2 Windowed Fourier Transform

It's a changing world, static is not an option.

2.1 Introduction

2D Fourier transform is a powerful tool to capture the frequency information of an image. The frequency information tells how frequent a pattern changes. This frequency of changes reflects the structural or textural features which are observed by human beings during pattern analysis. The frequency information is crucial to understand the content of an image.

However, Fourier spectrum is captured using the entire image as the window, it is a global information. In other words, we know there is a frequency in the image, but we cannot tell where the frequency is in the pattern. This is not a problem if the pattern has a homogenous structure across the pattern. For non-homogenous patterns, however, Fourier spectrum is not an effective representation, because different patterns can have similar Fourier spectrum. Figure 2.1 shows this phenomenon [1], although the two images are very different, however, their FT spectra are quite similar. This is a problem for image classification and retrieval. Therefore, we need a better tool to let us have a closer look at the patterns inside the images.

2.2 Short-Time Fourier Transform

The natural way to overcome this problem is to analyze the signal section by section or window by window. This is Short-Time Fourier Transform (STFT) which provides a way to analyze the signal in both time and frequency. In STFT, a window function is chosen in such a way that the portion of a nonstationary signal which is covered by the window function seems stationary. This window function is then convoluted with the original signal so that only the part of the signal covered by the window is selected. FT is then applied to the newly generated stationary

D. Zhang, *Fundamentals of Image Data Mining*, Texts in Computer Science, https://doi.org/10.1007/978-3-030-17989-2_2

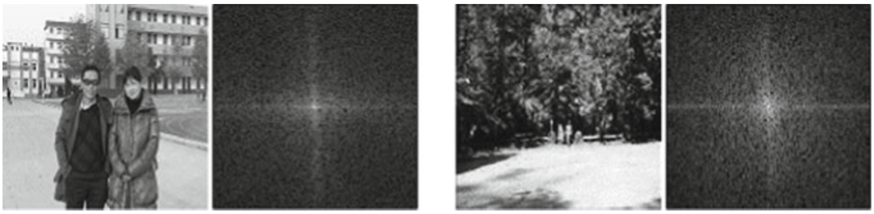


Fig. 2.1 Two images and their corresponding Fourier spectra on the right

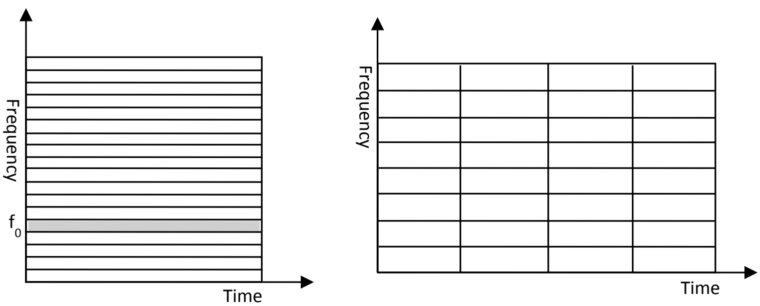
signal. The window is then moved to the next slot of signal, and FT is applied repeatedly until the whole signal is completely analyzed. For signal $f(x)$, its STFT is defined as

$$STFT(\tau,\omega) = \sum_{x=0}^{N} f(x) * W(x-\tau)e^{-j2\pi\omega x} \tag{2.1}$$

where $W(x)$ is the window, $*$ means convolution, τ represents the spatial position of the window, and ω represents the frequency captured at time τ. Similarly, the 2D STFT is given as

$$STFT(\tau_1,\tau_2,\omega) = \sum_{x=0}^{N}\sum_{y=0}^{N} f(x,y) * W(x-\tau_1, y-\tau_2)e^{-j2\pi\omega x} \tag{2.2}$$

Figure 2.2 shows the different spectrum layouts of FT and STFT on a 1D signal. The FT is applied on the entire signal which is equivalent to a single big window; it can be seen that the frequency resolution is higher than that of STFT. STFT is applied on four smaller windows, as can be expected the frequency resolution is lower than that of FT; however, the spatial resolution is higher than that of FT,



(a) FT in time-frequency plane (b) STFT in time-frequency plane

Fig. 2.2 Time–frequency illustration for FT and STFT because we can now examine the signal at four different locations. Therefore, STFT achieves a *trade-off* between frequency resolution and spatial resolution.

