# 6.891

## Computer Vision and Applications

Prof. Trevor. Darrell

Lecture 3:
- Multi-scale Image Representations
- Gaussian/Laplacian Pyramids
- QMF/Wavelets
- Steerable Filters
- Image statistics

Readings: F&P Chapter 7.7, 9.2; Simoncelli et al. handout

1

## Course Calendar

| Lecture | Date | Description | Readings | Assignments | Materials |
|---|---|---|---|---|---|
| 1 | 2/3 | Course Introduction Cameras and Lenses | Req: FP 1.1, 2.1, 2.2, 2.3, 3.1, 3.2 | PS0 out | Lecture 1 Lecture 1 (6 slides/page) |
| 2 | 2/5 | Image Filtering | Req: FP 7.1 - 7.6 | | Lecture 2 Lecture 2 (6 slides/page) |
| 3 | 2/10 | Image Representations: pyramids | Req: FP 7.7, 9.2 | | Handout 1 |
| 4 | 2/12 | Texture | Req: FP 9.1, 9.3, 9.4 | PS0 due | |
| | 2/17 | Monday Classes Held (NO LECTURE) | | | |
| 5 | 2/19 | Color | Req: FP 6.1-6.4 | PS1 out | |
| 6 | 2/24 | Local Features | | | |
| 7 | 2/26 | Multiview Geometry | Req: FP 10 | PS1 due | |
| 8 | 3/2 | Affine Reconstruction | Req: FP 12 | | |
| 9 | 3/4 | Projective Reconstruction | Req: FP 13 | PS2 out | |
| 10 | 3/9 | Scene Reconstruction | | | |
| 11 | 3/11 | Non-Rigid Motion | | PS2 due | |
| 12 | 3/16 | Morphable and Active Appearance Models | | EX1 out | |
| 13 | 3/18 | Model-Based Object Recognition | | EX1 due | |
| | 3/23-3/25 | Spring Break (NO LECTURE) | | | |

2

## Last time: Linear Filters

- Convolution kernels
- Edges and contrast
- Fourier transform
- Sampling and Aliasing

3

## Linear image transformations

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

transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

4

## An example of such a transform: the Fourier transform

discrete domain

Forward transform

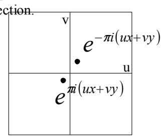
$$F[m,n] = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} f[k,l]e^{-\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

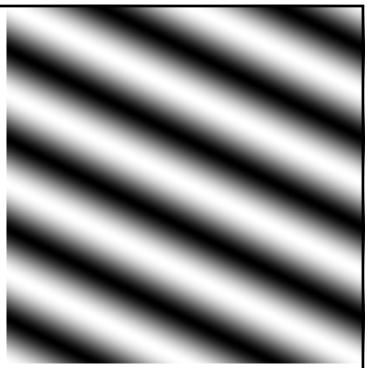
Inverse transform

$$f[k,l] = \frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} F[m,n]e^{+\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

5

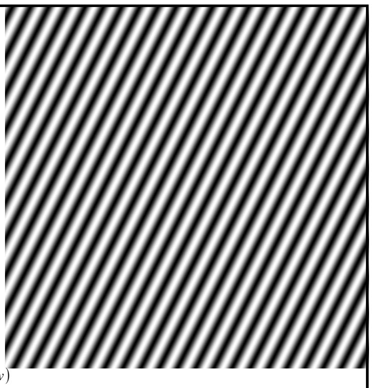
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

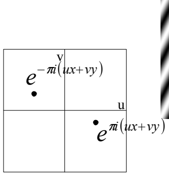


6

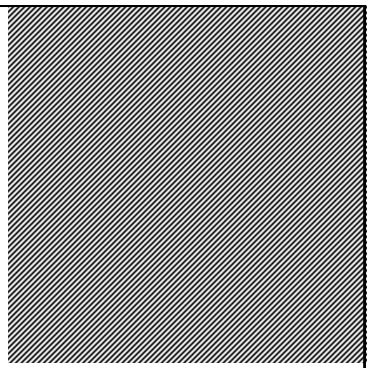
Here u and v are larger than in the previous slide.

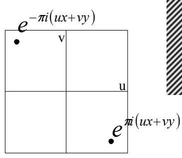





7

And larger still...

