Outline

- Image Restoration by Filtering (Textbook 5.3)
Image Restoration in the Frequency Domain

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]

\[ \hat{F}(u, v) = G(u, v)H_R(u, v) \]
Review

$X$ random variable \quad c \quad \text{deterministic constant}

**Expected value**

\[
E[c + X] = c + E[X] \\
E[cX] = cE[X]
\]

**Variance**

\[
\text{Var}[cX] = c^2\text{Var}[X] \\
\text{Var}[c + X] = \text{Var}[X]
\]
Review

corrupted image

\[ g(x, y) = f(x, y) \ast h(x, y) + \eta(x, y) \]

Expected value

\[ E[g(x, y)] = f(x, y) \ast h(x, y) + E[\eta(x, y)] \]

Variance

\[ \text{Var}[g(x, y)] = E[g(x, y)^2] - E^2[g(x, y)] \]

\[ \text{Var}[g(x, y)] = \text{Var}[\eta(x, y)] \]
Gaussian Noise + Arithmetic vs. Geometric Mean Filter

\[ g(x, y) = f(x, y) + \eta(x, y) \]

\( S_{xy} \)

filter window

output input

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

arithmetic mean geometric mean
Gaussian Noise + Arithmetic vs. Geometric Mean Filter

\[ g(x, y) = f(x, y) + \eta(x, y) \]

output          input

\[ \hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \]

arithmetic mean filtering

\[ \hat{f}(x, y) = \left[ \prod_{(s,t) \in S_{xy}} g(s, t) \right] ^{1/mn} \]

gameometric mean filtering

\[ \text{arithmetic mean} \hspace{1cm} \text{geometric mean} \]
Salt-and-Pepper Noise + Median Filter

\( g(x, y) = f(x, y) + \eta(x, y) \)

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

repeated application of median filter
Adaptive Filter

**arithmetic mean**

\[ m_{S_{xy}} = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \]

**arithmetic variance**

\[ \sigma^2_{S_{xy}} = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} (g(s, t) - m_{S_{xy}})^2 \]

**output of the filter**

\[ \hat{f}(x, y) = g(x, y) - \frac{\sigma^2_\eta}{\sigma^2_{S_{xy}}} \left[ g(x, y) - m_{S_{xy}} \right] \]
Gaussian Noise + Adaptive Filter

\[
\hat{f}(x, y) = g(x, y) - \frac{\sigma^2}{\sigma^2 S_{xy}} [g(x, y) - m S_{xy}]
\]

Properties:

- **Zero-noise**
  \[\sigma^2_\eta = 0 \quad \Rightarrow \quad \hat{f}(x, y) = g(x, y)\]

- **On edges**
  \[\sigma^2_\eta \ll \sigma^2 S_{xy} \quad \Rightarrow \quad \hat{f}(x, y) = g(x, y)\]
Adaptive Filter

\[ \hat{f}(x, y) = g(x, y) - \frac{\sigma^2}{\sigma^2_{S_{xy}}} \left[ g(x, y) - m S_{xy} \right] \]

\[ = g(x, y) * \left( \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{\sigma^2}{\sigma^2_{S_{xy}}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{mn} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) \]
FIGURE 5.13
(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. All filters were of size $7 \times 7$. 

Gaussian Noise + Adaptive Filter
Degradation Modeling
 Degradation Modeling

\[ g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \]

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]
Modeling Degradation Due to Atmospheric Turbulence

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

**FIGURE 5.25**
Illustration of the atmospheric turbulence model.
(a) Negligible turbulence.
(b) Severe turbulence, \( k = 0.0025 \).
(c) Mild turbulence, \( k = 0.001 \).
(d) Low turbulence, \( k = 0.00025 \).
(Original image courtesy of NASA.)
Modeling Uniform Linear Motion Blur

FIGURE 5.26
(a) Original image.
(b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$. 
Modeling Uniform Linear Motion Blur

T - duration of the exposure

blurred output:

\[ g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t)) \, dt \]
Modeling Uniform Linear Motion Blur

**T** - duration of the exposure

blurred output:

\[ g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t)) dt \]

\[ G(u, v) = F(u, v) \int_0^T e^{-2j\pi[u x_0(t) + v y_0(t)]} dt \]
Modeling Uniform Linear Motion Blur

T - duration of the exposure

blurred output:

\[ g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t)) \, dt \]

\[ G(u, v) = F(u, v) \int_0^T e^{-2j\pi [ux_0(t) + vy_0(t)]} \, dt \]

\[ \Rightarrow H(u, v) = \int_0^T e^{-2j\pi [ux_0(t) + vy_0(t)]} \, dt \]
Modeling Uniform Linear Motion Blur

\[ x_0(t) = \frac{at}{T}, \quad y_0(t) = \frac{bt}{T} \]

\[ H(u, v) = \int_0^T e^{-2j\pi[ux_0(t)+vy_0(t)]} dt \]

\[ = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua+ub)} \]
Image Restoration
Image Restoration by Inverse Filtering

