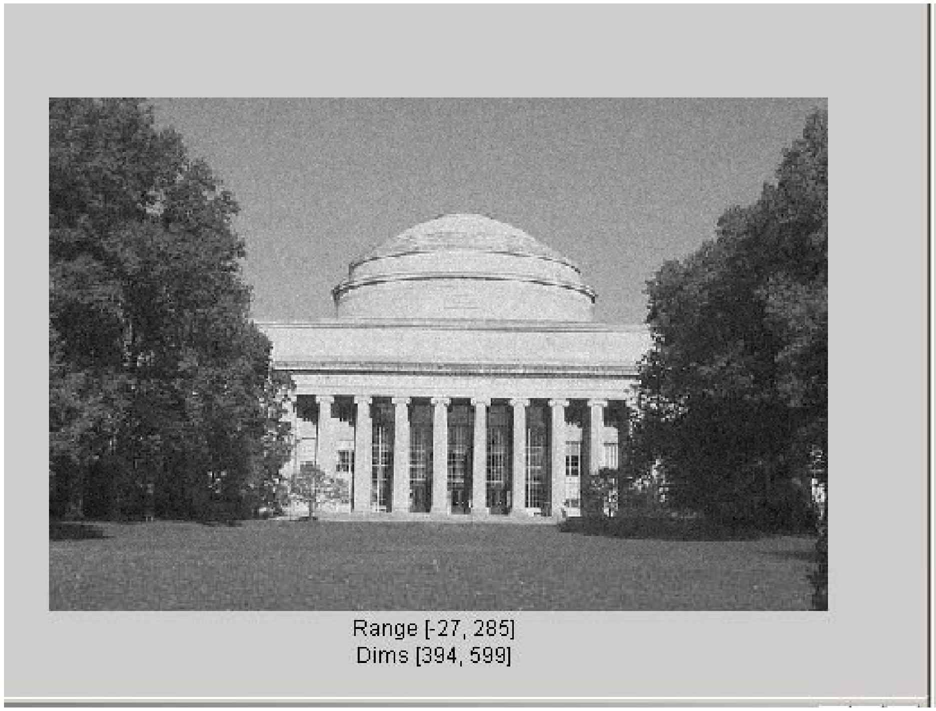
Pixel representation, image histogram

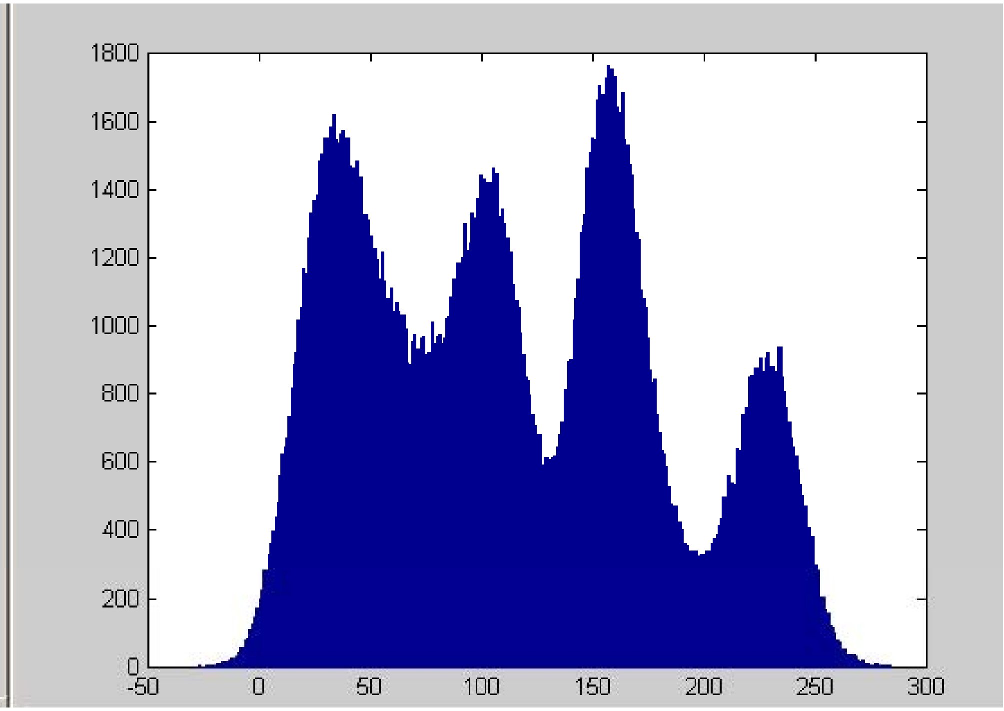




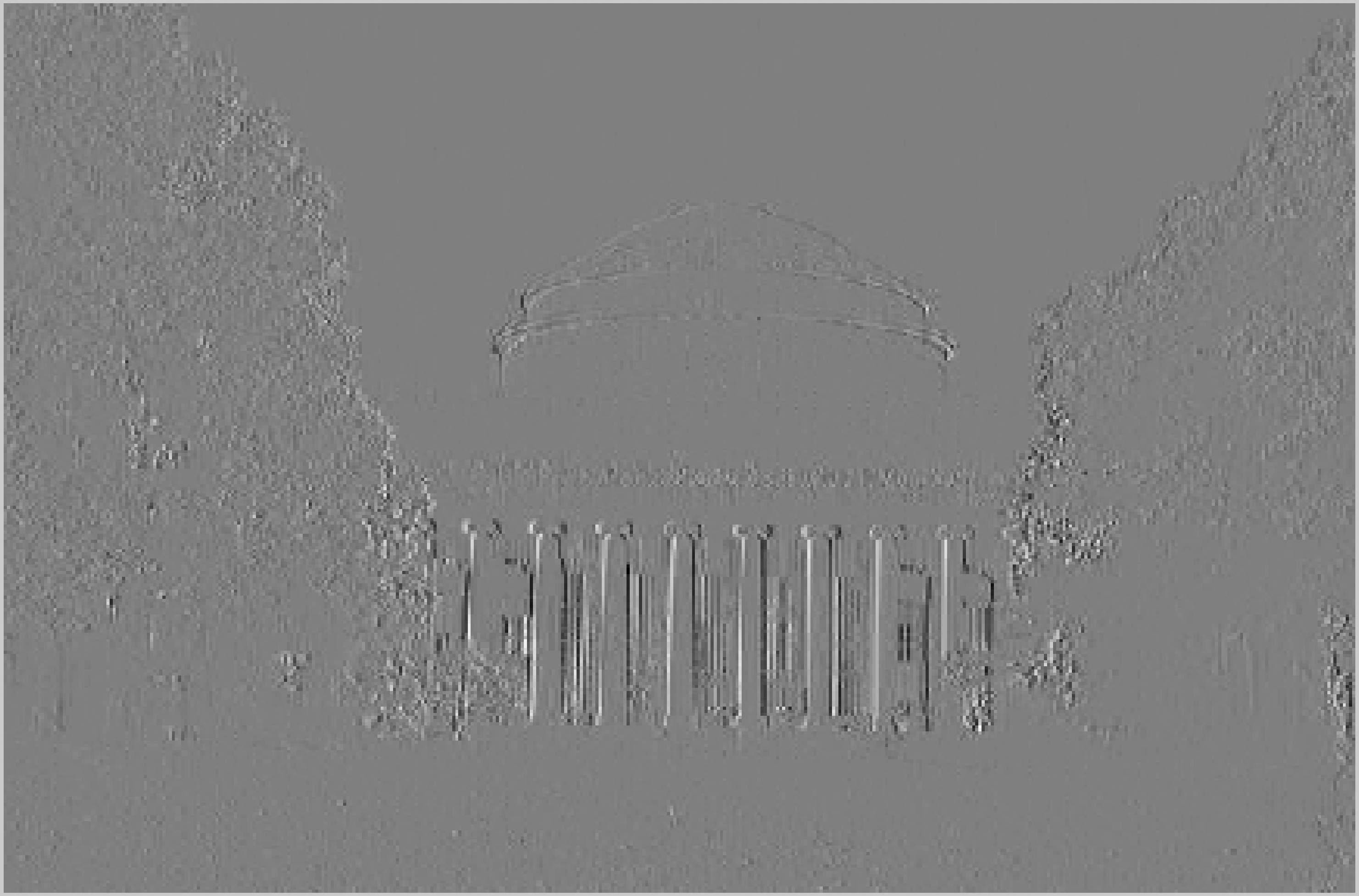
Freeman

Pixel representation, noisy image histogram

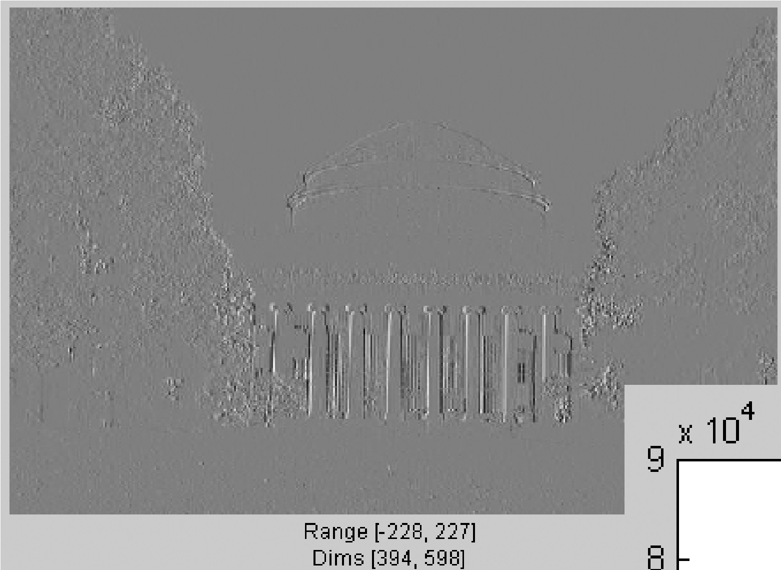




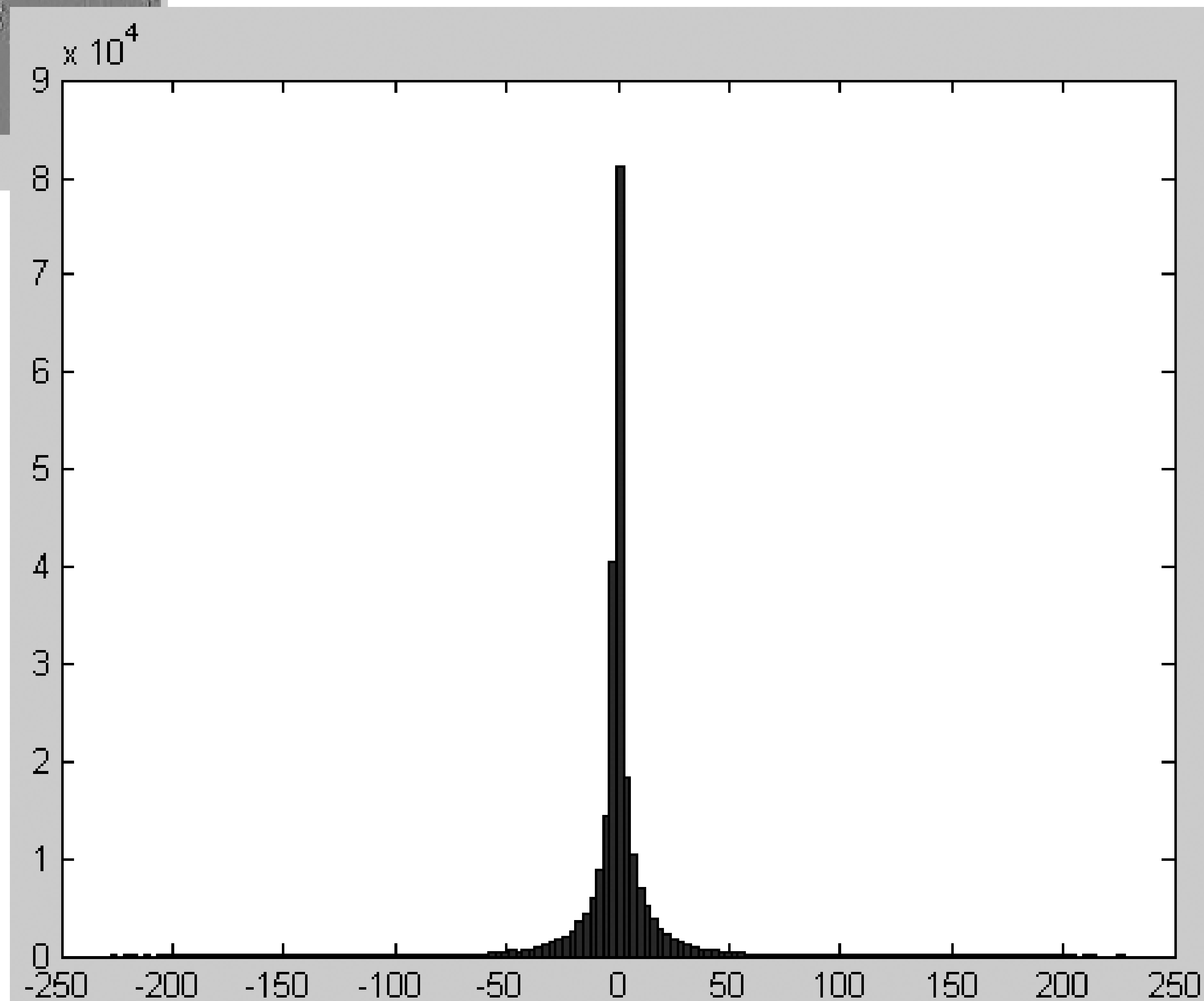
bandpass filtered image


Range [-228, 227]
Dims [394, 598]

