3 Filtering in the Frequency Domain

3.1 Background

The Fourier transform is a fundamental mathematical tool in image processing, especially in image filtering. The name of 'Fourier transform' or 'Fourier Series' can be traced back to the date of year 1822 when a French mathematician published his famous work – the Analytic Theory of Heat. In this work, he had shown that any periodic function can be expressed as the sum of sines and cosines of different frequencies weighted by different values. This 'sum' interpretation is usually termed as 'Fourier series' (in modern mathematical theory this is in fact a special case of orthogonal decomposition of a function within one of the functional space e.g. Hilbert space, where each axis is represented by a function, e.g., the sine or cosines functions.). For non-periodic function, this type of operations is called 'Fourier Transform'. For simplicity, in the following sections we will use the general term 'Fourier transform' to represent such a transform in both of the periodic and non-periodic functions.

The merit of the Fourier transform is that a band limited function or signal could be reconstructed without loss of any information. As we will see, this in fact allows us to work completely in the frequency domain and then return to the original domain. Working in the frequency domain is more intuitive and can enable the design of image filters easier.

Although the initial idea of the Fourier transform was applied to the heat diffusion, this idea has been quickly spread into other industrial fields and academic disciplines. With the advent of computer and discovery of the fast Fourier transform, the real-time practical processing of the Fourier transform becomes possible.

In the following sections, we will show step by step the basic concept of Fourier transform from 1-D continuous function to the 2-D case.

3.2 Preliminaries

3.2.1 Impulse function and its properties

Impulse function is one of the key concepts of sampling, either in spatial domain or in the frequency domain. A typical impulse function is defined as follows:

$$\delta(x) = \begin{cases} \infty & if\ x = 0 \\ 0 & otherwise \end{cases} \tag{3.2.1}$$

And its integral is subject to the following constrains:

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \quad (3.2.2)$$

A very important feature of the impulse function is that for any continuous function $f(x)$, the integral of the product of the function $f(x)$ with the impulse function is simply the value of the $f(x)$ at the location where the impulse happens. For example, if the location of the impulse is at T, the following mathematical expression can help us to get a sample from the function $f(x)$ at T.

$$\int_{-\infty}^{\infty} f(x)\delta(x-T)dx = f(T) \quad (3.2.3)$$

The above equation implies that the operation of getting a value from the function $f(x)$ at T can be performed by the integral of the product of this function with the impulse function at location T. This is the fundamental of the sampling theory.

3.3 Sampling and the Fourier Transform of Sampled Functions

3.3.1 Sampling

In the real world, a scene or an object could be mathematically represented by a continuous function. However, in today's digital world, to show or store an image of the scene in a computer the scene has to be converted into a discrete function. This digitalization process in fact can be called 'sampling'. In fact, sampling is the process/operation of converting a continuous function into a discrete function. One role of this operation is that it makes the image representation tractable in a computer memory or storage and displayable on screen as well as visible to humans. In this process, a key question is what kind of sampling frequencies should be adopted in the sampling process (or which point will be selected to describe the discrete version of the function)? A *sample* refers to a value or set of values at a point in time and/or space. Mathematically, the sampling principle can be described as follows (beginning from simple periodic functions):

Consider a continuous 1-D periodic function, $y(t) = y(t + nT)$ with $n = 0,1,2,3, \ldots$, where t represents the time in seconds. Now we are going to sample this function by measuring the value of continuous function every T seconds. T here is called sampling interval. Thus the sampled function (discrete version) can be represented by

$$y[n] = y(nT), n = 0,1,2,3, \ldots. \quad (3.3.1)$$

The sampling interval is an indicator of the frequency that we use to sample the function. Here we introduce a sampling frequency or rate that measures the number of samples in unit (usually in one second), $f = 1/T$. The sampling rate is conventionally measured in hertz or in samples per second. Intuitively, an ideal sampling rate allows us to reconstruct the original function successfully.

The output of sampling is a numerical sequence of the continuous function, usually represented by a matrix or vector of numbers. Image can be viewed as a 2D sampling version of the real world. A conventional representation of the image is a matrix of M number of rows and N number of columns. The spatial coordinates denotes the relative location of the pixels within the matrix (element index) while the entry of the matrix represents the value of the grey level intensity or color intensity at that pixel, usually represented by $I(i, j)$. The smallest coordinates are conventionally set as (1,1) in the popular MATLAB program. Note the notation (i, j) in a matrix does not correspond to the actual physical coordinates. For example, a function $f(x)$; $x = 0,0.5,1$ is represented by a matrix $[f(0), f(0.5), f(1)]$. The entry value at (1,2) is $f(0.5)$ with actual physical coordinates x at 0.5.

3.3.2 The Fourier Transform of Sampled Functions

The convolution theory tells us that the Fourier transform of the convolution of two functions is the product of the transforms of the two functions. Thus by applying the Fourier transform on the both side of Equation 3.2-3, we can get:

$$\begin{aligned}\tilde{F}\big(f(nT)\big) &= \tilde{F}\big(f(t)\otimes\delta(t)\big)\\ &= \frac{1}{\Delta \mathrm{T}}\sum_{n=-\propto}^{+\propto} F(u-\frac{n}{\Delta T})\end{aligned} \tag{3.3.2}$$

This shows that Fourier transform of the sample function is a sampled version of the Fourier transform $F(u)$ of continuous function $f(t)$. *Note that the sample pace in the frequency domain is the inverse of the sample pace in the spatial domain.*

