

## Filter Networks

Mats Andersson Johan Wiklund Hans Knutsson  
 Computer Vision Laboratory, Department of EE  
 Linköping University, SE-581 83 Linköping, Sweden  
 phone: +46 13 282650 fax: +46 13 138526 email: matsa@isy.liu.se

### Abstract

This paper presents a new and efficient approach for optimization and implementation of filter banks e.g. velocity channels, orientation channels and scale spaces.

The multi layered structure of a filter network enable a powerful decomposition of complex filters into simple filter components and the intermediary results may contribute to several output nodes. Compared to a direct implementation a filter network uses only a fraction of the coefficients to provide the same result. The optimization procedure is recursive and all filters on each level are optimized simultaneously. The individual filters of the network, in general, contain very few non-zero coefficients, but there are no restrictions on the spatial position of the coefficients, they may e.g. be concentrated on a line or be sparsely scattered.

An efficient implementation of a quadrature filter hierarchy for generic purposes using sparse filter components is presented.

**keywords** filter optimization, filter network, sequential convolution, sparse filters, efficient filtering.

### 1 Introduction

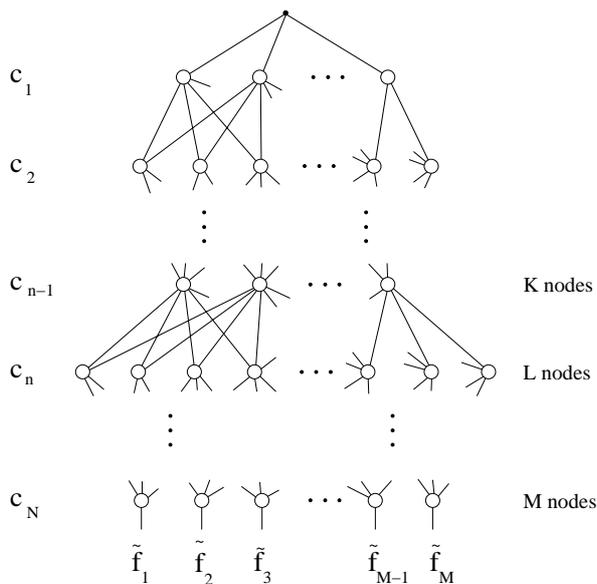


Figure 1: A general filter network of  $N$  levels with  $M$  nodes in the final level.

Figure 1 depicts a general filter network consisting of  $N$  levels and with  $M$  output nodes. The nodes constitute summation points and the filters are located on the arcs connecting two consecutive layers. In the general case all nodes on layer  $(n - 1)$  are connected to each node in layer  $n$ . The definition of the filter network can be divided into structure and internal properties.

The **structure** of the network are the properties that can be read out from a sketch like in fig. 1 i.e. the number of levels and the number of nodes on each level.

The **internal properties** comprise the number of coefficients for each filter in the network and their spatial position (coordinates). Moreover we need to define the ideal filter function  $f_m(\mathbf{u})$  and weight functions  $W_m(\mathbf{u})$  for the final layer, where  $\mathbf{u}$  are Fourier domain coordinates and  $m = [1 \dots M]$  the number of output nodes.

The task is to compute the filter coefficients in the network such that the weighted difference between the resulting and the ideal filter functions are minimized.

$$\min \epsilon^2 = \sum_{m=1}^M \left\| W_m(\mathbf{u}) \left( \tilde{f}_m(\mathbf{u}) - f_m(\mathbf{u}) \right) \right\|^2 \quad (1)$$

The *Fourier weighting function*  $W(\mathbf{u})$  provide an appropriate Fourier space metric. The metric will determine the importance of a close fit for different spatial frequencies, see [7].

