PYRAMIDAL IMPLEMENTATION OF DEFORMABLE KERNELS

Roberto Manduchi[†] *and Pietro Perona*[†‡]

[†]California Institute of Technology 116–81, Pasadena, CA 91125
[‡] D.E.I., Università di Padova, Via Gradenigo 6/a, 35131 Padova (Italy)
{manduchi,perona}@systems.caltech.edu

ABSTRACT

In computer vision and, increasingly, in rendering and image processing, it is useful to filter images with continuous rotated and scaled families of filters. For practical implementations, one can think of using a discrete family of filters, and then to interpolate from their outputs to produce the desired filtered version of the image.

We propose a multirate implementation of deformable kernels, capable to further reduce the computational weight. The "basis" filters are applied to the different levels of a pyramidal decomposition. The new system is not shift-invariant – it suffers from "aliasing". We introduce a new quadratic error criterion which keeps into account the ineherent system aliasing.

By using hypermatrix and Kronecker algebra, we are able to cast the global optimization task into a multilinear problem. An iterative procedure ("pseudo–SVD") is used to minimize the overall quadratic approximation error.

1. INTRODUCTION

Several image analysis techniques require the linear filtering of an image (or of a sequence of images) with a (possibly large) set of different kernels. Algorithms for motion flow computation, texture analysis and classification, stereo matching, curved lines grouping, brightness boundary detection, quite often require to convolve the image(s) with many suitably scaled and oriented versions of one or more prototype kernels.

Filtering an image with many different filters is computationally expensive. However, one may reduce the computational cost by exploiting the fact that the outputs of a bank of multi-oriented – multi-scaled filters are typically higly correlated. Consider for example the class of "steerable filters" [1]. In this case, there exists a finite set $\{g^{[r]}(\mathbf{x})\}$ of "basis" filters such that, for *any* orientation θ, a suitable linear combination of the outputs of such filters gives the version of the image filtered at orientation θ. If filtering at many different orientations is required, the computational weight is effectively reduced this way: convolve the image with the kernels $\{g^{[r]}(\mathbf{x})\}$ once for all, store the outputs, and then recombine them with coefficients that are a function of the desired orientations.

Unfortunately, the exact steerable decomposition exists only for a very restricted class of kernels [1]. We may then be interested in an approximated solution. Perona [2] studied the following problem: given a set $\{d(\mathbf{x},\sigma,\theta)\}$ of scaled (by σ) and oriented (of θ) versions of some given kernel $d(\mathbf{x})$, find a suitable decomposition order R and basis kernels and recombining functions $\{f^{[r]}(\mathbf{x}), t^{[r]}(\theta), s^{[r]}(\sigma), 1 \leq r \leq R\}$ such that the following quadratic error is minimized:

$$e^2 = \|d(\mathbf{x},\sigma,\theta) - \sum_{r=1}^{R} f^{[r]}(\mathbf{x})t^{[r]}(\theta)s^{[r]}(\sigma)\|^2 \qquad (1)$$

Perona's technique computes successive approximations using the singular value decomposition (SVD), after discretizing the variables $\mathbf{x}, \theta, \sigma$. Note that the algorithm can be put to work for *any* FIR prototype $d(\mathbf{x})$.

In order to reduce the number of elementary operations per input pixel (OPP), one may add the constraint that the filters $\{f^{[r]}(\mathbf{x})\}$ be separable, i.e. $\{f^{[r]}(\mathbf{x}) = u^{[r]}(x)v^{[r]}(y)\}$ (where $\mathbf{x} \overset{\mathrm{def}}{=} (x,y)$). Such an extension has been considered in [3] by Shy and Perona, which used the "pseudo–SVD" algorithm.

In the present paper, we take into exam a pyramidal implementation of the steerable–scalabe decomposition which further reduces the overall computational weight. To understand the idea at the basis of our work, consider a 1-D example: we want to find a set of "basis" filters to approximate the scaled versions $\{d(x,\sigma) = d(x/\sigma)\}$ of a prototype kernel $d(x)$ along a continuous range of scales of few octaves (see Fig. 1(a)). The support of the scaled kernels can be assumed to be proportional to the scale σ. A drawback of the SVD–based algorithm of Perona [2], is that the basis filters share the same support, corresponding to the largest support of the filters $\{d(x,\sigma)\}$ (see Fig. 1(b), where we set the decomposition order R to 4). This problem may be solved by choosing a different basis for the approximation space, by linearly combining the basis kernels found using the SVD. Consider for example a basis constituted by our scalable approximations to filters $\{d(x,\sigma_r)\}$, where $\{\sigma_r, 1 \leq r \leq R\}$ is a coarse sampling (perhaps logarithmic, like in the case of Fig. 1) of the scale axis. If the overall ap-

