Image analysis

CS/CME/BioE/Biophys/BMI 279

Oct. 31, 2023

Ron Dror

Thanks for filling out the feedback survey!

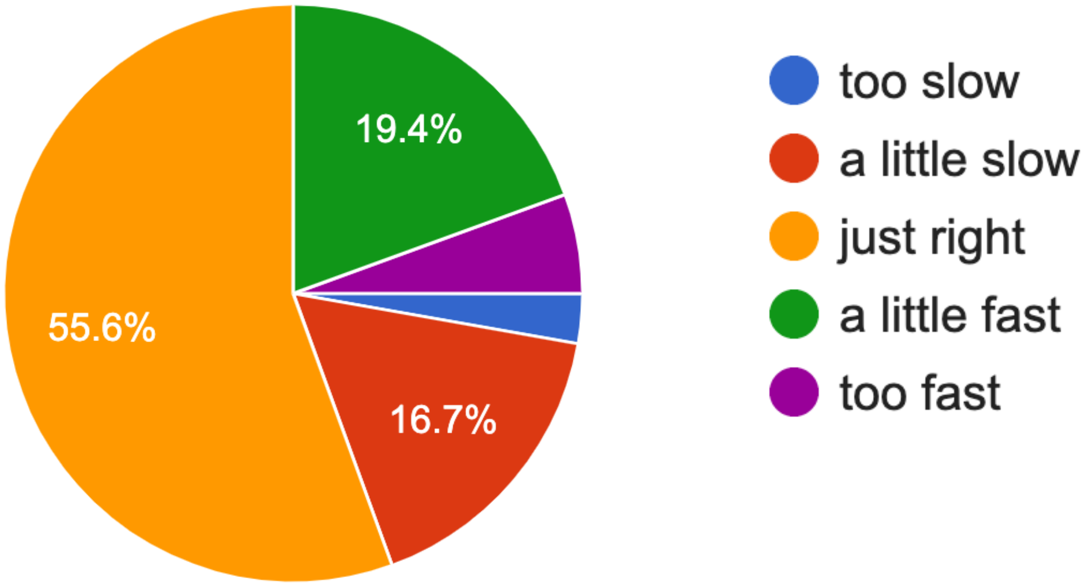
- Lots of positive feedback
- Some concerns and suggestions for improvement, which are especially valuable!
- Additional feedback is welcome at any point

Planned changes in response to your feedback

- We’ll add more office hours before deadlines, including on weekends
- We’ll hold optional coding tutorials for assignment 3
  - The first will take place on Friday. The second will take place next week and will cover additional materials. Recordings available afterwards.
  - Notes
    - Assignment 3 due Nov. 16. Available on course website later today.
    - We estimate the challenge problem will take 5–7 hours (purely optional, for extra credit)
- We’ll post sample questions for final exam on website

Feedback indicated we should leave many things as is

- Responses to “How is the pace of the class?”



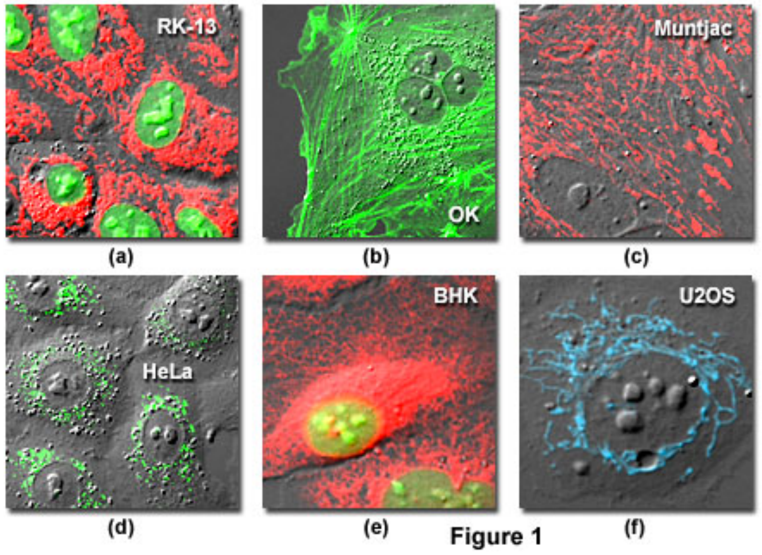
Outline

- 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

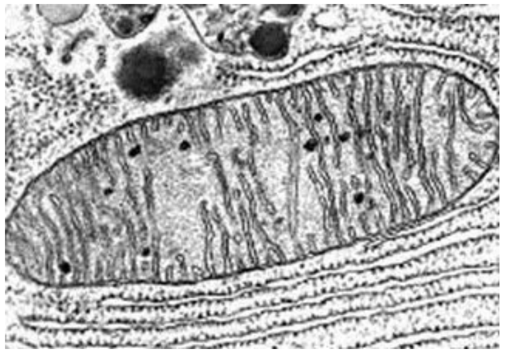
Most of what we know about the structure of cells comes from imaging

- Light microscopy, including fluorescence microscopy



https://www.microscopyu.com/articles/livecellimaging/livecellmaintenance.html

- Electron microscopy

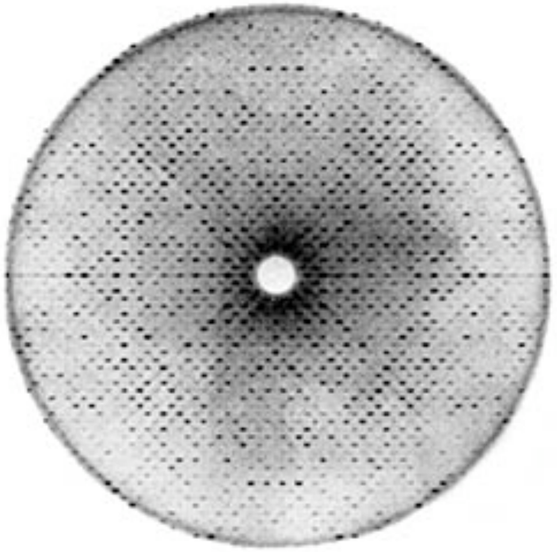


http://blog.library.gsu.edu/wp-content/uploads/2010/11/mtdna.jpg

Imaging is pervasive in structural biology

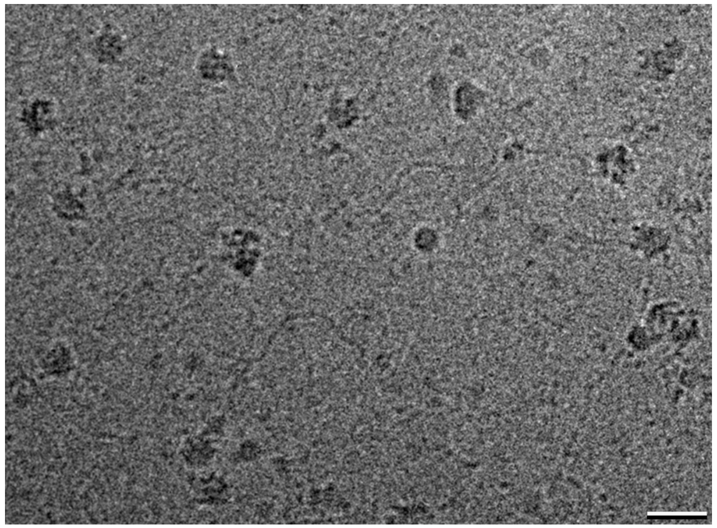
- The experimental techniques used to determine macromolecular (e.g., protein) structure also depend on imaging

X-ray crystallography



https://askabiologist.asu.edu/sites/default/files/resources/articles/crystal_clear/CuZn_SOD_C2_x-ray_diffraction_250.jpg

