Image Pyramids

**Goal:** Develop filter-based representations to decompose images into information at multiple scales, to extract features/structures of interest, and to attenuate noise.

**Motivation:**
- extract image features such as edges at multiple scales
- redundancy reduction and image modeling for
  - efficient coding
  - image enhancement/restoration
  - image analysis/synthesis

**Examples:**
- Gaussian Pyramid
- Laplacian Pyramid

**Matlab Tutorials:** imageTutorial.m and pyramidTutorial.m (up to line 200).
Linear Transform Framework

Projection Vectors: Let \( \vec{I} \) denote a 1D signal, or a vectorized representation of an image (so \( \vec{I} \in \mathcal{R}^N \)), and let the transform be

\[
\vec{a} = P^T \vec{I}.
\]  

(1)

Here,

- \( \vec{a} = [a_0, \ldots, a_{M-1}] \in \mathcal{R}^M \) are the transform coefficients.
- The columns of \( P = [\vec{p}_0, \vec{p}_1, \ldots, \vec{p}_{M-1}] \) are the projection vectors; the \( m^{th} \) coefficient, \( a_m \), is the inner product \( \vec{p}_m^T \vec{I} \)
- When \( P \) is complex-valued, we should replace \( P^T \) by the conjugate transpose \( P^{*T} \)

Sampling: The transform \( P^T \in \mathcal{R}^{M\times N} \) is said to be critically sampled when \( M = N \). Otherwise it is over-sampled (when \( M > N \)), or under-sampled (when \( M < N \)).

Basis Vectors: For many transforms of interest there is a corresponding basis matrix \( B \) satisfying

\[
\vec{I} = B \vec{a}.
\]  

(2)

The columns \( B = [\vec{b}_0, \vec{b}_1, \ldots, \vec{b}_{M-1}] \) are called basis vectors as they form a linear basis for \( \vec{I} \):

\[
\vec{I} = \sum_{m=0}^{M-1} a_m \vec{b}_m
\]
Linear Transform Framework (cont)

Completeness

- the transform is complete, encoding all image structure, if it is invertible.
- when critically sampled, it is complete if \( B = (P^T)^{-1} \) exists.
- if over-sampled, the transform is complete if \( \text{rank}(P) = N \).

In this case \( B \) is not unique – one choice is the pseudoinverse of \( P^T \)

\[
B = (PP^T)^{-1}P
\]

- if undersampled, then \( \text{rank}(P) < N \) and it is not invertible in general.

Self-Inverting

- the transform is self-inverting if \( PP^T = \alpha I_N \) for some constant \( \alpha \). Here \( I_N \) is the \( N \times N \) identity matrix. In this case, the basis matrix is simply \( B = \frac{1}{\alpha}P \).
- in the critically sampled case the transform is orthogonal (unitary), up to the constant \( \alpha \).

Example. The Fourier transform is a critically sampled, complex-valued, self-inverting linear transform (remember to use the conjugate transpose \( P^*T \)).
Gaussian Pyramid

Sequence of low-pass, down-sampled images, $[\vec{l}_0, \vec{l}_1, \ldots, \vec{l}_N]$. Usually constructed with a separable 1D kernel $h = [h_1, h_2, h_3, h_4, h_5]$, and a down-sampling factor of 2 (in each direction):

$$
\vec{l}_2 \cdot \\
\vec{l}_1 \cdot h_1 h_2 h_3 h_4 h_5 \\
\vec{l} = \vec{l}_0 \cdot \\
\ldots \\
$$

In matrix notation (for 1D) the mapping from one level to the next has the form:

$$
\vec{l}_{k+1} = R \vec{l}_k = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 1 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
\end{bmatrix} \begin{bmatrix}
\cdots \\
-h \\
-h \\
-h \\
\cdots \\
\end{bmatrix} \vec{l}_k
$$

down-sampling convolution

Typical weights for the impulse response from binomial weights

$$
h = \frac{1}{16}[1, 4, 6, 4, 1]$$
Gaussian Pyramid (cont)

Example of image and next four pyramid levels:

First three levels scaled to be the same size:

Properties of Gaussian pyramid:

- used for multi-scale edge estimation,
- efficient to compute coarse scale images. Only 5-tap 1D filter kernels are used,
- highly redundant, coarse scales provide much of the information in the finer scales.
Laplacian Pyramid

Over-complete decomposition based on difference-of-lowpass filters; the image is recursively decomposed into low-pass and highpass bands.

- Each band of the Laplacian pyramid is the difference between two adjacent low-pass images of the Gaussian pyramid, \([\vec{l}_0, \vec{l}_1, \ldots, \vec{l}_N]\).

