Discrete Fourier Transform

CITS4240 Computer Vision
Drafted by Du Huynh
March 2009

When moving from the continuous case to the discrete case, we need to firstly work out what our sampling rate is and at what frequency values we can compute the Fourier transform.

Suppose that our data are collected at the interval Δt (in seconds). If we have altogether T data terms, then our time axis is

$$\begin{aligned} t &= [\,0, \Delta t, 2\Delta t, \cdots, (T-1)\Delta t\,] \\ &= \Delta t\,[\,0, 1, 2, \cdots, T-1\,]\,. \end{aligned} \tag{1}$$

If Δt is the ***sampling interval***, then

$$s = \frac{1}{\Delta t} \tag{2}$$

is the ***sampling frequency*** or ***sampling rate*** (in Hz).

According to the Nyquist-Shannon sampling theorem: if an analog signal has been sampled at a frequency s, then the signal can be reconstructed up to the highest frequency $s/2$. This means that any fine details at frequency values above $s/2$ cannot be reconstructed. With this theorem in mind, we can work out that our frequency axis in the frequency domain is

$$\begin{aligned} \omega &= \left[-\frac{s}{2}, -\frac{s}{2}+\frac{s}{T}, -\frac{s}{2}+\frac{2s}{T}, \cdots, \frac{s}{2}-\frac{s}{T}\right] \\ &= s\left[-\frac{1}{2}, -\frac{1}{2}+\frac{1}{T}, -\frac{1}{2}+\frac{2}{T}, \cdots, \frac{1}{2}-\frac{1}{T}\right]. \end{aligned} \tag{3}$$

You may notice that the number of frequency values in the ω variable is also T, the same length as the time axis. Furthermore, the absolute values of all the terms in (3) are less than or equal to $s/2$.

Let $k, n = 0, 1, \cdots, T-1$. Let

$$\begin{aligned} t_k &= k\Delta t \\ \omega_n &= s\left(-\frac{1}{2}+\frac{n}{T}\right), \end{aligned}$$

and let the discrete sampled values of our signal be $f(t_k)$, for $k = 0, 1, \cdots, T-1$. The **Discrete Fourier Transform (DFT)** of f is defined as follows:

$$F(\omega_n) = \sum_{k=0}^{T-1} f(t_k) e^{-i2\pi\omega_n t_k/T}, \quad \text{for } n = 0, 1, \cdots, T-1, \tag{4}$$

and the **inverse Discrete Fourier Transform (IDFT)** is given by

$$f(t_k) = \frac{1}{T}\sum_{n=0}^{T-1} F(\omega_n) e^{i2\pi\omega_n t_k/T}, \quad \text{for } k = 0, 1, \cdots, T-1. \tag{5}$$

Note that there is a factor of $1/T$ for the definition of IDFT. This factor is not present for the continuous case.
Notice that because the signal has finite length, in order to have the Discrete Fourier Transform of the signal defined, we actually replicate the signal to make it periodic.
Furthermore, since signal f is often real (rather than complex) in practice, $F(-\omega)$ is simply the **complex conjugate** of $F(\omega)$, i.e.,

$$F(-\omega) = F^*(\omega).$$

If we have the DC component (i.e., the zero frequency component) at the centre of the array, the plot of the Fourier spectrum, $|F(\cdot)|$, is symmetrical.
Although the implementation of the Matlab functions `fft` and `ifft` is based on the definitions given in Eqs. (4) and (5), it uses the frequency values defined in (**??**). Matlab therefore provides the function `fftshift`, which allows us to shift the zero frequency to the centre.

Consider the term $\exp(-i2\pi\omega_n t_k/T)$ in Eq. (4), for some integers n and k. We can see that ω_n involves the factor s while t_k involves the factor $\Delta t = 1/s$. So

$$\begin{aligned}\exp(-i2\pi\omega_n \, t_k/T) &= \exp\left(-i2\pi s\left(-\frac{1}{2}+\frac{n}{T}\right)k\frac{\Delta t}{T}\right) \\ &= \exp\left(-i2\pi\left(-\frac{1}{2}+\frac{n}{T}\right)\frac{k}{T}\right).\end{aligned}$$

The above derivation shows that, if the sampling frequency s or sampling interval Δt is not known, it is usually assumed that $\Delta t = 1$ and $s = 1$ (this is the assumption adopted in Matlab). After having the discrete values $F(\omega_n)$ computed, we define the frequency values in accordance with the assumed sampling frequency, thus $\omega_n = -\frac{1}{2} + \frac{n}{T}$, for $n = 0, 1, \cdots, T-1$.

