Deformable kernels for early vision

Pietro Perona

Università di Padova — Dipartimento di Elettronica ed Informatica — Via Gradenigo 6A — 35131 Padova — Italy *and*
Laboratory for Information and Decision Systems — Massachusetts Institute of Technology, Rm. 35-312 — Cambridge, MA 02139
e-mail: perona@verona.mit.edu

Abstract

Early vision algorithms often have a first stage of linear-filtering that 'extracts' from the image information at multiple scales of resolution and multiple orientations. A common difficulty in the design and implementation of such schemes is that one feels compelled to discretize coarsely the space of scales and orientations in order to reduce computation and storage costs. This discretization produces anisotropies due to a loss of traslation-, rotation- scaling- invariance that makes early vision algorithms less precise and more difficult to design. This need not be so: one can compute and store efficiently the response of families of linear filters defined on a *continuum* of orientations and scales. A technique is presented that allows (1) to compute the best approximation of a given family using linear combinations of a small number of 'basis' functions; (2) to describe all finite-dimensional families, i.e. the families of filters for which a finite dimensional representation is possible with no error. The technique is general and can be applied to generating filters in arbitrary dimensions. Experimental results are presented that demonstrate the applicabilityof the technique to generating multi-orientation multi-scale 2D edge-detection kernels. The implementation issues are also discussed.

1 Introduction

A number of early vision and signal processing algorithms involve convolving the image with kernels at multiple orientations and scales. To cite maybe the earliest example, Granlund [7] suggested that an operator that computed an 'oriented energy' for each point of an image would be a useful first step for a variety of picture processing operations. He proposed obtaining this oriented energy by filtering the image with kernels localized in frequency and orientation, and taking at each point of the image the global maximum of the filter responses with respect to orientation θ. A number of schemes based on a similar style of filtering, localized in frequency, position and orientation, has been proposed in the literature for analyzing texture, motion, brightness and stereo information using.

Since edges, lines, textures, motions can exist at all possible orientations and scales of resolution one would like to be able to use families of filters that are tuned to all orientations. Typically one designs an 'optimal' kernel for a specific application and would like to convolve the image with deformations (rotations, scalings) of this 'template'. In reality one can only perform a finite (and small) number of filtering operations, hence the common practice of 'sampling' the set of orientations and scales. This operation has the strong drawback of introducing anisotropies and algorithmic difficulties in the computational implementations. It would be preferable to keep thinking in terms of a continuum, of angles for example, and be able to localize the orientation of an edge with the maximum accuracy allowed by the filter one has chosen.

This aim may sometimes be achieved by means of interpolation: one convolves the image with a small set of kernels, say at a number of discrete orientations, and obtains the result of the convolution at *any* orientation by taking linear combinations of the results. Since convolution is a linear operation the interpolation problem may be formulated just in terms of the kernels (for the sake of simplicity the case of rotations in the plane is discussed here): Given a kernel $F : \mathbf{R}^2 \rightarrow \mathbf{C}^1$, define the family of 'rotated' copies of F as: $F_\theta = F \circ R_\theta$, $\theta \in \mathbf{S}^1$, where $\mathbf{S}^1$ is the circle and R_θ is a rotation. Is it possible to express F_θ as

$$F_\theta(\mathbf{x}) = \sum_{i=1}^{n} \alpha(\theta)_i G_i(\mathbf{x}) \quad \forall \theta \in \mathbf{S}^1, \forall \mathbf{x} \in \mathbf{R}^2 \tag{1}$$

a *finite* linear combination of functions $G_i : \mathbf{R}^2 \rightarrow \mathbf{C}^1$?

An example of 'rotating' families of kernels that have a finite representation is well known: the first derivative along an arbitrary direction of a round Gaussian may be obtained by linear combination of the X- and Y-derivatives of the same. The common implementations of the Canny edge detector [4] are based on this principle. Unfortunately this function has poor orientation selectivity and therefore it is unsuited for edge detection if one wants to recover edge-junctions (see in Fig. 1 the comparison with a detector that uses narrow orientation-selective filters). Freeman and Adelson have recently noted that higher order derivatives of Gaussians, have the same property [5] (they call it "steerability"). These functions have higher orientation selectivity and can be used for contour detection and signal processing [6]. However, for most functions F of interest a finite decomposition of F_θ as in Eq. (1) cannot be found. For example the elongated kernels used in edge detection by [13, 14] do not have a finite decomposition as in Eq. (1).

One needs an approximation technique that, given an F_θ, allows one to generate a function $G_\theta^{[n]}$ which is sufficiently similar to F_θ and that can be expressed as a finite sum of n terms as in (1). How can one find the best approximating $G_\theta^{[n]}$? A different design perspective could also be taken: given a number n of filtering operations allowed, synthesize the best (with respect to the specific task at hand) kernel *within* the class of functions that can be *exactly* represented by a sum of n terms. Therefore it is useful to be able to answer to the question: What is the set of functions that can be represented exactly as in Eq. (1)? Neither this question, nor the approximation question have yet been addressed in the literature so far.

