Chapter 5

An Overview of Wavelet Based Image Compression Schemes

A fundamental goal of image data compression is to reduce the bit-rate for transmission and/or storage, while maintaining an acceptable image quality. Compression can be achieved by transforming the data orthogonally , projecting it on a basis function and then encoding these transformed coefficients. Because of the nature of the image signal and mechanism of human vision, the transform must be non-stationary and well localized, both in space and frequency domains. Moreover, the corresponding algorithm must be fast to avoid jitter at the output. Wavelet transform can localize in both time and frequency domains, is fast, and hence, best suited to satisfy the above requirements.

This chapter is organized as follows: Section 1 gives a brief introduction to wavelet-based image compression. In section 2 we discuss the zerotree and embedded zerotree wavelet algorithms. In section 3 we discuss the set partition in hierarchical trees scheme. Section 4 summarizes the chapter.

5.1 Introduction

It can be shown [64] that only about 10 to 15% of the total number of wavelet coefficients are sufficient for a reasonable compression ratio. However, in the Discrete Wavelet Transform (DWT), location of the significant coefficients (or *significant maps*) keeps on changing, unlike the Discrete Cosine Transform (DCT) case. This leads to a poor quality of compressed image. Some way of implicitly reducing the significant maps has to be worked out.

One of the early attempt in this direction was by Lewis and Knowles[77], where, inter-dependencies between sub-bands of different levels was exploited to code the location of significant coefficients. A similar approach was proposed by Shapiro [127] to predict the location of *insignificant* coefficients rather than significant ones, and thereby has coined the term *zerotree* data structure.

Using the zerotree hypothesis, embedded zerotree of wavelet coefficients (EZW) algorithm is developed by Shapiro [127] for a very low bit rate application. The EZW algorithm is based on the following key concepts:

1. Exploiting self-similarity inherent in the wavelet transform to predict the absence of significant information across different resolutions,

2. Successive-approximation quantization of significant wavelet coefficients and

3. Lossless data compression using adaptive arithmetic coding.

This technique is not only comparative in performance to other coding techniques, but also is extremely fast in execution and produces an embedded bit-stream. With an embedded bit-stream, the reception of code bits can be stopped at any point and the image can be reconstructed. Following this significant work, Said and Perlman [119] developed an alternative technique to EZW for better results. They called this technique *Set Partitioning In Hierarchical Trees* (SPIHT) algorithm. Zerotree entropy coding (ZTE) [52, 114] is a new technique for coding wavelet transform coefficients, which is based on EZW and SPIHT, but is significantly different.

Above mentioned algorithms have been successfully utilized in the inter-dependencies exhibited in wavelet transform for image compression using "zerotree" data structure, and are discussed below.

5.2 Zerotree

The wavelet pyramid gives a multiresolution representation of the input image. In such a representation, many coefficients are nearly zero and are *insignificant*. This is evident from Fig 5.1, for one level of wavelet decomposition. Of the total number of bits available for coding, a large number of bits must be spent on encoding the position of these coefficients that will be transmitted or stored as a non-zero value. The position of these coefficients is called *significant map*. It follows that improvement in encoding significant maps is reflected as gain in compression efficiency. To improve compression, a new data structure called *zerotree* is defined. The zerotree is based on the hypothesis that if a coefficient at a coarser level in a multiresolution decomposition is insignificant with respect to a given