### 2 Fourier space metric

The choice of  $W(\mathbf{u})$  should if possible be done in the light of the expected spectra for the signal and the noise. If no a priori information is available eq. 2 is proposed.

$$w(\mathbf{u}) = \rho^{-\alpha} + c \quad 0.5 \leq \alpha \leq 1.5 \quad (2)$$

This form of  $W(\mathbf{u})$  is due to two observations. For natural images there is, in general, no reason to expect a non-isotropic spectrum, i.e. the expected spectrum will only depend on the Fourier domain radius,  $\rho = \|\mathbf{u}\|$ . Secondly, there does not seem to be a large difference in terms of spectrum when imaging the real world at very different scales, say a microscope image vs a satellite image. Consequently the spectrum is approximately scale invariant,  $\rho^{-\alpha} \propto (a\rho)^{-\alpha}$ . The range of the parameter  $\alpha$  may be argued from the spectrum of randomly scattered lines and edges, see [7]. The constant  $c$  relates to the expected level of broadband noise and aliasing. Note that optimizing

without using a weighting function is equivalent to setting  $w(\mathbf{u}) = 1$ .

### 3 Filter network optimization

Ideally it would for a given network structure be desirable to optimize the number of coefficients in each arc and the position (coordinate) for each coefficient for the entire net simultaneously. This is, however, an extremely complex problem and a method for finding an overall optimal solution has not been found. (and it's doubtful if it ever will be). For this reason the definition of the internal properties of the network have to be made based on experience.

If the internal properties (the non zero coordinates of the filters) is defined it is possible to optimize all filters on the *same level* simultaneously with respect to the output filters and the current state of the network. This procedure is repeated for another layer and so on creating a sequential optimizer loop over the network structure.

The convergence of this approach cannot be guaranteed but initial tests prove that for realistic choices of filter structure and coefficient distribution the network converges to a stable solution which in terms of computational complexity outperforms a conventional implementation by orders of magnitude.

#### 3.1 Minimizing the error

The network is initialized to provide a non zero transfer function for each arc. The initialization is not critical but may require some consideration. Usually a qualified guess or setting the transfer functions to unity in the FD is sufficient. Pure randomizing is not always recommended as it may introduce unwanted symmetry effects. If a filter in the network is expected to be e.g. real and even there is no reason not to initialize the filter by such a function.

For each kernel (arc) in the network, a Fourier transform matrix  $B_{nkl}$  is computed such that the Fourier transform of the filter connecting node  $k$  on layer  $(n - 1)$  to node  $l$  on layer  $n$  is computed as  $B_{nkl} c_{nkl}$ . The column vector,  $c_{nkl}$ , contains the nonzero coefficients of the filter in this arc of the network. The rows of  $B_{nkl}$  contain the corresponding Fourier basis functions.

In order to minimize eq. 1 select a layer,  $n$ , for the current optimization step. For the moment  $n$  is assumed to be one of the intermediate layers i.e.  $n = [2 \dots N - 1]$ . The number of nodes in layer  $(n - 1)$  and  $n$  is  $K$  and  $L$  respectively, see fig. 1. Now compute the transfer functions,  $G_k(\mathbf{u})$   $k = [1 \dots K]$  from the top node to each of the nodes in layer  $(n - 1)$ . To simplify the subsequent notation,  $G_k(\mathbf{u})$  is reshaped as a diagonal matrix where the transfer function is located in the main diagonal and all off-diagonal values are zero. For the lower part of the network the transfer functions from each node in layer  $n$  to each of the final nodes are computed as:

$$\begin{aligned} H_{lm}(\mathbf{u}) \quad l &= [1 \dots L] \\ m &= [1 \dots M] \end{aligned} \quad (3)$$

For the same reason as above  $H_{lm}(\mathbf{u})$  is reshaped into diagonal form. For convenience, the frequency coordinate  $\mathbf{u}$ , is dropped from here on.

As a final step towards expressing eq. 1 in terms of the kernel coefficients of layer  $n$  we define:

$$\mathcal{B}_m = (G_1 H_{1m} B_{n11}, G_2 H_{1m} B_{n21}, \dots, G_k H_{lm} B_{nkl}, \dots, G_K H_{LM} B_{nKL}) \quad (4)$$

where

$$k = [1 \dots K] \quad l = [1 \dots L] \quad m = [1 \dots M]$$

Equation 1 can now be expressed as:

$$\min \epsilon^2 = \sum_{m=1}^M \|W_m (\mathcal{B}_m \mathbf{c}_n - f_m)\|^2 \quad (5)$$

where  $\mathbf{c}_n$  is a vector containing all filters connecting layer  $(n - 1)$  and layer  $n$ .

