Presented at: 7th Int'l Conf Computer Analysis of Images and Patterns,
Kiel, Germany. September 10-12, 1997.

Optimally Rotation-Equivariant Directional Derivative Kernels*

Hany Farid[1] and Eero P. Simoncelli[2]

[1] University of Pennsylvania, Philadelphia PA 19104-6228, USA
[2] New York University, New York, NY 10012, USA

Abstract. We describe a framework for the design of directional derivative kernels for two-dimensional discrete signals in which we optimize a measure of rotation-equivariance in the Fourier domain. The formulation is applicable to first-order and higher-order derivatives. We design a set of compact, separable, linear-phase derivative kernels of different orders and demonstrate their accuracy.

1 Introduction

A wide variety of algorithms in multi-dimensional signal processing are based on the computation of directional derivatives. For example, gradient measurements are used in computer vision as a first stage of many edge detection, depth-from-stereo, and optical flow algorithms. The motivation for such decompositions usually stems from a desire to locally characterize signals using Taylor series expansions (e.g., [7]).

Derivatives of discretely sampled signals are often computed as differences between neighboring sample values. This type of differentiation arises naturally from the definition of continuous derivatives, and is reasonable when the spacing between samples is well below the Nyquist limit. For example, it is used throughout the numerical analysis literature, where one typically has control over the sample spacing. But such differences are poor approximations to derivatives when the distance between samples is large and cannot be adjusted.

In the digital signal processing community, there has been a fair amount of work on the design of discrete differentiators (see e.g., [9]). This work is usually based on approximating the derivative of a continuous sinc function. The difficulty with this approach is that the resulting kernels typically need to be quite large in order to be accurate.

In addition to the difficulties described above, these two primary methods of differentiation are not well suited for multi-dimensional differentiation. In particular, one often relies on the linear-algebraic properties of multi-dimensional derivatives (gradients) that allow differentiation in arbitrary directions via linear combinations of separable axis derivatives[3]. In the computer vision literature,

* This work was supported by ARO Grant DAAH04-96-1-0007, DARPA Grant N00014-92-J-1647, NSF Grant SBR89-20230, and NSF CAREER Grant MIP-9796040 to EPS.

[3] For example, the derivative operator in the direction of unit vector $\hat{u}$ is $u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y}$.

many authors have used sampled Gaussian derivatives which exhibit better approximations to these algebraic properties than simple differences, but less computationally expensive than sinc functions. Danielsson [3] compared a number of derivative kernels and concluded that the Sobel operators exhibited the most rotation-equivariant behavior. Freeman and Adelson [8] characterized the complete class of rotation-equivariant kernels and termed these "steerable" filters.

We are interested in the optimal design of small separable kernels for efficient discrete differentiation. In previous work [10, 1], we described design techniques for matched pairs of one-dimensional kernels (a lowpass kernel and a differentiator) suitable for multi-dimensional differentiation. Axis derivatives were computed by applying the differentiator along the axis of choice and the lowpass kernel along all remaining axes. The error functional was a weighted least-squares error in the Fourier domain between the differentiator and the derivative of the lowpass kernel. In this paper, we generalize these notions to form a two-dimensional error functional that expresses the desired property of derivative kernels discussed above. This error functional is then minimized to produce a set of optimally rotation-equivariant derivative kernels.

2 Differentiation of Discrete Signals

Differentiation is an operation defined on continuous functions. The computation of derivatives on a discretely sampled function thus requires (at least implicitly) an intermediate interpolation step. The derivative of this interpolated continuous function is then re-sampled at the points of the original sampling lattice.

2.1 Example: Ideal Interpolation

To make this more precise, consider the classical situation in which the sampled function is assumed to have been formed by uniformly sampling a continuous function at the Nyquist rate. In this case, the correct interpolation of the discrete function $f[\cdot]$ is:

$$\overline{f}(x,y) = \sum_{k,l} f[k,l] \cdot c(x - kT, y - lT), \tag{1}$$

where T is the sample spacing (assumed to be identical along both the x and y axes), $\overline{f}(x,y)$ is the interpolated continuous function, and the interpolation function $c(x,y)$ is a separable product of ideal lowpass ("sinc") functions, $c(x,y) = s_T(x)\ s_T(y) = \frac{\sin(\pi x/T)}{\pi x/T}\ \frac{\sin(\pi y/T)}{\pi y/T}$. Assuming that the sum in Equation (1) converges uniformly, we can differentiate both sides of the equation. Without loss of generality, consider the partial derivative with respect to x:

$$D_x\{\overline{f}\}(x,y) = \sum_{k,l} f[k,l] \cdot D_x\{c\}(x - kT, y - lT), \tag{2}$$

where $D_x\{\cdot\}$ indicates a functional that computes the partial derivative of its argument in the x direction. Note that the derivative operator is only being applied to continuous functions, $\overline{f}$ and c.

