Efficient Deformable Filter Banks

Roberto Manduchi, Pietro Perona† and Doug Shy *
California Institute of Technology 136–93, Pasadena, CA 91125
† D.E.I., Università di Padova, Via Gradenigo 6/a, 35131 Padova (Italy)

September 2, 1997

Abstract

This paper describes efficient schemes for the computation of a large number of differentely scaled/oriented filtered versions of an image. We generalize the well-known steerable/scalable ("deformable") filter bank structure by imposing X-Y separability on the basis filters. The resulting systems, designed by an iterative projections technique, achieve substantial reduction of the computational cost.

To reduce the memory requirement, we adopt a multirate implementation. The resulting structure, however, is not shift-invariant. We introduce a design criterion for multirate deformable structures that jointly controls the approximation error and the shift-variance.

1 Introduction

Elementary visual structures such as lines, edges, texture, motion, are powerful "cues" to understand the structure of the outside world from its visual appearance (the image), and their identification is instrumental for almost any visual task. Classical image processing problems (enhancement, denoising) may also be approached successfully using elementary descriptors such as edges and textures. Image compression schemes using sub–bands coders, oriented along the preferred texture orientation in the image, proved advantageous in terms of visual rendition [1]. Velocity may be interpreted as orientation in the spatio–temporal domain, and motion-compensated spatio-temporal

*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. The author R. Manduchi is currently with Apple Computer, Inc., Cupertino, CA.

filters may be used succesfully for prediction, interpolation and smoothing [2] as well as for coding [3].

Regardless of the specific descriptor of interest, most techniques start processing the image (or image sequence) with a family of linear filters tuned at a wide range of orientations and scales of resolution (see [4] for an extensive bibliography on the subject). The multiscale/multiorientation image decomposition is then analyzed to detect features (usually, via a non-linear stage) and to measure their attributes (orientation, dominant scale, velocity).

While orthogonal structures have been intensively studied in the context of wavelet theory, several algorithms are designed to operate on redundant (or overcomplete) image decompositions. A drawback of this approach is that the computational cost to realize the analysis filter bank may easily become too high for practical use. In order to meet prescribed implementation constraints, thus, the use of fast filtering is mandatory.

This paper describes efficient schemes for the computation of a large number of differentely scaled/oriented filtered versions of an image, pushing forward previous results by Freeman and Adelson [5] and Perona [4]. To reduce the computational weight of multiscale/multiorientation filter banks, one may exploit the correlation among the filters in a two-stage scheme usually named "steerable" or "scalable" [5, 4] (or more generally, "deformable" [4]). We point out that such systems are particular instances of the "multistage separable" structure introduced by Treitel and Shanks [6]; this observation leads us to consider deformable filter bank structures with X-Y separable basis filters. Our implementation reduces effectively the computational weight associated with deformable filter banks. An iterative projections technique is used for the least-squares design of such structures.

While deformable filter banks are efficient in terms of computational cost, they require substantial extra memory to store intermediate frames. To reduce the overall memory requirement, one may use a multirate implementation: for those filters in the filter bank that are narrow-banded, only a subsampled version of the intermediate filtered frames needs to be stored.

In our implementation, we embed the filter bank in an analysis/synthesis separable pyramidal scheme. Such a structure, unfortunately, is not shift-invariant. Shift-invariance is usually considered a fundamental property of signal processing systems; on the other side, a system that deviates "slightly" from shift-invariance may still be suitable to most applications. In fact, if our task is the analysis of the visual structures in an image, aliasing may be tolerated as a signal–dependent noise, qualitatively not too different from other forms of "noise" that are traditionally considered.

This notion is at the basis of our novel least-squares design procedure for multirate FIR filters. Such a technique relies on the definition of a suitable "multirate" approximation criterion, which jointly controls the approximation error and the shift-variance. We extend this procedure to the design of our pyramidal deformable filter banks, achieving the result of joint reduction of computational weight and of memory requirement.

The paper is organized as follows. Section 2 describes the least-squares design of X-Y multistage

separable deformable filter banks. Section 3 introduces our algorithm to design least–squares multirate FIR filters, and describes the design of pyramidal separable deformable filter banks. Section 4 has the conclusion.

