4 – Image Pyramids

Admin stuff

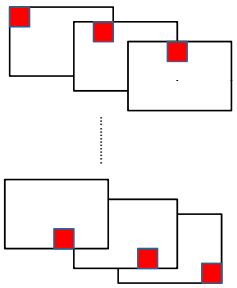
- Change of office hours on Wed 4th April
  - Mon 31st March 9.30-10.30pm (right after class)
- Change of time/date of last class
  - Currently Mon 5th May
  - What about Thursday 8th May?

Projects

- Time to pick!
- Every group must come and see my in the next couple of weeks during office hours!

Spatial Domain

Basis functions:

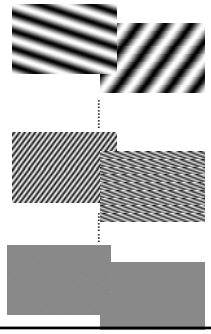


Tells you *where* things are….

… but no concept of *what* it is

Fourier domain

Basis functions:

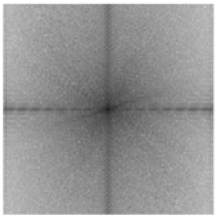


Tells you *what* is in the image….

… but not *where* it is

Fourier as a change of basis

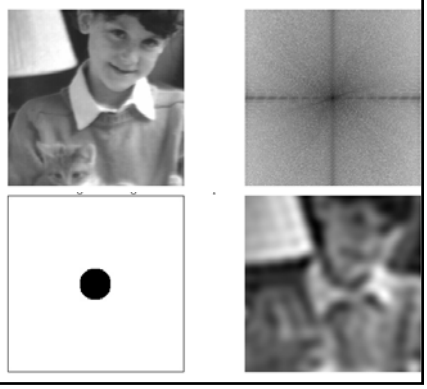
- Discrete Fourier Transform: just a big matrix



http://www.reindeergraphics.com

Low pass filtering

http://www.reindeergraphics.com



High pass filtering

http://www.reindeergraphics.com

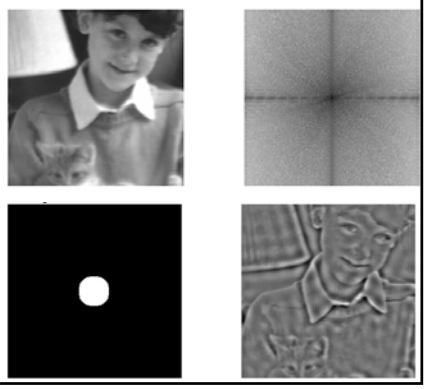


Image Analysis

- Want representation that combines *what* and *where*.

→ Image Pyramids

Why Pyramid?

….equivalent to….

Keep filters same size

- Change image size
- Scale factor of 2

Total number of pixels in pyramid?

1 + ¼ + 1/16 + 1/32…….. = 4/3

→ Over-complete representation

Practical uses

- Compression
  - Capture important structures with fewer bytes
- Denoising
  - Model statistics of pyramid sub-bands
- Image blending

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

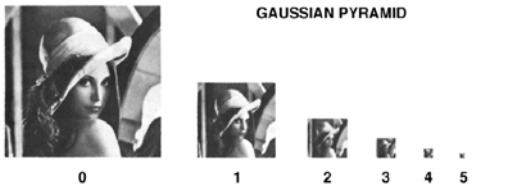
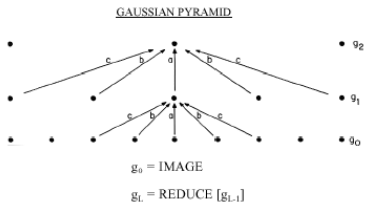


Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image The original image, level 0, measures 257 by 257 pixels and each higher level array is roughly half the dimensdons of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

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

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

The computational advantage of pyramids



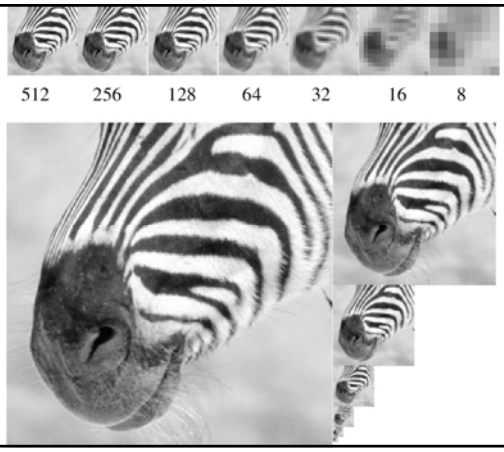
g_0 = IMAGE

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.

