The Gabor Transform as a Modulated Filter Bank System

Michel Poize*, Marc Renaudin**, Patrick Venier*

* France Telecom, CNET/CNS/CCI/RCA, 38240 MEYLAN, FRANCE
** Telecom Bretagne, CNET/CNS/CCI/CAT, 38240 MEYLAN, FRANCE

RÉSUMÉ

Nous décrivons un algorithme rapide pour le calcul de la transformation de Gabor. Cet algorithme, qui a pu être développé en adoptant un formalisme de type filtrage, permet une réduction significative de la complexité du calcul des coefficients de Gabor. Sa complexité est de l'ordre de celle d'une transformation de Fourier par bloc plus quelques opérations par point. On obtient ainsi un nombre d'opérations en O(M) pour un signal de M points alors que les algorithmes précédents les plus rapides étaient en O(M.LogM).

ABSTRACT

We describe a fast algorithm to compute the Gabor transform, which belongs to the class of complex lapped non-orthogonal transforms. This new algorithm relies on a filtering formalism and allows a significant decrease in CPU time for the computation of the Gabor coefficients. The complexity of this algorithm is equivalent to that of the block Fourier transform plus 2 to 4 operations for each pixel. The global complexity is thus in O(M) for an M points signal, and compares favorably with the most rapid current method in O(M logM).

1. INTRODUCTION

The pertinence of the Gabor transform is now well known in signal processing. Indeed, the Gabor transform first supplanted classical methods such as block Fourier transforms in signals analysis and is now used in advanced image representation algorithms such as texture extraction or optical flow computation [PORA 88] [SUPE 91] [BOVI 91] [DUNN 91]. In picture coding, the Gabor transform has also been used in different coding schemes and has proven its performance in such a context [EBRA 91].
As the Gabor transform belongs to the class of lapped non-orthogonal transforms, its computational complexity is very high. Although past work produced a number of fast algorithms, the computation time of the Gabor transform is still enormous, especially in the field of image processing and coding [ORR91].

In this article we propose a new fast algorithm to compute the Gabor transform. It is derived from a new theoretical approach to the problem of expanding a signal into a lapped non orthogonal modulated set of functions. The asymptotic complexity of this new algorithm for an M point signal is in O(M), to be compared to O(M.LogM) for the previous ones, all based on the Zak transform [GERT 90]. While the gain in terms of computational complexity is significant, we also show that the structure of this new algorithm is highly interesting from the perspective of VLSI implementation.

2. THE GABOR TRANSFORM

The Gabor transform of a signal consists of expanding this signal into a set of modulated and translated complex gaussian functions. Consider an M point discrete signal, organized in N adjacent blocks of T points each. A Gabor expansion basis can be obtained by modulating and translating a complex Gaussian prototype function g(x) as follows

$$G_{\omega,n}(x) = g(x-nT)W^{\omega(x-nT)} \tag{1}$$

with $0 \le n \le N-1$ (translation), $0 \le \omega \le T-1$ (modulation), and $0 \le x \le M-1$. Giving (1), a signal S(x) can be expressed as

$$S(x) = \sum_{\omega=0}^{T-1}\sum_{n=0}^{N-1} a_{\omega,n}G_{\omega,n}(x) \text{ for } 0 \le x \le M-1 \tag{2}$$

if the set of $G_{\omega,n}$ signals constitutes an effective basis. The $a_{\omega,n}$ coefficients are called the transform coefficients of the signal. The problem of computing these $a_{\omega,n}$ coefficients is not easy, because the set of $G_{\omega,n}$ signals is not orthogonal and requires the solution of a system of M equations and M unknown variables where = N*T. Some iterative solutions exist (projection, gradient descent method [DAUG 88]) but involve very considerable CPU time (several CPU hours for a 256*256 image on a SUN SPARC2 workstation).
The all acceptable solutions derive from the auxiliary functions method introduced by Bastiaans [BAST 80] [BAST 81]. This method is based on the introduction of a set of sequences, $\Gamma_{\omega,n}$, which are dual-biorthogonal to the $G_{\omega,n}$ functions, so that

$$(G_{\omega_2,n_2};\Gamma_{\omega_1,n_1}) = \delta_{\omega_1-\omega_2}\delta_{n_1-n_2} \tag{3}$$

which gives $(S;\Gamma_{\omega_1,n_1}) = \sum_{\omega,n} a_{\omega,n}(G_{\omega,n};\Gamma_{\omega_1,n_1}) = a_{\omega_1,n_1}$ (4).