One arrives at a definition of the derivative of the discrete signal by sampling both sides of the above equation on the original sampling lattice:

$$\begin{aligned} D_x\{\overline{f}\}(x,y)|_{x=nT,y=mT} &= \sum_{k,l} f[k,l] \cdot D_x\{c\}((n-k)T,(m-l)T) \\ &= \sum_{k,l} f[k,l] \cdot D\{s_T\}((n-k)T) \cdot s_T((m-l)T) \\ &= \sum_{k,l} f[k,l] \cdot d_T[n-k] \cdot \delta[m-l], \end{aligned} \tag{3}$$

where $d_T[\cdot]$ is the T-sampled sinc derivative, and $\delta[\cdot]$ is the T-sampled sinc (i.e., a Kroenecker delta function). Note that the right side of this expression is a convolution of the discretely sampled function, $f[\cdot]$, with the separable kernel $d_T[n-k]\delta[m-l]$. The continuous interpolation need never be performed.

If the original function was sampled at the Nyquist rate, then convolution with the sampled derivative of the sinc function will return an exact sampled derivative. In practice, however, the coefficients of this kernel decay very slowly and accurate implementation requires very large kernels. In addition, the sinc derivative operator has a large response at high frequencies, making it fragile in the presence of noise.

2.2 Alternative Interpolation Functions

The limitations of the sinc function lead us to consider alternative interpolation functions. Of course, if an interpolator other than the sinc function is used, the resulting derivative may not be that of the original continuous function. However, for many applications this is not a fundamental concern. Consider, for example, the problem of determining the local orientation of an edge. This can be achieved by measuring the gradient vector, which is perpendicular to the edge. If we use an interpolation kernel which preserves the structure of the edge, the gradient direction will still provide the desired information.

Since the separability of the sinc is desirable for computational efficiency (e.g., [5, 4]), we will consider an interpolator that retains this property, and will also assume that the two axes should be treated identically. Thus, the two-dimensional interpolator is written as a separable product, $c(x,y) = d_0(x) \cdot d_0(y)$. The partial derivative (with respect to x) of this interpolator is:

$$D_x\{c\}(x,y) = d_1(x) \cdot d_0(y), \tag{4}$$

where $d_1(x)$ is the derivative of $d_0(x)$. With this interpolator, the sampled derivative (as in Equation (3)) becomes:

$$\begin{aligned} \mathcal{D}_x\{\overline{f}\}(x,y)|_{x=nT,y=mT} &= \sum_{k,l} f[k,l] \cdot d_1((n-k)T) \cdot d_0((m-l)T) \\ &= \sum_{k,l} f[k,l] \cdot d_1[n-k] \cdot d_0[m-l]. \end{aligned} \tag{5}$$

The discrete derivatives are computed using *two* discrete one-dimensional kernels, $d_0[\cdot]$ and $d_1[\cdot]$, which are the T-sampled versions of $d_0(\cdot)$ and $d_1(\cdot)$, respectively. Note that the separability of the interpolator is retained in the derivative operator. As with the sinc function, we need never make explicit the underlying continuous function $d_0(\cdot)$.

At this point, we could simply choose a continuous function $d_0(\cdot)$, compute its derivative $d_1(\cdot)$, and T-sample the two functions. For example, it is common in computer vision to use a sampled Gaussian and its derivative. However, because the Gaussian is not strictly bandlimited, sampling introduces artifacts, thus destroying the derivative relationship between the resulting kernels. So, instead we choose to simultaneously design a pair of *discrete* kernels that optimally preserve the required derivative relationship.

To design such a pair of discrete kernels, we must state the differential relationship between them. Previously [1], pairs of kernels, $d_0[n]$ and $d_1[n]$, were designed such that their Fourier transforms approximate the correct derivative relationship:

$$j\omega D_0(\omega) = D_1(\omega), \qquad -\pi < \omega < \pi, \tag{6}$$

where capitalized functions correspond to the (continuous but periodic) discrete-space Fourier transform, $D_0(\omega) = \sum_n d_0[n]e^{-j\omega n}$. This constraint states that the derivative (in the Fourier domain) of the kernel $d_0[n]$ is equal to the kernel $d_1[n]$. For the current paper, we wish to impose a similar constraint in the two-dimensional Fourier domain. In particular, the derivative in an arbitrary direction (specified by a unit-vector $\hat{u}$) is:

$$\mathcal{D}_{\hat{u}}\{f\}(x,y) = u_x\mathcal{D}_x\{f\}(x,y) + u_y\mathcal{D}_y\{f\}(x,y), \tag{7}$$

where (u_x, u_y) are the components of $\hat{u}$. Thus, the two-dimensional version of the Fourier domain constraint of Equation (6) is:

$$j(\hat{u}\cdot\hat{\omega})D_0(\omega_x)D_0(\omega_y) = [u_xD_1(\omega_x)D_0(\omega_y) + u_yD_0(\omega_x)D_1(\omega_y)], \tag{8}$$

and should hold for $-\pi < \{\omega_x, \omega_y\} < \pi$ and for all unit vectors $\hat{u}$ (i.e., for all directions). We can now define a weighted least-squares error functional by integrating over these variables:

$$E\{D_0, D_1\} = \int_{\hat{\omega}} W(\hat{\omega}) \cdot \int_{\hat{u}} [j(\omega_x u_x + \omega_y u_y)D_0(\omega_x)D_0(\omega_y) - (u_xD_1(\omega_x)D_0(\omega_y) + u_yD_0(\omega_x)D_1(\omega_y))]^2, \tag{9}$$

where $W(\hat{\omega})$ is a weighting function. In order to avoid the trivial (zero) solution, we impose a constraint that the interpolator have unit response at D.C. (i.e., the kernel $d_0[n]$ has unit sum): $D_0(0) = 1$.

2.3 Higher-Order Derivatives

Higher-order derivative kernels may be designed using a similar strategy to that introduced in the previous section. In particular, the Nth-order directional derivative in direction $\hat{u}$ is:

$$\begin{aligned}\mathcal{D}_{\hat{u}}^N\{\overline{f}\}(x,y) &= [u_x\mathcal{D}_x + u_y\mathcal{D}_y]^N\{\overline{f}\}(x,y) \\ &= \sum_{p=0}^{N} b[p,N]u_x^p\ u_y^{(N-p)}\ \mathcal{D}_x^p\mathcal{D}_y^{(N-p)}\{\overline{f}\}(x,y). \end{aligned} \tag{10}$$

where $b[p,N] = N!/p!(N-p)!$ is the binomial coefficient. Combining this definition with the interpolation of Equation (2), and sampling both sides gives an expression for the discrete Nth-order directional derivative:

$$\begin{aligned}\mathcal{D}_{\hat{u}}^N\{\overline{f}\}(x,y)|_{x=nT,y=mT} = &\sum_{p=0}^{N} b[p,N]u_x^p\ u_y^{(N-p)}\cdot \\ &\sum_{k,l} f[k,l]d_p[n-k]d_{N-p}[m-l]. \end{aligned} \tag{11}$$

This expression is a sum of convolutions with separable kernels composed of a set of discrete one-dimensional derivative (and interpolation) kernels $\{d_p[n] \mid p = 0,1,\ldots,N\}$. As before, we place a constraint on these kernels in the Fourier domain:

$$j^N(\hat{u}\cdot\hat{\omega})^N D_0(\omega_x)D_0(\omega_y) = \sum_{p=0}^{N} b[p,N]u_x^p\ u_y^{(N-p)}\ D_p(\omega_x)D_{(N-p)}(\omega_y). \tag{12}$$

A least-squares error functional is formed by integrating over orientation and the two frequency axes:

$$\begin{aligned}E\{D_0,D_1,\ldots,D_N\} = &\int_{\hat{\omega}} W(\hat{\omega})\cdot \\ &\int_{\hat{u}} \left(\sum_{p=0}^{N} b[p,N]u_x^p\ u_y^{(N-p)}\ \left[j^N\omega_x^p\omega_y^{(N-p)}D_0(\omega_x)D_0(\omega_y) - D_p(\omega_x)D_{N-p}(\omega_y)\right]\right)^2 \end{aligned} \tag{13}$$

In order to avoid the trivial solution, we again impose a constraint that the interpolator ($d_0[n]$) have unit response at D.C.: $D_0(0) = 1$.