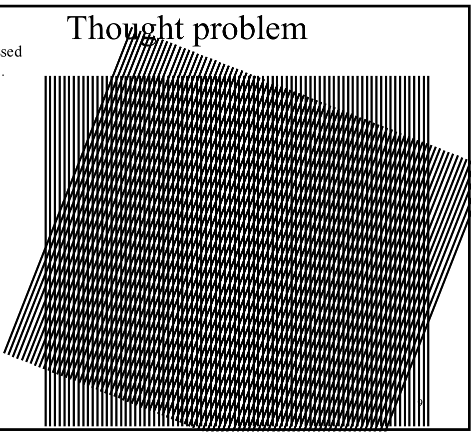





8

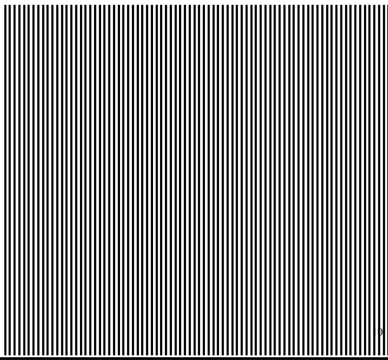
# Thought problem

Analyze crossed gratings…



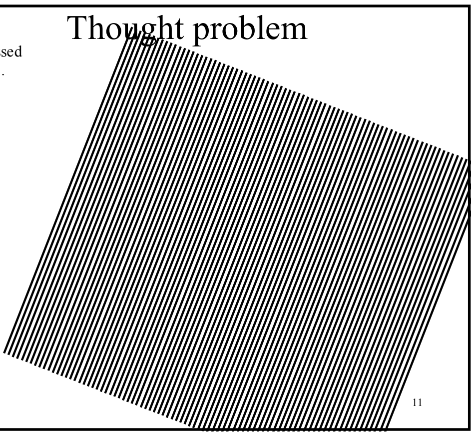
# Thought problem

Analyze crossed gratings…



# Thought problem

Analyze crossed gratings…

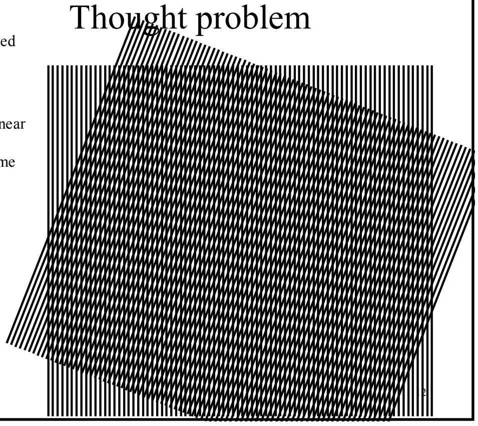


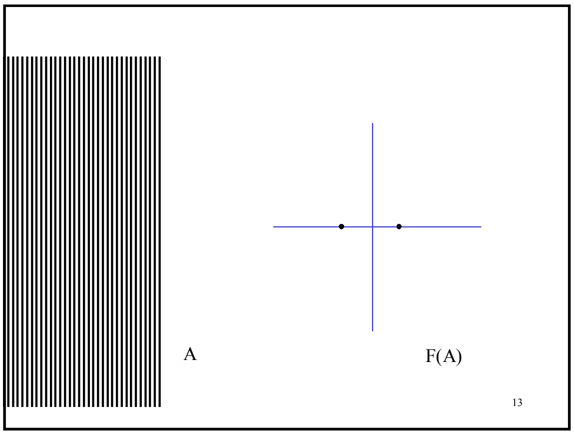
11

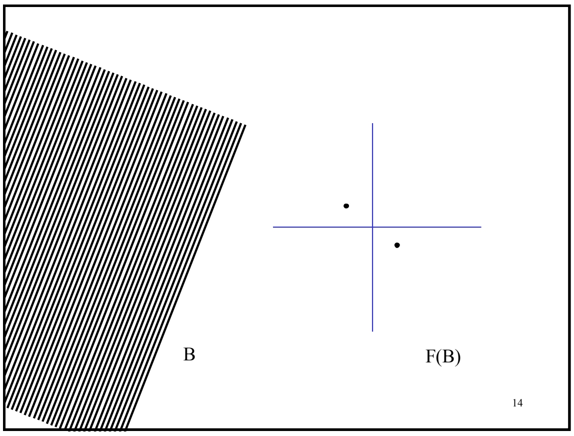
# Thought problem

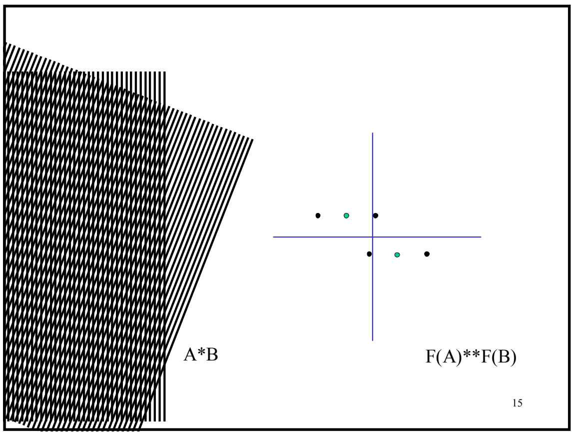
Analyze crossed gratings…

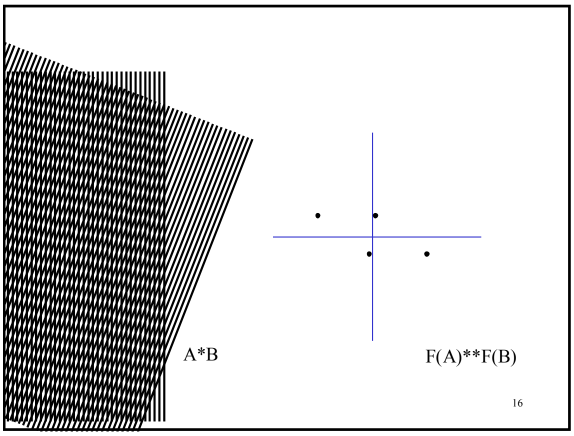
Where does perceived near horizontal grating come from?

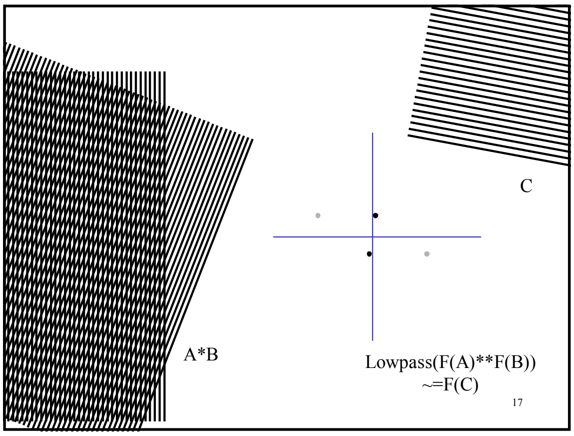





## Today

- Image pyramids
- Image statistics
- Color and spatial frequency effects

18

## What is a good representation for image analysis?

- Fourier transform domain tells you “what” (textural properties), but not “where”.
- Pixel domain representation tells you “where” (pixel location), but not “what”.
- Want an image representation that gives you a local description of image events—what is happening where.
- Should naturally represent objects across varying scale.

