October 1991 LIDS-P-2039

Deformable kernels for early vision

Pietro Perona

Università di Padova
Dipartimento di Elettronica ed Informatica
and
California Institute of Technology 116-81
Engineering and Applied Science
Pasadena, CA 91125
perona@verona.caltech.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 based on singular value decomposition and may be applied to generating filters in arbitrary dimensions. Experimental results are presented that demonstrate the applicability of the technique to generating multi-orientation multi-scale 2D edge-detection kernels. The implementation issues are also discussed.

This research was supported by the Army Research Office under grant DAAL-86-K-0171 (Center for Intelligent Control Systems).

1 Introduction

Points, lines, edges, textures, motions are present in almost all images of everyday's world. These elementary visual structures often encode a great proportion of the information contained in the image, moreover they can be characterized using a small set of parameters that are locally defined: position, orientation, characteristic size or scale, phase, curvature, velocity. It is threrefore resonable to start visual computations with measurements of these parameters. The earliest stage of visual processing, common for all the classical early vision modules, could consist of a collection of operators that calculate one or more dominant orientations, curvatures, scales, velocities at each point of the image or, alternatively, assign an 'energy', or 'probability', value to points of a position-orientation-phase-scale-etc. space. Ridges and local maxima of this energy would mark special interest loci such as edges and junctions. The idea that biological visual systems might analyze images along dimensions such as orientation and scale dates back to work by Hubel and Wiesel [21, 20] in the 1960's. In the computational vision literature the idea of analyzing images along multiple orientations appears at the beginning of the seventies with the Binford-Horn linefinder [19, 4] and later work by Granlund [16].

A computational framework that may be used to performs this proto-visual analysis is the convolution of the image with kernels of various shapes, orientations, phases, elongation, scale. This approach is attractive because it is simple to describe, implement and analyze. It has been proposed and demonstrated for a variety of early vision tasks [28, 27, 6, 3, 7, 18, 44, 34, 32, 35, 12, 31, 5, 45, 25, 26, 15, 39, 2]. Various 'general' computational justifications have been proposed for basing visual processing on the output of a rich set of linar filters: (a) Koenderink has argued that a structure of this type is an adequate substrate for local geometrical computations [29] on the image brightness, (b) Adelson and Bergen [2] have derived it from the 'first principle' that the visual system computes derivatives of the image along the dimensions of wavelength, parallax, position, time, (c) a third point of view is the one of 'matched filtering': where the kernels are synthesized to match the visual events that one looks for.

The kernels that have been proposed in the computational literature have typically been chosen according to one or more of three classes of criteria: (a) 'generic optimality' (e.g. optimal sampling of space-frequency space), (b) 'task optimality' (e.g. signal to noise ratio, localization of edges) (c) emulation of biological mechanisms. While there is no general consensus in the literature on precise kernel shapes, there is convergence on kernels roughly shaped like either Gabor functions, or derivatives or differences of either round or elongated Gaussian functions – all these functions have the advantage that they can be specified and computed easily. A good rule of the thumb in the choice of kernels for early vision tasks is that they should have good localization in space and frequency, and should be roughly tuned to the visual events that one wants to analyze.

Since points, edges, lines, textures, motions can exist at all possible positions, orientations, scales of resolution, curvatures one would like to be able to use families of filters that are tuned to all orientations, scales and positions. Therefore once a particular convolution kernel has been chosen one would like to convolve the image with deformations (rotations, scalings, stretchings, bendings etc.) of this 'template'. In reality one can afford only a finite (and small) number of filtering operations, hence the common practice of 'sampling' the set of orientations, scales, positions, curvatures, phases [1]. This operation has the strong drawback of introducing anisotropies

[1] Motion flow computation using spatiotemporal filters has been proposed by Adelson and Bergen [3] as a model of human vision and has been demonstrated by Heeger [18] (his implementation had 12 discrete spatio-temporal orientations and 3 scales of resolution). Work on texture with multiple-resolution multiple-orientation kernels is due to Knuttson and Granlund [28] (4 scales, 4 orientations, 2 phases), Turner [44] (4 scales, 4 orientations, 2 phases), Fogel and Sagi [12] (4 scales, 4 orientations, 2 phases), Malik and Perona [31] (11 scales, 6 orientations, 1 phase)

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 in terms of the kernels (for the sake of simplicity the case of rotations in the plane is discussed here): Given a kernel $F : \mathbb{R}^2 \to \mathbb{C}^1$, define the family of 'rotated' copies of F as: $F_\theta = F \circ R_\theta$, $\theta \in \mathbb{S}^1$, where $\mathbb{S}^1$ is the circle and R_θ is a rotation. Sometimes it is possible to express F_θ as

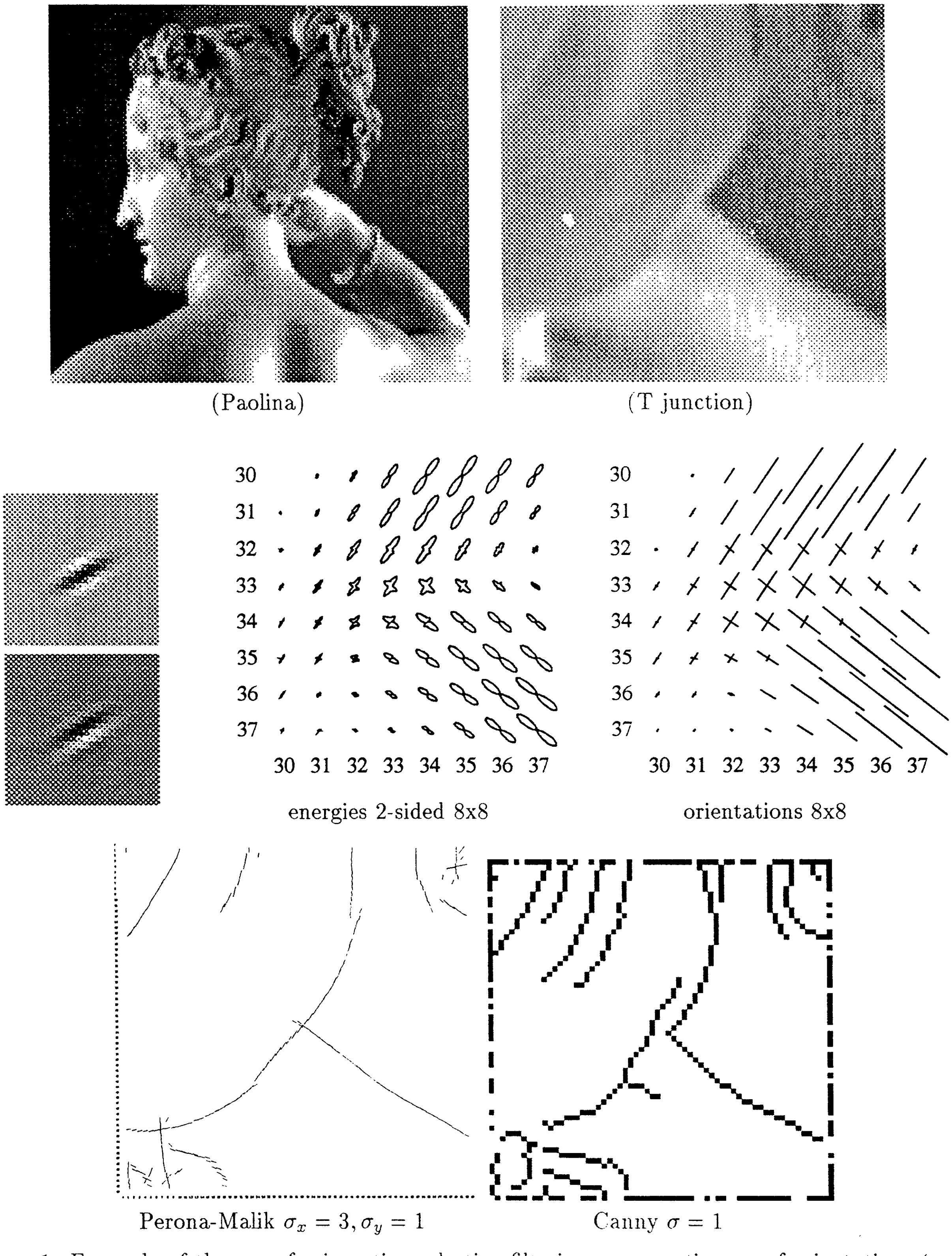
$$F_\theta(\mathbf{x}) = \sum_{i=1}^{n} \alpha(\theta)_i G_i(\mathbf{x}) \qquad \forall \theta \in \mathbb{S}^1, \forall \mathbf{x} \in \mathbb{R}^2 \tag{1}$$

a *finite* linear combination of functions $G_i : \mathbb{R}^2 \to \mathbb{C}^1$. It must be noted that, at least for positions and phases, the mechanism for realizing this in a systematic way is well understood: in the case of positions the sampling theorem gives conditions and an interpolation technique for calculating the value of the filtered image at any point in a continuum; in the case of phases a pair of filters in quadrature can be used for calculating the response at any phase [3, 33]. Rotation, scalings and other deformations are less well understood.

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 ($\sigma_x = \sigma_y$) Gaussian may be obtained by linear combination of the X- and Y-derivatives of the same. The common implementations of the Canny edge detector [7] are based on this principle. Unfortunately the kernel obtained this way 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 argued [15, 14] that it would be desirable to construct orientation-selective kernels that can be exactly rotated by interpolation (they call this property "steerability" and the term will be used in this paper) and have shown that higher order derivatives of round Gaussians, indeed all polynomials multiplied by a radially symmetric function are steerable (they have a more general result - see comments to Theorem 1). For high polynomial orders these functions may be designed to have higher orientation selectivity and can be used for contour detection and signal processing [15]. However, one must be aware of the fact that for most kernels 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 [38, 39] (see Fig. 1 top right) 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 is steerable, i.e. can be expressed as a finite sum of n terms as in (1). Freeman and Adelson propose to approximate the kernel with an adequately high order polynomial multiplied by a radially symmetric function (which they show is steerable). However, this method does not guarantee a parsimonious approximation: given a tolerable amount

and Bovik et al. [5] (n scales, m orientations, l phases). Work on stereo by Kass [27] (12 filters, scales, orientations and phases unspecified) and Jones and Malik [25, 26] (6 scales, 2-6 orientations, 2 phases). Work on curved line grouping by Parent and Zucker [35] (1 scale, 8 orientations, 1phase) and Malik and Gigus [30] (9 curvatures, 1 scale, 18 orientations, 2 phases). Work on brightness edge detection by Binford and Horn [19, 4] (24 orientations), Canny [7] (1-2 scales, ∞-6 orientations, 1 phase), Morrone,Owens and Burr [34, 32] (1-3 scales, 2-4 orientations, ∞ phases), unpublished work on edge and illusory contour detection by Heitger, Rosenthaler, Kübler and von der Heydt (6 orientations, 1 scale, 2 phases). Image compression by Zhong and Mallat [45] (4 scales, 2 orientations, 1 pahse).



