ON ADAPTIVE WAVELET PACKETS

*Ricardo L. de Queiroz** and *K. R. Rao*

Electrical Engineering Department
University of Texas at Arlington
Box 19016, Arlington, TX, 76019
queiroz@eepost.uta.edu or eekrr521@utacnvx.uta.edu

Abstract

An algorithm is presented for fast implementation of time-varying wavelet packets maintaining perfect reconstruction throughout transitions. It is based on filter banks known as the extended lapped transforms in an association following the paths of a binary-tree. Methods for pruning or expanding the branches of the tree are presented and a discussion on adaptation issues is carried.

I Introduction

Recently, concepts such as time-frequency representation and wavelet packets have gained a great deal of attention in the field of digital signal processing [1] because of the ability to trade time and frequency resolutions, maintaining orthogonality, and perfect reconstruction (PR). It is also well known that orthogonal wavelet packets can be designed by hierarchical association of PR paraunitary filter banks [2]. In these, synthesis filters are time reversed versions of the analysis ones. In the 2-band case, the two impulse responses of the analysis and synthesis filters, denoted as $f_m(n,k)$ and $g_m(n,k)$, respectively, are related by

$$p_{mn,k} = g_m(n,k) = f_m(L-1-n,k) \quad (1)$$

for $m = 0,1$ and $n = 0,1,\ldots,L-1$, where $L = 4K$, and K is the overlapping factor. In this paper we will concentrate on binary trees. However, this concept can be easily extended to m-ary trees.

The extended lapped transform (ELT) is certainly a good way of implementing hierarchical associations [3, 4], since it (i) is paraunitary; (ii) has a fast implementation algorithm; (iii) can be defined for several overlapping factors and number of channels; (iv) is reasonably regular (mainly if we chose the parametric design [3]).

*This work was supported in part by CNPq, Brazil, under Grant 200.804-90-1.

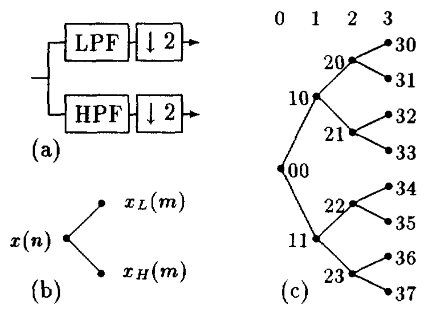

Figure 1. The binary tree notation. a)2-band filter bank; b)its equivalent representation in a tree; c) labelling of nodes in the tree.

II Binary trees notation

If the same paraunitary filter bank is used as the analysis cell for each stage, it is sufficient to describe the paths of the tree to completely describe the whole analysis-synthesis system. With the aid of Fig. 1 we present a more convenient notation. In Fig. 1a we see a 2-band maximally decimated filter bank with its low- and high-pass filters, as well as subsamplers. This system will be represented here by tree nodes and branches (Fig. 1b), where the signal flows in the nodes and the branches represent filters and decimators. We label the nodes in the tree as η_{ij}, where i is the stage number and j is the number of the node in that stage, just as indicated in Fig. 1c, for 3-stage binary tree. The familiar parenthood notation is used, so that node η_{ij} is parent of nodes $\eta_{i+1,2j}$ and $\eta_{i+1,2j+1}$, but child of $\eta_{i-1,j\oslash 2}$ ($\oslash \equiv$ integer division). We denote x_{ij} the signal flowing in η_{ij}, while x_{00} is the original input signal. As a remark, the number of a node in a level does not correspond to an increasing frequency ordering of bands.

The idea is to adaptively reshape the tree. In this case, it is convenient to define an infinite number of stages and an activity map. This map indicates if the node is active (its signal is being processed as an input to a filter bank) or not. Let $a_{i,j}(n) = 1$ denote an active node ij, being 0 otherwise. The rightmost node in a path will be called instantaneously virtual end node (IVEN). All nodes with a childhood relation to an IVEN are inactive at instant n, and clearly an IVEN is inactive. To prune a branch of the tree, it is sufficient to bypass the transform applied to the parent node of an IVEN which will become a new IVEN, de activating the other. The recurrence of this procedure applied to the desired branches will bypass the signal from the resulting IVEN to the right, pruning the tree. The inverse procedure expands the tree, activating nodes and branches.

