EE421/521
Image Processing

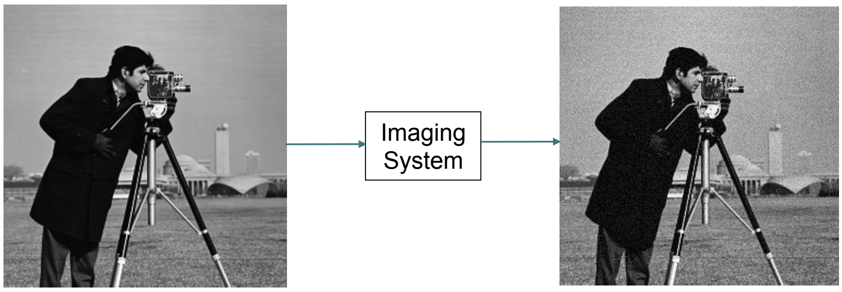
Lecture 10a
NOISE FILTERING

1

Introduction

2

Problem: Noise in the Image



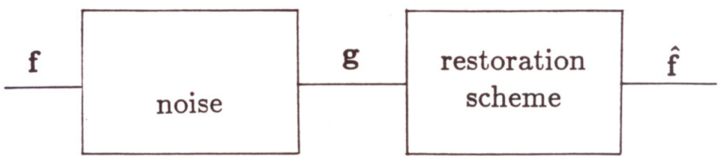
Original Image

Noisy Image

(random variations of intensity)

3

Noise Filtering



$\mathbf{f}$: original object

$\mathbf{g}$: distorted noisy image

$\hat{\mathbf{f}}$: restored image

$$\mathbf{g} = \mathbf{f} + \mathbf{n}$$

$$\mathbf{g} = \mathbf{f} + \mathbf{n}(\mathbf{f})$$

4

Signal Independent Additive Noise

$$g(i,j) = f(i,j) + n(i,j),$$

$n(i,j)$ is a signal-independent random noise process.

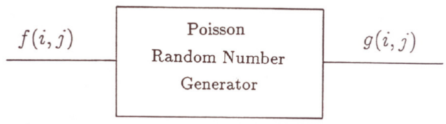
<u>Example</u>

- **Quantization noise**: Quantization of image density values to 8-bits introduces signal-independent noise with a uniform distribution.
- **Communication Noise**: The noise in the communication channel may sometimes be modelled as a signal-independent Gaussian noise.

5

Signal Dependent Additive Noise

Images at low light levels are corrupted by the Poisson noise associated with the discrete nature of light.



$$g(i,j) = f(i,j) + n_p(i,j)$$

$n_p(i,j)$ has zero mean and a variance equal to $f(i,j)$.

Signal to noise ratio (SNR) deteriorates as the average number of photons decreases

$$\mathrm{SNR} = \frac{f^2(i,j)}{\sigma^2_{n_p}(i,j)} = f(i,j)$$

<u>Example</u>: Medical imaging, Astronomical imaging.

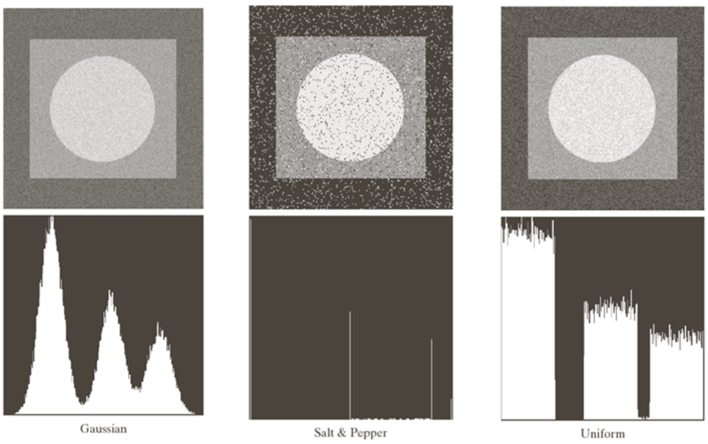
6

Noise Sources

- Shot (photon, film grain) noise
  - Caused by variation in the number of photons
  - Signal dependent, Poisson (similar to Gaussian)
- Salt-and-pepper (spike) noise
  - Caused by dead pixels, dust, scratches
  - Signal independent, impulsive
- Quantization noise
  - Caused by CCD quantization
  - Signal independent, uniform