This paper is organized as follows: the special case of the rotations (Eq. (1)) will be solved in section 2 with the purpose of developing some intuition for the method used to approach the general case. In section 3 and 3.1 the formalism used to solve the special case of rotations will be extended to more general transformations. The proofs are omitted and may be found in [12]. In section 4 experimental results and practical issues are presented and discussed for the case of rotations and scalings.

2 Steerable approximations

In order to solve the approximation problem proposed in the introduction one needs of course to define the 'quality' of the approximation $G_\theta^{[n]} \approx F_\theta$. There are two reasonable choices: (a) a distance $D(F_\theta, G_\theta^{[n]})$ in the space $\mathbf{R}^2 \times \mathbf{S}^1$ where F_θ is defined; (b) if F_θ is the kernel of some filter one is interested in the worst-case error in the 'output' space: the maximum distance $d(\langle F_\theta, f\rangle, \langle G_\theta^{[n]}, f\rangle)$ over all unit-norm f defined on $\mathbf{R}^2$. The symbols Δ_n and δ_n will indicate the 'optimal' distances, i.e. the minimum possible approximation errors using n components. These quantities may be defined using the distances induced by the L^2-norm:

Definition.

$$D_n(F_\theta, G_\theta^{[n]}) = \|F_\theta - G_\theta^{[n]}\|_{\mathbf{R}^2\times\mathbf{S}^1}$$

$$\Delta_n(F_\theta) = \inf_{G_\theta^{[n]}} D_n(F_\theta, G_\theta^{[n]})$$

$$d_n(F_\theta, G_\theta^{[n]}) = \sup_{\|f\|=1} \|\langle F_\theta - G_\theta^{[n]}, f\rangle_{\mathbf{R}^2}\|_{\mathbf{S}^1}$$

$$\delta_n(F_\theta) = \inf_{G_\theta^{[n]}} d_n(F_\theta, G_\theta^{[n]})$$

Consider the approximation to F_θ defined as follows:

Definition. Call $F_\theta^{[n]}$ the n-terms sum:

$$F_\theta^{[n]} = \sum_{i=1}^{n} \sigma_i a_i(\mathbf{x}) b_i(\theta) \tag{2}$$

with σ_i, a_i and b_i defined in the following way: let $\hat{h}(\nu)$ be the (discrete) Fourier transform of the function $h(\theta)$ defined by:

$$h(\theta) = \int_{\mathbf{R}^2} F_\theta(\mathbf{x}) \overline{F_{\theta'=0}(\mathbf{x})} d\mathbf{x} \tag{3}$$

and let ν_i be the frequences on which $\hat{h}(\nu)$ is defined, ordered in such a way that $\hat{h}(\nu_i) \geq \hat{h}(\nu_j)$ if $i \leq j$. Call $N \leq \infty$ the number of nonzero terms $\hat{h}(\nu_i)$. Set now:

$$\sigma_i = \hat{h}(\nu_i)^{1/2} \tag{4}$$

$$b_i(\theta) = e^{j2\pi\nu_i\theta} \tag{5}$$

$$a_i(\mathbf{x}) = \sigma_i^{-1} \int_{\mathbf{S}^1} \overline{F_\theta(\mathbf{x})} e^{j2\pi\nu_i\theta} d\theta \tag{6}$$

Then $F_\theta^{[n]}$ is the best n-dimensional approximation to F_θ in the following sense:

Theorem 1 *Given the definitions and notation introduced above, suppose that $F \in L^2(\mathbf{R}^2)$ then:*

1. *$\{a_i\}$ and $\{b_i\}$ are orthonormal sequences of functions.*
2. *$F_\theta^{[N]}$ is the smallest possible exact representation of F_θ, i.e. if $\exists\, M, \beta_i, g_i$ s.t. $F_\theta(\mathbf{x}) = \sum_{i=1}^{M} \beta_i(\theta) g_i(\mathbf{x})$ then $M \geq N$.*
3. *The number N of terms is finite iff the number M of indices i for which $a_i(\mathbf{x}) \neq 0_{L^2(\mathbf{R}^2)}$ is finite, and $N = M$.*
4. *$F_\theta^{[n]}$ is an optimal n-approximation of F_θ with respect to both distances:*

$$D_n(F_\theta, F_\theta^{[n]}) = \Delta_n(F_\theta) = \left(\sum_{i=n+1}^{N} \sigma_i^2\right)^{1/2}$$

$$d_n(F_\theta, F_\theta^{[n]}) = \delta_n(F_\theta) = \sigma_{n+1}$$

5. *$D_n, \delta_n \to 0$ for $n \to N$.*
6. *$F_\theta^{[n]} = F_0^{[n]} \circ R_\theta$.*
7. *$\exists \theta_1, \ldots, \theta_n$ s.t. $F_\theta^{[n]} = \sum_{i=1}^{n} \alpha_i(\theta) F_{\theta_i}^{[n]}$. In fact this is true for all $\theta_1, \ldots, \theta_n$ but a set of measure zero.*

Comment.