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

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



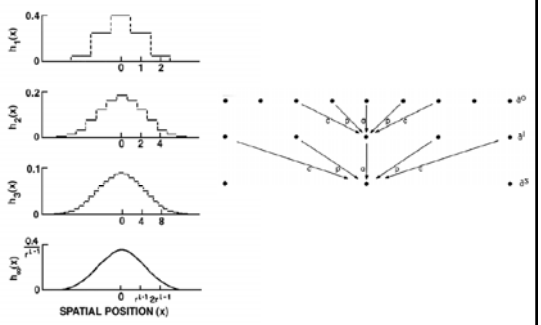
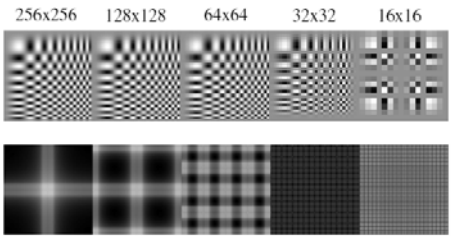


Fig. 2. The equivalent weighting functions $h_l(x)$ for nodes in levels 1, 2, 3, and infinity of the Gaussian pyramid. Note that axis scales have been adjusted by factors of 2 to aid comparison Here the parameter a of the generating kernel is 0.4, and the resulting equivalent weighting functions closely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

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

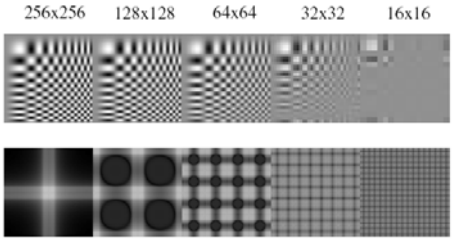
IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

Sampling without smoothing. Top row shows the images, sampled at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.



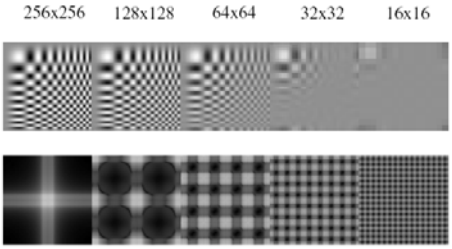
Slide credit: W.T. Freeman

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; bottom row shows the magnitude spectrum of these images.



Slide credit: W.T. Freeman

Sampling with more 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; bottom row shows the magnitude spectrum of these images.



Slide credit: W.T. Freeman

1D Convolution as a matrix operation

$x \oplus f = C_f x$

where f = (f_1 … f_N)
and

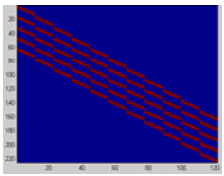
C = (f_N f_(N-1) f_(N-2) … f_1 0 ….. 0
0 f_N f_(N-1) … f_2 f_1 0 …0
……
0 0 0 …. 0 f_N f_(N-1) …. f_2 f_1)

Size of C is |x|-|f|+1 by |x|

2D Convolution as a matrix operation

$X \oplus g = C_g X(:)$

where g = (g_11 … g_1N
g_21 … g_2N
……
g_M1 …. g_MN)



Size of X is I x J
Size C_g is IJ – MN +1 by IJ
(for 'valid' convolution)

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

For 16 pixel 1-D image

U1 =

16 pixels

8 pixels

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

Im_1
Im_2
Im_3
….

