Published in: **First IEEE Int'l Conf on Image Processing,**
Austin Texas, vol I, pages 790--793, November 1994.

DESIGN OF MULTI-DIMENSIONAL DERIVATIVE FILTERS

Eero P. Simoncelli

GRASP Laboratory, Room 335C
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104-6228

ABSTRACT

Many multi-dimensional signal processing problems require the computation of signal gradients or directional derivatives. Traditional derivative estimates based on adjacent or central differences are often inappropriate for multi-dimensional problems. As replacements for these traditional operators, we design a set of matched pairs of derivative filters and lowpass prefilters. We demonstrate the superiority of these filters over simple difference operators.

1. INTRODUCTION

A wide variety of problems in multi-dimensional signal processing require the computation of gradients or directional derivatives. Typically, this requirement arises from a need to compute local Taylor series expansions to characterize signals. For example, in image processing, gradient measurements are used as a first stage of many edge detection, depth-from-stereo, and optical flow algorithms.

Multi-dimensional derivatives are usually computed as in one-dimensional signal processing, via differences between neighboring sample values (typically "backward" or "central" differences). These derivative estimates are often highly inaccurate [5]. This is especially true in applications that require computation of directional derivatives via linear combination of the axis derivatives (i.e., the components of the gradient).

As a replacement for traditional sample differences, we propose the use of matched pairs of derivative filters and lowpass prefilters. We formulate this as a filter design problem, and derive a set of small-kernel linear-phase filters. We demonstrate that derivatives computed via these filters are substantially more accurate than those computed via standard approaches.

2. COMPUTING DERIVATIVES OF SAMPLED SIGNALS

Assume we wish to compute the derivative of a sampled signal at the location of each sample. The derivative operator is a linear shift-invariant operator, and we may therefore view its application to a signal as a convolution operation. Since derivatives are defined on continuous functions, the computation on a discrete function requires (at least implicitly) an intermediate interpolation step with a continuous function $C(x)$. The derivative of this interpolated function must then be re-sampled at the points of the original sampling lattice.

The sequence of operations may be written for a one-dimensional signal as follows:

$$\begin{aligned}\frac{d}{dx}f(n) &\equiv \left[\frac{d}{dx}\left(\sum_m f(m)C(x-m)\right)\right]_{x=n} \\ &= \left[\sum_m f(m)\frac{dC}{dx}(n-m)\right],\end{aligned}$$

where we assume unit sample spacing in the discrete variables n and m for simplicity. Thus, the derivative operation is equivalent to convolution with a filter which is the sampled derivative of some continuous interpolation function $C(\vec{r})$. One could use an ideal lowpass (sinc) function, or a gentler function such as a Gaussian. An optimal choice would depend on a model for the original continuous signal (e.g., bandlimited at the Nyquist limit, with $1/\omega$ power spectrum) and a model of the noise in the sampled signal. We will refer to this interpolation filter as the *prefilter*.

For many practical applications, we would like a relatively small linear-phase derivative kernel. We therefore cannot use an ideal lowpass prefilter (which would lead to an infinite size kernel). But a non-ideal interpolator will introduce distortions in the original signal. More specifically, we will be computing the derivatives of an improperly interpolated signal. In situations where we need to make comparisons between or take combinations of a signal and its derivative[1], this suggests that we should also compute the convolution of the signal with the prefilter. We will then have computed two convolution results: the prefiltered original, and the derivative of the prefiltered original. Thus, it

[1]For example, when computing the directional derivative as a linear combination of axis derivatives.

makes sense that we should *simultaneously* design the derivative kernel and the prefilter kernel such that the convolution results will have the proper relationship.

