BSB 663
Image Processing
Apr. 16, 2012

Erkut Erdem

Image Pyramids
Review – Frequency Domain Techniques

• The name “filter” is borrowed from frequency domain processing
• Accept or reject certain frequency components
• Fourier (1807): Periodic functions could be represented as a weighted sum of sines and cosines

Image courtesy of Technology Review
**Review - Fourier Transform**

We want to understand the frequency \( w \) of our signal. So, let’s reparametrize the signal by \( w \) instead of \( x \):

\[
\begin{align*}
f(x) & \quad \rightarrow \quad \text{Fourier Transform} \quad \rightarrow \quad F(w)
\end{align*}
\]

For every \( w \) from 0 to \( \infty \), \( F(w) \) holds the amplitude \( A \) and phase \( f \) of the corresponding sine

\[
A \sin(\omega x + \phi)
\]

* How can \( F \) hold both? Complex number trick!

\[
F(\omega) = R(\omega) + iI(\omega)
\]

\[
A = \pm \sqrt{R(\omega)^2 + I(\omega)^2} \quad \phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}
\]

We can always go back:

\[
\begin{align*}
F(w) & \quad \rightarrow \quad \text{Inverse Fourier Transform} \quad \rightarrow \quad f(x)
\end{align*}
\]

Slide credit: A. Efros
Review - The Discrete Fourier transform

• Forward transform

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

• Inverse transform

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

- Vertical orientation
- Horizontal orientation
- 45 deg.
- fx in cycles/image
- Low spatial frequencies
- High spatial frequencies
- Log power spectrum

Slide credit: B. Freeman and A. Torralba
Review - The Convolution Theorem

• The Fourier transform of the convolution of two functions is the product of their Fourier transforms

\[ F[g * h] = F[g]F[h] \]

• The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms

\[ F^{-1}[gh] = F^{-1}[g] \ast F^{-1}[h] \]

• **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

Slide credit: A. Efros
Review - Filtering in frequency domain

Slide credit: D. Hoiem
Review - Low-pass, Band-pass, High-pass filters

low-pass:

High-pass / band-pass:

Slide credit: A. Efros
Template matching

• Goal: find in image

• Main challenge: What is a good similarity or distance measure between two patches?
  – Correlation
  – Zero-mean correlation
  – Sum Square Difference
  – Normalized Cross Correlation
Matching with filters

- Goal: find \( \square \) in image
- Method 0: filter the image with eye patch

\[
h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]
\]

What went wrong?
response is stronger for higher intensity

Input

Filtered Image
Matching with filters

• Goal: find in image

• Method 1: filter the image with zero-mean eye

\[ h[m, n] = \sum_{k,l} (f[k, l] - \bar{f}) (g[m + k, n + l]) \]

True detections
False detections
Matching with filters

- **Goal:** find in image
- **Method 2:** SSD

\[ h[m, n] = \sum_{k,l} (g[k, l] - f[m + k, n + l])^2 \]
Matching with filters

• Goal: find \[\text{in image}\]

• Method 2: SSD

\[h[m, n] = \sum_{k, l} (g[k, l] - f[m + k, n + l])^2\]

What’s the potential downside of SSD?

SSD sensitive to average intensity
Matching with filters

- Goal: find in image
- Method 3: Normalized cross-correlation

\[
\begin{align*}
h[m,n] &= \frac{\sum_{k,l} (g[k,l] - \bar{g})(f[m-k,n-l] - \bar{f}_{m,n})}{\left(\sum_{k,l} (g[k,l] - \bar{g})^2 \sum_{k,l} (f[m-k,n-l] - \bar{f}_{m,n})^2\right)^{0.5}} \\
\text{Matlab: normxcorr2(template, im)}
\end{align*}
\]
Matching with filters

• Goal: find 🕵️‍♀️ in image

• Method 3: Normalized cross-correlation
Matching with filters

- Goal: find in image
- Method 3: Normalized cross-correlation

Slide: Hoiem
Q: What is the best method to use?

A: Depends

• SSD: faster, sensitive to overall intensity
• Normalized cross-correlation: slower, invariant to local average intensity and contrast
Q: What if we want to find larger or smaller eyes?

