Understanding the Fourier Transform

Qian Kemao
School of Computer Engineering, Nanyang Technological University, Singapore 639798
mkmqian@ntu.edu.sg

ABSTRACT

Fourier transform is a reserved teaching topic for science and engineering students in almost all universities. Among the large audience, some of them feel it is too difficult. This article is written for them in a casual and friendly way, with the humble aim of giving them more confidence in reading formal textbooks. A few features and efforts have been introduced in this article, such as linking the concept to the cel animation, starting from a 2-D discrete Fourier transform rather than a 1-D continuous Fourier transform, and detailed derivation to rediscover the Fourier transform.

Keywords: Fourier transform, signals, image decomposition, frequency, amplitude

1. INTRODUCTION

Some students feel that Fourier transform is something *very* difficult and scary. Unfortunately, if you ask somebody who knows it, the answer is often like "Fourier transform? It is *very* easy. Just a transform to another domain using harmonics." This encouraging message is too discouraging to many of us. So, there must be a hill between two extremes. We have to climb over it in order to see the sceneries on the other side. Let's do it together in this article. Prepare a pen and a paper. We will also use MATLAB® for demonstration. As Fourier transform is a tool invented by Jeseph Fourier to transform signals, we start our journey from signals.

2. SIGNALS ARE OUR FRIENDS

A signal is a list of numbers carrying some information. For example, you take your body temperature and record it down every day, you get a signal. If the new reading today is much higher than the earlier recorded ones, likely you have a fever. This is the information carried by the signal, based on which you can decide whether you should see a doctor.

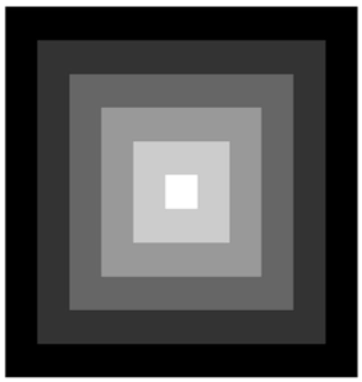
If the signal value depends on one variable, it is called a one-dimensional (1-D) signal. Our temperature signal above serves as such an example, which only depends on time. We write it as $f(t)$, meaning that at the time *t*, the signal value is $f(t)$. If the signal value depends on two variables, it is called a 2-D signal. An image is a perfect example. In fact we often vaguely call a 2-D signal as an image. We write an image as $f(x, y)$, meaning that at the location (x, y), the signal value is $f(x, y)$. Nobody can stop the game from getting more variables and thus a signal can also be 3-D or even higher dimensional.

In this article, we work on images because they are easily available. We can always take images using our cell phones, read them into our computers and process them for fun. The two variables of such images take integer values as $x = 0,1,2,\cdots,M-1$ and $y = 0,1,2,\cdots,N-1$. We often call each pair of (x, y) as a pixel. Since an image is a list of numbers arranged in 2-D along *x* and *y*, it can be written as

$$\begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \cdots & f(2,N-1) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \cdots & f(M-1,N-1) \end{bmatrix}. \quad (1)$$

We call it an $M \times N$ matrix because there are M rows and N columns.

You may feel a little weird. It does not look like an image that we have experienced before. Yes, you are right. We as human beings feel the brightness, but computers swallow data. So there is a mapping: larger value ↔ brighter, and smaller value ↔ darker. This is similar to the body temperature example. We feel the fever but the thermometer shows a number like 38.5°C. There is also a mapping: standard value ↔ normal, and larger value ↔ fever. Figure 2 below is an example for you to get familiar with the mapping. With the time going on, we should be happy with both representations.



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 0 \\ 0 & 50 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 50 & 0 \\ 0 & 50 & 100 & 150 & 150 & 150 & 150 & 150 & 100 & 50 & 0 \\ 0 & 50 & 100 & 150 & 200 & 200 & 200 & 150 & 100 & 50 & 0 \\ 0 & 50 & 100 & 150 & 200 & 250 & 200 & 150 & 100 & 50 & 0 \\ 0 & 50 & 100 & 150 & 200 & 200 & 200 & 150 & 100 & 50 & 0 \\ 0 & 50 & 100 & 150 & 150 & 150 & 150 & 150 & 100 & 50 & 0 \\ 0 & 50 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 50 & 0 \\ 0 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 1. A picture we see (left) and its number matrix (right)

Time for going out for a break. You can open the door without any problems because you can see everything surrounding you. Here the signals are a series of images you take by your eyes. You use them to judge where the door is and open it. Yes, signals are our good friends because they carry the information we need. If a computer reads them, they become 2-D matrices.

3. IMAGE DECOMPOSITION AND COMPOSITION

I believe you had ever watched some traditional 2-D animations like *Tom and Jerry* (my son's favorite) when you were a child. They are called cel animation. The cel stands for a material called celluloid which is transparent. Animators draw on a few cel layers and composite them into one image. The idea is illustrated in Fig. 2.