We gratefully acknowledge support from NSF Grant IRI 9306155 on "Geometry driven diffusions", NSF Research Initiation grant IRI 9211651, ONR grant NOOO14-93-1-0990, a NSF National Young Investigator Award to P.P.. This work is supported in part by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program; and by the California Trade and Commerce Agency, Office of Strategic Technology.

0-8186-7310-9/95 $4.00 © 1995 IEEE

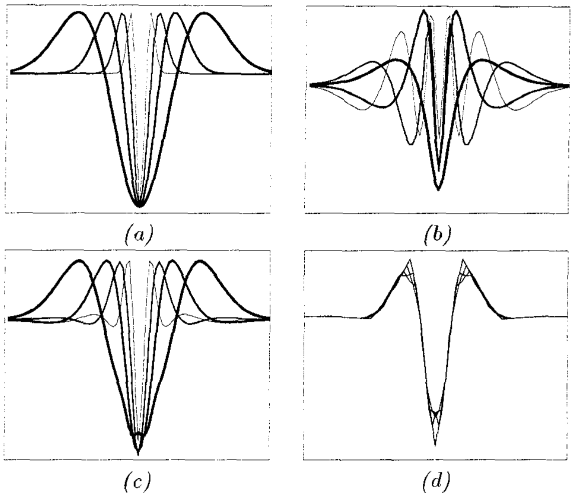

Figure 1: *(a) Example of scaled versions $d(x, \sigma = 2^r), r \in \{1, 5/3, 7/3, 3\}$ of a prototype kernel. (b) The $R = 4$ orthonormal basis filters obtained using the SVD technique [2]. The kernels' length is 81 pixels. (c) The new basis functions, chosen as the approximated versions of the kernels of (a). (d) The 8 impulse responses of the pyramidal system of Section 3 corresponding to $\sigma = 4.6$.*

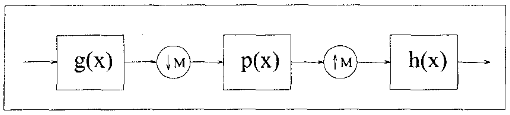

Figure 2: *Scheme for the multirate implementation of a filter.*

proximation error is reasonably small, we may expect that the supports of the new basis filters are again a function of the scale σ. Fig. 1 (c) shows the new basis functions relative to our former example. It may be seen that, except for the basis function corresponding to the smallest scale (drawn with the thinnest line), the support of the new basis functions can be assumed to be equal to that of the corresponding kernels $\{d(x, \sigma_r)\}$. The number of OPP using the new basis functions is then effectively reduced (since the number of OPP to implement an FIR filter is equal to the number of samples of the filter in its support). If large scales of the filters are required, then, in order to further reduce the computational weight, one may adopt a multirate implementation of the basis filters. The multirate implementation of FIR filters has been considered in the past [4]. Fig. 2 shows the most general scheme: a filter $d(x)$ is implemented as the cascade of a prefilter $g(x)$, a decimator by an integer factor M, a filter $p(x)$, an interpolator (or expander) by factor M, and a postfilter $h(x)$. Note that we consider only digital FIR filters, therefore it is understood that the variable x is discrete (actually integer). It is common practice in the literature to assume that $d(x)$ is strictly bandlimited within $[-\pi/M, \pi/M]$, so that it can be implemented using the multirate scheme of Fig. 2 without aliasing. Manduchi *et al.* [5] chose a different approach, and derived a novel approximation error criterion which keeps into account the inherent system's aliasing. More specifically, we can observe that a system like the one in Fig. 2 is periodically shift-invariant [6], i.e. it is fully characterized by M impulse responses $\{q_k(x + k)\}$, corresponding to inputs $\{\delta(x + k), 0 \leq k < M\}$. If the impulse responses are identical, then the system is alias-free. Otherwise, we can consider the following quadratic error:

$$e^2(n) = \sum_{k=0}^{M-1} \|d(x) - q_k(x)\|^2 \qquad (2)$$

An iterative linear procedure to design the components of the scheme of Fig. 2, in order to minimize error (2) for a given impulse response $d(x)$, has been described in [5].