3. DESIGN OF DERIVATIVE FILTERS

We will describe the filter design problem in the frequency domain. Let $P(\vec{\omega})$ be the Discrete Time Fourier transform (DTFT) of the prefilter, and $D(\vec{\omega})$ the DTFT of the derivative filter. Then our design method must attempt to meet the following requirements:

1. The derivative filters must be very good approximations to the derivative of the prefilter. That is, for a derivative along the x-axis, we would like $j\omega_x P(\vec{\omega}) \approx D(\vec{\omega})$, where ω_x is the component of the frequency coordinate in the x direction.

2. The lowpass prefilter should have linear phase, preferably zero phase.[2]

3. For computational efficiency and ease of design, it is preferable that the prefilter be separable. In this case, the derivatives will also be separable, and the design problem will be reduced to one dimension.

4. The design algorithm should include a model for signal and noise statistics, as in [2].

5. The prefilter should be rotationally symmetric.

We will not attempt to achieve rotational symmetry, since this constraint coupled with separability would force the prefilter to be Gaussian.

We can write a weighted least-squares error function in the frequency domain as follows:

$$E(P, D) = \int d\omega W(\omega) \left[-j\omega P(\omega) - D(\omega)\right]^2 .$$

$W(\omega)$ is a frequency weighting function. For the designs in this paper, we have chosen this function to roughly mimic the expected spectral content of natural images: $W(\omega) = 1/|\omega|^{0.5}$. With this simple error measure, we can compute solutions analytically and avoid complex optimization procedures that may get stuck in local minima.

In particular, the minimizing kernels may easily be found using eigenvector techniques. Let $\vec{p}$ be an N-vector containing the prefilter kernel, and let $\vec{d}$ be an N-vector[3] containing the derivative filter kernel. We may write a discrete approximation to the error function as:

$$E(\vec{p}, \vec{d}) = \left| W \left(F'\vec{p} - F\vec{d} \right) \right|^2 ,$$

where F is a matrix whose columns are the first N Fourier basis functions, and F' is a similar matrix containing the first N Fourier basis functions multiplied by ω. W is a diagonal matrix containing the frequency weighting function.

We now consolidate the terms of the equation as follows:

$$E(\vec{u}) = \left| M\vec{u} \right|^2 ,$$

where

$$M = \left(WF' \quad | \quad WF \right), \quad \text{and} \quad \vec{u} = \begin{pmatrix} \vec{p} \\ \vec{d} \end{pmatrix} .$$

The minimizing vector $\vec{u}$ is simply the minimal-eigenvalue eigenvector of the matrix $M^t M$. The actual filters are found by renormalizing this $\vec{u}$ so that $\vec{p}$ has unit D.C. response (i.e., the samples sum to 1.0). This design technique may be readily extended to higher-order derivatives.

4. EXAMPLE FILTER DESIGNS

We have designed a set of filter kernel pairs of different sizes, using the method described above. In all cases, we computed the solution using 1024-point FFTs. All prefilters were forced to be symmetric about their midpoints. The corresponding derivative filters are therefore anti-symmetric. The resulting filter taps are given in the table in figure 1.

For each kernel pair, figure 2 shows both the Fourier magnitude of the one-dimensional derivative kernel, and $|\vec{\omega}|$ times the Fourier magnitude of the prefilter. Consider, for example, the upper-left plot in figure 2. This shows a dashed graph of the Fourier magnitude of the two-point (sample difference) derivative operator $d_2(n) = [-0.686, 0.686]$. On the same plot, we show the Fourier magnitude of the *derivative* of the two-point averaging operator $p_2(n) = [0.5, 0.5]$, which is computed by multiplying its Fourier magnitude by the function $|\omega|$. Note that the derivative will be *underestimated* for low frequencies and *overestimated* for high frequencies.

Thus, unless the original continuous imagery is bandlimited to a region around a crossing point of the two curves, these filters are not very good derivative approximations. Of course, the derivative filter may be rescaled to improve the estimates at a particular frequency (the standard choice is $d_2(n) = [-1, 1]$), but unless this is done adaptively (i.e., matched to the signal spectrum), this will not improve the situation.

