O' Toole, John and Boashash, Boualem (2004) Optimisation of the Realisation of Quadratic Discrete Time-Frequency Distributions as a MATLAB Toolbox. In *Proceedings International Conference on Scientific and Engineering Computation*, Singapore.

Optimisation of the Realisation of Quadratic Discrete Time-Frequency Distributions as a MATLAB Toolbox

J. O' Toole & B. Boashash

Signal Processing Research Laboratory, Gardens Point,

Queensland University of Technology, Brisbane, QLD 4001, Australia.
j.otoole@qut.edu.au, b.boashash@qut.edu.au

Abstract: An optimised time-frequency signal analysis software tool is presented here as a MATLAB toolbox. This toolbox is implemented using existing and novel methods for a specific class of Time-Frequency Distributions (TFDs) called quadratic TFDs. This is first applied to a TFD called the Wigner-Ville Distribution (WVD), as all other quadratic TFDs can be represented as a smoothed (in both time and frequency) version of the WVD. This method is then extended to the general quadratic TFD case, where the complete implementation optimises both simulation speed and memory usage. Also some quadratic TFDs can be represented in their direct form, such as the Spectrogram and Rihaczek distribution. The optimization of the implementation of these distributions is also examined.

1 INTRODUCTION

Accurate signal analysis and processing require meaningful representations of the signals involved and as most real signals (e.g. communication, biomedical and animal signals) are nonstationary (i.e. time varying frequency content) traditional time or frequency analysis is not sufficient as time/frequency content is averaged over frequency/time. The Time-Frequency Distribution (TFD) provides an analysis tool which represents the energy of the signal distributed over both time and frequency. As the implementation of TFDs compared with frequency analysis is more computationally expense an optimised implementation is desired to make the tool useful to the signal analyst. The TFDs were implemented as a MATLAB toolbox with the main routines coded in C for efficiency. The focus of this communication is to provide an optimised method of implementation for a class of TFDs called quadratic TFDs, as defined by Boashash (2003), combining existing and novel methods.

The realisation of the discrete WVD will be examined in Sec. 2.3 with the implementation discussed in Sec. 2.4. The implementation describes existing optimisation methods developed by Chester (1983) and Boashash & Black.(1987) and proposes a method to optimize simulation speed which employs a hardware adaptive Fast Fourier Transform (FFT) described by Frigo & Johnson (1998). This will then be extended to the general case of quadratic discrete TFDs in Sec. 3. The optimisation of the realisation of some special case TFDs will be examined in Sec. 3.2, namely the Spectrogram and Rihaczek distribution. Simulation results will be examined in Sec. 4.

These methods have been applied to the MATLAB toolbox known as the Time-Frequency Signal Analysis (TFSA) package developed at the Signal Processing Research Lab, Queensland University of Technology. This package provides tools for computing quadratic and polynomial TFDs, ambiguity functions, wavelet transforms and scalograms and various methods of estimating instantaneous frequency. Further information on the TFSA is available at http://www.sprc.qut.edu.au/tfsa.

2. OPTIMISATION OF THE REALISATION OF THE DISCRETE WIGNER-VILLE DISTRIBUTION

2.1. Analytic Associate

The formation of a signal without negatives frequencies is called the analytic associate of the signal $s(t)$ and results in a complex signal $z(t)$. The use of the analytic signal in the realization of quadratic TFDs eliminates unwanted energy concentrated around the origin due to interference from the positive and negative frequencies in the TFDs [Boashash (2003)]. Also by using the analytic signal in quadratic TFDs the signal can be sampled at the Nyquist rate, rather than twice the Nyquist rate which is required to avoid aliasing for real signals [Boashash (1998)].

The most computationally efficient method of calculating this analytic associate of a finite discrete signal $s[n]$, for $n = 0,1, \ldots, N - 1$, is by multiplication in the frequency domain. This can be achieved through the use of FFT routines and is computed as follows [Marple (1999)]:

1. Calculate $S[k] = \mathrm{DTF}\{s[n]\}$ for $k = 0,1, \ldots, N\text{-}1$
2. Calculate

$$K = \begin{cases} S[k] & k = 0, k/2 \\ 2S[k] & k = 1,2,\ldots,k/2-1 \\ 0 & otherwise \end{cases}$$

3. Calculate $z[n] = \mathrm{IDFT}\{ Z[k] \}$

where DFT{.} and IDFT{.} stand for the Discrete Fourier Transform and the Inverse Discrete Fourier Transform respectively. This operation will be denoted $z[n] = A\{s[n]\}$. Note that if the real signal $s[n]$ is not a radix-2 value it can be made so by zero padding, thus increasing the simulation speed of the FFT routines.