A set of sequences $\Gamma_{\omega,n}$ is said to be dual- biorthogonal to a set of functions $G_{\omega,n}$ if

$$(G_{\omega_2,n_2};\Gamma_{\omega_1,n_1}) = \delta_{\omega_1-\omega_2}\delta_{n_1-n_2} \tag{5}$$

If such a family $\Gamma_{\omega,n}$ exists, an inner product gives simply the expansion coefficients. Indeed $S = \sum_{\omega,n} a_{\omega,n}G_{\omega,n}$

$$\Rightarrow (S;\Gamma_{\omega_1,n_1}) = \sum_{\omega,n} a_{\omega,n}(G_{\omega,n};\Gamma_{\omega_1,n_1}) = a_{\omega_1,n_1} \qquad (6).$$

Bastiaans showned that, in the case of continuous infinite signals, the $\Gamma_{\omega,n}$ family associated with the set of functions $G_{\omega,n}$ can be obtained analytically. Extending these results to the case of discrete finite signals, different authors [WEXL 90] [GERT 90] showned that the $\Gamma_{\omega,n}$ family has the same form as the $G_{\omega,n}$ family, i.e they can also be obtained by translating and modulating an auxiliary window function that we denote $\gamma(x)$: $\Gamma_{\omega,n}(x) = \gamma(x-nT)W^{\omega(x-nT)}$ (7).

The problem is then reduced to the search for the auxiliary window function $\gamma(x)$. $\gamma(x)$ can be obtained from the window function g(x) by

$$\gamma(x) = \frac{1}{T.Z(g(x))^*} \qquad (8)$$

where Z is the discrete ZAK transform (DZT) [AUSL 91] [ZEEV 92] defined by

$$Z_{g(\omega,p)} = \sum_{k=0}^{N-1} g(p+kT)e^{-2i\pi\frac{\omega k}{N}} \qquad (9).$$

One of the fastest algorithms available today is based on the Zak transform [ORR 91].

3. THE GABOR TRANSFORM AS A MODULATED FILTER BANK SYSTEM

The originality of our approach is to consider the Gabor transform as a filtering process. It results in a new and better theoretical formulation, and a more comprehensive approach to hardware implementation difficulties. This approach will also clearly demonstrate the close relation between previously proposed fast algorithms and uniformly-modulated filter-bank techniques.

3.1 Filter bank synthesizer

The Classical transform approach describes an image reconstruction process as the sum of the Gabor elementary functions weighted by their corresponding Gabor coefficients (Eq. 2). An extension of finite signals to infinite periodic signals allows us to introduce the filtering formalism. Let S(x) be the original signal and $G_{\omega,n}$ the elementary expansion functions. We express the reconstruction formula as follows :

$$S(x) = \sum_{n=-\infty}^{+\infty}\sum_{\omega=0}^{T-1} a_{\omega,n}G_{\omega,n}(x) \qquad (10).$$

We now have to transform equation (10) to express the system as a filter bank. We can do this provided that the $G_{\omega,n}$ are obtained from prototype sequences $G_{\omega,0}$ by translation of n*T, $G_{\omega,n}(x) = G_{\omega,0}(x-nT)$, and letting A_ω be T sequences, such as $A_\omega(n) = a_{\omega,n}$:

$$S(x) = \sum_{n=-\infty}^{+\infty}\sum_{\omega=0}^{T-1} A_\omega(n)G_{\omega,0}(x-nT) \qquad (11).$$

We now use T sequences $B_\omega(m)$, deduced from A_ω by upsampling by a factor T. We therefore have : $B_\omega(m)=0$ for $m \neq nT$. This allows us to introduce additional null terms into equation (11), thus making the convolution products and the equivalent filtering process more apparent : (12)

$$S(x) = \sum_{m=-\infty}^{+\infty}\sum_{\omega=0}^{T-1} B_\omega(m)G_{\omega,0}(x-m) = \sum_{\omega=0}^{T-1} B_\omega(m)*G_{\omega,0}$$

The second index on G is no more useful and $G_{\omega,0}$ will be denoted G_ω in the following.