Why do we bother to have different layers? Why not directly draw the complete image? The answer is to make the drawing easier. For example, if the red character (a rabbit?) hides away, we do not need to draw anything at all. We just composite the first and the third layers. The idea can be summarized as: decomposing an image into layers, manipulating the layers, and compositing a new image. This idea looks very simple, but it is a great idea and has been used as a common practice in the animation industry.

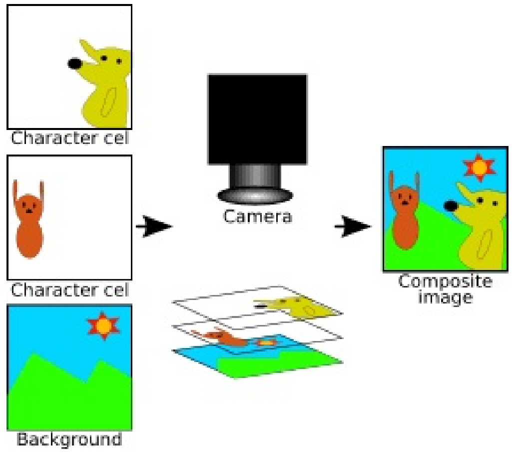


Figure 2. Cel animation (figure taken from http://en.wikipedia.org/wiki/Traditional_animation)

We want to apply this great idea to our images too, because we also need to process images very often. If we can decompose an image into layers, the processing is hopefully easier. Our decomposition can be written as

$$f(x,y) = l_0(x,y) + l_1(x,y) + l_2(x,y) + \cdots l_{L-1}(x,y) = \sum_{i=0}^{L-1} l_i(x,y), \tag{2}$$

where L is the number of layers. Unfortunately, it is quite difficult to give each layer $l_i(x,y)$ a semantic meaning such as foreground characters and background sceneries. Instead, we let $l_i(x,y)$ be a mathematical function, which is more abstract. Lacking the sematic meaning at the beginning might be the *first difficult point* for us. Nevertheless, we will try to find the meaning of the functions later. (Historically, the great idea was applied to signal analysis much earlier than the cel animation, because the Fourier transform is about 200 years old while the cel animation is about 100 years old.)

To make ourselves happier, it is good that the functions in different layers are similar, and it is even better if we have learned these functions before. What functions do we know? Many of us like polynomial functions most. Indeed somebody called Brook Taylor had already tried to decompose a signal into polynomials. You can find the so-called Taylor series in your Calculus book and will also find that it is about 300 years old! However, this decomposition is not ideal for image processing. Now use your pen to sketch $f(x) = x^4$, you will find that the function value goes up too quickly. That is why the Taylor's decomposition is useful only in a small region, but we want to decompose the entire image. In this article we try *sine* and *cosine* functions. Now you sketch $\sin x$ and $\cos x$, and you will find they are more obedient: they go up and down but never run away. We simply say that "they wave", or "they have waving structures".

The sine and cosine functions are bundled into *complex exponential functions* by the following Euler's formula,

$$\exp(jx) = \cos x + j\sin x, \tag{3}$$

where $j = \sqrt{-1}$ is the imaginary unit and x is an arbitrary real number. Using complex numbers makes our derivations and repetitions easier and more concise. However, some of you need to fight with both trigonometric functions and complex numbers at the same time, which is the *second difficult point*, I believe. I will give you all the necessary details in my derivations below so that you can follow them easily.

Time for a break again. Please do the following homework.

Q1: Please sketch $f_1(x)=\cos(\pi x/4)$, $f_2(x)=\sin(\pi x/4)$, $f_3(x)=\cos(\pi x/2)$, and $f_4(x)=2\cos(\pi x/4)$, for $x=0,1,2,\cdots,31$. Compare the functions: $f_1(x)$ vs. $f_2(x)$, $f_1(x)$ vs. $f_3(x)$, and $f_1(x)$ vs. $f_4(x)$. [**A1**: (i) $f_2(x)$ is a shifted version of $f_1(x)$ because $f_2(x)=\cos(\pi x/4-\pi/2)=f_1(x-2)$; (ii) $f_3(x)$ moves up and down faster than $f_1(x)$; (iii) $f_4(x)$ moves up and down more energetically than $f_1(x)$.]

Q2: Let $\omega=\pi/16$. Is $\cos(\omega x)$ periodic? What is the period? How about $\sin(\omega x)$ and $\exp(j\omega x)$? [**A2**: All three functions are periodical with the same period of 32.]

Q3: Prove $\exp(j2k\pi)=1$ for any integer *k*. [**A3**: Consider the real part and imaginary part, separately.]

Q4: Based on Q1, please try to create general forms of the cosine and complex exponential functions, respectively. [**A4**: A general form for the cosine function is $f(x)=A\cos(2\pi\xi x+\theta)$ where *A* is called amplitude, ξ is called frequency, and θ is called initial phase. A general form for the complex exponential function is $f(x)=A\exp\left[j(2\pi\xi x+\theta)\right]=\left[A\exp(j\theta)\right]\exp(j2\pi\xi x)=A'\exp(j2\pi\xi x)$ where $A'=A\exp(j\theta)$ is a complex amplitude. The exponential form becomes simpler. The magnitude and phase of A' are *A* and θ, respectively.