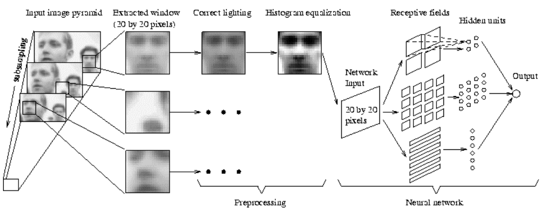
19

## Scaled representations

- Big bars (resp. spots, hands, etc.) and little bars are both interesting
  - Stripes and hairs, say
- Inefficient to detect big bars with big filters
  - And there is superfluous detail in the filter kernel
- Alternative:
  - Apply filters of fixed size to images of different sizes
  - Typically, a collection of images whose edge length changes by a factor of 2 (or root 2)
  - This is a pyramid (or Gaussian pyramid) by visual analogy

20

Example application: CMU face detector




From: http://www.ius.cs.cmu.edu/IUS/har2/har/www/CMU-CS-95-158R/

21

## Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

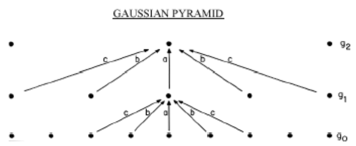
22

## The Gaussian pyramid

- Smooth with gaussians, because
  - a gaussian*gaussian=another gaussian
- Synthesis
  - smooth and sample
- Analysis
  - take the top image
- Gaussians are low pass filters, so repn is redundant

23

The computational advantage of pyramids




$g_0$ = 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.