$$\mathbf{c}_n = (c_{n11}, c_{n21}, \dots, c_{nkl}, \dots, c_{nKL})^T \quad (6)$$

Compute the partial derivatives

$$\frac{\partial \epsilon^2}{\partial \mathbf{c}_n} \equiv \frac{\partial \epsilon^2}{\partial \text{Re}[\mathbf{c}_n]} + i \frac{\partial \epsilon^2}{\partial \text{Im}[\mathbf{c}_n]} \quad (7)$$

Setting the partial derivative equal to zero results in

$$\mathbf{c}_n = \mathbf{A}^{-1} \mathbf{h} \quad (8)$$

where

$$\mathbf{A} = \sum_{m=1}^M \mathcal{B}_m^T W_m^2 \mathcal{B}_m \quad \mathbf{h} = \sum_{m=1}^M \mathcal{B}_m^T W_m^2 f_m \quad (9)$$

Note that the complexity involved in solving this linear equation system is only dependent on the number of nonzero coefficients in the filters connecting layer  $(n - 1)$  and  $n$ .

The kernels  $c_{nkl}$  of layer  $n$  are updated and the procedure is repeated for another layer until convergence. For the initial layer the frequency transfer function  $G$  degenerates to an identity matrix. The frequency transfer function  $H$  will in the same way be an identity operator in the optimization of the final layer.

#### 3.2 Fourier space sampling

The introduction of the Fourier transform matrices,  $B_{nkl}$ , in the previous section implies a sampling of the continuous Fourier space. In principle the higher the sampling density the 'closer' the sampled case solution will be to the continuous case. In practise using 2-3 times as many points, for each dimension, as the spatial size in pixels (voxels etc.) has proven to be adequate. Note that the number of samples does not change the size of the basic problem, i.e. the size of the matrix  $\mathbf{A}$  in eq. 9. However, further increasing the sample density will, as a rule, have an insignificant effect on the solution. As the Fourier space representation is repetitive samples are only needed in the interval  $-\pi/\Delta < u \leq \pi/\Delta$  where the inter-sample distance is given by  $\Delta$ .

### 3.3 A note on recursive optimization

As the optimizer considers one level of the network at each optimization step it may be worthwhile to consider this restriction in the initialization of the network and when deciding the order in which to optimize the layers of the network. In general it is recommended to start with a very simple network containing only the most necessary nodes and a small number of kernel coefficients. When developing a network it is equally important to remove arcs and coefficients that are not being used satisfactorily as to add new degrees of freedom.

If complex valued filters are present in the network these filters should, if possible, be placed close to the output nodes to minimize the number of complex multiplications. A real valued solution for a layer may be enforced by solving

$$\mathbf{c}_n = (\text{Re}[\mathbf{A}])^{-1} \text{Re}[\mathbf{h}] \quad (10)$$

which is obtained by

$$\frac{\partial \epsilon^2}{\partial \text{Re}[\mathbf{c}_n]} = 0 \quad (11)$$

As an example the network in fig. 3 converges after 10-15 iterations.

## 4 Examples

Filters do for natural reasons need a region of support of the same dimensionality as the signal. A careful combination of simple kernels in a filter network will in relation to a direct implementation be more efficient for signals with high outer dimensionality (3D and 4D data). The filter net example below is, however, limited to 2D to simplify visualization of the result. Note that the complexity of the optimization is only dependent on the total number of coefficients within the network and not the corresponding coordinates (i.e. the dimensionality of the resulting filters). The network optimizer is developed and implemented in Matlab 5.

### 4.1 A network for logarithmic quadrature channel decomposition of the FD

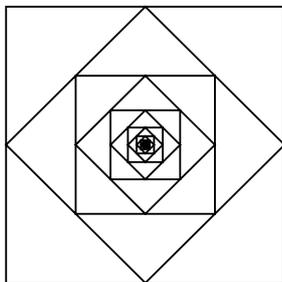


Figure 2: The proposed logarithmic decomposition of the Fourier domain.

