

# A STREAMING OBJECT ORIENTED IMPLEMENTATION OF THE MODAL DISTRIBUTION

*Thomas Lysaght*

Department of Computer  
Science, NUI Maynooth, Co.  
Kildare, Ireland  
Email:  
Tom.Lysaght@may.ie

*Victor Lazzarini*

Department of Music, NUI  
Maynooth, Co. Kildare,  
Ireland  
Email:  
Victor.Lazzarini@may.ie

*Joseph Timoney*

Department of Computer  
Science, NUI Maynooth, Co.  
Kildare, Ireland  
Email:  
Joseph.Timoney@may.ie

## ABSTRACT

The Modal distribution is a time-frequency distribution specifically designed to model the quasi-harmonic, multi-sinusoidal, nature of music signals and belongs to the Cohen general class of time-frequency distributions. A streaming, object-oriented implementation of the Modal distribution is presented which forms the basis for designing other members of the Cohen class. Implementation of this routine in the C++ Sound Object Library provides a fully portable tool for time-frequency analysis across multiple platforms. The theoretical background to the Cohen general class is outlined followed by an explanation of the design and implementation of the Modal distribution in the SndObj library. Suggestions for future extensions to the new Modal class and its integration with the entire library are explored.

## 1. INTRODUCTION

The Modal distribution was introduced by Pielemeier and Wakefield [1] as a member of the Cohen general class of time-frequency distributions [2] for the analysis of music signals. It is primarily a Wigner distribution, or more specifically, a smoothed pseudo-Wigner distribution (SPWD), with a kernel that takes account of the *modes* present in quasi-harmonic, multi-sinusoidal music signals. Being based on the Wigner distribution, it provides a more accurate measure of time-frequency localisation and does not suffer from the time-bandwidth trade-off inherent in spectrogram implementations. Superior accuracy in time and frequency localisation is desirable for the analysis of music signals where, for example, time resolutions of a few milliseconds are required for onset analysis and where partials may often have broadband characteristics. One drawback of the Wigner distribution is the existence of cross-terms amounting to beats between partials not existing in the original signal. The Modal distribution kernel is designed to minimize the effect of these cross terms for music signals.

A C++, object oriented, implementation would allow for integration of the Modal distribution routine with a variety of existing signal processing tools in the Sound Object Library[5]. The Sound Object Library is an object-oriented library for audio programming, written in C++. The library code is fully portable across Windows, Linux/Unix (with OSS/ALSA), Irix and MacOS X. The

SndObj library provides more than 100 classes that can be used for time- and frequency-domain signal processing, as well as sound and MIDI input/output. With its SndThread class, it can also manage audio processing threads. Particularly important for this work is the support found in the library for partial tracking and additive synthesis, which can be used to build analysis-resynthesis programs using the Modal transform for spectral analysis. The current implementation also implements the pseudo-Wigner and smoothed pseudo-Wigner distributions, and allows for further extension to the Modal class, thus, facilitating development of other members of the Cohen general class or the investigation of new time-frequency distributions through novel kernel design.

## 2. THEORETICAL BACKGROUND

Leon Cohen [2] proposed a general class of time-frequency distributions which are related through linear transformations. The set of all linear transformations of the Wigner distribution has come to be known as the Cohen general class. A two-dimensional kernel determines the linear transformation involved. Included in the Cohen general class are the spectrogram and distributions due to Riahaczek [4]. Further investigation has been carried out by O' Donovan [11]. The Cohen general class is given by:

$$C(t, \omega) = \frac{1}{4\lambda^2} \iiint s^* \left( u - \frac{1}{2} \tau \right) s \left( u + \frac{1}{2} \tau \right) \times \phi(\theta, \tau) e^{-j\theta t - j\tau \omega + j\theta u} du d\tau d\theta \quad (1)$$

where  $\phi(\theta, \tau)$  is a two dimensional kernel function which determines the distribution and its properties.  $\phi(\theta, \tau)$  typically implements filtering in time or frequency or both. The Wigner distribution in terms of the signal  $s(t)$  and the spectrum  $S(\omega)$  is given by:

$$W(t, \omega) = \frac{1}{2\pi} \int s^* \left( t - \frac{1}{2} \tau \right) s \left( t + \frac{1}{2} \tau \right) e^{-j\tau \omega} d\tau \quad (2)$$

$$= \frac{1}{2\pi} \int S^* \left( t - \frac{1}{2} \theta \right) S \left( t + \frac{1}{2} \theta \right) e^{j\theta \omega} d\theta$$

Here the kernel is unity. The Wigner distribution is said to be bilinear in the signal as the signal enters twice in the calculation and is the sum of products of the signal at past and future times, both past and future lengths of time being equal. This is in effect an autocorrelation with the lag variable,  $\tau$ , producing the time-relative-time or temporal autocorrelation function (TCF) given in (4). An important property of the Wigner distribution is that it is real with  $W^*(t, \omega) = W(t, \omega)$ . Also, the Wigner distribution gives a clear picture of the instantaneous frequency and group delay which is not the case for the spectrogram. These are important for resynthesis [1,7].