We adopt a pyramidal scheme for our scalable (1-D) and steerable-scalable-separable (2-D) decomposition, and use a measure of the overall quadratic approximation error similar to (2), as described in Section 3. One might attempt to minimize the approximation error by designing a multirate implementation of each basis filter (obtained using the procedure of [2]) separately. However, this 2-step approximation procedure does not necessarily yield the best solution: to achieve the minimum of the approximation error, we should design the components of the multirate system *simultaneously*. An algorithm to fullfill such a task is presented in Section 3. In order to describe our technique, some preliminary results are first derived in Section 2, where we use the formalism of the hypermatrix and the Kronecker algebra to manipulate the multilinear expressions that appear in the approximation problem.

2. SEPARABLE–STEERABLE–SCALABLE DECOMPOSITION

We show here how the problem of the steerable-scalable-separable decomposition for 2-D kernels can be formalized and solved using the hypermatrix and Kronecker algebra notation. The results of this section will be similar to those of [3], but the new formalism introduced here will be instrumental to extending our scheme to the multirate case, considered in Section 3. The problem is to minimize the quadratic error

$$e^2 = \|d(x, y, \sigma, \theta) - \sum_{r=1}^{R} u^{[r]}(x) v^{[r]}(y) t^{[r]}(\theta) s^{[r]}(\sigma)\|^2 \qquad (3)$$

with respect to the variables $\{u^{[r]}(x), v^{[r]}(y), t^{[r]}(\theta), s^{[r]}(\sigma)\}$. Note that $\{u^{[r]}(x), v^{[r]}(y)\}$ are the basis filters, while $\{t^{[r]}(\theta), s^{[r]}(\sigma)\}$ are the recombination functions. It is understood that all the variables $\{x, y, \sigma, \theta\}$ are discrete.

Hypermatrix algebra [7],[8] is a convenient formalism to manipulate *indexed arrays*. It is designed to describe operations on multilinear tensors, but its isomorphism with the Kronecker algebra makes it particularly suitable to reduce multiliner expressions into canonical forms, for example in order to compute quadratic minimizations.

A *hypermatrix* $\mathcal{A}_{j_1,\ldots,j_n}^{i_1,\ldots,i_m}$ is an indexed array represented by a unique matrix $\mathbf{A}$ via the following mapping (*mixed radix* representation):

$$|\mathcal{A}_{j_1,\ldots,j_n}^{i_1,\ldots,i_m}| = a_{ij} \; , \qquad (4)$$

$\mathbf{C} = \mathbf{A}^T$	$\Leftrightarrow$	$\mathcal{C}_i^j := \mathcal{A}_j^i$
$\mathbf{C} = \mathbf{A}\pm$	$\Leftrightarrow$	$\mathcal{C}_j^i := \mathcal{A}_j^i \pm \mathcal{B}_j^i$
$\mathbf{C} = \mathbf{AB}$	$\Leftrightarrow$	$\mathcal{C}_y^i := \mathcal{A}_j^i \mathcal{B}_y^j$
$\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$	$\Leftrightarrow$	$\mathcal{C}_{jy}^{ix} := \mathcal{A}_j^i \mathcal{B}_y^x$
$\mathbf{c} = \mathrm{vec}_c(\mathbf{A})$	$\Leftrightarrow$	$\mathcal{C}^{ji} := \mathcal{A}_j^i$

Table 1: *Useful dualities between hypermatrix and Kronecker algebra expressions.*

$$i = i_1 + i_2 \cdot \langle i_1 \rangle + \ldots + i_m \cdot \langle i_1 \rangle \langle i_2 \rangle \ldots \langle i_{m-1} \rangle \,,$$
$$j = j_1 + j_2 \cdot \langle j_1 \rangle + \ldots + j_n \cdot \langle j_1 \rangle \langle j_2 \rangle \ldots \langle j_{n-1} \rangle \,,$$

where $|\mathcal{A}_{j_1,\ldots,j_n}^{i_1,\ldots,i_m}|$ represents the entry of the hypermatrix corresponding to indices $(i_1, \ldots, i_m), (j_1, \ldots, j_n)$, a_{ij} is the entry of matrix $\mathbf{A}$ at the i-th row and j-column, and $\langle i \rangle$ represents the maximum value assumed by index i (it is assumed that all the indices start from 1). The correspondence between a hypermatrix $\mathcal{A}_{j_1,\ldots,j_n}^{i_1,\ldots,i_m}$ and a matrix $\mathbf{A}$ is denoted as

$$\mathcal{A}_{j_1,\ldots,j_n}^{i_1,\ldots,i_m} \Leftrightarrow \mathbf{A}$$

Hypermatrices may be manipulated using a number of rules described in [8]. In particular, there exists a one-to-one correspondence between hypermatrix expressions and matrix operations in the Kronecker algebra. We list in Table 1 a few dualities instrumental to our work (for a detalied treatment, see [8]).

