Gaussian-Shaped Circularly-Symmetric 2D Filter Banks

RADU MATEI
Faculty of Electronics and Telecommunications
Technical University of Iasi
Bldv. Carol I no.11, Iasi 700506
ROMANIA

Abstract: - In this paper we propose a class of 2D circularly-symmetric filter banks based on a 1D IIR Gaussian prototype. We approach two cases, namely dyadic and uniform filter banks. The Gaussian prototype is obtained using an efficient rational approximation, namely the Chebyshev-Padé approximation. Then, using a frequency transformation, the corresponding 2D filters with circular symmetry are derived. These filters can be implemented both as digital filters, or in the analog version, using for instance the concept of cellular neural networks, which is detailed in the paper.

Key-Words: - Gaussian filter banks, Chebyshev-Padé approximation, Cellular neural networks

1 Introduction
Filter banks have been extensively used in various image processing tasks [2]. Various different filtering approaches have been presented in the literature. For instance, an important application of filter banks is in texture discrimination [3], [6], i.e. feature extraction in order to make possible their classification. A texture can be identified based on the energy distribution in the frequency domain [3]. Therefore, using appropriate filter banks, the frequency spectrum of the texture image is decomposed into a number of sub-bands [2], [3], [7]. Analog implementation of such filter banks may be a noteworthy alternative in applications requiring fast processing, mainly using the computing power of systems such as cellular neural networks, as detailed below.

2 Cellular Neural Network Filters
The cellular neural network (CNN) is basically an array of identical analog processing elements called cells, which are identically connected [8]. When operating in the linear region, the CNN satisfies the following system of differential state equations [8], [9]:

\[
\frac{d}{dt} x_{ij}(t) = -x_{ij}(t) + \sum_{k,l} A_{kl} x_{kl}(t) + \sum_{p,q} B_{pq} u_{p,q} + I
\]

for all \( |x_{ij}(t)| < 1 \); \( y_{ij}(t) = x_{ij}(t) \).

One of the most important applications of CNNs in image processing is linear filtering [8], [9]. Generally, in the 2-D case, the linear CNN filter is described by the spatial transfer function [9]:

\[
H(\omega_1, \omega_2) = -B(\omega_1, \omega_2) / A(\omega_1, \omega_2)
\]

where \( A(\omega_1, \omega_2), B(\omega_1, \omega_2) \) are 2-D Discrete Space Fourier Transforms (DSFT) of feedback and control templates \( A, B \).

An important constraint is local connectivity, a key feature of CNNs. The design of 2-D high-order spatial filters requires large-size templates, which cannot be directly implemented. The templates currently implemented are \( 3 \times 3 \) or \( 5 \times 5 \) [8], [9]. Larger templates can only be implemented by decomposing them into elementary templates. Only separable templates can be written as a convolution of elementary templates. Correspondingly, the filtering function is a product of elementary functions. In IIR spatial filtering with \( H(\omega_1, \omega_2) \), the final state will be:

\[
X(\omega_1, \omega_2) = U(\omega_1, \omega_2) \cdot H(\omega_1, \omega_2) \cdot H_2(\omega_1, \omega_2) \cdot H_3(\omega_1, \omega_2) \cdot \ldots H_n(\omega_1, \omega_2)
\]

The image \( X(\omega_1, \omega_2) = U(\omega_1, \omega_2) \cdot H(\omega_1, \omega_2) \) resulted in the first filtering is re-applied to CNN input, giving the second output image \( X_2(\omega_1, \omega_2) = X(\omega_1, \omega_2) \cdot H_2(\omega_1, \omega_2) \) and so on, until the whole filtering is achieved. We get:

\[
H(\omega_1, \omega_2) = \prod H_k(\omega_1, \omega_2) = \prod B(\omega_1, \omega_2) \prod A(\omega_1, \omega_2)
\]

which implies a cascade interconnection of several IIR filters [13], [14]. The templates \( A \) and \( B \) can be expressed as discrete convolutions:

\[
B = B_1 \ast B_2 \ast \ldots \ast B_n \quad A = A_1 \ast A_2 \ast \ldots \ast A_n
\]

where \( B_i, A_j \) are \( 3 \times 3 \) or \( 5 \times 5 \) templates. In 2-D CNNs, this successive filtering is possible only for separable templates. Thus, a given filtering operation can be achieved as a sequence of elementary filtering tasks, directly implemented.