Freeman

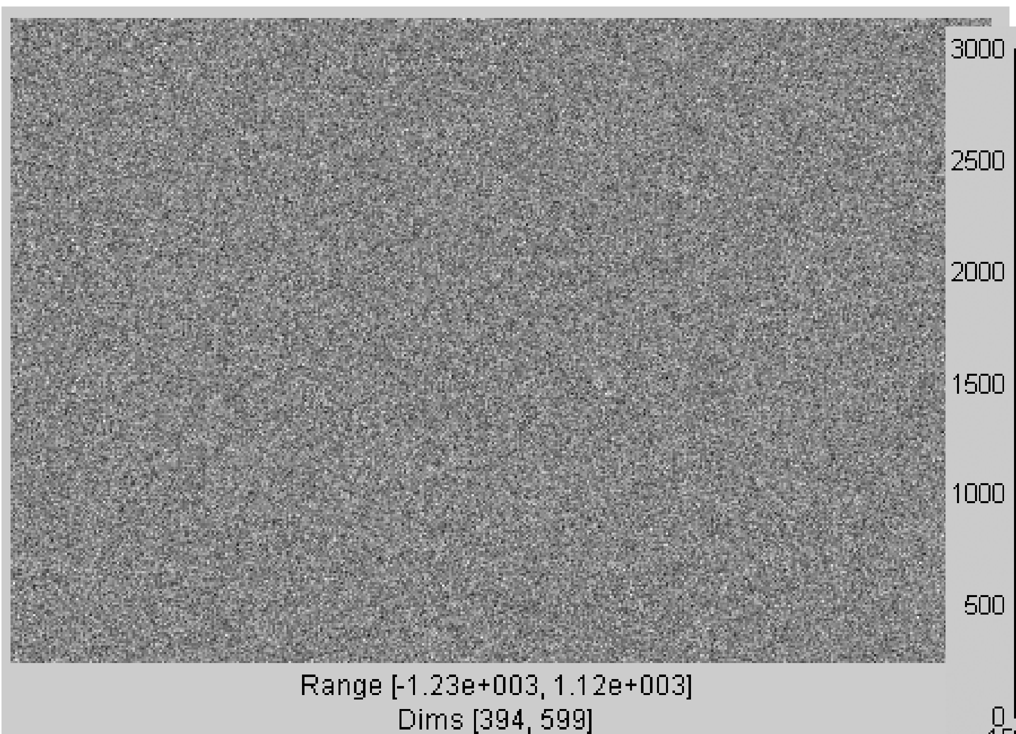


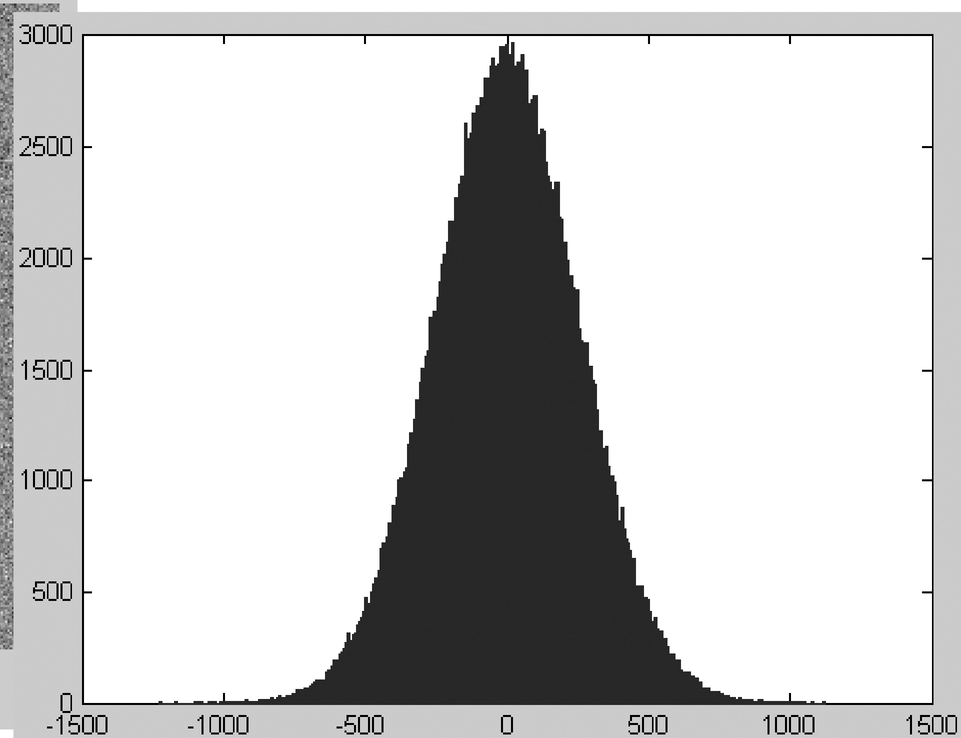
bandpassed representation image histogram