[2] A phase of $e^{-j\vec{\omega}/2}$ is also acceptable, producing estimates of the signal derivative *between* the samples.

[3] Note that the two kernels need not be the same size. For simplicity, we choose them to be so in this paper.

filter \ n	0	1	2
p_2	0.5		
d_2	0.6860408186912537		
p_3	0.5515803694725037	0.22420981526374817	
d_3	0.0	0.45527133345603943	
p_4	0.40843233466148377	0.09156766533851624	
d_4	0.27143725752830506	0.2362217754125595	
p_5	0.4308556616306305	0.24887460470199585	0.035697564482688904
d_5	0.0	0.2826710343360901	0.1076628714799881

Figure 1. Matched pairs of prefilter (p_i) and derivative (d_i) kernels. Shown are half of the taps for each filter: the others are determined by symmetry. All prefilters are symmetric and all derivative filters are anti-symmetric. Even-length kernels are (anti-)symmetric about $n = -0.5$, odd-length filters are (anti-)symmetric about $n = 0$.

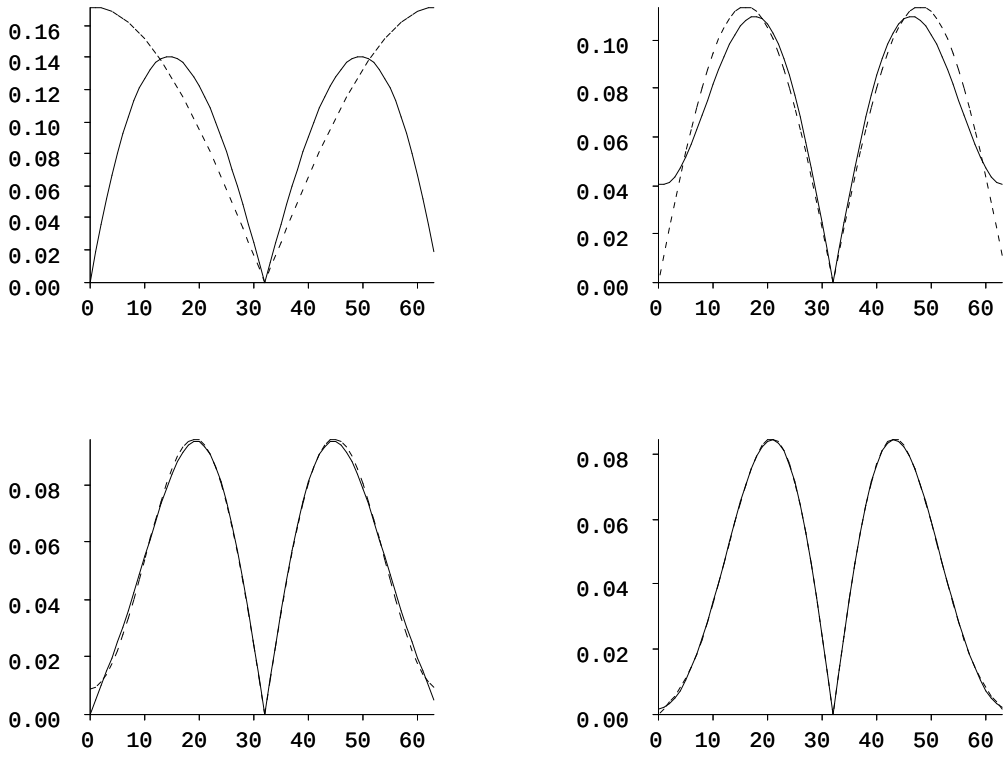

Figure 2. Illustration of 2-tap (upper left), 3-tap (upper right), 4-tap (lower left), and 5-tap (lower right) derivative/prefilter pairs. Shown are the magnitude of the Fourier transforms over the range $-\pi < \omega < \pi$ of: a) the derivative kernel (dashed line), and b) the frequency-domain derivative of the prefilter (that is, its Fourier magnitude multiplied by $|\omega|$). If these were perfect derivative/prefilter pairs, the curves would coincide. Note that the matches would be substantially worsened if we do not perform prefiltering (i.e., if we use an impulse as a prefilter kernel).