3 Circularly-Symmetric Filters
In the following approach we propose an efficient design technique for 2-D circular-symmetric filters, based on 1-D filters, considered as prototypes. Given a 1-D prototype with transfer function \( H_p(\omega) \), the corresponding 2-D filter function \( H(\omega_1, \omega_2) \) results through the transformation \( \omega \rightarrow \sqrt{\omega_1^2 + \omega_2^2} \):

\[
H(\omega_1, \omega_2) = H_p(\sqrt{\omega_1^2 + \omega_2^2})
\]

This can be interpreted as a rotation of the prototype function around its central axis, which generates the 2-D
decomposed into elementary, small-size (3×3 or 5×5) templates, writing it as a discrete convolution:

\[ B = c_k \cdot (C_1 \ast \ldots \ast C_n) \ast (D_1 \ast \ldots \ast D_m) \]  

(15)

If we use the 3×3 template \( C \) given in (8) to approximate the function \( \cos(\omega_1^2 + \omega_2^2) \), each 3×3 template \( C_i \) from (14) is obtained from \( C \) by subtracting the value \( r_i \) from the central element. Each 5×5 template \( D_j \) from (14) results as:

\[ D_j = C \ast C - 2\alpha_j \cdot C_1 + (\alpha_1^2 + \beta_1^2) \cdot C_0 \]  

(16)

\( C_0 \) is a 5×5 zero template, with central element one; \( C_1 \) is a 5×5 template obtained by bordering \( C \) (3×3) with zeros. Thus, starting from an arbitrary 1D prototype filter, a 2D filter with circular symmetry can be derived.

4 Filter Approximation Methods

The Gaussian function in frequency has the form:

\[ G(\omega) = \exp(-\sigma^2\omega^2/2) \]  

(17)

A CNN Gaussian-shaped filter can be designed by determining the template parameters that approximate a Gaussian with a desired selectivity. The Gaussian function is real and symmetric, therefore so will be the templates. We look for a rational approximation for the 1D Gaussian frequency response (17) in the form of a ratio of polynomials in \( \cos \omega \):

\[ G(\omega) = e^{-\frac{\omega^2}{2}} \cong \frac{B(\omega)}{A(\omega)} = \sum_{n=0}^{M} b_n \cos(\omega a_n) / \sum_{n=0}^{N} a_n \cos(\omega a_n) \]  

(18)

where \( \omega \in (-\pi, \pi) \) and \( a_0 = 1 \).

The degrees of the numerator and denominator \((M, N)\) may not be necessarily equal.

4.1 Chebyshev-Padé Approximation

One of the most efficient rational approximation (best tradeoff between accuracy and approximation order) is the Chebyshev-Padé rational approximation. It consists of a rational function which is determined from the Chebyshev series expansion of \( G(x) \) over the interval \([-1,1]\), similarly to the Padé approximation. Detailed references on this approximation can be found in [8], [9]. In the expression of \( G(\omega) \), we will now make the change of frequency variable of the form:

\[ x = \cos(\omega) \Leftrightarrow \omega = \arccos(x) \]  

(19)

and we obtain the Chebyshev-Padé approximation in:

\[ G(x) = e^{-\frac{x^2}{2}} \cong \sum_{i=0}^{m} c_i T_i(x - (a + 1)/2) \]  

\[ \sum_{n=0}^{N} a_n T_n(x - (a + 1)/2) \]  

where \( m = n = 3 \).