Freeman

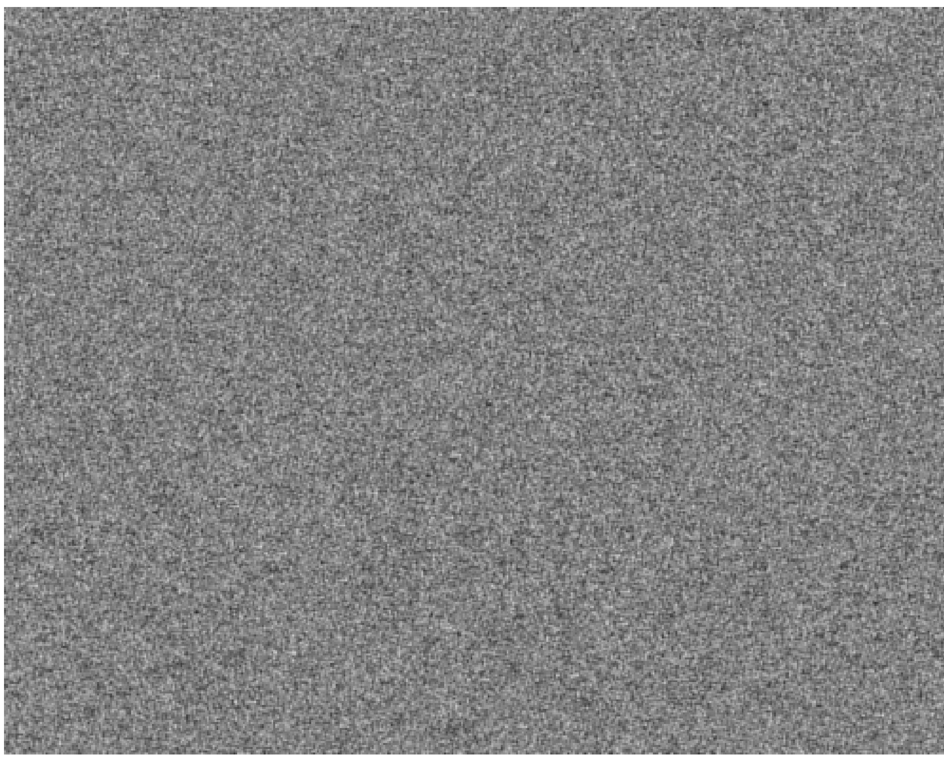
Pixel domain noise image and histogram

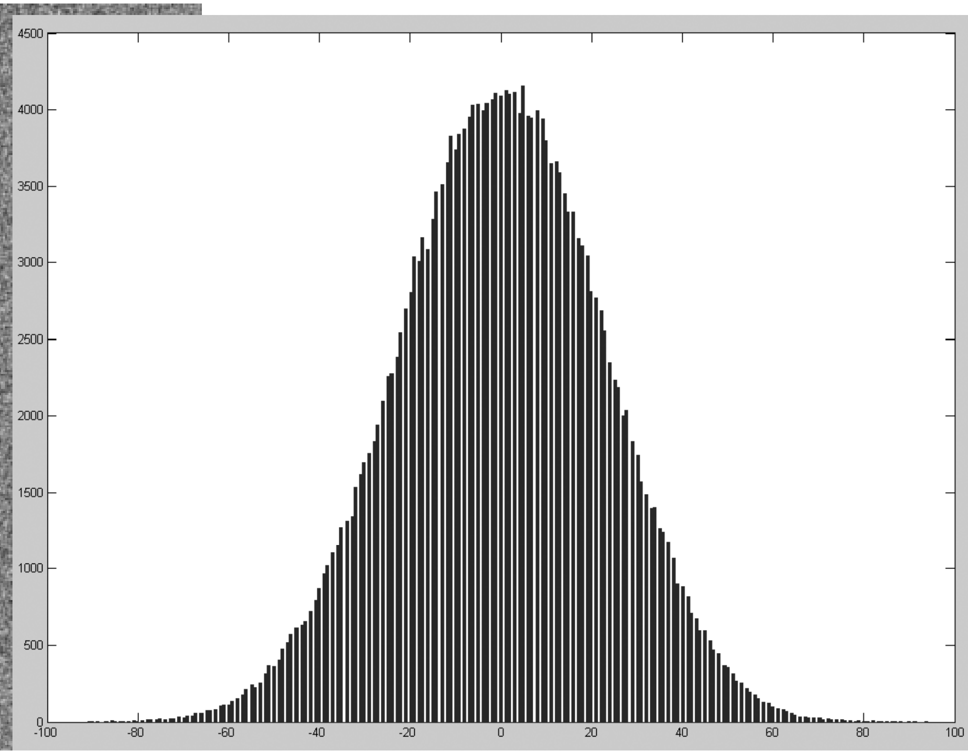




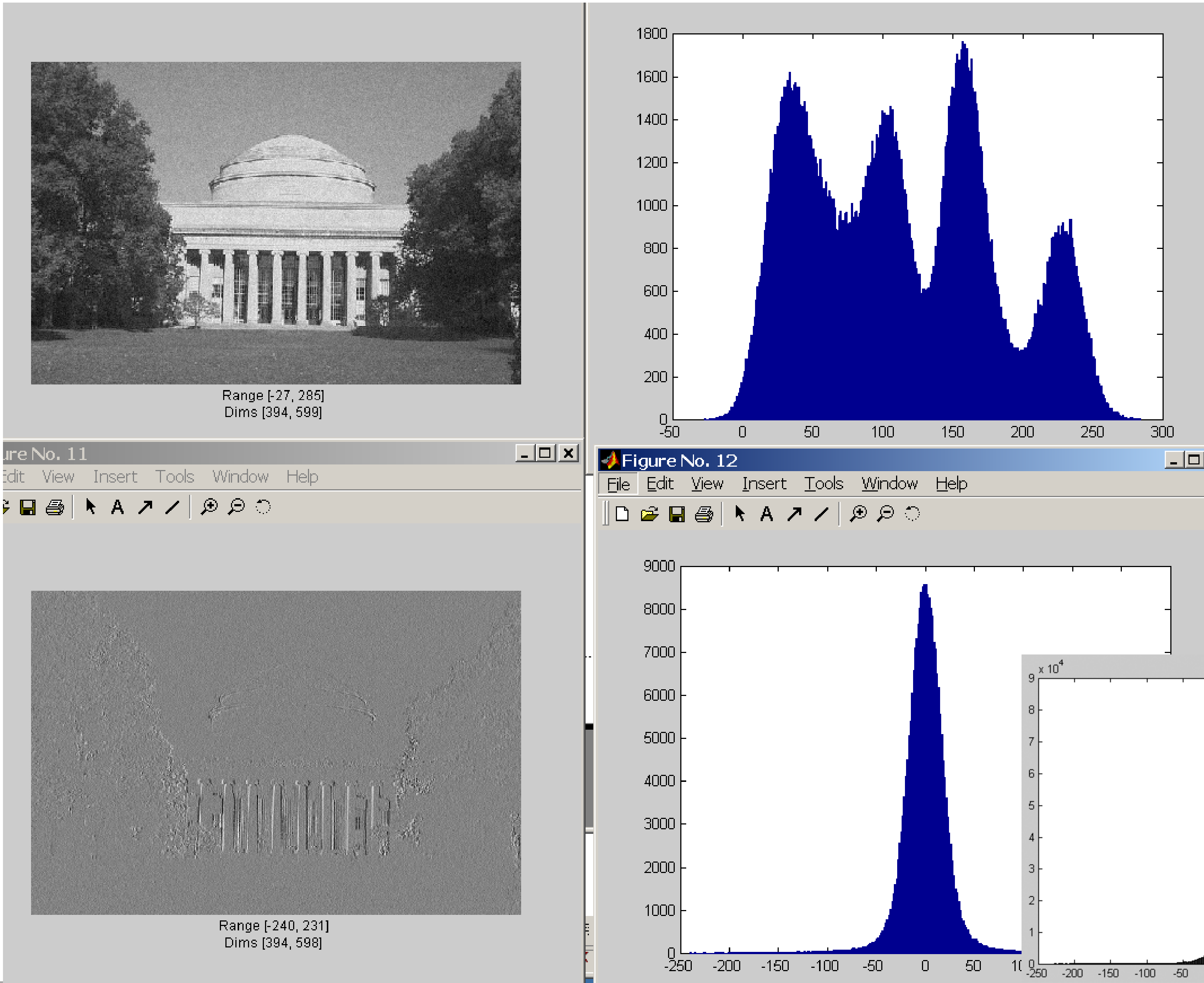
Freeman

Bandpass domain noise image and histogram

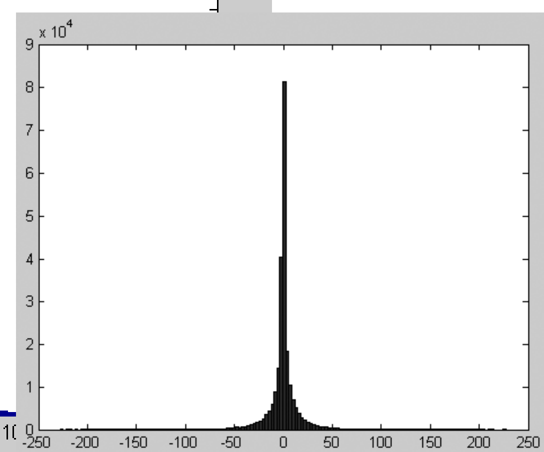




Noise-corrupted full-freq and bandpass images



But want the bandpass image histogram to look like this



Bayes theorem

$P(x, y) = P(x|y)\, P(y)$

By definition of conditional probability

so

Using that twice

$P(x|y)\, P(y) = P(y|x)\, P(x)$

and

$P(x|y) = P(y|x)\, P(x) \,/\, P(y)$

The parameters you want to estimate

What you observe

Likelihood function

Prior probability

Constant w.r.t. parameters x.

Freeman

Bayesian MAP estimator for clean bandpass coefficient values

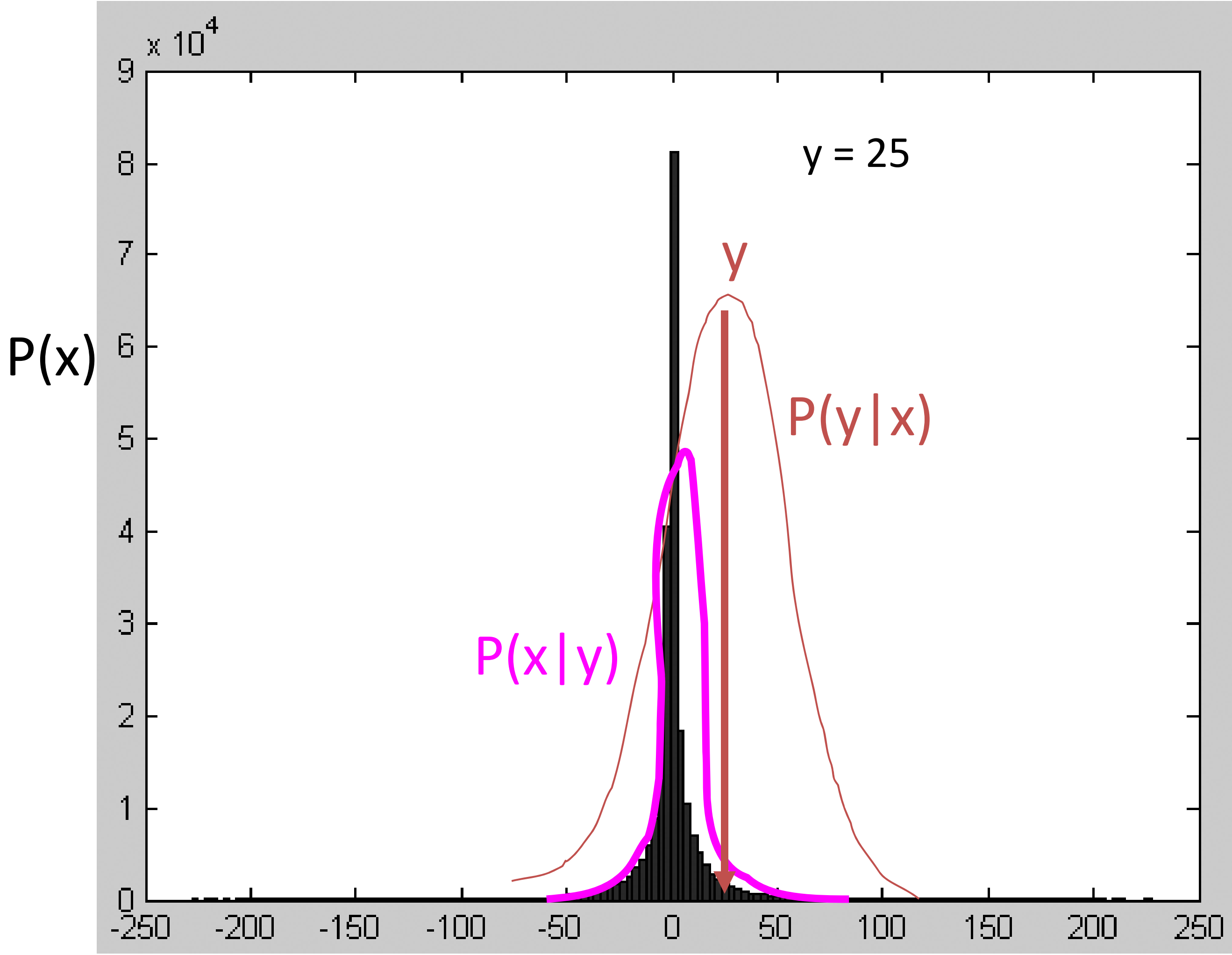
Let x = bandpassed image value before adding noise.
Let y = noise-corrupted observation.

By Bayes theorem

P(x|y) = k P(y|x) P(x)

P(y|x)

P(x|y)



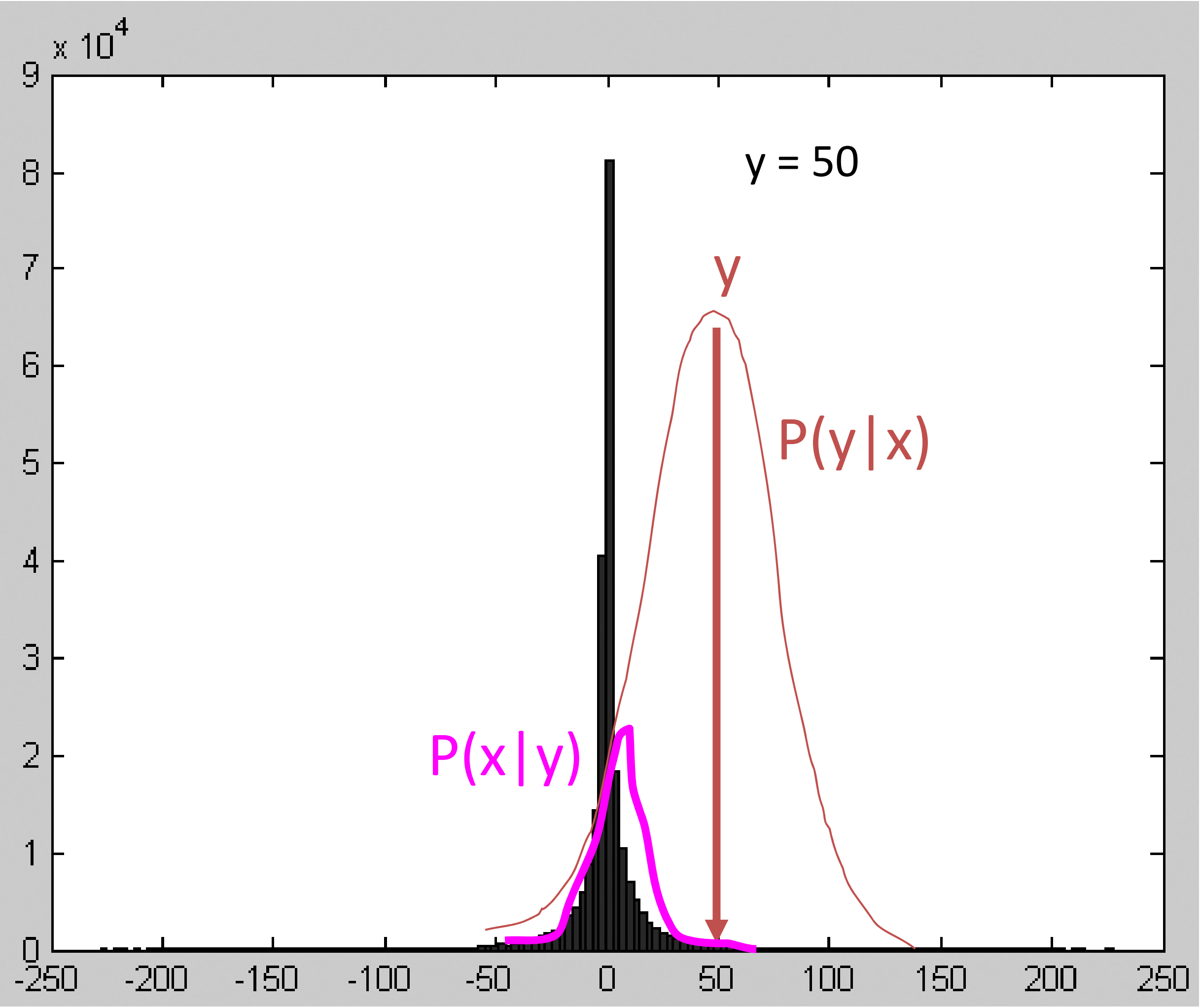
Freeman

Bayesian MAP estimator

Let x = bandpassed image value before adding noise.
Let y = noise-corrupted observation.