### 2.1. The Time-Relative-Time Function

The Wigner distribution is obtained by taking the Fourier transform with respect to  $\tau$  of the 2-dimensional  $b^f(t, \tau)$  autocorrelation function, given in Equation (4). The terminology used in this section is that of Poletti [6]. The temporal and spectral ranges of this function are first outlined before discussing the discrete implementation of the Wigner distribution. For a bandlimited and timelimited function:

$$f_{hw}(t) = [f(t) \circ h(t)]w(t) \quad (3)$$

the corresponding time-relative-time function is given

$$\begin{aligned} b_{hw}^f(t, \tau) &= f_{hw}\left(t + \frac{1}{2}\tau\right) f_{hw}^*\left(t - \frac{1}{2}\tau\right) \\ &= \left[ b^f(t, \tau) \circ \circ b^h(t, \tau) \right] b^w(t, \tau) \end{aligned} \quad (4)$$

Here  $\circ$  denotes convolution in the  $t$  direction. If  $h(t)$  is a  $\beta$  bandlimited function, then the function  $f_h(t) = f(t) \circ h(t)$  has a time-relative-time function:

$$b_h^f(t, \tau) = b^f(t, \tau) \circ \circ b^h(t, \tau) \quad (5)$$

that is  $\beta$  bandlimited in  $\tau$  and  $2\beta$  bandlimited in  $t$ . Also, given the function  $w(t)$  as  $\delta$  time limited, then the function  $f_w(t) = f(t)w(t)$  has a time-relative-time function:

$$b_w^f(t, \tau) = b^f(t, \tau) b^w(t, \tau) \quad (6)$$

that is  $\delta$  time limited in  $t$  and  $2\delta$  time limited in  $\tau$ . Using a discrete Fourier transform to obtain a discrete version of the Wigner distribution from the  $b^f(t, \tau)$  function, the sample rate in  $t$  of the original signal must be  $f_s > 4\beta$  in order to satisfy the sampling theorem, i.e., sampling is at twice the Nyquist rate. The sample rate in  $\tau$  is  $f_s > 2\beta$ . The continuous  $b_{hw}^f(t, \tau)$  function,

sampled in  $t$  at a rate of  $2f_s$ , and in  $\tau$  at rate  $f_s$  has the following discrete formulation:

$$b_{hw}^f\left(\frac{kt_s}{2}, nt_s\right) = f_{hw}\left(\frac{kt_s}{2} + \frac{nt_s}{2}\right) f_{hw}^*\left(\frac{kt_s}{2} - \frac{nt_s}{2}\right) \quad (7)$$

This function, then, has duration  $2T$  in  $\tau$  as shown in Figure 1.

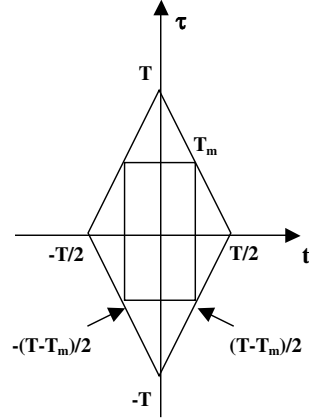


Figure 1. Extent of the windowed Time-relative-time function.

### 2.2. The Wigner Distribution

The discrete Wigner distribution [3] is written as the discrete Fourier transform of (7) with respect to  $n$  for each value of  $k$ :

$$\begin{aligned} W_{hw}^f\left(\frac{kt_s}{2}, \frac{m}{2T}\right) &= \sum_{n=-N}^{N-1} f_{hw}\left(\frac{kt_s}{2} + \frac{nt_s}{2}\right) f_{hw}^*\left(\frac{kt_s}{2} - \frac{nt_s}{2}\right) \\ &\quad \times e^{-j2\pi\left(\frac{mn}{2N}\right)} \end{aligned} \quad (8)$$

where  $k, m = -N, N-1$ . By windowing the function in Equation (7) above with the window function:

$$b^g(t, \tau) = P_{[T-T_m]}(t)g(\tau) \quad (9)$$

where  $g(\tau) = 0, |\tau| > T_m$  and  $P_{[T-T_m]}(t) = 0, |t| > \frac{(T-T_m)}{2}$ , the diamond-shaped region in the  $(t, \tau)$  plane in Figure 1 is limited to the rectangular region:

$$b_{hw}^f(t, \tau) b^g(t, \tau) = 0, |t| > \frac{(T-T_m)}{2}, |\tau| > T_m \quad (10)$$

