Outline

- Recap of Lec 10
- Image restoration
  - Weiner Filter
  - K-SVD
- Summary
Image Restoration

- Image Restoration from Degradation

\[ f(x,y) \xrightarrow{\text{degradation}} g(x,y) \xrightarrow{\text{restoration}} \hat{f}(x,y) \]

\[ h(x,y) \quad n(x,y) \]

\[ g(x,y) = h(x,y) \ast f(x,y) + n(x,y) \iff G(u,v) = H(u,v) F(u,v) + N(u,v) \]

- Degradation sources:
  - Noise - independent of (x,y)
  - Point Spread Function (PSF) - a function of (x, y), and assuming linear
## Noise Suppression

- **Spatial Filtering**
  - Linear: Mean, Gaussian, and Media Filters
  - Non-Linear: Bilateral Filters/Guided Filters

- **Freq Domain Filtering**
  - Low Pass Filters
  - Band pass Filters
  - Notch filters for repetitive patterns

---

**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 $7 \times 7$ median filter. (c) Result of adaptive median filtering with $S_{max} = 7$. 

Z. Li, ECE 484 Digital Image Processing, 2018
Inverse Filtering

- Degradation from PSF

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


![Diagram of inverse filtering process](image-url)
Problem with Inverse filters

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

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

Estimate of original image

Problem: 0 or small values

Sol: limit the frequency around the origin

\( \hat{f}(x,y) \)

blur \( \sigma = 1.0 \) pixels
The Inverse Problem

Inverse filter with cut-off:
\[
\hat{H}(u, v) = \begin{cases} 
\frac{1}{H(u, v)}, & |D(u, v)| \leq \varepsilon \\
0, & |D(u, v)| > \varepsilon 
\end{cases}
\]

Pseudo-inverse filter:
\[
\hat{H}(u, v) = \begin{cases} 
\frac{1}{H(u, v)}, & |H(u, v)| \geq \varepsilon \\
0, & |H(u, v)| < \varepsilon 
\end{cases}
\]

- Can the filter take values between \(1/H(u,v)\) and zero?
- Can we model noise directly?
MIT RLE was the center of innovation in signal processing before, during and after WWII

Wiener filter

- goal: restoration with minimum mean-square error (MSE)
  \[ \min_W e^2 = E\{ (f - \hat{f})^2 \} \]
- optimal solution (nonlinear):
  \[ \hat{f}(x, y) = E\{ f(x, y) | g(m, n), \forall (m, n) \} \]
- restrict to linear space-invariant filter
  \[ \hat{f}(x, y) = w(x, y) * g(x, y) \]
- find “optimal” linear filter \( W(u,v) \) with min. MSE ...
Sketch of Wiener Filter Derivation

Proof:

Aim is to find filter which minimizes

$$\mathcal{E} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( f(x, y) - \hat{f}(x, y) \right)^2 \, dxdy$$

$$\mathcal{E} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| f(x, y) - \hat{f}(x, y) \right|^2 \, dxdy$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| F(u, v) - \hat{F}(u, v) \right|^2 \, dudv$$ \text{ Parseval's Theorem}$$\hat{F} = WG = WHF + WN$$

$$F - \hat{F} = (1 - WH)F - WN$$

$$\mathcal{E} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| (1 - WH)F - WN \right|^2 \, dudv$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ |(1 - WH)F|^2 + |WN|^2 \right\} \, dudv$$ \text{ since } f(x,y) \text{ and } \eta(x,y) \text{ uncorrelated}

- Note, integrand is sum of two squares
Sketch of Proof

- Take the differentiation

Minimize integral if integrand minimum for all \((u,v)\)

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

\[
\frac{\partial}{\partial z} \rightarrow 2 \left( -(1 - W^*H^*)H|F|^2 + W^*|N|^2 \right) = 0
\]

\[
W^* = \frac{H|F|^2}{|H|^2|F|^2 + |N|^2}
\]

\[
W = \frac{H^*}{|H|^2 + |N|^2/|F|^2}
\]