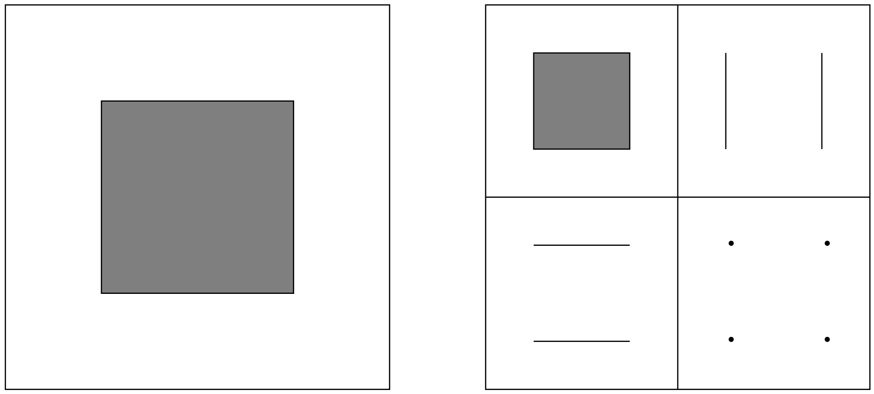

Figure 5.1: Wavelet decomposition. Left:Original synthetic square, Right: One level wavelet decomposition

threshold [1], θ then all coefficients in the same spatial location at finer levels are also insignificant with respect to the same threshold θ.

Specifically, in hierarchical decomposition systems, with the exception of the finest level, every coefficient at a given level can be related to a set of coefficients at the next finer level. The coefficient at the coarser level is called the *parent* and all coefficients corresponding to the same spatial location at the next finer level are called its *children*. For a given parent, the set of coefficients at all finer levels corresponding to the same location are called the *descendants*. Similarly, for a given child, the set of coefficients at all coarser levels corresponding to the same location are called *ancestors*. These relations are depicted in the Fig. 5.2. With the exception of lowest frequency sub-band corresponding to LL3, all parents have four children. For the lowest frequency sub-band, the parent-child relationship is defined such that each parent node has three children.

Zerotree coding reduces the cost of encoding the significance map using self-similarity across the levels. Even though the image has been transformed using a de-correlating transform, the occurrences of significant coefficients are not independent events. This technique is quite different from the previous attempts to exploit self-similarity across different scales in image coding [77, 115], as it is far easier to predict insignificant coefficients than to predict significant details across the scales. The focus is on reducing the cost of encoding the significance map, so that, for a given bit budget, more bits are available to encode expensive significant coefficients.

[1] a coefficient x is insignificant with respect to θ if $|x| < \theta$

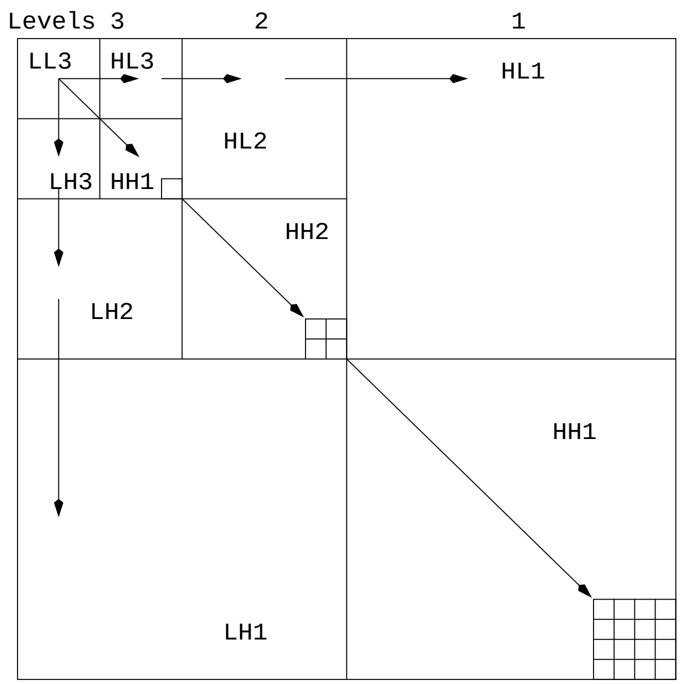

Figure 5.2: Parent offspring dependencies of sub-band: Arrow points from the sub-band of the parents to the sub-band of the children. LL3 is the lowest frequency and HH1 is the highest. Level 3 is the coarsest resolution and level 1 is finest resolution. Also shown is the wavelet tree consisting of all descendants of a single coefficient in sub-band HL3
.

