BIL 719 - Computer Vision
Mar 5, 2014

Image pyramids

Aykut Erdem
Dept. of Computer Engineering
Hacettepe University
Image Scaling

• This image is too big to fit on the screen. How can we reduce it?
• How to generate a half-sized version?
Image Sub-Sampling

Throw away every other row and column to create a 1/2 size image - called *image sub-sampling*.

Slide credit: S. Seitz
Image Sub-Sampling

1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide credit: S. Seitz
Good and Bad Sampling

Good sampling:
- Sample often or,
- Sample wisely

Bad sampling:
- Aliasing!
Aliasing

Slide credit: F. Durand
Aliasing

• Occurs when your sampling rate is not high enough to capture the amount of detail in your image
• Can give you the wrong signal/image—an alias

• To do sampling right, need to understand the structure of your signal/image
• Enter Monsieur Fourier...

• To avoid aliasing:
  - sampling rate $\geq 2 \times$ max frequency in the image
    • said another way: $\geq$ two samples per cycle
  - This minimum sampling rate is called the **Nyquist rate**
Aliasing

• When downsampling by a factor of two
  - Original image has frequencies that are too high

• How can we fix this?
Gaussian pre-filtering

- Solution: filter the image, *then* subsample

Slide credit: S. Seitz
• Solution: filter the image, *then* subsample
Compare with...

1/2

1/4  (2x zoom)

1/8  (4x zoom)

Slide credit: S. Seitz
Gaussian pre-filtering

- Solution: filter the image, then subsample

Slide credit: N. Snavely
Gaussian pyramid

\[ \{ \]

\[ F_0 \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \cdots \]

\[ F_0 \ast H \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \cdots \]

\[ F_1 \ast H \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \rightarrow \text{blur} \]

\[ \rightarrow \text{subsample} \]

\[ \cdots \]

\[ F_2 \]

Slide credit: N. Snavely
Image pyramids

Image information occurs at all spatial scales

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

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

**GAUSSIAN PYRAMID**

\[
g_0 = \text{IMAGE} \\
g_L = \text{REDUCE} [g_{L-1}] \\
\]

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 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\
\end{bmatrix}

(Slide credit: B. Freeman and A. Torralba)

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

\[ x_3 = G_2 x_2 \]

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

\[ x_3 = G_2 G_1 x_1 \]

\[ G_2 G_1 = \]

\[
\begin{array}{cccccccccccccccccccc}
1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 30 & 16 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 25 & 16 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
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.

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

\[ x_1 \]

\[ G_1 x_1 = x_2 \]

\[ x_2 \]

\[ \begin{align*}
(I - F_3 G_3) x_3 \\
(I - F_2 G_2) x_2 \\
F_1 G_1 x_1 \\
(I - F_1 G_1) x_1
\end{align*} \]
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
1D 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
Image blending
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.
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

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
Linear transforms

\[ \tilde{F} = U \tilde{f} \quad \Leftrightarrow \quad \tilde{f} = U^{-1} \tilde{F} \]

Note: not all important transforms need to have an inverse

Slide credit: A. Torralba
Linear transforms

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

Pixels

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Slide credit: A. Torralba
## Linear transforms

\[ \mathbf{F} = \mathbf{Uf} \]

<table>
<thead>
<tr>
<th>Pixels</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[ \mathbf{U}= \]

<table>
<thead>
<tr>
<th>Derivative</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Linear transforms

$\mathbf{F} = \mathbf{Uf}$

**Pixels**

\[
\mathbf{U} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

**Derivative**

\[
\mathbf{U} = \begin{bmatrix}
1 & -1 & 0 & 0 \\
0 & 1 & -1 & 0 \\
0 & 0 & 1 & -1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

**Integration**

\[
\mathbf{U}^{-1} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

- No locality for reconstruction
- Needs boundary

Slide credit: A. Torralba
Haar transform

The simplest set of functions:

\[ U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad U^{-1} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \]

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

The simplest set of functions:

\[ U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad U^{-1} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \]

To code a signal, repeat at several locations:

\[ U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \quad U^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \]

