

EFFICIENT COMPUTATION OF THE DISCRETE WIGNER DISTRIBUTION FUNCTION THROUGH A NEW ITERATIVE ALGORITHM

**Isabel García,*Consuelo Gonzalo,*Margarita Pérez Castellanos,*José A. Moreno,*J.M. Sánchez-Dehesa*

**Dep. Informática, Escuela Politécnica, Universidad de Extremadura, Av.Universidad S/N, Cáceres, Spain*

Phone/Fax:+34.27.257256/257202,E-mail: isabelga@unex.es

**Dep. Arquitectura y Tecnología de Sist. Informáticos, Facultad de Informática.UPM. Campus de Montegancedo, Boadilla del Monte, 28668 Madrid*

ABSTRACT

This paper presents a new iterative method to speed up the DWDF computation. At the present it has been considered from a computational point of view as an 1-D section of the Wigner Kernel (WK) N points FT's [1],[4]. We propose a new way to compute the DWDF based on the symmetry properties of the WK and the cosine function. The proposed algorithm is doubly based on a subdivision procedure: on the one hand we have subdivided for each m-value the sum over the k variable into $\log_2 N/4$ -PL partial sums, where PL is the k parity level. And the other hand for each n-value the algorithm computes the DWDF elements by grouping its in group depending on the m PL. The algorithm has been optimized to reduce the accesses of memory, and it improves the FFT algorithms when the number of samples is less than 256 and for this number the algorithm match the FFT algorithms.

1. INTRODUCTION

Some years ago the joint representations were considered as an alternative to classical signal representations, and those are being used extensively in areas as speech and image processing [1],[2]; it has been concluded that the Discrete Wigner Distribution Function (DWDF) presents better properties for Digital Signal Processing, than other joint representations. A common disadvantage to all the joint representations is its high computational complexity, that supposes a handicap for computing in real time. Several forms to compute the DWDF have been proposed in the literature to speed up the DWDF, using different methods, digital computation [1], optical processors [3], and VLSI architectures [4].

This paper presents a new iterative algorithm (AIT) to speed up the DWDF computation reducing the number of operations. We are looking for a new way to compute the DWDF, based on symmetrical properties of the DWDF elements, the WK and the exponential function in equation (1). The number of operations is drastically reduced by the iterative algorithm in relation to the conventional computation of the DWDF. Moreover the fast algorithms frequently used (FFT) are improved when the number of samples is less than 256.

Moreover the AIT is simpler than other FFT algorithms because of its iterative feature, and it is easy to compute on a specific architecture since only operations of additions and multiplications are required for it.

2. COMPUTATIONAL PROPERTIES OF DWDF

The DWDF for a 1-D signal $f(n)$ defined by N samples, is given by the expression:

$$W_f(n, m) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} r_f(n, k) e^{-2i(2\pi m/N)k} \quad (1)$$

where n and m are the spatial and frequential variable respectively. And where:

$$r_f(n, k) = f(n+k)f^*(n-k) \quad (2)$$

The $r_f(n, k)$ usually named the Wigner Kernel (WK), is a hermitian function with respect to the k variable, so the DWDF is always a real function. The computation of the DWDF can be splitted into two computational subproblems: the computation of the WK, equation (2), and the computation Fourier transforms of N points implied by the equation (1) for each n-value of the WK. The first step has been studied by the authors [5], and a model has been segmented and mapped into a generic number of Process Units [6] and it has been proposed a Parallel Process Unit for computing it [8].

We have assumed that the input signal must be a real signal, and the number of samples N is a power of two

The DWDF is always a real function, moreover if only real signals are considered, it is a symmetrical function with respect to the m variable, that is, $W_f(n, m) = W_f(n, -m)$. Based on this symmetry only half DWDF components should be computed. On other hand because of the WK symmetry with respect to the k variable, $r_f(n, k) = r_f(n, -k)$ and the sine function properties $\sin(x) = -\sin(-x)$, then the imaginary part of the exponential is zero for every k value. Besides the cosine function parity, $\cos(x) = \cos(-x)$, it could be concluded that the equation (1) can be rewrites it as:

$$W_f(n, m) = \frac{2}{N} * \left\{ \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} r_f(n, k) \cos \frac{2\pi mk}{N} \right\} + r_f(n, 0) \quad (3)$$

