Fast Computation of the Gabor Wavelet Transform

Gareth Loy
Department of Systems Engineering
Research School of Information Sciences and Engineering
Australian National University, Canberra 0200
gareth@syseng.anu.edu.au

Abstract

The Gabor filter is a valuable tool in computer vision, however, its high computational load precludes its use in many applications. This paper presents a novel method for approximating Gabor wavelets with a function that can be convolved more efficiently in the spatial domain. The method extends the well known separable nature of the Gabor filter, and demonstrates a means for reducing the computation required when determining the separable 1D convolutions. Experimental results are presented showing the performance of the new method. It is demonstrated that the new method provides an accurate approximation to the convolution with the 1D separated wavelet component of the Gabor kernel, while requiring only 60% *of the computation normally required for this operation.*

1 Introduction

Gabor filters allow local frequency information to be extracted from an image. Unlike Fourier analysis that determines a global frequency domain representation of the entire image, Gabor filters estimate the strength of certain frequency bands and orientations at each location in the image, giving a result in the spatial domain.

Gabor filters are a popular tool for image analysis, and have found widespread use in computer vision. Applications have included face tracking [5], face and object recognition [9, 8], and texture analysis [4, 1, 2]. The use of Gabor filters is supported by neurophysical studies that have suggested the behaviour of receptive fields of the simple cells in the primary visual cortex can be well approximated by Gabor filters [6, 3]. Accordingly Gabor filters have also found applications in biologically inspired early vision systems [7].

The drawback to Gabor filtering in computer vision is the high computational load required. Applying a Gabor filter to an image involves convolution with a set (or jet) of Gabor wavelets consisting of numerous wavelet kernels of different wavelengths and orientations. While the cost of performing these convolutions can be significantly reduced by determining them separably, the total computation is often still prohibitively high.

In this paper an accurate approximation to the Gabor wavelet is presented that facilitates significant increases in the efficiency of the separable convolutions. Section 2 of this paper reviews Gabor wavelets and examines their separable structure. Section 3 describes how these wavelets can be accurately approximated by a new wavelet that can be convolved with an image significantly more efficiently than a standard Gabor wavelet. Section 4 shows the results of this technique applied to real images, and Section 5 concludes with a discussion of further work.

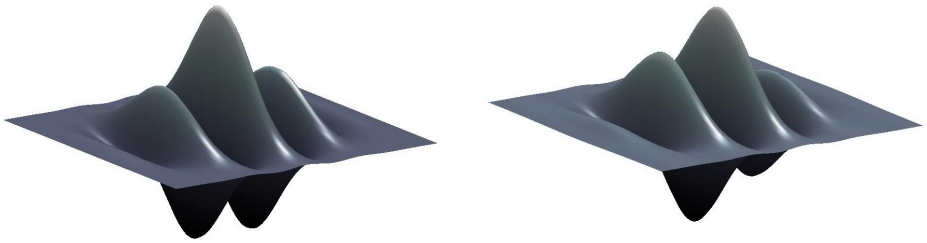
2 Background

2.1 Gabor Wavelets and Jets

A Gabor wavelet is a complex planar wave restricted by a two-dimensional Gaussian envelope. Figure 1 shows the real and imaginary components of a Gabor wavelet. Aside from scale and orientation, the only thing that can make two Gabor wavelets differ is the ratio between wavelength and the width of the Gaussian envelope.

Every Gabor wavelet has a certain wavelength and orientation, and can be convolved with an image to estimate the magnitude of local frequencies of that approximate wavelength and orientation in the image.

A jet of Gabor wavelets consists of numerous wavelets of different wavelengths and orientations. By convolving an image with a full jet of filters it is possible to quantify the magnitude and phase of the local frequency information within the bandwidths of the jet. All wavelets in a jet are typically identical apart from their size and shape.



a b

Figure 1. Quadrature components of a Gabor wavelet kernel, a. real (symmetric) component, b. imaginary (asymmetric) component.