2 Separable deformable filter banks

We consider FIR analysis filter banks as in Figure 1. If $N(\sigma, \theta)$ is the size of filter $d(\mathbf{x}, \sigma, \theta)$ (where σ represents the scale and θ represents the orientation), in general $\sum_{\sigma \in S} \sum_{\theta \in T} N(\sigma, \theta)$ elementary operations (sums or multiplications) per input pixel (OPPs) are required to implement the overall multiscale/multiorientation system of Figure 1 in a direct form realization. On the other side, some classes of multidimensional FIR filters can be realized with a smaller number of OPPs. In particular, if we are interested in analysis filter banks that represent the rotated version of a prototype kernel, we may exploit the idea of *steerable filters*, introduced by Freeman and Adelson [5], and developed further by Perona [4]. A steerable filter $d(\mathbf{x}, \theta)$ is such that it can be decomposed as (see Figure 1)

$$d(\mathbf{x}, \theta) = \sum_{r=0}^{R-1} d_r(\mathbf{x}) t_r(\theta) \tag{1}$$

We will call the filters $d_r(\mathbf{x})$ *basis filters* and the functions $t_r(\theta)$ *recombination functions*. Term R is called the *rank* of the decomposition. Perona [4] studied the conditions for a kernel $d(\mathbf{x}, \theta)$ (with continuous $\mathbf{x}$ and θ) to be exactly steerable.

One easily shows [7] that steerable filters are formally equivalent to multistage X-Y separable structures, introduced by Treitel and Shanks [6]. As in the case of multistage X-Y separable filters, the least–squares design of steerable filter banks [4] is performed by computing the SVD of a matrix built from the kernel $d(\mathbf{x}, \theta)$, where each row of the matrix corresponds to an orientation θ (similar considerations apply to the case of *scalable* filter banks [4]).

The computational savings using the discrete steerable scheme may be computed as follows (the case of scalable filters is analogous). The "direct" implementation requires NN_θ OPPs, where N is the size of the support of $d(\mathbf{x}, \theta)$ (assumed constant with θ), and N_θ is the number of orientations. Using the steerable implementation, R convolutions are computed once for all, then R multiplications (and $R - 1$ additions) per pixel are required for each orientation. The overall number of OPPs does not exceed $R(N + N_\theta)$, which is smaller than NN_θ for sufficiently small values of R. The price to pay is that we need to store R filtered version of the image, to be recombined for each orientation.

2.1 Least-squares design of separable deformable filter banks

The idea of multistage separability may be extended to higher dimensions, imposing X-Y separability on the basis filters of the steerable/scalable filter banks. In the case of steerable filter banks,

we consider decompositions of the type

$$\sum_{r=0}^{R-1} u_r(x) v_r(y) q_r(\sigma, \theta) \tag{2}$$

(see Figure 3). We will call this structure a *separable steerable filter bank*. The extension to scalable structures (to approximate rotated and scaled kernels) is straightforward.

The approximation problem we attack can be formalized as follows: given the set of kernels $\{d(x, y, \theta), x \in X, y \in Y, \theta \in T\}$, where X, Y, T are discrete (and finite) sets, and given the decomposition index (or rank) R, find the 1-D kernels $\{u_r(x), v_r(y), 0 \leq r < R\}$ and recombination functions $\{t_r(\theta), 0 \leq r < R\}$ that minimize the quadratic norm of the error

$$e(x, y, \theta) = d(x, y, \theta) - \sum_{r=0}^{R-1} u_r(x) v_r(y) t_r(\theta) \tag{3}$$

While in the case of 2-D multistage separable structures (e.g., multistage X-Y separable filters or non-separable steerable filter banks) the approximation task is solved using the SVD of the kernels' matrix, now we find ourselves dealing with more complex structures. In fact, the minimization task turns out to be a trilinear approximation problem, for which the 2-D SVD technique is not appropriate.

On the other side, it is easily seen that, if two variables (say, $\{u_r(x)\}$ and $\{v_r(x)\}$) are kept fixed, then minimizing the approximation error norm is a linear problem the third variable (here $\{t_r(\theta)\}$). This observation suggests the following iterative procedure: keep two variables fixed and minimize for the third one; then, cycle for the other variables. The approximation error is guaranteed to lower at each step, and since it is bounded from below by zero, the algorithm converges to a minimum. In [7] we use the formalism of hypermatrix algebra [8] and Kronecker algebra to translate the approximation task into a sequence of simple matrix operations.

Unfortunately, one cannot be sure that the minimum found this way will be global (the error surface being in general not convex), and the solution will depend on the starting point. Experimental tests and strategies for the selection of the starting point and for termination criteria are discussed in [7].

