Introduction to Computer Vision

Michael J. Black

Sept 2008

Lecture 5: Linear filtering

Info

- Matlab tutorial yesterday
  - http://www.cs.brown.edu/courses/cs143/MatlabTutorialCode.html
  - Do we need another?

- Problems 1&2 in Asgn1 due Friday at class time.

- Are you on the cs143list?

- Check web regularly

Goals

- Linear filtering
  - Foundations for asng1.
    - Problem 1

- Monday: image derivatives
  - Problem 2
- Wednesday: correlation, features
  - Problems 3&4

Homework

- Assignment 0 due
- Assignment 1 out
  - Grad credit – do extra credit questions.
  - Problems 1&2 due Friday Sept 19 (1 week)
  - Problems 3&4 due the week after

Office/TA hours

Michael's office hours (CIT 521)

Wednesday/Thursday 3:00-4:00

TA Hours (CIT 271):

Deqing: **Mondays from 7pm to 9pm**

Teodor: **Tuesdays from 5pm - 7pm.**

Upcoming talk

Gerard Medioni

University of Southern California
Monday, September 15, 2008 at 3pm
Room 368 (CIT 3rd floor)
Refreshments will be served at 2:45 pm

Tensor Voting in 2 to N dimensions: Fundamental Elements and a Few Applications

Ponce and Forsyth

http://decsai.ugr.es/mia/complementario/t1/book3chaps.html

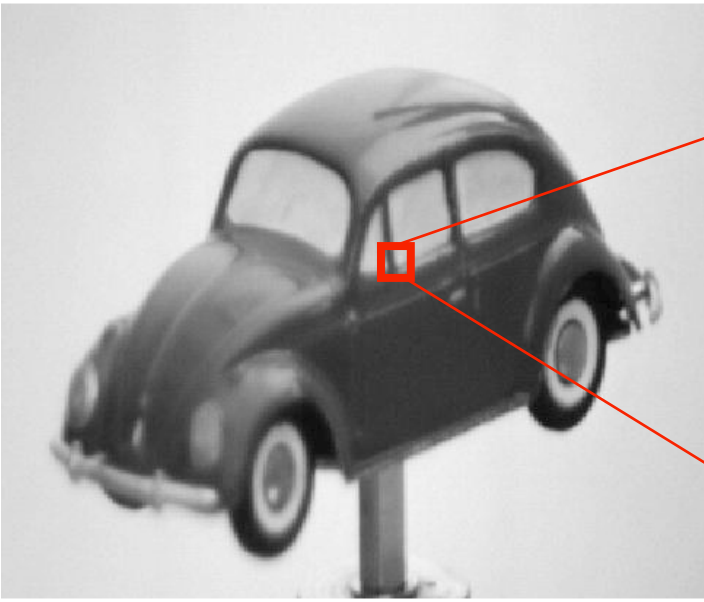

```
im = double(imread('/course/cs143/asgn/asgn0/flintstones.tif'));
% horzontally flipped
im1 = im(:, end:-1:1);
figure;
imshow(uint8(im1));

% log(im+1)
im2 = log(im+1);

% Scale so that the maximum is 255
im2 = 255*(im2-min(im2(:)))/max(im2(:)-min(im2(:)));
fprintf('the mean is %3.3f\n’, mean(im2(:)));
imshow(uint8(im2));

% Negative image
im3 = 255 - im1;
```

From images to understanding?



Huge array of numbers

64	60	69	100	149	151	176	182	179
65	62	68	97	145	148	175	183	181
65	66	70	95	142	146	176	185	184
66	66	68	90	135	140	172	184	184
66	64	64	84	129	134	168	181	182
59	63	62	88	130	128	166	185	180
60	62	60	85	127	125	163	183	178
62	62	58	81	122	120	160	181	176
63	64	58	78	118	117	159	180	176

Classifier?

CAR

Infeasible.
Reduce dimensionality.
Invariance to lighting, rotation, ….
Need to extract some salient structure - ***features***

Image Filtering

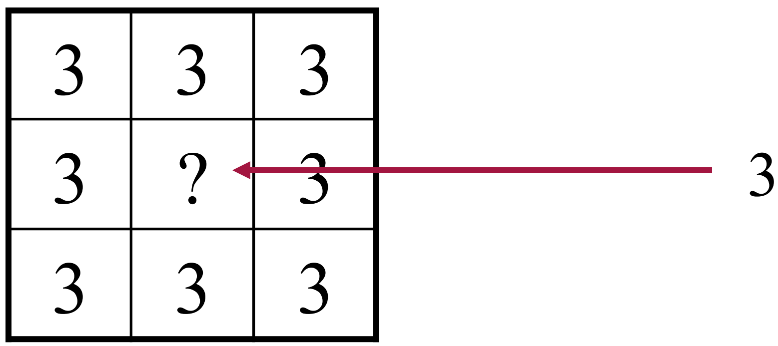
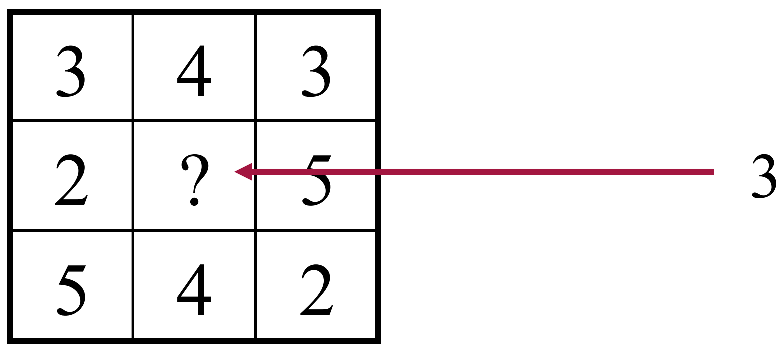


Image Filtering



What assumptions are you making to infer the center value?

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".

10	5	3
4	5	1
1	1	7

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	7	

Modified image data

Freeman

Linear Filtering

- Linear means that the response of the filter at a pixel is a linear combination of other pixels.
  - Typically using a local neighborhood.
  - Linear methods simplest.
- Useful to:
  - Integrate information over constant regions.
  - Modify images (e.g. smooth or enhance)
  - Scale.
  - Detect features.

2-D signals and convolutions

- Continuous $I(x,y)$
- Discrete $I[k,l]$ or $I_{k,l}$

- 2-D convolution (discrete)

$$f[m,n] = I \otimes g = \sum_{k=1}^{K} \sum_{l=1}^{L} I[m-k+\lfloor K/2 \rfloor, n-l+\lfloor L/2 \rfloor] g[k,l]$$

"filtered" image

filter "kernel"

2-D signals and correlation

- Continuous $I(x,y)$
- Discrete $I[k,l]$ or $I_{k,l}$

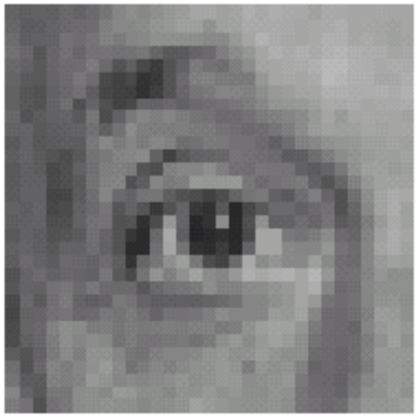
- 2-D correlation (discrete)

$$f[m,n] = I \otimes g = \sum_{k=1}^{K} \sum_{l=1}^{L} I[m+k-\lfloor K/2 \rfloor, n+l-\lfloor L/2 \rfloor] g[k,l]$$

