Computer Engineering Laboratory

Mekelweg 4, 2628 CD

Delft, The Netherlands

Web: http://ce.et.tudelft.nl

Scalable Video Coding: A Technical Report

Roya Choupani, Stephan Wong, and Mehmet Tolun

Abstract

Video streaming over the Internet has gained popularity during the recent years which is mainly the result of the introduction of videoconferencing and videotelephony. These in turn have made it possible to bring to life many applications such as transmitting video over the Internet and over telephone lines, surveillance and monitoring, telemedicine (medical consultation and diagnosis at a distance), and computer based training and education. The heterogeneous, dynamic and best-effort structure of the Internet however, can not guarantee any specific bandwidth for a connection. Many video coding standards have tried to deal with this problem by introducing the scalability feature as adapting video streams to the fluctuations in the available bandwidths. In this study, we have discussed the main technical features of more common scalable video coding techniques. The main problems of these methods and their applicability together with the available motion compensated video coding methods are discussed as well.

CONTENTS

I. INTRODUCTION

With the steady increase in the Internet access bandwidth, more and more applications start to use the streaming audio and video contents [1], [2]. In response to the increasing demand on streaming video applications over the best-effort Internet, the coding objective for streaming video has changed to optimize the video quality for a wide range of bit rates. In the streaming video applications, the servers normally have to serve a large amount of users with different screen resolutions and network bandwidth. When the users screen resolution is too small and also when the bandwidth between some users and the server is too narrow to support higher resolution sequences, the spatial scalability coding is needed to provide different resolutions. This in turn helps the server to accommodate different users with different bit rate or screen resolution capabilities. Several spatially scalable coding schemes have been proposed and accepted by some major video coding standards, such as H.263 [6], MPEG-2 [4], [5], and MPEG-4 [3]. In all of these schemes the aim is organizing frame data in a way that a user with low bandwidth or display resolution may receive it partially and ignore the remaining and hence have a low quality video. This concept is given by Fine Granularity Scalability (FGS) which is discussed in the following sections. Fine granularity scheme divides the video data into a base layer and one or more enhancement layers. The user receiving the base layer will have the lowest quality video but lowest bit rate at the same time. This enables a simple and flexible solution for transmission over heterogeneous networks, additionally providing adaptability for bandwidth variations and error conditions. Both multicast and unicast streaming applications are possible with minimal processing at server/network and low decoding complexity. It further allows simple adaptation for a variety of storage devices and terminals. For highest flexibility, scalability that provides a fine granularity at the bit stream level is desirable. The most important scalable features to

be considered are different spatial, temporal, and bits per pixel. For video coding, a lack of efficiency can generally be observed in combining scalable coding with the popular approach of hybrid motion-compensated prediction and block transform encoding, as implemented in most of today's standards. This is mainly caused by the recursive structure of the prediction loop, which causes a drift problem whenever incomplete information is decoded and has led to a situation where a wide acceptance of prospective applications has never occurred [7]. In the following sections basic concepts of scalability are defined first. The fine granularity and its implementation is given next. Discrete Wavelet Transform (DWT) based scalability is presented as a replacing method which can provide a continuous scalability rate for the video stream. Some problems and their solutions in implementing FGS and DWT based schemes are also discussed.

II. Fine Granularity Scalability

To adapt the data size to the changes in the bit rate of the connecting network, a unit in a video stream such as a frame or a macro-block, is divided into small items. A measure of the number of items comprising a unit is called its Granularity [18]. The first item of each unit contains the basic and coarsest part of the data and the remaining items contain refinements to the base item [19], [20]. The scheme of gradual refining/increasing the granularity of a unit is called Fine Granularity Scalability (FGS) [20], [22]. It is clear from the above definitions that a gradual increase in the frame size, bit rate or frame rate is achieved through adapting the granularity of a stream to the bit rate capability of the connecting network. Fine granularity scalability scheme defines the video content in a multi layers format [24], [23]. A higher quality for a video is achieved through increasing the number of layers decoded at the receiver side. Details of multi layer scalability and its encoding are given in the following sections.

A. Multi Layer Scalability

The objective of video coding for the Internet streaming has changed to optimizing the video quality over a given bit rate range instead of a single bit rate. With the heterogeneity available in the current networks specially the Internet, dynamic changes in the bit rate due to traffic load or delay in client side before processing, determining a specific bit rate for a video stream is impossible. The encoder should either choose the minimum possible bit rate which guarantees delivery without delay or choose an encoding scheme which can adapt with the fluctuations in the bit rate range. This means that it should be possible to partially decode the video stream at any bit rate within the bit rate range to reconstruct a video signal with the optimized quality at that bit rate. One of the solutions to this problem is encoding the video in several layers. The first layer contains the minimum data required. All remaining layers include refinements to the data carried by the base layer. This makes scalability possible as a receiver can receive some of these layers and ignore the rest depending on its current bit rate capacity. Generally however, only two layers called base and enhancement layers are used. Multi layer scalability is applicable to Bit Rate, Frame Rate and Spatial Scalable Coding Techniques [16], [17].

A.1 Bit Rate Scalability

Bit-rate or signal-to-noise ratio (SNR) scalability is a technique to code a video sequence into two layers at the same frame rate and the same spatial resolution, but different quantization accuracy. Figure 1 shows the SNR scalability decoder defined in MPEG-2 video-coding standard [28], [29]. The base-layer bit stream is decoded by the base layer variable-length decoder (VLD) first. The inverse quantizer in the base layer produces the reconstructed DCT coefficients. The enhancement bit stream is decoded by the VLD in the enhancement layer and the enhancement residues of the DCT coefficients are produced by the inverse quan-
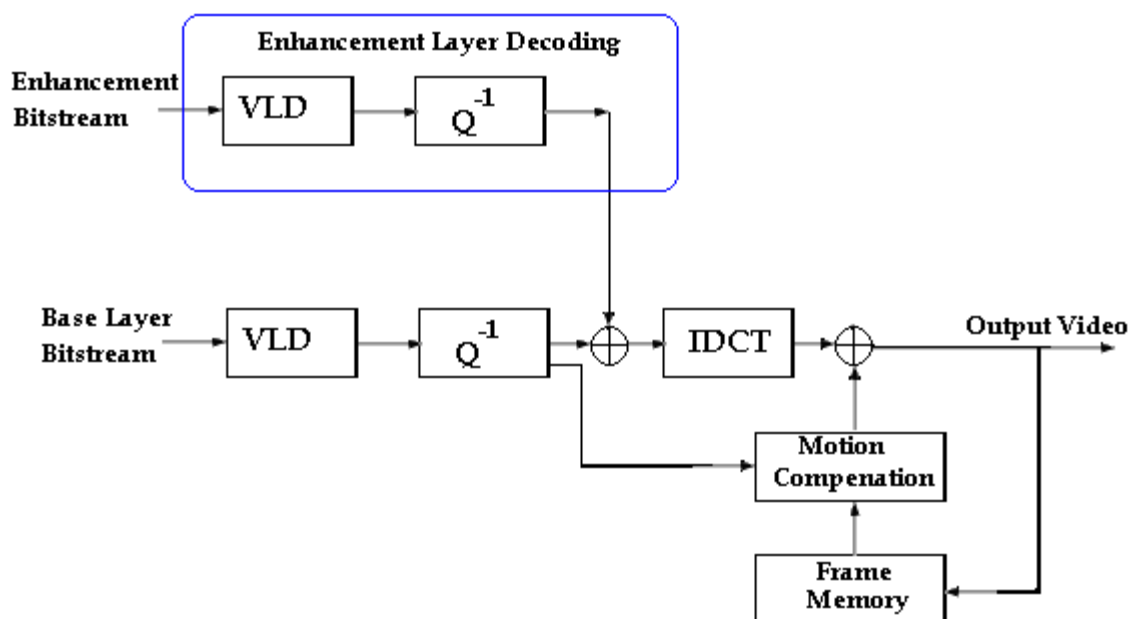
Fig. 1. SNR scalability decoder

tizer in the enhancement layer. A higher accuracy DCT coefficient is obtained by adding the base-layer reconstructed DCT coefficient and the enhancement-layer DCT residue. The DCT coefficients with a higher accuracy are given to the inverse DCT (IDCT) unit to produce reconstructed image domain blocks. In case of P frames IDCT will produce the residues that are added to the motion-compensated block from the previous frame. In this SNR scalability decoder, the enhancement-layer information is used in the motion-prediction loop [30]. Therefore, there are the following four possible results depending on the corresponding SNR scalability encoder and whether the enhancement layer information is received by the decoder or not.