3. SYMMETRIES OF THE DWDF WITH RESPECT TO DISCRETE VARIABLES

The figure 1 shows the cosine values of the function $\cos(2\pi mk/N)$ for $N=32$. It is obvious from it, that there are some m-values whose cosine values are coincident, for one or more k

k \ m	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
-15	0.981	-0.924	0.831	-0.707	0.556	-0.383	0.195	0	-0.195	0.383	-0.556	0.707	-0.831	0.924	-0.981
-14	-0.924	0.707	-0.383	0	0.383	-0.707	0.924	-1	0.924	-0.707	0.383	0	-0.383	0.707	-0.924
-13	0.831	-0.383	-0.195	0.707	-0.981	0.924	-0.556	0	0.556	-0.924	0.981	-0.707	0.195	0.383	-0.831
-12	-0.707	0	0.707	-1	0.707	0	-0.707	1	-0.707	0	0.707	-1	0.707	0	-0.707
-11	0.556	0.383	-0.981	0.707	0.195	-0.924	0.831	0	-0.831	0.924	-0.195	-0.707	0.981	-0.383	-0.556
-10	-0.383	-0.707	0.924	0	-0.924	0.707	0.383	-1	0.383	0.707	-0.924	0	0.924	-0.707	-0.383
-9	0.195	0.924	-0.556	-0.707	0.831	0.383	-0.981	0	0.981	-0.383	-0.831	0.707	0.556	-0.924	-0.195
-8	0	-1	0	1	0	-1	0	1	0	-1	0	1	0	-1	0
-7	-0.195	0.924	0.556	-0.707	-0.831	0.383	0.981	0	-0.981	-0.383	0.831	0.707	-0.556	-0.924	0.195
-6	0.383	-0.707	-0.924	0	0.924	0.707	-0.383	-1	-0.383	0.707	0.924	0	-0.924	-0.707	0.383
-5	-0.556	0.383	0.981	0.707	-0.195	-0.924	-0.831	0	0.831	0.924	0.195	-0.707	-0.981	-0.383	0.556
-4	0.707	0	-0.707	-1	-0.707	0	0.707	1	0.707	0	-0.707	-1	-0.707	0	0.707
-3								0							
-2	0.924	0.707	0.383	0	-0.383	-0.707	-0.924	-1	-0.924	-0.707	-0.383	0	0.383	0.707	0.924
-1	-0.981	-0.924	-0.831	-0.707	-0.556	-0.383	-0.195	0	0.195	0.383	0.556	0.707	0.831	0.924	0.981

Figura 1.-Cosine values, $\cos(2\pi mk/N)$, for $N=32$

values. In the same way, for a constant m value, there are some k values which cosine values are equal. These equalities or *symmetries* always appear with respect to the m and k values $-N/4, -N/8, -N/16, \dots, -2$, which are power of two. Thus, two symmetrical DWDF elements with respect to $m=-N/4=-8$ as $W_k(n, -10)$ and $W_k(n, -6)$, have identical cosine values for every even k value. And they have identical cosine value except the sign for every odd k value.

Since the particular manner that these symmetries in the cosine values are produced, it is necessary in order to establish the AIT algorithm, to define the *Parity Level* concept. In this sense a number could be expressed as two factors product, these are, an odd number and the expression 2^x , where x is named the *Parity Level (PL)* associated to the number which has been decomposed. So, for example, $x=0$ for the odd numbers.

The symmetrical elements with respect to the $m=-N/8=-2$, represented in the figure 1, for example, $W_k(n, -5)$ and $W_k(n, -3)$, are identical cosine values for every even k value whose PL number is equal to 2, and they are identical cosine values except the sign for every even k value whose $PL=1$, and they are different cosine values for every odd k value whose $PL=0$.

In the same way, for a constant m value, there are symmetries with respect to the k variable, and these symmetries depend on PL number associated to m value. Thus the $W_k(n, -1)$ DWDF element have identical cosine values, except the sign, for the symmetrical k values with respect to the $k=-N/4=-8$; this symmetry is the same for every m odd value. In this manner, the $W_k(n, -2)$ DWDF element has identical cosine values for the k values symmetrical with respect to $k=-N/4=-8$, and it has identical

cosine in absolute value for the k values which are symmetrical with respect to the $k=-N/8=-4$.

It is clear from the above paragraph that, the number of operations for computing the DWDF can be reduced by exploiting the equalities or symmetries between the cosine values. In order to exploit the two kind of symmetries, with respect to k and m variables, it is necessary to compute the DWDF elements according to the following method:

1. For exploiting the m -symmetries, it is suitable to split the sum on k variable (eq.3) into several *partial sums* according to PL number of k values PL. Thereby two DWDF elements (m values) are symmetrical if they have one or more partial sums in common (with or without change of sign), and solving the symmetry means to calculate the common partial sum and to accumulate it in both elements.
2. For exploiting the k -symmetries, it is useful to compute the m -elements grouped according to its PL value, because a new function, denoted by $R_k(n, k)$, will be defined for each group of m values. The WK will be replaced by this function in the equation (3), so it has a smaller number of components to be added. To obtain the $R_k(n, k)$ function the WK components corresponding to symmetrical k values with respect to $k=-N/4, -N/8, \dots, -2$, have been added or subtracted. Thus the WK components that in the Eq. (3) are multiplied by the same cosine values, are added and multiplied later, reducing the number of multiplications.

