

Smoothing and Convolution

Why smooth? Images have noise. The intensity measured at a pixel is the "true" intensity plus noise. If nearby pixels have similar "true" intensities, then we can use smoothing to reduce the noise. We can also think of smoothing as a simple example of how information can be passed between neighboring pixels.

Example: Suppose the true intensities in a region = 100 100 100 100 100 100. Measured intensities = $100 + n_i$, where n_i is some noise. If we average M pixels we get: $100 + \sum \frac{n_i}{M}$. If n_i are zero mean i.i.d. then $\sum \frac{n_i}{M}$ is a random variable with the same mean as n_i (i.e., 0) and much lower variance, which drops how?

1 Convolution

If we average pixels in a neighborhood, for the continuous, 1-D case we get:

$$h(t) = \frac{1}{2\delta} \int_{t-\delta}^{t+\delta} f(t') dt'.$$

Equivalently, we can define a function g such that $g(t) = \frac{\delta}{2}$ for $-\delta \leq t \leq \delta$, and 0 otherwise. Then:

$$h(t) = \int_{-\infty}^{\infty} f(t-u)g(u)du$$

This is convolution, and can be done with an arbitrary function g . We write $h = g * f$.

Try to visualize this as taking g and sliding it across the function f . For each position of g , the value of $h(t)$ is computed at the center of g 's position, by multiplying g and f together, and integrating.

As another example, we might want to take a weighted average of t 's neighborhood. For example, if we weight according to a Gaussian, we just have:

$$g = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}.$$

Then we convolve with g .

$$h(t) = \int_{-\infty}^{\infty} f(t-u)g(u)du$$

Convolution is linear and shift invariant. Linear, because each element of h is a linear combination of elements of f (we multiply and integrate). So we have: $g * kf = k(g * f)$ and $g * (f1 + f2) = g * f1 + g * f2$. Shift invariant because g is applied to every point in f in exactly the same way.

Because of this linearity, convolution has some important properties, such as commutivity and associativity.

2 Convolution Theorem

Now we get to the reason why the Fourier transform is important in understanding convolution. This is because of the convolution theorem. Let F, G, H be the Fourier transform of f, g , and h . The convolution theorem states that convolution in the spatial domain is equivalent to multiplication in the transform domain, ie, $f * g = h \iff FG = H$.

Proof: Suppose $h(t) = g * f(t)$. $H(w) = \int e^{-itw} (\int f(t-u)g(u)du)dt$ We change variables, so that $v = t - u$.

$$\begin{aligned} H(w) &= \int \int e^{-i(u+v)w} f(v)g(u)dudv \\ &= \left(\int e^{-ivw} f(v)dv \right) \left(\int e^{-iuw} g(u)du \right) = G(w)F(w) \end{aligned}$$

This is a remarkable theorem. It means that if we convolve a sine wave with any kernel, we produce a sine wave of the same frequency. One way to say this is that sines and cosines of varying frequencies are the eigenvectors of convolution. It means that convolution attenuates every frequency by different amounts, and we can understand the effects of convolution by taking the Fourier transform of the convolution kernel.

Simplest example: The Fourier transform of a delta function is a constant function. We can prove this by just noting that for $t = 0$ $\cos(kt) = 1$ for any k , and $\sin(kt) = 0$ for any k . This means that convolving with a delta function affects all frequencies the same (since it doesn't change them). (Why doesn't this violate Parseval's theorem?)

As another simple example, we note that if we convolve a function with a single, pure harmonic (sine or cosine of a specific frequency) we get out the same harmonic, with a possibly different amplitude.

As another example, we point out that the Fourier transform of a Gaussian is a Gaussian. If $g(t)$ is gaussian, the broader $g(t)$ is the narrower $G(w)$ is (and vice versa). This means that smoothing kills high frequency components of a signal. The more we smooth, the more high frequency components we kill. In fact, attenuating high frequency components of a signal can be taken to be the definition of smoothing. To some extent, this is why we learned the Fourier transform. So we can understand what smoothing really does.

We can also see why simple averaging isn't as good a way of smoothing. Let $f(t)$ be constant in the range $-T$ to T and 0 elsewhere. This is sometimes called a *box* function. Then its Fourier transform is:

$$F(w) = \int_{-T}^T e^{iwt} dt = 2 \sin(Tw)/w$$

the sinc function. This does not decay as fast as a Gaussian. Also it has oscillations that produce odd effects.

The perfect band-pass filter looks like this function in frequency space. Because the inverse Fourier transform is taken in the same way as the Fourier transform, in the inverse transform of a perfect band-pass is a sinc. It does not decay quickly in space.

Another interesting example is found by taking the derivative of a function. Suppose we have:

$$f(t) = a_0 + \sum a_k \cos(kt) + \sum b_k \sin(kt).$$

Then:

$$f'(t) = \sum -ka_k \sin(kt) + \sum kb_k \cos(kt)$$

and

$$F'(w) = iwF(w).$$

The Uncertainty Principle: The Heisenberg uncertainty principle essentially says that no function can be arbitrarily well-localized in both space and time. This is a proof that the situation with Gaussians is in some sense canonical.

3 Aliasing and the Sampling Theorem

We'll just consider these issues for the Fourier series, but qualitatively similar results hold for the Fourier transform.

Sampling means that we know the value of f at discrete intervals, $f(\frac{n\pi}{T})$ for $n = 0, +1, -1, +2, -2, \dots$. That is, we have $2T + 1$ uniform samples of $f(t)$, assuming f is a periodic function. Suppose now that f is band-limited to T . That means,

$$f(t) = a_0 + \sum_{k=1}^T a_k \cos(kt) + \sum_{k=1}^T b_k \sin(kt).$$

We know the value of $f(t)$ at $2T + 1$ positions, giving us $2T + 1$ equations. The a_i and b_i give us $2T + 1$ unknowns. These are linear equations. So we get a unique solution. This means that we can reconstruct f from its samples. However, if we have fewer samples, reconstruction is not possible. If there is a higher frequency component, then this can play havoc with our reconstruction of all low frequency components, and be interpreted as odd low frequency components.

This means that if we want to sample a signal, the best thing we can do is low-pass filter it and then sample it. Then, we get a signal that matches our original signal perfectly in the low-frequency components, and this is the best we can do. To shrink images, in fact, we low-pass filter them and then sample them.