1. The expression for the b_i is independent of F. Only σ_i and a_i depend on F. The b_i depend on the particular group of transformations (the rotations of the plane in this case) that is used to generate F_θ from F.
3. The 'if' part of statement 3 is the main theorem of [5]. The 'only if' part says that the functions described by Freeman and Adelson are *all* the steerable functions.

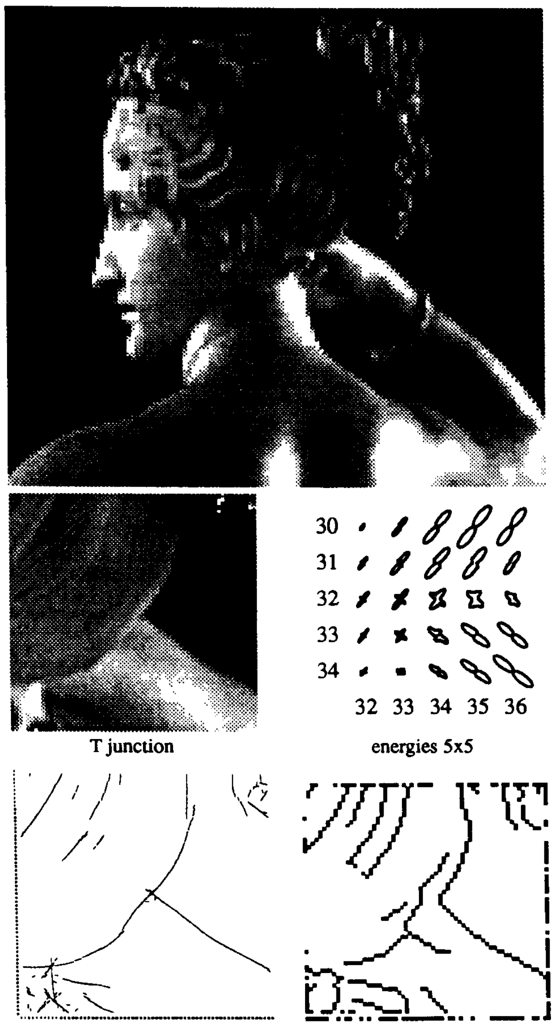


Figure 1: Example of the use of orientation-selective filtering on a continuum of orientations (see Perona and Malik [13, 14]). (Top) Original image. (Middle-left) Detail containing a T-junction (64x64 pixels from a region roughly at the centre of the image). (Middle-right) Modulus $R(x, y, \theta)$ of the output of a complex-valued filter (polar plot shown for 5x5 pixels in the region of the T-junction). The kernel of the filter (it is (gaus-3) in Fig. 2) is elongated to have hight orientation selectivity. Notice that in the region of the junction the response R has two local maxima in θ corresponding to the orientation of the edges. Searching for local maxima in (x,y) in a direction ortogonal to the maximizing θ's one can find the edges (Bottom left) with high accuracy (error around 1 degree in orientation and 0.1 pixels in position). (Bottom right) Comparison with the output of a Canny detector using the same kernel width (σ in pixel units).

4. For deciding at what point n to truncate the sum one plots the error δ_n or Δ_n v.s. n and looks for a knee in the curve, or for a value of N for which the error is less than some assigned value. See Fig. 2

6 This means that $F_\theta^{[n]}$ is steerable, i.e. its shape does not change with θ, modulo a rotation in the domain. Therefore $F_\theta^{[n]}$ may be seen as the best approximation to F_θ in the space of 'n-steerable' functions.

7.1. A set of size n of rotated copies of $F_0^{[n]}$ is enough for representing $F_\theta^{[n]}$. On the other hand this is less advantageous on the numerical point of view for two reasons: (1) The set F_{θ_i} is not orthonormal, so its numerical implementation is less efficient (it will require more significant bits in the calculations to obtain the same final precision). (2) The functions a_i are easier to approximate with sums of X-Y separable functions then the F_{θ_i} (see the experimental section 4.2, and Fig. 6).

7.2. The error $d(F_\theta, F_\theta^{[n]})$ of the n-approximation is constant with respect to θ since $F_\theta = F \circ R_\theta$ and $F_\theta^{[n]} = F_0^{[n]} \circ R_\theta$. There is *no anisotropy* even if $F^{[n]}$ is an approximation.

