TECHNICAL RESEARCH REPORT

Fast Reconstruction of Subband-Decomposed Progressively- Transmitted, Signals

by H. Jafarkhani, N. Farvardin

T.R. 97-18

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Fast Reconstruction of Subband-Decomposed Progressively-Transmitted Signals *

Hamid Jafarkhani
Nariman Farvardin
Department of Electrical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742

ABSTRACT

In this paper we propose a fast reconstruction method for a progressive subband-decomposed signal coding system. It is shown that unlike the normal approach which contains a fixed computational complexity, the computational complexity of the proposed approach is proportional to the number of refined coefficients. Therefore, using the proposed approach in image coding applications, we can update the image after receiving each new coefficient and create a continuously refined perception. This can be done without any extra computational cost compared to the normal case where the image is reconstructed after receiving a predefined number of bits.

1.INTRODUCTION

Progressive transmission of signals is a mechanism by which the encoder's output is transmitted in groups of bits (packets) and the decoder produces a higher quality replica of the signal based on receiving each new packet. Progressive transmission is a desirable feature in many practical signal transmission situations such as telebrowsing and database retrieval. Progressive transmission also provides the opportunity for interrupting transmission when the quality of the received signal has reached an acceptable level or when the receiver decides that the received signal is of no interest. Likewise, in applications where the receiver is more interested in specific parts of the signal rather than the entire signal (e.g., content-based image transmission), a valid question is how to transmit and reconstruct a signal with different levels of quality (distortion) in different temporal or spatial regions. For example, after receiving a rough reproduction of a medical image in a telemedicine application, the radiologist, at the receiver, may want to highlight a part of the image and request a higher fidelity (or even lossless) replica of only the highlighted area.

The use of subband decomposition or discrete wavelet transform (DWT) in image coding systems has received much attention in recent years [1]-[3]. Not only do these coding techniques provide good compression results, but also they are inherently multi-resolution. Recently, several rate-scalable subband image coding systems have been proposed in the literature which provide very good performances [4, 5, 6, 7].

One of the problems in a wavelet-based progressive transmission scheme is that the decoder, upon receiving new bits, has to perform the inverse filtering operation to reconstruct the image. The normal approach for doing this is to apply the inverse filters on the decoded versions of all wavelet coefficients to reconstruct a replica of the image. In this approach, even if one wavelet coefficient is refined, the reconstruction complexity will be the same as in the case when all coefficients are refined.

In a progressive transmission scenario however, at each step of progression, only some of the wavelet coefficients are refined and the other coefficients remain unchanged. Therefore, only a portion of the image pixels will change after receiving the new bits. To reduce the complexity of reconstruction, one can recompute only those pixels of the image that need to be changed. Not only does our proposed scheme provide a fast reconstruction of the output, but also it can update the output as each wavelet coefficient arrives and does not need to wait for receiving all packets before it starts the process of reconstruction. This can be done without any increase in complexity. This feature of our reconstruction scheme is not limited to progressive transmission systems and can be used for any packetized bit stream for on-line reconstruction of the output.

*Prepared in part through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002 and in part through support received from the National Institute of Standards and Technology grant 70NANBSH0051.

In Section 2, we propose a new way of reconstructing the signal which allows for updating only the necessary portions of the signal, instead of reconstructing the whole thing. The complexity of this new reconstruction approach is proportional to the number of refined coefficients. Section 3 generalizes the approach of Section 2 to a general filter bank structure for subband reconstruction and Section 4 contains concluding remarks.

2.ONE LEVEL OF DECOMPOSITION

In signal coding based on the DWT, the input signal is typically decomposed into two components: (i) a *low-resolution approximation* and (ii) a *detail signal*. This results in decomposing the input signal into low-pass and high-pass versions, generally referred to as subbands. Each of the resulting subbands can be further decomposed using the same approach. In this manner, the DWT decomposes a given input signal into a number of frequency bands [2]. At the receiver, the signal can be reconstructed using appropriate inverse DWT filters. Fig. 1 shows one level of two-band decomposition and reconstruction using linear-phase, finite-impulse response filter banks [3].