Single-particle cryo-electron microscopy



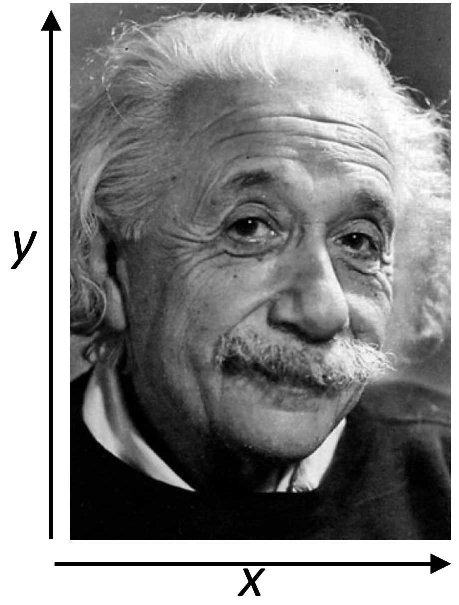
http://debkellylab.org/?page_id=94

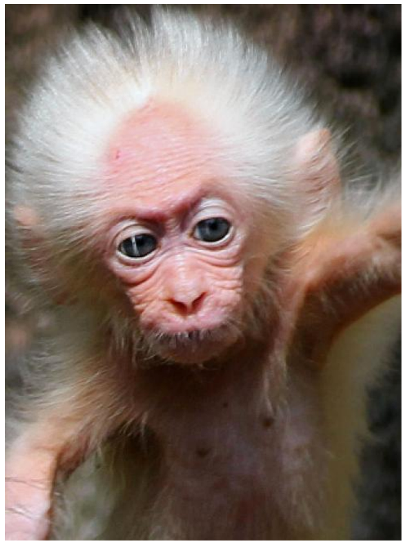
Computation plays an essential role in these imaging-based techniques

- We'll 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* 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





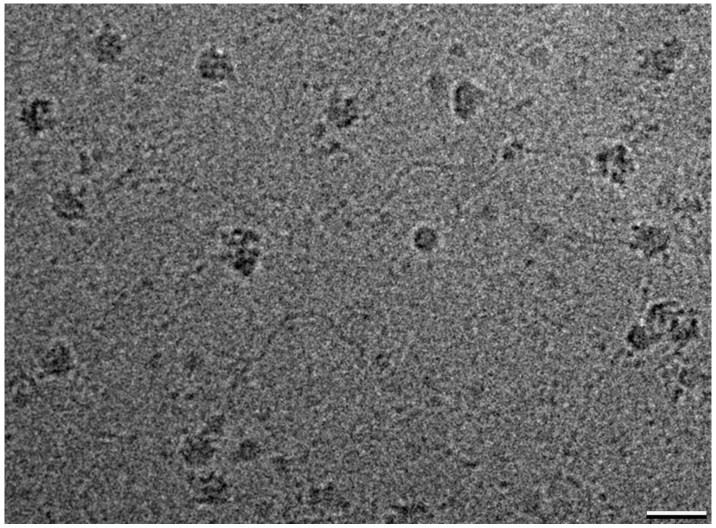
"Whereas Einstein proposed the concept of mass-energy equivalence, Professor Mihail Nazarov, 66, proposed monkey-Einstein equivalence."

From http://www.nydailynews.com/news/world/baby-monkey-albert-einstein-article-1.1192365

Reducing image noise

Experimentally determined images are always corrupted by *noise*

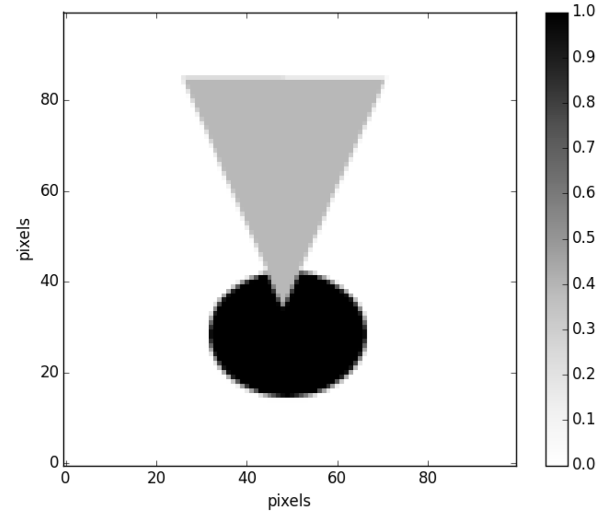
- "Noise" means any deviation from what the image would ideally look like



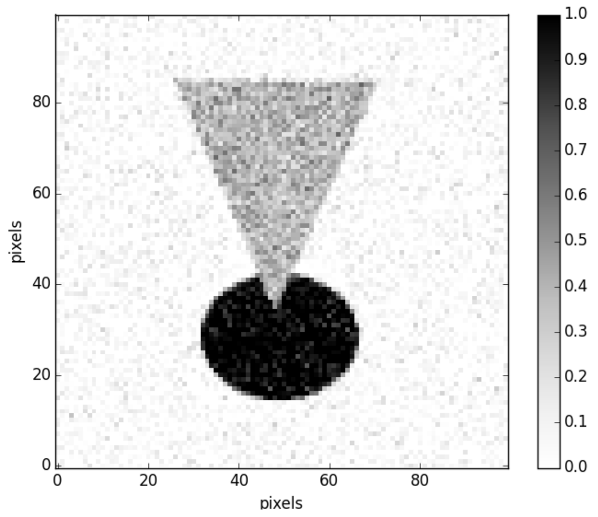
http://www.siox.org/pics/pferd-noisy.jpg

Image noise

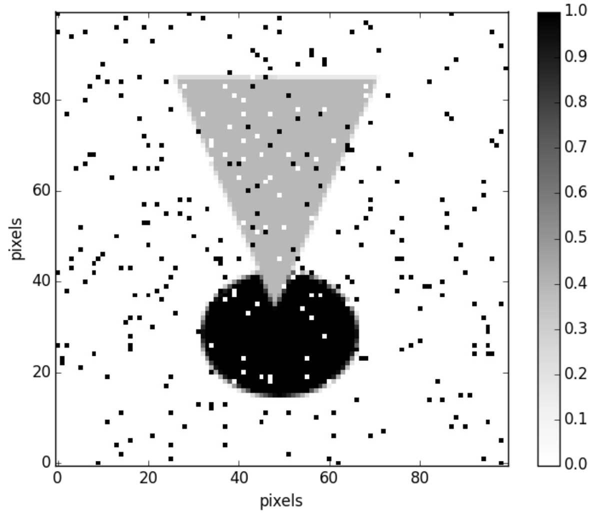
Original image



Noisy images



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



“Salt and peper noise”: random pixels replaced by very bright or dark values

Questions to discuss

Given a noisy image, what can we do computationally to reduce the noise?

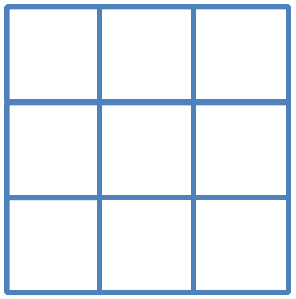
Do we need to make any assumptions about the image to ensure that a noise reduction technique works?