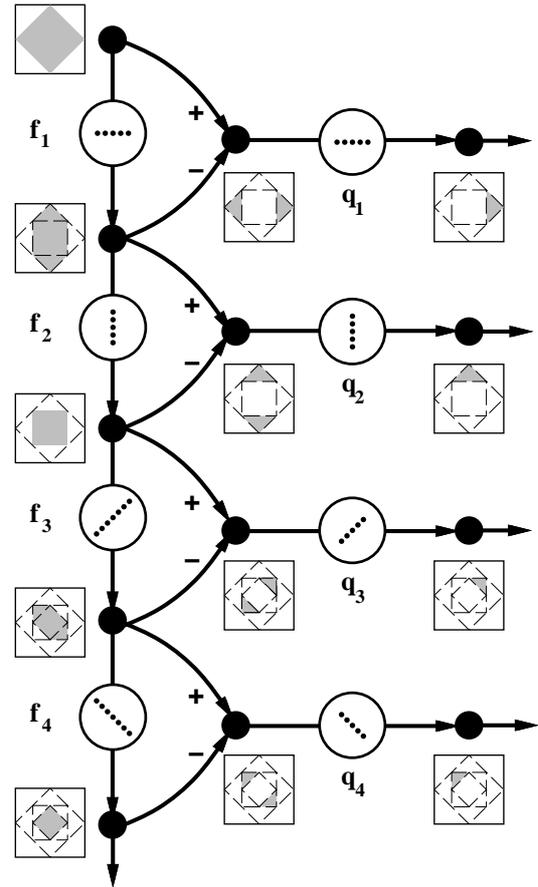


Figure 3: Filtering structure for one level of quadrature channel network.

This filtering structure was originally intended for efficient computation of quadrature filter responses in different orientations and scales for phase based optical flow. The network is however equally useful for a number of purposes such as e.g orientation estimation, enhancement, subband coding, local spectrum analysis etc. A schematic illustration of the quadrature channels is given in fig. 2. The network produce a logarithmic decomposition of the FD i.e. the relative bandwidth of the quadrature channels is constant.

Figure 3 illustrates the structure for one level of such a network. If this network is compared to the general network of fig. 1 a number of conditions are not fulfilled. The output nodes are not localized in the final layer and the arcs are not restricted to connect two subsequent layers. It is, however, straightforward to redraw the network of fig. 3 to meet the conditions in section 1 by introducing a number of dummy nodes, (in fact this was the way the network was optimized).

The large black dots in fig. 3 indicate the nodes (summation points) while the filtering is performed in the arcs. The net contain eight very simple filters, four real valued LP-filters  $[f_1, \dots, f_4]$  and four complex valued quadrature filters  $[q_1, \dots, q_4]$ . The number of coefficients and their corresponding coordinates (i.e. filter orientation) is illustrated by the dots on each arc. The square containing grey shaded

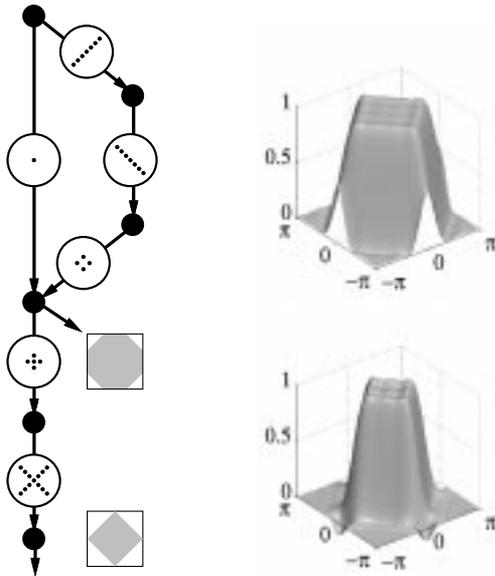


Figure 4: Initial filtering, two passbands separated by an half octave.

areas illustrate the energy contribution in the FD at each node.

Assume for the moment that the spectrum of the signal at the top node in fig. 3 only has energy contributions inside the shaded diamond, i.e. band limited to  $\pi$  along the diagonal directions. The filter  $f_1$  band limits the signal further in the horizontal direction which is illustrated at the next node. The difference before and after  $f_1$  is an even BP-channel that is fed to a quadrature filter  $q_1$  creating quadrature response in the horizontal direction. The vertical quadrature filter response is computed in the same way using  $f_2$  and  $q_2$  containing 5 coefficients each.

