CSSE463: Image Recognition Day 5

- Demo code posted
- Lab 2 due Wednesday.
  - Be sure you could perform morphological operations by hand as well
  - Example: compare *dilating twice using a 3x3 square* with dilating *once using a 5x5 square*.
- Fruit Finder due Friday, 11:59 pm.
  - Ask questions as they arise, about technique or about Matlab
- Today: Global vs local operations, filtering

- Questions?

Global vs. local operators

- Global operators
  - Use information from the entire image
  - $p\sim = f(p, p \varepsilon img)$
- Local operators
  - Transform each pixel based on its value or its neighbors' values only
  - $p\sim = f(p, p \varepsilon p_N)$

Enhancement: gray-level mapping

- Maps each pixel value to another value
- Could use a lookup table, e.g., [(0,0), (1, 3), (2, 5), …]
- Could use a function
  - Identity mapping, $y=x$ is straight line
  - Function values above $y=x$ are boosted, those below are suppressed.
  - Gamma function, $y = x^{(1/g)}$ (assuming x in range [0,1]) is a common a control in monitors/TVs.
  - g=2 shown to left
  - Effect?

Gamma mappings, y = x^(1/g)

Original

Dark (g = 0.5)

Light (g = 2)

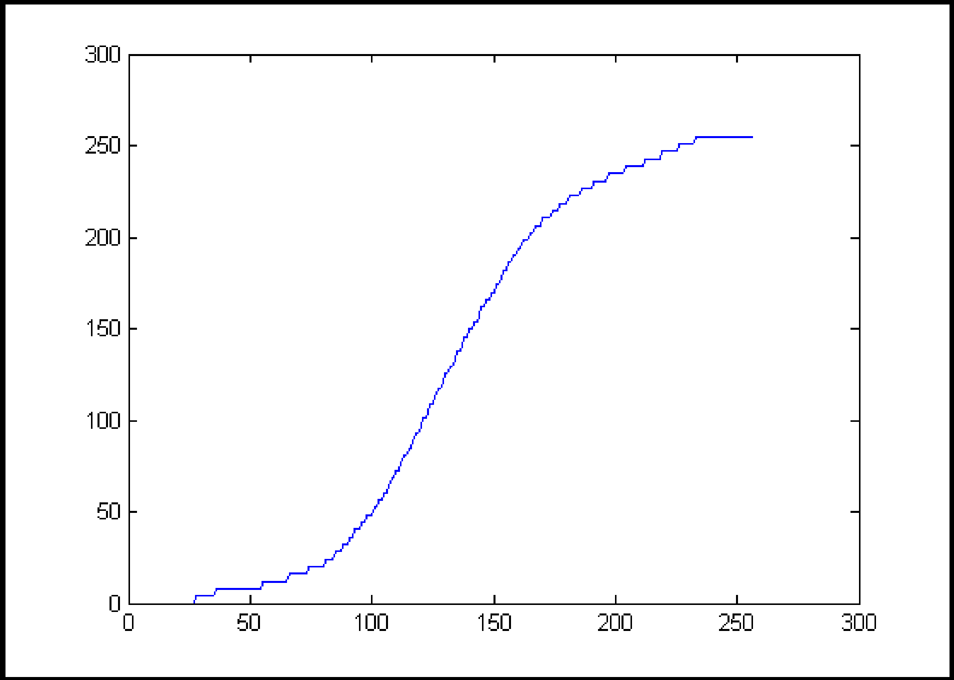
Very light (g = 4)