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 the 3x3 square of pixels centered on that pixel



Average the values of these nine pixels to get the new value of the central one

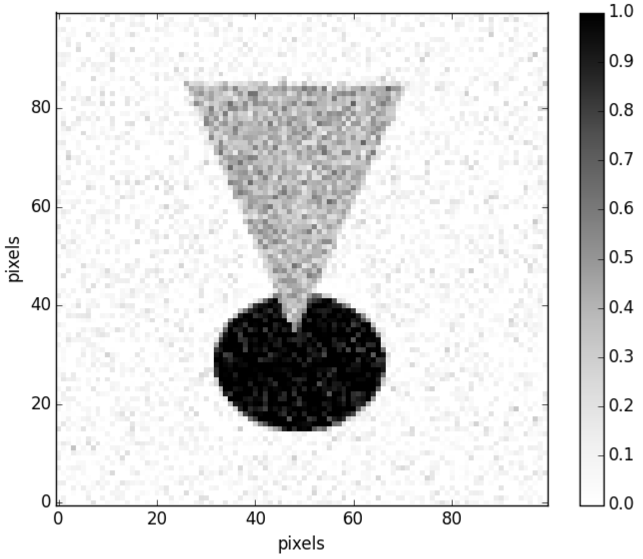
  - Equivalent to convolving the image with a 3x3 matrix:

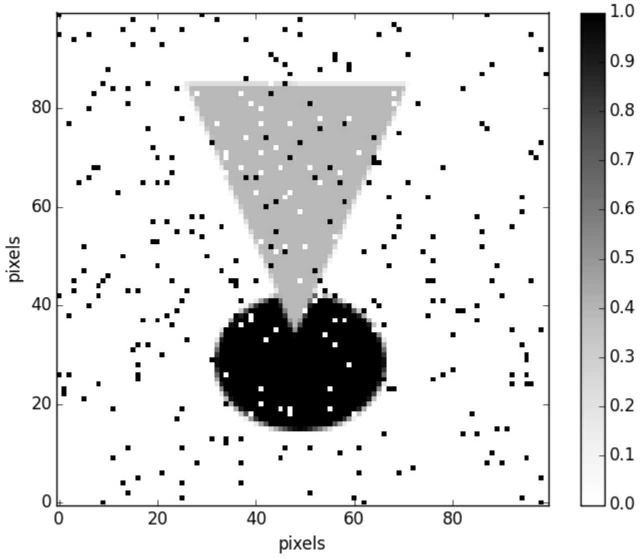
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Values sum to 1 so that average image brightness remains constant

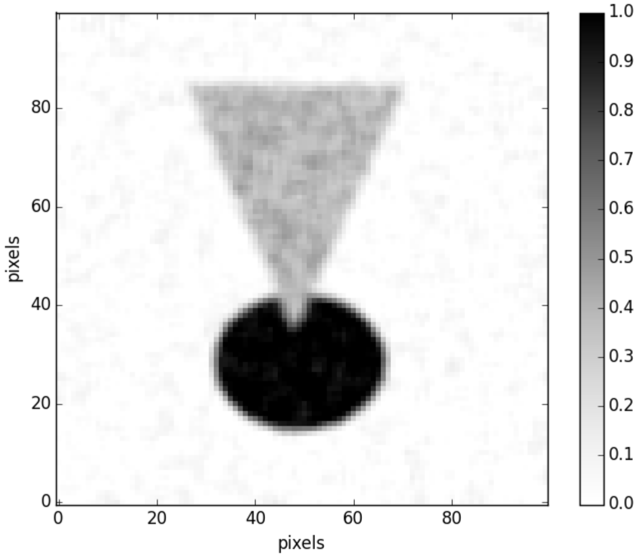
Mean filter

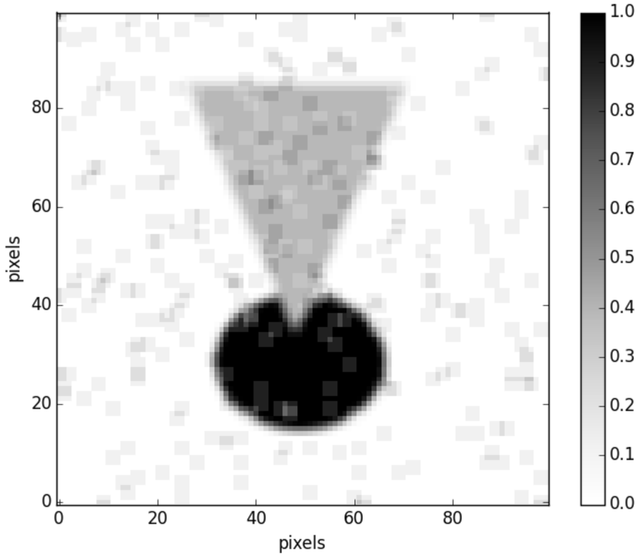
Original images





Result of 3x3 mean filter

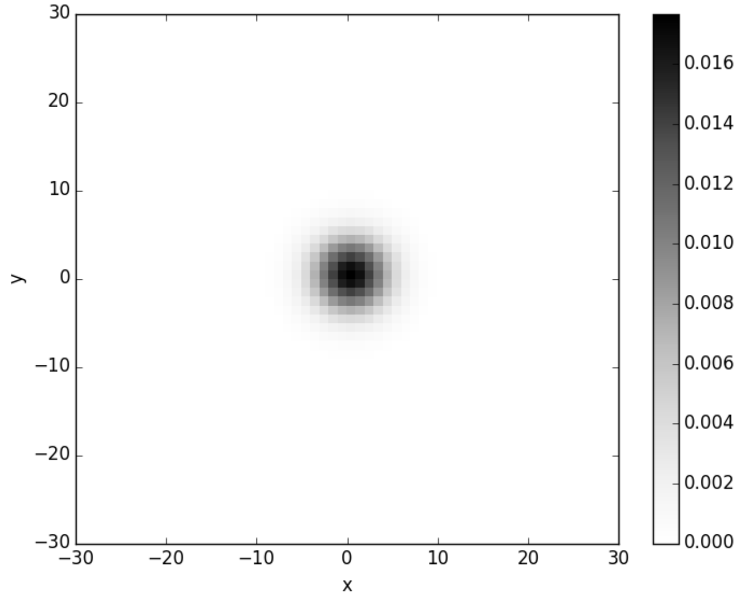




A larger filter (e.g., 5x5, 7x7) would further reduce noise, but would blur the image more

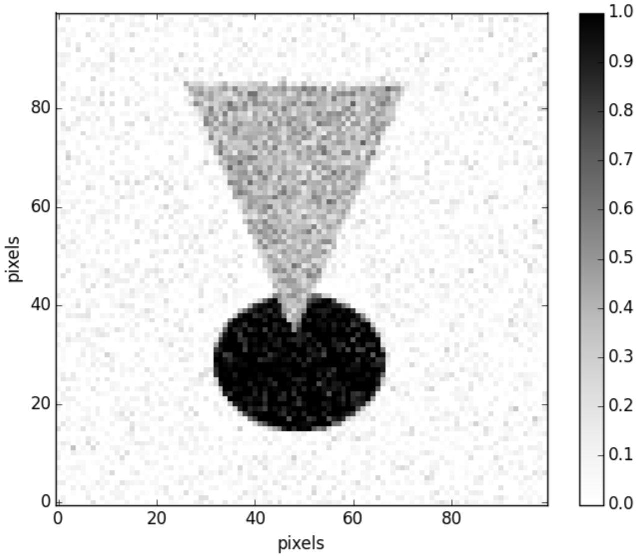
A better choice: use a smoother filter

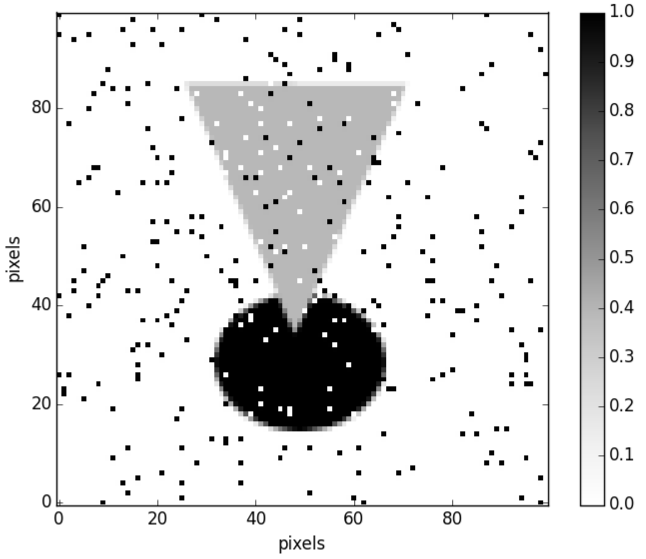
- We can achieve a better trade-off between noise reduction and distortion of the noise-free image by convolving the image with a smoother function
- One common choice is a (two-dimensional) Gaussian



