IBPRIA - 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, June 7-9, 2005

A Real-Time Gabor Primal Sketch for Visual Attention *

Alexandre Bernardino and José Santos–Victor

IST, Instituto de Sistemas e Robótica – Lisboa
{alex,jasv}@isr.ist.utl.pt
http://vislab.isr.ist.utl.pt/

Abstract. We describe a fast algorithm for Gabor filtering, specially designed for multi-scale image representations. Our proposal is based on three facts: first, Gabor functions can be decomposed in gaussian convolutions and complex multiplications which allows the replacement of Gabor filters by more efficient gaussian filters; second, isotropic gaussian filtering is implemented by separable 1D horizontal/vertical convolutions and permits a fast implementation of the non-separable zero-mean Gabor kernel; third, short FIR filters and the *à trous* algorithm are utilized to build a recursive multi-scale decomposition, which saves important computational resources. Our proposal reduces to about one half the number of operations with respect to state-of-the-art approaches.

1 Introduction

Gabor filtering is widely applied in image analysis and computer vision applications, such as image compression [5], texture classification [14], image segmentation [15], motion analysis [1] and visual attention [8]. The use of Gabor filters is motivated by information theoretic and biological facts. Gabor [6] showed that gaussian-modulated complex exponentials provide the best trade-off between spatial and frequency resolution. Neurophysiological studies show that visual cortex simple cells are well modeled by families of 2D Gabor functions [4]. Both facts raised considerable interest and suggest that neuronal structures may develop toward optimal information coding.

In the case of visual attention, recent models propose multi-scale image representations of different features like color, intensity and orientation [8]. Such a decomposition benefits, in terms of completeness and stability, on having more than one voice (frequency) per scale and orientation [11]. Therefore, a large number of different kernels may be needed to represent the image characteristics.

Whereas fast algorithms for Gabor filtering exist [18,13], multi-scale representations require analysis with many Gabor kernels, tuned to different orientations, scales and frequencies, which poses serious computational constraints in real-time scenarios. However, many computations are redundant. Here we exploit this redundancy to develop more efficient algorithms.

* Research partially funded by European Project IST 2001 37540 (CAVIAR).

In section 2 we review some of the underlying theory of Gabor analysis and show that image filtering with isotropic zero-mean Gabor kernels (non-separable) can be computed by the sum of two separable filtering operations. In section 3 we show that Gabor filtering can be factored in complex multiplications and gaussian convolutions, which allow significant computational improvements. In section 4 we apply this technique to multi-scale image analysis and propose an approximate algorithm that reduces computations more than 50%.

2 Isotropic Gabor Wavelets

Gabor functions consist on the multiplication of a complex exponential (carrier) and a gaussian function (envelope). We will focus on isotropic envelope functions because efficient separable implementations are currently available. Let $w_\sigma(x,y)$ be a two dimensional gaussian function with scale σ and, $c_\psi(x,y)$, $\psi = (\lambda, \theta)$ be a complex exponential function representing a plane-wave with wavelength λ and orientation θ:

$$w_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{and} \quad c_\psi(x,y) = e^{i\frac{2\pi}{\lambda}(x\cos\theta + y\sin\theta)}$$

To simplify notation, we will drop the spatial coordinates (x,y) and write a two dimensional Gabor function as $\mathbf{g}_{\sigma,\psi} = \mathbf{w}_\sigma \cdot \mathbf{c}_\psi$. This function has non zero mean value, which is not desirable for the purpose of feature extraction and multi-scale analysis. The zero-mean kernel is used instead [11]:

$$\gamma_{\sigma,\psi} = \mathbf{w}_\sigma \cdot (\mathbf{c}_\psi - k_{\sigma,\psi}) \tag{1}$$

where the scalar $k_{\sigma,\psi}$ is calculated so the kernel's average value is zero (Appendix A). We distinguish between the **Gabor function** (non-zero-mean function) and the **Gabor kernel** (zero-mean function). The Gabor kernel satisfies the admissibility condition for wavelets, thus being suited for multi-resolution analysis [12]. Apart from a scale factor, it is also known as the Morlet Wavelet. Examples of two dimensional Gabor kernels are shown in Figure 2.