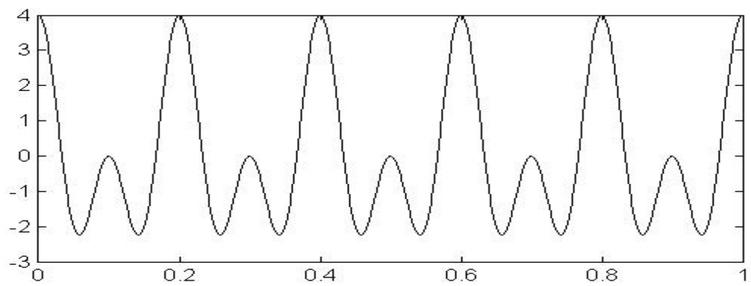
3.3.3 The Sampling Theorem

As stated before, the reason why we discuss the sampling and Fourier transforms is to find a suitable sampling rate. Insufficient sampling may lose partial information of a continuous function. So, what is the relationship between the proper sampling interval and the function frequency? In other words, under what circumstances is it possible to reconstruct the original signal successfully (perfect reconstruction)?

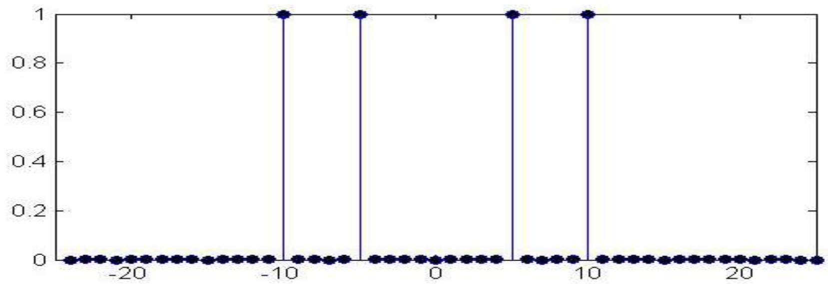
A partial answer is provided by the Nyquist–Shannon sampling theorem [7], which provides a sufficient (but not always necessary) condition under which perfect reconstruction is possible. The sampling theorem guarantees that band limited signals (i.e., signals which have a maximum frequency) can be reconstructed perfectly from their sampled version, if the sampling rate is more than twice the maximum frequency. Reconstruction in this case can be achieved using the Whittaker-Shannon interpolation formula [8].

The frequency equivalent to one-half of the sampling rate is therefore a bound on the highest frequency that can be unambiguously represented by the sampled signal. This frequency (half the sampling rate) is called the Nyquist frequency of the sampling system. Frequencies above the Nyquist frequency can be observed in the sampled signal, but their frequency is ambiguous. This ambiguity is called *aliasing later* shown in **Figure 27**. To handle this problem effectively, most analog signals are filtered with an anti-aliasing filter (usually a low-pass filter with cutoff near the Nyquist frequency) before their conversion to the sampled discrete representation. We will show this principle using the following examples:

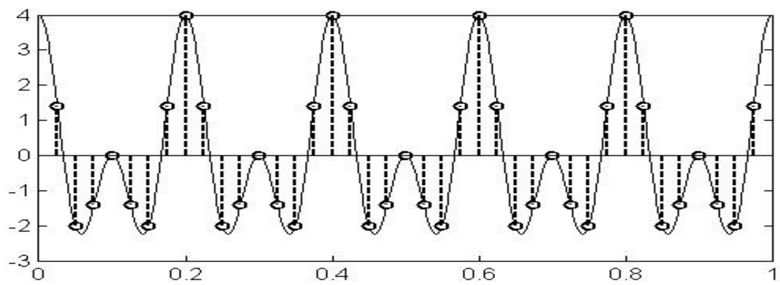
Let us consider a simple function $f(x) = \cos(2\pi f_1 x) + \cos(2\pi f_2 x)$ with f_1 set as 5 and f_2 set as 10. We expect that the power spectrum of the FFT of the signal will have peaks at 5 and 10. **Figure 27** (a) shows the original signal and (b) show the power spectrum. As expected, we can observe two peaks at u = 5 and 10 respectively in (b). According the Nyquist–Shannon sampling theorem, the minimum sampling rate is at least equal to the two times of the highest frequency of the signal. In our example, the trade-off rate is 20 (2×10). We now sample this signal with a rate larger than 20, say 40 to see what happens. **Figure 27** (c) shows the sampled signal using this rate, and (d) shows the corresponding power spectrum. We can see that the signal looks quite similar to the original one with the peaks at frequency 5 and 10. This is sometimes called *oversampling* and leads to a perfect reconstruction of the signal. Another case is to sample the signal at a lower rate than 20, for instance 10. **Figure 27** (d) and (f) show the sampled signal using rate 10 and its power spectrum. From this figure, we can see that trend of the signal is not preserved and the power spectrum looks different from the original one in (b). This is called *under-sampling or frequency aliasing*. In practice, most signals are band infinite. That means they usually have a very infinite frequency. The effect of aliasing almost exists in every case. However, in the case of high quality imaging system such an aliasing is not visually observable by human eyes.



