Discrete vs. Continuous Convolution and Fourier Transforms

Brian Curless
CSE 557
Fall 2009

Discrete convolution, revisited

One way to write out discrete signals is in terms of sampling:

$$f(x)\mathrm{III}(x;T) = \sum_{n=-\infty}^{\infty} f(x)\delta(x-nT) = \sum_{n=-\infty}^{\infty} f(nT)\delta(x-nT)$$

Rather than refer to this complicated notation, we will just say that a sampled version of $f(x)$ is represented by a "digital signal" $f[n]$, the collection of samples of $f(nT)$ sifted out by the shah function.

For a digital signal, we define **discrete convolution** as:

$$\begin{aligned} g[n] &= f[n] * h[n] \\ &= \sum_{n'} f[n']h[n-n'] \\ &= \sum_{n'} f[n']\tilde{h}[n'-n] \end{aligned}$$

where $\tilde{h}[n] = h[-n]$.

Discrete convolution, cont'd

What connection does discrete convolution have to continuous convolution?

We're essentially computing

$$f[n] * h[n] \text{ "=" } \left[f(x)\mathrm{III}(x)\right] * \left[h(x)\mathrm{III}(x)\right]$$

for some pair of functions $f(x)$ and $h(x)$ that pass through the samples $f[n]$ and $g[n]$.

It would be nice if this were the same as:

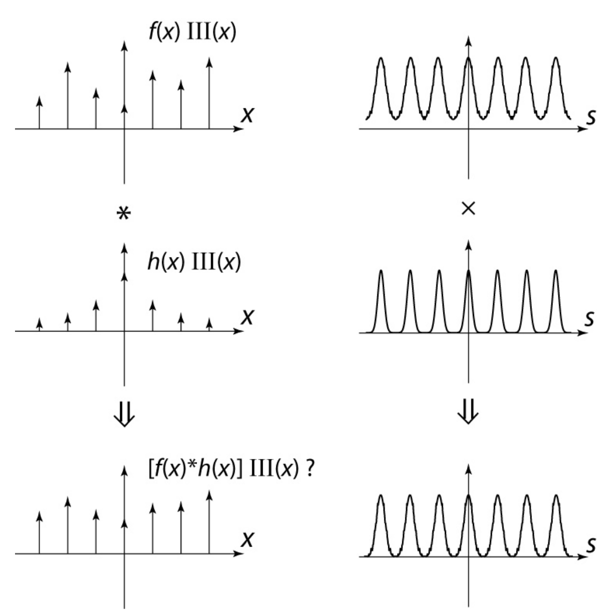
$$\left[f(x) * h(x)\right]\mathrm{III}(x)$$

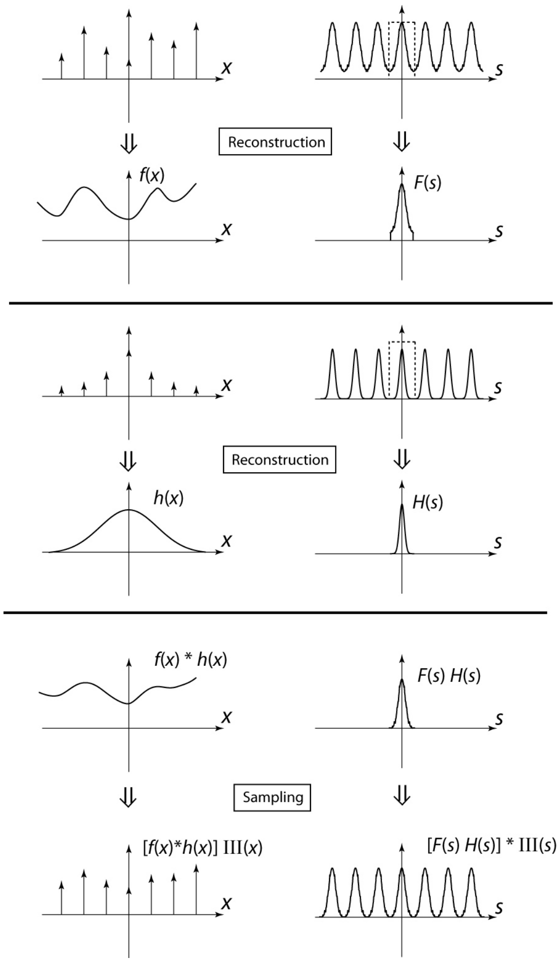
i.e., if we could think in terms of convolving continuous functions and then resampling.