The symbol ":=" is used to specify a hypermatrix assignment. The symbol $\otimes$ denotes the Kronecker product, while by $\mathrm{vec}_c(\mathbf{A})$ it is meant the stacking of the columns of $\mathbf{A}$ into a single vector. Another useful hypermatrix expression represents the point-by-point multiplication along one (or more indices): for example, $\mathcal{C}_j^i := \mathcal{A}_j^i \mathcal{B}_j^i$.

We are ready now to restate our minimization problem in terms of hypermatrix expressions. Equation (3) may be rewritten as

$$e^2 = \|\mathcal{D}^{xy\sigma\theta} - \left(\mathcal{U}_r^x \mathcal{V}_r^y \mathcal{S}_r^\sigma \mathcal{T}_r^\theta\right) 1^r\|^2 \tag{5}$$

where $|1^r| = 1 \, \forall r$, while the other hypermatrices in the expression correspond to the functions of (3) in a straightforward fashion. It is clear that the minimization task becomes a multilinear problem in $\mathcal{U}, \mathcal{V}, \mathcal{S}, \mathcal{T}$ and it can be solved, as in [3], by iteratively minimizing (5) with respect to each variable, while keeping all the other ones fixed [1].

Suppose, for example, that we want to minimize e^2 with respect to the orientation reconstruction function $t(\theta)$. After some manipulations, expression (5) may be rewritten as

$$e^2 = \|\mathcal{D}^{xy\sigma\theta} - \bar{\mathcal{A}}_{r\bar{\theta}}^{xy\sigma\theta} \bar{\mathcal{T}}^{r\bar{\theta}}\|^2 \tag{6}$$

where

$$\bar{\mathcal{A}}_{r\bar{\theta}}^{xy\sigma\theta} := \mathcal{A}_r^{xy\sigma} \mathcal{I}_{\bar{\theta}}^{\theta} \,, \; \mathcal{A}_r^{xy\sigma} = \mathcal{U}_r^x \mathcal{V}_r^y \mathcal{S}_r^\sigma \,,$$
$$\bar{\mathcal{T}}^{r\bar{\theta}} := \mathcal{T}_{\bar{\theta}}^{r}$$

$\mathcal{I}_{\bar{\theta}}^{\theta}$ is an identity hypermatrix: $|\mathcal{I}_{\bar{\theta}}^{\theta}| = \delta(\theta - \bar{\theta})$. The usefulness of expression (6) is in the fact the the corresponding matrix expression is in canonical form (a matrix times a vector). Using the dualities reported in Table 1 expression (6) becomes

$$e^2 = \|(\mathbf{A} \otimes \mathbf{I}_{\langle\theta\rangle})\mathbf{t} - \mathbf{d}\|^2 \tag{7}$$

where $\mathbf{A} \Leftrightarrow \mathcal{A}_r^{xy\sigma}$, $\mathbf{t} \Leftrightarrow \bar{\mathcal{T}}^{r\bar{\theta}}$, $\mathbf{d} \Leftrightarrow \mathcal{D}^{xy\sigma\theta}$, and $\mathbf{I}_{\langle\theta\rangle}$ is the $\langle\theta\rangle \times \langle\theta\rangle$ identity matrix.

By applying some properties of the Kronecker algebra [8], the minimizing vector $\mathbf{t}$ for (7) can be expressed as

$$\mathbf{t} = \left(\left((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\right) \otimes \mathbf{I}_{\langle\theta\rangle}\right) \mathbf{d} \tag{8}$$

Expression (8) contains a Kronecker product with an identity matrix, which produces a (typically) large sparse matrix. This drawbak can be avoided as follows. Let $\mathbf{T} \Leftrightarrow \mathcal{T}_r^\theta$ and $\mathbf{D} \Leftrightarrow \bar{\mathcal{D}}_\theta^{xy\sigma} := \mathcal{D}^{xy\sigma\theta}$. Then, noting that $\mathbf{t} = \mathrm{vec}_c(\mathbf{T})$, one can prove that identity (8) is equivalent to

$$\mathbf{T} = \left((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{D}\right)^T \tag{9}$$

Note that no Kronecker product is present in equation (9). However, as we show in the next section, this last reduction cannot be used in the multirate case.

