MMAT5390: Mathematical Image Processing

Chapter 4: Image Enhancement in the Frequency Domain

Image enhancement refers to the process, by which we improve the image so that it looks subjectively better. We do not really know what the image should look like, but we can tell whether it has been improved or not, by considering, for example, whether more detail can be seen, or whether unwanted flickering has been removed, or the contrast is better.

Image enhancement is broadly defined. It can be referred to as different imaging tasks, which include image denoising, image deblurring, image sharpening and so on. In this chapter, some image enhancement tasks will be discussed.

Image denoising aims to restore an image corrupted by noise. Below is a simple example.

The image on the left is a noisy image. The image on the right is the restored image. The noises are removed using a mathematical model (which we are going to learn in this chapter).

Image deblurring aims to restore an image, whose features are blurry due to various factors (e.g. motion, atmospheric turbulence). Here is a simple example of motion blur.

In the above figure, the left shows a blurry image caused by motion (e.g. taking the photo on a moving car). The right shows the restored image using a mathematical model (which we are going to learn in this chapter).

Image enhancement can either be done in the spatial domain or the frequency domain. In this chapter, we will firstly look at how image enhancement can be done in the frequency domain.

Simply speaking, image processing in the frequency domain is done as follows. The image is firstly transformed using discrete Fourier transform (which can be computed efficiently using fast Fourier transform (FFT)). The Fourier coefficients are then adjusted according to different imaging tasks. The restored image can be obtained by taking the inverse Fourier transform (which can again be computed efficiently using FFT). In fact, instead of using DFT, wavelet transform (such as Haar transform) can also be applied.

1 Image denoising in the frequency domain

We will first discuss image denoising algorithms in the frequency domain.

1.1 What is noise?

Before we talk about the denoising algorithm, we need to understand what image noise is. Some basic statistical definitions are necessary.

Definition 1.1. A **random variable** is the value assigned to an outcome of a random experiment.

A **random field** is a spatial function that assigns a random variable to each spatial position.

A **distribution function** of a random variable f is defined as:

$$P_f(z) = P(f < z)$$

In other words, it is the probability that $f < z$

The **probability density function** of f is defined as:

$$p_f(z) = \frac{d}{dz} P_f(z)$$

The **expected or mean value** of f is defined as:

$$\mu_f = E(f) = \int_{-\infty}^{+\infty} z p_z(z) \ dz$$

The **variance** of f is defined as:

$$\sigma_f^2 = E\left((f - \mu_f)^2\right) = \int_{-\infty}^{+\infty} (z - \mu_f)^2 p_f(z) \ dz$$

σ_f is called the **standard deviation** of f.

Remark. $\displaystyle\int_{-\infty}^{+\infty} p_f(z) \ dz = 1.$

Definition 1.2. Suppose we have n random variables $f_1, f_2, \cdots, f_n$. The **joint distribution function** is defined as:

$$P_{f_1, f_2, ..., f_n}(z_1, z_2, ..., z_n) = P(f_1 < z_1, f_2 < z_2, ..., f_n < z_n).$$

In other words, it is the probability that $f_1 < z_1, f_2 < z_2, ..., f_n < z_n$.

The **joint probability density function** is defined as:

$$p_{f_1, f_2, \cdots, f_n}(z_1, z_2, \cdots, z_n) = \frac{\partial^n P_{f_1, f_2, \cdots, f_n}(z_1, z_2, \cdots, z_n)}{\partial z_1 \partial z_2 \cdots \partial z_n}$$

$f_1, f_2, \cdots, f_n$ are **independent** if:

$$P_{f_1, f_2, \cdots, f_n}(z_1, z_2, \cdots, z_n) = P_{f_1}(z_1) P_{f_2}(z_2) \cdots P_{f_n}(z_n)$$

They are called **uncorrelated** if:

$$E(f_i f_j) = E(f_i) E(f_j) \qquad \text{for all } i, j \text{ with } i \neq j$$

Any two random variables f_i and f_j are **orthogonal** if:

$$E(f_i f_j) = 0$$

The **covariance** of two random variables f_i and f_j is:

$$C_{ij} \equiv E\left(\left(f_i - \mu_{f_i}\right)\left(f_j - \mu_{f_j}\right)\right)$$

Example 1.3. Show that if the covariance of two random variables is zero, the two variables are uncorrelated.

Solution.

$$\begin{aligned} 0 = C_{ij} &= E\left(f_i f_j - \mu_{f_i} f_j - \mu_{f_j} f_i + \mu_{f_i}\mu_{f_j}\right) \\ &= E(f_i f_j) - \mu_{f_i} \underbrace{E(f_j)}_{\mu_{f_j}} - \mu_{f_j} \underbrace{E(f_i)}_{\mu_{f_i}} + \mu_{f_i}\mu_{f_j} \\ &= E(f_i f_j) - \mu_{f_i}\mu_{f_j} \end{aligned}$$

$\therefore E(f_i f_j) = E(f_i)E(f_j)$. (uncorrelated)

Example 1.4. If f_i and f_j are independent, then they are uncorrelated.

Solution.

$$\begin{aligned} \mu_{f_i f_j} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xy p_{f_i f_j}(x,y)\ dx\ dy \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xy p_{f_i}(x) p_{f_j}(y)\ dx\ dy \qquad \text{(check)} \\ &= \int_{-\infty}^{+\infty} x p_{f_i}(x)\ dx \int_{-\infty}^{+\infty} y p_{f_j}(y)\ dy\ = \mu_{f_i}\mu_{f_j} \end{aligned}$$

<u>Noise as a random field</u>

A **noise** of an image can be considered as a 2D random field $f(\vec{r};\omega_i)$ where $\vec{r}$ represents the spatial location and ω_i represents the random outcome. (In other words, it assigns a random variable to each spatial location).

For a fixed $\vec{r}$, a random field becomes a random variable.

Definition 1.5. Let f be a noise considered as a 2D random variable f.

The **auto-correlation** is defined as:

$$R_{ff}(\vec{r}_1,\vec{r}_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} z_1 z_2 p_f(z_1,z_2;\vec{r}_1,\vec{r}_2)\ dz_1\ dz_2$$

The **auto-covariance** is defined as:

$$C_{ff}(\vec{r}_1,\vec{r}_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (z_1 - \mu_f(\vec{r}_1))(z_2 - \mu_f(\vec{r}_2)) p_f(z_1,z_2;\vec{r}_1,\vec{r}_2)\ dz_1\ dz_2$$

where $\mu_f(\vec{r}) = \displaystyle\int_{-\infty}^{+\infty} z p_f(z;\vec{r})\ dz$.

Example 1.6. Consider the noise of a 2×2 image consisting of the following 4 outcomes (with uniform probability):

$$\begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 5 \\ 4 & 6 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 4 & 4 \end{pmatrix}, \begin{pmatrix} 6 & 4 \\ 4 & 2 \end{pmatrix}$$

Find the mean, anto-correlation and auto-covariance of noise.

Solution. Mean: $\begin{pmatrix} 2.5 & 3.5 \\ 3.75 & 3.5 \end{pmatrix}$

$R_{ff}((1,1),(1,2)) = \dfrac{1\cdot 3 + 2\cdot 5 + 1\cdot 2 + 6\cdot 4}{4} = 9.75$ etc.

$C_{ff}((1,1),(1,2)) = \dfrac{1}{4}[(1-2.5)\cdot(3-3.5)+(2-2.5)\cdot(5-3.5)+(1-2.5)\cdot(2-3.5)+(6-2.5)\cdot(4-3.5)] = 1$ etc.

Definition 1.7. Given two random fields f and g (two series of noise / images generated by two different underlying random experiments)

The **cross correlation** is defined as:

$$R_{fg}(\vec{r}_1, \vec{r}_2) = E(f(\vec{r}_1;\omega_i)g(\vec{r}_2;\omega_j))$$

The **cross covariance** is defined as:

$$C_{fg}(\vec{r}_1, \vec{r}_2) = E\left([f(\vec{r}_1;\omega_i) - \mu_f(\vec{r}_1)]\,[g(\vec{r}_2;\omega_j) - \mu_g(\vec{r}_2)]\right)$$

f and g are called **uncorrelated** if for all $\vec{r}_1$ and $\vec{r}_2$,

$$C_{fg}(\vec{r}_1, \vec{r}_2) = 0.$$

Equivalently, it means:

$$E(f(\vec{r}_1;\omega_i)g(\vec{r}_2;\omega_j)) = E(f(\vec{r}_1;\omega_i))E(g(\vec{r}_2;\omega_j))$$

Example 1.8. Given two noises f and g. Suppose the possible outcomes of f and g (with uniform probability) are given by:

$$\left(\underbrace{\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}}_{f}, \underbrace{\begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}}_{g}\right), \left(\underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{f}, \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}}_{g}\right), \left(\underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}}_{f}, \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}_{g}\right)$$

Compute the cross correlation of f and g.

Solution. Mean of $f = \begin{pmatrix} 1/3 & 1 \\ 1 & 4/3 \end{pmatrix}$; Mean of $g = \begin{pmatrix} 2/3 & 2/3 \\ 1/3 & 2/3 \end{pmatrix}$

$\therefore C_{fg}((1,1),(2,2)) = \frac{1}{3}\left(\left(1-\frac{1}{3}\right)\left(1-\frac{2}{3}\right) + \left(0-\frac{1}{3}\right)\left(0-\frac{2}{3}\right) + \left(0-\frac{1}{3}\right)\left(1-\frac{2}{3}\right)\right)$ etc.

Types of noises

Impulse noise / shot noise

Change the value of an image pixel at random. The randomness may follow a distribution, such as the Poisson distribution:

$$p(k) = \frac{e^{-\lambda}\lambda^k}{k!}.$$

That is, the probability of having k pixels affected by the noise in a window of certain size, where λ is the average number of affected pixel in a window of a fixed size.

Gaussian noise

Noise at each pixel follows Gaussian probability density function:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}}\, exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Additive noise

Noisy Image = original image + noise

Multiplicative noise

Noisy image = original image $\times$ noise

Homogeneous noise

Noise parameters for the probability density function at each pixel are the same.

Zero-mean noise

Mean value of noise is zero ($\mu(i,j) = 0$).

Biased noise

The mean values of noise at some pixels are non-zero ($\mu(i,j) \neq 0$ for some i,j).

Independent noise

Given n pixels, the noise at each pixel (as random variable) are independent.

Uncorrelated noise

$E(X_i X_j) = E(X_i)E(X_j)$ for all i,j, where X_i is the noise at pixel i as a random variable.