Histogram Equalization

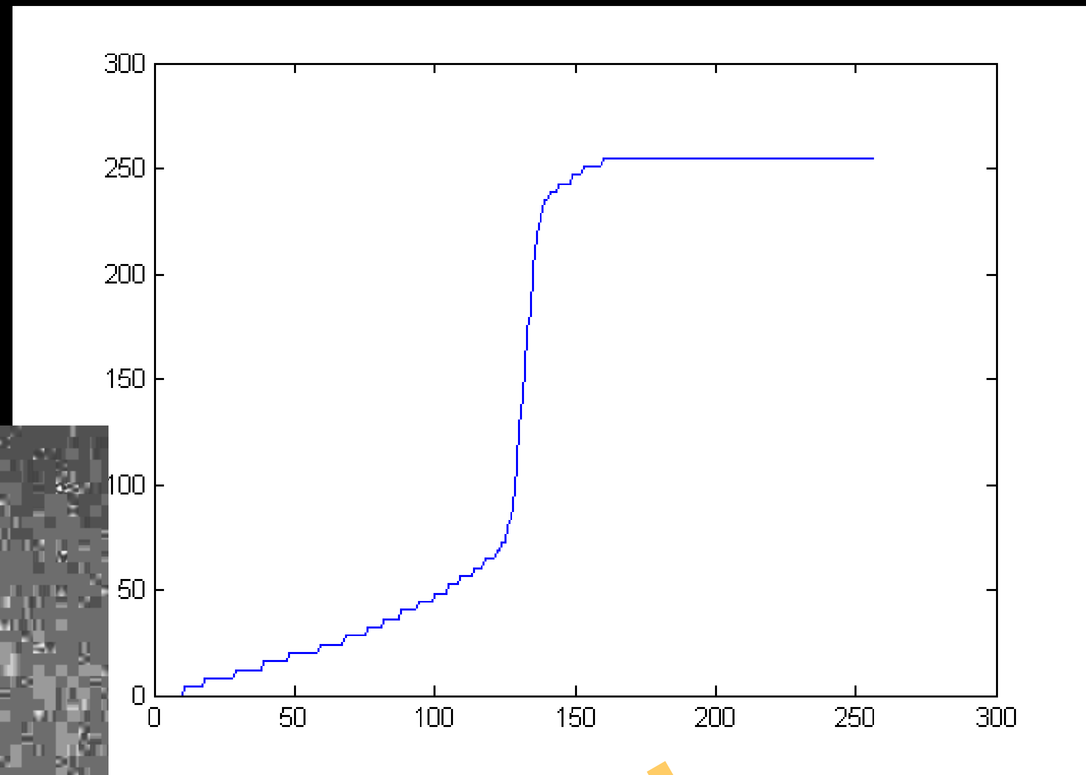
- Creates a mapping that flattens the histogram.
  - Uses full range [0, 255]
    - Good: “automatically” enhances contrast where needed.
  - Approx same level of pixels of each gray level
    - Unpredictable results.
  - Maintains the histogram’s shape, but changes the density of the histogram

- Good example of a *global* operation
- Next: pros and cons

HistEq on Sunset



HistEq on Matt



Whoops!

But where's the color?

- Can we use gray-level mapping on color images?
- Discuss how

Local operators

- The most common local operators are **filters**.
  - Today: for smoothing
  - Tomorrow: for edge detection

Image smoothing

- Gaussian distributions are often used to model noise in the image
  - $g = g_r + N(0, \sigma)$
    - g = sensed gray value
    - g_r = "expected" real grayvalue
    - $N(0, \sigma)$ is a Gaussian (aka, **N**ormal, or bell curve) with mean = 0, std. dev = σ.
    - Lots of Gaussian distributions in this course…
- Answer: average it out! 3 methods
  - Box filter
  - Gaussian filter
  - Median filter
- Filter