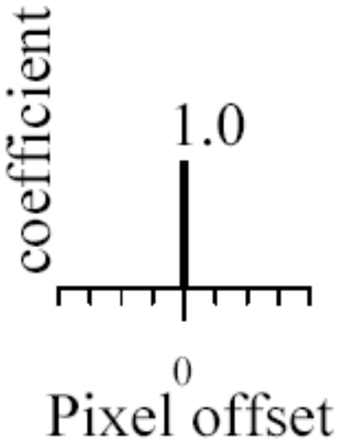
"filtered" image

filter "kernel"

Linear filtering (warm-up slide)



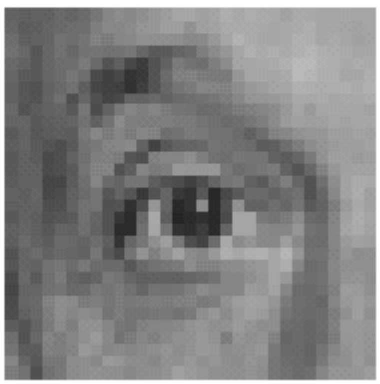
original



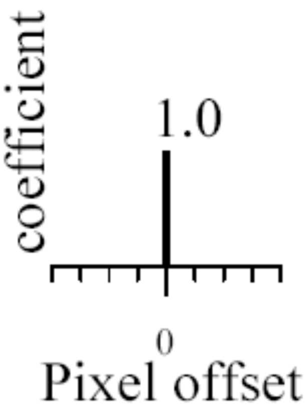
?

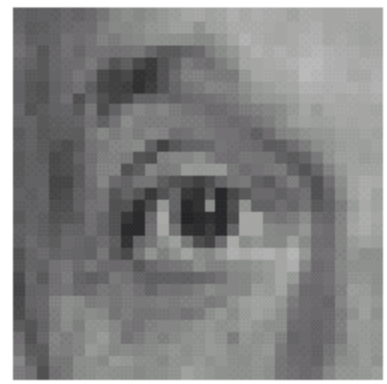
Freeman

Linear filtering (warm-up slide)



original

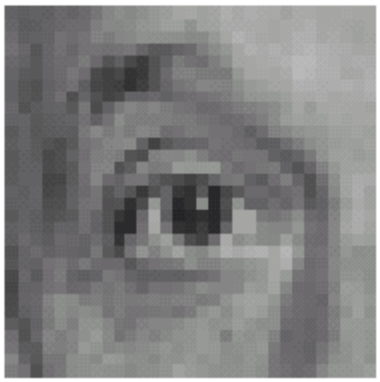




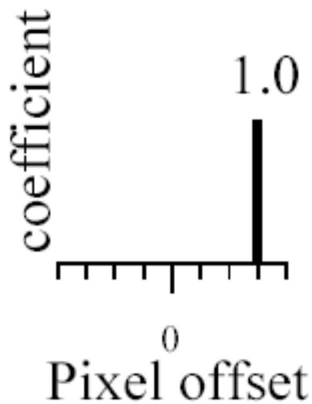
Filtered
(no change)

Freeman

Linear filtering



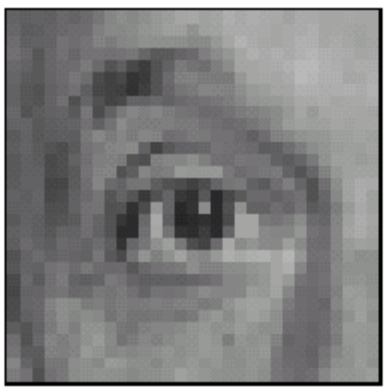
original



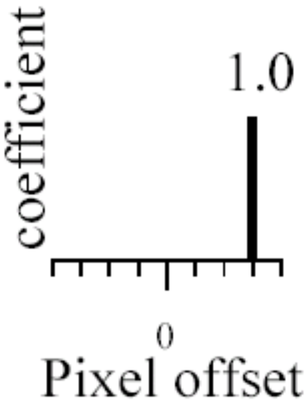
?

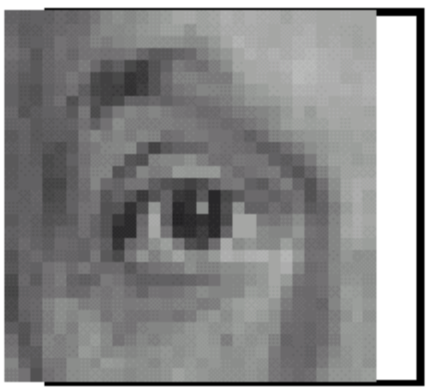
Freeman

shift



original

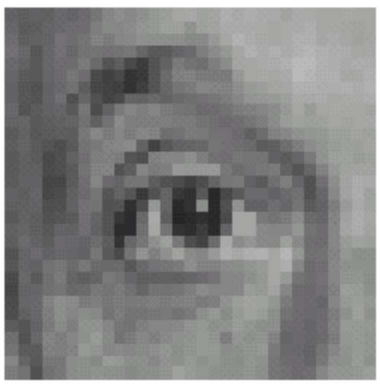




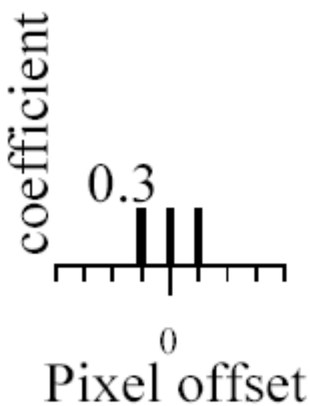
shifted

Freeman

Linear filtering



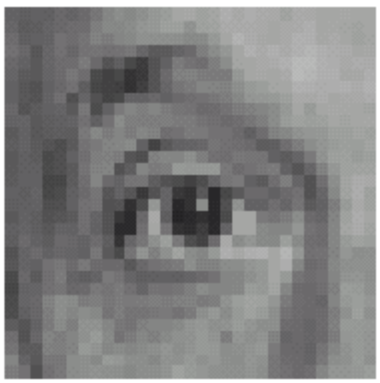
original



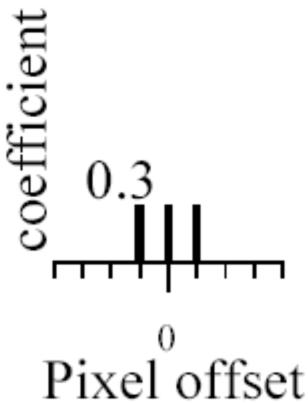
?