1. If the encoder uses the enhancement-layer information in the motion-prediction loop and the enhancement-layer information is received by the decoder, the enhancement layer coding efficiency is high.

2. If the encoder uses the enhancement-layer information in the motion-prediction loop and the enhancement-layer information is not received by the decoder, drift happens in the base layer and coding efficiency is low.

3. If the encoder does not use the enhancement-layer information in the motion-prediction loop and the enhancement-layer information is received by the decoder, drift happens in the enhancement layer and coding efficiency is low.

4. If the encoder does not use the enhancement-layer information in the motion-prediction loop and the enhancement-layer information is not received by the decoder, the result is the same as using the base layer only.

Therefore, either the base layer has a poor performance to ensure a good performance for the enhancement layer, or the enhancement layer has a poor performance to ensure a good performance for the base layer.

A.2 Temporal Scalability

Temporal scalability is a technique to code a video sequence into two layers at the same spatial resolution, but different frame rates [25], [26]. The base layer is coded at a lower frame rate. The enhancement layer provides the missing frames to form a video with a higher frame rate. Coding efficiency of temporal scalability is high and very close to non scalable coding [26], [?]. Figure 2 shows the structure of temporal scalability. Only P-type prediction is used in the base layer. The enhancement-layer prediction can be either P-type or B-type from the base layer or P-type from the enhancement layer. Motion compensation in the based layer uses only base layer information so no drift problem is expected here. However, as the time interval between consecutive frames in the base layer is increased, a slight decrease in the efficiency of compression step is expected.
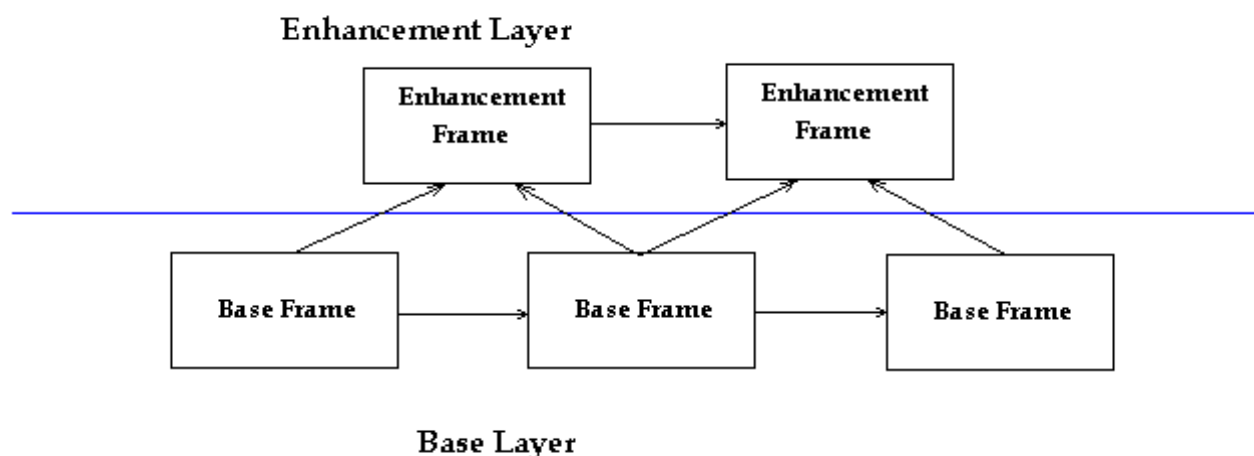
Fig. 2. Typical structure of a temporal scalability decoder

A.3 Spatial Scalability

Spatial scalability is a technique to code a video sequence into two layers at the same frame rate, but different spatial resolutions. The base layer is coded at a lower spatial resolution. The reconstructed base-layer picture is up-sampled to form the prediction for the high-resolution picture in the enhancement layer [16], [17]. Figure 3 shows a single-loop spatial scalability decoder. The advantage of single-loop spatial scalability is its simplicity. If the spatial resolution of the base layer is the same as that of the enhancement layer, i.e., the up-sampling factor being 1, this spatial scalability decoder can be considered as an SNR scalability decoder too. Unlike the SNR scalability decoder in MPEG-2, the above spatial scalability decoder does not include the enhancement-layer information into the prediction loop. Therefore, if the corresponding encoder does not include the enhancement layer information into the prediction loop either, the base-layer drift does not exist. Coding efficiency of the enhanced video using such an "open-loop" scalable coding method suffers from the fact that the enhancement information of the previous frame is not used in the prediction
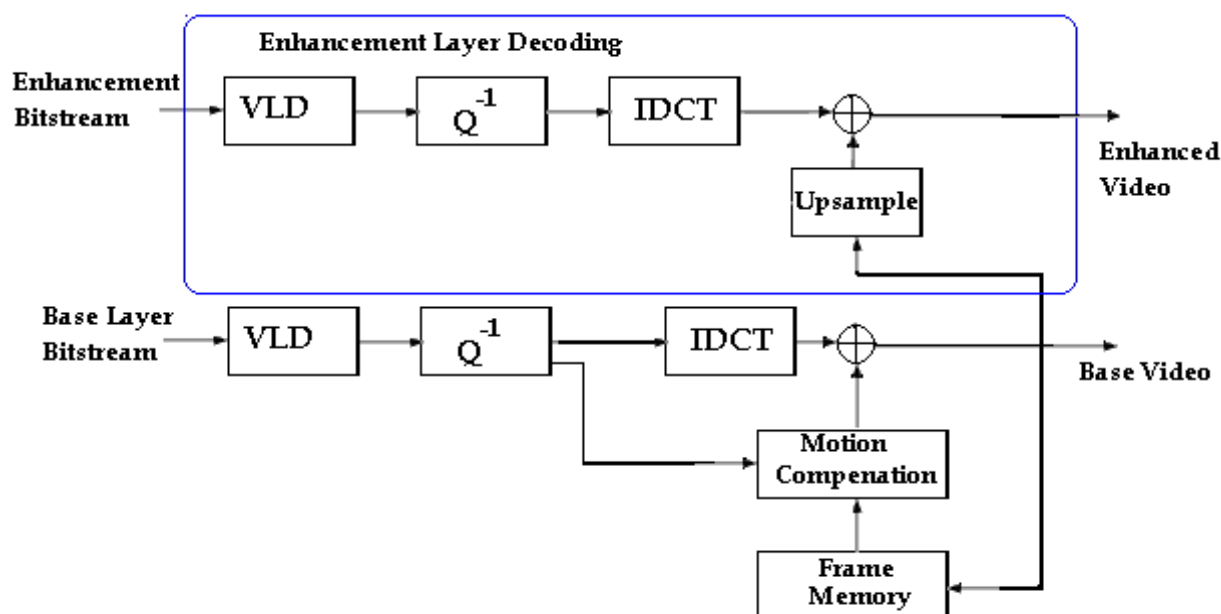
Fig. 3. Typical single-loop spatial scalability decoder

for the current frame. The spatial scalability decoders defined in MPEG-2 and MPEG-4 use two prediction loops, one in the base layer and the other in the enhancement layer [2]-[6]. The MPEG-2 spatial scalable decoder uses as prediction a weighted combination of up-sampled reconstructed frames from the base layer and the previously reconstructed frame in the enhancement layer, while the MPEG-4 spatial scalable decoder allows a "bi-directional" prediction using up-sampled reconstructed frame from the base layer as the "backward reference" and the previously reconstructed frame in the enhancement layer as the "forward reference". A common characteristic of the layered scalable coding techniques is that the enhancement layer is either entirely transmitted/received/decoded or it does not provide any enhancement at all. The major difference between the ideal FGS and the layered scalable coding techniques is that, although the ideal FGS coding technique also codes a video sequence into two layers, the enhancement bit stream can be truncated at any number of bits within each frame to provide partial enhancement proportional to the number of bits de-

coded for each frame. Therefore, FGS is ideally supposed to provide a continuous scalability for the video stream.