Many authors have pointed out that derivative techniques are often inaccurate and are not robust to noise. For example, Kearney et. al. [4] noted that gradient-based optical flow algorithms often perform poorly in highly "textured" regions of an image. We believe that this is primarily due to the choice of a standard "differencing" prefilter/derivative pair. Gradient-based approaches have been shown to surpass most other optical flow techniques in accuracy [1, 5].

In considering figure 2, notice that the three-tap filter is substantially more accurate than the two-tap filter. Accuracy improves further with the four- and five-tap filters.

5. EXPERIMENTAL RESULTS

We have tested the filters designed above in a simple orientation-estimation task. We generated images containing sinusoidal grating patterns of varying spatial frequency and orientation. For each pattern, we computed the gradient via separable convolution with the derivative/prefilter pair. For example, the x-derivative is computed by convolving in the vertical direction with the prefilter and in the horizontal direction with the derivative filter.

We then use these derivative measurements to compute a least-squares estimate of the orientation θ of the pattern. An error function is written as:

$$E(\theta) = \sum \left[\cos(\theta) I_x + \sin(\theta) I_y\right]^2$$

where I_x is the x-derivative image, I_y is the y-derivative image, and the summation is over all image pixels. We solve for the maximizing unit vector $\hat{u}(\theta) = [\cos(\theta), \sin(\theta)]$ using standard eigenvector techniques and then solve for the orientation, θ. Errors in orientation estimation as a function of pattern orientation are shown in figure 3. Errors as a function of pattern spatial frequency are shown in figure 4. Note that the errors are unacceptable for the 2-tap filter pair, but rapidly improve as we increase the size of the kernels.

We have also tested the filters described above in applications of estimating local orientation in still images (cf Freeman & Adelson [3]), and for optical flow estimation [5]. Due to space constraints, these results cannot be elaborated here. They indicate substantial improvements over conventional derivative measurements, as would be predicted from the plots of figure 2.

6. REFERENCES

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. Technical Report RPL-TR-9107, Queen's University, Kingston, Ontario, Robotics and Perception Laboratory Technical Report, July 1992.

[2] B. Carlsson, A. Ahlen, and M. Sternad. Optimal differentiators based on stochastic signal models. *IEEE Trans. Signal Proc.*, 39(2), February 1991.

[3] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Pat. Anal. Mach. Intell.*, 13(9):891–906, 1991.

[4] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Pat. Anal. Mach. Intell.*, 9(2):229–244, 1987.

[5] E. P. Simoncelli. *Distributed Analysis and Representation of Visual Motion.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, January 1993. Also available as MIT Media Laboratory Vision and Modeling Technical Report #209.

