# Fast Recursive Algorithms for Efficient Multi-scale Edge Detection

Rutuparna Panda, * L. Samantaray and B.N.Chatterji †

## Abstract

This paper presents fast recursive algorithms for efficient multi-scale edge detection. The key to our approach is to use the scale-changeable exponential bases of compact support to derive recursive filters. It has been shown that the scale factor can be modified to achieve optimal approximation to a Gaussian. Applications to edge detection problems including smoothing in the spatial domain are considered. The implementation of the present recursive filtering structure requires only three multiplications and three additions per sample point, which drastically reduces the computational effort required for smoothing and performing first-order derivative of an image.

**Keywords:** Gaussian filtering, IIR filters, Recursive filtering, Exponential kernel, Multi-scale representations, Edge detection, Computer vision

## 1 Introduction

The first- and second-order derivatives with respect to the image coordinates are very common in computer vision. The most common approach for both the first- and second-order derivatives is the finite differences method [1,2]. More recent approaches often depends on the concept of fitting a low degree polynomial and then differentiating this polynomial [3,4]. Poggio *et al.* [4] have proposed a smoothing cubic spline technique to improve the estimate of the intensity gradient in the presence of noise. Modern edge detection schemes [5,6] insist on initial smoothing operation, which is implicit to all least squares techniques. These authors showed the approach to be more or less equivalent to smoothing the image with a Gaussian low-pass filter in a preprocessing step.

The discrete convolution with a sampled Gaussian in single dimension is given by

$$\begin{aligned} out[n] &= \sum_{k=-N_0}^{k=N_0} in[n-k] \;\; g[k] \\ &= in[n] * g[n] \end{aligned} \tag{1}$$

*With the Dept. of Electronics and Telecommunication Engg., University College of Engg., Burla-768018. INDIA

†With the Dept. of Electronics and Electrical Communication Engg, Indian Institute of Technology, Kharagpur-721302, Corresponding author, e-mail : bnc@ece.iitkgp.ernet.in

where

$$g[n] = g(x,\sigma)|_{x=n} = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{x^2}{2\sigma^2})|_{x=n}, \quad n = \cdots, -2, -1, 0, 1, 2, \cdots$$

Note that $\sigma$ is real and $N_0$ an integer. $N_0$ is typically chosen as $N_0 \approx 5\sigma$ for a required $\sigma$. When $N_0 = 5\sigma$, the continuous Gaussian function g(x) is down by a factor of $3.723 \times 10^{-6}$ from its previous value at x=0. In single dimension, the use of direct convolution of Eq. (1) requires $(2N_0+1)$ multiplications and additions per sample point. Thus, the computational complexity depends on the mask size or the bandwidth of the filter.

This has motivated many researchers to approximate Gaussian filter by some simpler functions in order to reduce computational complexity. Deriche showed that the product of an exponential function and a piecewise linear one can be used to approximate Gaussian filters and the approximated one can then be implemented by an IIR filter [7]. Subsequently, Deriche used the product of an exponential and a cosine-type function to approximate a Gaussian filter [8]. Deriche's recursive filter [7] that has an impulse response per dimension given by $h[n] = k(\alpha|n|+1)e^{(-\alpha|n|)}$ requires eight multiplications and seven additions per sample point. The disadvantage of Deriche's filter is that it is non-isotropic, i.e., it is not circularly symmetric in 2-D [9]. Young and Vliet [9] proposed an alternative implementation of the Gaussian filter that requires six multiplications and six additions per sample point, and their recursive filter is circularly symmetric. With the recent development of multi-resolution signal decomposition techniques [10], there is also a strong need for edge detection techniques compatible with varying levels of resolutions. Unser *at. al* [11] have presented B-spline filtering techniques for signal representation and interpolation. This has motivated us to approximate a Gaussian by a new basis function. We use repeated convolution of exponential kernels of compact support to design the basis function. It has been shown that one can obtain a set of multi-resolution and multi-scale filters by changing the parameters $\alpha$ (scale factor) and m (resolution factor).

This basis function is used to design recursive filters to approximate Gaussian filter. The implementation of these recursive IIR and corresponding FIR filters requires only three multiplications and three additions, which drastically reduces the computational effort required for smoothing and performing first-order derivative of an image. Our approach for edge detection presented here, is some how related to the above approaches in the sense that we are also performing smoothing operation in the first stage of preprocessing.