24

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

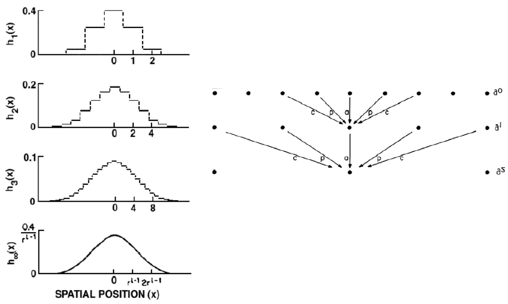



Fig. 2. The equivalent weighting functions $h_l(x)$ for nodes in levels 1, 2, 3, and infinity of the Gaussian pyramid. Note that axis scales have been adjusted by factors of 2 to aid comparison Here the parameter *a* of the generating kernel is 0.4, and the resulting equivalent weighting functions closely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

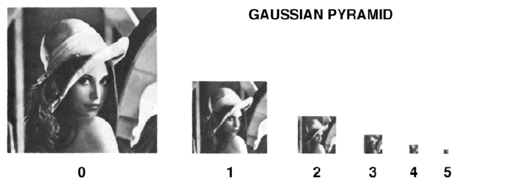
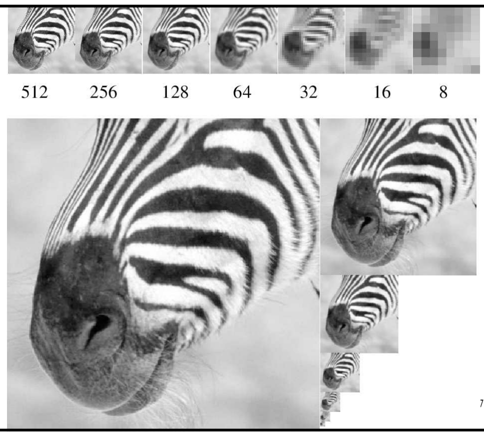



Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image The original image, level 0, meusures 257 by 257 pixels and each higher level array is roughly half the dimensdons of its predecessor. Thus, level 5 measures just 9 by 9 pixels.



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

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






# Linear image transforms

transformed image

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

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform



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

U1 =

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |



# Next pyramid level

U2 =

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

# b * a, the combined effect of the two pyramid levels

```
>> U2 * U1

ans =

   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
```



# The Laplacian Pyramid

- Synthesis
  - preserve difference between upsampled Gaussian pyramid level and Gaussian pyramid level
  - band pass filter - each level represents spatial frequencies (largely) unrepresented at other levels
- Analysis
  - reconstruct Gaussian pyramid, take top layer



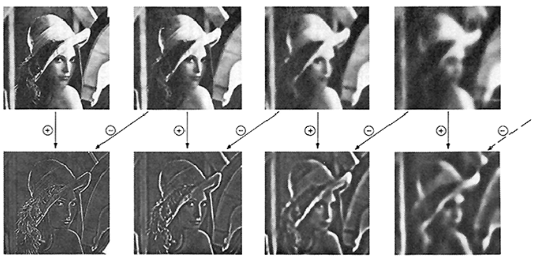


Fig 5. First four levels of the Gaussian and Laplacian pyramid. Gaussian images, upper row, were obtainedby expanding pyramid arrays (Fig. 4) through Gaussian interpolation. Each level of the Laplacian pyramid is the difference between the corresponding and next higher levels of the Gaussian pyramid.

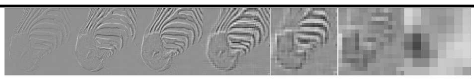


http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

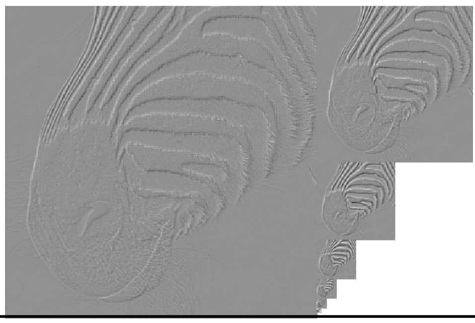
IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

512 256 128 64 32 16 8





512 256 128 64 32 16 8





# Application to image compression

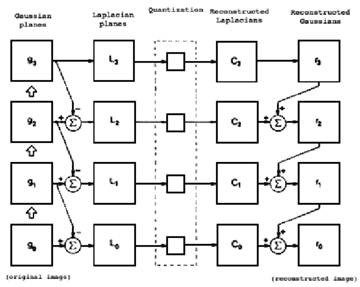



Fig. 10. A summary of the steps in Laplacian pyramid coding and decoding. First, the original image $g_0$ (lower left) is used to generate Gaussian pyramid levels $g_1$, $g_2$, … through repeated local averaging. Levels of the Laplacian pyramid $L_0$, $L_1$, … are then computed as the differences between adjacent Gaussian levels. Laplacian pyramid elements are quantized to yield the Laplacian pyramid code $C_0$, $C_1$, $C_2$, …. Finally, a reconstructed image $r_0$ is generated by summing levels of the code pyramid.