The proof of this theorem can be found in [12]. It is based on the singular value decomposition of $F_\theta(\mathbf{x})$, and the fact that the deformation involved (the rotations) is a group.

3 Deformable approximations

The success in finding the explicit finite-sum optimal approximation of the family F_θ in the previous section is not due to the fact that it was constructed using the group of rotations in the plane: the same results about optimal finite approximations are true for a much wider class of families F_θ (see [12] and [15](Chap.IV,Theorem 2.2)).

In some more general cases it is still possible to compute explicitly the finite-sum optimal approximation as in the case of rotations. This is reported synthetically in the next section.

3.1 Compact group representations

Call G a compact unitary group (eg. rotations in $\mathbf{R}^n$) whose elements act on $\mathbf{R}^n$ and consider a 'template' function $F \in L^2(\mathbf{R}^n)$. Define the 'deformations' of F by 'action' of elements T of G as: $F_T(\mathbf{x}) = F(T\mathbf{x})$.

Define a function H (cfr. (3)) of the elements T of the group G as:

$$H(T) = \int_{\mathbf{R}^n} F(T\mathbf{x})\overline{F(\mathbf{x})}d\mathbf{x} \tag{7}$$

Call D_{ij}^s the matrices of the s-th irreducible representation of the group G in $\mathbf{R}^n$ and call d_s the dimension of the same s-th representation. If G is a compact group we know (Peter-Weyl theorem, see e.g. Barut, Raczka [3]) that the $\sqrt{d_s}D_{ij}^s$ form an orthonormal basis of $L^2(G)$; therefore any $g \in L^2(G)$ can be written as a linear combination of such functions.

A generalization of the Fourier transform of a function of the group H is then defined as follows:

$$H(T) = \sum_s \sum_{ij} H_{ij}^s D_{ij}^s \tag{8}$$

$$H_{ij}^s = d_s \int_G H(T)\overline{D_{ij}^s(T)}dT \tag{9}$$

Define $b_{ij}^s = D_{ij}^s$ and $\lambda_{ij}^s = H_{ii}^s d_s$, and

$$a_{ij}^s(\mathbf{x}) = \int F_T(\mathbf{x})\overline{D_{ij}^s(T)}dT \tag{10}$$

A more general version of Theorem 1. is the following: **Theorem.** The optimal finite dimensional approximation of $F_T(x)$ is obtained truncating the series:

$$F_T(\mathbf{x}) = \sum_{sij} \lambda_{ii}^s b_{ij}^s(T) a_{ij}^s(\mathbf{x}) \tag{11}$$

3.2 An application: kernels for 3D edge detection and spatio-temporal filters

An application of the above formalism to a case of practical importance in signal processing and early vision is that of 3D edge detection and spatio-temporal filters.

Three dimensional images are common in medical applications. Tomographic data are collected as sets of 'slices', 2D images taken across the body and parallel to each other. It is very convenient to consider 'volumes' of slices as single 3D images. Another situation in which it is convenient to consider image data as functions of three variables is that of sequences of 2D images. In this case the third coordinate is time (in the notation that follows the time coordinate is lumped together with the spacial coordinates into the vector x). In both circumstances one wants to perform edge-detection, with the difference that now edges are 2D surfaces in 3D. Of course one may also want to detect 1D 'wires' and 0D points. The techniques used in 2D edge detection may be extended to 3D. In this case the kernels of the filters employed are directional derivatives of 3D Gaussian of appropriate variance, or similar functions [2, 8].

Notice that in 3D a 'template' function has to be rotated along three angles to obtain the continuum of filters necessary for boundary detection. 3D kernels may therefore be written as $F_{\phi,\theta,\psi}(\mathbf{x}) = F(\mathbf{x}) \circ R_{\phi,\theta,\psi}$ where R is a rigid rotation in $\mathbf{R}^3$. The group of rotations in $\mathbf{R}^3$, SO(3), has the following irreducible representation matrix expressions (see e.g. Barut, Raczka [3]):

$$D_{hk}^s(\theta, \phi, \psi) = e^{-ih\phi} d_{hk}^s(\theta) e^{-ik\psi} \tag{12}$$

$$d_{hk}^s(\theta) = \left(\frac{1+\cos\theta}{2}\right)^h P_{s-k}^{0,2h}(\cos\theta) \tag{13}$$

with $P_c^{a,b}$ the Jacobi polynomial (see e.g. Abramowitz, Stegun [1]). The expression for the finite-dimensional approximant may then be computed as:

$$F_{\phi,\theta,\psi}^{[l,m,n]}(\mathbf{x}) = \sum_{s,h,k=0,0,0}^{l,m,n} D_{hk}^s(\theta,\phi,\psi) G_{hk}^s(\mathbf{x}) \tag{14}$$

$$G_{hk}^s(\mathbf{x}) = \int_{SO(3)} F_{\phi,\theta,\psi}(\mathbf{x}) \overline{D_{hk}^s(\theta,\phi,\psi)} d\theta d\phi d\psi \tag{15}$$

