

Recursive Gabor Filtering

Ian T. Young, Lucas J. van Vliet, and Michael van Ginkel
 Pattern Recognition Group, Department of Applied Physics
 Lorentzweg 1, Delft University of Technology
 NL-2628 CJ Delft, The Netherlands
 e-mail: young@ph.tn.tudelft.nl

Abstract

In this paper we present a recursive algorithm for the Gabor filter that achieves—to within a multiplicative constant—the fastest possible implementation. For a signal consisting of N samples, our implementation requires $O(N)$ multiply-and-add (MADD) operations. Further, the complexity is independent of the values of σ and ω in the Gabor kernel and coefficients of the recursive equation have a simple, closed-form solution given σ and ω . Our implementation admits not only a forward Gabor transform from t but an inverse transform from t that is also $O(N)$ complexity.

1. Introduction

In two previous papers we developed [1] and refined [2] a method to implement the convolution of signals with Gaussian kernels through the use of two recursive filters of the form:

forward (n increasing):

$$w[n] = (in[n] - (b_1 \cdot w[n-1] + b_2 \cdot w[n-2] + b_3 \cdot w[n-3])) / b_0 \quad (1)$$

backward (n decreasing):

$$out[n] = (B \cdot w[n] - (b_1 \cdot out[n+1] + b_2 \cdot out[n+2] + b_3 \cdot out[n+3])) / b_0 \quad (2)$$

Note that in this implementation—illustrated in Figure 1—the forward filter has an infinite impulse response (IIR) which we can call $h[n]$ and the backward filter has the infinite impulse response $h[-n]$. The concatenation of the two filters as given in equations (1) and (2) leads to a total filter $p[n] = h[n] \cdot h[-n]$ whose Fourier transforms are related by $P(\omega) = |H(\omega)|^2$. In other words, the resulting filter $p[n]$ is zero-phase which is exactly what we would require from a convolution with a “zero-centered” Gaussian kernel.

In this paper we extend this approach from the Gaussian filter to the Gabor filter to produce a recursive, IIR filter that achieves the desired result with a minimum of computational complexity, $O(N)$ multiply-and-add (MADD) operations for a signal consisting of N samples.

In our first paper [1] we showed that the method of choosing the coefficients $\{B, b_0, b_1, b_2, b_3\}$ guaranteed that the resulting Gaussian filters would be stable.

Because our procedure for implementation of the

recursive Gabor filter relies heavily upon the method we used in the design of the recursive Gaussian filter, we repeat the salient aspects here.

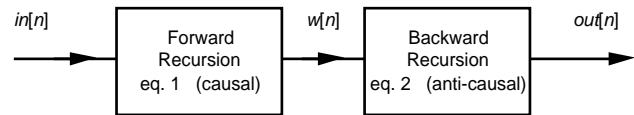


Figure 1: Gaussian filter as concatenation of two recursive filters.

2. Brief Review

Our approach to Gaussian filtering started with a rational approximation to the Gaussian that is given by [3, eq.26.2.20]:

$$g(t) = \frac{1}{\sqrt{2}} e^{-t^2/2} = \frac{1}{a_0 + a_2 t^2 + a_4 t^4 + a_6 t^6} \quad (3)$$

where

$$\begin{aligned} a_0 &= 2.490895 & a_2 &= 1.466003 \\ a_4 &= -0.024393 & a_6 &= 0.178257 \end{aligned} \quad (4)$$

We treated the approximation *not* as the Gaussian impulse response but rather as an approximation to its Fourier transform. That is, we used the well-known result:

$$G(\omega) = e^{-\omega^2/2} = \mathcal{F} \left\{ \frac{1}{\sqrt{2}} e^{-t^2/2} \right\} \quad (5)$$

Using q instead of ω and converting to the Laplace transform using $s = j\omega$, gave the rational approximation:

$$G_q(s) = \frac{A_0}{a_0 - (a_2 q^2) s^2 + (a_4 q^4) s^4 - (a_6 q^6) s^6} \quad (6)$$