White noise

Zero mean, uncorrelated and additive.

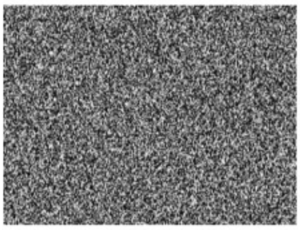
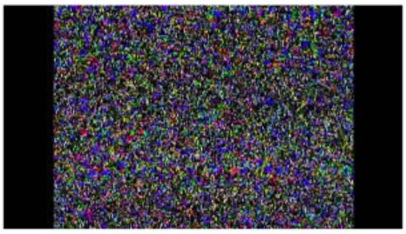
iid noise

Independent + Identically distributed.

Examples of noises

Examples of Gaussian noises and white noises are given below.


The above shows two images corrupted by Gaussian noises.


The above shows some examples of images corrupted by white noises.

Property of white noise

Intuitive idea: Suppose the white noise f is defined over a continuous domain $\mathbb{R}^2$. Furthermore, assume that f is non-zero only over $[0,1]\times[0,1]$ (that is, $f=0$ over $\mathbb{R}^2 \setminus [0,1]\times[0,1]$.

Consider

$$\hat{f}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j(xu+yv)}\ dx\ dy.$$

Note that:

$$\hat{f}(u,v) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m/N,n/N)e^{-j(\frac{mu+nv}{N})}((\frac{1}{N})(\frac{1}{N})),$$

which is similar to the discrete Fourier transform of the discretized signal of f.

Since f is a white noise, f is uncorrelated and has a zero mean at every pixel. Consider:

$$A(\tilde{x},\tilde{y}) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x+\tilde{x},y+\tilde{y})f(x,y))\ dx\ dy.$$

In most circumstances, $A(\tilde{x},\tilde{y})$ looks likes:

$$f(\tilde{x},\tilde{y}) \approx \begin{cases} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)^2\ dx\ dy & \text{if } (\tilde{x},\tilde{y}) = 0 \\ 0 & \text{if } (\tilde{x},\tilde{y}) \neq 0 \end{cases} = c\delta(\tilde{x},\tilde{y})$$

for f comes from a uncorrelated random field of zero mean.

On the other hand, let $s_1 = x + \tilde{x}, s_2 = y + \tilde{y}$, then:

$$
\begin{aligned}
\widehat{A}(u,v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(s_1,s_2)f(x,y)e^{-j((s_1-x)u+(s_2-y)v)}\ dx\ dy\ ds_1\ ds_2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(s_1,s_2)e^{-j(s_1u+s_2v)}\ ds_1\ ds_2\ \overline{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j(xu+yv)}\ dx\ dy} \\
&= \widehat{f(\cdot,\cdot)}(u,v)\overline{\widehat{f(\cdot,\cdot)}(u,v)} \\
&= \left|\widehat{f(\cdot,\cdot)}(u,v)\right|^2
\end{aligned}
$$

We can check that the Fourier transform of $c\delta$ = constant function. Therefore,

$$
\widehat{A}(u,v) = \left|\widehat{f(\cdot,\cdot)}(u,v)\right|^2 \approx \text{ constant for all } u,v
$$

Hence, the Fourier transform of a white noise has approximately constant magnitude.

1.2 Image denoising by low-pass filtering

Our goal in this subsection is as follows.

Goal:

1. Remove high frequency component (low pass filter) for image de-noising.
2. Remove low frequency component (high pass filter) for extraction of image details.

In order to denoise an image in the frequency domain, we need to identify the high/low frequency components of the DFT of f.

Let F be a $N \times N$ image, where N is even. Let $\hat{F}$ be the DFT of F.

High / Low frequency components of $\hat{F}$

Recall:

$$
\hat{F}(k,l) = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)e^{-j\frac{2\pi}{N}(mk+nl)}
$$

$\hat{F}(k,l)$ is called the Fourier coefficient at (k,l).

Observe that: for $0 \le k,l \le \dfrac{N}{2} - 1$

$$
\begin{aligned}
\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right) &= \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)e^{-j\frac{2\pi}{N}(m(\frac{N}{2}+k)+n(\frac{N}{2}+l))} \\
&= \frac{1}{N^2}\overline{\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)(-1)^{m+n}e^{-j\frac{2\pi}{N}(m(-k)+n(-l))}} \\
&= \frac{1}{N^2}\overline{\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)e^{-j\frac{2\pi}{N}(m(\frac{N}{2}-k)+n(\frac{N}{2}-l))}} \\
&= \overline{\hat{F}\left(\frac{N}{2}-k, \frac{N}{2}-l\right)}
\end{aligned}
$$

Remark. *Computing part of $\hat{F}$ can determine the rest!*

From above, we deduce that:

$$
\begin{aligned}
F(m,n) = & \sum_{0\le k,l\le \frac{N}{2}-1} \left[\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+k)m+(\frac{N}{2}+l)n]}\right] \\
& + \sum_{1\le k,l\le \frac{N}{2}-1} \left[\overline{\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-k)m+(\frac{N}{2}-l)n]}\right] \\
& + \sum_{0\le k,l\le \frac{N}{2}-1} \left[\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}-l\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+k)m+(\frac{N}{2}-l)n]}\right] \\
& + \sum_{1\le k,l\le \frac{N}{2}-1} \left[\overline{\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}-l\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-k)m+(\frac{N}{2}+l)n]}\right] \\
& + \sum_{0\le l\le \frac{N}{2}-1} \hat{F}\left(0, \frac{N}{2}+l\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+l)n]} + \sum_{1\le l\le \frac{N}{2}-1} \overline{\hat{F}\left(0, \frac{N}{2}+l\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-l)n]} \\
& + \sum_{0\le k\le \frac{N}{2}-1} \hat{F}\left(\frac{N}{2}+k, 0\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+k)m]} + \sum_{1\le k\le \frac{N}{2}-1} \overline{\hat{F}\left(\frac{N}{2}+k, 0\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-k)m]} + \hat{F}(0,0)
\end{aligned}
$$

Observations:

- When k and l are close to $\frac{N}{2}$, $\hat{F}\left(\underbrace{\frac{N}{2}+k}_{\approx N}, \underbrace{\frac{N}{2}+l}_{\approx N}\right)$ is associated to $\underbrace{e^{-j\frac{2\pi}{N}(k'm+l'n)}}_{\approx e^{-j\frac{2\pi}{N}(\tilde{k}m+\tilde{l}n)}}$, where k' and l' are close to N and $\tilde{k}$ and $\tilde{l}$ are close to 0.

  Therefore, Fourier coefficients at bottom right corner are associated to low frequency components.

- Similarly, we can check Fourier coefficients at the four corners are associated to low frequency components.

- On the other hand, Fourier coefficients in the middle are associated to high frequency components.

Remark.

- *High pass filtering = Remove Fourier coefficients at the four corners*
- *Low pass filtering = Remove Fourier coefficients in the middle.*

Centralization of Frequency domain

Let $F(x,y)$ be a $N\times N$ image with $0\le x\le N-1, 0\le y\le N-1$
Let $\hat{F}(u,v)$ be the DFT of F with $0\le u\le N-1, 0\le v\le N-1$

Note that:

- High-frequency components are located near $\left(\frac{N}{2}, \frac{N}{2}\right)$.
- Low-frequency components are located near 4 corners.

So, if we let $\tilde{F}(u,v) = \hat{F}\left(u-\frac{N}{2}, v-\frac{N}{2}\right)$ where $0\le u\le N-1, 0\le v\le N-1$, then

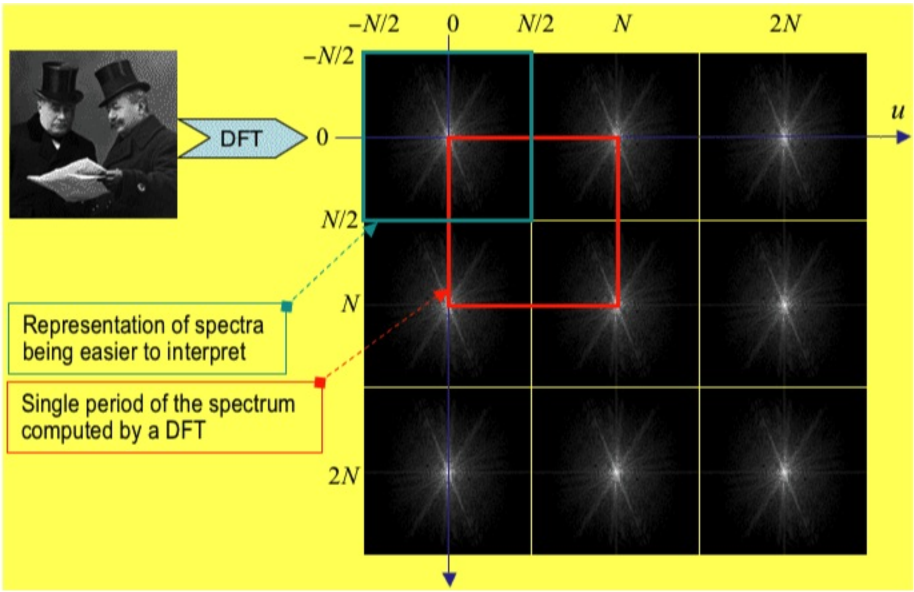
- High-frequency components are located at the four corners of $\tilde{F}(u,v)$.
- Low-frequency components are located at the middle part of $\tilde{F}(u,v)$.

Consider the discrete Fourier transform of $(-1)^{x+y}F(x,y)$:

$$\begin{aligned}
&DFT(F(x,y)(-1)^{x+y})(u,v)\\
=&\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{j\pi(x+y)}exp\left(-j2\pi\left(\frac{ux}{N}+\frac{vy}{N}\right)\right)\\
=&\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)exp\left(-j2\pi\left(\frac{(u-N/2)x}{M}+\frac{(v-N/2)y}{N}\right)\right)\\
=&\hat{F}\left(u-\frac{N}{2},v-\frac{N}{2}\right)
\end{aligned}$$

Therefore, to compute $\tilde{F}(u,v)$, we can compute DFT of $(-1)^{x+y}f(x,y)$.

The idea of centralization is illustrated below:



After centralization, the blue window is considered.

Definition 1.9. A **low-pass filter (LPF)** leaves low frequencies unchanged, while attenuating the high frequencies.

A **high-pass filter (HPF)** leaves high frequencies unchanged, while attenuating the low frequencies.