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

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

# Oriented pyramids

Laplacian pyramid is
multi-scale
band-pass

but is *over-complete*

Is this a problem?
*maybe*

Wavelets/QMFs are multi-scale, band-pass, complete…



# Wavelets/QMF's

High and low bandpass analysis filters…

U =

| | |
|---|---|
| 1 | 1 |
| 1 | -1 |

*(what about for synthesis?)*

\>> inv(U)

ans =

| | |
|---|---|
| 0.5000 | 0.5000 |
| 0.5000 | -0.5000 |



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 |



\>> inv(U)

ans =

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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.

| 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 $(-1)^n$.



Simoncelli and Adelson, in "Subband coding", Kluwer, 1990.

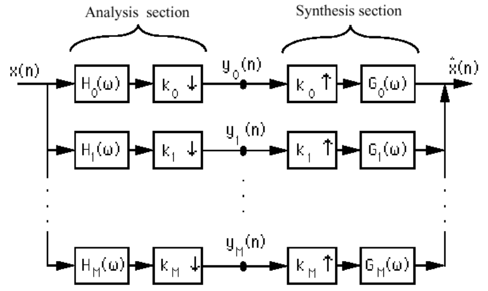



**Figure 4.2:** An analysis/synthesis filter bank.

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

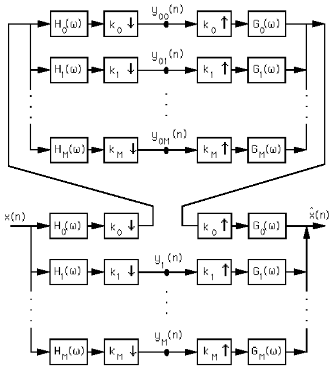


**Figure 4.3:** A non-uniformly cascaded analysis/synthesis filter bank.

43

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

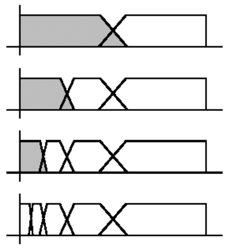


**Figure 4.4:** Octave band splitting produced by a four-level pyramid cascade of a two-band A/S system. The top picture represents the splitting of the two-band A/S system. Each successive picture shows the effect of re-applying the system to the lowpass subband (indicated in grey) of the previous picture.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

44

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

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

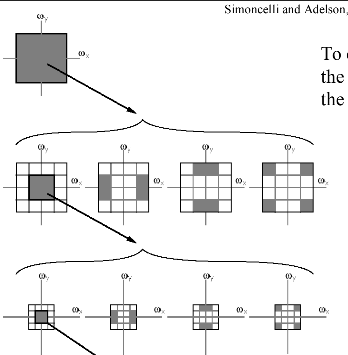
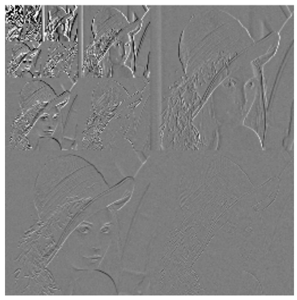


**Figure 4.12:** Idealized diagram of the partition of the frequency plane resulting from a 4-level pyramid cascade of separable 2-band filters. The top plot represents the frequency spectrum of the original image, with axes ranging from $-\pi$ to $\pi$. This is divided into four subbands at the next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

# Wavelet/QMF representation



46

# Good and bad features of wavelet/QMF filters

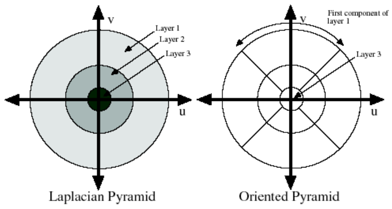
- Bad:
  - Aliased subbands
  - Non-oriented diagonal subband
- Good:
  - Not overcomplete (so same number of coefficients as image pixels).
  - Good for image compression (JPEG 2000)

47

# Steerable pyramids

- Good:
  - Oriented subbands
  - Non-aliased subbands
  - Steerable filters
- 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.

48




49

---

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

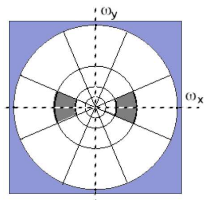
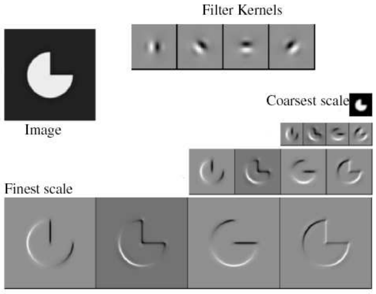



**Figure 1.** Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to $\pi$. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband).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

50

---




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

51

---

| | 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

52

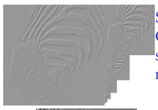
---

# Image pyramids

- Gaussian

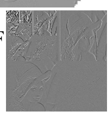
Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

- Laplacian



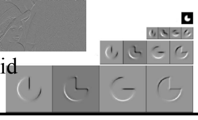
Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- 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.