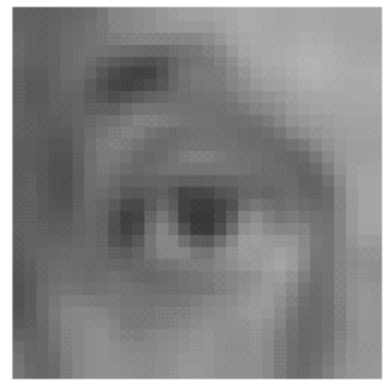
Freeman

Blurring



original

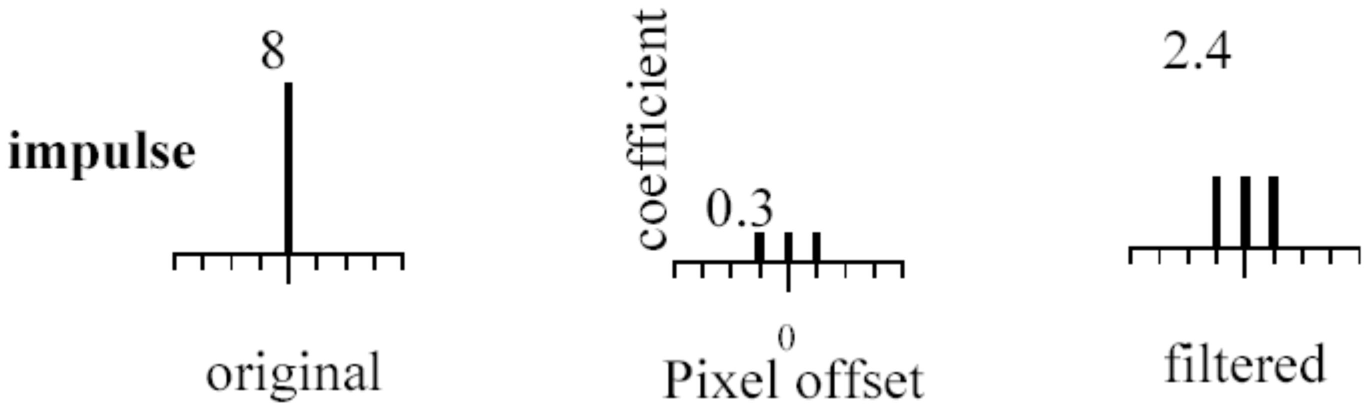




Blurred (filter applied in both dimensions).

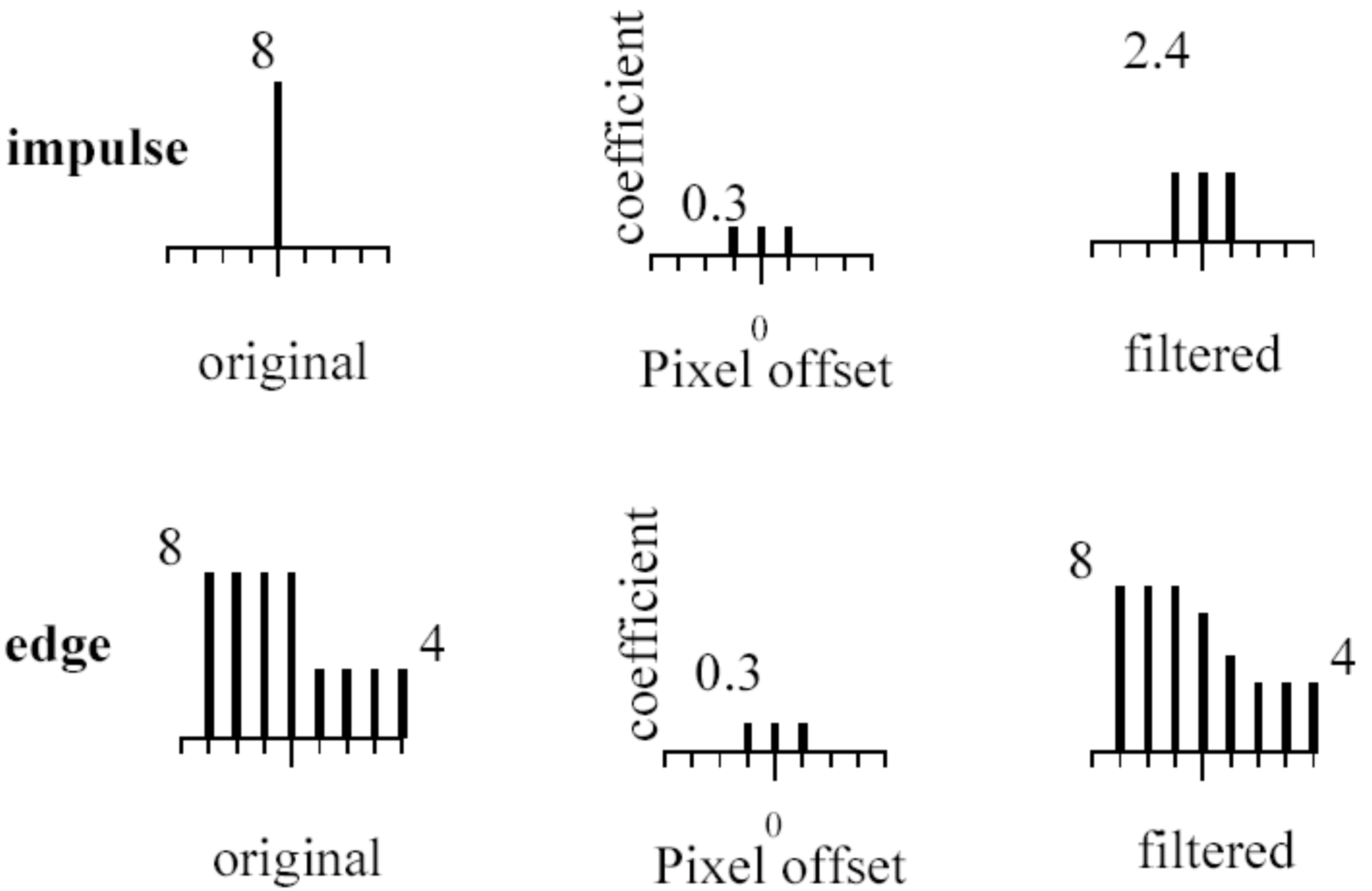
Freeman

Blur examples



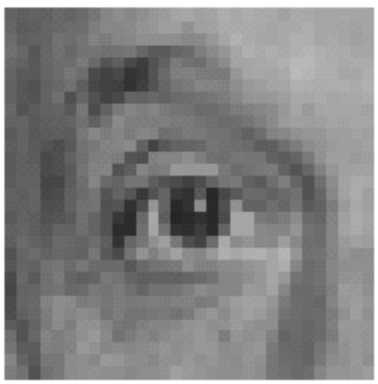
Freeman

Blur examples

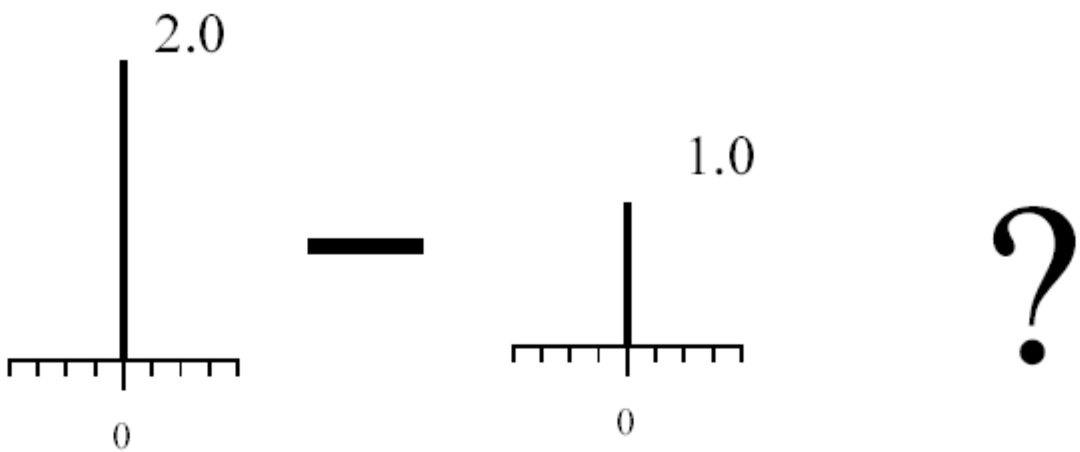


Freeman

Linear filtering (warm-up slide)