7

Noise Distributions



8

Other Noise Sources: Sinusoidal Noise

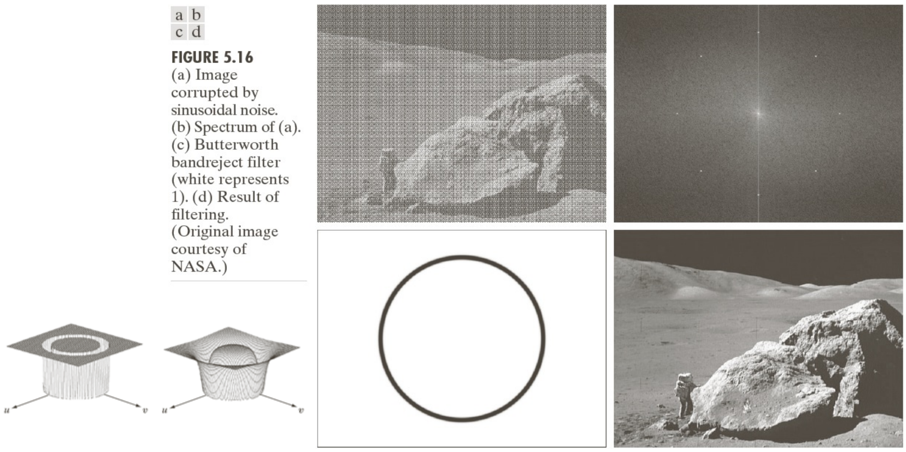


FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

9

Noise Filtering Methods

- Linear shift invariant (LSI) filtering
  - Low-pass filtering
  - LMMSE (Wiener) filter
- Locally adaptive (shift-varying) filtering
  - Local LMMSE filter
  - Directional smoothing
- Nonlinear filtering
  - Median filter

10

Noise Filtering

11

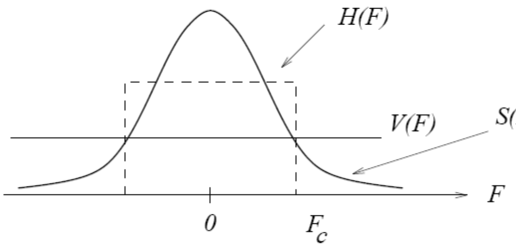
Low-pass Filtering

Additive, signal-independent, white noise

$$\underbrace{g(n_1,n_2)}_{\text{Noisy image}} = \underbrace{s(n_1,n_2)}_{\text{Noise-free image}} + \underbrace{v(n_1,n_2)}_{\text{Noise}}$$

Signal-to-noise ratio in dB:

$$\boxed{SNR = \log\frac{|S(f)|^2}{|V(f)|^2}}$$



Tradeoff between noise reduction and blurring.

12

Mean Filtering

- Approximates low-pass filtering

$$g(n_1,n_2) = s(n_1,n_2) + v(n_1,n_2)$$

$$\hat{s}(n_1,n_2) = \frac{1}{N}\sum_{(n_1,n_2)\in\aleph} g(n_1,n_2)$$

$\aleph$: Local neighborhood of (n_1,n_2)

N: Total number of pixels in $\aleph$

Mean Filtering
(in a 3x3 neighbourhood)

10	10	10	10	10	10
10	10	10	10	11	10
10	10	1	10	10	10
10	10	10	10	10	12
10	10	10	10	8	10
10	10	10	10	10	10

original

		9			
				10	

mean filtered

Median Filtering
(Order-Statistic Filtering)

- A non-linear filter that is edge preserving since it easily rejects outliers, avoids blurring edges

$$\hat{s}(n_1,n_2) = Med\{g(i_1,i_2)\} \quad \text{for } (i_1,i_2) \in \text{filter support}$$

15

Median Filtering
(in a 3x3 neighbourhood)

10	10	10	10	10	10
10	10	10	10	11	10
10	10	1	10	10	10
10	10	10	10	10	12
10	10	10	10	8	10
10	10	10	10	10	10

original

		10			
				10	

median filtered

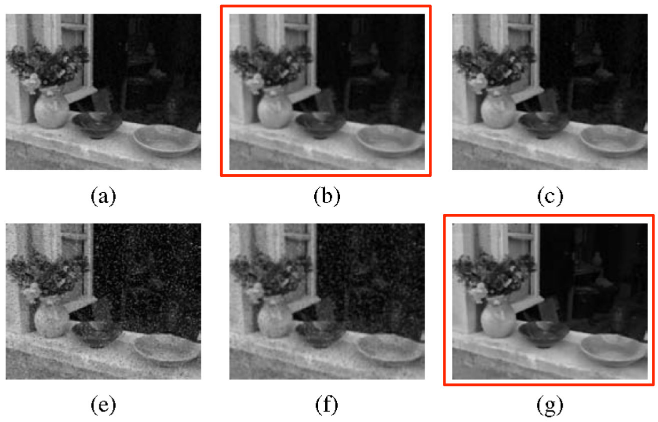
1	8
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	12