By Bayes theorem

P(x|y) = k P(y|x) P(x)



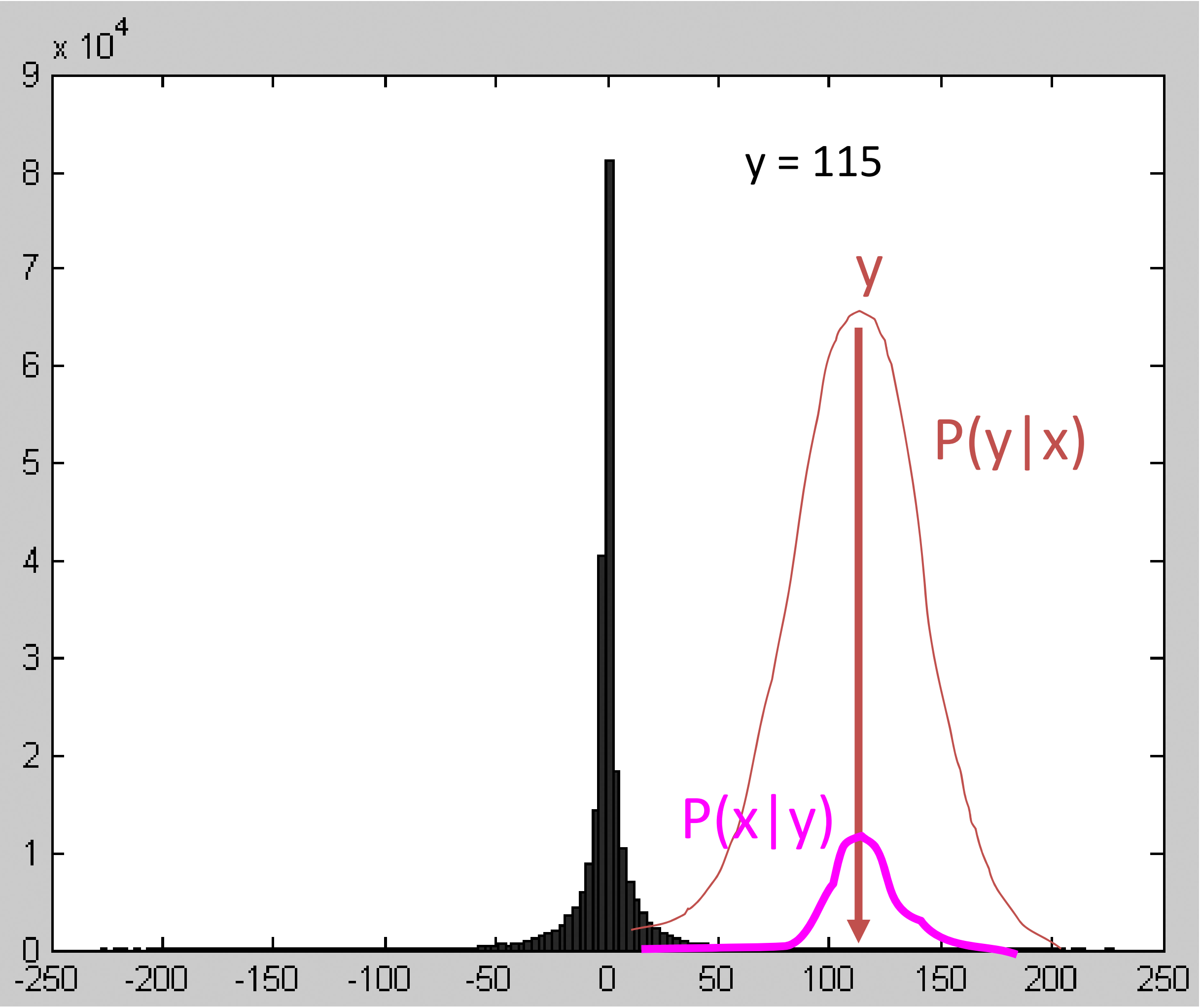
Freeman

Bayesian MAP estimator

Let x = bandpassed image value before adding noise.
Let y = noise-corrupted observation.

By Bayes theorem

P(x|y) = k P(y|x) P(x)



Freeman

MAP estimate, $\hat{x}$, as function of observed coefficient value, y

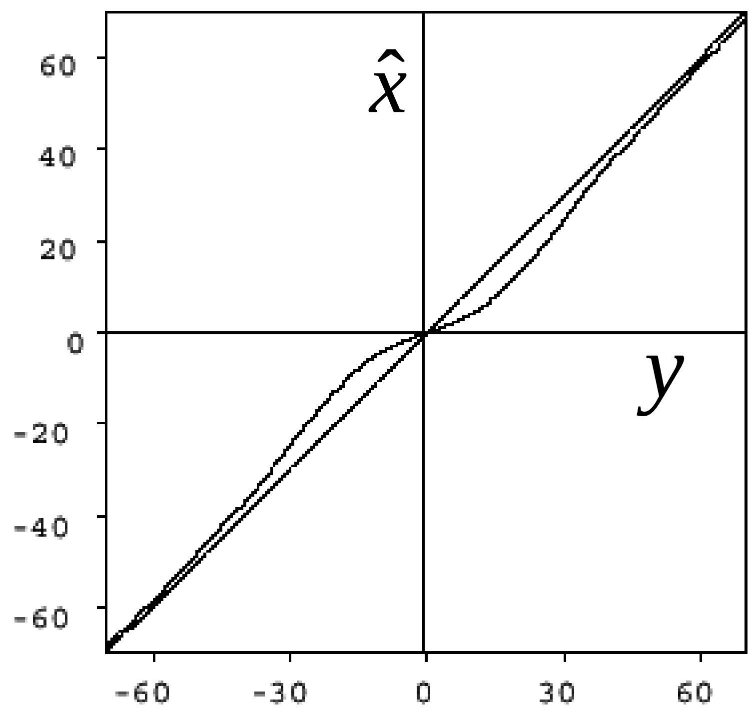
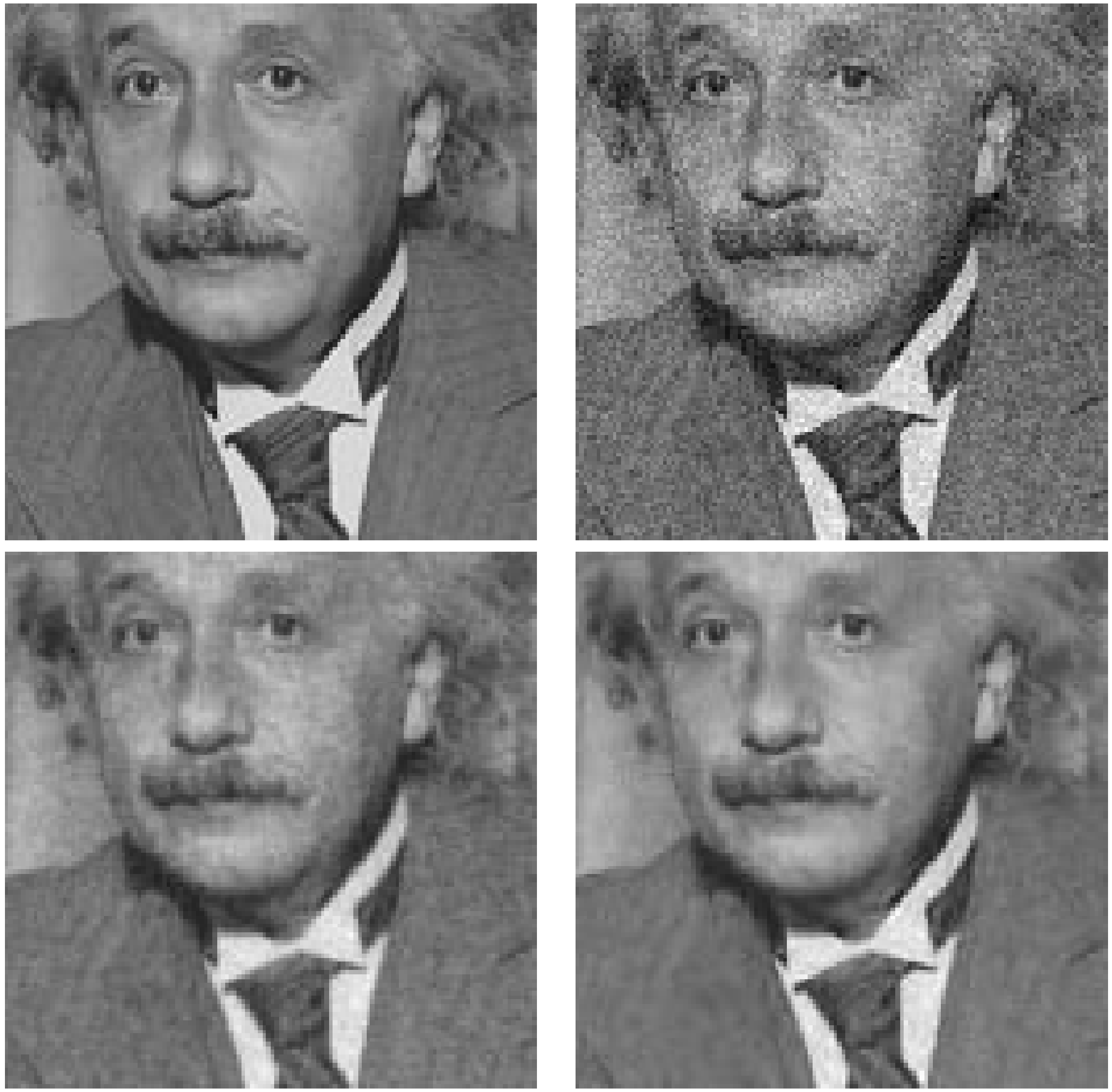


Figure 2: Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.

Simoncelli and Adelson, Noise Removal via Bayesian Wavelet Coring

 http://www-bcs.mit.edu/people/adelson/pub_pdfs/simoncelli_noise.pdf

Noise removal results



(a) (b) (c) (d)

Figure 4: Noise reduction example. (a) Original image (cropped). (b) Image contaminated with additive Gaussian white noise (SNR = 9.00dB). (c) Image restored using (semi-blind) Wiener filter (SNR = 11.88dB). (d) Image restored using (semi-blind) Bayesian estimator (SNR = 13.82dB).