## 2 Basis Filters

The implementation of the Gaussian filter in one or more dimensions has typically been done as a repeated convolution with a simple filter such as a uniform filter. It is a common practice to use three/four convolutions of a uniform filter (square kernel) to approximate a Gaussian. The width of the square kernel depends upon the spread $\sigma$. Here we consider a similar concept of repeated convolution of an exponential kernel to generate a new basis function to approximate a Gaussian.

### 2.1 Theory of generation of basis function

We define a discrete exponential kernel $p_m(\alpha, x)$ with compact support as :

$$p_m(\alpha, x) = l\left[e^{-\frac{\alpha}{m}x}U(x) - e^{-\alpha}e^{-\frac{\alpha}{m}(x-m)}U(x-m)\right], \quad 0 \le x \le m \tag{2}$$

where $\alpha$ is the scale factor, m is the sampling interval which is also considered as the resolution factor, U(x) is the unit step function, and $l$ is the constant of normalization. $l$ is determined by the unit area of $p(\alpha, x)$, that is :

$$\sum_{x=0}^{m} p_m(\alpha, x) = 1 \tag{3}$$

¿From equations (2) and (3), we have

$$l = \frac{\alpha}{m(1-e^{-\alpha})} \tag{4}$$

In Laplace domain, Eq. (2) can be written as

$$G(s) = \frac{\alpha}{m(1-e^{-\alpha})} \quad \frac{1}{(s+\frac{\alpha}{m})} \quad (1-e^{-\alpha}e^{-ms}) \tag{5}$$

$G(s)$ has a single pole at $s = -\frac{\alpha}{m}$. Now we can generate a new basis function by taking repeated convolutions of the above kernel Eq. (2) with itself. In 2-D, Gaussian filters are circularly symmetric. Hence, to preserve circular symmetry we impose the following condition

$$p_m(\alpha, x) = |p_m(\alpha, -x)| \tag{6}$$

By convolving the kernel Eq. (2) for four times with itself; i.e., by multiplying $G(s)$ for four times with itself with poles at $s = \frac{\alpha_2}{m} \pm j\frac{\alpha_1}{m}$ and $s = -\frac{\alpha_2}{m} \pm j\frac{\alpha_1}{m}$; we get

$$\begin{aligned} B(s) = & \frac{(\alpha_1^2+\alpha_2^2)^2}{4m^4(\cosh\alpha_2 - \cos\alpha_1)} \\ & \frac{1}{[(s-\frac{\alpha_2}{m})^2 + \frac{\alpha_1^2}{m^2}][(s+\frac{\alpha_2}{m})^2 + \frac{\alpha_1^2}{m^2}]} \\ & [1 - 4\cosh\alpha_2\cos\alpha_1 e^{-ms} \\ & +2(\cosh 2\alpha_2 + \cos 2\alpha_1 + 1)e^{-2ms} \\ & -4\cosh\alpha_2\cos\alpha_1 e^{-3ms} + e^{-4ms}] \end{aligned} \tag{7}$$

Thus, the new basis function is given as

$$\beta_m(\alpha, x) = l\sum_{j=0}^{4} w_j \quad g_m(x-jm) \quad U(x-jm) \tag{8}$$

where,

$$g(x) = \alpha_2\cosh\frac{\alpha_2}{m}x\sin\frac{\alpha_1}{m}x - \alpha_1\sinh\frac{\alpha_2}{m}x\cos\frac{\alpha_1}{m}x$$

$$\begin{aligned} l &= \frac{(\alpha_1^2+\alpha_2^2)}{8m\alpha_1\alpha_2(\cosh\alpha_2 - \cos\alpha_1)^2} \\ w_0 &= w_4 = 1 \\ w_1 &= w_3 = -4\cosh\alpha_2\cos\alpha_1 \\ w_2 &= 2(1+\cosh 2\alpha_2 + \cos 2\alpha_1) \end{aligned}$$

U(x) is the unit step function.

The basis function closely approximates the Gaussian. This new basis function resembles with that of a cubic B-spline function. But this basis function is more useful in the sense that it can be used in multiscale and multiresolution spaces. By adjusting $\alpha_1$ and $\alpha_2$ parameters, one can achieve a better bandwidth and cut-off rate compared to conventional cubic splines which will be discussed later in section 2.3.