3 Results

The error functionals in Equations (9) and (13) are both fourth-order in the optimization variables and cannot be optimized analytically. In order to obtain solutions, we fix the size of the kernels, and use conjugate gradient descent. As a starting point for the first-order kernels, we use the solution that minimizes the linear one-dimensional constraint of [1]:

$$E\{D_0,D_1\} = \int_{\omega} W'(\omega)[j\omega D_0(\omega) - D_1(\omega)]^2, \tag{14}$$

subject to $D_0(0) = 1$, with a weighting function of $W'(\omega) = 1/(|\omega| + \frac{\pi}{4})$. For the Nth-order kernels, we start from the solution that minimizes the linear one-dimensional set of constraints:

$$(j\omega)^{i-j} D_j(\omega) = D_i(\omega), \tag{15}$$

for $0 \leq j < i \leq N$, again subject to the constraint that $D_0(0) = 1$. For the weighting functions $W(\hat{\omega})$, we choose a "fractal" weighting of $W(\hat{\omega}) = 1/(\omega_x^2 + \omega_y^2 + \pi^2/16)$.

Based on this design, Table 1 gives a set of first-order derivative kernels of different sizes. Figure 1 shows a comparison of the differentiator $d_1[n]$ with the derivative of the interpolator $d_0[n]$, computed in the Fourier domain. If these kernels were perfectly matched (i.e., $d_1[n]$ is the derivative of $d_0[n]$), then the two curves should coincide. Also shown in this figure is a comparison of these kernels to a variety of other derivative kernels. For a fair comparison, the variance of the Gaussian in this figure was chosen so as to optimize the 1-D constraint of Equation (6). Note that even under these conditions, our kernels better preserve the required derivative relationship between the differentiator and interpolator kernel. Note also that the resulting differentiation kernels are bandpass in nature, and thus less susceptible to noise than typical sinc approximations.

Since the derivative kernels are designed for rotation-equivariance, we consider the application of estimating the orientation of a two-dimensional sinusoidal grating from the horizontal and vertical partial derivatives. The grating had a fixed orientation of 22.5 degrees and spatial frequency in the range $[1, 21]/64$ cycles/pixel. Figure 2 shows the estimation error as a function of spatial frequency for a variety of derivative kernels. In this example, orientation was determined using a total least squares estimator over a 16×16 patch of pixels in the center of the image. Note that the errors for our optimal kernels are substantially smaller than the other filters. The reasonably good performance of the Gaussian is due, in part, to our optimization of its variance. And finally, shown in Table 2 are a set of higher-order derivative kernels of different sizes.

d_0	0.223755	0.552490	0.223755			
d_1	-0.453014	0	0.453014			
d_0	0.092645	0.407355	0.407355	0.092645		
d_1	-0.236506	-0.267576	0.267576	0.236506		
d_0	0.036420	0.248972	0.429217	0.248972	0.036420	
d_1	-0.108415	-0.280353	0	0.280353	0.108415	
d_0	0.013846	0.135816	0.350337	0.350337	0.135816	0.013846
d_1	-0.046266	-0.203121	-0.158152	0.158152	0.203121	0.046266

Table 1. First-order derivative kernels. Shown are pairs of derivative ($d_1[n]$) and interpolator ($d_0[n]$) kernels of various sizes.

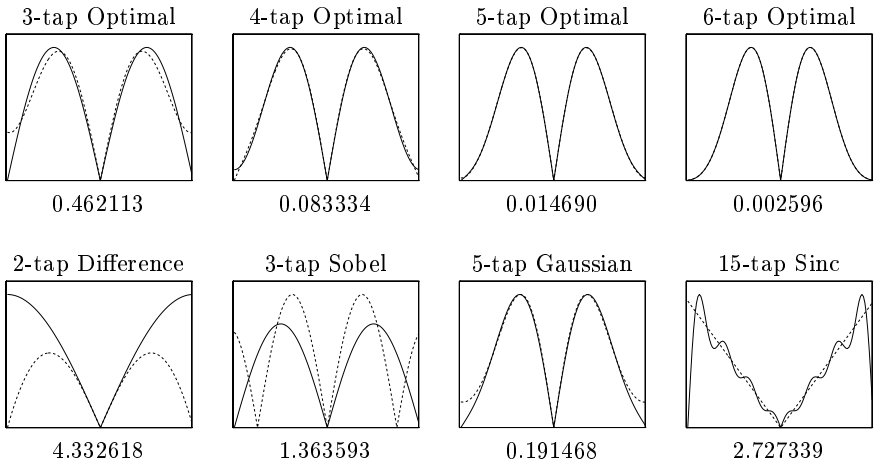

Fig. 1. First-order derivative kernels. Illustrated in each panel are the magnitude of the Fourier transform of the derivative kernel (solid line) and the frequency-domain derivative of the interpolator (dashed line) for our optimally designed kernels (see Table 1). Also illustrated, for comparison, are several other derivative kernels. Beneath the plots are the weighted RMS errors.