Q5: Sketch $f(x,y)=\exp(j\pi x/4+j\pi y/2)$. [**A5**: First, we cannot sketch a complex function directly. So we sketch its real part and imaginary part, separately. Second, it is hard to draw a 2-D wave. So we consider the contours. A contour is a line or a curve with the same value.]

4. OUR LAYERS HAVE STRUCTURES

We use the following 2-D layer functions,

$$\tilde{l}_{(\xi,\eta)}(x,y)=\tilde{F}(\xi,\eta)\tilde{b}_{(\xi,\eta)}(x,y)=\tilde{F}(\xi,\eta)\exp(j2\pi\xi x+j2\pi\eta y), \quad (4)$$

which are actually general complex exponential functions in their general form (Q4). Recall that $x=0,1,2,\cdots,M-1$ and $y=0,1,2,\cdots,N-1$. Here we have introduced a pair of parameters (ξ,η) for indexing, which can take any values. A different pair of (ξ,η) means a different layer. Each layer consists of two parts:

1) $\tilde{b}_{(\xi,\eta)}(x,y)$ represents a waving structure. Its real part is sketched in Fig. 3. Its imaginary part is similar but spatially shifted (see Q1). The horizontal and vertical waving periods are illustrated. It should be easy to find that $\xi=1/T_x$ and $\eta=1/T_y$, which obtain the name of *frequency*. Increasing or decreasing ξ will make the wave vertically denser or sparser. It is the same for η which acts horizontally. In most books, the authors do not use the word layer. Instead, they call each layer a *frequency component*. Can you accept it? This is actually the semantic meaning of a layer: each is a fast or slow waving structure. The layer structure is investigated in this section;
2) $\tilde{F}(\xi,\eta)$ is a complex amplitude of the structure $\tilde{b}_{(\xi,\eta)}(x,y)$. Its magnitude tells whether this layer is energetic and its phase tells whether this layer is spatially shifted (see Q1 and Q4). The complex amplitude is investigated in the next section.

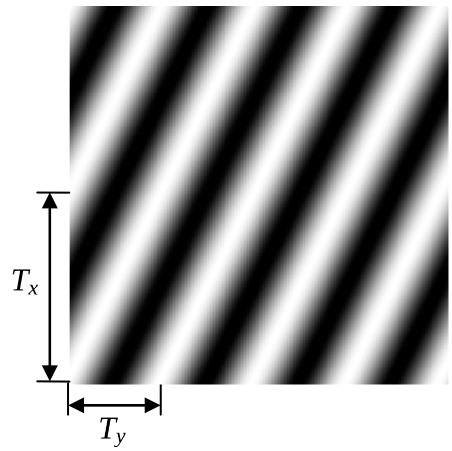


Figure 3. A general waving structure

Just now we said that (ξ,η) could take any values. It means that we have too many layers, which is not good. Fortunately, we can discover the periodicity of (ξ,η). For any integers *m* and *n*, we have

$$
\begin{aligned}
&\tilde{b}_{(\xi+m,\eta+n)}(x,y)\\
&=\exp\left[j2\pi(\xi+m)x+j2\pi(\eta+n)y\right]\leftarrow\left[\because Eq.(4)\right]\\
&=\exp\left(j2\pi\xi x+j2m\pi x+j2\pi\eta y+j2n\pi y\right)\\
&=\exp\left[j2\pi\xi x+j2\pi\eta y+j2\pi(mx+ny)\right]\\
&=\exp\left(j2\pi\xi x+j2\pi\eta y\right)\exp\left[j2\pi(mx+ny)\right]\leftarrow\left[\because \exp(\alpha+\beta)=\exp(\alpha)\exp(\beta)\right]\\
&=\exp\left(j2\pi\xi x+j2\pi\eta y\right)\leftarrow\left[\because \text{Q3}\right]\\
&=\tilde{b}_{(\xi,\eta)}(x,y)\leftarrow\left[\because Eq.(4)\right]
\end{aligned}
\quad . \qquad (5)
$$

It means that the layers (ξ,η), $(\xi+1,\eta)$, $(\xi,\eta+1)$, $(\xi+2,\eta)$, $(\xi,\eta+2)$, $(\xi+1,\eta+1)$, …, have the same waving structure. One is enough. This is important because now our (ξ,η) can be taken only from $0\le\xi<1$ and $0\le\eta<1$.

We now decide the number of layers we need. We worry that too few layers might not be sufficient to represent an image, but too many layers might not be convenient for us. Two numbers are quite special: *M*, the size of the image along *x*, and *N*, the size of the image along *y*. So this is how we guess:

$$\xi=\frac{0}{M},\frac{1}{M},\frac{2}{M},\cdots,\frac{M-1}{M}, \qquad (6)$$

$$\eta=\frac{0}{N},\frac{1}{N},\frac{2}{N},\cdots,\frac{N-1}{N}. \qquad (7)$$

We need to come back here if our guess does not work, but luckily it works, as we will see later. The waves with such neatly increasing frequencies are called *harmonics*.