A: Image Pyramid
Image information occurs over many different spatial scales. Image pyramids—multi-resolution representations for images—are a useful data structure for analyzing and manipulating images over a range of spatial scales.
Image pyramids

Image information occurs at all spatial scales

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

Image information occurs at all spatial scales

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

Slide credit: B. Freeman and A. Torralba
Review of Sampling

Image → Gaussian Filter → Low-Pass Filtered Image → Sample → Low-Res Image
The Gaussian pyramid

- Smooth with Gaussians, because
  - A Gaussian * Gaussian = another Gaussian
- Gaussians are low pass filters, so representation is redundant.
- Gaussian pyramid creates versions of the input image at multiple resolutions.
- This is useful for analysis across different spatial scales, but doesn’t separate the image into different frequency bands.

Slide adapted from: B. Freeman and A. Torralba
The computational advantage of pyramids

$g_0 = \text{IMAGE}$

$g_L = \text{REDUCE}[g_{L-1}]$

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

[Burt and Adelson, 1983]
The Gaussian 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 dimensions of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

[Burt and Adelson, 1983]
Convolution and subsampling as a matrix multiply (1D case)

\[ x_2 = G_1 x_1 \]

\[ G_1 = \begin{bmatrix}
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 & 1 & 4 & 6 & 4 & 1 & 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 & 1 & 4 & 6 & 4 & 1 & 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 & 1 & 4 & 6 & 4 & 1 & 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 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\
\end{bmatrix} 

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

Slide credit: B. Freeman and A. Torralba
Next pyramid level

\[ x_3 = G_2 x_2 \]

\[
G_2 = \\
\begin{array}{cccccccccc}
1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4
\end{array}
\]
The combined effect of the two pyramid levels

\[ x_3 = G_2 G_1 x_1 \]

\[ G_2 G_1 = \]

\[
\begin{array}{cccccccccccccccccccc}
1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 30 & 16 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 25 & 16 & 4 & 0
\end{array}
\]
Fig. 2. The equivalent weighting functions $h_i(x)$ 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.
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.
1D Gaussian pyramid matrix, for [1 4 6 4 1] low-pass filter

full-band image, highest resolution

lower-resolution image

lowest resolution image

Slide credit: B. Freeman and A. Torralba
Template Matching with Image Pyramids

Input: Image, Template

1. Match template at current scale

2. Downsample image

3. Repeat 1-2 until image is very small

4. Take responses above some threshold, perhaps with non-maxima suppression
Coarse-to-fine Image Registration

1. Compute Gaussian pyramid
2. Align with coarse pyramid
3. Successively align with finer pyramids
   - Search smaller range

Why is this faster?

Are we guaranteed to get the same result?
Image pyramids

Image information occurs at all spatial scales

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

Slide credit: B. Freeman and A. Torralba
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 provides an extra level of analysis as compared to Gaussian pyramid by breaking the image into different isotropic spatial frequency bands.
The Laplacian Pyramid

\[ x_1 \]

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

\[ x_1 \quad G_1 x_1 \]

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

$x_1 \rightarrow G_1 x_1 \rightarrow F_1 G_1 x_1$

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

\[ x_1 \rightarrow G_1 x_1 \rightarrow (I - F_1 G_1) x_1 \]

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

\( x_1 \)

\[ G_1 x_1 = x_2 \]

\[ F_1 G_1 x_1 \]

\[ (I - F_1 G_1) x_1 \]

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

\[ x_1 \xrightarrow{G_1} x_2 \]

\[ F_1G_1x_1 \]

\[ (I - F_2G_2)x_2 \]

Slide credit: B. Freeman and A. Torralba
The Laplacian Pyramid

\[ x_1 \quad G_1 x_1 = x_2 \quad x_2 \quad (I - F_3 G_3) x_3 \]

\[ (I - F_2 G_2) x_2 \quad F_1 G_1 x_1 \quad (I - F_1 G_1) x_1 \]

Slide credit: B. Freeman and A. Torralba
Upsampling

\[ y_2 = F_3 x_3 \]

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

\[
F_3 = \begin{bmatrix}
6 & 1 & 0 & 0 \\
4 & 4 & 0 & 0 \\
1 & 6 & 1 & 0 \\
0 & 4 & 4 & 0 \\
0 & 1 & 6 & 1 \\
0 & 0 & 4 & 4 \\
0 & 0 & 1 & 6 \\
0 & 0 & 0 & 4 \\
\end{bmatrix}
\]

Slide credit: B. Freeman and A. Torralba
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.