2.2.1 Spectrogram

When a signal $f(x)$ ($x \in [0, T]$) is analyzed by STFT, instead of a single FT spectrum, it results in a series of STFT spectra. Each of the STFT spectra is a windowed analysis of the signal $f(x)$ in a particular time slot $t \in [0, T]$. By concatenating the series of the STFT spectra vertically (in column) on the timeline, it creates a *spectrogram*. Figure 2.3 shows a spectrogram of a short sound wave. It can be observed that most of the energy is concentrated at the low frequencies; however, there are a number of particular high frequencies at different times of the sound, which are marked by the bright horizontal stripes.

Although STFT lets us do time–frequency analysis, the usually square windowing causes several side effects. First, the windowing causes the loss of low frequencies which are the most important information for signal representation. This is because low-frequency signals have longer periods/cycles, and in order to capture the low frequency, a signal must complete at least one full cycle within the window. Therefore, a window can only capture frequencies up to a certain limit.

For example, given a signal with a Nyquist sampling rate of 44,800 Hz:

- A window of 128 samples is equivalent to a period of 128/44,800 = 0.00285 s.
- Therefore, the lowest frequency the window can capture is 1/0.00285 s = 350 Hz.

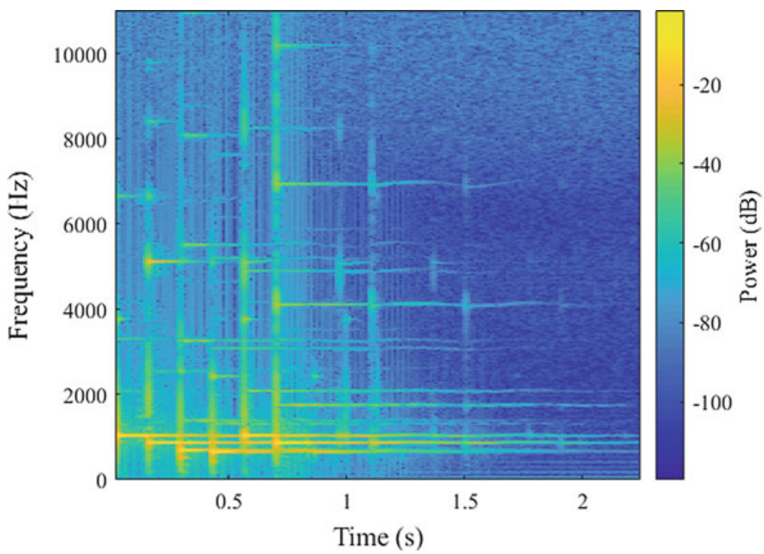


Fig. 2.3 The spectrogram of a sound wave

- In other words, the lowest frequency you can analyze with a window size of 128 samples at a sample rate of 44.8 kHz is 350 Hz.
- The second frequency which can be fit into the window is a two-cycle sine wave or a sine wave with a period of half of the window size; therefore, the second frequency a 128 window captures is 2 × 350 = 700 Hz.
- Similarly, the third frequency a 128 window captures is 3 × 350 = 1,050 Hz, so on so forth.
- In other words, the step size of the windowed frequency resolution (bin size Δu) is 350 Hz for a 128 window instead of 1 Hz for an ordinary FT. This has been shown in (1.25).
- Similarly, for a window of 64 samples, Δu = 700 Hz, while for a window of 256 samples, Δu = 175 Hz, so on so forth.
- Therefore, with STFT, we not only lose low frequencies but also lose frequency resolution, due to using only a single sized window.

Another issue with STFT is the shape of the window. The typical rectangular window causes severe frequency leakage, that is, a burst of high frequencies at both sides of the window. This is undesirable for signal or image representation which requires a compact spectrum. These issues related to STFT can be overcome to a certain extent by using non-rectangular and overlapping windows.

2.3 Gabor Filters

2.3.1 Gabor Transform

This leads to the use of Gaussian window which attenuates high frequencies at both sides of the window. The STFT with Gaussian window is called Gabor transform:

$$G(\tau_1, \tau_2, \omega) = \sum_{x=0}^{N} \sum_{y=0}^{N} f(x, y) * g(x - \tau_1, y - \tau_2) e^{-j2\pi\omega x} \tag{2.3}$$

where $g(x, y)$ is the Gaussian function:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \tag{2.4}$$

and σ_x, σ_y are the horizontal and vertical standard deviations which determine the size of the window. The window size can be varied to achieve the optimality between time and frequency.

Because convolution in spatial domain is equivalent to multiplication in frequency domain, in practice, STFT is computed by multiplying the Fourier transforms of $f(x, y)$ and $g(x, y)$.

