In this lecture we will look at linear image restoration techniques

– Differentiation of matrices and vectors
– Linear space invariant degradation
– Restoration in absence of noise
  • Inverse filter
  • Pseudo-inverse filter
– Restoration in presence of noise
  • Inverse filter
  • Wiener filter
  • Constrained least squares filter

1. Differentiation of Matrices and Vectors
   Notation:
   $A$ is a $M \times N$ matrix with elements $a_{ij}$.
   $x$ is a $N \times 1$ vector with elements $x_i$.
   $f(x)$ is a scalar function of vector $x$.
   $g(x)$ is a $M \times 1$ vector valued function of vector $x$.

2. Differentiation of Matrices and Vectors (cont...)
   Scalar derivative of a matrix.
   $A$ is a $M \times N$ matrix with elements $a_{ij}$.
   \[
   \frac{\partial A}{\partial t} = \begin{pmatrix}
   \frac{\partial a_{11}}{\partial t} & \cdots & \frac{\partial a_{1N}}{\partial t} \\
   \vdots & \ddots & \vdots \\
   \frac{\partial a_{M1}}{\partial t} & \cdots & \frac{\partial a_{MN}}{\partial t}
   \end{pmatrix}
   \]

3. Differentiation of Matrices and Vectors (cont...)
   Vector derivative of a function (gradient).
   $x$ is a $N \times 1$ vector with elements $x_i$.
   $f(x)$ is a scalar function of vector $x$.
   \[
   \frac{\partial f}{\partial x} = \nabla_x f = \left( \frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_N} \right)^T
   \]

4. Differentiation of Matrices and Vectors (cont...)
   Vector derivative of a vector (Jacobian):
   $x$ is a $N \times 1$ vector with elements $x_i$.
   $g(x)$ is a $M \times 1$ vector valued function of vector $x$.
   \[
   \frac{\partial g}{\partial x} = \begin{pmatrix}
   \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_N} \\
   \vdots & \ddots & \vdots \\
   \frac{\partial g_M}{\partial x_1} & \cdots & \frac{\partial g_M}{\partial x_N}
   \end{pmatrix}
   \]
Some useful derivatives.
\( x \) is a \( N \times 1 \) vector with elements \( x_i \).
\( b \) is a \( N \times 1 \) vector with elements \( b_i \).
\( C. \) Nikou – Digital Image Processing (E12)

It is the derivative of the scalar valued function \( b^T x \) with respect to vector \( x \).

\[ \frac{\partial}{\partial x} (b^T x) = b \]

It may be proved using the previous properties.

In the Fourier domain:
\[ G(k, l) = H(k, l)F(k, l) + N(k, l) \]

where multiplication is element-wise.

In matrix-vector form:
\[ g = Hf + \eta \]
where \( H \) is a doubly block circulant matrix and \( f, g, \) and \( \eta \) are vectors (lexicographic ordering).

We now consider a degraded image to be modelled by:
\[ g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \]

where \( h(x, y) \) is the impulse response of the degradation function (i.e. point spread function blurring the image).
The convolution implies that the degradation mechanism is linear and position invariant (it depends only on image values and not on location).

If the degradation function is unknown the problem of simultaneously recovering \( f(x,y) \) and \( h(x,y) \) is called blind deconvolution.
Linear Restoration

Using the imaging system
\[ g = Hf + \eta \]
we want to estimate the true image from the degraded observation with known degradation \( H \).

A linear method applies an operator (a matrix) \( P \) to the observation \( g \) to estimate the unobserved noise-free image \( f \):
\[ \hat{f} = Pg \]

Restoration in Absence of Noise

The Inverse Filter

When there is no noise:
\[ g = Hf \]
an obvious solution would be to use the inverse filter:
\[ P = H^{-1} \]
yielding
\[ \hat{f} = Pg = H^{-1}g = H^{-1}Hf = f \]

The Inverse Filter (cont...)

For a \( N \times N \) image, \( H \) is a \( N^2 \times N^2 \) matrix!

To tackle the problem we transform it to the Fourier domain.
\( H \) is doubly block circulant and therefore it may be diagonalized by the 2D DFT matrix \( W \):
\[ H = W^{-1}AW \]

Restoration in Absence of Noise

The Inverse Filter (cont...)

Which is the vectorized form of the DFT of the image:
\[ \hat{F} = \Lambda^{-1}G \quad \Leftrightarrow \quad \hat{F}(k,l) = \frac{G(k,l)}{H(k,l)} \]

Take the inverse DFT and obtain \( f(m,n) \).
Problem: what happens if \( H(k,l) \) has zero values?
Cannot perform inverse filtering!

The Pseudo-inverse Filter

A solution is to set:
\[ \hat{F}(k,l) = \begin{cases} 
\frac{G(k,l)}{H(k,l)}, & H(k,l) \neq 0 \\
0, & H(k,l) = 0 
\end{cases} \]
which is a type of pseudo-inversion.
Notice that the signal cannot be restored at locations where \( H(k,l)=0 \).
A pseudo-inverse filter also arises by the unconstrained least squares approach.