(Paolina)

(T junction)

energies 2-sided 8x8

orientations 8x8

Perona-Malik $\sigma_x = 3, \sigma_y = 1$

Canny $\sigma = 1$

Figure 1: Example of the use of orientation-selective filtering on a continuum of orientations (see Perona and Malik [38, 39]). (Top left) Original image. (Top right) A T-junction (64x64 pixel detail from a region roughly at the centre of the original image). (Middle left) The kernel of the filter (it is (gaus-3) in Fig. 3) is elongated to have high orientation selectivity. (Middle-centre) Modulus $R(x, y, \theta)$ of the output of the complex-valued filter (polar plot shown for 8x8 pixels in the region of the T-junction). (Middle-right) The local maxima of $|R(x, y, \theta)|$ with respect to θ. Notice that in the region of the junction one finds 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).

of error one would like to find an approximating $G_\theta^{[n]}$ that has minimum number n of components. 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 vision and signal processing literature so far.

This paper is organized as follows: the special case of the rotations (Eq. (1)) is explored and solved in section 2 and appendix A.1. In section 3 a few results from functional analysis are recalled to extend the approximation technique to all 'compact' deformations. In section 4 an application of the approximation technique to generating steerable filters for edge detection is described. In section 5 it is shown how to generate a steerable and scalable family. Experimental results and implementation issues are presented and discussed for the schema presented in sections 5 and 4.

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 $\mathbb{R}^2 \times \mathbb{S}^1$ 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 $\mathbb{R}^2$. 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.

$$\begin{aligned}
D_n(F_\theta, G_\theta^{[n]}) &= \|F_\theta - G_\theta^{[n]}\|_{\mathbb{R}^2\times\mathbb{S}^1} \\
\Delta_n(F_\theta) &= \inf_{G_\theta^{[n]}} D_n(F_\theta, G_\theta^{[n]}) \\
d_n(F_\theta, G_\theta^{[n]}) &= \sup_{\|f\|=1} \|\langle F_\theta - G_\theta^{[n]}, f\rangle_{\mathbb{R}^2}\|_{\mathbb{S}^1} \\
\delta_n(F_\theta) &= \inf_{G_\theta^{[n]}} d_n(F_\theta, G_\theta^{[n]})
\end{aligned}$$

Consider the approximation to F_θ defined as follows:

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

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

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:

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

and let ν_i be the frequencies 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 $\hat{h}(\nu_i)$. Finally, define the quantities:

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

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

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

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 $F \in L^2(\mathbb{R}^2)$ 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, β_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 $a_i(\mathbf{x}) \neq 0_{L^2(\mathbb{R}^2)}$ is finite, and $N = M$.*
4. *$F_\theta^{[n]}$ is an optimal n-approximation of F_θ with respect to both distances:*

$$D_n(F_\theta, F_\theta^{[n]}) = \Delta_n(F_\theta) = \left(\sum_{i=n+1}^{N} \sigma_i^2 \right)^{1/2} \tag{7}$$

$$d_n(F_\theta, F_\theta^{[n]}) = \delta_n(F_\theta) = \sigma_{n+1} \tag{8}$$

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 equivalent to the main theorem of [13] – Freeman and Adelson show that all functions that one can write as a linear combination of functions like the a_i's (polar-separable with sinusoidal θ component) are steerable. The 'only if' part says that the functions that they described are *all* the steerable functions.

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 the smallest integer n for which the error is less than some assigned value. See Figs. 4, 6.

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 $F_\theta^{[n]}$ is the best approximation to F_θ in the space of 'n-steerable' functions ('best' is intended with respect to the L^2-induced distance).

7.1. A set of size n of rotated copies of $F_0^{[n]}$ is enough for representing $F_\theta^{[n]}$, so we may choose to use this different decomposition instead of 2. On the other hand this representation has some numerical disadvantages: (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. 8).

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 appendix A.1. It is based on the fact that the triples (σ_i,$a_i(\mathbf{x})$,$b_i(\theta)$) are the singular value decomposition (SVD) of a linear operator associated to $F_\theta(\mathbf{x})$. (From now on the abbreviation SVD will be used to indicate the decomposition of a kernel into such triples).

3 Deformable functions

The existence of the optimal finite-sum approximation of the kernel $F_\theta(\mathbf{x})$ as decribed in the previous section and Sec. A.1 is not peculiar to the case of rotations. This is true in more general circumstances: this section collects a few facts of functional analysis that show that one can compute finite optimal approximations to continuous families of kernels whenever certain 'compactness' conditions are met.

Consider a parametrized family of kernels $F(\mathbf{x};\theta)$ where $\mathbf{x} \in X$ now indicates a generic vector of variables in a set X and $\theta \in T$ a vector of parameters in a set T. (The notation is changed slightly from the previous section.) Consider the sets A and B of continuous functions from X and T to the complex numbers, call $a(\mathbf{x})$ and $b(\theta)$ the generic elements of these two sets. Proceed as at the beginning of sec. A.1 and consider the operator $L : A \longrightarrow B$ defined by F as $(La(\cdot))(\theta) = \langle F(\cdot;\theta), a(\cdot)\rangle_A$.

A first theorem says that if the kernel F has bounded norm then the associated operator L is compact (see [8] pag. 316):

Theorem 2 *Let X and T be locally compact Hausdorff spaces and $F \in L_2(X \times T)$. Then L is well defined and is a compact operator.*

Such a kernel is commonly called a *Hilbert-Schmidt kernel.*

A second result tells us that if a linear operator is compact, then it has a discrete spectrum (see [9] pag. 323):

Theorem 3 *Let L be a compact operator on (complex) normed spaces, then the spectrum S of L is at most denumerable.*

A third result says that if L is continuous and operates on Hilbert spaces then the compactness property transfers to the adjoint of L (see [9] pag. 329):

Theorem 4 *Let L be a compact operator on Hilbert spaces, then the adjoint L^* is compact.*

Trivially, the composition of two compact operators is compact, so the operators LL^* and L^*L are compact and will have a discrete spectrum as guaranteed by theorem 3. The singular value decomposition (SVD) for the operator L can therefore be computed as the collection of triples $(\sigma_i, a_i, b_i), i = 0, \ldots$ where the σ_i constitute the spectra of both LL^* and L^*L and the a_i and b_i are the corresponding eigenvectors.

The last result can now be enunciated (see [40] Chap.IV,Theorem 2.2):

Theorem 5 *Let $L : A \rightarrow B$ be a linear compact operator between two Hilbert spaces. Let a_i, b_i, σ_i be the singular value decomposition of L, where the σ_i are in decreasing order of magnitude. Then*

1. *An optimal n-dimensional approximation to L is $L_n = \sum_{i=1}^{n} \sigma_i a_i b_i$*
2. *The approximation error is $d_n(L) = \sigma_{n+1}$, $\Delta_n^2(L) = \sum_{i=n+1}^{N} \sigma_i^2$*

As a result we know that when our original template kernel $F(\mathbf{x})$ and the chosen family of deformations $R(\theta)$ define a Hilbert-Schmidt kernel $F(\mathbf{x};\theta) = (F \circ R(\theta))(\mathbf{x})$ then it is possible to compute a finite discrete approximation as for the case of 2D rotations.

Are the families of kernels $F(\mathbf{x};\theta)$ of interest in vision Hilbert-Schmidt kernels? In the cases of interest for vision applications the 'template' kernel $F(\mathbf{x})$ typically has a finite norm, i.e. it belongs to $L_2(X)$ (all kernels used in vision are bounded compact-support kernels such as Gaussian derivatives, Gabors etc.). However, this is not a sufficient condition for the family $F(\mathbf{x};\theta) = F \circ R(\theta)(\mathbf{x})$ obtained composing $F(\mathbf{x})$ with deformations $R(\theta)$ (rotations, scalings) to be a Hilbert-Schmidt kernel: the norm of $F(\mathbf{x};\theta)$ could be unbounded. A sufficient condition for the associated family $F(\mathbf{x};\theta)$ to be a Hilbert-Schmidt kernel is that the inverse of the Jacobian of the transformation R, $|JR|^{-1}$ belongs to $L_2(T)$. In fact the norm of $F(\cdot;\cdot)$ is bounded above by the product of the norm of F in X and the norm of $|JR|^{-1}$ in T:

$$\begin{aligned}
||F(\cdot;\cdot)||^2 &= \int_T \int_X |F(\mathbf{x};\theta)|^2 d\mathbf{x} d\theta \\
&= \int_T \int_X |(F \circ R(\theta))(\mathbf{x})|^2 d\mathbf{x} d\theta \\
&\leq \int_T \int_X |JR(\theta)|^{-1} |F(\mathbf{y})|^2 d\mathbf{y} d\theta \\
&= ||F(\cdot)||^2 || |JR(\cdot)|^{-1} ||^2
\end{aligned}$$

Which is bounded by hypothesis.

A typical condition in which this arises is when the transformation R is unitary, e.g. a rotation, translation, or an appropriately normalized scaling, and the set T is bounded. In that case the norm of $||JR||^{-1}$ is equal to the measure of T. The following sections in this paper will illustrate the power of these results by applying them to the decomposition of rotating 2D kernels (section 2), 2D kernels into sums of X-Y-separable kernels (section 4.2), rotating and scaled kernels (section 5).

A useful subclass of kernels F for which the finite orthonormal approximation can be in part explicitly computed is obtained by composing a template function with transformations T_θ belonging to a compact group. This situation arises in the case of n-dimensional rotations and is useful for edge detection in tomographic data and spatiotemporal filtering. It is discussed in [36, 37].

