

Chapter 5 Image Restoration

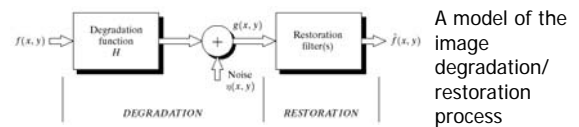
Contents

- Model
 - Noise model
 - Degradation Model
- Filter
 - Spatial Domain
 - Frequency Domain
- MATLAB® & Video
 - Restoration Examples
 - Video

Image Enhancement vs Restoration

- Enhancement
 - Goal: to improve an image in predefined sense
 - Subjective process
- Restoration:
 - Goal: to recover the original image
 - Objective process

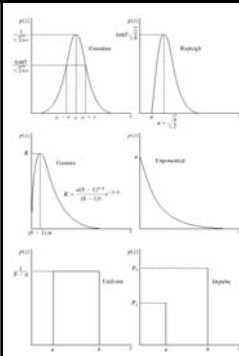
Restoration Model



A model of the image degradation/restoration process

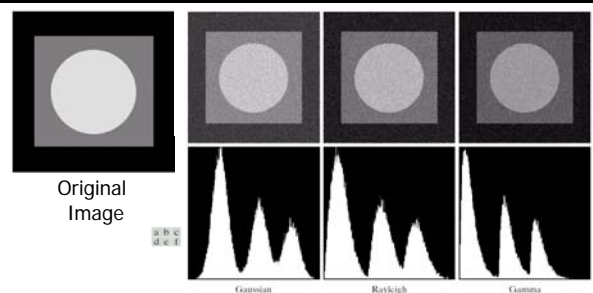
- Degradation model H and noise must be known/predicted first before restoration
- Spatial domain: $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$
- Frequency domain: $G(u, v) = H(u, v)F(u, v) + N(u, v)$

Noise PDF (Probability Density Function)



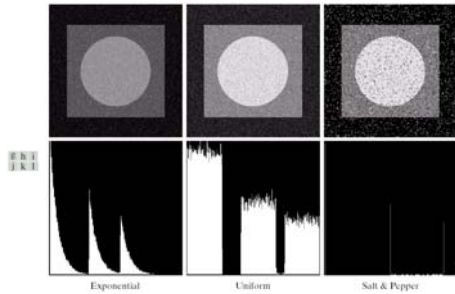
- Gaussian noise
 - Rayleigh noise
 - Erlang (gamma) noise
 - Exponential noise
 - Uniform noise
 - Impulse (S&P) noise
- $$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2}$$
- $$p(z) = \begin{cases} \frac{1}{b} e^{-z/b} & \text{for } z > 0 \\ 0 & \text{for } z < 0 \end{cases}$$
- $$p(z) = \begin{cases} \frac{a^k}{\Gamma(k)} z^{k-1} e^{-az} & \text{for } z > 0 \\ 0 & \text{for } z < 0 \end{cases}$$
- $$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$
- $$p(z) = \begin{cases} ac^{-z} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$
- $$p(z) = \begin{cases} P_1 & \text{for } z = a \\ P_2 & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

Image with noises



Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the original image

Image with noises (contd.)

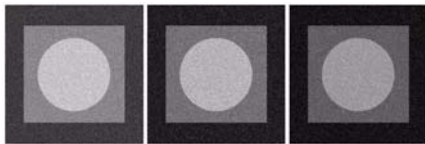


Images and histograms resulting from adding exponential, uniform, and impulse noise to the original image

Histogram of noises



Histograms computed using small strips (shown as inserts) from: (a) Gaussian; (b) Rayleigh; and (c) uniform noisy images as below



Pepper, Salt Noise and Their Filters

- a) Image corrupted by pepper noise with a probability of 0.1
- b) Image corrupted by salt noise with a probability of 0.1
- c) Result of filtering a) with a 3x3 contra harmonic filter of order 1.5

$$\hat{f}(x, y) = \frac{\sum_{(s, t) \in N} g(s, t)^{Q+1}}{\sum_{(s, t) \in N} g(s, t)^Q}$$

- d) Result of filtering b) with $Q=-1.5$

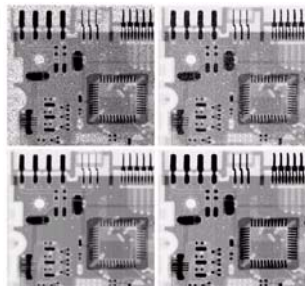


Image corrupted by sinusoid

Regular noise

- (a) Image corrupted by sinusoidal noise
- (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave)