But, is it the same?

Discrete convolution, cont'd

We can analyze this convolution in the Fourier domain:





Discrete Fourier Transform

Recall that the continuous 1D Fourier transform (FT) is:

$$F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi sx} dx$$

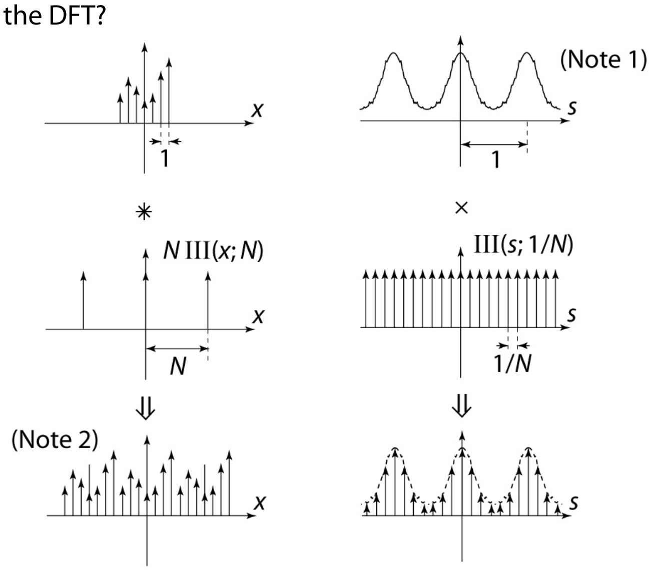
The discrete version of this is the **Discrete Fourier Transform (DFT)**:

$$F[k] = \sum_{n=0}^{N-1} f[n]\, e^{-i\frac{2\pi}{N}kn}$$

where it is assumed that the sampled signal is of finite length *N*.

Discrete Fourier Transform, cont'd

Is there a connection between the continuous FT and the DFT?



Note 1: horizontal axes not drawn to scale.
Note 2: amplitude scaled by N.

Yes! The DFT is essentially the FT of the input samples, after repeating them along the *x* axis.

Discrete Fourier Transform, cont'd

Summarizing, the continuous FT and inverse FT were:

Spatial domain → $F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi sx} dx$ → Frequency domain

Spatial domain ← $f(x) = \int_{-\infty}^{\infty} F(s) e^{i2\pi sx} ds$ ← Frequency domain

and we now have the DFT and inverse DFT:

Spatial domain → $F[k] = \sum_{n=0}^{N-1} f[n]\, e^{-i\frac{2\pi}{N}kn}$ → Frequency domain

Spatial domain ← $f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k]\, e^{i\frac{2\pi}{N}kn}$ ← Frequency domain

Notes:

- Properties of FT's generally apply to DFT's (e.g., convolution theorem).
- Brute force DFT computation is $O(n^2)$.
- The Fast Fourier Transform (FFT) algorithm computes the DFT in $O(n \log n)$.

Discrete convolution in 2D

Similarly, discrete convolution in 2D becomes:

$$g[n,m] = f[n,m] * h[n,m]$$
$$= \sum_{m'} \sum_{n'} f[n',m'] h[n-n', m-m']$$
$$= \sum_{m'} \sum_{n'} f[n',m'] \tilde{h}[n'-n, m'-m]$$

where $\tilde{h}[n,m] = h[-n,-m]$.

Further, the 2D DFT and inverse DFT are, for an N x M image:

$$F[k,l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m]\, e^{-i2\pi\left(\frac{kn}{N}+\frac{lm}{M}\right)}$$

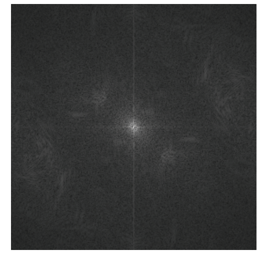
$$f[n,m] = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F[k,l]\, e^{i2\pi\left(\frac{kn}{N}+\frac{lm}{M}\right)}$$

As in 1D, the image and its DFT implicitly repeat, in this case tiling the 2D plane.

Spectral impact of sharpening

We can look at the impact of sharpening on the Fourier spectrum using DFTs:

Spatial domain | Frequency domain



$$\delta - \Delta^2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$