The discrete implementation of the pseudo-Wigner distribution with a frequency smoothing window function  $w(k)$ , with length  $M = 2L-1$ ,  $w(k) = 0$  for  $|k| \geq L$  is then defined by:

$$PWD\left(n, \frac{m\pi}{M}\right) = 2 \sum_{n=-L+1}^{L-1} g(n, k) p(k) e^{-2jk \frac{mn}{N}}, \quad (11)$$

$$m = 0, \dots, M$$

where

$$p(k) = w(k)w^*(-k) \quad \text{and} \quad g(n, k) = f(n+k)f^*(n-k)$$

### 2.2.1. Cross terms

Given a music signal model as follows:

$$s(t) = \sum_{k=1}^M A_k e^{j|w_k t + \phi_k|} \quad (12)$$

where  $k$  is the partial series index,  $t$  is time, and the  $k^{\text{th}}$  term in the summation represents a partial with constant amplitude  $A_k$ , frequency  $\omega_k$ , and phase  $\phi_k$ , the Wigner distribution is:

$$W_s(t, \omega) = \int_{-\infty}^{\infty} b^f(t, \tau) e^{-j\omega\tau} d\tau = \sum_{k=1}^M A_k^2 \delta(\omega - \omega_k)$$

$$+ \sum_{k=1}^M \sum_{l=1}^M A_k A_l \cos([\omega_k - \omega_l]t + \phi_k - \phi_l)$$

$$\times \delta\left(\omega - \frac{(\omega_k + \omega_l)}{2}\right) \quad (13)$$

The partials of  $s(t)$  (auto terms) are given by the first term in (13). The second double summation indicates the cross terms, arising from products between partials, which lie between any pair of auto terms. The magnitude of the cross terms is the product  $A_k A_l$  of the amplitudes of auto terms  $k$  and  $l$  and they oscillate at a frequency,  $(\omega_k + \omega_l)/2$  equal to the difference between the frequencies of the two auto terms. For strictly harmonic signals, the cross terms form a partial series an octave below the fundamental, resulting in cross terms which fall at the same frequencies of and therefore corrupt the autoterms. Other cross terms occur at partial frequencies not in the original signal.

### 2.3. The Modal distribution

The modal distribution was designed to minimise these cross terms for music signals. The modal kernel consists two different filter functions. The time-smoothing window,  $h_{LP}(p)$ , has the effect of smoothing the cross terms in the time direction, and the frequency-smoothing window,  $g_{LP}(l)$ , implements cross term suppression in cases of frequency modulation.  $h_{LP}(p)$ , is chosen to be a low pass filter with an upper cut-off just below the minimum frequency spacing in the distribution, this being the fundamental frequency for quasi-harmonic

signals. The discrete form of the modal distribution is defined by:

$$M(n, k) = \sum_{n=-L+1}^{L-1} R_{s,l}(n, l) g_{LP}(l) e^{-\frac{j2\pi kl}{2L}} \quad (14)$$

where  $R_{s,l}(n, l) = \sum_{p=-P}^P R_s(n-p, l) h_{LP}(p)$  is the time-smoothed temporal autocorrelation function.

## 3. IMPLEMENTATION

The modal distribution (MD) class `Modal` encapsulates all processing involved in the computation of the modal distribution and is modelled on existing time-frequency classes within the `SndObj` library, namely, the phase vocoder analysis (PVA) class and the spectrogram class (`IFgram`). These classes are derived from the FFT base class, which provides the mechanisms for short-time Fourier analysis. The FFT class itself is derived from `SndObj`. Figure 2 illustrates this inheritance structure.

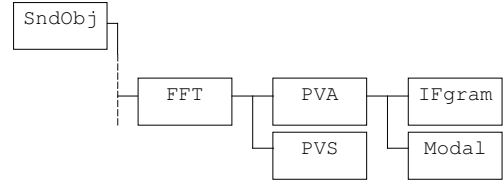


Figure 2. Inheritance diagram for the Modal class.

### 3.1. The Modal class

The design of the `Modal` class models each process needed in computing the modal distribution function. Figure 3 is a program flow diagram of the processes involved in this computation. It takes as input a sampled sound file and kernel functions as well as other necessary parameters shown in the constructor definition following.

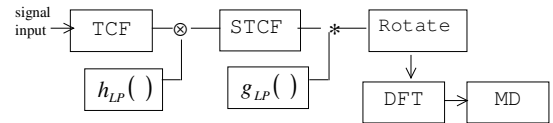


Figure 3. Stages of Modal distribution computation.

```

Modal::Modal(Table*window, Table*
swindow, SndObj* input, float scale,
int fftsize, int hopsize, float sr).
  