Note: filter is defined in the Fourier domain
Wiener filter

- Min MSE Filtering:
  - goal: restoration with minimum mean-square error (MSE)
    \[
    \min_W e^2 = E\{(f - \hat{f})^2\}
    \]
    \[
    \hat{f}(x, y) = w(x, y) \ast g(x, y)
    \]
  - find “optimal” linear filter \(W(u,v)\) with min. MSE
  - orthogonal condition \(E\{g(f - \hat{f})\} = 0\)
  - correlation function \(R_{fg}(x, y) = w(x, y) \ast R_{gg}(x, y)\)
  - wide-sense-stationary (WSS) signals
    \[
    R_{fg}(x_1, y_1, x_2, y_2) = E\{f(x_1, y_1)g(x_2, y_2)\} \xrightarrow{WSS} R_{fg}(x_1 - x_2, y_1 - y_2)
    \]
  - Fourier Transform: from correlation to spectrum
    \[
    S_{fg}(u, v) = \mathcal{F}\{R_{fg}(x, y)\}, \quad S_{gg}(u, v) = \mathcal{F}\{R_{gg}(x, y)\}
    \]
    \[
    W(u, v) = \frac{S_{fg}(u, v)}{S_{gg}(u, v)} = \frac{H^*(u, v)S_{ff}(u, v)}{|H(u, v)|^2S_{ff}(u, v) + S_{\eta\eta}(u, v)}
    \]
    \(S_{ff}\) and \(S_{\eta\eta}\) are the power spectra of the signal and noise, respectively
observations about Wiener filter

- If no noise, \( S_{\eta \eta} \to 0 \), it is a Pseudo Inv Filter:

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

\[
W(u, v)|_{S_{\eta \eta} \to 0} = \begin{cases} 
\frac{1}{H(u, v)}, & H(u, v) \neq 0 \\
0, & H(u, v) = 0
\end{cases}
\]

\( \Rightarrow \) Pseudo inverse filter

- If no blur, \( H(u, v) = 1 \) (Wiener smoothing filter)

\[
W(u, v)|_{H=1} = \frac{1}{1 + \frac{S_{\eta \eta}(u, v)}{S_{ff}(u, v)}} = \frac{SNR(u, v)}{SNR(u, v) + 1}
\]

\( \Rightarrow \) More suppression on noisier frequency bands
Wiener Filter

Wiener Filter implementation

1-D Wiener Filter Shape

Where $K$ is a constant (w.r.t. $u$ and $v$) chosen according to our knowledge of the noise level.
Wiener Filter example

\[ W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K} \]
Wiener filter example

- Wiener filter is more robust to noise, and preserves high-frequency details.

**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.
More deblurring with Wiener filter

Example 1: Focus deblurring with a Wiener filter

\[ \hat{F}(u,v) = W(u,v) \cdot G(u,v) \]

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

\[ \begin{align*}
g(x,y) & \quad | \quad \hat{f}(x,y) \\
\text{blur } \sigma = 1.5 \text{ pixels} & \quad \text{noise } \sigma = 0.3 \text{ grey levels} & \\
K = 1.0 \times 10^{-5} & \quad K = 1.0 \times 10^{-3} & \quad K = 1.0 \times 10^{-1}
\end{align*} \]
Deblurring

- Recovering from heavier blur
  - blur PSF sigma = 3.0 pixels
  - Gaussian noise sigma = 0.3.

\[ f(x,y) \quad g(x,y) \quad \hat{f}(x,y) \]

\[ K = 5.0 \times 10^{-4} \]
Wiener filter: when does it not work?

- Operating Wiener Filter....
  
  How much de-blurring is just enough?

[Image Analysis Course, TU-Delft]
Consider the horizontal motion

**Example 2: Motion deblurring**

Suppose there is blur only in the horizontal direction, e.g., camera pans as image is acquired.