III PR Time-Varying ELTs

We do not want to extend the derivations for a PR time-varying form for an ELT. The basic structure for a varying ELT is presented in [5]. Briefly, an ELT has a fast implementation block diagram given by the example in Fig. 2 for $K = 2$. As long as the blocks change with time, but remain orthogonal, the analysis-synthesis system maintains PR and orthogonality [5]. The ELT is a paraunitary filter bank described by the transform matrix whose elements are p_{mn}. In a time-varying form, at instant k, we have

$$p_{mn,k} = h(n,k)\cos\left[(2m+1)(2n+2N-L+3)\frac{\pi}{8}\right] \qquad (2)$$

$h(n,k)$ is a modulating window for instant k generated by a series of plane rotations denoted as Θ in Fig. 2. The angles of these rotations define the window (which is a low-pass prototype), generating the filter bank.

The key idea is to find a bypass state for an ELT, in which input is copied to output without changes. The reason for this is because we would be able to bypass the transform, but we would maintain PR in the transition. This state is achieved when [5] all angles are set to $\pi/2$ and the **Z** matrix in Fig. 2 is adequately chosen. Then

$$\mathbf{Z} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{K+1} \quad \Theta_i = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad (3)$$

Once the decision to change the transform to its bypass state is made, there will be a transition period where neither state is achieved. This transition lasts for K blocks of 2 samples. In the example in Fig. 2, suppose after computing ELT block $k-1$ it is decided to bypass the transform. Plane rotations up to $\Theta_1(k+1)$ and $\Theta_0(k)$ are already used and cannot be changed, without losing the PR property. The matrices available for change lie beyond this point and, following Fig. 2, it is easy to see

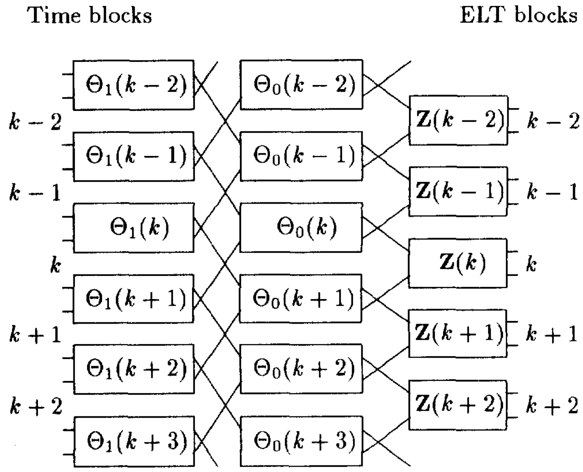

Figure 2. Flow graph for the time-varying ELT with $K = 2$.

that the blocks k and $k+1$ would use both new and old matrices Θ. Therefore they belong to a transition period and block $k+2$ is completely copied to the output. The instantaneous frequency response of the filter bank in the transition region (instants k and $k+1$ in the example) is, accordingly, transitory. Their instantaneous responses do not have good band attenuation, neither they are flat as in the bypass state. A substitution of **Z** by a make-up matrix [5] is also possible to either improve their filtering characteristics or to flatten them.

IV Adapting the tree

As pointed in the two previous sections, the tree can have its paths changed by bypassing terminal filter banks. With this in mind, we can set an activity map associated with the state of the ELT, bypassing it when the node on its left has $a_{ij}(n) = 0$. Thus, the binary tree is completely determined by the set of all a_{ij}. It is impossible to keep track of an infinite number of nodes and the number of nodes in a stage grows exponentially. Therefore, a maximum stage number N may be assumed, setting the maximum attainable frequency resolution . We can simplify the problem if we look at each filter bank in an independent manner and link the resulting $a_{ij}(n)$ to shape the tree. As one branch is pruned, we are exchanging frequency resolution by time resolution. There is no optimal time-frequency representation and for each established cost function one particular shape shall be set. Among several possibilities, we will use as *example* the performance of a signal coder.