The 6th order polynomial in the denominator of $G_q(s)$ has six roots which due to the special structure of the polynomial—only even powers of s —can be written as shown in equation (8) where:

$$m_0 = 1.16680 \quad m_1 = 1.10783 \quad m_2 = 1.40586 \quad (7)$$

These three values $\mathbf{m} = \{m_0, m_1, m_2\}$ are central to the Gabor filter as well. They are solely determined by the values of $\mathbf{a} = \{a_0, a_2, a_4, a_6\}$ and if another choice of \mathbf{a} is made—for example to optimize the Gaussian fit with another criterion—then the values of \mathbf{m} will change as well.

$$G_q(s) = \frac{A_0}{(qs + m_0)(qs + m_1 + jm_2)(qs + m_1 - jm_2)} \frac{1}{(qs - m_0)(qs - m_1 - jm_2)(qs - m_1 + jm_2)} \quad (8)$$

The expression $G_q(s)$ is then factored into the *product* of two terms, $G_q(s) = G_L(s) \cdot G_R(s)$: $G_L(s)$ with poles in the left-half plane and $G_R(s)$ with poles in the right-half plane. For reasons discussed in [1], we used the backward difference technique [4] to transform the two, stable, analog filters characterized by $G_L(s)$ and $G_R(s)$ into two, stable, digital filters $H_+(z)$ and $H_-(z)$. These filters have z-transforms given by:

$$H_+(z) = \frac{1}{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} \quad (9)$$

$$H_-(z) = \frac{B}{b_0 + b_1 z^1 + b_2 z^2 + b_3 z^3} \quad (10)$$

Equations (9) and (10) are precisely the z-transforms of equations (1) and (2), respectively. The forward filter, $h[n]$, corresponds to $H_+(z)$ and is stable because all of its poles are *inside* the unit circle; the backward filter, $h[-n]$, corresponds to $H_-(z)$ and is stable because all of its poles are *outside* the unit circle. The accuracy and speed of this recursive Gaussian implementation are discussed extensively in [1].

3. Gabor Filtering

While operators that focus on global information are essential to describing a variety of physical *systems*, operators that focus on local information are essential to analyzing physical *signals*. For example, linear, time-invariant (or shift-invariant) systems are usually analyzed with Fourier or Laplace transforms which are global operations. But as the examples in Figure 2 show, the information in a signal is usually local.

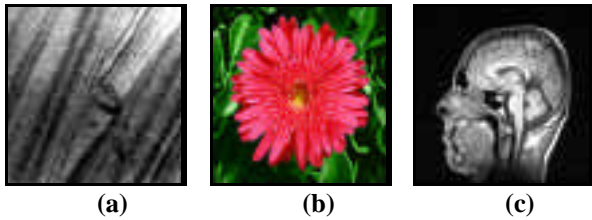


Figure 2: (a) borehole image of subsurface; (b) local color and shape information; (c) MRI image revealing (local) anatomical structures.

While there are many approaches to processing signals in such a way as to examine the local character of structure, the Gabor filter has a number of elegant properties that make it highly suitable for this goal. The one-dimensional Gabor filter is given by:

$$g(t; \sigma, \omega) = \frac{1}{\sqrt{2}} e^{-t^2/2\sigma^2} e^{j\omega t} \quad (11)$$

It is clear that the Gabor filter is the *modulation* of the Gaussian kernel given in equation (3) by the complex term $e^{j\omega t}$. The use of the complex term means that we

will be using complex arithmetic in all calculations.

To examine local structures we seek a filter width in *time* (or space) that is narrow. To obtain good frequency resolution we seek a filter width in *frequency* that is similarly narrow. The Gaussian envelope in equation (11) achieves the smallest possible time-bandwidth product and thus allows us to perform “local” spectral analysis [4], [5].

A number of authors have examined the possibilities for fast implementation of the Gabor filter [6], [7], [8]. To our knowledge the fastest algorithms that have been described are in Qiu [8] and, for a signal consisting of N samples, the transform (filter) has complexity $O(N \cdot \log N)$.