….
Im_16

Next pyramid level

U2 =

8 pixels

4 pixels

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   1   4  10  20  31  40  44  40  31  20  10   4
   0   0   0   0   0   0   0   0   1   4  10  20  31  40  44  40
   0   0   0   0   0   0   0   0   0   0   0   0   1   4  10  20
```

Im_1
Im_2
Im_3
….

….
Im_16

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

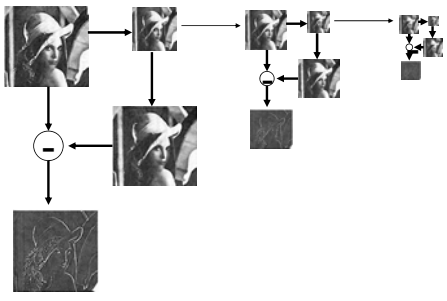
Image pyramids

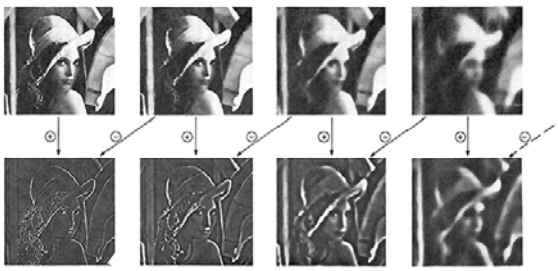
- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

The Laplacian Pyramid

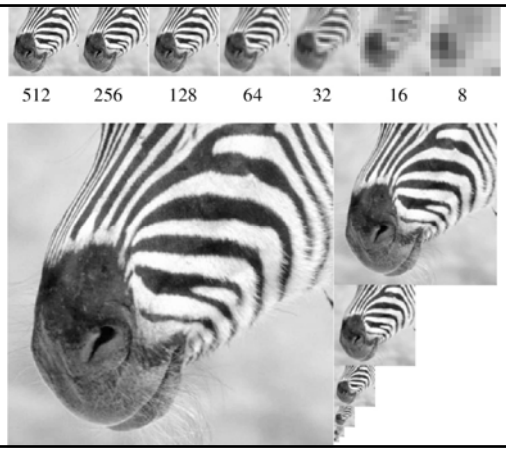
- Synthesis
  - preserve difference between upsampled Gaussian pyramid level and Gaussian pyramid level
  - band pass filter - each level represents spatial frequencies (largely) unrepresented at other levels
- Analysis
  - reconstruct Gaussian pyramid, take top layer

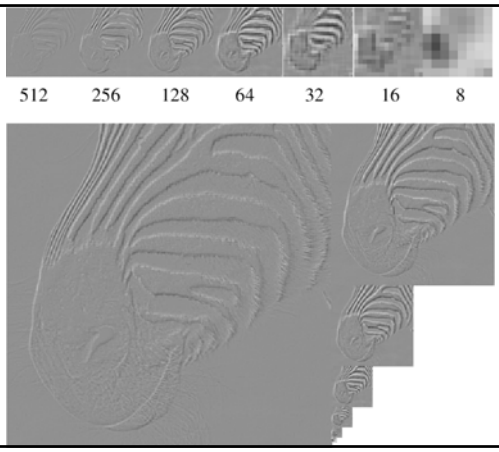
Laplacian pyramid algorithm





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





Why use these representations?

- Handle real-world size variations with a constant-size vision algorithm.
- Remove noise
- Analyze texture
- Recognize objects
- Label image features

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

Efficient search

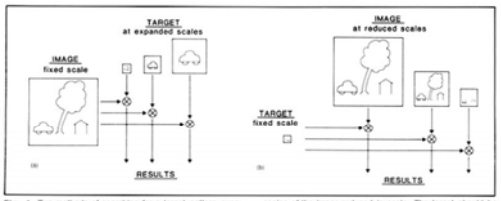
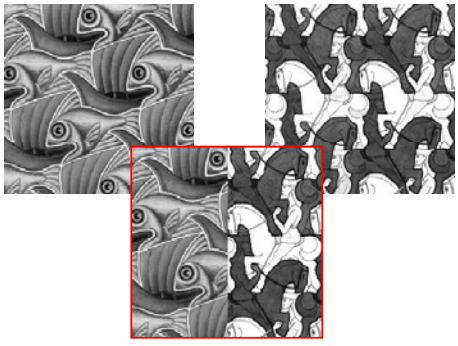


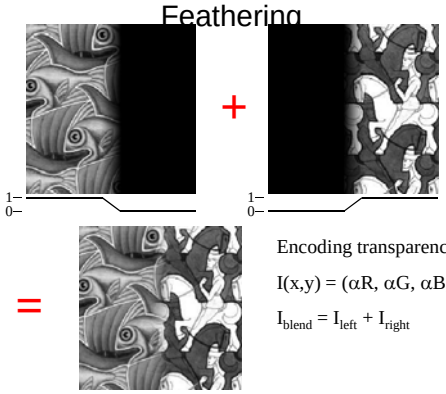
Fig. 1. Two methods of searching for a target pattern over many scales. In the first approach, (a), copies of the target pattern are constructed at several expanded scales, and each is convolved with the original image. In the second approach, (b), a single copy of the target is convolved with copies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

Image Blending



Feathering

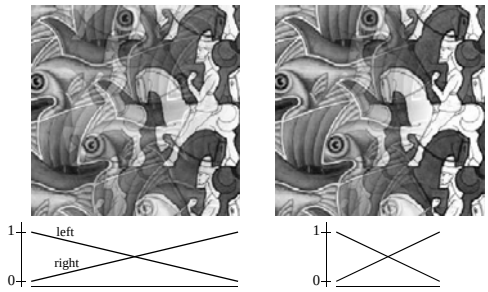


Encoding transparency

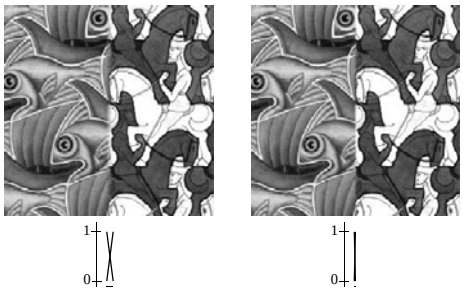
$I(x,y) = (\alpha R, \alpha G, \alpha B, \alpha)$

$I_{blend} = I_{left} + I_{right}$

Affect of Window Size



Affect of Window Size



Good Window Size

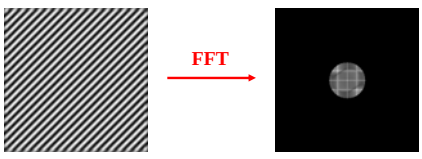
"Optimal" Window: smooth but not ghosted