2.2 Separable Formulation

The 2D Gabor wavelet kernel is separable, that is, it can be represented as a convolution of two orthogonal 1D components. These components are: a gaussian $g(x)$, and a wavelet $w(x)$ (a complex wave enveloped by a Gaussian), defined respectively by

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2}) \tag{1}$$

and

$$w(x) = g(x)e^{j\omega x} \tag{2}$$

where $j = \sqrt{-1}$ and ω is the frequency of the wavelet. These functions describe the separable components of a Gabor filter kernel, and are illustrated in Figure 2.

It follows that convolution of a Gabor kernel with an image can be calculated separably. For example, a horizontally aligned $n \times n$ Gabor kernel $\mathbf{K}$ can be written as

$$\mathbf{K} = \mathbf{g} * \mathbf{w}^\top \tag{3}$$

where $\mathbf{g}$ and $\mathbf{w}$ are $n \times 1$ vectors whose elements are define by regularly sampling $g(x)$ and $w(x)$ across intervals centred at $x = 0$. The convolution of $\mathbf{K}$ with an image $\mathbf{I}$ is then

$$\mathbf{I} * \mathbf{K} = \mathbf{I} * (\mathbf{g} * \mathbf{w}^\top) = (\mathbf{I} * \mathbf{g}) * \mathbf{w}^\top \tag{4}$$

By separating the 2D convolution into two 1D convolutions the computation is reduced from $O(pn^2)$ to $O(2np)$ for an $n \times n$ kernel and an image with p pixels. Note that $\mathbf{w}$ is complex, so the $\mathbf{I} * \mathbf{g}$ convolution is performed first in order that only the last convolution require complex arithmetic.

This separable formulation can be applied to any Gabor kernel. It is less straightforward to implement for filter orientations where the separable components of the kernel are not aligned with the vertical or horizontal pixel axes, however, it is feasible to separably determine a convolution with an arbitrarily aligned kernel.

3 A Faster Wavelet

In Section 2 it was demonstrated how a convolution with a separable 2D kernel can be performed more efficiently by serial calculation of two orthogonal 1D convolutions. In this section it will be shown how the 1D convolution with the complex wavelet component ($\mathbf{w}$ in Equation 4) can be performed more efficiently by approximating the wavelet cross-section with a function that can be further separated into a convolution involving a sparse component.

Section 3.1 introduces the *sparse convolution*, this is the device that delivers the computational savings when calculating the wavelet transform. Section 3.2 presents an approximation to the 1D Gabor wavelet function that can be applied using the sparse convolution. Section 3.3 describes the potential benefits in terms of reduced computation.

3.1 Sparse Convolution

The term *sparse convolution* is introduced to describe convolutions involving sparse matrices, that is, matrices most of whose elements are zero. In general, calculating the convolution of an $n \times m$ kernel $\mathbf{K}$ at a single pixel location in an image requires nm products and $nm - 1$ sums. However, if $\mathbf{K}$ is sparse, with a known sparse representation and p non-zero elements, it is only necessary to compute p products and $p - 1$ sums. For $p << nm$ this offers a significant computational saving.

In this paper sparse convolutions are denoted by $\star$ to distinguish them from the standard convolution, denoted by $*$. While convolution and sparse convolution are mathematically equivalent, using the sparse convolution when numer-