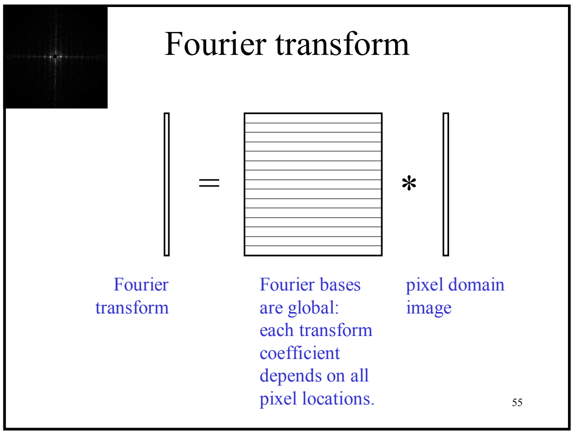
53

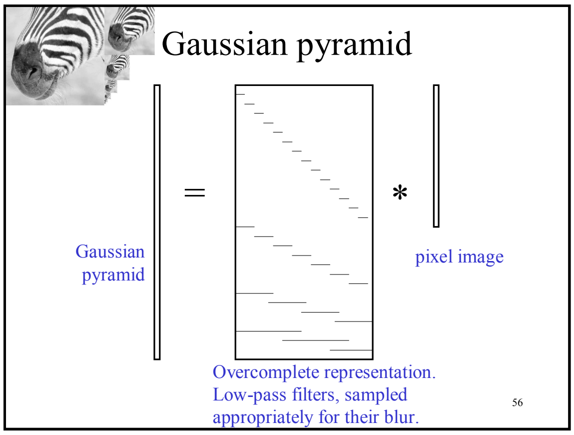
---

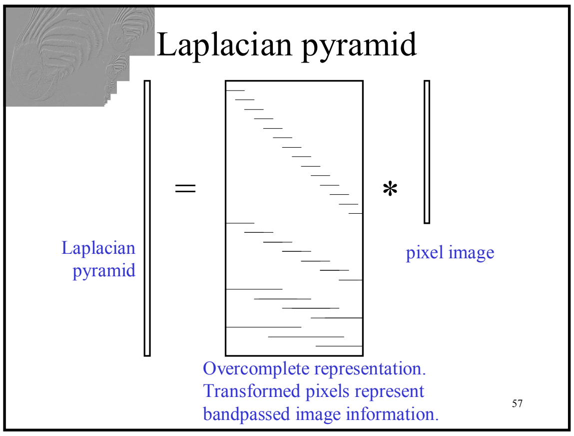
# Schematic pictures of each matrix transform

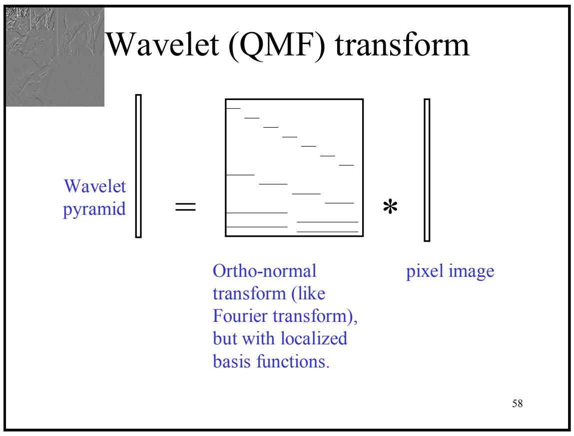
- Shown for 1-d images
- The matrices for 2-d images are the same idea, but more complicated, to account for vertical, as well as horizontal, neighbor relationships.