Simoncelli and Adelson, Noise Removal via Bayesian Wavelet Coring

http://www-bcs.mit.edu/people/adelson/pub_pdfs/simoncelli_noise.pdf

Slide Credits

- Bill Freeman
- and others, as noted…

More on statistics of natural scenes

- Olshausen and Field:
  - Natural Image Statistics and Efficient Coding, https://redwood.berkeley.edu/bruno/papers/stirling.ps.Z
  - Relations between the statistics of natural images and the response properties of cortical cells.
  - http://redwood.psych.cornell.edu/papers/field_87.pdf
- Aude Olivia:
  - http://cvcl.mit.edu/SUNSlides/9.912-CVC-ImageAnalysis-web.pdf

Today

- Review of Fourier Transform
- Sampling and Aliasing
- Image Pyramids
- Applications: Blending and noise removal

Next time: Feature Detection and Matching

- Local features
- Pyramids for invariant feature detection
- Local descriptors
- Matching

Appendix: Steering

Simple example of steerable filter

$$G_1^{60^\circ}(x) = \cos(60^\circ)G_1^{0^\circ}(x) + \sin(60^\circ)G_1^{90^\circ}(x)$$

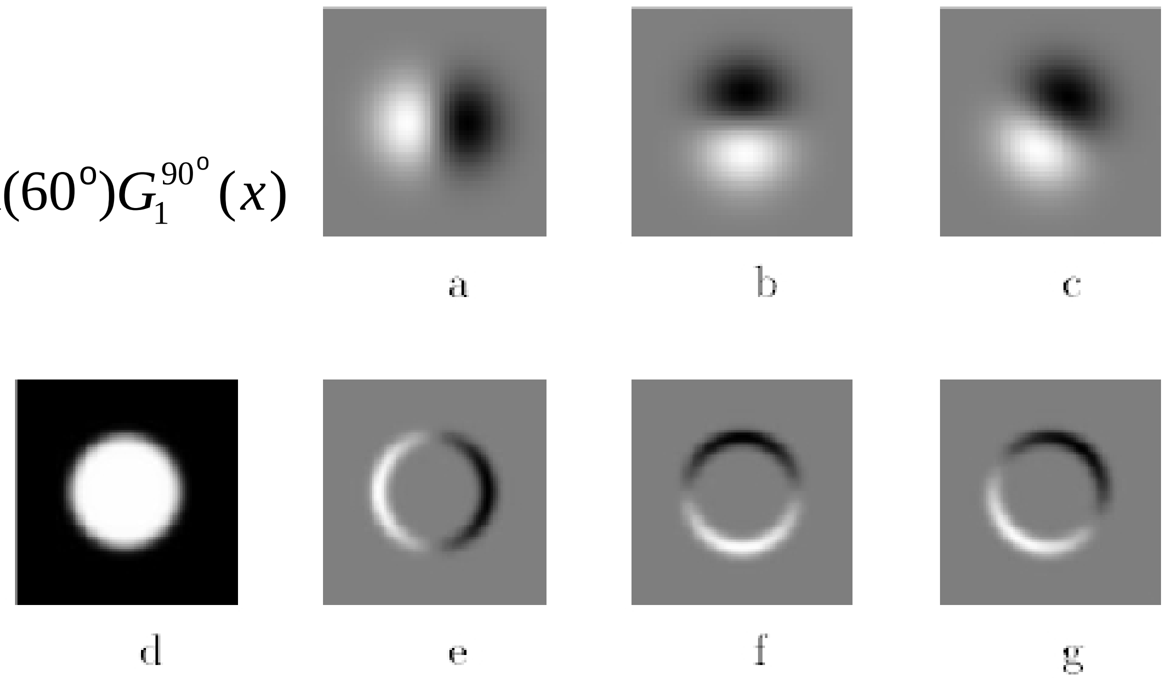


Fig. 1. Example of steerable filters: (a) $G_1^{0^\circ}$ first derivative with respect to x (horizontal) of a Gaussian; (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^θ, which is an arbitrary rotation of the first derivative of a Gaussian; (c) $G_1^{60^\circ}$, formed by $\frac{1}{2}G_1^{0^\circ} + \frac{\sqrt{3}}{2}G_1^{90^\circ}$. The same linear combinations used to synthesize G_1^θ from the basis filters will also synthesize the response of an image to G_1^θ from the responses of the image to the basis filters; (d) image of circular disk; (e) $G_1^{0^\circ}$ (at a smaller scale than pictured above) convolved with the disk (d); (f) $G_1^{90^\circ}$ convolved with (d); (g) $G_1^{60^\circ}$ convolved with (d), obtained from $\frac{1}{2}$ (image (e)) $+ \frac{\sqrt{3}}{2}$ (image (f)).

Freeman

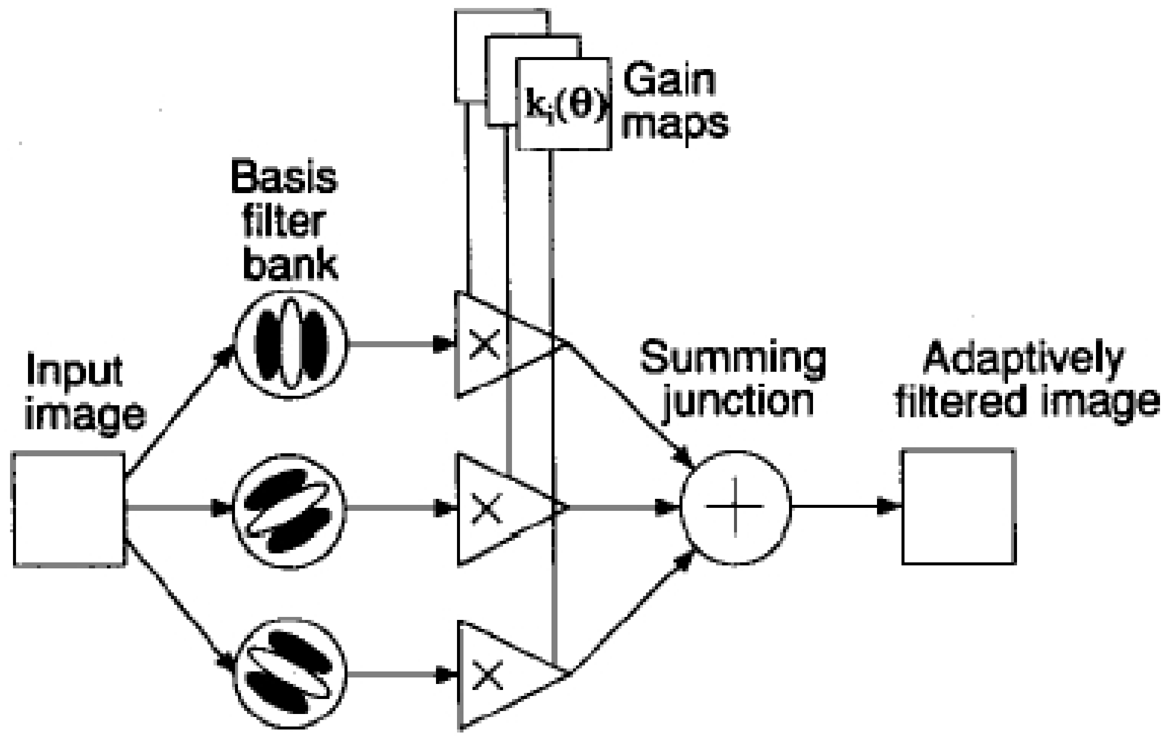


Fig. 3. Steerable filter system block diagram. A bank of dedicated filters process the image. Their outputs are multiplied by a set of gain maps that adaptively control the orientation of the synthesized filter.

Steering theorem

$$f(r,\phi) = \sum_{n=-N}^{N} a_n(r) e^{in\phi}$$

Theorem 2: Let T be the number of nonzero coefficients $a_n(r)$ for a function $f(r,\phi)$ expandable in the form of (9). Then, the minimum number of basis functions sufficient to steer $f(r,\phi)$ by (11) is T, i.e., M in (11) must be $>= T$.

Theorem 1: The steering condition (8) holds for functions expandable in the form of (9) if and only if the interpolation functions $k_j(\theta)$ are solutions of

$$\begin{pmatrix} 1 \\ e^{i\theta} \\ \vdots \\ e^{iN\theta} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ e^{i\theta_1} & e^{i\theta_2} & \cdots & e^{i\theta_M} \\ \vdots & \vdots & & \vdots \\ e^{iN\theta_1} & e^{iN\theta_2} & \cdots & e^{iN\theta_M} \end{pmatrix} \begin{pmatrix} k_1(\theta) \\ k_2(\theta) \\ \vdots \\ k_M(\theta) \end{pmatrix}.$$

Originally written as sines & cosines

Freeman

Steering theorem for polynomials

For an Nth order polynomial with even symmetry N+1 basis functions are sufficient.

Theorem 3: Let $f(x, y) = W(r)P_N(x, y)$, where $W(r)$ is an arbitrary windowing function, and $P_N(x, y)$ is an Nth order polynomial in x and y, whose coefficients may depend on r. Linear combinations of $2N + 1$ basis functions are sufficient to synthesize $f(x, y) = W(r)P_N(x, y)$ rotated to any angle. Equation (10) gives the interpolation functions $k_j(\theta)$. If $P_N(x, y)$ contains only even [odd] order terms (terms $x^n y^m$ for $n + m$ even [odd]), then $N + 1$ basis functions are sufficient, and (10) can be modified to contain only the even [odd] numbered rows (counting from zero) of the left-hand side column vector and the right-hand side matrix.

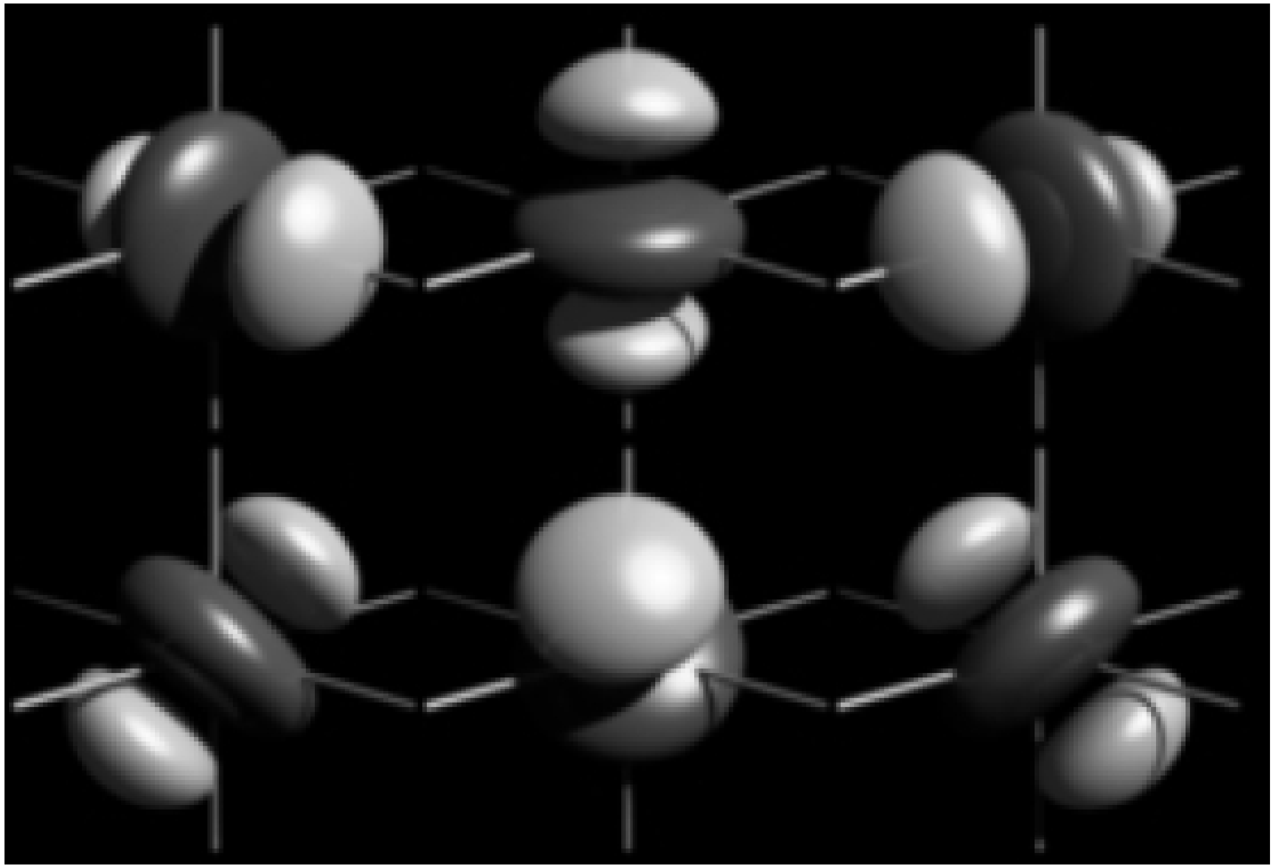


Figure 2-10: Example of a three-dimensional steerable filter. Surfaces of constant value are shown for the six basis filters of a second derivative of a three-dimensional Gaussian. Linear combinations of these six filters can synthesize the filter rotated to any orientation in three-space. Such three-dimensional steerable filters are useful for analysis and enhancement of motion sequences or volumetric image data, such as MRI or CT data. For discussions of steerable filters in three or more dimensions, see [59, 58, 33, 89]. (Martin Friedmann rendered this image with the Thingworld program).