We can see that (ξ,η) are fractions. Recall that (ξ,η) is used to index layers. We thus amplify them into integers for convenience by the following one-to-one mapping,

$$u = M\xi = 0,1,2,\cdots,M-1, \tag{8}$$
$$v = N\eta = 0,1,2,\cdots,N-1, \tag{9}$$

with which we define $l_{(u,v)}(x,y) = \tilde{l}_{(\xi,\eta)}(x,y)$, $F(u,v) = \tilde{F}(\xi,\eta)$, and $b_{(u,v)}(x,y) = \tilde{b}_{(\xi,\eta)}(x,y)$. Now the layers become

$$l_{(u,v)}(x,y) = F(u,v)l_{(u,v)}(x,y) = F(u,v)\exp\left(j2\pi ux/M + j2\pi vy/N\right). \tag{10}$$

You feel painful? Just a moment. Let's substitute Eq. (10) into Eq. (2) and replace the index *i* by (u,v), we get the following,

$$f(x,y) = \frac{1}{MN}\sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)\exp\left(j2\pi ux/M + j2\pi vy/N\right), \tag{11}$$

where we have scaled the summation by $1/(MN)$ as a convention. Here comes the good news: this equation is called *inverse Fourier transform*. So, do not give up. We are almost there! Take a break. No homework if you can recognize layers in Eq. (11).

5. OUR LAYERS HAVE COMPLEX AMPLITUDES

When an image is decomposed into layers, we need to know the complex amplitude of each layer, $F(u,v)$. But how? We want to solve this riddle. You may jump to Eq. (22) if you just want the result. Because many of us are quite curious on the riddle solving, we work patiently towards it.

At this moment, I cannot help calling the structure of each layer,

$$b_{(u,v)}(x,y) = \exp\left(j2\pi ux/M + j2\pi vy/N\right), \tag{12}$$

as a boy with a name (u,v). Who can recognize his amplitude? Yes, his girlfriend. Who is she then? The conjugate of the boy,

$$b^*_{(u,v)}(x,y) = \exp\left(-j2\pi ux/M - j2\pi vy/N\right). \tag{13}$$

Her name is also called (u,v). Since (u,v) is a general symbol for indexing, let's specify one particular boy with the name of (u',v') for further consideration. We send his girlfriend $b^*_{(u',v')}(x,y)$ to hug the entire image by multiplying her with it, and we then have

$$\begin{aligned}
&\sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y)b^*_{(u',v')}(x,y) \\
&= \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y)\exp\left(-j2\pi u'x/M - j2\pi v'y/N\right) \leftarrow [\because \text{Eq.(13)}] \\
&= \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\left\{\begin{array}{l}\left[\dfrac{1}{MN}\sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)\exp\left(j2\pi ux/M + j2\pi vy/N\right)\right] \\ \times\exp\left(-j2\pi u'x/M - j2\pi v'y/N\right)\end{array}\right\} \leftarrow [\because \text{Eq.(11)}] \\
&= \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\sum_{v=0}^{N-1}\sum_{u=0}^{M-1}\frac{1}{MN}F(u,v)\exp\left[j2\pi(u-u')x/M + j2\pi(v-v')y\right] \leftarrow [\because \text{rearrangement}] \\
&= \sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)\left\{\frac{1}{N}\sum_{y=0}^{N-1}\exp\left[j2\pi(v-v')y/N\right]\right\}\left\{\frac{1}{M}\sum_{x=0}^{M-1}\exp\left[j2\pi(u-u')x/M\right]\right\} \leftarrow [\because \text{change the sum order}] \\
&= \sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)H_yH_x \leftarrow [\because \text{notations}]
\end{aligned} \tag{14}$$

We evaluate H_x first. If $u = u'$, namely, u happens to be the boyfriend of u', then

$$H_x = \frac{1}{M}\sum_{x=0}^{M-1}\exp\left[j2\pi\left(u-u'\right)x/M\right] = \frac{1}{M}\sum_{x=0}^{M-1}\exp\left(0\right) = \frac{1}{M}\sum_{x=0}^{M-1}1 = \frac{M}{M} = 1, \tag{15}$$

which means that they have strong feeling and response between them. If $u \neq u'$, namely, u is NOT the boyfriend of u', then

$$\begin{aligned} H_x &= \frac{1}{M}\sum_{x=0}^{M-1}\exp\left[j2\pi\left(u-u'\right)x/M\right] \\ &= \frac{1}{M}\sum_{x=0}^{M-1}\left\{\exp\left[j2\pi\left(u-u'\right)/M\right]\right\}^x \leftarrow \left[\because \exp\left(\alpha\beta\right) = \left[\exp\left(\alpha\right)\right]^\beta\right] \end{aligned}. \tag{16}$$