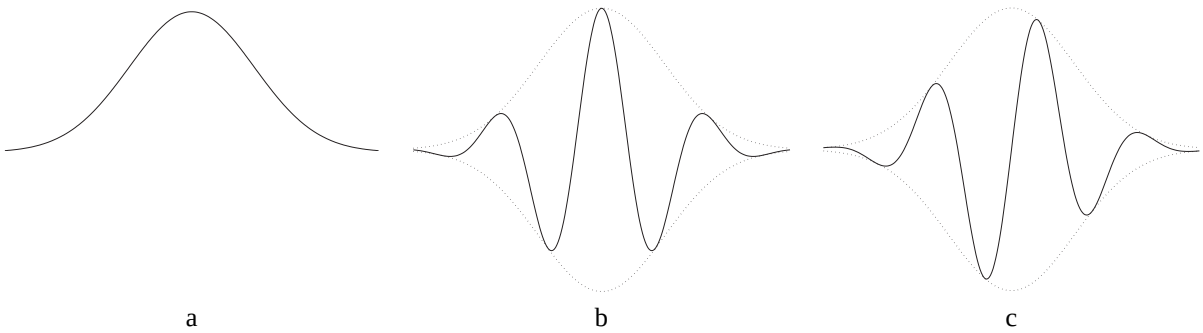


Figure 2. Separable components of a Gabor kernel, a. Gaussian envelope, b. Real (symmetric) part of $w(x)$, c. Imaginary (asymmetric) part of $w(x)$.

ically determining a convolution involving a sparse matrix drastically reduces the computation.

3.2 A Sparse Approximation to the Gabor Wavelet

An approximation to the cross-section of the Gabor wavelet is developed that can be constructed using a sparse convolution of two parallel components. That is, $w(x)$ in Equation 2 is replaced by

$$\hat{w}(x) = c(x) \star q(x) \tag{5}$$

where

$$c(x) = \begin{cases} \cos(x) & \text{for } \frac{\pi}{2} \leq x \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases},$$

$$q(x) = \begin{cases} g(x) & \text{for } \frac{x}{\pi} \bmod 4 = 0 \\ jg(x) & \text{for } \frac{x}{\pi} \bmod 4 = 1 \\ -g(x) & \text{for } \frac{x}{\pi} \bmod 4 = 2 \\ -jg(x) & \text{for } \frac{x}{\pi} \bmod 4 = 3 \\ 0 & \text{otherwise} \end{cases},$$

and $j = \sqrt{-1}$. Figure 3 illustrates these functions, and it is clear that $\hat{w}(x)$ can be constructed from a sparse convolution of $c(x)$ and $q(x)$. The function $c(x)$ is half a wavelength of a cosine wave, and $q(x)$ is zero everywhere except for a small number of control points corresponding to the turning points of $w(x)$. Figure 4 demonstrates how closely $\hat{w}(x)$ approximates $w(x)$.

The new wavelet $\hat{\mathbf{K}}$ is defined analogously to the separable formulation of the standard Gabor wavelet (Equation 3)

$$\hat{\mathbf{K}} = \mathbf{g} \star \hat{\mathbf{w}}^\top$$

where the elements of $\hat{\mathbf{w}}$ are defined by regularly sampling the function $\hat{w}(x)$ across an interval centred at $x = 0$.

As with the standard Gabor wavelet, when $\hat{\mathbf{K}}$ is convolved separably with an image $\mathbf{I}$ it is resolved into two 1D convolutions of orthogonal components

$$\mathbf{I} * \hat{\mathbf{K}} = (\mathbf{I} * \mathbf{g}) * \hat{\mathbf{w}}^\top$$

However, the second of these convolutions can be further separated by using the separable nature of $\hat{\mathbf{w}}$

$$(\mathbf{I} * \mathbf{g}) * \hat{\mathbf{w}}^\top = (\mathbf{I} * \mathbf{g}) * (\mathbf{c} \star \mathbf{q})^\top = \left((\mathbf{I} * \mathbf{g}) * \mathbf{c}^\top\right) \star \mathbf{q}^\top)$$

The result is that the convolution with $\hat{\mathbf{w}}^\top$ is reduced from $O(pn)$ to $O(p(k+s))$ for an image with p pixels, where $\hat{\mathbf{w}}$ is $n \times 1$, $\mathbf{c}$ is $k \times 1$ and $\mathbf{q}$ has s non-zero elements. Clearly $k + s < n$ for all practically scaled digital Gabor kernels.

