GENERAL

- Recitation for Problem Set 1 this Friday!
  - Recitation from 9am-10am.
  - Followed by regular office hours from 10am-11am.
  - Use Adith’s OH Zoom link in Canvas.
- You should have started working on problem set 1.
  - Today’s class will cover the last remaining topic (wavelets, for the last question).
- Make sure you see the feedback from problem set 0
  - Don’t use Open CV or external libraries.
  - Always remember to fill out the “information section”.
- Take the academic integrity policy seriously!

EFFICIENT COMPUTATION

Recursive Computation

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

- Sometimes can decompose into convolution with sparse kernels.
- Many implementations of convolve2d won’t make use of sparsity.
  - But you can write your own.

Smooth and Sub-sample

- Don’t smooth and subsample!
  - For sub-sampling by two, you’re computing 4x as many smooth filter outputs as you need to.

- Similarly, using zero-filling + convolution for upsampling is inefficient.
**EFFICIENT COMPUTATION**

**numpy Specifics**

1. In general, prefer algorithms that have lower total number of multiplies / adds.
2. Try to use `scipy.signal.convolve2d` (subject to rule 1). It is optimized for cache reads, parallel execution, etc.
   `(import from scipy.signal import convolve2d as conv2)

3. Similarly, avoid for loops and use element-wise operations on large arrays, matrix multiplies (np.matmul / np.dot), etc.

- Some of these things are faster in python because a single large operation runs natively instead of returning to the compiler. But they're also faster because these operations are often 'atomics' in lower-level libraries too (BLAS), and have been highly optimized for modern hardware.
- Thinking about designing your algorithm in terms of these atomic operations is useful beyond python.
- Some points in problem sets allocated for efficient code.

**MULTI-SCALE REPRESENTATIONS**

Fourier Transform useful for:

- Diagonalizing Convolution $A_k = SD_kS^*$
- Concentrating energy (high magnitudes) in fewer coefficients

- But is the FT interpretable? Does it tell us something about the image?

---

**EFFICIENT COMPUTATION**

**Notes on Parallelization**

If you are having trouble coming up with efficient implementations:

- First write your code with for loops.
  - This will be a good starting point to "see" what operations can be parallelized. Also, you can use its outputs as a basis of comparison.
  - This can also be a valid submission—you’ll lose some points for being inefficient, but still get most of the points if it is correct.
- Try to then see if you can re-arrange the for loops so that the ‘longest’ ones—that loop over image pixels—are inner-most.
- Then see if you can replace as many loops as possible (starting with inner-most ones) with element-wise operations / convolutions.

**MULTI-SCALE REPRESENTATIONS**

Fourier Transform useful for:

- Diagonalizing Convolution $A_k = SD_kS^*$
- Concentrating energy (high magnitudes) in fewer coefficients

- But is the FT interpretable? Does it tell us something about the image?
MULTI-SCALE REPRESENTATIONS

Fourier Transform useful for:

Diagonalizing Convolution \( A_x = S D_x S^* \)

Concentrating energy (high manitudes) in fewer coefficients

\[
\begin{bmatrix}
+ j & + j \\
+ j & + j \\
\end{bmatrix}
\]

But is the FT interpretable? Does it tell us something about the image?

MULTI-SCALE REPRESENTATIONS

Gaussian Pyramid

\[
\text{Gaussian Pyramid}
\]

MULTI-SCALE REPRESENTATIONS

\( F[u, v] \) is intuitively average variation in image at that frequency.

But averaged across the entire image.

This isn’t useful because images aren’t “stationary”

- Different parts of the image, “have different frequencies”.

FT decomposition of different levels (coarse/fine) of variation: but without sense of spatial location.

Multi-scale representations aim to address this.
MULTI-SCALE REPRESENTATIONS

Gaussian Pyramid

Useful for analyzing image at multiple scales.
E.g., apply the same method (edge detection / CNN) at multiple levels of the pyramid.