B. Bit Plane Coding of DCT Coefficients

In the conventional DCT coding, the quantized DCT coefficients are coded using run-length coding. The number of consecutive zeros before a nonzero DCT coefficient is called a "run". If a VLC table is used, the (run, value) symbols are coded and a separate "EOB" symbol is used to show the end of the DCT block. The major difference between the bit-plane coding method and the run-length coding method is that the bit-plane coding method considers each quantized DCT coefficient as a binary number of several bits instead of a decimal integer [31], [33]. For each DCT block, the 64 absolute values are grouped in an array after scanning in zigzag order. A bit-plane of the block is defined as an array of 64 bits, that takes one bit from each absolute value of the DCT coefficients at the same significant position. For each bit-plane of each block, (RUN, EOP) symbols are formed and coded using variable-length codes to produce the output bit stream. Starting from the most significant bit-plane (MSB plane), the symbols are formed of two parts:

1. The number of consecutive zeros before a 1 (RUN)

2. End-of-plane (EOP)

If a bit-plane contains all zeros, a special symbol named ALL-ZERO is used to represent it. The following example illustrates the procedure. Assume that the absolute values and the sign bits after zigzag ordering are given as follows:

Absolute values: 7, 5, 6, 1, 0, 0, 0, 1, 4, 9,...., 0, 0, 0, 0

sign bits: 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,...., 0, 0, 0, 0

where a 0 represents a positive number and a 1 shows a negative number. Also all 0 values have been considered as positive numbers. The maximum value in this block is found to be 9 and the number of bits to represent 9 in the binary format (1001) is 4. Therefore, the 4

bit-planes are considered in forming the (RUN,EOP) symbols. Writing every value in the binary format, the 4 bit-planes are formed as follows:

MSB   0, 0, 0, 0, 0, 0, 0, 0, 0, 1,.... 0, 0, 0, 0

MSB-1   1, 1, 1, 0, 0, 0, 0, 0, 1, 0,.... 0, 0, 0, 0

MSB-2   1, 0, 1, 0, 0, 0, 0, 0, 0, 0,.... 0, 0, 0, 0

MSB-3   1, 1, 0, 1, 0, 0, 0, 1, 0, 1,.... 0, 0, 0, 0

Converting the four bit-planes into (RUN, EOP) symbols, we have

MSB   (9,1)

MSB-1   (0,0),(0,0),(0,0)(5,1)

MSB-2   (0.0),(1,1)

MSB-3   (0,0),(0,0),(1,0),(3,0),(1,1)

Therefore, 12 (RUN, EOP) symbols are formed in this example. These symbols are coded using variable-length code together with the sign bits, as shown below. Each sign bit is put into the bit stream only once immediately after the VLC code that contains the MSB of that nonzero absolute value. For the above example the variable length codes (VLC) of (RUN,EOP) symbols and their sign bit codes (S) are given below:

MSB   VLC(9,1),S(0)

MSB-1   VLC(0,0),S(0), VLC(0,0), S(1), VLC(0,0), S(0), VLC(5,1),S(0)

MSB-2   VLC(0,0),VLC(1,1)

MSB-3   VLC(0,0),VLC(0,0),VLC(1,0),S(0),VLC(3,0),S(1),VLC(1,1)

For example, no sign bit follows the VLC codes of the MSB-2 plane because the sign bits have been coded after the VLC code in the MSB-1 plane. It is possible to design the optimal VLC tables for each bit plane. Although the statistics of the bit planes with the same significance are very close the statistics of the bit planes with different significance can be very different[32]. Compared to the multi-layer method, the MSB plane can be considered as the base layer and all other planes as enhancements. From applicability of a continuous FGS

scheme however, the method provides the possibilty of truncating the enhancement layer bit stream at any point[34].

III. Discrete Wavelet Transform

The fundamental idea behind wavelets is to analyze according to scale[9]. Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions. However, in wavelet analysis, the scale that we use to look at data plays a special role. Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large window, we would notice gross features. Similarly, if we look at a signal with a small window, we would notice small features. The result in wavelet analysis is to see both the gross and small features. This makes wavelets interesting and useful. By their definition, sine and cosine functions are non-local (and stretch out to infinity). They therefore are very poor in approximating sharp spikes. But with wavelet analysis, we can use approximating functions that are contained neatly in finite domains. Wavelets are well-suited for approximating data with sharp discontinuities. The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion using coefficients in a linear combination of the wavelet functions, data operations can be performed using just the corresponding wavelet coefficients. If the coefficients below a threshold are truncated, the data is sparsely represented which makes wavelets an excellent tool for data compression.

A. Basis Functions

An example from a two dimensional vector space can help making the concept more clear. Every two-dimensional vector $< x, y >$ is a combination of the vector $< 1, 0 >$ and $< 0, 1 >$. These two vectors are the basis vectors for $< x, y >$ because x multiplied by $< 1, 0 >$ is the vector $< x, 0 >$ and y multiplied by $< 0, 1 >$ is the vector $< 0, y >$ and the sum is $< x, y >$. The best basis vectors have the valuable extra property that the vectors are perpendicular, or orthogonal to each other. For the basis $< 1, 0 >$ and $< 0, 1 >$ this criteria is satisfied. A similar discussion is valid when considering the basis functions in a wavelet transform. Here instead of the vector $< x, y >$ we have a function $f(x)$. We can construct $f(x)$ by adding sines and cosines using combinations of amplitudes and frequencies. The sines and cosines are the basis functions of Fourier synthesis. By choosing the appropriate combination of sine and cosine function terms whose inner product add up to zero, we can set the additional requirement that they be orthogonal. The different wavelet families of basis functions or mother wavelets have been introduced and used in different applications. It is important to note that wavelet transforms do not have a single set of basis functions like the Fourier transform, which utilizes just the sine and cosine functions. Instead, wavelet transforms have an infinite set of possible basis functions. Some of these wavelets are given in figure 4.