### 2.2 Smoothing filter

Smoothing filters are implicitly used in a preprocessing step in new edge detection algorithms. This has motivated us to replace the Gaussian function by another highly localised function. Here we propose a new smoothing filter derived from the above basis function Eq. (8). The inverse filter, used for the smoothing purposes, is given as

$$B^{-1}(z) = \frac{1}{B(z)} = S(z) = \frac{k_1}{z + k_2 + z^{-1}} \tag{9}$$

where B(z) is the z-transform of the basis function $\beta(x)$, resolution m=1,

$$k_1 = \frac{8\alpha_1\alpha_2(\cosh\alpha_2 - \cos\alpha_1)^2}{(\alpha_1^2+\alpha_2^2)(\alpha_2\cosh\alpha_2\sin\alpha_1 - \alpha_1\sinh\alpha_2\cos\alpha_1)}$$

$$k_2 = \frac{(\alpha_1 \sinh 2\alpha_2 - \alpha_2 \sin 2\alpha_1)}{(\alpha_2 \cosh \alpha_2 \sin \alpha_1 - \alpha_1 \sinh \alpha_2 \cos \alpha_1)}$$

Thus, the smoothing filter in frequency domain is given by

$$S(\omega) = \frac{k_1}{k_2 + 2\cos \omega m} \tag{10}$$

Normalized frequency response of the smoothing filter $S(\omega)$ with different values of $\alpha_1$ and $\alpha_2$, with resolution one (m=1) seems to be more like a Gaussian. This filter act as an antialiasing filter found in conventional sampling theory.

### 2.3 Basis filters for image representation and smoothing

The symmetrical stable IIR filter S(z) Eq. (9) can be decomposed as the sum of two causal and anticausal sequences, which is well known. However, for a completeness, we describe the method briefly as follows:

$$\begin{cases} y^+(k) = f(k) + ay^+(k-1) & (k = 2, \cdots, N) \\ y^-(k) = f(k) + ay^-(k+1) & (k = N-1, \cdots, 1) \\ y(k) = l(y^+(k) + y^-(k) - f(k)) & (k = 1, 2, \cdots, N) \\ y^+(1) = f(1) & \\ y^-(N) = y^+(N) & \end{cases} \tag{11}$$

where N is the number of data points, $l = -k_1 a/(1 - a^2)$ and $a = (-k_2 + \sqrt{k_2^2 - 4})/2$ is the smallest root (in absolute value) of the polynomial $z^2 + k_2 z + 1$. It follows that the impulse response is given by

$$S(k) = la^{|k|} \tag{12}$$

Thus, the convolution of the input sequence $f(k)$ with the impulse response $S(k)$ can be obtained with just three multiplications and three additions. This filter is separable. Hence, this can be easily extended to higher dimension by applying it along the row and column coordinates.

Let us assume that the convolution of $\beta(x)$ with $y(k)$ gives the approximated function $\hat{f}(x)$:

$$\hat{f}(x) = \sum_{k=-\infty}^{+\infty} y(k)\beta(x-k) \tag{13}$$

Dealing with finite sequences $y(k)$, any function $f(x)$ can be represented as the sum of shifted basis functions $\beta(x)$. Thus, the present method of signal reconstruction follows a straight forward twofold algorithm as given below:

**Step 1:** Find $y(k)$ for each pixel using Eq. (11).

**Step 2:** Reconstruct the original image by convolving $y(k)$ with the FIR filter B(z).

Note that the FIR filter B(z) can also be implemented with just three multiplications and three additions per sample point. The global transfer function of the basis filter is

$$H(\omega) = S(\omega)B(\omega) \tag{14}$$

where $S(\omega)$ is the frequency response of the smoothing filter (pre-filter) and $B(\omega)$ is the Fourier transform of the basis function $\beta(x)$

$$\begin{aligned} B(\ \ \omega\ \ ) &= B(s)\ \ e^{2ms}|_{s=j\omega} \\ &= \frac{(\alpha_1^2 + \alpha_2^2)^2}{(\cosh \alpha_2 - \cos \alpha_1)^2} \\ &\quad \frac{(\cosh \alpha_2 - \cos \alpha_1)^2 + 2\cosh \alpha_2 \cos \alpha_1 (1 - \cos \omega m) - \sin^2 \omega m}{(\alpha_1^2 + \alpha_2^2)^2 + \omega^2 m^2 (2\alpha_2^2 - 2\alpha_1^2 + \omega^2 m^2)} \end{aligned}$$

It can be shown that the perfect low-pass characteristic can be achieved by properly adjusting the scale factor $\alpha$.