Slide credit: A. Torralba
Haar transform

\[ \tilde{F} = Uf \]

Apply the same decomposition to the Low pass component:

![Reordering rows](image)

Low pass

High pass

And repeat the same operation to the low pass component, until length 1. Note: each subband is sub-sampled and has aliased signal components.
Haar transform

The entire process can be written as a single matrix:

![Matrix Diagram]

\[ \hat{F} = Uf \]

Slide credit: A. Torralba
Haar transform

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

Properties:
- Orthogonal decomposition
- Perfect reconstruction
- Critically sampled

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & & & & \\
1 & -1 & & & & & & \\
1 & -1 & & & & & & \\
1 & -1 & & & & & & \\
1 & -1 & & & & & & \\
1 & -1 & & & & & & \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.12 & 0.12 & 0.25 & 0 & 0.5 & 0 & 0 & 0 \\
0.12 & 0.12 & 0.25 & 0 & -0.5 & 0 & 0 & 0 \\
0.12 & 0.12 & -0.25 & 0 & 0 & 0.5 & 0 & 0 \\
0.12 & -0.12 & 0 & 0.25 & 0 & 0 & 0.5 & 0 \\
0.12 & -0.12 & 0 & 0.25 & 0 & 0 & -0.5 & 0 \\
0.12 & -0.12 & 0 & -0.25 & 0 & 0 & 0 & 0.5 \\
0.12 & -0.12 & 0 & -0.25 & 0 & 0 & 0 & -0.5 \\
\end{bmatrix}
\]
2D Haar transform

Basic elements:

\[
\begin{array}{c|c}
1 & 1 \\
1 & -1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
1 & 1 & 1 & -1 \\
\end{array}
\]
2D Haar transform

Basic elements:

\[
\begin{pmatrix}
1 & 1 \\
1 & -1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 \\
1 & 1
\end{pmatrix}
\]

Low pass
2D Haar transform

Basic elements:

\[
\begin{array}{c|c}
1 & 1 \\
\hline
1 & -1
\end{array}
\]

\[
\begin{array}{c|c}
1 & 1 \\
\hline
1 & 1
\end{array}
\]

\[
\begin{array}{c|c}
1 & -1 \\
\hline
1 & 1
\end{array}
\]

\[
\begin{array}{c|c}
1 & -1 \\
\hline
1 & -1
\end{array}
\]

Low pass
2D Haar transform

Basic elements:

\[
\begin{array}{c|c|c|c}
1 & 1 & 1 & 1 \\
-1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
1 & 1 & 1 & 1 \\
-1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 \\
\end{array}
\]

Low pass
2D Haar transform

Basic elements:

\[
\begin{bmatrix}
1 \\
1 \\
-1 \\
1
\end{bmatrix}
\quad \begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix}
\quad \begin{bmatrix}
1 \\
1 \\
1 \\
-1
\end{bmatrix}
\quad \begin{bmatrix}
1 \\
1 \\
1 \\
-1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix} = 
\begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix} 
\]

Low pass

High pass vertical

High pass horizontal

High pass diagonal
2D Haar transform

Sketch of the Fourier transform

Slide credit: B. Freeman and A. Torralba

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
Steerable Pyramid

2 Level decomposition of white circle example:

Low pass residual

Subbands

Images from: http://www.cis.upenn.edu/~eero/steerpyr.html
We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below.

---

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

Slide credit: B. Freeman and A. Torralba
We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below.

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

Slide credit: B. Freeman and A. Torralba
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.

Simoncelli and Freeman, ICIP 1995
There is also a high pass residual...
Monroe
Dog or cat?
Almost no dog information
• Summary of pyramid representations
Image pyramids

- **Gaussian**
  - Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.
  - Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

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

- **Wavelet/QMF**
  - 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**
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

transformed image

Vectorized image
Gaussian pyramid

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

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

Overcomplete representation. Transformed pixels represent bandpassed image information.
Wavelet (QMF) transform

Wavelet pyramid

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

pixel image

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...

Slide credit: 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.