CMSC 491/691: Computer Vision **Spring 2024**

Lecture 3: 2024-02-07 Image Filtering II

Lecturer: Tejas Gokhale *Scribe: Bryan Yang*

3.1 Recap of Last Lecture

In our previous lecture, we explored how an image can be represented by a matrix of pixels, each with a value that captures its intensity within a range. We learned that each pixel contains a combination of these intensity values across three color channels: Red (R), Green (G), and Blue (B), which collectively define the full-color spectrum visible in the image. We also looked into two main types of image enhancement techniques:

- **Point Processing** operations such as identity/inversion and darkening/lightening adjust individual pixel values to alter the image's overall appearance without changing its content.
- **Image Filtering** methods, including that of convolution and correlation, utilize kernels to modify pixel values based on their surrounding pixels, which oftentimes result in an overall change in structure for the image. These changes can include the enhancement of features like edges and textures or applying blurs such as Gaussian or box filters.

Both of these methods enable the transformation of images for tasks ranging from basic editing to complex feature extraction in computer vision.

3.2 Fourier Series

A sine wave and a square wave differ significantly in their shapes. A sine wave is a single, pure frequency with no harmonics, characterized by its smooth, repeating oscillations. In contrast, a square wave has a much more abrupt transition between its high and low states.

3.2.1 Fourier's Theorem

With enough sine waves added together, however, one can construct an approximation to a square wave by carefully adjusting their frequencies, amplitudes, and phases. As more sine waves are added, particularly those at the correct odd harmonic frequencies, the approximation becomes increasingly accurate, with the summation converging towards the shape of the original square wave.

Fourier's theorem, states the following idea:

> Any periodic signal, no matter how complex, can be represented as the sum of a series of simple sinusoidal waves (sine and cosine functions) at various frequencies, amplitudes, and phases.

3.2.2 Mathematical Formulation

This process is known as the Fourier Series. In essence, the Fourier Series reveals that any periodic signal, regardless of its complexity, can be represented as the sum of sine and cosine functions, each multiplied by a coefficient that corresponds to the signal's amplitude at a specific frequency.

Mathematically, each term in the Fourier Series is a sine wave of the form:

$$A\sin(\omega x + \phi)$$

where:

- A is the amplitude of the wave,
- ω is the angular frequency (related to the standard frequency by $\omega = 2\pi f$),
- x is the variable (often time),
- ϕ is the phase shift of the wave.

3.3 Fourier Transform

The Fourier Transform is like a tool in mathematics that transforms a signal from the time or space domain into the frequency domain. In image processing, the Fourier Transform is essential for analyzing the frequency content of images, which allows us to see which frequencies are present in an image as well as how strong they are. The utility of this applies to tasks like image filtering, compression, and enhancement.

3.3.1 Discrete Fourier Transform (DFT)

When transforming an image into the frequency domain, we can apply various filters to selectively amplify or suppress certain frequencies, thus achieving effects such as blurring, sharpening, and noise reduction. Subsequently, the inverse Fourier Transform can be used to convert the modified frequency domain representation back into the spatial domain, resulting in the processed image.

For a 1D signal, the DFT is defined as:

$$F(u) = \sum_{n=0}^{N-1} f[n] \cdot \exp\left(-2\pi j \frac{un}{N}\right) \tag{3.1}$$

Likewise, for a 2D signal (an image), the 2D DFT is defined as:

$$F(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n] \cdot \exp\left(-2\pi j \left(\frac{um}{M} + \frac{vn}{N}\right)\right) \tag{3.2}$$

3.3.2 Magnitude and Phase

The 2D DFT can be further decomposed into magnitude and phase components, which encode different types of information about the image:

- **Magnitude**: Encodes the intensity information of the signal, representing how much of each frequency is present.
- **Phase**: Encodes the positional information, determining where in time or space the frequency components are located.

3.3.3 Inverse Fourier Transform

The inverse Fourier Transform is used to convert the frequency domain signal back into the spatial domain, allowing for a reconstruction of the original signal from its frequency components.

$$f[n] = \frac{1}{N} \sum_{u=0}^{N-1} F(u) \cdot \exp\left(2\pi j \frac{un}{N}\right) \tag{3.3}$$

3.3.4 Convolution Theorem

An important property of the Fourier Transform is the Convolution Theorem, which states that a convolution in the spatial domain is equivalent to the multiplication in the frequency domain.

$$\mathcal{F}\{h * f\} = \mathcal{F}\{h\} \cdot \mathcal{F}\{f\} \tag{3.4}$$

When considering the Inverse Fourier Transform, the theorem implies that multiplication in the spatial domain corresponds to convolution in the frequency domain:

$$\mathcal{F}^{-1}\{H \cdot F\} = h * f \tag{3.5}$$

where $H = \mathcal{F}\{h\}$ and $F = \mathcal{F}\{f\}$ are the Fourier Transforms of h and f respectively.

