ADAPTIVE TIME–FREQUENCY RESOLUTION IN MODULATED TRANSFORM AT REDUCED DELAY

David Virette, Balázs Kövesi and Pierrick Philippe

Orange Labs / France Télécom R&D – 2, avenue Pierre Marzin – 22307 Lannion
david.virette@orange-ftgroup.com

ABSTRACT

Cosine-modulated transforms such as the Modified Discrete Cosine Transform (MDCT) are key elements in audio coding. They allow efficient energy compaction and perceptual irrelevancy reduction. The frequency localization can be adapted to the signal characteristics and fast implementations exist. All of this has made MDCT the most popular transform in audio coding.

In this paper we solve a problem never addressed in the literature: in order to enable variable MDCT sizes in communication codecs, we demonstrate how frequency resolution can be adapted on the fly without using transition windows hence decreasing coding delay. A low complexity implementation of the method is also proposed.

Index Terms— Cosine transforms, transform coding, low delay audio coding, time-varying filters

1. INTRODUCTION

In audio coding, time frequency resolution is adapted, depending on the signal characteristics, in order to deal with the non stationary characteristics of sounds. Cosine-modulated transforms have been employed for years in audio coding, and most of them use variable resolution, for example MP3, MPEG-4 AAC or Dolby Digital [1].

Among the cosine modulated transform family, the MDCT, also known as the TDAC (Time Domain Aliasing Cancellation) or MLT (Modulated Lapped Transform), is one of the simplest solutions to avoid blocking effects: an overlap factor avoids discontinuities at block boundaries. This transform has been thoroughly described in [2], [3].

1.1. MDCT

The MDCT maps a discrete signal x_{n+tM}, $0 \le n < 2M$ into M frequency components $X_{t,k}$ at instant t using the following equation:

$$X_{t,k} = \sum_{n=0}^{2M-1} x_{t,n}\, p_{k,n} = \sum_{n=0}^{2M-1} x_{n+tM}\, p_{k,n} \qquad (1)$$

for $0 \le k < M$, with

$$p_{k,n} = w(n)\, c_{k,n},$$
$$c_{k,n} = \sqrt{\frac{2}{M}} \cos\left(\frac{\pi}{M} \left(n + \frac{M+1}{2} \right) \left(k + \frac{1}{2} \right) \right), \qquad (2)$$

where $p_{k,n}$ are the basis functions for the direct and inverse transforms, and $w(n)$ denotes the analysis window. It is the equivalent of the low pass prototype filter in the filterbank terminology. In order to recover the original sequence x, an inverse transform is applied according to

$$\tilde{x}_{t,n} = \sum_{k=0}^{M-1} X_{t,k}\, p_{k,n} \qquad (3)$$

for $0 \le n < 2M$. The $2M$ terms of $\tilde{x}_{t,n}$ are recovered using only M frequency components of $X_{t,k}$, and hence cannot exactly represent the original $x_{t,n}$ signal. In particular, they contain aliasing terms, consisting in unwanted components reverted in time. These aliasing terms are cancelled using a combination of two consecutives frames such that:

$$\hat{x}_{t,n} = \hat{x}_{n+tM} = \tilde{x}_{t-1,n+M} + \tilde{x}_{t,n} \qquad (4)$$

for $0 \le n < M$. For perfect reconstruction (PR) the window $w(n)$ needs to satisfy the following conditions to ensure that the aliasing is cancelled:

$$\begin{cases} w(n) = w(2M-1-n) \\ w^2(n) + w^2(M+n) = 1 \end{cases} \qquad (5)$$

A common window that satisfies the PR conditions is the sine window [3] defined by:

$$w(n) = -\sin\left[\left(n + \frac{1}{2} \right) \frac{\pi}{2M} \right] \qquad (6)$$

The MDCT considers overlapping frames with M samples, resulting in M frequency components, hence corresponding to a maximally decimated filterbank. The number of frequency components can be made variable over time. In this case, the technique described below is known as time-varying MDCT [4], [5].

1.2. Time varying MDCT

In order to adapt the frequency resolution to the signal, the MDCT can switch between transform sizes. This technique is known as block switching or window switching.

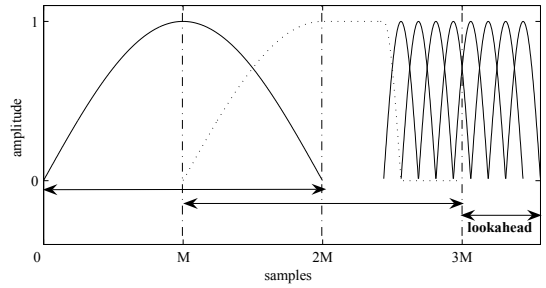

Fig. 1. Combination of windows: long window, transition window (dashed line), and eight short windows.