5.2.1 The EZW Algorithm

A coefficient is said to be an element of *zerotree* with respect to a given threshold θ, if along with all its dependents, it is insignificant with respect to θ. An element of a zerotree, for a threshold θ, is a zerotree root if it is not a descendant of a previously found zerotree root for that threshold θ. A zerotree is encoded with a special symbol indicating that the insignificance of the coefficients at a finer level is completely predictable. The significance map is efficiently represented as a string of symbols from a four symbol alphabet, which is then entropy-coded. The four symbols are (a) zerotree root, (b) isolated zerotree - where the coefficient is insignificant, but the descendants are not, (c) positive and (d) negative significant coefficient. Algorithm for EZW is given below:

```
BEGIN
read a coefficient
    IF the coefficient is significant
    BEGIN
       code it as positive if the coefficient is positive;
       code it as negative if the coefficient is negative;
    ENDIF
    ELSE
        IF coefficient is descendant of a zerotree root
    do not code ;/* insignificant*/
ENDIF
ELSE
    IF coefficient have significant descendants
       code an isolated zero symbol;
    ENDIF
    ELSE  code zerotree root symbol  ;
        ENDELSE
    ENDELSE
END.
```

Scanning methodology ensures that no child node is scanned before its parent and each of the coefficients at a given level is scanned before any coefficient in the next finer level. Fig. 5.3 shows a scanning pattern for a 3-level wavelet transform

To perform embedded coding, successive approximation quantization (SAQ) on scalars is done. The SAQ sequentially applies a sequence of thresholds $\theta_0, \theta_1 \ldots \theta_{N-1}$ to determine

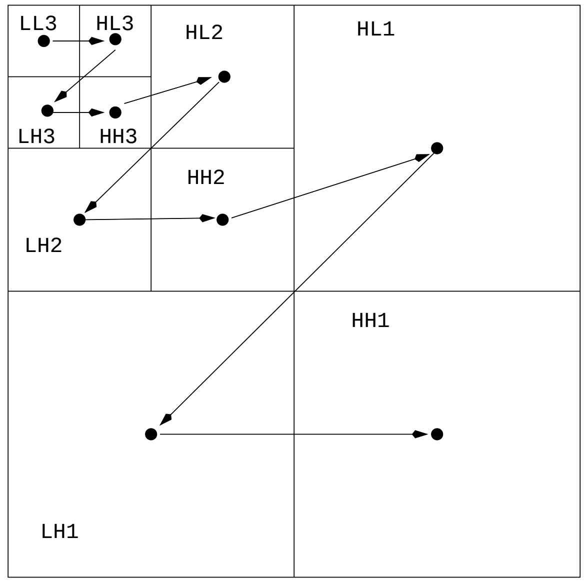

Figure 5.3: Scanning pattern of the sub-bands for encoding a significant map. Parents are scanned before children and all positions in a given sub-band are scanned before the scan moves to the next sub-band

significance. At each iteration or step, a new threshold is chosen so that $\theta_i = \theta_{i-1}/2$. The initial threshold is chosen so that $|x_j| < 2\theta_0$ for all transform coefficients x_j.

During encoding and decoding, two separate lists of wavelet coefficients are maintained. The *dominant* list contains the *coordinates* of coefficients that are not yet been found to be significant in the same relative order as the initial scan. In this scan, sub-bands are ordered, and, within each sub-band, the set of coefficients are ordered. As per the scanning process of Fig.5.3 all coefficients in a given sub-band appear on the initial dominant list prior to coefficients in the next sub-band. The *subordinate* list contains the *magnitudes* of those coefficients that have been found to be significant. For individual thresholds, each threshold list is scanned once.

During a *dominant pass*, coefficients with coordinates on the dominant list are compared to the threshold θ_i, to determine their significance, and if significant, its magnitude is appended to the subordinate list, and the coefficient in the dominant list is set to zero so that the significant coefficient does not prevent the occurrence of a zerotree on future dominant passes at smaller thresholds.