At first it may appear that all control points (the non-zero elements of $q(x)$) must be located at integer pixel locations, however, this is not the case. A control point at a non-integer pixel location can be precisely represented by fractionally weighted control points at both adjacent integer pixel locations. To emulate the placement of a control point $q(x)$ at a non-integer pixel location x two control points are placed at adjacent integer locations x_0 and x_1 ($x_0 < x < x_1$). The heights of these control points are determined linearly as

$$q(x_0) = (x_1 - x)q(x)$$

$$q(x_1) = (x - x_0)q(x)$$

In general all control points bar the central one may require sub-pixel placement (the central pixel is always located directly over the pixel about which the filter is centred).

3.3 Required Computation

To perform a convolution of an $n \times 1$ complex vector $\mathbf{w}$ at a single location in an image is an order $O(2n)$ operation (the 2 is due to the real and imaginary components).

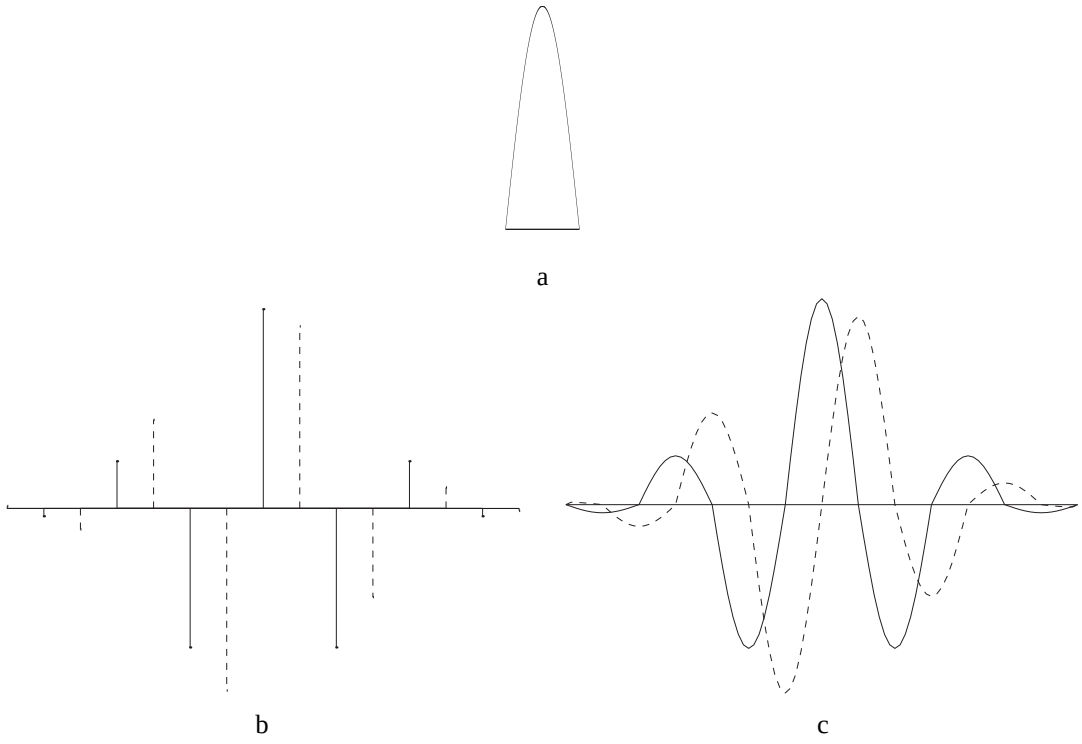


Figure 3. a. Half a wavelength of a cosine wave $c(x)$, b. Sparse function $q(x)$ consisting of discrete control points, real part shown solid, imaginary part shown dashed, c. The sparse approximation to the 1D wavelet $\hat{w}(x)$, real part shown solid, imaginary part shown dased,