We have learned in our secondary school that

$$\sum_{x=0}^{M-1}q^x = 1 + q + q^2 + \cdots + q^{M-1} = \frac{1-q^M}{1-q}. \tag{17}$$

Let

$$q = \exp\left[j2\pi\left(u-u'\right)/M\right], \tag{18}$$

we have

$$\begin{aligned} H_x &= \frac{1}{M}\sum_{x=0}^{M-1}q^x \leftarrow \left[\because Eq.(18)\right] \\ &= \frac{1}{M}\frac{1-q^M}{1-q} \leftarrow \left[\because Eq.(17)\right] \\ &= \frac{1}{M}\frac{1-\left\{\exp\left[j2\pi\left(u-u'\right)/M\right]\right\}^M}{1-\exp\left[j2\pi\left(u-u'\right)/M\right]} \leftarrow \left[\because Eq.(18)\right] \\ &= \frac{1}{M}\frac{1-\exp\left[j2\pi\left(u-u'\right)\right]}{1-\exp\left[j2\pi\left(u-u'\right)/M\right]} \leftarrow \left[\because \left[\exp\left(\alpha\right)\right]^\beta = \exp\left(\alpha\beta\right)\right] \\ &= \frac{1}{M}\frac{1-1}{1-\exp\left[j2\pi\left(u-u'\right)/M\right]} \leftarrow \left[\because \text{Q3} \quad \& \quad \left(u-u'\right) \quad \text{is} \quad \text{an} \quad \text{integer}\right] \\ &= 0 \end{aligned}, \tag{19}$$

which means that the girl has no feeling with this boy at all because he is not her girlfriend. We can evaluate H_y with a similar result. Put together, we have

$$H_y H_x = \begin{cases} 1, & if \quad \left(u,v\right) = \left(u',v'\right) \\ 0 & otherwise \end{cases}. \tag{20}$$

With this result we return to Eq. (14). Obviously only when $\left(u,v\right) = \left(u',v'\right)$, the $F\left(u,v\right)$ term is retained and otherwise disappears, so the final result is

$$\sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f\left(x,y\right)\exp\left(-j2\pi u'x/M - j2\pi v'y/N\right) = F\left(u',v'\right). \tag{21}$$

Because $\left(u',v'\right)$ can be arbitrary, we rewrite Eq. (21) as

$$F\left(u,v\right) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f\left(x,y\right)\exp\left(-j2\pi ux/M - j2\pi vy/N\right), \tag{22}$$

which is called Fourier transform or *forward Fourier transform*! The complex amplitude $F(u,v)$ has a formal technical name of Fourier spectrum. It is also a number matrix having the same size of the original image,

$$\begin{bmatrix} F(0,0) & F(0,1) & F(0,2) & \cdots & F(0,N-1) \\ F(1,0) & F(1,1) & F(1,2) & \cdots & F(1,N-1) \\ F(2,0) & F(2,1) & F(2,2) & \cdots & F(2,N-1) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ F(M-1,0) & F(M-1,1) & F(M-1,2) & \cdots & F(M-1,N-1) \end{bmatrix}. \quad (23)$$

Amazingly we rediscover the Fourier transform by ourselves! Let's put our two achievements together,

$$\boxed{\begin{aligned} F(u,v) &= \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y)\exp(-j2\pi ux/M - j2\pi vy/N) \\ f(x,y) &= \frac{1}{MN}\sum_{v=0}^{N-1}\sum_{u=0}^{M-1} F(u,v)\exp(j2\pi ux/M + j2\pi vy/N) \end{aligned}} \quad . \quad (24)$$

Before we happily take a break, let's clear one more thing. We have an image $f(x,y)$ with a size of *MN*. We decompose it into *MN* layers, and each has the same size as $f(x,y)$. It means that we totally have a size of $(MN)^2$, much larger than the original image. This seems not good. However, for each layer $l_{(u,v)}(x,y)$, we actually do not need to store the entire waving structure. Given a particular (u,v), the waving structure is always fixed, like the one shown in Fig. 2. Do we need to remember (u,v) then? Neither. We need all necessary (u,v) and will list them all out. What we need to store is only $F(u,v)$. Thus the Fourier transform is often written as $f(x,y) \Leftrightarrow F(u,v)$. If we know one, we know the other. Both sides have the same size of *MN*. Nowadays we seldom code the Fourier transform by ourselves because it is so popular that people have already done it for us. For example, in MATLAB®, we use `fft2` and `ifft2` for forward and inverse Fourier transforms, respectively.