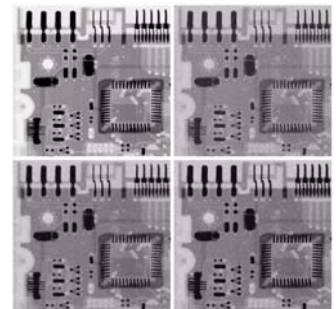
Gaussian Noise and Mean Filter

- a) X-ray image
- b) Image corrupted by additive Gaussian noise
- c) Result of filtering with an arithmetic mean filter of size 3x3

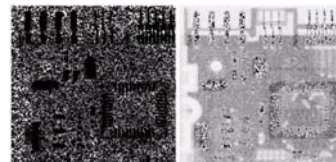
$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in N} g(s, t)$$

- a) Result of filtering with geometric mean filter of

$$\hat{f}(x, y) = \left[\prod_{(s, t) \in N} g(s, t) \right]^{1/n}$$



Wrong sign

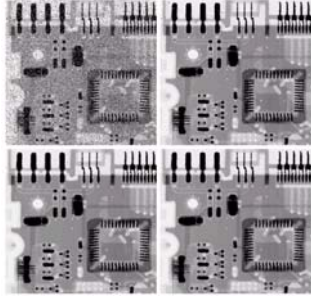


- Results of selecting the wrong sign in contra-harmonic filtering
- a) Result of filtering pepper corrupted image with a contra-harmonic filter of size 3x3 and $Q=-1.5$
 - b) Result of filtering salt corrupted image with a contra-harmonic filter of size 3x3 and $Q=1.5$

Salt & Pepper vs Median Filter

- a) Image corrupted by salt and pepper noise (prob. $P_a=P_b=0.1$)
- b) Result of one pass of median filter (3×3)
- c) Result of two pass of this filter
- d) Result of three pass of this filter

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{3 \times 3}} \{g(s, t)\}$$



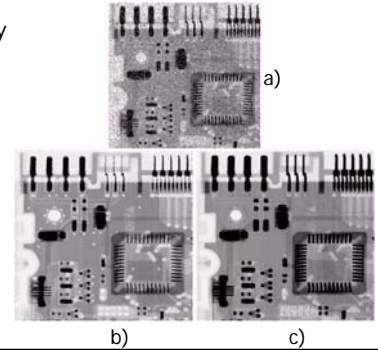
Max – Min Filter

Effect of Max – Min Filter to Salt and Pepper

- a) Image corrupted by salt and pepper noise (prob. $P_a=P_b=0.1$)
- b) Result of max filtering (3×3)
- c) Result of min filtering (3×3)

$$\hat{f}(x, y) = \max_{(s,t) \in S_{3 \times 3}} \{g(s, t)\}$$

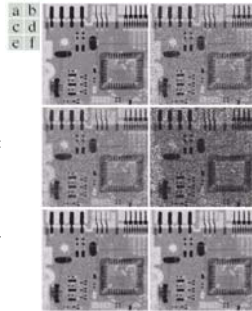
$$\hat{f}(x, y) = \min_{(s,t) \in S_{3 \times 3}} \{g(s, t)\}$$



Additive + S&P vs Mean, Median

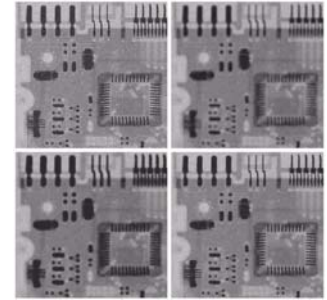
- a) Image corrupted by additive uniform noise
 - b) Additionally corrupted by additive salt-and-pepper noise
- Image in b) filtered with a 5×5 :
- c) Arithmetic mean filter
 - d) Geometric mean filter
 - e) Median filter
 - f) Alpha-trimmed mean filter with $d=5$

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_d} g(s, t)$$

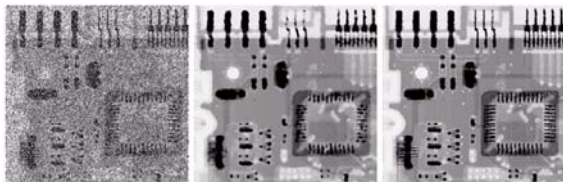


Gaussian noise vs Mean, Adaptive

- a) Image corrupted by additive Gaussian noise of zero mean and variance 1000
 - b) Result of arithmetic mean filtering
 - c) Result of geometric mean filtering
 - d) Result of adaptive noise reduction filtering
- Note: all filters were of size 7×7



S&P vs Median, Adapt. Median



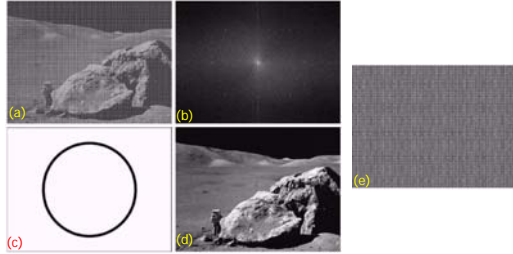
- a) Image corrupted by salt-and-pepper noise with probabilities $P_a=P_b=0.25$; b) Result of filtering with a 7×7 median filter; c) Result of adaptive median filtering with $S_{\max}=7$