**FIGURE 5.1**
A model of the image degradation/restoration process.

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]

\[ \hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \]
Image Restoration by Inverse Filtering

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]
Image Restoration by Inverse Filtering

\[ F(u, v) = H(u, v) F(u, v) + N(u, v) \]

\[ \hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)} \]
Inverse Filtering

\[ \hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)} \]

Bad news:

• Even when \( H(u,v) \) is known, there is always unknown noise

• Often \( H(u,v) \) has values close to zero
Example: Inverse Filtering

\[ H(u, v) = \exp \left\{ -k \left[ (u - M/2)^2 + (v - N/2)^2 \right]^{5/6} \right\} \]

Atmospheric turbulence effect
Example: Inverse Filtering

\[ \frac{G(u, v)}{H(u, v)} \]
Wiener Filtering = Mean Squared Error Filtering

• Incorporates both:
  • Degradation function
  • Statistical characteristics of noise

• Assumption: noise and the image are uncorrelated

• Optimizes the filter so that MSE is minimized

\[ e = \sum \sum (f(x, y) - \hat{f}(x, y))^2 \]
Wiener Filter — Derivation

\[ e = MN \sum_x \sum_y |f(x, y) - \hat{f}(x, y)|^2 \]
Wiener Filter — Derivation

unknown original \hspace{2cm} \text{after Wiener filtering}

\[ e = MN \sum_x \sum_y |f(x, y) - \hat{f}(x, y)|^2 \]

\[ = \sum_u \sum_v |F(u, v) - \hat{F}(u, v)|^2 \hspace{2cm} \text{Parseval’s Theorem} \]
Wiener Filter — Derivation

\[ e = MN \sum_x \sum_y |f(x, y) - \hat{f}(x, y)|^2 \]

\[ = \sum_u \sum_v |F(u, v) - \hat{F}(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v) - [F(u, v)H(u, v) + N(u, v)]W(u, v))|^2 \]

Unknown original

Corrupted original

Wiener filter
Wiener Filter — Derivation

independent signals

\[ e = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)] - N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)]|^2 + |N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)|^2|1 - H(u, v)W(u, v)|^2 + |N(u, v)|^2|W(u, v)|^2 \]
\[ e = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)] - N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)]|^2 + |N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)|^2|1 - H(u, v)W(u, v)|^2 + |N(u, v)|^2|W(u, v)|^2 \]

\[ \frac{\partial e}{\partial W(u, v)} = 0 \quad \Rightarrow \quad W(u, v) \]
Wiener Filter — Derivation

\[ \frac{\partial}{\partial z}(zz^*) = 2z^* \]

\[ e = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)] - N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)[1 - H(u, v)W(u, v)]|^2 + |N(u, v)W(u, v)|^2 \]

\[ = \sum_u \sum_v |F(u, v)|^2|1 - H(u, v)W(u, v)|^2 + |N(u, v)|^2|W(u, v)|^2 \]

\[ \frac{\partial e}{\partial W(u, v)} = |F|^2[2(1 - W^*H^*)(-H)] + |N|^2[2W^*] \]
Wiener Filter — Derivation

\[ \frac{\partial e}{\partial W(u, v)} = 0 \quad \Rightarrow \quad W^*(u, v) = \frac{|F(u, v)|^2 H(u, v)}{|H(u, v)|^2 |F(u, v)|^2 + |N(u, v)|^2} \]

\[ W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \frac{|N(u, v)|^2}{|F(u, v)|^2}} = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{|N(u, v)|^2}{|F(u, v)|^2}} \]
Wiener Filter — Derivation

\[ \frac{\partial e}{\partial W(u, v)} = 0 \quad \Rightarrow \quad W^*(u, v) = \frac{|F(u, v)|^2 H(u, v)}{|H(u, v)|^2 |F(u, v)|^2 + |N(u, v)|^2} \]

\[ W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}} = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}} \]

inverse filter
Wiener Filter -- Approximation

\[ W(u, v) = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{1}{\text{SNR}}} \]

Signal-to-noise ratio
Example: Wiener Filtering

**FIGURE 5.28** Comparison of inverse and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.
**Example: Wiener Filtering**

**FIGURE 5.29** (a) 8-bit image corrupted by motion blur 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 noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.
Next Class

- Image reconstruction from projections (Textbook 5.11)
- Radon Transform (Textbook 5.11.3)