2.2 *Realisation of the Discrete Wigner-Ville Distribution*

For a given analytic signal $z[n]$ the discrete WVD is represented as

$$W_z^g[n,k] = 2\sum_{m=0}^{M-1} K_z^g[n,m]e^{-j2\pi km/M} \quad (1)$$

where $K[n,m] = g[m]z[n + m]z^*[n - m]$ and $g[m] = w[m]w^*[-m]$ represents a window function $w[\cdot]$ of odd length M, as described by Boashash & Putland (2003). This is equivalent to

$$W_z^g[n,m] = 2\,\underset{m\to k}{\mathrm{DFT}}\{K_z^g[n,m]\}. \quad (2)$$

It is easily shown that the kernel $K[\cdot]$ displays conjugate symmetry in the lag direction (represented by variable m) which is expressed as

$$K_z^g[n,-m] = K_z^{*g}[n,m]. \quad (3)$$

Therefore to form the $K[\cdot]$ kernel only half the complex multiplications are needed.

Using the conjugate symmetrical property of $K[\cdot]$ again implies that the DFT of $K[\cdot]$ results in a real matrix, as stated by Oppenheim & Schafer (1999). As suggested by Chester & Taylor (1983) for a hardware solution and fully developed for software realisation by Boashash & Black (1987) this identity can halve the number of DFT operations required. This is achieved by letting

$$K_{combined}[n,m] = K[n_1,m] + jK[n_2,m] \quad (4)$$

where n_1 and n_2 are two successive values in the discrete time sequence 0,1, … , n_1, n_2, …, N -1. The DFT of $K_{combined}[\cdot]$ yields both the DFT of $K[n_1,m]$ and $K[n_2,m]$ as

$$\begin{aligned}\underset{m\to k}{\mathrm{DFT}}\{K[n_1,m]\} &= \Re\left[\underset{m\to k}{\mathrm{DFT}}\{K_{combined}[n,m]\}\right]\\ \underset{m\to k}{\mathrm{DFT}}\{K[n_2,m]\} &= \Im\left[\underset{m\to k}{\mathrm{DFT}}\{K_{combined}[n,m]\}\right]\end{aligned} \quad (5)$$

where $\Re[\cdot]$ and $\Im[\cdot]$ represent the real and imaginary operators respectively.

2.3 *Use of the FFTW Algorithm*

As the main computational expense in the generation of the Discrete WVD (DWVD) is the DFT (as $m \to k$) required on $K[n,m]$ at each time instant n, a highly optimized Fast Fourier Transform (FFT) would be advantageous. The FFTW (acronym for Fastest Fourier Transform in the West) algorithm was selected as this was found to greatly outperform other algorithms for this particular case. The FFTW algorithm developed by Frigo & Johnson (1998) is an adaptive algorithm which adapts the computation automatically for the specific hardware architecture involved. It is the also the algorithm employed by MATLAB's 'fft' function and thus can be employed by the toolbox directly or can be compiled as C code into the time-frequency toolbox as the algorithm is freely available at http://www.fftw.org (2004).

The main advantage of using the FFTW is its performance when operating on multiple FFTs at once. For the DWVD case there needs to be N FFTs performed. Rather than perform each n FFT separately, as Fig. 1 illustrates, the simulation time is greatly reduced by operating on all the FFTs at once. This is achieved by passing the 'fft' function the whole K[·] matrix.

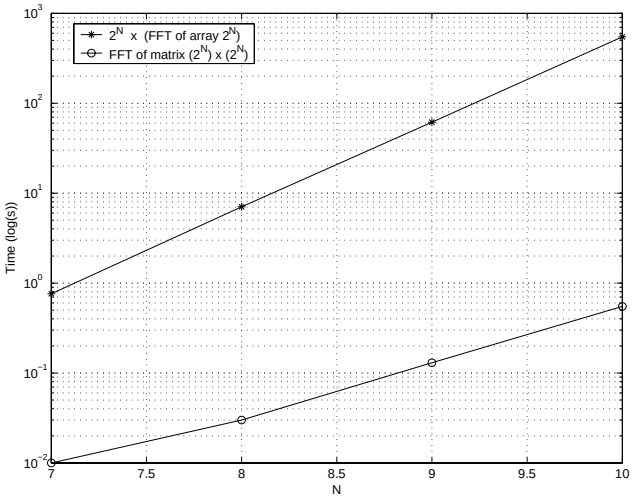

Fig.1. Plot of log simulation time versus size of 2^N arrays and $2^N \times 2^N$ matrix for comparison of different FFT algorithms. The matrix uses the FFTW algorithm (Frigo *et al*. (1998)) and the arrays use the decimation-in-time algorithm described by Oppenheim *et al*. (1999).