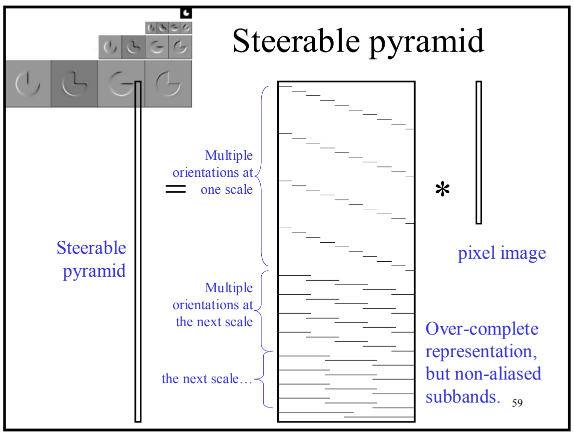
54

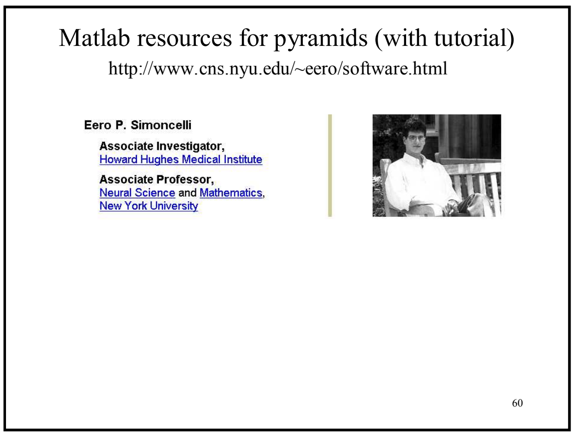

Fourier transform
=
*
Fourier transform
Fourier bases are global: each transform coefficient depends on all pixel locations.
pixel domain image
55


Gaussian pyramid
=
*
Gaussian pyramid
pixel image
Overcomplete representation. Low-pass filters, sampled appropriately for their blur.
56


Laplacian pyramid
=
*
Laplacian pyramid
pixel image
Overcomplete representation. Transformed pixels represent bandpassed image information.
57


Wavelet (QMF) transform
Wavelet pyramid
=
*
Ortho-normal transform (like Fourier transform), but with localized basis functions.
pixel image
58


Steerable pyramid
Multiple orientations at one scale
=
*
Steerable pyramid
pixel image
Multiple orientations at the next scale
the next scale…
Over-complete representation, but non-aliased subbands.
59


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
60

## Matlab resources for pyramids (with tutorial)

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

Laboratory for Computational Vision

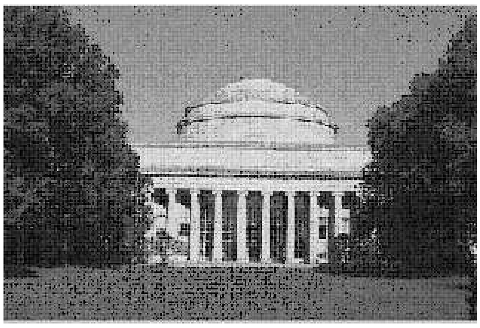
Home | People | Research | Publications | Software

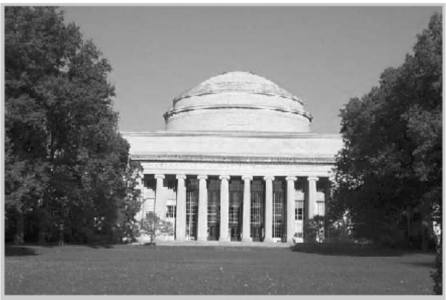
### Publicly Available Software Packages

- 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).

61

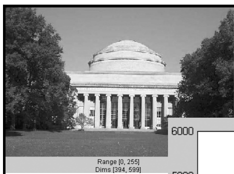
## Image statistics (or, mathematically, how can you tell image from noise?)

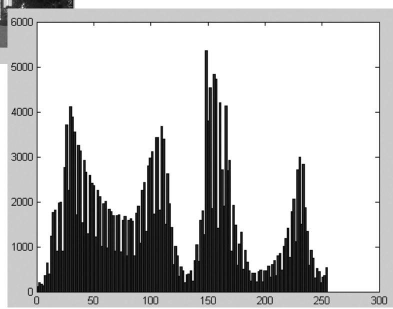




Range [0, 255]
Dims [394, 599]

## Pixel representation image histogram

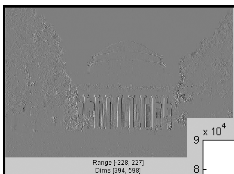


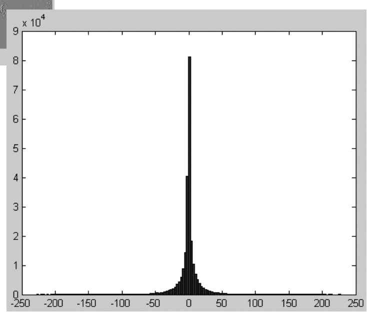




## bandpass filtered image

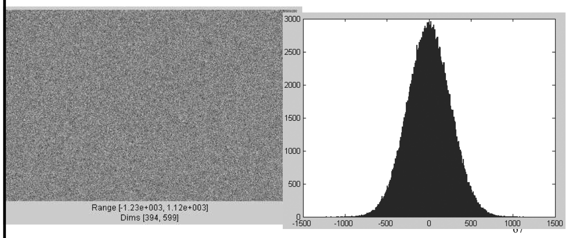
Range [-228, 227]
Dims [394, 598]