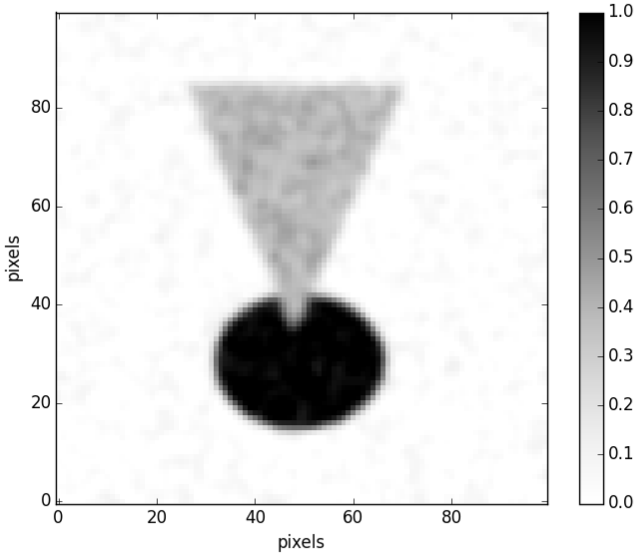
Gaussian filter

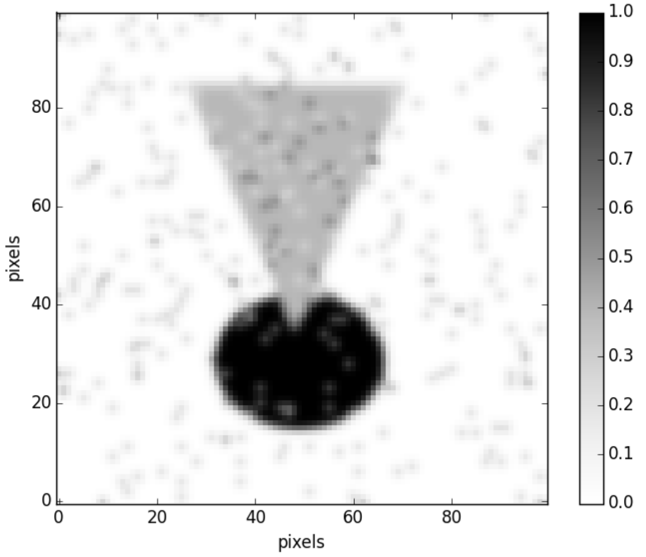
Original images





Result of Gaussian filter (standard deviation σ = 1 pixel)

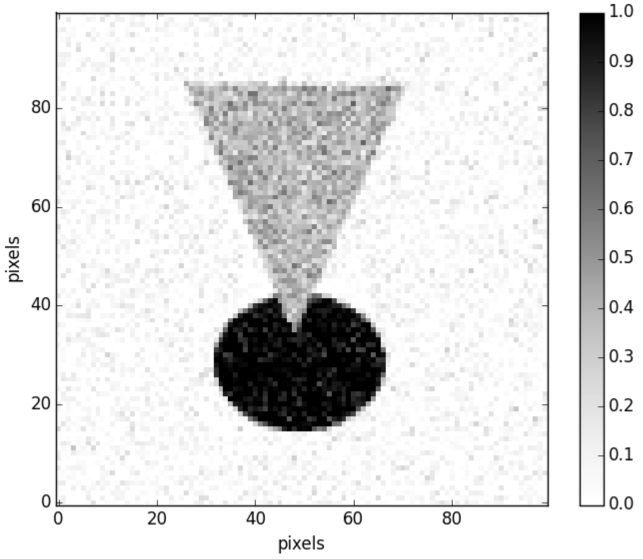


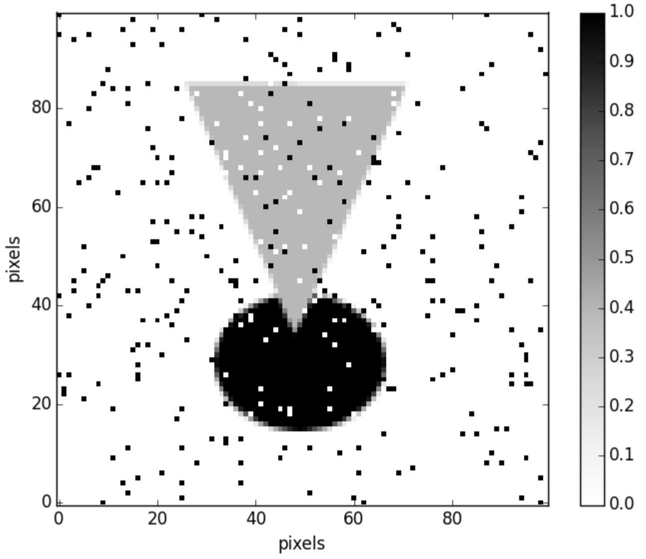


A larger (wider) Gaussian would further reduce noise, but would blur the image more

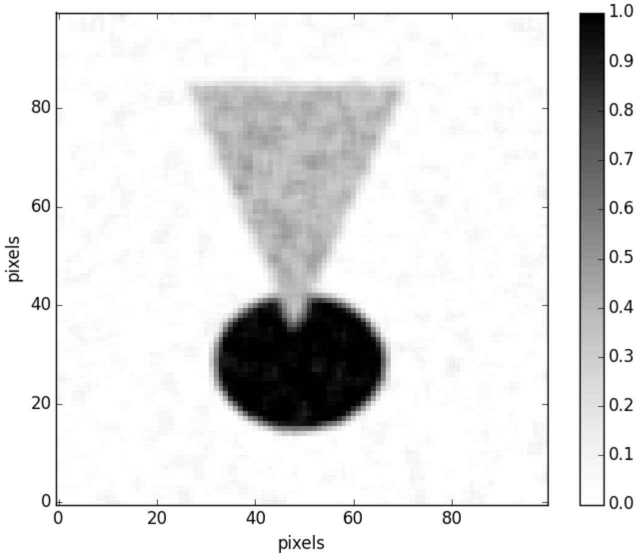
Mean filter (for comparison)