The disadvantage of employing the FFT on the matrix $K[\cdot]$ is that this matrix needs to be declared in memory, as opposed to the method of applying the FFT to each time instant where one vector of length M is needed. The increased memory space required is deemed to be an agreeable trade off for increased simulation speed.

2.4 *Computation of Discrete Wigner-Ville Distribution*

The optimisation of the realisation of the DWVD can be summarised as follows:

i. using the conjugate symmetrical property of $K[\cdot]$ as expressed in Eq. (3) only half the number of complex multiplications are needed,

ii. using this property again the number of FFTs required can be halved if $K[\cdot]$ is expressed as in Eq. (4) and recovered from the FFT as expressed in Eq. (5). Also only half of the kernel needs to be declared in memory,

iii. by employing the FFTW algorithm discussed in the previous Section on the whole matrix $K[:,:]$ improves simulation speed over using N FFT operations on $K[n,:]$.

Using these optimisation techniques the DWVD can be realised quite simply by the following steps:

1. Given input signal $z[n]$ defined in the region $n = 0, 1, \ldots, N - 1$ and window $w[n]$ of duration M points
2. Calculate analytic associate; $z[n] = A\{ s[n] \}$.
3. Declare matrix $K[\cdot]$ of dimension $N/2 \times M$ of type complex.
4. Form the $K[n,m] = w[m]w^*[-m]z[n + m]z^*[n - m]$ matrix.
5. Declare matrix $TFD[\cdot]$ of dimension $N \times M$ of type double.
6. Compute $K[n,k] = \mathrm{DFT}\{ K[n,m] \}$ as $m \to k$ using the FFTW routine.
7. Extract the DWVD from $K[\cdot]$ as $TFD[n,m] = \mathrm{Real}\{ K[n,k] \}$ for $n = 0, 2, \ldots, N - 1$ and $TFD[n,k] = \mathrm{Imaginary}\{ K[n,k] \}$ for $n = 1, 3, \ldots, N - 1$.

3. OPTIMISATION OF THE REALISATION OF QUADRATIC TIME-FREQUENCY DISTRIBUTIONS

3.1 *Realisation of the Discrete Quadratic Time-Frequency Distribution*

The discrete quadratic TFD is represented as [Boashash & Putland (2003)]

$$\rho_z[n,k] = 2\sum_{m=0}^{M-1} G[n,m] *_n z[n+m]z^*[n-m]e^{-j2\pi km/M} \quad (6)$$

where $G[\cdot]$ is called the time-lag kernel and $*_n$ denotes convolution in the discrete time direction n. If the window function $w[\cdot]$ is a rectangular window of length M then this TFD can be expressed as

$$\rho_z[n,k] = \underset{m\to k}{\mathrm{DFT}}\{G[n,m] *_n K[n,m]\} \quad (7)$$

where the kernel $K[\cdot]$ is as defined in Sec. 2.2. Therefore the TFD $\rho[n,k]$ can be represented as a smoothed (in both time and frequency) version of the WVD as

$$\rho_z[n,k] = \gamma[n,k] *_n *_k W_z[n,k] \quad (8)$$

where $\gamma[n,k] = \mathrm{DFT}\{\, G[n,m]\,\}$ as $(m \to k)$.

Assuming the condition that the kernel $G[n,m]$ is real and even in m (which is the case for most popular quadratic TFDs [Boashash (2003)]), then the same optimisation techniques used on the DWVD discussed in Sec. 2.4 can be applied to the realisation of $\rho[n,k]$.

3.2 *Computation of Discrete Quadratic Time-Frequency Distribution*

The optimization of the implementation of the quadratic TFDs can be summarized as follows (most overlap from the DWVD as the quadratic TFDs can be represented in terms of the WVD as in Eq. (8)):

i. if the time-lag kernel is real and even in m then the smoothed IAF kernel $R_z[n,m] = \{\, G[n,m] *_n (z[n+m]z^*[n-m])\,\}$ is conjugate symmetrical and therefore only half the number of complex multiplications are needed,

ii. assuming the property in (i) holds, then the number of FFTs required can be halved. This is achieved as for the DWVD but instead of the $\mathrm{DFT}\{K[n,m]\}$ we have $\mathrm{DFT}\{\, R[n,m]\,\}$ as $m \to k$,

iii. in the formation of the $N \times M$ $R[\cdot]$ kernel, the number of FFTs required for the convolution can be reduced from $3M$ to $5/4M$ by taking advantage of symmetrical properties of the kernels $K[\cdot]$ and $G[\cdot]$,

iv. by employing the FFTW algorithm discussed in Sec. 2.3 on the whole matrix $R[:,:]$ improves simulation speed over using N FFT operations on $R[n,:]$.