## bandpassed representation image histogram

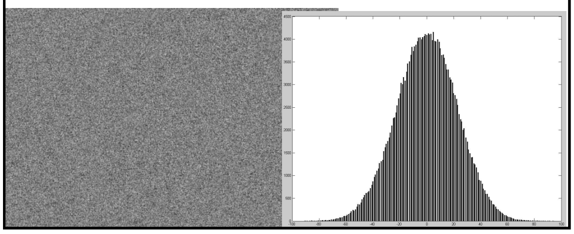


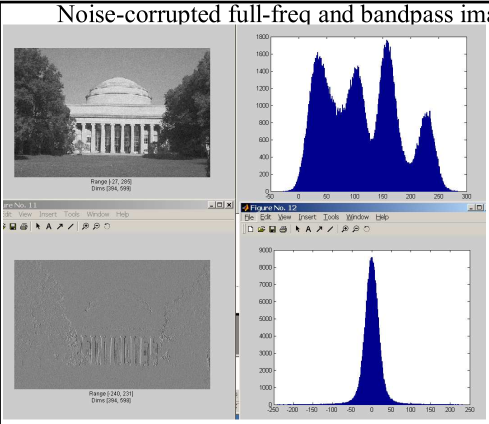
## Pixel domain noise image and histogram




## Bandpass domain noise image and histogram



## Noise-corrupted full-freq and bandpass images




69

## Bayes theorem

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

so

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.

70

## Bayesian MAP estimator for clean bandpass coefficient values

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

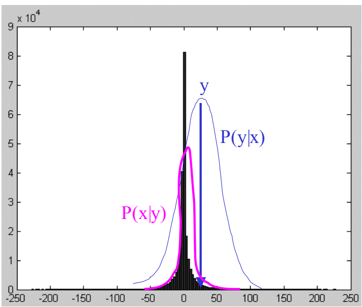
By Bayes theorem

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

P(x)

P(y|x)

P(x|y)




## Bayesian MAP estimator

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

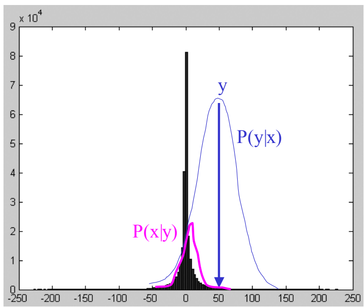
By Bayes theorem

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

P(x)

P(y|x)

P(x|y)



## Bayesian MAP estimator

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

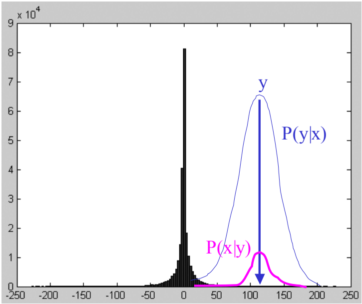
By Bayes theorem

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

P(x)

P(y|x)

P(x|y)




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

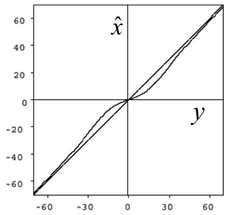



**Figure 2:** Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/simoncelli_noise.pdf **Simoncelli and Adelson, Noise Removal via Bayesian Wavelet Coring**

74

## Noise removal results

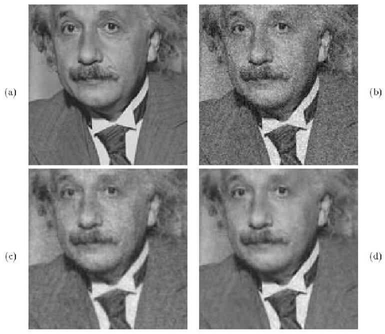



**Figure 4:** Noise reduction example. (a) Original image (cropped). (b) Image contaminated with additive Gaussian white noise (SNR = 9.00dB). (c) Image restored using (semi-blind) Wiener filter (SNR = 11.88dB). (d) Image restored using (semi-blind) Bayesian estimator (SNR = 13.82dB).

http://www-bcs.mit.edu/people/adelson/pub_pdfs/simoncelli_noise.pdf **Simoncelli and Adelson, Noise Removal via Bayesian Wavelet Coring**

75