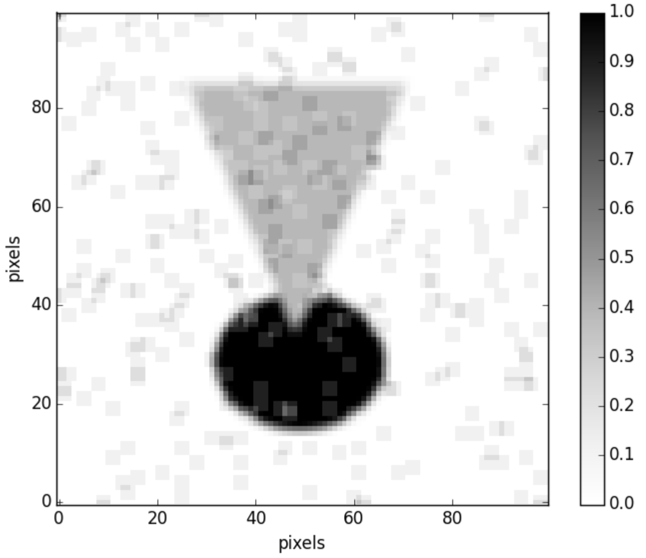
Original images





Filtered images

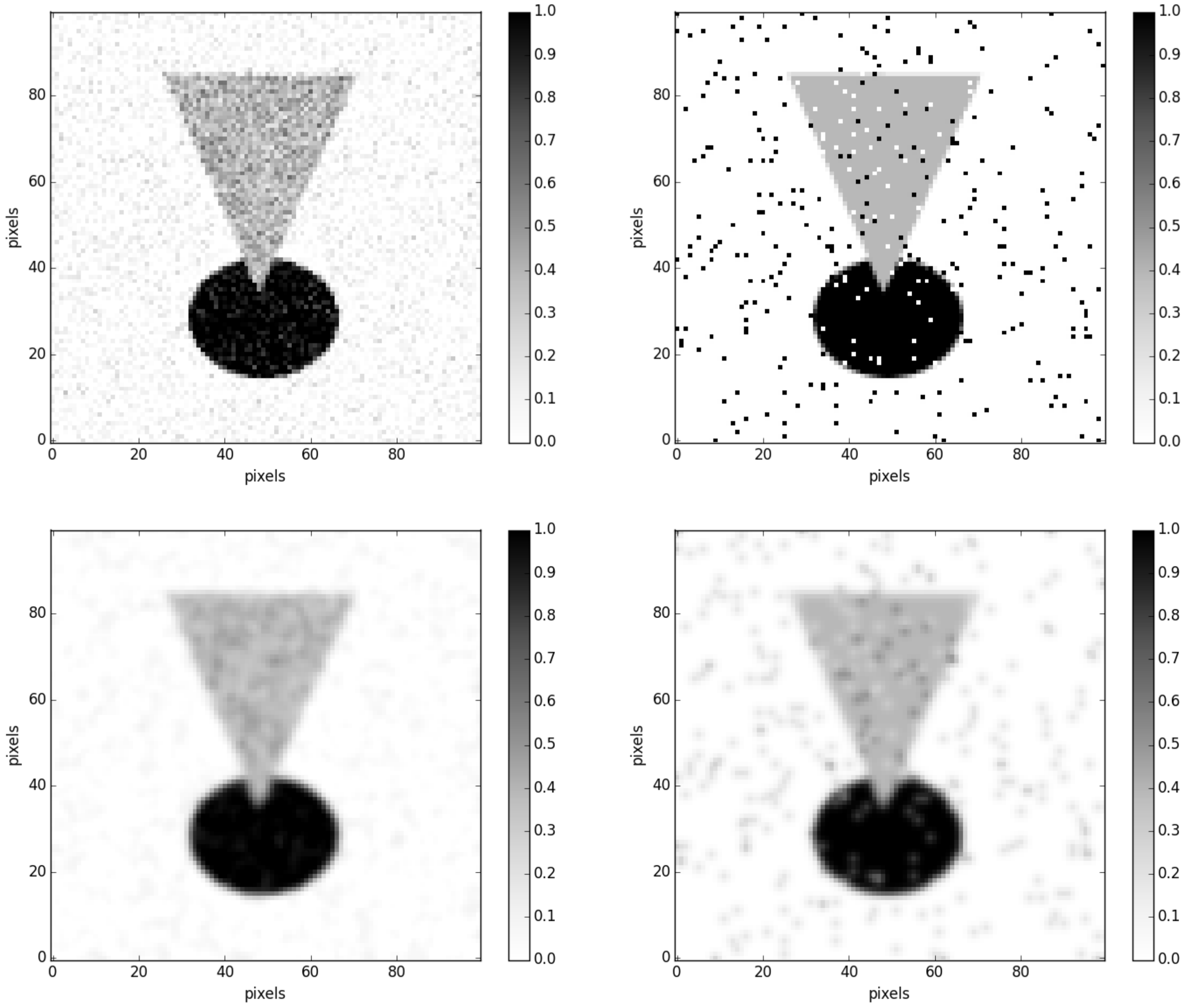




Gaussian filter (for comparison)

Original images

Filtered images



Reducing image noise

Frequency domain interpretation

Low-pass filtering

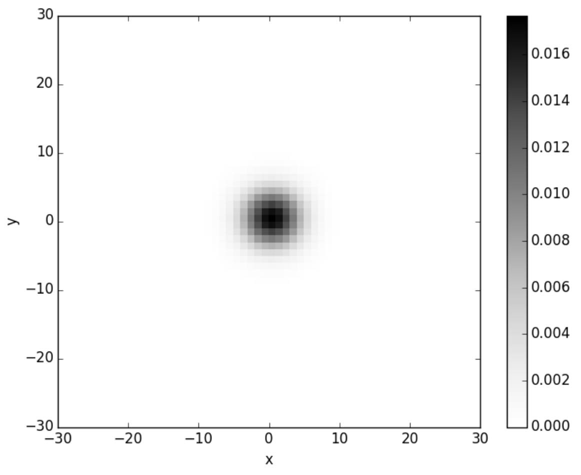
- Because the mean and Gaussian filters are convolutions, we can express them as multiplications in the frequency domain.
- Both types of filters reduce high frequencies while preserving low frequencies. They are thus known as *low-pass filters.*
- These filters work because real images have mostly low-frequency content, while noise tends to have a lot of high-frequency content.

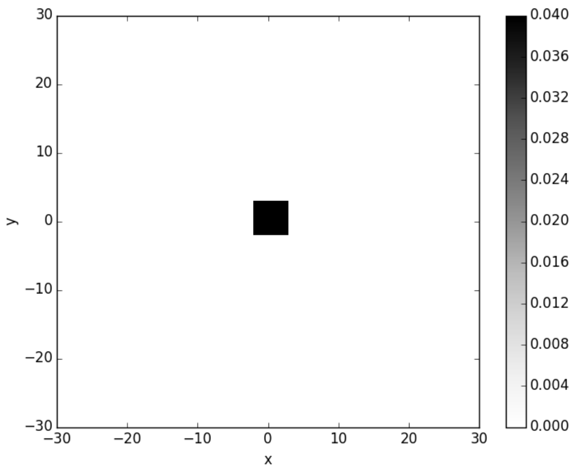
Low-pass filtering

Gaussian filter

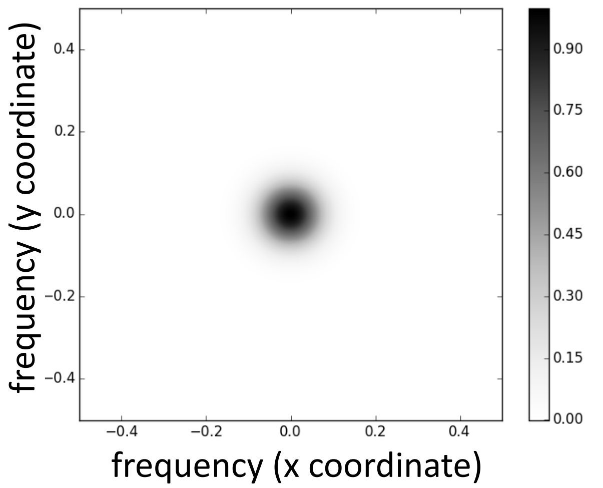
Mean filter

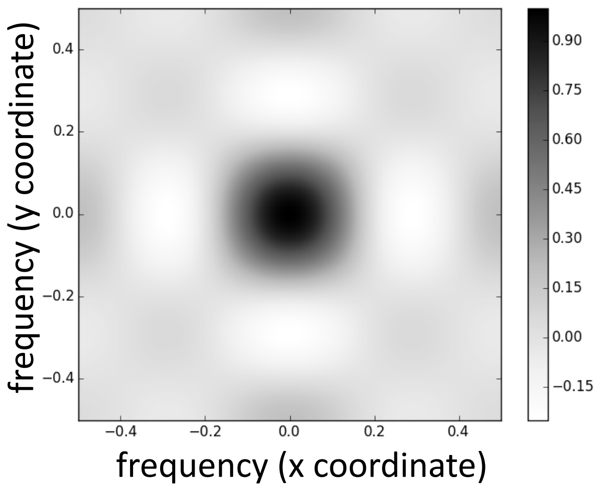
Filter in real domain





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

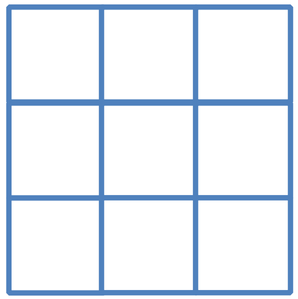
- As a filter becomes larger (wider), its frequency-domain representation becomes narrower
- In other words, making a mean or Gaussian filter larger will make it more low-pass (i.e., narrow the range of frequencies it passes)
  - Thus it will eliminate noise better, but blur the original image more