In this section, to convey the basic idea behind the proposed fast reconstruction approach, we limit ourselves to a one-level decomposition as in Fig. 1 where the reconstructed output signal can be written as:

$$\begin{aligned} y(n) = &\textstyle\sum_{m=-N_h}^{N_h} x'_h(n-m)h(m)+ \\ &\textstyle\sum_{m=-N_g}^{N_g} x'_g(n-m)g(m), \end{aligned} \tag{1}$$

where

$$x'_h(n) = \begin{cases} x_h(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \ X'_h(z) = X_h(z^2),$$
$$x'_g(n) = \begin{cases} x_g(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \ X'_g(z) = X_g(z^2), \tag{2}$$

and $2N_h + 1$ and $2N_g + 1$ are the lengths of the h and g filters, respectively.

To simplify the notation, we define $x_h(a) = x_g(a) = 0$ if $a \notin \mathbb{Z}$. So,

$$x'_i(n) = x_i(\frac{n}{2}), \ i = h, g. \tag{3}$$

Using the above notation, the output can be written as:

$$\begin{aligned} y(n) = &\textstyle\sum_{m=-N_h}^{N_h} x_h(\frac{n-m}{2})h(m)+ \\ &\textstyle\sum_{m=-N_g}^{N_g} x_g(\frac{n-m}{2})g(m). \end{aligned} \tag{4}$$

Now, let us consider a progressive transmission scheme where $y_l(n)$, $l = 1, 2, \cdots, K$, represents the different refinements of the output. The refinement of the output can be achieved by refining the input coefficients successively using $b_l^i(n)$ bits/sample to quantize $x_i(n)$ ($i = h, g$) at level l. The values of $b_l^i(n)$ are defined by a bit allocation algorithm or are specified inherently in the coding system. Note that $b_l^i(n) \leq b_{l+1}^i(n)$ and it is possible – and crucial to note – that we might have $b_l^i(n) = b_{l+1}^i(n)$, for some i. We use $x_i(n, b_1^i), x_i(n, b_2^i), \cdots, x_i(n, b_K^i)$ to represent the decoded versions of the subband coefficients $x_i(n)$, at different levels of refinement ($x_i(n, b_1^i)$ is the coarsest representation of $x_i(n)$ using $b_1^i(n)$ bits for quantization and $x_i(n, b_K^i)$ is the finest representation using $b_K^i(n)$ bits for quantization).

Using the normal method of reconstruction in subband coding systems, for each $l = 1, 2, \cdots, K$, upon obtaining x_h and x_g, the decoder reconstructs the l^{th} refinement of the output, y_l, according to:

$$\begin{aligned} y_l(n) = &\textstyle\sum_{m=-N_h}^{N_h} x_h(\frac{n-m}{2}, b_l^h)h(m)+ \\ &\textstyle\sum_{m=-N_g}^{N_g} x_g(\frac{n-m}{2}, b_l^g)g(m). \end{aligned} \tag{5}$$

In other words, given that y_l is reconstructed at the decoder, upon receiving additional bits from which $x_h(n, b_{l+1}^h)$ and $x_g(n, b_{l+1}^g)$ – the $(l+1)$st refinement of x_h and x_g – are obtained, the inverse DWT filters are used *anew* to compute an updated replica of the signal – the $(l+1)$st refinement of y.