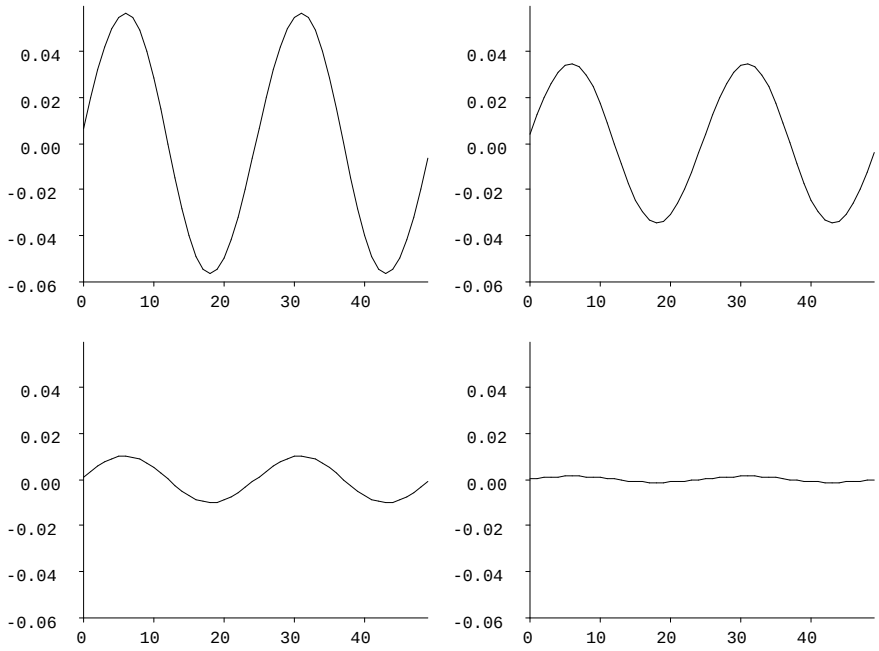

Figure 3. Test of rotation-invariance for the filter pairs shown in 2. We plot the orientation estimation error (in radians) as a function of pattern orientation, for a set of pattern orientations ranging from $-\pi/2$ to $\pi/2$ (axis labels are arbitrary). Spatial frequency was held constant at $\omega = \pi/2$.

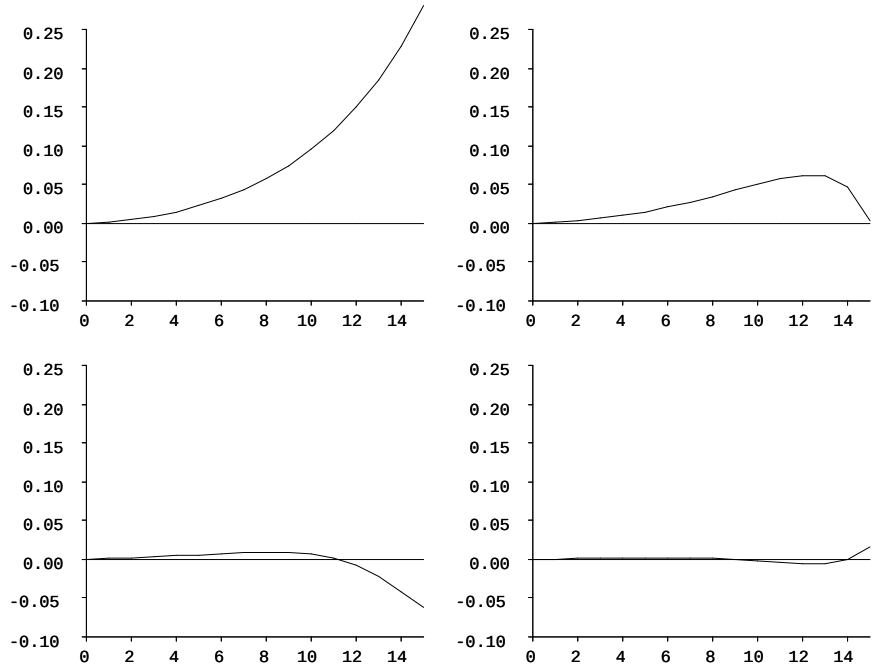

Figure 4. Test of scale-invariance for the filter pairs shown in 2. We plot the orientation estimation error (in radians) as a function of pattern spatial frequency, for a set of pattern frequencies ranging from $\pi/32$ to $31\pi/32$ (axis labels are arbitrary). Orientation was held constant at $\theta = \pi/6$. Horizontal lines provide a zero-error reference.

Errata: the frequency-weighting function in the equation giving the least-squares error should be squared, and the minus sign on $j\omega$ is incorrect (relative to the convention of the rest of the paper):

$$E(P, D) = \int d\omega W^2(\omega) \left[j\omega P(\omega) - D(\omega) \right]^2 .$$