4. The Algorithm

Our implementation is based upon a combination of our recursive Gaussian implementation plus a well-known property that relates modulation in the (discrete) time domain and shifting in the frequency domain. To be specific if the signal $x[n]$ has Fourier spectrum $X(\omega)$ then [4, pp 650-651]:

$$\mathcal{F}\{e^{j\omega_0 n} x[n]\} = X(e^{-j(\omega - \omega_0)} z) \quad (12)$$

Note that the replacement of z with $e^{-j\omega_0} z$ represents a rotation of angle ω_0 around the point $z = (0,0)$ in the complex z -plane. The direct consequence of this is that the Gaussian filter pair $H_+(z)$ and $H_-(z)$ become the Gabor filter pair $B_+(z)$ and $B_-(z)$:

$$B_+(z, \omega_0) = \frac{1}{\sqrt{3}} \sum_{k=0}^3 (b_k e^{jk\omega_0} z^{-k}) \quad (13)$$

$$B_-(z, \omega_0) = \frac{B}{\sqrt{3}} \sum_{k=0}^3 (b_k e^{-jk\omega_0} z^k) \quad (14)$$

Algebraic manipulation of the various terms using Mathematica [9] yields the following:

$$b_0 = 1 \quad scale = (m_0 + q)(m_1^2 + m_2^2 + 2m_1q + q^2)$$

$$b_1 = -q(2m_0m_1 + m_1^2 + m_2^2 + (2m_0 + 4m_1)q + 3q^2) / scale \quad (15)$$

$$b_2 = q^2(m_0 + 2m_1 + 3q) / scale$$

$$b_3 = -q^3 / scale$$

The coefficient of each term in z^k in equation (14) is just the complex conjugate of the term in z^{-k} in equation (13). The five coefficients $\{b_0, b_1, b_2, b_3, scale\}$ are real and are functions of just four numbers $\{m_0, m_1, m_2, q\}$. The first three numbers are given in equation (7). The last number q represents the value that must be used in equations (13) and (14) to achieve a desired standard deviation σ . Using a formula that is a slight improvement in speed and accuracy to that presented in [1], the relationship between q and σ is given by:

$$q = \frac{0.9804(-3.556) + 2.5091 \pm \sqrt{3.556^2 + 0.0561 \sigma^2 + 0.5784 \sigma - 0.2568}}{0.5 + 3.556 \sigma} \quad (16)$$

For $\sigma < 0.5$, the Gaussian envelope in equation (11) is too narrow, consequently undersampled and, therefore, to be avoided.

The resulting Gabor recursive equation pair is:

forward (n increasing):

$$w[n] = in[n] - (fd_1 \cdot w[n-1] + fd_2 \cdot w[n-2] + fd_3 \cdot w[n-3]) \quad (17)$$

backward (n decreasing):

$$out[n] = B \cdot w[n] - (bd_1 \cdot out[n+1] + bd_2 \cdot out[n+2] + bd_3 \cdot out[n+3]) \quad (18)$$

where:

$$\begin{aligned} B &= (m_0(m_1^2 + m_2^2) / scale)^2 \\ fd_1 = bd_1^* &= b_1 \cdot e^{j\omega} \\ fd_2 = bd_2^* &= b_2 \cdot e^{j2\omega} \\ fd_3 = bd_3^* &= b_3 \cdot e^{j3\omega} \end{aligned} \quad (19)$$

5. The initialization

Let us assume that the one-dimensional signal that is to be filtered is of length N , that is $n = 1, 2, \dots, N$. In the forward (causal) direction it is important to use proper values for the first three values of equation (17). These are given by:

$$w[1] = w[2] = w[3] = in[1] / (1 + fd_1 + fd_2 + fd_3) \quad (20)$$

The recursion then proceeds starting at $n = 4$. In the backward (anti-causal) direction, it is equally important to use proper values for the "first" three values of equation (18). These are given by:

$$out[N] = out[N-1] = out[N-2] = B \cdot w[N] / (1 + bd_1 + bd_2 + bd_3) \quad (21)$$

The recursion then starts with $n = N-3$ and continues to $N = 1$.

6. Results

The result of applying this algorithm to a simple impulse input to achieve the discrete time equivalent of equation (11), $g[n]$, is shown in Figure 3 for several values of σ and ω .