In this paper, we propose an alternative method in which we define $\Delta y_l(n) = y_{l+1}(n) - y_l(n)$ and $\Delta x_i(n, b_l^i) = x_i(n, b_{l+1}^i) - x_i(n, b_l^i)$, $i = h, g$. Clearly, Δy_l, $l = 1, 2, \cdots, K$, can be computed as follows:

$$\begin{aligned} \Delta y_l(n) = &\textstyle\sum_{m=-N_h}^{N_h} \Delta x_h(\frac{n-m}{2}, b_l^h)h(m)+ \\ &\textstyle\sum_{m=-N_g}^{N_g} \Delta x_g(\frac{n-m}{2}, b_l^g)g(m). \end{aligned} \tag{6}$$

Then, to reconstruct a new refinement of the signal, y_{l+1}, it suffices to use

$$y_{l+1}(n) = y_l(n) + \Delta y_l(n). \tag{7}$$

Now let us consider the complexity implication of this new approach and its comparison with the normal approach using (5). If the number of samples of the input signal (x) is N, then the number of samples in x_h and x_g will be $\frac{N}{2}$. If, at each level of refinement l, we use (5) to compute y_l, it costs $L = \frac{N}{2}(2N_h + 1) + \frac{N}{2}(2N_g + 1)$ multiplications, a quantity which is independent of the number of refined samples. Therefore, after K levels of refinement the computational complexity is KL multiplications. Using (6) and (7) to update the output signal costs $2N_i+1$ multiplications per each nonzero Δx_i coefficient, $i = h, g$ (number of nonzero $\Delta x_i \leq \frac{N}{2}$). If $\alpha_{il}\frac{N}{2}$ wavelet coefficients (x_i, $i = h, g$) are refined at the l^{th} level of refinement, then the total number of multiplications using (6) and (7) to update the output signal is $\sum_{l=1}^{K} \alpha_{hl}\frac{N}{2}(2N_h+1) + \sum_{l=1}^{K} \alpha_{gl}\frac{N}{2}(2N_g+1)$. Note that for large K's, $\sum_{l=1}^{K} \alpha_{il} << K$ for existing rate-scalable image coding systems [4, 5, 6, 7] and therefore the complexity of the proposed approach is much less than the complexity of the normal approach. For example, in the embedded zerotrees wavelet (EZW) coding approach of [4], not only are there many insignificant coefficients at low bit rates which are not transmitted and are assumed to be zero, but also the significant coefficients are refined in dominant and subordinate passes (at each pass some of the "significant" coefficients remain unchanged).

In addition to reducing the complexity of reconstruction, using (6) and (7) to reconstruct the output provides the capability to update the output signal upon receiving the new refinement of each wavelet coefficient, thus allowing for an *on-line* update of the output. Typically, for any signal coding system the output bit stream is transmitted in the form of packets. Consider a general signal coding system (or one level of refinement of a progressive system) and assume that the output is transmitted using M packets. The size of each packet and thus the value of M depends on the network characteristics. At the receiver, either we should wait to receive all M packets or we can reconstruct M intermediate versions of the output – one for each newly received packet. Using the normal reconstruction formula (5), results in a delay in the former case and an increase in computational complexity by a factor of M in the latter case. Using (6) and (7) to reconstruct the output will provide M intermediate versions of the output without increasing the complexity or delay. Also, note that if after receiving a rough reproduction of the signal, we just need to refine a portion of wavelet coefficients (corresponding to a spatial location), the computational cost using (6) and (7) is proportional to the number of transmitted coefficients (filter inputs) and less than the computational cost of the normal scheme.

To update the output, we need to know the set of output samples whose values are influenced by the refinement of each coefficient. This set depends on the location of the refined coefficient. Also, the samples near the boundaries need a special treatment because of the symmetric extension of the input signal[1]. Receiving $\Delta x_h(k)$ for the coefficients sufficiently far from the boundaries, $\frac{N_h}{2} < k < \frac{N-N_h}{2}$, affects only the output values $y(2k+n)$, $n = -N_h, -(N_h-1), \cdots, N_h$. When the received coefficient is near the origin, $0 \leq k \leq \frac{N_h}{2}$, the output samples $y(n)$, $n = 0, 1, \cdots, N_h + 2k$, are affected. For the remaining coefficients, $\frac{N-N_h}{2} \leq k < \frac{N}{2}$, the output values $y(2k+n)$, $n = -N_h, \cdots, N-2k$, are affected. Replacing x_h with x_g and N_h with N_g provides the same indices for the filter g and coefficients x_g.