B. Scale-varying Basis Functions

A basis function varies in scale by chopping up the same function or data space using different scale sizes. For example, if we have a signal over the domain from 0 to 1, We can divide the signal with two step functions that range from 0 to 1/2 and 1/2 to 1. Then we can divide the original signal again using four step functions from 0 to 1/4, 1/4 to 1/2, 1/2 to 3/4, and 3/4 to 1, and so on. Each set of representations code the original signal with
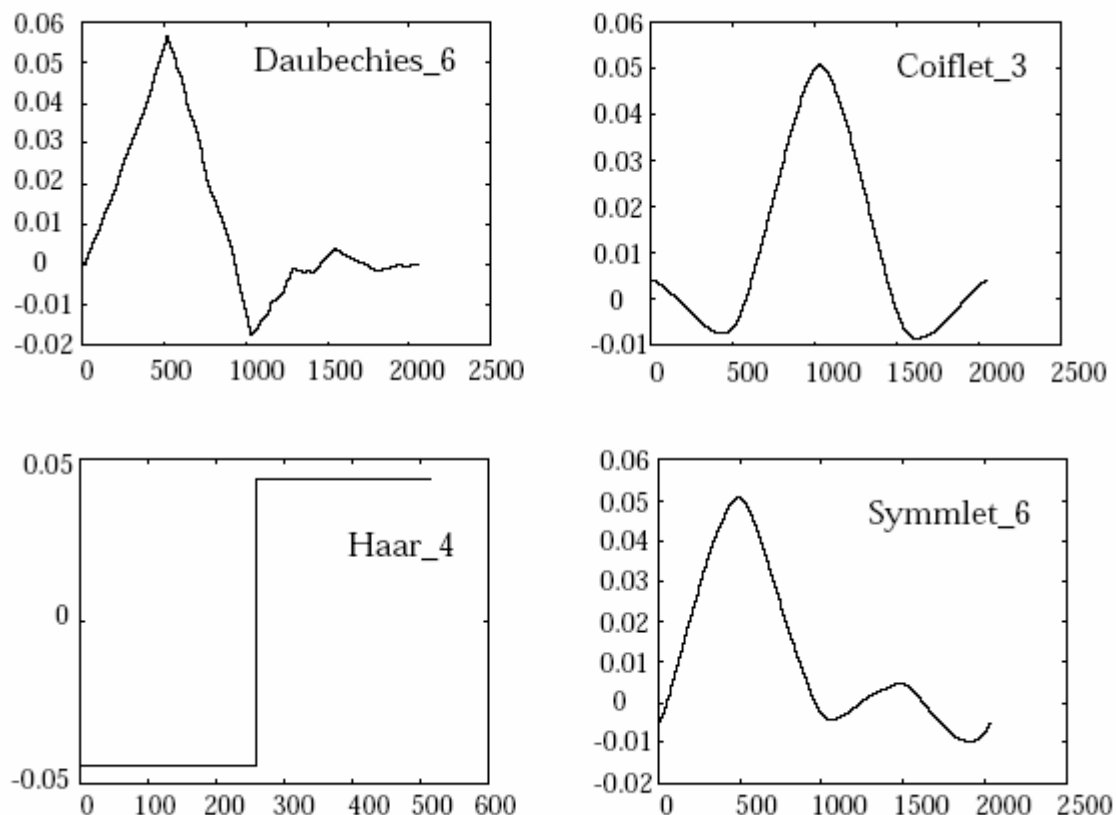
Fig. 4. Typical wavelets

a particular resolution or scale. [8] The most interesting feature of DWT is that individual wavelet functions are localized in space. This localization feature, along with wavelets' localization of frequency, makes many functions and operators using wavelets sparse when transformed into the wavelet domain. This sparseness, in turn, results in a number of useful applications such as data compression, detecting features in images, and removing noise from time series. In order to isolate signal discontinuities, one would like to have some very short basis functions. At the same time, in order to obtain detailed frequency analysis, one would like to have some very long basis functions. A way to achieve this is to have short

high-frequency basis functions and long low-frequency ones. This is exactly what we get with wavelet transforms. To span our data domain at different resolutions, the analyzing wavelet is used in a scaling equation:

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \Psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right) \tag{1}$$

Although it is called a discrete wavelet, it normally is a (piecewise) continuous function. In 1 j and k are integers and $s_0 > 1$ is a fixed dilation step. The translation factor 0 depends on the dilation step. The effect of discretizing the wavelet is that the time-scale space is now sampled at discrete intervals. We usually choose $s_0 = 2$ and for the translation factor we usually choose $\tau_0 = 1$.

IV. DWT Scalability

The wavelet transform has also been used for providing scalability characteristic for a video stream since it allows localization in both the space and frequency domains [9], [12], [13], [14]. Typically an image is decomposed into a hierarchy of frequency sub-bands that are processed in an independent manner. The decomposition is achieved by filtering along one spatial dimension at a time to effectively obtain four frequency bands as shown in Figure 5. The lowest sub-band, commonly referred to as Low-Low (LL), represents the information at coarser scales and it is decomposed and sub sampled to form another set of four sub-bands. This process can be continued until the number of levels of decomposition is attained. The analysis filters **h** and **g** efficiently decompose the image into independent frequency spectra of different bandwidths or resolutions [15], [14], producing different levels of detail. 'h' and 'g' are commonly referred to as the analysis filters. Here L and H stand for low and high frequency bands respectively. The various subband signals are recombined so that the original signal is reconstructed. The reconstruction is accomplished by upsampling
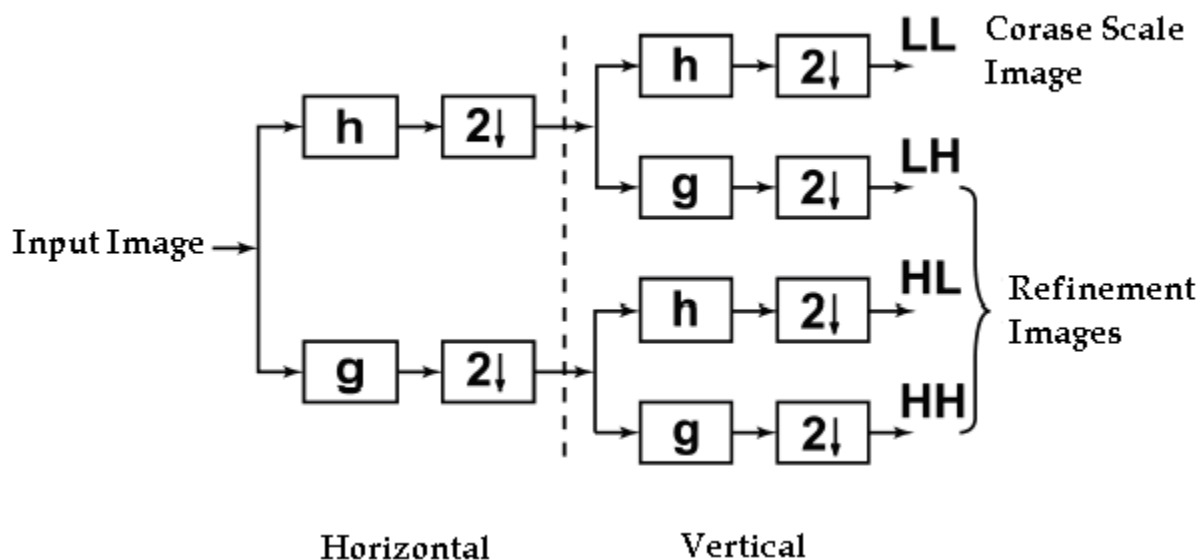
Fig. 5. One level of the wavelet transform decomposition