6. USE IT!

We are now able to see the sceneries: we should have no difficulty to understand low-pass, high-pass and band-pass filters. Before further explanation, we need to make it very clear on how (u,v) correspond to the waving speed. As we expect, the layer $(0,0)$ has the lowest waving speed: actually it is constant and does not wave at all. But it is quite contrast to our intuition that, the layer $(M-1,N-1)$ is not the fastest wave. From the following fact,

$$b_{(M-1,N-1)}(x,y) = \exp\left[j2\pi(M-1)x/M + j2\pi(N-1)y/N\right] = \exp(-j2\pi x/M - j2\pi y/N), \quad (25)$$

we find that it waves very slow and is only a little faster than the layer $(0,0)$. The fastest one happens at the layer $(M/2, N/2)$ if *M* and *N* are even numbers (Why? Please find the fastest layer if *M* and *N* are odd numbers.) Thus it is important to note that in the matrix (23), the center coefficients are for fast waving layers and those near borders are for slow waving layers.

It is more comfortable to have slow wave layers at the center. This can be easily achieved because the layers are periodic about (u,v). So we repeat the matrix (23) and select one period with $F(0,0)$ in the center, which is shown in the yellow box in Fig. 4. In MATLAB®, `fft2` gives the spectrum in the form of the matrix (23), but we can switch it to the form in Fig. 4 by using `fftshift` and switch it back by using `ifftshift`.

$$
\begin{matrix}
\begin{bmatrix}
F(0,0) & \cdots & F(0,N/2-1) & F(0,N/2) & \cdots & F(0,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M/2-1,0) & \cdots & F(M/2-1,N/2-1) & F(M/2-1,N/2) & \cdots & F(M/2-1,N-1) \\
F(M/2,0) & \cdots & F(M/2,N/2-1) & F(M/2,N/2) & \cdots & F(M/2,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M-1,0) & \cdots & F(M-1,N/2-1) & F(M-1,N/2) & \cdots & F(M-1,N-1)
\end{bmatrix} &
\begin{bmatrix}
F(0,0) & \cdots & F(0,N/2-1) & F(0,N/2) & \cdots & F(0,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M/2-1,0) & \cdots & F(M/2-1,N/2-1) & F(M/2-1,N/2) & \cdots & F(M/2-1,N-1) \\
F(M/2,0) & \cdots & F(M/2,N/2-1) & F(M/2,N/2) & \cdots & F(M/2,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M-1,0) & \cdots & F(M-1,N/2-1) & F(M-1,N/2) & \cdots & F(M-1,N-1)
\end{bmatrix} \\
\begin{bmatrix}
F(0,0) & \cdots & F(0,N/2-1) & F(0,N/2) & \cdots & F(0,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M/2-1,0) & \cdots & F(M/2-1,N/2-1) & F(M/2-1,N/2) & \cdots & F(M/2-1,N-1) \\
F(M/2,0) & \cdots & F(M/2,N/2-1) & F(M/2,N/2) & \cdots & F(M/2,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M-1,0) & \cdots & F(M-1,N/2-1) & F(M-1,N/2) & \cdots & F(M-1,N-1)
\end{bmatrix} &
\begin{bmatrix}
F(0,0) & \cdots & F(0,N/2-1) & F(0,N/2) & \cdots & F(0,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M/2-1,0) & \cdots & F(M/2-1,N/2-1) & F(M/2-1,N/2) & \cdots & F(M/2-1,N-1) \\
F(M/2,0) & \cdots & F(M/2,N/2-1) & F(M/2,N/2) & \cdots & F(M/2,N-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
F(M-1,0) & \cdots & F(M-1,N/2-1) & F(M-1,N/2) & \cdots & F(M-1,N-1)
\end{bmatrix}
\end{matrix}
$$

Figure 4. Another form of the spectrum which is more often used in spectral analysis

We have got everything ready. But why do we bother to do all these? We now return to the great idea of image decomposition and composition. We decompose an image into layers characterized by $F(u,v)$, alter or select these layers according to our wish, and then composite an new image. We illustrated this procedure in Fig. 5, which is a standard procedure that you can find in many books. An overhead bar is used to indicate that the signal has been altered.