The last example in Figure 3 shows an interesting effect. The total signal width was $N = 100$ and σ was 20.0. For the impulse response to be sufficiently small, we should be at or below the 3 value at both ends of the signal, that is, at $n = 1$ and $n = N$. This means that σ and N should satisfy the constraint $6 < N$. This constraint is not satisfied in the last example and that explains the truncated ends of the Gaussian.

7. Inverse Filtering

Our procedure also admits a direct implementation of the inverse transform. If the forward transform, represented by equations (17) and (18), is given (in the z -domain) by the equation:

$$OUT(z) = B_-(z, \omega) (B_+(z, \omega) \cdot IN(z)) \quad (22)$$

then the inverse transform, $IB(z)$, is simply:

$$IB(z) = \frac{1}{B_-(z, \omega) B_+(z, \omega)} \quad (23)$$

The filters $B_+(z)$ and $B_-(z)$ are all-pole filters which means that their inverses are all-zero filters that are non-recursive and stable. The difference equations are given by:

forward (n increasing):

$$u[n] = out[n] + fd_1 \cdot out[n-1] + fd_2 \cdot out[n-2] + fd_3 \cdot out[n-3] \quad (24)$$

backward (n decreasing):

$$in[n] = (u[n] + bd_1 \cdot u[n+1] + bd_2 \cdot u[n+2] + bd_3 \cdot u[n+3]) / B \quad (25)$$

8. The Recipe

Use of this recursive Gabor implementation then becomes a matter of using the following "recipe":

1. Choose σ and ω based upon the desired goal of the filtering;
2. Use equation (16) to determine g ;
3. Use equations (7) and (19) to determine the various coefficients.
4. Use equations (20) and (21) to initialize the procedure;
5. Apply the forward filter with equation (17);
6. Apply the backward filter with equation (18).

9. Conclusions

As equations (17) and (18) require a total of seven, complex multiplies and six, complex additions for each output point, the complexity of the algorithm for N input data points is $7 \cdot k \cdot N$ where k is the constant associated with one complex MADD.

To conclude we show in Figure 4 the effect on two images of Gabor filtering applied in several directions. The two-dimensional Gabor filter kernel is defined as:

$$g(x, y; \omega_x, \omega_y, \sigma_x, \sigma_y) = \frac{1}{2\sigma_x\sigma_y} e^{-x^2/2\sigma_x^2} \cdot e^{-y^2/2\sigma_y^2} \cdot e^{j(\omega_x x + \omega_y y)} \quad (26)$$

In the examples below we use $\sigma_x = \sigma_y = \sigma$. This filter can then be applied at various orientations, θ , through variation of the frequencies as $\omega_x = \omega \cdot \cos \theta$ and $\omega_y = \omega \cdot \sin \theta$. Automatic contrast scaling has been applied to the resulting images.

10. Acknowledgments

This work was partially supported by the Rolling Grants program of the Dutch Foundation for Fundamental Research in Matter (FOM).

11. References

- [1] I. T. Young and L. J. Van Vliet, "Recursive Implementation of the Gaussian Filter," *Signal Processing*, vol. 44, pp. 139-151, 1995.
- [2] L. J. Van Vliet, I. T. Young, and P. W. Verbeek, "Recursive Gaussian Derivative Filters," presented at 14th International Conference on Pattern Recognition, Brisbane, Australia, 1998.
- [3] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover Publications Inc., 1965.
- [4] A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Systems and Signals*, vol. ISBN 0-13-809731-3. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.
- [5] D. Gabor, "Theory of Communications," *Proc. IEE*, vol. 93, pp. 429-459, 1946.
- [6] T. T. Chinen and T. R. Reed, "A Performance Analysis of Fast Gabor Transform Methods," *Graphical Models and Image Processing*, vol. 59, pp. 117-127, 1997.
- [7] C. Richard and R. Lengelle, "Joint recursive processing of time-frequency representations and their modified version by the reassignment method," *Signal Processing*, vol. 60, pp. 163-179, 1997.
- [8] S. Qiu, F. Zhou, and P. E. Crandall, "Discrete Gabor transforms with complexity $O(N \log N)$," *Signal Processing*, vol. 77, pp. 159-170, 1999.
- [9] S. Wolfram, *Mathematica, A System for Doing Mathematics by Computer*, Second ed. Redwood City, California: Addison Wesley, 1991.