Let us consider a transition from a MDCT of size M to a MDCT of size M_s. This configuration is widely used in audio coding, especially in MPEG-4 AAC [1]. To deal with transient sounds: a long transform with M=1024 is normally used, excepted for transient sounds such as attacks (e.g. castanets, speech plosives) which are processed with eight successive short MDCT of size M_s=128. The process is illustrated in Figure 1.

In order to maintain the PR property, care has to be taken when frequency resolution changes. A commonly adopted solution for maintaining PR uses transition windows, performing a size M transform with special asymmetric weighting functions between long and short windows.

Transition windows are made of four portions [4], [5]: in order to cancel the time domain aliasing with the preceding long window, the transition window uses the long weighting function in its first half. The second portion contains a flat portion, i.e. a constant gain, followed by the short reverted weighting function that aims at removing the aliasing introduced by short windows. Finally a zero tail is added in the zone handled by the short windows sequence. Using this simple construction method, the aliasing components are cancelled.

Using such a method, transitions between long and short blocks have to be anticipated since transition windows need to be inserted. This anticipation comes at the price of an additional look-ahead delay: in the example presented in Figure 1, the switch to a shorter resolution is based on the knowledge of M/2 samples ahead; this is a source of additional delay for communication codecs. Note that the transition from short to long windows does not introduce extra delay.

1.3. Proposed solution

In this paper a general solution allowing direct transition between two MDCT sizes is presented. Hence, the coding delay introduced by window switching is removed. And this technique can be applied in low delay communication codecs such as MPEG-4 Low Delay AAC [6].

The solution can be expressed as a post processing operation: the direct transform directly switches from long to short resolution, i.e. without transition windows. The inverse transform is followed by a post processing operation that removes the aliasing components as shown in section 2.2. It is shown that PR can be achieved. The complexity of the proposed method is finally evaluated in section 3.

2. COMPENSATION SCHEME

In this section, we describe the compensation scheme, used at the inverse transform, to cancel the time domain aliasing terms when no transition windows are applied at the direct transform.

2.1. Matrix notation

In this paragraph, we introduce the direct and inverse transform of two consecutive frames. At the instant t-1, we present the transform operation for a long window and at the instant t for eight short windows. Boldface letters indicates vectors and matrix with elements defined in 1.1. The symbols $\mathbf{I}_M$ and $\mathbf{J}_M$ denote the $M \times M$ identity and counter-identity matrix. **diag** is an $M \times M$ diagonal matrix, and the operator $^{\mathrm{T}}$ denotes the transpose operation.

At instant t-1, similar to Eq. (1) the direct transform, is expressed by

$$\mathbf{X}_{t-1} = \mathbf{P}\ \mathbf{x}_{t-1} \tag{7}$$

where the matrix $\mathbf{P}$ is the transform matrix

$$\mathbf{P} = \mathbf{C}\ \mathbf{diag}(\mathbf{w}) = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,2M-1} \\ p_{1,0} & p_{1,1} & & \vdots \\ \vdots & & \ddots & \vdots \\ p_{M-1,0} & \cdots & \cdots & p_{M-1,2M-1} \end{bmatrix} \tag{8}$$

and $\mathbf{w} = [w(0), w(1), \ldots, w(2M\text{-}1)]^{\mathbf{T}}$. $\mathbf{x}_{t-1} = [x_{t-1,0}, x_{t-1,1}, \ldots, x_{t-1,2M-1}]^{\mathbf{T}}$ is the input signal of length $2M$. The corresponding inverse transform is defined by

$$\tilde{\mathbf{x}}_{t-1} = \mathbf{P}^{\mathrm{T}}\ \mathbf{X}_{t-1} \tag{9}$$

The PR condition is derived by cascading the direct and inverse transforms:

$$\tilde{\mathbf{x}}_{t-1} = \mathbf{P}^{\mathrm{T}}\mathbf{P}\ \mathbf{x}_{t-1} = \mathbf{diag}(\mathbf{w})\ \mathbf{C}^{\mathrm{T}}\mathbf{C}\ \mathbf{diag}(\mathbf{w})\ \mathbf{x}_{t-1} \tag{10}$$