What is the Optimal Window?

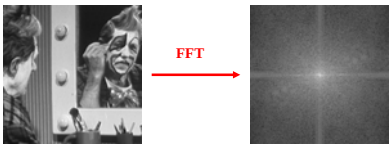
- To avoid seams
  - window >= size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature

Natural to cast this in the *Fourier domain*

- largest frequency <= 2*size of smallest frequency
- image frequency content should occupy one "octave" (power of two)



What if the Frequency Spread is Wide



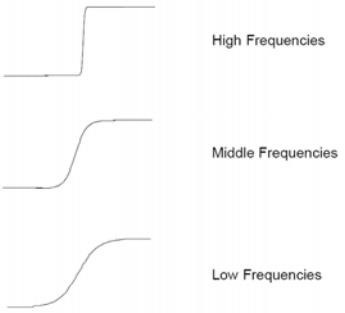
Idea (Burt and Adelson)

- Compute $F_{left} = FFT(I_{left})$, $F_{right} = FFT(I_{right})$
- Decompose Fourier image into octaves (bands)
  - $F_{left} = F_{left}^1 + F_{left}^2 + \ldots$
- Feather corresponding octaves F_{left}^i with F_{right}^i
  - Can compute inverse FFT and feather in spatial domain
- Sum feathered octave images in frequency domain

Better implemented in *spatial domain*

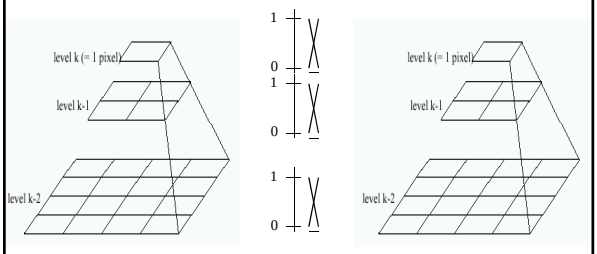
Multiresolution Spline

When splining two images, transition from one image to the other should behave:



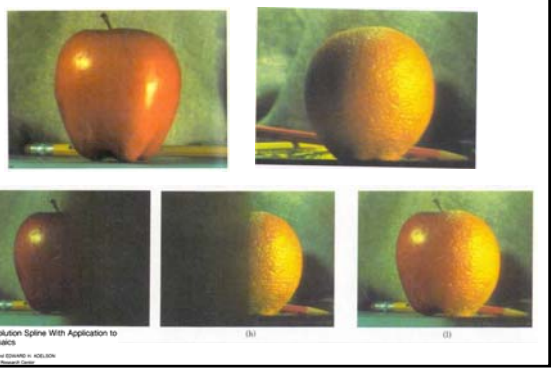
http://cs.haifa.ac.il/~dkeren/ip/lecture8.pdf

Pyramid Blending



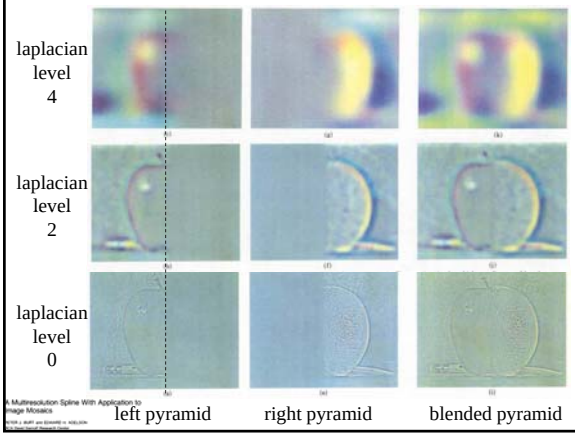
Left pyramid blend Right pyramid

Pyramid Blending



A Multiresolution Spline With Application to Image Mosaics

Pyramid Blending



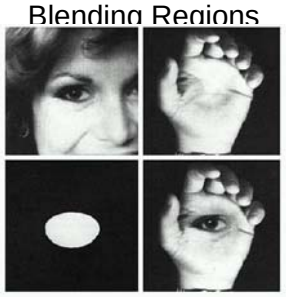
left pyramid right pyramid blended pyramid

Laplacian Pyramid: Region Blending

General Approach:

1. Build Laplacian pyramids *LA* and *LB* from images *A* and *B*
2. Build a Gaussian pyramid *GR* from selected region *R*
3. Form a combined pyramid *LS* from *LA* and *LB* using nodes of *GR* as weights:
   - *LS(i,j) = GR(I,j,)*LA(I,j) + (1-GR(I,j))*LB(I,j)*
4. Collapse the *LS* pyramid to get the final blended image

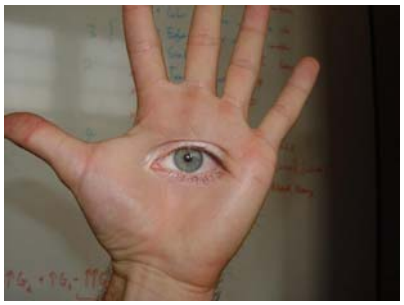
Blending Regions



A Multiresolution Spline With Application to Image Mosaics

PETER J. BURT and EDWARD H. ADELSON
RCA David Sarnoff Research Center

Horror Photo



© david dmartin (Boston College)

Simplification: Two-band Blending

- Brown & Lowe, 2003
  - Only use two bands: high freq. and low freq.