In this way, any DWDF element, for a constant n value, can be obtained as a sum of $\log_2 N/4 - PL(m)$ partial sums, which have been computed separately, being $PL(m)$ the m parity level, by the following equation:

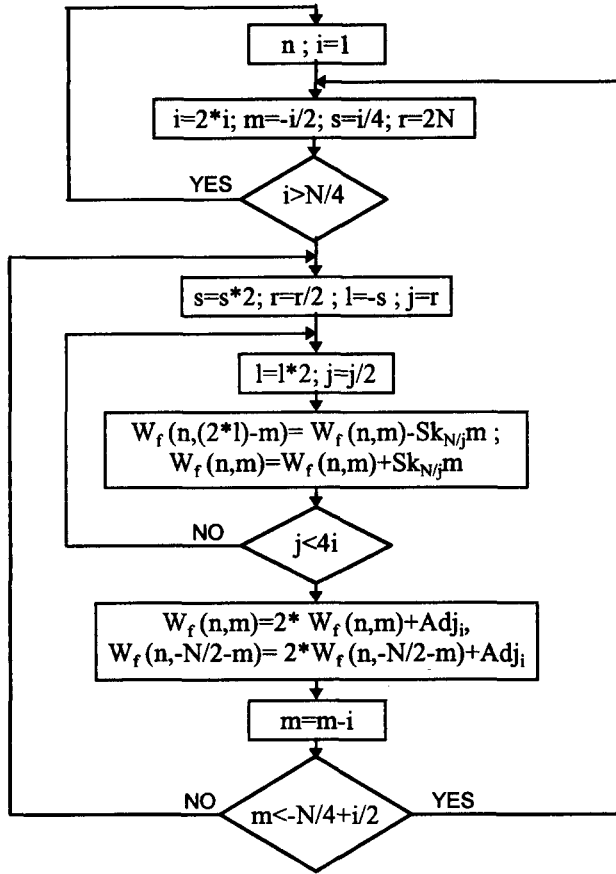


Figure 2.- AIT flowchart

$$W_f(n, m) = \frac{1}{N} \left[2 * \left\{ \sum_{j=4*i}^N Sk_{N/j} m \right\} + Adj_i(n, 0) \right] \quad (4)$$

Where $Sk_{N/j} m$ (5) is the partial sum that includes as many elements as N/j value, $i=2, 4, 8, \dots, N/4$ and $j=4*i, 8*i, 16*i, \dots, N$ taking the power of two values. And $Adj_i(n, 0)$ are the WK terms that are excluded from partial sums that depend on i value, and Adj_i is stated through an equation similar to (6), changing k value by 0. Index i and PL are related by the expression $PL = \log_2 i / 2$.

$$Sk_{N/j} m = \sum_{k=-\frac{N}{j}}^{\frac{N}{j}} Ri_f(n, (\frac{j}{2*i}k + \frac{j}{4*i})) \cos \frac{2\pi m (\frac{j}{2*i}k + \frac{j}{4*i})}{N} \quad (5)$$

Where $Ri_f(n, k)$ is given by:

$$Ri_f(n, k) = r_f(n, k) + \sum_{q=2}^i \sum_{p=1}^{\frac{q}{2}} (-1)^s \left[r_f(n, -k - \frac{pN}{q}) + r_f(n, k - \frac{pN}{q}) \right] \quad (6)$$

where $s=1$ if $q=i$ and $s=0$ if $q \neq i$; moreover $q=2, 4, 8, \dots, i$ take values of two powers from 1 to i and $p=1, 3, 5, 7, \dots, q/2-1$ always

N	N^2	FFT	AIT	AITM
8	64	12	1	1
16	256	32	6	5
32	1024	80	27	21
64	4096	192	112	85
128	16384	448	453	341
256	65536	10244	1818	1365

Table I.- Comparison between methods by number of multiplications

take odd values and k variable takes values from $k=-1$ to $k=N/(2*i)+1$

From the eq.(4), a symmetrical element to m, m' with respect to the $m=-N/4$, could be computed like show the eq. (7):

$$W_f(n, -\frac{N}{2} - m) = \frac{1}{N} \left[2 * \left\{ \sum_{j=4*i}^N (-1)^{\frac{j}{8}} Sk_{N/j} m \right\} + Adj_i(n, 0) \right] \quad (7)$$

Any additional multiplication it is not necessary for the $m'=-N/4-m$ symmetrical computation.