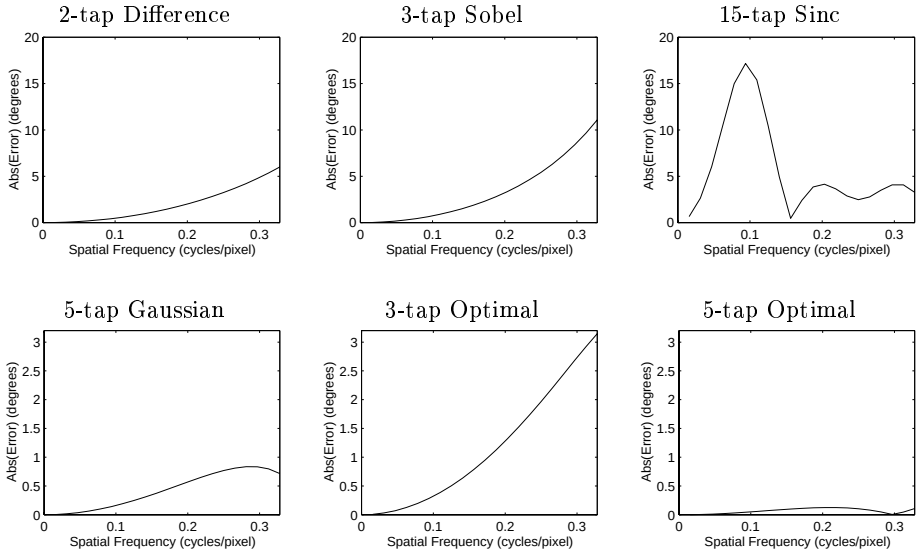

Fig. 2. Differential estimation of orientation. Illustrated is the absolute value of the error in an orientation estimation task for a sinusoidal grating as a function of spatial frequency. The grating was oriented at 22.5^o, the angle of maximal error. Note that top and bottom rows have different Y-axis scaling.

d_0	0.026455	0.248070	0.450951	0.248070	0.026455		
d_1	-0.097537	-0.309308	0	0.309308	0.097537		
d_2	0.236427	0.020404	-0.517610	0.020404	0.236427		
d_0	0.004423	0.121224	0.374352	0.374352	0.121224	0.004423	
d_1	-0.029091	-0.223003	-0.196061	0.196061	0.223003	0.029091	
d_2	0.105303	0.198956	-0.301356	-0.301356	0.198956	0.105303	
d_3	-0.230922	0.227030	0.503368	-0.503368	-0.227030	0.230922	
d_0	0.002013	0.051225	0.247548	0.398427	0.247548	0.051225	0.002013
d_1	-0.008593	-0.115977	-0.240265	0	0.240265	0.115977	0.008593
d_2	0.033589	0.180366	-0.028225	-0.370850	-0.028225	0.180366	0.033589
d_3	-0.107517	-0.074893	0.469550	0	-0.469550	0.074893	0.107517
d_4	0.201624	-0.424658	-0.252747	0.940351	-0.252747	-0.424658	0.201624

Table 2. Higher-order derivative kernels. Shown are sets of derivative ($d_p[n], p > 0$) and interpolator ($d_0[n]$) kernels for differentiation orders $N = 2, 3, 4$.

4 Conclusions

Differentiation of discretized signals is a very basic operation, widely used in numerical analysis, image processing, and computer vision. We have described a framework for the design of operators that are efficient (i.e., compact and separable) and optimally equivariant to rotations. This formulation can easily be extended to higher dimensions (e.g., three-dimensional derivatives for motion).

References

1. E P Simoncelli. Design of multi-dimensional derivative filters. In *First Int'l Conf on Image Processing*, Austin, Texas, November 1994.
2. B. Carlsson, A Ahlen, and M. Sternad. Optimal differentiators based on stochastic signal models. 39(2), February 1991.
3. Per-Erik Danielsson. Rotation-invariant linear operators with directional response. In *5th Int'l Conf. Patt. Rec.*, Miami, December 1980.
4. J De Vriendt. Fast computation of unbiased intensity derivatives in images using separable filters. *Int'l Journal of Computer Vision*, 13(3):259–269, 1994.
5. T Vieville and O Faugeras. Robust and fast computation of unbiased intensity derivatives in images. In *ECCV*, pages 203–211. Springer-Verlag, 1992.
6. R Deriche. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:78–87, 1990.
7. J J Koenderink and A J van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
8. W T Freeman and E H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
9. A V Oppenheim and R W Schafer. *Discrete-Time Signal Processing.* Prentice Hall, 1989.
10. E P Simoncelli. *Distributed Analysis and Representation of Visual Motion.* PhD thesis, EECS Dept., MIT, January 1993.

This article was processed using the LaTeX macro package with LLNCS style