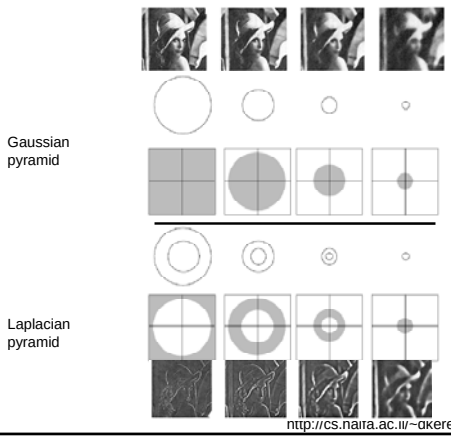
2-band Blending

Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

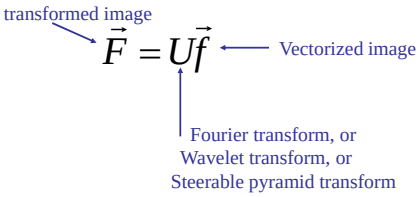


http://cs.haifa.ac.il/~dkeren/ip/lecture8.pdf

Image pyramids

- Gaussian
- Laplacian
- Wavelet/Quadrature Mirror Filters (QMF)
- Steerable pyramid

Wavelets/QMF's



Orthogonal wavelets (e.g. QMF's)

Forward / Analysis $\vec{F} = U\vec{f}$

Inverse / Synthesis $\vec{f} = V^T\vec{F}$

$$UV^T = I$$

The simplest orthogonal wavelet transform: the Haar transform

U =

1 1
1 -1

Haar basis is special case of Quadrature Mirror Filter family

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 =

    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.

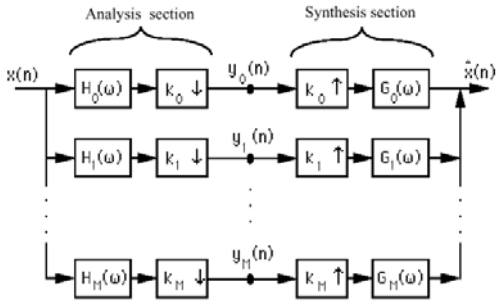


Figure 4.2: An analysis/synthesis filter bank.

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

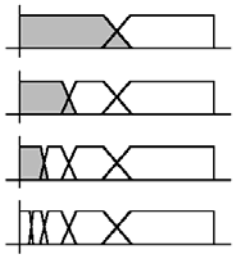
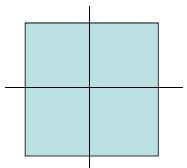
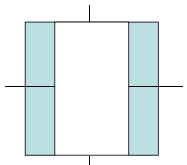


Figure 4.4: Octave band splitting produced by a four-level pyramid cascade of a two-band A/S system. The top picture represents the splitting of the two-band A/S system. Each successive picture shows the effect of re-applying the system to the lowpass subband (indicated in grey) of the previous picture.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

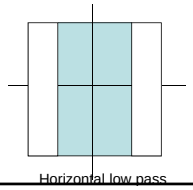
Now, in 2 dimensions…



Frequency domain



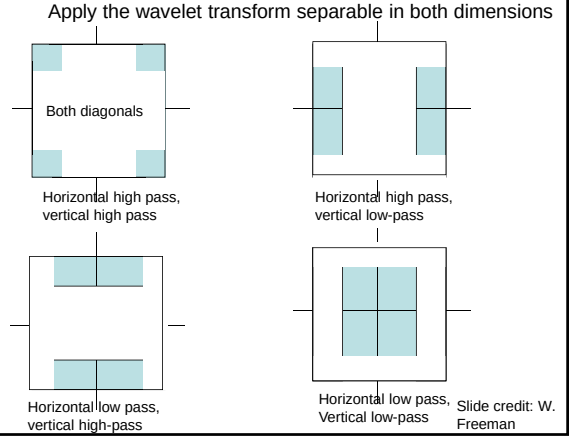
Horizontal high pass



Horizontal low pass

Slide credit: W. Freeman

Apply the wavelet transform separable in both dimensions



Slide credit: W. Freeman

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

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

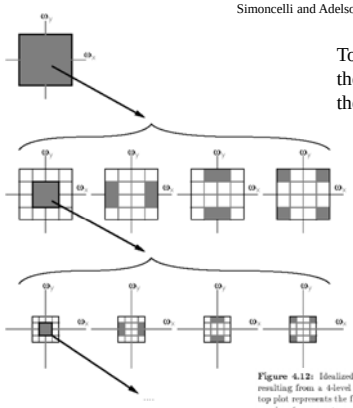
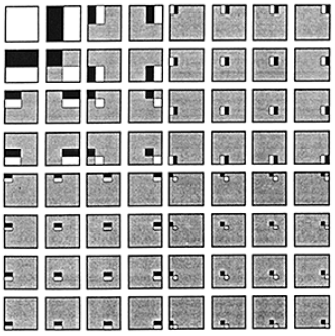


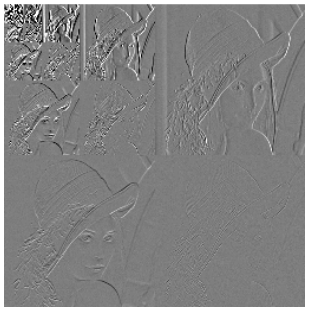
Figure 4.12: Idealized diagram of the partition of the frequency plane resulting from a 4-level pyramid cascade of separable 2-band filters. The top plot represents the frequency spectrum of the original image, with axes ranging from $-\pi$ to π. This is divided into four subbands at the next level.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