**Degradation model**

\[ g(x, y) = \frac{1}{T} \int_{-T/2}^{T/2} f(x - x_0(t), y) \, dt \]

Require \( H(u, v) \) for Wiener filter.
Motion Blur

- Observed blurred images

\[
G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux + vy)} \, dx \, dy,
\]
\[
= \frac{1}{T} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ \int_{-T/2}^{T/2} f(x - x_0(t), y) \, dt \right\} e^{-j2\pi(ux + vy)} \, dx \, dy
\]

Interchange order of spatial and temporal integration

\[
G(u, v) = \frac{1}{T} \int_{-T/2}^{T/2} \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - x_0(t), y) e^{-j2\pi(ux + vy)} \, dx \, dy \right\} \, dt
\]

Fourier transform of \( f(x - x_0(t), y) \).

\[
G(u, v) = \frac{1}{T} \int_{-T/2}^{T/2} F(u,v) e^{-j2\pi u x_0(t)} \, dt
\]
\[
= F(u, v) \frac{1}{T} \int_{-T/2}^{T/2} e^{-j2\pi u x_0(t)} \, dt
\]
\[
= F(u, v) H(u, v)
\]

where

\[
H(u, v) = \frac{1}{T} \int_{-T/2}^{T/2} e^{-j2\pi u x_0(t)} \, dt
\]

Suppose \( x_0(t) = st \), and \( sT = d \) pixels

\[
H(u, v) = \frac{1}{T} \int_{-T/2}^{T/2} e^{-j2\pi ust} \, dt
\]
\[
= \frac{1}{T} \left[ e^{-j2\pi ust} \right]_{-T/2}^{T/2}
\]
\[
= \frac{1}{j2\pi u} (e^{j\pi ud} - e^{-j\pi ud})
\]

\[
= \text{sinc}\pi ud
\]

\( H(u, v) \) has large area of zeros, a problem for Inverse Filters

\[ h(x, y) = \hat{h}_d(x) \delta(y) \]

FT of …
Motion Blur

- Recover motion blur from Wiener Filter
  - K need to be hand crafted.

### Motion deblurring with a Wiener filter

**blur = 20 pixels**

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

1. Compute the FT of the blurred image
2. Multiply the FT by the Wiener filter \( \hat{F}(u,v) = W(u,v) G(u,v) \)
3. Compute the inverse FT
Motion Blur Wiener Filtering Summary

Algorithm

1. Rotate image so that blur is horizontal
2. Estimate length of blur
3. Construct a bar modelling the convolution
4. Compute and apply a Wiener filter
5. Optimize over values of K

![Images of f(x,y), h(x,y), and \( \hat{f}(x,y) \)]

blur = 30 pixels
So far we assume the PSF is known or can be estimated

\[ g = A(h) f \]
\[ G = H F \]

i.e. that \( h(x, y) \) is known, so that given the observed image \( g(x, y) \), then the original image \( f(x, y) \) can be estimated (restored)

- What if \( h(x, y) \), or \( H \) is not known?
- Only \( g(x, y) \) is known.

Blind Deblurring
Blind Deblurring

- Estimating both $h(x, y)$ and $f(x, y)$ from observed $g(x, y)$, by solving the following optimization:

$$\min_{f,h} (g - A(h) f)^2 + \lambda p_f (f) + \mu p_h (h)$$

- Likelihood/loss function
- Image prior
- Blur prior

- Example image prior: suppressing high freq noise

$$p(f) = (\nabla f)^2$$

- Having iterative solutions, details in Ref [18], with source code.
Example from blind deblurring
**Summary**

- **Wiener Filtering**
  - Solving for a MSE objective function, that has freq domain solution
  - Basically inverse filter but reflect the Signal to Noise ratios at freq locations
  - Applications in deblurring, and motion deblurring