The dominant pass is followed by a *subordinate* pass. In this, all coefficients on the subordinate list are scanned and specifications of the magnitudes available to the decoder are refined to an additional bit of precision. During this pass, the *width* of the effective quantizer step size is cut to half. The process alternates between two passes and the threshold is halved before each dominant pass.

In the decoding phase, each decoded symbol refines and reduces the width of the uncertainty interval in which the true value of the coefficient may occur. The reconstruction value can be anywhere in that uncertainty interval. Practically, center of the uncertainty interval is used as reconstruction value.

5.3 The SPIHT Algorithm

Embedded Zerotree of Wavelet coefficients (EZW) [127] algorithm is a technique for coding wavelet transform coefficients. It claims simple, scalable, precise bit-rate control, better compression and an embedded bit stream as advantages.

Set partitioning in Hierarchical Trees (SPIHT) is in principle, similar to EZW, the major difference being the way the coefficients are partitioned and how the significant information is conveyed to the decoder. A brief description of SPIHT algorithm is given below. For complete information, refer to Said and Perlman [119]. The SPIHT algorithm is applied to coefficients resulting from the DWT. Normally, most of the energy in DWT is concentrated in the low frequency components. The variance decreases from the coarsest to finest levels of the sub-band pyramid. Moreover, there is spatial self-similarity between sub-bands, and, the coefficients are expected to be better magnitude-ordered if we move downward in the pyramid. A tree structure - *spatial orientation tree* - defines a spatial relationship on a hierarchical pyramid, similar to the *zerotree* structure of EZW. Spatial orientation (parent-offspring relation) is depicted in Fig.5.4, for four sub-band splitting. Each node of the tree corresponds to a pixel, and is identified by pixel coordinate. Its direct descendants (or offspring) correspond to 2×2 adjacent pixels of same spatial orientation in the next finer level of the pyramid. The Tree is defined in such a way that each node has either no or four offsprings, as shown in Fig5.4, arrows are oriented from the parent node to its four offspring. The coarsest level pixel of the pyramid are the tree roots, and are grouped into 2×2 adjacent pixels. However, their offspring branching rule is different, and in each group one of them (indicated by *) has no descendants.

Following symbols are used for the SPIHT algorithm:

1. $\mathcal{O}(i, j)$: set of offspring nodes of (i, j).

2. $\mathcal{D}(i, j)$: set of all descendants of (i, j).

3. $\mathcal{L}(i, j) = \mathcal{D}(i, j) - \mathcal{O}(i, j)$.

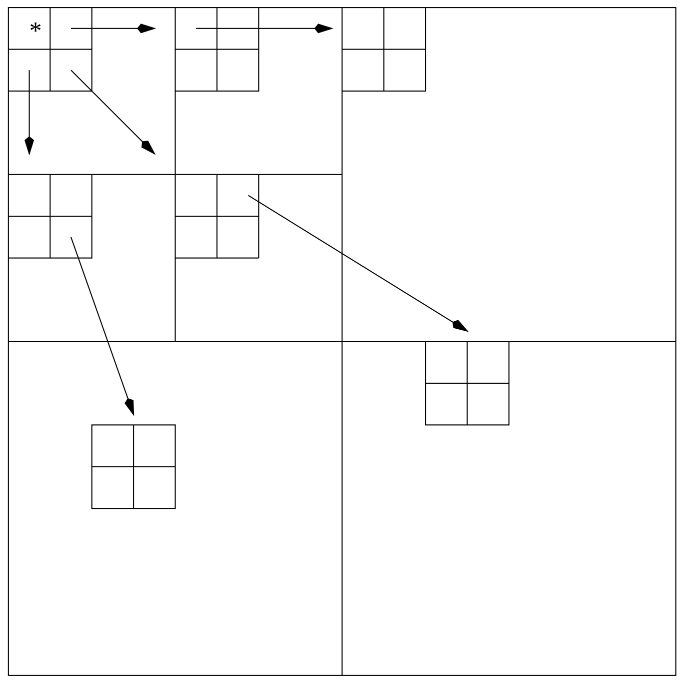

Figure 5.4: Parent offspring dependencies in the spatial orientation tree

4. $\mathcal{H}$: coordinates of all spatial orientation tree roots (nodes in the coarser pyramid level).