3.2 Polyphase decomposition of filter bank synthetizer

Equation (12) is illustrated through the diagram of figure 1. By denoting $G_{\omega,0}$ as G_ω we hide the translation in the convolution process. The set of filters $G_\omega(z)$ is obtained by modulation of the prototype filter $G_0(z)$. More precisely this set of filters is a uniformly modulated DFT filter-bank

$$G_0(z) = \sum_{n=-\infty}^{+\infty} a_n z^{-n} \Rightarrow G_\omega(z) = \sum_{n=-\infty}^{+\infty} a_n W^{-n\omega} z^{-n} \qquad (13).$$

Polyphase decomposition is obtained by substituting new indices m and k for n, writing n = mT + k, $0 \le k \le T-1$, $m \in]-\infty,+\infty[$ and introducing the polyphase components of

$$G_0(z) : E_k(z) = \sum_{m=-\infty}^{+\infty} a_{mT+k} z^{-m} \qquad \text{(see appendix).}$$

Classical identities for multirate systems allows us to transform the filtering system of Figure 1 into that of Figure 2, which represents the final structure of the reconstruction process, using the filtering formalism.

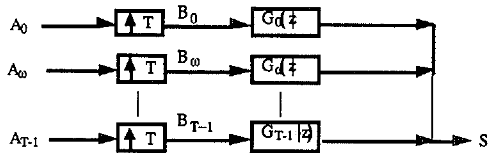

***Figure 1** : filter bank synthesizer structure.*

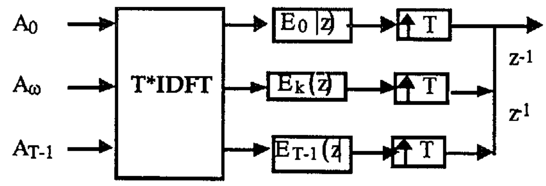

***Figure 2** : polyphase structure of the filter bank synthesizer.*

3.3 Filter-bank analyser and perfect reconstruction condition

The structure of the analysis filter bank is deduced from the synthesizer in a straightforward way. We thus obtain a dual structure as depicted in figure 3.

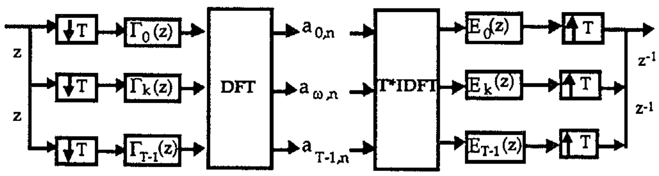

Figure 3 : analysis/synthesis system based on filter banks (IDFT stands for Inverse DFT).

In this sketch, direct and inverse DFT cancel each other out and can be removed. Due to sub- and up-sampling, the graph is reduced to T independent branches. The only way to obtain perfect reconstruction is to reduce each branch to identity or to the same pure delay expression. Perfect reconstruction conditions on E_k and Γ_k can then easily be obtained from the diagram of Figure 3 :

$$\Gamma_k(z).E_k(z) = \frac{1}{T} \qquad (14).$$

This set of equations is equivalent to the set of Fourier equations defining Bastiaans's auxiliary functions (Equation 8).

4. THE FILTER DESIGN PROCEDURE

In this section, the analytical expressions of the analysis and synthesis filters of the new algorithms performing the Gabor transform are clearly defined, thus exhibiting the dependencies between the prototype window function defining the transform and the filter coefficients. The design of both the synthesis and analysis filters are then investigated. We first study the synthesis filters, which can be designed directly from the basis functions. Second, the analysis filters are designed according to the perfect reconstruction conditions in addition to the basis functions.

The case that does not require perfect reconstruction is also studied. Indeed, this case is interesting when using the Gabor transform in lossy picture coding systems. We show that if perfect reconstruction is not required, the analysis filters can be approximated using filters with lower complexity, without introducing noticeable degradation in the coding-decoding system.