Basic steps of filtering in the frequency domain

1. Multiply $f(x,y)$ by $(-1)^{x+y}$, i.e. $\tilde{f}(x,y)=(-1)^{x+y}f(x,y)$.
2. Compute $\tilde{F}(u,v)=DFT(\tilde{f})(u,v)$.
3. Multiply $\tilde{F}$ by a real "filter" function $\tilde{H}(u,v)$ to get

$$G(u,v)=\tilde{H}(u,v)\tilde{F}(u,v)$$

   (point-wise multiplication, but not matrix multiplication)
4. Compute inverse DFT of $G(u,v)$.

5. Take real part of the result in Step 4.
6. Multiply the result in Step 5 by $(-1)^{x+y}$.

Remark. *1. H is taken to either remove low-frequency or high-frequency components.*

2. In the spatial domain,

$$\mathcal{F}^{-1}(G) = g = N^2\mathcal{F}^{-1}(H) * \mathcal{F}^{-1}(\tilde{F}) = N^2 h * \tilde{f}$$

Hence, filtering in frequency domain $\Leftrightarrow$ Linear filtering in spatial domain.

Examples of Low-Pass filters for image denoising

Note: From now on, we will assume we work on the centered spectum. That is, we consider $\hat{F}(u,v)$ where $-\frac{N}{2} \leq u \leq \frac{N}{2} - 1$ and $-\frac{N}{2} \leq v \leq \frac{N}{2} - 1$.

1. Ideal low-pass filter (ILPF):

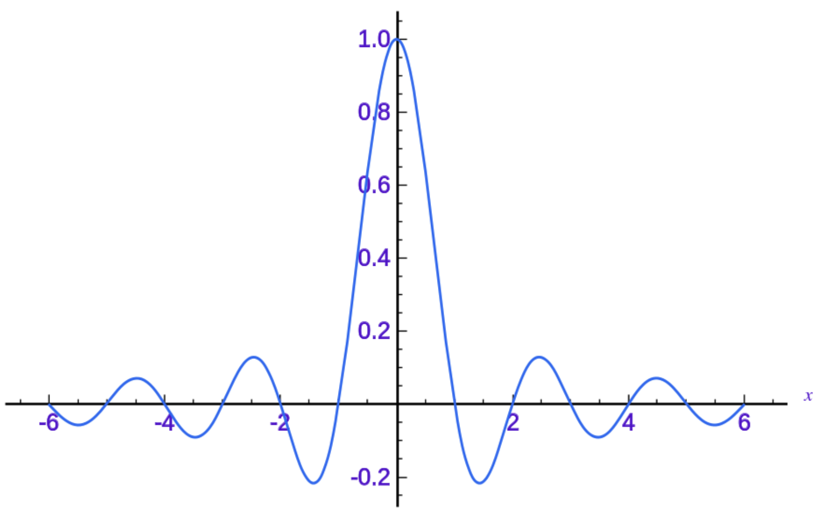
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) := dist(u, M\mathbb{Z})^2 + dist(v, N\mathbb{Z})^2 \leq D_0^2 \\ 0 & \text{if } D(u,v) > D_0^2 \end{cases},$$

where for any $x \in \mathbb{Z}$ and $y \in \mathbb{N} \setminus \{0\}$,

$$dist(x, y\mathbb{Z}) = \min\{|x - ky| : k \in \mathbb{Z}\} = \begin{cases} x - \lfloor \frac{x}{y} \rfloor y = \mathrm{mod}_y(x) & \text{if } \mathrm{mod}_y(x) \leq \frac{y}{2}, \\ \lceil \frac{x}{y} \rceil y - x = y - \mathrm{mod}_y(x) & \text{if } \mathrm{mod}_y(x) \geq \frac{y}{2}. \end{cases},$$

where the value of $\mathrm{mod}_y(x)$ is taken from $\{0, 1, \cdots, y-1\}$ (agrees with the topological definition of distance between a point and a set).

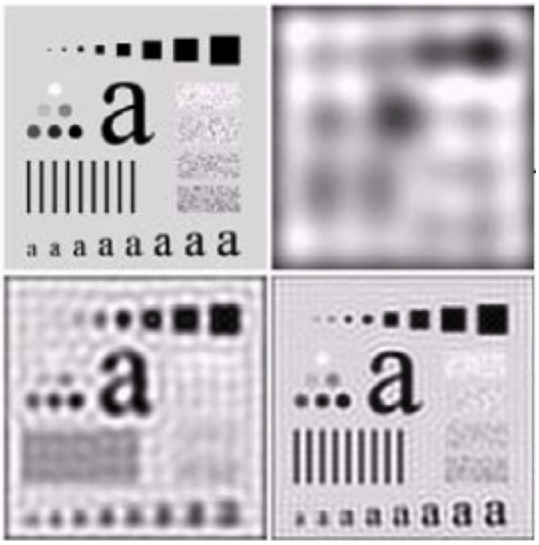
In the spatial domain, the filter looks like (in one dimensional case):



Good: Simple

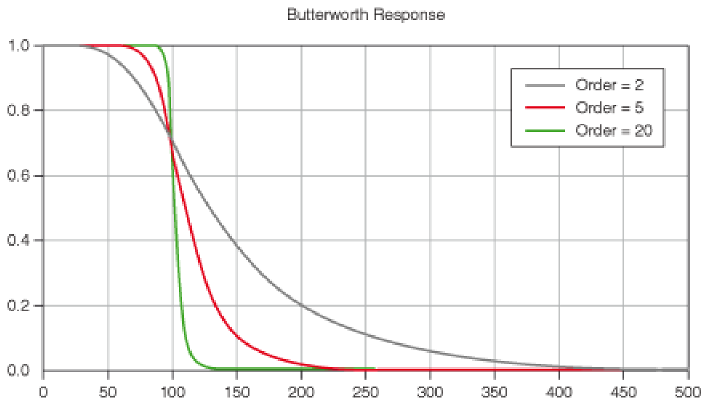
Bad: Produce the ringing effect (A pixel can be affected by pixels far away from it)

Ideal low-pass filter is applied to the following image with different D_0. Ringing effect is obviously observed.



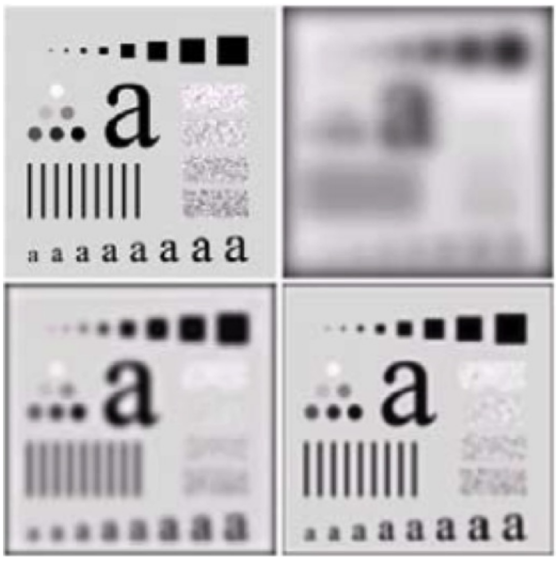
2. Butterworth low-pass filter (BLPF) of order n ($n \geq 1$ integer):

$$H(u,v) = \frac{1}{1 + (D(u,v)/D_0^2)^n}$$



<u>Good</u>: Produce less (or no visible) ringing effect if the order n is carefully chosen.

Butterworth low-pass filter is applied to the following image with different D_0 and n. No visible ringing is observed.



3. Gaussian low-pass filter:

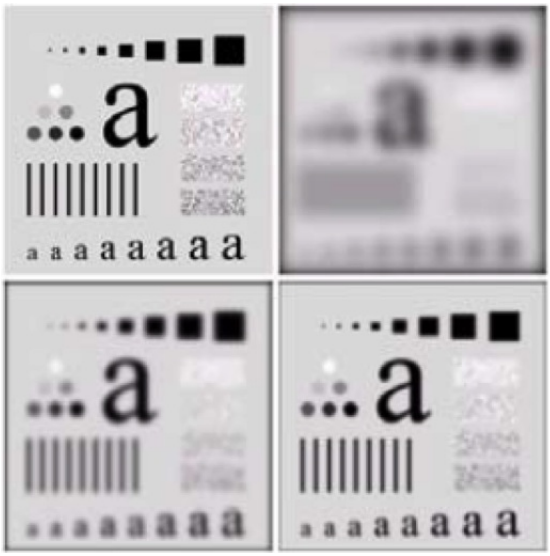
$$H(u,v) = exp\left(-\frac{D(u,v)}{2\sigma^2}\right)$$

σ is called the spread of the Gaussian function.

Good: Produce no visible ringing.

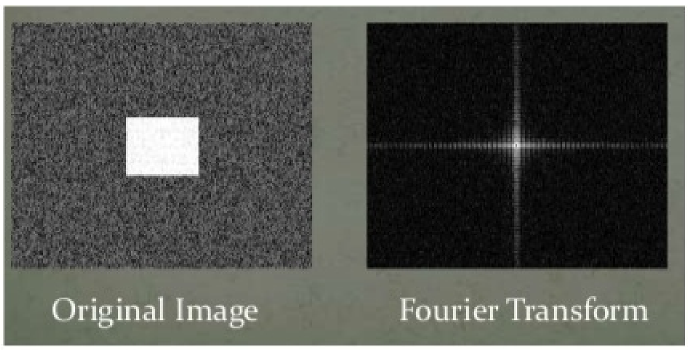
Why? Inverse DFT of a Gaussian function is also Gaussian.
Therefore, no visible ringing effect.

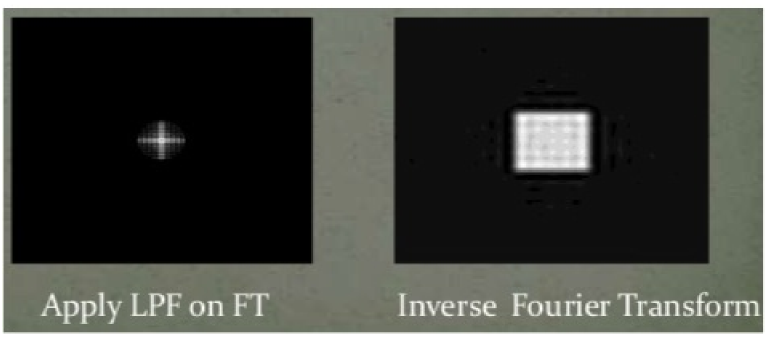
Gaussian low-pass filter is applied to the following image with different D_0 and n. No visible ringing is observed.