Bandreject Filters



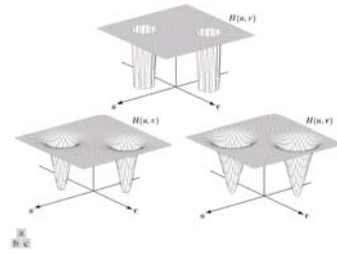
Perspective plots of (a) ideal; (b) Butterworth (of order 1); and (c) Gaussian bandreject filters

Sinusoidal vs BBRF



(a) Image corrupted by sinusoidal noise; (b) Spectrum of (a); (c) Butterworth bandreject filter (white represents 1); (d) Result of filtering; (e) Noise pattern of image (a) obtained by bandpass filtering

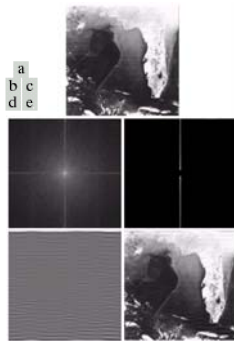
Notch (reject) Filters



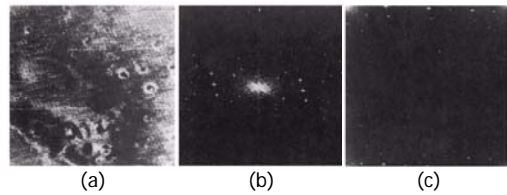
Perspective plots of (a) ideal; (b) Butterworth (of order 2); and (c) Gaussian notch (reject) filters

Notch Pass Filter

(a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines); (b) Spectrum; (c) Notch pass filter shown superimposed on (b); (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain; (e) Result of notch reject filtering

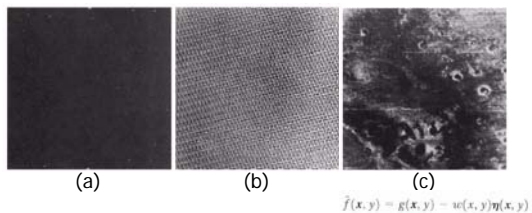


Optimum notch filtering



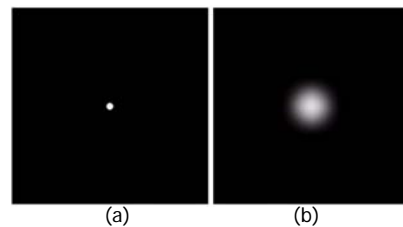
(a) Image of the Martian terrain taken by Mariner A; (b) Fourier spectrum showing periodic interference; (c) Fourier spectrum (without shifting)

Fourier spectrum of image & noise



(a) Fourier spectrum of $N(u,v)$; (b) Noise interference pattern $\eta(x,y)$; (c) Processed image

Estimation by experimentation



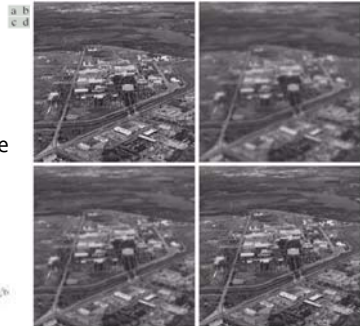
Degradation estimation by impulse characterization $H(u,v) = \frac{G(u,v)}{A}$
 (a) An impulse of light (shown, magnified); (b) Imaged (degraded) impulse.

Estimation by modeling

Illustration of the atmospheric turbulence model

- (a) Negligible turbulence
- (b) Severe turbulence ($k=0.0025$) → ref.
- (c) Mild turbulence ($k=0.001$)
- (d) Low turbulence ($k=0.00025$)

$$H(u, v) = e^{-k(u^2+v^2)^{5/6}}$$

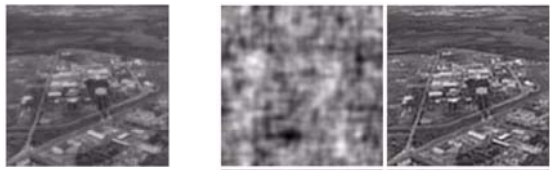


Blurring due motion



(a) Original image; (b) Result of blurring using function $H(u, v) = \frac{T}{\pi(mu + vb)} \sin[\pi(mu + vb)] e^{-j\pi u v}$, ($a=b=0.1$ and $T=1$)