where

$$\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{diag}(\mathbf{w}) \begin{bmatrix} \mathbf{I}_M - \mathbf{J}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M + \mathbf{J}_M \end{bmatrix} \mathbf{diag}(\mathbf{w})$$
$$\mathbf{P}^{\mathrm{T}}\mathbf{P} = \begin{bmatrix} \mathbf{U_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U_1} \end{bmatrix} \tag{11}$$

with

$$\mathbf{U_0} = \mathbf{diag}(\mathbf{w}_0)\left[\mathbf{I}_M - \mathbf{J}_M\right]\mathbf{diag}(\mathbf{w}_0) \tag{12}$$

and

$$\mathbf{U_1} = \mathbf{diag}(\mathbf{w}_M)\left[\mathbf{I}_M + \mathbf{J}_M\right]\mathbf{diag}(\mathbf{w}_M) \tag{13}$$

where $\mathbf{w}_0 = [w(0), w(1), \ldots, w(M\text{-}1)]^{\mathbf{T}}$ and $\mathbf{w}_M = [w(M), \ldots, w(2M\text{-}1)]^{\mathbf{T}}$ are two portions of length M of the window $\mathbf{w}$.

By adding two consecutive length M reconstructed signals, the PR condition can be expressed as:

$$\mathbf{U_0} + \mathbf{U_1} = \mathbf{I}_M \tag{14}$$

By replacing $\mathbf{U_0}$ and $\mathbf{U_1}$ in equation (14) it comes:

$$\mathbf{diag}\left(\mathbf{w}_M\right)\mathbf{J}_M\mathbf{diag}\left(\mathbf{w}_M\right) - \mathbf{diag}\left(\mathbf{w}_0\right)\mathbf{J}_M\mathbf{diag}\left(\mathbf{w}_0\right) = \mathbf{0} \tag{15}$$

and

$$\mathbf{diag}\left(\mathbf{w}_0\right)\mathbf{diag}\left(\mathbf{w}_0\right) + \mathbf{diag}\left(\mathbf{w}_M\right)\mathbf{diag}\left(\mathbf{w}_M\right) = \mathbf{I}_M \tag{16}$$

Eq. (15) and (16) turns directly in Eq. (5) for symmetric windows.

The next transform, at instant t, is processed with eight short blocks, Eq. (7) and (9) for the direct and inverse transform are similar with a reduced number of subband M_s and applied eight times for processing M samples. The equivalent M size matrix formulation for those eight short blocks can be written in a $\mathbf{P}$ matrix fashion.

$$\tilde{\mathbf{x}}_t = \mathbf{P_s}^{\mathbf{T}}\mathbf{P_s}\ \mathbf{x}_t \tag{17}$$

$$\mathbf{P_s}^{\mathbf{T}}\mathbf{P_s} = \begin{bmatrix} \mathbf{U_{s0}} & \mathbf{0} \\ \mathbf{0} & \mathbf{U_{s1}} \end{bmatrix} \tag{18}$$

$$\mathbf{U_{s0}} = \mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right)\begin{bmatrix} \mathbf{0}_{\frac{M-M_s}{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M_s} - \mathbf{J}_{M_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\frac{M-M_s}{2}} \end{bmatrix}\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right) \tag{19}$$

$$\mathbf{U_{s1}} = \mathbf{diag}\left(\mathbf{w}_{\mathbf{s},M_s}\right)\begin{bmatrix} \mathbf{I}_{\frac{M-M_s}{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M_s} + \mathbf{J}_{M_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{\frac{M-M_s}{2}} \end{bmatrix}\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},M_s}\right) \tag{20}$$

where $\mathbf{w_s}$ is the weighting window for the short MDCT transform. In that case, the eight short blocks are represented by an equivalent long window given in Figure 2.

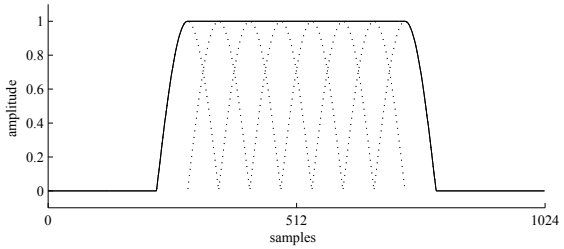

Fig. 2. Eight short windows, each of size $2M_s$ (dashed line), and equivalent $2M$ size window (solid line).