the lower resolution images and passing them through synthesis filters, as shown in Figure 6. In this case the filters used are referred to as the synthesis filters. Various types of analysis filters have been proposed in image and video compression. To use the wavelet transform for image compression we must quantize and binary encode the wavelet coefficients. Typically wavelet coefficients with large magnitude (or high energy) are assigned more bits and hence have a higher precision. Some of the coefficients are given zero bits and therefore not included in the compressed representation of the image. As in all transform coding techniques the location of the quantized coefficients must also be known by the decoder. This is accomplished using various techniques such as the zigzag scanning used in the JPEG and MPEG standards. Other approaches have also been introduced to organize the wavelet coefficients into quadtrees that allow a very compact representation for image compression and rate scalability.
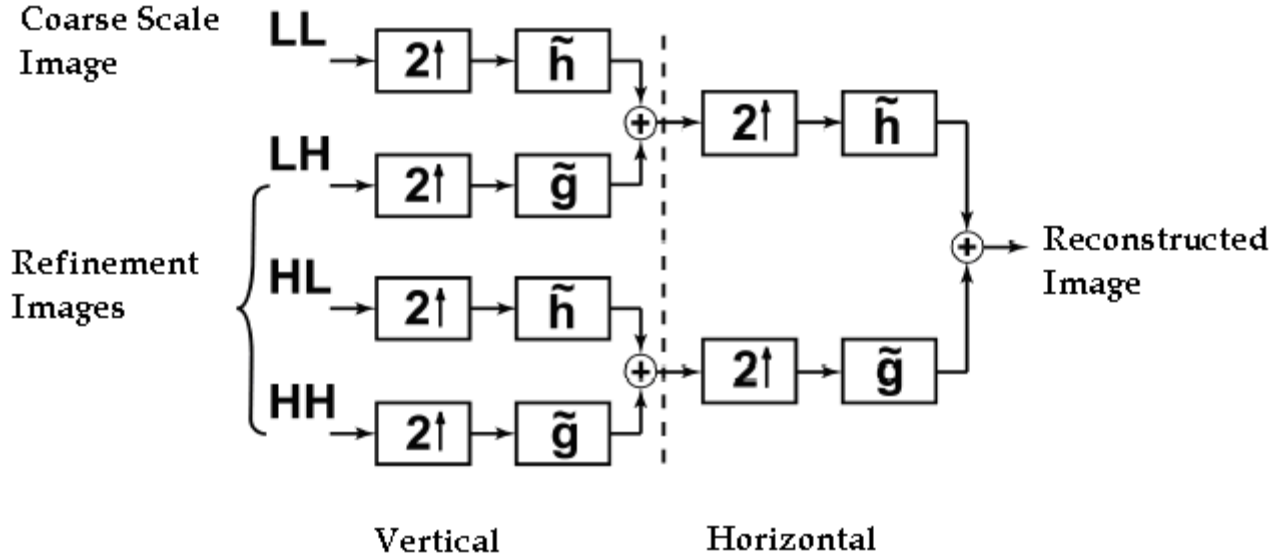
Fig. 6. One level of the wavelet transform reconstruction

A. Spatial Orientation Trees

As with DCT coefficients, most of the wavelet coefficients in the high frequency bands are very small. These small values are replaced by zero after the quantization step[13], [15]. The discrete wavelet transform however, has the extra characteristic of self similarity. If we consider a multilayer decomposition of an image using DWT, where the lower levels correspond to higher frequencies and higher levels correspond to lower frequencies, we can easily observe a decrease of energy when moving from a high frequency level to a low frequency level. Also the coefficients at a low level contain small energy, they coefficients at the same spatial orientation at a higher level will also contain low energy. This similarity between the coefficients at similar spatial locations of a multilayer wavelet decomposition is called self similarity characteristic. This characteristic can be exploited to reduce the size of bit stream data. Figure 7 shows the hierarchical structure of wavelet coefficients in a multilayer decomposition. Shapiro observed that using quadtree representations of the wavelet coeffi-
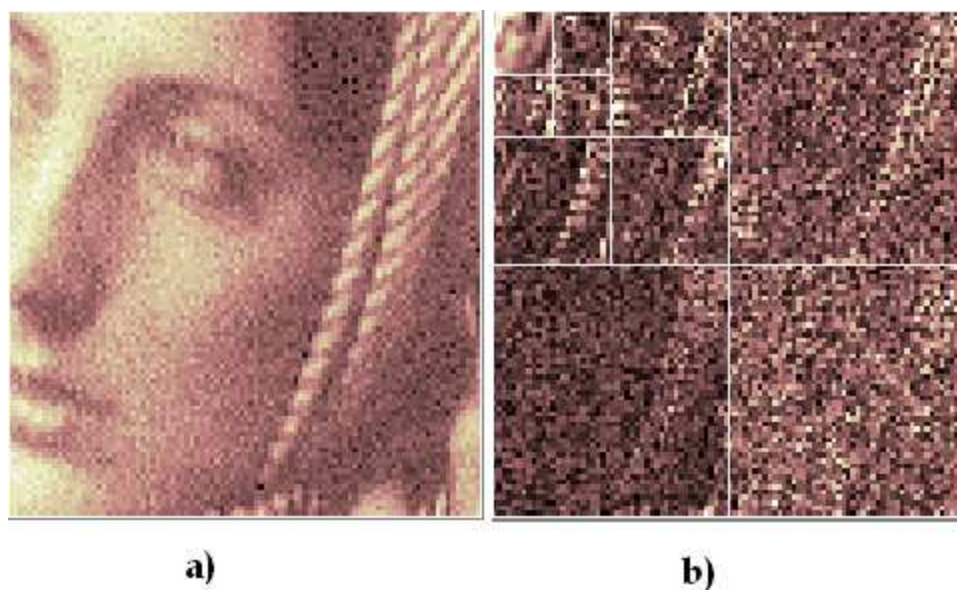
Fig. 7. Self-similarity of a Multilevel Wavelet Decomposition,a)original image, b)wavelet coefficients

cients provided a method for grouping the coefficients that belong to different subbands but have the same spatial location into one structure [10]. This structure, known as a spatial orientation tree (SOT), can then be represented by a small number of symbols. SOTs have been widely used in rate scalable coding of still images and video [10]. The self similarity feature of wavelet coefficient pyramid allows for the coding of a large number of insignificant wavelet coefficients by only coding the location of the root coefficient to which the entire set of coefficients are related. Such a set is commonly referred to as a zerotree [10]. . Rate scalability is achieved by organizing the coefficients in order of importance and progressively quantizing and binary encoding the wavelet coefficients. In this manner, by decoding the initial portion of the bit stream a coarse version of an encoded image is obtained at the receiving end. The image is then refined by decoding additional data from the compressed data stream. This process is terminated when a desired data rate or distortion is attained. If encoding of a bit stream can be terminated at any point where a requirement is met then it is called an embedded coding method. As with the DCT coding, one of major issues is

coding the position of non-zero coefficients. Two main zerotree based methods which use the self similarity of wavelet coefficients are Embedded Zerotree Wavelet (EZW) and Set Partitioning In Hierarchical Trees (SPIHT). The major difference between the two techniques lies in the fact that they use different strategies to scan the transformed coefficients and perform the encoding.

B. EZW

In the embedded zerotree wavelet (EZW) scheme, developed by Shapiro [10]. , the wavelet coefficients are grouped into Spatial Orientation Trees. To describe these data structures, first some terms are defined.

Threshold Value: All coefficient values are compared to a given value which is called a threshold value.

Significant: A coefficient value which is larger than the threshold value is called significant otherwise it is insignificant.

Zerotree: A tree which all if its values including root value are insignificant with respect to a threshold value.

Isolated Zero: If a coefficient value is insignificant but it is not the root of a zerotree because of a significant value at its lower nodes is called an isolated zero.

Descendants: All nodes which are either direct children of a node or are descendants of its children are called its descendants.