2.1.1 Experimental tests

We consider the separable steerable/scalable decomposition of a family of rotated and scaled kernels. The continuous prototype kernel is the separable function $\bar{d}(x, y) = \bar{d}_1(x)\bar{d}_2(y)$, where $\bar{d}_1(x)$ is the second derivative of a gaussian with standard deviation σ and $\bar{d}_2(y)$ is a gaussian with standard deviation $\sigma_2 = 3\sigma$ (filters with similar elongate shape are widely used in computer vision to achieve high orientation selectivity). The orientation θ is sampled in N_θ=20 points for $0 \leq \theta < 180°$, while the scale σ, ranging from 1 to 4 (two octaves), is sampled logarithmically on N_σ=12 points. The continuous kernels are sampled on Z^2 and windowed within a square of side N=N_x=N_y=35.

Each scaled/oriented kernel is normalized to unitary energy. The results for R=30 are shown in Figures 4 – 6. Two observations may be drawn from the experimental data:
1) Our algorithm designs basis kernels all with the same size. Observing Figure 4, though, it results clear that some basis kernels may be conveniently windowed to a smaller support without introducing relevant additional approximation error, and with reduced computational cost. The design of deformable filter banks with inhomogeneous kernels' sizes is described in [7]. However, note that:
2) For a sufficiently high number of scales/orientations, the computational weight due to the basis filters becomes negligible with respect to that of the recombination. In fact, the ratio between the OPPs for the two operations is approximately equal to $2N/N_\sigma N_\theta$.

3 Multirate deformable filter banks

Imposing X-Y separability on the basis filters of a deformable filter bank reduces the overall computational weight, at the price of supplementary memory for storing the intermediate filtered images. Frame memories are expensive, and it is desirable to reduce the memory requirement while enjoying reduced computational cost.

A solution is derived observing that some of the basis filters may be narrow-banded. The filtered signals are thus highly correlated, and may be subsampled before storing (therefore using less memory). These reduced rate versions are then interpolated and linearly recombined for each scale and orientation. Although this may seem computationally expensive at first, we show that with a suitable choice of the interpolator filters, the overall implementation results efficient both in terms of memory and of computational cost.

In the case of separable deformable filter banks, we adopt the multirate implementation for those 1-D basis filters that are suitably narrow-banded, and jointly optimize for the multirate 1-D filters and for the recombination functions. In particular, we use a pyramidal octave–band separable structure, represented in Figure 13.

3.1 Least squares design of multirate FIR filters

The general structure of a M-fold multirate FIR filter [9] is shown in Figure 8. Term M is the *multirate order* of the multirate filter. To analyze this system, the z–transform notation results handy. We will indicate sequences in the space (time) domain with lower case letters, and their z–transform with the corresponding upper case letters. In the case of Figure 8, $G(z)$ and $Q(z)$ are two FIR filters, interconnected via the cascade of an M-fold decimator followed by an M-fold interpolator.

The presence of decimator/interpolator blocks in the scheme allows us to use polyphase implementation to reduce the computational weight. For example, if M=2, we may redraw the scheme of Figure 8 using the Type 1 polyphase decomposition [10] of the kernels involved: here,

$G(z) = G_0(z^2) + z^{-1}G_1(z^2)$, where $g_0(x) = g(2x)$, $g_1(x) = g(2x+1)$. We thus obtain the scheme of Figure 9, where all the filters work on reduced sampling rate, and therefore, with respect to the scheme of Figure 8, the number of OPPs is reduced by one half. However, unless the filters are perfectly band-limited, the system is linear periodically time-variant (LPTV).

Rabiner and Crochiere [9] designed multirate filters with brick-wall frequency response using a minimax error criterion in the frequency domain. These filters are typically used for sampling rate conversion of acoustic signals. In our case, though, we are not interested in sharp low–pass responses, and the minimax criterion of [9] is not adequate for our applications. Instead, we propose a novel least-squares criterion for multirate systems based on the time domain, which explicitly keeps into account the shift–variance. For our following discussions, it is important to notice that the least–squares design of deterministic linear filters in the temporal domain has the following operatorial interpretation: approximating a target convolution kernel $d(x)$ with a kernel $u(x)$ in the least-squares sense is equivalent to minimizing the maximum punctual deviation from the desired output for all unit norm input signals [11].

Consider the multirate scheme of Figure 8, where, for simplicity's sake, we set M=2. As noted earlier, the system is LPTV, and therefore it is characterized by 2 impulse responses, corresponding, for instance, to inputs $\delta(x)$ and $\delta(x+1)$ (where $\delta(x)$ is the discrete impulse). Let $t^{(0)}(x)$ and $t^{(1)}(x+1)$ be such impulse responses. It is easily shown that

$$t^{(0)}(x) = q * \bar{g}_0(x) \ , \ t^{(1)}(x+1) = q * \bar{g}_1(x) \tag{4}$$

where $\bar{g}_0(x)$ and $\bar{g}_1(x)$ are obtained by interleaving $g_0(x)$ and $g_1(x)$ with null–valued samples.