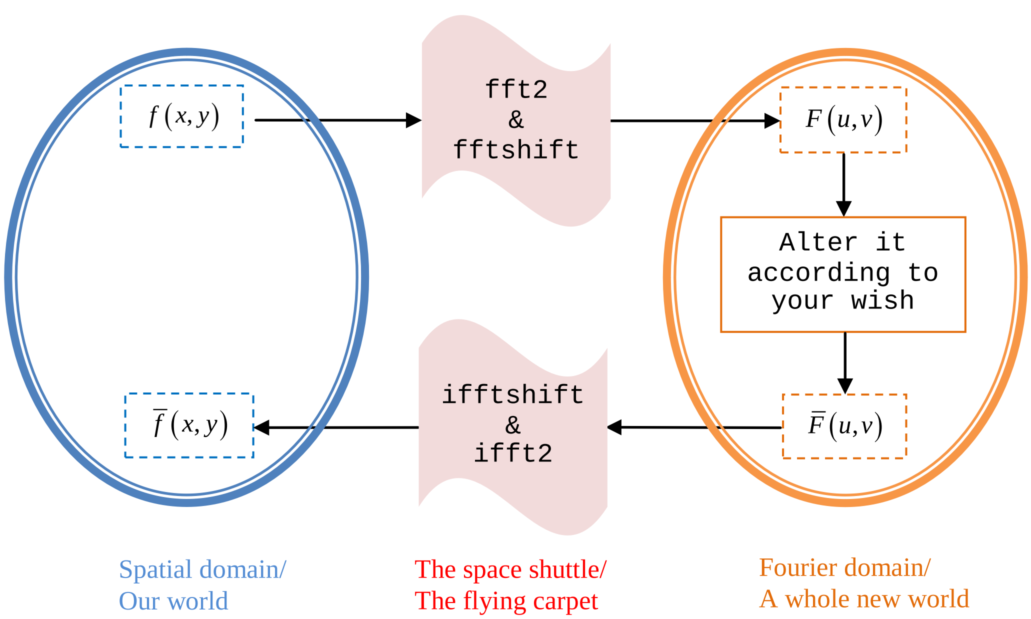
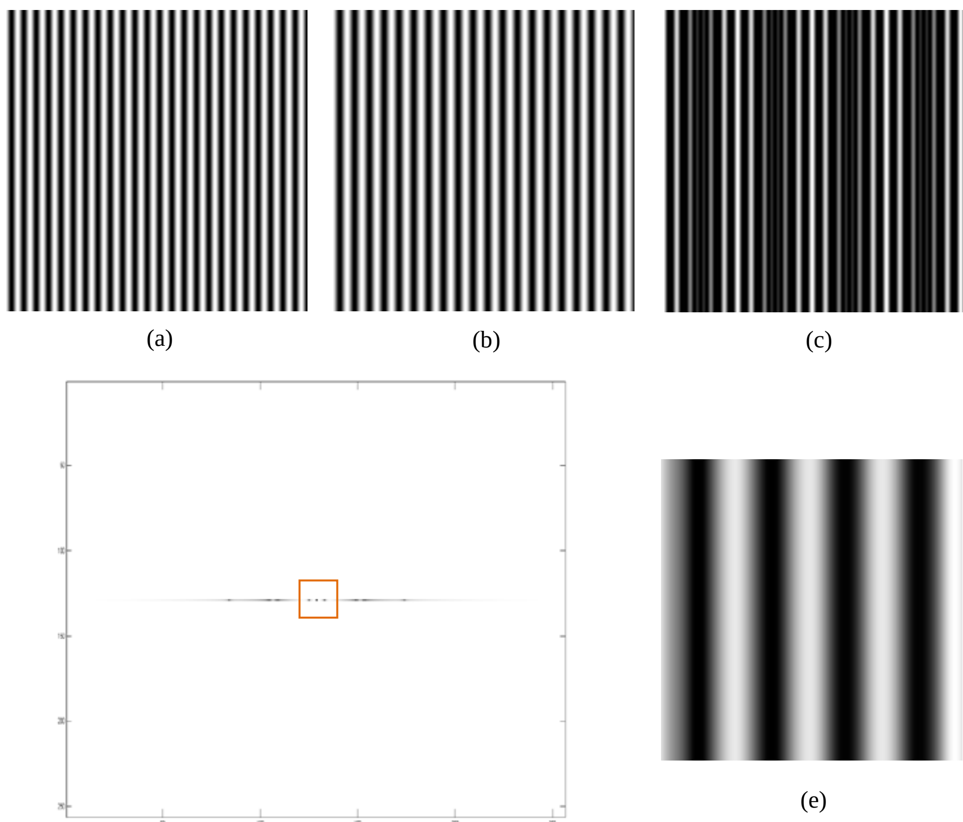


Figure 5. A standard procedure

Assume now you have an image. You want to remove details. Details mean sudden and fast changes. So we remove the layers with high waving speed, or in other words, high frequency components. To be more specific, we set $F(u,v)$ to zero if (u,v) are large, so that they are blocked. In Fig. 4, they are located farther to the center of the yellow box. The low waving speed coefficients, located near the center of the yellow box in Fig. 4, survive the aforementioned operation. We call it a low-pass filter. The word "low" means that the waving speed or the frequency is low. A high-pass filter is on the opposite which keeps the details. A band-pass filter stands between the low-pass and high-pass filters. Obviously the choice of the filter depends on your purpose.

Here comes one example. We want to test how strong a material is. To do so, we make stripes onto it, as shown in Fig. 6(a). The image size is 256×256, namely, $M = N = 256$. We then stretch the material. The stripes become Fig. 6(b). We want to find how the period of the stripes changes. This can be easily achieved by counting the number of stripes in each image and calculate the average spacing. There is a problem: in our real life, the material is often non-uniform and cannot take average like this. One solution is to overlay two images. The product of two images in Fig. 6(a) and 6(b) is shown in Fig. 6(c). We see some new stripes from which we can get our desired information. It is called a moiré pattern. You can try to find out how it relates to the deformation of the material.