Experimental results of image denoising by low pass filtering in the frequency domain

Image denoising using ideal low-pass filter in the frequency domain:





Note that ringing is obviously observed.

Image denoising using butterworth low-pass filter in the frequency domain:

Image denoising using Gaussian low-pass filter in the frequency domain:

Examples of high-pass filter:

1. Ideal high-pass filter (IHPF):

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0^2 \\ 1 & \text{if } D(u,v) > D_0^2 \end{cases}$$

   Bad: Produce ringing effect

2. Butterworth high-pass filter (BHPF) of order n:

$$H(u,v) = \frac{1}{1 + (D_0^2/D(u,v))^n}$$

   Good: Less ringing.

3. Gaussian high-pass filter:

$$H(u,v) = 1 - exp\left(-\frac{D(u,v)}{2\sigma^2}\right)$$

   Good: No ringing.

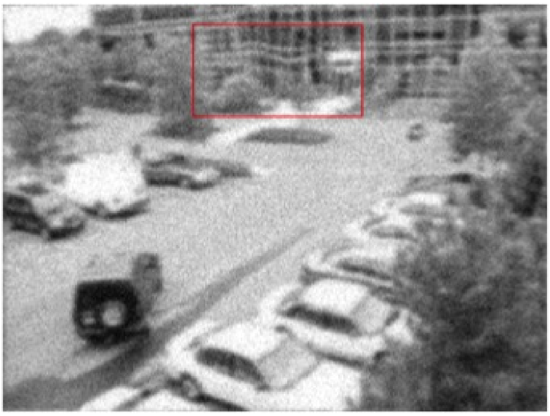
The ideal high-pass filter is applied on the following image on the left. The result after the high-pass filter is shown on the right:

2 Image deblurring in the frequency domain

2.1 Basic idea of image deblurring

Observation: Image can be degraded due to motion, turbulence, out of focus and so on. Below are some examples of blurred/degraded images.

Below shows an example of degraded image by the atmospheric turbulence:



Below shows an example of motion blur:

Goal: We would like to model image blur/degradation

In general, an observed image g can be modeled as:

$$g = H(f) + n$$

where H is the degradation function/operator and n is the additive noise.

Assumption on H:

1. H is position invariant: Let $g(x,y) = H(f)(x,y)$ and let $\tilde{f}(x,y) := f(x-\alpha, y-\beta)$. Then:
$$H(\tilde{f})(x,y) = g(x-\alpha, y-\beta)$$
2. Linear: $H(f_1+f_2) = H(f_1)+H(f_2)$ and $H(\alpha f) = \alpha H(f)$ where α is a scalar multiplication.
3. Assume the linearity can be extended to integral:
$$H\left(\iint \alpha(u,v) f(x-u, y-v)\ du\ dv\right) = \iint \alpha(u,v) H(f)(x-u, y-v)\ du\ dv$$

With the above assumptions, image degradation is in fact a convolution. Consider an impluse signal:
$$\delta(x,y) = \begin{cases} 1 & \text{if } (x,y) = (0,0) \\ 0 & \text{if } (x,y) \neq (0,0) \end{cases}$$

Then,
$$f(x,y) = f * \delta(x,y) = \sum_{\alpha=-\frac{M}{2}}^{\frac{M}{2}-1} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}-1} f(\alpha,\beta)\delta(x-\alpha, y-\beta)$$

$$\begin{aligned}
\therefore g(x,y) &= H(f)(x,y) \\
&= \sum_{\alpha}\sum_{\beta} f(\alpha,\beta) H(\delta)(x-\alpha, y-\beta) \quad \text{by linearity and position invariant} \\
&= \sum_{\alpha}\sum_{\beta} f(\alpha,\beta) h(x-\alpha, y-\beta) \quad \text{where } h(x,y) = H(\delta)(x,y) \\
&= f * h(x,y)
\end{aligned}$$

Hence, the degradation with the above assumption is actually a convolution.

Remark.

1. *Recall that h is called the point spread function.*
2. *In general, an observed image can be modelled as:*
$$g(x,y) = h * g(x,y) + n(x,y)$$
In the frequency domain, we have:
$$G(u,v) = cH(u,v)F(u,v) + N(u,v)$$
for some constant c.

2.2 Examples of degradation functions

1. Atmospheric turbulence blur:
In frequency domain, define:
$$H(u,v) = exp\left(-k(u^2+v^2)^{5/6}\right)$$
where k = degree of turbulence.

Remark. *Usually,*
$k = 0.0025$*: severe turbulence*
$k = 0.001$*: mild turbulence*
$k = 0.00025$*: low turbulence*

2. Out of focus blur:

   In the frequency domain, define $H(u,v)$ as the inverse DFT of

$$h(x,y) = \begin{cases} 1 & \text{if } x^2+y^2 \le D_0^2 \\ 0 & \text{otherwise} \end{cases}$$

   In some situations, a simple model will be used to describe out of focus blur by letting:

$$H(u,v) = \begin{cases} 1 & \text{if } u^2+v^2 \le D_0^2 \\ 0 & \text{otherwise} \end{cases}$$

   But this model is usually too simple and inaccurate.

3. Uniform Linear Motion Blur:

   Assume the image $f(x,y)$ undergoes planar motion during acquisition. Let $(x_0(t), y_0(t))$ be the motion components in the x- and y-directions. Denote the time by t and the duration of exposure by T.

   Then the observed image can be written as:

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

   We want to understand the motion blur in the frequency domain.

   Let $G(u,v) = DFT(g)(u,v)$, then:

$$\begin{aligned} G(u,v) &= \sum_x \sum_y g(x,y) e^{-j\frac{2\pi}{N}(ux+vy)} \quad \text{(assume the image is a } N \times N \text{ image)} \\ &= \sum_x \sum_y \int_0^T f(x-x_0(t), y-y_0(t))\ dt\ e^{-j\frac{2\pi}{N}(ux+vy)} \\ &= \int_0^T \left[\sum_x \sum_y f(x-x_0(t), y-y_0(t)) e^{-j\frac{2\pi}{N}(ux+vy)} \right] dt \end{aligned}$$

   Now, using the property that for

$$\tilde{f}(x,y) = f(x-x_0, y-y_0),$$
$$DFT(\tilde{f})(u,v) = DFT(f)(u,v) e^{-j\frac{2\pi}{N}(ux_0+vy_0)},$$

   we have

$$\begin{aligned} G(u,v) &= \int_0^T \left[DFT(f)(u,v) e^{-j\frac{2\pi}{N}(ux_0(t)+vy_0(t))} \right] dt \\ &= DFT(f)(u,v) \underbrace{\int_0^T e^{-j\frac{2\pi}{N}(ux_0(t)+vy_0(t))}\ dt}_{H(u,v)}, \end{aligned}$$

   Therefore, degradation function in the frequency domain is given by:

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

2.3 Image deblurring algorithms in the frequency domain

A blurred image g can be modelled as: $g = h * f(x, y) + n(x, y)$
where h = blur function in the spatial domain;
$f(x, y)$ is the original image (clean) and
$n(x, y)$ is the noise.

In the frequency domain:

$$\begin{aligned} G(u,v) = \mathcal{F}(g)(u,v) &= c\mathcal{F}(h)(u,v)\mathcal{F}(f)(u,v) + \mathcal{F}(n)(u,v) \\ &= cH(u,v)F(u,v) + N(u,v) \end{aligned}$$

for some constant $c > 0$. By replacing H by $cH(u, v)$, we can ignore the constant c.

Deblurring methods (Suppose H is known)

Method 1: Direct inverse filtering

Let $T(u,v) = \dfrac{1}{H(u,v) + \varepsilon sgn(H(u,v))}$. Compute $\hat{F}(u,v) = G(u,v)T(u,v)$. Find inverse DFT of $\hat{F}(u,v)$ to get an image $\hat{f}(x,y)$.

(Here, $sgn(z) = 1$ if $Re(z) \geq 0$ and $sgn(z) = -1$ otherwise.)

Good: Simple.

Bad: $\hat{F}(u,v) = G(u,v)T(u,v) = F(u,v) + \dfrac{N(u,v)}{H(u,v) + \varepsilon sgn(H(u,v))}$

$H(u,v)$ is usually big for (u,v) close to $(0,0)$ (associated to low frequency components) while small for (u,v) away from $(0,0)$. Therefore, $\dfrac{N(u,v)}{H(u,v) + \varepsilon sgn(H(u,v))}$ is big (large gain in high frequency) and noise dominants.

Below is an illustration of how direct inverse filtering can boast up noises.

Original Image   Blurred Image   Restored with $H^{-1}(u,v)$

A small amount of noise saturates the inverse filter.

Method 2: Modified inverse filtering

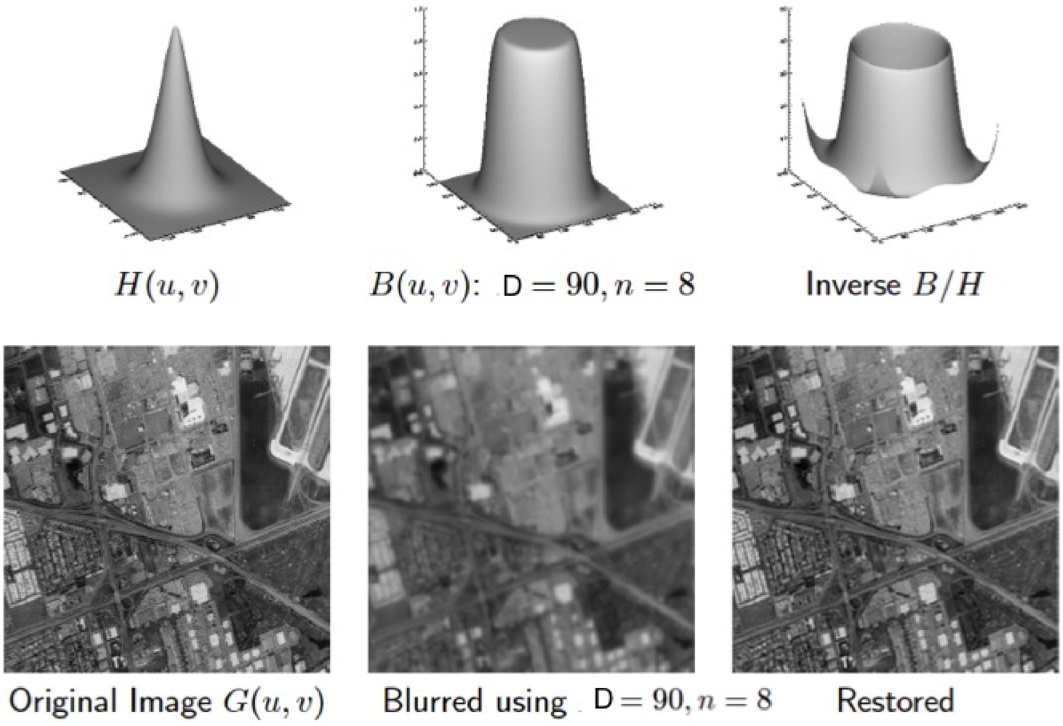
Let $B(u,v) = \dfrac{1}{1 + (\frac{u^2+v^2}{D^2})^n}$, and $T(u,v) = \dfrac{B(u,v)}{H(u,v) + \varepsilon sgn(H(u,v))}$, then

$$\hat{F}(u,v) = T(u,v)G(u,v) \approx F(u,v)B(u,v) + \frac{N(u,v)B(u,v)}{H(u,v) + \varepsilon sgn(H(u,v))}$$

$\dfrac{B(u,v)}{H(u,v) + \varepsilon sgn(H(u,v))}$ suppresses the high-frequency gain.