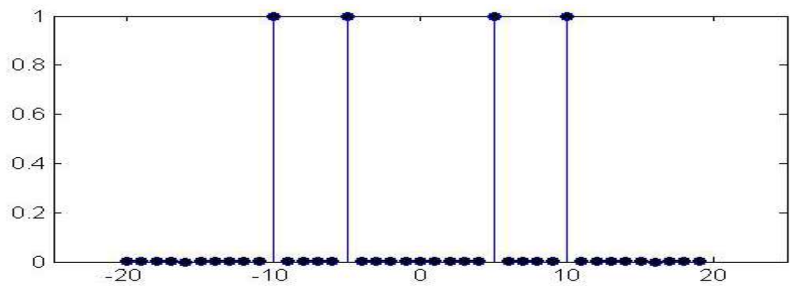
(a)



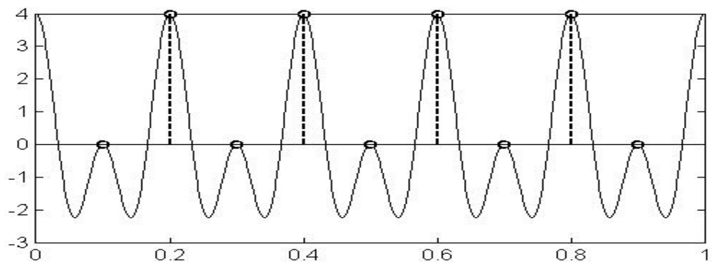
(b)



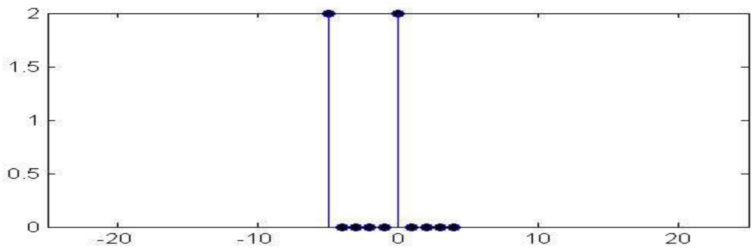
(c)



(d)



(e)



(f)

Figure 27 Illustration of the sampling theorem: (a) original; (b) the power spectrum; (c) the sampled signal at sampling rate 40 (oversampling); (d) the power spectrum of (c); (e) the sampled signal at sampling rate 10 (under sampling); (f) the power spectrum of (e).

3.4 Discrete Fourier Transform

3.4.1 One Dimensional DFT

Let us start with the 1-D discrete Fourier transform (DFT) for a discrete function $f(x)$ for $x = 0,1,2 \ldots M-1$. Its 1D Fourier transform can be calculated as follows:

$$\boldsymbol{F(u) = \sum_{x=0}^{M-1} f(x)\, e^{-j\omega}}, \tag{3.4.1}$$

$$\omega = \boldsymbol{2\pi\left(\frac{ux}{M}\right)} \tag{3.4.2}$$

where $u = 0,1,2, \ldots M-1$. is usually called frequency variables, similar to the coordinates x in the spatial domain. The exponential on the right side of the equation can be expanded into sinuses and cosines with variables determining their frequencies as follows:

$$e^{-j\omega} = \cos(\omega) + jsin(\omega), \tag{3.4.3}$$

The inverse discrete Fourier transform is:

$$f(x) = \frac{1}{M}\sum_{u=0}^{M-1} F(u)e^{j\omega}, \tag{3.4.4}$$

3.4.2 Relationship between the Sampling and Frequency Intervals

Consider a discrete function with the fixed sample interval ΔT (i.e., the time duration between samples) and it contains N samples. Thus the total duration of the function is $D = N * \Delta T$. Using Δu to represent the spacing in the Fourier frequency domain, the maximum frequency (i.e., the entire frequency range, also the maximum samples that will get during a time unit, usually per second) is just the inverse of the sampling interval.

$$\mathrm{U} = 1/\Delta T \tag{3.4.5}$$

Note that we have data samples and the frequency domain also has the same number of samples. The spacing per sample can be calculated as $\Delta u = \frac{U}{N} = \frac{1}{\Delta T * N}$. Thus, the relationship between ΔT and Δu can be stated using the following equation:

$$\Delta T = N * \Delta u \tag{3.4.6}$$

3.5 Extension to Functions of Two Variables

Similar to the 1D Fourier transform, the 2-D discrete Fourier transform can be expressed as follows:

$$F(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} I(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}, \tag{3.5.1}$$

Where $I(x, y)$ is the original image of size $M \times N$. u and v are in the same range.

The inverse Fourier Transform is defined as follows:

$$I(x,y) = \frac{1}{MN}\sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{3.5.2}$$

where $F(u, v)$ is the Fourier transform of the original image. Usually the Fourier transform is complex in general. It can be written in the polar form:

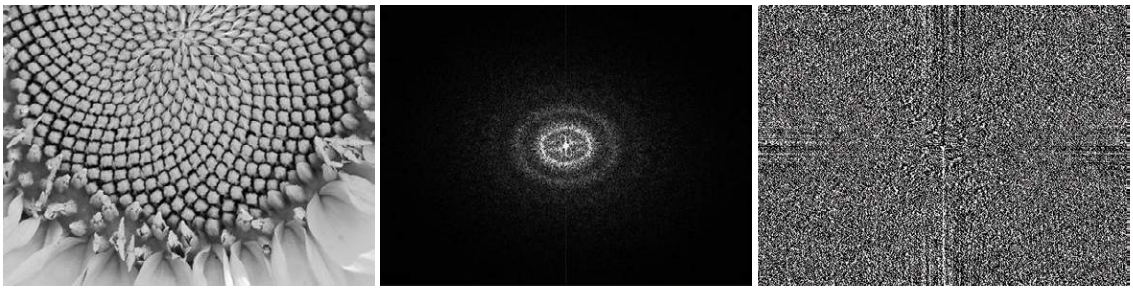
$$F(u,v) = a + bj = |F(u,v)|e^{j\phi(u,v)} \tag{3.5.3}$$

And we have the magnitude and the phase angle as follows:

$$|F(u,v)| = (a^2 + b^2)^{0.5} \tag{3.5.4}$$

$$\phi(u,v) = \tan^{-1}(\tfrac{a}{b}), \tag{3.5.5}$$

The power spectrum is the square of the magnitude $|F(u, v)|$. **Figure 28** shows an image, its power spectrum and phase angle respectively.