5. $S_n(i,j)$ is a function to indicate the significance of a set of coordinates $\{i,j\}$. If $max_{\{i,j\}}|c_{i,j} \geq 2^n$ then returns one else zero.

Then, except at the coarsest and finest levels, following relation holds good,

$$\mathcal{O}(i,j) = (2i,2j),(2i,2j+1),(2i+1,j),(2i+1,2j+1) \tag{5.1}$$

The SPIHT rules are then:

1. The initial partition is formed with the sets (i,j) and $\mathcal{D}(i,j) \in \mathcal{H}$,

2. If $\mathcal{D}(i,j)$ is significant then it is partitioned in to $\mathcal{L}(i,j)$ and plus the four single element sets with $(k,l) \in \mathcal{O}(i,j)$

3. If $\mathcal{L}(i,j)$ is significant then it is partitioned into the four sets $\mathcal{D}(k,l)$ with $(k,l) \in \mathcal{O}(i,j)$

Since the order in which the subsets are tested for significance is important, the significance information is stored in three ordered lists, called *list of significant sets* (LIS), *list* *of insignificant pixels* (LIP), and *list of significant pixels* (LSP). In all lists each entry is identified by a coordinate (i, j), which, in the LIP and LSP represents individual pixels, and in the LIS represents either the set $\mathcal{D}(i, j)$ or $\mathcal{L}(i, j)$. To differentiate between them we call the LIS entry of the type D if it represents $\mathcal{D}(i, j)$ and L type if it represents $\mathcal{L}(i, j)$.

Algorithm:

1. **Initialization:**

   Output $\lfloor log_2(max_{(i,j)}|c_{i,j}|)\rfloor$ where $c_{i,j}$ are wavelet transform coefficients; set the LSP as an empty list and add the coordinates $(i, j) \in \mathcal{H}$, to both LIP and to the LIS as type D entries.

2. **Sorting pass** (a) for each entry (i, j) in the LIP do: output $S_n(i, j)$; if $S_n(i, j) = 1$ then move (i, j) to the LSP and output sign of $c_{i,j}$

   (b) For each entry (i, j) in the LIS do:

   (i) if the entry is type D then output $S_n(\mathcal{D}(i, j))$ if $S_n(\mathcal{D}(i, j))$ then for each$(k, l) \in \mathcal{O}(i, j)$ output $S_n(\mathcal{D}(i, j))$ if $S_n(\mathcal{D}(i, j))$ then - for each $(k, l) \in \mathcal{O}(i, j)$ output $S_n(k, l)$

   – if $S_n(k, l) = 1$, add (k, l) to LSP and output sign of $c_{(k,l)}$;

   -if $\mathcal{L}(i, j) \neq 0$ then move (i, j) to the end of the LIS as an entry of type L

   (ii) if the entry is of type L then output $S_n(\mathcal{L}(i, j))$

   -if $S_n(\mathcal{L}(i, j)) = 1$ then add each $(k, l) \in \mathcal{O}(i, j)$ to the end of LIS as an entry of type D; remove (i, j) from the LIS

3. **Refinement pass** for each entry (i, j) in the LSP, except those included in the last sorting pass, output the n^{th} most significant bit of $|c_{i,j}|$

4. **Quantization-step update** decrement n by 1 and go to step 2.

5.4 Summary

We have discussed two methods for wavelet-based image and video compression schemes. The basic idea is to transform the image using orthogonal transformations, so that we have a large number of *insignificant* coefficients. The *significant* coefficients are coded. Both the methods discussed here are based on the zerotree principle - If the parent coefficient is insignificant with respect to a given threshold, then all the children/descendants are likely to be insignificant. For video, the technique is to estimate temporal (across frame) dependencies and code them, again in a transformed domain. Overall, this chapter gives the foundation for proposed zooming of compressed video scheme.