Bad: Has to choose D very carefully!!

Below is an illustration of how modified inverse filtering performs.



Method 3: Wiener Filter

The Wiener Filter is defined (in the frequency domain) as:

$$T(u,v) = \frac{\overline{H(u,v)}}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

where $S_n(u,v) = |N(u,v)|^2, S_f(u,v) = |F(u,v)|^2$ (Add parameters to avoid singularities)

If $S_n(u,v)$ and $S_f(u,v)$ are not known, then we let $K = S_n(u,v)/S_f(u,v)$ to get

$$T(u,v) = \frac{\overline{H(u,v)}}{|H(u,v)|^2 + K}$$

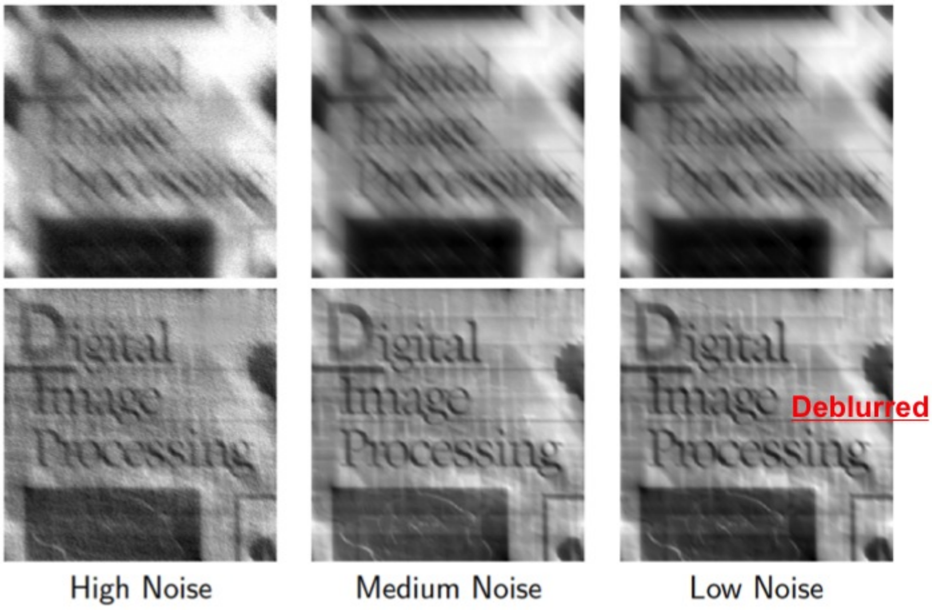
Hence, Wiener Filter can be described as the inverse filtering as follows:

$$\hat{F}(u,v) = \left[\underbrace{\left(\frac{1}{H(u,v)}\right)}_{\text{direct inverse filter}} \underbrace{\left(\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right)}_{\text{modifier}} \right] G(u,v)$$

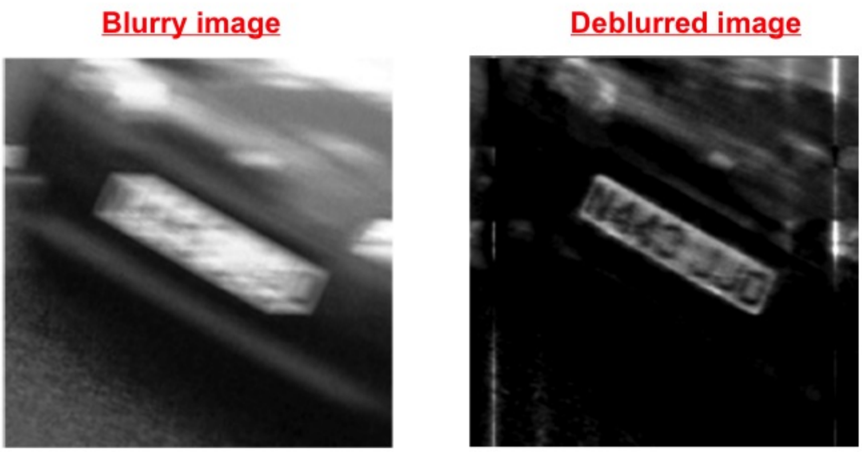
Below is an illustration of how Wiener filtering performs.

$K = 5.0$ e -4

And below is an illustration of how Wiener filtering performs on noisy and blurry images.



Finally, we show how Wiener filtering performs for deblurring the car license plate.



What does Wiener filter do mathematically?

We can show that (under certain condition), the Wiener filter minimizes the mean-square error (MSE).

(Sketch of proof)

We consider the continuous case to avoid the complicated indices. Let $g = h * f + n$, where h is the degradation function, n is the noise, f is the original clean image and g is our observed image.

Assume that f and n are spatially uncorrelated:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)n(x+r,y+s)\ dx\ dy = 0$$

for all r, s.

Then, we will show that the Wiener's filter minimizes the mean square error:

$$\mathcal{E}^2(\tilde{f}) = E((f-\tilde{f})^2) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [f(x,y)-\tilde{f}(x,y)]^2\ dx\ dy$$

where: $f(x,y)$ is the original image and $\tilde{f}(x,y) = \omega(x,y) * g(x,y)$ (where $g(x,y)$ is the observed image) is the Wiener filtered image.

Note that by Parserval's theorem:

$$\begin{aligned}\mathcal{E}^2 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (f(x,y)-\tilde{f}(x,y))^2\ dx\ dy \\ &= C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |F(u,v)-\tilde{F}(u,v)|^2\ du\ dv \quad \text{(Parseval's theorem)}\end{aligned}$$

for some constant C, where $F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j(xu+yv)}dxdy$ and $\tilde{F}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{f}(x,y)e^{-j(xu+yv)}dxdy$.

Let $G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j(xu+yv)}dxdy$ and $N(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} n(x,y)e^{-j(xu+yv)}dxdy$.

Then, we know: $\tilde{F} = WG = W(HF+N)$. In other words, $F - \tilde{F} = (1-WH)F - WN$ and

$$\mathcal{E}^2 = C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |(1-WH)F - WN|^2\ du\ dv$$

Since f and n are spatially uncorrelated, we can show that:

$$\mathcal{E}^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |(1-WH)F|^2 + |WN|^2\ du\ dv.$$

We can regard $\mathcal{E}^2$ to be depending on W.

To minimize $\mathcal{E}^2(W)$, we consider:

$$\frac{d}{dt}|_{t=0}\mathcal{E}^2(W+tV) = 0 \text{ for all } V.$$

Hence, $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} -(1-\bar{W}\bar{H})H|F|^2V - (1-WH)\bar{H}|F|^2\bar{V} + \bar{W}|N|^2V + W|N|^2\bar{V} = 0$ for all V.

Put $V = -(1-WH)\bar{H}|F|^2 + W|N|^2$, we get:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left|-(1-WH)\bar{H}|F|^2\bar{V} + W|N|^2\right|^2 = 0.$$

Thus,

$$2(-(1-\bar{W}\bar{H})H|F|^2 + \bar{W}|N|^2) = 0$$

$$\Leftrightarrow W = \frac{\bar{H}}{|H|^2 + |N|^2/|F|^2}.$$

Method 4: Constrained least square filtering

Drawback of Wiener filter:

1. $|N(u,v)|^2$ and $|F(u,v)|^2$ must be known.
2. Constant estimation of the ratio is not always suitable.

We consider a least square minimization model. The degradation process $g = h * f + n$ can be written in matrix form:

$$\vec{g} = D\vec{f} + \vec{n}$$

where $\vec{g} = \mathcal{S}(g), \vec{f} = \mathcal{S}(f), \vec{n} = \mathcal{S}(n)$, where $\mathcal{S}$ is the stacking operator.
Therefore, $\vec{g}, \vec{f}, \vec{n} \in \mathbb{R}^{MN}, D \in M_{MN\times MN}$.

Given $\vec{g}$, we want to find an estimate of f (or $\vec{f}$) such that it minimizes:

$$E(f) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |\nabla^2 f(x,y)|^2$$

subject to the constraint that: $||\vec{g} - D\vec{f}||^2 = \epsilon$

In the discrete case, we can estimate $\nabla^2 f(x,y)$ by

$$\nabla^2 f(x,y) \approx f(x+1,y) + f(x,y+1) + f(x-1,y) + f(x,y-1) - 4f(x,y).$$

More generally, in the discrete case,

$$\nabla^2 f = p * f,$$

where $p = \begin{pmatrix} 0 & & \cdots & & 0 \\ & & 1 & & \\ \vdots & 1 & -4 & 1 & \vdots \\ & & 1 & & \\ 0 & & \cdots & & 0 \end{pmatrix}$.

Remark.

1. *The constraint means we want $||\vec{n}||^2 = ||\vec{g} - D\vec{f}||^2$ has a fixed level of noise. (Control the noise level + allow noise)*
2. *The energy $E(f)$ enhances the smoothness of f.*

Suppose $E(f)$ can be written as: $(L\vec{f})^T(L\vec{f})$ (L is the transformation matrix representing the convolution with p). Then, the constrained least square problem has the optimal solution in the spatial domain satisfies:

$$[D^TD + \gamma L^TL]\vec{f} = D^T\vec{g}$$

for some suitable parameter γ (which is related to the Lagrange multiplier).