(a) (b) (c)

Figure 28 An image of sunflower (a), its power spectrum (b), and its phase component (c).

3.6 Some Properties of the 2-D Discrete Fourier Transform

The 2D Fourier transform has the following properties that are often used.

1) *Translation invariant (magnitude)*. If a function $I(x, y)$ is translated by an offset $(\Delta x, \Delta y)$, its Fourier transform becomes $F(u,v)e^{-j2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)}$. We can see that the translation in the spatial domain only affects the phase component by $e^{-j2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)}$. The magnitude $|F(u,v)|$ is not affected.
2) *Periodicity*. The 2D Fourier transform and its inverse are infinitely periodic u in the and v directions.
3) *symmetry*:
a) The Fourier transform of a real function is *conjugate symmetric*: $F^*(u,v) = F(-u,-v)$, and the magnetite is always symmetric, i.e, $|F(u,v)| = |F(-u,-v)|$.
b) The Fourier transform of a real and even function is symmetric: $F(u,v) = \ F(-u,-v)$.
4) *Linearity*: The Fourier transform of the sum of two signals equals to the sum of the Fourier transform of those two signals indepedently. $g(x,y) + f(x,y) \ <=> F(u,v) + G(u,v)$
5) *Scaling*: If the signal is spatially scaled wider or smaller, the corresponding Fourier transform will be scaled smaller or wider. The magnitude is also smaller or wider. $f(\alpha x, \beta y) <=> \frac{1}{\alpha\beta}F(\frac{u}{\alpha},\frac{v}{\beta})$.

The Fourier transform also has other properties. For a complete list of properties, interested readers may refer to some handbooks of signal/image processing and sites like Wikipedia and Mathworld.

3.7 The Basics of Filtering in the Frequency Domain

As we have stated in Chapter 2, in the spatial domain the filtering operation is sometimes performed by the convolution operation with a filtering mask (Note this is the case for linear filter. For nonlinear filter, such as the median filter, the operation can NOT be performed in terms of convolution due the linearity of convolution). Fortunately, there exist some relationship between the *convolution* and *Fourier* transform as follows:

$$\Im(g(x,y)\otimes f(x,y)) = G(u,v)F(u,v), \tag{3.7.1}$$

where $\otimes$ denotes the convolution operator, $\Im$ represents the Fourier transform operation. Here the Fourier transform of the result of convolution between two functions is the product of the Fourier transform of those two functions respectively.

The Fourier transform provides a direct theoretical explanation of the filtering technique used in the spatial domain. Using this technique, we have an intuitive perception of which frequency will be used to filter the signal. A straightforward step in the filtering design is to select a filter which removes some signals at certain frequency bands that we do not want to keep. The design of the filter will be down to how to specify $F(u, v)$ – the Fourier transform of the spatial filter mask $f(x, y)$. $F(u, v)$ can be referred to as the *filter transfer function*.

A good example is the commonly used filtering mask – Gaussian filtering. **Figure 29** shows an Gaussian filter function in the space domain and its power spectrum in frequency domain. Note the difference between the variance.

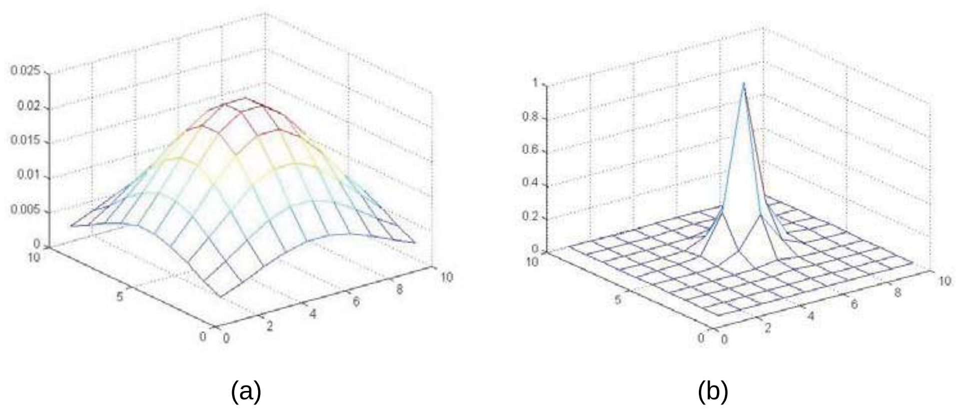


Figure 29 A Gaussian filter in the frequency domain (a) and its power spectrum in the frequency domain (b).

3.8 Image Smoothing Using Frequency Domain Filters

In this section we will introduce three main low pass filters including ideal lowpass Filters, Butterworth lowpass filters as well as Gaussian lowpass filters. In the following section, we will use the panda image to show the principles of the different filters:

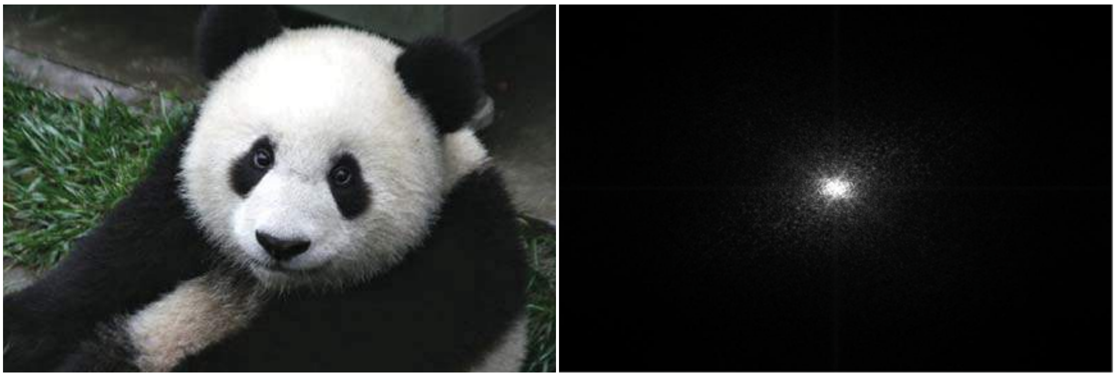


Figure 30 An image of a panda (a), and its power spectrum of the gray level image (b).