2.2. Perfect reconstruction during resolution changes

Here we demonstrate how aliasing can be suppressed using two appropriate weighting functions $\mathbf{w_1}$ and $\mathbf{w_2}$ applied after inverse transformation. This operation is called *compensation*. It can be seen from Eq. (19) and (20) that the last (M-M_s)/2 samples of the input vector $\mathbf{x}_t$ are directly recovered from the combined operation of direct and inverse transform: both $\mathbf{U_{s0}}$ and $\mathbf{U_{s1}}$ contain a portion of identity matrix. Hence, only the (M+M_s)/2 first elements need a post processing to ensure PR by aliasing elimination. In the case of direct transition the equation

$$\mathbf{U_1} + \mathbf{U_{s0}} \neq \mathbf{I}_M \tag{21}$$

cannot achieve PR.

However, we demonstrate that using an appropriate set of weighting functions $\mathbf{w_1}$ and $\mathbf{w_2}$, and an anti–aliasing matrix A, PR can be guaranteed i.e.:

$$\mathbf{diag}\left(\mathbf{w_1}\right)\mathbf{U_1} + \mathbf{diag}\left(\mathbf{w_2}\right)\mathbf{A}\mathbf{U_{s0}} = \mathbf{I}_M \tag{22}$$

The $M \times M$ matrix $\mathbf{A}$ aims at cancelling the aliasing:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{J}_{\frac{\mathbf{M}-\mathbf{M_s}}{\mathbf{2}}} \\ \mathbf{0} & \mathbf{I}_{\mathbf{M_s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\frac{\mathbf{M}-\mathbf{M_s}}{\mathbf{2}}} \end{bmatrix} \tag{23}$$

Rewriting Eq. (22), in a way similar to Eq. (15) and (16) gives the relationships for the weighting functions that ensure PR

$$\mathbf{diag}\left(\mathbf{w}_1\right)\mathbf{diag}\left(\mathbf{w}_M\right)\mathbf{J}_M\mathbf{diag}\left(\mathbf{w}_M\right) - \mathbf{diag}\left(\mathbf{w}_2\right)\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right)\begin{bmatrix} \mathbf{0} & \mathbf{J}_{\frac{\mathbf{M}+\mathbf{M_s}}{\mathbf{2}}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right) = \mathbf{0} \tag{24}$$

and

$$\mathbf{diag}\left(\mathbf{w}_1\right)\mathbf{diag}\left(\mathbf{w}_M\right)\mathbf{diag}\left(\mathbf{w}_M\right) + \mathbf{diag}\left(\mathbf{w}_2\right)\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right)\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{\frac{\mathbf{M}+\mathbf{M_s}}{\mathbf{2}}} \end{bmatrix}\mathbf{diag}\left(\mathbf{w}_{\mathbf{s},0}\right) = \mathbf{I}_M \tag{25}$$

This system is solved in $\mathbf{w_1}$ and $\mathbf{w_2}$ by a set of $2M$ linear equations with $2M$ unknown variables. As a result, the analytical expressions of $\mathbf{w_1}$ and $\mathbf{w_2}$ are obtained as functions of $\mathbf{w}$ and $\mathbf{w_s}$, as shown below.

2.3. Compensation windows

After some manipulation, the weighting functions are given by:

$$w_{1,n} = \frac{1}{w^2\left(M+n\right)},\ w_{2,n} = -\frac{w\left(n\right)}{w\left(M+n\right)} \tag{26}$$

for $0 \le n < \dfrac{M-M_s}{2}$, and

$$w_{1,n} = \frac{w\left(n\right)w\left(M-1-n\right) - w_s\left(m\right)w_s\left(M_s-1-m\right)}{d(n)},$$

$$w_{2,n} = \frac{w_s\left(M_s-1-m\right)}{d\left(n\right)} \tag{27}$$

for $\dfrac{M-M_s}{2} \le n < \dfrac{M+M_s}{2}$ and $m = n - \dfrac{M-M_s}{2}$

with $d(n) = w(M\text{-}1\text{-}n)\left[w(M\text{-}1\text{-}n)\ w_s(M_s\text{-}1\text{-}m)+w(n)w_s(m)\right]$

2.4. Compensation algorithm

Reverting to the sample notation, the compensation algorithm of Eq. (22), is split in two parts. The first part relates to the reconstruction of the samples in the interval $0 \le n < (M\text{-}M_s)/2$, and is expressed by:

$$\hat{x}_{t,n} = w_{1,n}\tilde{x}_{t-1,n+M} + w_{2,n}\hat{x}_{t,M-1-n} \quad (28)$$

where $\tilde{x}_{t-1,n+M}$ are recovered from the long window, and $\hat{x}_{t,M-1-n}$ represent the reconstructed signal obtained from the overlapped short windows.