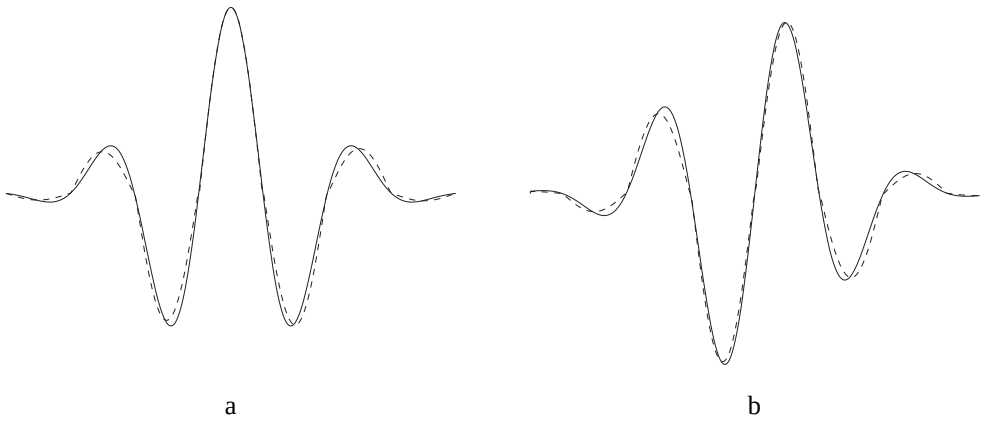


Figure 4. Symmetric (a) and asymmetric (b) quadrature components of a 1D wavelet function $w(x)$ (solid line), and the approximation $\hat{w}(x)$ (dashed)

Performing the same convolution with a vector $\hat{\mathbf{w}}$ can be reduced to order $O(k+s)$ if $\hat{\mathbf{w}}$ is written as a sparse convolution

$$\hat{\mathbf{w}} = \mathbf{c} \star \mathbf{q} \tag{6}$$

and k and s are the length of $\mathbf{c}$ and the number of non-zero control points in $\mathbf{q}$ respectively.

The potential computational savings can be quantified by determining the appropriate values of k and s for a given wavelet.

An $n \times 1$ complex wavelet vector with t turning points in the real (symmetric) part of its cross-section can be approximated by Equation 6, where the length of $\mathbf{c}$ is given by

$$k = 2\lfloor \frac{n-1}{2t} \rfloor + 1$$

and the number of control points in $\mathbf{q}$ is

$$s \leq 4b - 1.$$

These values are determined as follows: $\mathbf{c}$ is defined so it has a centre pixel, meaning its length must be an odd number of pixels, and it should be as close as possible to – but not greater than – half a wavelength of the wavelet function $w(x)$. The maximum number of control points in $\mathbf{q}$ is determined from the case where every point that could require sub-pixel placement does so. There are t real turning points, one of which is at the central pixel, so there can be at most $2t-1$ real control points. There are $b+1$ imaginary control points, but the outer-most two are on the edge of the wavelet so cannot be given sub-pixel accurate placement, therefore there can be at most $2b$ imaginary integer control points.

In summary, for a complex $n \times 1$ wavelet vector with t turning points a direct computation of the convolution will be an order $O(n)$ operation, whereas if it is calculated using the sparse convolution it will be an order $O(\lfloor \frac{n-1}{2t} \rfloor + 2t)$ operation.

Figure 5 illustrates how these orders increase as functions of wavelet size n and number of turning points t, and convincingly shows that the sparse approximation is more efficient to compute for all realistic values of n and t.

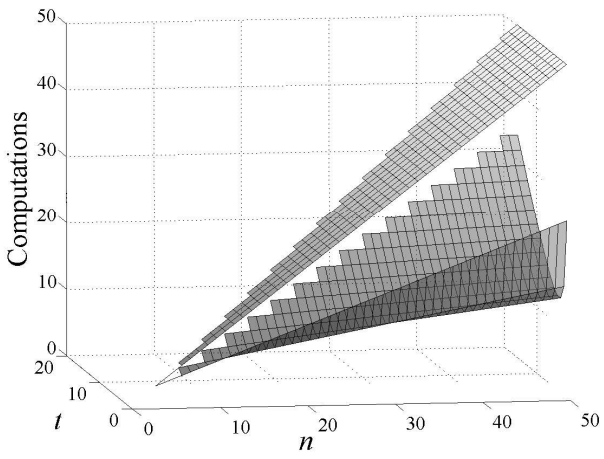


Figure 5. Computational order as a function of wavelet size n and number of turning points t for direct convolution (top surface) and sparse convolution (lower surface).