This method gives better approximation accuracy than Padé for the same order. The main drawback of the method is that the coefficients of the rational approximation can be only determined numerically using
a symbolic calculation software like MAPLE. An explicit dependence of the coefficients on the standard deviation \( \sigma \) cannot be derived easily, especially for higher orders, as can be done in the Padé approximation.

After finding the Chebyshev-Padé approximation in the form (20), i.e. the coefficients \( q_n \) \((m = 0, M)\) and \( p_n \) \((n = 0, N)\), the explicit expressions for the Chebyshev polynomials may be substituted, finally obtaining a simple rational expression in variable \( \omega \), similar to (18). We return to the original variable \( \omega \), then finally obtain a rational function in \( \cos \omega \), then a function of \( \cos(n \omega) \).

Examples: For \( \sigma = 1 \), the first order Chebyshev-Padé approximation is:
\[
G_1(\omega) = e^{-\sigma^2/2} \approx \frac{0.29628 + 0.295272 \cos \omega}{1 - 0.420415 \cos \omega}
\]  
resulting in a pair of 1×3 templates:
\[
B = [0.14763 \ 0.29628 \ 0.14763] \\
A = [-0.2102 \ 1.0 \ -0.2102]
\]  
For \( \sigma = 2 \), a second-order approximation is enough:
\[
G_2(\omega) = e^{-2\sigma^2} \approx \frac{0.018014 + 0.27496 \cos \omega + 0.010917 \cos 2\omega}{1 - 1.231918 \cos \omega + 0.28144 \cos 2\omega}
\]  
The second-degree polynomials cannot be factorized, having complex roots.

4.2 Iterative Filter Approximation

We look first for a rational approximation of the Gaussian (17) in the range \([-\pi, \pi]\), which represents the spatial transfer function \( H(\omega) \) of the corresponding 1-D filter:
\[
F(\omega) \equiv \left( b_0 + b_1 \cos \omega \right) / \left( a_0 + a_1 \cos \omega \right)
\]  
This approximation is sufficiently accurate for \( \sigma \leq 1 \), but inadequate for \( \sigma > 1 \) (the approximation error is too large). We can obtain it as a Chebyshev-Padé approximation, as before. At this point, the Gaussian (17) can be written as a power of a Gaussian with smaller dispersion: \( G(\omega) = G_k^2(\omega) \), where
\[
G_k(\omega) = \exp\left(-\sigma^2/2N\right) = \exp\left(-\sigma^2/2\omega^2/2\right)
\]  
and \( \sigma_k = \sigma/\sqrt{N} \), with positive integer \( N \).

The input image \( U \) is filtered \( N \) times with the same "kernel" function \( G_k(\omega) \); after each filtering step, the resulting output image is re-applied to the CNN input. We will call this operation an iterative filtering.

Example: We want to implement a Gaussian with \( \sigma = 1.8 \) using an iterative filtering; since we must have \( \sigma_1 = 1.8/\sqrt{N} \leq 1 \), we need a first-order filter iterated for \( N = 4 \) steps. We obtain: \( \sigma_1 = 1.8/\sqrt{N} = 0.9 \). Using the Chebyshev-Padé approach as in section 4.1, we get:
\[
H_{1,1} = -0.3706 + 0.3422 \cos \omega \\
1 - 0.292\cos \omega
\]  
The characteristics are plotted in Fig.1

5 Filter Banks

We will next discuss two types of filter banks which are usually applied in multi-resolution image processing, namely dyadic and uniform.