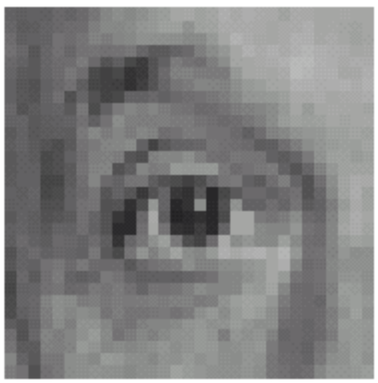


original

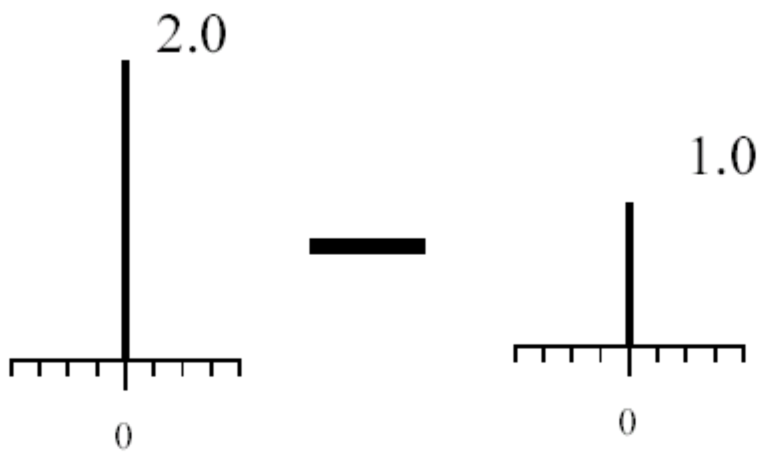


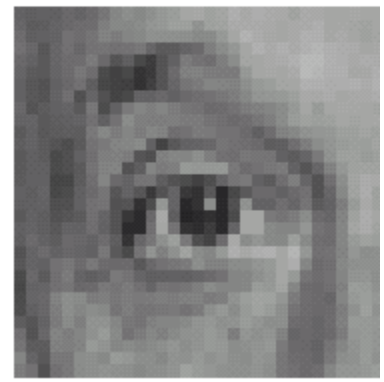
Freeman

Linear filtering (no change)



original

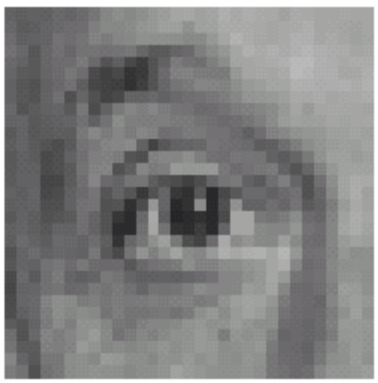




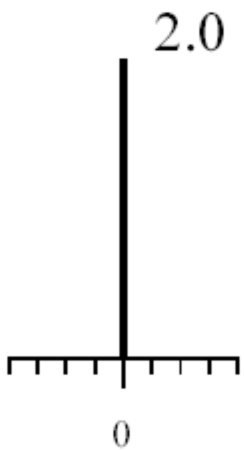
Filtered
(no change)

Freeman

Linear filtering



original

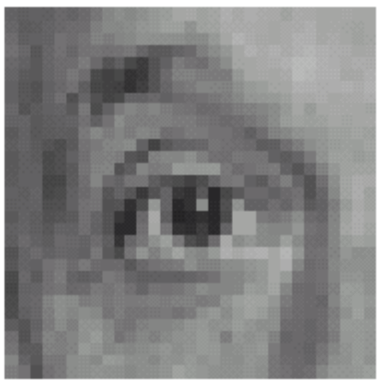


−

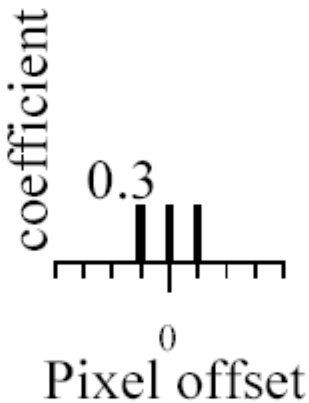
?

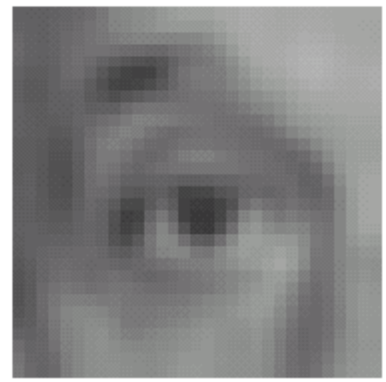
Freeman

(remember blurring)



original

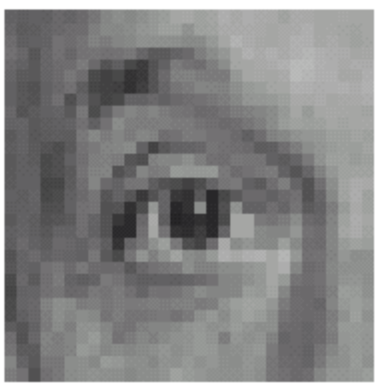




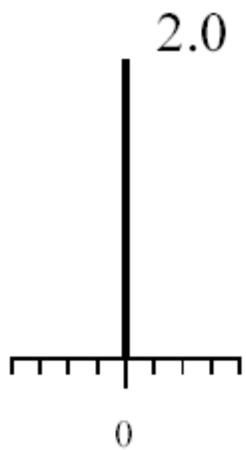
Blurred (filter applied in both dimensions).