Slide credit: B. Freeman and A. Torralba
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:
$L_1 = (I - F_1 G_1) x_1$
$L_2 = (I - F_2 G_2) x_2$
$L_3 = (I - F_3 G_3) x_3$
$x_2 = G_1 x_1$
$x_3 = G_2 x_2$
$x_4 = G_3 x_3$

Reconstruction of original image ($x_1$) from Laplacian pyramid elements:
$x_3 = L_3 + F_3 x_4$
$x_2 = L_2 + F_2 x_3$
$x_1 = L_1 + F_1 x_2$

Slide credit: B. Freeman and A. Torralba
Laplacian pyramid reconstruction algorithm: recover $x_1$ from $L_1$, $L_2$, $L_3$ and $g_3$

Slide credit: B. Freeman and A. Torralba
ID Laplacian pyramid matrix, for $[1 \ 4 \ 6 \ 4 \ 1]$ low-pass filter

high frequencies

mid-band frequencies

low frequencies

Slide credit: B. Freeman and A. Torralba
Laplacian pyramid applications

• Texture synthesis
• Image compression
• Noise removal

The Laplacian Pyramid as a Compact Image Code

PETER J. BURT, MEMBER, IEEE, AND EDWARD H. ADELSON

Slide credit: B. Freeman and A. Torralba
Image blending

(a)

(b)

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

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

Image information occurs at all spatial scales

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

Slide credit: B. Freeman and A. Torralba
Wavelet/QMF pyramid

• Subband coding

• Wavelet or QMF (quadrature mirror filter) pyramid provides some splitting of the spatial frequency bands according to orientation (although in a somewhat limited way).

• Image is decomposed into a set of band-limited components (subbands).

• Original image can be reconstructed without error by reassembling these subbands.
2D Haar transform

Basic elements:

```
 1 1
1 -1
```

```
1 1
1 -1
```

```
1 1
1 1
```

```
1 1
1 1
```

```
1 -1
1 -1
```

```
1 -1
1 -1
```

```
1 -1
1 -1
```

```
1 -1
1 1
```

```
1 -1
1 1
```

Slide credit: B. Freeman and A. Torralba
2D Haar transform

Sketch of the Fourier transform

Horizontal low pass, Vertical low-pass

Horizontal high pass, vertical low-pass

Horizontal low pass, vertical high-pass

Horizontal high pass, vertical high pass

Slide credit: B. Freeman and A. Torralba
Pyramid cascade


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. Slide credit: B. Freeman and A. Torralba
Wavelet/QMF representation

Same number of pixels!

Slide credit: B. Freeman and A. Torralba
Image pyramids

Image information occurs at all spatial scales

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

Slide credit: B. Freeman and A. Torralba
The Steerable pyramid provides a clean separation of the image into different scales and orientations.

2 Level decomposition of white circle example:

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

Slide credit: B. Freeman and A. Torralba
Steerable Pyramid

We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below.
Steerable Pyramid

We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below.
Steerable Pyramid

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.

There is also a high pass residual…

Slide credit: B. Freeman and A. Torralba
Image pyramids

• Gaussian

• Laplacian

• Wavelet/QMF

• Steerable pyramid

Slide credit: B. Freeman and A. Torralba
Image pyramids

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

- Laplacian

- Wavelet/QMF

- Steerable pyramid

Slide credit: B. Freeman and A. Torralba
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

- Steerable pyramid

Slide credit: B. Freeman and A. Torralba
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

Slide credit: B. Freeman and A. Torralba
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.
  - Shows components at each scale and orientation separately.
  - Non-aliased subbands. Good for texture and feature analysis.
  - But overcomplete and with HF residual.

- **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.

Slide credit: B. Freeman and A. Torralba
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.

\[ \vec{F} = U\vec{f} \]

Fourier transform, or Wavelet transform, or Steerable pyramid transform

Slide credit: B. Freeman and A. Torralba
Fourier transform

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

Slide credit: B. Freeman and A. Torralba
Gaussian pyramid

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

Overcomplete representation. Transformed pixels represent bandpassed image information.

Slide credit: B. Freeman and A. Torralba
Wavelet (QMF) transform

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

Slide credit: B. Freeman and A. Torralba
Over-complete representation, but non-aliased subbands.

Steerable pyramid

Multiple orientations at one scale

Multiple orientations at the next scale

the next scale...

Pixel image

Slide: B. Freeman and A. Torralba
Why use image pyramids?

- Handle real-world size variations with a constant-size vision algorithm.
- Remove noise
- Analyze texture
- Recognize objects
- Label image features
- Image priors can be specified naturally in terms of wavelet pyramids.
Reading Assignment #2 – Hybrid Images


• Due on 23rd of April