5.1 Dyadic Filter Bank

We propose a Gaussian filter bank with a dyadic structure, consisting in Gaussian-shaped filters with central frequencies at \( \omega_k = \pi/2^k \), with integer \( k = 1 \ldots N \). The \( k \)-th filter of the bank will have the frequency response of the form:
\[
H_k(\omega) = G_k(\omega - \omega_k) + G_k(\omega + \omega_k)
\]  
where \( G_k(\omega - \omega_k) \) and \( G_k(\omega + \omega_k) \) are the Gaussians \( G_k(\omega) \) shifted in frequency to the values \( \pm \omega_k \). We will consider the first filter of the bank \( H_1(\omega) \) to have a standard deviation \( \sigma = 2 \). The higher-order filters of the dyadic bank, i.e. \( H_k(\omega) \) for \( k \geq 2 \) will have the central frequencies \( \omega_k = \pi/2^k \) and the bandwidths decreasing in a geometrical progression, such that their quality factor is kept constant.

The frequency response of a Gaussian filter bank comprising the first four filters \( H_k(\omega) \), with \( k = 1, 2, 3, 4 \) is displayed in Fig.2(a) on the positive frequency range \( \omega \in [0, \pi] \).

As we know, a Gaussian function with zero mean, of the form (17), can be considered almost zero in the frequency range \( |\omega| > 3/\sigma \), with an error less than 1%. The -3dB cut-off frequency of a Gaussian filter can be found imposing the condition: \( \exp\left(-\sigma^2/2\omega^2/2\right) = 1/\sqrt{2} \), such that we obtain the expression of the -3dB bandwidth:
\[
BW_k = 2\omega_k = 2\sqrt{2\ln 2}/\sigma_k \equiv 0.833/\sigma_k = 0.833/2^k
\]  
If the filters of the bank are centered around the frequencies \( \omega_k = \pi/2^k \), their frequency responses will have the same quality factor, given by the relation:
\[
Q = \frac{\omega_k}{BW_k} = \frac{0.833}{\pi} = 0.265
\]  
For \( k = 1 \) we obtain the first filter: \( G_1(\omega) = \exp(-2\omega^2) \)
\[
H_1(\omega) = G_1(\omega - \pi/2) + G_1(\omega + \pi/2) = e^{-2(\omega-\pi/2)^2} + e^{-2(\omega+\pi/2)^2}
\]  
Using the Chebyshev-Padé approximation for \( G_k(\omega) \) given in (23), we easily obtain after some calculations:
\[
H_{1,1} = 1.826184 - 1.65223\cos(2\omega) + 0.078645\cos(4\omega) \\
7.067563 + 4.563075\cos(2\omega) + 1.037837\cos(4\omega)
\]
For the second filter with \( \sigma = 4 \), we can find directly a Chebyshev-Padé approximation for the frequency response:

\[
H_3(\omega) = G_1\left(\omega - \frac{\pi}{4}\right) + G_2\left(\omega + \frac{\pi}{4}\right) = e^{-\left(\frac{\omega - \pi}{\pi}\right)^2} + e^{-\left(\frac{\omega + \pi}{\pi}\right)^2}
\]

\[
H_5(\omega) = 10^{-3} \cdot \begin{cases} 
1.20657 + 1.38298\cos(\omega) - 0.4807\cos(2\omega) \\
-1.3477\cos(3\omega) - 0.68481\cos(4\omega) \\
1 - 1.65973\cos(\omega + 0.94838\cos(2\omega)) \\
-0.34278\cos(3\omega) + 0.06497\cos(4\omega)
\end{cases}
\]

A convenient factorized form is:

\[
H_5(\omega) = -0.01054 \cdot (\cos(\omega + 0.9592))\cos(\omega + 0.6668)
\]

\[
H_5(\omega) = \begin{cases} 
(\cos(\omega + 0.2837))\cos(\omega - 1.0082) \\
(0.7969 - 0.941\cos(\omega + 0.5\cos(2\omega)) \\
1.2554 - 1.6968\cos(\omega + 0.5\cos(2\omega))
\end{cases}
\]