3.8.1 Ideal Lowpass Filters

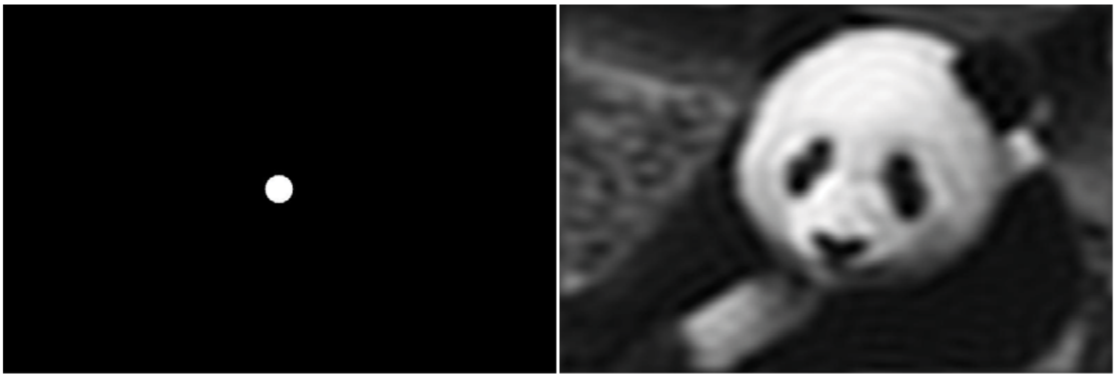
An ideal low pass filter is to switch off some high frequency signals and to allow the low frequency signal to get through based on a threshold. Mathematically, it is defined as follows:

$$M(u,v) = \begin{cases} 0, & u^2 + v^2 \geq T0 \\ 1, & u^2 + v^2 < T0 \end{cases} \tag{3.8.1}$$

Here, T0 is a threshold, usually a nonnegative number. It represents the square of the radius distance from the point (u, v) to the center of the filter, usually called *cut-off frequency.* Intuitively, the set of $u^2 + v^2 = T_0$ points at forms a circle. It blocks signals outside the circle by multiplying 0 and passes the signals outside the circle by multiplying 1. By doing this, we get the following filtered results in the frequency domain:

$$I(u,v).* M(u,v) = \begin{cases} 0, & u^2 + v^2 \geq T_0 \\ I(u,v), & u^2 + v^2 < T_0 \end{cases} \tag{3.8.2}$$

The operator represents the element wise dot product operation. **Figure 31** shows an image example and its filtered image using the ideal low pass filters.



(a) (b)

Figure 31 An ideal low pass filtering with cut-off frequency set as 10 (a), and the corresponding filtered image (b) from **Figure 30** (a).

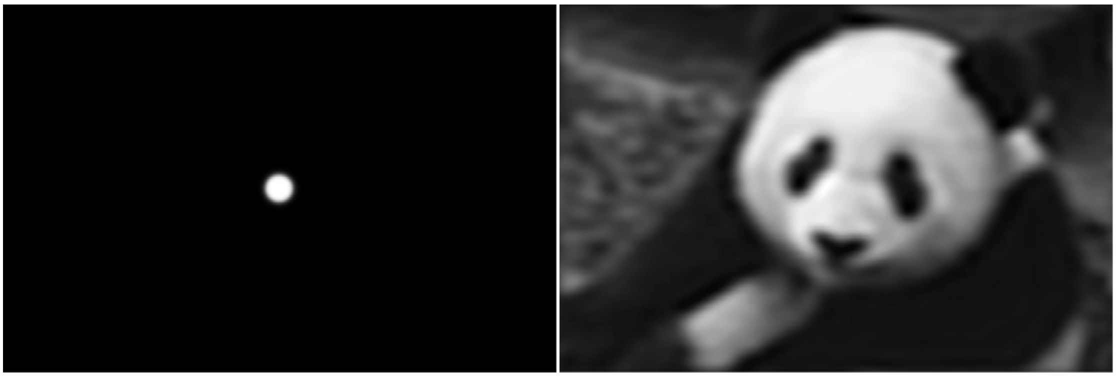
3.8.2 Butterworth Lowpass Filter

The ideal low pass filter has a sharp discontinuity as T_0. This will introduce some ring effects on the filtered image as shown in **Figure 31** (b). In contrast, the Butterworth lowpass filter is much smoother at T_0 and defined as follows:

$$M(u.v) = \frac{1}{1+\left[\frac{T(u,v)}{T_0}\right]^{2n}} \tag{3.8.3}$$

where $T(u, v)$ is the radius from the point (u, v) to the origin in the frequency domain.

Figure 32 shows an example of Butterworth low pass filter with $T_0 = 10$, and the filtered image from **Figure 29** (a). Comparing the filtered image with the one in **Figure 30**, we notice that the Butterworth low pass filter produces less ringing effects than an ideal low pass filter using the same cut-off frequency.