After applying the wavelet transform, a zerotree encoding algorithm described below is executed. First a quad-tree is created using the wavelet coefficients as shown in figure 8. Then the magnitude of each wavelet coefficient in the quad-tree, starting with the root of the tree located in the LL band of the decomposition, is compared to a threshold T. If the magnitudes of all the wavelet coefficients in the tree are smaller than T, the entire tree structure (that is
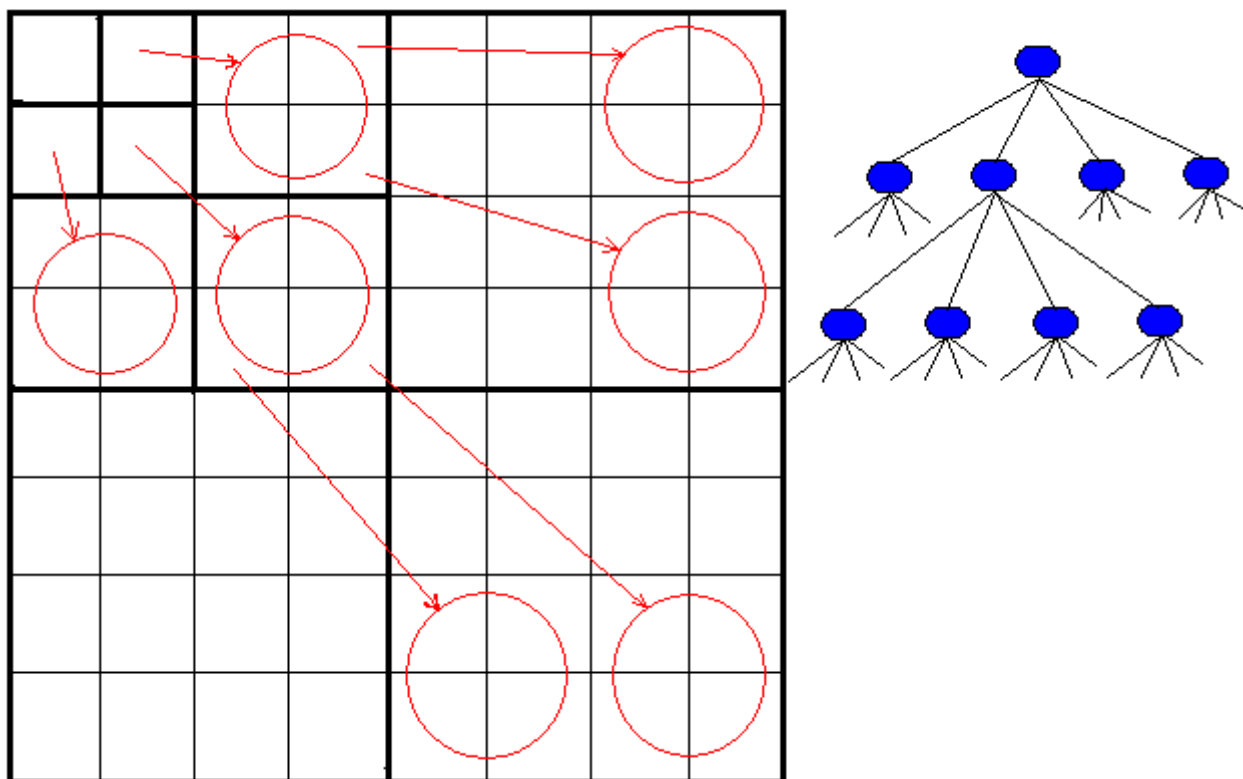
Fig. 8.   Quad-tree representation of the Wavelet coefficients

the root and all its descendant nodes) is represented by one symbol, the zerotree symbol t. If however, there exist significant (greater than T) wavelet coefficients in the tree, then the tree root is either represented as significant if its magnitude is greater than T, or insignificant when its magnitude is smaller than T. The descendant nodes are then each examined in turn to determine whether each is the root of a possible subzerotree structure, or not. This process is carried out such that all the nodes in all the trees are examined for possible subzerotree structures. The significant wavelet coefficients in a tree are represented by one of two symbols, P or N, depending on whether their values are positive or negative, respectively. An insignificant coefficient which is not in a zerotree is represented by a Z symbol. The process of classifying the coefficients as t,Z, P, or N is referred to as the dominant pass. This

is then followed by a subordinate pass in which the significant wavelet coefficients in the image are refined by determining whether their magnitudes lie within the intervals [T,3T/2) and [3T/2,2T). Those wavelet coefficients whose magnitudes lie in the interval [T,3T/2) are represented by the symbol 0 (LOW), whereas those with magnitudes lying in the interval [3T/2,2T) are represented by the symbol 1 (HIGH). To refine the coefficients which were marked either as P or N in subordinate pass, we push them in a FIFO in the dominant pass. These coefficients are reduced by current threshold value after refining. Subsequent to the completion of both the dominant and subordinate passes, the threshold value T is reduced by a factor of 2, and the process is repeated. To avoid recoding significant coefficients in the next pass, these coefficients are replaced by 0 in the image. This coding strategy, consisting of the dominant and subordinate passes followed by the reduction in the threshold value, is repeated until a target bit rate is achieved. This threshold reduction essentially acts as a uniform quantizer of the coefficients. The sequence in which the coefficients are examined is predefined and known by the decoder. This is sometimes referred to as a zerotree scanning order which is mostly a zig-zag order used in MPEG coding standard and shown in Figure 9. The compressed bit stream therefore consists of the initial threshold value, the tree symbols and the subordinate bits. This information is binary encoded using an entropy encoder. The tree symbols along with the scanning order describe where the wavelet coefficients are located in the quad-tree. These symbols and the subordinate bit stream are used to obtain their quantization values. An EZW decoder reconstructs the image by progressively updating the values of each wavelet coefficient in a tree as it receives the data. The decoder's decisions are always synchronized to those of the encoder, making this algorithm highly sensitive to transmission errors. To make the encoding method more clear two steps of EZW encoder are given in the following example. Assuming the initial threshold value being 32 we have:
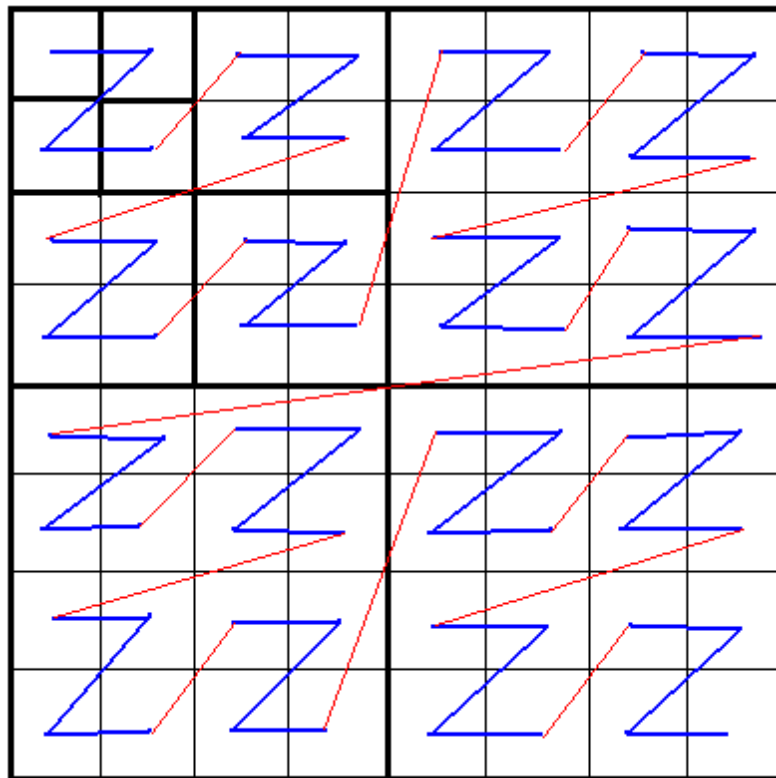
Fig. 9. Zig-Zag scanning order of the wavelet coefficients

Dominant Pass 1: PNZtPttttZttttttttPtt

Coefficients in FIFO : 61, -40, 51, 52

Subordinate Pass 1: 1011

Pass 2, Threshold = 16

Dominant Pass 2: ZtNPtttttttt

Coefficients in FIFO : 29, -8, 19, 20, -29, 25

Subordinate Pass 2: 100011

61	-40	51	9	6	14	-15	6
-29	25	12	-14	2	5	3	2
14	15	1	-8	5	-2	1	11
-12	6	15	7	3	-1	4	1
-2	10	1	52	1	3	2	-1
1	0	3	2	3	-1	4	0
2	-1	6	3	2	4	2	5
2	10	4	3	0	2	-1	5