The realisation of the quadratic TFDs follows the same procedure as that for the WVD as described in Sec. 2.4 except that $K[\cdot]$ is replaced with $R[\cdot]$, the convolved product of $G[\cdot]$ and $K[\cdot]$. This convolution method will be examined in more detail here.

3.2.1 *Optimised Convolution Method*

The convolution of the $N \times M$ IAF and the time-lag kernel defined as $R[n,m] = G[n,m] *_n K[n,m]$ is currently implemented directly as an optimised time method in the TFSA package. However a new method using frequency domain techniques (using FFTs and hence the FFTW) is being developed by the authors. As mentioned in the previous Section this new FFT method requires $5/4M$ FFTs and $M/2$ complex multiplications due to the symmetrical properties of the kernels.

The formation of $R[\cdot]$ using the FFT method is calculated as follows

$$R[n,m] = \underset{u\to n}{\mathrm{IDFT}}\{\underset{n\to u}{\mathrm{DFT}}\{G[n,m]\}.\underset{n\to u}{\mathrm{DFT}}\{K[n,m]\}\} \quad (10)$$

where IDFT stands for the inverse DFT. This method requires $3M$ FFT operations and M complex multiplications. Assuming again that the time-lag kernel $G[n,m]$ is real and even in m, i.e. $G[n,m] = G[n,-m]$ then only the positive m half of the kernel is needed. As $G[n,m]$ is also symmetrical in n about $N/2$ then the DFT of kernel results in a real matrix [Oppenheim & Schafer (1999)]. Taking advantage of this identity to reduce the number of FFTs required is achieved by defining a new matrix

$$G_{combined}[n,m] = G[n,m_1] + jG[n,m_2], \quad (11)$$

for m_1 and m_2 two successive values in the discrete time sequence $0,1,\ldots, m_1, m_2, \ldots, M/2$. As with the $K[\cdot]$ kernel in Sec. 2.2 the DFT of $G_{combined}[\cdot]$ yields both the DFT of $G[n,m_1]$ and $G[n,m_2]$ as

$$\begin{aligned}\underset{n\to u}{\mathrm{DFT}}\{G[n,m_1]\} &= \Re\left[\underset{n\to u}{\mathrm{DFT}}\{G_{combined}[n,m]\}\right]\\ \underset{n\to u}{\mathrm{DFT}}\{G[n,m_2]\} &= \Im\left[\underset{n\to u}{\mathrm{DFT}}\{G_{combined}[n,m]\}\right].\end{aligned} \quad (12)$$

Therefore only $M/4$ FFTs are required for $\mathrm{DFT}\{G[n,m]\}$ as $n \to u$. As the IAF $K[\cdot]$ displays symmetry in the m direction, only $M/2$ FFTs are required for $\mathrm{DFT}\{K[n,m]\}$ as $n \to u$. After these two DFT operations are performed the two matrices are multiplied together, as in Eq. (10), which results in $M/2$ complex multiplications. Finally this resultant matrix is IDFTed to result in matrix $R[\cdot]$, requiring $M/2$ FFT operations. The full $N \times M$ matrix $R[\cdot]$ can be recovered from this matrix as $R[n,-m] = R^*[n,m]$.Thus concluding that the optimised FFT convolution method reduces the number of FFTs by a factor of 7/4 and halves the number of complex multiplications required.

3.3 *Implementation of Special Case Quadratic TFDs*

The Spectrogram and Rihaczek TFDs can both be represented as quadratic TFDs. However they both can be represented in their direct form, which is more computationally efficient.

The Rihaczek distribution is a complex energy density which can be evaluated quite simply [Boashash (2003)] requiring only 1 FFT operation and $2N$ complex multiplications. The Spectrogram on the other hand requires M FFTs and NM complex multiplications in its direct form. The discrete Spectrogram can be represented as

$$S[n,k] = \left|\sum_{m=-M/2}^{M/2} s[m+n]w[m]e^{-j2\pi km/M}\right|^2 \quad (13)$$

where $s[\cdot]$ is a signal of length N and $w[\cdot]$ represents a window function of length M. As M FFTs are required the realisation of this distribution can be optimised if the FFTW algorithm is used on a whole matrix of dimension $N \times M$. This is simply implemented as

1. Compute a kernel function $K[n,m] = z[n + m]w[m]$ for $n = 0, \ldots, N-1$ and $m = -M/2, \ldots, M/2$
2. Perform the Fourier transform $S[n,k] = \mathrm{DFT}\{ K[n,m] \}$ as $m \rightarrow k$ using the FFTW algorithm.