The above algorithm can be used for image representation including smoothing in the spatial domain.

## 3 Edge detection

In this section, a qualitative comparison between the basis filter based edge detection and Sobel operator based edge detection is presented. Here we present a two step methodology for edge detection. The first step is the smoothing. The second step is to find first-order derivatives of smoothed images using a recursive derivative filter.

### 3.1 Theory

The first derivative of the operator Eq. (12) is given by the following antisymmetrical filter.

$$D(k) = \begin{cases} -\frac{l}{a}a^{|k|} & \text{for} \quad n > 0 \\ 0 & \text{for} \quad n = 0 \\ \frac{l}{a}a^{|k|} & \text{for} \quad n < 0 \end{cases} \tag{15}$$

where $D(k)$ is the first derivative operator for the impulse response shown in Eq. (12).

The above derivative filter can exactly be implemented as

$$\begin{cases} d^+(k) = y(k-1) + ad^+(k-1) & (k = 2, \cdots, N) \\ d^-(k) = -y(k+1) + ad^-(k+1) & (k = N-1, \cdots, 1) \\ d(k) = -l(d^+(k) + d^-(k)) & (k = 1, 2, \cdots, N) \\ d^+(1) = y(1) & \\ d^-(N) = d^+(N) & \end{cases} \tag{16}$$

The implementation of this derivative operator $D(k)$ requires only three multiplications and three additions per sample point.

Thus, the present algorithm for edge detection can be precisely implemented in four steps.

**Step 1:** The image is filtered with the smoothing filter $S(z)$ (using Eq. (11)) with a proper choice of $\alpha$ depending on the amount of noise.

**Step 2:** The sampled values of the image gradient are evaluated by convolution with the derivative operator (using Eq. (16)).

**Step 3:** The values are converted to polar coordinates.

**Step 4:** All points that are not local maxima along a line pointing in the direction of the gradient are set to zero (non-maxima suppression).

This is some how similar to the one proposed by Deriche [8]. Deriche used an exponential operator to find first-order derivatives, whereas we use the basis operator that provides better localisation because of its circular symmetry and compact support.

### 3.1.1 Results and discussion

Fig.1 present the typical result of the above filters on a synthetic image. Results are compared with the Sobel operator.

In this experiment, we have considered a well structured $256 \times 256$ synthetic image consisting of 64 circles of different radii and contrasts. Then we have added Gaussian noise to the original image. The noisy image (SNR=27 dB) thus generated is shown in Fig.1.(a). The output of the Sobel operator is shown in Fig.1.(b). The result from the above recursive filtering technique is displayed in Fig.1.(c). We have used here the basis filter with $R = 1$ and $\theta = 5^o$.

The neutral-extra edges found in Sobel operator cause no harm to the edge detection result. The only undesirable consequence is that they make the boundaries thicker. There are, however, many post processing techniques, e.g., thinning technique that can suppress this problem. In this paper, we have not considered this controversial factor. ¿From the results, we see that the edges are more sharp for the present method compared to that by Sobel operator. It provides better localisation compared to Sobel operator.

## 3.2 Edge detection using regularised smoothing filter

Poggio *et al.* showed that the solution minimizing the functional $\epsilon_s^2$ (Poggio,1988):

$$\epsilon_s^2 = \sum_{k=-\infty}^{+\infty} (f(k) - \hat{f}(k))^2 + \lambda \int_{-\infty}^{\infty} \left(\frac{\partial^2 \hat{f}(x)}{\partial x^2}\right)^2 dx = \epsilon_A^2 + \lambda \epsilon_r^2 \quad (17)$$