4. THE ITERATIVE ALGORITHM AIT

The iterative algorithm AIT for computing the DWDF elements is represented by its flowchart in figure 2.

The $W_f(n, 0)$, $W_f(n, -N/2)$ and $W_f(n, -N/4)$ DWDF elements are obtained outside the algorithm, because not any multiplications are required for their computation.

The algorithm must be repeated for each n -value. On the flowchart i represents the group of m -values and $Sk_{N/j} m$ are the partial sums that two elements the current m value and $(2*i-m)$ value. Each partial sum can be computed by the eq. (5). When all the symmetries have been into account, the current m -value and its symmetrical with respect to the $m=-N/4$ can be obtained, and a new value will be computed. When all the m -values belonging to the current group of m -values have been computed, a new group of values will be obtained.

The algorithm has been optimized in its application to reduce the number of memory accesses. Thus, the result is that the number of access is always less than the total number of elements to calculate N . Also it is optimized the required memory because of only $N/2-1$ cosines values are stored, instead of N^2 , and we have a simpler counter method, the value necessary at every time. Moreover the number of additions is also reduced, since each different partial sum is computed only once and it is stored in the corresponding elements.

At the moment, all the applications in which the DWDF is presented make use of existing Fourier Transformation algorithms for its computation, usually FFT, [1], [7]. The presented algorithm is specifically thought for DWDF computation, and improve FFT for certain values of signal samples. Table I shows the number of multiplications in our algorithm comparing with the conventional computation (N^2) method and the FFT ($N/2 \log N$) algorithm. The algorithm presented can be enhanced to reduce the number of multiplications over 25% (AITM).

6. REFERENCES

The result in the table I are referred to a single n value. To compute the DWDF elements for every n value it is necessary to apply the algorithm AIT over each n value. In this sense the AIT exploits the WK zeros, so in the total computation of the number of multiplications, the results in the table I must be multiplied approximately by $3N/4$ for the AIT, instead of N , equalizing so the number of multiplication when $N=256$ for the AITM.

5. CONCLUSIONS

In conclusion, it has been established a specific algorithm for DWDF computation, where the DWDF special properties have been exploited to the utmost, but maintaining its iterative character. In this sense the AIT algorithm is simpler than other fast algorithms, because only intern product operations are required. Furthermore the memory required to keep up the cosine values can be reduced from N^2 to $N/2+1$. Moreover the AIT algorithm practically surpasses the classical FFT algorithm nevertheless when the number of samples $N=256$, when the optimizations are took into account.

This work has been supported by the DGICYT project number pb92-0749

- [1] C. Gonzalo, J. Bescós, L.R. Berriel-Valdós and J. Santamaría, *Space-variant filtering through the Wigner Distribution Function*, Applied Optics, 28, 4, 730-736, 1989.
- [2] L. Cohen, *Time-Frequency Distributions: A review*, Proceedings of IEEE, Vol. 77, N° 7, 1989.
- [3] C. Gonzalo, J. Bescós, L.R. Berriel-Valdós and P. Artal, *Optical digital implementation of the Wigner Distribution function: use in space variant filtering of real images*, Applied Optics, 29, 17, 2569-2575, 1990.
- [4] I. Gertner and M. Shamash, *VLSI architectures to compute the Wigner Distribution*, the VLSI Journal, 9, 141-161, 1990.
- [5] C. Gonzalo, M.M. Pérez Castellanos, I. García, P. Martínez and J.M. Sanchez-Dehesa, *Parallel computation of Discrete Wigner Distribution function based on Neural Schemes*, IASTED'94, pp. 224-228, 1994.
- [6] C. Gonzalo, M.M. Pérez Castellanos, I. García, A. Díaz, *Segmentación de un Modelo Computacional Paralelo del kernel de la Distribución de Wigner Discreta*, DCIS/SICD'95, Zaragoza, Spain, Nov. 1995.
- [7] B. Boashash and P.J. Black, *An efficient Real-Time Implementation of the Wigner-Ville Distribution*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 11, 1987.
- [8] M.M. Perez-Castellanos, C. Gonzalo, A. Diaz Lavadores, I. García *Propuesta de una Unidad de Proceso Paralela para el cómputo del Kernel de Wigner en aplicaciones en tiempo real*, DCIS'96, pp 245-250, Sitges, Spain.