Image analysis

CS/CME/BioE/Biophys/BMI 279
Oct. 24, 2019
Adrian Sanborn
Ron Dror
Assignment 2 tips

• Your predictors will not perfectly recapitulate the structure of HRAS, because protein structure prediction is challenging.
• But you should get a fairly compact structure.
• TAs have added extra office hours next week.
Outline

• Images in molecular and cellular biology
• Reducing image noise
  – Mean and Gaussian filters
  – Frequency domain interpretation
  – Median filter
• Sharpening images
• Image description and classification
• Practical image processing tips
Images in molecular and cellular biology
Most of what we know about the structure of cells comes from imaging:

- **Light microscopy**, including fluorescence microscopy
  ![Image of fluorescence microscopy](https://www.microscopyu.com/articles/livecellimaging/livecellmaintenance.html)

- **Electron microscopy**
  ![Image of electron microscopy](http://blog.library.gsu.edu/wp-content/uploads/2010/11/mtdna.jpg)
Imaging can capture structure over time

Human white blood cells eating bacteria

Nucleus
S. aureus (GFP)

Videos by Lorenzo Labitiga
Imaging is pervasive in structural biology

- The experimental techniques used to determine macromolecular (e.g., protein) structure also depend on imaging.
Computation plays an essential role in these imaging-based techniques

- Some techniques require substantial computation before you can even see the image
- We will start with analysis of microscopy data, because it’s closest to our everyday experience with images
- In fact, the basic image analysis techniques we’ll cover initially also apply to normal photographs
Representations of an image

• Recall that we can think of a grayscale image as:
  – A function of two variables \((x\text{ and } y)\)
  – A two-dimensional array of brightness values
  – A matrix (of brightness values)

• A color image can be treated as:
  – Three separate images, one for each color channel (red, green, blue)
  – A function that returns three values (red, green, blue) for each \((x, y)\) pair
Reducing image noise
Experimentally determined images are always corrupted by *noise*

- “Noise” means any deviation from what the image would ideally look like

http://www.siox.org/pics/pferd-noisy.jpg
Image noise

Original image

Noisy images

“Gaussian noise”: normally distributed noise added to each pixel

“Salt and pepper noise”: random pixels replaced by very bright or dark values
Reducing image noise

Mean and Gaussian filters
How can we reduce the noise in an image?

- The simplest way is to use a “mean filter”
  - Replace each pixel by the average of a set of pixels surrounding it
  - For example, a 3x3 mean filter replaces each pixel with the average of a 3x3 square of pixels
  - This is equivalent to convolving the image with a 3x3 matrix:

\[
\begin{bmatrix}
\frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\
\frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\
\frac{1}{9} & \frac{1}{9} & \frac{1}{9}
\end{bmatrix}
= \frac{1}{9}
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

Why 1/9?
Values should sum to 1, so that overall brightness of the image remains constant.
Mean filter

Original images

Result of 3x3 mean filter
Mean filter

Original images

Result of 3x3 mean filter

A larger filter (e.g., 5x5, 7x7) would further reduce noise, but would blur the image more.
A better choice: use a smoother filter

- We can achieve a better tradeoff between noise reduction and distortion of the noise-free image by convolving the image with a smoother function.
- One common choice is a (two-dimensional) Gaussian.
- Rather than choosing exact size, choose the standard deviation.

\[
\begin{bmatrix}
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
0.022 & 0.097 & 0.159 & 0.097 & 0.022 \\
0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
\end{bmatrix}
\]

standard deviation = 1 pixel
Gaussian filter

Original images

Result of Gaussian filter (standard deviation $\sigma = 1$ pixel)
Mean filter (for comparison)

Original images

Filtered images
Gaussian filter (for comparison)

Original images

Filtered images
Reducing image noise

Frequency domain interpretation
Low-pass filtering

• Because the mean and Gaussian filters are convolutions, we can express them as multiplications in the frequency domain.

• Both types of filters reduce high frequencies while preserving low frequencies. They are thus known as low-pass filters.

• These filters work because real images have mostly low-frequency content, while noise tends to have a lot of high-frequency content.
Low-pass filtering

Filter in real domain

Magnitude profile in frequency domain (low frequencies are near center of plots)

Point value in the frequency domain describes the strength of the corresponding frequency in the real domain.
Low-pass filtering

Filter in real domain

Magnitude profile in frequency domain (low frequencies are near center of plots)

The Gaussian filter eliminates high frequencies more effectively than the mean filter, making the Gaussian filter better by most measures.
Low-pass filtering

- As a filter becomes larger (wider), its Fourier-domain representation becomes narrower.
- In other words, making a mean or Gaussian filter larger will make it more low-pass (i.e., narrow the range of frequencies it passes).
  - Thus it will eliminate noise better, but blur the original image more.
Reducing image noise