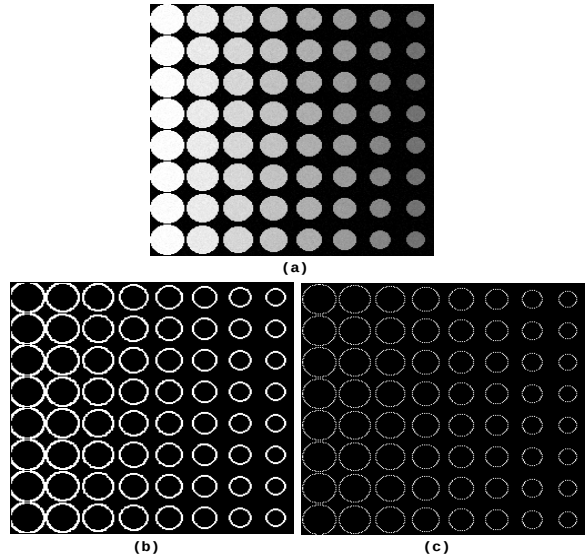


Figure 1: Example of Edge detection. (a) Noisy synthetic image (SNR=24 dB), (b) output from the Sobel operator, and (c) gradient magnitude after non-maxima deflection algorithm using the basis filter with $R = 1$ and $\theta = 5^o$.

closely approximate a Gaussian with spread controlled by the parameter $\lambda$. The solution of Eq. (17) using B-spline approximation is known as approximation cubic spline. A disadvantage of cubic spline approximation is that the resulting impulse response is not strictly positive and crosses zero, causing extraneous inflections in the filtered signal. This phenomenon can be suppressed by using the basis function presented in the preceding sections, which may thought of as a spline under tension. Now we have more degree of freedom. By changing both $\alpha$ (scale parameter) and $\lambda$ (regularisation parameter), one can achieve better approximation to a Gaussian. Thus, the inflection points in the filtered signal may be suppressed. ¿From Eq. (17), we have

$$\epsilon_r^2 = \int_{-\infty}^{\infty} \left( \sum_{i=-\infty}^{\infty} d^{(2)} * y(i) \beta^1(x-i) \right) \left( \sum_{j=-\infty}^{\infty} d^{(2)} * y(j) \beta^1(x-j) \right) dx$$

where $d^{(2)}$ is the second-order symmetric difference operator, $D^{(2)}(z) = (z-2+z^{-1})$ and $\beta^1(x)$ denote the basis function of order one. Note the $\beta(x)$ denotes the basis function of order three and is double derivative continuous. With a change of variable $j = i - k$, we get

$$\begin{aligned}
\epsilon_r^2 &= \sum_{i=-\infty}^{\infty} d^{(2)} * y(i) \sum_{k=-\infty}^{\infty} d^{(2)} * y(i-k) \int_{-\infty}^{\infty} \beta^1(x-i)\beta^1(x-i+k)dx \\
&= \sum_{i=-\infty}^{\infty} d^{(2)} * y(i) \sum_{k=-\infty}^{\infty} d^{(2)} * y(i-k) \int_{-\infty}^{\infty} \beta(x)dx \\
&= \sum_{i=-\infty}^{\infty} (d^{(2)} * y(i))(d^{(2)} * y * \beta(i))
\end{aligned}$$

Using the scalar product notations, $\epsilon_s^2$ can be written as

$$\epsilon_s^2 = \langle f, f\rangle - 2\langle f, b*y\rangle + \langle b*y, b*y\rangle + \lambda\langle d^{(2)}*y, d^{(2)}*y*b\rangle \quad (19)$$

where $f, b$ and $y$ represent discrete sequences.
Equating the derivative of Eq. (19) with respect to $y$ to zero, we obtain

$$b' * f(k) = b' * b * y(k) + \lambda(d^{(2)'} * d^{(2)} * b' * y(k))$$

which, in the z-transform domain, is equivalent to

$$B(z^{-1})F(z) = B(z^{-1})B(z)Y(z) + \lambda D^{(2)}(z^{-1})D^{(2)}(z)B(z^{-1})Y(z)$$

Solving for Y(z), we have

$$\begin{aligned}
Y(z) &= S_\lambda(z)F(z) \\
&= \frac{1}{B(z) + \lambda(-z+2-z^{-1})^2} \; F(z) \quad (20)
\end{aligned}$$