That is:

\[
\vec{b}_k = \vec{l}_k - E\vec{l}_{k+1}
\]

where \(E\vec{l}_{k+1}\) is an up-sampled, smoothed version of \(\vec{l}_{k+1}\) (so that it will have the same dimension as \(\vec{l}_k\)).

\[
E\vec{l}_{k+1} = \begin{bmatrix}
\ddots & -g & -g & -g & \\
-g & \ddots & -g & -g & \\
-g & -g & \ddots & -g & \\
& & & \ddots & -g
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \ddots \\
0 & 0 & 0 & \ddots & \ddots
\end{bmatrix}
\begin{bmatrix}
\vec{l}_{k+1}
\end{bmatrix}
\]

Often the filters used to construct the Gaussian and Laplacian pyramids, \(g\) and \(h\), are identical.

The Laplacian pyramid with \(L\) levels is given by \([\vec{b}_0, \vec{b}_1, \ldots, \vec{b}_{L-1}, \vec{l}_L]\).

The representation is overcomplete by a factor of roughly \(\frac{4}{3}\) for 2D images (i.e. \(1 + 1/4 + 1/16 + \ldots = 4/3\)).
Laplacian Pyramid (cont)

Construction of the Laplacian bands:

A Laplacian pyramid with four levels:
Laplacian Pyramid (cont)

**Construction:** of \([\vec{b}_0, \vec{b}_1, \ldots, \vec{b}_{L-1}, \vec{l}_L]\).

\[
\begin{align*}
\vec{l}_0 &= \vec{I} \\
\vec{l}_{k+1} &= R \vec{l}_k \\
\vec{b}_k &= \vec{l}_k - E \vec{l}_{k+1}
\end{align*}
\]

**Reconstruction:** of \(\vec{I}\) is exact (for any filters) and straightforward:

\[
\begin{align*}
\vec{l}_k &= \vec{b}_k + E \vec{l}_{k+1} \\
\vec{I} &= \vec{l}_0
\end{align*}
\]

**System Diagram:** shows the filters and sampling steps used to compute the pyramid, and to then reconstruct the image from the transform coefficients. Gaussian pyramid levels are computed using \(h(n)\). Filter \(g(n)\) is used with up-sampling so that adjacent Gaussian levels can be subtracted.

Analysis/synthesis diagram for a 2-layer Laplacian pyramid
Laplacian Pyramid Filters

In practice:

- often use same filters for \( h \) and \( g \) (apply same operators for smoothing and interpolation in construction and reconstruction)
- use separable lowpass filters
- desire isotropy so all orientations handled the same way.

Constraints on 5-tap lowpass filter \( h \):

- even-symmetry means that taps are \( h = \left( \frac{a_2}{2}, \frac{a_1}{2}, a_0, \frac{a_1}{2}, \frac{a_2}{2} \right) \).
- assume that dc signal is preserved, i.e. \( \hat{h}(0) = 1 \):
  \[
  \hat{h}(0) = \sum_{n=-2}^{2} h(n) e^{-in} = a_0 + a_1 + a_2 = 1.
  \]
- assume that spectrum decays to 0 at fold-over rate, i.e. \( \hat{h}(\pi) = 0 \):
  \[
  \hat{h}(\pi) = \sum_{n=-2}^{2} h(n) e^{-in} = a_0 - a_1 + a_2 = 0.
  \]
- So there are two linear equations for the three unknowns \( a_0 \), \( a_1 \), and \( a_2 \). There is therefore one free degree of freedom.
- For example, choose \( a_0 = \frac{6}{16} \), then \( h(n) \) is the binomial 5-tap filter
  \[
  h(n) = \frac{1}{16} (1, 4, 6, 4, 1).
  \]
On the name “Laplacian”

The well-known Laplacian derivative operator (isotropic second derivative) is given by

\[
\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

For Gaussian kernels, \( g(x; \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2} \),

\[
\frac{dg(x; \sigma)}{dx} = \frac{-x}{\sigma^2} g(x; \sigma)
\]

\[
\frac{d^2 g(x; \sigma)}{dx^2} = \left(\frac{x^2}{\sigma^2} - 1\right) \frac{1}{\sigma^2} g(x; \sigma)
\]

\[
\frac{dg(x; \sigma)}{d\sigma} = \left(\frac{x^2}{\sigma^2} - 1\right) \frac{1}{\sigma} g(x; \sigma)
\]

Therefore

\[
\frac{d^2 g(x; \sigma)}{dx^2} = c_0(\sigma) \frac{dg(x; \sigma)}{d\sigma} \approx c_1(\sigma) (g(x; \sigma) - g(x; \sigma + \Delta \sigma))
\]

That is, if the low-pass filter \( h \) used to create the Laplacian pyramid is Gaussian, then the Laplacian pyramid levels approximate the second derivative of the image at scale \( \sigma \).
Uses of Laplacian Pyramid: Coding

Multiscale image representations are natural for image coding and transmission. The same basic ideas underly jpeg encoding.