Reducing image noise

Median filter

Median filter

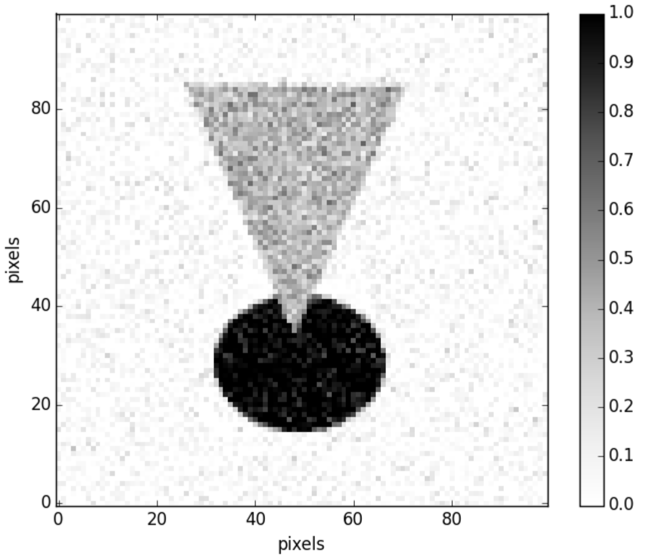
- A median filter ranks the pixel values in a square surrounding the pixel of interest, and picks the middle value

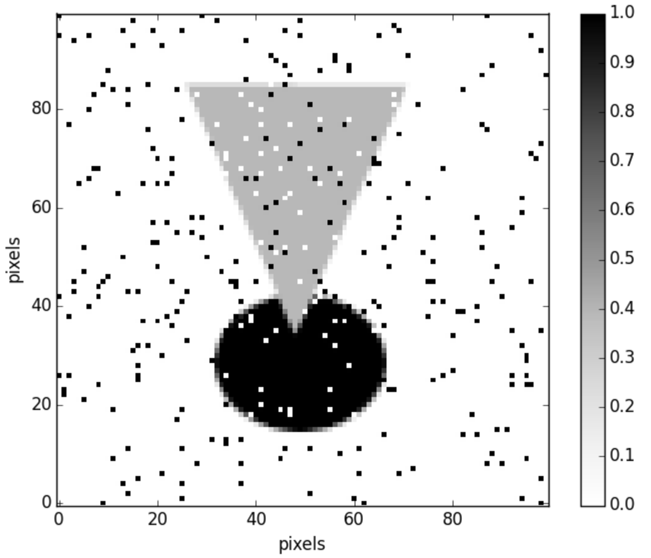


- This is particularly effective at eliminating noise that corrupts only a few pixel values (e.g., salt-and-pepper noise)
- This filter is *not* a convolution

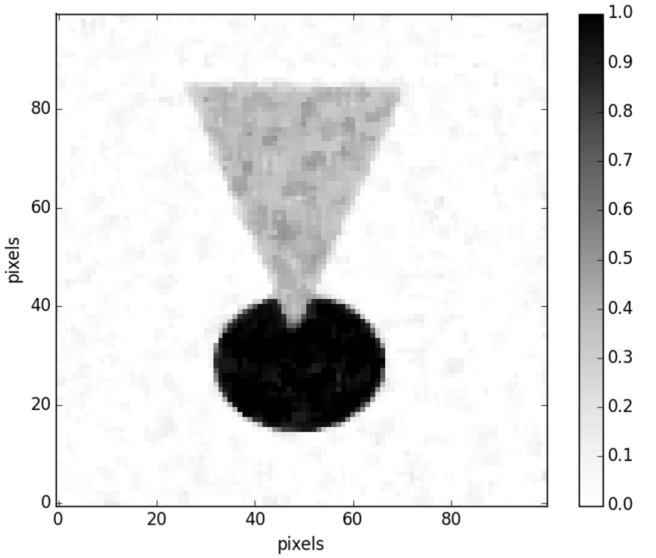
Median filter

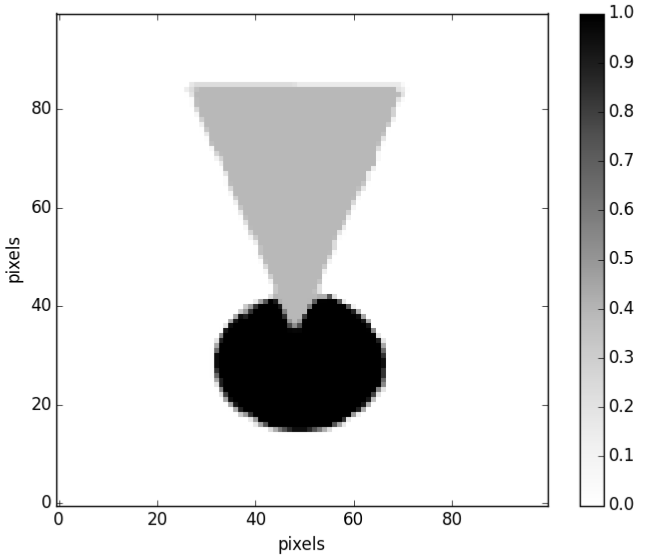
Original images





Result of 3x3 median filter





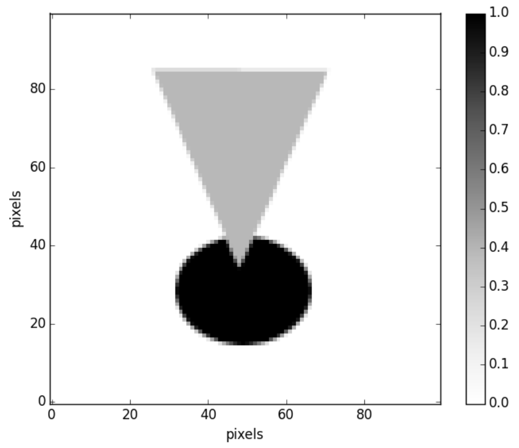
Using a larger window would further reduce noise, but would distort the image more

Sharpening images

High-pass filter

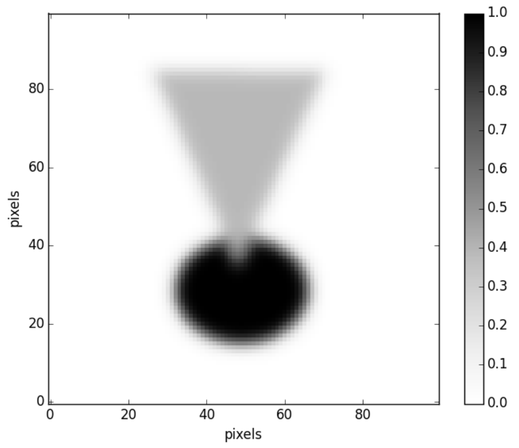
- A *high-pass filter* removes (or reduces) low-frequency components of an image, but not high-frequency ones
- The simplest way to create a high-pass filter is to subtract a low-pass filtered image from the original image
  - This removes the low frequencies but preserves the high ones
  - The filter 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