Freeman

Steerable quadrature pairs

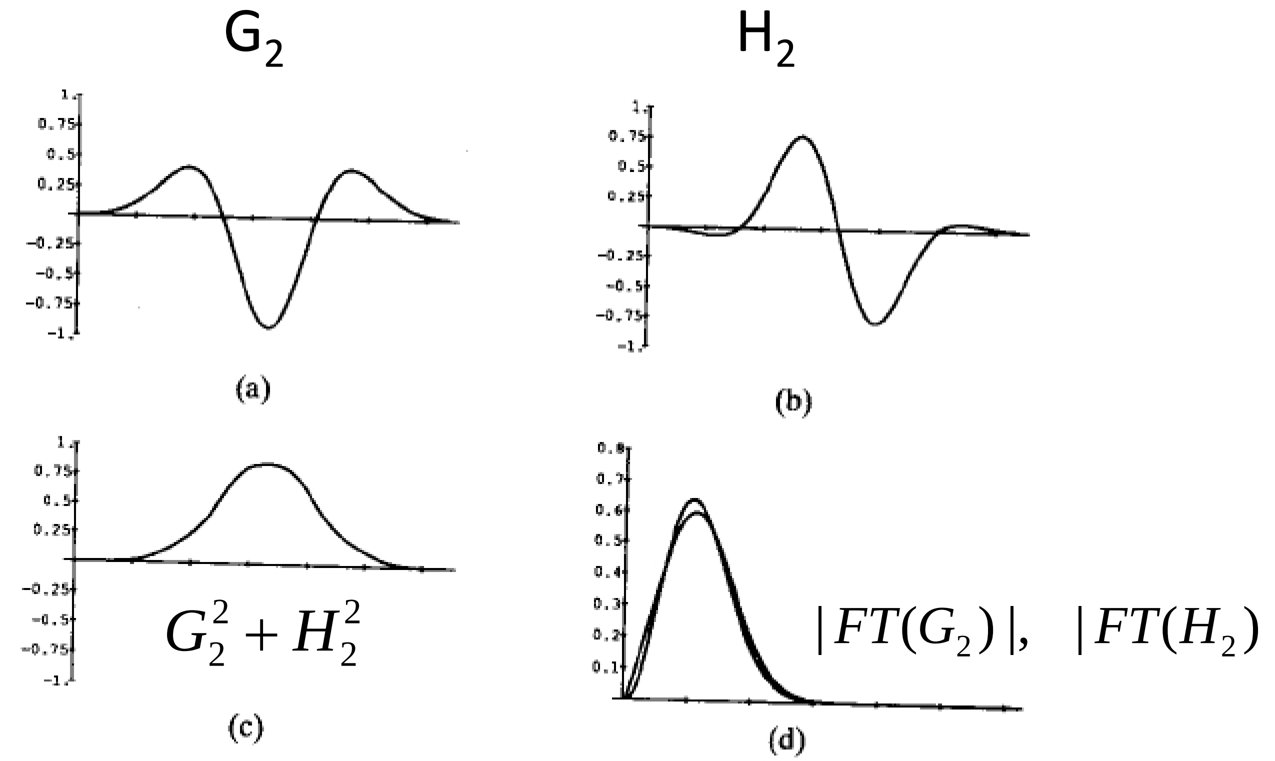


Fig. 4. (a) G_2, second derivative of Gaussian (in one dimension); (b) H_2, fit of third order polynomial (times Gaussian) to the Hilbert transform of (a); (c) energy measure: $(G_2)^2 + (H_2)^2$; (d) magnitudes of Fourier transforms of (a) and (b).

Freeman

How quadrature pair filters work

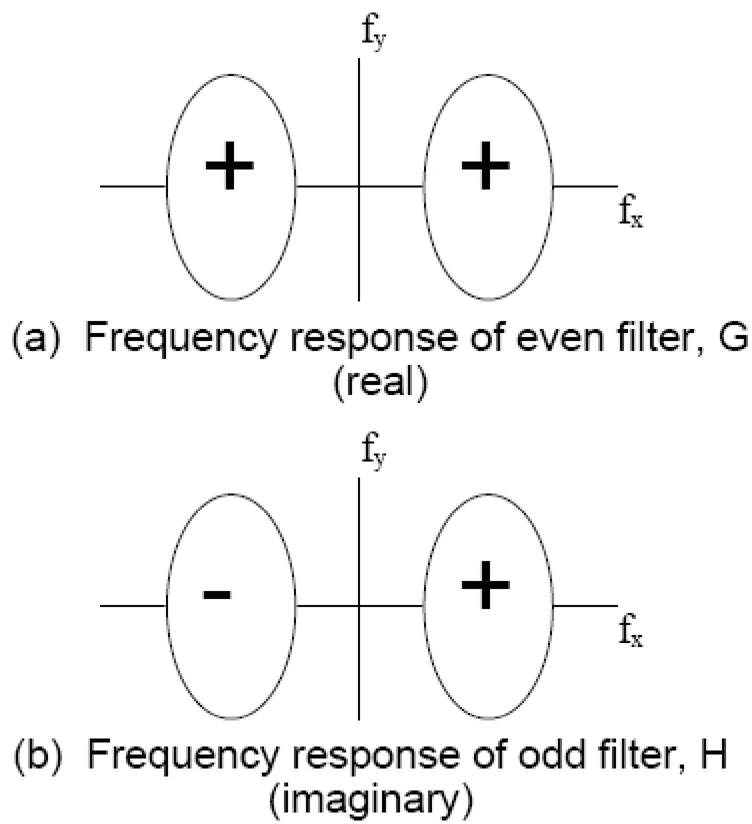


Figure 3-5: Frequency content of two bandpass filters in quadrature. (a) even phase filter, called G in text, and (b) odd phase filter, H. Plus and minus sign illustrate relative sign of regions in the frequency domain. See Fig. 3-6 for calculation of the frequency content of the energy measure derived from these two filters.

Freeman

How quadrature pair filters work

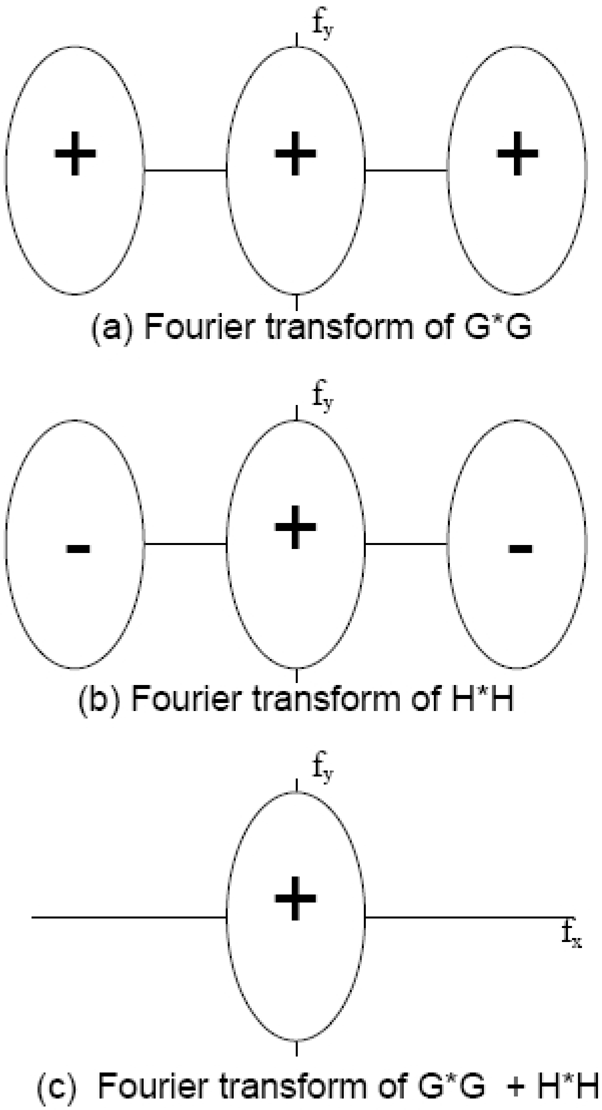
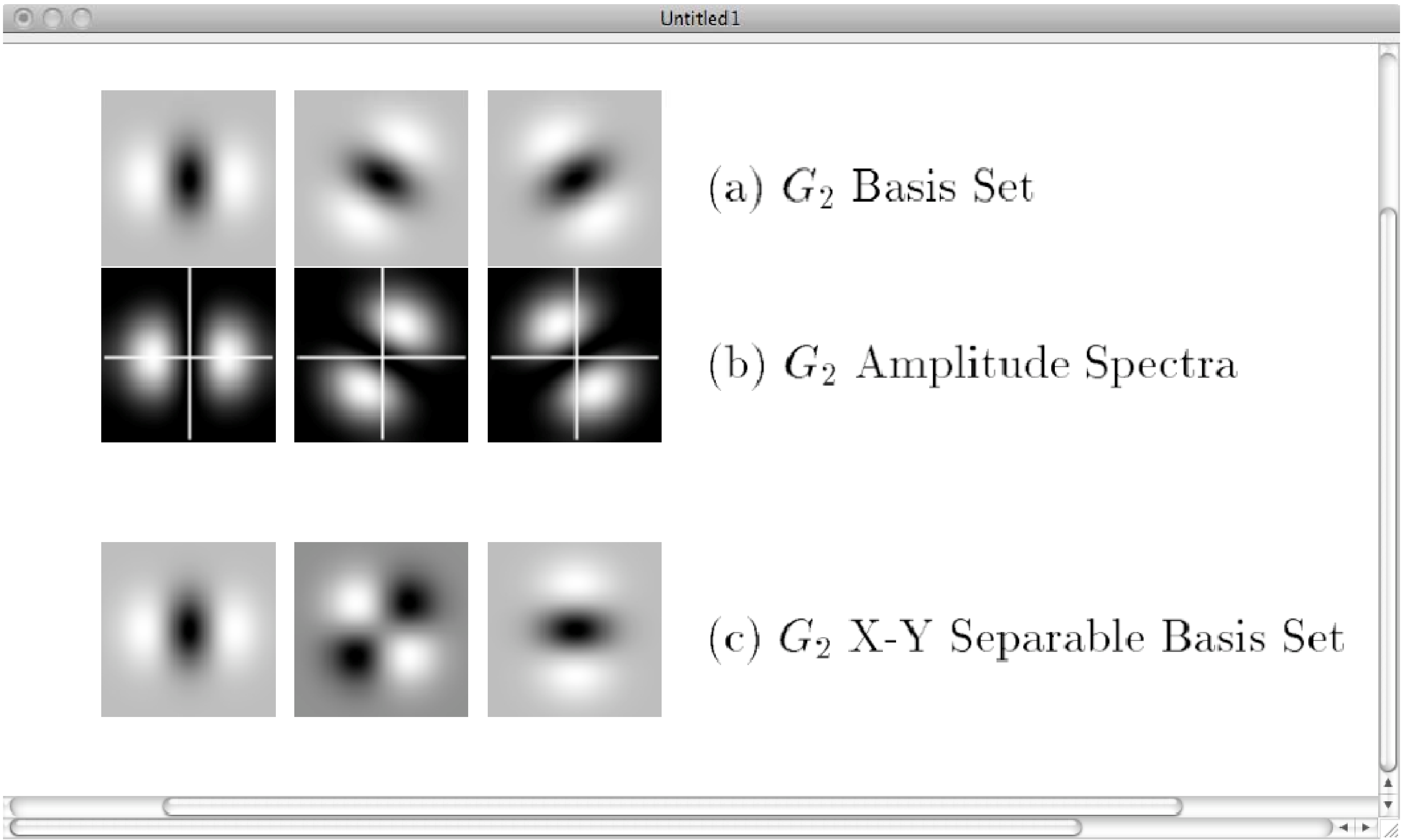
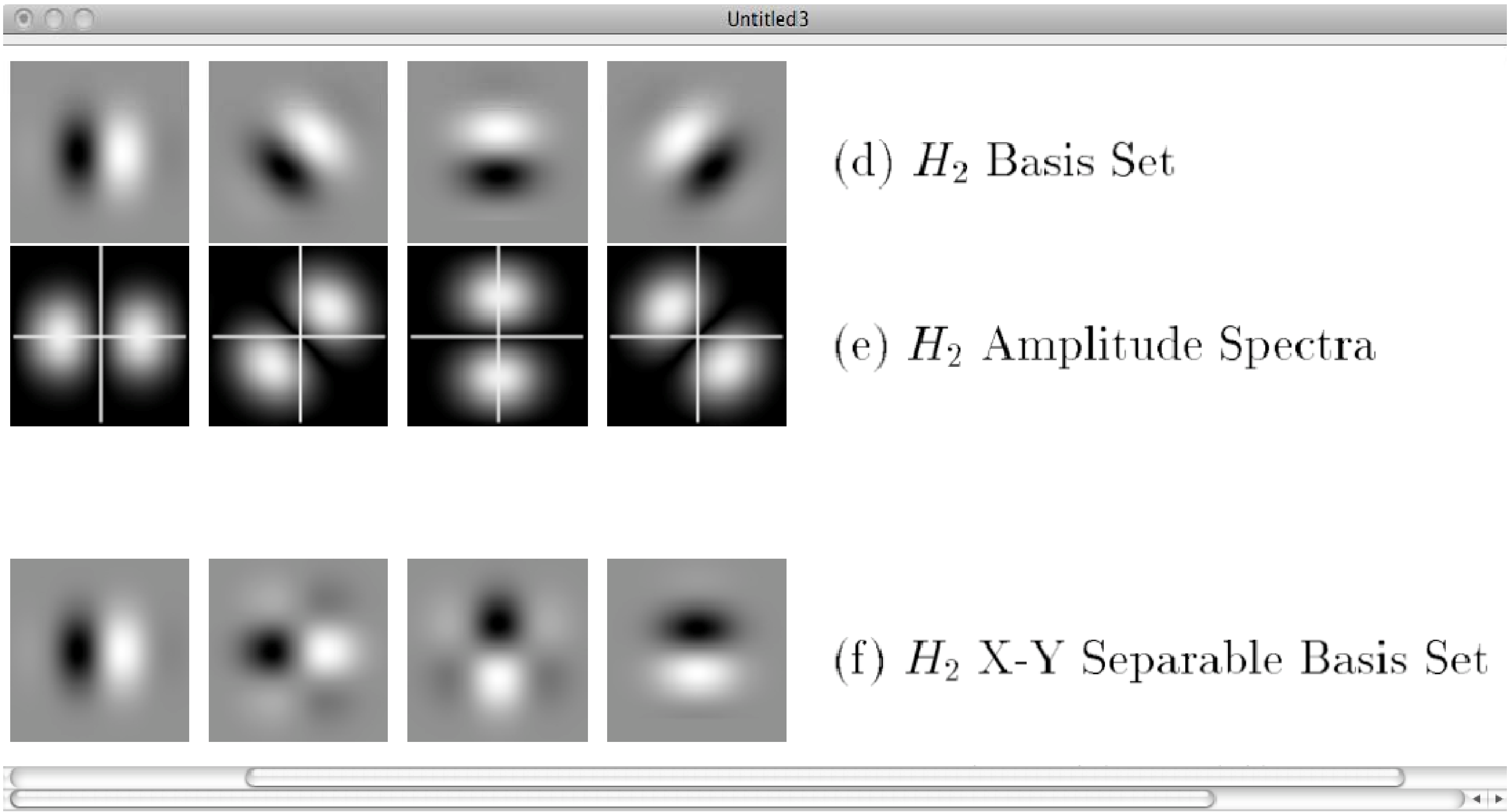