4 Steerable approximations: practical issues and experimental results

In this section the formalism described so far is applied to the problem of generating steerable and scalable approximations to convolution kernels for an edge-detector. The Gaussian-derivative

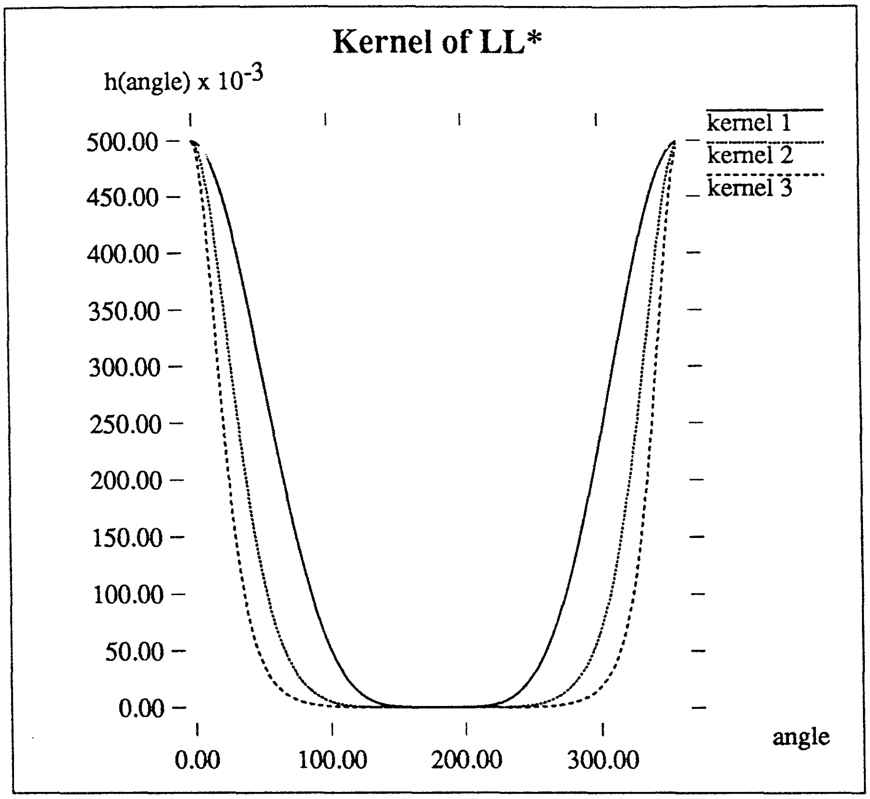


Figure 2: The kernel $h(\theta)$ (see eq. (3) and (23)). Angle θ in abscissa. The template function is as in Fig. 3 with $\sigma_x : \sigma_y$ ratios of 1:1, 1:2, 1:3. The orientation-selectivity of the filters with impulse response equal to the template functions can be estimated from these plots: the half-width of the peak is approximately 60^o, 35^o, 20^o.

kernels used by [38, 39] have been chosen for this example. These kernels have an elongated shape to achieve high orientation selectivity (equivalently, narrow orientation tuning). The real part of the kernels is a Gaussian $G(\mathrm{x}, \sigma_x, \sigma_y) = \exp -((x/\sigma_x)^2 + (y/\sigma_y)^2)$ differenciated twice along the Y axis. The imaginary part is the Hilbert transform of the real part taken along the Y axis (see Figures 3 and 6, top-left).

One more twist is added: the functions a_i in the decomposition may in turn be decomposed as sums of a small number of X-Y-separable functions making the implementation of the filters considerably faster.

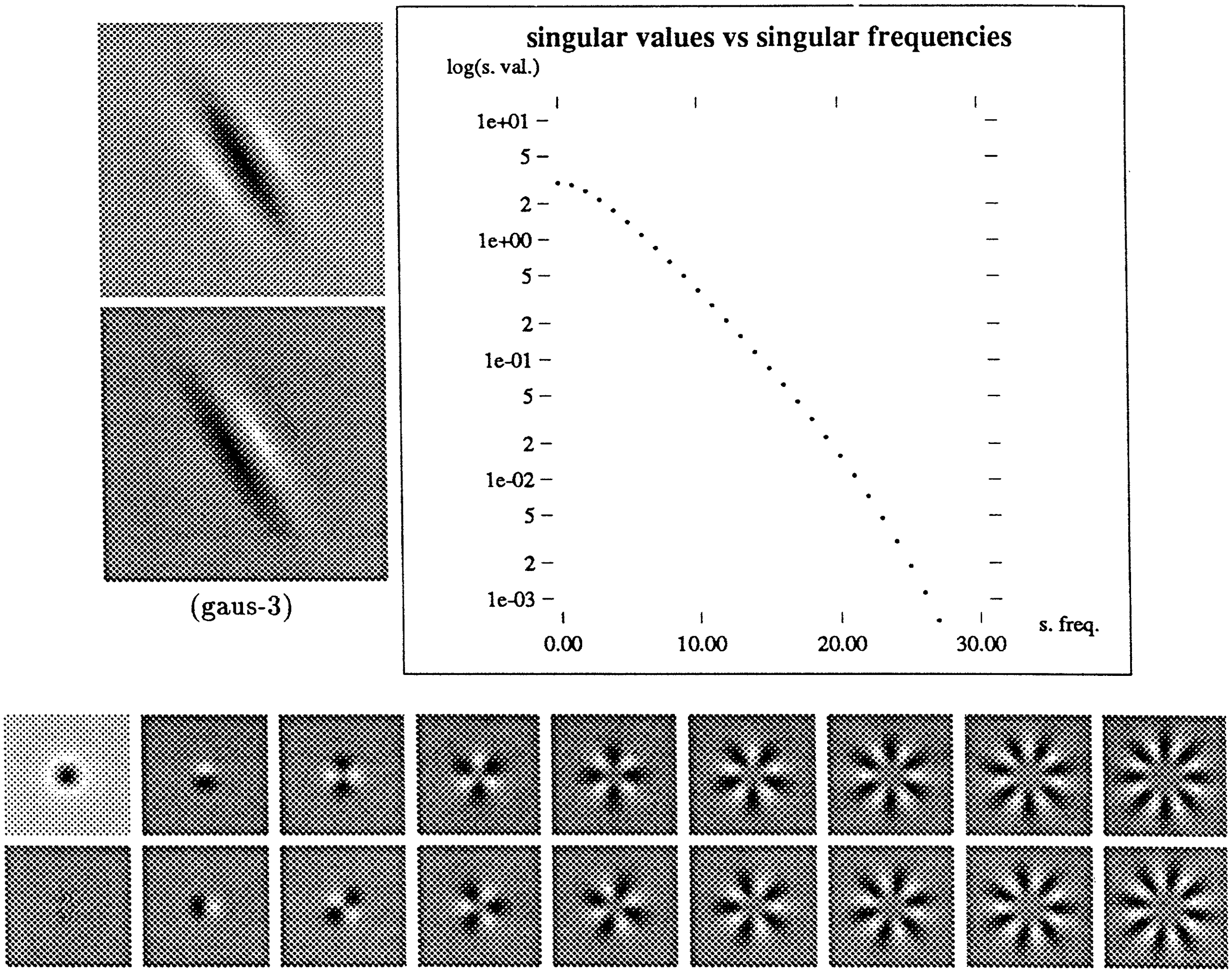
4.1 Rotations

In the case of rotations theorem 1 may be used directly to compute the decomposition. The calculations proceded as indicated in section 2. For convenience they are summarized in a recipe:

1. Select the 'template' kernel $F(\mathrm{x})$ of which one wants rotated versions $F_\theta(\mathrm{x}) = F \circ R_\theta(\mathrm{x}) = F(x\cos(\theta) + y\sin(\theta), -x\sin(\theta) + y\cos(\theta))$ and the maximum tolerable error η.

2. Compute numerically the function $h(\theta)$ using its definition (Eq. (3)). See Fig. 2.

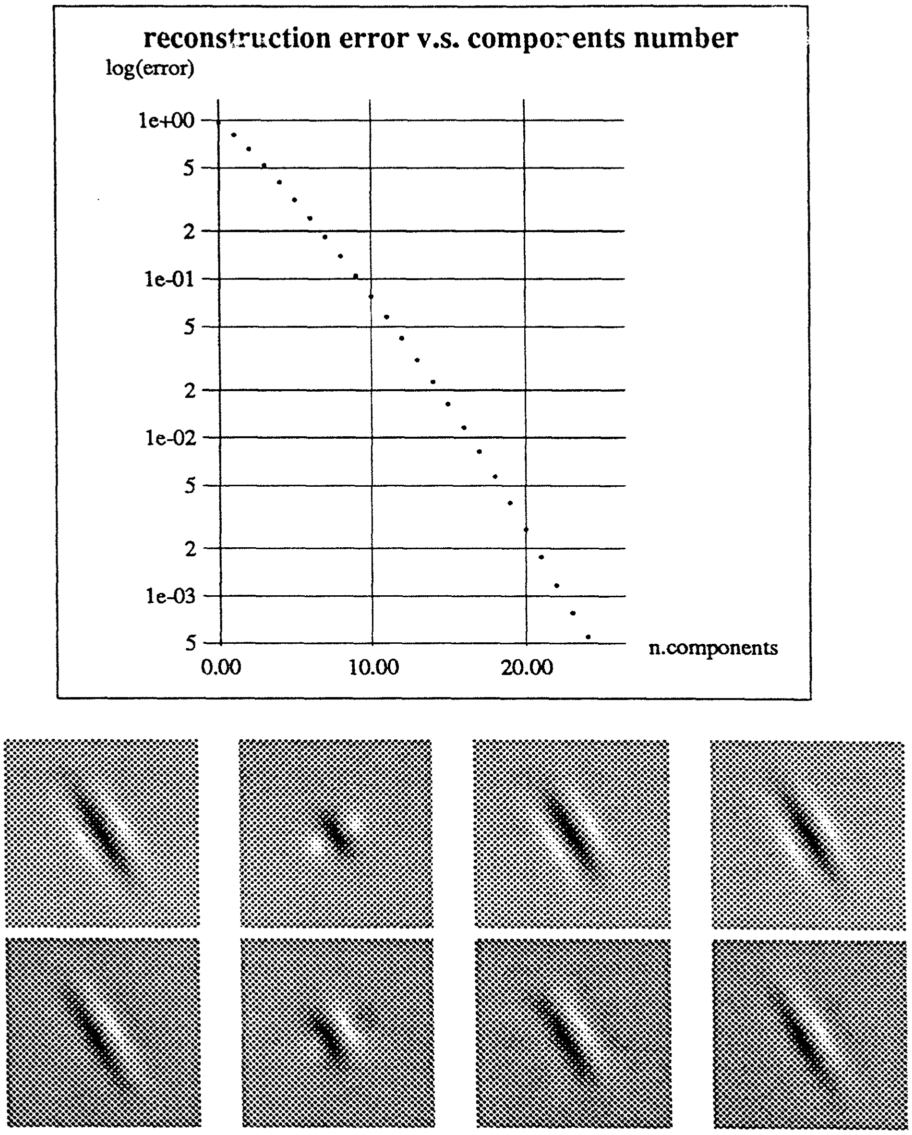
3. Compute the discrete Fourier transform $\hat{h}$ of h. Verify that the coefficients $\hat{h}_k$ are non-negative.