MULTI-SCALE REPRESENTATIONS

Laplacian Pyramid

\( X \rightarrow X - X \ast G_{\sigma} \)
**MULTI-SCALE REPRESENTATIONS**

**Laplacian Pyramid**

No orientation selectivity: See Steer's SteerPyramids
http://www.cns.nyu.edu/~eero/steerpy/

---

**Wavelet Pyramid**

Gaussian & Laplacian pyramids are good for analysis, but not really a representation.
Not easy / possible to go from pyramid coefficients back to original image (like Fourier Transform).

Consider a 2x2 Pixel Block

\[
\begin{pmatrix}
  a & c \\
  b & d \\
\end{pmatrix}
\]

\[
E = \frac{a + b + c + d}{2}
\]

\[
L_E = \frac{E - a - b}{2}
\]

\[
L_{E_E} = \frac{E - c}{2}
\]

\[
L_{E_C} = \frac{E - d}{2}
\]

---

**MULTI-SCALE REPRESENTATIONS**
MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Gaussian & Laplacian pyramids are good for analysis, but not really a representation. Not easy / possible to go from pyramid coefficients back to original image (like Fourier Transform).

Consider a 2x2 Pixel Block

$$L = \frac{a + b + c + d}{2} \quad H_1 = \frac{b - a - c}{2}$$
$$H_2 = \frac{a + d - c}{2} \quad H_3 = \frac{a + d - b}{2}$$

$$\begin{bmatrix} L \\ H_1 \\ H_2 \\ H_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Unitary Matrix

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Gaussian & Laplacian pyramids are good for analysis, but not really a representation. Not easy / possible to go from pyramid coefficients back to original image (like Fourier Transform).

Consider a 2x2 Pixel Block

$$L = \frac{a + b + c + d}{2} \quad H_1 = \frac{a - b - c}{2}$$
$$H_2 = \frac{c + d - a}{2} \quad H_3 = \frac{a + d - b}{2}$$

For every non-overlapping 2x2 Patch in the Image

$$\begin{bmatrix} L \\ H_1 \\ H_2 \\ H_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Unitary Matrix / Co-ordinate transform
MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

For every non-overlapping 2D patch in the image

\[ L = \frac{a + b + c + d}{2} \]
\[ H_1 = \frac{b + d - a - c}{2} \]
\[ H_2 = \frac{c + d - a - b}{2} \]
\[ H_3 = \frac{a + d - b - c}{2} \]

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Can be achieved by Convolution + Downsample

\[ L = \frac{a + b + c + d}{2} \]
\[ H_1 = \frac{b + d - a - c}{2} \]
\[ H_2 = \frac{c + d - a - b}{2} \]
\[ H_3 = \frac{a + d - b - c}{2} \]

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Still a Coordinate Transform!

\[ L = \frac{a + b + c + d}{2} \]
\[ H_1 = \frac{b + d - a - c}{2} \]
\[ H_2 = \frac{c + d - a - b}{2} \]
\[ H_3 = \frac{a + d - b - c}{2} \]

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Can Invert to get Image

\[ L = \frac{a + b + c + d}{2} \]
\[ H_1 = \frac{b + d - a - c}{2} \]
\[ H_2 = \frac{c + d - a - b}{2} \]
\[ H_3 = \frac{a + d - b - c}{2} \]
MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Apply Recursively

\[
L = \frac{a + b + c + d}{2} \\
H_1 = \frac{a + d - a - b}{2} \\
H_2 = \frac{b + d - a - c}{2} \\
H_3 = \frac{a + d - b - c}{2}
\]

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Apply Recursively

\[
L = \frac{a + b + c + d}{2} \\
H_1 = \frac{a + d - a - b}{2} \\
H_2 = \frac{b + d - a - c}{2} \\
H_3 = \frac{a + d - b - c}{2}
\]

Wavelet Transform

"Harr"
Others based on different filters
MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Applications: Analysis, image modeling & restoration, compression

MULTI-SCALE REPRESENTATIONS

Wavelet Pyramid

Applications: Analysis, image modeling & restoration, compression
Statistics / Estimation Recap

Setting

- There is a true image (or image-like object) $x$ that we want to estimate
- What we have is some degraded observation $y$ that is based on $x$
- The degradation might be “stochastic”: so we say $p(y|x)$