4.1. Application to window functions with a support of length 2T

4.1.1. Synthesis filters

In this case (prototype of length 2T), the overlapping between two adjacent windows is half a block to the left and half a block to the right of the block under consideration. Given the theoretical reasons expressed above, the window functions must be applied as depicted in Figure 4. We recall that the polynomials E_k which determine the synthesis filters are expressed as follows:

$$E_k(z) = \sum_{m=-\infty}^{m=+\infty} a_{mT+k} z^{-m} \quad \text{with } k \in [0, T-1] \qquad (15).$$

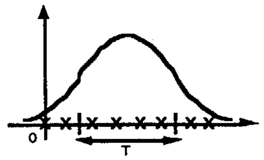

Figure 4 : Application of the window function to a blockof the signal in the 2T case.

In the 2T case considered, the degree of these polynomials is one. Indeed, when the support of the basis functions is limited to two blocks, the parameter "m" belongs to [0,1].Thus we have $E_k(z) = \sum_{m=0}^{m=1} a_{mT+k} z^{-m}$ with $k \in [0, T-1]$

$$\Rightarrow E_k(z) = a_k + a_{T+k} z^{-1} \quad \text{with } k \in [0, T-1] \qquad (16).$$

The synthesis filters are very easy to implement, because they can be built with finite impulse response filters of length two, whatever the dimension of the blocks processed (Cf Figure 5).

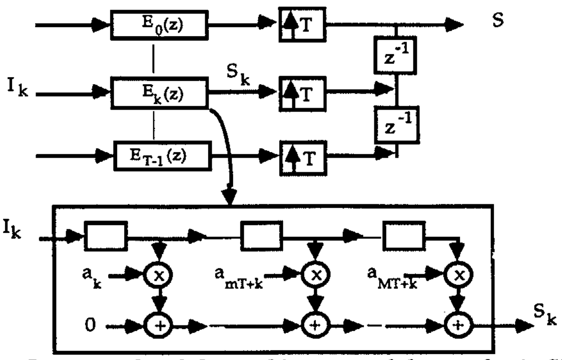

Figure 5 : example of the architecture of the synthesis filters.

4.1.2. Analysis filters

As shown above, the computation of the analysis filters is made by inverting the synthesis filters to obtain a perfect reconstruction (Equation 14). We then obtain from (14) :

$$\Rightarrow \Gamma_k(z) = \frac{1}{T \times a_k} \times \frac{1}{1 + \frac{a_{T+k}}{a_k} z^{-1}} \quad \text{with } k \in [0, T-1] \ (17).$$

We now have the expression of a recursive filter, whose stability is dependent on its unique pole. Two cases occur : the filter is stable and we can very simply deduce its implementation; the filter is unstable and the filtering computation has to be modified.

If the modulus of the pole is less than one, which is expressed as $a_{T+k} / a_k < 1$, the filter is stable and the use of Laurent's series in equation (17) leads to :

$$\Gamma_k(z) = \frac{1}{T \times a_k} \times \sum_{n=0}^{n=\infty} \left(-\frac{a_{T+k}}{a_k} \right)^n z^{-n} \qquad (18).$$

It is now possible to build a generic structure (Figure 6) corresponding to this expression. Figure 7 describes the structure of the analysis filter bank.

We do not investigate the case for which the modulus of the pole is equal to one. Justification is in the fact that we have taken care not to apply the basis functions centered on the sampling grid, thus avoiding the apparition of a zero in the coefficients of the Zak transform of the window (see figure 4). This condition is equivalent to having a pole of modulus equal to one in our formalism.