(a) (b)

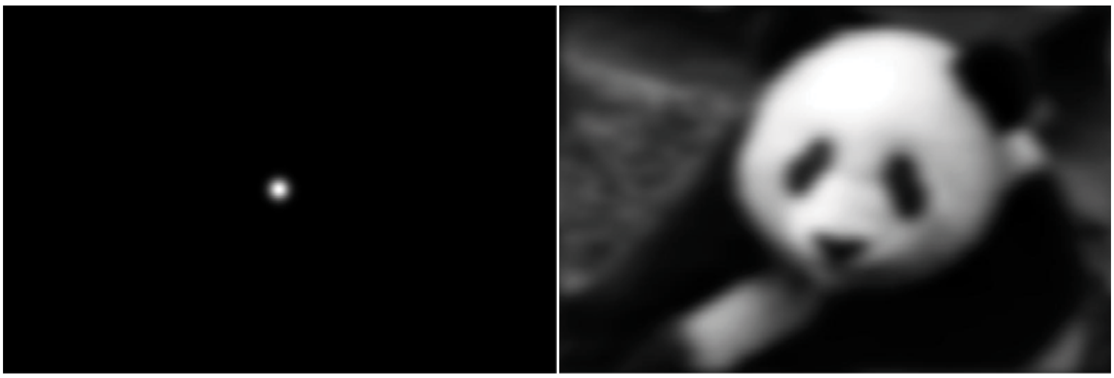
Figure 32 Butterworth low pass with cut-off frequency set as 10 (a), and the corresponding filtered image (b) from **Figure 31** (a).

3.8.3 Gaussian Lowpass Filters

In 2D image processing, an exemplar filter is Gaussian filtering. It is perhaps the most commonly used low pass filter in the literature. Mathematically in the frequency domain, it is defined as follows:

$$G(u, v) = Ae(-d(u, v)/2\sigma^2) \tag{3.8.4}$$

where d is an affine distance function from the center point to the current point (u, v), usually denoted by $d = \frac{1}{2\sigma^2}(u^2 + v^2)$ with σ denoting the standard deviation. **Figure 33** shows the plot of the 2D Gaussian low pass filter and the filtered panda image respectively.



(a) (b)

Figure 33 Gaussian low pass filter (a) and the smoothed panda image (b), with set as 10 and A is set to 1.

3.9 Image Sharpening Using Frequency Domain Filters

3.9.1 Ideal Highpass Filters

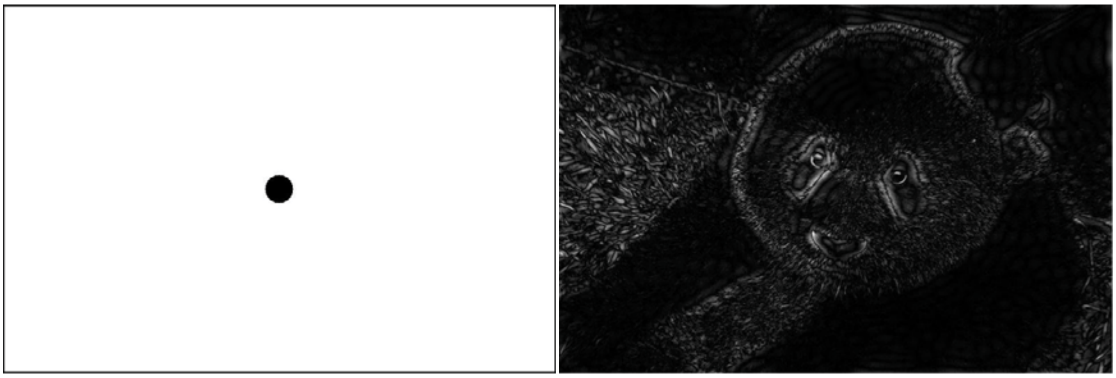
An ideal highpass filter is to switch off some low frequency signals and to allow the high frequency signal to pass based on a threshold. It implies the difference between 1 and the ideal low pass filtering. Mathematically, it is defined as follows:

$$M(u,v) = \begin{cases} 1, & u^2+v^2 \geq T0 \\ 0, & u^2+v^2 < T0 \end{cases} \tag{3.9.1}$$

In contrast to the ideal low pass filter, it blocks signals falling inside the circle by multiplying 0 and pass the signals settling outside the circle by multiplying 1. By doing this, we get the following filtered results in the frequency domain:

$$F(u,v).*M(u,v) = \begin{cases} F(u,v), & u^2+v^2 \geq T_0 \\ 0, & u^2+v^2 < T_0 \end{cases} \tag{3.9.2}$$

Figure 34 shows an example of using the ideal highpass filters applied to an image.



(a) (b)

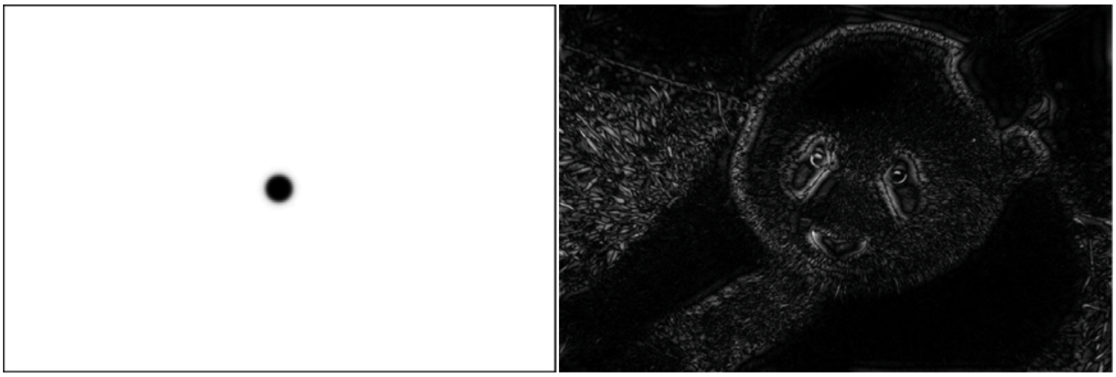
Figure 34 An ideal high pass filter (a) and corresponding sharpened image (b).