Cost Function - If a stationary model is assumed and we intend to minimize the mean square error, the greater energy compaction in fewer coefficients [6] results in less distortion for a given bit-rate. This will generally lead us to chose the full tree which has better frequency resolution [7] . However, as a transform is bypassed, the filter for the resulting subband is shortened, therefore a better spatial localization is attainable. Furthermore, since we are using same filters for analysis and synthesis, as in (1), a distortion in a coefficient in a particular subband would spread along a smaller region than if the filters were longer. It is necessary to specify other variables, such as the type of quantization-coding, before drawing any conclusion, but a non-linear cost function is likely to be best suited for the case. There is not much done in this sense and we do not want to extend too much on this subject. In our concern, we will seek the maximum time resolution whenever there is not much energy compaction provided by the transform. Let $x(m) = x_{ij}(m)$ and $x_L(n) = x_{i+1,2j}(n)$, $x_H(n) = x_{i+1,2j+1}(n)$. For this node, $a(n)$ is the activity signal. Let

$$\sigma_L^2 = E\left[x_L^2(n)\right] \quad \sigma_H^2 = E\left[x_H^2(n)\right] \tag{4}$$

The above variances are related to the variance of $x(n)$ since the ELT has filters (with gain of $\sqrt{2}$, for orthonormality) which obey the power complementary property.

$$\sigma_L^2 + \sigma_H^2 = 2\sigma_x^2 \tag{5}$$

The signal is not assumed stationary and the variances can be estimated continuously. Further computations over $x(n)$ would lead to more complexity and we can work directly with the decomposed signal. Then, a windowed estimation of the variance using a filter with impulse response $h(n)$ could be

$$\hat{\sigma}_L^2(n) = x_L^2(n) * h(n) \quad \hat{\sigma}_H^2(n) = x_H^2(n) * h(n) \tag{6}$$

Using the estimated variances for energy compaction calculations, we have a measure of the transform coding gain [6] as

$$G(n) = \frac{1}{2}\frac{\hat{\sigma}_L^2(n) + \hat{\sigma}_H^2(n)}{\hat{\sigma}_L(n)\hat{\sigma}_H(n)} \tag{7}$$

and we can compare $G(n)$ to a threshold g in order to decide if we set $a(n) = 1$ or not. Hence,

$$a(n) = u[G(n) - g] \tag{8}$$

where $u(x)$ is the step function.

Interrelation of the nodes - To determine all nodes to be made active or not we may use the maximum available frequency resolution, i.e., check nodes in a maximum stage N and, then, their parents. At each node, we may evaluate $a_{ij}(n)$ as in (8).

```
Start: a_ij = 0 (all ij); n ← N.
repeat
      for m = 0...2^n − 1
          if a_nm = 0
             evaluate a_nm
             if a_nm = 1
                make a_ij = 1 for η_ij ∈ path η_00 → η_nm.
      if all a_nm = 1 then stop else n ← n − 1.
until n = 0
```

A simpler way to determine the shape of the tree, but with less resolution, could be the following. Start from any configuration. Evaluate $a_{ij}(n)$ in the above procedure for IVENs and for nodes whose both child nodes are IVENs. On each of the parent nodes of IVENs (clearly they are active nodes) check whether $G(n)$ falls below threshold, in which case the particular node is made inactive. (At the next clock time for that node it becomes an IVEN and those old IVENs are ignored.) To expand a branch, check, on IVENs, whether $G(n)$ surpasses the threshold to turn the node active. As a consequence of this turn, its children nodes become IVENs. This terminal filter bank is solely a part of the adaptation algorithm and its output is not used, because the IVEN signal is the one which is actually quantized and coded. At a coarser spectral resolution it can appear that the spectrum is flat, but it could have some energy compaction at a finer resolution. This means that a node could not meet the specifications to be made active, but it would be better if it was made active in order to allow the activation of a child node, which has unbalanced low and high frequency components. An intermediary remedy could be to check one or two levels of (child) nodes beyond an IVEN.

Transition - As mentioned, there is a transition whenever the transform is bypassed. In a transition, it is preferable to avoid further changes until this transitory state is complete, by forcing the state to remain the same for a certain number of samples. In fact, we should design the filters, for estimating the variances, to be narrow low pass filters, in order to help to prevent constant changes. Although PR is assured, the transitory state is not desirable. Maybe, the best use for this adaptive approach resides on the analysis of signals composed by regions of distinct statistics, i.e., after a change, the statistics remain unchanged for a certain period.