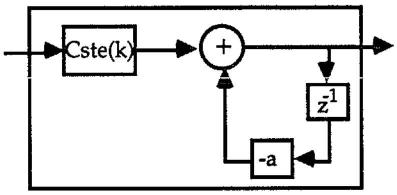

***Figure 6:** structure of the elementary first order recursive filter, Γec_k.*

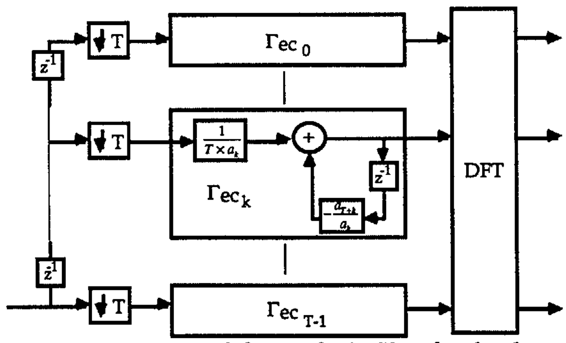

***Figure 7 :** structure of the analysis filter bank when the window functions have a limited support of two blocks.*

If now $a_{T+k} / a_k > 1$, the filter is unstable and we can rewrite equation (17) as follows :

$$\Gamma_k(z) = \frac{1}{T \times a_{T+k} z^{-1}} \times \frac{1}{1 + \frac{a_k}{a_{T+k}} z} \quad \text{with } k \in [0, T-1]$$

$$\Rightarrow \Gamma_k(z) = \frac{1}{T \times a_{T+k} z^{-1}} \times \sum_{n=0}^{n=+\infty} \left(-\frac{a_k}{a_{T+k}} \right)^n z^n = \frac{1}{T \times a_{T+k} z^{-1}} \times \sum_{n=-\infty}^{n=0} \left(-\frac{a_k}{a_{T+k}} \right)^{-n} z^{-n} \qquad (19).$$

We then obtain an anti-causal filter as its computation needs the signal to be reversed. The computation is still the same but the signal has to be exploited in the reverse order. When processing in real time, this implies storing the signal before treatment, thus increasing the memory capacity requirements.

4.2 Non perfect reconstruction analysis filter bank

It is of note that if a perfect reconstruction is not required, these analysis recursive filters can be approximated by finite impulse response filters resulting in a very simple structure, with no extra signal storage requirements. Simulations on images showned that this approximation of the filter responses using very few coefficients does not introduce any noticeable degradation while significantly decreasing the structure complexity. This is due to the rapid decrease in the filters responses. Thus, even if the filters are unstable, it is possible to implement them with a finite impulse-response filter structure avoiding the need of processing the signal in reverse order. Extra memory requirements are hence no longer needed. This modification is acceptable only if the trade-off between a) the complexity of the equivalent FIR filter (number of multipliers) and b) the complexity of the one pole recursive filter plus its extra memory used to reverse the signal, leads to a significant gain.

5. COMPARISON OF GABOR ALGORITHM COMPLEXITY

The discrete signal is taken to be of length M, organized in N blocks of T points each (M=NT). We still assume that the computational complexity of a T point DFT is $\frac{3}{2}T.Log_2(T)$ operations. Note that the algorithm proposed here is the only one whose the complexity per point is not dependent on the length of the original signal. Its complexity per point is only of $\frac{3}{2}T.Log_2(T)+3$ operations. Indeed, the complexities of previous algorithms [ORR 92] to process one point are directly related to M, the length of the original signal, or to N the number of blocks in this signal. Then, the whole complexity of the new algorithm proposed is only in O(M), which is far more efficient than the former ones.

CONCLUSION

A fast algorithm to compute the Gabor transform has been presented. Thus the computational complexity of this transform is no longer an obstacle to further investigations into its application to signal processing. The generic architectures defined above, which exhibit the inherent parallelism of the algorithm can now be exploited by efficient implementation on MIMD type or custom type parallel computers.
This article develops an approach, which is the starting point of the design process of new transforms taking into account the functional performances as well as the ability to be efficiently implemented on parallel computers. Indeed, this approach clearly exhibits the interactions between the algorithm, related to the functional requirement of the transform, and the architecture, strongly related to the ability to exploit a pre-fixed target hardware or silicon medium. In particular, the case where perfect reconstruction is not required can be deeply investigated to perfectly match a specific application in terms of performances and implementation costs.