The same procedure can be applied to more than one-level of decomposed signals. The two-dimensional (2-D) extension of the approach is also straightforward when we use a separable 2-D transform.

3.GENERAL FILTER BANK

In this section, we extend the idea for complexity reduction developed in the previous section to general filter banks. Fig. 2 illustrates a general decomposition and reconstruction structure using a set of passband filters called *filter banks.* This structure can be used to perform linear transformations like the discrete cosine transform, lapped orthogonal transform, Laplacian pyramid, Gabor transform, quadrature mirror filters, and DWT [8]. Although there exist "fast implementations" for many of these transformations, the study of this general structure is worthy of consideration as it unifies the application of our approach for different transformations and provides a general methodology for progressive transmission situations. Furthermore, our approach does not depend on the specific transformation used. In fact, the ratio of the computational complexity of the proposed approach to that of the normal approach for reconstruction is fixed for any transformation. So, the choice of the transformation affects the complexity of both approaches in the same manner.

To make the argument more precise, first, let us consider the example illustrated in Fig. 3 consisting of three levels of the two-band structure in Fig. 1. Later, we generalize our fast reconstruction approach to the filter bank shown in Fig. 2. The example in Fig. 3 is equivalent to Fig. 2 when $k_1 = 2$, $k_2 = 4$, $k_3 = k_4 = 8$ and filters g_i and h_i, $i = 1, 2, 3, 4$, are chosen appropriately. Given filters h and g in Fig. 3, the corresponding reconstruction filters in Fig. 2 can be found. The main equation governing the relationship between g_i, $i = 1, 2, 3, 4$ and g and h is the equivalence of the two structures in

[1] To have a smooth boundary after reconstruction, the signal is symmetrically extended near the boundaries.

Fig. 4 [8]. Using Fig. 4, one can establish the following equations relating different filters in Figs. 2 and 3:

$$G_1(z) = G(z), \tag{8}$$
$$G_2(z) = G(z^2)H(z), \tag{9}$$
$$G_3(z) = G(z^4)H(z^2)H(z), \tag{10}$$

and

$$G_4(z) = H(z^4)H(z^2)H(z). \tag{11}$$

Although the reconstruction operation in Figs. 2 and 3 are the same, the number of multiplications in Fig. 2 is more than the number of multiplications in Fig. 3. Specifically the output signal in Fig. 2 can be written as:

$$\begin{aligned} y(n) = & \sum_{m=-N_1}^{N_1} x_1'(n-m)g_1(m) + \\ & \sum_{m=-N_2}^{N_2} x_2'(n-m)g_2(m) + \\ & \sum_{m=-N_3}^{N_3} x_3'(n-m)g_3(m) + \\ & \sum_{m=-N_4}^{N_4} x_4'(n-m)g_4(m), \end{aligned} \tag{12}$$

where

$$x_1'(n) = \begin{cases} x_1(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \ X_1'(z) = X_1(z^2),$$
$$x_2'(n) = \begin{cases} x_2(\frac{n}{4}) & n = 4k \\ 0 & \text{otherwise} \end{cases}, \ X_2'(z) = X_2(z^4),$$
$$x_3'(n) = \begin{cases} x_3(\frac{n}{8}) & n = 8k \\ 0 & \text{otherwise} \end{cases}, \ X_3'(z) = X_3(z^8),$$
$$x_4'(n) = \begin{cases} x_4(\frac{n}{8}) & n = 8k \\ 0 & \text{otherwise} \end{cases}, \ X_4'(z) = X_4(z^8), \tag{13}$$