Backward adaptation - In order to prune or expand the same branches in analysis (transmitter) and synthesis (receiver), it is necessary to reconstruct $a_{ij}(n)$ at the receiver. In (7) and (8) we use ELT domain samples. If $G(n)$ in (7) use quantized samples $\hat{x}_L(n), \hat{x}_H(n)$ to estimate variances and if, for a node η_{ij}, we use

$$a_{ij}(n+1) = u[G_{ij}(n) - g_{ij}] \tag{9}$$

the receiver can recover a_{ij} without transmission of side-information, because it has available the past quantized ELT or time-domain samples and g_{ij} can be a fixed threshold. Both receiver and transmitter have to be synchronized, such that the transmitter has to use quantized values also when the transform is bypassed. Furthermore, the same nodal interrelation algorithm has to be used, always preparing future values of $a_{ij}(n+1)$. The filter $h(n)$ has to be causal and a simple first or second order IIR filter can be adequate, having a narrow low-pass band to avoid frequent transitions. Whenever a transition occurs, the states of the IIR filter may be reset, interrupting filtering until the transition is over. After that, filtering is resumed. This is because at a transition, the receiver will not have ELT samples and time-domain ones. Therefore, it will not be able to perform filtering on $\hat{x}_L^2(n), \hat{x}_H^2(n)$ unless the time-domain samples are recovered and transformed again. Setting the filters to avoid frequent changes, can give time enough to the IIR filter to set up again.

Forward adaptation - In case the activity map with all a_{ij} is sent in parallel, things get much easier. First, there is no need to calculate activity on the receiver side. Second, one can use any means to determine activity of the nodes, including non-causal filters. A non-causal $h(n)$ is naturally preferred since it will avoid very short changes. The binary signal $a_{ij}(n)$ can be processed to avoid short bursts and locally oscillatory behaviors. A possible solution is a recursive median filter, which, in the binary case, is easily computed using tables. The formula for it is:

$$a_{ij}(n) = \text{round}\left[\text{mean}\left(a_{ij}(n-k)\ldots a_{ij}(n+k)\right)\right] \quad (10)$$

This would prevent bursts of up to k isolated values of a_{ij} and would not oscillate if the input is an alternation of 0's and 1's. When an oscillation is encountered, the state just before it is preserved. The order in which node activity is evaluated may be found using the algorithm described earlier.

The great disadvantage is due to the transmission of side information, since all the activity map has to be transmitted. Assuming 1 bit per sample and a maximum level number N, it would require $N/2$ bits per sample as overhead. (Remember that if node η_{00} works in a sampling rate f_s, node η_{ij} works in a rate $2^{-i}f_s$.) However, assuming the filters would prevent very frequent changes, run-length coding can be applied to largely compress this map. Furthermore, as a node is active, all nodes connecting it to the root will also be active. Therefore, information for them is not necessary.

V Remarks

What could be imagined as a complex theoretical work became a simple strategy, but associated with a complex and interesting programming work. The independence of nodes make them suitable for a concurrent implementation where each node can be a self contained cell. A manager unit has the job of interacting the nodes, establishing hierarchies and deciding the shape of the tree.

It is worth mentioning the work in [8], where time-varying adaptive wavelet packets are also presented, although using a totally different approach.

Space limitations preclude the inclusion of simulation results using speech signals as well as bit-allocation considerations. These will be presented at the symposium and perhaps published elsewhere as a more complete paper, where all those topics outlined in Sec. IV could be explained in more detail.

References

[1] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, pp 14–38, vol. 8, Oct. 1991.

[2] A.K. Soman and P.P. Vaidyanathan, "Paraunitary filter banks and wavelet packets," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, vol. IV, pp. 397–400, 1992.

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

[4] H. S. Malvar, "Fast computation of wavelet transforms with the extended lapped transform," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, vol. IV, pp. 393–396, 1992.

[5] R.L. de Queiroz and K. R. Rao, "Adaptive extended lapped transforms," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Minneapolis, MN, Apr. 1993.

[6] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] R. L. de Queiroz and H. S. Malvar, "On the asymptotic performance of hierarchical transforms," *IEEE Trans. on Signal Processing*, Vol. 40, pp 2620–2622, Oct. 1992.

[8] C. Herley, J. Kovacevic, K. Ranchandran and M. Vetterli, "Arbitrary orthogonal tilings of the time-frequency plane." *Proc. of Intl. Symp. on Time-Frequency and Time-Scale Analysis,* Victoria, Canada, Oct. 1992.