Image analysis by convolution with Gabor kernels has been extensively studied in the literature. In practical terms, the filter will respond strongly when the local image structure is similar to the Gabor kernel shape, in terms of scale (σ), wavelength (λ), and orientation (θ). The convolution of an image $\mathbf{f}$ with a Gabor kernel $\gamma_{\sigma,\psi}$ is written as $\mathbf{z}_{\sigma,\psi} = \mathbf{f} * \gamma_{\sigma,\psi}$, and using the definition of the Gabor kernel (1), we get:

$$\mathbf{z}_{\sigma,\psi} = \underbrace{\mathbf{f} * \mathbf{g}_{\sigma,\psi}}_{\mathbf{z}^c_{\sigma,\psi}} - k_{\sigma,\psi} \underbrace{\mathbf{f} * \mathbf{w}_\sigma}_{\mathbf{z}^w_\sigma} \tag{2}$$

This convolution can be implemented by subtracting two terms: $\mathbf{z}^c_{\sigma,\psi}$ – a Gabor convolution; and $k_{\sigma,\psi}\mathbf{z}^w_\sigma$ – a scaled gaussian convolution. In the isotropic case both Gabor and gaussian functions are separable ($g(x,y) = g_x(x) \cdot g_y(y)$, and $w(x,y) = w_x(x) \cdot w_y(y)$) and convolutions can be performed with two cascaded (horizontal and vertical) 1D convolutions. Thus, even though the isotropic Gabor kernel γ is not separable itself (can not be written as the tensor product of two 1D filters), image filtering with this kernel can be implemented efficiently as the sum of two separable convolutions.

To date, the fastest implementation of gaussian [17] and Gabor convolutions [18] require 13 real (gaussian) and 13 complex (gabor) arithmetic operations per pixel per dimension. Considering a complex multiplication as 4 real multiplicationa and 2 real additions, the extension to 2-D signals requires, respectivelly, 26 and 108 real operations. Therefore, image convolution with Gabor kernels,consisting in 1 gaussian filtering, 1 Gabor filtering, 1 multiplication and 1 addition, has a total computational cost of 136 operations per pixel.

3 Gabor Convolution Factorization

We show that the Gabor convolution in (2) can be computed by multiplications with complex exponentials and gaussian convolutions. The motivation is that state-of-the-art gaussian filtering is significantly more efficient than Gabor filtering. We focus on the isotropic case but the method can also be applied to the anisotropic case. In fact, a separable implementation of anisotropic Gabor filtering has recently been proposed [7].

Image convolution with Gabor functions, denoted by $\mathbf{z}^c_{\sigma,\psi}$, is computed by:

$$z^c_{\sigma,\psi}(x,y) = \sum_{k,l} f(k,l) \cdot w_\sigma(x-k,y-l) \cdot c_\psi(x-k,y-l)$$

Since $\mathbf{c}_\psi(x-k,y-l) = c_\psi(x,y)\bar{c}_\psi(k,l)$ ($\bar{c}$ denotes complex conjugation), we can expand the previous expression into:

$$z^c_{\sigma,\psi}(x,y) = c_\psi(x,y) \cdot \sum_{k,l} \bar{c}_\psi(k,l) \cdot f(k,l) \cdot w_\sigma(x-k,y-l)$$

In compact form, the full convolution (2) is written as:

$$\mathbf{z}_{\sigma,\psi} = \underbrace{\mathbf{c}_\psi \cdot [(\mathbf{f} \cdot \bar{\mathbf{c}}_\psi) * \mathbf{w}_\sigma]}_{\mathbf{z}^c_{\sigma,\psi}} - k_{\sigma,\psi} \cdot \underbrace{(\mathbf{f} * \mathbf{w}_\sigma)}_{\mathbf{z}^w_\sigma} \tag{3}$$

With the IIR gaussian filter of [17] (26 real operations per pixel), the required computations on Eq. (3), are:

- a **modulation** ($\mathbf{f} \cdot \bar{\mathbf{c}}_\psi$) corresponding to **2 operations** per pixel;
- a **complex gaussian filtering** ($\mathbf{w}_\sigma$ convolved with $\mathbf{f} \cdot \bar{\mathbf{c}}_\psi$) requires **52 operations** per pixel;
- a **demodulation** operation (product of $\mathbf{c}_\psi$ with $(\mathbf{f} \cdot \bar{\mathbf{c}}_\psi) * \mathbf{w}_\sigma$) requires 1 complex multiplication per pixel, corresponding to **6 operations** per pixel;
- a **real gaussian filtering** ($\mathbf{f} * \mathbf{w}_\sigma$) requiring **26 operations** per pixel;
- a **real scaling** by $k_{\sigma,\psi}$, requires **1 operation** per pixel;
- and the **final subtraction**, corresponds to only **1 operation** per pixel because only the real part of Gabor kernels has non zero DC value.

Altogether we have 88 operations which, in comparison with the reference value of 136 operations, correspond to about **35**% savings in computation.

When multiple carriers (orientations/wavelengths) are considered, it is obvious from Eq. (3) that term $\mathbf{z}_\sigma^w$ is common to all. Fig. 1 shows a graphical representation of the method. Gaussian filtering contributes with 26 operations and each carrier contributes with additional 62 operations (our proposal) or 110 operations (direct Gabor filtering). If, for example, 4 orientations and 2 wavelengths are used, the total number of operations is $8 \times 62 + 26 = 522$ *vs* $8 \times 110 + 26 = 906$ (about **42**% savings). It is also worth mentioning that multi-scale image architectures often compute image gaussian expansions to support further processing[2,3]. Thus intermediate filtered images $\mathbf{z}_\sigma^w$ may already have been computed, which saves additional 26 operations per pixel.

4 Analysis at dyadic scales

Dyadic scale representations are very utilized in image analysis. Efficient recursive algorithms exist to build Gaussian and Laplacian pyramids [2] with L dyadic levels ($\sigma \approx 2^i, i = 0, .., L$). Usual approaches create image pyramids by successively filtering previous levels and sub-sampling by 2. Even though sub-sampling is useful in terms of storage and computation, it has the disadvantage of loosing translation invariance properties [12], thus reducing precision in the localization of relevant image structures. We consider the unsubsampled case, where image size is constant at all scales. In this case the *à trous* algorithm [12], is an efficient recursive technique to implement multi-resolution decompositions with constant size filters. If filter coefficients are properly chosen, we obtain good approximations to quasi-dyadic gaussian filters [2].

Consider a signal $f(x, y)$ and low-pass filter $q(x, y)$ with Fourier transform $\widetilde{q}(\omega_x, \omega_y)$. The first step of the *à trous* algorithm consists in obtaining a low-pass version of the original signal: $\mathbf{f}^1 = \mathbf{f} * \mathbf{q}$. In the next decomposition level a new filter is created by expanding the previous one with zero insertion, which, in the frequency domain, corresponds to a spectral compression $\widetilde{q}^1(\omega_x, \omega_y) = \widetilde{q}(2\omega_x, 2\omega_y)$. The new low-pass signal is computed by $\mathbf{f}^2 = \mathbf{f}^1 * \mathbf{q}^1$, and the procedure goes on recursively until the last scale level is reached. Since the convolution operation is linear, this is equivalent to filter the original signal $\mathbf{f}$ with filters $\mathbf{w}^i$ resulting from successive convolutions of the several $\mathbf{q}^k$. In the Fourier domain the equivalent filters are described by $\widetilde{w}^i(\omega_x, \omega_y) = \prod_{k=0}^{i} \widetilde{q}(2^k\omega_x, 2^k\omega_y)$. In [2], some base filters $\mathbf{q}$ were tested but not all choices approximate gaussian functions. We use the 1D filter $q_x = (.05, .25, .40, .25, .05)$ for $x = (-2, -1, 0, 1, 2)$ to generate a set of equivalent filters similar to dyadic gaussian functions.