(gaus-3)

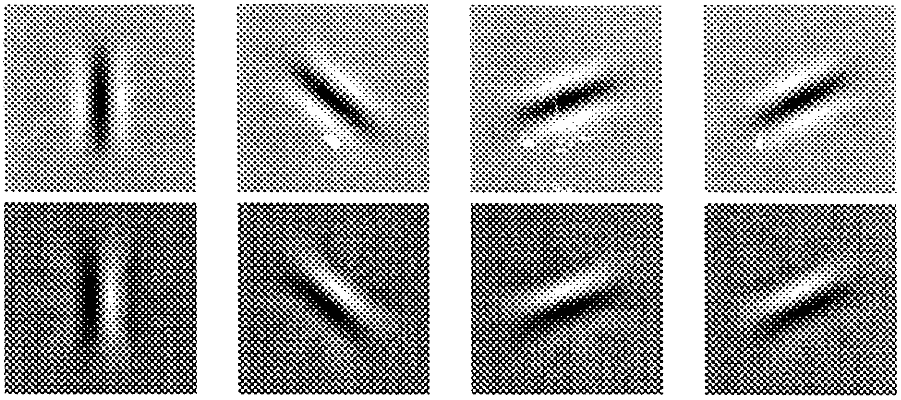
(sfnc.0) (sfnc.1) (sfnc.2) (sfnc.3) (sfnc.4) (sfnc.5) (sfnc.6) (sfnc.7) (sfnc.8)

Figure 3: The decomposition (a_i, b_i, σ_i) of a complex kernel used for brightness-edge detection [39]. (Top left) The template function (gaus-3) is shown rotated counterclockwise by 120^o. 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 the Hilbert transform of the real part along the Y axis. (Top-right) The first 28 σ_i (s.val) shown on a logarithmic scale plotted against the associated frequencies $\nu_i = i$ (s.freq). They decay exponentially: $\sigma_{i+1} \approx 0.75\sigma_i$. (Bottom) The functions a_i (sfnc.i) are shown for $i = 0 \ldots 8$. The real part is above; the imaginary part below. The functions $b_i(\theta)$ are complex exponentials (see text) with associated frequencies $\nu_i = i$.



(gaus-3) (rec.3) (rec.8) (rec.14)

Figure 4: (Bottom – left to right) Original kernel (gaus-3) as in Fig. 3. 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 plot (top) is respectively 52%, 13.9% and 2.2%. The reconstruction error ||gaus-3 - rec.i|| · $||\text{gaus-3}||^{-1}$ as measured on the reconstruction in the computer implementation is respectively 50.2%, 13.4% and 2.2%. The error decreases exponentially, approximately 25-30% for each component added. (See also the caption of Fig. 6).



(rec-0.9) (rec-50.9) (rec-111.9) (rec-117.9)

Figure 5: Functions reconstructed at angles 0^o, 50^o, 111^o, 117^o from the optimal decomposition shown in Fig. 3. Nine components were used. The reconstruction error is 13% at all angles (see caption of Fig. 4).

4. Order the $\hat{h}_k$ computed at the previous step by decreasing magnitude and call their square roots σ_i (see Fig. 3 (Top right)) and the corresponding frequencies ν_i (see Eq. (4)).
5. Define the functions $b_i(\theta)$ according to Eq. (5) and the ν_i calculated at the previous step.
6. Compute the error plots $\delta(n)$ and $\Delta(n)$ from eq. (7), (8) (see Fig. 6). Obtain the number n of components required for the approximation as the first integer where the error drops below the tolerable error η.
7. Compute the functions $a_i(\mathrm{x})$ using Eq. (6). (See Fig. 3).
8. The n-approximation of $F_\theta(\mathrm{x})$ can now be calculated using Eq. (2).

The numerical implementation of the formulae of section 2 is straightforward. In the implementation used to produce the figures and the data reported in this section the kernels F_θ were defined on a 128x128 array of single-precision floating-point numbers. The set of all angles was discretized in 128 samples. 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 were reduced by a factor of two exploiting the hermitian symmetry of these kernels; the number of components can also be halved – the experimental data given below and in the figures are calculated this way.

Notice first that the coefficients σ_i converge to zero exponentially fast; as a consequence the same is true for both errors. This is very important in practice since it implies that a very small number of coefficients is required. The kernel reconstructed using 9 components is shown at four different angles in Fig. 5. The reconstruction may be computed at any angle θ in a continuum.

In Fig. 4 (Bottom) the approximate reconstruction is shown for $n = 4, 9, 15$. Notice that the elongation and therefore the 'orientation selectivity' of the filter increases with the number of components. In Fig. 7 the modulus of the response of the complex filter to an edge is shown for two different kernels and increasing levels of approximation. 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. 6.

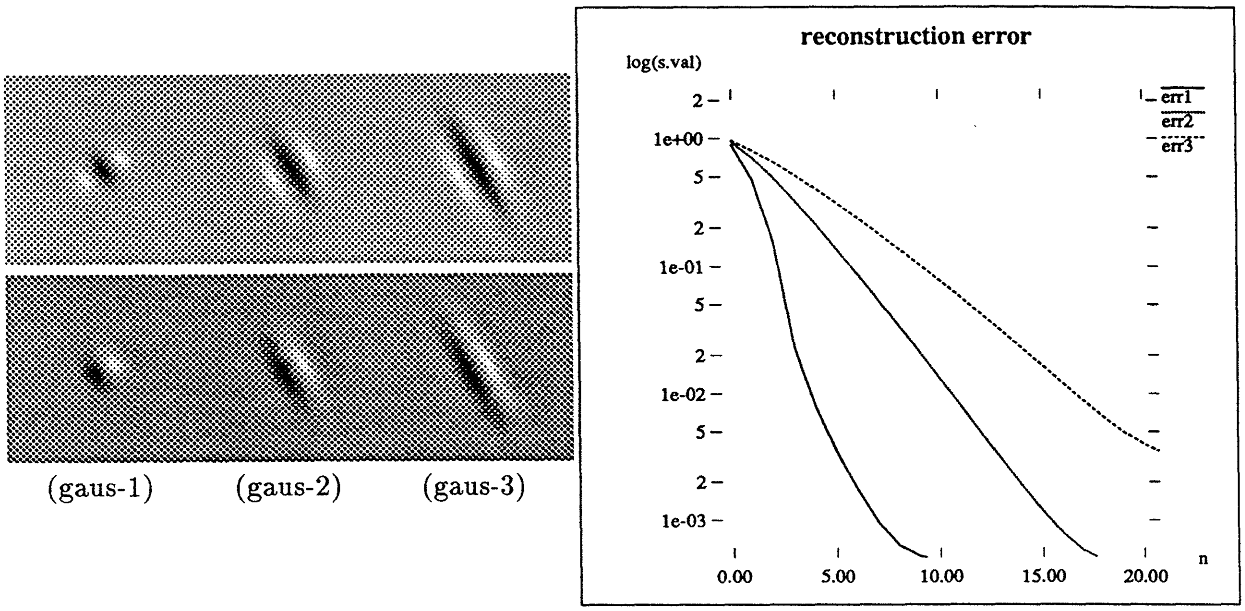
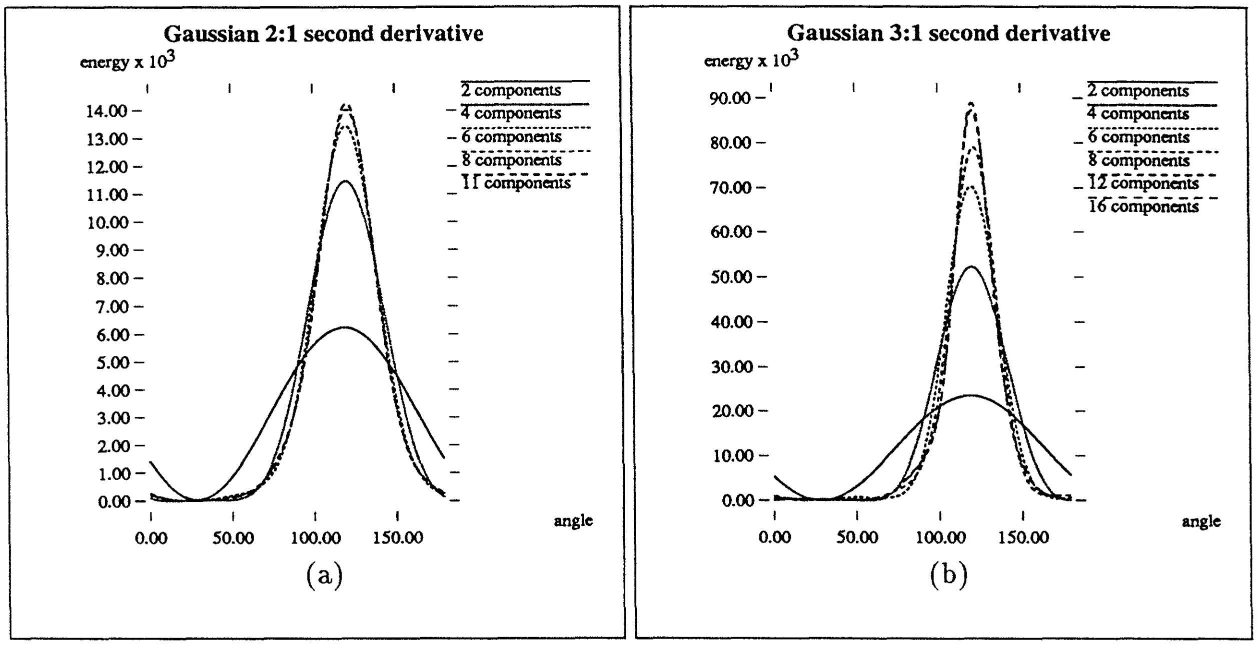


Figure 6: Comparison of the error plots for three kernels constructed differenciating Gaussians of different aspect ratios as in Fig. 3. (Left) The three kernels shown at an angle of 120°; the ratios $\sigma_x : \sigma_y$ are respectively $1:1$, $1:2$, $1:3$. (Right) The log of the reconstruction errors are plotted against the number of components employed. 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 roughly exponentially with respect to the number n of components employed: $\Delta_n \approx \exp(-\frac{n}{\tau})$ with $\tau \approx 1.7, 5.2, 8.2$.