Freeman

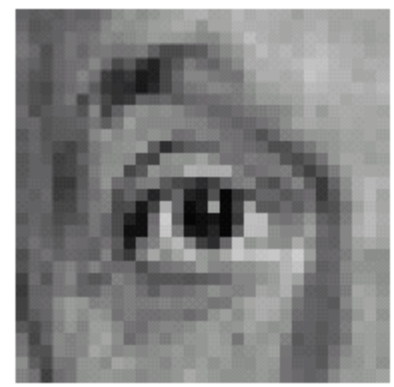
Sharpening



original



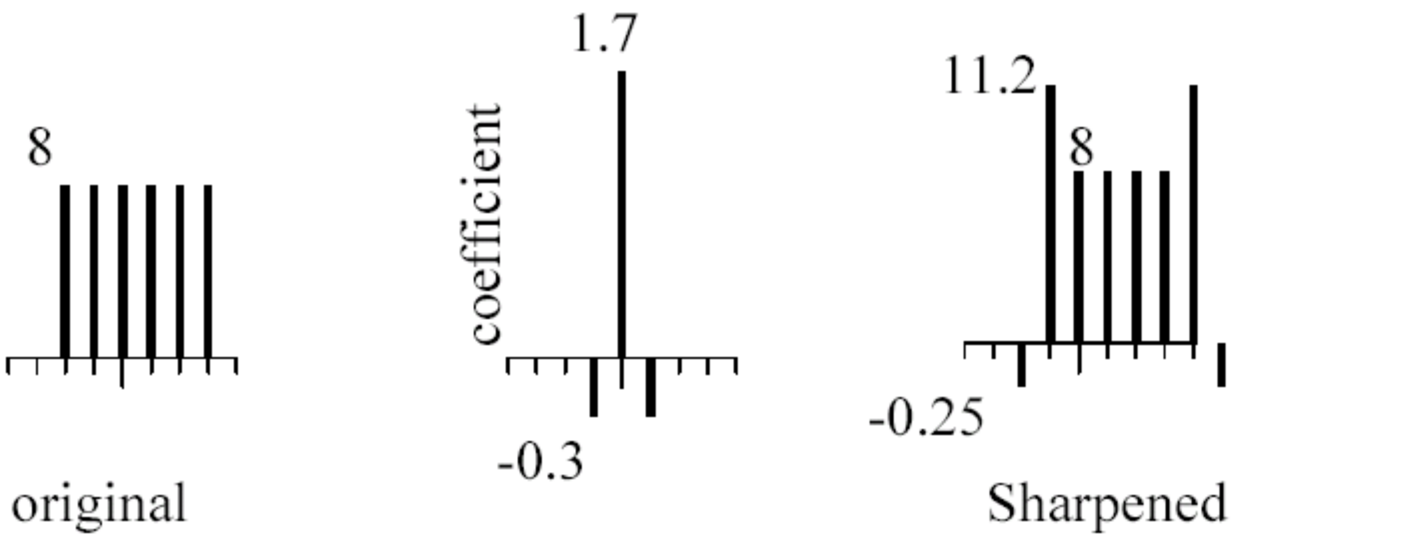
−



Sharpened
original

Freeman

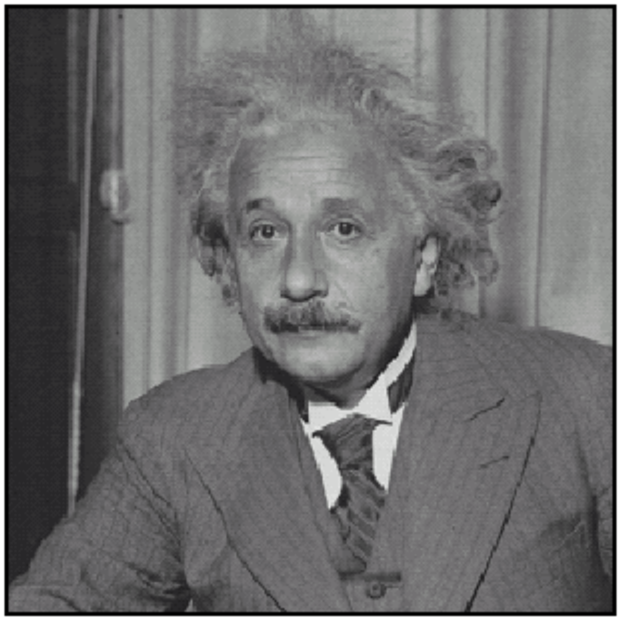
Sharpening example



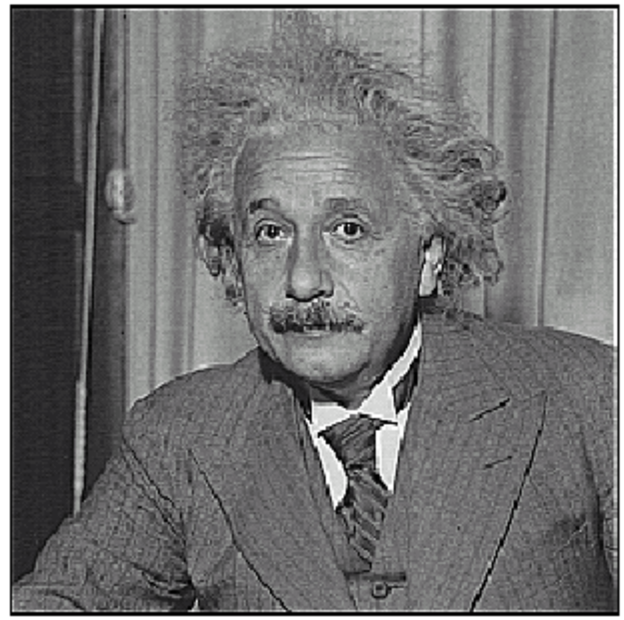
Sharpened (differences are accentuated; constant areas are left untouched).

Freeman

Sharpening



before



after

Freeman

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.

Average Filter

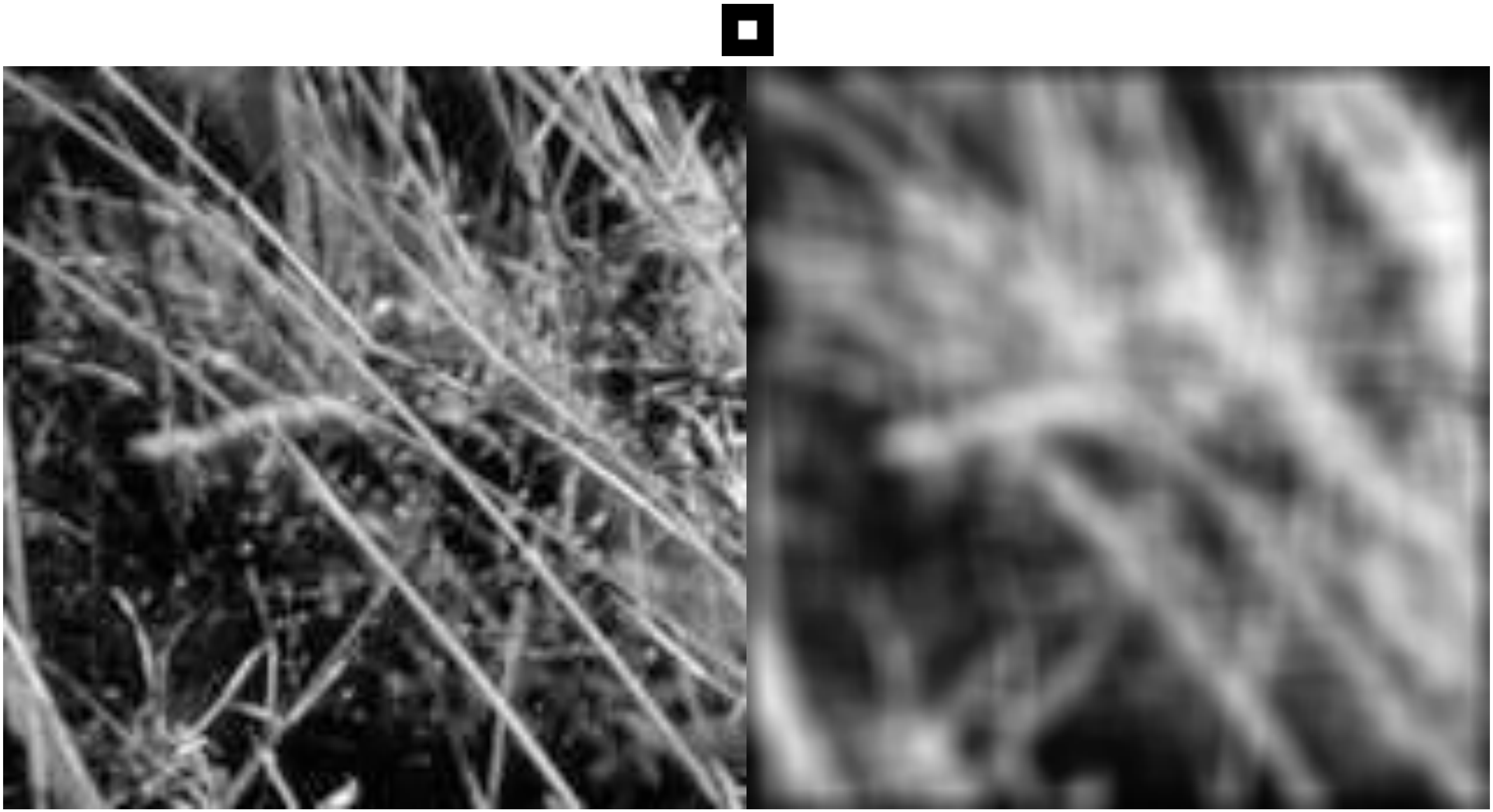
- Mask with positive entries, that sum 1.
- Replaces each pixel with an average of its neighborhood.
- If all weights are equal, it is called a BOX filter.