Box filters

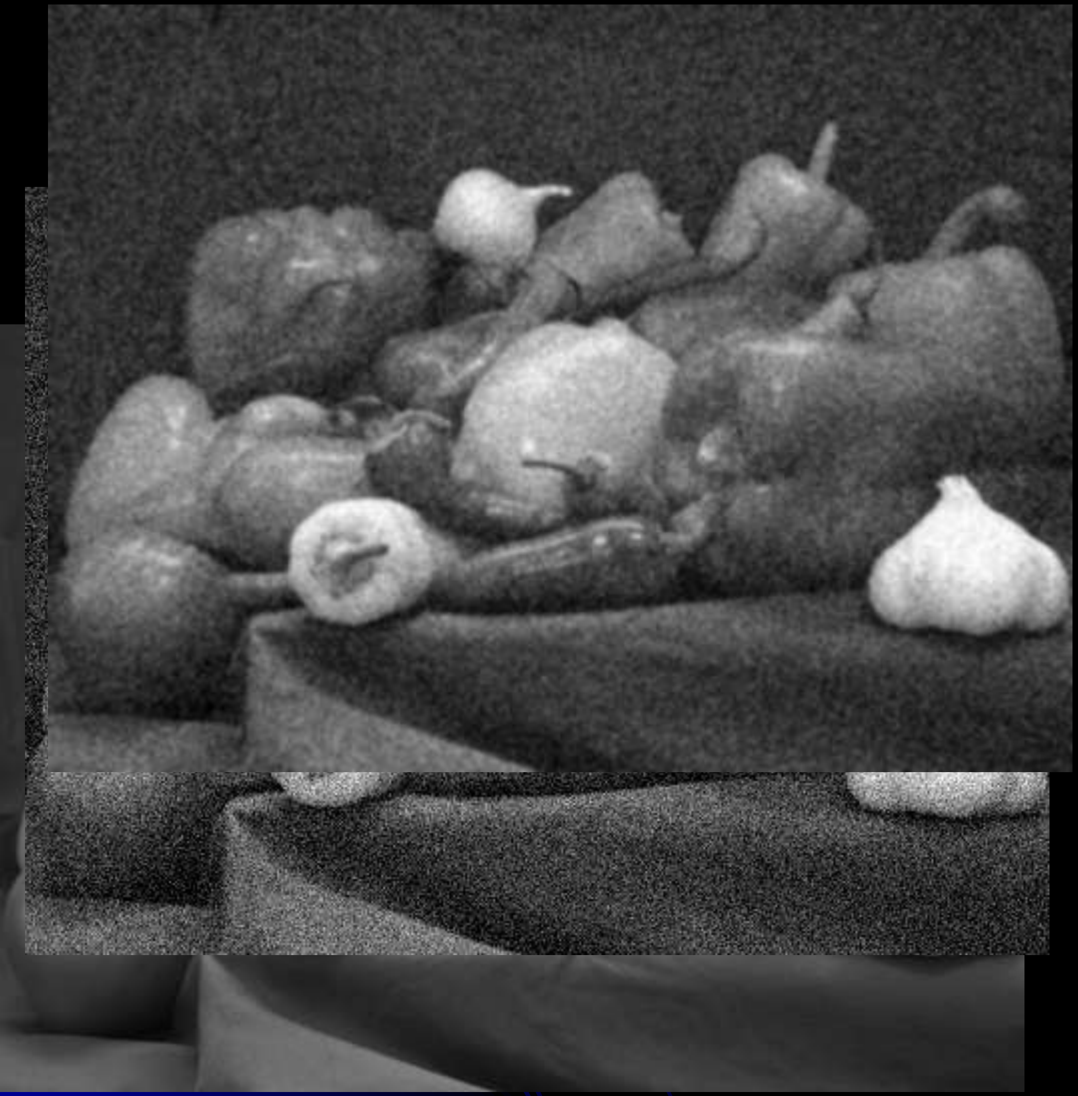
- Simplest.
- Improves homogeneous regions.
- Unweighted average of the pixels in a small neighborhood.
- For 5x5 neighborhood,

$$J(r,c) = \frac{1}{25}\sum_{i=-2}^{2}\sum_{j=-2}^{2} I(r+i,c+j)$$

I = orig image, J=filtered image

See why this is a “local operation?”

Gaussian filters



- Nicest theoretical properties.
- Average weighted by distance from center pixel. Weight of pixel (i,j):

$$W(i,j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}}$$

- Then use weight in box filter formula
- In practice, we use a discrete approximation to W(i,j)

Median filters

- Step edge demo
  - smoothGaussDemo
- Salt demo
  - smoothSaltDemo

- Averaging filters have two problems.
  - They blur edges.
  - They don't do well with "salt-and-pepper" noise:
    - Faulty CCD elements
    - Dust on lens
- Median filter: Replace each pixel with the median of the pixels in its neighborhood
  - More expensive
  - Harder to do with hardware
- But can be made somewhat efficient
  - (Sonka, p 129)
- Hybrid: sigma filtering

Q4,5

Discrete filters

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

- Discrete 3x3 box filter:
- To get the output at a single point, take cross-correlation (basically a dot-product) of filter and image at that point
- To filter the whole image, shift the filter over each pixel in the original image