3.9.2 Butterworth Highpass Filters

The Butterworth Highpass Filter is simply the difference between 1 and the Butterworth lowpass filter T. It is defined as follows:

$$M(u.v) = 1 - \frac{1}{1+\left[\frac{T(u,v)}{T_0}\right]^{2n}} \tag{3.9.3}$$

where again is the radius from the point to the origin in the frequency domain.



(a) (b)

Figure 35 Butterworth high pass filter (a) and the results of filtering the panda image (b).

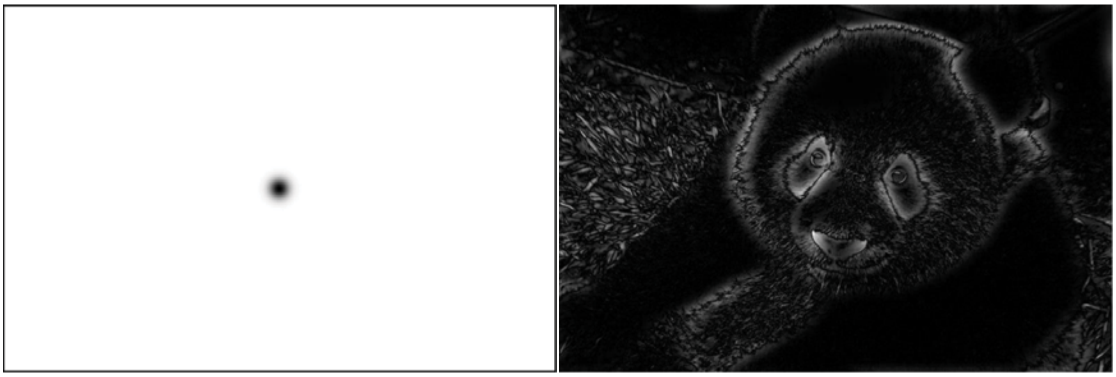
3.9.3 Gaussian High Pass Filters

A Gaussian high pass filter is defined as follows:

$$G(u,v) = 1 - Ae(-d(u,v)/2\sigma^2), \tag{3.9.4}$$

where d is the distance from the center point to the current point (u, v), usually denoted by $d = \frac{1}{2\sigma^2}(u^2 + v^2)$ with representing the standard deviation.

Figure 36 shows a Gaussian high pass filter and the results of applying this filter onto the panda image.



(a) (b)

Figure 36 Gaussian high pass filter (a) and the results of filtering the panda image (b).

3.9.4 The Fourier transform of Laplacian operator

Another common high pass filer is the well known Laplacian filter. It is a 2D isotropic measure of the second spatial derivatives of an image. It can capture the region with rapid intensity changes. Traditionally it is used for edge detection. But in recent years it has been widely used for blob detection based on its extended version the *Laplacian of Gaussian (LoG)*. It is mathematically defined as follows:

$$\nabla^2 I(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{3.9.5}$$

It is easy to prove that the Fourier transform of the above operator is:

$$\mathrm{F}(\nabla^2 I(x,y)) = -4\pi(u^2+v^2)I(u,v) \tag{3.9.6}$$

Thus the filter is as follows:

$$F = -4\pi(u^2+v^2) \tag{3.9.7}$$

The Laplacian filter is sensitive to noise. In practical, it is common to blur the image first and then apply the Laplacian operator onto the smoothed image. The transform is traditionally termed as *Laplacian of Gaussian* (LoG). The Fourier transform of the LoG transform can be approximately represented as the following forms:

$$-4(\mathrm{u}^2+\mathrm{v}^2)\exp\left(-0.5(u^2+v^2)/\sigma^2\right) \tag{3.9.8}$$

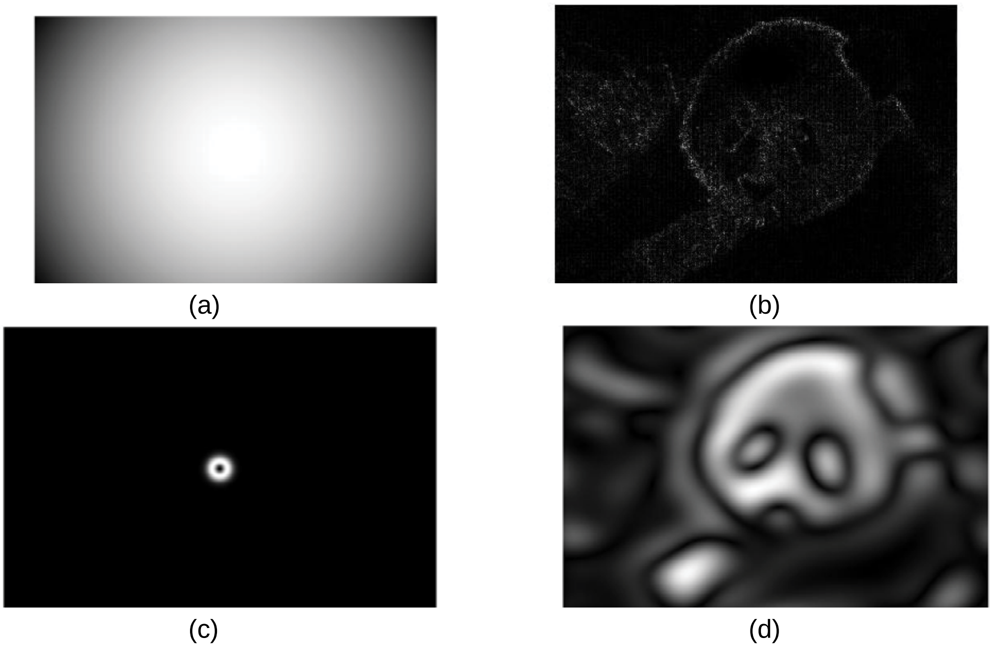


Figure 37 Laplacian fitler (up-left), the results of laplacian fitlered image (up-right). LoG filter (down-left). For better visibilty, here we show the inverse of the LoG filter, and the results of Log filter image (down-left).