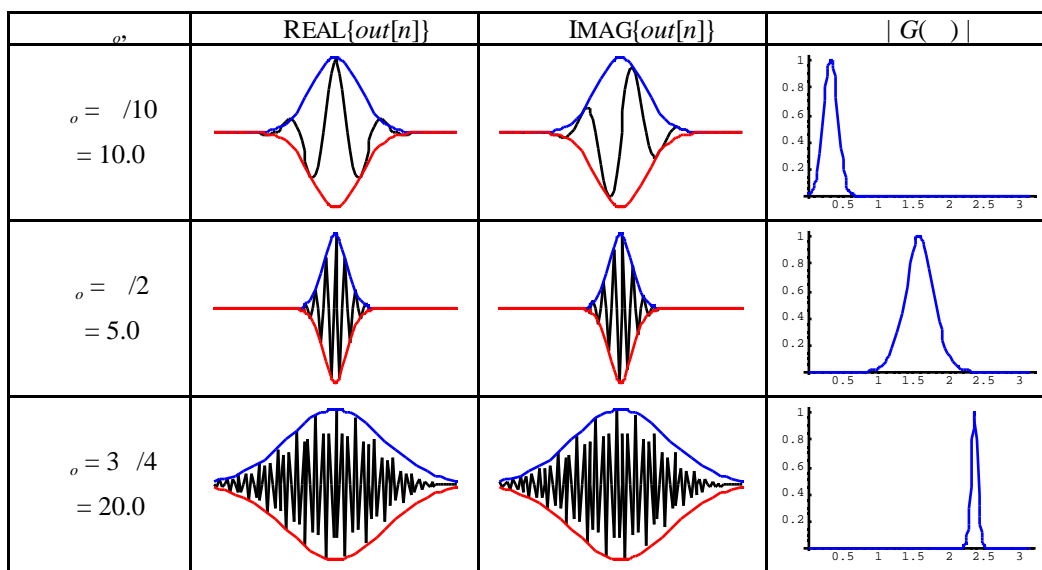


Figure 3: Complex impulse response $g[n]$ and Fourier spectrum $G(\omega)$ for various values of ω and σ . Signal length is $N=100$. Gaussian envelopes have been drawn for purposes of comparison.

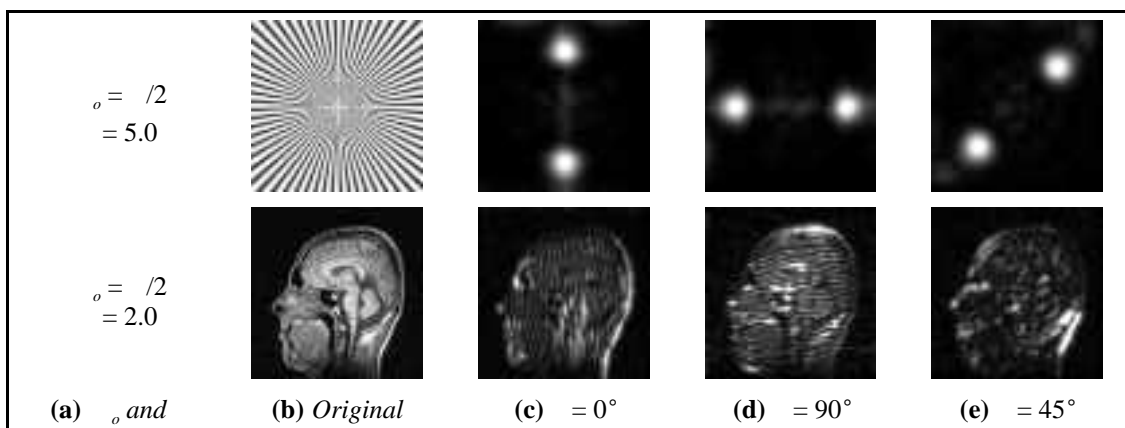


Figure 4: Gabor filtering of a real image produces a complex result. (a) Values of frequency ω and Gaussian width σ ; (b) Original image; (c)-(e) Absolute value of images processed at angles given.