F

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

(Camps)

Example: Smoothing by Averaging



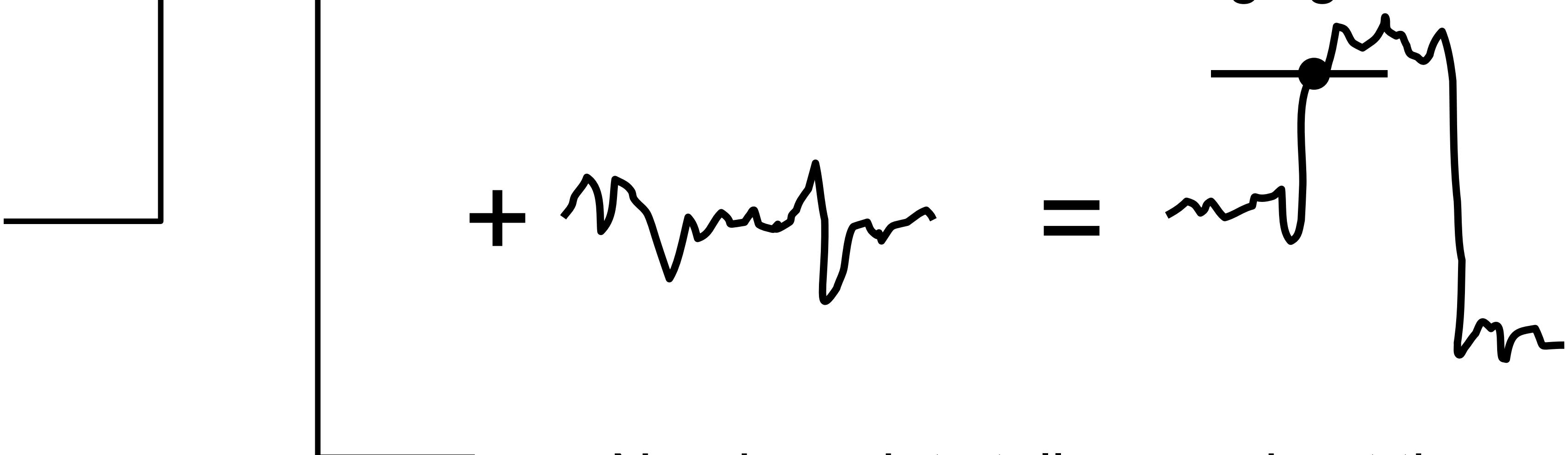
Linear systems

Basic properties. If T[.] is a linear operator, and a and b are scalars, then:

- homogeneity $T[a X] = a T[X]$
- additivity $T[X_1+X_2] = T[X_1]+T[X_2]$
- superposition $T[aX_1+bX_2] = aT[X_1]+bT[X_2]$
- Linear system ⇔ superposition

- Examples:
- matrix operations (additions, multiplication)
- convolutions

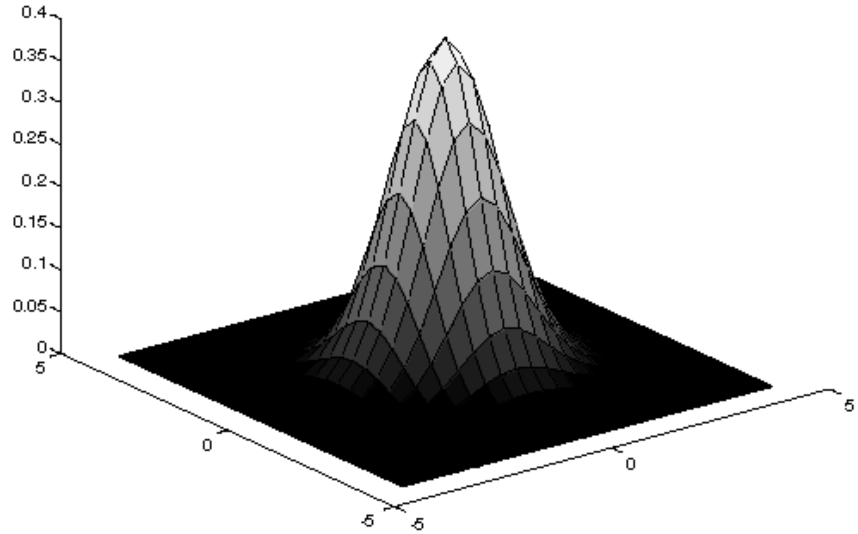
Smoothing as Inference About the Signal



Nearby points tell more about the signal than distant ones.

Gaussian Averaging

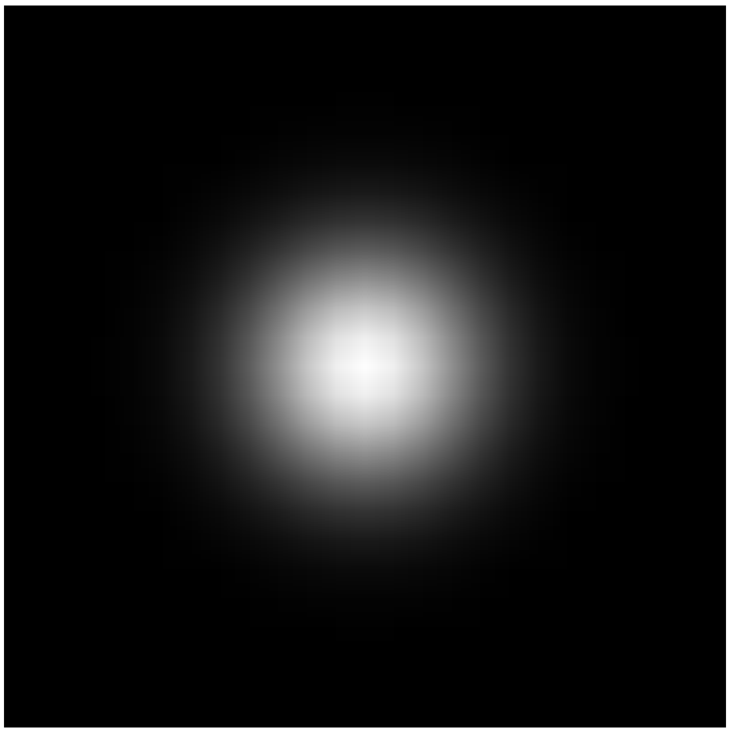
- Rotationally symmetric.
- Weights nearby pixels more than distant ones.
  - This makes sense as probabilistic inference.



