Practical Selection of a Basis from a Redundant Tree Structure

Oscar N. Bria

o.bria@ieee.org

Dpto de Informática, Cs. Ex., UNLP, Argentina

Abstract

The problem of selecting a basis within a redundant tree structure naturally arises in the frameworks of wavelet packet transform (WPT) and cosine packet transform (CPT) for signal representation.

The "actual" tree has to be consider before performing that selection using suitable information cost functions for the branches.

The "actual" redundant tree has many constraints. Among them is the bit overhead for coding explicitly the name of the resulting non redundant (prune) tree.

Besides, from computational or real time constraints it is convenient to perform sub-optimal top-down tree computation/search instead of the optimal bottom-up tree search after computing the complete tree.

Even when a optimal coding of the tree remains an open join coding problem, a practical selection algorithm is reported here that takes the above and others considerations into account.

The algorithm has been implemented an tested in the framework of a wavelet based audio coding system.

Variations of the algorithm can be used in many practical situations where a subset from an redundant structure has to be chosen with the constraint of minimal bit cost.

The Problem

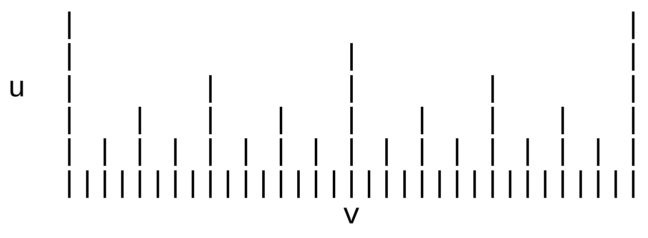
When there is a choice of bases for the representation of a signal, then it is possible to seek the best one by some criterion. If the choice algorithm is sufficiently cheap, then it is possible to assign each signal its very own adapted basis of waveforms. If this basis description is efficient, then that signal has been compressed.

The problem of selecting a basis within a redundant tree structure naturally arises in the frameworks of wavelet packet transform (WPT) and cosine packet transform (CPT) or Malvaar adapted transform, for signal representation. The so called actual tree has to be consider before performing that selection using suitable information cost functions instead of the classical entropy for selecting branches or subbands. Besides, from computational or real time constraints it is convenient to perform sub-optimal top-down tree computation/search instead of the optimal bottom-up tree search after computing the complete tree.

Admissible Bases

The next figure is a schematic representation of the library of wavelet coefficients presenting a tree arrangement.:

- The levels of the tree are u .
- The branches v correspond to blocks of coefficients of dyadic sizes.
- The refinement progress from top to down.
- The block on top is the N -size frame of the signal itself.



Fig, 1

The complete tree, with all its branches, is a redundant representation of the frame. The conditions to choose a complete orthonormal basis representation are the following:

1. Every column contains exactly one element.
2. Elements in a single row appear in contiguous blocks of 2^k elements, where $0 < k < n$ is an integer.
3. Row blocks begin an integer multiple of their length from left edge of the array.

Searching the Tree

Given and additive information cost function associated to the library of bases (see [Wic1]). If that library is a tree with finite depth, then we can find the best basis by computing the information cost of a vector in each node of the tree and comparing children to parent starting from the bottom (bottom-up search). This is actually a low complexity search since the additive property of additive information cost functions means that each node is examined only twice: once as a child and once as a parent. To search an L-level decomposition of an $N = 2^L$ sample periodic signal therefore requires just $O(N)$ comparisons.

Taswell (see [Tas2]) has shown that even for non additive cost function it is possible to apply a top-down search algorithm for finding a called near - best basis from a redundant tree. The advantage of this approach is a reduction in computations with non practical reduction in quality of compression and distortion. This algorithm is called sub-optimal top-down tree computation/search instead of the optimal bottom-up tree search after computing the complete tree. The top-down bases are selected in the opposite direction by depth-first top-down searches with the search terminated as soon as the cost of the children blocks or branches is greater than the cost of the parent block or branch.