Find the image \( f \), that, when it is blurred by \( H \), it will provide an observation as close as possible to \( g \), i.e., it minimizes the distance between \( Hf \) and \( g \). This distance is expressed by the norm:

\[
\min_{f} \left\| Hf - g \right\|^2
\]

\[
\Rightarrow \frac{\partial J}{\partial f} = 0 \quad \Rightarrow \frac{\partial}{\partial f} \left( \left\| Hf - g \right\|^2 \right) = 0
\]

\[
\Rightarrow 2H^T (Hf - g) = 0 \quad \Rightarrow 2H^T Hf = 2H^T g
\]

\[
\Rightarrow f = (H^T H)^{-1} H^T g
\]

Recall the imaging model with spatially invariant degradation and noise

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

\[
G(k,l) = H(k,l)F(k,l) + N(k,l)
\]

\[
g = Hf + \eta
\]

One approach to get around the problem is to limit the ratio \( G(k,l) / H(k,l) \) to frequencies near the origin that have lower probability of being zero.

We know that \( H(0,0) \) is usually the highest value of the DFT.

Thus, by limiting the analysis to frequencies near the origin we reduce the probability of encountering zero values.

Blurring degradation
Restoration in Presence of Noise
The Inverse Filter (cont...)

Inverse filter with cut-off

C. Nikou – Digital Image Processing (E12)

Restoration in Presence of Noise
Wiener Filter

So far we assumed nothing about the statistical properties of the image and noise. We now consider image and noise as random variables and the objective is to find an estimate of the uncorrupted image \( \hat{f} \) such that the mean square error between the estimate and the image is minimized:

\[
\min_{\hat{f}} \left( E \left[ (f - \hat{f})^2 \right] \right)
\]

where \( E[x] \) is the expected value of vector \( x \).

Recall also the definition of the correlation matrix between two vectors \( x \) and \( y \):

\[
R_{xy} = E[xy^T] = \begin{bmatrix}
E[x_1y_1] & E[x_1y_2] & \cdots & E[x_1y_n] \\
E[x_2y_1] & E[x_2y_2] & \cdots & E[x_2y_n] \\
\vdots & \vdots & \ddots & \vdots \\
E[x_ny_1] & E[x_ny_2] & \cdots & E[x_ny_n]
\end{bmatrix}
\]

We assume that the image and the noise are uncorrelated:

\[
R_{x} = R_{n} = 0
\]

We are looking for the best estimate:

\[
\min_{\hat{f}} \left( E \left[ (f - \hat{f})^2 \right] \right)
\]

Let’s confine our estimate to be obtainable by a linear operator on the observation:

\[
\hat{f} = Pg
\]

and the goal is to find the best matrix \( P \).

Denoting by \( p_n^r \) the \( n \)-th row of \( P \):

\[
J(\hat{f}) = E \left[ (f - \hat{f})^2 \right] = E \left[ \|f - \hat{f}\|^2 \right] = E \left[ \|f - Pg\|^2 \right]
\]

Denoting by \( p_n^r \) the \( n \)-th row of \( P \):

\[
J(\hat{f}) = E \left[ \sum_n (f_n - p_n^r g_n)^2 \right] = \sum_n E \left[ (f_n - p_n^r g_n)^2 \right]
\]
We can now minimize the sum with respect to each term:

\[ \frac{\partial}{\partial p_n} \left( R_{f\ell} - 2p_n^T R_{p\ell} + p_n^T R_{pp} \right) = 0 \]

\[ \Leftrightarrow -2R_{p\ell} + 2R_{pp} p_n = 0 \Leftrightarrow p_n = R_{pp}^{-1} R_{p\ell} \]

\[ p_n^* = R_{pp}^{-1} R_{p\ell} \]

Assembling the rows together:

\[ P = R_{gg} R_{gg}^{-1} \]

We have to compute the two matrices:

\[ R_{gg} = E \left[ gg^T \right] = E \left[ (Hf + \eta)(Hf + \eta)^T \right] \]

\[ = E \left[ Hff^T + Hf\eta^T + \eta f^T H^T + \eta^T \eta \right] \]

\[ = HR_{gg} H^T + HR_{gg} + R_{gg} H^T + R_{gg} \]

Assuming noise is uncorrelated with image:

\[ R_{gg} = HR_{ff} H^T + R_{gg} \]

Also,

\[ R_{gg} = E \left[ gg^T \right] = E \left[ f(f + \eta)^T \right] = ... = R_{gg} H^T \]

Finally the matrix we are looking for is

\[ P = R_{gg} R_{gg}^{-1} = R_{gg} H^T \left( HR_{gg} H^T + R_{gg} \right)^{-1} \]

The estimated uncorrupted image is

\[ \hat{f} = R_{gg} H^T \left( HR_{gg} H^T + R_{gg} \right)^{-1} g \]

which may be also expressed as

\[ \hat{f} = \left( H^T R_{gg}^{-1} H + R_{gg} \right)^{-1} H^T R_{gg}^{-1} g \]

This result is known as the Wiener filter or the minimum mean square error (MMSE) filter.