Given below is a Matlab script showing how to compute the Discrete Fourier Transform of a 1D signal that is composed of two sine waves of frequencies 0.8 Hz and 3 Hz (see Figure 1). Note that the Fourier spectrum shown in Figure 2(a) has peaks at these two frequency values. Sometimes it is useful to plot the Fourier spectrum of a signal in logarithmic scale so that frequencies having low amplitudes are more visible (see Figure 2(b)). As the input signal is real (rather than complex), the Fourier spectrum is symmetrical about the zero frequency.

```
% File name: test.m
% A small test on the fft function of Matlab.
% The Matlab function 'fft' implements the Discrete Fourier Transform.
%
% Du Huynh, March 2009.

% f consists of two sine waves of frequencies 3 Hz and 0.8 Hz.
% The amplitudes of these sine waves are 7 and 4 units, respectively.
freq = [3; 0.8];
amp = [7; 4];

% sampling frequency (in Hz)
s = 20;   % 20 samples per second

% samples are collected over 24 seconds, i.e. 24*20 = 480 samples
N = 480;            % number of samples
t = (0 : N-1)/s;    % time axis

% f is the input signal
f = amp'*sin(2*pi*freq*t);
```

```
% According to the Nyquist-Shannon sampling theorem, the highest
% frequency that can be recovered from the signal is s/2.  The
% array omega below has the same number of elements as the input
% signal f.  It contains the frequency values sampled by the
% discrete Fourier transform.
omega = linspace(-s/2, s/2, length(t)+1);
omega(end) = [];

% F is the discrete Fourier transform of f
F = fftshift(fft(f));

% plot the input signal f
figure;
plot(t, f, 'r-', 'LineWidth', 1.5);
set(gca, 'FontName', 'Times', 'FontSize', 16);
xlabel('time', 'FontName', 'Times', 'FontSize', 16);
axis tight
grid on

% plot the Fourier spectrum
figure;
plot(omega, abs(F), 'b-', 'LineWidth', 1.5);
set(gca, 'FontName', 'Times', 'FontSize', 16);
xlabel('frequency', 'FontName', 'Times', 'FontSize', 16);
ylabel('Fourier spectrum', 'FontName', 'Times', 'FontSize', 16);
grid on

% plot the Fourier spectrum in logarithmic scale
figure;
semilogy(omega, abs(F), 'b-', 'LineWidth', 1.5);
set(gca, 'FontName', 'Times', 'FontSize', 16);
xlabel('frequency', 'FontName', 'Times', 'FontSize', 16);
ylabel('Fourier spectrum in log10 scale', 'FontName', 'Times', 'FontSize', 16);
grid on
```

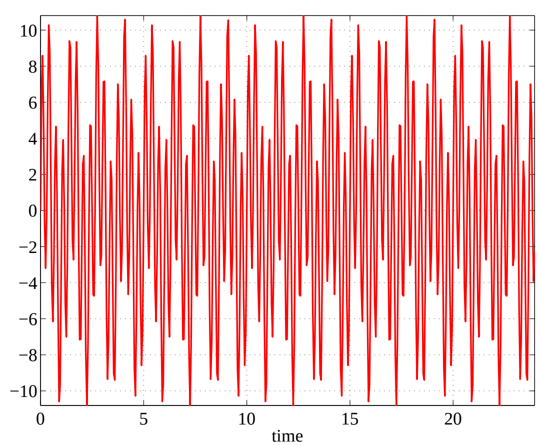

Figure 1: Example of a signal that is composed of two sine waves having frequencies 0.8 Hz and 3 Hz.

