Matlab Tutorial Continued

- Files, functions and images.

Announcements

- Week of Feb. 17th Jacobs office hours change.
  - Tuesday, 18th 3-4.
  - Friday, 21st 3:30-4:30
- TA office hours still Monday 17th 4-6.

Files

Matlab

Functions

- Format: function o = test(x,y)
- Name function and file the same.
- Only first function in file is visible outside the file.
- *Look at sample function*

Images

- Black and white image is a 2D matrix.
- Intensities represented as pixels.
- Color images are 3D matrix, RBG.
- *Matlab*

Debugging

- Add print statements to function by leaving off ;
- keyboard
- debug and breakpoint

Conclusions

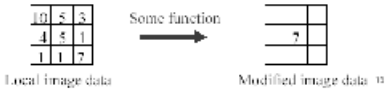
- Quick tour of matlab, you should teach yourself the rest. We'll give hints in problem sets.
- Linear algebra allows geometric manipulation of points.
- Learn to love SVD.

Linear Filtering

- About modifying pixels based on neighborhood. Local methods simplest.
- Linear means linear combination of neighbors. Linear methods simplest.
- Useful to:
  - Integrate information over constant regions.
  - Scale.
  - Detect changes.
- Fourier analysis.
- Many nice slides taken from Bill Freeman.

What is image filtering?

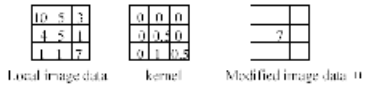
- Modify the pixels in an image based on some function of a local neighborhood of the pixels.



(Freeman)

Linear functions

- Simplest: linear filtering.
  - Replace each pixel by a linear combination of its neighbors.
- The prescription for the linear combination is called the "convolution kernel".



(Freeman)

Convolution

- Convolution kernel g, represented as matrix.
  - it's associative
- Result is:

$$f[m,n] = I \otimes g = \sum_{k,l} I[m \quad k, n \quad l]g[k,l]$$

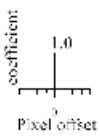
Linear filtering (warm-up slide)

original

?

Linear filtering (warm-up slide)

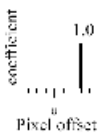
original



Filtered
(no change)

Linear filtering

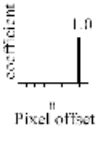
original

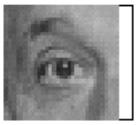


?

shift

original

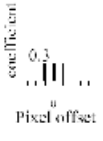




shifted

Linear filtering

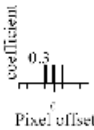
original

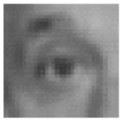


?

Blurring

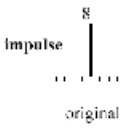
original



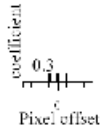


Blurred (filter applied in both dimensions).

Blur examples

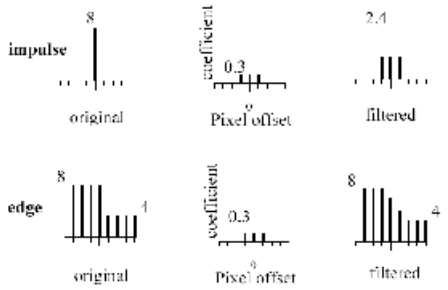


original

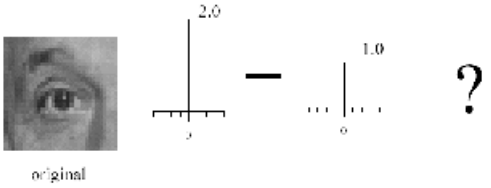


filtered

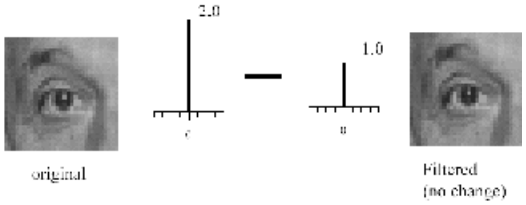
Blur examples



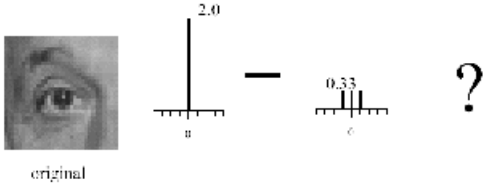
Linear filtering (warm-up slide)