The second part of the algorithm deals with the samples $(M\text{-}M_s)/2 \le n < (M+M_s)/2$. In this case, the reconstruction is given by

$$\hat{x}_{t,n} = w_{1,n}\tilde{x}_{t-1,n+M} + w_{2,n}\tilde{x}^{s}_{t,n} \quad (29)$$

with $\tilde{x}^{s}_{t,n}$ coming from the first short window.

3. APPLICATION AND COMPLEXITY

Figure 3 (b) shows the direct transition between long and short windows in the direct transform whereas Figure 3 (c) shows the equivalent windows used for the inverse transform. These windows directly replace the traditional windows in the inverse transform step shown in Figure 3 (a). It can be seen from Eq. (26) and (27), that the post processing cancels the windows of the inverse transform. Hence, combining the inverse transform and the post processing is simple. These combined weighting functions lead to a complexity decrease.

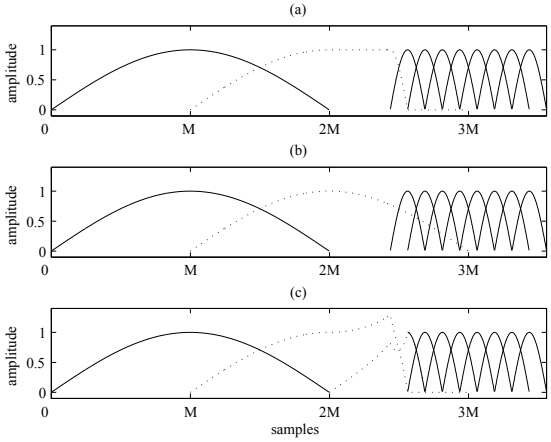

Fig. 3. Illustration of various window transitions. (a) Traditional window sequence: long window, long-short transition window (dashed line), eight short windows. (b) Direct transition between long (dashed line) and short (solid line) windows at the direct transform. (c) Compensation scheme for the inverse transform: in dashed line, the modified part of the long and first short window.

As described above, the proposal essentially relies on a compensation scheme in the inverse transform. Hence, the compensation scheme replaces the traditional window weighting by a new weighting function. This compensation implies one multiplication and one addition for each compensated sample. For a frame of M samples, it corresponds to $(M+M_s)/(2M)$ additional operation per sample. This approximates to half a MAC per sample, if $M_s << M$.

In terms of memory storage, the compensation windows, representing $M + M_s$ values, can be either stored in ROM or computed once at initialization.

4. CONCLUSION

In this paper, a novel technique that allows low delay time-varying MDCT has been introduced. It was demonstrated how the perfect reconstruction property of the time varying MDCT can be preserved. The additional delay, traditionally added at the encoder for anticipating transition windows is avoided using a compensation processing at the decoding side.

The complexity evaluation of the proposed method shows that this delay reduction comes at a negligible computational cost.

Albeit illustrated on the MDCT, this formalism extends to other modulated transforms such as ELT (Extended Lapped Transform) [3].

This method has been successfully applied to MPEG-4 Low Delay AAC [7] and it was demonstrated that time varying MDCT can be introduced in this coding scheme without additional delay.

4. REFERENCES

[1] M. Bosi, R.E. Goldberg, “Introduction to digital audio coding and standards,” Kluwer academic publishers, 2003.

[2] J. P. Princen, A. B. Bradley, “Analysis/Synthesis filter Bank Design Based on Time Domain Aliasing Cancellation,” *IEEE Trans. on ASSP*, Vol. 34, No. 5, October 1986.

[3] H.S. Malvar, “Signal Processing with Lapped Transforms,” Norwood , MA: Artech House, 1992.

[4] B. Edler: “Codierung von Audiosignalen mit überlappender Transformation und adaptiven Fensterfunktionen,” *Frequenz*, 1989.

[5] S. Shlien, “The modulated lapped transform, its time-varying forms, and its applications to audio coding standards,” *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 4, pp. 359 – 366, July 1997.

[6] E. Allamanche, R. Geiger, J. Herre, T. Sporer “MPEG-4 Low Delay Audio Coding Based on the AAC Codec,” in *Proc. 106th AES Conv.*, April 1999.

[7] P. Philippe, D. Virette “Proposed Core Experiment for Enhanced Low Delay AAC,” *Contribution m14237*, 79th MPEG meeting, Marrakech, January 2007.