Simplest Case (single pixel; $x, y \in \mathbb{R}$):

$$y = x + \sigma \epsilon \rightarrow p(y|x) = \mathcal{N}(y; \mu = x, \sigma^2)$$

Estimate $x$ from $y$: “denoising”

---

What we really want to do is maximize $p(x|y)$, not the likelihood $p(y|x)$

- $p(y|x)$ is the distribution of observation $y$ given true image $x$
- $p(x|y)$ is the distribution of true $x$ given observation $y$
- Bayes Rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Simple Derivation

$$p(x, y) = p(x)p(y|x) = p(y)p(x|y) \rightarrow p(y) = \frac{p(y|x)p(x)}{p(y)}$$

$$p(y) = \int p(y|x')p(x')dx'$$

---

$p(y|x)$ is our observation model, called the likelihood function

- Likelihood of observation $y$ given true image $x$
- Maximum Likelihood Estimator: $\hat{x} = \arg \max_x p(y|x)$

For $p(y|x) = \mathcal{N}(y; x, \sigma^2)$, what is ML Estimate?

$$p(y|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(y - x)^2}{2\sigma^2} \right)$$

$$\hat{x} = y$$

That’s a little disappointing.

---

$p(x|y)$: Likelihood

- $p(y|x)$: “Prior” Distribution on $x$
- $p(x)$: “Prior” Distribution on $x$

What we believe about $x$ before we see the observation

$p(x|y)$: Posterior Distribution of $x$ given observation $y$

- What we believe about $x$ after we see the observation
Statistics / Estimation Recap

Maximum A Posteriori (MAP) Estimation

\[ \hat{x} = \arg \max_x p(x|y) \]

- The most likely answer under the posterior distribution
- Other Estimators Possible: Try to minimize some "risk" or loss function \( L(\hat{x}, x) \)
  - Measures how bad an answer \( \hat{x} \) is when true value is \( x \)
- Minimize Expected Risk Under Posterior

\[ \hat{x} = \arg \min_x \int L(x, x') p(x'|y) dx' \]

Let's say \( L(x, x') = (x - x')^2 \). What is \( \hat{x} \)?

\[ \hat{x} = \mathbb{E}_{p(x|y)} x = \int x p(x|y) dx \]

Back to Denoising

Let's choose a simple prior:

\[ p(x) = \mathcal{N}(x; 0.5, 1) \]

\[ \hat{x} = \arg \max_x - \log p(y|x) - \log p(x) \]

(Turn this into minimization of a cost)
Let’s choose a simple prior:

\[ p(x) \sim \mathcal{N}(x; 0.5, 1) \]

\[ p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right) \]

\[ -\log p(x) = \min_x \frac{(y-x)^2}{2\sigma^2} + (x-0.5)^2 + C' \]

\[ \hat{x} = \arg \min_x \frac{(y-x)^2}{2\sigma^2} + (x-0.5)^2 + C' \]

\[ p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right) \]

\[ p(x) \sim \mathcal{N}(x; 0.5, 1) \]

How do you compute \( \hat{x} \)?

- In this case, simpler option available. Complete the squares, express as a single quadratic \( \propto (x - \hat{x})^2 \).
- More generally, to minimize \( C(x) \), find \( C'(x) = 0 \), check \( C''(x) > 0 \)

\[ C'(x) = \frac{x-y}{\sigma^2} + x - 0.5 \]
Maximum A Posteriori (MAP) Estimation

\[ \hat{x} = \arg \min_x \left( \frac{(y - x)^2}{2\sigma^2} + \frac{(x - 0.5)^2}{2} \right) \]

\[ C'(x) = \frac{x - y}{\sigma^2} + x - 0.5 = 0 \]

\[ \hat{x} = \frac{y + 0.5\sigma^2}{1 + \sigma^2} \]

\[ C''(x) = \frac{1}{\sigma^2} + 1 > 0 \quad \forall x \]

- Means the function is convex (quadratic functions with a positive coefficient for \(x^2\) are convex)

Let’s go beyond “single pixel images”: \( Y[n] = X[n] + \epsilon[n] \)