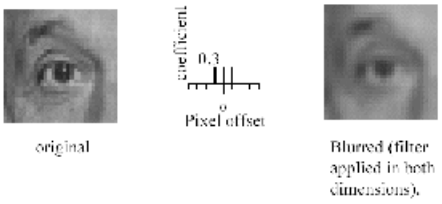
Linear filtering (no change)



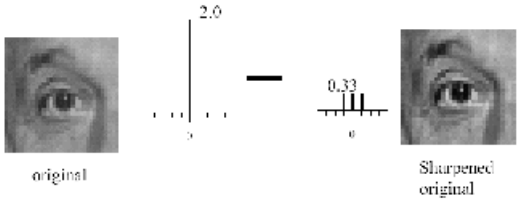
Linear filtering



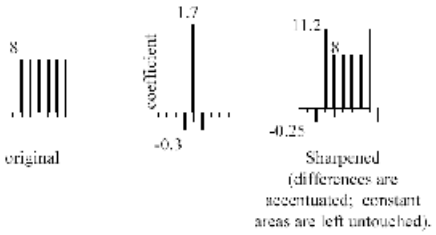
(remember blurring)



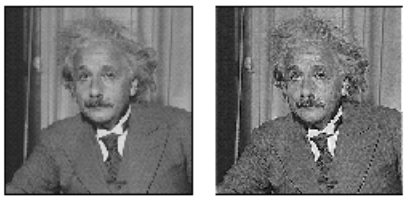
Sharpening



Sharpening example



Sharpening



before after

Filtering to reduce noise

- Noise is what we're not interested in.
  - We'll discuss simple, low-level noise today: Light fluctuations; Sensor noise; Quantization effects; Finite precision
  - Not complex: shadows; extraneous objects.
- A pixel's neighborhood contains information about its intensity.
- Averaging noise reduces its effect.

Additive noise

- I = S + N. Noise doesn't depend on signal.
- We'll consider:

$I_i = s_i + n_i$ with $E(n_i) = 0$

s_i deterministic.

n_i, n_j independent for $n_i \neq n_j$

n_i, n_j identically distributed

Average Filter

- Mask with positive entries, that sum 1.
- Replaces each pixel with an average of its neighborhood.
- If all weights are equal, it is called a BOX filter.

F

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

(Camps)

Does it reduce noise?

- Intuitively, takes out small variations.

$I(i,j) = \hat{I}(i,j) + N(i,j)$ with N(i, j) ~ N(0, σ)

$$O(i,j) = \frac{1}{m^2}\sum_{h=-m/2}^{m/2}\sum_{k=-m/2}^{m/2}\hat{I}(i-h, j-k) + N(i-h, j-k) =$$

$$= \frac{1}{m^2}\sum_{h=-m/2}^{m/2}\sum_{k=-m/2}^{m/2}\hat{I}(i-h, j-k) + \underbrace{\frac{1}{m^2}\sum_{h=-m/2}^{m/2}\sum_{k=-m/2}^{m/2}N(i-h, j-k)}_{\tilde{N}(i,j)}$$

$E(\hat{N}(i,j)) = 0$

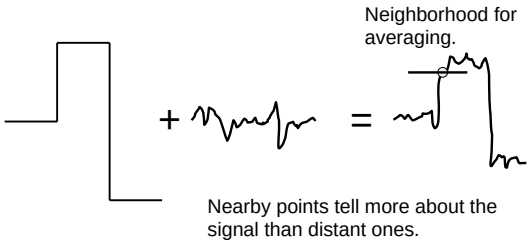
$E(\hat{N}^2(i,j)) = \frac{1}{m^2} m\sigma^2 = \frac{\sigma^2}{m} \Rightarrow \hat{N}(i,j) \sim N(0, \frac{\sigma}{\sqrt{m}})$