4 Practical issues and experimental results

The formalism described in the previous sections may be applied to the problem of generating convolution kernels for an edge-detector. In this section such an application is described in detail and the the practical issues involved are discussed. The Gaussian-derivative kernels used by [13, 14] have been chosen for this example, very similar kernels have been used by [11, 10]. The template functions F are complex kernels. The real part of the kernels is a Gaussian $G(\mathbf{x}, \sigma_x, \sigma_y) = \exp -((x/\sigma_x)^2 + (y/\sigma_y)^2)$ differenciated twice along the Y axis. The imaginary part is the Hilbert transform of the real part taken along the Y axis (see Figure 5 (Top)).

Two families of deformations are demonstrated experimentally in this section: (a) pure rotation, (b) scaling and rotation. The first case is simple since the group of rotations is compact, hence the SVD (a_i,b_i,σ_i) may be computed exactly as described in section 2. One more twist is added: the eigenfunctions a_i may in turn be decomposed as sums of a small number of X-Y-separable functions making the implementation of the filters considerably faster. The case of rotations and scalings is more difficult: the group of scalings is not compact (scalings are defined from 0 to ∞), therefore the theory developed above may not be applied directly; a method to circumvent this problem is proposed and demonstrated.

4.1 Rotations

The calculations proceeded as indicated in section 2 (see [12] for some extra figures). For convenience they are summarized in a recipe:

1. Choose the 'template' kernel of which one wants rotated versions.
2. Compute the function $h(\theta)$ using its definition (Eq. (3)).
3. Compute the Fourier transform $\hat{h}$ of h. The coefficients are non-negative. Order them by decreasing magnitude and call their square roots σ_i and the corresponding frequencies ν_i.
4. The functions $b_i(\theta)$ are defined by Eq. (5) and the ν_i calculated at the previous step.
5. Compute the functions a_i using Eq. (6). The first nine are shown in Fig. 2.
6. Compute the error plots $\delta(n)$ and $\Delta(n)$ from the statement of the first theorem (see Fig. 5). Choose a maximum tolerable error and derive n.
7. The n-approximation of $F_\theta(\mathrm{x})$ can now be calculated using Eq. (2).

The numerical implementation of the formulae of section 2 and section 3.1 is straightforward. In the implementation used to produce the figures and the data reported in this paper the kernels F_θ were defined on a 128x128 array of single-precision floating-point numbers. The Y-axis variance was $\sigma_y = 8$ pixels, and the X-axis variance was $\sigma_x = k\sigma_y$ with $k = 1, 2, 3$. Calculations may be reduced by a factor of two if the hermitian symmetry in these kernels is exploited (the number of components can also be halved; the experimental data given below and in the figures are calculated this way). The set of all angles was discretized in 128 samples.

The coefficients σ_i turned out to be converging to zero exponentially fast, therefore the same was true for both errors. The kernel reconstructed using 9 components is shown at four different angles in Fig. 3. The reconstruction may be computed at any angle θ in a continuum.

In Fig. (4) (Bottom) the approximation is shown for $n = 4, 9, 15$. Notice that the 'orientation selectivity' of the filter increases with the number of components. The number n of singular components required to reconstruct the $\sigma_x : \sigma_y = 1 : 1$, and $\sigma_x : \sigma_y = 1 : 2$ families is smaller as indicated by the plots and in the caption of Fig. 5.

4.2 X-Y separability

Whenever a function F is to be used as a kernel for a 2D convolution it is of practical importance to know wether the function is X-Y-separable, i.e. wether there are two functions f^x and f^y such that $F(x, y) = f^x(x)f^y(y)$. If this is true the 2D convolution can be implemented cheaply as a sequence of two 1D convolutions.

Even when the kernel F is not X-Y-separable it may be the sum of a small number of separable kernels: $F(x, y) = \sum_i f_i^x(x)f_i^y(y)$. One may notice the analogy of this decomposition with the one expressed in Eq. (1). The singular value decomposition (SVD) may be used to calculate the optimal n-component approximation for each one of the a_i. If the SVD of a_i is indicated as: $a_i(x, y) = \sum_{h=1}^{n_i} \rho_{ih} a_{ih}^x(x) a_{ih}^y(y)$ then the decomposition of $F_\theta(x, y)$ becomes:

$$F_\theta(x, y) = \sum_{i=1}^{N} \sigma_i b_i(\theta) \sum_{h=1}^{n_i} \rho_{ih} a_{ih}^x(x) a_{ih}^y(y) \qquad (16)$$