4 Evaluation of Performance

This section presents results of applying the sparse approximation of the Gabor wavelet to images, and comparing both the visual results and the time required to perform the computations.

Figure 6 shows the results of convolving an image with a standard 1D Gabor wavelet, and a sparse approximation of the same 1D wavelet. Visually the two resulting images appear identical, and analysis shows that the power of the two images differs by only 1.55%.

The CPU time required to determine the convolution with the real component of a 25×1 Gabor wavelet (with 2.5 wavelengths contained within the Gaussian envelope) was timed over fifty iterations on a large image, and compared with the time required to perform the same operation using a sparse approximation and the sparse convolution. The standard method took 120 seconds in Visual C++ on a 750 MHz Pentium III compared to 72 seconds using the sparse convolution.

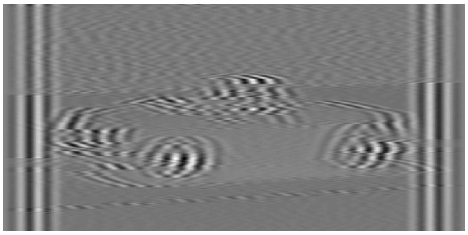
Thus a significant increase in efficiency can be obtained whilst providing an accurate approximation to the convolution with a true Gabor wavelet.

5 Conclusions and Future Work

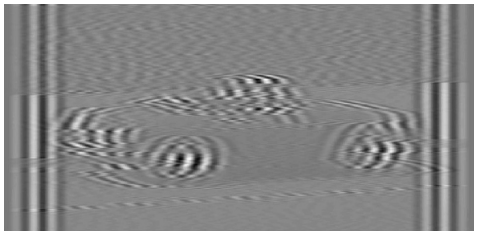
A method has been shown that offers significant increases in performance of computing one half of the separable convolution of a Gabor wavelet with an image.

It remains to increase the efficiency of the second half of this convolution, namely the separable convolution with a 1D Gaussian. Two main avenues towards this goal are currently under investigation. The first is to approximate the Gaussian by a piecewise linear function allowing successive convolutions to be calculated iteratively. The second is to approximate it as a sparse convolution of B-splines and a number of control points. Both these approaches potentially offer significant increases in efficiency with nominal degradation in performance, and in combination with the efficient method for convolving the 1D wavelet presented herein of-

a



b



c

Figure 6. a. Original image, b. Real component of response from standard horizontally aligned Gabor filter, c. Real component of response from sparse approximation of a horizontally aligned Gabor filter.

fer a significantly more efficient alternative to the standard implementation of the digital Gabor wavelet.

References

[1] A. C. Boviak, M. Clark, and W. S. Geisler. Multichannel texture analysising using localized spatial filters. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

[2] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[3] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987.

[4] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen. Object recognition robust under translations, deformations, and changes in background. *IEEE Transations on Pattern Analysis and Machine Intelligence*, pages 769–799, July 1997.

[5] S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, and D. Banin. Tracking facial feature points with gabor wavelets and shape models. In *Proc 1st International Conference on Audio- and Video-based Biometric Person Authentication*, 1997.

[6] D. A. P. S. F. Ronner. Visual cortical neurons as localized spatial frequency filters. *IEEE Trans on Systems, Man, Machines and Cybernetics*, 13(5):907–916, 1983.

[7] J. Thiem and G. Hartmann. Biologically-inspired design of digital gabor filters upon a hexagonal sampling scheme. In *Proceedings of International Conference on Pattern Recognition*, 2000.

[8] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. In L. C. J. *et al*, editor, *Intelligent Biometric Techniques for Fingerprint and Face Recoginition*. Springer Verlag, 1999.

[9] R. P. Würtz. Object recognition robust under translations, deformations, and changes in background. *IEEE Trans on Pattern Analysis and Machine Intelligence*, pages 769–799, July 1997.