Basis



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

Wavelet/QMF representation



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

Some other QMF's

- 9-tap QMF:
- Better localized in frequency

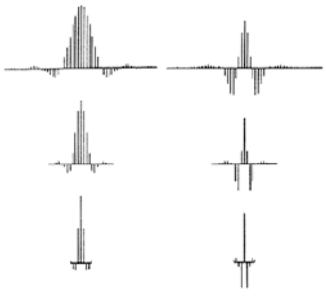


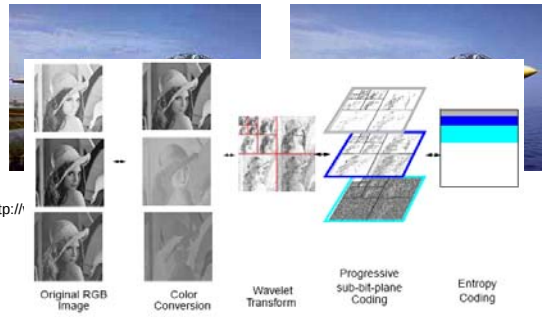
Figure 6: Left: the series of low-pass 1-D kernels derived recursively from a 9-tap QMF kernel. Right: the series of high-pass kernels derived in the same way.

http://web.mit.edu/persci/people/adelson/pub_pdfs/orthogonal87.pdf

Good and bad features of wavelet/QMF filters

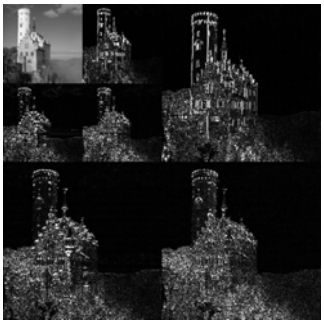
- Bad:
  - Aliased subbands
  - Non-oriented diagonal subband
- Good:
  - Not overcomplete (so same number of coefficients as image pixels).
  - Good for image compression (JPEG 2000)

Compression: JPEG 2000



http://www.rii.ricoh.com/%7Egormish/pdf/dcc2000_jpeg2000_joint_charts.pdf

Compression: JPEG 2000



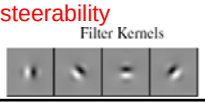
http://en.wikipedia.org/wiki/Image:Jpeg2000_2-level_wavelet_transform-lichtenstein.png

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Steerable filters

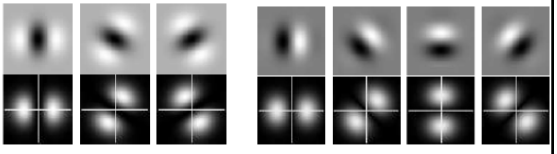
- Analyze image with oriented filters
- Avoid preferred orientation
- Said differently:
  - We want to be able to compute the response to an arbitrary orientation from the response to a few basis filters
  - By linear combination
  - Notion of steerability



Steerable basis filters

- Filters can measure local orientation direction and strength and phase at any orientation.

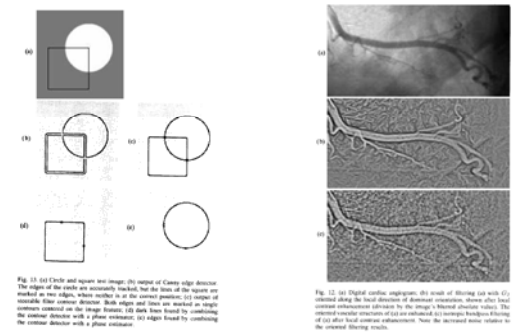
G2 H2

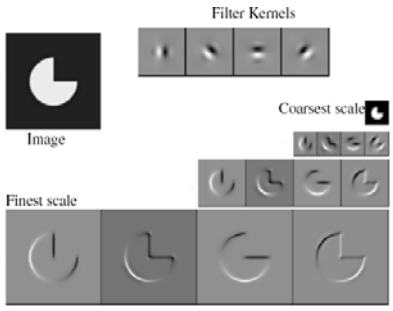


http://people.csail.mit.edu/billf/papers/steerpaper91FreemanAdelson.pdf

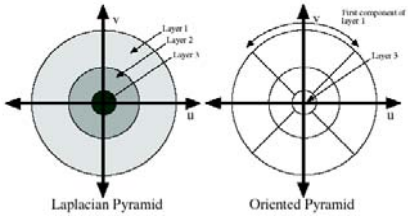
Steerability examples

http://people.csail.mit.edu/billf/papers/steerpaper91FreemanAdelson.pdf



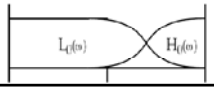


Reprinted from "Shiftable MultiScale Transforms," by Simoncelli et al., IEEE Transactions on Information Theory, 1992, copyright 1992, IEEE