Since the filter is symmetric, convolution is computed in the following way:

$$f^{i+1}(\cdot) = q_0 f^i(\cdot) + q_1 \left[f^i(\cdot - 2^i) + f^i(\cdot + 2^i)\right] + q_2 \left[f^i(\cdot - 2^{i+1}) + f^i(\cdot + 2^{i+1})\right]$$

In this form, only 6 multiplications and 8 additions per pixel are required to perform the 2D convolution. For a single carrier, we can compute a multi-scale approximation to (3) with 52 operations in the first level and 50 operations in the remaining levels (see Fig. 1). This corresponds to **62**% computation savings with respect to the reference value of 136 operations per pixel.

Finally, let us consider the multi-scale, multi-carrier problem. If S is the number of scales and C the number of carriers, our proposal requires $14 \times S + 2 \times C + 36 \times S \times C$ operations (see Fig. 1), while direct Gabor filtering requires $26 \times S + 2 \times C + 108 \times S \times C$ operations. If all combinations of carriers and scales are needed, then we attain up to **66**% computation savings. For example, considering 3 scales ($S = 3$), 4 orientations and 2 wavelengths ($C = 8$), the full decomposition takes 922 operations *vs* 2686 operations in the reference method.

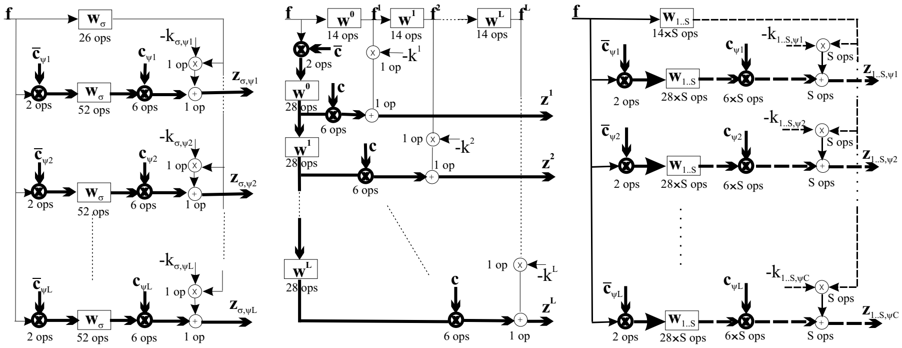

Fig. 1. Proposed Gabor filtering schemes: single-scale-multi-carrier (left), multi-scale-single-carrier (middle) and multi-scale-multi-carrier (right). Thick/Thin lines and boxes represent complex/real signals and filters. At each computational element we indicate the number of real operations required. Dashed lines represent vectors instead of scalars.

5 A "real-time" multi-scale quasi-gabor expansion

We have developed a quasi-dyadic Gabor image decomposition for the control of visual attention in an active vision system, implemented by the *à trous* algorithm with the first 4 scales generated by the proposed base filter, $\sigma = \{0.95, 2.12, 4.35, 8.75\}$. The definition of the carrier wavelengths, λ, is inspired on biological data. Simple and complex cells in the primary visual cortex have receptive fields that resemble Gabor functions of particular combinations and ranges of parameters [11]. In particular the half-amplitude frequency bandwidth (β) range from 0.5 to 2.5 octaves. This parameter depends only on the values of scale and wavelength, as follows.

In the radial frequency direction, an isotropic Gabor function is given by $\widetilde{\mathbf{g}}(\omega) = e^{-\frac{1}{2}\sigma^2(\omega - \frac{2\pi}{\lambda})^2}$, whose half-amplitude points $\omega_{1,2}$ and half-amplitude bandwidth β are, in octaves:

$$\omega_{1,2} = \frac{2\pi}{\lambda} \pm \frac{\sqrt{2\log(2)}}{\sigma} \quad \text{and} \quad \beta = \log_2 \frac{2\pi\sigma + \lambda\sqrt{2\log(2)}}{2\pi\sigma - \lambda\sqrt{2\log(2)}}$$

We have used wavelength values $\lambda = \{3.7, 7.4, 14.8, 29.6\}$. The half-amplitude bandwidth of each scale/wavelength combination is shown in table 1. We choose kernels whose half-amplitude bandwidth is approximately within biologically plausible values (italic entries in table 1). The kernel shapes are shown in Fig. 2, and resemble units tuned to edges, small texture patches and large texture patches, respectively. Roughly speaking, "edge" kernels will respond equally well in image locations corresponding to edges and textures with appropriate scale and orientation. "Texture" kernels will respond better in textured areas with the matched direction and wavelength.