(a) (b) (c)

(d) (e)

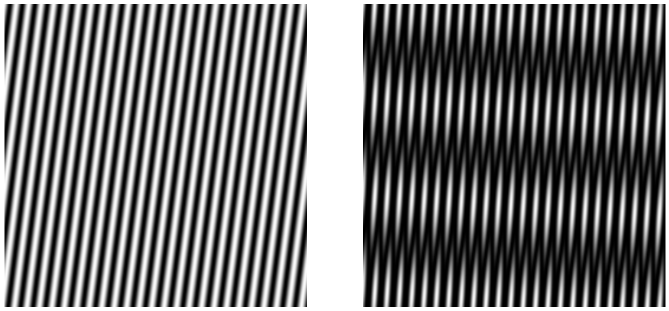
Figure 6. Process a moiré pattern in the Fourier domain

We see four cycles in Fig. 6(c) so easily but computers may feel frustrated. Something must be done to comfort them. We do a Fourier transform on Fig. 6(c), which gives the spectrum in Fig. 6(d) in the form of Fig. 4. What we show is $-\log\left(\left|F\left(u,v\right)\right|\right)$. We show the *absolute value* because $F\left(u,v\right)$ are complex numbers, and we show *negative logarithm* for better information visualization. The spectrum is enlarged in size in order to see it more clearly. We do a low pass filtering by retaining only the coefficients in the yellow box, followed by an inverse Fourier transform. We get Fig. 6(e). I believe that our computers will be happier. Is it a too simple toy example? No. This is a useful technique for deformation measurement and researchers are still using it even today. I have shown my MATLAB® code in Fig. 7. You can copy it to your MATLAB® editor and run it.

```
%---------------------------------------------------------------------------
% 1, Generate images in our world
%
[y x]=meshgrid(1:256,1:256);                      %pixel coordinates
f1=1+cos(0.6*y);                                  %generate the first image
subplot(2,3,1); imagesc(f1); colormap(gray);      %display
f2=1+cos(0.5*y);                                  %generate second image
subplot(2,3,2); imagesc(f2);                      %display
f3=f1.*f2;                                        %generate the moiré pattern
subplot(2,3,3); imagesc(f3);                      %display


%---------------------------------------------------------------------------
% 2, Fly to the new world
%
f4=fftshift(fft2(f3));                            %fft2 & fftshift
subplot(2,3,4); imagesc(-log(abs(f4)));           %display


%---------------------------------------------------------------------------
% 3, Work in the new world
%
f5=f4*0;                                          %initialize a new spectrum
f5(128-8:128+8,128-8:128+8)= ...
    f4(128-8:128+8,128-8:128+8);                  %keep low frequency
subplot(2,3,5); imagesc(-log(abs(f5)));           %display


%---------------------------------------------------------------------------
% 4, Fly back to our world
%
f6=ifft2(ifftshift(f5));                          %ifftshift & ifft2
subplot(2,3,6); imagesc(real(f6));                %display
%---------------------------------------------------------------------------
```

Figure 7. MATLAB® code for getting Fig. 6

Here is your computing homework: (i) generate stripes like Fig. 6(a); (ii), rotate it a little bit and generate an image like Fig. 8(a); (iii) calculate their product and observe the moiré pattern like Fig. 8(b); (iv) find a way to smooth the moiré pattern to please your computer; (v) think about its potential usage.



(a) (b)

Figure 8. Another moiré pattern

7. CONCLUSIONS

Fourier transform brings us to the layer world, which is more professionally called Fourier domain. I often call it "a whole new world". These days my son happens to be learning and practicing *A Whole New World* from *Aladdin*. Let's give us a reward by watching it at http://www.youtube.com/watch?v=-kl4hJ4j48s. Our Fourier transform processes a discrete image where $x = 0, 1, 2, \cdots, M-1$ and $y = 0, 1, 2, \cdots, N-1$. If x and y are continuous, we need the so-called continuous Fourier transform. Its theory is more difficult to prove but the concepts and transforms look quite similar. Although Fourier transform is magical, it is not free from problems. The biggest problem is its inefficiency in layer decomposition: the wave always splashes the entire layer. The long story about windowed Fourier transform, wavelet, curvelet, …, starts from here. Some further readings are given in [1-4].

ACKNOWLEDGEMENTS

This article was initially written for my CPE412/CSC408 students. The comments from Dr Weisi Lin, Dr Haixia Wang, and Ms Prerna Chikersal are appreciated. Thanks to Dr Seah Hock Soon for his encouragement and enthusiastic vision on e-Learning. Thanks to my family members, especially my 7-year old son. He stared at me when I typed this article during weekends, as if I was playing magic. He gave me a lot of satisfaction and energy. The work was partially supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd edition)*, (Pearson Education Inc., 2010).

[2] A. V. Oppenheim, A. S. Willsky and S. H. Nawab, *Signals & Systems (Second Edition)*, (Prentice Hall International, 1997).

[3] A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-time Signal Processing (Second Edition)*, (Prentice Hall International, 1999).

[4] S. Mallat, *A Wavelet Tour of Signal Processing: the Sparse Way (Third Edition)*, (Academic Press, San Diego, 2009).