Inverse Filtering



Restoring severe turbulenced image with

$$H(u, v) = e^{-k[(u-N/2)^2 + (v-N/2)^2]^m}$$

- (a) Using full filter
- (b) With H cutoff outside a radius of 40
- (c) Outside a radius of 70
- (d) Outside a radius of 85

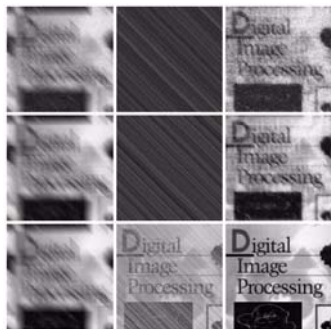
Comparison of Wiener filtering



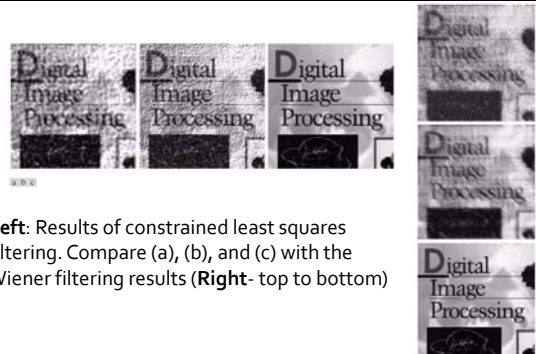
Left: Severe turbulenced image Right: (a) Result of full inverse filtering; (b) Radially limited inverse filter result; (c) Wiener filter result

Further comparison of Wiener filtering

- (a) Image corrupted by motion and additive noise
 - (b) Result of inverse filtering
 - (c) Result of Wiener filtering
 - (d)-(f) Same sequence, but with noise variance one order of magnitude less
 - (g)-(i) Same sequence, but with noise variance reduced five orders of magnitude from (a)
- Note in (h) how the deblurred image is quite visible through a "curtain" of noise.



Wiener & Least square filtering



Left: Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results (Right- top to bottom)

Constrained least square restoration

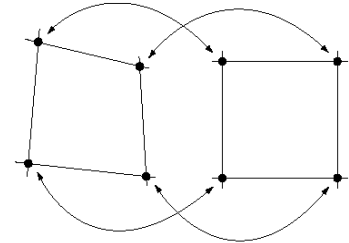


(a) (b) (c)

(a) Severe turbulenced image; (b) Iteratively determined constrained least squares restoration of the image, using correct noise parameters; (c) Result obtained with wrong noise parameters.

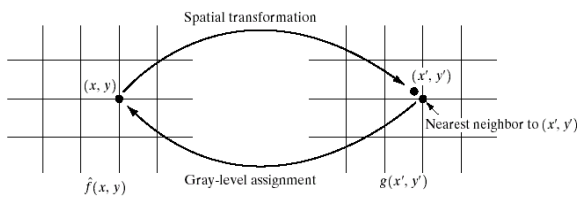
Spatial transformations: tiepoints

Corresponding tiepoints in two image segments



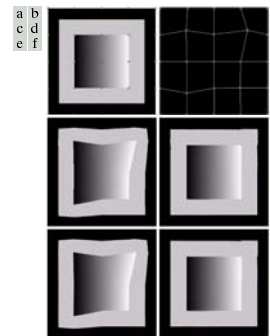
Gray level interpolation

Gray-level interpolation based on the nearest neighbor concept



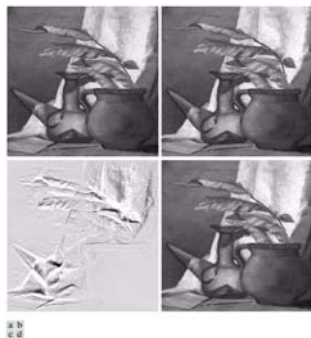
Geometric transformations

- (a) Image showing tiepoints
- (b) Tiepoints after geometric distortion
- (c) Geometrically distorted image, using nearest neighbor interpolation
- (d) Restored result
- (e) Image distorted using bilinear interpolation
- (f) Restored image



Geometric distortion (more textured image)

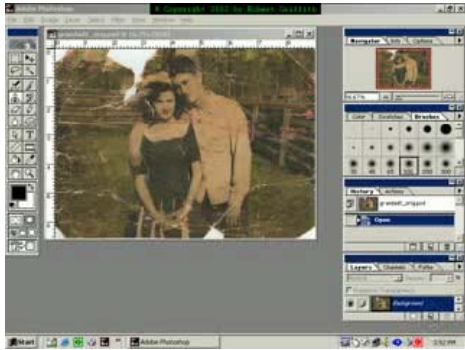
- (a) An image before geometric distortion
- (b) Image geometrically distorted using the same parameters as in previous figure
- (c) Difference between (a) and (b)
- (d) Geometrically restored image



MATLAB® Time

Howto, Filtering, etc.

Video time



37