If $t^{(0)}(x) = t^{(1)}(x)$, the system is shift–invariant, and it operates a convolution with a (fixed) kernel $t^{(0)}(x)$. Otherwise, inputting a signal $l(x)$, the output at point x_0 will be

$$y(x_0) = \begin{cases} \langle l(x), p^{(0)}(x - x_0) \rangle & , \quad \text{even } x_0 \\ \langle l(x), p^{(1)}(x - x_0) \rangle & , \quad \text{odd } x_0 \end{cases} \tag{5}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, and

$$p^{(0)}(x) = \begin{cases} t^{(0)}(x) & , \quad \text{even } x \\ t^{(1)}(x) & , \quad \text{odd } x \end{cases} , \ p^{(1)}(x) = \begin{cases} t^{(1)}(x) & , \quad \text{even } x \\ t^{(0)}(x) & , \quad \text{odd } x \end{cases} \tag{6}$$

Minimizing the maximum punctual deviation from the desired output, in this case, is equivalent to the following task:

$$\text{Minimize } \max \{ \|p^{(0)}(x) - d(x)\|^2, \|p^{(1)}(x) - d(x)\|^2 \} \tag{7}$$

This problem, unfortunately, is not in a least–squares formulation, and designing the optimal filters $g(x)$ and $q(x)$ for (7) involves a non–linear procedure. We may then consider the following sub–optimal task:

$$\text{Minimize } \mathcal{E}^2 = \frac{\|p^{(0)}(x) - d(x)\|^2 + \|p^{(1)}(x) - d(x)\|^2}{2} \tag{8}$$

To better understand the meaning of the proposed error criterion, let us translate it in terms of $t^{(0)}(x)$ and $t^{(1)}(x)$. Using (6), we have that

$$2\ \mathcal{E}^2 = \|t^{(0)}(x) - d(x)\|^2 + \|t^{(1)}(x) - d(x)\|^2 \tag{9}$$

The following upper bound holds:

$$\|t^{(0)}(x) - t^{(1)}(x)\|^2 \leq 2(\mathcal{E}^2 + \|t^{(0)}(x) - d(x)\|\|t^{(1)}(x) - d(x)\|) \tag{10}$$

Now, using (4), one can show that the minimization task is multilinear in $g_0(x)$, $g_1(x)$ and $q(x)$. An iterative algorithm to minimize form (9), similar in spirit to the technique of Section 2.1, is described in [7].

The minimization of form (8) has an interesting counterpart in the frequency domain. Using Parseval's equality, one can show that (8) is equivalent to the following task:

$$\text{Minimize } \int_0^{2\pi} \left| D\left(e^{j\omega}\right) - Q\left(e^{j\omega}\right) G\left(e^{j\omega}\right)/2 \right|^2 d\omega + \int_0^{2\pi} \left| Q\left(e^{j\omega}\right) G\left(e^{j(\omega+\pi)}\right)/2 \right|^2 d\omega \tag{11}$$

The first term of (11) represents the deviation from the target response, assuming the two filters are correctly bandlimited; the second term measures the residual spectral overlapping with the repetition of $G\left(e^{j\omega}\right)$ introduced by the inner decimation.

3.2 Pyramidal implementation of deformable filter banks

As mentioned earlier, we consider a slightly less general multirate scheme than in Figure 8; in fact, we embed our system in a pyramidal structure. The choice of a pyramidal scheme has been inspired by the inherentely self-similarity of the scaled kernels to be approximated.

The basic multirate structure is represented in Figure 10. The two systems are equivalent if we set $G(z) = H(z)$ and $Q(z) = H(z)U(z^2)$, or $G(z) = H(z)U(z^2)$ and $Q(z) = H(z)$. The overall pyramidal structure is shown in Figure 12 for the 1-D scalable case, and Figure 13 for the 2D case. In particular, we set the decimation filter equal to the interpolator, and keep it constant for all branches in the overall deformable structure.

In the scheme of Figure 12, the basis filters are characterized by two indices (i, j): index i signals the level (or *depth*) in the pyramid (i=0 means no inner rate alteration), while index j enumerates the filters in the same level. A basis filter at depth i is actually a multirate filter of multirate order 2^i, for which the transfer function of the decimator/interpolator is equal to $\prod_{l=1}^{i} H(z^{2l})$.