If, for a specific frequency value ω_0, we have $F(\omega_0) = a + bi$, where a and b are both scalars, then $F(-\omega_0) = a - bi$. If we zero out all the frequency components, leaving only $F(\omega_0)$ and

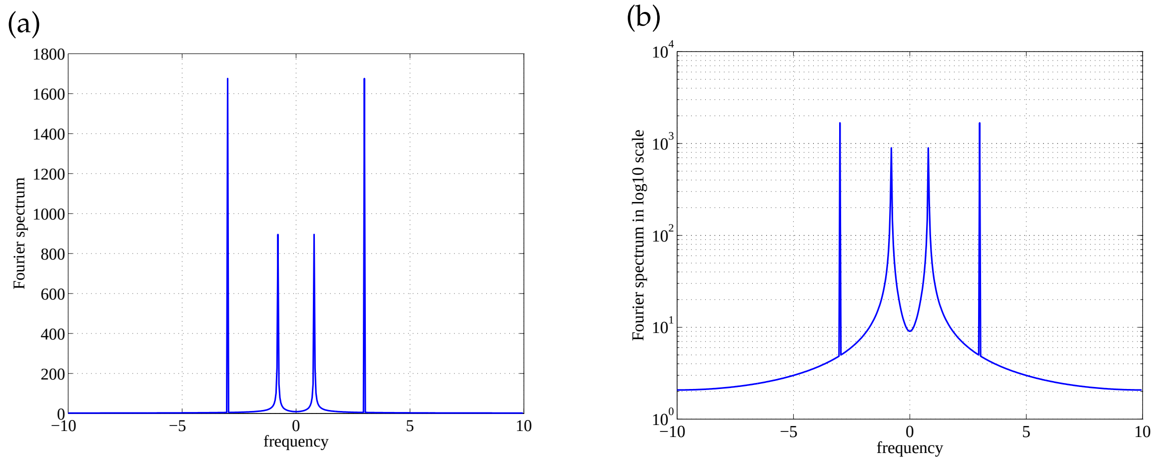

Figure 2: The Fourier spectrum of the input signal shown in Figure 1 in (a) linear scale; (b) logarithmic scale.

$F(-\omega_0)$, and apply the IDFT, then we obtain the partial reconstruction

$$
\begin{aligned}
\hat{f}_{\omega_0}(t) &= \frac{1}{T}\sum_{k=0}^{T-1} F(\omega_k)e^{i2\pi\omega_k t/T} \\
&\approx \frac{1}{T}\left(F(\omega_0)e^{i2\pi\omega_0 t/T} + F(-\omega_0)e^{i2\pi(-\omega_0)t/T}\right) \\
&= \frac{1}{T}\Big((a+bi)\left(\cos(2\pi\omega_0 t/T) + i\sin(2\pi\omega_0 t/T)\right) + \\
&\qquad (a-bi)\left(\cos(2\pi\omega_0 t/T) - i\sin(2\pi\omega_0 t/T)\right)\Big) \\
&= \frac{1}{T}\Big[\big(a\cos(2\pi\omega_0 t/T) - b\sin(2\pi\omega_0 t/T)\big) + \big(a\sin(2\pi\omega_0 t/T) + b\cos(2\pi\omega_0 t/T)\big)i + \\
&\qquad \big(a\cos(2\pi\omega_0 t/T) - b\sin(2\pi\omega_0 t/T)\big) - \big(a\sin(2\pi\omega_0 t/T) + b\cos(2\pi\omega_0 t/T)\big)i\Big] \\
&= \frac{2}{T}\big(a\cos(2\pi\omega_0 t/T) - b\sin(2\pi\omega_0 t/T)\big).
\end{aligned}
$$

Thus, using the frequency component at ω_0, the partial reconstruction, $\hat{f}_{\omega_0}(t)$, is a linear combination of the two basis cosine and sine functions corresponding to the frequency value ω_0. The complete reconstruction, $\hat{f}(t)$, is the total of the partial reconstructions from all frequency components.

If the signal f is defined in the spatial domain instead of the time domain, then the term t can be simply replaced by x. The same idea is extended to the two-dimensional case. The **2D DFT** and **2D IDFT** are given by:

$$F(\omega_n, \nu_m) = \sum_{k=0}^{N-1}\sum_{l=0}^{M-1} f(x_k, y_l)e^{-i2\pi(\omega_n x_k/N + \nu_m y_l/M)}, \tag{6}$$

for $n = 0, \cdots, N-1$; $m = 0, \cdots, M-1$;

$$f(x_k, y_l) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} F(\omega_n, \nu_m) e^{i2\pi(\omega_n x_k/N + \nu_m y_l/M)}, \tag{7}$$

for $k = 0, \cdots, N-1$; $l = 0, \cdots, M-1$.