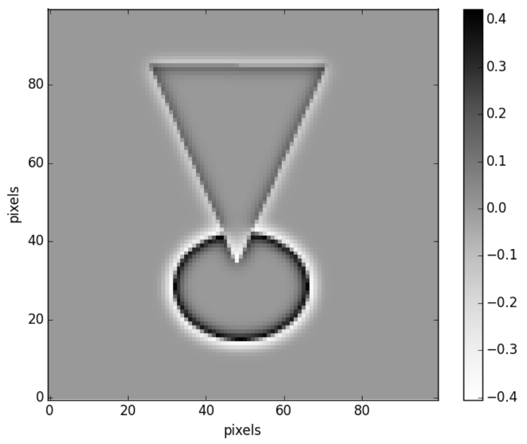
–

Low-pass filtered image



=

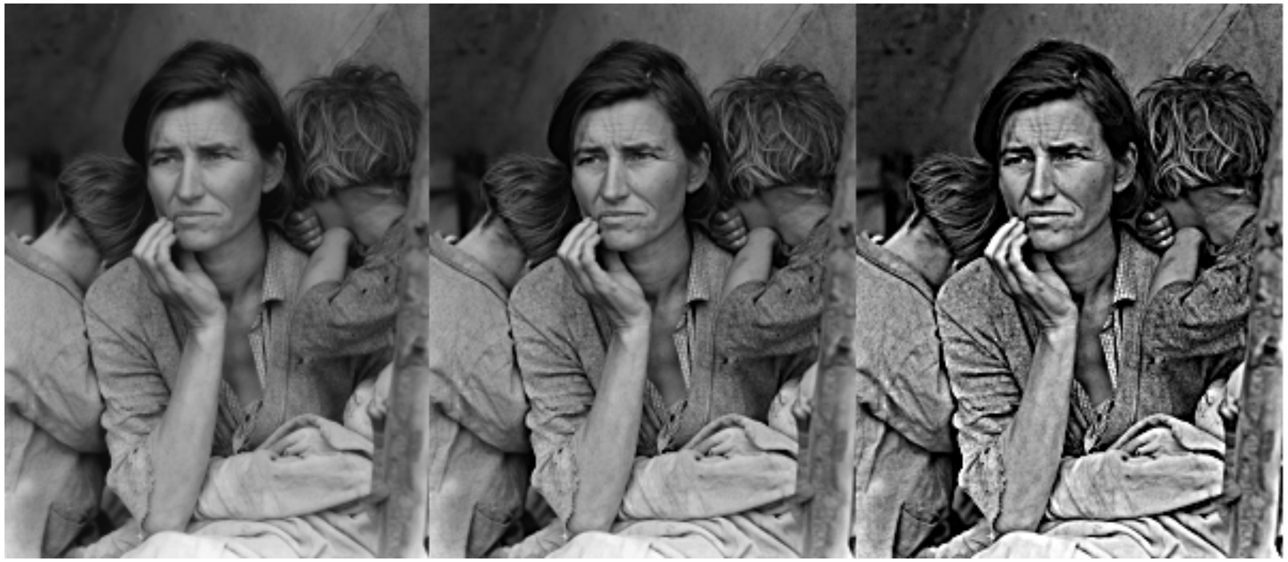
High-pass filtered image



How might one use a high-pass filter?

- To remove any “background” brightness that varies smoothly across the image
- To highlight edges in 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”

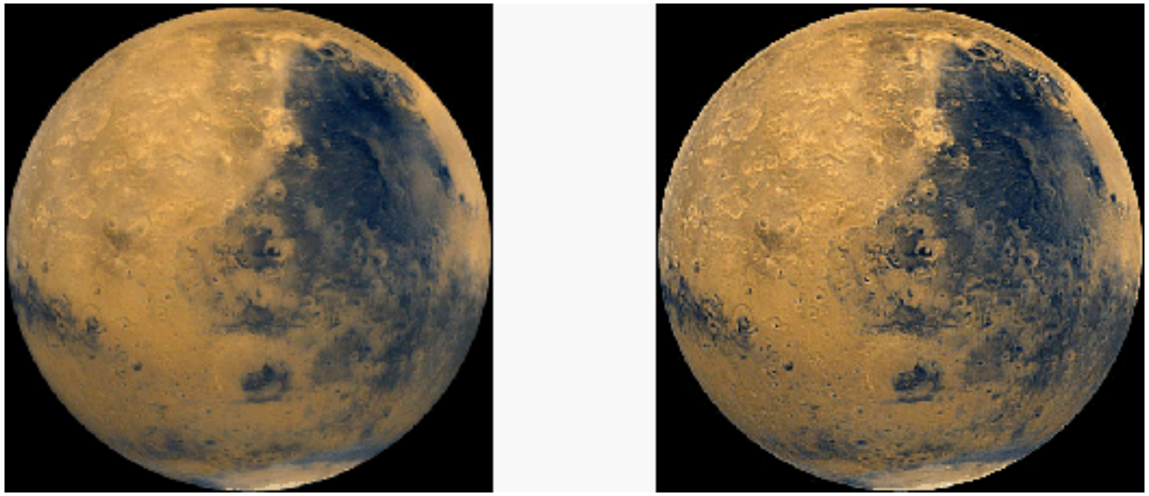
Original image **Mild sharpening (small scale factor)** **Stronger sharpening (larger scale factor)**



https://upload.wikimedia.org/wikipedia/commons/0/0b/Accutance_example.png

Image sharpening — another example

Original image Sharpened image



http://www.exelisvis.com/docs/html/images/unsharp_mask_ex1.gif

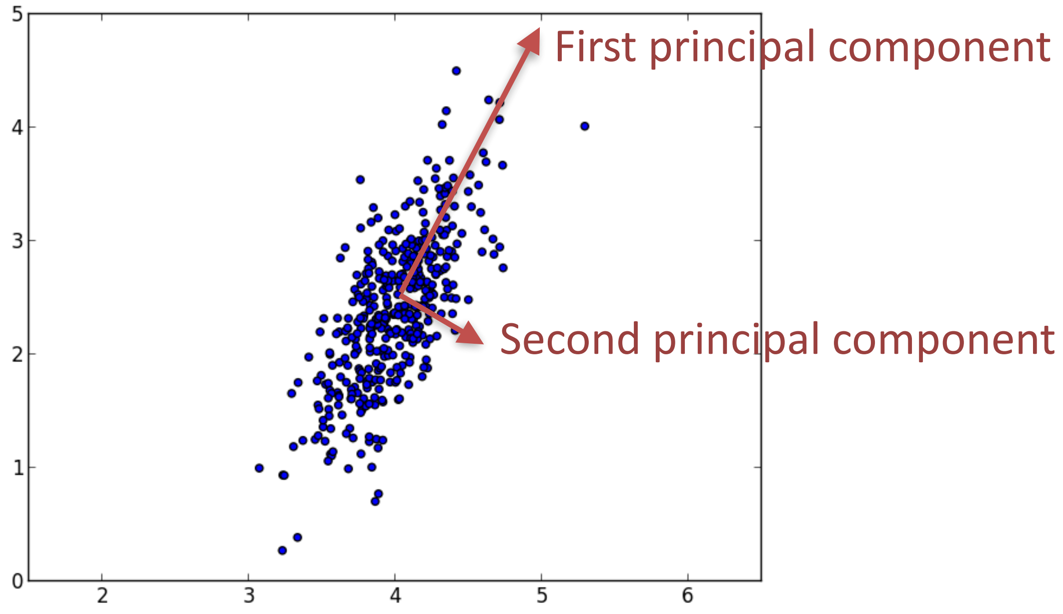
Image description and classification

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 types 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 Component Analysis.