The Actual Tree

The actual redundant tree has an associated climber. There is a variable number of bits for full coding the tree (each branch has a fixed length and a coding of its coefficients using its own quantizer/s[1]. And an a priori fixed number of bits for full coding the climber (each one coding the characteristics of each associated branch quantizer/s). There also exist a bit overhead for coding explicitly the name of the resulting prune[2]. As a mater of fact,

- the prune tree name,
- the coefficients of the prune tree branches,
- and the information of the prune associated climber branches,

should be jointly coded/quantized[3] [CrRa].

A Simple Approach

Consider one scalar quantization coder per branch with two parameters Z_k and Q_k for all the coefficients in branch k^{th} [Tas1]. A simpler approach enables the use of the same values of *Z* and Q for all N transform coefficients considered together as one collection instead of separated branches.

What remain particular in each branch is β_n, the number of bits used per quantized coefficient. Let us consider β_k, the maximum number of bits used in each branch and actually use it for coding within the branch. β_k has to be coded in the associated climber tree but the coding of *Z* and Q is reduced to the implicit or explicit specification of a couple of numbers per each frame of length N.

[1] Only one scalar quantizer, many scalar quantizers, or a vector quantizer.

[2] Non redundant tree.
[3] The three kind of information can be a priori freely interleaved. Besides, the prune tree branches can be coded using a lousy strategy, i.e., that joint coding can be see as a vector quantization.

The above simplifications have relaxing but do not drop the necessity for joint coding which remain an open problem.

Let us count the number of bits associated to each piece of information:

- If the number of coefficients of each tree branch is l_k then, each tree branch consumes $t_k = l_k \beta_k$ bits.
- Let $c_k = c = \log_2\left(\left\lceil \max_k \{\beta_k\} \right\rceil\right)$, the fix number of bits per climber branch.
- Let m, the fix total number of bits for coding the name of the prune tree[4] and eventually Z and Q .

Every possible prune tree T_i has a number of branches L_i . If B_i is the total number of bits for coding any possible prune tree, we are interested in:

$$\min_i \left\{ B_i = m + cL_i + \sum_{k \in T_i} t_k \right\}$$

Eq. 1

The Split Condition

When implementing a top-down computation/search tree, the split condition of the branches is,

$$b_{u+1,v} < b_{u,v}$$

$b_{u,v}$ is the number of bits of branch at row u and column v (see Fig, 1).

$$\frac{\beta_{u+1,v} + \beta_{u+1,v+\delta_u}}{2} < \beta_{u,v} - \frac{c}{l_u}$$

Eq. 2

In the last equation δ_u is the necessary gap between v's as a function of deep level u . Due to the fact that all the numbers are positive defined a necessary extra condition is,

[4] This number is a function of N (see [Whi2]).

$$\min_u \left\{ l_u \right\} > \frac{c}{\min_{u;v;\beta_{u,v} > 0} \left\{ \beta_{u,v} \right\}}$$

Eq. 3

which normally impose a limit to the deep of the tree for a given N.

The Practical Issue

For a given N there are $L = \log_2(N)$ possible level in the tree. If D is the number of quantization bits of the original signal it is reasonable to assume $c = \log_2(D)$. If $min_{u;v;\beta_{u,v}>0}\left\{\beta_{u,v}\right\} = 2$ then,

$$\min_u \left\{ l_u \right\} > \frac{\log_2(D)}{2}$$

Let $D = 16$, $N = 1024$

$$\min_u \left\{ l_u \right\} \geq 2$$

If the code rate is constant and known in advance, say R bits/frame, then for a particular branch k^{th} Eq. 2 can be rewritten.