3. A SCALABLE PYRAMIDAL DECOMPOSITION

We present in this section our scheme for the pyramidal implementation of the system. Only the 1-D scalable case is considered here, for simplicity's sake. Although the 1-D case is not useful for vision and image processing, it serves to illustrate for the complete 2-D scheme, described in [9].

As anticipated in the Introduction, we consider a multirate implementation of the larger filters in the basis set. In particular, we use the pyramidal scheme shown in Fig. 3. The pyramidal scheme seemes appropriate, since typically one is interested in a logarithmic sampling of the scale axis. The filter $h(x)$ has impulse response [1 , 2 , 1]. Its frequency response is a raised cosine (therefore it is low-pass), and it may be implemented with only two sums and one multiplication (by 2) per input sample. For the l-th level of the pyramid, the decimation ratio is 2^{l-1}.

In the l-th level of the pyramid of Fig. 3, the parameters involved are are i) the number K_l of kernels $\{p_{li}(x)\}$ and ii) their lengths $\{M_{li}\}$. In [9] we derive a heuristic procedure to determine such parameters, given the characteristics of the filters $\{d(x, \sigma)\}$ to be approximated.

If the pyramid is composed by L levels, then the overall system is characterized by 2^{L-1} different impulse responses $\{q_k(x, \sigma)\}$, as discussed in the Introduction. The quadratic error must include all of them:

$$e^2 = \sum_{k=1}^{2^{L-1}} \|d(x, \sigma) - q_k(x, \sigma)\|^2 \tag{10}$$

We briefly describe in the following our procedure to *simultanously* design the filters $\{p_{li}(x)\}$ in order to minimize e^2 in (10). Let $\mathbf{p}$ be the vector obtained by stacking the impulse responses of the filters $\{p_{li}(x)\}$ of Fig. 3. It is shown in [9] that, if $\mathbf{S} \Leftrightarrow \mathcal{S}_\sigma^r$ is the scale reconstruction matrix[2],

[1] Note, however, that although we are guaranteed to converge to some solution, it will not necessarily be the global minimum of (5) [3].

[2] Note that now the index r spans $\sum_{l=1}^{L} K_l$ values.

the vector $\mathbf{q}_{k\sigma}$ representing the k-th impulse response at scale σ may be written as

$$\mathbf{q}_{k\sigma} = (\mathbf{S}_{\cdot,\sigma} \otimes \mathbf{I})\mathbf{H}_k\mathbf{J}\mathbf{p} \tag{11}$$

In the above expression, $\mathbf{S}_{\cdot,\sigma}$ is the σ-th column of $\mathbf{S}$, $\mathbf{I}$ is a suitably sized indentity matrix, $\mathbf{J}$ is a block–diagonal matrix $\mathbf{J} = \mathrm{diag}(\mathbf{J}_{li})$ where each $\mathbf{J}_{li}$ is an *expansion* matrix, and $\mathbf{H}_k$ is a block–diagonal matrix $\mathbf{H}_k = \mathrm{diag}(\mathbf{H}_{kli})$ where each $\mathbf{H}_{kli}$ is a Toeplitz matrix representing the filtering with a kernel corresponding to the k-th impulse response of the l-branch of the pyramid after short–circuiting the filters $\{p_{li}(x)\}$. Such impulse responses may be easily computed analytically using the polyphase decomposition of the system [6].

We can thus minimize the quadratic error (10) using an iterative algorithm, in a fashion similar to the case of Section 2. In particular, the minimizing vector $\mathbf{p}$ for fixed scale reconstruction matrix $\mathbf{S}$ is

$$\mathbf{p} = \left(\mathbf{J}^T\left(\sum_{k=1}^{2^{L-1}} \mathbf{H}_k^T(\mathbf{S}^T\mathbf{S} \otimes \mathbf{I}_{\langle x\rangle})\mathbf{H}_k\right)\mathbf{J}\right)^{-1} \cdot \tag{12}$$

$$\cdot\ \mathbf{J}^T\left(\sum_{k=1}^{2^{L-1}} \mathbf{H}_k^T\right)(\mathbf{S}^T \otimes \mathbf{I}_{\langle x\rangle})\mathbf{d}$$

where $\mathbf{d} \Leftrightarrow \mathcal{D}^{x\sigma}$ is the vector representing $d(x,\sigma)$.