In the frequency domain,

$$\hat{F}(u,v) := DFT(f)(u,v) = \frac{1}{N^2}\frac{\overline{H(u,v)}}{|H(u,v)|^2 + \gamma|P(u,v)|^2}G(u,v)$$

where $H(u,v) = DFT(h)(u,v)$, $G(u,v) = DFT(f)(u,v)$; $P(u,v) = DFT(p)(u,v)$ with

$$p(x,y) = \begin{pmatrix} 0 & & & \cdots & & & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & \cdots & & 1 & & \cdots & 0 \\ 0 & \cdots & 1 & -4 & 1 & \cdots & 0 \\ 0 & \cdots & & 1 & & \cdots & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & & & \cdots & & & 0 \end{pmatrix}$$

Sketch of proof:

From calculus, we know the minimizer must satisfy:

$$\mathcal{D} = \frac{\partial}{\partial \vec{f}}[\vec{f}^T L^T L \vec{f} + \lambda(\vec{g} - D\vec{f})^T(\vec{g} - D\vec{f})] = 0$$

where λ is the Lagrange multiplier. Here,

$$\frac{\partial K}{\partial \vec{f}} = \begin{pmatrix} \frac{\partial K}{\partial f_1} & \frac{\partial K}{\partial f_2} & \cdots & \frac{\partial K}{\partial f_{N^2}} \end{pmatrix}^T$$

Easy to check:

$$\frac{\partial \vec{f}^T \vec{a}}{\partial \vec{f}} = \vec{a}; \ \frac{\partial \vec{b}^T \vec{f}}{\partial \vec{f}} = \vec{b}$$

Also, if A is an $N^2 \times N^2$ square matrix, then:

$$\frac{\partial \vec{f}^T A \vec{f}}{\partial \vec{f}} = (A + A^T)\vec{f}$$

$$\begin{aligned} \therefore \mathcal{D} = 0 &\Rightarrow (2L^T L)\vec{f} + \lambda(-D^T\vec{g} - D^T\vec{g} + 2D^T D\vec{f}) = 0 \\ &\Rightarrow (D^T D + \gamma L^T L)\vec{f} = D^T \vec{g} \end{aligned}$$

where $\gamma = \dfrac{1}{\lambda}$ and λ is the Lagrange multiplier.

Parameter γ can be determined by direct substitution into the equation:

$$[\vec{g} - D\vec{f}]^T[\vec{g} - D\vec{f}] = \epsilon$$

In the frequency domain, note that both D and L are block-circulant. Recall that a matrix A is **block-circulant** if:

$$A = \begin{pmatrix} A_0 & A_{N-1} & A_{N-2} & \cdots & A_1 \\ A_1 & A_0 & A_{N-1} & \cdots & A_2 \\ A_2 & A_1 & A_0 & \cdots & A_3 \\ \vdots & \vdots & \vdots & & \vdots \\ A_{N-1} & A_{N-2} & A_{N-3} & \cdots & A_0 \end{pmatrix}$$

where $A_0, A_1, \cdots, A_{N-1}$ are submatrices and they are themselves circulant matrices.

A matrix B is **circulant** if:

$$B = \begin{pmatrix} b_0 & b_{M-1} & b_{M-2} & \cdots & b_1 \\ b_1 & b_0 & b_{M-1} & \cdots & b_2 \\ b_2 & b_1 & b_0 & \cdots & b_3 \\ \vdots & \vdots & \vdots & & \vdots \\ b_{M-1} & b_{M-2} & b_{M-3} & \cdots & b_0 \end{pmatrix}$$

Remark. *Fact about circulant matrix:*

Let $B = \begin{pmatrix} b(0) & b(M-1) & \cdots & b(1) \\ b(1) & b(0) & \cdots & b(2) \\ \vdots & \vdots & \cdots & \vdots \\ b(M-1) & b(M-2) & \cdots & b(0) \end{pmatrix}$ *be a circulant matrix. Then the eigenvalues of* B *is given by:*

$$\lambda(k) = b(0) + b(1)e^{\frac{2\pi j}{M}(M-1)k} + b(2)e^{\frac{2\pi j}{M}(M-2)k} + \cdots + b(M-1)e^{\frac{2\pi j}{M}k}$$

where $k = 0, 1, 2, \cdots M-1$.

Its associated eigenvector is given by:

$$\vec{w}(k) = \begin{pmatrix} 1 \\ e^{\frac{2\pi j}{M}k} \\ e^{\frac{2\pi j}{M}2k} \\ \vdots \\ e^{\frac{2\pi j}{M}(M-1)k} \end{pmatrix}.$$

Recall that both D and L are block-circulant. In fact, D, L, D^T, L^T can be written as:

$$D = W\Lambda_D W^{-1}, D^T = W\Lambda_D^* W^{-1}, L = W\Lambda_L W^{-1}, L^T = W\Lambda_L^* W^{-1}$$

where W is invertible and Λ_D, Λ_L are diagonal matrices. In fact,

$$\Lambda_D(k,i) = \begin{cases} N^2\hat{D}\left(\text{mod}_N(k), \left\lfloor \frac{k}{N} \right\rfloor\right) & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

where $H = DFT(h)$. Similarly,

$$\Lambda_L(k,i) = \begin{cases} N^2\hat{L}\left(\text{mod}_N(k), \left\lfloor \frac{k}{N} \right\rfloor\right) & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

where $\hat{L} = DFT(p)$.

By substitution, $(D^T D + \gamma L^T L)\vec{f} = H^T\vec{g}$
$\Rightarrow \cancel{W}(\Lambda_D^*\Lambda_D + \gamma\Lambda_L^*\Lambda_L)W^{-1}\vec{f} = \cancel{W}\Lambda_D^* W^{-1}\vec{g}$
We can show that

$$\Lambda_D^*\Lambda_D = \begin{pmatrix} N^4|H(0,0)|^2 & & & & & \\ & N^4|H(1,0)|^2 & & & & \\ & & \ddots & & & \\ & & & N^4|H(N-1,0)|^2 & & \\ & & & & \ddots & \\ & & & & & N^4|H(N-1,N-1)|^2 \end{pmatrix}$$

and

$$\Lambda_L^*\Lambda_L = \begin{pmatrix} N^4|P(0,0)|^2 & & & & & \\ & N^4|P(1,0)|^2 & & & & \\ & & \ddots & & & \\ & & & N^4|P(N-1,0)|^2 & & \\ & & & & \ddots & \\ & & & & & N^4|P(N-1,N-1)|^2 \end{pmatrix}$$

Also, $W^{-1}\vec{f} = N\mathcal{S}(F), W^{-1}\vec{g} = N\mathcal{S}(G)$ where $F = DFT(f), G = DFT(g)$.

Combining all, we see that for all (u, v):

$$N^4[|\hat{H}(u,v)|^2 + \gamma|\hat{L}(u,v)|^2]NF(u,v) = N^2\overline{\hat{H}(u,v)}NG(u,v)$$

$$\Rightarrow N^2\frac{|\hat{H}(u,v)|^2 + \gamma|\hat{L}(u,v)|^2}{\overline{\hat{H}(u,v)}}F(u,v) = G(u,v)$$

Example 2.1. Consider a 3×3 image f. Let $\vec{f} = \mathcal{S}(f)$. Recall that the Laplacian of f can be calculated by:

$$\Delta = p * f \quad \text{where} \quad p = \begin{pmatrix} 0 & & \cdots & & 0 \\ & & 1 & & \\ \vdots & 1 & -4 & 1 & \vdots \\ & & 1 & & \\ 0 & & \cdots & & 0 \end{pmatrix}$$

Suppose $\mathcal{S}(\Delta f) = L\vec{f}$ for some matrix $L \in M_{9\times 9}$. Find L and show that L is block-circulant.

Solution. We can extend the image periodically as:

$$\begin{array}{c|ccc|c} f_{33} & f_{31} & f_{32} & f_{33} & f_{31} \\ \hline f_{13} & f_{11} & f_{12} & f_{13} & f_{11} \\ f_{23} & f_{21} & f_{22} & f_{23} & f_{21} \\ f_{33} & f_{31} & f_{32} & f_{33} & f_{31} \\ \hline f_{13} & f_{11} & f_{12} & f_{13} & f_{11} \end{array}$$

Then, L can be written as:

$$L = \underbrace{\left(\begin{array}{ccc|ccc|ccc} -4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & -4 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -4 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & -4 \end{array}\right)}_{\text{block-circulant}} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}$$

Example 2.2. Consider a 2×2 image f. Let $g = h * f$ where $h \in M_{2\times 2}$. Let $H \in M_{4\times 4}$ such that $H\vec{f} = \vec{g}$ where $\vec{f} = \mathcal{S}(f), \vec{g} = \mathcal{S}(g)$. What is H?

Solution. $g_{mn} = \sum_{i=0}^{1} \sum_{j=0}^{1} h(m-i, n-j) f_{ij}$

$$\therefore \begin{pmatrix} g_{00} \\ g_{10} \\ g_{01} \\ g_{11} \end{pmatrix} = \underbrace{\left(\begin{array}{c|c|c|c} h_{0,0} & h_{-1,0} & h_{0,-1} & h_{-1,-1} \\ h_{1,0} & h_{0,0} & h_{1,-1} & h_{0,-1} \\ h_{0,1} & h_{-1,1} & h_{0,0} & h_{-1,0} \\ h_{1,1} & h_{0,1} & h_{1,0} & h_{0,0} \end{array}\right)}_{H} \begin{pmatrix} f_{00} \\ f_{10} \\ f_{01} \\ f_{11} \end{pmatrix}$$

Let

$$H_u = \begin{pmatrix} h(0,u) & h(-1,u) \\ h(1,u) & h(0,u) \end{pmatrix} = \begin{pmatrix} h(0,u) & h(1,u) \\ h(1,u) & h(0,u) \end{pmatrix} \quad \text{by periodic condition}$$

then, it is easy to check

$$H = \begin{pmatrix} H_0 & H_{-1} \\ H_1 & H_0 \end{pmatrix} = \begin{pmatrix} H_0 & H_1 \\ H_1 & H_0 \end{pmatrix} \quad \text{by periodic condition}$$

Remark. *In general, suppose f is an $N \times N$ image, $g = h * f$ and $H\vec{f} = \vec{g}$. Then,*

$$H = \begin{pmatrix} H_0 & H_{-1} & H_{-2} & \cdots & H_{-N+1} \\ H_1 & H_0 & H_{-1} & \cdots & H_{-N+2} \\ H_2 & H_1 & H_0 & \cdots & H_{-N+3} \\ \vdots & \vdots & \vdots & & \vdots \\ H_{N-1} & H_{N-2} & H_{N-3} & \cdots & H_0 \end{pmatrix}$$

where

$$H_u = \begin{pmatrix} h(0,u) & h(N-1,u) & h(N-2,u) & \cdots & h(1,u) \\ h(1,u) & h(0,u) & h(N-1,u) & \cdots & h(2,u) \\ h(2,u) & h(1,u) & h(0,u) & \cdots & h(3,u) \\ \vdots & \vdots & \vdots & & \vdots \\ h(N-1,u) & h(N-2,u) & h(N-3,u) & \cdots & h(0,u) \end{pmatrix}$$

Easy to see that H is block circulant and H_u is circulant.