At this stage in the network LP-chain there are no energy contribution for  $\|\mathbf{u}\| > \pi/\sqrt{2}$  and it is consequently feasible to place the remaining filters in the diagonal direction. This implies a sampling distance of  $\Delta = \sqrt{2}$  as opposed to  $\Delta = 1$  for the horizontal and vertical channel.

Now compare the spectrum of the LP-channel in the last output node and the spectrum at the input node. Both these nodes share the same diamond shaped spectrum but the bandwidth of the output node is reduced by one octave. Consequently another network with the same structure can be attached to the last node to obtain the next four quadrature channels and so on to obtain the decomposition of fig. 2.

## 4.2 Sparse filters

A sequential use of the network in fig. 3 require some considerations to keep the number of coefficients constant for all levels. There are essentially two possibilities to accomplish this. The first is to apply a subsampling by one octave at the last node before the signal is fed to the input node of the next network. Subsampling is memory efficient but require interpolation if several channels are used simultaneously e.g. to interpolate a broadband filter. As memory

is getting cheaper an interesting alternative is to maintain the original size of the signal through the complete network and use *sparse filters*.

Using sparse filters the distance between the (non zero) filter coefficients is increased (by one octave) for each new network that is added. Using sparse filters with a sample distance of  $\Delta$  imply that the spectrum will be repetitive with a period of  $2\pi/\Delta$ . Both subsampling and sparse filters are straight forward to use with the network optimizer. In this example the sparse filter method is used to support a direct comparison between the scales without interpolation.

## 4.3 Initial filtering

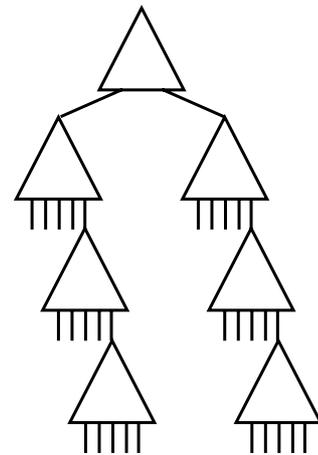


Figure 5: The complete network.

Before the network in fig. 3 can be applied some initial filtering is necessary. The sampling of the spatial domain imply a periodic FD. Assuming a sampling distance,  $\Delta = 1$ , the corners of fig. 2 are located at  $\pm[\pi, \pi]$  and  $\pm[\pi, -\pi]$ . Due to the periodic properties of the FD these four points cannot be separated. Consequently it is not feasible to infer a direction for these frequencies and the initial filtering step is, for that reason, an isotropic LP/HP filter. To the left in fig. 4 a network consisting of 5 levels using 37 real valued coefficients is proposed. This network produce two passbands separated by half an octave. The frequency function for the two output nodes are displayed to the right. The result from either of these two output nodes can be used as input to the network of fig. 3 or, as in this example, where they are both used in parallel to obtain two Fourier domain decompositions overlapping each other by an half octave. In fig. 5 this network is sketched. The top part corresponds to the network of fig. 4. The two output nodes are each connected to a network of fig. 4. A network with the same structure is then connected to the LP-nodes for two further levels.

## 4.4 The complete network

The network of fig. 5 produces 24 quadrature channels within 4 directions and 3 octaves and in addition isotropic HP and LP channels. The entire network requires 469 real

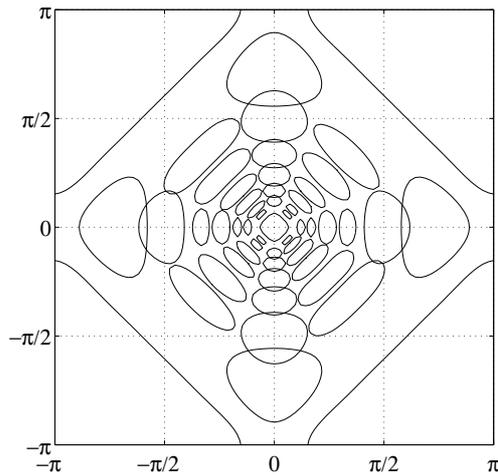


Figure 6: Iso-surface plot of resulting filters, three levels and two frequency bands.

valued multiplications per pixel which corresponds to 17-18 multiplications for each output node.