(a)

(b)

Figure 7: Magnitude of the response vs. orientation angle of orientation-selective kernels. The image is an edge oriented at 120° (notice the corresponding peak in the filter responses). The kernels are as in Fig. 6: (gaus-2) for (a) and (gaus-3) for (b). The plots show the response of the filters for increasing approximation. The first approximation (2 components) gives a broadly tuned response, while the other approximations (4,6,8 ... components) have more or less the same orientation selectivity (half-width of the peak at half-height). The peak of the response sharpens and the precision of the approximation is increased (1 % error for the top curves) when more components are used .

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. if 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 [41]: $F(x,y) = \sum_i f_i^x(x)f_i^y(y)$. One may notice the analogy of this decomposition with the one expressed in Eq. (1): the function $F(x,y)$ may be thought of as a kernel defining an operator from a space A of functions $f^x(x)$ to a space B of functions $f^y(y)$. At a glance on can see that the kernels a_i of Fig. 3 are Hilbert-Schmidt: they have bounded norm. Therefore (see sec. 3) a discrete optimal decomposition and approximation are possible. Again the SVD machine may be applied to calculate the decomposition of each one of the a_i and its optimal n_i-component approximation. 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 approximate decomposition of $F_\theta(x,y)$ expressed in Eq. (2) and (20) becomes:

$$F_\theta(x,y) = \sum_{i=1}^{N} \sigma_i b_i(\theta) \sum_{h=1}^{n_i} \rho_{ih} a_{ih}^x(x) a_{ih}^y(y) \tag{9}$$

How is this done in practice? For all practical pourposes the kernels a_i are defined on a discrete lattice. The SVD of a kernel defined on a discrete $M \times N$ rectangular lattice may be computed numerically using any one of the common numerical libraries [10, 42] as if it was a $M \times N$ square matrix A of rank R. The typical notation for the matrix SVD is: $A = UWV^T$ where U and V are orthonormal $M \times R$ and $R \times N$ matrices and W is $R \times R$ band-diagonal with positive entries w_i of decreasing magnitude along the diagonal. If this is written as

$$A = \sum_{k=1}^{R} w_k U_k V_k^T \tag{10}$$

where U_k and V_k are columns of U and V the analogy with Eq. (2) becomes obvious. Notice that the (hidden in the vector notation) row index of U plays the role of the coordinate y and the row index of V plays the role of the coordinate x. Rewriting the above in continuous notation we obtain:

$$a(x,y) = \sum_{k=1}^{R} w_k u_k(y) v_k(x) \tag{11}$$

The first two terms of the separable decompositions are shown in Fig. (8) for the functions a_3 and a_8.

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 drop after a few components are added. This can be seen from he curves in Fig. 8 (Bottom-left) where the log of the error is plotted against the number of X-Y-separable components. All the a_i of Fig. 3 can be decomposed this way in sums of X-Y-separable kernels. The number of components needed for approximating each with 1% accuracy or better is indicated in the plots of Fig. 8 (Bottom-right) the real and imaginary parts have roughly the same separability. One can see that the number of components increases linearly with i. The caption of Fig. 8 gives precise figures for the Gaussian 3 : 1 case.

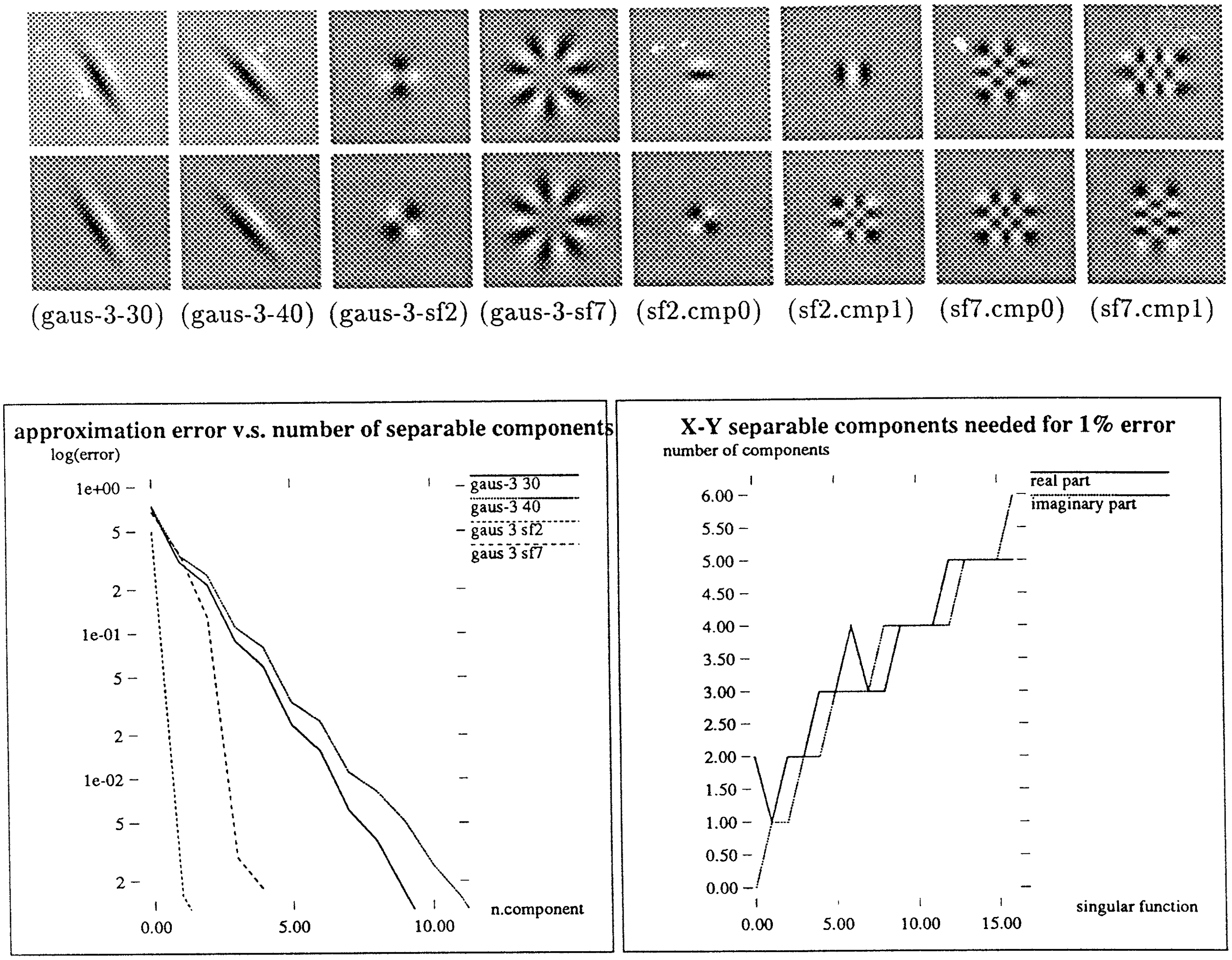


Figure 8: Comparison of the 'separability' of two rotated copies of the template function and two elements of the decomposition. The template function is (gaus-3) as in Fig. 3. (Top) The rotation angles are 120^o (gaus-3-30) and 130^o (gaus-3-40). The functions from the decomposition are a_3 (gaus-3-sf2) and a_8 (gaus-3-sf7) (cfr. Fig. 3). The first two separable components of a_3 are (sf2.cmp0) (sf2.cmp1) and of a_8 are (sf7.cmp0) and (sf7.cmp1). (Bottom left) Approximation error v.s. number of separable components compared for the four functions. The rotated copies of the template kernel require more terms than the functions a_i. The number of X-Y separable components necessary to achieve 1% error is: 7 and 8 for the rotated copies of the template function, and 1 and 3 for the singular functions. (Bottom right) The number of components necessary to approximate a_i to less than 1% error is plotted against i. From the plots one may deduce that this number is approximately equal to $1 + i/4$, so that the total number of 1-D convolutions required to implement the n-approximation is $2n + n^2/4$.

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. 8 (Bottom-left) 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. One must remember that beyond X-Y-separation there are a number of techniques for speeding up 2D FIR filtering, for example small generating kernel (SGK) filtering [1], that could further speed up the convolutions necessary to implement deformable filtering.

5 Rotation and scale

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

$$F_\sigma(\mathbf{x}) = \sum_i s_i(\sigma)d_i(\mathbf{x}) \qquad \sigma \in (0, \infty) \tag{12}$$

Unfortunately the domain of definition of s is not bounded (it is the real line) and therefore the kernel $F_\sigma(\mathbf{x})$ is not Hilbert-Schmidt (it has infinite norm). As a consequence the spectrum of the LL^* and L^*L operators is continuus and no discrete approximation may be computed.

One has therefore to renounce to the idea of generating a continuum of scales spanning the whole positive line. This is not a great loss: the range of scales of interest is never the entire real line. An interval of scales (σ_1, σ_2), with $0 < \sigma_1 \leq \sigma_2 < \infty$ is a very realistic scenario; if one takes the human visual system as an example, the range of frequencies to which it is most sensitive goes from approximatly 2 to 16 cycles per degree of visual angle i.e. a range of 3 octaves. In this case the interval of scales is compact and one can apply the results of section 3 and calculate the SVD and therefore an L_2-optimal finite approximation.

In this section the optimal scheme for doing so is proposed. The problem of simultaneously steering and scaling a given kernel $F(\mathbf{x})$ generating a family $F_{(\sigma,\theta)}(\mathbf{x})$ wich has a finite approximation will be tackled. Previous non-optimal schemes are due to Perona [36, 37] and Freeman and Adelson [15].

5.1 Polar-separable decomposition

Observe first that the functions a_i defined in eq.(6) are polar-separable. In fact $\mathbf{x}$ may be written in polar coordinates as $\mathbf{x} = \|\mathbf{x}\|R_{\phi(\mathbf{x})}\mathbf{u}$ where $\mathbf{u}$ is some fixed unit vector (e.g. the 1st coordinate axis versor) and $\phi(\mathbf{x})$ is the angle between $\mathbf{x}$ and $\mathbf{u}$ and $R_{\phi(\mathbf{x})}$ is a rotation by ϕ. Substituting the definition of F_θ in (6) we get:

$$\begin{aligned} a_i(\mathbf{x}) &= \sigma_i^{-1} \int_{\mathbb{S}^1} \overline{F(\|\mathbf{x}\|R_{\theta+\phi(\mathbf{x})}(\mathbf{u}))} e^{j2\pi\nu_i\theta} d\theta = \\ &= \sigma_i^{-1} e^{-j2\pi\nu_i\phi(\mathbf{x})} \int_{\mathbb{S}^1} \overline{F(\|\mathbf{x}\|R_\psi(\mathbf{u}))} e^{j2\pi\nu_i\psi} d\psi \end{aligned}$$

so that (2) may be also written as :

$$F_\theta(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i c_i(\|\mathbf{x}\|) e^{j2\pi\nu_i(\theta-\phi(\mathbf{x}))} \tag{13}$$

$$c_i(\|\mathbf{x}\|) = \sigma_i \int_{\mathbb{S}^1} \overline{F(\|\mathbf{x}\| R_{\psi(\mathbf{u})})} e^{j2\pi\nu_i\psi} d\psi \tag{14}$$

The scaling operation only affects the radial components c_i and does not affect the angular components. The problem of scaling the kernels a_i, and therefore F_θ through its decomposition, is then the problem of finding a finite (approximate) decomposition of continuously scaled versions of functions $c(\rho)$:

$$c_\sigma(\rho) = \sum_k s_k(\sigma) r_k(\rho) \qquad\qquad \sigma \in (\sigma_1, \sigma_2) \tag{15}$$

If the scale interval (σ_1, σ_2) and the function c are such that the operator L associated to F is compact then we can obtain the optimal finite decomposition via the singular value decomposition. The conditions for compactness of L are easily met in the cases of practical importance: it is sufficient that the interval (σ_1, σ_2) is bounded and that the norm of $c(\rho)$ is bounded ($\rho \in \mathbf{R}^+$).