The SVD of a kernel defined on a rectangular array can be computed using any one of the common numerical libraries [16]. Wether few or many components will be needed for obtaining a good approximation is again an empirical issue and will depend on the kernel in question. The decomposition of the singular functions a_i associated to the Gaussian-derivative functions used for these simulations is particularly advantageous; the approximation error typically shows a steep

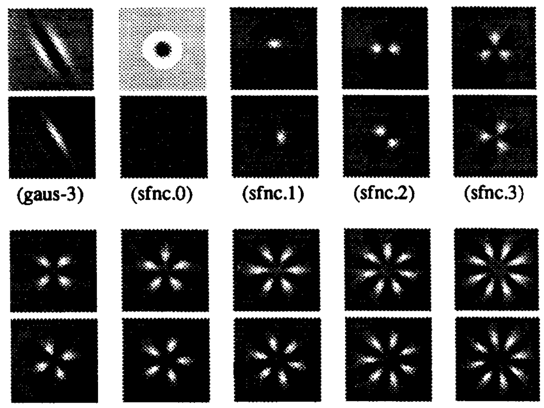


Figure 2: The decomposition (a_i, b_i, σ_i) of a complex kernel used in edge detection [14]. The template function (gaus-3) is shown rotated by 120°. Its real part (above) is the second derivative along the vertical (Y) axis of a Gussian with $\sigma_x : \sigma_y$ ratio of 1:3. The imaginary part (below) is obtained taking the Hilbert transform of the real part along the Y axis. The functions a_i (sfnc.i) are shown for $i = 0 \ldots 8$. Again the real part is above; the imaginary part below. The functions $b_i(\theta)$ are complex exponentials (see text) with associated frequencies $\nu_i = i$. The singular values σ_i decay exponentially: $\sigma_{i+1} \approx 0.75\sigma_i$.

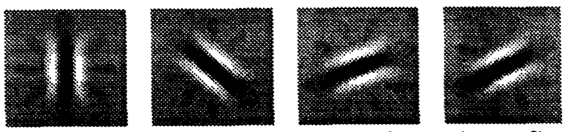


Figure 3: Functions reconstructed at angles 90° (rec-0.9), 140° (rec-50.9), 21°(rec-111.9), 27° (rec-117.9) using the $n = 9$ components of Fig. 2. The reconstruction error is 13% at all angles (see below). The real parts are shown.

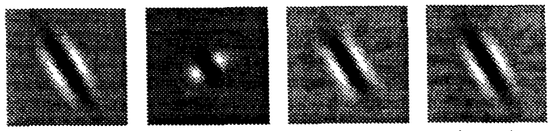
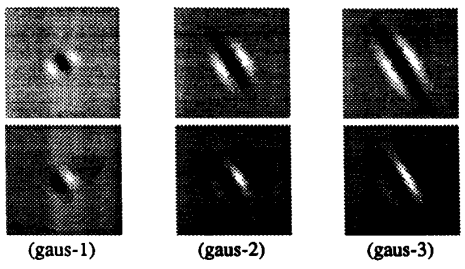


Figure 4: (Left to right) Original kernel (gaus-3) as in Fig. 2. Reconstruction of the kernel with 4 components (rec.3), with 9 components (rec.8) and with 15 components (rec.14). The optimal reconstruction error Δ_n for $n = 4, 9, 15$ calculated from the error plots of Fig. 5 is respectively 52%, 13.9% and 2.2%. The reconstruction error ||gaus-3 - rec.i|| / ||gaus-3|| measured on the reconstruction is respectively 50.2%, 13.4% and 2.2%. The error decreases exponentially, approximately 25-30% for each component added. The figures are given for the whole kernel, the real parts are shown.



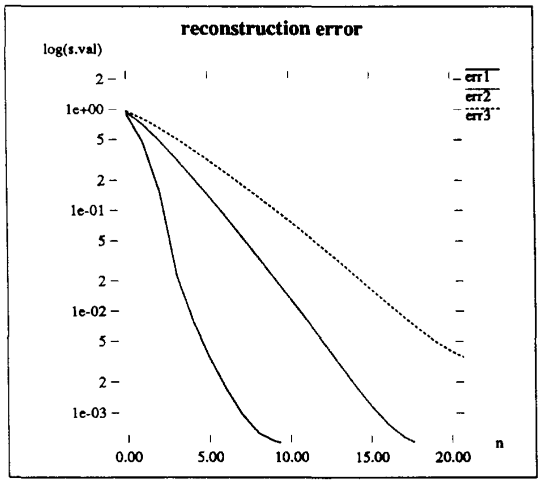
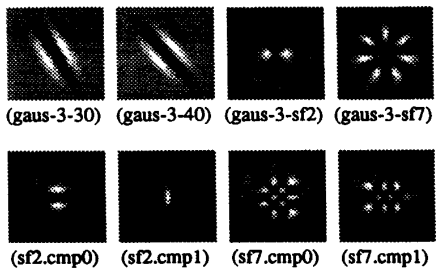