Fig. 10. Example data for wavelet coefficients

C. SPIHT

Spatial Set Partitioning In Hierarchical Tree or SPIHT is similar to EZW, in that it performs a partial ordering of the coefficients using a set of decreasing thresholds which are powers of two. In this case, the initial threshold corresponds to the largest power of two that is smaller than the magnitude of the largest coefficient [11]. The method implements scalability using the same idea as described in multi-plane scalable data where the bits from the most significant bit plane are sent first. However, SPIHT optimizes this method using the observation that if the coefficients are ordered by their number of significant bits, then the total number of transmitted data bits will be reduced. In ordering the coefficients by the number of significant bits, it does not consider the magnitude of the coefficient and two values

with the same number of significant bits may have different magnitude values. For example 17 (10001b) and 19 (10011b) have the same number of significant bits but different values. This ordering method results in partitioning the coefficients into sets with respect to the number of significant bits as shown in Figure 11 where each column represents a coefficient. Furthermore, SPIHT defines μ_n as the number of coefficients at each set [11]. For instance μ_4
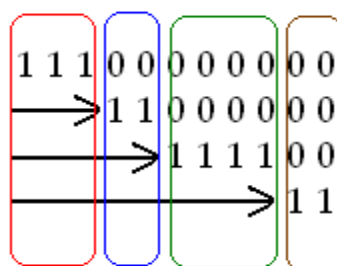

Fig. 11. Partitioning coefficients based on the number of significant bits

shows the number of coefficients c so that $2^4 < c < 2^5$. This helps encoder ignore the leading zeros and the first 1 bit. The main issue in SPIHT therefore is that the coefficients are not in any order as is assumed by the method. It uses a special sorting algorithm described below to partition the coefficients into sets. To find the correct position of the coefficients by the receiver, the encoder sends the results of comparisons in the sorting stage. If a comparison has a result of FALSE, a zero bit is sent otherwise a one bit is transmitted. The decoder can reverse the sorting process and find the locations of the coefficients. SPIHT consists of four steps.

- Initialization
- Sorting
- Refinement
- Quantization scale update

During the coding, the wavelet transformed coefficients are first organized into a quad-tree. This tree is similar to the one used in EZW encoding and shown in Figure 8. SPIHT groups the wavelet coefficients into three lists, the list of insignificant sets (LIS), the list of insignificant pixels (LIP), and the list of significant pixels (LSP). In the first step or initialization step, these lists are initialized to the set of subtree descendants of the nodes in the highest level, the nodes in the highest level, and an empty list, respectively. During the sorting pass, the algorithm traverses through the LIP testing the magnitude of its elements against the current threshold. If a coefficient from LIP has a value greater than the threshold it is moved to LSP and a 1 is sent as output otherwise a 0 is sent. The algorithm then examines the list of insignificant sets and performs a magnitude check on all the coefficients in at set. If a particular set is found to have significant items, it is then partitioned into subsets and tested for significance, otherwise a single bit is appended to the bit stream to indicate an insignificant set. During the partitioning process, the coefficients which are significant are mover to LSP and the rest are inserted into LIP. After a pass through the LIS is completed, a refinement step through the LSP, excluding those coefficients added during the previous sorting step, is initiated. The refinement step is accomplished by sending the bits at position m of each coefficient which has been in LSP during the previous pass where m shows the position of the most significant bit in the first pass and is decremented at each iteration. In the last step the threshold is decreased by a factor of two (resulting in uniform quantization of the coefficients) and the entire process is repeated from the sorting step until the maximum bit rate allowed is reached or the threshold is smaller than a minimum value. Since all branching decisions made by the encoder as it searches throughout the coefficients are appended to the bit stream, the locations of the coefficients being refined or classified is never explicitly transmitted. The output of the sorting-refinement process is then coded via

a variable length code (VLC) encoder. The decoder recreates the decisions of the encoder to reconstruct the image. The example below clarifies the procedure. The initial threshold
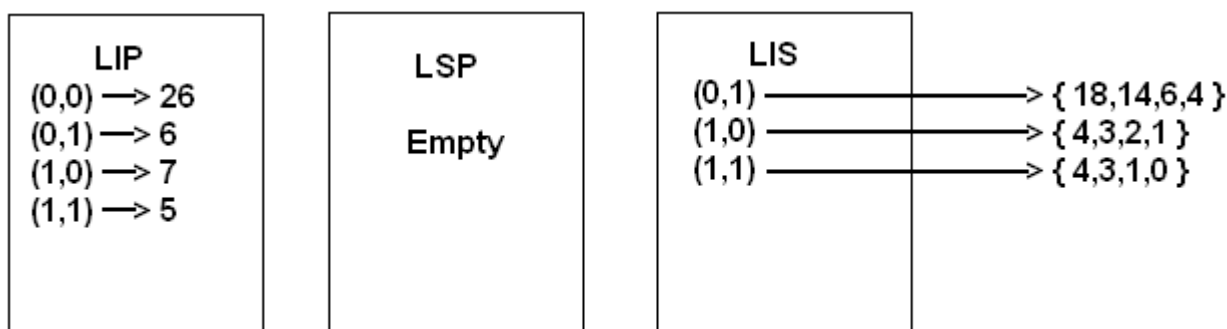

Fig. 12. Example data for SPIHT

Fig. 13. Initialization of LIP, LSP and LIS

value is 16. First LIP is examined and 26 is moved to LSP. A 1 is added to the output bit stream. Then three 0s are sent to the output for the remaining three coefficients in the LIP. Then LIS is examined. First set contains two significant elements and two insignificant ones. A 1 bit sent for the set, two 1s for two significant members and two 0s for two insignificant ones. Two significant elements are inserted to LSP and two insignificant ones to LIP. Three more 0s are sent for the remaining three members of LIS. As this pass is the first pass, no refinment is carried out. The output bit stream is: 100011100000 and the sets are as shown
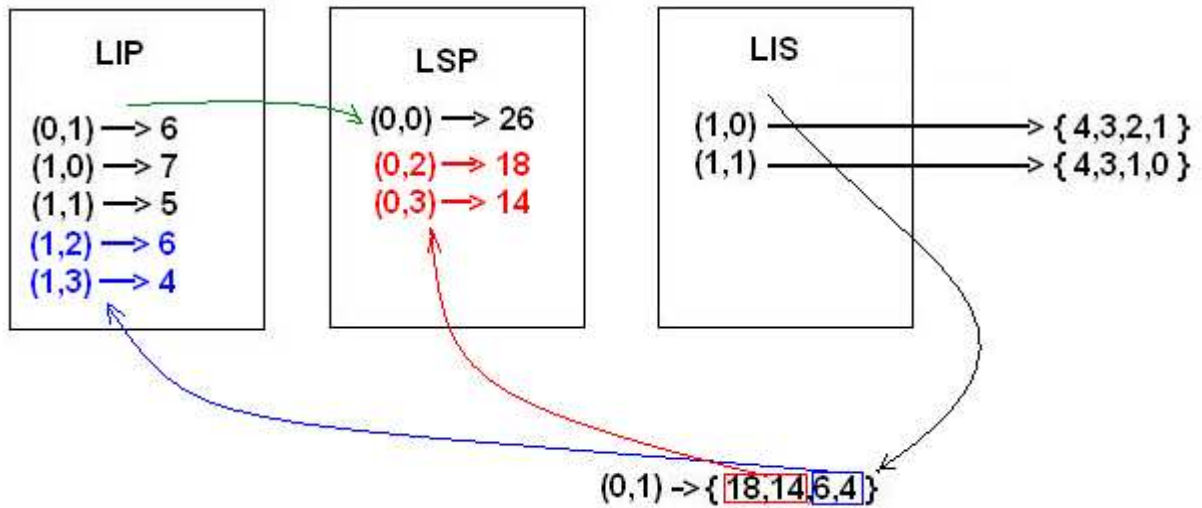
in Figure 14.

Fig. 14.   LIP, LSP and LIS after first pass

A second example to show how SPIHT works follows. Assume the data is as given in Figure 15.