Using the method described in section 3, from the factorized 1D prototype transfer function \( H_5(\omega) \) given in (34), it is straightforward to derive the templates for the corresponding circular filter. For instance, the feedback template \( A \) results of size \( 9 \times 9 \) and can be written as the discrete convolution of templates:

\[
A(9 \times 9) = A_1(5 \times 5) * A_2(5 \times 5)
\]

where \( A_1 \) is given below as an example for template parameter design:

\[
A_1 = \begin{bmatrix}
0.0156 & 0.0625 & 0.0937 & 0.0625 & 0.0156 \\
0.0625 & -0.1176 & -0.3602 & -0.1176 & 0.0625 \\
0.0937 & -0.3602 & 1.3299 & -0.3602 & 0.0937 \\
0.0625 & -0.1176 & -0.3602 & -0.1176 & 0.0625 \\
0.0156 & 0.0625 & 0.0937 & 0.0625 & 0.0156
\end{bmatrix}
\]

Unfortunately \( A_1 \) is not separable, i.e. it cannot be written as an outer product of two vectors, but rather as a sum of 3 outer products.

5.2 Uniform Filter Bank

Unlike the filter banks with dyadic structure discussed before, in which the bandwidth decreases with a given ratio (in particular two) as their center frequencies decrease towards lower values, in uniform filter banks the frequency responses are uniformly spread along the frequency range, as can be seen in Fig.3(a), for a filter bank with center frequencies \( \omega_k = k\pi/8 \), \( k = 0, \ldots, 7 \). In this case all filters will have the same bandwidth.

**Design example:**

For instance we want to design the filter

\[
H_{3,8}(\omega) = \exp(-32(\omega - 5\pi/8)) + \exp(-32(\omega + 5\pi/8))
\]

which has quite a high selectivity. Using the iterative method, we can realize this band-pass filter by first designing the filter:

\[
H_{3,4}(\omega) = \exp(-8(\omega - 5\pi/8)) + \exp(-8(\omega + 5\pi/8))
\]

and then iterating this filtering for four times:

\[
H_{3,4}(\omega) = H_{3,8}^4(\omega)
\]

Using the Chebyshev-Padé approximation we find the following rational realization for \( H_{3,4}(\omega) \):
\[ H_{3,4}(\omega) = 10^{-2} \cdot \left\{ \begin{array}{c} 0.5048 - 0.3375\cos\omega - 0.5425\cos(2\omega) \\ +0.4316\cos(3\omega) \end{array} \right\} + \left\{ \begin{array}{c} 1 + 1.5659\cos\omega + 1.1361\cos(2\omega) \\ +0.4518\cos(3\omega) + 0.1719\cos(4\omega) \end{array} \right\} \]

(40)

with the factorized form:
\[ H_{3,4}(\omega) = \frac{-0.0125(\cos\omega + 0.9764)(\cos\omega - 0.6525)}{(0.8548 + 1.1129\cos\omega + 0.5\cos(2\omega))} \]

(41)

Using the same procedure we can design all the filters of the form:
\[ H_{3,5}(\omega) = \exp\left(-32(\omega - k\pi/8)^2\right) + \exp\left(-32(\omega + k\pi/8)^2\right) \]

(42)

such that we obtain an uniform filter bank, displayed in Fig.4. It includes the low-pass filter \( H_{3,8}(\omega) = \exp\left(-32\omega^2\right) \).

6 Conclusion

We proposed an efficient design method for both dyadic and uniform two-dimensional circularly-symmetric filter banks, useful in various image processing tasks. The 2D filters are designed starting from 1D Gaussian prototypes with a desired selectivity. Using a rational approximation like Chebyshev-Padé and a frequency transformation, the factorized prototype transfer function gives a direct and systematic realization of the 2D circular filter, by calculating the template parameters. We discussed here mainly filter design methods used in CNN, but they may be also applied to 2D digital filters which are commonly used in image filtering.

References:


Fig.4: (a) Uniform Gaussian filter bank with \( N = 8 \) filters; (b) Ideal and approximated frequency response of the filter \( H_{3,5}(\omega) \); (c) Frequency response and contour plot of the circular filter