Figure 3-6: Derivation of energy measure frequency content for the filters of Fig. 3-5. (a) Fourier transform of $G * G$. (b) Fourier transform of $H * H$. Each squared response has 3 lobes in the frequency domain, arising from convolution of the frequency domain responses. The center lobe is modulated down in frequency while the two outer lobes are modulated up. (There are two sign changes which combine to give the signs shown in (b). To convolve H with itself, we flip it in f_x and f_y, which interchanges the + and − lobes of Fig. 3-5 (b). Then we slide it over an unflipped version of itself, and integrate the product of the two. That operation will give positive outer lobes, and a negative inner lobe. However, H has an imaginary frequency response, so multiplying it by itself gives an extra factor of −1, which yields the signs shown in (b)). (c) Fourier transform of the energy measure, $G * G + H * H$. The high frequency lobes cancel, leaving only the baseband spectrum, which has been demodulated in frequency from the original bandpass response. This spectrum is proportional to the sum of the auto-correlation functions of either lobe of Fig. 3-5 (a) and either lobe of Fig. 3-5 (b).


Untitled1
(a) G_2 Basis Set
(b) G_2 Amplitude Spectra
(c) G_2 X-Y Separable Basis Set

Freeman


Untitled3
(d) H_2 Basis Set
(e) H_2 Amplitude Spectra
(f) H_2 X-Y Separable Basis Set

Orientation analysis

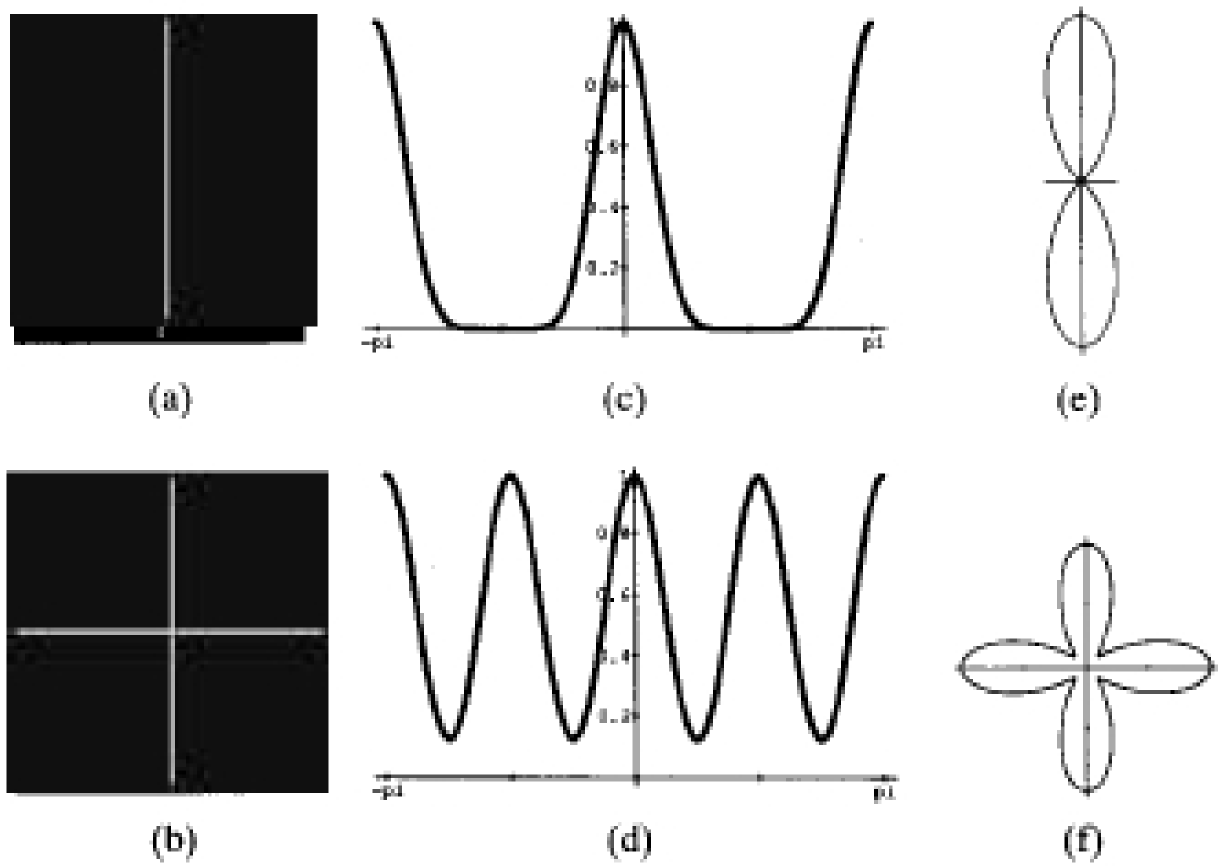
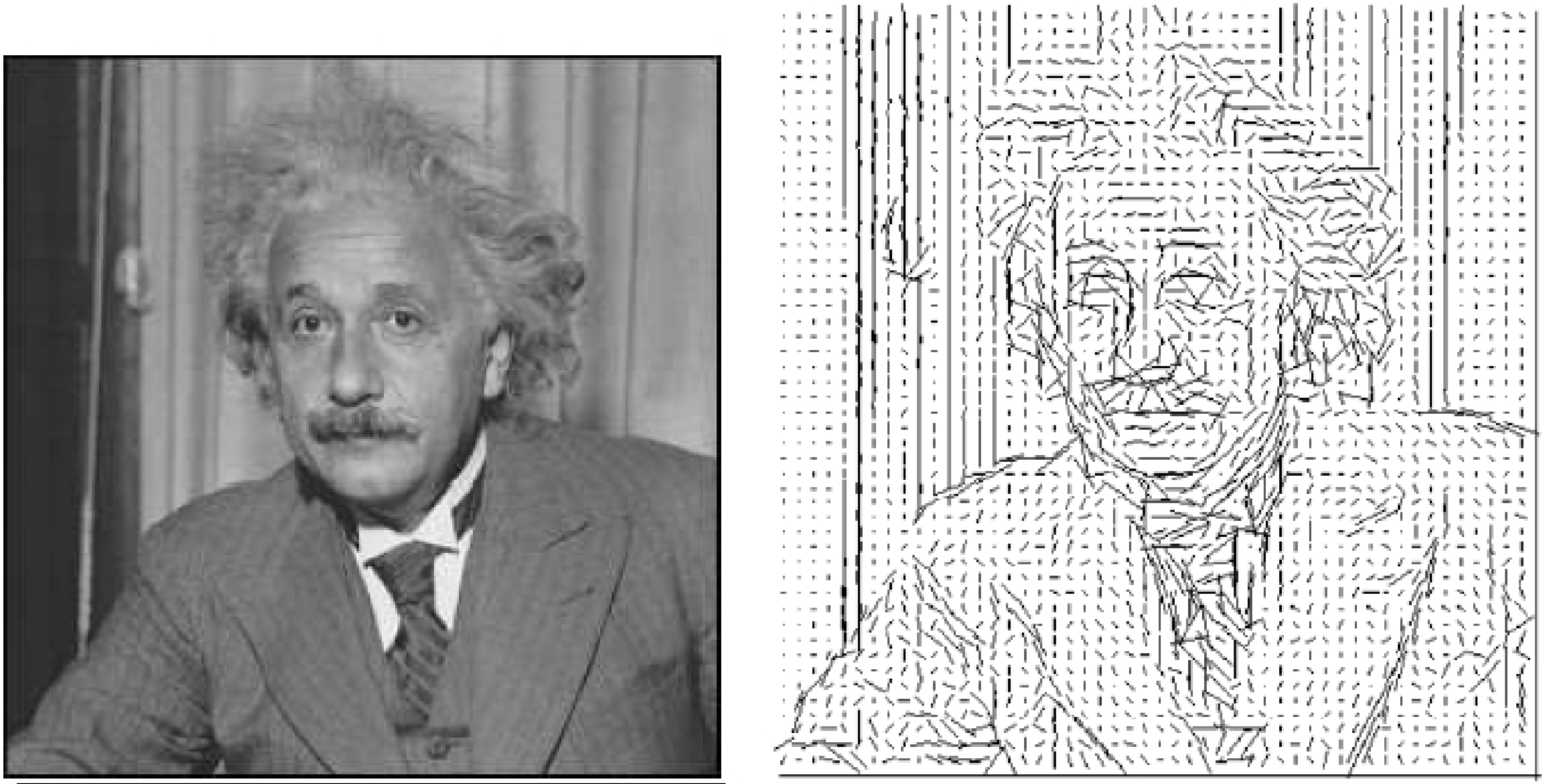


Fig. 9. Test images of (a) vertical line and (b) intersecting lines; (c) and (d) oriented energy as a function of angle at the centers of test images (a) and (b). Oriented energy was measured using the G_4, H_4 quadrature steerable pair; (e) and (f) polar plots of (c) and (d).