Fig. 15.   Example Data, SPIHT

Initialize: LIP = (1,1) , LIS = (1,1)D , LSP =

Dominant Pass 1:

T = 16

Is (1,1) significant? yes: 1

LSP = (1,1) 1 (sign bit)

Is D(1,1) significant? no: 0

LSP = (1,1) , LIP = {}, LIS = (1,1)D ⇒ 3 bits are sent.

Dominant Pass 2:

T = 8

Is D(1,1) significant? yes: 1

Is (1,2) significant? no: 0

Is (2,1) significant? no: 0

Is (2,2) significant? no: 0

LIP = (1,2), (2,1), (2,2) , LIS = (1,1)L

Is L(1,1) significant? yes: 1

LIS = (1,2)D, (2,1)D, (2,2)D

Is D(1,2) significant? yes: 1

Is (1,3) significant? yes: 1

LSP = (1,1), (1,3) 1 (sign bit)

Is (2,3) significant? yes: 1

LSP = (1,1), (1,3), (2,3) 1 (sign bit)

Is (1,4) significant? no: 0

Is (2,4) significant? no: 0

LIP = (1,2), (2,1), (2,2), (1,4), (2,4) , LIS = (2,1)D, (2,2)D

Is D(2,1) significant? no: 0

Is D(2,2) significant? no: 0

LIP = (1,2), (2,1), (2,2), (1,4), (2,4) , LIS = (2,1)D, (2,2)D , LSP = (1,1), (1,3), (2,3)

⇒ 14 bits are sent.

V. Discussion

There is a growing need for a video-coding standard to deliver video over a channel, such as the Internet, with a wide range of bit rates and a wide range of bit rate variations. The features in FGS coding are chosen to balance coding efficiency, implementation complexity and bit rate variations. However, most of the proposed methods suffer from the problem of low compression ratio compared to the motion compensated coding standards. This low performance is due to the fact that the temporal redundancy in the video sequence is not fully exploited. Also since multi subband decomposition requires multiple frames to be processed at the same time, more memory is needed for both the encoder and the decoder, which results in delay. The second problem is the discontinuity introduced by FGS methods. In these methods the granularity defines the bit rate. Even bit-plane coding of the DCT coefficients which is a very natural and easy way to meet the requirement of gradual changes in video quality when user channel bandwidth changes introduces a multi layer structure which is equivalent to the number of bits used for each quantization value. This restrictions brings with it the problem of efficient encoding of the prediction error values. Residue values obtained from comparing the macroblocks with the base layer in FGS will reduce the performance of the encoder and comparing with base plus enhancement layer values will cause drift problem if the decoder of the receiver can not go beyond base layer. Compared with non-scalable coding, which is the upper bound for any scalable coding techniques, FGS always shows worse performance than non-scalable coding methods.

References

[1] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, "Video Coding for Streaming Media Delivery on the Internet", IEEE Transaction on Circuits and Systems for Video Technology, March 2001.

[2] D. Wu, Y. Hou, W. Zhu, Y.Q. Zhang, and J. Peha, "Streaming Video over the Internet: Approaches and Directions", IEEE Transactions on Circuits and Systems for Video Technology, March 2001.

[3] "Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile", ISO/IEC 14 496–2/FPDAM4, July 2000.

[4] B. G. Haskell, A. Puri, and A. N. Netravali, "Digital Video: An Introduction to MPEG-2", New York: Chapman and Hall, Sept. 1996.

[5] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG–2 scalable video coding algorithm", IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 426-435, October 1996.

[6] International Telecommunication Union (ITU-T), ITU-T Recommendation H.263 Version 2: Video Coding for Low Bit Rate Communication, February 1998. (H.263+).

[7] T. Turletti and C. Huitema, "Video-conferencing on the Internet", IEEE/ACM Transaction on Networking, vol. 4, pp. 340-351, June 1996.

[8] G. Strang, "Wavelets", American Scientist, Vol. 82, pp. 250–255, 1992.

[9] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 1, pp. 109–122, February 1999.

[10] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients", IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3445–3462, December 1993.

[11] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243–250, June 1996.

[12] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", IEEE Transac-

tions on Image Processing, vol. 3, no. 5, pp. 572–588, September 1994.

[13] M. L. Comer, K. Shen, and E. J. Delp, "Rate-scalable video coding using a zerotree wavelet approach", in Proceedings of the Ninth Image and Multidimensional Digital Signal Processing Workshop, Belize City, Belize, pp. 162–163, March 1996.

[14] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 1, pp. 109–122, February 1999.

[15] Q. Wang and M. Ghanbari, "Scalable coding of very high resolution video using the virtual zerotree", IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 5, pp. 719-727, October 1997.

[16] E. J. Delp, P. Salama, E. Asbun, M. Saenz, and K. Shen, "Rate scalable image and video compression techniques", Proceedings of the 42nd Midwest Symposium on Circuits and Systems, Las Cruces, New Mexico, August 1999.

[17] W. Tan and A. Zakhor,"Real-Time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol", IEEE Transaction on Multimedia, Vol. 1, NO. 2, June 1999

[18] Li Weiping ,"Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Transaction on Circuits and Systems for Video Technology, VOL. 11, NO. 3, March 2001.

[19] A. Puri and T. Chen, "Multimedia Systems, Standards, and Networks", New York: Marcel Dekker, March 2000.

[20] M. Ghanbari, "Two-layer coding of video signals for VBR networks", IEEE Journal of Selected Areas Communications, vol. 7, pp. 771781, June 1989.

[21] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream

scaling", IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 191-199, April 1996.

[22] H. Gharavi and M. H. Partovi, "Multilevel video coding and distribution architectures for emerging broadband digital networks", IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 459-469, October 1996.

[23] H. Jiang, "Experiment on post-clip FGS enhancement", ISO/IEC JTC1/SC29/WG11, MPEG00/M5826, March 2000.

[24] W. Li and Y. Chen, "Experiment result on fine granularity scalability", ISO/IEC JTC1/SC29/WG11, MPEG99/M4473, March 1999.

[25] M. Domanski, A. Luczak, and S. Mackowiak, "Spatial-temporal scalability for MPEG video coding", IEEE Transaction on Circuits and Systems for Video Technology, vol. 10, pp. 1088-1093, October 2000.

[26] H. Katata, N. Ito, and H. Kusao, "Temporal-scalable coding based on image content", IEEE Transaction on Circuits and Systems for Video Technology, vol. 7, pp. 52-59, February 1997.

[27] G. J. Conklin and S. S. Hemami, "A comparison of temporal scalability techniques", IEEE Transaction on Circuits and Systems for Video Technology, vol. 9, pp. 909-919, September 1999.

[28] D. Wilson and M. Ghanbari, "Exploiting interlayer correlation of SNR scalable video", IEEE Transaction on Circuits and Systems for Video Technology, vol. 9, pp. 783-797, August 1999.

[29] R. Mathew and J. F. Arnold, "Layered coding using bitstream decomposition with drift correction", IEEE Transaction on Circuits and Systems for Video Technology, vol. 7, pp. 882-891, December 1997.

[30] M. Ghanbari and J. Azari, "Effect of bit rate variation of the base layer on the performance of two-layer video codecs", IIEEE Transaction on Circuits and Systems for Video Technology, vol. 4, pp. 8-17, February 1994.

[31] W. Li, F. Ling, and H. Sun, "Bitplane coding of DCT coefficients,",ISO/IEC JTC1/SC29/WG11, MPEG97/M2691, October 1997.

[32] F. Ling, W. Li, and H. Sun, "Bitplane coding of DCT coefficients for image and video compression", in Proceedings of SPIE Visual Communications and Image Processing (VCIP), San Jose, CA, January 1999.

[33] W. Li, "Fine granularity scalability using bit-plane coding of DCT coefficients", ISO/IEC JTC1/SC29/WG11, MPEG98/M4204, December 1998.

[34] F. Ling and X. Chen, "Report on fine granularity scalability using bitplane coding", ISO/IEC JTC1/SC29/WG11, M4311, November 1998.