	$\lambda = 3.7$	7.4	14.8	29.6
$\sigma = .95$	*2.68 (E)*	-	-	-
2.12	*.98 (ST)*	*2.26 (E)*	-	-
4.35	*.46 (LT)*	*.95 (ST)*	*2.18 (E)*	-
8.75	.23	*.46 (LT)*	*.95 (ST)*	*2.16 (E)*

Table 1. Half-amplitude bandwidth (in octaves) for each pair scale/wavelength. Italic entries are biologically plausible values. In parenthesis we indicate the appearance of the kernel: E – "edge" kernel, ST – small texture kernel, LT – large texture kernel.

SNR	Aerial	Texture	Misc
Average	30.39	30.06	29.95
Maximum	38.87	39.28	38.92
Minimum	23.82	13.83	7.15

Table 2. Signal to Error Ratio (in dB) between the output of FIR Gabor wavelets and the proposed approximation. Test images are from the collections *miscellaneous*, *aerial* and *texture* of the USC-SIPI database.

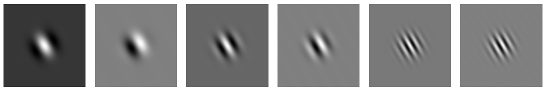

Fig. 2. Real and imaginary parts of: (left) an "edge" (**E**) Gabor kernel with half-frequency bandwidth in octaves $\beta = 2.46$; (center) a "small texture" (**ST**) kernel having $\beta = 1.04$; (right) a "large texture" (**LT**) kernel with $\beta = 0.51$.

Notice that not all combinations of wavelengths and scales are biologically plausible. A recursive dyadic decomposition will require $14 \times S + 2 \times C + 28 \times A_k + 6 \times R_k$ operations, where A_k is the number of levels to compute and R_k is the number of "interesting" kernels. With the IIR filters, some levels are not required and the number of operations is $26 \times S + 2 \times C + 52 \times R_k + 6 \times R_k$. In the proposed decomposition, we have $A_k = 60$ and $R_k = 36$, which lead to 1984 operations in the dyadic recursive decomposition and 2224 with IIR gaussian filters. For the sake of comparison, if the state-of-the art IIR Gabor filters are used, the number of computations would increase to 4024 operations.

6 Results

Figure 3 shows the output modulus of the proposed filter, applied to a common test image. The computation, in 128×128 greyscale images, takes about 0.2 seconds in a P4 2.66GHz processor.

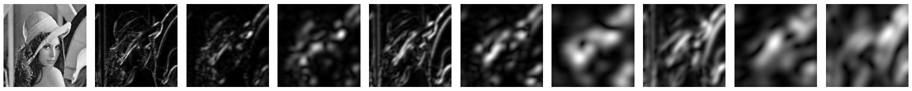

Fig. 3. Modulus of the Gabor wavelet decomposition (for orientation 135^o) applied to the image "Lenna" (left). Contrast has been normalized for visualization. From left to right, the kernel parameters (σ, λ) are, respectively: (0.95,3.7), (2.12,3.7), (4.35,3.7), (2.12,7.4), (4.35,7.4), (8.75,7.4), (4.35,14.8), (8.75,14.8), (8.75,29.6).

We have applied both the approximate method (with the *à trous* decomposition) and the exact method (with FIR Gabor wavelets) to images from the *miscellaneous*, *aerial* and *texture* classes [16], converted to greyscale and 128×128 pixel sizes. We have applied a decomposition of the type described in section 5, with 4 orientations, and the relative mean squared error between the two methods was computed for all images and filter channels. On average, the signal to error ratio is about 30dB (3% error). In some images with strong edges in the boundary, the error grows larger (7dB), but current work is dealing with eficient boundary conditions to address this problem.