- A Gaussian gives a good model of a fuzzy blob

Ponce & Forsyth

An Isotropic Gaussian



(0,0) in center

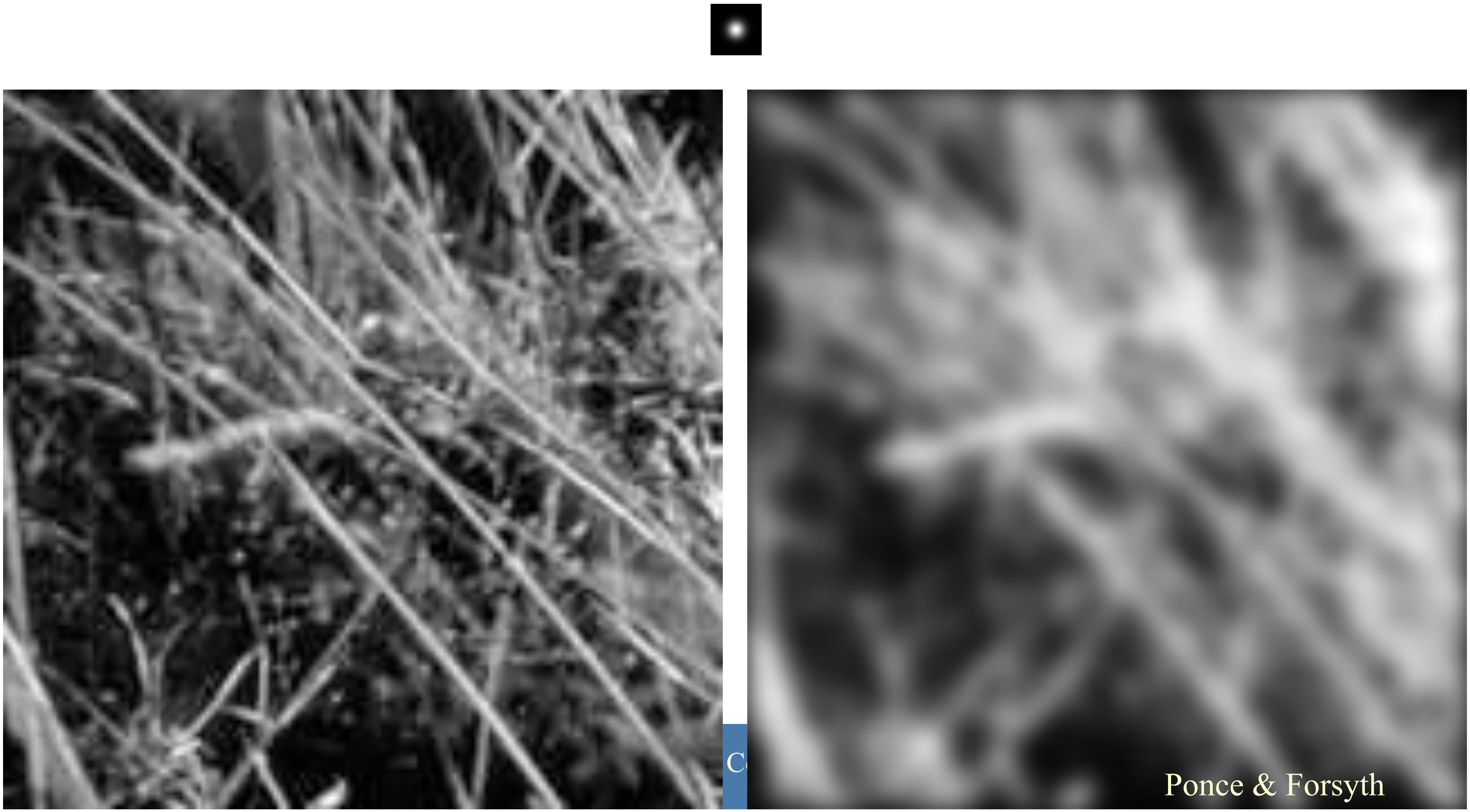
- The picture shows a smoothing kernel proportional to

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2} \right)$$

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

Ponce & Forsyth

Smoothing with a Gaussian



Homework

g=fspecial('gaussian',3,1)

g =

0.0751	0.1238	0.0751
0.1238	0.2042	0.1238
0.0751	0.1238	0.0751

B = imfilter(A,g,'symmetric','conv')
B = imfilter(A,g,'symmetric','corr')
% in the case of a symmetric filter, these are the same

Separable Gaussian

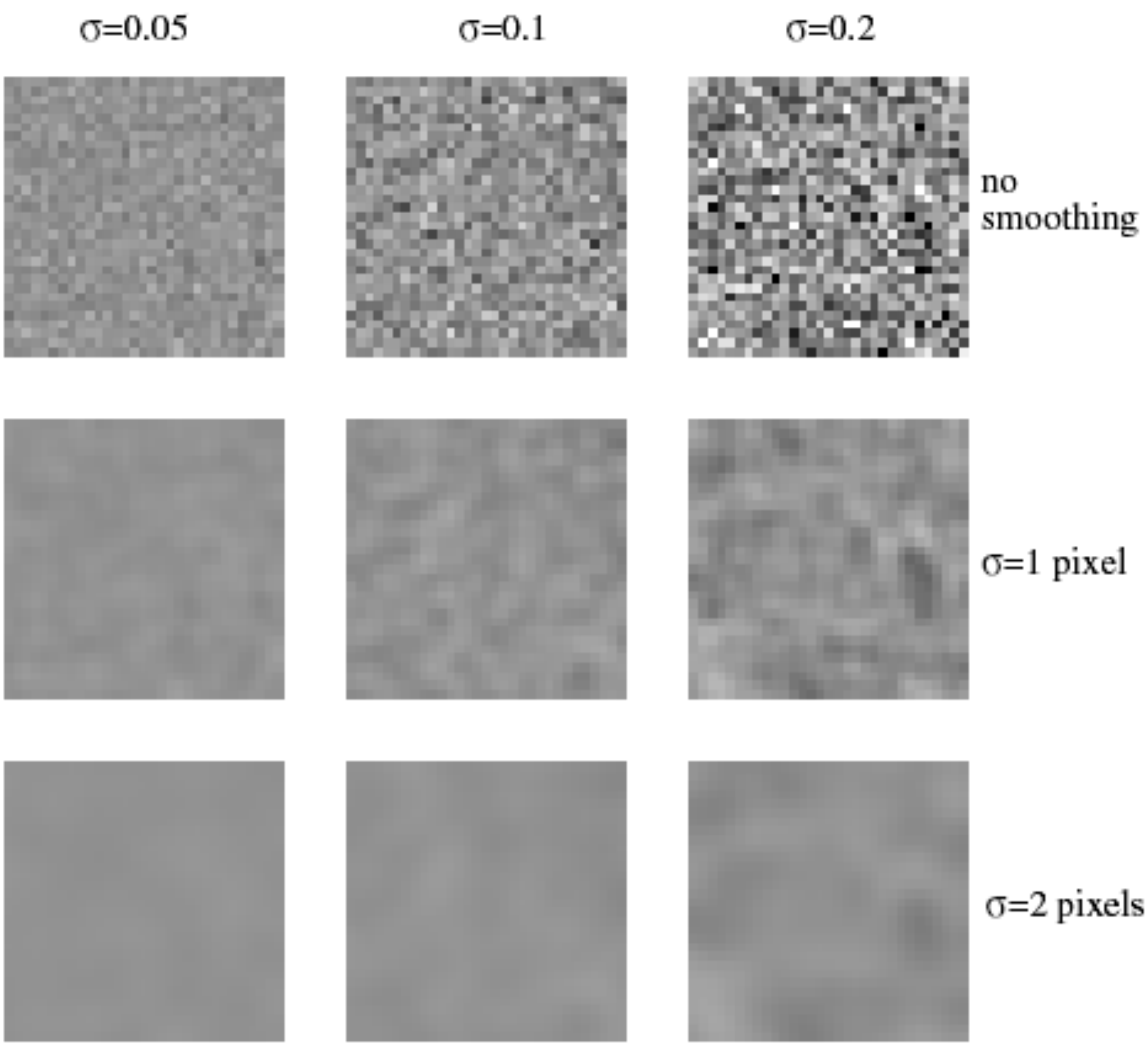
$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-x^2/(2\sigma^2)) = G_x$$

$$g(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/(2\sigma^2)) = G_y$$

Product?