4. SIMULATION RESULTS

To illustrate the optimisation of the realisation of the DWVD discussed in Sec. 2.4., the simulation speed of the algorithm was compared with another DWVD implementation that uses a standard FFT algorithm (Press *et al.* (1988)). This standard method iterates the FFT M times on vectors of length N to produce a TFD of dimension $N \times M$. These results tabulated in Table 1. were computed in the TFSA package on a 3 GHz Pentium IV Linux box with 1 Gigabyte of RAM.

Table 1. Simulation speed comparisons for the DWVD using the algorithm described in Sec. 2.4 (new method) and another method (standard method) which uses iterative FFTs on vectors of data.

N (Input Signal Sample Length)	New Method (sec)	Standard Method (sec)
256	0.1435	0.1735
512	0.7019	0.9254
1024	6.0271	8.8977
2048	42.277	68.905

The optimisation can also be illustrated for the general case of quadratic TFDs. This optimised method (new method) uses the algorithm discussed in Sec. 3.2, however as the optimised convolution method described in Sec. 3.2.1 has not yet been integrated into the TFSA package this method uses a time domain method of convolution. This was compared with an existing method (standard method) which uses M iterations of FFTs on data vectors of length N. The specific case of quadratic TFD that was tested here is the modified-B distribution as described by Hussain & Boashash (2002).

Table 2. Simulation speed comparisons for the modified-B distribution using the algorithm as per Table 1.

N (Input Signal Sample Length)	New Method (sec)	Standard Method (sec)
256	0.1484	0.1821
512	0.7239	0.9524
1024	6.1018	9.1427
2048	42.456	74.630

Finally the difference in simulation speed is illustrated in Table 3. between the direct and quadratic realisation of the Spectrogram, as discussed in Sec. 3.3

Table 3. Simulation speed comparisons for the Spectrogram with signal length N. The direct method uses the FFTW algorithm.

N (Input Signal Sample Length)	Direct Method (sec)	Quadratic Method (sec)
128	0.0380	0.3410
256	0.1837	0.9006
512	0.8481	3.4047
1024	2.5662	15.775

5. DISCUSSION AND CONCLUSION

The results for the optimisation of the realisation of discrete quadratic TFDs in Sec. 4 indicate a reduction in simulation speed of approximately 30%. It is worth noting that this is a further optimisation of simulation speed over an already optimised time-frequency toolbox (TFSA package). It is expected that the realisation of the optimised convolution method discussed in Sec. 3.2.1 will result in even lower simulation speeds for the TFSA package.

To conclude an optimised methodology of realising discrete quadratic TFDs was presented using existing and novel techniques. As time-frequency analysis is computationally expense compared with standard frequency analysis a reduction in computation should help make this analysis more accessible to the signal processing community.

REFERENCES

Boashash, B. (ed.), 2003, Part I: Introduction to the Concepts of TFSAP, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference,* Part I, Chapters 1-3, Oxford: Elsevier.

Boashash, B. 1988, Note on the Use of the Wigner Distribution for Time-Frequency Signal Analysis, *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 36, 9:1518-1521.

Boashash, B., Black, P. J. 1987, An Efficient Real-Time Implementation of the Wigner-Ville Distribution, *IEEE Trans. Of Acoustics, Speech and Signal Processing,* Vol. ASSP-35, 11:1611-1618.

Boashash, B. (ed.), Putland, G. R. 2003, Discrete Time-Frequency Distributions, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference,* Part III, Chapters 6, Article 6.1, Oxford: Elsevier.

Chester, D., Taylor, J., Doyle, M. 1983, On the Wigner Distribution *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-83*, Vol. 8, 491-494.

Frigo, M., Johnson, S. G. 1998, FFTW: An Adaptive Software Architecture for the FFT, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-98*, Vol. 3, 1381-1384.

Hussain, Z. M., Boashash, B. 2002, Adaptive Instantaneous Frequency Estimation of Multicomponent FM Signals Using Quadratic Time-Frequency Distributions, *IEEE Trans. Signal Processing,* Vol. 50, 8:1866-1875.

Marple, S. L. Jr. 1999, Computing the Discrete-Time `Analytic' Signal via FFT, *IEEE Trans. Signal Processing*, Vol. 47, 9:2600-2603.

Oppenheim, A. V., Schafer R. W. 1999, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ 07458: Prentice Hall.

Press, W. H., Teukolsky, S. A., Flannery, B. P., Vetterling, W. T. 1988, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge: Cambridge University Press.