- If noise is independent at each pixel

\[ p\left(\{Y[n]\} | \{X[n]\}\right) = \prod_n p(Y[n]|X[n]) = \prod_n \mathcal{N}(Y[n]; X[n], \sigma^2) \]

- Similarly, if prior \( p(\{X[n]\}) \) is defined to model pixels independently:

\[ p(\{X[n]\}) = \prod_n \mathcal{N}(X[n]; 0.5, 1) \]

- Product turns into sum after taking log

Note: This is a minimization over multiple variables

- But since the different variables don’t interact with each other, we can optimize each pixel separately

\[ \hat{X}[n] = \arg \min_{X[n]} \sum_n \left( \frac{(Y[n] - X[n])^2}{2\sigma^2} + \frac{(X[n] - 0.5)^2}{2} \right) \]

\[ \hat{X}[n] = \frac{Y[n] + 0.5\sigma^2}{1 + \sigma^2} \quad \forall n \]
IMAGE RESTORATION

Multi-variate Gaussians

- Re-interpret these as multi-variate Gaussians

For $X \in \mathbb{R}^d$:

$$p(X) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp \left( -\frac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) \right)$$

- Here, $\mu \in \mathbb{R}^d$ is a mean “vector”, same size as $X$
- Co-variance $\Sigma$ is a symmetric, positive-definite matrix $d \times d$ matrix

When the different elements of $X$ are independent, off-diagonal elements of $\Sigma$ are 0.

How do we represent our prior and likelihood as a multi-variate Gaussian?

$$X \in \mathbb{R}^d \quad p(X) = \mathcal{N}(X; \mu, \Sigma)$$

$$X \in \mathbb{R}^d \quad \mu \in \mathbb{R}^d \quad \Sigma \in \mathbb{R}^{d \times d}$$

But now, we can use these multi-variate distributions to model correlations and interactions between different pixels.

Let’s stay with denoising, but talk about defining a prior in the Wavelet domain.

Instead of saying, individual pixel values are independent, let’s say individual wavelet coefficients are independent.

Let’s also put different means and variances on different wavelet coefficients.

All the ‘derivative’ coefficients have zero mean.

Variance goes up as we go to coarser levels.

More specifically:

Represent $X$ and $Y$ as vectorized images

$$p(Y|X) = \mathcal{N}(Y; X, \sigma^2 I)$$

Here the mean-vector for $Y$ is $X$

The co-variance matrix is a diagonal matrix with all diagonal entries $= \sigma^2$

$$p(X) = \mathcal{N}(X; \mu, \Sigma)$$

$$\mu = X, \Sigma = \sigma^2 I$$

$$\det(\Sigma) = (\sigma^2)^d$$

$$\Sigma^{-1} = \sigma^{-2} I$$

$$(Y - \mu)^T \Sigma^{-1} (Y - \mu) = (Y - X)^T \frac{1}{\sigma^2} I (Y - X) = \frac{1}{\sigma^2} (Y - X)^T I (Y - X)$$

$$= \frac{1}{\sigma^2} (Y - X)^T (Y - X) = \frac{1}{\sigma^2} \|Y - X\|^2$$

$$= \frac{1}{\sigma^2} \sum_a (Y[a] - X[a])^2$$

$$\tilde{X} = \arg \min_X \frac{1}{\sigma} \|Y - X\|^2 + \|X - 0.5\|^2$$

Thus, we model $X$ as a Gaussian with mean $\mu$ and variance $\Sigma$. We can modify this prior by adding a penalty term to encourage sparsity.
Let $C = W X$, where $W$ represents the Wavelet transform matrix

- $W$ is unitary, $W^{-1} = W^T$
- $X = W^T C$

Define our prior on $C$:

$$p(C) = \mathcal{N}(C; \mu_c, D_c)$$

- Now, $D_c$ is a diagonal matrix, with entries equal to corresponding variances of coefficients
- Off-diagonal elements 0 implies Wavelet co-efficients are un-correlated
- A prior on $C$ implies a prior on $X$
  - $C$ is just a different representation for $X$