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/(2\sigma^2))$$

$$G_x \otimes (G_y \otimes I) = (G_x \otimes G_y) \otimes I = G_{xy} \otimes I$$



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

Ponce & Forsyth

Multi-Resolution Image Representation

- Gaussian pyramids
- Laplacian Pyramids

Source: Irani

Motivation for studying scale.

ELDER AND ZUCKER: LOCAL SCALE CONTROL FOR EDGE DETECTION AND BLUR ESTIMATION

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 20, NO. 7, JULY 1998

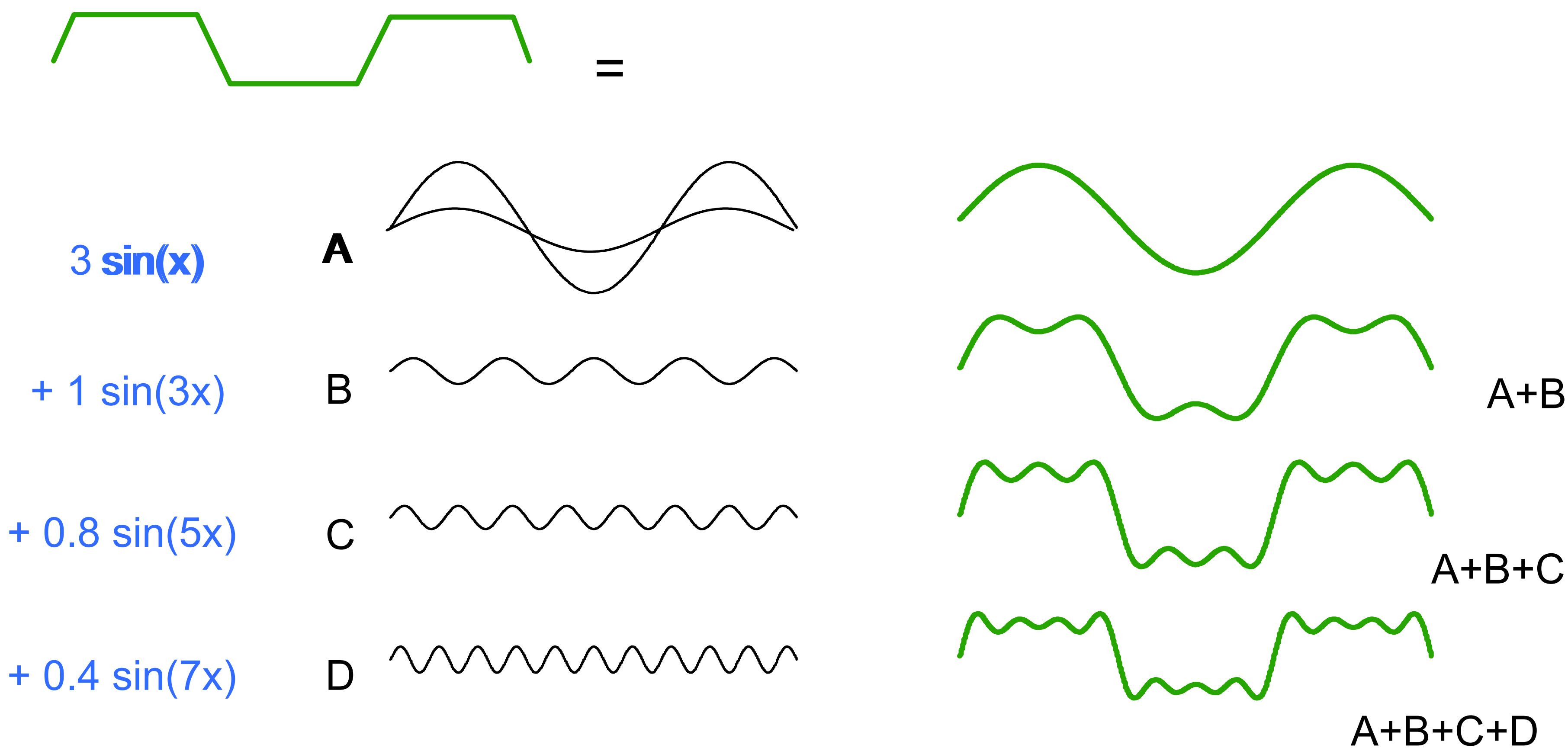
Motivation for studying scale.

ELDER AND ZUCKER: LOCAL SCALE CONTROL FOR EDGE DETECTION AND BLUR ESTIMATION

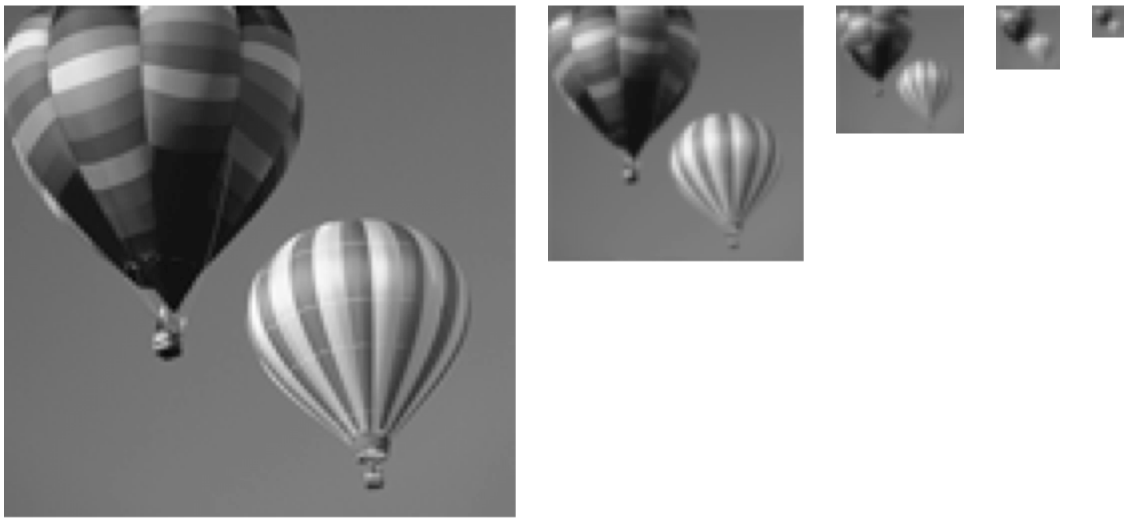
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 20, NO. 7, JULY 1998

A little Fourier so we can talk about spatial frequency:

Fourier Transform in Picutres



Gaussian Pyramid



High resolution → **Low resolution**

Source: Irani