**Approach:** Use quantization levels that become more coarse as one moves to higher frequency pass bands.

- high frequency coefficients are more coarsely coded (i.e., to fewer bits) than lower frequency bands.
- this quantization matches human contrast sensitivity
- vast majority of the coefficients are in high frequency bands.

**Advantages:**

- Eliminates blocking artifacts of JPEG at low frequencies because of the overlapping basis functions.
- approach also allows for progressive transmission, since low-pass representations are reasonable approximations to the image.
- coding and image reconstruction are simple
Uses of Laplacian Pyramid: Restoration (Coring)

Transform coefficients for the Laplacian transform are often near zero. Significantly non-zero values are generally sparse.

Histograms of transform coefficients are often well approximated by a so-called ”generalized Laplacian” density, \( c e^{-|x/s|^k} \), where

- \( \gamma \) is usually between 0.7 and 1.2
- \( s \) controls the variance
- peaked at 0, with heavy tails

Coring:

- set all sufficiently small transform coefficients to zero,
- leave others unchanged, and possibly clip at large magnitudes.


Uses of Laplacian Pyramid: Image Compositing

**Goal:** Seamlessly stitch together images into an image mosaic (i.e., *register* the images and *blurring* the boundary), by smoothing the boundary in a scale-dependent way to avoid boundary artifacts.

**Method:**

- assume images $I_1(\overline{n})$ and $I_2(\overline{n})$ are registered and let $m_1(\overline{n})$ be a mask that is 1 at pixels where we want the brightness from $I_1(\overline{n})$ and 0 otherwise (i.e., where we want to see $I_2(\overline{n})$).
- create Gaussian pyramid for $m_1(\overline{n})$, denoted $\{l_0(\overline{n}), l_1(\overline{n}), ..., l_L(\overline{n})\}$
- create Laplacian pyramids for $I_1(\overline{n})$ and $I_2(\overline{n})$, denoted by
  \[
  \{b_{1,0}(\overline{n}), ..., b_{1,L-1}(\overline{n}), l_{1,L}(\overline{n})\} \quad \text{and} \quad \{b_{2,0}(\overline{n}), ..., b_{2,L-1}(\overline{n}), l_{2,L}(\overline{n})\}
  \]
- create blended pyramid $\{b_{0,0}(\overline{n}), ..., b_{0,L-1}(\overline{n}), l_{0,L}(\overline{n})\}$ where
  \[
  b_{0,j}(\overline{n}) = b_{1,j}(\overline{n}) l_j(\overline{n}) + b_{2,j}(\overline{n}) (1 - l_j(\overline{n}))
  \]
  \[
  l_{0,L}(\overline{n}) = l_{1,L}(\overline{n}) l_L(\overline{n}) + l_{2,L}(\overline{n}) (1 - l_L(\overline{n}))
  \]
- collapse blended pyramid to reconstruct image
Uses of Laplacian Pyramid: Enhancement

**Goal:** Create a high fidelity image from a set of images taken with different focal lengths, shutter speeds, etc.

- Images with different focal lengths will have different image regions in focus.
- Images with different shutter speeds may have different contrast and luminance levels in different regions.

**Approach:**

- Given pyramids for two images $I_1(\vec{n})$ and $I_2(\vec{n})$, construct 2 or 3 levels of a Laplacian pyramid:

\[
\{b_{1,0}(\vec{n}), \ldots, b_{1,L-1}(\vec{n}), l_{1,L}(\vec{n})\} \quad \text{and} \quad \{b_{2,0}(\vec{n}), \ldots, b_{2,L-1}(\vec{n}), l_{2,L}(\vec{n})\}
\]

- At level $j$, define a mask $m(\vec{n})$ that is 1 when $|b_{1,j}(\vec{n})| > |b_{2,j}(\vec{n})|$ and 0 elsewhere.

- Then form the blended pyramid with levels $b_{0,j}[\vec{n}]$ given by

\[
b_{0,j}[\vec{n}] = m[\vec{n}] b_{1,j}[\vec{n}] + (1 - m[\vec{n}]) b_{2,j}[\vec{n}]
\]

- Averaged the low-pass bands from the two pyramids.