and $2N_i + 1$ is the length of g_i and $N_1 = N_g$, $N_2 = 2N_g + N_h$, $N_3 = 4N_g + 3N_h$, and $N_4 = 7N_h$. So, using all coefficients to reconstruct the output, we need $L_1 = N[\frac{2N_1+1}{2} + \frac{2N_2+1}{4} + \frac{2N_3+1}{8} + \frac{2N_4+1}{8}]$ multiplications in Fig. 2 and $L_2 = N[(2N_h+1)(\frac{1}{8}+\frac{1}{4}+\frac{1}{2}) + (2N_g+1)(\frac{1}{8}+\frac{1}{4}+\frac{1}{2})]$ multiplications in Fig. 3. After computing L_1 in terms of N_h and N_g, we can show that

$$L_1 = N[1+3N_g+3N_h] > L_2 = N[\frac{7}{4}(N_h+N_g+1)], \tag{14}$$

establishing that the number of multiplications corresponding to the implementation of Fig. 3 is less than that of Fig. 2.

Now, we concentrate on the main issue of this section and show that an approach similar to what was proposed in Section 2 can be used for the filter bank shown in Fig. 2. To simplify the notation, we define $x_i(a) = 0$ if $a \notin \mathbb{Z}$. So,

$$x_i'(n) = x_i(\frac{n}{2^i}), \ i = 1,2,3, \text{ and } x_4'(n) = x_4(\frac{n}{8}). \tag{15}$$

Using the above notation, the output can be written as:

$$\begin{aligned} y(n) = & \sum_{m=-N_1}^{N_1} x_1(\frac{n-m}{2})g_1(m) + \\ & \sum_{m=-N_2}^{N_2} x_2(\frac{n-m}{4})g_2(m) + \\ & \sum_{m=-N_3}^{N_3} x_3(\frac{n-m}{8})g_3(m) + \\ & \sum_{m=-N_4}^{N_4} x_4(\frac{n-m}{8})g_4(m). \end{aligned} \tag{16}$$

In the normal approach for reconstructing the output of a progressive transmission scheme, the different levels of refinement of the output, $y_1, y_2, \cdots, y_K$, are computed after receiving $x_i(n, b_1^i), x_i(n, b_2^i), \cdots, x_i(n, b_K^i)$ as follows:

$$\begin{aligned} y_l(n) = & \sum_{m=-N_1}^{N_1} x_1(\frac{n-m}{2}, b_l^1)g_1(m) + \\ & \sum_{m=-N_2}^{N_2} x_2(\frac{n-m}{4}, b_l^2)g_2(m) + \\ & \sum_{m=-N_3}^{N_3} x_3(\frac{n-m}{8}, b_l^3)g_3(m) + \\ & \sum_{m=-N_4}^{N_4} x_4(\frac{n-m}{8}, b_l^4)g_4(m). \end{aligned} \tag{17}$$

As before, we define $\Delta y_l(n) = y_{l+1}(n) - y_l(n)$ and $\Delta x_i(n, b_l^i) = x_i(n, b_{l+1}^i) - x_i(n, b_l^i)$, where Δy_l, $l = 1, 2, \cdots, K$, is given by:

$$\begin{aligned} \Delta y_l(n) = & \sum_{m=-N_1}^{N_1} \Delta x_1(\frac{n-m}{2}, b_l^1)g_1(m) + \\ & \sum_{m=-N_2}^{N_2} \Delta x_2(\frac{n-m}{4}, b_l^2)g_2(m) + \\ & \sum_{m=-N_3}^{N_3} \Delta x_3(\frac{n-m}{8}, b_l^3)g_3(m) + \\ & \sum_{m=-N_4}^{N_4} \Delta x_4(\frac{n-m}{8}, b_l^4)g_4(m). \end{aligned} \tag{18}$$

Then, given that $y_l(n)$ is available at the decoder, $y_{l+1}(n)$ can be computed by adding $\Delta y_l(n)$ to $y_l(n)$. Equation (18) is a generalization of (6) and therefore provides a fast reconstruction scheme as was discussed in Section 2.