3.9.5 Unsharp filter, Highboost Filtering, and High-Frequency-Emphasis Filtering

The unsharp filter is a very simple operator that subtracts a smoothed (unsharped) version of an image. This operation is called *unsharp filter* and the resulting difference is usually called *unsharp mask*. It is majorly used in the printing and publishing industry for enhancing image edges.

In principle the unsharping mask is an edge image $e(x, y)$ from the input image $I(x, y)$ by the unsharping filter:

$$e(x,y) = I(x,y) - I_s(x,y) \tag{3.9.9}$$

where $I_s(x, y)$ is a blurred version of the image $I(x, y)$. In order to enhance or boost the edge image (high frequency part), the edge image $e(x, y)$ is then added back to the original image a weighting strategy as follows:

$$g(x,y) = I(x,y) + w * e(x,y) \tag{3.9.10}$$

Generally, w is non negative and it controls the contribution of the edge component of the image. It is can be easily seen that when is w set to be 0, there is no additional contribution of the edge part. When w is larger than 1, it is sometimes called *high boost filtering in the spatial domain*.

We can easily perform the above operation in the frequency domain by applying the Fourier transform on the above equation. The final form can be as follows:

$$G(u, v) = I(u, v) + w(1 - H(u, v)I(u, v)) \tag{3.9.11}$$

where $I(u, v)$ is the Fourier transform of the original image $I(x, y)$ and $G(u, v)$ is the Fourier transform of the boosted image $g(x, y)$.

3.9.6 Frequency Domain Homomorphic Filtering

Image can also be expressed as the product of illumination and reflectance. This is known as the *illumination-reflectance* model. This model is first developed by MIT, USA. This model basically states that an image is produced by a sensor based on the amount of energy that it received from the object. The energy that an image sensor or camera received is determined by illumination radiation source and the features of the reflectance of the object itself. Their relationship is viewed as a product. It is mathematically expressed as follows:

$$f(x, y) = i(x, y) * r(x, y) \tag{3.9.12}$$

where $f(x, y)$ is the image. $i(x, y)$ is the illumination component and $r(x, y)$ the reflectance component.

The reflectance is the features of the object itself. Generally we are more interested in reflectance than illumination. A basic idea is to enhance reflectance and to reduce illumination. In order to do this, a common approach is called *Homomorphic Filtering*. To utilize the advantages of the Fourier transform, we need to convert the product into sum by applying the log operator on both sides of the equation. Thus we get:

$$\log(f) = \log(i) + \log(r) \tag{3.9.13}$$

Now the illumination-reflectance model in the log space becomes a linear model. It is easier for us to process this in the frequency domain. It has been shown that the illumination component majorly usually lies in the low frequency band and reflectance in the higher frequency band. This motivates us to borrow some high pass filter to enhance the reflectance part and degrade the other. Once doing this, we then can use the inverse Fourier transform to convert it back to the log space, and finally deploy the exponent operation to convert the log space into the original space. A common flow chart of the *Homomorphic Filtering* is shown in **Figure 38**.

Figure 38 The flowchart of the homomorphic filtering procedure.

Figure 39 shows an example of image enhancement using the homomorphic filtering.

(a) (b)

Figure 39 An example of image illumination correction using the homomorphic filtering.

3.10 Summary

The Fourier transform is a key component of image processing. In other words, without Fourier transform, modern digital image processing will lack amounts of contributions to the community. In this chapter, we have introduced the principles of the discrete Fourier transform in 1D and 2D cases. We also illustrated the sampling theorem. Keep in mind that the sampling rate of a signal should be at least two times higher than the highest frequency of the image content. We further presented several low pass and high pass filters in the frequency domain including ideal low pass filter, Butterworth low pass filter, as well as Gaussian filter. A simple rule is that the sum of the low pass filter and the corresponding filter of the same type are usually equivalent to one. Other filters are also common in the field of image processing such as the median filter. However, those filters cannot be easily interpreted in the frequency domain. We have introduced them in Chapter 2.

3.11 References and Further Reading

[7] C.E. Shannon, "Communication in the presence of noise", Proc. Institute of Radio Engineers, vol. 37, no. 1, pp. 10–21, Jan. 1949. Reprint as classic paper in: Proc. IEEE, vol. 86, no. 2, (Feb. 1998)

[8] E.T. Whittaker, "On the Functions Which are Represented by the Expansions of the Interpolation Theory", Proc. Royal Soc. Edinburgh, Sec. A, vol. 35, pp. 181–194, 1915.

[9] Wikipedia: Nyquist–Shannon sampling theorem. http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem. Accessed on 10 October, 2009.

3.12 Problems

(12) What is Nyquist-Shannon sampling theorem?

(13) W hat is the relationship of lowpass filter and high pass filter?

(14) Explain the different steps and motivation of the homomorphic filtering.