Diagonalization of H

Let H be the block-circulant matrix as defined above. Define a matrix with elements:

$$W_N(k,n) := \frac{1}{\sqrt{N}} exp\left(\frac{2\pi j}{N}kn\right) \qquad 0 \le n \le N-1$$

Consider the **Kronecker product** $\otimes$ of W_N with itself:

$$W := W_N \otimes W_N$$

The **Kronecker product** of two matrices are given by:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1N}B \\ a_{21}B & a_{22}B & \cdots & a_{2N}B \\ \vdots & \vdots & & \vdots \\ a_{N1}B & a_{N2}B & \cdots & a_{NN}B \end{pmatrix}$$

Easy to check: $W^{-1} = W_N^{-1} \otimes W_N^{-1}$ where:

$$W_N^{-1}(k,n) := \frac{1}{\sqrt{N}} exp\left(-\frac{2\pi j}{N}kn\right) \qquad 0 \le n \le N-1$$

Let

$$\Lambda(k,i) = \begin{cases} N^2 \hat{H}\left(\text{mod}_N(k), \left\lfloor \frac{k}{N} \right\rfloor\right) & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

where $\hat{H}$ = DFT of the point spread function h, $\left\lfloor \frac{k}{N} \right\rfloor$ = largest integer smaller than or equal to $\frac{k}{N}$ and $\text{mod}_N(k) = k(\text{mod } N)$ (e.g. $10(\text{mod } 3) = 1$)

Then, we can show that $H = W\Lambda W^{-1}$ and $H^{-1} = W\Lambda^{-1}W^{-1}$.

Also, $H^T = W\Lambda^* W^{-1}$. (Λ^* is the complex conjugate of Λ)

By direct calculation, it is easy to check that $W^{-1}\vec{g} = N\mathcal{S}(G)$ where $G = DFT(g)$.

Example 2.3. Assume that:

$$G = \begin{pmatrix} g_{00} & g_{01} & g_{02} \\ g_{10} & g_{11} & g_{12} \\ g_{20} & g_{21} & g_{22} \end{pmatrix} \quad \text{and} \quad W_3^{-1} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & exp\left(-\frac{2\pi j}{3}\right) & exp\left(-\frac{2\pi j}{3}2\right) \\ 1 & exp\left(-\frac{2\pi j}{3}2\right) & exp\left(-\frac{2\pi j}{3}\right) \end{pmatrix}$$

Then:

$$W^{-1} = W_3^{-1} \otimes W_3^{-1}$$

$$
= \frac{1}{3}\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} \\
1 & 1 & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}4} \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}3} \\
1 & 1 & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2}
\end{pmatrix}
$$

Note that $e^{-\frac{2\pi j}{3}3} = e^{-2\pi j} = 1$, and $e^{-\frac{2\pi j}{3}4} = e^{-\frac{2\pi j}{3}3}e^{-\frac{2\pi j}{3}} = e^{-\frac{2\pi j}{3}}$, then

$$
W^{-1}g = \frac{1}{3}\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} \\
1 & 1 & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 \\
1 & 1 & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\
1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 \\
1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2}
\end{pmatrix}
\begin{pmatrix} g_{00} \\ g_{10} \\ g_{20} \\ g_{01} \\ g_{11} \\ g_{21} \\ g_{02} \\ g_{12} \\ g_{22} \end{pmatrix}
$$

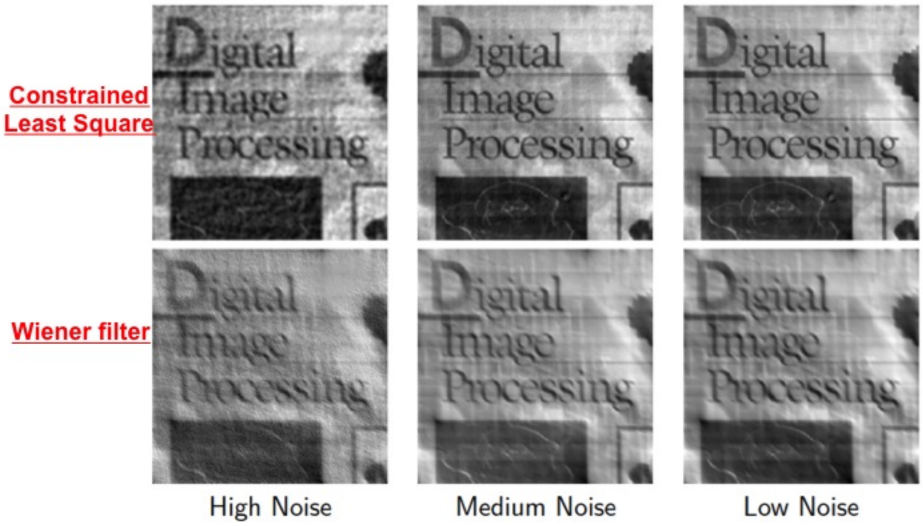
$$
= \frac{1}{3}\begin{pmatrix}
g_{00} + g_{10} + g_{20} + g_{01} + g_{11} + g_{21} + g_{02} + g_{12} + g_{22} \\
g_{00} + g_{10}e^{-\frac{2\pi j}{3}} + g_{20}e^{-\frac{2\pi j}{3}2} + g_{01} + g_{11}e^{-\frac{2\pi j}{3}} + g_{21}e^{-\frac{2\pi j}{3}2} + g_{02} + g_{12}e^{-\frac{2\pi j}{3}} + g_{22}e^{-\frac{2\pi j}{3}2} \\
\vdots \\
\vdots \\
\vdots
\end{pmatrix}
$$

Careful examination of the elements of this vector shows that they are the Fourier components of G, multiplied with 3, compared at various combinations of frequencies (u, v), for $u, v = 0, 1, 2$, and arranged as follows:

$$
3 \times \begin{pmatrix} \hat{G}(0,0) \\ \hat{G}(1,0) \\ \hat{G}(2,0) \\ \hat{G}(0,1) \\ \hat{G}(1,1) \\ \hat{G}(2,1) \\ \hat{G}(0,2) \\ \hat{G}(1,2) \\ \hat{G}(2,2) \end{pmatrix}
$$

Note that $\hat{G}(m,n) = \frac{1}{9}\sum g_{kl}e^{-\frac{2\pi j}{3}(mk+nl)} \Rightarrow \hat{G}(1,0) = \frac{1}{9}\sum g_{kl}e^{-\frac{2\pi j}{3}(k)}$. This shows that $W^{-1}g$ yields N times the Fourier transform of G, as a column vector.

Below is an illustration of how Constrained least square filtering performs on noisy and blurry images (compared with the Wiener's filtering).



3 Image sharpening in the frequency domain

The goal of image sharpening is to enhance an image so that it shows more obvious edges.

Method 1: Laplacian mask

Recall: $\Delta f(x,y) = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$.

By Taylor's expansion:

$$\begin{aligned}
\frac{\partial^2 f}{\partial x^2} &\approx \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2} \\
\frac{\partial^2 f}{\partial y^2} &\approx \frac{f(x,y+h) - 2f(x,y) + f(x,y-h)}{h^2} \\
\therefore \Delta f(x,y) &\approx \frac{f(x,y+h) + f(x,y-h) + f(x+h,y) + f(x-h,y) - 4f(x,y)}{h^2}
\end{aligned}$$

In the case of images, we let $h = 1$. Hence, $\Delta f(x,y) = l * f(x,y)$ for some matrix l.

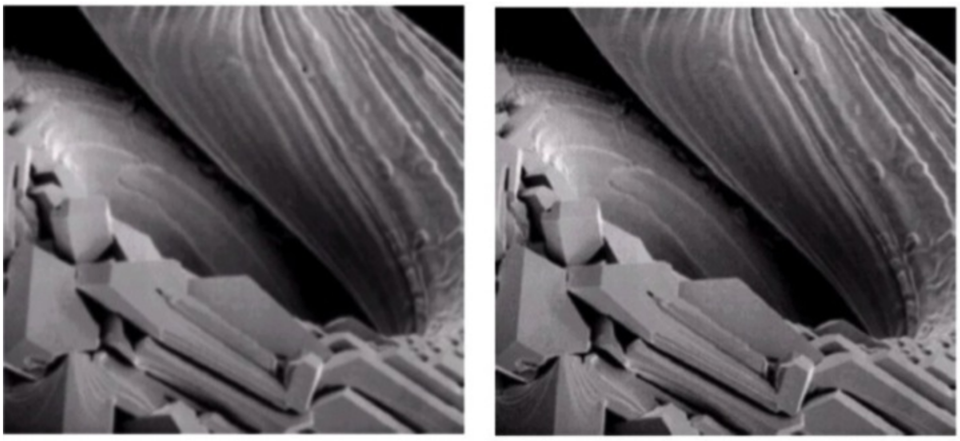
In the frequency domain,

$$\begin{aligned}
DFT(g) &= DFT(f) - DFT(\Delta f) \\
&= DFT(F)(u,v) + DFT(l)DFT(F)(u,v) \\
&= (1 - H_{laplace}(u,v))F(u,v)
\end{aligned}$$

where $H_{laplace}(u,v) = DFT(l)(u,v)$

Below is an illustration of how Laplacian masking performs on two different images:

Original image　　Laplacian mask



Original image　　Laplacian mask

<u>Method 2:</u> Unsharp masking

<u>Idea:</u> Image sharpening = Add back high frequency component.

Definition 3.1. Let f be an input image (may be blurry). Compute smoother image (by Gaussian filter or mean filter) f_{smooth}. Define the sharper image g as:

$$g(x, y) = f(x, y) + k(f(x, y) - f_{smooth}(x, y))$$

When $k = 1$, the method is called <u>unsharp masking</u>.
When $k > 1$, the method is called <u>highboost filtering</u>.