To update the output, we need to know the list of output samples which must be updated due to receiving each coefficient. Table 1 shows the required updates for the coefficients that are sufficiently far from the boundaries. The first row contains the filter coefficients which are used to update the corresponding outputs in the second row. To make the details of the reconstruction scheme more clear, let us provide a numerical example.

3.1.EXAMPLE

Although our scheme works for any scalable transform coding system, we pick a particular coding system to show the practical usefulness of our scheme. We use the EZW approach of [4] for this purpose. In [4], an 8×8 wavelet decomposed image has been used to present the details of the EZW algorithm. We use the same example (Table 2) and assume that refining each coefficient affects all output pixels in the same row or column (a logical assumption on the length of filters for an 8×8 image). The normal reconstruction approach takes 1024 multiplications to create each refinement of the output. Without going through the details of the EZW, as it is shown in [4], the first dominant pass results in refining coefficients in locations 11, 12, 13, and 54 (the numbers indicate the row and column indices, respectively). Reconstructing the image using our approach takes 160 multiplications instead of 1024. The first subordinate pass (the second level of refinement) refines the same four coefficients [4] and thus results in the same amount of computational reduction. The second dominant pass (the third level of refinement) changes coefficients in locations 21 and 22. Our approach requires 80 multiplications to reconstruct the third image. The second subordinate pass (the fourth level of refinement) refines the aforementioned six wavelet coefficients. To create the fourth image, our approach requires 240 multiplications. Therefore, to reconstruct the first four images, the fast scheme needs a total of 640 multiplications compared to 4096 multiplications. Also, note that since we have had 16 coefficient refinements in the first four passes, we could have reconstructed 16 intermediate images without any additional computational complexity. Constructing these 16 images would cost 16384 multiplications using normal wavelet reconstruction formulas.

4.CONCLUSIONS

We have proposed a scheme for fast reconstruction of a subband-decomposed progressively transmitted signal. Using the proposed approach, we can update the image after receiving each new coefficient and create a continuously refined perception without any extra computational cost (compared to the case that we reconstruct the image after receiving each packet). In existing scalable image coding systems [4]-[7], insignificant coefficients remain zero at the decoder for a few steps of refinement. Also, at each step of the refinement, some of the coefficients remain unchanged. Unrefined coefficients do not add to the computational complexity of our new approach for reconstructing the image. On the other hand, the complexity of the normal reconstruction scheme does not depend on the value of the coefficients (or refinements).

In some applications, after receiving a preliminary draft of the image, the receiver only needs the upgrade of a particular portion of the image. Using our proposed approach, the complexity reduction in this case is equal to the ratio of the number of pixels in the required area to the number of pixels in the whole image.

Future work includes the generalization of the scheme for filter banks with integer coefficients which replace multiplications by binary shifts. [2]

References

[1] J.W. Woods, and S.D. O'Neil, "Subband Coding of Images," *IEEE Trans. Acoust., Speech and Signal Proc.*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.

[2] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition : The Wavelet Representation," *IEEE Trans. Pattern Anal. and Mach. Intel.*, vol. 11, pp. 674–693, Jul. 1989.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. Image Proc.*, vol. IP-1, pp. 205–220, Apr. 1992.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Proc.*, vol. 41, pp. 3445-3462, December 1993.

[2] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Goverment.

[5] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," submitted to *IEEE Trans. Circ. & Syst. Video Tech.*.

[6] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. on Image Proc.*, vol. IP-3, pp. 572-588, September 1994.

[7] H. Jafarkhani and N. Farvardin, "A Scalable Wavelet Image Coding Scheme Using Multi-Stage Pruned Tree-Structured Vector Quantization," *IEEE Int. Conf. on Image Proc.*, Washington, DC, Oct. 1995.

[8] M. Vitterli, "Multirate Filter Banks for Subband Coding," in *Subband Image Coding*, J. W. Woods, Ed., *Kluwer Academic Publishers*, Norwell, MA, pp. 43–100, 1991.