Figure 5: Comparison of the error plots for three aspect ratios. (Top) The three kernels shown at an angle of 120°; the real parts (above) being second derivative of Gaussians along the X axis, and the imaginary parts (below) being the X-axis Hilbert transforms of the real parts. The rations $\sigma_x : \sigma_y$ are respectively $1:1, 1:2, 1:3$. (Bottom) Plots of the log. of the reconstruction errors. For 10% reconstruction error 3, 6, 10 components are needed. For 5% reconstruction error 3, 7, 12 components are needed. Notice that for these Gaussian-derivative functions the reconstruction error decreases exponentially with the number of components employed: $\Delta_n \approx \exp(-\tau n)$ with $\tau \approx 1.7, 5.2, 8.2$.



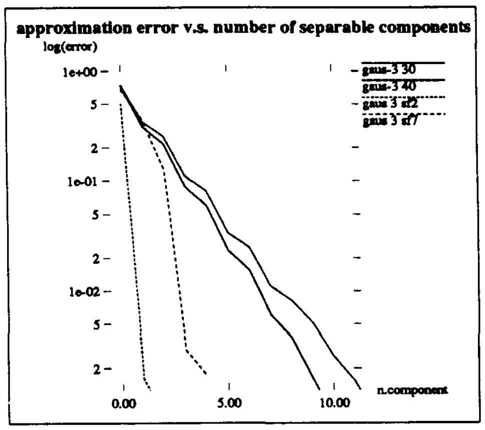


Figure 6: Comparison of the 'separability' of two rotated copies of the template function and two elements of the decomposition. The template function is (gaus-3) as in Fig. 2. The real parts are shown. (Top) The rotation angles are 120° (gaus-3-30) and 130° (gaus-3-40). The singular functions are a_3 (gaus-3-sf2) and a_8 (gaus-3-sf7) (cfr. Fig. 2). (Middle) The first two separable components of a_3 and a_8. (Bottom) The number of X-Y separable components necessary to approximate these functions within 1% error is: 7 and 8 for the rotated copies of the template function, and 1 and 3 for the singular functions. The number of components necessary to approximate a_i to less than 1% error is approximately $1 + i/4$, so that the total number of 1-D convolutions required to implement the n-approximation is $n + n^2/4$.

drop after a few components are added (see lower curves in Fig. 6 (Bottom) where the log of the error is plotted against the number of X-Y-separable components). All the a_i of Fig. 2 have been decomposed in sums of X-Y-separable kernels. The number of components needed for approximating each with 1% accuracy or better is indicated in the caption of Fig. 6.

It is important to notice that rotated versions of the original template functions F cannot be represented by sums of X-Y-separable functions with the same parsimony (see again Fig. 6 (Bottom) upper curves). This is one more reason to represent $F_\theta^{[n]}$ as a sum of orthonormal singular functions, rather than as as sum of rotated copies of the template function (Theorem 1, statement 7.), as discussed at the end of Sec. 2.

4.3 Rotation and scaling

Most of filter-based early vision and signal processing algorithms analyze the image at multiple scales of resolution. Typically only a discrete and small set of scales is employed due to the computational costs involved with filtering and storing images, although most of the algorithms are defined on, and would take advantage of, the availability of a continuum of scales. The problem of multi-scale filtering is

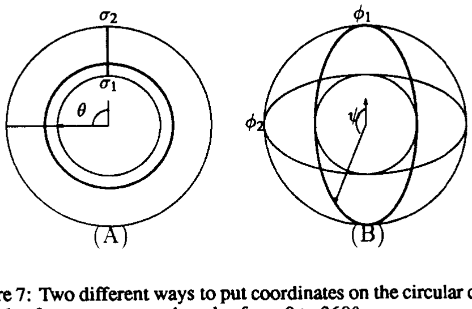


Figure 7: Two different ways to put coordinates on the circular crown of scales from σ_1 to σ_2 and angles from 0 to 360°.

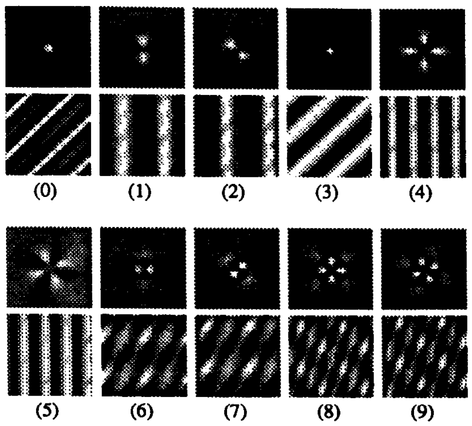


Figure 8: The first 10 terms of the decomposition of a rotation-scaling family. Real part shown. a_i above, b_i below.

somewhat analogue to the multi-orientation filtering problem that has been analyzed so far: given a template function $F(\mathbf{x})$ and defined $F_\sigma(\mathbf{x})$ as $F_\sigma(\mathbf{x}) = \sigma^{1/2}F(\sigma\mathbf{x})$, $\sigma \in (0,\infty)$ one would like to be able to write F_σ as a (small) linear combination:

$$F_\sigma(\mathbf{x}) = \sum_i s_i(\sigma)d_i(\mathbf{x}) \tag{17}$$

It is possible to convince oneself that the most direct approaches to the problem present considerable difficulties. A discussion of this point may be found in [12].