In the frequency domain, let

$$DFT(f_{smooth})(u, v) = H_{LP}(u, v)DFT(f)(u, v),$$

where H_{LP} is the low pass filter.

Then: $DFT(g)(u, v) = [1 + k(1 - H_{LP}(u, v))]DFT(f)(u, v)$.

Below is an illustration of how unsharp masking performs:

(a) Original (b) Global, Gaussian 121×121

Exercises

The entries of all matrices below are indexed by $\{0, 1, \cdots, M-1\} \times \{0, 1, \cdots, N-1\}$ unless otherwise specified.

1. Apply

   i. the ideal low-pass filter of radius 2;
   ii. the ideal high-pass filter of radius 2;
   iii. the Butterworth low-pass filter of radius 2 and order 2;
   iv. the Butterworth high-pass filter of radius 2 and order 2;
   v. the Gaussian low-pass filter of spread 2;
   vi. the Gaussian high-pass filter of spread 2;

   on the following 4×4 images:

   (a) $f_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$;

   (b) $f_2 = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$;

   (c) $f_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.

2. (a) Let H_1 be a Butterworth low-pass filter applicable to $M \times N$ images with radius $D_0 > 0$ and order $n > 0$, with $M \geq 6$ and $N \geq 8$.
   Suppose $3H_1(1, 2) = 4H_2(3, 1) = 6H_3(2, 4)$. Find D_0 and n.

   (b) Let (a, b, c) be a Pythagorean triple, i.e. $a, b, c \in \mathbb{N} \setminus \{0\}$ satisfying $a^2 + b^2 = c^2$.
   Let H_2 be a Gaussian low-pass filter applicable to $M \times N$ images with spread $\sigma > 0$, with $M \geq 2c$ and $N \geq 4c$.
   Suppose $H_2(a, b) = 8H_2(0, 2c)$. Find σ in terms of c.

   (c) Let H_3 be a Butterworth high-pass filter applicable to 10×10 images with radius $D_0 > 0$ and order $n > 0$.
   Suppose $H_3(2, 7) = \frac{1}{3}$ and $H_3(5, 1) = \frac{4}{5}$. Find D_0 and n.

3. Compute the degradation functions in the frequency domain that correspond to the following $M \times N$ convolution kernels h, i.e. find $H \in M_{M\times N}(\mathbb{C})$ such that

$$DFT(h * f)(u,v) = H(u,v)DFT(f)(u,v)$$

for any periodically extended $f \in M_{M\times N}(\mathbb{R})$:

(a) Assuming integer k satisfies $k \le \min\{\frac{M}{2}, \frac{N}{2}\}$,

$$h_1(x,y) = \begin{cases} \frac{1}{(2k+1)^2} & \text{if } dist(x, M\mathbb{Z}) \le k \text{ and } dist(y, N\mathbb{Z}) \le k, \\ 0 & \text{otherwise;} \end{cases}$$

(b) Letting $r > 1$,

$$h_2(x,y) = \begin{cases} \frac{r}{r+4} & \text{if } D(x,y) = 0, \\ \frac{1}{r+4} & \text{if } D(x,y) = 1, \\ 0 & \text{otherwise;} \end{cases}$$

(c)

$$h_3(x,y) = \begin{cases} \frac{1}{4} & \text{if } D(x,y) = 0, \\ \frac{1}{8} & \text{if } D(x,y) = 1, \\ \frac{1}{16} & \text{if } D(x,y) = 2, \\ 0 & \text{otherwise;} \end{cases}$$

(d)

$$h_4(x,y) = \begin{cases} -4 & \text{if } D(x,y) = 0, \\ 1 & \text{if } D(x,y) = 1, \\ 0 & \text{otherwise;} \end{cases}$$

(e) Letting $a, b \in \mathbb{Z}$ and $T \in \mathbb{N} \setminus \{0\}$ such that $|a|(T-1) < M$ and $|b|(T-1) < N$,

$$h_5(x,y) = \begin{cases} \frac{1}{T} & \text{if } (x,y) \in \{(at, bt) : t = 0, 1, \cdots, T-1\} \\ 0 & \text{otherwise.} \end{cases}$$

4. Deconvolve the following $M \times N$ images g in the frequency domain with respect to the convolution kernel h, via

i. direct inverse filtering;

ii. modified inverse filtering with $D_0 = \frac{M}{2}$ and $n = 2$;

iii. Wiener filter with $K \equiv \frac{1}{4}$.

The reader is reminded of the constant c involved.

(a) $g_1 = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$, $h_1 = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$;

(b) $g_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, $h_2 = \begin{pmatrix} 2 & -1 \\ 0 & 0 \end{pmatrix}$;

(c) $g_3 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$, $h_3 = \frac{1}{4}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$;

(d) $g_4 = \begin{pmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$, $h_4 = \begin{pmatrix} -2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$;

(e) $g_5 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}$, $h_5 = \frac{1}{8}\begin{pmatrix} 4 & 2 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$.

5. Prove that for any $f \in M_{M\times N}(\mathbb{C})$,

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |DFT(f)(m,n)|^2 = \frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} |f(k,l)|^2 .$$

6. Given $g \in M_{M\times N}(\mathbb{R})$, block circulant $D, L \in M_{MN\times MN}(\mathbb{R})$ and $\varepsilon > 0$, the constrained least square filtering scheme aims to minimize $\|L\vec{f}\|_2^2$ subject to $\|\vec{g}-D\vec{f}\|_2^2 = \varepsilon$ over $f \in M_{M\times N}(\mathbb{R})$, where $\vec{f} = \mathcal{S}(f)$ and $\vec{g} = \mathcal{S}(g)$ vectorized by the stack operator $\mathcal{S}$.

   (a) Given Lagrange multiplier λ for the equality constraint, show that if f is a minimizer of the above constrained minimization problem, then

   $$(\lambda D^T D + L^T L)\vec{f} = \lambda D^T \vec{g}.$$

   (b) i. For every $k \in \mathbb{N}\setminus\{0\}$, denote the $k\times k$ DFT matrix by U_k. Show that $\sqrt{M}U_M$ is unitary.

      ii. Let $V \in M_{M\times M}(\mathbb{C})$ and $U \in M_{N\times N}(\mathbb{C})$ both be unitary. Show that $U\otimes V$ is unitary.

      iii. Let C be an $M\times M$ circulant matrix in the form of:

      $$C = \begin{pmatrix} c_0 & c_{M-1} & c_{M-2} & \cdots & c_1 \\ c_1 & c_0 & c_{M-1} & \cdots & c_2 \\ c_2 & c_1 & c_0 & \cdots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{M-1} & c_{M-2} & c_{M-3} & \cdots & c_0 \end{pmatrix}.$$

      Show that $U_M C \overline{U_M}$ is diagonal. Hence find the eigenvalues of C.

      iv. Show that $(U_N \otimes U_M)D(\overline{U_N}\otimes\overline{U_M})$ is diagonal. Hence find the eigenvalues of D.

   (c) To simplify scaling and indexing, for every $k \in \mathbb{N}\setminus\{0\}$, consider $W_k \in M_{k\times k}(\mathbb{C})$ in replacement of U_k, where W_k is defined by

   $$W_k(x,y) = \frac{1}{\sqrt{k}} e^{2\pi j \frac{xy}{k}}$$

   for any $x, y \in \mathbb{Z}\cap[0, k-1]$.

      i. Prove that for any $x, y \in \mathbb{Z}\cap[0, M-1]$,

      $$W_M D_{k-l}\overline{W_M}(x,y) = \begin{cases} \sum\limits_{m=0}^{M-1} D_{k-l,m} e^{2\pi j \frac{mx}{M}} & \text{if } x = y, \\ 0 & \text{otherwise,} \end{cases}$$

      where $D_{k-l,m}$ is the value of the entries on the diagonal of D_{k-l} with indices $\{(x,y) : x - y \in m + M\mathbb{Z}\}$.

      ii. Prove that for any $x, y \in \mathbb{Z}\cap[0, MN-1]$,

      $$(W_N\otimes W_M)D(\overline{W_N}\otimes\overline{W_M})(x,y) = \begin{cases} MN\overline{DFT(h)(\mathrm{mod}_M(x), \lfloor \frac{x}{M}\rfloor)} & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

      where $h \in M_{M\times N}(\mathbb{R})$ satisfies $D\mathcal{S}(f) = \mathcal{S}(h * f)$ for every $f \in M_{M\times N}(\mathbb{R})$.

   (d) Prove that

   $$(W_N \otimes W_M)\mathcal{S}(f) = \sqrt{MN}\mathcal{S}(\overline{DFT(f)})$$

   for any $f \in M_{M\times N}(\mathbb{R})$.

   (e) Hence, establish the constrained least square filtering scheme for $M\times N$ images:

   $$DFT(f)(u,v) = \frac{\lambda\overline{DFT(h)(u,v)}}{MN[\lambda|DFT(h)(u,v)|^2 + |DFT(p)(u,v)|^2]} DFT(g)(u,v),$$

   where $p \in M_{M\times N}(\mathbb{R})$ satisfies $D\mathcal{S}(f) = \mathcal{S}(p * f)$ for every $f \in M_{M\times N}(\mathbb{R})$.

7. Deconvolve the following images with respect to the given convolution kernels via constrained linear filtering:

   (a) $g_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, $h_1 = \frac{1}{4}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$, $p_1 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$, $\varepsilon_1 = \frac{1}{4}$,
   $$f_1 = \operatorname*{argmin}_{\substack{f \in M_{2\times 2}(\mathbb{R}) \\ \|g_1 - h_1 * f\|_F^2 = \varepsilon_1}} \|p_1 * f\|_F^2;$$

   (b) $g_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $h_2 = \begin{pmatrix} -2 & 1 \\ 1 & 0 \end{pmatrix}$, $p_2 = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}$ $\varepsilon_2 = 1$,
   $$f_2 = \operatorname*{argmin}_{\substack{f \in M_{2\times 2}(\mathbb{R}) \\ \|h_2 * f\|_F^2 = \varepsilon_@}} \|p_2 * f\|_F^2.$$

8. Using the following low-pass filters:

   i. ideal low-pass filter of radius 2,
   ii. Butterworth low-pass filter of radius 2 and order 2,
   iii. Gaussian low-pass filter of spread 2,

   perform unsharp masking on the following 4×4 images:

   (a) $f_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$;

   (b) $f_2 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$;

   (c) $f_3 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.