It is found that the frequency response or Fourier transform of $g(x, y)$ is also a Gaussian $G(u, v)$, and the window size of $G(u, v)$ is inversely proportional to that of $g(x, y)$, that is,

$$\sigma_u = \frac{1}{2\pi\sigma_x} \tag{2.5}$$

$$\sigma_v = \frac{1}{2\pi\sigma_y} \tag{2.6}$$

This relationship can be used to determine the window size in spatial domain. It is known that lower frequencies are more important than higher frequencies for signal analysis and representation. Therefore, in the frequency plane, lower frequencies are given higher resolution than higher frequencies. This is achieved by giving the lower frequencies narrower bandwidth while giving the higher frequencies wider bandwidth. Typically, the bandwidths are arranged in *octave*.

2.3.2 Design of Gabor Filters

Because both Gabor function and its frequency response are Gaussians, and the relationship of the two Gaussians is given by (2.5) and (2.6), Gabor filters are designed on frequency domain. Because a 2D Gaussian function extends to infinity, there is too much overlap or redundancy between two adjacent Gaussian functions. To remove the redundancy, the 2D Gaussian functions in Gabor filters are cut at the half height, and the top half of the function is used as the Gaussian window. For a Gaussian function with standard deviation of σ, the Full Width at Half Maximum (FWHM) is $2\sqrt{2\ln 2}\sigma$ (Fig. 2.4).

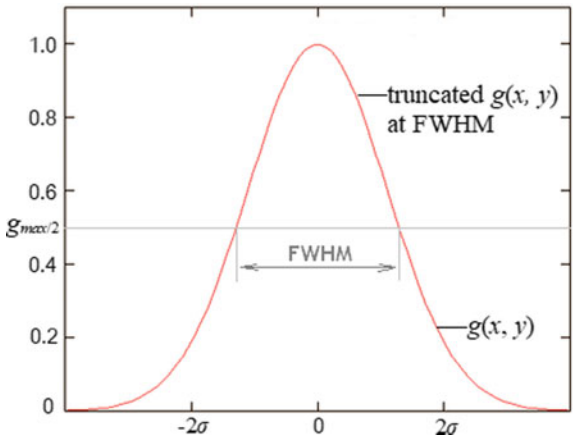


Fig. 2.4 The full width at half maximum of a Gaussian function

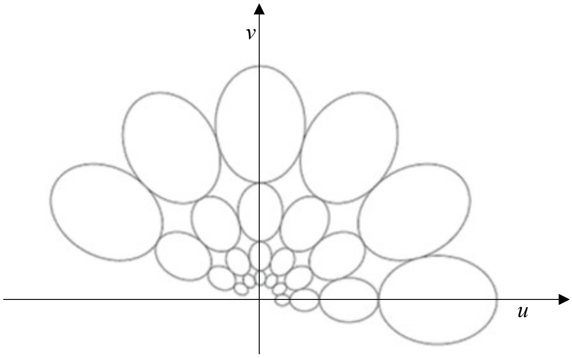


Fig. 2.5 The half-amplitude of Gabor filters in the frequency domain using four scales and six orientations

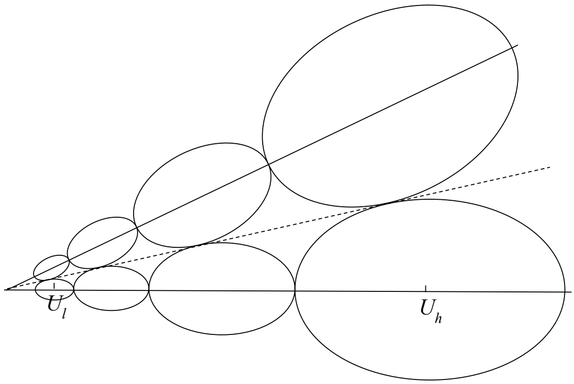


Fig. 2.6 Bandwidth tiling in frequency plane using Gaussian windows

Based on the above discussions, the half-amplitude of Gabor filters tiling of the spectrum plane is given in Fig. 2.5.

- Suppose the lowest and highest horizontal frequencies are U_l and U_h, respectively.
- The window at U_l has the smallest width (bandwidth) σ_l.
- The next window at aU_l has a width of $a\ \sigma_l$.
- The mth window at $a^m U_l$ has a width of $a^m\ \sigma_l$.
- The width of the window at $U_h = a^{M-1} U_l$ is $a^{M-1}\ \sigma_l$.
- The octave is then rotated at an interval of $\theta = \pi/k$ to tile the half frequency plane (Fig. 2.6).