Even if these conditions are met, the calculations usually cannot be performed analytically. One can employ a numerical routine as in sec. 4.2 for X-Y-separation and for each c_i (below indicated as c^i) obtain an SVD expansion of the form:

$$c^i_\sigma(\rho) = \sum_k \gamma^i_k s^i_k(\sigma) r^i_k(\rho) \tag{16}$$

As discussed before one can calculate the approximation error from the sequence of the singular values γ^i_k. Finally, substituting (16) into (14) the scale-orientation expansion takes the form (see Fig. 11):

$$F_{\theta,\sigma}(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i e^{j2\pi\nu_i(\theta-\phi(\mathbf{x}))} \sum_{k=1}^{n_i} \gamma^i_k s^i_k(\sigma) r^i_k(\|\mathbf{x}\|) \tag{17}$$

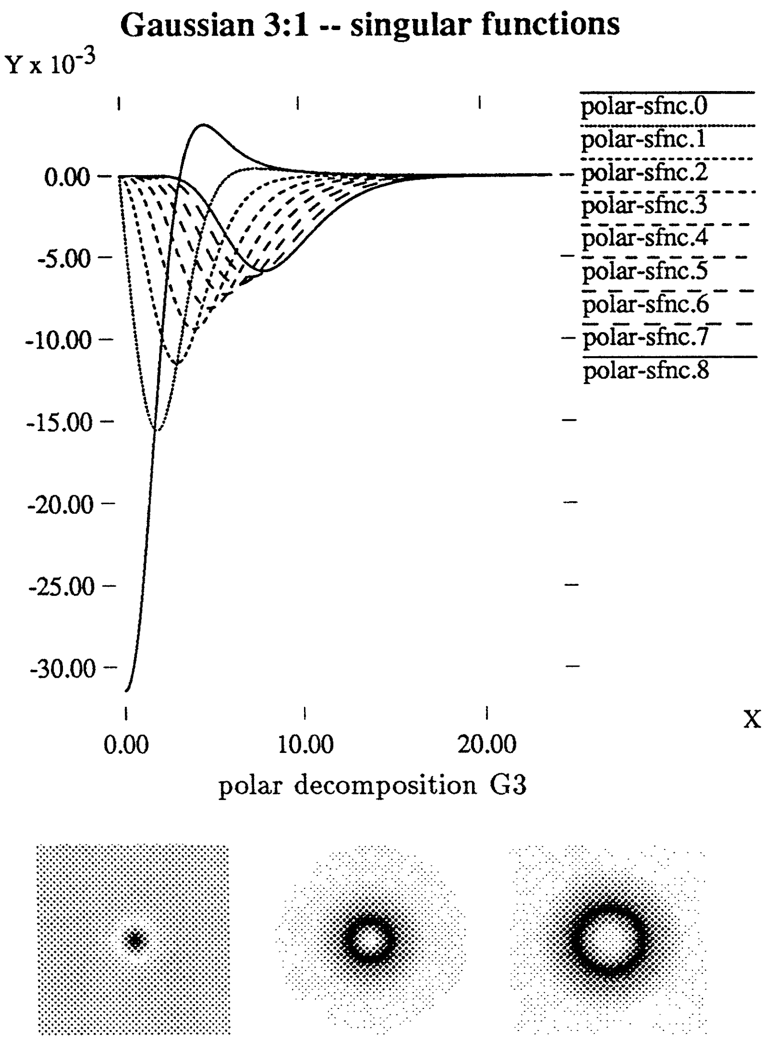
Filtering an image I with a deformable kernel built this way proceeds as follows: first the image is filtered with kernels $a^i_k(\mathbf{x}) = \exp(-j2\pi\nu_i\phi(\mathbf{x}))r^i_k(\|\mathbf{x}\|)$, $i = 0, \ldots, N$, $k = 0, \ldots, n_i$, the outputs I^i_k of this operation can be combined as $I_{\theta,\sigma}(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i b_i(\theta) \sum_{k=1}^{n_i} \gamma^i_k s^i_k(\sigma) I^i_k(\mathbf{x})$ to yeld the result. The filtering operations described above can of course be implemented as X-Y-separable convolutions as described in sec. 4.2.

5.1.1 Polar-separable decomposition, experimental results

An orientation-scale decomposition was performed on the usual kernel (second derivative of a Gaussian and its Hilbert transform, $\sigma_x : \sigma_y = 3 : 1$). The decomposition described in sec. 4.1 was taken as a starting point. The corresponding functions c_i of eq. 13 are shown in Fig. 9.

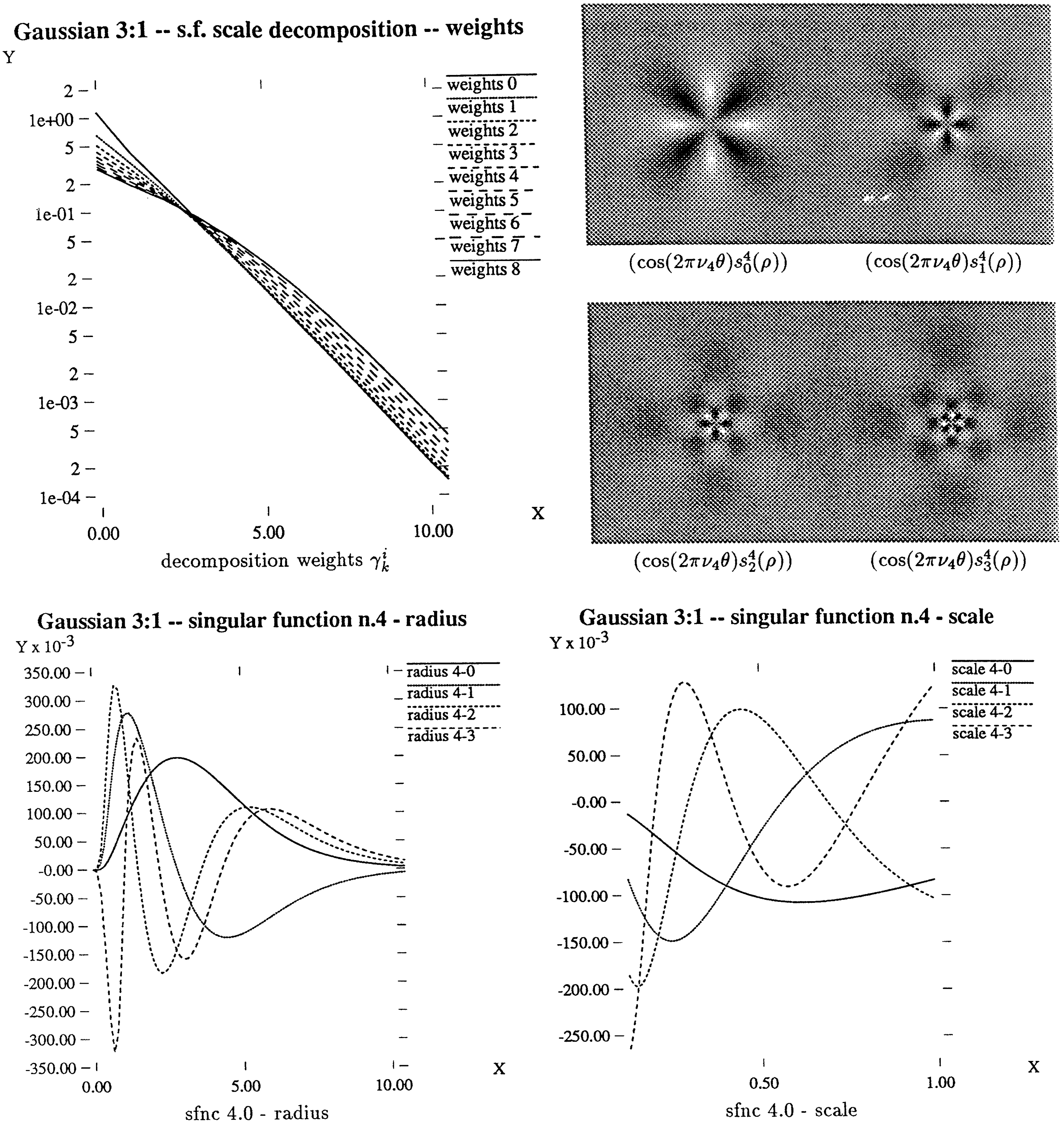
The interval of scales chosen was (σ_1, σ_2) s.t. $\sigma_1 : \sigma_2 = 1 : 8$, an interval which is arguably ample enough for a wide range of visual tasks.

The range of scales was discretized in 128 samples for computing numerically the singular value decomposition $(\gamma^i_k, s^i_k, r^i_k)$ of $c^i_\sigma(\rho)$. The computed weights γ^i_k are plotted on a logarithmic scale in Fig. 10 (Top). The 'X' axis corresponds to the k index, each curve is indexed by i, $i = 0, \ldots, 8$. One can see that for all the c_i the error decreases exponentially at approximately the same rate.



sfnc 0 sfnc 4 sfnc 8

Figure 9: (Top)The plots of $c_i(\rho)$, the radial part of the singular functions a_i (cfr. eq. 13). The θ part is always a complex exponential. The original kernel is the same as in fig. 3. (Bottom) The 0th, 4th and 8th components c_0, c_4 and c_8 represented in two dimensions.