A different approach is represented in Fig. (7.B). The idea is to couple the scaling and rotation problem and to reparametrize the circular crown of rotations and scaling of interest. Instead of using the 'natural' (scale,angle)=(σ,θ) coordinates (Fig. (7.A)), one can use two coordinates (ϕ,ψ), the first of which determines the orientation of an ellypse within the crown, the second of which determines the position within the ellypse. In this parametrization the coordinate axis are tangent instead of normal to the boundaries. Within each ellyptical trajectory, the maximal scale and minimal scale functions are orthogonal and no discontinuities are present.

This scheme has been tested numerically. Approximately 40 singular functions are necessary for the decomposition of the Gaussian 2nd derivative kernel with ratios $\sigma_x : \sigma_y = 1:2$ (see Fig. 5 (Top)-center), scales in a ratio $\sigma_2 : \sigma_1 = 4$, and error around 10%. The real part of the first few singular functions is shown in Fig. 6. The functions come in couples: the singular functions $a_i(x,y)$ on above, and the $b_i(\phi,\psi)$ below.

5 Conclusions

A technique has been presented for implementing families of deformable kernels for early vision applications. The technique generates the optimal discrete approximation to a given family when the deformations involved form a compact group as in the case of rotations. It can be extended to cases where the deformations do not belong to a compact group as for scaling in the plane.

Unlike common techniques used in early vision where the set of orientations is discretized, here the kernel and the response of the corresponding filter may be computed in a continuum for *any* value of the deformation parameter, with no anisotropies. The approximation error is constant with respect to the deformation parameter and is computable a priori. This allows one to recover edges with great spatial and angular accuracy.

Experimentation with elongated kernels used for edge detection shows that the complexity of this optimal implementation is comparable to that of an implementation in which the orientations are discretized. It is cheaper if an implementation with X-Y-separable kernels is desired.

6 Acknowledgements

I am grateful to Massimo Porrati, Alberto Grunbaum, David Donoho, Charles Desoer and Richard Murray for helpful conversations. The simulations in this work have been carried out using Paul Kube's "viz" image-manipulation package, Paul also gave al lot of assistance. Part of this work was carried out at the International Computer Science Institute at Berkeley. This work was partially sponsored by U.S. Army Research Office grant number DAAL 03-86-K-0171 (Center for Intelligent Control Systems).

References

[1] Milton Abramowitz and Irene Stegun, editors. *Handbook of mathematical functions*. Dover, 1965.

[2] Edward Adelson and James Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[3] Asim Barut and Ryszard Raczka. *Theory of group representations and applications*. World Scientific, 1986.

[4] John Canny. A computational approach to edge detection. *IEEE trans. PAMI*, 8:679–698, 1986.

[5] William Freeman and Edward Adelson. Steerable filters for image analysis. Technical Report 126, MIT, Media Laboratory, 1990.

[6] William T. Freeman and Edward H. Adelson. Steerable filters for early vision, image analysis and wavelet decomposition. In *Third International Conference on Computer Vision*, pages 406–415. IEEE Computer Society, 1990.

[7] Goesta H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8:155–173, 1978.

[8] David Heeger. Optical flow from spatiotemporal filters. In *Proceedings of the First International Conference on Computer Vision*, pages 181–190, 1987.

[9] M.C. Morrone and D.C. Burr. Feature detection in human vision: a phase dependent energy model. *Proc. R. Soc. Lond. B*, 235:221–245, 1988.

[10] M.C. Morrone and R.A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, 1987.

[11] Pietro Perona. Finite representation of deformable functions. Technical Report 90-034, International Computer Science Institute, 1947 Center st., Berkeley CA 94704, 1990.

[12] Pietro Perona and Jitendra Malik. Detecting and localizing edges composed of steps, peaks and roofs. Technical Report UCB/CSD 90/590, Computer Science Division (EECS), U.C.Berkeley, 1990.

[13] Pietro Perona and Jitendra Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Third International Conference on Computer Vision*, pages 52–57. IEEE Computer Society, Osaka, 1990.

[14] Allan Pinkus. *n-Widths in Approximation Theory*. Springer Verlag, 1985.

[15] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.