Freeman

Orientation analysis



(a) (b)

Fig. 8. (a) Original image of Einstein; (b) orientation map of (a) made using the lowest order terms in a Fourier series expansion for the oriented energy as measured with G_2 and H_2. Table XI gives the formulas for these terms.

Free

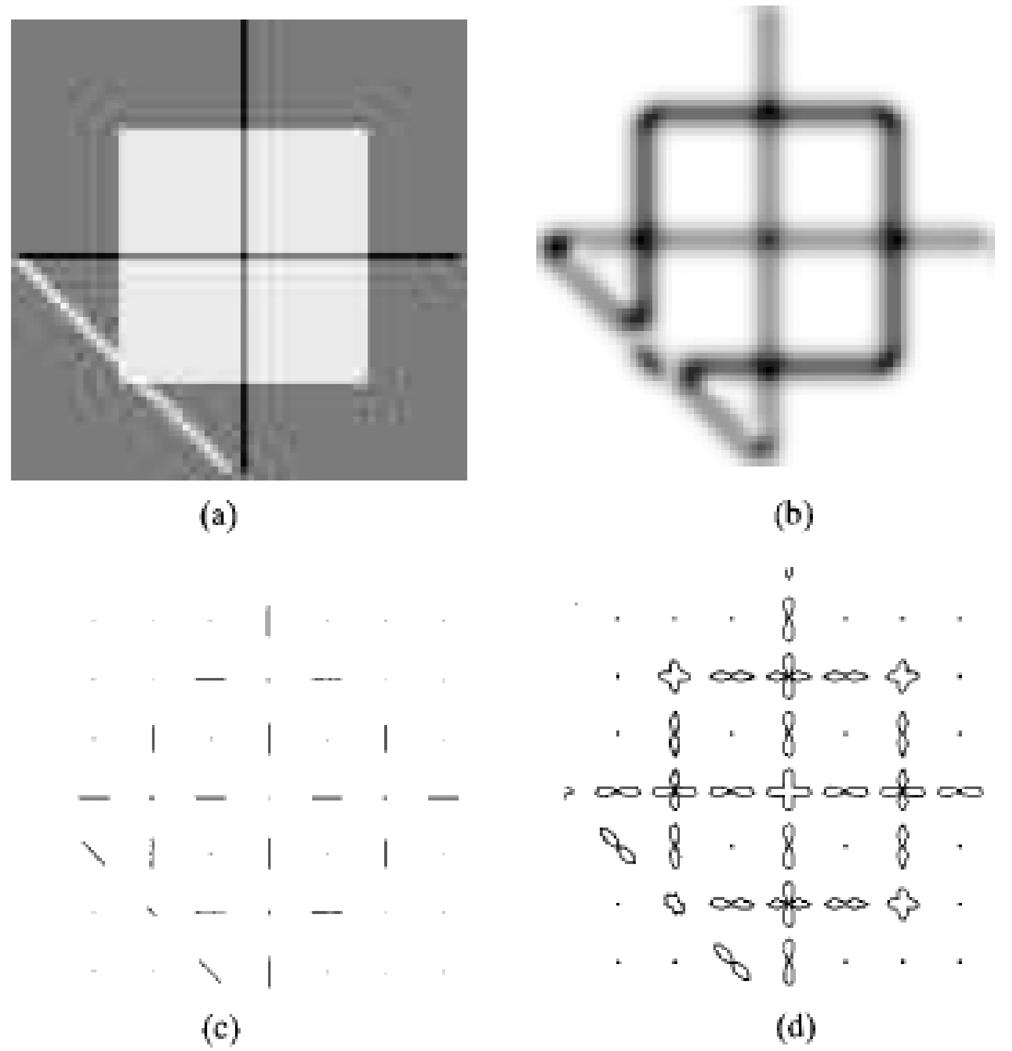


Fig. 10. Measures of orientation derived from G_4 and H_4 steerable filter outputs: (a) Input image for orientation analysis; (b) angular average of oriented energy as measured by G_4, H_4 quadrature pair. This is an oriented features detector; (c) conventional measure of orientation: dominant orientation plotted at each point. No dominant orientation is found at the line intersection or corners; (d) oriented energy as a function of angle, shown as a polar plot for a sampling of points in the image (a). Note the multiple orientations found at intersection points of lines or edges and at corners, shown by the florets there.

Freeman

Untitled5

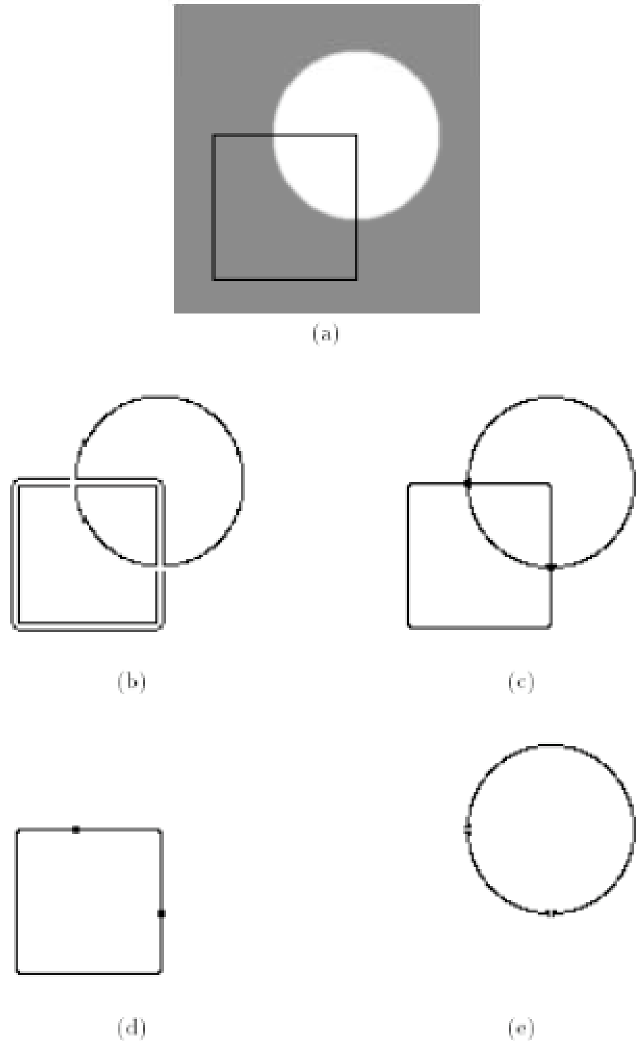


Figure 4-1: (a) Circle and square test image. (b) Output of Canny edge detector. The edges of the circle are accurately tracked, but the lines of the square are marked as two edges, neither at the correct position. (c) Output of steerable filter contour detector. Both edges and lines are marked as single contours, centered on the image feature. (d) Dark lines found by combining the contour detector with a phase estimator. (e) Edges found by combining the contour detector with a phase estimator.

Freeman

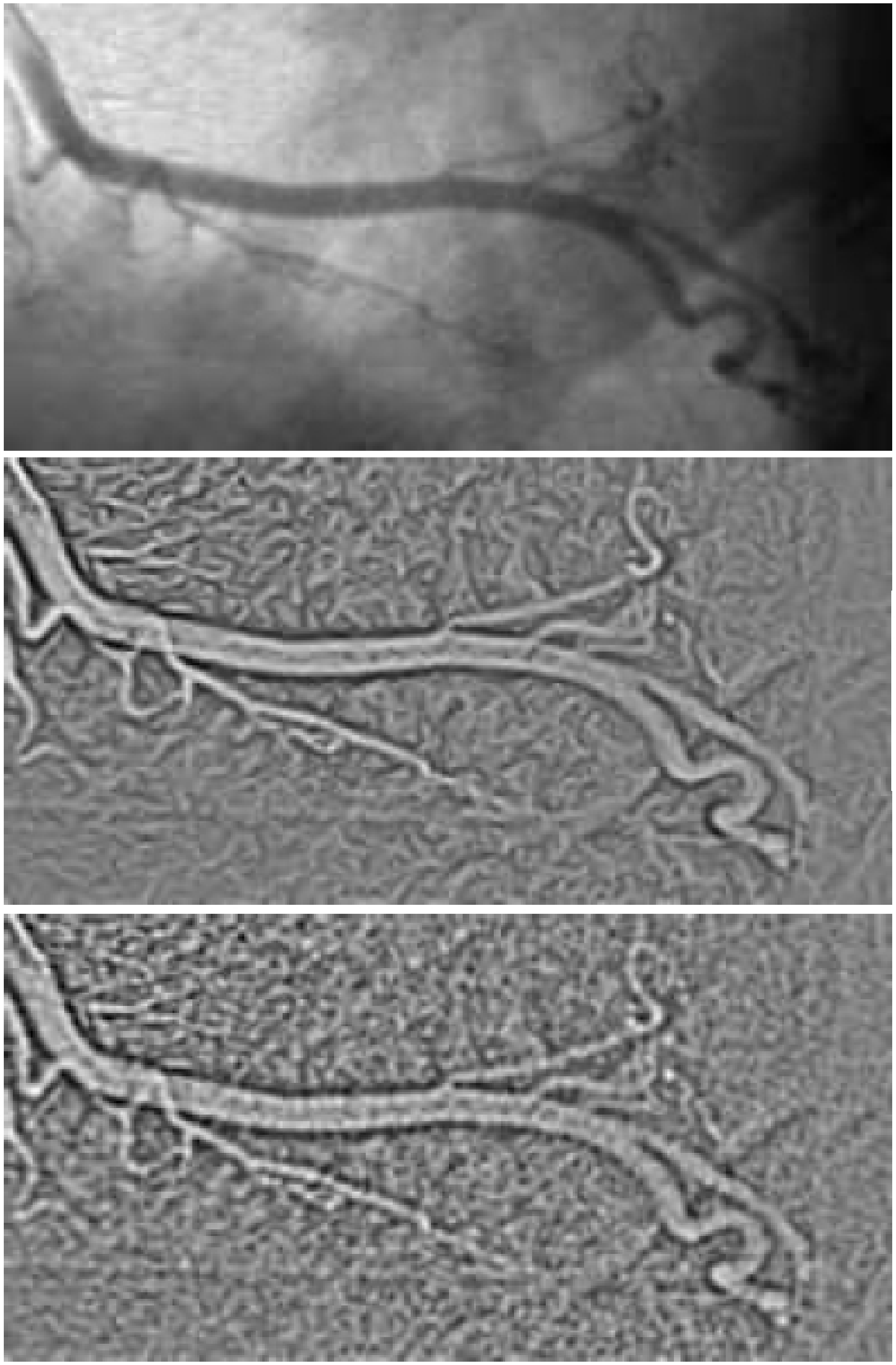


Fig. 12. (a) Digital cardiac angiogram; (b) result of filtering (a) with G_2 oriented along the local direction of dominant orientation, shown after local contrast enhancement (division by the image's blurred absolute value). The oriented vascular structures of (a) are enhanced; (c) isotropic bandpass filtering of (a) after local contrast enhancement. Note the increased noise relative to the oriented filtering results.