In fig. 6 an iso-surface plot at the 3dB band width of the resulting quadrature filters is displayed including the final LP-filter and the initial HP-filter. Note that for completeness the quadrature channels are displayed in both sides of the FD.

The left side of fig. 7 contain some selected frequency functions from the first quadrature channel network. The top figure show the Fourier transform of the horizontal LP-filter,  $f_1$ , in fig. 3. Below that the Fourier transform of the corresponding quadrature filter,  $q_1$ . At the bottom left the result at the first horizontal quadrature channel node. This result corresponds to the product of the above frequency functions in the initial frequency function displayed at the lower right in fig. 4. The right part of fig. 7 show the corresponding frequency functions for a filter in the fourth direction and on the second network level. Note that the filters are sparse on this level ( $\Delta = 2\sqrt{2}$ ) which is reflected by the periodic repetition in the Fourier transform of  $f_4$  and  $q_4$ .

## 5 Conclusion

A novel method for optimizing efficient filters using filter networks has been presented. The optimization procedure is recursive and based on the ideal filter functions at the output nodes and a weighting function defining the Fourier space metric. It was demonstrated how the filter network optimizer can be used to implement a quadrature filter bandpass pyramid for practical purposes using spatially sparse kernels. For a  $512 \times 512$  image this implementation require 25 times less multiplications compared to conventional FFT.

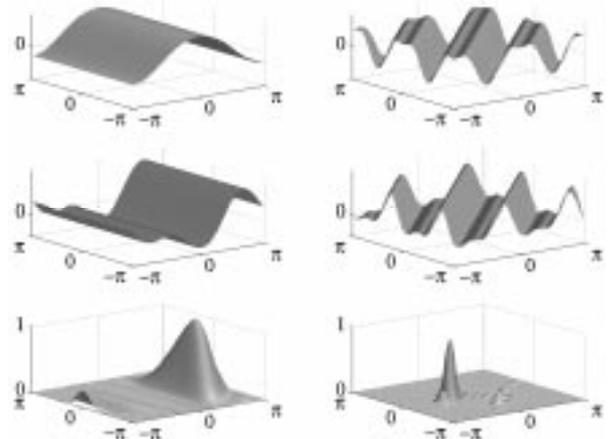


Figure 7: Left: from network level 1 first LP filter and quadrature filter and resulting response at output node 1. Right: Corresponding plots for network level 2 and direction 4, note that the filter components are sparse in the spatial domain which results in a repetition in the FD.

## 6 Acknowledgment

This work was supported by NUTEK, the Swedish National Board for Industrial and Technical Development and by WITAS, The Wallenberg Laboratory for Technical Development and Autonomous Systems which is gratefully acknowledged.

## References

- [1] M. Andersson. *Controllable Multidimensional Filters in Low Level Computer Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, September 1992. Dissertation No 282, ISBN 91-7870-981-4.
- [2] R. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 2nd edition, 1986.
- [3] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice-Hall signal processing series. Prentice-Hall, 1984. ISBN 0-13-604959-1.
- [4] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
- [5] H. Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.
- [6] H. Knutsson and M. Andersson. Optimization of Sequential Filters. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 87–90, Linköping, Sweden, March 1995. SSAB. LiTH-ISY-R-1797. URL: <http://www.isy.liu.se/cvl/ScOut/TechRep/TechRep.html>.
- [7] H. Knutsson, M. Andersson, and J. Wiklund. Advanced Filter Design. In *Proceedings of the Scandinavian Conference on Image analysis*, Greenland, June 1999. SCIA. Also as report LiTH-ISY-R-2142.
- [8] J. Wiklund and H. Knutsson. A Generalized Convolver. In *Proceedings of the 9th Scandinavian Conference on Image Analysis*, Uppsala, Sweden, June 1995. SCIA.