16

Mean vs. Median Filtering

Gaussian noise

Shot noise



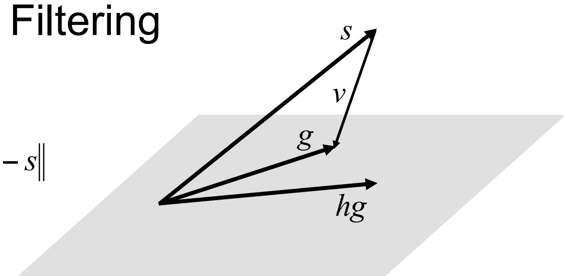
Noisy image

Mean filtered

Median filtered

17

Linear Minimum Mean Squared Error (LMMSE) Filtering



Find h to minimize $\|\hat{s} - s\|$

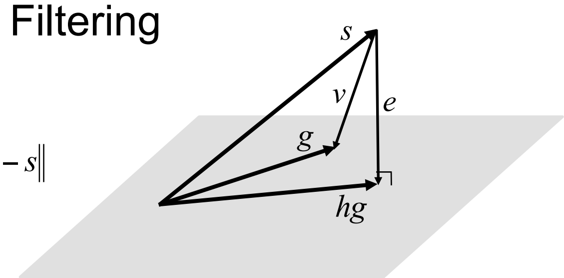
$g = s + v$ Observation model

$\hat{s} = hg$ Linear filter

18

Linear Minimum Mean Squared Error (LMMSE) Filtering

Find h to minimize $\|\hat{s} - s\|$



$g = s + v$	Observation model
$\hat{s} = hg$	Linear filter
$e = \hat{s} - s$	Estimation error
$e \perp hg$	MMSE achieved by orthogonality
$e \perp g$	becasuse e is perpendicular to the space spanned by hg

19

LMMSE Filtering

Orthogonality principle:

$$\hat{s} - s \perp g$$

Estimation error is orthogonal to observation

Thus

$$E\{[\hat{s}(n_1,n_2) - s(n_1,n_2)]g(k_1,k_2)\} = 0, \quad \forall\ (n_1,n_2) \text{ and } (k_1,k_2)$$

where

Linear time-invariant filter

$$\hat{s}(n_1,n_2) = \sum_{i_1=-\infty}^{\infty} \sum_{i_2=-\infty}^{\infty} h(i_1,i_2)g(n_1-i_1,n_2-i_2)$$

20

LMMSE Filtering

Stationary signal assumption

$$\sum_{i_1}\sum_{i_2} h(i_1,i_2)\underbrace{E\{g(n_1-i_1,n_2-i_2)g(k_1,k_2)\}}_{R_{gg}(n_1-i_1-k_1,n_2-i_2-k_2)} = \underbrace{E\{s(n_1,n_2)g(k_1,k_2)\}}_{R_{sg}(n_1-k_1,n_2-k_2)}$$

The discrete Wiener-Hopf equation (noncausal, IIR Wiener filter):

$$h(n_1,n_2) * *R_{gg}(n_1,n_2) = R_{sg}(n_1,n_2)$$

In the frequency domain

$$H(f_1,f_2) = \frac{P_{sg}(f_1,f_2)}{P_{gg}(f_1,f_2)}$$

Optimum LTI Filter for Additive Noise Model

- Assume that the image and noise are wss, zero-mean, and uncorrelated. Then, given the additive noise model,

$$\boxed{R_{sg}(n_1,n_2)} = E\{s(i_1,i_2)g(i_1-n_1,i_2-n_2)\}$$
$$= E\{s(i_1,i_2)s(i_1-n_1,i_2-n_2)\} + E\{s(i_1,i_2)v(i_1-n_1,i_2-n_2)\} = \boxed{R_{ss}(n_1,n_2)}$$
$$\boxed{R_{gg}(n_1,n_2)} = E\{(s(i_1,i_2)+v(i_1,i_2))(s(i_1-n_1,i_2-n_2)+v(i_1-n_1,i_2-n_2))\}$$
$$= \boxed{R_{ss}(n_1,n_2) + R_{vv}(n_1,n_2)}$$

- The LMMSE noise removal filter is given by

$$\boxed{H(f_1,f_2) = \frac{P_{ss}(f_1,f_2)}{P_{ss}(f_1,f_2)+P_{vv}(f_1,f_2)}}$$

Local (Space-Varying) LMMSE Filtering

Space-varying filter

$$\hat{s}(n_1,n_2) = \sum_{i_1} \sum_{i_2} h_{n_1,n_2}(i_1,i_2) g(n_1 - i_1, n_2 - i_2)$$