Gaussian 3:1 -- s.f. scale decomposition -- weights

decomposition weights γ_k^i

$(\cos(2\pi\nu_4\theta)s_0^4(\rho))$ $(\cos(2\pi\nu_4\theta)s_1^4(\rho))$

$(\cos(2\pi\nu_4\theta)s_2^4(\rho))$ $(\cos(2\pi\nu_4\theta)s_3^4(\rho))$

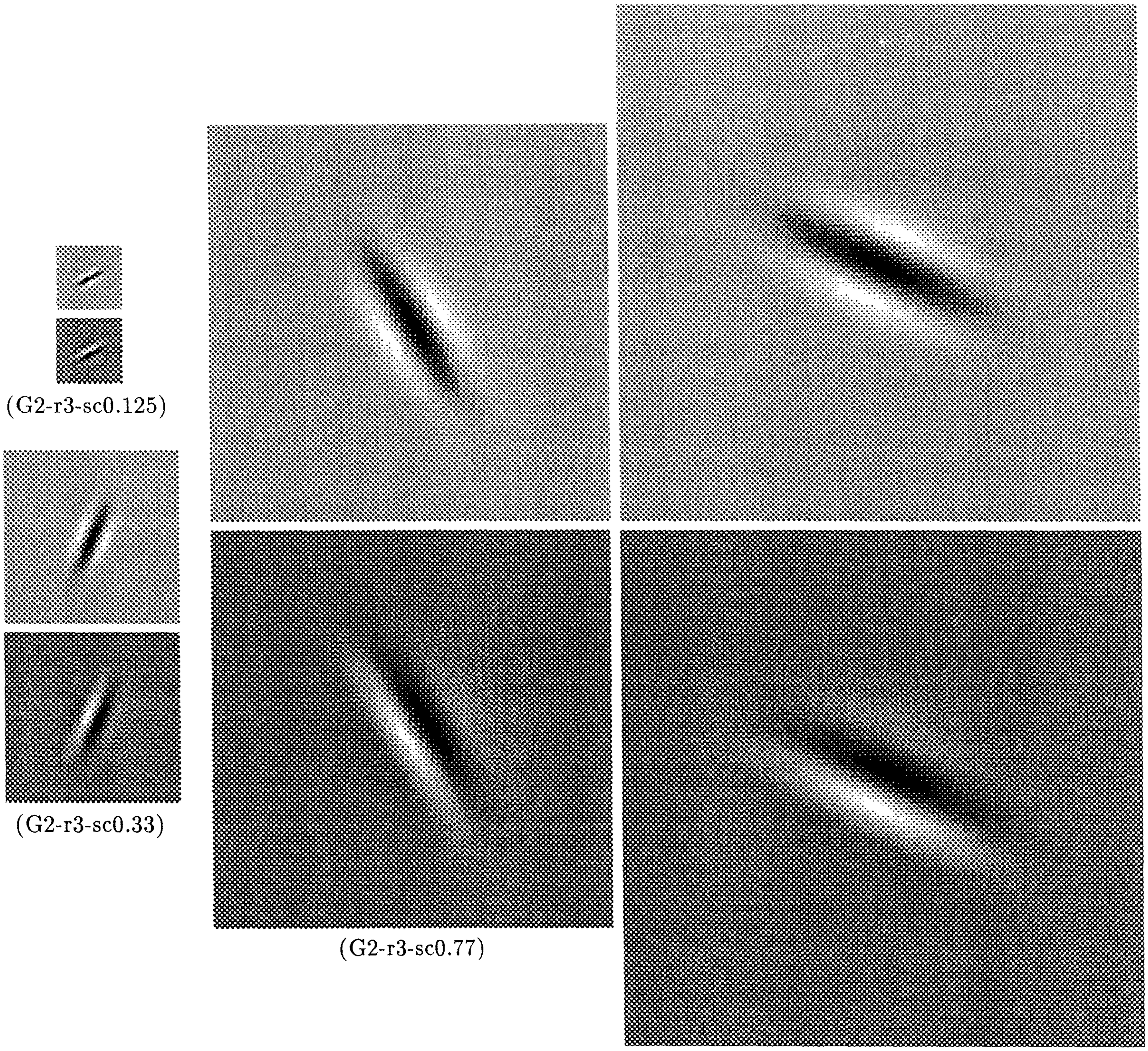
Gaussian 3:1 -- singular function n.4 - radius

sfnc 4.0 - radius

Gaussian 3:1 -- singular function n.4 - scale

sfnc 4.0 - scale

Figure 10: Scale-decomposition of the radial component of the functions a_i. The interval of scales σ is $\sigma \in (0.125, 1.00)$. See also Fig. 11. (Top-left) The weights γ_k^i of each polar functions' decomposition ($i = 0, \ldots, 8$, k along the x axis). The decay of the weights is exponential in k; 5 to 8 components are needed to achieve 1% error (e.g 5 for the 0th, 7 for the 4th and 8 for the 8th shown in fig 9). (Bottom) The first four radial (left) and scale (right) components of the 5th singular function: $r_k^4(\rho)$ and $s_k^4(\sigma)$, $k = 0, \ldots, 3$ (see Eq. (16)). (Top-right) The real parts of the first four scale-components of the 5th singular function $a_5$19cos$(2\pi\nu_4\theta)s_k^4(\rho)$ with $k = 0, \ldots, 3$ (see Eq. (17)).



(G2-r3-sc0.125)

(G2-r3-sc0.33)

(G2-r3-sc0.77)

(G2-r3-sc1.00)

Figure 11: The kernel at different scales and orientations: the scales are (left to right) 0.125, 0.33, 0.77, 1.00. The orientations are (left to right) 30^o, 66^o, 122^o, 155^o. The kernels shown here were obtained from the scale-angle decomposition shown in the previous figures.

The components $r^i_k(\rho)$ and $s^i_k(\sigma)$, $i = 4$, $k = 0, \ldots, 3$ are shown in the two plots at the bottom of Fig. 10.

In figure Fig. 11 reconstructions of the kernel based on a 1% error decomposition are shown for various scales and angles. A maximum of 1% error was imposed on the original steerable decomposition, and again on the scale decomposition of each single a_i. The measured error was 2.5% independently from angle and scale. The total number of filters required to implement a 3-octave 1% (nominal, 2.5% real) approximation error of the 3:1 Gaussian pair is 16 (rotation) times 8 (scale) = 128. If 10% approximation error is allowed the number of filters decreases by approximately a factor 4 to 32.

6 Open issues and discussion

A few issues remain open to investigation:

1. Sometimes one would like to generate a discrete decomposition of a family of filters that obeys other constraints than just being the most parsimonious one. For example (a.1) hardware limitations could constrain the shape of the interpolating funtions $b(\theta)$, (a.2) one would like to build pyramid implementations of the decomposition for speeding up the filtering stage (work on this issue has been done by Simoncelli et al. [11]).

2. Another interesting question mentioned in the introduction is the synthesis of the discrete decomposition directly from the specification of an early vision task, rather than passing through the synthesis of a 2D (nD) kernel which then is deformed somewhat arbitrarily. Work in this direction has been done by Hueckel [22, 23], Hummel [24], and Haralick [17] who approached the problem of feature (step edge, line in [23]) detection and localization as one of projecting image neighbourhoods on small-dimension linear subspaces, and deriving the relevant parameters (orientation, position) of the feature from this reduced representation.

   Hummel's approach is particularly interesting: the parameters describing the feature are modelled as continuous random variables. The neighbourhood operators (= kernels of the linear filters) used to project each neighbourhood onto a small-dimensional subspace space are selected using the Karhunen-Loève transform. Such procedure guarantees that the projection maximizes the variance of the parameters and therefore the parameters thus obtained are maximally informative.

   The similarity of the kernels derived by Hueckel amd Hummel to the a_i depicted in Figure 3 is not totally surprising: the polar separability and the fact that the tangential component of the kernels is sinusoidal has to be expected from the fact that one of the parameters in question is a rotation in the plane.

7 Conclusions

A technique has been presented for implementing families of deformable kernels for early vision applications. A given family of kernels obtained by deforming continuously a template kernel is approximated by interpolating a finite discrete set of kernels. The technique may be applied if and only if the family of kernels involved satisfy a compactness condition. This improves upon previous work by Freeman and Adelson on steerable filters in that (a) it is formulated with maximum generality to the case of any compact deformation, or, equivalently any compact family of kernels, and (b) it provides a design technique which is guaranteed to find the most parsimonious discrete approximation.

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

8 Acknowledgements

I am very grateful to Massimo Porrati, Alberto Grunbaum, David Donoho, Federico Girosi and Frank Ade for giving me good advice and useful references. I would also like to acknowledge useful conversations with Ted Adelson, Stefano Casadei, Charles Desoer, Peter Falb, Bill Freeman, Milan Jovovic, Takis Konstantopoulos, Olaf Kübler, Paul Kube, Jitendra Malik, Stephane Mallat, Sanjoy Mitter, Richard Murray. The simulations in this work have been carried out using Paul Kube's "viz" image-manipulation package, and would have taken much longer without Paul's very friendly support. The images in this paper have been printed with software kindly provided by Eero Simoncelli. Some of the simulations have been run on a computer kindly provided by prof. Canali of the Università di Padova. Part of this work was carried out at the International Computer Science Institute at Berkeley. This work was partially conducted while at MIT-LIDS with the Center for Intelligent Control Systems sponsored by U.S. Army Research Office grant number DAAL 03-86-K-0171.

References

[1] J-F. Abramatic and O. Faugeras. Sequential convolution techniques for image filtering. *IEEE trans. Acoust., Speech, Signal Processing*, 30(1):1–10, 1982.

[2] E. Adelson and J. Bergen. *Computational models of visual processing. M. Landy and J. Movshon eds.*, chapter "The plenoptic function and the elements of early vision". MIT press, 1991. Also appeared as MIT-MediaLab-TR148. September 1990.

[3] E. Adelson and James Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am.*, 2(2):284–299, 1985.

[4] T.O. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17:205–244, 1981.

[5] A. Bovik, M. Clark, and W.S. Geisler. *IEEE trans. Pattern Anal. Mach. Intell.*, 1990.

[6] P.J. Burt and E.A. Adelson. The laplacian algorithm as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[7] John Canny. A computational approach to edge detection. *IEEE trans. Pattern Anal. Mach. Intell.*, 8:679–698, 1986.

[8] Gustave Choquet. *Lectures on analysis*, volume I. W. A. Benjamin Inc., New York, 1969.

[9] J. Dieudonne. *Foundations of modern analysis.* Academic Press, New York, 1969.

[10] J.J. Dongarra, C.B. Moler, J.R. Bunch, and G.W. Stuart. *Linpack, user's guide.* Society for Industrial and Applied Mathematics, 1972.