Fourier construction

- Slice Fourier domair
  - Concentric rings for different scales
  - Slices for orientation
  - Feather cutoff to make steerable
  - Tradeoff steerable/orthogonal



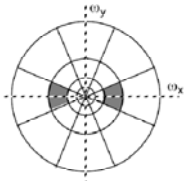


Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For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

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

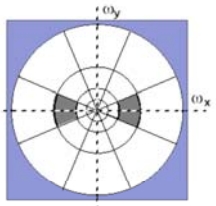


Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For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

Non-oriented steerable pyramid

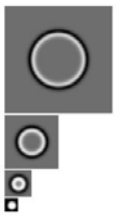


Figure 4: A 3-level $k = 1$ (non-oriented) steerable pyramid. Shown are the bandpass images and the final lowpass image.

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

3-orientation steerable pyramid

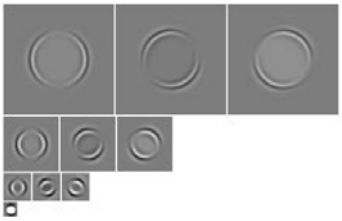


Figure 5: A 3-level $k = 3$ (second derivative) steerable pyramid. Shown are the three bandpass images at each scale and the final lowpass image.

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

Steerable pyramids

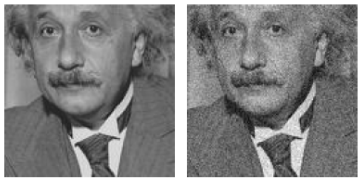
- Good:
  - Oriented subbands
  - Non-aliased subbands
  - Steerable filters
- Bad:
  - Overcomplete
  - Have one high frequency residual subband, required in order to form a circular region of analysis in frequency from a square region of support in frequency.

	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

Application: Denoising



How to characterize the difference between the images?

How do we use the differences to clean up the image?

http://www.cns.nyu.edu/pub/lcv/simoncelli96c.pdf

Application: Denoising

Usually zero, sometimes big Usually close to zero, very rarely big

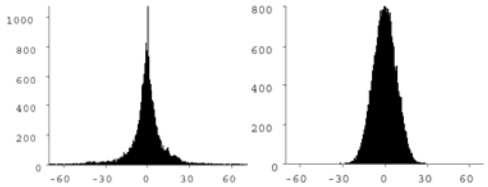
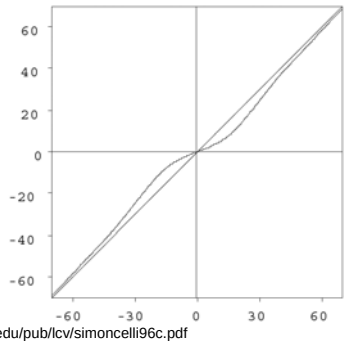


Figure 1. Histograms of a mid-frequency subband in an octave-bandwidth wavelet decomposition for two different images. Left: The"Einstein" image. Right: A white noise image with uniform pdf.