Special cases:
No blur \((H=I, g=f+\eta)\): \[ \hat{f} = R_{gg}^{-1} \left( R_{gg} + R_{gg} \right)^{-1} g \]

No noise \((R_{gg}=0, g=Hf)\): \[ \hat{f} = H^{-1} g \]

This is the inverse filter.

No blur, no noise \((H=I, R_{gg}=0)\): \[ \hat{f} = g \]

Do nothing on the observation.

The size of the matrix to be inverted poses difficulties and Wiener filter is implemented in the Fourier domain.

This occurs when \(H\) is doubly block circulant (represents convolution) and the image \(f\) and noise \(\eta\) are wide-sense stationary (w.s.s).

Definition of a w.s.s. signal:
1) \(E[f(m,n)]=\mu\), independent of \(m,n\).
2) \(E[f(m,n)f(k,l)]=r(m-k,n-l)\), independent of location.
Reminder: the inverse DFT complex exponential matrix diagonalizes any circulant matrix:

\[ H = W^{-1}A_nW \]

The columns of \( W^{-1} \) are the eigenvectors of any circulant matrix \( H \).

The corresponding eigenvalues are the DFT values of the signal producing the circulant matrix.

Remember also that \( W^T = W \)

and that \( (W^{-1})^T = W^{-1} \)

We will employ the following relations:

\[ f_1 = HHW \Lambda W \]

\[ f_1^* = (HHW \Lambda W)^* \]

\[ f_1 = (NW^{-1})A_n^* W = W^{-1}A_n^* W \]

\[ f_1^* = (NW^{-1})^* A_n^* W = W^{-1}A_n^* W \]

The Wiener solution is now transformed to the Fourier domain:

\[ \hat{f} = R_n H^{-1}(HR_n H^{-1} + R_{\eta})^{-1} g \]

\[ \Rightarrow \hat{f} = A_n^* (A_n^* A_n + A_{\eta})^{-1} W g \]

Notice that the matrices are diagonal.

If \( S_f(k,l) \) is not zero we may define the Signal to Noise Ratio in the frequency domain:

\[ SNR(k,l) = \frac{S_f(k,l)}{S_{\eta}(k,l)} \]

and the Wiener filter becomes:

\[ F(k,l) = \frac{H^*(k,l)}{|H(k,l)|^2 + SNR^{-1}(k,l)} G(k,l) \]

A well known estimate of \( S_f(k,l) \) is the periodogram (the ML estimate of \( R_f \) when \( f \) is assumed Gaussian):

\[ S_f(k,l) = \frac{1}{MN}|F(k,l)|^2 \]

In practice, as \( F(k,l) \) is unknown, we use

\[ \hat{S}_f(k,l) = \frac{1}{MN}|\hat{G}(k,l)|^2 \]
Restoration in Presence of Noise

Wiener Filter (cont...)  

When we do not have information on the power spectra the Wiener filter is not optimal. Another idea is to introduce a smoothness term in our criterion. We define smoothness by the quantity \( \|Qf\|^2 \) where \( Q \) is a high pass filter operator, e.g. the Laplacian, \( (Q \) is a doubly block circulant matrix representing convolution by the Laplacian). We look for smooth solutions minimizing \( \|Qf\|^2 \).

Restoration in Presence of Noise
Constrained Least Squares Filter

We have the following constrained least squares (CLS) optimization problem:

\[
\text{Minimize } \|Qf\|^2 \quad \text{subject to } \quad Hf = g
\]

yielding the Lagrange multiplier minimization of the function:

\[
J(f, \lambda) = \|Hg - Hf\|^2 + \lambda \|Qf\|^2
\]

Parameter \( \lambda \) controls the degree of smoothness:

\( \lambda = 0 \), \( f = (H^TH + \lambda Q^TQ)^{-1}H^Tg \) pseudo-inverse, ultra rough solution

\( \lambda \to \infty \), \( f = 0 \) ultra smooth solution

In the Fourier domain, the constrained least squares filter becomes:

\[
F(k, l) = \frac{H^*(k, l)}{|H(k, l)|^2 + \lambda |Q(k, l)|^2} G(k, l)
\]

Keep always in mind to zero-pad the images properly.
Restoration in Presence of Noise
Constrained Least Squares Filter (cont...)  

Low noise: Wiener and CLS generate equal results.  
High noise: CLS outperforms Wiener if $\lambda$ is properly selected.  
It is easier to select the scalar value for $\lambda$ than to approximate the SNR which is seldom constant.

Restoration Performance Measures

Mean square error (MSE):

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left[ f(m,n) - \hat{f}(m,n) \right]^2$$

The Signal to Noise Ratio (SNR) considers the difference between the two images as noise:

$$SNR = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left[ f(m,n) - \hat{f}(m,n) \right]^2}$$