Orthogonality principle

Non-stationary signal

$$\sum_{i_1} \sum_{i_2} h_{n_1,n_2}(i_1,i_2) R_{gg}(n_1,n_2;k-i_1,l-i_2) = R_{sg}(n_1,n_2;k,l)$$

23

Local (Adaptive) LMMSE Filtering

Assume that the neighboring pixels are uncorrelated:

$$w(n_1,n_2) = s(n_1,n_2) - \mu_s(n_1,n_2)$$

$$\boxed{R_{ww}(n_1,n_2;k,l) = \sigma_s^2(n_1,n_2)\delta(k,l)}$$

where the local mean and variance are allowed to be space-varying:

$$\mu_s(n_1,n_2) = E[s(n_1,n_2)] \approx \frac{1}{N} \sum_{(k,l)\in\aleph} s(n_1+k, n_2+l)$$

$$\sigma_s^2(n_1,n_2) \approx \frac{1}{N} \sum_{(k,l)\in\aleph} \left(s(n_1+k,n_2+l) - \mu_s(n_1,n_2)\right)^2$$

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

$\aleph$

24

Local LMMSE Filtering

Likewise, define the observed residual image:

$$y(n_1,n_2) = g(n_1,n_2) - \mu_g(n_1,n_2)$$

Assuming that the observation noise is zero mean, we have

$$\mu_g(n_1,n_2) = \mu_s(n_1,n_2)$$

Thus, we can write the following observation model in terms of the residual images:

$$y(n_1,n_2) = w(n_1,n_2) + v(n_1,n_2)$$

25

Local LMMSE Filtering

Assuming that the original image and the noise are uncorrelated we also have

$$\boxed{R_{yy}(n_1,n_2;k,l) = \sigma_g^2(n_1,n_2)\delta(k,l)}$$

where

$$\sigma_g^2 = \sigma_s^2 + \sigma_v^2$$

26

Local LMMSE Filtering

Hence, from the orthogonality principle

$$h_{n_1,n_2}(k,l)\sigma_g^2(n_1,n_2)\delta(k,l) = \sigma_s^2(n_1,n_2)\delta(k,l)$$

which implies

$$h_{n_1,n_2}(k,l) = \begin{cases} \dfrac{\sigma_s^2(n_1,n_2)}{\sigma_s^2(n_1,n_2)+\sigma_v^2(n_1,n_2)} & \text{if } (k,l)=(0,0) \\ 0 & \text{elsewhere} \end{cases}$$

27

Local LMMSE Filtering

The filtered image is then given by

$$\boxed{\hat{s}(n_1,n_2) = \mu_g(n_1,n_2) + \frac{\sigma_s^2}{\sigma_g^2}\left(g(n_1,n_2) - \mu_g(n_1,n_2)\right)}$$

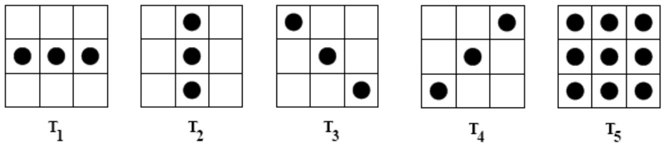
where

$$\sigma_s^2 = \max\left(\sigma_g^2 - \sigma_v^2, 0\right)$$

- Approaches local averaging (box filter) when σ_s^2 is small,
- No filtering at all when σ_s^2 is large (edge-preserving).

28

Directional Spatial Filtering



- *Method I* Compute variances over each of the directions, apply an averaging filter in the direction of the smallest variance.
- *Method II*

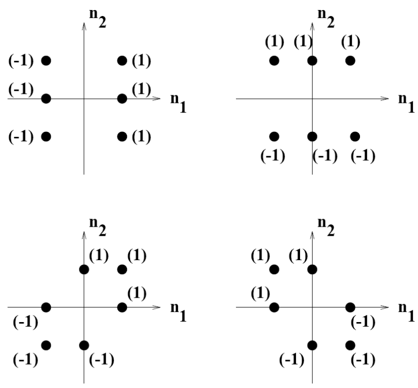
$$T = T_1 T_2 T_3 T_4 T_5$$

- T_i is the local LMMSE filter applied over the respective window.

Better noise reduction around edges, since at least one of the windows should have a small variance.