```

For the modal distribution computation, although cross term filtering allows for FFT sub sampling at hop periods equal to the filter length, the temporal correlation function must be computed at each sample point. Furthermore, any streaming computation of the Modal distribution needs to take account of the fact that both past and future samples are needed to compute the autocorrelation function at each sample point. Therefore, for each hop period, upon the invocation of the `Modal` class `DoProcess()` method, `hopsize` number of

samples are buffered in `m_samplesframe` implemented as a circular array. Beginning with the first signal sample at point  $-(T-T_m)/2$  (Figure 1) in the  $t$ -direction, the number of samples in each successive TCF frame, increases in the  $\tau$  direction as the number of past and future samples grows. This number keeps increasing until the number of samples in each TCF frame reaches the FFT length,  $2T_m$ . Up to this point each TCF computation requires an odd number of input samples. From this point onwards each TCF computation requires  $2T_m$  samples. The most important attributes and methods of the Modal class are shown in Figure 4.

class Modal : public PVA	
attributes	
<code>float* m_samplesframe</code>	array holds input samples
<code>float** m_tcfframe</code>	array holds TCF frames
<code>float *m_stcframe</code>	buffer holds STCF frame
<code>float** m_modframe</code>	hold MD frame
<code>Table *m_stable</code>	cross-term filter window
methods	
<code>TCFsamples(int overlap)</code>	compute TCF frames
<code>TCFconvolv()</code>	convolve TCF frames
<code>RotateRenum()</code>	rotate windowed STCF
<code>TCFAnalysis()</code>	compute FFT of STCF

Figure 4. The Modal class.

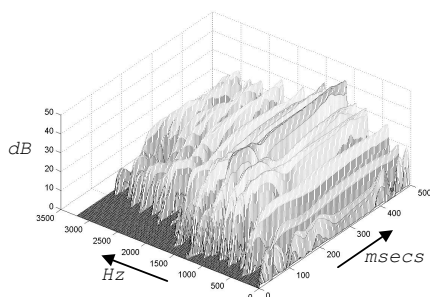


Figure 5. Modal Distribution of Trumpet note F#3

The TCF function is smoothed in the time direction in order to implement cross-term suppression. With streaming it is necessary to buffer a number of TCF frames up to the smoothing filter length. A 2-D buffer `m_stcframe` holds these frames and the smoothed result is stored in `m_stcframe`. The FFT sub sampling determines the hop period or the interval between smoothing operations carried out by the `TCFconvolv()` method. Before applying the FFT, each STCF frame is windowed with the frequency smoothing window and then rotated by switching the positive and negative halves of the window. Only the real coefficients of the FFT are used to compute the Modal distribution output amplitudes due to the distribution being real valued. The frequency positions are estimated in Hertz from the corresponding bin values. A modal distribution for the lower partials of a trumpet tone (F#3  $\approx 193\text{Hz}$ ) is shown in Figure 5.

#### 4. CONCLUSIONS AND FUTURE WORK

The streaming, object-oriented implementation of the Modal distribution within the SndObj library described in this paper provides an easily accessible tool for music signal analysis and processing. The class structure of the design makes possible the integration of the Modal routine with many of the tools necessary for sound analysis and modification such as time stretching and vocoding. In particular, the support found in the library for partial tracking and additive synthesis can be used to build analysis-resynthesis programs using the Modal transform. Furthermore, this implementation provides a convenient basis on which to implement other members of the Cohen general class and novel time-frequency distributions derived from specialised kernels. Future work will involve an analysis of the speed of computation of the Modal distribution within the library and of real-time realisations of the routine on various platforms.

#### 5. REFERENCES

- [1] Pielemeier, W. J., and Wakefield, G. "A high-resolution time-frequency representation for musical instrument signals", *Journal of Acoustical Society of America*, 99(4), Pt. 1, April 1996.
- [2] Cohen, L. *Time Frequency Analysis*. Prentice-Hall, New Jersey, 1995.
- [3] Mecklenbrauker, W. F. G., and Classen, T. A. C. M., "The Wigner Distribution – A Tool for time-frequency signal analysis; part II: discrete time signals", *Philips J. Res.*, Vol. 35, pp. 276-300, 1980.
- [4] Rihaczek, A., W., "Signal Energy Distribution in Time and Frequency", *IEEE Transactions Info. Theory*, Vol. 14, pp. 369-374, 1968.
- [5] Lazzarini, V. "The sound object library". *Organised Sound 5 (1)*. pp. 35-49, 2000.
- [6] Poletti, M. A., "The development of a discrete transform for the Wigner distribution and ambiguity function", *Journal of Acoustical Society of America*, 84(1), pp. 238-252, July 1988.
- [7] Boudreaux-Bartels, G. F. "Time-Varying Filtering and Signal Estimation Using Wigner Distribution Synthesis Techniques", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 3, June 1986.
- [8] O'Donovan, J., and Furlong, J., "A Joint time-frequency model of auditory masking", *Journal of Acoustical Society of America*, 104, 1998.