With this arrangement, the parameters of the window at are obtained as follows [2]:

$$\sigma_u = \frac{a-1}{a+1} \cdot \frac{U_l}{\sqrt{2\ln 2}} \tag{2.7}$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\sqrt{\frac{U_h^2}{2\ln 2} - \sigma_u^2} \tag{2.8}$$

The Gabor transform lets us do a better time and frequency analysis than STFT, due to the use of Gaussian and overlapping windows. However, because the Gabor function is an infinite window, there is much overlap between Gabor windows. This translates to redundancy in the extracted information from the transformed coefficients. Although the FWHM truncation reduces the redundancy, it causes missing spectral information in frequency plane. Neither case is desirable for image analysis and representation. To overcome this issue, orthogonal wavelets with multiresolution are introduced in the following section.

2.3.3 Spectra of Gabor Filters

Based on the above design, each Gabor filter is determined by two parameters: scale (σ or bandwidth) and orientation; therefore, by changing the scales and orientations,

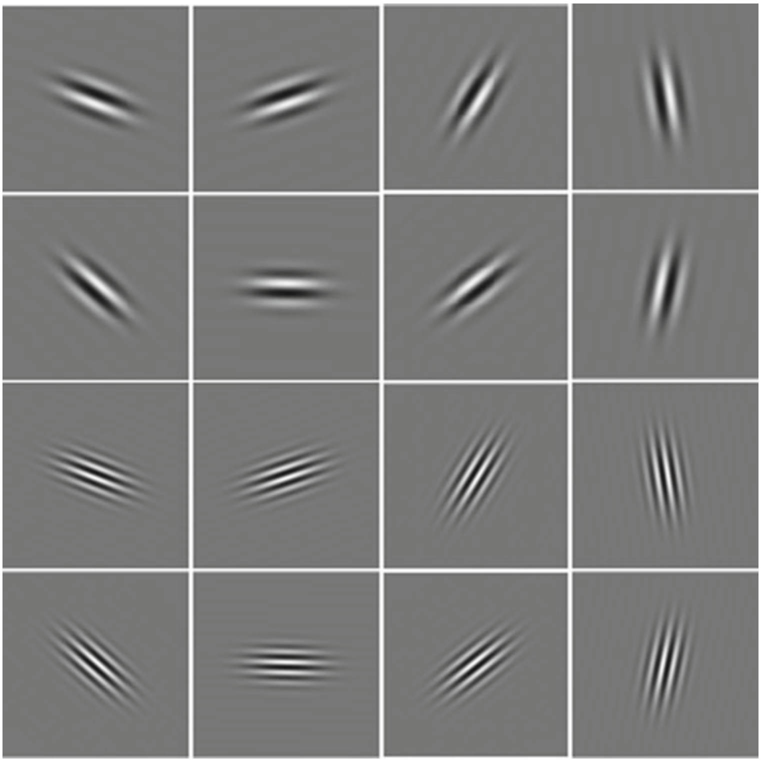


Fig. 2.7 Spectra of Gabor filters at different scales and orientations

various Gabor filters are generated. Figure 2.7 shows the real components of Gabor filters at different scales and orientations. On 2D plane, each Gabor filter is oval-shaped, and the center of the filter is given higher weight. The scale determines the granularity, with lower scale filters capturing rough features of an image and higher scale filters capturing fine features of an image. The orientation let the filters capturing image profiles and edges from different angles. The combination of both scales and orientations provides Gabor filters a powerful capability on image analysis.

2.4 Summary

In this chapter, two windowed FT methods are introduced and discussed in detail. Both STFT and Gabor filters allow for time/space–frequency analysis. Because of using shifting windows, the output of STFT on a 1D signal is a 2D spectrogram instead of a single 1D FT spectrum. It can be observed that the spectrogram reveals a lot more frequency information than a single FT spectrum. However, due to the use of windows, we sacrifice some frequency resolution. That means, instead of 1 frequency per bin in an FT spectrum, a bin in STFT represents a band of frequencies. The bandwidth of an STFT bin depends on the window size; the narrow the window, the wider the bin. We also lose some low frequencies due to windowing. Therefore, it is important to learn the trade-off between window size and bin size when using STFT.

Compared with STFT, Gabor filters provide a better solution in terms of the trade-off, because Gabor filters use multiple filter size. Furthermore, the use of Gaussian window by Gabor filters produces more desirable results than the rectangular window.

2.5 Exercises

1. Match each of the following signals to its corresponding spectrogram underneath the signals and explain why you match it that way.