29

Method III: Apply Averaging Along the Detected Edge



30

Bilateral Filtering

$$g(i,j) = \frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$ Bilateral filtering kernel

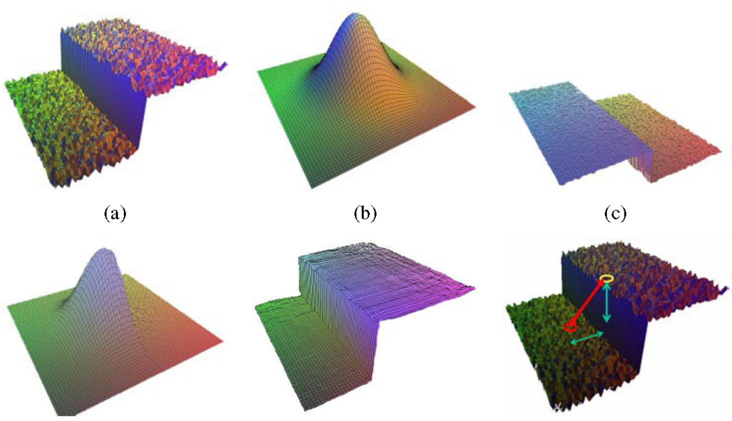
(b) $$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right)$$ Smoothing weight

(c) $$r(i,j,k,l) = \exp\left(-\frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2}\right)$$ Range weight (data dependent)

(d) $$w(i,j,k,l) = d(i,j,k,l)\ r(i,j,k,l)$$ Combined (bilateral) weight

31

Bilateral Filtering



(a) (b) (c)

(d) (e) (f)

32

Bilateral Filtering

(a) noisy image

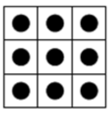
(b) linear time-invariant filter

(d) bilateral filter (edge preserving)

33

Adaptive Median Filtering

$z_{\min}$ = minimum intensity value in S_{xy}
$z_{\max}$ = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity value at coordinates (x, y)
$S_{\max}$ = maximum allowed size of S_{xy}



S_{xy}

Stage A (Zmed impulse?)

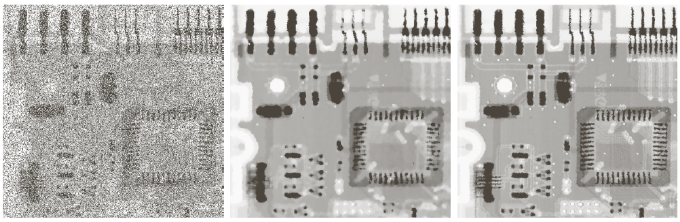
$A1 = z_{\text{med}} - z_{\min}$
$A2 = z_{\text{med}} - z_{\max}$
If $A1 > 0$ AND $A2 < 0$, go to stage B
Else increase the window size
If window size $\leq S_{\max}$ repeat stage A
Else output z_{med}

Stage B (Zmed is not impulse)

$B1 = z_{xy} - z_{\min}$
$B2 = z_{xy} - z_{\max}$
If $B1 > 0$ AND $B2 < 0$, output z_{xy}
Else output z_{med} (Zxy is impulse)

34

Adaptive Median Filtering



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

- Removes impulsive noise (by using the median value)
- Less distortion in edges (by starting with a small window and by being able to keep the original value)

35

Summary of Noise Filters

- LSI noise filtering
  - Noise reduction at the expense of spatial blurring
- Local linear filtering
  - Edge-disabled
- Directional filtering
  - Edge-preserving
- Median filtering
  - Works for impulsive noise, preserves edges

36

Project 1.10 (3.1)

Noise Filtering

Due 22.12.2013 Sunday

37

Local LMMSE Filtering

The filtered image is given by

$$\boxed{\hat{s}(n_1,n_2) = \mu_g(n_1,n_2) + \frac{\sigma_s^2}{\sigma_g^2}\left(g(n_1,n_2) - \mu_g(n_1,n_2)\right)}$$

where

$$\sigma_s^2 = \max\left(\sigma_g^2 - \sigma_v^2, 0\right)$$

- Approaches local averaging (box filter) when σ_s^2 is small,
- No filtering at all when σ_s^2 is large (edge-preserving).

38

Problem 1.10

1. Select a monochrome image and display it.
2. Add 20 dB Gaussian noise to the image and display the result. Use the following formula to obtain the noise variance:

$$PSNR = 10\log\frac{255^2}{\sigma_v^2} = 20dB$$

3. Filter the noisy image with the local LMMSE filter. Use a 3x3 window to find the local mean and variance. Display and comment on the resulting image.
4. Repeat Step 3 with a 7x7 window.
5. Compare your results with MATLAB's adaptive filter.

Next Lecture

- BLUR IDENTIFICATION