7 Conclusions

We have presented a novel algorithm for the computation of Gabor features. Improvements are obtained by an efficient decomposition of Gabor convolution into gaussian convolutions and complex multiplications, and the reuse of intermediate computations in a multi-scale framework. The method reduces computations to about one half when compared to the state-of-the-art. The application of Gabor filters is far from being limited to visual attention. One can find Gabor analysis in object representation [9] texture classification [10], motion estimation [1] and image compression [5]. Therefore, many other applications may benefit from the results presented in this paper.

Appendix A – Computation of Gabor Kernel's *k* parameter

A Gabor Kernel is defined in the frequency domain as:

$$\widetilde{g}(\omega_x, \omega_y) = \widetilde{w}\left(\omega_x - \frac{2\cos\theta}{\lambda\pi^{-1}}, \omega_y - \frac{2\sin\theta}{\lambda\pi^{-1}}\right) - k\widetilde{w}\left(\omega_x, \omega_y\right)$$

Parameter k is computed such that the kernels' DC value is zero.

$$k = \frac{\widetilde{w}(-\frac{2\cos\theta}{\lambda\pi^{-1}}, -\frac{2\pi\sin\theta}{\lambda\pi^{-1}})}{\widetilde{w}(0,0)}$$

With the *à trous* algorithm, the equivalent envelope filters $\widetilde{q}^i(\omega_x, \omega_y)$ have the following Fourier transform :

$$\prod_{k=0}^{i}\left(a\cos(2^{k+1}\omega_x) + b\cos(2^k\omega_x) + c\right) \cdot \left(a\cos(2^{k+1}\omega_y) + b\cos(2^k\omega_y) + c\right)$$

where $a = 0.1$, $b = 0.5$ and $c = 0.4$. Thus, the value of k comes:

$$\prod_{k=0}^{i} \left(a \cos \frac{2^k \cos \theta}{4\pi^{-1}\lambda} + b \cos \frac{2^k \cos \theta}{2\pi^{-1}\lambda} + c \right) \cdot \left(a \cos \frac{2^k \sin \theta}{4\pi^{-1}\lambda} + b \cos \frac{2^k \sin \theta}{2\pi^{-1}\lambda} + c \right)$$

References

1. E. Bruno and D. Pellerin. Robust motion estimation using gabor spatial filters. In *Proc. of the 10th European Signal Processing Conference*, September 2000.
2. P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, 4(31):532–540, April 1983.
3. J. Crowley, O. Riff, and J. Piater. Fast computations of characteristic scale using a half-octave pyramid. In *CogVis 2002, International Workshop on Cognitive Computing*, Zurich, October 2002.
4. J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.
5. S. Fischer and G. Cristóbal. Minimum entropy transform using gabor wavelets for image compression. In *Proc. of Int. Conf. on Image Analysis and Processing*, Palermo, Italy, September 2001.
6. D. Gabor. Theory of communication. *J. IEE*, 93:429–459, 1946.
7. Jan-Mark Geusebroek, Arnold W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. In *ECCV (1)*, pages 99–112, 2002.
8. L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of attention. *Vision Research*, 40:1489–1506, 2000.
9. V. Krueger and G. Sommer. Gabor wavelet networks for object representation. In *DAGM Symposium*, Kiel, Germany, September 2000.
10. P. Kruizinga, N. Petkov, and S.E. Grigorescu. Comparison of texture features based on gabor filters. In *Proc. of the 10th Int. Conf. on Image Analysis and Processing*, pages 142–147, Venice, Italy, September 1999.
11. T. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10), October 1996.
12. S. Mallat. *A Wavelet Tour of Signal Processing, 2nd Ed.* Academic Press, 1999.
13. O. Nestares, R. Navarro, and J. Portilla. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, January 1998.
14. T. Randen and Husøy. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.
15. T. Tangsukson and J.P. Havlicek. Am-fm image segmentation. In *Proc. IEEE Int. Conf. on Image Processing*, pages 104–107, Vancouver, Canada, September 2000.
16. Signal University of Southern California and Image Processing Institute. The usc-sipi image database. http://sipi.usc.edu/services/database.
17. I. Young and L. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44:139–151, 1995.
18. I. Young, L. van Vliet, and M. van Ginkel. Recursive gabor filtering. *IEEE Trans. on Signal Processing*, 50(11):2798–2805, 2002.