[11] E. Adelson E. Simoncelli, W. Freeman and D. Heeger. Shiftable multi-scale transforms. Technical Report 161, MIT-Media Lab, 1991.

[12] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminators. *Biol. Cybern.*, 61:103–113, 1989.

[13] W. Freeman and E. Adelson. Steerable filters for image analysis. Technical Report 126, MIT, Media Laboratory, 1990.

[14] W. Freeman and E Adelson. The design and use of steerable filters for image analysis, enhancement and multi-scale representation. *IEEE trans. Pattern Anal. Mach. Intell.*, 1991.

[15] W. T. Freeman and E. 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.

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

[17] R. Haralik. Digital step edges from zero crossing of second directional derivatives. *IEEE trans. Pattern Anal. Mach. Intell.*, 6(1):58–68, 1984.

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

[19] B. Horn. The binford-horn linefinder. Technical report, MIT AI Lab. Memo 285, 1971.

[20] D. Hubel and T. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol. (Lond.)*, 148:574–591, 1959.

[21] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol. (Lond.)*, 160:106–154, 1962.

[22] M. Hueckel. An operator which locates edges in digitized pictures. *J. Assoc. Comp. Mach.*, 18(1):113–125, 1971.

[23] M. Hueckel. A local visual operator which recognizes edges and lines. *J. Assoc. Comp. Mach.*, 20(4):643–647, 1973.

[24] R. Hummel. Feature detection using basis functions. *Comp. Vision, Graphics and Image Proc.*, 9:40–55, 1979.

[25] D. Jones and J. Malik. Computational stereopsis–beyond zero-crossings. *Invest. Ophtalmol. Vis. Sci. (Supplement)*, 31(4):529, 1990.

[26] D. Jones and J. Malik. Using orientation and spatial frequency disparities to recover 3d surface shape – a computational model. *Invest. Ophtalmol. Vis. Sci. (Supplement)*, 32(4):710, 1991.

[27] Michael Kass. Computing visual correspondence. In *Proceedings: Image Understanding Workshop*, pages 54–60, McLean, Virginia, June 1983. Science Applications, Inc.

[28] H. Knuttson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *Workshop on Computer Architecture for Pattern Analysis ans Image Database Management*, pages 206–213. IEEE Computer Society, 1983.

[29] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367–375, 1987.

[30] J. Malik and Ziv Gigus. A model for curvilinear segregation. *Invest. Ophtalmol. Vis. Sci. (Supplement)*, 32(4):715, 1991.

[31] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America – A*, 7(5):923–932, 1990.

[32] M.C. Morrone and D.C. Burr. *Robots and biological systems*, chapter : "A model of human feature detection based on matched filters". Academic Press, eds. P. Dario and G. Sandini, 1990.

[33] M.C. Morrone, D.C. Burr, J. Ross, and R. Owens. Mach bands depend on spatial phase. *Nature*, (324):250–253, 1986.

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

[35] Pierre Parent and Steven Zucker. Trace inference, curvature consistency, and curve detection. *IEEE trans. Pattern Anal. Mach. Intell.*, 11(8):823–839, 1989.

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

[37] P. Perona. Deformable kernels for early vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–227, June 1991.

[38] P. Perona and J. 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.

[39] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proceedings of the Third International Conference of Computer Vision*, pages 52–57. IEEE Computer Society, Osaka, 1990.

[40] A. Pinkus. *n-Widths in Approximation Theory.* Springer Verlag, 1985.

[41] W. K. Pratt. An intelligent image processing display terminal. In *Proc. SPIE Tech. Symp.*, page Vol 27, San Diego, 1979.

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

[43] Walter Rudin. *Functional Analysis.* Mc.Graw-Hill, 1973.

[44] M.R. Turner. Texture discrimination by gabor functions. *Biol. Cybern.*, 55:71–82, 1986.

[45] S. Zhong and S. Mallat. Compact image representation from multiscale edges. In *Proceedings of the Third International Conference of Computer Vision.* IEEE Computer Society, Osaka, 1990.

A Appendix

A.1 Proof of theorem 1

What follows is a derivation to prove theorem 1. The proof is summarized at the end of the section

Proof. The family of functions F_θ defined by eq. (1) may be thought of as the kernel associated to a linear operator $L : A \to B$, defined by

$$b(\theta) = (La)(\theta) = \int_{\mathbb{R}^2} F_\theta(\mathrm{x}) a(\mathrm{x}) d\mathrm{x} \tag{18}$$

where $A = L^2(\mathbb{R}^2)$, and $B = L^2(\mathbb{S}^1)$, the square integrable functions defined on the plane and the circle respectively, and $a \in A, b \in B, \theta \in \mathbb{S}^1$. Let L^* denote the adjoint to L, i.e. the unique linear operator satisfying the equality

$$\langle La, b \rangle_B = \langle a, L^* b \rangle_A \tag{19}$$

with $\langle \cdot, \cdot \rangle_C$ indicating the inner product of a Hilbert space C.

If $||F|| \leq \infty$ then F_θ is a Hilbert-Schmidt kernel and L has a discrete spectrum; then F_θ can be written as a sum:

$$F_\theta(\mathrm{x}) = \sum_{i=1}^{N} \sigma_i a_i(\mathrm{x}) b_i(\theta) \tag{20}$$

where the σ_i^2 are the nonzero (positive, in fact) eigenvalues of the auto-adjoint operators $LL^* = L \circ L^*$ and $L^*L = L^* \circ L$, and the a_i and b_i are the associated eigenfunctions of L^*L and LL^* respectively, and N could be infinite. The collection of triples $(\sigma_i, a_i, b_i)_{i=1,\ldots,N}$ is the singular value decomposition (SVD) of L (see e.g. [43]).

Observe that expression (20) is in the desired form of (1), with the additional advantage that the a_i and b_i are orthonormal bases of A and B (see below). Therefore if one could calculate the SVD (i.e. the a_i, b_i and σ_i) of L explicitly one would be able to write F_θ as in (1), and the problem would be solved.

L^* can be computed from its definition (19) and (18) and is:

$$(L^* b)(\mathrm{x}) = \int_{\mathbb{S}^1} \overline{F_\theta(\mathrm{x})} b(\theta) d\theta \tag{21}$$

hence the LL^* operator is

$$(LL^* b)(\theta) = \int_{\mathbb{S}^1} H(\theta, \theta') b(\theta') d\theta' \tag{22}$$

$$H(\theta, \theta') = \int_{\mathbb{R}^2} F_\theta(\mathrm{x}) \overline{F_{\theta'}(\mathrm{x})} d\mathrm{x} \tag{23}$$

and L^*L is

$$(L^* La)(\mathrm{x}) = \int_{\mathbb{R}^2} K(\mathrm{x}, \mathrm{x}') a(\mathrm{x}') d\mathrm{x}' \tag{24}$$

$$K(\mathrm{x}, \mathrm{x}') = \int_{\mathbb{S}^1} F_\theta(\mathrm{x}) \overline{F_\theta(\mathrm{x}')} d\theta \tag{25}$$

Observe that the kernel associated with LL^* is a function of the difference of its arguments only. To see that change the variable of integration in (23), $\mathrm{y} = R_{\theta'}\mathrm{x}$, obtaining $H(\theta, \theta') = H(\theta - \theta', 0) = h(\theta - \theta')$.

The eigenvalue-eigenvector problem for LL^*

$$LL^* b_i = \lambda_i b_i \tag{26}$$

can tnen be solved explicitly substituting (22) in (26):

$$\int_{\mathbb{S}^1} h(\theta - \theta') b_i(\theta') d\theta' = \lambda_i b_i(\theta) \tag{27}$$

The equation holds between the Fourier transforms of the two terms of the shift-invariant convolution of eq. (27):

$$\hat{h}(\nu)\hat{b}_i(\nu) = \lambda_i \hat{b}_i(\nu) \tag{28}$$

The nonzero solutions of this equation are the couples $(\lambda_i, \hat{b}_i)$ s.t. $\lambda_i = \hat{h}(\nu_i)$ and $\hat{b}_i(\nu) = \delta(\nu - \nu_i)$; therefore

$$\sigma_i = \hat{h}(\nu_i)^{1/2} \qquad b_i(\theta) = e^{j2\pi\nu_i\theta} \tag{29}$$

where, by convention, the frequency numbers ν_i are ordered so that $\hat{h}(\nu_i)$ is a nonincreasing sequence: $\hat{h}(\nu_i) \geq \hat{h}(\nu_{i+1}) > 0$. For multiple eigenvalues any linear combinations of the corresponding eigenfunctions (eigenvectors) will also be eigenfunctions.

The eigenfunctions of the L^*L operator cannot be easily determined directly from its integral expression (24) as for LL^*. However one can make use of the fact that L^*b_i is an eigenfunction of L^*L, which can be verified as follows: $L^*L(L^*b_i) = L^*(LL^*b_i) = L^*\lambda_i b_i = \lambda_i(L^*b_i)$. The unit-norm eigenfunctions of L^*L are therefore the functions defined by

$$a_i(\mathbf{x}) = \lambda_i^{-1/2} L^* b_i = \sigma_i^{-1} \int_{\mathbb{S}^1} \overline{F_\theta(\mathbf{x})} e^{j2\pi\nu_i\theta} d\theta \tag{30}$$

i.e. at each $\mathbf{x}$, $a_i(\mathbf{x})$ is the conjugate of the Fourier coefficient of $F_\theta(\mathbf{x})$ corresponding to the frequency $-\nu_i$.

In conclusion (the numbers refer to the corresponding statements of the theorem):

1. Follows from the properties of SVD and the fact that the sum (20) is built using the SVD triples.
2. As above.
3. From equation (30) and the fact that the dimension of the range of L^* is equal to N, the number of nonzero eigenvalues.
4. Follows from SVD properties.
5. Follows from the fact that $\sigma_i \geq \sigma_{i+1}$ and that $\sum_i \sigma_i = ||L||^2 < \infty$. May be seen directly from SVD properties.
6. From (13) one can see that $F_\theta^{[n]}$ is a function of $|\mathbf{x}|$ and $\theta - \phi(\mathbf{x})$ only.
7. The functions a_i are linearly independent, so any collection of n of them spans an n-dimensional subspace of $L^2(\mathbb{R}^2)$. This is the same subspace spanned by any linearly independent collection $F_{\theta_i}^{[n]} i = 1, \ldots, n$. The thesis follows from the fact $F_{\theta_i}^{[n]} = F_{\theta=0}^{[n]} \circ R_{\theta_i}$. The coefficients α_i can be obtained with the usual change of basis formulae.

□