Median filter
Median filter

- A median filter ranks the pixels in a square surrounding the pixel of interest, and picks the middle value.
- This is particularly effective at eliminating noise that corrupts only a few pixel values (e.g., salt-and-pepper noise).
- This filter is not a convolution.
Median filter

Original images

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

Result of 3x3 median filter

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
Median filter

Original images

Result of 3x3 median filter

Using a larger window would further reduce noise, but would distort the image more
Sharpening images
High-pass filter

- A high-pass filter removes (or reduces) low-frequency components of an image, but not high-frequency ones.
- The simplest way to create a high-pass filter is to subtract a low-pass filtered image from the original image.
  - This removes the low frequencies but preserves the high ones.
  - The filter matrix itself can be computed by subtracting a low-pass filter matrix (that sums to 1) from an “identity” filter matrix (all zeros except for a 1 in the central pixel).

Original image  Low-pass filtered image  High-pass filtered image
High-pass filter

Horizontal edge in original image

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
9 & 9 & 9 & 9 & 9 \\
9 & 9 & 9 & 9 & 9 \\
\end{bmatrix}
\]

Low-pass filtered (3 x 3 mean filter)

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
3 & 3 & 3 & 3 & 3 \\
6 & 6 & 6 & 6 & 6 \\
9 & 9 & 9 & 9 & 9 \\
\end{bmatrix}
\]

High-pass filtered (edges are emphasized)

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
-3 & -3 & -3 & -3 & -3 \\
3 & 3 & 3 & 3 & 3 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
How might one use a high-pass filter?

• To highlight edges in the image
• To remove any “background” brightness that varies smoothly across the image
• Image sharpening
  – To sharpen the image, one can add a high-pass filtered version of the image (multiplied by a fractional scaling factor) to the original image
  – This increases the high-frequency content relative to low-frequency content
  – In photography, this is called “unsharp masking”

![Original image](https://upload.wikimedia.org/wikipedia/commons/0/0b/Accutance_example.png)

<table>
<thead>
<tr>
<th>Original image</th>
<th>Mild sharpening (small scale factor)</th>
<th>Stronger sharpening (larger scale factor)</th>
</tr>
</thead>
</table>

https://upload.wikimedia.org/wikipedia/commons/0/0b/Accutance_example.png
Image sharpening — another example

Original image

Sharpened image

Image description and classification
Describing images concisely

• The space of all possible images is very large. To fully describe an $N$-by-$N$ pixel grayscale image, we need to specify $N^2$ pixel values.

• We can thus think of a single image as a point in an $N^2$-dimensional space.

• Classifying and analyzing images becomes easier if we can describe them (even approximately) with fewer values.

• For many classes of images, we can capture most of the variation from image to image using a small number of values
  – This allows us to think of the images as points in a lower-dimensional space
  – We’ll examine one common approach: Principal Components Analysis.
Principal component analysis (PCA)

• Basic idea: given a set of points in a multi-dimensional space, we wish to find the linear subspace (line, plane, etc.) that best fits those points.

Werner et al., PLoS ONE 2014
Principal component analysis (PCA)

• Basic idea: given a set of points in a multi-dimensional space, we wish to find the linear subspace (line, plane, etc.) that best fits those points.

• If we want to specify a point $\mathbf{x}$ with just one number (instead of two), we can specify the closest point to $\mathbf{x}$ on the line described by the first principal component (i.e., project $\mathbf{x}$ onto that line).

• In a higher dimensional space, we might specify the point lying closest to $\mathbf{x}$ on a plane specified by the first two principal components.
Principal component analysis (PCA)

- How do we pick the principal components?
  - First subtract off the mean value of the points, so that the points are centered around the origin.
  - The first principal component is chosen to minimize the sum squared distances of the points to the line it specifies.
  - This is equivalent to picking the line that maximizes the variance of the full set of points after projection onto that line.
  - The $k$th principal component is calculated the same way, but required to be orthogonal to previous principal components.
Example: face recognition

- A popular face recognition algorithm relies on PCA
  - Take a large set of face images (centered, frontal views)
  - Calculate the first few principal components
  - Approximate each new face image as a sum of these first few principal components (each multiplied by some coefficient).
  - Classify faces by comparing these coefficients to those of the original face images

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm
Practical image processing tips
Practical image processing tips
(Thanks to Leo Kesselman)

• When viewing an image, you might need to scale all the intensity values up or down so that it doesn’t appear all black or all white
• You might also need to adjust the “gamma correction” to get the image to look right
  – This applies a nonlinear (but monotonically increasing) function to each pixel value
• ImageJ is a great program for looking at scientific images

You’re not responsible for this material, but you may find it useful