Any filter suitable for pyramidal decomposition is a candidate for $h(x)$, so long as it can be implemented with few OPPs. This is a fundamental requirement because we store the basis filters' outputs in their decimated versions, for memory parsimony. For each scale and orientation, we interpolate back these signals before recombination.

In our experiments, we have used a simple 3-taps raised-cosine filter $h(x) = [a, 1, a]$ for the decimator/interpolator (the optimization of $h(x)$ is discussed in [7]). To interpolate of a factor 2

using the polyphase implementation of Figure 11, we need only 0.5 multiplications and 0.5 sums per output sample.

To appreciate the reduction of the computational burden for the recombination, consider a 1-D scalable filter bank where all R basis filters are implemented in a 2-multirate scheme. For each scale/orientation, we need 0.5 (for interpolation) plus $0.5R$ (for recombination) multiplications per input sample. A comparison with the non-multirate case (where the reconstruction requires R multiplications per input sample per scale/orientation) shows that, even if the multirate implementation may require to increase the filter bank's rank to compensate for larger approximation error, it makes for the substantial reduction of the computational cost associated with the basis filtering and the recombination.

The problem of the *a priori* selection of the multirate order and kernel's size for each basis filter is discussed in [7], where we also derive an iterative technique for the least-squares design of pyramidal deformable filter banks. In brief, with respect to the case of simple separable deformable filter banks, one has to consider the effect of the decimators/interpolators, and of the fact that now the basis filters are characterized by a variable number of impulse responses.

Our pyramidal structure can be extended to the 2-D case, for steerable/scalable decompositions. The most general scheme would consider non-separable subsampling lattices [10], however we restrict ourselves to the case of separable lattices and filters (see Figure 13). Note that, if I_x and I_y are the maximum depths of the filters in X and in Y respectively, the system is characterized by $2^{(I_x+I_y)}$ impulse responses.

4 Conclusion

We have described a technique for the efficient implementation of deformable filter banks, based on the use of separable basis filters and multirate implementation. The filter bank is embedded in a pyramidal structure, designed under a novel multirate error criterion that jointly minimizes the approximation error and the shift-variance. The computational weight and memory requirement may be substantially reduced with respect to the common steerable/scalable decomposition based on SVD.

The Matlab software to implement the filter banks described in this paper may be found at `http://www.vision.caltech.edu/manduchi/def.tar.Z`.

References

[1] A. Ikonomopoulos and M. Kunt. High compression image coding via directional filtering. *Signal Processing*, 8(2):1823–1826, April 1985.

[2] E. Dubois. Motion–compensated filtering of time–varying images. *Multidimensional Systems and Signal Processing*, 3:211–239, 1992.

[3] E. Chang and A. Zakhor. Subband video coding based on velocity filters. In *Proc. IEEE Int. Symp. Circuits Syst.*, volume 5, pages 2288–2291, San Diego, 1992.

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

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

[6] S. Treitel and J. Shanks. The design of multistage separable planar filters. *IEEE Trans. Geosci. Electron.*, 9(1):10–27, 1971.

[7] Perona P. Manduchi, R. and Shy D. Efficient implementation of deformable filter banks. Technical Report CNS-TR-97-04, California Institute of Technology, 1997. Also available at `http://www.vision.caltech.edu/manduchi/deformable.ps.Z`.

[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] 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.

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

[11] T.W. Parks and D.P. Kolba. Interpolation minimizing maximum normalized error for band–limited signals. *IEEE Trans. Acoust., Speech, Signal Processing*, 26(4):381–384, August 1978.

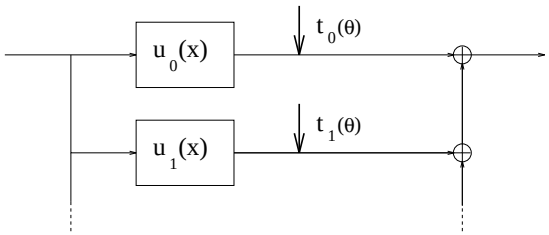


Figure 1:

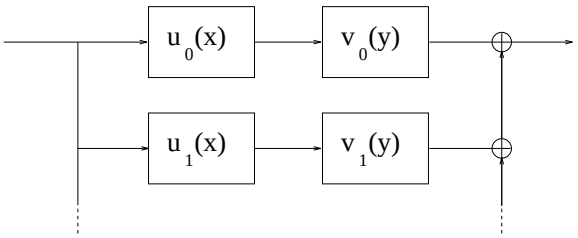


Figure 2:

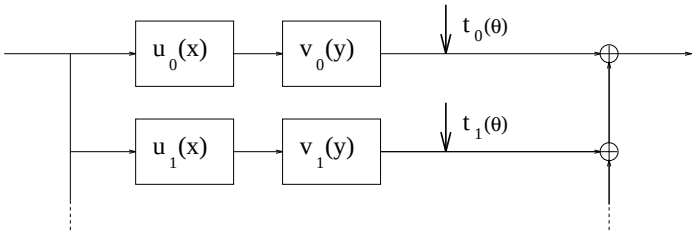


Figure 3:

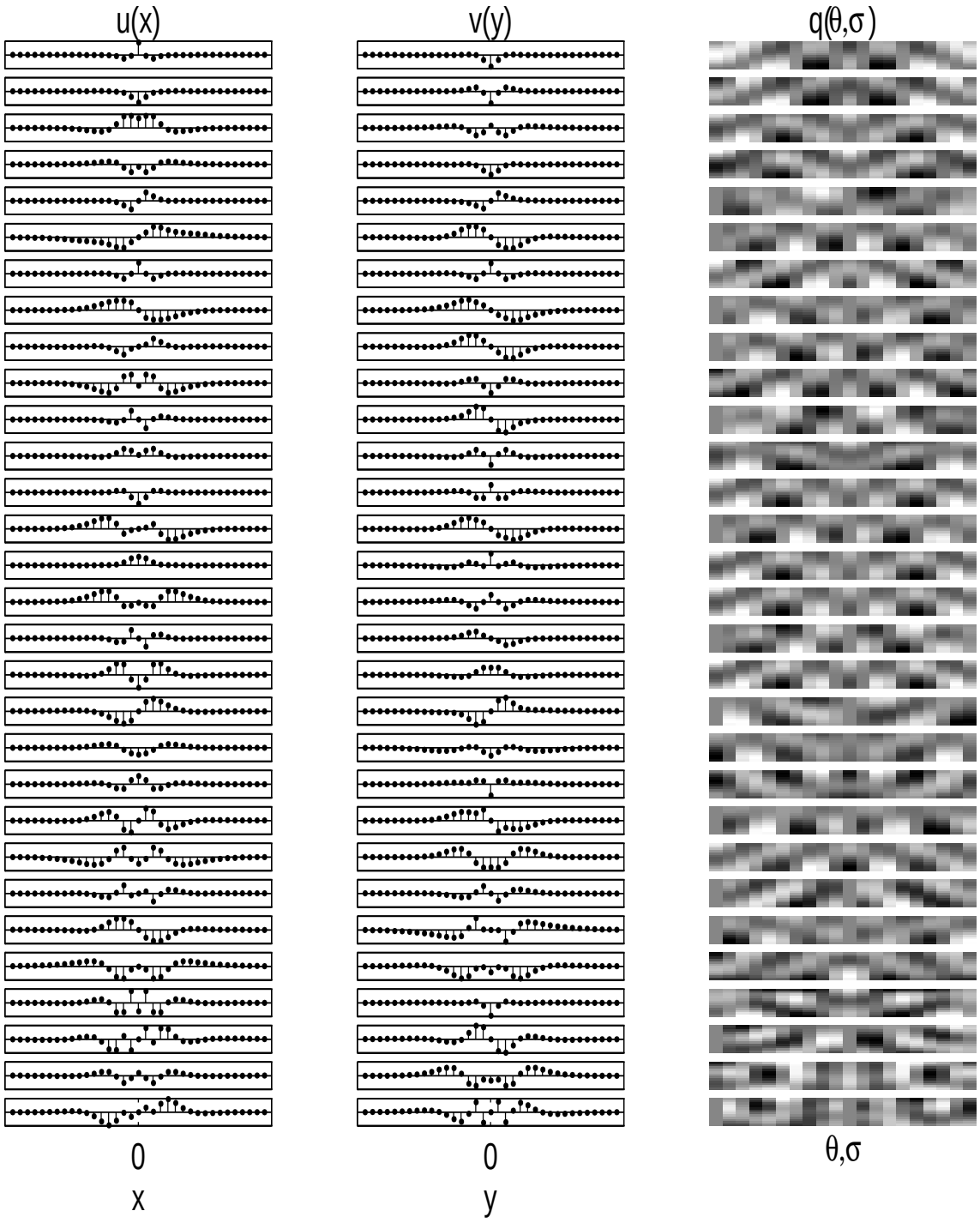


Figure 4:

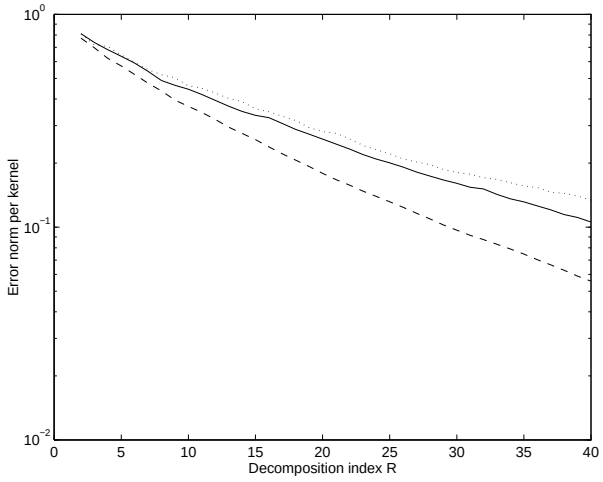


Figure 5:

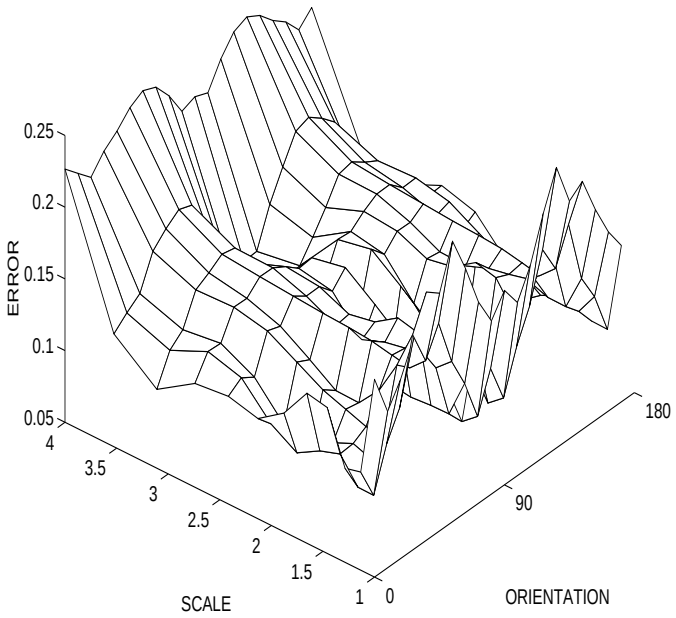
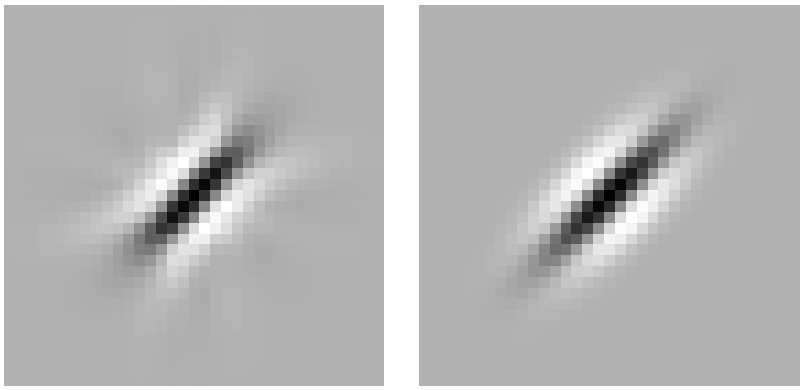


Figure 6:



(a) (b)

Figure 7:

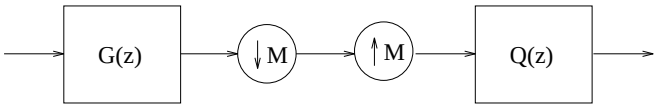


Figure 8:

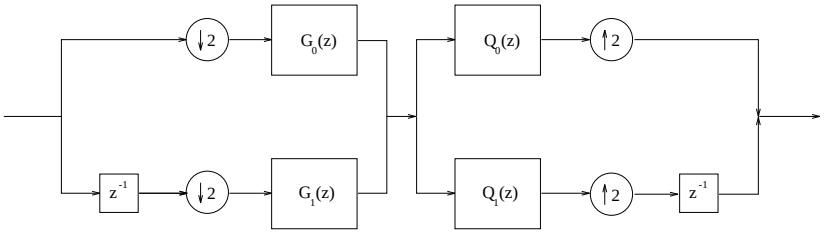


Figure 9:

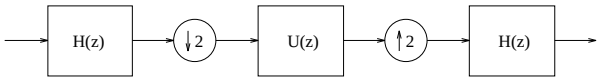


Figure 10:

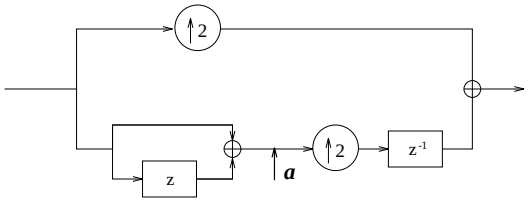


Figure 11:

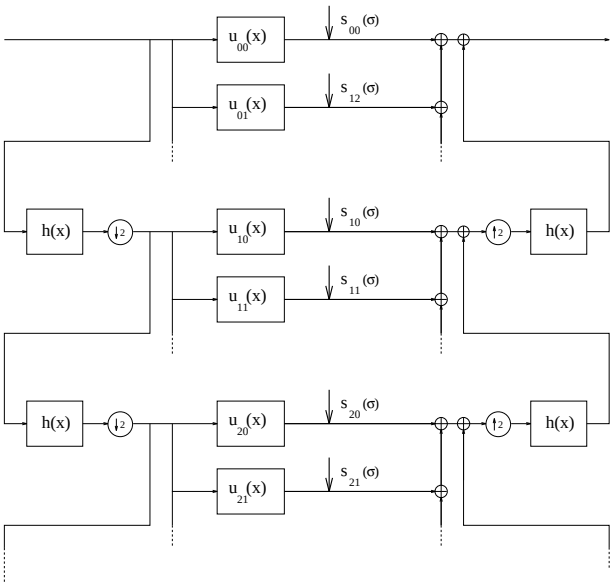


Figure 12:

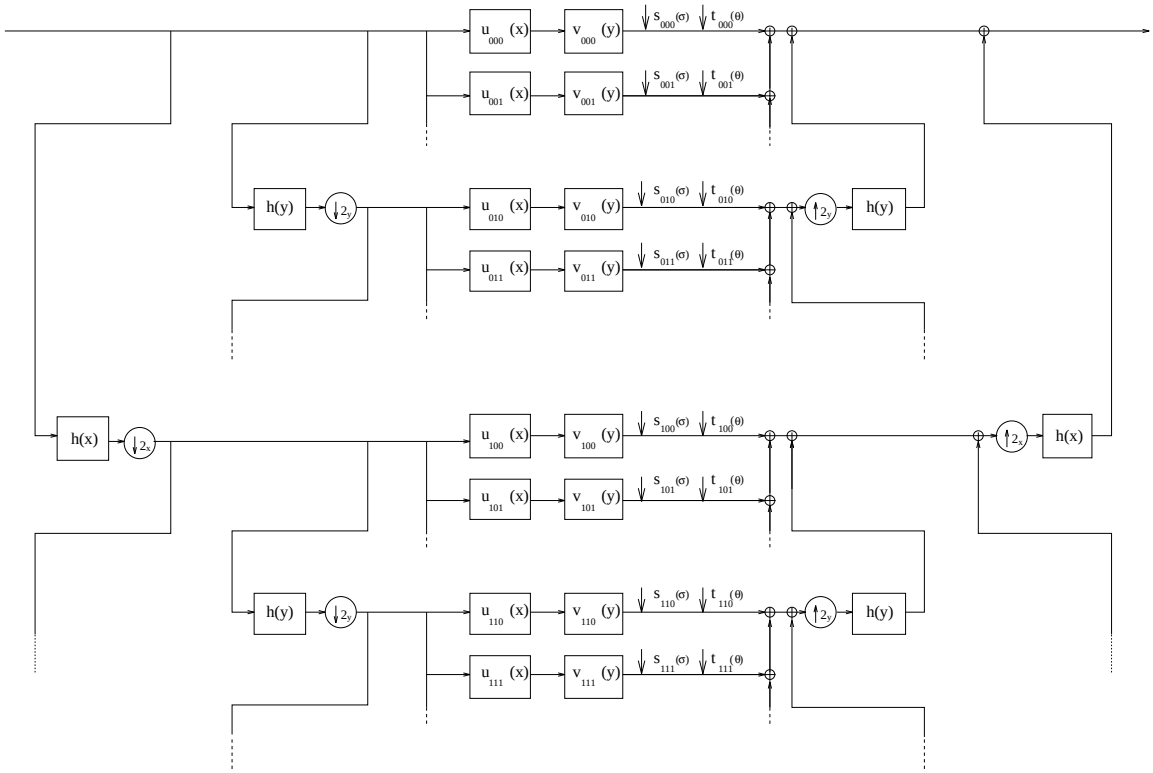


Figure 13: