Notes on Filter Design

Psych 267/CS 348D/EE 365

Prof. David J. Heeger

September 15, 1998

There are a lot of ways to design discrete, linear filters (e.g., see Ch. 7 of Oppenheim and Schafer). Here, we derive a weighted least-squares design method, a very simple method that works pretty well most of the time.

We want to design a real-valued filter $h[n]$ with a finite (hopefully very small) number of taps such that it has a desired frequency response, $H[k]$. For example, a 5-tap filter has frequency response:

$$H[k] = \sum_{n=-2}^{2} h[n] \exp\left[-j\frac{2\pi\, k\, n}{M}\right], \tag{1}$$

for $-2 \leq n \leq 2$ and $0 \leq k \leq M-1$. Here $H[k]$ is the frequency response of $h[n]$, i.e., it is the DFT of the impulse response for an impulse sequence of length M.

Let's say that $\tilde{H}[k]$ is the desired frequency response. Our goal is to choose the filter taps, $h[n]$, to minimize:

$$\sum_{k=0}^{M-1} (H[k] - \tilde{H}[k])^2.$$

Even-Symmetric Filters. First, let's work through an example for a 5-tap, even symmetric filter. That is:

$$\begin{aligned} h_0 &= h[0] \\ h_1 &= h[1] = h[-1] \\ h_2 &= h[2] = h[-2], \end{aligned}$$

where $\mathbf{h} = (h_0, h_1, h_2)^t$ are the three distinct filter taps. There are only three distinct taps because we are enforcing even symmetry. The frequency response of this filter is obtained by writing out all the terms in Eq 1:

$$\begin{aligned} H[k] &= h[-1] \exp\left[j\frac{2\pi\, k}{M}\right] + h[1] \exp\left[-j\frac{2\pi\, k}{M}\right] \\ &\quad + h[-2] \exp\left[j\frac{4\pi\, k}{M}\right] + h[2] \exp\left[-j\frac{4\pi\, k}{M}\right] \\ &\quad + h[0]. \end{aligned}$$

Using the fact that $2\cos(x) = \exp(jx) + \exp(-jx)$,

$$H[k] = h_0 + 2h_1 \cos[2\pi(k/M)] + 2h_2 \cos[4\pi(k/M)].$$

Hence, we want to choose $\mathbf{h}$ to minimize:

$$||\mathbf{Ch} - \tilde{\mathbf{H}}||^2, \tag{2}$$

where the vector $\tilde{\mathbf{H}}$ is the desired frequency response, $\tilde{\mathbf{H}} = \tilde{H}[k]$, and where $\mathbf{C}$ is an Mx3 matrix. The columns of $\mathbf{C}$ are cosine basis vectors. The zeroth column is: $C_0[k] = 1$, the first column is: $C_1[k] = 2\cos[2\pi(k/M)]$, and the second column is: $C_2[k] = 2\cos[4\pi(k/M)]$.

The least-squares (regression) solution is given by the usual formula:

$$\hat{\mathbf{h}} = (\mathbf{C}^t\mathbf{C})^{-1}\mathbf{C}^t\tilde{\mathbf{H}}, \tag{3}$$

where $\mathbf{C}^t$ means the transpose of $\mathbf{C}$.

Odd-Symmetric Filters. We can use the same approach to design an odd symmetric filter. For a 5-tap odd symmetric filter the vector $\mathbf{h}$ is given by:

$$\begin{aligned} h_0 &= 0 \\ h_1 &= -h[1] = h[-1] \\ h_2 &= -h[2] = h[-2]. \end{aligned}$$

The derivation is essentially the same except that you end up with sinusoids instead of cosinusoids in the columns of $\mathbf{C}$ because the frequency response of the filter is now given by:

$$H[k] = 2jh_1\sin[2\pi(k/M)] + 2jh_2\sin[4\pi(k/M)].$$

Weighted Least-Squares. Often, we care more about some frequency components than others. For example, we might want to enforce that the filter have zero dc response. Or we might want to enforce that the frequency response be very small (or zero) for some other set of frequency components. In these cases, it is helpful to use a weighted least squares method. Use large weights for frequency components that you care a lot about and use small (or zero) weights for the other frequency components. Using weighted least squares, we want to choose $\mathbf{h}$ to minimize:

$$\sum_{k=0}^{M-1}(w[k])^2(H[k] - \tilde{H}[k])^2,$$

where $w[k]$ are the weights. This can be written in matrix notation as follows:

$$||\mathbf{Ah} - \mathbf{b}||^2,$$

where $\mathbf{b} = w[k]\tilde{H}[k]$ is a weighted version of the desired frequency response. The columns of $\mathbf{A}$ are weighted versions of the (co)sine basis vectors (columns of $\mathbf{C}$). In particular, the ith column of $\mathbf{A}$ is given by: $A_i[k] = w[k]C_i[k]$. The solution (as above) is given by:

$$\hat{\mathbf{h}} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{b}.$$

Derivative Filters Many image processing algorithms depend on computing derivatives of a digital image: edge detectors (Laplacian zero crossings, gradient magnitude), steerable filters, motion estimation, depth from stereo, anisotropic diffusion. But derivatives are only defined for continuous functions of continuous variables, not for discretely-sampled and quantized signals. Often, people use simple differences between an adjacent pair of pixels to approximate the derivative. But one can do much better by designing a set of matched pairs of derivative filters and lowpass prefilters.