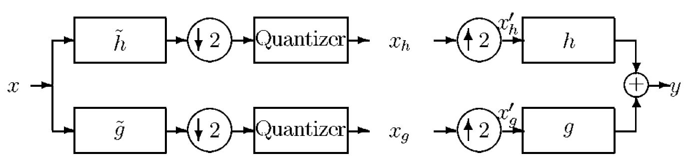

Figure 1: Two-Band Subband Analysis and Synthesis Structure.

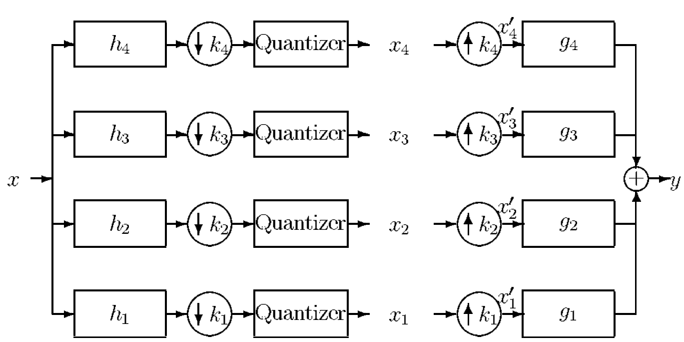

Figure 2: Subband Analysis and Synthesis Filter Banks.

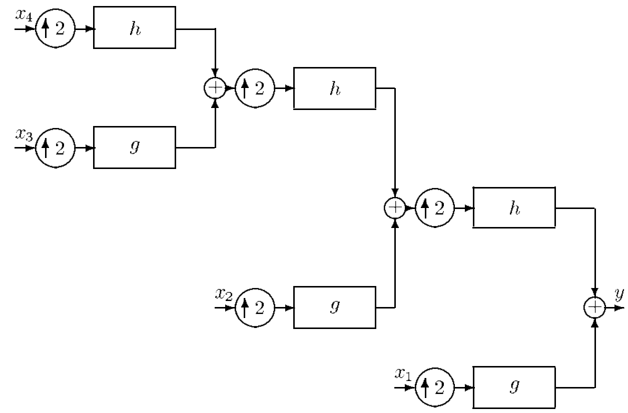

Figure 3: Wavelet Synthesis Filter Bank.

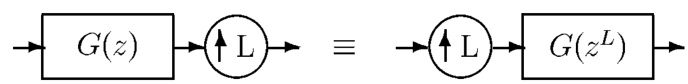

Figure 4: Two Equivalent Structures.

$x_1(k)$	$g_1(-N_1)$	$\cdots$	$g_1(0)$	$\cdots$	$g_1(N_1)$
$\frac{N_1}{2} < k < \frac{N-N_1}{2}$	$y(2k-N_1)$	$\cdots$	$y(2k)$	$\cdots$	$y(2k+N_1)$
$x_2(k)$	$g_2(-N_2)$	$\cdots$	$g_2(0)$	$\cdots$	$g_2(N_2)$
$\frac{N_2}{4} < k < \frac{N-N_2}{4}$	$y(4k-N_2)$	$\cdots$	$y(4k)$	$\cdots$	$y(4k+N_2)$
$x_3(k)$	$g_3(-N_3)$	$\cdots$	$g_3(0)$	$\cdots$	$g_3(N_3)$
$\frac{N_3}{8} < k < \frac{N-N_3}{8}$	$y(8k-N_3)$	$\cdots$	$y(8k)$	$\cdots$	$y(8k+N_3)$
$x_4(k)$	$g_4(-N_4)$	$\cdots$	$g_4(0)$	$\cdots$	$g_4(N_4)$
$\frac{N_4}{8} < k < \frac{N-N_4}{8}$	$y(8k-N_4)$	$\cdots$	$y(8k)$	$\cdots$	$y(8k+N_4)$

Table 1: Update Table for Coefficients Sufficiently Far from the Boundaries.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Table 2: Example of 3-level wavelet transform of an image [4].