http://www.cns.nyu.edu/pub/lcv/simoncelli96c.pdf

Application: Denoising

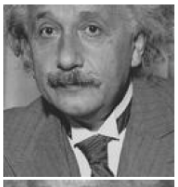
Coring function:

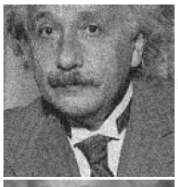


http://www.cns.nyu.edu/pub/lcv/simoncelli96c.pdf

Application: Denoising

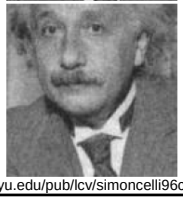
Original

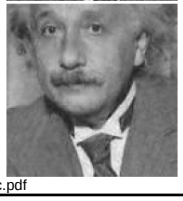




Noise-corrupted

Wiener filter





Steerable pyramid coring

http://www.cns.nyu.edu/pub/lcv/simoncelli96c.pdf

- Summary of pyramid representations

Image pyramids

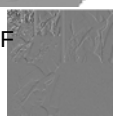
- Gaussian

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.

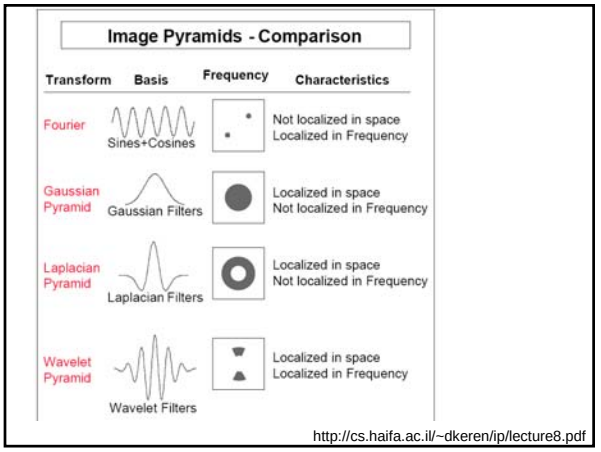
- Wavelet/QMF



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

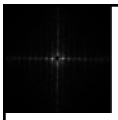
- Steerable pyramid

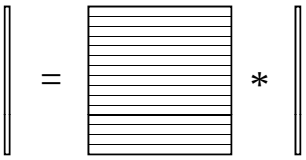
Shows components at each scale and orientation separately. Non-aliased subbands. Good for texture and feature analysis.



http://cs.haifa.ac.il/~dkeren/ip/lecture8.pdf

Fourier transform





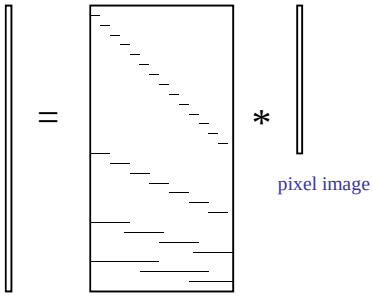
Fourier transform

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

pixel domain image

Slide credit: W. Freeman

Gaussian pyramid



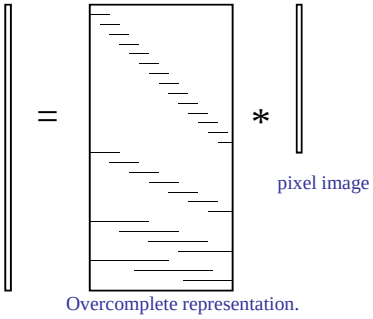
Gaussian pyramid

pixel image

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

Slide credit: W. Freeman

Laplacian pyramid



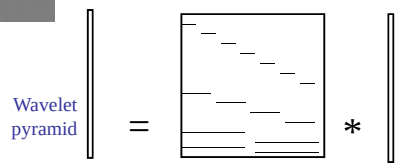
Laplacian pyramid

pixel image

Overcomplete representation. Transformed pixels represent bandpassed image information.

Slide credit: W. Freeman

Wavelet (QMF) transform

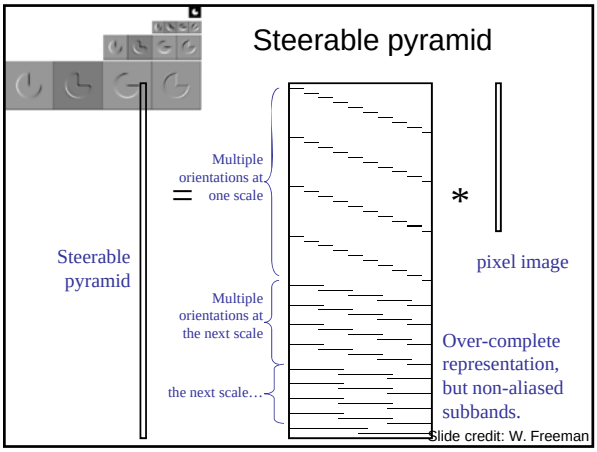


Ortho-normal transform (like Fourier transform), but with localized basis functions.

pixel image

Slide credit: W. Freeman

Steerable pyramid



Over-complete representation, but non-aliased subbands.

Slide credit: W. Freeman

Matlab resources for pyramids (with tutorial)

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

Ted Adelson (MIT) Bill Freeman (MIT)

Matlab resources for pyramids (with tutorial)

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