We conceive of the derivative operation (on a discrete signal) as performing three steps:

1. Reconstruct (interpolate) a continuous function from the discrete signal: $p(x) * a[n]$. Here $a[n]$ is a discrete signal, $p(x)$ is an interpolation filter (e.g., a sinc or some other low pass filter), and $*$ means convolution.
2. Take the derivative of the interpolated continuous signal: $\frac{\partial}{\partial x}(p(x) * a[n])$.
3. Sample the continuous derivative: $\mathcal{S}[\frac{\partial}{\partial x}(p(x) * a[n])]$, where $\mathcal{S}$ is the sampling operation.

Altogether, these three steps are the same as convolving with a discrete filter:

$$\mathcal{S}[\tfrac{\partial}{\partial x}(p(x) * a[n])] = \{\mathcal{S}[\tfrac{\partial}{\partial x}p(x)]\} * a[n],$$

where $d[n] = \mathcal{S}[\frac{\partial}{\partial x}p(x)]$ is a discrete filter kernel (the sampled derivative of a lowpass prefilter).

One could use an ideal lowpass (sinc) function for the prefilter, or a gentler function such as a Gaussian. But for many practical applications, we would like a relatively small filter kernel so we cannot use an ideal lowpass filter (which would have an infinite size kernel). On the other hand, the important thing for many applications is that we end up with a pair of signals, one which is the derivative of the other. A non-ideal interpolator will introduce some distortions, making it inappropriate to compare the original signal with its "derivative." This suggests that we should compute two convolution results: (1) the prefiltered original computed by convolving with the discrete prefilter $p[n]$, and (2) the derivative of the prefiltered original computed by convolving with the discrete derivative filter $d[n]$.

Now we wish to design a discrete prefilter $p[n] = \mathcal{S}[p(x)]$ and a discrete derivative filter $d[n] = \mathcal{S}[\frac{\partial}{\partial x}d(x)]$ so that the latter is the derivative of the former. In the frequency domain, we want:

$$D[k] \approx -j\,2\pi\,(k/M)\,P[k],$$

where $P[k]$ is the frequency response of $p[n]$, $D[k]$ is the frequency response of $d[n]$, and $0 \leq k \leq (M-1)$. This equation is based on the derivative property of the Fourier transform: the Fourier transform of the derivative of a function equals a complex ramp $-j\omega$ times Fourier transform of the original function. For an intuition for the derivative property, recall that the derivative of $\frac{\partial}{\partial x}\cos(\omega x) = -\omega\sin(\omega x)$.

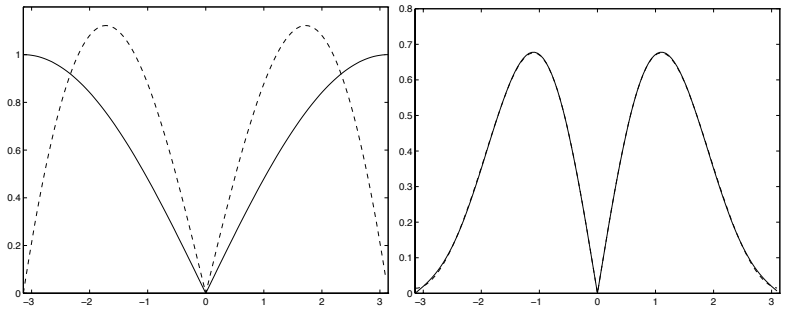

Figure 1: Frequency responses of 2-tap, finite difference (left) and 5-tap (right) derivative/prefilter pairs. Shown are the magnitude of the Fourier transforms of: a) the derivative kernel (dashed line), and b) the frequency-domain derivative of the prefilter (that is, its Fourier magnitude multiplied by $\omega = 2\pi(k/M)$.

Using weighted least-squares, we want to minimize:

$$\sum_{k=0}^{M-1} (w[k])^2 (D[k] + j2\pi(k/M)P[k])^2.$$

This can be rewritten as:

$$||\mathbf{Ad} + \mathbf{A'p}||^2,$$

where $\mathbf{p}$ is a vector containing the prefilter kernel, $\mathbf{d}$ is a vector containing the derivative kernel, $\mathbf{A}$ contains weighted versions of the Fourier basis functions as above, and $\mathbf{A'}$ is a similar matrix containing the Fourier basis functions multiplied by $\omega = 2\pi(k/M)$. After consolidating terms, we want to minimize:

$$||\mathbf{Mu}||^2,$$

where

$$\mathbf{M} = (\mathbf{A'} \,|\, \mathbf{A}) \quad \text{and} \quad \mathbf{u} = \begin{pmatrix} \mathbf{p} \\ \mathbf{d} \end{pmatrix}.$$

The solution $\hat{\mathbf{u}}$ is given by the eigenvector corresponding to the smallest eigenvalue of $\mathbf{M}^t\mathbf{M}$. Then the filters are both renormalized (by the same scale factor) so that $p[n]$ has unit dc response (i.e., the samples of $p[n]$ sum to one).

An example of a pair of 5-tap filters are:

$$p[n] = [.035698 \; .24887 \; .43086 \; .24887 \; .035698]$$

$$d[n] = [.10766 \; .282670 \; -.28267 \; -.10766]$$

The frequency responses of these two filters are compared in the figure.

References

[1] E P Simoncelli. Design of multi-dimensional derivative filters. In *International Conference on Image Processing*, volume I, pages 790–793, 1994.