Let us first consider the following definitions,

C_p	: Quality of coding for parent
C_i	: Quality of coding for left child
C_d	: Quality of coding for right child
C_c	: Worse quality of coding for children $C_c = min\left\{C_i, C_d\right\}$
B	: Total number of available bits at present level
H	: Header information number of bits
B_x	: Bits Available for coding at present level
b_x	: Bits/coeff. required for coding at this particular level
l	: Length of the parent branch (number of coefficients)
$\frac{l}{2}$	: Length of each child branch (number of coefficients)

The following relation hold:

$$B_p = B - H$$
$$B_c = B - 2H = B_i + B_d = B_p - H$$
$$B_i = B_c b_i / \left(b_i + b_d\right)$$
$$B_d = B_c b_d / \left(b_i + b_d\right)$$

The so called quality of coding is simply the number of bits available per bit required ,

$$C_x = \frac{B_x}{b_x \cdot l}$$

resulting for parent and combined children

$$C_p = \frac{B_p}{b_p \cdot l} \qquad\qquad C_c = \frac{B_c}{b_c \cdot l}$$

If our goal is to meet the best quality condition in every step of the computing /searching algorithm,

```
if   $C_c > C_P$ ,
     Select Children;
elseif
     Select Parent;
end
```

The children quality selection condition can be rewritten

$$\text{if} \qquad \frac{B_c}{b_c} > \frac{B_p}{b_p} \qquad \Rightarrow \qquad \text{children}$$

or

$$\text{if} \qquad \frac{B - 2H}{\frac{b_i + b_d}{2}} > \frac{B - H}{b_p} \qquad \Rightarrow \qquad \text{children}$$

We can split the above inequality into two factors

$$F1 = \frac{b_p}{\left(b_i + b_d\right)/2} \qquad ; \qquad F2 = \frac{B - 2H}{B - H}$$

if $F1 * F2 > 1 \quad \Rightarrow$ children

- F1: Quality amplification factor of the dyadic decomposition**.** If F1 is greater than one there is a chance for selecting the children instead of the parent.
- F2: Quality attenuation factor of the "real" binary coding**.** It takes values from 0 to 1.

Experiments and Discussion

The algorithm has been implemented and used for audio coding in **WinWave** by García, Quentrequeo, and Otero [GaQO]. Some results are summarized here:

- In WinWawe a top-down tree computation/search is implement in a frame by frame basis according to the practical issues above explained.
- In WinWave the allocation bit is made on the basis of maximum magnitude in each branch weighted by psychoacoustic factors depending of spectral position.
- In comparisons made in WinWave between top-down and bottom-up search there are not detectable differences in quality.
- Computational time is significant reduced when the top-down approach s used.

I present here a numerical implementation issue related to the practical use of libraries of orthogonal bases. Even when it is just a detail it is a very important one when implemented a real compression algorithm using wavelet packets on a real computer machine.

As a final remark, the joint coding approach has to be explored. A good reference to begin with that approach is [CrRa].

References

[GaQO] "Algoritmo para el Procesamiento de Señales usando Filtrado por Octavas",
Trabajo de Grado Licenciatura en Informática.
García, Quentrequeo, and Otero
Facultad de Ciencias Exactas, UNLP, 1996.

[CrRa] "Joint Thresholding and Quantizer Selection for Transforming Image Coding:
Entropy - Constrained Analysis and Applications to Baseline JPEG.
M. Crouse and K. Ramchandran.
IEEE Transaction on Image Processing, February 1997.

[Tas1] "Satisfying Search Algorithms for Selecting Near - Best Bases in Adaptive Tree - Structured Wavelet Transforms."
Carl Taswell.
Stanford University, 1995

[Tas2] "Top - Down and Bottom - Up Tree Search Algorithm for Selecting Bases in Wavelet Packet Transforms."
Carl Taswell.
Stanford University, 1995.

[Whi1] Adapted Wavelet Analysis from Theory to Software."
Mladen Victor Whickerhouser.
IEEE Press, 1994.

[Whi2] "Acoustic Signal Compression with Wave Packets"
Mladen Victor Whickerhouser.
Yale University, 1989.