This theorem is particularly useful because it allows us to perform convolution operations by simple multiplication in the frequency domain, which can be computationally more efficient.

3.3.5 Filtering in the Fourier Domain

Filtering an image in the Fourier domain can be done by applying a mask that controls the frequencies that are allowed to pass through. More specifically, this mask has a threshold that determines which frequencies are allowed to pass and which are attenuated or blocked.

Two of the most common filtering methods are low-pass and high-pass filtering:

- **Low Pass Filtering**: A Low Pass Filter (LPF) works by allowing the low frequencies to pass through while blocking/attenuating the high frequencies. In the Fourier domain, this filter is typically visualized as a mask that is centered around the origin, which corresponds to the low-frequency components. By permitting only these lower frequencies to remain, the LPF blurs the image, which reduces noise and smooths out variations.
- **High Pass Filtering**: In contrast to the LPF, a High Pass Filter (HPF) allows the high frequencies to pass while blocking/attenuating the lower frequencies. While a low pass filter in the Fourier domain

is represented as a mask centered around the origin, which lets through low-frequency components (remember the area around the center represents the lowest frequencies), a high pass filter can be visualized as the opposite. The high frequencies in an image are often associated with the finer details and edges. Thus, applying an HPF enhances these details, making edges more pronounced and sharpening the image.

With the Convolution Theorem, we can apply these filters to the desired image in the Fourier domain, in which the convolution becomes a point-wise multiplication, allowing us to apply the desired filter by multiplying the image's Fourier transform with the filter's frequency response. The theorem simplifies the convolution process, which can be computationally expensive in the spatial domain, especially in the case of large kernels.

The Convolution Theorem, thus, provides an efficient mechanism to perform filtering operations that can change the appearance of an image, by manipulating the frequency components directly.

3.4 Gaussian/Laplacian Pyramids

3.4.1 Gaussian Filters

In image processing, Gaussian filters are particularly used for blurring and smoothing applications. They depend on a kernel that approximates the shape of a 2D Gaussian bell curve to weigh the central pixels more heavily than the edges. Recall that this filter approximates the function:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.6}$$

where σ represents the standard deviation of the distribution, and x and y are pixel positions relative to the center of the kernel.

The effect of the Gaussian filter is to blur images by reducing detail and noise. We will soon see how the Gaussian filter is often used as a preprocessing step in image processing as well as in other image analysis algorithms.

3.4.2 Laplacian Filters

The Laplacian filter is an edge-detection operator that uses a second-order derivative approach that highlights regions of rapid intensity change in an image and hence, is excellent for edge detection. It can be represented as a convolution kernel just like the Gaussian filter counterpart and is often applied to an image that has already been smoothed with a Gaussian filter to reduce sensitivity to noise. The Laplacian filter is also considered isotropic, which means it detects edges equally in all directions.

3.4.3 Gaussian Pyramids

By repeatedly applying a Gaussian filter and then downsampling the image (taking every other pixel to reduce the image's dimensions), one can construct what is called a Gaussian pyramid, which is a stack of images at progressively lower resolutions that represent the same scene as if it were viewed from increasing distances.

The operation here defines the process for generating the k-th level of the Gaussian pyramid:

$$G_k = ds(Gaussian(G_{k-1})) \tag{3.7}$$

where G_k is the k-th level of the pyramid, and ds is the function that reduces image resolution by a factor of 2. Each level G_k is obtained by applying a Gaussian blur to the previous level G_{k-1} and then downsampling the result.

3.4.4 Laplacian Pyramids

Upsampling is like the reverse process of downsampling, where an image is enlarged by inserting new pixels between existing ones. The value of these new pixels can be calculated by certain interpolation methods that often use the same Gaussian kernel in a reversed manner.

The Laplacian pyramid essentially represents the multi-scale decomposition of an image, in which edge information is captured at different scales. It is constructed by subtracting each level of the Gaussian pyramid from the next higher resolution level, then upsampling/interpolating to match the original image size.

The operation here defines the process for generating the k-th level of the Laplacian pyramid:

$$L_k = G_k - us(G_{k+1}) \tag{3.8}$$

where L_k is the k-th level of the Laplacian pyramid, and us is the function that increases image resolution. Each level L_k is created by subtracting the upscaled version of the next Gaussian pyramid level G_{k+1} from the current level G_k.

3.4.5 Applications of Pyramids

Both Gaussian and Laplacian pyramids have many applications in image-processing tasks such as image blending, compression, noise removal, or even feature extraction. In image blending, for instance, one could use a Laplacian pyramid to isolate details from two images at multiple scales and then combine these details seamlessly. Hence, their use in each of these tasks demonstrates the high practical applicability of Fourier Transform techniques.