Thus, the regularised smoothing filter $S_\lambda(z)$ is given as

$$S_\lambda(z) = \frac{k_1}{z + k_2 + z^{-1} + k_1\lambda(z^{-2} - 4z^{-1} + 6 - 4z + z^2)} \quad (21)$$

where $k_1$ and $k_2$ are defined in section 2.2.
This IIR regularised smoothing filter can be represented as the product of two causal and anticausal sequences as

$$S_\lambda(z) = S^+(z)S^+(1/z)$$

with

$$S^+(z) = \frac{(1 - 2\rho\cos(\omega) + \rho^2)}{1 - 2\rho\cos(\omega)z^{-1} + \rho^2 z^{-2}} \quad (22)$$

where $\rho$ and $\omega$ are the magnitude and argument of the two smallest complex conjugate roots ($z_1 = \rho e^{j\omega}$ and $z_2 = \rho e^{-j\omega}$) of the characteristic polynomial in the denominator of Eq. (21). After a series of lengthy mathematical simplifications, we get

$$\rho = \left(\frac{4k_1\lambda - 1 - \sqrt{\xi}}{4k_1\lambda}\right)\left(\frac{8k_1\lambda + 4k_1\lambda\sqrt{3 + 36k_2\lambda}}{\xi}\right)^{1/2}$$

$$\tan(\omega) = \left(\frac{36k_2\lambda - 1}{\xi}\right)^{1/2}$$

$$\xi = 1 - 4k_1k_2\lambda + 4k_1\lambda\sqrt{3 + 36k_2\lambda}$$

Therefore, the impulse response of this filter is

$$S_\lambda(k) = k_0 \;\; \rho^{|k|} \;\; [(\cos(\omega|k|) + \gamma\sin(\omega|k|))] \quad (23)$$

where (18)

$$\gamma = \frac{1-\rho^2}{1+\rho^2} \;\; \frac{1}{\tan(\omega)}$$

and the normalising term $k_0$ is given as

$$k_0 = \frac{1+\rho^2}{1-\rho^2} \;\; \frac{1 - 2\rho\cos(\omega) + \rho^2}{1 + 2\rho\cos(\omega) + \rho^2}$$

By taking the advantage of product decomposition, this regularisation filter can be implemented with just four multiplications and four additions per sample point. It is note worthy to mention here that the algorithm described in section 3.1 can be used for less noisy images. For more noisy images, the present regularisation filter is more useful. In other words, $S(z)$ in **Step 1** of the edge detection algorithm described in section 3.1 should be replaced by $S_\lambda(z)$. To conserve space, here we do not provide experimental results with more noisy images. Rather, we provide more theoretical discussions to reveal the suitability of this scheme over other regularisation approaches (Poggio,1988; Chen,1995).

## 4 Conclusion

This paper presents fast recursive optimal filtering structures for multi-scale edge detection. It has been shown that one can obtain a set of multi-resolution and multi-scale basis functions, a property that is quite desirable for multi-resolution and multi-scale signal representations. By adjusting the values of $\alpha$ (scale factor), one can achieve better noise suppression without increasing number of operations per sample point. The regularisation filter is presented in a more generalized framework, which provides better noise immunity and accurate localisation. These algorithms can also be applied to real images. The future scope of the work includes the design of more types of well structured input images and their use in the performance evaluation and comparison, to further reveal the suitability of this scheme. This work can also be extended to find second-order derivative of images.

## References


1. B. K. P. Horn,“ *Robot vision*,” New York: McGraw-Hill, 1986.

2. J. Kittler, “ On the accuracy of the Sobel edge detector,” *Image Vision Comput.*, vol. 1, pp. 37-42, 1983.

3 R. M. Haralick, “ Digital step edges from zero crossing of second directional derivatives,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, pp. 58-68, 1984.

4 T. Poggio, H. Voorhees and A.Yuille,“ A regularized solution to edge detection,” *J. Complexity*,vol. 4, pp. 106-123, 1988.

5 D. Marr and E. Hildreth, “ Theory of edge detection,” *Proc. Roy. Soc. London* B207, pp. 187-217, 1980.

6 J. F. Canny, “A computational approach to edge detection,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, pp. 679-697, 1986.

7 R. Deriche,“Fast algorithm for low-level vision,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 78-87, 1990.

8 R. Deriche,“Recursively implementing the Gaussian and its derivatives,” *Int. Conf. on Image Processing*, Singapore, pp. 263-267, 1992.

9 I. T. Young, and L.J. Van Vliet,“Recursive implementation of the Gaussian filter. *Signal Processing*,” vol. 44, pp. 139-151, 1995.

10 S. G. Mallat,“A theory for multiresolution signal decomposition : the wavelet representation,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 674-693,1989.

11 M. Unser, A. Aldroubi and M. Eden, “B-spline signal processing : Part I - Theory,” *IEEE Trans. Signal Processing*, vol. 41, pp. 821-833, 1993.