(Camps)

Matlab Demo of Averaging

Example: Smoothing by Averaging

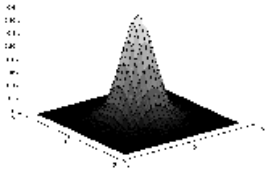
Smoothing as Inference About the Signal



Nearby points tell more about the signal than distant ones.

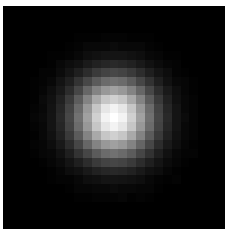
Gaussian Averaging

- Rotationally symmetric.
- Weights nearby pixels more than distant ones.
  - This makes sense as probabalistic inference.



- A Gaussian gives a good model of a fuzzy blob

An Isotropic Gaussian

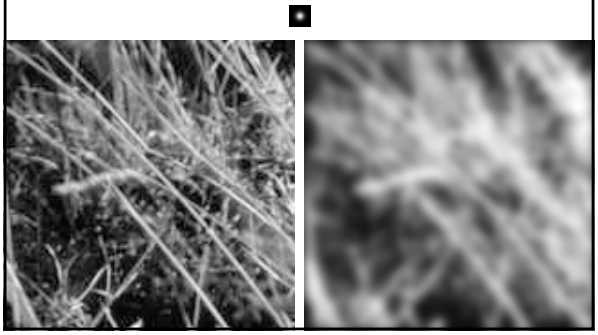


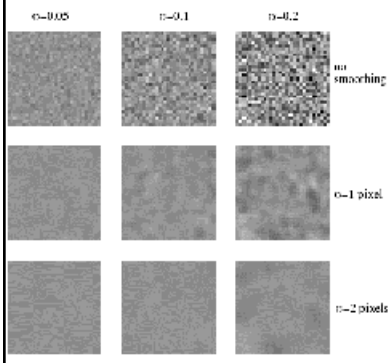
- The picture shows a smoothing kernel proportional to

$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

(which is a reasonable model of a circularly symmetric fuzzy blob)

Smoothing with a Gaussian



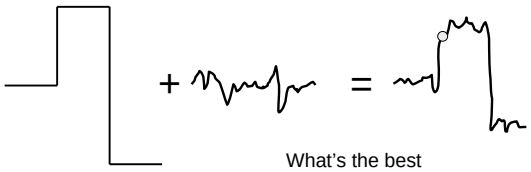


The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realizations of an image of gaussian noise.

Efficient Implementation

- Both, the BOX filter and the Gaussian filter are separable:
  - First convolve each row with a 1D filter
  - Then convolve each column with a 1D filter.

Smoothing as Inference About the Signal: Non-linear Filters.



What's the best neighborhood for inference?

Filtering to reduce noise: Lessons

- Noise reduction is probabilistic inference.
- Depends on knowledge of signal and noise.
- In practice, simplicity and efficiency important.

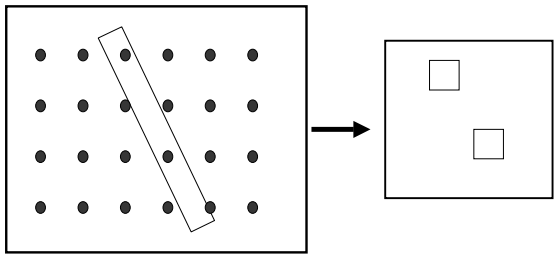
Filtering and Signal

- Smoothing also smooths signal.
- *Matlab*
- Removes detail
- *Matlab*
- This is good and bad:

  - Bad: can't remove noise w/out blurring shape.

  - Good: captures large scale structure; allows subsampling.

Subsampling



Matlab

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.