Principal component analysis (PCA)

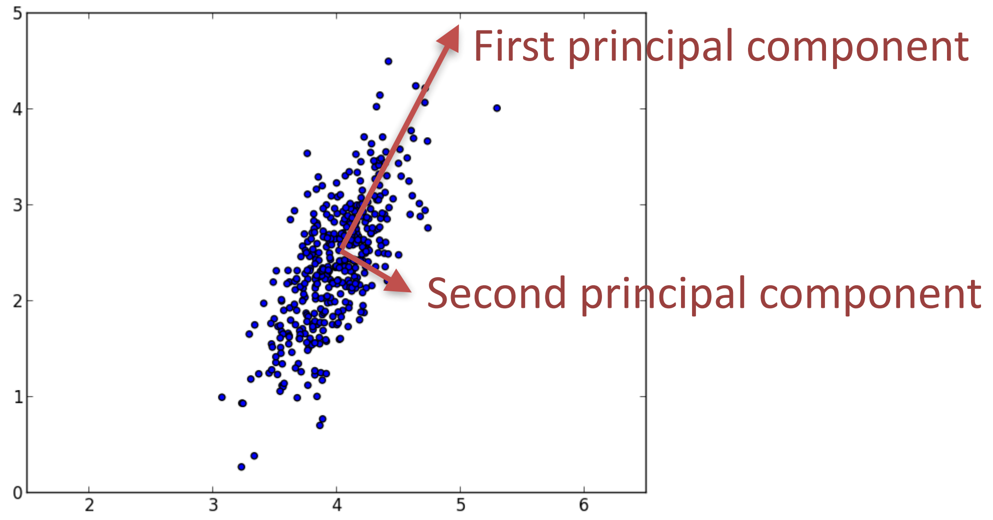
- Basic idea: given a set of points in a multi-dimensional space, we wish to find the linear subspace (line, plane, etc.) that best fits those points.



- If we want to specify a point ***x*** with just one number (instead of two), we can specify the closest point to ***x*** on the line described by the first principal component (i.e., project ***x*** onto that line).
- In a higher-dimensional space, we might specify the point lying closest to ***x*** on a plane specified by the first two principal components.

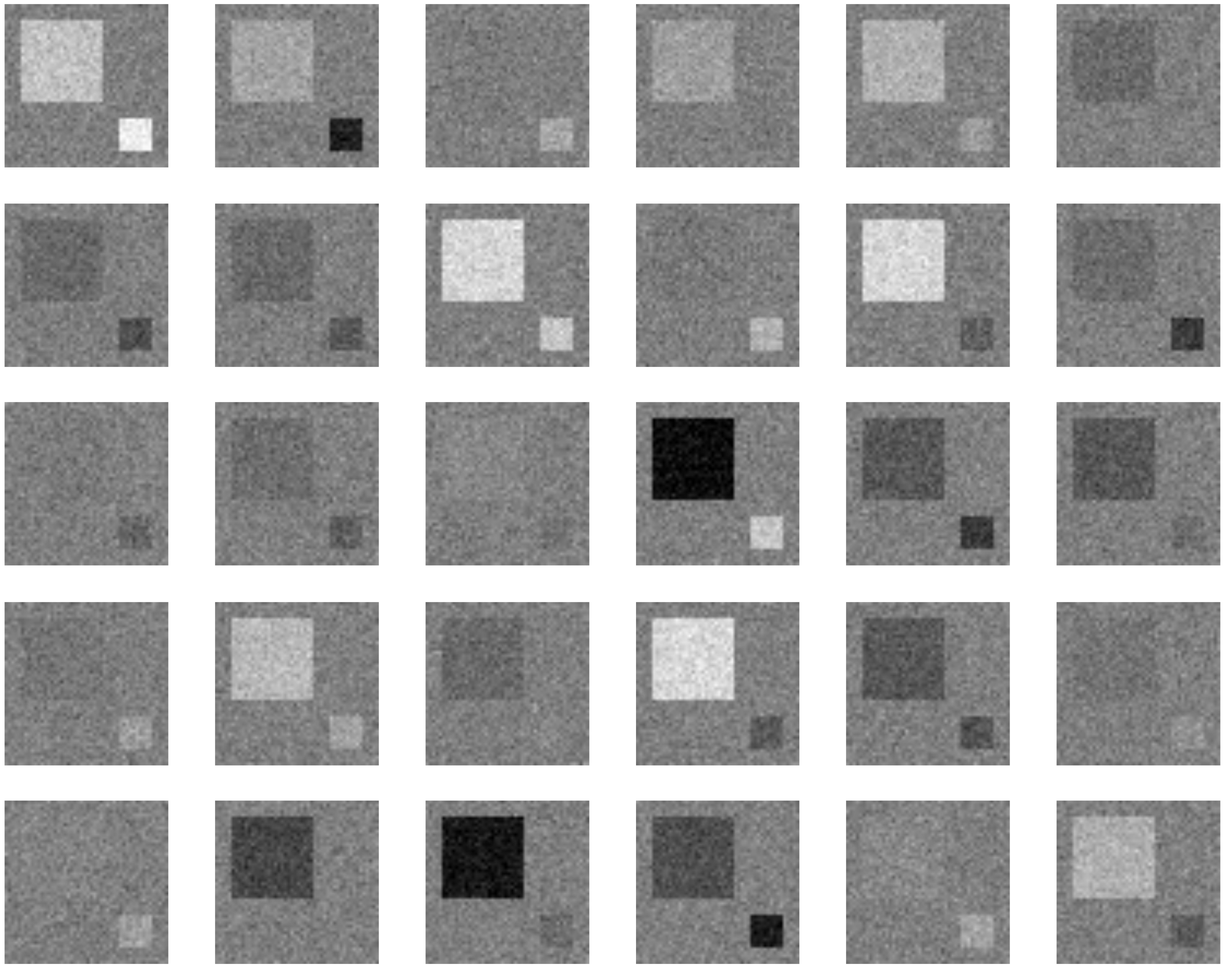
Principal component analysis (PCA)

- 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 specified by that principal component. 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



PCA on images

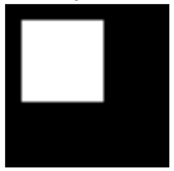
Below are randomly sampled images from a large collection. Given a single image from this collection, can we summarize it with just two numbers?



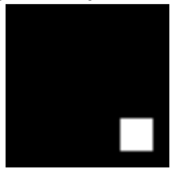
Yes. We can just specify the brightnesses of the two inset squares. These are the coefficients of the first two principal components (k_1 and k_2).

Mean image (M)

1st principal component (PC1)



2nd principal component (PC2)



Each image in the collection can be approximated as $M + k_1$*PC1 + k_2*PC2 for some values of k_1 and k_2

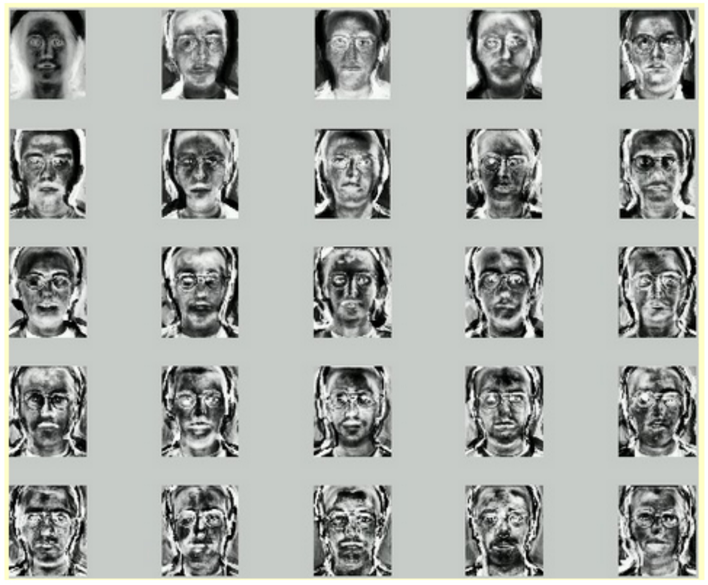
Credit: Joe Paggi

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

Original images

Principal components (viewed as 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

Optional material