In Fig. 3 we show the results of the pyramidal scalable approximation of the function $d(x,\sigma)$ of Fig. 1(a), for σ ranking from 2 to 8 (2 octaves). We used a four levels pyramid, with one filter per each level, and set the filter lengths as $\{M_{11} = 11, M_{21} = 11, M_{31} = 9, M_{41} = 7\}$. The optimal basis filters are shown in the left column of Fig. 3. The overall error is $e^2 = 0.013$. Note that we have imposed the central sample of each kernel to be equal to -1, and that the filters turned out to be linear–phase, i.e. symmetric (symmetry can actually be constrained in the design procedure). In the right column of Fig. 3 we reported the scale reconstruction functions $s_{li}(\sigma)$. In order to implement the basis filters, a total of 15 multiplications and 30 additions per input sample are required (note that the filter at the l-th level of the pyramid operates on a signal decimated by 2^{l-1}, and that the filter symmetry can be exploited to reduce the number of multiplications). Then, 4 multiplications and 3 additions per input sample are required for each scale in the reconstruction. The computational weight reduction with respect to the non–multirate scalable implementation (with basis filters as in Fig. 1(c)) is apparent, considering that it would require 70 multiplications and 140 additions per input sample to realize the basis filters. In Fig. 1(d), the 8 impulse responses of the multirate system, corresponding to the instance $\sigma = 4.6$, are shown.

The 2-D case (pyramidal–separable–steerable–scalable decomposition) is a conceptually straightforward extension of the scheme proposed in this section, although the notation becomes significantly more complicated, and it is studied in [9]. In particular, in [9] we consider a separable pyramid with L_x levels along th x-axis, and L_y levels along the y-axis. Hence, the overall system is characterized by $2^{L_x+L_y-2}$ impulse responses.

4. REFERENCES

[1] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906, 1991.

[2] P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):488–499, May 1995.

[3] D. Shy and P. Perona. X-Y separable pyramid steerable scalable kernels. In *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 237–244, Seattle, June 1994.

[4] L.R. Rabiner and R.E. Crochiere. A novel implementation for narrow–band FIR digital filters. *IEEE Trans. Acoust., Speech, Signal Processing*, 23:457–464, October 1975.

[5] R. Manduchi, P. Perona, and D. Shy. Least–squares multirate implementation of FIR filters. Submitted for publication.

[6] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, 1993.

[7] G. Tait, W. Vetter, and P. Bélanger. The array matrix for multivariate analysis. In *Third International Symposium on Large Engineering Systems*, pages 69–76, July 1980.

[8] D. G. Antzoulatos and A. A. Sawchuk. Hypermatrix algebra: Theory. *Computer Vision, Graphics and Image Processing: Image Understanding*, 57(1):24–41, January 1993.

[9] R. Manduchi, P. Perona, and D. Shy. Efficient implementations of deformable kernels. In preparation.

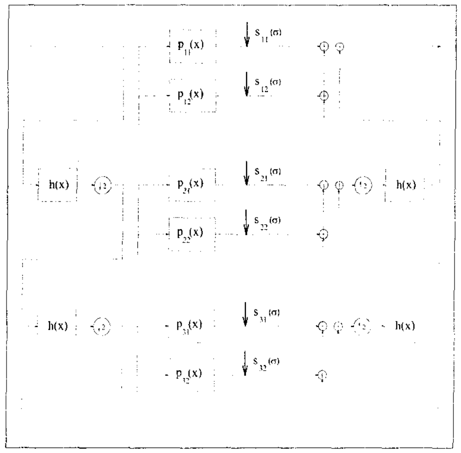

Figure 3: *Pyramidal scheme used in the multirate implementation of the scalable filters.*

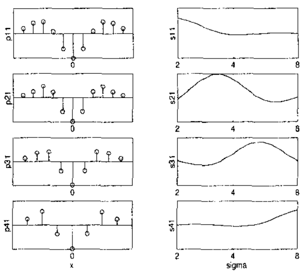

Figure 4: *Basis kernels (left column) and reconstruction functions (right column) corresponding to the scalable decomposition of the filters of Fig. 1(a) using the pyramidal scheme of Fig. 3 with 4 levels (one basis filter per each level). The central sample of the kernels was constrained to -1.*