Appendix : polyphase decomposition.
In this particular case (expansion of a signal on a Gabor basis) the filters $G_\omega(z)$ are obtained by modulating the prototype filter $G_0(z)$ by complex exponential functions. Thus, we obtain a uniformly DFT modulated filter-bank :

$$G_0(z) = \sum_{n=-\infty}^{+\infty} a_n z^{-n} \Rightarrow G_\omega(z) = \sum_{n=-\infty}^{+\infty} a_n W^{-n\omega} z^{-n}$$

Assuming $n = mT + k$ with $k \in [0, T-1]$, $m \in]-\infty,+\infty[$ and $E_k(z) = \sum_{m=-\infty}^{+\infty} a_{mT+k} z^{-m}$ we get

$$G_\omega(z) = \sum_{k=0}^{T-1} \sum_{m=-\infty}^{+\infty} a_{mT+k} W^{-(mT+k)\omega} z^{-mT-k}$$

$$= \sum_{k=0}^{T-1} W^{-k\omega} z^{-k} \sum_{m=-\infty}^{+\infty} a_{mT+k} z^{-mT} = \sum_{k=0}^{T-1} W^{-k\omega} z^{-k} E_k(z^{-T})$$

The output of the filter bank can be expressed as the sum of the outputs of each filter :

$$\sum_{\omega=0}^{T-1} B_\omega(z) G_\omega(z) = \sum_{\omega=0}^{T-1} B_\omega(z) \sum_{k=0}^{T-1} W^{-k\omega} z^{-k} E_k(z^{-T})$$

$$= \sum_{k=0}^{T-1} z^{-k} E_k(z^{-T}) \left(\sum_{\omega=0}^{T-1} W^{-k\omega} B_\omega(z) \right)$$

An IDFT appears by permuting the indices ω and k. Using the classical identities for multirate systems, we obtain the synthesis filter bank of Figure 2.

REFERENCES

[AUSL 91] L. Auslander, I.C. Gertner "The Discrete Zak Transform Application to Time-Frequency Analysis and Synthesis of Nonstationary Signals" IEEE Transaction on Signal processing, Vol 39 No.4, pp.825-835, Avril 1991.
[BAST 80] M.J Bastiaans, "Gabor's Expansion of a Signal into Gaussian Elementary Signals", Proceedings of the IEEE, Vol. 68, N°. 4, pp. 538-539, April 1980.
[BAST 81] M.J Bastiaans, "A sampling theorem for the complex spectrogram, and Gabor's expansion of a signal in Gaussian elementary signals", Optical Engineering, Vol. 20, N°. 4, pp.594-598, July/August 1981.
[BOVI 91] A.C Bovik, "Analysis of Multichannel Narrow-Band Filters for Image Texture Segmentation", IEEE Trans. on Signal Processing, Vol. 39, N° 9, pp. 2025-2043, Sept 1991.
[DAUG 88] J.G Daugman, "Complete Discrete 2-D Gabor Tranforms by Neural Networks for Image Analysis and Compression", IEEE Trans. on Acoustic Speech and Signal Processing, Vol. 36, N° 7, July 1988.
[DUNN 91] D. Dunn, W. Higgins, A. Maida, J. Wakeley "Texture boundary classification using Gabor elementary functions", SPIE Vol 1606 Visual Communication and image Processing, pp 541-552, 1991.
[GERT 90] I. Gertner , Y. Y. Zeevi, "On the Zak gabor Representation of Images" SPIE vol 1360 Visual Communication and Image Processing, pp 1738-1748, 1990.
[EBRA 91] T. Ebrahimi, "Image Compression by Gabor Expansion", Optical Engineering, Vol. 30, N° 7, July 91.
[ORR 91] R.S. Orr, "Computational Assesment of Gabor Representations", Proc. ICASSP, pp 2217-2220, 1991.
[PORA 88] M. Porat, Y.Y Zeevi, "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 10, N°. 4, pp 452-468, July 1988.
[SUPE 91] B.J.Super,A.C.Bovik"Three-dimensional orientation from texture using Gabor wavelets",SPIE Vol 1606 Visual Communication and image Processing, pp 574-586, 1991.
[WEXL 90] J. Wexler , S.Raz "Discrete Gabor Expansions" Elsevier. Signal Processing 21, pp. 207-220, 1990.
[ZEEV 92] Y.Y. Zeevi, I. Gertner "The Finite Zak Transform: An efficient Tool for image Representation and Analysis" Journal of Visual Communication and Image representation, Vol. 3, pp. 13-23, March 1992.